A Comparison of Two-Level Preconditioners
Multigrid and Deflation

Kees Vuik ¹, Jok Tang ², Scott MacLachlan ³, Reinhard Nabben ⁴

¹ Delft University of Technology
Delft Institute of Applied Mathematics

² Vortech Computing

³ Tufts University
Department of Mathematics

⁴ Technische Universität Berlin
Institut für Mathematik

Schnelle Löser für Partielle Differentialgleichungen
Oberwolfach, Germany
May 22nd - 28th, 2011
Main Problem

Problem
Solve the linear system

\[ Ax = b, \quad A \in \mathbb{R}^{n \times n} \]

Properties of Coefficient Matrix A
- Large but sparse
- Real and symmetric
- Nonnegative eigenvalues
- Ill-conditioned (i.e. condition number \( \kappa = \frac{\lambda_{\text{max}}}{\lambda_{\text{min}}} \) is large)
Introduction

Main Problem

**Problem**
Solve the linear system

\[ Ax = b, \quad A \in \mathbb{R}^{n \times n} \]

**Properties of Coefficient Matrix A**
- Large but sparse
- Real and symmetric
- Nonnegative eigenvalues
- Ill-conditioned (i.e. condition number \( \kappa = \frac{\lambda_{\text{max}}}{\lambda_{\text{min}}} \) is large)
Standard Iterative Methods

Preconditioned Conjugate Gradients Method (PCG)¹

Solve iteratively:

\[ M^{-1}Ax = M^{-1}b \]

where \( M^{-1} \) is a preconditioner

Bottleneck

The spectrum of \( M^{-1}A \) often consists of unfavorable eigenvalues

Consequence

Slow convergence of the iterative process

---

Preconditioned Conjugate Gradients Method (PCG)\(^1\)

Solve iteratively:

\[
M^{-1}Ax = M^{-1}b
\]

where \(M^{-1}\) is a preconditioner

Bottleneck

The spectrum of \(M^{-1}A\) often consists of unfavorable eigenvalues

Consequence

Slow convergence of the iterative process

---

Two-Level PCG Method (Two-Level PCG)

Solve iteratively:

\[ \mathcal{P}Ax = \mathcal{P}b \]

where \( \mathcal{P} \) is a \textbf{two-level preconditioner}

Components of \( \mathcal{P} \)

- Traditional preconditioner \( M^{-1} \)
- Projection matrix \( P \)
- Correction matrix \( Q \)

Idea: Eliminate all unfavorable eigenvalues from the spectrum of \( A \)

Consequence

Faster convergence of the iterative process
Two-Level PCG Methods

Solve iteratively:

$$\mathcal{P}Ax = \mathcal{P}b$$

where $\mathcal{P}$ is a two-level preconditioner

**Components of $\mathcal{P}$**

- Traditional preconditioner $M^{-1}$
- Projection matrix $P$
- Correction matrix $Q$

**Idea:** Eliminate all unfavorable eigenvalues from the spectrum of $A$

**Consequence**

Faster convergence of the iterative process
Two-Level Preconditioning

World of Two-Level Preconditioners

- Deflation
- Multigrid
- Domain Decomposition

Connection?
Two-Level PCG Methods

Definition

A two-level PCG method is a PCG method with a two-level preconditioner derived from deflation, multigrid or domain decomposition.

Main Questions

- What is the connection between the different two-level preconditioners?
- Can we construct a generalized two-level PCG method?
- How do the two-level PCG methods behave in practice?
- Which two-level PCG method is the best one?
Two-Level PCG Methods

Definition

A **two-level PCG method** is a PCG method with a two-level preconditioner derived from deflation, multigrid or domain decomposition.

Main Questions

- What is the connection between the different two-level preconditioners?
- Can we construct a generalized two-level PCG method?
- How do the two-level PCG methods behave in practice?
- Which two-level PCG method is the best one?
Introduction

Two-Level PCG Methods

Definition

A two-level PCG method is a PCG method with a two-level preconditioner derived from deflation, multigrid or domain decomposition.

Main Questions

- What is the connection between the different two-level preconditioners?
- Can we construct a generalized two-level PCG method?
- How do the two-level PCG methods behave in practice?
- Which two-level PCG method is the best one?
Outline

1. Introduction
2. Two-Level PCG Methods
3. Comparison of Two-Level PCG Methods
4. Conclusions
1 Introduction

2 Two-Level PCG Methods

3 Comparison of Two-Level PCG Methods

4 Conclusions
Two-Level PCG Methods

Projection Matrix

**Definition of Projection Matrix** $P$ and Correction Matrix $Q$

$$P := I - AQ, \quad Q := ZE^{-1}Z^T, \quad E := Z^T AZ, \quad Z \in \mathbb{R}^{n \times k},$$

where $Z$ is the projection-subspace matrix consisting of projection vectors

**Remarks**

- Space spanned by the columns of $Z$ is the space to be projected out $\rightarrow$
- Effectiveness of $P$ depends on the choice of $Z$
- $E$ has dimensions $k \times k$ $\rightarrow$ $E^{-1}$ might be easy to compute
- $Q$ is an approximation of $A^{-1}$ based on a coarse grid

**Choices of Projection Vectors**

- Approximated eigenvectors (deflation)
- Subdomain vectors (domain decomposition)
- Interpolation / restriction vectors (multigrid)
### Two-Level PCG Methods

#### Projection Matrix

**Definition of Projection Matrix** $P$ and **Correction Matrix** $Q$

\[
P := I - AQ, \quad Q := ZE^{-1}Z^T, \quad E := Z^TAZ, \quad Z \in \mathbb{R}^{n \times k},
\]

where $Z$ is the projection-subspace matrix consisting of projection vectors.

**Remarks**

- Space spanned by the columns of $Z$ is the space to be projected out → Effectiveness of $P$ depends on the choice of $Z$
- $E$ has dimensions $k \times k$ → $E^{-1}$ might be easy to compute
- $Q$ is an approximation of $A^{-1}$ based on a coarse grid

**Choices of Projection Vectors**

- Approximated eigenvectors (deflation)
- Subdomain vectors (domain decomposition)
- Interpolation / restriction vectors (multigrid)
Two-Level PCG Methods

Traditional and Projection Preconditioners

Difference between traditional and projection preconditioners

- $M^{-1}$ is usually an approximation of $A$
- $P$ is a projection matrix

$M^{-1}$ and $P$ should be complementary to each other

Ultimate Goal

Find $M^{-1}$ and $Z$ such that the resulting two-level preconditioner gets rid of all unfavorable eigenvalues of $A$
Two-Level PCG Methods

Traditional and Projection Preconditioners

Difference between traditional and projection preconditioners

- $M^{-1}$ is usually an approximation of $A$
- $P$ is a projection matrix

$M^{-1}$ and $P$ should be complementary to each other

Ultimate Goal

Find $M^{-1}$ and $Z$ such that the resulting two-level preconditioner gets rid of all unfavorable eigenvalues of $A$
Parameters of Two-Level Preconditioners

Parameters can be derived from the theory of
- deflation
- multigrid
- domain decomposition

Interpretation and Choices of Parameters

<table>
<thead>
<tr>
<th></th>
<th>Deflation</th>
<th>Multigrid</th>
<th>DDM</th>
</tr>
</thead>
<tbody>
<tr>
<td>$M^{-1}$</td>
<td>good preconditioner</td>
<td>smoother</td>
<td>subdomain solves</td>
</tr>
<tr>
<td>$P$</td>
<td>deflation matrix</td>
<td>coarse-grid correction</td>
<td>coarse-grid correction</td>
</tr>
<tr>
<td>$Z$</td>
<td>deflation-subspace</td>
<td>interpolation</td>
<td>interpolation</td>
</tr>
<tr>
<td>$Z^T$</td>
<td>deflation-subspace</td>
<td>restriction</td>
<td>restriction</td>
</tr>
<tr>
<td>$k$</td>
<td>$k \ll n$</td>
<td>$1 \ll k$</td>
<td>$1 \ll k \ll n$</td>
</tr>
<tr>
<td>$Ex = y$</td>
<td>direct</td>
<td>recursive</td>
<td>direct / iterative</td>
</tr>
</tbody>
</table>
Background of Two-Level PCG Methods

Parameters of Two-Level Preconditioners

Parameters can be derived from the theory of
- deflation
- multigrid
- domain decomposition

Interpretation and Choices of Parameters

<table>
<thead>
<tr>
<th></th>
<th>Deflation</th>
<th>Multigrid</th>
<th>DDM</th>
</tr>
</thead>
<tbody>
<tr>
<td>$M^{-1}$</td>
<td>good preconditioner</td>
<td>smoother</td>
<td>subdomain solves</td>
</tr>
<tr>
<td>$P$</td>
<td>deflation matrix</td>
<td>coarse-grid correction</td>
<td>coarse-grid correction</td>
</tr>
<tr>
<td>$Z$</td>
<td>deflation-subspace</td>
<td>interpolation</td>
<td>interpolation</td>
</tr>
<tr>
<td>$Z^T$</td>
<td>deflation-subspace</td>
<td>restriction</td>
<td>restriction</td>
</tr>
<tr>
<td>$k$</td>
<td>$k \ll n$</td>
<td>$1 \ll k$</td>
<td>$1 \ll k \ll n$</td>
</tr>
<tr>
<td>$Ex = y$</td>
<td>direct</td>
<td>recursive</td>
<td>direct / iterative</td>
</tr>
</tbody>
</table>
Deflated PCG Method

Solve iteratively:

\[ M^{-1} P A x = M^{-1} P b \]

where \( P = I - AQ \)

Additive Coarse-Grid Correction Method

Solve iteratively:

\[ (M^{-1} + Q) A x = (M^{-1} + Q) b \]

(Abstract) Balancing Neumann-Neumann Method

Solve iteratively:

\[ (P^T M^{-1} P + Q) A x = (P^T M^{-1} P + Q) b \]
Standard Two-Level PCG Methods

Deflated PCG Method 1 2 3
Solve iteratively:

\[ M^{-1}PAx = M^{-1}Pb \]
where \( P = I - AQ \)

Additive Coarse-Grid Correction Method 4
Solve iteratively:

\[ (M^{-1} + Q)Ax = (M^{-1} + Q)b \]

(Abstract) Balancing Neumann-Neumann Method 5
Solve iteratively:

\[ (P^TM^{-1}P + Q)Ax = (P^TM^{-1}P + Q)b \]

---

Two-Level PCG Methods

Deflated PCG Method

Solve iteratively:

\[ M^{-1} PAx = M^{-1} Pb \]

where \( P = I - AQ \)

Additive Coarse-Grid Correction Method

Solve iteratively:

\[ (M^{-1} + Q)Ax = (M^{-1} + Q)b \]

(Abstract) Balancing Neumann-Neumann Method

Solve iteratively:

\[ (P^T M^{-1} P + Q)Ax = (P^T M^{-1} P + Q)b \]

---

Two-Level PCG Methods

Standard Two-Level PCG Methods

Theorem

Solving iteratively:
\[(P^T M^{-1} P + Q)Ax = (P^T M^{-1} P + Q)b\]

is equivalent with solving iteratively:
\[P^T M^{-1} Ax = P^T M^{-1} b\]

using starting vector \(x_0 = P^T \bar{x} + Qb\) with arbitrary \(\bar{x}\)

Reduced Balancing / Deflated PCG Method

Solve iteratively:
\[P^T M^{-1} Ax = P^T M^{-1} b\]

with starting vector \(x_0 = P^T \bar{x} + Qb\)

---

Standard Two-Level PCG Methods

Theorem

Solving iteratively:

\[(P^T M^{-1} P + Q)Ax = (P^T M^{-1} P + Q)b\]

is equivalent with solving iteratively:

\[P^T M^{-1} Ax = P^T M^{-1} b\]

using starting vector \(x_0 = P^T \tilde{x} + Qb\) with arbitrary \(\tilde{x}\)

Reduced Balancing / Deflated PCG Method

Solve iteratively:

\[P^T M^{-1} Ax = P^T M^{-1} b\]

with starting vector \(x_0 = P^T \tilde{x} + Qb\)

---

Two-Level PCG Methods

Standard Two-Level PCG Methods

Adapted Deflation Method

Instead of the reduced balancing / deflated PCG method with

\[ P^T M^{-1} Ax = P^T M^{-1} b \]

we can also solve its stabilized version

\[ (P^T M^{-1} + Q) Ax = (P^T M^{-1} + Q) b \]

Remarks

- Adapted deflation method can be derived from both deflation and domain decomposition
- Adapted deflation method is also a multigrid method!
- \( P \) follows from

\[ (I - PA) = (I - M^{-1} A) P^T \]

so that \( P = P^T M^{-1} + Q \) is also a multigrid V(1,0)-cycle preconditioner
Adapted Deflation Method

Instead of the reduced balancing / deflated PCG method with

\[ P^T M^{-1} Ax = P^T M^{-1} b \]

we can also solve its stabilized version

\[ (P^T M^{-1} + Q) Ax = (P^T M^{-1} + Q) b \]

Remarks

- Adapted deflation method can be derived from both deflation and domain decomposition
- Adapted deflation method is also a multigrid method!
- \( \mathcal{P} \) follows from

\[ (I - \mathcal{P}A) = (I - M^{-1}A)P^T \]

so that \( \mathcal{P} = P^T M^{-1} + Q \) is also a multigrid V(1,0)-cycle preconditioner
Adapted Deflation Method

Instead of the reduced balancing / deflated PCG method with

\[ P^T M^{-1} Ax = P^T M^{-1} b \]

we can also solve its stabilized version

\[ (P^T M^{-1} + Q) Ax = (P^T M^{-1} + Q) b \]

Remarks

- Adapted deflation method can be derived from both deflation and domain decomposition
- Adapted deflation method is also a multigrid method!
- \( P \) follows from

\[ (I - P A) = (I - M^{-1} A) P^T \]

so that \( P = P^T M^{-1} + Q \) is also a multigrid V(1,0)-cycle preconditioner
Standard Two-Level PCG Methods

Multigrid V(1,1)-Cycle Method

- Solve $\mathcal{P}$ from

\[(I - \mathcal{P}A) = (I - M^{-1}A)P^T(I - M^{-1}A)\]

where $M^{-1}$ is a preconditioner that can even be nonsymmetric

- The resulting multigrid V(1,1)-cycle preconditioner is

\[\mathcal{P} = M^{-1}P + P^TM^{-1} + Q - M^{-1}PAM^{-1}\]
Two-Level PCG Methods

Standard Two-Level PCG Methods

Multigrid V(1,1)-Cycle Method

- Solve $P$ from

$$(I - PA) = (I - M^{-1}A)P^T(I - M^{-1}A)$$

where $M^{-1}$ is a preconditioner that can even be nonsymmetric.

- The resulting multigrid V(1,1)-cycle preconditioner is

$$P = M^{-1}P + P^TM^{-1} + Q - M^{-1}PAM^{-1}$$
General Two-Level PCG Methods

Solve iteratively:

$$\mathcal{P} Ax = \mathcal{P} b$$

where $\mathcal{P}$ is a two-level preconditioner based on $M^{-1}$, $P$ and $Q$

Idea of Two-Level Preconditioner

$\mathcal{P}$ gets rid of both small and large eigenvalues of $A$
## General Two-Level PCG Methods

### Possible Choices for $\mathcal{P}$

<table>
<thead>
<tr>
<th>Name</th>
<th>Method</th>
<th>Operator $\mathcal{P}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>PCG</td>
<td>Traditional PCG</td>
<td>$M^{-1}$</td>
</tr>
<tr>
<td>AD</td>
<td>Additive CGC</td>
<td>$M^{-1} + Q$</td>
</tr>
<tr>
<td>DEF1</td>
<td>Deflated PCG 1</td>
<td>$M^{-1}P$</td>
</tr>
<tr>
<td>DEF2</td>
<td>Deflated PCG 2</td>
<td>$P^T M^{-1}$</td>
</tr>
<tr>
<td>BNN</td>
<td>Abstract Balancing</td>
<td>$P^T M^{-1}P + Q$</td>
</tr>
<tr>
<td>R-BNN1</td>
<td>Reduced Balancing 1</td>
<td>$P^T M^{-1}P$</td>
</tr>
<tr>
<td>R-BNN2</td>
<td>Reduced Balancing 2</td>
<td>$P^T M^{-1}$</td>
</tr>
<tr>
<td>A-DEF1</td>
<td>Adapted Deflated PCG 1</td>
<td>$M^{-1}P + Q$</td>
</tr>
<tr>
<td>A-DEF2</td>
<td>Adapted Deflated PCG 2</td>
<td>$P^T M^{-1} + Q$</td>
</tr>
<tr>
<td>MG</td>
<td>Multigrid V(1,1)-Cycle</td>
<td>$M^{-1}P + P^T M^{-1} + Q - M^{-1} P A M^{-1}$</td>
</tr>
</tbody>
</table>
Two-Level PCG Methods

Generalized Two-Level PCG Method

**Algorithm**

1: $x_0 := V_{\text{start}}$, $r_0 := b - Ax_0$, $y_0 := M_1 r_0$, $p_0 := M_2 y_0$
2: for $j := 0, 1, \ldots$, until convergence do
3: $w_j := M_3 A p_j$
4: $\alpha_j := \frac{(r_j, y_j)}{(p_j, w_j)}$
5: $x_{j+1} := x_j + \alpha_j p_j$
6: $r_{j+1} := r_j - \alpha_j w_j$
7: $y_{j+1} := M_1 r_{j+1}$
8: $\beta_j := \frac{(r_{j+1}, y_{j+1})}{(r_j, y_j)}$
9: $p_{j+1} := M_2 y_{j+1} + \beta_j p_j$
10: end for
11: $x_{\text{it}} := V_{\text{end}}$
## Two-Level PCG Methods

### Generalized Two-Level PCG Method

#### Parameters in Algorithm

<table>
<thead>
<tr>
<th>Method</th>
<th>$\mathcal{V}_{\text{start}}$</th>
<th>$\mathcal{M}_1$</th>
<th>$\mathcal{M}_2$</th>
<th>$\mathcal{M}_3$</th>
<th>$\mathcal{V}_{\text{end}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>PREC AD</td>
<td>$\bar{x}$</td>
<td>$M^{-1}$</td>
<td>$I$</td>
<td>$I$</td>
<td>$x_{j+1}$</td>
</tr>
<tr>
<td></td>
<td>$\bar{x}$</td>
<td>$M^{-1} + Q$</td>
<td></td>
<td>$I$</td>
<td>$x_{j+1}$</td>
</tr>
<tr>
<td>DEF1 DEF2</td>
<td>$\bar{x}$</td>
<td>$M^{-1}$</td>
<td>$I$</td>
<td>$P$</td>
<td>$Qb + P^T x_{j+1}$</td>
</tr>
<tr>
<td></td>
<td>$Qb + P^T \bar{x}$</td>
<td>$M^{-1}$</td>
<td></td>
<td>$P^T$</td>
<td>$x_{j+1}$</td>
</tr>
<tr>
<td>BNN</td>
<td>$\bar{x}$</td>
<td>$P^T M^{-1} P + Q$</td>
<td>$I$</td>
<td>$I$</td>
<td>$x_{j+1}$</td>
</tr>
<tr>
<td>R-BNN1</td>
<td>$Qb + P^T \bar{x}$</td>
<td>$P^T M^{-1} P$</td>
<td></td>
<td>$I$</td>
<td>$x_{j+1}$</td>
</tr>
<tr>
<td>R-BNN2</td>
<td>$Qb + P^T \bar{x}$</td>
<td>$P^T M^{-1}$</td>
<td></td>
<td>$I$</td>
<td>$x_{j+1}$</td>
</tr>
<tr>
<td>A-DEF1</td>
<td>$\bar{x}$</td>
<td>$M^{-1} P + Q$</td>
<td>$I$</td>
<td>$I$</td>
<td>$x_{j+1}$</td>
</tr>
<tr>
<td>A-DEF2</td>
<td>$Qb + P^T \bar{x}$</td>
<td>$P^T M^{-1} + Q$</td>
<td></td>
<td>$I$</td>
<td>$x_{j+1}$</td>
</tr>
<tr>
<td>MG</td>
<td>$\bar{x}$</td>
<td>$M^{-1} P + P^T M^{-1} + Q - M^{-1} P A M^{-1}$</td>
<td>$I$</td>
<td>$I$</td>
<td>$x_{j+1}$</td>
</tr>
</tbody>
</table>
Comparisons

Different comparisons possible:

- Typical parameters in the two-level preconditioners
- Arbitrary (but fixed) parameters in the two-level preconditioners

Example

Comparison: For each method its optimized set of parameters can be taken

- DEF1:
  - approximated eigenvectors as columns of $Z$
  - incomplete Cholesky preconditioner for $M^{-1}$
  - direct solution of $Ex = y$

- MG:
  - standard interpolation operator for $Z$
  - Gauss-Seidel smoother for $M^{-1}$
  - recursive solution of $Ex = y$
Comparisons

Different comparisons possible:

- **Typical** parameters in the two-level preconditioners
- **Arbitrary** (but fixed) parameters in the two-level preconditioners

Example

**Comparison**: For each method its optimized set of parameters can be taken

- **DEF1**:
  - approximated eigenvectors as columns of $Z$
  - incomplete Cholesky preconditioner for $M^{-1}$
  - direct solution of $Ex = y$

- **MG**:
  - standard interpolation operator for $Z$
  - Gauss-Seidel smoother for $M^{-1}$
  - recursive solution of $Ex = y$
Comparison of Two-Level PCG Methods

Comparisons

Different comparisons possible:
- Typical parameters in the two-level preconditioners
- Arbitrary (but fixed) parameters in the two-level preconditioners

Example

Comparison: For each method it is allowed to take
- a block-Jacobi preconditioner $M^{-1}$ (domain decomposition)
- approximated eigenvectors as columns of $Z$ (deflation)
- recursive solution of $Ex = y$ (multigrid)
Comparisons

Different comparisons possible:
- Typical parameters in the two-level preconditioners
- Arbitrary (but fixed) parameters in the two-level preconditioners

Example

**Comparison**: For each method it is allowed to take
- a block-Jacobi preconditioner $M^{-1}$ (*domain decomposition*)
- approximated eigenvectors as columns of $Z$ (*deflation*)
- recursive solution of $Ex = y$ (*multigrid*)
Introduction

Two-Level PCG Methods

Comparison of Two-Level PCG Methods

Conclusions
Previous Comparisons

Previous Works
Comparisons of DEF1, AD and BNN have already been performed \(^1\ 2\ 3\)

Main Result
In exact arithmetic, DEF1 performs better than both BNN and AD

---

Comparison of Two-Level PCG Methods

Spectral Analysis

Theorem

AD has a worse condition number compared to the other two-level PCG methods

Theorem

Class 1 (DEF1, DEF2, R-BNN1 and R-BNN2):

\[
\sigma(M^{-1}PA) = \sigma(P^TM^{-1}A) = \sigma(P^TM^{-1}PA) = \{0, 0, \ldots, 0, \lambda_{k+1}, \ldots, \lambda_n\}
\]

Theorem

Class 2 (BNN, A-DEF1, A-DEF2):

\[
\sigma((P^TM^{-1}P + Q)A) = \sigma((M^{-1}P + Q)A) = \sigma(P^TM^{-1} + QA) = \{1, 1, \ldots, 1, \mu_{k+1}, \ldots, \mu_n\}
\]
Spectral Analysis

Theorem
AD has a worse condition number compared to the other two-level PCG methods

Theorem
Class 1 (DEF1, DEF2, R-BNN1 and R-BNN2):
\[ \sigma \left( M^{-1} PA \right) = \sigma \left( P^T M^{-1} A \right) = \sigma \left( P^T M^{-1} PA \right) = \{0, 0, \ldots, 0, \lambda_{k+1}, \ldots, \lambda_n\} \]

Theorem
Class 2 (BNN, A-DEF1, A-DEF2):
\[ \sigma \left( (P^T M^{-1} P + Q)A \right) = \sigma \left( (M^{-1} P + Q)A \right) = \sigma \left( (P^T M^{-1} + Q)A \right) = \{1, 1, \ldots, 1, \mu_{k+1}, \ldots, \mu_n\} \]
Main Results

Theorem

Spectrum of DEF1, DEF2, R-BNN1 or R-BNN2:

\[ \sigma = \{0, \ldots, 0, \lambda_{k+1}, \ldots, \lambda_n\} \]

Spectrum of BNN, A-DEF1 or A-DEF2:

\[ \sigma = \{1, \ldots, 1, \mu_{k+1}, \ldots, \mu_n\} \]

Then:

\[ \lambda_{k+1} = \mu_{k+1}, \ldots, \lambda_n = \mu_n \]

Theorem

Let \( \bar{x} \in \mathbb{R}^n \) be arbitrary. The following methods produce exactly the same iterates:

- BNN with \( \nu_{\text{start}} = Qb + P^T \bar{x} \)
- DEF2, A-DEF2, R-BNN1 and R-BNN2 (with \( \nu_{\text{start}} = Qb + P^T \bar{x} \))
- DEF1 (with \( \nu_{\text{start}} = \bar{x} \)) based on \( x_{j+1} = Qb + P^T x_{j+1} \)
Comparison of Two-Level PCG Methods

Main Results

**Theorem**

Spectrum of DEF1, DEF2, R-BNN1 or R-BNN2:

\[ \sigma = \{0, \ldots, 0, \lambda_{k+1}, \ldots, \lambda_n\} \]

Spectrum of BNN, A-DEF1 or A-DEF2:

\[ \sigma = \{1, \ldots, 1, \mu_{k+1}, \ldots, \mu_n\} \]

Then:

\[ \lambda_{k+1} = \mu_{k+1}, \ldots, \lambda_n = \mu_n \]

**Theorem**

Let \( \bar{x} \in \mathbb{R}^n \) be arbitrary. The following methods produce exactly the same iterates:

- BNN with \( \mathcal{V}_{\text{start}} = Qb + P^T \bar{x} \)
- DEF2, A-DEF2, R-BNN1 and R-BNN2 (with \( \mathcal{V}_{\text{start}} = Qb + P^T \bar{x} \))
- DEF1 (with \( \mathcal{V}_{\text{start}} = \bar{x} \)) based on \( x_{j+1} = Qb + P^T x_{j+1} \)
Numerical Experiment

Typical Convergence Behavior

2D bubbly flow problem; Poisson equation with a discontinuous coefficient; contrast $\epsilon = 10^3$, finite differences on a uniform grid, $Ax = b$ with $n = 62^2$ and $k = 64^2$. 
Consequences

Best Method

- All methods (except AD and A-DEF1) have approximately the same convergence behavior
- DEF1 ($\mathcal{P} = M^{-1}P$), DEF2 ($\mathcal{P} = P^TM^{-1}$) and R-BNN2 ($\mathcal{P} = P^TM^{-1}$) have the lowest cost per iteration

Most Robust Method?

- Compare methods with respect to perturbed starting vector
- Compare methods with respect to severe termination criterion
- Compare methods with respect to inaccurate $E^{-1}$
Comparison of Two-Level PCG Methods

Consequences

Best Method

- All methods (except AD and A-DEF1) have approximately the same convergence behavior
- DEF1 ($\mathcal{P} = M^{-1}P$), DEF2 ($\mathcal{P} = P^TM^{-1}$) and R-BNN2 ($\mathcal{P} = P^TM^{-1}$) have the lowest cost per iteration

Most Robust Method?

Compare methods with respect to
- perturbed starting vector
- severe termination criterion
- inaccurate $E^{-1}$
Theoretical Comparison

Perturbating $E^{-1}$ by a Small Parameter $\epsilon$

- Class 1 (DEF1, DEF2, R-BNN1 and R-BNN2):
  \[ \sigma \approx \{ O(\epsilon), \ldots, O(\epsilon), \lambda_{k+1}, \ldots, \lambda_n \} \]

- Class 2 (BNN, A-DEF1, A-DEF2):
  \[ \sigma \approx \{ 1, 1, \ldots, 1, \mu_{k+1}, \ldots, \mu_n \} \]

Consequence
- Class 1 (DEF1, DEF2, R-BNN1 and R-BNN2) is not robust
- Class 2 (BNN, A-DEF1, A-DEF2) is robust
Comparison of Two-Level PCG Methods

Theoretical Comparison

Perturbing $E^{-1}$ by a Small Parameter $\epsilon$

- Class 1 (DEF1, DEF2, R-BNN1 and R-BNN2):
  \[
  \sigma \approx \{O(\epsilon), \ldots, O(\epsilon), \lambda_{k+1}, \ldots, \lambda_{n}\}
  \]

- Class 2 (BNN, A-DEF1, A-DEF2):
  \[
  \sigma \approx \{1, 1, \ldots, 1, \mu_{k+1}, \ldots, \mu_{n}\}
  \]

Consequence

- Class 1 (DEF1, DEF2, R-BNN1 and R-BNN2) is not robust
- Class 2 (BNN, A-DEF1, A-DEF2) is robust
2D bubbly flow problem; Poisson equation with a discontinuous coefficient; contrast $\epsilon = 10^3$, finite differences on a uniform grid, $Ax = b$ with $n = 62^2$ and $k = 64^2$
Comparison of Two-Level PCG Methods

Typical Robustness Experiments

Convergence Behavior (larger perturbation in $E^{-1}$)

2D bubbly flow problem; Poisson equation with a discontinuous coefficient; contrast $\epsilon = 10^3$, finite differences on a uniform grid, $Ax = b$ with $n = 62^2$ and $k = 64^2$
Consequence

- Class 1 (DEF1, DEF2, R-BNN1 and R-BNN2) is not robust
- Class 2 (BNN, A-DEF1, A-DEF2) is robust
Recall

- Multigrid V(1,1)-cycle (MG) preconditioner:
  \[ P = M^{-1}P + P^T M^{-1} + Q - M^{-1} P A M^{-1} \]

- Deflation (DEF1) preconditioner:
  \[ P = M^{-1}P \]

Main Question

Is MG more effective than DEF1?

Answer

MG is often more effective than DEF1
But not always!
Multigrid V(1,1)-Cycle versus Deflation

Recall

- Multigrid V(1,1)-cycle (MG) preconditioner:
  \[ P = M^{-1} P + P^T M^{-1} + Q - M^{-1} P A M^{-1} \]

- Deflation (DEF1) preconditioner:
  \[ P = M^{-1} P \]

Main Question

Is MG more effective than DEF1?

Answer

MG is often more effective than DEF1
But not always!
Recall

- Multigrid V(1,1)-cycle (MG) preconditioner:
  \[ P = M^{-1} P + P^T M^{-1} + Q - M^{-1} P A M^{-1} \]

- Deflation (DEF1) preconditioner:
  \[ P = M^{-1} P \]

Main Question

Is MG more effective than DEF1?

Answer

MG is often more effective than DEF1

But not always!
Recall

- Multigrid V(1,1)-cycle (MG) preconditioner:
  \[ P = M^{-1} P + P^T M^{-1} + Q - M^{-1} P A M^{-1} \]

- Deflation (DEF1) preconditioner:
  \[ P = M^{-1} P \]

Main Question

Is MG more effective than DEF1?

Answer

MG is often more effective than DEF1
But not always!
Comparison of Two-Level PCG Methods

Multigrid $V(1,1)$-Cycle versus Deflation

Example

- $M^{-1}A = \text{diag}(1, 1.25, 1.5, 1.75)$
- $Z = [v_1 \ v_2]$ with $v_1$ and $v_2$ to be eigenvectors corresponding to the two smallest eigenvalues of $M^{-1}A$

Then, the spectra are given by

$$\sigma_{MG} = \{0.4375, 0.75, 1, 1\}, \quad \sigma_{DEF1} = \{0, 0, 1.5, 1.75\}$$

resulting in

$$\kappa_{MG} = 2.2857 > 1.1667 = \kappa_{DEF1}!$$
Multigrid V(1,1)-Cycle versus Deflation

Example

- \( M^{-1}A = \text{diag}(1, 1.25, 1.5, 1.75) \)
- \( Z = [v_1 \ v_2] \) with \( v_1 \) and \( v_2 \) to be eigenvectors corresponding to the two smallest eigenvalues of \( M^{-1}A \)

Then, the spectra are given by

\[
\sigma_{\text{MG}} = \{0.4375, 0.75, 1, 1\}, \quad \sigma_{\text{DEF1}} = \{0, 0, 1.5, 1.75\}
\]

resulting in

\[
\kappa_{\text{MG}} = 2.2857 > 1.1667 = \kappa_{\text{DEF1}}!
\]
Comparison of $\kappa_{MG}$ and $\kappa_{DEF1}$

Figure: $Z$ consists of eigenvectors corresponding to the smallest eigenvalues of $M^{-1}A$ where $M^{-1}$ is arbitrary. $\kappa_{MG} < \kappa_{DEF1}$ holds in Regions $A_1$ and $A_2$, while $\kappa_{DEF1} < \kappa_{MG}$ holds in Regions $B_1$ and $B_2$. 

\[
\lambda_{k+1}(M^{-1}A) = \lambda_n
\]

\[
\lambda_{k+1}(M^{-1}A) = 2 - \lambda_n
\]

\[
\lambda_{k+1}(M^{-1}A) = 2 - (\lambda_n)^{-1}
\]

\[
\lambda_{k+1}(M^{-1}A) = \lambda_n^2(2 - \lambda_n)
\]
Observations

- DEF1 can be more effective than MG in some cases.
- For ‘effective’ $M^{-1}$, MG is usually faster and more robust but also more expensive.
- It is possible to make each iteration of DEF1 as expensive as MG, while DEF1 is faster than MG.
Observations

- DEF1 can be more effective than MG in some cases.
- For ‘effective’ $M^{-1}$, MG is usually faster and more robust but also more expensive.
- It is possible to make each iteration of DEF1 as expensive as MG, while DEF1 is faster than MG.
1 Introduction

2 Two-Level PCG Methods

3 Comparison of Two-Level PCG Methods

4 Conclusions
Conclusions

- The connection between different worlds

Two-Level Preconditioning

- Deflation
- Multigrid
- Domain Decomposition

Connection?
Conclusions

Lessons

- The connection between different worlds is surprisingly much stronger

Two-Level Preconditioning

- Deflation
- Multigrid
- Domain Decomposition

References:

Conclusions

Main Conclusions

Lessons

- Some reduced forms of two-level PCG methods are not robust \(^a\) \(^b\)
- Some equivalent methods have different robustness properties \(^c\) \(^d\)
- The optimal two-level PCG method depends on many aspects \(^e\) \(^f\) \(^g\) \(^h\)

\(^h\) Y.A. ERLANGGA, R. NABBEN, SIAM Journal on Scientific Computing, 30, 1572–1595, 2008
Conclusions

Main Conclusions

Lessons

- Some reduced forms of two-level PCG methods are not robust \(^a\ b\)
- Some equivalent methods have different robustness properties \(^c\ d\)
- The optimal two-level PCG method depends on many aspects \(^e\ f\ g\ h\)

Main Conclusions

Lessons

- Some reduced forms of two-level PCG methods are not robust \(^a\) \(^b\)
- Some equivalent methods have different robustness properties \(^c\) \(^d\)
- The optimal two-level PCG method depends on many aspects \(^e\) \(^f\) \(^g\) \(^h\)