Parallel block-diagonal preconditioner with projectors for diffusion equation

Vasily Kramarenko [1] Yuri Kuznetsov [2] Igor Konshin [1,3]

[1]Marchuk Institute of Numerical Mathematics of Russian Academy of Science

[2]University of Houston

[3]Dorodnicyn Computing Centre of FRC CSC RAS

24 September, 2018

Outline

1. problem formulation
2. discretization
3. preconditioner construction
4. preconditioning process
5. comparison results

Problem formulation

Classical 2nd order equation

$$
\begin{aligned}
-\nabla\left(D\nabla\, p\right) &= f \quad \text{in } \Omega,\\
p &= p_0 \quad \text{on } \partial\Omega,
\end{aligned}
$$

$D = d_0 I$ in Ω.

In area Ω m subareas are selected. These subareas corresponds to inclusions.

$\partial\omega_s \cap \partial\omega_t = \varnothing$, $\partial\omega_s \cap \partial\Omega = \varnothing$, $s, t = 1, \dots, m$.

In inclusions define diffusion coefficient d_s, $d_s \gg d_0$

Example of Ω with inclusions

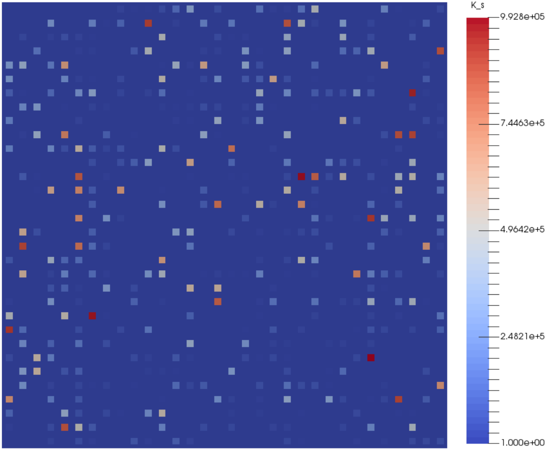


Figure: Area Ω with inclusions

Discretization

For discretization P1 finite element method is used. In inclusion

$$A_s = d_s A_{\lambda,s} = A_{\lambda,s} + (d_s - d_0)\, A_{\lambda,s}$$

As a result matrix $A\overline{p} = \overline{g}$ is a sum

$$A = A_0 + A_1$$

where A_0 is matrix "without" inclusions and

$$A_1 = \sum_{s=1}^{m} (d_s - 1)\, N_s A_{\lambda,s} N_s^T, \tag{1}$$

Generalized Eigenvalue problem

Consider generalized eigenvalue problem

$$A_t\overline{w} = \lambda M_t\overline{w}$$

where $0 = \lambda_{t,1} < \lambda_{t,2} \leqslant \ldots \leqslant \lambda_{t,m_t}$ are eigenvalues, M_t is mass matrix.
We can construct decomposition.

$$A_t = M_t W_t \Lambda_t W_t^T M_t$$

where $\Lambda_t = diag\{\lambda_{t,1}, \lambda_{t,2}, \ldots, \lambda_{t,m_t}\}$ and $W_t = [\overline{w}_{t,1}, \ldots, \overline{w}_{t,m_t}]$,
If we consider diffusion matrix from our case $\lambda_{t,1} = 0$,
$\overline{w}_{t,1} = \frac{1}{\|\overline{e}_t\|}_{M_t}\overline{e}_t$

Preconditioner Construstion

Consider the preconditioner, based on following idea. For each subdomain

$$B_t = \alpha_t M_t W_t \left(I_t - \hat{I}_t \right) W_t^T M_t$$

where

$$\hat{I}_t = \begin{pmatrix} 0 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \dots & & & \\ 0 & 0 & \dots & 1 \end{pmatrix}$$

Remark:

$$W_t^T M_t W_t = M_t W_t W_t^T = W_t W_t^T M_t = I_t$$

After using this trick

$$B_t = \alpha_t \left(M_t - M_t \overline{w}_{t,1} \overline{w}_{t,1}^T M_t \right)$$

Kuznetsov Y., Kramarenko V., Preconditioners with projectors for mixed hybrid finite element methods // Russian Journal of Numerical Analysis and Mathematical Modelling, Vol. 32 (2017), No. 1, P. 39–45.
DOI: 10.1515/rnam-2017-0004.

Preconditioner Construstion(2)

If $A_1 = \sum_{t=1}^{m} N_t A_t N_t^T$ we will construct preconditioner for second part of A

$$B_1 = \sum_{t=1}^{m} N_t B_t N_t^T$$

where $\alpha_t = (d_t - d_0)\|M_t^{-1} A_t\|$.
For other part of matrix we use diagonal preconditioner

$$B_0 = \alpha_0 M_0$$

where $\alpha_0 = \|M^{-1} A_0\|$ and as a result

$$B = B_0 + B_1$$

Preconditioning process (Coarse mesh system)

In ours case

$$B = \sum_t \alpha_t N_t \left(M_t - M_t \overline{w}_{t,1} \overline{w}_{t,1}^T M_t \right) N_t^T + B_0$$

$$B = M - \sum_t \alpha_t N_t M_t \overline{w}_{t,1} \overline{w}_{t,1}^T M_t N_t^T$$

Consider $Bv = g$

$$Mv - \sum_t \alpha_t N_t M_t \overline{w}_{t,1} \overline{w}_{t,1}^T M_t N_t^T v = g$$

or

$$v - \sum_t \alpha_t M^{-1} N_t M_t \overline{w}_{t,1} \overline{w}_{t,1}^T M_t N_t^T v = M^{-1} g$$

Let $\xi_t = \overline{w}_{t,1}^T M_t N_t^T v$

$$v - \sum_t \alpha_t M^{-1} N_t M_t \overline{w}_{t,1} \xi_t = M^{-1} g$$

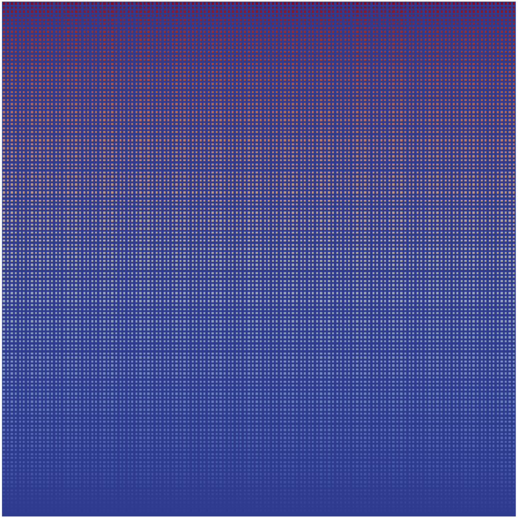
For calculation ξ_t we should solve system $(I - Q)\,\overline{\xi}_t = \hat{g}$, where $Q = \{q_{ij}\}$, $\hat{g} = \{\hat{g}\}$

$$q_{ij} = \overline{w}_{t,1}^T M_t N_t^T \alpha_t M^{-1} N_t M_t \overline{w}_{t,1}$$

$$\hat{g}_t = \overline{w}_{t,1}^T M_t N_t^T g$$

For our case Q is a diagonal matrix

We consider unit square with regular mesh



with zero Dirichlet boundary conditions.

Methods for comparison

The BiCGStab method is used for solving system.
For preconditioning four methods are used

1. ILU(k) $k = 7$ (from PETSc)
2. AMG (from Trilinos)
3. ILU2(τ) $\tau = 10^{-3}$ (from INMOST)
4. BDP

Comparison results

preconditioner		#iter	Time	T_{it}
PETSc ILU(7)	$d_s = 10^4$	125	0.99	0.00792
Trilinos AMG		122	1.66	0.01360
INMOST ILU2(10^{-3})		131	1.93	0.01581
BDP		544	2.16	0.00397
PETSc ILU(7)	$d_s \in [1; 10^4]$	126	0.99	0.00785
Trilinos AMG		85	1.15	0.01352
INMOST ILU2(10^{-3})		86	1.80	0.02093
BDP		644	2.43	0.00377
PETSc ILU(7)	$d_s = 10^6$	120	0.94	0.00783
Trilinos AMG		750	9.71	0.01294
INMOST ILU2(10^{-3})		279	2.42	0.00867
BDP		528	1.98	0.00375
PETSc ILU(7)	$d_s \in [1; 10^6]$	134	1.04	0.00776
Trilinos AMG		551	7.14	0.01295
INMOST ILU2(10^{-3})		133	1.99	0.01496
BDP		472	1.77	0.00375

Table: Preconditioners comparison. Inclusions number $m = 64 \times 64$, inclusion size $n = 4 \times 4$,matrix size $N_A = 261121$.

Comparison results

preconditioner		#iter	Time	T_{it}
PETSc ILU(7)	$d_s = 10^4$	206	6.34	0.030777
Trilinos AMG		146	11.26	0.07712
INMOST ILU2(10^{-3})		242	11.08	0.04578
BDP		1122	17.53	0.01562
PETSc ILU(7)	$d_s \in [1; 10^4]$	218	6.71	0.03082
Trilinos AMG		111	8.57	0.07720
INMOST ILU2(10^{-3})		149	10.78	0.07234
BDP		1445	21.92	0.01516
PETSc ILU(7)	$d_s = 10^6$	205	6.37	0.03107
Trilinos AMG		663	50.04	0.07547
INMOST ILU2(10^{-3})		338	10.98	0.03248
BDP		866	13.13	0.01516
PETSc ILU(7)	$d_s \in [1; 10^6]$	183	5.71	0.03120
Trilinos AMG		891	67.93	0.07624
INMOST ILU2(10^{-3})		355	19.45	0.05478
BDP		1336	20.48	0.01532

Table: Preconditioners comparison. Inclusions number $m = 128 \times 128$, inclusion size $n = 4 \times 4$, matrix size $N_A = 1046529$.

Comparison results

preconditioner		#iter	Time	T_{it}
PETSc ILU(7)	$d_s = 10^4$	565	17.10	0.03026
Trilinos AMG		88	6.87	0.07806
INMOST ILU2(10^{-3})		191	11.56	0.06052
BDP		1016	15.00	0.01476
PETSc ILU(7)	$d_s \in [1; 10^4]$	1177	35.2	0.029905
Trilinos AMG		71	5.58	0.07859
INMOST ILU2(10^{-3})		126	10.49	0.08325 0
BDP		1262	18.97	0.01503
PETSc ILU(7)	$d_s = 10^6$	1982	59.12	0.02982
Trilinos AMG		483	36.63	0.07583
INMOST ILU2(10^{-3})		1094	53.49	0.04889
BDP		966	14.10	0.01459
PETSc ILU(7)	$d_s \in [1; 10^6]$	1223	36.63	0.02995
Trilinos AMG		513	39.08	0.07617
INMOST ILU2(10^{-3})		299	21.29	0.07120
BDP		1152	17.15	0.01488

Table: Preconditioners comparison. Inclusions number $m = 64 \times 64$, inclusion size $n = 8 \times 8$, matrix size $N_A = 1046529$.

Comparison results

preconditioner		#iter	Time	T_{it}
PETSc ILU(7)	$d_s = 10^4$	3641	449.99	0.12358
Trilinos AMG		92	34.41	0.37402
INMOST ILU2(10^{-3})		322	74.85	0.2324
BDP		2609	165.9	0.06358
PETSc ILU(7)	$d_s \in [1; 10^4]$	1517	189.21	0.12472
Trilinos AMG		76	28.57	0.37592
INMOST ILU2(10^{-3})		206	63.98	0.31058
BDP		2996	192.85	0.06436
PETSc ILU(7)	$d_s = 10^6$	3996	495.52	0.12400
Trilinos AMG		518	189.17	0.36519
INMOST ILU2(10^{-3})		1441	278.21	0.19306
BDP		1839	117.47	0.06387
PETSc ILU(7)	$d_s \in [1; 10^6]$	3871	368.87	0.09529
Trilinos AMG		544	105.23	0.19343
INMOST ILU2(10^{-3})		492	80.80	0.16422
BDP		2539	76.18	0.0300

Table: Preconditioners comparison. Inclusion number $m = 128 \times 128$, inclusion size $n = 8 \times 8$, matrix size $N_A = 4190209$.

Acceleration results

	N_{proc}	1	2	4	8	16	32	64
PETSc ILU(7)	#iter	10000	10000	10000	10000	10000	4383	10000
	Time	1137.28	536.83	301.08	198.38	190.04	42.83	56.99
	Accur.	2.0e-09	6.0e-10	5.0e-02	2.0e-09	4.0e-10	1.0e-09	3.0e-09
Trilinos AMG	#iter	635	647	455	570	626	673	497
	Time	5258.48	137.10	50.32	36.36	30.63	15.04	5.59
	S_1	1.0	1.88	5.13	7.05	8.43	17.18	46.23
INMOST ILU2(10^{-3})	#iter	183	259	288	262	346	370	305
	Time	75.52	53.24	32.68	16.92	16.7	10.2	4.87
	S_1	1.0	1.41	2.31	4.46	4.52	7.40	15.50
BDP	#iter	1848	1343	1219	1375	1527	1579	1608
	Time	138.89	50.62	24.87	16.42	15.66	8.5	4.92
	S_1	1.0	2.74	5.58	8.45	8.86	16.34	28.22

Table: Dependence of velocity convergence from processors number $d_s \in [1; 10^6]$, $m = 64, n = 8, N_A = 1046529$.

Acceleration results

	N_{proc}	1	2	4	8	16	32	64
PETSc ILU(7)	#iter	1248	2046	2173	1767	1337	1629	2476
	Time	147.04	115.13	65.94	35.25	25.55	16	14.75
	S_1	1.0	1.27	2.22	4.17	5.75	9.19	9.96
Trilinos AMG	#iter	116	128	117	133	117	112	114
	Time	48.85	27.89	13.36	8.71	5.82	2.59	1.49
	S_1	1.0	1.75	3.65	5.60	8.39	18.86	32.78
INMOST ILU2(10^{-3})	#iter	156	208	226	200	206	217	237
	Time	53,09	35.51	20.35	10.51	8.20	4.86	3.12
	S_1	1.0	1.49	2.60	5.05	6.47	10.92	17.01
BDP	#iter	1643	2022	1544	1341	1556	1482	1295
	Time	123.45	75.46	30.93	16.06	15.48	7.93	4.03
	S_1	1.0	1.63	3.99	7.69	7.97	15.56	30.63

Table: Dependence of velocity convergence from processors number $d_s = 10^4$, $m = 64, n = 8, N_A = 1046529$.