Naval Research Laboratory
Washington, DC 20375-5320

NRL/MR/6180--02-8627

Numerical Simulation of Solid Combustion with a Robust Conjugate-Gradient Solution for Pressure

YI-ZUN J. WANG

Potomac State College of West Viriginia University
Keyser, WV

RAMAGOPAL ANANTH
PATRICIA A. TATEM

Navy Technology Center for Safety and Survivability
Chemistry Division

July 31, 2002

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
July 31, 2002	Memorandum Report	October 1, 1999-1 June 2002

4. TITLE AND SUBTITLE

Numerical Simulation of Solid Combustion with a Robust Conjugate-Gradient Solution for Pressure

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER
62123N

6. AUTHOR(S)

Yi-Zun J. Wang,* Ramagopal Ananth, and Patricia A. Tatem

5d. PROJECT NUMBER
MA-123-001-6001

5e. TASK NUMBER
MA-123-001-6001

5f. WORK UNIT NUMBER
MA-123-001-6001

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Naval Research Laboratory, Code 6180
4555 Overlook Avenue
Washington, DC 20375-5320

8. PERFORMING ORGANIZATION REPORT NUMBER

NRL/MR/6180--02-8627

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Office of Naval Research
800 North Quincy Street
Arlington, VA 22217-5000

10. SPONSOR / MONITOR'S ACRONYM(S)

11. SPONSOR / MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION / AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

*Potomac State College of West Virginia University, Keyser, WV

14. ABSTRACT

A Bi-Conjugate Gradient method (Bi-CGSTAB) is studied and tested for solid combustion in which the gas and solid phases are coupled by a set of conditions describing mass, momentum and heat transport across the interface. The conjugate gradient method is found to yield robust solutions for pressure and lead to accurate prediction of the burning rates along the length of the polymer plate. Specifically, the conjugate gradient method is significantly simpler with faster convergence that the multi-grid method (MGID) for a given accuracy in the predicted burning rates.

15. SUBJECT TERMS

Solid combustion, Conjugate-gradient methods, Multi-grid method, Burn rate

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Dr. Ramagopal Ananth
Unclassified	Unclassified	Unclassified	UL	25	19b. TELEPHONE NUMBER *(include area code)* (202) 767-3197

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std. Z39.18

CONTENTS

NOMENCLATURE

Text

A	coefficient matrix
a1...a5	coefficients
c0	coefficient matrix of dependent variable in the finite-volume discretization
c1	coefficient matrix in x direction of the differential term in finite-volume equation
c2	coefficient matrix in y direction of the differential term in finite-volume equation
E	total energy density of gas mixture, ergs/cm^3
f	source term vector
H	height, cm
J	molecular diffusive flux, gmole/cm^2sec
L	length, cm
P	pressure, dynes/cm^2
δP	change in pressure, dynes/cm^2
q	conductive heat flux, ergs/cm^2sec
r,s	residual vectors
T	temperature, K
t	time, sec.
u	dependent variable scaler
u	dependent variable vector
v	velocity vector
W	rate of combustion, gmole/cm^3sec
x	horizontal axial coordinate
y	vertical axial coordinate

Greek Symbols

ρ	gas density, kg/m^3
ε	energy density rate, ergs/cm^3-sec.
γ	specific heat ratio
τ	stress tensor
ΔH_c	heat of combustion
ω	implicitness parameter

Subscripts

i	index in x-direction
j	index in y-direction
k	k th specie
r	radiative component

Superscripts

n	new time step
o	previous time step

NUMERICAL SIMULATION OF SOLID COMBUSTION WITH A ROBUST CONJUGATE-GRADIENT SOLUTION FOR PRESSURE

1. INTRODUCTION

Solid materials, such as the insulation on bulkheads and doors in a ship, often form boundary layer flames due to no-slip and pyrolysis of the solid. The burning rate plays a key role in the evaluation of fire threat and depends on both the fluid dynamics near the surface and the intrinsic burning characteristics of the solid phase. A numerical model was developed by Ananth et al. [1-4] to describe temperature and local burning rate distributions along the length of a polymer (poly methyl methacrylate, PMMA) slab. Time dependent solutions of full Navier-Stokes equations are obtained by using Barely Implicit Correction to Flux Corrected Transport (BIC-FCT) algorithms for combustion of a PMMA plate under forced flow of air past the plate. A multi-variable fixed point (MVFP) iterative method is developed to describe the coupling between the gas and the solid phases.

In BIC-FCT schemes, an implicit elliptic partial differential equation for pressure arises from the momentum and energy equations as described by G. Patnaik et al. [5]. In the simulation of solid combustion, the elliptic equation must be solved at every MVFP iteration and at each time step. Since the solution of the pressure equation is of significant computational cost, a robust method is needed. The objective of this work is to implement a bi-conjugate gradient method for the solution of the pressure equation and predict the burning rate distributions along the polymer slab. We will show that a bi-conjugate gradient method (Bi-CGSTAB) is computationally reliable, efficient, and accurate for slow flow past the solid plate by comparing it with a multigrid method (MGRID), which is used currently. Unlike MGRID, Bi-CGSTAB is simple, transparent, easier to implement with direct accuracy control, and can port among different machines (e.g., SGI unix work station and the SunSPARC cluster).

To understand how the Bi-CGSTAB method works, it is tested using a simple problem, which has an analytical solution prior to the simulation of the combustion problem. Both MGRID and Bi-CGSTAB methods are applied to the solid combustion problem, and the solutions and CPU times compared at steady state (12,000 time steps). The method of Bi-CGSTAB shows favorable features in convergence and accuracy.

2. LITERATURE REVIEW

Numerical modeling of combustion problems often involves the solution of elliptic partial differential equations subjected to complicated boundaries and interfaces. The general form of an elliptic equation is given by

$$a_1 u + a_2 \nabla \bullet a_3 \nabla u + \nabla a_4 u = a_5 \qquad (2.1)$$

Manuscript approved June 25, 2002.

with boundary conditions given by

$$b_1 u + b_2 \frac{\partial u}{\partial n} = b_3 . \qquad (2.2)$$

Here u is a dependent variable, a_1 to a_5 and b_1 to b_3 are coefficients, and n representing normal to the boundary can be x or y.

In general, the elliptic partial differential equation is discretized by a finite volume method. This results in a set of linear algebraic equations, $\mathbf{Au=f}$, where A is the coefficient matrix and **u** and **f** are the vectors containing the dependent variables and source terms, respectively. Since the elliptic partial differential equation is linear, it leads to a linear set of algebraic equations, which may be solved using an iterative algorithm. However, in time dependent formulations, such as BIC-FCT, the coefficients in the elliptic equation are time-dependent. Therefore, the coefficients contained in the matrix A vary with time. As combustion progresses in time, the coefficient matrix may rapidly fill up and become non-symmetric or non-triangular. In these situations, common elliptic solvers, such as a direct method, and the iterative technique ADI (Alternating-Direction Implicit) method, become expensive [6]. In recent years, multigrid methods and conjugate gradient methods have been recognized as feasible methods to solve such elliptic problems to achieve the required accuracy and efficiency.

The simplest of the conjugate gradient methods is based on the idea of minimizing the function

$$\mathbf{S(u)} = (1/2)\ \mathbf{u} \bullet (\mathbf{Au}) - \mathbf{f} \bullet \mathbf{u} \qquad (2.3)$$

The function **S** is minimized when the gradient is zero as shown below

$$\nabla \mathbf{S} = \mathbf{Au} - \mathbf{f} = \mathbf{0} \qquad (2.4)$$

The iterative method uses successive approximations to reach the final solution. Each successive iteration searches for a correction to the previous iteration in a direction normal to the subspace of the most recent approximation, until the entire vector space is searched. The methods work well for those slow flows problems, wherein viscous or diffusive effects are dominant. A typical conjugate gradients method for solving elliptic equations is Conjugate Gradients-Squared method (CG-S) of Sonneveld, et al.[7]. In this method, the residual vectors $\mathbf{r} = \mathbf{Au-f}$ generated by the method satisfy a 3-term recurrence relation. The method has the feature of fast convergence.

Bi-CG methods are developed based on CG-S and do not directly relate to the function minimization concept. These methods have two 3-term recurrence relations for better stability. It is found that CG-S based methods may lead to a rather irregular convergence behavior and in some cases rounding errors can even result in severe cancellation effects in the solution. Van Der Vorst [8] proposed a variant of Bi-CG, named Bi-CGSTAB, to eliminate these negative effects. Van Der Vorst showed that the Bi-CGSTAB provides fast and smoothly convergence

without giving up the attractive speed of convergence of CG-S. G. Patnaik [9] has recently implemented the Bi-CGSTAB method to study 3-dimensional premixed hydrogen-air flames. He also successfully parallelized the code to run on Origin2000 machine. The present work implements the Bi-CGSTAB to the solid combustion problem and is an extension of the work of G. Patnaik [9].

We have been using a multigrid method (MGRID) developed by DeVore [10] to solve the pressure equation. This method for solving the elliptic boundary-value problem was originally found by Douglas [11]. Subsequently, DeVore [10] converted the method to vectorized form for higher speed computations. Basically, the method uses the solution obtained from a sequence of smaller spaces to approximate the desired solution in the largest space. Since the smaller spaces have geometrically fewer unknowns, they require less computational effort to yield a solution than does the largest space. The method combines direct and iterative methods. (It can make use of coarse grids with mesh spacing for two fine-grid points, say 2h as two levels, 4h, 8h, or more levels.) The approximate solution for the fine-grid can be obtained by a direct method, such as Gaussian elimination. In order to improve the approximated solution on the fine grids, the method does a small number of relaxation iterations on the solution obtained from the first approximation. This iteration process is based on the residual vector, and then the desired values from the fine grid are projected onto the coarse grid. This process is a correction process, or a cycle, and can be done by several iteration steps. According to the report from DeVore [10], a level-4 and cycle-2 algorithm might be a good choice. However, he recommended that the best choices for many of these options, in general, depend on the properties of the coefficients in the equation, the number of spatial dimensions, the number of unknowns, and the accuracy required in the solution. Thus, the method is complex and less transparent. Furthermore, for the solid combustion problem, the results reported here will show that it is also time consuming, due to the intergrid transfers and recursive cycling.

3. MATHEMATICAL FORMULATION

We consider air flow past a PMMA plate, which includes a non-burning leading plate and a post flame plate as shown in Figure 1. The plate is ignited near the entire surface by adding external energy for a short time. The fuel vapor generated at the surface mixes with the incoming air by convection and diffusion and undergoes combustion. The development of a boundary layer diffusion flame is described by the time-dependent mass, momentum, total energy, and specie equations, which are given by

continuity

$$\frac{\partial \rho}{\partial t} = -\nabla \bullet \rho v, \qquad (3.1)$$

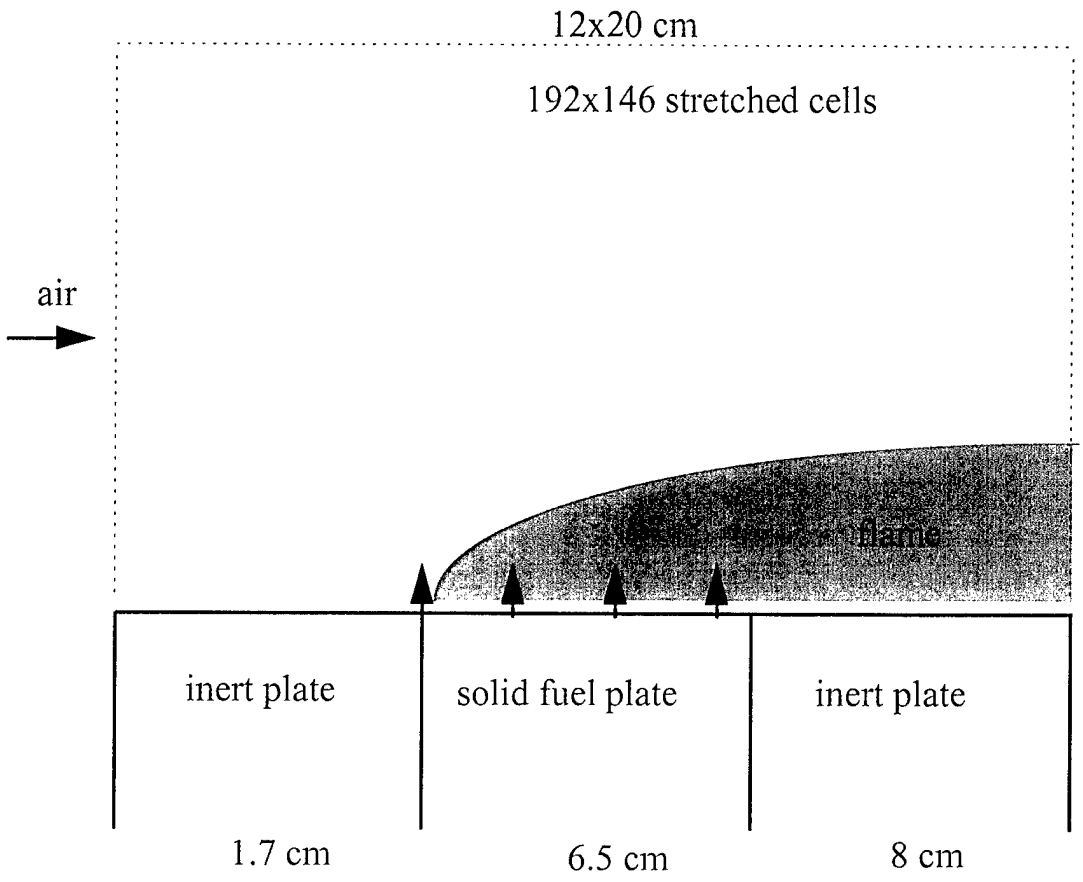

Figure 1 Boundary Layer Combustion of a Solid Fuel Plate

momentum

$$\frac{\partial \rho v}{\partial t} = -\nabla \bullet \rho v v - \nabla P + \nabla \bullet \tau , \qquad (3.2)$$

energy

$$\frac{\partial E}{\partial t} = -\nabla \bullet (E + P) v + \nabla \bullet (v \bullet \tau) + \nabla \bullet q + W_k \Delta H_c + q_r , \qquad (3.3)$$

and specie

$$\frac{\partial C_k}{\partial t} = -\nabla \bullet v C_k + \nabla \bullet J_k + W_k , \qquad (3.4)$$

where the total energy density E is

$$E = \varepsilon + \frac{1}{2} \rho v^2 \quad . \tag{3.5}$$

The pressure P and internal energy density ε are related by the equation of state, which is given by

$$P = (\gamma - 1)\, \varepsilon. \tag{3.6}$$

Here ρ is the gas density, v is the velocity vector,τ is the stress tensor, q is conductive heat flux, W_k is the rate of combustion of specie k, ΔH_c is the heat of combustion, q_r is the radiative heat flux, C_k is the specie concentration, J_k is the molecular diffusive flux, and γ is the ratio of specific heats.

The convective, diffusive, and reaction parts of the equations (3.1) to (3.6) are computed separately and are combined using time step-splitting [12]. The pressure term is included in the convection algorithms, which are subject to the Courant condition in an explicit formulation. The convective part of the equations (3.2) and (3.3) are given by

$$\frac{\partial \rho v}{\partial t} = - \nabla \bullet \rho\, v\, v - \nabla P\,, \tag{3.7}$$

and

$$\frac{\partial E}{\partial t} = - \nabla \bullet (E + P)\, v\,, \tag{3.8}$$

To solve this time-dependent system of convection equations numerically, Casulli and Greenspan [13] showed that it is not necessary to treat every term in a finite-difference algorithm implicitly to avoid the time step constraint imposed by the Courant condition. They showed that only those terms containing the pressure in equation (3.7) and the velocity in equation (3.8) must be treated implicitly. Patnaik et al. [5] applied the concepts of Casulli and Greenspan [13] to obtain solutions for pressure and the velocity in by BIC-FCT formulation. An implicitness parameter, ω, is introduced and it can be vary the degree of implicitness of the algorithm. In general, we can have $0.5 \le \omega \le 1$, where the implicit terms are centered in time for $\omega = 0.5$. For $\omega < 0.5$, the method is found to be unstable for sufficiently large time steps.

The algorithm has two stages. One stage is an explicit predictor that determines the provisional value ρ and v in the following two equations,

$$\frac{\bar{\rho} - \rho^o}{\Delta t} = - \nabla \bullet \rho^o\, v^o \tag{3.9}$$

and

$$\frac{\bar{\rho}\,\bar{v} - \rho^o v^o}{\Delta t} = - \nabla \bullet \rho^o\, v^o\, v^o - \nabla P^o\,. \tag{3.10}$$

The bar denotes predictor values at the new time step, and the superscript o and n are used to denote the values at old time step and new time, respectively, in the correction. The implicit forms of momentum and energy equations are

$$\frac{\rho^n v^n - \rho^o v^o}{\Delta t} = -\nabla \bullet \rho^o v^o v^o - \nabla \bullet [\omega P^n + (1-\omega) P^o] \qquad (3.11)$$

and

$$\frac{E^n - E^o}{\Delta t} = -\nabla \bullet (E^o + P^o)[\omega v^n + (1-\omega) v^o]. \qquad (3.12)$$

Since $\omega = 1$, the algorithm is completely implicit and reverts to the original equations as equations (3.7) and (3.8).

The change in pressure, δP, is defined as

$$\delta P \equiv \omega (P^n - P^o). \qquad (3.13)$$

Then the correction equation for momentum can be obtained in terms of δP by subtracting equation (3.10) from (3.11),

$$\frac{\rho^n v^n - \bar{\rho}\,\bar{v}}{\Delta t} = -\nabla\omega (P^n - P^o) = -\nabla \delta P. \qquad (3.14)$$

By rearranging equation (3.14) and letting $\rho^n = \bar{\rho}$, the new velocity is obtained as

$$v^n = -\frac{\Delta t}{\bar{\rho}} \nabla \delta P + \bar{v}, \qquad (3.15)$$

then a correction equation for energy using the equation of state is obtained as

$$\varepsilon^n = \frac{\delta P}{(\gamma - 1)\omega} + \varepsilon^o, \qquad (3.16)$$

where the ω factor appears from the definition of δP from equation (3.13). δP can be found by substituting equations (3.15), (3.16), and (3.5) into equation (3.12) to obtain

$$\frac{\bar{\rho}\,\bar{v}^2 - \rho^o v^{o2}}{2\Delta t} + \frac{\delta P}{(\gamma-1)\omega\,\Delta t} = \omega\,\Delta t\,\nabla\bullet\left(\frac{E^o + P^o}{\bar{\rho}}\right)\nabla\,\delta P$$

$$-\omega\,\nabla\bullet(E^o + P^o)\,\bar{v} - (1-\omega)\,\nabla\bullet(E^o + P^o)\,v^o. \qquad (3.17)$$

Note that the kinetic energy change is included explicitly. For convenience, we define the quantity $\bar{E}$,

$$\frac{\bar{E} - E^o}{\Delta t} \equiv - \nabla\bullet(E^o + P^o) [\omega \bar{v} + (1-\omega) v^o] \qquad (3.18)$$

so that equation (3.17) can be rewritten as

$$\frac{\delta P}{-(\gamma-1)\omega \Delta t} - \omega \Delta t \nabla\bullet \left(\frac{E^o + P^o}{\bar{\rho}} \right) \nabla \delta P = \frac{\bar{E} - E^o}{\Delta t} - \frac{\bar{\rho} \bar{v}^2 - \rho^o v^{o2}}{2\Delta t} . \qquad (3.19)$$

By dividing $- \omega \Delta t$ from both sides of Eq. (3.19), then

$$\frac{\delta P}{-(\gamma-1)\omega^2 \Delta t^2} + \nabla\bullet \left(\frac{E^o + P^o}{\bar{\rho}} \right) \nabla \delta P = \frac{\bar{E} - E^o}{-\omega \Delta t^2} + \frac{\bar{\rho} \bar{v}^2 - \rho^o v^{o2}}{2\omega \Delta t^2} \qquad (3.20)$$

The barely implicit correction (BIC) is carried out in three stages. In the first stage, equations (3.9), (3.10), and (3.18) are integrated with any one-step explicit method. In the second stage, the pressure correction equation (3.20) is solved by a numerical elliptic solver. In the last stage, equations (3.15) and (3.16) are used to obtain the values of momentum and energy at the new time-step. These values of the momentum and energy are then added to the contributions from the diffusive, reactive, and radiative parts of the equations (3.2) and (3.3) to obtain final values of the energy and momentum.

Our focus here is to obtain the solution of equation (3.20) for δP, which is of the form given by equation (2.1). The right hand side of equation (3.20) is evaluated explicitly. Therefore, the right hand side of equation (3.20) is known and the only unknown is δP. Clearly equation (3.20) is linear and can be solved by Bi-CGSTAB iterative method.

4. NUMERICAL FORMULATION

In order to solve δP numerically, equation (3.20) can be written in the form of equation (2.3) by recognizing

$$u = \delta P \qquad (4.1)$$

where

$$a_1 = \frac{-1}{(\gamma-1)\omega^2 \Delta t^2} \qquad (4.2)$$

$$a_2 = 1 , \qquad (4.3)$$

$$a_3 = \frac{E^o + P^o}{\rho}, \tag{4.4}$$

$$a_4 = 0, \tag{4.5}$$

and

$$a_5 = \frac{E - E^o}{-\omega \Delta t^2} + \frac{\bar{\rho}\,\bar{v}^2 - \rho^o v^{o2}}{2\omega \Delta t^2}. \tag{4.6}$$

Therefore, $a_1..a_5$ are known and δP is the unknown. In general, at a fixed time, the coefficients $a_1...a_5$ and variable δP vary with position (x and y). Next, a set of algebraic equations are derived from equations (4.1) to (4.6) following the procedure outlined by G. Patnaik [9] for premixed flames.

4.1 Algebraic Equations in Matrix Form

Equation (4.1) is discretized based on a two dimensional finite volume grid. A typical cell is shown in Figure 2. Here, the grid contains nx and ny cells in x and y directions, respectively. Also, i and j are cell numbers in x and y directions, respectively.

The descritized equation for a typical cell (i,j) is given by

$$\begin{aligned} c_0(i,j)u(i,j) + [c_1(i,j)u(i-1,j) - c_1(i+1,j)u(i,j)] + [c_1(i+1,j)u(i+1,j) - c_1(i,j)u(i,j)] \\ + [c_2(i,j)u(i,j-1) - c_2(i,j+1)u(i,j)] + [c_2(i,j+1)u(i,j+1) - c_2(i,j)u(i,j)] = f(i,j) \\ \text{for } i=1,nx \text{ and } j=1,ny \end{aligned} \tag{4.7}$$

The values of the coefficients and the variables at cell numbers 0, (nx+1), and (ny+1) correspond to the guard cells and are evaluated from the boundary conditions. Inside the domain they are given by the following equations. The first term (self-term) coefficients of equation (4.7) are given by

$$c_0(i,j) = \Delta x(i) \Delta y(j) a_1(i,j). \tag{4.8}$$

The forcing terms on the right hand side of equation (4.7) are given by

$$f(i,j) = \Delta x(i) \Delta y(j) a_5(i,j). \tag{4.9}$$

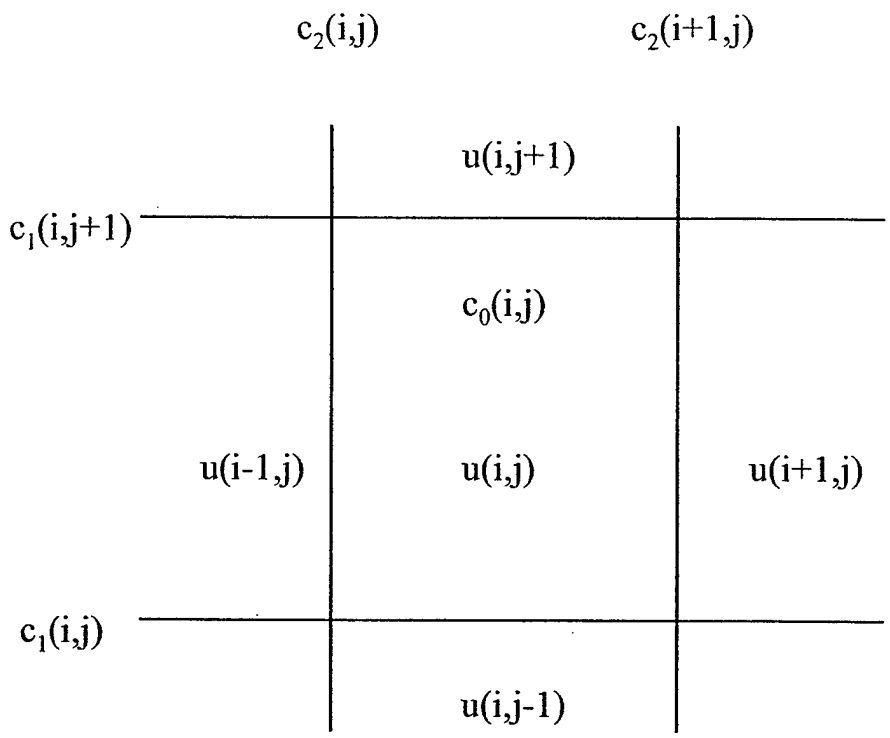

Figure 2 A Typical Cell for Bi-CGSTAB

The x coefficients in equation (4.7) are given by

$$c_1(i,j)= 0.5(a_3(i-1,j)+a_3(i,j))2\Delta y(j)/(\Delta x(i-1))+\Delta x(i)). \qquad (4.10)$$

The y coefficientsin equation (4.7) are given by

$$c_2(i,j)= 0.5(a_3(i,j-1)+a_3(i,j))2\Delta x(i)/(\Delta y(j-1))+\Delta y(j)). \qquad (4.11)$$

If the coefficient matrix a_2 is not an identity matrix, then $c_1(i,j)$ and $c_2(i,j)$ become

$$c_1(i,j) = a_2(i,j)c_1(i,j), \qquad (4.12)$$

and

$$c_2(i,j) = a_2(i,j)c_2(i,j). \qquad (4.13)$$

Therefore, equation (4.7) can be rewritten as

$$[K(i,j)]u(i,j) + c_1(i+1,j)u(i+1,j)+ c_1(i,j)u(i-1,j) +c_2(i,j)u(i,j-1) +c_2(i,j+1)u(i,j+1) = f(i,j)$$

for i=1,nx and j=1,ny,

(4.14)

where

$$K=[c_0(i,j)-c_1(i,j)-c_1(i+1,j)-c_2(i,j)-c_2(i,j+1)] . \quad (4.15)$$

Equations (4.14) and (4.15) represent a set of linear algebraic equations of the form

$$\mathbf{Au=f}, \quad (4.16)$$

which are solved by an iterative method as discussed in section 4.2.

4.2 Iterative Solution

The Bi-CGSTAB algorithms developed by Van Der Vorst [8] are shown in Table 1. Initially, an arbitrary vector $\underline{\mathbf{r}}$ is determined, say by $\underline{\mathbf{r}} = \mathbf{r}_0$, where $\mathbf{r_0}$ is an initial residual matrix. The values of $\mathbf{r_0}$ are calculated from $\mathbf{r=f-Au_0}$. Then the iteration loop process starts (I=1,2,3...). Within each loop, Bi-CGSTAB carries out two updates to the residual r of the current solution **u**. The two updated residuals are calculated from $\mathbf{s} = \mathbf{r}_{i-1} - \alpha \mathbf{v}_i$ and $\mathbf{r}_i = \mathbf{s} - \omega_i \mathbf{t}$. This explains Bi- as the name of the algorithm. If the residual **s** is small enough, the second updated residual will be skipped. The iteration process is continued until a pre-specified tolerance condition is satisfied.

Preconditioning or reformulation of the equation (4.14) is often useful in speeding up the convergence. Van Der Vorst suggested the following preconditioning of the equations

$$\underline{\mathbf{r}} = \{u(i,j) + K^{-1}[\, c1(i+1,j)u(i+1,j)+ c1(i,j)u(i-1,j) +c2(i,j)u(i,j-1) +c2(i,j+1)u(i,j+1)]\}-K^{-1}\, f(i,j), \quad (4.17)$$

where

$$K^{-1}=1/\,[c0(i,j)-c1(i,j)-c1(i+1,j)-c2(i,j)-c2(i,j+1)]. \quad (4.18)$$

If the magnitude of K is less than the magnitude of K^{-1}, then the magnitude of residual vector is reduced and the iteration process is faster, because the initial guess of u is closer to the solution. The iteration process with a preconditioned algorithm will not change the original system. However, it changes $\underline{\mathbf{r}}$, where $\underline{\mathbf{r}}$ is an arbitrary vector, e.g., $\mathbf{r} = r_0$, and $r_o = f - Ax_0$ for initial guess.

The convergence criterion in Bi-CGSTAB is determined by the square root of the residual, i.e., σ = square root of $(\underline{r}_0, r_{j-1})$, where $(\underline{r}_0, r_{j-1})$ is a dot product of two residual vectors from sequential steps in the calculation. The tolerance is selected based on this convergence criterion and is specified directly. In the computer code, a parameter named norm is introduced outside iteration loop, and norm = 1/square root of (f,f).

Bi-CGSTAB Algorithm	Bi-CGSTAB with Preconditioning
Initial guess $u_0 \Rightarrow r_0 = f - Au_0$	Initial guess $u_0 \Rightarrow r_0 = f - Au_0$
An arbitrary vector $\underline{r} = r_0$	An arbitrary vector $\underline{r} = r_0$
$\rho_0 = \alpha = \omega_0 = 1$	$\rho_0 = \alpha = \omega_0 = 1$
$v_0 = p_0 = 0$	$v_0 = p_0 = 0$
For i=1,2,3,..(start iteration loops)	For i=1,2,3,...
$\rho_i = (\underline{r}_0, r_{i-1})$	$\rho_i = (\underline{r}_0, r_{i-1})$
$\beta = (\rho_i / \rho_{i-1})(\alpha/\omega_{i-1})$	$\beta = (\rho_i / \rho_{i-1})(\alpha/\omega_{i-1})$
$p_i = r_{i-1} + \beta_i (p_{i-1} - \omega_{i-1} v_{i-1})$	$p_i = r_{i-1} + \beta_i (p_{i-1} - \omega_{i-1} v_{i-1})$
	Solve y from $Ky = p_i$
$v_i = Ap_i$	$v_i = Ay$ $(y = pK^{-1})$
$\alpha = \rho_i / (\underline{r}_0, v_i)$	$\alpha = \rho_i / (\underline{r}_0, v_i)$
$s = r_{i-1} - \alpha v_i$ (update residual)	$s = r_{i-1} - \alpha v_i$
	Solve y from $Kz = s$
$t = As$	$t = Az$ $(z = sK^{-1})$
$\omega_i = (t,s)/(t,t)$	$\omega_i = (K^{-1}t, K^{-1}s)/(K^{-1}t, K^{-1}t)$
$u_i = u_{i-1} + \alpha p_i + \omega_i s$	$u_i = u_{i-1} + \alpha p_i + \omega_i s$
If u_i is accurate enough then quit	If u_i is accurate enough then quit
$r_i = s - \omega_i t$ (update residual)	$r_i = s - \omega_i t$
end	end

Table 1 Bi-CGSTAB Algorithms with and without Pre-conditioning [8]

The convergence criterion is set as the product of σ and norm. This could help set a suitable criterion to avoid unrealistic tolerance, since usually σ is a very small number if the program converges. However, if the source term f is zero, norm cannot be used.

As discussed earlier, the values at the guard cells, which are on the boundaries of the domain, appear explicitly in the governing algebraic equations (4.14). In Bi-CGSTAB method, the boundary conditions are specified at each iteration step when the updated residual is calculated. For the solid combustion problem, the pressure gradient is set to zero at all boundaries except the outflow. At the downstream of the fuel plate, we set pressure to be atmospheric, i.e., $\delta P=0$. In BIC-FCT formulation, all of the boundary conditions are implemented at the interface of the guard cell and the first cell in the domain. Therefore, the Dirilecht boundary condition is implemented by setting the values of the guard cells as a reflection of the internal cell values, $\delta P(nx+1,j) = \delta P_G = -\delta P\ (nx,j)$.

5. NUMERICAL TESTS

The solid combustion problem involves both pressure gradient and pressure boundary conditions. For these boundary conditions, which may affect accuracy and convergence, the computer code is tested by comparing the numerical results against analytical solutions for a test problem.

Steady temperature of the electric heater (Arpact [14]) can be described by the following equations on a rectangular domain of size 2Lx2H as shown in Figures 3a and 3b.

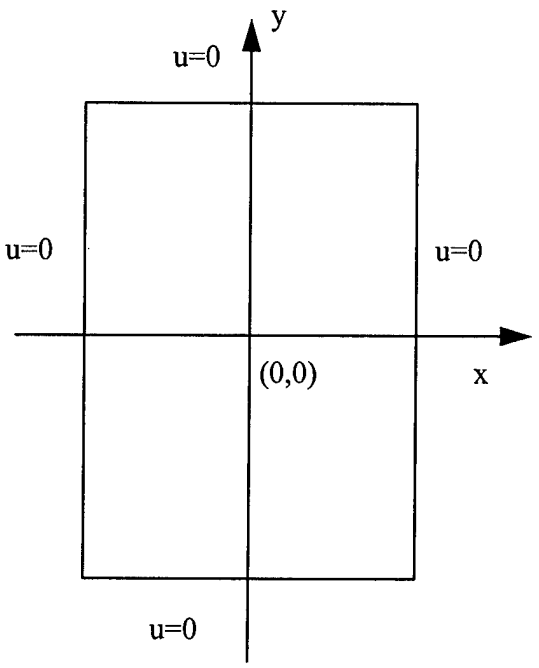

Figure 3a Heat Transfer from a Square Heater

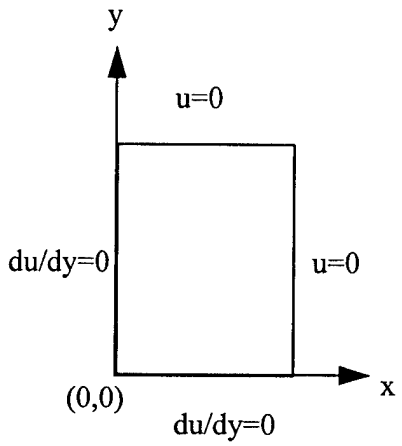

Figure 3b Computational Domain for the Test Problem

The temperature field is described by the following equation:

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + 1 = 0. \qquad (5.1)$$

The boundary conditions are

$$\frac{\partial u}{\partial x}(0, y) = 0, \quad \frac{\partial u}{\partial y}(x,0) = 0, \quad u(L,y) = 0, \quad u(x, H) = 0. \qquad (5.2a,b,c,d)$$

Equation (5.1) is discretized on a 100x100 cell uniform grid. The resulting equations (4.17) are solved by Bi-CGSTAB for the temperature field, u, and are shown in Table 2.

		Analytical	Numerical (100x100 cells), tolerance=1.E-3					
			Case 1			Case 2		
x	y	u(x,y)	u(x,y)	Error	%error	u(x,y)=0.	Error	%error
0.1	0.1	0.28972	0.28913	5.90E-04	0.2	0.29202	2.30E-03	0.79
0.3	0.1	0.26958	0.26852	1.06E-03	0.39	0.27143	1.85E-03	0.69
0.5	0.1	0.22754	0.22598	1.56E-03	0.69	0.22893	1.39E-03	0.61
0.7	0.1	0.16012	0.15796	2.16E-03	1.35	0.16099	8.70E-04	0.54
0.9	0.1	0.06227	0.05935	2.92E-03	4.69	0.06256	2.90E-04	0.47
0.1	0.3	0.26957	0.26852	1.05E-03	0.39	0.27143	1.86E-03	0.69
0.3	0.3	0.25118	0.24974	1.44E-03	0.57	0.25264	1.46E-03	0.58
0.5	0.3	0.21264	0.21080	1.84E-03	0.87	0.21370	1.06E-03	0.50
0.7	0.3	0.15035	0.14806	2.29E-03	1.52	0.15100	6.50E-04	0.43
0.9	0.3	0.05888	0.05601	2.87E-03	4.87	0.05907	1.90E-04	0.32
0.1	0.5	0.22753	0.22598	1.55E-03	0.68	0.22893	1.40E-03	0.62
0.3	0.5	0.21263	0.21080	1.83E-03	0.86	0.21370	1.07E-03	0.50
0.5	0.5	0.18116	0.17909	2.07E-03	1.14	0.18192	7.60E-04	0.42
0.7	0.5	0.12949	0.12719	2.30E-03	1.78	0.12993	4.40E-04	0.34
0.9	0.5	0.05149	0.04891	2.58E-03	5.01	0.05165	1.60E-04	0.31
0.1	0.7	0.16011	0.15796	2.15E-03	1.34	0.16099	8.80E-04	0.55
0.3	0.7	0.15034	0.14806	2.28E-03	1.52	0.15100	6.60E-04	0.44
0.5	0.7	0.12948	0.12719	2.29E-03	1.77	0.12993	4.50E-04	0.35
0.7	0.7	0.09444	0.09222	2.22E-03	2.35	0.09468	2.40E-04	0.25
0.9	0.7	0.03892	0.03677	2.15E-03	5.52	0.03897	5.00E-05	0.13
0.1	0.9	0.06226	0.05935	2.91E-03	4.67	0.06256	3.00E-04	0.48
0.3	0.9	0.05885	0.05601	2.84E-03	4.83	0.05907	2.20E-04	0.37
0.5	0.9	0.0515	0.04891	2.59E-03	5.03	0.05165	1.50E-04	0.29
0.7	0.9	0.0389	0.03677	2.13E-03	5.48	0.03897	7.00E-05	0.18
0.9	0.9	0.01754	0.01608	1.46E-03	8.32	0.01751	-3.00E-05	0.17

Numerical treatments for zero boundary conditions:

Case 1: $u(nx+1,j) = -u(nx.j)$ Case 2: $u(nx+1,j) = 0.$

$u(i,ny+1) = -u(i,ny)$ $u(i, ny+1) = 0.$

Table 2 Comparision of Bi-CGSTAB Computations with the Exact Analytical Solution for the Square Heater

Also shown in Table 2 is the analytical solution, which is represented as a series (n=6) given by

$$u(x,y)/L^2 = 0.5*[1-(x/L)^2]-2\Sigma_{n=0}^{\infty} \frac{(-1)^n(\cosh \lambda_n y)}{(\lambda_n L)^3 \cosh \lambda_n h} \cos(\lambda_n x) \qquad (5.3)$$

where $\lambda_n L=(2n+1)\pi/2$, n=1,2,..6, and n represents the number of terms in the series. The numerical solution was obtained for two different boundary conditions. Case 1 uses the reflective boundary conditions. In Case 2 the guard cell values are directly set to zero. Clearly, the numerical solution is generally within few percent of the analytical solution with the reflective boundary conditions. Solutions for case 2, however, have almost a factor of ten less error near the domain corners than the those for case 1. This suggests that the exact implementation of the boundary conditions may affect the accuracy of the solution near the corners of the domain.

6. RESULTS AND DISCUSSION

For PMMA combustion, the rectangular domain shown in Figure 1 is descritized by 192x144 finite volume cells. The cells are packed closely near the leading edge region of the PMMA plate and are stretched with distance from the leading edge. The smallest cells are 0.2 mm size. The gases are ignited for the first 2000 time steps (time step=1e-05 sec). At each time step, solution of equation (3.20) and the final solutions of equations (3.1) to (3.6) are obtained by time step splitting.

The boundary conditions at the PMMA surface describe the no-slip condition, the interfacial transport of mass, momentum and energy, and pyrolysis kinetics, all of which are given elsewhere [1-4]. The interfacial temperature, concentrations of the species, and fuel ejection rate due to pyrolysis are unknowns and are determined as a part of the solution. At each time step, a multi-variable fixed point iterative (MVFP) method is used to generate new values of the interfacial quantities. The new interfacial values are used as boundary conditions and equations (3.20), and (3.1) to (3.6) are re-solved. After convergence is obtained for the interfacial quantities, the computations are performed for the next time step. These iterations are important since they can significantly affect the local burning rates of PMMA. Computations were performed on SGI (Silicon Graphics) work station equipped with 300 MHZ, R10000, MIPS processor.

Figure 4 shows the solutions of equation (3.20) for pressure using Bi-CGSTAB. They correspond to steady state (12000 time steps of 5e-05 sec). They show a small raise in δP near the leading edge of the PMMA plate (x=1.7 cm). Combustion rates are highest near this region, where fresh air comes in contact with the fuel vapor generated at the PMMA surface. The combustion rates decrease sharply with the distance from the leading edge region and result in a sharp decrease in δP as exhibited by Figure 4. A much higher raise in δP was found during the ignition period and it falls steadily with time.

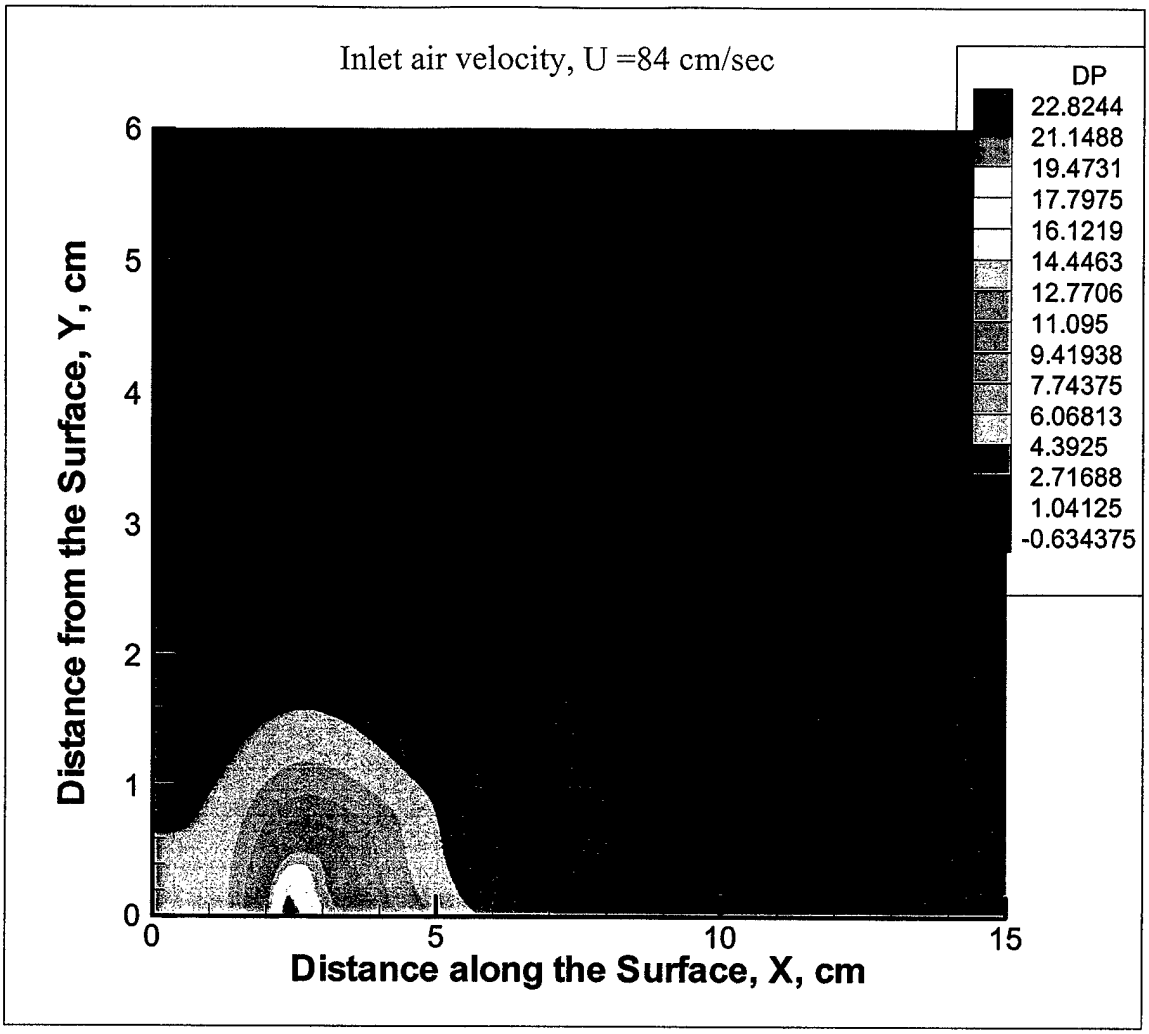

Figure 4 Solutions of the Pressure Equation using Bi-CGSTAB for PMMA Combustion

Figures 5 and 6 show the axial and vertical velocity components obtained from equation (3.16). Clearly, the axial velocity decreases as the gases approach the intense combustion region near the leading edge of the PMMA plate and accelerate downstream with distance x. The axial velocity is small near the surface due to no-slip, reaches peak value just above the flame, and approaches free stream value (84.0 cm/sec) far from the surface in y-direction. The vertical component of the velocity shows a large increase near the leading edge where δP is the largest. The vertical velocity of the fuel vapors due to PMMA pyrolysis is much smaller that the peak value shown in Figure 6. Together these results show that combustion near the leading edge region diverts the air flow upward away from the PMMA surface. This may have important implications on how a suppression agent such as water mist might distribute itself near this region of intense combustion.

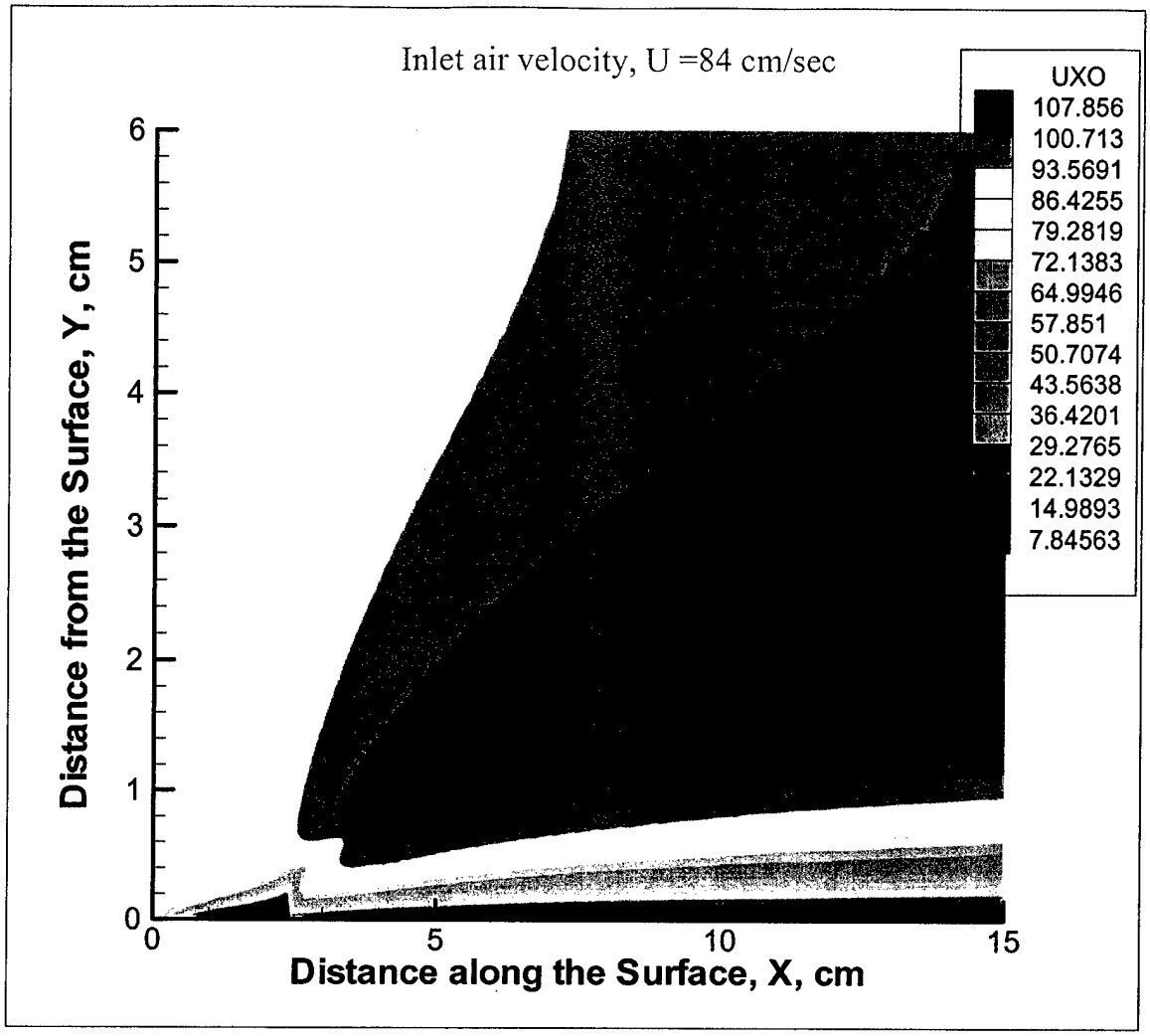

Figure 5 Axial Velocity Contours for PMMA Combustion Obtained by using Bi-CGSTAB

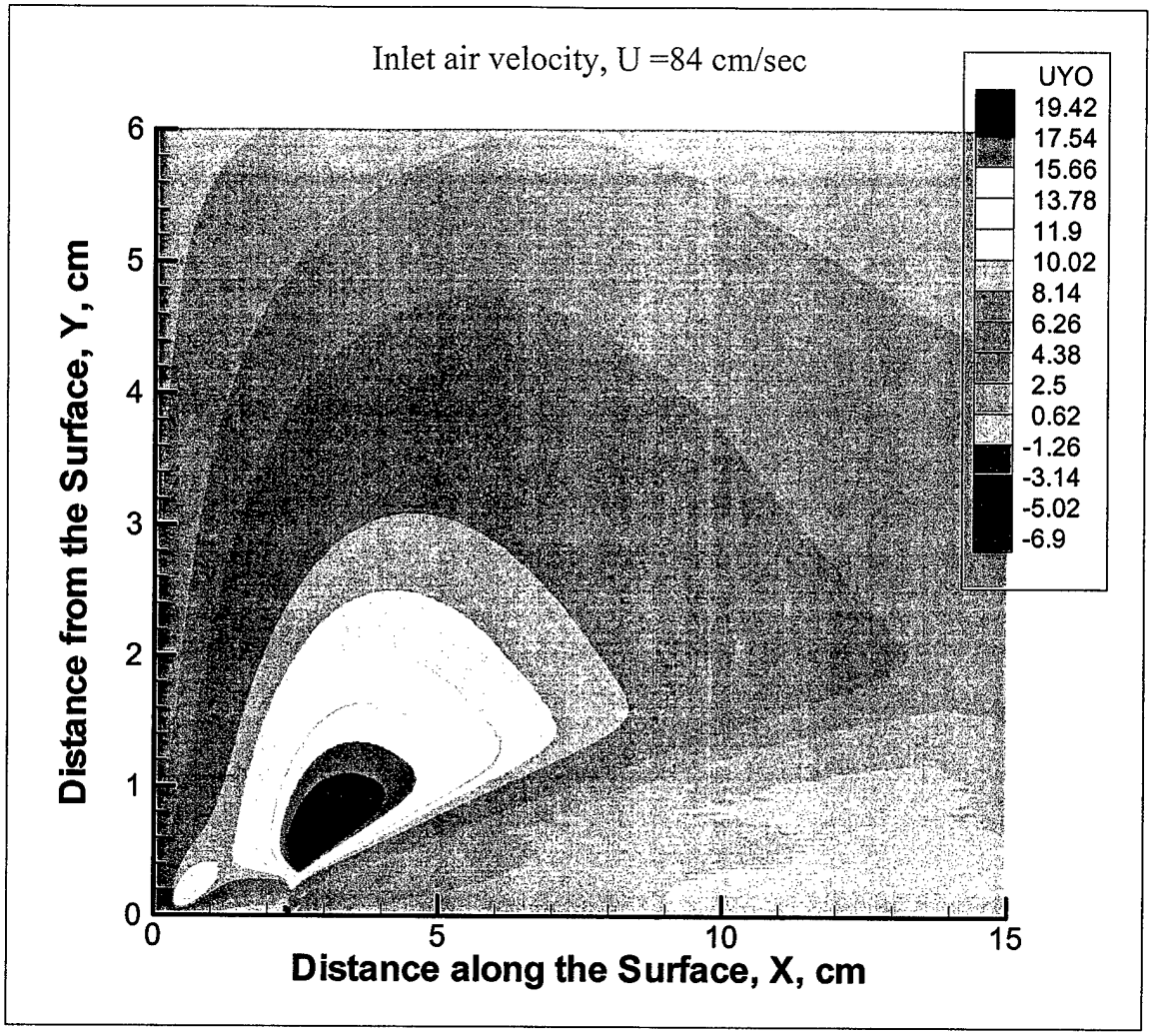

Figure 6 Vertical Velocity Contours Obtained by using Bi-CGSTAB

At the PMMA surface the local regression rate is directly proportional to the local heat feed back from the flame to the surface. The heat transport drives the PMMA pyrolysis that turns the solid into fuel vapor. Solutions of the full Navier-Stokes equations are used to evaluate the heat feed back. The local regression rate along the length of the PMMA plate is shown in Figure 7. The regression rate is highest near the leading edge and decreases sharply within a short distance from the leading edge. The sharp decrease in the regression rate is due to the boundary layer structure of the flame. The boundary layer thickness increases sharply with distance from the leading edge and results in a heat feed back profile that decreases with the distance.

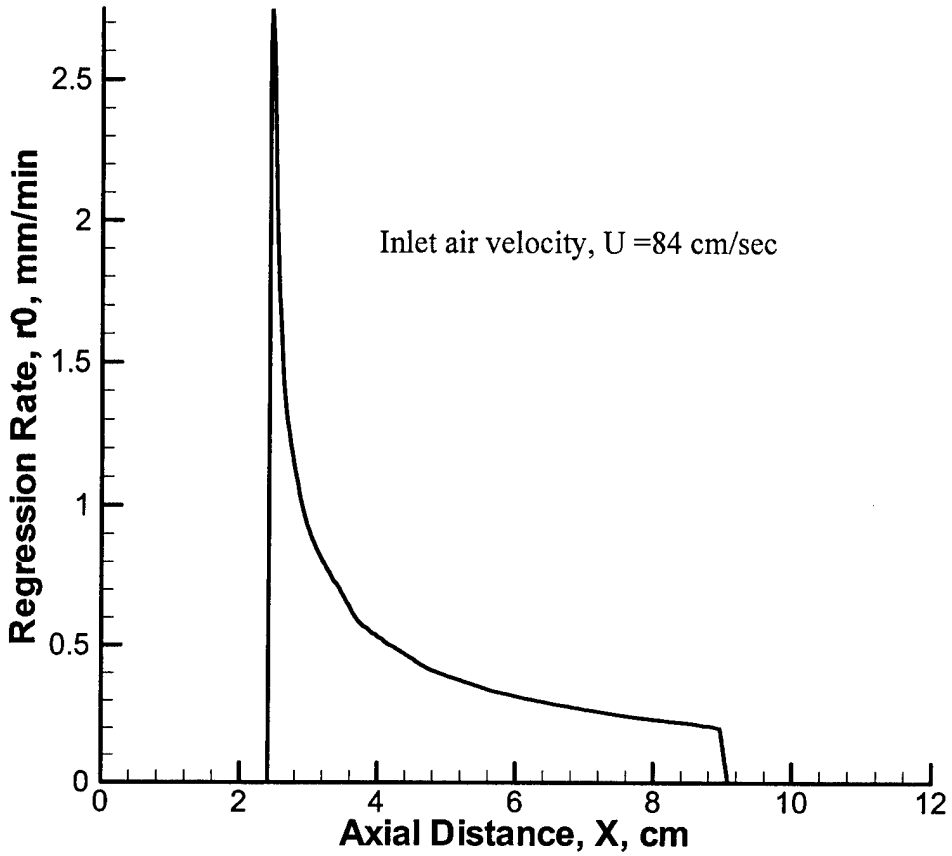

Figure 7 Regression Rate Profile for the Boundary Layer Combustion of PMMA

Instead of using Bi-CGSTAB, computations were also performed using the multi-grid method (MGRID, with level 4 and cycle 2) for identical inputs, which were discussed above. Therefore, all of the computations are identical except for the solution of the pressure equation (3.20). The results are summarized in the second and third columns of Table 3. The leading edge of the PMMA plate is located at I=41 cell in the computational domain. Table 3 shows that the temperature (K), pressure (dynes/cm^2), and local burning rates (gm/cm^{2}sec) are nearly the same for both methods near the leading edge (I=42, j=1) at steady state (12001 time steps). The maximum flame temperatures, which occur inside the domain (i=64,j=10), are also identical for both the methods. Table 3 also shows the values of pressure at the inlet (i=0), where the pressure gradient is set to zero ($d\delta P/dx=0$), and at the outlet (i=193), where the pressure is set to atmospheric ($\delta P=0$). The computations with Bi-CGSTAB appear to satisfy the exit boundary condition better than those with MGRID especially during the transient (6001 time step).

CPU time (sec.)	Mgrid	Bi-CGSTAB	Bi-CGSTAB
		tolerance=1.e-3	tolerance=1.e-5
Implicit	**9.5233e-5**	**3.4231e-5**	**7.6190e-5**
Results			
At 12001th step			
T(42,1)	1010	1010	1010
T(64,10)	1910	1910	1910
$\delta p(42,1)$	22.1	24.5	25.1
$\delta p(64,10)$	12.3	13.2	13.6
$r_0(42,1)$	53.8e-4	54.5e-4	54.5e-4
Boundary cells			
$\delta p(1,1)$	dynes/cm^2		
At first step	3.29	3.29	3.29
2001th step	4260	3120	4700
4001th step	7.83	5.14	5.40
6001th step	7.27	4.95	5.39
8001th step	5.25	5.34	5.38
10001th step	5.31	5.63	5.43
12001th step	5.17	5.23	5.41
$\delta p(192,1)$	dynes/cm^2		
At first step	3.29	3.29	3.29
2001th step	66.1	57.0	82.1
4001th step	-116	1.25	-11.4
6001th step	-146	-0.021	0.704
8001th step	-0.213	0.189	0.846
10001th step	0.780	0.091	0.281
12001th step	2.03	1.02	0.588

Table 3 Comparision of Bi-CGSTAB with MGRID Results for PMMA Combustion

Computer time (clock time) profiling was also performed for both the methods to compare their efficiency. These are shown as seconds per time step for each stage of the algorithm. The computation time for the solution of the pressure equation (3.20) is shown as "Implicit" in Table 3. Clearly, solving the pressure equation is significantly faster (factor of 2.5)

with MGRID. The number of MVFP iterations was found to be roughly the same for both the methods.

Table 3 also shows the computations with two different tolerance limits for the Bi-CGSTAB iterations. Increased tolerance affected the pressure slightly but did not significantly improve the temperature and burning rates. The computation time, however, increased significantly with the increased tolerance as one would expect.

7. CONCLUSIONS

Simulation of solid combustion problems involve an elliptic equation for pressure, which must be solved multiple times at each time step in order to obtain local burning rate profiles accurately. A robust conjugate gradient method (Bi-CGSTAB) has been implemented for the solution of PMMA combustion. The conjugate gradient method is shown to provide superior performance in computation time over a multi-grid method (MGRID) without any loss in the accuracy for the temperature, burning rate, and pressure with tolerance limit set at 1e-03.

8. AKNOWLEDGMENTS

The authors sincerely thank Dr. Gopal Patnaik of the Laboratory for Computational Physics and Fluid Dynamics of Materials Science and Component Technology Directorate, Naval Research Laboratory for providing us the conjugate gradient alogorithms developed for his premixed flames and for his technical advice. Thanks are also due to Dr. John Hoover for his assistance for computer hardware. This work is funded by the Office of Naval Research through Naval Research Laboratory, and Code 334, under the Damage Control Task of FY01 Surface Ship Hull, Mechanical, and Electrical Technology Program, PE602121N. One of the authors, Ms. Y-J. Wang, was sponsered through ASEE summer faculty research program at the Naval Research Laboratory.

9. REFERENCES

[1] Ananth, R., Tatem. P.A., and Ndubizu, C.C., "A Numerical Model for the Development of a Boundary Layer Diffusion Flame Over a Porous Flat Plate", NRL Memorandum Report NRL/MR/6183-01-8547, Naval Research Laboratory, Washington, DC, 2001

[2] Ananth, R., Ndubizu, C.C., Tatem, P.A., Patnaik, G. and Kailasanath, K., "Boundary-Layer Diffusion Flame Over a Porous Plate", Second Joint Meeting of the US Sections of the Combustion Institute, Oakland, CA, March 26-28, (2001)

[3] Ananth, R., Ndubizu, C.C., Tatem, P.A., Patnaik, G. and Kailasanath, K., "Full Navier-Stokes Modeling of a Boundary-Layer Diffusion Flame Over a Porous Flat Plate", Sixth International Microgravity Combustion Workshop, Cleveland, OH, May 22-24, (2001)

[4] Ananth, R., Ndubizu, C.C., Tatem, P.A., Patnaik, G. and Kailasanath, K., "Suppression Dynamics of a Boundary-Layer Diffusion Flame", Halon Options Technical Working Conference, Albuquerque, NM, April 24-26, (2001)

[5] Patnaik, G., R. H. Guirguis, J.P. Boris, and E.S. Oran, "A Barely Implicit Correction for Flux-Corrected Transport", NRL Memorandum Report 5855, Naval Research Laboratory, Washington, DC, (1986)

[6] Oran, E. S. and J. P. Boris, "Numerical Simulation of Reactive Flow", Elsevier Publishing Co., New York, (1987)

[7] Sonneveld, P., "CGS: A Fast Lanczos-Type Solver for Non-Symmetric Linear Systems", SIAM J. Sci. Statist. Comput., **10**, 36 (1989)

[8] Van Der Vorst, H.A., "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 (1992)

[9] Patnaik, G., Naval Research Laboratory, Washington, Dc, Private Communication

[10] DeVore, C.R., "Vectorization and Implementation of an Efficient Multi-grid Algorithm for the Solution of Elliptic Partial Differential Equaitons", NRL Memorandum Report 5504, Naval Research Laboratory, Washington, DC (1984)

[11] Douglas, C.C., "Multi-Grid Algorithms with Application to Elliptic Boundary- Value Problems", SIAM J. Numer. Anal., **21**, 236 (1984)

[12] Patnaik, G., Laskey, K.J., Kailasanath, K., Oran, E.S., and Brun, T.A., "FLIC-A Detailed , Two-Dimensional Flame Model", NRL Memorandum Report 6555, Naval Research Laboratory, Washington, DC (1989)

[13] Casulli, V. and Greenspan, D., "Pressure Method for the Numerical Solution of Transient Compressible Fluid Flows", Int. J. for Numerical Methods in Fluids, **4**, 1001 (1984)

[14] Arpaci, V.S., "Conduction Heat Transfer", Addison-Wesley Publishing Company, Reading, MA, (1966)