Various ways to use a second level preconditioner

C. Vuik [1], J.M. Tang [1], R. Nabben [2], and Y. Erlangga [3]

[1] Delft University of Technology
Delft Institute of Applied Mathematics

[2] Technische Universität Berlin
Institut für Mathematik

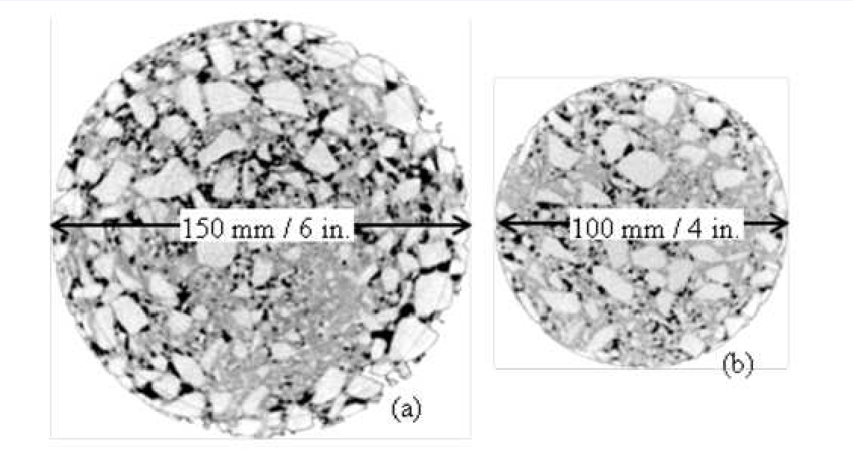
[3] University of British Columbia
Department of Earth and Ocean Sciences

SIAM CSE 2009
March 2-6, 2009

Outline

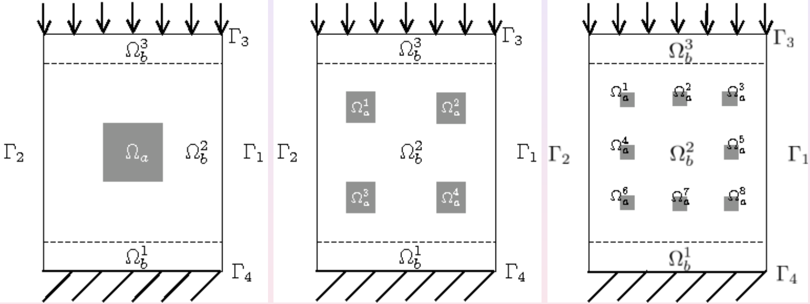
Tomography scans of bitumen species

Mechanical problem

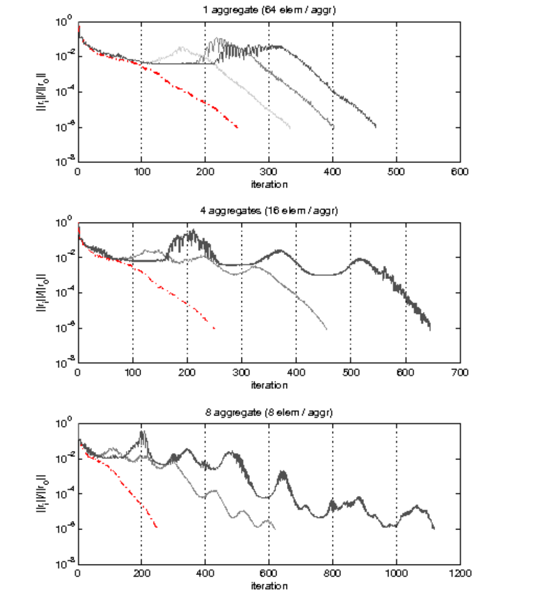


Collaboration with: Tom Jönsthövel, Martin van Gijzen and Tom Scarpas

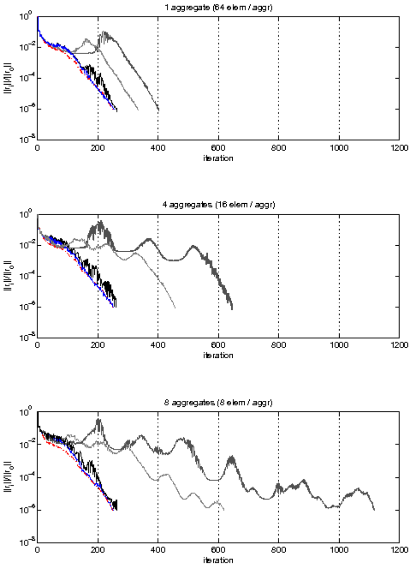
2D simplification



Convergence of PCG with various size of the jumps



Convergence of Deflated PCG



Preconditioned Conjugate Gradient Method

Preconditioned Conjugate Gradients Method (PCG)[1]

Solve iteratively:

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

where M is a traditional preconditioner that resembles A

Requirements for Preconditioner M

- $Mz = y$ is relatively easy to solve
- $M^{-1}A$ has a smaller condition number than A

Theorem [2]

Exact error of PCG after iteration j:

$$||x - x_j||_A \leq 2||x - x_0||_A \left(\frac{\sqrt{\tilde{\kappa}(M^{-1}A)} - 1}{\sqrt{\tilde{\kappa}(M^{-1}A)} + 1} \right)^j$$

Preconditioned Conjugate Gradient Method

Problem of PCG

The spectrum of $M^{-1}A$ contains a number of small eigenvalues

Consequence

$\tilde{\kappa}\left(M^{-1}A\right)$ is large $\longrightarrow$ Slow convergence of the iterative process

Question

Can the convergence of PCG be improved by eliminating those small eigenvalues in some way?

Deflated PCG

History

CG	Ar	1950
Preconditioned CG	$M^{-1}Ar$	1980
Block Preconditioned CG	$\sum_{i=1}^{r}(M_i^{-1})Ar$	1990
Block Preconditioned Deflated CG	$\sum_{i=1}^{r}(M_i^{-1})PAr$	2000

Definitions

Given are

SPD coefficient matrix, $A \in \mathbb{R}^{n \times n}$
projection subspace matrix, $Z \in \mathbb{R}^{n \times k}$, with full rank

Projection operator

Define $E \in \mathbb{R}^{k \times k}$, $Q \in \mathbb{R}^{n \times n}$, and the projection matrix, $P \in \mathbb{R}^{n \times n}$, as follows:

$$P := I - AQ, \quad Q := ZE^{-1}Z^T, \quad E := Z^T AZ.$$

In addition, $M \in \mathbb{R}^{n \times n}$ is an SPD matrix that is called the preconditioner.

Properties of the projection operators

Lemma

Let A, Z, Q and P be as in the previous definition. Then, the following equalities hold:

(a) $P = P^2$;

(b) $PA = AP^T$;

(c) $P^T Z = \mathbf{0}$, $P^T Q = \mathbf{0}$;

(d) $PAZ = \mathbf{0}$, $PAQ = \mathbf{0}$;

(e) $QA = I - P^T$, $QAZ = Z$, $QAQ = Q$;

(f) $Q^T = Q$.

Two-Level PCG methods

List of considered methods

Name	Method	Operator	References
PREC	Traditional Preconditioned CG	M^{-1}	Go96L, Mei77V
AD	Additive Coarse-Grid Correction	$M^{-1}+Q$	Bra86PS,Tos05W
DEF1	Deflation Variant 1	$M^{-1}P$	Vui99SM
DEF2	Deflation Variant 2	P^TM^{-1}	Kol98,Nic87 Sa00YEG
A-DEF1	Adapted Deflation Variant 1	$M^{-1}P+Q$	Smi96BG, Tro01OS
A-DEF2	Adapted Deflation Variant 2	$P^TM^{-1}+Q$	Smi96BG, Tro01OS
BNN	Abstract Balancing	$P^TM^{-1}P+Q$	Man93
R-BNN1	Reduced Balancing Variant 1	$P^TM^{-1}P$	–
R-BNN2	Reduced Balancing Variant 2	P^TM^{-1}	Man93,Tos05W

General Two-Level PCG Method for solving $Ax = b$

Two-Level PCG

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

Choices of $\mathcal{V}_{\text{start}}, \mathcal{M}_1, \mathcal{M}_2, \mathcal{M}_3, \mathcal{V}_{\text{end}}$ for each method

Choices

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	$Qb+P^Tx_{j+1}$
DEF2	$Qb+P^T\bar{x}$	M^{-1}	P^T	I	x_{j+1}
A-DEF1	$\bar{x}$	$M^{-1}P+Q$	I	I	x_{j+1}
A-DEF2	$Qb+P^T\bar{x}$	$P^TM^{-1}+Q$	I	I	x_{j+1}
BNN	$\bar{x}$	$P^TM^{-1}P+Q$	I	I	x_{j+1}
R-BNN1	$Qb+P^T\bar{x}$	$P^TM^{-1}P$	I	I	x_{j+1}
R-BNN2	$Qb+P^T\bar{x}$	P^TM^{-1}	I	I	x_{j+1}

Choices for Z

Choices

- Z is sufficiently sparse, so that Z and AZ can be stored in two vectors;
- Z is full, so that Z and AZ are full matrices.

Motivation

- DDM, the columns of Z correspond to subdomains
- (approximated) eigenvector deflation methods

Costs to compute $Z^T y$ or (AZy)

- one inner product
- one matrix vector multiplication

Extra computational cost per iteration of the two-level PCG methods

IP = inner products, MVM = matrix-vector multiplications, VU = vector updates and CSS = coarse system solves

	Theory		Implementation		
Method	$Py, P^T y$	Qy	**IP / MVM**	**VU**	**CSS**
AD	0	1	2	0	1
DEF1	1	0	2	1	1
DEF2	1	0	2	1	1
A-DEF1	1	1	3	1	1
A-DEF2	1	1	4	1	2
BNN	2	1	5	2	2
R-BNN1	2	0	4	2	2
R-BNN2	1	0	2	1	1

Comparison

Theorem

The following two statements hold:

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

Interpretation

DEF1, DEF2, R-BNN1, and R-BNN2 have identical spectra, and the same is true for BNN, A-DEF1, and A-DEF2

Comparison

Theorem

Let the spectra of DEF1 and BNN be given by

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

respectively. Then, the eigenvalues within these spectra can be ordered such that the following statements hold:

- $\lambda_i = 0$ *and* $\mu_i = 1$, *for* $i = 1, \ldots, k$;
- $\lambda_i = \mu_i$, *for* $i = k + 1, \ldots, n$.

Comparison

Theorem

Suppose that the spectrum of DEF1, DEF2, R-BNN1, or R-BNN2 is

$\{0, \ldots, 0, \lambda_{k+1}, \ldots, \lambda_n\}$, *with* $\lambda_{k+1} \leq \lambda_{k+2} \leq \ldots \leq \lambda_n$

and the spectrum of BNN, A-DEF1, or A-DEF2 is

$\{1, \ldots, 1, \mu_{k+1}, \ldots, \mu_n\}$, *with* $\mu_{k+1} \leq \mu_{k+2} \leq \ldots \leq \mu_n$.

Then, $\lambda_i = \mu_i$ *for all* $i = k+1, \ldots, n$.

Equivalent methods

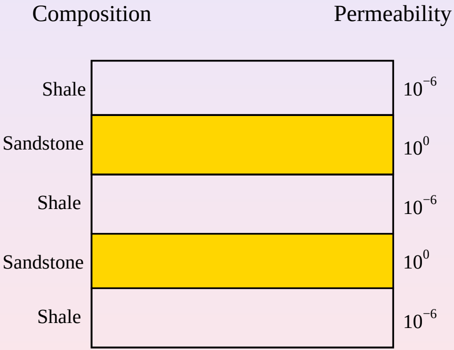
Theorem

Let $\bar{x} \in \mathbb{R}^n$ be an arbitrary vector. The following methods produce exactly the same iterates in exact arithmetic:

- *BNN with $\mathcal{V}_{start} = Qb + P^T\bar{x}$;*
- *DEF2, A-DEF2, R-BNN1 and R-BNN2 (with $\mathcal{V}_{start} = Qb + P^T\bar{x}$);*
- *DEF1 (with $\mathcal{V}_{start} = \bar{x}$) whose iterates are based on $Qb + P^T x_{j+1}$.*

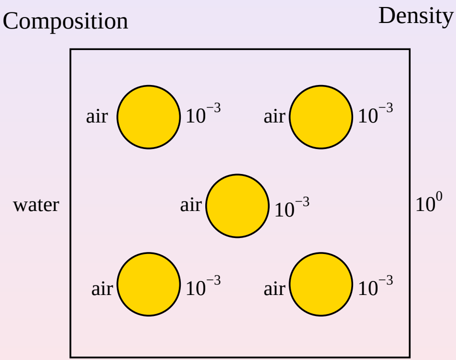
Numerical comparison

Porous media problem

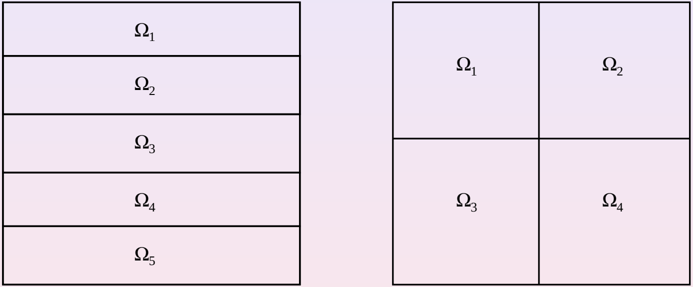


Numerical comparison

Bubbly flow problem



Geometry of subdomains



(a) Porous media problem. (b) Bubbly flow problem.

Figure: Geometry of subdomains Ω_j. Number of subdomains is fixed in the porous media problem, whereas it can be varied in the bubbly flow problem.

Standard parameters

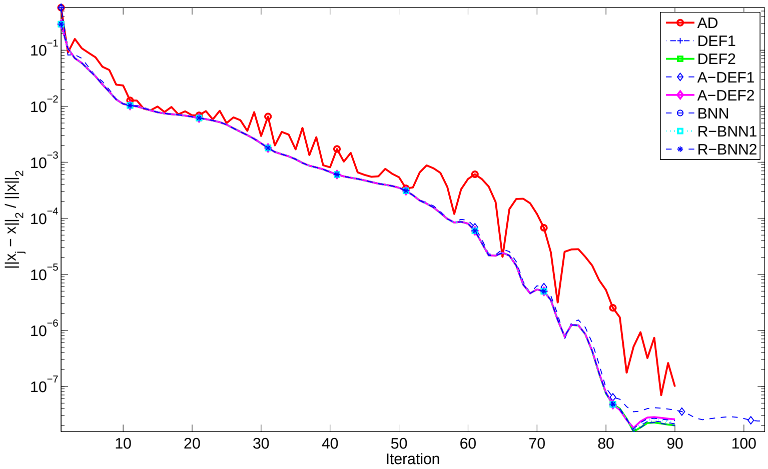


Figure: Relative errors in 2−norm during the iterative process, for the porous media problem with

Approximate coarse solves

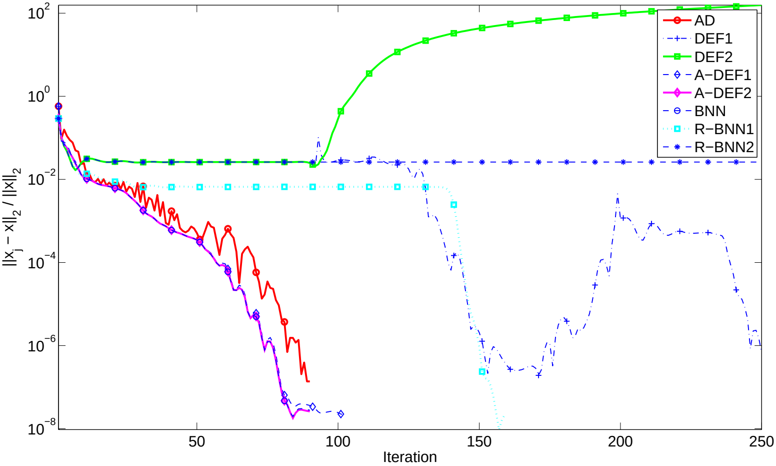


Figure: Relative errors in 2−norm during the iterative process for the porous media problem with

Severe termination criteria

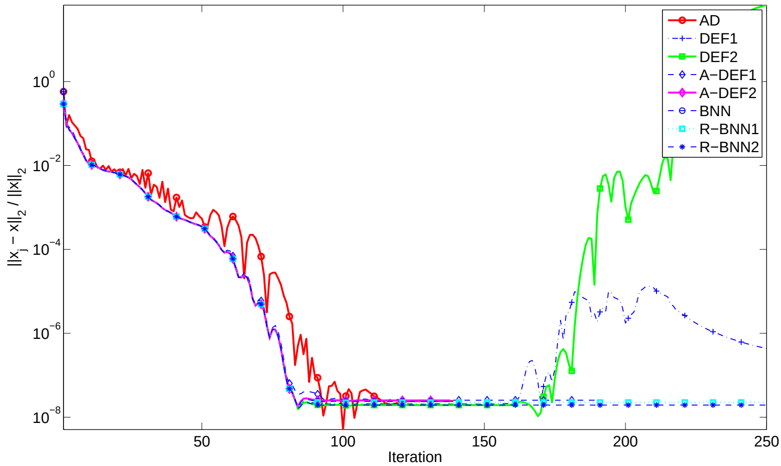
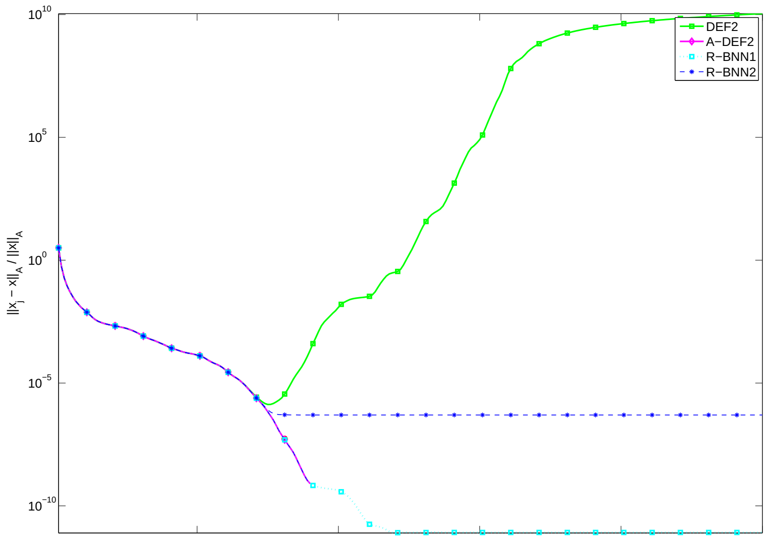


Figure: Relative 2−norm errors during the iterative process for the porous media problem with

Perturbed starting vectors



Conclusions

Conclusions

- Various methods from deflation, additive coarse grid correction and balancing can be written in one framework
- DEF1, DEF2, R-BNN1, and R-BNN2 have identical spectra
- BNN, A-DEF1, and A-DEF2 have identical spectra
- Theoretically the methods are very close with respect to convergence
- With respect to cost and robustness of implementation there are serious differences
- A-DEF2 seems to be the most robust and fastest method

Further reading

- http://ta.twi.tudelft.nl/nw/users/vuik/pub_it_def.html
- http://ta.twi.tudelft.nl/nw/users/vuik/papers/Tan07NVE.pdf
  accepted for publication in Journal of Scientific Computing