Available online at www.sciencedirect.com
SCIENCE DIRECT®
Journal of Computational Physics 196 (2004) 566–590
JOURNAL OF COMPUTATIONAL PHYSICS
www.elsevier.com/locate/jcp

A comparison of implicit time integration methods for nonlinear relaxation and diffusion ☆

Robert B. Lowrie *

Computer and Computational Sciences Division (CCS-2), Los Alamos National Laboratory, Mail Stop D413, Los Alamos, NM 87545, USA

Received 4 September 2002; received in revised form 3 November 2003; accepted 6 November 2003

Abstract

Several time integration methods for nonlinear systems are compared. All of the time discretizations are based on the θ-method, but differ in their treatment of the implicit nonlinear terms. One method converges the implicit nonlinear terms to a small tolerance and is often referred to as nonlinearly consistent (NC). Newton's method, or its approximation Newton–Krylov, is used to converge the nonlinearities. The other methods considered are linearized and comparisons are made for a relaxation problem and a radiation diffusion problem. The linearized one-step method that uses the full Jacobian is shown to have similar accuracy as NC methods. The lagged linearization method and an extension that is second-order accurate are also studied. A truncation error analysis complements the numerical results. For the relaxation problem, it is shown that each of the second-order accurate linearized methods may be more accurate than an NC method, depending on the degree of nonlinearity in the problem. For the radiation diffusion problem, in general the NC method is most accurate and allows a larger time step. However, the linearized methods perform surprisingly well.

Keywords: Newton–Krylov method; Nonlinear systems; Implicit differencing; Radiation diffusion

1. Introduction

The implicit time integration of nonlinear systems has two important aspects

1. The time discretization. Some examples are implicit Runge–Kutta, backward Euler, and Crank–Nicolson.
2. The treatment of the implicit nonlinear terms. One example is to converge the nonlinearities to a small tolerance. Another possibility is to linearize the implicit terms, such as evaluating the solution-dependent coefficients at the old time level.

☆ This work was performed under the auspices of the US Department of Energy by Los Alamos National Laboratory under Contract W-7405-ENG-36.
* Tel.: +1-505-667-2121; fax: +1-505-667-3726.
E-mail address: lowrie@lanl.gov (R.B. Lowrie).

0021-9991/$ - see front matter
doi:10.1016/j.jcp.2003.11.016

This study will compare various treatments of the implicit nonlinear terms for two problems that are related to radiation diffusion, both of which have strong nonlinear transients. The comparisons will be made for a single second-order time discretization (Crank–Nicolson), which is commonly used when the nonlinear terms are converged (see, for example [2,5,17]).

Previous research that is closely related to the present work is that of Knoll et al. [9], who study the same relaxation problem and radiation diffusion problem, along with several other nonlinear problems. However, the linearized [1] methods they consider are only first-order accurate. Concurrent with the present work [13,18] also present results for the radiation diffusion problem, using a different spatial discretization, and compare several methods, including converging the nonlinear terms and "one step" [3, Section 3.2] methods. Refs. [9,13,18] also consider operator-splitting effects, while this study will concentrate only on the effects of linearization.

Ref. [9] includes a modified equation analysis and concluded that one reason the Newton–Krylov (NK) method is accurate is because it is "nonlinearly consistent" (NC), in that the entire residual is evaluated at the same time level and implicit nonlinear terms are converged to a small tolerance. The NC property has been put forward as desirable [2,4,5,9,12,15,17], but the question remains under what conditions is the NC property *necessary* in order to obtain accurate results for nonlinear, multiple time scale problems. To begin to answer this question, this study will compare an NC method with several other linearized treatments.

In terms of accuracy for a given time step, this study will reinforce the idea that generally, an NC method is hard to beat. But further study of linearized methods is still needed for the following reasons:

- Much of the previous work on second-order NC methods has used a first-order linearization (cf. Section 3.3) as a basis of comparison (for example, see [2,9,12,17]).This study will suggest several viable second-order schemes as a basis for future comparisons.
- Many existing codes are based on linearized methods. What benefits will these codes realize by converging the nonlinearities? Can some of these benefits be realized by using a better linearized method, which depending on the implementation, may be an easier code modification?
- For a given accuracy, compare the efficiency of second-order linearized and NC methods.

This study will by no means fully address all of these issues, but is simply a first step and will hopefully motivate future work.

The next section defines the implicit nonlinear treatments. In a general setting, a truncation error analysis is then performed for each treatment. The analysis is applied to a simple relaxation problem and it is shown that each second-order accurate linearized method may be more accurate than NC methods, depending on the degree of nonlinearity in the problem. Numerical results back up the analysis. A nonlinear radiation diffusion problem is then considered, both for smooth conditions, and the Marshak conditions (see, for example [11]) presented in [7–9].

2. The time discretization

This section will describe the time discretization and notation used in this study. Consider a system of ordinary differential equations

$$\mathbf{u}'(t) = \mathbf{r}(\mathbf{u}), \tag{1}$$

where $\mathbf{u}(t)$ is the vector of unknowns. Throughout this study, the notation $(\cdot)'$ indicates differentiation with respect to the time variable, t.

[1] In the present context, "linearized" means that the time integration method is formulated so that it performs a fixed, small number (one or two) of linear solves per time step.

The focus in this study is on cases where the function $\mathbf{r}(\mathbf{u})$ is nonlinear; for example, $\mathbf{r}(\mathbf{u})$ might result from the spatial discretization of a nonlinear system of partial differential equations. This study will concentrate on the θ-method time discretization, given by

$$\frac{\mathbf{u}^{n+1}-\mathbf{u}^n}{\Delta t} = \theta\mathbf{r}(\mathbf{u}^{n+1}) + (1-\theta)\mathbf{r}(\mathbf{u}^n), \tag{2}$$

where $0 \leqslant \theta \leqslant 1$ is a specified parameter. For $\theta = 1/2$, one obtains the well-known Crank–Nicolson (trapezoidal) method, which assuming sufficient regularity in t, is second-order accurate in time for small Δt.

Although time discretizations other than (2) (e.g., implicit Runge–Kutta or multistage methods; see [1]) may be more appropriate for a particular problem, the main focus of this study will be the treatment of the implicit nonlinear term $\mathbf{r}(\mathbf{u}^{n+1})$. Each treatment is the subject of the next section.

3. Treatments of the implicit nonlinear term

This section describes the various treatments of the nonlinear term $\mathbf{r}(\mathbf{u}^{n+1})$ in Eq. (2). It must be emphasized that all of the methods in this study are equivalent whenever $\mathbf{r}(\mathbf{u})$ is linear.

3.1. Nonlinearly consistent

One possibility is at each time step, converge $\mathbf{r}(\mathbf{u}^{n+1})$ to a small tolerance. This approach results in what is often referred to as a nonlinearly consistent (NC) treatment. There are several approaches to converging the nonlinearities, including the Newton–Krylov (NK) method. Using NK to solve (2) has been demonstrated to be accurate and efficient for a wide range of problems [4,5,7,14,15,17].

Although the terminology here may be confusing, it must be stressed that NK is a specific example of an NC method. With sufficiently small convergence criteria, all NC methods for Eq. (2) have identical accuracy and differ only in their robustness and efficiency in converging the nonlinear terms.

3.2. Beam and Warming

A second-order approximation of $\mathbf{r}^{n+1} \equiv \mathbf{r}(\mathbf{u}^{n+1})$ is

$$\mathbf{r}^{n+1} = \mathbf{r}^n + (\partial_{\mathbf{u}}\mathbf{r})^n\delta\mathbf{u} + \mathrm{O}(\delta\mathbf{u}^2), \tag{3}$$

where $\partial_{\mathbf{u}}\mathbf{r}$ is the Jacobian of $\mathbf{r}(\mathbf{u})$ and $\delta\mathbf{u} = \mathbf{u}^{n+1} - \mathbf{u}^n$. If the $\mathrm{O}(\delta\mathbf{u}^2)$-terms are ignored, then Eq. (2) transforms to the following linear system:

$$[1 - \theta\Delta t(\partial_{\mathbf{u}}\mathbf{r})^n]\delta\mathbf{u} = \Delta t\mathbf{r}^n. \tag{4}$$

This method is the two-level version of the Beam and Warming scheme (BW) [3] (see [6, Section 11.3]). Beam and Warming cite [10], where (4) can also be interpreted as a one-stage Rosenbrock (“semi-implicit” Runge–Kutta) method.

The BW method is also equivalent to a single Newton iteration of NK, using a small tolerance on NK’s inner linear solve. It therefore should be emphasized that the BW method requires forming the matrix $\partial_{\mathbf{u}}\mathbf{r}$, or at least accurately estimating its action (cf. Section 6.2). Any approximations to $\partial_{\mathbf{u}}\mathbf{r}$ may significantly decrease accuracy.

3.3. Lagged nonlinearities

For many physical systems, the nonlinear operator $\mathbf{r}(\mathbf{u})$ may be factored as

$$\mathbf{r}(\mathbf{u}) = N(\mathbf{u})\mathbf{u} + \mathbf{b}, \tag{5}$$

where $N(\mathbf{u})$ is a matrix and $\mathbf{b}$ a constant vector. For a linear system, $N = \partial_{\mathbf{u}}\mathbf{r}$. However, for a nonlinear system, N is often much easier to form than $\partial_{\mathbf{u}}\mathbf{r}$.

A common time integration method is then to lag the nonlinearities in $\mathbf{r}^{n+1}$ as

$$\mathbf{r}^{n+1} = N^n\mathbf{u}^{n+1} + \mathbf{b} + \mathrm{O}(\delta\mathbf{u}). \tag{6}$$

This study will refer to the approximation above as the Lagged treatment. Note that unlike the linearization (3), the Lagged treatment is only first-order accurate with respect to $\delta\mathbf{u}$. As a single-stage method, Eq. (2) with the Lagged treatment is

$$[1 - \theta\Delta t N^n]\delta\mathbf{u} = \Delta t\mathbf{r}^n, \tag{7}$$

which should be compared with Eq. (4). This method has often been used as the basis of comparison for NC methods [2,9,12,17] and is the only first-order method (for any θ) in this study. Refs. [7,8] refer to this method as the 'semi-implicit' method.

3.4. Predictor–corrector

Second-order accuracy may also be obtained by using Eq. (7) as a predictor, and then performing another linear solve in a corrector step. Specifically,

$$\frac{\mathbf{u}^* - \mathbf{u}^n}{\Delta t} = N^n\mathbf{u}^* + \mathbf{b}, \tag{8a}$$

$$\frac{\mathbf{u}^{n+1} - \mathbf{u}^n}{\Delta t} = \theta(N^*\mathbf{u}^{n+1} + \mathbf{b}) + (1 - \theta)\mathbf{r}^n. \tag{8b}$$

For $\theta = 1$, this method is simply two Picard iterations for backward-Euler time differencing.

Note that this predictor–corrector method differs from those covered in [1,10], in that it uses the Lagged method for the predictor, as opposed to an explicit method. As a result, the method (8a) and (8b) is A-stable, but requires two linear solves per time step. However, if an existing code is already using the Lagged method, then adding the corrector step may be straightforward.

4. Truncation error analysis

In this section the truncation errors are compared for each of the methods of the previous section. The error analysis will help explain some the numerical results obtained in this study. Many of the results and techniques used in this section are well known and therefore the presentation is brief.

4.1. Nonlinearly consistent methods

For NC methods, expand $\mathbf{u}^{n+1}$ and $\mathbf{r}^{n+1}$ in terms of their time-level-n quantities to obtain

$$\frac{\mathbf{u}^{n+1}-\mathbf{u}}{\Delta t}=\mathbf{u}'+\frac{1}{2}\Delta t\mathbf{u}''+\frac{1}{6}\Delta t^2\mathbf{u}'''+\mathrm{O}(\Delta t^3), \tag{9}$$

$$\mathbf{r}^{n+1}=\mathbf{r}+\Delta t(\partial_{\mathbf{u}}\mathbf{r})\left(\mathbf{u}'+\frac{1}{2}\Delta t\mathbf{u}''\right)+\frac{1}{2}\Delta t^2(\partial_{\mathbf{u}}^2\mathbf{r})(\mathbf{u}')^2+\mathrm{O}(\Delta t^3), \tag{10}$$

where without a superscript, the term is assumed to be evaluated at time level n. Through $\mathrm{O}(\Delta t^2)$, the above expressions assume that $\mathbf{u}'''$ and the tensor $\partial_{\mathbf{u}}^2\mathbf{r}$ exist.

Insert the above expressions into (2) to yield

$$\mathbf{u}'=\mathbf{r}-\frac{1}{2}\Delta t\mathbf{u}''-\frac{1}{6}\Delta t^2\mathbf{u}'''+\theta\Delta t(\partial_{\mathbf{u}}\mathbf{r})\mathbf{u}'+\frac{1}{2}\theta\Delta t^2\left[(\partial_{\mathbf{u}}\mathbf{r})\mathbf{u}''+(\partial_{\mathbf{u}}^2\mathbf{r})(\mathbf{u}')^2\right]+\mathrm{O}(\Delta t^3). \tag{11}$$

To simplify this expression, differentiate it twice to obtain

$$\mathbf{u}''=(\partial_{\mathbf{u}}\mathbf{r})\mathbf{u}'-\frac{1}{2}\Delta t\mathbf{u}'''+\theta\Delta t\left[(\partial_{\mathbf{u}}^2\mathbf{r})(\mathbf{u}')^2+(\partial_{\mathbf{u}}\mathbf{r})\mathbf{u}''\right]+\mathrm{O}(\Delta t^2), \tag{12}$$

and

$$\mathbf{u}'''=(\partial_{\mathbf{u}}\mathbf{r})\mathbf{u}''+(\partial_{\mathbf{u}}^2\mathbf{r})(\mathbf{u}')^2+\mathrm{O}(\Delta t). \tag{13}$$

These last two expressions may be used to eliminate $(\partial_{\mathbf{u}}^2\mathbf{r})(\mathbf{u}')^2$ and $(\partial_{\mathbf{u}}\mathbf{r})\mathbf{u}'$ in Eq. (11), so that

$$\mathbf{u}'-\mathbf{r}=\Delta t\left(\theta-\frac{1}{2}\right)\mathbf{u}''-\frac{1}{6}\Delta t^2(6\theta^2-6\theta+1)\mathbf{u}'''+\mathrm{O}(\Delta t^3). \tag{14}$$

This shows the well-known result that (2) is second-order accurate when $\theta=1/2$ and first-order otherwise.

4.2. Beam and Warming method

Using the same assumptions of solution regularity as for NC methods, the modified equation for the BW method is

$$\mathbf{u}'-\mathbf{r}=\Delta t\left(\theta-\frac{1}{2}\right)\mathbf{u}''-\frac{1}{6}\Delta t^2\left\{(6\theta^2-6\theta+1)\mathbf{u}'''+3\theta\left[\mathbf{u}'''-(\partial_{\mathbf{u}}\mathbf{r})\mathbf{u}''\right]\right\}+\mathrm{O}(\Delta t^3). \tag{15}$$

This method is also second-order when $\theta=1/2$. Whether the $\mathrm{O}(\Delta t^2)$ error term is better behaved in (14) or in (15) is problem dependent; it must be stressed that both derivations used the same assumptions of smoothness. In this study, it will be shown that for some cases, the BW method has lower error, while for others, NC methods have lower error.

4.3. Lagged method

For the Lagged method, the modified equation is

$$\mathbf{u}'-\mathbf{r}=\Delta t\left(\theta N\mathbf{u}'-\frac{1}{2}\mathbf{u}''\right)+\mathrm{O}(\Delta t^2). \tag{16}$$

If one assumes that the tensor $\partial_{\mathbf{u}}N$ exists, then an alternative form for (16) is

$$\mathbf{u}' - \mathbf{r} = \Delta t\left(\theta - \frac{1}{2}\right)\mathbf{u}'' - \Delta t\theta(\partial_{\mathbf{u}}N)\mathbf{u}'\mathbf{u} + \mathrm{O}(\Delta t^2), \tag{17}$$

which may then be directly compared with (14) or (15). Eq. (17) was also derived for a relaxation problem in [9].

On the other hand, the presence of the $\partial_{\mathbf{u}}N$ term does not mean that the Lagged method is less accurate than NC($\theta = 1$) for all problems. It could be that the $\partial_{\mathbf{u}}N$ term in (17) compensates for the other error term. As a simple example, consider a scalar relaxation problem with $r(u) = -cu^2$, where c is a constant. A Lagged linearization is $N(u) = -cu$ and $b = 0$. For $\theta = 1$, it is easy to show that the first-order error drops in (16), so that the method is second-order accurate. For this special case, the Lagged linearization results in a harmonic average over the time step.

4.4. Predictor–corrector method

To derive the truncation error for the PC method requires an explicit expression for $\mathbf{u}^*$. Eq. (8a) may be rearranged to read

$$\mathbf{u}^* = \mathbf{u}^n + \Delta t(N^n\mathbf{u}^* + b). \tag{18}$$

Apply this expression recursively to obtain

$$\mathbf{u}^* = \mathbf{u}^n + \Delta t\mathbf{r}^n + \Delta t^2 N^n\mathbf{r}^n + \mathrm{O}(\Delta t^3). \tag{19}$$

Eq. (8b) then yields

$$\mathbf{u}' - \mathbf{r} = \Delta t\left(\theta - \frac{1}{2}\right)\mathbf{u}'' - \frac{1}{6}\Delta t^2\left[\left(6\theta^2 - 6\theta + 1\right)\mathbf{u}''' + 6\theta(N - \partial_{\mathbf{u}}\mathbf{r})(N - \theta\partial_{\mathbf{u}}\mathbf{r})\mathbf{r}\right] + \mathrm{O}(\Delta t^3). \tag{20}$$

Like the BW and NC methods, the PC method is second-order accurate when $\theta = 1/2$. Through $\mathrm{O}(\Delta t^2)$, the derivation of (20) requires that $\mathbf{u}'''$ and the tensor $\partial_{\mathbf{u}}^2 N$ exist, which are similar regularity assumptions as for the BW and NC methods. Comparing (20) and (14), the last term involving N is the only difference with the error for NC methods. In the next section it will be shown that this additional term may increase or decrease the error, depending on the problem.

5. Relaxation problem

The results so far can now be applied to a simple problem. Consider the following scalar ordinary differential equation:

$$\frac{\mathrm{d}T}{\mathrm{d}t} = \alpha(T)T, \tag{21}$$

where

$$\alpha(T) = \begin{cases} -T^p & \text{for } p \geqslant 0, \\ -1/(T^{-p} + a) & \text{for } p < 0, \end{cases} \tag{22}$$

along with the initial condition $T(t = 0) = 1$. The parameter p is the degree of nonlinearity and the parameter a keeps the solution well behaved as $T \to 0$. This problem was studied in [9], with $p = -3$ and $a = 0.02$, as a test for nonlinear time integrators.

Note that the exact solution satisfies $0 \leqslant T(t) \leqslant 1$. The solution evolves on a time scale given by $-1/\alpha$. This time scale is often referred to as the dynamical time scale, τ_{dyn}. The minimum value of τ_{dyn} depends on p as

$$\tau_{\text{dyn}}^{\min} = \begin{cases} 1 & \text{for } p \geqslant 0, \\ a & \text{for } p < 0. \end{cases} \tag{23}$$

5.1. Error analysis

This section will emphasize that depending on the value of p in Eq. (22), both the BW and PC methods may be more accurate than NC methods. In other words, in terms of the truncation error analysis, none of these methods is the most accurate for all problems. For brevity, the analysis is restricted $\theta = 1/2$, so that each method is second-order accurate. The Lagged method, which in general is only first order, is not analyzed in this section.

To simplify the analysis, assume that a is very small, so that (22) may be approximated as

$$\alpha(T) = -T^p. \tag{24}$$

From Eqs. (14) and (15), with $\theta = 1/2$, the ratio of the leading-order truncation error of BW to that of an NC method is given by

$$\frac{(\text{TE})_{\text{BW}}}{(\text{TE})_{\text{NC}}} = \frac{3(\partial_{\mathbf{u}}\mathbf{r})\mathbf{u}'' - 2\mathbf{u}'''}{\mathbf{u}'''}. \tag{25}$$

With Eqs. (21) and (24), the expression above reduces to

$$\frac{(\text{TE})_{\text{BW}}}{(\text{TE})_{\text{NC}}} = \frac{1-p}{1+2p}. \tag{26}$$

If conditions are such that the expressions for truncation error hold and the leading-order terms dominate, then BW is more accurate than an NC method if

$$\left| \frac{(\text{TE})_{\text{BW}}}{(\text{TE})_{\text{NC}}} \right| < 1. \tag{27}$$

This expression is satisfied if

$$\begin{cases} p < -2, & \text{or} \\ p > 0. \end{cases} \tag{28}$$

Therefore, the additional iterations that a Newton method performs actually decreases the accuracy whenever (28) is satisfied. Note also that BW is third-order accurate for the special case of $p = 1$ and an NC method is third-order when $p = -1/2$. Also, for the trivial case of $p = -1$, the BW and NC methods are exact. The PC method is not exact for $p = -1$, because it relies on the factorization (6), which is not needed in this trivial case.

Similarly, the PC method gives

$$\frac{(\text{TE})_{\text{PC}}}{(\text{TE})_{\text{NC}}} = \frac{1+6p-p^2}{(1+p)(1+2p)}. \tag{29}$$

Therefore,

$$\left|\frac{(\mathrm{TE})_{\mathrm{PC}}}{(\mathrm{TE})_{\mathrm{NC}}}\right| < 1 \quad \text{if} \quad \begin{cases} p < p_L, & \text{or} \\ p_R < p < 0, & \text{or} \\ p > 1, \end{cases} \tag{30}$$

where $p_L = -(9+\sqrt{73})/2 \approx -8.772$ and $p_R = (-9+\sqrt{73})/2 \approx -0.228$. The PC method is third-order accurate whenever $p = 3 \pm \sqrt{10}$.

The main point of this section is that none of the methods in this study are optimal for *all* nonlinear problems. For the very idealized problem studied here, both the BW and PC methods have a wide range of p that give more accurate results than NC methods.

Note that for other equations, and in particular systems, the truncation error ratio will be a function of the solution state; see Eq. (25). The analysis may then be much more involved. The manner in which the results of this section extend to more complicated problems is left for future work.

5.2. Maximum time step

This section shows that the maximum allowable time steps for each method are comparable in magnitude and that they exceed the dynamical timescale of the problem. The maximum time step, $\Delta t^{\max}$, is chosen so that for all $\Delta t \leqslant \Delta t^{\max}$

$$T^n \geqslant T^{n+1} \geqslant 0, \tag{31}$$

a condition satisfied by the exact solution. The first inequality implies absolute stability, while the second ensures positivity. For all of the methods in this study, the maximum time step is determined by the positivity condition. To simplify the presentation, assume that Δt is constant over the simulation.

For an NC method, the maximum allowable time step is determined by the maximum value of $|\alpha(T)|$. Using the fact that $0 \leqslant T^n \leqslant 1$, it is then straightforward to show that there is a T^{n+1} for an NC method that satisfies (31) for $\Delta t \leqslant \Delta t^{\max}_{\mathrm{NC}}$, where

$$\Delta t^{\max}_{\mathrm{NC}} = \begin{cases} \frac{1}{1-\theta} & \text{if } p \geqslant 0, \\ \frac{a}{1-\theta} & \text{if } p < 0. \end{cases} \tag{32}$$

The requirement $a > 0$ is apparent. It should be stressed that even with $\Delta t \leqslant \Delta t^{\max}_{\mathrm{NC}}$, there may be other real-valued solutions to the NC difference equation. The assumption here is that the initial guess for the iterative method is close enough to the desired solution that spurious roots are avoided.

It can be shown that for this problem, $\Delta t^{\max}_{\mathrm{PC}} = \Delta t^{\max}_{\mathrm{Picard}} = \Delta t^{\max}_{\mathrm{NC}}$. However, the BW method has a different maximum time step. To simplify the analysis, the assumption that $0 \leqslant a \leqslant 1$ is made. Then,

$$\Delta t^{\max}_{\mathrm{BW}} = \begin{cases} \infty & \text{if } p \geqslant \theta^{-1} - 1, \\ \frac{1}{1-(p+1)\theta} & \text{if } \theta^{-1} - 1 > p \geqslant 0, \\ \frac{a}{1-\theta} & \text{if } 1 - \theta^{-1} \leqslant p < 0, \\ \frac{-4ap\theta}{[\theta(1+p)-1]^2} & \text{if } p < 1 - \theta^{-1}. \end{cases} \tag{33}$$

Therefore, the maximum time step for BW is less restrictive than the other methods for $\theta^{-1} - 1 \geqslant p \geqslant 0$, but more restrictive for $p < 1 - \theta^{-1}$.

As an example, let $\theta = 1/2$. Then for $p = 3$, BW unconditionally satisfies (31), while $\Delta t^{\max}_{\mathrm{NC}} = 2$. In this case $\tau^{\min}_{\mathrm{dyn}} = 1$ (see Eq. (23)), so for accuracy reasons, one probably should not exceed $\Delta t = 1$ by much anyway. For $p = -3$, one finds $\Delta t^{\max}_{\mathrm{BW}} = 3a/2$, while $\Delta t^{\max}_{\mathrm{NC}} = 2a$. In this case, $\tau^{\min}_{\mathrm{dyn}} = a$.

In summary, all of the methods permit $\Delta t > \tau_{\text{dyn}}^{\min}$. An L-stable integration method (as opposed to Crank–Nicolson) might allow an even larger maximum time step, but one might expect time accuracy to suffer whenever $\Delta t \gg \tau_{\text{dyn}}^{\min}$.

5.3. Numerical results

First consider $p = -3$ and $a = 0.02$, which was studied in [9]. A smaller value of a will also be considered in this section, so that the analysis in Section 5.1 applies. As in [9], the error is computed as

$$\text{Error} = \left.\frac{|T_{\text{numerical}} - T_{\text{exact}}|}{T_{\text{exact}}}\right|_{t=t^*}, \tag{34}$$

where $t^* = 0.36$.

Fig. 1 compares the results for all of the methods with $\theta = 1/2$. In this case, the BW method is the most accurate over the time step range. Note that for $a = 0$ in (22), the analysis of Section 5.1 gives that BW($\theta = 1/2$) is more accurate than NC($\theta = 1/2$). The PC($\theta = 1/2$) method attains second-order accuracy, although typically it must take a time step of one-fourth that of NC or BW to attain the same level of accuracy. Again, this result is roughly consistent with the analysis. Note that in this case $\Delta t_{\text{BW}}^{\max} = 0.03$, while for the other methods, $\Delta t^{\max} = 0.04$. Therefore, for $\Delta t = 0.04$ the BW method exhibits slight negativities in the region where T approaches zero.

There is very little difference between Lagged($\theta = 1$) and Lagged($\theta = 1/2$). That Lagged($\theta = 1/2$) performs so poorly is a good indicator of the nonlinearity of this problem, because all of the methods here are identical for a linear problem.

Fig. 2 plots the observed error ratios for $a = 0.02$ and $a = 0.001$, along with the truncation error estimates from Eqs. (26) and (29) with $p = -3$. For $a = 0.001$, the error was computed at $t^* = 0.32$ (at $t = 0.36$, the exact solution has relaxed to $T = 0$). The maximum time step also decreases to $\Delta t_{\text{BW}}^{\max} = 0.0015$ and $\Delta t^{\max} = 0.002$ for the other methods, so that negative solutions are observed for the larger time steps and times past $t^* = 0.32$. The $a = 0.02$ results for the PC method are not shown on the plot, as they range in

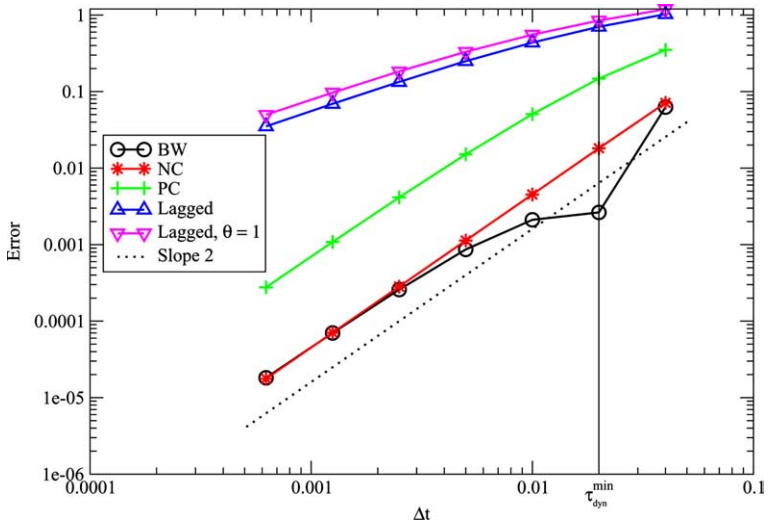

Fig. 1. Errors for relaxation problem, with Eq. (22), $a = 0.02$. Unless noted, all methods used $\theta = 1/2$.

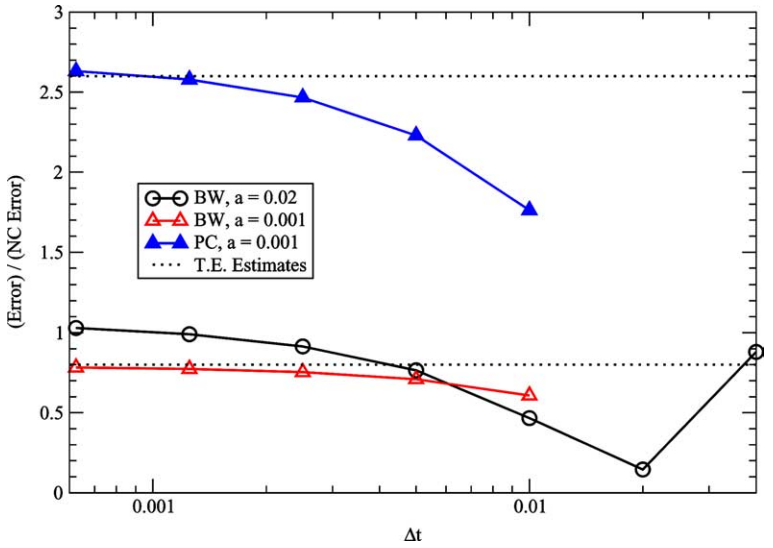

Fig. 2. Ratio of BW and PC errors to NC error for relaxation problem, with Eq. (22), $p = -3$, $\theta = 1/2$. The truncation error (TE) estimates were computed from Eqs. (26) and (29), with the upper line for PC and the lower line for BW.

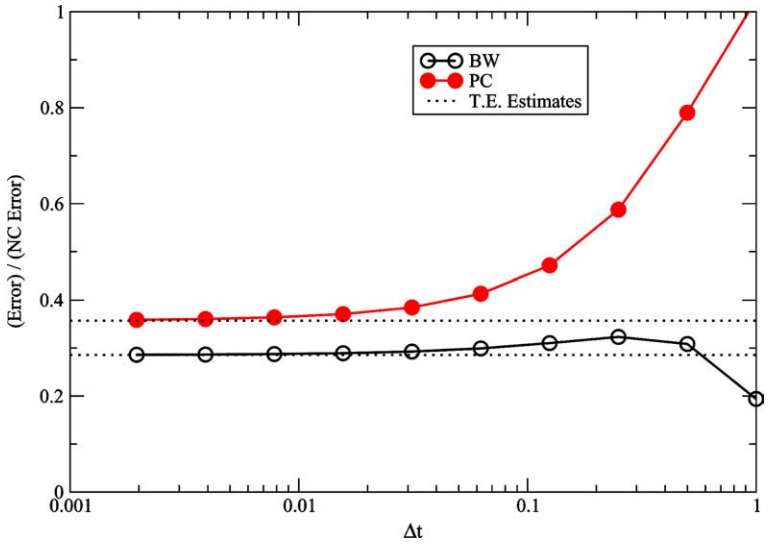

Fig. 3. Ratio of BW and PC errors to NC error for relaxation problem, with Eq. (24), $p = 3$, $\theta = 1/2$. The truncation error (TE) estimates were computed from Eqs. (26) and (29), with the upper line for PC and the lower line for BW.

value from approximately 5 for the large time step, to 16 for the small time step. As expected, the agreement with theory improves with decreasing a.

Finally, Fig. 3 plots the error ratios for the relaxation problem using Eq. (24) and $p = 3$. The error was computed at $t^* = 4$. Again, there is good agreement with the theory.

6. Radiation diffusion problem

A radiation diffusion model is given by the system

$$\partial_t T = \sigma(E - T^4), \tag{35a}$$

$$\partial_t E = \partial_x(D\partial_x E) - \sigma(E - T^4), \tag{35b}$$

where $T(x,t)$ is the temperature and $E(x,t)$ the radiation energy. The spatial domain is $0 \leqslant x \leqslant 1$ and

$$\sigma = T^{-3}, \quad D = (3\sigma + |\partial_x E|/E)^{-1}. \tag{36}$$

The boundary conditions may be written as

$$\frac{1}{4}E(0,t) - \frac{1}{6\sigma}(\partial_x E)(0,t) = 1, \tag{37a}$$

$$\frac{1}{4}E(1,t) + \frac{1}{6\sigma}(\partial_x E)(1,t) = V_R, \tag{37b}$$

where V_R is a constant. If $E(x,t) \equiv 0$, then Eq. (35a) is a special case of the relaxation problem covered in Section 5. For Marshak-wave conditions, the NC($\theta = 1, 1/2$) and Lagged($\theta = 1$) methods were compared in [7,8] and these results are covered in Section 6.6. Results will also be presented in this study for conditions with smoother results, so that spatial truncation-error analysis applies.

6.1. Spatial discretization

The spatial discretization used is a cell-centered finite-volume method, which is described in [7]. Briefly, in semi-discrete form, Eqs. (35a), (35b) are written as

$$\partial_t T_j = \sigma(T_j)(E_j - T_j^4), \tag{38a}$$

$$\partial_t E_j = \frac{(D\partial_x E)_{j+1/2} - (D\partial_x E)_{j-1/2}}{\Delta x} - \sigma(T_j)(E_j - T_j^4), \tag{38b}$$

where

$$D_{j+1/2} = \left[3\sigma(T_{j+1/2}) + |\partial_x E|_{j+1/2}/E_{j+1/2}\right]^{-1}, \tag{39a}$$

$$(\partial_x E)_{j+1/2} = \frac{E_{j+1} - E_j}{\Delta x}, \tag{39b}$$

$$T_{j+1/2} = (T_{j+1} + T_j)/2, \tag{39c}$$

$$E_{j+1/2} = (E_{j+1} + E_j)/2. \tag{39d}$$

Here j is the cell index, with $1 \leqslant j \leqslant N$. For smooth solutions, this discretization is second-order accurate in space.

The discretization of the boundary conditions will be given here for the $x = 0$ boundary; a similar approach is used at $x = 1$. In the first cell ($j = 1$), the value of $(D\partial_x E)_{1/2}$ is needed. Eq. (37a) is discretized as

$$\frac{1}{4}E_{1/2} - \frac{1}{6\sigma(T_1)}\frac{E_1 - E_{1/2}}{\Delta x/2} = 1, \tag{40}$$

where the subscript "1/2" indicates the value at the boundary and the subscript "1" indicates the value in the first cell. This equation gives $E_{1/2}$ in terms of known quantities. Then set

$$(\partial_x E)_{1/2} = \frac{E_1 - E_{1/2}}{\Delta x/2}, \tag{41a}$$

$$T_{1/2} = T_1, \tag{41b}$$

and use Eq. (39a) to compute $D_{1/2}$.

The boundary procedure above is locally only first-order accurate. A second-order method was also implemented, but for the problems in this study, the results did not improve and robustness was slightly degraded.

6.2. Solver details

Each of the time integration methods, given by Eqs. (2), (4) and (7), and each step of (8a) and (8b), may be expressed as a vector function of the unknown $\mathbf{u}^{n+1}$

$$\mathbf{F}_i(\mathbf{u}^{n+1}) = 0, \tag{42}$$

where the subscript 'i' denotes the particular method and the dependence on $\mathbf{u}^n$ and Δt has been suppressed. Note that $\mathbf{F}_{\mathrm{NC}}$ is a nonlinear function, while for the other methods, $\mathbf{F}_i$ is linear. In this study, each method used an iterative method to converge their discrete equation to the same tolerance, as

$$\frac{\|\mathbf{F}_i(\mathbf{u}^{n+1})\|_2}{\|\mathbf{F}_i(\mathbf{u}^n)\|_2} \leqslant 10^{-7}. \tag{43}$$

For example, the NC method converged its nonlinearities using Newton–Krylov (NK), such that the nonlinear residual satisfies (43).

All of the methods used right-preconditioned GMRES for their linear solver. The preconditioner used was effectively the Lagged($\theta = 1$) method, similar to [7], but implemented as a tridiagonal solve. This preconditioner adds a negligible cost to each GMRES iteration.

Each NK linear solve decreased its residual by a factor of $\gamma = 10^{-p}$, where $p \geqslant 1$ (see [7] for more details on γ). For the problems in this study, numerical experiments for various integer-p found that $p = 2$ minimized the CPU time.

Each method evaluates the product of its Jacobian matrix with a vector. The NC and BW methods evaluate $(\partial_{\mathbf{u}}\mathbf{r})\delta\mathbf{u}$, which is approximated using the matrix-free technique [7]:

$$(\partial_{\mathbf{u}}\mathbf{r})\delta\mathbf{u} \approx \frac{\mathbf{r}(\mathbf{u} + \varepsilon\delta\mathbf{u}) - \mathbf{r}(\mathbf{u})}{\varepsilon}. \tag{44}$$

The Lagged and PC methods evaluate $N\delta\mathbf{u}$, which for ease of coding, is computed in a similar way

$$N\mathbf{u} \approx \frac{\tilde{\mathbf{r}}(\mathbf{u} + \varepsilon\delta\mathbf{u}) - \tilde{\mathbf{r}}(\mathbf{u})}{\varepsilon}. \tag{45}$$

where $\tilde{\mathbf{r}} = N\mathbf{u} + \mathbf{b}$. In Eqs. (44) and (45),

$$\varepsilon = \varepsilon_0 \|\mathbf{u}\|_2 / \|\delta\mathbf{u}\|_2. \tag{46}$$

The NC and BW methods set $\varepsilon_0 = 10^{-8}$ (approximately the square root of machine precision). Because $\tilde{\mathbf{r}}(\mathbf{u})$ is linear, with exact arithmetic, the results of Eq. (45) are independent of ε_0. Using values in the range $10^{-8} \leqslant \varepsilon_0 \leqslant 1000$, it was found that to within round-off, the PC and Lagged results do not change. The PC and Lagged results presented in this study used $\varepsilon_0 = 1$.

Note that for efficiency, one should consider implementing the PC and Lagged methods by explicitly forming the N matrix (as the preconditioner does). For many problems, N is much easier to form than the full Jacobian matrix. During a GMRES solve, knowing N and performing the matrix–vector multiply explicitly is often much more efficient than using Eq. (45). For more complicated problems, it also allows the use of many more off-the-shelf preconditioners. However, to keep the code simple and to have comparable linear solver costs, Eq. (45) was used.

6.3. Time step selection

Two different time step choices are used in this study. One approach ramps up the time step to a specified final value, as prescribed in [7]. Let Δt_{final} be the final time step. Then for time step number n, with $n \geqslant 1$, the time step is computed as

$$\Delta t^n = f^n \Delta t_{\text{final}}, \tag{47a}$$

where the first eight values of f^n are given by

$$\{0.1, 0.1, 0.2, 0.2, 0.3, 0.3, 0.4, 0.4\}. \tag{47b}$$

For $n > 8$, set $\Delta t^n = \Delta t_{\text{final}}$.

The second approach used in this study follows that in [8,16]. The idea is to estimate the dynamical timescale, assuming that the solution is dominated by a wave-like behavior. The wave speed is estimated as

$$v^n = \frac{\Delta x}{\Delta t^n} \frac{2\sum_j \left| E_j^{n+1} - E_j^n \right|}{\sum_j \left| E_{j+1}^{n+1} - E_{j-1}^{n+1} \right|}. \tag{48}$$

Then the next time step is computed as

$$\Delta t^{n+1} = \text{CFL}\Delta x / v^n, \tag{49}$$

where CFL is specified and can be thought of as a Courant number. As in [8], the initial time step is set as $\Delta t^1 = 10^{-9}$. When CFL $= 1$, the dominant radiation front will move at approximately one cell per time step, while for CFL $\gg 1$, one would expect that the radiation front would be poorly resolved.

6.4. Accuracy measures

Two different solution norms are considered in this study. In order to measure the performance of a time integration scheme, a common practice is to fix the spatial mesh and measure the rate of convergence to the $\Delta t = 0$ solution [2,7–9,17]. Specifically, on an equally spaced mesh, the norm

$$L_2^{\Delta t=0}(T_r) = \sqrt{\frac{1}{N}\sum_{j=1}^{N}[T_{r,j} - T_{r,j}^{\Delta t=0}]^2} \tag{50}$$

will be used, where N is the number of mesh cells and T_r is the radiation temperature, defined as

$$T_r = E^{1/4}. \tag{51}$$

The value $T_{r,i}$ is the numerical solution at cell-index j, while $T_{r,j}^{\Delta t=0}$ is the value on the same spatial mesh but with (ideally) $\Delta t = 0$. An analytic expression for $T_{r,j}^{\Delta t=0}$ is typically unknown, so a calculation with a sufficiently small Δt is used instead.

This study will also consider the error norm

$$L_2(T_r) = \sqrt{\frac{1}{N}\sum_{j=1}^{N}[T_{r,j} - T_{r,j}^{\text{base}}]^2}, \tag{52}$$

where ideally $T_{r,j}^{\text{base}}$ is the exact solution, projected onto the mesh. Again, typically the exact solution is unknown, so a suitably fine-mesh calculation is used. The number of fine mesh cells is chosen as $N_{\text{fine}} = 2^m N$, where m is an integer with $m \geqslant 1$. In this way, each coarse-mesh cell is made up of a union of fine-mesh cells. The fine-mesh values are projected to the coarse mesh by averaging

$$T_{r,j}^{\text{base}} = \frac{1}{2^m}\left(\sum_{k=1}^{2^m} T_{r,2^m(j-1)+k}^{\text{fine}}\right), \tag{53}$$

where $T_{r,j}^{\text{fine}}$ is the solution in cell-j on a mesh with N_{fine} cells. For a finite-volume method, this projection is exact for the cell-averaged conserved quantities. Because T_r is not a conserved quantity, the projection is only second-order accurate in space, which is within the accuracy of the spatial discretization used in this study.

6.5. Results for smooth conditions

This section presents results for the initial condition

$$E(x,0) = E_L + (E_R - E_L)\frac{1 + \tanh[50(x - 0.25)]}{2}, \tag{54}$$

$$T(x,0) = E(x,0)^{1/4}, \tag{55}$$

with $E_L = 4$ and $E_R = 0.004$. In Eq. (37b), $V_R = 0.001$. The reason for studying this case is that the solution is smoother than the standard Marshak case presented in Section 6.6 and therefore is more appropriate for verifying an error convergence rate derived from a Taylor-series analysis.

Fig. 4 plots a sample solution for this case. Fig. 5 compares results of the various time integration methods at the radiation front. The BW and NC results appear coincident with the base solution, while the front position of the PC method lags slightly. The Lagged method gives only a slight improvement in going from $\theta = 1$ to $\theta = 1/2$, which again, is a good indicator of the degree of nonlinearity in the problem. Note for a linear problem, all of the $\theta = 1/2$ methods are equivalent and give the same results.

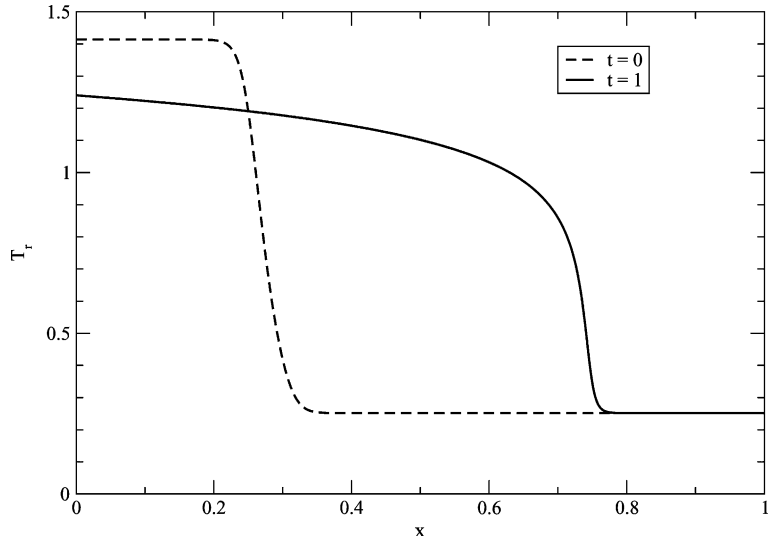

Fig. 4. Smooth radiation diffusion base solution, $t = 1$.

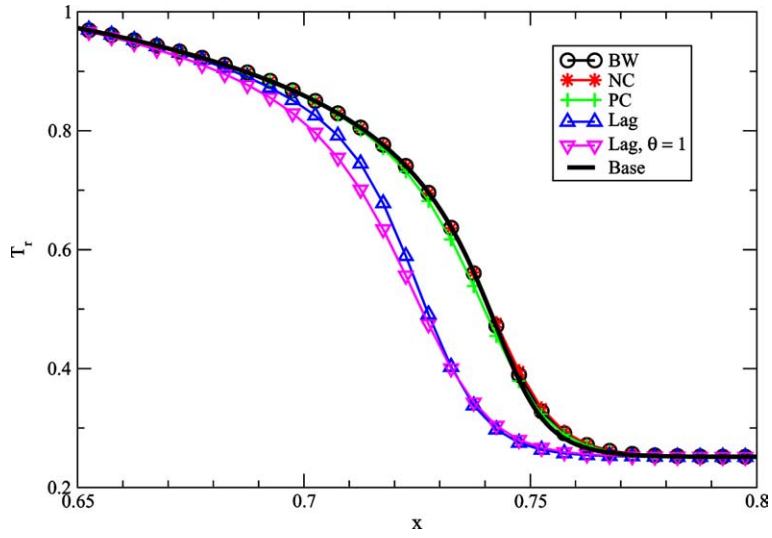

Fig. 5. Sample results for problem of Fig. 4, $t = 1$, 200 mesh cells, CFL = 0.4, and $\theta = 1/2$ (unless noted). The BW and NC results are nearly coincident.

6.5.1. Convergence for a fixed mesh size

The $L_2^{\Delta t=0}$-norm convergence results are shown in Fig. 6. For this problem, CFL = 1 corresponds to an average Δt of 0.012. For a given mesh and problem, it was found that each method has a $\mathrm{CFL}^{\max}$, so that for $\mathrm{CFL} > \mathrm{CFL}^{\max}$, the method fails in some manner. This problem gave a $\mathrm{CFL}_{\mathrm{NC}}^{\max} = 6.1$, beyond which the GMRES failed to converge. However, no backtracking was implemented in the Newton–Krylov implementation. Presumably, backtracking would permit a larger $\mathrm{CFL}^{\max}$. On the other hand, the results here show that the NC solution error is quite large for CFL = 6.1 (although still much smaller than the

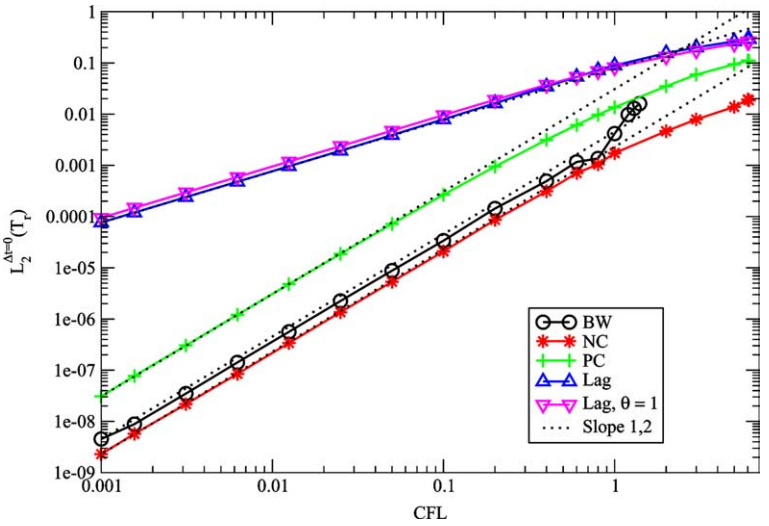

Fig. 6. Problem of Fig. 4, convergence of solution in $L_2^{\Delta t=0}$, $t = 1$, 200 cells, and $\theta = 1/2$ (unless noted). Norms were computed relative to NC(CFL $= 5 \times 10^{-5}$); norms relative to NC(CFL $= 10^{-4}$) appear identical on this scale.

other methods), so the ability to run at an even larger CFL is undesirable if time accuracy is to be maintained.

For BW, $\mathrm{CFL}_{\mathrm{BW}}^{\max} = 1.4$, beyond which at some time, nonphysical, negative solutions were found and the simulation halted. Both the PC and Lagged methods have a $\mathrm{CFL}^{\max} > \mathrm{CFL}_{\mathrm{NC}}^{\max}$ which was not explored.

Aside from the Lagged method, the methods have second-order convergence in $L_2^{\Delta t=0}$. Clearly, the NC method maintains second-order convergence for larger CFL than the other methods, up to approximately CFL $= 1$. The BW method begins to deviate from second-order convergence at CFL $= 0.6$, while the PC method deviates at approximately CFL $= 0.2$. In general, the NC method has the smallest error values, while the PC method has values that are nearly an order-of-magnitude higher than the values of the BW and NC methods.

6.5.2. Convergence for a fixed CFL

To study the effects of the spatial mesh size, the mesh was refined at a fixed CFL $= 0.4$. The errors are plotted in Fig. 7. The BW method is consistently the most accurate. However, given the results in Fig. 6, the lower error of BW here is most likely a fortuitous cancellation of spatial and temporal error components. Both the BW and NC methods attain second-order accuracy at about a mesh size of 300 cells mesh size, while the PC method does not appear to be second-order until nearly 1000 cells. This behavior roughly follows from Fig. 6, where the PC method is not yet second-order at CFL $= 0.4$.

Note that increasing the temporal order-of-accuracy greatly improved the L_2 error; the error is not dominated by spatial errors. For example, Fig. 7 shows that $L_2(T_r)$ for BW($\theta = 1/2$, 200 cells) is over an order-of-magnitude lower than Lagged($\theta = 1/2$, 200 cells).

6.6. Results for Marshak conditions

This section presents results for the conditions used in [7–9]; specifically,

$$E(x,0) = 1 \times 10^{-5}, \quad T(x,0) = E(x,0)^{1/4}, \tag{56}$$

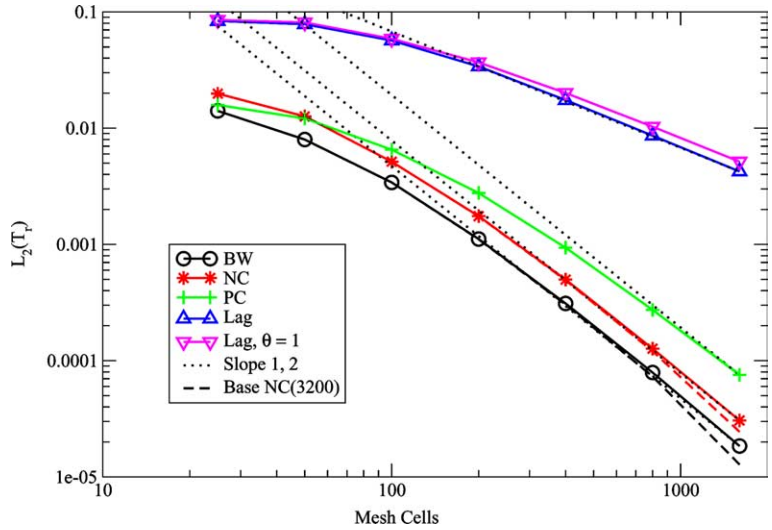

Fig. 7. Problem of Fig. 4, space–time convergence of solution, $t = 1$, $\theta = 1/2$ (unless noted), CFL = 0.4. Error computed using the NC(6400 cells) solution as the base solution. Using NC(3200 cells) as the base gave a slight difference for BW and NC, shown by their adjacent dashed lines.

and $V_R = 0$. The base solution used for comparison is shown in Fig. 8. Note the sharp front, which for coarse meshes, will slow the spatial convergence rate for all of the methods.

Sample solutions are shown in Fig. 9. Just as with all of the cases in this paper, the Lagged($\theta = 1/2$) results are only a small improvement over Lagged($\theta = 1$), showing that treating the nonlinearities accurately over the time step is necessary for this problem.

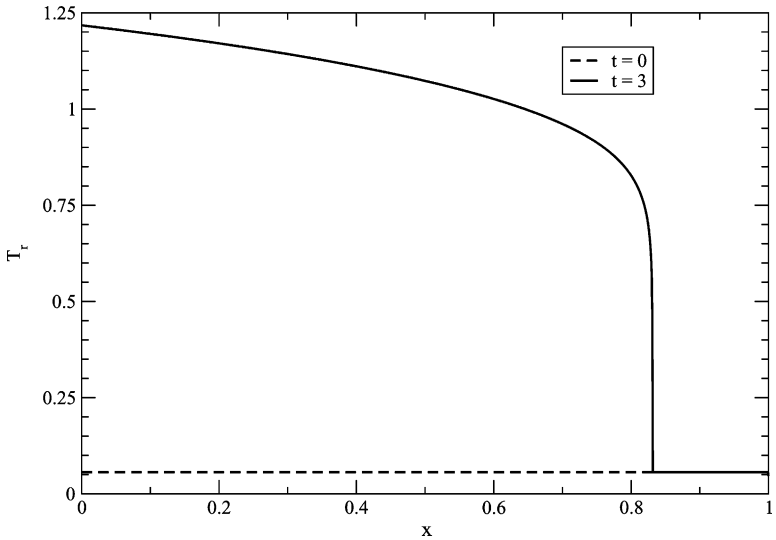

Fig. 8. Marshak radiation diffusion base solution, $t = 3$.

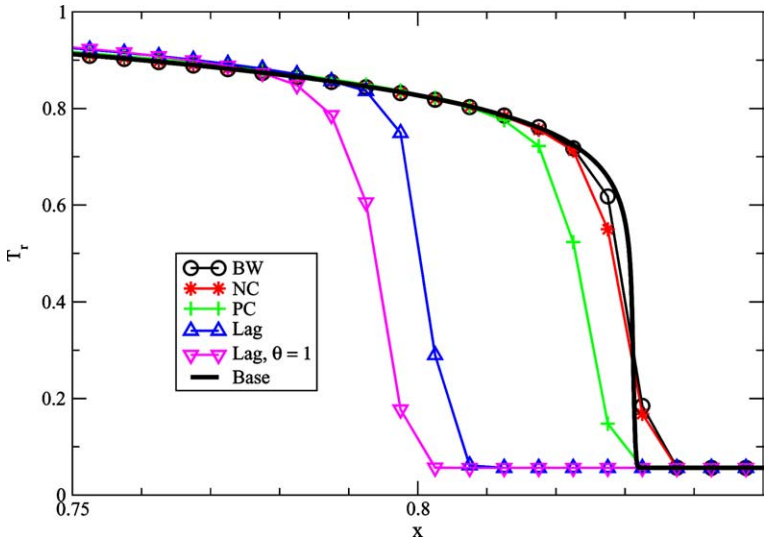

Fig. 9. Sample results for problem of Fig. 8, $t = 3$, 200 mesh cells, CFL = 0.4, and $\theta = 1/2$ (unless noted).

6.6.1. Convergence for a fixed mesh size

To compare with the results of [8], results were generated using the time step ramping given by Eq. (47a), (47b). Convergence in the $L_2^{\Delta t=0}$-norm, at time $t = 1$, for the various methods is shown in Fig. 10. Non-physical, negative solution values were produced by the BW method when $\Delta t_{\text{final}} > 0.0094$ and by the PC method when $\Delta t_{\text{final}} > 0.16$. As in [8], the other methods were run up to $\Delta t_{\text{final}} = 0.2$. Larger Δt_{final} could have been run for the NC and Lagged methods, but this was not the focus here.

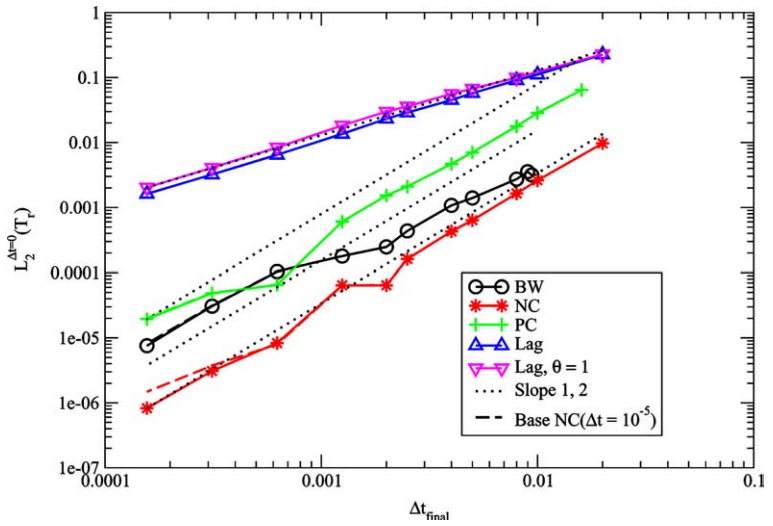

Fig. 10. Problem of Fig. 8, at $t = 1$, convergence of solution using ramping of Eq. (47a), (47b), 200 cells, and $\theta = 1/2$ (unless noted). Norms were computed relative to NC($\Delta t = 10^{-4}$); norms computed with a base of NC($\Delta t = 10^{-5}$) give a slight difference for the BW and NC methods, shown by their adjacent dashed lines.

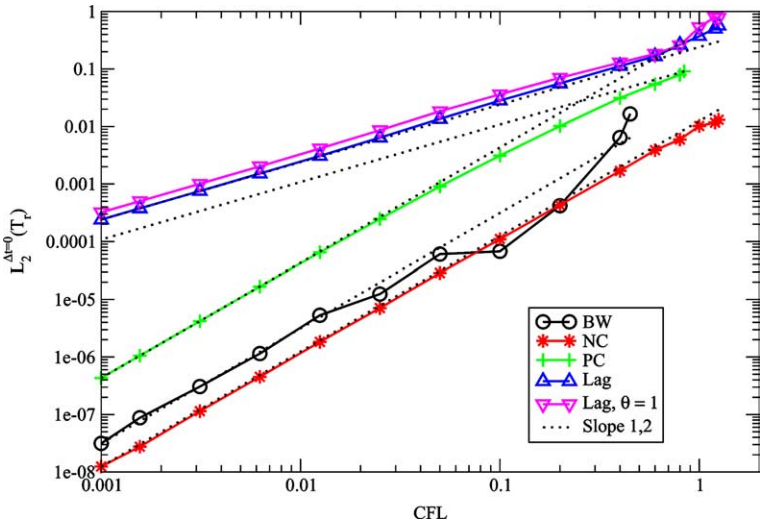

Fig. 11. Problem of Fig. 8, convergence of solution using CFL time step control, $t = 3$, 200 cells and $\theta = 1/2$ (unless noted). Norms were computed relative to NC(CFL $= 5 \times 10^{-5}$); norms relative to NC(CFL $= 10^{-4}$) appear identical on this scale.

Note that the "L_2 Error" in [8] is computed as $\sqrt{N}L_2^{\Delta t=0}$ (where $N = 200$); otherwise, the NC values here and the "NK2" values plotted in [8] compare well. However, unlike [8], the results here clearly show first-order accuracy for the Lagged ("Semi-Implicit") method, over the entire range of Δt.

Ref. [8] plotted results only for $\Delta t_{\text{final}} \geqslant 0.002$. For smaller time steps, all of the second-order methods show erratic behavior in $L_2^{\Delta t=0}$. By using the CFL time step control, the $L_2^{\Delta t=0}$ convergence is better behaved for most of the methods, as shown in Fig. 11. For this problem, CFL $= 1$ corresponds to an average Δt of 0.021. Results for the NC, BW, and PC methods are plotted up to their respective values of $\text{CFL}^{\max}$, beyond which each of these methods generated negative solution values. The maximum CFL values for this problem are $\text{CFL}_{\text{NC}}^{\max} = 1.25$, $\text{CFL}_{\text{BW}}^{\max} = 0.45$, and $\text{CFL}_{\text{PC}}^{\max} = 0.84$. The value of $\text{CFL}^{\max}$ for the Lagged method was not computed and results are given only up to CFL $= 1.25$.

A major limitation to the $\text{CFL}^{\max}$ values, particularly for the NC method, is the use of the Crank–Nicolson time discretization. An L-stable method would probably allow the NC method a much larger time step. On the other hand, Fig. 11 shows that above CFL ≈ 1, the convergence rate for NC is beginning to deviate from second-order accuracy. From CFL $= 1$ to CFL $= 1.2$, the NC convergence rate was measured to be 0.74. If we assume that the higher-order error terms of an L-stable method behave similar to those of Crank–Nicolson, then in terms of accuracy, the benefits of a larger $\text{CFL}^{\max}$ may not be significant.

For CFL < 0.2, the $L_2^{\Delta t=0}$-norms for BW and NC are very similar, while the PC method has values that are over an order-of-magnitude larger. Also, the NC method is able to maintain a second-order convergence rate for larger CFL values. The erratic behavior of the BW method is a concern and should be a focus of future study.

Fig. 12 demonstrates that the $L_2^{\Delta t=0}$-convergence behavior and each method's $\text{CFL}^{\max}$ are only weakly dependent on mesh size. At least for this problem, the CFL time control is a reasonable, mesh-size independent method of selecting the time step. Also interesting is that the erratic behavior of BW(200 cells) is reduced significantly for both the 100 and 400 cell cases.

6.6.2. Convergence for a fixed CFL

Fig. 13 shows the mesh convergence results for a fixed CFL $= 0.4$. The convergence rate attains second-order accuracy only for the PC method, and somewhere between first and second order for the NC and BW

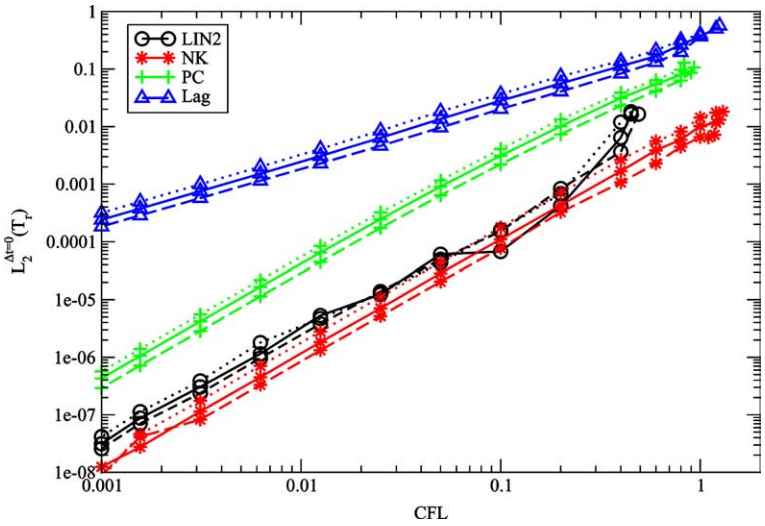

Fig. 12. The $\theta = 1/2$ results of Fig. 11, with results added for 100 cells (dotted line) and 400 cells (dash line).

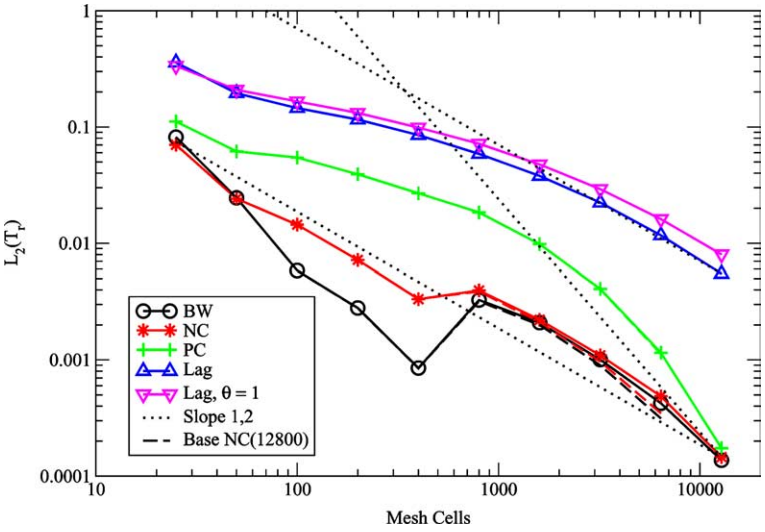

Fig. 13. Problem of Fig. 8, space–time convergence of solution, $t = 3$, $\theta = 1/2$ (unless noted), CFL = 0.4. Error computed using the NC(25,600 cells) solution as the base solution. Using NC(12,800 cells) as the base gave differences for BW and NC, shown by their adjacent dashed lines.

methods. Note that at these mesh sizes plotted, the foot of the radiation front remains unresolved, as shown in Fig. 14. Presumably, because of the lower temporal errors of the NC and BW methods, they are more sensitive to the spatial mesh than the other methods.

Even when there are unresolved spatial scales, the benefits of an accurate time integration method are apparent. Over a wide range of mesh sizes, the BW and NC methods are nearly an order-of-magnitude more accurate than the PC method, while the PC method is at least a factor of four more accurate than the Lagged method.

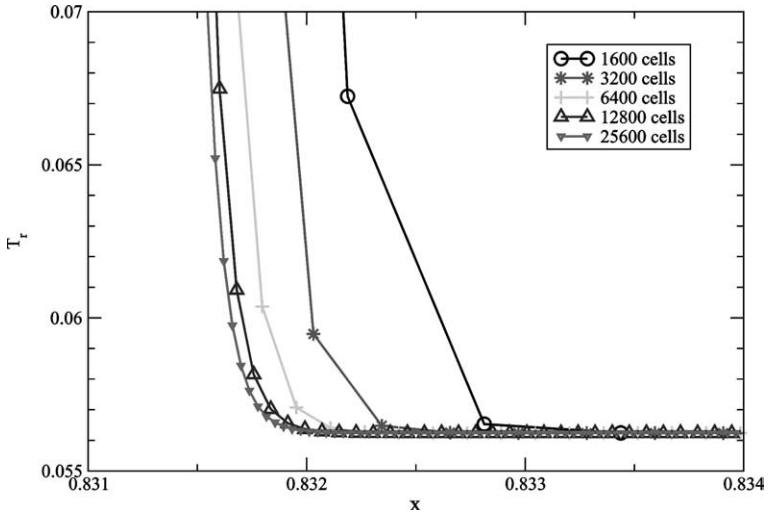

Fig. 14. Problem of Fig. 8, $t = 3$, results at foot of radiation front, NC(CFL = 0.4). Symbols indicate a cell-centered value.

6.6.3. Measure of nonlinear residual

The average and maximum values of the nonlinear residual are plotted in Fig. 15. The nonlinear residual is defined by Eq. (43), using the NC difference equation for the operator $\mathbf{F}(\mathbf{u})$.

The nonlinear residual values are quite large for CFL > 0.1. In fact, for CFL > 0.1, the BW nonlinear residual values are above the PC values, where the BW $L_2^{\Delta t=0}$ values are much lower than the PC values. The nonlinear residual is not necessarily a good measure of the $L_2^{\Delta t=0}$-norm. For the second-order methods, the nonlinear residual here also converges as Δt^2, but this is simply a statement that all second-order methods are within $O(\Delta t^2)$ of each other.

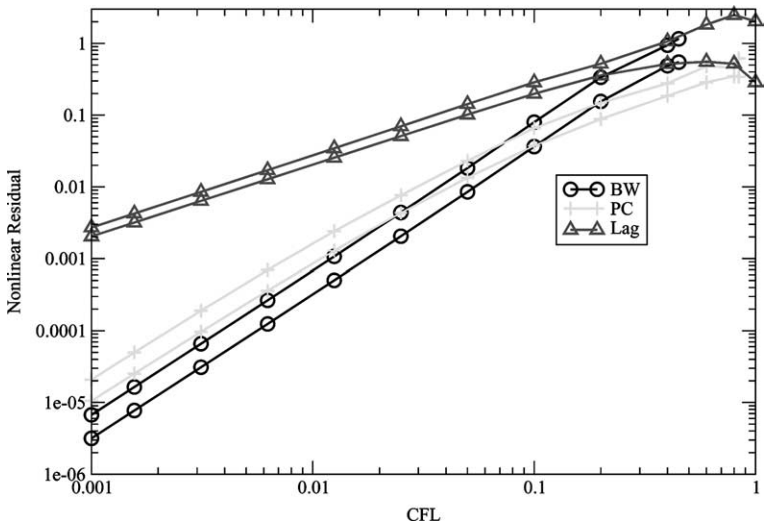

Fig. 15. Same cases as in Fig. 11, but plotting the nonlinear residual for the linearized methods. For each method, the upper and lower lines are, respectively, the maximum and average values over the simulation.

At a particular CFL, the BW method is equivalent to this particular NC method (Newton–Krylov), if the NC method uses BW's maximum nonlinear residual value from Fig. 15 and the same linear tolerance as BW. Some examples of the difference between the NC and BW methods, as NC's nonlinear tolerance in increased, are shown in Fig. 16. Interestingly, the difference is not smoothly varying; instead, it approaches zero very rapidly near the value of BW's maximum nonlinear residual.

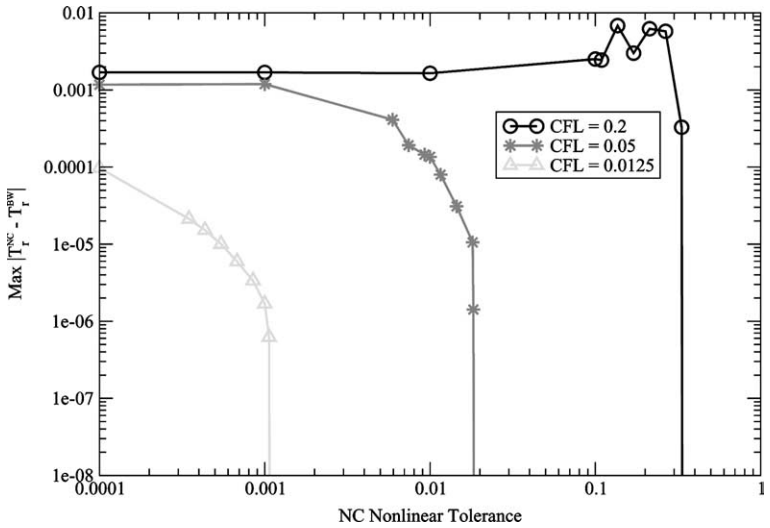

Fig. 16. Problem of Fig. 8, difference between NC and BW methods, $t = 3$, 200 cells, $\theta = 1/2$. For each CFL, the difference is zero at the BW's maximum nonlinear residual shown in Fig. 15.

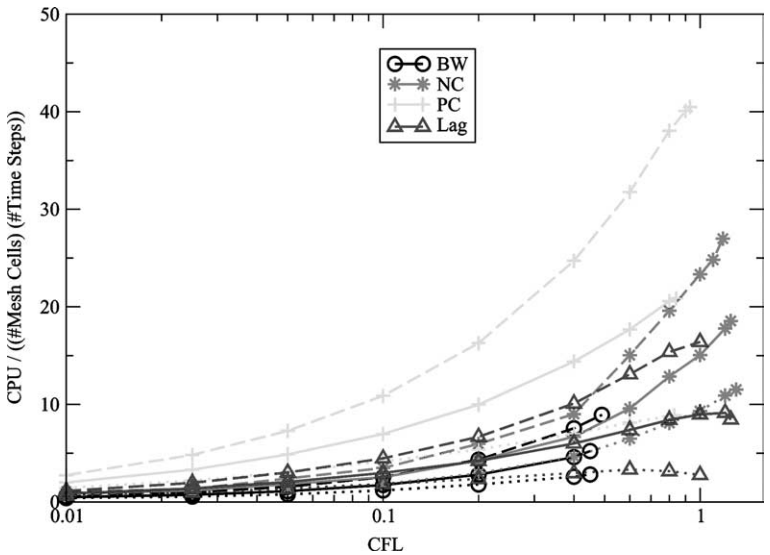

Fig. 17. Problem of Fig. 8, average CPU time per mesh cell per time step, no preconditioner, $t = 3$, $\theta = 1/2$, 100 cells (dotted line), 200 cells (solid line), 400 cells (dashed line). Times normalized by value for preconditioned NC(CFL = 0.001, 200 cells).

6.6.4. Method efficiency

For the Marshak conditions, a brief efficiency study is given here. It must be emphasized that these efficiency trends may not extend to other problems, and in particular, to multiple space dimensions. All calculations were performed on a HP/Compaq ES45 (1 GHz, 8 MB Cache).

Figs. 17 and 18 show the effectiveness of the preconditioner, in terms of CPU time. With the preconditioner, at a given CFL, the results are fairly independent of mesh size. Fig. 18 also shows that the cost per time step of the NC method (implemented as Newton–Krylov) increases quickly for CFL > 0.02. This is

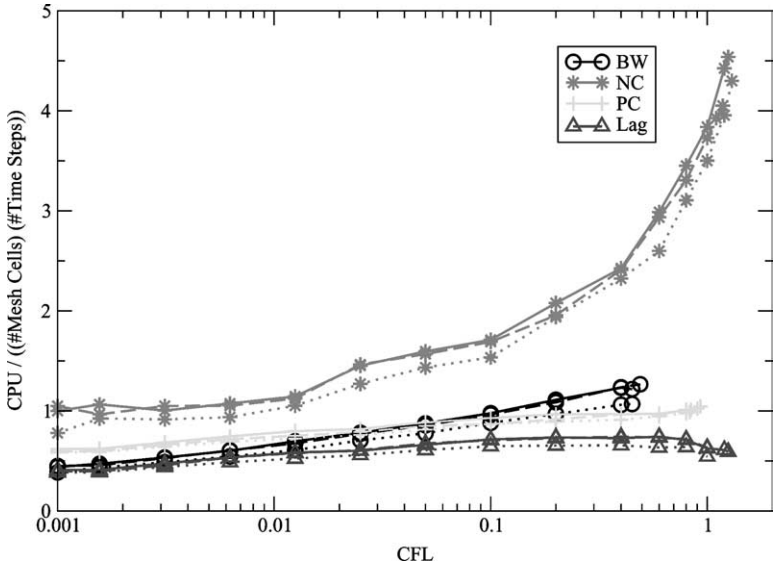

Fig. 18. Same as Fig. 17, with preconditioner.

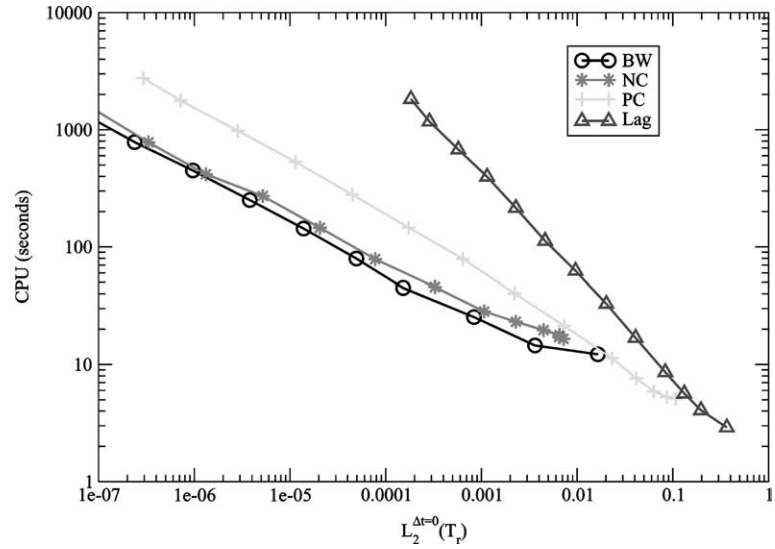

Fig. 19. Problem of Fig. 8, CPU time versus $L_2^{\Delta t=0}(T_r)$, for each method with preconditioner, $t = 3$, 400 cells, $\theta = 1/2$.

because of the increase in linear solves per time step as the time step increases. At CFL = 0.02, NK averaged 3.66 linear solves per time step, while at CFL = 1.25, it averaged 8.00 solves. It follows that the ability of the NC method to use larger time steps may not always translate into a decreased overall simulation time. At their respective $\mathrm{CFL}^{\max}$ and 400 cells, NK($\mathrm{CFL}^{\max} = 1.18$) took 16.55 s versus 12.22 s for BW($\mathrm{CFL}^{\max} = 0.49$).

Fig. 19 shows the CPU time for a specified $L_2^{\Delta t=0}(T_r)$. The BW method has a slight edge over the NC method, while both methods are more efficient than the PC method, particularly for very small $L_2^{\Delta t=0}(T_r)$. The second-order methods are clear winners over the first-order (for nonlinear problems) Lagged method.

Again, it must be stressed that these run times are for a single problem, in 1-D, using an effective preconditioner. The PC and Lagged methods can be implemented in a much more efficient manner; see Section 6.2 for more discussion. The CPU trends for the smooth problem in Section 6.5 are similar as presented here, but more work is needed on other problems and systems in order to proclaim a clear winner.

7. Conclusions

The following observations may be made:

1. For the problems in this study, the linearized methods perform surprisingly well when compared with the nonlinear consistent (NC) method. Both the analysis and numerical results emphasized this point. Although each second-order method has advantages, there was no clear winner, and in particular, no order-of-magnitude differences.
2. Of the second-order methods, the NC method allowed larger time steps and maintained second-order convergence over a broader range of time steps. However, the cost analysis for the radiation diffusion problem showed that NC's cost per time step increases rapidly as the time step is increased past values where the linearized methods operate.
3. All of the methods deviate from their truncation error estimates if the time step becomes too large. In the absence of a good estimate for the dynamical timescale, this may present difficulties for more complicated problems. A time integration scheme with error control should be considered for future work.
4. Except near its maximum time step, the BW method has similar error levels as NC methods. For the relaxation problem, the PC method can be more accurate than either BW or NC methods, but for the radiation diffusion problem, PC has significantly more relative error.
5. The accuracy of the linearized methods is *not* a result of decreasing the nonlinear residual to a small value in a single step. Consequently, there remains hope that there is a linearized method that does not require forming an accurate Jacobian, but is more accurate than the PC method. Another possibility is to run an NC method with a large nonlinear tolerance; however, for systems in conservation form, care must be taken if conservation is to be maintained. To ensure the same level of conservation as the linearized methods, NC methods must decrease their linear tolerance to compensate for an increase in the nonlinear tolerance.

It must be stressed that many of these points need to be explored further, particularly on other systems of equations and other problems. Fortunately, with an existing Newton implementation, one should easily be able to check the benefits (or not) of converging the nonlinearities by trying the BW method. Again, the BW method is the Newton method, restricted to a single Newton iteration.

Acknowledgements

The author thanks Dana Knoll, Michael Pernice, Vince Mousseau, and Jim Morel for invaluable discussions. In addition, the author is grateful for the many corrections and insightful comments made by the anonymous referees.

References

[1] U.M. Ascher, L.R. Petzold, Computer Methods for Ordinary Differential Equations and Differential-Algebraic Equations, SIAM, Philadelphia, PA, 1998.
[2] 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, Journal of Computational Physics 167 (1) (2001) 99–130.
[3] R.M. Beam, R.F. Warming, An implicit finite-difference algorithm for hyperbolic systems in conservation law form, Journal of Computational Physics 22 (1976) 87–110.
[4] L. Chacon, D.C. Barnes, D.A. Knoll, G.H. Miley, An implicit energy-conservative 2D Fokker–Planck algorithm: II. Jacobian-free Newton–Krylov solver, Journal of Computational Physics 157 (2) (2000) 654–682.
[5] L. Chacon, D.A. Knoll, J.M. Finn, An implicit, nonlinear reduced resistive MHD solver, Journal of Computational Physics 178 (1) (2002) 15–36.
[6] C. Hirsch, in: Numerical Computation of Internal and External Flows, vol. 1, Wiley, New York, 1988.
[7] D. Knoll, W. Rider, G. Olson, An efficient nonlinear solution method for non-equilibrium radiation diffusion, Journal of Quantitative Spectroscopy and Radiative Transfer 63 (1999) 15–29.
[8] D. Knoll, W. Rider, G. Olson, Nonlinear convergence, accuracy, and time step control in non-equilibrium radiation diffusion, Journal of Quantitative Spectroscopy and Radiative Transfer 70 (2001) 25–36.
[9] D.A. Knoll, L. Chacon, L.G. Margolin, V.A. Mousseau, On balanced approximations for time integration of multiple time scale systems, Journal of Computational Physics 185 (2) (2003) 583–611.
[10] L. Lapidus, J.H. Seinfeld, Numerical Solution of Ordinary Differential Equations, Academic Press, New York, 1971.
[11] D. Mihalas, B.W. Mihalas, Foundations of Radiation Hydrodynamics, Oxford University Press, Oxford, 1984.
[12] V.A. Mousseau, D.A. Knoll, J.M. Reisner, An implicit nonlinearly-consistent method for the two-dimensional shallow-water equations with Coriolis force, Monthly Weather Review 130 (11) (2002) 2611–2625.
[13] C.C. Ober, J.N. Shadid, Studies on the accuracy of time-integration methods for the radiation–diffusion equations, Journal of Computational Physics (in press).
[14] M. Pernice, M.D. Tocci, A multigrid-preconditioned Newton–Krylov method for the incompressible Navier–Stokes equations, SIAM Journal of Scientific Computing 23 (2) (2001) 398–418.
[15] J. Reisner, V. Mousseau, D. Knoll, Application of the Newton–Krylov method to geophysical flows, Monthly Weather Review 129 (9) (2001) 2404–2415.
[16] W.J. Rider, D.A. Knoll, Time step size selection for radiation diffusion calculations, Journal of Computational Physics 152 (1999) 790–795.
[17] W.J. Rider, D.A. Knoll, G.L. Olson, A multigrid Newton–Krylov method for multimaterial equilibrium radiation diffusion, Journal of Computational Physics 152 (1) (1999) 164–191.
[18] J.N. Shadid, C.C. Ober, R.P. Pawlowski, D.L. Ropp. Studies on solution methods for nonlinear, multiple-time-scale, PDE simulations: examples from diffusion reaction systems, in: Proceedings of the Seventh Copper Mountain Conference on Iterative Methods, Copper Mountain, CO, March 2002.