Numerical flow modelling in a locally refined grid

INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN ENGINEERING, VOL. 37, 497–515 (1994)

NUMERICAL FLOW MODELLING IN A LOCALLY REFINED GRID

H. C. DE LANGE and L. P. H. DE GOEY

Faculty of Mechanical Engineering, WH.3-138, Eindhoven University of Technology, P.O. Box 513, 5600 MB Eindhoven, The Netherlands

SUMMARY

An algorithm is presented to model two-dimensional, non-isothermal, low Mach number flows with a local steep density gradient. The algorithm uses an adaptive, locally refined, non-staggered grid and has been developed, especially for modelling laminar flames. The governing equations, based on a stream-function–vorticity formulation, are presented and discretized using hybrid finite differences. A (isothermal) test problem is presented to compare the accuracy of the results of the solver presented in this paper, with the results of algorithms found in the literature. However, this test problem proves to be not well suited for the application of a locally refined grid, since it does not contain a local steep gradient. For this reason an additional test problem is constructed that clearly shows the advantages of the locally refined grid as compared to a uniform grid with respect to both the calculation time as well as the number of grid nodes needed. Furthermore, a laminar premixed flame is modelled with simple chemistry to show that the algorithm, presented in this paper, converges to a stabilized flame when an adaptive grid technique is used.

1. INTRODUCTION

In recent years, the construction of burners for industrial and domestic use has drawn increasing attention. Where simple design rules used to be sufficient, the energy scarcity and the growing awareness of environmental hazards make more sophisticated design tools indispensable. One of these new tools could be a computer model of the combustion process, which makes it possible to predict the consequences of changes in the burner and to find the conditions which ensure an optimal efficiency at low exhaust of pollutants. However, the physical and chemical features of flames are extremely difficult to describe, although there have been great achievements in combustion science in the last decades. In engineering, it is still almost impossible to construct burners supported by computer modelling. This is caused by the difficulty of separating the effects of the geometry of the burner from both the physical and chemical properties of the flame. Flames in domestic and industrial burners are, therefore, still very difficult to model and their characteristics are, for the time being, only moderately understood.

An important problem, which is inevitably encountered in the numerical modelling of two- or three-dimensional combustion processes, is brought about by the large difference between the scale of the burner (usual dimensions of the order of 0·1 m) and the flame thickness (of the order of 0·1 mm). In a non-refined (uniform) grid, an accurate solution would need a huge number of grid points. Solving the combustion equations on such a grid is only possible at the cost of an enormous number of calculations. Therefore, one would need an extremely fast computer with immense storage. This problem can be solved by the use of a locally refined grid, which calls for the development of a special solver.

0029–5981/94/030497–19$14.50

Received 7 April 1992
Revised 24 August 1993

The grid-refinement technique,[1] chosen in this paper, adds grid points to a coarse grid in areas where a given property has steep gradients. A non-staggered grid is used, although this type of grid is notorious for checkerboard oscillations in the flow field.[2] This problem is circumvented by solving the flow field using a stream-function–vorticity formulation.

The numerical algorithm, presented in this paper and shown schematically in Figure 1, has been developed, especially for the modelling of two-dimensional laminar flames in a Cartesian co-ordinate system. The governing conservation equations are solved iteratively, starting from some initial solution, and the program stops as soon as a stationary solution is found. Each iteration starts with the calculation of the density, the physical parameters (e.g. the thermal conductivity and viscosity) and the reaction rate. Next, the flow field is updated to ensure that the solution satisfies the continuity equation at each iteration. This is done by first recalculating the stream function. The new solution of the stream function leads to new velocity components. Finally, the scalar variables (temperature, mass fractions and vorticity) are solved. The locally refined grid is adapted when needed.

The equations describing the combustion process are presented in Section 2. In Section 3.1, the discretization of the conservation equations for vorticity, energy and mass fractions is presented. The discretization of the stream function and flow-field equations is described in Section 3.2. The method used to calculate the flow field at the boundaries of the calculation domain and at the boundaries of different refinement levels is presented in Sections 4 and 5, respectively. There are numerous methods to solve the resulting set of linearized equations. In Section 6, some of these methods are briefly presented and compared. In Section 7.1, a test problem is presented to show the accuracy of the proposed algorithm. An additional example, presented in Section 7.2, clearly shows the advantages of a locally refined grid. At the end of this paper (Section 8), a flame is modelled using a one-step chemical model. This last example shows that the algorithm is able to find a converged result, using adaptive gridding. Furthermore, the calculation time needed to obtain this solution is about 1 h on a workstation, which shows that modelling of flames is indeed feasible for engineering purposes.

2. FLOW-FIELD CALCULATION

The equations, which govern a stationary deflagration process, are the conservation equations of mass, momentum, energy and species mass fractions, together with the equation of state of the system.[3-5] The energy and species mass fraction equations are of the same form:

$$\nabla\cdot(\rho\mathbf{v}f) - \nabla\cdot(\Gamma_f\nabla f) = S_f \tag{1}$$

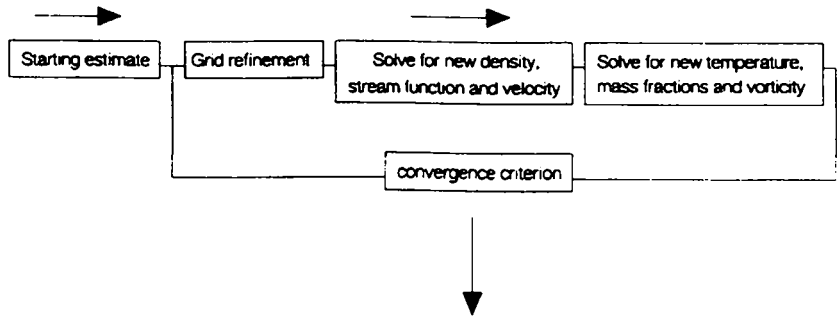

Figure 1. Block structure of the program

where ρ is the mixture density, $\mathbf{v}$ is the velocity (u, v), f is either the enthalpy or the mass fraction ($Y_i = \rho_i/\rho; i = 1,.., N$) of one of the N species, and Γ_f is introduced as the thermal conductivity or the mass diffusion coefficient. The right-hand side (S_f) of equation (1) represents the chemical source term, which will be introduced in Section 8 for the case of a one-step chemical reaction.

Summing equation (1) over all species leads to the well-known continuity equation

$$\nabla \cdot (\rho \mathbf{v}) = 0 \tag{2}$$

since the diffusion fluxes and chemical source terms add to zero by definition.

The momentum-balance equation of the mixture is written as

$$\rho \mathbf{v} \cdot \nabla \mathbf{v} + \nabla \cdot \mathbf{M} = -\nabla P, \tag{3}$$

which has the familiar form of a single-fluid momentum equation. In equation (3), $P = \sum_i P_i$ is the sum of the partial pressures and $\mathbf{M}$ indicates the viscous stress tensor.[3]

In two-dimensional (2-D) combustion modelling, equations (1–3) are $(N + 3)$-independent coupled conservation equations, while we have $N + 4$ independent variables: $Y_i (1 \leqslant i \leqslant N - 1)$, P, $\mathbf{v}$, ρ and T. This set of equations is of common use in combustion research.[6-9] The equation of state completes the set of equations. We use the ideal gas law:

$$P = \rho R T \tag{4}$$

where R is the specific gas constant of the mixture. In fuel/air combustion, the abundant quantity of nitrogen makes it possible to assume that R is independent of the mixture composition. It may be concluded from equation (3) that all pressure variations in the flame are much smaller than the atmospheric pressure P_{atm} in low Mach number deflagration processes. Therefore, equation (4) implies that the density may be regarded as a function of the temperature only: $\rho = P_{atm}/(RT)$. The effect of pressure variations is eliminated from the 2-D flow equations by the introduction of a stream function and a vorticity,[1] although a stream-function–vorticity formulation is rarely used in problems with a non-constant density. The vorticity is defined as

$$\boldsymbol{\omega} \equiv \nabla \times \mathbf{v} \tag{5}$$

In 2-D systems, only the z-component of $\boldsymbol{\omega}$ is needed, which will be denoted as ω from here on. Within the (Cartesian) calculation domain, ω has to satisfy the vorticity equation, which is found from the momentum equations:

$$\nabla \cdot (\rho \mathbf{v} \omega) + \nabla \times (\nabla \cdot \mathbf{M}) = \frac{\partial \rho}{\partial x}\frac{\partial \zeta}{\partial y} - \frac{\partial \rho}{\partial y}\frac{\partial \zeta}{\partial x} \tag{6}$$

with $\zeta = (u^2 + v^2)/2$. Substituting the definition of $\mathbf{M}$ gives

$$\begin{aligned} \nabla \times (\nabla \cdot \mathbf{M}) = & -\nabla \cdot (\mu \nabla \omega) - 2\frac{\partial^2 \mu}{\partial x \partial y}\left(\frac{\partial u}{\partial x} - \frac{\partial v}{\partial y}\right) + \frac{\partial \mu}{\partial y}\left(\frac{\partial \omega}{\partial y} - 2\nabla^2 u\right) \\ & + \frac{\partial \mu}{\partial x}\left(\frac{\partial \omega}{\partial x} + 2\nabla^2 v\right) + \left\{\frac{\partial^2 \mu}{\partial x^2} - \frac{\partial^2 \mu}{\partial y^2}\right\}\left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right) \end{aligned} \tag{7}$$

where μ is the dynamic viscosity. Note that equations (6) and (7) can be rewritten as a convection-diffusion equation like equation (1), when the right-hand sides of both equations (except for the $-\nabla \cdot (\mu \nabla \omega)$ term) are combined into a source term. This will be used in Section 3.1. The stream function (φ) is defined in such a manner that it describes a 2-D flow field, which always satisfies

the continuity equation (equation (2))

$$\partial\varphi/\partial y \equiv \rho u \tag{8a}$$

$$-\partial\varphi/\partial x \equiv \rho v \tag{8b}$$

According to equation (5), the stream function is coupled to the vorticity by

$$\frac{\partial}{\partial x}\left(\frac{1}{\rho}\frac{\partial\varphi}{\partial x}\right)+\frac{\partial}{\partial y}\left(\frac{1}{\rho}\frac{\partial\varphi}{\partial y}\right)=\omega \tag{9}$$

Instead of having to calculate u, v and P from a combination of the momentum equations (3) and the continuity equation (2), we now have to solve equations (6)–(9) for u, v, φ and ω. This will be done by first updating the stream function φ from an estimate of the vorticity using equation (9). Subsequently, the velocity components are calculated from the newly found stream function according to equation (8). Finally, a new vorticity solution is calculated according to equations (6) and (7). This iterative procedure is repeated until a stationary solution is reached.

3. DISCRETIZATION

Here several methods are presented for discretizing the combustion equations. In Section 3.1 the discretization of the scalar convection–diffusion equations for T, Y_i and ω is described. The convection–diffusion equations, which describe the scalar fields not only of T and Y_i (equation (1)) but also of ω (equations (6) and (7)), will be discretized using fairly standard finite difference methods.[2] In Section 3.2 the discretization of the flow-field equations for u, v and φ is discussed. These equations are discretized in two ways: one of these is of low order and fairly standard in modelling of incompressible flow systems and the other is of higher order and new.

3.1. The scalar equations

Stability problems occur at large local Peclet numbers (defined in equations (10a)–(10d) when convection–diffusion equations are discretized using the central-difference scheme. Hybird, power-law or upwind discretization[2] is, therefore, used for these equations. Note that it is possible to find stable solutions using central difference scheme, even for large Peclet numbers, as long as the relaxation parameters are chosen according to the CFL criterion.[10] However, the thus found, stable solution may contain non-physical oscillations. This is caused by the fact that a Taylor-series interpolation, as used in the central-difference scheme, is not sufficient for large Peclet numbers.

Discretization of the scalar equations around point P using the points N, E, S, W, n, e, s and w, as defined in Figure 2, results in

$$\psi_P f_P = \psi_N f_N + \psi_S f_S + \psi_E f_E + \psi_W f_W + S_P \tag{10}$$

where

$$\psi_N = \Gamma_n/(\Delta y_n \Delta y)\,\Psi(Pe_n), \qquad Pe_n = (\rho v)_n \Delta y_n/\Gamma_n \tag{10a}$$

$$\psi_S = \Gamma_s/(\Delta y_s \Delta y)\,\Psi(-Pe_s), \qquad Pe_s = (\rho v)_s \Delta y_s/\Gamma_s \tag{10b}$$

$$\psi_E = \Gamma_e/(\Delta x_e \Delta x)\,\Psi(Pe_e), \qquad Pe_e = (\rho u)_e \Delta x_e/\Gamma_e \tag{10c}$$

$$\psi_W = \Gamma_w/(\Delta x_w \Delta x)\,\Psi(-Pe_w), \qquad Pe_w = (\rho u)_w \Delta x_w/\Gamma_w \tag{10d}$$

$$\psi_P = \psi_N + \psi_S + \psi_E + \psi_W \tag{10e}$$

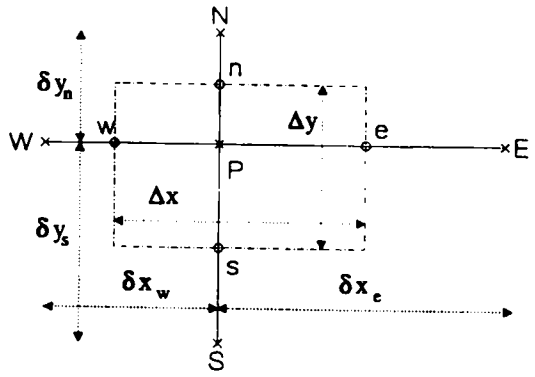

Figure. 2 The control volume

and

$$\Delta x = x_e - x_w \tag{10f}$$

$$\Delta y = y_n - y_s. \tag{10g}$$

The length scales Δx_e, Δx_w, Δy_n and Δy_s are defined in Figure 2. In this equation, S_P is the source term at point P, which accounts for the effect of the chemical reaction. The source term (S_P) in the vorticity equation (formed by the right-hand side terms of equations (6) and (7) as explained) will be calculated using central differences. The parameter values in the Peclet numbers (ρu, ρv and Γ) at the intermediate points (n, s, e, w) are calculated using harmonic interpolation.[2]

The function Ψ in equation (10) depends on the type of discretization. It is given by either

$$\Psi(Pe) = \max(0,\ 1{\cdot}0 - 0{\cdot}5\,|Pe|) + \max(0,\ -Pe) \quad \text{for the hybrid scheme} \tag{11a}$$

$$\Psi(Pe) = \max(0,\ (1{\cdot}0 - 0{\cdot}1\,|Pe|)^5) + \max(0,\ -Pe) \quad \text{for the power-law scheme} \tag{11b}$$

or

$$\Psi(Pe) = 1{\cdot}0 + \max(0,\ -Pe) \quad \text{for the upwind scheme} \tag{11c}$$

Other discretization schemes (e.g. central differences) are easily implemented by changing the expression for Ψ.

The results, obtained when using the hybrid and power-law schemes are more accurate compared to those calculated using the upwind scheme; the upwind scheme as used by Smooke *et al.*,[1] leads to artificial diffusive fluxes at all local Peclet numbers.[11,12] An acccurate approximation of the diffusive fluxes is essential in combustion modelling, especially near to the reaction zone. The power-law and hybrid scheme take care of this, as long as the grid near the reaction zone is so fine that the local Peclet numbers remain smaller than 2. Cross-wind diffusion may also cause a deviation, which may be omitted by the use of skewed schemes.[11,12] These skewed schemes lead to quite extensive calculations and have not been used.

The power-law scheme[2] uses an interpolation function close to the analytical one-dimensional solution of the convection–diffusion problem without source terms and with constant coefficients (ρu, ρv and Γ). Therefore, it yields an accurate solution in simple (sourceless) one-dimensional problems. The power-law scheme may be improved in multidimensional problems, by using the same interpolation function with a correction for 2-D fluxes.[13] However, this corrected scheme leads to a substantial increase in the number of calculations and has, therefore, not been used.

3.2. *The flow field*

The flow field will be calculated from the stream function according to equation (8). The stream function is calculated from equation (9), which is a combination of the definition of the vorticity (equation (5)) and the stream function (equation (8)). Therefore, the discretization of equation (9) should be in accordance with the discretization of equation (8). This means that equation (8) will be discretized first to find the velocity components from a known stream-function solution. Next, these discretized equations are used in a discretized form of the vorticity definition (equation (5)) to find an equation from which a new stream function may be calculated (note that this is not the sequence in which the equations will be used in the program). The discretization is presented for the case of an equidistant grid; the equations for non-equidistant grids may be derived analogously.

Central-difference discretization of equation (8) may be used to solve u_P and v_P:

$$\rho_P u_P = (\varphi_N - \varphi_S)/2\Delta y \tag{12a}$$

and

$$\rho_P v_P = -(\varphi_E - \varphi_W)/2\Delta x \tag{12b}$$

However, the second-order discretization error made by using equation (12) is proportional to the third spatial derivative of φ. These third derivatives are equal to the second spatial derivatives of ρu and ρv by definition. Therefore, fourth-order equations to calculate u_P and v_P are easily derived:

$$\rho_N u_N + 4\rho_P u_P + \rho_S u_S = 6(\varphi_N - \varphi_S)/2\Delta y \tag{13a}$$

and

$$\rho_E v_E + 4\rho_P v_P + \rho_W v_W = -6(\varphi_E - \varphi_W)/2\Delta x \tag{13b}$$

which are the integrated forms of equation (8), using Simpson's integration method.

As mentioned, the stream function will be calculated using the central-difference discretization of equation (5):

$$(u_n - u_s)/\Delta y - (v_e - v_w)/\Delta x = \omega_P \tag{14}$$

The use of equation (12) or equation (13) for u and v at the intermediate points gives the equation from which φ must be calculated. Using (12) gives

$$\left\{\frac{1}{\rho_n}(\varphi_N - \varphi_P) - \frac{1}{\rho_s}(\varphi_P - \varphi_S)\right\}\Big/\Delta y^2 + \left\{\frac{1}{\rho_e}(\varphi_E - \varphi_P) - \frac{1}{\rho_w}(\varphi_P - \varphi_W)\right\}\Big/\Delta x^2 = \omega_P \tag{15}$$

This result may also be found by straightforward second-order central-difference discretization of equation (9). However, the use of equation (13) in equation (14) gives a less trivial and more accurate equation

$$\left\{\frac{1}{\rho_e}[6(\varphi_E - \varphi_P)/\Delta x + \rho v_E + \rho v_P] - \frac{1}{\rho_w}[6(\varphi_P - \varphi_W)/\Delta x + \rho v_P + \rho v_W]\right\}\Big/4\Delta x$$
$$+ \left\{\frac{1}{\rho_n}[6(\varphi_N - \varphi_P)/\Delta y - \rho u_N - \rho u_P] - \frac{1}{\rho_s}[6(\varphi_P - \varphi_S)/\Delta y - \rho u_P - \rho u_S]\right\}\Big/4\Delta y = \omega_P \tag{16}$$

In both cases, the density at the intermediate points is calculated as the harmonic average of the surrounding values. As far as we know, the higher-order approximations given by equations (13) and (16) are new.

4. BOUNDARY CONDITIONS FOR φ AND ω

The boundary conditions of flow problems are usually formulated in terms of the velocity components. The stream-function–vorticity formulation calls for additional boundary conditions for ω and φ. Since the solutions of both variables are described by a second-order differential equation (equations (6) and (9)), they both have to be determined by means of boundary conditions on all boundaries. The values of φ and ω on the boundaries have to be determined from the flow field. The solutions following from these boundary conditions and the differential equations for φ and ω describe the flow field within the calculation domain. It is well-known[14] that these boundary conditions may be the cause of instability of the numerical algorithm. Therefore, some attention is drawn to this subject. We will distinguish three types of boundaries: walls, outflow boundaries and symmetry walls, which will be treated subsequently.

On walls, u, v and T are known and ρ is found from the equation of state $\rho = P_{\text{atm}}/RT$. The stream function (φ) may then be calculated by analytical or numerical integration of either ρu for walls along the y-axis or ρv for walls along the x-axis according to equation (8). In the problems presented in this paper, φ is evaluated analytically. The vorticity (ω) at the point P on the wall is calculated by a second-order approximation of its definition (equation (5)) in the fictitious point P″, half-way between the point P on the wall and the first interior point P′ next to the wall[14] (see Figure 3). In case of an equidistant grid (analogous equations are derived easily for non-equidistant grids), we use

$$\omega_P = -\omega_{P'} + (u_A - u_B)/(y_A - y_B) + (u_C - u_D)/(y_C - y_D) - 2(v_P - v_{P'})/(x_P - x_{P'}) \quad (17a)$$

or

$$\omega_P = -\omega_{P'} + 2(u_P - u_{P'})/(y_P - y_{P'}) - (v_A - v_B)/(x_A - x_B) - (v_C - v_D)/(x_C - x_D) \quad (17b)$$

for $x_A - x_B$ or $y_A - y_B$ equal to zero, respectively. The points A, B, C and D are defined as in Figure 3. Note that equation (17) implies that the values of ω at the wall vary as the flow field changes. The use of this boundary condition did not give rise to instability problems, in all cases tested.

Many flow problems (such as the second test problem in Section 7 and the flame modelled in Section 8) have an outflow boundary at which the flow field (and thus also the stream function) is not known beforehand. An outflow boundary parallel to the y-axis is treated here as an example. The mass-flow component perpendicular to the boundary (ρu) is calculated according to equation (13a), which assures mass conservation. The other (parallel) mass-flow component (ρv), as well as the temperature and mass fractions are calculated by assuming that their derivatives perpendicular to the boundary (e.g. $\partial(\rho v)/\partial x$) are equal to zero to first order ($(\rho v)_P = (\rho v)_{P'}$); u and v are determined subsequently, after evaluation of the density, again by using the equation of state and the temperature. It then becomes possible to calculate the vorticity (ω) with the help of equation (17). However, it is, of course, not possible to calculate the stream function by means of numerical integration of ρu, since ρu has been determined from φ. Instead, the stream function will be calculated by assuming that the first-order approximation to the second-order derivative ($\partial^2\varphi/\partial x^2$) is equal to zero. This approximation, known as Roache's condition,[15] is consistent with the assumption that the derivative of the parallel component ρv of the mass-flux vector is equal to zero: $\partial(\rho v)/\partial x = \partial^2\varphi/\partial x^2 = 0$. This condition provides very accurate results in incompressible flow computations, provided that the downstream boundary is positioned sufficiently far away from regions, where gradients in u, v and φ are small. The first-order approximation to both $\partial(\rho v)/\partial x = 0$ and $\partial^2\varphi/\partial x^2 = 0$ is responsible for an inconsistency in the solution at this boundary in terms of the truncation error. This error will be insignificant if the outflow boundary is positioned far away from regions of physical activity.

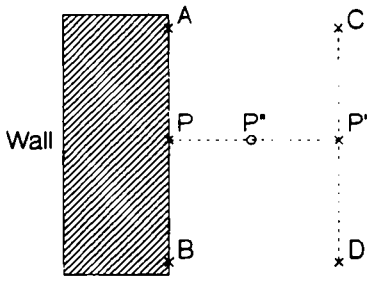

Figure 3. Definition of the boundary points used in equation (17), P" is a fictitious point halfway between the boundary point P and the interior point P'

The velocity component, perpendicular to a symmetry boundary, must be equal to zero. We treat a symmetry boundary along the x-axis as an example here, meaning that v equals zero. Therefore, integration of this velocity component gives a constant stream function along this boundary. Furthermore, the derivative perpendicular to the symmetry line of the velocity component parallel to the wall is equal to zero ($\partial u/\partial y = 0$ in our example). This means that both $\partial u/\partial y$ and $\partial v/\partial x$ are equal to zero and, therefore, the vorticity on a symmetry bound must be equal to zero. The parallel velocity component (u) is calculated using equation (13) by introducing a fictitious mirror point outside of the computational domain and (in our example) setting $(\rho u)_S - (\rho u)_N$ and $\varphi_N + \varphi_S - 2\varphi_P$ equal to zero (note that either N or S is P').

5. ADAPTIVE GRID

We use a grid refinement technique in which, starting from a coarse grid, additional points[1] are introduced in regions where steep gradients occur, half-way between grid points of the coarser grid level (see Figure 4). This introduction of additional points is performed in such a manner that it is possible to use the discretized equations, presented in Section 3.1, within the refined grid (the points marked with ×: interior points). Furthermore, there is always at most one intermediate point (marked with ○) between two interior points or between an interior point and a wall point. An interpolation strategy is developed for these intermediate points, which preserves the accuracy of the solution. This interpolation procedure passes the solution of the coarse grid on to the refined grid. To pass the solution on in the other direction (from refined to coarse), no additional action is needed. It proceeds automatically at the grid nodes of the coarse grid, which have become part of the refined grid.

The refinement procedure is started by introducing additional grid points to the initial coarse grid. When the refined grid at this level satisfies the refinement criterion (which will be discussed later) everywhere, the same procedure is repeated on the first level of refinement. This opens the second refinement level, and so on. This procedure is repeated until the given number of refinement levels is filled. In a test problem, presented in Section 7.2, it is shown how the accuracy of the solution varies as a function of the number of refinement levels.

The next time the refinement procedure is activated, the flow field and the scalar fields have probably changed. Before the refinement procedure is started the grid has to be cleared of those points, which have become dispensable, i.e. points that lie in regions where the gradient has become small. However, the points of the coarsest grid always remain present.

The choice of the refinement criterion is still open. The best choice would be one that gives a smooth distribution of the discretization error of all equations. This would plead for a refine-

ment criterion based on third- and higher-order derivatives of all variables. Grid refinements should, therefore, be chosen in such a way that they measure the derivatives of all variables (global grid refinement) or of each variable separately (which leads to the use of different grids for different variables). Fortunately, the regions where steepest gradients occur do (almost) coincide for all variables in combustion problems; moreover, these regions more or less coincide for all derivatives. Therefore, it suffices to use a global grid refinement based on, for example, the first spatial derivative of the temperature.

Note from equation (1) that it is also possible to obtain an estimate of the third- and higher-order derivatives of the scalar variables by means of the derivatives of the reaction rate. However, we will not use this as a refinement criterion in this paper, because we will not deal with the chemical details of the combustion model.

The method of grid refinement chosen, has the advantage that it is always possible to find the value of the variables by either the discretized equation (equations (10)–(16)) or by interpolation. Therefore, if an interpolation scheme is used in which the second-order derivative is represented correctly, there is no loss of order of approximation. The chosen interpolation is

$$f_P = f_{P'} + (f_A + f_B - f_C - f_D)/2 \tag{18}$$

with points P, P', A, B, C and D as shown in Figure 4 and f refers to the dependent scalar variables T, Y_i or ω. Higher-order interpolation schemes (including more grid points) based on Taylor-expansion methods are not used because they tend to yield non-physical oscillations.

We now turn to the determination of the flow-field variables in the interpolation points. One of the velocity components (u_P or v_P) in the interpolation points (depending on the direction of interpolation) is calculated using equation (13) to assure mass conservation around the point P'. The other velocity component may be calculated using equation (18) on either ρu or ρv. Furthermore, the stream function (φ_P) is calculated using the central-difference approximation to its derivative. This gives

$$(\varphi_A - 2\varphi_P + \varphi_B)/\Delta y^2 = (\rho_A u_A - \rho_B u_B)/2\Delta y \tag{19a}$$

or

$$(\varphi_A - 2\varphi_P + \varphi_B)/\Delta x^2 = -(\rho_A v_A - \rho_B v_B)/2\Delta x \tag{19b}$$

for $\Delta x = (x_A - x_B)/2$ or $\Delta y = (y_A - y_B)/2$ equal to zero, respectively.

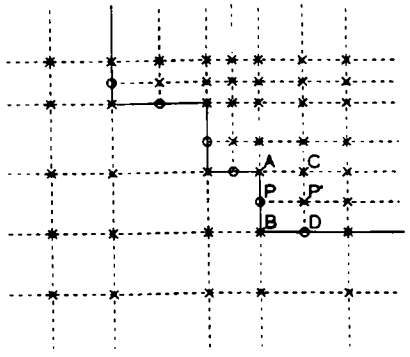

Figure 4. The locally refined grid continuous line: the boundary of the refinement (×) points determined by the discretized equations (interior points) (○) points determined by interpolation

For managing the grid points, a fairly excessive (but easy) method is chosen in which each point is described by its kind (interior, boundary or interpolation), place (x, y), the grid-point numbers of the eight neighbours (N, NE, E, SE, S, SW, W, NW) in the refinement level of P and the grid-point numbers of the four (in north, east, south and west direction) possible intermediate points in the higher refinement level. This means storage of 13 integer and 2 real values for every point. It is possible to manage the grid with less use of memory.[1] However, this does not seem to be of interest, since the storage of all variables consumes far more memory. Furthermore, the method chosen makes it possible to find all points and values needed in the calculations fairly quickly, which is of course an advantage with respect to the calculation time.

6. SOLUTION PROCEDURE

The calculations in each iteration loop, as presented in Figure 1, start with the calculation of the density (according to the equation of state), the transport coefficients and source terms using the most recent values of all variables. The discretized equations for T, Y_i and ω become linear and uncoupled in this way. After this, the stream-function field (φ) is solved according to equation (16) (with the old values of u and v). Next, the flow field (u, v) is updated using equation (13) with the new values of the stream function. The flow field now satisfies the continuity equation and new solutions to the discretized scalar equations for T, Y_i and ω may be calculated. This iteration loop is repeated until a stable solution is reached, while the grid is updated each time after a given number of iteration loops.

A set of linear equations (first, equation (16) for φ and, next, equation (10) for T, Y_i and ω) has to be solved during each iteration loop. There are numerous ways to do so. Some of these (the Jacobi, Runga–Kutta, Gauss–Seidel and A(lternating) D(irection) I(mplicit) method) will be described here shortly and compared in terms of the convergence rate in the combustion problem (see Section 8).

In the Jacobi method, the linear equations (equations (10) and (16)) are rewritten as

$$f'_P = (S_P + \sum_{N,S,E,W} \psi_j f_j^n)/\psi_P \tag{20}$$

where superscript n denotes the nth iteration loop. The value of f in the $(n+1)$th iteration is calculated using underrelaxation (when needed):

$$f_P^{n+1} = f_P^n + \eta(f'_P - f_P^n) \tag{21}$$

with relaxation parameter η. Equations (20) and (21) may be combined into

$$\frac{f_P^{n+1} - f_P^n}{\eta/\psi_P} = S_P - \psi_P f_P^n + \sum_{N,S,E,W} \psi_j f_j^n \tag{22}$$

For the variables ω, T and Y_i, this means that each Jacobi iteration is in fact a transient Euler-explicit time step, with a local time step equal to η/ψ_P; the left-hand side of equation (22) represents the discretized form of the transient term $\partial f/\partial t$ in the conservation equations. This is of course not true for φ, since there is no transient equation for φ. Therefore, the successive solutions of the Jacobi iterations only represent a proper time sequence when, on each time step, an accurate solution of equation (16) is calculated.

It is also possible to use more than one Jacobi loop (equation (22)) on the same set of linearized equations (i.e. S and ψ remain fixed), i.e. to solve the linearized equations with more accuracy before performing a new linearization. However, this does not lead to a decrease of the number of

global iteration steps, which is probably caused by the sensitivity of the solution to small changes in the fuel-consumption rate in equation (10).

This is also true for Runga–Kutta (RK) methods, in which a number solutions of previous Jacobi iterations (f^n, f^{n-1},$\cdots$) are combined into a new solution (f^{n+1}). We have tested the two-step and rational three-step method,[16] which both turn out to lead to the same type of convergence behaviour. There are in fact two possibilities to use RK-methods. First, they may be applied on the linearized equations (using the same S and ψ on each Jacobi iteration loop). This means that the source-term update is delayed, which causes convergence to slow down. Second, they have been applied to the total set of equations (with updates of S and ψ after each Jacobi iteration loop), thus predicting the evaluation of the coupled variables. In this case, the RK-methods tend to slow down convergence (or, in some cases, even cause divergence). This may again be caused by the sensitivity of the solution to changes in the fuel-consumption rate.

Another alternative is the Gauss–Seidel method, which uses the most recent value of f_j on the right-hand side of equation (20) instead of f_j^n. This has the advantage that it should lead to the stable solution quicker than the Jacobi method and does not need storage of the old solution. However, a big disadvantage of the Gauss–Seidel method is that, when the numbering of the grid points is structured, the errors tend to propagate along the direction in which the update of the solution takes place. In the case of combustion modelling, this gives rise to a flame front, which tends to move back and forth during the calculation. When an adaptive, locally refined grid is used, the region of refinement moves back and forth with the flame (thus causing a slowly converging solution), unless we wait until the flame has stabilized sufficiently, before the grid is updated. Therefore, on adaptive locally refined grids, it seems more sensible to use Jacobi instead of Gauss–Seidel iteration.

The methods described previously, treat the coupling of the solution of a variable in different grid points (almost) fully explicit, which has the big advantage that no matrix equations have to be solved during the calculation process. Note that the numbering of grid nodes and the matrix may very well be unstructured in the locally refined grid. However, it is possible to maintain structured node numbering in the locally refined grid in such a way that the Alternating Direction Implicit (ADI) method[16] can be used. The advantage of the ADI method is that it is more stable and leads to the stationary solution quicker than the Jacobi and Gauss–Seidel methods. Furthermore, the number of calculations needed in one iteration loop is only slightly larger than in the Jacobi or Gauss–Seidel methods. An additional advantage compared to Gauss–Seidel iteration is that the errors do not propagate in one direction but are always spread homogeneously.

However, the use of the ADI method has two major disadvantages. First, it calls for the additional storage of variables to fill the tri-diagonal matrix equation and, second, the numbering of the grid points and the coupling of different refinement levels on a locally refined grid calls for additional attention. In terms of reduction of calculation time, it might be rewarding to perform further study on the use of more complex implicit coupling of the different refinement levels. Using the ADI iteration method on a locally refined grid has not been done before and the optimization of an ADI solver on such a grid is, therefore, subject for future study.

The different iteration methods have been compared for typical combustion problem described in Section 8. The speed-up caused by the use of ADI on a equidistant grid is clearly shown in Figure 5(a), where the residuals during the solution procedure using ADI are compared with those using Jacobi and Gauss–Seidel methods. However, as Figure 5(b) shows, the application of ADI still has to be improved, before it is useful on a locally refined grid. Furthermore, each iteration in the ADI method takes roughly 1·5 times as much computing effort as an iteration in the Jacobi or Gauss–Seidel method. The use of the ADI method in its present state of develop-

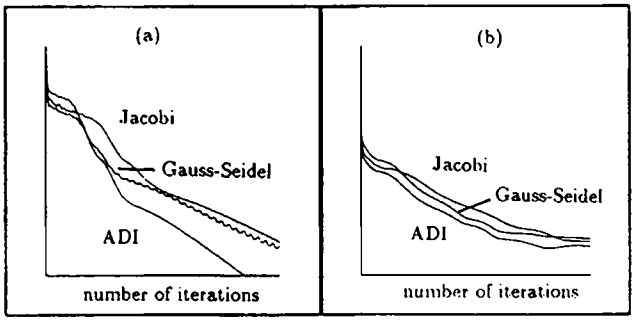

Figure 5. Comparison of the decrease of the residuals during the iteration process of Jacobi, Gauss–Seidel and ADI: (a) on an equidistant grid; (b) on a fixed locally refined grid

ment on a locally refined grid, therefore, increases computation time. In the remainder of this paper, we will use the Jacobi method instead.

7. TEST PROBLEMS

The accuracy of the discretization methods presented in this paper is tested using two benchmark problems: a lid-driven cavity and the flow over a thermal step.

7.1. A lid-driven cavity

The lid-driven cavity problem (as shown in Figure 6) is used to compare the accuracy of our solver with that of other non-staggered grid solvers.[17] The different discretization methods described in Section 3.1 will be compared, for the scalar equations (equations (10) and (11) applied to the vorticity equation) as well as for the flow field (equations (12) and (15) and equations (13) and (16)). In this benchmark, the flow field is solved for an incompressible isothermal ($\rho = 1$) flow with an additional term

$$8\mu\left(a\frac{\partial^4 b}{\partial y^4} + 2\frac{\partial^2 a}{\partial x^2}\frac{\partial^2 b}{\partial y^2} + b\frac{\partial^4 a}{\partial x^4}\right) - 64\left(Ba\frac{\partial a}{\partial x} - Ab\frac{\partial b}{\partial y}\right) \tag{23a}$$

with

$$a = x^4 - 2x^3 + x^2 \tag{23b}$$

$$b = y^4 - y^2 \tag{23c}$$

$$A = -24x^5 + 60x^4 - 56x^3 + 24x^2 - 4x \tag{23d}$$

$$B = -24y^5 + 8y^3 - 4y \tag{23e}$$

added to the source term on the right-hand side of the vorticity equation (6). The analytical solution (shown in Figure 7) to this problem is given by

$$u_{ex}(x, y) = 8a\frac{\partial b}{\partial y} \tag{24a}$$

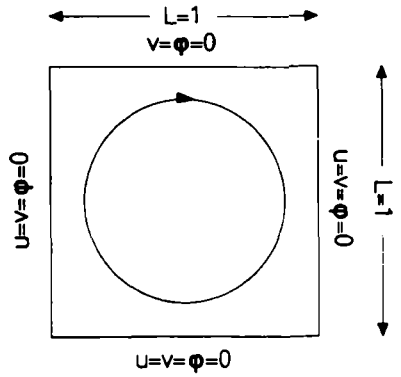

Figure 6. Geometry treated in the lid-driven cavity benchmark

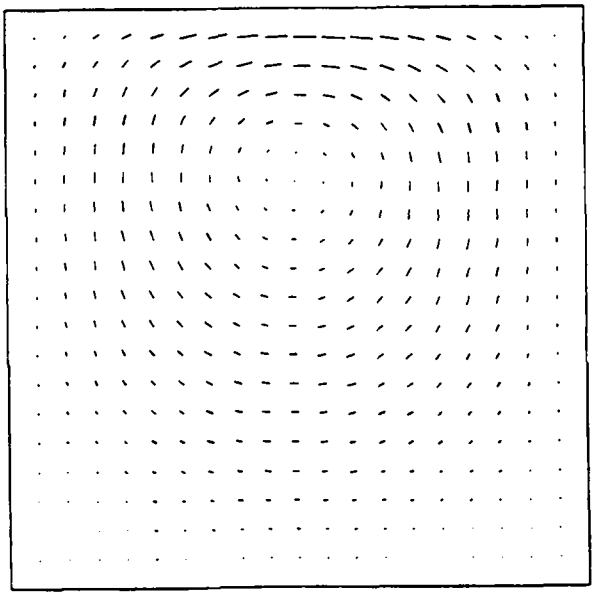

Figure 7. The velocity vectors according to equation (24)

and

$$v_{ex}(x, y) = -8b\frac{\partial a}{\partial x} \tag{24b}$$

Note that this solution does not depend on the viscosity (μ).

In Table I, the numerical values (u_{num} and v_{num}) are compared with the exact solution by means of the deviation

$$\sigma_u = \left(\frac{\sum_N (u_{ex} - u_{num})^2}{N}\right)^{1/2} \tag{25}$$

where N denotes the number of grid nodes (σ_v is defined accordingly).

Table I. Deviations ($\sigma_{u,v}$) as defined in equation (25) for $\mu = 1$ and $\mu = 0{\cdot}01$

Grid	$\sigma_u(\mu = 1)$	$\sigma_v(\mu = 1)$	$\sigma_u(\mu = 0{\cdot}01)$	$\sigma_v(\mu = 0{\cdot}01)$
Shih[1] 7 (21 × 21) Using (12) and (15)	0·003	0·003		
With hybrid discretization				
Uniform 21 × 21	0·00289	0·00306	0·00301	0·00307
Locally refined	0·005	0.006	0.00701	0.007
With power-law discretization				
Uniform 21 × 21	0·00289	0·00306	0·00363	0.00412
Locally refined	0·005	0·006	0·008	0·01
With upwind discretization				
Uniform 21 × 21	0·00284	0·00317	0·00835	0·0120
Locally refined	0·005	0·006	0·014	0·023
Using (13) and (16)				
With hybrid discretization				
Uniform 11 × 11	0·00125	0·00103	0·00570	0·00584
Uniform 21 × 21	0·00024	0·00018	0·00078	0·00085
Locally refined	0·00084	0·00085	0·00377	0·00371

As Table I shows, our results compare well with the results of Shih *et al.*[17] for all three types of discretization of the vorticity equation, combined with the use of the flow-field discretization techniques given by equations (12) and (15). The hybrid and power-law schemes give the best results. This is explained by the fact that these schemes have no artificial diffusion for small Peclet numbers (< 2), as opposed to the upwind scheme. For this test problem, the higher-order flow-field discretization scheme given in equations (13) and (16) improve the accuracy of the solution on the 21 × 21 grid by approximately a factor of 5. In fact, the $\sigma_{u,v}$ values found on an 11 × 11 grid using the higher-order equations are of the same order of magnitude as the $\sigma_{u,v}$ values found on the 21 × 21 grid using equations (12) and (15).

The accuracy and stability problems for $\mu < 0{\cdot}1$, mentioned by Shih *et al.*,[17] are circumvented by the use of the discretization schemes, presented in this paper. Therefore, stable and accurate results are found for all $\mu(> 0)$.

The results of a 2-layer refined grid on a 11 × 11 coarse grid, using the first derivative of the vorticity as the refinement criterion (see Figure 8), are also presented in Table I. These grids consist of about 300–400 grid points. The values in Table I indicate that this problem is not well suited for the application of a local grid refinement. This is also shown in Figure 8 in which three grids are presented, using the first derivative of φ, ω or $u^2 + v^2$ as the refinement criterion, respectively. The differences in these grids show that different variables have their steepest gradient in different regions. In effect, this means that the solution to this problem calls for local refinement on the whole domain and a uniform grid is, therefore, preferred.

7.2. Flow over a thermal step

In the previous test problem, the locally refined grid proves to be of little use. This is due to the fact that the accuracy of the solution is of the same order of magnitude in most of the calculation domain. The performance of the algorithm, proposed in this paper, is tested by means of an additional test problem, which contains a relatively small area with steep gradients. In this test

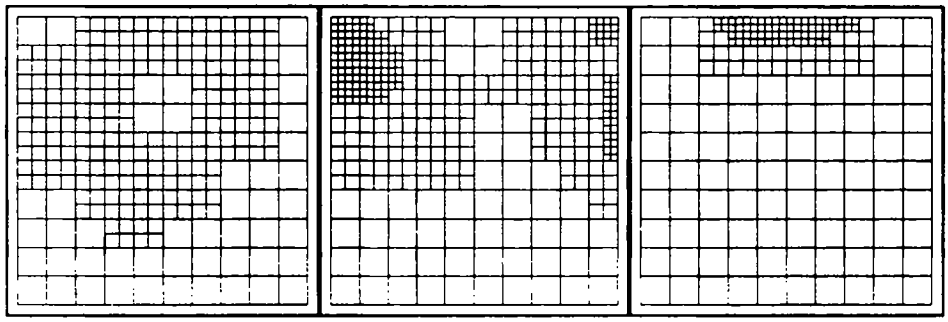

Figure 8. The 2-layer refined grid on an 11×11 grid from left to right: based on gradients of φ, ω and $u^2 + v^2$

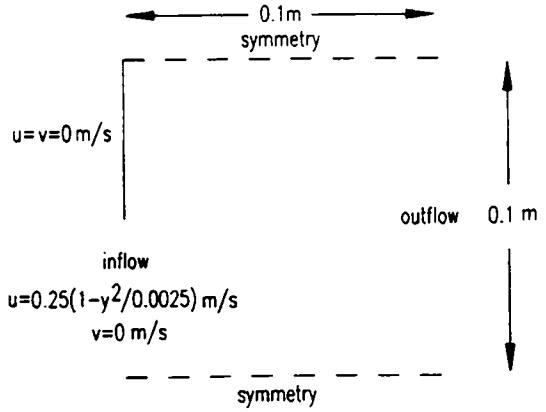

Figure 9. The calculation domain for the flow over a thermal step

problem, the flow field over a thermal step is solved in a square domain of $0{\cdot}1 \times 0{\cdot}1$ m^2 as shown in Figure 9. The temperature field is fixed and the energy equation is, therefore, not solved. The resulting flow field resembles that of a combustion problem. The use of a fixed temperature field rather than using a real two-dimensional flame as a problem, is done to have a spatially fixed 'flame'. The position of the flame depends on the accuracy of the solution in a real combustion problem, which means that the accuracy of the flow-field solution is tested separately. The thermal step is constructed by fixing the temperature as follows:

$$T(x, y) = 300\,\{1 + 2\exp\kappa(r^2/r_0^2 - 1)\} \quad \text{for } r < r_0 \tag{26a}$$

and

$$T(x, y) = 300\,\{5 - 2\exp\kappa(1 - r^2/r_0^2)\} \quad \text{for } r > r_0 \tag{26b}$$

with $r^2 = x^2 + y^2$ and $r_0 = 0{\cdot}05$. Note that the temperature increases by a factor of 5. Therefore, the density decreases by a factor of 5 according to the equations of state (equation (4)). In order to get a temperature gradient at $r = r_0$ of about 500 K/mm, which is comparable to the gradient in an atmospheric methane/air flame, κ is chosen equal to 25. The values of the viscosity and density are given in Table III. The isotherms and streamlines for this test problem are shown in Figure 10.

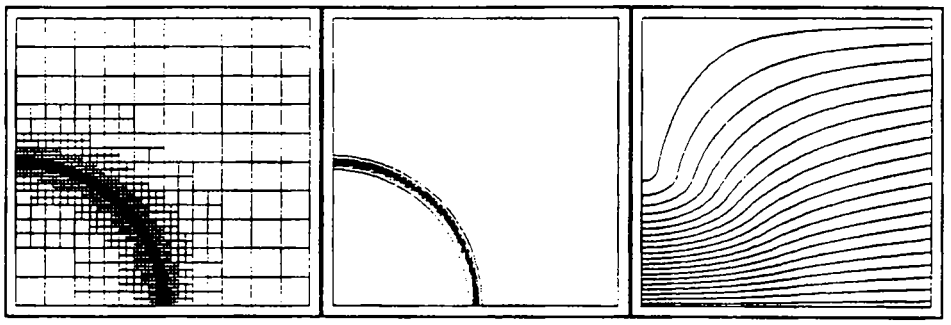

Figure 10. The results for the flow over a thermal step (from left to right): the 4-layer refined grid on an 11 × 11 basis, the isotherms and the streamlines

Table II. The results of the thermal step problem, presented by means of the values of u, v and φ in the centre of the calculation domain; the deviation defined in equation (27) for u, v and φ; the computational time, number of iterations (Niter) needed to obtain a stable solution and the number of grid nodes used

Grid	φ_{mid}	u_{mid}	v_{mid}	δ_u	δ_v	δ_ω	δ_φ	Time	N_{iter}	Points
Uniform										
11 × 11	5·55	0·364	0·208	0·044	0·135	0·694	0·017	00:00:01	150	121
21 × 21	5·59	0·340	0·197	0·057	0·135	0·565	0·014	00:00:18	750	441
41 × 41	5·66	0·339	0·197	0·053	0·111	0·431	0·012	00:05:24	3000	1681
81 × 81	5·69	0·357	0·212	0·020	0·041	0·166	0·005	00:58:55	8000	6541
161 × 161	5·70	0·368	0·221	0·0	0·0	0·0	0·0	15:48:52	32000	25921
11 × 11										
2 layers	5·67	0·344	0·196	0·041	0·080	0·407	0·012	00:00:12	500	418
3 layers	5.69	0·358	0·209	0·018	0·031	0·154	0·005	00:01:50	2000	852
4 layers	5.68	0·369	0·219	0·007	0·012	0·062	0·003	00:09:21	5000	1626
21 × 21										
2 layers	5·69	0·358	0·210	0·019	0·036	0·165	0·005	00:02:17	2000	1047
3 layers	5.67	0·370	0·221	0·008	0·017	0·075	0·004	00:09:59	5200	1722

Note that the refined grid is fixed during the iteration process, because of the fixed temperature field.

The results, accuracy and computational time of the solutions are compared in Table II for 5 equidistant grids and 5 refined grids based on coarse uniform grids of 11 × 11 and of 21 × 21 points. It is worth noting that the grid spacing in the fourth layer of refinement on the 11 × 11 grid (also presented in Figure 10) equals the spacing in the uniform 161 × 161 grid. This is also true for the third layer of refinement on the coarse 21 × 21 grid. Results for both velocity components as well as for the stream function in the centre point of the domain are presented in Table II. The results on the 161 × 161 grid are used as reference values (they are the most accurate). This is possible because all points occurring in different calculations are part of the grid points in this fine uniform mesh. The comparison is made by means of a global relative deviation (δ) defined as

$$\delta_f \equiv \frac{\sum_N |f - f_{161\times161}|}{\sum_N |f|} \tag{27}$$

where f represents either u, v, ω or φ in a node (x, y) of the grid under examination and $f_{161\times161}$ denotes the solution in the same node on the uniform 161×161 grid. Furthermore, N denotes the number of nodes of the grid under examination. Therefore, δ_f is a measure for the average relative deviation of f from $f_{161\times161}$.

Table II clearly shows the benefits of the locally refined grid. The refined 4-layer grid on the coarse 11×11 grid and the refined 3-layer grid on the coarse 21×21 grid are obviously more accurate than the solution on the uniform 81×81 grid, while they use 25 per cent of the grid points and only 15 per cent of the computation time. As a matter of fact, the accuracy of the solution on the 81×81 grid is comparable with that on the refined 3-layer grid on a coarse 11×11 grid and on the 2-layer refined grid on a coarse 21×21 grid, whereas the refined grids use only 15 per cent of the number of points and 3 per cent of the calculation time.

8. COMBUSTION MODELLING

In order to show that the presented algorithm converges towards the stable solution when the adaptive grid technique is used, we show some numerical results obtained for a laminar premixed methane/air flame on a slot burner (Figure 11). The combustion process is assumed to be governed by a one-step reaction (fuel (fu) + oxygen (ox) → products (pr)) with an Arrhenius-like fuel-mass consumption rate equal to

$$\dot{\rho}^{fu} = -A\rho^{\alpha+\beta}(Y^{fu})^{\alpha}(Y^{ox})^{\beta}\exp(-T_a/T) \tag{28}$$

where A, α, β and T_a are rate parameters. This expression for the fuel-mass consumption rate has been used as the right-hand side of the conservation equation (1) for the fuel-mass fraction. Multiplied by the stoichiometric ratio (s, mass of oxygen consumed per unit mass of fuel) or by the enthalpy release ($-\Delta H$, heat of combustion per unit mass of fuel), equation (28) is the right-hand side of equation (1) for the oxygen mass fraction and the enthalpy, respectively. In order to get a physically realistic flame, we have chosen a combination of physical and chemical parameters (see Table III) so that the flame speed of methane/air mixtures in a flat burner-stabilized flame are correctly reproduced.[18] The mass-diffusion rates of methane and oxygen in nitrogen have to be added to complete the set of parameters presented. They are chosen in accordance with the kinetic theory of binary mixtures.[3-5]

In Figure 12, the results of this calculation are presented by means of the calculated 2-layer refined grid on a coarse 51×21 grid. This figure illustrates the isotherms and the streamlines. This stabilized flame has been calculated without difficulties in approximately 2000 Jacobi iterations with a relaxation factor $\eta = 0{\cdot}4$, while the grid is adapted on each 500th iteration. The total calculation time was about 1 h on a Silicon Grafics 35/4D workstation.

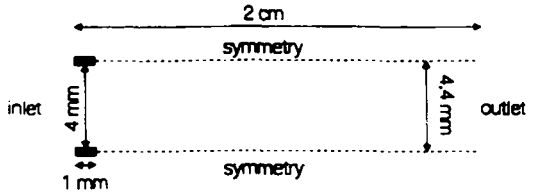

Figure 11. The geometry of the Bunsen-like slot burner, inlet conditions: stoichiometric methane/air mixture, temperature $T = 300$ K, parabolic velocity profile with maximum velocity of 1·21 m/s.

Table III. The physical and chemical parameters used in the flame model

Parameter	Value	Dimension
ρ_{STP}	1·241	kg/m^3
λ	$\lambda_{ref}(T/1500)^{\gamma}$	J/(mK s)
λ_{ref}	0·092	J/(mK s)
γ	0·77	1
μ	$0{\cdot}71\lambda/c_p$	kg/(ms)
c_p	1365	J/(kg K)
ΔH	$4{\cdot}813 \times 10^7$	J/kg
s^{ox}	3·883	kg^{ox}/kg^{fu}
A	$2{\cdot}6 \times 10^{15}$	$(kg/m^3)^{1-\alpha-\beta}/s$
α	2·8	1
β	1·2	1
T_a	16900	K

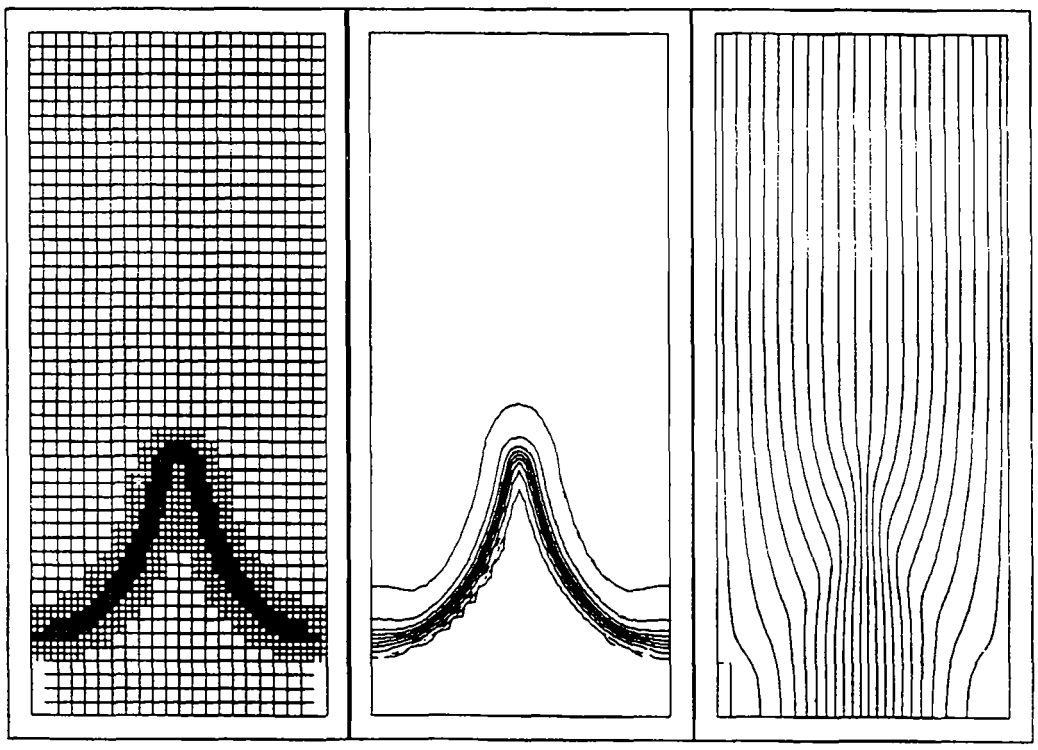

Figure 12. The resulting flame (from left to right): the 2-layer refined grid, the isotherms (from 400 to 2200 K in steps of 200 K) and the streamlines

9. CONCLUSIONS

The stream-function–vorticity formulation, used in this paper, makes it possible to perform flow-field calculations on a locally refined, non-staggered grid without non-physical checker-board oscillations. The resulting set of linear equations is solved using the easily implemented

Jacobi method. Furthermore, it appears that implementation of more complex linear solvers, such as ADI, does not improve the rate of convergence towards the stable solution in the locally refined grid.

The uniform grid solver compares favourably with the results of algorithms found in the literature, especially when higher-order flow-field equations (formed by equations (13) and (16)) are used in combination with the hybrid or the power-law discretization method for the scalar equations. Furthermore, in the test problem which is characterized by a local steep density gradient (comparable with the gradients present in laminar premixed methane/air flames), the solution of the flow field in a locally refined grid proves to give approximately the same accuracy as a uniform grid in about 3 per cent of the calculation time, using about 15 per cent of the grid points.

The flame calculation clearly shows that the algorithm, presented in this paper, converges towards a stable solution when the adaptive grid technique is used. The time needed to perform such a calculation is about 1 h on a Silicon Grafics 35/4D workstation, which makes the use of the algorithm feasible in combustion engineering.

REFERENCES

1. M. D. Smooke, R. E. Mitchell and D. E. Keyes, 'Numerical solution of two-dimensional axisymmetric laminar diffusion flames', *Combust. Sci. Technol.* **67**, 85–122 (1989).
2. S. V. Patankar, *Numerical Heat Transfer and Fluid Flow*, McGraw-Hill, New York, 1980.
3. R. B. Bird, W. E. Stewart and E. N. Lightfoot, *Transport Phenomena*, Wiley, New York, 1960.
4. J. O. Hirschfelder, C. F. Curtiss and R. B. Bird, *Molecular Theory of Gases and Liquids*, Wiley, New York, 1954.
5. S. Chapman and T. G. Cowling, *The Mathematical Theory of Non-Uniform Gases*, Cambridge Univ. Press, Cambridge, 1970.
6. J. D. Buckmaster and G. S. S. Ludford, *Theory of Laminar Flames*, Cambridge Univ. Press, Cambridge, 1982.
7. J. D. Buckmaster (ed.), *The Mathematics of Combustion*, SIAM, Philadelphia, 1985.
8. F. A. Williams, *Combustion Theory*, Addison–Wesley, Massachusetts, 1965.
9. R. A. Strehlow, *Combustion Fundamentals*, McGraw-Hill, New York, 1984.
10. R. Peyret and T. D. Taylor, *Computational Methods for Fluid Flow*, Springer, Berlin, 1983.
11. G. D. Raithby, 'Skew Upstream Differencing Schemes for Problems involving Fluid Flow', *Comput. Methods Appl. Mech. Eng.*, **9**, 153–164 (1976).
12. Y. A. Hassan, J. G. Rice and J. H. Kim, 'A stable mass-flow-weighted two-dimensional skew upwind scheme', *Numer. Heat Transfer*, **6**, 395–408 (1983).
13. G. D. Thiart, 'Finite-difference scheme for the numerical solution of fluid flow and heat transfer problems on nonstaggered grids', *Numer. Heat Transfer B*, **17**, 43–62 (1990).
14. S. C. R. Dennis, D. B. Ingham and R. N. Cook, 'Finite-difference methods for calculating steady incompressible flows in three dimensions', *J. Comput. Phys.*, **33**, 325–339 (1979).
15. P. J. Roache, *Computational Fluid Mechanics*, Hermosa, Albuquerque, 1972.
16. C. A. J. Fletcher, *Computational Techniques for Fluid Dynamics*, Springer, Berlin, 1988.
17. T. M. Shih, C. H. Tan and B. C. Hwang, 'Effects of grid staggering on numerical schemes', *Int. j. numer. methods fluids*, **9**, 193–212 (1989).
18. W. E. Kaskan, 'The dependence of flame temperature on the mass burning velocity', *Proc. 6th Int. Symp. on Combustion*, The Combustion Institute, Pittsburgh, 1967, pp. 134–143.