Nonlinear Preconditioning and its Application in Multicomponent Problems

Thesis by

Lulu Liu

In Partial Fulfillment of the Requirements

For the Degree of

Doctor of Philosophy

King Abdullah University of Science and Technology, Thuwal,

Kingdom of Saudi Arabia

December, 2015

The thesis of Lulu Liu is approved by the examination committee

Committee Chairperson: David Keyes

Committee Member: Shuyu Sun

Committee Member: Ravi Samtaney

Committee Member: Rolf Krause

ABSTRACT

Nonlinear Preconditioning and its Application in Multicomponent Problems

Lulu Liu

The Multiplicative Schwarz Preconditioned Inexact Newton (MSPIN) algorithm is presented as a complement to Additive Schwarz Preconditioned Inexact Newton (ASPIN). At an algebraic level, ASPIN and MSPIN are variants of the same strategy to improve the convergence of systems with unbalanced nonlinearities; however, they have natural complementarity in practice. MSPIN is naturally based on partitioning of degrees of freedom in a nonlinear PDE system by field type rather than by subdomain, where a modest factor of concurrency can be sacrificed for physically motivated convergence robustness. ASPIN, originally introduced for decompositions into subdomains, is natural for high concurrency and reduction of global synchronization.

The ASPIN framework, as an option for the outermost solver, successfully handles strong nonlinearities in computational fluid dynamics, but is barely explored for the highly nonlinear models of complex multiphase flow with capillarity, heterogeneity, and complex geometry. In this dissertation, the fully implicit ASPIN method is demonstrated for a finite volume discretization based on incompressible two-phase reservoir simulators in the presence of capillary forces and gravity. Numerical experiments show that the number of global nonlinear iterations is not only scalable with respect to the number of processors, but also significantly reduced compared with

the standard inexact Newton method with a backtracking technique. Moreover, the ASPIN method, in contrast with the IMPES method, saves overall execution time because of the savings in timestep size.

We consider the additive and multiplicative types of inexact Newton algorithms in the field-split context, and we augment the classical convergence theory of ASPIN for the multiplicative case. Moreover, we provide the convergence analysis of the MSPIN algorithm. Under suitable assumptions, it is shown that MSPIN is locally convergent, and desired superlinear or even quadratic convergence can be obtained when the forcing terms are picked suitably. Numerical experiments show that MSPIN can be significantly more robust than Newton methods based on global linearizations, and that MSPIN can be more robust than ASPIN, and maintain fast convergence even for challenging problems, such as high-Reynolds number Navier-Stokes equations.

ACKNOWLEDGEMENTS

First and foremost, I would like to express my deepest thanks and appreciation to my advisor, Prof. David Keyes, for his support and guidance on my research. I am grateful to him for his patience, encouragement and suggestions throughout my PhD studies. His enthusiasm and diligence as a scientific researcher inspire me a lot to do the research.

The research in my thesis is supported by KAUST's baseline funding and Extreme Computing Research Center, which provided me opportunities to attend several international conferences. The PETSc group of Argonne National Laboratory is gratefully acknowledged for their software support.

Finally, I would like to thank my family for their love, support and encouragement.

TABLE OF CONTENTS

LIST OF ABBREVIATIONS

ASPIN	Additive Schwarz Preconditioned Inexact Newton
IMPES	IMplicit Pressure Explicit Saturation
INB	Inexact Newton method with Backtracking
MSPIN	Multiplicative Schwarz Preconditioned Inexact Newton
PV	Pore Volume
PVI	Pore Volume Injected

LIST OF FIGURES

LIST OF TABLES

Chapter 1

Introduction

1.1 An Overview of Nonlinear Solvers

In contemporary simulations, many problems are governed by nonlinear partial differential equations (PDEs), which result in large scale algebraic systems using discretization techniques such as the finite difference method, the finite volume method, or the finite element method.

There are several important categories of solvers for discretized nonlinear PDEs:

- Newton-type methods: Starting from a global linearization by Newton's method or some inexact variants, we solve the resulting linear system by a linear solver such as SOR, multigrid, or Krylov subspace methods. It is not necessary to explicitly form or store the matrix of the linear system. For example, various "matrix-free" approaches can be used to construct the Jacobian-vector product approximation in Newton-Krylov methods, namely Jacobian-free Newton-Krylov methods [1]. There is no guarantee that Newton-type methods can converge when an initial guess is far from the exact solution, and therefore they are often augmented by popular globalization techniques such as line search or trust-region methods [2].

- Nonlinear Krylov methods: These include nonlinear Richardson [3], nonlinear

conjugate gradients (NCG) [4], nonlinear GMRES (NGMRES) [5] and Anderson mixing methods [6, 7]. The nonlinear Richardson iteration and NCG can be viewed as a simple extension of corresponding methods in linear cases, the Richardson iteration and CG, but the associated step length is determined by line search methods. NGMRES and Anderson mixing combine several previous iterations and solve residual minimizing problems for weights. The updated solution is then constructed as the combination of previous solutions and residuals using those weights.

- Decomposition methods: We consider nonlinear variants of subspace or domain decomposition and multilevel algorithms, which often serve as nonlinear preconditioners or accelerators due to no guarantee of convergence. Here we consider three nonlinear algorithms: Gauss-Seidel-Newton algorithm solves block subproblems using Newton's method in the multiplicative form; nonlinear additive Schwarz method (NASM) [8] solves local subdomain problems in parallel and sum resulting corrections into a global search direction; Full approximation scheme (FAS) [9] solves coarse-grid problems in the error-smoothing process, and then the coarse-grid error extracted from the solution is interpolated to the fine grid. Finally, the fine-grid error is used to update the current fine-grid approximation.

- Limited-memory quasi-Newton methods: Instead of computing Jacobian directly, the class of methods form an approximation of Jacobian or the inverse Jacobian by storing low rank updates. Three popular variants are L-BFGS [10], "good" Broyden, and "bad" Broyden [11, 12].

In addition to above methods, there are many related variants or other globalization techniques that will be discussed in the next section.

1.2 Nonlinear Preconditioning

1.2.1 Background

Newton-like methods in their many variants are often favored for the solution of nonlinear systems, especially large algebraic systems arising from discretized differential equations. For instance, Newton-Krylov-Schwarz methods have been successfully applied in many areas [1, 13, 14, 15, 16]. However, the problem of "nonlinear stiffness" frequently arises, in which progress in updating the state variables with a global Newton step is retarded by an often small subset of the variable-equation pairs due to the lack of a good starting point. Global Newton-like methods may waste considerable computational resources while the majority of state variables barely evolve for many iterations until some critical feature (e.g., boundary or interior layer, aerodynamic shock, reaction zone, contact discontinuity, phase transition) of the solution falls into place, following which the desired superlinear asymptotic convergence of Newton polishes the root. These unproductive iterations typically require the solution of large linear systems with ill-conditioned Jacobians. Worse, Newton may stagnate indefinitely.

As in the examples of [2, 17], the domain of convergence of Newton's method may shrink as the relative nonlinearity of the function increases. Therefore, it is difficult to find a good initial guess for problems with "unbalanced nonlinearities." Some globalization techiniques have been developed to bring the solution into the domain of convergence from a convenient initial point, such as the line search method and the trust region method [2, 11, 18, 19], mesh sequencing methods [20, 21], pseudo-transient continuation methods [22, 23]. Two complementary approches to handle strong nonlinearities are nonlinear elimination [24, 25, 26, 27] and nonlinear preconditioning [1, 28, 29]. Nonlinear elimination implicitly eliminates local high nonlinearities by removing appropriate variables deemed to cause trouble before applying

a global Newton iteration. Nonlinear preconditioning reduces nonlinearities of the function by changing coordinates, i.e., to solve the preconditioned system instead of the original system by an outer Jacobian-free Newton method. The dissertation focuses on the nonlinear preconditiong technique.

1.2.2 Development of Nonlinear Preconditioning

As a result of nonlinearities of the problem, the solution structure has disparate spatial scales, namely "nonlinear stiffness" or "unbalanced nonlinearities." Classical examples are reaction fronts and shock waves. For the nonlinearly stiff problem, Newton's method is often frustrated due to a considerable number of unproductive iterations or even stagnation.

Nonlinear Schwarz algorithms have been widely used as iterative methods [30, 31, 32, 33], and numerical experiments show that they are generally not very robust, except for monotone problems. In fact, however, the nonlinear Schwarz can serve as an excellent nonlinear preconditioner. To conquer unbalanced nonlinearities and improve global convergence properties, the additive Schwarz preconditioned inexact Newton (ASPIN) algorithm was devised in [28] as an inner-outer Newton solver, with most of the work performed on independent subproblems in inner iterations, plus relatively few outer iterations on a transformed system, on which Newton is supposed to converge quickly. The key idea of ASPIN is to transform the original system into a modified system with the same root, and to solve it using a Jacobian-free [1] inexact Newton method.

ASPIN has virtues beyond potentially more robust nonlinear convergence:

– The inner nonlinear subproblems possess smaller working sets and, being less nonlinearly stiff, may require in aggregate less work than the outer iterations they replace, resulting in a net reduction of execution time.

– The auxiliary inner iteration linear problems are smaller than the global system, so the sets of processors that need to synchronize frequently (e.g., because of inner products or recurrences) to carry out their solution are smaller. Furthermore, these sets may be asynchronous relative to each other.

– Since the outer iteration is Jacobian-free and the inner iterations work on subsets of the original system, ASPIN can be implemented with modest additions to an existing Newton code, and it is available in the popular PETSc framework [34].

So far, ASPIN has been employed successfully to solve some challenging problems, such as a lid-driven cavity flow [17, 28, 35], a transonic full potential flow [36], incompressible Navier-Stokes flows at high Reynolds numbers [37, 38, 39], and two-phase flow in porous media [40]. Various enhancements of ASPIN have been proposed, including two-level versions [17, 35, 39, 41, 42] to improve the conditioning of the outer Jacobian-free linear problem, the extension to unstructured finite element elliptic problems [43], and the incorporation of a rescaling step [37]. For a taxonomy of nonlinear preconditioning methods placing ASPIN in a broader context of scalable nonlinear solvers, see [3].

Theoretical support for the local convergence of the ASPIN algorithm at up to a quadratic rate, for suitable assumptions, was provided in [44] and the results are similar to those obtained for the inexact Newton method. Also, some preliminary convergence analysis [45] of ASPIN was studied for a class of semilinear elliptic PDEs. It is quite difficult to prove global convergence of the ASPIN method, by contrast, a novel "G-ASPIN" (Globalized ASPIN) method employing Trust-Region control strategies for nonlinear programming problems was presented in [46], which is globally convergent. The key idea of this globalization approach is to consider the nonlinearly preconditioned Newton step of ASPIN as the first-order conditions of a particular preconditioned quadratic programming problem. For nonlinear programming, some additive and multiplicative preconditioned trust-region approaches regarded as right

nonlinear preconditioners were presented in [47, 48].

1.3 Motivation and Objectives

As far as we know, there are very few attempts [40] to apply ASPIN to reservoir simulation problems until now, although ASPIN successfully handles high nonlinearities for challenging computational fluid dynamics problems. One objective of this thesis is to apply ASPIN in simulating more general two-phase flow involving the terms of capillary pressure and gravity [49].

The classical ASPIN was introduced in [28] as a nonlinear domain decomposition method; however, its algebraic generalization to other types of decompositions was anticipated. A second objective is to develop an algebraic variant of ASPIN based on field splitting. Under field splitting, a Gauss-Seidel-like sequential ordering of the subproblems is often natural, whereas domain decomposition naturally generates concurrent problems of Jacobi character. The Gauss-Seidel-like variant of ASPIN is referred to as the multiplicative Schwarz preconditioned inexact Newton (MSPIN) algorithm [50].

A third objective is to carry out convergence analysis of the MSPIN algorithm [51], which provides the theoretical support for the multiplicative version of nonlinear preconditioning.

1.4 Organization of Dissertation

The dissertation consists of six chapters. The main contribution is presented in Chapter 4 through Chapter 6.

The next chapter gives some preliminary knowledge, including some details of inexact Newton method with backtracking (INB), the definition of F-differentiability, and a brief overview of the software framwork PETSc, in which our examples are

implemented.

In Chapter 3, we review the algorithmic and theoretical framework of ASPIN.

In Chapter 4, we describe the incompressible two-phase flow model in a two-dimensional porous medium and the fully implicit numerical scheme. In addition, some numerical results using ASPIN and concluding remarks are given.

In Chapter 5, we introduce two types of the nonlinear field-split preconditioned inexact Newton algorithm and derive formulae for preconditioned functions and corresponding Jacobian matrices, and then present details of the field-split preconditioned inexact Newton algorithm implementation, as well as an illustrative example of two scalar components. This chapter also contains a proof of the equivalence of the original nonlinear system and the nonlinearly preconditioned system under some reasonable conditions for applications, and the extension in the case of $N > 2$ components. Finally, some numerical results are given.

Chapter 6 briefly reviews the MSPIN algorithm corresponding to $N \geqslant 2$ components and its properties. A proof of local convergence of MSPIN is given, and desired superlinear or even quadratic convergence can be obtained by choosing suitable forcing terms. Next, a simple example illustrates that some conditions in the assumptions are not unreasonably strict.

We conclude with the niche for the field-split preconditioned inexact Newton algorithm variants and some natural future work in Chapter 7.

Chapter 2

Preliminaries

2.1 Inexact Newton with Backtracking (INB)

We briefly review the classical Inexact Newton method with Backtracking (INB) technique [2, 52, 53], which serves as the basic block of both ASPIN and MSPIN. The INB method is used both as a nonlinear solver for the transformed global system and for subsystems in the original coordinates.

Considering the discrete nonlinear function $F : R^n \to R^n$, we want to find a vector $x^* \in R^n$ such that

$$F(x^*) = 0, \tag{2.1}$$

where $F = [F_1, F_2, \ldots, F_n]^T$ and $x = [x_1, x_2, \ldots, x_n]^T$.

The fundamental form of Newton's method for solving $F(x) = 0$ is given in Algorithm 1, which may also be viewed as the two-term multivariate Taylor expansion of the function $F(x)$ with respect to the current approximation $x^{(k)}$. Some convergence analysis including the Kantorovich analysis for the Newton's method are found in [2, 54]. At each step, however, it is generally very expensive to solve Newton equations for the exact solution using the direct solvers if the number of unknowns is large and, moreover, it may not be justified when $x^{(k)}$ is far from the exact solution x^*. Therefore, it is reasonable to employ iterative methods to solve Newton equa-

Algorithm 1 Newton's method

An initial iterate $x^{(0)}$ is given.
for $k = 0, 1, 2, \ldots$ until convergence **do**
  Solve $F'(x^{(k)})d^{(k)} = F(x^{(k)})$
  Set $x^{(k+1)} = x^{(k)} - d^{(k)}$
end for

Algorithm 2 IN (Inexact Newton method)

An initial iterate $x^{(0)}$ is given.
for $k = 0, 1, 2, \ldots$ until convergence **do**
  Choose $\eta_k \in [0, 1)$
  Find $d^{(k)}$ which satisfies
    $\|F(x^{(k)}) - F'(x^{(k)})d^{(k)}\| \leq \eta_k \|F(x^{(k)})\|$.
  Set $x^{(k+1)} = x^{(k)} - d^{(k)}$
end for

tions approximately for an inexact Newton direction (a trade-off between accuracy and amount of work per iteration), referred to as the inexact Newton (IN) method [2, 11, 52, 55]. The framework is described in Algorithm 2. Here η_k, called a "forcing term," determines how accurately the Jacobian linear system needs to be solved using Krylov subspace methods [56, 57], such as GMRES [58, 59]. It is noted that Algorithm 2 gives Newton's method when $\eta_k = 0$.

When the approximation $x^{(k)}$ is sufficiently close to x^*, it can be shown that the inexact Newton method is locally convergent under some reasonable assumptions. The convergence is q-superlinear if $\eta_k \to 0$, or q-quadratic if $\eta_k = \mathcal{O}(\|F(x^{(k)})\|)$ and F' is Lipschitz continuous [11, 55]. Some choices of η_k based on norms or updates and residuals available as by-products of the main computation have been suggested [60, 61, 62]. In order to achieve desirably fast local convergence and tend to avoid oversolving the Newton equation, two common choices of forcing terms were proposed by Eisenstat and Walker [62]:

- Choice 1. $\eta_0 \in [0, 1)$ is given and choose

$$\eta_k = \frac{\left\|F(x^{(k)}) - F(x^{(k-1)}) - F'(x^{(k-1)})d^{(k-1)}\right\|_2}{\|F(x^{(k-1)})\|_2}, \quad k = 1, 2, \ldots, \tag{2.2}$$

or

$$\eta_k = \frac{\left| \|F(x^{(k)})\|_2 - \|F(x^{(k-1)}) + F'(x^{(k-1)})d^{(k-1)}\|_2 \right|}{\|F(x^{(k-1)})\|_2}, \quad k = 1, 2, \ldots. \tag{2.3}$$

- Choice 2. Given $\gamma \in [0, 1]$, and $\sigma \in (1, 2]$, and $\eta_0 \in [0, 1)$, choose

$$\eta_k = \gamma \left(\frac{\|F(x^{(k)})\|_2}{\|F(x^{(k-1)})\|_2} \right)^{\sigma}, \quad k = 1, 2, \ldots. \tag{2.4}$$

To avoid η_k becoming too small, two corresponding safeguards are also needed:

- For Choice 1, modify η_k by $\eta_k \leftarrow \max\{\eta_k, \eta_{k-1}^{\frac{1+\sqrt{5}}{2}}\}$, if $\eta_{k-1}^{\frac{1+\sqrt{5}}{2}} > 0.1$.
- For Choice 2, modify η_k by $\eta_k \leftarrow \max\{\eta_k, \gamma\eta_{k-1}^{\sigma}\}$, if $\gamma\eta_{k-1}^{\sigma} > 0.1$.

For Choice 1, η_k in (2.2) or (2.3) gives a certain q-superlinear local convergence. By contrast, Choice 2 does not directly reflect the agreement between F and its local linear model at the previous step, as does Choice 1. However, it is effective against oversolving the Newton equation and offers up to q-quadratic local convergence. Moreover, Choice 2 may be useful in applications due to flexibility allowed by two parameters γ and σ in (2.4).

Unfortunately, the inexact Newton method may not converge if the initial guess is not close to x^*, and then one has to combine the inexact Newton method with the globalization strategies such as line search, trust-region or hybrid steepest descent [2, 11, 18, 19, 63, 64] techniques in order to ensure the convergence.

The idea of the line search algorithm is to find an acceptable $x^{(k+1)}$ based on the function $f(x) = \frac{1}{2}\|F(x)\|^2$ by shortening the step length along the descent direction $d^{(k)}$. Here we also list the details of the backtracking line search algorithm in Algorithm 3. We first try the full Newton step by setting $\lambda^{(k)} = 1$. If $x^{(k+1)} = x^{(k)} - d^{(k)}$ is not acceptable, we could backtrack (reduce $\lambda^{(k)}$) until $\lambda^{(k)}$ is found such that

Algorithm 3 Backtracking line search

Input: $x^{(k)}$ and $d^{(k)}$.
Define $f(x) = \frac{1}{2}\|F(x)\|^2$. Give $\alpha \in (0,1)$ and $0 < \theta_{min} < \theta_{max} < 1$.
Let $\lambda^{(k)} = 1$.
while $f(x^{(k)} - \lambda^{(k)} d^{(k)}) > f(x^{(k)}) - \alpha\lambda^{(k)} \nabla f(x^{(k)})^T d^{(k)}$ **do**
  choose some $\theta \in [\theta_{min}, \theta_{max}]$
  $\lambda^{(k)} = \theta\lambda^{(k)}$
end while

Algorithm 4 INB (Inexact Newton backtracking algorithm)

An initial iterate $x^{(0)}$ is given.
for $k = 0, 1, 2, \ldots$ until convergence **do**
  Choose $\eta_k \in [0, 1)$
  Find $d^{(k)}$ such that
    $\|F(x^{(k)}) - F'(x^{(k)})d^{(k)}\| \leq \eta_k \|F(x^{(k)})\|$.
  Determine a step size $\lambda^{(k)}$ using the backtracking line search technique.
  Set $x^{(k+1)} = x^{(k)} - \lambda^{(k)} d^{(k)}$.
end for

$x^{(k+1)} = x^{(k)} - \lambda^{(k)} d^{(k)}$ is acceptable. In Algorithm 3, the new step length is determined by constructing a one-dimensional quadratic or cubic function to model $f(x^{(k)} - \lambda^{(k)} d^{(k)})$, namely the quadratic or cubic line search methods. For more details, see [2].

Finally, we combine the inexact Newton method with a backtracking linear search technique, leading to the globally inexact Newton backtracking (INB) algorithm shown in Algorithm 4.

2.2 Differentiability

Let $\mathcal{L}(R^n, R^m)$ denote the linear space of real $m \times n$ matrices.

Definition 1. *(Definition 3.1.1 in [54]) A mapping $H : D \subset R^n \to R^m$ is* ***Gateaux- (or G-) differentiable*** *at an interior point x of D if there exists a linear operator*

$A \in \mathcal{L}(R^n, R^m)$ *such that, for any* $h \in R^n$,

$$\lim_{t \to 0}(1/t)\|F(x+th) - F(x) - tAh\| = 0. \tag{2.5}$$

Definition 2. *(Definition 3.1.5 in [54]) The mapping* $H : D \subset R^n \to R^m$ *is* ***Fréchet-(or F-) differentiable*** *at an interior point* x *of* D *if there is an* $A \in \mathcal{L}(R^n, R^m)$ *such that*

$$\lim_{h \to 0}(1/\|h\|)\|F(x+h) - F(x) - Ah\| = 0. \tag{2.6}$$

The linear operator A *is denoted by* $F'(x)$*, and is called* ***F-derivative*** *of* F *at* x*.*

Note that the mapping H is G-differentiable at x whenever it is F-differentiable at x, and it follows that any property of the G-derivative also holds for the F-derivative.

2.3 PETSc

PETSc [34], standing for the Portable, Extensible Toolkit for Scientific Computation, is programmed in C to build data structures and provide routines for the scalable solutions of PDEs deriving from the scientific applications such as fluid dynamics, biology, fusion, geosciences, and nano-science. It is developed as an open-source software package at Argonne National Laboratory, which supports MPI, GPUs through CUDA or OpenCL, and hybrid MPI-GPU parallelism. PETSc is available and tutorials can be downloaded via `http://www.mcs.anl.gov/petsc`.

One attractive feature of PETSc is to include a large suite of parallel linear, nonlinear and timestepping solvers, which can be easily used in scientific codes written in C, C++, Fortran and Python.

- Linear solvers: The combination of a Krylov subspace method (KSP) and a preconditioner (PC) is often an efficient iterative solver for linear systems. In Table 2.1, we summarize the basic Krylov subspace methods and preconditioning

methods supported in PETSc. In addition, PETSc also provides physics-based block preconditioners (PCFIELDSPLIT), and both matrix-free and matrix-based multigrid preconditioners are fully supported. The shell preconditioner (PCSHELL) uses an application-provided routine with its own private data storage. The PC type, namely combination of preconditioners (PCCOMPOSITE), allows one to construct a composible preconditioner by combining already defined preconditioners and solvers. Complex multilevel or multistage preconditioners can be constructed due to flexibility of PCSHELL and PCCOMPOSITE preconditioners.

- Nonlinear solvers (SNES): PETSc includes Newton-like nonlinear solvers based on line search techniques and trust region methods. It also provides an interface to nonlinear Krylov methods, quasi-Newton methods, the full approximation scheme (FAS), nonlinear additive Schwarz methods and nonlinear preconditioners. When solving a nonlinear system, users must set a routine for evaluating the nonlinear function. The approximation of the corresponding Jacobian can be computed using user-provided analytical formula, multicolored finite differencing [65], "matrix-free" Jacobian-vector product approximation [1] or automatic differentiation using ADIC [66] or ADIFOR [67]. In addition, PETSc also includes capabilities for variational inequalities (VIs) with box constraints, which allows problems with bounds on solution such as phase-field models to be handled properly without resorting to "cut-off" or other techniques.

- ODE and DAE solvers (TS): PETSc provides the library of time-stepping ODE solvers, namely TS, for ODEs and DAEs arising from the discretization of time-dependent PDEs, including fully implicit general linear methods (GL) and theta methods, IMEX (semi-implicit) methods and the explicit Euler and Runge-Kutta methods with variable timesteps. TS also can be used to solve steady-

KSP	PC
Richardson	Jacobi
Chebyshev	Block Jacobi
Conjugate Gradient	SOR (and SSOR)
BiConjugate Gradient	SOR with Eisenstat trick
Generalized Minimal Residual	Incomplete LU
Flexible Generalized Minimal Residual	Additive Schwarz
Deflated Generalized Minimal Residual	Generalized Additive Schwarz
Generalized Conjugate Residual	Algebraic Multigrid
BiCGSTAB	BDDC
Conjugate Gradient Squared	Linear solver
Transpose-Free QMR	Combination of preconditioners
MINRES	LU
Conjugate Residual	Cholesky
Least Squares Method	Shell for user-defined PC

Table 2.1: Krylov subspace methods and preconditioners.

state problems with a pseudo-time stepping approach [22, 23].

Another attractive feature is that PETSc has interfaces to various external packages like SuperLU, SuperLU_DIST, SUNDIALS, MATLAB, Mathematica, Hypre, MUMPS, ParMeTis, Trilinos, and so on. For example, the package SuperLU_DIST extends PETSc with the parallel LU decomposition, and the TS library in PETSc allows one to use an external parallel ODE solver, the CVODE component of SUNDIALS.

PETSc is one of the world's most widely used libraries for high-performance computational science. PETSc and its developers have won many prizes. For example, the core development group won the the SIAM/ACM Prize in Computational Science and Engineering for 2015. PETSc has been employed in many notable simulations such as the Gordon Bell Finalist at SC 2009, and the Gordon Bell Special Prize at SC 1999, SC 2003 and SC 2004.

Since it leverages much relevant open-source software and since it gives our own developments a recognized pathway for distribution, in this dissertation, all of codes for numerical examples are implemented in PETSc.

Chapter 3

Additive Schwarz Preconditioned Inexact Newton (ASPIN)

3.1 Linear and Nonlinear Preconditioning

For ill-conditioned linear systems arising from typical applications such as fluid dynamics, iterative methods, such as Krylov subspace methods, are often favored due to their intrinsic appeal of requiring only "blackbox" matrix-vector multiplies with the true Jacobian, while being preconditioned with related simplified discrete operators. However, iterative solvers allowed only capped memory resources may suffer from slow convergence or even divergence. The technique of linear preconditioning is successful to improve both the efficiency and robustness of iterative methods.

We want to find the solution to the the original (ill-conditioned) system

$$Ax - b = 0, \tag{3.1}$$

where $A \in R^{n \times n}$ and $x,\ b \in R^n$. The idea underlying linear preconditioning is to replace (3.1) with an equivalent system

$$M^{-1}(Ax - b) = 0, \tag{3.2}$$

which is then solved by an iterative method. Here the nonsingular matrix M (or sometimes M^{-1}) is called a preconditioner. There are many ways to define the preconditioner M, but it should satisfy the following requirements:

- It is inexpensive to solve linear systems with the operator M.
- The operator M is close to A in the sense that (3.2) is better conditioned.

The preconditioning technique can be generalized to solve nonlinear systems with strong nonlinearities, which is referred to as nonlinear preconditioning. We solve the preconditioned system $\mathcal{F}(x) = H(F(x)) = 0$ instead of the original system $F(x) = 0$, where H is viewed as a nonlinear preconditioner. Similarly, there are some requirements for H:

- If $H(w) = 0$, then $w = 0$. This implies that $\mathcal{F}(x^*) = 0$ if $F(x^*) = 0$.
- H is close to F^{-1} in some sense such that $\mathcal{F}(x) = 0$ is better balanced.
- $H(F(y))$ is easily computable for given $y \in R^n$.
- It is easy to approximate the matrix-vector multiplication $(H(F(y))'z$ for $y,\ z \in R^n$ when $\mathcal{F}(x) = 0$ is solved by a Newton-Krylov method.

The nonlinear preconditioner H is defined implicitly and it is not necessary to specify its form. In the next section, we will show how to define $\mathcal{F}(x)$ directly without reference to H.

3.2 Review of ASPIN

3.2.1 Framework

The Additive Schwarz Preconditioned Inexact Newton (ASPIN) was proposed by Cai and Keyes [28], and some numerical experiments [17, 28, 35, 36, 37] show that

this algorithm works well for some problems in computational fluid dynamics. Tests for the standard driven cavity flow problem, show that ASPIN still converges and maintains a superlinear Newton-like fast convergence rate for a much larger range of Reynolds numbers than the Newton-Krylov-Schwarz algorithm, which stagnates at a large Reynolds number ($\mathcal{O}(10^3)$).

In this section, we quickly review the outline of ASPIN framework based on domain decomposition. We consider a given nonlinear system

$$F(x) = 0. \tag{3.3}$$

Let

$$\Omega = \bigcup_{i=1}^{N} \Omega_i \tag{3.4}$$

be a domain partition, where subdomains Ω_i may overlap each other. The associated nonlinearly preconditioned system

$$\mathcal{F}(x) = 0 \tag{3.5}$$

is constructed in the following way:

$$\mathcal{F}(x) = \sum_{i=1}^{N} T_{\Omega_i}(x), \qquad \Omega_i \in \Omega, \tag{3.6}$$

where $T_{\Omega_i}(x)$ is obtained by solving local nonlinear systems

$$F_{\Omega_i}(x - T_{\Omega_i}(x)) = 0, \quad i = 1, \ldots, N. \tag{3.7}$$

The nonlinearly transformed version (3.5) of the original system is solved using an inexact Newton method, which requires the Jacobian of the preconditioned system

$\mathcal{J}(x) = \mathcal{F}'(x)$. Following [28, 44], the Jacobian $\mathcal{J}(x)$ can be written in the form

$$\mathcal{J}(x) = \sum_{i=1}^{N} T'_{\Omega_i}(x) = \sum_{i=1}^{N} R_i^T [R_i J(x - T_{\Omega_i}(x)) R_i^T]^{-1} R_i J(x - T_{\Omega_i}(x)), \tag{3.8}$$

where $J(x) = F'(x)$ is the Jacobian of $F(x)$ and R_i is the restriction operator such that $F_{\Omega_i}(x) = R_i F(x)$, $i = 1, 2, \ldots, N$. It is very expensive and inconvenient to directly calculate the Jacobian in (3.8), since $T'_{\Omega_i}(x)$ is computed by different arguments, $i = 1, 2, \ldots, N$. In practice, it is recommended in [28] to use the following approximation

$$\begin{aligned} \hat{\mathcal{J}} &= \sum_{i=1}^{N} R_i^T [R_i J(x) R_i^T]^{-1} R_i J(x) \\ &= \sum_{i=1}^{N} J_{\Omega_i}^{-1} J, \end{aligned}$$

where $J_{\Omega_i}^{-1} = R_i^T [R_i J(x) R_i^T]^{-1} R_i$ and $J = J(x)$.

We describe the ASPIN algorithm in **Algorithm 5**. In our work, we solve the local nonlinear systems (3.7) using Newton's method, and approximate the Jacobian J_{Ω_i} by a multicolored finite difference scheme, instead of analytical subdomain Jacobian matrices. For each local nonlinear iteration, LU decomposition is used for solving the local Jacobian system. To achieve an efficient implementation, subdomain problems are solved in parallel. We employ an inexact Newton method with a backtracking technique as the global nonlinear solver, and no preconditioning is done for the global Jacobian system (3.12). The standard cubic backtracking algorithm is used to pick $\lambda^{(k)}$ such that

$$f(x^{(k)} - \lambda^{(k)} d^{(k)}) \leqslant f(x^{(k)}) - \alpha \lambda^{(k)} (g^{(k)})^T \sum_{i=1}^{N} J_{\Omega_i}^{-1} J d^{(k)}, \quad f(x) = \frac{1}{2} \|\mathcal{F}(x)\|^2, \tag{3.9}$$

where the parameter associated with backtracking is set to $\alpha = 10^{-4}$.

Algorithm 5 ASPIN

Let $\alpha \in (0,1)$ and $0 < \theta_{min} < \theta_{max} < 1$ be given.

1. Compute the nonlinear residual $g^{(k)} = \mathcal{F}(x^{(k)})$ through the following steps.

   (a) Starting from $g_i^{(k)} = 0$, find $g_i^{(k)} = T_{\Omega_i}(x^{(k)})$ by solving local nonlinear systems
   $$F_{\Omega_i}(x^{(k)} - g_i^{(k)}) = 0, \quad , i = 1, \ldots, N. \tag{3.10}$$

   (b) Form the global residual
   $$\mathcal{F}(x^{(k)}) = \sum_{i=1}^{N} g_i^{(k)}. \tag{3.11}$$

   (c) Check the stopping conditions on $\mathcal{F}(x^{(k)})$.

2. Find the inexact Newton direction $d^{(k)}$ by solving the Jacobian system approximately,
   $$\sum_{i=1}^{N} J_{\Omega_i}^{-1} J d^{(k)} = g^{(k)}, \tag{3.12}$$
   such that
   $$\|g^{(k)} - \sum_{i=1}^{N} J_{\Omega_i}^{-1} J d^{(k)}\| \leq \eta_k \|g^{(k)}\| \tag{3.13}$$
   where $\eta_k \in [0, \eta_{max}]$ is a forcing term.

3. Compute the new approximate solution
   $$x^{(k+1)} = x^{(k)} - \lambda^{(k)} d^{(k)}, \tag{3.14}$$
   where the step length $\lambda^{(k)}$ is determined by performing a line search along $d^{(k)}$.

3.2.2 Some Analysis

In this section, we list some important results of the ASPIN algorithm. The preconditioned function $\mathcal{F}(x)$ is defined in (3.6) and $f(x) = \frac{1}{2}\|\mathcal{F}(x)\|^2$.

Assumption 3. *([28]) $F'(x)$ is continuous in a neighborhood D of the exact solution x^*. $F'(x^*)$ and $R_i F'(x^*) R_i^T$ are nonsingular, $i = 1, \ldots, N$.*

Theorem 4. *([28]) Under Assumption 3, $F(x)$ and $\mathcal{F}(x)$ have the same solution in*

a neighborhood of the exact solution u^* in D.

Theorem 5. *([44]) Let $F(x)$ and $f(x)$ be twice continuously differentiable in $N(x^*, r)$, and there exists a constant $C > 0$ such that*

$$\|\nabla f(x) - \nabla f(y)\| \leqslant C\|x - y\|, \quad \|\nabla^2 f(x) - \nabla^2 f(y)\| \leqslant C\|x - y\|, \tag{3.15}$$

for any $x, y \in N(x^*, r)$. Consider a sequence $\{x^{(k)}\}$ generated by the ASPIN method such that $x^{(k)} \to x^*$, and then

(i) If $\eta_k \to 0$, $x^{(k)} \to x^*$ superlinearly ;

(ii) If $\eta_k = \mathcal{O}(\|\mathcal{F}(x^{(k)})\|)$, $x^{(k)} \to x^*$ quadratically .

For further analysis and a convergence proof, see [28, 44, 45].

Chapter 4

Numerical Simulation of the Two-phase Flow with ASPIN

Multiphase flow behavior in porous media has been intensively studied over the past few decades, due to its importance in a variety of disciplines such as oil and gas recovery [68, 69, 70], groundwater contamination [71, 72], and basin modeling [73]. Such flows are complicated by various modeling features, including heterogeneity of absolute permeability, gravity and capillary pressure effects, and relative permeability functions. For reservoir simulators, there are various solution techniques with different degrees of coupling in the nonlinear solver and implicitness in the discretization.

The IMplicit Pressure Explicit Saturation (IMPES) method and its variations (e.g., the improved IMPES) [74] remain popular schemes for solving multiphase flow equations. The numerical stability of IMPES is limited not only by the CFL condition for the saturation equation, but also more seriously by the splitting error from decoupling pressure and saturation equations. Splitting degrades its performance, especially with strong nonlinearities from relative permeability and capillarity. To maintain robustness of IMPES, including physically feasible saturation fractions between 0 and 1, the time step of the IMPES scheme must often be reduced below what may be required for temporal accuracy. An alternative is to solve the multiphase flow system fully implicitly, which allows a large timestep size, but this approach requires

a nonlinear solver at each time step. It takes a considerable amount of simulation time to solve large nonlinear systems using Newton's method or its many variations, especially as an increasing number of unknowns for a high resolution to the fine scale variations. Efforts have been devoted to developing techniques to reduce the number of Newton iterations and execution time to solve linear systems. For instance, a reordering procedure [75] is used to speed up the solution of the nonlinear system derived from two-phase and three-phase flow.

In last few years, Krylov subspace methods like GMRES and BiCGSTAB as inner solvers for inexact Newton methods have been successfully employed in multiphase flow simulators, and preconditioners are usually required to avoid breakdowns or unaffordably slow convergence. To accelerate linear solvers, the two-stage preconditioners [76, 77] have better performance than traditional preconditioners, such as block Jacobi or ILU, for solving two-phase flow problems in porous media.

As an option for the outermost solver, one attractive scheme is the nonlinear Additive Schwarz Preconditioned Inexact Newton (ASPIN) method proposed by Cai and Keyes [28], which has demonstrated robustness in fluid dynamics problems involving shocks and fronts and possesses domain-decomposed distributed memory scalability, but is barely explored for the highly nonlinear models of complex multiphase flow with capillarity, heterogeneity, and complex geometry. The ASPIN framework is initially implemented in MATLAB and applied to fully implicit discretization arising from the two-phase flow in the absence of capillary forces and gravity in heterogeneous porous media [40], where numerical tests involving different phase viscosity ratios and permeability heterogeneity show that this algorithm is more robust and faster than that of additive Schwarz linear preconditioning in the face of challenging problems for a moderate number of subdomains.

We demonstrate ASPIN methods to simulate more general two-phase flow involving the terms of capillary pressure and gravity, which is governed by coupled systems

consisting of an elliptic pressure equation and a parabolic transport equation. A combination of the finite volume spatial discretization and the backward Euler scheme in time differencing results in a fully implicit coupled nonlinear system.

4.1 Basic Concepts

Porosity

Porosity measures how much of the medium is the pore space. In a porous medium, the dead-end pores can not store and transmit fluids, and therefore we only consider this term as the "effective porosity" associated with the interconnected pores:

$$\phi = \frac{\text{volume of the pore space available for flow within the medium}}{\text{the total volume of medium}},$$

which depends on pressure due to rock compressibility.

Permeability

The (absolute) permeability, denoted by $\mathbf{K}$, measures the capacity of a porous medium to transmit fluids through its interconnected pores. By a change of basis (adjusting the coordinate system to coincide with main flow directions), it is possible to get a diagonal tensor $\mathbf{K} = \text{diag}(K_{11}, K_{22}, K_{33})$. If $K_{11} = K_{22} = K_{33}$, it is called isotropic porous medium; otherwise, it is anisotropic.

Phase

Phase refers to a chemically homogeneous region of fluid that is separated from other fluid continua by a interface, where the physical fluid properties are discontinuous. The fluids are regarded as separate phases, so called "multiphase flow", if the phases

are immiscible. In this dissertation, we focus on the two-phase flow (e.g. water and oil).

Saturation

The saturation of a phase α measures how much of the pore space is filled with phase α, and it is defined as

$$S_\alpha = \frac{\text{volume of phase } \alpha \text{ within the medium}}{\text{the volume of pore space}}.$$

4.2 Two-phase Flow Model

In this section, we consider the immiscible displacement of incompressible two-phase flow in a two-dimensional porous medium, which is described by the saturation equation and Darcy's law of the wetting and non-wetting phases. In this model, we take into account gravity effects and capillary forces that also play an essential role in contributing the strong nonlinearities besides the relative permeability functions. It is noted that phase densities are constants for two-phase incompressible flow, even though the density of the wetting phase might differ significantly from that of the non-wetting phase.

4.2.1 Description of the Problem

Based on mass conservation, the saturation equation for each fluid phase is given by:

$$\phi \frac{\partial S_\alpha}{\partial t} + \nabla \cdot \mathbf{u}_\alpha = q_\alpha, \quad \alpha = w,\ n, \tag{4.1}$$

where ϕ is the porosity of the porous media, the wetting phase (e.g, water) and non-wetting phase (e.g, oil) are denoted by the subscripts w and n, respectively. Here,

S_α, $\mathbf{u}_\alpha$, and q_α are, respectively, the saturation, velocity, and the external volumetric flow rate of phase α. According to Darcy's Law for each phase,

$$\mathbf{u}_\alpha = -\lambda_\alpha \mathbf{K}(\nabla p_\alpha - \rho_\alpha \mathbf{g}), \quad \alpha = w,\ n, \tag{4.2}$$

where $\mathbf{K}$ is the absolute permeability tensor of the porous media, p_α, ρ_α are the pressure, density, and

$$\mathbf{g} = g\nabla z,$$

where g is the magnitude of the gravitational acceleration, and z is the depth at the position x. The mobility function λ_α is the ratio of relative permeability $k_{r\alpha}$ and the viscosity μ_α

$$\lambda_\alpha = \frac{k_{r\alpha}}{\mu_\alpha}.$$

The saturation constraint is given as

$$S_w + S_n = 1, \tag{4.3}$$

and the capillary pressure function p_c is defined as

$$p_c = p_n - p_w. \tag{4.4}$$

The total mobility is expressed as $\lambda_t = \lambda_w + \lambda_n$, and we also define $q_t = q_w + q_n$. The formulation for the incompressible two-phase flow can be obtained by substituting (4.2), (4.3), (4.4) into (4.1):

$$-\nabla \cdot (\lambda_t \mathbf{K}\nabla p_w) = q_t + \nabla \cdot (\lambda_n \mathbf{K}\nabla p_c) - \nabla \cdot ((\lambda_n \rho_n + \lambda_w \rho_w)\mathbf{K}\mathbf{g}), \tag{4.5}$$

$$\phi \frac{\partial S_w}{\partial t} - \nabla \cdot (\lambda_w \mathbf{K}(\nabla p_w - \rho_w \mathbf{g})) = q_w. \tag{4.6}$$

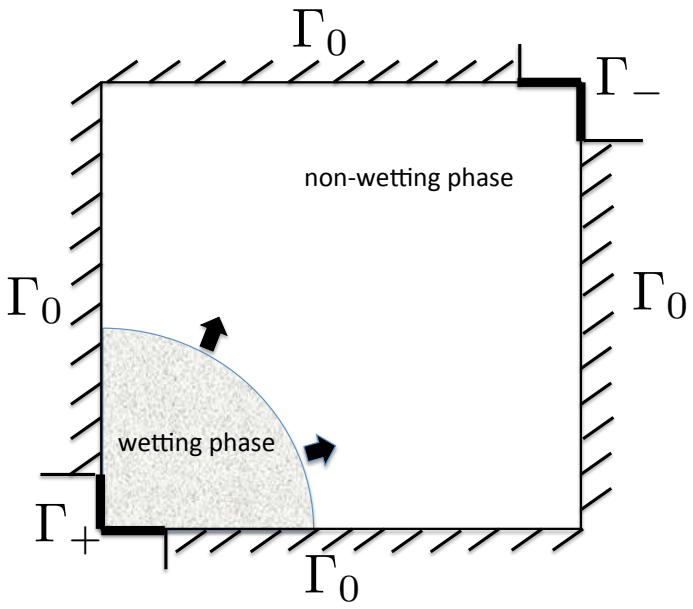

Figure 4.1: The setting of the quarter-five spot problem.

To close the coupled pressure-saturation equations (4.5) and (4.6), we require appropriate boundary and initial conditions. Let $\Gamma = \Gamma_+ \cup \Gamma_- \cup \Gamma_0$ be the boundary of the rectangular domain, where Γ_+, Γ_-, and Γ_0 denote the inflow boundary, the outflow boundary, and no flow boundary, respectively. The general boundary conditions are stated as

$$\begin{aligned} &\mathbf{u}_w \cdot \mathbf{n} = u_B^w(x,y), \quad \mathbf{u}_n \cdot \mathbf{n} = u_B^n(x,y), \quad \text{on } \Gamma_+, \\ &p_w(x,y) = p_B(x,y), \quad \lambda_n \mathbf{K} \nabla p_c \cdot \mathbf{n} = 0, \quad \text{on } \Gamma_-, \\ &\mathbf{u}_w \cdot \mathbf{n} = 0, \quad \mathbf{u}_n \cdot \mathbf{n} = 0, \quad \text{on } \Gamma_0, \end{aligned} \tag{4.7}$$

where $\mathbf{n}$ is the unit outer normal vector. For example, Figure 4.1 shows the setting of the quarter-five spot problem.

The initial condition is given by

$$S_w|_{t=0} = S^{init} \quad \text{in } \Omega. \tag{4.8}$$

In this work, we solve equations (4.5) and (4.6) simultaneously in the primary unknowns p_w and S_w.

4.2.2 Relative Permeability and Capillary Pressure

The relative permeability describes how one phase flows in the presence of other phases, and it is determined empirically or obtained via the experiments in the laboratory. Extensive literature gives analytical expression for the relationship between the relative permeabilities and the saturation of the wetting phases [74, 78, 79, 80, 81].

In this dissertation, the relative permeabilities are given by:

$$k_{rw} = S_e^2; \quad k_{rn} = (1 - S_e)^2, \tag{4.9}$$

where S_e is the normalized saturation defined as:

$$S_e = \frac{S_w - S_{rw}}{1 - S_{rw} - S_{rn}}, \tag{4.10}$$

S_{rn} is the irreducible saturation of the non-wetting phase, and S_{rw} is the connate water saturation, i.e., the saturation of the wetting phase trapped in the pores of the rock during formation of the rock.

A discontinuity of the pressure, referred to as the capillary pressure p_c, occurs across an interface between two immiscible fluids as a consequence of the interfacial tension. In general, the capillary pressure depends on many factors, such as the wetting phase saturation S_w, the surface tension σ, permeability k, porosity ϕ, and so on. For simplicity, the capillary pressure formula that we use throughout the dissertation refers to [82, 83]

$$p_c(S_w) = -B_c \log(S_e), \tag{4.11}$$

where the capillary pressure parameter B_c is inversely proportional to $\sqrt{k}$, the square root of the permeability.

4.2.3 PVI

We measure time by Pore Volume Injected (PVI) [84], which is analogous to dimensionless time. PVI is defined by

$$\text{PVI} = \frac{Qt}{V_p}, \tag{4.12}$$

where Q is the total volumetric flow rate, t is the dimensional time, and V_p is the total Pore Volume (PV).

4.2.4 Discretization

The fully implicit approach [76, 85, 72] gives the highest robustness in long term simulation, compared with other numerical discretization schemes to different extent of implicitness and coupling for the governing equations, such as the IMplicit Pressure-Explicit Saturation (IMPES) [74, 86] or iterative IMPES [87, 88, 89, 90], sequential methods [91], semi-implicit discretization [92] and the adaptive implicit method (AIM) [93, 94].

In our work, the finite volume method is applied to discretize the model equations for the spatial terms, and the implicit first order scheme (Backward Euler) is used for the time differencing approximation. The unknowns p_w and S_w are approximated by cell-wise constants. The permeability and mobilities are not well-defined at the interface, and hence approximated separately by harmonic average values and standard upstream weighting. The upwind direction for each phase α is determined using its phase velocity $\mathbf{u}_\alpha$ [95, 96].

We consider a rectangular domain Ω in $\mathbb{R}^2$, and a grid cell is denoted by Ω_i in Ω. The time interval $[0, T]$ is divided as $0 = t_0 < t_1 < \cdots < t_{N_t} = T$, and the time step length $\triangle t_k = t_{k+1} - t_k$. Cell interfaces are defined by $\gamma_{ij} = \partial\Omega_i \cap \partial\Omega_j$, and $\mathbf{n}_{ij}$ are associated normal vectors pointing from cell i to cell j. At the time t_k, the solution vector is denoted by $u^k = (P_1^k, \ldots, P_N^k, S_1^k, \ldots, S_N^k)^T$, where N is the number of cells.

Taking the integral over Ω_i for the pressure equation (4.5):

$$
\begin{aligned}
& -\int_{\Omega_i} \nabla \cdot (\lambda_t^{k+1} \mathbf{K} \nabla p_w^{k+1}) dx \\
& -\int_{\Omega_i} \nabla \cdot (\lambda_n^{k+1} \mathbf{K} \nabla p_c^{k+1}) dx \\
& +\int_{\Omega_i} \nabla \cdot ((\lambda_n^{k+1} \rho_n + \lambda_w^{k+1} \rho_w) \mathbf{K} \mathbf{g}) dx \\
& -\int_{\Omega_i} q_t^{k+1} dx \\
= & \sum_j \int_{\gamma_{ij}} \left[-\lambda_t^{k+1} \mathbf{K} \nabla p_w^{k+1} - \lambda_n^{k+1} \mathbf{K} \nabla p_c^{k+1} + (\lambda_n^{k+1} \rho_n + \lambda_w^{k+1} \rho_w) \mathbf{K} \mathbf{g} \right] \cdot \mathbf{n}_{ij} \, dv \\
& -\int_{\Omega_i} q_t^{k+1} \, dx, \qquad (4.13)
\end{aligned}
$$

which is approximated using the standard two-point flux approximation finite volume scheme as follows:

$$
\begin{aligned}
F_p(u^{k+1}) &= \sum_j |\gamma_{ij}| \, [\lambda_t^{up}]_{ij}^{k+1} K_{ij}^H \frac{P_i^{k+1} - P_j^{k+1}}{\triangle x_i + \triangle x_j} \\
&+ \sum_j |\gamma_{ij}| \, [\lambda_n^{up}]_{ij}^{k+1} K_{ij}^H \frac{P_{c,\,i}^{k+1} - P_{c,\,j}^{k+1}}{\triangle x_i + \triangle x_j} \\
&- \sum_j |\gamma_{ij}| \, ([\lambda_n^{up}]_{ij}^{k+1} \rho_n + [\lambda_w^{up}]_{ij}^{k+1} \rho_w) K_{ij}^H g \frac{z_i - z_j}{\triangle x_i + \triangle x_j} \\
&- |\Omega_i| q_{t,i}^{k+1}, \qquad (4.14)
\end{aligned}
$$

where

$$
K_{ij}^H = (\triangle x_i + \triangle x_j) \left(\frac{\triangle x_i}{K_i} + \frac{\triangle x_j}{K_j} \right)^{-1}. \qquad (4.15)
$$

Similarly, integrating the saturation equation (4.6):

$$\begin{aligned} & \frac{\Phi_i}{|\Omega_i|}\int_{\Omega_i}\frac{\partial S_w}{\partial t}+\frac{1}{|\Omega_i|}\int_{\Omega_i}\nabla\cdot(-\lambda_w^{k+1}\mathbf{K}(\nabla p_w^{k+1}-\rho_w\mathbf{g}))\,dx-\frac{1}{|\Omega_i|}\int_{\Omega_i}q_w^{k+1}dx \\ = \; & \frac{\Phi_i}{|\Omega_i|}\int_{\Omega_i}\frac{\partial S_w}{\partial t}+\frac{1}{|\Omega_i|}\sum_j\int_{\gamma_{ij}}\left[-\lambda_w^{k+1}\mathbf{K}(\nabla p_w^{k+1}-\rho_w\mathbf{g})\right]\cdot\mathbf{n}_{ij}\,dv-q_{w,i}^{k+1}. \quad (4.16) \end{aligned}$$

We define the cell-average of the saturation at time $t=t_k$ as

$$S_i^k=\frac{1}{|\Omega_i|}\int_{\Omega_i}S_w(x,t_k)dx, \quad (4.17)$$

and apply the backward Euler scheme to approximate the derivative term with respect to the time. We obtain the finite volume numerical scheme for the saturation equation:

$$\begin{aligned} F_s(u^{k+1}) \; = \; & \frac{\Phi_i}{\triangle t_k}(S_i^{k+1}-S_i^k)+\frac{2}{|\Omega_i|}\sum_j|\gamma_{ij}|\,[\lambda_w^{up}]_{ij}^{k+1}K_{ij}^H\frac{P_i^{k+1}-P_j^{k+1}}{\triangle x_i+\triangle x_j} \\ & -\frac{2}{|\Omega_i|}\sum_j|\gamma_{ij}|\,[\lambda_w^{up}]_{ij}^{k+1}K_{ij}^H\rho_w g\frac{z_i-z_j}{\triangle x_i+\triangle x_j}-q_{w,i}^{k+1}. \quad (4.18) \end{aligned}$$

The fully implicit formulation (4.14) and (4.18) result in the nonlinear system

$$F(u^k)=\begin{bmatrix} F_p(u^k) \\ F_s(u^k) \end{bmatrix}=0 \quad (4.19)$$

for each time step. When solving such nonlinear systems using Newton method or its many variations, however, we maybe face the numerical challenges arising from heterogeneous permeability of high contrast, strong nonlinearities of relative permeability, and spatially varied capillary pressure functions. Therefore, some techniques, such as reordering techniques [75] and a two-stage Gauss-Seidel preconditioner [76] and multiscale approaches [69], have been developed to reduce the execution time for linear solvers and reduce the number of Newton iterations. Nonlinear preconditioners

are also good candidates to handle "nonlinear stiffness" and reduce the computational costs.

4.3 Numerical Results

In this section, we present some numerical results using ASPIN methods with subdomain overlap. Our implementation is done using the Portable Extensible Toolkit for Scientific Computing (PETSc) [34], which is an open-source software package. Uniform grids and the regular quadrilateral partitions are used in our experiments, and the number of processors is the same as the number of subdomains. The primary variables for all the tests are the pressure p_w and the saturation S_w of the wetting phase, which are located in the center of the cells. The injection well and production well are modeled as the inflow boundary and the outflow boundary.

The global preconditioned nonlinear iteration is stopped when

$$\|\mathcal{F}(u^{(k)})\|_2 \leq \varepsilon^{rel}_{\text{global-nonlinear}}\|\mathcal{F}(u^{(0)})\|_2, \quad \text{or} \quad \|\mathcal{F}(u^{(k)})\|_2 \leq \varepsilon^{abs}_{\text{global-nonlinear}} \tag{4.20}$$

is first satisfied. We set $\varepsilon^{rel}_{\text{global-nonlinear}} = 10^{-4}$ and $\varepsilon^{abs}_{\text{global-nonlinear}} = 10^{-8}$ for all test cases. GMRES is used for solving the global Jacobian systems. The global linear iteration is stopped when

$$\|\mathcal{F}(u^{(k)}) - \sum_{i=1}^{N} J^{-1}_{\Omega_i} Jp^{(k)}\|_2 \leq \varepsilon_{\text{global-linear}}\|\mathcal{F}(u^{(k)})\|_2 \tag{4.21}$$

is first satisfied. We select $\varepsilon_{\text{global-linear}} = 10^{-4}$ for all the tests. The local nonlinear iteration on each subdomain is stopped when

$$\|\mathcal{F}_{\Omega_i}(g^{(k)}_{i,l})\|_2 \leq \varepsilon^{rel}_{\text{local-nonlinear}}\|\mathcal{F}_{\Omega_i}(g^{(k)}_{i,0})\|_2, \quad \text{or} \quad \|\mathcal{F}_{\Omega_i}(g^{(k)}_{i,l})\|_2 \leq \varepsilon^{abs}_{\text{local-nonlinear}} \tag{4.22}$$

is first satisfied. We select $\varepsilon^{rel}_{\text{local-nonlinear}} = 10^{-3}$ and $\varepsilon^{abs}_{\text{local-nonlinear}} = 10^{-8}$ for all the tests. Due to their small size of the local nonlinear problems, LU decomposition is used to solve each local Jacobian system. All global and local Jacobian matrices are constructed by using a multi-colored finite difference with tolerance 10^{-10}. The standard cubic backtracking algorithm is used for global nonlinear problems. The parameters associated with inexact Newton methods with backtracking are: $\alpha = 10^{-4}$, $\lambda_{min} = 0.1$, $\lambda_{max} = 0.5$. We get the initial guess by solving the single phase pressure equation using the initial saturation.

The adaptive time-step scheme is used for all the tests. The time step is halved if the global (or local) Newton iteration does not satisfy the stopping criteria (4.20), (4.21), (4.22) within 10 (or 25) iterations, and is doubled if the previous time step was not reduced. Also we set the maximum time step $\triangle t_{max}$, and the time step varies but under the limit $\triangle t_{max}$.

4.3.1 Example 1 — Quarter Five-Spot Problem

The reservoir domain is $[0, 500\text{m}] \times [0, 500\text{m}]$ divided into 256×256 cells. The simulation is carried out up to 5 years (0.5 PVI), starting with a time step of 0.0001 PVI and the maximum time step of 0.01 PVI. The reservoir is initially filled by the non-wetting phase, and the initially saturation is set to be zero: $S_w = 0$. The wetting phase is injected with the flow rate 0.1 PV/year at the inflow boundary (the lower left corner), and the pressure p_w is fixed to 101325 Pa at the outflow boundary (the upper right corner). The no flow boundary condition is imposed on other boundaries.

In this quarter five-spot simulation, we consider two types of permeability field: (A) homogeneous field with 100 md and (B) heterogeneous field (almost centered about 100 md) with a variation of three orders of magnitude shown in Figure 4.2, and ignoring the capillary pressure and gravity. The random heterogeneous field is generated by a simplified geostatistical method [96]. We generate a field of indepen-

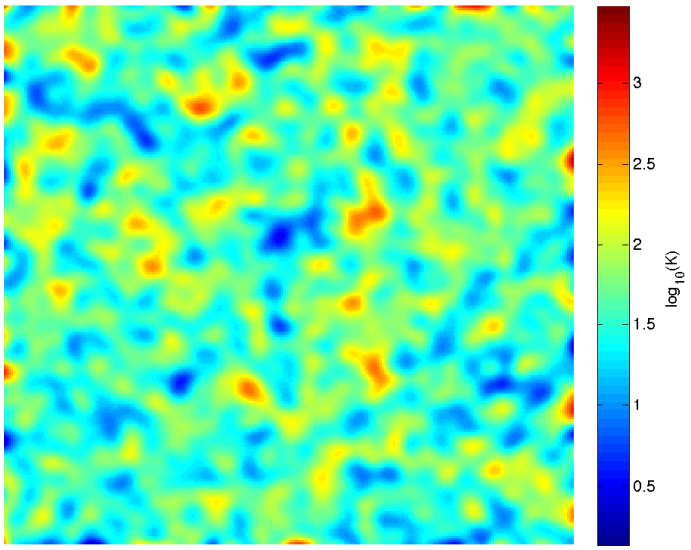

Figure 4.2: Permeability field. The permeability varies between 1.3 md and 3.0 darcy.

dent, normally distributed variables and convolve it with a Gaussian kernel, which results in the porosity $\tilde{\phi}$ as a simple approximation of a Gaussian field. Using $\tilde{\phi}$, we get the permeability by the Carman-Kozeny relation:

$$K = \frac{1}{2\tau A_v} \frac{\tilde{\phi}^3}{(1-\tilde{\phi})^2}, \tag{4.23}$$

where $\tau = 0.81$, $d_p = 10$ μm and $A_v = 6/d_p$. The porosity is set to $\phi = 0.2$, and the parameters for the fluid are $\rho_w = 1000$ kg/m^3, $\rho_n = 660$ kg/m^3, $\mu_w = 1$ cp and $\mu_n = 0.45$ cp. Figure 4.3 shows saturation plots for the homogeneous and heterogeneous fields after 0.1 PVI, 0.3 PVI, and 0.5 PVI.

To investigate the scalability, we fix the mesh size and vary the number of processors from 4 to 64 shown in Table 4.1. Whether the simulation in the homogeneous permeability field or in heterogeneous permeability field, the average number of ASPIN iterations and the number of time steps still do not change much, but the average number of GMRES iterations increases considerably with the number of processors. It is expected that the simulation takes more time in heterogeneous porous media, due to the permeability field with a variation of three orders of magnitude. Table 4.1

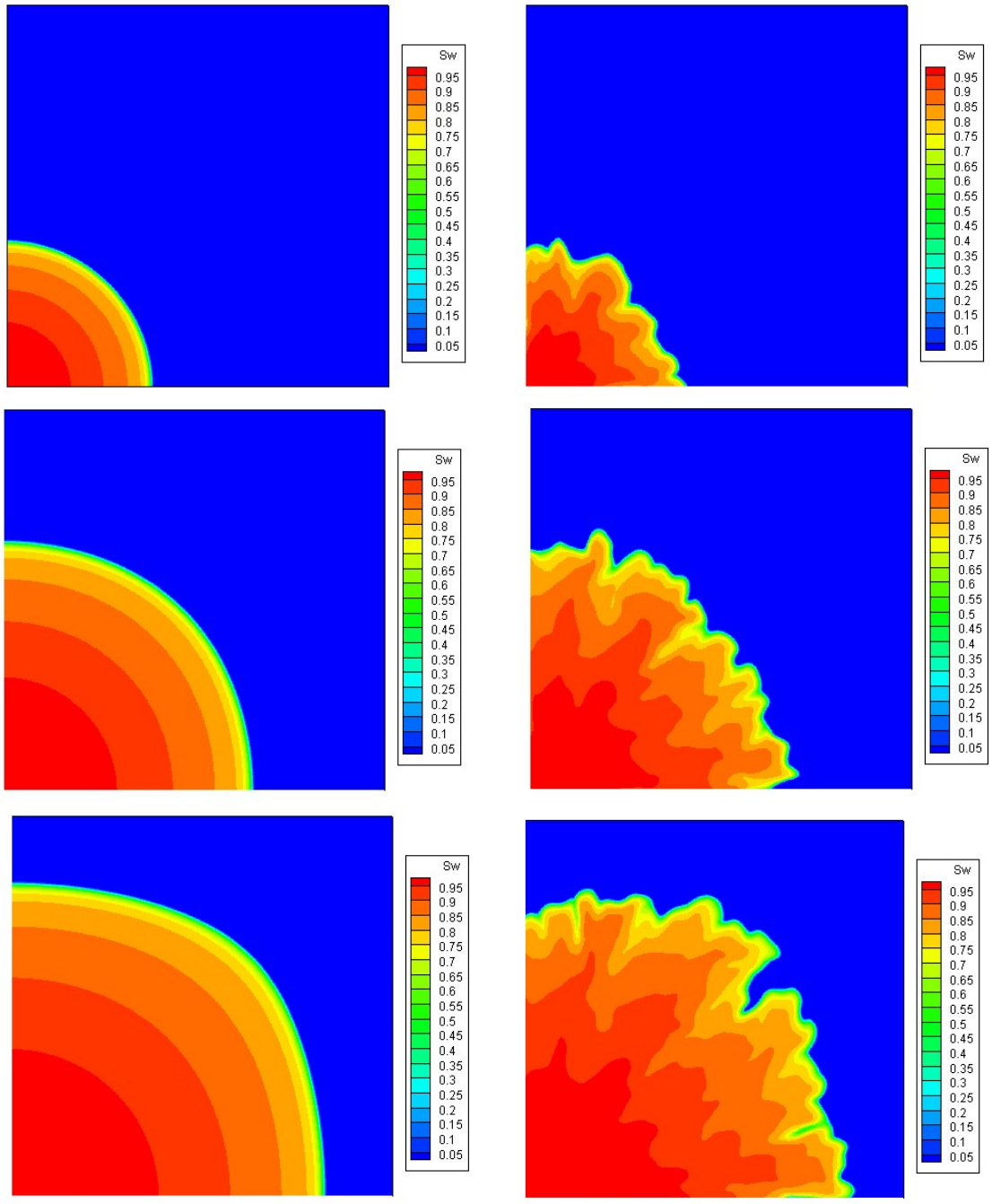

Figure 4.3: The quarter-five spot problem with the mesh 256×256 at different times $t = 0.1$ PVI, $t = 0.3$ PVI, $t = 0.5$ PVI. The flow behavior corresponds to the homogeneous field (left) and heterogeneous field (right).

shows that there is no significant difference with respect to the number of time steps and the average number of ASPIN iterations between in the homogeneous field and in heterogeneous field.

	Average number of ASPIN iterations per time step	
Subdomain partition	case A: Homogeneous field	case B: Heterogeneous field
$2 \times 2 = 4$	2.3	2.5
$4 \times 4 = 16$	2.9	3.0
$8 \times 8 = 64$	3.6	3.6
	Average number of GMRES iterations per time step	
Subdomain partition	case A: Homogeneous field	case B: Heterogeneous field
$2 \times 2 = 4$	36.5	66.7
$4 \times 4 = 16$	91.2	190.2
$8 \times 8 = 64$	329.2	470.8
	Average execution time per time step	
Subdomain partition	case A: Homogeneous field	case B: Heterogeneous field
$2 \times 2 = 4$	3.724e+03	4.424e+03
$4 \times 4 = 16$	6.924e+02	8.584e+02
$8 \times 8 = 64$	1.983e+02	2.525e+02
	The number of time steps up to 0.5 PVI	
Subdomain partition	case A: Homogeneous field	case B: Heterogeneous field
$2 \times 2 = 4$	57	60
$4 \times 4 = 16$	57	60
$8 \times 8 = 64$	61	65

Table 4.1: A 256×256 mesh on different numbers of processors, the simulation time is 0.5 PVI, starting with the initial time step 0.0001 PVI, and the overlap = 2.

4.3.2 Example 2 — Layered Permeability

The dimensions of the model are 300 meters long by 180 meters wide, and the simulation grid consists of 50×90 blocks. The simulation time is 0.5 PVI, starting with the initial time step $\triangle t = 0.00001$ PVI and the initial saturation $S_w = 0$. The wetting phase is injected with the flow rate 0.11 PV/year from the left boundary, and non-wetting phase completely filling the whole reservoir is displaced by the wetting phase toward the right boundary with a fixed pressure $p_w = 101325$ Pa. We again assume the top and bottom boundaries of the reservoir to be impermeable, i.e., the normal component of both the wetting phase flux and the non-wetting phase flux across the boundaries to vanish. The porosity is set to be $\phi = 0.2$, and the parameters for the fluid are $\rho_w = 1000$ kg/m^3, $\rho_n = 660$ kg/m^3, $\mu_w = 1$ cp and $\mu_n = 0.45$ cp. Figure 4.4 shows permeability distribution with the alternate layered field (1md and 100 md).

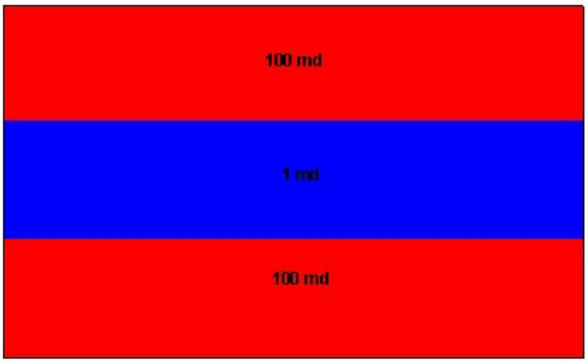

Figure 4.4: Layered alternate permeability with 100 md and 1 md.

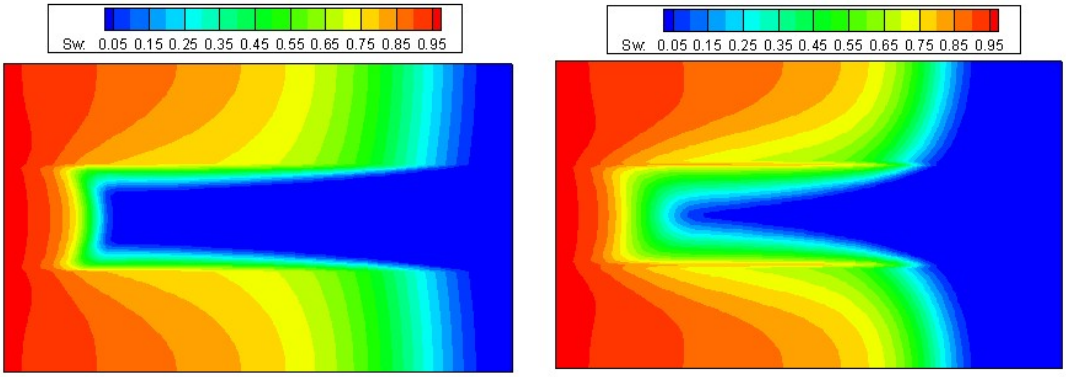

Figure 4.5: Saturation distribution in layered porous media with mesh 50×90 at different time $t = 0.5$ PVI: $B_c = 2.5$ bar (left), $B_c = 1, 10$ bar (right).

The capillary pressure function is given in the equation (4.11), and we consider two cases: (C) B_c is 2.5 bar in the domain and (D) B_c is defined as

$$B_c = \begin{cases} 10 \text{ bar} & k = 1 \text{ md} \\ 1 \text{ bar} & k = 100 \text{ md} \end{cases} \tag{4.24}$$

The numerical tests are again carried out using both in capillary homogeneous ($B_c = 2.5$ bar) and capillary heterogeneous ($B_c = 1$ or 10 bar) domain. There are major differences in saturation plots, as shown in Figure 4.5. As expected, the flow behavior with homogeneous capillary pressure is essentially influenced by the low permeability field, which causes the great delay of displacement in the middle layer.

	Average number of ASPIN iterations per time step	
Subdomain partition	case C: $B_c = 2.5$ bar	case D: $B_c = 1, 10$ bar
$1 \times 3 = 3$	2.7	2.7
$1 \times 6 = 6$	2.8	2.8
$1 \times 9 = 9$	2.9	2.9
	Average number of GMRES iterations per time step	
Subdomain partition	case C: $B_c = 2.5$ bar	case D: $B_c = 1, 10$ bar
$1 \times 3 = 3$	27.6	28.2
$1 \times 6 = 6$	79.6	81.3
$1 \times 9 = 9$	118.4	118.6
	Average execution time per time step	
Subdomain partition	case C: $B_c = 2.5$ bar	case D: $B_c = 1, 10$ bar
$1 \times 3 = 3$	1.192e+04	7.654e+03
$1 \times 6 = 6$	5.338e+03	3.965e+03
$1 \times 9 = 9$	3.850e+03	2.644e+03
	The number of time steps up to 0.5 PVI	
Subdomain partition	case C: $B_c = 2.5$ bar	case D: $B_c = 1, 10$ bar
$1 \times 3 = 3$	779	476
$1 \times 6 = 6$	681	473
$1 \times 9 = 9$	679	450

Table 4.2: A 50×90 mesh on different numbers of processors, the simulation time is 0.5 PVI, starting with the initial time step 0.00001 PVI, and the overlap = 2.

In the case of capillary pressure heterogeneity, in contrast, it is observed that the wetting phase penetrates the low permeability layer in large measure, resulting in the later breakthrough.

Table 4.2 shows the performance of ASPIN methods with respect to the number of processors from 3 to 9 in both cases. The execution time decreases as a result of local nonlinear solvers in parallel. Similarly, it is also clearly seen that the algorithm is almost nonlinearly scalable but the number of global linear iterations increases when the domain is divided into more subdomains.

4.3.3 Example 3 — Layer 26 From SPE10 Model 2

We next consider a quarter-five spot problem with permeability and porosity data imported from the revised two-dimensional SPE-10 models in [97, 98]. The 26th

layer is used here, and the domain is rectangular with 60 by 220 cells. The injection rate is 58.8 bbl/day, and the fixed pressure is 4000 psi in the production well. The parameters for the fluid properties and residual saturations are $\rho_w = 1000$ kg/m^3, $\rho_n = 660$ kg/m^3, $\mu_w = 0.3$ cp, $\mu_n = 3.0$ cp, and $S_{rw} = S_{rn} = 0.2$. The initial water saturation is equal to S_{rw}. The position of wells and boundary conditions are similar to those of the example 1. The simulation starts from 1 day up to 300 days.

Numerical tests are carried out using 8 processors in parallel. We use this example to compare the performance of the ASPIN method, the inexact Newton method with a backtracking technique (INB), and the IMPES method. The ASPIN method is implemented with the partition of 2 by 4 subdomains and the overlap of 4 cells. As for the INB method, we solve the correction equation using GMRES, but LU decomposition as the preconditioner for the linear solvers in order to ensure fast convergence. For these two methods, the stopping criteria of global nonlinear and linear solvers are the same with statements in the beginning of this section, except the time step is halved if the global nonlinear solver does not converge within 20 steps. All global and local Jacobian matrices are constructed by using a multi-colored finite difference with tolerance 10^{-12}, and the maximum time step is 10 days. The IMPES method solves the pressure equation using an external library SuperLU_DIST [99] as the direct solver, and an adaptive time-step method [100], based on the change in S_w, is used to guarantee stability in time when the saturation is updated explicitly.

Figure 4.6 tracks the number of Newton iterations for both ASPIN and INB methods with an initial time step of 1 day. It shows that the ASPIN method always takes fewer iterations to converge than the INB method. Table 4.3 demonstrates that the ASPIN method greatly reduces the number of global nonlinear iterations compared with INB. In contrast with IMPES, the ASPIN method gains savings of an order of magnitude in timestep size, with reductions in execution time that scale according to the number of timesteps. Overall execution time savings are modified

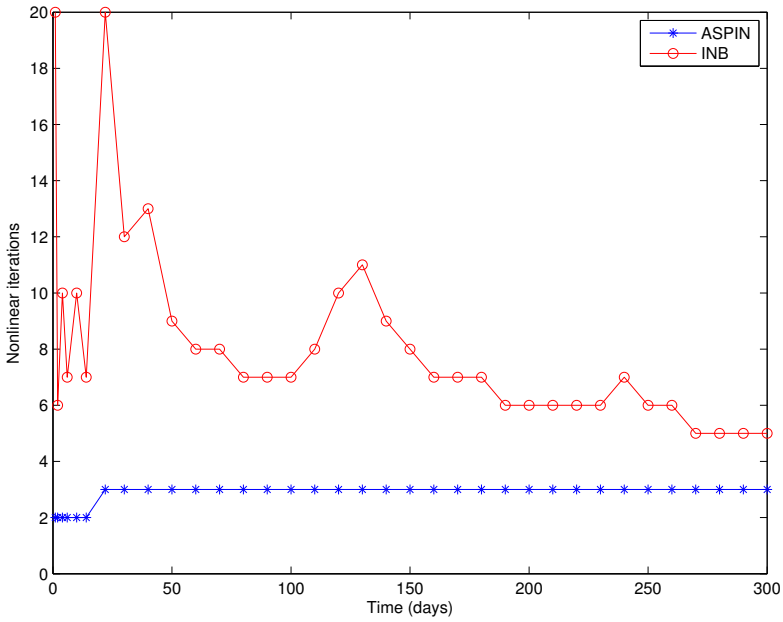

Figure 4.6: Convergence history for the Layer 26 from SPE10 using ASPIN method and the inexact Newton method with a backtracking technique (INB) up to 300 days. The maximum time step is 10 days in both cases.

by the higher cost per step of ASPIN versus IMPES.

	ASPIN	INB	IMPES
No. of time steps	35	35	2358
Average timestep size (days)	8.57	8.57	0.13
Newton iterations	99	287	-
Execution time (sec)	1.295e+02	6.482e+02	1.268e+03

Table 4.3: A 60×220 mesh on 8 processors, the simulation time is 300 days, starting with the initial time step 1 days, and the overlap = 4 in the ASPIN method.

4.4 Concluding Remarks

In this chapter, we present the numerical results for the two-phase flow including the effects of capillary pressure, using the fully parallel nonlinearly preconditioned inexact Newton method, and we notice that the number of Newton iterations does not change much, but the global linear iterations increase substantially when the

number of processors increases. The degradation of one level domain decomposition is attributed to the lack of inter-subdomain communication. To make linear solver scalable, two-level ASPIN methods [17, 35, 39] are needed to reduce the number of global linear iterations.

Load balancing is another issue to be improved in future. We expect that fewer unknowns are included for more difficult local nonlinear systems, however, it is very difficult to achieve this goal for the fixed layout of domain decomposition. In the quarter-five spot problem, for instance, there is actually no need for many nonlinear iterations on some subdomains when the wetting phase front is away from the production well, but the computational cost for each domain varies as the wetting phase front advances. Therefore, it is a future challenge to realize load balancing using the adaptive domain decomposition strategy.

Chapter 5

Field-split Preconditioned Inexact Newton Algorithms

In this chapter, we propose an algebraic variant of ASPIN based on field splitting, which typically generates subproblems involving nonintersecting subsets of the original unknowns, whereas the classical ASPIN in [28] typically generates subproblems whose unknowns overlap as induced by subdomain geometry. The field-split preconditioned inexact Newton method [50] generates a number of subproblems proportional to the number of identifiable strongly coupled physical phenomena. In contrast, the granularity of subproblems is arbitrarily large in the classical ASPIN in [28] limited only by resolution consideration. In multiphysics problems of large scale, the classical ASPIN may naturally nest inside of the field-split preconditioned inexact Newton methods, or *vice versa*.

5.1 Field-split Preconditioned Inexact Newton Algorithm

Consider a nonlinear root-finding problem (2.1). In order to demonstrate the field-split preconditioned inexact Newton framework, the nonlinear function $F(x)$ is split

conformally into two nonoverlapping components representing different physical aspects as

$$F(x) = F(u, v) = \begin{bmatrix} G(u, v) \\ H(u, v) \end{bmatrix} = 0, \quad x = [u, v]^T. \tag{5.1}$$

As with the domain-based ASPIN in [28], the field-split preconditioned inexact Newton algorithm solves a preconditioned problem

$$\mathcal{F}(x) = \mathcal{F}(u, v) = \begin{bmatrix} g(u, v) \\ h(u, v) \end{bmatrix} = 0, \tag{5.2}$$

which has the same solution with (5.1), where $g(u, v)$ and $h(u, v)$ are defined as the physical variable corrections of any given $x = [u, v]^T$, and the INB algorithm is used as the nonlinear solver for the global preconditioned system, as well as the two submodels. In this section, we focus on the form of nonlinearly preconditioned function $\mathcal{F}(x)$ and the Jacobian calculation corresponding to the field-split preconditioned inexact Newton algorithms. We introduce two types of field-split preconditioned inexact Newton algorithms: ASPIN and MSPIN.

5.1.1 Field-split ASPIN

We describe first the construction of the Jacobi-type nonlinear preconditioner. In the ASPIN algorithm, the submodels are solved independently for the physical variable corrections, and these corrections form the preconditioned system.

For any given $x = [u, v]^T \in R^n$, define $g = g(u, v)$ and $h = h(u, v)$ as the solutions of the submodels in the original nonlinear system (5.1)

$$G(u - g, v) = 0, \tag{5.3}$$

$$H(u, v - h) = 0. \tag{5.4}$$

The Jacobi-type nonlinearly preconditioned function $\mathcal{F}_J$ is then defined as

$$\mathcal{F}_J(x) = \begin{bmatrix} g(u,v) \\ h(u,v) \end{bmatrix}, \quad x = [u,v]^T. \tag{5.5}$$

The Jacobian of $\mathcal{F}_J(x)$ is not as explicitly available as $F'(x)$ due to the implicit definition of $\mathcal{F}_J(x)$ via (5.3) and (5.4); therefore, it is necessary to derive a formula for the application of the Jacobian matrix corresponding to the preconditioned function $\mathcal{F}_J$ in order to apply Newton's method or its variations. We next describe an approximate, readily computable Jacobian form of $\mathcal{F}_J(x)$.

Let $p = u - g(u,v)$ and $q = v - h(u,v)$. Taking the derivative of (5.3) with respect to u, we have

$$\frac{\partial G}{\partial p}(I_u - \frac{\partial g}{\partial u}) = 0, \tag{5.6}$$

where I_u is the identity matrix that has the same dimension as the u block. Assuming $\frac{\partial G}{\partial p}$ is nonsingular, equation (5.6) implies

$$\frac{\partial g}{\partial u} = I_u. \tag{5.7}$$

Next, we take the derivative of (5.3) with respect to v, yielding

$$\frac{\partial G}{\partial p}(-\frac{\partial g}{\partial v}) + \frac{\partial G}{\partial v} = 0, \tag{5.8}$$

which is equivalent to

$$\frac{\partial g}{\partial v} = (\frac{\partial G}{\partial p})^{-1}\frac{\partial G}{\partial v}. \tag{5.9}$$

Similarly, taking the derivatives of (5.4) with respect to u and v, we obtain

$$\frac{\partial h}{\partial u} = (\frac{\partial H}{\partial q})^{-1}\frac{\partial H}{\partial u} \tag{5.10}$$

and

$$\frac{\partial h}{\partial v} = I_v, \tag{5.11}$$

where I_v is again the identity matrix of appropriate size. From expressions (5.7), (5.9), (5.10), and (5.11), we can write down the Jacobian of $\mathcal{F}_J(x)$ as follows:

$$\mathcal{J}(u,v) = \begin{bmatrix} g_u & g_v \\ h_u & h_v \end{bmatrix} = \begin{bmatrix} (\frac{\partial G}{\partial p})^{-1} & \\ & (\frac{\partial H}{\partial q})^{-1} \end{bmatrix} \begin{bmatrix} \frac{\partial G}{\partial p} & \frac{\partial G}{\partial v} \\ \frac{\partial H}{\partial u} & \frac{\partial H}{\partial q} \end{bmatrix}. \tag{5.12}$$

Due to the continuity of $F(x)$, we know that $g(u,v) \to 0$ and $h(u,v) \to 0$, i.e., $p \to u$, $q \to v$, when $x = [u,v]^T$ is sufficiently close to the exact solution. In practice, it is more convenient to use the following approximate Jacobian

$$\hat{\mathcal{J}}(u,v) = \begin{bmatrix} G_u^{-1} & \\ & H_v^{-1} \end{bmatrix} \begin{bmatrix} G_u & G_v \\ H_u & H_v \end{bmatrix} = \begin{bmatrix} G_u & \\ & H_v \end{bmatrix}^{-1} J(u,v). \tag{5.13}$$

The approximate Jacobian matrix $\hat{\mathcal{J}}$ will generally be dense and expensive to form explicitly. However, only the multiplication by $\hat{\mathcal{J}}$ with a given vector x, $y = \hat{\mathcal{J}}x$, is required for Krylov subspace methods when we solve the Jacobian system. In our implementation, the matrix-vector multiplication $\hat{\mathcal{J}}x = y$ is carried out as follows:

1. Perform the multiplication $w = Jx$, $w = [w_1, w_2]^T$.
2. Solve $G_u y_1 = w_1$ and $H_v y_2 = w_2$.
3. Form the result $y = [y_1, y_2]^T$.

Remark 6. *In the linear case, this algorithm is the same as physics-based block Jacobi linear preconditioning. If we have the linear system*

$$F(x) = Ax - b, \tag{5.14}$$

where

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}, \quad b = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}, \tag{5.15}$$

then the preconditioned system has the form of

$$\mathcal{F}_J(x) = M^{-1}(Ax - b), \qquad M = \begin{bmatrix} A_{11} & 0 \\ 0 & A_{22} \end{bmatrix}. \tag{5.16}$$

5.1.2 Field-split MSPIN

In distinction from the ASPIN algorithm, the submodels are solved sequentially for the physical variable corrections in the MSPIN algorithm, and the preconditioned system again consists of these corrections. We describe the details of the construction of the Gauss-Seidel-like nonlinear preconditioner.

For any given $x = [u, v]^T \in R^n$, the preconditioned function

$$\mathcal{F}_{GS}(x) = \begin{bmatrix} g(u, v) \\ h(u, v) \end{bmatrix} \tag{5.17}$$

is obtained by solving

$$G(u - g, v) = 0, \tag{5.18}$$

for g. With the values of u, v, g, the following system is solved

$$H(u - g, v - h) = 0, \tag{5.19}$$

for h. Since the preconditioned function $\mathcal{F}_{GS}(x)$ is implicitly defined via (5.18) and (5.19), the calculation of the corresponding Jacobian is not straightforward. Below we discuss a computable form of the inverse action.

As shown in section 5.1.1, the derivatives of $g(u, v)$ with respect to u and v are

$$\frac{\partial g}{\partial u} = I_u \tag{5.20}$$

and

$$\frac{\partial g}{\partial v} = (\frac{\partial G}{\partial p})^{-1}\frac{\partial G}{\partial v}, \tag{5.21}$$

respectively. Taking the derivative of (5.19) with respect to u,

$$\frac{\partial H}{\partial p}(I_u - \frac{\partial g}{\partial u}) + \frac{\partial H}{\partial q}(-\frac{\partial h}{\partial u}) = 0. \tag{5.22}$$

Assuming $\frac{\partial H}{\partial q}$ is nonsingular, the equations (5.20) and (5.22) imply

$$\frac{\partial h}{\partial u} = 0. \tag{5.23}$$

Taking the derivative of (5.19) with respect to v,

$$\frac{\partial H}{\partial p}(-\frac{\partial g}{\partial v}) + \frac{\partial H}{\partial q}(I_v - \frac{\partial h}{\partial v}) = 0, \tag{5.24}$$

which is equivalent to

$$\frac{\partial h}{\partial v} = I_v - (\frac{\partial H}{\partial q})^{-1}\frac{\partial H}{\partial p}\frac{\partial g}{\partial v}. \tag{5.25}$$

Substituting (5.21) into (5.25), we have

$$\frac{\partial h}{\partial v} = (\frac{\partial H}{\partial q})^{-1}(\frac{\partial H}{\partial q} - \frac{\partial H}{\partial p}(\frac{\partial G}{\partial p})^{-1}\frac{\partial G}{\partial v}). \tag{5.26}$$

Integrating (5.20), (5.21), (5.23), (5.26) into the Jacobian of $\mathcal{F}_{GS}$, we get

$$\mathcal{J} = \begin{bmatrix} g_u & g_v \\ h_u & h_v \end{bmatrix} = \begin{bmatrix} (\frac{\partial G}{\partial p})^{-1} & \\ -(\frac{\partial H}{\partial q})^{-1}\frac{\partial H}{\partial p}(\frac{\partial G}{\partial p})^{-1} & (\frac{\partial H}{\partial q})^{-1} \end{bmatrix} \begin{bmatrix} \frac{\partial G}{\partial p} & \frac{\partial G}{\partial v} \\ \frac{\partial H}{\partial p} & \frac{\partial H}{\partial q} \end{bmatrix}, \tag{5.27}$$

or, more concisely,

$$\mathcal{J}(u,v) = \begin{bmatrix} \frac{\partial G}{\partial p} & \\ \frac{\partial H}{\partial p} & \frac{\partial H}{\partial q} \end{bmatrix}^{-1} \begin{bmatrix} \frac{\partial G}{\partial p} & \frac{\partial G}{\partial v} \\ \frac{\partial H}{\partial p} & \frac{\partial H}{\partial q} \end{bmatrix}. \tag{5.28}$$

In our implementation, it is more convenient to use the following approximate Jacobian

$$\hat{\mathcal{J}}(u,v) = \begin{bmatrix} G_p & \\ H_p & H_v \end{bmatrix}^{-1} \begin{bmatrix} G_p & G_v \\ H_p & H_v \end{bmatrix} = \begin{bmatrix} G_p & \\ H_p & H_v \end{bmatrix}^{-1} J(p,v). \tag{5.29}$$

Similarly to ASPIN, Krylov subspace methods such as GMRES are used to solve the Jacobian system, which require the multiplication of $\hat{\mathcal{J}}$ with a given vector x, $\hat{\mathcal{J}}x = y$, instead of any explicit calculation and storage of $\hat{\mathcal{J}}$.

Remark 7. *If the matrices A_{11} and A_{22} are nonsingular, then we have*

$$\begin{bmatrix} A_{11} & 0 \\ A_{21} & A_{22} \end{bmatrix}^{-1} = \begin{bmatrix} I & 0 \\ 0 & A_{22}^{-1} \end{bmatrix} \begin{bmatrix} I & 0 \\ -A_{21} & I \end{bmatrix} \begin{bmatrix} A_{11}^{-1} & 0 \\ 0 & I \end{bmatrix}. \tag{5.30}$$

With Remark 7, the matrix-vector multiplication $\hat{\mathcal{J}}x = y$ is carried out as follows:

1. Perform the multiplication $w = Jx$, $w = [w_1, w_2]^T$.
2. Let $z_2 = w_2$ and solve $G_p z_1 = w_1$.
3. Perform $z_2 = z_2 - H_p z_1$ and set $y_1 = z_1$.
4. Solve $H_v y_2 = z_2$.
5. Form the result $y = [y_1, y_2]^T$.

Remark 8. *In the linear case, this algorithm is the same as physics-based block Gauss-Seidel linear preconditioning. If we have the linear system:*

$$F(x) = Ax - b, \tag{5.31}$$

where

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}, \quad b = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}, \tag{5.32}$$

then

$$\mathcal{F}_{GS}(x) = M^{-1}(Ax - b), \qquad M = \begin{bmatrix} A_{11} & 0 \\ A_{21} & A_{22} \end{bmatrix}. \tag{5.33}$$

5.2 Implementation of the Field-split ASPIN and MSPIN

In this section, we describe a complete field-split preconditioned inexact Newton algorithm and implementation details. We also present an illustrative example.

5.2.1 The Field-split ASPIN and MSPIN

Splitting $F(x)$ into $G(u, v)$ and $H(u, v)$, $x = [u, v]^T$. We set $x^{(0)} = [u^{(0)}, v^{(0)}]^T$ as the initial guess. The current approximate solution is denoted by $x^{(k)}$ and $x^{(k+1)}$ is the new approximate solution obtained through the following algorithm.

In Algorithm 7, the determination of the partition of the physical variables can be the most interesting part of implementation, because the best choice is generally problem-specific. Once the partition and the initial guess are given, we move on to step 1.

In step 1(a), we solve nonlinear subproblems (5.34), (5.35), (5.36), (5.37) using

Algorithm 6 $g^{(k)}$ and $h^{(k)}$

Starting from $g_0^{(k)} = 0$ and $h_0^{(k)} = 0$, find $g^{(k)}$ and $h^{(k)}$ by solving subproblems

ASPIN

Solve for $g_i^{(k)}$ and $h_j^{(k)}$ simultaneously

$$G(u^{(k)} - g_i^{(k)}, v^{(k)}) = 0, \tag{5.34}$$

$$H(u^{(k)}, v^{(k)} - h_j^{(k)}) = 0. \tag{5.35}$$

MSPIN

Solve for $g_i^{(k)}$

$$G(u^{(k)} - g_i^{(k)}, v^{(k)}) = 0, \tag{5.36}$$

Solve for $h_j^{(k)}$ with the new $g_i^{(k)}$

$$H(u^{(k)} - g_i^{(k)}, v^{(k)} - h_j^{(k)}) = 0, \tag{5.37}$$

where $i = 0, 1, \ldots$ until

$$\|G(u^{(k)} - g_i^{(k)}, v^{(k)})\| \leqslant \epsilon_{sub-nonlinear-rtol} \|G(u^{(k)} - g_0^{(k)}, v^{(k)})\|,$$

and $j = 0, 1, \ldots$ until

$$\|H(u^{(k)}, v^{(k)} - h_j^{(k)})\| \leqslant \epsilon_{sub-nonlinear-rtol} \|H(u^{(k)}, v^{(k)} - h_0^{(k)})\|,$$

or

$$\|H(u^{(k)} - g^{(k)}, v^{(k)} - h_j^{(k)})\| \leqslant \epsilon_{sub-nonlinear-rtol} \|H(u^{(k)} - g^{(k)}, v^{(k)} - h_0^{(k)})\|,$$

where $g^{(k)}$ is the solution of (5.36).

the INB method. It is noted that we need the solution $g^{(k)}$ of the submodel (5.36) before we solve the second submodel (5.37).

In step 2, no preconditioning is employed for Krylov subspace iterative methods such as GMRES when we solve the global Jacobian system (5.40) using the INB framework. In fact, nonlinear preconditioning automatically offers a linear preconditioner for the original Jacobian system. The Jacobian formulae (5.13) and (5.29) correspond to the block Jacobi preconditioning and the block Gauss-Seidel preconditioning for the original unpreconditioned equation, respectively.

Algorithm 7 Field-split ASPIN and MSPIN for problems with two components

Let $\alpha \in (0,1)$ and $0 < \theta_{min} < \theta_{max} < 1$ be given.

1. Compute the nonlinear residual $\mathcal{F}(x^{(k)})$ by solving the submodels.

   (a) Find $g^{(k)}$ and $h^{(k)}$ by solving subproblems using Algorithm 6.

   (b) Form the global residual

   $$\mathcal{F}(x^{(k)}) = \begin{bmatrix} g^{(k)} \\ h^{(k)} \end{bmatrix}. \tag{5.38}$$

   (c) Check the stopping conditions on $\mathcal{F}(x^{(k)})$

   $$\|\mathcal{F}(x^{(k)})\| \leqslant \epsilon_{global-nonlinear-rtol}\|\mathcal{F}(x^{(0)})\|. \tag{5.39}$$

2. The initial guess is zero for $d^{(k)}$. Find the inexact Newton direction $d^{(k)}$ by approximately solving

   $$\hat{\mathcal{J}}d^{(k)} = \mathcal{F}(x^{(k)}), \tag{5.40}$$

   such that

   $$\|\mathcal{F}(x^{(k)}) - \hat{\mathcal{J}}d^{(k)}\| \leq \epsilon_{global-linear-rtol}\|\mathcal{F}(x^{(k)})\|, \tag{5.41}$$

   where $\hat{\mathcal{J}}$ has the form (5.13) or (5.29) corresponding to ASPIN and MSPIN, respectively.

3. Compute the new approximate solution

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

   where the step length $\lambda^{(k)}$ is determined by performing linesearch along $d^{(k)}$.

5.2.2 A Simple Example

Following [42], we present a simple example that demonstrates how the field-split preconditioned inexact Newton algorithm rounds out a complex contour landscape for the function whose root is sought.

Consider the system $F(x)$ of two nonlinear equations in two unknowns,

$$F_1(x_1, x_2) = (x_1 - x_2^3 + 1)^3 - x_2^3, \tag{5.42}$$

$$F_2(x_1, x_2) = 2x_1 + 3x_2 - 5. \tag{5.43}$$

It is easy to verify that $x^* = [1, 1]^T$ is a root of this system.

By setting

$$F_1(x_1 - \delta_1^J, x_2) = 0, \tag{5.44}$$

$$F_2(x_1, x_2 - \delta_2^J) = 0, \tag{5.45}$$

we can solve explicitly for $\delta_i^J(x_1, x_2)$, $i = 1, 2$, because of the algebraic simplicity of the system, and derive a nonlinearly preconditioned system $\mathcal{F}_J(x)$ corresponding to ASPIN as follows:

$$\delta_1^J(x_1, x_2) = x_1 - x_2^3 + 1 - x_2, \tag{5.46}$$

$$\delta_2^J(x_1, x_2) = \frac{2}{3}x_1 + x_2 - \frac{5}{3}. \tag{5.47}$$

Similarly, we derive a nonlinearly preconditioned system $\mathcal{F}_{GS}(x)$ corresponding to MSPIN

$$\delta_1^{GS}(x_1, x_2) = x_1 - x_2^3 + 1 - x_2, \tag{5.48}$$

$$\delta_2^{GS}(x_1, x_2) = \frac{2}{3}x_2^3 + \frac{5}{3}x_2 - \frac{7}{3}. \tag{5.49}$$

by solving

$$F_1(x_1 - \delta_1^{GS}, x_2) = 0, \tag{5.50}$$

$$F_2(x_1 - \delta_1^*, x_2 - \delta_2^{GS}) = 0, \tag{5.51}$$

where δ_1^* is the solution of (5.50). (5.46–5.47) are the components of $\mathcal{F}_J$ and (5.48–5.49) are the components of $\mathcal{F}_{GS}$ in the rootfinding problems $\mathcal{F}_J = 0$ and $\mathcal{F}_{GS} = 0$, respectively. In a practical implementation of the field-split preconditioned inexact Newton algorithm, we usually cannot write down the preconditioned system explicitly,

Table 5.1: The number of nonlinear iterations. The outer global tolerance is 10^{-8}, and the inner component tolerances are both 10^{-3}.

Initial guess x_0	INB	ASPIN	MSPIN
$x_0 = (0,0)^T$	11	7	6
$x_0 = (0,2)^T$	10	7	5
$x_0 = (2,0)^T$	1	8	6
$x_0 = (2,2)^T$	11	7	5

but we solve a Jacobian-free linear system (5.40) for the Newton correction and evaluate the function via submodel solvers.

The three columns of Table 5.1 show, respectively, the number of Newton iterations for solving $F(x) = 0$ using inexact Newton methods with the cubic backtracking technique (INB), for solving $\mathcal{F}_J = 0$, and for solving $\mathcal{F}_{GS} = 0$, respectively, starting from the four different initial guesses: $x_0 = (0,0)^T$, $x_0 = (0,2)^T$, $x_0 = (2,0)^T$, $x_0 = (2,2)^T$. Compared with ASPIN and MSPIN, the number of nonlinear iterations using the INB method is more sensitive to the initial estimates. The respective convergence histories are interpreted in the light of the contours distribution of nonlinear systems that the three methods solve, respectively. From Figures 5.1 through 5.3, it is seen that the preconditioned systems are better balanced, in that the contours are more elliptical.

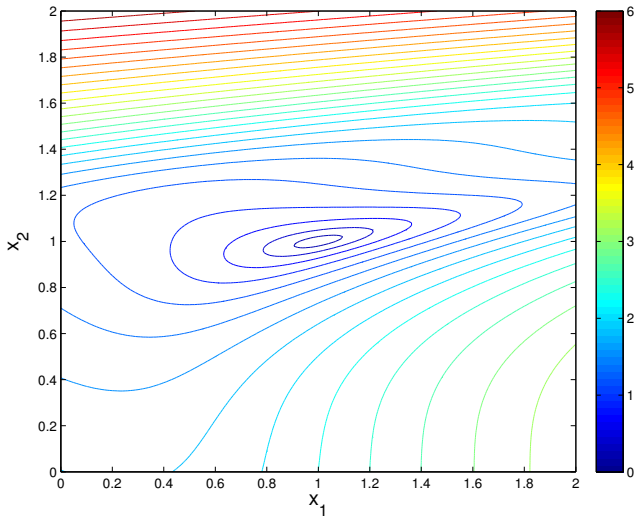

Figure 5.1: Contours of $\log(\|F(x)\| + 1)$

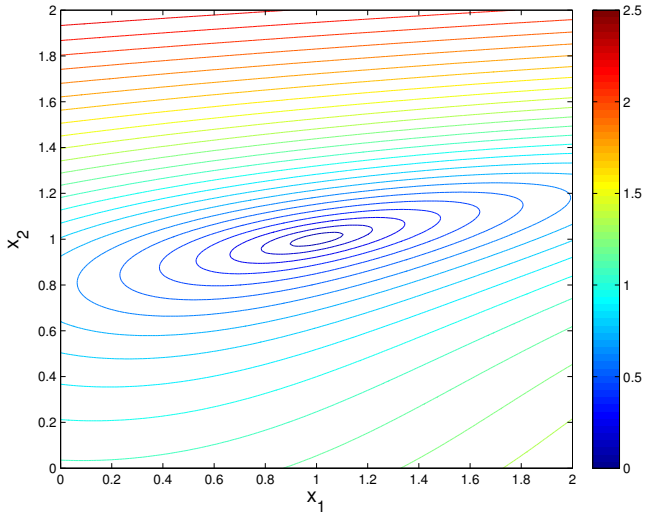

Figure 5.2: Contours of $\log(\|\mathcal{F}_J(x)\| + 1)$

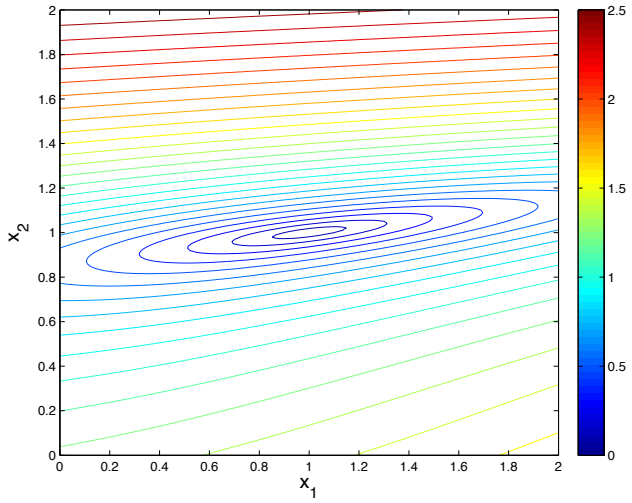

Figure 5.3: Contours of $\log(\|\mathcal{F}_{GS}(x)\| + 1)$

5.3 Some Analysis of the Field-split ASPIN and MSPIN

In this section, under some reasonable assumptions, we show that the preconditioned system (5.2) based on physical variable partition has the same solution as the original system (5.1).

Considering the nonlinear problem (5.1), let $D \subset R^n$ be a neighborhood of the exact solution x^* and we make the following assumptions for $F(x)$:

Assumption 9. *The function $F(x)$ is well-defined in D, and the Jacobian*

$$F'(x) = \begin{bmatrix} G_u & G_v \\ H_u & H_v \end{bmatrix}, \quad x = [u, v]^T, \tag{5.52}$$

is continuous in D and the matrix $F'(x^*)$ is nonsingular. In addition, G_u and H_v are invertible at the exact solution x^*.

Lemma 10. *(see [8]). Assume that $F'(x)$ exists for all $x \in D$, we define a uniformly bounded linear operator $DF(x', x'')$ for all $x', x'' \in D$*

$$DF(x', x'') = \int_0^1 F'(x' + t(x'' - x'))dt, \tag{5.53}$$

such that

$$F(x') - F(x'') = DF(x', x'')(x' - x''). \tag{5.54}$$

If $F'(x)$ is continuous at x^*, then

$$\|DF(x', x'') - F'(x^*)\| \to 0, \quad x', x'' \to x^*. \tag{5.55}$$

It can be shown that these assumptions are satisfied, for instance, for the class of monotone nonlinear elliptic partial differential equations [8, 101]. Under these assumptions, the theory about the local solvability of subproblems [8] can be applied for the cases (ASPIN or MSPIN) based on physics-based partition. Here we briefly provide the proof for our special cases.

Lemma 11. *Under Assumption 9, the nonlinear submodels (5.3), (5.4), (5.18), (5.19) are all uniquely solvable in a sufficiently small neighborhood U of the exact solution x^* in D.*

Proof. We show unique solvability for the submodel (5.19) here, and other cases hold due to Theorem 1.1 in [8]. Let $x = [u, v] \in D \subset R^n$, where $u \in D_1 \subset R^{n_1}$ and $v \in D_2 \subset R^{n_2}$, $n_1 + n_2 = n$. Let

$$S = \{1, \ldots, n\} \tag{5.56}$$

be an index set, and we define $S_H = \{i_1, i_2, \ldots, i_{n_2}\} \subset S$ as

$$S_H = \{i \mid F_i \text{ belongs to the function } H, i \in S\} \tag{5.57}$$

and a restriction matrix $R_H \in R^{n_2 \times n}$ is defined by

$$(R_H)_{k,l} = \begin{cases} 1, & l = i_k, \\ 0, & l \neq i_k, \end{cases} \tag{5.58}$$

and then $H(x) = R_H F(x)$ in (5.1). According to Assumption 4.1, $H_v(x^*)$ is invertible. We choose a sufficiently small neighborhood D' of the exact solution $x^*$and let $\hat{x} = [\hat{u}, \hat{v}]^T$. Then we describe a mapping $\phi : D_2 \times D' \to D_2$ defined by:

$$\phi(h, \hat{x}) = h + H_v(x^*)^{-1} H(\hat{u} - \hat{g}, \hat{v} - h), \tag{5.59}$$

where $\hat{g}$ is the solution of

$$G(\hat{u} - g, \hat{v}) = 0. \tag{5.60}$$

It should be noted that we must have $\hat{g} \to 0$ as $\hat{x} \to x^*$ due to the continuity assumption on F and the local unique solvability of (5.60).

Next, we prove that the mapping ϕ is a contraction with respect to h.

$$\begin{aligned}\phi(h',\hat{x})-\phi(h'',\hat{x}) &= h'-h''+H_v(x^*)^{-1}R_H[F(\hat{u}-\hat{g},\hat{v}-h')-F(\hat{u}-\hat{g},\hat{v}-h'')]\\ &= [I-H_v(x^*)^{-1}R_HDF(x',x'')R_H^T](h'-h''),\end{aligned}$$

where R_H^T is the transpose of the matrix R_H defined in (5.58), $x'=[\hat{u}-\hat{g},\hat{v}-h']^T$ and $x''=[\hat{u}-\hat{g},\hat{v}-h'']^T$. Under the Assumption 4.1 and (5.55), we have $DF(x',x'')\rightarrow F'(x^*)$ as $h',h''\rightarrow 0$ and $\hat{x}\rightarrow x^*$. Hence, there exists a neighborhood $U'\subset D$ of x^* and a positive real number r, such that

$$\|I-H_v(x^*)^{-1}R_HDF(x',x'')R_H^T\|\leqslant\frac{1}{2} \tag{5.61}$$

holds for $\hat{x},x',x''\in U'$ and $h',h''\in B_r(\mathbf{0})$, where $B_r(\mathbf{0})$ is an open ball of $\mathbf{0}$ with a radius r. From (5.59), we have $\phi(0,\hat{x})=H_v(x^*)^{-1}H(\hat{u}-\hat{g},\hat{v})$. Since $H(x^*)=0$ and $\hat{g}\rightarrow 0$ as $\hat{x}\rightarrow x^*$, which implies $[\hat{u}-\hat{g},\hat{v}]^T\rightarrow x^*$ as $\hat{x}\rightarrow x^*$, we could choose a suitable neighborhood U'' of x^* such that

$$\|\phi(0,\hat{x})\|\leqslant\frac{1}{2}r \tag{5.62}$$

hold as long as $\hat{x}\in U''$. Let $U\subset U'\bigcap U''$ be a sufficiently small neighborhood; then, according to the local version of the well-known Banach fixed point theorem (the Corollary 1.2 in [102]), there exists a unique fixed point h^* close to $\mathbf{0}$ that satisfies $h^*=\phi(h^*,\hat{x})$, which implies $H(\hat{u}-\hat{g},\hat{v}-h^*)=0$ in a small neighborhood U of x^*. □

Lemma 12. *There exists a neighborhood $\hat{D}\subset D$ of x^*, such that the Jacobian matrix $\mathcal{J}$ defined by (5.12) or (5.28) is nonsingular for any $x\in\hat{D}$.*

Proof. Since $F'(x)$ is continuous and $G_u(x^*)$ and $H_v(x^*)$ are invertible, Lemma 2.3.3 in [54] shows that there exists a neighborhood $D'\subset D$ of x^* such that G_u and H_v are invertible, and G_u^{-1} and H_v^{-1} are continuous in D'. Let $p=u-g(u,v)$ and

$q = v - h(u,v)$, $x = [u,v]^T \in D$; we know that $g, h \to 0$ as $x \to x^*$ due to the continuity of F.

For the Jacobian matrix $\mathcal{J}$ defined by (5.12), we have

$$\lim_{x \to x^*} \mathcal{J}(u,v) = \begin{bmatrix} G_u^{-1}(x^*) & \\ & H_v^{-1}(x^*) \end{bmatrix} F'(x^*) = \mathcal{J}(x^*). \tag{5.63}$$

For the Jacobian matrix $\mathcal{J}$ defined by (5.28), we have

$$\lim_{x \to x^*} \mathcal{J}(u,v) = \begin{bmatrix} G_u(x^*) & \\ H_u(x^*) & H_v(x^*) \end{bmatrix}^{-1} F'(x^*) = \mathcal{J}(x^*). \tag{5.64}$$

Whether (5.63) or (5.64), $\mathcal{J}(x^*)$ is nonsingular. Hence, we may choose a suitable neighborhood $\hat{D} \subset D' \subset D$ of x^* such that $\mathcal{J}$ is nonsingular for any $x \in \hat{D}$. □

It should be pointed out that the regularity of $\mathcal{J}$ in (5.28) close to x^* has been given in [103], but it is obtained in a different way. Using Lemma 11 and Lemma 12, we have the following theorem.

Theorem 13. *Under Assumption 9, the original system (5.1) and the preconditioned system (5.2) based on physical variable partition have the same solution in a neighborhood of x^* in D.*

Proof. Assume that $x^* = [u^*, v^*]^T$ is the exact solution of the original system (5.1); we have

$$G(u^*, v^*) = 0, \quad H(u^*, v^*) = 0. \tag{5.65}$$

According to the definition of the physical variable correction in (5.3), (5.4), (5.18) and (5.19), we have

ASPIN		MSPIN	
$G(u^* - g(u^*, v^*), v^*) = 0,$	(5.66)	$G(u^* - g(u^*, v^*), v^*) = 0,$	(5.68)
$H(u^*, v^* - h(u^*, v^*)) = 0.$	(5.67)	$H(u^* - g(u^*, v^*), v^* - h(u^*, v^*)) = 0.$	(5.69)

Due to the local unique solvability shown in Lemma 4.3 and comparing (5.65) with (5.66), (5.67), (5.68), (5.69), whether in ASPIN or MSPIN, we have $g(u^*, v^*) = 0$ and $h(u^*, v^*) = 0$, i.e., x^* is a solution of the preconditioned system $\mathcal{F}$ in (5.2).

We next prove that x^* is also a unique solution of $\mathcal{F}$ in some neighborhood $\hat{U}$ of x^*. From Lemma 4.4 and Proposition 2.1 in [44], it is shown that $\mathcal{F}$ is a continuously differentiable function and $\mathcal{F}'$ is nonsingular in a neighborhood $U_1 \subset \hat{D}$ of x^* ($\hat{D}$ from Lemma 4.4). According to the inverse function theorem of calculus, the solution is unique in a neighborhood $U_2 \subset U_1 \subset \hat{D}$.

Hence, there exists a neighborhood $U' \subset U_2 \bigcap U$ (U from Lemma 4.3) such that F and $\mathcal{F}$ have the same solution x^* in U'. □

5.4 Extension to Multicomponent for ASPIN and MSPIN

We can generalize the notation and theory to $N > 2$ components for both ASPIN and MSPIN algorithms. The extension for the ASPIN framework is straightforward. Next we provide the extension of the Jacobian form (5.28) of the preconditioned systems corresponding to the MSPIN framework.

The nonlinear function $F(x)$ is split into $N \geqslant 2$ components representing different

physical aspects as

$$F(x) = \begin{bmatrix} \hat{F}_1(u_1, \ldots, u_N) \\ \vdots \\ \hat{F}_N(u_1, \ldots, u_N) \end{bmatrix} = 0, \quad x = [u_1, \ldots, u_N]^T. \tag{5.70}$$

The preconditioned function

$$\mathcal{F}_{GS}(x) = \begin{bmatrix} T_1(u_1, \ldots, u_N) \\ \vdots \\ T_N(u_1, \ldots, u_N) \end{bmatrix} \tag{5.71}$$

is obtained by solving the following equations sequentially

$$\begin{aligned} &\hat{F}_1(\delta_1, u_2, u_3, \ldots, u_N) = 0, \\ &\hat{F}_2(\delta_1, \delta_2, u_3, u_4 \ldots, u_N) = 0, \\ &\qquad \vdots \\ &\hat{F}_N(\delta_1, \delta_2, \delta_3, \delta_4, \ldots, \delta_N) = 0, \end{aligned} \tag{5.72}$$

where $\delta_i = u_i - T_i$, $i = 1, 2, \ldots, N$.

We define

$$\hat{u}_i = \begin{bmatrix} u_1 \\ \vdots \\ u_i \end{bmatrix}, \quad \hat{u}_i^c = \begin{bmatrix} u_{i+1} \\ \vdots \\ u_N \end{bmatrix}, \quad \hat{T}_i = \begin{bmatrix} T_1 \\ \vdots \\ T_i \end{bmatrix}, \quad \hat{\delta}_i = \begin{bmatrix} \delta_1 \\ \vdots \\ \delta_i \end{bmatrix}, \tag{5.73}$$

and then $\hat{F}_i$ in (5.72) is written as

$$\hat{F}_i(\hat{u}_i - \hat{T}_i, \hat{u}_i^c) = 0, \tag{5.74}$$

where $i = 1, 2, \ldots, N$. Taking the derivative of (5.74) with respect to $\hat{u}_i$, we have

$$\frac{\partial \hat{F}_i}{\partial \hat{\delta}_i}(I_i - \frac{\partial \hat{T}_i}{\partial \hat{u}_i}) = 0, \tag{5.75}$$

where I_i is the identity matrix that has the same dimension as the $\hat{u}_i$ block. Next, we take the derivative of (5.74) with respect to $\hat{u}_i^c$, yielding

$$\frac{\partial \hat{F}_i}{\partial \hat{\delta}_i}(-\frac{\partial \hat{T}_i}{\partial \hat{u}_i^c}) + \frac{\partial \hat{F}_i}{\partial \hat{u}_i^c} = 0. \tag{5.76}$$

Using (5.75) and (5.76), we have

$$\begin{bmatrix} \frac{\partial \hat{F}_i}{\partial \hat{\delta}_i} & \frac{\partial \hat{F}_i}{\partial \hat{u}_i^c} \end{bmatrix} = \frac{\partial \hat{F}_i}{\partial \hat{\delta}_i} \begin{bmatrix} \frac{\partial \hat{T}_i}{\partial \hat{u}_i} & \frac{\partial \hat{T}_i}{\partial \hat{u}_i^c} \end{bmatrix}, \tag{5.77}$$

from which we deduce that

$$A = \begin{bmatrix} \frac{\partial \hat{F}_1}{\partial \delta_1} & \frac{\partial \hat{F}_1}{\partial u_2} & \frac{\partial \hat{F}_1}{\partial u_3} & \cdots & \cdots & \frac{\partial \hat{F}_1}{\partial u_N} \\ \frac{\partial \hat{F}_2}{\partial \delta_1} & \frac{\partial \hat{F}_2}{\partial \delta_2} & \frac{\partial \hat{F}_2}{\partial u_3} & \cdots & \cdots & \frac{\partial \hat{F}_2}{\partial u_N} \\ \vdots & \vdots & \vdots & & & \vdots \\ \frac{\partial \hat{F}_N}{\partial \delta_1} & \frac{\partial \hat{F}_N}{\partial \delta_2} & \frac{\partial \hat{F}_N}{\partial \delta_3} & \cdots & \cdots & \frac{\partial \hat{F}_N}{\partial \delta_N} \end{bmatrix} \tag{5.78}$$

$$= \begin{bmatrix} \frac{\partial \hat{F}_1}{\partial \delta_1} & & & & \\ \frac{\partial \hat{F}_2}{\partial \delta_1} & \frac{\partial \hat{F}_2}{\partial \delta_2} & & & \\ \vdots & \vdots & \ddots & & \\ \frac{\partial \hat{F}_N}{\partial \delta_1} & \frac{\partial \hat{F}_N}{\partial \delta_2} & \frac{\partial \hat{F}_N}{\partial \delta_3} & \cdots & \frac{\partial \hat{F}_N}{\partial \delta_N} \end{bmatrix} \begin{bmatrix} \frac{\partial T_1}{\partial u_1} & \cdots & \frac{\partial T_1}{\partial u_N} \\ \vdots & & \vdots \\ \frac{\partial T_N}{\partial u_1} & \cdots & \frac{\partial T_N}{\partial u_N} \end{bmatrix}.$$

Hence, the Jacobian of the preconditioned system (5.71) corresponds to the block Gauss-Seidel linear preconditioning for the matrix A.

5.5 Numerical Results

In this section, we apply a 2-component physics-based partitioning strategy to three model boundary value problems that are paradigmatic of its uses in practice. In the first, which involves a single unknown with different behavior in different subdomains of *a priori* known locations, one partition contains unknowns comprising an interior spike, and the other contains the unknowns in the regions of solution smoothness. In the second, which is a prototype for coupling in multi-physics problems, one partition contains unknowns involved in a nonlinear ODE, and the other contains the unknowns in an algebraic system, serving as coefficients in the ODE. In the third, one partition contains all of the vorticity unknowns in a Navier-Stokes problem, which come from a nonlinear PDE with the velocities as coefficients, and the other contains the velocity unknowns, which satisfy linear equations for a given vorticity distribution.

5.5.1 Nonlinear Boundary Value Problem

In this example, we consider a nonlinear boundary value problem [24] given by

$$-u'' + u^3 + (4 \times 10^8 (x - 0.5)^2 - 2 \times 10^4)u - 10^9 e^{-3(\frac{x-0.5}{0.01})^2} = 0, \ x \in (0, 1), \quad (5.79)$$

$$u(0) = 0, \qquad u(1) = 0, \quad (5.80)$$

which has a solution satisfying the boundary conditions to well within the machine-roundoff

$$u(x) = 10^3 e^{-(\frac{x-0.5}{0.01})^2}. \quad (5.81)$$

There is a large spike around $x = 0.5$ as shown in Figure 5.4. Using the usual second-order central difference approximation for u'', the discretization of this ODE on a fine uniform grid results in a system of nonlinear equations. The resulting nonlinear problem is solved using three methods: the inexact Newton backtracking method

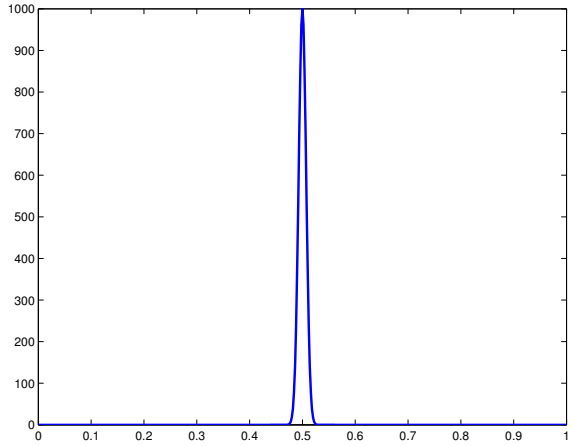

Figure 5.4: The exact solution $u(x)$.

(INB), ASPIN and MSPIN. For the latter two, the equations are split into two sets denoted by G and H, where the system G refers to the equations discretized at the points around $x = 0.5$.

We clarify the details of the field-split preconditioned inexact Newton algorithms for this example using 100 discretized equations. The system G consists of the discretized equations 48-53 with unknowns u_{48}-u_{53}, and other equations belong to the system H with unknowns u_1-u_{47} and u_{54}-u_{100}. The values for u_{47} and u_{54} from the previous iterate for H are treated as the boundary values for the system G, and the values of u_{48} and u_{53} from the previous iterate for G as the boundary values for the system H. The initial guess is zero for $u_1, u_2, \ldots, u_{100}$. Following the experience in [37], the global system is solved using the INB method with the parameters $\epsilon_{global-nonlinear-rtol} = 10^{-6}$, $\epsilon_{global-linear-rtol} = 10^{-8}$, and $\epsilon_{sub-nonlinear-rtol} = 10^{-3}$ is the stopping condition for the subproblems nonlinear iterations.

Table 5.2 shows the number of Newton iterations for the different methods on problems of different resolution. The nonlinearly preconditioned Newton's methods are effective in reducing the number of global nonlinear iterations.

Table 5.2: Comparison of the number of nonlinear iterations for different methods in the rightmost three columns. "No. points" indicates the number of grid points used to discretize the ODE. The points between G_{left} and G_{right} are unknowns in G.

No. points	G_{left}	G_{right}	Its. INB	Its. ASPIN	Its. MSPIN
100	48	53	5	2	1
500	236	265	5	2	2
1000	471	530	6	2	1
5000	2351	2650	5	2	2

5.5.2 An ODE Coupled to an Algebraic System in 1D

We consider an ODE coupled to an algebraic system in 1D:

$$-(vu_x)_x + \lambda u^2 = 1 \text{ on } (0,1), \quad \text{subject to } u(0) = 0,\ u(1) = 1, \tag{5.82}$$

$$\exp(v-1) + v = \frac{1}{\frac{1}{1+u} + \frac{1}{1+u_x^2}}, \tag{5.83}$$

where u_x is discretized using forward differences, and the discretization places v at staggered points. This is example #28 in the scalable nonlinear equations solvers subdirectory of PETSc, modified by the addition of a quadratic term in u, so that both components are nonlinear after splitting. Fifty grid points are used for this problem. The function $u_0(x) = x(1-x)$ and $v_0(x) = 1 + 0.5\sin(2\pi x)$ are used to be the initial guesses for $u(x)$ and $v(x)$. Considering the partition with respect to u and v, we split the problem into two submodels G and H that are dominant in u and v, respectively. The system G consists of the discretized equations of (5.82) with u-unknowns using the most recent values of v as the coefficients, while the discretized equations of (5.83) with v-unknowns belong to the system H, where the values of u_x are computed using the most recent values of u. We set the tolerance parameters as $\epsilon_{global-nonlinear-rtol} = 10^{-10}$, $\epsilon_{global-linear-rtol} = 10^{-8}$, and $\epsilon_{sub-nonlinear-rtol} = 10^{-3}$ is the stopping condition for the subproblems nonlinear iterations. The analytical

Table 5.3: Global nonlinear and linear iterations using globalized INB, ASPIN and MSPIN. $\epsilon_{global-nonlinear-rtol} = 10^{-10}$, $\epsilon_{global-linear-rtol} = 10^{-8}$, and $\epsilon_{sub-nonlinear-rtol} = 10^{-3}$. " $*$" indicates that linear iterations are not available, since the nonlinear methods stagnate at the line search.

	Number of PIN iterations				
Methods	$\lambda = 0$	$\lambda = 100$	$\lambda = 1000$	$\lambda = 3000$	$\lambda = 5000$
INB	6	6	15	-	9
ASPIN	6	-	-	-	-
MSPIN	5	5	5	5	5
	Average number of GMRES iterations per PIN				
INB	12	18	14	$*$	7
ASPIN	21	$*$	$*$	$*$	$*$
MSPIN	10	9	8	8	7

Jacobian matrices are used in this example.

Table 5.3 shows the number of Newton iterations and linear iterations for the different methods on problems with different parameters λ. The MSPIN method is effective in reducing the number of global Newton iterations, but the ASPIN method is very sensitive to the parameter λ. The failure of the ASPIN method and the INB method happens when the line search fails.

5.5.3 Driven Cavity Flow Problem

We consider the two-dimensional lid-driven cavity flow problem [17, 28, 35, 104] in the domain $\Omega = (0,1)\times(0,1)$ with three unknowns: the velocity u, v and the vorticity ω.

$$\begin{cases} -\Delta u - \frac{\partial \omega}{\partial y} = 0, \\ -\Delta v + \frac{\partial \omega}{\partial x} = 0, \\ -\frac{1}{Re}\Delta \omega + u\frac{\partial \omega}{\partial x} + v\frac{\partial \omega}{\partial y} = 0. \end{cases} \tag{5.84}$$

Here $u = 1, v = 0$ on the top boundary and $u = 0, v = 0$ on the other boundaries. The boundary condition on ω is given by its definition:

$$\omega(x, y) = -\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}. \tag{5.85}$$

Considering the partition with respect to velocity unknowns and the vorticity unknowns, we split the system (5.84) into two submodels

$$G: \quad \begin{cases} -\Delta u - \frac{\partial \omega}{\partial y} = 0, \\ \\ -\Delta v + \frac{\partial \omega}{\partial x} = 0, \end{cases} \tag{5.86}$$

and

$$H: \quad -\frac{1}{Re}\Delta\omega + u\frac{\partial \omega}{\partial x} + v\frac{\partial \omega}{\partial y} = 0. \tag{5.87}$$

A finite difference scheme with the 5-point stencil is used to discretize the PDEs, which is a standard PETSc example [34]. Upwinding is employed in the vorticity equation and the vorticity boundary condition is differenced inward with respect to the normal direction. It is noted that this is only a first-order discretization. As the Reynolds number is increased on a fixed mesh, the discretization artificially diffuses the boundary layers, but point of pushing the Reynolds number beyond the discretization is to obtain parameterized nonlinear algebraic problems of fixed size that are increasingly difficult for (unpreconditioned) globalized Newton methods.

Here, the subproblems obtained from the discretization of (5.86) and (5.87) are linear, which are solved by GMRES with ILU(0) preconditioner. We set the tolerance parameters as $\epsilon_{global-nonlinear-rtol} = 10^{-10}$, $\epsilon_{global-linear-rtol} = 10^{-6}$, $\epsilon_{sub-linear-rtol} = 10^{-3}$. The Jacobian matrices are formed using a finite difference scheme. We run the test for the uniform meshes 64×64, 128×128, and 256×256.

Newton can be sensitive to the initial guess. Our first tests set the initial guess to be zero for u, v, and ω. Table 5.4 compares a global inexact Newton-GMRES-

Table 5.4: Global nonlinear and linear iterations using globalized INB, ASPIN, and MSPIN on different mesh sizes. The initial guess is zero for u, v, and ω. $\epsilon_{global-linear-rtol} = 10^{-6}$, $\epsilon_{global-nonlinear-rtol} = 10^{-10}$. The finite difference step size for the matrix-free Jacobian applications is 10^{-8}. " $**$" indicates that nonlinear iterations are not available, because linear iterations fail to converge after 10000 steps.

64×64 mesh			
Number of PIN iterations			
Methods	$Re = 10$	$Re = 100$	$Re = 1000$
INB	3	5	17
ASPIN	4	9	10
MSPIN	4	5	4
Average number of GMRES iterations per PIN			
INB	19	24	26
ASPIN	23	37	39
MSPIN	13	18	15
128×128 mesh			
Number of PIN iterations			
Methods	$Re = 10$	$Re = 100$	$Re = 1000$
INB	3	5	$**$
ASPIN	4	9	13
MSPIN	4	5	5
Average number of GMRES iterations per PIN			
INB	37	62	-
ASPIN	24	42	64
MSPIN	14	21	20
256×256 mesh			
Number of PIN iterations			
Methods	$Re = 10$	$Re = 100$	$Re = 1000$
INB	3	5	$**$
ASPIN	4	10	18
MSPIN	4	6	6
Average number of GMRES iterations per PIN			
INB	93	200	-
ASPIN	25	47	139
MSPIN	15	24	28

ILU with backtracking (INB) against ASPIN and MSPIN. (A fill level of 6 is chosen for ILU since level 5 or 6 minimizes runtime for the linear solver overall across all convergent experiments, balancing cost per iteration against the number of iterations.) For ASPIN and MSPIN, the two linear problems involved in (5.13) and (5.29) are solved directly. For small Reynolds numbers, corresponding to weak nonlinearity, the

INB can have the smallest number of outer Newton steps. However, the advantage is quickly lost with increasing Reynolds number and for modest Reynolds number, INB fails to converge nonlinearly altogether even with exact linear solves. The threshold for the 128×128 mesh for this problem and discretization was determined to be $Re = 770.0$ in [28]. Both the field-split preconditioned inexact Newton methods converge for all Reynolds numbers. The MSPIN method is superior in nonlinear iterations to the ASPIN method. The average number of GMRES iterations increases when the mesh size is increased for a fixed Reynolds number. In contrast, the number of nonlinear iterations using the ASPIN method increases, and the average number of GMRES iterations increases considerably, when the mesh size varies from 64×64 to 128×128 to 256×256.

Next we change the initial guess. The initial guess is still zero for u, v, and ω, except $u = 1$ on the top boundary. The ASPIN method fails to converge once the Reynolds number passes the value $Re = 1136$ on the 128×128 mesh, and suffers from stagnation before the Reynolds number reaches 1136, while MSPIN converges for a much larger range of Reynolds numbers, as shown in Figure 5.5. The MSPIN method is not very sensitive to changes of parameters via the tests above, such as the initial guess, the mesh size, and the Reynolds number and is therefore more robust than the ASPIN method.

The nonlinear preconditioning demonstrated here is one of several globalization techniques for systems for which INB fails if applied directly, such as the driven cavity on a fine mesh at high Reynolds number. One may apply mesh sequencing, initializing the fine mesh with converged results from recursively coarsened meshes. One may invoke parameter continuation, initializing the high Reynolds problem with solutions at recursively lower values (noting that at infinitesimal Reynolds number, the problem becomes linear and SPD in each variable). One may also apply pseudo-transient continuation by prepending a temporal evolution term to the vorticity transport equa-

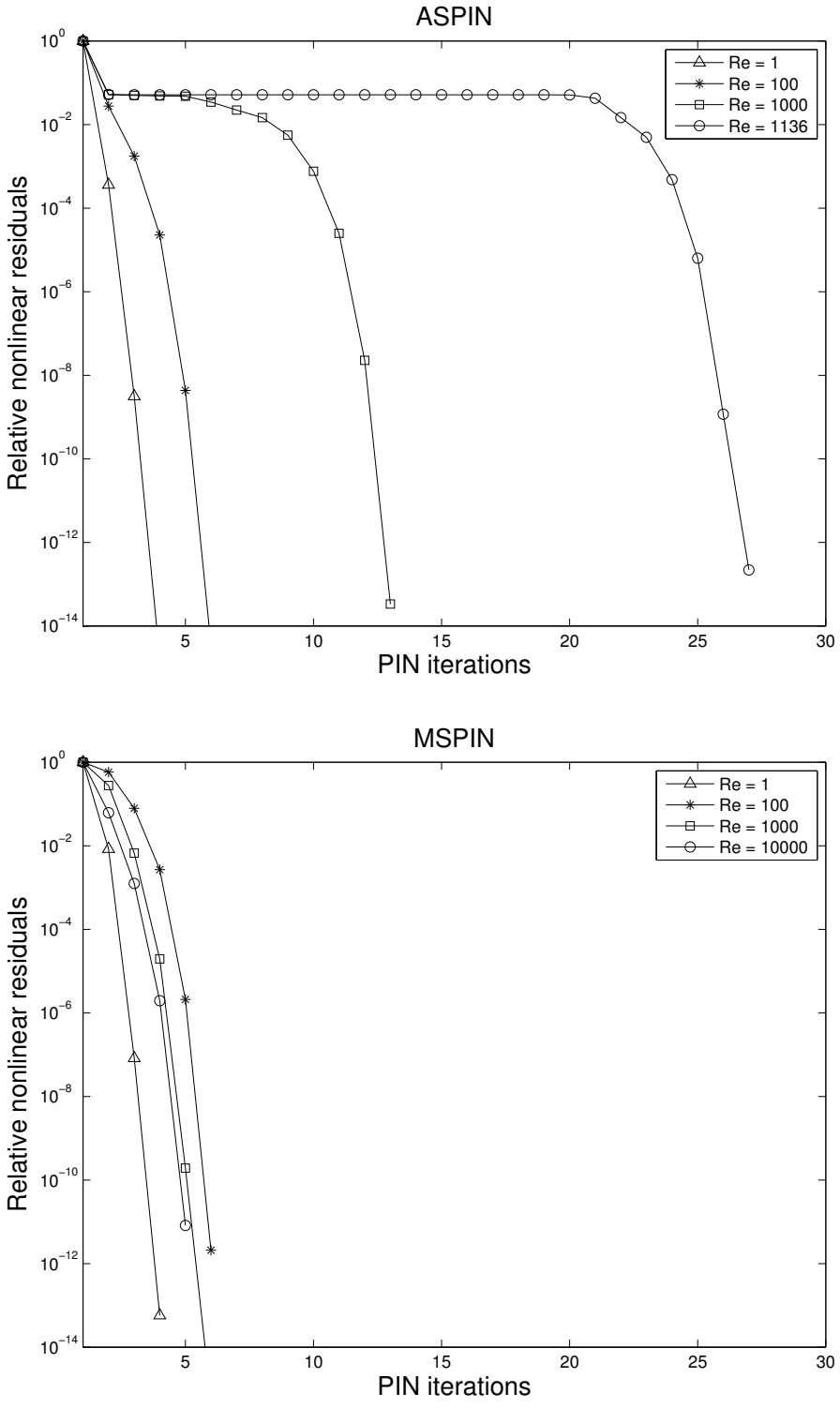

Figure 5.5: Nonlinear residual history for the driven cavity flow problem with different Reynolds numbers. The initial guess is still zero for u, v, ω, except $u = 1$ on the top boundary.

Table 5.5: Execution times for strong scaling of the lid-driven cavity for ASPIN and MSPIN on a 256×256 mesh at Reynolds number 1000, for tight and loose relative convergence tolerances on the subproblems and global preconditioner linear systems solutions. The initial guess is zero for u, v, and ω. $\epsilon_{global-linear-rtol} = 10^{-6}$, $\epsilon_{global-nonlinear-rtol} = 10^{-8}$. $\epsilon_{sub-rtol}$ denotes the relative tolerance for the subproblems (which are linear in this example), and we specify $\epsilon_{Jac-rtol}$ as the relative tolerance for the linear problems in (5.13) and (5.29). The finite difference step size for the matrix-free Jacobian applications is 10^{-8}. " N_p" indicates the number of processors, which does not have to be square. Performance for INB is not shown since it fails to converge on this mesh and this Reynolds number from a zero initial guess.

Execution time (s)					
		256×256 mesh			
Methods	N_p	$\epsilon_{sub-rtol} = 10^{-3}$ $\epsilon_{Jac-rtol} = 10^{-3}$	$\epsilon_{sub-rtol} = 10^{-3}$ $\epsilon_{Jac-rtol} = 10^{-6}$	$\epsilon_{sub-rtol} = 10^{-6}$ $\epsilon_{Jac-rtol} = 10^{-3}$	$\epsilon_{sub-rtol} = 10^{-6}$ $\epsilon_{Jac-rtol} = 10^{-6}$
ASPIN	4	2363.98	3273.88	2194.43	3219.34
	16	687.68	943.91	654.32	976.95
	32	397.12	589.86	395.26	588.95
	64	272.4	412.27	276.03	405.92
MSPIN	4	175.31	245.59	199.64	248.18
	16	57.04	74.06	56.74	74.48
	32	32.17	45.86	34.55	46.33
	64	22.31	31.64	23.90	31.79

tion, and approaching the desired steady state through a physically motivated transient, beginning with a very small timestep and adaptively increasing it in inverse proportion to fractional power of steady-state residual reductions. These and other strategies are discussed in [1], with references to the literature. Our purpose in this contribution is to offer a complementary technique to desensitize Newton to the initial guess and improve the nonlinear conditioning in a fundamental way that may, if necessary, be further combined with the others.

Finally we show some parallel results on a 16-rack IBM Blue Gene/P supercomputer with quad-core PowerPC 450 I/O nodes (850MHz, 4GB RAM). In Table 5.5, we consider the most nonlinear and the most linearly ill-conditioned problem of Table 5.4, on which INB does not converge unassisted. On this, we demonstrate modest strong scalability for ASPIN and MSPIN within the PETSc framework. For ASPIN and MSPIN, all subproblems and linear problems involved in the nonlinearly

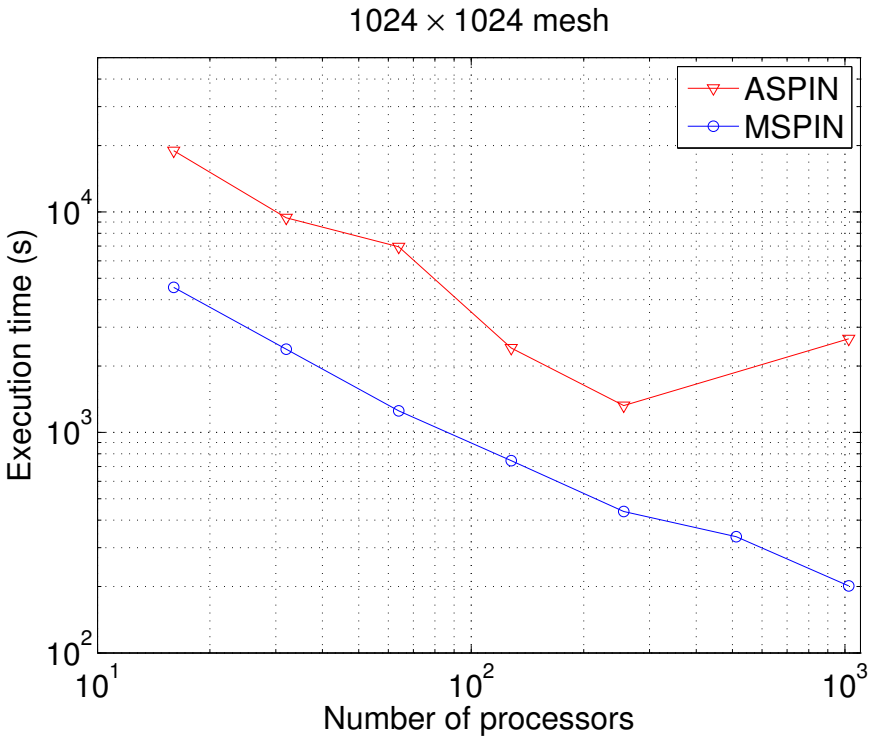

Figure 5.6: Strong scaling for the driven cavity flow problem on a 1024×1024 mesh at Reynolds number 1000. The initial guess is still zero for u, v, ω. $\epsilon_{global-linear-rtol} = 10^{-3}$, $\epsilon_{global-nonlinear-rtol} = 10^{-8}$, $\epsilon_{sub-rtol} = 10^{-3}$ and $\epsilon_{Jac-rtol} = 10^{-3}$. $\epsilon_{sub-rtol}$ denotes the relative tolerance for the subproblems (which are linear in this example), and we specify $\epsilon_{Jac-rtol}$ as the relative tolerance for the linear problems in (5.13) and (5.29). The finite difference step size for the matrix-free Jacobian applications is 10^{-8}. Execution time for ASPIN using 512 processors is not shown since it fails to converge on this mesh and this Reynolds number from a zero initial guess.

preconditioned Jacobian application are solved by GMRES with BoomerAMG preconditioners [105]. We vary the number of processors at Reynolds number 1000 and the scaling behaviors are shown for both methods and all four relative tolerance tunings. Because performance is generally dependent on linear convergence tolerances, we experiment with four combinations of loose and tight tolerances for the individual subproblems and the systems in (5.13) and (5.29). We see that execution times are more sensitive to the outer tolerances than to the inner.

Figure 5.6 shows strong scaling behaviors for ASPIN and MSPIN on a 1024×1024 mesh at Reynolds number 1000, using 16, 32, 64, 128, 256, 512, 1024 processors. For

ASPIN and MSPIN, all subproblems and linear problems involved in the nonlinearly preconditioned Jacobian application are also solved by GMRES with BoomerAMG preconditioners. In terms of the execution time, MSPIN scales well for up to 1024 processors, while ASPIN scales well for up to 256 processors and it fails to converge using 512 processors on this mesh and this Reynolds number from a zero initial guess.

5.6 Orderings and Groupings

In the MSPIN algorithm, we have to decide the field-split partitioning for both variables and equations, and then the subproblems are solved in order. Different orderings and different groupings result in different nonlinear preconditioners. It is shown from the following two examples that orderings and groupings can change the equality of the nonlinear preconditioning — even dramatically.

5.6.1 Driven Cavity Flow Problem

For the driven cavity flow in Section 5.5.3, we switch the ordering as follows:

$$G: \quad -\frac{1}{Re}\Delta\omega + u\frac{\partial\omega}{\partial x} + v\frac{\partial\omega}{\partial y} = 0. \tag{5.88}$$

and

$$H: \quad \begin{cases} -\Delta u - \frac{\partial\omega}{\partial y} = 0, \\ \\ -\Delta v + \frac{\partial\omega}{\partial x} = 0. \end{cases} \tag{5.89}$$

We first solve for the vorticity components ω and then for velocity components u, v. It is noted that we arrange the unknowns in the order of ω, u, v for the global system.

Our first test is to set the zero initial guess for ω, u, v. In terms of the number of outer Newton steps, Table 5.6 shows that the MSPIN method loses a clear advantage over the ASPIN method as is shown in Table 5.4.

Table 5.6: Global nonlinear and linear iterations using ASPIN and MSPIN on different mesh sizes. We arrange the unknowns in the order of ω, u, v for the global system. The initial guess is zero. $\epsilon_{global-linear-rtol} = 10^{-6}$, $\epsilon_{global-nonlinear-rtol} = 10^{-10}$. The finite difference step size for the matrix-free Jacobian applications is 10^{-8}.

64×64 mesh				
	Number of PIN iterations			
Methods	Re = 100	Re = 1000	Re = 5000	Re = 10000
ASPIN	9	10	9	10
MSPIN	9	9	8	9
	Average number of GMRES iterations per PIN			
Methods	Re = 100	Re = 1000	Re = 5000	Re = 10000
ASPIN	37	39	37	41
MSPIN	19	19	18	18
128×128 mesh				
	Number of PIN iterations			
Methods	Re = 100	Re = 1000	Re = 5000	Re = 10000
ASPIN	9	13	12	11
MSPIN	9	12	11	10
	Average number of GMRES iterations per PIN			
Methods	Re = 100	Re = 1000	Re = 5000	Re = 10000
ASPIN	42	65	55	61
MSPIN	21	26	23	23
256×256 mesh				
	Number of PIN iterations			
Methods	Re = 100	Re = 1000	Re = 5000	Re = 10000
ASPIN	10	19	15	14
MSPIN	10	17	15	14
	Average number of GMRES iterations per PIN			
Methods	Re = 100	Re = 1000	Re = 5000	Re = 10000
ASPIN	48	140	106	110
MSPIN	23	43	35	33

Next we change the initial guess. All the unknowns ω, u, v are set to be zero except $u = 1$ on the top boundary. The nonlinear residual history is shown in Figure 5.7. The MSPIN method fails to converge once the Reynolds number passes the value $Re = 2564$ on the 128×128 mesh, because linear iterations fail to converge after 10000 steps. In contrast, Figure 5.5 shows that the MSPIN method works well up to $Re = 10000$ when we first solve for the velocity components u, v and then for the vorticity component ω.

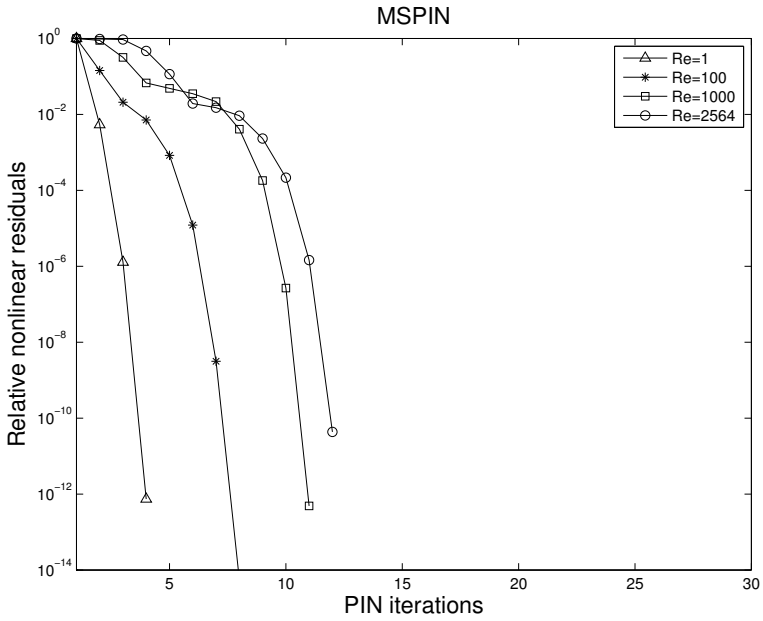

Figure 5.7: Nonlinear residual history for the flow problem with different Reynolds numbers. The initial guess is still zero for u, v, ω, except $u = 1$ on the top boundary. The MSPIN algorithm fails when $Re \geqslant 2565$.

5.6.2 Natural Convection Cavity Flow Problem

We consider a benchmark problem [106] that describes the two-dimensional natural convection cavity flow of a Boussinesq fluid with Prandtl number 0.71 in an upright square cavity $\Omega = (0, 1) \times (0, 1)$. Following [107], the nondimensional steady-state Navier-Stokes equations in vorticity-velocity form and energy equation are formulated as:

$$\begin{cases} -\Delta u - \frac{\partial \omega}{\partial y} = 0, \\ -\Delta v + \frac{\partial \omega}{\partial x} = 0, \\ -(\frac{Pr}{Ra})^{0.5}\Delta\omega + u\frac{\partial \omega}{\partial x} + v\frac{\partial \omega}{\partial y} - \frac{\partial T}{\partial x} = 0, \\ -(\frac{1}{PrRa})^{0.5}\Delta T + u\frac{\partial T}{\partial x} + v\frac{\partial T}{\partial y} = 0, \end{cases} \tag{5.90}$$

where Pr and Ra denote the Prandtl number and the Rayleigh number, respectively. There are four unknowns: the velocity u, v, the vorticity ω and the temperature T.

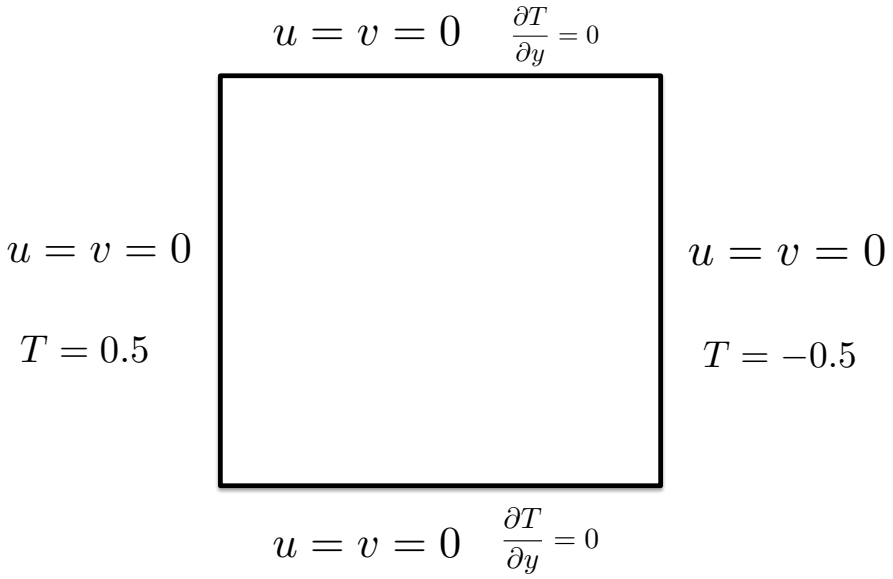

Figure 5.8: Boundary conditions for natural convection cavity flow

On the solid walls (the thickness of the walls are assumed to be zero), both velocity components u, v are zero, and the vorticity is determined from its definition in (5.85). The horizontal walls are insulated ($\frac{\partial T}{\partial y} = 0$), and the vertical walls are at temperatures 0.5 (left) and -0.5 (right). The temperature difference keeps the fluid circulating in the cavity.

Considering the partition with respect to velocity unknowns, the vorticity unknown and the temperature unknowns, we split the system (5.90) into three submodels:

$$F_T: \quad -(\frac{1}{PrRa})^{0.5}\Delta T + u\frac{\partial T}{\partial x} + v\frac{\partial T}{\partial y} = 0, \tag{5.91}$$

$$F_\omega: \quad -(\frac{Pr}{Ra})^{0.5}\Delta \omega + u\frac{\partial \omega}{\partial x} + v\frac{\partial \omega}{\partial y} - \frac{\partial T}{\partial x} = 0, \tag{5.92}$$

$$F_{u,v}: \quad \begin{cases} -\Delta u - \frac{\partial \omega}{\partial y} = 0, \\ \\ -\Delta v + \frac{\partial \omega}{\partial x} = 0. \end{cases} \tag{5.93}$$

The discretization is similar to that in the driven cavity problem, and upwinding is used in the vorticity equation and the temperature equation.

There are four schemes for groupings:

- Grouping A with two subsystems

$$\begin{aligned} \hat{F}_1 &: F_T \\ \hat{F}_2 &: F_\omega,\ F_{u,v} \end{aligned}$$

- Grouping B with two subsystems

$$\begin{aligned} \hat{F}_1 &: F_T,\ F_\omega \\ \hat{F}_2 &: F_{u,v} \end{aligned}$$

- Grouping C with two subsystems

$$\begin{aligned} \hat{F}_1 &: F_T,\ F_{u,v} \\ \hat{F}_2 &: F_\omega \end{aligned}$$

- Grouping D with three subsystems

$$\begin{aligned} \hat{F}_1 &: F_T \\ \hat{F}_2 &: F_\omega \\ \hat{F}_3 &: F_{u,v} \end{aligned}$$

It is noted that the subproblems corresponding to Grouping B and Grouping D obtained from the discretization of equations are linear, which are solved by GMRES with BoomerAMG preconditioners. With Grouping A or Grouping C, one subproblem is linear and the other one is still nonlinear, which is solved by INB with the tolerance $\epsilon_{sub-nonlinear-rtol} = 10^{-4}$.

The initial guess is zero for u, v, and ω, and the linear interpolation for T using two known temperatures on the vertical sides. Figure 5.9 shows contours of temperature T and vorticity ω at different Rayleigh numbers. Table 5.7 compares INB against the MSPIN algorithms corresponding to different groupings. When MSPIN algorithms with Grouping B and Grouping D converge on a fixed mesh at a fixed Rayleigh

Table 5.7: Global nonlinear and linear iterations using INB and MSPIN on different mesh sizes at different Rayleigh numbers. The initial guess is zero for u, v, and ω, and the linear interpolation for T using two known temperatures on the vertical sides. $\epsilon_{global-linear-rtol} = 10^{-6}$, $\epsilon_{global-nonlinear-rtol} = 10^{-10}$, and $\epsilon_{sub-linear-rtol} = 10^{-6}$. "-" indicates that the nonlinear iterations are not available, because linear iterations fail. "*" indicates that the nonlinear iterations are not available, because the subproblems fails to converge or backtracking fails.

	64×64 mesh, 4 processors									
Ra	INB		Grouping A $F_T\|F_\omega, F_{u,v}$		Grouping B $F_T, F_\omega\|F_{u,v}$		Grouping C $F_T, F_{u,v}\|F_\omega$		Grouping D $F_T\|F_\omega\|F_{u,v}$	
	Newton	GMRES	Newton	GMRES	Newton	GMRES	Newton	GMRES	Newton	GMRES
10^3	5	41	4	5	5	17	4	15	5	17
5×10^3	6	46	6	8	7	21	5	19	7	21
10^4		-	*		7	27	8	23	6	27
5×10^4	20	58	*		23	37	14	54	23	37
10^5		-	*		18	61		-	17	65
	128×128 mesh, 16 processors									
Ra	INB		Grouping A $F_T\|F_\omega, F_{u,v}$		Grouping B $F_T, F_\omega\|F_{u,v}$		Grouping C $F_T, F_{u,v}\|F_\omega$		Grouping D $F_T\|F_\omega\|F_{u,v}$	
	Newton	GMRES	Newton	GMRES	Newton	GMRES	Newton	GMRES	Newton	GMRES
10^3	5	133	4	5	5	18	4	16	5	18
5×10^3	6	166	5	8	7	23	5	20	6	23
10^4		-	*		7	28	10	30	7	28
5×10^4		-	*			-	*		9	63
10^5		-	*		18	110		-	16	83
	256×256 mesh, 64 processors									
Ra	INB		Grouping A $F_T\|F_\omega, F_{u,v}$		Grouping B $F_T, F_\omega\|F_{u,v}$		Grouping C $F_T, F_{u,v}\|F_\omega$		Grouping D $F_T\|F_\omega\|F_{u,v}$	
	Newton	GMRES	Newton	GMRES	Newton	GMRES	Newton	GMRES	Newton	GMRES
10^3		-	4	5	5	18	4	16	4	18
5×10^3		-	5	7	7	27	5	21	6	27
10^4		-	*		7	31	9	32	7	31
5×10^4		-	*			-		-	8	82
10^5		-	*			-		-	19	97

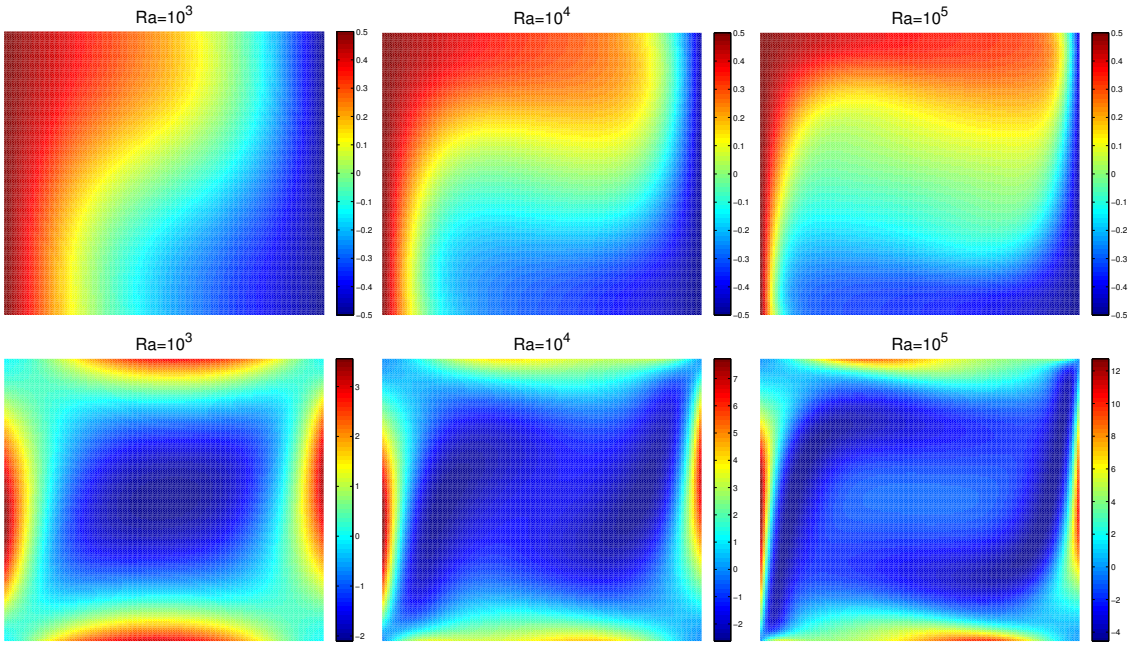

Figure 5.9: Contours of temperature T (top) and vorticity ω (bottom).

number, they have similar numbers of Newton iterations and GMRES iterations. In Table 5.7, MSPIN algorithms with Grouping A, B or C fail to converge in some cases. However, impressively, the MSPIN algorithm with Grouping D works for all the tests. Therefore, the groupings play an essential role in determining the quality of nonlinear preconditioning.

Chapter 6

Convergence Analysis for the MSPIN Algorithm

6.1 MSPIN with Multiple Components

We consider a nonlinear root-finding problem (2.1) and assume that the function $F(x)$ is continuously differentiable. In the multiplicative Schwarz preconditioned inexact Newton (MSPIN) algorithm, the nonlinear function $F(x)$ is split conformally into $2 \leqslant N \leqslant n$ nonoverlapping components representing distinct physical features as

$$F(x) = F(u_1, \ldots, u_N) = \begin{bmatrix} \hat{F}_1(u_1, \ldots, u_N) \\ \vdots \\ \hat{F}_N(u_1, \ldots, u_N) \end{bmatrix} = 0, \tag{6.1}$$

where $x = [x_1, \ldots, x_n]^T = [u_1, \ldots, u_N]^T \in R^n$. u_i and $\hat{F}_i$, denote subpartitions of x and F, respectively, $i = 1, \ldots, N$. In the MSPIN algorithm, the submodels are solved sequentially for the physical variable corrections, and the preconditioned system whose root is to be found consists of the sum of these corrections.

The multiplicative Schwarz preconditioned function

$$\mathcal{F}(x) = \begin{bmatrix} T_1(u_1, \ldots, u_N) \\ \vdots \\ T_N(u_1, \ldots, u_N) \end{bmatrix} \tag{6.2}$$

is obtained by solving the following equations:

$$\begin{aligned} &\hat{F}_1(u_1 - T_1(x), u_2, u_3, \ldots, u_N) = 0, \\ &\hat{F}_2(u_1 - T_1(x), u_2 - T_2(x), u_3, \ldots, u_N) = 0, \\ &\qquad\qquad \vdots \\ &\hat{F}_N(u_1 - T_1(x), u_2 - T_2(x), u_3 - T_3(x), \ldots, u_N - T_N(x)) = 0. \end{aligned} \tag{6.3}$$

As with the ASPIN algorithm in [28], the MSPIN method solves the global preconditioned problem in (6.2) using the INB algorithm.

6.1.1 Some Properties

We discuss key properties of the MSPIN algorithm under reasonable assumptions. Specifically, we prove that the preconditioned function $\mathcal{F}(x)$ is continuously differentiable and we give a formula for the Jacobian $\mathcal{F}'(x)$.

Let

$$S = \{1, \ldots, n\}$$

be an index set, and

$$F(x) = [F_1(x), F_2(x), \ldots, F_n(x)]^T = [\hat{F}_1(x), \hat{F}_2(x), \ldots, \hat{F}_N(x)]^T, \tag{6.4}$$

where $\hat{F}_i(x) = \hat{F}_i(u_1, \ldots, u_N)$ is defined in (6.1). We define $S_i = \{i_1, i_2, \ldots, i_{n_i}\} \subset S$ as

$$S_i = \{j \mid F_j(x) \text{ belongs to the function } \hat{F}_i(x), j \in S\}, \tag{6.5}$$

where $i = 1, 2, \ldots, N$, and $n_1 + n_2 + \ldots + n_N = n$. We define a set of restriction matrices $R_i \in R^{n_i \times n}$ as follows:

$$(R_i)_{k,l} = \begin{cases} 1, & l = i_k, \\ 0, & \text{otherwise}, \end{cases} \tag{6.6}$$

and then we define

$$E_i = R_i^T R_i \in R^{n \times n}, \quad i = 1, \ldots, N. \tag{6.7}$$

It is clear that $F(x) = \sum_{i=1}^{N} R_i^T \hat{F}_i(x)$.

Illustrating with $n = 3$, $N = 2$, let $S_1 = \{1\}$, $S_2 = \{2, 3\}$, $n_1 = 1$, and $n_2 = 2$, to get

$$\hat{F}_1(x) = F_1(u_1, u_2), \quad \hat{F}_2(x) = \begin{bmatrix} F_2(u_1, u_2) \\ F_3(u_1, u_2) \end{bmatrix}, \tag{6.8}$$

where $x = [x_1, x_2, x_3]^T$, $u_1 = x_1$, $u_2 = [x_2, x_3]^T$.

$$R_1 = [1, 0, 0], \quad R_2 = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{6.9}$$

$$E_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad E_2 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{6.10}$$

For the nonlinear problem (6.1), we make the following assumptions for $F(x)$:

Assumption 14. *The function $F(x)$ is well-defined in a neighborhood U of the exact solution x^*,*

(i) the Jacobian

$$F'(x) = J(x) = \begin{bmatrix} \frac{\partial \hat{F}_1}{\partial u_1} & \frac{\partial \hat{F}_1}{\partial u_2} & \frac{\partial \hat{F}_1}{\partial u_3} & \cdots & \frac{\partial \hat{F}_1}{\partial u_N} \\ \frac{\partial \hat{F}_2}{\partial u_1} & \frac{\partial \hat{F}_2}{\partial u_2} & \frac{\partial \hat{F}_2}{\partial u_3} & \cdots & \frac{\partial \hat{F}_2}{\partial u_N} \\ \vdots & \vdots & \vdots & & \vdots \\ \frac{\partial \hat{F}_N}{\partial u_1} & \frac{\partial \hat{F}_N}{\partial u_2} & \frac{\partial \hat{F}_N}{\partial u_3} & \cdots & \frac{\partial \hat{F}_N}{\partial u_N} \end{bmatrix} \tag{6.11}$$

is continuous in U and $J(x^*)$ is nonsingular.

(ii) the submatrices $R_i J(x^*) R_i^T$ are invertible, $i = 1, 2, \ldots, N$.

We now recall results from Lemma 4.3 and Theorem 4.5 in [50] in the following lemma. It guarantees that $\mathcal{F}(x)$ in (6.2) is well defined and continuous, and shows the equivalence of two nonlinear systems (6.1) and (6.2).

Lemma 15. *([50]) Under Assumption 14, there exists a neighborhood $D_1 \subset U$ of the exact solution x^* such that*

(i) The subproblems in (6.3) are all uniquely solvable. There is a unique continuous function $T_i(x)$ such that (6.3) hold for any $x \in D_1$ with $T_i(x^*) = 0$, $i = 1, 2, \ldots, N$.

(ii) The nonlinear system (6.1) and (6.2) have the same solution in D_1.

We define

$$DF(x, y) = \int_0^1 J(x + t(y - x))dt, \tag{6.12}$$

and $\tilde{T}_i \in R^n$ is defined as

$$\tilde{T}_0(x) = \mathbf{0} \in R^n, \quad \forall x \in D_1, \tag{6.13}$$

and

$$\tilde{T}_1(x) = \begin{bmatrix} T_1(x) \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \quad \tilde{T}_i(x) = \begin{bmatrix} T_1(x) \\ \vdots \\ T_i(x) \\ \mathbf{0}_{N-i} \end{bmatrix}, \quad \tilde{T}_N(x) = \begin{bmatrix} T_1(x) \\ T_2(x) \\ \vdots \\ T_N(x) \end{bmatrix}, \tag{6.14}$$

for any $x \in D_1$, where $\mathbf{0}_{N-i} \in R^{N-i}$ is a zero vector, $i = 2, \ldots, N-1$. From (i) in Lemma 15, it is seen that

$$\tilde{T}_i(x) = \sum_{k=1}^{i} R_k^T T_k(x) \tag{6.15}$$

is continuous and $\tilde{T}_i(x) \to 0$ as $x \to x^*$, $i = 1, 2, \ldots, N$.

Following [103], the multiplicative Schwarz preconditioned function $\mathcal{F}(x)$ in (6.2) is also written as

$$\mathcal{F}(x) = \mathcal{T}_1(x) + \sum_{i=2}^{N} \mathcal{T}_i(x - \tilde{T}_{i-1}(x)), \tag{6.16}$$

where

$$\mathcal{T}_i(x - \tilde{T}_{i-1}(x)) = R_i^T T_i(x). \tag{6.17}$$

Due to continuity on T_i in a neighborhood D_1 from Lemma 15, it follows from [8, 103] that we have

$$\mathcal{T}_i(x') - \mathcal{T}_i(x'') = D\mathcal{T}_i(x', x'')(x' - x''), \quad \forall\ x', x'' \in D_1, \tag{6.18}$$

where $D\mathcal{T}_i(x', x'')$ is defined as

$$\mathcal{T}_i(x', x'') \;=\; R_i^T [R_i DF R_i^T]^{-1} R_i DF, \quad DF = DF(x' - \mathcal{T}_i(x'), x'' - \mathcal{T}_i(x'')). \tag{6.19}$$

Theorem 1.7 and Remark 1.8 in [103] show that

$$\begin{aligned} & \mathcal{F}(x') - \mathcal{F}(x'') \\ = & \left(I - \prod_{i=1}^{N} \left[I - D\mathcal{T}_i(x' - \tilde{T}_{i-1}(x'), x'' - \tilde{T}_{i-1}(x'')) \right] \right) (x' - x''), \end{aligned} \tag{6.20}$$

for any $x', x'' \in D_1$.

Remark 16. *We make the right-to-left operator convention:*

$$\prod_{i=1}^{N} A_i = A_N \cdots A_2 A_1. \tag{6.21}$$

Note that $z = x - \tilde{T}_{i-1}(x)$, and then using (6.15) and (6.17) we have

$$\begin{aligned} z - \mathcal{T}_i(z) &= x - \tilde{T}_{i-1}(x) - \mathcal{T}_i(x - \tilde{T}_{i-1}(x)) \\ &= x - \tilde{T}_{i-1}(x) - R_i^T T_i(x) \\ &= x - \tilde{T}_i(x), \end{aligned} \tag{6.22}$$

and it follows from (6.19) and (6.20) that

$$\begin{aligned} & \mathcal{F}(x') - \mathcal{F}(x'') \\ = & \left(I - \prod_{i=1}^{N} (I - Q_i(x', x'')) \right) (x' - x''), \quad \text{for } x', x'' \in D_1, \end{aligned} \tag{6.23}$$

where $Q_i(x', x'')$ is defined as

$$R_i^T [R_i DF(x' - \tilde{T}_i(x'), x'' - \tilde{T}_i(x'')) R_i^T]^{-1} R_i DF(x' - \tilde{T}_i(x'), x'' - \tilde{T}_i(x'')). \tag{6.24}$$

Theorem 17. *Let Assumption 14 hold. Then there is a neighborhood $D \subset U$ of the exact solution x^* such that $\mathcal{F}(x)$ defined in (6.2) is continuously differentiable.*

Moreover,

$$\mathcal{F}'(x) = I - \prod_{i=1}^{N}(I - W_i(x)), \quad x \in D, \tag{6.25}$$

where $W_i(x) = R_i^T[R_i J(x - \tilde{T}_i(x))R_i^T]^{-1} R_i J(x - \tilde{T}_i(x))$.

Proof. Under Assumption 14, Lemma 15 shows that there exists a neighborhood $D_1 \subset U$ of x^* such that $\mathcal{F}(x)$ is continuous, and then (6.23) holds for any $x', x'' \in D_1$. Since $R_i J(x^*) R_i^T$ is invertible and $J(x)$ is continuous, by Lemma 2.3.3 in [54], there exists a neighborhood $D_2 \subset U$ of x^* such that $R_i J(x) R_i^T$ is invertible and $[R_i J(x) R_i^T]^{-1}$ is continuous in D_2, $i = 1, 2, \ldots, N$.

Let $D \subset D_1 \bigcap D_2 \subset U$ be a neighborhood of the exact solution x^* such that $x - \tilde{T}_i(x) \in D_2$ for any $x \in D$ and $i = 1, 2, \ldots, N$, and then we define

$$A(x) = I - \prod_{i=1}^{N}(I - W_i(x)), \quad x \in D, \tag{6.26}$$

where $W_i(x) = R_i^T[R_i J(x - \tilde{T}_i(x))R_i^T]^{-1} R_i J(x - \tilde{T}_i(x))$, $i = 1, 2, \ldots, N$. Note that $A(x)$ is continuous in D. Due to the continuity of $J(x)$ and $\tilde{T}_i(x)$, we have

$$DF(x + h - \tilde{T}_i(x + h), x - \tilde{T}_i(x)) \to J(x - \tilde{T}_i(x)), \quad \text{as } h \to 0. \tag{6.27}$$

Using (6.23) and (6.26), we have

$$\lim_{\|h\| \to 0} \frac{\|\mathcal{F}(x + h) - \mathcal{F}(x) - A(x)h\|}{\|h\|} = 0, \tag{6.28}$$

which implies that $\mathcal{F}$ is Fréchet-differentiable and $\mathcal{F}'(x) = A(x)$. □

In [103], the expression in (6.25) of $\mathcal{F}'(x)$ was employed to develop the damped Richardson scheme instead of the INB method as the solver of the preconditioned system (6.2). The following theorem implies that the formula (6.25) can be simplified to the form of multiplication of two matrices, and then the matrix-vector multiplica-

tion can be carried out instead of explicitly forming the full Jacobian matrix $\mathcal{F}'(x)$ when we solve the Newton equation.

Theorem 18. *Let Assumption 14 hold, and D be the neighborhood determined in Theorem 17. Let $x = [u_1, u_2, \ldots, u_N]^T$, where $u_i \in R^{n_i}$ and $\delta_i = u_i - T_i(x)$, $i = 1, 2, \ldots, N$. Then $\mathcal{F}'(x)$ from Theorem 17 has the form*

$$\mathcal{J}(x) = \begin{bmatrix} \frac{\partial \hat{F}_1}{\partial \delta_1} & & & \\ \frac{\partial \hat{F}_2}{\partial \delta_1} & \frac{\partial \hat{F}_2}{\partial \delta_2} & & \\ \vdots & \vdots & \ddots & \\ \frac{\partial \hat{F}_N}{\partial \delta_1} & \frac{\partial \hat{F}_N}{\partial \delta_2} & \cdots & \frac{\partial \hat{F}_N}{\partial \delta_N} \end{bmatrix}^{-1} \begin{bmatrix} \frac{\partial \hat{F}_1}{\partial \delta_1} & \frac{\partial \hat{F}_1}{\partial u_2} & \frac{\partial \hat{F}_1}{\partial u_3} & \cdots & \frac{\partial \hat{F}_1}{\partial u_N} \\ \frac{\partial \hat{F}_2}{\partial \delta_1} & \frac{\partial \hat{F}_2}{\partial \delta_2} & \frac{\partial \hat{F}_2}{\partial u_3} & \cdots & \frac{\partial \hat{F}_2}{\partial u_N} \\ \vdots & \vdots & \vdots & & \vdots \\ \frac{\partial \hat{F}_N}{\partial \delta_1} & \frac{\partial \hat{F}_N}{\partial \delta_2} & \frac{\partial \hat{F}_N}{\partial \delta_3} & \cdots & \frac{\partial \hat{F}_N}{\partial \delta_N} \end{bmatrix}. \tag{6.29}$$

Proof. Let $\mathbf{0}_{u_i} \in R^{n_i \times n_i}$ be the zero matrix that has the same dimension as the u_i block, $i = 1, 2, \ldots, N$. We define

$$\mathcal{L}(x) \equiv \begin{bmatrix} l_1 \\ l_2 \\ \vdots \\ l_N \end{bmatrix} = \begin{bmatrix} \frac{\partial \hat{F}_1}{\partial \delta_1} & & & \\ \frac{\partial \hat{F}_2}{\partial \delta_1} & \frac{\partial \hat{F}_2}{\partial \delta_2} & & \\ \vdots & \vdots & \ddots & \\ \frac{\partial \hat{F}_N}{\partial \delta_1} & \frac{\partial \hat{F}_N}{\partial \delta_2} & \cdots & \frac{\partial \hat{F}_N}{\partial \delta_N} \end{bmatrix} \tag{6.30}$$

and

$$\mathcal{H}(x) \equiv \begin{bmatrix} h_1 \\ h_2 \\ \vdots \\ h_N \end{bmatrix} = \begin{bmatrix} \mathbf{0}_{u_1} & \frac{\partial \hat{F}_1}{\partial u_2} & \frac{\partial \hat{F}_1}{\partial u_3} & \cdots & \frac{\partial \hat{F}_1}{\partial u_N} \\ & \mathbf{0}_{u_2} & \frac{\partial \hat{F}_2}{\partial u_2} & \cdots & \frac{\partial \hat{F}_2}{\partial u_N} \\ & & \ddots & \ddots & \vdots \\ & & & \mathbf{0}_{u_{N-1}} & \frac{\partial \hat{F}_{N-1}}{\partial u_N} \\ & & & & \mathbf{0}_{u_N} \end{bmatrix}, \tag{6.31}$$

where $l_i \in R^{n_i \times n}$, $h_i \in R^{n_i \times n}$, $i = 1, 2, \ldots, N$, and then we have

$$\begin{aligned} \mathcal{L}(x) &= \sum_{i=1}^{N} R_i^T l_i, \\ \mathcal{H}(x) &= \sum_{i=1}^{N} R_i^T h_i. \end{aligned}$$

We define $I_{u_k} \in R^{n_k \times n_k}$ as the identity matrix that has the same dimension as the u_k block, $1 \leqslant k \leqslant N$, $k \neq i$. Note that

$$I - W_i(x) = \begin{bmatrix} I_{u_1} & & & & & & \\ & \ddots & & & & & \\ & & I_{u_{i-1}} & & & & \\ w_{i1} & \cdots & w_{i,i-1} & \mathbf{0}_{u_i} & w_{i,i+1} & \cdots & w_{i,_N} \\ & & & & I_{u_{i+1}} & & \\ & & & & & \ddots & \\ & & & & & & I_{u_N} \end{bmatrix}, \tag{6.32}$$

where

$$w_{i,k} = \begin{cases} -\left(\frac{\partial \hat{F}_i}{\partial \delta_i}\right)^{-1} \frac{\partial \hat{F}_i}{\partial \delta_k}, & k < i, \\ -\left(\frac{\partial \hat{F}_i}{\partial \delta_i}\right)^{-1} \frac{\partial \hat{F}_i}{\partial u_k}, & k > i. \end{cases} \tag{6.33}$$

It is then verified that

$$\begin{aligned} l_k(I - W_i(x)) &= l_k, \quad k < i, \\ h_k(I - W_i(x)) &= h_k, \quad k > i, \\ l_k(I - W_i(x)) &= -h_k, \quad k = i, \end{aligned}$$

from which we may deduce that

$$\begin{aligned}
\mathcal{L}(x)\prod_{i=1}^{N}(I-W_i(x)) &= \left(\sum_{i=1}^{N}R_i^T l_i\right)\prod_{i=1}^{N}(I-W_i(x)) \\
&= \left(\sum_{i=1}^{N-1}R_i^T l_i - R_N^T h_N\right)\prod_{i=1}^{N-1}(I-W_i(x)) \\
&= \left(\sum_{i=1}^{N-2}R_i^T l_i - \sum_{i=N-1}^{N}R_i^T h_i\right)\prod_{i=1}^{N-2}(I-W_i(x)) \\
&= \left(R_1^T l_1 - \sum_{i=2}^{N}R_i^T h_i\right)(I-W_1(x)) \\
&= -\sum_{i=1}^{N}R_i^T h_i = -\mathcal{H}(x),
\end{aligned}$$

and it follows that

$$\mathcal{L}(x)\left(I-\prod_{i=1}^{N}(I-W_i(x))\right) = \mathcal{L}(x)+\mathcal{H}(x),$$

which completes the proof. □

Theorem 18 obtains in a different way the same formula of the Jacobian $\mathcal{F}'(x)$ shown in Section 5.4 or [50]. In the practical implementation of MSPIN, it is more convenient to use the following approximate Jacobian

$$\mathcal{J}(x) \approx \hat{\mathcal{J}}(x) = L(\bar{x})^{-1}J(\bar{x}), \quad \bar{x} = [\delta_1, \delta_2, \ldots, \delta_{N-1}, u_N]^T, \tag{6.34}$$

where $\delta_i = u_i - T_i$, $i = 1, 2, \ldots, N-1$, and $L(x)$ is defined as

$$L(x) = \begin{bmatrix}
\frac{\partial \hat{F}_1}{\partial u_1} & & & & \\
\frac{\partial \hat{F}_2}{\partial u_1} & \frac{\partial \hat{F}_2}{\partial u_2} & & & \\
\vdots & \vdots & \ddots & & \\
\frac{\partial \hat{F}_N}{\partial u_1} & \frac{\partial \hat{F}_N}{\partial u_2} & \frac{\partial \hat{F}_N}{\partial u_3} & \cdots & \frac{\partial \hat{F}_N}{\partial u_N}
\end{bmatrix}, \tag{6.35}$$

which is the lower triangular part of $J(x)$. Since $F'(x) = J(x)$ is continuous and $R_i J(x^*) R_i^T$ are invertible, we know that $L(x)$ is continuous and $L(x^*)$ is nonsingular. Lemma 2.3.3 in [54] shows that there exists a neighborhood $U' \subset U$ of the exact solution x^* such that $L(x)$ are invertible, and $L(x)^{-1}$ are continuous in the neighborhood U'. Due to the continuity of $T_i(x)$ (see (i) in Lemma 15), we know that $\delta_i = u_i - T_i(x) \to u_i^*$ as $x \to x^*$, where $x^* = [u_1^*, u_2^*, \ldots, u_N^*]^T$ and $i = 1, 2, \ldots, N$. Therefore, it is seen that

Remark 19. *There is a neighborhood $D' \subset D$ of the exact solution x^* such that $L(x)^{-1}$, $\mathcal{J}(x)$, $\hat{\mathcal{J}}(x)$, $\mathcal{J}(x)^{-1}$ and $\hat{\mathcal{J}}(x)^{-1}$ are continuous in D', and*

$$\lim_{x \to x^*} \mathcal{J}(x) = \lim_{x \to x^*} \hat{\mathcal{J}}(x) = \mathcal{J}(x^*). \tag{6.36}$$

6.1.2 The MSPIN Framework

For discussing the convergence properties, we describe a complete MSPIN framework in Algorithm 8. Considering N different physically motivated subsets of the overall, we split $F(x)$ into $\hat{F}_1(x), \hat{F}_2(x), \ldots, \hat{F}_N(x)$, and the unknowns have partition structure corresponding to $x = [u_1, u_2, \ldots, u_N]^T$. In the implementation of MSPIN in Algorithm 8, the most interesting part is how to determine the partition of the physical variables, i.e., the set of equations that belongs to $\hat{F}_i$ for each i, because the best choice is generally problem-specific. The basic objective is to segregate subsystems that retard global Newton convergence from the rest, and to make these systems as small as possible. However, the ideal partitions could evolve during the course of the problem and no theory is currently in hand for this. Once the partitioning and the initial guess are given, together with the customization of the parameters, we move on to step 1.

In step 1, based on the current approximate solution $x^{(k)} = [u_1^{(k)}, \ldots, u_N^{(k)}]^T$, the solutions $T_i^{(k)}$ of the submodels $\hat{F}_i$, $i = 1, 2, \ldots, N$, forming the global residual of the

Algorithm 8 MSPIN for problems with N components

Specify he initial guess $x^{(0)} = [u_1^{(0)}, u_2^{(0)}, \ldots, u_N^{(0)}]^T$.
Set the parameters $\alpha \in (0, \frac{1}{2})$, $\eta_{\max} \in (0, 1)$ and $0 < \theta_{\min} < \theta_{\max} < 1$.
for $k = 0, 1, 2, \ldots$, until convergence, **do**

1. Compute the nonlinear residual $\mathcal{F}(x^{(k)})$ by solving the submodels.

   (a) Starting from the initial guess $T_{i,0}^{(k)} = 0$, $i = 1, 2, \ldots, N$, find $T_1^{(k)}$, $T_2^{(k)}, \ldots, T_N^{(k)}$ by solving the following subproblems sequentially:

$$\begin{aligned}
&\hat{F}_1(u_1^{(k)} - T_1^{(k)}, u_2^{(k)}, u_3^{(k)}, \ldots, u_N^{(k)}) = 0,\\
&\hat{F}_2(u_1^{(k)} - T_1^{(k)}, u_2^{(k)} - T_2^{(k)}, u_3^{(k)}, \ldots, u_N^{(k)}) = 0,\\
&\qquad\qquad \vdots\\
&\hat{F}_N(u_1^{k} - T_1^{(k)}, u_2^{(k)} - T_2^{(k)}, u_3^{(k)} - T_3^{(k)}, \ldots, u_N^{(k)} - T_N^{(k)}) = 0.
\end{aligned}$$

   (b) Form the global residual

$$\mathcal{F}(x^{(k)}) = \begin{bmatrix} T_1^{(k)} \\ T_2^{(k)} \\ \vdots \\ T_N^{(k)} \end{bmatrix}. \tag{6.37}$$

   (c) Check the stopping conditions on $\mathcal{F}(x^{(k)})$.

2. Choose η_k and find the inexact Newton direction $d^{(k)}$ by approximately solving

$$\hat{\mathcal{J}}(x^{(k)})d^{(k)} = \mathcal{F}(x^{(k)}), \tag{6.38}$$

   such that

$$\|\mathcal{F}(x^{(k)}) - \hat{\mathcal{J}}(x^{(k)})d^{(k)}\| \le \eta_k \|\mathcal{F}(x^{(k)})\|, \tag{6.39}$$

   where $\hat{\mathcal{J}}$ has the form (6.34), and the forcing term $\eta_k \in [0, \eta_{\max}]$.

3. Determine the step length $\lambda^{(k)}$ along the direction $d^{(k)}$.

   Set $\lambda^{(k)} = 1$
   while $f(x^{(k)} - \lambda^{(k)} d^{(k)}) > f(x^{(k)}) - \alpha\lambda^{(k)}\mathcal{F}(x^{(k)})^T \hat{\mathcal{J}}(x^{(k)})d^{(k)}$ **do**
   choose some $\theta \in [\theta_{\min}, \theta_{\max}]$
   $\lambda^{(k)} = \theta\lambda^{(k)}$
   end while

4. Compute the new approximate solution.

$$x^{(k+1)} = x^{(k)} - \lambda^{(k)} d^{(k)}. \tag{6.40}$$

end for

transformed system are determined and checked for convergence.

In step 2, an approximate Newton step is accepted provided that (6.39) is satisfied. Note that the global Jacobian system (6.38) is solved using GMRES with no additional linear preconditioning. The Jacobian formula (6.34) implies that nonlinear preconditioning automatically provides a block Gauss-Seidel linear preconditioning for the original unpreconditioned equation.

In step 3, we carry out a backtracking line search algorithm to determine the step length $\lambda^{(k)}$ along the inexact Newton direction $d^{(k)}$. The backtracking linesearch technique is based on the following merit function

$$f(x) \equiv \frac{1}{2}\|\mathcal{F}(x)\|_2^2, \tag{6.41}$$

and $\lambda^{(k)}$ is picked such that

$$f(x^{(k)} - \lambda^{(k)} d^{(k)}) \leqslant f(x^{(k)}) - \alpha\lambda^{(k)} \nabla f(x^{(k)})^T d^{(k)}. \tag{6.42}$$

An easy calculation gives

$$\nabla f(x^{(k)}) = \mathcal{J}(x^{(k)})^T \mathcal{F}(x^{(k)}), \tag{6.43}$$

which is replaced by $\hat{\mathcal{J}}(x^{(k)})^T \mathcal{F}(x^{(k)})$ in a practical algorithm.

For details about the implementation of MSPIN, see [50].

6.2 Local Convergence

In this section, we give a theoretical foundation for the MSPIN algorithm, and show that the MSPIN algorithm is locally convergent. Unfortunately and unsurprisingly, it does not seem possible to carry out a global convergence analysis of MSPIN, because

this algorithm and its properties shown in Section 2 are based on the assumptions on $F(x)$ in the neighborhood of the exact solution x^*.

Let $N(x,r) = \{y \mid \|y - x\| < r\}$ represent an open ball with the center x and a radius r, and $\|\cdot\| = \|\cdot\|_2$ for all the matrices and vectors in the remainder of the paper. The main assumptions are as follows:

Assumption 20. *Assumption 14 holds, and we define*

$$K \equiv \max\{\|\mathcal{J}(x^*)\|, \|\mathcal{J}(x^*)^{-1}\|\}. \tag{6.44}$$

(a) In step 1(a) of Algorithm 8, we solve the subproblems exactly for $T_i^{(k)}$, for each k;

(b) In step 3 of Algorithm 8, we assume α is small enough so that

$$\alpha \leqslant \frac{1 - \eta_{\max}}{64K^4(3 + \eta_{\max})}; \tag{6.45}$$

(c) There exists a fixed small number $r > 0$ such that $N(x^*, r) \subset D$, where D is the neighborhood determined in Lemma 15. For any $x \in N(x^*, r)$, we further assume that

(c_1) $L(x)^{-1}$, $\mathcal{J}(x)$, $\hat{\mathcal{J}}(x)$, $\mathcal{J}(x)^{-1}$ and $\hat{\mathcal{J}}(x)^{-1}$ are continuous;

(c_2) $\|\mathcal{J}(x)\| \leqslant 2K$, $\|\mathcal{J}(x)^{-1}\| \leqslant 2K$, where $\mathcal{J}$ has the form (6.29) ;

(c_3) $\|\hat{\mathcal{J}}(x)\| \leqslant 2K$, $\|\hat{\mathcal{J}}(x)^{-1}\| \leqslant 2K$, where $\hat{\mathcal{J}}$ has the form (6.34) ;

(c_4) $\|\mathcal{J}(x) - \hat{\mathcal{J}}(x)\| \leqslant \frac{1-\eta_{\max}}{4K(1+\eta_{\max})}$;

(c_5) $\|\mathcal{F}(x) - \mathcal{F}(x^*) - \mathcal{J}(x^*)(x - x^*)\| \leqslant \frac{1}{2K}\|x - x^*\|$.

In (a), we assume that all of subproblems are solved exactly, since introduction of the secondary iteration makes the discussion more complicated.

In (b), the proofs of Lemma 21 and Lemma 24 require the assumption (6.45) for α, and more details is found in [44].

In (c), we can find a number $r_1 > 0$ from Remark 19 such that (c_1)-(c_4) hold for any $x \in N(x^*, r_1) \subset D$. Theorem 17 shows that $\mathcal{F}(x)$ is continuously differentiable in a neighborhood of x^*, and hence we can find a number $r_2 > 0$ such that the last inequality (c_5) holds in $N(x^*, r_2) \subset D$ by Lemma 3.2.10 in [54]. Let $r = \min\{r_1, r_2\}$, and $N(x^*, r)$ is the open ball required in (c). Section 6.4 gives an example that satisfies assumptions (c_1)-(c_5).

Note that a line search procedure is carried out with initialization $\lambda^{(k)} = 1$ in step 3 of Algorithm 8. Recalling results from Remark 3.1 and Theorem 3.5 in [44], we can obtain the following lemma. It shows that step 3 of Algorithm 8 will terminate the execution of the while-loop in a finite number of steps when the gradient of f satisfies a mild condition. Moreover, there exists a positive lower bound for the resulting $\lambda^{(k)}$.

Lemma 21. *([44]) Assume there exists $\gamma > 0$ such that*

$$\|\nabla f(y) - \nabla f(z)\| \leqslant \gamma \|y - z\|, \quad \forall\ y, z \in N(x^*, r).$$

Let $x \in N(x^*, \frac{r}{2})$ satisfy $\mathcal{F}(x) \neq 0$ and $\|\mathcal{F}(x)\| \leqslant \frac{r}{8K}$, and we find an inexact Newton direction $d \in R^n$ such that

$$\|\mathcal{F}(x) - \hat{\mathcal{J}}(x)d\| \leq \eta \|\mathcal{F}(x)\|, \quad \eta \in [0, \eta_{\max}],$$

then it holds that $\mathcal{F}(x)^T \mathcal{J}(x)d > 0$ and the while-loop in step 3 of Algorithm 8 will terminate in finite iterations and the determined step length λ satisfies

$$\lambda \geqslant \min\{1, \frac{\alpha\theta_{\min}\mathcal{F}(x)^T \mathcal{J}(x)d}{\gamma\|d\|^2}\}. \tag{6.46}$$

Furthermore, we have $\|\mathcal{F}(x - \lambda d)\| < \|\mathcal{F}(x)\|$.

Lemma 22. *If $x \in N(x^*, r)$, then we have*

$$\|x - x^*\| \leqslant 2K\|\mathcal{F}(x)\|. \tag{6.47}$$

Proof. Since

$$\|x - x^*\| \leqslant \|\mathcal{J}(x^*)^{-1}\|\|\mathcal{J}(x^*)(x - x^*)\|, \tag{6.48}$$

by (6.44) and (c_5) in Assumption 20 and $\mathcal{F}(x^*) = 0$, we have

$$\begin{aligned} \|\mathcal{F}(x)\| &\geqslant \|\mathcal{J}(x^*)(x - x^*)\| - \|\mathcal{F}(x) - \mathcal{F}(x^*) - \mathcal{J}(x^*)(x - x^*)\| \\ &\geqslant \frac{1}{\|\mathcal{J}(x^*)^{-1}\|}\|x - x^*\| - \frac{1}{2K}\|x - x^*\| \\ &\geqslant (\frac{1}{K} - \frac{1}{2K})\|x - x^*\| = \frac{1}{2K}\|x - x^*\|, \end{aligned}$$

which implies that (6.47) holds. □

We state the main result in this section.

Theorem 23. *Assume there exists $\gamma > 0$ such that*

$$\|\nabla f(y) - \nabla f(z)\| \leqslant \gamma\|y - z\|, \quad \forall\ y, z \in N(x^*, r).$$

If $x^{(0)} \in N(x^*, \frac{r}{2})$ satisfies $\mathcal{F}(x^{(0)}) \neq 0$, $\|\mathcal{F}(x^{(0)})\| \leqslant \frac{r}{8K}$, then the MSPIN algorithm generates a sequence $\{x^{(k)}\} \subset N(x^*, \frac{r}{2})$ such that $\{\|\mathcal{F}(x^{(k)})\|\}$ is strictly decreasing, and $x^{(k)} \to x^*$.

Proof. We prove that the MSPIN algorithm can generate a sequence $\{x^{(k)}\} \subset N(x^*, \frac{r}{2})$ such that $\{\|\mathcal{F}(x^{(k)})\|\}$ is strictly decreasing by mathmatical induction.

Initial step. In step 2 of Algorithm 8, we find $d^{(0)}$ such that

$$\|\mathcal{F}(x^{(0)}) - \hat{\mathcal{J}}(x^{(0)})d^{(0)}\| \leq \eta_0\|\mathcal{F}(x^{(0)})\|, \quad \eta_0 \in [0, \eta_{\max}], \tag{6.49}$$

and Lemma 21 shows that we could find a suitable step length $\lambda^{(0)}$ and the inequality $\|\mathcal{F}(x^{(1)})\| < \|\mathcal{F}(x^{(0)})\|$ holds, where $x^{(1)} = x^{(0)} - \lambda^{(0)} d^{(0)}$, $\lambda^{(0)} \in (0, 1]$.

From (6.49), we have

$$\begin{aligned} \|\mathcal{F}(x^{(0)}) - \hat{\mathcal{J}}(x^{(0)})(\lambda^{(0)} d^{(0)})\| &= \|\lambda^{(0)} \left(\mathcal{F}(x^{(0)}) - \hat{\mathcal{J}}(x^{(0)}) d^{(0)}\right) + (1 - \lambda^{(0)})\mathcal{F}(x^{(0)})\| \\ &\leqslant \left(\eta_0 \lambda^{(0)} + (1 - \lambda^{(0)})\right) \|\mathcal{F}(x^{(0)})\| \\ &\leqslant \|\mathcal{F}(x^{(0)})\|, \end{aligned}$$

and then it holds from (c_3) in Assumption 20 that

$$\begin{aligned} \|\lambda^{(0)} d^{(0)}\| &= \|\hat{\mathcal{J}}(x)^{-1} \left(\hat{\mathcal{J}}(x)(\lambda^{(0)} d^{(0)}) - \mathcal{F}(x^{(0)}) + \mathcal{F}(x^{(0)})\right) \| \\ &\leqslant \|\hat{\mathcal{J}}(x)^{-1}\| \left(\|\mathcal{F}(x^{(0)}) - \hat{\mathcal{J}}(x^{(0)})(\lambda^{(0)} d^{(0)})\| + \|\mathcal{F}(x^{(0)})\|\right) \\ &\leqslant 4K \|\mathcal{F}(x^{(0)})\| \\ &< 4K \frac{r}{8K} = \frac{r}{2}. \end{aligned}$$

Hence,

$$\begin{aligned} \|x^{(1)} - x^*\| &= \|x^{(0)} - x^* - \lambda^{(0)} d^{(0)}\| \\ &\leqslant \|x^{(0)} - x^*\| + \|\lambda^{(0)} d^{(0)}\| \\ &< \frac{r}{2} + \frac{r}{2} = r, \end{aligned}$$

which implies $x^{(1)} \in N(x^*, r)$. By Lemma 22 and $\|\mathcal{F}(x^{(1)})\| < \|\mathcal{F}(x^{(0)})\|$, we have

$$\|x^{(1)} - x^*\| \leqslant 2K \|\mathcal{F}(x^{(1)})\| < 2K \|\mathcal{F}(x^{(0)})\| = 2K \frac{r}{8K} < \frac{r}{2}. \tag{6.50}$$

Hence, the MSPIN algorithm can generate $x^{(1)}$ starting from $x^{(0)}$, and $x^{(1)} \in N(x^*, \frac{r}{2})$.

Inductive Step. We assume that there is a k such that the MSPIN algorithm

has generated a sequence $\{x^{(1)}, x^{(2)}, \ldots, x^{(k)}\} \subset N(x^*, \frac{r}{2})$, and

$$\|\mathcal{F}(x^{(k)})\| < \cdots < \|\mathcal{F}(x^{(2)})\| < \|\mathcal{F}(x^{(1)})\|. \tag{6.51}$$

For a given $x^{(k)}$, we compute the inexact Newton direction $d^{(k)}$ in step 2 of Algorithm 8, such that

$$\|\mathcal{F}(x^{(k)}) - \hat{\mathcal{J}}(x^{(k)})d^{(k)}\| \leq \eta_k \|\mathcal{F}(x^{(k)})\|, \quad \eta_k \in [0, \eta_{\max}]. \tag{6.52}$$

Lemma 21 guarantees that $x^{(k+1)} = x^{(k)} - \lambda^{(k)} d^{(k)}$ is generated with $\lambda^{(k)} \in (0, 1]$ and $\|\mathcal{F}(x^{(k+1)})\| < \|\mathcal{F}(x^{(k)})\|$. Similarly to the procedure in the initial step above, we can prove that $x^{(k+1)} \in N(x^*, \frac{r}{2})$. This completes the inductive step.

Next we prove that $x^{(k)} \to x^*$. From (6.52), we have

$$\begin{aligned} \mathcal{F}(x^{(k)})^T \hat{\mathcal{J}}(x^{(k)}) d^{(k)} &= \mathcal{F}(x^{(k)})^T \mathcal{F}(x^{(k)}) - \mathcal{F}(x^{(k)})^T \left(\mathcal{F}(x^{(k)}) - \hat{\mathcal{J}}(x^{(k)}) d^{(k)} \right) \\ &\geqslant \|\mathcal{F}(x^{(k)})\|^2 - \|\mathcal{F}(x^{(k)})\| \|\mathcal{F}(x^{(k)}) - \hat{\mathcal{J}}(x^{(k)}) d^{(k)}\| \\ &\geqslant (1 - \eta_k) \|\mathcal{F}(x^{(k)})\|^2 \\ &\geqslant 2(1 - \eta_{\max}) f(x^{(k)}), \end{aligned} \tag{6.53}$$

and Lemma 21 shows that the step 3 of Algorithm 8 determines

$$\lambda^{(k)} \geqslant \min\{1, \frac{\alpha \theta_{\min} \mathcal{F}(x^{(k)})^T \mathcal{J}(x^{(k)}) d^{(k)}}{\gamma \|d^{(k)}\|^2}\} \tag{6.54}$$

such that

$$f(x^{(k+1)}) = f(x^{(k)} - \lambda^{(k)} d^{(k)}) \leqslant f(x^{(k)}) - \alpha \lambda^{(k)} \mathcal{F}(x^{(k)})^T \hat{\mathcal{J}}(x^{(k)}) d^{(k)}. \tag{6.55}$$

By (6.53) and (6.55), it follows that

$$f(x^{(k+1)}) \leqslant (1 - 2\alpha\lambda^{(k)}(1 - \eta_{\max}))f(x^{(k)}). \tag{6.56}$$

Note that $\{f(x^{(k)})\}$ is strictly monotonic decreasing and nonnegative, and hence $\lim_{k\to\infty} f(x^{(k)})$ exists and it is nonnegative. We can employ a proof by contradiction to show that $\lim_{k\to\infty} f(x^{(k)}) = 0$. Assume that

$$\lim_{k\to\infty} f(x^{(k)}) = \sigma > 0. \tag{6.57}$$

Using (6.52) and $\|\mathcal{F}(x^{(0)})\| \leqslant \frac{r}{8K}$, we have

$$\begin{aligned}
\|d^{(k)}\| &= \|\hat{\mathcal{J}}(x)^{-1}\left(\hat{\mathcal{J}}(x)d^{(k)} - \mathcal{F}(x^{(k)}) + \mathcal{F}(x^{(k)})\right)\| \\
&\leqslant \|\hat{\mathcal{J}}(x)^{-1}\|\left(\|\mathcal{F}(x^{(k)}) - \hat{\mathcal{J}}(x^{(k)})d^{(k)}\| + \|\mathcal{F}(x^{(k)})\|\right) \\
&\leqslant 2K(1+\eta_k)\|\mathcal{F}(x^{(k)})\| \\
&\leqslant 2K(1+\eta_{\max})\|\mathcal{F}(x^{(0)})\| \\
&< 2K(1+\eta_{\max})\frac{r}{8K} = \frac{r(1+\eta_{\max})}{4},
\end{aligned} \tag{6.58}$$

and then it follows from (6.53) that

$$\frac{\mathcal{F}(x^{(k)})^T\mathcal{J}(x^{(k)})d^{(k)}}{\|d^{(k)}\|^2} > \frac{2(1-\eta_{\max})f(x^{(k)})}{\left(\frac{r(1+\eta_{\max})}{4}\right)^2} \geqslant \frac{32(1-\eta_{\max})\sigma}{r^2(1+\eta_{\max})^2} > 0. \tag{6.59}$$

Let $\epsilon = \frac{1}{2}\min\{1, \frac{32\alpha\theta_{\min}(1-\eta_{\max})\sigma}{\gamma r^2(1+\eta_{\max})^2}\} \leqslant \frac{1}{2}$. Therefore, by (6.54) and (6.59), we have

$$\lambda^{(k)} > \min\{1, \frac{32\alpha\theta_{\min}(1-\eta_{\max})\sigma}{\gamma r^2(1+\eta_{\max})^2}\} > \epsilon > 0, \tag{6.60}$$

which implies that

$$\lim_{k\to\infty} \frac{f(x^{(k+1)})}{f(x^{(k)})} \leqslant 1 - 2\alpha\epsilon(1-\eta_{\max}) < 1 \tag{6.61}$$

holds for $\alpha \in (0, \frac{1}{2})$ and $\eta_{\max} \in (0,1)$ (given in Algorithm 8) from (6.56). It is then seen from (6.57) that

$$\lim_{k\to\infty} \frac{f(x^{(k+1)})}{f(x^{(k)})} = 1, \tag{6.62}$$

which contradicts (6.61). Thus, we have $\lim_{k\to\infty} f(x^{(k)}) = 0$, and then Lemma 22 implies that

$$\|x^{(k)} - x^*\| \leqslant 2K\|\mathcal{F}(x^{(k)})\| = 2K\sqrt{2f(x^{(k)})} \to 0, \tag{6.63}$$

i.e., $x^{(k)} \to x^*$ as $k \to \infty$. □

6.3 Convergence Rate

We discuss the local convergence of the MSPIN algorithm in the previous section. In this section, we investigate the convergence rate of the MSPIN algorithm under some reasonable assumptions.

From [44], we borrow the following result.

Lemma 24. *Let $f(x)$ be twice continuously differentiable in $N(x^*, r)$ and there exists $\beta > 0$ such that*

$$\|\nabla f(y) - \nabla f(z)\| \leqslant \beta\|y - z\|, \quad \|\nabla^2 f(y) - \nabla^2 f(z)\| \leqslant \beta\|y - z\|,$$

hold for any $y, z \in N(x^*, r)$. Let $\{x^{(k)}\}$ be the sequence generated by the MSPIN method such that $x^{(k)} \to x^*$. If $\eta_k \to 0$, then $x^{(k+1)} = x^{(k)} - d^{(k)}$ for all sufficiently large k.

Lemma 24 implies that under reasonable conditions on $\nabla f(x)$ and $\nabla^2 f(x)$, the

while-loop in step 3 of Algorithm 8 is not executed and $\lambda^{(k)} = 1$ is acceptable when k is sufficiently large.

Under the conditions of (c) in Assumption 20, the following Lemma shows that $\mathcal{J}(x)$ and $\hat{\mathcal{J}}(x)$ are Lipschitz continuous in a neighborhood of x^*.

Lemma 25. *Let $F(x)$ be twice continuously differentiable in $N(x^*, r)$, and $\mathcal{J}(x)$ and $\hat{\mathcal{J}}(x)$ are defined in (6.29) and (6.34), respectively. Then there is a neighbrhood $V \subset N(x^*, \frac{r}{2})$ of the exact solution x^* such that both $\mathcal{J}(x)$ and $\hat{\mathcal{J}}(x)$ are Lipschitz continuous in V.*

Proof. In the neighborhood $N(x^*, \frac{r}{2})$, $F''(x)$, $T_i'(x)$, $L(x)^{-1}$ and $E_i J(x)$ are continuous , and then we define

$$M_1 := \sup_{x\in N(x^*,\frac{r}{2})} \|F''(x)\|, \qquad M_2 := \max_i \sup_{x\in N(x^*,\frac{r}{2})} \|\tilde{T}_i'(x)\|,$$
$$M_3 := \sup_{x\in N(x^*,\frac{r}{2})} \|L(x)^{-1}\|, \qquad M_4 := \max_i \sup_{x\in N(x^*,\frac{r}{2})} \|E_i J(x)\|,$$

where E_i is defined in (6.7), $\tilde{T}_i \in R^n$ is defined in (6.14) and $\tilde{T}_i'(x) = \sum_{k=1}^{i} R_k^T T_k'(x)$.

For any $x, y \in N(x^*, \frac{r}{2})$, by Lemma 3.3.5 and Lemma 3.2.3 in [54], we have

$$\|J(x) - J(y)\| \leqslant \sup_{0\leqslant t\leqslant 1} \|F''(x + t(y - x))\|\|x - y\| \leqslant M_1\|x - y\|, \tag{6.64}$$

$$\|\tilde{T}_i(x) - \tilde{T}_i(y)\| \leqslant \sup_{0\leqslant t\leqslant 1} \|\tilde{T}_i'(x + t(y - x))\|\|x - y\| \leqslant M_2\|x - y\|, \tag{6.65}$$

where $i = 1, 2, \ldots, N$. By (6.64) and since $L(x)$ is a lower triangular matrix, we have

$$\|L(x) - L(y)\| \leqslant \max_i \|E_i(J(x) - J(y))E_i\| \leqslant \|J(x) - J(y)\| \leqslant M_1\|x - y\|. \tag{6.66}$$

Since $T_i(x)$ is continuous and $\tilde{T}_i(x) = \sum_{k=1}^{i} R_k^T T_k(x)$, then $x - \tilde{T}_i(x) \to x^*$ as $x \to x^*$, and hence there exists a neighborhood $V_i \subset N(x^*, \frac{r}{2})$ such that $x - \tilde{T}_i(x) \in N(x^*, \frac{r}{2})$, for any $x \in V_i$, $i = 1, 2 \ldots, N$. We set $V = \bigcap_{i=1}^{N} V_i \subset N(x^*, \frac{r}{2})$. For any $x, y \in V$ and

for each i, by (6.64) and (6.65), it follows that

$$\begin{aligned}\|J(x-\tilde{T}_i(x))-J(y-\tilde{T}_i(y))\| &\leqslant M_1\|(x-y)-(\tilde{T}_i(x)-\tilde{T}_i(y))\| \\ &\leqslant M_1(1+M_2)\|x-y\|. \end{aligned} \tag{6.67}$$

In a similar way, we have

$$\|L(x-\tilde{T}_i(x))-L(y-\tilde{T}_i(y))\| \leqslant M_1(1+M_2)\|x-y\|, \tag{6.68}$$

where $i=1,2,\ldots,N$.

By setting

$$\hat{L}(x)=L(x-\tilde{T}_N(x)), \quad B_i(x)=E_iJ(x-\tilde{T}_i(x)), \quad i=1,2,\ldots,N, \tag{6.69}$$

we recast (6.29) as

$$\mathcal{J}(x)=L(x-\tilde{T}_N(x))^{-1}\sum_{i=1}^{N}E_iJ(x-\tilde{T}_i(x))=\sum_{i=1}^{N}\mathcal{J}_i(x), \tag{6.70}$$

where $\mathcal{J}_i(x)=\hat{L}(x)^{-1}B_i(x)$. Using (6.67) and (6.68), for any $x,y\in V$, we have

$$\begin{aligned}\|\mathcal{J}_i(x)-\mathcal{J}_i(y)\| &= \|\hat{L}(x)^{-1}B_i(x)-\hat{L}(y)^{-1}B_i(y)\| \\ &\leqslant \|\hat{L}(x)^{-1}B_i(x)-\hat{L}(x)^{-1}B_i(y)\| \\ &\quad +\|\hat{L}(x)^{-1}B_i(y)-\hat{L}(y)^{-1}B_i(y)\| \\ &\leqslant \|\hat{L}(x)^{-1}\|\|B_i(x)-B_i(y)\| \\ &\quad +\|\hat{L}(x)^{-1}\|\|\hat{L}(x)-\hat{L}(y)\|\|\hat{L}(y)^{-1}\|\|B_i(y)\| \\ &\leqslant M_3M_1(1+M_2)\|x-y\|+M_3^2M_1(1+M_2)M_4\|x-y\| \\ &= M_1M_3(1+M_2)(1+M_3M_4)\|x-y\|. \end{aligned} \tag{6.71}$$

Thus,

$$\begin{aligned} \|\mathcal{J}(x) - \mathcal{J}(y)\| &= \|\sum_{i=1}^{N} \mathcal{J}_i(x) - \sum_{i=1}^{N} \mathcal{J}_i(y)\| \leqslant \sum_{i=1}^{N} \|\mathcal{J}_i(x) - \mathcal{J}_i(y)\| \\ &\leqslant NM_1M_3(1+M_2)(1+M_3M_4)\|x-y\|, \end{aligned} \tag{6.72}$$

which shows that $\mathcal{J}(x)$ is Lipschitz continuous in V.

We recast (6.34) as

$$\hat{\mathcal{J}}(x) = L(x - \tilde{T}_{N-1}(x))^{-1} J(x - \tilde{T}_{N-1}(x)) = \sum_{i=1}^{N} \hat{\mathcal{J}}_i(x), \tag{6.73}$$

where $\hat{\mathcal{J}}_i(x) = L(x - \tilde{T}_{N-1}(x))^{-1} E_i J(x - \tilde{T}_{N-1}(x))$. Similarly to (6.71), we can prove that

$$\|\hat{\mathcal{J}}(x) - \hat{\mathcal{J}}(y)\| \leqslant NM_1M_3(1+M_2)(1+M_3M_4)\|x-y\|, \tag{6.74}$$

for any $x, y \in V$. Therefore, $\hat{\mathcal{J}}(x)$ is also Lipschitz continuous in V. □

Theorem 26. *Let $F(x)$ and $f(x)$ be twice continuously differentiable in $N(x^*, r)$, and there exists a constant $C > 0$ such that*

$$\|\nabla f(x) - \nabla f(y)\| \leqslant C\|x-y\|, \quad \|\nabla^2 f(x) - \nabla^2 f(y)\| \leqslant C\|x-y\|, \tag{6.75}$$

for any $x, y \in N(x^*, r)$. Consider a sequence $\{x^{(k)}\}$ generated by the MSPIN method such that $x^{(k)} \to x^*$, and then

(i) If $\eta_k \to 0$, $x^{(k)} \to x^*$ superlinearly ;

(ii) If $\eta_k = \mathcal{O}(\|\mathcal{F}(x^{(k)})\|)$, $x^{(k)} \to x^*$ quadratically .

Proof. According to Lemma 24, we have

$$x^{(k+1)} = x^{(k)} - d^{(k)}, \tag{6.76}$$

where $\|\mathcal{F}(x^{(k)}) - \hat{\mathcal{J}}(x^{(k)})d^{(k)}\| \leq \eta_k \|\mathcal{F}(x^{(k)})\|$, if $\eta_k \to 0$ for all the sufficient large k. By Lemma 25, there exists a $\gamma > 0$ and a neighbrhood $V \subset N(x^*, \frac{r}{2})$ such that $\mathcal{J}(x)$ and $\hat{\mathcal{J}}(x)$ are Lipschitz continuous with Lipschitz constant γ in V.

Let k be large enough such that $x^{(k)}$, $x^* + t(x^{(k)} - x^*) \in V$ and (6.76) hold. Then we obtain

$$\begin{aligned} & \|\hat{\mathcal{J}}(x^{(k)}) - \mathcal{J}(x^* + t(x^{(k)} - x^*))\| \\ \leqslant\ & \|\hat{\mathcal{J}}(x^{(k)}) - \hat{\mathcal{J}}(x^*)\| + \|\mathcal{J}(x^*) - \mathcal{J}(x^* + t(x^{(k)} - x^*))\| \\ \leqslant\ & \gamma(1+t)\|x^{(k)} - x^*\|, \quad t \in [0,1]. \end{aligned} \tag{6.77}$$

The mean value theorem shows that

$$\mathcal{F}(x^{(k)}) = \mathcal{F}(x^{(k)}) - \mathcal{F}(x^*) = \left(\int_0^1 \mathcal{J}(x^* + t(x^{(k)} - x^*))dt \right) (x^{(k)} - x^*). \tag{6.78}$$

Note that $\|\hat{\mathcal{J}}(x^{(k)})^{-1}\| \leqslant 2K$ because $x^{(k)} \in N(x^*, r)$. Using (6.77) and (6.78),

$$\begin{aligned} & \|x^{(k)} - x^* - \hat{\mathcal{J}}(x^{(k)})^{-1}\mathcal{F}(x^{(k)})\| \\ \leqslant\ & \|\hat{\mathcal{J}}(x^{(k)})^{-1}\| \left(\int_0^1 \|\hat{\mathcal{J}}(x^{(k)}) - \mathcal{J}(x^* + t(x^{(k)} - x^*))\|dt \right) \|(x^{(k)} - x^*)\| \\ \leqslant\ & 3\gamma K \|(x^{(k)} - x^*)\|^2. \end{aligned} \tag{6.79}$$

From (6.78), it holds from (c_2) in Assumption 20 that

$$\|\mathcal{F}(x^{(k)})\| \leqslant \left(\int_0^1 \|\mathcal{J}(x^* + t(x^{(k)} - x^*)\|dt \right) \|x^{(k)} - x^*\| \leqslant 2K\|x^{(k)} - x^*\|, \tag{6.80}$$

and it follows that

$$\begin{aligned} \|\hat{\mathcal{J}}(x^{(k)})^{-1}\mathcal{F}(x^{(k)}) - d^{(k)}\| &\leqslant \|\hat{\mathcal{J}}(x^{(k)})^{-1}\|\|\mathcal{F}(x^{(k)}) - \hat{\mathcal{J}}(x^{(k)})d^{(k)}\| \\ &\leqslant \eta_k\|\hat{\mathcal{J}}(x^{(k)})^{-1}\|\|\mathcal{F}(x^{(k)})\| \\ &\leqslant 4\eta_k K^2\|x^{(k)} - x^*\|. \end{aligned} \tag{6.81}$$

Finally, we note that

$$\begin{aligned} x^{(k+1)} - x^* &= (x^{(k)} - x^*) + (x^{(k+1)} - x^{(k)}) \\ &= (x^{(k)} - x^*) - d^{(k)} \\ &= \left(x^{(k)} - x^* - \hat{\mathcal{J}}(x^{(k)})^{-1}\mathcal{F}(x^{(k)})\right) + \left(\hat{\mathcal{J}}(x^{(k)})^{-1}\mathcal{F}(x^{(k)}) - d^{(k)}\right), \end{aligned} \tag{6.82}$$

and then we use (6.79) and (6.81) to conclude

$$\|x^{(k+1)} - x^*\| \leqslant 3\gamma K\|x^{(k)} - x^*\|^2 + 4\eta_k K^2\|x^{(k)} - x^*\|, \tag{6.83}$$

which completes the proof. □

Theorem 26 implies that the choice of the forcing term η_k influences the convergence rate of the MSPIN algorithm. The Newton iteration can achieve superlinear or even quadratic rates of convergence by appropriately choosing η_k.

6.4 An Illustration

In this section, a simple example illustrates that the conditions (c_1)-(c_5) in Assumption 20 are not unreasonably strict. The following lemma is required in the later discussion.

Lemma 27.

$$1 + x < e^x < 1 + 3x, \quad x \in (0, 1), \tag{6.84}$$

$$e^x \leqslant 1 + \frac{1}{2}x, \quad x \in (-1, 0], \tag{6.85}$$

$$|e^x - 1| \leqslant e^{|x|} - 1. \tag{6.86}$$

Proof. The first two inequalities (6.84) and (6.85) is easy, and we prove the last one here. The inequality (6.86) is trivially satisfied when $x \geqslant 0$. Let $g(x) = e^x + e^{-x}$, for $x \leqslant 0$, and we have $e^x \leqslant e^{-x}$ and $g'(x) < 0$ $(x < 0)$, which implies that $g(x) \geqslant g(0) = 2$. Therefore, it holds that

$$-(e^{|x|} - 1) = 1 - e^{-x} \leqslant e^x - 1 < e^{-x} - 1 = e^{|x|} - 1, \quad x < 0.$$

In summary, for any real number x, (6.86) always holds. □

Consider the system $F(x)$ of two nonlinear equations in two unknowns,

$$\begin{aligned} F_1(x_1, x_2) &= (x_1 - x_2^3 - 1)^3 - x_2^3, \\ F_2(x_1, x_2) &= x_1 + 2\mu e^{x_2} - 1 - 2\mu, \end{aligned}$$

where $\mu \in (0, 1)$. It is easy to verify that $x^* = [1, 0]^T$ is a solution of this system. The nonlinearly preconditioned system $\mathcal{F}(x) = [T_1(x), T_2(x)]^T$ is obtained by solving

$$\begin{aligned} F_1(x_1 - T_1(x), x_2) &= (x_1 - T_1(x) - x_2^3 - 1)^3 - x_2^3 = 0, \\ F_2(x_1 - T_1(x), x_2 - T_2) &= x_1 - T_1(x) + 2\mu e^{x_2 - T_2(x)} - 1 - 2\mu = 0, \end{aligned}$$

from which we get

$$x_1 - T_1(x) - x_2^3 - 1 = x_2, \tag{6.87}$$

and it follows that

$$F_2(x_1 - T_1(x), x_2 - T_2(x)) = x_2^3 + x_2 - 2\mu + 2\mu e^{x_2 - T_2(x)} = 0. \tag{6.88}$$

Thus, we obtain

$$\mathcal{F}'(x) = \mathcal{J}(x) = \begin{bmatrix} 1 & -3x_2^2 - 1 \\ 0 & 1 + \frac{1}{2\mu}(3x_2^2 + 1)e^{-(x_2 - T_2(x))} \end{bmatrix}, \tag{6.89}$$

and

$$\hat{\mathcal{J}}(x) = \begin{bmatrix} 1 & -3x_2^2 - 1 \\ 0 & 1 + \frac{1}{2\mu}(3x_2^2 + 1)e^{-x_2} \end{bmatrix}. \tag{6.90}$$

Moreover,

$$\mathcal{J}(x^*) = \hat{\mathcal{J}}(x^*) = \begin{bmatrix} 1 & -1 \\ 0 & 1 + \frac{1}{2\mu} \end{bmatrix}, \tag{6.91}$$

and

$$K = \max\{\|\mathcal{J}(x^*)\|, \|\mathcal{J}(x^*)^{-1}\|\} = 1 + \frac{1}{2\mu}.$$

Besides, it is easily verified the continuity of $L(x)^{-1}$, $\mathcal{J}(x)$, $\hat{\mathcal{J}}(x)$, $\mathcal{J}(x)^{-1}$ and $\hat{\mathcal{J}}(x)^{-1}$.

Let $r \in (0, \frac{\mu}{3})$ and we assume that $x = [x_1, x_2]^T \in N(x^*, r)$, i.e.,

$$(x_1 - 1)^2 + x_2^2 \leqslant r^2, \tag{6.92}$$

which implies that $|x_2| \leqslant r$. Using (6.84) and (6.88), it holds that

$$\begin{aligned}
&e^{-x_2} \leqslant e^{|x_2|} \leqslant e^r < e^{\frac{\mu}{3}} < 1 + \mu < 2, \\
&3x_2^2 + 1 \leqslant 3r^2 + 1 < 3(\frac{\mu}{3})^2 + 1 < \frac{1}{3}\mu^2 + 1, \\
&\frac{1}{2\mu}e^{-(x_2 - T_2(x))} = \frac{1}{2\mu - x_2^3 - x_2} < \frac{1}{2\mu - r^3 - r} < \frac{1}{\mu + (\mu - 2r)} < \frac{1}{\mu},
\end{aligned}$$

and it follows that

$$\begin{aligned}\|\hat{\mathcal{J}}(x)\| &= 1+\frac{1}{2\mu}(3x_2^2+1)e^{-x_2} \leqslant 1+\frac{1}{\mu}(\frac{1}{3}\mu^2+1) < 2+\frac{1}{\mu} = 2K,\\ \|\mathcal{J}(x)\| &= 1+\frac{1}{2\mu}(3x_2^2+1)e^{-(x_2-T_2(x))} \leqslant 1+\frac{1}{\mu}(\frac{1}{3}\mu^2+1) < 2+\frac{1}{\mu} = 2K.\end{aligned}$$

Note that

$$\|\mathcal{J}(x)^{-1}\| = 1 \leqslant 2K, \quad \|\hat{\mathcal{J}}(x)^{-1}\| = 1 \leqslant 2K. \quad \forall\ x \in N(x^*, r). \tag{6.93}$$

Since

$$|\frac{1}{2\mu}(x_2^3+x_2)| \leqslant \frac{1}{2\mu}(r^3+r) \leqslant \frac{r}{\mu} < \frac{1}{3}, \quad r \in (0, \frac{\mu}{3}), \tag{6.94}$$

by (6.88), it holds that

$$e^{-1} < 1-\frac{1}{3} < e^{x_2-T_2(x)} = 1-\frac{1}{2\mu}(x_2^3+x_2) < 1+\frac{1}{3} < e, \tag{6.95}$$

which implies that $|x_2 - T_2(x)| < 1$. From (6.84), (6.85) and (6.95), we have

$$x_2 - T_2(x) < -\tfrac{1}{2\mu}(x_2^3+x_2) \leqslant \tfrac{1}{2\mu}(|x_2|^3+|x_2|), \qquad \text{if } x_2 - T_2(x) > 0, \tag{6.96}$$

$$-\tfrac{1}{\mu}(|x_2|^3+|x_2|) \leqslant -\tfrac{1}{\mu}(x_2^3+x_2) \leqslant x_2 - T_2, \qquad \text{if } x_2 - T_2(x) \leqslant 0. \tag{6.97}$$

Hence,

$$|x_2 - T_2(x)| \leqslant \frac{1}{\mu}(|x_2|^2+1)|x_2| \leqslant \frac{1}{\mu}(r+1)|x_2| < (\frac{1}{3}+\frac{1}{\mu})|x_2|,$$

and then we have

$$|T_2(x)| \leqslant |x_2 - T_2(x)| + |x_2| < (\frac{4}{3}+\frac{1}{\mu})r < \frac{4}{9}\mu+\frac{1}{3} < 1, \quad r \in (0, \frac{\mu}{3}). \tag{6.98}$$

According to (6.84), (6.86) and (6.98), we have

$$\begin{aligned}\|\mathcal{J}(x)-\hat{\mathcal{J}}(x)\| &= \frac{1}{2\mu}(3x_2^2+1)e^{-x_2}|e^{T_2(x)}-1| \\ &\leqslant \frac{1}{2\mu}(\frac{1}{3}\mu^2+1)e^r(e^{|T_2(x)|}-1) \\ &\leqslant \frac{1}{2\mu}(\frac{1}{3}+1)e^r(3|T_2(x)|) \\ &\leqslant \frac{2e^r}{3\mu}[3(\frac{4}{3}+\frac{1}{\mu})r] \leqslant \frac{2er}{\mu}(\frac{4}{3}+\frac{1}{\mu}).\end{aligned}$$

It is seen that the condition (c_4) is satisfied when

$$r \leqslant \frac{\mu(1-\eta_{\max})}{8eK(\frac{4}{3}+\frac{1}{\mu})(1+\eta_{\max})} = \frac{\mu(1-\eta_{\max})}{4e(2+\frac{1}{\mu})(\frac{4}{3}+\frac{1}{\mu})(1+\eta_{\max})}, \tag{6.99}$$

and then it is noted that

$$\frac{\mu(1-\eta_{\max})}{8eK(\frac{4}{3}+\frac{1}{\mu})(1+\eta_{\max})} \leqslant \frac{\mu}{8eK(\frac{4}{3}+\frac{1}{\mu})} = \frac{3\mu^2}{8eK(4\mu+3)} \leqslant \frac{\mu^2}{8eK} < \frac{\mu^2}{3K} < \frac{\mu}{3}. \tag{6.100}$$

From (6.88) and (6.89), we have

$$\mathcal{J}(x) = \begin{bmatrix} 1 & -3x_2^2-1 \\ 0 & 1+\frac{1}{2\mu}\frac{3x_2^2+1}{1-\frac{1}{2\mu}(x_2^3+x_2)} \end{bmatrix}, \tag{6.101}$$

and it follows that

$$\|\mathcal{J}(x^*+t(x-x^*))-\mathcal{J}(x^*)\| = \frac{1}{2\mu}\left|\frac{3(tx_2)^2+1}{1-\frac{1}{2\mu}((tx_2)^3+(tx_2))}-1\right|,\ t\in[0,1]. \tag{6.102}$$

Note that $K=1+\frac{1}{2\mu}>1$, $\mu\in(0,1)$ and $|x_2|<r$. Let $0<r\leqslant\frac{\mu^2}{3K}<\frac{\mu}{3}<1$, we have

$$\left|\frac{1}{2\mu}((tx_2)^3+(tx_2))\right| \leqslant \frac{1}{2\mu}(|x_2|^3+|x_2|) \leqslant \frac{1}{2\mu}(r^3+r) \leqslant \frac{r}{\mu} \leqslant \frac{\mu}{3K} < 1, \tag{6.103}$$

and then it holds that

$$-\frac{\mu}{\mu+3K} = \frac{1}{1+\frac{\mu}{3K}} - 1 \leqslant \frac{3(tx_2)^2+1}{1-\frac{1}{2\mu}((tx_2)^3+(tx_2))} - 1, \tag{6.104}$$

$$\frac{3(tx_2)^2+1}{1-\frac{1}{2\mu}((tx_2)^3+(tx_2))} - 1 \leqslant \frac{3r^2+1}{1-\frac{\mu}{3K}} - 1 \leqslant \frac{\mu r+1}{1-\frac{\mu}{3K}} - 1. \tag{6.105}$$

From (6.102), (6.104), and (6.105), we have

$$\sup_{0\leqslant t\leqslant 1} \|\mathcal{J}(x^*+t(x-x^*)) - \mathcal{J}(x^*)\| \leqslant \max\{\frac{1}{2(\mu+3K)}, \frac{1}{2\mu}(\frac{\mu r+1}{1-\frac{\mu}{3K}} - 1)\}. \tag{6.106}$$

According to the mean value theorem, we have

$$\begin{aligned} & \|\mathcal{F}(x) - \mathcal{F}(x^*) - \mathcal{J}(x^*)(x-x^*)\| \\ = & \left\| \left(\int_0^1 (\mathcal{J}(x^*+t(x-x^*)) - \mathcal{J}(x^*))dt \right)(x-x^*) \right\| \\ \leqslant & \sup_{0\leqslant t\leqslant 1} \|\mathcal{J}(x^*+t(x-x^*)) - \mathcal{J}(x^*)\| \|x-x^*\|. \end{aligned}$$

Therefore, it is seen that the condition (c_5) is satisfied when

$$\frac{1}{2\mu}(\frac{\mu r+1}{1-\frac{\mu}{3K}} - 1) \leqslant \frac{1}{2K}, \tag{6.107}$$

from which we obtain

$$r \leqslant \frac{1}{3K}(2-\frac{\mu}{K}). \tag{6.108}$$

It is easily verified that

$$\frac{\mu^2}{3K} \leqslant \frac{1}{3K}(2-\frac{\mu}{K}), \tag{6.109}$$

because $\mu^2 + \frac{\mu}{K} = \mu^2 + \frac{\mu}{1+\frac{1}{2\mu}} \leqslant 2\mu \leqslant 2$.

Using (6.99), (6.100), (6.108) and (6.109), to sum up, the assumptions (c_1)-(c_5)

hold for any $x \in N(x^*, r)$, where r satisfies the following inequality

$$r \leqslant \frac{\mu(1-\eta_{\max})}{4e(2+\frac{1}{\mu})(\frac{4}{3}+\frac{1}{\mu})(1+\eta_{\max})}. \tag{6.110}$$

Chapter 7

Conclusion and Future Work

We present nonlinearly preconditioned inexact Newton methods with physics-based field-split partitioning and extend the existing theory for existence and uniqueness for monotone nonlinear systems in a straightforward way to the Gauss-Seidel version. MSPIN is typically employed to generate a relatively small number of sequential subproblems involving nonintersecting subsets of the original unknowns, whereas the classical ASPIN in [28] is typically employed to generate a large number of concurrent subproblems whose unknowns overlap as induced by subdomain geometry; however, there is nothing fundamental in the algebra that so restricts the respective flavors. Numerical results illustrate that the nonlinear preconditioners are effective in improving on the performance of the global Newton iterations. In all of our examples, the MSPIN algorithm is more robust than ASPIN.

Field splitting is an option that can be combined with domain-splitting. As illustrated, the former is often useful for coarse granularity where intuition can play a role in problem decomposition. The latter is readily extensible to fine granularity, particularly in weak parallel scaling. A combination of the two may be natural for large-scale simulations with natural partitions between the models. Even though the domain-based ASPIN is robust for some problems [28, 36, 37], local subproblems may still fail to converge due to unbalanced nonlinearities or the lack of a good initial guess [45]. The field-split algorithms provide additional options.

The MSPIN algorithm [50] was introduced as a physics-based field-split method to cope with "nonlinear stiffness," however, until now the convergence of MSPIN has not been discussed and no theory predicts even local convergence rates. In this dissertation, we provide the MSPIN algorithm with a local convergence proof. Based on local assumptions on $F(x)$, it is shown that the preconditioned function $\mathcal{F}(x)$ is continuously differentiable, and we obtain the Jacobian of the transformed system $\mathcal{F}'(x)$ in a more intuitive way than previously. Under some reasonable assumptions, we discuss the local convergence property and the convergence rate. Customarily desired superlinear or even quadratic convergence can be obtained when the forcing terms η_k are picked suitably.

Finally, there remains much future work for the MSPIN algorithm. Picking the suitable components and the corresponding subproblems is not trivial; different groupings and different orderings play an important role in the quality of the nonlinear preconditioning, however, there is not yet theory to show how to determine a good ordering or even good partitions for physical fields. Moreover, the Jacobian of the preconditioned system is expressible only in the form of matvecs and it therefore a challenge to precondition further. The linear systems for the Newton corrections of the nonlinearly Schwarz-preconditioned forms of the rootfinding problem have the same left-hand sides as the linearly Schwarz-preconditioned forms of the problem in its original coordinates. (Only the right-hand sides are different, and this is, of course, essential to the nonlinear convergence improvement offered by ASPIN and MSPIN.) In parabolic contexts, such one-level Schwarz preconditioning of the linear Newton correction problems is known to be sufficient for convergence independent of mesh parameters and decomposition granularity. However, in elliptic contexts, such one-level Schwarz preconditioning does not provide convergence scalability in either strong or weak scaling. Therefore, multilevel forms of additive and multiplicative nonlinear preconditioning are ripe for exploration.

REFERENCES

[1] D. A. Knoll and D. E. Keyes, "Jacobian-free Newton–Krylov methods: a survey of approaches and applications," *J. Comput. Phys.*, vol. 193, no. 2, pp. 357–397, 2004.

[2] J. E. Dennis and R. B. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*. SIAM, Philadelphia, 1996.

[3] P. Brune, M. G. Knepley, B. F. Smith, and X. Tu, "Composing scalable nonlinear algebraic solvers," Argonne National Laboratory, Preprint ANL/MCS-P2010-0112, 2013.

[4] R. Fletcher and C. M. Reeves, "Function minimization by conjugate gradients," *Comput. J.*, vol. 7, no. 2, pp. 149–154, 1964.

[5] C. W. Oosterlee and T. Washio, "Krylov subspace acceleration of nonlinear multigrid with application to recirculating flows," *SIAM J. Sci. Comput.*, vol. 21, no. 5, pp. 1670–1690, 2000.

[6] D. G. Anderson, "Iterative procedures for nonlinear integral equations," *J. Assoc. Comput. Mach.*, vol. 12, no. 4, pp. 547–560, 1965.

[7] A. Toth and C. T. Kelley, "Convergence analysis for Anderson acceleration," *SIAM J. Numer. Anal.*, vol. 53, no. 2, pp. 805–819, 2015.

[8] M. Dryja and W. Hackbusch, "On the nonlinear domain decomposition method," *BIT*, vol. 37, no. 2, pp. 296–311, 1997.

[9] A. Brandt, "Multi-level adaptive solutions to boundary-value problems," *Math. Comp.*, vol. 31, no. 138, pp. 333–390, 1977.

[10] R. H. Byrd, J. Nocedal, and R. B. Schnabel, "Representations of quasi-Newton matrices and their use in limited memory methods," *Math. Programming*, vol. 63, no. 1-3, pp. 129–156, 1994.

[11] C. T. Kelley, *Iterative Methods for Linear and Nonlinear Equations*. SIAM, Philadelphia, 1995.

[12] H.-R. Fang and Y. Saad, "Two classes of multisecant methods for nonlinear acceleration," *Numer. Linear Algebra Appl.*, vol. 16, no. 3, pp. 197–221, 2009.

[13] 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.*, vol. 19, no. 1, pp. 246–265, 1998.

[14] D. A. Knoll, P. R. McHugh, and V. A. Mousseau, "Newton-Krylov-Schwarz methods applied to the tokamak edge plasma fluid equations," in *Domain-Based Parallelism and Problem Decomposition in Computational Science and Engineering*. SIAM, Philadelphia, 1995, pp. 75–96.

[15] X.-C. Cai, W. D. Gropp, D. E. Keyes, and M. D. Tidriri, "Newton-Krylov-Schwarz methods in CFD," in *Proceedings of an International Workshop on Numerical Methods for the Navier-Stokes Equations, F. Hebeker and R. Rannacher, eds., Notes on Numerical Fluid Mechanics*, vol. 47. Vieweg-Verlag, Braunschweig, Germany, 1994, pp. 17–30.

[16] M. Munteanu, L. F. Pavarino, and S. Scacchi, "A scalable Newton-Krylov-Schwarz method for the bidomain reaction-diffusion system," *SIAM J. Sci. Comput.*, vol. 31, no. 5, pp. 3861–3883, 2009.

[17] X.-C. Cai, D. E. Keyes, and L. Marcinkowski, "Nonlinear additive Schwarz preconditioners and application in computational fluid dynamics," *Int. J. Numer. Meth. Fluids*, vol. 40, no. 12, pp. 1463–1470, 2002.

[18] J. Nocedal and S. Wright, *Numerical Optimization*. Second Edition, Springer-Verlag, New York, 2006.

[19] A. R. Conn, N. I. M. Gould, and P. L. Toint, *Trust Region Methods*. SIAM, Philadelphia, 2000, vol. 1.

[20] M. D. Smooke and R. M. Mattheij, "On the solution of nonlinear two-point boundary value problems on successively refined grids," *Appl. Numer. Math.*, vol. 1, no. 6, pp. 463–487, 1985.

[21] 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 method: application to computational fluid dynamics and computational physics," *J. Comput. Phys.*, vol. 92, no. 1, pp. 1–66, 1991.

[22] T. S. Coffey, C. T. Kelley, and D. E. Keyes, "Pseudotransient continuation and differential-algebraic equations," *SIAM J. Sci. Comput.*, vol. 25, no. 2, pp. 553–569, 2003.

[23] C. T. Kelley and D. E. Keyes, "Convergence analysis of pseudo-transient continuation," *SIAM J. Numer. Anal.*, vol. 35, no. 2, pp. 508–523, 1998.

[24] P. J. Lanzkron, D. J. Rose, and J. T. Wilkes, "An analysis of approximate nonlinear elimination," *SIAM J. Sci. Comput.*, vol. 17, no. 2, pp. 538–559, 1996.

[25] X.-C. Cai and X.-F. Li, "Inexact Newton methods with restricted additive Schwarz based nonlinear elimination for problems with high local nonlinearity," *SIAM J. Sci. Comput.*, vol. 33, no. 2, pp. 746–762, 2011.

[26] 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*, vol. 110, pp. 96–107, 2015.

[27] 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," *ETNA*, vol. 37, pp. 239–251, 2010.

[28] X.-C. Cai and D. E. Keyes, "Nonlinearly preconditioned inexact Newton algorithms," *SIAM J. Sci. Comput.*, vol. 24, no. 1, pp. 183–200, 2002.

[29] D. E. Keyes, L. C. McInnes, C. Woodward, W. Gropp, E. Myra, M. Pernice, J. Bell, J. Brown, A. Clo, J. Connors, E. Constantinescu, D. Estep, K. Evans, C. Farhat, A. Hakim, G. Hammond, G. Hansen, J. Hill, T. Isaac, X. M. Jiao, K. Jordan, D. Kaushik, E. Kaxiras, A. Koniges, K. Lee, A. Lott, Q. M. Lu, J. Magerlein, R. Maxwell, M. McCourt, M. Mehl, R. Pawlowski, A. P. Randles, D. Reynolds, B. Rivière, U. Rüde, T. Scheibe, J. Shadid, B. Sheehan, M. Shephard, A. Siegel, B. Smith, X.-Z. Tang, C. Wilson, and B. Wohlmuth, "Multiphysics simulations challenges and opportunities," *Int. J. High. Perform. Comput. Appl.*, vol. 27, no. 1, pp. 4–83, 2013.

[30] P.-L. Lions, "On the Schwarz alternating method. I," in *First international symposium on domain decomposition methods for partial differential equations*, Paris, France. SIAM, Philadelphia, 1988, pp. 1–42.

[31] X.-C. Tai and M. Espedal, "Rate of convergence of some space decomposition methods for linear and nonlinear problems," *SIAM J. Numer. Anal.*, vol. 35, no. 4, pp. 1558–1570, 1998.

[32] S. H. Lui, "On Schwarz alternating methods for the incompressible Navier–Stokes equations," *SIAM J. Sci. Comput.*, vol. 22, no. 6, pp. 1974–1986, 2001.

[33] B. F. Smith, P. Bjorstad, and W. D. Gropp, *Domain decomposition: parallel multilevel methods for elliptic partial differential equations.* Cambridge University Press, 2004.

[34] S. Balay, M. F. Adams, J. Brown, P. Brune, K. Buschelman, K. Buschelman, V. Eijkhout, W. D. Gropp, D. Kaushik, M. G. Knepley, L. C. McInnes, K. Rupp, B. F. Smith, and H. Zhang, "The Portable, Extensible Toolkit for Scientific Computing, code and documentation," 2014, available from http://www.mcs.anl.gov/petsc.

[35] L. Marcinkowski and X.-C. Cai, "Parallel performance of some two-level ASPIN algorithms," in *Lecture Notes in Computational Science and Engineering, ed. R. Kornhuber, R. H. W. Hoppe, D. E. Keyes, J. Periaux, O. Pironneau and J. Xu*, vol. 40. Springer-Verlag, Heidelberg, 2005, pp. 639–646.

[36] X.-C. Cai, D. E. Keyes, and D. P. Young, "A nonlinear additive Schwarz preconditioned inexact Newton method for shocked duct flow," in *Proceedings of the 13th International Conference on Domain Decomposition Methods*, Lyon, France. ddm.org, 2000, pp. 343–350.

[37] F.-N. Hwang and X.-C. Cai, "A parallel nonlinear additive Schwarz preconditioned inexact Newton algorithm for incompressible Navier–Stokes equations," *J. Comput. Phys.*, vol. 204, no. 2, pp. 666–691, 2005.

[38] 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, the 15th International Conference on Domain Decomposition Methods*, Berlin, Germany. ddm.org, 2004, pp. 201–208.

[39] F.-N. Hwang and X.-C. Cai, "A class of parallel two-level nonlinear Schwarz preconditioned inexact Newton algorithms," *Comput. Methods Appl. Mech. Engrg.*, vol. 196, no. 8, pp. 1603–1611, 2007.

[40] J. O. Skogestad, E. Keilegavlen, and J. M. Nordbotten, "Domain decomposition strategies for nonlinear flow problems in porous media," *J. Comput. Phys.*, pp. 439–451, 2013.

[41] F.-N. Hwang and X.-C. Cai, "A combined linear and nonlinear preconditioning technique for incompressible Navier-Stokes equations," ed. J. Dongarra, K. Madsen, and J. Wasniewski. Springer-Verlag, Heidelberg, 2006, pp. 313–322.

[42] F.-N. Hwang, "Some parallel linear and nonlinear Schwarz methods with applications in computational fluid dynamics," Ph.D. dissertation, University of Colorado at Boulder, Boulder, CO, USA, 2004, aAI3153838.

[43] X.-C. Cai, L. Marcinkowski, and P. S. Vassilevski, "An element agglomeration nonlinear additive Schwarz preconditioned Newton method for unstructured finite element problems," *Appl. Math.*, vol. 50, no. 3, pp. 247–275, 2005.

[44] H.-B. An, "On convergence of the additive Schwarz preconditioned inexact Newton method," *SIAM J. Numer. Anal.*, vol. 43, no. 5, pp. 1850–1871, 2005.

[45] S. H. Lui, "Nonlinearly preconditioned Newton's method," in *Proceedings of the 14th International Conference on Domain Decomposition Methods*, Cocoyoc, Mexico. ddm.org, 2002, pp. 95–105.

[46] C. Groß and R. Krause, "On the Globalization of ASPIN employing Trust-Region Control Strategies - Convergence Analysis and Numerical Examples," Instuitute of Computational Science, Universita della Svizzera italiana, Tech. Rep. 2011-03, 2011.

[47] ——, "A new Class of Non–linear Additively Preconditioned Trust–Region Strategies: Convergence Results and Applications to Non-linear Mechanics," Institute for Numerical Simulation, University of Bonn, INS preprint 904, March 2009, submitted to Math. Comput., under review.

[48] ——, "A Generalized Recursive Trust-Region Approach - Nonlinear Multiplicatively Preconditioned Trust-Region Methods and Applications," Institute of Computational Science, Universita della Svizzera italiana, Tech. Rep. 2010-09, March 2010.

[49] L.-L. Liu, D. E. Keyes, and S.-Y. Sun, "Fully implicit two-phase reservoir simulation with the additive Schwarz preconditioned inexact Newton method," in *SPE Reservoir Characterisation and Simulation Conference and Exhibition*. www.onepetro.org, Sep 16 - 18, 2013, DOI: 10.2118/166062-MS.

[50] L. Liu and D. E. Keyes, "Field-split preconditioned inexact Newton algorithms," *SIAM J. Sci. Comput.*, vol. 37, no. 3, pp. A1388–A1409, 2015.

[51] ——, "Convergence analysis for the multiplicative Schwarz preconditioned inexact Newton algorithm," *submitted to SIAM J. Numer. Anal.*

[52] S. C. Eisenstat and H. F. Walker, "Globally convergent inexact Newton methods," *SIAM J. Optim.*, vol. 4, no. 2, pp. 393–422, 1994.

[53] M. Pernice and H. F. Walker, "NITSOL: A Newton iterative solver for nonlinear systems," *SIAM J. Sci. Comput.*, vol. 19, no. 1, pp. 302–318, 1998.

[54] J. M. Ortega and W. C. Rheinboldt, *Iterative Solution of Nonlinear Equations in Several Variables*. SIAM, Philadelphia, 2000.

[55] R. S. Dembo, S. C. Eisenstat, and T. Steihaug, "Inexact Newton methods," *SIAM J. Numer. Anal.*, vol. 19, no. 2, pp. 400–408, 1982.

[56] P. N. Brown and Y. Saad, "Hybrid Krylov methods for nonlinear systems of equations," *SIAM J. Sci. Stat. Comput.*, vol. 11, no. 3, pp. 450–481, 1990.

[57] ——, "Convergence theory of nonlinear Newton-Krylov algorithms," *SIAM J. Optim.*, vol. 4, no. 2, pp. 297–330, 1994.

[58] Y. Saad and M. H. Schultz, "GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems," *SIAM J. Sci. Statist. Comput.*, vol. 7, no. 3, pp. 856–869, 1986.

[59] Y. Saad, *Iterative Methods for Sparse Linear Systems*. SIAM, Philadelphia, 2003.

[60] H.-B. An, Z.-Y. Mo, and X.-P. Liu, "A choice of forcing terms in inexact Newton method," *J. Comput. Appl. Math.*, vol. 200, no. 1, pp. 47–60, 2007.

[61] M. A. Gomes-Ruggiero, V. L. R. Lopes, and J. V. Toledo-Benavides, "A globally convergent inexact Newton method with a new choice for the forcing term," *Annals of Operations Research*, vol. 157, no. 1, pp. 193–205, 2008.

[62] S. C. Eisenstat and H. F. Walker, "Choosing the forcing terms in an inexact Newton method," *SIAM J. Sci. Comput.*, vol. 17, no. 1, pp. 16–32, 1996.

[63] R. P. Pawlowski, J. P. Simonis, H. F. Walker, and J. N. Shadid, "Inexact Newton dogleg methods," *SIAM J. Numer. Anal.*, vol. 46, no. 4, pp. 2112–2132, 2008.

[64] M. J. D. Powell, "A hybrid method for nonlinear equations," in *Numerical methods for nonlinear algebraic equations*, vol. 7. P. Rabinowitz, ed., Gordon and Breach, London, 1970, pp. 87–114.

[65] T. F. Coleman and J. J. Moré, "Estimation of sparse Jacobian matrices and graph coloring problems," *SIAM J. Numer. Anal.*, vol. 20, no. 1, pp. 187–209, 1983.

[66] P. Hovland, B. Norris, and B. Smith, "Making automatic differentiation truly automatic: coupling PETSc with ADIC," *Future Gener. Comput. Sy.*, vol. 21, no. 8, pp. 1426–1438, 2005.

[67] C. Bischof, A. Carle, P. Hovland, P. Khademi, and A. Mauer, "ADIFOR 2.0 user's guide (Revision *d*)," Argonne National Laboratory, Tech. Rep. ANL/MCS-TM-192.

[68] M. Hayek, E. Mouche, and C. Mügler, "Modeling vertical stratification of CO_2 injected into a deep layered aquifer," *Adv. Water Resour.*, vol. 32, no. 3, pp. 450–462, 2009.

[69] P. Jenny, S. H. Lee, and H. A. Tchelepi, "Adaptive fully implicit multi-scale finite-volume method for multi-phase flow and transport in heterogeneous porous media," *J. Comput. Phys.*, vol. 217, no. 2, pp. 627–641, 2006.

[70] J. Douglas, Jr., F. Furtado, and F. Pereira, "On the numerical simulation of waterflooding of heterogeneous petroleum reservoirs," *Comput. Geosci.*, vol. 1, no. 2, pp. 155–190, 1997.

[71] Z. Chen and R. E. Ewing, "Stability and convergence of a finite element method for reactive transport in ground water," *SIAM J. Numer. Anal.*, vol. 34, no. 3, pp. 881–904, 1997.

[72] P. Bastian and R. Helmig, "Efficient fully-coupled solution techniques for two-phase flow in porous media: Parallel multigrid solution and large scale computations," *Adv. Water Resour.*, vol. 23, no. 3, pp. 199–216, 1999.

[73] U. T. Mello, J. R. P. Rodrigues, and A. L. Rossa, "A control-volume finite-element method for three-dimensional multiphase basin modeling," *Mar. Pet. Geol.*, vol. 26, no. 4, pp. 504–518, 2009.

[74] Z. Chen, G. Huan, and Y. Ma, *Computational methods for multiphase flows in porous media*. SIAM, Philadelphia, 2006, vol. 2.

[75] F. Kwok and H. A. Tchelepi, "Potential-based reduced Newton algorithm for nonlinear multiphase flow in porous media," *J. Comput. Phys.*, vol. 227, no. 1, pp. 706–727, 2007.

[76] C. N. Dawson, H. Klie, M. F. Wheeler, and C. S. Woodward, "A parallel, implicit, cell-centered method for two-phase flow with a preconditioned Newton–Krylov solver," *Comp. Geosci.*, vol. 1, no. 3-4, pp. 215–249, 1997.

[77] H. Klie, M. Rame, and M. Wheeler, "Two-stage preconditions for inexact Newton methods in multi-phase reservoir simulation," *Tech.Rep.CRPC-TR96641*, 1996.

[78] D. W. Peaceman, *Fundamentals of numerical reservoir simulation*. Elsevier, 1977, vol. 6.

[79] M. A. Theodoropoulou, V. Sygouni, V. Karoutsos, and C. D. Tsakiroglou, "Relative permeability and capillary pressure functions of porous media as related to the displacement growth pattern," *Int. J. Multiphas. Flow.*, vol. 31, no. 10, pp. 1155–1180, 2005.

[80] R. Brooks and A. Corey, "Hydraulic properties of porous media," *Hydrology Papers, Civil Engineering Department, Colorado State University, Fort Collins, CO.*, 1964.

[81] Y. Yortsos and J. Chang, "Capillary effects in steady-state flow in heterogeneous cores," *Transp. Porous Media*, vol. 5, no. 4, pp. 399–420, 1990.

[82] H. A. Friis and S. Evje, "Numerical treatment of two-phase flow in capillary heterogeneous porous media by finite-volume approximations," *Int. J. Numer. Anal. Model.*, vol. 9, no. 3, pp. 505–528, 2011.

[83] H. Hoteit and A. Firoozabadi, "Numerical modeling of two-phase flow in heterogeneous permeable media with different capillarity pressures," *Adv. Water. Resour.*, vol. 31, no. 1, pp. 56–73, 2008.

[84] V. Ginting, R. Ewing, Y. Efendiev, and R. Lazarov, "Upscaled modeling in multiphase flow applications," *Comput. Appl. Math.*, vol. 23, no. 2-3, pp. 213–233, 2004.

[85] J. E. P. Monteagudo and A. Firoozabadi, "Comparison of fully implicit and IMPES formulations for simulation of water injection in fractured and unfractured media," *Inter. J. Numer. Meth. Eng.*, vol. 69, no. 4, pp. 698–728, 2007.

[86] A. Negara, M. F. El-amin, and S. Sun, "Simulation of CO_2 plume in porous media: consideration of capillarity and buoyancy effects," *International Journal of Numerical Analysis and Modeling, Series B*, vol. 2, no. 4, pp. 315–337, 2011.

[87] S. Lacroix, Y. Vassilevski, J. Wheeler, and M. Wheeler, "Iterative solution methods for modeling multiphase flow in porous media fully implicitly," *SIAM J. Sci. Comput.*, vol. 25, no. 3, pp. 905–926, 2003.

[88] B. Lu, "Iteratively coupled reservoir simulation for multiphase flow in porous media," Ph.D. dissertation, University of Texas at Austin, 2008.

[89] B. Lu and M. F. Wheeler, "Iterative coupling reservoir simulation on high performance computers," *Pet. Sci.*, vol. 6, no. 1, pp. 43–50, 2009.

[90] J. Kou and S. Sun, "A new treatment of capillarity to improve the stability of IMPES two-phase flow formulation," *Comput. Fluids*, vol. 39, no. 10, pp. 1923–1931, 2010.

[91] R. C. MacDonald and K. H. Coats, "Methods for numerical simulation of water and gas coning," *Trans. SPE AIME.*, vol. 10, no. 04, pp. 425–436, 1970.

[92] J. S. Nolen and D. W. Berry, "Tests of the stability and time-step sensitivity of semi-implicit reservoir stimulation techniques," *Trans. SPE AIME.*, vol. 12, no. 03, pp. 253–266, 1972.

[93] G. W. Thomas and D. Thurnau, "Reservoir simulation using an adaptive implicit method," *Soc. Pet. Eng. J.*, vol. 23, no. 05, pp. 759–768, 1983.

[94] P. A. Forsyth and P. H. Sammon, "Practical considerations for adaptive implicit methods in reservoir simulation," *J. Comput. Phys.*, vol. 62, no. 2, pp. 265–281, 1986.

[95] Y. Brenier and J. Jaffré, "Upstream differencing for multiphase flow in reservoir simulation," *SIAM J. Numer. Anal.*, vol. 28, no. 3, pp. 685–696, 1991.

[96] K. A. Lie, S. Krogstad, I. S. Ligaarden, J. R. Natvig, H. M. Nilsen, and B. Skaflestad, "Open-source MATLAB implementation of consistent discretisations on complex grids," *Comp. Geosci.*, vol. 16, no. 2, pp. 297–322, 2012.

[97] J. E. Aarnes, T. Gimse, and K. A. Lie, "An introduction to the numerics of flow in porous media using MATLAB," in *Geometric Modelling, Numerical Simulation, and Optimization: Applied Mathematics at SINTEF.* Springer, 2007, pp. 265–306.

[98] M. A. Christie and M. J. Blunt, "Tenth SPE comparative solution project: A comparison of upscaling techniques," in *SPE Reser. Eval. Eng.*, vol. 4. Society of Petroleum Engineers, 2001, pp. 308–317.

[99] X. S. L. and J. W. D., "SuperLU_DIST: A scalable distributed-memory sparse direct solver for unsymmetric linear systems," *ACM Trans. Math. Software*, vol. 29, no. 2, pp. 110–140, June 2003.

[100] J. E. P. Monteagudo and A. Firoozabadi, "Control-volume method for numerical simulation of two-phase immiscible flow in two-and three-dimensional discrete-fractured media," *Water. Resour. Res.*, vol. 40, no. 7, 2004, DOI: 10.1029/2003WR002996.

[101] X.-C. Cai and M. Dryja, "Domain decomposition methods for monotone nonlinear elliptic problems," *Contemp. Math.*, vol. 180, pp. 21–27, 1994.

[102] A. Granas and J. Dugundji, *Fixed Point Theory.* Springer, 2003.

[103] R. Ernst, B. Flemisch, and B. Wohlmuth, "A multiplicative Schwarz method and its application to nonlinear acoustic-structure interaction," *ESAIM, Math. Model. Numer. Anal.*, vol. 43, no. 3, pp. 487–506, 2009.

[104] C. Hirsch, *Numerical Computation of Internal and External Flows*. John Wiley & Sons, New York, 1990.

[105] V. E. Henson and U. M. Yang, "BoomerAMG: A parallel algebraic multigrid solver and preconditioner," *Appl. Numer. Math.*, vol. 41, no. 1, pp. 155–177, 2002.

[106] G. De Vahl Davis, "Natural convection of air in a square cavity: a bench mark numerical solution," *Int. J. Numer. Methods Fluids*, vol. 3, no. 3, pp. 249–264, 1983.

[107] C.-H. Zhang, W. Zhang, and G. Xi, "A pseudospectral multidomain method for conjugate conduction-convection in enclosures," *Numer. Heat. Tr. B-FUND*, vol. 57, no. 4, pp. 260–282, 2010.

APPENDICES

A Papers Submitted and Under Preparation

• L. Liu, D. E. Keyes, S. Sun, "Fully Implicit Two-phase Reservoir Simulation With the Additive Schwarz Preconditioned Inexact Newton Method", *SPE Reservoir Characterization and Simulation Conference and Exhibition (RCSC)*, 16-18 September 2013, Abu Dhabi, UAE. DOI: http://dx.doi.org/10.2118/166062-MS.

• L. Liu, D. E. Keyes, "Field-split Preconditioned Inexact Newton Algorithms", *SIAM Journal on Scientific Computing*, vol. 37, no. 3, pp. A1388-A1409, 2015.

• L. Liu, D. E. Keyes, "Convergence Analysis for the Multiplicative Schwarz Preconditioned Inexact Newton Algorithm", *Submitted to SIAM Journal on Numerical Analysis*, 2015.

• L. Liu, D. E. Keyes. "Nonlinear Schwarz Preconditioning", *In preparation for the Proceedings of the 23rd International Conference on Domain Decomposition Methods, Springer Lecture Notes in Computational Science and Engineering*, 2015.