TWO-LEVEL NONLINEAR ELIMINATION BASED PRECONDITIONERS FOR INEXACT NEWTON METHODS WITH APPLICATION IN SHOCKED DUCT FLOW CALCULATION[*]

FENG-NAN HWANG[†], HSIN-LUN LIN[†], AND XIAO-CHUAN CAI[‡]

Abstract. The class of Newton methods is popular for solving large sparse nonlinear algebraic systems of equations arising from the discretization of partial differential equations. The method offers superlinear or quadratic convergence when the solution is sufficiently smooth and the initial guess is close to the desired solution. However, in many practical problems, the solution may exhibit some non-smoothness in part of the computational domain, due to, for example, the presence of a shock wave. In this situation, the convergence rate of Newton type methods deteriorates considerably. In this paper, we introduce a two-level nonlinear elimination algorithm, in which we first identify a subset of equations that prevents Newton from having the fast convergence and then iteratively eliminate them from the global nonlinear system of equations. We show that such implicit nonlinear elimination restores the fast convergence for problems with local non-smoothness. As an example, we study a compressible transonic flow in a shocked duct.

Key words. Nonlinear PDEs, nonlinear elimination, inexact Newton, finite difference, shock wave

AMS subject classifications. 65H10, 65N06, 65N55

1. Introduction.

In [13], an interesting nonlinear elimination algorithm (NE) was introduced for solving large sparse nonlinear systems of equations whose solution is badly scaled in part of the computational domain. The key idea of NE is to implicitly remove these components and obtain a better balanced system for which the classical inexact Newton method can be applied. The technique works extremely well for some relatively simple cases, and several recent attempts motivated by this technique have been made in order to make inexact Newton methods work for other nonlinear systems [3, 4, 5, 6, 7, 10, 11, 12]. In NE, an important step is to iteratively eliminate the identified bad component using a subdomain Newton method, which by itself may fail or take too many iterations to converge. In [13] it was suggested that the nonlinear elimination algorithm can be used in a nested fashion; i.e., NE can be used inside the outer NE when the regular inexact Newton fails to converge in the implicit removing step for the subnonlinear system. Even though the idea of nested NE is simple, but it has never been studied and to actually realize it is quite difficult. The aim of this paper is to formulate a two-level NE and embed it into the classical inexact Newton methods, which can be interpreted as a nonlinear Schur complement algorithm.

We briefly recall the classical inexact Newton algorithm with backtracking (INB) [8, 9], which is used as the basic building block of our algorithms for the global and some subnonlinear systems. Consider a given nonlinear function $F(x)$: $R^n \rightarrow R^n$, we

[*]The first two authors were supported in part by the National Science Council of Taiwan, 96-2115-M-008-007-MY2 and the last author was supported in part by the Department of Energy, DE-FC-02-06ER25784, and in part by the National Science Foundation, ACI-0305666, CCF-0634894, and CNS-0722023.

[†]Department of Mathematics, National Central University, Jhongli City, Taoyuan County 32001, Taiwan (*hwangf@math.ncu.edu.tw* and *942201024@cc.ncu.edu.tw*).

[‡]Department of Computer Science, University of Colorado at Boulder, Boulder, CO 80309, USA (*cai@cs.colorado.edu*).

are interested in finding a vector $x^* \in R^n$, such that

$$F(x^*) = 0, \tag{1.1}$$

starting from an initial guess $x^{(0)} \in R^n$. Here $F = (F_1, \ldots, F_n)^T$, $F_i = F_i(x_1, \ldots, x_n)$, and $x = (x_1, \ldots, x_n)^T$.

Algorithm 1.1 (Inexact Newton with Backtracking (INB)).

```
Given $x^{(0)}$
Evaluate $F(x^{(0)})$ and $\|F(x^{(0)})\|$
Set $k = 0$
While ($\|F(x^{(k)})\| > \varepsilon_1 \|F(x^{(0)}\|$) and ($\|F(x^{(k)})\| > \varepsilon_2$) do
    Compute the Jacobian matrix $F'(x^{(k)})$
    Inexactly solve the Jacobian system $F'(x^{(k)})s^{(k)} = -F(x^{(k)})$
    Update $x^{(k+1)} = x^{(k)} + \lambda^{(k)} s^{(k)}$, where $\lambda^{(k)} \in (0,1]$ is determined to satisfy
        $\|F(x^{(k)} + \lambda^{(k)} s^{(k)})\| \leq (1 - \alpha\lambda^{(k)})\|F(x^{(k)})\|$
    Set $k = k + 1$
End While
```

Here ε_1 and ε_2 are the relative and absolute stopping conditions. For the applications that we are interested, n is usually large, and in this case, the algorithm has three expensive operations: the evaluation of $F(\cdot)$, the construction of the Jacobian matrix, and the solve of the Jacobian system. It is important to note that all three operations are global in the sense that all components of x and F are involved in all three operations. However, as observed in many numerical experiments, the trigger of these expensive "all components involved" operations is often local. In other words, only a small number of $F_1, F_2, \ldots, F_n$ are large and these "bad components" are not random, they are often associated with certain interesting physics of the solution of the PDE. For example, in the shocked duct flow problem that we are looking at, all these "bad components" are associated with the shock wave located in a small region inside the computational domain. In other applications, they may be associated with a boundary layer or other local singularities [13, 14, 16]. NE is a subproblem solver inside a global INB that is designed to smooth out these "bad components" so that the total number of global INB is reduced.

The rest of the paper is organized as follows. In Section 2, we formulate the multilevel NE algorithm. We describe a shocked duct flow problem in Section 3. Some numerical results and concluding remarks are given in Sections 4 and 5, respectively.

2. Multilevel nonlinear elimination algorithms.

We begin with the one-level nonlinear elimination algorithm. The first step is to split the residual components, $F_1, F_2, ..., F_n$, into two sets consisting of the "bad components" to be eliminated and the good components to be solved by the classical inexact Newton algorithm. Let $I = \{1, 2, \ldots, n\}$ be an index set; i.e. one integer for each unknown x_i and each residual function F_i. Assume that S_1^b ("b" for bad) is a subset of I with m components and S_1^g ("g" for good) with $(n-m)$ components is its complement; that is

$$I = S_1^b \cup S_1^g.$$

Usually $m \ll n$. For this partition, we define two subspaces

$$V_1^b = \{v | v = (v_1, ..., v_n)^T \in R^n, v_k = 0 \text{ if } k \notin S_1^b\}$$

and

$$V_1^g = \{v | v = (v_1, ..., v_n)^T \in R^n, v_k = 0 \text{ if } k \notin S_1^g\},$$

respectively, and the corresponding restriction operators, R_1^b and R_1^g, which transfers data from R^n to V_1^b and V_1^g, respectively. Here, we use the subscript 1 to indicate the partition, the subspaces, and the restriction/interpolation operators at the first level, which is used to distinguish the corresponding ones defined later at the second level. Using the restriction operator R_1^b, we define the sub-nonlinear function $F_{S_1^b} : R^n \to V_1^b$ as

$$F_{S_1^b}(x) = R_1^b(F(x)).$$

For any given $x \in R^n$, $T^b(x)$: $R^n \to V_1^b$ is defined as the solution of the following subspace nonlinear system,

$$F_{S_1^b}(R_1^g x + T^b(x)) = 0. \tag{2.1}$$

Using the subspace mapping functions, we introduce a new global nonlinear function,

$$y = G(x) \equiv R_1^g x + T^b(x).$$

Note that for a given x, the evaluation of $G(x)$ is not straightforward, a nonlinear system corresponding to the subspace V_1^b has to be solved using either the classical INB algorithm restricted to the subspace V_1^b or a NE algorithm in the subspace V_1^b. Let us summarize the above procedure as the following algorithm.

Algorithm 2.1 (Evaluate $y = G(x)$).

```
If flag=0 then $y = x$ else
    If flag=1: one-level nonlinear elimination:
    Solve (2.1) by INB using $R_1^b x$ as an initial guess.
    If flag=2: two-level nonlinear elimination:
    Solve (2.1) by one-level NE using $R_1^b x$ as an initial guess.
endif
Compute $y = R_1^g x + T^b(x)$
```

Here *flag* is an input parameter from somewhere else of the algorithm to indicate if a nonlinear elimination is needed and if the one-level or two-level NE is to be called. Two-level NE becomes necessary when the local problem (2.1) is still too difficult to solve by INB, and in this case, another partition of the index set S_1^b into two subsets is needed, i.e. $S_1^b = S_2^b \cup S_2^g$. At the second level for the subset S_1^b two subspaces of R^n, V_2^b and V_2^g, the corresponding restriction operators, R_2^b and R_2^g, as well as sub-nonlinear function $F_{S_2^b}$ can all be defined in a manner similar to the ones at the first level.

Now NE in conjunction with INB can be realized in the following algorithm.

Algorithm 2.2 (INB-NE).

```
Given $x^{(0)}$. Set $k = 0$ and flag= 1
Compute $y^{(0)} = G(x^{(0)})$.
Evaluate $F(y^{(0)})$ and $\|F(y^{(0)})\|$
While ($\|F(y^{(k)})\| > \varepsilon_1 \|F(y^{(0)}\|$) and ($\|F(y^{(k)})\| > \varepsilon_2$) do
        Compute $F'(y^{(k)})$
        Inexactly solve $F'(y^{(k)})s^{(k)} = -F(y^{(k)})$
        Update $x^{(k+1)} = x^{(k)} + \lambda^{(k)} s^{(k)}$, where $\lambda^{(k)}$ is determined to satisfy
```

```
            $\|F(G(x^{(k)} + \lambda^{(k)} s^{(k)}))\| \leq (1 - \alpha\lambda^{(k)})\|F(y^{(k)})\|$
        Compute $y^{(k+1)} = G(x^{(k+1)})$
        Evaluate $F(y^{(k+1)})$ and $\|F(y^{(k+1)})\|$
        If $\|F(y^{(k)})\| < \varepsilon_3 \|F(y^{(0)})\|$ then flag=0
        Set $k = k + 1$
    End While
```

INB-NE can be interpreted as follows. Find the solution $y^* \in R^n$ of (1.1) by solving a right nonlinearly preconditioned system, $F(G(x^*)) = 0$. Once x^* is found, the solution of the original system can be obtained as $y^* = G(x^*)$. It was shown theoretically in [13] that under certain assumptions, INB-NE possesses local quadratic convergence provided that the subspace nonlinear problems (2.1) are solved exactly. Note that if $F(x)$ is linear, i.e.

$$F(x) \equiv \left(\begin{array}{cc} B & E \\ F & C \end{array}\right)\left(\begin{array}{c} R_1^b x \\ R_1^g x \end{array}\right) - \left(\begin{array}{c} f \\ g \end{array}\right) = \left(\begin{array}{c} 0 \\ 0 \end{array}\right)$$

then

$$y = \left(\begin{array}{c} T^b(x) \\ R_1^g x \end{array}\right) = \left(\begin{array}{c} -B^{-1}E(R_1^g x) + B^{-1} f \\ R_1^g x \end{array}\right).$$

As a consequence, solving $F(G(x)) = 0$ is mathematically equivalent to decouple it into two steps: first solve the reduced system, $U(R_1^g x) = g - FB^{-1}f$, where $U = C - FB^{-1}E$ is the Schur complement matrix then compute $R_1^b x = -B^{-1}E(R_1^g x) + B^{-1}f$. However, rather than solving the Schur complement system, in practice, it is desirable and often more efficient to solve the full system, since the Schur complement is denser, and good preconditioners may not be available.

Note that in our approach the subproblem corresponding to the bad components is simply a restriction of the global nonlinear system to the subdomain, not a Schur complement of the global system with respect to the subdomain consisting of the bad components. The Schur complement approach is considerably more expensive and is not studied in this paper.

Since the extra function evaluations of $G(x)$ are needed, NE is intended for the cases, in which INB fails to converge or experiences unacceptably slow convergence. As suggested by [13](the bottom of pp. 555), when the intermediate solution is close to the exact solution, NE is switched back INB by letting $G(x) = x$. The switching condition is controlled by ε_3 in Algorithm 2.

3. A shocked duct flow problem.

Compressible flows passing through a diverge-converge duct are governed by the compressible Navier-Stokes equations [1, 2]. Instead of solving Navier-Stokes equations, we consider a simpler model problem, a quasi-one-dimensional full potential problem [6, 16] defined on the interval, $0 \leq x \leq 2$, as follows.

$$\left\{\begin{array}{l} (A(x)\rho(\phi_x)\phi_x)_x = 0, \\ \phi(0) = 0 \ \text{ and } \ \phi(L) = \phi_R, \end{array}\right. \tag{3.1}$$

where $A(x)$ is the area of the cross-section of the duct at x

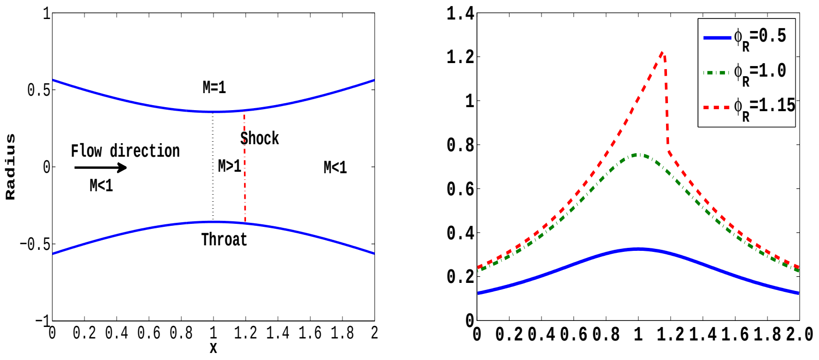
$$A(x) = 0.4 + 0.6(x-1)^2$$

and the density function is described as

$$\rho(u) = (c^2)^{1/(\gamma-1)} = \left(1 + \frac{1}{2}(\gamma-1)(1-u^2)\right)^{\frac{1}{\gamma-1}}. \tag{3.2}$$

Here $\gamma = 1.4$ is the specific heat for air, $u = \phi_x$ is the flow velocity, and c is the speed of sound. See the left figure of Fig. 3.1 for the geometric configuration of the shocked duct flow problem. Although this problem looks quite simple, it is still considered as a difficult test problem for the convergence of inexact Newton methods because the solution has a strong shock as the value of ϕ_R becomes larger than 1.15 in the domain, as shown in Fig. 3.1 (right figure). The flow is supersonic at the points in the interval (0,2), where the Mach number, $M = |u|/c$, is greater than 1.

FIG. 3.1. *Left: transonic flow in a converge-diverge duct; Right: Mach number curves for different right boundary condition ϕ_R, grid size $h = 1/256$.*



To approximate (3.1) by a standard finite difference method, we begin by introducing a uniform grid $0 = x_0 < x_1 < x_2 ...... < x_n = 2$ with the grid size $h = 2/n$. Let $\Phi = (\phi_1^h, \phi_2^h, \cdots, \phi_{n-1}^h)^T$ be the numerical approximations at these interior grid points. We define points, $x_{i-1/2}$ and $x_{i+1/2}$, as the midpoints of subintervals $[x_{i-1}, x_i]$ and $[x_i, x_{i+1}]$, respectively. Consider the subinterval $[x_{i-1/2}, x_{i+1/2}]$, we approximate $(A\rho(\phi_x)\phi_x)_x$ at the point x_i using a second-order centered finite difference method, i.e.

$$\frac{A_{i+\frac{1}{2}}\rho_{i+\frac{1}{2}}\dfrac{\phi_{i+1}^h - \phi_i^h}{h} - A_{i-\frac{1}{2}}\rho_{i-\frac{1}{2}}\dfrac{\phi_i^h - \phi_{i-1}^h}{h}}{h} = 0.$$

For the leftmost grid point, we have $A_{\frac{3}{2}}\rho_{\frac{3}{2}}(\phi_2^h - \phi_1^h) - A_{\frac{1}{2}}\rho_{\frac{1}{2}}\phi_1^h = 0$ and for the rightmost grid point we have $A_{n-\frac{1}{2}}\rho_{n+\frac{1}{2}}(K - \phi_{n-1}^h) - A_{n-\frac{3}{2}}\rho_{n-\frac{3}{2}}(\phi_{n-1}^h - \phi_{n-2}^h)$. Here $A_{i\pm1/2} = A((x_{i\pm1} \pm x_i)/2)$ and $\rho_{i\pm1/2} = \rho((\phi_x)_{i\pm1/2})$.

For purely subsonic flows, using (3.2) for calculating the flow density is sufficient. However, for transonic flows, this formulation needs to be modified in order to capture the shock. By applying a first-order density upwinding scheme as suggested by Young *et al.* [14, 16], a modified flow density value at the point $x_{i+1/2}$ is expressed as

$$\widetilde{\rho}_{i+\frac{1}{2}} = \rho_{i+\frac{1}{2}} - \widetilde{\mu}_{i+\frac{1}{2}}(\rho_{i+\frac{1}{2}} - \rho_{i-\frac{1}{2}}),$$

where the switching parameter $\widetilde{\mu}_{i+\frac{1}{2}}$ is defined as $\widetilde{\mu}_{i+\frac{1}{2}} = \max\{\mu_{i-\frac{1}{2}}, \mu_{i+\frac{1}{2}}, \mu_{i+\frac{3}{2}}\}$ with $\mu_{i+1/2} = \max\{0, 1 - M_c^2/M_{i+1/2}^2\}$. Here M_c is called the cutoff Mach number and $M_{i+1/2}$ is the numerical Mach number at $x_{i+1/2}$ given by

$$M_{i+1/2} \approx (u_x)_{i+1/2}/\rho_{i+1/2}^{\frac{1}{\gamma-1}} \approx \left(\frac{\phi_{i+1}^h - \phi_i^h}{h}\right) / \rho\left(\frac{\phi_{i+1}^h - \phi_i^h}{h}\right)^{\frac{1}{\gamma-1}}.$$

In summary, the discrete shocked duct flow problem can be written as a large sparse nonlinear system of algebraic equations,

$$F(\Phi) = 0, \tag{3.3}$$

where $F(\Phi) = (F_1(\Phi), F_2(\Phi), \cdots, F_n(\Phi))^T$ with F_i defined as

$$F_i(\Phi) = \begin{cases} (A_{\frac{3}{2}}\widetilde{\rho}_{\frac{3}{2}})\phi_2^h - (A_{\frac{3}{2}}\widetilde{\rho}_{\frac{3}{2}} + A_{\frac{1}{2}}\widetilde{\rho}_{\frac{1}{2}})\phi_1^h \\ (A_{i+\frac{1}{2}}\widetilde{\rho}_{i+\frac{1}{2}})\phi_{i+1}^h - (A_{i+\frac{1}{2}}\widetilde{\rho}_{i+\frac{1}{2}} + A_{i-\frac{1}{2}}\widetilde{\rho}_{i-\frac{1}{2}})\phi_i^h + (A_{i-\frac{1}{2}}\widetilde{\rho}_{i-\frac{1}{2}})\phi_{i-1}^h \\ \qquad \text{for any} \quad 2 \le i \le n-2 \\ (A_{i+\frac{1}{2}}\widetilde{\rho}_{i+\frac{1}{2}})K - (A_{n-\frac{1}{2}}\widetilde{\rho}_{n-\frac{1}{2}} + A_{n-\frac{3}{2}}\widetilde{\rho}_{n-\frac{3}{2}})\phi_{n-1}^h + (A_{n-\frac{3}{2}}\widetilde{\rho}_{n-\frac{3}{2}})\phi_{n-2}^h. \end{cases}$$

In our implementation, the Jacobian matrix of $F(\Phi)$ is constructed approximately by using the forward finite differences. Note that for the case of purely subsonic flows, the formulation (3.3) leads to a symmetric, weakly diagonally dominant, tridiagonal Jacobian matrix, while for the case of transonic flows the associate Jacobian is non-symmetric due to the derivative of the upwinding density coefficient $\widetilde{\rho}_{i\pm\frac{1}{2}}$ corresponding to the supersonic region.

4. Numerical experiments and observations. In this section, we present some numerical results for solving the shocked duct flow problem (3.3) using the classical inexact Newton method and the new algorithm. The stopping condition for Newton is

$$\|F(x^{(k)})\| \le \max\{10^{-8}\|F(x^{(0)})\|, 10^{-10}\},$$

and a linear initial guess that interpolates the boundary conditions is used for Newton for all test cases. In our implementation of the classical inexact Newton method with backtracking, as described in Algorithm 1.1, a right preconditioned GMRES [15] is used for solving the global Jacobian system with zero initial guess. The stopping condition for GMRES is

$$\|F(x^{(k)}) + (F'(x^{(k)})M_k^{-1})(M_k s^{(k)})\| \le \max\{\eta\|F(x^{(k)})\|, 10^{-10}\}.$$

Here $\eta = 10^{-6}$ and M_k^{-1} is a block Jacobi preconditioner constructed using the matrix $F'(x^{(k)})$. In the tests, we partition the computational domain into 15 non-overlapping subdomains and therefore M_k^{-1} has 15 blocks. The global Newton step is updated by

$$x^{(k+1)} = x^{(k)} + \lambda^{(k)} s^{(k)}.$$

The step length, $\lambda^{(k)} \in [\lambda_{\min}, \lambda_{\max}] \subset (0, 1]$, is selected so that

$$\|F(x^{(k)} + \lambda^{(k)} s^{(k)})\| \le (1 - \alpha\lambda^{(k)})\|F(x^{(k)})\|,$$

where the two parameters $\lambda_{\min}$ and $\lambda_{\max}$ act as safeguards, which are required for strong global convergence and the parameter α is used to assure that the reduction of $\|F\|$ is sufficient. Here, a quadratic linesearch technique [8] is employed to determine the step length $\lambda^{(k)}$, with $\alpha = 10^{-4}$, $\lambda_{\min} = 1/10$ and $\lambda_{\max} = 1/2$.

In the implementation of the new algorithm, two more nested Newton solvers are needed. We simply use the inexact Newton just described for all nonlinear solvers.

4.1. Classical inexact Newton. We first show some results using the inexact Newton methods *with* and *without* backtracking for solving the problem. In Table 4.1, we show the number of Newton iterations on three grids of size 1/64, 1/128 and 1/256 with four different boundary conditions $\phi_R = 0.5, 1.0, 1.15, 1.18$. For this set of tests, the inexact Newton without backtracking fails to converge when the grid is fine and ϕ_R is large, but INB converges in all cases. However, the stronger the shock wave is the more INB iterations are needed for convergence. In the left figure of Fig. 4.1, we show the convergence history of INB on the three different grids. For all cases, INB converges rapidly at the beginning (the nonlinear residual is reduced by more than one order of magnitude in the first few iterations) and then stagnates for a while before exhibiting the quadratic convergence behavior. Clearly, the finer the grid the longer the stagnation period becomes. To understand how INB updates the intermediate solution during the stagnation period, we focus on the case with grid size equals to 1/128. INB takes 223 steps to converge, and the 11 selected Mach curves corresponding to the computed velocities are shown in the right figure of Fig. 4.1. It is interesting to observe that at most grid points the solution convergence happens after the second INB iteration, and the rest of the INB iterations are devoted exclusively for grid points near the shock. Note that, practically speaking, after the second INB, the Newton corrections are needed only in the neighborhood of the shock, but the Newton calculations (including the nonlinear residual evaluation and the Jacobian solve) are actually carried out for the whole computational domain. This is clearly a waste of computation!

Table 4.1
A comparison of the number of iterations of inexact Newton without backtracking (IN) and INB. 'Div.' means divergence.

	IN		
grid sizes (h)	1/64	1/128	1/256
$\phi_R = 0.50$	3	3	3
$\phi_R = 1.00$	6	6	6
$\phi_R = 1.15$	13	22	Div.
$\phi_R = 1.18$	14	25	Div.
	INB		
$\phi_R = 0.50$	3	3	3
$\phi_R = 1.00$	4	4	4
$\phi_R = 1.15$	46	223	735
$\phi_R = 1.18$	83	278	1009

To further understand the situation from an algebraic viewpoint, we partition the interval $\Omega = (0.0, 2.0)$ into three subintervals, $\Omega_1 = (0.0, 0.8)$, $\Omega_2 = (0.8, 1.3)$, and $\Omega_3 = (1.3, 2.0)$, with the middle interval contains the shock. Correspondingly, we partition the nonlinear vector-valued function $F(\cdot)$ into three pieces, $F_1(\cdot)$ for the subinterval to the left of the shock neighborhood, $F_2(\cdot)$ for the subinterval containing the shock neighborhood, and $F_3(\cdot)$ for the subinterval to the right of the shock neighborhood. Note that the solution components at the grid points in $\Omega_1 \cup \Omega_3$ represent the *good* components while the ones in Ω_2 correspond to the *bad* components. In Fig. 4.2, we show the residual of a test run on a $h = 1/128$ grid using INB. We include the history of the complete residual $\|F\|$, the smooth part of the residual $\sqrt{\|F_1\|^2 + \|F_3\|^2}$, and the non-smooth part of the residual $\|F_2\|$. It is important to note that the residual is completely dominated by F_2; the two curves virtually sit on top of each other in Fig. 4.2.

FIG. 4.1. *Left: Convergence history of INB norm of nonlinear residuals for different grid sizes. $\phi_R = 1.15$; Right: Convergence history of Mach number curves, $h = 1/128$.*

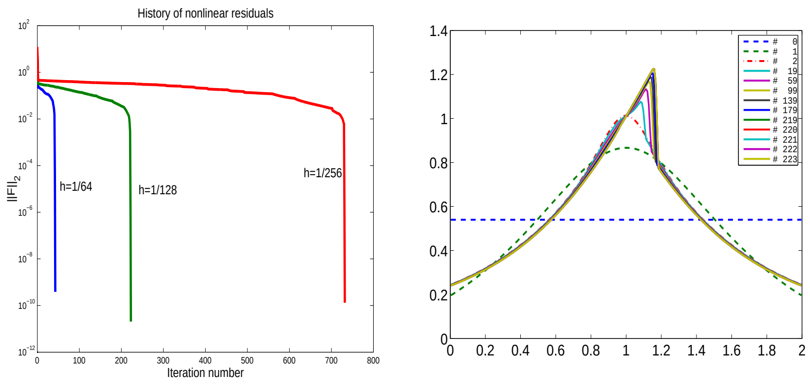
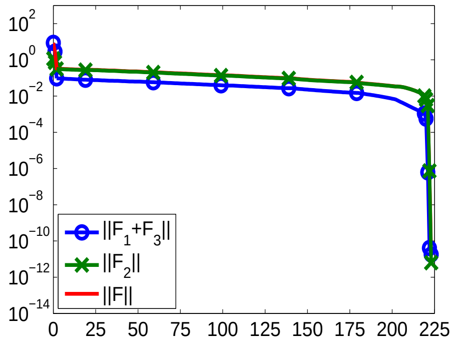


FIG. 4.2. *History of nonlinear residual for sub-functions corresponding the bad, the good, and all components, $\|F_2\|$, $\|F_1 + F_3\|$, and $\|F\|$, respectively. Note that the curves corresponding to $\|F_2\|$ and $\|F\|$ are virtually on top of each other.*



4.2. One-level INB-NE. On the other hand, in Fig. 4.3, we show the residual of the same test case on a $h = 1/128$ grid as in Fig. 4.2 by using the one-level INB-NE algorithm (INB-NE1), in which the bad component near the shock is eliminated with an inner Newton iteration. It is clear that, after the elimination, the component F_2 is no longer the dominant term in the overall residual, and the convergence of the outer Newton takes only 5 iterations.

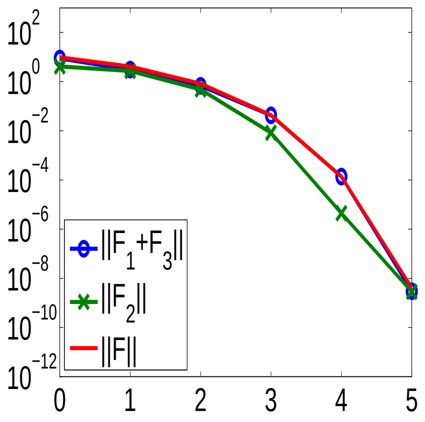
When using INB-NE1, a key question is how to properly pick the bad components. In Table 4.2, we show the number of iterations with different choices of the "bad" interval. Note that the shock is located at the point near $x = 1.2$. As mentioned before, to exactly solve the local problem is essential for the fast convergence of INB-NE1. In practice, from our numerical experiences, the elimination calculation has to be carried out with a certain degree of accuracy. Otherwise, INB-NE1 may fail to converge. Hence, for the results in Table 4.2, we use the following stopping condition

$$\|F_{S_1^b}(x_{b,1}^{(k)})\| \le \max\{10^{-8}\|F_{S_1^b}(x_{b,1}^{(0)})\|, 10^{-10}\}$$

for the nonlinear system, and

$$\|F_{S_1^b}(x_{b,1}^{(k)}) + F'_{S_1^b}(x_{b,1}^{(k)})s^{(k)}\| \le \max\{10^{-2}\|F_{S_1^b}(x_{b,1}^{(k)})\|, 10^{-10}\}$$

FIG. 4.3. *After nonlinear elimination, the nonlinear residual of INB corresponding the good components becomes more dominant and INB converges very fast.* $h = 1/128$. *Note that the curves corresponding to* $\|F_1 + F_3\|$ *and* $\|F\|$ *are virtually on top of each other.*



for the subdomain Jacobian system, which is solved by GMRES without preconditioning. Table 4.2 suggests that the appropriate subinterval of the “bad” components should include all grid points near the location of the duct throat and the shock.

TABLE 4.2
Subinterval selection for INB-NE1: $h = 1/128$, $\varepsilon_3 = 10^{-6}$.

subinterval	its	subinterval	its	subinterval	its	subinterval	its	subinterval	its
$[0.8, 1.0]$	144	$[0.9, 1.0]$	162	$[1.0, 1.1]$	167	$[1.1, 1.2]$	40	$[1.2, 1.3]$	208
$[0.8, 1.1]$	92	$[0.9, 1.1]$	115	$[1.0, 1.2]$	9	$[1.1, 1.3]$	40	$[1.2, 1.4]$	197
$[0.8, 1.2]$	6	$[0.9, 1.2]$	8	$[1.0, 1.3]$	7	$[1.1, 1.4]$	41	$[1.2, 1.5]$	154
$[0.8, 1.3]$	6	$[0.9, 1.3]$	7	$[1.0, 1.4]$	7	$[1.1, 1.5]$	24		
$[0.8, 1.4]$	6	$[0.9, 1.4]$	5	$[1.0, 1.5]$	7				
$[0.8, 1.5]$	8	$[0.9, 1.5]$	5						

To make the INB-NE algorithm more efficient, it is wise to sometimes switch to the outer INB iteration from the inner elimination iteration. This is controlled by the parameter ϵ_3 in Algorithm 2.2. In Table 4.3, we show the number of INB iterations in the bad subdomain with different values of ϵ_3. When ϵ_3 is too large, more outer iterations is needed, but below certain value, it is simply a waste of computation. In Fig. 4.4, we show the history of Mach number distribution curves corresponding to both $x^{(k)}$ and $y^{(k)}$. In Algorithm 2.2, $x^{(k)}$ is the solution from the outer Newton iteration and $y^{(k)}$ is the modified $x^{(k)}$ with the subspace correction. Note that NE correctly detects the location of the shock at the second iteration, but it takes 77 iterations to solve the local problem. Clearly after the subspace correction, the sequence $y^{(k)}$ converges quickly to the desired solution. Hence, the switching should take place as soon as the location of the shock is detected. For experiments in the rest of the section, we set $\epsilon_3 = 10^{-4}$.

Table 4.4 summarizes the number of outer Newton iterations, the average number of inner Newton iterations and the average number of GMRES iterations for solving the global Jacobian systems in INB-NE1. We observe that once the bad components are removed the total number of outer INB iterations stays small regardless the size of the grid and the right boundary condition, which controls the strength of the shock.

Table 4.3
The inner INB iteration numbers in INB-NE1 algorithm with different ε_3. Subinterval: $[0.8, 1.3]$. $h = 1/128$. $\phi_R = 1.15$.

	inner INB its		
NE iteration	$\varepsilon_3 = 10^{-2}$	$\varepsilon_3 = 10^{-4}$	$\varepsilon_3 = 10^{-6}$
0	3	3	3
1	5	5	5
2	77	77	77
3	32	32	32
4	0	34	34
5	0	0	34
6	0		

Table 4.4
A summary of the number of the outer INB iterations, the average number of inner Newton iterations per outer INB iteration and the average number of GMRES iterations for solving the global Jacobian systems in INB-NE1 with $\varepsilon_3 = 10^{-4}$.

	outer INB its, ave. inner INB its (ave. GMRES its)			
(ϕ_R)	subinterval	$h = 1/64$	$h = 1/128$	$h = 1/256$
0.5	[0.8,1.3]	3,3.0 (30.0)	3,3.0 (30.0)	3, 3.0 (30.0)
1.00	[0.8,1.3]	4,3.8 (30.0)	4,3.8 (30.0)	4, 3.8 (30.0)
1.15	[0.8,1.3]	5,10.4 (31.2)	5,30.2 (31.2)	6,99.8 (32.7)
1.18	[0.8,1.7]	5,17.6 (32.4)	5,53.0 (32.2)	9,212.4 (33.6)

4.3. Two-level INB-NE. When using the one-level algorithm, as discussed in the previous subsection, for some cases, the subspace Newton solver may need many iterations (e.g. the cases of $h = 1/256$ with $\phi_R = 1.15$ and 1.18 in Table 4.4) and sometime the subspace Newton may even fail to converge. In this situation, one may want to use the one-level algorithm recursively; i.e., further partition the subdomain into "good" and "bad" sub-subdomains and then introduce a third Newton solver in the "bad" sub-subdomain. For both levels, we use the following stopping condition

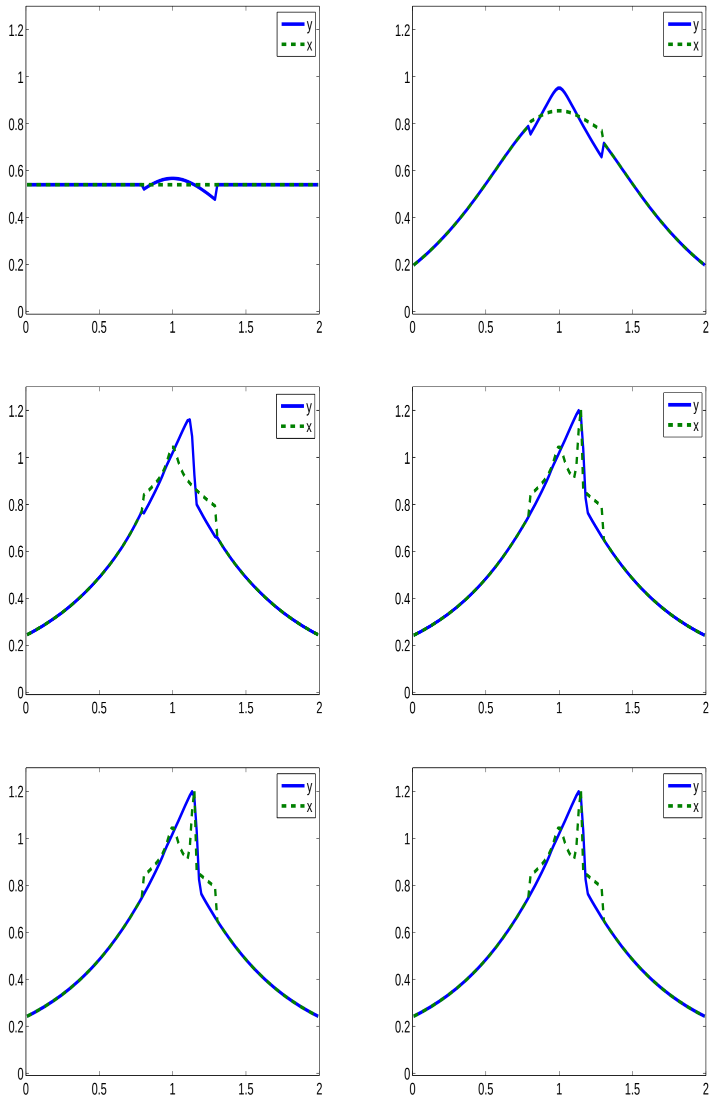
$$\|F_{S_*^b}(x_{b,*}^{(k)})\| \leq \max\{10^{-8}\|F_{S_*^b}(x_{b,*}^{(0)})\|, 10^{-10}\}$$

for the nonlinear systems, and

$$\|F_{S_*^b}(x_{b,*}^{(k)}) + F'_{S_1^b}(x_{b,*}^{(k)})s^{(k)}\| \leq \max\{10^{-6}\|F_{S_*^b}(x_{b,*}^{(k)})\|, 10^{-10}\}$$

for the subdomain Jacobian systems, which are solved by GMRES without preconditioning and with a zero initial guess. The notation "*" represents 1: 1st level or 2: 2nd level. In Table 4.5, we show the number of iterations of three nested INB iterations for different choices of subintervals at each level. How to choose an appropriate subinterval at each level is mostly empirical. From our numerical experiences, the subinterval at the second level should be covered by that at the first level and both subintervals should include the shock and the throat. Note that with this additional level of nonlinear elimination, the number of INB iterations in the first level can be kept small. Fig. 4.5 shows the history of Mach number curves corresponding to x and the original solution y. Similar to INB-NE1, with the help of NE2, INB correctly detects the location of the shock at the 1st iteration, and it takes only 5 NE iterations to solve the local problem.

Fig. 4.4. *From top to bottom, from left to right: the outer iteration starting from 0 to 5 in INB-NE1 for the Mach number curves corresponding to x and original solution y. Subinterval: [0.8,1.3], $h = 1/128$, $\phi_R = 1.15$, and $\varepsilon_3 = 10^{-4}$*



5. Concluding remarks and future work. When solving nonlinear system of algebraic equations using inexact Newton methods, the convergence is often determined by a small number of equations in the system that are much more nonlinear than the others. In the paper, we developed several methods that implicitly eliminate these highly nonlinear components through an approximate inner subdomain Newton iterations. The number of outer Newton iterations, which are considerably more expensive than the inner subdomain iterations, can be drastically reduced if the highly nonlinear components are correctly identified and sufficiently removed. A shocked duct flow problem was carefully studied. For this problem, all the bad components of

Table 4.5

2nd level subinterval selections for INB-NE2. The number in the table are the number of 1st level NE iterations, the average number of 2nd-level INB iterations, and the average number of the inner-most INB iterations. Note that the duct throat is located at $x = 1.0$ *and the shock is at around* $x = 1.2$. $h = 1/128$, $\phi_R = 1.15$, *and* $\varepsilon_3 = 10^{-4}$

1st level subinterval	2nd level subinterval		
[0.5,1.5]	$[0.8, 1.3]$	$[0.9, 1.3]$	$[1.0, 1.3]$
	4,4.8,25.8	4,4.8,18.7	4,5.5,15.7
[0.8,1.3]	$[0.85, 1.25]$	$[0.95, 1.25]$	$[1.05, 1.25]$
	5,6.4,37.7	5,4.0,22.1	5,5.0,17.0
[1.0,1.3]	$[1.05, 1.15]$	$[1.1, 1.15]$	$[1.10, 1.25]$
	7,8.2,17.3	7,17.3,11.8	7,13.3,23.9

the nonlinear system are near the shock wave, and our numerical results showed that once these bad components are approximately removed, the number of outer Newton iterations is reduced from over 200 to just 5, for a particular example on a $h = 1/128$ grid. A two-level version of the algorithm was also introduced using a combination of the idea of two-level nonlinear elimination and classical inexact Newton type methods.

The focus of the paper was on the convergence of the one-level and two-level algorithms and we did not discuss anything related to computing time. As a future project, we will consider the extension of the algorithms to two and three dimensional spaces. In INB-NE, a judicious choice of the bad subspace is crucial for fast convergence of Newton methods. In the shocked duct flow problem, as illustrated numerically in the previous section, this choice depends on the location of the shock. We know where these bad components physically are and we observe that they do not move during the solution process, hence Algorithm 2.2 can be employed. In practice, it may not always be possible to determine before hand which subsystem to be eliminated. Therefore, it is necessary to develop a domain decomposition version of Algorithm 2.2 [7], which extends NE where a single local problem is considered to multiple local problems. The new nonlinear domain decomposition based algorithm is able to identify this subspace without already knowing the solution profile and it is more suitable for large scale parallel processing.

REFERENCES

[1] J. D. Anderson, Jr., *Fundamentals of Aerodynamics*, McGraw-Hill, New York, NY, 1991.

[2] J. D. Anderson, Jr., *Computational Fluid Dynamics: The Basics with Applications*, McGraw-Hill, New York, NY, 1995.

[3] X.-C. Cai, W. D. Gropp, D. E. Keyes, R. G. Melvin, and D.P. Young, *Parallel Newton-Krylov-Schwarz algorithms for the transonic full potential equation*, SIAM J. Sci. Comput., 19 (1998), pp. 246-265.

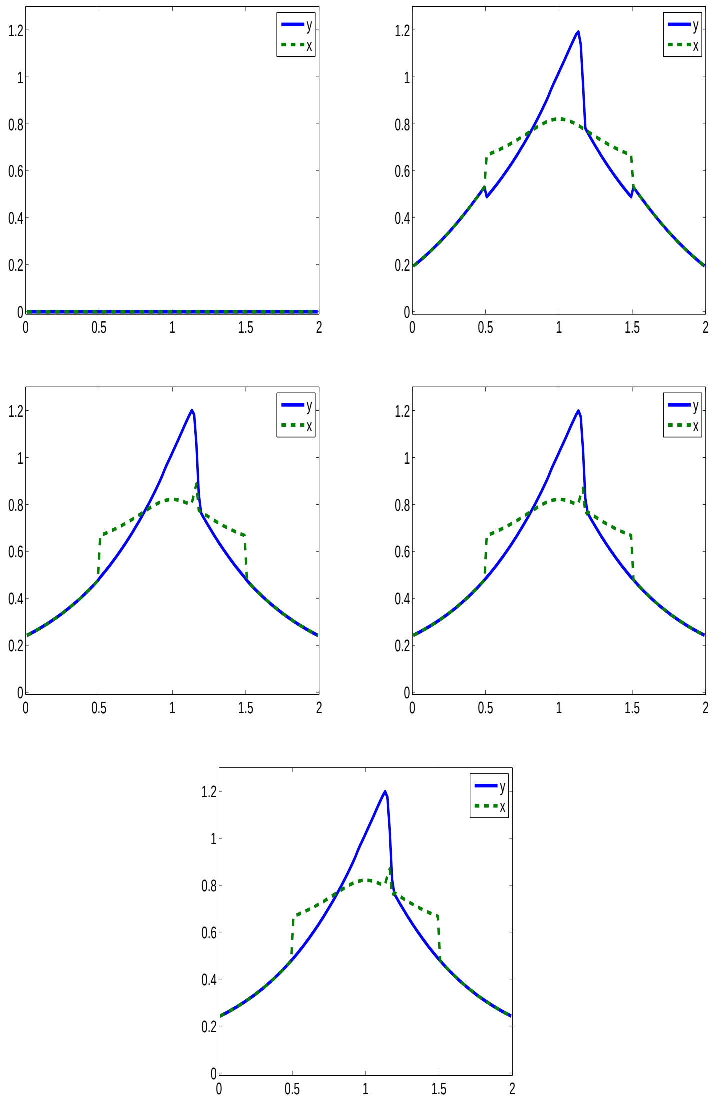
[4] X.-C. Cai and D. E. Keyes, *Nonlinearly preconditioned inexact Newton algorithms*, SIAM J. Sci. Comput., 24 (2002), pp. 183-200.

[5] X.-C. Cai, D. E. Keyes, and L. Marcinkowski, *Nonlinear additive Schwarz preconditioners and applications in computational fluid dynamics*, Int. J. Numer. Meth. Fluids, 40 (2002), pp. 1463-1470.

[6] X.-C. Cai, D. E. Keyes, D. P. Young, A nonlinear additive Schwarz preconditioned inexact Newton method for shocked duct flow, Domain Decomposition Methods in Science and Engineering, N. Debit, M. Garbey, R. Hoppe, J. Periaux, D. E. Keyes, and Y. Kuznetsov eds., CIMNE, Barcelona, 2002 pp. 343-350.

[7] X.-C. Cai, *Nonlinear overlapping domain decomposition methods*, in Domain Decomposition Methods in Science and Engineering XVIII, M. Bercovier, M.J. Gander, R. Kornhuber, and O. Widlund, eds, Lect. Notes in Computational Science and Engieering, 70, Springer-Verlag, Heidelberg, 2009, pp. 217-224.

FIG. 4.5. *From top to bottom, from left to right: the outer iteration starting from 0 to 4 in INB-NE2 for the Mach number curves corresponding to x and original solution y.* $h = 1/128$, $\phi_R = 1.15$. *1st subinterval:* $[0.5, 1, 5]$ *and 2nd subinterval:* $[1.0, 1, 3]$, $\varepsilon_3 = 10^{-4}$.



[8] J. DENNIS AND R. SCHNABEL, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, SIAM, Philadelphia, PA, 1996.

[9] S. C. EISENSTAT AND H. F. WALKER, *Choosing the forcing terms in an inexact Newton method*, SIAM J. Sci. Comput., 17 (1996), pp. 16-32.

[10] F.-N. HWANG AND X.-C. CAI, *Improving robustness and parallel scalability of Newton method through nonlinear preconditioning*, in Domain Decomposition Methods in Science and Engineering, R. Kornhuber, R. Hoppe, D.E. Keyes, J. Periaux, O. Pironneau, and J. Xu eds., Lecture Notes in Computational Science and Engineering, 40, Springer-Verlag, Heidelberg, 2004, pp. 201-208.

[11] F.-N. HWANG, X.-C. CAI, *A parallel nonlinear additive Schwarz preconditioned inexact Newton algorithm for incompressible Navier-Stokes equations*, J. Comput. Phy., 204 (2005), pp.

666-691.

[12] F.-N. Hwang and X.-C. Cai, *A class of parallel two-level nonlinear Schwarz preconditioned inexact Newton algorithms*, Comput. Methods Appl. Mech. Engrg., 196 (2007), pp. 1603-1611.

[13] P. J. Lanzkron, D. J. Rose, and J. T. Wilkes, *An analysis of approximate nonlinear elimination*, SIAM, J. Sci. Comput., 17 (1996), pp. 538-559.

[14] D. P. Young, R. G. Melvin, M. B. Bieterman, F. T. Johnson, S. S. Samant, and J. E. Bussoletti, *A locally refined rectangular grid finite element methods: Application to computational fluid dynamics and computational physics*, J. Comput. Phys., 92 (1991), pp. 1-66.

[15] Y. Saad and M. H. Schultz, *GMRES: A generalized minimal residual algorithm for solving nonsysmetric linear systems*, SIAM J. Sci. Stat. Comp., 7 (1986), pp. 856-869.

[16] D. P. Young, W. P. Huffman, R. G. Melvin, C. L. Hilmes, and F. T. Johnson, *Nonlinear elimination in aerodynamic analysis and design optimization*, in Large-Scale PDE-Constrained Optimization, L. T. Biegler, O. Ghattas, M. Heinkenschloss, and B. van Bloemen Waanders, Lect. Notes in Comp. Sci., 30, Springer-Verlag, New York, 2003, pp. 17-44.