DELFT UNIVERSITY OF TECHNOLOGY

REPORT 07-04

Theoretical and Numerical Comparison of Various Projection Methods derived from Deflation, Domain Decomposition and Multigrid Methods

J.M. Tang, R. Nabben, C. Vuik, Y.A. Erlangga

ISSN 1389-6520

Reports of the Department of Applied Mathematical Analysis

Delft 2007

Theoretical and Numerical Comparison of Various Projection Methods derived from Deflation, Domain Decomposition and Multigrid Methods [1]

J.M. Tang [2]   R. Nabben [3]   C. Vuik [4]   Y.A. Erlangga [5]

January, 2007

[1] Most of this work has been done during the visit of the first and third author at Technischen Universität Berlin in November 2006.

[2] Delft University of Technology, Faculty of Electrical Engineering, Mathematics and Computer Science, Delft Institute of Applied Mathematics, P.O. Box 5031, 2600 GA Delft, The Netherlands, phone: +3115-2787290, fax: +3115-2787209 (j.m.tang@tudelft.nl). Supported by the Dutch BSIK/BRICKS project.

[3] Technischen Universität Berlin, Institut für Mathematik, MA 3-3, Straße des 17. Juni 136, D-10623 Berlin, Germany, phone: +4930-31429291, fax: +4930-31429621 (nabben@math.tu-berlin.de). Supported by the Deutsche Forschungsgemeinschaft (DFG), Project NA248/2-2.

[4] Delft University of Technology, Faculty of Electrical Engineering, Mathematics and Computer Science, Delft Institute of Applied Mathematics, P.O. Box 5031, 2600 GA Delft, The Netherlands, phone: +3115-2785530, fax: +3115-2787209 (c.vuik@tudelft.nl)

[5] Technischen Universität Berlin, Institut für Mathematik, MA 3-3, Straße des 17. Juni 136, D-10623 Berlin, Germany, phone: +4930-31429290, fax: +4930-31429621 (erlangga@math.tu-berlin.de). Supported by the Deutsche Forschungsgemeinschaft (DFG), Project NA248/2-2.

Abstract

For various applications, it is well-known that a two-level-preconditioned Krylov method is an efficient method for solving large and sparse linear systems. Beside a traditional preconditioner like incomplete Cholesky decomposition, a projector has been included as preconditioner to get rid of a number of small and large eigenvalues of the matrix. In literature, various projection methods are known coming from the fields of deflation, domain decomposition and multigrid. From an abstract point of view, these methods are closely related. The aim of this paper is to compare these projection methods both theoretically and numerically using various elliptic test problems. We investigate their convergence properties and stability by considering implementation issues, rounding-errors, inexact coarse solves and severe termination criteria. Finally, we end up with a suggestion of the optimal second-level preconditioner, which is as stable as the abstract balancing preconditioner and as cheap and fast as the deflation preconditioner.

Keywords: deflation, domain decomposition, multigrid, preconditioning, Krylov methods, implementation, Poisson equation, hybrid methods, coarse grid corrections.

AMS subject classifications. 65F10, 65F50, 65N22, 65N55.

CONTENTS

LIST OF ALGORITHMS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

Introduction

The Conjugate Gradient (CG) method [13] is a very popular iterative method to solve large linear systems of equations

$$Ax = b, \quad A = [a_{ij}] \in \mathbb{R}^{n \times n}, \tag{1.1}$$

whose coefficient matrix A is sparse and symmetric positive semi-definite (SPSD). The convergence rate of CG depends on the condition number of the coefficient matrix of the linear system, i.e., after m iterations of CG, the error is bounded by,

$$||x - x_k||_A \leq 2||x - x_0||_A \left(\frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1} \right)^m , \tag{1.2}$$

where x_0 is the starting vector, $\kappa = \kappa(A, x_0)$ denotes the effective spectral condition number of A related to x_0, and $||x||_A$ is the $A-$norm of x, defined as $||x||_A = \sqrt{x^T A x}$. If κ is large it is advisable to solve, instead, a preconditioned system $M^{-1}Ax = M^{-1}b$, where the symmetric positive definite (SPD) preconditioner M^{-1} is chosen such that $M^{-1}A$ has a more clustered spectrum or a smaller condition number than that of A. Furthermore, systems $My = z$ must be cheap to solve relative to the improvement it provides in convergence rate.

Nowadays, the design and analysis of preconditioners for the CG method are in the main focus, whenever a linear system with SPSD coefficient matrix needs to be solved. Even fast solvers, like multigrid or domain decomposition methods, are used as preconditioners. Traditional preconditioners are diagonal scaling, basic iterative methods, approximate inverse preconditioning and incomplete Cholesky preconditioners. However, it appears that the resulting preconditioned CG method shows slow convergence in many applications with highly refined grids and flows with high coefficient ratios in the original differential equations. In these cases, the presence of small eigenvalues has a harmful influence on the convergence of preconditioned CG.

Nowadays, it appears that beside traditional preconditioners, a second level preconditioning can be used to get rid of the small eigenvalues. This new type is also known as projectors or projection methods where extra coarse linear systems have to be solved. Deflation is one of the frequently used projection methods, see e.g. [8, 15, 26, 29, 37]. Other typical examples of projectors are additive coarse grid correction [4, 5] and abstract balancing [16–18] methods which are well-known in the field of multigrid and domain decomposition methods.

Various projectors appear to be very useful for problems with large jumps in the coefficients, combined with domain decomposition methods [31, 34] and in combination with block-Jacobi type preconditioners in parallel computing [9, 36, 38]. Recently, it appeared that the two-level preconditioning can also be useful for problems with constant coefficients, which are solved on sequential computers [33].

At a first glance, the projectors from deflation, domain decomposition and multigrid seem not to be comparable. Eigenvector approximations are usually used as deflation projectors, whereas special projections are built to transfer information to the whole domain or to a coarser grid in the cases of domain decomposition and multigrid. However, from an abstract point of view, these projections are comparable, or even identical in some sense. In [23–25], theoretical comparisons are given of the deflation, abstract balancing and additive coarse grid correction projectors. It has been shown that the deflation method is expected to be faster in convergence than the additive coarse grid correction or abstract balancing method by considering e.g. the effective condition numbers, although the spectra of deflation and abstract balancing are almost the same. For certain starting vectors, deflation and abstract balancing even produce the same iterates. In the concise numerical experiments considering porous media flows, it appeared that although these projectors seem to be comparable, some of them are instable in the sense that the residuals of some methods stagnate or even diverge, if the required accuracy is (too) high. More recent papers about stability of projectors can also be found in e.g. [2, 11, 12].

The comparisons of deflation, abstract balancing and coarse grid correction were mainly based on theoretical aspects, whereas the numerical comparison had been done concisely in the references given above. Additionally, there are more attractive projection methods available in literature, which are not included in these comparisons. Moreover, some projection methods basically employ the same operators, where some slight differences can only be noticed in the numerical implementation. Therefore, in this paper we consider a wide set of projection methods used in different fields. Some more attractive projection methods, which are not known in literature, are also included in this set. First this set will be compared theoretically, by considering the corresponding spectral properties, their numerical implementation and equivalences. Then the main focus will be on the numerical experiments, where these methods will be tested on their convergence properties and stability. The effect of the different implementations will be analyzed extensively. The following questions will be investigated in this paper:

- which implementations are stable with respect to rounding errors?
- which implementations can be applied if one uses inaccurate coarse solvers, severe termination criteria and perturbed starting vectors?
- is there a second level preconditioner, which is as stable as abstract balancing and as cheap and fast as deflation?

Besides the preconditioners considered in this paper, some other variants are known as augmented subspace CG [7], deflated Lanczos method [29] and the Odir and Omin version of CG combined with extra vectors [1]. Since these methods have already been compared in [29], we refer to this overview paper for the details about this comparison, see also [30].

This paper is organized as follows. In Chapter 2, we introduce the methods and their algorithms which will be compared. Chapter 3 is devoted to the theoretical comparison of these methods. The main of this paper is the numerical comparison which is the topic of Chapter 4. Finally, the conclusions are drawn in Chapter 5.

CHAPTER 2

Notations, Methods and their Algorithms

In this chapter, we will give some notations of matrices applied through this paper. Subsequently, the methods will be defined and motivated, which will be further compared theoretically and numerically later on. It also includes their algorithms, because they are of importance during the comparisons.

2.1 Definition of Matrices

In Table 2.1 one can find the matrices used in this paper. We assume that $k \ll n \in \mathbb{N}$ and I is the identity matrix of appropriate size. In addition, note that in the definition of the (coarse grid) correction matrix Q, we assume that the inverse exists, and otherwise, the pseudo-inverse of E is used.

Meaning	Matrix	Dimensions
Given SPSD Coefficient matrix	A	$n \times n$
Given SPD Preconditioning matrix	M^{-1}	$n \times n$
Given Deflation Subspace matrix	Z	$n \times k$
Coarse matrix	$E := Z^T AZ$	$k \times k$
Correction matrix	$Q := ZE^{-1}Z^T$	$n \times n$
Deflation matrix	$P := I - AQ$	$n \times n$

Table 2.1: Notations of the matrices.

From an abstract point of view, all methods, which will be considered for comparison consist of an arbitrary SPD preconditioner M^{-1} combined with one or more deflation matrices P and/or correction matrices Q. In the next subsections, we will give a concise explanation and choices for the matrices in the different fields. Nevertheless, from our point of view, the given matrices M^{-1} and Z are just arbitrary but full rank matrices. This abstract setting allows us to compare the different approaches used in domain decomposition, multigrid and deflation.

2.1.1 Matrices in Domain Decomposition

In the projection methods used in domain decomposition, such as the balancing Neumann-Neumann or the (two-level) additive coarse grid correction method, the preconditioner M^{-1} consists of the local (exact or inexact) solves on the subdomains. For example, M^{-1} can be the additive Schwarz preconditioner. Moreover, Z describes a restriction operator, while Z^T is the prolongation or interpolation operator based on the subdomains, which are rectangular but full-ranked. In these cases, E is the coarse grid or Galerkin matrix and Q is called the coarse grid correction matrix. To speed up the convergence of the additive coarse grid correction method, a coarse grid correction Q can be added. Finally, matrix P can be seen as a subspace correction in which each subdomain is agglomerated into a single cell. More details can be found in [31,34].

2.1.2 Matrices in Multigrid

Like in the domain decomposition approach, Z is a restriction operator and Z^T is the prolongation operator in the multigrid field, with the difference that there can be a connection between some subdomains. Matrices E and Q are again the coarse grid/Galerkin and the coarse grid correction matrices, corresponding to the Galerkin approach. Matrix P can be seen as a coarse grid correction using an interpolation operator with extreme coarsening, where linear systems with E are usually solved recursively. In the context of multigrid projection methods, M^{-1} should work as a smoother before a coarse grid correction P. We refer to [35,40] for more details.

2.1.3 Matrices in Deflation

In the deflation projection methods, M^{-1} can be an arbitrary preconditioner such as the Incomplete Cholesky factorization. Furthermore, matrix Z consists of the so-called deflation vectors used in the deflation matrix P. In this case, the column space of Z builds the deflation subspace, i.e., the space to be projected out of the residuals. It consists of for example eigenvectors, approximations of eigenvectors, but also piecewise constant or linear vectors which are strongly related to domain decomposition. If one chooses for eigenvectors, the corresponding eigenvalues would be moved to zero in the spectra of PA. This fact has motivated the name deflation method. In literature it is also known as the spectral preconditioner, see also e.g. [11].

As mentioned earlier, P is the projection or deflation matrix, where the column space of Z is the deflation subspace. Since $k \ll n$ and that Z has rank k, coarse matrix E can be easily computed and factored and it is SPSD for any Z. Usually, systems with E are solved directly using e.g. Cholesky decomposition.

2.2 Linear System and Preconditioners

Before defining the projection methods, we first deal with the linear system in more detail and we will derive general combinations of separate preconditioners in this section. In this paper, 'seperate preconditioners' mean separate matrices like M^{-1}, Q and P without their additive or multiplicative combinations.

For the standard preconditioned CG method we solve

$$\mathcal{P}\mathcal{A}x = \mathfrak{b}, \quad \mathcal{P}, \mathcal{A} \in \mathbb{R}^{n \times n}, \tag{2.1}$$

where usually $\mathfrak{b} = M^{-1}b$ is the right-hand side, $\mathcal{A} = A$ represents the SPSD coefficient matrix and $\mathcal{P} = M^{-1}$ is an SPD and separate preconditioner. For example, one can apply CG with the preconditioner $\mathcal{P} = M^{-1}_{\text{PREC}}$, abbreviated by PREC in this paper.

Moreover, $\mathcal{A}$ can also be a combination of the SPSD matrix A and a projection matrix P such that $\mathcal{A}$ is still SPSD, while $\mathcal{P}$ is a standard separate preconditioner. This will be done in the deflation methods, see Subsection 2.3.1.

Subsequently, instead of choosing a separate preconditioner, different preconditioners/projectors can be combined in an additive or multiplicative way which can also be used for $\mathcal{P}$. This will be decribed below.

2.2.1 Additive Combination of Preconditioners

The additive combination of two separate SPSD preconditioners C_1 and C_2 leads to $\mathcal{P}_{a_2}$ which is the additive preconditioner consisting of two separate preconditioners, i.e.,

$$\mathcal{P}_{a_2} := C_1 + C_2, \tag{2.2}$$

which should be SPD. Of course the summation of the preconditioners can be done by different weightings of C_1 and C_2. Moreover, (2.2) can be easily generalized to $\mathcal{P}_{a_i}$ for more SPD preconditioners $C_1, C_2, \ldots, C_i$.

2.2.2 Multiplicative Combination of Preconditioners

The multiplicative combination of preconditioners can be easily explained by considering the stationary iterative methods induced by the preconditioner. We have

$$\begin{array}{rcl} x^{i+\frac{1}{2}} & = & x^i + C_1(b - Ax^i), \\ x^{i+1} & = & x^{i+\frac{1}{2}} + C_2(b - Ax^{i+\frac{1}{2}}), \end{array} \tag{2.3}$$

with two separate SPD preconditioners C_1 and C_2. This implies

$$x^{i+1} = x^i + \mathcal{P}_{m_2}(b - Ax^i), \tag{2.4}$$

where

$$\mathcal{P}_{m_2} := C_1 + C_2 - C_2AC_1 \tag{2.5}$$

is the multiplicative operator consisting of two separate preconditioners.

In addition, C_1 and C_2 can again be combined with another preconditioner C_3 in a multiplicative way, by taking $C_1 := \mathcal{P}_{m_2}$ and $C_2 := C_3$ in Expressions (2.3)–(2.5). This yields

$$\mathcal{P}_{m_3} = C_1 + C_2 + C_3 - C_2AC_1 - C_3AC_2 - C_3AC_1 + C_3AC_2AC_1. \tag{2.6}$$

Again this can be generalized to $\mathcal{P}_{m_i}$ for more separate SPSD preconditioners $C_1, C_2, \ldots, C_i$.

2.3 Definition of Methods

In this section, the projection methods will be given and motivated.

2.3.1 Deflation Methods

The deflation technique has been exploited by several authors, among them are [9, 10, 15, 19, 20, 22–24, 26, 29, 37]. Below we first describe the deflation method following [37]. Thereafter we repeat this procedure using another theoretically equivalent approach of the deflation method as used in [15, 19, 20, 26, 29].

Deflation Method

In order to solve $Ax = b$ we employ

$$x = (I - P^T)x + P^T x. \tag{2.7}$$

Because Q is symmetric and $(I-P^T)x = QAx = Qb$ can be immediately computed, we only need to compute $P^T x$ in (2.7). In light of the identity $AP^T = PA$ which will be shown in the next chapter, we solve the deflated system

$$PA\tilde{x} = Pb \tag{2.8}$$

for $\tilde{x}$ using CG. Then, because one can show that $P^T\tilde{x} = P^T x$, solution x can be obtained via (2.8) by premultiply $\tilde{x}$ by P^T and add it to Qb. Obviously, (2.8) is singular and an SPSD system can only be solved by CG as long as the right-hand side is consistent, i.e., as long as $b = Ax$ for some x see also [14]. If matrix A is nonsingular, than this is certainly true for (2.8), where the same projection is applied to both sides of the nonsingular system. If A is singular, then this projection can also be applied in many cases, see [32, 33].

Subsequently, the deflated system can also be solved by using an SPD preconditioner M^{-1}, leading to

$$\widetilde{P}\widetilde{A}\tilde{x} = \widetilde{P}b,$$

with

$$\widetilde{A} := M^{-1/2}AM^{-1/2}, \quad \widetilde{P} = I - \widetilde{A}Z(Z^T\widetilde{A}Z)^{-1}Z^T.$$

After some transformations this gives

$$M^{-1}PAx = M^{-1}Pb, \tag{2.9}$$

see [37] for details.

The linear system (2.9) is of the form of (2.1) by taking $\mathcal{P} = M^{-1}$, $\mathcal{A} = PA$ and $\mathfrak{b} = M^{-1}Pb$. Note that this is well-defined since it can be shown that $\mathcal{A}$ is still SPSD. The resulting method is called Deflation Method Variant 1 (DEF1).

Alternative Derivation of Deflation Method

Another way to describe the deflation technique is done by e.g. [15, 19, 20, 26, 29] which proceeds as follows.

Let x' be a random vector and suppose

$$x_0 := Qb + P^T x'.$$

Then, the solution of $Ax = b$ can be constructed in the form

$$x = x_0 + w, \tag{2.10}$$

where

$$w = x - x_0 = x - Qb - P^T x' = P^T(x - x'), \tag{2.11}$$

by noting that $Qb = QAx$ and $P^T = I - QA$. Due to (2.11), we have

$$w = P^T w, \tag{2.12}$$

since P^T is a projector satisfying $(P^T)^2 = P^T$. Moreover, premultiplying (2.10) by A yields

$$Aw = r_0, \quad r_0 := b - Ax_0. \tag{2.13}$$

Next, since (2.12) holds, w is also a solution of the deflated system

$$AP^T y = r_0. \tag{2.14}$$

Conversely, given a non-unique solution y of (2.14), we set $w := P^T y$ which provides the required solution of (2.13) that saisfies (2.12). Hence, the solution of $Ax = b$ can be constructed in the form

$$x = x_0 + P^T y, \tag{2.15}$$

where

$$x_0 := Qb + P^T x', \quad x' \text{ random}, \tag{2.16}$$

and y is the unique solution of the deflated system

$$AP^T y = r_0, \quad r_0 := b - Ax_0. \tag{2.17}$$

Again, the deflated system (2.17) can also be solved with the preconditioner M^{-1}, leading to

$$M^{-1}AP^T y = M^{-1}r_0, \tag{2.18}$$

and applying (2.15) to find solution x. After some rewriting, solution x can be uniquely determined from

$$P^T M^{-1}Ax = P^T M^{-1}b, \tag{2.19}$$

where the unique solution x can be found provided that x_0 as given in (2.16) has been used. We refer to e.g. [15] for more details.

The resulting abstract method with various known names is not only used in the field of deflation, but also in domain decomposition and multigrid, where the numerical implementation may differ although the linear system is identical, see also Section 2.5. In this paper, we call it Deflation Variant 2 (DEF2) and Reduced Balancing Variant 2 (R-BNN2) which are equivalent and only differ in the implementation. In fact, the implementation of DEF2 is equal to the approach as applied in e.g. [29], where the deflation method has been derived by combining a deflated Lanczos procedure and the standard CG algorithm. On the other hand, R-BNN2 is the approach where deflation has been incorporated into the CG algorithm in a direct way [15] but it is also the approach where a so-called hybrid variant has been employed in a domain decomposition method [34].

Note that (2.18) can be written in the form of (2.1) by taking $\mathcal{P} = M^{-1}$, $\mathcal{A} = AP^T$ which is SPSD and $\mathfrak{b} = M^{-1}r_0$. However, (2.19) is of a different type since it can obviously not be written as (2.1) with an SPD operator $\mathcal{P}$ and an SPSD matrix $\mathcal{A}$. Fortunately, we will see in the next chapter that both DEF2 and R-BNN2 are identical to a method, where the resulting linear system appears to be of the form of (2.1). Hence, DEF2 and R-BNN2 are appropriate methods and we denote their operators as

$$\mathcal{P}_{\text{DEF2}} = \mathcal{P}_{\text{R-BNN2}} = P^T M^{-1}. \tag{2.20}$$

Remark 2.1. *The difference between DEF1 and DEF2 is that their projection operators are flipped. Moreover, in DEF1 the 'uniqueness-step', which can be informally written as*

$$x = Qb + P^T x, \tag{2.21}$$

has been done at the end so that one can use an arbitrarily chosen starting vector x_0. On the contrary, this uniqueness step 2.21 has been carried out in the beginning of DEF2 which can be interpreted as adopting a 'special' starting vector.

2.3.2 Additive Method

If one substitutes an arbitrary preconditioner $C_1 = M^{-1}$ and a coarse grid correction matrix $C_2 = Q$ into the additive combination as given in (2.2), then this implies

$$\mathcal{P}_{\text{AD}} = M^{-1} + Q. \tag{2.22}$$

Using the additive Schwarz preconditioner for M^{-1}, the abstract form (2.22) includes the additive coarse grid correction preconditioner introduced by Bramble et al. [3]. This operator has further been analyzed by e.g. [4, 5, 27]. Moreover, if the multiplicative Schwarz preconditioner has been taken as M^{-1}, we obtain the so-called Hybrid-2 preconditioner, see [34, p. 47]. In the multigrid language this preconditioner is sometimes called an additive multigrid preconditioner. In this paper, we call the resulting method with the operator $\mathcal{P}_{\text{AD}}$ the additive (AD) method for convenience.

2.3.3 Adapted Deflation Methods

We take again $C_1 = Q$ and $C_2 = M^{-1}$, but now as a multiplicative combination as given in (2.5) which implies

$$\begin{aligned} \mathcal{P}_{\text{A-DEF1}} &= M^{-1} + Q - M^{-1}AQ \\ &= M^{-1}P + Q. \end{aligned} \tag{2.23}$$

In the multigrid language this preconditioner results from the non-symmetric multigrid iteration scheme, where one first applies a coarse grid correction followed by a smoothing step. Note that, although Q and M^{-1} are SPSD preconditioners, (2.23) is a non-symmetric operator and, even more, it is not symmetric with respect to the inner product induced by A.

In addition, $\mathcal{P}_{\text{A-DEF1}}$ as given in (2.23) can also be seen as an adapted deflation preconditioner, since the deflation preconditioner $M^{-1}P$ is combined in an additive way with a coarse grid correction Q. Hence, we call the method associated to the operator $\mathcal{P}_{\text{A-DEF1}}$ the adapted deflation variant 1 (A-DEF1) method.

Subsequently, we can also reverse the order of Q and M^{-1} in (2.5), i.e., we choose $C_1 = M^{-1}$ and $C_2 = Q$, then Expression (2.5) yields

$$\begin{aligned} \mathcal{P}_{\text{A-DEF2}} &= Q + M^{-1} - QAM^{-1} \\ &= P^TM^{-1} + Q. \end{aligned} \tag{2.24}$$

Using the additive Schwarz preconditioner for M^{-1}, the preconditioner $\mathcal{P}_{\text{A-DEF2}}$ is called the two-level Hybrid-II Schwarz preconditioner in [31, p. 48]. In the multigrid methods, M^{-1} is used as a smoother and then $\mathcal{P}_{\text{A-DEF2}}$ is the (non-symmetric) multigrid preconditioner. Again, $\mathcal{P}_{\text{A-DEF2}}$ is non-symmetric, also with respect to the inner product induced by A. Fortunately, in the next chapter we will see that A-DEF2 with special choices of the starting vector is identical to the abstract balancing method, which is based on a symmetric operator.

As in the case of $\mathcal{P}_{\text{A-DEF1}}$, the operator $\mathcal{P}_{\text{A-DEF2}}$ can also be seen as an adapted deflation preconditioner, since the deflation preconditioner P^TM^{-1} is combined with a coarse grid correction Q in an additive way. Therefore, we call the method corresponding to $\mathcal{P}_{\text{A-DEF2}}$ the adapted deflation variant 2 (A-DEF2) method.

2.3.4 Abstract Balancing Methods

The operators $\mathcal{P}_{\text{A-DEF1}}$ and $\mathcal{P}_{\text{A-DEF2}}$ can be symmetrized by using the multiplicative combination of three preconditioners. If one takes $C_1 = Q,\ C_2 = M^{-1}$ and $C_3 = Q$

in Expression (2.6), we obtain

$$\begin{aligned}\mathcal{P}_{\text{BNN}} &= M^{-1} + 2Q - M^{-1}AQ - QAM^{-1} - QAQ + QAM^{-1}AQ \\ &= Q + M^{-1}(I - AQ) - QAM^{-1}(I - AQ) \\ &= P^T M^{-1} P + Q.\end{aligned}$$

In the multigrid language this preconditioner results from a symmetric multigrid iteration scheme, where one first applies a coarse grid correction followed by a smoothing step and ended with another coarse grid correction.

In combination with the additive Schwarz preconditioner for M^{-1}, and after some scaling and special choices of Z, $\mathcal{P}_{\text{BNN}}$ is well-known as the Balancing-Neumann-Neumann preconditioner, introduced by Mandel [16]. It has further been analyzed by e.g. [6, 17, 18, 28, 34]. In the abstract form, $\mathcal{P}_{\text{BNN}}$ is called the Hybrid-1 preconditioner in [34, p. 34]. Here, we call it the abstract balancing preconditioner (BNN) following [16–18].

Moreover, we consider two variants of the BNN method. In the first variant we omit the seperate term Q in $\mathcal{P}_{\text{BNN}}$ giving us

$$\mathcal{P}_{\text{R-BNN1}} = P^T M^{-1} P,$$

which still remains a symmetric operator. It is an operator which is rather unknown in literature and therefore it can be interesting to investigate its properties. The corresponding method is called the reduced balancing variant 1 (R-BNN1) method.

Furthermore, the second variant of BNN is called the reduced balancing variant 2 (R-BNN2) method and has already been defined in Subsection 2.3.1, since it is strongly related to the deflation method. We recall that the corresponding operator has the following form:

$$\mathcal{P}_{\text{R-BNN2}} = P^T M^{-1}.$$

As mentioned in Subsection 2.3.1, this is a non-symmetric operator, but in the next chapter we however will see that both $\mathcal{P}_{\text{R-BNN1}}$ and $\mathcal{P}_{\text{R-BNN2}}$ are identical to $\mathcal{P}_{\text{BNN}}$ in special cases. Therefore, we classify these methods as variants of the original abstract balancing method rather than on variants of deflation methods.

2.4 Table of the Methods

After introducing the methods, we give all operators of the methods which will be used for comparison in Table 2.2. In the CG method they can be interpreted as the preconditioners $\mathcal{P}$ as in (2.1) with $\mathcal{A} = A$.

Remark 2.2. *Denote $\mathcal{P}_{\gamma_i}$ the class of operators of*

$$\gamma_1 M + \gamma_2 P^T M^{-1} + \gamma_3 P^T M^{-1} P + \gamma_4 M^{-1} P + \gamma_5 Q \tag{2.25}$$

with $\gamma_i = \{0, 1\}$ for $i = 1, \ldots, 5$ such that $\sum_{i=1}^{4} \gamma_i = 1$. Then each of the methods given in Table 2.2 belongs to the class of $\mathcal{P}_{\gamma_i}$.

Remark 2.3. *Recall that both DEF2 and R-BNN2 are based on the same operator and differ only in the numerical implementation, see Section 2.5.*

2.5 Implementation of the Methods

In this section we consider the implementation of the methods by considering their algorithms. This will be done in order to analyze the proposed methods with respect to the amount of work, rounding errors, stability etc.

No.	Abbreviation	Method	Operator
0	PREC	Classical Preconditioner	M^{-1}
1	AD	Additive Coarse Grid Correction	$M^{-1}+Q$
2	DEF1	Deflation Variant 1	$M^{-1}P$
3	DEF2	Deflation Variant 2	P^TM^{-1}
4	A-DEF1	Adapted Deflation Variant 1	$M^{-1}P+Q$
5	A-DEF2	Adapted Deflation Variant 2	$P^TM^{-1}+Q$
6	BNN	Abstract Balancing	$P^TM^{-1}P+Q$
7	R-BNN1	Reduced Balancing Variant 1	$P^TM^{-1}P$
8	R-BNN2	Reduced Balancing Variant 2	P^TM^{-1}

Table 2.2: List of methods which will be compared.

2.5.1 Algorithms

The implementation of all methods we will compare can be found in Algorithms 1–9. If possible, in the caption of each algorithm we give the references to the most related implementations known in the literature.

Algorithm 1 PREC (M^{-1}) solving $Ax=b$, [21]

```
$x_0$ random, $r_0 := b - Ax_0$
$z_0 := M^{-1}r_0,\ p_0 := z_0$
for $j := 0, \ldots,$ until convergence do
    $w_j := Ap_j$
    $\alpha_j := (r_j, z_j)/(p_j, w_j)$
    $x_{j+1} := x_j + \alpha_j p_j$
    $r_{j+1} := r_j - \alpha_j w_j$
    Precondition: $z_{j+1} := M^{-1}r_{j+1}$
    $\beta_j := (r_{j+1}, z_{j+1})/(r_j, z_j)$
    $p_{j+1} := z_{j+1} + \beta_j p_j$
end for
$x := x_{j+1}$
```

Algorithm 2 AD ($M^{-1}+Q$) solving $Ax=b$, [31]

```
$x_0$ random, $r_0 := b - Ax_0$
$z_0 := M^{-1}r_0 + Qr_0,\ p_0 := z_0$
for $j := 0, \ldots,$ until convergence do
    $w_j := Ap_j$
    $\alpha_j := (r_j, z_j)/(p_j, w_j)$
    $x_{j+1} := x_j + \alpha_j p_j$
    $r_{j+1} := r_j - \alpha_j w_j$
    Precondition: $\hat{z}_{j+1} := M^{-1}r_{j+1}$
    Correction: $\tilde{z}_{j+1} := Qr_{j+1}$
    $z_{j+1} := \hat{z}_{j+1} + \tilde{z}_{j+1}$
    $\beta_j := (r_{j+1}, z_{j+1})/(r_j, z_j)$
    $p_{j+1} := z_{j+1} + \beta_j p_j$
end for
$x := x_{j+1}$
```

Algorithm 3 DEF1 ($M^{-1}P$) solving $Ax = b$, [37]

```
$x_0$ random, $r_0 := b - Ax_0$
$\hat{r}_0 = Pr_0,\ z_0 := M^{-1}\hat{r}_0,\ p_0 := z_0$
for $j := 0, \dots,$ until convergence do
    $w_j := Ap_j$
    Projection: $\hat{w}_j := Pw_j$
    $\alpha_j := (\hat{r}_j, z_j)/(p_j, \hat{w}_j)$
    $\tilde{x}_{j+1} := \tilde{x}_j + \alpha_j p_j$
    $\hat{r}_{j+1} := \hat{r}_j - \alpha_j \hat{w}_j$
    Precondition: $z_{j+1} := M^{-1}\hat{r}_{j+1}$
    $\beta_j := (\hat{r}_{j+1}, z_{j+1})/(\hat{r}_j, z_j)$
    $p_{j+1} := z_{j+1} + \beta_j p_j$
end for
$x := Qb + P^T x_{j+1}$
```

Algorithm 4 DEF2 ($P^T M^{-1}$) solving $Ax = b$, [29]

```
$x_0$ random, $x_s := Qb + P^T x_0,\ r_0 := b - Ax_s$
$z_0 := M^{-1}r_0,\ p_0 := P^T z_0$
for $j := 0, \dots,$ until convergence do
    $w_j := Ap_j$
    $\alpha_j := (r_j, z_j)/(p_j, w_j)$
    $x_{j+1} := x_j + \alpha_j p_j$
    $r_{j+1} := r_j - \alpha_j w_j$
    Precondition: $z_{j+1} := M^{-1}r_{j+1}$
    $\beta_j := (r_{j+1}, z_{j+1})/(r_j, z_j)$
    Projection: $y_{j+1} = P^T z_{j+1}$
    $p_{j+1} := y_{j+1} + \beta_j p_j$
end for
$x := x_{j+1}$
```

Algorithm 5 A-DEF1 ($M^{-1}P + Q$) solving $Ax = b$, [31]

```
$x_0$ random, $r_0 := b - Ax_0$
$z_0 := M^{-1}Pr_0 + Qr_0,\ p_0 := z_0$
for $j := 0, \dots,$ until convergence do
    $w_j := Ap_j$
    $\alpha_j := (r_j, z_j)/(p_j, w_j)$
    $x_{j+1} := x_j + \alpha_j p_j$
    $r_{j+1} := r_j - \alpha_j w_j$
    Projection: $y_{j+1} := Pr_{j+1}$
    Precondition: $\tilde{z}_{j+1} := M^{-1}y_{j+1}$
    Correction: $\hat{z}_{j+1} := Qr_{j+1}$
    $z_{j+1} := \tilde{z}_{j+1} + \hat{z}_{j+1}$
    $\beta_j := (r_{j+1}, z_{j+1})/(r_j, z_j)$
    $p_{j+1} := z_{j+1} + \beta_j p_j$
end for
$x := x_{j+1}$
```

Algorithm 6 A-DEF2 ($P^TM^{-1}+Q$) solving $Ax=b$, [31]

```
$x_0$ random, $x_s := Qb + P^Tx_0,\ r_0 := b - Ax_s$
$z_0 := P^TM^{-1}r_0 + Qr_0,\ p_0 := z_0$
for $j := 0, \ldots,$ until convergence do
    $w_j := Ap_j$
    $\alpha_j := (r_j, z_j)/(p_j, w_j)$
    $x_{j+1} := x_j + \alpha_j p_j$
    $r_{j+1} := r_j - \alpha_j w_j$
    Precondition: $y_{j+1} := M^{-1}r_{j+1}$
    Projection: $\hat{z}_{j+1} := P^Ty_{j+1}$
    Correction: $\tilde{z}_{j+1} := Qr_{j+1}$
    $z_{j+1} := \hat{z}_{j+1} + \tilde{z}_{j+1}$
    $\beta_j := (r_{j+1}, z_{j+1})/(r_j, z_j)$
    $p_{j+1} := z_{j+1} + \beta_j p_j$
end for
$x := x_{j+1}$
```

Algorithm 7 BNN ($P^TM^{-1}P+Q$) solving $Ax=b$, [16]

```
$x_0$ random, $r_0 := b - Ax_0$
$z_0 := P^TM^{-1}Pr_0 + Qr_0,\ p_0 := z_0$
for $j := 0, \ldots,$ until convergence do
    $w_j := Ap_j$
    $\alpha_j := (r_j, z_j)/(p_j, w_j)$
    $x_{j+1} := x_j + \alpha_j p_j$
    $r_{j+1} := r_j - \alpha_j w_j$
    Projection: $\hat{y}_{j+1} := Pr_{j+1}$
    Precondition: $y_{j+1} := M^{-1}\hat{y}_{j+1}$
    Projection: $\hat{z}_{j+1} := P^Ty_{j+1}$
    Correction: $\tilde{z}_{j+1} := Qr_{j+1}$
    $z_{j+1} := \hat{z}_{j+1} + \tilde{z}_{j+1}$
    $\beta_j := (r_{j+1}, z_{j+1})/(r_j, z_j)$
    $p_{j+1} := z_{j+1} + \beta_j p_j$
end for
$x := x_{j+1}$
```

Algorithm 8 R-BNN1 ($P^TM^{-1}P$) solving $Ax=b$

```
$x_0$ random, $x_s := Qb + P^Tx_0,\ r_0 := b - Ax_s$
$z_0 := P^TM^{-1}Pr_0,\ p_0 := z_0$
for $j := 0, \ldots,$ until convergence do
    $w_j := Ap_j$
    $\alpha_j := (r_j, z_j)/(p_j, w_j)$
    $x_{j+1} := x_j + \alpha_j p_j$
    $r_{j+1} := r_j - \alpha_j w_j$
    Projection: $\hat{y}_{j+1} := Pr_{j+1}$
    Precondition: $y_{j+1} := M^{-1}\hat{y}_{j+1}$
    Projection: $z_{j+1} := P^Ty_{j+1}$
    $\beta_j := (r_{j+1}, z_{j+1})/(r_j, z_j)$
    $p_{j+1} := z_{j+1} + \beta_j p_j$
end for
$x := x_{j+1}$
```

Algorithm 9 R-BNN2 ((P^TM^{-1})) solving $Ax = b$, [34]

```
$x_0$ random, $x_s := Qb + P^T x_0,\ r_0 := b - Ax_s$
$z_0 := P^T M^{-1} r_0,\ p_0 := z_0$
for $j := 0, \ldots,$ until convergence do
    $w_j := Ap_j$
    $\alpha_j := (r_j, z_j)/(p_j, w_j)$
    $x_{j+1} := x_j + \alpha_j p_j$
    $r_{j+1} := r_j - \alpha_j w_j$
    Precondition: $y_{j+1} := M^{-1} r_{j+1}$
    Projection: $z_{j+1} := P^T y_{j+1}$
    $\beta_j := (r_{j+1}, z_{j+1})/(r_j, z_j)$
    $p_{j+1} := z_{j+1} + \beta_j p_j$
end for
$x := x_{j+1}$
```

2.5.2 Projection and Precondition Steps

In each algorithm except for PREC, one or more projection and precondition steps have to be carried out. Except for DEF1 and DEF2, the projection and precondition steps are basically combined resulting in the preconditioned/projected residuals z_{j+1}, which consists of the whole operator based on the projections and preconditioners. DEF2 is the only method where a projection or precondition step has been applied on the search directions p_{j+1}. Finally, DEF1 is the only method where one substitutes PA into A in the beginning of each iteration step, i.e., the projection has been carried out on w_j.

2.5.3 Starting Vectors

Note that in DEF2, A-DEF2, R-BNN1 and R-BNN2, the random starting vector x_0 is used to create a new 'special' starting vector x_s. These special choices appear to be crucial in these methods, which will be further explained in the next chapter. Furthermore, in the other methods, there is no need to compute and use x_s.

2.5.4 Termination Criteria

In all methods we terminate the iterative process if the norm of the relative preconditioned/projected residual, i.e., $||z_{j+1}||_2/||z_1||_2$, is below a tolerance ϵ or if the maximum allowed number of iterations has been reached. Note that this is done for convenience, since z_{j+1} has different meanings for each method. For example, z_{j+1} is the preconditioned update residual in DEF2, whereas z_{j+1} represents the preconditioned-projected update residual in R-BNN2. Although the comparisons of the methods look unfair due to this fact, they appear to be comparable by also comparing the real errors in the numerical experiments.

2.5.5 SPD Preconditioner

As earlier mentioned in this chapter, $\mathcal{P}$ as given in Expression (2.1) should be SPD to guarantee convergence of CG. This is however only the case for PREC, AD and BNN. Due to x_s, it can be proven that the methods DEF2, A-DEF2, R-BNN1 and R-BNN2 can be transformed into BNN (see next chapter), so in fact they are equivalent to a method with an SPD preconditioner. Note that in general this does not hold anymore if one perturbs the starting vector x_s. This will be further investigated in Chapter 4.

Moreover, recall that the operator $M^{-1}P$ of DEF1 is also proper, because in this method we based the SPD preconditioner M^{-1} on the SPSD matrix PA rather than on A. Since PA is singular, we correct the solution x_{j+1} at the end so that it leads to the unique solution x. Note furthermore that if we implement the method in a naive way by combining $M^{-1}P$ in one step in the algorithm, we obtain a method, which will generally not work in combination with CG .

Finally, A-DEF1 is the only method which does not have an SPD operator and which can not be decomposed or transformed into an SPD preconditioner. Nonetheless, we will see that it works properly in a lot of numerical experiments, see Chapter 4.

2.5.6 Computational Costs

Below we give a sketch of the required computational costs for each iterate of the methods. For now it is rather difficult to give detailed information about these costs, because it depends for example on the choice of M and Z and, moreover, on the way of implementation and storage of the matrices.

First we give the required operations needed for each iterate in PREC, see Table 2.3.

Operation	**Number**
Matrix-vector multiplication (MVM)	1
Inner products (IP)	4
Vector updates (VU)	3
Preconditioning	1

Table 2.3: Computational costs of each iterate of PREC.

Next, we present the efficient implementation of the extra operations Py, $P^T y$ and Qy for an arbitrary vector y which has to be carried out in the projection methods, see Algorithms 10–12. Moreover, the computational work of these algorithms is given in Table 2.4. Note that E and its Cholesky decomposition and AZ are computed and stored beforehand, so that matrix-vector multiplications with A are not required in the algorithms. Moreover, we distinguish two cases considering Z and AZ:

- Z is sparse and both Z and AZ can each be stored in one vector;
- Z is full and therefore, Z and AZ are full matrices.

Algorithm 10 Operation Py

```
$y_1 := Z^T y$
solve $Ey_2 = y_1$
$y_3 := (AZ)y_2$
$Py := y - y_3$
```

Remark 2.4. *If both $P\hat{y}$ and $Q\hat{y}$ should be computed for the same vector $\hat{y}$, like in A-DEF1 and BNN, then the first two steps of Algorithm 12 are not needed. In this case, $Q\hat{y}$ only requires one inner product if Z is sparse or one matrix-vector multiplication if Z is full.*

Algorithm 11 Operation $P^T y$

```
$y_1 := (AZ)^T y$
solve $Ey_2 = y_1$
$y_3 := Zy_2$
$P^T y := y - y_3$
```

Algorithm 12 Operation Qy

```
$y_1 := Z^T y$
solve $Ey_2 = y_1$
$y_3 := Zy_2$
```

(a) Z is full.

Operation	$Py, P^T y$	Qy
Matrix-vector multiplication (MVM)	2	2
Inner products (IP)	0	0
Vector updates (VU)	1	0
Coarse System Solves (CSS)	1	1

(b) Z is sparse.

Operation	$Py, P^T y$	Qy
Matrix-vector multiplication (MVM)	0	0
Inner products (IP)	2	2
Vector updates (VU)	1	0
Coarse System Solves (CSS)	1	1

Table 2.4: Computational costs of Py, $P^T y$ and Qy.

Subsequently, the extra computations for each iterate in the projection methods with respect to PREC are presented in Table 2.5. Recall that AZ and E^{-1} have already been computed before the iteration process starts. Obviously, AD is the cheapest method, while BNN and R-BNN1 are the most expensive projection methods. Finally, observe that using a projection method is only efficient if Z is sparse or if the number of deflation vectors is relatively small in the case of a full Z.

(a) Z is full.

Method	Py, P^Ty	Qy	**MVM**	**IP**	**VU**	**CCS**
AD	0	1	2	0	0	1
DEF1	1	0	2	0	1	1
DEF2	1	0	2	0	1	1
A-DEF1	1	1	3	0	1	1
A-DEF2	1	1	4	0	1	2
BNN	2	1	5	0	2	2
R-BNN1	2	0	4	0	2	2
R-BNN2	1	0	2	0	1	1

(b) Z is sparse.

Method	Py, P^Ty	Qy	**MVM**	**IP**	**VU**	**CCS**
AD	0	1	0	2	0	1
DEF1	1	0	0	2	1	1
DEF2	1	0	0	2	1	1
A-DEF1	1	1	0	3	1	1
A-DEF2	1	1	0	4	1	2
BNN	2	1	0	5	2	2
R-BNN1	2	0	0	4	2	2
R-BNN2	1	0	0	2	1	1

Table 2.5: Extra computational costs of each iterate of the projection methods compared to PREC. IP = inner products, VU = vector updates and CCS = coarse system solves.

CHAPTER 3 Theoretical Comparison

This chapter is devoted to give a theoretical comparison between the methods defined in the previous chapter. We start with some preliminaries in Section 3.1. Subsequently, a comparison of the eigenvalue distributions of the operators associated to the projection methods will be given in Section 3.2. Finally, we show that the abstract balancing method and some other projection methods are equal in exact arithmetic, see Section 3.3.

3.1 Preliminary Results

In this section some preliminary results are presented which are needed in some proofs in this chapter. We start with some results from linear algebra, where $\sigma(A) = \{\lambda_1, \lambda_2, \ldots, \lambda_n\}$ denotes the spectrum of an arbitrary matrix A with eigenvalues λ_i.

Lemma 3.1. *Let $A, B \in \mathbb{R}^{n\times n}$ be arbitrary invertible or singular matrices. Then the following equations hold:*

(a) $\sigma(AB) = \sigma(BA)$*;*

(b) $\sigma(A+I) = \sigma(A) + \sigma(I)$*;*

(c) $\sigma(A) = \sigma(A^T)$*;*

Proof. (a) Let λ and v be an eigenvalue and corresponding eigenvector of AB. We consider two cases:

- λ is nonzero: the corresponding v satisfies $ABv \neq \mathbf{0}$, so in particular we have $Bv \neq \mathbf{0}$. Then the following equations are equivalent

$$\begin{array}{rcl} ABv &=& \lambda v; \\ BABv &=& \lambda Bv; \\ BAw &=& \lambda w, \end{array}$$

  where $w = Bv \neq \mathbf{0}$.

- λ is zero: we have

$$\det(AB) = \det(BA) = 0,$$

  and hence, if λ is a zero-eigenvalue of AB then it is also a zero-eigenvalue of BA.

In other words, for both cases λ is an eigenvalue of both AB and BA.

(b) Let λ and v be an eigenvalue and corresponding eigenvector of $A+I$, then the following equations are equivalent

$$\begin{aligned}(A+I)v &= \lambda v;\\ Av &= (\lambda-1)v;\\ Av &= \mu v,\end{aligned}$$

where $\mu=\lambda-1$. In other words, λ is an eigenvalue of $A+I$ is equivalent to $\lambda-1$ is an eigenvalue of A.

(c) By definition of determinants, we have $\det(A-\lambda I)=\det(A^T-\lambda I)$ for all λ. □

Subsequently, we give some preliminaries on the matrices as defined in Table 2.1, see Lemma 3.2. Most of them have been proven in literature, but for completeness we also state these proofs here.

Lemma 3.2. *The following equalities hold:*

(a) $P=P^2$*;*

(b) $AP^T=PA$*;*

(c) $QA=I-P^T$*;*

(d) $P^TZ=\mathbf{0}$*;*

(e) $PAZ=\mathbf{0}$*;*

(f) $Q^T=Q$*;*

(g) $QAQ=Q$

(h) $QAP^T=\mathbf{0}$*;*

(i) $QP=\mathbf{0}$*;*

(j) $QAZ=Z$*.*

Proof. (a)

$$\begin{aligned}P^2 &= (I-AZE^{-1}Z^T)(I-AZE^{-1}Z^T)\\ &= I-2AZE^{-1}Z^T+AZE^{-1}Z^TAZE^{-1}Z^T\\ &= I-2AZE^{-1}Z^T+AZE^{-1}Z^T\\ &= I-AZE^{-1}Z^T\\ &= P.\end{aligned}$$

(b)

$$\begin{aligned}AP^T &= A(I-ZE^{-1}Z^TA)\\ &= A-AZE^{-1}Z^TA\\ &= (I-AZE^{-1}Z^T)A\\ &= PA.\end{aligned}$$

(c)

$$\begin{aligned}QA &= ZE^{-1}Z^TA\\ &= I-(I-ZE^{-1}Z^TA)\\ &= I-P^T.\end{aligned}$$

(d)

$$\begin{aligned}P^TZ &= Z-ZE^{-1}Z^TAZ\\ &= Z-Z\\ &= \mathbf{0}.\end{aligned}$$

(e)

$$\begin{aligned} PAZ &= AZ - AZE^{-1}Z^TAZ \\ &= AZ - AZ \\ &= \mathbf{0}. \end{aligned}$$

(f)

$$\begin{aligned} Q^T &= (ZE^{-1}Z^T)^T \\ &= ZE^{-1}Z^T \\ &= Q. \end{aligned}$$

(g)

$$\begin{aligned} QAQ &= (ZE^{-1}Z^T)A(ZE^{-1}Z^T) \\ &= ZE^{-1}Z^T \\ &= Q. \end{aligned}$$

(h)

$$\begin{aligned} QAP^T &= QA(I - QA) \\ &= QA - QAQA \\ &= QA - QA \\ &= \mathbf{0}. \end{aligned}$$

(i)

$$\begin{aligned} QP &= Q(I - AQ) \\ &= Q - QAQ \\ &= Q - Q \\ &= \mathbf{0}. \end{aligned}$$

(j)

$$\begin{aligned} QAZ &= (ZE^{-1}Z^T)AZ \\ &= Z. \end{aligned}$$

□

3.2 Comparison of Spectra of Projection Methods

From the literature we already know that the eigenvalue distribution of the operator corresponding to PREC is always worse than e.g. AD, DEF1 and BNN. For example, in [39] we have shown that

$$\tilde{\kappa}\left(M^{-1}PA\right) < \kappa\left(M^{-1}A\right), \tag{3.1}$$

for any SPD matrices A and M and an arbitrary full-ranked Z. This means that the effective condition number of DEF1 is always below the condition number of PREC. It appears that the effective condition number of PREC is always larger than those of the other projection methods. Therefore, we restrict ourselves to the projection methods in this chapter.

Next, in [23, 24] it has been shown that the effective condition number of DEF1 is below the condition number of both AD and BNN, i.e.,

$$\tilde{\kappa}\left(M^{-1}PA\right) < \kappa\left(M^{-1}A + QA\right), \tag{3.2}$$

and

$$\tilde{\kappa}\left(M^{-1}PA\right) < \kappa\left(P^TM^{-1}PA + QA\right), \tag{3.3}$$

for all full-ranked Z and SPD matrices A and M^{-1}.

Besides the comparisons of AD, DEF1 and BNN done in e.g. [23–25], some more relations between the eigenvalue distribution of these and the other projection methods can be found below.

First we show that the operators of DEF1, DEF2, R-BNN1 and R-BNN2 have the same spectra.

Lemma 3.3. *The spectra corresponding to the operators of DEF1, DEF2, R-BNN1 and R-BNN2 are the same, i.e.,*

$$\sigma\left(M^{-1}PA\right) = \sigma\left(P^TM^{-1}A\right) = \sigma\left(P^TM^{-1}PA\right).$$

Proof. (i) Equality $\sigma\left(M^{-1}PA\right) = \sigma\left(P^TM^{-1}A\right)$ follows from

$$\sigma\left(M^{-1}PA\right) =^{\text{L3.1a}} \sigma\left(AM^{-1}P\right) =^{\text{L3.1c}} \sigma\left(P^TM^{-1}A\right).$$

(ii) Equality $\sigma\left(M^{-1}PA\right) = \sigma\left(P^TM^{-1}PA\right)$ holds since

$$\begin{aligned} \sigma\left(M^{-1}PA\right) &=^{\text{L3.2a}} \sigma\left(M^{-1}P^2A\right) \\ &=^{\text{L3.2b}} \sigma\left(M^{-1}PAP^T\right) \\ &=^{\text{L3.1a, L3.2d,e}} \sigma\left(P^TM^{-1}PA\right). \end{aligned}$$

□

Next, we show that the eigenvalues of the corresponding operators of BNN, A-DEF1 and A-DEF2 are identical.

Lemma 3.4. *The spectra of BNN, A-DEF1 and A-DEF2 are the same, i.e.,*

$$\sigma\left((P^TM^{-1}P + Q)A\right) = \sigma\left((M^{-1}P + Q)A\right) = \sigma\left((P^TM^{-1} + Q)A\right).$$

Proof. (i) Equality $\sigma\left((P^TM^{-1}P + Q)A\right) = \sigma\left((M^{-1}P + Q)A\right)$ follows from

$$\begin{aligned} \sigma\left(P^TM^{-1}PA + QA\right) &=^{\text{L3.2c}} \sigma\left(P^TM^{-1}PA - P^T + I\right) \\ &=^{\text{L3.1b}} \sigma\left(P^T(M^{-1}PA - I)\right) + \sigma(I) \\ &=^{\text{L3.1a}} \sigma\left((M^{-1}PA - I)P^T\right) + \sigma(I) \\ &=^{\text{L3.2b}} \sigma\left(M^{-1}P^2A - P^T\right) + \sigma(I) \\ &=^{\text{L3.1b}} \sigma\left(M^{-1}P^2A - P^T + I\right) \\ &=^{\text{L3.2c}} \sigma\left(M^{-1}P^2A + QA\right) \\ &=^{\text{L3.2a}} \sigma\left(M^{-1}PA + QA\right). \end{aligned}$$

(ii) Equality $\sigma\left((P^TM^{-1} + Q)A\right) = \sigma\left((P^TM^{-1}P + Q)A\right)$ is true because

$$\begin{aligned} \sigma\left(P^TM^{-1}A + QA\right) &=^{\text{L3.2c}} \sigma\left(P^TM^{-1}A - P^T + I\right) \\ &=^{\text{L3.1b}} \sigma\left(P^TM^{-1}A - P^T\right) + \sigma(I) \\ &=^{\text{L3.1c}} \sigma\left(AM^{-1}P - P\right) + \sigma(I) \\ &=^{\text{L3.2a}} \sigma\left(AM^{-1}P^2 - P\right) + \sigma(I) \\ &=^{\text{L3.1a}} \sigma\left(PAM^{-1}P - P\right) + \sigma(I) \\ &=^{\text{L3.1c}} \sigma\left(P^TM^{-1}AP^T - P^T\right) + \sigma(I) \\ &=^{\text{L3.2b}} \sigma\left(P^TM^{-1}PA - P^T\right) + \sigma(I) \\ &=^{\text{L3.1b}} \sigma\left(P^TM^{-1}PA - P^T + I\right) \\ &=^{\text{L3.2c}} \sigma\left(P^TM^{-1}PA + QA\right). \end{aligned}$$

□

As a consequence, the methods DEF1, DEF2, R-BNN1 and R-BNN2 can be seen as one class of projection methods. Conversely, BNN, A-DEF1 and A-DEF2 form another class of projection methods. These two classes can be connected by Theorem 2.8 of [24], which states that if $\sigma(M^{-1}PA) = \{0, \ldots, 0, \mu_{k+1}, \ldots, \mu_n\}$, is given, then $\sigma(P^TM^{-1}PA + QA) = \{1, \ldots, 1, \mu_{k+1}, \ldots, \mu_n\}$. It appears that the reverse statement also holds. For completeness, these results are given in Theorem 3.1.

Theorem 3.1. *Let the spectra of DEF1 and BNN be given by*

$$\sigma(M^{-1}PA) = \{\lambda_1, \ldots, \lambda_n\}, \quad \sigma(P^TM^{-1}PA + QA) = \{\mu_1, \ldots, \mu_n\},$$

respectively. Then, the order of the eigenvalues within these spectra can be changed such that

$$\lambda_i = 0, \quad \mu_i = 1, \quad i = 1, \ldots, k.$$

and

$$\lambda_i = \mu_i, \quad i = k+1, \ldots, n.$$

Proof. First of all, by using Lemma 3.2e and 3.2j, we have

$$(P^TM^{-1}P + Q)AZ = P^TM^{-1}PAZ + QAZ = \mathbf{0} + Z = Z,$$

and

$$M^{-1}PAZ = \mathbf{0}.$$

As a consequence, the columns of Z are the eigenvectors corresponding to the eigenvalues of BNN and DEF1 which are equal to 1 and 0, respectively.

Next, due to Theorem 2.8 of [24], it suffices to show that if $\sigma(P^TM^{-1}PA + QA) = \{1, \ldots, 1, \mu_{k+1}, \ldots, \mu_n\}$ holds then this implies $\sigma(M^{-1}PA) = \{0, \ldots, 0, \mu_{k+1}, \ldots, \mu_n\}$. The proof is as follows.

Consider the eigenvalues μ_i and corresponding eigenvectors v_i with $i = k + 1, \ldots, n$ of BNN, i.e.,

$$(P^TM^{-1}P + Q)Av_i = \mu_i v_i,$$

which implies

$$P^T(P^TM^{-1}P + Q)Av_i = \mu_i P^T v_i. \tag{3.4}$$

Applying Lemma 3.2a, 3.2b, 3.2d, 3.2i, we have

$$(P^T)^2M^{-1}PA + P^TQA = P^TM^{-1}P^2A + \mathbf{0} = P^TM^{-1}PAP^T.$$

Using the latter expression, Eq. (3.4) can be rewritten into

$$P^TM^{-1}PAw_i = \mu_i w_i.$$

where $w_i := P^Tv_i$. Note that due to Lemma 3.2d we have $P^Tx = 0$ if $x \in \mathrm{Col}(Z)$. However, $w_i \neq \mathbf{0}$ since $v_i \notin \mathrm{Col}(Z)$ for $i = k + 1, \ldots, n$. Hence, μ_i is also an eigenvalue of $P^TM^{-1}PA$. Lemma 3.3 gives

$$\sigma\left(M^{-1}PA\right) = \sigma\left(P^TM^{-1}PA\right),$$

so that μ_i is also an eigenvalue of DEF1. □

Hence, both operators of DEF1 and BNN lead to almost the same spectra with the same clustering. The zero eigenvalues of DEF1 are replaced by eigenvalues, which are one if BNN is used. Next, we give Corollary 3.1 which connects all methods in terms of spectra. It can be concluded that they all lead to almost the same spectra with the same clusters of eigenvalues.

Corollary 3.1. *Suppose the spectrum of DEF1, DEF2, R-BNN1 or R-BNN2 is given by $\{0, \ldots, 0, \lambda_{k+1}, \ldots, \lambda_n\}$. Moreover, let the spectrum of DEF1, DEF2, R-BNN1 or R-BNN2 be $\{1, \ldots, 1, \mu_{k+1}, \ldots, \mu_n\}$. Then, if λ_i and μ_i in the spectra are increasingly sorted, then $\lambda_i = \mu_i$ for all $i = k + 1, \ldots, n$.*

Remark 3.1. *The convergence of the methods depend on more aspects rather than only on the spectra of the preconditioned systems. It can also be influenced by aspects like starting vectors, rounding errors and manner of implementation. Although BNN, A-DEF1 and A-DEF2 are the same by considering their spectra, they all have their own properties and convergence rates. These also hold for DEF1, DEF2, R-BNN1 and R-BNN2.*

3.3 Identity of Abstract Balancing and Other Projection Methods

In this section, we will compare BNN and the methods DEF2, A-DEF2, R-BNN1 and R-BNN2. Since the BNN operator is SPD, it works appropriately for an SPSD matrix A. This is in contrast to DEF2, A-DEF2, R-BNN1 and R-BNN2, which obviously correspond to non-SPD operators. Sometimes the latter methods works appropriately, but this can not be proved. Fortunately, for special starting vectors, we can show that these methods are completely identical to BNN in exact arithmetic, see also Section 2.5.2 of [34]. As a consequence, these methods give appropriate operators as well, so that CG in combination with them will work fine.

It can be shown that Line 8 and Line 11 are not needed in Algorithm 7 if we extend the first line with

$$x_0 := Qb + P^T x_0. \tag{3.5}$$

In other words both $\hat{y}_{j+1} := Pr_{j+1} = r_{j+1}$ and $\tilde{z}_{j+1} := Qr_{j+1} = \mathbf{0}$ hold for all $j = 1, 2, \ldots$ and for special starting vector as given in Exp. (3.5), see also [34]. For completeness both results and corresponding proofs are given in Lemma 3.5 and 3.6.

Lemma 3.5. *Suppose that in Line 1 of Algorithm 7 has been extended with $x_0 := Qb + P^T x_0$. Then, Line 11 of this algorithm gives $Qr_{j+1} = \mathbf{0}$ for all $j = 1, 2, \ldots$ and $Qr_1 = \mathbf{0}$.*

Proof. The proof is given by induction.

- *Starting Case*: we prove that $Qr_1 = \mathbf{0}$ and $QAp_1 = \mathbf{0}$. First,

$$\begin{aligned} Qr_1 &= Q\left(b - A(Qb - P^T x_0)\right) \\ &= Qb - QAQb + QAP^T x_0 \\ &=^{\text{L3.2g,h}} Qb - Qb \\ &= \mathbf{0}. \end{aligned}$$

  Moreover,

$$\begin{aligned} QAp_1 &= QA(P^T M^{-1} P + Q) r_1 \\ &= QAP^T M^{-1} P r_1 + Qr_1 \\ &=^{\text{L3.2h}} \mathbf{0} + \mathbf{0} \\ &= \mathbf{0}. \end{aligned}$$

- *Inductive Hypothesis (IH)*: we assume that $Qr_j = \mathbf{0}$ and $QAp_j = \mathbf{0}$.

- *Inductive Step*: we show that $Qr_{j+1} = \mathbf{0}$ and $QAp_{j+1} = \mathbf{0}$:

$$\begin{aligned} Qr_{j+1} &= Q(r_j - \alpha_j A p_j) \\ &= Qr_j - \alpha_j QAp_j \\ &=^{\text{IH}} \mathbf{0} - \mathbf{0} \\ &= \mathbf{0}, \end{aligned}$$

  and also

$$\begin{aligned} QAp_{j+1} &= QA(z_{j+1} + \beta_j p_j) \\ &= QAz_{j+1} + \beta_j QAp_j \\ &= QA(P^T M^{-1} P + Q) r_{j+1}) \\ &= QAP^T M^{-1} P r_{j+1} + QAQr_{j+1} \\ &=^{\text{L3.2h}} \mathbf{0} + \mathbf{0} \\ &= \mathbf{0}. \end{aligned}$$

□

Lemma 3.6. *Suppose that Line 1 of Algorithm 7 has been extended with $x_0 := Qb + P^T x_0$. Then Line 8 of Algorithm 7 gives $Pr_{j+1} = r_{j+1}$ for all $j = 1, 2, \ldots$ and $Pr_1 = r_1$.*

Proof. We give this proof also by induction.

- *Starting Case*: we show that $Pr_1 = r_1$ and $PAp_1 = Ap_1$:

$$\begin{array}{rcl} Pr_1 & = & P\left(b - A(Qb - P^T x_0)\right) \\ & = & Pb - PAQb - PAP^T x_0 \\ & =^{\text{L3.2h}} & Pb - PAP^T x_0 \\ & =^{\text{L3.2b}} & Pb - A(P^T)^2 x_0 \\ & =^{\text{L3.2a}} & Pb - AP^T x_0 \\ & = & b - AQb - AP^T x_0 \\ & = & b - A(Qb + P^T x_0) \\ & = & b - Ax_0 \\ & = & r_1. \end{array}$$

  and furthermore,

$$\begin{array}{rcl} PAp_1 & = & PAP^T M^{-1} Pr_1 \\ & = & PAP^T M^{-1} r_1 \\ & =^{\text{L3.2b}} & A(P^T)^2 M^{-1} r_1 \\ & =^{\text{L3.2a}} & AP^T M^{-1} r_1 \\ & = & AP^T M^{-1} Pr_1 \\ & = & Ap_1, \end{array}$$

  where we used $p_1 = P^T M^{-1} Pr_1 + Qr_1 = P^T M^{-1} Pr_1$ in the last step, applying Lemma 3.5.

- *Inductive Hypothesis (IH)*: we assume that $Pr_j = r_j$ and $PAp_j = Ap_j$.

- *Inductive Step*: we prove that $Pr_{j+1} = r_{j+1}$ and $PAp_{j+1} = Ap_{j+1}$:

$$\begin{array}{rcl} Pr_{j+1} & = & Pr_j - \alpha_j PAp_j \\ & =^{\text{IH}} & r_j - \alpha_j Ap_j \\ & = & r_{j+1}, \end{array}$$

  and

$$\begin{array}{rcl} PAp_{j+1} & = & PA(z_{j+1} + \beta_j p_j) \\ & = & PAz_{j+1} + \beta_j PAp_j \\ & = & PAP^T M^{-1} Pr_{j+1} + \beta_j Ap_j \\ & = & PAP^T M^{-1} r_{j+1} + \beta_j Ap_j \\ & =^{\text{L3.2b}} & A(P^T)^2 M^{-1} r_{j+1} + \beta_j Ap_j \\ & =^{\text{L3.2a}} & AP^T M^{-1} r_{j+1} + \beta_j Ap_j \\ & = & AP^T M^{-1} Pr_{j+1} + \beta_j p_j \\ & = & A(z_{j+1} + \beta_j p_j) \\ & = & Ap_{j+1}, \end{array}$$

  where we applied $z_{j+1} = P^T M^{-1} Pr_{j+1} + Qr_{j+1} = P^T M^{-1} Pr_{j+1}$ which is the result of Lemma 3.5.

□

Remark 3.2. *Note that we need the result of Lemma 3.5 in order to obtain Lemma 3.6. In other words, if $Qr_j \neq \mathbf{0}$ then Lemma 3.6 would not be true. Moreover, note that from Lemma 3.6 we proved implicitly that the residuals are orthogonal to the deflation subspace matrix Z, i.e.,*

$$Z^T r_j = \mathbf{0}, \quad j = 1, 2, \ldots. \tag{3.6}$$

To be more precise, using induction we can show Expression (3.6) and, therefore, $Pr_j = r_j - AZE^{-1}(Z^T r_j) = r_j$ follows immediately.

Hence, by comparing the algorithms we conclude that if (3.5) has been adapted, then the BNN method is completely identical to R-BNN1, R-BNN2, A-DEF2 and also to DEF2, since the operator $P^T M^{-1}$ is the same in R-BNN2 and DEF2. This result is summarized in Corollary 3.2.

Corollary 3.2. *BNN with the special starting vector (i.e., BNN extended with Expression (3.5) is identical to DEF2, A-DEF2, R-BNN1 and R-BNN2.*

Remark 3.3. *Because of rounding errors, it may be possible that Lemma 3.5 and 3.6 are not fully satisfied in numerical experiments. Therefore, although BNN, DEF2,A-DEF2, R-BNN1 and R-BNN2 are identical in exact arithmetic, all methods except for BNN may lead to inaccurate solutions and even instabilities in these numerical experiments. In these cases, Line 8 and 11 of the BNN algorithm appear to be important and can not be omitted in these methods.*

CHAPTER 4 Numerical Experiments

After comparing the projection methods theoretically in the previous chapter, we will perform numerical experiments in this chapter to give also a numerical comparison of these methods.

First we describe the test problems and the settings of the numerical experiments. Subsequently, the results of these experiments will be presented by giving the number of iterations and the $2-$norms of the exact errors in table form and by giving the exact errors in the $A-$norm during the iteration process in figure form. These errors can be computed because we will also solve the resulting linear systems in a direct way Solutions obtained with a direct and iterative solver will be denoted by x and x_i, respectively. Moreover, figures with the residuals and exact errors in the $2-$norm and during the iteration processes are omitted in this paper, since they all show the same comparable behaviors in our test cases.

We will start with the numerical experiments using standard parameters, which means that we apply an appropriate termination criterion and an exact computation of E^{-1} and starting vector in these experiments. Subsequently, numerical experiments will be performed with inexact matrix E^{-1}, a severe termination criterion and a perturbed starting vector, respectively. Finally, we end this chapter with some experiments and notes on reorthogonalization strategies for non-converging methods.

4.1 Settings

The test problems and matrices M and Z will be described, which are used in the numerical experiments.

4.1.1 Test Problems

In the numerical experiments we consider three different 2-D test problems, which will be described below

Laplace Problem (TP1)

In Test Problem 1 (TP1), we consider the standard Laplace equation, i.e.,

$$\Delta p(\mathbf{x}) = \mathbf{0}, \quad \mathbf{x} = (x, y) \in \Omega = (0,1)^2, \tag{4.1}$$

with a Dirichlet condition on the boundary $y = 1$ and homogenous Neumann conditions on the other boundaries.

Porous Media Problem (TP2)

In Test Problem 2 (TP2), the Poisson equation with a discontinuous coefficient, i.e.,

$$-\nabla \cdot \left(\frac{1}{\rho(\mathbf{x})} \nabla p(\mathbf{x}) \right) = \mathbf{0}, \quad \mathbf{x} = (x, y) \in \Omega = (0,1)^2, \tag{4.2}$$

is considered with ρ and p denoting the permeability and fluid pressure, respectively. In addition, we have again a Dirichlet condition on the boundary $y = 1$ and homogenous Neumann conditions on the other boundaries. The form of the permeability ρ is given in Figure 4.1. Furthermore, we define the contrast ϵ as the jump between the highest and lowest permeability, i.e., $\epsilon := 10^{-6}/10^{0} = 10^{-6}$.

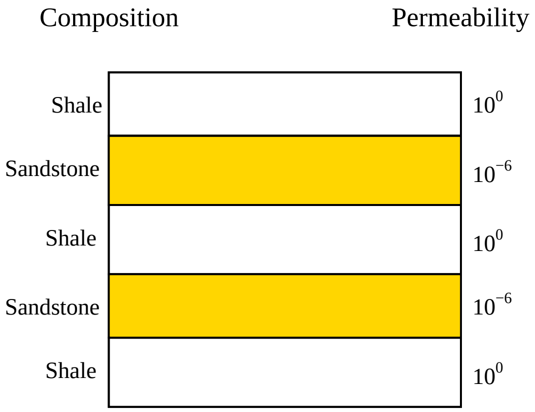


Figure 4.1: The permeability in the porous media problem (TP2).

Bubbly Flow Problem (TP3)

In Test Problem 3 (TP3), we investigate a bubbly flow problem. In this case, we have again the Poisson equation with discontinuous coefficient, i.e.,

$$-\nabla \cdot \left(\frac{1}{\rho(\mathbf{x})} \nabla p(\mathbf{x}) \right) = \mathbf{0}, \tag{4.3}$$

where we choose for non-homogeneous Neumann boundaries such that the resulting linear system is compatible. Finally, the form of the permeability ρ is given in Figure 4.2, where we again define the contrast as $\epsilon := 10^{-3}/10^{0} = 10^{-3}$.

4.1.2 Matrix A and Preconditioner M^{-1}

We use a standard second-order finite-difference scheme to discretize the test problems, which results in our main linear system

$$Ax = b, \quad A \in \mathbb{R}^{n \times n}. \tag{4.4}$$

The exact definition of A depends on the test problems. In the numerical experiments the gridsizes and permeabilities can be varied.

Moreover, in this paper we will restrict ourselves to the Incomplete Cholesky (IC) preconditioner M_{IC}, which is defined as $M_{\text{IC}} := LD^{-1}L^T$ with lower triangular matrix $L = [l_{ij}]$ and diagonal matrix $D = [d_{ij}]$ satisfying [21]:

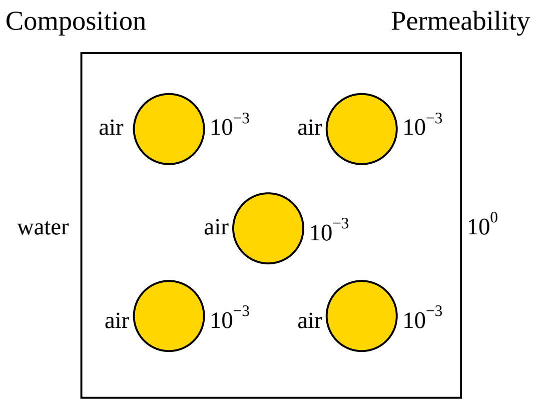


Figure 4.2: The permeability in the bubbly flow problem (TP3).

- $l_{ij} = 0$ if $a_{ij} = 0$;
- $(LD^{-1}L^T)_{ij} = a_{ij}$ if $a_{ij} \neq 0$ and $l_{ii} = d_{ii}$.

4.1.3 Deflation Subspace Matrix Z

Let the open domain Ω be divided into k open subdomains $\Omega_j,\ j = 1, 2, \ldots, k$, such that $\overline{\Omega} = \cup_{j=1}^{k} \overline{\Omega}_j$ and $\cap_{j=1}^{k} \Omega_j = \emptyset$ where $\overline{\Omega}_j$ is Ω_j including its adjacent boundaries. The discretized domain and subdomains are denoted by Ω_h and Ω_{h_j}, respectively. For each Ω_{h_j} with $j = 1, 2, \ldots, k$, we introduce a deflation vector z_j as follows:

$$(z_j)_i := \begin{cases} 0, & \mathbf{x}_i \in \Omega_h \setminus \overline{\Omega}_{h_j}; \\ 1, & \mathbf{x}_i \in \Omega_{h_j}, \end{cases}$$

where $\mathbf{x}_i$ is a grid point in the discretized domain Ω_h. Then the deflation subspace matrix Z is defined as

$$Z := [z_1\ z_2\ \cdots\ z_k] \in \mathbb{R}^{n \times k}.$$

Hence, Z consists of orthogonal disjunct piecewise-constant vectors.

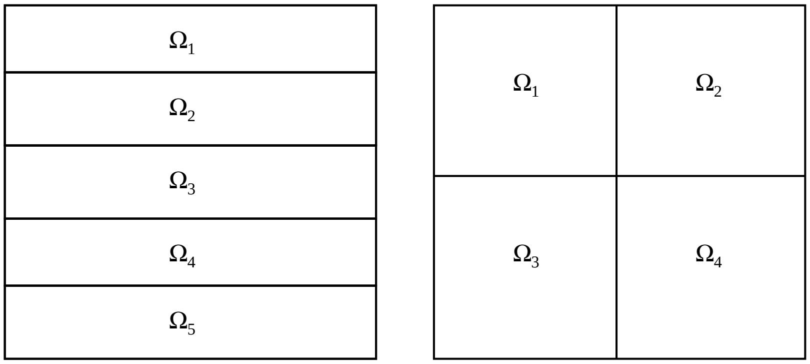
Remark 4.1. *In TP1 and TP2, matrix A is invertible. However, due to the Neumann boundary conditions, we have a singular matrix A in TP3, which satisfies the condition that the rowsums are zero. Due to the construction of the deflation vectors, E also appears to be singular. In this case, we can only use the pseudo-inverse rather than the real inverse of E. Here, we choose for another approach where we omit the last column of Z, i.e.,*

$$Z := [z_1\ z_2\ \cdots\ z_{k-1}] \in \mathbb{R}^{n \times k-1},$$

resulting in an invertible matrix E. We do not lose generality, since the row sums of A are zero and therefore the null space of PA with both choices of Z is the same, see also [32,33].

Finally, in this paper we restrict ourselves to two settings of domain decompositions of Ω_j: one with layers and the other with blocks. The associated geometries are depicted in Figure 4.3.

In TP1 we only use layers as deflation subdomains Ω_j. Moreover, in TP2, we also apply layers as deflation subdomains, where each layer of the problem corresponds to one subdomain. Hence, the number of deflation vectors is equal to the total number of shale and sandstone layers in the problem. Finally, instead of choosing layers as subdomains, we will apply blocks as deflation subdomains in



(a) Subdomains in layers used in TP1 and TP2.

(b) Subdomains in blocks used in TP3.

Figure 4.3: Geometry of subdomains Ω_j.

TP3. This means that each block can consist of regions with several permeabilities, making the situation in this case more sophisticated than in TP1 or TP2. Note that the deflation vectors are fixed in TP2, while they can be varied in TP1 and TP3.

In this case, we deal with a linear system, which is singular but this is not troublesome, see also the next section.

4.2 Numerical Results using Standard Parameters

Numerical experiments are performed using standard parameters with termination criterion $||z_{j+1}||/||z_1|| < \delta = 10^{-8}$, exact E^{-1} and exact starting vectors. The results considering TP1, TP2 and TP3 will be presented in the next subsections.

4.2.1 Test Problem 1: Laplace Problem

The results of the numerical experiments considering TP1 can be found in Table 4.1 and Figure 4.4. The number of deflation vectors k and the number of grid points n have been varied in these experiments.

Method	$n = 29^2, k = 5$		$n = 54^2, k = 5$		$n = 41^2, k = 7$		$n = 55^2, k = 7$	
	# It.	$\|\|x - x_i\|\|_2$	# It.	$\|\|x - x_i\|\|_2$	# It.	$\|\|x - x_i\|\|_2$	# It.	$\|\|x - x_i\|\|_2$
PREC	57	4.4×10^{-7}	74	6.5×10^{-7}	79	9.5×10^{-7}	100	4.7×10^{-6}
AD	47	9.1×10^{-7}	58	9.7×10^{-7}	65	1.9×10^{-6}	76	2.1×10^{-6}
DEF1	44	5.0×10^{-7}	55	1.3×10^{-6}	60	7.4×10^{-7}	74	2.1×10^{-6}
DEF2	44	5.0×10^{-6}	55	1.3×10^{-6}	60	7.4×10^{-7}	74	2.1×10^{-6}
A-DEF1	44	9.9×10^{-7}	57	3.1×10^{-6}	63	2.6×10^{-6}	77	2.3×10^{-6}
A-DEF2	45	2.1×10^{-7}	56	6.5×10^{-7}	60	7.4×10^{-7}	74	2.1×10^{-6}
BNN	44	5.2×10^{-7}	56	6.5×10^{-7}	60	7.4×10^{-7}	74	2.1×10^{-6}
R-BNN1	45	2.1×10^{-7}	56	6.5×10^{-7}	60	7.4×10^{-7}	74	2.1×10^{-6}
R-BNN2	45	2.1×10^{-7}	56	6.5×10^{-7}	60	7.4×10^{-7}	74	2.1×10^{-6}

Table 4.1: Number of required iterations for convergence and the 2−norm of the exact error of all proposed methods for TP1 with standard parameters.

Observation 4.1. *Considering both Table 4.1 and Figure 4.4, we observe that*

- *all methods, except for PREC, AD and A-DEF1, perform more or less the same;*
- *A-DEF1 is somewhat slower in convergence and less accurate in the solutions than the other methods except for IC and AD;*
- *as expected, PREC is obviously the slowest method, followed by AD. However, note that the differences between AD and the other projection methods are relatively small.*

4.2.2 Test Problem 2: Layer Problem

Similar to the previous subsection, the results of the numerical experiments with TP2 are presented in Table 4.2 and Figure 4.5.

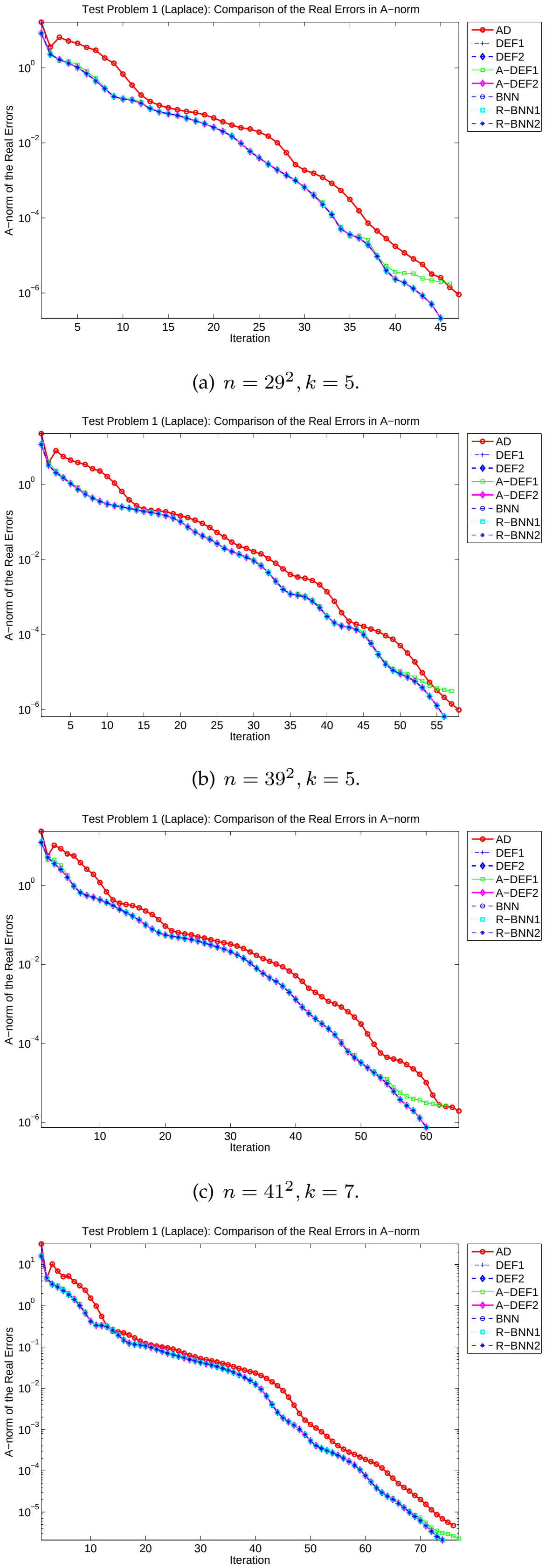
Method	$n = 29^2, k = 5$		$n = 54^2, k = 5$		$n = 41^2, k = 7$		$n = 55^2, k = 7$	
	# It.	$\|\|x - x_i\|\|_2$	# It.	$\|\|x - x_i\|\|_2$	# It.	$\|\|x - x_i\|\|_2$	# It.	$\|\|x - x_i\|\|_2$
PREC	101	2.2×10^{-7}	135	3.9×10^{-7}	188	2.3×10^{-7}	236	6.3×10^{-7}
AD	60	5.0×10^{-7}	74	1.7×10^{-6}	76	1.7×10^{-6}	92	3.2×10^{-7}
DEF1	53	2.2×10^{-7}	68	5.9×10^{-7}	68	3.0×10^{-7}	85	4.1×10^{-7}
DEF2	53	2.2×10^{-7}	68	5.9×10^{-7}	68	2.9×10^{-7}	85	4.3×10^{-7}
A-DEF1	60	4.6×10^{-7}	76	7.0×10^{-7}	69	2.5×10^{-6}	86	8.8×10^{-7}
A-DEF2	53	2.5×10^{-7}	69	2.6×10^{-7}	68	2.8×10^{-7}	85	4.5×10^{-7}
BNN	53	2.4×10^{-7}	69	2.6×10^{-7}	68	2.8×10^{-7}	85	4.3×10^{-7}
R-BNN1	53	2.2×10^{-7}	69	2.6×10^{-7}	68	2.8×10^{-7}	85	4.2×10^{-7}
R-BNN2	53	2.2×10^{-7}	69	2.8×10^{-7}	68	2.9×10^{-7}	85	4.3×10^{-7}

Table 4.2: Number of required iterations for convergence and the 2−norm of the exact error of all proposed methods for TP2 with $\epsilon = 10^{-6}$ and standard parameters.

Observation 4.2. *Considering both Table 4.2 and Figure 4.5, we can make exactly the same observations as done in the previous subsection:*

- *all methods, except for PREC, AD and A-DEF1, perform more or less the same;*
- *A-DEF1 is somewhat slower in convergence and less accurate in the solutions than the other methods except for IC and AD;*
- *PREC is obviously the slowest method, followed by AD. The differences between AD and the other projection methods are relatively small, although the plots show erratic behavior for AD.*

Finally, note that the methods, especially PREC, require more iterations to converge compared to TP1.



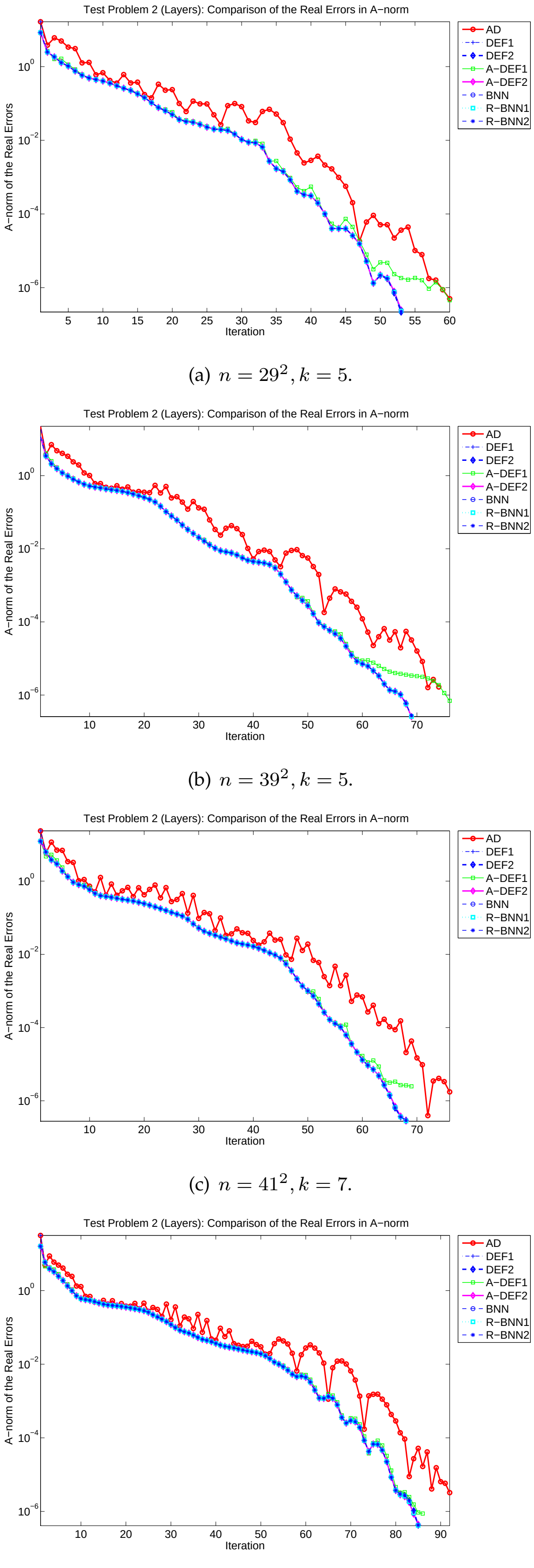
(a) $n = 29^2, k = 5$.

(b) $n = 39^2, k = 5$.

(c) $n = 41^2, k = 7$.

(d) $n = 55^2, k = 7$.

Figure 4.4: Exact errors in the $A-$norm for TP1 with standard parameters.



(a) $n = 29^2, k = 5$.

(b) $n = 39^2, k = 5$.

(c) $n = 41^2, k = 7$.

(d) $n = 55^2, k = 7$.

Figure 4.5: Exact errors in the $A-$norm for TP2 with standard parameters.

4.2.3 Test Problem 3: Bubbly Flow Problem

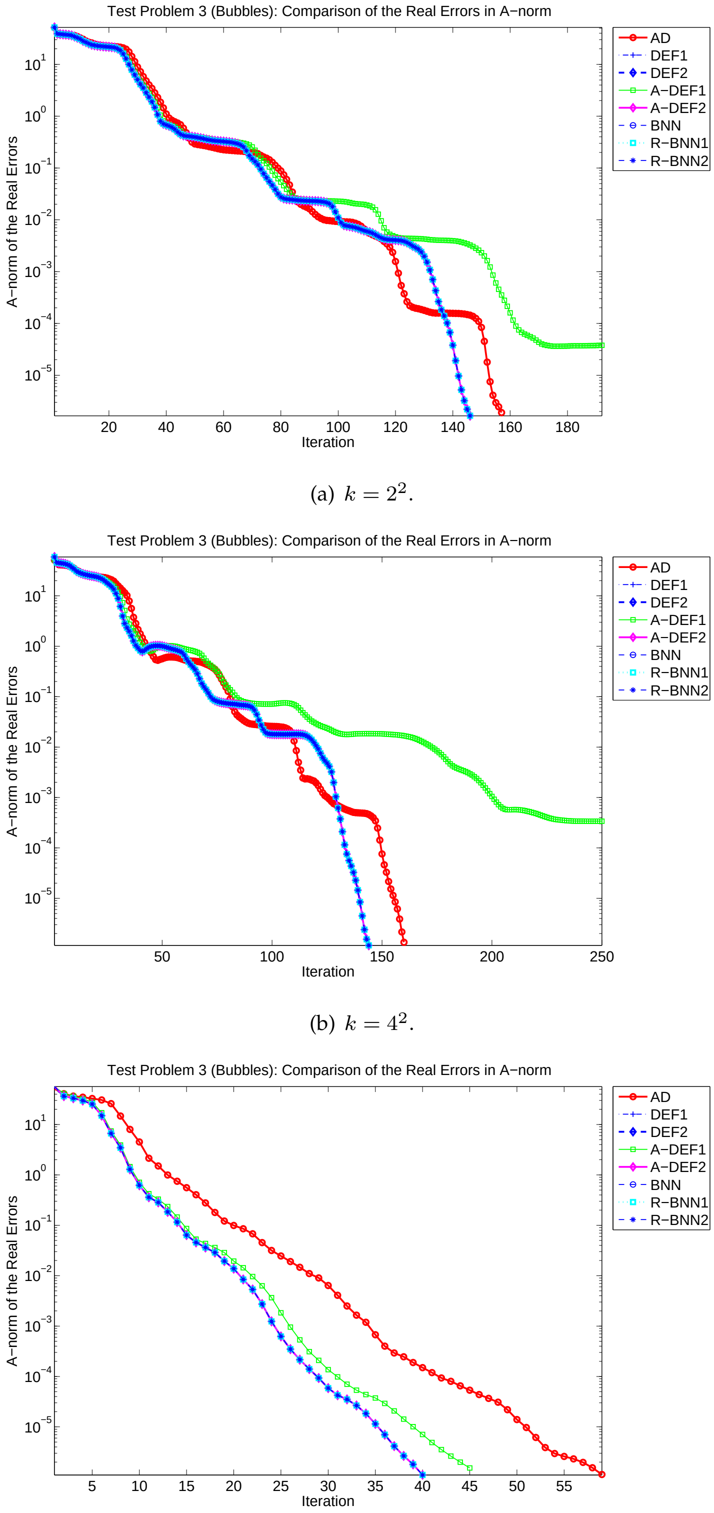
Similar to the previous subsections, the results of the numerical experiments with TP3 are presented in Table 4.3 and Figure 4.6. Now, we keep the number of grid points n constant and we vary the number of deflation vectors k and also the contrast ϵ of the density.

Method	$k = 2^2$		$k = 4^2$		$k = 8^2$	
	# It.	$\|\|x - x_i\|\|_2$	# It.	$\|\|x - x_i\|\|_2$	# It.	$\|\|x - x_i\|\|_2$
PREC	135	1.8×10^{-5}	135	1.8×10^{-5}	135	1.8×10^{-5}
AD	157	1.9×10^{-6}	160	1.3×10^{-6}	59	1.1×10^{-6}
DEF1	146	1.7×10^{-6}	144	1.2×10^{-6}	39	1.8×10^{-6}
DEF2	146	1.7×10^{-6}	144	1.2×10^{-6}	39	1.8×10^{-6}
A-DEF1	192	3.8×10^{-5}	NC	3.4×10^{-4}	45	1.5×10^{-6}
A-DEF2	146	1.7×10^{-6}	144	1.2×10^{-6}	40	1.1×10^{-6}
BNN	146	1.7×10^{-6}	144	1.2×10^{-6}	40	1.1×10^{-6}
R-BNN1	146	1.7×10^{-6}	144	1.2×10^{-6}	40	1.1×10^{-6}
R-BNN2	146	1.7×10^{-6}	144	1.2×10^{-6}	40	1.1×10^{-6}

Table 4.3: Number of required iterations for convergence and the $2-$norm of the exact error of all proposed methods for TP3 with $\epsilon = 10^{-3}, n = 64^2$ and standard parameters. 'NC' means no convergence within 250 iterations.

Observation 4.3. *Considering both Table 4.3 and Figure 4.6, we observe that*

- *all methods perform the same, except for PREC, AD and A-DEF1.*
- *A-DEF1 converges badly in some experiments, especially for the cases with $k = 2^2$ and $k = 4^2$;*
- *in the cases of $k = 2^2$ and $k = 4^2$, the number of deflation vectors is apparently too few to capture all eigenvectors corresponding to the small eigenvalues which is the result of the presence of the bubbles. Therefore, we hardly see improvements by comparing all projection methods to PREC. Note moreover that in these cases PREC requires less iterations than the other methods (which is unexpected and undesired), but the corresponding solution is somewhat less accurate than the others.*



(a) $k = 2^2$.

(b) $k = 4^2$.

(c) $k = 8^2$.

Figure 4.6: Exact errors in the $A-$norm for TP3 with $\epsilon = 10^{-3}, n = 64^2$ and standard parameters.

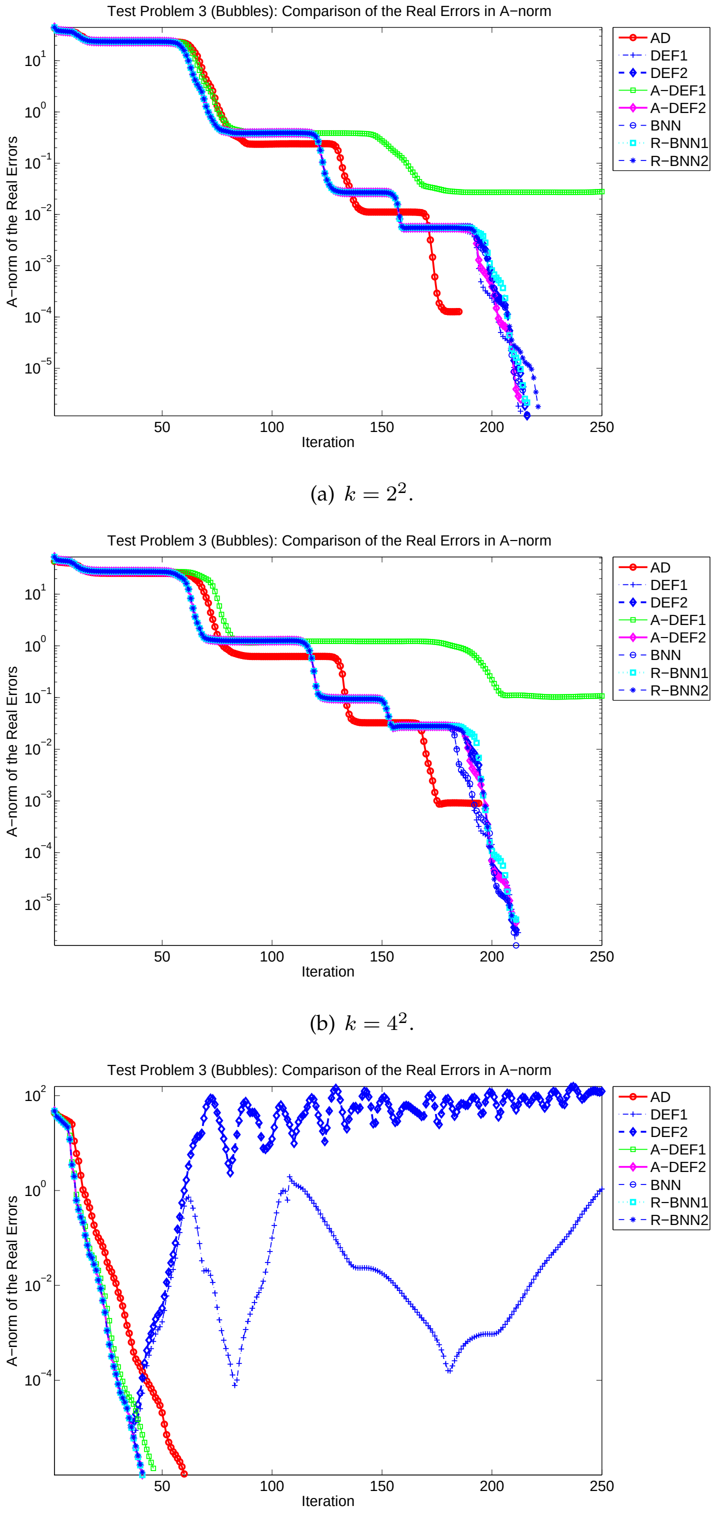
Subsequently, the results with $\epsilon = 10^{-6}$ can be found in Table 4.4 and Figure 4.7.

Method	$k = 2^2$		$k = 4^2$		$k = 8^2$	
	# It.	$\|\|x - x_i\|\|_2$	# It.	$\|\|x - x_i\|\|_2$	# It.	$\|\|x - x_i\|\|_2$
PREC	180	1.7×10^{-6}	180	1.7×10^{-6}	180	1.7×10^{-6}
AD	185	1.3×10^{-4}	194	9.0×10^{-4}	60	1.1×10^{-6}
DEF1	213	1.5×10^{-6}	212	2.8×10^{-6}	NC	1.1×10^{0}
DEF2	216	1.2×10^{-6}	211	3.2×10^{-6}	NC	$1.2 \times 10^{+2}$
A-DEF1	NC	2.8×10^{-2}	NC	1.1×10^{-1}	46	1.4×10^{-6}
A-DEF2	213	2.5×10^{-6}	211	4.5×10^{-6}	41	1.0×10^{-6}
BNN	216	1.2×10^{-6}	211	1.6×10^{-6}	41	1.0×10^{-6}
R-BNN1	216	2.2×10^{-6}	211	5.1×10^{-6}	41	1.0×10^{-6}
R-BNN2	221	1.8×10^{-6}	211	2.6×10^{-6}	41	1.2×10^{-6}

Table 4.4: Number of required iterations for convergence and the 2−norm of the exact error of all proposed methods for TP3 with $\epsilon = 10^{-6}, n = 64^2$ and standard parameters. 'NC' means no convergence within 250 iterations.

Observation 4.4. *Considering both Table 4.4 and Figure 4.7, we notice that*

- *the methods A-DEF2, BNN, R-BNN1 and R-BNN2 perform more or less the same.*
- *as in the previous subsection with $\epsilon = 10^{-3}$, A-DEF1 has some problems for the cases with $k = 2^2$ and $k = 4^2$ where the method hardly converges;*
- *the projection methods perform again worse compared to PREC for $k = 2^2$ and $k = 4^2$;*
- *both DEF1 and DEF2 diverge in the case of $k = 8^2$. In the figure we see that they do not reach a high accuracy in contrast to the other methods. Apparently, DEF1 and DEF2 are sensitive for the value of the contrast ϵ. It seems that these methods can not reach the required accuracy for this ill-conditioned problem.*



(a) $k = 2^2$.

(b) $k = 4^2$.

(c) $k = 8^2$.

Figure 4.7: Exact errors in the $A-$norm for TP3 with $\epsilon = 10^{-6}, n = 64^2$ and standard parameters.

4.3 Numerical Results using Inaccurate Coarse Solves

In practice it can be difficult to find an accurate solution y of the coarse system $Ey = z$ at each iterate of a projection method. Instead, only a approximated solution $\tilde{y}$ may be available if one applies for instance an iterative solver for $Ey = z$ with a relatively large stopping tolerance. In this case, $\tilde{y}$ can be interpreted as $\tilde{y} = \tilde{E}^{-1}x$ where $\tilde{E}$ is an inexact matrix based on E. This motivates our next experiment where we use $\tilde{E}^{-1}$, which is defined as follows:

$$\tilde{E}^{-1} := (I + \psi R)E^{-1}(I + \psi R), \quad \psi > 0, \tag{4.5}$$

where $R \in \mathbb{R}^{n \times n}$ is a symmetric random matrix with elements from the interval $[-0.5, 0.5]$. Note that this perturbed matrix $\tilde{E}^{-1}$ is not general, since it is constant at each iterate. The sensitivity of the methods to this inaccurate coarse matrix with various values of ψ will be investigated in this section. Note that the results for PREC are not influenced by this adaption of E^{-1}. In this section they are are only included for reference.

The numerical experiments and results considering all test problems will be presented in the same manner as in the previous section.

4.3.1 Test Problem 1: Laplace Problem

The results of TP1 with $n = 29^2$ and $k = 5$ can be found in Table 4.5 and Figures 4.8 and 4.9.

Method	$\psi = 10^{-8}$		$\psi = 10^{-6}$		$\psi = 10^{-4}$	
	# It.	$\|\|x - x_i\|\|_2$	# It.	$\|\|x - x_i\|\|_2$	# It.	$\|\|x - x_i\|\|_2$
PREC	57	4.4×10^{-7}	57	4.4×10^{-7}	57	4.4×10^{-7}
AD	47	9.1×10^{-7}	47	9.1×10^{-7}	47	9.0×10^{-7}
DEF1	44	1.5×10^{-6}	177	9.2×10^{-7}	135	1.1×10^{-7}
DEF2	44	1.5×10^{-6}	NC	$1.9 \times 10^{+3}$	NC	$2.1 \times 10^{+3}$
A-DEF1	44	9.9×10^{-7}	44	9.9×10^{-7}	44	1.0×10^{-6}
A-DEF2	45	2.1×10^{-7}	45	2.1×10^{-7}	45	2.1×10^{-7}
BNN	45	2.1×10^{-7}	45	2.1×10^{-7}	45	2.1×10^{-7}
R-BNN1	45	4.3×10^{-7}	45	2.8×10^{-5}	71	8.3×10^{-4}
R-BNN2	45	1.2×10^{-6}	NC	1.0×10^{-4}	NC	1.1×10^{-2}

Method	$\psi = 10^{-2}$		$\psi = 10^{0}$	
	# It.	$\|\|x - x_i\|\|_2$	# It.	$\|\|x - x_i\|\|_2$
PREC	57	4.4×10^{-7}	57	4.4×10^{-7}
AD	48	7.4×10^{-7}	74	2.0×10^{-6}
DEF1	90	4.7×10^{-7}	74	4.9×10^{-7}
DEF2	NC	$1.1 \times 10^{+3}$	NC	$1.1 \times 10^{+2}$
A-DEF1	44	1.8×10^{-6}	NC	8.0×10^{0}
A-DEF2	45	2.5×10^{-7}	NC	$1.3 \times 10^{+1}$
BNN	45	2.1×10^{-7}	52	2.8×10^{-5}
R-BNN1	130	4.3×10^{-7}	75	3.5×10^{-5}
R-BNN2	NC	8.3×10^{-1}	NC	$1.1 \times 10^{+1}$

Table 4.5: Number of required iterations for convergence and the 2−norm of the exact error of all proposed methods for TP1 with parameters $n = 29^2, k = 5$ and inaccurate E^{-1}.

Observation 4.5. *Considering Table 4.5 and Figure 4.8 and 4.9, we observe that*

- *for $\psi \leq 10^{-8}$, all methods converge appropriately;*

- *BNN is the most stable method in all test cases, although it also shows some problems in the case of $\psi = 10^{0}$. Therefore, in fact $\psi = 10^{-2}$ is the smallest perturbation which can be chosen for a fair comparison between the methods;*

- *for $\psi \leq 10^{-2}$, AD, BNN, A-DEF1 and A-DEF2 show good results and, hence, they are the most stable methods;*

- *DEF2 and R-BNN2 do not converge at all for all perturbations $\psi \geq 10^{-6}$, while R-BNN1 converges only properly for sufficiently small ψ.*

- *DEF1 converges to the solution, but with a slow speed. Therefore it is more robust than for instance DEF2. Notice that the speed of convergence of DEF1 is faster for larger ψ. The explanation is that the smallest eigenvalues are nearly zero for small ψ, while for larger ψ the smallest eigenvalues also become larger resulting in faster convergence.*

4.3.2 Test Problem 2: Poisson Problem

Similar to the previous subsection, the results considering TP2 can be found in Table 4.6 and Figure 4.10. Note that we use a different range of ψ compared to TP1, since the condition of matrix E is different in these test problems.

Method	$\psi = 10^{-16}$		$\psi = 10^{-12}$		$\psi = 10^{-8}$		$\psi = 10^{-4}$	
	# It.	$\|\|x - x_i\|\|_2$	# It.	$\|\|x - x_i\|\|_2$	# It.	$\|\|x - x_i\|\|_2$	# It.	$\|\|x - x_i\|\|_2$
PREC	101	2.2×10^{-7}	101	2.2×10^{-7}	101	2.2×10^{-7}	101	2.2×10^{-7}
AD	60	4.9×10^{-7}	60	5.0×10^{-2}	60	4.5×10^{-7}	53	1.9×10^{-5}
DEF1	53	2.2×10^{-7}	NC	2.3×10^{-2}	234	2.1×10^{-6}	91	2.1×10^{-1}
DEF2	53	2.2×10^{-7}	NC	$2.8 \times 10^{+4}$	NC	$3.7 \times 10^{+4}$	NC	$3.5 \times 10^{+5}$
A-DEF1	60	7.3×10^{-7}	60	7.0×10^{-7}	60	7.9×10^{-7}	77	1.4×10^{-5}
A-DEF2	53	2.6×10^{-7}	53	2.4×10^{-7}	53	2.5×10^{-7}	57	5.4×10^{-6}
BNN	53	2.5×10^{-7}	53	2.3×10^{-7}	53	2.4×10^{-7}	49	1.6×10^{-5}
R-BNN1	53	2.4×10^{-7}	53	1.9×10^{-5}	53	1.9×10^{-1}	78	1.1×10^{-3}
R-BNN2	53	2.2×10^{-7}	53	5.7×10^{-5}	54	5.7×10^{-1}	NC	$4.1 \times 10^{+3}$

Table 4.6: Number of required iterations for convergence and the 2−norm of the exact error of all proposed methods for TP2 with parameters $n = 29^2, k = 5, \epsilon = 10^{-6}$ and inaccurate E^{-1}.

Observation 4.6. *Considering both Table 4.6 and Figure 4.10, the following observations can be made:*

- *the most stable methods are AD, BNN, A-DEF2. Note that for $\psi \geq 10^{-8}$, A-DEF1, R-BNN1 and R-BNN2 converge, but not always to the correct solution.*

- *DEF1 and DEF2 are obviously the worst methods, although DEF1 becomes better for larger ψ.*

4.3.3 Test Problem 3: Bubbly Flow Problem

The results for TP3 can be found in Table 4.7 and Figure 4.11.

Observation 4.7. *Based on Table 4.7 and Figure 4.11, we observe that*

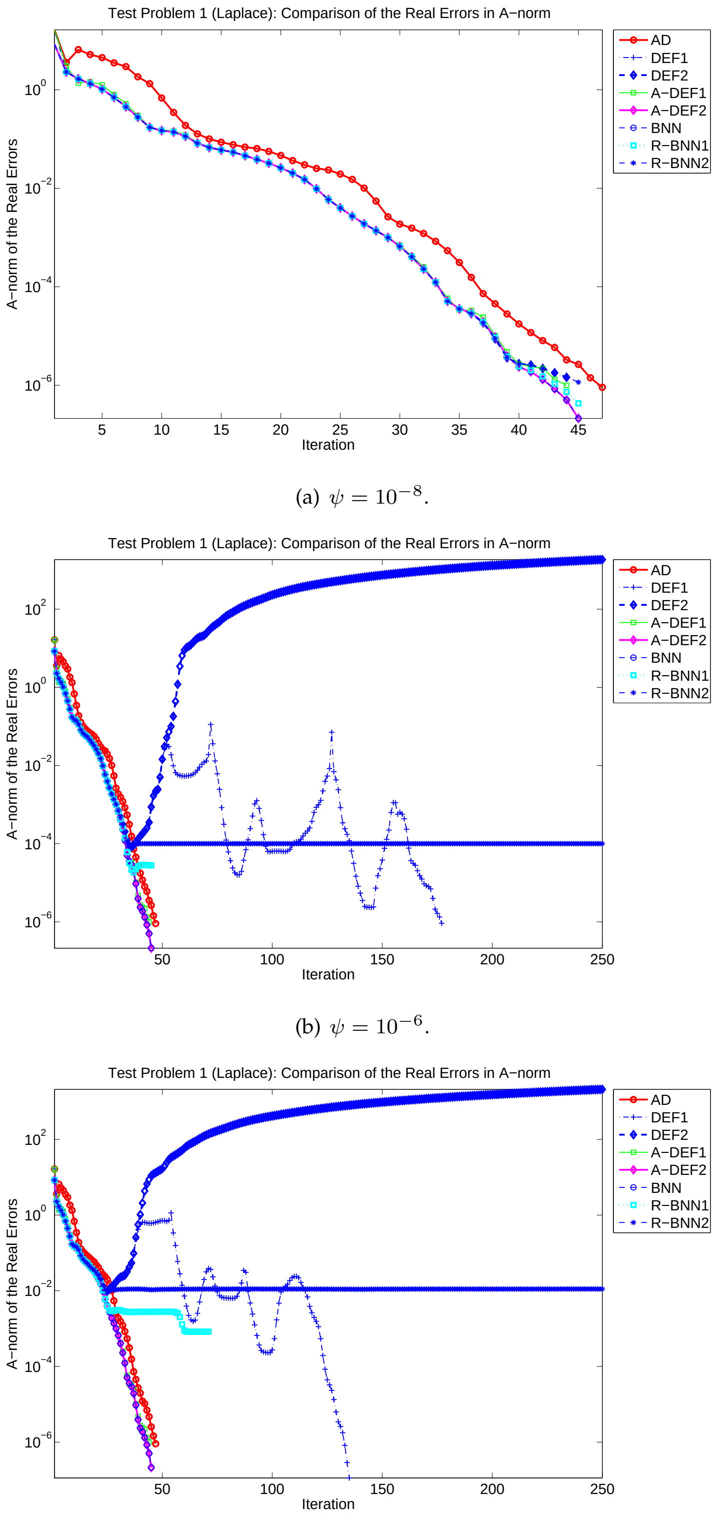
- *all methods are stable as long as $\psi \leq 10^{-12}$;*

Method	$\psi = 10^{-12}$		$\psi = 10^{-8}$		$\psi = 10^{-4}$		$\psi = 10^{-2}$	
	# It.	$\|\|x - x_i\|\|_2$	# It.	$\|\|x - x_i\|\|_2$	# It.	$\|\|x - x_i\|\|_2$	# It.	$\|\|x - x_i\|\|_2$
PREC	135	1.8×10^{-5}	135	1.8×10^{-5}	135	1.8×10^{-5}	135	1.8×10^{-5}
AD	59	1.1×10^{-6}	59	1.1×10^{-6}	59	1.3×10^{-6}	66	3.8×10^{-5}
DEF1	39	1.8×10^{-6}	NC	2.0×10^{-2}	NC	4.6×10^{0}	NC	$1.6 \times 10^{+1}$
DEF2	39	1.8×10^{-7}	NC	$1.5 \times 10^{+4}$	NC	$8.4 \times 10^{+3}$	NC	$1.0 \times 10^{+3}$
A-DEF1	46	8.1×10^{-6}	46	8.1×10^{-7}	43	2.1×10^{-6}	NC	$2.0 \times 10^{+1}$
A-DEF2	40	1.1×10^{-6}	40	1.1×10^{-6}	40	1.2×10^{-6}	NC	$2.3 \times 10^{+1}$
BNN	40	1.1×10^{-6}	40	1.1×10^{-6}	40	1.1×10^{-6}	43	6.6×10^{-4}
R-BNN1	40	1.1×10^{-6}	40	1.5×10^{-4}	NC	6.7×10^{-3}	NC	3.4×10^{-1}
R-BNN2	40	1.1×10^{-6}	NC	1.4×10^{-3}	NC	6.6×10^{0}	NC	$3.6 \times 10^{+1}$

Table 4.7: Number of required iterations for convergence and the $2-$norm of the exact error of all proposed methods for TP2 with parameters $n = 64^2, k = 8^2, \epsilon = 10^{-3}$ and inaccurate E^{-1}.

- *the most stable methods appear to be AD and BNN in the test cases, although A-DEF1 and A-DEF2 perform appropriately for $\psi \leq 10^{-4}$;*

- *obviously, DEF1, DEF2 and also R-BNN1 and R-BNN2 converge badly for most $\psi \geq 10^{-8}$. Note that for $\psi = 10^{-8}$, R-BNN1 converges but not to an accurate solution.*

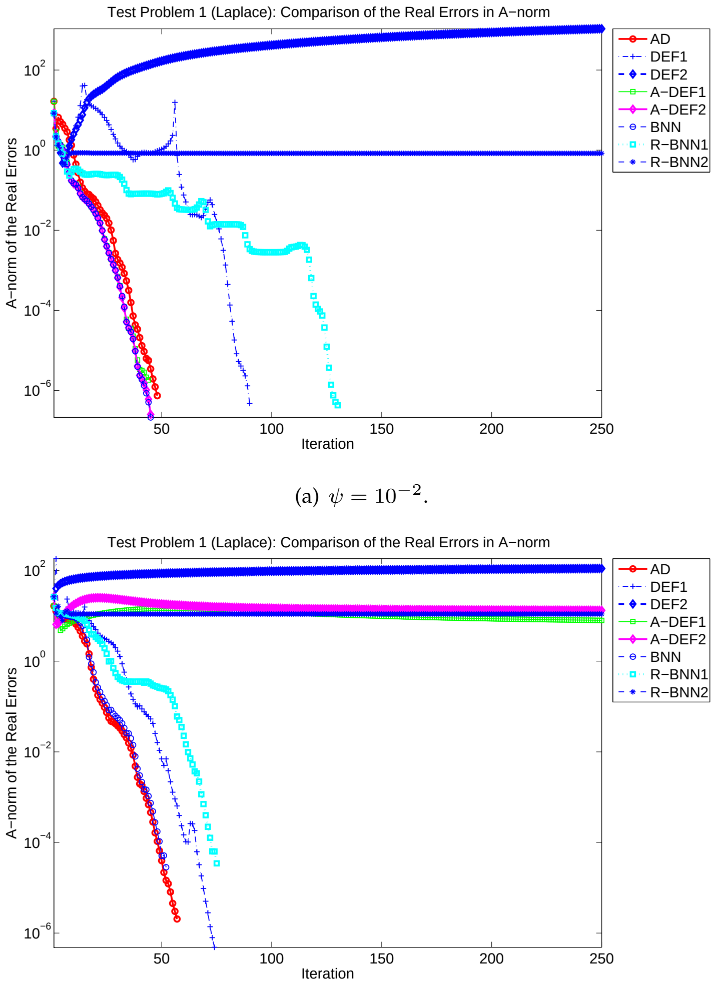


(a) $\psi = 10^{-8}$.

(b) $\psi = 10^{-6}$.

(c) $\psi = 10^{-4}$.

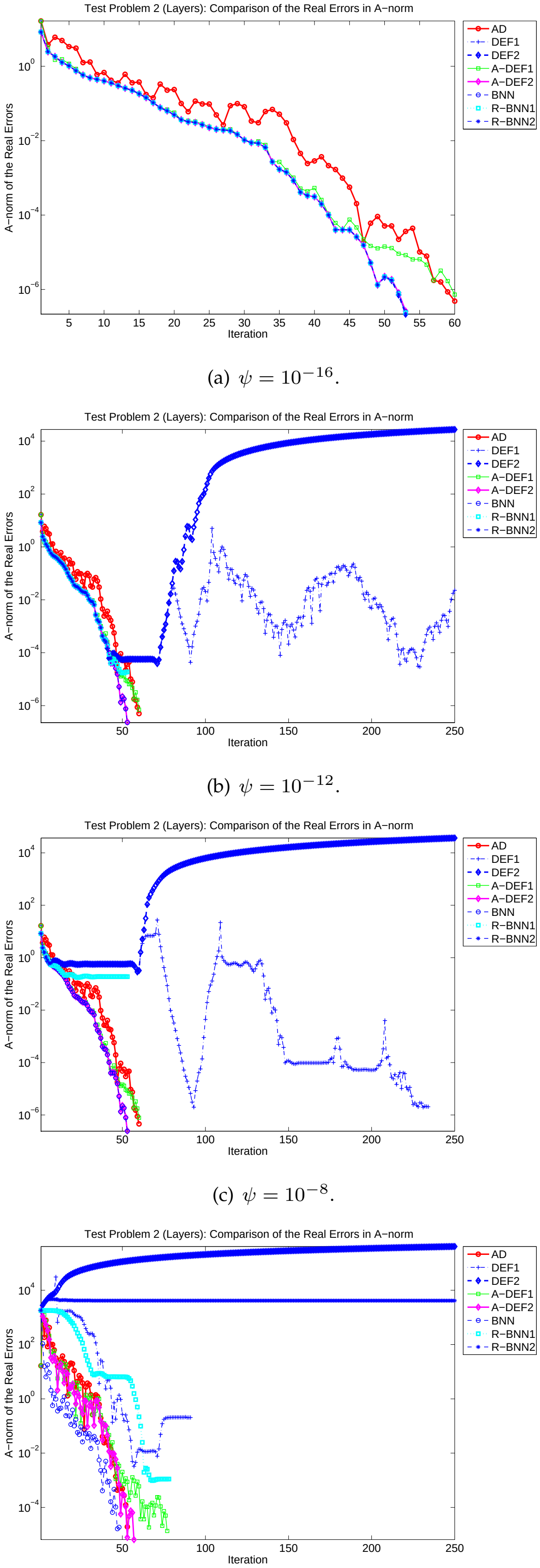
Figure 4.8: Exact errors in the $A-$norm for TP1 with $n = 29^2, k = 5, \psi = 10^{-4}, 10^{-6}, 10^{-8}$ and inaccurate E^{-1}.



(a) $\psi = 10^{-2}$.

(b) $\psi = 10^{0}$.

Figure 4.9: Exact errors in the $A-$norm for TP1 with $n = 29^2, k = 5, \psi = 10^{-2}, 10^{0}$ and inaccurate E^{-1}.



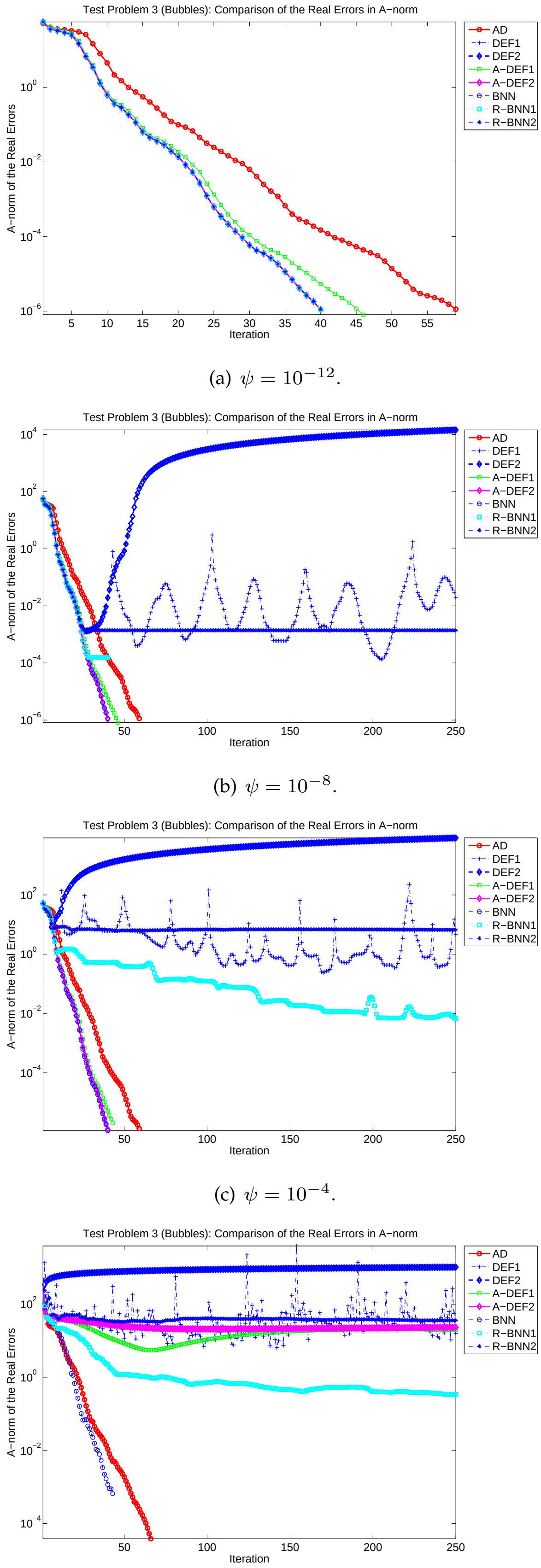
(a) $\psi = 10^{-16}$.

(b) $\psi = 10^{-12}$.

(c) $\psi = 10^{-8}$.

(d) $\psi = 10^{-4}$.

Figure 4.10: Exact errors in the A−norm for TP2 with $n = 29^2, k = 5, \epsilon = 10^{-6}$ and inaccurate E^{-1}.



(a) $\psi = 10^{-12}$.

(b) $\psi = 10^{-8}$.

(c) $\psi = 10^{-4}$.

(d) $\psi = 10^{-2}$.

Figure 4.11: Exact errors in the $A-$norm for TP3 with $n = 64^2, k = 8^2, \epsilon = 10^{-3}$ and inaccurate E^{-1}.

4.4 Numerical Results using Severe Termination Criteria

In the previous numerical experiments we applied the following termination criterion for the iterative process:

$$\frac{||z_{j+1}||_2}{||z_1||_2} < \delta, \quad \delta = 10^{-8}. \tag{4.6}$$

In this section we perform numerical experiments using various values of δ. In other words, we investigate the behavior of the methods by using different tolerances. Note that sometimes δ will be chosen (too) small leading to non-realistic experiments. In these cases, the results have no physical meaning anymore since the machine precision (which depends on the condition number of A) is reached. However, the goal of this section is to test the sensitivity of and to give insights into the methods to δ, rather than to perform realistic experiments.

4.4.1 Test Problem 1: Laplace Problem

The results of TP1 with different values of δ can be found in Table 4.8 and Figure 4.12.

Method	$\delta = 10^{-8}$		$\delta = 10^{-12}$		$\delta = 10^{-16}$		$\delta = 10^{-20}$	
	# It.	$\|\|x - x_i\|\|_2$	# It.	$\|\|x - x_i\|\|_2$	# It.	$\|\|x - x_i\|\|_2$	# It.	$\|\|x - x_i\|\|_2$
PREC	57	4.4×10^{-7}	74	1.4×10^{-11}	85	8.1×10^{-13}	106	8.1×10^{-13}
AD	47	9.1×10^{-7}	67	2.4×10^{-11}	80	8.3×10^{-13}	92	8.3×10^{-13}
DEF1	44	5.0×10^{-7}	59	6.6×10^{-11}	NC	8.6×10^{-6}	NC	8.6×10^{-6}
DEF2	44	5.0×10^{-6}	59	6.6×10^{-11}	NC	$2.2 \times 10^{+3}$	NC	$2.2 \times 10^{+3}$
A-DEF1	44	9.9×10^{-7}	73	7.4×10^{-11}	110	7.9×10^{-13}	167	7.9×10^{-13}
A-DEF2	45	2.1×10^{-7}	60	4.3×10^{-11}	73	8.4×10^{-13}	84	8.4×10^{-13}
BNN	44	5.2×10^{-7}	60	4.3×10^{-11}	73	7.8×10^{-13}	84	7.8×10^{-13}
R-BNN1	45	2.1×10^{-7}	60	4.3×10^{-11}	73	8.0×10^{-13}	84	8.0×10^{-13}
R-BNN2	45	2.1×10^{-7}	60	4.3×10^{-11}	73	7.0×10^{-13}	NC	7.0×10^{-13}

Table 4.8: Number of required iterations for convergence and the $2-$norm of the exact error of all proposed methods for TP1 with parameters $n = 29^2, k = 5$ and various termination criterion.

Observation 4.8. *Based on Table 4.8 and Figure 4.12, we see that*

- *all methods perform well as long as $\delta \geq 10^{-12}$;*
- *DEF1 and DEF2 show problems where DEF2 even diverges for $\delta \leq 10^{-16}$. Moreover, for $\delta \leq 10^{-20}$, A-DEF1 and R-BNN2 also have difficulties;*
- *AD, A-DEF2, BNN and R-BNN1 give good convergence results for all cases and, therefore, they are extremely stable in this experiment.*

4.4.2 Test Problem 2: Layer Problem

The results with TP2 are given in Table 4.9 and Figure 4.13.

Observation 4.9. *Considering both Table 4.8 and Figure 4.12, identical observations as in TP1 can be seen:*

- *all methods perform well for $\delta \geq 10^{-12}$;*

Method	$\delta = 10^{-8}$		$\delta = 10^{-12}$		$\delta = 10^{-16}$		$\delta = 10^{-20}$	
	# It.	$\|\|x - x_i\|\|_2$	# It.	$\|\|x - x_i\|\|_2$	# It.	$\|\|x - x_i\|\|_2$	# It.	$\|\|x - x_i\|\|_2$
PREC	101	2.2×10^{-7}	131	3.8×10^{-8}	201	3.6×10^{-8}	220	3.6×10^{-8}
AD	60	5.0×10^{-7}	76	2.0×10^{-8}	87	2.0×10^{-8}	106	2.0×10^{-8}
DEF1	53	2.2×10^{-7}	69	6.3×10^{-8}	NC	1.4×10^{-6}	NC	1.4×10^{-6}
DEF2	53	2.2×10^{-7}	69	7.7×10^{-8}	NC	$2.3 \times 10^{+5}$	NC	$2.3 \times 10^{+5}$
A-DEF1	60	4.6×10^{-7}	97	2.5×10^{-8}	136	2.5×10^{-8}	167	2.5×10^{-8}
A-DEF2	53	2.5×10^{-7}	69	7.0×10^{-9}	81	7.0×10^{-9}	97	7.0×10^{-9}
BNN	53	2.4×10^{-7}	69	6.0×10^{-9}	81	6.1×10^{-9}	97	6.1×10^{-9}
R-BNN1	53	2.2×10^{-7}	69	2.9×10^{-8}	81	2.9×10^{-8}	103	2.9×10^{-8}
R-BNN2	53	2.2×10^{-7}	69	7.7×10^{-8}	NC	7.7×10^{-8}	NC	7.7×10^{-8}

Table 4.9: Number of required iterations for convergence and the 2−norm of the exact error of all proposed methods for TP2 with parameters $n = 29^2, k = 5, \epsilon = 10^{-6}$ and various termination criterion.

- *for $\delta \leq 10^{-16}$, DEF1 and DEF2 are troublesome and additionally A-DEF1 and R-BNN2 show difficulties as well for $\delta \leq 10^{-20}$.*
- *finally, PREC, AD, A-DEF2, BNN and R-BNN1 give good convergence results for all cases.*

4.4.3 Test Problem 3: Bubbly Flow Problem

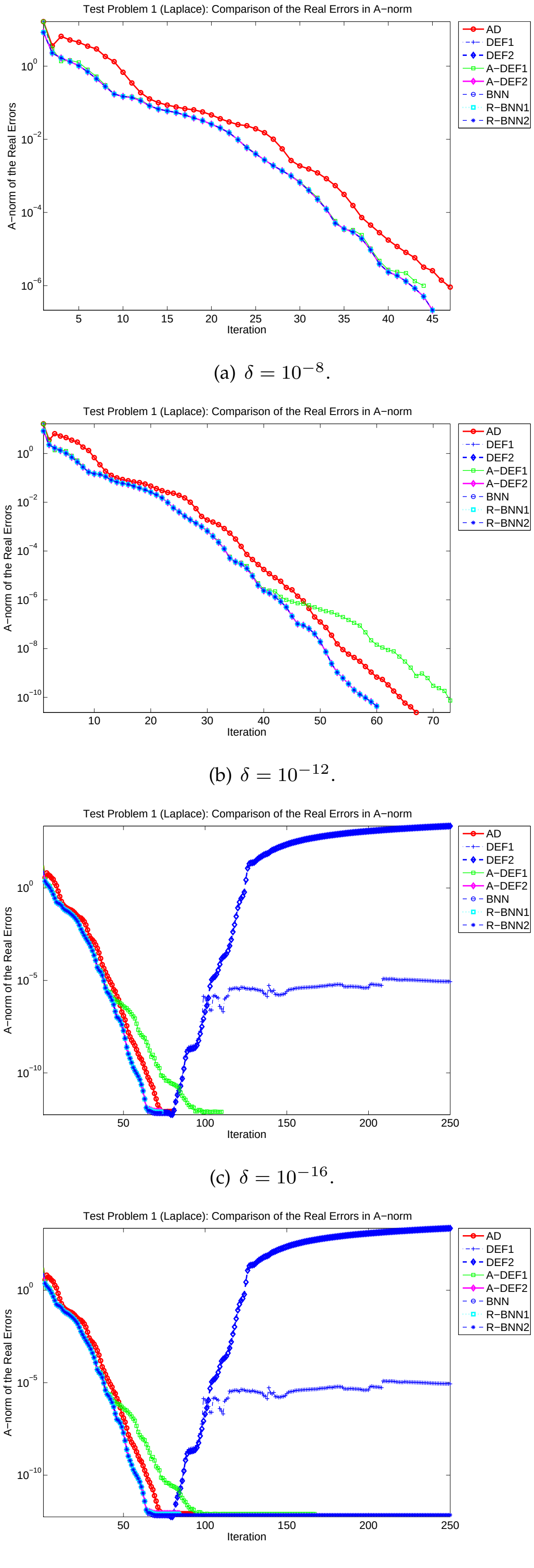
The results for TP3 are presented in Table 4.10 and Figure 4.14.

Method	$\delta = 10^{-8}$		$\delta = 10^{-10}$		$\delta = 10^{-12}$		$\delta = 10^{-14}$	
	# It.	$\|\|x - x_i\|\|_2$	# It.	$\|\|x - x_i\|\|_2$	# It.	$\|\|x - x_i\|\|_2$	# It.	$\|\|x - x_i\|\|_2$
PREC	135	1.8×10^{-5}	162	6.7×10^{-9}	170	9.2×10^{-11}	177	6.7×10^{-11}
AD	59	1.1×10^{-6}	71	1.9×10^{-8}	86	3.1×10^{-10}	NC	6.1×10^{-10}
DEF1	39	1.8×10^{-6}	51	2.0×10^{-8}	NC	6.5×10^{-9}	NC	6.5×10^{-9}
DEF2	39	1.8×10^{-6}	51	2.0×10^{-8}	NC	9.8×10^{-7}	NC	9.8×10^{-7}
A-DEF1	45	1.5×10^{-6}	58	1.5×10^{-8}	71	9.2×10^{-11}	87	8.7×10^{-11}
A-DEF2	40	1.1×10^{-6}	52	1.1×10^{-8}	61	1.3×10^{-10}	194	8.0×10^{-11}
BNN	40	1.1×10^{-6}	52	1.1×10^{-8}	61	1.4×10^{-10}	193	1.1×10^{-10}
R-BNN1	40	1.1×10^{-6}	52	1.1×10^{-8}	61	1.2×10^{-10}	198	8.8×10^{-11}
R-BNN2	40	1.1×10^{-6}	52	1.1×10^{-8}	71	2.6×10^{-10}	NC	1.2×10^{-10}

Table 4.10: Number of required iterations for convergence and the 2−norm of the exact error of all proposed methods for TP2 with parameters $n = 64^2, k = 8^2, \epsilon = 10^{-3}$ and various termination criterion.

Observation 4.10. *Considering both Table 4.10 and Figure 4.14, we observe that*

- *all methods perform the same if $\delta \geq 10^{-10}$,;*
- *for $\delta \leq 10^{-12}$, DEF1 and DEF2 become unstable due to the fact that after a certain moment in the iteration process, the zero-eigenvalues (which are in fact nearly zero-eigenvalues) play a role and cause the bad convergence;*
- *for $\delta \leq 10^{-14}$, A-DEF1, A-DEF2, BNN and R-BNN1 are the only methods which still converge.*


Test Problem 1 (Laplace): Comparison of the Real Errors in A−norm

(a) $\delta = 10^{-8}$.

Test Problem 1 (Laplace): Comparison of the Real Errors in A−norm

(b) $\delta = 10^{-12}$.

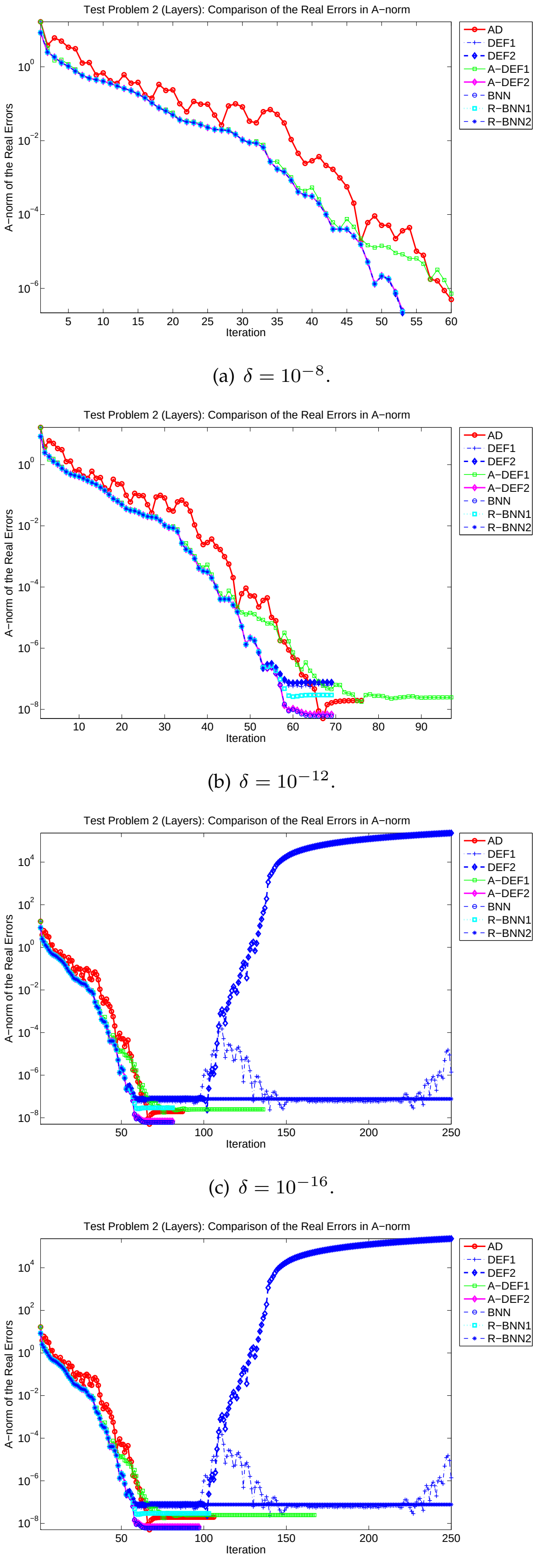
Test Problem 1 (Laplace): Comparison of the Real Errors in A−norm

(c) $\delta = 10^{-16}$.

Test Problem 1 (Laplace): Comparison of the Real Errors in A−norm

(d) $\delta = 10^{-20}$.

Figure 4.12: Exact errors in the $A-$norm for TP1 with $n = 29^2, k = 5^2$ and various termination criterion.



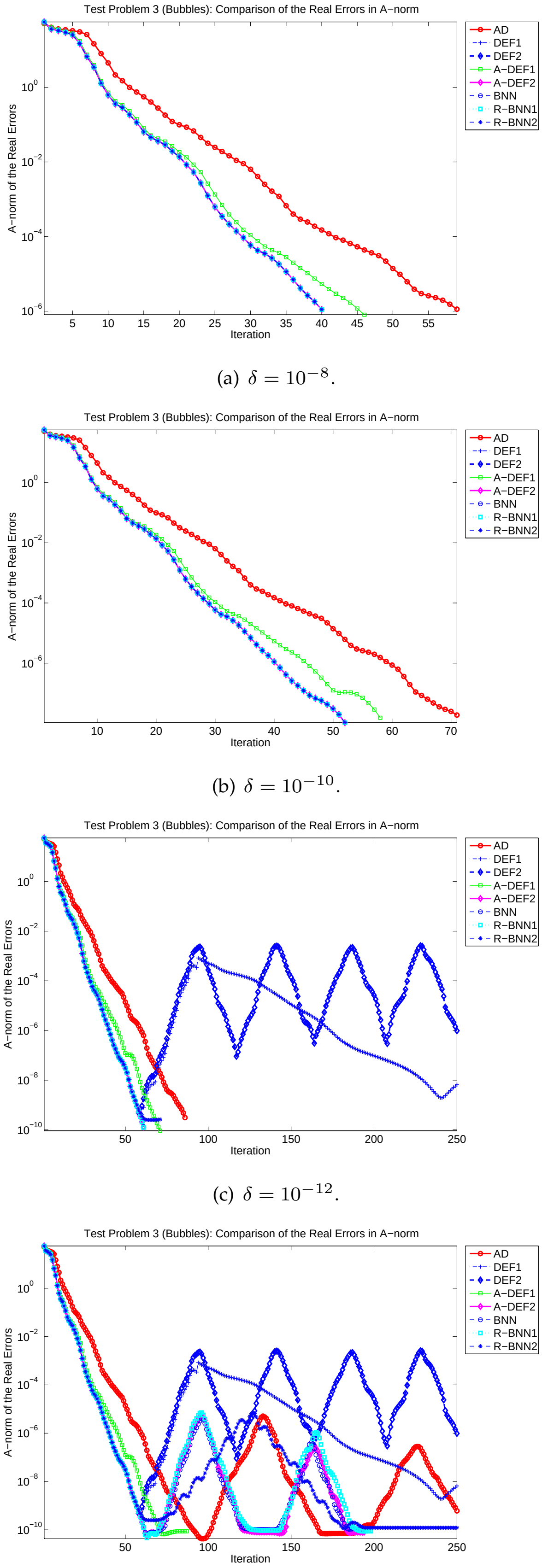
(a) $\delta = 10^{-8}$.

(b) $\delta = 10^{-12}$.

(c) $\delta = 10^{-16}$.

(d) $\delta = 10^{-20}$.

Figure 4.13: Exact errors in the $A-$norm for TP2 with $n = 29^2$, $k = 5$ and various termination criterion.



(a) $\delta = 10^{-8}$.

(b) $\delta = 10^{-10}$.

(c) $\delta = 10^{-12}$.

(d) $\delta = 10^{-14}$.

Figure 4.14: Exact errors in the $A-$norm for TP3 with $n = 64^2, k = 8^2, \epsilon = 10^{-3}$ and various termination criterion.

4.5 Numerical Results using Perturbed Starting Vectors

In the methods DEF2, A-DEF2, R-BNN1 and R-BNN2 we have to use the special starting vector x_s defined by

$$x_s := Qb + P^T x_0, \quad x_0 \text{ is random.} \tag{4.7}$$

If BNN is also based on x_s then we have shown in Section 3.3 that the methods are theoretically identical. Hence, these methods perform the same in exact arithmetics. In this section, we will perturb x_s in DEF2, A-DEF2, R-BNN1 and R-BNN2, denoted as $\tilde{x}_s$ which is defined as follows:

$$\tilde{x}_s := (1 + \gamma y_0) x_s, \quad \gamma \geq 0, \tag{4.8}$$

where y_0 is a random vector with elements from the interval $[-0.5, 0.5]$. Using $\tilde{x}_s$ for different γ, we perform the same numerical experiments as in the previous sections and investigate whether the associated methods are sensitive for these perturbed starting vectors.

Recall that if we use the original starting vector in DEF2, A-DEF2, R-BNN1 and R-BNN2, then they give exactly the same results and solution as BNN in exact arithmetic. Although DEF2, R-BNN1 and R-BNN2 correspond to a singular operator, a unique solution can be obtained in this case. However, in the case of perturbed starting vectors, there are no equivalences anymore between BNN and the other methods. If DEF2, R-BNN1 or R-BNN2 converges, the solution is non-unique. Therefore, if one of these methods converges but not to the right solution, then we correct this non-unique solution in the experiments using the 'uniqueness' step, similar to DEF1 mentioned in Remark 2.21. Note that this uniqueness step is not required in the A-DEF2 method since it corresponds to a non-singular operator.

4.5.1 Test Problem 1: Laplace Problem

The results considering TP1 can be found in Table 4.11 and Figure 4.15.

Method	$\gamma = 0$		$\gamma = 10^{-8}$		$\gamma = 10^{-6}$		$\gamma = 10^{0}$	
	# It.	$\|\|x - x_i\|\|_2$	# It.	$\|\|x - x_i\|\|_2$	# It.	$\|\|x - x_i\|\|_2$	# It.	$\|\|x - x_i\|\|_2$
DEF2	44	5.0×10^{-7}	44	5.4×10^{-7}	NC	$1.7 \times 10^{+12}$	NC	$2.5 \times 10^{+18}$
A-DEF2	45	2.1×10^{-7}	45	2.1×10^{-7}	45	2.1×10^{-7}	51	5.6×10^{-7}
R-BNN1	45	2.1×10^{-7}	45	2.1×10^{-7}	45	2.1×10^{-7}*	44	4.5×10^{-7}*
R-BNN2	45	2.1×10^{-7}	45	4.4×10^{-7}	NC	3.0×10^{-5}	NC	7.7×10^{0}

Table 4.11: Number of required iterations for convergence and the 2−norm of the exact error of all proposed methods for TP1 with parameters $n = 29^2$ and $k = 5$ and perturbed starting vectors. An asterisk (*) means that an extra uniqueness step is applied in that test case.

Observation 4.11. *Considering both Table 4.11 and Figure 4.15, it can be seen that*

- *for $\gamma \leq 10^{-8}$, all involved methods perform well;*
- *for $\gamma \geq 10^{-6}$, DEF2 and R-BNN2 fail to converge, while A-DEF2 and R-BNN1 still work appropriately. This latter observation even holds for relatively large γ, although A-DEF2 requires somewhat more iterations.*

Method	$\gamma = 0$		$\gamma = 10^{-8}$		$\gamma = 10^{-6}$		$\gamma = 10^{0}$	
	# It.	$\|\|x - x_i\|\|_2$	# It.	$\|\|x - x_i\|\|_2$	# It.	$\|\|x - x_i\|\|_2$	# It.	$\|\|x - x_i\|\|_2$
DEF2	53	2.2×10^{-7}	53	2.2×10^{-7}	NC	$1.2 \times 10^{+12}$	NC	$4.0 \times 10^{+18}$
A-DEF2	53	2.5×10^{-7}	53	2.5×10^{-7}	53	2.1×10^{-7}	54	2.5×10^{-7}
R-BNN1	53	2.3×10^{-7}	53	2.5×10^{-7}	53	$2.3 \times 10^{-7*}$	53	$2.7 \times 10^{-7*}$
R-BNN2	53	2.2×10^{-7}	53	2.5×10^{-7}	53	$1.6 \times 10^{-5*}$	NC	$1.6 \times 10^{+1}$

Table 4.12: Number of required iterations for convergence and the 2−norm of the exact error of all proposed methods for TP2 with parameters $n = 29^2, k = 5, \epsilon = 10^{-6}$ and perturbed starting vectors. An asterisk (*) means that an extra uniqueness step is applied in that test case.

4.5.2 Test Problem 2: Layer Problem

Results of TP2 can be found in Table 4.12 and Figure 4.16.

Observation 4.12. *Considering both Table 4.12 and Figure 4.16, similar results as in the previous subsection can be obtained, i.e.,*

- *for $\delta \leq 10^{-8}$, we see that all involved methods converge appropriately;*
- *for $\delta \geq 10^{-6}$, it can be observed that DEF2 and R-BNN2 fail in convergence or converge to the wrong solution (even after the uniqueness step), while A-DEF2 and R-BNN1 still converge.*

4.5.3 Test Problem 3: Bubbly Flow Problem

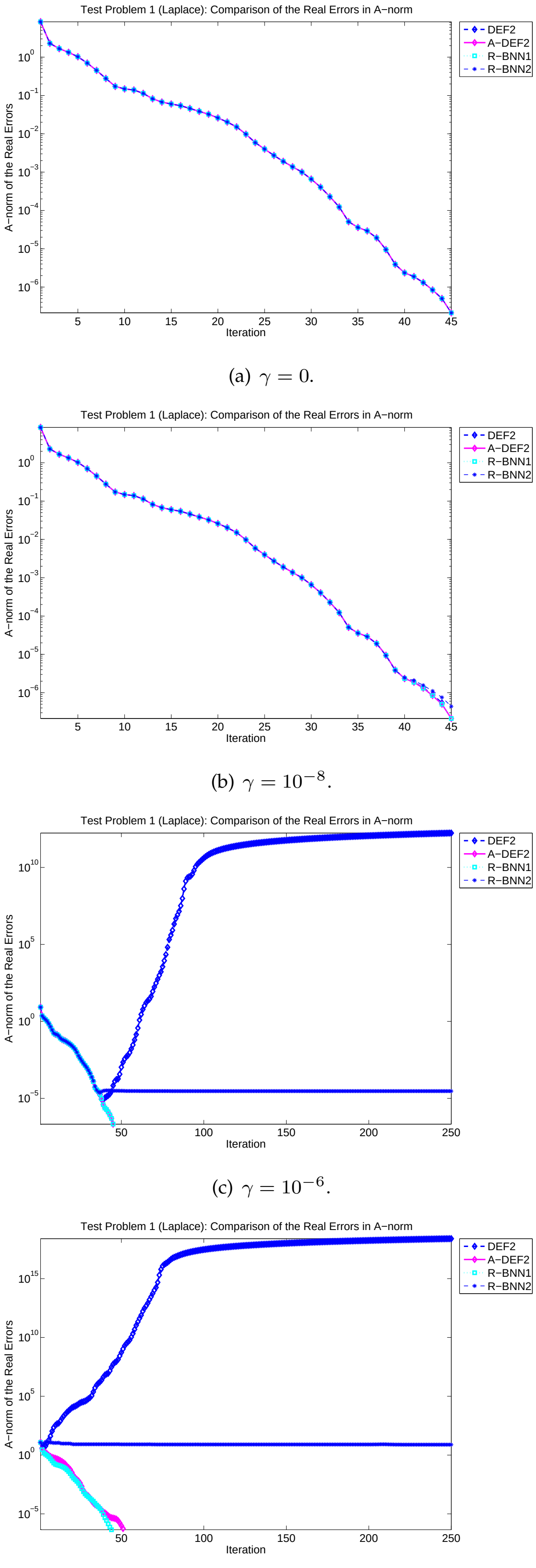
Results of TP3 are presented in Table 4.13 and Figure 4.17.

Method	$\gamma = 0$		$\gamma = 10^{-8}$		$\gamma = 10^{-6}$		$\gamma = 10^{0}$	
	# It.	$\|\|x - x_i\|\|_2$	# It.	$\|\|x - x_i\|\|_2$	# It.	$\|\|x - x_i\|\|_2$	# It.	$\|\|x - x_i\|\|_2$
DEF2	39	1.1×10^{-6}	NC	$7.3 \times 10^{+2}$	NC	$5.0 \times 10^{+4}$	NC	$2.3 \times 10^{+10}$
A-DEF2	40	1.1×10^{-6}	40	1.1×10^{-6}	40	1.1×10^{-6}	41	1.2×10^{-6}
R-BNN1	40	1.1×10^{-6}	40	1.1×10^{-6}	40	$1.1 \times 10^{-6*}$	40	$1.5 \times 10^{-6*}$
R-BNN2	40	1.1×10^{-6}	40	1.6×10^{-6}	NC	6.5×10^{-5}	NC	$4.3 \times 10^{+1}$

Table 4.13: Number of required iterations for convergence and the 2−norm of the exact error of all proposed methods for TP3 with parameters $n = 64^2, k = 8^2, \epsilon = 10^{-3}$ and perturbed starting vectors. An asterisk (*) means that an extra uniqueness step is applied in that test case.

Observation 4.13. *Considering both Table 4.13 and Figure 4.17, we observe that*

- *all involved methods perform well for $\gamma \leq 10^{-8}$, except for DEF2 which fails in the case of $\gamma = 10^{-8}$.*
- *for $\gamma \geq 10^{-6}$, DEF2 and R-BNN2 fail in convergence, while A-DEF2 and R-BNN1 still work fine. The latter observation even holds for relatively large γ.*



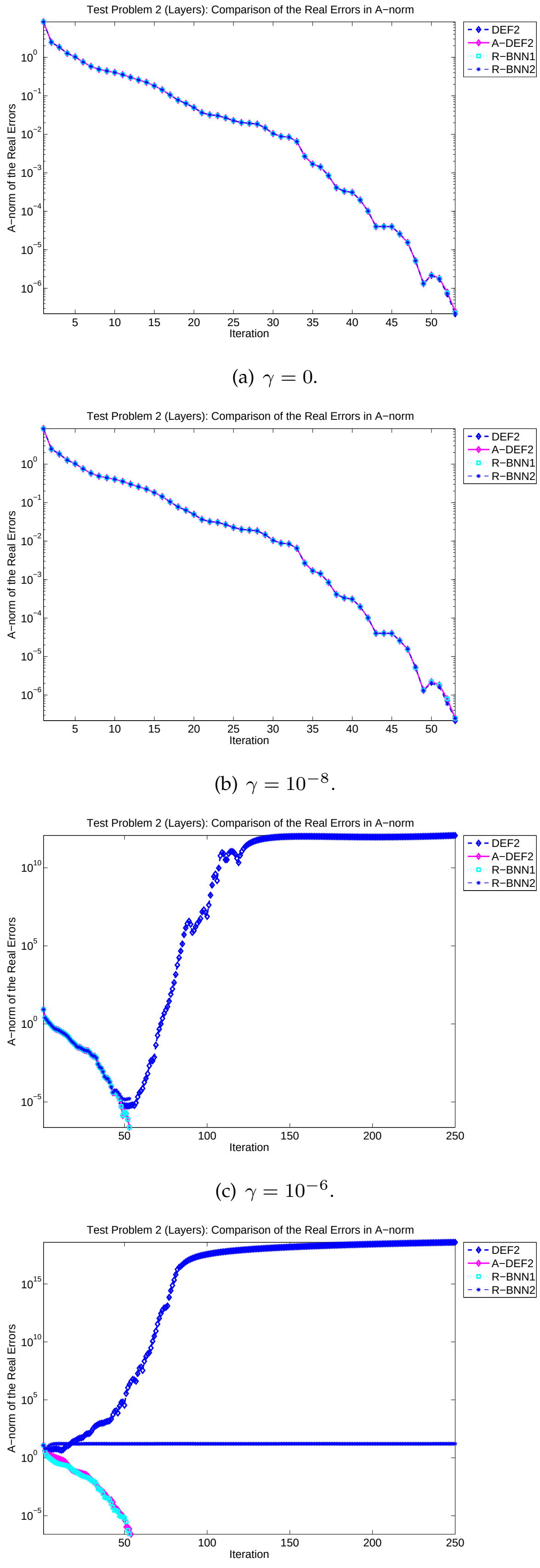
(a) $\gamma = 0$.

(b) $\gamma = 10^{-8}$.

(c) $\gamma = 10^{-6}$.

(d) $\gamma = 1$.

Figure 4.15: Exact errors in the $A-$norm for TP1 with $n = 29^2, k = 5^2$ and perturbed starting vectors.



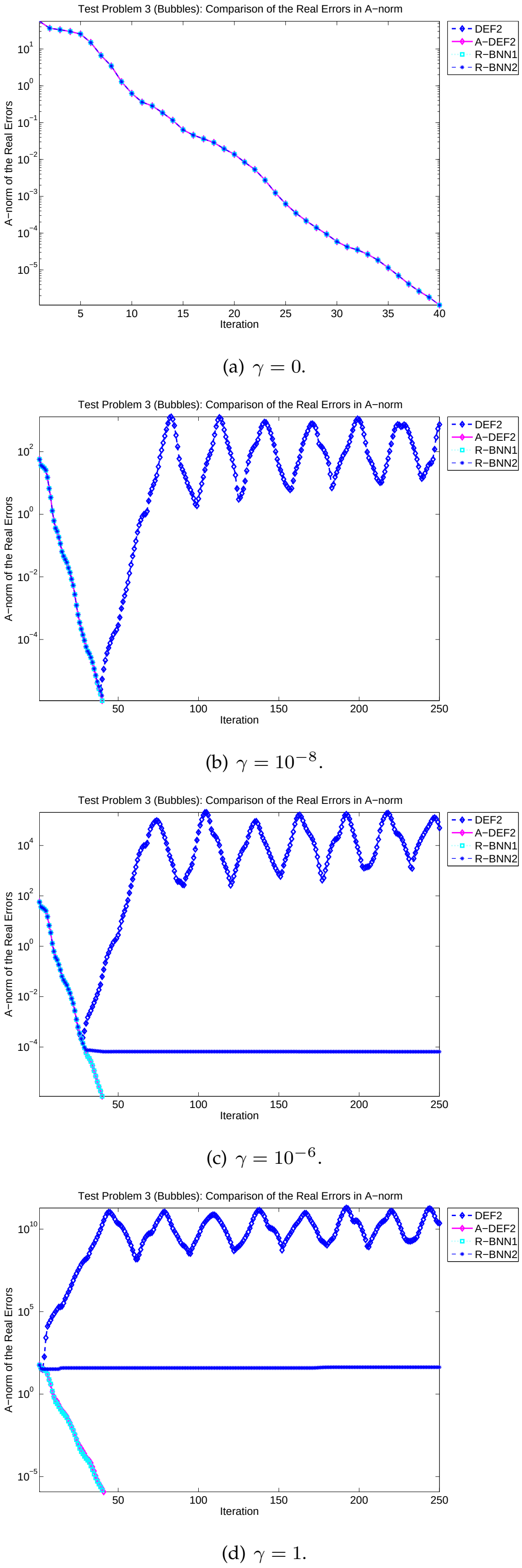
(a) $\gamma = 0$.

(b) $\gamma = 10^{-8}$.

(c) $\gamma = 10^{-6}$.

(d) $\gamma = 1$.

Figure 4.16: Exact errors in the $A-$norm for TP2 with $n = 29^2, k = 5^2$ and perturbed starting vectors.



(a) $\gamma = 0$.

(b) $\gamma = 10^{-8}$.

(c) $\gamma = 10^{-6}$.

(d) $\gamma = 1$.

Figure 4.17: Exact errors in the $A-$norm for TP3 with $n = 64^2, k = 8^2$ and perturbed starting vectors.

4.6 Reorthogonalization Strategies for Non-Converging Methods

In each of the previous sections we have seen some test cases where several methods show divergence, stagnation or erratic behavior by considering the exact errors. One of the reasons is that the residuals within the iteration process gradually lost the orthogonality with respect to the columns of Z, which is also observed by e.g. Saad et al. [29]. For example, in DEF1 we should have

$$Z^T \hat{r}_j = Z^T P r_j = \mathbf{0}, \quad j = 1, 2, \ldots, \tag{4.9}$$

where we have used Lemma 3.2d. Moreover, due to the starting vector x_s in DEF2, A-DEF2, R-BNN1 and R-BNN2, the following expression should be satisfied in these methods:

$$Z^T r_j = \mathbf{0}, \quad j = 1, 2, \ldots, \tag{4.10}$$

see also Exp. (3.6). However, it appears that in the bad converging methods, the corresponding Equations (4.9) and (4.10) do not hold during the iteration process.

One remedy to recover orthogononality in the bad-converging methods is described in e.g. [29]. Define first the so-called reorthogonalization operator W as

$$W := I - Z(Z^T Z)^{-1} Z^T. \tag{4.11}$$

Then, W is orthogonal to the deflation subspace matrix Z, i.e.,

$$Z^T W := Z^T (I - Z(Z^T Z)^{-1} Z^T) = Z^T - Z^T = \mathbf{0}. \tag{4.12}$$

Subsequently, orthogonality of the residuals within the iterative process can be preserved by multiplying r_j by W right after r_j is computed in the algorithms:

$$r_j := W r_j, \quad j = 1, 2, \ldots. \tag{4.13}$$

For the DEF1 algorithm, one substitutes $\hat{r}_j (= P r_j)$ into r_j of Exp. (4.13). As a consequence, the new residuals satisfy (4.9) or (4.10) due to (4.12).

Remark 4.2. *Relations like (4.9) and (4.10) do not hold in the algorithms of AD, A-DEF1 and BNN. In the case of AD and BNN, this is not bad because they appear extremely stable in the most test cases. This is in contrast to A-DEF1 which is unstable in several numerical experiments. The instability of A-DEF1 in these cases can not be resolved using the above described reorthogonalization strategies.*

Remark 4.3. *Note that the reorthogonalization operator (4.13) is relatively cheap. Due to the construction of the subdomain deflation vectors, $Z^T Z$ is also a diagonal matrix which is cheap to obtain. Furthermore, the other operations within the reorthogonalization-operator W are also cheap to perform.*

In the next subsections, a few numerical experiments of the previous sections will be repeated, using the above described reorthogonalization strategy for the non-converging methods.

4.6.1 Results using Standard Parameters: TP3 with $k = 8^2$ and $\epsilon = 10^{-6}$

We reconsider TP3 with $k = 8^2$ and $\epsilon = 10^{-6}$ (Subsection 4.2.3). In this test case, we have seen that DEF1 and DEF2 did not converge within 250 iterations, whereas the other methods converged appropriately. The original results and the new results using reorthogonalization of the residuals of DEF1 and DEF2 can be found in Table 4.14 and Figure 4.18.

Method	# It.	$\|\|x - x_i\|\|_2$
PREC	180	1.7×10^{-6}
AD	60	1.1×10^{-6}
DEF1	40 (NC)	4.4×10^{-3} (1.1×10^{0})
DEF2	40 (NC)	4.8×10^{-3} $(1.2 \times 10^{+2})$
A-DEF1	46	1.4×10^{-6}
A-DEF2	41	1.0×10^{-6}
BNN	41	1.0×10^{-6}
R-BNN1	41	1.0×10^{-6}
R-BNN2	41	1.2×10^{-6}

Table 4.14: Number of required iterations for convergence and the 2−norm of the exact error of all proposed methods for TP3 with parameters $n = 64^2, k = 8^2, \epsilon = 10^{-3}$ and standard parameters. The reorthogonalization strategy has been used for DEF1 and DEF2. The results without reorthogonalization are given in the brackets.

Observation 4.14. *Based on Table 4.14 and Figure 4.18 we observe that*

- *after reorthogonalization of the residuals for DEF1 and DEF2, all proposed methods converge smoothly by considering the residuals;*

- *however, the consequence of the modifications of the residuals is that we do not obtain an accurate solution by investigating the exact errors; the exact errors of both DEF1 and DEF2 stagnate using the reorthogonalization process and the final exact error is not below the smallest exact error during the iteration process without reorthogonalization.*

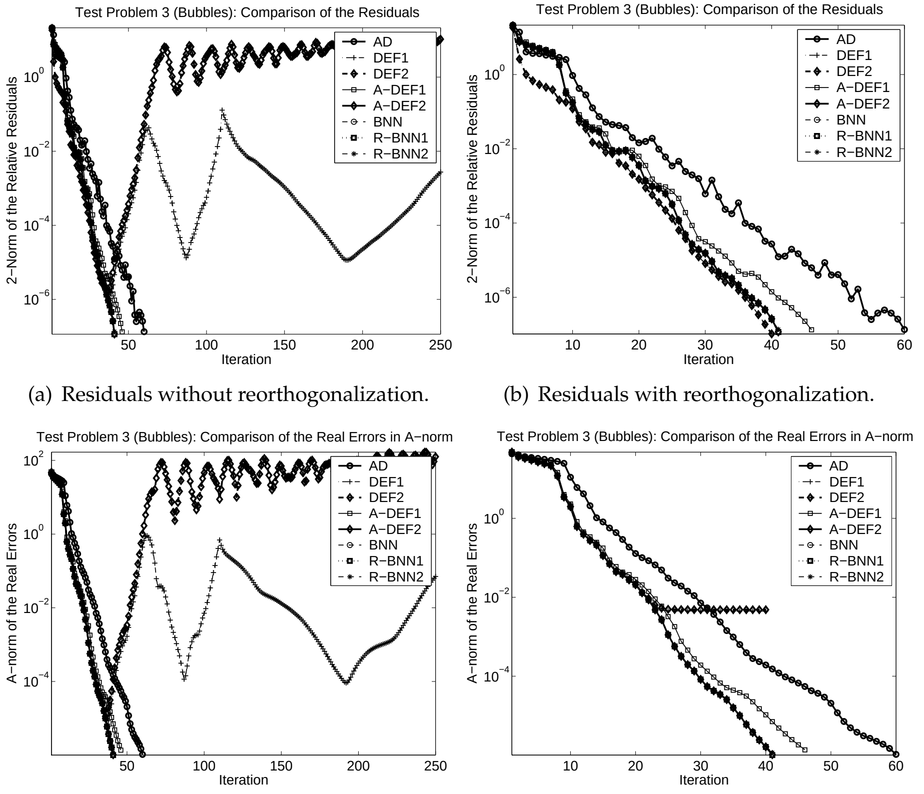
4.6.2 Results using Inaccurate Coarse Solves: TP1 with $\psi = 10^{-4}$

We consider again TP1 with $\psi = 10^{-4}$ (Subsection 4.3.1), where DEF2 and R-BNN2 have not converged within 250 iterations, while DEF1 and R-BNN1 did converge but not to an accurate solution. The original and new results using reorthogonalization of the residuals for DEF1, DEF2, R-BNN1 and R-BNN2 can be found in Table 4.15 and Figure 4.19.

Observation 4.15. *Based on Table 4.15 and Figure 4.19, the same observations as in the previous subsection can be done:*

- *after reorthogonalization of the residuals for DEF1, DEF2, R-BNN1 and R-BNN2, all proposed methods converge smoothly by considering the residuals;*

- *the exact errors of DEF1, DEF2, R-BNN1 and R-BNN2 stagnate using the reorthogonalization process and the final exact error is not below the smallest exact error during the original iteration process.*



(a) Residuals without reorthogonalization.

(b) Residuals with reorthogonalization.

(c) Exact errors without reorthogonalization.

(d) Exact errors with reorthogonalization.

Figure 4.18: Residuals in the 2−norm and exact errors in the A−norm for TP3 with $n = 64^2, k = 8^2, \epsilon = 10^{-3}$ and standard parameters in the cases without and with reorthogonalization of DEF1 and DEF2.

4.6.3 Results using Severe Termination Criteria: TP2 with $\delta = 10^{-16}$

We reconsider TP2 with $\delta = 10^{-16}$ (Subsection 4.4.2). In this test case, we have seen that DEF1, DEF2 and R-BNN2 do not converge within 250 iterations. The old and new results of DEF1, DEF2 and R-BNN2 can be found in Table 4.16 and Figure 4.20.

Observation 4.16. *Considering Table 4.16 and Figure 4.20, the same observations as in the previous subsections can be made:*

- *after reorthogonalization of the residuals for DEF1, DEF2 and R-BNN2, all proposed methods converge appropriately by considering the residuals;*
- *we do not obtain accurate solutions for DEF1, DEF2 and R-BNN2 by investigating the exact errors.*

4.6.4 Results using Perturbed Starting Vectors: TP3 with $k = 8^2, \epsilon = 10^{-3}$ and $\gamma = 10^{-6}$

We reconsider TP3 with $\gamma = 10^{-6}$ (Subsection 4.5.3). In this test case, DEF2 and R-BNN2 did not converge. Both original and new results using reorthogonalization for DEF2 and R-BNN2 can be found in Table 4.17 and Figure 4.21.

Method	# It.	$\|\|x - x_i\|\|_2$
PREC	57	4.4×10^{-7}
AD	47	9.0×10^{-7}
DEF1	44 (135)	1.1×10^{-2} (1.1×10^{-7})
DEF2	44 (NC)	1.1×10^{-2} $(2.1 \times 10^{+3})$
A-DEF1	44	1.0×10^{-6}
A-DEF2	45	2.1×10^{-7}
BNN	45	2.1×10^{-7}
R-BNN1	45 (71)	1.1×10^{-2} (8.3×10^{-4})
R-BNN2	45 (NC)	1.1×10^{-2} (1.1×10^{-2})

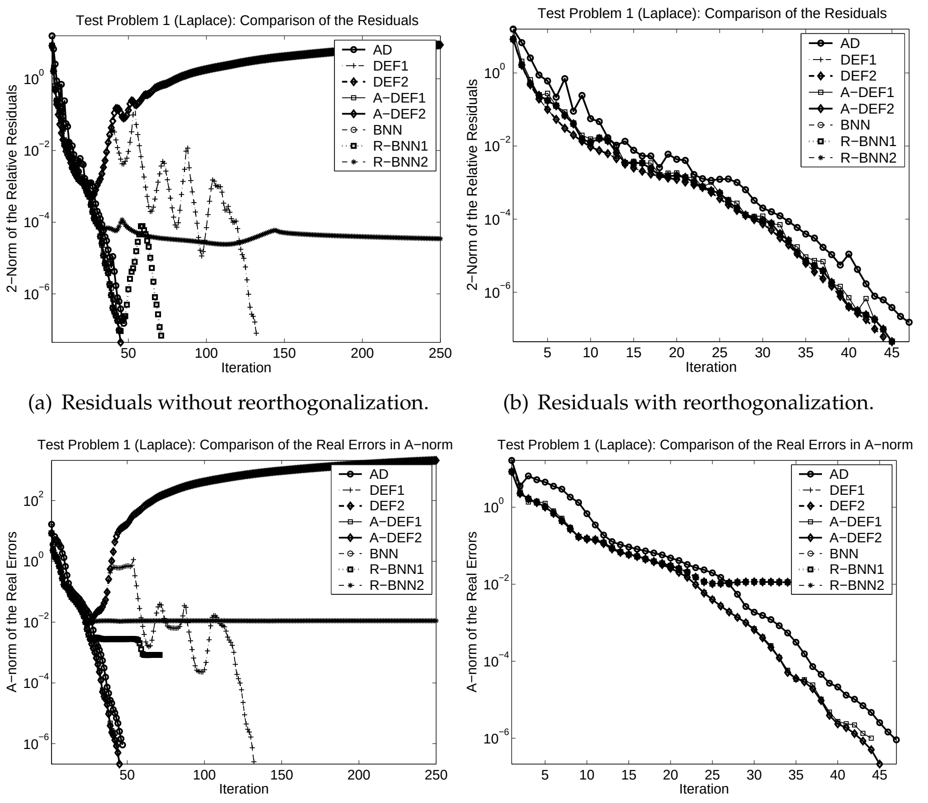
Table 4.15: Number of required iterations for convergence and the 2−norm of the exact error of all proposed methods for TP1 with parameters $n = 29^2, k = 5$ and inaccurate coarse solves ($\psi = 10^{-4}$). The reorthogonalization strategy has been used for DEF1, DEF2, R-BNN1 and R-BNN2. The results without reorthogonalization are given in the brackets.

Method	# It.	$\|\|x - x_i\|\|_2$
PREC	201	3.6×10^{-8}
AD	87	2.0×10^{-8}
DEF1	81 (NC)	6.0×10^{-8} (1.4×10^{-6})
DEF2	81 (NC)	7.7×10^{-8} $(2.3 \times 10^{+5})$
A-DEF1	136	2.5×10^{-8}
A-DEF2	81	7.0×10^{-9}
BNN	81	6.1×10^{-9}
R-BNN1	81	2.9×10^{-8}
R-BNN2	81 (NC)	7.7×10^{-8} (7.7×10^{-8})

Table 4.16: Number of required iterations for convergence and the 2−norm of the exact error of all proposed methods for TP2 with parameters $n = 29^2, k = 5$ and severe termination criterion ($\delta = 10^{-16}$). The reorthogonalization strategy has been used for DEF1, DEF2 and R-BNN2. The results without reorthogonalization are given in the brackets.

Method	# It.	$\|\|x - x_i\|\|_2$
DEF2	39 (NC)	5.9×10^{-3} $(5.0 \times 10^{+4})$
A-DEF2	40	1.1×10^{-6}
R-BNN1	40	1.1×10^{-6}
R-BNN2	NC (NC)	1.2×10^{-1} (6.5×10^{-5})

Table 4.17: Number of required iterations for convergence and the 2−norm of the exact error of all proposed methods for TP3 with parameters $n = 64^2, k = 8^2, \epsilon = 10^{-3}$ and perturbed starting vector ($\gamma = 10^{-6}$). The reorthogonalization strategy has been used for DEF2 and R-BNN2. The results without reorthogonalization are given in the brackets.



(a) Residuals without reorthogonalization.

(b) Residuals with reorthogonalization.

(c) Exact errors without reorthogonalization.

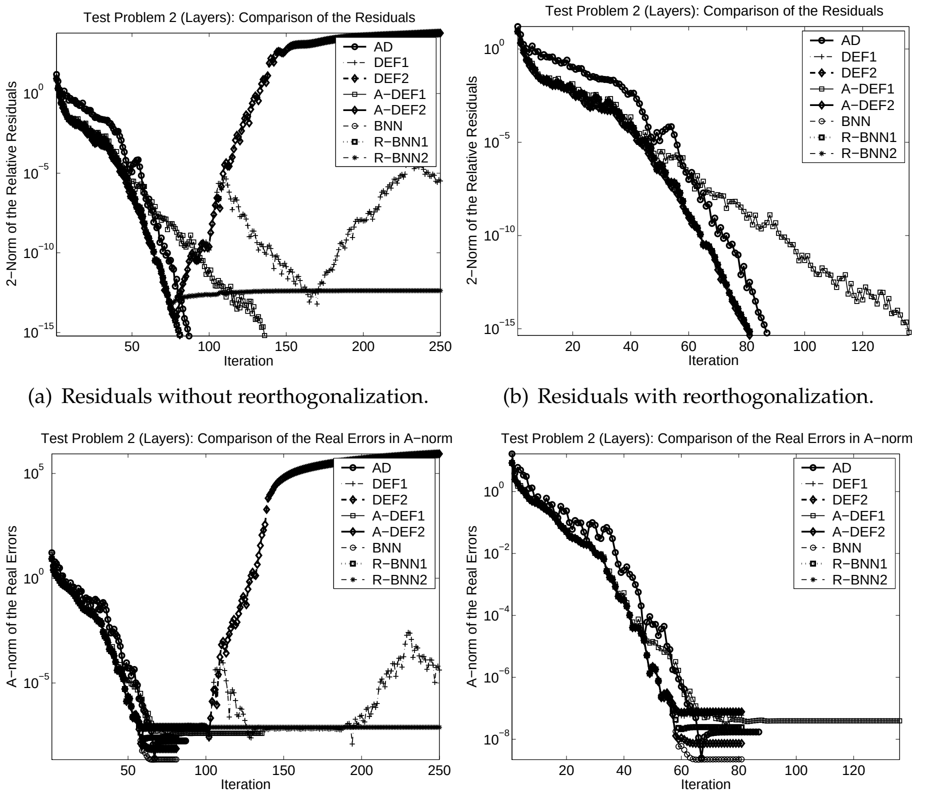
(d) Exact errors with reorthogonalization.

Figure 4.19: Residuals in the 2−norm and exact errors in the A−norm for TP1 with $n = 29^2$, $k = 5$ and standard parameters in the cases without and with reorthogonalization of DEF1, DEF2, R-BNN1 and R-BNN2.

Observation 4.17. *Based on Figure 4.21, we notice that*

- *by considering the residuals, DEF2 converges whereas R-BNN2 still show very erratic behavior (as seen earlier in some test cases of DEF1) after reorthogonalization; This may be caused by the nearly zero-eigenvalues which contribute to the convergence;*

- *by considering the exact errors, DEF2 stagnates while R-BNN2 shows erratic behavior.*



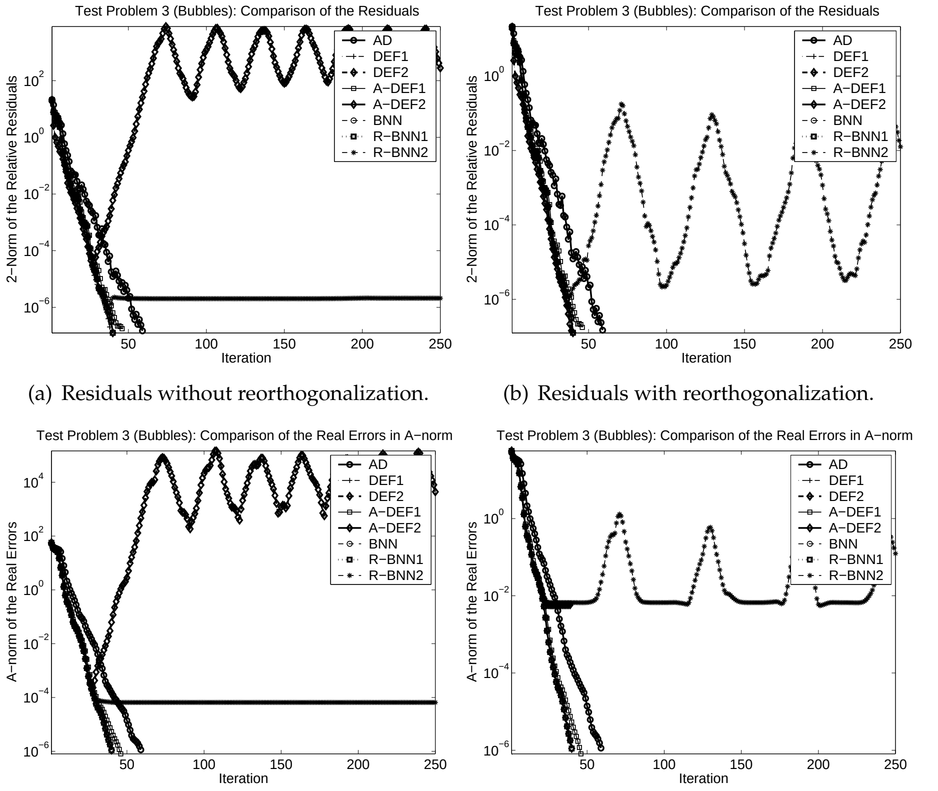
(a) Residuals without reorthogonalization.

(b) Residuals with reorthogonalization.

(c) Exact errors without reorthogonalization.

(d) Exact errors with reorthogonalization.

Figure 4.20: Residuals in the $2-$norm and exact errors in the $A-$norm for TP2 with $n = 29^2, k = 5$ and severe termination criterion ($\delta = 10^{-16}$) in the cases without and with reorthogonalization of DEF1, DEF2 and R-BNN2.



(a) Residuals without reorthogonalization.

(b) Residuals with reorthogonalization.

(c) Exact errors without reorthogonalization.

(d) Exact errors with reorthogonalization.

Figure 4.21: Residuals in the 2−norm and exact errors in the A−norm for TP3 with $n = 64^2, k = 8^2, \epsilon = 10^{-3}$ and perturbed starting vector ($\gamma = 10^{-6}$) in the cases without and with reorthogonalization of DEF2 and R-BNN2.

CHAPTER 5
Conclusions

In this paper we have compared several projection methods coming from deflation, domain decomposition and multigrid, both theoretically and numerically. The conclusions are drawn below.

Theoretical Results with Eigenvalue Distributions

Theoretically, it has been proved that DEF1 is the best method [23–25]. In this paper, we have seen that the operators of all projection methods, except for additive coarse grid correction, have comparable eigenvalue distributions. In fact, there are two classes consisting of identical methods in the sense of eigenvalues. The first class consists of DEF1, DEF2, R-BNN1 and R-BNN2 and the second class includes BNN, A-DEF1 and A-DEF2. It has also been shown that the associated spectrum of the methods of the first class is more favorable than those of the second class. This means for instance that in numerical experiments one should apply deflation rather than balancing provided that both methods are stable in these experiments.

Theoretical Results with Special Starting Vectors

It has been shown that the expensive operator of the BNN method can be reduced to simplier operators which are used in the DEF2, A-DEF2, R-BNN1 and R-BNN2 methods for special starting vectors. Hence, some methods of the two classes with the same spectral properties are the same in exact arithmetics.

Numerical Results with Standard Parameters

From a numerical point of view, we have observed that the theoretical results only hold for standard parameters with exact computations of the coarse matrices E^{-1}, appropriate choices for the termination criterion, no perturbations in the starting vectors etcetera. In these cases the numerical results confirm the theory that all projection methods perform more or less the same, although A-DEF1 shows problems in some test cases. The last observation gives no lose of generalization since the application of the non-SPSD operator of A-DEF1 in CG can not be fully understood theoretically.

Numerical Results with Inaccurate Coarse Solves

If the dimensions of the coarse matrix E become large, it is not beneficial to perform the computations with E^{-1} in a direct manner; one should turn to for example iterative solvers and it will be favorable to do that inaccurately to save computational time. For small perturbations, the projection methods still work fine using these iterative solvers. However, in the second part of the experiments we have seen that DEF1, DEF2 and also R-BNN1 and R-BNN2 are sensitive for perturbations in E^{-1}. Hence, these methods do not look to be appropriate for inaccurate coarse solves with large perturbations. The best methods in these experiments are BNN, A-DEF1 and A-DEF2.

Numerical Results with Severe Termination Criterion

If matrix A is ill-conditioned and the tolerance of the termination criterion chosen by the user becomes too severe, it will be advantageous if the projection method will not fail to converge. By doing some numerical experiments with varying tolerances, we have seen that both DEF1 and DEF2 have problems for a too strict tolerance while the other methods work properly for severe termination criteria. In other words, BNN, A-DEF1, A-DEF2, R-BNN1 and R-BNN2 have no problems with strict tolerances.

Numerical Results with Perturbed Special Starting Vector

As mentioned above, BNN is theoretically equal to DEF2, A-DEF2, R-BNN1 and R-BNN2 if the special starting vector has been used. Besides the fact that some of these reduced variants (namely R-BNN1 and R-BNN2) are not able to deal with inaccurate coarse solves, some of them are also sensitive for perturbations in the special starting vector. Both DEF2 and R-BNN2 are unstable, while BNN, A-DEF2 and R-BNN1 still work fine after these perturbations. These can be of importance if one uses multigrid-like subdomains where the number of subdomains k is very large and the starting vector can not be obtained accurately.

Numerical Results with Reorthogonalization Techniques

Reorthogonalization of the update residuals during the iterative process may improve the non-converging methods in the sense that a solution can be found in considerable time. However, due to these corrections accurate solutions can not be obtained and, therefore, these reorthogonalization techniques do not seem to be an appropriate remedy for non-converging projection methods.

Main Conclusions

DEF1 is the best method considering the theory and the amount of work per iteration. In numerical experiments, DEF1 also works fine, if one applies a realistic stopping criterion and relatively small perturbations of E^{-1}. If one should choose between DEF1 and DEF2, it is obvious to choose for DEF1, although they both show the same stability properties. However, the main difference is that DEF2 is the only method which diverges if it is unstable, while DEF1 is more robust in the sense that if it is unstable then the solution stagnates or the convergence is slow.

Moreover, one should realize that if one wants to perform BNN at low cost by choosing a special starting vector then these variants are as unstable as DEF1 or DEF2. If one uses perturbed starting vectors, then DEF2 and the most reduced variants of BNN are unstable. As a consequence, DEF1 can be better applied rather than these methods.

Finally, after doing the numerical experiments and considering all numerical aspects, we conclude that BNN and A-DEF2 are the best c.q. the most stable methods. However, in the operators of BNN two deflation matrices are involved, which makes the method expensive in use. On the other hand, we have A-DEF2 which behaves like BNN both theoretically and numerically, but the advantage of A-DEF2 is that computations with the associated operator can be done at lower cost since only one deflation matrix is involved. Hence, A-DEF2 is the best method considering the theory, numerical experiments and the computational costs.

BIBLIOGRAPHY

[1] S.F. Ashby, T.A. Manteuffel and P.E. Saylor, *A taxonomy for conjugate gradient methods*, SIAM J. Numer. Anal., **27**, pp. 1542–1568, 1990.

[2] R. Blaheta, *Multilevel Iterative Methods and Deflation*, ECCOMAS CFD 2006, European conference on Computational Fluid Dynamics, Egmond aan Zee, The Netherlands, September 5-8, 2006, Eds.: P. Wesseling, E. Onate and J. Periaux, 2006.

[3] J.H. Bramble, J.E. Pasciak and A.H. Schatz, *The construction of preconditioners for elliptic problems by substructuring*, I, Math. Comp, **47** , pp. 103–134, 1986.

[4] M. Dryja, *An additive Schwarz algorithm for two- and three dimensional finite element elliptic problems*, Domain Decomposition methods, SIAM, Philadelphia, pp. 168–172, 1989.

[5] M. Dryja and O.B. Widlund, *Towards a unified theory of domain decomposition algorithms for elliptic problems*, Third International Symposium on Domain Decomposition Methods for Partial Differential Equations, SIAM, Philadelphia, pp. 3–21, 1990.

[6] M. Dryja and O.B. Widlund, *Schwarz Methods of Neumann-Neumann Type for Three-Dimensional Elliptic Finite Element Problems*, Comm. Pure Appl. Math., **48**(2), pp. 121–155, 1995.

[7] J. Ehrel and F. Guyomarc'h, *An augmented subspace conjugate gradient*, Report RR-3278, INRIA, Rocquencourt, France, 1997.

[8] M. Eiermann, O.G. Ernst and O. Schneider, *Analysis of acceleration strategies for restarted minimal residual methods*, J. Comput. Appl. Math., **123**, pp. 261–292, 2000.

[9] J. Frank and C. Vuik, *On the construction of deflation-based preconditioners*, SIAM Journal on Scientific Computing, **23**, pp. 442–462, 2001.

[10] H. de Gersem, K. Hameyer, *A deflated iterative solver for magnetostatic finite element models with large differences in permeability*, Eur. Phys. J. Appl. Phys., **13**, pp. 45–49, 2000.

[11] L Giraud and S Gratton, *On the sensitivity of some spectral preconditioners*, SIAM J. Matrix Anal. App., **27**, pp. 1089–1105, 2006.

[12] I.G. Graham and R. Scheichl, *Robust Domain Decomposition Algorithms for Multiscale PDEs*, BICS Preprint 14/06, University of Bath, 2006.

[13] M.R. Hestenes and E. Stiefel, *Methods of Conjugate Gradients for Solving Linear Systems*, J. Res. Nat. Bur. Stand., **49**, pp. 409–436, 1952.

[14] E.F. Kaasschieter, *Preconditioned conjugate gradients for solving singular systems*, Journal of Computational and Applied Mathematics , **24**, pp. 265–275, 1988.

[15] L. Yu. Kolotilina, *Preconditioning of systems of linear algebraic equations by means of twofold deflation*, I. Theory, J. Math. Sci., **89**, pp. 1652–1689, 1998.

[16] J. Mandel, *Balancing domain decomposition*, Commun. Appl. Numer. Meth., **9**, pp. 233–241, 1993.

[17] J. Mandel, *Hybrid domain decomposition with unstructured subdomains*, Domain Decomposition Methods in Science and Engineering, Sixth International Conference of Domain Decomposition, Como, Italy, June 15-19, 1994.

[18] J. Mandel and M. Brezina, *Balancing domain decomposition for problems with large jumps in coefficients*, Mathematics of Computation, **216**, pp. 1387–1401, 1996.

[19] L. Mansfield, *On the Conjugate Gradient Solution of the Schur Complement System Obtained from Domain Decomposition*, SIAM J. Num. Anal., **27**, pp. 1612-1620, 1990.

[20] L. Mansfield, *Damped Jacobi preconditioning and coarse grid deflation for Conjugate Gradient iteration on parallel computers*, SIAM J. Sci. Stat. Comput., **12**, pp. 1314–1323, 1991.

[21] J.A. Meijerink and H.A. Van der Vorst, *An iterative solution method for linear systems of which the coefficient matrix is a symmetric M-matrix*, Mathematics of Computation, **31**, pp. 148–162, 1977.

[22] R.B. Morgan, *A restarted GMRES method augmented with eigenvectors*, SIAM J. Matrix Analysis and Applications, **16**, pp. 1154–1171, 1995.

[23] R. Nabben and C. Vuik, *A comparison of Deflation and Coarse Grid Correction applied to porous media flow*, SIAM J. Numer. Anal., **42**, pp. 1631–1647, 2004.

[24] R. Nabben and C. Vuik *A Comparison of Deflation and the Balancing Preconditioner*, SIAM J. Sci. Comput.,27, pp. 1742–1759, 2006.

[25] R. Nabben and C. Vuik *A comparison of general versions of deflation, balancing and additive coarse grid correction*, 2006, *submitted*.

[26] R.A. Nicolaides, *Deflation of Conjugate Gradients with applications to boundary value problems*, SIAM J. Matrix Anal. Appl., **24**, pp. 355–365, 1987.

[27] A. Padiy, O. Axelsson, B. Polman, *Generalized augmented matrix preconditioning approach and its application to iterative solution of ill-conditioned algebraic systems*, SIAM J. Matrix Anal. Appl., **22**, pp. 793–818, 2000.

[28] L. F. Pavarino and O. B. Widlund, *Balancing Neumann-Neumann methods for incompressible Stokes equations*, Comm. Pure Appl. Math, **55**(3), pp. 302–335, 2002.

[29] Y. Saad, M. Yeung, J. Erhel and F. Guyomarc'h, *A deflated version of the Conjugate Gradient Algorithm*, SIAM J. Sci. Comput., **21** (5), pp. 1909–1926, 2000.

[30] V. Simoncini and D.B. Szyld, *Recent computational developments in Krylov Subspace Methods for linear systems*, Research Report 05-9-25, Department of Mathematics, Temple University, September 2005 (revised May 2006). *To appear in* Numerical Linear Algebra with Applications.

[31] B. Smith, P. Bjørstad and W. Gropp, *Domain Decomposition*, Cambridge University Press, Cambridge, 1996.

[32] J.M. Tang and C. Vuik, *On Deflation and Singular Symmetric Positive Semi-Definite Matrices*, J. Comp. Appl. Math., 2006, *in press*.

[33] J.M. Tang and C. Vuik, *An Efficient Deflation Method applied on 3-D Bubbly Flow Problems*, *submitted to* ETNA, 2006.

[34] A. Toselli and O.B. Widlund. *Domain Decomposition: Algorithms and Theory*, Computational Mathematics, Vol. 34, Springer, Berlin, 2005.

[35] U. Trottenberg, C.W. Oosterlee and A. Schüller, *Multigrid*, Academic Press, London, 2001.

[36] F. Vermolen, C. Vuik, A. Segal, *Deflation in preconditioned Conjugate Gradient methods for finite element problems*, In: Conjugate Gradient and Finite Element Methods, Ed: M. Křižek and P. Neittaanmäki and R. Glowinski and S. Korotov, Springer, Berlin, pp. 103–129, 2004.

[37] C. Vuik, A. Segal and J.A. Meijerink, *An efficient preconditioned CG method for the solution of a class of layered problems with extreme contrasts in the coefficients*, J. Comp. Phys., **152**, pp. 385–403, 1999.

[38] C. Vuik and J. Frank, *Coarse grid acceleration of a parallel block preconditioner*, Future Generation Computer Systems, **17**, pp. 933–940, 2001.

[39] C. Vuik, R. Nabben and J.M. Tang, *Deflation acceleration for Domain Decomposition Preconditioners*, Proceedings of the 8th European Multigrid Conference on Multigrid, Multilevel and Multiscale Methods,, Ed.: P. Wesseling, C.W. Oosterlee, P. Hemker, The Hague, The Netherlands, September 27-30, 2005.

[40] P. Wesseling, *An Introduction to Multigrid Methods*, John Wiley & Sons Ltd, 1992. Corrected Reprint. Philadelpha: R.T. Edwards, Inc., 2004.