NONLINEAR PRECONDITIONING APPLIED TO INTERNAL FLOWS

by

Ersin Yıldız

B.S., Mechanical Engineering, Yıldız Technical University, 2015

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Master of Science

Graduate Program in Mechanical Engineering
Boğaziçi University

2018

ACKNOWLEDGEMENTS

In the beginning, I want to thank my family for their encouragement and endless support throughout my entire life.

I would like to thank my supervisor Assoc. Prof. Ali Ecder for his guidance, comments and academic vision throughout my study.

I would like to express my gratitude to my friends, especially to those in Hisarüstü. I have so many great memories with them.

I would like to thank to all of the CFD Lab. members, especially to Ali Berk Kahraman. His companionship was irreplaceable.

ABSTRACT

NONLINEAR PRECONDITIONING APPLIED TO INTERNAL FLOWS

Solution of large and sparse nonlinear system is required in many computational science and engineering areas. Well-known inexact Newton method converges fast with a good initial guess, however such an initial guess is hard to obtain for the problems with unbalanced nonlinearity, such as fluid flow problems with a high Reynolds number. Nonlinear preconditioning has been proposed to deal with this issue by constructing a preconditioned nonlinear system and solving it by inexact Newton. This approach is regarded as left nonlinear preconditioner since it changes the nonlinear system. On the other hand, right nonlinear preconditioner eliminates some previously chosen components of the unknowns to form a better approximate solution to participate in inexact Newton iterations. We consider three preconditioning methods: additive Schwarz preconditioned inexact Newton (ASPIN) which is based on domain decomposition methods, field-split preconditioned inexact Newton (FSPIN) which splits the components and solves them in an additive Schwarz manner, and nonlinear elimination preconditioned inexact Newton (NEPIN). In this thesis, various studies have been conducted to observe the behavior of the algorithms with respect to two different implementations of finite difference method on boundaries, some tolerance parameters and some algorithmic parameters for steady-state lid-driven cavity problem at several values of Reynolds number. Our numerical tests obtained on parallel show that NEPIN is more robust than inexact Newton and other nonlinear preconditioning methods.

ÖZET

DOĞRUSAL OLMAYAN ÖNKOŞULLAMA VE İÇ AKIŞLARA UYGULANMASI

Büyük ve seyrek doğrusal olmayan sistemlerin çözümü bir çok hesaplamalı bilimlerde ve mühendislik alanlarında gereklidir. Güzel bir başlangıç noktası tedarik edilmiş ünlü Newton metod hızlı şekilde yakınsayabilmektedir, fakat öyle bir başlangıç noktası elde etmek doğrusal olmayan açıdan dengesiz problemler için zordur. Doğrusal olmayan önkoşullama bu durumla baş etmek üzere öne sürülmüş olup, bu işlemi önkoşullanmış bir doğrusal olmayan sistem oluşturup, onu kesin olmayan Newton metodu ile çözerek yapmaktadır. Bu yaklaşım doğrusal olmayan sistemi değiştirdiği için, sol doğrusal olmayan önkoşullama olarak addedilir. Öte yandan, sağ doğrusal olmayan önkoşullama, sıradaki Newton iterasyonuna katılmak üzere daha iyi bir yaklaşık sonuç elde etmek amacıyla, önceden seçilmiş bir takım bilinmeyen elemanları elimine eder. Üç adet doğrusal olmayan önkoşullama metodu inceleyeceğiz: alan parçalama tabanlı bir method olan additive Schwarz preconditioned inexact Newton (ASPIN), bilinmeyen elemanları ayıran ve çözen bir method olan field-split preconditioned inexact Newton (FSPIN), ve nonlinear elimination preconditioned inexact Newton (NEPIN). Bu tezde, çeşitli Reynolds sayılarındaki zamandan bağımsız kapak-tahrikli çukur problemi için, tezde sunulan algoritmaların, sınırlardaki iki tür sonlu farklar ayrıklaştırma uygulamasına, bazı tölerans değişkenlerine ve bazı algoritmasal değişkenlere göre davranışlarının gözlenmesi amacıyla birçok sayıda çalışma yapılmıştır. Paralel hesaplama ile elde edilen sayısal testlerimiz gösteriyor ki NEPIN diğer metodlardan ve Newton methodundan daha dayanıklıdır.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

A	System matrix of a linear system
A_i	ith partition of a system matrix of a linear system
a	A scalar used in forcing term calculation
b	Right hand side vector of a linear system
C_i	ith corner of the square domain
c	The number of components to be eliminated
e_j	jth unit vector
F	Nonlinear function
F_i	ith component or ith partition of a nonlinear function
$\mathcal{F}$	Nonlinear preconditioned function
f	Merit function
G	Representation of nonlinear preconditioner
$g^{(k)}$	Preconditioned function evaluated at the kth iteration
$g^{(k,j)}$	Intermediate $g^{(k)}$ at jth iteration
$g_i^{(k,j)}$	ith partition of the intermediate $g^{(k)}$ at jth iteration
g_i	Subdomain corrector in right nonlinear preconditioners
h	The length between two adjacent mesh points
h_o	Overlap size in additive Schwarz nonlinear preconditioner
I_i	Identity matrix corresponds to ith subdomain
J	Jacobian matrix
$\mathcal{J}$	Jacobian of the nonlinearly preconditioned function
$\mathcal{J}_i$	ith partition of $\mathcal{J}$
L	The length used in Reynolds number calculation
l^2	A vector norm
M^{-1}	Preconditioner matrix
m	A parameter in two simple examples
$maxit_{global-nonlinear}$	Maximum iteration for global nonlinear systems
$maxit_{local-nonlinear}$	Maximum iteration for local nonlinear systems

N	The total number of subdomains or sub-systems
n	Size or dimension of the system or domain
n_i	Dimension of the ith subdomain
p	Non-dimensional pressure term
$p^{(k)}$	Newton direction at the kth nonlinear iteration
q	The number of colors in multi-colored Jacobian calculation
R	A space in 1-dimension
Re	Reynolds number
R^n	A space in n-dimension
R_i	Restriction matrix corresponds to ith sub-system
R_i^T	Interpolation matrix corresponds to ith sub-system
r	Nonlinear residual operator or vector
S	Index set of whole domain
S_i	Index set of ith subdomain or ith subset
$s^{(k)}$	Initial slope at the kth nonlinear iteration
T	Subset corrector in nonlinear elimination preconditioner
T_i	Subdomain corrector in additive Schwarz preconditioner
t	Non-dimensional time
u	The unknown vector of a nonlinear or linear system
u^*	The exact solution of a nonlinear or linear system
$u^{(k)}$	An approximate solution at the kth iteration
$u^{(0)}$	Initial guess for the solution of a nonlinear function
u_e	Eliminated approximate solution u
u_i	ith component or ith partition of the vector u
u_i^c	Complementary of the ith partition of the vector u
$(u_i^c)^{(k)}$	Complementary of the ith partition of the vector $u^{(k)}$
$\vec{V}$	Velocity vector
V_i	Any vector corresponds to ith subspace
V_m	The mean velocity used in Reynolds number calculation
x	Direction of horizontal axis in space
y	Direction of vertical axis in space

β	The parameter that continuation method to be applied on
β^*	The true value of the parameter in continuation method
$\beta^{(k)}$	The intermediate value of β
Γ_i	ith edge of the square domain
γ	A scalar used in forcing term calculation
Δ	Laplacian operator
$\vec{\nabla}$	Gradient operator
∂	Partial differentiation operator
ϵ	Step size in Jacobian matrix calculation
ϵ_{switch}	A parameter in NEPIN algorithm
$\epsilon_{global-nonlinear}$	Tolerance for global nonlinear iterations
$\epsilon_{global-linear}$	Tolerance for global linear iterations
$\epsilon_{local-nonlinear}$	Tolerance for local nonlinear iterations
η_k	Forcing term at the kth nonlinear iteration
θ^k	The angle between $p^{(k)}$ and the negative steepest direction
κ	Condition number operator
$\lambda^{(k)}$	Damping parameter at k nonlinear iteration
$\lambda_j^{(k)}$	jth calculated damping parameter at k nonlinear iteration
λ_{max}	Maximum value $\lambda^{(k)}$ can take
λ_{min}	Minimum value $\lambda^{(k)}$ can take
ν	Kinematic viscosity
π	Pi number
ϕ	A vector consists of e_j's corresponds to the same colors
Ω	Domain
Ω_i	ith subdomain
Ω_i^c	Complementary of ith subdomain
ω	Vorticity component
$\vec{\omega}$	Vorticity vector
ω_k	k component of vorticity

LIST OF ACRONYMS/ABBREVIATIONS

ASPEN	Additive Schwarz Preconditioned Exact Newton
ASPIN	Additive Schwarz Preconditioned Inexact Newton
CFD	Computational Fluid Dynamics
CPU	Central Processing Unit
CSC	Compressed Sparse Column
FSPIN	Field-Split Preconditioned Inexact Newton
GMRES	Generalized Minimal Residual
ILU	Incomplete LU
IN	Inexact Newton
INB	Inexact Newton with Backtracking
INB-ANE	Inexact Newton with Backtracking-Adaptive Nonlinear Elimination Preconditioning
NEPEN	Nonlinear Elimination Preconditioned Exact Newton
NEPIN	Nonlinear Elimination Preconditioned Inexact Newton
NKS	Newton-Krylov-Schwarz
SOR	Successive Over Relaxation

1. INTRODUCTION

The main objective of this thesis is to develop robust, scalable and parallel algorithm and software to solve large nonlinear systems obtained from the discretization of partial differential equations in computational fluid dynamics (CFD). More specifically, we focus on the utilization of Schwarz methods and nonlinear elimination method which are applied to well known Newton method to precondition the original system nonlinearly in order to solve the nonlinear flow problems for the cases where Newton's method stagnates or fails.

While developing an algorithm, one needs to consider two concepts; robustness and scalability. The algorithm is called robust if the solution process is not affected much when some of the parameters are changed. For example, in fluid dynamics, Reynolds number (Re) is a non-dimensional parameter that specifies the regime of the flow. A change in Reynolds number can alter the solution very severely for classical nonlinear solvers such as Newton's method and even the solution process may diverge. Some other parameters are generally the grid size, the initial guess for iterative solver, the Mach number in compressible flow. We aim to develop an algorithm that is not much sensitive to these parameters. Scalability is how the algorithm behaves when the number of processors and the size of problem grows. This issue is being important everyday since we solve larger problems on computers that have increasing number of processors. Thus, an algorithm should exploit the power of multi-processor architecture of a central processing unit (CPU) especially for large problems.

Domain decomposition methods have been an interesting area of research for three decades, however its idea and first proposed study originated in 1870s. German mathematician, H. A. Schwarz has proved the existence of the solution of elliptic boundary value problems with Dirichlet boundary conditions over an irregular domain decomposed into two subdomains by solving the sub-problems on the corresponding subdomains alternatingly. Although Schwarz's method was not intended as a numerical method, after the PhD Thesis of Lions [3] in 1980s, Schwarz's method started to

be considered as a numerical method. Today, it is widely used method and known as Classical Alternating Schwarz method.

The idea of domain decomposition for boundary value problems is dividing the domain into smaller subdomains and solving the problem on the subdomains with appropriate boundary conditions. Schwarz decomposed the domain into overlapped subdomains, however it is possible to use non-overlapping subdomains and in that case it is called Schur complement methods. In most cases, solving subproblems is easier than solving the whole system. The distribution of the data to be used in subdomain solution routine and the collection of the solution of subproblems determinates the class of the method. For detailed information on domain decomposition procedure we refer [4].

The application of domain decomposition to scientific computing varies with some different usage. In parallel computing, domain decomposition means distributing the data over a number of processors and solving the subproblems simultaneously. In this case, solution of the subproblems has to be independent of the solution of the other subdomains. Another use of domain decomposition is decomposing the domain with regard to the regions that can be modeled by different partial differential equations, for example fluid-solid interaction problems. Other than those approaches, domain decomposition method can be used as preconditioners for linear and nonlinear systems to improve their convergence. In this thesis, we employ domain decomposition methods to develop linear and nonlinear preconditioners to solve nonlinear flow problems.

1.1. Linear Preconditioning

In many computational engineering area, the solution of large linear systems arising from the discretization of differential equations is required. Since the system is large, employing direct methods such as Gaussian elimination may not be the best choice or may even not be applicable with regard to both storage and CPU time. Thus, iterative methods are used to solve such systems and one of the most known method is Generalized Minimal Residual (GMRES) Method [5] which uses Krylov subspace.

Solving linear systems iteratively may be hard for some ill-conditioned systems. The linear system $Au = b$ is called ill-conditioned when the condition number of A matrix $\kappa(A) = ||A||\ ||A^{-1}||$ is large. The condition number depends on many things such as discretization of the differential equations, the grid size and some problem parameters. To overcome this issue, one may solve $M^{-1}Au = M^{-1}b$ instead of solving the original system $Au = b$ where M^{-1} is called linear preconditioner matrix. M^{-1} should be close to A^{-1} to obtain small condition number $\kappa(M^{-1}A)$. In ideal case, M^{-1} should be equal to A^{-1}. However, forming A^{-1} is expensive as solving the original linear system. Thus, it is better to find an approximation of A^{-1} as a linear preconditioner matrix.

The type of the linear preconditioner is related with how we approximate the system matrix A^{-1} to obtain linear preconditioner M^{-1}. For example, Jacobi, Successive Over Relaxation (SOR), incomplete LU (ILU) factorization preconditioners are some of linear preconditioner types, for details see [6]. In this work, we use additive Schwarz linear preconditioner based on domain decomposition method to solve ill-conditioned linear system within the nonlinear solvers.

The linear preconditioning described above is considered as left linear preconditioning since the preconditioner is applied to the left of system matrix A. With this regard, we apply M^{-1} to the right of A to obtain right preconditioned system $AM^{-1}\ Mu = b$. To solve this type of preconditioned system, one needs to solve $AM^{-1}\ y = b$ for y at first. Then, u is found by solving $Mu = y$.

1.2. Nonlinear Preconditioning

Nonlinear systems arising from the discretization of nonlinear differential equations are encountered in many areas. In contrast to an approach that deals with nonlinear systems by linearizing the system and solving corresponding linearized system, we focus on solving nonlinear system in a coupled way using well known Newton's method.

Newton's method can be derived by two-termed multivariable Taylor expansion of the function F at the current approximate solution $u^{(k)}$ for the next approximate solution $u^{(k+1)}$ as demonstrated below where higher order terms are neglected.

$$F(u^{(k+1)}) = F(u^{(k)}) + J(u^{(k)})(u^{(k+1)} - u^{(k)}) \tag{1.1}$$

Here, J is the Jacobian of the nonlinear function F, meaning that $J = F' = \left(\frac{\partial F_i}{\partial u_j}\right)_{n \times n}$. Note that, F is a vector valued function and the unknown vector u contains the unknowns of the nonlinear system. Newton's method is an iterative method that computes the next approximate solution $u^{(k+1)}$ by solving the linear system

$$J(u^{(k)})(u^{(k+1)} - u^{(k)}) = -F(u^{(k)}) \tag{1.2}$$

with a given current approximate solution $u^{(k)}$. Note that, we derived above equation by setting $F(u^{(k+1)}) = 0$ in Equation 1.1 since we aim to reach the solution which reduces the nonlinear residual $F(u^{(k+1)})$.

Newton method is commonly used in many areas in scientific computing thanks to its easy implementation and fast convergence for a good initial guess. However, finding such a good initial guess is difficult especially for the systems that have *unbalanced nonlinearity* for example fluid flow problems with high Reynolds number [1].

We use the term "unbalanced nonlinear" or "nonlinear stiff" when the solution of the nonlinear problem has disparate spatial scales. This happens when some equations are locally more nonlinear than others, especially near shock waves, reaction fronts or the boundary of the domain. Unbalanced nonlinearity affects the convergence of Newton's method such that it may stagnate to a minimum or diverge. For example in lid-driven cavity problem with high Reynolds number, vorticity equation has unbalanced nonlinearity and Newton's method fails to converge even with globalization techniques such as line search. Nonlinear preconditioning aims to deal with this convergence problem of Newton's method for nonlinearity unbalanced problems.

The first nonlinear preconditioning concept was proposed by Chan and Jackson [7] in 1984. Their idea was applying linear preconditioner to the linear Jacobian systems within the Newton iterations without forming or storing Jacobian matrix. For any vector v. they regarded the matrix-vector product $w = M^{-1}v$ as a solution of the nonlinear system $F(w) = Jw - v$ where M^{-1} is the preconditioner matrix and $M = J$ is the Jacobian matrix. The nonlinear derived system was solved by using nonlinear relaxation methods. In that case, the convergence of Newton method was not affected by the application of nonlinear preconditioning since it was applied to the Jacobian system within Newton iteration.

After many years, a new nonlinear preconditioning concept was introduced by Cai and Keyes [1] in 2002. Their aim was to remove the unbalanced nonlinearities of the nonlinear system. This approach influenced scientific community and the forthcoming studies have improved the concept of nonlinear preconditioning.

In analogy to linear preconditioning, nonlinear preconditioning can be categorized as left and right nonlinear preconditioning. However, the difference between left and right nonlinear preconditioning is not very straightforward as in linear preconditioning. We describe them in the following subsections.

1.2.1. Left Nonlinear Preconditioning

The class of left nonlinear preconditioning aims to change the nonlinearity of the system, by constructing a new nonlinear system which has the same solution as the original system. Cai and Keyes [1] proposed Additive Schwarz Preconditioned Inexact Newton (ASPIN) method which is domain decomposition based in order to construct a preconditioned system implicitly and they showed that ASPIN converges well for a wide range of Reynolds number up to $Re = 10^4$ for steady-state incompressible lid-driven cavity problem. Construction of the preconditioned system was done by solving local nonlinear problems on subdomains in additive Schwarz manner simultaneously. However, the scalability of the algorithm was not so good since an increase in the number of subdomains was affecting linear iteration number by considerable amount.

Two-level variation of ASPIN algorithm was proposed by Cai *et al.* [8] in which local nonlinear subproblems are solved on a coarse grid to reduce the number of global linear iterations. Hwang and Cai [9] improved the two-level ASPIN algorithm by integrating local linear coarse solution instead of local nonlinear coarse solution and this reduced the cost of the two-level ASPIN.

The cost of the left nonlinear preconditioning is usually expensive because constructing a preconditioned nonlinear system involves a number of solution of nonlinear or linear systems. Thus, it is best to employ it when the problem has high unbalanced nonlinearity such that Newton's method fails to converge or stagnates.

1.2.2. Right Nonlinear Preconditioning

The idea of right nonlinear preconditioning is eliminating the unknowns that cause high nonlinearities before Newton iterations by solving the troubled unknowns in sub-nonlinear problem or problems. By doing this, Newton iterations receive a better approximate solution to participate to the following linear Jacobian system in Newton iteration and the nonlinear system remains the same on the contrary of left nonlinear preconditioners.

Nonlinear elimination preconditioned Newton method was proposed by Lanzkron *et al.* [10]. Cai and Li [11] introduced restricted additive Schwarz based nonlinear elimination preconditioning which solves all of the subdomain nonlinear problems over each ones subdomain in a nonlinear restricted additive Schwarz manner to eliminate local high nonlinearities. They showed that it works well even for steady-state incompressible lid-driven cavity problem with high Reynolds number up to $Re = 10^5$. Yang and Hwang [2] recently proposed an adaptive nonlinear elimination preconditioner algorithm that decides which variables cause high nonlinearity and eliminates them by solving sub-nonlinear problems.

Right nonlinear preconditioners are cheaper and easier to implement than left nonlinear preconditioner methods since they do not construct a new nonlinear system.

1.3. Motivation and Objectives

In literature, many nonlinear preconditioners have been proposed and tested on some nonlinear problems mainly on steady-state incompressible lid-driven cavity problem. However, most of the tests comprise the parametric studies for the relevant nonlinear preconditioner. Comparison of different nonlinear preconditioner methods, especially the comparison between right and left nonlinear preconditioners has not been conducted. One objective of this thesis is to make a comparison between different nonlinear preconditioners including right and left preconditioners for steady-state incompressible lid-driven cavity problem.

The algorithms of nonlinear preconditioner methods include many tolerance parameters and some algorithmic choices within them. Our second objective is investigate the effects of certain parameters to the nonlinear preconditioners. We also study the algorithmic choices such as partitioning choices in ASPIN algorithm, the choice of the threshold parameter which determines the application of elimination routine in nonlinear elimination preconditioned inexact Newton (NEPIN) algorithm to have a better understanding of the behavior of the algorithms.

When we examine the literature on the solution of finite difference discretized steady-state incompressible lid-driven cavity problem by nonlinear preconditioned methods, we see that there is no any solution plots to show the verification of the solution for ASPIN and field-split preconditioned inexact Newton (FSPIN) methods. They use first order discretization for vorticity component on boundaries. According to the streamlines that we obtained, we found that the solution of this discretizated problem is not physically true for high Reynolds numbers. On the other hand, the NEPIN study by Yang and Hwang [2] used so-called efficient 2nd order discretization for vorticity components on boundaries which leads better approximate solution for steady-state incompressible lid-driven cavity problem especially for high Reynolds numbers. Our last objective is to solve this discretizated problem by ASPIN and FSPIN which has not been done before within our knowledge and observe the results of these algorithms.

1.4. Organization of the Dissertation

This thesis comprises of five chapters. In the next chapter, we describe inexact Newton algorithm and some techniques on choosing some certain parameters within the algorithm. After a brief convergence analysis of inexact Newton, we mention how we deal with the computation of Jacobian matrices. Lastly, the framework of additive Schwarz linear preconditioner and brief information on parameter continuation method are given. We describe the nonlinear preconditioner methods and explain their algorithms in detail in Chapter 3. Chapter 4 is devoted to numerical analyses. First, to show the effect of left and right nonlinear preconditioner, a number of analyses have been done on two simple examples with unbalanced nonlinearity. Then, we perform many tests on steady-state incompressible lid-driven cavity problem by using the algorithms described in Chapter 2 and 4. Some parametric studies have been conducted and a wide discussion on the results has been done. We conclude our study in Chapter 5 and point out some possible further research interests.

2. INEXACT NEWTON METHOD AND RELATED TOOLS

In the previous chapter, very brief introduction to Newton's method has been given. Most of the time we are unwilling to solve the Jacobian system within Newton's method exactly since it may cause some problems that will be discussed, or even it may not be possible due to computer limits. Thus, iterative linear solvers are employed to solve Jacobian systems with an acceptable tolerance and this brings the inexactness to the Newton's method.

In this chapter, inexact Newton (IN) method [12] is described which is one of the well known and most used method to solve nonlinear set of equations in a coupled way. Then some choices for forcing factor term in inexact Newton algorithm have been given which brings the inexactness of Newton's method and also acts as a tolerance parameter on the solution of the Jacobian systems. We discuss one of the globalization technique namely line search algorithm. Since the calculation of Jacobian matrix is an important issue with regards to storage limits and CPU time, how we managed to deal against these constraints is explained. After discussing the convergence of the inexact Newton method, additive Schwarz linear preconditioner and continuation method have been explained.

2.1. Inexact Newton Method Algorithm and Details

Inexact Newton method is a popular method to solve nonlinear set of equations because of its easy implementation and rapid convergence when the initial guess is close to the desired solution. It is a building block to our study as all of the nonlinear solver algorithms in this study include inexact or exact Newton method, in other words, the preconditioning methods serve Newton's method to make it more robust.

Consider the discrete nonlinear function $F : R^n \to R^n$. The nonlinear system is defined as

$$F(u) = 0 \tag{2.1}$$

where $u = (u_1, u_2, ..., u_n)^T$ and F is a set of functions $F = (F_1, F_2, ..., F_n)^T$. Each of these functions are $F_i = F_i(u_1, u_2, ..., u_n)$. Assume that, the system in Equation 2.1 has an exact solution u^*. We seek an approximate solution u that minimizes the nonlinear residual $r(u) = F(u)$.

Let $u^{(0)}$ be the initial guess and $u^{(k)}$ be the current approximate solution of the nonlinear system $F(u) = 0$. For a given current approximate solution, IN computes the new approximate solution $u^{(k+1)}$ by following the algorithm in Figure 2.1.

Step 1: Find the inexact Newton direction $p^{(k)}$ by solving following Jacobian system approximately

$$J(u^{(k)})p^{(k)} = -F(u^{(k)}) \tag{2.2}$$

such that

$$\|F(u^{(k)}) + J(u^{(k)})p^{(k)}\| \leq \eta_k \|F(u^{(k)})\| \tag{2.3}$$

for some $\eta_k \in [0, 1)$.

Step 2: Compute the new approximate solution

$$u^{(k+1)} = u^{(k)} + \lambda^{(k)} p^{(k)} \tag{2.4}$$

where $\lambda^{(k)}$ is a damping parameter.

Figure 2.1. Inexact Newton Algorithm.

The inexact Newton direction $p^{(k)}$ (also called search direction) is found in Step 1 by solving the Jacobian system shown in Equation 2.2. Here, the right hand side of the linear system is negative of the function evaluation at the current approximate solution $u^{(k)}$ and Jacobian matrix evaluated at $u^{(k)}$ is the system matrix of the linear system in the left hand side. The negative sign in the right hand side corresponds to the decrease in the nonlinear residual of the system. In other words, it ensures the direction to be the steepest descend direction not the gradient direction.

In IN algorithm, η_k is forcing term that determines how accurately the Jacobian system needs to be solved by an iterative linear solver such as Generalized Minimal Residual (GMRES) [5]. However, the original Newton method was proposed without any condition to the solution of the Jacobian system meaning that Equation 2.2 had to be solved exactly by for example Gaussian elimination. Two significant issues may arise if the Jacobian system shown in Equation 2.2 is solved exactly:

(i) If the system is large, computing the exact solution of the Jacobian system is very expensive and time-consuming.
(ii) Oversolving the Jacobian system may lead poor convergence or divergence when $u^{(k)}$ is far away from the exact solution.

The choice of η_k may be critical since it affects the cost and the convergence of the algorithm. Many approaches have proposed which are based on residual norms, updates and previous information. We mention the two common choices proposed by [13].

(i) Choice 1: Given $\eta_0 \in [0, 1)$, calculate η_k by using one of the following equations

$$\eta_k = \frac{||F(u^{(k)}) - F(u^{(k-1)}) - J(u^{(k-1)})p^{(k-1)}||}{||F(u^{(k-1)})||} \tag{2.5}$$

$$\eta_k = \frac{abs\left(||F(u^{(k)})|| - ||F(u^{(k-1)}) + J(u^{(k-1)})p^{(k-1)}||\right)}{||F(u^{(k-1)})||} \tag{2.6}$$

(ii) Choice 2: Given $\gamma \in [0,1]$, $\alpha \in (1,2]$ and $\eta_0 \in [0,1)$, choose

$$\eta_k = \gamma \left(\frac{||F(u^{(k)})||}{||F(u^{(k-1)})||}\right)^{\alpha} \tag{2.7}$$

In Step 2 of the IN algorithm in Figure 2.1, $\lambda^{(k)} \in [\lambda_{min}, \lambda_{max}]$ is a damping factor which determines how far should we go through the search direction $p^{(k)}$. λ_{min} and λ_{max} are used to preserve the strong convergence of the algorithm, in our implementation, they equal to 0.1 and 1 respectively. λ_{min} prevents very small steps which is significant because the fact that very small steps may lead to stagnation. We note that, in some studies, combination of line search method (especially cubic backtracking) and inexact Newton is called inexact Newton with backtracking (INB) method.

To ensure the reduction in nonlinear residual, $\lambda^{(k)}$ should be selected so that

$$f(u^{(k)} + \lambda^{(k)}p^{(k)}) \leqslant f(u^{(k)}) + \alpha^{(k)}\lambda^{(k)}s^{(k)} \tag{2.8}$$

satisfied where f is a merit function defined $f : R^n \to R$, $f(u) = \frac{1}{2}||F(u)||^2$ and $s^{(k)}$ is the initial slope which can be approximated by $s^{(k)} = -||F(u^{(k)})||^2$. In the literature, several globalization techniques are used on the determination of $\lambda^{(k)}$ such as line search and trust region methods. In this thesis, we briefly describe two line search techniques that we use. For details of the globalization techniques, we refer [14].

2.2. Line Search Techniques

The idea of line search techniques is that one can take small step rather than taking full step $p^{(k)}$ in Equation 2.4 when taking full step does not fulfill the reduction in the nonlinear residual norm. We use two different line search routines in our implementations; half-step routine and cubic backtracking routine.

In cubic backtracking routine, we search for an optimum $\lambda^{(k)}$ value which minimizes a one-dimensional minimization problem

$$min_{\lambda^{(k)} \in (0,1]} \; f(u^{(k)} + \lambda^{(k)} p^{(k)}). \tag{2.9}$$

Although the exact solution is possible, we find the optimum $\lambda^{(k)}$ value approximately by constructing quadratic and cubic polynomials by following [14]. To construct quadratic polynomial, we need three information about the minimization problem. Using $F(u^{(k)})$, $J(u^{(k)})$ and $F(u^{(k)} + p^{(k)})$ which are already calculated during Newton iteration, a candidate $\lambda_j^{(k)}$ can be found as root of the quadratic minimization problem.

If newly calculated $\lambda_j^{(k)}$ does not satisfy the condition shown in Equation 2.8, a cubic polynomial model is needed to be constructed which requires four information. In addition to the previous information, we include the information comes from the newly calculated $\lambda_j^{(k)}$, namely $F(u^{(k)} + \lambda_j^{(k)} p^{(k)})$. Until the acceptable $\lambda^{(k)} \in [\lambda_{min}, \lambda_{max}]$ is found, cubic polynomial models are constructed and solved for new $\lambda_j^{(k)}$ ensuring that $\lambda_{min} = 0.1$.

In half-step routine, we reduce $\lambda_j^{(k)}$ to $\lambda_{j+1}^{(k)} = \lambda_j^{(k)}/2$ if $\lambda_j^{(k)}$ does not satisfy the condition in Equation 2.8 starting from $\lambda_0^{(k)} = 1$ through the minimum limit $\lambda_{min} = 0.1$. Note that the cost of the half-step routine is much lower than cubic backtracking since it does not require any solution of a minimization problem.

2.3. Multi-colored Jacobian Generation

Jacobian matrix J is an essential since we solve nonlinear problems by using Newton's method. There are several techniques to calculate Jacobian such as finite differences, automatic differentiation and analytic formula. Alternatively, one may use matrix-free implementation if Krylov subspace methods are used to solve the linear Jacobian system as discussed in [15]. Since finite difference Jacobian calculation is very expensive for large problems, in our calculations, we generate Jacobian matrices by using multi-colored forward finite difference method [16] and store them as sparse matrices in Compressed Sparse Column (CSC) format.

Jacobian is a matrix that includes the derivatives of function F with respect to the vector u such that its entries are calculated by formula $\frac{\partial F_i(u)}{\partial u_j}$ for $i, j = 1, 2, ..., n$ where i and j are rows and columns of the Jacobian matrix respectively. Using forward finite difference scheme, we can approximate each partial derivative term of the Jacobian matrix at a given point u by

$$\frac{\partial F_i(u)}{\partial u_j} \approx \frac{F_i(u + \epsilon e_j) - F_i(u)}{\epsilon} \tag{2.10}$$

where e_j denotes jth unit vector. Parameter ϵ is the finite difference step size which is an important issue in the implementation. In the approximation shown in Equation 2.10, we see two error sources; the discretization of function F and the selection of ϵ. We have to choose as small as possible ϵ to have a good approximation of the derivatives. However if ϵ is chosen too small, the error grows because of the two facts that; the subtraction of two real valued function in finite precision and the division of two very small real number. To have a optimal value, we use $\epsilon = 10^{-8}$ which equals to the square root of the unit roundoff in double precision as used in [8, 17].

Instead of calculating each term in Jacobian matrix, one can calculate one column of it by one formula as follows.

$$\frac{\partial F(u)}{\partial u_j} \approx \frac{F(u + \epsilon e_j) - F(u)}{\epsilon} \tag{2.11}$$

To demonstrate the cost of the Jacobian calculation, let n be the system size of a problem. Then the size of Jacobian matrix would be $n \times n$. To form full Jacobian matrix by forward finite differences, one needs $n + 1$ function evaluations. When the problem is large, this becomes very expensive and time-consuming.

However, if the Jacobian matrix is sparse, we can save the cost by exploiting the pattern of the Jacobian. Consider an index set of the system $S = (1, 2, ..., n)$ where each integer corresponds for one F_i and u_i. We label each index in the set with a color by using a rule such that indices i and j are labeled with the same color only if F_i and F_j do not depend on the same variable u_k. After labeling the whole set, we get color partitions as $S_1, S_2, ..., S_q$ where q is the number of colors. For one color, we can calculate a column that includes all non-zero derivatives concerning the indices belong that color by one formula given below.

$$v = \frac{F(u + \phi) - F(u)}{\epsilon} \tag{2.12}$$

Here the vector ϕ is $\phi = \epsilon \left(\sum_{k \in S_i} e_k \right)$. After computing all columns, we get a compressed Jacobian matrix whose size is $n \times q$. In our calculations, we uncompressed this compressed Jacobian and store it as a sparse matrix.

Notice that, one needs only $q + 1$ function evaluations to form the full Jacobian matrix by using multi-colored finite differencing instead of $n + 1$ evaluations, and $q + 1$ is much smaller than $n + 1$ for well structured Jacobian matrices.

2.4. Convergence of Inexact Newton with Backtracking

In spite of including a globalization technique, inexact Newton method is still fragile. It converges fast with a good initial guess for the problems that do not have unbalanced nonlinearity. However, when we change some problem parameters such as Reynolds number for flow problems, inexact Newton method may stagnate and even diverge.

Here we investigate the angle θ^k between the Newton direction and the negative steepest direction for the current approximate solution $u^{(k)}$. Newton direction becomes weak if θ^k is too close to $\pi/2$. According to the observation done by Tuminaro et all [18], in a bad case, the combination of F and J may results in

$$\frac{1}{\kappa(J^{(k)})} \leqslant cos(\theta_k) \leqslant \frac{2}{\kappa(J^{(k)})} \tag{2.13}$$

where $cos(\theta_k) = \frac{-(s^{(k)})^T(J(u^{(k)})^T F(u^{(k)}))}{||(s^{(k)})^T|| \, ||J(u^{(k)})^T F(u^{(k)})||}$ and $\kappa(J^{(k)})$ is the condition number of the Jacobian evaluated at $u^{(k)}$. This means that if the Jacobian is ill-conditioned, $cos(\theta_k)$ becomes smaller, then the Newton direction becomes nearly orthogonal to the negative steepest direction. Moreover, preconditioning the linear system does not remediate this issue since linear preconditioner matrix M^{-1} does not appear in Equation 2.13 [19].

2.5. Additive Schwarz Linear Preconditioner

We have mentioned that the Jacobian system within the inexact Newton method is solved by an iterative method such as GMRES for some important reasons discussed earlier. For large and ill-conditioned system, it becomes an essential to employ a linear preconditioner to the Jacobian system to avoid too many linear iterations in GMRES. Applying Schwarz typed linear preconditioner to GMRES within the inexact Newton is proposed as Newton-Krylov-Schwarz (NKS) method by Cai *et al.* [20] and it has been being used in many studies.

In this section, we give the framework of one-level additive Schwarz linear preconditioner, which is based on domain decomposition methods, for a general linear system $Au = b$.

To construct the linear preconditioner matrix M^{-1} in an additive Schwarz manner, we first partition the domain into a number of subdomains. Consider the domain Ω and let Ω be divided into N subdomains. We refer Ω_i to ith subdomain and $\Omega_1, \Omega_2, ..., \Omega_N$ to the partition of the domain in the sense that $\bigcup_{i=1}^{N} \Omega_i = \Omega$, and $\Omega_i \subset \Omega$. Here, overlap is allowed. Therefore, if we let n_i be the dimension of subdomain Ω_i, then $\sum_{i=1}^{N} n_i \geq n$ where n is the dimension of Ω.

The restriction matrix R_i is defined as $R_i : R^n \to R^{n_i}$ which restricts the data from the global domain into subdomains. R_i is a $n_i \times n$ matrix filled with either 0 or 1 regarding to the partition. In the same manner, the interpolation matrix is defined as a transpose of R_i such that $R_i^T : R^{n_i} \to R^n$ which distributes the data from subdomains to whole domain. Here, the representation of restriction and interpolation matrices, R_i and R_i^T, should not be confused with the representation of the space R^n.

The additive Schwarz linear preconditioner matrix M^{-1} is formed by

$$M^{-1} = \sum_{i=1}^{N} R_i^T A_i^{-1} R_i \tag{2.14}$$

where A_i^{-1} is the subspace inverse of A_i. A_i is a $n_i \times n_i$ matrix that consists of some of the entries of the system matrix A considering the corresponding subdomain Ω_i. Note that, to form the linear preconditioner matrix, we approximate the inverse of the system matrix A by summing the inverses of the subdomain matrices in an additive Schwarz manner. Thus, we do not use the right hand side of the linear system $Au = b$.

2.6. Continuation Method

We have mention that, Newton's method may fail for some problems with unbalanced nonlinearities even with the globalization methods such as line search and trust region. In fluid dynamics problems, one of the most common source of the unbalanced nonlinearity is Reynolds number.

For the problems that Newton's method fails due to high Reynolds number, one approach to overcome this issue is continuation method. Starting with a low or moderate Reynolds number, continuation method solves a number of subproblems by incremental Reynolds number.

In this section, we briefly describe continuation method which is used in our study for two purposes; to validate our numerical results obtained by nonlinearly preconditioned methods and to ensure the convergence of the local nonlinear problems defined on subdomains in additive Schwarz preconditioned inexact Newton (ASPIN) method which is to be explained in forthcoming chapters.

Consider the nonlinear function $F : R^n \to R^n$, the nonlinear system $F(u) = 0$ where u is the unknown vector and the exact solution vector u^* such that $F(u^*) = 0$. The nonlinear system can be written as $F(u, \beta) = 0$ defined as $F : R^{n+1} \to R^n$ where β is some physical parameter such as Reynolds number or some computational parameter. The aim is to find the solution which approximates the exact solution u^* for some certain value β^*.

Let $u^{(0)}$ be the initial guess for solving $F(u) = 0$ by Newton's method and suppose that Newton's method can not converge to the desired solution u^* for β^*. Continuation method offers a solution procedure to find u^* for β^* by solving a number of problems $F(u, \beta) = 0$ sequentially with changing the parameter $\beta \in [\beta^{(1)}, \beta^*]$. That means, for a given initial guess $u^{(0)}$ and the first known or selected parameter value $\beta^{(1)}$, we solve the systems $F(u^{(k)}, \beta^{(k)}) = 0$ by Newton's method to find the intermediate solution $u^{(k)}$ for $k = 1, 2, ...$ where $\beta^{(k)}$'s are the intermediate parameters such that $\beta^{(k)} \in [\beta^{(1)}, \beta^*]$.

Each intermediate solution $u^{(k)}$ is used as a initial guess for the forthcoming problem to find the next intermediate solution $u^{(k+1)}$. This procedure is done until the desired solution u^* is found for the true parameter value β^*. For details regarding continuation method, we refer [21].

3. NONLINEAR PRECONDITIONER METHODS

Inexact Newton (IN) converges fast for the well conditioned nonlinear systems, for example a system obtained from the discretization of Navier-Stokes equations with low Reynolds number. However, after some Reynolds number, inexact Newton can not deal with the nonlinearity of the system. In literature, several methods are used to overcome this issue such as parameter continuation, pseudo-time stepping, mesh sequencing [1]. However, in this study, we focus on nonlinear preconditioners which apply direclty to the original nonlinear systems.

In the class of left nonlinear preconditioners, we construct a new nonlinear system which is easier to solve and has the same solution as the original nonlinear system $F : R^n \to R^n$, and solve the preconditioned system by using inexact Newton. The aim is to attack the nonlinearity of the original nonlinear system. We denote the preconditioned nonlinear function as $\mathcal{F}$ which is defined implicitly for most cases and $\mathcal{F}$ depends on both F and u. The explicit form can only be obtained for small problems as shown in the two simple examples in Chapter 4.

Here, in order to demonstrate the implicit representation of left nonlinear preconditioners, we denote G as a nonlinear preconditioner operator such that $G : R^n \to R^n$ to be applied to the original nonlinear function F to construct nonlinearly preconditioned function $\mathcal{F}$ as given below.

$$\mathcal{F}(u) = G(F(u)) \tag{3.1}$$

Following [1], we show the desirable properties of G:

(i) If $G(u) = 0$, then $u = 0$.
(ii) In some sense $G \approx F^{-1}$.
(iii) $G(F(v))$ is easily computable for any vector $v \in R^n$.

(iv) Calculation of Jacobian-vector product $(G(F(v)))'w$ is easily computable if Krylov subspace method is used.

On the other hand, the class of right nonlinear precoditioners aims to calculate a better approximate solution $u^{(k)}$ to participate in the following Newton iteration in order to enhance the convergence of Newton's method. Thus, the nonlinear system remains the same as original nonlinear system. Calculating a better approximate solution $u^{(k)}$ determines the type of the right nonlinear preconditioner. In this study, we focus on nonlinear elimination preconditioner that eliminates the previously chosen group of unknowns in a sub-nonlinear problem to form a better $u^{(k)}$.

This thesis includes two left nonlinear preconditioning methods and one right nonlinear preconditioning method, namely additive Schwarz preconditioned inexact Newton (ASPIN) method , field-split preconditioned inexact Newton (FSPIN) method and nonlinear elimination preconditioned inexact Newton (NEPIN) method, respectively.

3.1. Additive Schwarz Preconditioned Inexact Newton Method

Additive Schwarz nonlinear preconditioner is based on domain decomposition methods in which the solution of subdomains are needed to construct a preconditioned function evaluation which composes more nonlinearly balanced system. Solution of local nonlinear problems on the subdomains can be done simultaneously in parallel since the subdomain problems are independent in additive Schwarz method. After evaluating the nonlinear preconditioned function, the search direction is found by solving preconditioned Jacobian system.

The organization of this section is as follows. First, the framework of nonlinear additive Schwarz preconditioner is given. After defining the Jacobian of the preconditioned system, the construction of the preconditioned Jacobian and the solution procedure of linear Jacobian systems within nonlinear iterations are explained. Lastly, the algorithm of ASPIN and domain partitioning strategy are described.

3.1.1. Nonlinear Additive Schwarz Preconditioner

In this subsection, we review the framework of additive Schwarz nonlinear preconditioner which is based on domain decomposition methods. Consider the domain Ω and let Ω be divided into N subdomains. We refer Ω_i to ith subdomain and $\Omega_1, \Omega_2, ..., \Omega_N$ to the partition of the domain in the sense that

$$\bigcup_{i=1}^{N} \Omega_i = \Omega, \text{ and } \Omega_i \subset \Omega. \tag{3.2}$$

Here, overlap is allowed. Therefore, if we let n_i be the dimension of subdomain Ω_i, then

$$\sum_{i=1}^{N} n_i \geq n \tag{3.3}$$

where n is the dimension of Ω. We introduce the subspaces of R^n. For a given subdomain Ω_i, we define a vector V_i whose entries which correspond to the concerning subdomain are nonzero ,but the rest are zero.

$$V_i = \{v | v = (v_1, ..., v_n)^T \in R^n, v_k = 0, \text{if } k \notin \Omega_i\} \tag{3.4}$$

V_i can be demonstrated as

$$V_i = R_i V \text{ such that } V_i \subset R^n \tag{3.5}$$

where R_i is the restriction matrix whose kth column is the kth column of the identity matrix if the index k corresponds to the concerning subdomain Ω_i, otherwise zero. We should note that, the representation of restriction and interpolation matrices, R_i and R_i^T, should not be confused with the representation of the n-dimensional space R^n.

Subdomain nonlinear function (also called local nonlinear function) $F_i : R^n \to R^n$ is defined by using restriction matrix as

$$F_i = R_i F. \tag{3.6}$$

Consider a vector valued function $T_i(v)$ for any vector v such that $v \in R^n$ defined as a solution of the following local nonlinear system.

$$F_i(v - T_i(v)) = 0 \tag{3.7}$$

We introduce the nonlinearly preconditioned function $\mathcal{F}$ evaluated at the current approximate solution $u^{(k)}$ as a collection of the solution of the above local nonlinear systems as shown below.

$$\mathcal{F}(u^{(k)}) = \sum_{i=1}^{N} T_i(u^{(k)}) \tag{3.8}$$

Evaluation of the nonlinearly preconditioned function for a given vector involves a number of solutions of those local nonlinear systems on corresponding subdomains.

For a given initial guess $u^{(0)}$, additive Schwarz preconditioned inexact Newton (ASPIN) finds an approximation to the exact solution u^* of the nonlinear system $F(u) = 0$ by solving the nonlinearly preconditioned system

$$\mathcal{F}(u) = 0 \tag{3.9}$$

by using inexact Newton.

We note that, it is proven by [22] that the local nonlinear systems shown in Equation 3.7 are uniquely solvable. Furthermore, it is also proven by [1] that the preconditioned system has the same solution as the original system.

3.1.2. Computing the Jacobian of the Preconditioned System

The solution of nonlinearly preconditioned system shown in Equation 3.9 requires Jacobian calculation since it is solved by inexact Newton. The computation of the Jacobian of the original function F is straightforward because the function is defined explicitly. However, the nonlinearly preconditioned function $\mathcal{F}$ is only defined implicitly in the most cases. Therefore, calculation of the Jacobian of the nonlinearly preconditioned function is a bit toilsome and explained in this subsection.

Consider the Jacobian of the original function

$$J = F' = \left(\frac{\partial F_i}{\partial u_j}\right)_{n\times n} \tag{3.10}$$

and, let J_i be the Jacobian of the local (or subdomain) nonlinear system defined on Ω_i

$$J_i = \left(R_i J R_i^T\right) \tag{3.11}$$

for $i = 1, 2, ..., N$. The Jacobian of the preconditioned system and the Jacobian of the preconditioned local nonlinear systems are denoted respectively as

$$\mathcal{J} = \mathcal{F}' = \left(\sum_{i=1}^{N} \mathcal{J}_i\right)_{n\times n} \tag{3.12}$$

and

$$\mathcal{J}_i = \left(\frac{\partial T_i}{\partial u_j}\right)_{n\times n}. \tag{3.13}$$

To derive the Jacobian of the preconditioned nonlinear system, consider a subdomain Ω_i and the complementary of that subdomain Ω_i^c. Suppose that we want to compute the Jacobian of the preconditioned nonlinear system at a point $u \in R^n$ and u can be represented as $u = (u_i, u_i^c)$. Equation 3.7 can be rewritten as

$$F_i(u_i - T_i(u_i, u_i^c), u_i^c) = 0. \tag{3.14}$$

We take the derivative of Equation 3.14 with respect to u_i and obtain

$$\left(\frac{\partial F_i}{\partial v_i}\right)\left(I_i - \frac{\partial T_i(u)}{\partial u_i}\right) = 0 \tag{3.15}$$

where $v_i = u_i - T_i(u_i, u_i^c)$ and I_i is identity matrix of the corresponding to ith subdomain. Assuming that $\frac{\partial F_i}{\partial v_i}$ is not singular, we can rewrite Equation 3.15 as

$$\frac{\partial T_i(u)}{\partial u_i} = I_i = \left(\frac{\partial F_i}{\partial v_i}\right)^{-1} \frac{\partial F_i}{\partial v_i}. \tag{3.16}$$

Then, we take the derivative of Equation 3.14 with respect to u_i^c and we get

$$-\frac{\partial F_i}{\partial v_i}\frac{\partial T_i(u)}{\partial u_i^c} + \frac{\partial F_i}{\partial u_i^c} = 0. \tag{3.17}$$

Leaving $\frac{\partial T_i(u)}{\partial u_i^c}$ on the left hand side, we have

$$\frac{\partial T_i(u)}{\partial u_i^c} = \left(\frac{\partial F_i}{\partial v_i}\right)^{-1} \frac{\partial F_i}{\partial u_i^c}. \tag{3.18}$$

Note that

$$\frac{\partial T_i(u)}{\partial u} = \frac{\partial T_i(u)}{\partial u_i} + \frac{\partial T_i(u)}{\partial u_i^c} \tag{3.19}$$

since Ω_i and Ω_i^c are not overlapped, in other words $\Omega_i \cup \Omega_i^c = \Omega$ and $\Omega_i \cap \Omega_i^c = \emptyset$.

Substituting Equations 3.16 and 3.18 into Equation 3.19, we obtain

$$\frac{\partial T_i(u)}{\partial u} = \left(\frac{\partial F_i}{\partial v_i}\right)^{-1} \frac{\partial F_i}{\partial z_i} \tag{3.20}$$

where $z_i = (v_i, u_i^c)$. We get a formula for the Jacobian of the nonlinearly preconditioned system by summing up Equation 3.20 for all subdomains and evaluating u at $u^{(k)}$

$$\mathcal{J} = \sum_{i=1}^{N} \left(J_i(z_i)\right)^{-1} J(z_i) \mid_{z_i=(u_i^{(k)} - T_i(u^{(k)}),(u_i^c)^{(k)})} \tag{3.21}$$

It is stated by [1] that, although the calculation of the above formula is possible, it is more convenient and cheaper to use its approximate form in computational implementation. For the current approximate solution $u^{(k)}$, we use the following formula.

$$\mathcal{J} = \sum_{i=1}^{N} \left(J_i(z_i)\right)^{-1} J(z_i) \mid_{z_i=u^{(k)}} \tag{3.22}$$

It is interesting that, the Jacobian of the preconditioned function shown in Equation 3.22 corresponds to a linear additive Schwarz preconditioned Jacobian. We see that in our numerical experiments, the Jacobian of the preconditioned function is well-conditioned and does not need any linear preconditioning.

Although the Jacobian of the original function and the local Jacobians are generally sparse, the Jacobian of the preconditioned function is often dense and expensive to compute. Since we use a Krylov subspace method [5] to solve the linear Jacobian system, we only need to calculate a matrix-vector product y formulated as

$$y = \sum_{i=1}^{N} \left(J_i(u^{(k)})\right)^{-1} J(u^{(k)})\, v \tag{3.23}$$

for any $v \in R^n$. We deal this computation through the following steps.

(i) Compute the matrix-vector product $w = Jv$ by finite difference approximation where $w = (w_1, w_2, ..., w_N)^T$

(ii) Find y_i as a solution of $J_i\ y_i = w_i$ for $i = 1, 2, .., N$ by direct solver

(iii) Form the result $y = (y_1, y_2, ..., y_N)^T$

We note that, it is proven by [1] that the Jacobian of the preconditioned system $\mathcal{J}$ is not singular for any u which is in the neighborhood of u^* and the Jacobian of the preconditioned system is well conditioned as long as the number of subdomains is not large.

3.1.3. Additive Schwarz Preconditioned Inexact Newton Algorithm

In this subsection, we describe additive Schwarz preconditioned inexact Newton algorithm and discuss the details of the algorithm. Let $u^{(0)}$ be the initial guess and $u^{(k)}$ be the current approximate solution of the nonlinear system $F(u) = 0$. For a given current approximate solution $u^{(k)}$, ASPIN algorithm computes the new approximate solution $u^{(k+1)}$ by following the steps in the algorithm demonstrated in Figure 3.1.

In Step 1 a) of ASPIN algorithm, we solve the local subdomain systems with Newton's method simultaneously. Each local nonlinear system has n_i unknowns and n_i equations to be solved. The linear Jacobian systems in the local nonlinear systems are solved directly. Line search is employed to ensure the convergence of the local nonlinear systems. Note that, solution of the local nonlinear systems $F_i(u^{(k)} - g_i^{(k,j)}) = 0$ can be written as $F_i(u_i^{(k)} - g_i^{(k,j)}, (u_i^c)^{(k)}) = 0$. This implies that $(u_i^c)^{(k)}$ is required to solve local nonlinear systems. However, the values of $(u_i^c)^{(k)}$ do not change during the solution of the local nonlinear systems. In fact, they act as a boundary condition for the local nonlinear systems.

Step 1: Compute the nonlinear residual $\mathcal{F}$ at $u^{(k)}$, $\mathcal{F}(u^{(k)}) = g^{(k)}$, by following steps:

a) Solve the following local nonlinear systems to get $g_i^{(k,j)}$ for $i = 1, 2, ..., N$

$$F_i(u^{(k)} - g_i^{(k,j)}) = 0 \tag{3.24}$$

with the initial guess $g_i^{(k,0)} = 0$ by using Newton's method.

b) Form the global residual

$$g^{(k,j)} = \sum_{i=1}^{N} g_i^{(k,j)} \tag{3.25}$$

c) Check the stopping condition on $g^{(k,j)}$. If satisfied, set $g^{(k)} = g^{(k,j)}$ and go to Step 2.

Step 2: Find the inexact Newton direction $p^{(k)}$ by solving the Jacobian system approximately

$$\sum_{i=1}^{N} J_i^{-1} J p^{(k)} = g^{(k)} \tag{3.26}$$

such that

$$\|g^{(k)} - \sum_{i=1}^{N} J_i^{-1} J p^{(k)}\| \leq \eta_k \|g^{(k)}\| \tag{3.27}$$

for some $\eta_k \in [0, 1)$.

Step 3: Compute the new approximate solution

$$u^{(k+1)} = u^{(k)} - \lambda^{(k)} p^{(k)} \tag{3.28}$$

where $\lambda^{(k)}$ is a damping parameter.

Figure 3.1. Additive Schwarz preconditioned inexact Newton algorithm.

In some cases where the Reynolds number is high, some local nonlinear systems may become hard to be solved by Newton's method at the first iterations of the outer ASPIN iterations. To overcome this, we employ parameter continuation technique [21] with respect to Reynolds number for those subdomain nonlinear systems if Newton's method fails for that local nonlinear system.

After forming the residual of the preconditioned system $\mathcal{F}(u^{(k,j)})$, we check the stopping condition on it. Step 1 a) and b) are done until the stopping condition shown below is satisfied.

$$abs\left(||\mathcal{F}(u^{(k,j)})|| - ||\mathcal{F}(u^{(k,j-1)})||\right) \leqslant 10^{-6} \tag{3.29}$$

Note that, $||\mathcal{F}(u^{(k)})||$ is related to $||F(u^{(k)})||$ in some sense since both systems have the same solution.

In Step 2, we solve global preconditioned Jacobian system to calculate the search direction $p^{(k)}$. Matrix-free Generalized Minimal Residual (GMRES) Method is used with zero initial guess to solve the global linear systems. η_k is the forcing term that acts as a relative tolerance on the solution of the corresponding linear system.

The next approximate solution $u^{(k+1)}$ is computed in Step 3. To ensure the decrease in the residual norm of the preconditioned function, we check the condition

$$f(u^{(k+1)}) \leqslant f(u^{(k)}) - a\lambda^{(k)}s^{(k)} \tag{3.30}$$

where $a = 10^{-4}$ and $s^{(k)}$ is the initial slope [14]. Here f is the merit function defined as

$$f(u) = \frac{1}{2}||\mathcal{F}(u)||^2. \tag{3.31}$$

Starting with the first value $\lambda^{(k)} = 1$, we calculate $\lambda^{(k)}$ values by cubic backtracking line search technique [14] until the condition shown in Equation 3.30 is satisfied.

Note that, calculation of $f(u^{(k+1)})$ in Equation 3.31 involves the calculation of the residual of the preconditioned function at the intermediate solution $u^{(k+1)}$. By doing this, one can store the residual of the preconditioned system $\mathcal{F}(u^{(k+1)})$ that satisfies the condition shown in Equation 3.30 and skip Step 1 of the algorithm in the forthcoming ASPIN iteration.

3.1.4. Partitioning Strategy

Partitioning is one of the key components of ASPIN since the convergence of the method depends on partitioning and a good partitioning is generally problem specific. For the most part, partitioning has three characteristics; pattern, number of subdomains and overlap. Some strategies have been used in partitioning pattern such as mesh point based, element based and physics based in domain decomposition studies. We employ mesh point base partitioning style in our study since it is relevant with finite difference approximation of the partial differential equations.

Regular quadrilaterals is the most used partitioning pattern in the ASPIN studies since they are generally done on square domain. So it is analogous to divide domain into small quadrilaterals. Figure 3.2 demonstrates the regular quadrilateral partitioning pattern over a square domain.

Here, global domain consists of nine nodes (displayed by 'o') in each direction, in total 9×9 nodes. The dashed lines indicate 3×3 nonoverlapping partitioning whereas the solid lines indicate subdomains with overlap size $h_o = 1$.

Notice that, by increasing the overlap size h_o by one, we include one more line of mesh points to the subdomains if the mesh points to be included is inside of the domain. Also notice that, the numbers of mesh points in the subdomains are equal when $h_o = 0$. However, they vary if the subdomains are overlapping.

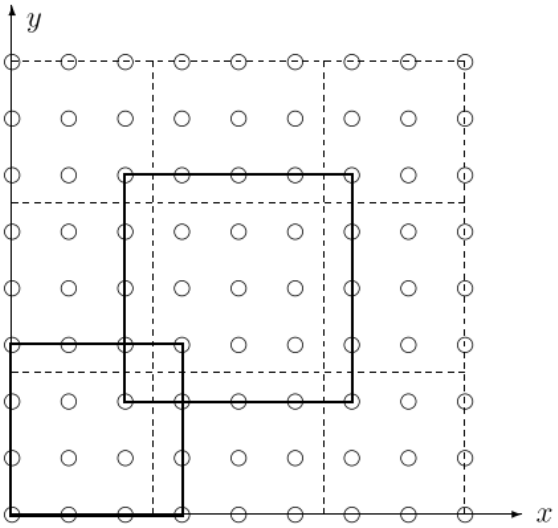

Figure 3.2. Regular quadrilateral partitioning pattern [1].

In this study, various overlap sizes and partition numbers both in x and y direction for regular quadrilateral partitioning are tested. Many other partitioning patterns and strategies can be developed and tested.

3.2. Field-Split Preconditioned Inexact Newton

Field-split preconditioned inexact Newton (FSPIN) method can be regarded as a variation of ASPIN, based on field-splitting decomposition in the sense that the physical unknown variables of the problem are grouped to create a number of nonintersecting subsets. The sub-nonlinear problems regarding the subsets are solved independently to create nonlinear preconditioned function evaluation. After the preconditioned Jacobian system is solved to obtain search direction, the next approximate solution is calculated in a very similar way as ASPIN.

We describe the framework of field-split preconditioner and give the formula of the Jacobian of the preconditioned nonlinear function. Since it is very similar to ASPIN, we give the algorithm by not describing it in detail.

3.2.1. Field-Split Nonlinear Preconditioner

Consider root finding problem of nonlinear function $F(u) = 0$. We split the original nonlinear function into two subsets to demonstrate the framework of field-split nonlinear preconditioner as

$$F(u) = F(u_1, u_2) = \begin{pmatrix} F_1(u_1, u_2) \\ F_1(u_1, u_2) \end{pmatrix} \tag{3.32}$$

where the unknown vector is also split as $u = (u_1, u_2)^T$. Field-split preconditioned inexact Newton method solves the following nonlinearly preconditioned function .

$$\mathcal{F}(u) = \mathcal{F}(u_1, u_2) = \begin{pmatrix} g_1(u_1, u_2) \\ g_2(u_1, u_2) \end{pmatrix} \tag{3.33}$$

Here, $g_1(u_1, u_2)$ and $g_2(u_1, u_2)$ are defined as the solutions of following subproblems

$$\begin{aligned} F_1(u_1 - g_1, u_2) &= 0 \\ F_2(u_1, u_2 - g_2) &= 0 \end{aligned} \tag{3.34}$$

The Jacobian of the field-split preconditioned system can be calculated in the same way as in non-overlapping two subdomain partitioned additive Schwarz preconditioner. Since the derivation of the preconditioned Jacobian is similar to the one we describe in additive Schwarz nonlinear preconditioner, here we only give the formula of its appropriate approximation by referring [23].

$$\mathcal{J} = \sum_{i=1}^{2} \left(J_i(u_1, u_2)\right)^{-1} J(u_1, u_2) \tag{3.35}$$

Since we use a Krylov subspace method [5] to solve the linear Jacobian system, we only need to calculate a matrix-vector product for any vector v such that $v \in R^n$. We deal this computation through the following steps.

(i) Compute the matrix-vector product $w = Jv$ by finite difference approximation where $w = (w_1, w_2)^T$

(ii) Solve $J_1 y_1 = w_1$ and $J_2 y_2 = w_2$ by using direct solver

(iii) Form the result $y = (y_1, y_2)^T$

3.2.2. Field-Split Preconditioned Inexact Newton Algorithm

Let the initial guess be $u^{(0)} = (u_1^{(0)}, u_2^{(0)})$. At the kth iteration, field-split preconditioned inexact Newton method computes the next approximate solution $u^{(k+1)}$ using the previously calculated current approximate solution $u^{(k)}$ by following the steps in the algorithm in Figure 3.3.

Although ASPIN and FSPIN are very similar to each other, there is a conceptual difference between them so that they are different methods. ASPIN is domain decomposition based method which includes the decomposition of the domain in a number of subdomains. However, in FSPIN, we decompose the field variables of the differential equation into a number of groups. In other words, the difference is determination of the local problems and this changes the solution procedure significantly.

3.3. Nonlinear Elimination Preconditioned Inexact Newton

We have mentioned that, the source of the nonlinear imbalance of the nonlinear systems is the dominance of some subset of unknown variables on some area in the domain. Nonlinear elimination is a routine that solves some sub-problems for corresponding subset of the unknowns. In nonlinear elimination preconditioned inexact Newton (NEPIN), we aim to obtain a better approximate solution $u^{(k)}$ to participate in the forthcoming Newton iteration, by employing nonlinear elimination routine. This procedure is done if the current $u^{(k)}$ results in a nonlinear residual higher than some threshold value. This means that, nonlinear elimination preconditioner is employed when it is needed. If the nonlinear residual evaluated at $u^{(k)}$ is not problematic, Newton method applies to the system directly. Application of elimination routine affects only the approximate solution $u^{(k)}$, not the nonlinear function F.

Step 1: Compute the nonlinear residual $\mathcal{F}$ at $u^{(k)}$, $\mathcal{F}(u^{(k)}) = g^{(k)}$ by following steps:

a) Solve the following local nonlinear systems for $g_1^{(k,j)}$ and $g_2^{(k,j)}$

$$\begin{aligned} F_1(u_1^{(k)} - g_1^{(k,j)}, u_2^{(k)}) = 0 \\ F_2(u_1^{(k)}, u_2^{(k)} - g_2^{(k,j)}) = 0 \end{aligned} \tag{3.36}$$

with the initial guess $g_i^{(k,0)} = 0$ by using Newton's method.

b) Form the global residual

$$g^{(k,j)} = \begin{bmatrix} g_1^{(k,j)} \\ g_2^{(k,j)} \end{bmatrix} \tag{3.37}$$

c) Check the stopping condition on $g^{(k,j)}$. If satisfied, set $g^{(k)} = g^{(k,j)}$ and go to Step 2.

Step 2: Find the inexact Newton direction $p^{(k)}$ by solving the Jacobian system approximately

$$\sum_{i=1}^{2} J_i^{-1} J p^{(k)} = g^{(k)} \tag{3.38}$$

such that

$$\|g^{(k)} - \sum_{i=1}^{2} J_i^{-1} J p^{(k)}\| \leq \eta_k \|g^{(k)}\| \tag{3.39}$$

for some $\eta_k \in [0, 1)$.

Step 3: Compute the new approximate solution

$$u^{(k+1)} = u^{(k)} - \lambda^{(k)} p^{(k)} \tag{3.40}$$

where $\lambda^{(k)}$ is a damping parameter.

Figure 3.3. Field-split preconditioned inexact Newton algorithm.

In this section, the framework of nonlinear elimination routine applied to inexact Newton is given. Then, the algorithm of NEPIN is explained in detail.

3.3.1. Framework of Nonlinear Elimination Preconditioner

Consider the nonlinear system $F(u) = 0$ arising from the discretization of partial differential equations where $F : R^n \rightarrow R^n$ where u is the unknown solution vector. We introduce nonlinear elimination preconditioner G where nonlinearly preconditioned system is in the following form.

$$F(G(u)) = 0 \tag{3.41}$$

$G(u)$ acts as a nonlinear preconditioner and for the most cases, it is only defined implicitly. Notice that the original nonlinear function F remains the same whereas the unknown solution vector is changed which is an input of the original function F.

Let $S = (1, 2, ..., n)$ be the index set of the system meaning that each index stands for an unknown component u_i and a nonlinear function component F_i. Assume that S_1 is the collection of indices with c number of components that needs to be eliminated such that $S_1 \cup S_2 = S$ and $S_1 \cap S_2 = \varnothing$ where the rest of the indices are grouped in S_2. For a given u which is divided as $u = (u_1, u_2)^T$ such that $u_1 \in S_1$ and $u_2 \in S_2$, we define $T(u) : R^n \rightarrow R^c$ as a solution of the nonlinear subsystem

$$F_1(\ T(u),\ u_2\ ) = 0 \tag{3.42}$$

where $F_1 : R^n \rightarrow R^c$ is the sub-nonlinear function defined in S_1. After the elimination, we form the new approximate solution $u_e = (\ T(u),\ u_2\ )$ whose problematic components hopefully becomes unproblematic. Subscript e stands to distinguish the newly eliminated approximate solution.

Note that, application of nonlinear elimination preconditioner requires solving some number of linear or nonlinear sub-problems defined in the corresponding subspace.

Determination of the components to be eliminated is a great challenge and mostly problem dependent. Determination routine can be done at the beginning of each outer Newton iterations adaptively, see [2].

3.3.2. Nonlinear Elimination Preconditioned Inexact Newton Algorithm

For a given current approximate solution $u^{(k)}$, nonlinear elimination preconditioned inexact Newton (NEPIN) algorithm computes the next approximate solution $u^{(k+1)}$ by following the algorithm given in Figure 3.4.

Step 1: Check the nonlinear residual evaluated at $u^{(k)}$ and apply nonlinear elimination preconditioner if needed.

if $||F(u^{(k)})|| > \epsilon_{switch}$ **then**

Compute $y = G(u^{(k)})$ by solving the system shown in Equation 3.42 by IN and let $u^{(k)} = y$

end if

Step 2: Find the inexact Newton direction $p^{(k)}$ by solving the Jacobian system approximately

$$M^{-1}\ J(u^{(k)})\ p^{(k)} = -M^{-1}\ F(u^{(k)}) \tag{3.43}$$

with a linear preconditioner M^{-1}.

Step 3: Compute the new approximate solution

$$u^{(k+1)} = u^{(k)} + \lambda^{(k)} p^{(k)} \tag{3.44}$$

where $\lambda^{(k)}$ is a damping parameter.

Figure 3.4. Nonlinear elimination preconditioned inexact Newton algorithm.

Nonlinear elimination preconditioner algorithm requires partitioning information of S which means that user needs to provide a routine that determines the subspace of R^n to be eliminated. By pre-knowledge of the solution, partition to be eliminated

can be a portion of domain or a physical component of the multi-component partial differential equations. This type of partitioning is static partitioning meaning that at each global nonlinear iteration same partitioning is used. For example, Cai and Li [11] solved the subspace nonlinear systems on the whole domain in a restricted additive Schwarz manner. On the other hand, one can design a routine that determinates the partition to be eliminated at the beginning of each global Newton iteration. As an example to this adaptive routine, the algorithm of Yang and Hwang [2] determines the partition based on the nonlinear residual of each components of $F(u)$.

In Step 1, the parameter ϵ_{switch} is a threshold value for activation of nonlinear elimination preconditioner. If the nonlinear residual norm evaluated at the current solution $u^{(k)}$ is less than ϵ_{switch}, algorithm does not employ elimination routine. This exists to prevent unnecessary elimination routine where Newton is expected to work well itself for the current solution $u^{(k)}$.

On the contrary of right nonlinear preconditioner methods, the Jacobian system within the NEPIN algorithm is generally not well-conditioned since the nonlinear function is unchanged. This brings the need of a linear preconditioner and user has the freedom of choosing the linear preconditioner of any kind. In addition, solution of the Jacobian system within the subspace nonlinear iterations may also need linear preconditioner if iterative solver is used, but in general, the size of the subspace systems are small and they have smaller condition than global ones. Moreover, The sub system or subsystems may be linear depending on the choice of components or variables to be eliminated.

4. TEST RESULTS AND DISCUSSION

In this chapter, we present the results of our tests and explain them in detail. In the first test, we consider two simple nonlinearly unbalanced problem to demonstrate the effects of right and left nonlinear preconditioners on the problem and the nonlinear iterations. In the second test, we solve steady-state incompressible lid-driven cavity problem on a square domain in velocity-vorticity formulation of the Navier-Stokes equations. The nonlinear set of equations is obtained by two different discretization model which are described in Subsection 4.2.1. We compare the algorithms described in Chapter 2 and 3 for various Reynolds numbers. Many parametric studies have also been conducted so that the behaviors of the algorithms on certain parameters are understood well.

4.1. Test 1: Two Simple Problems with Unbalanced Nonlinearity

We solve two simple examples and their nonlinearly preconditioned form by exact Newton to show the effects of nonlinear preconditioner on nonlinear systems. First, consider the nonlinear function $F(u)$, $F : R^2 \to R^2$ with two variables

$$F(u) = \begin{pmatrix} (u_1 - u_2^3 + 1)^m - u_2^m \\ 3u_1 + 2u_2 - 5 \end{pmatrix} \tag{4.1}$$

for the parameter $m = 1, 3, 5$ which changes the nonlinearity of the function where the unknown vector is $u = (u_1, u_2)^T$. The root of this nonlinear system is $u^* = (1, 1)^T$. One can see that, the first equation is more nonlinear than the second equation and for large m, the nonlinear system becomes more nonlinearly unbalanced.

In nonlinear elimination preconditioner as a right nonlinear preconditioner, the original nonlinear system shown in Equation 4.1 remains the same as discussed in Chapter 3. We choose the component to be eliminated before Newton iteration. Let the component to be eliminated be u_1 since the first equation is more nonlinearly unbalanced. For the current approximate solution $u^{(k)}$, nonlinear elimination routine

solves the first equation of Equation 4.1 for u_1 at the beginning of each Newton iteration as demonstrated below.

$$u_1 - u_2^{(k)} - (u_2^{(k)})^3 + 1 = 0 \tag{4.2}$$

The intermediate approximate solution is formed by using the newly calculated u_1, to participate in the forthcoming Newton iteration.

On the other hand, to derive the left nonlinearly preconditioned system in an additive Schwarz nonlinear preconditioner manner, we introduce left preconditioned system and function $\mathcal{F}(u)$, $\mathcal{F} : R^2 \to R^2$. The components of the left preconditioned system $g_1(u_1, u_2)$, $g_2(u_1, u_2)$ which are defined as solutions of $F_1(u_1 - g_1, u_2) = 0$ and $F_2(u_1, u_2 - g_2) = 0$ respectively. Here, the partitions are simply the components of the nonlinear system, namely u_1 and u_2. In this simple case with two variable, the explicit form of the preconditioned function can be derived and it is shown below. Unfortunately, explicit form is usually unavailable for large problems.

$$\mathcal{F}(u) = \begin{pmatrix} g_1(u_1, u_2) \\ g_2(u_1, u_2) \end{pmatrix} = \begin{pmatrix} u_1 - u_2 - u_2^3 + 1 \\ \frac{3}{2}u_1 + u_2 - \frac{5}{2} \end{pmatrix} \tag{4.3}$$

Notice that, $u^* = (1, 1)^T$ is also the root of the preconditioned function. However, exponential m does not appear in the preconditioned function, in other words the preconditioned function is the same for different m values. This implies that m can not change the nonlinearity of the preconditioned function shown in Equation 4.3.

We solve the original problem shown in Equation 4.1 by three method, namely exact Newton, nonlinear elimination preconditioned exact Newton (NEPEN), additive Schwarz preconditioned exact Newton (ASPEN). Notice that the methods become exact Newton variant since we solve the Jacobian systems within the Newton iterations exactly for this very small problem. We include nonlinear elimination routine that solves Equation 4.2 for NEPEN. For ASPEN, we derive the left preconditioned nonlinear system shown in Equation 4.3 and solve it by exact Newton. Four different initial

guesses $u^{(0)} = (0,0)^T, (0,2)^T, (2,0)^T, (2,2)^T$ are used. The Jacobians are calculated by using forward finite differences with step size $\epsilon = 10^{-7}$. Original nonlinear system is solved for different m values. Table 4.1 shows the number of nonlinear iterations of exact Newton for all methods. We stop the nonlinear iterations if $||F(u^{(k)})|| < 10^{-6}$ for Newton and NEPEN, $||\mathcal{F}(u^{(k)})|| < 10^{-6}$ for ASPEN at the kth iteration.

Table 4.1. The numbers of exact Newton iterations for the first simple example with unbalanced nonlinearity.

	$F(u)$						$\mathcal{F}(u)$
	Newton			NEPEN			ASPEN
$u^{(0)}$	$m=1$	$m=3$	$m=5$	$m=1$	$m=3$	$m=5$	
$(0,0)^T$	5	15	20	5	5	5	5
$(0,2)^T$	5	9	13	5	5	5	5
$(2,0)^T$	5	1	7	5	5	5	5
$(2,2)^T$	5	10	13	5	5	5	5

We see that in Table4.1, NEPEN and ASPEN generally require less iterations than Newton especially for high m value. Moreover, NEPEN and ASPEN are more robust since they are not affected by changing the initial guess whereas changing the initial guess affects the number of Newton iterations of the original system drastically. It is seen that the number of NEPEN iterations and ASPEN iterations are the same and the number of NEPEN iterations remain same for all m values. We explain this situation in forthcoming paragraphs after the figures.

The contour plots of the merit functions $f(u) = log_{10}(10||F(u)||_2^2 + 1)$ for the original function $F(u)$ and $f(u) = log_{10}(10||\mathcal{F}(u)||_2^2 + 1)$ for the left preconditioned function $\mathcal{F}(u)$ are demonstrated in Figure 4.1, Figure 4.2 and Figure 4.3, respectively. Here we use logarithmic function to respond to the skewness of the residual of the function. l^2 norm is used to calculate the residual norms and multiplier by ten ensures the visibility of the contour especially near the roots. To prevent the value of the merit function from approaching negative infinity, inside of the function is added by one.

The figures also show the solution paths starting from the initial guess $u^{(0)} = (2, 2)^T$ through the root of the system for Newton, NEPEN and ASPEN, respectively. The blue circles denote the intermediate approximate solution points throughout the nonlinear iterations. The results on the figures are obtained for $m = 5$.

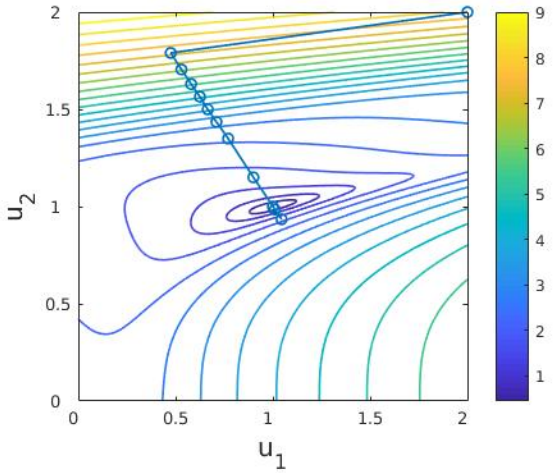

Figure 4.1. Contour plot of $log_{10}(10||F(u)||_2^2 + 1)$ and the solution path of Newton iterations for the first simple example with unbalanced nonlinearity.

It is obvious that, contours of the preconditioned system $\mathcal{F}(u)$ are more elliptical and regular. On the other side, contours of the original system $F(u)$ indicates the nonlinear imbalance of the system since the contours are irregular. In addition to this, the colors of the contours shows that, in the neighborhood of the root, the residuals of the original function are larger than the residuals of the preconditioned system.

Notice that, the numbers of iterations for NEPEN and ASPEN are equal. Moreover, Figure 4.2 and Figure 4.3 show that the intermediate approximate solutions for NEPEN and ASPEN are similar even though the nonlinear systems are different. We interpret this situation as follows: The nonlinear elimination equation shown in Equation 4.2 is the same as the first equation of the left preconditioned system shown in Equation 4.3 by the nature of problem with two variable. In addition, the second equation of the left preconditioned system shown in Equation 4.3 is the same as the second equation of the original system shown in Equation 4.1 by the nature of the

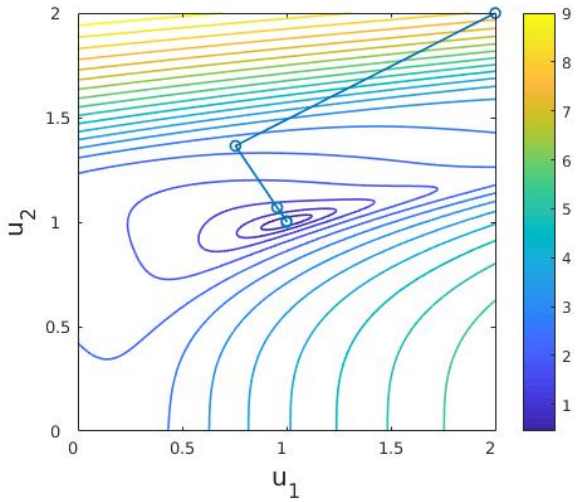

Figure 4.2. Contour plot of $log_{10}(10||F(u)||_2^2 + 1)$ and the solution path of NEPEN iterations for the first simple example with unbalanced nonlinearity.

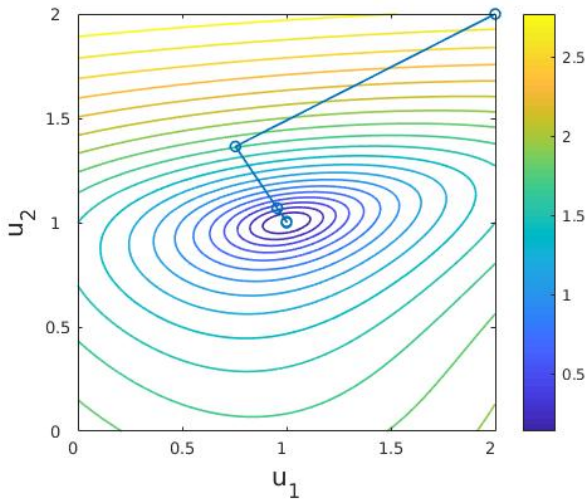

Figure 4.3. Contour plot of $log_{10}(10||\mathcal{F}(u)||_2^2 + 1)$ and the solution path of ASPEN iterations for the first simple example with unbalanced nonlinearity.

equation itself. Therefore, the application of NEPEN and ASPEN affects the solution procedure in a very same manner such that they yield almost the same results. We note that, this is a rare coincidence that could happen in very small problems such as this problem with two unknowns.

We test the second example to avoid the situation discussed above by changing the second equation in the nonlinear system. Consider the nonlinear system and function $F(u)$, $F : R^2 \to R^2$ with two variables

$$F(u) = \begin{pmatrix} (u_1 - u_2^3 + 1)^m - u_2^m \\ 4u_1^2 - u_2^2 - 8u_1 + 4 \end{pmatrix} \tag{4.4}$$

for the parameter $m = 1, 3, 5$. $u_1^* \cong (1.56408, 1.12817)^T$ and $u_2^* \cong (0.56019, 0.87961)^T$ are the two roots of the system. In NEPEN, the first equation of the original function shown in Equation 4.4 is solved to eliminate u_1 variable as in the first example. In ASPEN, the left nonlinearity preconditioned system of the original system is derived as the same manner as in the first example and demonstrated below.

$$\mathcal{F}(u) = \begin{pmatrix} g_1(u_1, u_2) \\ g_2(u_1, u_2) \end{pmatrix} = \begin{pmatrix} u_1 - u_2 - u_2^3 + 1 \\ u_2 - 2u_1 + 2 \end{pmatrix} \tag{4.5}$$

The first root of the original system is also the root of the nonlinearly preconditioned system which is $u^* \cong (1.56408, 1.12817)^T$. However, the second root $u_2^* \cong (0.56019, 0.87961)^T$ is not the root of the preconditioned system (4.5) because we lose one root while taking square root of an intermediate equality at the routine where we apply additive Schwarz preconditioner. The same methods and parameters are used with the first example and Table 4.2 shows the nonlinear iteration numbers for each method. The cases where the number of iterations are written with underlines converged to the second root of the system. The rest of the cases converged to the first root of the system.

NEPEN and ASPEN converge well for all m values and initial guesses with a small number of nonlinear iteration. In the case with $m = 1$, Newton converges faster

Table 4.2. The numbers of exact Newton iterations for the second simple example with unbalanced nonlinearity.

	$F(u)$						$\mathcal{F}(u)$
	Newton			NEPEN			ASPEN
$u^{(0)}$	$m=1$	$m=3$	$m=5$	$m=1$	$m=3$	$m=5$	
$(0,0)^T$	5	-	-	6	6	6	8
$(0,2)^T$	5	10	14	8	8	8	5
$(2,0)^T$	6	-	-	6	6	6	8
$(2,2)^T$	5	8	11	8	8	8	5

than NEPEN and ASPEN since the nonlinear system is well-conditioned for $m = 1$. However, Newton's method may fail for certain initial guesses for $m = 3, 5$. It is clear that NEPEN and ASPEN are not sensitive to the parameter m.

In this second example, second equations of the original functions F and the preconditioned function $\mathcal{F}$ are different. Thus, the number of nonlinear iterations for NEPEN and ASPEN are not the same as in the first example. In addition to this, we see that NEPEN iterations are not dependent to parameter m. The nonlinear system shown in Equation 4.4 is a function of m but the equation solved in the nonlinear elimination routine is independent of m. Insensitivity of NEPEN to parameter m comes from the fact that m does not appear in the nonlinear elimination routine.

Unlike the first simple example, the second simple example has two roots. The underline numbers in the Table 4.2 indicate that those cases converged to the second root of the system. Clearly, the reason of this situation is the initial guesses. For a selected initial guess, the algorithms tend to converge the nearest root. However, since the left preconditioned nonlinear system has only the first root, ASPEN converged to the first root only.

The contour plots of the merit functions $f(u) = log_{10}(10||F(u)||_2^2 + 1)$ for the original function $F(u)$ and $f(u) = log_{10}(10||\mathcal{F}(u)||_2^2 + 1)$ for the left preconditioned function $\mathcal{F}(u)$ are demonstrated in Figure 4.4, Figure 4.5 and Figure 4.6. The figures also show the solution paths starting from the initial guess $u^{(0)} = (2, 2)^T$ through the root of the system. The results on the figures are obtained for $m = 5$.

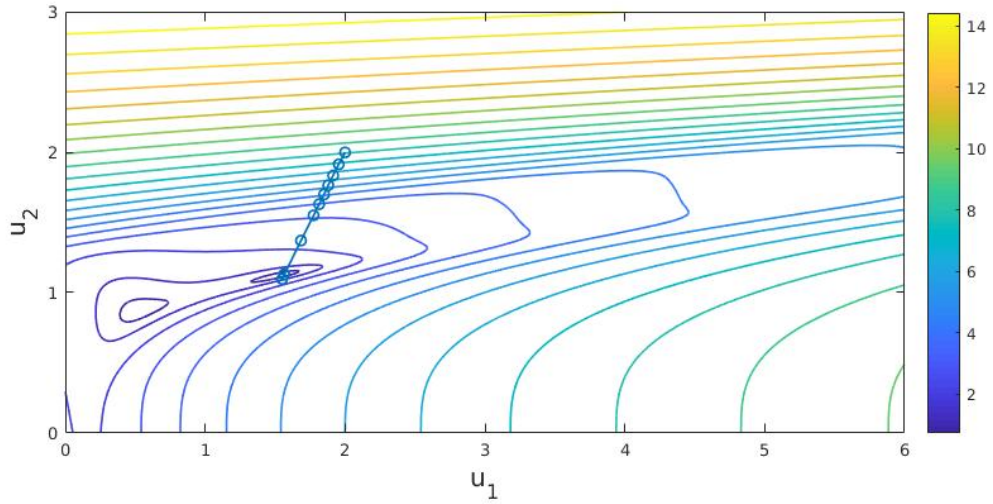

Figure 4.4. Contour plot of $log_{10}(10||F(u)||_2^2 + 1)$ and the solution path of Newton iterations for the second simple example with unbalanced nonlinearity.

The contours of the preconditioned system $\mathcal{F}(u)$ are more elliptical and regular whereas the contours of the original system $F(u)$ indicate the nonlinear imbalance of the system. The colors of the contours shows that, in the neighborhood of the root, the residuals of the original function are larger than the residuals of the preconditioned system. In addition, we see that the second root of the original system does not appear in the contours of the left preconditioned function since we lose that root while performing left preconditioning routine.

The intermediate approximate solutions for NEPEN and ASPEN are different as seen from the solution paths in Figure 4.5 and Figure 4.6. Unlike the first simple example, the solution paths of ASPEN and NEPEN are different than each other since the second equation in the systems (4.4) and (4.5) are different.

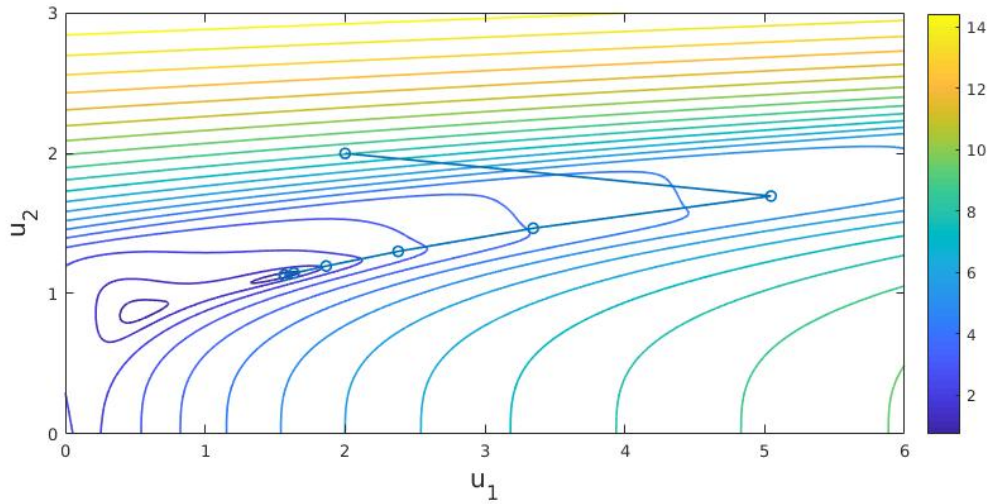

Figure 4.5. Contour plot of $log_{10}(10||F(u)||_2^2 + 1)$ and the solution path of NEPEN iterations for the second simple example with unbalanced nonlinearity.

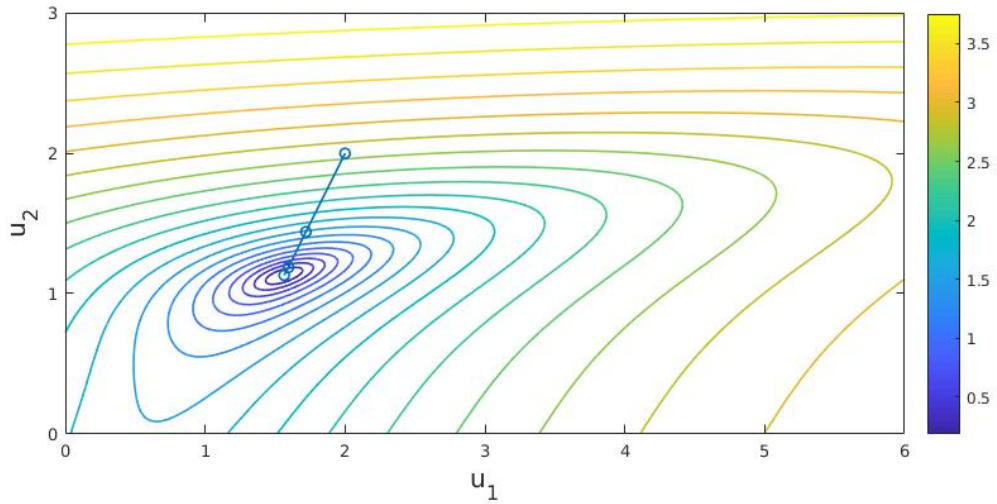

Figure 4.6. Contour plot of $log_{10}(10||\mathcal{F}(u)||_2^2 + 1)$ and the solution path of ASPEN iterations for the second simple example with unbalanced nonlinearity.

4.2. Test 2: Lid-driven Cavity Problem

We consider steady-state incompressible lid-driven cavity problem on square domain $\Omega = [0,1] \times [0,1]$ with three unknowns u, v and ω, namely horizontal velocity component, vertical velocity component and vorticity. The lid is moving through the positive direction of x and the other boundaries are defined as stationary walls. We solve Navier-Stokes equations in velocity-vorticity formulation given below.

$$\begin{aligned} -\Delta u - \frac{\partial \omega}{\partial y} &= 0 \\ -\Delta v + \frac{\partial \omega}{\partial x} &= 0 \\ -\frac{1}{Re}\Delta \omega + u\frac{\partial \omega}{\partial x} + v\frac{\partial \omega}{\partial y} &= 0 \end{aligned} \tag{4.6}$$

where vorticity component is given by its definition as

$$\omega(x,y) = -\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}. \tag{4.7}$$

In Equation 4.6, Reynolds number is a physical parameter defined as a ratio of inertia forces to viscous forces within a fluid. In Reynolds number formula $Re = \frac{V_m L}{\nu}$, V_m corresponds to velocity of the lid, L is the length of the side of the square cavity and ν is the kinematic viscosity. For high Reynolds number, convective terms dominate and the system becomes nonlinearly unbalanced. Thus, it affects the nonlinearity of the vorticity (3th) equation.

4.2.1. Discretization of the Problem

We discretize the governing nonlinear partial differential equations by using finite difference method on several different mesh sizes. Mesh points are ordered such that point numbers are increasing from left to right on a row and from bottom to top on a column. Ordering the unknown components is mesh point based meaning that at the mesh point (i,j), unknowns are ordered as $u_{(i,j)}$, $v_{(i,j)}$, and $\omega_{(i,j)}$.

Standard central finite differencing is used to discretize the gradient terms and 1st order terms whereas upwinding is used for convective terms. Dirichlet boundary conditions for velocity components are imposed as given below and the boundaries are demonstrated in Figure 4.7.

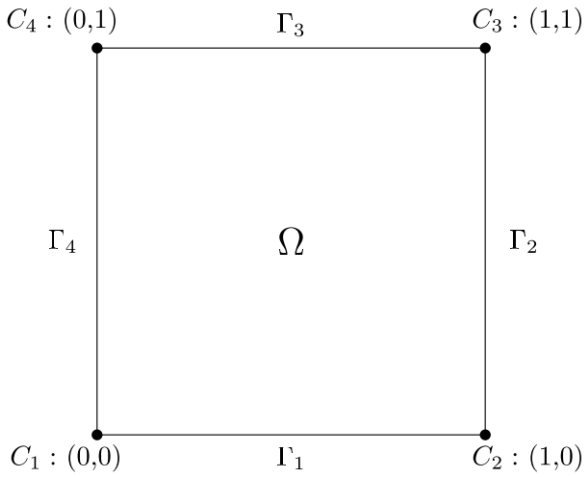

Figure 4.7. The computational domain of the lid-driven cavity problem [2].

(i) $u = 1$ on Γ_3, $u = 0$ elsewhere
(ii) $v = 0$ on the all boundaries

Boundary condition for vorticity component is a critical issue since it changes the nonlinear system and solution of the nonlinear system consequently. Moreover, the solution to some nonlinear systems may not be physically true for moderate and/or high Reynolds number. Therefore, we test two different discretization models in which the only difference is vorticity discretization on the boundaries. We employ the following two discretization models for vorticity equation shown in Equation 4.7 on the boundaries:

(i) Regular 1st order approximation in the first model
(ii) So-called efficient 2nd order approximation, using only the mesh points adjacent to boundary in the second modal

The 1th order approximation in the first model is very straightforward: 1th order forward or backward finite differences using 2 mesh points. An extensive devotion should be made for the 2nd model. Following [24], we describe 2nd order approximation to vorticity equations on boundaries by deriving the discrete equations for edge Γ_1 and corner C_1. Let the subindices r, l, a, b denote the right of, the left of, above and below a given mesh point, respectively. In a same manner, consider the subindices rr, ll, aa, bb, ra, ba, la, lb denoting a mesh point, for example rr means two right mesh point of the corresponding mesh point. For the mesh point on edge Γ_1, classical second order finite difference equation of vorticity equation is

$$\omega = -\frac{-3u + 4u_a - u_{aa}}{2h} + \frac{v_r - v_l}{2h} + O(h^2) \tag{4.8}$$

where h is regular mesh spacing size. The discretized form of Equation 4.7 for the mesh point above Γ_1 is

$$\omega_a = -\frac{u_{aa} - u}{2h} + \frac{v_{ar} - v_{al}}{2h} + O(h^2). \tag{4.9}$$

Taking the sum of the above expressions cancels u_{aa} term and gives

$$\omega + \omega_a + 2\frac{u_a - u}{h} - \frac{v_{ar} - v_{al}}{2h} - \frac{v_r - v_l}{2h} = O(h^2) \tag{4.10}$$

which uses only the adjacent mesh points. For the corner mesh points, we use the same technique by including the equations of the adjacent mesh points and performing appropriate summation to cancel out other mesh point data than adjacent mesh point data. We give the derived equation for corner mesh point C_1 below.

$$\omega + \omega_a + \omega_r + \omega_{ar} + 2\frac{u_a - u}{h} + 2\frac{u_{ar} - u_r}{h} - 2\frac{v_r - v}{h} - 2\frac{v_{ar} - v_a}{h} = O(h^2) \tag{4.11}$$

Here, we describe why we test two different discretization for vorticity equation on the boundaries. All of ASPIN studies on steady-state incompressible lid-driven cav-

ity problem with finite difference discretization used regular 1st order discretization for vorticity components on the boundaries. Although the composed nonlinear system has a solution, the solution becomes physically untrue with increasing Reynolds number after $Re = 100$. We see that in their journal articles, there are no solution plot demonstrated which is not surprising. Therefore, it may be better to regard the analyses performed by using Model 1 discretization as a benchmark nonlinear system. On the other hand, Yang and Hwang [2] used efficient 2nd order approximation for vorticity equation on the boundaries for steady-state incompressible lid-driven cavity problems in velocity-vorticity formulation to solve it by using adaptive nonlinear elimination preconditioned inexact Newton (INB-ANE). Their presented and plotted results are physically meaningful and have a good match with the ones of Ghia *et al.* [25].

We test two different discretization models: *Model 1* with 1st order approximation and *Model 2* efficient 2nd order approximation for vorticity equation on the boundaries. We perform various analysis to investigate the behavior of our algorithms on the nonlinear system arised from the two different discretization models. Besides many parametric analyses, using Model 1 discretization, we compare our results with some previously performed studies. The same parametric analyses have been conducted with Model 2 discretization as well. Ensuring the solution of Model 2 discretization is physically meaningful, we compare our solution with the ones of Ghia *et al.* [25].

4.2.2. Details of Numerical Experiments

In our numerical experiments, we investigate the effects of various parameters on the solution of steady-state incompressible lid-driven cavity problem discretized by two discretization models described in the previous subsection. We employ four nonlinear solvers namely, inexact Newton(IN), additive Schwarz preconditioned inexact Newton (ASPIN), field-split preconditioned inexact Newton (FSPIN) and nonlinear elimination preconditioned inexact Newton(NEPIN). In this subsection, we give the details of the numerical experiments.

We solve the problem arising from the three discretization models on two mesh sizes namely 64×64, 128×128 and 256×256 where Reynolds number varies $Re = 1, 10, 100, 1000, 5000, 10000$. Initial guess in all tests is zero for all unknowns. We stop nonlinear iterations if the sufficient reduction in nonlinear residual is achieved meaning that for IN and NEPIN

$$||F(u^{(k)})|| \leqslant \epsilon_{global-nonlinear}||F(u^{(0)})||, \tag{4.12}$$

for ASPIN and FSPIN

$$||\mathcal{F}(u^{(k)})|| \leqslant \epsilon_{global-nonlinear}||\mathcal{F}(u^{(0)})|| \tag{4.13}$$

where $u^{(0)}$ is the initial guess and $u^{(k)}$ is the approximate solution at kth iteration. We use $\epsilon_{global-nonlinear} = 10^{-9}$ in all tests unless otherwise is told. We define maximum nonlinear iteration number $maxit_{global-nonlinear} = 40$ to stop the nonlinear iterations if the problem tends to diverge or stagnates.

Matrix free Generalized Minimal Residual (GMRES) is employed to solve the global linear system within inexact Newton (IN) and NEPIN with an additive Schwarz linear preconditioner partitioned by 4×4 subdomains with overlap size $h_o = 2$. In ASPIN and FSPIN algorithm, the global Jacobian system is already well-conditioned under the favor of left nonlinear preconditioner. Thus, no linear preconditioner is employed. Initial guess for global linear systems is zero meaning that $p^{(k,0)} = 0$. Global linear iterations are stopped if the sufficient relative residual norm is achieved such that: for ASPIN and FSPIN algorithms

$$\|\mathcal{F}(u^{(k)}) - \sum_{i=1}^{N} J_{S_i}^{-1} J p^{(k)}\| \leq \epsilon_{global-linear}\|\mathcal{F}(u^{(k)})\|, \tag{4.14}$$

for IN and NEPIN algorithms

$$\|F(u^{(k)}) - J(u^{(k)})p^{(k)}\| \leq \epsilon_{global-linear}\|F(u^{(k)})\|. \tag{4.15}$$

In Equations 4.14 and 4.15, $\epsilon_{global-linear}$ acts as a forcing term η_k given in the algorithms. We use two constant tolerances for global linear systems $\epsilon_{global-linear} = 10^{-3}, 10^{-6}$. To calculate a norm of a vector, l^2 norm is used in all of our algorithms where norm calculation is done by taking square root of the summation of the squares of the vector components.

Within the nonlinearly preconditioned methods, local nonlinear systems have to be solved and we use inexact Newton method with two different relative tolerance values $\epsilon_{local-nonlinear} = (10^{-3}, 10^{-6})$. Initial guess is zero for local nonlinear systems unless the concerned algorithm defines a particular one such as ASPIN and FSPIN do. We limit the number of local nonlinear iterations $maxit_{local-nonlinear} = 25$. Local linear systems within the local nonlinear systems are solved directly using backslash operator. We should note that, in FSPIN and NEPIN, subproblems may be linear not nonlinear depending on the kind of splitting or the selection of components to be eliminated. We treat those linear subproblems as if they are nonlinear meaning that we still employ inexact Newton. In that case inexact Newton solves linear problem and finds the solution with one or two nonlinear iterations.

In all of our algorithms , we never calculate full Jacobian matrix J. Local Jacobian matrices are formed and stored as sparse matrices by using multi-colored forward finite difference approximation with step size $\epsilon = 10^{-8}$. Coloring routine is done by pre-knowledge of the Jacobian matrix sparsity pattern. Compressed Jacobian matrices are calculated in parallel where evaluation of each color is formed by one processor.

In ASPIN, we test various partitioning types with overlap size $h_o = 0, 1, ..., 5$ namely 1×2, 1×4, 2×1, 2×2, 2×4, 4×1, 4×2, 4×4 partitionings such that the first number indicates the number of subdomains through x axis and the second number indicates the number of subdomains through y axis. In FSPIN, the components are split such that they constitute two subsystems; uv and ω. In NEPIN, all vorticity components are chosen to be eliminated.

In some of our test we employ line search techniques namely cubic backtracking and half step line search to find optimum step length if the full step does not lead better approximate solution in nonlinear iterations. However, at the first few iterations of some cases, taking full step leads better convergence even if the full step cause an increase in nonlinear residual. Considering that situation, we mostly employ half step line search in ASPIN and FSPIN algorithms and no line search in IN and NEPIN algorithms unless otherwise is told.

Our code is fully written in Julia Language [26] without using any library or package. We exploit the multi-core architecture of CPUs by solving sub-systems in parallel since they do not depend on each other.

4.2.3. Numerical Verification of the Solution

In this subsection, we verify the solution of the nonlinear system discretized by Model 2. We solved steady-state incompressible lid-driven cavity problem for $Re = 1000, 3200, 5000$ on 128×128 and 256×256 meshes with NEPIN method. Global linear systems are solved by additive Schwarz preconditioned matrix-free GMRES until the relative residual condition defined by $\epsilon_{global-linear} = 10^{-6}$ is satisfied. For the elimination of vorticity components at the beginning of the Newton iterations, we used Newton method with the tolerance $\epsilon_{local-linear} = 10^{-3}$.

The following three figures, namely Figure 4.8, Figure 4.9 and Figure 4.10, demonstrate the streamline plots and the velocity profiles regarding the cases where Reynolds number $Re = 1000$, $Re = 3200$, and $Re = 5000$, respectively. Subfigures in each figure show respectively; the streamline plot(top), the profile of u component of the velocity at $x = 0.5$(middle) and the profile of v component of the velocity at $y = 0.5$(bottom).

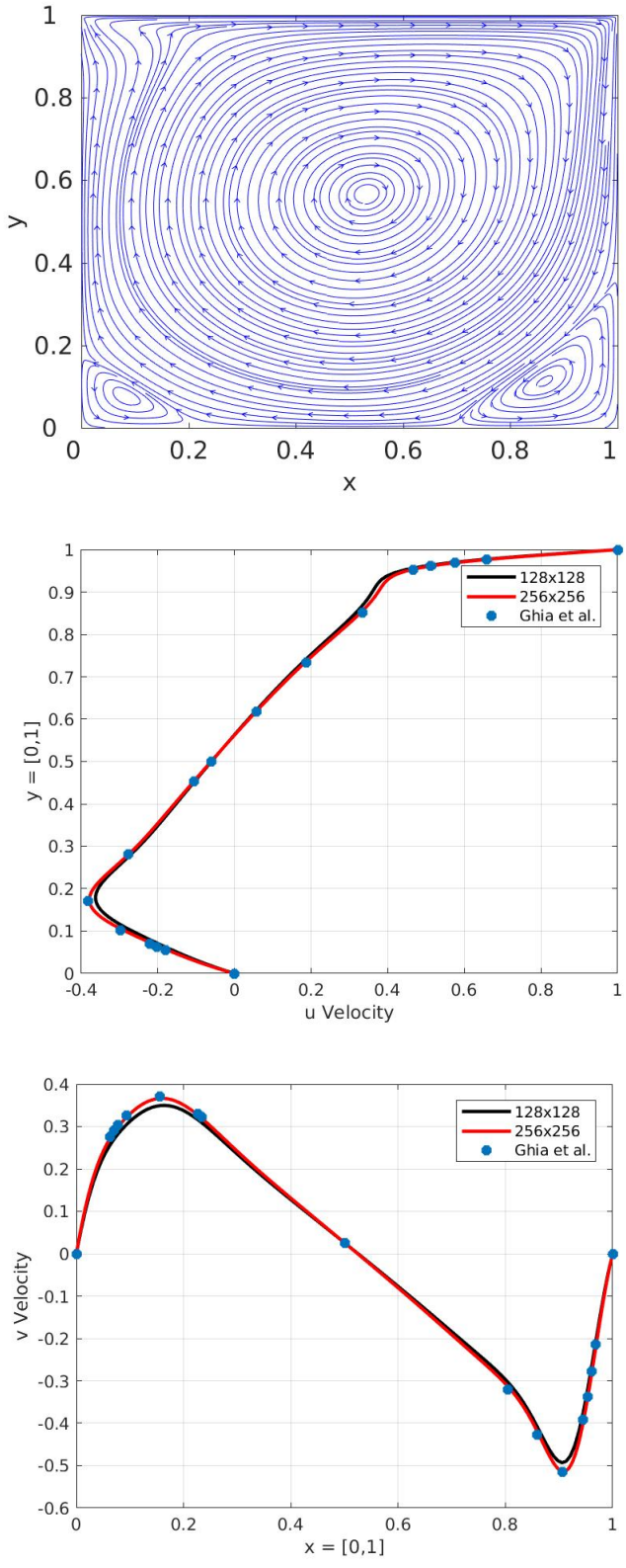

Figure 4.8. Streamlines and velocity profiles for steady-state lid-driven cavity problem discretized by Model 2 at $Re = 1000$.

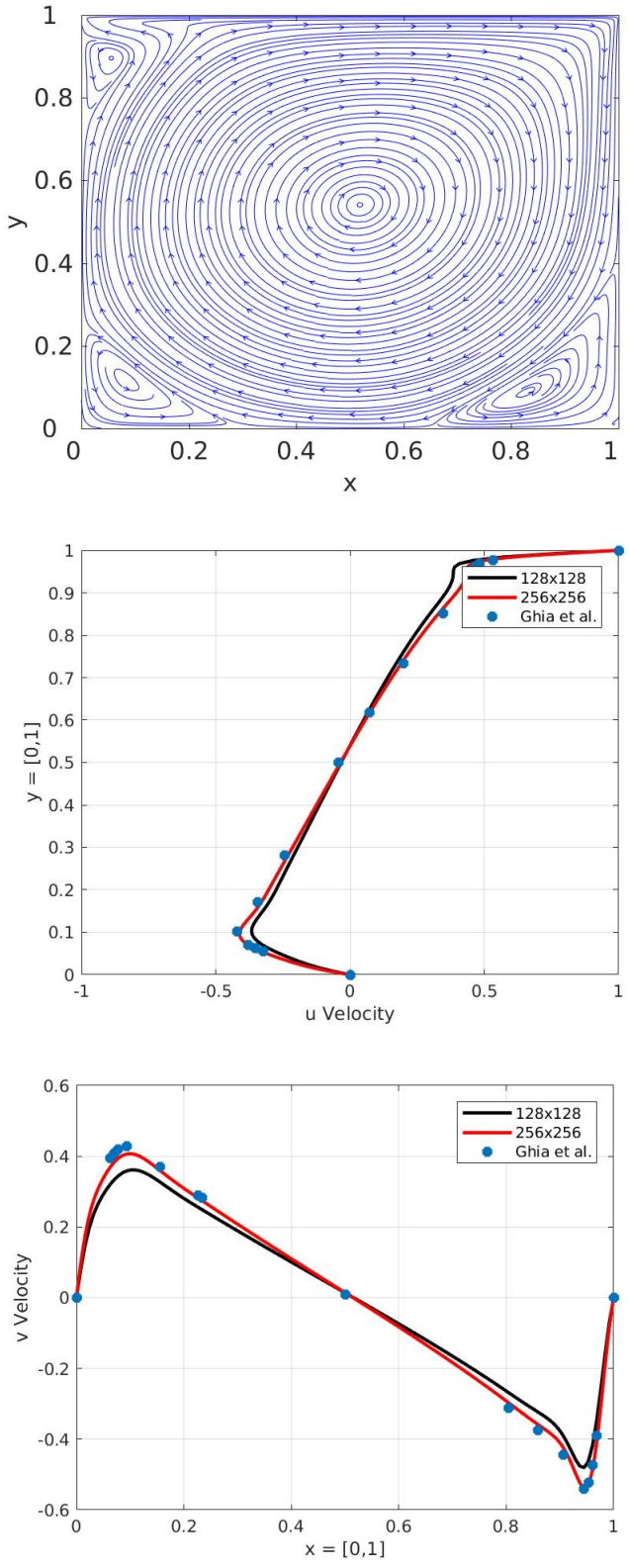

Figure 4.9. Streamlines and velocity profiles for steady-state lid-driven cavity problem discretized by Model 2 at $Re = 3200$.

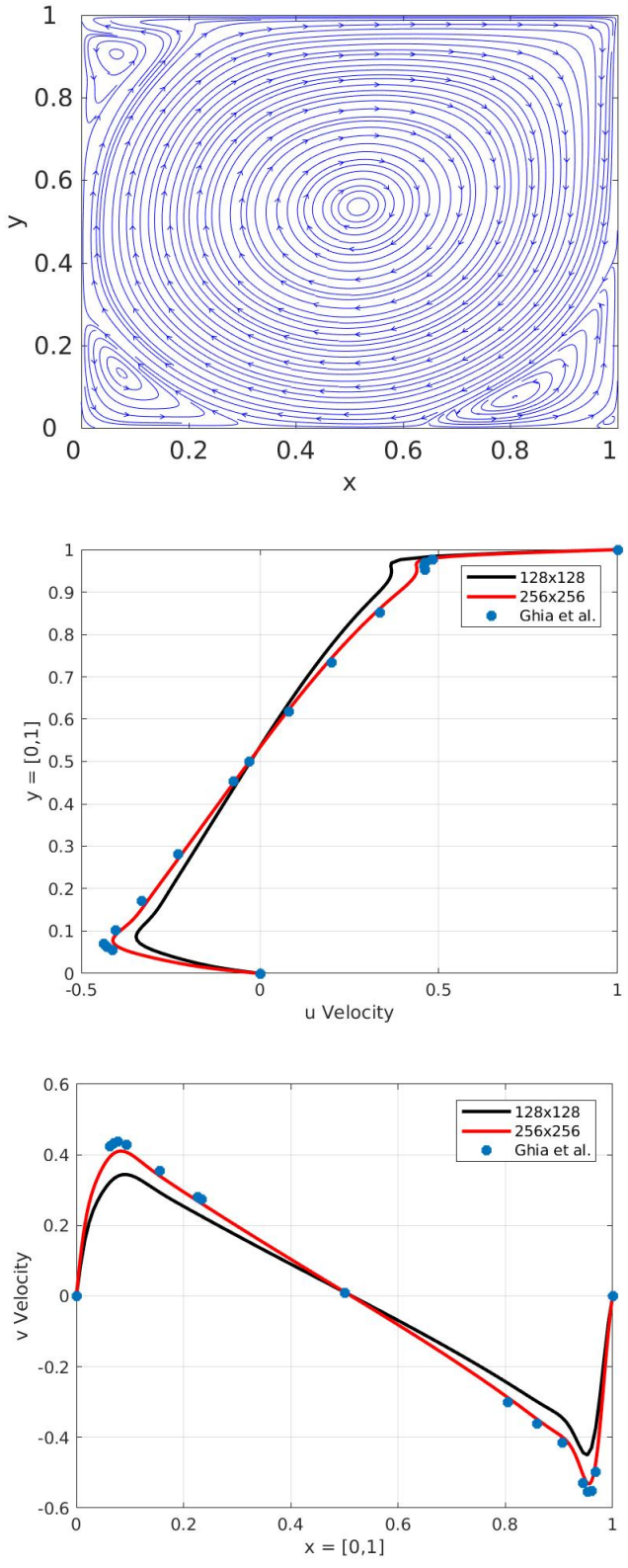

Figure 4.10. Streamlines and velocity profiles for steady-state lid-driven cavity problem discretized by Model 2 at $Re = 5000$.

Figures show that, as Reynolds number increases, the first two vortices start to appear in the left and right bottom corners on the square domain. After $Re = 1000$, third vortex is being formed in the left top corner and it is clearly seen for $Re = 3200$. When Reynolds number reaches to $Re = 5000$, forth vortex starts to occur in the right bottom corner below the first vortex and it shrinks the shape of the first vortex.

We test two mesh sizes 128×128 and 256×256 to validate our solution with well known published study by Ghia *et al.* [25]. By increasing the mesh size, we get more accurate solution. Due to our computer limits, we could not test the mesh sizes higher than 256×256. However, up to $Re = 5000$, figures show that 256×256 mesh yields accurate results.

4.2.4. Results and Discussion

In this section, we present the results of our numerical experiments and compare the methods with various parameters on steady-state incompressible lid-driven cavity problem. In tables, we show the number of nonlinear iterations and the number of GMRES iterations per nonlinear iteration. Fractional numbers are rounded up for the average number of GMRES iterations which are presented in the parenthesis on the tables.

We first discuss lid-driven cavity problem discretized by Model 1. Parametric studies and comparisons between the methods are done. We also compare our ASPIN results with the study done by Cai and Keyes [1] and our FSPIN results with the study done by Liu and Keyes [23]. Then we do some analyses with lid-driven cavity problem discretized by Model 2 which approximates lid-driven cavity problem better than the one with Model 1 especially for moderate and high Reynolds number.

4.2.4.1. Analyses obtained by Model 1. We compare nonlinear preconditioning methods with inexact Newton. Table 4.3 shows the global nonlinear iterations and the number of GMRES iteration per global nonlinear iteration for different methods and

Reynolds numbers. We solve the problem on 256×256 mesh with the tolerances $\epsilon_{local-nonlinear} = 10^{-3}$ and $\epsilon_{global-linear} = 10^{-6}$. Domain is divided into 2×2 subdomains with overlap size $h_o = 2$ in ASPIN. In FSPIN, the unknown components are split into two group uv and w which leads linear subproblems. Half step line search is used in local nonlinear systems and global nonlinear systems in ASPIN and FSPIN algorithms.

Table 4.3. Comparison between methods for different Reynolds numbers with respect to the number of nonlinear iterations and average GMRES iterations in lid-driven cavity problem discretized by Model 1.

	Re				
	1	10	100	1000	10000
ASPIN	3(46)	3(45)	4(52)	7(49)	6(39)
FSPIN	3(21)	4(25)	6(39)	11(57)	-
NEPIN	4(69)	5(73)	7(83)	9(81)	22(77)
IN	5(133)	5(133)	6(142)	-	-

Table 4.3 shows that, as Reynolds number increases, the problem becomes harder to be solved because it changes the nonlinear balance of the nonlinear function. Thus, for all methods, nonlinear iteration number generally grows when Reynolds number increases. Inexact Newton method fails to converge for $Re = 1000, 10000$ where nonlinearly preconditioned methods converge well even for the problems with high Reynolds numbers except FSPIN at $Re = 10000$ on this mesh.

The numbers of nonlinear iterations are close to each other in ASPIN for all Reynolds number, whereas in FSPIN and NEPIN nonlinear iterations increase noteworthily as Reynolds number increase. This implies that for the lid-driven cavity problem discretized by Model 1 on 256×256 mesh ASPIN is more robust than FSPIN and NEPIN.

It is clear in Table 4.3 that, the number of average GMRES iterations in preconditioned methods are considerably less than IN. This means that memory usage is reduced in GMRES routine for preconditioned methods and this is important espe-

cially for large problems. FSPIN requires less GMRES iterations than ASPIN for low Reynolds numbers because the fact that average GMRES iteration numbers are related with the number of subsystems in left nonlinear preconditioning methods. FSPIN has two groups whereas ASPIN has four subdomains. Thus it is expected that FSPIN requires less GMRES iterations than ASPIN. However, as Reynolds number increases, FSPIN starts to lose its effectiveness where ASPIN remains robust. We note that in ASPIN algorithm, overlap affects the average number of GMRES iterations and we point this issue in further analyses.

In Table 4.4, we demonstrate the effects of global linear tolerance and local nonlinear tolerance on the number of nonlinear iterations and the average number of GMRES iterations for tolerances $\epsilon_{local-nonlinear} = 10^{-3}$,$10^{-6}$ and $\epsilon_{global-linear} = 10^{-3}$,$10^{-6}$. The problem is solved on 256×256 mesh by using half step line search for both global and local nonlinear systems. No line search is used in NEPIN's global iterations. In ASPIN, we partition the domain into 2×2 subdomains with overlap size $h_o = 2$. Unknown components are split into two linear subsystems uv and w in FSPIN.

As seen in Table 4.4, the numbers of nonlinear iterations do not change much when we change $\epsilon_{local-nonlinear}$ and they decrease when we change $\epsilon_{global-linear}$ from 10^{-3} to 10^{-6} especially for high Reynolds numbers in ASPIN and FSPIN. On the other hand, NEPIN with $\epsilon_{global-linear} = 10^{-3}$ fails to converge for $Re = 10000$ for this mesh which implies that the global linear systems in NEPIN algorithms should be solved accurately with sufficiently small tolerance. Since $\epsilon_{global-linear}$ acts as a tolerance for global linear systems, the number of GMRES iterations increases for small $\epsilon_{global-linear}$.

Note that, although the subsystems are linear in FSPIN and NEPIN due to the splitting strategy and the choice of the components to be eliminated, we still solve them by using inexact Newton. We observed that for high Reynolds number, those local systems are solved in two iteration of IN even usual linear systems should be solved in one. As a result of this, FSPIN and NEPIN iterations may be affected by $\epsilon_{local-nonlinear}$ slightly. We believe that the reason of this situation is the error comes from the approximation of corresponding local Jacobians due to finite differences.

Table 4.4. The effects of $\epsilon_{global-linear}$ and $\epsilon_{local-nonlinear}$ on the number of nonlinear iterations and average GMRES iterations for ASPIN, FSPIN and NEPIN methods in lid-driven cavity problem discretized by Model 1.

	Re				
	1	10	100	1000	10000
	$\epsilon_{global-linear} = 10^{-3}$, $\epsilon_{local-nonlinear} = 10^{-3}$				
ASPIN	4(28)	4(26)	5(28)	10(21)	6(19)
FSPIN	4(13)	4(15)	7(24)	12(35)	-
NEPIN	6(44)	6(44)	8(45)	11(44)	-
	$\epsilon_{global-linear} = 10^{-3}$, $\epsilon_{local-nonlinear} = 10^{-6}$				
ASPIN	4(28)	4(26)	5(28)	9(20)	6(19)
FSPIN	4(13)	4(15)	7(24)	12(35)	-
NEPIN	6(44)	6(44)	8(47)	10(42)	-
	$\epsilon_{global-linear} = 10^{-6}$, $\epsilon_{local-nonlinear} = 10^{-3}$				
ASPIN	3(46)	3(45)	4(52)	7(49)	6(39)
FSPIN	3(21)	4(25)	6(39)	11(57)	-
NEPIN	4(69)	5(73)	7(83)	9(81)	22(77)
	$\epsilon_{global-linear} = 10^{-6}$, $\epsilon_{local-nonlinear} = 10^{-6}$				
ASPIN	3(45)	3(45)	4(52)	7(49)	6(39)
FSPIN	3(21)	4(24)	6(39)	12(57)	-
NEPIN	4(69)	5(74)	7(83)	9(81)	22(72)

We investigate the effects of the choice of global line search technique in Table 4.5 for various Reynolds numbers. Lid-driven cavity problem is solved on 256×256 mesh and following tolerances are used: $\epsilon_{local-nonlinear} = 10^{-3}$ and $\epsilon_{global-linear} = 10^{-6}$. In ASPIN, partitioning is 2×2 with overlap size $h_o = 2$. In FSPIN, the unknown components are split into two group uv and w.

Table 4.5. Line search effects on the number of nonlinear iterations and average GMRES iterations for ASPIN, FSPIN and NEPIN methods in lid-driven cavity problem discretized by Model 1.

	Re				
	1	10	100	1000	10000
	No line search				
ASPIN	3(46)	3(45)	4(52)	7(49)	8(39)
FSPIN	3(21)	4(25)	5(39)	-	-
NEPIN	4(69)	5(73)	7(83)	9(81)	22(77)
IN	5(133)	5(133)	6(142)	-	-
	Cubic backtracking				
ASPIN	3(46)	3(45)	4(52)	7(49)	6(38)
FSPIN	3(21)	4(25)	10(38)	18(62)	-
NEPIN	-	-	30(69)	14(79)	27(73)
IN	33(114)	30(117)	30(127)	25(136)	-
	Half step line search				
ASPIN	3(46)	3(45)	4(52)	7(49)	6(39)
FSPIN	3(21)	4(25)	6(39)	11(57)	-
NEPIN	-	40(65)	18(71)	11(79)	21(70)
IN	21(116)	20(119)	21(129)	18(137)	-

ASPIN and FSPIN algorithms are not sensitive to global line search method for low Reynolds numbers since the search directions always reduce the residual of the preconditioned function. In ASPIN algorithm, employing a line search algorithm saves two nonlinear iterations at $Re = 10000$. FSPIN algorithm is very sensitive to line search for $Re = 1000$ and it even changes the robustness of the algorithm.

We observe big difference in the number of nonlinear iterations in IN and NEPIN algorithms in Table 4.5. Remember that, in IN and NEPIN we do not construct new nonlinear function, we stick with the original nonlinear function. Thus, we consider the residual of the original nonlinear function. In our implementations and results, we observed the following situation. In some cases, in order to step forward through a point which leads less nonlinear residual ,we might need to take a few steps through points which increase the nonlinear residual at the first few nonlinear iterations. Then, IN and NEPIN can find good direction at those points. Therefore, employing a line search technique for IN and NEPIN slows down the convergence because it makes difficult to reach good points that leads good reduction in nonlinear residual. Interestingly, this issue influences NEPIN much since it does not converge with line search algorithms for low Reynolds numbers. However, we should note that employing line search makes it possible to converge IN for $Re = 1000$.

In the forthcoming two tables, we analyze various overlap sizes for different number of subdomains and different pattern in regular quadrilateral partitioning for ASPIN on 256×256 mesh. Overlap sizes varies as $h_o = 0, 1, ..., 5$ and overlap values correspond one mesh point distance in regular structured mesh. We present our results on two tables; Table 4.6 for 1×2, 1×4, 2×1, 2×2 partitions and Table 4.7 for 2×4, 4×1, 4×2, 4×4 partitions. Tolerances are set $\epsilon_{local-nonlinear} = 10^{-3}$ and $\epsilon_{global-linear} = 10^{-6}$. Half step line search algorithm is employed in both global and local nonlinear systems.

It is clear that growing Reynolds number increases the number of ASPIN iterations for all partition and overlap combinations except 2×1 and 2×2 partitioned cases where the maximum numbers of ASPIN iterations are obtained at $Re = 1000$. In the number of GMRES iterations, there is no descent pattern related with the increase in Reynolds number.

We see that some cases with 4×1, 4×2 and 4×4 partitioning fail to converge at $Re = 10000$. The reason of the failures is that solving the subdomains which touch the lid may become problematic depending on the partitioning and overlap. Even though we employ globalization method and continuation method, in some cases we can not

Table 4.6. The number of nonlinear iterations and GMRES iterations for different Reynolds numbers and partitions in lid-driven cavity problem discretized by Model 1.

		Re				
	overlap	1	10	100	1000	10000
1×2	0	3(64)	3(64)	4(67)	4(59)	5(40)
	1	3(40)	3(41)	4(46)	4(39)	5(27)
	2	3(32)	3(34)	4(37)	4(32)	5(22)
	3	3(27)	3(30)	4(33)	4(28)	5(19)
	4	3(25)	3(27)	4(30)	4(26)	5(17)
	5	3(23)	3(25)	4(27)	4(23)	5(16)
1×4	0	3(97)	3(87)	5(103)	5(86)	6(76)
	1	3(60)	3(56)	5(64)	5(59)	6(46)
	2	3(47)	3(45)	5(53)	5(48)	6(37)
	3	3(41)	3(40)	5(45)	5(42)	6(31)
	4	3(37)	3(35)	5(41)	5(37)	6(28)
	5	3(34)	3(33)	5(37)	5(34)	6(26)
2×1	0	3(79)	3(71)	5(71)	10(71)	6(58)
	1	3(48)	3(46)	6(46)	10(44)	6(37)
	2	3(39)	3(36)	6(37)	9(36)	6(29)
	3	3(32)	3(32)	6(32)	9(31)	6(26)
	4	3(28)	3(29)	6(29)	9(28)	6(23)
	5	3(26)	3(27)	6(27)	9(25)	6(21)
2×2	0	3(84)	3(88)	4(96)	7(86)	7(76)
	1	3(56)	3(58)	4(63)	7(60)	6(48)
	2	3(45)	3(46)	4(52)	7(49)	6(39)
	3	3(39)	3(40)	4(46)	7(41)	6(34)
	4	3(34)	3(35)	4(42)	7(37)	6(32)
	5	3(32)	3(33)	4(38)	7(35)	6(30)

Table 4.7. The number of nonlinear iterations and GMRES iterations for different Reynolds numbers and partitions in lid-driven cavity problem discretized by Model 1 (continued).

		Re				
	overlap	1	10	100	1000	10000
2 × 4	0	3(117)	3(105)	4(111)	6(115)	8(97)
	1	3(75)	3(71)	4(78)	6(78)	7(63)
	2	3(60)	3(58)	4(66)	6(63)	7(50)
	3	3(50)	3(51)	4(59)	6(53)	7(44)
	4	3(46)	3(46)	4(52)	6(48)	6(39)
	5	3(43)	3(43)	4(49)	6(44)	6(37)
4 × 1	0	3(101)	3(91)	5(106)	9(99)	-
	1	3(65)	3(59)	5(65)	9(62)	8(51)
	2	3(52)	3(47)	5(52)	8(48)	8(41)
	3	3(44)	3(41)	4(45)	8(41)	-
	4	3(39)	3(37)	4(41)	7(38)	10(36)
	5	3(35)	3(34)	4(37)	8(35)	9(32)
4 × 2	0	3(117)	3(109)	5(124)	9(123)	-
	1	3(79)	3(73)	5(86)	8(81)	8(65)
	2	3(61)	3(58)	4(66)	9(64)	8(52)
	3	3(54)	3(50)	4(59)	8(54)	21(53)
	4	3(48)	3(43)	4(53)	7(49)	11(44)
	5	3(43)	3(41)	4(49)	7(44)	9(41)
4 × 4	0	3(130)	3(124)	5(141)	11(152)	12(123)
	1	3(85)	3(83)	5(97)	9(100)	23(80)
	2	3(68)	3(68)	4(74)	8(76)	9(60)
	3	3(59)	3(59)	4(64)	7(63)	-
	4	3(52)	3(53)	4(59)	8(58)	14(56)
	5	3(48)	3(47)	4(55)	7(52)	11(47)

avoid the failure on the solution of these subdomains at high Reynolds numbers because of the two facts; the partitioning and the harmful effect of the communication between subdomains in Step 1 of ASPIN algorithm. However, the failure cases with no overlap are not related with the communication effect since the overlap is $h_o = 0$. In addition to this, we see that the cases with 4×2 overlap size $h_o = 3$ and 4×4 overlap size $h_o = 1$ have barely success to converge with large number of ASPIN iterations without the subdomain failure.

Overlap size has a great impact on GMRES iteration in ASPIN algorithm since increasing overlap enhances the communication between subdomains. It is easily seen that, overlap decreases the number of average GMRES iterations significantly for all cases. Moreover, it slightly reduces the number of ASPIN iterations for some cases.

The numbers of GMRES iterations appear to be affected mainly by the total number of subdomains rather than Reynolds number for a fixed overlap size. We see that in 4×4 partitioning, GMRES iterations are very higher than any other partitioning. In addition to this, when we compare the partitionings 1×4 with 4×1 or 2×4 with 4×2 whose number of subdomains are equal, generally the ones with smaller number of subdomains through x direction need less GMRES iterations. This situation is also held for the number of ASPIN iterations. Thus, for this problem on this mesh, we conclude that dividing the domain in x axis causes more increase in the number of ASPIN and average GMRES iterations than dividing the domain in y axis.

In Table 4.8, we compare the result of our ASPIN algorithm with the main work of ASPIN algorithm conducted by Cai and Hwang [1] on 128×128 mesh partitioned 4×4 subdomains with overlap sizes $h_o = 0, 1$ for different Reynolds numbers. Following tolerances are used which are the same as in [1]; $\epsilon_{global-nonlinear} = 10^{-10}$, $\epsilon_{global-linear} = 10^{-6}$ and $\epsilon_{local-nonlinear} = 10^{-3}$. We note that, in [1], cubic line search algorithm was employed where we use both half step and cubic backtracking line search techniques.

We see that our two algorithm, one with half step line search and one with cubic backtracking, execute the same performance for low Reynolds numbers and there are

Table 4.8. Comparison between our results with the main work of ASPIN algorithm in lid-driven cavity problem discretized by Model 1.

		Re				
	overlap	1	10	100	1000	10000
$\text{ASPIN}_{\text{half-step line search}}$	0	3(95)	4(96)	5(109)	9(100)	9(84)
$\text{ASPIN}_{\text{cubic backtracking}}$		3(95)	4(96)	5(109)	8(99)	8(84)
Cai and Keyes [1]		2(118)	3(100)	4(153)	8(141)	7(130)
$\text{ASPIN}_{\text{half-step line search}}$	1	3(63)	3(63)	5(74)	8(64)	8(55)
$\text{ASPIN}_{\text{cubic backtracking}}$		3(63)	3(63)	5(74)	8(66)	7(54)
Cai and Keyes [1]		2(71)	3(62)	4(87)	8(80)	7(69)

very slight differences for high Reynolds numbers between them. In the main work of ASPIN, some of the number of nonlinear iterations are lesser than our results with one or two iterations. However, the numbers of GMRES iterations are fewer considerably in our results even though the partitions and overlap sizes are equal. We believe that some minor differences in the implementation lead this issue.

The comparison between our FSPIN algorithm and the main work of FSPIN algorithm proposed by Liu and Keyes [23] is demonstrated in Table 4.9 on three different meshes 64×64, 128×128 and 256×256 for $Re = 10, 100, 1000$. Following tolerances are used which are the same as in [23]; $\epsilon_{global-nonlinear} = 10^{-10}$, $\epsilon_{global-linear} = 10^{-6}$. In [23], local linear systems are solved by incomplete LU (ILU) factorization preconditioned GMRES with the tolerance $\epsilon_{local-linear} = 10^{-3}$ where we solve the local systems by inexact Newton with tolerance $\epsilon_{local-nonlinear} = 10^{-3}$. Field variables are split in a such manner that u and v constitute one partition and w constitute a partition itself. We note that, we consider the additive Schwarz based field-split nonlinear preconditioner in [23]. We also note that, in [23], cubic line search algorithm was employed where we use both half step and cubic backtracking line search techniques.

We observe that for all mesh sizes, we obtain the same numbers of nonlinear iterations as in [23] for our FSPIN algorithm with cubic backtracking whereas our global linear iterations are less than the main work of FSPIN method. The increment in the

Table 4.9. Comparison between our results with the main work of FSPIN algorithm in lid-driven cavity problem discretized by Model 1.

	Re		
	10	100	1000
	64×64		
$\text{FSPIN}_{\text{half-step line search}}$	4(23)	6(32)	9(33)
$\text{FSPIN}_{\text{cubic backtracking}}$	4(23)	9(32)	10(33)
Liu and Keyes [23]	4(23)	9(37)	10(39)
	128×128		
$\text{FSPIN}_{\text{half-step line search}}$	4(24)	6(36)	11(42)
$\text{FSPIN}_{\text{cubic backtracking}}$	4(24)	9(35)	13(44)
Liu and Keyes [23]	4(24)	9(42)	13(64)
	256×256		
$\text{FSPIN}_{\text{half-step line search}}$	4(25)	6(39)	11(57)
$\text{FSPIN}_{\text{cubic backtracking}}$	4(24)	10(38)	18(62)
Liu and Keyes [23]	4(25)	10(47)	18(139)

number of global linear iterations may originate from the implementation difference in solving local systems which is explained above.

We also see that in Table 4.9, employing half step line search routine improves the method in terms of both the number of nonlinear iterations and GMRES iterations for $Re = 100, 1000$. For $Re = 10$, the choice of line search method does not affect the solution process significantly.

In this analysis, we also end up with having mesh independency test such that the mesh size slightly increases the number of nonlinear iterations for high Reynolds numbers whereas it remains the same for low Reynolds numbers. This implies that FSPIN algorithm is acceptably robust for this problem where the maximum Reynolds number is 1000.

We next investigate the effect of ϵ_{switch} on the solution of NEPIN algorithm for different Reynolds numbers on 256×256 mesh in Table 4.10. Tolerances are set as $\epsilon_{global-linear} = 10^{-6}$ and $\epsilon_{local-nonlinear} = 10^{-3}$. Recall that we choose all vorticity components to be eliminated in nonlinear elimination routine in NEPIN algorithm.

Table 4.10. The effect of ϵ_{switch} on NEPIN algorithm for various Reynolds numbers in lid-driven cavity problem discretized by Model 1.

	Re				
ϵ_{switch}	1	10	100	1000	10000
100	4(68)	5(74)	7(83)	10(80)	22(75)
10	4(68)	5(74)	7(83)	9(80)	22(77)
1	4(69)	5(73)	7(83)	9(81)	22(77)
0.1	4(69)	5(73)	7(83)	9(82)	22(77)
0.01	4(69)	5(73)	7(83)	9(81)	22(77)

In Table 4.10, we observe that the numbers of NEPIN iterations are not sensitive to ϵ_{switch} which is a threshold parameter that decides the usage of nonlinear elimination routine. Recall that elimination routine is employed only if $||F(u^{(k)})|| \leqslant \epsilon_{switch}$ at the current nonlinear iteration. For low Reynolds numbers, it is expected that the number of nonlinear iterations remains unchanged for different ϵ_{switch} since the scales of ϵ_{switch} may be skipped by one or two nonlinear iterations.

To show the details regarding the effect of ϵ_{switch} for the cases where Reynolds number is high, we give the absolute residual norms up to four decimal points at NEPIN iterations for $Re = 1000$ case in Table 4.11.

In Table 4.11, equal sign "=" is used to avoid writing the same residual norms which are the same as the left of that cell. The superscript "$*$" is used to indicate that the application of nonlinear elimination routine within the NEPIN has ended just before that iteration since the absolute residual norm is lower than ϵ_{switch}. The minus sign "$-$" is used at 10th iterations of some ϵ_{switch} because the algorithms have converged at the 9th iterations.

Table 4.11. The absolute residual norms up to four decimal points at NEPIN iterations for $Re = 1000$ in lid-driven cavity problem discretized by Model 1.

	ϵ_{switch}				
NEPIN it.	100	10	1	0.1	0.01
0	15.9373	=	=	=	=
1	12360.7493	=	=	=	=
2	1639.8957	=	=	=	=
3	485.7903	=	=	=	=
4	39.3328*	=	=	=	=
5	10.2811	7.7626*	=	=	=
6	0.8627	0.60522	0.4163*	=	=
7	0.0600	0.0256	0.02090	0.0283*	=
8	0.0003	6.6683e-5	4.8885e-5	2.9742e-5	3.6580e-5*
9	4.2513e-8	1.0460e-8	2.9045e-9	4.1975e-9	1.1201e-9
10	4.4709e-10	-	-	-	-

When we decrease ϵ_{switch} value, nonlinear elimination routine is being employed for more iterations. Consider the two cases where $\epsilon_{switch} = 10$ and $\epsilon_{switch} = 1$. The residual norm at 5th iteration for $\epsilon_{switch} = 10$ is below ϵ_{switch} value and at the next iteration NEPIN acts just as Newton iteration since it does not include nonlinear elimination routine. On the other hand, the case where $\epsilon_{switch} = 1$ employs the nonlinear elimination at 6th iteration. We see that the resultant residual norms of the 6th iterations of both cases are not so different such that they have not major impact on the rest of the solution prosedure. Same thing is also held for other ϵ_{switch} values concluding that the number of NEPIN iterations is not so sensitive to ϵ_{switch} for this problem on this mesh.

4.2.4.2. Analyses obtained by Model 2. We now demonstrate our results for steady-state incompressible lid-driven cavity problem discretized by Model 2 which leads better solution especially for moderate and high Reynolds numbers. Here, we perform very similar tests as we did in the previous subsection. The tolerances and the details

regarding the tests are the same as described in Model 1 tests unless otherwise reported. Note that, our methods fail to converge at $Re = 10000$ for this discretization. Thus, we solve the problem with $Re = 5000$ at the maximum case in respect to Reynolds number.

We compare inexact Newton method with nonlinearly preconditioned methods for different Reynolds numbers in Table 4.12. We solve the problem on 256×256 mesh with the tolerances $\epsilon_{local-nonlinear} = 10^{-3}$ and $\epsilon_{global-linear} = 10^{-6}$. Domain is divided into 2×2 subdomains with overlap size $h_o = 2$ in ASPIN. In FSPIN, the unknown components are split into two group uv and w which leads linear subproblems. Half step line search is used in local nonlinear systems and global nonlinear systems in ASPIN and FSPIN algorithms.

Table 4.12. Comparison between methods for different Reynolds numbers with respect to the number of nonlinear iterations and average GMRES iterations in lid-driven cavity problem discretized by Model 2.

	Re				
	1	10	100	1000	5000
ASPIN	3(45)	3(47)	4(53)	9(64)	-
FSPIN	3(23)	4(26)	7(43)	12(85)	-
NEPIN	5(69)	5(70)	7(81)	9(100)	21(130)
IN	5(131)	5(130)	6(143)	-	-

Different than the results obtained from the problem discretized by Model 1, ASPIN method fails to converge for $Re = 5000$. However, NEPIN converges for all Reynolds numbers in the tests. Inexact Newton can only converge for low Reynolds numbers as in the tests with Model 1. This is an important point such that ASPIN fails at $Re = 5000$ for the nonlinear system discretized by Model 2 meaning that ASPIN is not robust for this nonlinear system. We claim this conclusion because there is no previously done study on the solution of steady-state incompressible lid-driven cavity problem where efficient 2nd order discretization used on the boundaries for vorticity component by ASPIN and FSPIN algorithm. FSPIN method converges up to $Re = 1000$ as it is the case in Model 1.

When we compare the results based on Model 1 and Model 2 discretization, we see that Model 2 constitutes slightly harder nonlinear system to be solved for low and moderate Reynolds numbers since its number of nonlinear iterations and average GMRES iterations are slightly greater than Model 1's.

In Table 4.13, we demonstrate the effects of global linear tolerance and local nonlinear tolerance on the number of nonlinear iterations and the average number of GMRES iterations for tolerances $\epsilon_{local-nonlinear} = 10^{-3}$,$10^{-6}$ and $\epsilon_{global-linear} = 10^{-3}$,$10^{-6}$. The problem is solved on 256×256 mesh by using half step line search for both global and local nonlinear systems. No line search is used in NEPIN's global iterations. In ASPIN, we partition the domain into $2\times$ subdomains with overlap size $h_o = 2$. Unknown components are split into two linear subsystems uv and w in FSPIN.

We have found that $\epsilon_{local-nonlinear}$ does not affect the solution so much for Model 2 discretization and the same conclusion is valid for Model 1 too. However, it appears that $\epsilon_{local-nonlinear}$ has an impact on the solution of NEPIN method for high Reynolds number such as decreasing $\epsilon_{local-nonlinear}$ from 10^{-3} to 10^{-6} increases the number of NEPIN iterations by three. This situation implies that for high Reynolds numbers, the tolerance of the nonlinear elimination routine on vorticity component may influence the solution noteworthily.

We investigate the effects of the choice of global line search technique in Table 4.14 for various Reynolds numbers. Lid-driven cavity problem is solved on 256×256 mesh and following tolerances are used: $\epsilon_{local-nonlinear} = 10^{-3}$ and $\epsilon_{global-linear} = 10^{-6}$. In ASPIN, partitioning is 2×2 with overlap size $h_o = 2$. In FSPIN, the unknown components are split into two groups; uv and w.

The results demonstrated in Table 4.14 are analogous with the results of the lid-driven cavity problem discretized by Model 1. Therefore, to avoid repetition we do not explain the table here.

Table 4.13. The effects of $\epsilon_{global-linear}$ and $\epsilon_{local-nonlinear}$ on the number of nonlinear iterations and average GMRES iterations for ASPIN, FSPIN and NEPIN methods in lid-driven cavity problem discretized by Model 2.

	Re				
	1	10	100	1000	5000
	$\epsilon_{global-linear} = 10^{-3}$, $\epsilon_{local-nonlinear} = 10^{-3}$				
ASPIN	4(27)	4(25)	5(28)	13(39)	-
FSPIN	4(14)	5(16)	8(26)	14(58)	-
NEPIN	6(45)	6(42)	8(45)	11(59)	-
	$\epsilon_{global-linear} = 10^{-3}$, $\epsilon_{local-nonlinear} = 10^{-6}$				
ASPIN	4(27)	4(25)	5(28)	14(39)	-
FSPIN	4(14)	5(16)	8(26)	14(59)	-
NEPIN	6(45)	6(42)	8(45)	11(59)	-
	$\epsilon_{global-linear} = 10^{-6}$, $\epsilon_{local-nonlinear} = 10^{-3}$				
ASPIN	3(45)	3(47)	4(53)	9(64)	-
FSPIN	3(23)	4(26)	7(43)	12(85)	-
NEPIN	5(69)	5(70)	7(81)	9(100)	21(130)
	$\epsilon_{global-linear} = 10^{-6}$, $\epsilon_{local-nonlinear} = 10^{-6}$				
ASPIN	3(45)	3(47)	4(53)	9(64)	-
FSPIN	3(23)	4(26)	7(43)	12(85)	-
NEPIN	5(69)	5(69)	7(81)	9(101)	24(133)

Table 4.14. Line search effects on the number of nonlinear iterations and average GMRES iterations for ASPIN, FSPIN and NEPIN methods in lid-driven cavity problem discretized by Model 2.

	Re				
	1	10	100	1000	5000
	No line search				
ASPIN	3(45)	3(47)	4(53)	-	-
FSPIN	3(23)	4(26)	5(43)	-	-
NEPIN	5(69)	5(70)	7(81)	9(100)	21(130)
IN	5(131)	5(130)	6(143)	-	-
	Cubic backtracking				
ASPIN	3(45)	3(47)	4(53)	10(63)	-
FSPIN	3(23)	4(26)	12(42)	15(93)	-
NEPIN	-	-	32(71)	16(93)	12(118)
IN	22(62)	22(65)	23(73)	24(94)	-
	Half-step line search				
ASPIN	3(45)	3(47)	4(53)	9(64)	-
FSPIN	3(23)	4(26)	7(43)	12(85)	-
NEPIN	-	-	20(72)	10(94)	18(131)
IN	17(62)	17(65)	18(73)	18(94)	-

In the forthcoming two tables, we analyze various overlap sizes for different numbers of subdomains and different patterns in regular quadrilateral partitioning for ASPIN on 256×256 mesh. Overlap sizes varies as $h_o = 0, 1, ..., 5$ and overlap values correspond one mesh point distance in regular structured mesh. We present our results on two tables; Table 4.15 for 1×2, 1×4, 2×1, 2×2 partitions and Table 4.16 for 2×4, 4×1, 4×2, 4×4 partitions. Tolerances are set $\epsilon_{local-nonlinear} = 10^{-3}$ and $\epsilon_{global-linear} = 10^{-6}$. Half-step line search algorithm is employed in both global and local nonlinear systems.

As shown in the Table 4.15 and Table 4.16, ASPIN algorithm can not converge for $Re = 5000$ for all cases. Moreover, some partitioning and overlap combinations leads failure for $Re = 1000$ too. The numbers of ASPIN iterations are very similar for $Re = 1, 10, 100$ between Model 1 and Model 2 discretization, however, the numbers of ASPIN iterations of the results of Model 2 for $Re = 1000$ are considerably larger than the ones obtained from Model 1.

We next investigate the effect of ϵ_{switch} on the solution of NEPIN algorithm for different Reynolds number on 256×256 mesh in Table 4.17. Tolerances are set as $\epsilon_{global-linear} = 10^{-6}$ and $\epsilon_{local-nonlinear} = 10^{-3}$.

In Table 4.17, we observe that the number of NEPIN iterations is not sensitive to ϵ_{switch} as the results of Model 1 . Here, we do not give the residual norms of the iterations to avoid repetition since it is a very similar situation as Model 1 case.

Table 4.15. The number of nonlinear iterations and GMRES iterations for different Reynolds numbers and partitions in lid-driven cavity problem discretized by Model 2.

		Re				
	overlap	1	10	100	1000	5000
1×2	0	3(65)	3(63)	5(72)	11(82)	-
	1	3(40)	3(41)	4(42)	11(50)	-
	2	3(32)	3(32)	5(36)	11(40)	-
	3	3(27)	3(28)	4(30)	11(34)	-
	4	3(25)	3(25)	4(27)	10(32)	-
	5	3(22)	3(23)	4(24)	10(30)	-
1×4	0	3(96)	4(95)	5(98)	8(133)	-
	1	3(59)	4(61)	5(62)	8(79)	-
	2	3(47)	4(48)	5(51)	8(62)	-
	3	3(40)	4(41)	5(43)	10(54)	-
	4	3(36)	4(36)	5(39)	10(50)	-
	5	3(33)	4(35)	5(36)	9(46)	-
2×1	0	3(75)	3(69)	6(66)	-	-
	1	3(46)	3(44)	6(42)	-	-
	2	3(35)	3(35)	6(34)	-	-
	3	3(31)	3(31)	6(29)	-	-
	4	3(27)	3(28)	6(27)	10(31)	-
	5	3(25)	3(24)	6(24)	11(28)	-
2×2	0	3(83)	3(81)	4(95)	18(130)	-
	1	3(55)	3(58)	4(66)	10(79)	-
	2	3(45)	3(47)	4(53)	9(64)	-
	3	3(38)	3(41)	4(46)	9(56)	-
	4	3(34)	3(36)	4(42)	10(51)	-
	5	3(31)	3(33)	4(39)	10(47)	-

Table 4.16. The number of nonlinear iterations and GMRES iterations for different Reynolds numbers and partitions in lid-driven cavity problem discretized by Model 2 (continued).

		Re				
	overlap	1	10	100	1000	5000
2×4	0	3(114)	3(104)	5(122)	9(169)	-
	1	3(76)	3(72)	5(85)	9(109)	-
	2	3(60)	3(59)	5(69)	9(88)	-
	3	3(51)	3(51)	5(61)	9(77)	-
	4	3(47)	3(46)	5(54)	12(73)	-
	5	3(40)	3(42)	5(49)	9(64)	-
4×1	0	3(98)	3(91)	4(96)	-	-
	1	3(60)	3(55)	4(61)	17(91)	-
	2	3(48)	3(45)	4(48)	-	-
	3	3(41)	3(40)	4(41)	-	-
	4	3(37)	3(35)	4(37)	-	-
	5	3(34)	3(32)	4(34)	-	-
4×2	0	3(118)	3(111)	4(119)	9(181)	-
	1	3(74)	3(73)	4(81)	12(118)	-
	2	3(62)	3(58)	4(65)	12(93)	-
	3	3(53)	3(50)	4(57)	10(82)	-
	4	3(48)	3(45)	4(51)	8(71)	-
	5	3(43)	3(41)	4(48)	10(67)	-
4×4	0	3(126)	3(127)	5(144)	-	-
	1	3(86)	3(84)	5(96)	-	-
	2	3(69)	3(69)	5(79)	-	-
	3	3(59)	3(58)	5(67)	-	-
	4	3(51)	3(53)	5(60)	-	-
	5	3(47)	3(48)	5(55)	10(79)	-

Table 4.17. The effect of ϵ_{switch} on NEPIN algorithm for various Reynolds numbers in lid-driven cavity problem discretized by Model 2.

	Re				
ϵ_{switch}	1	10	100	1000	5000
100	5(69)	5(71)	7(81)	9(102)	21(130)
10	5(69)	5(71)	7(81)	9(102)	21(130)
1	5(69)	5(69)	7(81)	9(100)	21(131)
0.1	5(69)	5(69)	7(81)	9(100)	21(130)
0.01	5(69)	5(70)	7(81)	9(100)	21(130)

5. CONCLUSIONS AND FUTURE WORK

Finding fast, robust, parallel and scalable algorithms to solve large, sparse linear and nonlinear systems arising from the discretization of partial differential equations is an interesting area of research since the solution of such linear and nonlinear systems are required in many computational science and engineering application. In this conclusion chapter, we summarize our experiences and recommendations on solving large, sparse nonlinear systems and the nonlinear preconditioner algorithms. We also give some remarks for further research.

Nonlinear elimination preconditioned inexact Newton (NEPIN) method is more robust than other methods that we use since it converges for the problems discretizated by two models for steady-state incompressible lid-driven cavity problem with high Reynolds number. However, in most cases, the numbers of NEPIN iterations are larger than the numbers of additive Schwarz preconditioned inexact Newton (ASPIN) and field-split preconditioned inexact Newton (FSPIN) iterations. We should note that, the nonlinear system obtained by Model 2 discretization of the steady-state lid-driven cavity problem in velocity-vorticity formulation has not been solved by ASPIN and FSPIN methods before. We observe that, changing the discretization of one component on boundaries can influence the convergence of ASPIN severely such that it may fail for the problems at high Reynolds numbers.

The cost of left nonlinear preconditioners ASPIN and FSPIN is expensive than right nonlinear preconditioner NEPIN because of the construction of preconditioned nonlinear system in left nonlinear preconditioning. On the other hand, the cost of Newton-Krylov-Schwarz (NKS) method is the lowest compared with others. Thus, NKS should be the first choice for the problem it converges well.

According to the results on the effect of global linear tolerance and local nonlinear tolerance that we obtained, local nonlinear tolerance does not lead remarkable changes. On the other hand, global linear tolerance should be sufficiently small to have good

convergence with minimum number of nonlinear iterations. This is especially important in NEPIN algorithm since it may diverge for large global linear tolerance.

As globalization methods, line search algorithms improve ASPIN and FSPIN methods. However, employing line search worsen Newton's method and NEPIN algorithm in our experiences. It appears logical when we interpret the solution procedure that is explained in the previous chapter. However, our experiences do not match with the concept of globalization methods. Although we can not exploit the use of line search algorithms for Newton and NEPIN, they converged well without using a globalization method for the cases they are expected to converge.

One should be careful while partitioning the domain for ASPIN algorithm since it influences the construction of preconditioned function and the solution procedure severely. Large number of subdomains in regular quadrilateral partitioning increases the linear iterations in the solution of Jacobian system. In addition, it is observed that some cases with large number of subdomains have failed to converge for high Reynolds numbers. Thus, we recommend to use less number of subdomain as possible in ASPIN algorithm to have a good convergence in a short computation time. Overlap size should be chosen by ensuring the optimal reduction on the solution of Jacobian system without increasing the subdomain sizes too much. We should note that such a wide range of partitioning and overlap combinations like ours have not been tested before, and many other partitioning strategies with various overlap sizes can be investigated in further studies for ASPIN algorithm.

The selection of component to be eliminated in nonlinear elimination routine of NEPIN algorithm is important since it would affect the solution process severely. Some other components or a group of components may be tested. Furthermore, domain decomposition concept can be involved in the elimination routine such that some components in a certain subdomain can be selected to be eliminated.

In this thesis, we study only steady-state nonlinear systems. In transient nonlinear systems, the solution of the previous time step can be a good initial guess for

Newton's method to find the solution of the next time step. Therefore, there may be no need for nonlinear preconditioning. However, nonlinear preconditioners can still be employed for the cases where some time dependent strong source functions lead to slow convergence or failure in Newton's method at some time steps.

In the linear case, the implementation of right and left preconditioners is straightforward. In the nonlinear case, however, integration of right and left nonlinear preconditioners has not been tried before and the selection of preconditioners may results in various combinations. We believe that this might be a good research topic since nonlinear elimination routine on a nonlinearly preconditioned function can reduce the number of nonlinear iterations.

REFERENCES

1. Cai, X.-C. and D. E. Keyes, "Nonlinearly Preconditioned Inexact Newton Algorithms", *SIAM Journal Scientific Computing*, Vol. 24, No. 1, pp. 183–200, 2002.

2. Yang, H. and F.-N. Hwang, "An adaptive nonlinear elimination preconditioned inexact Newton algorithm for highly local nonlinear multicomponent PDE systems", *Applied Numerical Mathematics*, 2018.

3. Lions, P. L., *Interprétation stochastique de la methode alternée de Schwarz*, Ph.D. Thesis, R. Acad. Sci. Paris, 1978.

4. Smith, B., P. E. Bjorstad and W. D. Gropp, *Domain decomposition: parallel multilevel methods for elliptic partial differential equations*, Cambridge University Press, New York, 1996.

5. Saad, Y. and M. H. Schultz, "GMRES: a Generalized Minimal Residual Algorithm for Solving Nonsymmetric Linear Systems", *SIAM Journal on Scientific and Statistical Computing*, Vol. 7, No. 3, pp. 856–869, 1986.

6. Saad, Y., *Iterative methods for sparse linear systems*, SIAM, Philadelphia, 1995.

7. Chan, T. F. and K. R. Jackson, "Nonlinearly preconditioned Krylov subspace methods for discrete Newton algorithms", *SIAM Journal on Scientific and Statistical Computing*, Vol. 5, No. 3, pp. 533–542, 1984.

8. Cai, X.-C., D. E. Keyes and L. Marcinkowski, "Non-linear additive Schwarz preconditioners and application in computational fluid dynamics", *International Journal for Numerical Methods in Fluids*, Vol. 40, No. 12, pp. 1463–1470, 2002.

9. Hwang, F.-N. and X.-C. Cai, "A class of parallel two-level nonlinear Schwarz preconditioned inexact Newton algorithms", *Computer methods in applied mechanics*

and engineering, Vol. 196, No. 8, pp. 1603–1611, 2007.

10. Lanzkron, P. J., D. J. Rose and J. T. Wilkes, "An analysis of approximate nonlinear elimination", *SIAM Journal on Scientific Computing*, Vol. 17, No. 2, pp. 538–559, 1996.

11. Cai, X.-C. and X. Li, "Inexact Newton methods with restricted additive Schwarz based nonlinear elimination for problems with high local nonlinearity", *Siam Journal on Scientific Computing*, Vol. 33, No. 2, pp. 746–762, 2011.

12. Dembo, R. S., S. C. Eisenstat and T. Steihaug, "Inexact Newton Methods", *SIAM Journal on Numerical analysis*, Vol. 19, No. 2, pp. 400–408, 1982.

13. Eisenstat, S. C. and H. F. Walker, "Choosing the Forcing Terms in an Inexact Newton Method", *SIAM Journal on Scientific Computing*, Vol. 17, No. 1, pp. 16–32, 1996.

14. Dennis Jr, J. E. and R. B. Schnabel, *Numerical methods for unconstrained optimization and nonlinear equations*, SIAM, Philadelphia, 1996.

15. Knoll, D. A. and D. E. Keyes, "Jacobian-free Newton–Krylov methods: a survey of approaches and applications", *Journal of Computational Physics*, Vol. 193, No. 2, pp. 357–397, 2004.

16. Coleman, T. F. and J. J. Moré, "Estimation of sparse Jacobian matrices and graph coloring blems", *SIAM Journal on Numerical Analysis*, Vol. 20, No. 1, pp. 187–209, 1983.

17. Hwang, F.-N. and X.-C. Cai, "A parallel nonlinear additive Schwarz preconditioned inexact Newton algorithm for incompressible Navier–Stokes equations", *Journal of Computational Physics*, Vol. 204, No. 2, pp. 666–691, 2005.

18. Tuminaro, R. S., H. F. Walker and J. N. Shadid, "On backtracking failure in

Newton–GMRES methods with a demonstration for the Navier–Stokes equations", *Journal of Computational Physics*, Vol. 180, No. 2, pp. 549–558, 2002.

19. Hwang, F.-N., *Some Parallel Linear and Nonlinear Schwarz Methods with Applications in Computational Fluid Dynamics*, Ph.D. Thesis, University of Colorado at Boulder, 2004.

20. Cai, X.-C., W. D. Gropp, D. E. Keyes and *et al.*, "Newton-Krylov-Schwarz Methods in CFD", *Numerical methods for the Navier-Stokes equations*, pp. 17–30, 1994.

21. Gunzburger, M. D. and J. S. Peterson, "Predictor and steplength selection in continuation methods for the Navier-Stokes equations", *Computers & Mathematics with Applications*, Vol. 22, No. 8, pp. 73–81, 1991.

22. Dryja, M. and W. Hackbusch, "On the nonlinear domain decomposition method", *BIT Numerical Mathematics*, Vol. 37, No. 2, pp. 296–311, 1997.

23. Liu, L. and D. E. Keyes, "Field-split preconditioned inexact Newton algorithms", *SIAM Journal on Scientific Computing*, Vol. 37, No. 3, pp. A1388–A1409, 2015.

24. Prudencio, E. E., R. Byrd and X.-C. Cai, "Parallel full space SQP Lagrange–Newton–Krylov–Schwarz algorithms for PDE-constrained optimization problems", *SIAM Journal on Scientific Computing*, Vol. 27, No. 4, pp. 1305–1328, 2006.

25. Ghia, U., K. N. Ghia and C. T. Shin, "High-Re Solutions for Incompressible Flow Using the Navier-Stokes Equations and a Multigrid Method", *Journal of Computational Physics*, Vol. 48, No. 3, pp. 387–411, 1982.

26. Bezanson, J., A. Edelman, S. Karpinski and *et al.*, "Julia: A fresh approach to numerical computing", *SIAM Review*, Vol. 59, No. 1, pp. 65–98, 2017.

APPENDIX A: DERIVATION: VELOCITY-VORTICITY FORMULATION

The common used formulations of the Navier-Stokes equations are primitive formulation, streamfunction-vorticity formulation and pure streamfunction formulation. In literature, there are few studies with velocity-vorticity formulation of the Navier-Stokes equations than others. Thus, we show the derivation of the equations in velocity-vorticity formulation.

Consider the non-dimensional form of the Navier-Stokes equations in primitive formulation without body force term as shown below

$$\frac{\partial \vec{V}}{\partial t} + \vec{V}.\vec{\nabla}\vec{V} = -\vec{\nabla}p + \frac{1}{Re}\nabla^2\vec{V} \tag{A.1}$$

where Reynolds number is $Re = \frac{V_m L}{\nu}$ such that V_m is the mean velocity of the fluid, L is the length related with the geometry and ν is kinematic viscosity of a fluid. We get vorticity transport equation by taking the curl of Equation A.1. In the resultant equation, unknowns are velocity and vorticity components. The pressure term vanishes since $\vec{\nabla} \times \vec{\nabla}p = 0$.

$$\frac{\partial \omega}{\partial t} + \vec{V}.\vec{\nabla}\omega - \omega.\vec{\nabla}\vec{V} = \frac{1}{Re}\nabla^2\omega \tag{A.2}$$

In two dimension, the only existing components of velocity and vorticity are u, v and ω_k respectively. Therefore, $\omega.\vec{\nabla}\vec{V}$ vanishes. Moreover, since we consider consider the steady-state formulation of the equation, we neglect $\partial/\partial t$ term. Thus, Equation A.2 can be written as shown below.

$$\vec{V}.\vec{\nabla}\omega = \frac{1}{Re}\nabla^2\omega \tag{A.3}$$

After performing the dot product between velocity vector and the gradient of the vorticity, we get the steady-state vorticity transport equation in two dimension as shown below.

$$u\frac{\partial\omega}{\partial x} + v\frac{\partial\omega}{\partial y} - \frac{1}{Re}\nabla^2 w = 0 \tag{A.4}$$

By definition, vorticity is the curl of the velocity field such that $\vec{w} = \vec{\nabla} \times \vec{V}$. We obtain the forthcoming equation by taking the curl of the both sides of vorticity definition equation.

$$\nabla^2\vec{V} = -\vec{\nabla} \times \omega \tag{A.5}$$

Here, scalar ω represents the only vorticity component perpendicular to xy plane, in other words w_k. Note that, the equality $\vec{\nabla} \times (\vec{\nabla} \times \vec{V}) = \vec{\nabla}(\vec{\nabla}.\vec{V}) - \nabla^2\vec{V}$ is held and $\vec{\nabla}(\vec{\nabla}.\vec{V}) = 0$ due to incompressible assumption. We remind that Equation A.5 is in two dimensional space and it consists of two scalar equations. After equalization the terms that share same dimension in space, we obtain two equations such that one for u velocity component and one for v velocity component as follows.

$$\begin{aligned} -\nabla^2 u - \frac{\partial\omega}{\partial y} &= 0 \\ -\nabla^2 v + \frac{\partial\omega}{\partial x} &= 0 \end{aligned} \tag{A.6}$$