MULTIGRID SOLUTION OF FLAME SHEET PROBLEMS ON SERIAL AND PARALLEL COMPUTERS*

CRAIG C. DOUGLAS†, ALEXANDRE ERN‡ AND MITCHELL D. SMOOKE§

Abstract. Flame sheet problems are on the natural route to the numerical solution of detailed chemistry, laminar diffusion 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 more traditional stream function-vorticity approach. The numerical solution of the resulting nonlinear coupled elliptic partial differential equations involves damped Newton iterations, adaptive grid procedures, and multigrid methods. We focus on nonlinear damped Newton multigrid, using either one way or correction schemes. Results on serial and parallel processors are presented.

Key words. multigrid, combustion, flame sheet, Navier-Stokes, vorticity-velocity, nonlinear methods, iterative methods, parallel computing.

AMS(MOS) subject classifications. 80A32, 80-08, 65C20, 65N20, 65F10.

1. Introduction. 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 involve temperature and fluid dynamics variables, dozens of chemical 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 equations are strongly coupled together and are also characterized by the presence of stiff source terms and nonlinearities. 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. 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 combined with the numerical solution of combustion related problems. The solutions are calculated on cheap, but fast workstations as well as on small parallel computers.

In the present work, we consider a flame sheet problem [5] rather than a finite rate chemistry model for an axisymmetric laminar diffusion flame. This alleviates the memory and CPU requirements on the computer simulations. The numerical techniques presented in this paper, however, also apply to combustion problems with

* This work is supported in part by ONR and DOE Office of Energy Sciences.

† IBM Research Division, Thomas J. Watson Research Center, P. O. Box 218, Yorktown Heights, NY 10598-0218, USA and Department of Computer Science, Yale University, P. O. Box 208285, New Haven, CT 06520-8285, USA.

‡ Department of Mechanical Engineering, Yale University, P. O. Box 208286, New Haven, CT 06520-8286, USA and CERMICS, ENPC, La Courtine, 93167, Noisy-le-Grand Cedex, FRANCE.

§ Department of Mechanical Engineering, Yale University, P. O. Box 208286, New Haven, CT 06520-8286, USA.

finite rate chemistry [3] [4]. 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 at least 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 many of the nonlinearity features associated with the original flame problem.

In the flame sheet model [5], 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. By introducing Schvab-Zeldovich variables, one can derive a source free convective-diffusive equation for a single conserved scalar. The temperature and the stable major species profiles can be obtained from the solution of the conserved scalar equation coupled to the flow field equations. 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. Finally, flame sheets are routinely employed to initialize multidimensional diffusion flames.

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 versus 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 in order.

There have been three formulations of the Navier-Stokes equations used recently for steady, low Mach number flame problems:

- stream function–vorticity
- primitive variables
- vorticity–velocity

Stream function–vorticity reduces the number of equations by one. However, accurate vorticity boundary values in terms of the stream function are hard to specify and result in severely ill conditioned Jacobians. Furthermore, the formulation does not extend to three dimensions in a reasonable manner. The primitive form of the Navier–Stokes equations is easily extendible to three dimensions and unsteady problems, allows for accurate boundary conditions, but generally requires a staggered grid to discretize the governing equations. However, a staggered grid is a problem in complex geometries such as a three dimensional curvilinear system. Vorticity–velocity allows for more accurate vorticity boundary conditions than stream function–vorticity, yields better conditioned Jacobians, uses nonstaggered grids, and extends easily to three dimensional problems [3].

A detailed formulation of the vorticity–velocity problem can be found in [2] and [3]. Define the velocity vector $v = (v_r, v_z)$ with radial (v_r) and axial (v_z) components. The normal component of the vorticity is $\omega = \frac{\partial}{\partial z} v_r - \frac{\partial}{\partial r} v_z$. 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 ω and using the continuity equation.

Let ρ be the density, μ the viscosity, g the gravity vector, $\mathrm{div}(v)$ the cylindrical divergence of the velocity vector, S the conserved scalar, and D a diffusion coefficient. The components of $\overline{\nabla}\beta$ are $(\frac{\partial}{\partial z}\beta, -\frac{\partial}{\partial r}\beta)$. Then the governing equations are the following:

$$
(1) \qquad \left\{
\begin{array}{l}
\dfrac{\partial^2 v_r}{\partial r^2} + \dfrac{\partial^2 v_r}{\partial z^2} = \dfrac{\partial \omega}{\partial z} - \dfrac{1}{r}\dfrac{\partial v_r}{\partial r} + \dfrac{v_r}{r^2} - \dfrac{\partial}{\partial r}\left(\dfrac{v \cdot \nabla \rho}{\rho}\right), \\[2ex]
\dfrac{\partial^2 v_z}{\partial r^2} + \dfrac{\partial^2 v_z}{\partial z^2} = -\dfrac{\partial \omega}{\partial r} - \dfrac{1}{r}\dfrac{\partial v_r}{\partial z} - \dfrac{\partial}{\partial z}\left(\dfrac{v \cdot \nabla \rho}{\rho}\right), \\[2ex]
\dfrac{\partial^2 \mu\omega}{\partial r^2} + \dfrac{\partial^2 \mu\omega}{\partial z^2} + \dfrac{\partial}{\partial r}\left(\dfrac{\mu\omega}{r}\right) = \\[2ex]
\qquad\qquad \rho v_r \dfrac{\partial \omega}{\partial r} + \rho v_z \dfrac{\partial \omega}{\partial z} - \dfrac{\rho v_r}{r}\omega + \overline{\nabla}\rho \cdot \nabla \dfrac{v^2}{2} - \overline{\nabla}\rho \cdot g + \\[2ex]
\qquad\qquad 2\left(\overline{\nabla}(\mathrm{div}(v)) \cdot \nabla\mu - \nabla v_r \cdot \overline{\nabla}\dfrac{\partial \mu}{\partial r} - \nabla v_z \cdot \overline{\nabla}\dfrac{\partial \mu}{\partial z}\right), \\[2ex]
\dfrac{1}{r}\dfrac{\partial}{\partial r}(r\rho D\dfrac{\partial S}{\partial r}) + \dfrac{\partial}{\partial z}(\rho D\dfrac{\partial S}{\partial z}) = \rho v_r \dfrac{\partial S}{\partial r} + \rho v_z \dfrac{\partial S}{\partial z}.
\end{array}
\right.
$$

The temperature and major species profiles are recovered from the conserved scalar S as detailed, for instance, in [6]. The density is computed using the perfect gas law and, in the low Mach number approximation valid for these flame configurations, one can use the outlet (constant) pressure. Hence, the pressure field is eliminated from the governing equations as a dependent unknown and can be recovered, once a computed numerical solution of (1) is obtained, by solving a Laplace type equation derived by taking the divergence of the momentum equations.

A schematic of the physical configuration is given in Figure 1. It consists of an inner cylindrical fuel jet (radius $R_I = 0.2$cm), an outer coflowing annular oxidizer jet (radius $R_O = 2.5$cm), and a dead zone extending to $R_{max} = 7.5$cm. In the present study, the fuel is methane and the oxidizer is air. The inlet velocity profile of the fuel and oxidizer jets is a plug flow of 35cm/s, and the inlet velocity is zero in the dead zone. Furthermore, the flame length is approximately $L_f = 3$cm and the length of the computational domain is set to $L = 30$cm. Although the fuel and oxidizer reservoirs are at room temperature (300°K), we need to assume 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. The boundary conditions are summarized as follows.

- Axis of symmetry ($r = 0$): $v_r = 0$, $\dfrac{\partial v_z}{\partial r} = 0$, $\omega = 0$, and $\dfrac{\partial S}{\partial r} = 0$.
- Outer zone ($r = R_{max}$): $\dfrac{\partial v_r}{\partial r} = 0$, $\dfrac{\partial v_z}{\partial r} = 0$, $\omega = \dfrac{\partial v_r}{\partial z}$, and $S = 0$.

- Inlet ($z=0$): $v_r = 0$, $v_z = v_z^0(r)$, $\omega = \dfrac{\partial v_r}{\partial z} - \dfrac{\partial v_z}{\partial r}$, and $S = S^0(r)$.
- Exit ($z=L$): $v_r = 0$, $\dfrac{\partial v_z}{\partial z} = 0$, $\dfrac{\partial \omega}{\partial z} = 0$, and $\dfrac{\partial S}{\partial z} = 0$.

3. General Solution Algorithm. The partial differential equations (1) together with the boundary conditions 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, 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 (1) 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. The boundary conditions 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 (see [2] and [3] for more details). The vorticity inlet boundary condition is discretized using the vorticity values at the first two lines of the computational domain. 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.

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

$$J(U^n)\Delta U^n = -\lambda^n F(U^n), \qquad 0 < \lambda^n \leq 1, \qquad n = 0, 1, \ldots, \tag{2}$$

with convergence tolerance $\|\Delta U^n\| < 10^{-5}$. 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 [6]. We introduce scale factors α_l, $l \in [1, 4]$, for each dependent variable. These determine how much each component is weighted in the norm above. It is worthwhile to point out that an appropriate choice of the scale factors can yield significant savings in the execution time [3].

At each Newton step, (2) is solved approximately using either Bi-CGSTAB or a restarted version of GMRES combined with a Gauss-Seidel (GS) left preconditioner. The convergence of this 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. Computational experience shows that such termination criterion brings enough information on the update vector ΔU^n back to the Newton iteration (see [3] for more details).

Due to the nonlinearity of the original problem, a traditional time relaxation process is used to produce a parabolic in time problem and bring the starting estimate into the convergence domain of the Newton iteration (2). The original nonlinear

TABLE 1
Numerical results for one way nonlinear multigrid during the time relaxation phase

Linear solver	Operation	Levels			
		1	2	3	4
Bi-CGSTAB/GS	CPU minutes for 20 time steps	41.5	6.3	1.2	0.25
	Speedup in time	1.0	6.6	34.6	166.0
GMRES/GS	CPU minutes for 20 time steps	—	5.8	1.2	0.26
	Speedup in time	—	7.2	34.6	159.6

The speedups are with respect to the unilevel solution time (41.5mn).

TABLE 2
Numerical results for one way nonlinear multigrid

Linear solver	Operation	Levels			
		1	2	3	4
Bi-CGSTAB/GS	Total CPU minutes	96.2	22.9	17.7	16.5
	Speedup in time	1.0	4.2	5.4	5.8
GMRES/GS	Total CPU minutes	—	29.7	23.0	23.1
	Speedup in time	—	3.2	4.2	4.2

The speedups are with respect to the unilevel solution time (96.2mn).

elliptic problem is transformed into

$$F(U^{n+1}) + \mathcal{D}\frac{U^{n+1} - U^n}{\Delta t^{n+1}} = 0, \tag{3}$$

where $\mathcal{D}$ is a diagonal scaling matrix and Δt^{n+1} is the $(n+1)^{\text{st}}$ time step. A fully implicit scheme solves (3), again with a damped Newton method. The number of time steps needed to bring the initial guessed solution into the convergence domain of the Newton iteration (2) depends on the size of the grid, and the coarser the grid, the fewer time steps are necessary.

4. Numerical Results. In this section, we present several numerical results obtained on an IBM RISC System/6000 (model 560). We use both *one way multigrid* and *damped Newton multigrid* methods [1]. In one way multigrid methods, no correction problem is ever solved: a coarse grid problem is solved and the solution is interpolated onto the next finer grid as an initial guess. The process is repeated recursively until the problem on the finest grid is approximated well enough. Damped Newton multigrid methods also start on a coarse mesh. However, the Jacobians are saved for use in linear correction problems. See [2] for more details.

Asymptotically, as the mesh spacing approaches zero, the interpolant of the computed solution on one grid lies in the convergence domain of the Newton method on the next finer grid. 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 time relaxation process needs only to be performed on the coarsest level. This procedure reduces considerably the amount of work spent in the time relaxation phase, as reflected in Table 1.

Table 2 and Table 3 contain the results for the one way multigrid and damped Newton multigrid methods. In these tables, the CPU time refers to the total time

TABLE 3
Numerical results for damped Newton multilevel iterations

Linear solver	Operation	Levels 1	2	3	4
Bi-CGSTAB/GS	Total CPU minutes	96.2	20.4	15.7	14.7
	Speedup in time	1.0	4.8	6.2	6.6
GMRES/GS	Total CPU minutes	—	19.0	9.7	9.2
	Speedup in time	—	5.1	9.9	10.5

The speedups are with respect to the unilevel solution time (96.2mn).

TABLE 4
Numerical results for parallel iterations on an IBM SP2

One way multilevel					
Linear solver	Operation	Processors 1	2	4	8
Bi-CGSTAB/GS	Total CPU minutes	4.26	2.30	1.26	0.77
	Speedup	1.0	1.8	3.4	5.5
GMRES/GS	Total CPU minutes	6.32	3.31	1.84	1.09
	Speedup	1.0	1.9	3.4	5.8
Damped Newton multilevel					
Bi-CGSTAB/GS	Total CPU minutes	4.57	2.58	1.20	0.77
	Speedup	1.0	1.8	3.8	5.9
GMRES/GS	Total CPU minutes	5.20	2.59	1.35	0.86
	Speedup	1.0	2.0	3.8	6.0

Four levels were used in each run.

needed to solve (2), including the 20 preliminary time steps. The damped Newton multigrid is significantly faster than the one way 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.

We reran a subset of the multigrid examples on an IBM SP2 parallel computer. The parallelization is done by using a sparse matrix domain decomposition (i.e., treating the matrix as a domain). This minimized the communication during the iterative solves. The Jacobians and nonlinear residuals are computed in a similar, parallel fashion (see [4] for more details). Table 4 contains the results (for Bi-CGSTAB/GS, the number of linear iterations for correction cycles on the finest level was increased 10% over the serial case). We note that the preconditioner is local so that the number of iterations of either Bi-CGSTAB/GS or GMRES/GS to reduce the residual norm by a fixed amount is now also a function of the number of processors. As reflected in Table 4, we obtain parallel efficiencies in the range 70%–80% using 8 processors. Higher efficiencies should be obtained for detailed chemistry diffusion flame problems where most of the CPU time is spent in the time relaxation phase evaluating Jacobians [4].

5. Conclusions. In this paper, we presented results for a new numerical procedure to solve flame sheet problems. The governing equations use the vorticity-velocity

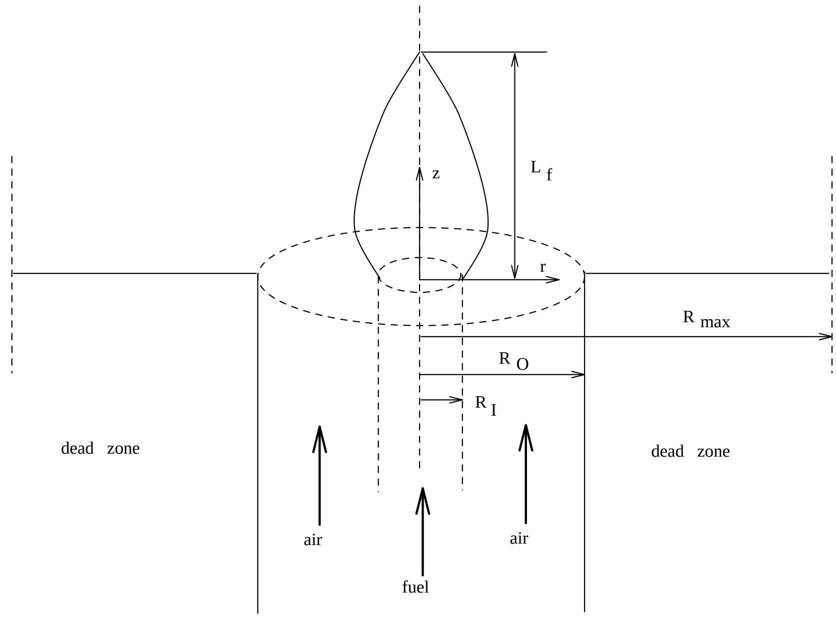

FIG. 1. *Physical configuration* (*not in scale*)

formulation of the Navier-Stokes equations coupled together with a conserved scalar equation. The numerical solution method uses one way nonlinear multigrid and damped Newton multilevel iterations on *nonstaggered* grids. The latter approach yields lower execution times than the former but at a higher cost in storage.

When this research began, a very fast workstation (for its time) took nearly 100 CPU minutes to solve the unilevel problem. By using a new model, combined with better numerical algorithms and improved technologies, the time has been reduced to about a CPU minute. We note that for three dimensional problems, we should obtain speedups much greater than those seen in this paper.

REFERENCES

[1] R. E. BANK AND D. J. ROSE, *Analysis of a multilevel iterative method for nonlinear finite element equations*, Math. Comp., 39 (1982), pp. 453–465.

[2] C. C. DOUGLAS AND A. ERN, *Numerical solution of flame sheet problems with and without multigrid methods*, in Sixth Copper Mountain Conference on Multigrid Methods, N. D. Melson, T. A. Manteuffel, and S. F. McCormick, eds., vol. CP 3224, Hampton, VA, 1993, NASA, pp. 143–157.

[3] A. ERN, *Vorticity-velocity modeling of chemically reacting flows*, PhD thesis, Yale University, February 1994. Mechanical Engineering Department.

[4] A. ERN, C. C. DOUGLAS, AND M. D. SMOOKE, *Detailed chemistry modeling of laminar diffusion flames on parallel computers*, tech. report, Department of Mechanical Engineering, Yale University, New Haven, 1994. In preparation.

[5] R. E. MITCHELL, A. F. SAROFIM, AND L. A. CLOMBURG, *Experimental and numerical investigation of confined laminar diffusion flames*, Combust. and Flame, 37 (1980), pp. 227–244.

[6] M. D. SMOOKE, R. E. MITCHELL, AND D. E. KEYES, *Numerical solution of two-dimensional axisymmetric laminar diffusion flames*, Combust. Sci. and Tech., 67 (1989), pp. 85–122.