HiSST: 2nd International Conference on High-Speed Vehicle Science & Technology

11–15 September 2022, Bruges, Belgium

CEAS
Council of European Aerospace Societies

Development of an adaptive strand-Cartesian solver in 2D for non-equilibrium aerothermodynamics simulations

Chay W. C. Atkins[1] *and Ralf Deiterding*[2]

Keywords: *hypersonic, strand, overset*

Abstract

The extreme conditions experienced in hypersonic flight can be difficult to reproduce in ground test facilities. As such, the use of computational simulations is vital in the design of Thermal Protection Systems (TPSs) for hypersonic vehicles. Many TPSs use materials that ablate and this leads to changes in shape of the body with time. As such, the computational mesh needs to be updated, whilst maintaining a high resolution in the shock and boundary layer regions, which can require significant user input. In this work, a prototype strand/Cartesian Adaptive Mesh Refinement (CAMR) solver has been created using the AMROC (Adaptive Mesh Refinement in Object-oriented C++) framework, that enables meshes to be generated with minimal user input. The strand/CAMR technique combines a "strand" mesh, grown from a discretised surface, in the near-body region with an adaptive Cartesian mesh in the off-body region in order to highly resolve off-body shocks and boundary layers. The development of the off-body and near-body two-temperature Navier-Stokes solvers, and the overset algorithms used to join the two regions, is described. A series of test cases that aim to verify and validate the hypersonic 2D/axisymmetric strand/CAMR solver are presented. An order-of-accruacy test is carried out on an overset domain to verify the implementation of the new spatial- and time-integration methods. A high-enthalpy experiment is simulated in order to validate the new solver and investigate the influence of the overset mesh on heat flux predictions. Finally, the automated surface deformation enabled by the new solver is demonstrated through the simulation of a recessing nose-tip. The results indicate that the strand/CAMR technique can be used to accurately simulate vehicles in hypersonic flows and offers a high level of automation.

1. Introduction

Travelling at hypersonic speeds through a planet's atmosphere leads to extremely high post shock temperatures. Without adequate thermal protection, the vehicle and its payload are unlikely to survive the harsh conditions of hypersonic flight. In order to design a Thermal Protection System (TPS) for a given mission, an accurate assessment of the surface heat loads and shear stresses is required. Ground test facilities can be used in the design process, however, they are often unable to fully match the predicted flight conditions [1]. As such, computational modelling plays an important role in the design of TPSs for hypersonic vehicles.

A number of hypersonic Computational Fluid Dynamics (CFD) solvers can be found in the literature, and have been shown to accurately predict the heat fluxes obtained in high-enthalpy flow experiments [2, 3, 4, 5, 6, 7, 8, 9]. However, it has been demonstrated that the predictions are sensitive to the computational mesh, with the quality of the mesh in the shock region and on the surface both influencing the results [10, 11, 12]. This problem is compounded by the fact that many TPSs change shape during flight due to ablation, and the freestream Mach number changes as a vehicle progresses through its flight. Thus, a mesh that adequately resolves the shock features and surface boundary layer for the

[1] *University of Southampton, c.w.atkins@soton.ac.uk*

[2] *University of Southampton, r.deiterding@soton.ac.uk*

initial vehicle shape and flight conditions could give poor results at later points in the flight. This has led to the development of automated mesh morphing techniques when simulating hypersonic vehicles, used to move the grid with the surface and align the grid to the bow shock [7, 13, 14]. However, as hypersonic vehicles become more complex, with control surfaces and flexible TPS designs being incorporated, new automated meshing techniques may be needed.

In recent years strand/Cartesian Adaptive Mesh Refinement (CAMR) solvers have been developed [15] that are able to generate computational meshes around complex bodies [16] with minimal user input. The method uses a near-body solver and an off-body solver, which are joined using overset meshing techniques (see Fig. 1).

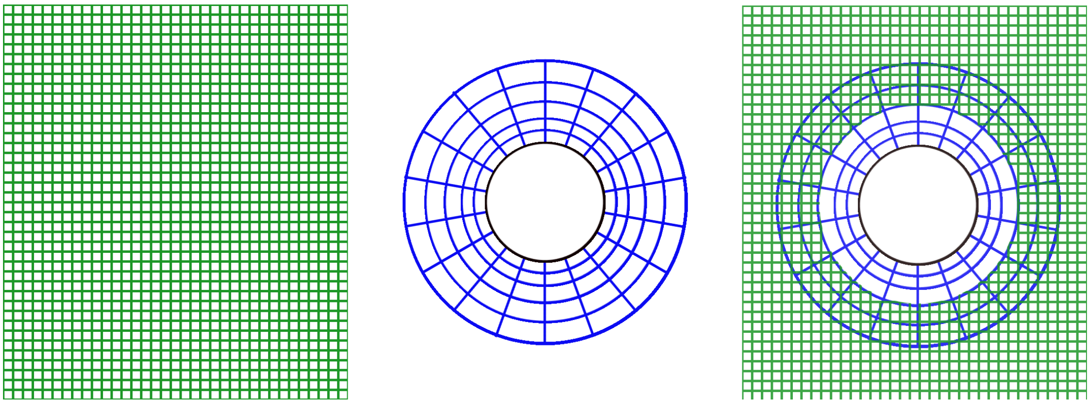
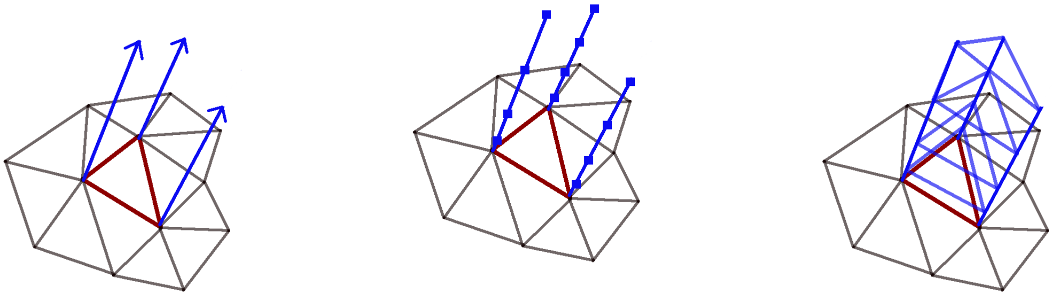


Fig 1. An illustration of a near-body mesh overset onto a Cartesian off-body mesh.

The near-body mesh is automatically generated by growing "strands" from a discretised surface, which are then joined to form cells, as shown in Fig. 2. Various techniques have been developed to ensure a high quality mesh in the near-body region, including the use of multiple strands from a given point [17], multiple layers of strands and strand generation using iso-surfaces [18]. The off-body region utilises a CAMR solver, which is able to efficiently resolve flow features and match the near-body mesh cell sizes in the overset boundary region using cell refinement.



(a) Growth vectors are created for each strand.

(b) Nodes are placed along each of the strand growth vectors, with the spacing determined by a stretching function.

(c) The nodes are joined to create layers of cells.

Fig 2. An illustration of the strand mesh generation procedure on a three-dimensional surface.

There are several advantages to a strand/CAMR solver. It has been demonstrated that the strand technique can generate high-quality meshes around complex, three-dimensional bodies [16]. The strand technique enables the use of semi-structured computational meshes, with structured data in the wall normal direction and either structured or unstructured data in each layer of the mesh. This results in more computationally efficient data structures as the solution vectors from neighbouring elements can be grouped in memory. It also allows efficient solution techniques to be more easily implemented, such as line relaxation time-integration techniques [19] and high-order-accurate spatial discretisation [20, 21]. The off-body CAMR solver incorporates the advantages inherent in an AMR solver. Firstly, computationally efficient data structures can be used. Secondly, the adaptivity enables flow features to be captured in a computationally efficient manner, where cells are clustered in areas of high gradients in the solution. As such, CAMR solvers are able to highly resolve shock structures, potentially alleviating the surface heat flux inaccuracies caused by poor grid alignment with the shock [12]. Finally, the Cartesian geometry means high-order reconstruction can be more easily implemented than on an unstructured mesh.

In this paper, the development of a prototype 2D/axisymmetric hypersonic strand/CAMR solver is presented. The governing equations and thermodynamic nonequilibrium models are presented in Section 2, the numerical implementation, strand mesh generation algorithm and the overset connectivity algorithms are detailed in Section 3. The results from the verification and validation studies are presented in Section 4.

2. Governing Equations and Models

The nonequilibrium thermochemistry encountered in hypersonic flows has been modelled using a multi-species, two-temperature model [22]. In this model, the thermodynamic nonequilibirum is accounted for by using one temperature to represent the internal energy in the translational and rotational modes, T_{tr}, and a second to represent the energy in the vibrational and electronic modes, and the free electron energy, T_{ve}.

The governing equations for the model are given by

$$\frac{\partial \mathbf{Q}}{\partial t} + \frac{\partial \mathbf{F}}{\partial x} - \frac{\partial \mathbf{F}^v}{\partial x} + \frac{\partial \mathbf{G}}{\partial y} - \frac{\partial \mathbf{G}^v}{\partial y} = \mathbf{W} \,. \tag{1}$$

The vector of conserved variables **Q**, the inviscid flux vectors **F** and **G**, and the source vector **W** take the form

$$\mathbf{Q} = \begin{bmatrix} \rho_1 \\ \vdots \\ \rho_N \\ \rho u \\ \rho v \\ \rho e^{ve} \\ \rho E \end{bmatrix}, \quad \mathbf{F} = \begin{bmatrix} \rho_1 u \\ \vdots \\ \rho_N u \\ \rho u^2 + p \\ \rho uv \\ \rho e^{ve} u \\ u(\rho E + p) \end{bmatrix}, \quad \mathbf{G} = \begin{bmatrix} \rho_1 v \\ \vdots \\ \rho_N v \\ \rho uv \\ \rho v^2 + p \\ \rho e^{ve} v \\ v(\rho E + p) \end{bmatrix}, \quad \mathbf{W} = \begin{bmatrix} \dot{w}_1 \\ \vdots \\ \dot{w}_N \\ 0 \\ 0 \\ Q_{ve} \\ 0 \end{bmatrix}. \tag{2}$$

The viscous flux vector, $\mathbf{F}^v$, is given by

$$\mathbf{F}^v = \begin{bmatrix} -J_{x,1} \\ \vdots \\ -J_{x,N_s} \\ \tau_{x,x} \\ \tau_{y,x} \\ \kappa_{ve} \frac{\partial T_{ve}}{\partial x} - \sum_{s=1}^{N_s} J_{x,s} e_{ve} \\ \kappa_{tr} \frac{\partial T_{tr}}{\partial x} + \kappa_{ve} \frac{\partial T_{ve}}{\partial x} + u\tau_{x,x} + v\tau_{y,x} - \sum_{s=1}^{N_s} J_{x,s} h_s \end{bmatrix}, \tag{3}$$

where $\tau_{i,j}$ is the viscous stress tensor, κ_{tr} and κ_{ve} are the mixture translational-rotational and vibrational-electronic thermal conductivities, respectively. The species diffusive flux, $J_{x,s}$ is based on a modified version of Fick's diffusion law, which ensures the sum of the diffusion fluxes is zero [23]. The vector $\mathbf{G}^v$ takes a similar form, with the x derivatives replaced by y derivatives.

The thermodynamic properties, transport properties and source terms are all calculated using Mutation++ [24]. Mutation++ is a library that contains state-of-the-art thermochemical models in a computationally efficient, object-oriented C++ package. The library has been designed to work with high enthalpy flows and is able to calculate the thermodynamic properties and source terms using Park's two-temperature model.

The mixture density, ρ, is calculated as the sum of the partial densities, ρ_s, and the pressure, p, is determined using the ideal gas equation and Dalton's Law of partial pressures. Within Mutation++, the internal energies are modelled using a two-temperature Rigid-Rotator Harmonic-Oscillator model. The mixture viscosity and translational thermal conductivity can be calculated using Wilke's mixing rule [25], the Gupta-Yos mixing rule [26] or by solving the linear equations derived from the Boltzmann equations using the Chapmann-Enskog procedure [27, 28]. The Chapmann-Enskog approach is the most accurate [29] and thus is the default method used in this work. The internal energy conductivities are calculated using the Eucken approximation and the collision cross-sections are taken from the literature [24]. The average diffusion coefficient for each species is calculated as the reciprocal molar average of the binary diffusion coefficient relative to all other species.

The species production rate is calculated using a modified Arrhenius equation, where the forward reaction rate is a function of the rate controlling temperature T_c, and is given by

$$k_{f,r}(T_c) = A_{f,r} T_c^{\eta_{f,r}} \exp\left[-\theta_r / T_c\right] . \tag{4}$$

A_r is the reaction rate constant, θ_r is the activation temperature and $\eta_{f,r}$ is a constant. The rate controlling temperature, T_c, is the geometric average of the translational-rotational and vibrational-electronic temperatures,

$$T_c = \sqrt{T_{tr} \cdot T_{ve}} . \tag{5}$$

The forward reaction rate constants used in this work are those of Park [22]. The backwards reaction rate is calculated using its relationship to the forward reaction rate and the equilibrium constant, where the equilibrium constant is calculated using Gibb's free energy minimisation.

For a mixture of neutral species, the energy transfer between the translational-rotational and vibrational-electronic modes is modelled within Mutation++ using

$$Q_{ve} = \sum_{s \neq e} \left(Q_s^{T-V} + Q_s^{C-V} + Q_s^{C-el} \right) . \tag{6}$$

The translational-vibrational energy exchange, Q_s^{T-V}, is modelled using a Landau-Teller type formula,

$$Q_s^{T-V} = \rho_s \frac{\mathrm{d}e_s^v}{\mathrm{d}t} = \rho_s \frac{e_s^v(T_{tr}) - e_s^v}{\tau_{v,s}^{T-V}} , \tag{7}$$

where the species relaxation times are found using the modified Millikan and White relationship [30], with the Park correction factor [31]. The source terms accounting for change in the vibrational and electronic energy due to chemical reactions, Q_s^{C-V}, Q_s^{C-el}, are given by

$$Q_s^{C-V} = c_1 \dot{w}_s e_s^v \quad , \quad Q_s^{C-el} = c_1 \dot{w}_s e_s^{el} . \tag{8}$$

In both cases a non-preferential model is used where $c_1 = 1$.

3. Numerical Implementation

The governing equations have been implemented within a finite volume based strand/CAMR solver. A strand/CAMR solver requires a number of components: a CAMR solver, a strand mesh solver, a strand mesh generator, and overset tools capable of determining the domain connectivity and communicating the required information.

3.1. Cartesian AMR Solver

In the off-body region a CAMR solver is used to discretise space and integrate the fluid equations in time. A common implementation of a CAMR solver is the block-Structured Adaptive Mesh Refinement (SAMR) algorithm of Berger and Colella [33]. In the SAMR technique cells are marked for refinement and formed into rectangular blocks which are overlayed with finer cells [34]. The refinement between the levels is determined by the refinement factor. This process occurs recursively on each layer to produce a hierarchy of refined, rectangular grids (see Figure 3).

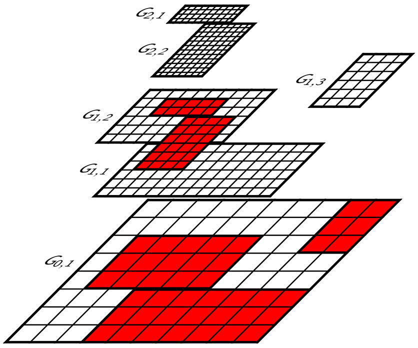


Fig 3. A SAMR refinement hierarchy with structured rectangular refinement blocks.

The off-body solver is based on the adaptive computational framework AMROC (Adaptive Mesh Refinement in Object-oriented C++). AMROC implements the SAMR algorithm of Berger and Colella [33] on parallel computers with distributed memory [35]. Various shock capturing finite volume methods have been integrated with the framework to allow highly adaptive simulations of flow fields that include discontinuities. In this work, a patch integrator that implements the governing equations from Section 2 has been created for modelling hypersonic flows. The hypersonic integrator uses AMROC's Total Variation Diminishing (TVD)-MUSCL extrapolation and an extended version of the Advection Upstream Splitting Method (AUSM) flux scheme [36] to calculate the inviscid fluxes. A central difference scheme is used to calculate the viscous fluxes and the source term is integrated in time using Strang splitting. This results in a second-order-accurate method in space and time.

3.2. Strand Solver

The near-body strand region uses a modified version of the AMROC patch integrator that enables the Cartesian grid to be mapped to a stretched, body-fitted grid. The Cartesian grid uses the computational coordinates (ξ, η) and the solution is based in physical, (x, y) coordinates. The mapping is carried out using the face normal vector in physical space, $\mathbf{n} = [n_x, n_y]^T$, the face length in physical space, $|\mathbf{n}|$ and the transformation metrics $\left(\frac{\partial \xi}{\partial x}\right), \left(\frac{\partial \xi}{\partial y}\right), \left(\frac{\partial \eta}{\partial x}\right), \left(\frac{\partial \eta}{\partial y}\right)$.

The inviscid fluxes are calculated by rotating the solution vectors either side of a face to be ortho-normal to the cell face. The flux is then evaluated as a normal flux and rotated back. The reconstruction of the left and right states at the face, $\mathbf{Q}_l$ and $\mathbf{Q}_r$, is carried out on the computational mesh using either zero-order extrapolation or TVD-MUSCL reconstruction. In the reconstruction, the variables are first transformed to computational space, using

$$\hat{s} \equiv Js \,, \tag{9}$$

where s is a scalar variable in physical space, $\hat{s}$ is the scalar transformed to computational space and J is the Jacobian of the transformation is given by

$$J = \frac{\partial x}{\partial \xi}\frac{\partial y}{\partial \eta} - \frac{\partial y}{\partial \xi}\frac{\partial x}{\partial \eta}\,. \tag{10}$$

The MUSCL extrapolation and limiting is performed first, before the variables are transformed back to physical space to calculate the derived variables. The reconstruction can be carried out using the conserved variables or using the species densities, x and y momentum, pressure and vibrational-electronic temperature.

The viscous fluxes are calculated by evaluating $\mathbf{F}^v$ and $\mathbf{G}^v$ at each face, using the transformation metrics to calculate the gradients. The derivatives of the computational coordinates with respect to the physical coordinates are obtained from the mapping, either from the analytic solution (if available) or using second-order-accurate numerical derivatives. The derivatives of the flow variables on the computational domain are found using central-difference formulas. The resulting fluxes are then multiplied by the components of the face normal vector to calculate the flux in physical space.

Thin near-wall cells are often required on the strand domain for accurate heat flux predictions. These cells can place a severe restriction on the maximum time step when using explicit time-integration methods. As part of this research implicit time-integration methods have been implemented within the strand solver so that larger time steps can be used in steady-state simulations. The unique structure of the strand mesh enables operator splitting to be used, where the fluxes in the wall-normal direction can be integrated separately to the fluxes in the wall-parallel direction. This allows for Implicit/Explicit (IMEX) methods to be used where only the fluxes in the wall-normal direction are integrated implicitly. As a result, the Jacobian only needs to be constructed for the wall-normal fluxes and efficient block tri-diagonal solvers can be used to solve the linear system. In this research both a line-implicit backward Euler method and a dimensionally split backward/forward Euler method have been implemented in the strand solver.

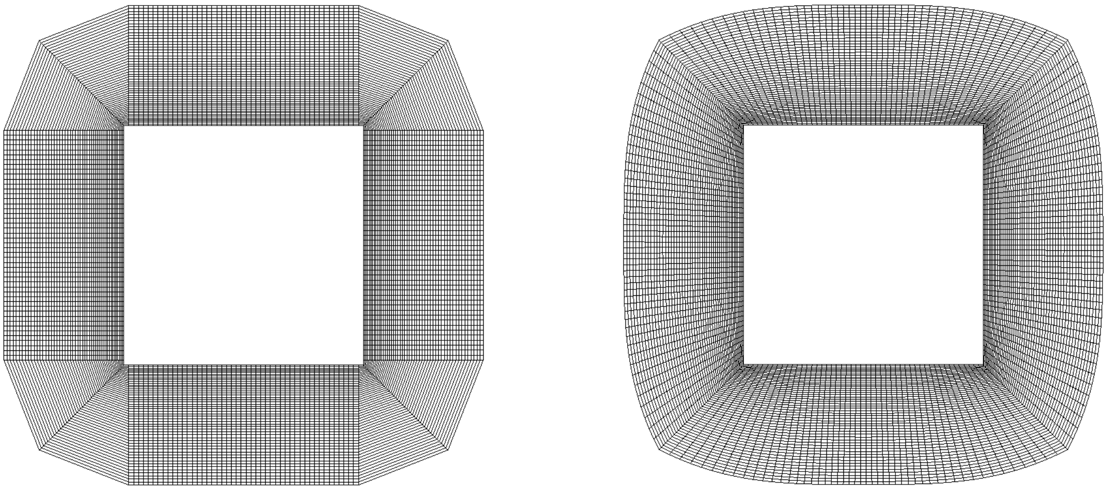
3.3. Strand Mesh Generation

For the mapped near-body mesh, the mapping and the resulting mesh is automatically generated using the strand mesh technique, outlined in Ref. [37]. In this method, a surface is input, strands are grown from the surface and the cells are created by joining the nodes on the strands together. First, initial growth vectors are established at each surface node by calculating the normal vector to the surface at each node. The node normal vector is calculated by averaging the normal vectors of the surrounding faces. Then, in order to increase the resolution at convex corners, the strands are smoothed by averaging the growth vectors with those of neighbouring strands (see Fig. 4). The smoothing is controlled by a minimum smoothing residual that is input by the user, with a smaller residual resulting in more smoothing.

The strands are then grown from the surface. The initial cell spacing at the wall, number of cells and the length of each strand are input by the user and a hyperbolic tangent stretching function is used to meet these criteria. The mesh is then checked for any strands that cross and the length of these strands is reduced in a process called "clipping", as shown in Fig. 5.

Finally, ghost cells are created at the surface and at each end of the mesh. At the surface, ghost cells are mirrored about the surface, as this simplifies the setting of boundary conditions. At each end of the surface, the ghost cell creation depends on whether the surface is periodic or not. For periodic surfaces the ghost cells are created using the growth vectors and clipping indices from the opposite end of the mesh. If the surface is not periodic, the initial and final strands are constrained to a plane and the ghost cells are created by mirroring cells about this plane.

The automated meshing enabled by a strand/CAMR solver means that the surface can change shape and the mesh can be regenerated with minimal user input. The high level of automation can be utilised when simulating ablating TPSs, which recess as material is lost through sublimation, pyrolysis and erosion. In this work, a surface deformation algorithm has been developed to allow the mesh to transition between two surfaces in an automated way.



(a) No smoothing applied.

(b) Smoothing residual of $1{\times}10^{-7}$.

Fig 4. The nodes of two strand meshes generated using the surface of the box. One can see that the smoothing process increases the resolution at the corners.

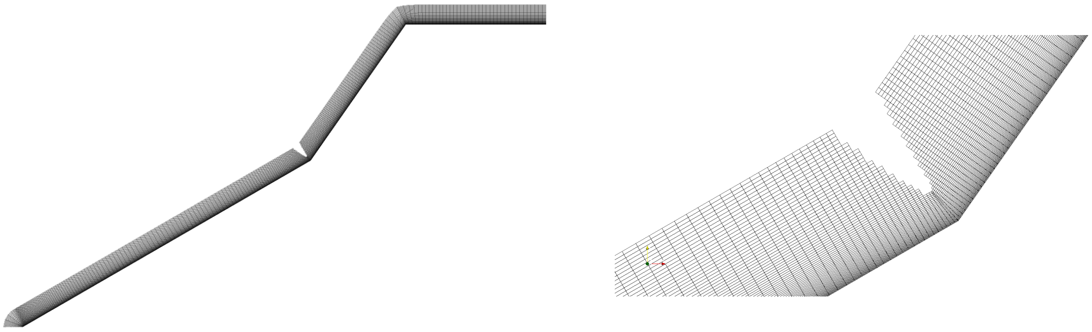


Fig 5. A double wedge with strand clipping applied.

For two surfaces with the same number of vertices and the same vertex connectivity, vertex motion vectors, $\mathbf{v}_{m,s}$, can be determined using the start and end location of each vertex (see Fig. 6). Each vertex can then be moved along its respective motion vector using

$$\mathbf{p}_0^{n+1} = \mathbf{p}_0^n + (\phi^n + \Delta\phi)\mathbf{v}_{m,s}\,. \tag{11}$$

At the start of the motion $\phi^0 = 0$ and the motion is complete when $\phi = 1$. The distance each node travels in a single time step is determined by $\Delta\phi$. The maximum value of $\Delta\phi$ must be found using a stability condition.

In this work, it is assumed that steady-state fluid solutions will be coupled to an unsteady material response solver. This technique is common in hypersonic flow modelling due to the large differences in timescales between the flow and the material response [38, 39]. When using this "loose coupling" method, the surface motion does not need to be considered in the fluid equations and does not need to be time accurate. As such, the stability of the mesh motion is solely dependent on the mesh geometry. In an overset simulation, the maximum motion of a surface vertex is limited by the change in the shape of the level surface used to create a hole in the background mesh. The level surface is able to deform by

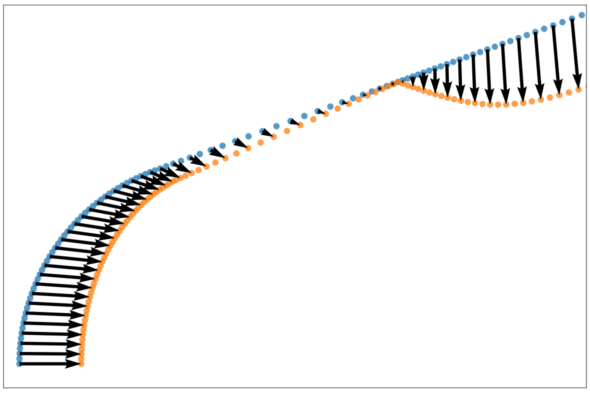


Fig 6. Example surface motion vectors for a recessing surface.

a maximum of N_g min$(\Delta\mathbf{x}_g)$, where N_g is the number of ghost cells on the background mesh and $\Delta\mathbf{x}_g$ is a vector containing the dimensions of the Cartesian ghost cells on the background mesh. Limiting the change in the level surface in this way ensures that all of the cells on the background mesh will contain valid data after the level surface is updated and the hole cutting on the background mesh is performed.

3.4. Overset Domain Assembly

The off-body and near-body solvers must be joined to create a continuous domain. To do so, the overset method, introduced by Benek *et al.* [40], is used, where the domains are joined using boundary conditions. In the overset method the cells at the boundaries of each domain, known as "receptor cells", must be filled with valid information from another domain. This is shown in Fig. 7, where the dark red points represent the receptor centres where flow variables need to be interpolated to in order to define the boundary conditions. The boundary conditions are set by interpolating data from "donor cells" on one mesh to the receptor cells on the other mesh. The Cartesian receptor data is interpolated from the body-fitted domain and the body-fitted receptor data is interpolated from the Cartesian domain. The process of overset domain assembly consists of identifying the receptor cells, finding the donor cells that can be used to fill the receptor cells and then exchanging the required information.

The receptors are found on the off-body mesh by cutting a hole in the mesh that is inside the near-body mesh but outside of the body itself. The receptors are identified as the ghost cells at the edge of the hole. The hole is cut in the background mesh using a level-set method that was previously implemented in AMROC for immersed boundary simulations [35]. The level-set is specified using the Closest Point Transform (CPT) algorithm of Mauch [41], which solves the Eikonal equation to determine the distance of each cell from an arbitrary, triangulated surface. The surface is created using the near-body mesh geometry. On the near-body mesh, the receptors are identified as those at the top of each strand.

Once the receptors cells on both grids have been found, the receptor cell centres are exchanged. At the exchange of the receptors, a point-to-point, inter-processor communication pattern is established to enable efficient overset data exchange on distributed memory machines. The algorithm uses a pre-existing AMROC library [35], which establishes the communication pattern by analysing overlapping bounding boxes from the domains. If the bounding box enclosing the receptor centres from one domain overlaps with the data from another domain, then the two processors will communicate. If not, then no communication is carried out between the processors once the communication pattern has been established.

The data used to fill the receptors is created by interpolating vectors of conserved variables to receptor centres using data from donor cells. Finding the donor cells on the Cartesian off-body mesh is simple, as the cell index containing a given point can be found using the Cartesian box dimensions and cell edge lengths. Linear interpolation using the surrounding cells is then used to calculate the vector of

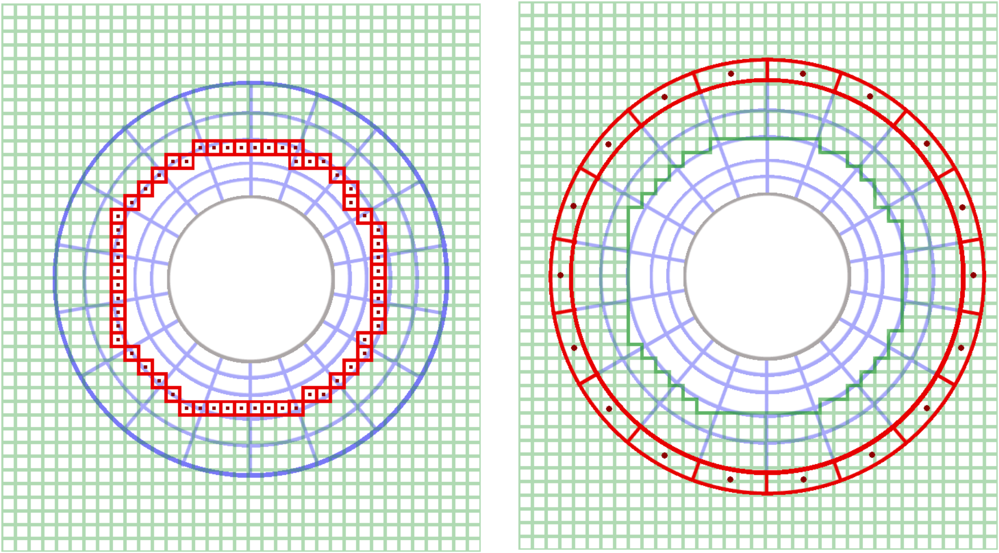


Fig 7. An example of ghost cell locations for the overset boundaries.

conserved variables at the receptor cell centres. On the near-body mesh, the cells surrounding a given interpolation point cannot be found directly as they are dependent on the mapping. In order to find the donor cells a stencil walking algorithm has been implemented using the method described in Ref. [42]. Once the cell containing the interpolation point has been found the bi-linear interpolation weights of the surrounding donor cells are calculated. The donor cell indices and the interpolation weights are then stored so that the donor search only needs to be carried out when the mesh changes.

4. Results

4.1. Overset Order-of-Accuracy Tests

Order-of-accuracy tests were conducted using a time-accurate test case to verify the implementation of both the mapped-mesh spatial- and time-integration, and the overset algorithms. In the test case an isobaric density increase with a Gaussian profile is linearly advected through the domain. The initial density field is given by

$$\rho = \rho_0 + \rho_a \exp\left(-\frac{x^2+y^2}{R^2}\right) , \tag{12}$$

where ρ is the density, ρ_a is the user-specified amplitude and R is a user-specified constant that controls the radius of the density bump. The velocity in each dimension is input by the user and held constant over the simulation. Periodic boundary conditions are used for all of the boundaries so that the Gaussian bump can move through any boundary and return to the domain. This test case was selected for the order-of-accuracy tests as the smooth solution means that the MUSCL limiting can be turned off and full second-order-accuracy can be achieved. To allow the second-order-accuracy of the overset and spatial schemes to be verified in this unsteady test case, an explicit second-order-accurate Strong Stability Preserving (SSP) Runge-Kutta scheme was used, where overset exchanges are carried out prior to each Runge-Kutta stage.

For the order-of-accuracy tests the bump parameters were set to be $\rho_a = 1.0$ and $R = 0.25$. A domain of $x \in [-1, 1]$ and $y \in [-1, 1]$ was used and the x and y velocities were set to be 1.0. The simulation was run for two time units, resulting in the bump moving through the corners of the boundary and returning to the centre of the domain. The domain consisted of a mapped mesh that was fully embedded into a single-level background mesh. The mapping is given by:

$$x = \xi + 0.125\,\text{sin}(0.75\,\pi\,\eta) \tag{13}$$
$$y = \eta + 0.075\,\text{sin}(1.25\,\pi\,\xi)\,. \tag{14}$$

A visualisation of the overset domain and advected density pulse is shown in Fig. 8. The observed order-of-accuracy was obtained by uniformly refining the grid from 50×50 cells to 200×200 cells and the time step from 8 ms to 2 ms. AMROC's in-built error estimators were used to compare the numerical solution to the analytic solution at each time step. The L2-norm of the density error was used as the measure of the numerical accuracy.

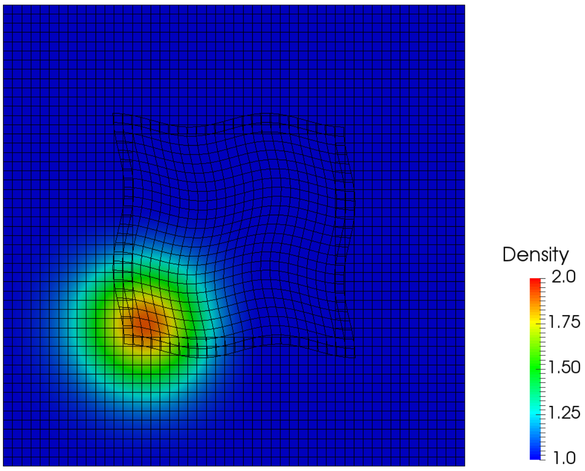


Fig 8. The overset domain used for the order-of-accuracy tests, showing the advected Gaussian density bump.

The order-of-accuracy tests were carried out for different overset time-stepping methods. The first used the SSP Runge-Kutta method on both domains (SSP2-SSP2), the second used the forward Euler method on the off-body domain and an unpartitioned backward Euler (FE-BE) method on the near-body domain and the final simulation used the forward Euler method on the off-body domain and the dimensionally partitioned forward/backward Euler method on the near-body domain (FE-F/BE). The results from the order-of-accuracy tests are shown in Fig. 9. It can be seen that all of the methods give the expected order-of-accuracy. These results provide strong verification evidence for the mapped mesh spatial integration, overset routines and time integration methods.

4.2. High-Enthalpy Cylinder Simulation

To validate the new solver's heat flux predictions and examine the influence of the overset methods on the surface results an experiment of a 90mm diameter cylinder in a high-enthalpy flow of air was simulated. The experiment was conducted in the High Enthalpy Shock Tunnel Göttingen (HEG) facility at the Deutsches Zentrum für Luft- und Raumfahrt (DLR) and measurements were taken of the surface pressure and heat flux. The inflow conditions, given in Table 1, were taken from Ref. [6].

Table 1. Freestream conditions for the HEG cylinder simulation.

T_∞	ρ_∞	U_∞	Y_{N_2}	Y_N	Y_{O_2}	Y_O	Y_{NO}
694 K	3.26 g/m^3	4776 m/s	0.7356	0.0	0.1340	0.07955	0.0509

The strand mesh was created using a quarter cylinder surface with the strands set to be 2 mm in length. The CAMR domain had a geometry of 75 mm $\times$ 100 mm, with a 150×150 cell base mesh. An isothermal wall condition was used on the cylinder surface with a temperature of 300 K. The simulation used a

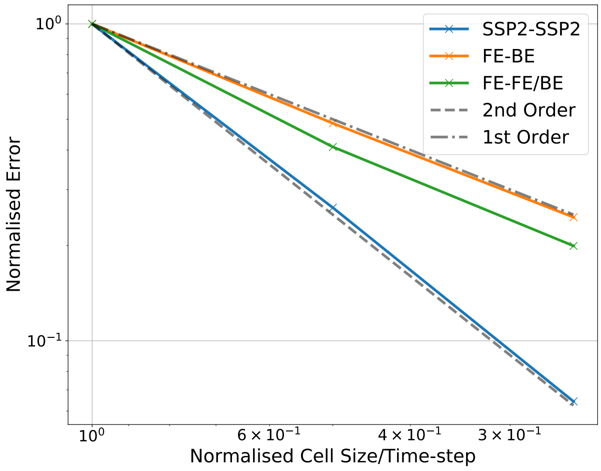


Fig 9. The results from the overset order-of-accuracy tests.

five species air mixture with the Park reaction rate constants. The simulation was integrated in time until the density residual in the near-body region fell to below 1.0$\times10^{-8}$. A grid convergence study was conducted and the final mesh used three levels of refinement in the off-body region, 100×70 cells in the near-wall region and a near-wall spacing of $2\,\mu$m.

Figure 10 shows the nitrogen and temperature fields. One can see the coarsening of the Adaptive Mesh Refinement (AMR) away from the stagnation region, where the shock becomes more diffused. The streamlines from this area passed far from the surface, so coarsening the mesh in this region gives an efficient off-body domain. The image of the nitrogen field shows the chemical nonequilibrium, as the nitrogen dissociates through the shock layer, before recombining in the cooler boundary layer. The thermal nonequilibrium is shown to a lesser extent by the spike in the translational-rotational temperature immediately behind the shock wave, which then rapidly falls away as energy is exchanged with the vibrational and electronic modes.

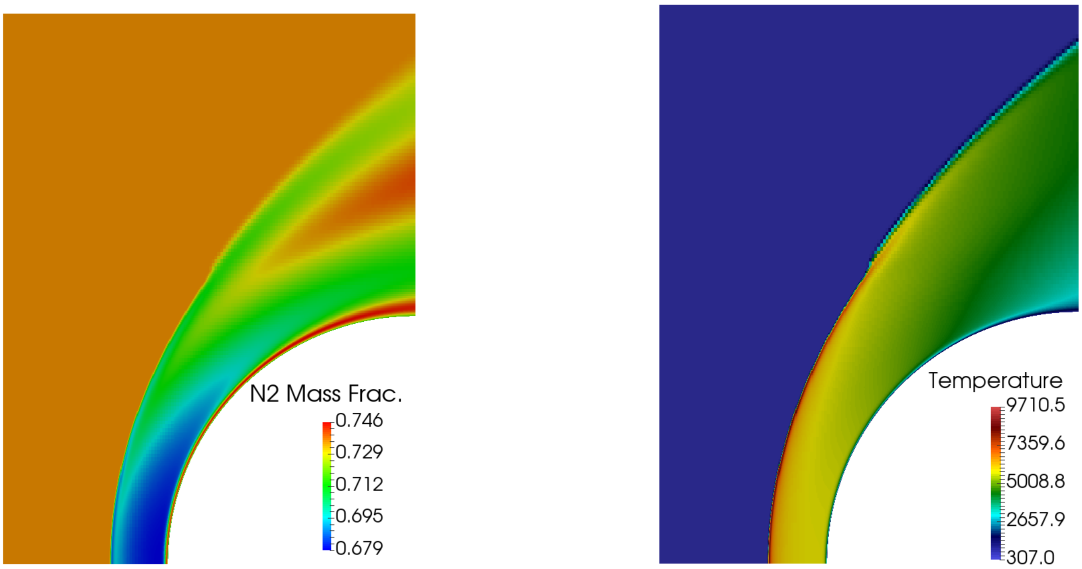
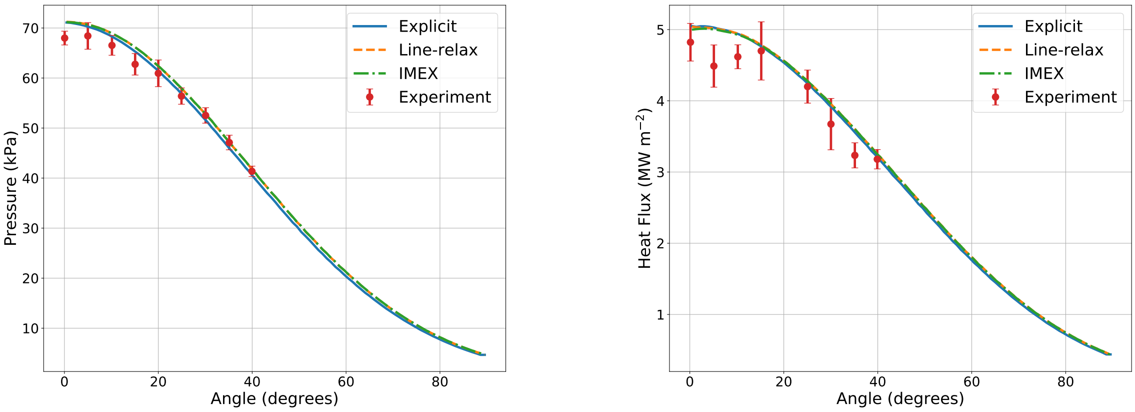


Fig 10. The molecular nitrogen (left) and temperature field (right) in the HEG cylinder simulation.



(a) Surface pressure.

(b) Surface heat flux.

Fig 11. A comparison of the experimental and simulated surface results in the HEG cylinder test case when using different time-integration methods.

This case was used to validate the new implicit and IMEX time-integration methods. The results using the implicit methods are compared with the experimental results and the explicit strand/CAMR results in Fig. 11. It can be seen that excellent agreement is obtained between the different time-integration methods. The AMROC results agree extremely well with the experimental data, with the majority of pressure and heat flux results falling within the experimental uncertainty. This provides excellent validation evidence and demonstrates that the strand/CAMR solver is able to accurately compute stagnation region surface heat fluxes in high-enthalpy flows.

The IMEX method was able to reduce the computational time by approximately 26% compared to the line-implicit method. This was largely due to a reduction in the Jacobian creation time as only the wall-normal direction was integrated implicitly. This clearly demonstrates the benefit of the strand based IMEX method when the Courant-Friedrichs-Lewy (CFL) is limited by the explicit off-body region.

4.2.1. Single Domain Comparisons

Comparisons with single-domain simulations were used to assess the impact of the overset mesh on the results and the computational efficiency of the strand/CAMR method. The experiment was simulated using a single-domain body-fitted mesh in AMROC. To ensure a fair comparison between the two solvers the respective mesh spacings at the shock and at the wall were set to be equal. This required the body-fitted mesh to have a mesh spacing of $2\,\mu$m at the wall, to match the near-body strand mesh, and $125\,\mu$m at the shock, to match the off-body CAMR mesh. To capture the entire shock region the body-fitted mesh extended to 55 mm from the surface. The body-fitted mesh was created using a hyperbolic tangent stretching function, and required 500 cells in the wall-normal direction in order to give the required resolution. This resulted in a total cell count of 49,500. In addition to the single-domain AMROC simulation, the single-domain, two-temperature solver results from Ref. [6] were also used in the comparison.

The surface pressure and heat flux results from the overset and single-domain AMROC simulations are compared with the simulated results from Ref. [6] in Fig. 12. One can see that excellent agreement is obtained between the two AMROC simulations. A comparison of the temperature fields and meshes in the shock region for the two simulations is shown in Fig. 13. One can see that the shock locations in the two simulations are in excellent agreement. These results indicate that the use of an overset mesh has a negligible impact on the surface heating results for blunt body flows.

The AMROC heat flux results compare reasonably well with the results from Ref. [6]. However, there are some differences in the stagnation region, and smaller differences on the upper section of the cylinder.

The single-domain AMROC results clearly show the discrepancy is not due to the use of overset methods, but more likely due to differences in numerical methods and models. For example, the simulations in Ref. [6] used a three-dimensional domain and the centre-line results were used for comparison. In addition, different species production rates and thermodynamic relaxation models were used in the two solvers.

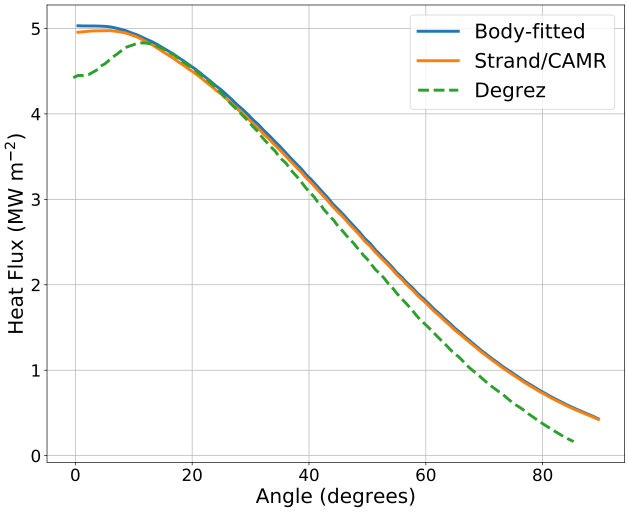


Fig 12. A comparison of the surface heat flux in the HEG cylinder simulation when using single and overset domains.

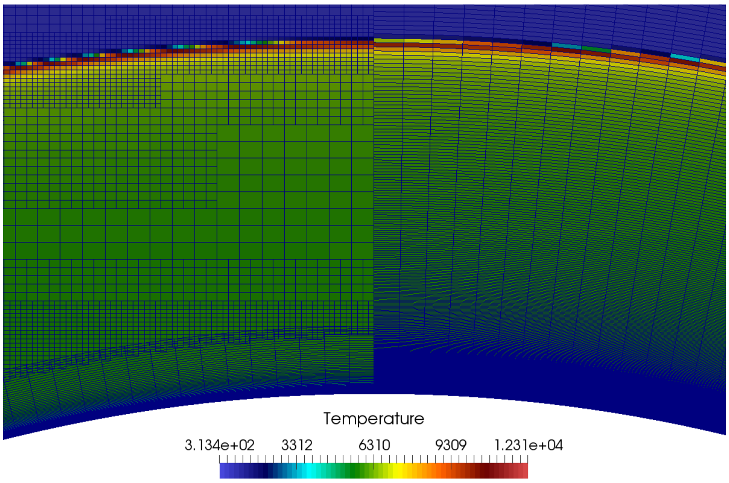


Fig 13. The body-fitted and strand/CAMR mesh in the stagnation region.

4.3. Recessing Nose-tip

The recession rate of a TPS is strongly influenced by the state of the boundary layer. A turbulent boundary layer results in higher heat fluxes and shear stresses, and therefore a higher rate of recession. When the state of the boundary layer changes on a vehicle's surface this can result in differential recession rates between the laminar and turbulent regions. This was observed during the Passive Nosetip Technology (PANT) programme, where extensive studies of ablating nose-tips were conducted [43, 44]. Figure 14 shows an example of a nose-tip shape that arose during experiments. The concave shape was attributed to transition from laminar flow in the stagnation region to turbulent flow close to the sonic point [43]. The shape is characterised by a flat section close to the stagnation point, where the flow is laminar. This is followed by a gouge at the sonic point, caused by the transition to turbulence and a "scalloped" region downstream of the transition [44].

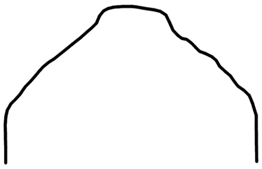


Fig 14. An ablated graphite nose-tip, in a transitional flow regime (redrawn from Ref. [44]).

Researchers found that the nose-tip shape shown in Fig. 14 could lead to the development of an unsteady, double-cone type shock structure. This was experimentally investigated using shape stable, blunted double-cone geometries by Abbett *et al.* [45]. The experiments showed that the unsteadiness was caused by the growth of the separated region at the double-cone intersection. The separated region would grow until the bow shock move far enough from the surface that the mass trapped in the separated region could escaped over the shoulder of the upper cone [45]. This would cause the size of the separated region to reduce and the shock structure to collapse towards the surface. The process was seen to repeat periodically, resulting in a "pulsating" flow pattern. The violent nature of the unsteadiness raised concerns that it could lead to structural issues if it were to occur on a hypersonic vehicle [45].

The aim of this test case is to demonstrate the automated surface motion and mesh refinement that is enabled by the strand/CAMR solver. A hypothetical experiment, similar to those conducted in the PANT program, is simulated with the new solver. In the experiment a nose-tip is exposed to an arc-heater flow and ablates in such a way to produce the shape shown in Fig. 14 and the resulting unsteady flow.

The flow field in the arc-heater simulation is based on the DLR L3K arc-heater. The nozzle was first simulated in isolation and then the outflow from the nozzle was used as the inflow to the arc heater test section. The test piece was placed at 0.1 m from the nozzle exit, which in experiments has been found to give a nominal heat flux of 10 MW m^{-2}. To validate the computational set-up, a simulation of a forward facing cylinder was conducted as this is a typical configuration used with the DLR arc-heater [46]. The cylinder surface used an isothermal boundary condition, with the temperature set as 350 K, as per Ref. [46]. The species concentrations were set to give the equilibrium composition at 350 K, using the catalytic wall boundary condition. The simulation was run using the AUSM flux scheme and a five species mixture of air. The flow field for the validation case is shown in Fig. 15 and heat flux results are shown in Fig. 16. One can see that the heat flux is in good agreement with the expected 10 MW m^{-2} heat flux in the stagnation region. It can be seen that there are significant heat flux contributions from both the diffusive and conductive terms.

A nose-tip was created based on the IRV-2 vehicle, with the geometry taken from Ref. [39]. The nose-tip radius was 0.01905 m and the cone angle 9 degrees. The ablated shape was based on the parameterised shapes used in the PANT experiments [45]. The shape is specified by a nose radius for the first cone, R_n, first cone angle, θ_1, first cone length, L, fillet radius, R_f, second cone angle, θ_2, shoulder radius, R_s, and test piece diameter, D, (see Fig. 17).

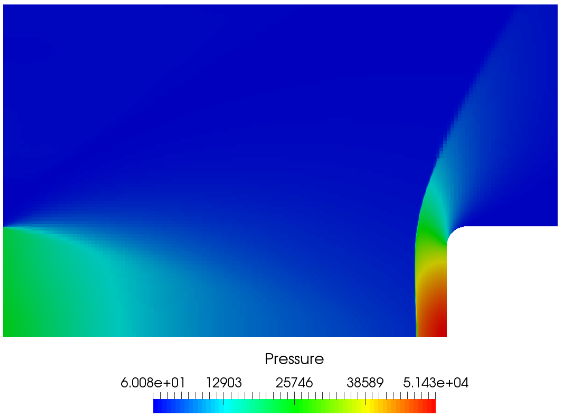


Fig 15. The pressure field in the DLR L3K test box when using a cylindrical test piece.

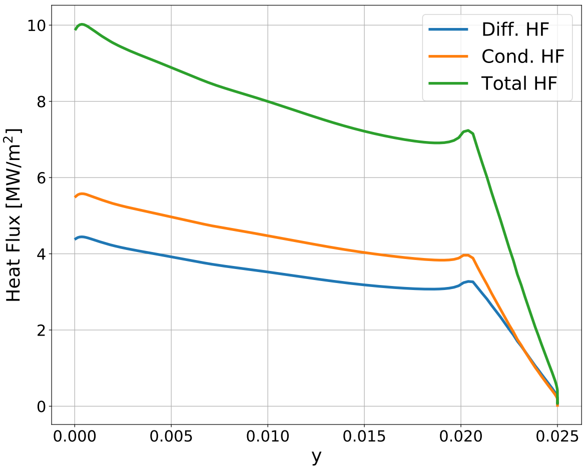


Fig 16. The surface heat flux results for the cylindrical test piece in the DLR L3K arc heater.

The geometric parameters were selected as those likely to give an unsteady flow based on the results from the experimental studies detailed in Ref. [45]. In addition, they were chosen to give a profile qualitatively similar to that obtained in ablation experiments. The parameters used in the simulations are given in Table 2. Finally, the surface of the second cone was given a sinusoidal profile in order to mimic the scalloping that can result from turbulent flow ablation and to further test the automated mesh generation algorithms. The ablated experimental surface from Ref. [44] and the two surfaces used in the simulation (unablated and ablated) are shown in Fig. 18.

Table 2. The parameters used to define the ablated nose-tip shape.

$D/2$ (m)	L_1/D	R_n/D	θ_1 (deg)	θ_2 (deg)	R_f/D	R_s/D
0.01905	0.25	0.15	20	65	0.075	0.15

The strand mesh around the original nose-tip shape was created using 51 nodes on each strand, with a wall spacing of $1\,\mu$m and strand length of 1 mm. The background domain included the entire test box and used a 100×70 base mesh with five levels of refinement, each with a refinement factor of two.

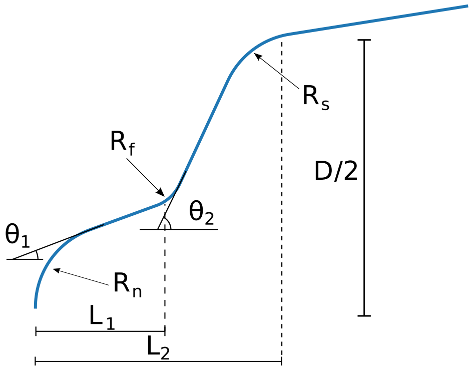


Fig 17. The parameters used to define the ablated nose-tip shape.

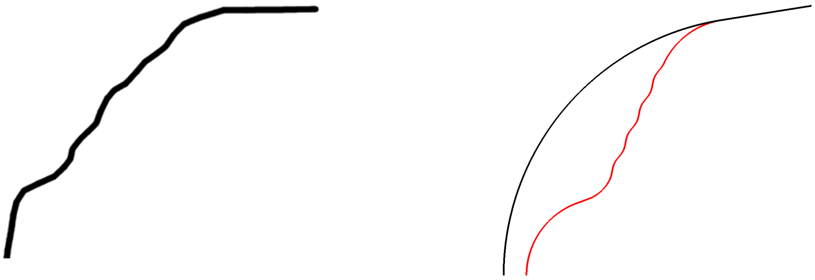


Fig 18. The upper half of the ablated nose-tip image redrawn from Ref. [44] (left) and the unablated and ablated nose-tip surfaces used in the simulations (right).

First, the unablated shape was run until the flow became steady. Then the surface was dynamically updated by moving the surface mesh from the initial surface to the final surface during the simulation, without having to re-initialise the flow field. Figure 19 shows the stages of the mesh motion from the original nose-tip shape to the ablated nose-tip shape. The only inputs required by the user were the initial surface, final surface and the surface deformation CFL (set to be 0.75).

The pulsating flow is shown in Fig. 20. One can see that the dynamic mesh adaptation is able to accurately track the shock structures as they evolve. This test case clearly demonstrates the new solver's ability to automatically generate meshes around recessing surfaces with a large surface deformation, whilst resolving the dynamic shock structures.

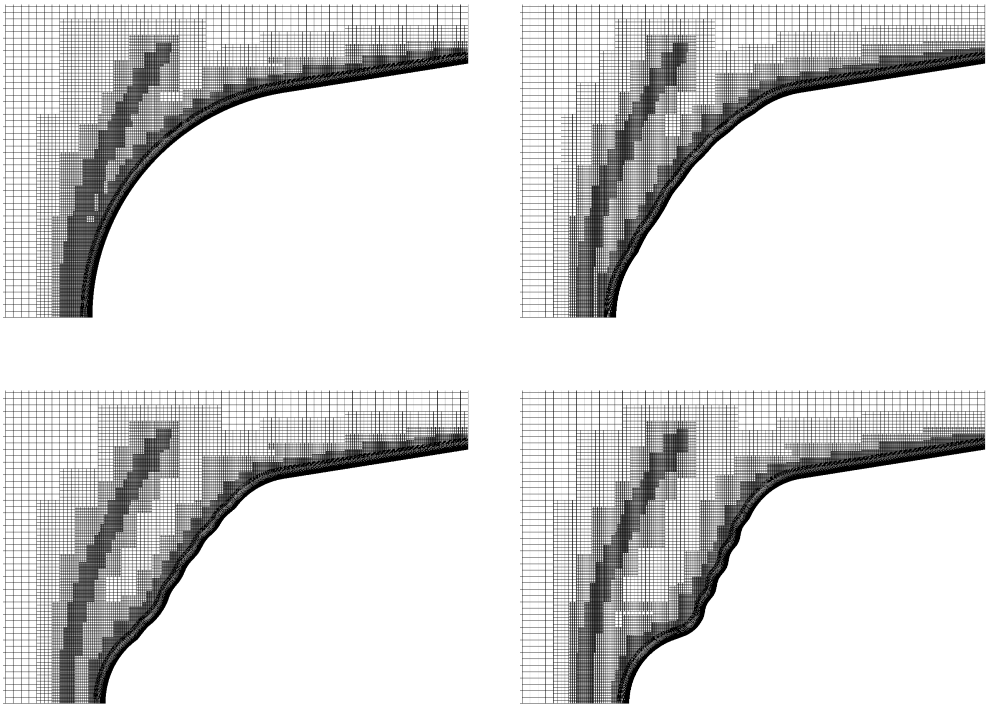


Fig 19. The automated mesh motion for the ablating nose-tip.

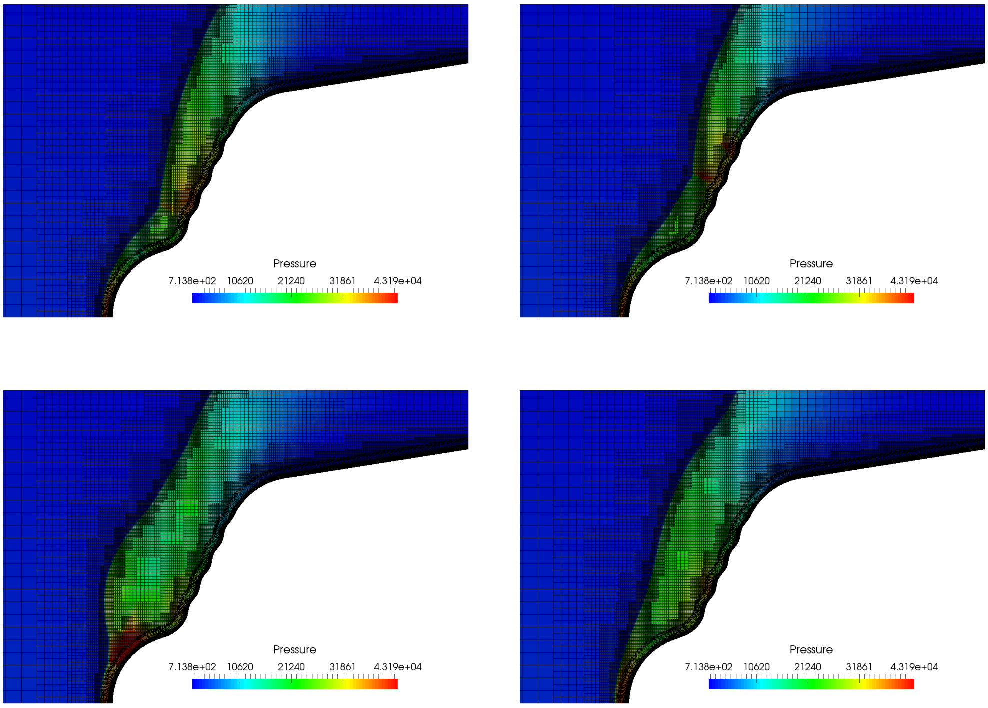


Fig 20. The automated mesh adaptation captures the pulsating shock structure.

5. Conclusion

A 2D/axi-symmetric strand/CAMR solver, that uses a thermochemical nonequilibrium two-temperature model, has been developed for modelling hypersonic flows. With minimal user input the new solver is able to generate a mesh from a discretised surface and can be used for highly-automated mesh deformation simulations. The overset solver has been implemented for use on distributed memory systems, with efficient point-to-point communication algorithms implemented for the overset boundary condition exchange. The solver has been verified and validated with a number of test cases and the automated meshing for recessing surfaces has been demonstrated. This work shows that the strand/CAMR approach is able to accurately predict surface pressure and heat fluxes, whilst enabling a high level of automation both in terms of shock capturing and surface deformation. Consequently, strand/CAMR solvers could reduce the resources required for through-flight CFD analysis of ablating vehicles.

Acknowledgments

The funding provided by UK Defence Science and Technology Laboratory (Dstl) is gratefully acknowledged.

References

[1] E. Josyula, *Hypersonic Nonequilibrium Flows: Fundamentals and Recent Advances*. American Institute of Aeronautics and Astronautics, 2015.

[2] G. V. Candler and R. W. MacCormack, "Computation of weakly ionized hypersonic flows in thermochemical nonequilibrium," *Journal of Thermophysics and Heat Transfer*, vol. 5, no. 3, pp. 266–273, 1991.

[3] M. J. Wright, G. V. Candler, and D. Bose, "Data-Parallel Line Relaxation method for the Navier-Stokes equations," *AIAA Journal*, vol. 36, no. 9, 1998.

[4] L. C. Scalabrin and I. D. Boyd, "Numerical simulation of weakly ionized hypersonic flow for reentry configurations," in *9th AIAA/ASME Joint Thermophysics and Heat Transfer Conference*, no. June, AIAA, 2006.

[5] I. Nompelis, T. Drayna, and G. Candler, "A parallel unstructured implicit solver for hypersonic reacting flow simulation," in *Parallel Computational Fluid Dynamics 2005*, pp. 389–395, Elsevier, 2006.

[6] G. Degrez, A. Lani, M. Panesi, O. Chazot, and H. Deconinck, "Modelling of high-enthalpy, high-Mach number flows," *Journal of Physics D: Applied Physics*, vol. 42, 2009.

[7] G. V. Candler, B. Johnson, Heath, I. Nompelis, P. K. Subbareddy, T. W. Drayna, V. Gidzak, and M. D. Barnhardt, "Development of the US3D code for advanced compressible and reacting flow simulations," in *53rd AIAA Aerospace Sciences Meeting*, 2015.

[8] V. Casseau, T. J. Scanlon, and R. E. Brown, "Development of a two-temperature open-source CFD model for hypersonic reacting flows," in *20th AIAA International Space Planes and Hypersonic Systems and Technologies Conference*, pp. 1–14, 2015.

[9] S. R. Copeland, *A Continuous Adjoint Formulation For Hypersonic Flows In Thermochemical Non-Equilibrium*. PhD thesis, Stanford University, 2015.

[10] P. A. Gnoffo and J. A. White, "Computational aerothermodynamic simulation issues on unstructured grids," in *37th AIAA Thermophysics Conference*, no. July, pp. 1–17, 2004.

[11] G. V. Candler, M. D. Barnhardt, T. W. Drayna, I. Nompelis, D. M. Peterson, and P. Subbareddy, "Unstructured grid approaches for accurate aeroheating simulations," in *18th AIAA Computational Fluid Dynamics Conference*, no. June, 2007.

[12] A. Martin and I. D. Boyd, "Mesh tailoring for strongly coupled computation of ablative material in nonequilibrium hypersonic flow," in *10th AIAA/ASME Joint Thermophysics and Heat Transfer Conference*, no. June, 2010.

[13] D. Saunders, S. Yoon, and M. Wright, "An approach to shock envelope grid tailoring and its effect on reentry vehicle solutions," in *45th AIAA Aerospace Sciences Meeting and Exhibit*, 2007.

[14] A. Martin and I. D. Boyd, "Strongly coupled computation of material response and nonequilibrium flow for hypersonic ablation," in *41st AIAA Thermophysics Conference*, 2009.

[15] R. L. Meakin, W. M. Chan, S. A. Pandya, and J. Sitaraman, "On strand grids for complex flows," in *18th AIAA Computational Fluid Dynamics Conference*, 2007.

[16] B. Roget, J. Sitaraman, V. Lakshminarayan, and A. Wissink, "Prismatic mesh generation using minimum distance fields," in *Tenth International Conference on Computational Fluid Dynamics*, 2018.

[17] R. Haimes, "MOSS: Multiple orthogonal strand system," in *22nd International Meshing Roundtable*, Springer, 2014.

[18] J. Sitaraman, V. K. Lakshminarayan, B. Roget, and A. M. Wissink, "Progress in strand mesh generation and domain connectivity for dual-mesh CFD simulations," *55th AIAA Aerospace Sciences Meeting*, vol. m, no. January, pp. 1–23, 2017.

[19] A. Katz and A. M. Wissink, "Efficient solution methods for strand grid applications," *AIAA Journal*, vol. 52, no. 2, 2014.

[20] A. Katz and D. Work, "High-order flux correction/finite difference schemes for strand grids," *Journal of Computational Physics*, vol. 282, pp. 360–380, 2015.

[21] O. Tong, A. Katz, Y. Yanagita, and D. Work, "Verification and validation of a high-order strand grid method for two-dimensional turbulent flows," *Computers and Fluids*, vol. 154, pp. 335–346, 2017.

[22] C. Park, *Nonequilibrium Hypersonic Aerothermodynamics*. Wiley-Interscience, 1990.

[23] K. Sutton and P. A. Gnoffo, "Multi-component diffusion with application to computational aerothermodynamics," *7th AIAA / ASME Joint Thermophysics and Heat Transfer Conference*, 1998.

[24] J. B. Scoggins and T. E. Magin, "Development of Mutation++ : MUlticomponent Thermodynamics And Transport properties for IONized gases library in C++," in *11th AIAA/ASME Joint Thermophysics and Heat Transfer Conference*, 2014.

[25] C. R. Wilke, "A viscosity equation for gas mixtures," *The Journal of Chemical Physics*, vol. 18, no. 4, pp. 517–519, 1950.

[26] R. N. Gupta, J. M. Yos, and R. A. Thompson, "A review of reaction rates and thermodynamic and transport properties for the 11-species air model for chemical and thermal nonequilibrium calculations to 30,000K," Tech. Rep. February, NASA, 1989.

[27] T. E. Magin, "Transport algorithms for partially ionized and unmagnetized plasmas," *Journal of Computational Physics*, vol. 198, pp. 424–449, 2004.

[28] V. Giovangigli, B. Graille, T. Magin, and M. Massot, "Multicomponent transport in weakly ionized mixtures," *Plasma Sources Science and Technology*, vol. 19, pp. 1–6, 2010.

[29] G. E. Palmer and M. J. Wright, "Comparison of methods to compute high-temperature gas viscosity," *Journal of Thermophysics and Heat Transfer*, vol. 17, no. 2, pp. 232–239, 2003.

[30] R. C. Millikan and D. R. White, "Systematics of vibrational relaxation," *The Journal of chemical physics*, vol. 39, no. 12, pp. 3209–3213, 1963.

[31] C. Park, "Review of chemical-kinetic problems of future NASA missions, I - Earth entries," *Journal of Thermophysics and Heat Transfer*, vol. 7, no. 1, pp. 9–23, 1993.

[32] G. Duffa, *Ablative Thermal Protection Systems Modeling*. 2011.

[33] M. J. Berger and P. Colella, "Local adaptive mesh refinement for shock hydrodynamics," *Journal of Computational Physics*, vol. 82, no. 1, pp. 64–84, 1989.

[34] R. Deiterding, *Parallel Adaptive Simulation of Multi-Dimensional Detonation Structures*. PhD thesis, 2003.

[35] R. Deiterding, "Block-structured adaptive mesh refinement: Theory, implementation and application," *ESAIM Proceedings*, vol. 34, pp. 97–150, 2011.

[36] M.-S. Liou and C. J. Steffen Jr, "A new flux splitting scheme," *Journal of Computational Physics*, vol. 107, pp. 23–39, 1993.

[37] A. Katz, A. M. Wissink, V. Sankaran, R. L. Meakin, and W. M. Chan, "Application of strand meshes to complex aerodynamic flow fields," *Journal of Computational Physics*, vol. 230, no. 17, pp. 6512–6530, 2011.

[38] G. V. Candler, "Rate dependent energetic processes in hypersonic flows," *Progress in Aerospace Sciences*, vol. 72, pp. 37–48, 2015.

[39] D. W. Kuntz, B. Hassan, and D. L. Potter, "Predictions of ablating hypersonic vehicles using an iterative coupled fluid/thermal approach," *Journal of Thermophysics and Heat Transfer*, vol. 15, 2001.

[40] J. Benek, P. Buning, and J. Steger, "A 3-D chimera grid embedding technique," in *7th Computational Physics Conference*, p. 1523, 1985.

[41] S. Mauch, *Efficient algorithms for solving static Hamilton-Jacobi equations*. PhD thesis, California Institute of Technology, 2003.

[42] R. L. Meakin, "Composite overset structured grids," in *Handbook of grid generation* (J. F. Thompson, B. K. Soni, and N. P. Weatherill, eds.), ch. 11, CRC press, 1999.

[43] A. B. Shimizu, J. E. Ferrell, and C. A. Powars, "Passive Nosetip Technology (PANT) Program: Nosetip transition and shape change tests in the AFFDL 50MW rent arc - Data report," tech. rep., 1974.

[44] M. R. Wool, "Passive Nosetip Technology (PANT) Program: Summary of experimental and analytical results," tech. rep., Acurex Corporation, 1975.

[45] M. J. Abbett, L. Cooper, T. J. Dahm, and M. D. Jackson, "Passive Nosetip Technology (PANT) Program: Unsteady flow on ablated nosetip shapes - PANT Series G test and analysis report," Tech. Rep. December, 1973.

[46] M. K. Kim, B. Esser, U. Koch, and A. Gülhan, "Numerical and experimental study of high enthalpy flows in a hypersonic plasma wind tunnel: L3K," in *42nd AIAA Thermophysics Conference*, no. AIAA Paper 2011-3777, 2011.