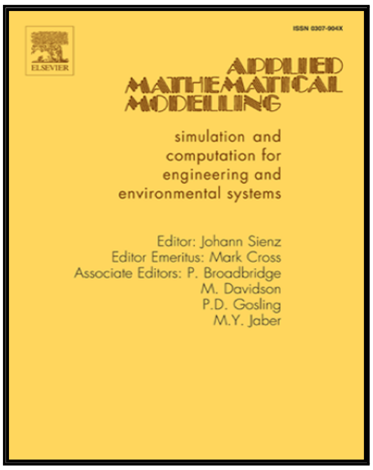


ISSN 0307-904X

APPLIED MATHEMATICAL MODELLING

simulation and
computation for
engineering and
environmental systems

Aerodynamic optimisation of a hypersonic reentry vehicle based on solution of the Boltzmann–BGK equation and Evolutionary Optimisation

B. Evans, S.P. Walton

PII: S0307-904X(17)30462-6
DOI: 10.1016/j.apm.2017.07.024
Reference: APM 11874

To appear in: *Applied Mathematical Modelling*

Received date: 12 March 2017
Revised date: 29 June 2017
Accepted date: 18 July 2017

Please cite this article as: B. Evans, S.P. Walton, Aerodynamic optimisation of a hypersonic reentry vehicle based on solution of the Boltzmann–BGK equation and Evolutionary Optimisation, *Applied Mathematical Modelling* (2017), doi: 10.1016/j.apm.2017.07.024

Highlights

- An original approach for solving optimal design of a space vehicle in rarefied hypersonic flow
- Novel flow solver based on the solution of the Boltzmann–BGK equation.
- First ever coupling of Boltzmann solver to an evolutionary optimiser
- Boltzmann–BGK solver rigorously tested on a number of examples

Aerodynamic optimisation of a hypersonic reentry vehicle based on solution of the Boltzmann–BGK equation and Evolutionary Optimisation[☆]

B. Evans[a,*], S. P. Walton[b]

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

Abstract

Over the past decade there has been a surge in the interest, both academic and commercial, in supersonic and hypersonic passenger transport. This paper outlines an original approach for solving the problem of optimal design and configuration of a space vehicle operating in rarefied hypersonic flow. The approach utilises a novel flow solver based on the solution of the Boltzmann–BGK equation. For the first time this solver has been coupled to an evolutionary optimiser to assist in navigation of the unfamiliar hypersonic design space.

The Boltzmann–BGK solver is rigorously tested on a number of examples and is shown to handle rarefied gas dynamics examples across a range of length scales. The examples, presented here for the first time, include: a Riemann–type gas expansion problem, drag prediction of a nano–particle and supersonic flow across an aerofoil. Finally the solver is coupled to the evolutionary optimiser Modified Cuckoo Search approach. The coupled solver–optimiser design tool is then used to explore the optimum configuration of the forebody of a generic space reentry vehicle under a range of design conditions.

In all examples considered the flow solver produces valid solutions. It is also found that the evolutionary optimiser is successful in navigating the unfamiliar design space.

Keywords: Boltzmann, Modified Cuckoo Search, Evolutionary Optimisation, rarefied gas flow, hypersonic

1. Introduction

1.1. Background to the Problem

Over the past decade there has been a surge in the interest, both academic and commercial, in supersonic and hypersonic passenger transport. Not since the great investment in the Concorde aircraft during the 1960s and 1970s has there been such a focus on supersonic air travel. Commercial companies ranging from Virgin Galactic [1] to Reaction Engines [2] and SpaceX [3] have set their sights on developing mainstream solutions for civilian passenger aircraft operating at hypersonic speeds. The aircraft and spacecraft being developed by companies such as these also have the shared aerodynamic mission requirement of being able to operate at high speed in rarefied atmospheric conditions. This presents a new challenge to computational fluid dynamics practitioners who have largely relied on continuum solvers, such as those based on the solution of the Euler or Navier–Stokes equations, for aerodynamic modelling analysis in the aerospace vehicle design cycle.

[☆] The intellectual property rights in the research data are asserted by the authors and their research partners.

[*] Corresponding Author: Tel.: +441792602129

Email addresses: b.j.evans@swansea.ac.uk (B. Evans), s.p.walton@swansea.ac.uk (S. P. Walton)

In the upper atmosphere, where this new generation of aerospace vehicles are designed to operate, often the low density environment necessitates a more fundamental approach to aerodynamic modelling. This paper presents a novel solver based on a full solution of the Boltzmann equation and shows how this solver can be effectively utilised in the context of design, by coupling it with an evolutionary optimisation algorithm. The combined Boltzmann solver and optimisation strategy are used to consider a range of hypersonic reentry design problems in two dimensions.

First the reasoning behind the requirement for a molecular solution approach, based on solution of the Boltzmann equation, is explained in Section 1.2. The solver will be introduced in Section 2 followed by the results of a series of solver validation studies presented in Section 3. The Modified Cuckoo Search evolutionary optimisation algorithm is outlined in Section 4 followed by a discretisation convergence study for the hypersonic test case in Section 5. Finally the results of the optimisation study and discussion are included in Section 6 followed by some concluding remarks and recommendations in Section 7.

1.2. Molecular Gas Dynamics

A gas flow may be modelled at either the macroscopic or microscopic level. At the macroscopic level the assumption must be made that the gas properties can be regarded as continuous variables. If this assumption is reasonable then the Navier–Stokes equations provide the appropriate mathematical description of the physics of the flow.

We can classify a gas flow in terms of the non–dimensional Knudsen number, Kn, defined as

$$Kn = \frac{\lambda}{L} \tag{1}$$

where λ is the mean free path of molecules in the flow, i.e. the average distance travelled between collisions, and L is some typical length scale in the flow. The continuum assumption is traditionally considered valid for $Kn < 0.1$[4]. However, this can be misleading if the global Knudsen number is defined using a single, fixed, L. In reality Kn varies throughout space, therefore we can more precisely define a local Kn by defining L as the scale length of the macroscopic gradients.

$$L = \frac{\rho}{d\rho/dx} \tag{2}$$

Where ρ is the fluid density. The Knudsen number limits on the conventional formulations to describe gas flows is shown schematically in Figure 1.

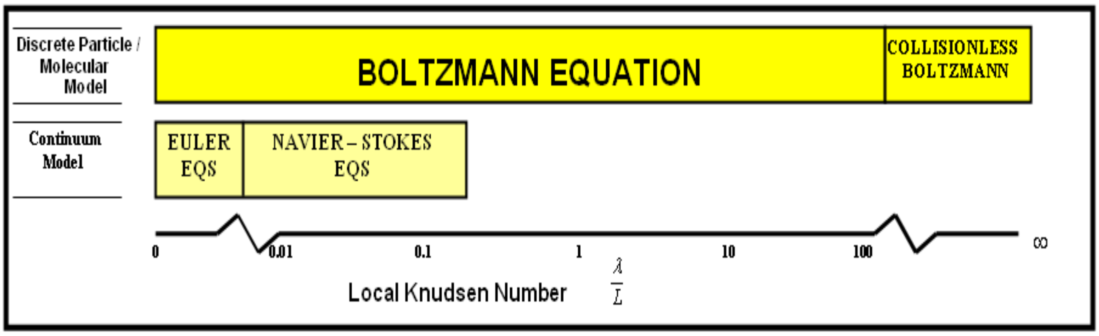


Figure 1: The Knudsen Regime

Figure 1 indicates that there is a significant range within the Knudsen regime where the traditional continuum equations are no longer valid. There are many hypersonic applications which involve low density and therefore high Kn flow, generally involving flight at high altitudes. For example, it has been noted that flow in the nose region of the space shuttle above 92km cannot be treated properly by purely continuum assumptions [5]. In particular the failings are illustrated when considering the velocity and temperature boundary conditions. The conventional viscous no–slip condition begins to fail and the flow velocity at a surface takes on some finite value. When considering the temperature slip condition, the continuum equations may still be applied, with modifications. As altitude continues to increase, a point is reached where we can no longer cling onto the continuum equations. Then we must seek solutions to the fundamental Boltzmann equation and think in terms of the underlying molecular kinetics. The air density eventually becomes low enough for the collisions between molecules to be ignored. For a space vehicle, this free molecular regime begins at an altitude of about 150km.

1.2.1. The Boltzmann-BGK Equation

It will be useful at this stage to introduce the Boltzmann equation and its BGK form simplification [6]. Note that its derivation can be found in the following references [4, 7, 8]. The full Boltzmann equation can be written as,

$$\frac{\partial(nf)}{\partial t} + \boldsymbol{c}.\frac{\partial(nf)}{\partial \boldsymbol{r}} + \boldsymbol{F}.\frac{\partial(nf)}{\partial \boldsymbol{c}} = \frac{1}{Kn}Q(f,f) \tag{3}$$

where $f = f(\boldsymbol{r}, \boldsymbol{c}, t)$ is the distribution function, n is the molecular number density, $\boldsymbol{F}$ describes any force fields that might be present, such as gravitational or electrostatic and $Q(f, f)$ is the term accounting for molecular collisions.

The Boltzmann equation is an integro–differential equation and analytical solutions are restricted to extremely simple applications [9, 10]. The term $Q(f, f)$ is a five-fold integral, in three spatial dimensions, and is the source of many of the difficulties in solving this equation [7]. As a result of this, simplification of the Boltzmann equation is usually focussed on some form of complexity reduction of the collision term. The most useful of these simplifications by Bhatnagar, Gross and Krook [6] relies on the assumption that departures from thermodynamic equilibrium are relatively small. This assumption leads to the simplification that the effect of molecular collisions in a gas is to return a non–equilibrium molecular velocity distribution to equilibrium at a rate proportional to the molecular collision frequency such that the Boltzmann-BGK equation becomes,

$$\frac{\partial}{\partial t}(nf) + \boldsymbol{c}.\frac{\partial(nf)}{\partial \boldsymbol{r}} + \boldsymbol{F}.\frac{\partial(nf)}{\partial \boldsymbol{c}} = \nu(\boldsymbol{r}, t)((nf_0) - (nf)). \tag{4}$$

where $\nu(\boldsymbol{r}, t)$ is a term proportional to the molecular collision frequency and f_0 is the local Maxwellian equilibrium distribution function defined as,

$$f_0(\boldsymbol{c}) = (\frac{\beta^3}{\pi^{3/2}})\exp(-\beta^2(\boldsymbol{c} - \boldsymbol{c_0})^2) \tag{5}$$

in three dimensions where $\boldsymbol{c_0}$ is the bulk velocity of the flow and $\beta = (2RT)^{-1/2} = \sqrt{m/(2kT)}$, R is the gas constant, T is the gas temperature measured in Kelvin, m is the molecular mass and k is the Boltzmann constant (taken as $1.380650524 \times 10^{-23}$ Joules/Kelvin).

A helpful derivation of the Maxwellian distribution function is provided by Vincenti and Kruger in [11]. 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 stream velocity, $\boldsymbol{c_0}$ and the temperature, T, which are obtained by taking integrals over f. However, computationally, the BGK term is significantly less demanding than the full Boltzmann equation right-hand side term. Solution of the Boltzmann–BGK equation will form the basis of all of the work presented in this paper.

2. Boltzmann-BGK Solver

2.1. Physical and Velocity Space Discretisation

The solution approach for the Boltzmann–BGK equation will briefly explained in this section. For a full description of the algorithm derivation the reader is referred to [12].

Since the single dependent variable, the velocity distribution function, in the Boltzmann–BGK equation is defined across both *physical space* and *velocity space*, a suitable discretisation approach must be adopted for both domains. The two dimensional physical space domain, Ω_r, is discretised into an unstructured assembly of discontinuous, linear, triangular elements with nodes at the vertices as shown in Figure 2. A discontinuous discretisation approach was chosen in order to naturally capture solution discontinuities such as shock waves. This scheme also lends itself naturally to parallelisation of the solver which is important because of the high demand placed on memory in the solution process.

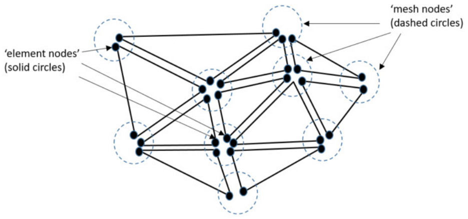
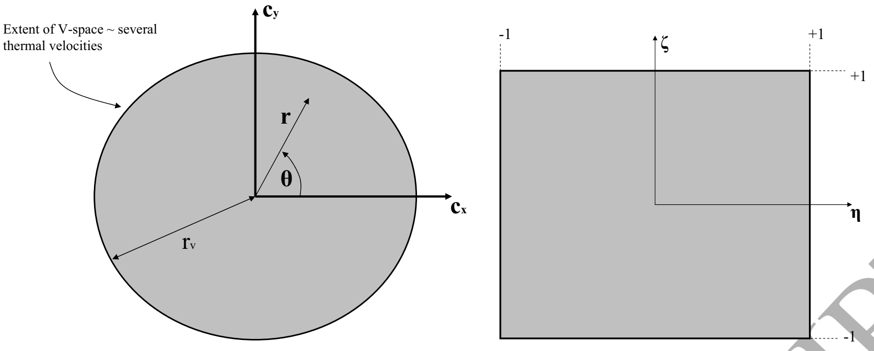


Figure 2: Physical space discretisation using an assembly of discontinuous, unstructured, triangular elements

The corresponding two dimensional velocity space domain, Ω_c, is, in principle, infinite in extent i.e. there is no natural limitation on the maximum speed of a molecule. However, a finite limit, r_v, must be placed on the radial extent of the domain. The 'rule of thumb' that was used on the velocity space limit in this work is that the the velocity space should be several times larger than the mean thermal (peculiar) molecular velocity [4]. The velocity space domain description is shown in Figure 3 (a).

As the velocity space domain contains no internal geometries it can be discretised as a single high order (spectral) element. This is advantageous for efficient integration over the domain. It is convenient to map the domain from polar coordinates in real space into a quadrilateral element in the (η, ζ) plane, shown in Figure 3 (b), using the transformation

$$\begin{aligned} \eta &= \frac{2r}{r_v} - 1 \\ \zeta &= \frac{\theta}{\pi} \end{aligned} \tag{6}$$



(a) (b)

Figure 3: (a) 'Real' Velocity Space Domain; (b) A Standard Quadrilateral Velocity Space Element in the (η, ζ) Plane

where (r, θ) are polar coordinates in real v–space and r_v is the radius of the v–space domain, i.e. the maximum molecular speed. A high order quadrature method is then applied to the element. In the η direction, a Lobatto quadrature is applied whereas in the ζ direction a constant spacing/constant weighting discretisation is applied. This results in a rotationally symmetric distribution of sampling points with no preferred radial direction, when the points are mapped back into real space. The coordinates of the quadrature points and the associated weightings in the (η, ζ) plane are shown in Figure 4 (a) for a (20 x 20) discretisation. If these points are then mapped back into real space, the (u, v) plane, the corresponding coordinates and weights are as shown in Figure 4 (b).

2.2. Nomenclature

Since the molecular number density, n, and the distribution function, f, always appear together in the Boltzmann–BGK equation it is possible to solve for the variable nf. There is no computational advantage of separating these variables since they appear together in the computation of the moment integrals at the post–processing stage. Throughout this paper the subscript/superscript notation $(nf)^m_{\boldsymbol{r},\boldsymbol{c}}$ will be used implying the value of nf at timestep m, spatial coordinate $\boldsymbol{r}$ and velocity coordinate $\boldsymbol{c}$.

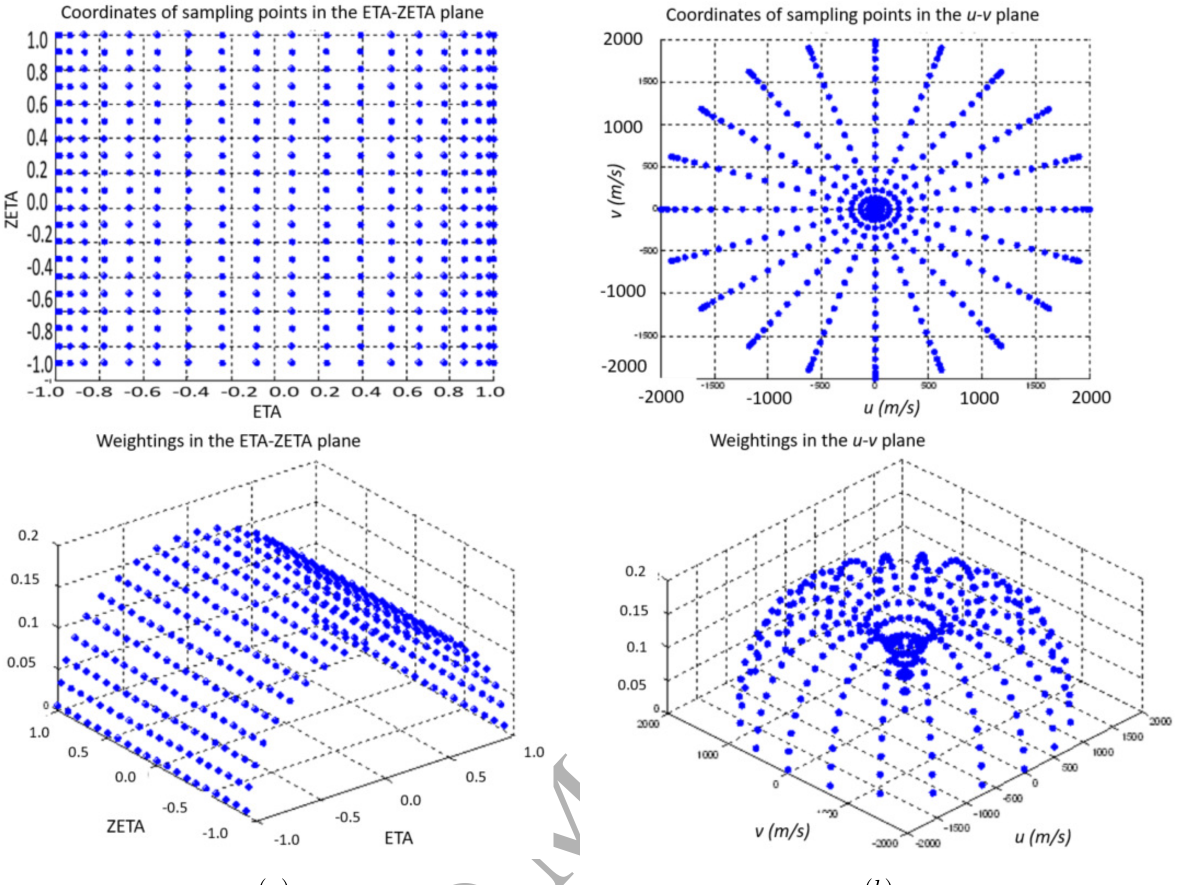
2.3. Equation Discretisation

A two–step discontinuous Taylor–Galerkin [13, 14] approach is implemented taking advantage of the approximation

$$(nf)_i^{m+\frac{1}{2}} = (nf)_i^m + (\frac{\Delta t}{2})Q^m - \frac{\Delta t}{2} c_i \frac{\partial (nf)_j}{\partial r_j}\Big|^m \tag{7}$$

where Q is computed as $Q = \nu(\boldsymbol{r}, t)((nf_0) - (nf))$ with f_0 representing the Maxwellian equilibrium distribution function, given in equation (5), modified to its two dimensional form, $f_0(\boldsymbol{c}) = \left(\beta^2/\pi\right) exp(-\beta^2(\boldsymbol{c} - \boldsymbol{c_0})^2)$ and ν is the molecular collision frequency determined as

$$\nu(\boldsymbol{r}, t) = \int_{-\infty}^{+\infty} \sigma_T |\boldsymbol{c} - \boldsymbol{c_1}| f \, d\boldsymbol{c_1} \tag{8}$$



(a) (b)

Figure 4: (a) Quadrature Coordinates and Weights in the (η,ζ) Plane; (b) Quadrature Coordinates and Weights in the (u,v) Plane

where σ_T is the total collision cross section [4]. If we make the assumption of hard sphere molecules, we obtain the simplest possible expression for the total collision cross section, $\sigma_T = \pi d^2$, where d is the molecular diameter.

2.3.1. Step 1

A piecewise–constant increment $\Delta(nf)_{re,\boldsymbol{c}}$ is computed on each physical space element according to

$$\Delta(nf)_{re,\boldsymbol{c}} = \frac{\Delta t}{2}\Sigma Q^m_{k,\boldsymbol{c}} N_k - \frac{\Delta t}{2} F^m_{ik} \frac{\partial N_k}{\partial r_i}\Big]_{re,\boldsymbol{c}} \tag{9}$$

where the summation k extends over the three nodes of element re, Δt is the global timestep governed by the Courant stability condition [12]. N_k is the standard, piecewise–linear finite element shape function associated with node k in physical space and

$$F^m_{ik,\boldsymbol{c}} = F_i\left((nf)^m_{k,\boldsymbol{c}}\right) = \boldsymbol{c}(nf)^m_{k,\boldsymbol{c}}. \tag{10}$$

The element fluxes at the half–timestep are then approximated by the piecewise linear discontinuous representation

$$F_i^{m+\frac{1}{2}}]_{re,\boldsymbol{c}} = F_i((nf)^m_{k,\boldsymbol{c}} + \Delta(nf)_{re,\boldsymbol{c}})N_k. \tag{11}$$

2.3.2. Second Step

A piecewise–linear approximation for $\Delta(nf)$ on each physical space element is assumed which is discontinuous at the element edges. The element nodal values of the solution increment over the complete timestep are determined according to

$$M_L]_{re}\Delta(nf)_{k,\boldsymbol{c}} = \Delta t M_L]_{re} Q^{m+\frac{1}{2}} + \Delta t \int_{\Gamma_{re}} F^{m+\frac{1}{2}}_{n,\boldsymbol{c}} N_k \mathrm{d}\Gamma_{re} - \Delta t \int_{\Omega_{re}} F^{m+\frac{1}{2}}_{ik,\boldsymbol{c}} \frac{\partial N_k}{\partial r_i} \mathrm{d}\Omega_{re,\boldsymbol{c}} \tag{12}$$

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 domain.

For inter–element edges, the direction of the flux across the edge must be calculated based on the convection velocity. The convection velocity is determined by the v–space node under consideration (i.e. the molecular velocity). If the flux is ‘into the element’, the integral in the first term on the RHS of (12) is given a value based on the corresponding upstream element edge flux. If the flux is ‘out of the element’, the same term is given a negative value based on the normal edge flux at that edge in the element. This ensures the local conservativeness of the scheme, since the fluxes of the distribution function are transferred directly from one element to the next.

In the BGK formulation shown here, the term ν is regarded as a collision frequency term and governs the rate at which the distribution function is restored to equilibrium. The form of the BGK collision term is such that the distribution function will be restored to equilibrium in a timescale,

$$\tau = O(\frac{1}{\nu}). \tag{13}$$

This places a further restriction on the allowable timestep size

$$\Delta t < \frac{1}{\nu}. \tag{14}$$

in addition to the Courant condition.

2.4. Boundary Condition Application

There are a range of potential approaches to deal with boundaries in the context of solution of the Boltzmann equation as detailed in the literature [15, 8, 16], often implemented in the context of the simplified lattice Boltzmann approach [17]. In this work, application of the boundary conditions in the algorithm is achieved by an appropriate modelling of $F^{m+\frac{1}{2}}_{n,\boldsymbol{c}}$ in the first term on the right-hand side of equation 12. Essentially three types of boundary need to be considered; inflow, outflow and wall. Note that a more extensive discussion of the boundary condition implementation can be found in [12].

2.4.1. Inflow

The assumption is made that the gas flow entering the physical space domain is in thermodynamic equilibrium and governed by a set of user–prescribed macroscopic properties. As such, for molecular velocities directed into the physical space domain at an inflow boundary the inter–element flux is constructed as

$$F^{m+\frac{1}{2}}_{n,\boldsymbol{c}} = \boldsymbol{c}.\boldsymbol{n}(\frac{\beta^2}{\pi})\exp(-\beta^2(\boldsymbol{c}-\boldsymbol{c_o})). \tag{15}$$

$(\beta^2/\pi)\exp(-\beta^2(\boldsymbol{c}-\boldsymbol{c_o}))$ is the Maxwellian distribution function in two dimensions. If the molecular velocity is directed out of the physical space domain the flux is computed as usual according to equation 10.

2.4.2. Outflow

At a physical space domain boundary flagged as an outflow the assumption is made that the gradient of macroscopic variables is zero perpendicular to the boundary and that the underlying molecular velocity distribution function is constant across the boundary. Therefore fluxes both into and out of the domain are constructed based on values inside the domain. This approach requires that outflow boundaries are positioned sufficiently far downstream for any unwanted reflection effects to be negligible.

2.4.3. Wall

At a wall the condition that must be enforced is zero mass flux across the boundary. In a molecular kinetic theory description this is expressed as

$$\int_{\Gamma_r}\int_{-\infty}^{+\infty} F_{n,\boldsymbol{c}}\,\mathrm{d}\boldsymbol{c}\,\mathrm{d}\Gamma_r = 0 \tag{16}$$

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 absorbed by the wall and remitted in equilibrium with wall, i.e. they are reflected back into the domain with a Maxwellian distribution based on the wall temperature. This is termed diffuse reflection. The remaining fraction, $(1-\alpha)$, are not absorbed by the wall and simply reflect directly back into the domain. This is termed specular reflection. These two models, for the interaction of a flux of molecules with a solid surface, was first suggested by Maxwell [44]. The term α is known as the 'absorption coefficient'. The distribution function of the net reflected flux of molecules is, therefore, constructed as

$$f(\boldsymbol{c},\boldsymbol{r},t) = (1-\alpha)Rf(\boldsymbol{c},\boldsymbol{r},t) + \alpha Mf(\boldsymbol{c},\boldsymbol{r},t), \qquad \text{for} \quad \boldsymbol{c}.\boldsymbol{n} \leq 0 \tag{17}$$

where

$$Rf(\boldsymbol{c},\boldsymbol{r},t) = f(\boldsymbol{c}-2\boldsymbol{n}(\boldsymbol{n}.\boldsymbol{c}),\boldsymbol{r},t) \tag{18}$$

$$Mf(\boldsymbol{c},\boldsymbol{r},t) = \eta(\boldsymbol{r},t)M_w(\boldsymbol{c}) \tag{19}$$

and $\boldsymbol{n}$ is the outward facing unit normal at the wall. If T_w is the wall temperature and R is the gas constant, then M_w is determined as

$$M_w = \exp(-\frac{c^2}{2\,R\,T_w}) \tag{20}$$

The parameter η is used to enforce the zero perpendicular mass flux condition using the method outlined in [12].

2.5. Post–processing for macroscopic variables

At each point in physical space, the mean value of any molecular quantity, Q, is defined as

$$\overline{Q} = \int_{-\infty}^{\infty} Qf(\boldsymbol{c})\,\mathrm{d}\boldsymbol{c}. \tag{21}$$

By setting Q to the appropriate molecular quantity, we can obtain the macroscopic properties as follows;

-density ρ: $Q = m$ where m is the molecular mass

-bulk velocity $\boldsymbol{v_i}$: $Q = \boldsymbol{c_i}$

-static pressure p_i: $Q = mc_i'^2$ where $\boldsymbol{c_i'} = \boldsymbol{c_i} - \boldsymbol{v_i}$ (thermal / peculiar velocity).

To determine the temperature, we simply use the definition of kinetic temperature,

$$T_k = \frac{p}{R\rho}. \tag{22}$$

We evaluate the integral in equation (21) by transforming the coordinate system from the real v–space coordinates to the (η, ζ) plane. Moving from Cartesian to polar coordinates gives us

$$\begin{aligned} u = r\cos\theta \quad &\Rightarrow \quad \frac{\partial u}{\partial r} = \cos\theta, \frac{\partial u}{\partial \theta} = -r\sin\theta \\ v = r\sin\theta \quad &\Rightarrow \quad \frac{\partial v}{\partial r} = \sin\theta, \frac{\partial v}{\partial \theta} = r\cos\theta. \end{aligned} \tag{23}$$

The Jacobian, J, of the transformation from Cartesian to polar coordinates is, therefore,

$$|J| = \begin{vmatrix} \frac{\partial u}{\partial r} & \frac{\partial u}{\partial \theta} \\ \frac{\partial v}{\partial r} & \frac{\partial v}{\partial \theta} \end{vmatrix} = r\cos^2\theta + r\sin^2\theta = r \tag{24}$$

so that

$$\overline{Q} = \int_{-\infty}^{\infty} Qf(\boldsymbol{c})\,\mathrm{d}\boldsymbol{c} \Rightarrow \int_{-\pi}^{+\pi}\int_{0}^{r_v} Qf(r,\theta)\,r\,\mathrm{d}r\,\mathrm{d}\theta. \tag{25}$$

The mapping from the real polar v–space to the (η, ζ) plane gives

$$\begin{aligned} r = \frac{r_v}{2}(\eta+1) \quad &\Rightarrow \quad \frac{\partial r}{\partial \eta} = \frac{r_v}{2}, \frac{\partial r}{\partial \zeta} = 0 \\ \theta = \zeta\pi \quad &\Rightarrow \quad \frac{\partial \theta}{\partial \eta} = 0, \frac{\partial \theta}{\partial \zeta} = \pi \end{aligned} \tag{26}$$

so that the Jacobian, J, of the transformation is

$$|J| = \begin{vmatrix} \frac{\partial r}{\partial \eta} & \frac{\partial r}{\partial \zeta} \\ \frac{\partial \theta}{\partial \eta} & \frac{\partial \theta}{\partial \zeta} \end{vmatrix} = \begin{vmatrix} \frac{r_v}{2} & 0 \\ 0 & \pi \end{vmatrix} = \frac{\pi r_v}{2}. \tag{27}$$

We can rewrite equation (25) as

$$\overline{Q} = \int_{-1}^{+1}\int_{-1}^{+1} Qf(\eta,\zeta)\frac{\pi r_v^2}{4}(\eta+1)\,\mathrm{d}\eta\,\mathrm{d}\zeta \tag{28}$$

so that the integral in equation (21) may be evaluated using the high order quadrature method as

$$\overline{Q} = \frac{\sum_{i=1}^{N} w_i Q(nf)_i |J|}{\sum_{i=1}^{N} w_i (nf)_i |J|} \tag{29}$$

where the summations are over all the v–space sampling points in the discretisation, w_i is the weighting associated with the point and $|J| = \pi r_v^2(\eta_i + 1)/4$.

The macroscopic variables are calculated at the element nodes. For post–processing of the results, data defined at the mesh nodes is required. This is achieved using the simple element weighted averaging procedure.

$$\overline{Q_m} = \frac{\sum_{i=1}^{n} A_i \overline{Q_i}}{\sum_{i=1}^{n} A_i} \tag{30}$$

where $\overline{Q_m}$ is the value of the macroscopic variable at the mesh node, the summations are over each discontinuous node, i, meeting at the mesh node, A_i is the area of the element associated with discontinuous node i and $\overline{Q_i}$ is the value of the macroscopic variable at discontinuous node i.

2.6. Parallelisation

Due to the large memory requirement involved in discretising both physical and velocity space domains, parallelisation of the algorithm is required for realistic problems to be tackled (even in two dimensions). The code has therefore been parallelised via physical space domain decomposition utilising the METIS family of graph partitioning libraries [18]. For more details the reader is referred to [12].

3. Solver Validation Study

A range of test cases were considered to validate the performance and capabilities of the Boltzmann solver prior to commencing the optimisation study. These cases considered flow problems across a range of Knudsen number, Mach number and absolute length scales.

3.0.1. High Knudsen, Low Mach Gas Expansion

The first validation case considered was specifically chosen as a problem that would be challenging for a traditional continuum solver; the problem of modelling Argon gas initially trapped within a tube expanding into a vacuum. This problem was set up such that the Knudsen number is based on the initial gas density and reference length, L, the initial extent of the gas as shown in Figure 5. This figure also shows the physical space mesh on which this simulation was run containing 1,538 triangular elements (i.e. 3 x 1538 = 4614 discontinuous nodes) and 961 nodes in the velocity space discretisation. This results in a total of 4614 x 961 = 4,434,054 degrees of freedom (DoFs) in the system. This resolution of grids was chosen following a mesh convergence study (detailed in [12]).

A summary of the computational parameters used to conduct this transient simulation are shown in Table 1. Note that the term β used to non-dimensionalise time and velocity is defined as $\beta = (2\bar{R}T)^{-1/2} = \sqrt{m/(2kT)}$ where $\bar{R}$ is the universal gas constant, T is the initial gas temperature (in Kelvin), m is the molecular mass and k is the Boltzmann constant ($1.380650524e^{-23} J/K$). It is evident that, despite this being a relatively simple problem run on a coarse physical space mesh the computational demand is high.

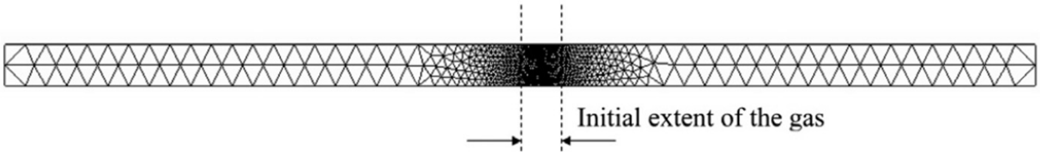


Figure 5: Physical space mesh used for the gas expansion case

Parameter	Value
Kn	100
p-space mesh	1538 elements (4614 discontinuous nodes)
v-space mesh	961 nodes
DoFs	4614 x 961 = 4,434,054
r_v	1,000 m/s
compute	8 cores
wallclock runtime	6hrs to $t/\beta L = 8.0$

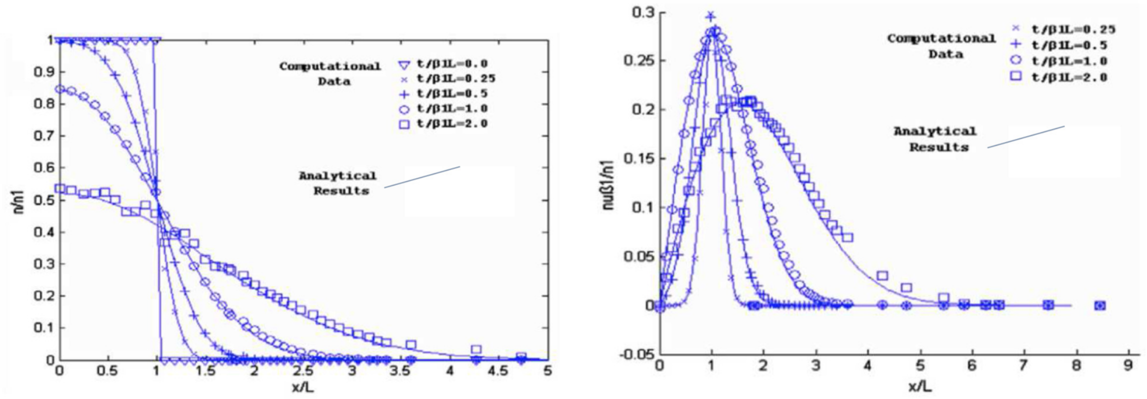
Table 1: Summary of computational requirements for the gas expansion validation case

The results in Figure 6 show excellent agreement with the analytical solution [4] to this problem up until a non-dimensional time, $t/\beta L$, of 2.0. Beyond this time oscillations in the solution were observed (evident in Figure 6 at $t/\beta L = 2.0$) which are due to reflection effects from the farfield outflow boundaries. This problem could be remedied by increasing the lateral extent of the physical space domain.

3.0.2. Low Knudsen Supersonic Flow Simulation

The second validation case study was chosen to test the solver's ability to simulate continuum, supersonic flow phenomenon and in particular the thermal effects across a shock wave. A test case was chosen at low Knudsen number, within the continuum regime, at a low supersonic Mach number to determine if the correct stand-off bow shock location and associated temperature jump could be predicted. The computational parameters used for this test case are shown in Table 2. Note that computationally this is far more expensive than a typical continuum solver, requiring 16 cores of a PC cluster running for 24 hours to achieve a solution converged to three significant figures in the lift and drag coefficient predictions.

Figure 7 shows the predicted pressure contour, Mach number and temperature predictions for this case resulting from the Boltzmann solver. Note that no physical space mesh refinement was carried out along the shock front and therefore the shock is artificially 'smeared'. However, the results are still in excellent agreement with continuum solver predictions [7] and the computed temperature jump across the shock of 1.29 is very close to the analytical prediction for the temperature jump across a Mach 1.5 normal shock given by ($T_{downstream}/T_\infty = [2\gamma M^2 - (\gamma-1)][(\gamma-1)M^2+2]/(\gamma+1)^2M^2 = 1.32$. The stand-off distance of the bow shock of 0.25c (where c is the chord length) is in good agreement with the range of solutions in [19] for a freestream Mach number, M_∞ of 1.5. These results gave confidence that the solver has the inherent ability to predict shock wave position and strength effectively without any requirement for additional numerical stabilisation.

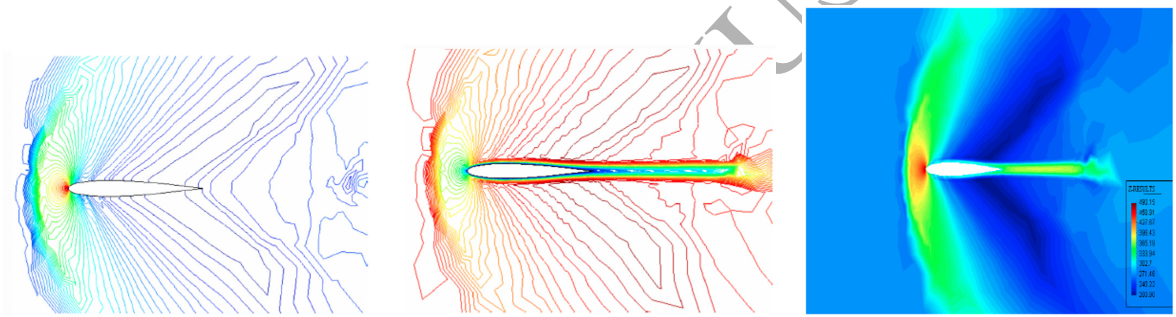


(a) (b)

Figure 6: Comparison of Boltzmann solver results with analytical solution for gas expansion case showing: (a) normalised molecular number density; (b) normalised centreline velocity

Parameter	Value
Kn	0.002
Re	10,000
M_∞	1.5
aerofoil	NACA0012
p-space mesh	7640 elements (22920 discontinuous nodes)
v-space mesh	400 nodes
DoFs	22920 x 400 = 9,168,000
r_v	2,000 m/s
α	0.9
compute	16 cores
No. timesteps	30,000 to steady state
wallclock runtime	24hrs

Table 2: Summary of computational requirements for the supersonic aerofoil case



(a) (b) (c)

Figure 7: Boltzmann solver results for a NACA0012 aerofoil at freestream Mach 1.5 showing: (a) constant pressure contours; (b) constant local Mach contours; (c) constant temperature contours

Parameter	Value
Kn	0.0214
Re	0.25 - 2.0
p-space mesh	29,261 elements (87,783 discontinuous nodes)
v-space mesh	1,600 nodes
DoFs	87,783 x 1,600 = 140,452,800
r_v	2,000 m/s
α	0.9
compute	16 cores
No. timesteps	30,000 to steady state
wallclock runtime	72hrs

Table 3: Summary of computational requirements for the nano–particle drag prediction test case

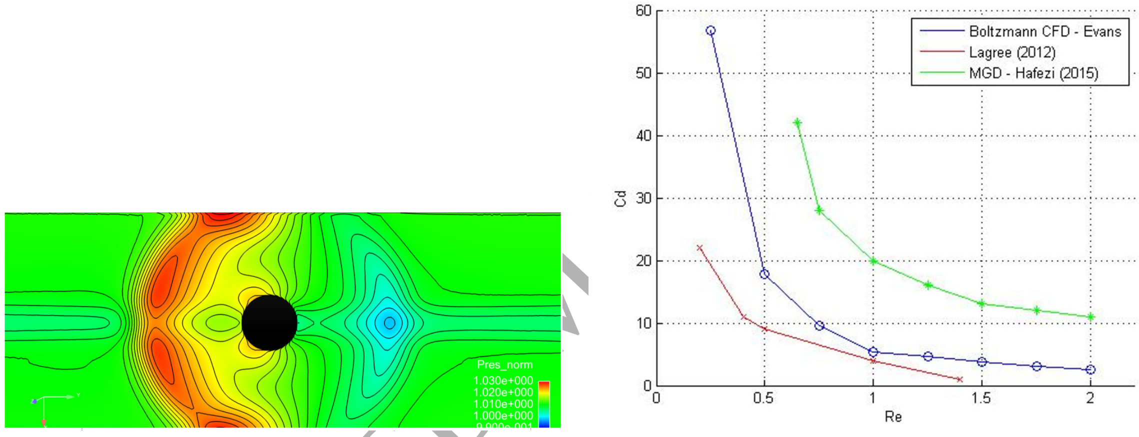
3.0.3. Transition Knudsen Nano-scale Drag Prediction

The final validation test case considered is a notoriously challenging [20, 21] computational problem: the prediction of the drag forces on a nano–particle. In this case the Knudsen number is in the transition range between continuum flow and true molecular flow ($Kn = 0.0214$) because of the small length scales involved. In this case a 2D circular particle of diameter $2nm$ was considered in the centre of a channel of width $8nm$ at a range of Reynolds' numbers from 0.2 to 2.0. The computational parameters used for this are shown in Table 3.

There is no analytical solution or accurate experimental data available for cases at this length scale but there is a wide range of results from computational predictions in the literature for drag on particles at this scale [20, 21]. Figure 8 shows a comparison of the drag coefficient predicted by the Boltzmann–BGK solver across the considered Re range with the molecular gas dynamics (MGD) approach used by Hafezi et al [20] and the continuum (Navier–Stokes) boundary condition correction approach of Lagree et al [21]. The results from this solver lie in the range between these two approaches. This, again, gives confidence that, with no adaptation of the algorithm, this solver is capable of solving problems across a range of length scales from nano to macro.

3.1. Solver Validation Summary

The Boltzmann–BGK solver outlined in Section 2 has been rigorously tested on range of flow problems across a range of Knudsen, Reynolds and Mach numbers including flows involving free surfaces and shock waves. For more details on the solver validation in the continuum regime, including comparison with continuum solver results, the reader is referred to [7, 12]. Having tested the algorithm on on these non-continuum test cases the main focus of this work was undertaken (i.e. to couple the Boltzmann–BGK solver to an optimisation algorithm in order to study design considerations for vehicles operating under hypersonic conditions in the upper atmosphere). This work is outlined in the remainder of this paper.



(a) (b)

Figure 8: Results for $Kn = 0.024$ flow over a nano–particle showing: (a) pressure distribution (normalised by freestream at $Re = 2.0$); (b) drag prediction comparison with MGD results (Hafezi) and Navier–Stokes results (Lagree)

4. Optimisation

There is a long history of applying classical gradient based optimisation methods to shape optimisation [22]. Gradient based techniques take an initial design as an input and determines the gradient of the objective function we wish to minimise with respect to each design variable. The primary advantage of these methods is a high computational efficiency due to a high rate of convergence [24]. This speed has enabled these techniques to make an impact in real design cycles [25]. Despite this success, it is well understood that gradient based techniques are highly dependent on the initial design. It can be shown that given a bad enough initial design an otherwise obtainable optimum solution can not be found [24]. This motivates the application of gradient free optimisation techniques [26].

During the 1950s and 1960s, computer scientists investigated the possibility of applying the concepts of evolution as an optimisation tool for engineers and this gave birth to a subclass of gradient free methods called genetic algorithms (GA) [27]. Many algorithms have been developed that have been inspired by nature, for example particle swarm optimisation (PSO) [28], differential evolution (DE) [29] and Cuckoo Search (CS) [30].

Evolutionary techniques are heuristic in nature and use large populations of agents to search the design space. In our application each agent represents a particular aerodynamic design. The objective function can be calculated at the position of each agent to define its fitness, then a series of rules are followed to move the agents, hopefully, towards the global minimum. These rules are split into two broad categories. Mutation operations which search new areas of the design space and crossover operations which search areas where good designs have already been found. Each iteration, or generation, these rules are repeated. In practice testing for global convergence is not straight forward and these types of algorithms tend to be left to run for a given time. A successful gradient free optimisation technique will find a balance between exploring new parts of the search space and refining areas of the search space where current information suggests the minimum might be. The objective function will need to be evaluated for each agent each generation, which results in a large number of objective function evaluations. This means that the computational efficiency of gradient free techniques is often inferior to gradient based methods [31]. The problem is particularly significant when considering applications where a single objective function evaluation represents a significant computational cost, such as the one presented here.

4.1. Modified Cuckoo Search

CS was introduced in 2009 by Yang and Deb [36] taking inspiration from the reproductive strategy of cuckoos. The key component of the strategy, mimicked by CS, is the aggressive behaviour of cuckoos which lay eggs in the nests of other species. If the cuckoo egg differs significantly from the eggs of the host bird, there is the chance that the host may abandon the nest and the eggs. Cuckoo eggs have slowly evolved to prevent this by mimicking the patterns of the eggs belonging to other local species of birds [30].

In the CS algorithm a population of agents, or eggs, are generated to each represent a particular set of variables. The objective function is evaluated for each egg, and the eggs are placed in nests. In sequence each cuckoo takes a random Lévy flight, which is an optimum random search pattern [37], starting at a random egg and produces a new egg at its new location. The location corresponds to a new set of variables for the objective function. The fitness of the new egg is determined by evaluating the objective function at that location. The cuckoo picks a nest at random and the current egg in that nest is replaced if the new egg has a better fitness. To model the risk of a host bird discovering eggs with poor fitness, a fraction of the worst solutions are discarded at each generation and new solutions are randomly generated.

Although this method can eventually reach a better optimum solution than other popular gradient free techniques, much like in natural evolution the number of generations, and hence objective function evaluations, required to reach this optimum can be very large [30]. The eggs are essentially searching independently from one another and a certain percentage of the failing searches are restarted each generation. This motivated the development of modified cuckoo search (MCS) [26].

MCS introduces exploitation into the algorithm by exchanging information between the eggs found so far. A search is performed along a line between two eggs, selected at random from a group making up a fraction of the best. This modification resulted in a rapid convergence to the global minimum, even at high numbers of dimensions [26].

Both CS and MCS have been successfully applied to a number of problems [38, 39]. More recently a number of improvements to MCS were made [40] which are used in this paper. The effect of these improvements were tested on a number of rotated and shifted benchmark problems, showing that MCS out performs a number of other optimisation algorithms. An open source version of the MCS algorithm used in this paper is available [41]. The tuning parameters used are the same as reported in [40]. Due to the high computational cost of the solver a population size of 10 was selected and the optimiser was run for 10 generations. The initial generation was randomly generated using Latin hypercube sampling of the search space.

4.2. Objective Functions

In this paper we consider four objective functions, $d_1(\boldsymbol{p})$, $d_2(\boldsymbol{p})$, $d_3(\boldsymbol{p})$ and $d_4(\boldsymbol{p})$, which will be defined in this section. In all cases the optimiser aims to minimise the value of the objective function. The objective functions are defined as follows:

$$d_1(\boldsymbol{p}) = C_d \tag{31}$$

$$d_2(\boldsymbol{p}) = -C_d \tag{32}$$

$$d_3(\boldsymbol{p}) = T_{max} \tag{33}$$

$$d_4(\boldsymbol{p}) = \begin{cases} -C_d + \frac{(T_{max} - T_c)}{T_c} & T_{max} > T_c \\ -C_d & T_{max} \leq T_c \end{cases} \tag{34}$$

Where $\boldsymbol{p}$ are the shape parameters (to be defined in Section 4.3), T_{max} is the maximum local surface temperature on the body and T_c is a designer selected penalty based on the maximum temperature, e.g. T_c could be set by the designer such that if $T_{max} > T_c$ the optimiser will try to reduce the temperature. The results of optimising $d_1(\boldsymbol{p})$ and $d_2(\boldsymbol{p})$ will tell us the rough range of C_d achievable with the parameterisation selected. The work flow we wish to illustrate is designing a shape which finds a balance between maximising the drag and minimising the temperature. $d_4(\boldsymbol{p})$ is a mathematical expression of this objective. Since we are working in two dimensions selecting a value for T_c is not straight forward. Optimising $d_3(\boldsymbol{p})$ will determine the lowest achievable maximum temperature which will be used to select a realistic value for T_c.

4.3. Shape Parameterisation

Given a priori knowledge of the high computational expense of direct Boltzmann equation solution, it was decided to limit the study in this case to two dimensions. Note that future work to extend the capabilities presented in this paper to 3D is discussed in Section 7. For the same reason, it was also beneficial to limit the dimensionality of the design space as far as possible in order to keep the appropriate number of agents used in the evolutionary optimisation algorithm to a minimum.

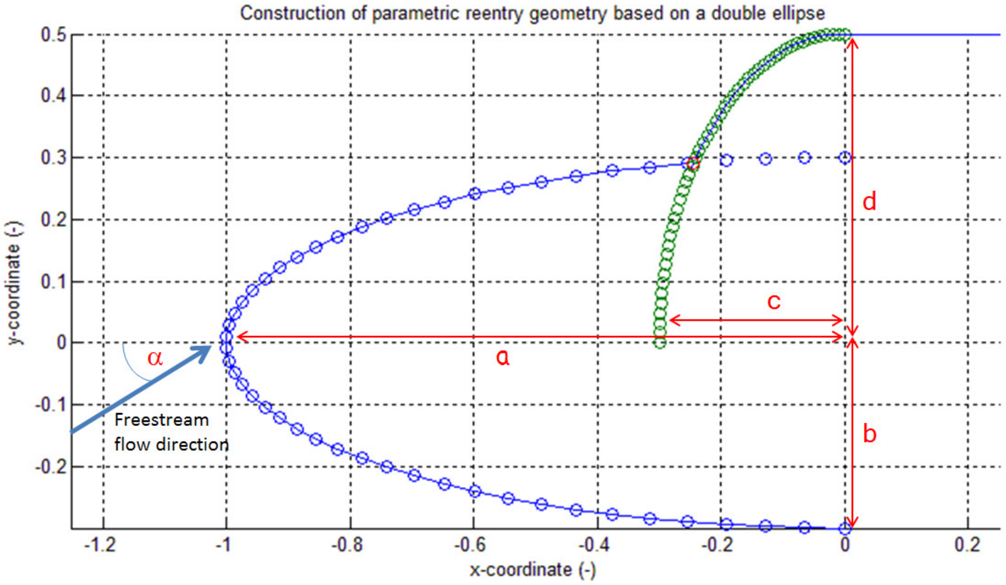


Figure 9: Definition of parameters used in the geometry parameterisation and optimisation Plane

It was decided to construct a generic ‘reentry vehicle type’ geometry out of a double ellipse constructed as shown in Figure 9 focussing the front portion of the vehicle. Note that this is usually the most critical geometric aspect of a reentry vehicle (particularly in terms of heating) during the hypersonic reentry phase of a space mission. Four geometric parameters were used to construct the geometry: the major radius of the body ellipse, a, the minor radius of the body ellipse b, the minor radius of the cockpit ellipse, c and the major radius of the cockpit ellipse, d. As well as these four geometric parameters, the angle of attack of the oncoming freestream flow, α was included as a fifth design parameter. This double-ellipise geometry is then extruded downstream before it ‘meets’ the outlet boundary. This is done in order to avoid unwanted, and non-physical, refelection effects from the outlet on the part of the geometry under consideration. Note that it is only the integrated forces on the section of geometry detailed in Figure 9 that contributes to the objective function in the optimisation study.

Table 4 shows the minimum and maximum values of each these parameters used to define the extent of the design space explored. These were chosen in an attempt to minimise the number of invalid geometries that would be created whilst ensuring a reasonably large, but sensible, design space.

Parameter	Min	Max
a	1.0	2.0
b	0.2	0.5
c	0.1	0.7
d	0.6	1.0
α	-60°	$+60^{\circ}$

Table 4: Summary of computational requirements for the nano–particle drag prediction test case

4.4. Coupling the Optimiser with the Solver

A key advantage of a gradient free optimiser, such as modified cuckoo search, is that it is straightforward to couple it with a solver. The optimiser can be treated as a black box, independent of the problem being solved. It generates a population of designs, in the form of parametrisations as discussed above. This generation of designs is passed to the mesh generator and solver then solved in parallel. The relevant objective function is then calculated, for each design, and passed back to the optimiser. Using this information the optimiser then generates the set of designs which make up the next generation and passes these back to the mesh generator and solver. The process iterates for a predetermined time.

5. Mesh Convergence Study

The freestream conditions for the hypersonic vehicle design optimisation study were chosen as $M_\infty = 25$, $Kn_\infty \approx 20$ and $Re_\infty \approx 5.0$ to correspond with the typical conditions that a space reentry vehicle might experience at an altitude approximately 100km [43]. For the computation of both Reynolds and Knudsen number the reference length was chosen as unity (the baseline length of the body major radius parameter). It was decided that, due to the computational expense of solution of the Boltzmann–BGK equation, relatively low fidelity (in terms of numerical convergence) solutions would be sought during the optimisation phase given that the optimiser is simply making decisions based on simply whether one design is 'better' than another and not taking into account absolute value of the fitness function (see Section 4).

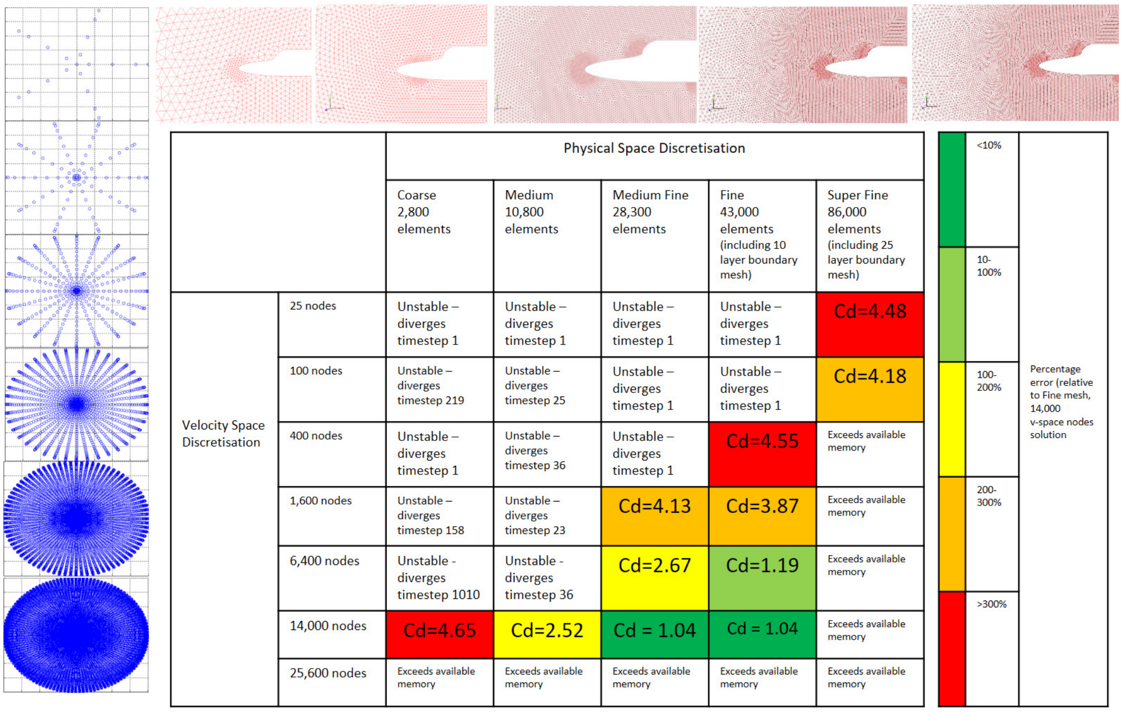
Since solution of the Boltzmann equation requires disretisation of both physical space and velocity space, a matrix of solutions must be generated to understand the interaction of the level of spatial disretisation in both domains and the interaction between these. It was decided that convergence based simply on the total drag force acting on the vehicle would be considered as this is the main focus of the design optimisation cases. Note that the when quoting drag coefficients in the results below, the drag is taken to be the normalised resultant force acting on the body (i.e. the combination of lift and drag force) since a reentry vehicle can orientate itself arbitrarily on reentry to either maximise or minimise drag. The drag coefficent is therefore computed as,

$$C_d = \frac{\sqrt{F_x^2 + F_y^2}}{q_\infty L} \tag{35}$$

where F_x is the force acting in the positive x direction and F_y is the force acting in the positive y direction in relation to the geometry description in Figure 9, q_∞ is the freestream dynamic pressure and L is the standard length scale used to normalise force coefficients which in this case was set to 1.0. Figure 10 shows the result from this study indicating the predicted net drag coefficient predicted for various combinations of physical and velocity space disretisation. Note that at low levels of spatial disretisation the solver algorithm became unstable and running at higher resolutions than those considered was not possible due to computational resource (memory) restrictions.

Figure 11 provides a visualisation of the simulated flow field on the 'baseline' geometry used for the mesh convergence study. These results were achieved by running on the 'fine' physical space mesh (see Figure 10) and using 14,000 velocity space nodes.

Assuming that the (three significant figure) converged drag solution on the baseline geometry considered is 1.04, the combination of 'Medium Fine' resolution physical space mesh with 14,000 nodes in the velocity space mesh was chosen as the appropriate compromise of sufficient spatial disretisation for minimum computational requirement. Note that at this resolution steady state convergence was achieved within 24hrs running on 32 cores of an Intel Xeon Sandy Bridge based PC Cluster using Infiniband for interconnect. A summary of the modelling parameters used and computational resource requirements for a typical simulation within the optimisation study is shown in Table 5. Note that since the purpose of these simulations was to rank the designs in order dependent on the pre–defined objective function convergence was assumed once the C_d value had settled to 2 significant figures (sf).



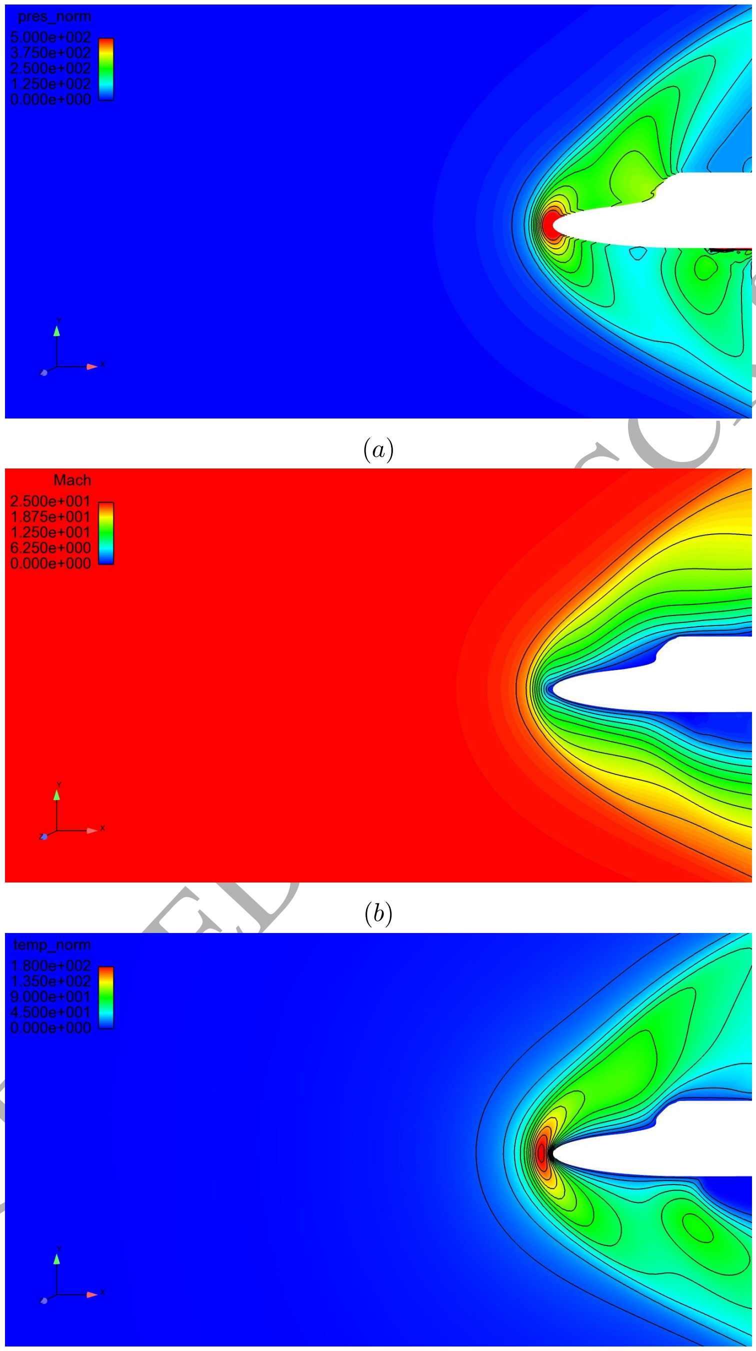
		Physical Space Discretisation				
		Coarse 2,800 elements	Medium 10,800 elements	Medium Fine 28,300 elements	Fine 43,000 elements (including 10 layer boundary mesh)	Super Fine 86,000 elements (including 25 layer boundary mesh)
Velocity Space Discretisation	25 nodes	Unstable – diverges timestep 1	Unstable – diverges timestep 1	Unstable – diverges timestep 1	Unstable – diverges timestep 1	Cd=4.48
	100 nodes	Unstable – diverges timestep 219	Unstable – diverges timestep 25	Unstable – diverges timestep 1	Unstable – diverges timestep 1	Cd=4.18
	400 nodes	Unstable – diverges timestep 1	Unstable – diverges timestep 36	Unstable – diverges timestep 1	Cd=4.55	Exceeds available memory
	1,600 nodes	Unstable – diverges timestep 158	Unstable – diverges timestep 23	Cd=4.13	Cd=3.87	Exceeds available memory
	6,400 nodes	Unstable - diverges timestep 1010	Unstable - diverges timestep 36	Cd=2.67	Cd=1.19	Exceeds available memory
	14,000 nodes	Cd=4.65	Cd=2.52	Cd = 1.04	Cd = 1.04	Exceeds available memory
	25,600 nodes	Exceeds available memory	Exceeds available memory	Exceeds available memory	Exceeds available memory	Exceeds available memory

🟩	<10%	Percentage error (relative to Fine mesh, 14,000 v-space nodes solution
🟩	10-100%	
🟨	100-200%	
🟧	200-300%	
🟥	>300%	

Figure 10: Disretisation convergence study results. This figure shows the impact of physical space and velocity space resolution on the predicted (2D) drag coefficient for a generic ‘double ellipse’ geometry at a freestream Mach number of 25, Knudsen number of 100 and Reynolds number of 1.0 (both Kn and Re lengthscales set as $L = 1.0$). Across the top of the figure are depictions of the physical space meshes used for the study with increasing resolution from left to right. On the left of the table are the various velocity space discretisations used. Note that successful simulations are colour–coded according to their error relative to the ‘converged result’.

Parameter	Value
Kn	1.0
Re	1,000
$Mach$	25
Geometry	Double-ellipse
p-space mesh	28,300 elements (84,900 discontinuous nodes)
v-space mesh	14,000 nodes
DoFs	84,900 x 14,000 = 1.2 billion
r_v	13,000 m/s
α	0.9
compute	32 cores
No. timesteps	2,000 to (2sf drag) steady state
wallclock runtime	24hrs

Table 5: Summary of typical single simulation computational requirements for the hypersonic vehicle reentry optimisation cases



(*a*)

(*b*)

(*c*)

Figure 11: Solution of the case used for the mesh convergence study on the 'fine' physical space mesh and using 14,000 velocity space nodes showing: (a) normalised pressure distribution, p/p_∞; (b) Mach number distribution; (c) normalised pressure distribution, T/T_∞

6. Results

All of the optimisation cases considered were run at a freestream Mach number, M_∞ of 25, Knudsen number, Kn_∞ of 20 and Reynolds number, Re_∞ of 5.0. The length-scale used to compute both Kn_∞ and Re_∞ was set to 1.0. These conditions relate to typical conditions encountered by a space reentry vehicle, such as the space shuttle, at 100km altitude.

6.1. Case 1 - Drag Minimisation

6.1.1. Optimiser Behaviour

Figure 12 shows the first generation of designs for the drag minimisation example. The first generation is a random sampling of the design space prior to any optimisation iterations. The lowest value in the first generation was 0.92831. The behaviour of the optimiser is presented in Figure 13. An improvement is made to the best design in the first generation. Figure 14 shows the final population of designs, the lowest value for drag is 0.398.

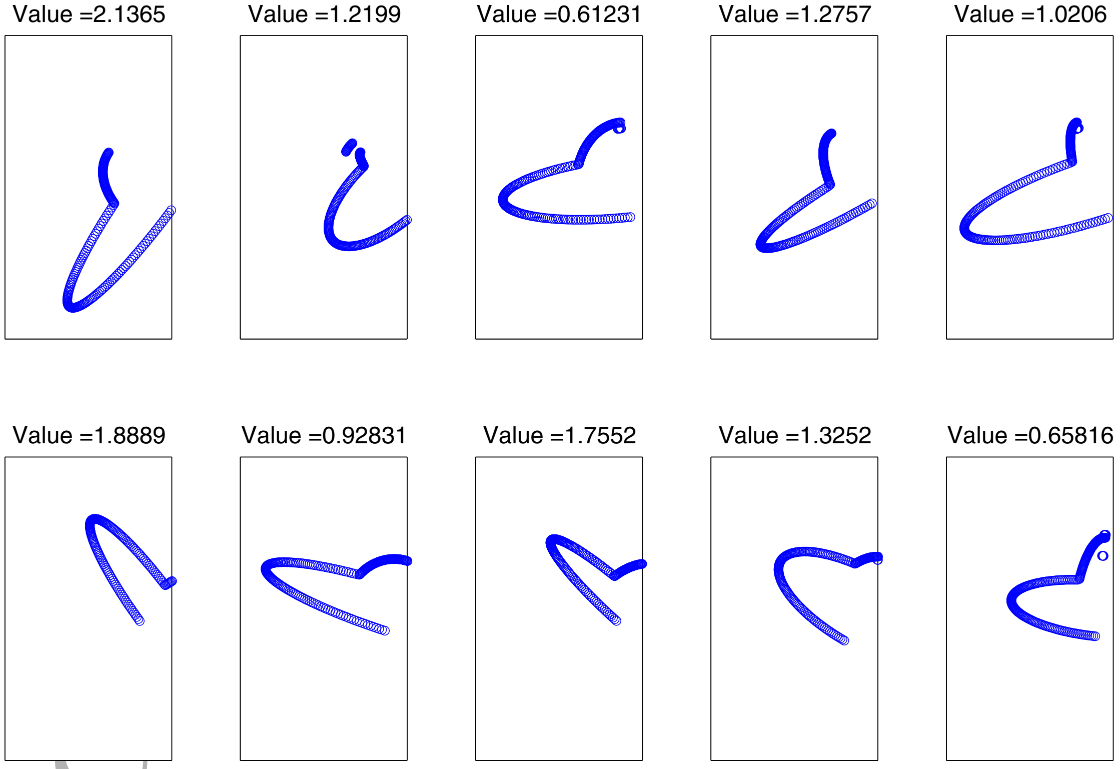


Figure 12: Geometries and objective function values at the end of generation 1 for drag minimisation. These are raw outputs from the parameterisation, some spurious points appear which are rejected during mesh generation.

6.1.2. Optimised Geometry

Figure 15 shows the design parameters and simulated flow around the reentry vehicle optimised for minimum drag. This was the first case chosen as there is an intuitive solution to this problem as one would expect the

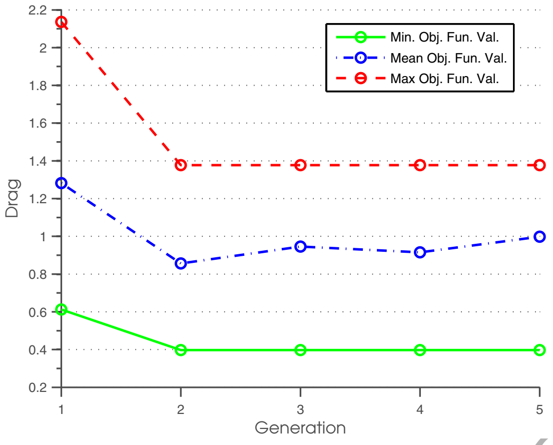


Figure 13: Objective function value history for the drag minisation case

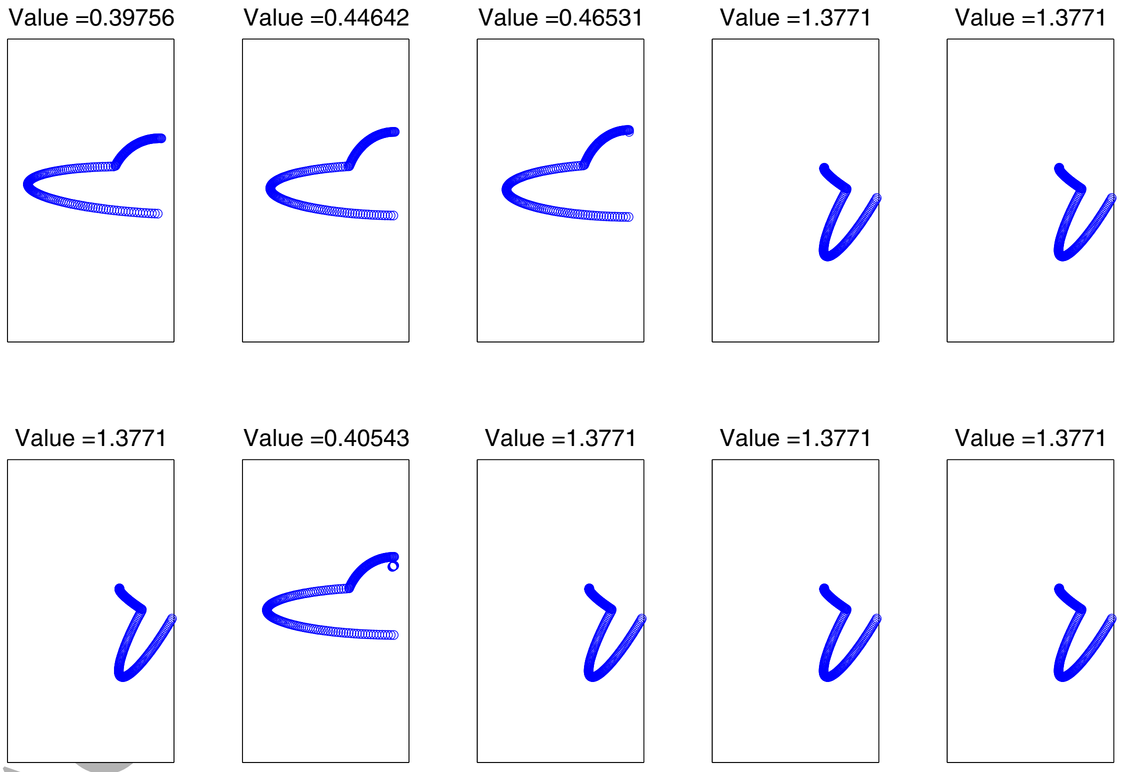
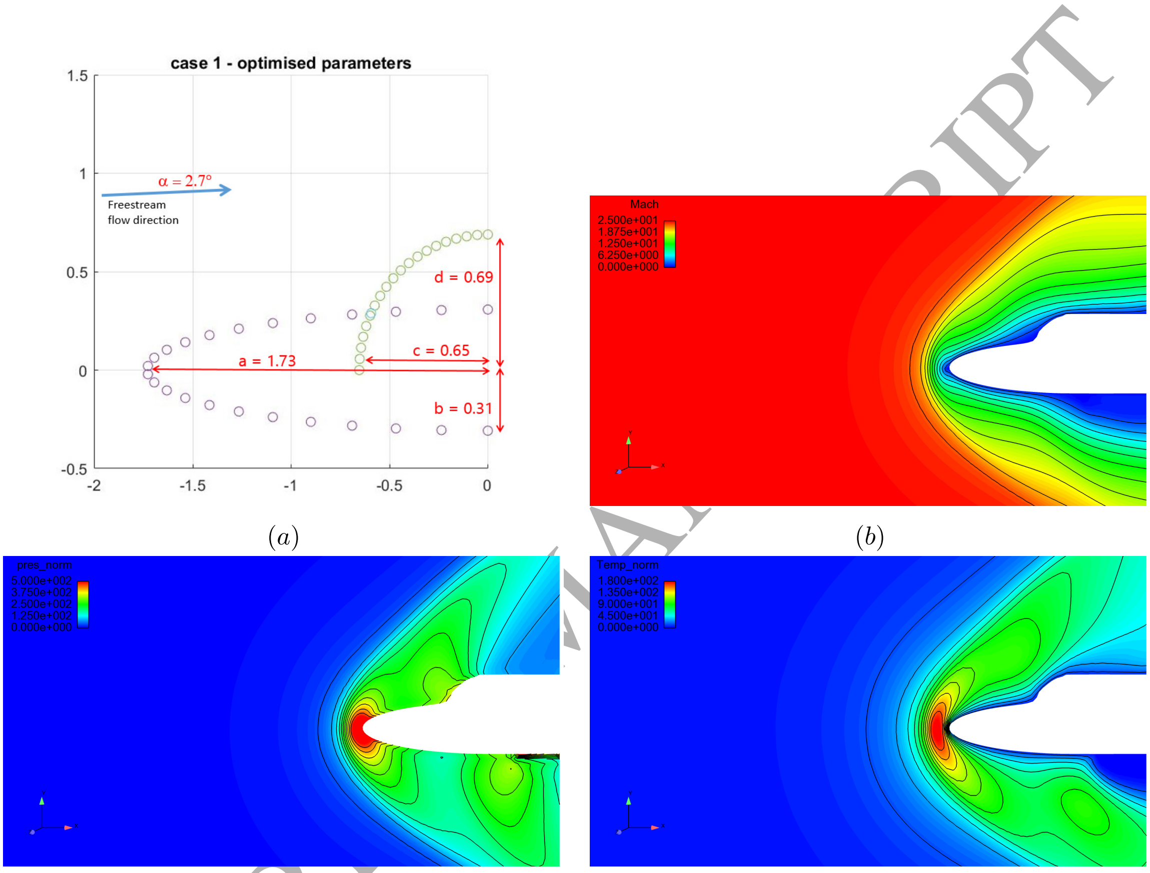


Figure 14: Geometries and objective function values at the end of generation 5 for drag minimisation. These are raw outputs from the parameterisation, some spurious points appear which are rejected during mesh generation.

body shape to be as streamlined as possible and orientated at low angle of attack to the freestream flow. After only 5 generations that is exactly what the optimiser concluded. Note that since the stand-off distance of the main shock system generated is close to the nose of the body, the peak temperature recorded on the surface (locally) is relatively high as indicated in summary Table 6. It is also evident that there appears to be some non-physical behaviour on the underside of the vehicle close to the outlet. It is believed that this is related to reflection effects from the outlet and their interaction with the extruded underside wall. Note that this region of the geometry is not included in the computation of forces for the objective function. The results from this initial test case gave confidence that the optimiser was predicting sensible design solutions.



(*a*) (*b*)

(*c*) (*d*)

Figure 15: Optimised case 1 (drag minimisation) showing: (a) optimised parameters; (b) Mach number distribution; (c) normalised pressure, p/p_∞ distribution; (d) normalised temperature, T/T_∞ distribution

6.2. Case 2 - Drag Maximisation

6.2.1. Optimiser Behaviour

The initial sampling of the design space for the drag maximisation example can be seen in Figure 16. In order to make the optimiser behave as a maximisation tool we have multiplied the drag coefficient by negative one. The highest drag found by the initial sampling is 1.8473. In this example no improvement upon this is found by the optimiser, but as shown in Figure 17 the mean drag of the population does increase. The final population of designs obtained using the optimiser is shown in Figure 18.

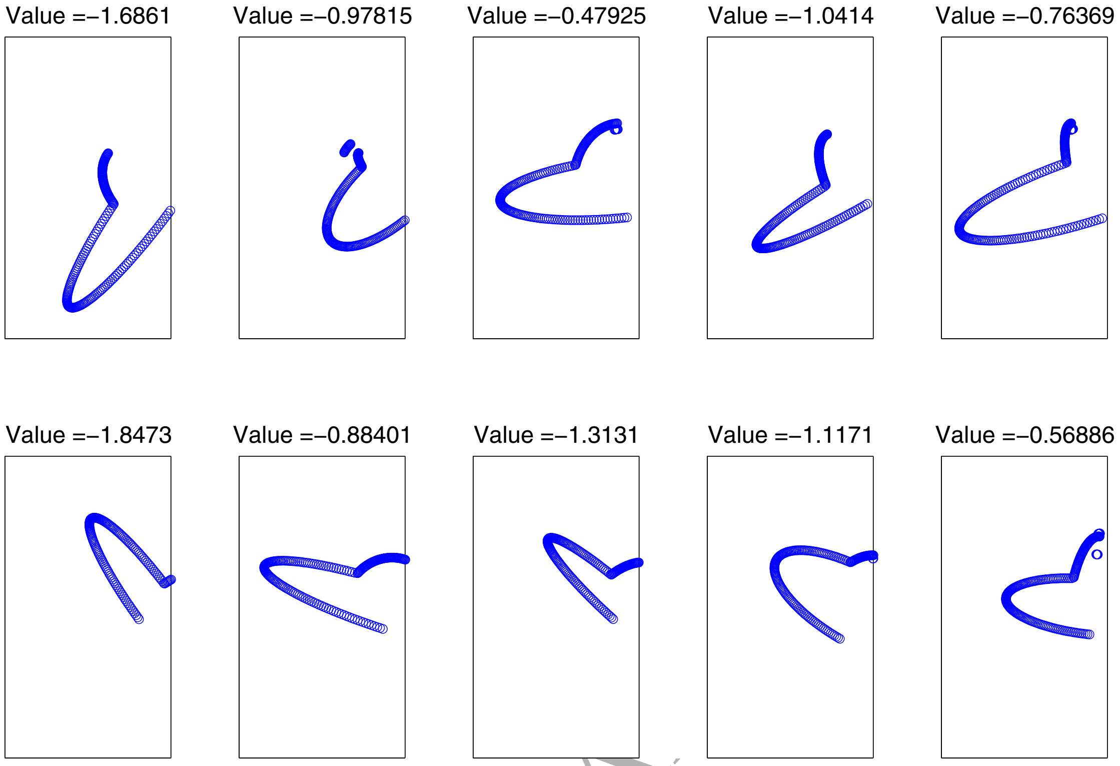


Figure 16: Geometries and objective function values at the end of generation 1 for drag maximisation. These are raw outputs from the parameterisation, some spurious points appear which are rejected during mesh generation.

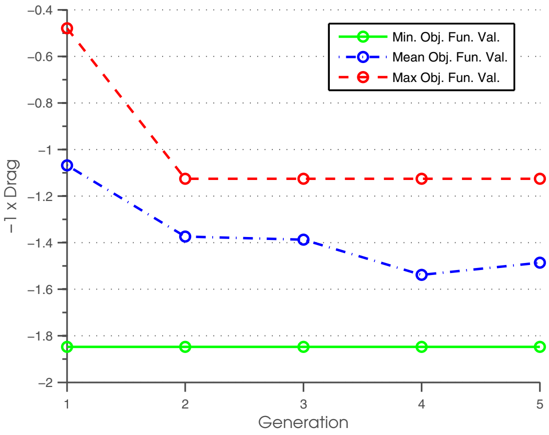


Figure 17: Objective function value history for the drag maxisation case.

6.2.2. Optimised Geometry

Figure 19 shows the geometric configuration and flow orientation determined by the optimiser to maximise drag on the vehicle. Unsurprisingly, the body shape has increased in height and the flow orientation has moved close the largest allowed angle of attack (as defined by the design space limits). This results in a large, strong shock system on the underside of the vehicle with a significant high pressure region acting on a large surface area

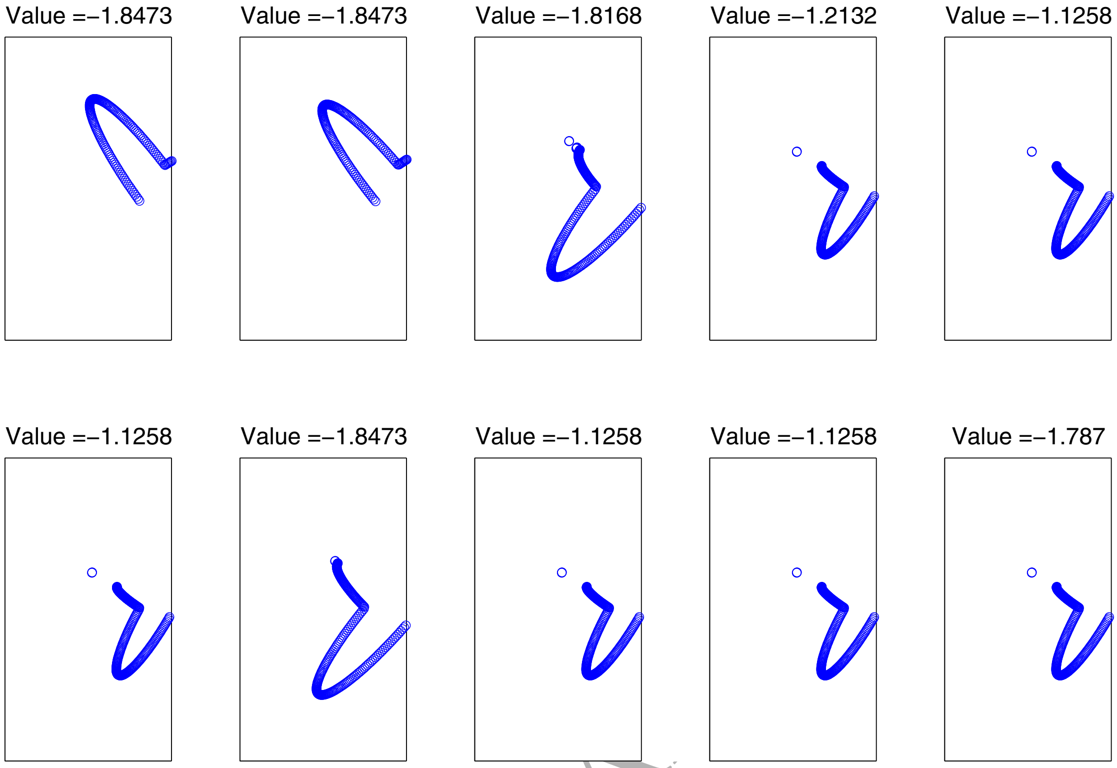


Figure 18: Geometries and objective function values at the end of generation 5 for drag maximisation. These are raw outputs from the parameterisation, some spurious points appear which are rejected during mesh generation.

resulting in a large net force. However, since the stand-off distance of the shock is slightly larger than in case 1 this actually results in a slightly lower maximum surface temperature as indicated in Table 6 even though the extent of the high temperature region across the surface is larger than in case 1. Of course, if the full mission requirements for a vehicle such as this were taken into account this is an unsuitable design since it would result in high levels of drag during the ascent stages of the space flight.

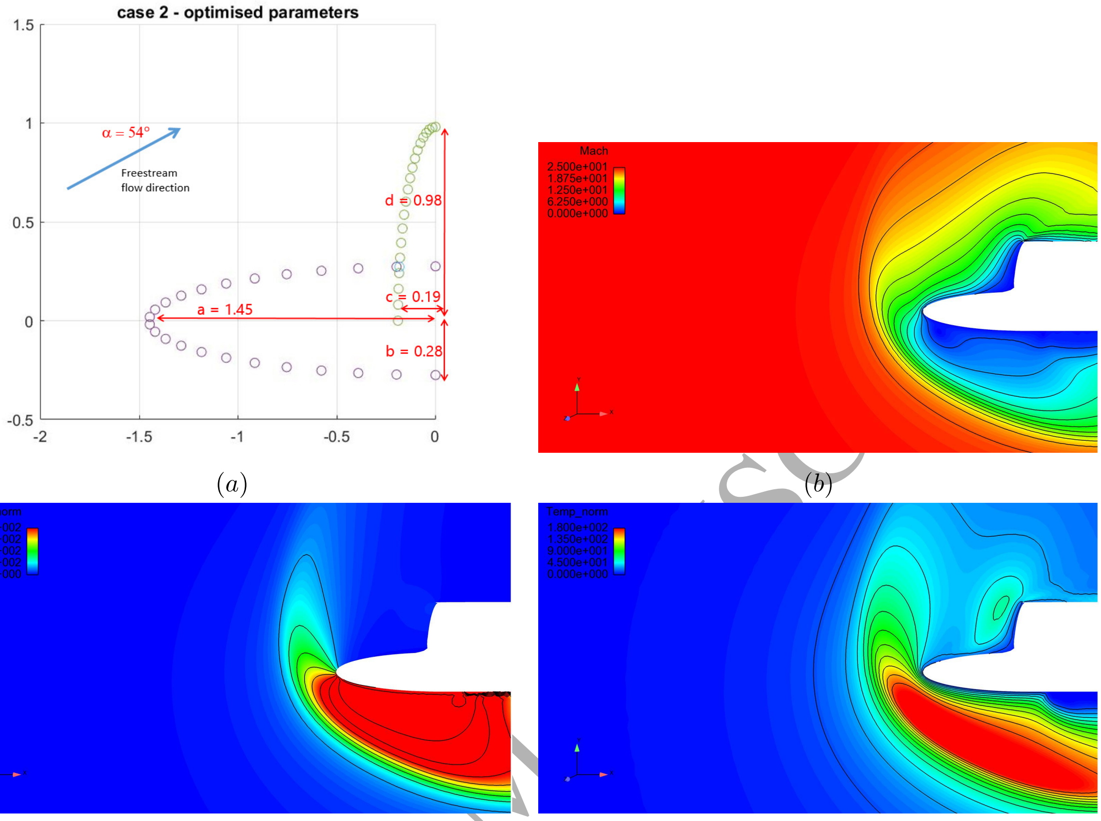
6.3. Case 3 - Temperature Minimisation

6.3.1. Optimiser Behaviour

In Case 3 the objective of the optimiser was to minimise the maximum surface temperature on the profile. The objective function values of the initial sampling of the design space are shown in Figure 20, the lowest temperature in this random sampling was 558.437. This design was improved in the first generation as shown in Figure 21. The lowest temperature obtained in the final population of designs, all shown in Figure 22, was 550.476.

6.3.2. Optimised Geometry

Figure 23 shows the optimised conditions for surface temperature minimisation without any consideration of the net forces acting on the vehicle. It is interesting to note that the resulting geometry is very similar to that of case 1 but the freestream flow is at a slightly higher angle of attack. The implication of this is that the



(a)

(b)

(c)

(d)

Figure 19: Optimised case 2 (drag maximisation) showing: (a) optimised parameters; (b) Mach number distribution; (c) normalised pressure, p/p_∞ distribution; (d) normalised temperature, T/T_∞ distribution

maximum surface temperature experienced by a reentry vehicle such as this in the early stages of reentry must be highly sensitive to the vehicle orientation relative to the freestream flow.

6.4. Case 4 - Multi-Objective Optimisation

6.4.1. Optimiser Behaviour

The final example is a multi-objective optimisation problem. We have added a penalty to the maximum temperature as discussed above. The objective function values, along with profiles, of the initial sampling of the design space are shown in Figure 24. In Figure 25 a small improvement of the objective function over a number of generations can be observed. There is an improvement from -0.030449 to -0.033569. The final population of designs is shown in Figure 26.

6.4.2. Optimised Geometry

In the final test case, the most realistic requirement for a space vehicle during reentry was tested i.e. achieving high drag (for rapid deceleration) whilst attempting to minimise surface heating. This was achieved using the

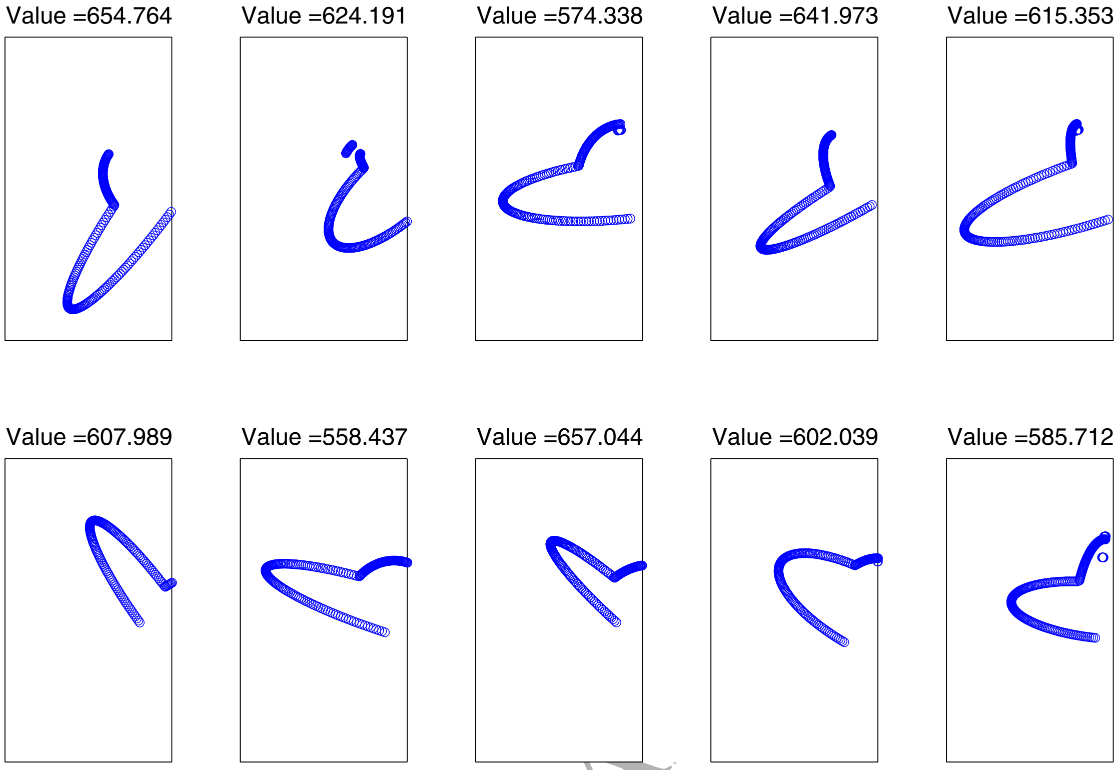


Figure 20: Geometries and objective function values at the end of generation 1 for temperature minimisation. These are raw outputs from the parameterisation, some spurious points appear which are rejected during mesh generation.

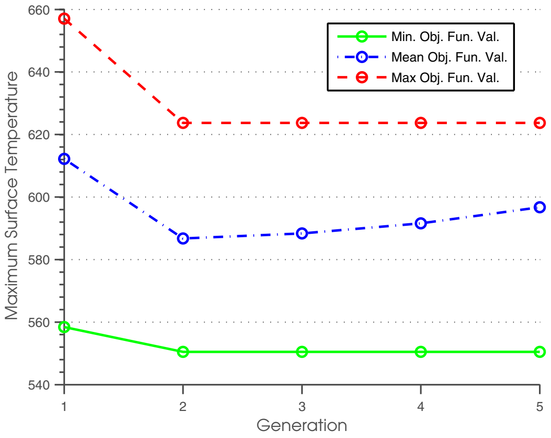


Figure 21: Objective function value history for the temperature minimisation case.

objective function detailed in equation 34 in which T_c was set to 550K. This temperature was chosen as the reference for the temperature penalty since case 3 had shown that achieving this maximum temperature on the body surface under the given flow conditions was, in principle, possible. Temperatures above this therefore added a penalty to the objective function. It is clear from Figure 27 that the resulting geometry in this case is is the longest of each of the cases ($a = 1.84$) and the flow is orientated at a moderate angle of attack ($\alpha = 18°$).

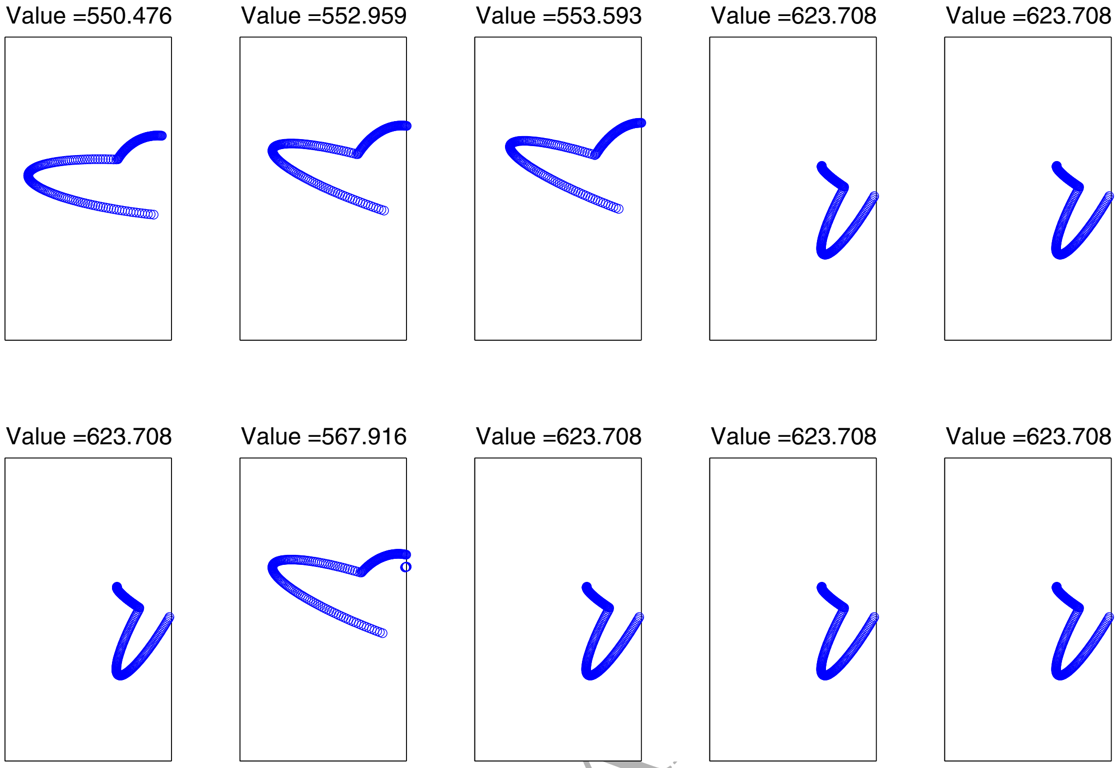


Figure 22: Geometries and objective function values at the end of generation 5 for temperature minimisation. These are raw outputs from the parameterisation, some spurious points appear which are rejected during mesh generation.

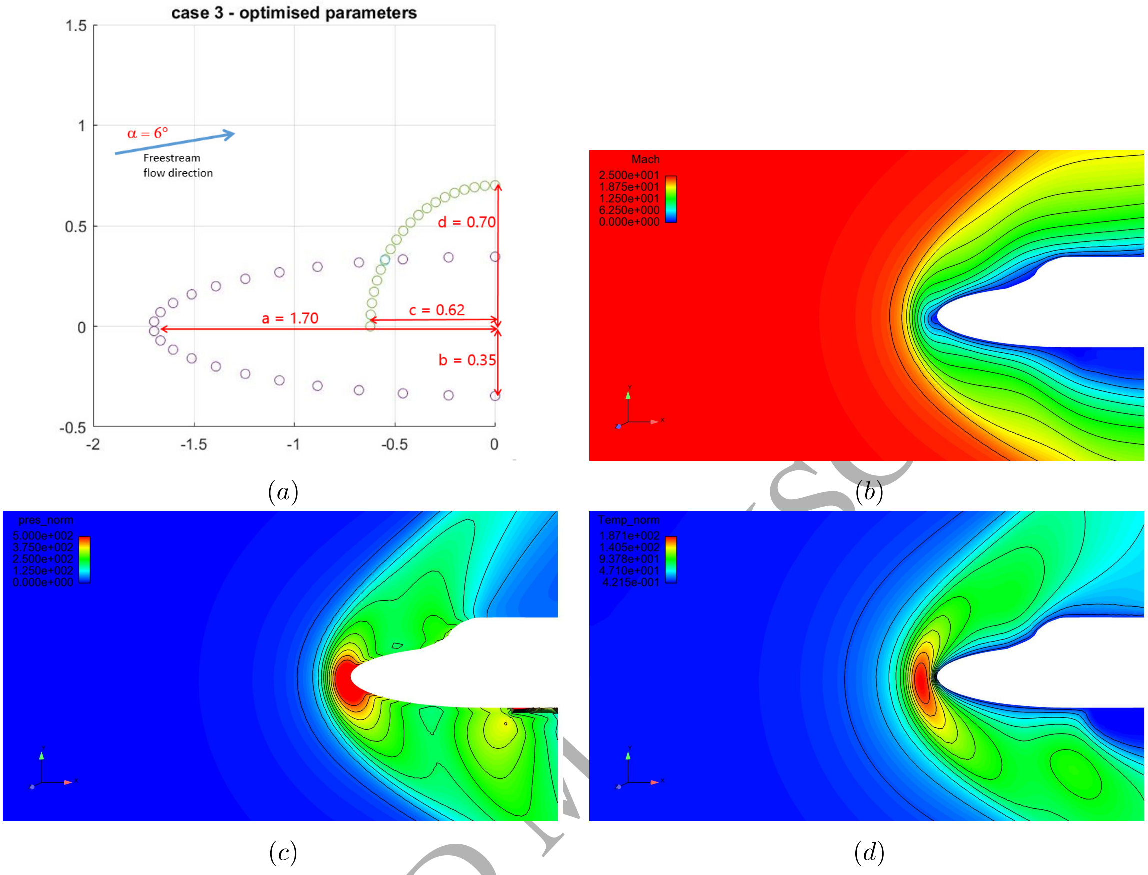
	Case 1	Case 2	Case 3	Case 4
	(drag min)	(drag max)	(temp min)	(drag max with temp constraint)
C_d	0.398	1.847	0.848	1.02
$T_{max}(K)$	4300	3443	550	1093

Table 6: Summary of C_d and max surface temperature for the optimised configuration in each of the four optimisation cases considered

This configuration seems to be a good compromise between the requirement for achieving a high deceleration from space flight into conventional atmospheric flow whilst minimising surface heating. Conveniently, although not considered in the analysis presented here, this geometry would also likely be appropriate for the low drag requirement at low angle of attack during the ascent phase. Perhaps it is no coincidence therefore that this flow configuration and geometry is most similar (out of the four cases) to the actual space shuttle geometry and orientation during reentry.

7. Conclusions

This paper has outlined the potential for a completely novel approach to the solution of highly rarefied, high Mach number flows to be used as a design tool for future space vehicles when coupled to a suitable design optimisation strategy . The Boltzmann–BGK solver has been demonstrated to be a powerful method for solving



(*a*)

(*b*)

(*c*)

(*d*)

Figure 23: Optimised case 3 (temperature minimisation) showing: (a) optimised parameters; (b) Mach number distribution; (c) normalised pressure, p/p_∞ distribution; (d) normalised temperature, T/T_∞ distribution

a wide variety of flow problems across the Knudsen regime and, critically, solve problems outside of the scope of traditional continuum solvers. The ability of this approach to provide insight into the considerations necessary in the design of the forebody of a space vehicle for atmospheric reentry has been demonstrated. The work has shown how the vehicle geometry and flight condition might be modified in order to either minimise or maximise drag, to minimise body surface heating or to maximise drag with surface temperature penalties.

Clearly, in its current form, the approach outlined is restrictively expensive computationally and therefore limited to 2D in this paper. However, the work outlined does provide a platform for the development of more efficient techniques based on the principles outlined. Future work will be undertaken to explore solver acceleration by means of alternative velocity space discretisation schemes and the use of reduced order modelling approaches for solution acceleration in the context of the evolutionary optimisation utilised. Improving the outflow boundary condition to remove unwanted reflection effects will be considered based on the 'non–reflecting boundary approaches' often used in lattice Boltzmann schemes. Viable methods for extension of the solver to three dimensions will also be explored including more efficient parallelisation by means of both physical and

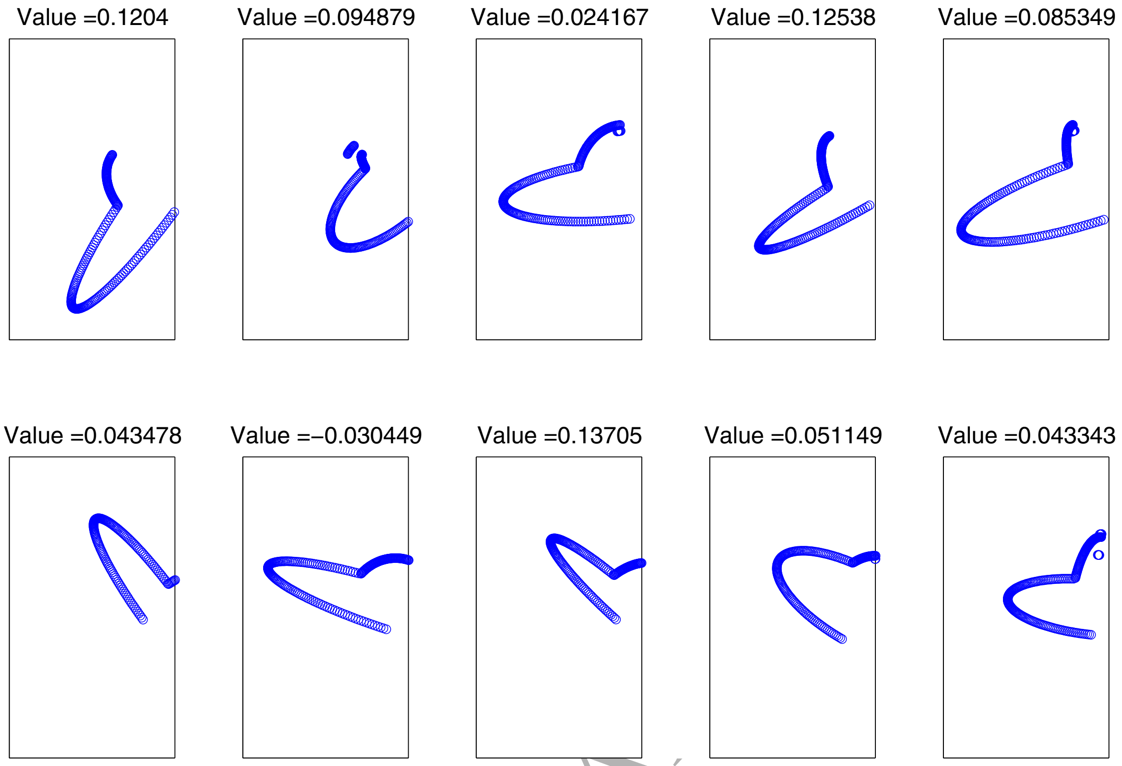


Figure 24: Geometries and objective function values at the end of generation 1 for the multi–objective optimisation. These are raw outputs from the parameterisation, some spurious points appear which are rejected during mesh generation.

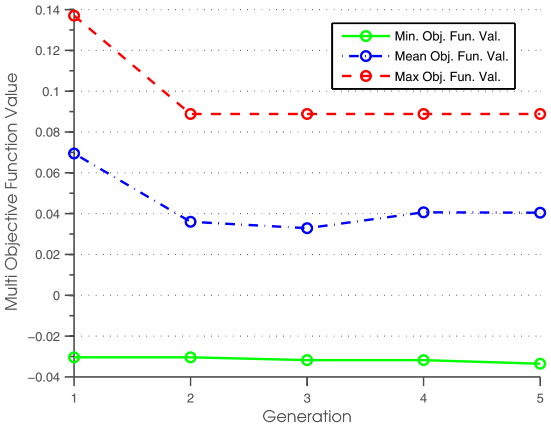


Figure 25: Objective function value history for the multi–objective optimisation

velocity space partitioning. As the accessibility of High Performance Computing continues to grow, there is the potential that techniques such as that presented in this paper will become powerful design approaches for 21st Century space vehicles.

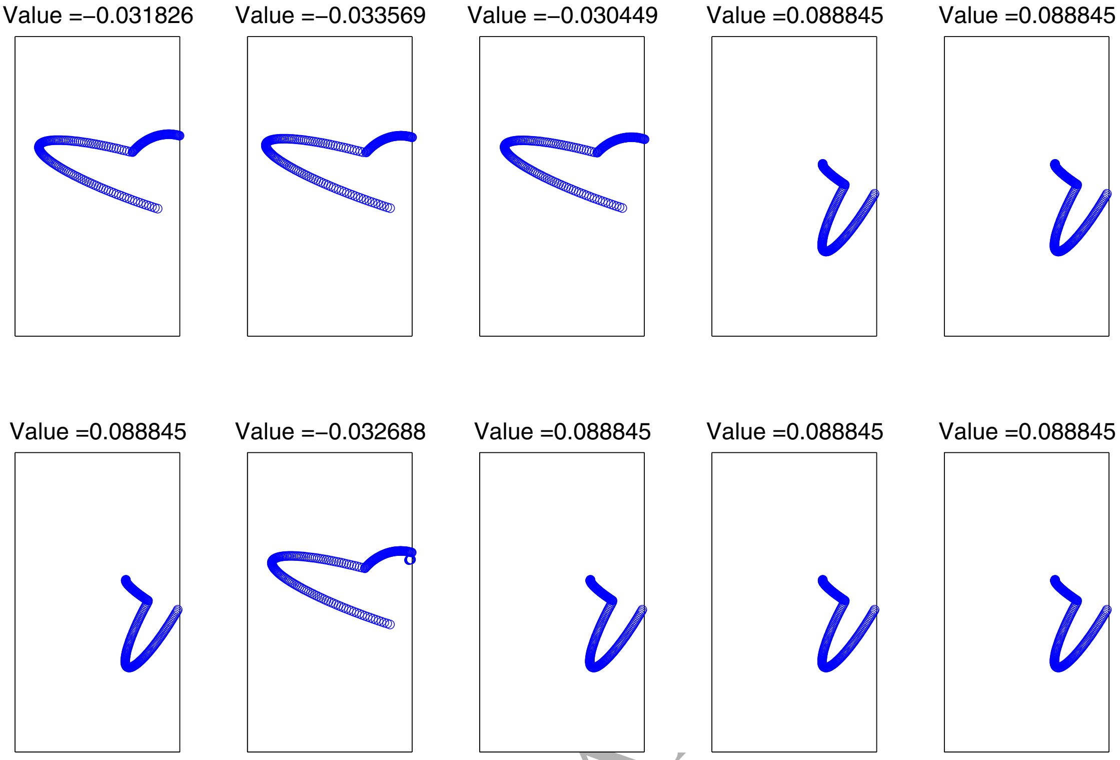


Figure 26: Geometries and objective function values at the end of generation 5 for multi–objective optimisation. These are raw outputs from the parameterisation, some spurious points appear which are rejected during mesh generation.

Acknowledgements

The authors would like to acknowledge the support provided by the Zienkiewicz Centre for Computational Engineering at Swansea University for the provision of the computational resources which has allowed this work to take place.

References

[1] http://www.virgingalactic.com/ accessed on 28th February 2017

[2] http://www.reactionengines.co.uk/ accessed on 28th February 2017

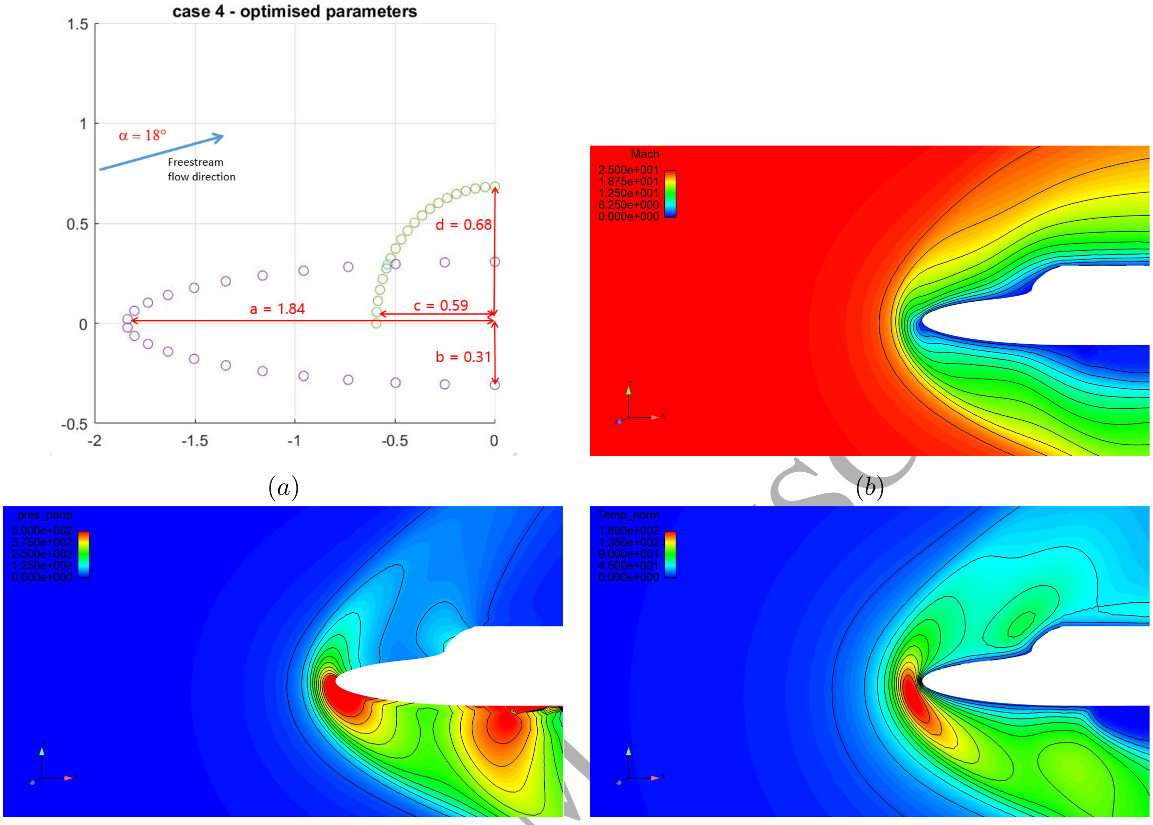
[3] http://www.spacex.com/ accessed on 28th February 2017

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

[5] J.N. Moss, G.A. Bird, Direct Simulation of Transitional Flow for Hypersonic Re-entry Conditions, *AIAA Paper no. 84-0223*, 1984

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

[7] B Evans, Finite element solution of the Boltzmann equation for macroscopic gas flows, PhD Thesis, Swansea University, 2008



(*a*) (*b*)

(*c*) (*d*)

Figure 27: Optimised case 4 (multi–objective optimisation) showing: (a) optimised parameters; (b) Mach number distribution; (c) normalised pressure, p/p_∞ distribution; (d) normalised temperature, T/T_∞ distribution

[8] C. Cercignani, Theory and Application of the Boltzmann Equation, *Springer*, 1988.

[9] M.M. Kuzetsov, Analystical Solution of the Boltzmann Equation in a Knudsen Layer, *J.* App. Mech. & Tech. Phys., **4**(2), 604–607, 1971

[10] S.A. Trugman, A.J. Taylor, Analytic solution of the Boltzmann equation with applications to electron transport in inhomogeneous semiconductors, *P*hys. Rev. B , **33**(8), 5575–5584, 1986

[11] W.G. Vincenti, Introduction to Physical Gas Dynamics, *John Wiley and Sons, Inc.*, New York, 1965

[12] B. Evans, K. Morgan, O. Hassan, A discontinuous finite element solution of the Boltzmann kinetic equation in collisionless and BGK forms for macroscopic gas flows, *Applied Mathematical Modelling*, **35**, 996–1015, 2011

[13] B. Cockburn, Discontinuous Galerkin Methods, *Journal of Applied Mathematics and Mechanics*, **83** (11), 731–754, 2002

[14] J. Donea, A Taylor–Galerkin Method for Convective Transport, *Int. J. Numer. Meth. Fluids*, **20**, 101–120, 1984

[15] N. Bellemo, P. LeTallec, B. Perthame, Nonlinear Boltzmann equation solutions and applications to fluid

dynamics, *App. Mech. Rev.*, **48** (12), 1995

[16] G. Busoni, A. Palczewski, A stationary Boltzmann equation near equilibrium, *Math Models Methods Appl. Sci.*, **3**, 395–440, 1993

[17] Chen, S., Doolen, G. D., Lattice Boltzmann Method for Fluid Flows, *Annual Review of Fluid Mechanics*, **30**, 329-364, 1998

[18] G. Karypis, V. Kumar, METIS4.0: Unstructured graph partitioning and sparse and sparse matrix ordering system. Technical report, Department of Computer Science, University of Minnesota (1998), http://www.cs.unn.edu/ metis

[19] Arias O., and Falcinelli O., and Fico Jr. N., and Elaskar S., *Finite Volume Simulation of Flow Over a NACA 0012 Using Jameson, MacCormack, Shu, and TVD Esquemes*, Mechanica Computacional, Vol. XXVI, 03097-3116 (2007)

[20] Hafezi, F., Ransing, R., Lewis, R., The calculation of drag on nano-cylinders, *International Journal for Numerical Methods in Engineering*, 2016

[21] Lagree PY., Small Re flows, 2013. [Online]. Paris: Institut Jean Le Rond Alembert, Available: http://www.lmm.jussieu.fr/ lagree/COURS/M2MHP/petitRe.pdf [Accessed 30th October 2015]

[22] G. Renner and A. Ekárt, Genetic algorithms in computer aided design, *Computer Aided Design*, **35**, 709–726, 2003.

[23] A. Jameson, Efficient Aerodynamic Shape Optimization, in proceedings of the *10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, 2004.

[24] M. Harbeck, A. Jameson, Exploring the Limits of Shock-free Transonic Airfoil Design, in proceedings of the *43rd Aerospace Sciences Meeting and Exhibition*, 2005.

[25] J.C. Vassberg, A. Jameson, Aerodynamic Shape Optimization of a Reno Race Plane, *International Journal of Vehicle Design*, **28**, 4, 318–338, 2002.

[26] S. Walton, O. Hassan, K. Morgan, and M. R. Brown, Modified cuckoo search: a new gradient free optimisation algorithm, *Chaos, Solitons & Fractals*, 44(9):710–718, 2011.

[27] M. Mitchell, An Introduction to Genetic Algorithms, sixth ed., MIT Press, 1999.

[28] D. Bratton, J. Kennedy, Defining a standard for particle swarm optimization, *Proceedings of the 2007 IEEE Swarm Intelligence Symposium*, 2007.

[29] R. Storn, K. Price, Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces, *Journal of Global Optimisation*, 11, 341–359, 1997.

[30] X.-S. Yang, S. Deb, Engineering optimisation by cuckoo search, *International Journal of Mathematical Modelling and Numerical Optimisation*, 1, 330–343, 2010

[31] D. Quagliarella, A. Vicini, Viscous single and multicomponent airfoil design with genetic algorithms, *Finite Elements in Analysis and Design*, 37, 365–380, 2001.

[32] A.L. Marsden, M. Wang, J.E. Dennis Jr., P. Moin, Suppression of vortex-shedding noise via derivative-free shape optimization, *Physics of Fluids*, **16**, 10, 2004.

[33] K.C. Giannakoglou, D.I. Papadimitriou, I.C. Kampolis, Aerodynamic shape design using evolutionary algorithms and new gradient-assisted metamodels, *Computer methods in applied mechanics and engineering*, **195**, 6312–6329, 2006.

[34] P.I.K. Liakopoulos, I.C. Kampolis, K.C. Giannakoglou, Grid enabled, hierarchical distributed metamodel-assisted evolutionary algorithms for aerodynamic shape optimization, *Future Generation Computer Systems*,

24, 701–708, 2008.

[35] J. Periaux, D.S. Lee, L.F. Gonzalez, K. Srinivas, Fast reconstruction of aerodynamic shapes using evolutionary algorithms and virtual nash strategies in a CFD design environment, *Journal of Computational and Applied Mathematics*, **232**, 61–71, 2009.

[36] X.-S. Yang and S. Deb, Cuckoo search via Lévy flights, In *Proceedings of World Congress on Nature & Biologically Inspired Computing (NaBIC 2009, India)*, pages 210–214. IEEE Publications, 2009.

[37] G. M. Viswanathan, Lévy flights and superdiffusion in the context of biological encounters and random searches, *Physics of Life Reviews*, 5:133–150, 2008.

[38] S. Walton, O. Hassan, and K. Morgan, Selected engineering applications of gradient free optimisation using cuckoo search and proper orthogonal decomposition, *Archives of Computational Methods in Engineering*, 20(2):123–154, 2013.

[39] D.S. Naumann, B. Evans, S. Walton, and O. Hassan, A novel implementation of computational aerodynamic shape optimisation using modified cuckoo search, *Applied Mathematical Modelling*, 2015.

[40] S. Walton, M. R. Brown, O. Hassan, and K. Morgan, Comment on ”Cuckoo search: A new nature-inspired optimization method for phase equilibrium calculations” by V. Bhargava, S. Fateen, A. Bonilla-Petriciolet [Fluid Phase Equilibria 337 (0) (2013) 191 - 200], *Fluid Phase Equilibria*, 2013.

[41] S. Walton, Open source project; http://code.google.com/p/modified-cs/, 2011.

[42] S. Walton, O. Hassan, and K. Morgan, Reduced order mesh optimisation using proper orthogonal decomposition and a modified cuckoo search, *International Journal for Numerical Methods in Engineering*, 93(5):527–550, 2013.

[43] J.D. Anderson Jr., Hypersonic and high temperature gas dynamics, 2nd edition, *AIAA*, 2006

[44] J.C. Maxwell, On the dynamic theory of gases, *Phil. Trans. Soc.*, **157** (49), 1867