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 $\kappa = \frac{\lambda_{\max}}{\lambda_{\min}}$ is large)

TU Delft

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

[1] M.R. Hestenes and E. Stiefel, J. Research Nat. Bur. Standards, **49**, 409–436, 1952.

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

[1] M.R. Hestenes and E. Stiefel, J. Research Nat. Bur. Standards, **49**, 409–436, 1952.

Two-Level PCG Methods

Two-Level PCG Method (Two-Level PCG)

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

TU Delft

Two-Level PCG Methods

Two-Level PCG Method (Two-Level PCG)

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

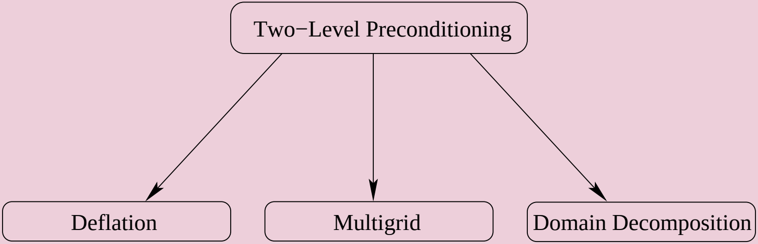
Consequence

Faster convergence of the iterative process

TU Delft

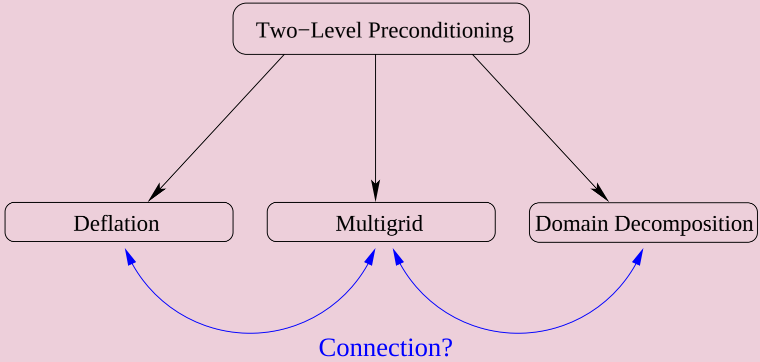
Two-Level Preconditioning

World of Two-Level Preconditioners



Two-Level Preconditioning

World of Two-Level Preconditioners



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?

TU Delft

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?

TU Delft

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

TU Delft

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 $\rightarrow$ 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)

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 $\rightarrow$ 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)

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

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

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

	Deflation	Multigrid	DDM
M^{-1}	good preconditioner	smoother	subdomain solves
P	deflation matrix	coarse-grid correction	coarse-grid correction
Z	deflation-subspace	interpolation	interpolation
Z^T	deflation-subspace	restriction	restriction
k	$k \ll n$	$1 \ll k$	$1 \ll k \ll n$
$Ex = y$	direct	recursive	direct / iterative

TU Delft

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

	Deflation	Multigrid	DDM
M^{-1}	good preconditioner	smoother	subdomain solves
P	deflation matrix	coarse-grid correction	coarse-grid correction
Z	deflation-subspace	interpolation	interpolation
Z^T	deflation-subspace	restriction	restriction
k	$k \ll n$	$1 \ll k$	$1 \ll k \ll n$
$Ex = y$	direct	recursive	direct / iterative

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:

$$(P^T M^{-1} P + Q)Ax = (P^T M^{-1} P + Q)b$$

[1] R.A. Nicolaides, SIAM J. Matrix Anal. Appl., **24**, 355–365, 1987.
[2] Z. Dostal, Int. J. Computer Math., **23**, 315–323, 1988.
[3] C. Vuik, A. Segal and J.A. Meijerink, J. Comp. Phys., **152**, 385–403, 1999.
[4] J.H. Bramble, J.E. Pasciak and A.H. Schatz, I. Math. Comp., **47**, 103–134, 1986.
[5] J. Mandel, Commun. Appl. Numer. Meth., **9**, 233–241, 1993.

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:

$$(P^T M^{-1} P + Q)Ax = (P^T M^{-1} P + Q)b$$

[1] R.A. Nicolaides, SIAM J. Matrix Anal. Appl., **24**, 355–365, 1987.
[2] Z. Dostal, Int. J. Computer Math., **23**, 315–323, 1988.
[3] C. Vuik, A. Segal and J.A. Meijerink, J. Comp. Phys., **152**, 385–403, 1999.
[4] J.H. Bramble, J.E. Pasciak and A.H. Schatz, I. Math. Comp., **47**, 103–134, 1986.
[5] J. Mandel, Commun. Appl. Numer. Meth., **9**, 233–241, 1993.

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:

$$(P^T M^{-1} P + Q)Ax = (P^T M^{-1} P + Q)b$$

[1] R.A. Nicolaides, SIAM J. Matrix Anal. Appl., **24**, 355–365, 1987.

[2] Z. Dostal, Int. J. Computer Math., **23**, 315–323, 1988.

[3] C. Vuik, A. Segal and J.A. Meijerink, J. Comp. Phys., **152**, 385–403, 1999.

[4] J.H. Bramble, J.E. Pasciak and A.H. Schatz, I. Math. Comp., **47**, 103–134, 1986.

[5] J. Mandel, Commun. Appl. Numer. Meth., **9**, 233–241, 1993.

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 [1] [2]

Solve iteratively:

$$P^T M^{-1} Ax = P^T M^{-1} b$$

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

[1] A. Toselli and O.B. Widlund, Comp. Math.., **34**, Springer, Berlin, 2005.

[2] Y. Saad, M. Yeung, J. Erhel and F. Guyomarc'h, SIAM J. Sci. Comput., **21**, 1909–1926, 2000.

TUDelft

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 [1] [2]

Solve iteratively:

$$P^T M^{-1} Ax = P^T M^{-1} b$$

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

[1] A. Toselli and O.B. Widlund, Comp. Math.., **34**, Springer, Berlin, 2005.

[2] Y. Saad, M. Yeung, J. Erhel and F. Guyomarc'h, SIAM J. Sci. Comput., **21**, 1909–1926, 2000.

Standard Two-Level PCG Methods

Adapted Deflation Method

Instead of the reduced balancing / deflated PCG method with

$$P^T M^{-1} A x = P^T M^{-1} b$$

we can also solve its stabilized version

$$(P^T M^{-1} + Q) A x = (P^T M^{-1} + Q) b$$

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

TUDelft

Standard Two-Level PCG Methods

Adapted Deflation Method

Instead of the reduced balancing / deflated PCG method with

$$P^T M^{-1} A x = P^T M^{-1} b$$

we can also solve its stabilized version

$$(P^T M^{-1} + Q) A x = (P^T M^{-1} + Q) b$$

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

TU Delft

Standard Two-Level PCG Methods

Adapted Deflation Method

Instead of the reduced balancing / deflated PCG method with

$$P^T M^{-1} A x = P^T M^{-1} b$$

we can also solve its stabilized version

$$(P^T M^{-1} + Q) A x = (P^T M^{-1} + Q) b$$

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

TUDelft

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
$$\mathcal{P} = M^{-1}P + P^TM^{-1} + Q - M^{-1}PAM^{-1}$$

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
$$\mathcal{P} = M^{-1}P + P^T M^{-1} + Q - M^{-1}PAM^{-1}$$

TU Delft

General Two-Level PCG Methods

General Two-Level PCG

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 $\mathcal{P}$

Name	Method	Operator $\mathcal{P}$
PCG	Traditional PCG	M^{-1}
AD	Additive CGC	$M^{-1} + Q$
DEF1	Deflated PCG 1	$M^{-1}P$
DEF2	Deflated PCG 2	$P^T M^{-1}$
BNN	Abstract Balancing	$P^T M^{-1} P + Q$
R-BNN1	Reduced Balancing 1	$P^T M^{-1} P$
R-BNN2	Reduced Balancing 2	$P^T M^{-1}$
A-DEF1	Adapted Deflated PCG 1	$M^{-1}P + Q$
A-DEF2	Adapted Deflated PCG 2	$P^T M^{-1} + Q$
MG	Multigrid V(1,1)-Cycle	$M^{-1}P + P^T M^{-1} + Q - M^{-1}PAM^{-1}$

Generalized Two-Level PCG Method

Algorithm

$x_0 := \mathcal{V}_{\text{start}},\ r_0 := b - Ax_0,\ y_0 := \mathcal{M}_1 r_0,\ p_0 := \mathcal{M}_2 y_0$
for $j := 0, 1, \ldots,$ until convergence **do**
  $w_j := \mathcal{M}_3 A p_j$
  $\alpha_j := \frac{(r_j, y_j)}{(p_j, w_j)}$
  $x_{j+1} := x_j + \alpha_j p_j$
  $r_{j+1} := r_j - \alpha_j w_j$
  $y_{j+1} := \mathcal{M}_1 r_{j+1}$
  $\beta_j := \frac{(r_{j+1}, y_{j+1})}{(r_j, y_j)}$
  $p_{j+1} := \mathcal{M}_2 y_{j+1} + \beta_j p_j$
end for
$x_{\text{it}} := \mathcal{V}_{\text{end}}$

TU Delft

Generalized Two-Level PCG Method

Parameters in Algorithm

Method	$\mathcal{V}_{\text{start}}$	$\mathcal{M}_1$	$\mathcal{M}_2$	$\mathcal{M}_3$	$\mathcal{V}_{\text{end}}$
PREC	$\bar{x}$	M^{-1}	I	I	x_{j+1}
AD	$\bar{x}$	$M^{-1} + Q$	I	I	x_{j+1}
DEF1	$\bar{x}$	M^{-1}	I	P	$Qb + P^T x_{j+1}$
DEF2	$Qb + P^T \bar{x}$	M^{-1}	P^T	I	x_{j+1}
BNN	$\bar{x}$	$P^T M^{-1} P + Q$	I	I	x_{j+1}
R-BNN1	$Qb + P^T \bar{x}$	$P^T M^{-1} P$	I	I	x_{j+1}
R-BNN2	$Qb + P^T \bar{x}$	$P^T M^{-1}$	I	I	x_{j+1}
A-DEF1	$\bar{x}$	$M^{-1} P + Q$	I	I	x_{j+1}
A-DEF2	$Qb + P^T \bar{x}$	$P^T M^{-1} + Q$	I	I	x_{j+1}
MG	$\bar{x}$	$M^{-1} P + P^T M^{-1} + Q - M^{-1} P A M^{-1}$	I	I	x_{j+1}

TU Delft

Comparison of Two-Level PCG Methods

Comparisons

Different comparisons possible:

- Typical parameters in the two-level preconditioners [1]
- 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 $Ex = y$

[1] S.P. MacLachlan, J.M. Tang and C. Vuik, Journal of Computational Physics, **227**, 9742–9761, 2008.

Comparison of Two-Level PCG Methods

Comparisons

Different comparisons possible:

- Typical parameters in the two-level preconditioners [1]
- 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 $Ex = y$

[1] S.P. MacLachlan, J.M. Tang and C. Vuik, Journal of Computational Physics, **227**, 9742–9761, 2008.

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)

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)

TU Delft

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

[1] R. Nabben and C. Vuik, SIAM J. Numer. Anal., **42**, 1631–1647, 2004.
[2] R. Nabben and C. Vuik, SIAM J. Sci. Comput., **27**, 1742–1759, 2006.
[3] R. Nabben and C. Vuik, Num. Lin. Alg. Appl., **15**, 355-372, 2008

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):

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

Theorem

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

$$\begin{aligned}\sigma\left((P^TM^{-1}P+Q)A\right) = \sigma\left((M^{-1}P+Q)A\right) &= \sigma\left((P^TM^{-1}+Q)A\right) \\ &= \{1, 1, \ldots, 1, \mu_{k+1}, \ldots, \mu_n\}\end{aligned}$$

TU Delft

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):

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

Theorem

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

$$\begin{aligned}\sigma\left((P^TM^{-1}P + Q)A\right) = \sigma\left((M^{-1}P + Q)A\right) &= \sigma\left((P^TM^{-1} + Q)A\right) \\ &= \{1, 1, \ldots, 1, \mu_{k+1}, \ldots, \mu_n\}\end{aligned}$$

TU Delft

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:

$$\lambda_{k+1} = \mu_{k+1}, \quad \ldots, \quad \lambda_n = \mu_n$$

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 $x_{j+1} = Qb + P^T x_{j+1}$

TU Delft

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:

$$\lambda_{k+1} = \mu_{k+1}, \quad \ldots, \quad \lambda_n = \mu_n$$

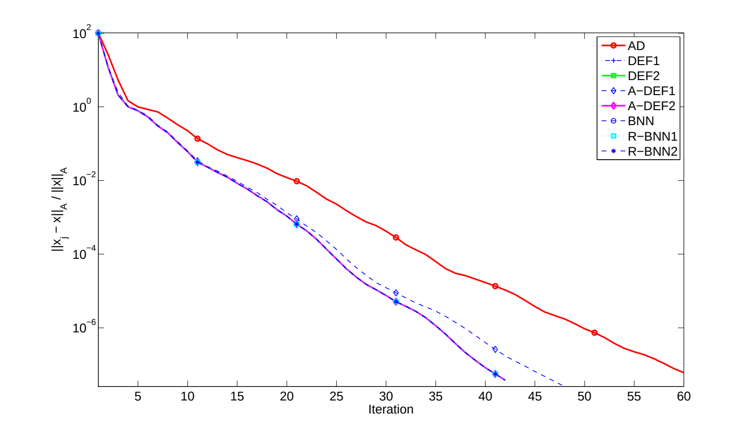
Theorem

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 $k = 64^2$

elft

Consequences

Best Method

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

Most Robust Method?

Compare methods with respect to

- perturbed starting vector
- severe termination criterion
- inaccurate E^{-1}

Consequences

Best Method

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

Most Robust Method?

Compare methods with respect to

- perturbed starting vector
- severe termination criterion
- inaccurate E^{-1}

TU Delft

Theoretical Comparison

Perturbating E^{-1} by a Small Parameter ϵ

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

$$\sigma \approx \{\mathcal{O}(\epsilon), \ldots, \mathcal{O}(\epsilon), \lambda_{k+1}, \ldots, \lambda_n\}$$

- 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

Theoretical Comparison

Perturbating E^{-1} by a Small Parameter ϵ

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

$$\sigma \approx \{\mathcal{O}(\epsilon), \ldots, \mathcal{O}(\epsilon), \lambda_{k+1}, \ldots, \lambda_n\}$$

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

$$\sigma \approx \{1, 1, \ldots, 1, \mu_{k+1}, \ldots, \mu_n\}$$

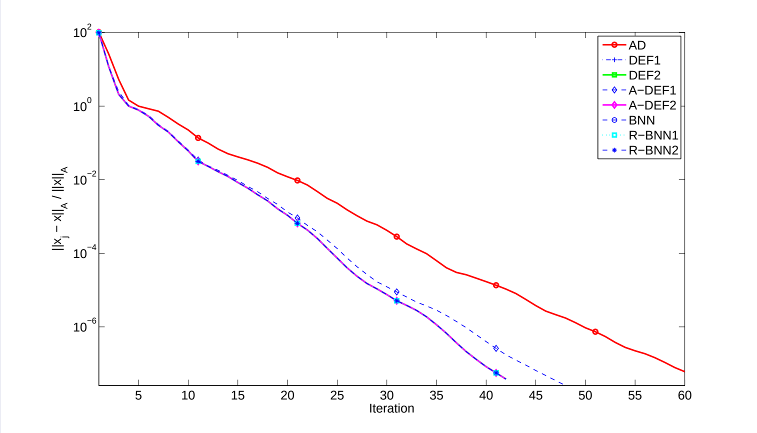
Consequence

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

TU Delft

Robustness Experiments

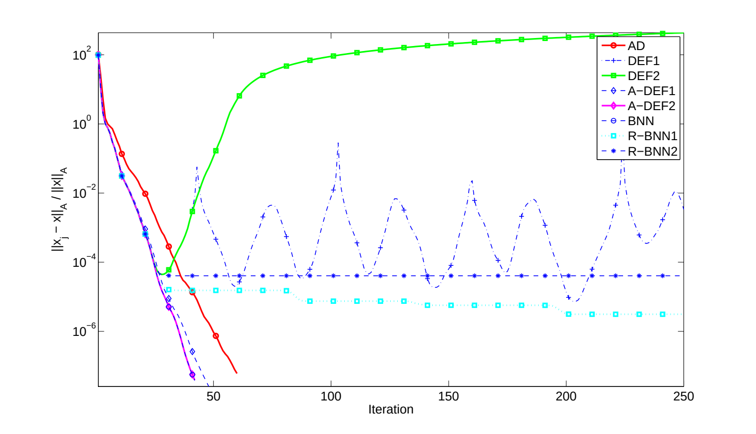
Typical Convergence Behavior (small perturbation in E^{-1})



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 $k = 64^2$

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 $k = 64^2$

elft

Theoretical Comparison

Consequence

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

Multigrid V(1,1)-Cycle versus Deflation

Recall

- Multigrid V(1,1)-cycle (MG) preconditioner:

$$\mathcal{P} = M^{-1}P + P^T M^{-1} + Q - M^{-1}PAM^{-1}$$

- Deflation (DEF1) preconditioner:

$$\mathcal{P} = M^{-1}P$$

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:

$$\mathcal{P} = M^{-1}P + P^T M^{-1} + Q - M^{-1}PAM^{-1}$$

- Deflation (DEF1) preconditioner:

$$\mathcal{P} = M^{-1}P$$

Main Question

Is MG more effective than DEF1?

Answer

MG is often more effective than DEF1
But not always!

TU Delft

Multigrid V(1,1)-Cycle versus Deflation

Recall

- Multigrid V(1,1)-cycle (MG) preconditioner:

$$\mathcal{P} = M^{-1}P + P^T M^{-1} + Q - M^{-1}PAM^{-1}$$

- Deflation (DEF1) preconditioner:

$$\mathcal{P} = M^{-1}P$$

Main Question

Is MG more effective than DEF1?

Answer

MG is often more effective than DEF1
But not always!

TU Delft

Multigrid V(1,1)-Cycle versus Deflation

Recall

- Multigrid V(1,1)-cycle (MG) preconditioner:

$$\mathcal{P} = M^{-1}P + P^T M^{-1} + Q - M^{-1}PAM^{-1}$$

- Deflation (DEF1) preconditioner:

$$\mathcal{P} = M^{-1}P$$

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

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

$$\kappa_{\text{MG}} = 2.2857 > 1.1667 = \kappa_{\text{DEF1}}!$$

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

$$\kappa_{\text{MG}} = 2.2857 > 1.1667 = \kappa_{\text{DEF1}}!$$

TU Delft

Comparison of MG and DEF1

Comparison of κ_{MG} and κ_{DEF1}

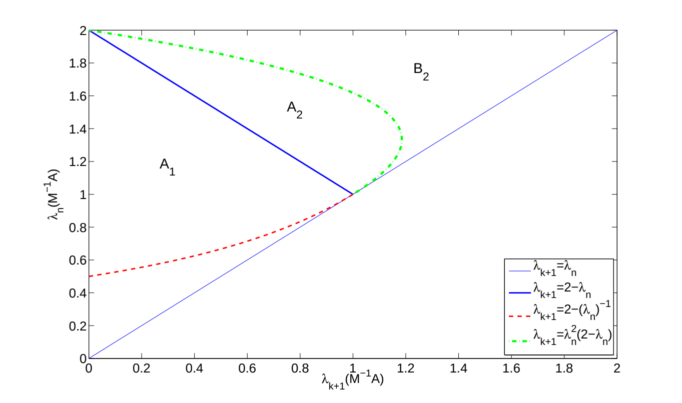


Figure: Z consists of eigenvectors corresponding to the smallest eigenvalues of $M^{-1}A$ where M^{-1} is arbitrary. $\kappa_{\mathrm{MG}} < \kappa_{\mathrm{DEF1}}$ holds in Regions A_1 and A_2, while $\kappa_{\mathrm{DEF1}} < \kappa_{\mathrm{MG}}$ holds in Regions B_1 and B_2.

Comparison of MG and DEF1

Observations [1]

- 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] J.M. Tang, S.P. MacLachlan, R. Nabben, C. Vuik, SIAM. J. Matrix Anal. and Appl., **31**, 1715–1739, 2010

TUDelft

Comparison of MG and DEF1

Observations [1]

- 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

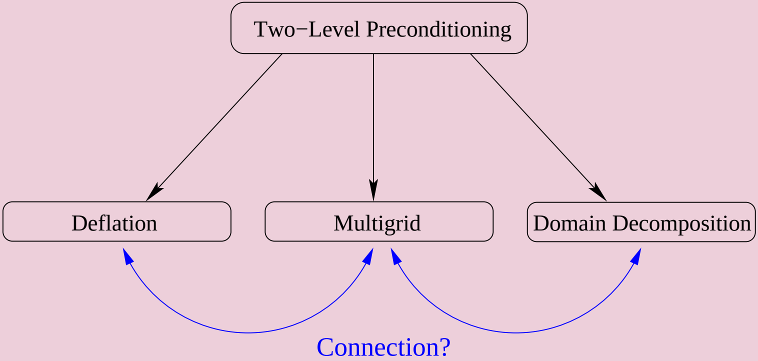
[1] J.M. TANG, S.P. MACLACHLAN, R. NABBEN, C. VUIK, SIAM. J. Matrix Anal. and Appl., **31**, 1715–1739, 2010

TUDelft

Conclusions

Lessons

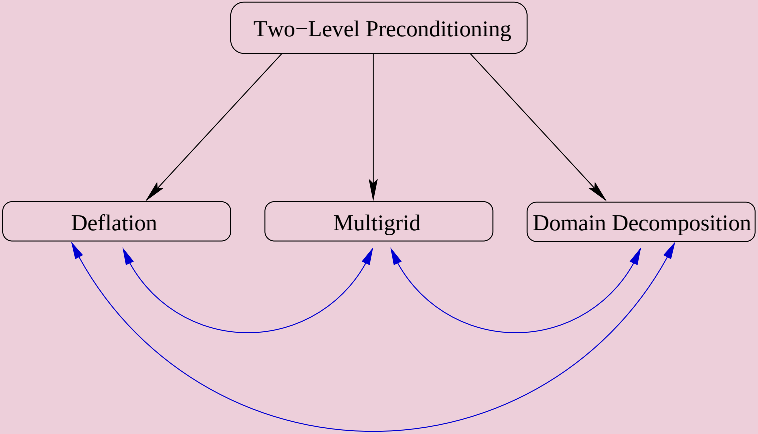
- The connection between different worlds



Conclusions

Lessons

- The connection between different worlds is surprisingly much stronger [1] [2]



[1] J.M. Tang, R. Nabben, C. Vuik and Y.A. Erlangga, Journal of Scientific Computing, **39**, 340–370, 2009

[2] J.M. Tang, S.P. MacLachlan, R. Nabben, C. Vuik, SIAM. J. Matrix Anal. and Appl., **31**, 1715–1739, 2010

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]

[a] A. Toselli and O.B. Widlund, Comp. Math., **34**, Springer, Berlin, 2005.
[b] J. Mandel, Commun. Appl. Numer. Meth., **9**, 233–241, 1993.
[c] Y. Saad, M. Yeung, J. Erhel and F. Guyomarc'h, SIAM J. Sci. Comput., **21**, 1909–1926, 2000.
[d] C. Vuik, A. Segal and J.A. Meijerink, J. Comp. Phys., **152**, 385–403, 1999.
[e] S.P. MacLachlan, J.M. Tang and C. Vuik, Journal of Computational Physics, **227**, 9742–9761, 2008.
[f] J.M. Tang, R. Nabben, C. Vuik and Y.A. Erlangga, Journal of Scientific Computing, **39**, 340–370, 2009
[g] J.M. Tang, S.P. MacLachlan, R. Nabben, C. Vuik, SIAM. J. Matrix Anal. and Appl., **31**, 1715–1739, 2010
[h] Y.A. Erlangga, R. Nabben, SIAM Journal on Scientific Computing, **30**, 1572–1595, 2008

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]

[a] A. Toselli and O.B. Widlund, Comp. Math., **34**, Springer, Berlin, 2005.
[b] J. Mandel, Commun. Appl. Numer. Meth., **9**, 233–241, 1993.
[c] Y. Saad, M. Yeung, J. Erhel and F. Guyomarc'h, SIAM J. Sci. Comput., **21**, 1909–1926, 2000.
[d] C. Vuik, A. Segal and J.A. Meijerink, J. Comp. Phys., **152**, 385–403, 1999.
[e] S.P. MacLachlan, J.M. Tang and C. Vuik, Journal of Computational Physics, **227**, 9742–9761, 2008.
[f] J.M. Tang, R. Nabben, C. Vuik and Y.A. Erlangga, Journal of Scientific Computing, **39**, 340–370, 2009
[g] J.M. Tang, S.P. MacLachlan, R. Nabben, C. Vuik, SIAM. J. Matrix Anal. and Appl., **31**, 1715–1739, 2010
[h] Y.A. Erlangga, R. Nabben, SIAM Journal on Scientific Computing, **30**, 1572–1595, 2008

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]

---

[a] A. Toselli and O.B. Widlund, Comp. Math., **34**, Springer, Berlin, 2005.
[b] J. Mandel, Commun. Appl. Numer. Meth., **9**, 233–241, 1993.
[c] Y. Saad, M. Yeung, J. Erhel and F. Guyomarc'h, SIAM J. Sci. Comput., **21**, 1909–1926, 2000.
[d] C. Vuik, A. Segal and J.A. Meijerink, J. Comp. Phys., **152**, 385–403, 1999.
[e] S.P. MacLachlan, J.M. Tang and C. Vuik, Journal of Computational Physics, **227**, 9742–9761, 2008.
[f] J.M. Tang, R. Nabben, C. Vuik and Y.A. Erlangga, Journal of Scientific Computing, **39**, 340–370, 2009
[g] J.M. Tang, S.P. MacLachlan, R. Nabben, C. Vuik, SIAM. J. Matrix Anal. and Appl., **31**, 1715–1739, 2010
[h] Y.A. Erlangga, R. Nabben, SIAM Journal on Scientific Computing, **30**, 1572–1595, 2008