Electronic Transactions on Numerical Analysis.
Volume 66, pp. 83–107, 2026.

ISSN 1068–9613.
DOI: 10.1553/etna_vol66s83

ETNA
Kent State University and
Johann Radon Institute (RICAM)

AN ADAPTIVE TWO-LEVEL NONLINEAR ELIMINATION PRECONDITIONED SPACE–TIME SOLVER FOR HYPERBOLIC PDEs WITH SHOCKS*

CHANG-WAN LIANG† AND FENG-NAN HWANG†

Abstract. Fully coupled space–time methods for time-dependent partial differential equations (PDEs) have recently gained popularity because of their potential to exploit parallelization in the temporal domain, driven by advances in computing power. These algorithms require solving large, sparse nonlinear systems simultaneously, necessitating a robust and efficient nonlinear solver as a critical component of the entire solution algorithm. This paper studies some nonlinear preconditioned Newton algorithms for the space–time formulation of hyperbolic equations with shocks. In such cases, the classical inexact Newton method with backtracking (INB) suffers from long stagnation due to local strong nonlinearity. Nonlinear preconditioning, such as nonlinear elimination, has been shown to improve the robustness of INB for many types of PDEs; however, it is not effective for hyperbolic PDEs. To address this, we investigate a variant of nonlinear elimination preconditioners for hyperbolic PDEs, considering their characteristics to mitigate the associated difficulties. To further reduce the overhead of the subspace correction step, we consider a two-level version of the nonlinear elimination preconditioned inexact Newton method (NEPIN), where the levels correspond to successive nonlinear elimination steps. The present study focuses on the mathematical properties, convergence behavior, and serial performance of the proposed algorithms. We conducted a comparative performance study of two nonlinear preconditioned iterative algorithms, INB with adaptive nonlinear elimination (INB-ANE) and NEPIN, where nonlinear elimination techniques act as right or left nonlinear preconditioners, respectively, in conjunction with inexact Newton algorithms. We present numerical results using Riemann problems for Burgers' equation and the Buckley–Leverett equation. This indicates that NEPIN outperforms INB-ANE in identifying the correct shock location and introducing less interface pollution after subspace correction, prior to the global update. Additionally, the number of Newton iterations required to converge for NEPIN is almost independent of the time step and mesh size. Finally, a stiffness analysis is conducted to provide further insight into the effectiveness of NEPIN in a hyperbolic space–time formulation.

Key words. parallel-in-time algorithms, space–time methods, hyperbolic PDEs, shock waves, nonlinear elimination preconditioning, inexact Newton method, coarse-grid correction, multilevel algorithms

AMS subject classifications. 65H10, 49M15

1. Introduction.

Time-marching methods are commonly used numerical algorithms for solving systems of nonlinear time-dependent partial differential equations (PDEs) [28]. In particular, hyperbolic PDEs are often used to model physical wave phenomena and engineering problems, with applications in hydrodynamics [1], aerodynamics [18], electrodynamics [3], and traffic flow dynamics [5]. Once spatial discretization is obtained through numerical schemes such as finite differences, finite elements, or finite volumes, a time-marching algorithm advances the solution in time. Two classes of parallel computing techniques aim to accelerate the solution process. The first parallelization approach is used within each time step, while the sequence between time steps remains purely sequential. Although this approach is easy to implement in parallel, its sequential nature can become a bottleneck in achieving significant speedup in parallel performance. Conversely, several parallel-in-time methods have been proposed, representing active research topics in scientific computing. Parareal is a representative method that uses coarse-time integration in serial for predictions and fine-time

*Received August 15, 2025. Accepted July 17, 2026. Published online on September 7, 2026. Recommended by Eran Treister.

This work was partly supported by the National Science and Technology Council (NSTC), Taiwan, under grant number NSTC 112-2115-M-008-007-MY2. An early version of this work appeared in the Proceedings of the 28th International Conference on Domain Decomposition Methods in 2024 [21]. In the present paper we provide a comprehensive treatment with expanded results and in-depth analysis.

†Department of Mathematics, National Central University, Jhongli District 320317, Taoyuan City, Taiwan (C.-W. Liang: treewithout@gmail.com. F.-N. Hwang: hwangf@math.ncu.edu.tw). Corresponding author: F.-N. Hwang.

ETNA
Kent State University and
Johann Radon Institute (RICAM)

integration in parallel for correction [6, 11]. The multilevel version of Parareal is known as the multigrid reduction-in-time (MGRIT) method [14]. However, Parareal-type methods may not perform well for problems involving strong advection or hyperbolic PDEs [11, 14].

Alternatively, a fully coupled space–time discretization method is formulated differently from approaches that use only spatial-domain parallelization. In the space–time approach, the temporal domain is treated as an additional spatial coordinate, and the system of PDEs is discretized on a so-called space–time coordinate domain. Its notable feature is that the numerical solution can be obtained in the entire space–time domain. It can considerably reduce the sequential nature compared to conventional time-marching methods. Benefiting from the high degree of parallelization, the space–time method has attracted more attention in recent years and has been successfully used in many applications, such as PDE-constrained optimization problems [2, 29], inverse problems [7, 8], the problem of transport in a porous medium [13], the fractional Fokker–Planck equation [34], and the problem of high-Rayleigh-number thermal convective flow [31]. Li and Cai [20] also developed an optimal convergence theory for the two-level space–time additive Schwarz method in the space–time formulation of parabolic PDE problems, demonstrating that the method's convergence is independent of the number of subdomains and the grid size.

As a consequence of the fully coupled space–time discretization, the resulting algebraic systems are typically very large, sparse, and highly nonlinear. Therefore, developing robust nonlinear solvers becomes essential. This work aims to develop an efficient and robust solution algorithm for solving hyperbolic PDEs that exhibit solution discontinuities and highly localized nonlinear behavior. The inexact Newton with backtracking (INB) method is a well-known approach for solving nonlinear equations; however, it exhibits slow convergence in the presence of shocks of interest. In this work, we consider both a right nonlinear elimination (NE) preconditioner used in conjunction with INB, called INB-ANE [17], and a left-preconditioned inexact Newton method (NEPIN) [22], aiming to enhance the robustness of INB as a solution algorithm for hyperbolic PDEs in the space–time formulation. However, as illustrated in Section 3, the numerical examples show that the nonlinear elimination, initially designed for elliptic or mixed parabolic–elliptic types of PDEs, does not perform as well, particularly when dealing with shocks in the space–time domain. Identifying suitable subdomains is crucial because they affect the local solution within the global Newton iteration. Both physics-based approaches [16, 17, 27] and algebraic-based approaches using component-wise residuals and overall residual norm information [23] are employed across various application problems, including multiphase flow [23, 33], blood flow or arterial wall problems [12, 25], the Boltzmann equation [15], radiation diffusion [4], and full potential transonic flow problems [16, 17].

The numerical solution of hyperbolic PDEs poses several challenges in effectively identifying problematic components. Firstly, a small nonlinear residual norm does not necessarily imply a small error. Consequently, a residual-norm-based indicator may provide insufficient information to define subspace problems, making it difficult to correctly identify regions with strong local nonlinearity. Secondly, the areas corresponding to problematic components may be isolated or may not form a connected domain. To address these issues, we propose two modifications specifically designed for shock-dominated hyperbolic PDEs. By incorporating these ideas into INB-ANE or NEPIN, we obtain variants of these methods that, in our numerical experiments, show improved robustness and efficiency relative to INB. To reduce the overhead of subspace correction, we investigated a two-level version of NEPIN, where the two levels correspond to successive elimination steps. Our numerical results indicate that this approach can reduce computational overhead in the tested cases. A related space–time Newton method with nested iteration was explored in [19], where strong mesh-independent behavior was demonstrated for several hyperbolic systems. The present work instead investigates nonlinear

ETNA
Kent State University and
Johann Radon Institute (RICAM)

elimination as an alternative and complementary strategy for improving robustness within a grid sequencing approach.

Although nonlinear preconditioning has been numerically demonstrated to be a valuable technique for enhancing the robustness and efficiency of Newton-type methods, a comprehensive understanding of this technique remains incomplete. Beyond algorithmic development, explaining why nonlinear elimination can improve convergence behavior remains an important open question. Recently, Luo et al. [24] introduced a new metric, the stiffness ratio, motivated by the theory of stiff ordinary differential equations (ODEs), to quantify the unbalanced nature of nonlinear systems. Their approach interprets Newton iterations through a dynamical systems viewpoint by relating Newton's method to stiff ODEs. Based on this perspective, we further perform a stiffness analysis to gain qualitative insight into the behavior of NEPIN and INB for the space–time hyperbolic formulation.

Building upon our earlier conference paper [21], which first introduced NEPIN for hyperbolic PDEs, this journal version extends the work by: (i) developing the adaptive two-level NEPIN method, (ii) presenting a detailed comparative study with INB-ANE, and (iii) providing a stiffness-based analysis that offers new insights into solver performance. The proposed NEPIN method is particularly motivated by shock-dominated hyperbolic problems, where strong nonlinear behavior is often localized near shocks while information propagates primarily along characteristic directions. This directional and localized propagation structure differs from the mechanisms commonly encountered in elliptic or parabolic problems and motivates the use of localized nonlinear elimination before the global Newton correction. Although the proposed method is designed for space–time parallelism, the present study focuses on its mathematical properties, convergence behavior, and computational efficiency in serial execution. A detailed parallel implementation and scalability analysis will be addressed in future work.

The remainder of this paper is organized as follows. Section 2 introduces the mathematical models, their discretization, and the test problems considered in this work. In Section 3, we present the nonlinear elimination preconditioning techniques, discuss the challenges arising in hyperbolic PDEs, and introduce the proposed two-level NEPIN method. Numerical results and stiffness analysis are presented in Section 4. Finally, conclusions are given in Section 5.

2. Mathematical models, their discretization, and test problems.

2.1. Burgers' equation and its fully coupled space–time formulation. To explain the proposed method more clearly, let us consider a model problem: the one-dimensional (1D) Burgers' equation in conservation form:

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

with some proper initial and boundary conditions, if needed. Let

$$U = [U_1, U_2, \ldots, U_n]^T$$

denote the global solution vector containing all approximate nodal values in the space–time domain, where $n = N_x \times N_t$. Here, N_x and N_t denote the numbers of spatial and temporal grid points, respectively. Each component U_q corresponds to the approximate solution $u_i^j \approx u(x_i, t_j)$ at the grid point (x_i, t_j), with

$$x_i = x_0 + i\,\Delta x, \quad t_j = t_0 + j\,\Delta t, \quad q = jN_x + i,$$

for $1 \le i \le N_x$ and $0 \le j \le N_t - 1$. The index $j = 0$ corresponds to the initial time level, while indices $j \ge 1$ correspond to unknown space–time nodes. We use the local Lax–Friedrichs (LLF) flux [14] for spatial discretization and the forward Euler method for temporal

ETNA
Kent State University and
Johann Radon Institute (RICAM)

discretization to derive a large, sparse, nonlinear system of equations,

$$F(U) = 0,$$

where $F = [F_1, F_2, \ldots, F_n]^T$, and each component $F_q \equiv f_i^j$ corresponds to the discrete residual at the grid point (x_i, t_j), which is defined as

$$\begin{aligned} f_i^j = u_i^j &- \left(u_{i-1}^{j-1} + |u_i^{j-1}| + |u_{i-1}^{j-1}|\right) \frac{\Delta t}{4\Delta x}\, u_{i-1}^{j-1} \\ &- \left[1 - \left(|u_{i+1}^{j-1}| + 2|u_i^{j-1}| + |u_{i-1}^{j-1}|\right) \frac{\Delta t}{4\Delta x}\right] u_i^{j-1} \\ &+ \left(u_{i+1}^{j-1} - |u_{i+1}^{j-1}| - |u_i^{j-1}|\right) \frac{\Delta t}{4\Delta x}\, u_{i+1}^{j-1}, \end{aligned}$$

for $1 \le i \le N_x$ and $1 \le j \le N_t - 1$. At the initial time level $j = 0$, the residual enforces the prescribed initial condition:

$$f_i^0 = u_i^0 - u_0(x_i), \quad 1 \le i \le N_x.$$

At inflow boundaries, prescribed values (e.g., u_0^j) are substituted directly into the residual expressions, while outflow boundaries (e.g., $u_{N_x+1}^j$) are handled through the numerical flux. This treatment modifies only a few boundary components while preserving the overall sparsity and block lower–triangular structure of the global nonlinear system. The Jacobian matrix of the global nonlinear system $F(U) = 0$ has a block lower-triangular structure because each residual f_i^j depends only on the current solution u_i^j and the previous time level u^{j-1}. The diagonal blocks correspond to the identity matrix (derivatives with respect to u^j), while the first subdiagonal blocks contain the spatial Jacobians $-J_j = -\partial F(u^{j-1})/\partial u^{j-1}$. Accordingly, the global Jacobian can be written as

$$J = \begin{bmatrix} I & 0 & 0 & \cdots & 0 \\ -J_1 & I & 0 & \cdots & 0 \\ 0 & -J_2 & I & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & -J_{N_t-1} & I \end{bmatrix},$$

where each J_j is an $N_x \times N_x$ tridiagonal matrix given by

$$J_j = \begin{bmatrix} \beta_1 & \gamma_1 & 0 & \cdots & 0 \\ \alpha_2 & \beta_2 & \gamma_2 & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \alpha_{N_x-1} & \beta_{N_x-1} & \gamma_{N_x-1} \\ 0 & \cdots & 0 & \alpha_{N_x} & \beta_{N_x} \end{bmatrix}.$$

ETNA
Kent State University and
Johann Radon Institute (RICAM)

The coefficients are obtained by differentiating the discrete residual f_i^j with respect to u_{i-1}^{j-1}, u_i^{j-1}, and u_{i+1}^{j-1}:

$$\alpha_i = \frac{\partial f_i^j}{\partial u_{i-1}^{j-1}} = \left(-2u_{i-1}^{j-1} - |u_i^{j-1}| - 2|u_{i-1}^{j-1}| + \frac{u_{i-1}^{j-1}}{|u_{i-1}^{j-1}|}u_i^{j-1}\right)\frac{\Delta t}{4\Delta x},$$

$$\beta_i = \frac{\partial f_i^j}{\partial u_i^{j-1}} = \left(-u_{i-1}^{j-1}\frac{u_i^{j-1}}{|u_i^{j-1}|} + |u_{i+1}^{j-1}| + 4|u_i^{j-1}| + |u_{i-1}^{j-1}| - \frac{u_i^{j-1}}{|u_i^{j-1}|}u_{i+1}^{j-1}\right)\frac{\Delta t}{4\Delta x} - 1,$$

$$\gamma_i = \frac{\partial f_i^j}{\partial u_{i+1}^{j-1}} = \left(u_i^{j-1}\frac{u_{i+1}^{j-1}}{|u_{i+1}^{j-1}|} + 2u_{i+1}^{j-1} - 2|u_{i+1}^{j-1}| - |u_i^{j-1}|\right)\frac{\Delta t}{4\Delta x},$$

where, for simplicity, the dependence on the time index j is omitted. At points where $u = 0$, the factors $u/|u|$ are interpreted through a generalized derivative of $|u|$; in our implementation, we take this value to be zero.

The Jacobian entries are obtained by differentiating the discrete LLF residual within each smooth sign configuration of the solution. Since the LLF residual is piecewise smooth due to the presence of absolute values, the resulting Jacobian is interpreted as a generalized (piecewise) linearization.

2.2. Illustrative test problems. We consider three test problems for Burgers' equations: the single-shock, rarefaction-wave, and double-shock problems [10]. This study examines the performance of INB, as well as a family of nonlinear elimination preconditioned inexact Newton algorithms.

- Test problem 1: A single-shock problem defined on $[0, 1] \times [0, 1]$ with the initial and boundary conditions
$$u(x,t) = \begin{cases} 0.5 & \text{if } t = 0, \\ 1 & \text{if } x = 0. \end{cases}$$
- Test problem 2: A rarefaction-wave problem defined on $[-1, 1.5] \times [0, 1]$ with the initial and boundary conditions
$$u(x,t) = \begin{cases} -0.5 & \text{if } t = 0,\ x \le 0, \\ 1 & \text{if } t = 0,\ x > 0. \end{cases}$$
- Test problem 3: A double-shock problem defined on $[0, 2] \times [0, 1]$ with the initial and boundary conditions
$$u(x,t) = \begin{cases} 1.5 & \text{if } t = 0,\ x \le 0.5, \\ 0.5 & \text{if } t = 0,\ x > 0.5, \\ 2.5 & \text{if } x = 0. \end{cases}$$

Figures 2.1 and 2.2 show the family of characteristics and solution plots for each case, respectively. As shown in Figure 2.2, the solution is constant along the characteristic curves. The initial waveform advances along the direction of the characteristic line from the beginning to the end. If two characteristic lines with different slopes intersect, the shock occurs at the point of intersection. For the single-shock problem (see the left panel of Figure 2.2), we observe that the shock moves from left to right as time passes because of the differences in wave speed. For the rarefaction problem (see the middle panel of Figure 2.2), two waves travel in opposite directions, and any two characteristics never intersect. Therefore, the solution within the region where no characteristics pass satisfies the so-called entropy condition, such

ETNA
Kent State University and
Johann Radon Institute (RICAM)

that the solution increases linearly. For the double-shock problem (as shown in the right panel of Figure 2.1), the domain initially contains three regions with distinct characteristic slopes. As time evolves, two shocks with different speeds develop and eventually merge into a single shock front.

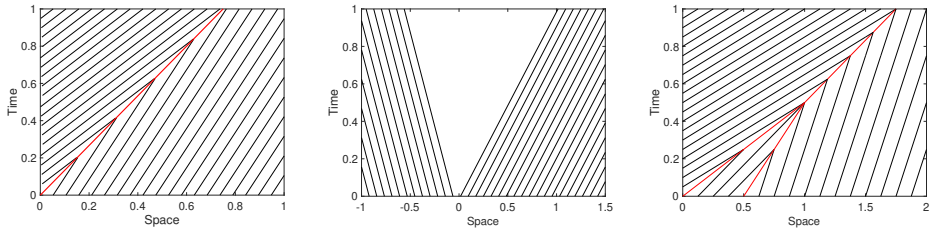

FIG. 2.1. *The characteristic (black lines) and shock (red lines) of the single-shock (left), rarefaction-wave (middle), and double-shock (right) problems.*

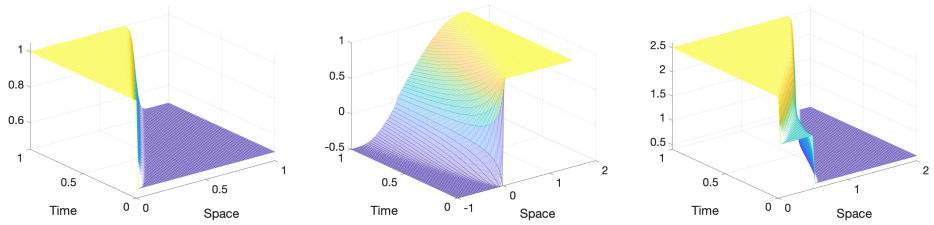

FIG. 2.2. *Numerical solution plots for the single-shock (left), the rarefaction-wave (middle), and the double-shock (right) problems on the grid of 64 × 128.*

2.3. Performance of INB for model test problems. We first investigate how the classical INB method performs for these test problems. Recall that INB generates the new approximate solution

$$U^{(k+1)} = U^{(k)} + \lambda^{(k)} d^{(k)},$$

where $d^{(k)}$ is the inexact Newton direction found by approximately solving the Jacobian system. In this work, the step length $\lambda^{(k)}$ is determined using a quadratic backtracking line search applied to the merit function $\phi(U) = \frac{1}{2}\|F(U)\|^2$, ensuring a sufficient decrease in $\phi(U)$ [9]. As shown in Table 2.1, the number of Newton iterations increases as the grid is refined for the single- and double-shock problems. A double-shock problem is even more challenging than a single-shock problem; more iteration counts are needed for convergence. However, if no shock is present, the number of Newton iterations is almost independent of the grid size, as in the rarefaction-wave problem.

The left panel of Figure 2.3 shows a typical convergence behavior of the inexact Newton methods: the residual norm stagnates for several iterations without progress and then demonstrates quadratic convergence as the intermediate solution enters the convergence region. To understand what happens during the stagnation period of Newton's iteration, we plot the evolution of the algebraic error, which is the difference between the intermediate and final converged solutions for different iterations at $t = 1.0$ (see the right panel of Figure 2.3).

ETNA
Kent State University and
Johann Radon Institute (RICAM)

TABLE 2.1
A summary of the number of Newton iterations using INB for three test problems. A zero initial vector is used for this numerical experiment. The convergence criterion for INB is the same as that used in the numerical results section and is based on a prescribed tolerance for the l^2-norm of the nonlinear residual.

$N_x \times N_t$	16×32	32×64	64×128	128×256
Single shock	11	17	28	59
Rarefaction wave	6	7	8	9
Double shocks	19	33	54	210

The maximum pointwise error, which represents the numerical intermediate shock location, gradually moves to its final position. Once INB correctly captures the location of the shock, the fast convergence of INB is restored.

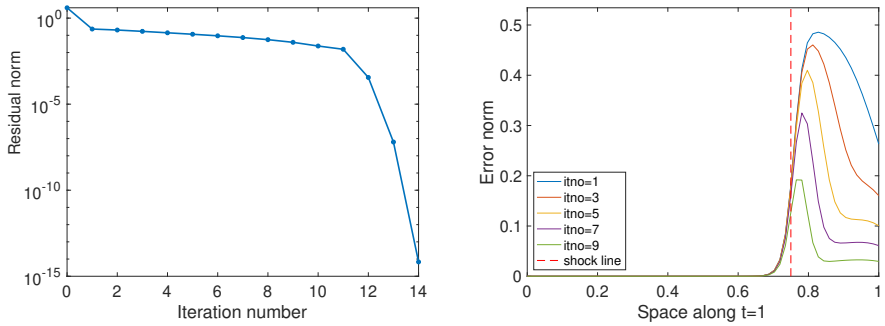

FIG. 2.3. *Single-shock problem. The history of the nonlinear residual norm for the INB (left) and the pointwise error between the intermediate and final space–time solutions at $t = 1.0$ (right).*

These examples illustrate the limitations of the INB method in addressing problems with shocks that occur. To overcome this issue, we introduced a nonlinear elimination preconditioning technique to enhance the robustness and efficiency of the INB method, which will be discussed in the next section when solving the given problem.

3. Nonlinear elimination preconditioning technique. We begin by introducing some notation used for both left and right preconditioners based on nonlinear elimination. Let $S = \{1, 2, \ldots, n\}$ be an index set that corresponds to the unknowns $U_1, U_2, \ldots, U_n$ and the nonlinear residuals $F_1, F_2, \ldots, F_n$. Then, we assume that the solution components can be classified into "good" and "bad" components using the ordered subsets of S, denoted by S_g and S_b as

$$S_g = \{i_1, i_2, \ldots, i_{n_g}\} \quad \text{and} \quad S_b = \{j_1, j_2, \ldots, j_{n_b}\},$$

such that $S_b \cup S_g = S$ and $S_b \cap S_g = \emptyset$ with $n_g + n_b = n$. Then we can define the restriction operator in matrix form for the good and bad components, respectively, as $R_g \in \mathbb{R}^{n_g \times n}$ and $R_b \in \mathbb{R}^{n_b \times n}$ by

$$(R_g)_{m,l} = \begin{cases} 1, & \text{if } l = i_m, \\ 0, & \text{otherwise,} \end{cases} \qquad \text{for } m = 1, 2, \ldots, n_g \text{ and } l = 1, 2, \ldots, n,$$

and

$$(R_b)_{m,l} = \begin{cases} 1, & \text{if } l = j_m \\ 0, & \text{otherwise,} \end{cases} \qquad \text{for } m = 1, 2, \ldots, n_b \text{ and } l = 1, 2, \ldots, n,$$

ETNA
Kent State University and
Johann Radon Institute (RICAM)

which extract the good components and bad components, respectively. Now, we introduce two nonlinear subspace functions for good and bad components as $F_g : \mathbb{R}^n \to \mathbb{R}^{n_g}$ and $F_g = R_g F$, as well as $F_b : \mathbb{R}^n \to \mathbb{R}^{n_b}$ and $F_b = R_b F$. In addition, we have $U = [U_b,\ U_g]^T \in \mathbb{R}^n$, where $U_b = R_b U$ and $U_g = R_g U$. Furthermore, $T_b(U) :\ \mathbb{R}^n \to \mathbb{R}^{n_b}$ is denoted by the solution to the nonlinear subspace problem.

3.1. INB with adaptive nonlinear elimination (INB-ANE). As shown in Algorithm 1, INB-ANE consists of three key steps: subspace correction, global update, and determination of a new partition. From the implementational point of view, in addition to the global Jacobian solver (step 8) and backtracking line search routine (step 9) for determining the step size (also needed for classical INB), the user of INB-ANE needs to provide the mechanism to determine the bad component set S_b (step 11) and arrange the solution and function components accordingly in the split form:

$$F(U_b, U_g) = \begin{bmatrix} F_b(U_b, U_g) \\ F_g(U_b, U_g) \end{bmatrix} = 0. \tag{3.1}$$

Then, we define the nonlinear subspace problem

$$F_b(\hat{U}_b, U_g) = 0,$$

where $\hat{U}_b$ is determined by INB, provided that U_g is available. For the nonlinear elimination approach, the partitioning strategies available in the literature include fixed partitioning and dynamic partitioning. Fixed partitioning can be based on geometric domain decomposition [16] or the nonlinearity degree of the equations [32]. Dynamic partitioning can be further subdivided into methods based on physical characteristics, such as the Mach number for compressible flows [17], and methods based on field equation residuals [30].

Algorithm 1 INB with adaptive nonlinear elimination (INB-ANE).

Input: Initial guess $U^{(0)}$
Output: Solution U^* of system $F(U) = 0$

1: Initialize the partition: $S_b^{(0)} = \emptyset$ and $S_g^{(0)} = S$, and $k = 0$
2: Evaluate $F(U^{(0)})$ and $\|F(U^{(0)})\|$
3: **while** $\|F(U^{(k)})\| > \varepsilon_1 \|F(U^{(0)})\|$ and $\|F(U^{(k)})\| > \varepsilon_2$ **do**
4: **Subspace correction:**
5: Given $U^{(k)} \in \mathbb{R}^n$, approximately solve the subspace problem $F_b(\hat{U}_b^{(k)}, U_g^{(k)}) = 0$ for $\hat{U}_b^{(k)}$ by INB
6: Compute $Z = [\hat{U}_b^{(k)},\ U_g^{(k)}]^T$
7: **Global update:**
8: Inexactly solve $J(Z)d^{(k)} = -F(Z)$ by a right preconditioned Krylov subspace method
9: Update $U^{(k+1)} = Z + \lambda^{(k)} d^{(k)}$, where $\lambda^{(k)} \in (0, 1]$
10: **Determine new partition:**
11: $S^{(k)} = S_b^{(k)} \cup S_g^{(k)}$
12: Set $k = k + 1$
13: **end while**

REMARK 3.1. Suppose that the bad component set, S_b, is set to be empty. In that case, the subspace correction in the first iteration ($k = 0$) is skipped, equivalent to performing one

ETNA
Kent State University and
Johann Radon Institute (RICAM)

Newton step to obtain sufficient information to determine the components to be eliminated in the next iteration.

3.2. Nonlinear elimination preconditioned inexact Newton (NEPIN). We continue using the partitioning of the unknowns and residuals introduced in Section 3.1 for the INB-ANE method; see (3.1). The nonlinear system is decomposed into bad and good components, where the bad components are associated with strongly nonlinear behavior, such as shock localization.

3.2.1. Nonlinear elimination step. Given a current iterate (U_b, U_g), the nonlinear elimination strategy seeks to approximately eliminate the bad components by computing a correction $T_b(U_b, U_g)$ such that

$$F_b(U_b - T_b,\, U_g) \approx 0.$$

Here, T_b represents an approximate update that reduces the residual associated with the bad components while keeping the good components fixed. This step targets the dominant nonlinear difficulty identified in Section 3.1.

Using this correction, we define the nonlinearly preconditioned function

$$\mathcal{F}(U_b, U_g) = \begin{bmatrix} T_b(U_b, U_g) \\ F_g(U_b, U_g) \end{bmatrix}.$$

By construction, $\mathcal{F}(U^*) = 0$ if and only if $F(U^*) = 0$, since the auxiliary correction $T_b(U)$ vanishes at the exact solution. Solving $\mathcal{F}(U) = 0$ is equivalent to solving the original system $F(U) = 0$, but the transformed system is typically much better suited for Newton-type methods.

3.2.2. Approximate Jacobian and Newton update. Forming the exact Jacobian of $\mathcal{F}$ is unnecessary in practice. Following the nonlinear elimination method of [22], we employ a block-diagonal approximation of the preconditioned Jacobian. Let J denote the Jacobian of the original system $F(U)$, and let

$$J_b = R_b J R_b^T$$

denote its restriction to the bad components, where R_b is the restriction operator defined at the beginning of Section 3. The approximate preconditioned Jacobian is given by

$$\hat{\mathcal{J}}(U) = \begin{bmatrix} J_b^{-1} & 0 \\ 0 & I_g \end{bmatrix} J(U),$$

where I_g is the identity matrix corresponding to the good components.

At each nonlinear iteration, Newton's method is applied to the nonlinearly preconditioned system

$$\hat{\mathcal{J}}(U) d = -\mathcal{F}(U).$$

In practice, this system is not formed explicitly; instead, the equivalent left-preconditioned linear system with Jacobian $J(U)$ and a modified right-hand side is solved inexactly using a Krylov subspace method.

A detailed description of the nonlinear elimination method and its algorithmic formulation can be found in [22]. The present work focuses on its algorithmic structure and performance for shock-dominated space–time discretizations, and a concrete realization of the method is given in Algorithm 2.

ETNA
Kent State University and
Johann Radon Institute (RICAM)

Algorithm 2 Nonlinear elimination preconditioned inexact Newton algorithm (NEPIN).

Input: Initial guess $U^{(0)}$
Output: Solution U^* of system $F(U) = 0$

Initialize the partition: $S_b^{(0)} = \emptyset$ and $S_g^{(0)} = S,\ k = 0$
Evaluate $F(U^{(0)})$ and $\|F(U^{(0)})\|$
while $\|F(U^{(k)})\| > \varepsilon_1 \|F(U^{(0)})\|$ and $\|F(U^{(k)})\| > \varepsilon_2$ **do**
  Subspace correction:
    Given $U^{(k)} \in \mathbb{R}^n$, solve the subspace problem $F_b(U_b^{(k)} - T_b^{(k)}, U_g^{(k)}) = 0$ approximately, using INB for $T_b^{(k)}$
    Compute global residual:
    $$g^{(k)} = [J_b^{(k)} T_b^{(k)},\ F_g(U_b^{(k)} - T_b^{(k)}, U_g^{(k)})]^T,\ J_b^{(k)} = R_b J(U_b^{(k)} - T_b^{(k)}, U_g^{(k)}) R_b^T$$
  Global update:
    Inexactly solve $J(U_b^{(k)} - T_b^{(k)}, U_g^{(k)}) s^{(k)} = -g^{(k)}$ by a preconditioned Krylov subspace method
    Update $U^{(k+1)} = U^{(k)} + \lambda^{(k)} s^{(k)}$, where $\lambda^{(k)} \in (0, 1]$
  Determine new partition:
    $S^{(k)} = S_b^{(k)} \cup S_g^{(k)}$ based on the partitioning criterion
  Set $k = k + 1$
end while

REMARK 3.2. In Algorithm 2, the step size $\lambda^{(k)}$ is determined by a backtracking line search based on the merit function $\|\mathcal{F}(U)\|$, which is chosen differently from the one in [22]. In their algorithm, the merit function of backtracking uses $\|F(U)\|$. The computational expense associated with $\|\mathcal{F}(U)\|$ is higher compared to $\|F(U)\|$. However, opting for $\|F(U)\|$ as the criterion may result in nonconvergence over time. Therefore, we decided to employ $\|\mathcal{F}(U)\|$ as our chosen criterion.

REMARK 3.3. The key steps of NEPIN and INB-ANE are summarized in Table 3.1, which highlights both the similarities and the fundamental differences between the two methods. In both approaches, a nonlinear subspace problem is solved to eliminate the bad components, leading to an intermediate state Z that is used to evaluate the Jacobian. The two methods differ, however, in how this elimination is incorporated into the subsequent Newton step.

In NEPIN, nonlinear elimination acts as a left preconditioner: the inexact Newton direction is obtained by solving a Jacobian system with a modified right-hand side that reflects the elimination of the bad components, and the solution is updated from the current iterate. In contrast, INB-ANE applies nonlinear elimination as a right preconditioner: the Newton step is computed using the standard residual evaluated at the eliminated state Z, and the solution update is performed starting from Z rather than from the original iterate. This distinction is central to the different convergence behavior of the two methods.

3.3. Challenges of nonlinear elimination in hyperbolic equations. The comparison between INB and INB-ANE for a single-shock case with various grid sizes is shown in Table 3.2. An initial guess vector of a zero vector is used for this numerical example. In addition, the algebraic-based approach is used to define the components to be eliminated that meet the condition

$$|F_i(U)| > \rho \|F(U)\|_1,$$

ETNA
Kent State University and
Johann Radon Institute (RICAM)

TABLE 3.1
Step-by-step comparison of NEPIN and INB-ANE.

NEPIN (left preconditioning)	INB-ANE (right preconditioning)
For a given partition $U = [U_b, U_g]$:	For a given partition $U = [U_b, U_g]$:
1. Compute the subspace correction T_b by solving $$F_b(U_b - T_b, U_g) = 0.$$	1. Compute the eliminated state $\hat{U}_b$ by solving $$F_b(\hat{U}_b, U_g) = 0.$$
2. Define the intermediate state $$Z = [U_b - T_b,\, U_g].$$	2. Define the reduced state $$Z = [\hat{U}_b,\, U_g].$$
3. Form the left-preconditioned residual $$g = \begin{bmatrix} J_b(Z)T_b \\ F_g(Z) \end{bmatrix}.$$	3. Form the nonlinear residual $$F(Z).$$
4. Compute an inexact Newton direction d by solving $$J(Z)d = -g.$$	4. Compute an inexact Newton direction d by solving $$J(Z)d = -F(Z).$$
5. Update the solution $$U \leftarrow U + \lambda d.$$	5. Update the solution $$U \leftarrow Z + \lambda d.$$

where $0 < \rho < 1$ is a pre-chosen constant, such as $\rho = 1/n$, with $n = N_x \times N_t$. The other possible strategy is the physics-based approach, where the physical properties correspond to the PDE to determine which components to eliminate. As shown in Table 3.2, using a nonlinear elimination-based preconditioner with INB results in a reduction in iteration counts. However, there is still some dependence on the iteration number as the grid is refined.

TABLE 3.2
Single-shock problem. The total global number of Newton iterations of INB and INB-ANE with an initial guess set to be a zero vector.

	Single-shock case			
$N_x \times N_t$	64×128	128×256	256×512	512×1024
INB	28	59	129	283
INB-ANE	24	49	105	231

As noted in [21], after conducting intensive numerical experiments through the plots of the evolution of the pointwise residual, pointwise error, components to be eliminated, and the history of the residual norm, we identified two potential problems with the residual-based elimination strategy. Firstly, a small residual does not necessarily imply a small error, making it challenging to effectively identify the problematic components. Using the single-shock problem as an example, we plotted the initial pointwise residual and error when a zero vector was used as the initial guess. As seen in Figure 3.1, the residuals and errors do not exhibit a strong pointwise correlation for algebraically smooth error components; therefore, the

ETNA
Kent State University and
Johann Radon Institute (RICAM)

subspace correction step cannot effectively reduce the influence of problematic components on Newton's convergence.

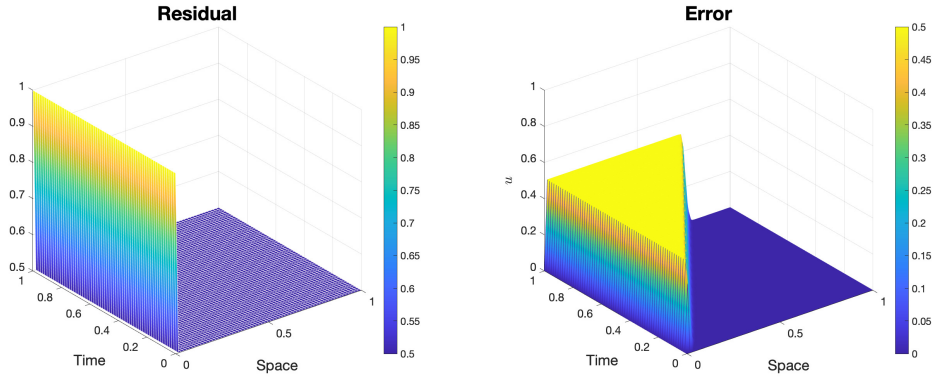

FIG. 3.1. *Single-shock problem. Pointwise space–time distributions of the residual (left) and the algebraically smooth error (right) at the first iteration using a zero initial guess. Although the residual vanishes over extended regions of the domain, the error remains nonzero because the residual is determined by the action of the stencil-based discrete operator on neighboring values. This illustrates that, for algebraically smooth error components, the residual and error do not exhibit a strong pointwise correlation.*

Hyperbolic problems exhibit directional propagation of information along characteristic paths, leading to a triangle-shaped numerical domain of dependence in space–time (see Figure 3.2, left). Therefore, when a subdomain contains a void region in its interior (see Figure 3.2, right) some propagation information may not be effectively transmitted across the subdomain. This weakened information transfer can reduce the efficiency of the iterative process.

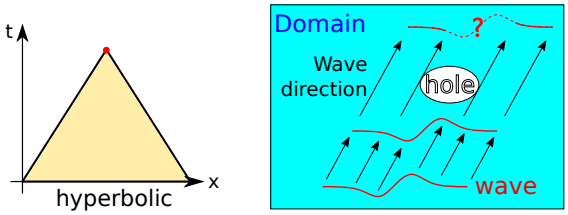

FIG. 3.2. *Illustration of directional information propagation for hyperbolic PDEs in the space–time domain. Left: the triangle-shaped numerical domain of dependence induced by characteristic propagation. Right: information loss caused by a disconnected subdomain containing an interior void, where propagation information cannot be effectively transmitted across the domain.*

3.4. Remedies: coarse-grid correction and morphological closing. We begin by summarizing two strategies from [21] that tackle challenges in the classical nonlinear elimination approach for hyperbolic problems, followed by a discussion of their practical implementation. We apply an effective coarse-grid correction technique, developed in multigrid methods, to address the first issue of the small residual not accurately reflecting the small error. We first solve the same nonlinear problem on a very coarse grid, then use this interpolated coarse-grid

ETNA
Kent State University and
Johann Radon Institute (RICAM)

solution as an initial guess for the finest problem. By doing so, we can eliminate the low-frequency error in the intermediate solution. We illustrate this effect using a simple schematic example intended only to highlight a potential difficulty in using residual information to identify dominant nonlinear components. Consider a discrete conservation-law operator evaluated at a fixed time level, and suppose that the exact solution contains a sharp jump (shock) at that time. As shown in the top panel of Figure 3.3, if a spatially constant initial guess (e.g., zero) is used, the resulting residual may be small or identically zero at many grid points, despite the presence of a large solution error localized near the shock (indicated by the red square). In this situation, residual-based criteria alone provide little information about where the dominant error resides.

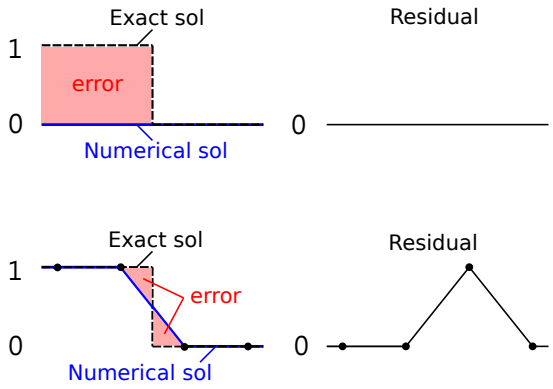

FIG. 3.3. *The relation between residual and error components for two cases, if a zero initial guess (top) or an interpolated coarse solution (bottom) is used as an initial guess.*

In contrast, when a coarse-grid correction is applied, as illustrated in the bottom panel of Figure 3.3, the interpolated solution introduces localized deviations near the shock region. These deviations give rise to nonzero residual components (highlighted by the red triangles), which more accurately reflect the location of the dominant nonlinear error. This example illustrates why coarse-grid information can be helpful in revealing shock-related error components that are not apparent from residual information alone, and it motivates the nonlinear elimination strategy adopted in this work.

Secondly, we employ the morphological closing operation, borrowed from image processing, to remove small holes in the subdomain (see Figure 3.4). The morphological closing operation is a dilation followed by an erosion using a disk structuring element. Figure 3.5 provides an example of the original domain and the domain obtained after applying the morphological closing operation in the INB-ANE algorithm. The removal of holes, gaps, and isolated islands in the original domain is evident.

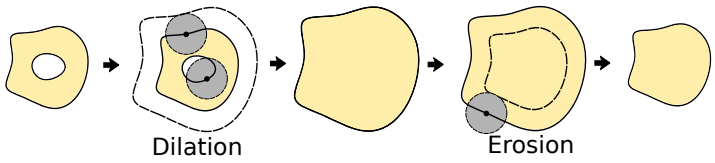

FIG. 3.4. *The morphological closing operation is a dilation followed by an erosion, using the same structuring element for both operations.*

ETNA
Kent State University and
Johann Radon Institute (RICAM)

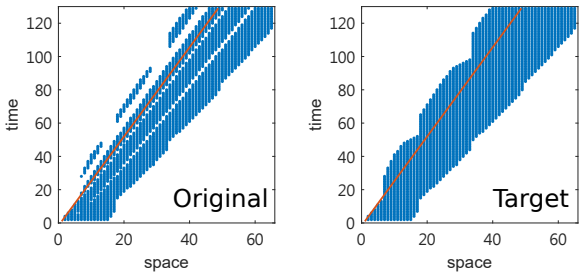

FIG. 3.5. *The original subdomain (left) and the one obtained after the morphological closing operation (right).*

To realize the above ideas in collaboration with INB-ANE and NEPIN, our proposed method consists of two additional steps: The first step involves a preprocessing stage to construct the initial guess using the interpolated coarse-grid solution obtained with INB. The coarse-grid size does not need to be very fine to bring the coarse solution into the convergence region, since its primary role is to eliminate the low-frequency error rather than to produce a good initial guess for INB. INB should not have convergence issues. After determining the new partition (e.g., Step 11 of Algorithm 2 (NEPIN)), the second step involves modifying the partitioned sets using a morphological closing operation in the INB-ANE and NEPIN partition stages. In the implementation, as shown in Figure 3.6, we use a 1D flag array to identify whether a particular component belongs to S_b. In addition, we introduce another 2D array, denoted as $\mathrm{flag}_{\mathrm{st}}$, corresponding to the space–time domain configuration. Furthermore, we have two one-to-one mappings between the 1D and 2D arrays (e.g., the `reshape` command in MATLAB) to pass the information between the solution vector-space domain and the solution time-space domain. We perform the closing operation using, for example, the MATLAB command `imclose`, which employs a structuring element (SE) to define the neighborhood around each pixel. The structuring element can be a disk, a square, or any other shape on $\mathrm{flag}_{\mathrm{st}}$ to modify the partitioned set, resulting in $\mathrm{flag}'_{\mathrm{st}}$ before passing the information back to the solution-vector space.

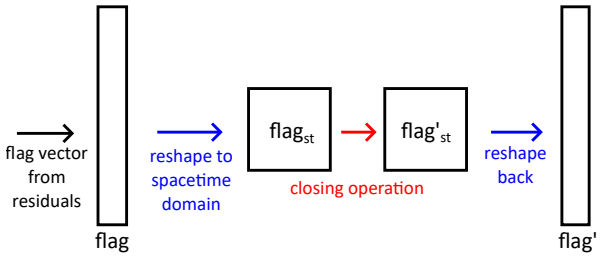

FIG. 3.6. *The closing operation in the partition stage.*

3.5. Two-level nonlinear elimination. The INB method is a common approach for subspace correction; however, it often requires many iterations for convergence, making the nonlinear elimination method impractical. To address this issue, a two-level nonlinear elimination algorithm is proposed. This algorithm applies NEPIN or INB-ANE to solve subspace nonlinear systems associated with the bad components, thereby reducing overall computing times.

ETNA
Kent State University and
Johann Radon Institute (RICAM)

The key idea of the two-level strategy is to progressively identify dominant nonlinear interactions using residual information. Firstly, grid points associated with relatively large residual contributions are selected as bad components and form the level-1 elimination set. These bad components are then treated as a new global domain, within which another residual-based partition is performed to define the level-2 elimination set. In this way, the nonlinear elimination progressively focuses on regions with stronger nonlinear effects. This hierarchical construction is motivated by the observation that, in shock-dominated hyperbolic PDEs, strong nonlinear interactions are often localized near discontinuities or steep solution fronts. By recursively applying nonlinear elimination to residual-dominated regions, the method improves the effectiveness of the subspace correction.

Figure 3.7 provides a visual representation of this two-level process, where the unknown is initially divided into good (U_g) and bad (U_b) parts. After updating the intermediate bad components of the solution, it is further divided into U_g' and U_b', and updated again. Finally, the unknowns are combined to form the entire subspace correction.

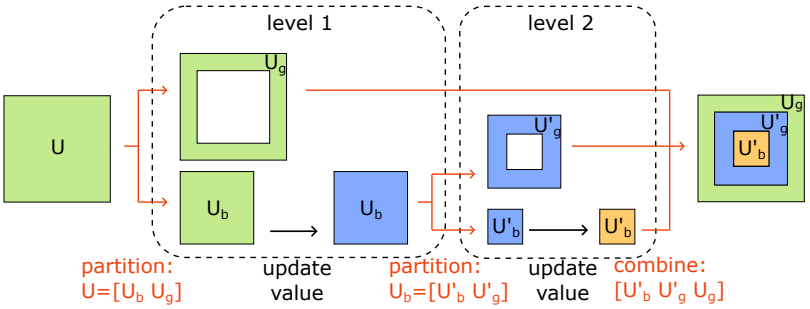

FIG. 3.7. *The partitions of the two-level nonlinear elimination domain.*

In summary, Algorithm 3 represents the complete process of the two-level NEPIN algorithm, including integrating the results of the subspace correction obtained from Algorithm 4.

Algorithm 3 Two-level NEPIN.

Input: Initial guess $U^{(0)}$

Output: Solution U of system $F(U) = 0$

1: Initialize the partition: $S_b^{(0)} = \emptyset$ and $S_g^{(0)} = S,\ k = 0$
2: **while** $\|F(U^{(k)})\| > \varepsilon_{global}$ **do**
3: $U' =$ **Two-level subspace correction** $(U^{(k)}, S_b^{(k)}, S_g^{(k)})$
4: Compute $[T_b^{(k)}, 0]^T = U^{(k)} - U'$
5: Compute global residual:

$$g^{(k)} = [J_b^{(k)} T_b^{(k)},\ F_g]^T, \quad J_b^{(k)} = R_b J(U_b^{(k)} - T_b^{(k)}, U_g^{(k)}) R_b^T$$

6: Inexactly solve $J(U_b^{(k)} - T_b^{(k)}, U_g^{(k)}) s^{(k)} = -g^{(k)}$
7: Update $U^{(k+1)} = U^{(k)} + \lambda^{(k)} s^{(k)}$, where $\lambda^{(k)} \in (0, 1]$
8: Determine the first-level partition: $S^{(k+1)} = S_b^{(k+1)} \cup S_g^{(k+1)}$
9: Set $k = k + 1$
10: **end while**

ETNA
Kent State University and
Johann Radon Institute (RICAM)

Algorithm 4 $[U'] =$ Two-level subspace correction $(\hat{U}, S_b, S_g)$.

Input: Solution $\hat{U}$, partition S_b and S_g before the subspace correction
Output: Solution U' after the subspace correction

Set $U^{(0)} = \hat{U}$
while $\|F(U)\| > \varepsilon_1^{\text{sub}} \|F(U^{(0)})\|$ **do**
  Given $U = [U_b, U_g]$, solve the first-level subspace problem:

$$F_b(U_b - T_b, U_g) = 0 \text{ for } T_b$$

  Compute $U' = [U_b - T_b, U_g]^T$
  while $\|F(U')\| > \varepsilon_2^{\text{sub}} \|F(U^{(0)})\|$ **do**
    Determine the second-level partition: $S_b = S_b' \cup S_g'$
    Given $U' = [U_b', U_g', U_g]^T$, solve the second-level subspace problem:

$$F_b(U_b' - T_b', U_g', U_g) = 0 \text{ for } T_b'$$

    Compute $U' = [U_b' - T_b', U_g', U_g]^T$
  end while
end while

4. Numerical results and discussion. This section presents numerical results for the two Burgers' cases described in Section 2.2, namely the single-shock and double-shock cases, in which INB suffers from convergence problems and traditional INB-ANE fails to detect the problematic components correctly. Additionally, we consider the Buckley–Leverett (BL) equation, which is derived from the law of mass conservation, as a model for two-phase fluid flow in porous media. We refer to this as a two-phase problem throughout this section.

4.1. Additional test case: Buckley–Leverett equation. Let $[0, 20] \times [0, 10]$ be a space–time domain. The 1D BL equation can be written as

$$u_t + [f(u)]_x = 0$$

and the flux function as

$$f(u) = \frac{u^2}{u^2 + a(1-u)^2},$$

where a is a constant that represents the ratio of the viscosity of water to that of oil. The initial condition is imposed as

$$u(x,t) = \begin{cases} 1 & \text{if } x = 0, \\ 0 & \text{if } x > 0, \end{cases}$$

and the computational domain is $x \in [0, 20]$. No boundary condition is required at $x = 0$, since all characteristics travel to the right, and no information enters the domain through this boundary.

The left panel of Figure 4.1 shows a flux function with $a = 1/2$ and $f(u)$ that has an inflection point (also see the right panel of Figure 4.1 for $f'(u)$), which implies that the Riemann solution involves both shock and rarefaction. The variable u represents water saturation, ranging from 0 to 1. A value of $u = 1$ represents pure water and $u = 0$ represents pure oil. The left panel of Figure 4.2 displays characteristic curves associated with the BL equation in the smooth regions of the solution, where the red line indicates the shock location.

ETNA
Kent State University and
Johann Radon Institute (RICAM)

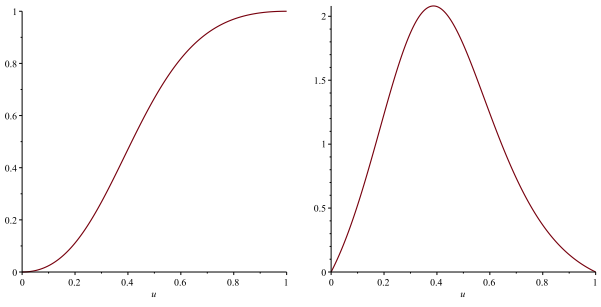

FIG. 4.1. *The flux function $f(u)$ (left) and $f'(u)$ (right) with $a = 1/2$ for the BL equation.*

The BL solution contains a rarefaction wave to the left of the shock, whose location is indicated by the red line in the left panel of Figure 4.2. The corresponding solution is shown in the right panel. The same spatial and temporal discretization schemes used for Burgers' equation in Section 2 are also used for the BL equation.

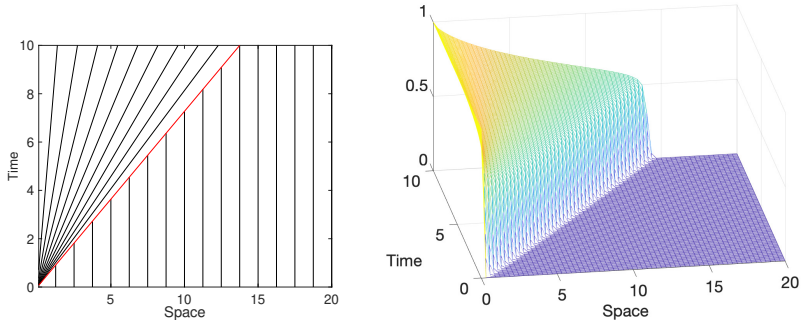

FIG. 4.2. *The family of characteristics and the shock line (left) and solution (right) to the BL equation.*

4.2. Stopping condition and other parameter setting. For INB and INB-ANE, the stopping condition is set to when the approximate solution satisfies

$$\|F(U^{(k)})\|_2 \leq 10^{-8}.$$

Similarly, for NEPIN, we have the nonlinearly preconditioned function,

$$g(U^{(k)}) = \begin{bmatrix} J_b^{(k)} T_b^{(k)}(U^{(k)}) \\ F_g(U^{(k)}) \end{bmatrix},$$

and the stopping condition is

$$\|g(U^{(k)})\| \leq 10^{-8}.$$

The same initial guess for each method is generated by bilinear interpolation of the coarse solution on the $(N_x, N_t) = (4, 8)$ grid. For INB, the generalized minimum residual method (GMRES) [26] is applied to the Jacobian of the original nonlinear system, while for NEPIN and INB-ANE, it is applied to the global Jacobian system resulting from the nonlinear

ETNA
Kent State University and
Johann Radon Institute (RICAM)

elimination formulation. A zero initial guess is used for GMRES. The local Jacobian systems appearing in the subspace correction steps are of much smaller size and are solved directly using LU factorization. For defining the nonlinear elimination preconditioner, the criterion for selecting the bad components is to calculate the average of the point-by-point residual values,

$$F_i(U^{(k)}) > \frac{\|F(U^{(k)})\|_1}{n} \quad \text{for all } 1 \le i \le (N_x \times N_t),$$

with $n = N_x \times N_t$.

Throughout the following sections, the table entries are defined as follows: "GMRES" denotes the average number of GMRES iterations per Newton iteration, and "Time" represents the computing time in seconds.

TABLE 4.1
The single-shock and double-shock problems with different algorithms. We set the stopping conditions for global Newton iterations to absolute tolerance = 1.E-08, for subspace Newton iterations to relative tolerance = 1.E-03, and GMRES iterations to tolerance = 1.E-06. We choose the disk of close operation with radius = N_x. Best timing results for each grid are highlighted in bold.

Single-shock problem					
	$N_x \times N_t$	64×128	128×256	256×512	512×1024
INB	Global Newton step (GMRES)	5 (20)	8 (31)	25 (98)	43 (171)
	Time (s)	**0.17**	**0.33**	3.18	21.45
INB-ANE	Global Newton step (GMRES)	4 (16)	5 (20)	14 (56)	22 (88)
	Subspace Newton step	7	10	32	66
	Time (s)	0.38	0.58	4.02	29.51
INB-ANE+HF	Global Newton step (GMRES)	4 (16)	4 (16)	4 (16)	5 (20)
	Subspace Newton step	6	7	24	59
	Time (s)	0.42	0.50	**2.07**	16.49
NEPIN+HF	Global Newton step (GMRES)	4 (16)	5 (20)	5 (20)	7 (26)
	Subspace Newton step	6	8	21	43
	Time (s)	0.36	0.58	2.08	**13.78**
Double-shock problem					
	$N_x \times N_t$	64×128	128×256	256×512	512×1024
INB	Global Newton step (GMRES)	14 (54)	34 (135)	49 (195)	268 (1071)
	Time (s)	**0.26**	0.97	6.23	146.97
INB-ANE	Global Newton step (GMRES)	7 (27)	11 (43)	42 (167)	163 (651)
	Subspace Newton step	17	45	113	484
	Time (s)	0.45	1.07	12.14	221.35
INB-ANE+HF	Global Newton step (GMRES)	4 (16)	5 (20)	5 (20)	6 (24)
	Subspace Newton step	15	40	63	328
	Time (s)	0.47	1.01	4.46	81.74
NEPIN+HF	Global Newton step (GMRES)	5 (20)	5 (20)	6 (23)	7 (27)
	Subspace Newton step	16	37	56	312
	Time (s)	0.48	**0.94**	**4.17**	**78.57**

4.3. Efficiency and robustness of INB and NEPIN methods for Burgers' equation. This section compares the efficiency of the INB-ANE and NEPIN methods for Burgers' equation. Table 4.1 summarizes numerical results for the single- and double-shock problems using grid sizes ranging from 64×128 to 512×1024. In addition to the subspace-based methods, results from the standard INB method are reported as a baseline and to illustrate a

ETNA
Kent State University and
Johann Radon Institute (RICAM)

fundamental limitation of a two-grid nested-iteration strategy in shock-dominated problems. In particular, the convergence of INB may stagnate when the method fails to correctly identify shock locations, even when a coarse solution is used as the initial guess, and the grid is refined. The four algorithms considered are the standard INB method, the traditional INB-ANE method, and their variants incorporating a hole-filling strategy, namely INB-ANE+HF and NEPIN+HF. For the hole fill-in procedure, we employ a morphological closing operation using a disk-shaped structuring element with radius proportional to the spatial grid size N_x.

From Table 4.1, it is observed that the number of global Newton iterations required by the traditional INB-ANE method increases rapidly as the grid is refined. This trend, already evident in the single-shock case, becomes substantially more pronounced for the double-shock problem. For example, when the grid size increases from 256×512 to 512×1024, the number of global Newton iterations for INB-ANE increases by nearly a factor of five. The INB results further demonstrate that grid refinement alone does not resolve this difficulty: when shock locations are not accurately captured by the intermediate solution, the Newton iterations remain dominated by incorrectly identified bad components, leading to stagnation.

To address this issue, the hole-filling strategy is incorporated into both INB-ANE- and NEPIN-based preconditioners to improve the identification of bad components arising from the hyperbolic nature of the equation. The results show that INB-ANE+HF significantly reduces the number of global Newton iterations on fine grids. In particular, for the grid size 512×1024, the number of global Newton iterations is reduced to approximately 23% of that required by INB-ANE, corresponding to a reduction of about 77%. However, the number of global Newton iterations alone does not fully determine the overall computational cost. The number of subspace Newton iterations, together with the size and frequency of the selected subspaces, also plays an important role in the total execution time, especially on refined grids.

Overall, the NEPIN+HF method performs best among the methods considered. Its advantage is particularly evident for large-scale problems, where it achieves the lowest total computing time in both the single-shock and double-shock cases, indicating improved robustness in shock identification on fine meshes.

4.4. Elimination component distribution analysis. Figure 4.3 illustrates the evolution of bad components for the single-shock case on the space–time domain during the first four global iterations of INB-ANE+HF and NEPIN+HF on the 256×512 grid. The blue region represents the distribution of bad components, while the white region corresponds to good components. The red line indicates the exact location of the shock.

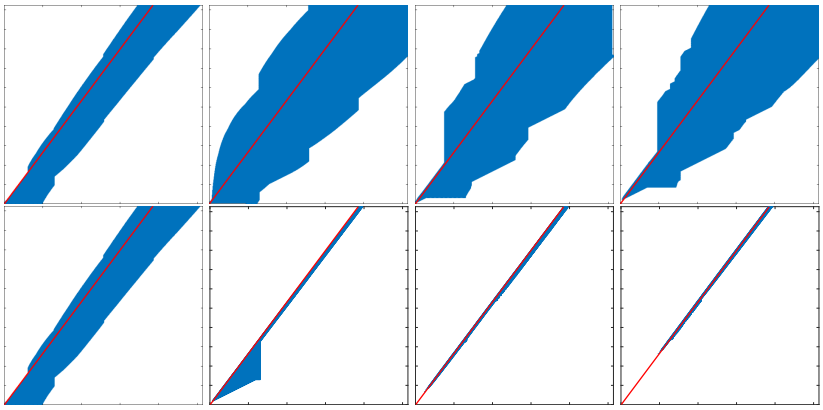

FIG. 4.3. *A typical evolution of bad components with INB-ANE+HF (first row) and NEPIN+HF (second row).*

ETNA
Kent State University and
Johann Radon Institute (RICAM)

As discussed earlier, after the initial phase, both INB-ANE+HF and NEPIN+HF reduce to the classical Newton method once the set of bad components becomes empty. Consequently, during the first global iteration, both methods exhibit identical residual values and select subspaces of equal size. However, starting from the second global iteration, a clear difference in the distribution of bad components emerges. Although the same nonlinear preconditioning technique is applied on the left or right side, this choice constitutes the main distinction between NEPIN+HF and INB-ANE+HF. As shown in Figure 4.3, NEPIN+HF identifies a smaller set of bad components that are more closely aligned with the shock line compared to INB-ANE+HF. As a result, NEPIN+HF requires a smaller subspace for correction and incurs less average overhead per subspace problem than INB-ANE+HF. This behavior provides an important practical advantage, particularly for large-scale problems.

To understand why INB-ANE+HF requires a larger set of bad components, we consider the case illustrated on the left of Figure 4.4. This figure shows the distribution of bad components in the space–time domain after the first global iteration. Along the line $t = 0.5$, the bad components are located in the interval $x \in [0.3, 0.6]$. The right panels of Figure 4.4 show, respectively, the intermediate solution (top) and the pointwise residual (bottom) for INB-ANE+HF and NEPIN+HF after the first global iteration at $t = 0.5$. By examining the interface between the bad and good components in the numerical solution, we observe that spurious discontinuities are present in the INB-ANE+HF solution. The corresponding pointwise residuals exhibit pronounced peaks at these interfaces, which strongly influence the selection of bad components in the subsequent global iteration.

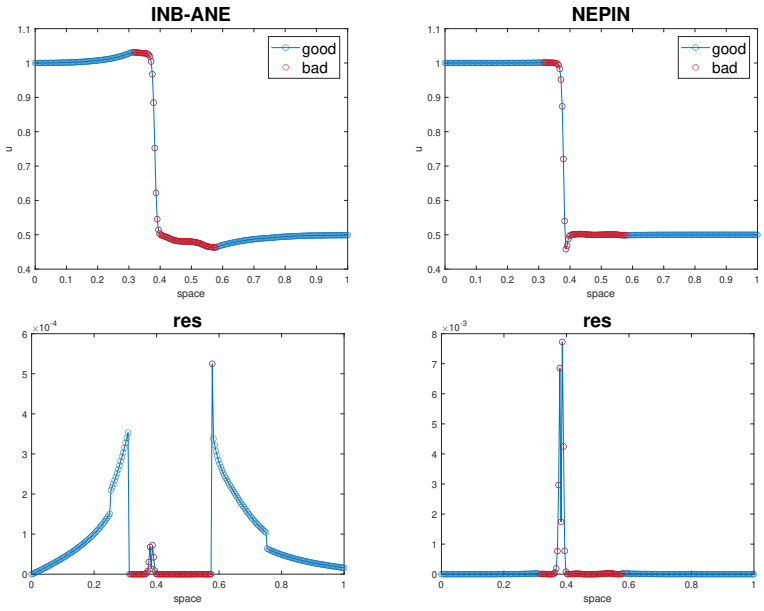

FIG. 4.4. *Single-shock case: Cross-sectional solution (first row) and the pointwise residual component along* $t = 0.5$ *for INB-ANE+HF (left column) and NEPIN+HF (right column) on a* 256×512 *grid.*

In contrast, the NEPIN+HF method is less sensitive to the artificial boundaries introduced by subspace partitioning. As a result, it is able to more accurately capture the shock location, leading to a more localized selection of bad components. This improved shock identification contributes to faster and more robust convergence of the Newton iterations.

ETNA
Kent State University and
Johann Radon Institute (RICAM)

4.5. Performance comparison of INB and NEPIN+HF for the two-phase flow problem. In Table 4.2, we compare the iteration counts and corresponding computing times for the two-phase flow problem obtained using three methods: INB, one-level NEPIN+HF, and two-level NEPIN+HF. The numerical experiments are performed on a sequence of uniformly refined space–time grids ranging from 64×128 to 512×1024. For NEPIN+HF, we additionally report the computational overhead associated with the subspace correction steps, measured in terms of extra iterations and computing time relative to INB. The stopping tolerances for the first- and second-level subspace corrections in Algorithm 4 are set to $\varepsilon_1^{\text{sub}} = 10^{-3}$ and $\varepsilon_2^{\text{sub}} = 10^{-1}$, respectively.

TABLE 4.2
A comparison of one-level and two-level NEPIN+HF in terms of the iterations and time for the two-phase flow problem. Best timing results for each grid are highlighted in bold.

		Two-phase flow problem			
		64×128	128×256	256×512	512×1024
INB	Global Newton step	39	73	219	374
	Time (s)	0.52	2.2	34.15	238.31
One-level NEPIN+HF	Global Newton step	7	9	12	15
	Level-1 subspace Newton step	25	64	101	310
	Time (s)	0.78	1.63	8.68	88.35
Two-level NEPIN+HF	Global Newton step	7	9	12	15
	Level-1 subspace Newton step	15	30	57	83
	Level-2 subspace Newton step	9	22	21	49
	Time (s)	**0.48**	**1.14**	**5.46**	**32.37**

Table 4.2 shows that, as in the Burgers' problem with shocks, the number of Newton iterations required by INB and the corresponding computing time increase significantly as the grid is refined. This effect is even more pronounced for the two-phase flow problem, which poses additional challenges for Newton-type methods posed by the simultaneous presence of shock and rarefaction waves. In contrast, NEPIN+HF demonstrates substantially improved performance. In the finest grid tested (512×1024), NEPIN+HF reduces the total computing time by approximately a factor of three compared to INB, while exhibiting only a mild dependence on grid refinement. However, for the one-level NEPIN+HF method, the computational overhead associated with the subspace correction (the first level of nonlinear elimination) remains relatively high. By applying nonlinear elimination hierarchically, as in the two-level NEPIN+HF method, the overhead due to subspace corrections is significantly reduced. As a result, the total computing time is further reduced, yielding a more than 60% reduction relative to the one-level NEPIN+HF method.

4.6. Stiffness analysis for INB and NEPIN+HF. Next, we perform a stiffness analysis [24] for INB and NEPIN+HF. The purpose of the present stiffness analysis is primarily to provide qualitative insight into the stagnation behavior observed during the nonlinear convergence process, rather than to establish rigorous local or global convergence theory. Recall that the Newton iteration with backtracking for solving a nonlinear system $F(U) = 0$ is given by

$$U^{(k+1)} = U^{(k)} - \lambda^{(k)} J^{-1}(U^{(k)}) F(U^{(k)}),$$

where $J(U)$ denotes the Jacobian of F, and $\lambda^{(k)} \in (0, 1]$ is the step size. In the continuous-time limit, this iteration leads to the Newton ODE

$$J(U)\frac{\mathrm{d}U}{\mathrm{d}t} = -F(U).$$

ETNA
Kent State University and
Johann Radon Institute (RICAM)

Linearizing $F(U)$ around a steady-state solution U^* yields the generalized eigenvalue problem

$$Av = \mu Bv,$$

where $A = -J(U^*)$ and $B = J(U)$. The stiffness ratio is then defined as

$$S = \frac{\max_j |\mathrm{Re}(\mu_j)|}{\min_j |\mathrm{Re}(\mu_j)|}.$$

A large value of S indicates a stiff system and suggests that reducing strong nonlinear interactions through nonlinear preconditioning may improve the convergence behavior of INB for solving $\mathcal{F}(U) = 0$.

We revisit the single-shock and double-shock cases of Burgers' equation, along with the two-phase flow problem presented in the previous section. An interpolation from the coarse-grid solution on the 4×8 grid is used to set the initial guess. All results in this section are obtained using the 128×256 grid. Comparing the Newton ODE system corresponding to INB and NEPIN+HF, Figure 4.5 displays the evolution of the stiffness ratio as Newton iterates for both methods. The stiffness ratios decrease monotonically for the single- and double-shock cases. In addition, for the two-phase flow problem, we observe a similar trend, with the difference in the ratio fluctuating until approximately the 35th iteration. It should be noted that all the ratios converge to 1 as expected, which is also consistent with the relationship: the matrix pair $(-J(U^*), J(U^{(k)})) \to (-J(U^*), J(U^*))$ as $U^{(k)} \to U^*$, where the generalized eigenvalues are all $\mu = -1$. Consequently, the stiffness ratio tends towards 1.

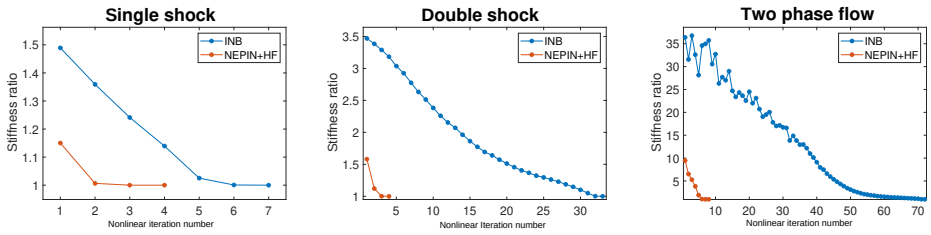

FIG. 4.5. *A comparison of stiffness ratio histories with respect to nonlinear iterations for INB and NEPIN+HF in single-shock (left), double-shock (middle) and two-phase flow (right) problems.*

At the beginning of the Newton method, a noticeable difference in the stiffness ratio is observed between the cases with and without preconditioning. In the single-shock case, preconditioning reduces the stiffness ratio by about 20% compared to no preconditioning. Similarly, for the double-shock and two-phase flow scenarios, preconditioning reduces the stiffness ratio by more than half compared to the cases without preconditioning. After the first iteration, the stiffness ratio remains low and quickly converges to 1.

In Figure 4.6, we track the frequency of Newton backtracking utilized in each global Newton step and compute the associated stiffness ratio of the Newton ODE. The red and green points denote the respective correlations produced by INB and NEPIN+HF. We categorize these points based on the number of backtracking instances, with the single-shock and double-shock cases divided into two groups (0 and 1) and the two-phase flow cases into three groups (0, 1, and 2). For each group, we use a box plot to observe the distribution of the stiffness ratio. In the single-shock case context, delineating the number of Newton backtracking instances can be readily achieved using a stiffness ratio threshold of 1.2. Specifically, when the stiffness ratio exceeds 1.2, a single Newton backtracking instance is employed, whereas when the stiffness

ETNA
Kent State University and
Johann Radon Institute (RICAM)

ratio falls below 1.2, no Newton backtracking is initiated. The box-plot analysis suggests a correlation between the number of Newton backtracking instances and the median stiffness ratio for the double-shock and two-phase flow problems. Specifically, it is observed that a higher number of Newton backtracking instances corresponds to a higher median stiffness ratio. On the other hand, NEPIN predominantly exhibits distributions with fewer Newton backtracking instances compared to INB. Using the two-phase flow scenario as an illustration, it is noteworthy that all points within group 0 originate from NEPIN. From these examples, it can be inferred that the analysis of the stiffness ratio is also applicable to space–time hyperbolic equations.

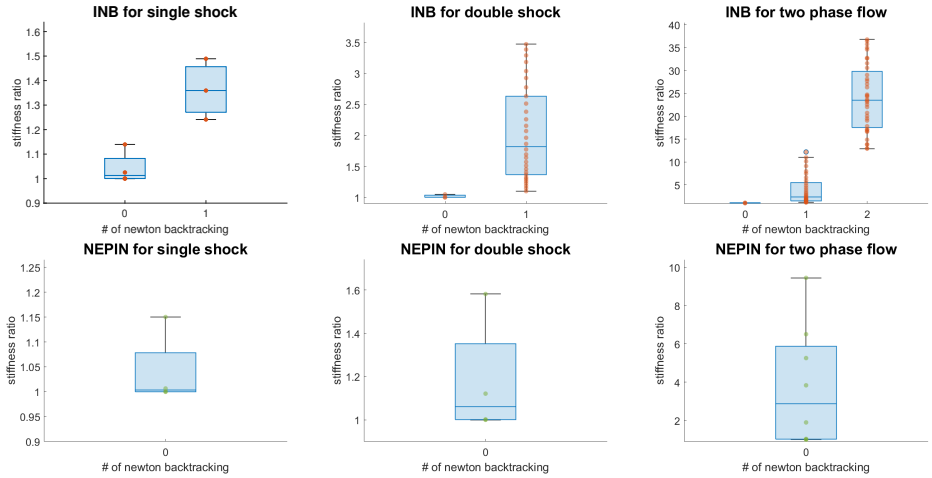

FIG. 4.6. *The relationship between stiffness ratio and the number of Newton backtracking steps for INB (first row) and NEPIN+HF (second row).*

We emphasize that the present stiffness analysis is primarily interpretive in nature. In its current form, the analysis requires knowledge of the converged solution and relies on the solution of generalized eigenvalue problems. A more detailed study of the associated block spectral structures and the development of practical diagnostics for nonlinear preconditioning remain topics for future work.

5. Conclusions. In this study, we have investigated a variant of the nonlinear elimination preconditioning technique in conjunction with inexact Newton methods to better align with the characteristics of space–time hyperbolic equations. Two types of nonlinear elimination preconditioned methods were studied, applied either on the right or left: INB-ANE and NEPIN. The first ingredient was the coarse-grid correction technique, which gradually addresses the challenges posed by low-frequency errors by using the initial guess of the Newton method, generated by interpolating the solution of the coarse grid. The second ingredient was to characterize the properties of the domain of dependence by using morphological closing operations to fill the hole in the computational region associated with the subspace problem. We also studied a two-level version of NEPIN to reduce the overhead of subspace corrections, which proved beneficial in our numerical experiments. The numerical results for the fully coupled space–time solution of Burgers' and two-phase flow problems demonstrated that the enhanced method effectively decreases the number of global Newton iterations. The iteration count exhibited insensitivity to variations in grid sizes.

ETNA
Kent State University and
Johann Radon Institute (RICAM)

We further performed a stiffness analysis motivated by the connection between Newton's method and numerical ODEs. In particular, we compared the stiffness ratios associated with the Newton ODEs for INB and NEPIN. Using an example of an unbalanced nonlinear system, the numerical results suggest that the stiffness ratio associated with NEPIN remains significantly smaller and less sensitive to the degree of nonlinearity than that of INB. Additionally, we observed a correlation between the stiffness ratio and the number of Newton backtracking iterations. In several numerical examples, including the single- and double-shock problems and the two-phase flow problem, larger stiffness ratios were often accompanied by more Newton backtracking steps and increased numbers of global Newton iterations. Compared with INB, the NEPIN method generally reduces the stiffness ratio and improves the efficiency of the Newton method.

Although the present study focuses on one-dimensional model problems, the proposed nonlinear elimination method is not restricted to one-dimensional settings. Extensions to multidimensional hyperbolic PDEs and more complex multiphase flow problems are possible. However, these problems often involve multiple characteristic families, more complicated wave interactions, and multidimensional wave propagation patterns, which make the identification of suitable nonlinear elimination regions more challenging.

The present study focuses on the mathematical properties and serial performance of the proposed algorithms. Given their inherently parallel structure, a distributed-memory implementation and a full scalability performance investigation will be pursued in future work.

REFERENCES

[1] D. S. BALSARA, *An analysis of the hyperbolic nature of the equations of radiation hydrodynamics*, J. Quant. Spectrosc. Radiat. Transf., 61 (1999), pp. 617–627.
[2] A. T. BARKER AND M. STOLL, *Domain decomposition in time for PDE-constrained optimization*, Comput. Phys. Commun., 197 (2015), pp. 136–143.
[3] G. BOILLAT, *Nonlinear electrodynamics: Lagrangians and equations of motion*, J. Math. Phys., 11 (1970), pp. 941–951.
[4] T. A. BRUNNER, T. S. HAUT, AND P. F. NOWAK, *Nonlinear elimination applied to radiation diffusion*, Nucl. Sci. Eng., 194 (2020), pp. 939–951.
[5] R. M. COLOMBO, *Hyperbolic phase transitions in traffic flow*, SIAM J. Appl. Math., 63 (2002), pp. 708–721.
[6] X. DAI AND Y. MADAY, *Stable Parareal in time method for first- and second-order hyperbolic systems*, SIAM J. Sci. Comput., 35 (2013), pp. A52–A78.
[7] X. DENG, X.-C. CAI, AND J. ZOU, *A parallel space-time domain decomposition method for unsteady source inversion problems*, Inverse Probl. Imaging, 9 (2015), pp. 1069–1091.
[8] ———, *Two-level space-time domain decomposition methods for three-dimensional unsteady inverse source problems*, J. Sci. Comput., 67 (2016), pp. 860–882.
[9] J. E. DENNIS, JR. AND R. B. SCHNABEL, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, SIAM, Philadelphia, 1996.
[10] H. DE STERCK, T. A. MANTEUFFEL, S. F. MCCORMICK, AND L. OLSON, *Numerical conservation properties of $H(\mathrm{div})$-conforming least-squares finite element methods for the Burgers equation*, SIAM J. Sci. Comput., 26 (2005), pp. 1573–1597.
[11] M. J. GANDER AND S. VANDEWALLE, *Analysis of the Parareal time-parallel time-integration method*, SIAM J. Sci. Comput., 29 (2007), pp. 556–578.
[12] S. GONG AND X.-C. CAI, *A nonlinear elimination preconditioned inexact Newton method for heterogeneous hyperelasticity*, SIAM J. Sci. Comput., 41 (2019), pp. S390–S408.
[13] T.-T.-P. HOANG, C. JAPHET, M. KERN, AND J. E. ROBERTS, *Space-time domain decomposition for reduced fracture models in mixed formulation*, SIAM J. Numer. Anal., 54 (2016), pp. 288–316.
[14] A. J. HOWSE, H. DE STERCK, R. D. FALGOUT, S. MACLACHLAN, AND J. SCHRODER, *Parallel-in-time multigrid with adaptive spatial coarsening for the linear advection and inviscid Burgers equations*, SIAM J. Sci. Comput., 41 (2019), pp. A538–A565.
[15] J. HUANG, C. YANG, AND X.-C. CAI, *A nonlinearly preconditioned inexact Newton algorithm for steady state lattice Boltzmann equations*, SIAM J. Sci. Comput., 38 (2016), pp. A1701–A1724.

ETNA
Kent State University and
Johann Radon Institute (RICAM)

[16] F.-N. HWANG, H.-L. LIN, AND X.-C. CAI, *Two-level nonlinear elimination based preconditioners for inexact Newton methods with application in shocked duct flow calculation*, Electron. Trans. Numer. Anal., 37 (2010), pp. 239–251.
https://etna.ricam.oeaw.ac.at/vol.37.2010/pp239-251.dir/pp239-251.pdf

[17] F.-N. HWANG, Y.-C. SU, AND X.-C. CAI, *A parallel adaptive nonlinear elimination preconditioned inexact Newton method for transonic full potential equation*, Comput. & Fluids, 110 (2015), pp. 96–107.

[18] A. JAMESON, *The evolution of computational methods in aerodynamics*, J. Appl. Mech., 50 (1983), pp. 1052–1070.

[19] O. A. KRZYSIK, H. DE STERCK, R. D. FALGOUT, AND J. B. SCHRODER, *Parallel-in-time solution of scalar nonlinear conservation laws*, SIAM J. Sci. Comput., 47 (2025), pp. A3134–A3160.

[20] S. LI AND X.-C. CAI, *Convergence analysis of two-level space-time additive Schwarz method for parabolic equations*, SIAM J. Numer. Anal., 53 (2015), pp. 2727–2751.

[21] C.-W. LIANG AND F.-N. HWANG, *Nonlinear elimination preconditioned space-time solution algorithms for hyperbolic problems*, in Domain Decomposition Methods in Science and Engineering XXVIII, P. Bjørstad, X.-C. Cai, V. Dolean, D. E. Keyes, R. Kornhuber, and J. Xu, eds., Lect. Notes Comput. Sci. Eng. 155, Springer, Cham, 2026, pp. 101–108.

[22] L. LIU, F.-N. HWANG, L. LUO, X.-C. CAI, AND D. E. KEYES, *A nonlinear elimination preconditioned inexact Newton algorithm*, SIAM J. Sci. Comput., 44 (2022), pp. A1579–A1605.

[23] L. LUO, X.-C. CAI, AND D. E. KEYES, *Nonlinear preconditioning strategies for two-phase flows in porous media discretized by a fully implicit discontinuous Galerkin method*, SIAM J. Sci. Comput., 43 (2021), pp. S317–S344.

[24] L. LUO, C.-W. LIANG, F.-N. HWANG, AND X.-C. CAI, *Insights into nonlinear preconditioning for Newton-type methods through stiffness analysis*, Preprint, 2026.

[25] L. LUO, W.-S. SHIU, R. CHEN, AND X.-C. CAI, *A nonlinear elimination preconditioned inexact Newton method for blood flow problems in human artery with stenosis*, J. Comput. Phys., 399 (2019), Art. No. 108926, 20 pages.

[26] Y. SAAD, *Iterative Methods for Sparse Linear Systems*, 2nd ed., SIAM, Philadelphia, 2003.

[27] H. TANG, S. WANG, C. YIN, Y. DI, Y.-S. WU, AND Y. WANG, *Fully-coupled multi-physical simulation with physics-based nonlinearity-elimination preconditioned inexact Newton method for enhanced oil recovery*, Commun. Comput. Phys., 25 (2019), pp. 244–265.

[28] H. YANG AND X.-C. CAI, *Parallel fully implicit two-grid methods for distributed control of unsteady incompressible flows*, Internat. J. Numer. Methods Fluids, 72 (2013), pp. 1–21.

[29] ———, *Two-level space-time domain decomposition methods for flow control problems*, J. Sci. Comput., 70 (2017), pp. 717–743.

[30] H. YANG AND F.-N. HWANG, *An adaptive nonlinear elimination preconditioned inexact Newton algorithm for highly local nonlinear multicomponent PDE systems*, Appl. Numer. Math., 133 (2018), pp. 100–115.

[31] ———, *A nonlinear elimination preconditioner for fully coupled space-time solution algorithm with applications to high-Rayleigh number thermal convective flow problems*, Commun. Comput. Phys., 26 (2019), pp. 749–767.

[32] H. YANG, F.-N. HWANG, AND X.-C. CAI, *Nonlinear preconditioning techniques for full-space Lagrange-Newton solution of PDE-constrained optimization problems*, SIAM J. Sci. Comput., 38 (2016), pp. A2756–A2778.

[33] H. YANG, S. SUN, Y. LI, AND C. YANG, *Parallel reservoir simulators for fully implicit complementarity formulation of multicomponent compressible flows*, Comput. Phys. Commun., 244 (2019), pp. 2–12.

[34] M. ZHENG, F. LIU, I. TURNER, AND V. ANH, *A novel high order space-time spectral method for the time fractional Fokker-Planck equation*, SIAM J. Sci. Comput., 37 (2015), pp. A701–A724.