A parallel deflated Krylov solver for finite element problems

Kees Vuik, Guus Segal and Fred Vermolen
c.vuik@math.tudelft.nl

http://ta.twi.tudelft.nl/users/vuik/

Delft University of Technology

Sparse Days and Grid Computing at St. Girons,

Hotel La Clairiere, St. Girons, France, June 10-13, 2003

Contents

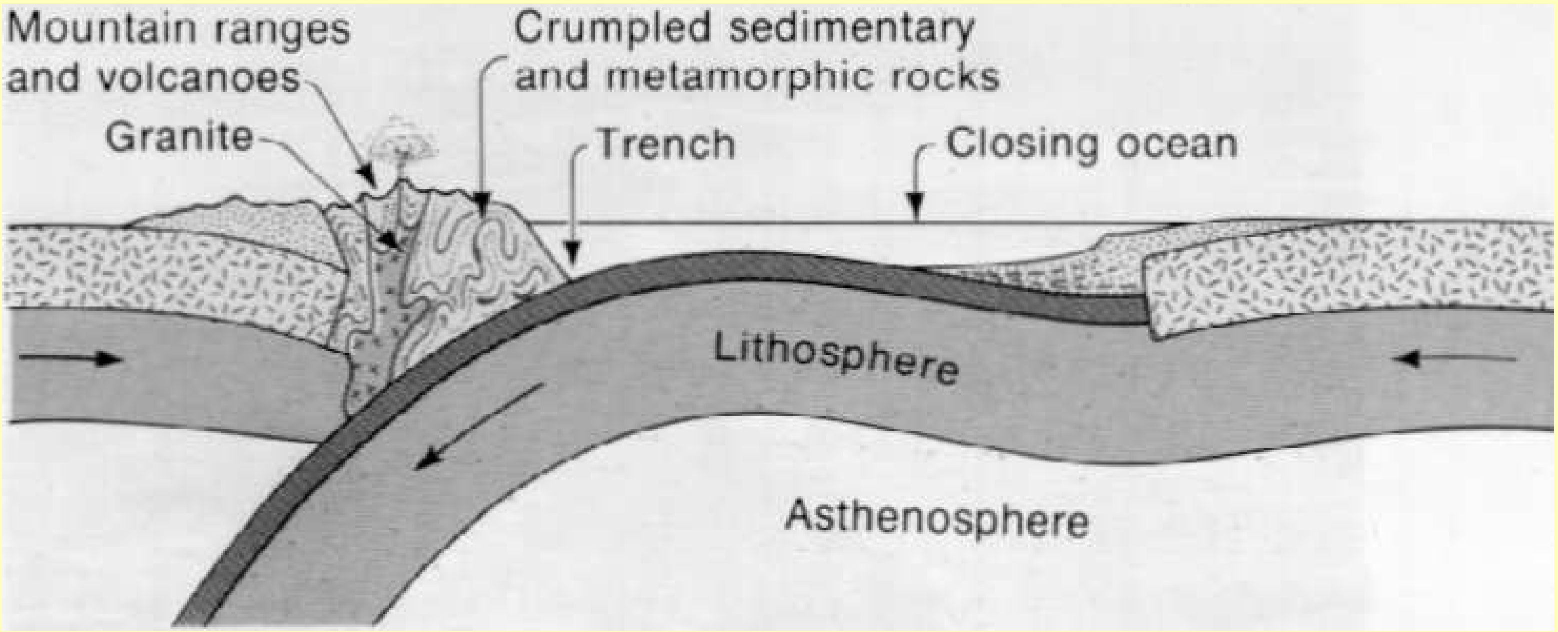
Reinhard Nabben, Jason Frank, Koos Meijerink, Erwin Dufour,

Gjalt Wijma, Larbi el Yaakoubi

1. Introduction

Motivation

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

Incompressible Navier-Stokes problems

Discretized incompressible Navier-Stokes

- Momentum equations
- Pressure equation
- Transport equation

Coupled problem

$$\begin{pmatrix} \mathbf{Q} & \mathbf{G} \\ \mathbf{G}^T & \mathbf{0} \end{pmatrix} \begin{pmatrix} u \\ p \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \end{pmatrix}, \; u \in \mathbb{R}^n \text{ and } p \in \mathbb{R}^m$$

Solve the system $Ax = b$

TU Delft

Literature review

- Robust preconditioners
  (M)ICCG vd Vorst, Meijering, Gustafsson
  ILUT Saad, MRILU Ploeg, Wubs
  Navier-Stokes Elman, Silvester, Wathen, Golub
  RIF Benzi, Tuma

TU Delft

Literature review

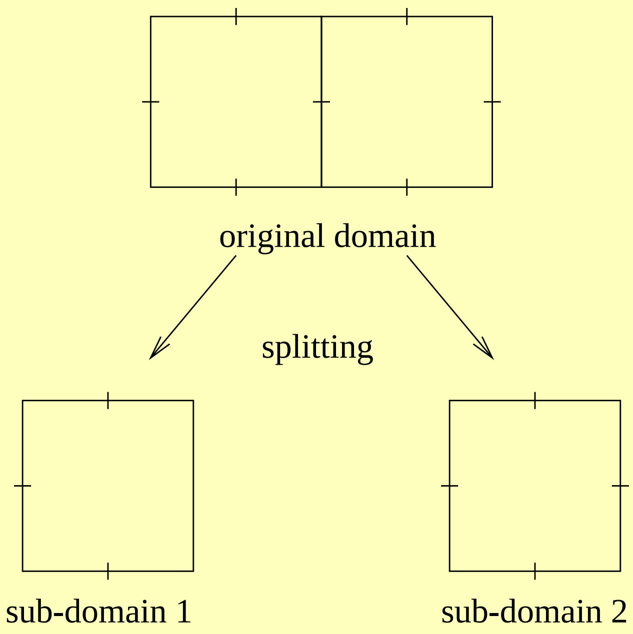
- Robust preconditioners
  (M)ICCG vd Vorst, Meijering, Gustafsson
  ILUT Saad, MRILU Ploeg, Wubs
  Navier-Stokes Elman, Silvester, Wathen, Golub
  RIF Benzi, Tuma
- Parallel preconditioners
  Block variants see above
  ILU Bastian, Horton, Vuik, Nooyen, Wesseling
  SPAI Grote, Huckle, Benzi, Tuma, Chow, Saad

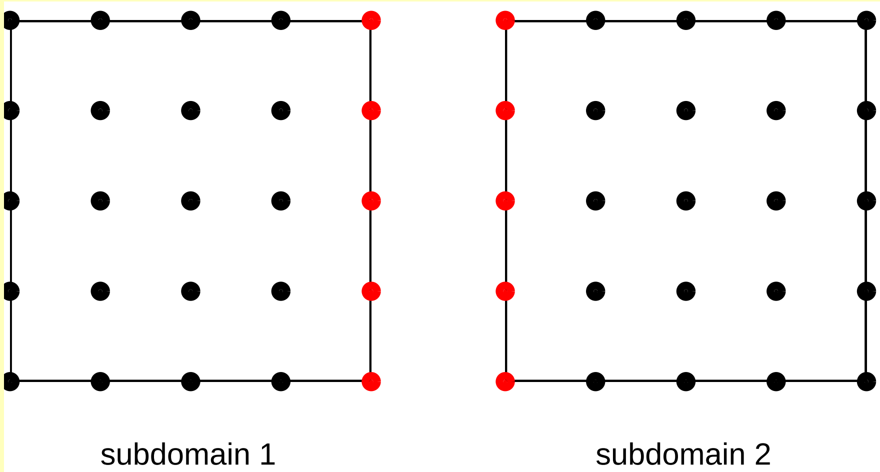
Literature review

- Robust preconditioners
  (M)ICCG vd Vorst, Meijering, Gustafsson
  ILUT Saad, MRILU Ploeg, Wubs
  Navier-Stokes Elman, Silvester, Wathen, Golub
  RIF Benzi, Tuma
- Parallel preconditioners
  Block variants see above
  ILU Bastian, Horton, Vuik, Nooyen, Wesseling
  SPAI Grote, Huckle, Benzi, Tuma, Chow, Saad
- Acceleration of parallel preconditioners
  CGC Notay, vd Velde, Benzi, Frommer, Nabben, Szyld, Chan, Mathew, Dryja, Widlund, Padiy, Axelsson, Polman
  Deflation Nicolaides, Mansfield, Kolotilina, Frank, Vuik
  Morgan, Chapman, Saad, Burrage, Ehrel, Pohl
  FETI Farhat, Roux, Mandel, Klawonn, Widlund

2. A parallel Krylov method for finite element problems

Data distribution





TU Delft

Parallelization of ICCG

ICCG

$k = 0,\ r_0 = b - Ax_0,\ p_1 = z_1 = L^{-T}L^{-1}r_0;$

while $\|r_k\|_2 > \varepsilon$ **do**

$$k = k + 1;$$

$$\alpha_k = \frac{(r_{k-1}, z_{k-1})}{(p_k, Ap_k)};$$

$$x_k = x_{k-1} + \alpha_k p_k;$$

$$r_k = r_{k-1} - \alpha_k Ap_k;$$

$$z_k = L^{-T}L^{-1}r_k;$$

$$\beta_k = \frac{(r_k, z_k)}{(r_{k-1}, z_{k-1})};$$

$$p_{k+1} = z_k + \beta_k p_k;$$

end while

Explanation for a 1D example

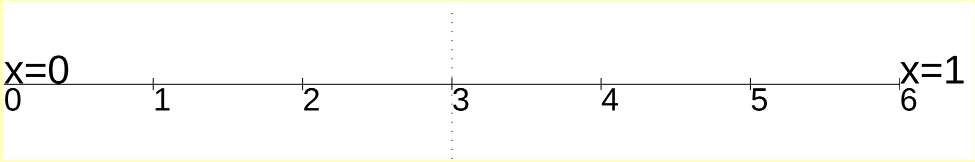
Building blocks

- vector update
- inner product
- matrix vector product
- preconditioner vector product

$$-\frac{d^2y}{dx^2} = f, \;\; y(0) = y(1) = 0.$$

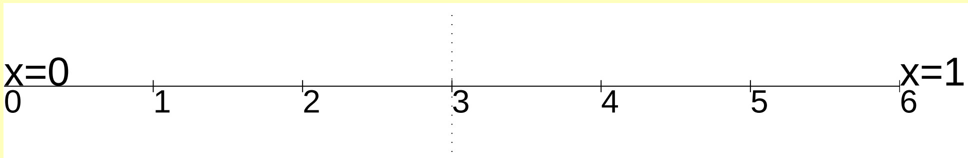
Take $n = 5$ and decompose the domain into two subdomains (1 and 2)

Vector update



We define $I_1 = \{1, 2, 3, \}$ and $I_2 = \{3, 4, 5\}$. Note that there is an overlap of 1 point.

Vector update



We define $I_1 = \{1, 2, 3, \}$ and $I_2 = \{3, 4, 5\}$. Note that there is an overlap of 1 point.

Global vector $\mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{pmatrix}$, local vectors $\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}$ and $\begin{pmatrix} x_3 \\ x_4 \\ x_5 \end{pmatrix}$.

Vector update is straight forward.

Inner product

- Determine the local innerproduct
- Sum the local innerproducts by MPI_ALLREDUCE

Inner product

- Determine the local innerproduct
- Sum the local innerproducts by MPI_ALLREDUCE

But

TU Delft

Inner product

- Determine the local innerproduct
- Sum the local innerproducts by MPI_ALLREDUCE

But

The contributions of the interface points are used more than once.

Solution: use the interface points only in one local inner product.

Matrix vector product

$$A = \begin{pmatrix} 2 & -1 & 0 & 0 & 0 \\ -1 & 2 & -1 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} + \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & -1 & 2 & -1 \\ 0 & 0 & 0 & -1 & 2 \end{pmatrix}$$

Matrix vector product

$$A = \begin{pmatrix} A_{11} & 0 \\ 0 & 0 \end{pmatrix} + \begin{pmatrix} 0 & 0 \\ 0 & A_{22} \end{pmatrix}$$

Matrix vector product

$$A = \begin{pmatrix} A_{11} & 0 \\ 0 & 0 \end{pmatrix} + \begin{pmatrix} 0 & 0 \\ 0 & A_{22} \end{pmatrix}$$

The global matrix vector product $\mathbf{p} = A\mathbf{x}$:

1. Determine $\begin{pmatrix} p_1 \\ p_2 \\ p_3^l \end{pmatrix} = A_{11} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}$ and $\begin{pmatrix} p_3^r \\ p_4 \\ p_5 \end{pmatrix} = A_{22} \begin{pmatrix} x_3 \\ x_4 \\ x_5 \end{pmatrix}$ in parallel.
2. Communication: send p_3^l from CPU1 to CPU2 and send p_3^r from CPU2 to CPU1. (nearest neighbour communication)
3. Determine on both processors $p_3 = p_3^l + p_3^r$ in parallel.

Parallelization of a block preconditioner

Take as preconditioner the following

$$\mathbf{p} = P^{-1}\mathbf{x} = \left(\sum_{i=1}^{p} R_i^T P_{i,i}^{-1} R_i \right) \mathbf{x}$$

where

$$P_{i,i} \approx A_{i,i}$$

Parallelization of a block preconditioner

Take as preconditioner the following

$$\mathbf{p} = P^{-1}\mathbf{x} = \left(\sum_{i=1}^{p} R_i^T P_{i,i}^{-1} R_i \right) \mathbf{x}$$

where

$$P_{i,i} \approx A_{i,i}$$

In our example

$$R_1 = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{pmatrix} \text{ and } R_2 = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

Parallelization of a block preconditioner

The global preconditioner vector product $\mathbf{p} = P^{-1}\mathbf{x}$:

1. Determine $\begin{pmatrix} p_1 \\ p_2 \\ p_3^l \end{pmatrix} = P_{11}^{-1} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}$ and $\begin{pmatrix} p_3^r \\ p_4 \\ p_5 \end{pmatrix} = P_{22}^{-1} \begin{pmatrix} x_3 \\ x_4 \\ x_5 \end{pmatrix}$ in parallel.
2. Communication: send p_3^l from CPU1 to CPU2 and send p_3^r from CPU2 to CPU1. (nearest neighbour communication)
3. Determine on both processors $p_3 = p_3^l + p_3^r$ in parallel.

3. Deflation and Coarse Grid Acceleration

A is SPD, Conjugate Gradients

$$P = I - AZE^{-1}Z^T \text{ with } E = Z^T AZ$$

and $Z = [z_1 ... z_m]$, where $z_1, ..., z_m$ are independent deflation vectors.

Properties

1. $P^T Z = 0$ and $PAZ = 0$
2. $P^2 = P$
3. $AP^T = PA$

Deflated ICCG

$$x = (I - P^T)x + P^T x,$$

TU Delft

Deflated ICCG

$$x = (I - P^T)x + P^T x,$$

$$(I - P^T)x = ZE^{-1}Z^T Ax = ZE^{-1}Z^T b,$$

TU Delft

Deflated ICCG

$$x = (I - P^T)x + P^T x,$$

$$(I - P^T)x = ZE^{-1}Z^T Ax = ZE^{-1}Z^T b, \qquad AP^T x = PAx = Pb.$$

TU Delft

Deflated ICCG

$$x = (I - P^T)x + P^T x,$$

$$(I - P^T)x = ZE^{-1}Z^T Ax = ZE^{-1}Z^T b, \qquad AP^T x = PAx = Pb.$$

DICCG

$k = 0,\ \hat{r}_0 = Pr_0,\ p_1 = z_1 = L^{-T}L^{-1}\hat{r}_0;$

while $\|\hat{r}_k\|_2 > \varepsilon$ **do**

$k = k + 1;$

$\alpha_k = \frac{(\hat{r}_{k-1}, z_{k-1})}{(p_k, PAp_k)};$

$x_k = x_{k-1} + \alpha_k p_k;$

$\hat{r}_k = \hat{r}_{k-1} - \alpha_k PAp_k;$

$z_k = L^{-T}L^{-1}\hat{r}_k;$

$\beta_k = \frac{(\hat{r}_k, z_k)}{(\hat{r}_{k-1}, z_{k-1})}; \qquad p_{k+1} = z_k + \beta_k p_k;$

end while

Variants for values at interfaces

$z_i = 1$ on Ω_i and $z_i = 0$ on $\Omega \setminus \bar{\Omega}_i$

1. **no overlap**

   $z_i = 1$ at one subdomain
   $z_i = 0$ at other subdomains

2. **complete overlap**

   $z_i = 1$ at all subdomains

3. **average overlap**

   $z_i = \frac{1}{n_{neighbors}}$ at all subdomains

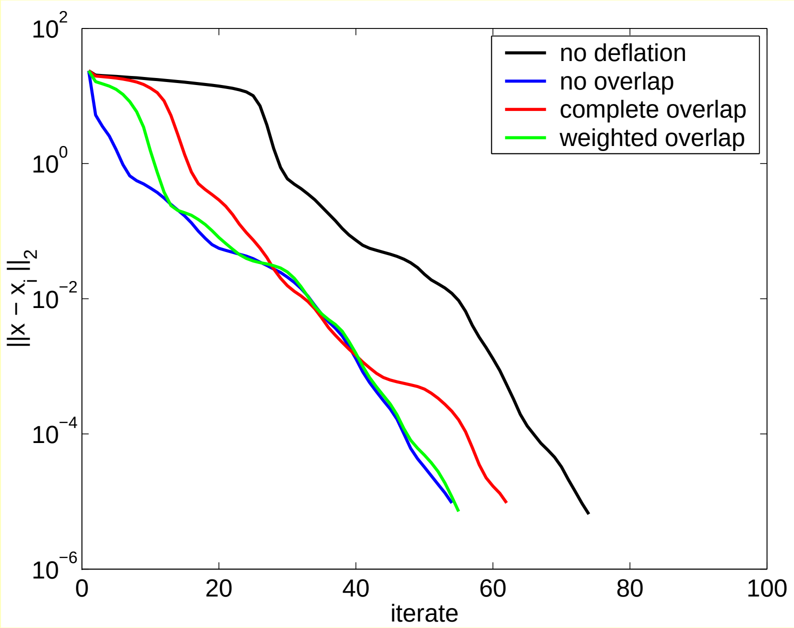
4. **weighted overlap** ($-\text{div}(\sigma \nabla u) = f$)

   $$z_i = \frac{\sigma(i)}{\sum \sigma(neighbors)}$$

TU Delft

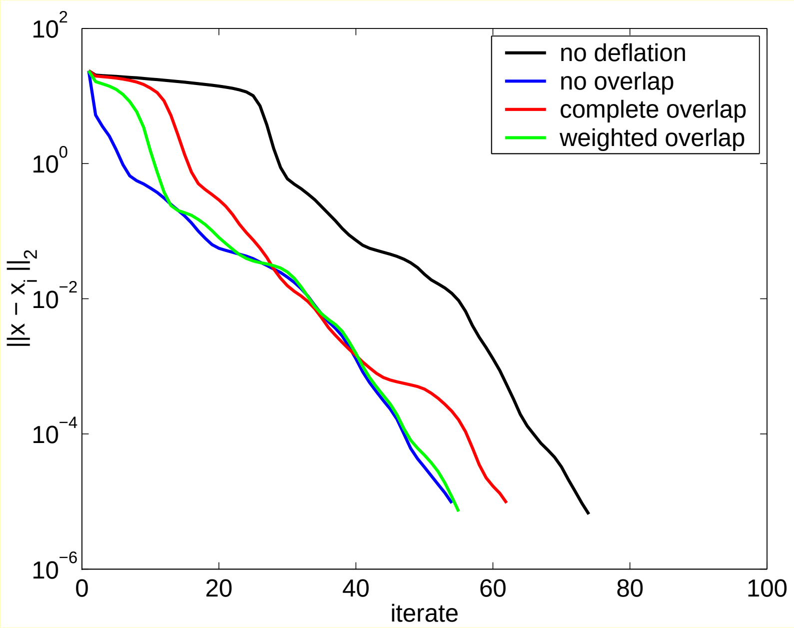
Error for Block IC and Deflation

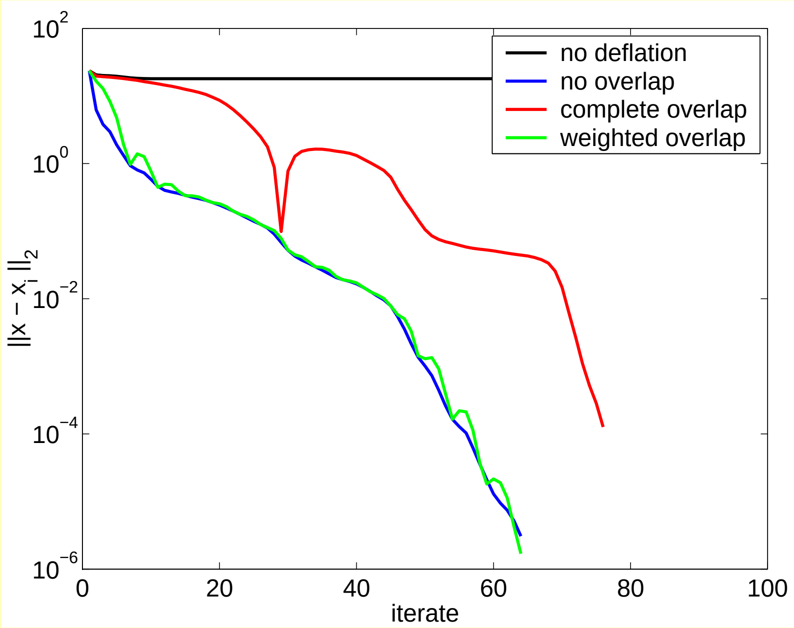
Results for constant coefficients



Error for Block IC and Deflation

Results for constant coefficients and disontinuous coefficients





Parallel implementation (initialization)

Processor 1		Processor 2
Make z_1		Make z_2
	communication	
$z_{2\Gamma}$		$z_{1\Gamma}$
Make Az_1 and $Az_{2\Gamma}$		Make Az_2 and $Az_{1\Gamma}$
	communication	
sum up		sum up
$E_{11} = z_1^T Az_1$, $E_{12} = z_1^T Az_{2\Gamma}$		$E_{22} = z_2^T Az_2$, $E_{12} = z_2^T Az_{1\Gamma}$
	communication	
Determine Choleski decomposition of E		

Parallel implementation (during iteration)

$$P\mathbf{v} = \mathbf{v} - AZ(Z^TAZ)^{-1}Z^T\mathbf{v} = \mathbf{v} - AZE^{-1}Z^T\mathbf{v}$$

Processor 1		Processor 2
Compute z_1^Tv		Compute z_2^Tv
	communication	
$y = E^{-1}\begin{pmatrix} z_1^Tv \\ z_2^Tv \end{pmatrix}$		
	communication	
$\mathbf{v} - y_1Az_1 - y_2Az_{2\Gamma}$		$\mathbf{v} - y_1Az_{1\Gamma} - y_2Az_2$

TU Delft

Coarse Grid Correction of ICCG

Definition

- $Z \in \mathbb{R}^{n \times m}$ with independent columns.
- $E = Z^T A Z \in \mathbb{R}^{m \times m}$, E is SPD.
- $P_C = L^{-T} L^{-1} + \sigma Z E^{-1} Z^T$.

CICCG

$k = 0,\ r_0 = b - Ax_0,\ p_1 = z_1 = L^{-T} L^{-1} r_0;$

while $\|r_k\|_2 > \varepsilon$ **do**

$$k = k + 1;$$

$$\alpha_k = \frac{(r_{k-1}, z_{k-1})}{(p_k, Ap_k)};$$

$$x_k = x_{k-1} + \alpha_k p_k;$$

$$r_k = r_{k-1} - \alpha_k A p_k;$$

$$z_k = P_C r_k = L^{-T} L^{-1} r_k + \sigma Z E^{-1} Z^T r_k;$$

$$\beta_k = \frac{(r_k, z_k)}{(r_{k-1}, z_{k-1})}; \qquad p_{k+1} = z_k + \beta_k p_k;$$

end while

Properties of Deflation and CGC

$P_D = I - AZE^{-1}Z^T$ $\qquad$ $P_C = I + \sigma ZE^{-1}Z^T$

TU Delft

Properties of Deflation and CGC

$P_D = I - AZE^{-1}Z^T$ $\qquad$ $P_C = I + \sigma ZE^{-1}Z^T$

Properties of P_D

- $P_D A$ is symmetric and positive semidefinite
- P_D is a projection, $P_D AZ = 0$
- since $P_D A$ is singular, a good termination criterion is important

Properties of Deflation and CGC

$$P_D = I - AZE^{-1}Z^T \qquad P_C = I + \sigma ZE^{-1}Z^T$$

Properties of P_D

- $P_D A$ is symmetric and positive semidefinite
- P_D is a projection, $P_D AZ = 0$
- since $P_D A$ is singular, a good termination criterion is important

Properties of P_C

- P_C is symmetric positive definite
- $A^{\frac{1}{2}}(P_C - I)A^{\frac{1}{2}}$ is a projection

Properties of Deflation and CGC

Definition

Eigenpair $\{\lambda_i, v_i\}$, so $Av_i = \lambda_i v_i$ with $0 < \lambda_1 \leq \ldots \leq \lambda_n$.

Take $Z = [v_1 \ldots v_m]$.

TU Delft

Properties of Deflation and CGC

Definition

Eigenpair $\{\lambda_i, v_i\}$, so $Av_i = \lambda_i v_i$ with $0 < \lambda_1 \leq \ldots \leq \lambda_n$.

Take $Z = [v_1 \ldots v_m]$.

Theorem

- the spectrum of $P_D A$ is $\{0, \ldots, 0, \lambda_{m+1}, \ldots, \lambda_n\}$

- the spectrum of $P_C A$ is $\{\sigma + \lambda_1, \ldots, \sigma + \lambda_m, \lambda_{m+1}, \ldots, \lambda_n\}$

Properties of Deflation and CGC

Definition

Eigenpair $\{\lambda_i, v_i\}$, so $Av_i = \lambda_i v_i$ with $0 < \lambda_1 \leq \ldots \leq \lambda_n$.
Take $Z = [v_1 \ldots v_m]$.

Theorem

- the spectrum of $P_D A$ is $\{0, \ldots, 0, \lambda_{m+1}, \ldots, \lambda_n\}$

- the spectrum of $P_C A$ is $\{\sigma + \lambda_1, \ldots, \sigma + \lambda_m, \lambda_{m+1}, \ldots, \lambda_n\}$

Corollary

DICCG converges faster than CICCG if $Z = [v_1 \ldots v_m]$.

TU Delft

Deflation and Coarse Grid Correction (preliminaries)

Notation: A, B are Hermitian, $A \succeq B$, if $A - B$ is positive semidefinite

TU Delft

Deflation and Coarse Grid Correction (preliminaries)

Notation: A, B are Hermitian, $A \succeq B$, if $A - B$ is positive semidefinite

Some results from Horn and Johnson, Matrix Analysis

$$\lambda_k(A) + \lambda_1(B) \leq \lambda_k(A + B) \leq \lambda_k(A) + \lambda_n(B)$$

Deflation and Coarse Grid Correction (preliminaries)

Notation: A, B are Hermitian, $A \succeq B$, if $A - B$ is positive semidefinite

Some results from Horn and Johnson, Matrix Analysis

$$\lambda_k(A) + \lambda_1(B) \leq \lambda_k(A+B) \leq \lambda_k(A) + \lambda_n(B)$$

If A, B are positive definite with $A \succeq B$, then $\lambda_i(A) \geq \lambda_i(B)$.

Deflation and Coarse Grid Correction (preliminaries)

Notation: A, B are Hermitian, $A \succeq B$, if $A - B$ is positive semidefinite

Some results from Horn and Johnson, Matrix Analysis

$$\lambda_k(A) + \lambda_1(B) \leq \lambda_k(A+B) \leq \lambda_k(A) + \lambda_n(B)$$

If A, B are positive definite with $A \succeq B$, then $\lambda_i(A) \geq \lambda_i(B)$.

Suppose that B has rank at most m. Then

- $\lambda_k(A+B) \leq \lambda_{k+m}(A), \quad k = 1, 2, \cdots n-m,$
- $\lambda_k(A) \leq \lambda_{k+m}(A+B), \quad k = 1, 2, \cdots n-m.$

Deflation and Coarse Grid Correction (main result)

Theorem

Let A be symmetric positive definite and Z has rank $Z = m$. Let $E := Z^T AZ$. Then

$$\begin{aligned} \lambda_1(P_D A) = \cdots = \lambda_m(P_D A) &= 0 \\ \lambda_n(P_D A) &\leq \lambda_n(P_C A) \\ \lambda_{m+1}(P_D A) &\geq \lambda_1(P_C A) \end{aligned}$$

Deflation and Coarse Grid Correction (main result)

Theorem

Let A be symmetric positive definite and Z has rank $Z = m$. Let $E := Z^T AZ$. Then

$$\begin{aligned} \lambda_1(P_D A) = \cdots = \lambda_m(P_D A) &= 0 \\ \lambda_n(P_D A) &\leq \lambda_n(P_C A) \\ \lambda_{m+1}(P_D A) &\geq \lambda_1(P_C A) \end{aligned}$$

Theorem

$Z_1 \in \mathbb{R}^{n \times r}$, $Z_2 \in \mathbb{R}^{n \times s}$, $rank Z_1 = r$ and $rank Z_2 = s$. If $Im Z_1 \subseteq Im Z_2$, then

$$\begin{aligned} \lambda_n((I - AZ_1E_1^{-1}Z_1^T)A) &\geq \lambda_n((I - AZ_2E_2^{-1}Z_2^T)A) \\ \lambda_{r+1}((I - AZ_1E_1^{-1}Z_1^T)A) &\leq \lambda_{s+1}((I - AZ_2E_2^{-1}Z_2^T)A) \end{aligned}$$

Deflation and Coarse Grid Correction combined with a preconditioner

Definition

$$P_{CM^{-1}} := M^{-1} + \sigma Z E^{-1} Z^T.$$

Theorem

Let A and M be symmetric positive definite. Let $Z \in \mathbb{R}^{n \times m}$ with $rank Z = m$. Let $E := Z^T A Z$. Then

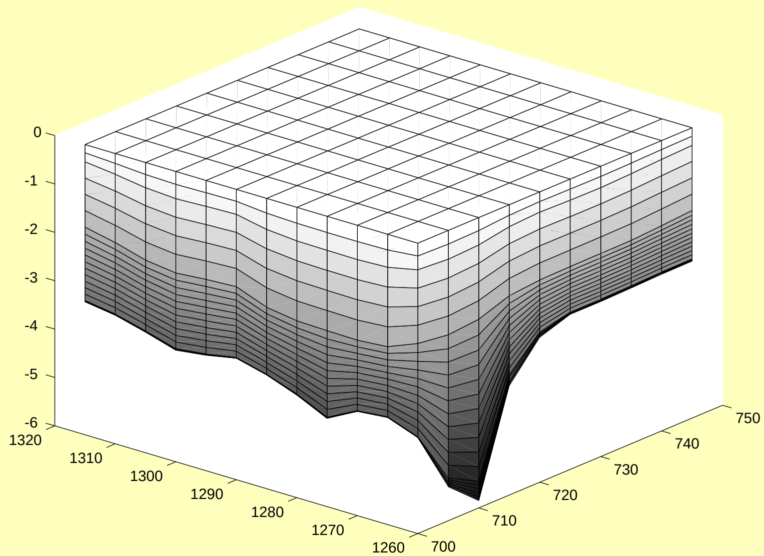
$$\begin{aligned} \lambda_n(M^{-1} P_D A) &\leq \lambda_n(P_{CM^{-1}} A), \\ \lambda_{m+1}(M^{-1} P_D A) &\geq \lambda_1(P_{CM^{-1}} A). \end{aligned}$$

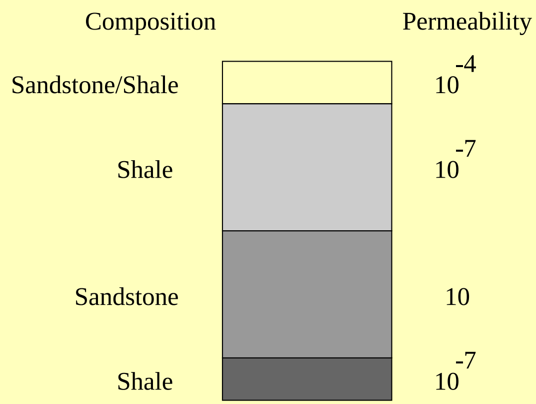
Corollary

DICCG converges faster than CICCG for general projection vectors.

4. Numerical experiments

Oil flow problem



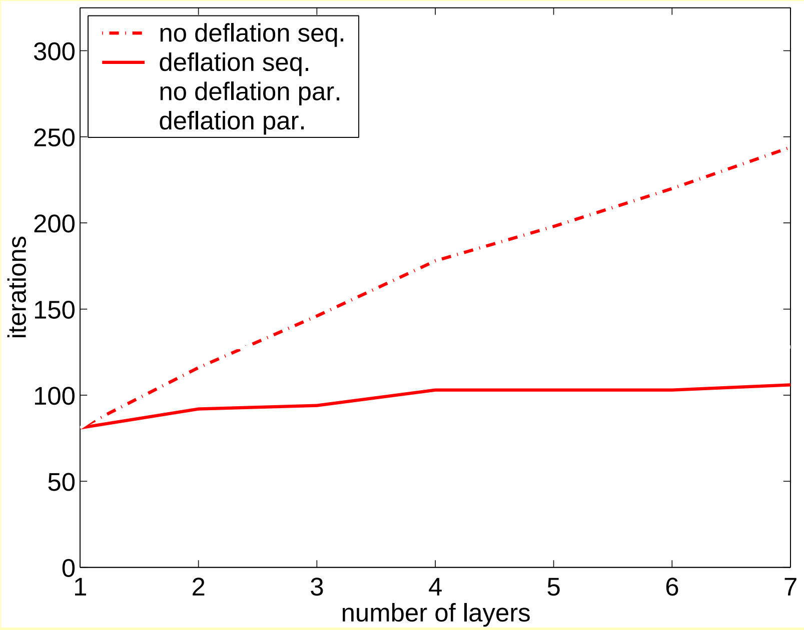


method	Deflation	CGC
iterations	36	47
CPU time	5.9	8.2

TU Delft

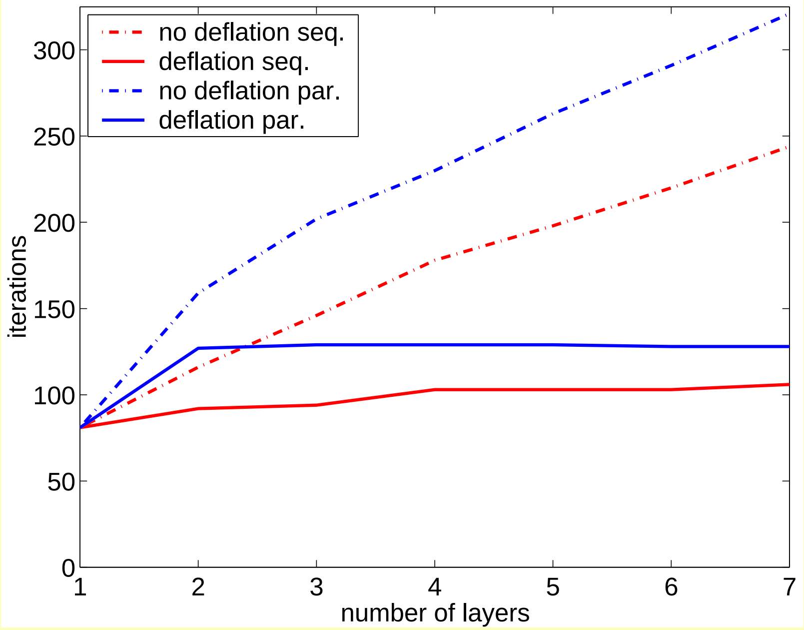
Poisson on parallel layers

Iterations



Poisson on parallel layers

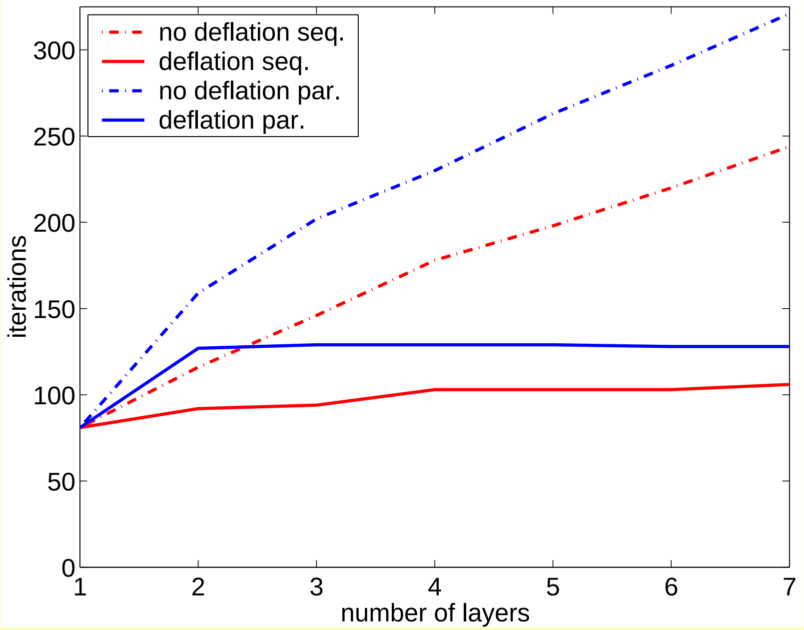
Iterations



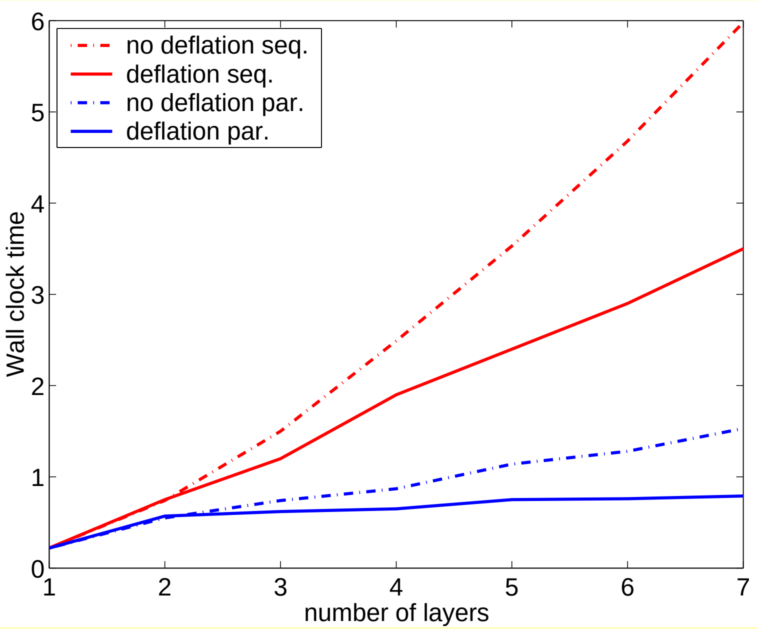
TU Delft

Poisson on parallel layers

Iterations

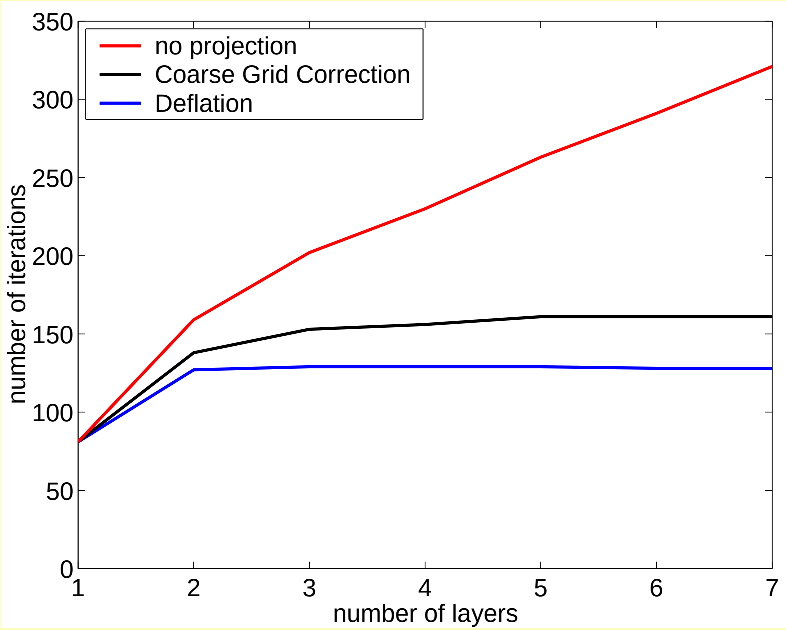


Wall clock time



Poisson on layers and blocks

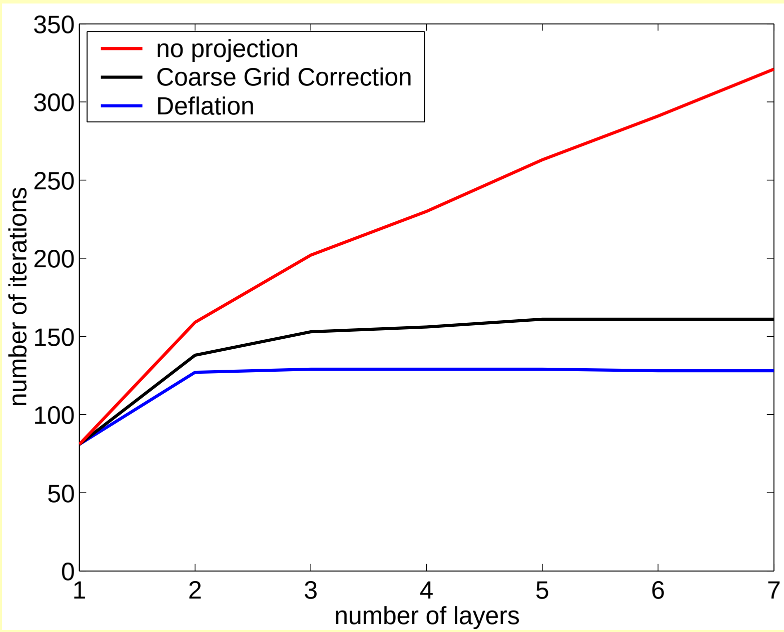
Layers



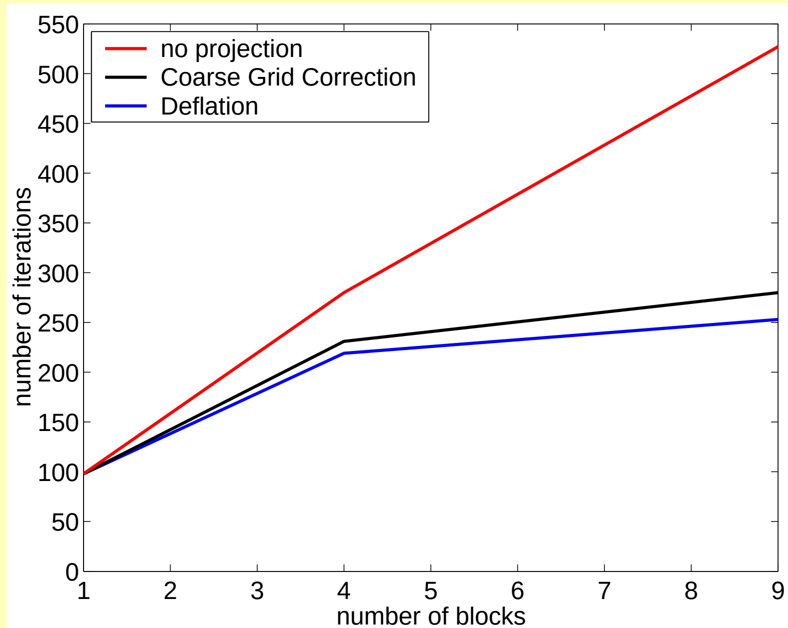
TU Delft

Poisson on layers and blocks

Layers



Blocks



5. Conclusions

- Block preconditioned Krylov methods combined with Deflation or CGC are well parallelizable (scalable, good speed up).
- For the vertex centered case, the weighted overlap strategy is optimal
- DICCG is more efficient than CICCG.
- Choices for the deflation vectors lead to comparable results in DICCG and CICCG.
- DICCG is a robust and efficient method to solve diffusion problems with discontinuous coefficients.

Further information

- http://ta.twi.tudelft.nl/nw/users/vuik/pub_it_def.html
- C. Vuik, A. Segal and J.A. Meijerink
  J. Comp. Phys., 152, pp. 385-403, 1999.
- J. Frank and C. Vuik
  SIAM Journal on Scientific Computing, 23, pp. 442–462, 2001
- C. Vuik, A. Segal, L. El Yaakoubi and E. Dufour
  Applied Numerical Mathematics, 41, pp. 219–233, 2002
- F.J. Vermolen, C. Vuik and A. Segal
  J. of Comp. Methods in Sciences and Engineering, to appear
- R. Nabben and C. Vuik
  A comparison of Deflation and Coarse Grid Correction, to appear

Overview

Krylov	Ar
Preconditioned Krylov	$L^{-T}L^{-1}Ar$
Block Preconditioned Krylov	$\sum_{i=1}^{m}(L_i^{-T}L_i^{-1})Ar$
Block Preconditioned Deflated Krylov	$\sum_{i=1}^{m}(L_i^{-T}L_i^{-1})PAr$