

LA-SUB--99-9-Pt. 2

Discrete approximations of detonation flows with structured
detonation reaction zones by discontinuous front models: A
program burn algorithm based on Detonation Shock Dynamics

John. B. Bdzil¹

¹Los Alamos National Laboratory, Los Alamos, New Mexico, 87545

Thomas L. Jackson²

²Center for Simulation of Advanced Rockets, University of Illinois, Urbana, IL 61801

and

D. Scott Stewart³

³Theoretical and Applied Mechanics, University of Illinois, Urbana, IL 61801, USA *

RECEIVED

JUL 19 1999

OSTI

February 2, 1999

Abstract

In the design of explosive systems the generic problem that one must consider is the propagation of a well-developed detonation wave sweeping through an explosive charge with a complex shape. At a given instant of time the lead detonation shock is a surface that occupies a region of the explosive and has a dimension that is characteristic of the explosive device, typically on the scale of meters. The detonation shock is powered by a detonation reaction zone, sitting immediately behind the shock, which is on the scale of 1 millimeter or less. Thus, the ratio of the reaction zone thickness to the device dimension is of the order of 1/1000 or less. This scale disparity can lead to great difficulties in computing three-dimensional detonation dynamics. An attack on the dilemma for the computation of detonation systems has led to the invention of sub-scale models for a propagating detonation front that we refer to herein as program burn models. The program burn model seeks not to resolve the fine scale of the reaction zone in the sense of a DNS simulation. The goal of a program burn simulation is to resolve the hydrodynamics in the inert product gases on a grid much coarser than that required to resolve a physical reaction zone. We first show that traditional program burn algorithms for detonation hydrocodes used for explosive design are inconsistent and yield incorrect shock dynamic behavior. To overcome these inconsistencies, we are developing a new class of program burn models based on detonation shock dynamic (DSD) theory. It is hoped that this new class will yield a consistent and robust algorithm which reflects the correct shock dynamic behavior.

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

MASTER

*Corresponding Author: Theoretical and Applied Mechanics, University of Illinois, 216 Talbot Laboratory, 104 S. Wright St., Urbana, IL 61801, USA (FAX: 217-244-5707, email: dss@uiuc.edu)

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.

1 Introduction

In the design of explosive systems the generic problem that one must consider is the propagation of a well-developed detonation wave sweeping through an explosive charge with a complex shape. At a given instant of time the lead detonation shock is a surface that occupies a region of the explosive and has a dimension that is characteristic of the explosive device, typically on the scale of meters. The detonation shock is powered by a detonation reaction zone, sitting immediately behind the shock, which is on the scale of 1 millimeter or less. Thus, the ratio of the reaction zone thickness to the device dimension is of the order of 1/1000 or less. This scale disparity can lead to great difficulties in computing three-dimensional (3D) detonation dynamics.

Assume (as we do for the rest of the paper) that the physical problem of modeling the dynamic propagation of the detonation and the motion of the reacted products in the following flow is completely described by a solution to the compressible Euler equations for a reactive flow, with a specified equation of state for the explosive and reaction rate of the form

$$e = e(p, v, \lambda), \quad r = r(p, v, \lambda),$$

where p, v, λ are the pressure, specific volume and the progress variable of chemical reaction. Note that $\lambda = 0$ corresponds to unreacted explosive and $\lambda = 1$ corresponds to completely reacted explosive. The prediction of the detonation dynamics can be achieved in principle by a direct numerical solution (DNS) of the Euler equations. In order to get a high quality solution to the reactive Euler equations, it is essential to have enough points in the reaction zone. Unfortunately even with modern algorithms, as many as 20-50 cells in the streamwise direction may be required to resolve the detonation reaction zone to sufficient accuracy so as to compute the detonation speed. When one then considers the consequences of such a fine scale for the reaction zone, combined with the requirement for global temporal and spatial accuracy in the meter-sized domain of the engineering device, huge computational resources are required [1] (even given today's TeraFlop parallel computing resources) for DNS of a detonation wave sweeping through a system.

The computational barrier to 3D design of explosive systems through direct simulation of the reactive Euler equations is not newly discovered, and dates back to the use of computers to design explosive systems that started systematically shortly after WWII. A dilemma of sorts presents itself. One needs to try to make predictions in engineering systems but one cannot overcome the stiff computational requirements needed to compute on the engineering device scale. One could compute DNS simulations that are resolved for very small dimensions, but those are at a minimum at least two orders of magnitude smaller than the engineering system scale. The dilemma posed above associated with trying to solve a physically correct but computationally intractable model is similar in spirit to direct simulation of turbulence on engineering device scales. In that discipline the need to resolve the physics of turbulence on larger engineering scales has led to the invention of classes of sub-scale models for turbulence and most recently to large eddy simulation.

An attack on the dilemma for the computation of detonation systems has led to the invention of sub-scale models for a propagating detonation front that we refer to herein as program burn (PB) models. The program burn model seeks *not* to resolve the fine scale of the reaction zone in the sense of a DNS simulation. The goal of a PB simulation (PBS) is to resolve the hydrodynamics in the inert product gases on a grid much coarser than that required to resolve a physical reaction zone. Thus a PBS must deposit a prescribed

amount of energy (and more generally mass and momentum) into a *very few* number of computational cells behind a pre-calculated shock front. The effective reaction zone in a PBS is always the region behind a pre-calculated shock front where source terms are added to account for the deposition of energy. For practical reasons, the effective reaction zone is always constrained to be a finite number of cells thick (between one and four say).

The region where the source terms contribute, in the limit of zero cell thickness, limits to a sharp front across which there are jumps in the dependent state variables. The program burn source doses, while historically prescribed purely by the prescription of the discrete algorithm used in a particular code, must limit to a delta function source centered at the location of the sharp front, which is then externally prescribed by pre-calculating the shock location. The delta function source terms must, of course, be represented in the partial differential equations that represent the program burn model, independent of its discretization and the algorithms used to solve it. One thing is clear from this discussion, the solutions of the reactive Euler equations *are not* solutions of the equations of the program burn model.

In this paper we consider the following problem: How does one make consistent and robust discrete approximations of physical detonation flows with a finite length reaction zone as modeled by the reactive Euler equations, with a discrete approximation to a program burn model for which the reaction zone and shock is collapsed entirely to a single discontinuous front?

The whole scheme where a Program Burn model have solutions that are in some sense close to those of the Euler equations for a reactive flow depends very much on the accuracy of the approximate theory in regards to the shock dynamics. This issue must be decided irrespective of numerics. In section 2 we briefly present direct numerical simulations (DNS) of the reactive Euler equations that are to be used as the benchmark calculations for the rest of the paper. The geometry considered will be either planar, cylindrical or spherical. For cylindrical/spherical geometry, curvature of the lead shock is present. In section 3 we compare the solutions obtained from DNS to the recent asymptotic theory of detonation shock dynamics (DSD), a key ingredient of the more modern implementations of program burn. In section 4 we regress somewhat by presenting the traditional pressure-based program burn (TPBPB) model; such a description is essential for understanding the rest of the paper. In addition, some numerical calculations are presented showing the strengths and weaknesses of the model. Section 5 contains a short description of the more modern implementation of TPBPB, called the modified pressure-based program burn model (MPBPB). There does not exist any reference which describes MPBPB, and so is presented here for the first time. Solutions obtained from MPBPB are compared with solutions obtained from TPBPB and DNS. The essential difference between TPBPB and MPBPB is that TPBPB uses a Huygen's construction for the shock propagation rule (shock propagates with the Chapman-Jouguet speed), while if curvature is present, MPBPB uses a propagation rule based on DSD. It is shown that even with this modification in the propagation law, certain features of the downstream states are incorrectly captured. Attempts to resolve this inconsistency is the major focus of section 6. A class of new algorithms are proposed based on DSD theory, which we shall refer to as DSD program burn (DSDPB) models. Finally, conclusions are given in section 7.

2 Direct Numerical Simulations

In this section we present the reactive Euler equations that will be used as benchmark calculations for which the program burn models, presented in subsequent sections, will be compared too. We therefore assume that the DNS calculations are "exact", and that any differences in solution structure will be due to the various approximations inherent in the program burn models themselves.

For the DNS calculations, the conservative formulation of the reactive Euler equations are given by

$$\vec{U}_t + \vec{F}_x = \vec{G} + \vec{q}r, \quad (1)$$

where

$$\vec{U} = [\rho, \rho u, E, \rho\lambda]^T, \quad (2)$$

$$\vec{F} = [\rho u, \rho u^2 + p, u(E + p), \rho u\lambda]^T, \quad (3)$$

$$\vec{G} = -\frac{j}{x} [\rho u, \rho u^2, u(E + p), \rho u\lambda]^T, \quad \vec{q} = [0, 0, 0, 1]^T, \quad (4)$$

where ρ is the density, p the pressure, u the velocity, E the total energy defined by

$$E = \rho \left(e + \frac{1}{2} u^2 \right). \quad (5)$$

e the specific internal energy, and λ the mass fraction of the deficient component ($\lambda = 0$ for unreacted material, $\lambda = 1$ for completely reacted material). The geometric source terms from the flow divergence are represented explicitly by \vec{G} . The choice of j determines the geometry; $j = 0$ for planar, $j = 1$ for cylindrical, or $j = 2$ for spherical geometry. If one assumes a cylindrical/spherical shock, the shock total curvature κ is related to the radius x from the center of the coordinate system by $\kappa = j/x$.

To close the system, constitutive laws for the internal energy and the reaction rate must be given. For illustration purposes, we take the example of a condensed phase explosive considered in [1]. The equation of state is taken to be that of an ideal gas

$$e = \frac{p}{\rho(\gamma - 1)} - Q\lambda. \quad (6)$$

where γ is the ratio of specific heats and Q is the heat of reaction for the detonation. The reaction rate is given by

$$r = 2.5147 \mu s^{-1} (1 - \lambda)^{1/2}. \quad (7)$$

The values $Q = 4mm^2/\mu s^2$ and $\gamma = 3$ are taken, with upstream conditions $p_o = 0$, $\rho_o = 2g/cc$ and $u_o = 0$. These values give a Chapman-Jouguet detonation speed of $D_{CJ} = 8mm/\mu s$, and a steady-state one-dimensional reaction zone length of $4mm$.

To carry out the DNS, these equations are solved by a high-resolution Euler solver, namely, a third-order TVD Runge-Kutta scheme with a fifth-order WENO spatial scheme, [6], [7] and [8]. The grid is assumed uniform with $\Delta x = 0.1mm$, which puts roughly 40 grid points in the reaction zone. Results for twice the number of grid points, and hence has 80 grid points in the reaction zone, gives essentially the same results. In all cases the CFL number was taken to be 0.4. Wave structures are presented in Figure 1 for the case of cylindrical geometry and in Figure 2 for the case of spherical geometry.

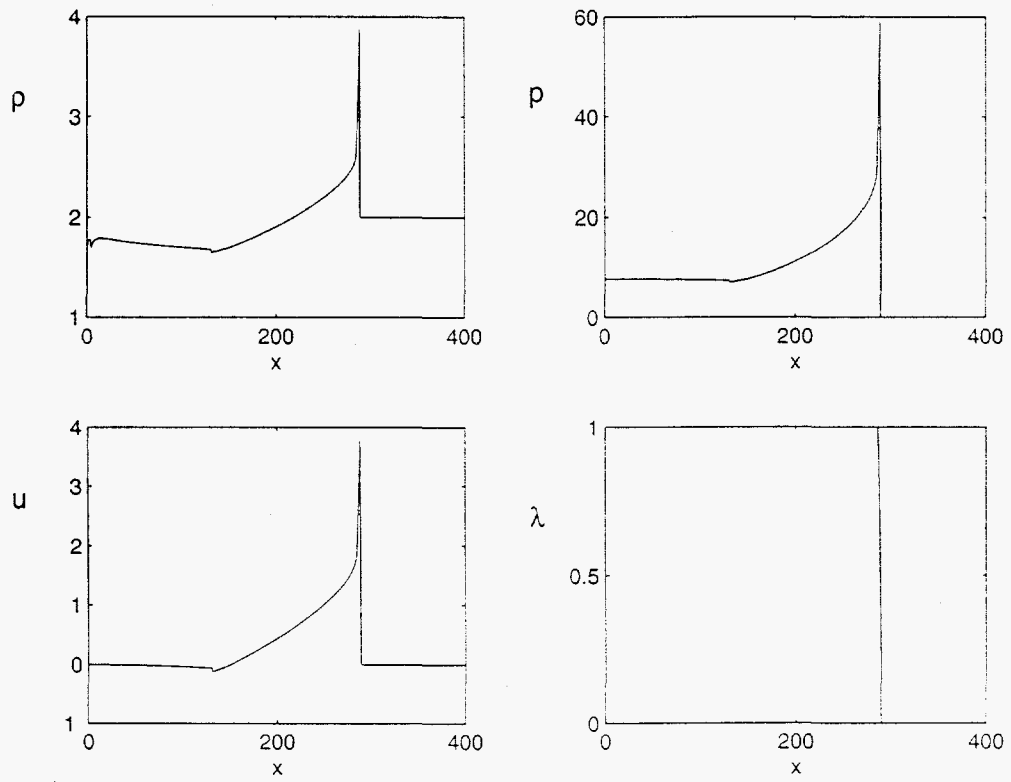


Figure 1: Plot of the structure for the case $j = 1$ (cylindrical) at time $t = 40\mu s$.

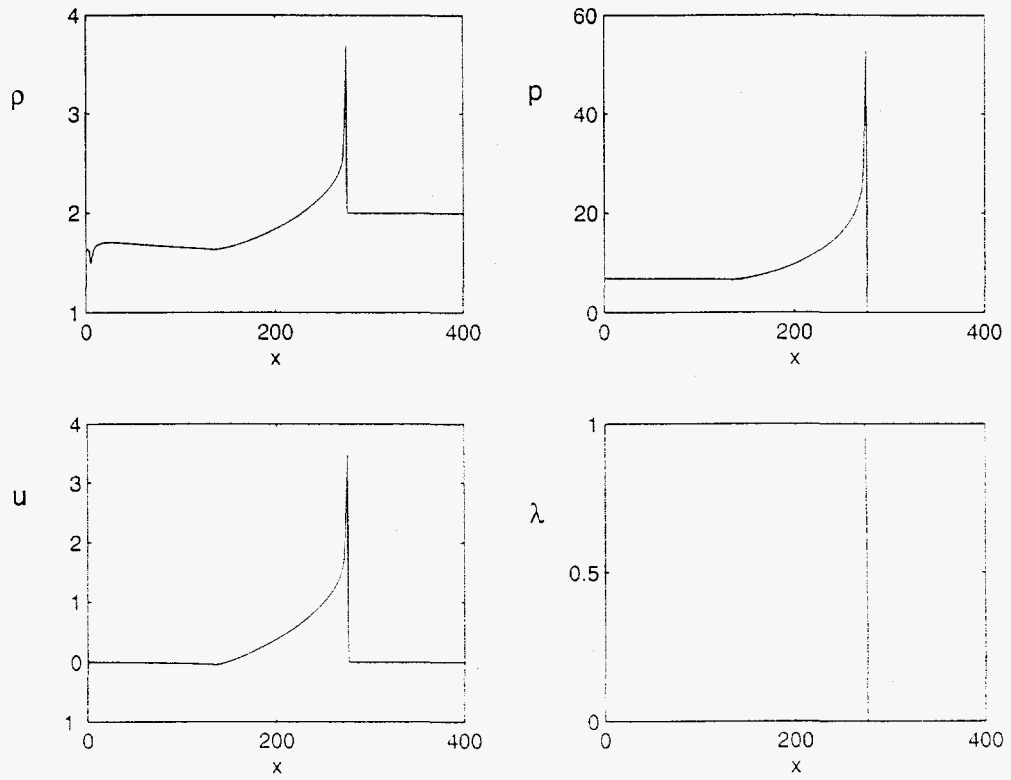


Figure 2: Plot of the structure for the case $j = 2$ (spherical) at time $t = 40\mu s$.

3 DSD asymptotic theory and comparison to DSD

In this section we briefly state the asymptotic theory of detonation shock dynamic (DSD) theory, a key ingredient of the program burn model that will be presented in subsequent sections. We also compare certain flow features between DSD theory and the DNS calculations of the reactive Euler equations presented in the previous section.

3.1 DSD Theory

Detonation shock dynamic (DSD) theory is an asymptotic theory which describes the motion of the detonation shock by means of a relation between the normal shock velocity D_n , the shock curvature κ , and their time derivatives. For a through review of the theory, its assumptions and limitations, see [4]. For our purposes here we shall only focus on the quasi-steady, one-dimensional theory. Following [10, 11], the relevant equations, consistent with the reactive Euler equations presented in a nearly integrable form that reflects the conserved first integrals of the governing equations if the flow were steady and plane, are

$$\frac{\partial(\rho U_n)}{\partial n} + \kappa \rho (U_n + D_n) = 0. \quad (8)$$

$$\frac{\partial(\rho U_n^2 + p)}{\partial n} + \kappa \rho U_n (U_n + D_n) = 0. \quad (9)$$

$$\frac{\partial}{\partial n} \left(e + pv + \frac{1}{2} U_n^2 \right) = 0 \quad (10)$$

$$\frac{\partial \lambda}{\partial n} = -\frac{1}{U_n} (r), \quad (11)$$

where n is the coordinate normal to the detonation front, and $U_n = u_n - D_n$ is the relative normal velocity in the shock-attached frame.

An alternative form of the energy equation, dubbed the *master equation*, is found by using the chain rule on $e(\rho, p, \lambda)$ in (10), using the mass equation to substitute for the spatial derivative of ρ , and then using the momentum equation to substitute for the spatial derivative of the pressure p . With the standard definition of the sound speed by for an ideal EOS, $c^2 = \gamma p / \rho$, one obtains

$$(c^2 - U_n^2) \frac{\partial U_n}{\partial n} = Qr(\gamma - 1) - \kappa c^2 (U_n + D_n). \quad (12)$$

The *generalized CJ conditions* follow from the master equation. When the flow is locally sonic and the velocity gradient is finite it follows that when

$$\eta \equiv c^2 - U_n^2 = 0, \quad (13)$$

the right hand side of (12) must also be zero, i.e.,

$$\Phi \equiv Qr(\gamma - 1) - \kappa c^2 (U_n + D_n) = 0. \quad (14)$$

The first condition is the *sonic condition*, while the second is the *thermicity condition*. These conditions hold for detonations that travel near or at the CJ detonation velocity. The simultaneous requirement that the sonic and thermicity conditions be satisfied require

that there is a relationship between κ and D_n . For such solutions one can find the sonic, or star (*), states. The solution of this nonlinear eigenvalue problem can be done numerically if desired and the star states can be found as a function of the local curvature κ . A plot of the star states is shown in Figure 3 for the condensed phase example of the previous section. Note that for $\kappa = 0$, the star states are the *CJ* states, and $D_n = D_{CJ}$.

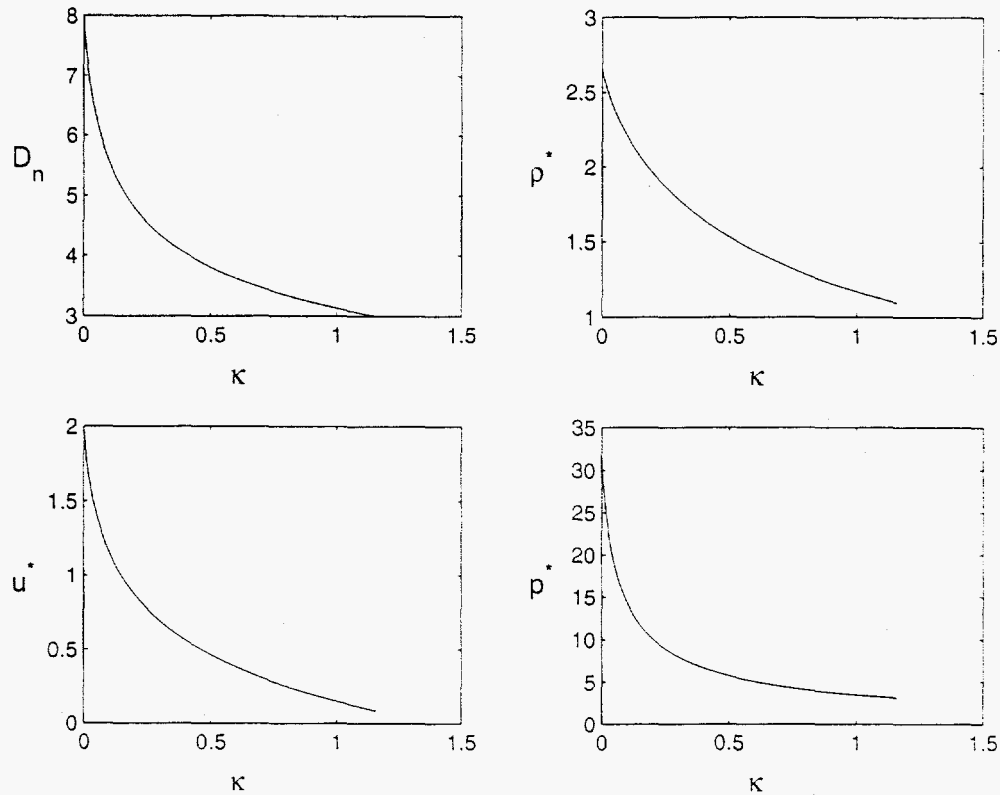


Figure 3: Plot of the shock speed D_n and the star states as a function of κ .

3.2 DSD-DNS Comparisons

Comparisons of DSD theory with direct numerical simulations (DNS) have been carried out in [1] for the case of a two-dimensional raterstick problem, a converging channel problem, and a diverging channel problem. In all three cases the shock front locations as computed from DSD theory and from DNS were compared and good agreement between the two was found. However, a more direct comparison between DSD and DNS can be carried out by considering cylindrical or spherical geometry where the curvature is explicitly known and

the equations are essentially one-dimensional. Such early comparisons were carried out by Buckiet, albeit with a very noisy random choice method for the DNS. [5].

To compare with DSD theory, we show in Figures 4 and 5 the shock speed D_n and the star states as a function of curvature κ for cylindrical and spherical geometry, respectively. In each figure, the solid curve corresponds to DSD theory, and the circles correspond to the DNS calculations. The wave front was determined to be the value at which the reaction progress variable λ was 0.1; the speed is then the time derivative. Note the good agreement for both cylindrical and spherical geometries as the curvature goes to zero, i.e., the long time solution. For large values of the curvature, the agreement between the two diverge, either due to the transient effects of the DNS calculations at the earlier times or due to the first order and quasi-steady approximation of DSD theory where the time derivatives have been ignored.

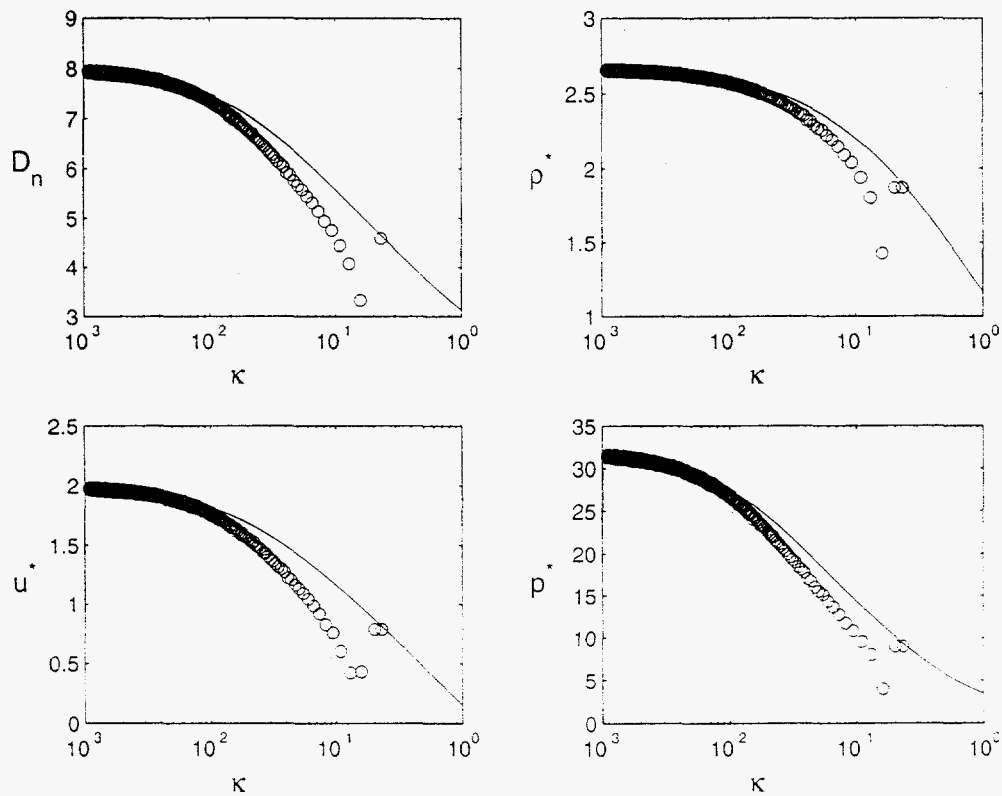


Figure 4: Plot of D_n and the star states as a function of κ for DSD (solid) and DNS (circles). Cylindrical geometry.

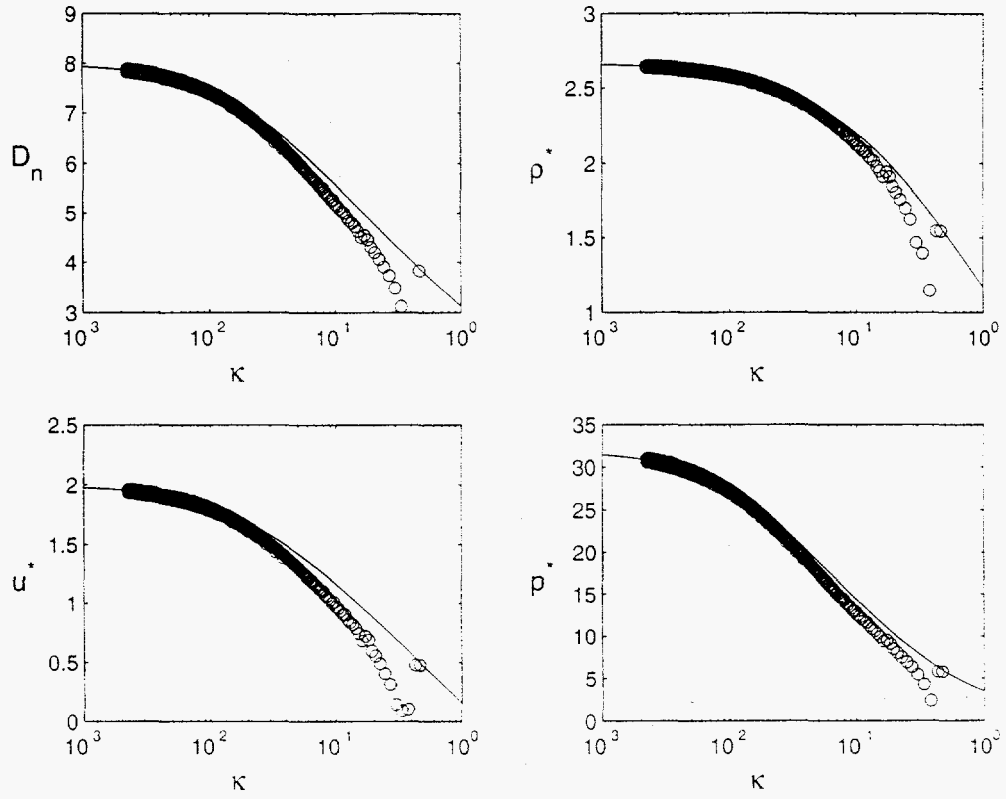


Figure 5: Plot of D_n and the star states as a function of κ for DSD (solid) and DNS (circles). Spherical geometry.

4 Traditionally implemented program burn models

In this section we discuss the basic ideas behind the implementation of program burn as it has traditionally been implemented in design hydrocodes used for explosive engineering. Although several versions exist, we shall discuss only one model, the traditional pressure-based program burn model (TPBPB). The other models have similar strengths and weaknesses, and only one model is sufficient to clarify the discussion.

Program burn was first posed as a numerical algorithm, not as a differential system. One of the earliest published references to an algorithm of this type is found in [2]. The algorithm has the following ingredients: i) There is a pre-determined, computational grid and a chosen algorithm for the inert hydrodynamics. The grid defines the domain of the explosive and the algorithms are used to solve the Euler equations for the (inert) explosive products. ii) A graded set of "burn-times", t_b , are assigned to each computational cell on the grid. The burn-times are the times that the detonation shock front crosses the coordinates of the initial position of the computational cell. The traditional way to compute the burn-times is to select the unreacted explosive geometry, pick the locus of an initial Chapman-Jouguet (CJ) detonation, and then compute the motion of the detonation shock emanating from the initial locus by means of a Huygen's construction. The Huygen's construction propagates the shock normal to itself at the constant CJ wave speed, D_{CJ} . iii) A cell-based algorithm either adds energy to designated burning cells or modifies the equation of state in cells during the interval of the shock passage over the cells, as dictated by the pre-calculated burn-times. The equation of state adjustment has been done in various ways through either increments in the pressure or specific volume.

In what follows we give a description of a traditional pressure-based program burn algorithm which modifies the equation of state in the burning cells. The definitions of the burn-fraction, the burn-time field, pre-calculated shock motion and modification of the equation of state are key ingredients of the model.

4.0.1 Burn-fraction

Based on a previously calculated assignment, each cell is assigned a burn-time, t_b . If the present time of a computational cell is below the burn-time, $t < t_b$, then the cell is not burning and the burn-fraction Y is assigned zero. If $t > t_b$, then the burn-fraction must be calculated. The burn-fraction is usually assigned to be the volume fraction of the undisturbed cell that has been crossed by the detonation shock at that time, and hence has a computed value, $0 < Y < 1$. The details of the computation depends on the specific grid and algorithm and whether the burn-times are stored at cell centers or at the nodes. If the whole cell has been crossed the burn-fraction is simply $Y = 1$.

4.0.2 Burn-time field

Once the burn-fraction algorithm is selected, the discrete field of burn-fractions can be pre-calculated from the discrete field of burn-times. While (as the grid is resolved) the burn-times limit to a piecewise continuous field in the domain of the unshocked explosive, the discrete burn-fraction field must limit to a singular Heaviside function which is attached to the contours of the burn-time field (i.e., the pre-calculated shock position). The burn-time field is pre-calculated and the traditional way to do this is to use a Huygen's construction. Thus, once the unreacted explosive geometry is selected, the initial locus of an initial CJ

detonation is picked, and the motion of the detonation shock that emanates from the initial locus is computed by means of a Huygen's construction.

4.0.3 Shock surface motion and the limits of discrete fields

The way to express these ideas mathematically is as follows. Let the burn-time field, which has a discrete representation on a grid which limits to a piecewise continuous field in the domain of the unreacted explosive, be given by

$$t_b(\vec{x}).$$

Then, at a fixed time t_0 , the shock locations are the contours of the burn-time field

$$\vec{x} = \vec{x}_s : t_0 = t_b(\vec{x}_s).$$

The limit of the discrete burn-fraction field at a time t_0 as the mesh is resolved is represented by the Heaviside function

$$H(\vec{x} - \vec{x}_s(t_0)).$$

As an example, consider a one-dimensional detonation wave propagating with constant positive speed D_{CJ} . Then, according to Huygen's construction, we have

$$\frac{dx_s}{dt} = D_{CJ}, \quad (15)$$

where x_s is the location of the detonation front at time t . Integrating we get the motion rule for the front

$$x_s(t) = x_o + tD_{CJ}, \quad (16)$$

where x_o is the initial position. The domain $x < x_o$ is assumed to be completely reacted, and is unreacted for $x > x_o$. This relationship can be inverted to yield the burn-time field

$$t_b(x_s) = \frac{x_s - x_o}{D_{CJ}}. \quad (17)$$

For the discrete approximation, let the numerical grid have a uniform mesh, x_i , with grid spacing Δx . Then the discrete version of the burn-field can be written as

$$t_b(x_i) = \frac{x_i - x_o}{D_{CJ}}, \quad \text{for } x_i > x_o. \quad (18)$$

Note that the burn-time is not defined for $x_i \leq x_o$, which indicates that this region of the flow field has already reacted. Also note that the burn-time is piecewise continuous in the unreacted domain.

For the prescription of the burn-fraction, which we shall denote by Y_i , we update Y_i according to the rule

$$Y_i = \begin{cases} 0 & x_i > x_s, \\ \frac{x_s - x_i}{\Delta x} & x_s - \Delta x < x_i < x_s, \\ 1 & x_i < x_s - \Delta x. \end{cases} \quad (19)$$

This particular description of the burn-fraction is defined over a single cell; more general rules can be defined without loss of generality. In the limit as $\Delta x \rightarrow 0$, we see that the burn-fraction approaches a Heaviside function. Figure 6 shows a sketch of the shock position as a function of time and a sketch of the burn-fraction Y . The use of the burn-fraction Y is described in more detail in the following section.

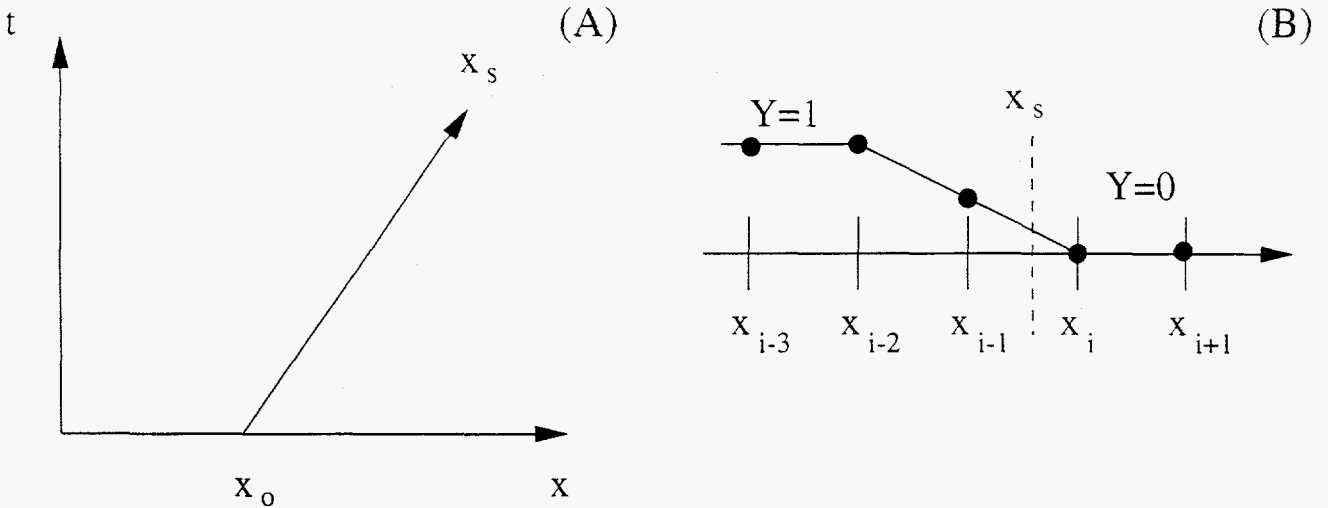


Figure 6: (A) Sketch of the shock location $x_s(t)$ as a function of time. (B) Sketch of the burn-fraction Y on a discrete grid.

4.0.4 Modification of the equation of state and apparent weak detonation structure

In the traditional pressure-based program burn algorithm one assumes an equation of state for the inert products

$$e_{products}(p, v) \equiv e(p, v).$$

Since condensed explosives are being considered, the initial pressures (one barr) are extremely small compared to the detonation pressures behind the lead shock (hundreds of Kilo-bars) such that the pressure ahead of the shock in the unreacted explosive can be considered to be zero. This is similar to the strong shock approximation. In a PBS in the burning cells, where the burn-fraction Y is between zero and one, the equation of state is modified by replacing p with p/Y to obtain $e_{products}(p/Y, v)$. This is equivalent to replacing the pressure with a partial pressure which is reduced by the burn-fraction for that cell. When $Y = 1$, the equation of state for the products is recovered. Finally, in the unburnt cells in the unreacted explosive one must give an energy that is consistent with the heat of detonation. This is done in the following way. One considers the standard Rankine-Hugoniot relations for a gasdynamic discontinuity for a steady Chapman-Jouguet discontinuity traveling at laboratory speed D_{CJ} . One then sets the energy datum e_0 in the unreacted explosive consistent with that algebra. The EOS for TPBPB can thus be written as

$$e = e_0 \{1 - H(\bar{x} - \bar{x}_s(t))\} + H(\bar{x} - \bar{x}_s(t)) e_{products}(p/Y, v). \quad (20)$$

An example of selecting e_0 is presented in the following section.

Thus, during a PBS, as one looks at the pressure variation across the shock, the pressure starts out from zero and is brought up to a high value near the CJ-pressure. Indeed, when

the burn-fraction Y is zero then the pressure is necessarily assumed to be zero, in fact the scheme computes the pressure based on an assumed equation of state and so the underlying hydrodynamic algorithm increments the pressure in such a way that the internal energy is assumed to be *finite*. A simple conclusion is that the effective reaction zone structure of traditional program burn starts at the unreacted state at the ambient pressure, and *not* at the shock state. If the program burn algorithm can be interpreted in terms of an effective distributed rate law, then the corresponding detonation structure looks like a weak detonation, and not a strong detonation. Note that the argument against a weak detonation structure is absent in a PBS, since the pre-calculated shock motion provides the sequenced burn-times for the cells that trigger the change in the equation of state in the vicinity of the shock. An alternative interpretation is that the PB scheme is a capturing scheme which intends to capture states that are near or at the steady state equilibrium CJ-values and hence cut off, or do not represent in any way, a physical reaction zone structure from the inert unreacted shock state (the von Neumann spike) to the sonic point that normally would be computed as part of the reaction zone in a DNS.

4.1 Example: Ideal EOS

To illustrate the traditional implementation of program burn, we will start with an equation of state (EOS) for the detonation products, $e(p, v)$ and be even more specific by using the gamma law equation of state

$$e(p, v) = \frac{1}{\gamma - 1} \frac{p}{\rho}.$$

4.1.1 CJ states

To compute the CJ-states, we first assume that the unburnt upstream state (with the strong shock approximation) ahead of the wave is given by

$$\rho = \rho_o, \quad u = 0, \quad p = 0, \quad e = e_o, \quad (21)$$

with e_o unspecified at this point but will be chosen in the course of the analysis. Let $[\] = (\)_o - (\)_b$ denote the jump in a quantity across the interface from the o -state ($Y = 0$) to the completely burnt state ($Y = 1$) denoted by a b -subscript. The normal jump conditions across the interface moving with speed D_n are given by

$$[\rho(u_n - D_n)] = 0, \quad (22)$$

$$[\rho u_n(u_n - D_n) + p] = 0, \quad (23)$$

$$[E(u_n - D_n) + u_n p] = 0, \quad (24)$$

where E is the total energy defined earlier. With the assumption of the ideal EOS in the burnt products, the algebra of the above jump conditions are reduced to a quadratic equation in the normal particle velocity u_n , say. If we identify the speed D_n as the CJ value (D_{CJ}), the quadratic equation for u_n can be solved to give

$$u_n = \frac{D_{CJ} \pm \sqrt{D_{CJ}^2 - 2(\gamma^2 - 1)e_o}}{\gamma + 1}.$$

The CJ state is associated with the zero of the argument of the radical and lead to the identification of either the D_{CJ} in terms of the energy e_0 or vice a versa. Since we generally regard D_{CJ} as being given experimentally, we choose to write the condition as

$$e_0 = \frac{D_{CJ}^2}{2(\gamma^2 - 1)}. \quad (25)$$

Then the CJ states are

$$\rho_{CJ} = \rho_0 \left(\frac{\gamma + 1}{\gamma} \right), \quad p_{CJ} = \frac{\rho_0 D_{CJ}^2}{\gamma + 1}, \quad u_{CJ} = \frac{D_{CJ}}{\gamma + 1}. \quad (26)$$

It also follows (not demonstrated) that the CJ state is locally sonic. Note that in working out the Rankine-Hugoniot jump conditions across a program burn discontinuity, from the unreacted explosive to the burnt explosive where the burn-fraction Y is set equal to one, one obtains exactly the same Rankine-Hugoniot algebra as the reactive Euler equation where λ is set equal to one. Thus, the variation of a burn-fraction variable has no effect on the calculation of the CJ-states themselves.

As an example, we take the condensed phase explosive found in [1]. With $\gamma = 3$, $\rho_0 = 2$ and $D_{CJ} = 8$, we get for the CJ states

$$\rho_{CJ} = \frac{8}{3}, \quad p_{CJ} = 32, \quad u_{CJ} = 2. \quad (27)$$

4.1.2 Equation of state with modified pressure and effects on the structure

In keeping with the notion that one replaces p with p/Y in the burning cells with $0 < Y \leq 1$, the ideal EOS becomes

$$e = \frac{1}{\gamma - 1} \frac{p}{Y\rho}.$$

Again one assumes that in the fresh material one has the same initial specific internal energy e_0 , and the role of e_0 is the same as the heat of detonation.

To further analyze this structure let $U_n = u_n - D_n$ be the relative normal velocity in the shock-attached frame. For a quasi-steady traveling wave, the RH-relations hold throughout the structure, except now the internal energy has the dependence on the burn-fraction Y . As before, one can again solve the RH-relations

$$\begin{aligned} \rho U_n &= -\rho_0 D_n \\ \rho U_n^2 + p &= \rho_0 D_n^2 \\ e + \frac{1}{2} U_n^2 + \frac{p}{\rho} &= e_0 + \frac{1}{2} D_n^2, \quad \text{with } e = \frac{p}{\rho Y(\gamma - 1)}. \end{aligned}$$

for a quadratic equation in U_n with solutions

$$U_n = -\frac{[1 + (\gamma - 1)Y]D_n \pm \sqrt{D_n^2 - 2(\gamma - 1)Y[2 + (\gamma - 1)Y]e_0}}{2 + (\gamma - 1)Y}. \quad (28)$$

When $Y = 0$ the plus root corresponds to the unreacted flow state, and hence to the starting point for a weak detonation structure.

$$U_n = -D_n, \quad \text{or } u_n = 0.$$

The root associated with the minus sign is pathological and has $U_n = 0$ or $u_n = D_n$, and corresponds to a finite pressure but infinite density. In contrast, the standard strong shock state $U_n = -(\gamma - 1)/(\gamma + 1)D_n$ is achieved if the equation of state $e = pv/(\gamma - 1)$ is used instead of the modified equation of state $e = (p/Y)v/(\gamma - 1)$.

The issue is which state is selected, and we turn to the acoustic character of the distributed structure next. From the fundamental definition of the sound speed,

$$c^2 = \frac{p/\rho^2 - \partial e/\partial \rho}{\partial e/\partial p},$$

we have

$$c^2 = \frac{p}{\rho} [1 + (\gamma - 1)Y].$$

Next, if we use the energy equation $e + p/\rho + U_n^2/2 = D_n^2/2 + e_0$ and use the definitions of e and the last result for c^2 , we can eliminate p/ρ in favor of c and write an expression for the sonic parameter, η , as follows

$$\eta \equiv c^2 - U_n^2 = \left[e_0 + \frac{1}{2}(D_n^2 - U_n^2) \right] (\gamma - 1)Y - U_n^2.$$

If the detonation wave starts out on the weak branch, then at $Y = 0$, $c = 0$, and $U_n = -D_n$, the sonic parameter $\eta = -D_n^2 < 0$, and the wave is supersonic at the point of the lead disturbance. In fact one can compute the sonic locus in a (U_n^2, Y) - plane by setting $c^2 = U_n^2$ to obtain

$$U_n^2 = \frac{\gamma^2(\gamma - 1)e_0Y}{[1 + 1/2(\gamma - 1)Y]}. \quad (29)$$

The character of the structure of the (weak) detonation can be characterized by plotting its trajectory in a (U_n, Y) - plane. The weak CJ solution trajectory starts from the undisturbed state, $U_n = -D_n$ and terminates at the sonic state. Figure 7 shows this trajectory for the specific case of $D_n = D_{CJ}$. Note the square root behavior in U_n as $Y \rightarrow 1$, suggesting that the normal derivative has a square root singularity. This is due to the fact that the thermicity condition in the master equation does not vanish at the sonic point.

The other required ingredient for a weak detonation is a supersonic trigger. Ordinarily the supersonic trigger is regarded as a-physical. But for its application as a numerical algorithm, program burn assigns times at which the cell releases its energy. Specifically, the value of the burn-fraction is changed from $Y = 0$ to $Y = 1$ in proportion to how much of the particle cell has been crossed by an assumed shock wave. Therefore, the distribution of times when the cell is crossed by a shock is known a-priori, and is used to create the supersonic trigger. For steady, one-dimensional flow for a CJ detonation, the burn-times simply and exactly reflect the CJ detonation velocity.

We note that the state variables do depend on the burn-fraction if the burn-fraction were distributed in a discrete representation; i.e., not resolved to a Heaviside step function. Then the burn-fraction distribution on a finite mesh has the appearance of a pseudo-reaction zone structure. In the following discussion, for convenience, we will model this distribution not by a difference based scheme, but instead modeled by an "effective" rate law in the steady detonation frame.

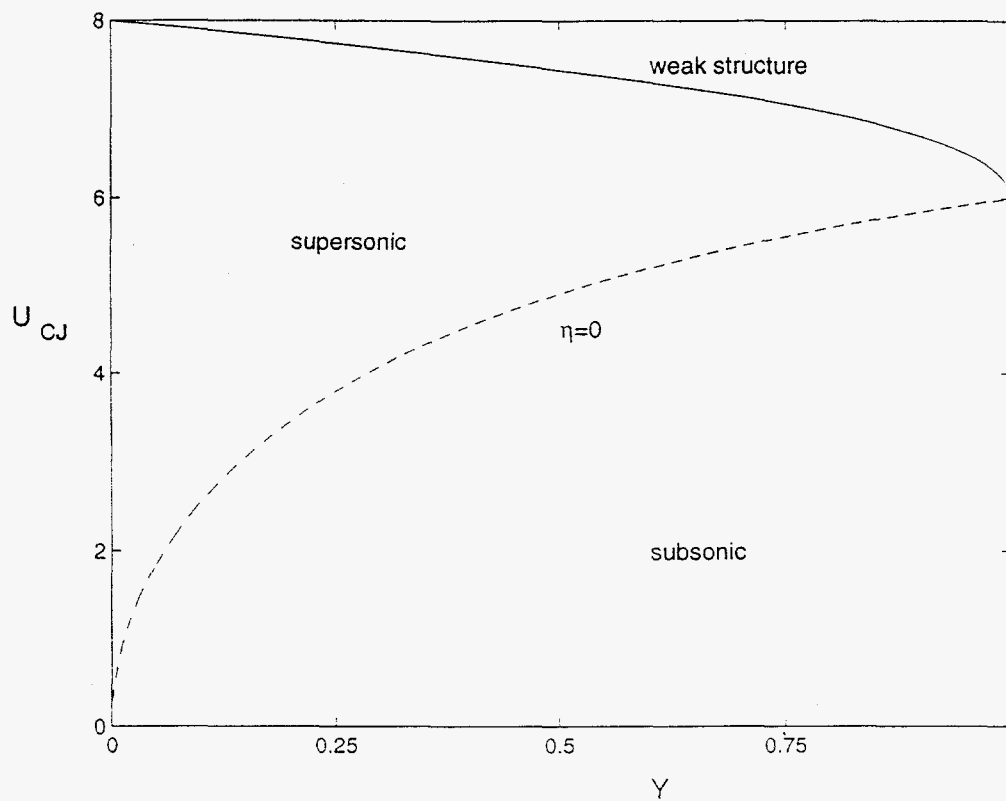


Figure 7: U_{CJ}, Y -plane showing the trajectory of the weak, CJ detonation. The dash curve corresponds to the sonic locus given by (29) and the solid curve corresponds to the weak structure given by (28).

$$U_n \frac{\partial Y}{\partial n} = R(Y), \quad (30)$$

where $R(Y)$ is an effective rate. In actual practice this rate is not given at all, rather the numerical scheme that defines the burn-fraction merely makes an assignment for the increase in Y such that it goes to $Y = 1$ when the detonation shock crosses the computational cell completely and $R(Y)$ is inferred from the details of that assignment. But certainly $R(Y)$ is both grid and algorithm-dependent.

Integration of (30), with the weak-structure relation between U_n and Y and the condition that $Y = 0$ at $x = 0$ (which is equivalent to the specification of the triggering event at the program burn-time), leads to a distribution function $Y(x)$ which has the basic profile shown in Figure 6.

An important observation is that the thickness of the heat-release zone in the program burn reaction zone will be a function of the grid thickness and can be computed asymptotically as $O(\Delta x)$, such that as $\Delta x \rightarrow 0$, the program burn reaction zone vanishes, as measured relative to *any* physical length scale. Thus the effect of the numerical algorithm that $R(Y)$ imitates is to approximate a delta function, centered at the burn-times and spaces on the grid as dictated by the burn table.

4.2 Numerical results of TPBPB and comparisons to DNS

We present some numerical results comparing the solutions obtained using the traditional pressure-based program burn model (TPBPB) to the solutions obtained from a DNS calculation. We use the condensed phase explosive described in [1]. The equations and numerical scheme for the DNS calculations were presented in detail in Section 2. For the TPBPB model we solve the corresponding non-reactive Euler equations with the EOS given by (20) and (25). Although current codes use a second-order scheme, we choose to use the same high-order scheme that is used for the DNS calculations to minimize errors resulting from different numerical algorithms, thus isolating any differences between the two solutions as arising from the various assumptions in the TPBPB model itself. To restate, we assume that the DNS calculations are "exact", and that any differences in solution structure will be due to the various approximations inherent in the TPBPB model. A mesh which has only one grid point in the reaction zone ($\Delta x = 2mm$) is used. The reason we choose this particular grid size is that typical implementation of the program burn methodology uses only a fine enough grid to resolve the hydrodynamics behind the wave front. The grid chosen here is thus typical of that used in engineering practice; no attempt is made here to optimize nor study the effect of grid spacing on the solution structures.

Figure 8 shows the structure from the DNS (solid) and from the TPBPB (circles) calculations for planar geometry. In each case, the solutions were stopped when the shock location reached $x_s(t) = 100mm$. The arrival times of the two calculations is seen to be approximately the same (for DNS, $t = 12.72\mu s$; for TPBPB, $t = 12.05\mu s$), the 5% relative difference being due to differences in the grid resolutions and to the modeling assumptions of the reaction zone by the TPBPB model. Note how well the program burn model captures the overall structure. The only differences are seen in the density plot, where the DNS calculates a weak density jump downstream of the lead shock while the TPBPB calculations (with the coarser grid) does not, and in the shock region where the DNS calculations show a strong detonation profile and the TPBPB calculations show a weak detonation profile. We also ran long-time solutions, until the shock was located at $x_s(t) = 900mm$ (Figure 9).

The arrival times of the two calculations have a relative difference of less than 1% (for DNS, $t = 112.77\mu s$; for TPBPB, $t = 111.97\mu s$). Again, note how well the program burn model captures the overall structure.

The major weakness of the TPBPB model, however, occurs when curvature is present. Figure 10 shows the structure from the DNS and from the TPBPB calculations for the case of cylindrical geometry. Since the TPBPB uses a Huygen's construction to propagate the shock, we see that the arrival time of the shock to the location $x_s = 100mm$ is much quicker ($t = 12.075\mu s$) than that of the DNS calculations ($t = 15.3\mu s$); this represents roughly a 21% error in the arrival times. This large difference is not due to grid resolution, but rather to the TPBPB modeling of the shock speed using a Huygen's construction. Since Huygen's construction over-estimates the speed of the shock when curvature is present, we also see noticeable differences in the solution structures downstream of the lead shock. As in the planar case, we also ran long-time solutions, until the shock was located at $x_s(t) = 900mm$ (Figure 11; a close up look at the structure is shown in Figure 12). The arrival times of the two calculations is seen to be converging (for DNS, $t = 127.5\mu s$; for TPBPB, $t = 112.0\mu s$), a relative difference of about 12%, which is still unacceptably too large for most engineering applications. In terms of the structure, the program burn model does seem to capture rather well the overall structure at the longer times. A closer look at the time behavior can be examined by comparing the shock speed and the star states to those obtained from DSD theory (see Figure 13). Note that the shock speed over-predicts the shock speed obtained from DSD, and that the star states are only asymptotic to the star states obtained from DSD theory.

The above results illustrates the strength and weaknesses of using the traditional program burn model to capture the physics of real detonations. For the planar case, the shock is propagated at the correct CJ speed, and the structure is represented well with only $1/40^{th}$ the number of grid points. This represents significant computational savings. However, when curvature is present there are major differences in not only the shock location but also in the structure of the solution. These differences are due to the fact that Huygen's construction over estimates the speed of the propagating shock. Since curvature is present in almost all engineering devices, it is essential to properly take into account effects due to curvature. It is this weakness that we address in the subsequent sections of this paper.

5 Modified pressure-based program burn model (MPBPB)

In the previous section we have seen that when curvature is present, the traditional pressure-based program burn model is deficient in that the use of Huygen's construction over estimates the speed of the detonation front, leading to significant differences of the shock location and the structure between the DNS and the TPBPB simulations. A simple modification can be made by extending the theory to include a shock speed which is curvature dependent, as is found in DSD theory. We modify the burn-times to include the curvature dependence, and we pre-compute the upstream internal energy as

$$e_0 = \frac{D_n^2}{2(\gamma^2 - 1)}, \quad (31)$$

where $D_n = D_n(\kappa)$ is the speed of the front with curvature dependence determined from DSD theory (see Section 3 for details). We refer to this modification of the burn-times and the upstream internal energy using DSD theory as the modified pressure-based program

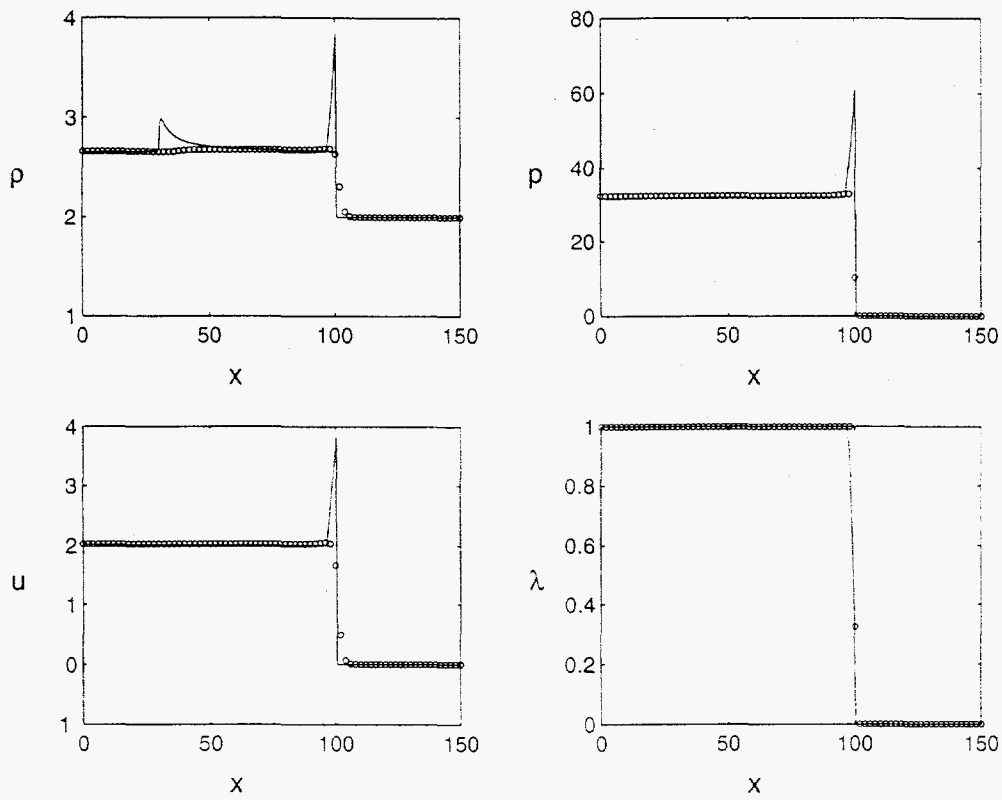


Figure 8: Plot of the structure for planar geometry at $x_s(t) = 100mm$. Circles correspond to the TPBPB model ($t = 12.05\mu s$), and the solid curve to DNS ($t = 12.72\mu s$).

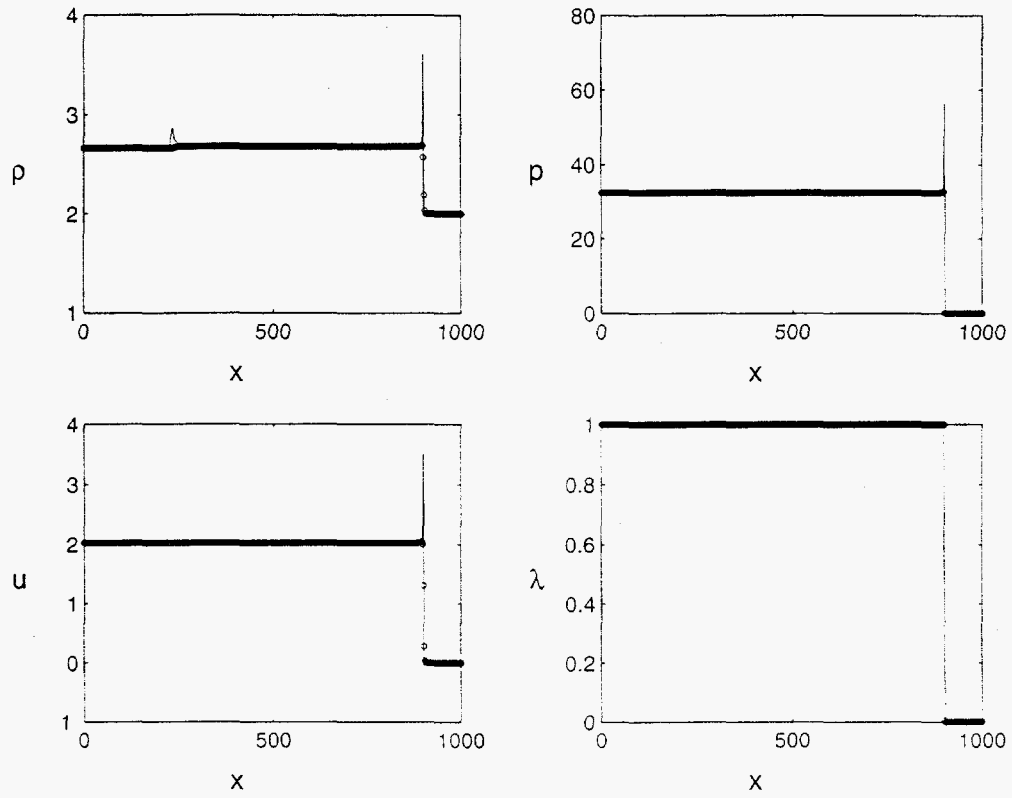


Figure 9: Plot of the structure for planar geometry at $x_s(t) = 900mm$. Circles correspond to the TPBPB model ($t = 111.97\mu s$), and the solid curve to DNS ($t = 112.77\mu s$).

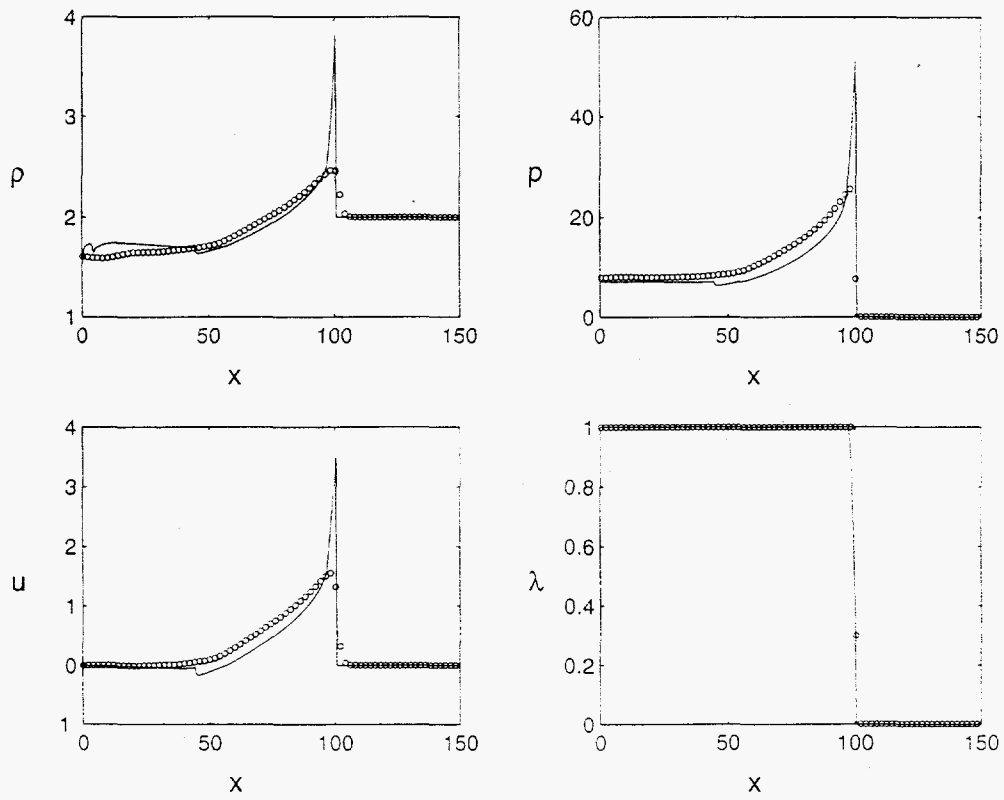


Figure 10: Plot of the structure for cylindrical geometry at $x_s(t) = 100\text{mm}$. Circles correspond to the TPBPB model ($t = 12.075\mu\text{s}$), and the solid curve to DNS ($t = 15.3\mu\text{s}$).

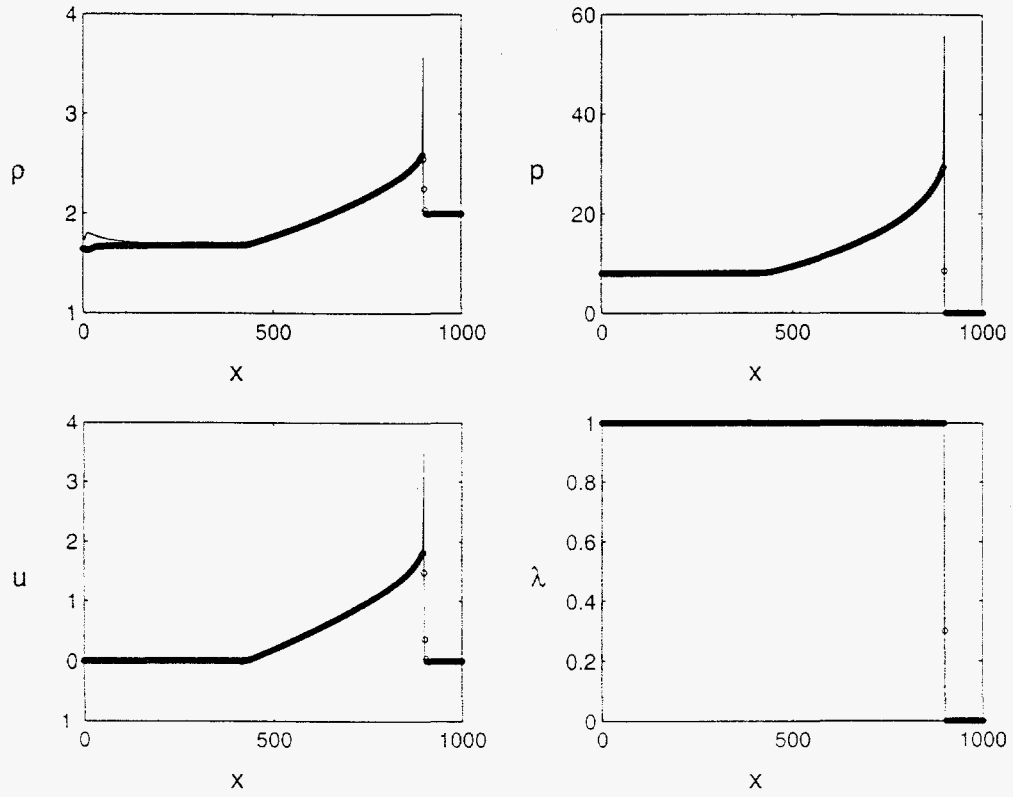


Figure 11: Plot of the structure for cylindrical geometry at $x_s(t) = 900mm$. Circles correspond to the TPBPB model ($t = 112.0\mu s$), and the solid curve to DNS ($t = 127.5\mu s$).

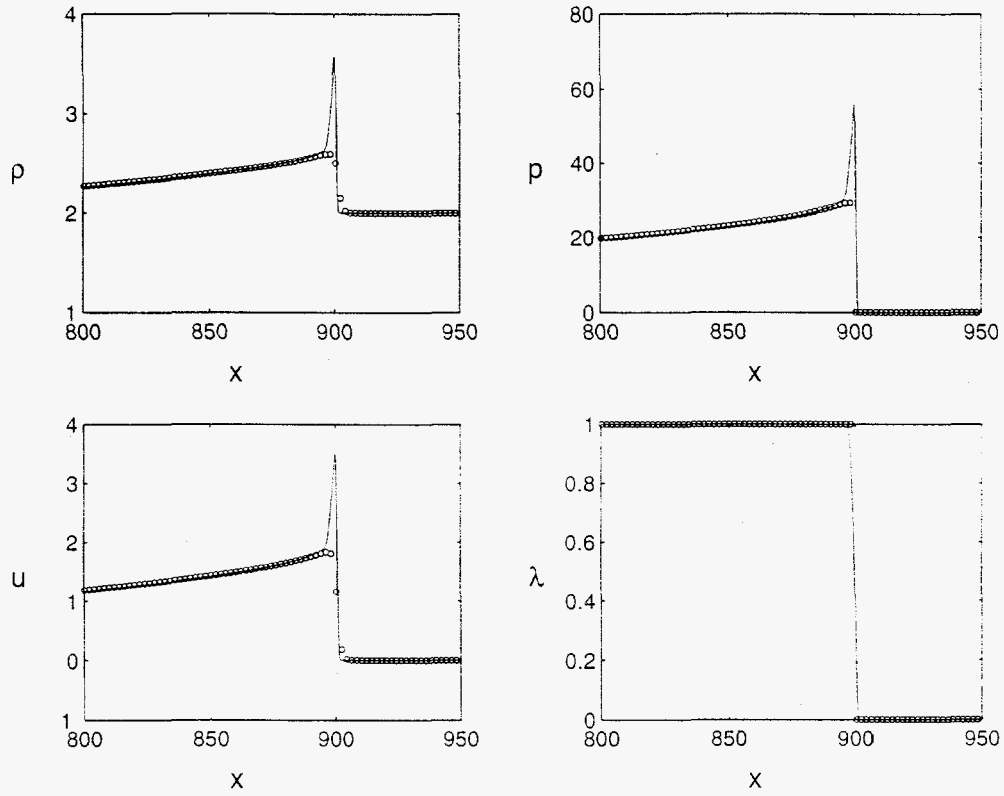


Figure 12: Blow-up of the shock structure shown in Figure 11 for cylindrical geometry at $x_s(t) = 900 \text{ mm}$. Circles correspond to the TPBPB model ($t = 112.0 \mu\text{s}$), and the solid curve to DNS ($t = 127.5 \mu\text{s}$).

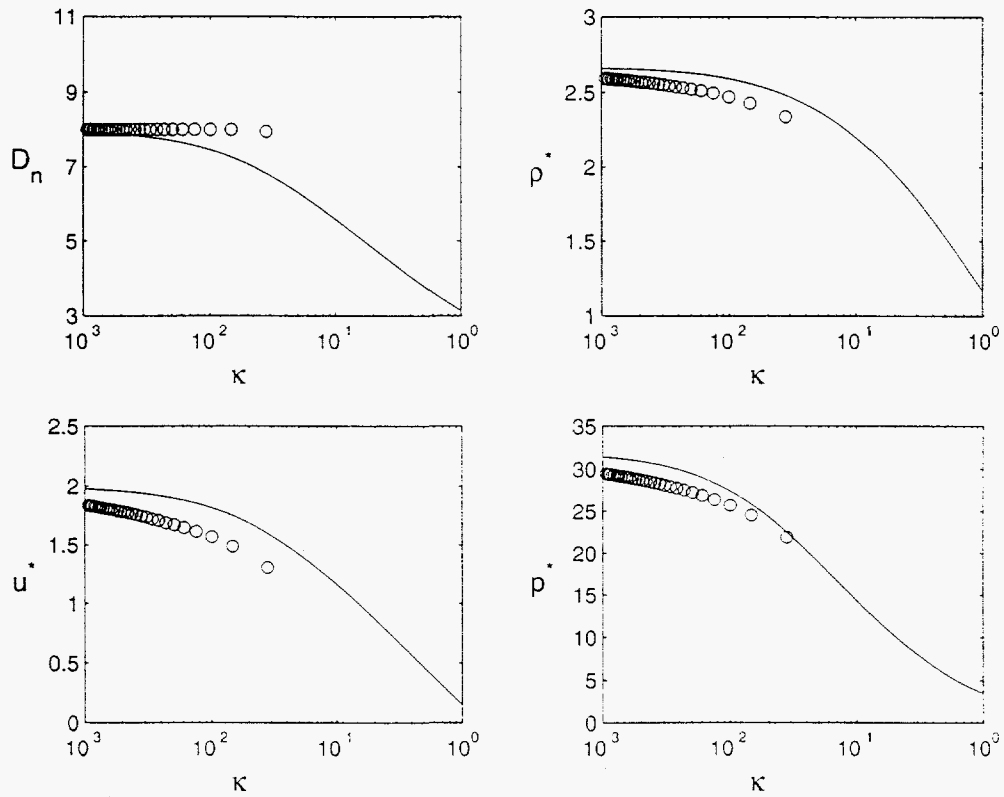


Figure 13: Plot of D_n and the star states as a function of κ for DSD (solid) and TPBPB (circles).

burn model, or MPBPB. Figure 14 shows results for the condensed phase explosive described in [1] in cylindrical geometry. Note that a simple change in the way the burn-times are computed and in the definition of the upstream internal energy can lead to significant changes in the errors. As before, the solutions were stopped when the shock location reached $x_s(t) = 100mm$. The arrival times of the two calculations is seen to be approximately the same (for DNS, $t = 15.3\mu s$; for MPBPB, $t = 14.2\mu s$), the 7% relative difference being a major improvement when compared to the 21% relative difference in the arrival times between TPBPB and DNS. Comparing Figures 10 and 14, we see that the MPBPB model captures the overall physics better than the TPBPB model. The long time solution, when $x_s(t) = 900mm$, is shown in Figures 15 and 16, and should be compared to Figures 11 and 12, respectively, from the TPBPB model.

However, the MPBPB model fails to capture the correct sonic (or star) states. Capturing the correct star states is an important indicator of how well a given scheme does since both the strong detonation and the weak detonation should terminate at this point. We plot in Figure 17 the star states as computed here from the numerical simulations to the star states determined from DSD theory. There is still an unacceptably large discrepancy in the star states. This shows that although the correct speed can be modeled using DSD theory, the overall structure is still not correct. This is a major deficiency of the model. We show in Figure 18 the star states for a grid resolution of $\Delta x = 0.5mm$. In these calculations the energy released is still over a single grid point, so reducing Δx reduces the effective reaction zone. Alternatively, one could keep the reaction zone fixed so that reducing Δx would imply more points in the reaction zone; we have not done this comparison but plan to do so in the future. Note that there is better agreement in the star states. Also note that the oscillations in the shock speed D_n observed in Figure 17 have been reduced by grid resolution. A further refinement would violate the spirit of the program burn model, and so no further grid refinements were carried out.

One final comment. Although the major TPBPB codes were modified to include curvature dependence using DSD theory (which we referred here as the MPBPB model) in the mid-1990's, no comparisons between TPBPB and MPBPB to DNS have been published. This paper constitutes the first discussion of these models.

6 DSD Program Burn Model

In the previous two sections we showed that when curvature is present and for small times, (i) the traditional pressure-based program burn (TPBPB) model fails to capture the correct shock speed and structure, and (ii) a simple modification of the burn-times using DSD theory (the MPBPB model) corrects the shock speed difficulty, but the solution structure remains incorrect in that the sonic states are not captured correctly. Since a simple grid resolution does not remedy these difficulties without grossly violating the spirit of a program burn model, one must turn to the basic model and include certain features that are absent in the previous two models. How this might be done is the subject of this section.

In this section we present the program burn equations that are used to model the reactive Euler equations described in Section 2. The Program Burn equations are the same as the inert Euler equations if the source terms are dropped. Then the equations of the Program Burn model are

$$\bar{U}_t + \bar{F}_x = \bar{G} + \bar{Q}R_{\delta(x-x_s(t))}. \quad (32)$$

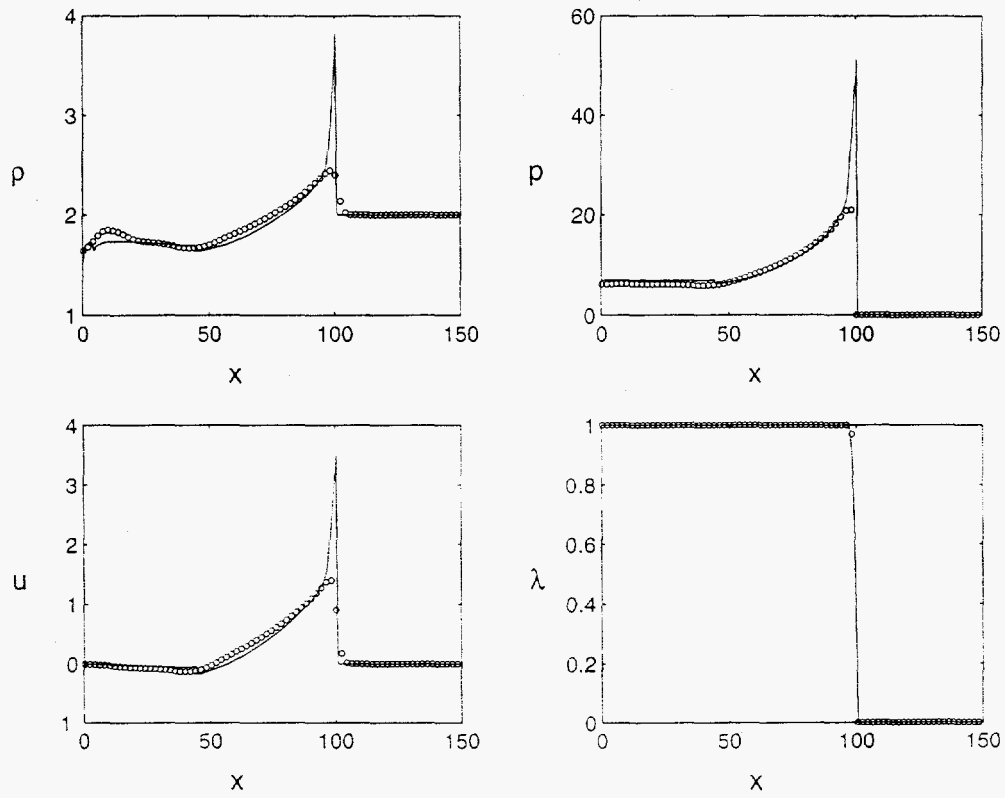


Figure 14: Plot of the structure for cylindrical geometry with $\Delta x = 2mm$ at $x_s(t) = 100mm$. Circles correspond to the MPBPB model ($t = 14.2\mu s$), and the solid curve to DNS ($t = 15.3\mu s$).

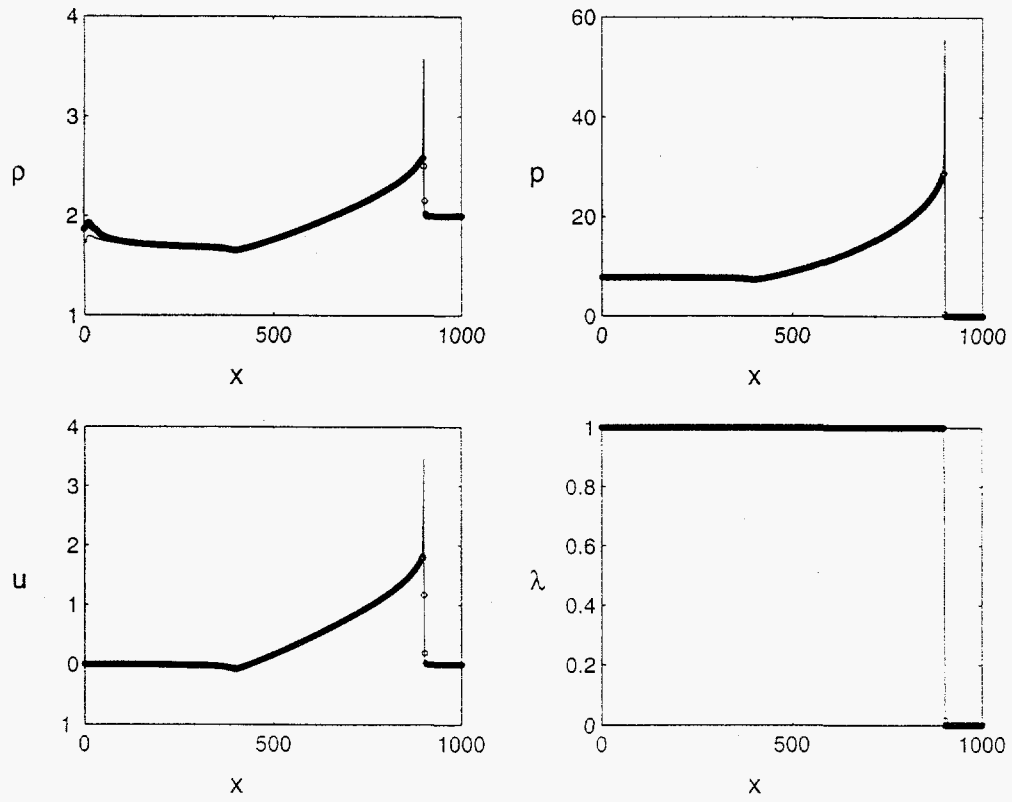


Figure 15: Plot of the structure for cylindrical geometry with $\Delta x = 2mm$ at $x_s(t) = 900mm$. Circles correspond to the MPBPB model ($t = 116.76\mu s$), and the solid curve to DNS ($t = 127.5\mu s$).

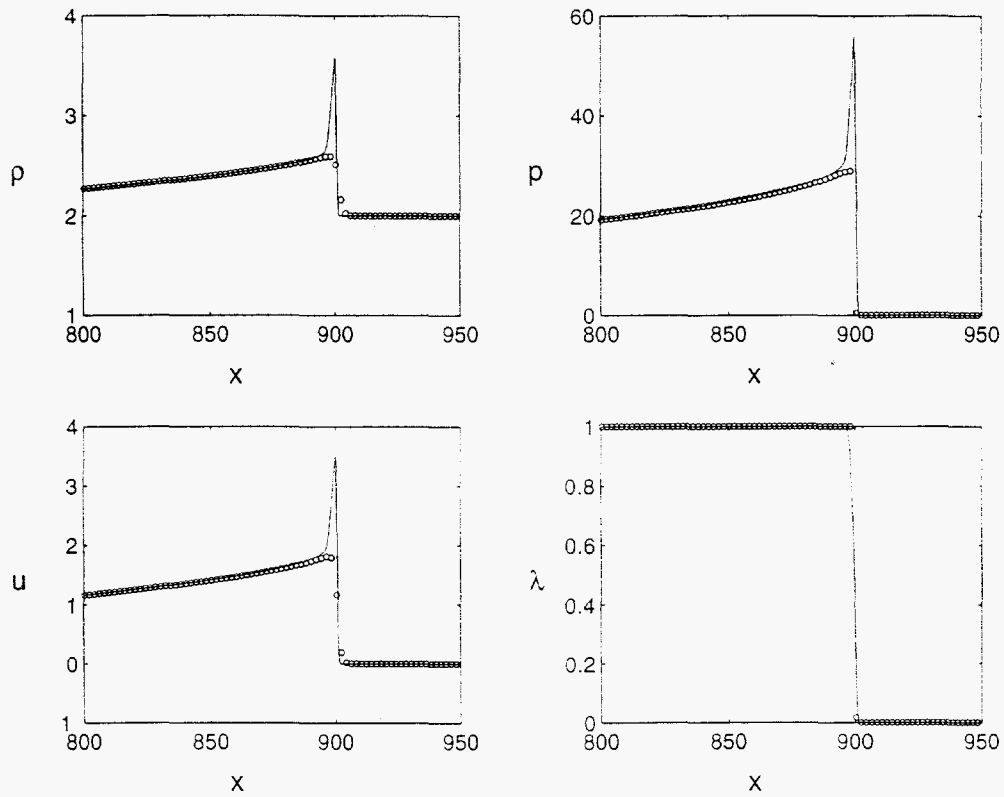


Figure 16: Blow-up of the shock structure shown in Figure 15 for cylindrical geometry with $\Delta x = 2\text{mm}$ at $x_s(t) = 900\text{mm}$. Circles correspond to the MPBPB model ($t = 116.76\mu\text{s}$), and the solid curve to DNS ($t = 127.5\mu\text{s}$).

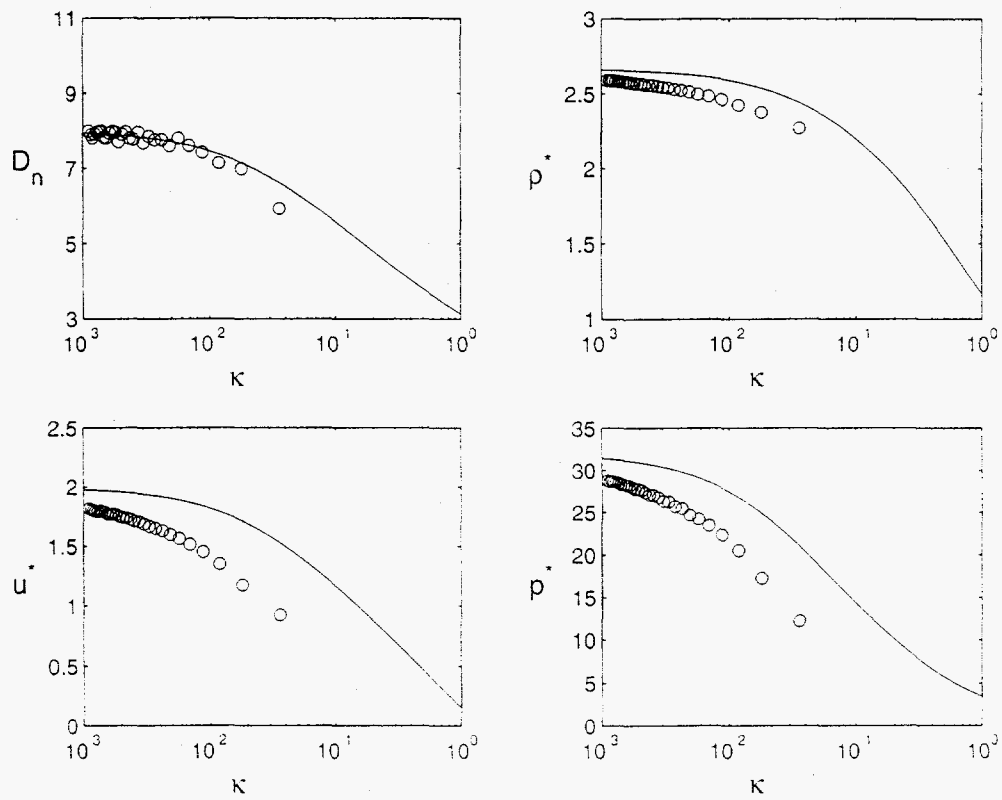


Figure 17: Plot of D_n and the star states as a function of κ for DSD (solid) and MPBPB (circles) with $\Delta x = 2mm$.

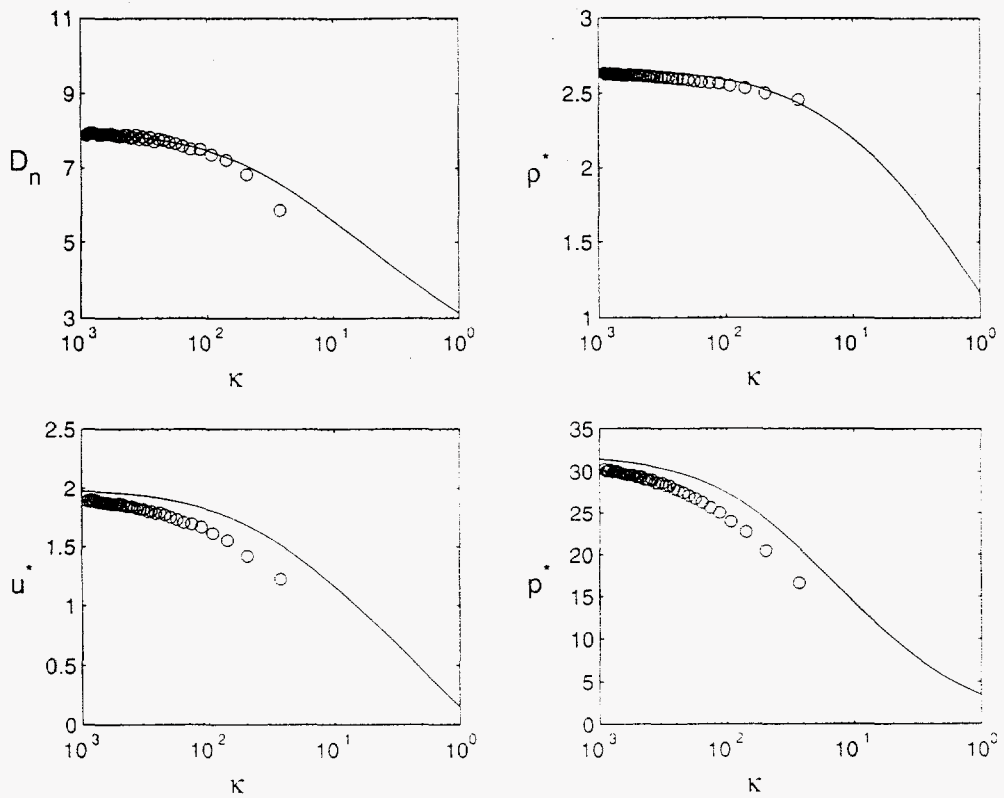


Figure 18: Plot of D_n and the star states as a function of κ for DSD (solid) and MPBPB (circles) with $\Delta x = 0.5mm$.

where

$$\vec{U} = [\rho, \rho u, E]^T, \quad (33)$$

$$\vec{F} = [\rho u, \rho u^2 + p, u(E + p)]^T. \quad (34)$$

$$\vec{G} = -\frac{j}{x} [\rho u, \rho u^2, (E + p)u]^T, \quad \vec{Q} = [Q_1, Q_2, Q_3]^T. \quad (35)$$

and again E is the total energy defined by

$$E = \rho \left(e + \frac{1}{2} u^2 \right), \quad (36)$$

and e is the internal energy. Specific choices for e will be given below. Note that the PB equations are a *three* equation system instead of a *four* equation system. Also note that $R_{\delta(x-x_s(t))}$ is a delta function centered on the program shock locus $x = x_s(t)$ and is assumed to be known. The geometric source term \vec{G} is identical in its first three components to its DNS counterpart. The choice of \vec{Q} and R_{δ} will be made later in the course of the analysis.

We will consider two models for the internal energy. The first model mimics the TPBPB model and is written, for an ideal gas, as

$$e = \frac{p/Y}{\rho(\gamma - 1)}. \quad (37)$$

where γ is the ratio of specific heats and Y is the burn fraction with $0 \leq Y \leq 1$. The upstream value of the internal energy is given by (31). We will refer to this as Model I. The second model is the standard equation of state without the burn fraction, and for an ideal gas is written as

$$e = \frac{p}{\rho(\gamma - 1)}. \quad (38)$$

The upstream value of the internal energy is given by $e_0 = 0$, consistent with the strong shock approximation. We will refer to this as Model II.

6.1 Specification of the source strengths \vec{Q}

To completely specify the program burn PDEs one must identify the source term strength \vec{Q} . Various specifications are made and analyzed below.

6.1.1 Quasi-steady assumption without explicit curvature

One way to determine values of \vec{Q} is to make the quasi-steady assumption and neglect the explicit dependence of curvature in the program burn equations. The lead shock is taken to be at $x = x_s(t)$ with speed D_n , which can depend on curvature and is given by the D_n, κ relation from DSD theory. Across the shock we allow for doses to the mass, momentum and energy, and the jump conditions across the program-burn shock are given by

$$[\rho(u_n - D_n)] = Q_1[Y], \quad (39)$$

$$[\rho u_n(u_n - D_n) + p] = Q_2[Y], \quad (40)$$

$$[E(u_n - D_n) + u_n p] = Q_3[Y], \quad (41)$$

where $[\phi] = \phi_o - \phi^*$, and the star states are determined using DSD theory. Note that in writing down these normal jump conditions we assumed that $R_\delta = dY/dn$, where n is the normal coordinate. Since the shock location is known, and both the upstream states and the star states are known, the jump relations become formulas for explicit evaluation of the doses \vec{Q} .

Evaluating the jumps leads to the following specifications for the components of \vec{Q}

$$\begin{aligned} Q_1 &= \rho_o D + \rho^*(u^* - D_n), \\ Q_2 &= p^* + \rho^* u^*(u^* - D_n), \\ Q_3 &= \rho_o e_o D_n + E^*(u^* - D_n) + u^* p^* . \end{aligned} \quad (42)$$

A plot of the star states was given previously in Figure 3 for the condensed phase explosive example given in [1]. The values of \vec{Q} as computed from these formulas is shown in Figure 19 for both Models I and II, respectively. Results for the two models are given in Figures 20–23 for Model I and in Figures 24–27 for Model II. Comparing these figures to those of the MPBPB shows that we have only slightly improved the errors in the structure and in the star states, but still too large for practical purposes. Also note that there are oscillations in the shock speed when using Model I (see Figure 22) but absent in Model II (compare with Figure 26). Before we complete this section, we address the following question and issue. If discrete approximations are taken for $R_{\delta(x-x_s(t))}$, then a discrete analysis is needed to show how the solution varies on the grid for the discrete analog of a quasi-steady traveling wave, and to ascertain if the desired end-states are captured by various algorithms. We do this in the appendix and show that there can indeed be oscillations in the solution structure when using Model I but absent when using Model II.

We now turn to other formulations or specifications of \vec{Q} below.

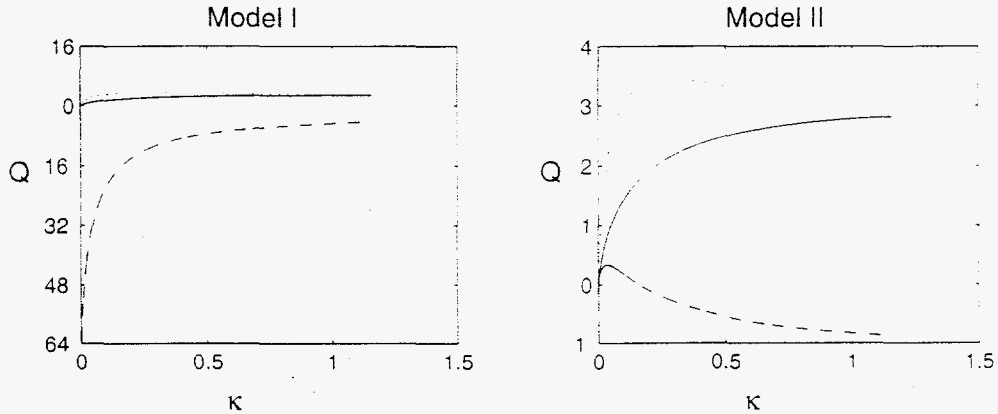


Figure 19: Plot of Q_1 (solid), Q_2 (dotted) and Q_3 (dashed) as a function of κ for Models I and II, respectively.

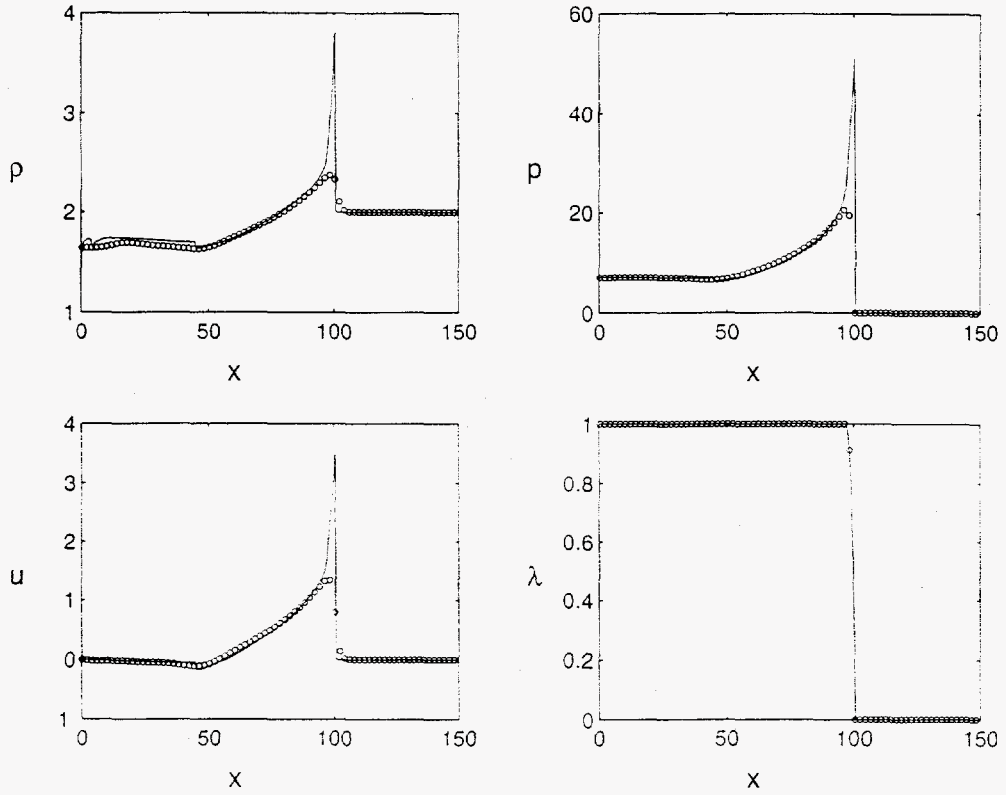


Figure 20: Plot of the structure for cylindrical geometry with $\Delta x = 2mm$ at $x_s(t) = 100mm$. Circles correspond to the DSDPB model ($t = 14.2\mu s$), and the solid curve to DNS ($t = 15.3\mu s$). Model I.

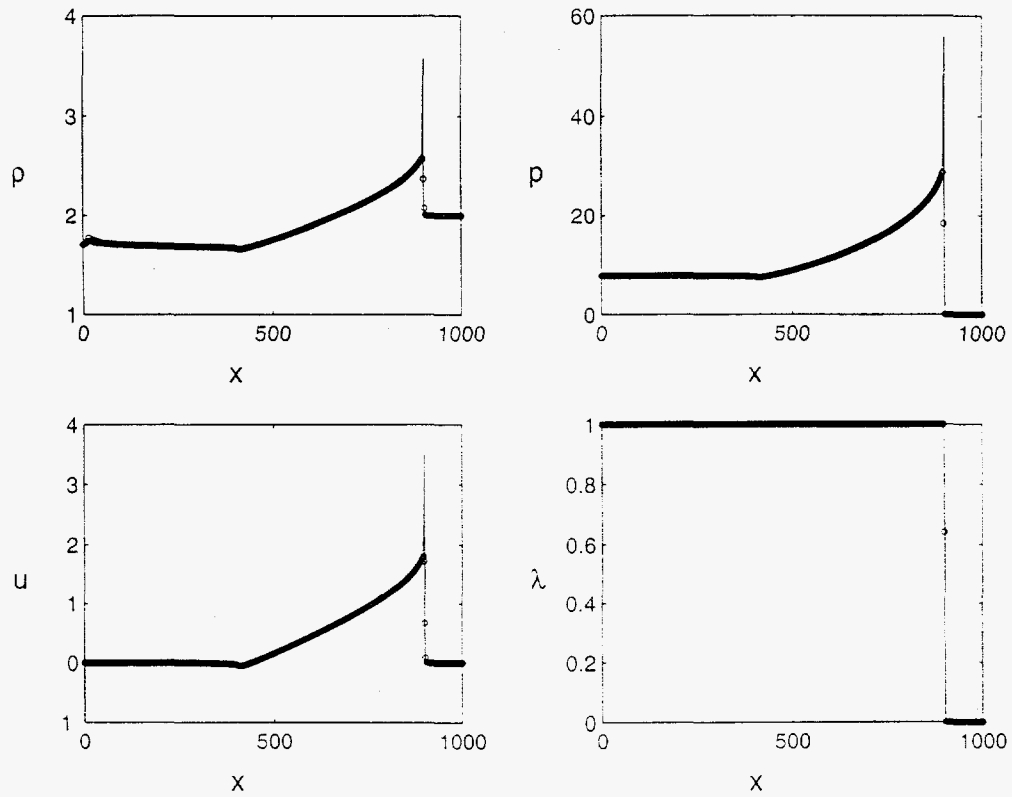


Figure 21: Plot of the structure for cylindrical geometry with $\Delta x = 2mm$ at $x_s(t) = 900mm$. Circles correspond to the DSDPB model ($t = 116.2\mu s$), and the solid curve to DNS ($t = 127.5\mu s$). Model I.

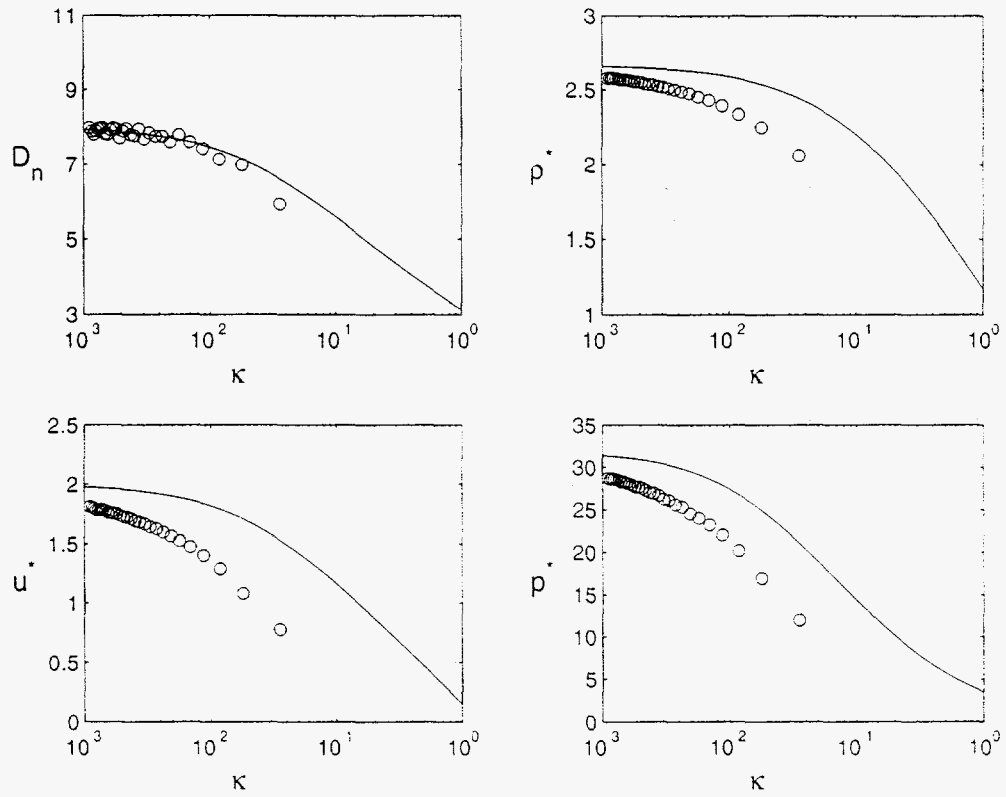


Figure 22: Plot of D_n and the star states as a function of κ for DSD (solid) and DSDPB (circles) with $\Delta x = 2mm$. Model I.

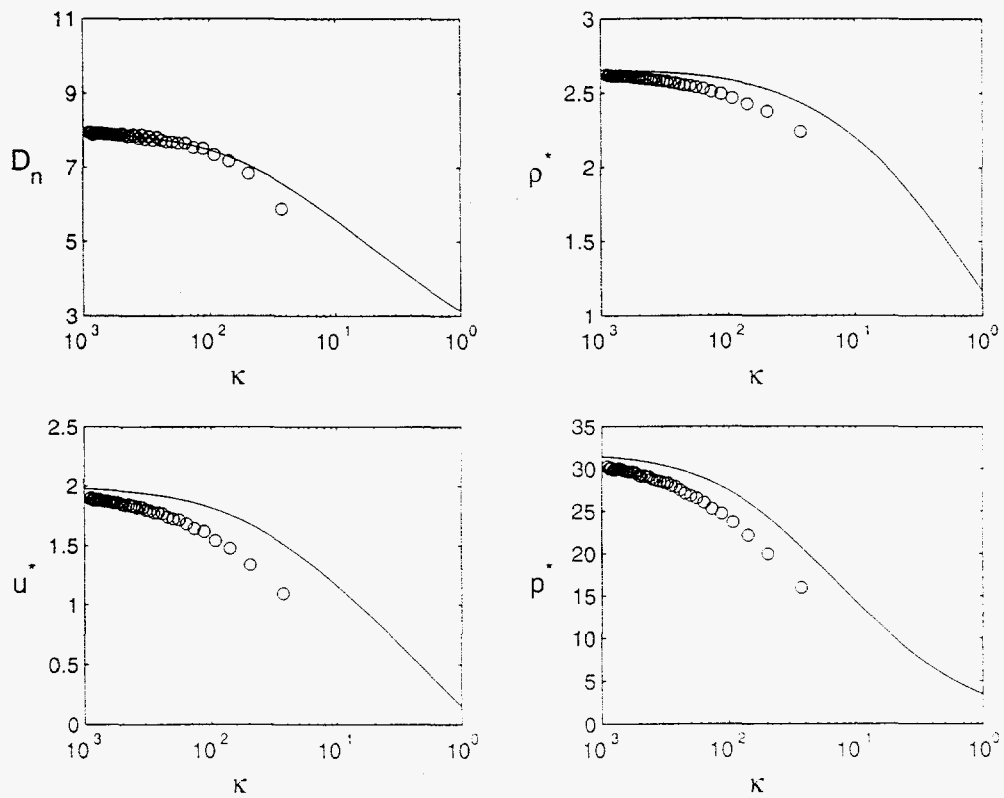


Figure 23: Plot of D_n and the star states as a function of κ for DSD (solid) and DSDPB (circles) with $\Delta x = 0.5mm$. Model I.

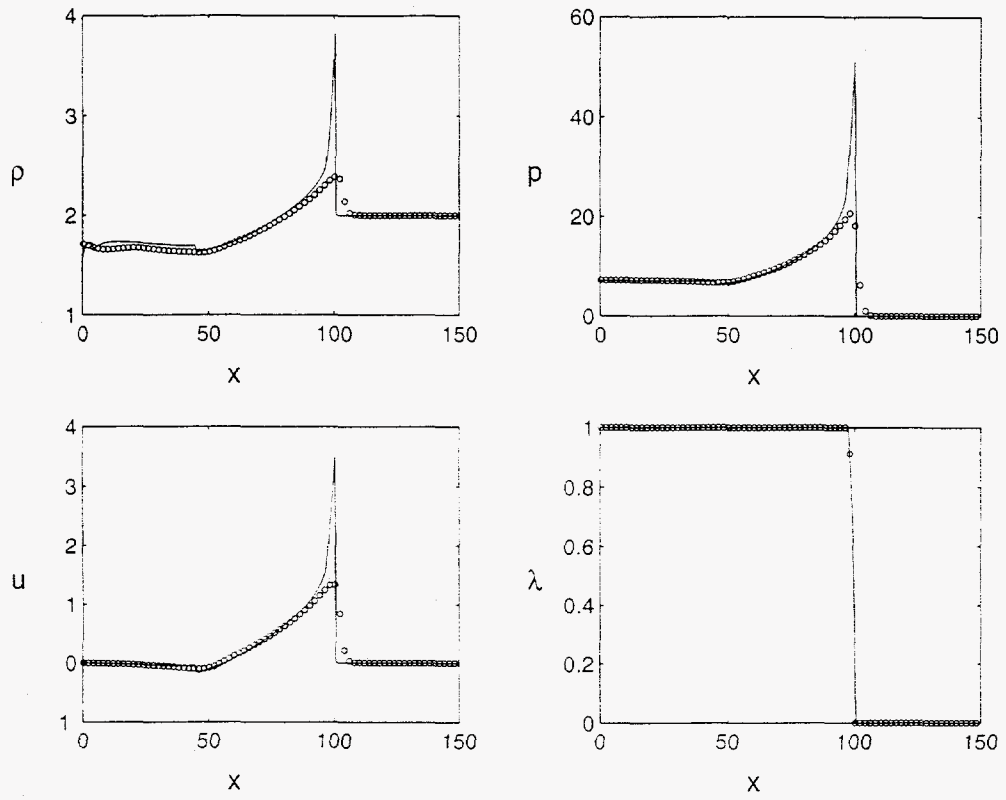


Figure 24: Plot of the structure for cylindrical geometry with $\Delta x = 2mm$ at $x_s(t) = 100mm$. Circles correspond to the DSDPB model ($t = 14.2\mu s$), and the solid curve to DNS ($t = 15.3\mu s$). Model II.

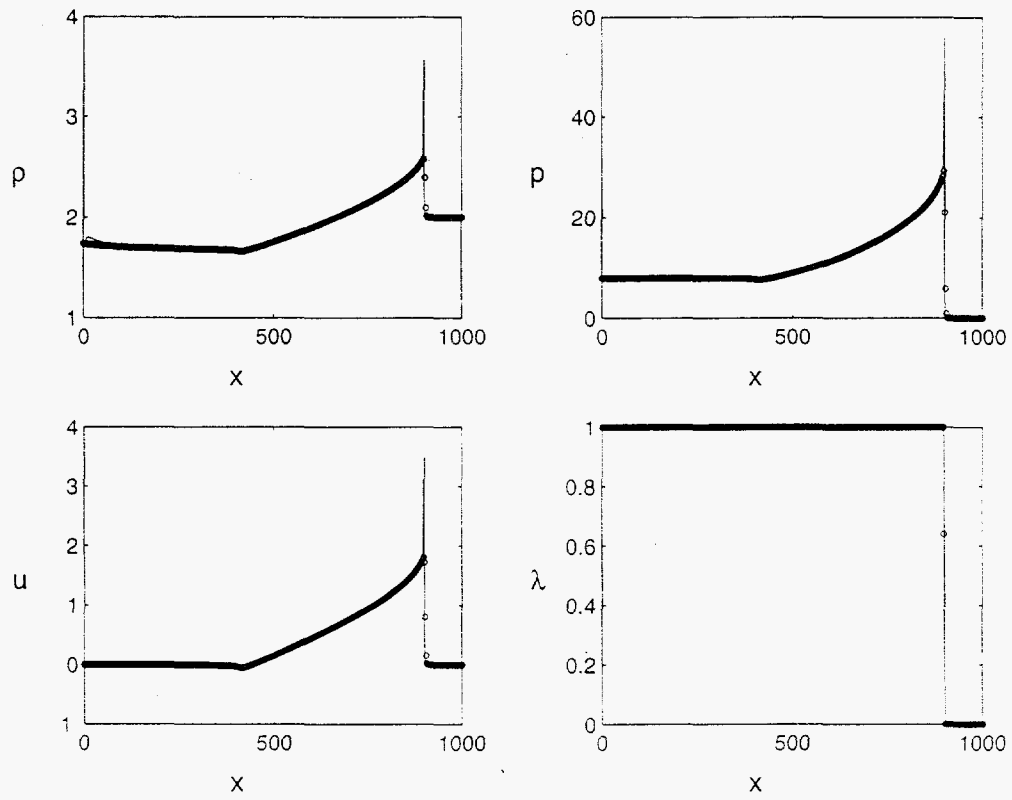


Figure 25: Plot of the structure for cylindrical geometry with $\Delta x = 2mm$ at $x_s(t) = 900mm$. Circles correspond to the DSDPB model ($t = 116.2\mu s$), and the solid curve to DNS ($t = 127.5\mu s$). Model II.

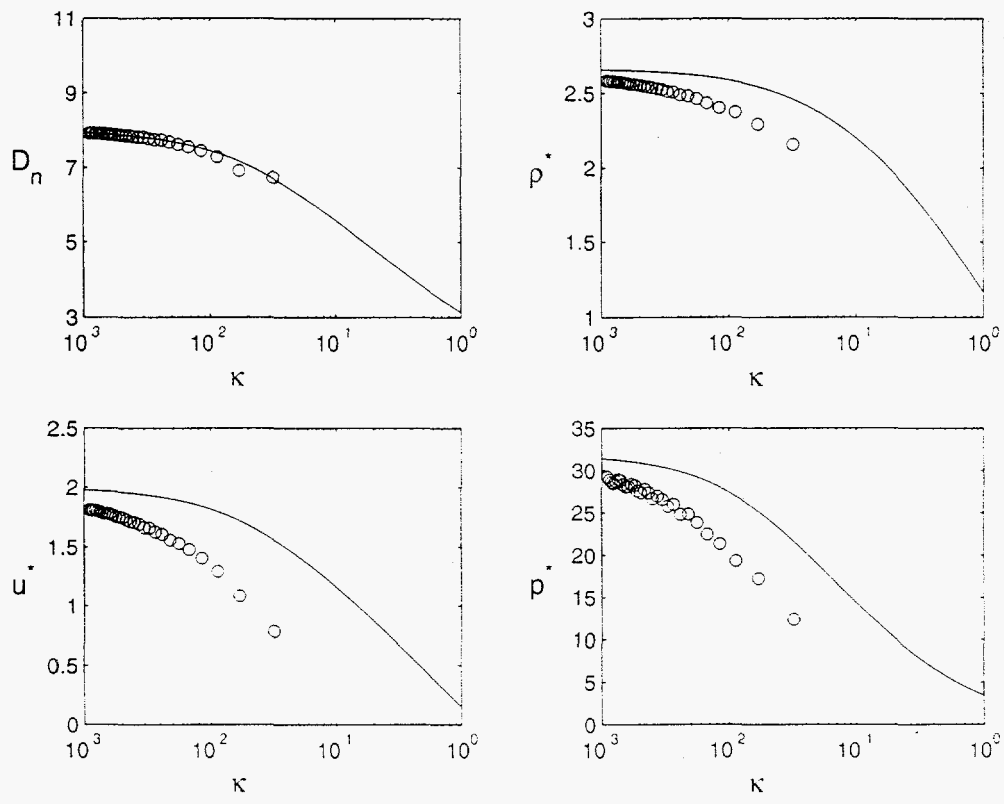


Figure 26: Plot of D_n and the star states as a function of κ for DSD (solid) and DSDPB (circles) with $\Delta x = 2mm$. Model II.

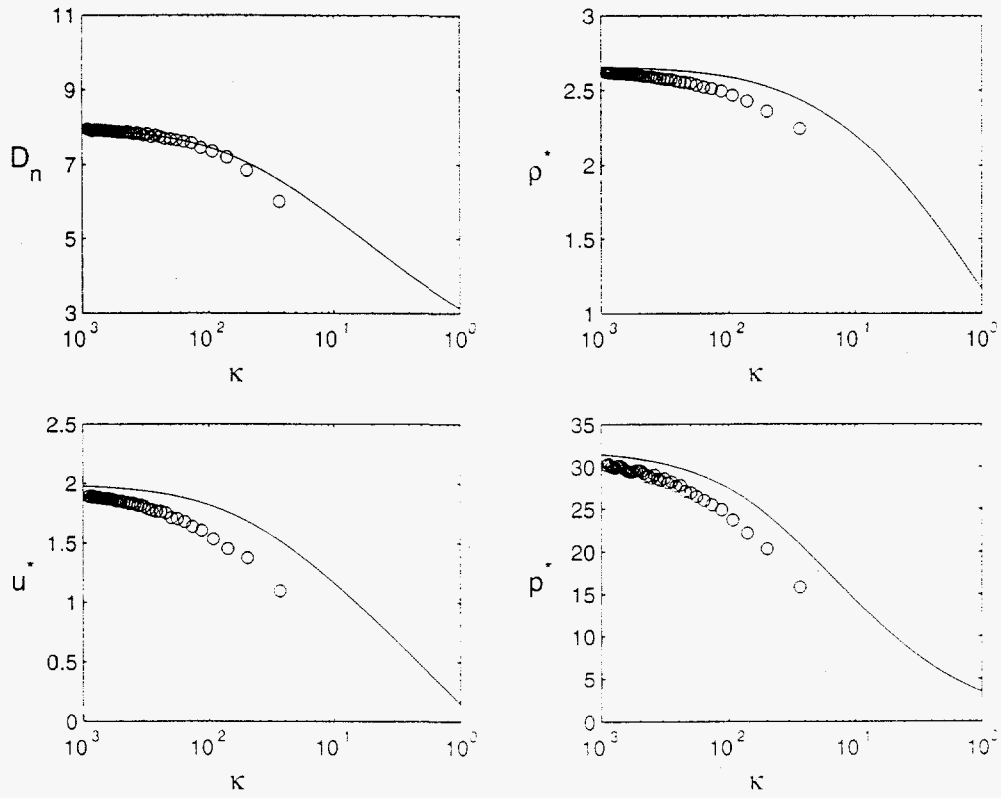


Figure 27: Plot of D_n and the star states as a function of κ for DSD (solid) and DSDPB (circles) with $\Delta x = 0.5mm$. Model II.

6.1.2 Quasi-steady assumption with explicit curvature

Let us again assume that the structure is quasi-steady, but now keep the explicit dependence of curvature in the equations. In this case normal jump relations can not be written down, and we turn to the DSD framework to determine \bar{Q} . For Model I where the EOS is given by (37), the quasi-steady analysis with curvature leads to the following system of equations

$$\frac{du}{dY} = \frac{\Phi}{\eta^2}, \quad (43)$$

$$\frac{dp}{dY} = (Q_2 - uQ_1)R_\delta \left(\frac{dY}{dn}\right)^{-1} - \rho(u - D_n) \frac{du}{dY}, \quad (44)$$

$$\frac{d\rho}{dY} = \frac{-\kappa\rho u + Q_1 R_\delta \left(\frac{dY}{dn}\right)^{-1}}{u - D_n} - \frac{\rho}{u - D_n} \frac{du}{dn}, \quad (45)$$

where

$$\eta^2 = c^2 - (u - D)^2, \quad c^2 = \frac{p}{\rho} [1 + (\gamma - 1)Y], \quad (46)$$

is the local sonic condition, and

$$\begin{aligned} \Phi = & -c^2 \kappa u \left(\frac{dY}{dn}\right)^{-1} + (u - D_n) \frac{p}{\rho Y} + \frac{1}{\rho} \left[(\gamma - 1) Q_3 - \right. \\ & \left. - (u - D_n + (\gamma - 1)Y u) Q_2 + u Q_1 \left(u - D_n + \frac{1}{2}(\gamma - 1)Y u \right) \right] R_\delta \left(\frac{dY}{dn}\right)^{-1} \end{aligned} \quad (47)$$

is the thermicity condition. These equations assume that the burn fraction Y is continuously differentiable in the effective reaction zone. In writing down these equations we have switched to a coordinate system where the burn fraction Y is the independent variable.

For Model II where the EOS is given by (38), the quasi-steady analysis with curvature leads to the following system of equations

$$\frac{du}{dY} = \frac{\Phi}{\eta^2}, \quad (48)$$

$$\frac{dp}{dY} = (Q_2 - uQ_1)R_\delta \left(\frac{dY}{dn}\right)^{-1} - \rho(u - D_n) \frac{du}{dY}, \quad (49)$$

$$\frac{d\rho}{dY} = \frac{-\kappa\rho u + Q_1 R_\delta \left(\frac{dY}{dn}\right)^{-1}}{u - D_n} - \frac{\rho}{u - D_n} \frac{du}{dn}, \quad (50)$$

where

$$\eta^2 = c^2 - (u - D)^2, \quad c^2 = \gamma p / \rho, \quad (51)$$

is the local sonic condition, and

$$\begin{aligned} \Phi = & -c^2 \kappa u \left(\frac{dY}{dn}\right)^{-1} + \frac{1}{\rho} \left[(\gamma - 1) Q_3 - \right. \\ & \left. - (\gamma u - D_n) Q_2 + u Q_1 \left(\gamma u - D_n - \frac{\gamma - 1}{2} u \right) \right] R_\delta \left(\frac{dY}{dn}\right)^{-1} \end{aligned} \quad (52)$$

is the thermicity condition.

We plan to use both models above to determine \bar{Q} , and then use these values in our program burn code to determine if accounting for curvature helps to correctly capture the sonic states.

7 Conclusions

We have presented a comprehensive review of the traditional program burn algorithm, and have compared solutions to those of a direct numerical simulation. It was shown that if curvature is present, then the traditional program burn algorithm not only overpredicts the shock speed, but also incorrectly captures the downstream sonic states. A slight modification to the burn times, based on detonation shock dynamic theory, can correct the shock speed difficulty but the downstream sonic states are still not correct. We are currently exploring alternative algorithms which not only will have the correct shock speed, but will also capture the correct downstream sonic states.

A APPENDIX

In this appendix we present an analysis of discrete representations of steady traveling waves.

A.1 Numerical Scheme I

We start out by illustrating the ideas set forth in this section with a simple first order numerical scheme. First we discretize using a uniform mesh in x and take uniform steps in time

$$x = i\Delta x, \quad i = -N, \dots, 0, \dots, N, \quad t = n\Delta t, \quad n = 0, \dots \quad (53)$$

We assume a first order finite difference scheme for (32), using a forward difference scheme for time, and a backward difference scheme for the spatial derivative, resulting in

$$\bar{U}_i^{n+1} = \bar{U}_i^n - \frac{\Delta t}{\Delta x} [\bar{F}_i^n - \bar{F}_{i-1}^n] + \Delta t \bar{S}_i^n, \quad (54)$$

where $\bar{S} = \bar{Q}R_\delta$. The initial conditions correspond to the strong shock approximation, and are

$$\rho = \rho_o, \quad u = 0, \quad p = 0, \quad e = e_o, \quad Y = 0. \quad (55)$$

for the downstream state, and

$$\rho = \rho^*, \quad u = u^*, \quad p = p^*, \quad e = e^* \equiv \frac{p^*}{\rho^*(\gamma - 1)}, \quad Y = 1. \quad (56)$$

for the upstream state, with e_o given by either (37) of Model I or by (38) of Model II. The star states are computed from a $D_n - \kappa$ code with a particular value of κ , the curvature. See Figure 3 for a specific example.

A.1.1 Steady Wave Analysis

We impose a constraint that we seek a solution to the difference relations that corresponds to a steady traveling wave at speed D_n . Thus,

$$\bar{U}_t + D_n \bar{U}_x = 0 \quad (57)$$

which has as the solution $\bar{U} = \bar{U}(x - D_n t)$. We now discretize using the first-order scheme given above, and write

$$\bar{U}_i^{n+1} = \bar{U}_i^n - \frac{\Delta t}{\Delta x} D_n [\bar{U}_i^n - \bar{U}_{i-1}^n] \quad (58)$$

We now choose to require (54) and (58) to hold simultaneously, as a requirement that the discrete wave be steady. Eliminating \bar{U}_i^{n+1} results in

$$-D_n(\bar{U}_i^n - \bar{U}_{i-1}^n) + (\bar{F}_i^n - \bar{F}_{i-1}^n) = \Delta x \bar{S}_i^n \quad (59)$$

This last equation is a nonlinear recurrence relation relating \bar{U}_i^n and \bar{U}_{i-1}^n . Thus, given state i , we can find state $i - 1$.

A.1.2 End State Analysis

We determine the admissible end states that are consistent with the traveling wave recursion. Assume that the wave structure is centered on the grid from $i = -N$ to $i = 0$. We now multiply (59) by Δx and then sum over the wave structure. Many of the sums that appear can be simplified by appealing to the following summation rules

$$\sum_{i=-N}^{i=0} (\phi_{i+1} - \phi_{i-1}) = \phi_0 + \phi_1 - \phi_{-N} - \phi_{-(N+1)} \quad (60)$$

$$\sum_{i=-N}^{i=0} (\phi_{i+1} - 2\phi_i + \phi_{i-1}) = -\phi_0 + \phi_1 - \phi_{-N} + \phi_{-(N+1)} \quad (61)$$

$$\sum_{i=-N}^{i=0} (\phi_{i+1} - \phi_i) = \phi_1 - \phi_{-N} \quad (62)$$

$$\sum_{i=-N}^{i=0} (\phi_i - \phi_{i-1}) = \phi_0 - \phi_{-(N+1)} \quad (63)$$

for any function ϕ . Applying these summation rules results in

$$-D_n (\vec{U}_0^n - \vec{U}_{-(N+1)}^n) + (\vec{F}_0^n - \vec{F}_{-(N+1)}^n) = \sum_{i=-N}^{i=0} \Delta x \vec{S}_i^n \quad (64)$$

Now suppose that the summation is over a large enough domain so that

$$\vec{U}_{-(N+1)}^n \approx \vec{U}^* \quad (65)$$

where the star states have been determined according to a $D_n - \kappa$ code. Furthermore, let us assume that

$$\vec{S}_i^n = \vec{Q} \frac{Y_i - Y_{i-1}}{\Delta x} \quad (66)$$

where Y is the burn fraction and is zero ahead of the shock and goes to one in the completely reacted state behind the shock. We do not put any restrictions on the form of Y other than it be continuous, so that

$$\begin{aligned} \sum_{i=-N}^{i=0} \Delta x \vec{S}_i^n &= \vec{Q} \sum_{i=-N}^{i=0} (Y_i - Y_{i-1}) \\ &= -\vec{Q} \end{aligned} \quad (67)$$

Solving for \vec{Q} yields

$$\vec{Q} = D_n (\vec{U}_0 - \vec{U}^*) - (\vec{F}_0 - \vec{F}^*) \quad (68)$$

which, in component form, is identical to (42).

A.1.3 Results

We wish to solve the nonlinear recursion formula(59), subject to the initial conditions (55). The goal is to obtain the correct end states (56).

For the numerical results to be presented below, we first define x_R to be the length of the reaction zone behind the shock. If $|x_R|$ is small, then we dump the heat into the system rather quickly; if $|x_R|$ is large, the opposite is true. We also define the parameter

$$k = \frac{x_R}{\Delta x}, \quad (69)$$

which gives the number of grid points in the reaction zone. We shall use k as the parameter that we vary instead of Δx ; once k is chosen the corresponding Δx can easily be found from the above relation. In all calculations, we take $\Delta t = \Delta x/2$, which satisfies the CFL condition. The parameters which characterizes the solutions are k , μ , and x_R . All calculations are stopped when the numerical solution has reached a constant state. At this location we define the relative error as

$$R_E = \left\{ \left(\frac{\rho^* - \rho^*(num)}{\rho^*} \right)^2 + \left(\frac{u^* - u^*(num)}{u^*} \right)^2 + \left(\frac{p^* - p^*(num)}{p^*} \right)^2 \right\}^{1/2} \quad (70)$$

where the star states are those from a $D_n - \kappa$ code, and *num* means the star states obtained numerically from the recursion formula.

For the prescription of the burn fraction, we update Y according to the rule

$$Y = \begin{cases} 0 & x_i < x_s, \\ \frac{x_s - x_i}{k\Delta x} & x_s - k\Delta x < x_i < x_s, \\ 1 & x_i < x_s - k\Delta x. \end{cases} \quad (71)$$

Selected results are shown in Figure 28 and Tables (1) and (2) for Model I and Figure 29 and Table (3) for Model II. Note that in all cases the relative error is an order of magnitude less than one percent. One interesting observation: the results for Model II are somewhat more accurate than those of Model I. Although the codes are essentially identical, the root finding procedure finds the correct root for Model II, while it has trouble in Model I as κ increases. Indeed, the code did not converge for $\kappa = 1$ when using Model I, but converged to the exact root for Model II. This suggests that numerical schemes using Model I might also experience this problem when the curvature gets large.

$x_R(mm)$	$k = 1$	$k = 2$	$k = 5$	$k = 10$	$k = 20$
0.1	0.006	0.018	0.028	0.036	0.045
1.0	0.003	0.005	0.005	0.006	0.016
2.0	0.003	0.004	0.004	0.005	0.006
4.0	0.002	0.004	0.003	0.004	0.005

Table 1: Table of relative errors R_E in percent for $\mu = 0$ and $\kappa = 0$. Model I.

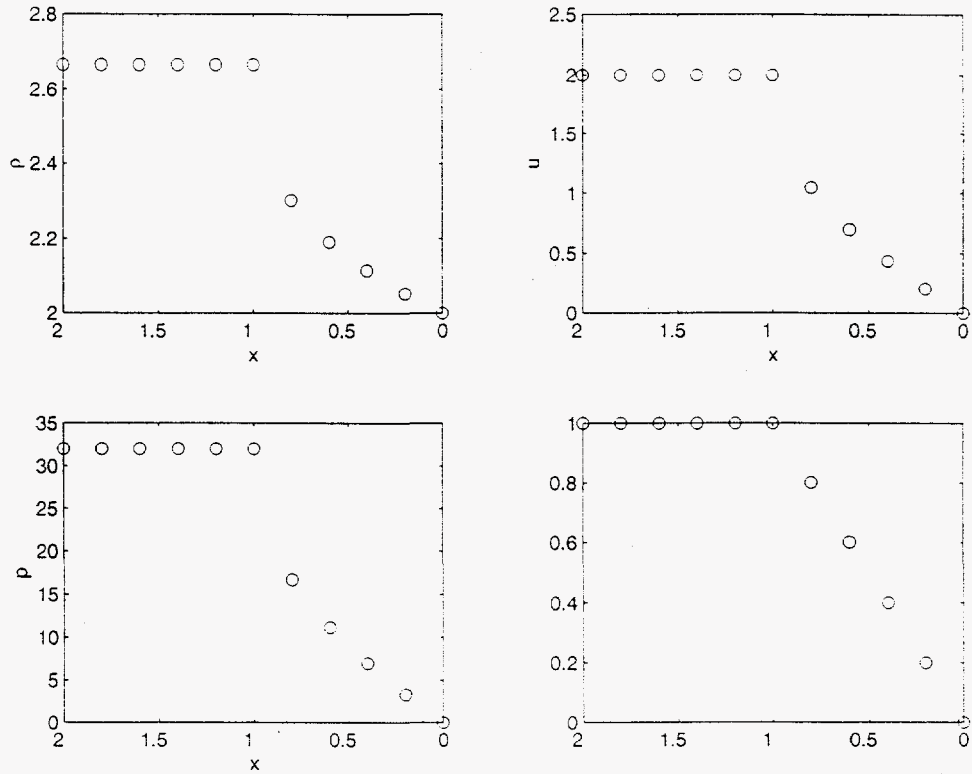


Figure 28: Plot of structure for $\kappa = 0$, $\mu = 0$, $x_R = 1mm$, and $k = 5$. Model I.

κ	$k = 1$	$k = 2$	$k = 5$	$k = 10$	$k = 20$
0.0	0.003	0.005	0.005	0.006	0.016
0.05	0.007	0.009	0.013	0.016	0.020
0.1	0.006	0.008	0.012	0.015	0.018

Table 2: Table of relative errors R_E in percent for $\mu = 0$ and $x_R = 1$. Model I.

κ	$k = 1$	$k = 2$	$k = 5$	$k = 10$	$k = 20$
0.0	0.002	0.002	0.002	0.002	0.003
0.05	0.002	0.002	0.003	0.002	0.004
0.1	0.003	0.005	0.003	0.005	0.005
1.0	0.0	0.0	0.0	0.0	0.0

Table 3: Table of relative errors R_E in percent for $\mu = 0$ and $x_R = 1$. Model II.

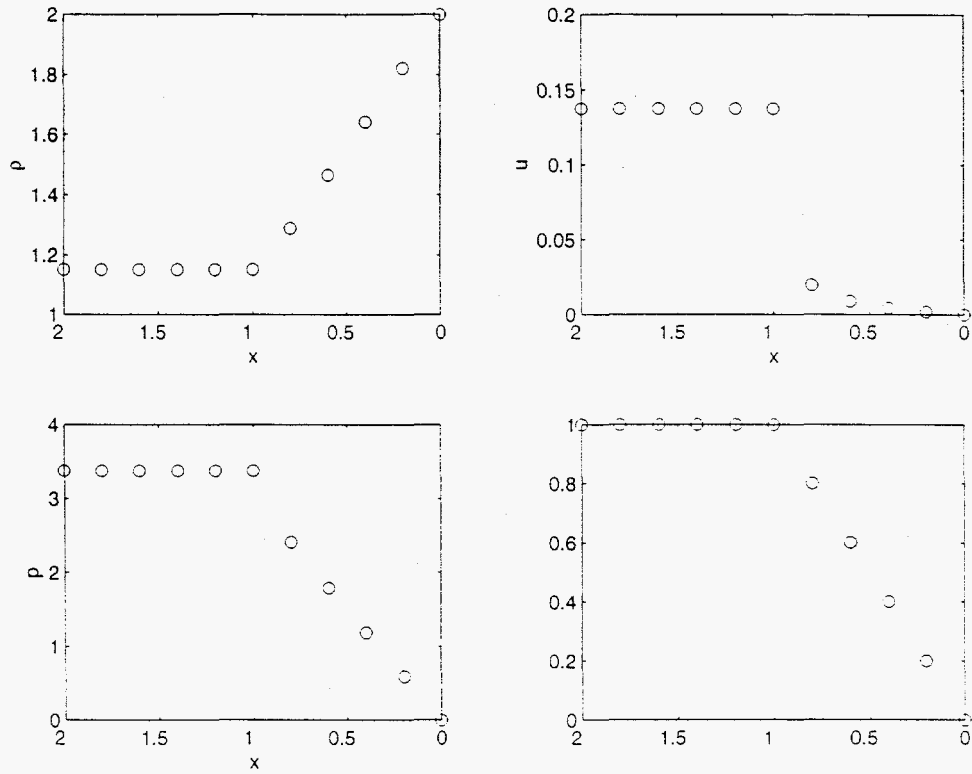


Figure 29: Plot of structure for $\kappa = 1$, $\mu = 0$, $x_R = 1mm$, and $k = 5$. Model II.

A.2 Numerical Scheme II

An alternative numerical scheme is presented in this section. Following the analysis of section 4, we discretize (32) using a forward difference scheme in time and a centered difference scheme in space. We next impose the constraint of a steady wave, resulting in the following nonlinear recurrence equation

$$-D_n(\bar{U}_{i+1}^n - \bar{U}_{i-1}^n) + (\bar{F}_{i+1}^n - \bar{F}_{i-1}^n) = 2\Delta x \bar{S}_i^n \quad (72)$$

Given states $i+1$ and i , we can find state $i-1$. An end state analysis gives (42). The results are essentially the same as the scheme above for κ close to zero so we do not present any results here. However, as κ increases from zero, the results for Model I gave solutions which had a limit cycle, one of them being near the correct end states. These oscillations can be removed with viscosity. As κ increased further, the results using Model I not only had a limit cycle, but also gave incorrect end states. This behavior was not observed when using Model II. Compare Figures 30 and 31 for example. The correct star states for $\kappa = 0.3284$ are $\rho^* = 1.7389$, $u^* = 0.6528$ and $p^* = 7.5214$. Again, this indicates that the numerical scheme, once chosen, is sensitive to the model used for the equation of state as κ increases from zero.

A.3 Numerical Scheme III

A third alternative numerical scheme is presented in this section. The numerical scheme is the MacCormack finite-difference method (1969), which is a predictor-corrector scheme of the Lax-Wendroff type. In the predictor step we use a first-order backward difference stencil for the flux, while in the corrector step we use a first-order forward difference stencil. Variations of this include using a forward stencil in the predictor step and a backward stencil in the corrector step, or alternating between the two: i.e., forward differences in the predictor and backward differences in the corrector on the first time step, and backward differences in the predictor and forward differences in the corrector on the second time step, etc. The mesh is the same given in the previous sections. The predictor step at the grid point i is given by

$$\bar{U}_i^{n+1/2} = \bar{U}_i^n - \frac{\Delta t}{\Delta x} [\bar{F}_i^n - \bar{F}_{i-1}^n] + \Delta t \bar{S}_i^n \quad (73)$$

and for the corrector step

$$\bar{U}_i^{n+1} = \frac{1}{2} \left\{ \bar{U}_i^{n+1/2} + \bar{U}_i^n - \frac{\Delta t}{\Delta x} [\bar{F}_{i+1}^{n+1/2} - \bar{F}_i^{n+1/2}] + \Delta t \bar{S}_i^{n+1/2} \right\} \quad (74)$$

where we use the notation

$$\bar{F}_i^n = \bar{F}(\bar{U}_i^n) \quad (75)$$

The initial conditions are given by (55) and the end states by (56).

A.3.1 Steady Wave Analysis

Following the analysis of the previous sections, we impose a constraint that we seek a solution to the difference relations that corresponds to a steady traveling wave at speed D_n . Thus,

$$\bar{U}_t + D_n \bar{U}_x = 0 \quad (76)$$

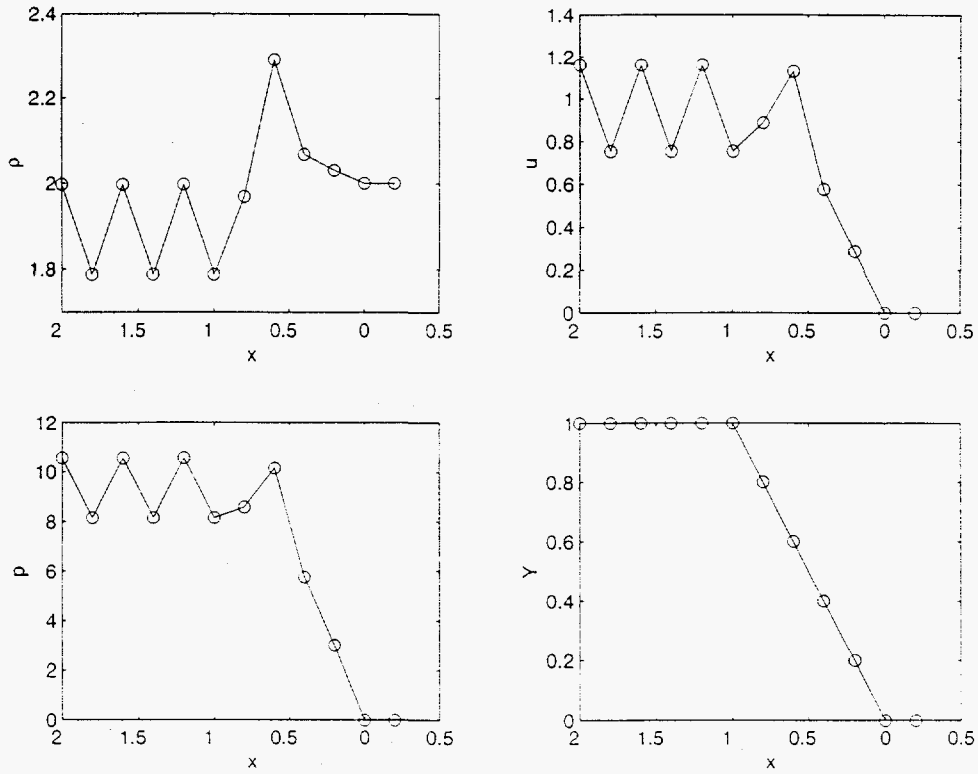


Figure 30: Plot of structure for $\kappa = 0.3284$, $\mu = 0$, $x_R = 1mm$, and $k = 5$. Model I.

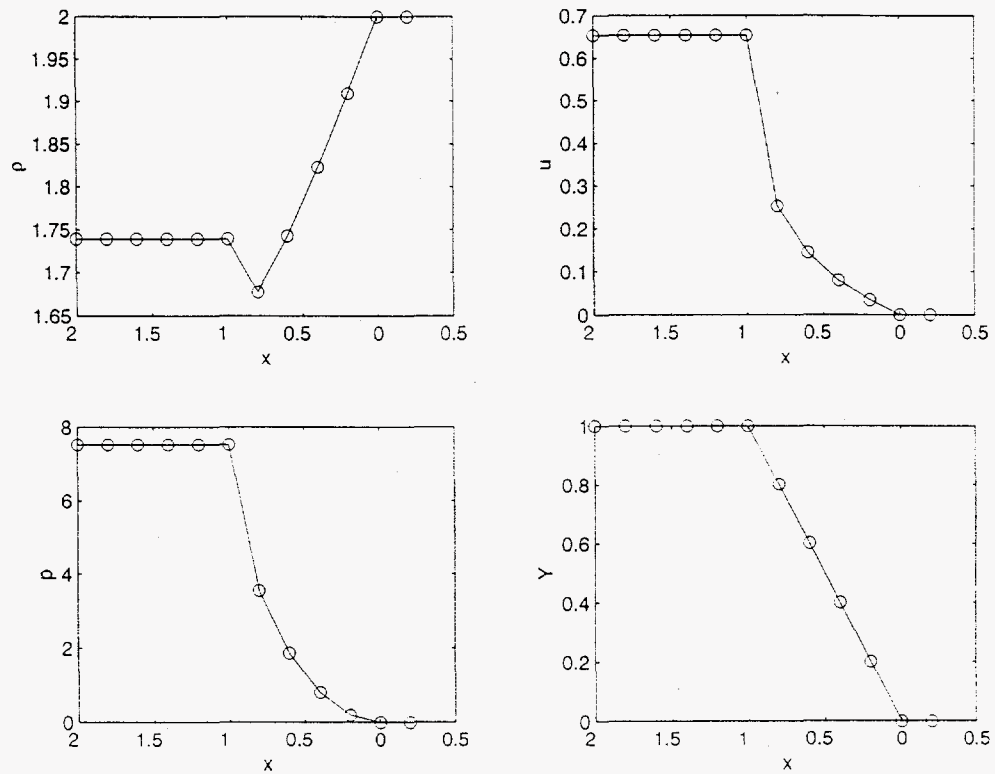


Figure 31: Plot of structure for $\kappa = 0.3284$, $\mu = 0$, $x_R = 1mm$, and $k = 5$. Model II.

which has as the solution $\vec{U} = \vec{U}(x - D_n t)$. We now discretize using MacCormack's scheme

$$\vec{U}_i^{n+1/2} = \vec{U}_i^n - \frac{\Delta t}{\Delta x} D_n [\vec{U}_i^n - \vec{U}_{i-1}^n] \quad (77)$$

for the predictor step, and for the corrector step

$$\vec{U}_i^{n+1} = \frac{1}{2} \left\{ \vec{U}_i^{n+1/2} + \vec{U}_i^n - \frac{\Delta t}{\Delta x} D_n [\vec{U}_{i+1}^{n+1/2} - \vec{U}_i^{n+1/2}] \right\}. \quad (78)$$

These two steps can be combined to obtain

$$\vec{U}_i^{n+1} = \vec{U}_i^n - \frac{\Delta t}{\Delta x} \frac{D_n}{2} [\vec{U}_{i+1}^n - \vec{U}_{i-1}^n] + \left(\frac{\Delta t}{\Delta x} \right)^2 \frac{D_n^2}{2} [\vec{U}_{i+1}^n - 2\vec{U}_i^n + \vec{U}_{i-1}^n]. \quad (79)$$

We now choose to require (73)-(74) and (79) hold simultaneously, as a requirement that the discrete wave be steady. Eliminating \vec{U}_i^{n+1} we obtain

$$\begin{aligned} & -\frac{\Delta t}{\Delta x} \frac{D_n}{2} [\vec{U}_{i+1}^n - \vec{U}_{i-1}^n] + \left(\frac{\Delta t}{\Delta x} \right)^2 \frac{D_n^2}{2} [\vec{U}_{i+1}^n - 2\vec{U}_i^n + \vec{U}_{i-1}^n] \\ & = -\frac{\Delta t}{2\Delta x} [\vec{F}_{i+1}^{n+1/2} - \vec{F}_i^{n+1/2} + \vec{F}_i^n - \vec{F}_{i-1}^n] + \frac{\Delta t}{2} [\vec{S}_i^{n+1/2} + \vec{S}_i^n], \end{aligned} \quad (80)$$

which can be simplified to

$$\begin{aligned} & D_n [\vec{U}_{i+1}^n - \vec{U}_{i-1}^n] - \frac{\Delta t}{\Delta x} D_n^2 [\vec{U}_{i+1}^n - 2\vec{U}_i^n + \vec{U}_{i-1}^n] \\ & = [\vec{F}_{i+1}^{n+1/2} - \vec{F}_i^{n+1/2} + \vec{F}_i^n - \vec{F}_{i-1}^n] - \Delta x [\vec{S}_i^{n+1/2} + \vec{S}_i^n]. \end{aligned} \quad (81)$$

Equation (81) is a nonlinear recurrence relation relating \vec{U}_{i+1}^n , \vec{U}_i^n , and \vec{U}_{i-1}^n . As before we can use this relation to solve for \vec{U}_{i-1}^n .

A.3.2 End State Analysis

Following the previous analysis, we determine the admissible end states that are consistent with the traveling wave recursion. Assume that the wave structure is centered on the grid from $i = -N$ to $i = 0$. We now multiple (81) by Δx and then sum over the wave structure. Applying these summation rules results in

$$\begin{aligned} & D_n [\vec{U}_0^n + \vec{U}_1^n - \vec{U}_{-N}^n - \vec{U}_{-(N+1)}^n] - \frac{\Delta t}{\Delta x} D_n^2 [\vec{U}_1^n - \vec{U}_0^n + \vec{U}_{-(N+1)}^n - \vec{U}_{-N}^n] \\ & = [\vec{F}_1^{n+1/2} - \vec{F}_{-N}^{n+1/2} + \vec{F}_0^n - \vec{F}_{-(N+1)}^n] - \Delta x \sum_{i=-N}^{i=0} [\vec{S}_i^{n+1/2} + \vec{S}_i^n] \end{aligned} \quad (82)$$

Now suppose that the summation is over a large enough domain so that

$$\vec{U}_0^n \approx \vec{U}_1^n, \quad \vec{U}_{-N}^n \approx \vec{U}_{-(N+1)}^n. \quad (83)$$

From (73) we see that if $i = 0$ or if $i = -N$, then we have the approximations

$$\vec{U}_i^{n+1/2} \approx \vec{U}_i^n, \quad (84)$$

and hence

$$\bar{F}_i^{n+1/2} \approx \bar{F}_i^n. \quad (85)$$

The last expression shows

$$\bar{F}_0^n - D_n \bar{U}_0^n - (\bar{F}^{*n} - D_n \bar{U}^{*n}) = \sum_{i=-N}^{i=0} \Delta x [\bar{S}_i^{n+1/2} + \bar{S}_i^n]. \quad (86)$$

Once the choice for $\bar{S} = \bar{Q} R_\delta$ has been made, the values of \bar{Q} are known explicitly, and it is then a simple matter to integrate the system to determine the structure.

As before, we assume

$$\bar{S}_i^{n+1/2} = \bar{Q} \frac{Y_i - Y_{i-1}}{\Delta x}, \quad \bar{S}_i^n = \bar{Q} \frac{Y_{i+1} - Y_i}{\Delta x}, \quad (87)$$

and hence

$$\begin{aligned} \sum_{i=-N}^{i=0} \Delta x [\bar{S}_i^{n+1/2} + \bar{S}_i^n] &= \sum_{i=-N}^{i=0} \bar{Q} \frac{Y_{i+1} - Y_{i-1}}{2} \\ &= -\bar{Q} \end{aligned} \quad (88)$$

Solving for \bar{Q} yields (68), which in component form, is identical to (42).

Figure 32 shows results using Model I for $\kappa = 0$. Note that the structure is quite different than that of the numerical schemes presented in the two previous sections. The structure is more diffusive and the end states are achieved much further downstream. Results of both models are identical for κ small. However, as κ increases, the results using Model I did not converge to the correct end states, while those of Model II did. Compare Figures 33 and 34. The correct star states for $\kappa = 0.3284$ are $\rho^* = 1.7389$, $u^* = 0.6528$ and $p^* = 7.5214$. Again, this suggests that the formulation using Model I is more difficult to resolve numerically than that using Model II.

A.4 Conclusions of the Steady Analysis

In this study we considered two different models for the equation of state. The first model uses an EOS that is used in most codes today. The model includes the ratio of the pressure and the burn fraction. The second model is the standard equation of state. Three numerical schemes are presented, and results for each scheme using the two models are compared.

Preliminary conclusions indicate that using the equation of state as defined in Model II have better numerical convergent properties than using the equation of state as defined in Model I. This fact seems independent of the underlying numerical scheme. In addition, MacCormack's scheme converges to the exact end states over a much larger domain than when using the first order schemes due to the artificial damping mechanism built into the scheme.

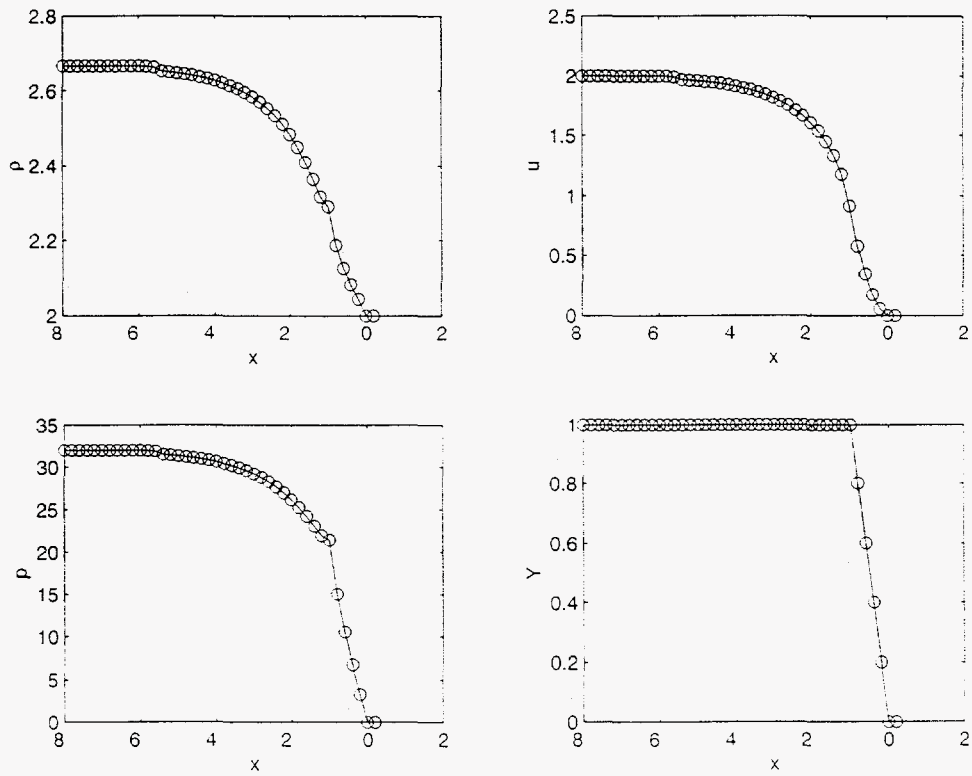


Figure 32: Plot of structure for $\kappa = 0$, $\mu = 0$, $x_R = 1mm$, and $k = 5$. Model I.

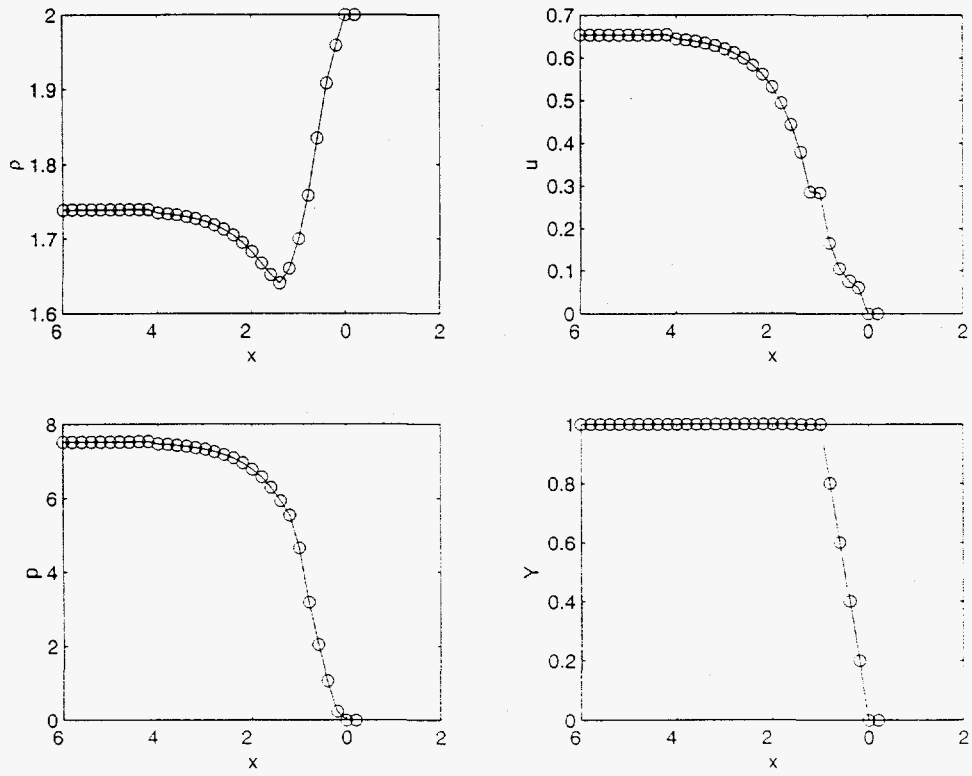


Figure 34: Plot of structure for $\kappa = .3284$, $\mu = 0$, $x_R = 1mm$, and $k = 5$. Model II.

References

- [1] Bdzil, J. B. Aslam, T. D. and Stewart, D. S.. "Resolved 3D Detonation Reaction Zones: Memory and Time Requirements for DNS". Los Alamos Report. LA-UR-97-792. Los Alamos National Laboratory, Los Alamos, NM (1997)
- [2] Wilkins, M. L., "Calculation of Elastic-Plastic Flow", in *Methods in Computational Physics, Advances in Research and Applications*, Volume 3, Fundamental Methods in Hydrodynamics, Alder, B., Fernbach, S. and Rotenberg, M. eds. Academic Press. (1964), p. 211 - 262.
- [3] Fickett, Wilson, "Detonation in Miniature".
- [4] Stewart, D. S., "The Dynamics of Multi-dimensional Detonation in Gaseous and Condensed Phases Explosives: Topical Review", in the Proceedings of the 27th (International) Conference on Combustion, Boulder, CO, July (1998).
- [5] Buckiet, Bruce, Random Choice Method.
- [6] Shu and Osher 1988
- [7] Jiang and Shu 1996
- [8] Xu Aslam and Stewart 1997
- [9] Whitham, G. B. **Linear and Nonlinear Waves** (1974) Wiley/Interscience pub. p. 277
- [10] J. Yao and D.S. Stewart, 1995.
- [11] J. Yao and D.S. Stewart, 1996.