Journal of Computational Physics 229 (2010) 3237–3249

Contents lists available at ScienceDirect

Journal of Computational Physics

journal homepage: www.elsevier.com/locate/jcp

A fully second order implicit/explicit time integration technique for hydrodynamics plus nonlinear heat conduction problems

Samet Y. Kadioglu *, Dana A. Knoll

Multiphysics Methods Group, Reactor Physics Analysis and Design, Idaho National Laboratory, P.O. Box 1625, MS 3840, Idaho Falls, ID 83415, United States

ARTICLE INFO

Article history:
Received 8 June 2009
Received in revised form 29 October 2009
Accepted 30 December 2009
Available online 11 January 2010

Keywords:
Hydrodynamics
Nonlinear heat conduction
Implicit/explicit algorithm

ABSTRACT

We present a fully second order implicit/explicit time integration technique for solving hydrodynamics coupled with nonlinear heat conduction problems. The idea is to hybridize an implicit and an explicit discretization in such a way to achieve second order time convergent calculations. In this scope, the hydrodynamics equations are discretized explicitly making use of the capability of well-understood explicit schemes. On the other hand, the nonlinear heat conduction is solved implicitly. Such methods are often referred to as IMEX methods [2,1,3]. The Jacobian-Free Newton Krylov (JFNK) method (e.g. [10,9]) is applied to the problem in such a way as to render a nonlinearly iterated IMEX method. We solve three test problems in order to validate the numerical order of the scheme. For each test, we established second order time convergence. We support these numerical results with a modified equation analysis (MEA) [21,20]. The set of equations studied here constitute a base model for radiation hydrodynamics.

1. Introduction

This paper presents a demonstrated fully second order implicit/explicit algorithm for solving hydrodynamics (e.g., conservation laws equations) plus nonlinear heat conduction (e.g., diffusion equation) problems. These equations can be viewed as a model for radiation hydrodynamics. Radiation hydrodynamics models are used in astrophysics, inertial confinement fusion, and other high temperature flow systems, and they represent a classic multiphysics hydrodynamics problem. These problems are difficult to tackle numerically, since they exhibit multiple time scales. For instance, radiation and hydrodynamics processes occur on time scales that differ from each other by many order of magnitudes. A common numerical solution strategy is to employ a hybrid implicit/explicit discretization [3,5,12]. Generally, the hydrodynamics part is treated explicitly, whereas an implicit method is applied to diffusion part. The motivation behind this hybridization is that if one uses all explicit discretizations, then due to very stiff diffusion process the explicit time steps are often impractically small to satisfy stability conditions [11]. Therefore it is often considered prudent to separate the diffusion part and treat it implicitly.

This paper can be considered as an advancement to [3]. For instance [3] solves hydrodynamics part (refer to Eq. (4)) explicitly and presents/compares three different implicit strategies for the diffusion part (refer to Eq. (5)). The explicit discretization in [3] is based on a Godunov type upwinding method. The first implicit strategy in [3] is based on the implicit Euler method and involves temperature linearization in the diffusion term. This is referred to as the *classical operator splitting* algorithm. The scheme results in solving a system of linear equations. The time accuracy for this method is only first order. The second approach is based on the implicit Euler discretization and involves no linearization. This results in a system of

* Corresponding author. Tel.: +1 208 526 4368; fax: +1 208 526 2930.
E-mail addresses: Samet.Kadioglu@inl.gov (S.Y. Kadioglu), dana.knoll@inl.gov (D.A. Knoll).

0021-9991/$ - see front matter Published by Elsevier Inc.
doi:10.1016/j.jcp.2009.12.039

nonlinear equations which are solved based on the Jacobian-Free Newton Krylov (JFNK) method [10]. The overall time accuracy is still first order. In the third strategy, the Jacobian-Free Newton Krylov method is implemented within a Predictor–Corrector framework. This employs a combination of the implicit Euler (prediction step) and the Crank–Nicolson method (correction step) [21]. The third method was an attempt to achieve second order time accuracy. Although it exhibits second order convergence for pure diffusion problems, it fails to be second order accurate when considering the hydrodynamics effects.

In all of the three implicit methods in [3], the hydrodynamics part is solved outside of the implicit block. This leads to operator splitting with un-converged nonlinearities. In other words, the hydrodynamics part has no, or limited, influence from the nonlinear iterations and vise versa. As a result, the nonlinearities in the coupled system are not converged completely. We note that the early IMEX methods [2,1] which are designed to solve convection-diffusion type problems (e.g., solving a combination of hyperbolic and parabolic equations similar to our system of equations) considers a similar solution strategy as in [3]. Basically, they employ an explicit and an implicit discretization for the convection and the diffusion part, respectively. The whole algorithm is implemented such a way that the explicit part is independent of the implicit loop and thus nonlinearities remain un-converged.

One way of converging nonlinearities for the radiation hydrodynamics system is to discretize the entire system implicitly. However, this results in (especially for multidimensional problems) a large nonlinear system of equations. It will be computationally more costly to solve this system let alone considering additional difficulties coming from the implicit shock capturing [22]. Furthermore, the fully implicit solution would result in a mixed hyperbolic–parabolic system. Our IMEX approach results in a much simpler parabolic scalar for this problem.

In this study, we introduce a nonlinearly converged IMEX strategy to achieve second order time convergence. The basic idea is to solve the hydrodynamics equations as part of a nonlinear function evaluation (e.g., when forming the IMEX function(refer to Section 3.3)) within the JFNK framework. In this way, there is a continuous interaction between the hydrodynamics (explicit block) and diffusion process (implicit block). In other words, the improved solutions at each nonlinear iteration (implicit block) are immediately felt by the explicit block, then the improved hydrodynamics solutions are readily available to form the next set of nonlinear residuals.

We remark that the reason why we choose the JFNK method as the preferred nonlinear solver for our nonlinearly converged IMEX algorithm is based on the fact that forming and storing the Jacobian matrix in a standard Newton method is often complicated. The complication even gets worse when using a combination of implicit and explicit discretizations. The JFNK method avoids forming or storing the Jacobian matrices. Therefore, it is our choice of method as the nonlinear solver in this study.

The organization of this paper is as follows. In Section 2, the governing equations are defined. In Section 3, the numerical solution procedure is described. Section 3 also includes our modified equation analysis. In Section 4, the computational results are presented. Section 5 contains our concluding remarks.

2. Governing equations

In this paper, we consider a *low-energy density* radiation hydrodynamics model (as in [3]) formulated in spherically symmetric coordinates.

$$\frac{\partial \rho}{\partial t} + \frac{1}{r^2}\frac{\partial}{\partial r}(r^2 \rho u) = 0, \tag{1}$$

$$\frac{\partial}{\partial t}(\rho u) + \frac{1}{r^2}\frac{\partial}{\partial r}(r^2 \rho u^2) + \frac{\partial p}{\partial r} = 0, \tag{2}$$

$$\frac{\partial E}{\partial t} + \frac{1}{r^2}\frac{\partial}{\partial r}[r^2 u(E+p)] = \frac{1}{r^2}\frac{\partial}{\partial r}\left(r^2 \kappa \frac{\partial T}{\partial r}\right), \tag{3}$$

where ρ, u, p, E, and T are the mass density, flow velocity, fluid pressure, total energy density of the fluid, and the fluid temperature, respectively. κ is the coefficient of thermal conduction (or diffusion coefficient) and in general is a nonlinear function of ρ and T. This simplified radiation hydrodynamics model allows one to study the dynamics of nonlinearly coupling two distinct physics, compressible fluid flow and nonlinear diffusion. In this study, we will use an ideal gas equation of state, i.e., $p = R\rho T = (\gamma - 1)\rho\epsilon$, where R is the specific gas constant per unit mass (which is equal to the universal gas constant divided by the molecular weight of the material), γ is the ratio of specific heats, and ϵ is the internal energy of the fluid per unit mass. Throughout this paper γ is set to $\gamma = \frac{5}{4}$. The coefficient of thermal conduction is assumed to be written as a power law in density and temperature, i.e., $\kappa = \kappa_0 \rho^a T^b$, where κ_0,a and b are constants [13].

As mentioned in the introduction, a common approach to solve Eqs. (1)–(3) is to split them into two pieces one being the pure hydrodynamics part (e.g., hyperbolic conservation laws) and the other accounting the effects of radiation transport (e.g., diffusion equation). This strategy (also known as operator splitting) is often used by many practitioners to deal with the multiple physics phenomena [4]. For instance, the pure hydrodynamics equations can be written as

$$\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial (A\mathbf{F})}{\partial V} + \frac{\partial \mathbf{G}}{\partial r} = 0, \tag{4}$$

where $\mathbf{U} = (\rho, \rho u, E)$, $\mathbf{F}(\mathbf{U}) = (\rho u, \rho u^2, u(E+p))$, and $\mathbf{G}(\mathbf{U}) = (0, p, 0)$. Then the diffusion equation becomes

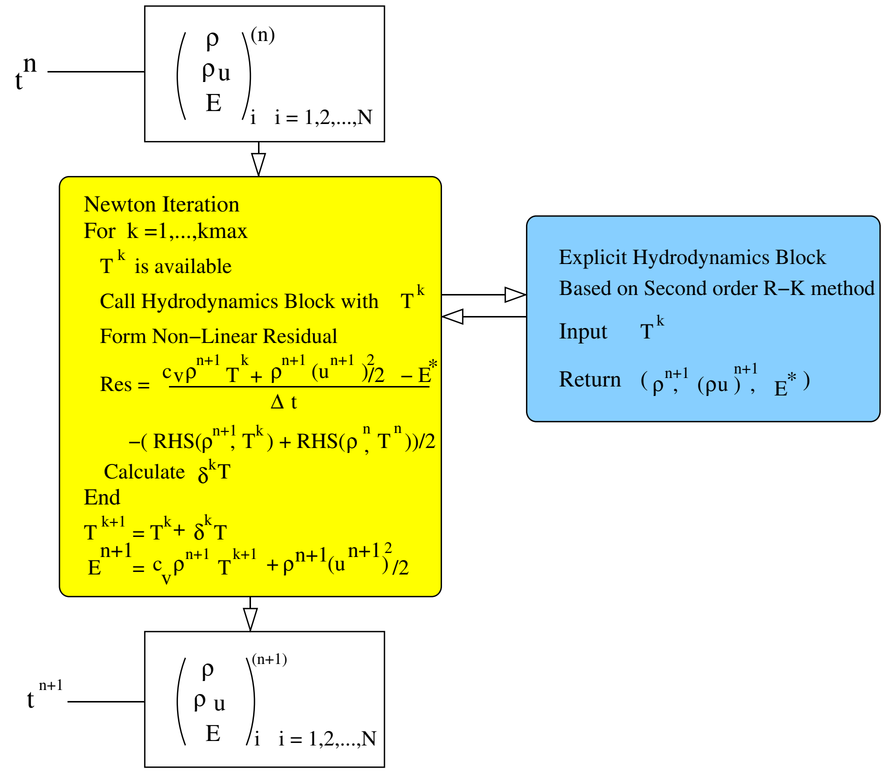
t^n
$\begin{pmatrix}\rho\\ \rho u\\ E\end{pmatrix}_i^{(n)}$ i = 1,2,...,N
Newton Iteration
For k =1,...,kmax
T^k is available
Call Hydrodynamics Block with T^k
Form Non–Linear Residual
$\mathrm{Res} = \dfrac{c_v \rho^{n+1} T^k + \rho^{n+1} (u^{n+1})^2/2 - E^*}{\Delta t}$
$-(\mathrm{RHS}(\rho^{n+1}, T^k) + \mathrm{RHS}(\rho^n, T^n))/2$
Calculate $\delta^k T$
End
$T^{k+1} = T^k + \delta^k T$
$E^{n+1} = c_v \rho^{n+1} T^{k+1} + \rho^{n+1} (u^{n+1})^2/2$
Explicit Hydrodynamics Block
Based on Second order R–K method
Input T^k
Return $(\rho^{n+1}, (\rho u)^{n+1}, E^*)$
t^{n+1}
$\begin{pmatrix}\rho\\ \rho u\\ E\end{pmatrix}_i^{(n+1)}$ i = 1,2,...,N

Fig. 1. Flowchart of the second order nonlinearly converged IMEX algorithm.

$$\frac{\partial E}{\partial t} = \frac{\partial}{\partial V}\left(A\kappa \frac{\partial T}{\partial r}\right). \tag{5}$$

Here $V = \frac{4}{3}\pi r^3$ is the generalized volume coordinate in one-dimensional spherical geometry, and $A = 4\pi r^2$ is the associated cross-sectional area. Notice that the total energy density, E, obtained by Eq. (4) just represents the hydrodynamics component and it must be augmented by Eq. (5).

3. Numerical algorithm

The numerical algorithm we present here consists of an explicit and an implicit block. The explicit block solves Eq. (4). The implicit block solves Eq. (5). We will briefly describe these algorithm blocks in Sections 3.1 and 3.2. The explicit block is embedded within the implicit block. In other words, at each nonlinear iteration the explicit block is called as part of a nonlinear function evaluation. This is done to obtain a nonlinearly converged algorithm that leads to second order calculations. We note that similar discretizations, but without converging nonlinearities, can lead to order reduction in time convergence [3].

The numerical algorithm is executed as follows. At beginning of each Newton iteration, we have the temperature values based on the current Newton iterate. This temperature is passed to the explicit block which returns the updated density, momentum, and a prediction to total energy density. Then we form residuals (e.g., forming the IMEX function) for the diffusion equation out of updated density, velocity, and predicted energy density fields. With the IMEX function in hand, we can execute the JFNK method. After the Newton method convergences, we get second order converged temperature and total energy density. We provided a schematic illustration of the algorithm execution in Fig. 1.

We remark that we also successfully tested this IMEX strategy in [8] for coupling neutron diffusion and thermomechanics in order to simulate transient behavior of fast burst metal reactors. One drawback of this algorithm implementation is that it is computationally more expensive in comparison to having explicit block independent of the implicit loop. Nevertheless, with this algorithm, one is able to achieve the second order time accuracy of the numerical scheme.

3.1. Explicit block

Our explicit time discretization is based on a second order TVD Runge–Kutta method [6,7,18,19]. The main reason why we choose this methodology is that it preserves the strong stability properties of the explicit Euler method. This is important

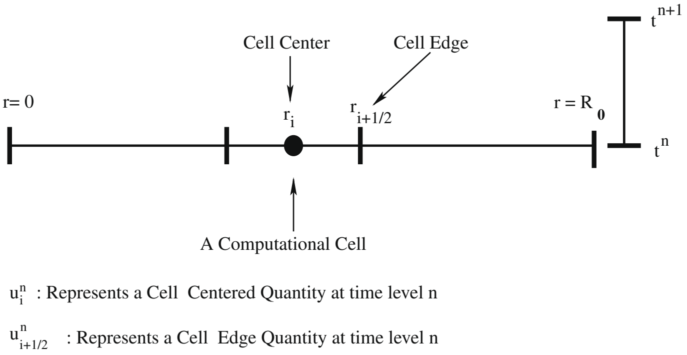
Cell Center
Cell Edge
r= 0
r_i
r_i+1/2
r = R_0
t^n+1
t^n
A Computational Cell
u_i^n : Represents a Cell Centered Quantity at time level n
u_i+1/2^n : Represents a Cell Edge Quantity at time level n

Fig. 2. Computational conventions.

because it is well known that solutions to the conservation laws usually involve discontinuities (e.g., shock or contact discontinuities) and [6,7] suggest that a time integration method which has the strong stability preserving property leads to non-oscillatory calculations (especially at shock or contact discontinuities).

A second order TVD Runge–Kutta method for $\mathbf{U}_t = L(\mathbf{U})$ can be cast as

$$\mathbf{U}^1 = \mathbf{U}^n + \Delta t L(\mathbf{U}^n), \tag{6}$$

$$\mathbf{U}^{n+1} = \frac{1}{2}\mathbf{U}^n + \frac{1}{2}\mathbf{U}^1 + \frac{1}{2}\Delta t L(\mathbf{U}^{1,*}). \tag{7}$$

Here, we assume $\mathbf{U} = (\rho, \rho u, E)$ and the operator L represents $-(F + G)$ in Eq. (4). The implicit temperature, i.e., T^k, at each kth Newton iteration is inserted in $L(\mathbf{U}^{1,*})$ such that $E^{1,*} = c_v \rho^1 T^k + \frac{1}{2}\rho^1(u^1)^2$ where $c_v = \frac{R}{\gamma-1}$ is the fluid specific heat. This provides the tight nonlinear coupling between the implicit and explicit blocks upon the convergence of the Newton iteration. The numerical L operator is evaluated based on the Local Lax Friedrichs fluxing procedure [11,22]. For instance, consider the following spatial discretization of Eq. (4),

$$\mathbf{U}_i^1 = \mathbf{U}_i^n - \frac{\Delta t}{\Delta V_i}\left(A_{i+1/2}F_{i+1/2}^n - A_{i-1/2}F_{i-1/2}^n\right) - \frac{\Delta t}{\Delta r}\left(G_{i+1/2}^n - G_{i+1/2}^n\right), \tag{8}$$

where $\Delta V_i = V(r_{i+1/2}) - V(r_{i-1/2})$ and $A_{i\pm 1/2} = A(r_{i\pm 1/2})$. Here indices i and $i + 1/2$ represent cell center and cell edge values, respectively (refer to Fig. 2). The *Local Lax Friedrichs method* defines $F_{i+1/2}$ as

$$F_{i+1/2} = \frac{F\left(\mathbf{U}_{i+1/2}^R\right) + F\left(\mathbf{U}_{i+1/2}^L\right)}{2} - \alpha_{i+1/2}\frac{\mathbf{U}_{i+1/2}^R - \mathbf{U}_{i+1/2}^L}{2}, \tag{9}$$

where $\alpha = max\{|\lambda_1^L|, |\lambda_1^R|, |\lambda_2^L|, |\lambda_2^R|, |\lambda_3^L|, |\lambda_3^R|\}$ in which $\lambda_1 = u - c$, $\lambda_2 = u$, $\lambda_3 = u + c$, and c is the sound speed. The sound speed is defined by

$$c = \sqrt{\frac{\partial p}{\partial \rho}}, \tag{10}$$

where $\frac{\partial p}{\partial \rho} = RT$ in this study. $\mathbf{U}_{i+1/2}^R$ and $\mathbf{U}_{i+1/2}^L$ in (9) are the interpolated values at $(i + 1/2)$th cell edge from the right and left side, i.e.,

$$\begin{aligned} \mathbf{U}_{i+1/2}^R &= \mathbf{U}_{i+1} - \frac{\Delta r}{2}\mathbf{U}_{r,i+1}, \\ \mathbf{U}_{i+1/2}^L &= \mathbf{U}_i + \frac{\Delta r}{2}\mathbf{U}_{r,i}, \end{aligned} \tag{11}$$

where

$$\mathbf{U}_{r,i} = minmod(a,b) = \begin{cases} a & \text{if } |a| < |b| \text{ and } ab > 0, \\ b & \text{if } |b| < |a| \text{ and } ab > 0, \\ 0 & \text{if } ab \leqslant 0, \end{cases} \tag{12}$$

where

$$a = \frac{\mathbf{U}_{i+1} - \mathbf{U}_i}{\Delta r}, \tag{13}$$

$$b = \frac{\mathbf{U}_i - \mathbf{U}_{i-1}}{\Delta r}. \tag{14}$$

For more algorithmic details regarding the explicit discretizations of conservation laws, we refer to [11,22].

3.2. Implicit block

The explicit block, which has been impacted by T^k, produces the following solution vector

$$\mathbf{U}^n \to \mathbf{U}^* = \begin{pmatrix} \rho^{n+1} \\ (\rho u)^{n+1} \\ E^* \end{pmatrix}.$$

This information is used to discretize Eq. (5) as follows,

$$\frac{(c_v \rho^{n+1} T^{n+1} + \frac{1}{2}\rho^{n+1}(u^{n+1})^2 - E^*)_i}{\Delta t} = \frac{1}{2}\frac{\partial}{\partial V}\left(A\kappa^{n+1}\frac{\partial T^{n+1}}{\partial r}\right)_i + \frac{1}{2}\frac{\partial}{\partial V}\left(A\kappa^{n}\frac{\partial T^{n}}{\partial r}\right)_i, \tag{15}$$

where

$$\frac{\partial}{\partial V}\left(A\kappa\frac{\partial T}{\partial r}\right)_i = \frac{A_{i+1/2}\kappa_{i+1/2}(T_{i+1} - T_i)/\Delta r}{\Delta V_i} - \frac{A_{i-1/2}\kappa_{i-1/2}(T_i - T_{i-1})/\Delta r}{\Delta V_i}. \tag{16}$$

This implicit discretization resembles the so called Crank–Nicolson method [21,20]. When the Newton method converges the nonlinearities in this discretization (e.g., Eq. (15)), we obtain the fully updated solution vector, i.e.,

$$\mathbf{U}^* \to \mathbf{U}^{n+1} = \begin{pmatrix} \rho^{n+1} \\ (\rho u)^{n+1} \\ E^{n+1} \end{pmatrix}$$

The nonlinear solver used to solve the implicit block is based on the Jacobian-Free Newton Krylov method [10,9].

3.3. The Jacobian-Free Newton Krylov method and forming the IMEX function

The Jacobian-Free Newton Krylov method is the combination of the Newton method that solves a system of nonlinear equations and a Krylov subspace method that solves the Newton correction equations. With this method, Newton-like super-linear convergence is achieved in the nonlinear iteration, without the complexity of forming or inverting the Jacobian matrix from a standard Newton method. The effects of the Jacobian are probed only through approximate matrix-vector products required in the Krylov iterations. A detailed description of the methodology behind the Jacobian-Free Newton Krylov method is given below.

The Newton method solves $\mathbf{F}(T) = 0$ (e.g., assume Eq. (15) is written in this form) iteratively over a sequence of linear system defined by

$$\mathbf{J}(T^k)\delta T^k = -\mathbf{F}(T^k), \quad T^{k+1} = T^k + \delta T^k, \quad k = 0, 1, \ldots \tag{17}$$

where $\mathbf{J}(T^k) = \frac{\partial \mathbf{F}}{\partial T}$ is the Jacobian matrix and δT^k is the update vector. The Newton iteration is terminated based on a required drop in the norm of the nonlinear residual, i.e.,

$$\|\mathbf{F}(T^k)\|_2 < tol_{res}\|\mathbf{F}(T^0)\|_2 \tag{18}$$

where tol_{res} is a given tolerance.

The linear system (17) (Newton correction equation) is solved by using the Arnoldi based Generalized Minimal RESidual method (GMRES) [16] which belongs to the general class of the Krylov subspace methods [14]. We note that these subspace methods are particularly suitable choice when dealing with non-symmetric linear systems. In GMRES, an initial linear residual, $\mathbf{r}_0$, is defined for a given initial guess δT_0,

$$\mathbf{r}_0 = -\mathbf{F}(T) - \mathbf{J}\delta T_0. \tag{19}$$

Here we dropped the index k convention since the Krylov (GMRES) iteration is performed at a fixed k. Let j be the Krylov iteration index. The jth Krylov iteration minimizes $\|\mathbf{J}\delta T_j + \mathbf{F}(T)\|_2$ within a subspace of small dimension, relative to n (the number of unknowns), in a least-squares sense. δT_j is drawn from the subspace spanned by the Krylov vectors $\{\mathbf{r}_0 \mathbf{J}\mathbf{r}_0 \mathbf{J}^2\mathbf{r}_0, \ldots \mathbf{J}^{j-1}\mathbf{r}_0\}$, and can be written as

$$\delta T_j = \delta T_0 + \sum_{i=0}^{j-1} \beta_i (\mathbf{J})^i \mathbf{r}_0, \tag{20}$$

where the scalar β_i minimizes the residual. Notice that each Krylov iteration requires one matrix-vector multiplication and these iterations are terminated based on a by product estimate of the residual that does not involve explicit construction of intermediate residual vectors [16]. One particularly attractive feature of this methodology is that it does not require forming the Jacobian matrix. Instead, only matrix-vector multiplications, $\mathbf{J}v$, are needed, where $v \in \{\mathbf{r}_0 \mathbf{J}\mathbf{r}_0 \mathbf{J}^2\mathbf{r}_0, \ldots\}$. This leads to the so-called *Jacobian-Free* implementations in which the action of the Jacobian matrix can be approximated by

$$\mathbf{J}v = \frac{\mathbf{F}(T + \epsilon v) - \mathbf{F}(T)}{\epsilon}, \tag{21}$$

where $\epsilon = \frac{1}{n\|v\|_2}\sum_{i=1}^{n} b|u_i| + b, n$ is the dimension of the linear system and b is a constant whose magnitude is within a few orders of magnitude of the square root of machine roundoff (typically 10^{-6} for 64-bit double precision).

Here, we briefly describe how to form the IMEX function $F(T)$. We will refer $F(T)$ as the IMEX function, since it uses the both explicit (hydrodynamics) and implicit (diffusion) information. Notice that for a method that uses all implicit information, $F(T)$ would correspond to a regular nonlinear residual function. The following pseudo code describes how to form $F(T)$ (we also refer to Fig. 1).

```
Evaluating $F(T^k)$:
Given $T^k$ where $k$ represents the current Newton iteration.
Call Hydrodynamics block with $(\rho^n, u^n, E^n, T^k)$ $\rho^{n+1}, u^{n+1}, E^*$ are produced
Form $F(T^k)$ based on the Crank–Nicolson method (Eqs. (15) and (16))
$F(T^k) = \frac{[c_v \rho^{n+1} T^k + \frac{1}{2}\rho^{n+1}(u^{n+1})^2 - E^*]}{\Delta t} - \frac{1}{2}\frac{\partial}{\partial V}\left(A\kappa^k \frac{\partial T^k}{\partial r}\right) - \frac{1}{2}\frac{\partial}{\partial V}\left(A\kappa^n \frac{\partial T^n}{\partial r}\right)$
```

It is important to note that we are not iterating between the implicit block and the explicit block. Instead we are executing the explicit block inside of a nonlinear function evaluation defined by $F(T^k)$. The unique properties of JFNK allow us to perform a Newton iteration on this IMEX function, and thus JFNK is a required component of this nonlinearly converged IMEX approach.

3.4. Modified equation analysis

In this section, we carry out a truncation error analysis to show that our proposed numerical procedure is second order. First, we consider the explicit block (more precisely, we consider the energy part of Eqs. (6) and (7))

$$E^1 = E^n - \Delta t \frac{\partial}{\partial x}\left\{u^n\left[p^n + c_v \rho^n T^n + \frac{1}{2}\rho^n (u^n)^2\right]\right\}, \tag{22}$$

$$E^* = \frac{E^n + E^1}{2} - \frac{\Delta t}{2}\frac{\partial}{\partial x}\left\{u^1\left[p^1 + c_v \rho^1 T^{n+1} + \frac{1}{2}\rho^1 (u^1)^2\right]\right\}, \tag{23}$$

where $*$ represents the explicitly calculated energy density. Here, we assume that the Newton iteration is already converged, i.e., $T^{n+1} = T^k$ at the kth iterate. Also notice that for simplicity, the spatial derivatives are written in one-dimensional cartesian coordinates. Substituting Eq. (22) into Eq. (23), we have

$$E^* = E^n - \frac{\Delta t}{2}\frac{\partial}{\partial x}\left\{u^n\left[p^n + c_v \rho^n T^n + \frac{1}{2}\rho^n (u^n)^2\right]\right\} - \frac{\Delta t}{2}\frac{\partial}{\partial x}\left\{u^1\left[p^1 + c_v \rho^1 T^{n+1} + \frac{1}{2}\rho^1 (u^1)^2\right]\right\}. \tag{24}$$

We use the following Taylor series

$$T^{n+1} = T^n + \Delta t \frac{\partial T^n}{\partial t} + \frac{\Delta t^2}{2}\frac{\partial^2 T^n}{\partial t^2} + O(\Delta t^3), \tag{25}$$

and let $L(E^n) = -\frac{\partial}{\partial x}\{u^n[p^n + c_v \rho^n T^n + \frac{1}{2}\rho^n (u^n)^2]\}$, then (24) becomes

$$E^* = E^n + \frac{\Delta t}{2} L(E^n) - \frac{\Delta t}{2}\frac{\partial}{\partial x}\left\{u^1\left[p^1 + c_v \rho^1 \left(T^n + \Delta t \frac{\partial T^n}{\partial t} + \frac{\Delta t^2}{2}\frac{\partial^2 T^n}{\partial t^2} + O(\Delta t^3)\right) + \frac{1}{2}\rho^1 (u^1)^2\right]\right\}. \tag{26}$$

Now, we insert $T^n = T^1 - \Delta t \frac{\partial T^n}{\partial t} - \frac{\Delta t^2}{2}\frac{\partial^2 T^n}{\partial t^2} + O(\Delta t^3)$ in (26) to get

$$E^* = E^n + \frac{\Delta t}{2} L(E^n) + \frac{\Delta t}{2} L(E^1) + O(\Delta t^3), \tag{27}$$

where $L(E^1) = -\frac{\partial}{\partial x}\left\{u^1\left[p^1 + c_v \rho^1 T^1 + \frac{1}{2}\rho^1 (u^1)^2\right]\right\}$. Further simplification comes from the following

$$L(E^1) = L(E^n) + \Delta t \frac{\partial L}{\partial t} + O(\Delta t^2). \tag{28}$$

Substituting (28) into (27), we get

$$E^* = E^n + \Delta t L(E^n) + \frac{\Delta t^2}{2}\frac{\partial L}{\partial t} + O(\Delta t^3). \tag{29}$$

Now, we consider the implicit block (Eq. (15))

$$\frac{E^{n+1} - E^*}{\Delta t} = \frac{1}{2}\frac{\partial}{\partial x}\left(\kappa^{n+1}\frac{\partial T^{n+1}}{\partial x}\right) + \frac{1}{2}\frac{\partial}{\partial x}\left(\kappa^n \frac{\partial T^n}{\partial x}\right). \tag{30}$$

Again, the equation is written in cartesian coordinates for simplicity and we assume that the Newton iteration is already converged. We consider the following Taylor series

$$E^{n+1} = E^n + \Delta t \frac{\partial E^n}{\partial t} + \frac{\Delta t^2}{2} \frac{\partial^2 E^n}{\partial t^2} + O(\Delta t^3), \tag{31}$$

$$\kappa^{n+1} = \kappa^n + \Delta t \frac{\partial \kappa^n}{\partial t} + \frac{\Delta t^2}{2} \frac{\partial^2 \kappa^n}{\partial t^2} + O(\Delta t^3). \tag{32}$$

Using (31), (32), (29), and (25) in (30), the truncation term becomes

$$\begin{aligned}\tau^n = {} & E^n + \Delta t \frac{\partial E^n}{\partial t} + \frac{\Delta t^2}{2} \frac{\partial^2 E^n}{\partial t^2} - \left(E^n + \Delta t L(E^n) + \frac{\Delta t^2}{2} \frac{\partial L(E^n)}{\partial t} \right) \\ & - \Delta t \frac{1}{2} \frac{\partial}{\partial x} \left[\left(\kappa^n + \Delta t \frac{\partial \kappa^n}{\partial t} + \frac{\Delta t^2}{2} \frac{\partial^2 \kappa^n}{\partial t^2} \right) \frac{\partial}{\partial x} \left(T^n + \Delta t \frac{\partial T^n}{\partial t} + \frac{\Delta t^2}{2} \frac{\partial^2 T^n}{\partial t^2} \right) \right] - \Delta t \frac{1}{2} \frac{\partial}{\partial x} \left(\kappa^n \frac{\partial T^n}{\partial x} \right) + O(\Delta t^3).\end{aligned} \tag{33}$$

Cancelling the common terms and grouping the other terms together, we get

$$\begin{aligned}\tau^n = {} & \Delta t \left[\frac{\partial E^n}{\partial t} - L(E^n) \right] + \frac{\Delta t^2}{2} \frac{\partial}{\partial t} \left[\frac{\partial E^n}{\partial t} - L(E^n) \right] - \left[\frac{\Delta t}{2} \frac{\partial}{\partial x} \left(\kappa^n \frac{\partial T^n}{\partial x} \right) + \frac{\Delta t}{2} \frac{\partial}{\partial x} \left(\kappa^n \frac{\partial T^n}{\partial x} \right) \right] \\ & - \left[\frac{\Delta t^2}{2} \frac{\partial}{\partial x} \left(\kappa^n \frac{\partial T^n_t}{\partial x} \right) + \frac{\Delta t^2}{2} \frac{\partial}{\partial x} \left(\kappa^n_t \frac{\partial T^n}{\partial x} \right) \right] + O(\Delta t^3).\end{aligned} \tag{34}$$

This further simplifies by using

$$\frac{\partial}{\partial t} \left[\frac{\partial}{\partial x} \left(\kappa^n \frac{\partial T^n}{\partial x} \right) \right] = \frac{\partial}{\partial x} \left(\kappa^n_t \frac{\partial T^n}{\partial x} \right) + \frac{\partial}{\partial x} \left(\kappa^n \frac{\partial T^n_t}{\partial x} \right). \tag{35}$$

Then we have

$$\tau^n = \Delta t \left[\frac{\partial E^n}{\partial t} - L(E^n) - \frac{\partial}{\partial x} \left(\kappa^n \frac{\partial T^n}{\partial x} \right) \right] + \frac{\Delta t^2}{2} \frac{\partial}{\partial t} \left[\frac{\partial E^n}{\partial t} - L(E^n) - \frac{\partial}{\partial x} \left(\kappa^n \frac{\partial T^n}{\partial x} \right) \right] + O(\Delta t^3). \tag{36}$$

From the energy equation (e.g., Eq. (3)) we know that $\frac{\partial E^n}{\partial t} - L(E^n) - \frac{\partial}{\partial x} \left(\kappa^n \frac{\partial T^n}{\partial x} \right) = 0$, thus (36) becomes $\tau^n = O(\Delta t^3)$. This concludes that our numerical procedure is second order. Given this analysis, we expect that our combined IMEX method will be second order accurate in time.

3.5. Time step control

The time step control criteria we use here is originally proposed in [15]. The idea is to estimate the dominant wave propagation speed in the problem. In one dimension this involves calculating the ratio of temporal to spatial derivatives of the dependent variables. In principle, it is sufficient to consider the following hyperbolic equation rather than using the entire system of the governing equations

$$\frac{\partial E}{\partial t} + \upsilon_f \frac{\partial E}{\partial r} = 0, \tag{37}$$

where the unknown υ_f represents the front velocity. This gives

$$\upsilon_f = -\frac{\partial E / \partial t}{\partial E / \partial r}. \tag{38}$$

As noted in [15], to avoid problems from lack of smoothness the following numerical approximation is used to calculate υ_f

$$\upsilon_f^n = \frac{\sum(|E_i^n - E_i^{n-1}| / \Delta t)}{\sum(|E_{i+1}^n - E_{i-1}^n| / 2\Delta r)}. \tag{39}$$

Then the new time step is determined by the Courant–Friedrichs–Lewy (CFL) condition

$$\Delta t^{n+1} = C \frac{\|\Delta r\|}{\upsilon_f^n}, \tag{40}$$

where $\|\Delta r\|$ uses the L_1 norm as in Eq. (39). We can further simplify Eq. (40) by using Eq. (39), i.e.,

$$\Delta t^{n+1} = \frac{1}{2} \frac{\sum |E_{i+1}^n - E_{i-1}^n|}{\sum(|E_i^n - E_i^{n-1}| / \Delta t)}. \tag{41}$$

We remark that the time steps determined by this procedure is always compared with the pure hydrodynamics time steps and the most restrictive ones are selected. The hydrodynamics time steps are calculated by

$$\Delta t^{Hydro,n+1} = CFL \times \frac{\Delta r}{max_i |u + c|_i}, \tag{42}$$

where u is the fluid velocity and c is the sound speed (e.g., refer to Eq. (10)). The coefficient CFL is set to 0.5.

Alternative time step control criterion are used for radiation hydrodynamics problems [4]. One commonly used approach is based on the observation of the maximum relative changes in E. For instance,

$$\Delta t^{n+1} = \Delta t^n \sqrt{\frac{(\Delta E/E)^{n+1}}{(\Delta E/E)_{max}}}, \tag{43}$$

where

$$\left(\frac{\Delta E}{E}\right)^{n+1} = max_i \left(\frac{|E_i^{n+1} - E_i^n|}{E_i^{n+1} + E_0}\right), \tag{44}$$

where the parameter E_0 is an estimate for the lower bound of the energy density. Comparing (43) to (41) we observed that (41) is computationally more efficient. Therefore, we use (41) in our test calculations.

4. Numerical results

4.1. Smooth problem test

In this test, we run the code until a particular final time so that the computational solutions are free of shock waves and steep thermal fronts. The problem is to follow the evolution of the nonlinear waves that results from an initial total energy deposition. The initial total energy density is given by

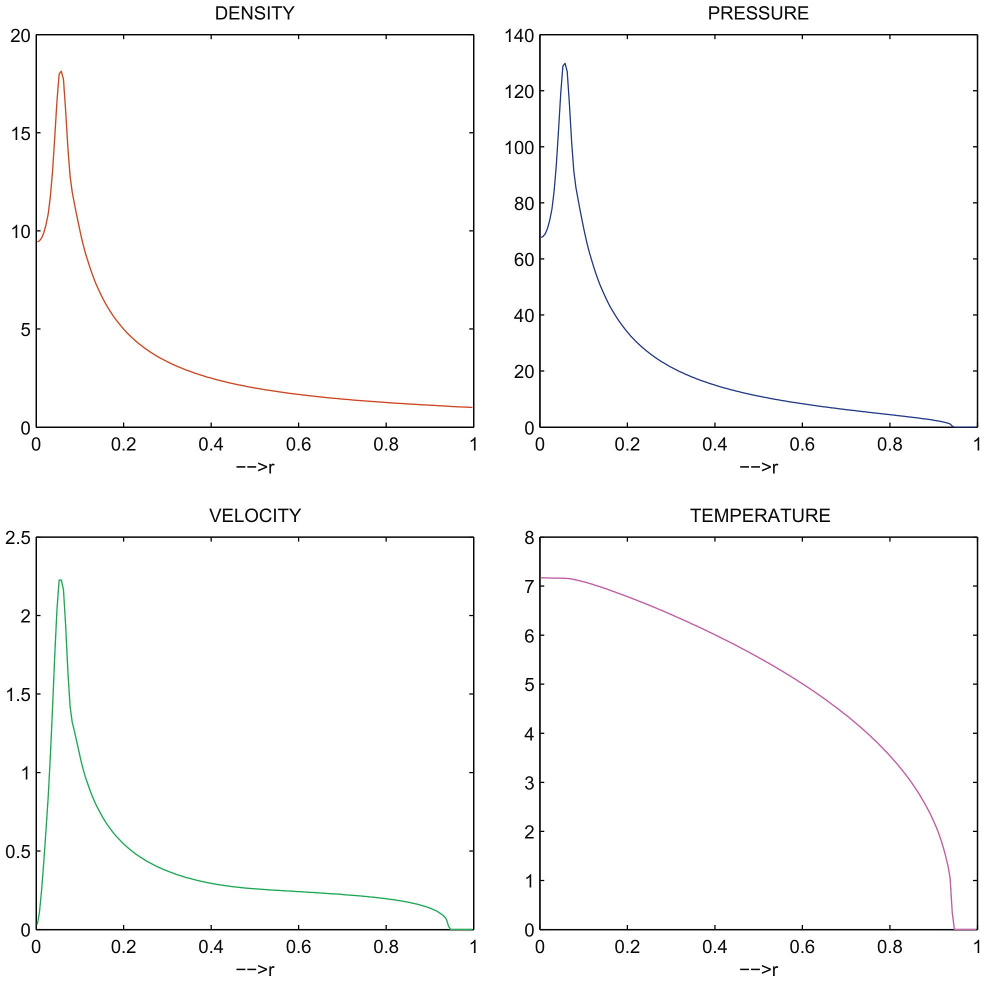

Fig. 3. Solution profiles resulting from the smooth problem test. The solutions are calculated for $t_{final} = 0.01$ with $M = 200$ cell points.

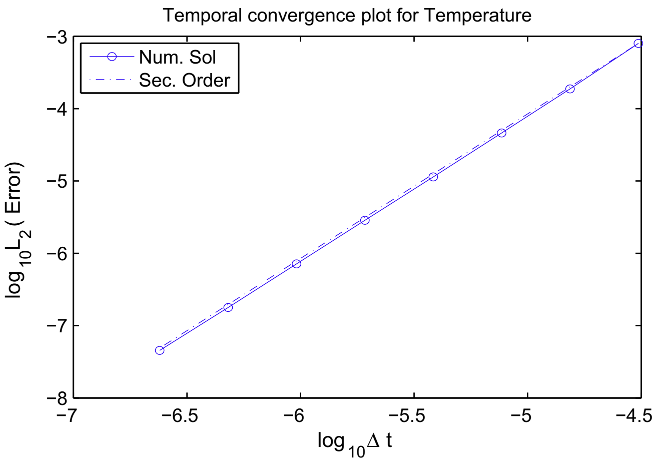

Fig. 4. Temporal convergence plot for the smooth problem test. $t_{final} = 0.01$ with $M = 200$ cell points.

$$E(r,0) = \frac{\varepsilon_0 \exp(-r^2/c_0^2)}{(c_0\sqrt{\pi})^3}, \tag{45}$$

where c_0 is a constant and set to 1/4 for this test. Note that $c_0 \to 0$ gives a delta function at origin. We use the cell averaged values of E as in [3]. In other words, we integrate Eq. (45) over the ith cell from $r_{i-1/2}$ to $r_{i+1/2}$ so that

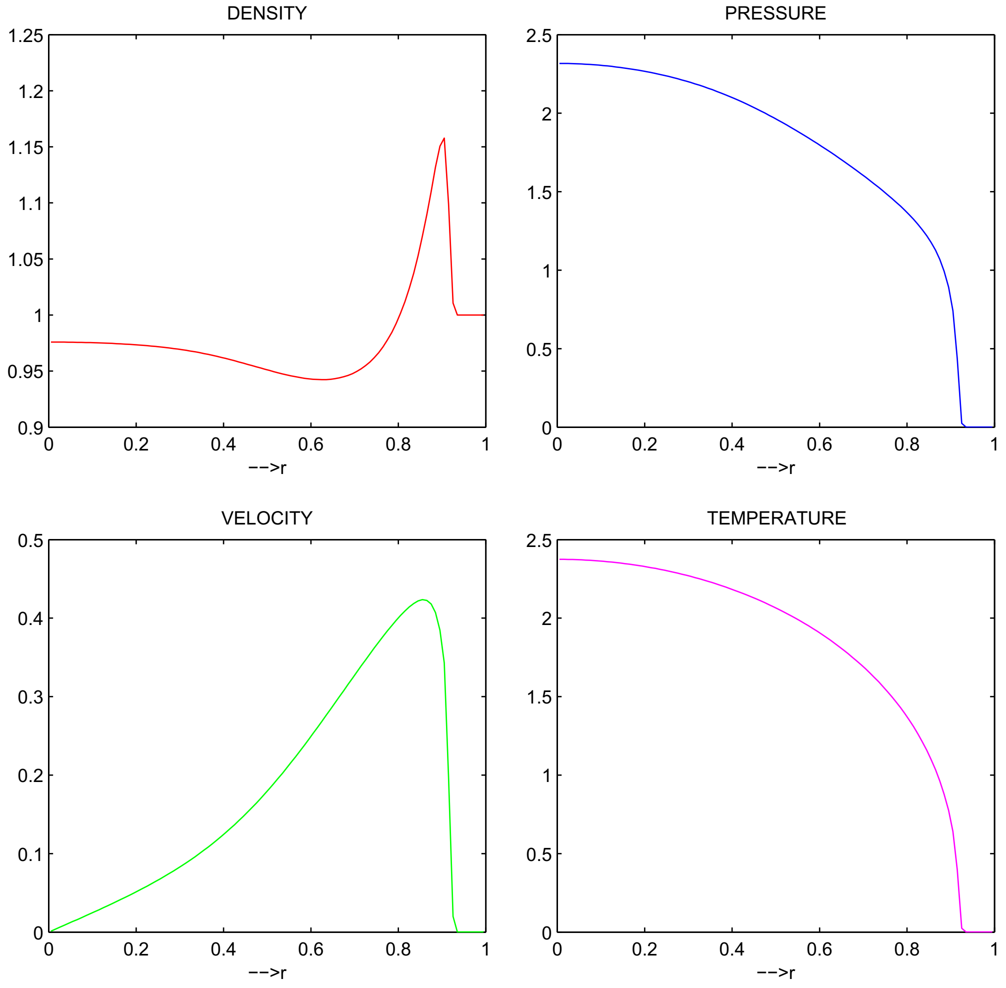

Fig. 5. Solution profiles resulting from the weak point explosion test. The solutions are calculated for $t_{final} = 0.05$ with $M = 100$ cell points.

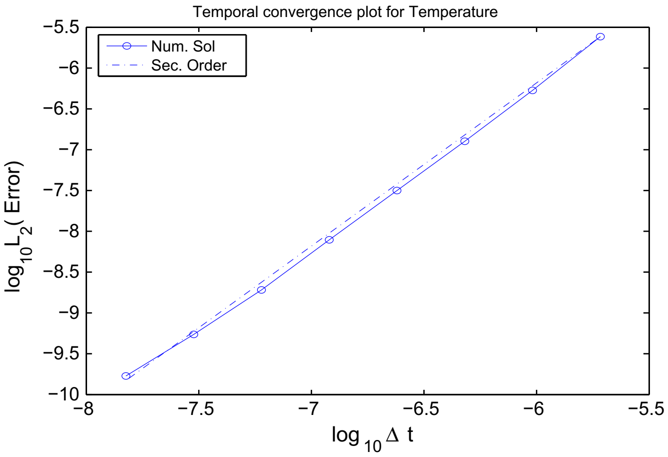

Fig. 6. Temporal convergence plot for the weak point explosion test. $t_{final} = 0.05$ with $M = 100$ cell points.

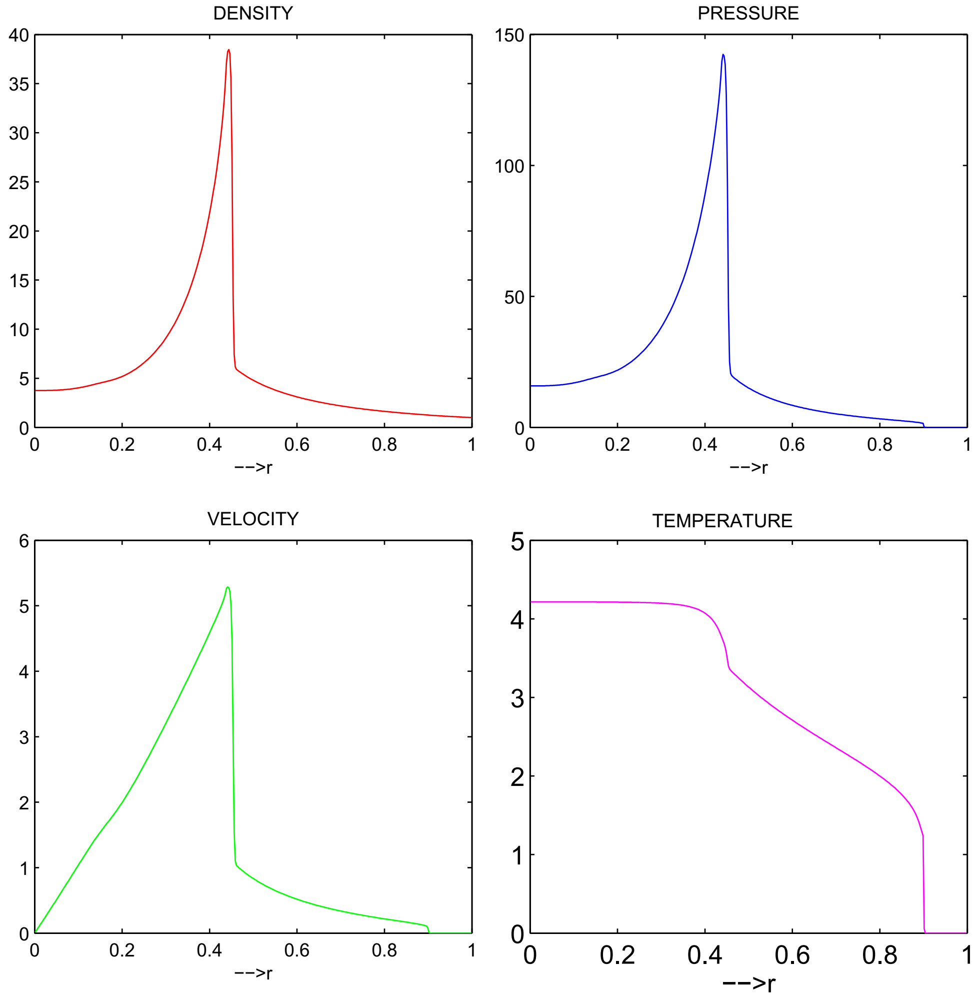

Fig. 7. Solution profiles resulting from the strong point explosion test. The solutions are calculated for $t_{final} = 0.05145$ with $M = 400$ cell points.

$$E_i = \frac{\varepsilon_0[erf(r_{i+1/2}/c_0) - erf(r_{i-1/2}/c_0)] - 2\pi c_0^2[r_{i+1/2}E(r_{i+1/2}) - r_{i-1/2}E(r_{i-1/2})]}{\Delta V_i}, \tag{46}$$

where the symbol *erf* denotes the error function. The initial density is set to $\rho = 1/r$. The initial temperature is calculated by using $E = c_v\rho T + \frac{1}{2}\rho u$ where the initial $u = 0$. The boundary conditions for the hydrodynamics variables are *reflective* and *outflow* boundary conditions at the left and right ends of the computational domain, respectively. The zero-flux boundary conditions are used for the temperature at both ends (e.g., $\partial T/\partial r|_{r=0} = 0$). Finally, the coefficient of thermal conduction is set to $\kappa(T) = T^{5/2}$.

We run the code until $t = 0.01$ with $\varepsilon_0 = 100$ and 200 cell points. The size of the computational domain is set to 1 (e.g., $R_0 = 1$ in Fig. 2). Fig. 3 shows the computed solutions for density, pressure, velocity, and temperature. As can be seen, there is no shock formation or steep thermal fronts occurred around this time. Fig. 4 shows our numerical time convergence analysis. To measure the rate of time convergence, we run the code with a fixed mesh (e.g., $M = 200$ cell points) and different time step refinements to a final time (e.g., $t = 0.01$). Then we measure the L_2 norm of errors between *two* consecutive time step refinements and plot the rate of decrease in these errors. It is clear from Fig. 4 that we established second order time convergence. We remark that [3] does not present second order accurate results for this test.

4.2. Point explosions

In this section, we present our numerical findings for point explosion problems in which shock discontinuities and steep thermal fronts exist. A detailed problem description can be found in [17,3]. A point explosion is characterized by the release of a large amount of energy in a small region of space. Depending on the magnitude of the energy deposition, we will have weak or strong explosions. If the initial explosion energy is not large enough, the diffusive effect is limited to region behind

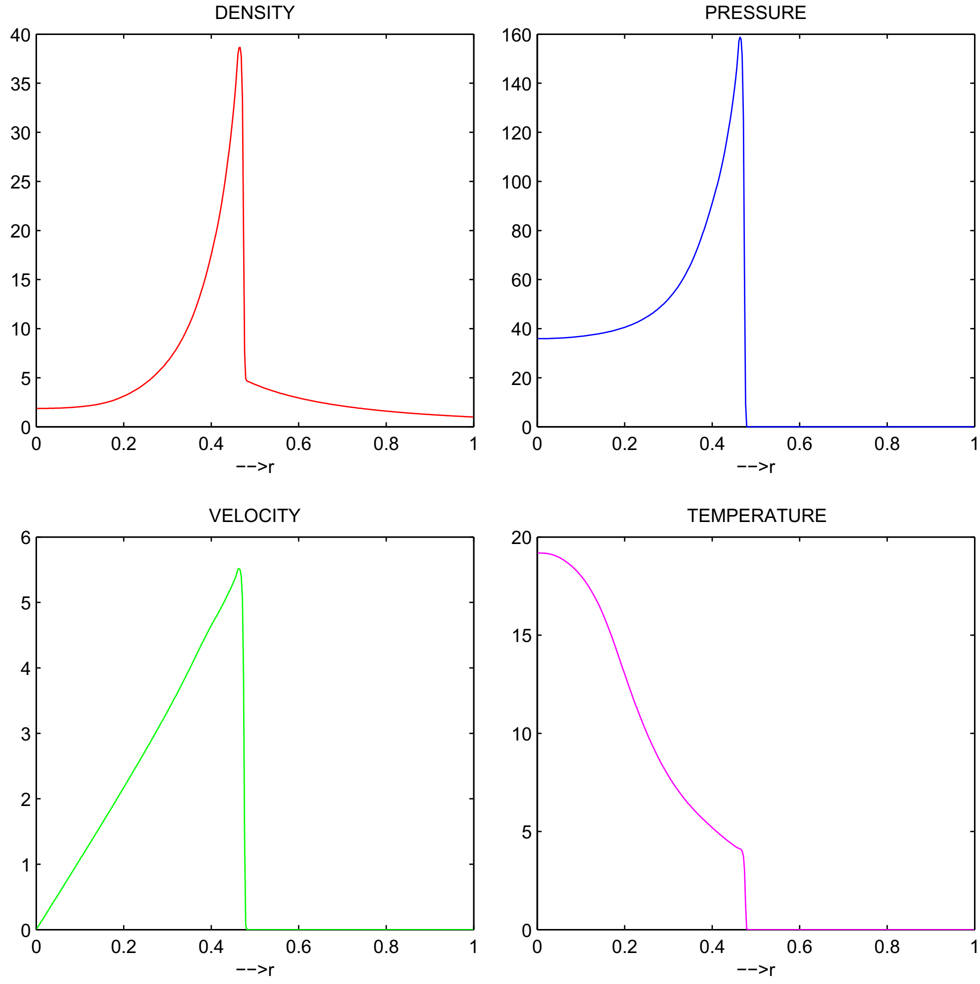

Fig. 8. Pure hydrodynamics solutions resulting from the strong point explosion test. The solutions are calculated for $t_{final} = 0.05145$ with $M = 400$ cell points.

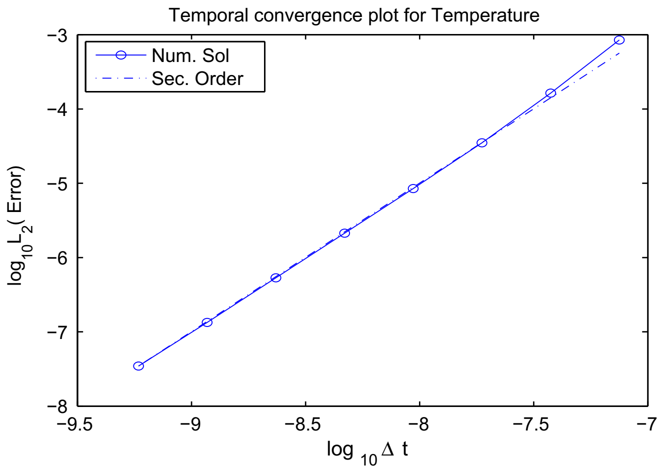

Fig. 9. Temporal convergence plot for the strong point explosion test. $t_{final} = 0.001$ with $M = 400$ cell points.

the shock. However, if the explosion energy is large, then the thermal front precedes the hydrodynamics front. These behaviors have been observed in our weak and strong explosion tests.

For a weak explosion test, we set $\varepsilon_0 = 20$ $c_0 = 1/4$ in Eq. (46) with ambient density $\rho = 1$ and the coefficient of the thermal conduction $\kappa(T) = T^{5/2}$. The problem evolves to the final time $t = 0.05$ with 100 cell points. The computational domain size is set to 1 for this and the next test. Fig. 5 depicts the profiles of density, pressure, velocity, and temperature at this time. In this case, there is shock formation in the solutions near $r = 0.9$. Notice that the thermal front does not exceed the shock front. Fig. 6 shows the time convergence analysis. Clearly, we obtained second order accurate results. We note that [3] suffers from order reduction (i.e., [3] is first order for this test).

Next, we present strong explosion test results. We set $\varepsilon_0 = 235 c_0 = 1/300$ in Eq. (46) with $\rho = r^{-2.111}$ and $\kappa(\rho, T) = \frac{T^{13/2}}{\rho^2}$. Notice that the nonlinearity in the coefficient of the thermal conduction is much larger in this test and κ now depends on ρ which comes from the explicit hydrodynamics. This time we compute the solutions until $t = 0.05145$ with 400 cell points. Fig. 7 shows density, pressure, velocity, and temperature profiles. In this case, a hydrodynamical shock is formed around $r = 0.45$. The hydrodynamical shock is more apparent when we run the pure hydrodynamics code (e.g., neglecting the heat conduction term in Eq. (3). Refer to results in Fig. 8). At this time ($t = 0.05145$), the thermal front (located near $r = 0.9$) propagates faster than the hydrodynamical shock due to large initial energy deposition. Fig. 9 shows the time convergence analysis. Again, we obtained second order convergence.

5. Conclusion

We have presented a new JFNK-based implicit/explicit time integration technique for solving hydrodynamics plus nonlinear heat conduction problems (e.g., radiation hydrodynamics in the low-energy density diffusion limit). The key to implement our nonlinearly converged implicit/explicit algorithm is to carry the explicit integration as part of a nonlinear function evaluation within the implicit block. This way, the improved time accuracy of the nonlinear iteration is readily felt by the explicit block and vice versa. We have performed a modified equation analysis (truncation error analysis) to show that our proposed numerical method is second order. Clearly, our numerical results verify that we have obtained second order time convergent calculations. The algorithm is tested for only low-energy radiation hydrodynamics problems. The extension to the high-energy radiation hydrodynamics is currently being investigated and the outcome will be presented in a separate journal paper. Additionally, we expect there are other multiple time scale nonlinear systems where this JFNK-based IMEX algorithm can make a positive impact.

Acknowledgment

The submitted manuscript has been authored by a contractor of the US Government under Contract No. DEAC07-05ID14517 (INL/JOU-09-16056). Accordingly, the US Government retains a non-exclusive, royalty-free license to publish or reproduce the published form of this contribution, or allow others to do so, for US Government purposes.

References

[1] U.M. Ascher, S.J. Ruuth, R.J. Spiteri, Implicit–explicit Runge–Kutta methods for time dependent partial differential equations, Appl. Numer. Math. 25 (1997) 151–167.
[2] U.M. Ascher, S.J. Ruuth, B. Wetton, Implicit-explicit methods for time dependent PDE's, SIAM J. Numer. Anal. 32 (1995) 797–823.

[3] J.W. Bates, D.A. Knoll, W.J. Rider, R.B. Lowrie, V.A. Mousseau, On consistent time-integration methods for radiation hydrodynamics in the equilibrium diffusion limit: low-energy-density regime, J. Comput. Phys. 167 (2001) 99–130.

[4] R.L. Bowers, J.R. Wilson, Numerical Modeling in Applied Physics and Astrophysics, Jones and Bartlett, Boston, 1991.

[5] W. Dai, P.R. Woodward, Numerical simulations for radiation hydrodynamics. I. diffusion limit, J. Comput. Phys. 142 (1998) 182.

[6] S. Gottlieb, C.W. Shu, Total variation diminishing Runge–Kutta schemes, Math. Comput. 221 (1998) 73–85.

[7] S. Gottlieb, C.W. Shu, E. Tadmor, Strong stability-preserving high-order time discretization methods, Siam Rev. 43-1 (2001) 89–112.

[8] S.Y. Kadioglu, D.A. Knoll, C. Oliveria, Multi-physics analysis of spherical fast burst reactors, Nucl. Sci. Eng. 163 (2009) 1–12.

[9] C.T. Kelley, Solving nonlinear equations with Newton's method, Siam (2003).

[10] D.A. Knoll, D.E. Keyes, Jacobian-Free Newton Krylov methods: a survey of approaches and applications, J. Comput. Phys. 193 (2004) 357–397.

[11] R.J. Leveque, Finite Volume Methods for Hyperbolic Problems, Cambridge University Press, 1998 (Texts in Applied Mathematics).

[12] R.B. Lowrie, J.E. Morel, J.A. Hittinger, The coupling of radiation and hydrodynamics, Astrophys. J. 521 (1999) 432.

[13] R.E. Marshak, Effect of radiation on shock wave behavior, Phys. Fluids 1 (1958) 24–29.

[14] J.K. Reid, On the methods of conjugate gradients for the solution of large sparse systems of linear equations. Large sparse sets of linear equations, Academic Press, New York, 1971.

[15] W.J. Rider, D.A. Knoll, Time step size selection for radiation diffusion calculations, J. Comput. Phys. 152-2 (1999) 790–795.

[16] Y. Saad, Iterative methods for sparse linear systems, Siam (2003).

[17] A.I. Shestakov, Time-dependent simulations of point explosions with heat conduction, Phys. Fluids 11 (1999) 1091–1095.

[18] C.W. Shu, S. Osher, Efficient implementation of essentially non-oscillatory shock capturing schemes, J. Comput. Phys. 77 (1988) 439.

[19] C.W. Shu, S. Osher, Efficient implementation of essentially non-oscillatory shock capturing schemes II, J. Comput. Phys. 83 (1989) 32.

[20] J.C. Strikwerda, Finite Difference Schemes Partial Differential Equations, Wadsworth & Brooks/Cole, Advance Books & Software, Pacific Grove, CA, 1989.

[21] J.W. Thomas, Numerical Partial Differential Equations I (Finite Difference Methods), Springer-Verlag, New York, 1998 (Texts in Applied Mathematics).

[22] J.W. Thomas, Numerical Partial Differential Equations II (Conservation Laws and Elliptic Equations), Springer-Verlag, New York, 1999 (Texts in Applied Mathematics).