Seventh International Conference on Computational Fluid Dynamics (ICCFD7), Big Island, Hawaii, July 9-13, 2012

ICCFD7-1904

Fast Iterative Methods for Navier-Stokes Equations with SST Turbulence Model and Chemistry

O. Peles*, E. Turkel** and S. Yaniv*
Corresponding author: eliturkel@gmail.com

* Rocket Systems Division/ IMI, Israel
** School of Mathematics/Tel-Aviv University.

Abstract: The steady state compressible Navier-Stokes equations are solved with an explicit Runge-Kutta scheme accelerated by multigrid and an implicit preconditioner. Extensions are made to include a k-ω/SST turbulence model and chemical reactions. The implicit smoother enables the use of high CFL numbers yielding fast convergence. We present applications to turbulent solutions for flows about wings and plumes and for flows with several chemical reactions.
Keywords: Navier-Stokes, convergence acceleration, SST turbulence, chemistry.

1 Introduction

The compressible steady state Navier-Stokes system of equations is a nonlinear mixed hyperbolic-elliptic-parabolic system. Classical methods for solving these equations use second order accurate methods in space. The standard technique is to march the time dependent equations in (pseudo) time until a steady state is achieved. The time marching is done by either an explicit method (Lax-Wendroff, Runge-Kutta etc.), ADI or else by a full Newton method coupled with a Krylov technique for the resultant large sparse linear equations. This is frequently supplemented with a multigrid (MG) acceleration. In spite of all these techniques, the convergence to the steady state requires many iterations frequently ranging from several hundred MG cycles to thousands of explicit time steps. This slow convergence becomes even worse for high Reynold's number flows when a highly stretched mesh is used in the boundary layer requiring small time steps. Local time stepping and implicit methods alleviate but do not eliminate this slowdown. Rossow and later Swanson and Turkel [1,2,3] introduced an implicit preconditioner which allows for rapid convergence to a steady state, with CFL=1000, that is typically about 5 times faster, in CPU time, than previous optimal solvers. Frequently, convergence to engineering accuracy is obtained in tens of MG cycles and convergence to machine accuracy in about 100 MG cycles using a 3 stage explicit Runge-Kutta scheme. The original results were obtained for the steady state compressible Navier-Stokes equations coupled with an algebraic (Baldwin-Lomax) turbulence mode. In this paper we extend these previous results in several directions.

We consider extensions from an algebraic turbulence model to a multi-equation turbulence model. As a typical example we have chosen the k-ω SST equations. In order to advance the turbulence equations with a large CFL we introduce an implicit preconditioner similar to that used for the fluid equations. This preconditoner again allows much larger time steps than usually used for these equations allowing a rapid convergence to a steady state. In addition, the preconditoner increases the robustness of the convergence.

In another direction of extensions we consider the addition of fluids with reactants. Thus, in addition to the fluid equations there are many chemical reactions. These reactions add a stiff source term that severely reduces the allowable time step for the fluid. Typically, industrial calculations use CFL=.05 for these calculations. The introduction of the implicit smoother allows a much larger time step and again a faster and more robust convergence to a steady state.

2 The k-ω/SST Equations

In previous work with the acceleration technique only algebraic turbulence modes (Baldwin-Lomax) were considered. Thus, the implicit preconditioner was applied only to the fluid equations. We now consider the extension of this technique to multi-equation turbulence models (see [4]). For the turbulence model equations, frequently, the ADI method is used to advance the equations in pseudo-time to a steady state at each flow time step. This solution algorithm decouples the fluid and turbulence equations. The fluid equations are solved for a given turbulent coefficient – μ_t and the turbulence equations are solved separately to update μ_t. However, ADI while formally unconditionally stable, in practice, allows one to increase the CFL only by a factor of about 5-10 over the explicit time restriction. This implies that the time step used for the turbulence model is much smaller than that used for the fluid equations, which can be 1000 times larger than the explicit CFL because of the implicit preconditioner.

Here, we introduce the RK implicit smoother into the k-ω/SST model which enables us to efficiently solve the weakly coupled system of equations of the Navier-Stokes equations with the two equation turbulence model. We first describe some details of the turbulence model and its numerical approximation. The turbulent viscosity is defined as

$$\nu_t = \frac{\mu_t}{\rho} = \frac{a_1 k}{\max\left(a_1\omega, SF_2\right)}$$

where S is the vorticity and y- is the minimum distance from the wall. F1 and F2 are defined as:

$$F_1 = \tanh\left\{\min\left[\max\left(\frac{\sqrt{k}}{0.09\omega y}, \frac{500\nu}{\omega y^2}\right), \frac{4\rho\sigma_{\omega 2} k}{CD_{k\omega} y^2}\right]^4\right\}$$

$$F_2 = \tanh\left[\left[\max\left(\frac{2\sqrt{k}}{\beta^* \omega y}, \frac{500\nu}{y^2 \omega}\right)\right]^2\right]$$

Then the SST turbulence equations are given by:

$$\frac{\partial k}{\partial t} + \vec{u}\cdot\nabla k = P_k - \beta^* k\omega + \frac{\partial}{\partial x_j}\left[\left(\nu + \sigma_k \nu_T\right)\frac{\partial k}{\partial x_j}\right]$$

$$\frac{\partial \omega}{\partial t} + \vec{u}\cdot\nabla \omega = \alpha S^2 - \beta\omega^2 + \frac{\partial}{\partial x_j}\left[\left(\nu + \sigma_\omega \nu_T\right)\frac{\partial \omega}{\partial x_j}\right] + 2\left(1 - F_1\right)\sigma_{\omega 2}\frac{1}{\omega}\frac{\partial k}{\partial x_j}\frac{\partial \omega}{\partial x_j}$$

$$\text{with} \quad P_k = \min\left(\tau_{ij}\frac{\partial u_i}{\partial x_j}, 10\beta^* k\omega\right)$$

where ω is the stress tensor. In many applications $\tau_{ij}\frac{\partial u_i}{\partial x_j}$ is replaced by μS^2 .

$$CD_{kw} = \max\left(2\rho\sigma_{\omega 2}\frac{1}{\omega}\frac{\partial k}{\partial x_i}\frac{\partial \omega}{\partial x_i}, 10^{-10}\right)$$

$$\phi = \phi_1 F_1 + \phi_2\left(1 - F_1\right)$$

$$\alpha_1 = \frac{5}{9}, \quad \alpha_2 = 0.44$$

$$\beta_1 = \frac{3}{40}, \quad \beta_2 = 0.0828, \quad \beta^* = \frac{9}{100}$$

$$\sigma_{k1} = 0.85, \quad \sigma_{k2} = 1$$

$$\sigma_{\omega 1} = 0.5, \quad \sigma_{\omega 2} = 0.856$$

3 Numerical Scheme for SST

The turbulence model equations are solved using a pseudo time integration with a low storage Runge-Kutta scheme: For each R-K stage we have:

$$k^{(q)} = k^{(0)} + \alpha_q \overline{\Delta k}^{(q-1)}$$

$$\omega^{(q)} = \omega^{(0)} + \alpha_q \overline{\Delta \omega}^{(q-1)}$$

$q = 1..$number of runge kutta stages, $\overline{\Delta k}$ and $\overline{\Delta \omega}$ are smoothed values, where:

$$\Delta k^{(q-1)} = -\frac{\Delta t}{Vol}\sum_{all\ faces}\left(F_c + F_v\right)^{(q-1)} S + \Delta t R_k{}^{(q-1)}$$

and similarly for ω. Using the terminology of the RK/implicit smoother, the equations are:

$$\frac{\partial}{\partial t}\begin{pmatrix} k \\ \omega \end{pmatrix} + \sum_{all\ faces}\begin{pmatrix} F_{c,k} \\ F_{c,\omega} \end{pmatrix} S + \sum_{all\ faces}\begin{pmatrix} F_{v,k} \\ F_{v,\omega} \end{pmatrix} S = \begin{pmatrix} R_k \\ R_\omega \end{pmatrix}$$

The numerical scheme for the smoother is:

$$\left[I + \varepsilon\Delta t\left(\frac{1}{Vol}\sum_{all\ faces}\left(A^+ + A_v\right)^{(q-1)} S - R^{(q-1)}(k,\omega)\right)\right]\left(\frac{\overline{\Delta k}}{\Delta\omega}\right)^p{}_{local}$$

$$= \begin{pmatrix} \Delta k \\ \Delta\omega \end{pmatrix}^q - \varepsilon\Delta t\left(\frac{1}{Vol}\sum_{all\ faces}\left(A^- - A_v\right)^{(q-1)} S\right)\left(\frac{\overline{\Delta k}}{\Delta\omega}\right)^{p-1}{}_{NB}$$

where p=1,...,m, m is the number of smoothing steps, $\left(\frac{\overline{\Delta k}}{\Delta\omega}\right)^q = \left(\frac{\overline{\Delta k}}{\Delta\omega}\right)^m$ and ε is an implicit parameter. In addition, $A^+ = u^+ I$, $A^- = u^- I$, $\boldsymbol{A_v}$ is the viscous flux Jacobian:

$$A_v = \frac{S}{Vol}\begin{pmatrix} \nu+\sigma_k\nu_T & 0 \\ 0 & \nu+\sigma_\omega\nu_T \end{pmatrix} \qquad (A_v)_{NB} = -A_v$$

R is the Jacobian of the source term. Only the decaying terms are used in the source flux Jacobian:

$$R(k,\omega) = \begin{pmatrix} \frac{\partial R_k}{\partial k} & \frac{\partial R_k}{\partial \omega} \\ \frac{\partial R_\omega}{\partial k} & \frac{\partial R_\omega}{\partial \omega} \end{pmatrix} = -\begin{pmatrix} \beta^*\omega & 0 \\ \beta^* k & 2\beta\omega \end{pmatrix}.$$

This system is solved with a Gauss-Seidel method using a small number of iterations m.

4 Results for SST Turbulence Model

We consider the combined fluid plus turbulence set of equations. The two sets are weakly coupled, and the implicit smoother is applied, separately, to both the fluid and turbulence equations. At present, multigrid is used only for the fluid equations. We describe the SST- turbulent transonic calculations for the flowfield around an RAE2822 wing and the turbulent flowfield of a rocket motor.

4.1 Turbulent transonic flowfield around RAE2822 wing

The free stream conditions are: M = 0.73, α = 2.79°. Figure 1 shows the convergence history (density-red, k-green) of the calculation of the flow around a RAE2822 wing. The calculation uses sequencing of two coarse grids and three levels of multigrid on the finest grid, in the fluid only. A second order upwind scheme with a Sweby limiter is used. The fluid CFL is 100,000 and turbulent CFL is 20,000; machine accuracy is obtained. Figure 1 shows the convergence and figure 2 shows the comparison between FLDYNS (an in-house code) solution and the experiment.

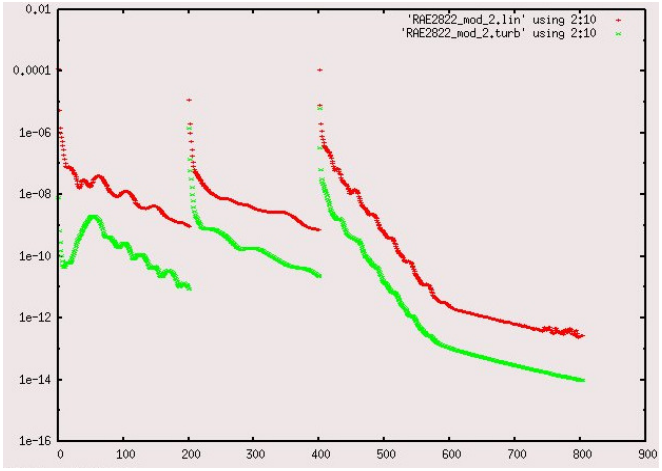

Figure 1: Density (red) and turbulent kinetic energy (green) convergence history

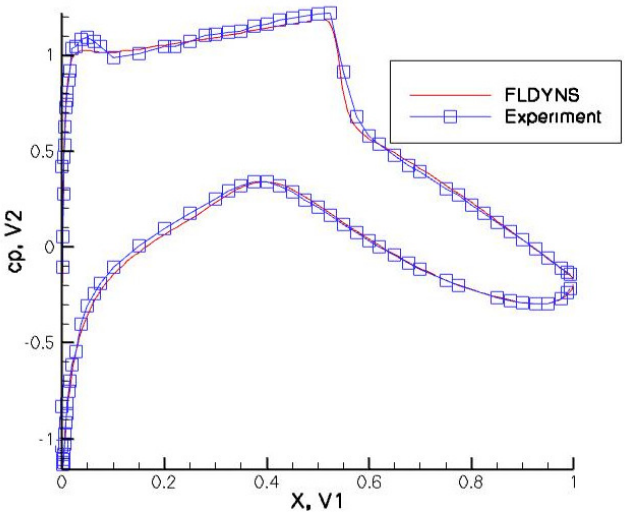

Figure 2: Comparison between FLDYNS solution and experiment

5 Chemistry

Turbulent reactive flow is important for various engineering problems. For example, designing of combustion chambers and modeling of rocket motor and rocket motor plumes. In rocket motor plumes, for example, turbulent flow is essentially different from laminar flow and real plumes can't be modeled without considering turbulence. The chemical reactions also add specific phenomena as after-burning when the combustion product meets the atmospheric oxygen starts to burn again. Reliable modeling of reactive flow in plumes is very important, for example, to evaluate the IR signature of the plume [7]. We again extend the idea of the RK/implicit smoother to reactive flow.

The challenge of numerical simulation of reactive flow is dealing with the stiffness of system, both in the turbulence model (K-ω SST in this work) and especially the stiffness from the chemical reactions. This stiffness is the result of a very wide range of the evolution time scales of the reactants.

5.1 Governing equations

The chemical reaction source terms of the inhomogeneous Navier-Stokes equations are: $(\dot{\rho}_1, \dot{\rho}_2, \dot{\rho}_3, ..., 0, 0)$. The gas mixture diffusion terms are added to the species mass conservation equations. The conservative variables are:

$$Q = (\rho_i, \rho\vec{u}, E) \quad \vec{u} = (u_x, u_y, u_z).$$

Let $u^2 = u_x^2 + u_y^2 + u_z^2$

then

$$E = \rho\left(e(T) + \frac{1}{2}u^2\right) = \sum_{i=1}^{N} \rho_i e_i(T) + \frac{1}{2}\rho u^2 = \sum_{i=1}^{N} \omega_i w_i e_i(T) + \frac{1}{2}\rho u^2$$

The mean internal energy in mass units is $e(T) = \sum_{i=1}^{N} y_i e_i(T)$

5.2.1 Source terms

We use the primitive variables $q=(\omega_i,u,T)$. We also define another set of primitive variables $U=(\rho_i,u,P)$. The equation of state is $P=\frac{\rho RT}{W}=RT\sum_{i=1}^{n}\omega_i$ with the mean molecular weight $W^{-1}=\sum_{i=1}^{n}\frac{\rho_i}{\rho}\frac{1}{w_i}$. The Jacobian matrix from primitive to conservative variables is

$$dQ=\begin{pmatrix} w_1 & 0 & 0 & 0 & 0 \\ 0 & w_2 & 0 & 0 & 0 \\ 0 & 0 & w_3 & 0 & 0 \\ uw_1 & uw_2 & uw_3 & \rho & 0 \\ w_1\left(e_1(T)+\frac{1}{2}u^2\right) & w_2\left(e_2(T)+\frac{1}{2}u^2\right) & w_3\left(e_3(T)+\frac{1}{2}u^2\right) & \rho u & \rho c_v \end{pmatrix} dq = Jdq$$

The chemical source term is $(\dot{\omega}_1,\dot{\omega}_2,\dot{\omega}_3,0,\dot{T})$. Each reaction of Nr reactions in the chemical model is described by the reaction equation:

$$\sum \upsilon'_{ik}\,\chi_k <=> \sum \upsilon''_{ik}\,\chi_k$$

where υ' and υ'' are the forward and backward stochiometric coefficients respectively.

$$\dot{\omega}_k=\sum_{i=1}^{N_r}\upsilon_{ik}q_i \qquad \upsilon_{ik}\equiv\upsilon''_{ik}-\upsilon'_{ik}$$

The progress variable q is given by

$$q_i=K_{f,i}(T)\prod_k\omega_k^{\upsilon'_{ik}}-K_{r,i}(T)\prod_k\omega_k^{\upsilon''_{ik}}$$

$$k_{f,i}=AT_i^{\beta_i}e^{-E_i/RT} \qquad k_{r,i}=k_{f,i}/k_{c,i}$$

A_i, β_i and E_i are the Arrhenius constants: A_i is the rate constant, β_i is the temperature exponent and E_i is the activation energy,

$$K_p=\exp\left(\frac{\Delta S}{R}-\frac{\Delta H}{RT}\right) \text{ and } k_{r,i}=K_p\left(\frac{P_{atm}}{RT}\right)^{\sum_{k\in K_i}\upsilon_{ik}}$$

$$\Delta S=\sum_{k\in K_i}\upsilon_{ik}S_k \qquad \Delta H=\sum_{k\in K_i}\upsilon_{ik}H_k \qquad \dot{T}=-\frac{1}{\rho c_v}\sum_k e_k(T)\,w_k\dot{\omega}_k$$

So, in conservative variables the source terms are:

$$\frac{d}{dt}\begin{pmatrix}\rho_1\\ \rho_2\\ \rho_3\\ \rho u\\ E\end{pmatrix}=\begin{pmatrix} w_1 & 0 & 0 & 0 & 0\\ 0 & w_2 & 0 & 0 & 0\\ 0 & 0 & w_3 & 0 & 0\\ uw_1 & uw_2 & uw_3 & \rho & 0\\ w_1\left(e_1(T)+\frac{1}{2}u^2\right) & w_2\left(e_2(T)+\frac{1}{2}u^2\right) & w_3\left(e_3(T)+\frac{1}{2}u^2\right) & \rho u & \rho c_v\end{pmatrix}\begin{pmatrix}\dot{\omega}_1\\ \dot{\omega}_2\\ \dot{\omega}_3\\ 0\\ \dot{T}\end{pmatrix}=\begin{pmatrix}w_1\dot{\omega}_1\\ w_2\dot{\omega}_2\\ w_3\dot{\omega}_3\\ 0\\ 0\end{pmatrix}$$

5.2.2 Diffusion terms

An additional diffusion term $\nabla\left(\frac{\mu}{Sc}\nabla\frac{\rho_i}{\rho}\right)$ appears in the RHS of the mass conservation equations which is the result of the gas mixture. On the RHS side of the energy conservation equation appears $\nabla\left(\sum_i\frac{\mu}{Sc}\nabla\frac{\rho_i}{\rho}\right)$ where μ is the viscosity and Sc is the Schmidt number for turbulent flow. $\frac{\mu}{Sc}$ is replaced by $\frac{\mu_l}{Sc_l}+\frac{\mu_t}{Sc_t}$ where the subscripts l and t refer to the laminar and turbulent viscosities and Schmidt numbers respectively.

5.3 RK/Implicit Smoother for Real gas

For each point, we solve the system

$$\left(I+\frac{\delta t}{Vol}\sum_{all\ faces}P_n^{+}ds\right)\delta\tilde{Q}_{i,j}^{n+1}=\delta Q_{i,j}^{n}-\frac{\delta t}{Vol}\sum_{all\ faces}P_n^{-}\delta\tilde{Q}_{NB}^{n+1}ds$$

where δQ are the residuals before smoothing, $\delta\tilde{Q}$ are the residuals in the n-th step of the smoothing, and the subscript NB denotes neighbor cells. This equation is implicit and we solve it iteratively using the Gauss-Seidel method. $P_n^{\pm}=\frac{1}{2}\left(P_n\pm|P_n|\right)$ where $P_n\equiv\frac{\partial Q}{\partial W}A_c\frac{\partial W}{\partial Q}$ and $|P_n|\equiv T\Lambda T^{-1}$ and T is the matrix of eigenvectors, Λ is a diagonal matrix containing the eigenvalues. $\frac{\partial U}{\partial Q}$ is the Jacobian matrix from the primitives δU to the conservatives δW. $\frac{\partial Q}{\partial U}$ is the Jacobian of the inverse transformation. If $A_p=\frac{\partial\vec{F}}{\partial U}$ and $A_c=\frac{\partial\vec{F}}{\partial Q}=\frac{\partial\vec{F}}{\partial U}\frac{\partial U}{\partial Q}=A_p\frac{\partial U}{\partial Q}$, $P_n\equiv\frac{\partial U}{\partial Q}A_p$,

$$\frac{\partial U}{\partial Q}=\begin{pmatrix}1/\rho & 0 & 0 & 0 & -u/\rho\\ 0 & 1/\rho & 0 & 0 & -v/\rho\\ 0 & 0 & 1/\rho & 0 & -w/\rho\\ u(1-\gamma) & v(1-\gamma) & w(1-\gamma) & (\gamma-1) & \frac{U^2}{2}(\gamma-1)\\ 0 & 0 & 0 & 0 & 1\end{pmatrix}$$

$$A_p = \begin{pmatrix} \rho u_k + n_x \rho u & n_y \rho u & n_z \rho u & n_x & u u_k \\ n_x \rho v & \rho u_k + n_y \rho v & n_z \rho v & n_y & v u_k \\ n_x \rho w & n_y \rho w & \rho u_k + n_z \rho w & n_z & w u_k \\ n_x \rho \left(\frac{U^2}{2} + \frac{p}{\rho} + e \right) + \rho u u_k & \rho n_y \left(\frac{U^2}{2} + \frac{p}{\rho} + e \right) + \rho v u_k & n_z \rho \left(\frac{U^2}{2} + \frac{p}{\rho} + e \right) + \rho w u_k & \frac{u_k \gamma}{(\gamma - 1)} & u_k \left(\frac{U^2}{2} - \Pi_{i=1..N} \right) u_k \\ n_x \rho y_i & n_y \rho y_i & n_z \rho y_i & 0 & u_k I_{NxN} \end{pmatrix}$$

$$U^2 = |\vec{u}|^2$$

$$\Pi_i = \frac{RT}{(\gamma - 1)\rho w_i} - e_i \left(= 0 \quad for \quad real \quad gas \right)$$

$$P = \begin{pmatrix} u_k & 0 & 0 & n_x / \rho & 0 \\ 0 & u_k & 0 & n_y / \rho & 0 \\ 0 & 0 & u_k & n_z / \rho & 0 \\ n_x p \gamma & n_y p \gamma & n_z p \gamma & u_k & 0 \\ n_x \rho y_i & n_y \rho y_i & n_z \rho y_i & 0 & u_k I_{NxN} \end{pmatrix}$$

$$|P| = \begin{pmatrix} |u_k| + (1 - |m_0|) c n_x n_x & (1 - |m_0|) c n_x n_y & (1 - |m_0|) c n_x n_z & m_0 n_x / \rho & 0 \\ (1 - |m_0|) c n_x n_y & |u_k| + (1 - |m_0|) c n_y n_y & (1 - |m_0|) c n_y n_z & m_0 n_y / \rho & 0 \\ (1 - |m_0|) c n_x n_z & (1 - |m_0|) c n_z n_y & |u_k| + (1 - |m_0|) c n_z n_z & m_0 n_z / \rho & 0 \\ m_0 n_x p \gamma & m_0 n_y p \gamma & m_0 n_z p \gamma & |u_k| + (1 - |m_0|) c & 0 \\ m_0 n_x \rho y_i & m_0 n_y \rho y_i & m_0 n_z \rho y_i & (1 - |m_0|) y_i / c & |u_k| I_{NxN} \end{pmatrix}$$

$$m_0 = \min \left(|m_k|, 1 \right) \mathrm{sgn} \left(m_k \right) \quad m_k = u_k / c \quad u_k = \vec{u} \cdot \vec{n}$$

5.4 RK/Implicit Smoother for reactive flow

For reactive flow, we need to add the source term Jacobian to the RHS of the smoother

$$\left[I + \varepsilon \Delta t \left(\frac{1}{Vol} \sum_{all \ faces} \left(A_c + A_v \right)_{local}^{(q-1)} S - R^{(q-1)} \right) \right] \overline{\Delta Q}^{p}{}_{local}$$

$$= \Delta Q^q - \varepsilon \Delta t \left(\frac{1}{Vol} \sum_{all \ faces} \left(A_c + A_v \right)_{NB}^{(q-1)} S \right) \overline{\Delta Q}^{p-1}{}_{NB}$$

In primitive variables we use only part of the Jacobian entries:

$$R_J\{\omega,T,u\}=\begin{pmatrix} \frac{\partial\dot{\omega}_1}{\partial\omega_1} & 0 & 0 & 0 & \frac{\partial\dot{\omega}_1}{\partial T} \\ 0 & \frac{\partial\dot{\omega}_2}{\partial\omega_2} & 0 & 0 & \frac{\partial\dot{\omega}_2}{\partial T} \\ 0 & 0 & \frac{\partial\dot{\omega}_3}{\partial\omega_3} & 0 & \frac{\partial\dot{\omega}_3}{\partial T} \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{\partial\dot{T}}{\partial T} \end{pmatrix}$$

The implementation in FLDYNS requires the Jacobian in $\{\rho,P,u\}$ variables given by

$$R_J\{\rho,P,u\}=\begin{pmatrix} \frac{\partial\dot{\omega}_1}{\partial\omega_1}-\frac{\partial\dot{\omega}_1}{\partial T}\frac{TW}{\rho} & -\frac{w_1}{w_2}\frac{\partial\dot{\omega}_1}{\partial T}\frac{TW}{\rho} & -\frac{w_1}{w_3}\frac{\partial\dot{\omega}_1}{\partial T}\frac{TW}{\rho} & 0 & \frac{w_1W}{\rho R}\frac{\partial\dot{\omega}_1}{\partial T} \\ -\frac{w_2}{w_1}\frac{\partial\dot{\omega}_2}{\partial T}\frac{TW}{\rho} & \frac{\partial\dot{\omega}_2}{\partial\omega_2}-\frac{\partial\dot{\omega}_2}{\partial T}\frac{TW}{\rho} & -\frac{w_2}{w_3}\frac{\partial\dot{\omega}_2}{\partial T}\frac{TW}{\rho} & 0 & \frac{w_2W}{\rho R}\frac{\partial\dot{\omega}_2}{\partial T} \\ -\frac{w_3}{w_1}\frac{\partial\dot{\omega}_3}{\partial T}\frac{TW}{\rho} & -\frac{w_3}{w_2}\frac{\partial\dot{\omega}_3}{\partial T}\frac{TW}{\rho} & \frac{\partial\dot{\omega}_3}{\partial\omega_3}-\frac{\partial\dot{\omega}_3}{\partial T}\frac{TW}{\rho} & 0 & \frac{w_3W}{\rho R}\frac{\partial\dot{\omega}_3}{\partial T} \\ 0 & 0 & 0 & 0 & 0 \\ \frac{RT}{w_1}\left(\frac{\partial\dot{\omega}_1}{\partial\omega_1}-\frac{T}{P}\frac{\partial\dot{P}}{\partial T}\right) & \frac{RT}{w_2}\left(\frac{\partial\dot{\omega}_2}{\partial\omega_2}-\frac{T}{P}\frac{\partial\dot{P}}{\partial T}\right) & \frac{RT}{w_3}\left(\frac{\partial\dot{\omega}_3}{\partial\omega_3}-\frac{T}{P}\frac{\partial\dot{P}}{\partial T}\right) & 0 & \frac{T}{P}\frac{\partial\dot{P}}{\partial T} \end{pmatrix}$$

where $$\frac{\partial\dot{P}}{\partial T}=RT\sum\frac{\partial\dot{\omega}_i}{\partial T}+\frac{\partial\dot{T}}{\partial T}\frac{P}{T}$$

5.5 Determination of the temperature from the internal energy

For a given internal energy e_0, we want to determine the temperature (and/or the pressure). Since for non-ideal gases the internal energy is a non-linear function of the temperature, we have to solve it numerically using the Newton-Raphson method. We solve the equation $f(T)=e(T)-e_0=0$ iteratively:

$$T_{n+1}=T_n-\frac{f(T)}{f'(T)}=T_n-\frac{f(T_n)}{f'(T_n)}$$

since $\frac{de(T)}{dT}=c_v$ we get $T_{n+1}=\frac{c_v(T_n)T_n-e(T_n)+e_0}{c_v(T_n)}$

5.6 Results

5.6.1 1-D Steady State Reactive Flow

Our first implementation is a test case for a 1-d steady state, reactive flow. The test problem has been taken from NASA'S NPARC validation website. The boundary conditions and simple (1D) geometry are shown in figure 3. This example contains Hydrogen-Oxygen-Water reactions (+ radicals).

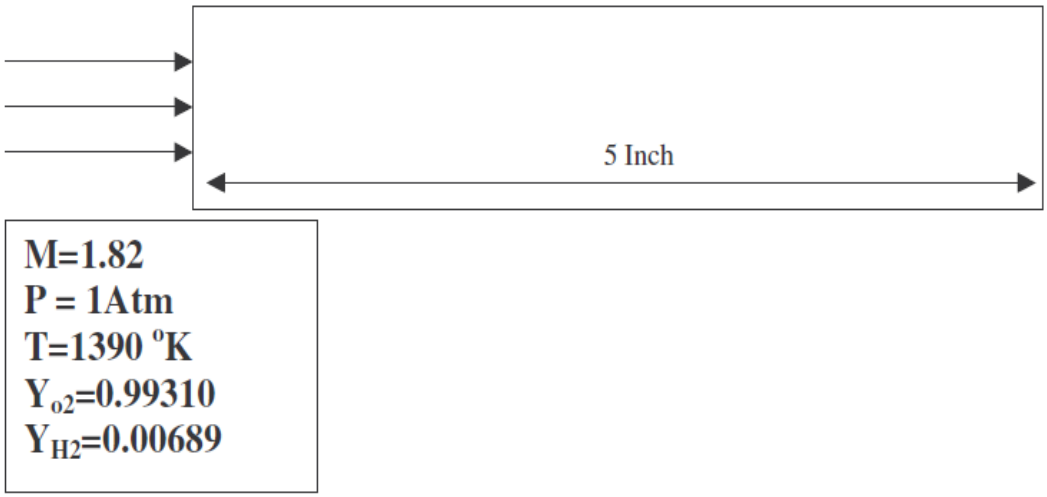

Figure 3: NPARC validation test geometry and BC.

The axial distribution of temperature and, water mass fraction are shown in figures 4-5. In these figures we also compare the results to those obtained from the CFD++ code. The time delay between the solutions of the test code and from CFD++ can be explained by a numerical ignition delay.

The convergence history for various CFL numbers and implicit factors ε is shown in figure 6. In the NPARC website this example was computed with CFL=0.05. With the preconditioner the asymptotic CFL used was 16. The convergence improves as we decrease the implicit factor ε. For ε below 0.4 the computation became unstable.

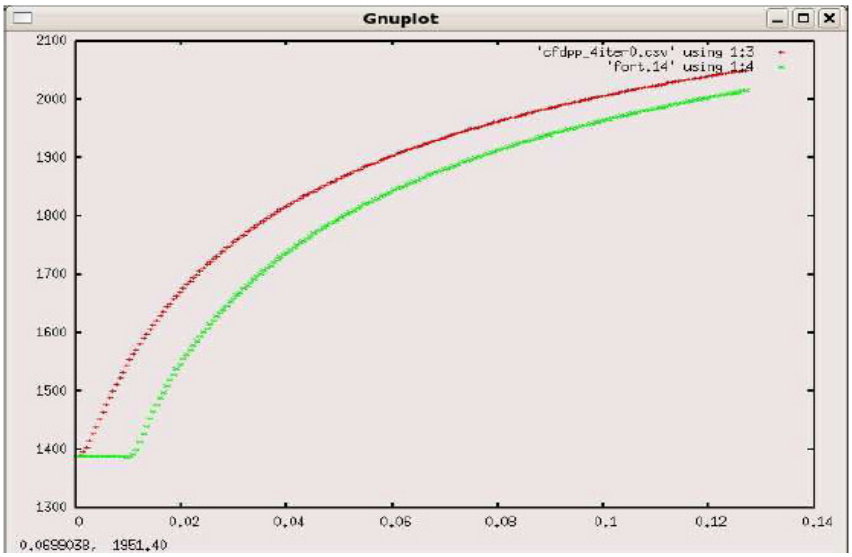

Figure 4: Temperature distribution. CFD++ in green, Test codes in red.

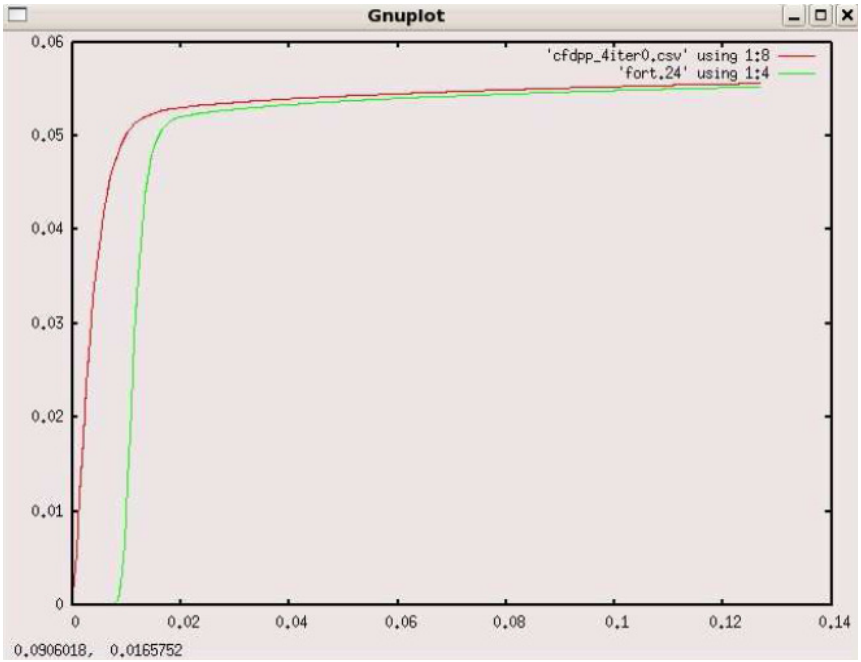

Figure 5: water (H2O) distribution. CFD++ in green, Test codes in red.

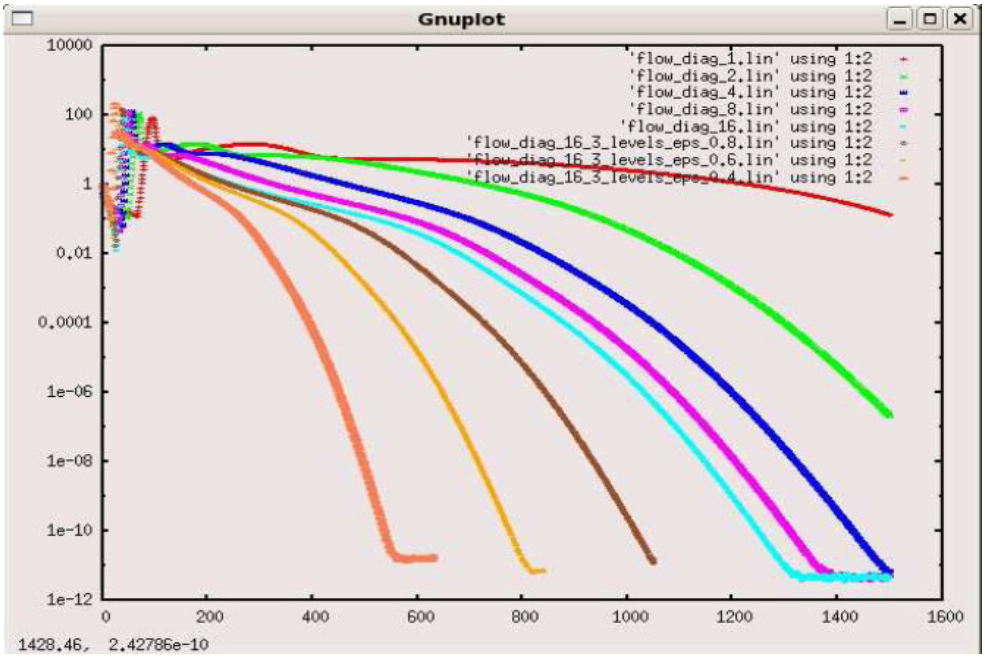

Figure 6: Convergence history. CFL 1–red, CFL 2–green, CFL 4–blue, CFL 8–magenta, CFL 16-light blue (all with ε=1). CFL 16; ε=0.8 – brown, CFL 16; ε=0.6 – yellow, CFL 16; ε=0.4 – orange.

5.6.2 Rapid expansion diffuser

In this problem, a high Mach steam is injected into a nozzle-like device. The detailed geometry and boundary descriptions are shown in figure 7. This example has been taken from [8]. The gas mixture contains Hydrogen-Oxygen-Water reactions (+ radicals). Figure 8 presents the density convergence history. The chemical reactions create a shock close to the entrance of the nozzle. This shock does not occur in the non-reactive case. Figure 9 presents the contour map of the temperature, Mach number and some of the species mass fraction. Good agreement between our results and those presented in [8] are obtained.

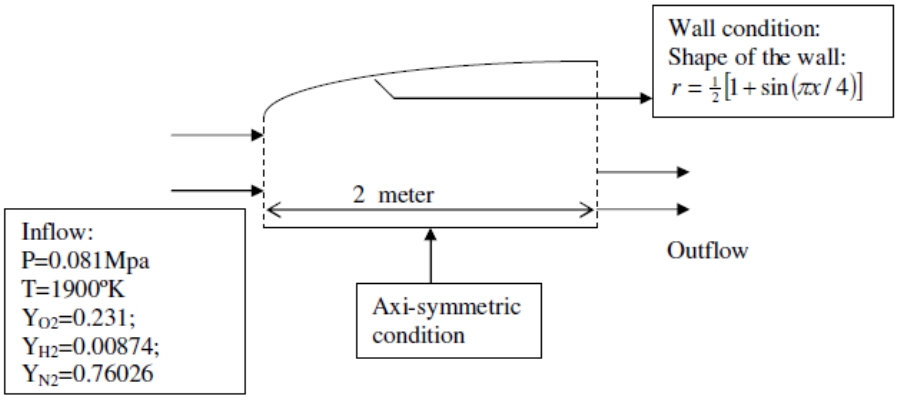

Figure 7: Problem definition

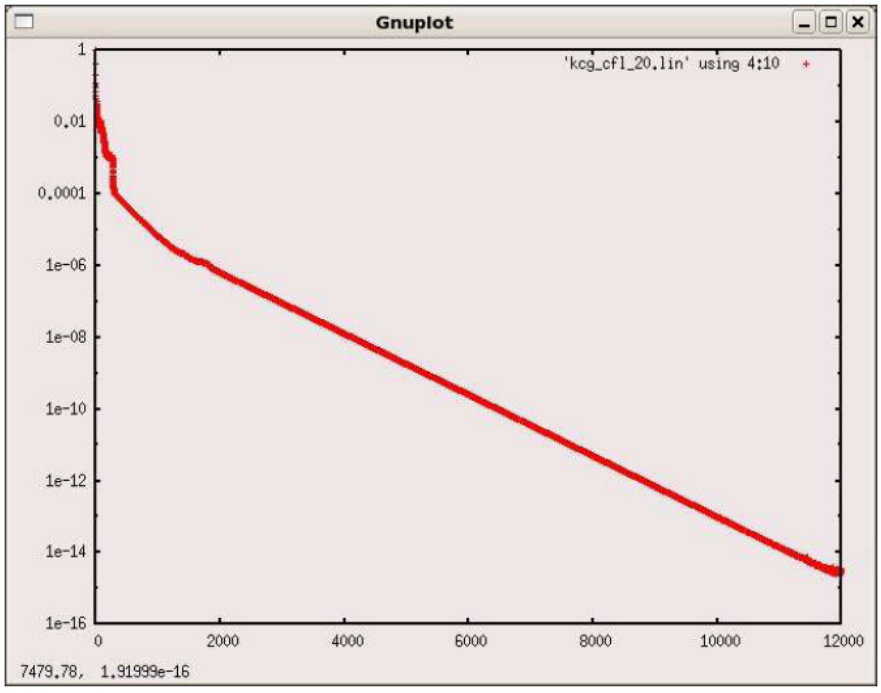

Figure 8: convergence history for CFL 20 without multigrid

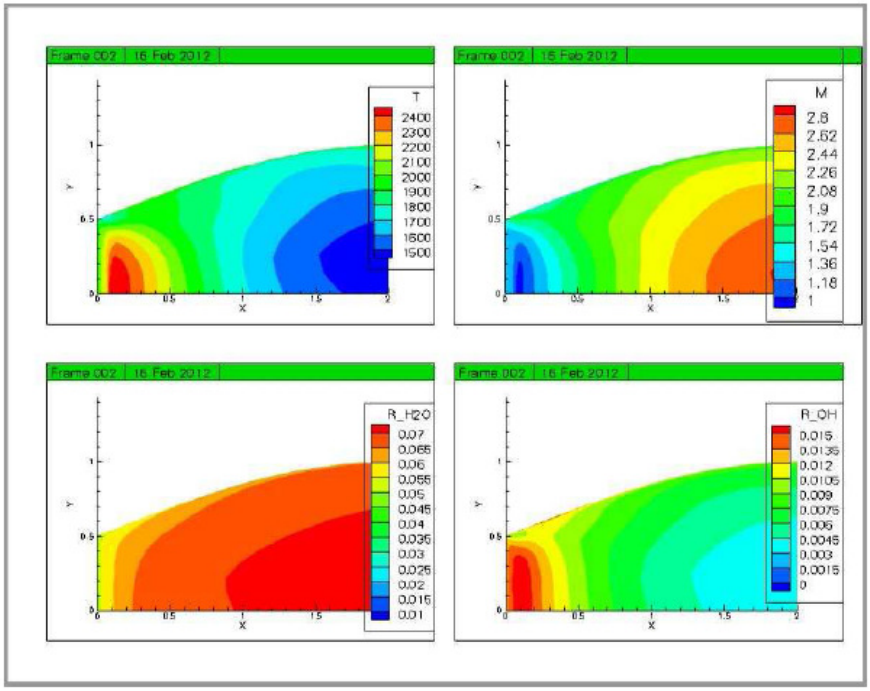

Figure 9: Temperature, Mach, species mass fraction contours

5.6.3 Blunt projectile

We next consider a blunt projectile is flying in a stochiometric mixture of hydrogen and oxygen. The convergence history is presented in figure 10. The rise of the temperature behind the bow shock causes the ignition. The density contours map is shown in figure 11. The axial distribution of the species mass fraction and the temperature are shown in figure 12 and 13 and compared to the results from [9].

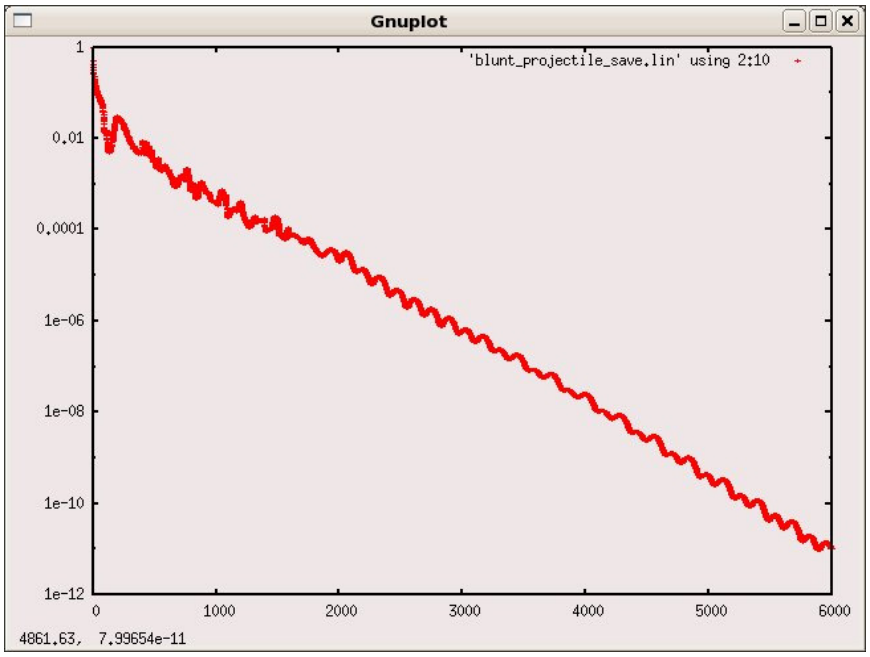

Figure 10: Convergence history without multigrid

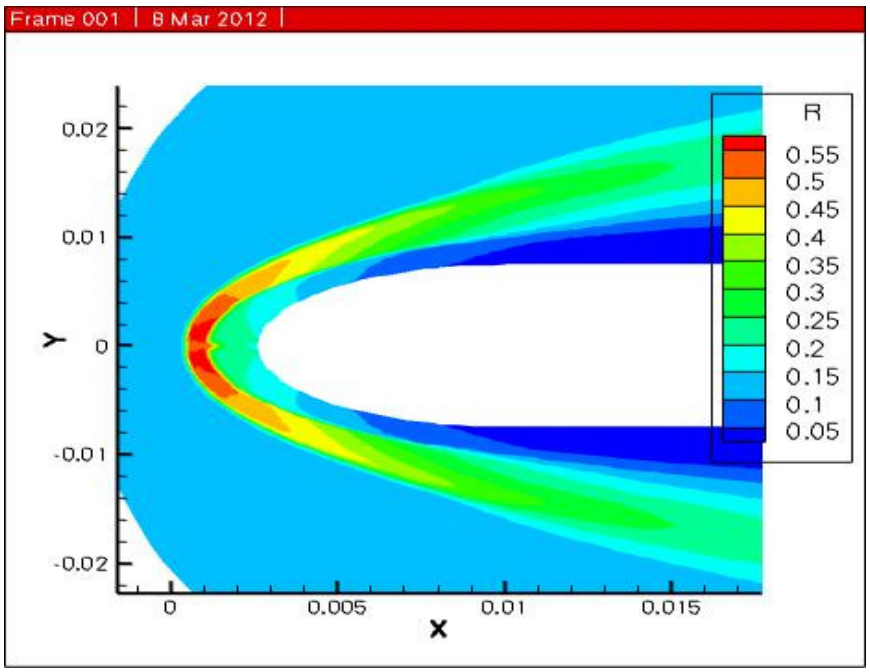

Figure 11: density contours map

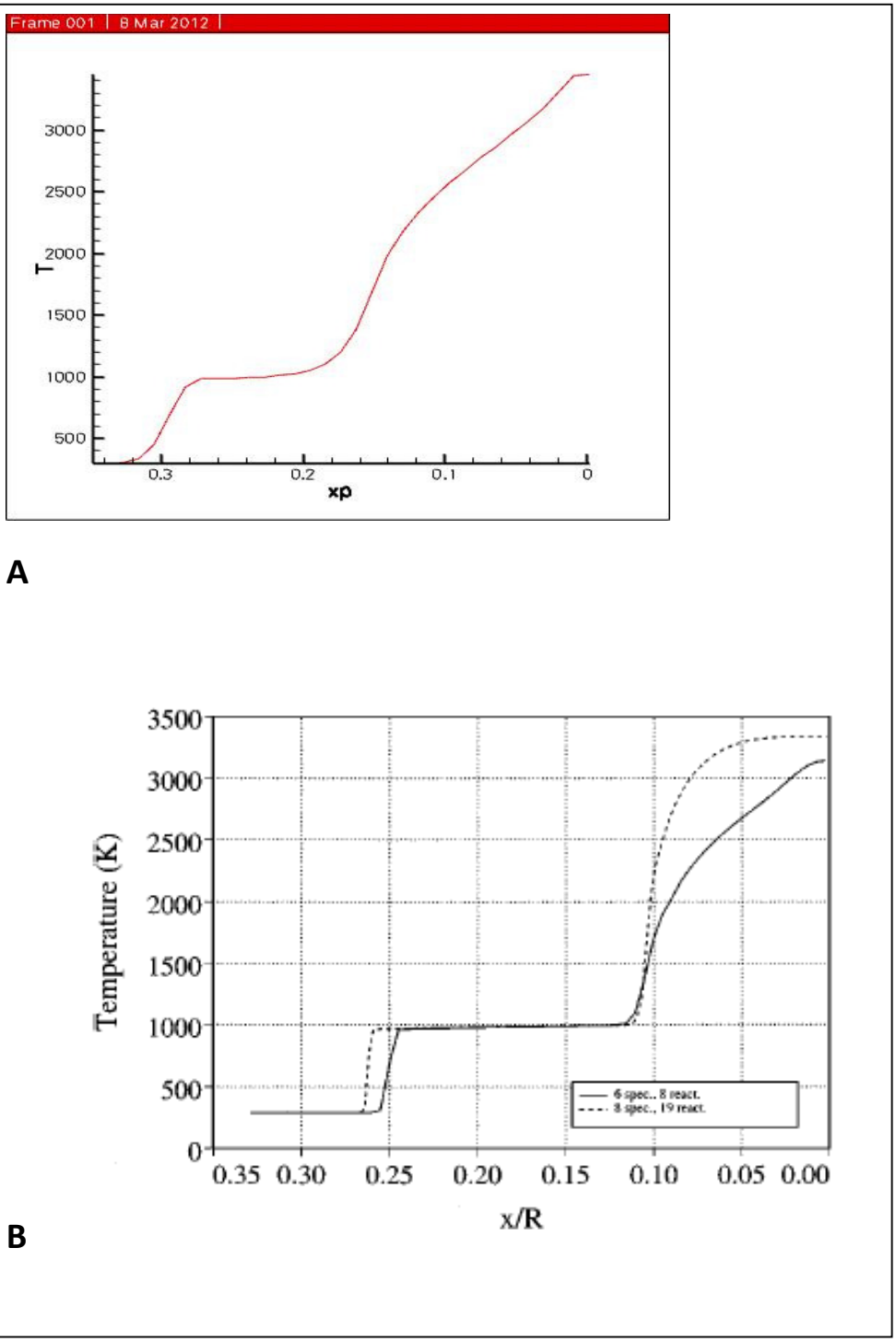

Figure 12: Axial distribution of temperature. A – FLDYNS. B – Sheffer 1998

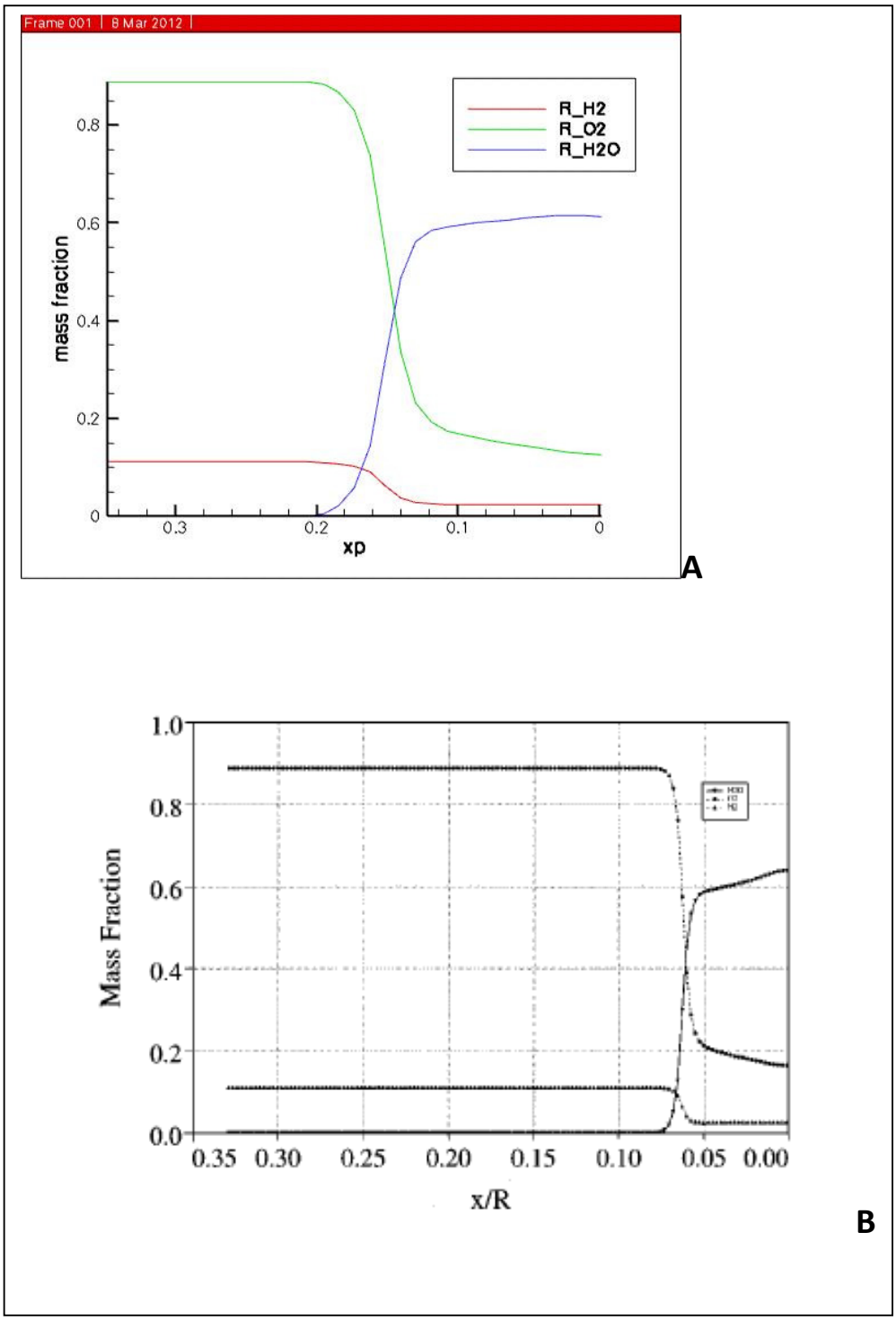

Figure 13: Axial distribution of species mass fraction. A – FLDYNS. B – Sheffer 1998

5.6.4 Rocket motor plume

We now calculate the rocket motor plume exiting from the motor nozzle into a low Mach number free stream flow. The plume boundary conditions are defined on the nozzle throat where the flow velocity

is sonic and the species mass fractions are given. The species used for this problem are: H, O, OH, H2, O2, CO, CO2, H2O, HCL and N2. The reactions are described in [7]. Figure 14 shows the convergence history of the density (red) and the turbulent kinetic energy (green). Figures 15-18 show the Mach contours, temperature contours, k contours and species mass fraction, respectively.

The calculation uses sequencing of two levels of coarse grids and three levels of multigrid on the finest level. A second order upwind scheme with a Sweby limiter is used. The fluid CFL is 100,000 and turbulent CFL is 200.

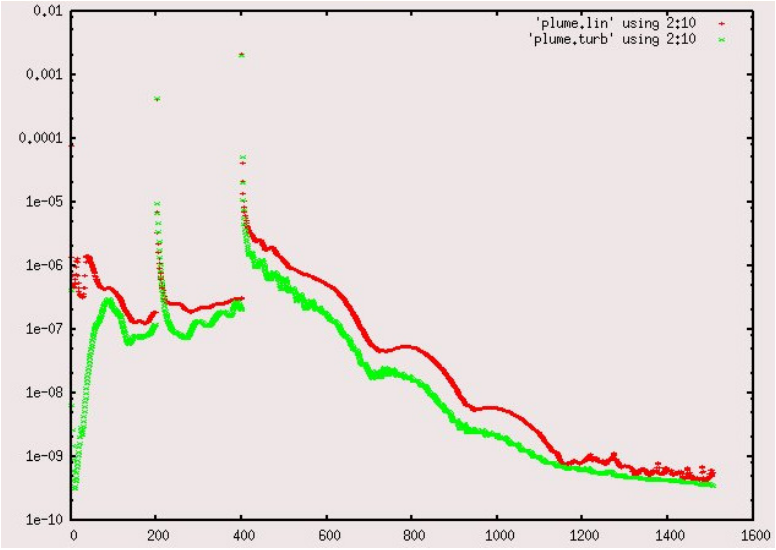

Figure 14: Convergence history of the density (red) and the turbulent kinetic energy (green)

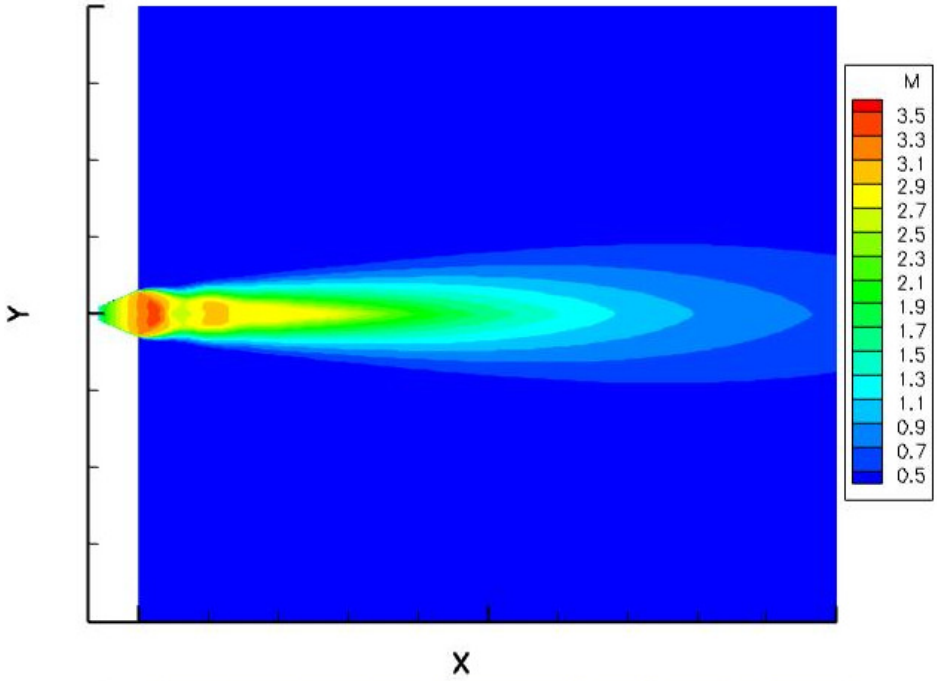

Figure 15: Mach contours of the plume

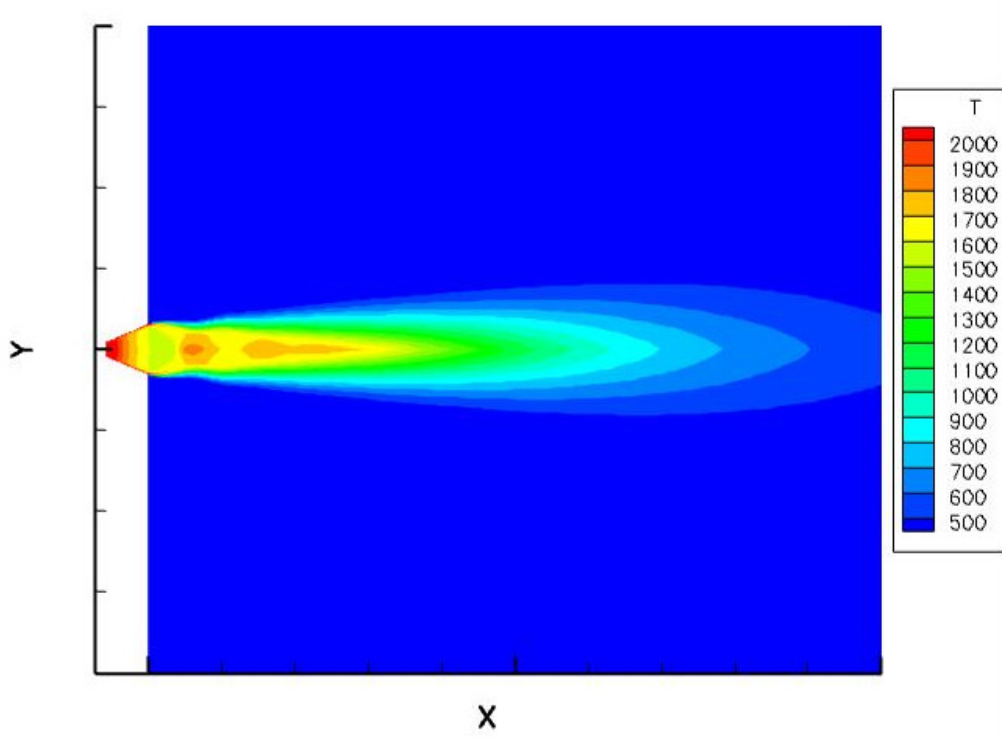

Figure 16: Temperature contours of the plume

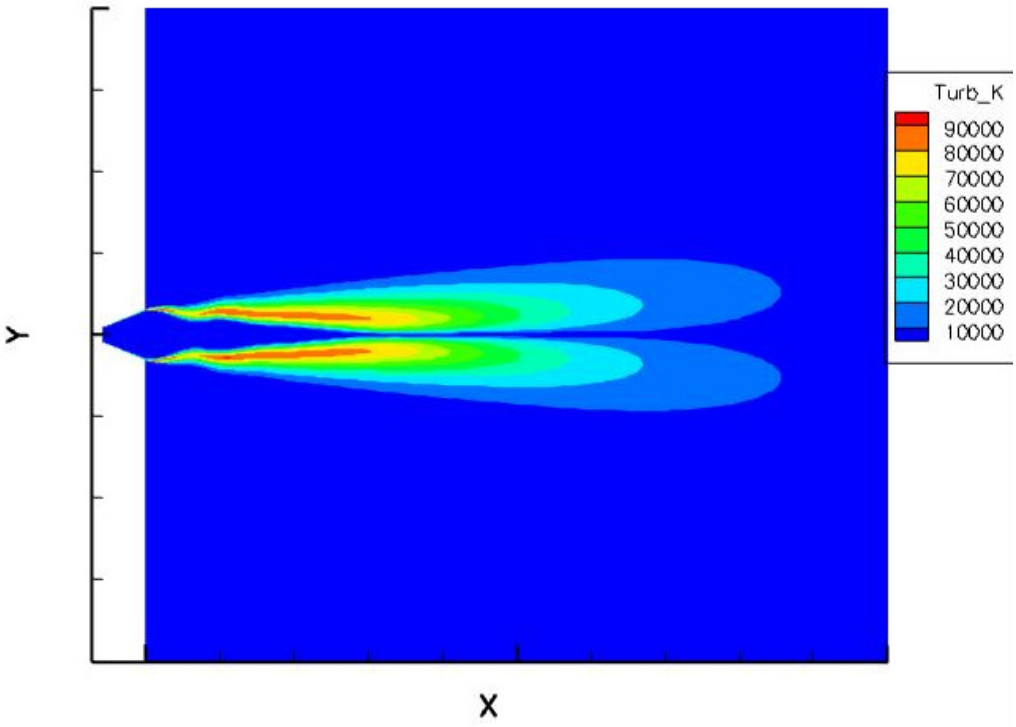

Figure 17: Turbulent kinetic energy contours of the plume

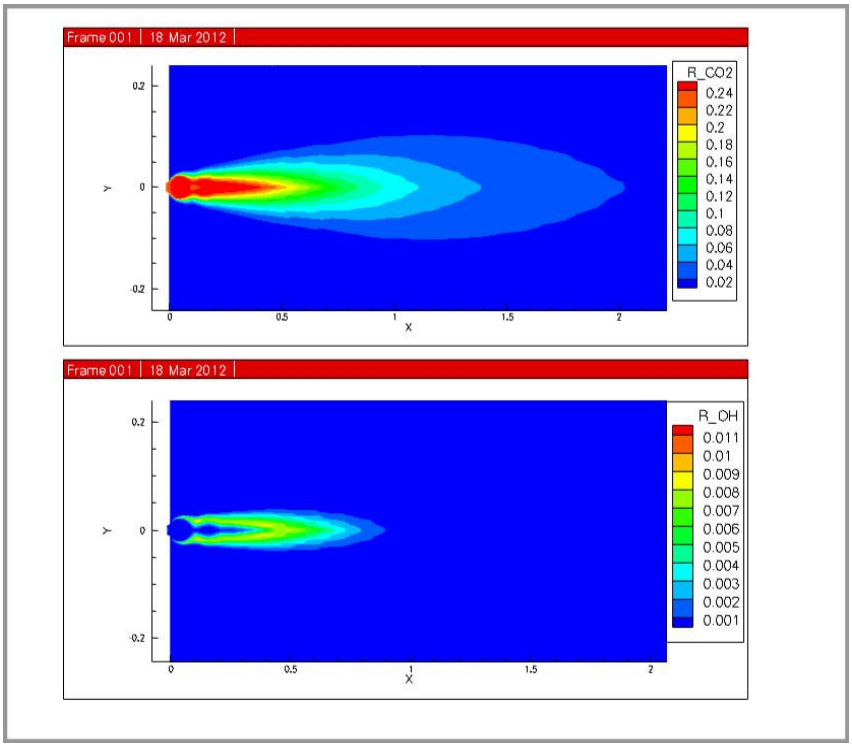

Figure 18: CO2 and OH mass fraction contours of the plume

6 Conclusions

The RK/Implicit smoother has been extended to solve the turbulence k-ω/SST model equations. We solved for flow around a transonic RAE2822 wing and a supersonic jet plume using the combined fluid and SST-turbulence equations. We obtained excellent convergence rates and an accurate solution.

For reactive flow the source terms introduce stiffness into the Navier-Stokes equations. We introduced the source term Jacobian into the RK/Implicit smoother to solve the problem. The algorithm and complicated cases such as reactive rapid diffuser and reactive blunt projectile are presented for viscous reactive flow and the turbulent, reactive rocket motor plume with the k-ω/SST turbulence model is presented and very good results were obtained.

References

[1]. C-C. Rossow, "Convergence Acceleration for Solving the Compressible Navier-Stokes Equations", AIAA J. 44: 345--352, 2006.

[2]. R. C. Swanson, E. Turkel, C. -C. Rossow and V.N. Vatsa, "Convergence Acceleration for Multistage Time-Stepping Schemes", AIAA--2006-3523.

[3]. R. C. Swanson, E. Turkel and C. -C. Rossow, "Convergence Acceleration of Runge-Kutta Schemes for Solving the Navier-Stokes Equations", J. Comp. Physics 224:365--388, 2007.

[4]. O. Peles, S Yaniv and E. Turkel, “Convergence Acceleration of Runge-Kutta Schemes using RK/Implicit Smoother for Navier-Stokes Equations with SST Turbulence”,Proceedings of 52nd Israel Annual Conference on Aerospace Sciences, 2012.

[5]. A. Jameson, W. Schmidt and E. Turkel, "Numerical Solutions of the Euler Equations by Finite Volume Methods Using Runge-Kutta Time-Stepping Schemes", AIAA Paper 8l-l259, 1981.

[6]. C. Swanson and E. Turkel, "On Central Difference and Upwind Schemes", J. Comp. Physics 292--306, 1992.
[7]. G. Avital, Y. Cohen, L. Gamss, Y. Kanelbaum, J. Macales, B. Triemann, S. Yaniv, M. Lev, J. Stricker, A. Sternlieb, "Experimental and Computational Study of Infrad Emission from Underexpanded Rocket Exhaust Plumes", Journal of Termophysics and Heat Trannsfer, Vol. 15, No. 4, October-December 2001.
[8]. T. J. Chung, "Computational Fluid Dynamics," Cam- bridge University Press, Cambridge, 2002
[9]. S. G. Sheffer, L. Martinelli, A. Jameson, "An Efficient Multigrid Algorithm for Compressible Reactive Flows", Journal of Computational Physics 144, 484–516 (1998)