Projection acceleration of Krylov solvers

C. Vuik [1], J.M. Tang [1], R. Nabben [2]

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

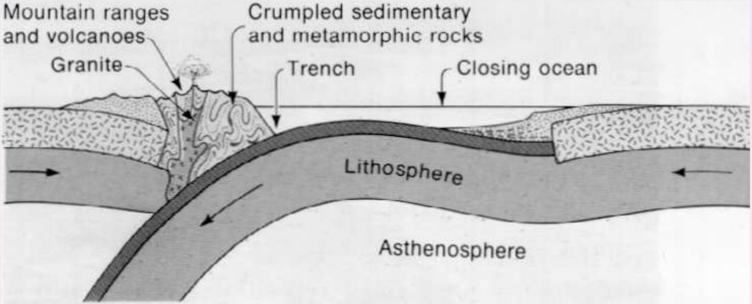
[2] Technische Universität Berlin
Institut für Mathematik

ICIAM 2007
July 16–20, 2007

Outline

Layered problem

Knowledge of the fluid pressure in rock layers is important for an oil company to predict the presence of oil and gas in reservoirs.



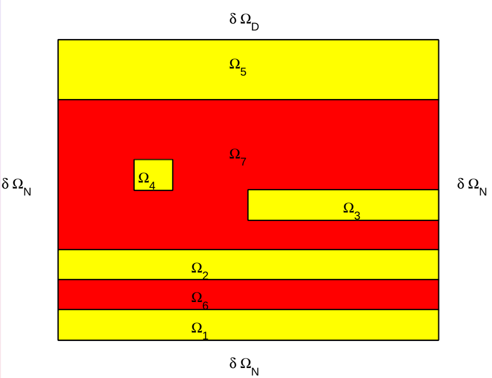
The earth's crust has a layered structure

Bubbly flow

Simulation of flows with bubbles and droplets

Mathematical model for layered problem

Computation of fluid pressure $-\text{div}(\sigma \nabla p(x)) = 0$ on Ω, p fluid pressure, σ permeability



$\sigma_h = 1$ (sand) $\sigma_l = \varepsilon = 10^{-7}$ (shale)

Properties and Applications

Problem

$$Ax = b$$

A is sparse and SPD
Condition number of A is $O(10^7)$, due to large contrast in permeability

Applications

- reservoir simulations
- porous media flow
- electrical power networks
- semiconductors
- magnetic field simulations
- bubbly flow

Spectrum of IC preconditioned matrix

Definitions

L is the Incomplete Cholesky factor of A

k^s is the number of high-permeability domains not connected to a Dirichlet boundary

Theorem

The IC preconditioned matrix $L^{-1}AL^{-T}$ has k^s eigenvalues of $O(\varepsilon)$.

Deflated ICCG

Idea: remove the bad eigenvectors from the error/residual.

Krylov	Ar
Preconditioned Krylov	$M^{-1}Ar$
Block Preconditioned Krylov	$\sum_{i=1}^{m}(M_i^{-1})Ar$
Block Preconditioned Deflated Krylov	$\sum_{i=1}^{m}(M_i^{-1})PAr$

Choices

Various choices

- Projection vectors
  Physical vectors, eigenvectors, coarse grid projection vectors (constant, linear, ...)
- Projection method
  Deflation, coarse grid projection, balancing, augmented, FETI
- Implementation
  sparseness, with(out) using projection properties, optimized, ...

Deflation Method

Notation

Projection matrix is defined by $P := I - AQ$ with

- correction matrix $Q := ZE^{-1}Z^T$
- coarse matrix $E := Z^TAZ$
- projection subspace matrix $Z \in \mathbb{R}^{n \times r}$ where $r \ll n$

Remarks

- E has dimensions $r \times r \; \rightarrow \; E^{-1}$ is easy to compute
- Q is an approximation of A^{-1} based on a subspace

Deflated PCG

Solve iteratively:

$$M^{-1}PAx = M^{-1}Pb$$

where $P = I - AQ$

Physical deflation vectors

Subdomains

k is number of subdomains

$\Omega_i, i = 1, \ldots, k^s$ high-permeability subdomains without a Dirichlet B.C.;
$i = k^s + 1, \ldots, k^h$ remaining high-permeability subdomains

Physical deflation vectors [1]

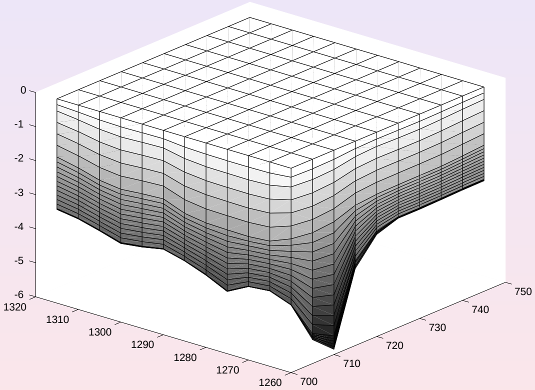
- define z_i for $i \in \{1, \ldots, k^s\}$
- $z_i = 1$ on $\bar{\Omega}_i$ and $z_i = 0$ on $\bar{\Omega}_j, j \neq i, j \in \{1, \ldots, k^h\}$
- z_i satisfies equation:

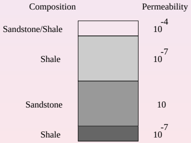
$$-\text{div}(\sigma_j \nabla z_i) = 0 \text{ on } \Omega_j, j \in \{k^h + 1, \ldots, k\},$$

with appropriate boundary conditions

[1] C. Vuik, A. Segal, L. El Yaakoubi and E. Dufour, *A comparison of various deflation vectors applied to elliptic problems with discontinuous coefficients*, Applied Numerical Mathematics, **41**, pp. 219–233, 2002.

Geometry oil flow problem





Results oil flow problem

Varying σ_{shale}

σ	ICCG		DICCG	
	λ_{min}	iter	λ_{min}	iter
10^{-3}	$1.5 \cdot 10^{-2}$	26	$6.9 \cdot 10^{-2}$	20
10^{-5}	$2.2 \cdot 10^{-4}$	59	$7.7 \cdot 10^{-2}$	20
10^{-7}	$2.3 \cdot 10^{-6}$	82	$7.7 \cdot 10^{-2}$	20

Varying accuracy

accuracy	ICCG		DICCG	
	iter	CPU	iter	CPU
10^{-5}	82	18.9	20	6.3
10^{-3}	78	18.0	12	4.1
10^{-1}	75	17.2	2	1.2

Standard Projection Methods

Deflated PCG

Solve iteratively:

$$M^{-1}PAx = M^{-1}Pb$$

where $P = I - AQ$

Additive Coarse Grid Correction Method [1]

Solve iteratively:

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

Balancing Neumann-Neumann Method [2]

Solve iteratively:

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

[1] 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.

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

Idea of Projection

General Projected PCG

Solve iteratively:

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

where $\mathcal{P}$ is a projection operator based on M^{-1}, P and Q

Idea of Projection Operator

$\mathcal{P}$ gets rid of both small and/or large eigenvalues of A

Choices for $\mathcal{P}$

- Traditional PCG: $\mathcal{P} = M^{-1}$
- Deflated PCG: $\mathcal{P} = M^{-1}P$
- Additive: $\mathcal{P} = M^{-1} + Q$
- Balancing: $\mathcal{P} = P^T M^{-1} P + Q$
- Reduced Balancing / Deflation: $\mathcal{P} = P^T M^{-1}$

General Projection Methods

Possible Choices for $\mathcal{P}$

Name	Method	Operator $\mathcal{P}$
PCG	Traditional PCG	M^{-1}
DEF1	Deflated PCG 1	$M^{-1}P$
AD	Additive Coarse Grid Correction	$M^{-1} + Q$
BNN	Abstract Balanced PCG	$P^T M^{-1} P + Q$
DEF2	Deflated PCG 2	$P^T M^{-1}$
R-BNN2	Reduced Balanced PCG 2	$P^T M^{-1}$
R-BNN1	Reduced Balanced PCG 1	$P^T M^{-1} P$
A-DEF1	Adapted Deflated PCG 1	$M^{-1} P + Q$
A-DEF2	Adapted Deflated PCG 2	$P^T M^{-1} + Q$

Origin of the Methods

Methods can be derived from the theory of

- deflation
- domain decomposition
- multigrid

General Projection Methods

Deflation

- M is a preconditioner
- P is the deflation matrix
- Z is the deflation subspace matrix consisting of approximated eigenvectors
- r is small
- $Ex = y$ is solved directly

Multigrid

- M is a smoother
- P is a coarse grid correction
- Z is the restriction operator
- Z^T is the prolongation/interpolation operator
- r is relatively large
- $Ex = y$ is solved recursively

General Projection Methods

Domain Decomposition

- M deals with exact/inexact solves on subdomains
- P is a subspace correction
- Z is the restriction operator
- Z^T is the prolongation/interpolation operator
- $1 \ll r \ll n$
- $Ex = y$ is solved directly/iteratively

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

Best Method of our List?

- Theoretical comparison
- Numerical comparison

First step: Compare the condition numbers of system $\mathcal{P}A$

[1] 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.

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

[3] R. Nabben and C. Vuik, *A comparison of abstract versions of deflation, balancing and additive coarse grid correction preconditioners*, Report 07-01, 2007. (submitted)

Theoretical Comparison

Theorem (Class 1)

DEF1, DEF2, R-BNN1 and R-BNN2 have the same condition numbers:

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

Theorem (Class 2)

BNN, A-DEF1, A-DEF2 have the same condition numbers:

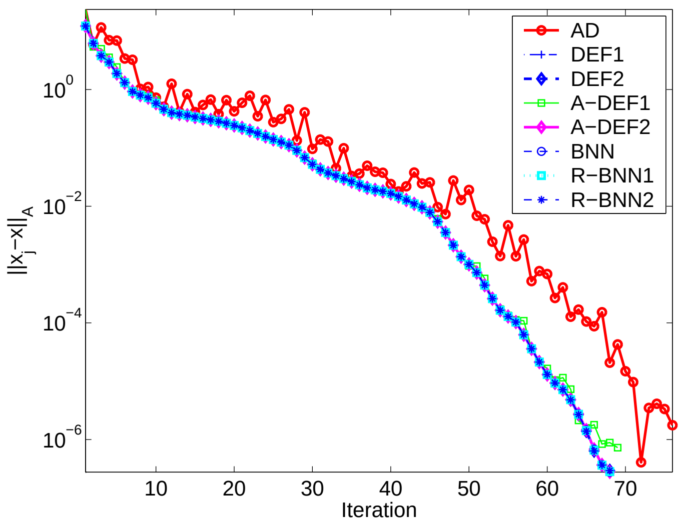
$$\begin{aligned}\sigma\left((P^TM^{-1}P+Q)A\right) = \sigma\left((M^{-1}P+Q)A\right) &= \sigma\left((P^TM^{-1}+Q)A\right)\\ &= \{1, 1, \ldots, 1, \lambda_{r+1}, \ldots, \lambda_n\}\end{aligned}$$

Theorem

AD has a worse condition number compared to the other projection methods

Numerical Comparison

Typical Convergence Behavior



Results

Best Method

DEF1 ($\mathcal{P} = M^{-1}P$), DEF2 ($\mathcal{P} = P^T M^{-1}$) and R-BNN2 ($\mathcal{P} = P^T M^{-1}$) are the best methods, because

- their corresponding matrices have the best condition numbers
- they have the lowest cost per iteration

Most Stable Method?

Compare methods with respect to

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

Theoretical Comparison

Spectrum after Perturbating E^{-1} with a Small Matrix $\epsilon Rand(k,k)$

- DEF1, DEF2, R-BNN1 and R-BNN2:

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

- BNN, A-DEF1, A-DEF2:

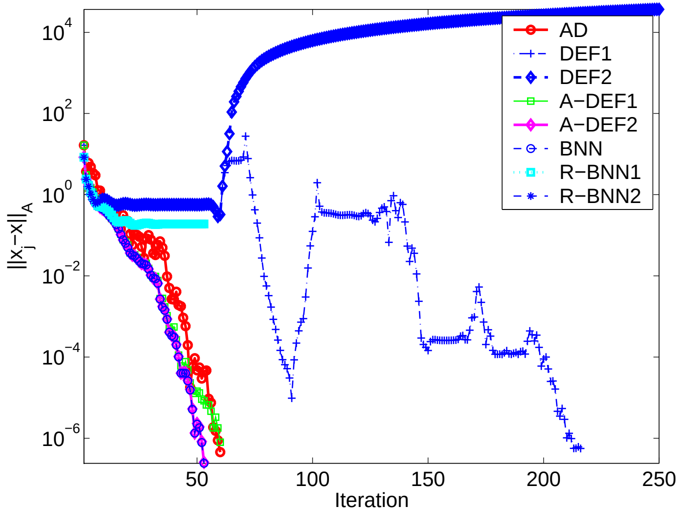
$$\sigma \approx \{1+\mathcal{O}(\epsilon), 1+\mathcal{O}(\epsilon), \ldots, 1+\mathcal{O}(\epsilon), \lambda_{r+1}, \ldots, \lambda_n\}$$

Consequence

Class 1 is unstable, whereas class 2 is stable

Numerical Comparison

Typical Convergence Behavior



Results

Best Method?

A-DEF2 ($\mathcal{P} = P^T M^{-1} + Q$) is the best method because

- it is fast and stable
- it has a low cost per iteration

Conclusions [1]

- DEF1, DEF2, and R-BNN2 have the best condition numbers and the lowest cost per iteration
- BNN, A-DEF1, A-DEF2 are the most stable methods

[1] J.M. Tang, R. Nabben, C. Vuik and Y.A. Erlangga, *Theoretical and numerical comparison of various projection methods derived from deflation, domain decomposition and multigrid methods*, *submitted*. (See also DUT Report 07-04)

Problem Setting

Question 2

How to choose Z for bubbly flows?

Background

- Simulation of flows with bubbles and droplets
- Flow governed by the Navier-Stokes equations with unknowns p and u:

$$\begin{cases} \frac{\partial u}{\partial t} + u \cdot \nabla u + \frac{1}{\rho} \nabla p = \frac{1}{\rho} \nabla \cdot \mu \left(\nabla u + \nabla u^T \right) + g \\ \nabla \cdot u = 0 \end{cases}$$

- Solution using operator-splitting methods

Analysis of Eigenvalues and Eigenvectors

Eigenvectors associated with $\mathcal{O}(10^{-3})$-eigenvalues

- constant in bubbles
- linear elsewhere

Approximations

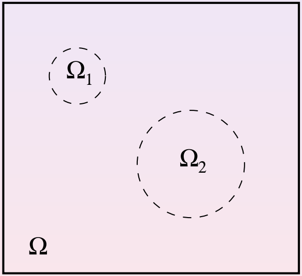
The eigenvectors remain good approximations if

- the linear parts are perturbed arbitrarily
- the constant part are perturbed by a constant

Consequence

Levelset projection vectors can approximate these eigenvectors

Levelset Projection



Projection subspace matrix

$Z = [z_1 \; z_2 \; \cdots \; z_r]$ consists of

$$(z_j)_i = \begin{cases} 0, & x_i \in \Omega \setminus \bar{\Omega}_j \\ 1, & x_i \in \Omega_j \end{cases}$$

Analysis of Eigenvalues and Eigenvectors

Eigenvectors associated with $\mathcal{O}(1)$-eigenvalues

Smooth and slow-varying in the whole domain

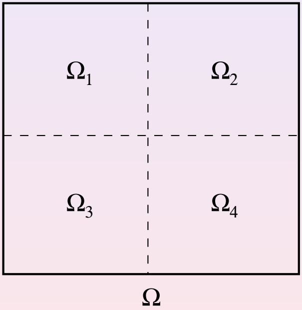
Approximations

These eigenvectors remain good approximations if they are slightly perturbed

Consequence

Subdomain projection vectors can approximate these eigenvectors

Subdomain Projection



Projection subspace matrix

$Z = [z_1 \; z_2 \; \cdots \; z_r]$ consists of

$$(z_j)_i = \begin{cases} 0, & x_i \in \Omega \setminus \bar{\Omega}_j \\ 1, & x_i \in \Omega_j \end{cases}$$

Properties of Projection Vectors

Levelset Projection Vectors

- Projection of $\mathcal{O}(10^{-3})$-eigenvalues to zero
- Very sparse structure
- Only a few vectors required
- Variable at each time step

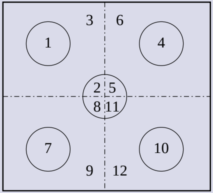
Subdomain Projection Vectors

- Projection of $\mathcal{O}(1)$-eigenvalues to zero
- Sparse structure
- Reasonable number of vectors required
- Fixed at each time step

Further Analysis

Combination of Levelset and Subdomain Projection

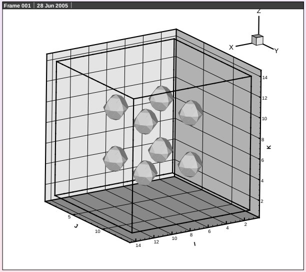
Both approaches can be combined leading to levelset-subdomain projection:



Properties of Levelset-Subdomain Projection Vectors

- Projection of both $\mathcal{O}(10^{-3})$- and $\mathcal{O}(1)$-eigenvalues to zero
- Sparse structure
- Many levelset-subdomain projection vectors are required
- Variable at each time step

Experiment 1



Settings

- 3-D bubbly flow, water and air, density ratio $= 10^3$
- Finite differences, uniform Cartesian grid, $n = 100^3$
- $Ax = b$ is solved, ICCG and DICCG, tol $= 10^{-8}$

Results

Results for 8 Bubbles

Method	# Iter.	CPU
ICCG	291	43.0
DICCG-2^3	160	29.1
DICCG-5^3	72	14.2
DICCG-10^3	**36**	**8.2**
DICCG-20^3	22	27.2

DICCG$-r$ = DICCG with r subdomain deflation vectors
 # Iter. = number of iterations
CPU = computational time in seconds

Discussion of the Results

Conclusions of Experiment 1

- DICCG performs much better compared to ICCG
- Optimal choice of r in 3-D : $r \approx \sqrt{n}$
- Deflation vectors approximate the 'bad' eigenvectors

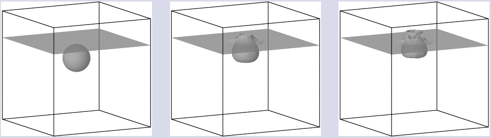
Question

What about realistic time-dependent problems?

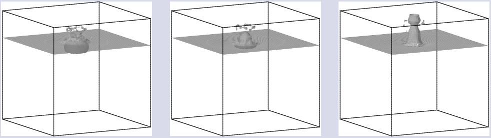
Experiment 2

Rising Bubble in Water without Surface Tension

Simulation of the first 250 time steps:



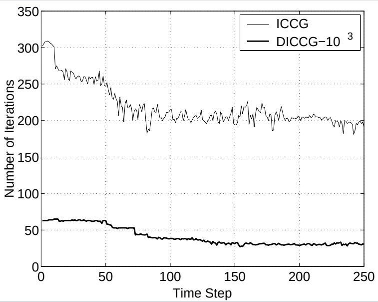
(a) $t = 0$ (b) $t = 50$ (c) $t = 100$



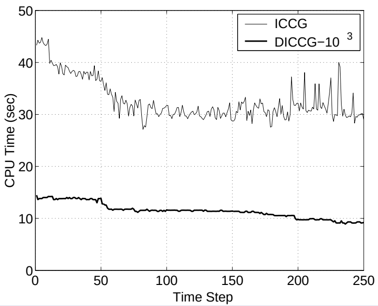
(d) $t = 150$ (e) $t = 200$ (f) $t = 250$

Results

Results ICCG and DICCG with $r = 10^3$



(a) number of iterations



(b) CPU time (in seconds)

Discussion of the Results

Conclusions of Experiment 2

- DICCG with $r = 10^3$ performs better compared to ICCG
- DICCG hardly depends on the geometry of the problem

Conclusions

- DICCG is a robust and efficient method to solve elliptic problems with discontinuous coefficients
- The choice of the projection vectors is important for the success of a projection method
- DEF1, DEF2, and R-BNN2 have the best condition numbers and the lowest cost per iteration
- BNN, A-DEF1, A-DEF2 are the most stable methods

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