Robust Preconditioned Conjugate Gradient for the GPU and Parallel Implementations

Rohit Gupta, Martin van Gijzen, Kees Vuik

GPU Technology Conference 2012, San Jose CA.

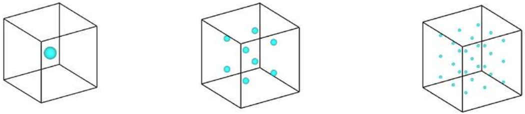
TU Delft

Delft
University of
Technology

Content

- Introduction
- Preconditioning
- Deflation
  1. Stripe-Wise Deflation
     - Implementation
     - Results and Observations
  2. Block-Wise Deflation
     - Implementation
     - Results and Observations
- Conclusions

Problem Description



Air bubbles rising in Water.

Mass-Conserving Level-Set method to solve the Navier Stokes equation. Marker function ϕ changes sign at interface.

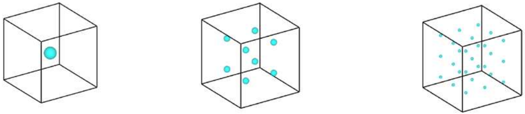
$$S(t) = \{x|\phi(x,t) = 0\}. \tag{1}$$

Interface is evolved using advection of Level-Set function

$$\frac{\partial \phi}{\partial t} + u.\bigtriangledown \phi = 0 \tag{2}$$

[1] A mass-conserving Level-Set method for modeling of multi-phase flows. S.P. van der Pijl, A. Segal and C. Vuik. International Journal for Numerical Methods in Fluids 2005; 47:339–361

Problem Description



Air bubbles rising in Water.

$$-\nabla.(\frac{1}{\rho(x)}\nabla p(x)) = f(x),\ x \in \Omega \quad (1)$$

$$\frac{\partial}{\partial n}p(x) = g(x),\ x \in \partial\Omega \quad (2)$$

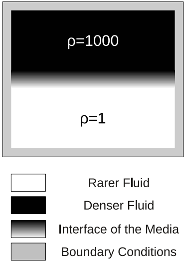
- Pressure-Correction (above) equation is discretized to a linear system $Ax = b$.
- Most time consuming part is the solution of this linear system
- A is Symmetric Positive-Definite (SPD) so Conjugate Gradient is the method of choice.

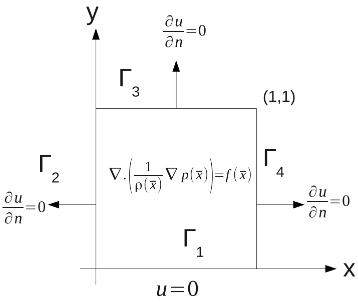
Two definitions

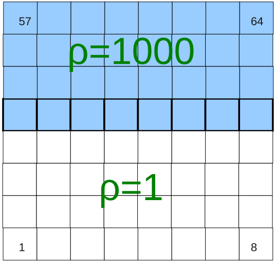
- Condition Number $\rightarrow \kappa(A) := \frac{\lambda_n}{\lambda_1}$
- Stopping Criteria $\rightarrow \frac{\|b - Ax_k\|_2}{\|r_0\|} \leq \epsilon$

1. ϵ is the tolerance we set for the solution.
2. $\lambda_n \geq \lambda_{n-1} \cdots \geq \lambda_2 \geq \lambda_1$. λ's are the eigenvalues of A.
3. x_k is the solution vector after k iterations of (P)CG.
4. r_0 is the initial residual.

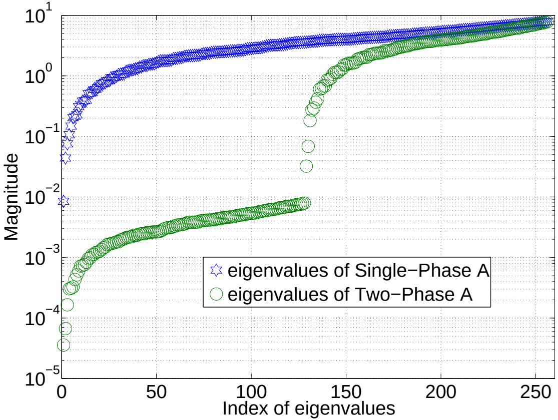
Test Problem - Computational Domain







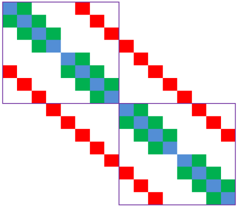
Spectrum of the Coefficient Matrix (semilog scale).



Huge jump at Interface due to contrast in densities leads to ill-conditioned *A*.

Preconditioning

Block Incomplete Cholesky Preconditioning [1] [2]



Within blocks the computation is sequential.

[1] An Iterative Solution Method for Linear Systems of Which the Coefficient Matrix is a Symmetric M-Matrix. J.A. Meijerink, H.A.van der Vorst (1977).Math. Comp. (American Mathematical Society).

[2] Iterative methods for sparse linear systems. 2nd ed., Society for Industrial and Applied Mathematics, Philadelphia, 2003. Yousef Saad.

Preconditioning

Truncated Neumann Series Preconditioning[1][2]

$$M^{-1} = K^T K, \text{ where } K = (I - \tilde{L} + \tilde{L}^2 + \cdots). \tag{3}$$

1. More terms give better approximation.
2. In general the series converges if $\| \tilde{L} \|_\infty < 1$.
3. As much parallelism on offer as Sparse Matrix Vector Product.

$\tilde{L}$ is the strictly lower triangular of $\tilde{A}$,
where $\tilde{A} = D^{\frac{-1}{2}} A D^{\frac{-1}{2}}$ and $D = diag(A)$.

[1] A vectorizable variant of some ICCG methods. Henk A. van der Vorst. SIAM Journal of Scientific Computing. Vol. 3 No. 3 September 1982.

[2] Approximating the Inverse of a Matrix for use in Iterative Algorithms on Vector Processors. P.F. Dubois. Computing (22) 1979.

Deflation

Background.

Removes small eigenvalues from the eigenvalue spectrum of A.
The linear system $Ax = b$ can then be solved by employing the splitting,

$$x = (I - P^T)x + P^T x \text{ where } P = I - AQ. \quad (4)$$

$$\Leftrightarrow Pb = PA\hat{x}. \quad (5)$$

$Q = ZE^{-1}Z^T$, $E = Z^T AZ$.
E is the coarse system that is solved every iteration.
Z is the deflation sub-space matrix. It contains an approximation of the eigenvectors of $M^{-1}A$.

For our experiments Z consists of piecewise constant vectors.

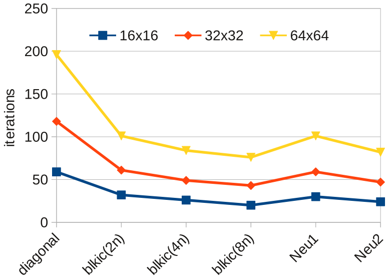
Deflation

Deflated Preconditioned Conjugate Gradient Algorithm

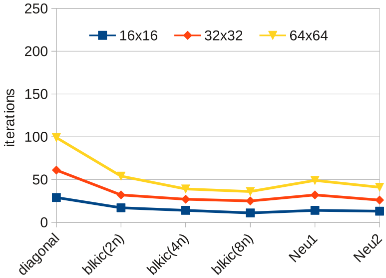
Select x_0. Compute $r_0 := b - Ax_0$ and $\hat{r}_0 = Pr_0$, Solve $My_0 = \hat{r}_0$ and set $p_0 := y_0$.
for j:=0,..., until convergence **do**
  $\hat{w}_j := PAp_j$
  $\alpha_j := \frac{(\hat{r}_j, y_j)}{(p_j, \hat{w}_j)}$
  $\hat{x}_{j+1} := \hat{x}_j + \alpha_j p_j$
  $\hat{r}_{j+1} := \hat{r}_j - \alpha_j \hat{w}_j$
  Solve $My_{j+1} = \hat{r}_{j+1}$
  $\beta_j := \frac{(\hat{r}_{j+1}, y_{j+1})}{(\hat{r}_j, y_j)}$
  $p_{j+1} := y_{j+1} + \beta_j p_j$
end for
$x_{it} := Qb + P^T x_{j+1}$

Effect of Deflation

Convergence[1].



with preconditioning only.



with preconditioning and deflation.

[1] Conjugate Gradient. Deflation vectors are $2n$. Precision Criteria $1e-6$.

Implementation - PCG

1. Finite Difference discretization leads to $5/7$ point stencil for a $2/3$D grid.
2. Diagonal Format of storage for the coefficient matrix A.
3. Preconditoning and SpMV (Sparse Matrix Vector) Products take the bulk of time.
4. CUBLAS/CUSP libraries for efficient implementation.

Implementation - Preconditioning

Truncated Neumann Series Preconditioning

Two variants of Neumann Preconditioning have been tried:-
$K = (I - \tilde{L})$ or $K = (I - \tilde{L} + \tilde{L}^2)$
$y = M^{-1}r$, where $M = K^T K$, is implemented as

1. $s = (I - \tilde{L})x$,
2. $y = (I - \tilde{L}^T)s$.

- Only $\tilde{L}$ is stored.
- Degree of Parallelism $=$ Problem Size, $N = n \times n$.

Implementation - Deflation

Operations involved in deflation[1] [2].

- $a1 = Z^T p$.
- $m = E^{-1} a1$.
- $a2 = AZm$.
- $\hat{w} = p - a2$.

where, $E = Z^T AZ$ is the Galerkin Matrix and Z is the matrix of deflation vectors.

[1] Efficient deflation methods applied to 3-D bubbly flow problems. J.M. Tang, C. Vuik Elec. Trans. Numer. Anal. 2007.

[2] An efficient preconditioned CG method for the solution of a class of layered problems with extreme contrasts in the coefficients. C. Vuik, A. Segal, J.A. Meijerink J. Comput. Phys. 1999.

Implementation - Deflation

Choices for solving inner system.

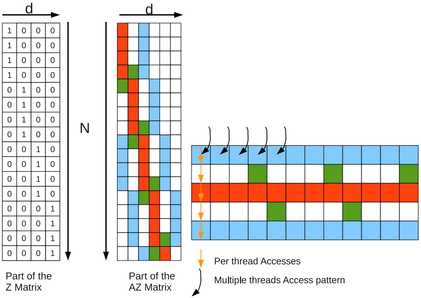
- $a1 = Z^T p$.
- $m = E^{-1} a1$.
- $a2 = AZm$.
- $\hat{w} = p - a2$.

Two choices for step 2.

◯ Explicit Inverse Calculation ◯ Triangular Solve

Implementation - Deflation

Storage of Z and of AZ.



Z Matrix has this structure due to stripe-wise domains.

For AZ the aforementioned data structure has the advantages of the DIA Storage format[1].

[1] Efficient Sparse Matrix-Vector Multiplication on CUDA. N. Bell and M. Garland, 2008 , NVIDIA Corporation, NVR-2008-04

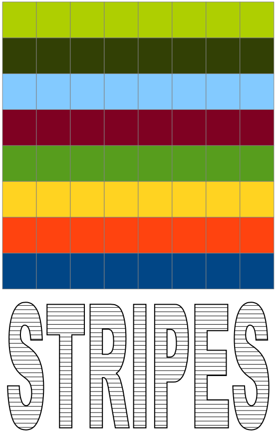
Implementation - Deflation

Choices for deflation vectors - I

1. Stripe-wise Vectors
2. Block Deflation Vectors
3. Level-Set based vectors

Implementation - Deflation

Choices for deflation vectors - II



Results - Test Problem - One Interface

Hardware

1. CPU - single core of Q9550-2.83 GHz.
2. GPU - Tesla C2070.

Timing and Speedup Definition

Speedup is measured as a ratio of the time taken(T) to complete k iterations (of the DPCG method) on the two different architectures,

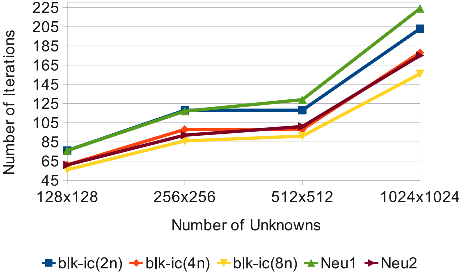
$$Speedup = \frac{T_{CPU}}{T_{GPU}} \quad (4)$$

Z is chosen piecewise constant. At least $2n$ deflation vectors are chosen for problem size $N = n \times n$.

Results

Test Problem - Convergence comparison.

Block-IC vs. Truncated Neumann[1 2].



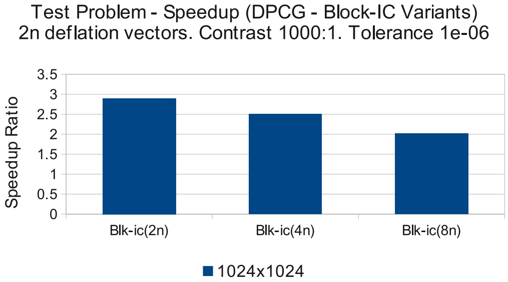
[1] Conjugate Gradient. Deflation vectors are $2n$. Precision Criteria $1e-6$. Contrast $1000:1$.

[2] blk-ic(2n) means Block-Incomplete Cholesky with **block-size**$=2n$.

Results

Test Problem - SpeedUp.

Setup Time *excluded*.

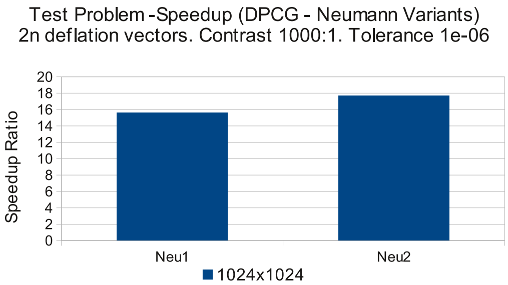


Small Speed-Up

Results

Test Problem - SpeedUp.

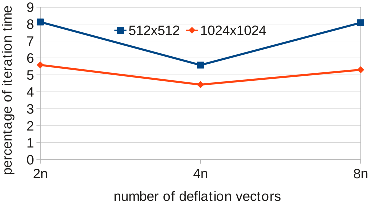
Setup Time *excluded*.



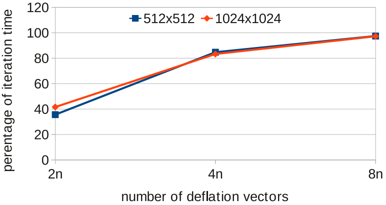
Large Speed-Up

Results

Test Problem -Deflation Solution Method Comparison -CPU. Setup Times *Only*.



Triangular Solve

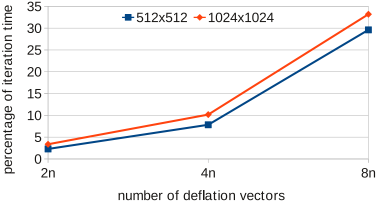


Explicit Inverse

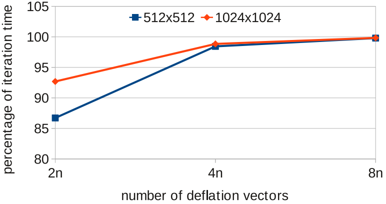
1. Both these results are for a CPU Implementation of Deflation.
2. Two different Grid Sizes are compared.

Results

Test Problem -Deflation Solution Method Comparison -GPU. Setup Times *Only*.



Triangular Solve



Explicit Inverse

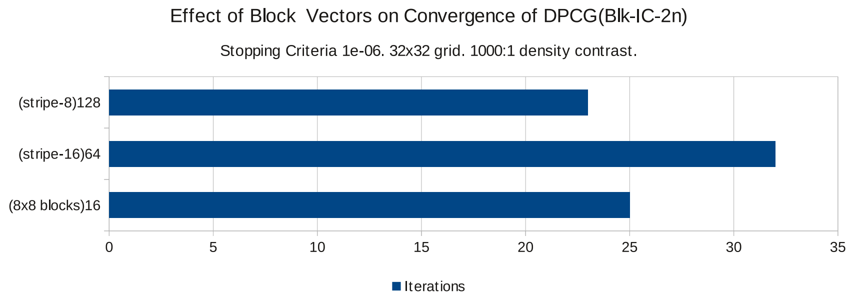
1. Both these results are for a GPU Implementation of Deflation.
2. Two different Grid Sizes are compared.
3. Size of Matrix that has to be inverted is $2n \times 2n$, where $N = n \times n$.

Lessons Learnt

1. Using Explicit Inverse Calculation gives better speedup but setup times are prohibitively large.
2. Fewer Deflation vectors can make inversion cheaper.
3. More effective (but less in number) deflation vectors must be chosen.

Results

Test Problem - Effect of Deflation vector choices.

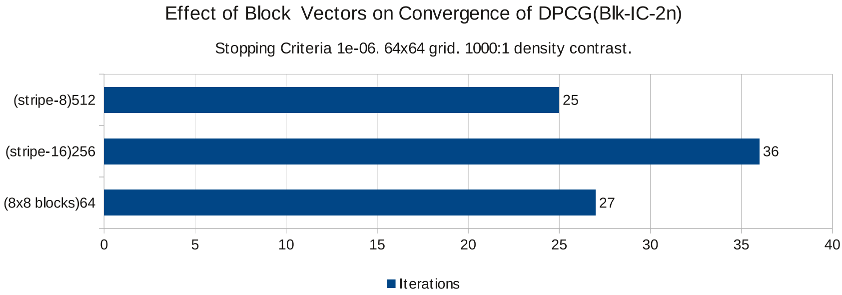


Experiments done in Matlab. DPCG with Block Incomplete Cholesky preconditioning (block size - 2n).

Block-Vectors are as effective at 1/8th the size of stripe-wise vectors.

Results

Test Problem - Effect of Deflation vector choices.



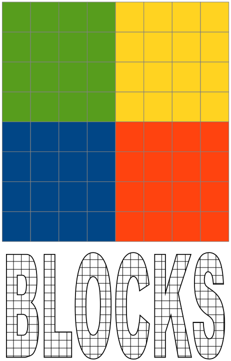
Experiments done in Matlab. DPCG with Block Incomplete Cholesky preconditioning (block size - 2n).
Block-Vectors are as effective at 1/8th the size of stripe-wise vectors.

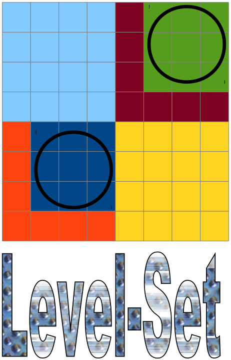
Lessons Learnt

1. Using Explicit Inverse Calculation gives better speedup but setup times are prohibitively large.
2. Fewer Deflation vectors can make inversion cheaper.
3. More effective (but less in number) deflation vectors must be chosen.

What if block vectors (or Level-Set) could be used with explicit inverse based deflation?

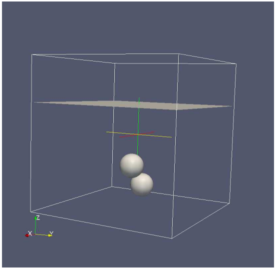
Choices for deflation vectors - Blocks and Level-Set





Realistic Problem

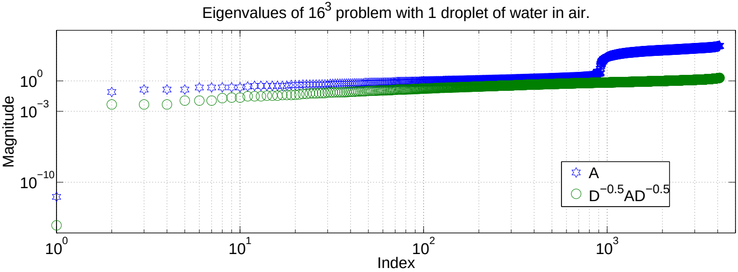
Computational Domain



1. Neumann boundary condition on all sides for pressure.
2. Density is calculated using the Level-Set Approach.
3. Density Contrast is $1e \pm 03$. Stopping Criteria is $1e - 06$.
4. Problem is defined over a unit cube.

Realistic Problem

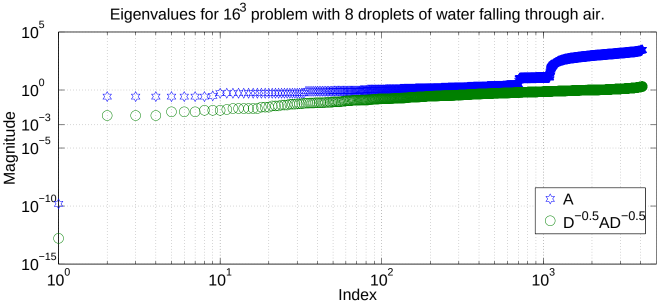
Distribution of Eigenvalues (loglog Scale).



D is the diagonal of A.

Realistic Problem

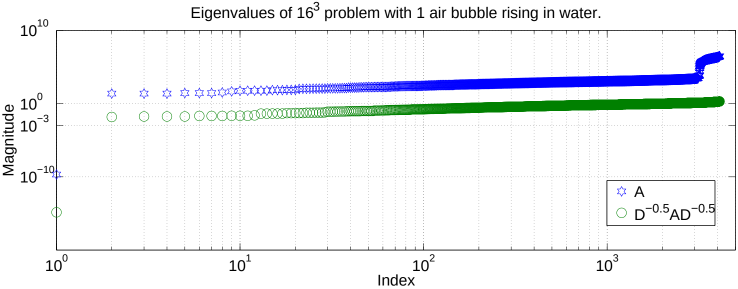
Distribution of Eigenvalues (loglog Scale).



D is the diagonal of *A*.

Realistic Problem

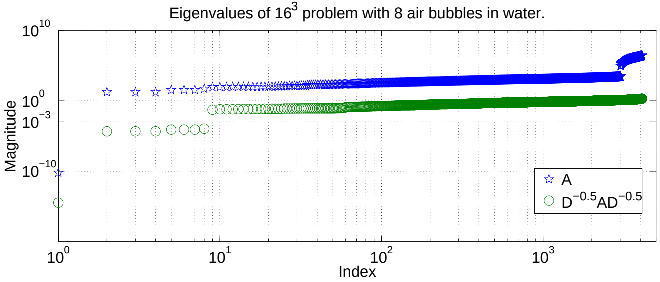
Distribution of Eigenvalues (loglog Scale).



D is the diagonal of *A*.

Realistic Problem

Distribution of Eigenvalues (loglog Scale).

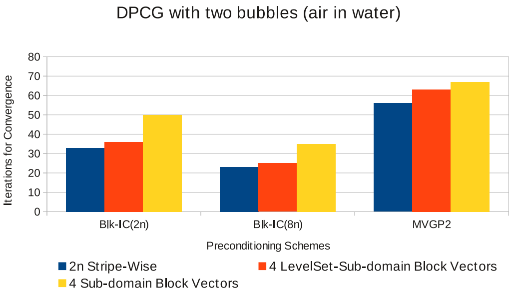


7 eigenvalues of the order of the density contrast in addition to 1 zero eigenvalue due to the density contrast.

D is the diagonal of *A*.

Changes in Implementation.

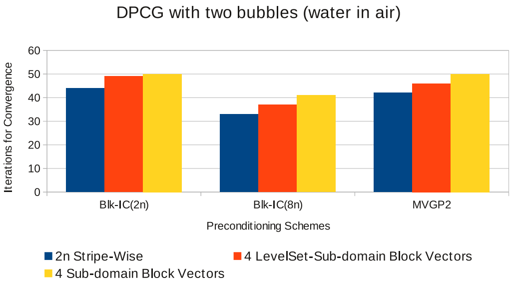
Stripe-wise vectors do not give good results for bubbles in 3D problem as well.



2n=Stripe-wise 2n vectors, LS-4blk=Level-Set vectors combined with block-shaped sub-domain vectors,
S=Sub-Domain Vectors(blocks)
$32x32$ grid with two bubbles and density contrast $1000 : 1$. Air bubbles in water.
A move to Improved (block and level-set) vectors is required.

Changes in Implementation.

Stripe-wise vectors do not give good results for bubbles in 3D problem as well.



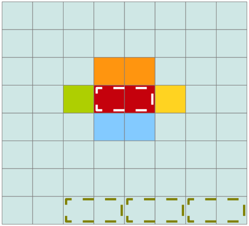
2n=Stripe-wise 2n vectors, LS-4blk=Level-Set vectors combined with block-shaped sub-domain vectors, S=Sub-Domain Vectors(blocks)
$32x32$ grid with two bubbles and density contrast $1 : 1000$. Water droplets in air.

A move to Improved (block and level-set) vectors is required.

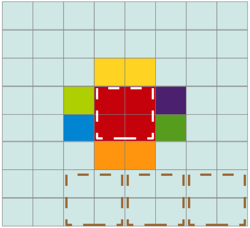
Changes in Implementation.

Storage for AZ must be changed

AZ could require more storage or difficult to store as before.



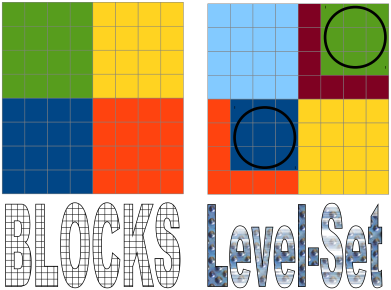
Stripe-Wise Case



Block-Wise Case

Changes in Implementation.

- Block Structure of deflation vectors requires more diagonals.
- Level-Set based deflation vectors would leave non-zeros in AZ at unpredictable locations.

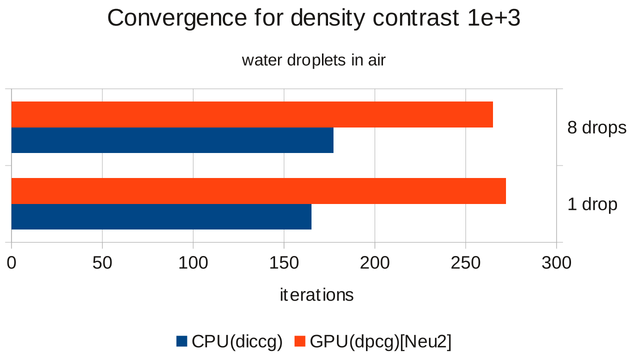


Changes in Implementation.

Using CUSP to store AZ in HYB format.
Any kind of vectors can be chosen in Z.
Multiplication of A with Z also handled by CUSP (done on device).

Results

Realistic Problem - 3D

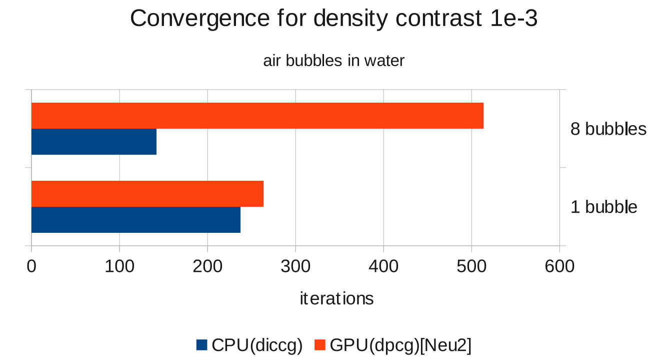


Results for a 128^3 grid for density contrast $1e\pm3$. Stopping Criteria $1e-06$.
8 Block-wise deflation vectors were used for Z.
DICCG refers to Deflated Incomplete Cholesky (preconditioned) CG.

DPCG(Neu2) refers to Deflated Neumann (variant2) (preconditioned) CG.

Results

Realistic Problem - 3D

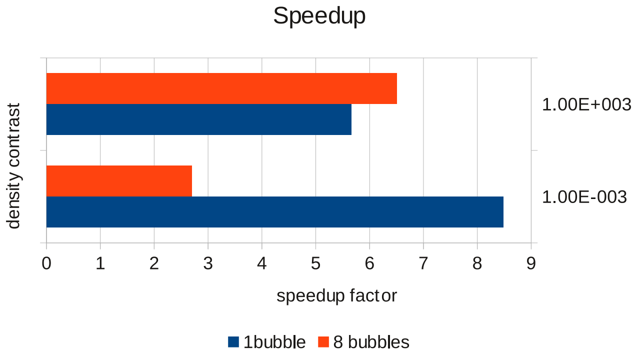


Results for a 128^3 grid for density contrast $1e{\pm}3$. Stopping Criteria $1e-06$.
8 Block-wise deflation vectors were used for Z.
DICCG refers to Deflated Incomplete Cholesky (preconditioned) CG.

DPCG(Neu2) refers to Deflated Neumann (variant2) (preconditioned) CG.

Results

Realistic Problem - 3D



Results for a 128^3 grid for density contrast $1e{\pm}3$. Stopping Criteria $1e-06$.
8 Block-wise deflation vectors were used for Z.
DICCG refers to Deflated Incomplete Cholesky (preconditioned) CG.

DPCG(Neu2) refers to Deflated Neumann (variant2) (preconditioned) CG.

Lessons Learnt

The Multiple Bubble Problem.

1. For the air bubbles in water problem Neumann Preconditioning doesn’t work well.
2. 1 is true only for multiple bubbles. For single bubble Neumann works comparable to IC.
3. Neumann works reasonably for water droplets in air, since there are no eigenvalues of the order of the density contrast.

Move scaling out of the Preconditioner

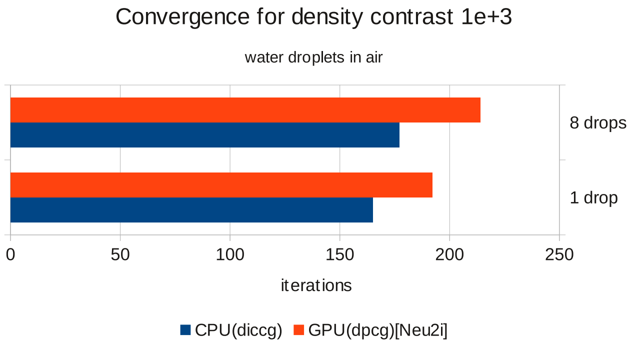
We re-define the Neumann2 Truncation based preconditioner as follows:

$$M^{-1} = K^T D^{-1} K, \text{ where } K = (I - LD^{-1} + (LD^{-1})^2 + \cdots). \qquad (5)$$

Please note that now we use L instead of $\tilde{L}$ and also A, x and b are NOT scaled.

Results

Realistic Problem - 3D - with Neumann corrected

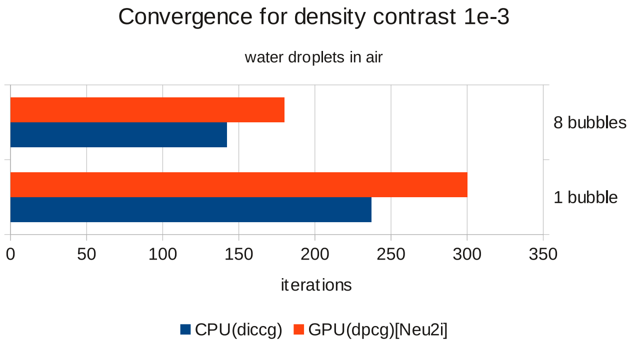


Results for a 128^3 grid for density contrast $1e{\pm}3$. Stopping Criteria $1e-06$.
8 Block-wise deflation vectors were used for Z.
DICCG refers to Deflated Incomplete Cholesky (preconditioned) CG.

DPCG(Neu2i) refers to Deflated Neumann (variant2) (preconditioned) CG and scaling only in the preconditioner not in A, x and b.

Results

Realistic Problem - 3D - with Neumann corrected

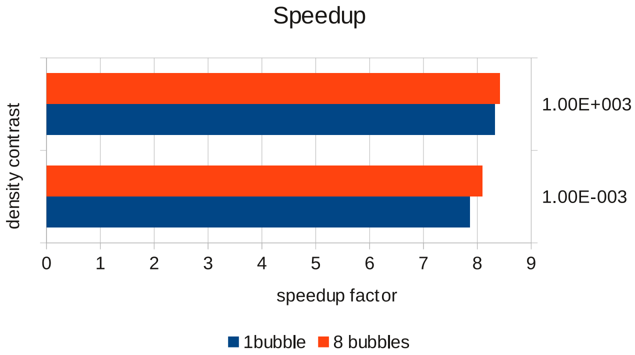


Results for a 128^3 grid for density contrast $1e\pm3$. Stopping Criteria $1e-06$.
8 Block-wise deflation vectors were used for Z.
DICCG refers to Deflated Incomplete Cholesky (preconditioned) CG.

DPCG(Neu2i) refers to Deflated Neumann (variant2) (preconditioned) CG and scaling only in the preconditioner not in A, x and b.

Results

Realistic Problem - 3D - with Neumann corrected



Results for a 128^3 grid for density contrast $1e\pm3$. Stopping Criteria $1e-06$.
8 Block-wise deflation vectors were used for Z.
DICCG refers to Deflated Incomplete Cholesky (preconditioned) CG.

DPCG(Neu2i) refers to Deflated Neumann (variant2) (preconditioned) CG and scaling only in the preconditioner not in A, x and b.

Results

Bubbly Flow Simulations.

- Overall speedup of the order one-half of speedup for stationary case.(this changed one day back, thanks to Thomas Bradley[1])
- Program structure must be improved to avoid all/de-allocation of memory every time step.

[1] Using nvidia-smi with the persistent load option for the driver makes the initial time for loading the driver disappear.:)

References

- Efficient Two-Level Preconditioned Conjugate Gradient Method on the GPU. Report 11-15.`bit.ly/qMNHd4`
- Masters Thesis of Rohit Gupta. `bit.ly/pB5ilK`
- GPU page of our Department.`http://ta.twi.tudelft.nl/users/vuik/gpu.html`

Comments/Suggestions/Questions

Theory is when you know everything but nothing works.
Practice is when everything works but no one knows why.
In reality theory and practice go hand in hand: Nothing usually works[1] and no one knows why[2] :)[3]

---

[1]*Existence requirement for Research.
[2]*Existence requirement for Research Candidate.
[3]Ergo! Another PhD project brought to life.

Preconditioning - IP[1]

$$M^{-1} = KK^T, \text{ where } K = (I - LD^{-1}). \tag{6}$$

$$\text{Stencil for A } = (-1, -1, 4, -1, -1). \tag{7}$$

$$\text{Corresponding Stencil for } M^{-1} = (\frac{1}{4}, \frac{1}{16}, \frac{1}{4}, \frac{9}{8}, \frac{1}{4}, \frac{1}{16}, \frac{1}{4}) \tag{8}$$

1. Drop the lowest terms (i.e.$\frac{1}{16}$).

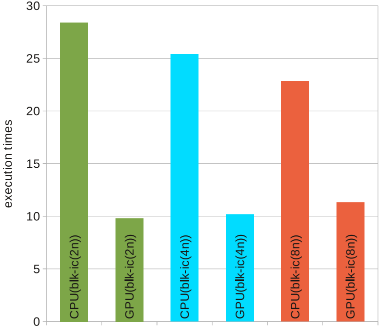
2. M^{-1} has the same sparsity pattern as A.

Degree of Parallelism for $M^{-1}r$ is N.

[1] A Parallel Preconditioned Conjugate Gradient Solver for the Poisson Problem on a Multi-GPU Platform, M. Ament. PDP 2010

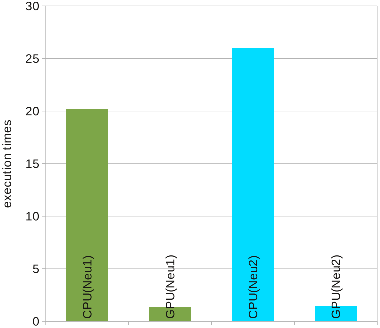
Wall Clock Times Comparison.

Block-IC vs. Truncated Neumann.



Block-IC variants.

Small Speed-Up.



Truncated Neumann variants.

Large Speed-Up.