NUMERICAL SOLUTION OF FLAME SHEET PROBLEMS WITH AND WITHOUT MULTIGRID METHODS*

Craig C. Douglas
Department of Computer Science
Yale University
New Haven, Connecticut

Alexandre Ern
Department of Mechanical Engineering
Yale University
New Haven, Connecticut

SUMMARY

Flame sheet problems are on the natural route to the numerical solution of multidimensional flames, which, in turn, are important in many engineering applications. In order to model the flame structure more accurately, we use the vorticity-velocity formulation of the fluid flow equations instead of the streamfunction-vorticity approach. The numerical solution of the resulting nonlinear coupled elliptic partial differential equations involves a pseudo transient process and a steady state Newton iteration. Rather than working with dimensionless variables, we introduce scale factors that can yield significant savings in the execution time. In this context, we also investigate the applicability and performance of several multigrid methods, focusing on nonlinear damped Newton multigrid, using either one way or correction schemes.

1. INTRODUCTION

Recent advances in the development of computational algorithms and supercomputers have provided new extremely powerful tools with which to investigate chemically reacting systems that were computationally infeasible only a few years ago (see [1], [2], [3], and [4]). The difficulties associated with solving high heat release combustion problems stem from the large number of dependent unknowns, the nonlinear character of the governing partial differential equations and the different length scales present in the problem. Typical combustion problems may involve, in addition to the temperature and the fluid dynamics variables, dozens of species defined at each grid point and require the resolution of curved fronts whose thickness is on the order of thousandths of the domain diameter, across which critical fields vary by orders of magnitude. As a result of the fluid dynamics-thermochemistry interaction and its effect on the flame structure, the governing

*This work was supported in part by CERMICS, ENPC, IBM, the Office of Naval Research, and the Department of Energy.

equations are strongly coupled together and are also characterized by the presence of stiff source terms and nonlinearities. Hence, Newton methods with sophisticated control strategies, including damping and adaptive continuation techniques are needed. However, in spite of these difficulties, the numerical modeling of multidimensional laminar (or turbulent) flames has been recently motivated by the growing demand for high fuel efficiency combined with low pollutant emission. While three dimensional turbulent flame simulations still remain infeasible on current supercomputers, axisymmetric laminar diffusion flames constitute a problem of practical importance since they are the flame type of several combustion devices. Hence, new robust numerical models of such a system will provide an efficient tool to probe flame structures and investigate the coupled effects of complex transport phenomena with chemical kinetics.

As part of an ongoing effort to expand combustion modeling capabilities, we investigate computationally the performance of several multigrid techniques (see [5], [6], [7], and [8]) combined with the numerical solution of combustion related problems. In the present work, we consider a flame sheet problem rather than a finite rate chemistry model for an axisymmetric laminar diffusion flame in order to alleviate the memory and CPU requirements on the computer simulations. The numerical techniques presented in this paper, however, also apply to combustion problems with finite rate chemistry [9]. We note that a flame sheet model adds only one field to the hydrodynamic fields that describe the underlying flow. A detailed kinetics model adds as many fields as species considered in the kinetic mechanism, each with its own coupled conservation equation. Since the CPU time and the memory requirements scale with the square of the number of dependent unknowns, the flame sheet model considerably reduces the cost of the computer simulations while still keeping the coupling and nonlinearity features associated with the original problem.

In the flame sheet model, the chemical reactions are described with a single one step irreversible reaction corresponding to infinitely fast conversion of reactants into stable products. This reaction is assumed to be limited to a very thin exothermic reaction zone located at the locus of stoichiometric mixing of fuel and oxidizer, where temperature and products of combustion are maximized. To further simplify the governing equations, one neglects thermal diffusion effects, assumes constant heat capacities and Fick's law for the ordinary mass diffusion velocities, and takes all the Lewis numbers equal to unity [2]. With these approximations, the energy equation and the major species equations take on the same mathematical form and by introducing Schvab-Zeldovich variables, one can derive a source free convective-diffusive equation for a single conserved scalar. Although no information can be recovered about minor or intermediate species in the flame sheet limit, the temperature and the stable major species profiles in the system can be obtained from the solution of the conserved scalar equation coupled to the flow field equations. Further, the location of the physical spatially distributed reaction zone and its temperature distribution can be adequately predicted by the flame sheet model for many important fuel-oxidizer combinations and configurations. Since being studied as a means of obtaining an approximate solution to use as an initial iterate for a one dimensional detailed kinetics computation in [10], flame sheets have been routinely employed to initialize multidimensional diffusion flames.

In §2, a comparison of three possible formulations of the problem is presented, including the governing equations and boundary conditions. In §3, the general solution algorithms are presented, including a damped Newton method, Jacobian evaluation, linear solvers (Bi-CGSTAB or GMRES), and the pseudo transient process. In §4, various multigrid methods are discussed in the context of flame sheets. In §5, numerical experiments are presented. Finally, in §6, some conclusions are

reached.

2. VORTICITY-VELOCITY FORMULATION

In diffusion flames the combustion process is primarily controlled by the rate at which the fuel and oxidizer are brought together in stoichiometric proportions. Thus, independently of the submodel used for the chemical kinetics (finite rate vs. flame sheet), the overall accuracy of the numerical solution strongly depends on an accurate representation of the flow field. Hence, a brief discussion on the various formulations of the Navier-Stokes equations in the context of laminar combustion problems is of order.

The first numerical solution of two dimensional axisymmetric laminar diffusion flames was obtained using the streamfunction-vorticity formulation [2]. This approach is attractive for three reasons:

1. It eliminates the coupling associated with the presence of the pressure in the momentum equations.

2. It reduces the number of equations to be solved by one.

3. It also has the important advantage that continuity is explicitly satisfied locally.

However, the specification of boundary conditions meets with difficulties when one attempts to specify vorticity boundary values. In particular, a zero vorticity boundary condition at the inlet of the computational domain results in a rough approximation of the true solution, thus severely altering the resulting velocity field [3]. On the other hand, the specification of vorticity boundary values in terms of the streamfunction requires the discretization of second order derivatives, thus yielding off diagonal terms in the Jacobian matrix which result in having to solve severely ill conditioned linear systems. Another important difficulty associated with the streamfunction-vorticity approach is that the extension to three dimensional configurations through the introduction of a vector potential instead of the scalar streamfunction is cumbersome and computationally expensive since it introduces additional dependent variables.

Alternatively, a primitive form of the Navier-Stokes equations has been recently implemented for several axisymmetric laminar diffusion flames (see [3] and [4]). In this approach, the velocity field is computed using the momentum equations and the pressure field is recovered from the continuity equation. As a result of the difference in nature of the governing equations, the discrete pressure field has to be determined in a manner consistent with the discrete continuity equation. This can be achieved to machine zero on a staggered grid. However, staggered mesh schemes do also have drawbacks in complex geometries configurations where non-orthogonal curvilinear coordinates are used and when using sophisticated numerical techniques such as multigrid methods (see [11] and [12]). Although feasible ([13] and [14]), the development of staggered grid based multigrid solvers is computationally cumbersome since the transfer operators between levels do not coincide for each dependent variable in order to preserve a staggered grid arrangement on all levels. This difficulty may even be further exacerbated in three dimensional configurations. Finally, it is worthwhile to note that two and three dimensional solutions of incompressible viscous flows on a nonstaggered grid have been reported (see [11] and [12]). However, the extension of such procedures to highly compressible systems where the density can vary by several orders of magnitude inside the computational domain may still yield some complications.

The vorticity-velocity formulation constitutes a third approach to the numerical solution of the Navier-Stokes equations. A review of incompressible fluid flow computations using this formulation is well documented in [15]. The vorticity-velocity formulation of the Navier-Stokes equations has been recently extended to two and three dimensional compressible flows and implemented for the numerical solution of flame sheet problems (see [16] and [17]). As motivated in these references, a vorticity-velocity formulation allows replacement of the first order continuity equation with additional second order equations. Whereas the streamfunction-vorticity formulation also accomplishes the same replacement in two dimensions, vorticity-velocity is extensible to three and allows more accurate formulation of boundary conditions in a numerically compact way. Furthermore, off diagonal convective terms in off diagonal blocks that exert a strong influence in a streamfunction-vorticity formulation disappear. Another important attractive feature of the vorticity-velocity formulation is that the governing equations can be discretized on a nonstaggered grid, thus allowing the implementation of a multigrid algorithm at a relatively low overhead in additional programming (see [16], [17], and [18]).

The flame sheet governing equations consist of the conservation of total mass, momentum and a conserved scalar equation. The conservation of total mass and momentum equations constitute the flow field problem and are formulated using the vorticity-velocity formulation of the compressible axisymmetric Navier-Stokes equations. A source free convective-diffusive equation for a conserved scalar is solved coupled together with the flow field equations and the temperature and major stable species profiles in the system can be recovered from the conserved scalar (see [2], [19], and references therein).

We introduce the velocity vector $v = (v_r, v_z)$ with radial and axial components v_r and v_z, respectively, and the normal component of the vorticity

$$\omega = \frac{\partial v_r}{\partial z} - \frac{\partial v_z}{\partial r}. \tag{1}$$

The vorticity transport equation is formed by taking the curl of the momentum equations, which eliminates the partial derivatives of the pressure field. A Laplace equation is obtained for each velocity component by taking the gradient of (1) and using the continuity equation. This yields the governing equations in the following form:

$$\begin{aligned}
\tfrac{\partial^2 v_r}{\partial r^2} + \tfrac{\partial^2 v_r}{\partial z^2} &= \tfrac{\partial \omega}{\partial z} - \tfrac{1}{r}\tfrac{\partial v_r}{\partial r} + \tfrac{v_r}{r^2} - \tfrac{\partial}{\partial r}\left(\tfrac{v\cdot\nabla\rho}{\rho}\right), \\
\tfrac{\partial^2 v_z}{\partial r^2} + \tfrac{\partial^2 v_z}{\partial z^2} &= -\tfrac{\partial \omega}{\partial r} - \tfrac{1}{r}\tfrac{\partial v_r}{\partial z} - \tfrac{\partial}{\partial z}\left(\tfrac{v\cdot\nabla\rho}{\rho}\right), \\
\tfrac{\partial^2 \mu\omega}{\partial r^2} + \tfrac{\partial^2 \mu\omega}{\partial z^2} + \tfrac{\partial}{\partial r}\left(\tfrac{\mu\omega}{r}\right) &= \rho v_r \tfrac{\partial \omega}{\partial r} + \rho v_z \tfrac{\partial \omega}{\partial z} - \tfrac{\rho v_r}{r}\omega + \overline{\nabla}\rho\cdot\nabla\tfrac{v^2}{2} - \overline{\nabla}\rho\cdot g + \\
&\quad 2\left(\overline{\nabla}(\mathrm{div}(v))\cdot\nabla\mu - \nabla v_r\cdot\overline{\nabla}\tfrac{\partial\mu}{\partial r} - \nabla v_z\cdot\overline{\nabla}\tfrac{\partial\mu}{\partial z}\right), \\
\tfrac{1}{r}\tfrac{\partial}{\partial r}(r\rho D\tfrac{\partial S}{\partial r}) + \tfrac{\partial}{\partial z}(\rho D\tfrac{\partial S}{\partial z}) &= \rho v_r\tfrac{\partial S}{\partial r} + \rho v_z\tfrac{\partial S}{\partial z},
\end{aligned} \tag{2}$$

where ρ is the density, μ the viscosity, g the gravity vector, $\mathrm{div}(v)$ the cylindrical divergence of the velocity vector, S the conserved scalar, D a diffusion coefficient, and the components of $\overline{\nabla}\beta$ are $(\frac{\partial\beta}{\partial z}, -\frac{\partial\beta}{\partial r})$. The density is computed using the perfect gas law and, in the low Mach numbers approximation valid for these flame configurations, one can use the outlet (constant) pressure.

Table 1: Boundary conditions

Axis of symmetry ($r = 0$)	$v_r = 0$	$\frac{\partial v_z}{\partial r} = 0$	$\omega = 0$	$\frac{\partial S}{\partial r} = 0$
Outer zone ($r = R_{max}$)	$\frac{\partial v_r}{\partial r} = 0$	$\frac{\partial v_z}{\partial r} = 0$	$\omega = \frac{\partial v_r}{\partial z}$	$S = 0$
Inlet ($z = 0$)	$v_r = 0$	$v_z = v_z^0(r)$	$\omega = \frac{\partial v_r}{\partial z} - \frac{\partial v_z}{\partial r}$	$S = S^0(r)$
Exit ($z = L$)	$v_r = 0$	$\frac{\partial v_z}{\partial z} = 0$	$\frac{\partial \omega}{\partial z} = 0$	$\frac{\partial S}{\partial z} = 0$

Consequently, in the above formulation, the pressure field is eliminated from the governing equations as a dependent unknown and can be recovered, once a computed numerical solution of (2) is obtained, by solving a Laplace type equation derived by taking the divergence of the momentum equations [15].

Recalling that all of the Lewis numbers are taken equal to unity, the quantity ρD is given by the viscosity coefficient μ divided by a reference Prandtl number and we use an approximate value for air, $Pr = 0.75$. Hence, in this model, the determination of all the transport coefficients is reduced to the specification of a transport relation for the viscosity and we use the same power law as the one given in [2]. We also note that, due to the high temperature gradients present in the system, the viscosity derivatives in the right hand side of the vorticity transport equation (2) can not be neglected. Our numerical experiments show that such an approximation leads to significant differences in the numerical solution, especially for the radial velocity profile. Finally, a conservative form of the convective terms can also be considered but it yields slower convergence rates without any significant changes in the computed solution.

A schematic of the physical configuration is given in Figure 1. It consists of an inner cylindrical fuel jet (radius R_I =0.2cm), an outer co-flowing annular oxidizer jet (radius R_O =2.5cm) and a dead zone extending to R_{max} =7.5cm. The inlet velocity profile of the fuel and oxidizer are a plug flow of 35cm/s. This yields a typical value for the Reynolds number of 550. Further, the flame length is approximately L_f =3cm [19] and the length of the computational domain is set to L =30cm. Although the fuel and oxidizer reservoirs are at room temperature (300°Kelvin), we need to assume, in the flame sheet model, that the temperature already reaches the peak temperature value along the inlet boundary at $r = R_I$. This peak temperature is estimated for a methane-air configuration to be 2050°K. Hence, the inlet profile of the conserved scalar, $S^0(r)$, is specified in such a way that the resulting temperature distribution blends the room temperature reservoirs and the peak temperature by means of a narrow Gaussian centered at R_I. The narrowness of the Gaussian profile has a relevant influence on the calculated flame length, so that its parameters have to be determined appropriately [19]. The boundary conditions are summarized in Table 1. Finally, we note that the use of the definition of the vorticity (1) for the vorticity outlet boundary condition does not yield any relevant changes in the computed solution.

3. GENERAL SOLUTION ALGORITHM

The partial differential equations (2) together with the boundary conditions (see Table 1) are discretized on a two dimensional tensor product grid. A solution is first obtained on an initial coarse grid. Additional mesh points are then adaptively inserted in regions of high physical activity by equidistributing weight functions of the local gradient and curvature of the numerical solution

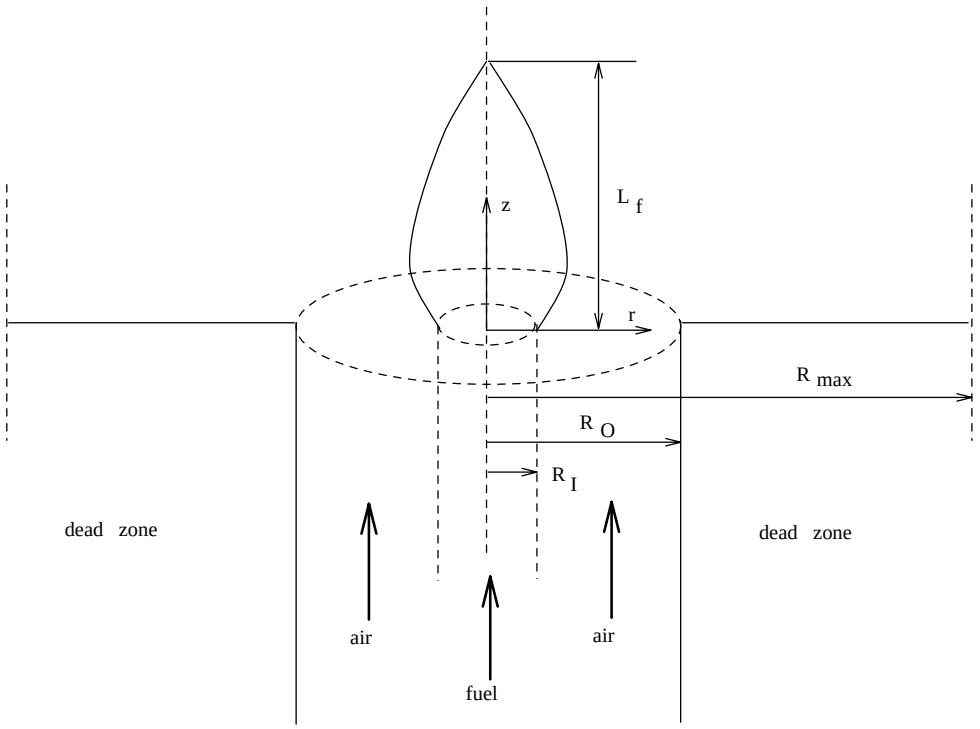

Figure 1: Physical configuration (not in scale)

[2], which yields a 129×161 grid. To verify the grid independence of the solution, we refined this grid to 257×219 points. The relative error between the two solutions was found to be lower than 2% and differences were only encountered in the outflow region where the grids were still kept somewhat coarse. However, the flame length and the temperature distribution inside the flame were accurately predicted on the 129×161 grid. Hence, this grid will be considered as the finest grid in the present work.

The spatial operators in the partial differential equations (2) are approximated with finite difference expressions. Diffusion and source terms are evaluated using centered differences. We adopt a monotonicity preserving upwind scheme for the convective terms (see [20, p. 304]), for instance,

$$v_r \frac{\partial S}{\partial r} = \max\{(v_r)_{i-\frac{1}{2}}, 0\} \frac{S_i - S_{i-1}}{r_i - r_{i-1}} - \max\{-(v_r)_{i+\frac{1}{2}}, 0\} \frac{S_{i+1} - S_i}{r_{i+1} - r_i}. \tag{3}$$

The boundary conditions given in Table 1 involve only zero or first order derivatives. For the latter terms, first order back or forward differences can be used, except for two boundary conditions which require a more accurate treatment. First, as motivated in [17], the vorticity inlet boundary condition is discretized using the vorticity values at the first two lines of the computational domain. More specifically, at an inlet point $(i, 1)$, we discretize the equation $\omega = \frac{\partial v_r}{\partial z} - \frac{\partial v_z}{\partial r}$ as follows:

$$\frac{1}{2}(\omega_1 + \omega_2) = \frac{(v_r)_2}{z_2 - z_1} - \frac{(v_z)_{i+1} - (v_z)_{i-1}}{r_{i+1} - r_{i-1}}. \tag{4}$$

It is also of critical importance for the accuracy of the numerical solution that the axial velocity boundary condition on the axis of symmetry be evaluated using a second order scheme. At any

point $(1, j)$, we have

$$(v_z)_2 - (v_z)_1 = \frac{(r_2 - r_1)^2}{2} \frac{\partial^2 v_z}{\partial r^2} + \mathcal{O}\left((r_2 - r_1)^2\right).$$

The right hand side is evaluated using the Laplace equation for v_z in (2). On the axis of symmetry, this reduces to

$$\frac{\partial^2 v_z}{\partial r^2} = -\frac{\partial^2 v_z}{\partial z^2} - \frac{\partial \omega}{\partial r} - \frac{\partial}{\partial z}\left(\frac{v_z}{\rho}\frac{\partial \rho}{\partial z}\right).$$

The radial derivative of the vorticity can be discretized with a first order difference while still yielding an overall second order accuracy for v_z. By comparing our numerical solutions with a primitive variable solution of the same problem [19], we found that these two boundary conditions exerted a strong influence on the overall accuracy of the numerical solution.

The discretization of the partial differential equations (2) together with the boundary conditions (Table 1) yields a set of algebraic equations of the form $F(U) = 0$, which is solved using a damped Newton method

$$J(U^n)\Delta U^n = -\lambda^n F(U^n), \qquad n = 0, 1, \ldots, \tag{5}$$

with convergence tolerance $\|\Delta U^n\|_S < 10^{-5}$. The Jacobian matrix $J(U^n)$ is computed numerically using vector function evaluations and the grid nodes are split into nine independent groups which are perturbed simultaneously (see [2] for more details). Selected cases were rerun with a more stringent convergence tolerance of 10^{-6}, without any significant changes in the numerical solution. Rather than working with dimensionless variables, we introduce a scale factor α_l, $l \in [1, n_c]$, for each dependent variable ($n_c = 4$ for the flame sheet problem). The norm of the discrete vector ΔU^n is then given by

$$\|\Delta U^n\|_S = \sqrt{\sum_{i\in[1,n_r]} \sum_{j\in[1,n_z]} \sum_{l\in[1,n_c]} (\alpha_l \Delta U^n(l,i,j))^2}. \tag{6}$$

It is worthwhile to point out that an appropriate choice of the scale factors can yield significant savings in the execution time. This point will be further illustrated with numerical experiments in §5.1.

The linear system (5) is inverted at each Newton step through an inner iteration. This inner iteration may consist of either the Bi-CGSTAB algorithm [21] or a restarted version of GMRES [22] combined with a Gauss-Seidel (GS) left preconditioner. This choice is motivated in [16] through various numerical simulations of flame sheet problems. Although a single Bi-CGSTAB/GS iteration requires approximately 1.5 times more time than an average GMRES/GS iteration, both algorithms yield total execution times which are in general within a few percent of each other. The former has lower memory requirements (see the end of §5.2 for more details). The convergence of the inner iteration is based on the norm of the left preconditioned linear residual using an absolute tolerance equal to one-tenth of the Newton tolerance. Such termination criterion brings enough information on the update vector ΔU^n back to the Newton iteration (see [16] for more details).

Due to the nonlinearity of the original problem, a pseudo transient process is used to produce a parabolic in time problem and bring the starting estimate into the convergence domain of the steady Newton method. The original nonlinear elliptic problem is cast into a parabolic form by appending a pseudo transient term $\frac{\partial U}{\partial t}$ to the original set of algebraic equations $F(U) = 0$, and a fully implicit scheme solves (again with Newton method)

$$\mathcal{F}(U^{n+1}) = F(U^{n+1}) + \frac{U^{n+1} - U^n}{\Delta t^{n+1}} = 0, \tag{7}$$

where Δt^{n+1} is the $(n+1)^{\text{st}}$ time step. The number of time steps needed to bring the initial guessed solution into the convergence domain of the steady Newton iteration depends on the size of the grid, and the coarser the grid, the fewer relaxation steps are necessary. This point will be further discussed in §5.2.

4. MULTIGRID TECHNIQUES

The multigrid philosophy applied to our model problem is derived from [5], [7], and [8]. We assume that there is a sequence of spaces $\mathcal{M}_i$, $i = 1, ..., k$, where the $\mathcal{M}_i$ approximate $\mathcal{M}_1$. We further suppose there exist *restriction* and *prolongation* mappings

$$\begin{cases} \mathcal{R}_i : \ \mathcal{M}_i \to \mathcal{M}_{i+1}, & 1 \le i \le k-1, \\ \mathcal{P}_i : \ \mathcal{M}_i \to \mathcal{M}_{i-1}, & 2 \le i \le k. \end{cases}$$

between neighboring spaces. We also assume there is a sequence of problems (5) represented by J_i.

A multilevel correction algorithm, where the finest level is level 1 and the coarsest level is level k, is simply defined by

Algorithm MGC (lev, $\{J_j, x_j, b_j\}_{j=1}^{k}$, $\{\mathcal{P}_j\}_{j=2}^{k}$, $\{\mathcal{R}_j\}_{j=1}^{k-1}$)
1. $x_{lev} \leftarrow Solver_{lev}(J_{lev}, x_{lev}, b_{lev})$
2. If $lev < k$, then repeat 2a–2d until some condition is met:
   2a. $x_{lev+1} \leftarrow 0$, $b_{lev+1} \leftarrow \mathcal{R}_{lev}(b_{lev} - J_{lev} x_{lev})$
   2b. MGC ($lev + 1$, $\{J_j, x_j, b_j\}_{j=1}^{k}$, $\{\mathcal{P}_j\}_{j=2}^{k}$, $\{\mathcal{R}_j\}_{j=1}^{k-1}$)
   2c. $x_{lev} \leftarrow x_{lev} + \mathcal{P}_{lev+1} x_{lev+1}$
   2d. $x_{lev} \leftarrow Solver_{lev}(J_{lev}, x_{lev}, b_{lev})$

In our case, the solver on every level is either Bi-CGSTAB/GS or GMRES/GS. In Step 1 on level k, our stopping criterion was that the linear residual was adequately reduced (see §3). On the other levels, the stopping criteria was either an upper limit on the number of iterations or that the linear residual was adequately reduced.

A common condition in step 2 is to do steps 2a–2d some specified number of times (e.g., 0 for one way multigrid, 1 for a V Cycle, or 2 for a W Cycle). In §5.2, a V Cycle took less overall time than any other choice for a condition in step 2. However, many V Cycles were necessary, starting from the finest level (see the definition of Algorithm NIC below).

Brandt's FAS algorithm [6] is a nonlinear variant of Algorithm MGC. A nonlinear smoother is used in steps 1 and 2d, the actual solution is computed on every level, and corrections are computed before interpolation in step 2c (see [23] for more details).

We use a nested iteration multilevel algorithm since we do not have an adequate initial guess to the solution initially.

Algorithm NIC (lev, $\{J_j, x_j, b_j\}_{j=1}^{k}$, $\{\mathcal{P}_j\}_{j=2}^{k}$, $\{\mathcal{R}_j\}_{j=1}^{k-1}$)
1. MGC (k, $\{J_j, x_j, b_j\}_{j=1}^{k}$, $\{\mathcal{P}_j\}_{j=2}^{k}$, $\{\mathcal{R}_j\}_{j=1}^{k-1}$)
2. Do steps 2a–2b with $lev = k-1, \cdots, 1$:
   2a. $x_{lev} \leftarrow \mathcal{P}_{lev+1} x_{lev+1}$
   2b. MGC (lev, $\{J_j, x_j, b_j\}_{j=1}^{k}$, $\{\mathcal{P}_j\}_{j=2}^{k}$, $\{\mathcal{R}_j\}_{j=1}^{k-1}$)

A *damped Newton multilevel algorithm* is defined by introducing an additional step before each reference to Algorithm MGC in just Algorithm NIC. Before each reference to Algorithm MGC, a Jacobian is formed and a damped Newton step is performed. The last Jacobian on a level is saved for use in multilevel correction steps. A *one way multilevel algorithm* means that Algorithm MGC never performs any portion of its step 2 as part of its use by Algorithm NIC. We always use a damped Newton iteration, but we drop the term damped Newton when referring to one way multilevel methods.

The difference between FAS and damped Newton multilevel methods is easy to categorize. FAS uses a nonlinear iterative method (e.g., nonlinear Gauss-Seidel) while damped Newton uses standard linear solvers. When evaluating the nonlinear function is inexpensive, FAS usually produces an approximate solution faster than the damped Newton multilevel method. However, when the function evaluations are expensive, the damped Newton multilevel method usually produces an approximate solution faster than FAS. In a typical diffusion flame problem with finite rate chemistry [9], the function evaluations are horrendously expensive, so we did not explore FAS. For a flame sheet problem solved using FAS, see [24].

5. NUMERICAL RESULTS

In this section, we present several numerical results obtained on an IBM RISC System/6000 (model 560). In §5.1, we focus on unigrid calculations and emphasize the importance of the scale factors α_l in (6) in order to appropriately monitor the convergence of the outer damped Newton iteration. Our numerical experiments show that the overall execution times can be decreased by up to an order of magnitude by taking a large scale factor for all of the vorticity corrections in the computational domain. The execution times can be decreased by an additional factor of six and ten by combining the unigrid numerical procedure with damped Newton multilevel iterations, using either one way or correction schemes, respectively. The corresponding numerical results are presented in §5.2.

5.1. Unigrid tests

In this section, we discuss the influence of the scale factors α_l in (6) on the whole convergence history of the numerical solution. By modifying these scale factors, we shift the balance of work required in the outer Newton iteration and in the inner linear iterations between the different degrees of freedom present in the system. In particular, a large scale factor for the vorticity component asks for less accuracy in the computed vorticity corrections that are brought back to the Newton iteration, thus reducing considerably the amount of work at each Newton step. As indicated in our numerical experiments, this does not yield any loss of accuracy for the other components of the numerical solution (the radial and axial velocity and the conserved scalar). Another important consequence is that much larger time steps can be taken, even at the beginning of the pseudo transient process when the solution is approximated with a very "coarse" initial guess. Furthermore, only a few time steps are required (typically 20) before the numerical solution already lies in the convergence domain of the steady Newton iteration (5). With lower scale factors for the vorticity, most of the CPU time is spent during the pseudo transient iterations, since much

smaller time steps need to be taken and the convergence domain of the iteration scheme (5) becomes much narrower. Our numerical experiments indicate that a scale factor for the vorticity of 10^3 can yield savings in CPU time of up to an order of magnitude without altering the velocity and temperature profiles of the numerical solution.

5.2. Multigrid acceleration

In this section, we present further improvements in the total execution times obtained by combining the numerical procedure described in §3 and §5.1 with damped Newton multilevel iterations, using either one way or correction schemes. In all of the results, the speedups represent ratios of CPU times.

We consider the finest level to be a 129×161 grid and we construct three additional coarser grids by successively discarding every other node from one grid to the coarser one. This yields a coarsest grid of 17×21 points. It is worthwhile to note that the use of even coarser grids in these problems meets with difficulties since the calculated flame speeds become excessively large due to the influence of numerical diffusion and/or conduction (see [25]) and the Newton iteration (5) fails to converge.

In the one way nonlinear multigrid approach, we solve the nonlinear problem $F(U) = 0$ in one cycle, starting at the coarsest level and ending at the finest. Asymptotically, as the mesh spacing approaches zero, the interpolant of the computed solution on one grid lies in the convergence domain of Newton method on the next finer grid [26]. In our numerical calculations, this was found to be the case for all levels considered, when using either cubic or linear interpolation between levels. As a consequence, the pseudo transient process needs only to be performed on the coarsest level, in order to bring the initial guess into the convergence domain of the steady Newton iteration on this level. This procedure is particularly attractive for two reasons:

1. By time stepping on the coarsest level, we reduce considerably the amount of work spent in the pseudo transient phase.

2. On coarser grids, less computer time is needed to solve (5).

The first set of numerical experiments was performed using Bi-CGSTAB/GS as the linear smoother. The numerical results obtained during the pseudo transient phase are presented in Table 2. On our workstation, the time stepping requires 15 seconds on the coarsest level as opposed to over 40 minutes on the finest, thus yielding a speedup of 166. Table 3 breaks down the numerical results for the steady state Newton iterations. Note that the CPU time spent during the pseudo transient process has been included in the computation of the speedups presented in Table 3. A speedup of a factor of four is achieved using the one way nonlinear multigrid on two levels, which is due to the significant decrease of smoothing steps done on the finest level. With three and four levels, we obtained speedups of 5.4 and 5.8, respectively. The four level multigrid improves only marginally the execution times, since it decreases the CPU time spent on the third level, while most of the work is already concentrated in the smoothing iterations on the finest level. Finally, it is interesting to note that linear interpolation between levels yields lower execution times than cubic interpolation when Bi-CGSTAB/GS is used as the linear smoother.

We also implemented the one way nonlinear multigrid algorithm using GMRES/GS as the linear smoother with 25 Krylov vectors. This requires 15 Mb of additional storage for the Krylov space.

Table 2: Numerical results for one way nonlinear multigrid during the pseudo transient phase with Bi-CGSTAB/GS as the linear smoother

Operation	Levels			
	1	2	3	4
BiCGSTAB/GS iterations	634	352	217	160
Speedup in time	1.0	6.6	34.6	166.0

Table 3: Numerical results for one way nonlinear multigrid

Operation	Levels			
	1	2	3	4
smooth(1)	1632	371	384	378
smooth(2)	–	723	390	380
smooth(3)	–	–	326	346
smooth(4)	–	–	–	192
Speedup in time	1.0	4.2	5.4	5.8

Smooth(i) represents the total number of Bi-CGSTAB/GS steps done on level i during the steady state Newton iterations.

We found in our numerical experiments that the use of cubic interpolation between levels yielded lower execution times than linear interpolation and that it was more efficient to adaptively increase the time step slightly faster during the pseudo transient phase with respect to the Bi-CGSTAB/GS calculations. The numerical results are given in Tables 4 and 5. We obtain a speedup of 160 for the pseudo transient phase on four levels. As indicated in Table 5, the total execution times delivered are greater than the ones obtained with Bi-CGSTAB/GS. This latter algorithm seems therefore to be a preferrable linear smoother when using one way nonlinear multigrid. Note also that the unigrid calculation fails to converge since GMRES/GS stagnates.

In order to solve the linear systems more efficiently, especially the one on the finest level, we perform damped Newton multilevel iterations, making use of the Jacobians computed on all levels coarser than the current one (see algorithm MGC in §4 for more details). The numerical results

Table 4: Numerical results for one way nonlinear multigrid during the pseudo transient phase with GMRES/GS as the linear smoother

Operation	Levels		
	2	3	4
GMRES/GS iterations	572	367	258
Speedup in time	7.2	34.6	159.6

Table 5: Numerical results for one way nonlinear multigrid

Operation	Levels		
	2	3	4
smooth(1)	530	945	945
smooth(2)	1559	592	590
smooth(3)	–	481	825
smooth(4)	–	–	161
Speedup in time	3.2	4.2	4.2

Smooth(i) represents the total number of GMRES/GS steps done on level i during the steady state Newton iterations. The speedups are with respect to the unigrid solution time in Table 3.

Table 6: Numerical results for damped Newton multilevel iterations

Operation	Levels			
	1	2	3	4
smooth(1)	1632	238	268	243
smooth(2)	–	1096	645	673
smooth(3)	–	–	861	1243
smooth(4)	–	–	–	799
Speedup in time	1.0	4.8	6.2	6.6

Smooth(i) represents the total number of Bi-CGSTAB/GS steps done on level i during the steady state Newton iterations.

presented in Table 6 are obtained using 30 steps of Bi-CGSTAB/GS as the linear smoother, which may seem at first glance to be an excessive number of iterations. We obtain a speedup of 6.6 when using four levels. A comparison of Tables 3 and 6 shows that the balance of smoothing iterations is shifted towards the coarsest levels when using damped Newton multilevel iterations, thus yielding lower execution times (approximately 12%) than the ones obtained with the one way nonlinear multigrid. However, it is worthwhile to point out that this improvement comes at the expense of storage since the one way nonlinear multigrid requires 39 Mb and the damped Newton multilevel iterations require up to 62 Mb. This difference is due mainly to the fact that damped Newton multilevel correction methods require saving a Jacobian on every level instead of just one.

Finally, we also performed damped Newton multilevel iterations using GMRES/GS as the linear smoother. In our numerical experiments, we found that the choice of 25 Krylov vectors delivered lower execution times than 20 or 30. We also used cubic and linear interpolation in algorithm NIC and MGC, respectively (see §4). The numerical results are presented in Table 7. We obtain a speedup of a factor of 10.5 when using four levels, thus significantly improving the maximum speedup obtained with Bi-CGSTAB/GS. Using damped Newton multilevel iterations and GMRES/GS as the linear smoother, the whole numerical solution for the flame sheet problem on a 129×161 grid is obtained in about 9 minutes on our workstation. On a supercomputer, the CPU

Table 7: Numerical results for damped Newton multilevel iterations

Operation	Levels		
	2	3	4
smooth(1)	218	216	219
smooth(2)	2272	565	585
smooth(3)	–	1179	1159
smooth(4)	–	–	1020
Speedup in time	5.1	9.9	10.5

Smooth(i) represents the total number of GMRES/GS steps done on level i during the steady state Newton iterations. The speedups are with respect to the unigrid solution time in Table 3.

times will drop dramatically.

6. CONCLUSIONS

In this paper, we presented a new numerical procedure to solve flame sheet problems. The governing equations use the vorticity-velocity formulation of the Navier-Stokes equations coupled together with a conserved scalar equation. By appropriately monitoring the norm of the correction vector in the damped Newton iteration, significant savings in the overall execution time can be obtained. These performances can be further improved by combining the above numerical procedure with one way nonlinear multigrid and damped Newton multilevel iterations. The latter approach yields lower execution times than the former but at a higher cost in storage. With four levels of grids, a speedup of 5.8 is obtained with a one way nonlinear multigrid and Bi-CGSTAB/GS as the linear smoother. Similarly, damped Newton multilevel iterations and GMRES/GS as the linear smoother obtain a speedup of more than a factor of 10. For three dimensional problems, we should obtain speedups much greater than 10.

REFERENCES

[1] M. D. Smooke and V. Giovangigli. Numerical modeling of axisymmetric laminar diffusion flames. *IMPACT Comput. Sci. Engng.*, 4:46–79, 1992.

[2] M. D. Smooke, R. E. Mitchell, and D. E. Keyes. Numerical solution of two-dimensional axisymmetric laminar diffusion flames. *Combust. Sci. and Tech.*, 67:85–122, 1989.

[3] Y. Xu and M. D. Smooke. Application of a primitive variable Newton's method for the calculation of an axisymmetric laminar diffusion flame. *J. Comput. Phys.*, 104:99–109, 1993.

[4] Y. Xu and M. D. Smooke. Primitive variable modeling of multidimensional laminar flames. *Combust. Sci. and Tech.*, 88:1–25, 1993.

[5] R. E. Bank and D. J. Rose. Analysis of a multilevel iterative method for nonlinear finite element equations. *Math. Comp.*, 39:453–465, 1982.

[6] A. Brandt. Multi–level adaptive solution to boundary–value problems. *Math. Comp.*, 31:333–390, 1977.

[7] C. C. Douglas. Multi–grid algorithms with applications to elliptic boundary–value problems. *SIAM J. Numer. Anal.*, 21:236–254, 1984.

[8] C. C. Douglas and J. Douglas. A unified convergence theory for abstract multigrid or multilevel algorithms, serial and parallel. *SIAM J. Numer. Anal.*, 30:136–158, 1993.

[9] A. Ern, C. C. Douglas, and M. D. Smooke. Numerical simulation of laminar diffusion flames with multigrid methods. In preparation.

[10] D. E. Keyes and M. D. Smooke. Flame sheet starting estimates for counterflow diffusion flame problems. *J. Comput. Phys.*, 73:267–288, 1987.

[11] S. W. Armfield. Finite difference solutions of the Navier-Stokes equations on staggered and non-staggered grids. *Computer Fluids*, 20:1–17, 1991.

[12] F. Sotiropoulos and S. Abdallah. A primitive variable method for the solution of three-dimensional incompressible viscous flows. *J. Comput. Phys.*, 103:336–349, 1992.

[13] Wei Shyy and Chia-Sheng Sun. Development of a pressure-correction/staggered-grid based multigrid solver for incompressible recirculating flows. *Computer Fluids*, 22:51–76, 1993.

[14] Z. Zhu and C. A. J. Fletcher. A study of sequential solutions for the reduced/complete Navier-Stokes equations with multigrid acceleration. *Computer Fluids*, 19:43–60, 1991.

[15] T. B. Gatski. Review of incompressible fluid flow computations using the vorticity-velocity formulation. *Appl. Numer. Math.*, 7:227–239, 1991.

[16] A. Ern, V. Giovangigli, D. E. Keyes, and M. D. Smooke. Towards polyalgorithmic linear system solvers for nonlinear elliptic problems. *SIAM J. Sci. Comput.*, 15:to appear, 1994. Also available as Yale University Department of Mechanical Engineering Research Report ME–101–93, New Haven, CT, March, 1993.

[17] A. Ern and M. D. Smooke. Vorticity-velocity formulation for three-dimensional steady compressible flows. *J. Comput. Phys.*, 105:58–71, 1993.

[18] M. Napolitano and L. A. Catalano. A multigrid solver for the vorticity-velocity Navier-Stokes equations. *Int. J. Numer. Methods Fluids*, 13:49–59, 1991.

[19] Y. Xu. *Numerical calculations of an axisymmetric laminar diffusion flame with detailed and reduced reaction mechanisms.* PhD thesis, Yale University, December 1991. Mechanical Engineering Department.

[20] G. A. Sod. *Numerical methods in fluid dynamics.* Cambridge Univ. Press, London, 1985.

[21] H. A. Van der Vorst. Bi-CGSTAB: a fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems. *SIAM J. Sci. Stat. Comput.*, 13:631–644, 1992.

[22] Y. Saad and M. H. Schultz. GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM J. Sci. Stat. Comput.*, 7:856–869, 1986.

[23] C. C. Douglas. Implementing abstract multigrid or multilevel methods. Technical Report YALEU/DCS/TR-952, Department of Computer Science, Yale University, New Haven, 1993.

[24] C. Liu, Z. Liu, and S. F. McCormick. Multigrid methods for numerical simulation of laminar diffusion flames. *AIAA*, 93-0236:1–11, 1993.

[25] M. D. Smooke, J. A. Miller, and R. J. Kee. On the use of adaptive grids in numerically calculating adiabatic flame speeds. In N. Peters and J. Warnatz, editors, *Numerical Methods in Laminar Flame Propagation*, pages 65–70. Friedr. Vieweg & Sohn, Braunschweig, 1982.

[26] M. D. Smooke and R. M. M. Mattheij. On the solution of nonlinear two-point boundary value problems on successively refined grids. *Applied Numer. Math.*, 1:463–487, 1985.