

**Krylov Subspace Methods for Simultaneous
Primal-Dual Solutions and Superconvergent
Functional Estimates**

by

James Lu

Submitted to the Department of Aeronautics and Astronautics
in partial fulfillment of the requirements for the degree of

Master of Science in Aeronautics and Astronautics

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

Sept 2002

© James Lu, MMII. All rights reserved.

The author hereby grants to MIT permission to reproduce and
distribute publicly paper and electronic copies of this thesis document
in whole or in part.

Author
Department of Aeronautics and Astronautics
August 23, 2002

Certified by
David L. Darmofal
Associate Professor
Thesis Supervisor

Accepted by
Edward M. Greitzer
Professor of Aeronautics and Astronautics
Chair, Committee on Graduate Students

Krylov Subspace Methods for Simultaneous Primal-Dual Solutions and Superconvergent Functional Estimates

by

James Lu

Submitted to the Department of Aeronautics and Astronautics
on August 23, 2002, in partial fulfillment of the
requirements for the degree of
Master of Science in Aeronautics and Astronautics

Abstract

This thesis is concerned with the iterative solution of non-symmetric linear (primal) systems and the dual systems associated with functional outputs of interest. The main goal of the work is to arrive at efficient Krylov subspace implementations for the situation where both primal and dual solutions are needed and accurate predictions for the functional outputs are desired.

A functional-output convergence result in the context of Krylov subspace methods is given. It is shown that with an appropriate choice of the starting shadow vector, methods based on the nonsymmetric Lanczos process obtain iterates giving superconvergent functional estimates. Furthermore, the adjoint solution may be obtained simultaneously with essentially no extra computational cost. The proposed methodology is demonstrated by the construction of a modified superconvergent variant of the QMR method.

The class of Krylov subspace methods based on Arnoldi process is also examined. Although by itself, superconvergence is not naturally present in the Arnoldi process, primal and dual problems may be coupled in a manner that allows for superconvergence. In particular, the proposed methodology is demonstrated by the construction of a primal-dual coupled superconvergent variant of the GMRES method.

Numerical experiments support the claim of superconvergence and demonstrate the viability of the proposed approaches.

Thesis Supervisor: David L. Darmofal
Title: Associate Professor

Acknowledgments

I would like to acknowledge the guidance and support of my thesis advisor, Professor David Darmofal, who provided many insightful ideas and suggestions for the work presented in this thesis. Thanks also goes to David Venditti for his assistance in providing the FUN2D test case.

Contents

1	Introduction	11
2	Functional-output Convergence Results	15
2.1	Krylov Subspace Methods	15
2.1.1	Subspace Correction Framework	15
2.1.2	Methods Based on Arnoldi Process	16
2.1.3	Methods Based on Lanczos Process	17
2.2	Adjoint Analysis	17
2.2.1	Krylov Subspace Output Convergence Results	19
3	Simultaneous Primal-Dual Superconvergent QMR	25
3.1	Nonsymmetric Lanczos Process	25
3.2	Conventional QMR	29
3.3	Modifications for Simultaneous Dual Solution	30
3.4	Preconditioning and Superconvergence	32
3.5	Algorithm Implementation	36
3.6	Numerical Experiments	40
3.6.1	Example 1	40
3.6.2	Example 2	41
4	Coupled Primal-Dual Superconvergent GMRES	45
4.1	Coupling Strategy for Superconvergence	46
4.2	Arnoldi Process and Equality Constraint	46

4.2.1	Algorithm Implementation	48
4.2.2	CSGMRES Algorithm	50
4.3	Numerical Experiments	52
4.3.1	Example 1	52
4.3.2	Example 2	53
5	Conclusions and Future Work	57
A	SSQMR MATLAB code	59
B	CSGMRES MATLAB code	65

List of Figures

3-1	Plots from SSQMR Example 1 : Poisson problem	43
3-2	Plots from SSQMR Example 2 : Euler flow	44
4-1	Plots from CSGMRES Example 1 : Poisson problem	54
4-2	Plots from CSGMRES Example 2 : Euler flow	55

Chapter 1

Introduction

The need for the solution of nonsymmetric linear systems arises in a large variety of situations. Among those is an important class that arise from the numerical solution of partial differential equations (PDEs) modelling physical problems. For PDEs that result from engineering analysis and design optimization, the principal quantities of interest are often functional-outputs of the solution. With appropriate linearization of the governing equation and the functional output, one often encounters linear systems for the primal variable \mathbf{x} of the form :

$$\mathbf{Ax} = \mathbf{b}, \tag{1.1}$$

and the principal quantity of interest is a linear functional of the form,

$$J^{\text{pr}}(\mathbf{x}) = \mathbf{g}^T \mathbf{x}. \tag{1.2}$$

Associated with the functional output of interest, one can consider the adjoint vector \mathbf{y} which is the solution to

$$\mathbf{A}^T \mathbf{y} = \mathbf{g}, \tag{1.3}$$

and the associated dual functional,

$$J^{\text{du}}(\mathbf{y}) = \mathbf{b}^T \mathbf{y}. \quad (1.4)$$

With the primal and dual functional defined as in (1.2) and (1.4), we have the following equivalence,

$$J^{\text{pr}}(\mathbf{A}^{-1}\mathbf{b}) = J^{\text{du}}(\mathbf{A}^{-T}\mathbf{g}). \quad (1.5)$$

The adjoint solution has found a multitude of uses in engineering computational simulations. For instance, the adjoint can be employed in design optimization to efficiently calculate the gradients of the outputs with respect to the control factors (Jameson [25], Reuther, Jameson and Alonso [34], Giles and Pierce [21], Elliot and Peraire [12], Anderson and Venkatakrisnan [3]). Furthermore, the adjoint solution can be used to estimate and control errors in functional outputs of computational simulations (Becker and Rannacher [6], Peraire and Patera [32], Giles and Pierce [33], Venditti and Darmofal [46, 47, 45]). Motivated by the numerous uses of the adjoint solution, this thesis is concerned with the situation where solutions for both the primal and dual problem are required.

The main contributions of this thesis are twofold. Firstly, the simultaneous iteration of both the primal and dual problems is demonstrated to have the benefit of superconvergence. Secondly, efficient implementations of coupled primal-dual solutions are presented.

Chapter 2 presents the underlying theory of Krylov subspace methods and develops the concept of superconvergent functional estimates in the context of iterative solution of linear systems. In Chapter 3, the nonsymmetric Lanczos process is presented where it is demonstrated that the dual solution may be simultaneously obtained with little additional computational costs and superconvergence may be brought out naturally; a modified superconvergent variant of the quasi-minimal resid-

ual method is presented. Finally, in Chapter 4, the Arnoldi process for constructing orthogonal bases is combined with superconvergence ideas to construct a smoothly converging primal-dual coupled superconvergent variant of the generalized minimal residual method.

Chapter 2

Functional-output Convergence Results

2.1 Krylov Subspace Methods

In Sect. 2.1.1 the general subspace correction framework is described. Several Krylov subspace methods are shown to lie in this framework, allowing the functional-output convergence for those methods to be analyzed in a unified way, given in Sect. 2.2.

2.1.1 Subspace Correction Framework

Consider a sequence of nested finite-dimensional correction and test subspaces,

$$\begin{aligned}\{\mathbf{0}\} &= \mathcal{C}_0 \subseteq \mathcal{C}_1 \subseteq \cdots \subseteq \mathcal{C}_N, \\ \{\mathbf{0}\} &= \mathcal{T}_0 \subseteq \mathcal{T}_1 \subseteq \cdots \subseteq \mathcal{T}_N,\end{aligned}\tag{2.1}$$

with the dimensions indexed by the subscripts. A subspace correction iterative method for the primal linear system (1.1) obtains iterates \mathbf{x}_n satisfying

$$\begin{aligned}\mathbf{x}_n &= \mathbf{x}_0 + \mathbf{c}_n, \quad \mathbf{c}_n \in \mathcal{C}_n, \\ \mathbf{r}_n^{\text{pr}}(\mathbf{x}_n) &\perp \mathbf{t}, \quad \forall \mathbf{t} \in \mathcal{T}_n,\end{aligned}\tag{2.2}$$

where \mathbf{x}_0 is the initial guess and

$$\mathbf{r}_n^{\text{pr}}(\mathbf{x}_n) \equiv \mathbf{b} - \mathbf{A}\mathbf{x}_n.\tag{2.3}$$

Krylov subspace methods lie in the general framework of subspace correction, limited mostly to cases $\mathcal{C}_n = \mathcal{K}_n(\mathbf{r}_0^{\text{pr}}, \mathbf{A})$, where

$$\mathcal{K}_n(\mathbf{r}_0^{\text{pr}}, \mathbf{A}) \equiv \text{span}\{\mathbf{r}_0^{\text{pr}}, \mathbf{A}\mathbf{r}_0^{\text{pr}}, \dots, \mathbf{A}^{n-1}\mathbf{r}_0^{\text{pr}}\},\tag{2.4}$$

with an exception to this rule being GMERR [49]. We now describe a few commonly used Krylov subspace methods that fit well into the subspace correction framework.

2.1.2 Methods Based on Arnoldi Process

The Arnoldi process generates an orthonormal basis of $\mathcal{K}_n(\mathbf{r}_0^{\text{pr}}, \mathbf{A})$ with the disadvantage of entailing linearly growing memory and computational costs per iteration. Amongst Krylov subspace methods, the most reknown are the Full Orthogonal Method (FOM) [36] and the more stable Generalized Minimal Residual Method (GMRES) [38]. Let us consider applying FOM and GMRES to the primal linear system (1.1). Table 2.1 summarizes the well-known properties of FOM and GMRES [11] in the framework of (2.2). We note that the GMRES selection of subspaces is equivalent to minimizing the residual in the case when $\mathbf{x}_n = \mathbf{x}_0 + \mathbf{c}_n$, $\mathbf{c}_n \in \mathcal{K}_n(\mathbf{r}_0^{\text{pr}}, \mathbf{A})$.

	FOM	GMRES
\mathcal{C}_n	$\mathcal{K}_n(\mathbf{r}_0^{\text{pr}}, \mathbf{A})$	$\mathcal{K}_n(\mathbf{r}_0^{\text{pr}}, \mathbf{A})$
\mathcal{T}_n	$\mathcal{K}_n(\mathbf{r}_0^{\text{pr}}, \mathbf{A})$	$\mathbf{A}\mathcal{K}_n(\mathbf{r}_0^{\text{pr}}, \mathbf{A})$

Table 2.1: Subspaces for FOM and GMRES

	BiCG	QMR
\mathcal{C}_n	$\mathcal{K}_n(\mathbf{r}_0^{\text{pr}}, \mathbf{A})$	$\mathcal{K}_n(\mathbf{r}_0^{\text{pr}}, \mathbf{A})$
\mathcal{T}_n	$\mathcal{K}_n(\mathbf{w}_1, \mathbf{A}^T)$	$\mathcal{U}_n(\mathbf{w}_1, \mathbf{\Omega}_n^{\text{pr}}, \mathbf{A}^T)$

Table 2.2: Subspaces for BiCG and QMR

2.1.3 Methods Based on Lanczos Process

Krylov subspace methods based on the non-symmetric Lanczos process requires two starting vectors. Applied to the primal linear system (1.1), this class of method is started by taking \mathbf{r}_0^{pr} and an arbitrary initial shadow vector \mathbf{w}_1 , the only requirement being $\mathbf{w}_1^T \mathbf{r}_0^{\text{pr}} \neq 0$. Biorthogonal bases for two Krylov subspaces are generated, although only one of which is explicitly used in constructing the iterates \mathbf{x}_n . The best-known methods based on the Lanczos process are Biconjugate Gradients (BiCG) [13] and more stable Quasi-Minimal Residual method (QMR) [18]. For the primal linear system (1.1), Table 2.2 summarizes the corresponding subspaces. Note that $\mathcal{U}_n(\mathbf{w}_1, \mathbf{\Omega}_n^{\text{pr}}, \mathbf{A}^T) \equiv \text{range}(\mathbf{W}_{n+1} \mathbf{D}_{n+1}^{-1} (\mathbf{\Omega}_n^{\text{pr}})^{-1} \mathbf{L}_n)$ is a n -dimensional subspace of $\mathcal{K}_{n+1}(\mathbf{w}_1, \mathbf{A}^T)$ and $\mathbf{W}_{n+1}, \mathbf{D}_{n+1}, \mathbf{\Omega}_n^{\text{pr}}, \mathbf{L}_n$ are matrices in the notation of [19]. Many transpose-free variants of BiCG and QMR have been constructed [16], some of which will be discussed later.

2.2 Adjoint Analysis

For the iterative solution of the linear system (1.1), we have the following a priori error norm estimate,

$$\|\mathbf{x} - \mathbf{x}_n\| \leq \|\mathbf{A}^{-1}\| \|\mathbf{r}_n^{\text{pr}}\|. \quad (2.5)$$

The above estimate is optimal in the sense that no higher exponent on the residual norm is possible. Thus, in general the norm of the solution error and arbitrary output functionals are expected to converge at a rate no higher than that of the residual. Under the special circumstances that certain functionals of the approximate solution converge at orders of the residual norm higher than that given by the global error estimate (2.5), those quantities are said to be superconverging. In particular, an iterative method is defined to be superconvergent if the superconvergence phenomenon can be demonstrated for arbitrary linear systems and linear functionals. This usage of the term superconvergence is a carry over of a concept usually used in contexts such as Galerkin finite element methods to describe the convergence of functional estimates [1, 48]. Note that this phenomenon is distinct from superconvergence as used in [8] to loosely describe the faster residual convergence when certain projected solutions are used as initial guess. The question of whether an iterative procedure is superconvergent may be settled using adjoint analysis. Denote $\bar{\mathbf{y}}$ as an arbitrary adjoint approximation and $\bar{\mathbf{r}}^{\text{du}}$ the corresponding dual residual defined by

$$\bar{\mathbf{r}}^{\text{du}} = \mathbf{g} - \mathbf{A}^T \bar{\mathbf{y}}. \quad (2.6)$$

The error in the functional-output provided by a primal iterate \mathbf{x}_n may be written as,

$$J^{\text{pr}}(\mathbf{x}) - J^{\text{pr}}(\mathbf{x}_n) = \bar{\mathbf{y}}^T \mathbf{r}_n^{\text{pr}} + (\bar{\mathbf{r}}^{\text{du}})^T (\mathbf{A}^{-1} \mathbf{r}_n^{\text{pr}}). \quad (2.7)$$

Using the triangle inequality and taking the infimum over all $\bar{\mathbf{y}}$, it follows that the error in the functional output provided by the primal iterates is bounded by the

following :

$$|J^{\text{Pr}}(\mathbf{x}) - J^{\text{Pr}}(\mathbf{x}_n)| \leq \min_{\bar{\mathbf{y}}} \left(|\bar{\mathbf{y}}^T \mathbf{r}_n^{\text{Pr}}| + \frac{\|\bar{\mathbf{r}}^{\text{du}}\| \|\mathbf{r}_n^{\text{Pr}}\|}{\sigma_{\min}} \right), \quad (2.8)$$

where σ_{\min} denotes the smallest singular value of \mathbf{A} , satisfying

$$\sigma_{\min} = \min_{\mathbf{z}, \mathbf{z} \neq \mathbf{0}} \frac{\|\mathbf{A}\mathbf{z}\|}{\|\mathbf{z}\|}. \quad (2.9)$$

As an immediate consequence, for general subspace correction methods (2.2),

$$|J^{\text{Pr}}(\mathbf{x}) - J^{\text{Pr}}(\mathbf{x}_n)| \leq \min_{\bar{\mathbf{y}} \in \mathcal{T}_n} \frac{\|\bar{\mathbf{r}}^{\text{du}}\| \|\mathbf{r}_n^{\text{Pr}}\|}{\sigma_{\min}}. \quad (2.10)$$

The above bound suggests that the convergence of functional estimates provided by subspace correction methods is controlled by the residual norm of the primal iterates as well as the closeness of approximation the sequence of test spaces for the primal problem, \mathcal{T}_n , provide for the adjoint solution.

2.2.1 Krylov Subspace Output Convergence Results

In this section, functional-output convergence results for the Krylov methods of Sect. 2.1 and Sect. 2.2 are obtained by applying the bound (2.10).

FOM In the FOM approach, the bound (2.10) gives,

$$|J_{\text{pr}}(\mathbf{x}) - J_{\text{pr}}(\mathbf{x}_n^{\text{Pr}})| \leq \min_{\bar{\mathbf{y}} \in \mathcal{K}_n(\mathbf{r}_0^{\text{Pr}}, \mathbf{A})} \frac{\|\bar{\mathbf{r}}^{\text{du}}\| \|\mathbf{r}_n^{\text{FOM,pr}}\|}{\sigma_{\min}}. \quad (2.11)$$

Thus, for FOM iterates the functional convergence rate can be determined by the approximation of the dual problem over the nested subspaces $\mathcal{K}_n(\mathbf{r}_0^{\text{Pr}}, \mathbf{A})$:

$$\|\bar{\mathbf{r}}_n^{\text{du}}\| \equiv \min_{\bar{\mathbf{y}} \in \mathcal{K}_n(\mathbf{r}_0^{\text{pr}}, \mathbf{A})} \|\mathbf{g} - \mathbf{A}^T \bar{\mathbf{y}}\|. \quad (2.12)$$

The above minimization problem is equivalent to the orthogonality condition,

$$\bar{\mathbf{r}}_n^{\text{du}} \perp \mathbf{A}^T \mathcal{K}_n(\mathbf{r}_0^{\text{pr}}, \mathbf{A}). \quad (2.13)$$

The fact that (2.13) implies (2.12) may be shown by the following argument. Denote $\bar{\mathbf{y}}_n$ as the minimizer corresponding to the minimal residual of (2.12) and $\bar{\mathbf{y}}$ to be an arbitrary vector in $\mathcal{K}_n(\mathbf{r}_0^{\text{pr}}, \mathbf{A})$. Then, we have

$$\begin{aligned} \|\mathbf{g} - \mathbf{A}^T \bar{\mathbf{y}}\|^2 &= \|\mathbf{g} - \mathbf{A}^T [(\bar{\mathbf{y}} - \bar{\mathbf{y}}_n) + \bar{\mathbf{y}}_n]\|^2 \\ &= \|\mathbf{g} - \mathbf{A}^T \bar{\mathbf{y}}_n\|^2 - 2(\mathbf{g} - \mathbf{A}^T \bar{\mathbf{y}}_n, \mathbf{A}^T (\bar{\mathbf{y}} - \bar{\mathbf{y}}_n)) \\ &\quad + \|\mathbf{A}^T (\bar{\mathbf{y}} - \bar{\mathbf{y}}_n)\|^2. \end{aligned} \quad (2.14)$$

By the orthogonality condition (2.13), the middle term of the above expression is zero. Hence, $\bar{\mathbf{y}}_n$ is the minimizer :

$$\|\mathbf{g} - \mathbf{A}^T \bar{\mathbf{y}}\|^2 \geq \|\mathbf{g} - \mathbf{A}^T \bar{\mathbf{y}}_n\|^2 \quad \forall \bar{\mathbf{y}} \in \mathcal{K}_n(\mathbf{r}_0^{\text{pr}}, \mathbf{A}). \quad (2.15)$$

(2.12) is equivalent to solving the dual problem within the framework of a family of generalized minimum error (GMERR) methods as described by Weiss [49]. Specifically, a GMERR method applied to the dual system finds iterates $\mathbf{y}_n^{\text{GMERR}}$ within the Krylov subspaces generated by \mathbf{A} ,

$$\mathbf{y}_n^{\text{GMERR}} \in \mathbf{y}_0 + \mathbf{A} \mathcal{K}_n(\mathbf{p}_0, \mathbf{A}), \quad (2.16)$$

where \mathbf{p}_0 is an arbitrary starting vector and $\mathbf{y}_n^{\text{GMERR}}$ is chosen so that the error norm is minimized,

$$\|\mathbf{y} - \mathbf{y}_n^{\text{GMERR}}\| = \min_{\bar{\mathbf{y}} \in \mathbf{y}_0 + \mathbf{A}\mathcal{K}_n(\mathbf{p}_0, \mathbf{A})} \|\mathbf{y} - \bar{\mathbf{y}}\|. \quad (2.17)$$

In a manner similar to (2.14), it may be shown that the minimization property (2.17) is equivalent to the error orthogonality condition,

$$\mathbf{y} - \mathbf{y}_n^{\text{GMERR}} \perp \mathbf{A}\mathcal{K}_n(\mathbf{p}_0, \mathbf{A}), \quad (2.18)$$

and hence to the residual orthogonality condition,

$$\mathbf{g} - \mathbf{A}^T \mathbf{y}_n^{\text{GMERR}} \perp \mathcal{K}_n(\mathbf{p}_0, \mathbf{A}). \quad (2.19)$$

Therefore, with the identification $\mathbf{p}_0 = \mathbf{A}^T \mathbf{r}_0^{\text{pr}}$ and $\mathbf{y}_0 = \mathbf{0}$, the approximation problem (2.12) is the same as (2.17). The convergence of GMERR for arbitrary \mathbf{p}_0 has not been shown. For the case that $\mathbf{p}_0 = \mathbf{g}$, it is known that the subspaces $\mathcal{K}_n(\mathbf{p}_0, \mathbf{A})$ form a sequence of convergent approximations for the adjoint solution if and only if \mathbf{A} is normal [10]. In fact, unless \mathbf{A} is normal and \mathbf{r}_0^{pr} related to \mathbf{g} in some way, the convergence of (2.12) is expected to be poor. Hence, functional estimates obtained from FOM iterates are expected to converge at a rate no higher than that of the residual norm.

GMRES For GMRES, the functional error bound (2.10) is :

$$|J_{\text{pr}}(\mathbf{x}) - J_{\text{pr}}(\mathbf{x}_n^{\text{GMRES}})| \leq \min_{\bar{\mathbf{y}} \in \mathbf{A}\mathcal{K}_n(\mathbf{r}_0^{\text{pr}}, \mathbf{A})} \frac{\|\bar{\mathbf{r}}^{\text{du}}\| \|\mathbf{r}_n^{\text{GMRES,pr}}\|}{\sigma_{\min}}. \quad (2.20)$$

Analogous to the FOM case discussed above, for GMRES iterates the convergence

rate of functional estimates is controlled by :

$$\|\bar{\mathbf{r}}_n^{\text{du}}\| \equiv \min_{\bar{\mathbf{y}} \in \mathbf{AK}_n(\mathbf{r}_0^{\text{pr}}, \mathbf{A})} \|\mathbf{g} - \mathbf{A}^T \bar{\mathbf{y}}\|. \quad (2.21)$$

In general, the above is not an improvement over (2.12). Hence, the conclusion is also that the functional estimates obtained from GMRES iterates is only linearly convergent with the residual norm.

BiCG For bi-conjugate gradients, the functional error bound is,

$$|J_{\text{pr}}(\mathbf{x}) - J_{\text{pr}}(\mathbf{x}_n^{\text{BiCG}})| \leq \min_{\bar{\mathbf{y}} \in \mathcal{K}_n(\mathbf{w}_1, \mathbf{A}^T)} \frac{\|\bar{\mathbf{r}}^{\text{du}}\| \|\mathbf{r}_n^{\text{BiCG,pr}}\|}{\sigma_{\min}}. \quad (2.22)$$

If GMRES is applied to the dual problem (1.3) with the zero vector as the initial guess, the dual residual norm is minimized,

$$\|\mathbf{r}_n^{\text{GMRES,du}}\| = \min_{\bar{\mathbf{y}} \in \mathcal{K}_n(\mathbf{g}, \mathbf{A}^T)} \|\mathbf{g} - \mathbf{A}^T \bar{\mathbf{y}}\|. \quad (2.23)$$

Hence, if the test space for BiCG is started with $\mathbf{w}_1 = \mathbf{g}$, we have the following bound,

$$|J_{\text{pr}}(\mathbf{x}) - J_{\text{pr}}(\mathbf{x}_n^{\text{BiCG}})| \leq \frac{\|\mathbf{r}_n^{\text{GMRES,du}}\| \|\mathbf{r}_n^{\text{BiCG,pr}}\|}{\sigma_{\min}}. \quad (2.24)$$

If \mathbf{A} is relatively well-behaved so that the eigenvalue convergence bound is descriptive, $\|\mathbf{r}_n^{\text{GMRES,du}}\|$ converges to zero at a rate bounded by $\|\mathbf{r}_n^{\text{BiCG,pr}}\|$. This shows that BiCG with $\mathbf{w}_1 = \mathbf{g}$ is superconvergent at double the residual convergence rate for the functional (1.2). For other choices of test space starting vectors, for example $\mathbf{w}_1 = \mathbf{r}_0^{\text{pr}}$ as is almost always used in implemenetations, superconvergence for J_{pr} does not hold. Numerical evidence for this behavior is shown in a future paper [29].

QMR Finally, we examine the convergence bound for QMR,

$$|J_{\text{pr}}(\mathbf{x}) - J_{\text{pr}}(\mathbf{x}_n^{\text{QMR}})| \leq \min_{\bar{\mathbf{y}}_n \in \mathcal{U}_n(\mathbf{w}_1, \mathbf{\Omega}_n^{\text{pr}}, \mathbf{A}^T)} \frac{\|\bar{\mathbf{r}}_n^{\text{du}}\| \|\mathbf{r}_n^{\text{QMR,pr}}\|}{\sigma_{\min}}. \quad (2.25)$$

Similar to the BiCG case, QMR is not superconvergent if the starting shadow vector for the non-symmetric Lanczos process $\mathbf{w}_1 \neq \mathbf{g}$. Convergence of functional estimates is expected to improve significantly if $\mathbf{w}_1 = \mathbf{g}$, as can be seen in some of the numerical examples presented in Chapter 3. Also in Chapter 3, a metric for quasi-residual minimization is chosen with demonstrated improved output convergence. Viewed in the framework introduced here, this is equivalent to modifying the subspace $\mathcal{U}_n(\mathbf{g}, \mathbf{\Omega}_n^{\text{pr}}, \mathbf{A}^T)$ through the choice of $\mathbf{\Omega}_n^{\text{pr}}$.

Chapter 3

Simultaneous Primal-Dual Superconvergent QMR

In this chapter, the QMR algorithm is discussed in detail with the view towards implementation, building on the more abstract description given in the previous chapter. Firstly, the underlying non-symmetric Lanczos process is presented, showing how the left shadow vector \mathbf{w}_1 may be chosen arbitrarily. Then, it is shown that besides the improved functional convergence as already demonstrated, choosing $\mathbf{w}_1 = \mathbf{g}$ has the additional advantage that the dual solution may be obtained simultaneously and at little extra cost using the same Lanczos process. Lastly, based on the adjoint analysis already presented, an approach to further improve functional convergence is developed.

3.1 Nonsymmetric Lanczos Process

The nonsymmetric Lanczos process constructs biorthogonal and \mathbf{A} -orthogonal basis vectors of Krylov subspaces which are then used in methods such as QMR. That is, it constructs vectors \mathbf{v}_j 's and \mathbf{w}_j 's that satisfy the biorthogonality condition

$$\mathbf{w}_m^T \mathbf{v}_n = \begin{cases} \delta_n, & m = n, \\ 0, & m \neq n. \end{cases} \quad (3.1)$$

and vectors \mathbf{p}_j and \mathbf{q}_j that satisfy the \mathbf{A} -orthogonality condition

$$\mathbf{q}_m^T \mathbf{A} \mathbf{p}_n = \begin{cases} \epsilon_n, & m = n, \\ 0, & m \neq n. \end{cases} \quad (3.2)$$

We use the Lanczos process based on coupled two-term recurrences rather than that based on three-term recursions as used in [18] since even though they are mathematically equivalent, the former is numerically more robust than the latter, as observed in [19]. Also, to simplify matters the look-ahead process [17] is not included.

- Initialization

- Normalize initial vectors \mathbf{v}_1 and \mathbf{w}_1 .
- Set $\mathbf{p}_0 = \mathbf{q}_0 = \mathbf{0}$, $\epsilon_0 = \rho_1 = \xi_1 = 1, n = 1$.

- At iteration n :

1. If $\epsilon_{n-1} = 0$, stop. Otherwise compute $\delta_n = \mathbf{w}_n^T \mathbf{v}_n$. If $\delta_n = 0$, then stop.
2. Update

$$\begin{aligned} \mathbf{p}_n &= \mathbf{v}_n - \mathbf{p}_{n-1}(\xi_n \delta_n / \epsilon_{n-1}), \\ \mathbf{q}_n &= \mathbf{w}_n - \mathbf{q}_{n-1}(\rho_n \delta_n / \epsilon_{n-1}) \end{aligned} \quad (3.3)$$

3. Compute

$$\begin{aligned}
\epsilon_n &= \mathbf{q}_n^T \mathbf{A} \mathbf{p}_n, \\
\beta_n &= \epsilon_n / \delta_n.
\end{aligned} \tag{3.4}$$

Update

$$\begin{aligned}
\tilde{\mathbf{v}}_{n+1} &= \mathbf{A} \mathbf{p}_n - \mathbf{v}_n \beta_n, \quad \rho_{n+1} = \|\tilde{\mathbf{v}}_{n+1}\|, \\
\tilde{\mathbf{w}}_{n+1} &= \mathbf{A}^T \mathbf{q}_n - \mathbf{w}_n \beta_n, \quad \xi_{n+1} = \|\tilde{\mathbf{w}}_{n+1}\|.
\end{aligned} \tag{3.5}$$

4. If $\rho_{n+1} = 0$ or $\xi_{n+1} = 0$, then stop. Else, update

$$\begin{aligned}
\mathbf{v}_{n+1} &= \tilde{\mathbf{v}}_{n+1} / \rho_{n+1}, \\
\mathbf{w}_{n+1} &= \tilde{\mathbf{w}}_{n+1} / \xi_{n+1}.
\end{aligned} \tag{3.6}$$

The result of the above iteration may be summarized compactly. Firstly, we introduce the notation

$$\begin{aligned}
\mathbf{V}_n &\equiv \begin{bmatrix} \mathbf{v}_1 & \mathbf{v}_2 & \cdots & \mathbf{v}_n \end{bmatrix}, \\
\mathbf{W}_n &\equiv \begin{bmatrix} \mathbf{w}_1 & \mathbf{w}_2 & \cdots & \mathbf{w}_n \end{bmatrix}, \\
\mathbf{P}_n &\equiv \begin{bmatrix} \mathbf{p}_1 & \mathbf{p}_2 & \cdots & \mathbf{p}_n \end{bmatrix}, \\
\mathbf{Q}_n &\equiv \begin{bmatrix} \mathbf{q}_1 & \mathbf{q}_2 & \cdots & \mathbf{q}_n \end{bmatrix}.
\end{aligned} \tag{3.7}$$

Then, it may be seen from the Lanczos process that the above satisfies

$$\mathbf{V}_n = \mathbf{P}_n \mathbf{U}_n,$$

$$\begin{aligned}
\mathbf{W}_n &= \mathbf{Q}_n \mathbf{\Gamma}_n^{-1} \mathbf{U}_n \mathbf{\Gamma}_n, \\
\mathbf{A} \mathbf{P}_n &= \mathbf{V}_{n+1} \mathbf{L}_n, \\
\mathbf{A}^T \mathbf{Q}_n &= \mathbf{W}_{n+1} \mathbf{\Gamma}_{n+1}^{-1} \mathbf{L}_n \mathbf{\Gamma}_n,
\end{aligned} \tag{3.8}$$

where the matrices $\mathbf{\Gamma}_n$, \mathbf{U}_n and \mathbf{L}_n are defined as

$$\mathbf{\Gamma}_n = \text{diag}(\gamma_1, \gamma_2, \dots, \gamma_n), \tag{3.9}$$

$$\mathbf{U}_n = \begin{bmatrix} 1 & \xi_2 \delta_2 / \epsilon_1 & 0 & \cdots & 0 \\ 0 & 1 & \xi_3 \delta_3 / \epsilon_2 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ 0 & \ddots & \ddots & \ddots & \xi_n \delta_n / \epsilon_{n-1} \\ 0 & \cdots & \cdots & 0 & 1 \end{bmatrix}, \tag{3.10}$$

$$\mathbf{L}_n = \begin{bmatrix} \beta_1 & 0 & 0 & \cdots & 0 \\ \rho_2 & \beta_2 & 0 & \cdots & 0 \\ 0 & \rho_3 & \beta_3 & \cdots & 0 \\ 0 & 0 & \rho_4 & \cdots & 0 \\ 0 & \ddots & \ddots & \ddots & \beta_n \\ 0 & \cdots & \cdots & 0 & \rho_{n+1} \end{bmatrix}, \tag{3.11}$$

the scalars $\xi_j, \delta_j, \epsilon_j, \rho_j$ are constants defined in the Lanczos process, and the constants γ_j satisfy the relation

$$\gamma_j = \begin{cases} 1 & j = 1, \\ \gamma_{j-1} \rho_j / \xi_j & 1 < j \leq n. \end{cases} \tag{3.12}$$

In the next section, we give a brief description of how QMR uses the Lanczos process to generate iterates that approximate the linear system (1.1).

3.2 Conventional QMR

At each iteration, QMR seeks an iterate \mathbf{x}_n^{pr} within the Krylov subspace

$$\mathbf{x}_n \in \mathbf{x}_0 + \text{span}\{\mathbf{r}_0^{\text{pr}}, \mathbf{A}\mathbf{r}_0^{\text{pr}}, \mathbf{A}^2\mathbf{r}_0^{\text{pr}}, \dots, \mathbf{A}^{n-1}\mathbf{r}_0^{\text{pr}}\}. \quad (3.13)$$

With the initial vector taken to be the normalized initial residual, $\mathbf{v}_1 \equiv \mathbf{r}_0^{\text{pr}}/\rho_1$, $\rho_1 \equiv \|\mathbf{r}_0^{\text{pr}}\|$, it may be seen from the Lanczos process that

$$\text{span}\{\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_n\} = \text{span}\{\mathbf{r}_0^{\text{pr}}, \mathbf{A}\mathbf{r}_0^{\text{pr}}, \mathbf{A}^2\mathbf{r}_0^{\text{pr}}, \dots, \mathbf{A}^{n-1}\mathbf{r}_0^{\text{pr}}\}. \quad (3.14)$$

Hence, from (3.13) \mathbf{x}_n^{pr} may be written as

$$\mathbf{x}_n = \mathbf{x}_0 + \mathbf{V}_n \mathbf{U}_n^{-1} \mathbf{z}_n, \quad (3.15)$$

where the matrices \mathbf{V}_n , \mathbf{U}_n are defined in (3.7) and (3.10) respectively and the vector \mathbf{z}_n is yet to be determined. Using the identities given in (3.8), it may be seen that

$$\begin{aligned} \mathbf{r}_n^{\text{pr}} &= \mathbf{b} - \mathbf{A}\mathbf{x}_n \\ &= \mathbf{r}_0^{\text{pr}} - \mathbf{V}_{n+1} \mathbf{L}_n \mathbf{z}_n. \end{aligned} \quad (3.16)$$

Using the fact that $\mathbf{v}_1 = \mathbf{r}_0^{\text{pr}}/\rho_1$, and introducing an $(n+1) \times (n+1)$ diagonal weight matrix $\mathbf{\Omega}_n^{\text{pr}}$

$$\mathbf{\Omega}_n^{\text{pr}} = \text{diag}(\omega_1^{\text{pr}}, \omega_2^{\text{pr}}, \dots, \omega_{n+1}^{\text{pr}}), \quad (3.17)$$

(3.16) may be written as

$$\mathbf{r}_n^{\text{pr}} = \mathbf{V}_{n+1} \left(\mathbf{\Omega}_n^{\text{pr}} \right)^{-1} \left(\rho_1 \omega_1^{\text{pr}} \mathbf{e}_1^{n+1} - \mathbf{\Omega}_n^{\text{pr}} \mathbf{L}_n \mathbf{z}_n \right). \quad (3.18)$$

Finally, \mathbf{z}_n is chosen so that the 2-norm of the quasi-residual is minimized :

$$\mathbf{z}_n = \arg \min_{\mathbf{z}} \left\| \rho_1 \omega_1^{\text{pr}} \mathbf{e}_1^{n+1} - \mathbf{\Omega}_n^{\text{pr}} \mathbf{L}_n \mathbf{z} \right\|. \quad (3.19)$$

Since the matrix \mathbf{L}_n has a bidiagonal structure, the above minimization may be done recursively, essentially performing QR decomposition of \mathbf{L}_n using successive Givens rotations.

The original QMR algorithm is formulated with a diagonal weighting matrix $\mathbf{\Omega}_n$ and convergence has been shown for arbitrary weights $\omega_j^{\text{pr}} \neq 0$ [18]. Extension of the weight matrix to block diagonal form having upper triangular blocks has also been done [41]. However, in practice the weight matrix is usually set to unity owing to the lack of a better choice. Moreover, in practice the two starting vectors are usually taken to be the same and the fact that the same Lanczos iteration contains a dual problem is not utilized. The desire of not using \mathbf{A}^T partly led to the development of transpose-free variants of QMR [14, 20].

3.3 Modifications for Simultaneous Dual Solution

With modifications to the conventional QMR algorithm, the solution of the dual problem may be found simultaneously with the primal problem using the same Lanczos process. Firstly, we observe that (3.8) implies not only

$$\mathbf{A}\mathbf{V}_n = \mathbf{V}_{n+1}\mathbf{L}_n\mathbf{U}_n, \quad (3.20)$$

but also

$$\mathbf{A}^T\mathbf{W}_n = \mathbf{W}_{n+1}\mathbf{\Gamma}_{n+1}^{-1}\mathbf{L}_n\mathbf{U}_n\mathbf{\Gamma}_n. \quad (3.21)$$

This suggests taking the normalized shadow starting vector \mathbf{w}_1 for the Lanczos process to be

$$\mathbf{w}_1 = \frac{\mathbf{g}}{\xi_1}, \quad \xi_1 = \|\mathbf{g}\|. \quad (3.22)$$

It may be observed that in the Lanczos process, given a certain \mathbf{v}_1 , the choice of \mathbf{w}_1 is arbitrary so long as $\mathbf{w}_1^T\mathbf{v}_1 \neq 0$. Only in rare circumstances will the non-orthogonality condition not be satisfied. Note that this is a departure from the standard implementation of QMR, which chooses the starting vector for the Lanczos iteration to be $\mathbf{w}_1 = \mathbf{v}_1$.

Similar to the primal iterates, we seek iterates for the dual problem of the form

$$\mathbf{y}_n = \mathbf{W}_n\mathbf{\Gamma}_n^{-1}\mathbf{U}_n^{-1}\mathbf{\Gamma}_n\mathbf{k}_n, \quad (3.23)$$

where the vector \mathbf{k}_n is yet to be determined. Then, the dual residual \mathbf{r}_n^{du} is of the form

$$\mathbf{r}_n^{\text{du}} = \mathbf{g} - \mathbf{A}^T\mathbf{y}_n$$

$$= \mathbf{W}_{n+1} \left(\boldsymbol{\Omega}_n^{\text{du}} \right)^{-1} \left(\xi_1 \omega_1^{\text{du}} \mathbf{e}_1^{n+1} - \boldsymbol{\Omega}_n^{\text{du}} \boldsymbol{\Gamma}_{n+1}^{-1} \mathbf{L}_n \boldsymbol{\Gamma}_n \mathbf{k}_n \right), \quad (3.24)$$

where the dual weight parameter matrix $\boldsymbol{\Omega}_n^{\text{du}}$ has been introduced. Analogous to the approach of QMR for the primal problem, \mathbf{k}_n is sought to minimize the 2-norm quasi-residual :

$$\mathbf{k}_n = \arg \min_{\mathbf{k}} \left\| \xi_1 \omega_1^{\text{du}} \mathbf{e}_1^{n+1} - \boldsymbol{\Omega}_n^{\text{du}} \boldsymbol{\Gamma}_{n+1}^{-1} \mathbf{L}_n \boldsymbol{\Gamma}_n \mathbf{k} \right\|. \quad (3.25)$$

Note that the term $\boldsymbol{\Gamma}_{n+1}^{-1} \mathbf{L}_n \boldsymbol{\Gamma}_n$ in (3.24) is the analog of \mathbf{L}_n in (3.16). In fact,

$$\boldsymbol{\Gamma}_{n+1}^{-1} \mathbf{L}_n \boldsymbol{\Gamma}_n = \begin{bmatrix} \beta_1 & 0 & \cdots & 0 \\ \xi_2 & \beta_2 & \cdots & 0 \\ 0 & \ddots & \ddots & 0 \\ 0 & \ddots & \xi_n & \beta_n \\ 0 & \cdots & 0 & \xi_{n+1} \end{bmatrix}, \quad (3.26)$$

and comparison with \mathbf{L}_n defined in (3.11) shows that $\boldsymbol{\Gamma}_{n+1}^{-1} \mathbf{L}_n \boldsymbol{\Gamma}_n$ is just \mathbf{L}_n with the replacement $\rho_j \rightarrow \xi_j$. Thus, the adjoint iterates \mathbf{y}_n may be obtained from the Lanczos vectors with the same procedure as the standard QMR method for the primal problem by replacing $\mathbf{v}_j \rightarrow \mathbf{w}_j$, $\mathbf{p}_j \rightarrow \mathbf{q}_j$ and $\rho_j \rightarrow \xi_j$.

3.4 Preconditioning and Superconvergence

For Krylov subspace methods, preconditioning is often necessary for fast and robust convergence. In this section, a superconvergence strategy for the preconditioned system is presented. It is shown that same strategy may be applied to both the primal and the dual system.

With the use of preconditioners \mathbf{M}_1 and \mathbf{M}_2 , the primal problem (1.1) effectively becomes

$$\begin{aligned}
\mathbf{A}'\mathbf{x}' &= \mathbf{b}', \quad \text{where } \mathbf{A}' \equiv \mathbf{M}_1^{-1}\mathbf{A}\mathbf{M}_2^{-1}, \\
\mathbf{x}' &\equiv \mathbf{M}_2\mathbf{x}, \\
\mathbf{b}' &\equiv \mathbf{M}_1^{-1}\mathbf{b}.
\end{aligned} \tag{3.27}$$

The value of the primal linear functional is required to be invariant under the preconditioning transformation, that is,

$$J^{\text{pr}}(\mathbf{x}) = \mathbf{g}^T \mathbf{x} = \mathbf{g}'^T \mathbf{x}', \tag{3.28}$$

thereby obtaining the expression for \mathbf{g}' ,

$$\mathbf{g}' = \mathbf{M}_2^{-T} \mathbf{g}. \tag{3.29}$$

Then, the adjoint for the preconditioned system, \mathbf{y}' , is just the solution to

$$\mathbf{A}'^T \mathbf{y}' = \mathbf{g}'. \tag{3.30}$$

From (3.30), it may be verified that

$$\mathbf{y}' = \mathbf{M}_1^T \mathbf{y}. \tag{3.31}$$

Furthermore, from (3.30) and (3.27),

$$\mathbf{b}^T \mathbf{y} = \mathbf{b}'^T \mathbf{y}'. \tag{3.32}$$

Equation (3.31) shows that the adjoint for the original system (1.3) may be recovered from that of the preconditioned system by simply multiplying with the matrix \mathbf{M}_1^{-T} at the end of the iterations. In what follows, we will work exclusively with the preconditioned systems.

As a means of improving the output superconvergence, we seek to minimize

$$(\check{\mathbf{y}}'_{n+1})^T \mathbf{r}'_n \quad (3.33)$$

where $\check{\mathbf{y}}'_{n+1}$ is an estimate of the adjoint available at the $(n+1)$ th iteration and \mathbf{r}'_n is the primal residual at the n th iteration, both for the preconditioned system. Note that the check mark on $\check{\mathbf{y}}'_{n+1}$ is used to differentiate the adjoint estimates that we use to determine weight parameters and the adjoint estimates carried forward in our simultaneous primal-dual method.

The above provides a natural scaling for the QMR weight parameters. Specifically, by letting,

$$\omega_j^{\text{pr}} = (\check{\mathbf{y}}'_{n+1})^T \mathbf{v}_j, \quad 1 \leq j \leq n+1, \quad (3.34)$$

then,

$$\begin{aligned} (\check{\mathbf{y}}'_{n+1})^T \mathbf{r}'_n &= (\check{\mathbf{y}}'_{n+1})^T \begin{bmatrix} \mathbf{v}_1 & \mathbf{v}_2 & \cdots & \mathbf{v}_{n+1} \end{bmatrix} \left(\mathbf{\Omega}_n^{\text{pr}} \right)^{-1} \left(\rho_1 \omega_1^{\text{pr}} \mathbf{e}_1^{(n+1)} - \mathbf{\Omega}_n^{\text{pr}} \mathbf{L}_n \mathbf{z}_n \right) \\ &= \begin{bmatrix} (\check{\mathbf{y}}'_{n+1})^T \mathbf{v}_1 & \cdots & (\check{\mathbf{y}}'_{n+1})^T \mathbf{v}_{n+1} \end{bmatrix} \left(\mathbf{\Omega}_n^{\text{pr}} \right)^{-1} \left(\rho_1 \omega_1^{\text{pr}} \mathbf{e}_1^{(n+1)} - \mathbf{\Omega}_n^{\text{pr}} \mathbf{L}_n \mathbf{z}_n \right) \\ &= \begin{bmatrix} 1 & 1 & \cdots & 1 \end{bmatrix} \left(\rho_1 \omega_1^{\text{pr}} \mathbf{e}_1^{(n+1)} - \mathbf{\Omega}_n^{\text{pr}} \mathbf{L}_n \mathbf{z}_n \right). \end{aligned} \quad (3.35)$$

Thus, with the ω_j^{pr} as chosen in (3.34), the linear error term (3.33) in the linear functional is approximately equal to the sum of the entries of the primal quasi-residual. By selecting \mathbf{z}_n to minimize this weighted primal quasi-residual, the primal iterates

obtained from our modified QMR approach should more accurately approximate the linear functional.

However, the scheme as described above could suffer from non-recursivity because at each iteration n , all weights $\omega_j^{\text{pr}}, 1 \leq j \leq n$ must be updated and all prior vectors might need to be stored. To maintain recursivity, we introduce a truncated approximation to the adjoint, $\check{\mathbf{y}}'_{i,n+1}$, where the index i is an approximation parameter. We define $\check{\mathbf{y}}'_{i,n+1}$ as

$$\check{\mathbf{y}}'_{i,n+1} \equiv \left(\frac{(\check{\mathbf{Y}}'_{i+1})^T \mathbf{v}_1}{\mathbf{w}_1^T \mathbf{v}_1} \right) \mathbf{w}_1 + \left(\frac{(\check{\mathbf{Y}}'_{i+2})^T \mathbf{v}_2}{\mathbf{w}_2^T \mathbf{v}_2} \right) \mathbf{w}_2 + \cdots + \left(\frac{(\check{\mathbf{Y}}'_{i+n+1})^T \mathbf{v}_{n+1}}{\mathbf{w}_{n+1}^T \mathbf{v}_{n+1}} \right) \mathbf{w}_{n+1}, \quad (3.36)$$

where we take $\check{\mathbf{Y}}'_i$ to be the adjoint iterate from the standard QMR algorithm. Although other choices may be thought of, using this truncated approximation, we have

$$(\check{\mathbf{y}}'_{i,n+1})^T \mathbf{v}_j = (\check{\mathbf{Y}}'_{i+j})^T \mathbf{v}_j. \quad (3.37)$$

Associated with a certain choice of i is the storage needed for the i vectors $[\mathbf{p}_1 \cdots \mathbf{p}_i]$, each of the dimension of problem size. However, no extra computational work is required. For the numerical experiments considered here, a small i (≈ 3) works well enough.

With $\check{\mathbf{y}}'_{i,n+1}$ as defined in (3.37), we take the weight parameters to be

$$\begin{aligned} \omega_j^{\text{pr}} &= (\check{\mathbf{y}}'_{i,n+1})^T \mathbf{v}_j \\ &= (\check{\mathbf{Y}}'_{i+j})^T \mathbf{v}_j. \end{aligned} \quad (3.38)$$

A similar strategy of weight parameter determination may be done for the dual problem. Analogous to (3.38), we take the dual weights to be

$$\omega_j^{\text{du}} = (\check{\mathbf{X}}'_{i+j})^T \mathbf{w}_j, \quad (3.39)$$

where similarly, $\check{\mathbf{X}}'_n$ is the primal solution estimate obtained at iteration n using the QMR method.

It is to be noted that although (3.38) and (3.39) imply forming the QMR iterates $\check{\mathbf{X}}'_n$ and $\check{\mathbf{Y}}'_n$ and performing inner products for the calculation of each weight, this is not necessary. Instead, they are equivalently but cheaply calculated from scalars obtained in the i Lanczos iterations ahead of the current primal and dual iterates.

3.5 Algorithm Implementation

In this section, an implementation of the superconvergent simultaneous primal-dual QMR (SSQMR) method is described. The Lanczos iteration is carried i steps ahead of both the primal and dual iterates. Compared to conventional QMR, it requires a constant extra storage of two sets of vectors $\mathbf{P} = [\mathbf{p}_1 \mathbf{p}_2 \cdots \mathbf{p}_i]$ and $\mathbf{Q} = [\mathbf{q}_1 \mathbf{q}_2 \cdots \mathbf{q}_i]$ where each vector is of the same size as \mathbf{b} . Additionally, three sets of i scalars need to be stored : $\boldsymbol{\beta} = [\beta_1 \beta_2 \cdots \beta_i]$, $\boldsymbol{\rho} = [\rho_1 \rho_2 \cdots \rho_i]$ and $\boldsymbol{\xi} = [\xi_1 \xi_2 \cdots \xi_i]$. After convergence by some criteria, the primal and dual solutions for the original systems are recovered by observing (3.27) and (3.31). The algorithm for SSQMR is given below; MATLAB code is given in Appendix A.

Algorithm with Lanczos Forward Index i :

- Initialization
 - Set initial guesses to be the zero vector : $\mathbf{x}_0^{\text{pr}} = \mathbf{x}_0^{\text{du}} = \mathbf{0}$.
 - Let $\rho_1 = \|\mathbf{M}_1^{-1}\mathbf{b}\|$, $\mathbf{v}_1 = \mathbf{M}_1^{-1}\mathbf{b}/\rho_1$. Let $\xi_1 = \|\mathbf{M}_2^{-T}\mathbf{g}\|$, $\mathbf{w}_1 = \mathbf{M}_2^{-T}\mathbf{g}/\xi_1$.
 - Check that $\mathbf{w}_1^T \mathbf{v}_1 \neq 0$, otherwise restart.
 - Set $\mathbf{p}_0 = \mathbf{q}_0 = \mathbf{d}_0^{\text{pr}} = \mathbf{d}_0^{\text{du}} = \mathbf{0}$.

- Set $c_0^{\text{pr}} = c_0^{\text{du}} = \epsilon_0 = 1$, $\vartheta_0^{\text{pr}} = \vartheta_0^{\text{du}} = 0$, $\eta_0^{\text{pr}} = \eta_0^{\text{du}} = -1$.
- Set $w_{\text{-}}c_0^{\text{pr}} = w_{\text{-}}c_0^{\text{du}} = 1$, $w_{\text{-}}\vartheta_0^{\text{pr}} = w_{\text{-}}\vartheta_0^{\text{du}} = 0$, $w_{\text{-}}\eta_0^{\text{pr}} = w_{\text{-}}\eta_0^{\text{du}} = -1$.
- Initialize vector storage $\mathbf{P} = [\mathbf{p}_1 \mathbf{p}_2 \cdots \mathbf{p}_i]$ and $\mathbf{Q} = [\mathbf{q}_1 \mathbf{q}_2 \cdots \mathbf{q}_i]$, scalar storage $\boldsymbol{\beta} = [\beta_1 \beta_2 \cdots \beta_i]$, $\boldsymbol{\rho} = [\rho_1 \rho_2 \cdots \rho_i]$ and $\boldsymbol{\xi} = [\xi_1 \xi_2 \cdots \xi_i]$.
- Set counter values for weight parameter accumulation :

$$k_0 = 1$$

FOR $h = 1 : i$ DO

$$m(h) = 2 - h$$

END

- FOR $n = 1, 2, 3, \dots$, DO

1. If $\epsilon_n = 0$ or $\delta_n = 0$, stop. Otherwise, compute $\delta_n = \mathbf{w}_n^T \mathbf{v}_n$.
2. Update counter $k_n = (k_n \bmod i) + 1$. Update vectors

$$\begin{aligned} \mathbf{p}_n &= \mathbf{v}_n - \mathbf{p}_{n-1}(\xi_n \delta_n / \epsilon_{n-1}), \\ \mathbf{q}_n &= \mathbf{w}_n - \mathbf{q}_{n-1}(\rho_n \delta_n / \epsilon_{n-1}). \end{aligned} \quad (3.40)$$

Store $\mathbf{P}(:, k_n) = \mathbf{p}_n$, $\mathbf{Q}(:, k_n) = \mathbf{q}_n$.

3. Compute $\tilde{\mathbf{p}}_n = \mathbf{A}(\mathbf{M}_2^{-1} \mathbf{p}_n)$ and $\tilde{\mathbf{q}}_n = \mathbf{M}_1^{-T} \mathbf{q}_n$. Update $\epsilon_n = \tilde{\mathbf{q}}_n^T \tilde{\mathbf{p}}_n$, and set $\beta_n = \epsilon_n / \delta_n$.

Update

$$\begin{aligned} \tilde{\mathbf{v}}_{n+1} &= \mathbf{M}_1^{-1} \tilde{\mathbf{p}}_n - \beta_n \mathbf{v}_n, \\ \tilde{\mathbf{w}}_{n+1} &= \mathbf{M}_2^{-T} (\mathbf{A}^T \tilde{\mathbf{q}}_n) - \beta_n \mathbf{w}_n. \end{aligned} \quad (3.41)$$

Update $\rho_{n+1} = \|\tilde{\mathbf{v}}_{n+1}\|$, $\xi_{n+1} = \|\tilde{\mathbf{w}}_{n+1}\|$. Store $\boldsymbol{\beta}(k_n) = \beta_n$, $\boldsymbol{\rho}(k_n) = \rho_n$, $\boldsymbol{\xi}(k_n) = \xi_n$

If $\rho_{n+1} \neq 0$ and $\xi_{n+1} \neq 0$, update

$$\begin{aligned}
\mathbf{v}_{n+1} &= \tilde{\mathbf{v}}_{n+1}/\rho_{n+1}, \\
\mathbf{w}_{n+1} &= \tilde{\mathbf{w}}_{n+1}/\xi_{n+1}.
\end{aligned} \tag{3.42}$$

4. Update

$$\begin{aligned}
w_{\vartheta_n^{\text{pr}}} &= \frac{\rho_{n+1}}{w_{\mathcal{C}_{n-1}^{\text{pr}}}\beta_n}, w_{\mathcal{C}_n^{\text{pr}}} = \frac{1}{\sqrt{1 + (w_{\vartheta_n^{\text{pr}}})^2}}, \\
w_{\eta_n^{\text{pr}}} &= -w_{\eta_{n-1}^{\text{pr}}} \frac{\rho_n (w_{\mathcal{C}_n^{\text{pr}}})^2}{\beta_n (w_{\mathcal{C}_{n-1}^{\text{pr}}})^2} \\
w_{\vartheta_n^{\text{du}}} &= \frac{\xi_{n+1}}{w_{\mathcal{C}_{n-1}^{\text{du}}}\beta_n}, w_{\mathcal{C}_n^{\text{du}}} = \frac{1}{\sqrt{1 + (w_{\vartheta_n^{\text{du}}})^2}}, \\
w_{\eta_n^{\text{du}}} &= -w_{\eta_{n-1}^{\text{du}}} \frac{\xi_n (w_{\mathcal{C}_n^{\text{du}}})^2}{\beta_n (w_{\mathcal{C}_{n-1}^{\text{du}}})^2}
\end{aligned} \tag{3.43}$$

5. Weight parameters $\omega_{n-i+1}^{\text{pr}}$ and $\omega_{n-i+1}^{\text{du}}$ calculation

FOR $h = 1 : i$ DO

IF $m(h) = 1$ THEN

$$w_{\mathcal{Q}^{\text{pr}}}(h) = \delta_n, w_{\mathcal{D}^{\text{pr}}}(h) = 0, w_{\mathcal{X}^{\text{pr}}}(h) = 0$$

$$w_{\mathcal{Q}^{\text{du}}}(h) = \delta_n, w_{\mathcal{D}^{\text{du}}}(h) = 0, w_{\mathcal{X}^{\text{du}}}(h) = 0$$

END

IF $1 < m(h) < i$ THEN

$$\text{IF } n \neq 1 \text{ THEN } w_{\mathcal{Q}^{\text{pr}}}(h) = -(\xi_n \delta_n / \epsilon_{n-1}) w_{\mathcal{Q}^{\text{pr}}}(h)$$

$$w_{\mathcal{D}^{\text{pr}}}(h) = w_{\eta_n^{\text{pr}}} w_{\mathcal{Q}^{\text{pr}}}(h) + w_{\mathcal{D}^{\text{pr}}}(h) \left(w_{\vartheta_{n-1}^{\text{pr}}} w_{\mathcal{C}_n^{\text{pr}}} \right)^2$$

$$w_{\mathcal{X}^{\text{pr}}}(h) = w_{\mathcal{X}^{\text{pr}}}(h) + w_{\mathcal{D}^{\text{pr}}}(h)$$

$$\text{IF } n \neq 1 \text{ THEN } w_{\mathcal{Q}^{\text{du}}}(h) = -(\rho_n \delta_n / \epsilon_{n-1}) w_{\mathcal{Q}^{\text{du}}}(h)$$

$$w_{\mathcal{D}^{\text{du}}}(h) = w_{\eta_n^{\text{du}}} w_{\mathcal{Q}^{\text{du}}}(h) + w_{\mathcal{D}^{\text{du}}}(h) \left(w_{\vartheta_{n-1}^{\text{du}}} w_{\mathcal{C}_n^{\text{du}}} \right)^2$$

$$w_{\mathcal{X}^{\text{du}}}(h) = w_{\mathcal{X}^{\text{du}}}(h) + w_{\mathcal{D}^{\text{du}}}(h)$$

END

IF $m(h) = i - 1$ THEN

$$m(h) = 2 - i$$

$$\omega_{n-i+1}^{\text{pr}} = w_{\mathcal{X}}^{\text{du}}(h)$$

$$\omega_{n-i+1}^{\text{du}} = w_{\mathcal{X}}^{\text{pr}}(h)$$

END

END

6. IF $n \geq i$ THEN

$$\begin{aligned}
\vartheta_{n-i}^{\text{pr}} &= \frac{\omega_{n-i+1}^{\text{pr}} \boldsymbol{\rho}((k_n \bmod i) + 1)}{\omega_{n-i}^{\text{pr}} c_{n-i-1}^{\text{pr}} \boldsymbol{\beta}((k_n \bmod i) + 1)} \\
c_{n-i}^{\text{pr}} &= \frac{1}{\sqrt{1 + (\vartheta_{n-i}^{\text{pr}})^2}} \\
\eta_{n-i}^{\text{pr}} &= -\eta_{n-i-1}^{\text{pr}} \frac{\boldsymbol{\rho}((k_n \bmod i) + 1) (c_{n-i}^{\text{pr}})^2}{\boldsymbol{\beta}((k_n \bmod i) + 1) (c_{n-i-1}^{\text{pr}})^2} \\
\mathbf{d}_{n-i}^{\text{pr}} &= \eta_{n-i}^{\text{pr}} \mathbf{P}(:, (k_n \bmod i) + 1) + \left(\vartheta_{n-i-1}^{\text{pr}} c_{n-i}^{\text{pr}} \right)^2 \mathbf{d}_{n-i-1}^{\text{pr}} \\
\mathbf{x}_{n-i}^{\text{pr}} &= \mathbf{x}_{n-i-1}^{\text{pr}} + \mathbf{d}_{n-i}^{\text{pr}} \\
\omega_{n-i}^{\text{pr}} &= \omega_{n-i+1}^{\text{pr}} \\
\\
\vartheta_{n-i}^{\text{du}} &= \frac{\omega_{n-i+1}^{\text{du}} \boldsymbol{\xi}((k_n \bmod i) + 1)}{\omega_{n-i}^{\text{du}} c_{n-i-1}^{\text{du}} \boldsymbol{\beta}((k_n \bmod i) + 1)} \\
c_{n-i}^{\text{du}} &= \frac{1}{\sqrt{1 + (\vartheta_{n-i}^{\text{du}})^2}} \\
\eta_{n-i}^{\text{du}} &= -\eta_{n-i-1}^{\text{du}} \frac{\boldsymbol{\xi}((k_n \bmod i) + 1) (c_{n-i}^{\text{du}})^2}{\boldsymbol{\beta}((k_n \bmod i) + 1) (c_{n-i-1}^{\text{du}})^2} \\
\mathbf{d}_{n-i}^{\text{du}} &= \eta_{n-i}^{\text{du}} \mathbf{Q}(:, (k_n \bmod i) + 1) + \left(\vartheta_{n-i-1}^{\text{du}} c_{n-i}^{\text{du}} \right)^2 \mathbf{d}_{n-i-1}^{\text{du}} \\
\mathbf{x}_{n-i}^{\text{du}} &= \mathbf{x}_{n-i-1}^{\text{du}} + \mathbf{d}_{n-i}^{\text{du}} \\
\omega_{n-i}^{\text{du}} &= \omega_{n-i+1}^{\text{du}}
\end{aligned} \tag{3.44}$$

END

7. FOR $h = 1 : i$ DO

$$m(h) = m(h) + 1$$

END

- Obtain primal and dual solutions for the original systems

$$\begin{aligned} \text{Primal solution} &= (\mathbf{M}_2)^{-1} \mathbf{x}_n^{\text{pr}}, \\ \text{Dual solution} &= (\mathbf{M}_1)^{-T} \mathbf{x}_n^{\text{du}}. \end{aligned} \tag{3.45}$$

3.6 Numerical Experiments

The proposed method with weight parameters determined using equations (3.38) and (3.39) is denoted as Superconvergent Simultaneous QMR (SSQMR). The method which solves both the primal and the dual problem but with all weight parameters set to unity is denoted as Simultaneous QMR (SQMR). The conventional QMR method has unity weights and initial vectors are chosen to be the same : $\mathbf{w}_1 = \mathbf{v}_1$. QMR is separately applied to the primal and dual problems and the results are compared to SSQMR and SQMR.

For all the examples shown here, the Lanczos forward index i for SSQMR is taken to be 3, since this has been shown to give good results. Also, the initial guesses \mathbf{x}'_0 and \mathbf{y}'_0 for all the methods are chosen to be the zero vector. No convergence criteria is used; rather, the number of iteration steps is specified in each case. The test problems are not excessively large so that the solutions for the linear systems are available up to level of round-off.

3.6.1 Example 1

In this example, we consider the second order finite difference discretization on a 51×51 grid of

$$\nabla^2 u = \frac{1}{\pi} \exp^{-(x+2)^2 - (y-1/2)^2}, \quad \Omega \in [0, 1] \times [0, 1],$$

$$u|_{\Gamma} = 0, \tag{3.46}$$

resulting in the matrix \mathbf{A} with 12205 non-zero elements. The linear functional is the discretized approximation of

$$\int_0^1 \int_0^1 u(x, y) \sin(\pi x) \sin(\pi y) dx dy. \tag{3.47}$$

The left and right preconditioners were obtained from the incomplete LU factorization of \mathbf{A} using a drop tolerance of 2×10^{-2} , resulting in \mathbf{L} and \mathbf{U} having 12059 and 12057 non-zero elements respectively.

From Figure 3-1 it may be seen that the primal and dual functional estimates obtained using standard QMR converge at roughly the same rate as the respective residual norms, showing that it is not superconvergent. SQMR obtains better functional estimates as a result of the better choice of starting vectors. In fact, the method appears to exhibit superconvergence. However, the behavior is not very consistent. On the other hand, SSQMR obtains functional estimates that are consistently superconverging at twice the order of residual convergence and improvement in functional estimates over SQMR is clearly seen.

3.6.2 Example 2

In this example, the linear system considered is one arising from a first order backward Euler implicit scheme for solving a first order upwind discretization of the two-dimensional compressible flow Euler equations using the NASA Langley unstructured flow solver FUN2D [2]. The specific problem is the transonic flow around the NACA 0012 airfoil (freestream Mach number of 0.8 and 1.25 degrees angle of attack). The mesh is composed of 1980 triangular elements with 990 nodes. The linear output is the airfoil drag functional linearized about the current iterate.

The matrix \mathbf{A} has 108532 non-zero entries. The left and right preconditioners were obtained from the incomplete LU factorization of \mathbf{A} using a drop tolerance of 10^{-2} ,

resulting in \mathbf{L} and \mathbf{U} having 88253 and 93588 non-zero elements respectively. Again, from Figure 3-2 we observe that the functional error convergence slopes of SSQMR are roughly twice that of conventional QMR, confirming the prediction made regarding superconvergence. In this example, the functional error of SQMR is close to that of SSQMR, showing that the unity parameter happens to be quite a good choice in this case. Still, SSQMR consistently gives better functional estimates than SQMR.

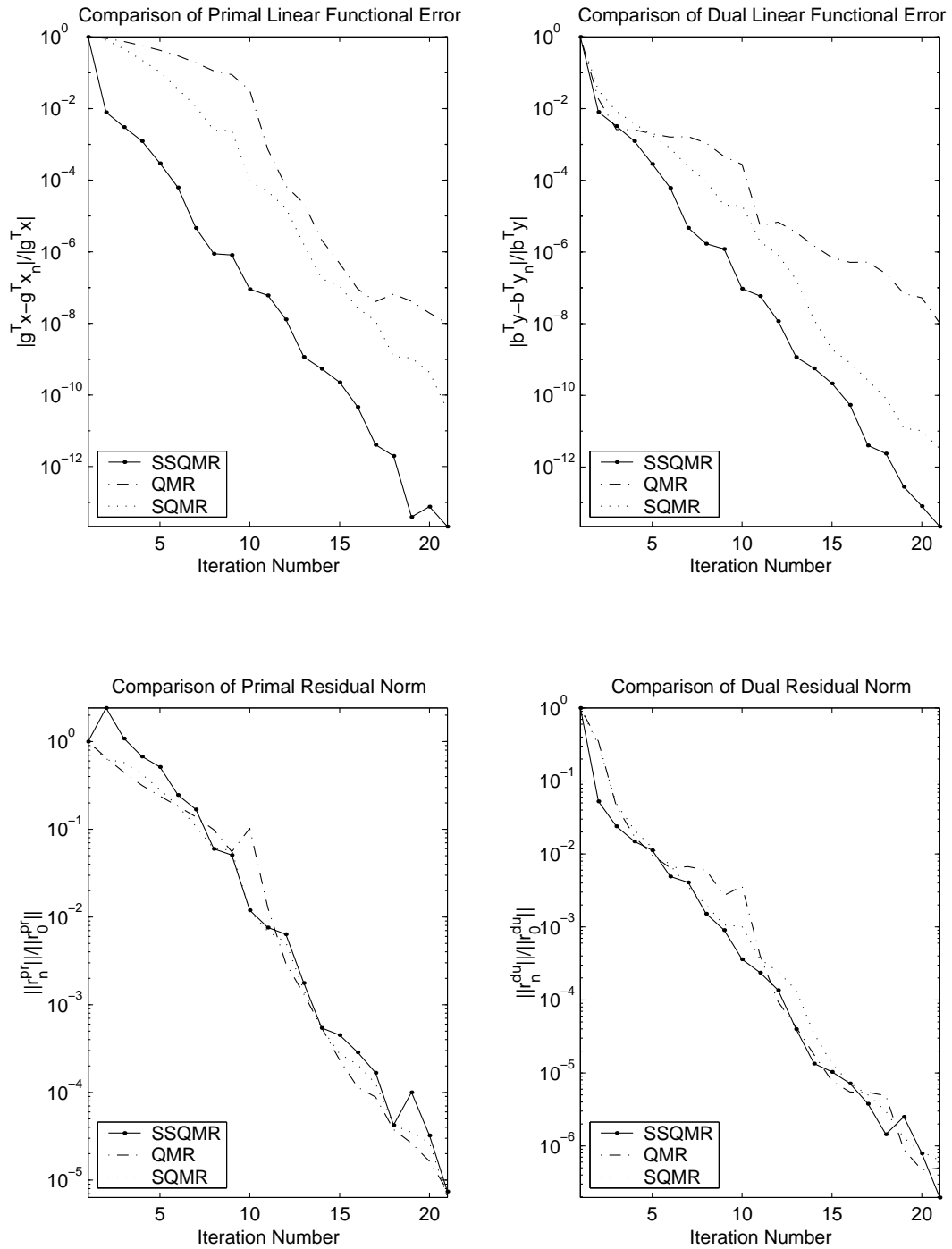


Figure 3-1: Plots from SSQMR Example 1 : Poisson problem

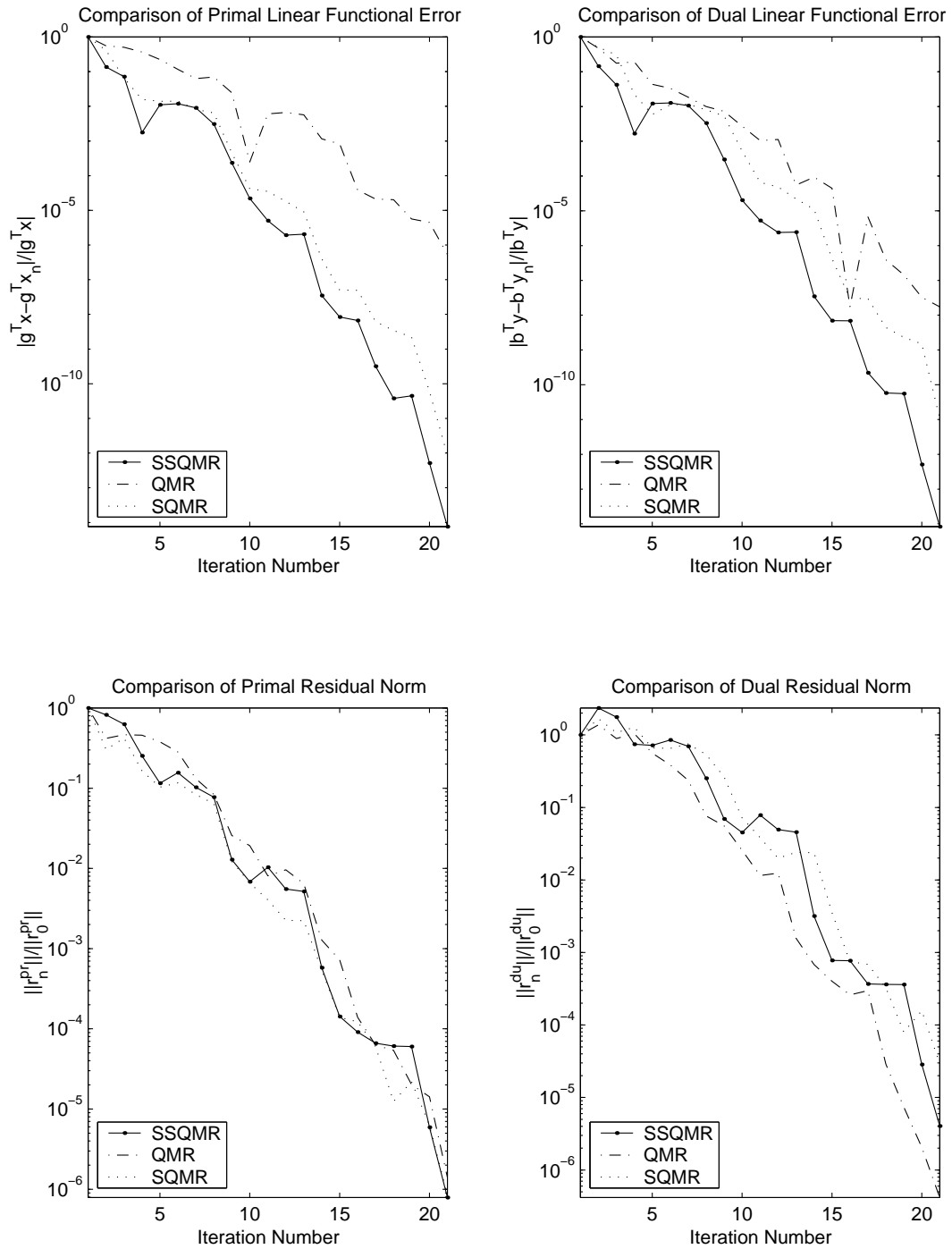


Figure 3-2: Plots from SSQMR Example 2 : Euler flow

Chapter 4

Coupled Primal-Dual

Superconvergent GMRES

GMRES [38] has found widespread use because of its robustness and its residual minimization property. However, as demonstrated in Chapter 2, it is not superconvergent since the sequence of test spaces \mathcal{T}_n do not approximate the adjoint well.

A way to make the adjoint Krylov subspace available is to iterate the dual problem in parallel with the primal. However, taking the adjoint Krylov subspace as the test space for the primal problem could result in the loss of desirable properties such as smooth, robust residual convergence. Hence, the approach proposed here is to use the same test space as conventional GMRES, but seek the norm-minimizing primal iterate within the correction space whose residual is orthogonal to the adjoint approximation of the previous iteration. In this way, the smooth residual convergence behavior of GMRES is largely retained while attaining superconvergence. The same strategy described above for the primal problem is also applied to the adjoint.

In the following section, a primal-dual coupling strategy is proposed. Then, casting the Krylov subspace constrained minimization problem in the form of Arnoldi vectors, it is shown how the modification for superconvergence involves solving an equality constrained least squares problem. Finally, an algorithm implementation for the coupled superconvergent GMRES (CSGMRES) is given.

4.1 Coupling Strategy for Superconvergence

Consider the primal and dual system, (1.1) and (1.3). Applied to the primal and dual problem, GMRES at every iteration count n solves the n -dimensional linear least squares problems for the respective iterates,

$$\begin{aligned}\mathbf{x}_n^{\text{GMRES}} &= \arg \min_{\bar{\mathbf{x}} \in \mathcal{C}_n^{\text{pr}}} \|\mathbf{b} - \mathbf{A}\bar{\mathbf{x}}\|, \\ \mathbf{y}_n^{\text{GMRES}} &= \arg \min_{\bar{\mathbf{y}} \in \mathcal{C}_n^{\text{du}}} \|\mathbf{g} - \mathbf{A}^T \bar{\mathbf{y}}\|,\end{aligned}\tag{4.1}$$

where the correction spaces for the primal and dual problems are

$$\mathcal{C}_n^{\text{pr}} \equiv \mathbf{y}_0 + \mathcal{K}_n(\mathbf{r}_0^{\text{pr}}, \mathbf{A}), \quad \mathcal{C}_n^{\text{du}} \equiv \mathbf{x}_0 + \mathcal{K}_n(\mathbf{r}_0^{\text{du}}, \mathbf{A}^T).\tag{4.2}$$

The proposed approach is a constrained minimization statement. That is, it locates minimizers,

$$\begin{aligned}\mathbf{x}_n^{\text{CSGMRES}} &= \arg \min_{\bar{\mathbf{x}} \in \mathcal{C}_n^{\text{pr}}} \|\mathbf{b} - \mathbf{A}\bar{\mathbf{x}}\|, \text{ s.t. } (\mathbf{b} - \mathbf{A}\bar{\mathbf{x}})^T \mathbf{y}_{n-1}^{\text{CSGMRES}} = 0, \\ \mathbf{y}_n^{\text{CSGMRES}} &= \arg \min_{\bar{\mathbf{y}} \in \mathcal{C}_n^{\text{du}}} \|\mathbf{g} - \mathbf{A}^T \bar{\mathbf{y}}\|, \text{ s.t. } (\mathbf{g} - \mathbf{A}^T \bar{\mathbf{y}})^T \mathbf{x}_{n-1}^{\text{CSGMRES}} = 0.\end{aligned}\tag{4.3}$$

4.2 Arnoldi Process and Equality Constraint

Arnoldi process applied to each of the primal and dual problem produces orthonormal basis vectors of the respective Krylov subspaces, $\mathbf{V}_n \equiv [\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_n]$ and $\mathbf{W}_n \equiv [\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_n]$ satisfying the relations

$$\begin{aligned}
\mathbf{A}\mathbf{V}_n &= \mathbf{V}_{n+1}\bar{\mathbf{H}}_n^{\text{pr}}, \\
\mathbf{A}^T\mathbf{W}_n &= \mathbf{W}_{n+1}\bar{\mathbf{H}}_n^{\text{du}},
\end{aligned} \tag{4.4}$$

where $\bar{\mathbf{H}}_n^{\text{pr}}, \bar{\mathbf{H}}_n^{\text{du}}$ are the $(n+1) \times n$ upper Hessenberg matrices associated with the primal and dual Arnoldi process respectively. Since the n th primal and dual iterate are in the respective subspaces, they may be written in the form

$$\begin{aligned}
\mathbf{x}_n &= \mathbf{x}_0 + \mathbf{V}_n\mathbf{k}_n, \\
\mathbf{y}_n &= \mathbf{y}_0 + \mathbf{W}_n\mathbf{l}_n.
\end{aligned} \tag{4.5}$$

Using (4.4) it is evident that with the iterates denoted as above, the residuals are

$$\begin{aligned}
\mathbf{r}_n^{\text{pr}} &= \mathbf{V}_{n+1}(\beta^{\text{pr}}\mathbf{e}_1^{n+1} - \bar{\mathbf{H}}_n^{\text{pr}}\mathbf{k}_n), \\
\mathbf{r}_n^{\text{du}} &= \mathbf{W}_{n+1}(\beta^{\text{du}}\mathbf{e}_1^{n+1} - \bar{\mathbf{H}}_n^{\text{du}}\mathbf{l}_n),
\end{aligned} \tag{4.6}$$

where $\beta^{\text{pr}} \equiv \|\mathbf{r}_0^{\text{pr}}\|$, $\beta^{\text{du}} \equiv \|\mathbf{r}_0^{\text{du}}\|$. Since the columns of \mathbf{V}_{n+1} and \mathbf{W}_{n+1} are orthonormal, the GMRES iterates may be chosen with

$$\begin{aligned}
\mathbf{k}_n^{\text{GMRES}} &= \arg \min_{\mathbf{k}} \|\beta^{\text{pr}}\mathbf{e}_1^{n+1} - \bar{\mathbf{H}}_n^{\text{pr}}\mathbf{k}\|, \\
\mathbf{l}_n^{\text{GMRES}} &= \arg \min_{\mathbf{l}} \|\beta^{\text{du}}\mathbf{e}_1^{n+1} - \bar{\mathbf{H}}_n^{\text{du}}\mathbf{l}\|.
\end{aligned} \tag{4.7}$$

Let us now look at the problem of determining $\mathbf{k}_n, \mathbf{l}_n$ for CSGMRES strategy, (4.3). Define the $n \times 1$ vectors \mathbf{p}_n and \mathbf{q}_n by

$$\begin{aligned}
\mathbf{p}_n &\equiv (\bar{\mathbf{H}}_n^{\text{pr}})^T \mathbf{V}_{n+1}^T \mathbf{y}_{n-1}^{\text{CSGMRES}}, \\
\mathbf{q}_n &\equiv (\bar{\mathbf{H}}_n^{\text{du}})^T \mathbf{W}_{n+1}^T \mathbf{x}_{n-1}^{\text{CSGMRES}}.
\end{aligned} \tag{4.8}$$

and define the scalars c_n, d_n as

$$c_n \equiv (\mathbf{y}_{n-1}^{\text{CSGMRES}})^T \mathbf{r}_0^{\text{pr}}, d_n \equiv (\mathbf{x}_{n-1}^{\text{CSGMRES}})^T \mathbf{r}_0^{\text{du}}. \tag{4.9}$$

Then, the CSGMRES strategy is equivalent to the respective n -dimensional least squares problem with a one-dimensional constraint,

$$\begin{aligned}
\mathbf{k}_n^{\text{CSGMRES}} &= \arg \min_{\mathbf{k}} \|\beta^{\text{pr}} \mathbf{e}_1^{n+1} - \bar{\mathbf{H}}_n^{\text{pr}} \mathbf{k}\|, \quad \text{s.t. } \mathbf{p}_n^T \mathbf{k} = c_n, \\
\mathbf{l}_n^{\text{CSGMRES}} &= \arg \min_{\mathbf{l}} \|\beta^{\text{du}} \mathbf{e}_1^{n+1} - \bar{\mathbf{H}}_n^{\text{du}} \mathbf{l}\|, \quad \text{s.t. } \mathbf{q}_n^T \mathbf{l} = d_n.
\end{aligned} \tag{4.10}$$

4.2.1 Algorithm Implementation

A number of different methods exist to solve the equality constrained least squares (LSE) problems (4.10) [27, 22, 23, 28]. However, we note that our LSE problems have certain features. Firstly, the matrices $\bar{\mathbf{H}}_n^{\text{pr}}$ and $\bar{\mathbf{H}}_n^{\text{du}}$ are upper Hessenberg matrices, therefore it is simple to perform QR factorization using successive Givens rotations, as is done in Saad's original implementation of GMRES [38]. Moreover, we note that while the vectors $\mathbf{p}_n, \mathbf{q}_n$ and scalars c_n, d_n may be quite different at each iteration, the matrices $\bar{\mathbf{H}}_n^{\text{pr}}$ and $\bar{\mathbf{H}}_n^{\text{du}}$ are merely updated by appending columns. Therefore, it is desirable to implement the successive LSE problems (4.10) based on successive Givens rotations of matrices $\bar{\mathbf{H}}_n^{\text{pr}}$ and $\bar{\mathbf{H}}_n^{\text{du}}$ and then make the necessary adjustments to enforce the equality constraints.

In [24], a method of solving LSE was derived using the method of Lagrange multipliers. Specifically, the primal LSE problem is equivalent to solving the extended system

$$\begin{bmatrix} (\bar{\mathbf{H}}_n^{\text{pr}})^T \bar{\mathbf{H}}_n^{\text{pr}} & \mathbf{p}_n \\ \mathbf{p}_n^T & 0 \end{bmatrix} \begin{bmatrix} \mathbf{k}_n \\ \lambda \end{bmatrix} = \begin{bmatrix} \beta^{\text{pr}} (\bar{\mathbf{H}}_n^{\text{pr}})^T \mathbf{e}_1^{n+1} \\ c_n \end{bmatrix}. \quad (4.11)$$

The first step is to solve the unconstrained least-squares problem for \mathbf{k}_n^{ls} ,

$$\left[(\bar{\mathbf{H}}_n^{\text{pr}})^T \bar{\mathbf{H}}_n^{\text{pr}} \right] \mathbf{k}_n^{\text{ls}} = \beta^{\text{pr}} (\bar{\mathbf{H}}_n^{\text{pr}})^T \mathbf{e}_1^{n+1}, \quad (4.12)$$

which may be done in the usual manner of QR decomposition by successive Givens rotations on $\bar{\mathbf{H}}_n^{\text{pr}}$,

$$\mathbf{Q}_n^{\text{pr}} \bar{\mathbf{H}}_n^{\text{pr}} = \begin{bmatrix} \mathbf{R}_n^{\text{pr}} \\ 0 \end{bmatrix}, \quad (4.13)$$

where \mathbf{R}_n^{pr} is $n \times n$ and upper triangular. Also, the right hand side of (4.12) transforms to

$$\mathbf{Q}_n^{\text{pr}} \beta^{\text{pr}} \mathbf{e}_1 = \begin{bmatrix} \mathbf{s}_n^{\text{pr}} \\ \cdot \end{bmatrix}, \quad (4.14)$$

where \mathbf{s}_n^{pr} is an $n \times 1$ vector and the last entry of the above vector is ignored, hence not shown. \mathbf{k}_n^{ls} is then obtained by,

$$\mathbf{k}_n^{\text{ls}} = \left(\mathbf{R}_n^{\text{pr}} \right)^{-1} \mathbf{s}_n^{\text{pr}}. \quad (4.15)$$

Then, writing

$$\mathbf{k}_n = \mathbf{k}_n^{\text{ls}} + \mathbf{z}_n^{\text{pr}}, \quad (4.16)$$

the problem (4.11) is equivalent to determining \mathbf{z}_n^{pr} . It may be seen that the equation of the first block row of (4.11) is automatically satisfied if \mathbf{z}_n^{pr} is of the form $\mathbf{z}_n^{\text{pr}} = \mathbf{j}_n^{\text{pr}} \lambda$, where \mathbf{j}_n^{pr} satisfies the equation

$$\left[(\bar{\mathbf{H}}_n^{\text{pr}})^T \bar{\mathbf{H}}_n^{\text{pr}} \right] \mathbf{j}_n^{\text{pr}} = \mathbf{p}_n. \quad (4.17)$$

The value for λ is determined by the equation

$$\mathbf{p}_n^T \mathbf{j}_n^{\text{pr}} \lambda = c_n - \mathbf{p}_n^T \mathbf{k}_n^{\text{ls}}. \quad (4.18)$$

After some algebraic manipulations,

$$\mathbf{z}_n^{\text{pr}} = \left(\frac{c_n - \mathbf{p}_n^T \mathbf{k}_n^{\text{ls}}}{\|\mathbf{t}_n^{\text{pr}}\|^2} \right) \left(\mathbf{R}_n^{\text{pr}} \right)^{-1} \mathbf{t}_n^{\text{pr}}, \quad (4.19)$$

where \mathbf{t}_n^{pr} is defined as

$$\mathbf{t}_n^{\text{pr}} = \left(\mathbf{R}_n^{\text{pr}} \right)^{-T} \mathbf{p}_n. \quad (4.20)$$

4.2.2 CSGMRES Algorithm

- Initialization

– Obtain initial guesses \mathbf{x}_0 and \mathbf{y}_0 . Compute

$$\mathbf{x}'_0 = \mathbf{M}_2 \mathbf{x}_0, \quad \mathbf{y}'_0 = \mathbf{M}_1^T \mathbf{y}_0. \quad (4.21)$$

– Compute initial preconditioned residuals

$$\mathbf{r}'_0{}^{\text{pr}} = \mathbf{M}_1^{-1}(\mathbf{b} - \mathbf{A}\mathbf{x}_0), \quad \mathbf{r}'_0{}^{\text{du}} = \mathbf{M}_2^{-T}(\mathbf{g} - \mathbf{A}^T \mathbf{y}_0). \quad (4.22)$$

– Let $\beta^{\text{pr}} = \|\mathbf{r}'_0{}^{\text{pr}}\|$, $\beta^{\text{du}} = \|\mathbf{r}'_0{}^{\text{du}}\|$. Normalize : $\mathbf{v}_1 = \mathbf{r}'_0{}^{\text{pr}}/\beta^{\text{pr}}$, $\mathbf{w}_1 = \mathbf{r}'_0{}^{\text{du}}/\beta^{\text{du}}$.

• FOR $n = 1, 2, 3, \dots$, DO

– Primal Arnoldi

$$\begin{aligned} \tilde{\mathbf{v}}_n &= \mathbf{M}_1^{-1} \mathbf{A} \mathbf{M}_2^{-1} \mathbf{v}_n. \\ \bar{\mathbf{H}}_n^{\text{pr}}(i, n) &= (\tilde{\mathbf{v}}_n, \mathbf{v}_i), i = 1, 2, \dots, n. \\ \hat{\mathbf{v}}_{n+1} &= \tilde{\mathbf{v}}_n - \sum_{i=1}^n \bar{\mathbf{H}}_n^{\text{pr}}(i, n) \mathbf{v}_i. \\ \bar{\mathbf{H}}_n^{\text{pr}}(n+1, n) &= \|\hat{\mathbf{v}}_{n+1}\|. \\ \mathbf{v}_{n+1} &= \hat{\mathbf{v}}_{n+1} / \bar{\mathbf{H}}_n^{\text{pr}}(n+1, n). \end{aligned}$$

– Dual Arnoldi

$$\begin{aligned} \tilde{\mathbf{w}}_n &= \mathbf{M}_2^{-T} \mathbf{A}^T \mathbf{M}_1^{-T} \mathbf{w}_n. \\ \bar{\mathbf{H}}_n^{\text{du}}(i, n) &= (\tilde{\mathbf{w}}_n, \mathbf{w}_i), i = 1, 2, \dots, n. \\ \hat{\mathbf{w}}_{n+1} &= \tilde{\mathbf{w}}_n - \sum_{i=1}^n \bar{\mathbf{H}}_n^{\text{du}}(i, n) \mathbf{w}_i. \\ \bar{\mathbf{H}}_n^{\text{du}}(n+1, n) &= \|\hat{\mathbf{w}}_{n+1}\|. \\ \mathbf{w}_{n+1} &= \hat{\mathbf{w}}_{n+1} / \bar{\mathbf{H}}_n^{\text{du}}(n+1, n). \end{aligned}$$

– Primal LSE

$$\mathbf{p}_n = (\bar{\mathbf{H}}_n^{\text{pr}})^T (\mathbf{V}_{n+1}^T \mathbf{y}'_{n-1}), \quad c_n = \beta^{\text{pr}} \mathbf{v}_1^T \mathbf{y}'_{n-1}.$$

Obtain \mathbf{Q}_n^{pr} , \mathbf{R}_n^{pr} with a Givens rotation on $\bar{\mathbf{H}}_n^{\text{pr}}$.

Obtain \mathbf{s}_n^{pr} from $(\mathbf{Q}_n^{\text{pr}})^T \beta^{\text{pr}} \mathbf{e}_1$.

$$\mathbf{k}_n^{\text{ls}} = \left(\mathbf{R}_n^{\text{pr}}\right)^{-1} \mathbf{s}_n^{\text{pr}}, \quad \mathbf{t}_n^{\text{pr}} = \left(\mathbf{R}_n^{\text{pr}}\right)^{-T} \mathbf{p}_n.$$

$$\mathbf{k}_n = \mathbf{k}_n^{\text{ls}} + \left(\frac{c_n - \mathbf{p}_n^T \mathbf{k}_n^{\text{ls}}}{\|\mathbf{t}_n^{\text{pr}}\|^2} \right) \left(\mathbf{R}_n^{\text{pr}} \right)^{-1} \mathbf{t}_n^{\text{pr}}.$$

– Dual LSE

$$\mathbf{q}_n = (\bar{\mathbf{H}}_n^{\text{du}})^T (\mathbf{W}_{n+1}^T \mathbf{x}'_{n-1}), \quad d_n = \beta^{\text{du}} \mathbf{w}_1^T \mathbf{x}'_{n-1}.$$

Obtain $\mathbf{Q}_n^{\text{du}}, \mathbf{R}_n^{\text{du}}$ with a Givens rotation on $\bar{\mathbf{H}}_n^{\text{du}}$.

Obtain \mathbf{s}_n^{du} from $(\mathbf{Q}_n^{\text{du}})^T \beta^{\text{du}} \mathbf{e}_1$.

$$\mathbf{l}_n^{\text{ls}} = \left(\mathbf{R}_n^{\text{du}} \right)^{-1} \mathbf{s}_n^{\text{du}}, \quad \mathbf{t}_n^{\text{du}} = \left(\mathbf{R}_n^{\text{du}} \right)^{-T} \mathbf{q}_n.$$

$$\mathbf{l}_n = \mathbf{l}_n^{\text{ls}} + \left(\frac{d_n - \mathbf{q}_n^T \mathbf{l}_n^{\text{ls}}}{\|\mathbf{t}_n^{\text{du}}\|^2} \right) \left(\mathbf{R}_n^{\text{du}} \right)^{-1} \mathbf{t}_n^{\text{du}}.$$

– Form iterates

$$\mathbf{x}'_n = \mathbf{x}'_0 + \mathbf{V}_n \mathbf{k}_n,$$

$$\mathbf{y}'_n = \mathbf{y}'_0 + \mathbf{W}_n \mathbf{l}_n.$$

- Obtain solutions to original systems

$$\mathbf{x}_n = \mathbf{M}_2^{-1} \mathbf{x}'_n,$$

$$\mathbf{y}_n = \mathbf{M}_1^{-T} \mathbf{y}'_n.$$

4.3 Numerical Experiments

CSGMRES is applied to the same preconditioned primal and dual problems as those given in Sect. 3.6.1 and Sect. 3.6.2.

4.3.1 Example 1

From Figure 4-1, it is seen that both the primal and dual iterates provide functional estimates that initially appear to converge faster than the respective residual norms, indicating that the test spaces provide good approximations to the corresponding dual problem. This may be seen as a consequence of the problem matrix being symmetric. However, GMRES is clearly not superconvergent and CSGMRES iterates provide significantly better functional output estimates. Also, note that after the few initial iterations, the residual norms of GMRES and CSGMRES exhibit very similar residual convergence behavior.

4.3.2 Example 2

The results of this example are shown in Figure 4-2. It can be seen that the convergence of functional estimates obtained from conventional GMRES iterates is linear in the respective residuals. Again, the improvement in functional estimates provided by CSGMRES iterates is apparent, while the smooth residual norm convergence behavior is maintained.

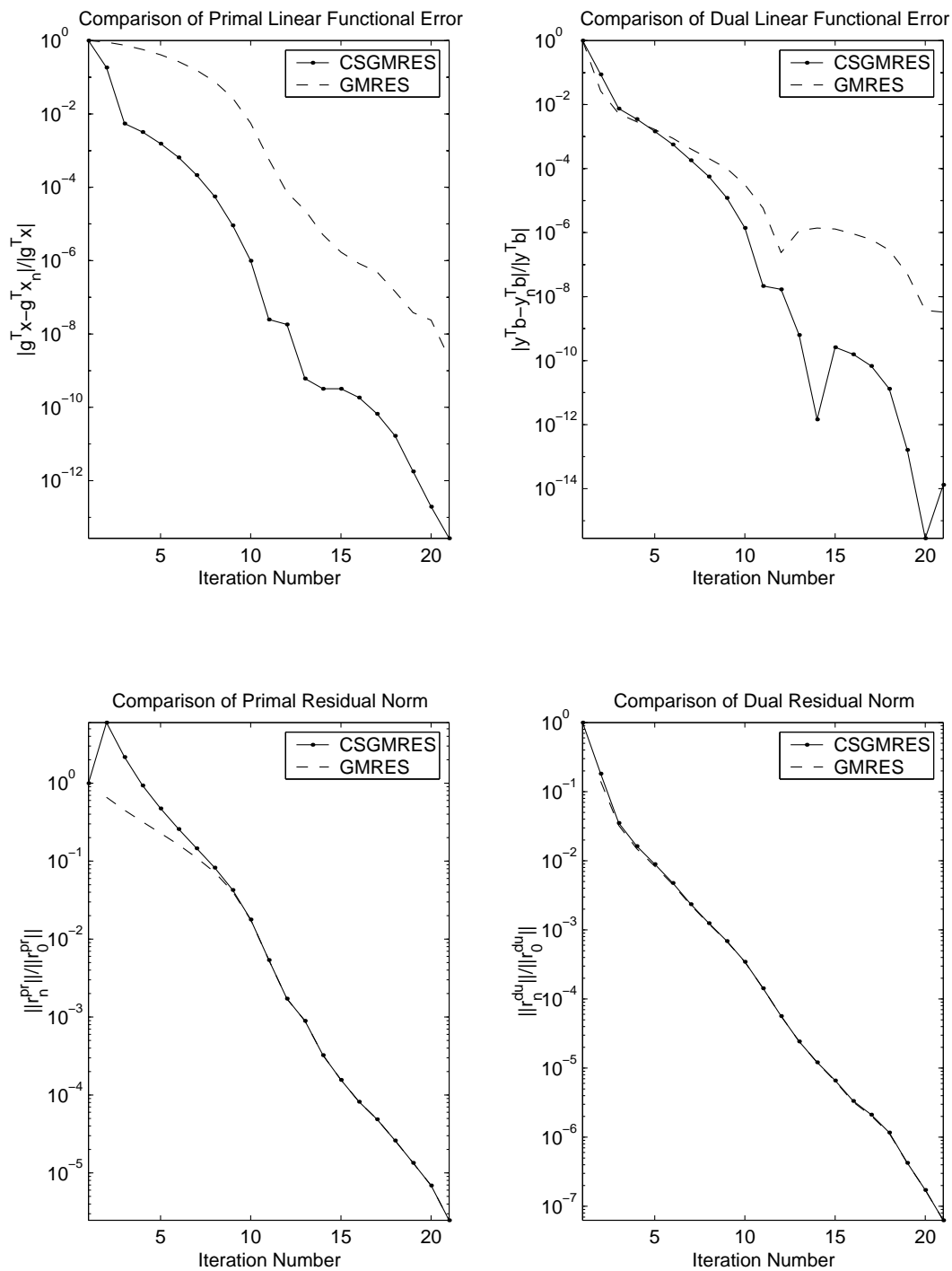


Figure 4-1: Plots from CSGMRES Example 1 : Poisson problem

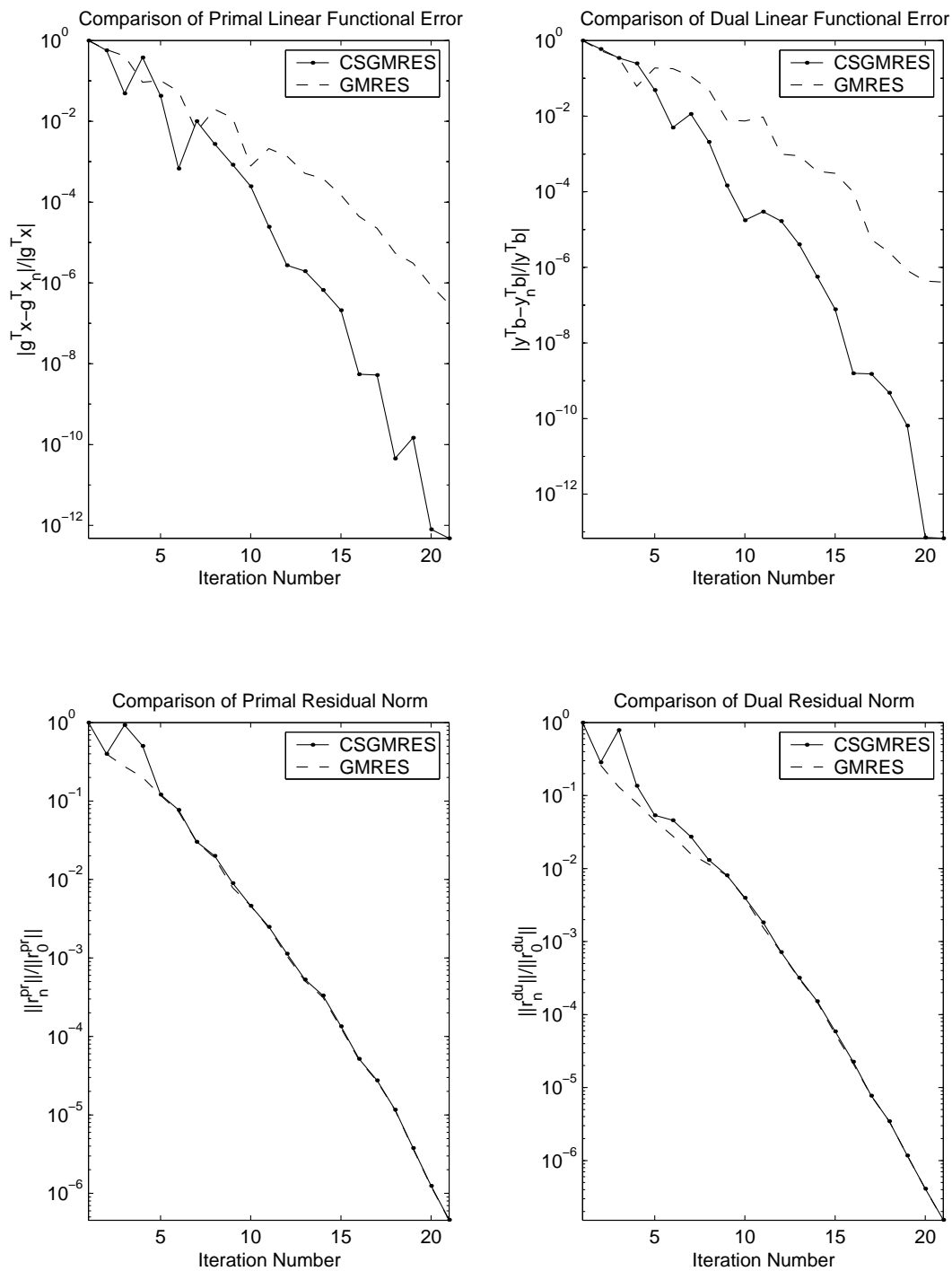


Figure 4-2: Plots from CSGMRES Example 2 : Euler flow

Chapter 5

Conclusions and Future Work

Using adjoint analysis, the convergence of functional estimates using iterates obtained from several Krylov subspace methods is studied. Analysis shows that for methods based on the non-symmetric Lanczos process, an appropriate choice of the initial shadow vector allows the attainment of superconvergence for a chosen functional output. Moreover, this choice allows for the simultaneous solution of the dual problem associated with the output. A superconvergent QMR for primal-dual solutions is constructed and demonstrated to be a viable approach. Krylov methods based on the Arnoldi process are shown to lack superconvergence. However, with appropriate coupling of primal and dual problems, superconvergent variants may be constructed. The viability of such an approach is demonstrated by the construction of a superconvergent variant of GMRES.

The potential benefits of the proposed simultaneous primal-dual strategy to the general nonlinear setting remains a topic for further research.

Appendix A

SSQMR MATLAB code

function

```
[flag,pr_soln,du_soln,pr_res_norm,du_res_norm,pr_func_est,du_func_est]  
= SSQMR(A,b,g,iter,i,M1,M2)
```

```
b_norm = norm(b);
```

```
g_norm = norm(g);
```

```
x_pr = zeros(length(b),1);
```

```
x_du = zeros(length(b),1);
```

```
v = M1\b;
```

```
rho = norm(v);
```

10

```
v = v/rho;
```

```
w = M2'\g;
```

```
zeta = norm(w);
```

```
w = w/zeta;
```

```
p = zeros(length(b),1);
```

```
q = zeros(length(b),1);
```

```
d_pr = zeros(length(b),1);
```

```
d_du = zeros(length(b),1);
```

```
c_pr = 1;
```

20

```

c_du = 1;
epsilon = 1;
theta_pr = 0;
theta_du = 0;
eta_pr = -1;
eta_du = -1;
w_c_pr = 1;
w_c_du = 1;
w_theta_pr = 0;
w_theta_du = 0;
w_eta_pr = -1;
w_eta_du = -1;

```

```

k = 0;
for h = 1 : i
    m(h) = 2 - h;
end

```

```

for n = 1 : iter
    delta = w'*v;
    if abs(delta) < 1e-14 | abs(epsilon) < 1e-14
        flag = 1
        break
    end

```

```

k = mod(k,i) + 1;

```

```

p = v - p*(zeta*delta/epsilon);
q = w - q*(rho*delta/epsilon);
P(:,k) = p;

```

```

Q(:,k) = q;
p_tilde = A*(M2\p);
q_tilde = M1'\q;
epsilon1 = q_tilde'*p_tilde;
beta = epsilon1/delta;
v_tilde = M1\p_tilde - beta*v;
w_tilde = M2'\(A'*q_tilde) - beta*w;
BETA(k) = beta;
RHO(k) = rho;
ZETA(k) = zeta;
rho1 = norm(v_tilde);
zeta1 = norm(w_tilde);
v = v_tilde/rho1;
w = w_tilde/zeta1;
w_theta_pr1 = rho/(w_c_pr*beta);
w_c_pr1 = 1/sqrt(1+(w_theta_pr1)^2);
w_eta_pr = - w_eta_pr*(rho*(w_c_pr1)^2)/(beta*(w_c_pr)^2);
w_theta_du1 = zeta/(w_c_du*beta);
w_c_du1 = 1/sqrt(1+(w_theta_du1)^2);
w_eta_du = - w_eta_du*(zeta*(w_c_du1)^2)/(beta*(w_c_du)^2);

for h = 1 : i
if m(h) == 1
w_q_pr(h) = delta; w_d_pr(h) = 0; w_x_pr(h) = 0;
w_q_du(h) = delta; w_d_du(h) = 0; w_x_du(h) = 0;
end

if (1 <=m(h)) & (m(h) < i )
if n ~ = 1
w_q_pr(h) = -(zeta*delta/epsilon)*w_q_pr(h);

```

```

end
w_d_pr(h) = w_eta_pr*w_q_pr(h) + w_d_pr(h)*(w_theta_pr*w_c_pr1)^2;
w_x_pr(h) = w_x_pr(h) + w_d_pr(h);
if n ~ = 1
    w_q_du(h) = -(rho*delta/epsilon)*w_q_du(h);
end
w_d_du(h) = w_eta_du*w_q_du(h) + w_d_du(h)*(w_theta_du*w_c_du1)^2;
w_x_du(h) = w_x_du(h) + w_d_du(h);
end

```

90

```

if m(h) == i-1
    m(h) = 2-i;
omega_pr1 = w_x_du(h);
omega_du1 = w_x_pr(h);
end
end

```

```

rho = rho1;
zeta = zeta1;
epsilon = epsilon1;
w_c_pr = w_c_pr1;
w_c_du = w_c_du1;
w_theta_pr = w_theta_pr1;
w_theta_du = w_theta_du1;

```

100

```

if n == i-1
omega_pr = omega_pr1;
omega_du = omega_du1;
end

```

110

```

if n >= i
theta_pr1 = omega_pr1*RHO(mod(k,i)+1)/(omega_pr1*c_pr*BETA(mod(k,i)+1));
c_pr1 = 1/sqrt(1+(theta_pr1)^2);
eta_pr = - eta_pr*RHO(mod(k,i)+1)*(c_pr1)^2/(BETA(mod(k,i)+1)*(c_pr)^2);
d_pr = eta_pr*P(:,mod(k,i)+1) + (theta_pr*c_pr1)^2*d_pr;
x_pr = x_pr + d_pr;
omega_pr = omega_pr1;
theta_pr = theta_pr1;
c_pr = c_pr1;

```

120

```

theta_du1 = omega_du1*ZETA(mod(k,i)+1)/(omega_du1*c_du*BETA(mod(k,i)+1));
c_du1 = 1/sqrt(1+(theta_du1)^2);
eta_du = - eta_du*ZETA(mod(k,i)+1)*(c_du1)^2/(BETA(mod(k,i)+1)*(c_du)^2);
d_du = eta_du*Q(:,mod(k,i)+1) + (theta_du*c_du1)^2*d_du;
x_du = x_du + d_du;
omega_du = omega_du1;
theta_du = theta_du1;
c_du = c_du1;

```

```

pr_soln = M2\x_pr;
du_soln = M1'\x_du;
pr_res_norm(n-i+1) = norm(b-A*pr_soln)/b_norm;
du_res_norm(n-i+1) = norm(g-A'*du_soln)/g_norm;
pr_func_est(n-i+1) = g'*pr_soln;
du_func_est(n-i+1) = b'*du_soln;
end

```

```

for h = 1 : i
m(h) = m(h)+1;
end

```

140

end

flag = 0;

150

Appendix B

CSGMRES MATLAB code

function

```
[pr_soln,du_soln,pr_res_norm,du_res_norm,pr_func_est,du_func_est]  
= CSGMRES(A,b,g,iter,M1,M2)
```

```
b_norm = norm(b);
```

```
g_norm = norm(g);
```

```
precond_b = M1\b;
```

```
precond_g = M2'\g;
```

```
beta_pr = norm(precond_b);
```

```
beta_du = norm(precond_g);
```

10

```
V(:,1) = precond_b/beta_pr;
```

```
W(:,1) = precond_g/beta_du;
```

```
for n = 1 : iter
```

```
u = M1\((A*(M2\V(:,n))));
```

```
v = M2'\(A'*(M1'\W(:,n)));
```

```
for i = 1 : n
```

```
    H_pr(i,n) = V(:,i)'\*u;
```

```
    u = u - H_pr(i,n)*V(:,i);
```

20

```

    H_du(i,n) = W(:,i)'*v;
v = v - H_du(i,n)*W(:,i);
end

```

```

H_pr(n+1,n) = norm(u);
V(:,n+1) = u/H_pr(n+1,n);
H_du(n+1,n) = norm(v);
W(:,n+1) = v/H_du(n+1,n);

```

```

e1 = zeros(n+1,1);
e1(1) = 1;

```

30

```

if n == 1
    p = H_pr'*V'*g;
    k = p'\g'*precond_b;
    q = H_du'*W'*b;
    l = q'\b'*precond_g;
    [Q_pr,R_pr] = qr(H_pr);
    [Q_du,R_du] = qr(H_du);

```

else

40

```

    p = H_pr'*(V'*y);
    [Q_pr,R_pr,k] = lse(Q_pr,R_pr,H_pr,beta_pr*e1,p,y'*precond_b);
    q = H_du'*W'*x;
    [Q_du,R_du,l] = lse(Q_du,R_du,H_du,beta_du*e1,q,x'*precond_g);

```

end

```

x = V(:,1:n)*k;
y = W(:,1:n)*l;
pr_soln = M2\x;
du_soln = M1'y;

```

50

```

pr_res_norm(n) = norm(b-A*pr_soln)/b_norm;
du_res_norm(n) = norm(g-A'*du_soln)/g_norm;
pr_func_est(n) = g'*pr_soln;
du_func_est(n) = b'*du_soln;

```

```

end

```

60

```

function [Q,R,k] = lse(Q,R,H,beta_e1,p,c)

```

```

[m,n] = size(H);
R(m,n-1)=0;
Q(m,m)=1;
[Q,R] = qrinsert(Q,R,n,H(:,n));

```

```

s = Q'*beta_e1;
k_ls = R\s;
t = R'\p;
k = k_ls + ((c-p'*k_ls)/norm(t)^2)*(R\t);

```

70

Bibliography

- [1] M. Ainsworth and J. T. Oden. A posteriori error estimation in finite element analysis. *Computer Methods in Applied Mechanics and Engineering*, 142:1–88, 1997.
- [2] W. K. Anderson and D. L. Bonhaus. An implicit upwind algorithm for computing turbulent flows on unstructured grids. *Computer and Fluids*, 23(7):1–21, 1994.
- [3] W. K. Anderson and V. Venkatakrishnan. Aerodynamics design optimization on unstructured grids with a continuous adjoint formulation. *AIAA*, 0643, 1997.
- [4] T. Barth and T. A. Manteuffel. Estimating the spectrum of a matrix using the roots of the polynomials associated with the QMR iteration. Report, Center for Computational Mathematics, University of Colorado at Denver, 1993.
- [5] T. Barth and T. A. Manteuffel. Variable metric conjugate gradient methods. In *Proceedings of the 10th International Symposium on Matrix Analysis and Parallel Computing*, Keio University, Yokohama, Japan, March 1994.
- [6] R. Becker and R. Rannacher. Weighted a posteriori error control in finite element methods. In *Proc. ENUMATH-97*, Singapore, 1998. World Scientific Pub.
- [7] B. Beckermann and A. B. J. Kuijlaars. Superlinear convergence of conjugate gradients. *SIAM Journal on Numerical Analysis*, 39:300–329, 2001.
- [8] T. F. Chan and W. L. Wan. Analysis of projection methods for solving linear systems with multiple right-hand sides. *SIAM Journal on Scientific Computing*, 18:1698–1721, 1997.

- [9] J. Cullum and A. Greenbaum. Relations between Galerkin and norm-minimizing iterative methods for solving linear systems. *SIAM Journal on Matrix Analysis and Applications*, 17:223–247, 1996.
- [10] R. Ehrig and P. Deuffhard. GMERR - an error minimizing variant of GMRES. Report SC 97-63, Konrad-zure-Zentrum für Informationstechnik, Berlin, 1997.
- [11] M. Eiermann and O. G. Ernst. Geometric aspects in the theory of Krylov subspace methods. *Acta Numerica*, 10:251–312, 2001.
- [12] J. Elliot and J. Peraire. Aerodynamic optimization on unstructured meshes with viscous effects. AIAA 97-1849, 1997.
- [13] R. Fletcher. Conjugate gradient methods for indefinite systems. In *Lecture Notes in Mathematics 506*, pages 73–89. Springer, Berlin, 1976.
- [14] R. W. Freund. A transpose-free quasi-minimal residual algorithm for non-Hermitian linear systems. *SIAM Journal on Scientific Computing*, 14:470–482, 1993.
- [15] R. W. Freund, G. H. Golub, and N. M. Nachtigal. Iterative solution of linear systems. *Acta Numerica*, 1:57–100, 1992.
- [16] R. W. Freund, G. H. Golub, and N. M. Nachtigal. Recent advances in Lanczos-based iterative methods for nonsymmetric linear systems. In *Algorithmic Trends in Computational Fluid Dynamics*, pages 137–162. Springer-Verlag, 1993.
- [17] R. W. Freund, M. H. Gutknecht, and N. M. Nachtigal. An implementation of the look-ahead Lanczos algorithm for non-Hermitian matrices. *SIAM Journal on Scientific Computing*, 14:137–158, 1993.
- [18] R. W. Freund and N. M. Nachtigal. QMR: a quasi-minimal residual method for non-Hermitian linear systems. *Numerische Mathematik*, 60:315–339, 1991.

- [19] R. W. Freund and N. M. Nachtigal. An implementation of the QMR method based on two-term recurrences. *SIAM Journal on Scientific Computing*, 15:313–337, 1994.
- [20] R. W. Freund and T. Szeto. A transpose-free quasi-minimal residual squared algorithm for non-Hermitian linear systems. In R. Vichnevetsky, D. Knight, and G. Richter, editors, *Advances in Computer Methods for Partial Differential Equations – VII*, pages 258–264. IMACS, 1992.
- [21] M. B. Giles and N. A. Pierce. An introduction to the adjoint approach to design. *Flow, Turbulence and Combustion*, 65:393–415, 2000.
- [22] G. H. Golub and C. F. van Loan. *Matrix Computations*. John Hopkins University Press, Baltimore, MD, USA, third edition, 1996.
- [23] M. Gulliksson and P-A Wedin. Modifying the QR-decomposition to constrained and weighted linear least squares. *SIAM Journal on Matrix Analysis and Applications*, 13:1298–1313, 1992.
- [24] M. T. Heath. Some extensions of al algorithm for sparse linear least squares problems. *SIAM Journal on Scientific Computing*, 3:223–237, 1982.
- [25] A. Jameson. Aerodynamic design via control theory. *Journal of Scientific Computing*, 3:233–260, 1988.
- [26] M. Kilmer, E. Miller, and C. Rappaport. QMR-based projection techniques for the solution of non-Hermitian systems with multiple right-hand sides. *SIAM Journal on Scientific Computing*, 23:761–780, 2001.
- [27] C. L. Lawson and R. J. Hanson. *Solving least squares problems*. Prentice-Hall, Englewoods Cliffs, NJ, 1974.
- [28] C. Van Loan. On the method of weighting for equality-constrained least-squares problems. *SIAM Journal on Numerical Analysis*, 22:851–864, 1985.

- [29] J. Lu and D. L. Darmofal. Functional-output convergence results for Krylov subspace methods and a superconvergent variant of GMRES. *Numerische Mathematik*, in preparation.
- [30] L. Machiels. Output bounds for iterative solutions of linear partial differential equations. Technical Report UCRL-JC-137864, Lawrence Livermore National Laboratory, March 2000.
- [31] A. T. Patera and E. M. Rønquist. A general output bound result : application to discretization and iteration error estimation and control. *Mathematical Models and Methods in Applied Science*, 11:685–712, 2001.
- [32] J. Peraire and A. T. Patera. Asymptotic a posteriori finite element bounds for the outputs of noncoercive problems : the Helmholtz and Burgers equations. *Computer Methods in Applied Mechanics and Engineering*, 171:77–86, 1999.
- [33] N. A. Pierce and M. B. Giles. Adjoint recovery of superconvergent functionals from PDE approximations. *SIAM Review*, 42:247–264, 2000.
- [34] J. J. Reuther, A. Jameson, and J. J. Alonso. Constrained multipoint aerodynamics shape optimization using an adjoint formulation and parallel computers, parts 1 and 2. *J. Aircraft*, 36(1):51–60,61–74, 1999.
- [35] M. Rozložník and R. Weiss. On the stable implementation of the generalized minimal error method. *Journal of Computational and Applied Mathematics*, 98:49–62, 1998.
- [36] Y. Saad. Krylov subspace methods for solving large unsymmetric linear systems. *Mathematics of Computation*, 37:105–126, 1981.
- [37] Y. Saad. *Iterative methods for sparse linear systems*. PWS Publishing, Boston, Massachusetts, 1996.
- [38] Y. Saad and M. H. Schultz. GMRES : A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM Journal on Scientific and Statistical Computing*, 7:856–869, 1986.

- [39] P. Sonneveld. A fast Lanczos-type solver for nonsymmetric linear systems. *SIAM Journal on Scientific Computing*, 10:36–52, 1989.
- [40] D. B. Szyld and J. A. Vogel. FQMR : a flexible quasi-minimal residual method with inexact preconditioning. *SIAM Journal on Scientific Computing*, 23:363–380, 2001.
- [41] C. H. Tong. A family of quasi-residual methods for nonsymmetric linear systems. *SIAM Journal on Scientific Computing*, 15:89–105, 1994.
- [42] H. A. van der Vorst. Bi-CGSTAB : A fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems. *SIAM Journal on Scientific and Statistical Computing*, 13:631–644, 1992.
- [43] H. A. van der Vorst. The superlinear convergence behaviour of GMRES. *Journal of Computational and Applied Mathematics*, 48:631–644, 1993.
- [44] H. A. van der Vorst. An overview of approaches for the stable computation of hybrid BiCG methods. *Applied Numerical Mathematics*, 19:235–254, 1995.
- [45] D. A. Venditti. *Grid Adaptation for functional outputs of compressible flow simulations*. PhD dissertation, Massachusetts Institute of Technology, Department of Aeronautics and Astronautics, 2002.
- [46] D. A. Venditti and D. L. Darmofal. Adjoint error estimation and grid adaptation for functional outputs : application to quasi-one-dimensional flow. *Journal of Computational Physics*, 164:204–227, 2000.
- [47] D. A. Venditti and D. L. Darmofal. Grid adaptation for functional outputs : application to two-dimensional inviscid flows. *Journal of Computational Physics*, 176:40–69, 2002.
- [48] L. B. Wahlbin. *Superconvergence in Galerkin Finite Element Methods*. Springer Verlag, Berlin/Heidelberg, 1995.

- [49] R. Weiss. Error-minimizing Krylov subspace methods. *SIAM Journal on Scientific Computing*, 16:511–527, 1994.