Received July 2003
Revised February 2004
Accepted February 2004

Parallel computation of two-dimensional laminar inert and chemically reactive multi-species gas flows

P.R. Ess
Department of Aerospace Engineering, University of Bristol, Queen's Building, University Walk, Bristol, UK and
Department of Aeronautics, Imperial College, London, UK

C.B. Allen
Department of Aerospace Engineering, University of Bristol, Queen's Building, University Walk, Bristol, UK

Abstract

Purpose – A computational fluid dynamics code for the calculation of laminar hypersonic multi-species gas flows in chemical non-equilibrium in axisymmetric or two-dimensional configuration on shared and distributed memory parallel computers is presented and validated. The code is designed to work efficiently in combination with an automatic domain decompositioning method developed to facilitate efficient parallel computations of various flow problems.

Design/methodology/approach – The baseline implicit numerical method developed is the lower-upper symmetric Gauss-Seidel scheme, which is combined with a sub-iteration scheme to achieve time-accuracy up to third-order. The spatial discretisation is based on Roe's flux-difference splitting and various non-linear flux limiters maintaining total-variation diminishing properties and up to third-order spatial accuracy in continuous regions of flow. The domain subdivision procedure is designed to work for single- and multi-block domains without being constrained by the block boundaries, and an arbitrary number of processors used for the computation.

Findings – The code developed reproduces accurately various types of flows, e.g. flow over a flat plate, diffusive mixing and oscillating shock induced combustion around a projectile fired into premixed gas, and demonstrates close to linear scalability within limits of load imbalance.

Research limitations/implications – The cases considered are axisymmetric or two-dimensional, and assume laminar flow. An extension to three-dimensional turbulent flows is left for future work.

Originality/value – Results of a parallel computation, utilising a newly developed automatic domain subdivision procedure, for oscillating shock-induced combustion around a projectile and various other cases are presented. The influence of entropy correction in Roe's flux-difference splitting algorithm on diffusive mixing of multi-species flows was examined.

Keywords Parallel connection, Laminar flow, Equilibrium methods

Paper type Research paper

International Journal for Numerical Methods in Heat & Fluid Flow
Vol. 15 No. 3, 2005
pp. 228-256

0961-5539
DOI 10.1108/09615530510583865

Various advice with respect to computing related issues, especially the implementation of efficient parallel code, by Dr Ian Stewart is gratefully acknowledged. The Department of Mathematics at the University of Bristol is gratefully acknowledged for providing substantial computational time on their parallel computer.

Introduction

The computational fluid dynamics code developed and presented in this paper is aimed at the calculation of hypersonic multi-species gas flows in chemical non-equilibrium in axisymmetric or two-dimensional configuration. The numerical study of such flows has become more feasible due to the steadily increasing performance of computers, but is still expensive for realistic configurations. Edwards (1992) states that solving the reacting flow around a hypersonic vehicle in powered flight requires hundreds of hours of computing time on a Cray Y-MP. For a three-dimensional computation of viscous, reactive flow mesh sizes easily exceed a million cells (von Lavante *et al.*, 2001) for solving only a specific, limited problem. Fine meshes are required to properly resolve shock waves and viscous shear layers, severely reducing the allowed time-step, especially in the case of a time-accurate calculation. This reflects the excessive computational cost of correctly predicting these flows, which usually requires the inclusion of multiple species and complex reaction schemes to simulate the non-equilibrium chemistry. In order to achieve reasonable run-times the computational code developed has been parallelized for distributed memory parallel computers.

The application of computational results must be accepted with great care when not confirmed by appropriate experiments. Edwards (1992) describes the numerical analysis of hypersonic, chemically reacting flows, Marvin (1992) provides "A CFD Validation Roadmap For Hypersonic Flows" and Dolling (1992) presents "Problems in the Validation of CFD Codes Through Comparison with Experiment" when concerned with the design of hypersonic vehicles.

The test cases considered in this paper are conceptually different to ensure the proper functioning of the respective part of the flow solver. Initially, the flow over a flat plate is simulated to verify the correct boundary layer development and, hence, implementation of viscous terms. Next, the diffusive mixing of two jets of different species is reproduced numerically to ensure the correct diffusion flux calculation. As a last, and most demanding example, shock, oscillating shock and detonation induced combustion around a projectile fired into a premixed combustible multi-species gas is calculated. With this, very demanding, case the quality of the time-accurate numerical integration is tested.

A selection of works on time-accurate calculation of chemically reacting, viscous flow can be found for ram-accelerators and expansion tubes. Nusca and Kruczynski (1996) solve the viscous flow around a realistic projectile with four fins and Nusca (2002) the axisymmetric flow around a projectile. Similar time-accurate, axisymmetric calculations for flow around a projectile were performed by Yungster *et al.* (1998) and Choi *et al.* (1998), concentrating on the starting process. A more detailed study of shock-wave and boundary-layer interaction on the projectile and tube wall surfaces is presented by Yungster (1992). Time-accurate calculations for chemically reactive flow in expansion tubes were performed by the same authors, Choi *et al.* (1999) and Yungster and Radhakrishnan (2001). Again, Choi *et al.* (2000*a*, *b*) examine oscillating shock-induced combustion, as experimentally recorded by Lehr (1972). In these papers by Choi *et al.* different time-marching methods and limiters are examined. For the lower-upper symmetric Gauss-Seidel (LU-SGS) scheme employed by Choi *et al.* only second-order accuracy in time is maintained, as in most other papers mentioned before. Results for the latter two papers by Choi *et al.* are an important basis for the validation of the flow solver presented here, which employs both higher-order accuracy in time, and particularly effective parallel decomposition and execution.

The parallel flow solver for the calculation of chemically reactive steady-state and time-accurate flows utilizing detailed chemical reaction schemes presented in this paper offers up to third order accuracy in space and time. Out of various possibilities for the spatial discretization described in Hirsch (1988, Vol. 1), a cell-centred finite-volume scheme is employed in combination with a discretization of the domain by multi-block grid systems based on structured meshes. The cell-centred finite-volume approach has been chosen for its inherent mass conservation and because it avoids numerical singularities or ambiguities that can occur at domain boundaries for a node-based discretization.

For the cell-centred finite-volume approach fluxes through the cell faces are considered. Hirsch (1988, Vol. 2, pp. 408-583), contains a broad selection of upwind and high-resolution schemes for the calculation of fluxes through cell faces. Based on the experience made by other researchers Roe's flux-difference splitting (FDS) scheme is selected for its good shock capturing capabilities (Sweby, 1984) and superior reproduction of contact discontinuities (Shuen *et al.*, 1990). The FDS is used in combination with various total-variation diminishing (TVD) flux limiters described in Yee and Shinn (1989), Yee *et al.* (1990) and Shuen (1992). Roe's FDS scheme is known to exhibit the carbuncle phenomenon, as discussed in further detail by Gressier and Moschetta (2000) and Pandolfi and D'Ambrosio (2001). While most analysis concentrate on the entropy correction required to avoid the carbuncle phenomenon in connection with shock waves, in this paper the influence of entropy correction on the solution accuracy with respect to multi-species diffusion is examined. This allows to assess to what extent the accurate calculation of diffusive effects is degraded when the physical soundness of shock waves is ensured.

As for the spatial discretization there are many possibilities available to accomplish the temporal discretization. The selection of an implicit scheme is mandatory due to the high stiffness of the system of equations, where the stiffness of a system of equations is characterized by the minimum and maximum absolute value of the eigenvalues. The chemical source terms introduce largely different time-scales, severely limiting the allowable time step for an explicit method. Again, based on the experience made by researchers solving chemically reacting, supersonic flows, shown for instance in Shuen (1992) and Choi *et al.* (2000a), the LU-SGS scheme is used to solve the resulting system of equations and march the solution forward in time. The fundamentals of this method can be found in papers by Jameson and Turkel (1981) and Yoon and Jameson (1988). Owing to the scheme's inherent factorization error time-accuracy cannot be achieved directly. To compensate for the factorization error and obtain a time-accurate scheme of up to third order accuracy, backward differences are used in combination with several sub-iterations performed for each physical time step (Choi *et al.*, 2000*a*).

The species properties are calculated using polynomials as suggested by McBride *et al.* (1993) and the transport properties applying rules of the kinetic gas theory, as outlined by Hirschfelder *et al.* (1954) and Gardiner (1984). The chemical reaction scheme used was provided by Wilson and MacCormack (1992), where some modifications were applied in order to make the original scheme by Jachimowski (1988), developed for the NASP program, more accurate for higher pressure levels.

The paper is presented as follows: after the introduction the governing equations, their numerical implementation and aspects of parallelization are described. Next, the three test cases and related numerical simulations are presented before conclusions are drawn.

Governing equations

Using density ρ, velocities u and v, pressure p, total energy E, species mass fraction Y, chemical source term ω, shear tension τ, diffusion and energy fluxes $\underline{j}$ and $\underline{\varepsilon}$ and axial and radial coordinates x and y, the axisymmetric, laminar Navier-Stokes equations for a chemically reacting multi-species gas flow with N_s components are given in conservative, differential form by

$$\frac{\partial}{\partial t}\underline{Q} + \frac{\partial}{\partial x}(\underline{G}_c - \underline{G}_d) + \frac{1}{y}\,\frac{\partial}{\partial y} y(\underline{H}_c - \underline{H}_d) = \underline{S} \tag{1}$$

with

$$\underline{Q} = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho E \\ \rho Y_1 \\ \vdots \\ \rho Y_{N_S-1} \end{bmatrix}, \quad \underline{S} = \begin{bmatrix} 0 \\ 0 \\ \frac{p}{y} - \frac{\tau_{\theta\theta}}{y} \\ 0 \\ \omega_1 \\ \vdots \\ \omega_{N_s-1} \end{bmatrix} \quad \underline{G}_c = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ u(\rho E + p) \\ \rho u Y_1 \\ \vdots \\ \rho u Y_{N_s-1} \end{bmatrix}, \quad \underline{H}_c = \begin{bmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ v(\rho E + p) \\ \rho v Y_1 \\ \vdots \\ \rho v Y_{N_s-1} \end{bmatrix}$$

$$\underline{G}_d = \begin{bmatrix} 0 \\ \tau_{xx} \\ \tau_{xy} \\ u\tau_{xx} + v\tau_{xy} - \varepsilon_x \\ -j_{x_1} \\ \vdots \\ -j_{x_{N_s-1}} \end{bmatrix}, \quad \underline{H}_d = \begin{bmatrix} 0 \\ \tau_{yx} \\ \tau_{yy} \\ u\tau_{xy} + v\tau_{yy} - \varepsilon_y \\ -j_{y_1} \\ \vdots \\ -j_{y_{N_s-1}} \end{bmatrix}$$

where

$$\tau_{xx} = \frac{2}{3}\mu\left(2\frac{\partial u}{\partial x} - \frac{\partial v}{\partial y} - \frac{v}{y}\right) \tag{2}$$

$$\tau_{yy} = \frac{2}{3}\mu\left(2\frac{\partial v}{\partial y} - \frac{\partial u}{\partial x} - \frac{v}{y}\right) \tag{3}$$

$$\tau_{xy} = \mu\left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right) \tag{4}$$

$$\tau_{\theta\theta} = \frac{2}{3}\mu\left(2\frac{v}{y} - \frac{\partial v}{\partial y} - \frac{\partial u}{\partial x}\right) \tag{5}$$

and μ is the viscosity of the gas mixture. The species diffusion is calculated using Fick's law, where the diffusion coefficient $D_{i,\mathrm{mix}}$ for species i with respect to the gas mixture replaces the binary diffusion coefficients (Warnatz and Maas, 1993)

$$\underline{j}_i = -\rho D_{i,\mathrm{mix}} \nabla Y_i \tag{6}$$

Fick's law is an approximation of the multi-component diffusion equations obtained from the kinetic theory of gases, which significantly reduces the computational costs of diffusion flux calculation. The validity of this approximation is discussed by Williams (1985).

Further, the energy flux vector is calculated by summarizing heat conduction and enthalpy diffusion according to

$$\underline{\varepsilon} = -\lambda \nabla T + \sum_{i=1}^{N_s} h_i \underline{j}_i \tag{7}$$

where λ is the heat conductivity coefficient.

Of great importance for any reactive flow is the production and destruction of a species, described by the source term

$$\omega_j = M_j \sum_{l=1}^{N_r} [M]_l \cdot (\nu''_{j,l} - \nu'_{j,l}) \cdot \left(k_{f,l} \prod_{i=1}^{N_s} [M_i]^{\nu'_{i,l}} - k_{b,l} \prod_{i=1}^{N_s} [M_i]^{\nu''_{i,l}} \right) \tag{8}$$

where the third body efficiency $[M]_l$ for reaction l can be expressed as

$$[M]_l = \alpha_{l,0} + \sum_{k=1}^{N_s} \alpha_{l,k} \ [M_k] \ \text{ and } \quad \alpha_{l,0} = \begin{cases} 1 & \text{for max } (\alpha_{l,1}, \ldots, \alpha_{l,N_s}) = 0 \\ 0 & \text{otherwise} \end{cases}$$

and $\nu''_{j/i,l}$ and $\nu'_{j/i,l}$ are the stoichiometric coefficients for the reaction scheme used. Further, the concentration of species i is given by $[M_i]$. The forward reaction rates $k_{f,l}$ are calculated using the extended Arrhenius equation, while the reverse rates $k_{b,l}$ are determined with the help of the equilibrium constant calculated from the Gibbs free-energy condition. The calculation of the chemical source terms is straigtforward with known species properties and chemical reaction scheme. However, great care must be taken when selecting a particular reaction scheme for a specific problem, as reactions schemes and their development can be rather complex and the schemes are not generally valid at all thermodynamic conditions. More details can be found in Warnatz and Maas (1993).

Thermodynamic closure of the system of equations is achieved by the the thermal equation of state for a multi-species gas;

$$p = \rho R T = \rho R^* T \sum_{i=1}^{N_s} \frac{Y_i}{M_i} \tag{9}$$

with R^* as universal gas constant and $R = R^*/M$ as gas constant for the mixture. Further, the enthalpy of the gas mixture is calculated according to

$$h = \sum_{i=1}^{N_s} Y_i h_i = \sum_{i=1}^{N_s} Y_i \left(h_{fi}^0 + \int_{T_{\text{ref}}}^{T} c_{p_i} \, \mathrm{d}T \right) \tag{10}$$

where h_{fi}^0 is the heat of formation at the given reference temperature T_{ref}. The specific heats at constant pressure c_{p_i} are computed utilizing polynomials with appropriate coefficients provided by McBride *et al.* (1993).

The transport properties viscosity, thermal conductivity and binary diffusion coefficients, are calculated for each species following the laws of kinetic gas theory (Hirschfelder *et al.*, 1954). Next, mixing rules are applied to obtain the appropriate properties for the multi-species gas up to a 5-10 per cent accuracy, as outlined by Warnatz and Maas (1993).

Numerical method

The governing equations are solved with an implicit, cell-centred, structured multi-block, finite-volume scheme with an accuracy up to third order in time and space. The conservative variables and the source terms are approximated as constant within a cell of fixed geometry in order to develop the finite-volume formulation for the governing equations. At the same time the residual vector $\underline{R}$ is introduced to give

$$V \frac{\partial}{\partial t} \underline{Q} = - \int_V \left[\frac{\partial}{\partial x} (\underline{G}_c - \underline{G}_d) + \frac{1}{y} \frac{\partial}{\partial y} y (\underline{H}_c - \underline{H}_d) \right] \mathrm{d}V + V \underline{S} = -\underline{R} \tag{11}$$

where the volume integration of the fluxes can be replaced by a surface integration using the Gauss identity. Further, this decoupling of the temporal and spatial discretization allows the introduction of the semi-discrete form of the equations, where both parts can be derived independently.

Spatial discretization

Roe's FDS in combination with various TVD limiters is used to discretize inviscid fluxes. TVD techniques allowing up to third order spatial accuracy are implemented as described by Yee *et al.* (1990). They are used with arithmetic or Roe averaged states, following the procedures outlined by Shuen *et al.* (1990). An entropy fix must be applied to certain problems, especially flows around blunt bodies, in order to avoid non-physical solutions displaying the so-called carbuncle phenomenon. The viscous fluxes are implemented by central differences, reflecting their physical property.

Temporal discretization

Instead of directly integrating the system of equations in the time domain, for each physical time step a steady-state problem is solved in a pseudo time domain. For this, a new residual $\underline{R}^*$ is introduced according to

$$\underline{R}^* = V \frac{\partial}{\partial t} \underline{Q} + \underline{R}(\underline{Q}) \tag{12}$$

Hence, the problem

$$\underline{R}^*(\underline{Q}[t_i, t^*]) = \underline{0}, \quad t^* \to \infty \tag{13}$$

is solved for an intermediate, physical time level t_i, marching the solution forward in pseudo time t^*. Third order accuracy in time for non-constant time steps is obtained by the following formulation

$$\left.\frac{\partial Q}{\partial t}\right|^{n+1} = (c_1 + c_2 + c_3)\Delta Q^{n+1} + (c_2 + c_3)\Delta Q^n + c_3 \Delta Q^{n-1} + O(\Delta t^3) \tag{14}$$

with

$$c_1 = \frac{(\Delta t^{n+1} + \Delta t^n)(\Delta t^{n+1} + \Delta t^n + \Delta t^{n-1})}{\Delta t^{n+1}\Delta t^n(\Delta t^n + \Delta t^{n-1})} \quad c_2 = -\frac{\Delta t^{n+1}(\Delta t^{n+1} + \Delta t^n + \Delta t^{n-1})}{\Delta t^n \Delta t^{n-1}(\Delta t^{n+1} + \Delta t^n)}$$

$$c_3 = \frac{\Delta t^{n+1}(\Delta t^{n+1} + \Delta t^n)}{\Delta t^{n-1}(\Delta t^n + \Delta t^{n-1})(\Delta t^{n+1} + \Delta t^n + \Delta t^{n-1})}$$

For a physical time step the iteration scheme

$$\left[\frac{1}{\Delta t^*}\underline{I} + \left.\frac{\partial \underline{R}^*}{\partial \underline{Q}}\right|^m\right]\Delta \underline{Q}^{m+1} = -\underline{R}^*|^m \tag{15}$$

with m as the sub-iteration count, has to be applied until the residual in the pseudo time domain $\underline{R}^*$ vanishes.

LU-SGS scheme

In equation (15) the Jacobi matrices corresponding to the right-hand side $\underline{R}^*$ are included to the left-hand side by $\partial \underline{R}^*/\partial \underline{Q}$. For a mathematically complete left-hand side formulation, Jacobi matrices for the chemical source term, the inviscid fluxes, the viscous fluxes and the source term due to axisymmetry would have to be computed. In this work all Jacobi matrices except the axisymmetric one are included in the left-hand side. The components of these matrices are given by Shuen (1992)), also providing a Jacobi matrix for the thin-shear layer approximated viscous fluxes. The resulting system of equations is then solved by the LU-SGS scheme, as introduced by Yoon and Jameson (1988).

For a time-accurate integration a suitable time step must be found locally for each cell, and then the minimum local time step be used globally. Even for a steady-state calculation a finite time step is beneficial as the factorization error reduces the stability of the implicit method. The time step calculation requires the knowledge of the convective and diffusive spectral radii, Λ_c and Λ_d, based on the eigenvalues of the inviscid and viscous Jacobi matrices and the local cell metrics. Details of the calculation of the spectral radii and eigenvalues of the flux matrices can be found in Ess (2003) and Kunz and Lakshminarayana (1992). The local time step is then calculated according to

$$\Delta t = CFL\frac{1}{\Lambda_t} = CFL\frac{1}{\Lambda_c + \Lambda_d} \tag{16}$$

In the case of a fully implicit left-hand side and an integration in time employing sub-iterations, only the inviscid time step restrictions need to be applied.

With a suitable time step calculated the time-marching integration can proceed and its convergence progress is monitored by the calculation of the residual. Here, a reference state is required to normalize the change of conserved variables. The residual for a cell is then calculated by applying the L_2-norm to the change of the conserved variables that was divided by the reference state and the local time step. The division by the local time step does not apply for a pure Newton-type iteration where the time step tends to infinity.

$$\varepsilon_{\text{cell}} = \frac{1}{\Delta t}\left[\left(\frac{\Delta\rho}{\rho_\infty}\right)^2 + \left(\frac{\Delta(\rho u)}{\rho_\infty U_\infty}\right)^2 + \left(\frac{\Delta(\rho v)}{\rho_\infty U_\infty}\right)^2 + \left(\frac{\Delta(\rho E)}{\rho_\infty E_\infty}\right)^2 + \sum_{i=1}^{N_s-1}\left(\frac{\Delta(\rho Y_i)}{\rho_\infty}\right)^2\right]^{1/2} \tag{17}$$

The overall residual ε is then obtained by averaging the residual obtained for all cells;

$$\varepsilon = \frac{1}{N_{\text{cells}}}\sum_{N_{\text{cells}}}\varepsilon_{\text{cell}} \tag{18}$$

This method of calculating the residual was applied to the results of the iteration and those of the sub-iteration scheme.

Aspects of parallelization

The calculation of chemically reacting multi-species gas flows is computationally very expensive, even when considering flows in two spatial dimensions only. This is more pronounced when dealing with unsteady physical problems that require a time-accurate integration.

On the computational side current progress in increasing the speed of a processor is somewhat limited, but orders of magnitude in speed up can be achieved when using many processors in parallel. This does have great implications on the computer program used to calculate the solution for a physical problem, and there are many ways of parallelizing, which greatly depend on the computer architecture available. A very good overview of parallel computing for computational fluid dynamics is given by Roose and Van Driessche (1995).

The LU-SGS scheme used in this work solves the system of equations in two sweeps, where the new solution of the current cell depends on that of the previously calculated, new solutions of neighbour cells. This can be vectorized well, but not parallelized without non-sychronized code execution and massive communication requirements between the processes. This disadvantage inspired the approach to parallelize the flow solver based on a subdivision of the computational domain into parts of roughly equal size.

Each process completely solves the computational problem for the assigned subdomain and then communicates the results to neighbouring processes only, and the root process if required for saving the flow field data to disk. This yields acceptable communication overhead between the processes and allows almost linear scalability if the subdomains are of equal size and the subdomain boundaries of equal length. The set-up has a drawback, though. The communication between neighbouring subdomains is only carried out in between a sub-iteration or iteration and, therefore, introduces an explicit coupling of the subdomains while within the subdomain

an implicit coupling exists. In most cases, the convergence deteriorates if no sub-iterations are performed, depending on the number of processors used. Further, in the specific case of the numerical method developed here the factor c controlling diagonal dominance of the system of equations solved with the LU-SGS scheme can be chosen to be as low as 1.25 for a calculation without domain subdivision, once the initial stage of the integration in the time domain is passed. When the same calculation is repeated with a domain subdivision, values a lot lower than $c = 2.5$ were found to cause stability problems. In the theoretical limit, each cell of the domain could be defined as a subdomain and an implicit Euler scheme would reduce to an explicit Euler scheme.

For calculations that employ sub-iterations this problem is somewhat solved because the update of values from neighbouring subdomains is performed after each sub-iteration. Therefore, for a set of sub-iterations that produced one converged time-step, the coupling between the subdomains becomes implicit as well.

The code presented in this paper uses the message-passing interface libraries, short MPI, in order to parallelize the algorithm to run on distributed memory systems. Despite being written for distributed memory systems there is no problem in using such a code based on MPI on shared memory systems.

Automatic domain subdivision

In many cases of parallel computation where a subdivision of the computational domain is performed, it is manually predefined or defined to coincide with the block boundaries of a multi-block domain. This allows to specifically select the subdomain shape and to hopefully reduce the communication between subprocesses thanks to carefully selected subdomain boundaries. However, it becomes cumbersome to manually design subdomains for each of various cases with different numbers of processors available for a parallel computation. Also, if the subdomain shape must coincide with a block of a multi-block domain, the number of processors that must be used is fixed to at least the number of blocks, which is very inconvenient. If there are more processors than blocks, some processors are idle resulting in an inefficient load balancing. A solution to this problem is the creation of many more subdomains as there are processors. As outlined by Struckmeier and Pfreundt (1993), several subdomains can then be assigned to a single processor to balance the load. It does include extra effort with respect to administration and, more importantly, to communication in between a much higher number of subdomains than originally required.

Therefore, an automatic subdivision procedure has been developed, that generates a domain decomposition for an arbitrary number of processors and independent of the number of grid blocks or their shape.

The core procedure of subdomain creation starts with a single cell in the flow domain not used by any other subdomain. This cell is considered the new subdomain, that is expanded into alternating directions, step by step adding unused neighbouring cells in the selected direction and switching to the next direction. This process is repeated until no more unused neighbouring cells are available, because there are physical boundaries or neighbouring subdomains preventing a further expansion, or the target subdomain size is reached. The subdomain creation process is repeated until the target number of subdomains is reached. Depending on the geometrical constraints, the number of cells and the number of subdomains required, there will generally be

cells unused in the domain. These cells have to be assigned to existing subdomains in a sensible way that aims at a low load imbalance and a compact subdomain shape. The last stage of the domain subdivision includes a final optimization of the subdomain shape in order to, again, achieve a compact subdomain shape. This can be necessary in case previous parts of the domain subdivision produced some highly stretched fragments of a subdomain.

The basic idea behind this domain subdivision procedure is to create subdomains as close as possible to an ideal subdomain with square shape. As the surface introduces communication overhead to the neighbouring subdomains it is of vital importance for parallel performance to have a high ratio of subdomain volume over surface, or in the two-dimensional case subdomain area over boundary length. The individual shape and size of the subdomains is strongly influenced by the number of cells in the domain, the number of subdomains required, and, perhaps most importantly, the geometrical shape of the domain itself. Hence, the domain subdivisions presented with the validation cases are not purely based on rectangular subdomains. The fact that the subdomains differ from an ideal rectangular shape is very useful to demonstrate the ability of the flow solver to handle any given domain subdivision and maintain calculation efficiency and solution integrity. Further details of this procedure can be found in Ess (2003).

Quantities for assessment of parallel performance

The computational domain is assumed to contain N_{cells} cells, which must be distributed between N_{procs} processors. A certain number of cells $N_{\text{cells, subdomain}(i)}$ will be assigned to each subdomain, where in this work the number of subdomains always equals the number of processors. The target size in cells for such a subdomain is

$$N_{\text{cells, target}} = \text{integer}(N_{\text{cells}}/N_{\text{procs}} + 0.5) \tag{19}$$

where integer(x) is the mathematically rounded integer number of x. This can then be used to calculate the load imbalance LIB according to

$$\text{LIB} = \max \frac{(N_{\text{cells, subdomain}(i)} - N_{\text{cells, target}})}{N_{\text{cells, target}}} \tag{20}$$

Only subdomains with a size greater than the target subdomain size are of interest here, because only those effectively slow down the overall performance. Since the target cell number for a subdomain can only be an integer number, the number of cells in total resulting from the target cell number applied to each subdomain in general exceeds, at best equals, the number of cells within the domain. This allows to reduce the cell number of the root process slightly when setting cell numbers of other subdomains to the target number, so that the increased work load of the root process due to additional administrative overhead can be balanced with respect to the work load of the other processes.

Further, the time spent on computation and directly related communication is identified as t_c, the time spent on loading or saving data files or initializing the computation discarded. Only the computation and directly related communication time is of importance, because it represents by far the majority of time spent on the solution of a flow problem and it is the time needed to assess parallel efficiency. It is important

to note that t_c here is not meant to measure the accumulated time spent on all processors, but the effective, or wall clock time, spent on the computation. Hence, for a parallel computation the time needed to solve a problem should ideally be the time spent on a single-processor computation divided by the number of processors used. In a real situation the overhead of the parallel code implementation and the communication between processes will increase the time required. The parallel speedup PAS is calculated according to

$$\mathrm{PAS} = t_{c,1\mathrm{processor}}/t_{c,N\mathrm{procs}} \tag{21}$$

Directly derived is the parallel efficiency PEF, obtained according to

$$\mathrm{PEF} = \mathrm{PAS}/N_{\mathrm{procs}} \tag{22}$$

Both parallel speed up and parallel efficiency will be severely influenced by the load imbalance. Therefore, there is a great interest in keeping the load imbalance as low as possible.

However, the penalty of the current approach comes with the decay of convergence when compared with a single-processor calculation. In the general case a reduction of convergence must be expected when many processors are used for an implicit calculation. As the domain is divided into more subdomains, more explicit coupling between cells is introduced and the convergence of the implicit method will approach more that of an explicit method.

Subdomain interconnection

Based on a prescribed domain subdivision each of the subdomains need to be connected to their neighbours in order to propagate the fluxes through the domain. This is achieved by surrounding each subdomain with overlap cells that are updated with values from the neighbouring domain after each iteration or, if required, sub-iteration. A cell at the boundary of a subdomain needs two overlap cells representing neighbouring subdomain cells in each relevant direction as neighbours, as shown in Figure 1. They are used to properly calculate the TVD flux limiters and viscous fluxes for those boundary cells of a subdomain.

Because the overlap cells are updated only after each iteration or sub-iteration, the coupling between the subdomains is explicit, while all other cells within a subdomain are coupled implicitly.

During the parallel computation the updating of the overlap cells is the only major communication that occurs, apart from some minor communication needed to synchronize the overall computation with the root process. This synchronization implies global communication of iteration counts, residual of calculation and time-progress. However, since only subdomains neighbouring to each other need to update the corresponding cells and overlap cells, communication is very limited.

Validation

Flow over flat plate

An important function of the flow solver is the correct calculation of viscous effects. For this, the flow over a flat plate at fixed temperature is examined in order to compute the boundary layer and temperature profiles developing.

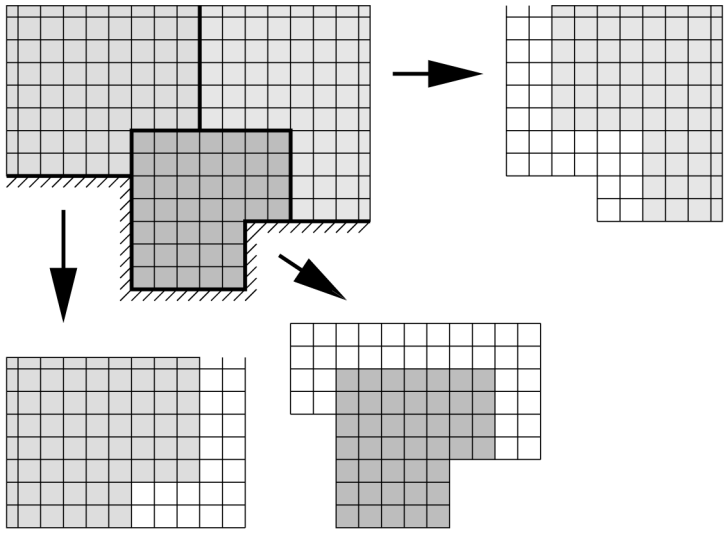


Figure 1.
Overlap cells around subdomains

The inflow Mach number, pressure and temperature are 2, 2 566 Pa and 221.6 K, respectively. These conditions are used in calculations performed by Lawrence *et al.* (1986), who compared their results to experimental data. The incoming air gas flow is simulated by a two-species gas mixture containing oxygen and nitrogen of molar fractions of $X_{O_2} = 0.235$ and $X_{N_2} = 0.765$. Further, the temperature along the flat plate of a length of 1 m is selected to equal the inflow temperature.

The set-up of the flow problem is shown in Figure 2 and the domain subdivision for the parallel computations with 2, 4, 8 and 16 processors in Figure 3. For the accurate

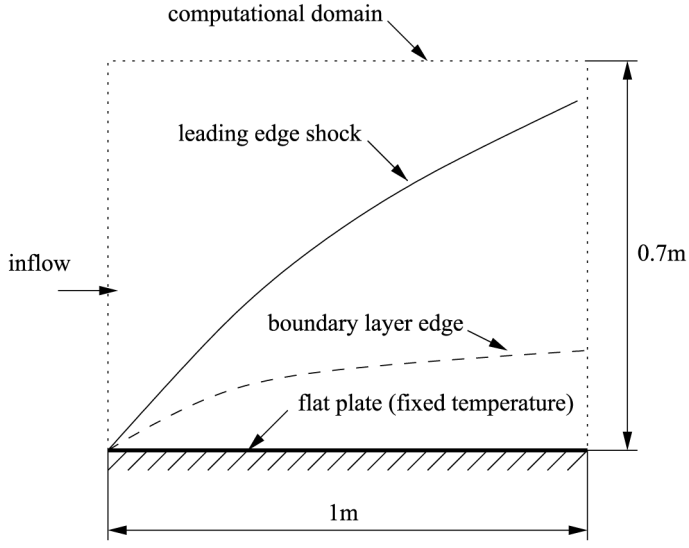


Figure 2.
Set-up for viscous boundary layer flow over flat plate

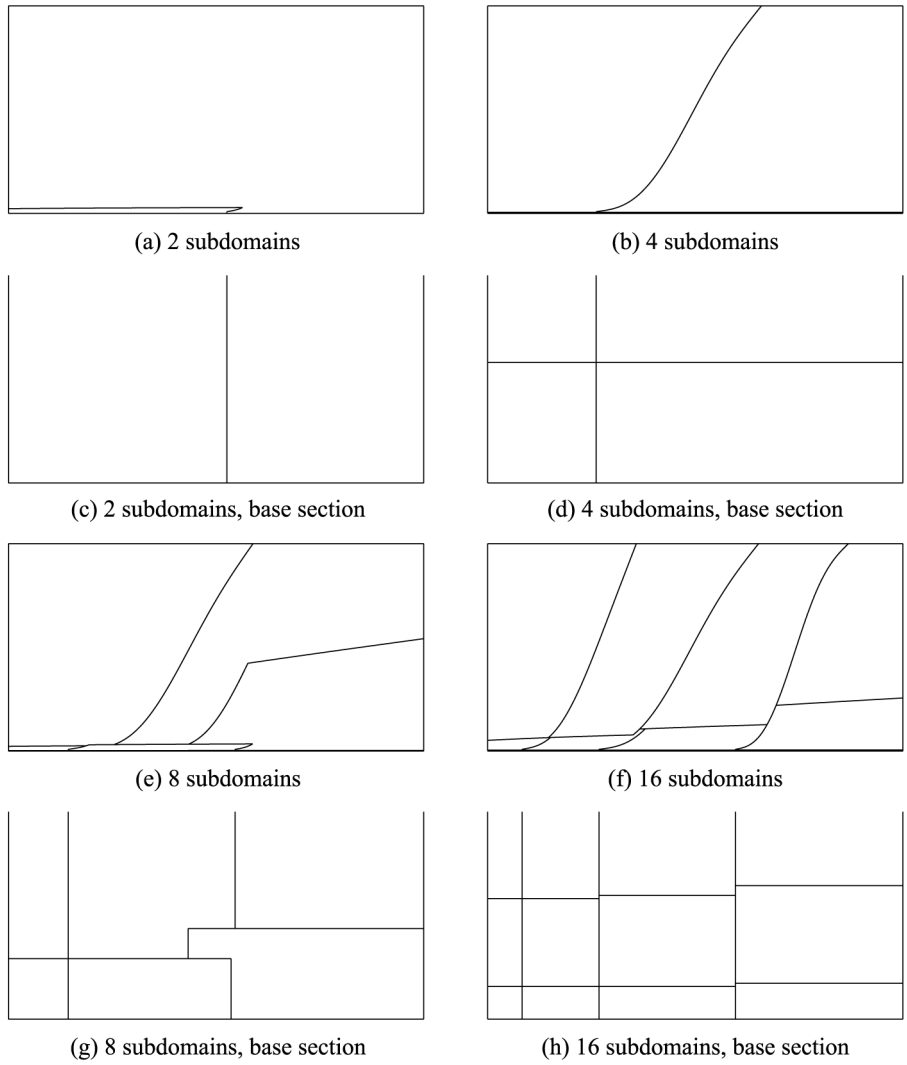
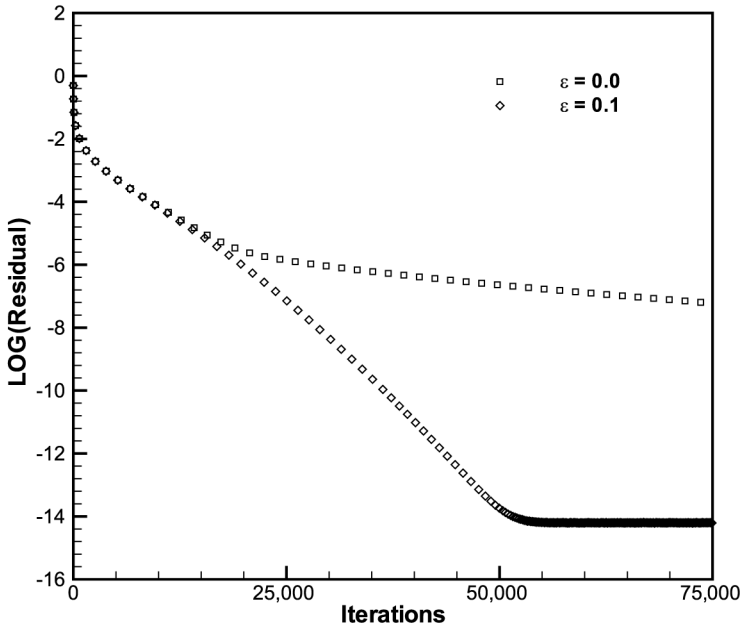


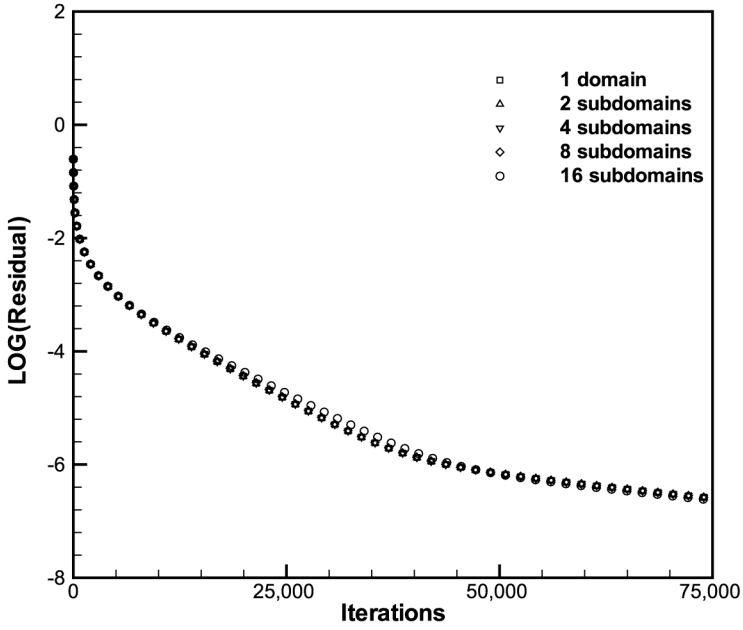
Figure 3.
Domain subdivision for flow over flat plate

computation of the flow it is important to capture the oblique shock originating from the leading edge of the flat plate within the computational domain, because only then the pressure profile along the boundary layer is correct. This yields the computational domain being split into a base block and an upper block, where the grid cells are clustered significantly more in the base block. Therefore, the domain subdivision is presented for the whole domain and the base block at the same time.

The calculations presented here investigate the influence of the entropy correction parameter and that of the domain subdivision and parallel computation. The convergence histories are shown in Figure 4. Much improved convergence can be observed for a value of $\varepsilon = 0.1$ used as entropy correction. However, since this type of



(a) single processor calculations



(b) parallel calculations

Figure 4.
Convergence histories for computations of flow over flat plate using different entropy correction coefficients and numbers of processors

problem does not need any entropy correction to deal with the carbuncle phenomenon and the solution is physically more correct using $\varepsilon = 0.0$, this value is used for the parallel computations. In this particular case the influence of the domain subdivision on the convergence is minor, and only the calculation with 16 processors slightly differs from the other calculations.

In the cases presented here a slow convergence can be observed despite the high *CFL* number of the order 10^5 used, which is caused by the application of the LU-SGS scheme to boundary layer flows in highly stretched grids. One aim of this study was to investigate the effect of the entropy correction on the solution of the boundary layer flow. In order to minimize the error due to spatial discretization a mesh more refined in the direction normal to the wall than necessary in a standard calculation was used. This leads to high aspect ratios in the cells along the wall, which, in combination with the flow present, is known to cause problems with LU-SGS type schemes (Wright *et al.*, 1996). An improved convergence can be obtained when each line of cells along the flat plate is solved in a coupled, line-implicit manner, instead of sweeps through the whole domain. Details of the error smoothing capabilities of different methods can be found in Wesseling (1992).

However, this is more expensive computationally and with respect to memory requirements, which degrades some of the advantages gained over the LU-SGS scheme. Further, for complex cases with solid walls and boundary layer flows in multiple directions, along the ξ and η coordinates in the computational domain, a line-implicit solution is difficult to implement, as the choice of coupling direction is not obvious. An even more significant problem emerges in the context of parallel computation with automated creation of subdomains, as presented in this paper, because only smaller segments along the flat plate could be solved in a fully coupled manner, depending on the domain subdivision present. As the additional cost of the LU-SGS scheme compared to an explicit one is moderate, it is, especially considering chemically reactive flows, still a suitable time-marching method, even for flows containing boundary or shear layers with associated high aspect ratio cells in the meshes used.

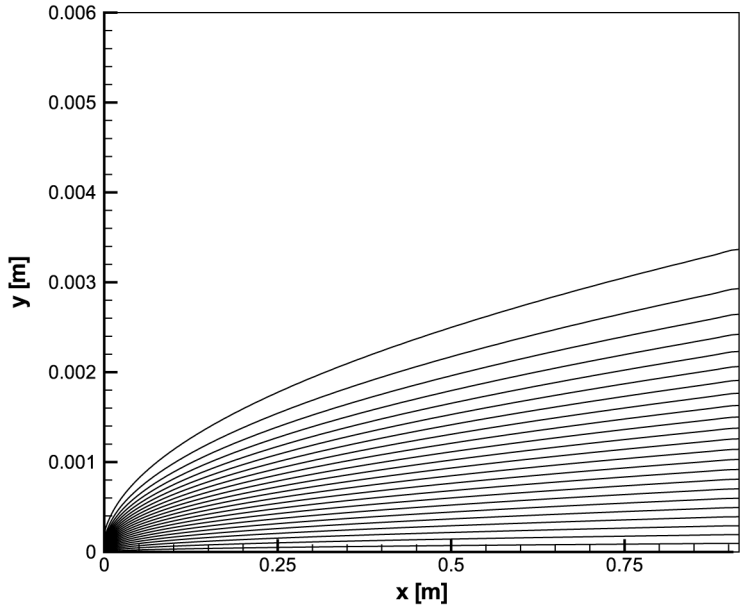
Horizontal velocity contours in the base part of the domain are shown in Figure 5, and pressure contours for the entire domain in Figure 6. The results for the parallel computations are identical and only one solution is shown. The parallel performance is given in Table I, where an overhead of around 8 per cent can be observed, which is approximately independent of the number of processors used. This nearly linear scalability is an excellent result.

The velocity and temperature profiles at a stream-wise position of $x = 0.915$ m are shown in Figure 7, where they are compared to numerical values obtained by Lawrence *et al.* (1986). The solutions differ very slightly for different entropy correction parameters.

However, the overall agreement is very good, despite the species properties in the present code being calculated by means of kinetic gas theory instead of Sutherland's law, which was applied by Lawrence *et al.* (1986). Also, the representation of air as a mixture of oxygen and nitrogen does not degrade the solution quality.

Diffusive mixing problem

A further important aspect of multi-species flows is the diffusive mixing between two gas jets of different species composition. In this numerical experiment a pure nitrogen



Note: One contour level represents a change of Δu = 25m/s. The aspect ratio is not maintained

Figure 5.
Horizontal velocity contours of flow over flat plate for base part of the domain

jet and a pure oxygen jet are entering the rectangular computational domain with a height of 0.001 m and a length of 1 m, nitrogen in the upper and oxygen in the lower part. Both gases enter the domain at the same inflow velocity of $u = 1,000\,\mathrm{m/s}$, pressure of $p = 1 \times 10^5$ Pa and temperature of 273.15 K. The set-up is shown in Figure 8 and the domain subdivisions resulting for 2, 4, 8 and 16 processors in Figure 9. The mesh is refined along the centerline of the domain and towards the inflow boundary. As for the flow over the flat plate, the influence of both entropy correction parameter and domain subdivision are examined.

The diffusion problem is treated analytically by Heiser and Pratt (1994), where the thickness δ_m of the mixing layer is approximated to

$$\delta_\mathrm{m} \approx 8\sqrt{\frac{D_{\mathrm{N_2}-\mathrm{O_2}}x}{u}} \tag{23}$$

The binary diffusion coefficient $D_{\mathrm{N_2}-\mathrm{O_2}} = 1.75 \times 10^{-5}\,\mathrm{m^2/s}$ used for the analytical solution is that obtained by means of kinetic gas theory (Hirschfelder *et al.*, 1954), and therefore in agreement with that used in the numerical method. The position downstream is x and the inflow velocity of the two jets is u. Further, the molar fraction of nitrogen can be calculated according to

$$X_{\mathrm{N_2}} = 0.5\left[1 + \mathrm{erf}\left(\frac{4y}{\delta_m}\right)\right] \text{ with } \mathrm{erf}(\chi) \equiv \frac{2}{\sqrt{\pi}}\int_0^{\chi} \mathrm{e}^{-t^2}\,\mathrm{d}t$$

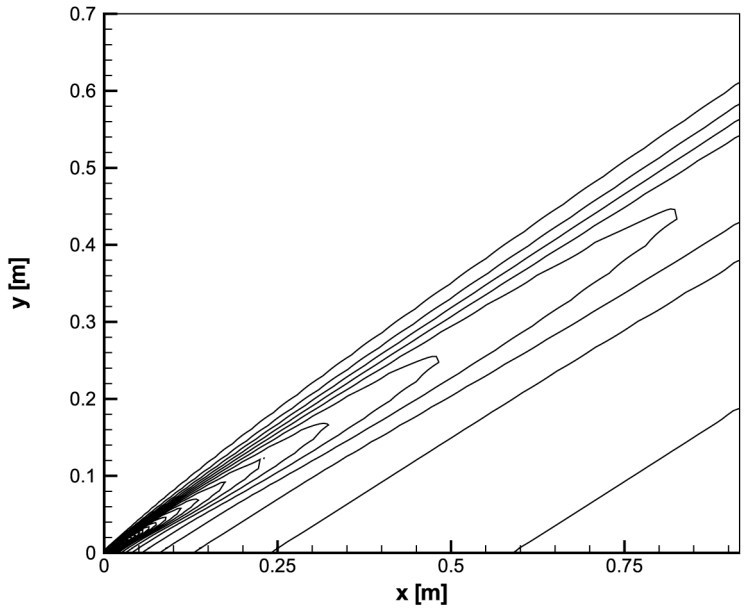


Figure 6.
Pressure contours of flow over flat plate

Note: One contour level represents a change of $\Delta p = 5Pa$. The aspect ratio is not maintained

N_{procs}	1	2	4	8	16
LIB	–	0.008	0	0.017	0.026
PEF	1	0.927	0.929	0.892	0.911

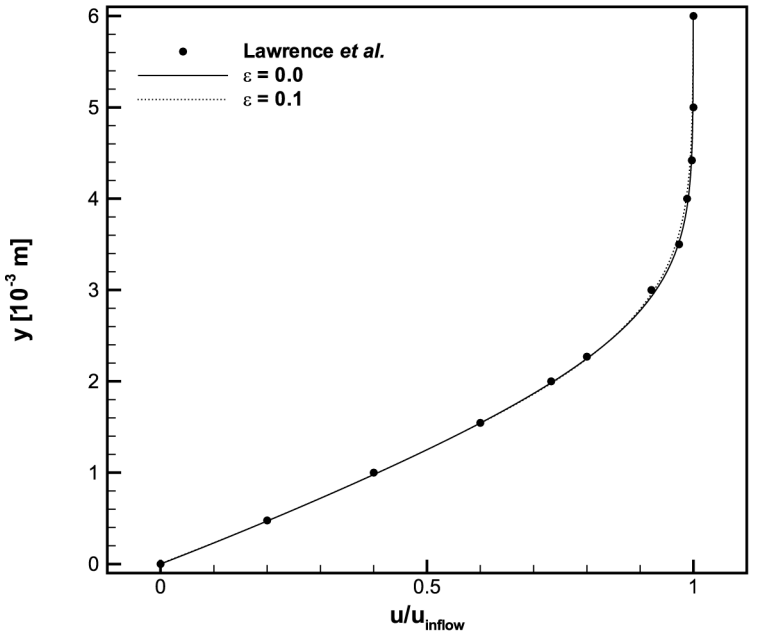
Table I.
Load imbalance, LIB, and parallel efficiency, PEF, for flow over flat plate

Note: Results were obtained on a parallel computer with Intel *P*3 processors running at 1 GHz, connected with a Myrinet network system. Intel and Myrinet are trademarks of the respective companies

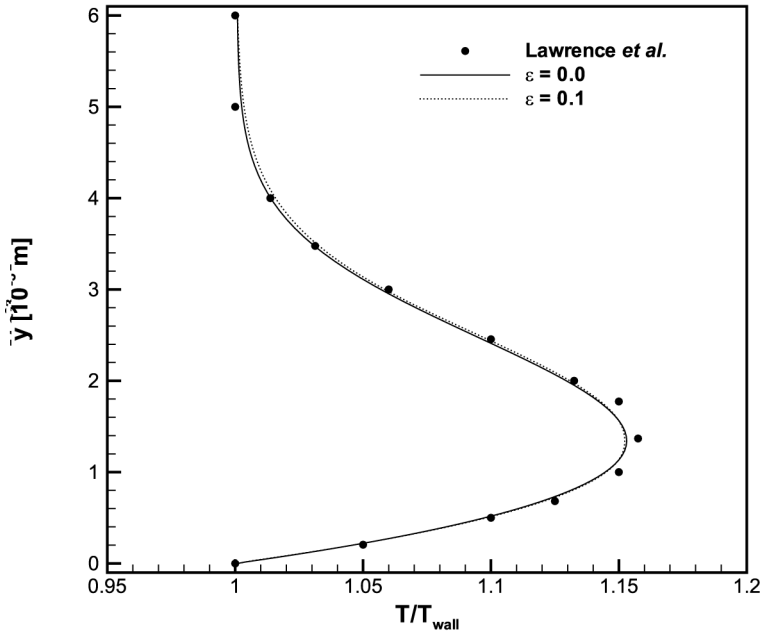
From known molar fractions the mass fractions can be calculated. It is apparent, that an initially sharp mass fraction profile is being smeared out by diffusive effects as the flow proceeds downstream.

Results obtained by the numerical method developed vary significantly, depending on the magnitude of entropy correction used. In order to demonstrate this, three calculations with entropy correction values of $\varepsilon = 0.0$, 0.5 and 1.0 are first performed with the full equations and subsequently with the physical diffusion switched off. The convergence histories for the calculations using *CFL* numbers of order 10^1 are shown in Figure 10, where a strong influence of the entropy correction parameter can be observed. With physical diffusion switched off the convergence is worse than in the case when the correct equations are applied.

The oxygen mass fractions obtained for a position 1 m downstream are shown in Figure 11, where cases with $\varepsilon = 0.5$ are omitted for clarity of the presentation.



(a) velocity profile



(b) temperature profile

Figure 7.
Results for viscous boundary layer flow over flat plate

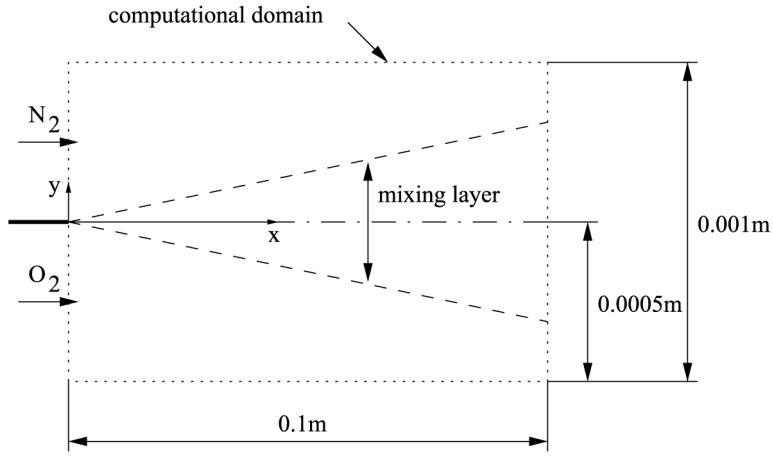


Figure 8.
Set-up for diffusive mixing problem

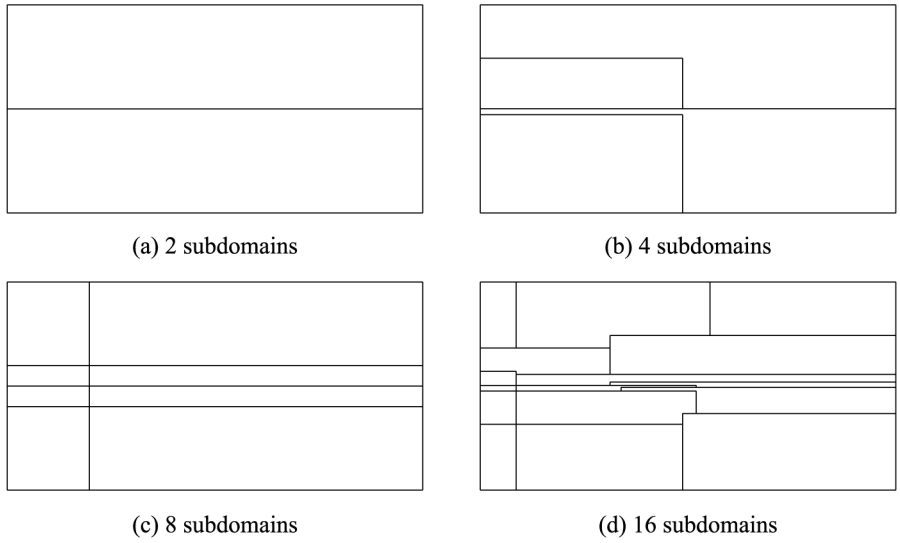
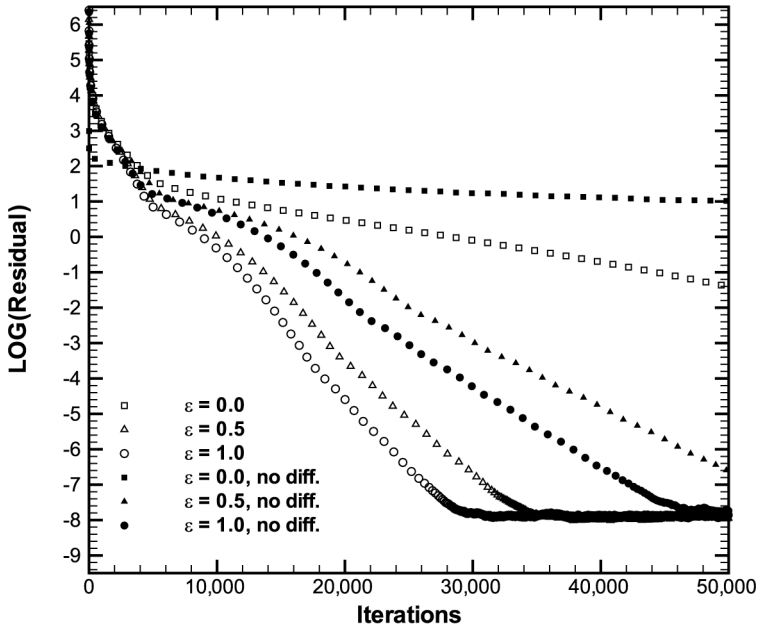


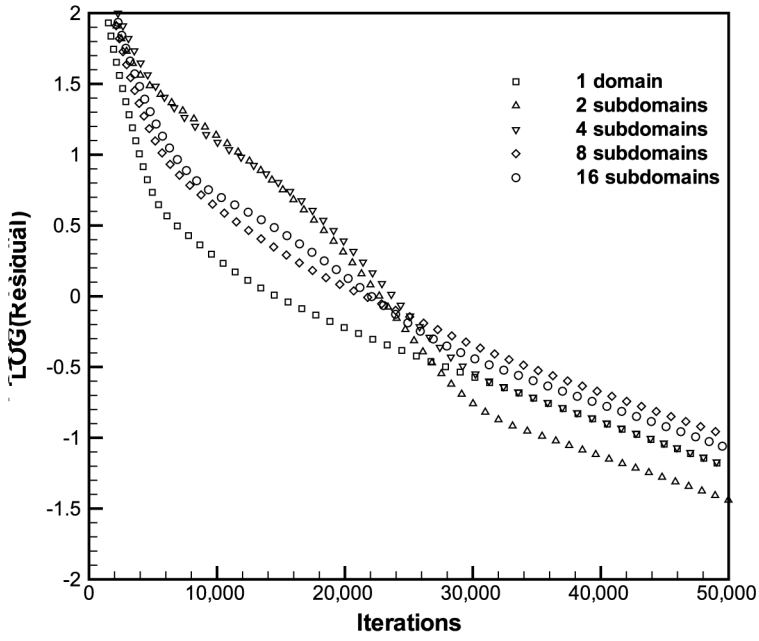
Figure 9.
Domain subdivision for diffusive mixing problem

Further, the profile of the oxygen mass fraction is smeared out more for higher entropy correction. While the slope of the oxygen mass fraction in the centre of the domain at $y = 0$ m is reproduced correctly when physical diffusion is included into the equations, it is much steeper otherwise.

The convergence for the parallel computations, with the entropy correction set equal to zero, is influenced by the domain subdivision. In this particular case the convergence is affected by the explicit coupling between the subdomains, because the signal propagation in a steady-state calculation is different across subdomain boundaries. The fact that the parallel convergence for two subdomains shown in Figure 10 is better than that for a single processor calculation is an interesting result. It must be connected with the domain subdivision coinciding with the upper and lower block boundaries

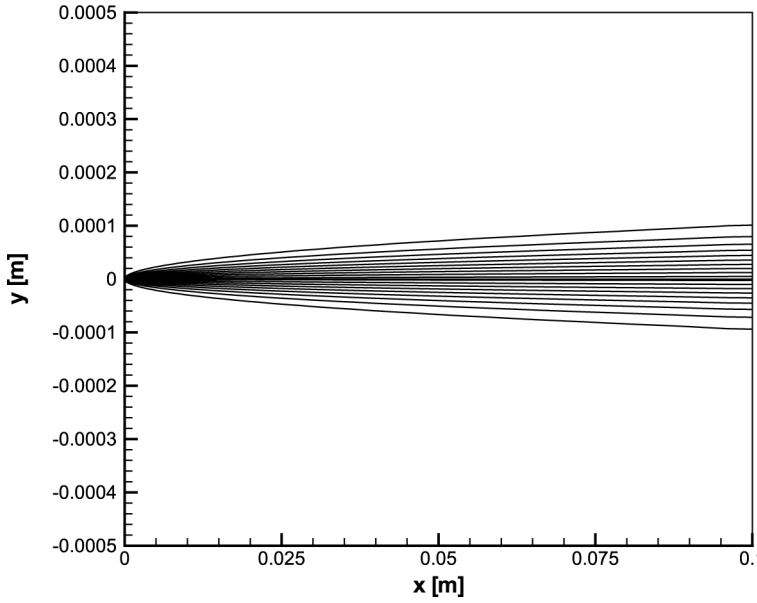


(a) single processor calculations

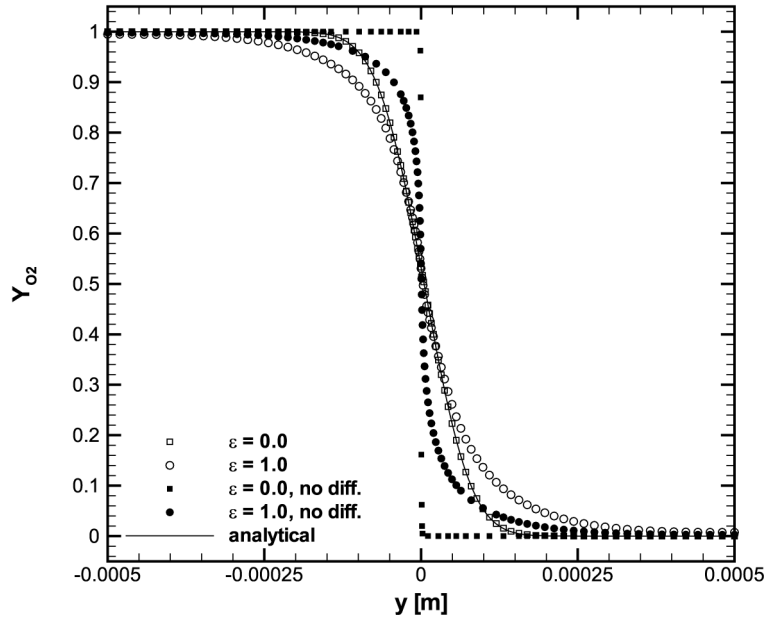


(b) parallel calculations

Figure 10.
Convergence histories for computation of diffusive mixing problem using different entropy correction coefficients and numbers of processors



(a) oxygen mass fraction contours



(b) oxygen mass fraction at exit plane of domain

Note: One contour level is equivalent to a change of mass fraction of ΔY $(O_2) = 0.05$. The aspect ratio is not maintained

Figure 11.
Results for diffusive mixing problem with different entropy correction and physical diffusion switched on and off

and the solution progress due to the sweeps performed when solving the resulting system of equations with the LU-SGS scheme. It highlights the influence of the coupling achieved by the solution method on the convergence for specific flow problems discussed in the previous subsection.

The resulting solutions are identical and therefore only one solution is shown for the oxygen mass fraction contours in Figure 11. The parallel performance is shown in Table II, where an overhead of around 5 per cent was found for the parallel computation. Only with the significantly increased load imbalance of the case with 16 processors the overall efficiency decreases.

Ideally the entropy correction should not be used in order to model the physically correct species diffusion. In that particular case the physical solution is reproduced numerically very well. However, stability considerations require a certain amount of entropy correction especially for hypersonic blunt body flows. Yee *et al.* (1990) suggested values of entropy correction vary from $\varepsilon = 0$ for flows with simple unsteady shocks up to 0.25 for blunt body flows. Choi *et al.* (2000a) use an even higher value of $\varepsilon = 0.4$ to ensure solution stability. In work by Gaitonde (1992) values up to $\varepsilon = 0.8$ were used to obtain correct solutions for supersonic flows around cylinders, depending on the mesh used.

Oscillating shock-induced combustion

When a projectile is fired into a combustible gas mixture steady shock-induced combustion, oscillating shock-induced combustion or detonation may occur, depending on the gas state and flight Mach number. Lehr (1972) did extensive experiments to examine these phenomena and its findings are widely used for the validation of numerical methods for reactive flow computations.

Therefore, the present code is used to reproduce the flow field for a projectile flying into the gas mixture at Mach numbers of 4.08, 4.48 and 6.46. The initial pressure and temperature are 43,383 Pa and 292 K, air is mixed with hydrogen stoichiometrically.

As pointed out by Choi *et al.* (2000*a*) viscous effects are of minor importance in this experiment. Hence, the inviscid, axisymmetric equations for a reactive multi-species gas are used. The gas is represented by nine species, with nitrogen assumed inert, and the reaction mechanism by 19 elementary reaction equations, as suggested by Choi *et al.* (2000a). The numerical set-up is chosen to equal the baseline method presented by Choi *et al.* (2000a), and the calculations performed using a mesh with 300 cells in direction normal and 200 cells tangential to the projectile surface.

The computational mesh and the domain subdivisions for 8, 16 and 32 processors are shown in Figure 12. The resulting Mach number contours calculated on the same mesh, with 300×200 cells, for the different projectile velocities are shown in Figure 13(a), (b) and (d). For the projectile Mach number 4.48 the oscillating

N_{procs}	1	2	4	8	16
LIB	–	0	0.032	0	0.080
PEF	1	0.952	0.947	0.978	0.865

Note: Results were obtained on a parallel computer with Intel *P*3 processors running at 1 GHz, connected with a Myrinet network system. Intel and Myrinet are trademarks of the respective companies

Table II.
Load imbalance, LIB, and parallel efficiency, PEF, for diffusive mixing problem

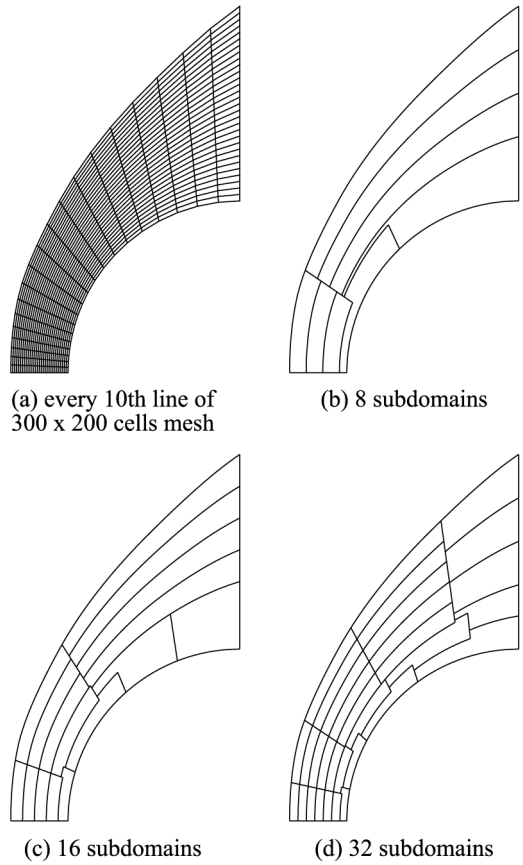


Figure 12.
Mesh and subdivision for calculation of shock-induced combustion around projectile

shock-induced combustion is already fully developed. In order to demonstrate the sensitivity of the physical problem to spatial resolution of the mesh the case with the projectile Mach number 6.46 was calculated on a mesh with half the cell number in both geometrical directions. On the coarse mesh the separation of the detonation wave into shock wave and flame front is not predicted at all, as shown in Figure 13(c).

The stagnation point temperature histories of the calculations with different projectile velocities are shown in Figure 14. In the case of the flight Mach number 4.48 the resolution of the oscillations greatly depend not only on the spatial resolution of the mesh but also on the accuracy of the time-marching integration. A *CFL* number of 1 was used to calculate the time-step for the third order accurate stencil representing the temporal discretization. The diagonal dominance factor $c = 2.5$ and an error tolerance of equal or less than 10^{-8} was applied for the sub-iterations. The oscillation frequency obtained from calculations with the present code is 427 kHz, which compares well with the 425 kHz measured in the experiment and also reproduced numerically by Choi *et al.* (2000*a*) and results in a relative error in frequency below 0.5 per cent. The history of the density along the stagnation streamline for a time between 70 and 80 μs is shown in

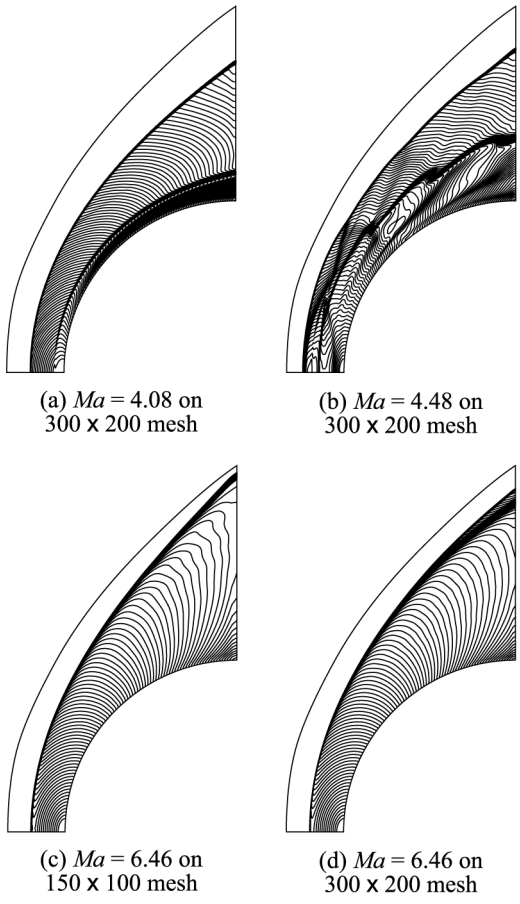
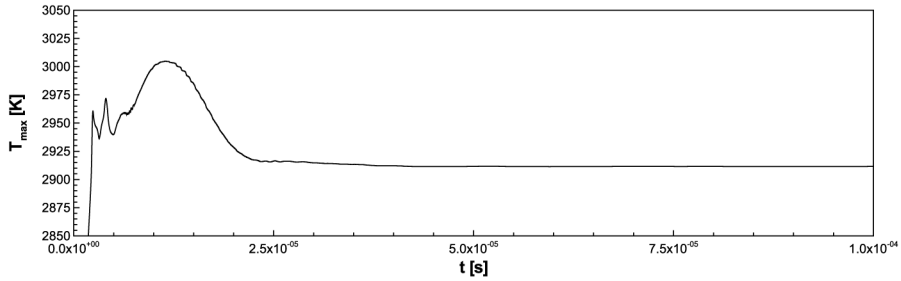


Figure 13.
Mach number contours for projectile at flow Mach numbers, Ma = 4.08, 4.48 and 6.46

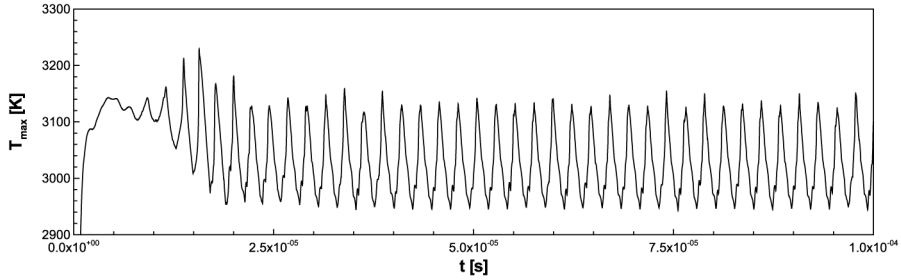
Figure 15. Calculations for this particular case were also performed with an accuracy of temporal discretisation reduced to two, and less strict sub-iteration convergence limit. This leads to the oscillations appearing at a later time and the oscillation frequency reproduced less accurate.

The cases with a steady flow around the projectile, for Mach numbers 4.08 and 6.46, were also reproduced by steady-state calculations, resulting in the same flow field as the time-accurate calculations.

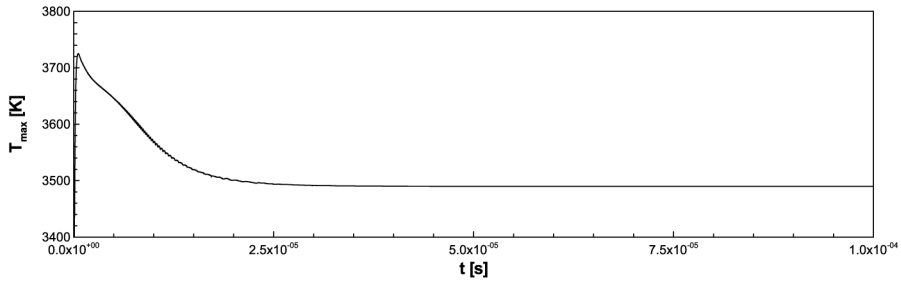
Parallel calculations of the flow around the projectile with a flight Mach number of 4.48 were performed in order to investigate the influence of the domain subdivision on the solution accuracy. As noted before, the explicit coupling of the subdomains is compensated by the sub-iterations, because they reinforce the spatially implicit coupling. This was confirmed by the parallel computations where no influence of the domain subdivision could be noticed. The parallel performance of the calculations is



(a) temperature for $Ma = 4.08$



(b) temperature for $Ma = 4.48$



(c) temperature for $Ma = 6.46$

Figure 14.
Stagnation point temperature over time for projectile with flow Mach numbers, Ma = 4.08, 4.48 and 6.46

shown in Table III, where parallel efficiency is directly related to load imbalance. Taking the load imbalance into account, the overhead associated with the parallel execution of the code is marginal.

Conclusions

A parallelized computational fluid dynamics code for the calculation of laminar chemically reactive and inert multi-species gas flows has been presented and validated with numerical, analytical and experimental results.

The validation has shown good reproduction of boundary layer flows, diffusive mixing and steady-state and time-accurate shock and detonation induced combustion.

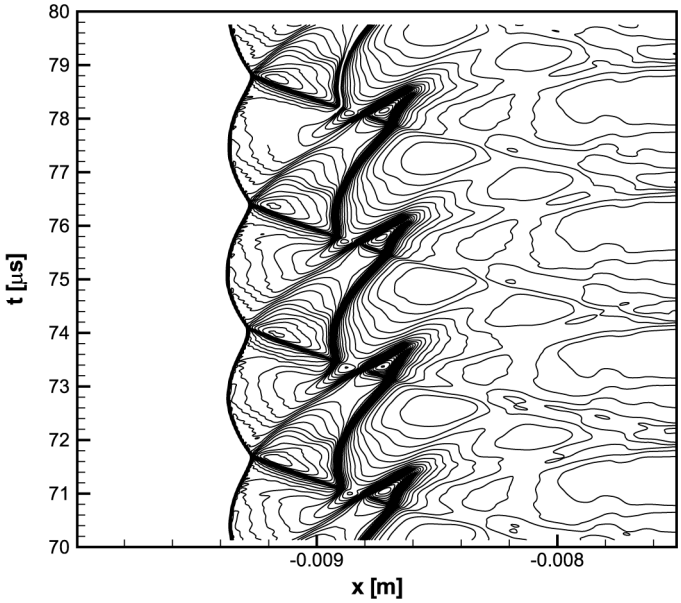


Figure 15.
Density history for stagnation point streamline for flow around projectile at flow Mach number, Ma = 4.48

Table III.
Parallel efficiency for calculations of oscillating shock-induced combustion around projectile flying into stoichiometric hydrogen-air mixture at a flight Mach number, Ma = 4.48

N_{procs}	1	8	16	32
LIB	–	0.0095	0.0192	0.1424
PEF	1	0.9794	0.9542	0.8673

Note: Results were obtained on a parallel computer with Intel *P3* processors running at 1 GHz, connected with a Myrinet network system. Intel and Myrinet are trademarks of the respective companies

Apart from some overhead when compared with a single-processor calculation, the parallel computation achieved an excellent parallel performance, that was only reduced in the case of a domain subdivision with load imbalance. This implies that for the present code the additional communication required for a higher number of processors is marginal and does not significantly degrade the parallel performance. With the numbers of processors used, subdomain sizes small enough than the communication overhead would become significant were not reached.

The present code represents a validated, efficient and versatile tool developed for the simulation of hypersonic multi-species flows in chemical non-equilibrium. Future work considered includes the implementation of turbulence models and an extension to three spatial dimensions.

The computational cost of three-dimensional, turbulent and chemically reactive flows with complex chemical reaction schemes becomes much higher as that of comparable two-dimensional laminar flows. This problem becomes even more pronounced considering the time-accurate solution of unsteady physical cases and, therefore, a parallelization can currently be considered the only practical way to solve

them. As an LU-SGS scheme comparable to that used in this paper for parallel computation has been introduced for computation in three spatial dimensions by Rieger and Jameson (1988), the method developed and presented in this paper appears suitable for the further extension suggested. This is true for both time and memory requirements of the calculation, as the domain subdivision and calculation of the problem on a shared memory computer implies memory requirements for each processor proportionate to the number of processors used.

References

Choi, J-Y., Jeung, I-S. and Yoon, Y. (1998), "Numerical study of scram accelerator starting characteristics", *AIAA Journal*, Vol. 36 No. 6, pp. 1029-38.

Choi, J-Y., Jeung, I-S. and Yoon, Y. (1999), "Unsteady-state simulation of model ram accelerator in expansion tube", *AIAA Journal*, Vol. 37 No. 5, pp. 537-43.

Choi, J-Y., Jeung, I-S. and Yoon, Y. (2000a), "Computational fluid dynamics algorithms for unsteady shock-induced combustion, part 1: validation", *AIAA Journal*, Vol. 38 No. 7, pp. 1179-95.

Choi, J-Y., Jeung, I-S. and Yoon, Y. (2000b), "Computational fluid dynamics algorithms for unsteady shock-induced combustion, part 2: comparison", *AIAA Journal*, Vol. 38 No. 7, pp. 1179-95.

Dolling, D.S. (1992), "Problems in the validation of CFD codes through comparison with experiment", *AGARD Conference Proceedings 514, "Theoretical and Experimental Methods in Hypersonic Flows"*.

Edwards, T.A. (1992), "CFD analysis of hypersonic, chemically reacting flow fields", *AGARD Conference Proceedings 514, "Theoretical and Experimental Methods in Hypersonic Flows"*.

Ess, P.R. (2003), "Numerical simulation of blunt-body generated detonation waves in viscous hypersonic ducted flows", PhD thesis, Department of Aerospace Engineering, University of Bristol, Bristol.

Gaitonde, D. (1992), "High-speed viscous flows past blunt bodies and compression corners with flux-split methods", Technical Report AD-A253, 413WL-TR-92-3018, Flight Dynamics Directorate, Wright Laboratory, Air Force Systems Command, Wright-Patterson Air Force Base, Ohio, 45433-6553.

Gardiner, W.C., Jr (1984), *Combustion Chemistry*, Springer, Berlin.

Gressier, J. and Moschetta, J-M. (2000), "Robustness versus accuracy in shock-wave computations", *International Journal for Numerical Methods in Fluids*, Vol. 33, pp. 313-32.

Heiser, W.H. and Pratt, D.T. (1994), *Hypersonic Airbreathing Propulsion*, Education Series, AIAA, Washington, DC.

Hirsch, C. (1988), *Numerical Computation of Internal and External Flows*, Vol. 1-2, Wiley, Chichester, New York, NY, Brisbane, Toronto and Singapore.

Hirschfelder, J.O., Curtiss, C.F. and Bird, R.B. (1954), *Molecular Theory of Gases and Liquids*, Wiley, New York, NY.

Jachimowski, C.J. (1988), "An analytical study of the hydrogen-air reaction mechanism with application to scramjet combustion", Technical Report NASA Technical Paper 2791, NASA, Langley Research Center, Hampton,VA.

Jameson, A. and Turkel, E. (1981), "Implicit schemes and LU decompositions", *Mathematics of Computation*, Vol. 37 No. 156, pp. 385-97.

Kunz, R.F. and Lakshminarayana, B. (1992), "Stability of explicit Navier-Stokes procedures using k-ε and k-ε/algebraic Reynolds stress turbulence models", *Journal of Computational Physics*, Vol. 103, pp. 141-59.

Lawrence, S.L., Tannehill, J.C. and Chaussee, D.S. (1986), "An upwind algorithm for the parabolized Navier-Stokes equations", paper presented at the AIAA/ASME 4th Fluid Mechanics, Plasma Dynamics and Lasers Conference, No. AIAA-86-1117.

Lehr, H.F. (1972), "Experiments on shock-induced combustion", *Astronautica Acta*, Vol. 17, pp. 589-97.

McBride, B.J., Gordon, S. and Reno, M.A. (1993), "Coeffcients for calculating thermodynamic and transport properties of individual species", Technical Report NASA Technical Memorandum 4513, NASA, Lewis Research Center, Cleveland, OH.

Marvin, J.G. (1992), "A CFD validation roadmap for hypersonic flows", *AGARD Conference Proceedings 514, "Theoretical and Experimental Methods in Hypersonic Flows"*.

Nusca, M.J. (2002), "Numerical simulation of the ram accelerator using a new chemical kinetics mechanism", *Journal of Propulsion and Power*, Vol. 18 No. 1, pp. 44-52.

Nusca, M.J. and Kruczynski, D.L. (1996), "Reacting flow simulation for a large-scale ram accelerator", *Journal of Propulsion and Power*, Vol. 12 No. 1, pp. 61-9.

Pandolfi, M. and D'Ambrosio, D. (2001), "Numerical instabilities in upwind methods: analysis and cures for the carbuncle phenomenon", *Journal of Computational Physics*, Vol. 166, pp. 271-301.

Rieger, H. and Jameson, A. (1988), "Solution of the three-dimensional compressible Euler and Navier-Stokes equations by an implicit LU scheme", paper presented at the 26th AIAA Aerospace Sciences Meeting, Reno, Nevada, 11-14 January, No. AIAA-1988-0619.

Roose, D. and van Driessche, R. (1995), "Parallel computers and parallel algorithms for cfd: an introduction", Chapter 1, *Special Course on Parallel Computing in CFD*, No. AGARD-R-807.

Shuen, J-S. (1992), "Upwind differencing and LU factorization for chemical nonequilibrium Navier-Stokes equations", *Journal of Computational Physics*, Vol. 99, pp. 233-50.

Shuen, J-S., Liou, M-S. and van Leer, B. (1990), "Inviscid flux-splitting algorithms for real gases with non-equilibrium chemistry", *Journal of Computational Physics*, Vol. 90, pp. 371-95.

Struckmeier, J. and Pfreundt, F.J. (1993), "On the effciency of simulation methods for the Boltzmann equation on parallel computers", *Parallel Computing*, Vol. 19, pp. 103-19.

Sweby, P.K. (1984), "High resolution schemes using flux limiters for hyperbolic conservation laws", *SIAM Journal of Numerical Analysis*, Vol. 21 No. 5, pp. 995-1011.

von Lavante, E., Zeitz, D. and Kallenberg, M. (2001), "Numerical simulation of supersonic air flow with transverse hydrogen injection", *Journal of Propulsion and Power*, Vol. 17 No. 6, pp. 1319-26.

Warnatz, J. and Maas, U. (1993), *Technische Verbrennung*, Springer, Berlin.

Wesseling, P. (1992), *An Introduction to Multigrid Methods*, Wiley, Chichester.

Williams, F.A. (1985), *Combustion Theory*, 2nd ed., Benjamin/Cummings, Menlo Park, CA.

Wilson, G.J. and MacCormack, R.W. (1992), "Modeling supersonic combustion using a fully implicit numerical method", *AIAA Journal*, Vol. 30 No. 4, pp. 1008-15.

Wright, M.J., Candler, G.V. and Prampolini, M. (1996), "Data-parallel lower-upper relaxation method for the Navier-Stokes equations", *AIAA Journal*, Vol. 34 No. 7, pp. 1371-7.

Yee, H.C. and Shinn, J.L. (1989), "Semi-implicit and fully implicit shock-capturing methods for nonequilibrium flows", *AIAA Journal*, Vol. 27 No. 3, pp. 299-307.

Yee, H.C., Klopfer, G.H. and Montagné, J.L. (1990), "High-resolution shock-capturing schemes for inviscid and viscous hypersonic flows", *Journal of Computational Physics*, Vol. 88, pp. 31-61.

Yoon, S. and Jameson, A. (1988), "Lower-upper symmetric Gauss-Seidel method for the Euler and Navier-Stokes equations", *AIAA Journal*, Vol. 26 No. 9, pp. 1025-6.

Yungster, S. (1992), "Numerical study of shock-wave/boundary-layer interactions in premixed combustible gases", *AIAA Journal*, Vol. 30 No. 10, pp. 2379-87.

Yungster, S. and Radhakrishnan, K. (2001), "Simulation of unsteady hypersonic combustion around projectiles in an expansion tube", *Shock Waves*, Vol. 11, pp. 167-77.

Yungster, S., Radhakrishnan, K. and Rabinowitz, M.J. (1998), "Reacting flow establishment in ram accelerators: a numerical study", *Journal of Propulsion and Power*, Vol. 14 No. 1, pp. 10-17.