More numerical experiments on linear and nonlinear elasticity problems

M. Neytcheva, A. Padiy and B. Polman

Abstract

This presentation describes the contribution of the group at KUN to Work-packages 2 and 3 in the following directions:

(1) study of the properties and the behavior of variable versus fixed preconditioners, based on the so-called *separate displacement component* (SDC) formulation of the linear elasticity problem;

(2) solution of problems with nearly incompressible material properties, based on a saddle point formulation of the linear elasticity problem;

(3) solution of nonlinear elasticity problems using the already developed linear solvers.

We present further tests of the performance of the linear solvers on the benchmarks formulated by MOST and GEAM. Also, we report the performance of a robust algebraic multilevel solution technique on irregular locally refined meshes. Finally, we report the theoretical and implementation work done to develop a solver which provides a uniform approach to solve elasticity problems with compressible and almost incompressible materials.

Contents

1 The set of benchmark problems tested at KUN 3

2 Main research directions 3

3 Formulation of the elasticity problem and survey of the possible solution techniques and preconditioners used 3

3.1 Stopping tests 5

3.2 The WALL problem 6

4 Numerical experiments on the Copernicus HIPERGEOS set of linear benchmark problems 8

5 Numerical experiments on the Copernicus HIPERGEOS set of nonlinear benchmark problems 14

6 Fixed versus variable block-diagonal preconditioning for the SDC ordering 15

7 Elasticity solver based on a saddle point formulation 17

8 Conclusions 17

9 Ongoing work 19

1 The set of benchmark problems tested at KUN

The set of benchmark problems run at KUN includes the test problems supplied by both IGAS (Ostrava) and CICT (Sofia) groups. In order to test the iterative solvers and preconditioning techniques developed at KUN, suitable interface programs were made, on one side to use the matrices generated by TGRID (IGAS) and, on the other side, to incorporate the elasticity element stiffness matrix generator created at CICT. Table 1 surveys the tackled problems and the methods studied.

Problem	supplied by	LU		D-SAMLI		Remark
		Var.	Fix.	Var.	Fix.	
STRIP-40	IGAS	■	★	■	■	192000 unkn.
STRIP-80	IGAS	■	■	∅	∅	1536000 unkn.
STOPE-40	IGAS	■	■	■	∅	192000 unkn.
STOPE-80	IGAS	■	★	∅	∅	1536000 unkn.
HEADGATE-1	IGAS	■	■	∅	∅	262548 unkn.
HEADGATE-2	IGAS	■	■	∅	∅	519948 unkn.
HEADGATE-3	IGAS	■	■	∅	∅	777348 unkn.
MOST-32 (lin)	CICT	■	■	■	■	1 pile, 1 soil layer; 98304 unkn.
MOST-63 (lin)	CICT	★	★	■	■	1 pile, 1 soil layer; 750141 unkn.
WALL	KUN	n.a.	n.a	■	■	a 2D analog of the MOST benchmark
MOST-32 (nonlin.)	CICT	★	∅	★	∅	1 pile, 1 soil layer; 98304 unkn.

■	-	detailed tests	★	-	representative tests
∅	-	no tests			

Table 1: Benchmark problems

2 Main research directions

The research work at KUN during the period July, 1997 - January, 1998 was focused on the following three issues:

(1) study of the properties and the behavior of variable versus fixed preconditioners, based on the so-called *separate displacement component* (SDC) formulation of the linear elasticity problem;

(2) solution of problems with nearly incompressible material properties, based on a saddle point formulation of the linear elasticity problem;

(3) solution of nonlinear elasticity problems using the already developed linear solvers.

3 Formulation of the elasticity problem and survey of the possible solution techniques and preconditioners used

In the sake of completeness of this report and to define some notations we include briefly the matrix-vector formulation of the linear elasticity problem based on the SDC ordering of the unknown displacements obtained after some finite element discretization and summarize the approaches to construct a preconditioner to the arising elasticity stiffness matrix.

If $u_k^{(x)}$, $u_k^{(y)}$ and $u_k^{(z)}$ denote the unknown displacement components in x-, y- and z-direction, correspondingly, then the SDC ordering means $\left(u_1^{(x)}, u_2^{(x)}, \cdots, u_n^{(x)}, u_1^{(y)}, u_2^{(y)}, \cdots, u_n^{(y)}, u_1^{(z)}, u_2^{(z)}, \cdots, u_n^{(z)}\right)$. The resulting stiffness matrix A admits the following 3×3 block form, $A = \begin{bmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{bmatrix}$.

Note that the diagonal blocks A_{ii} are discrete analogs of the following anisotropic Laplacians

$$\begin{aligned} \widetilde{\mathcal{D}}_1 &= a\frac{\partial^2}{\partial x_1^2} + b\frac{\partial^2}{\partial x_2^2} + b\frac{\partial^2}{\partial x_3^2}, \quad \widetilde{\mathcal{D}}_2 = b\frac{\partial^2}{\partial x_1^2} + a\frac{\partial^2}{\partial x_2^2} + b\frac{\partial^2}{\partial x_3^2}, \\ \widetilde{\mathcal{D}}_3 &= b\frac{\partial^2}{\partial x_1^2} + b\frac{\partial^2}{\partial x_2^2} + a\frac{\partial^2}{\partial x_3^2}, \end{aligned} \tag{3.1}$$

and the presence of anisotropy in these operators requires special care to construct an efficient preconditioner for the iterative solution method.

Due to the spectral equivalence relations between the stiffness matrix A and its block-diagonal part $A_D = \begin{bmatrix} A_{11} & & \\ & A_{22} & \\ & & A_{33} \end{bmatrix}$ (with a small spectral equivalence constant for values of ν away from $1/2$) A_D becomes a natural and attractive preconditioner for A (see earlier reports and the references therein).

More general, to construct a preconditioner M for A of the form

$$M = \begin{bmatrix} M_{11} & & \\ & M_{22} & \\ & & M_{33} \end{bmatrix}. \tag{3.2}$$

we have several possibilities.

(a1) Let $M \equiv A_D$. This is the choice when solving STOPE, STRIP and HEADGATE. Since these three problems are discretized using linear basis functions on tetrahedral finite elements, the diagonal blocks for the SDC ordering have as an extra advantage a 7-diagonal nonzero structure which has been further exploited when constructing a preconditioner for A.

(a2) Let $M = blockdiag\{\widetilde{A}_{ii}\}, i = 1, 2, 3$, where the blocks $\widetilde{A}_{ii}$ correspond to a finite element discretization using finite elements different from those used for A. This is the case when solving MOST, where trilinear basis functions on brick elements are used for A and linear basis functions on tetrahedra are used for $\widetilde{A}_{ii}$.

(a3) Let $M = blockdiag\{\widetilde{Q}_{ii}\}, i = 1, 2, 3$, where $\widetilde{Q}_{ii}$ are discrete analogs to another (simpler) differential operator, such as isotropic Laplacian, for example. This has been used when solving the WALL problem. There, instead of (3.1), the preconditioner is based on

$$\widetilde{\mathcal{D}}_B = E^* \begin{bmatrix} b\frac{\partial^2}{\partial x_1^2} + b\frac{\partial^2}{\partial x_2^2} + b\frac{\partial^2}{\partial x_3^2} & 0 & 0 \\ 0 & b\frac{\partial^2}{\partial x_1^2} + b\frac{\partial^2}{\partial x_2^2} + b\frac{\partial^2}{\partial x_3^2} & 0 \\ 0 & 0 & b\frac{\partial^2}{\partial x_1^2} + b\frac{\partial^2}{\partial x_2^2} + b\frac{\partial^2}{\partial x_3^2} \end{bmatrix}.$$

For the background of this approach we refer to [1].

Whatever method has been chosen to construct M, we have two possibilities to solve systems with it.

(b1) One approach is to use an inner solver to solve systems with each of the diagonal blocks M_{ii}. In general, the inner solver is another preconditioner iterative procedure and since the action of the preconditioner depends on the used inner stopping criteria, the inner solver acts as a *variable* (changing from one outer iteration to another) preconditioner to A.

(b2) The second approach is to construct some approximations to the diagonal blocks of M and solve with those approximated blocks once per each iteration. This possibility is referred to as *fixed* preconditioning.

As approximations of the diagonal blocks M_{ii} in case (b2) the same matrices have been used as those used to precondition them in the (b1) case, namely, dynamic block-incomplete factorization (D-MBILU), multiplicative (short) Algebraic Multilevel Iteration (D-SAMLI) method and additive Algebraic Multilevel Iteration (I-AMLI) method. Various versions of the D-MBILU preconditioner were tested. One of them is referred to as `zebra`. The preconditioner constructed by this version is weaker compared to some other D-MBILU versions but it uses special orderings which facilitate parallel implementations (see [8]).

Three different solution methods were used as outer solvers - preconditioned conjugate gradient (*pcg*) method, generalized conjugate gradient-least squares (*gcgls*) method and generalized conjugate gradient-minimal residual (*gcgmr*) method. All three have been tested with various inner stopping tests. Both *gcg* methods have also been tested with varying number of search directions (1,3 or 5), keeping them as few as possible in order to keep the computational cost and the memory requirements for the inner solver low.

3.1 Stopping tests

The stopping criteria which have been used for the different benchmark problems are as defined by the partner group which has defined the corresponding benchmark, namely, as shown in Table 2.

Benchmark: STRIP, STOPE, HEADGATE	
Outer solver (*pcg, gcgls, gcgmr*)	
- initial guess:	zero
- stopping test:	$\|r^{(i)}_{outer}\| \leq 10^{-4}\|r^{(0)}_{outer}\|$
* r_{outer} - the iteratively computed residual	
Inner solver (*pcg*)	
- initial guess:	$[prec]^{-1}rhs$
- stopping test:	$\|r^{(i)}_{inner} \leq min(10^{-3}, \sqrt{\|r^{(i)}_{outer}\|^2/\|r^{(0)}_{outer}\|^2})$

Benchmark:MOST	
- initial guess:	zero
- stopping test:	$\|\tilde{r}^{(i)}_{outer}\| \leq 10^{-6}$
* r_{outer} - the computed pseudoresidual	
Inner solver (*pcg*)	
- initial guess:	$[prec]^{-1}rhs$
- stopping test:	$\|r^{(i)}_{inner} \leq min(10^{-3}, \|r^{(i)}_{outer}\|^2)$

Coarsest level solver (*diag.prec.pcg*)	
- initial guess:	zero
- stopping test:	$\|r^{(i)}_{coars}\|^2 \leq 10^{-3}\|r^{(0)}_{coars}\|^2$
* Used in the coarsest level solver for the D-SAMLI preconditioner.	

Table 2: Stopping tests

3.2 The WALL problem

We include a description of the WALL problem because it has not appeared in previous reports.

In the WALL problem we consider a concrete wall placed in a homogeneous soil layer and subjected by external surface forces $\underline{\gamma}$ and volume forces $\mathbf{g}$, see Figure 1.

We assume that there is no external force acting in the direction along the wall. In this case the displacement field is zero in this direction and, therefore, the problem can be described in terms of a two–dimensional model. In the computations we take the volume force $\mathbf{g}$ in the form $(0, -g_v)$. It represents the gravity force acting on the continuous media. Consider also a vertical external load $\underline{\gamma} = (0, -\gamma_v)$ applied to the top of the wall. We assume that the load is uniformly distributed over the loading surface.

This problem is symmetric with respect to the central plane of the wall, the displacement field is also symmetric and, therefore, the problem can be considered only in one half of the domain, i.e. $\Omega = [0, L_d/2] \times [0, H_d]$. The following boundary conditions are chosen in the experiments: homogeneous Dirichlet b.c. on the bottom side of the domain (displacements are equal zero), homogeneous Neumann b.c. on the rest of the boundary (stresses are equal zero), inhomogeneous Neumann b.c. on the loading surface and "rigid contact" b.c. ($u_x = 0$) on the plane of the symmetry, i.e. when $x = 0$. The Young's

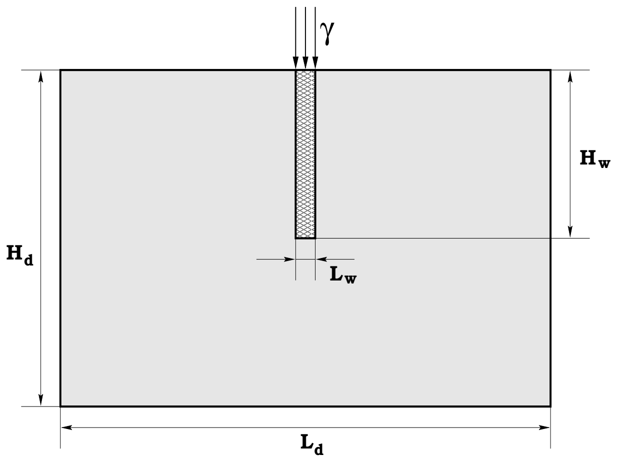

Figure 1: The WALL problem, vertical cross–section of the domain

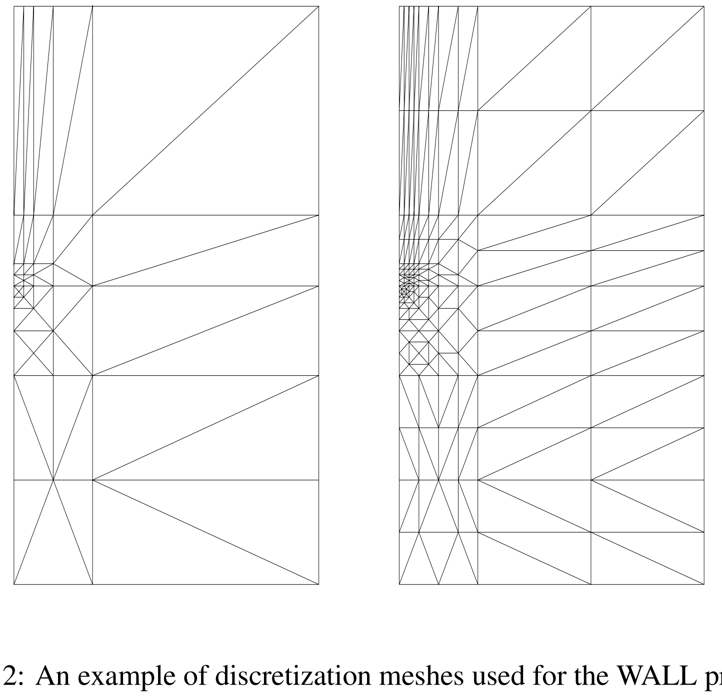

Figure 2: An example of discretization meshes used for the WALL problem

modulus E and the Poisson ratio ν are defined as

$$E = \begin{cases} E_w, & 0 \le x \le L_d/2,\ 0 \le y \le H_d, \\ E_s, & \text{otherwise,} \end{cases}$$

$$\nu = \begin{cases} \nu_w, & 0 \le x \le L_d/2,\ 0 \le y \le H_d, \\ \nu_s, & \text{otherwise.} \end{cases}$$

We fix the values $L_d, H_d, L_w, H_w, E_s, \nu_s, E_w$ and ν_w to be the following:

$$L_d = 37.2\,m, \quad H_d = 31.0\,m, \quad L_w = 1.2\,m, \quad H_w = 15.0\,m,$$

$$E_s = 1.0 \cdot 10^7\,Pa, \quad \nu_s = 0.3, \quad E_w = 3.15 \cdot 10^{10}\,Pa, \quad \nu_w = 0.2.$$

The above parameter setting corresponds to the benchmark problem defined in [7].

We discretized the problem on a sequence of irregular triangular meshes, refined near the bottom end of the wall. An illustration of the used triangulations is given in Figure 2.

4 Numerical experiments on the Copernicus HIPERGEOS set of linear benchmark problems

This section includes some examples of the test runs on STRIP, STOPE, HEADGATE and MOST. Although a much larger number of tests has been performed for various values of the solution and factorization parameters (especially with D-MBILU), we include a representative number of tests to illustrate the insensitivity of the solvers with respect to those parameters.

The results are found in Tables 3 – 9 and Tables 17 – 26 in the Appendix. The tables contain the type of the outer solution method, the type of the preconditioner, the factorization parameter ω (if applicable) and experimental data regarding fixed versus variable preconditioning, different number of search directions (for the *gcg* methods) and various values for the accuracy of the inner solver. Whenever ”fixed” appears in the ”Inner stopping tolerance” column of the tables, the results correspond to the fixed version of the preconditioner, i.e., no inner solution procedure is applied to the diagonal blocks $M_{ii}, i = 1, 2, 3$ but the action of the corresponding preconditioner is applied only once instead.

The general observations regarding the tests with the preconditioners D-MBILU and D-BILU-zebra is that the performance fits in more or less one and the same pattern.

- The fixed preconditioning version for all benchmarks except HEADGATE produces more iterations with the global stiffness matrix A, which are more expensive due to the fact that the off-diagonal blocks of A are substantially denser than the diagonal blocks. However, the loss in computing time is not dramatic, while the fixed preconditioning strategy is simpler because it avoids the tunning of the inner stopping criterion.

  The HEADGATE benchmark befefits in addition from the block-incomplete factorization preconditioning due to the fact that the size of the domain in one direction (in this case x) is larger than the others and the difference between HEADGATE-1, HEADGATE-2 and HEADGATE-3 is only in the number of unknowns in this direction. The latter makes the convergence rate almost independent on the size of the problem and the theory is nicely confirmed with the numerical experiments (see Tables 21 – 24).

Search direct. used	Inner stopping tolerance	Number of outer iterations	Av. number of inner iterations	Solution time (in sec.) total execution	solution	inner solver
Solution method: *gcgls* Preconditioner: dbilu						
1	fixed	39	1.00000	58.54	47.83	8.66
1	0.001	11	5.63889	54.54	44.95	33.22
1	0.0005	11	6.27778	58.37	49.20	37.44
1	0.0001	10	7.87879	62.87	53.38	42.57
3	fixed	39	1.00000	59.67	50.49	8.68
3	0.001	11	5.58333	55.08	45.54	33.07
3	0.0005	11	6.13889	58.49	48.75	36.30
3	0.0001	10	7.87879	63.37	53.93	42.52
Solution method: *gcgls* Preconditioner: zebra						
1	fixed	77	1.00000	105.74	95.48	17.55
1	0.001	12	9.56410	83.02	73.23	59.96
1	0.0005	11	11.3056	87.21	77.26	64.94
1	0.0001	10	15.6061	102.24	92.30	80.98
3	fixed	77	1.00000	110.54	101.06	17.59
3	0.001	11	9.30555	75.84	66.35	53.69
3	0.0005	11	11.3333	86.99	77.66	65.01
3	0.0001	10	15.6061	102.29	92.65	81.02

Table 3: *strip* $40 \times 40 \times 40$ $\omega = 0.3$

- The performance dependes only to a minor extent on the value of the factorization parameter ω. It turns out that for D-MBILU the best performance (smallest computing times) are obtained for $\omega = 0.9$ amd for D-BILU-zebra - for $\omega = 0.3$.
- For the variable preconditioning version, the dependence on the inner stopping tolerance is also insignificant with respect to the resulting solution time.
- For the gcg-type methods, it suffices to use as few search directions as possible (one, two or three).

All the above confirms that the dynamic block-modified incomplete factorization techniques (on pseudoregular meshes as in the considered cases) are fast and robust, and compete successfully with the multilevel techniques. The problematic issue with the BILU methods is that they do not parallelize well which makes them less attractive for parallel implementations.

The general observations regarding the tests with the AMLI preconditioners show also their robustness with respect to discontinuities in the problem coefficients (for both D-SAMLI and I-AMLI) and with respect to anisotropies (especially for I-AMLI). In addition, these techniques parallelize well and scale well with the problem size and the increasing number of processors on a parallel computer system.

Search direct. used	Inner stopping tolerance	Number of outer iterations	Av. number of inner iterations	Solution time (in sec.) total execution	solution	inner solver
Solution method: *gcgls* Preconditioner: dbilu						
1	fixed	32	1	51.92	39.37	7.56
1	0.001	12	4	56.04	44.71	32.21
1	0.0005	11	5	56.03	45.24	33.72
1	0.0001	11	6	64.94	53.38	41.67
3	fixed	32	1	53.25	42.02	7.56
3	0.001	11	4	52.18	41.58	29.45
3	0.0005	11	5	57.25	45.50	33.43
3	0.0001	11	6	64.94	53.59	41.35
Solution method: *gcgls* Preconditioner: zebra						
1	fixed	72	1	99.17	87.65	17.32
1	0.001	21	5	104.99	93.17	71.95
1	0.0005	14	6	85.50	74.12	59.66
1	0.0001	12	9	97.34	85.95	73.49
3	fixed	72	1	104.08	92.71	17.34
3	0.001	21	5	109.85	98.29	75.75
3	0.0005	15	6	92.73	80.91	64.38
3	0.0001	12	9	98.56	86.80	73.54

Table 4: *strip* $40 \times 40 \times 40$ $\omega = 0.9$

Search direct. used	Inner stopping tolerance	Number of outer iterations	Av. number of inner iterations	Solution time (in sec.) total execution	solution	inner solver
Solution method: *gcgls* Preconditioner: dbilu						
1	fixed	59	1.00000	747.84	654.12	124.75
1	0.001	12	6.38462	594.68	515.01	400.59
1	0.0005	11	7.02778	586.67	511.16	405.65
1	0.0001	11	9.44444	723.83	648.32	542.77
3	fixed	59	1.00000	763.04	686.45	124.63
3	0.001	12	6.33333	599.09	518.76	397.48
3	0.0005	11	6.97222	590.39	514.25	402.57
3	0.0001	11	9.36111	729.32	648.51	536.89
Solution method: *gcgls* Preconditioner: zebra						
1	fixed	136	1.00000	1578.07	1500.22	289.56
1	0.001	30	6.44086	1331.83	1251.07	977.76
1	0.0005	19	9.81667	1198.12	1122.07	945.91
1	0.0001	11	20.0000	1323.61	1247.59	1142.06
3	fixed	136	1.00000	1651.42	1574.98	289.34
3	0.001	27	6.66667	1248.42	1171.51	910.33
3	0.0005	17	9.90741	1102.30	1025.47	858.13
3	0.0001	11	20.0000	1335.57	1253.06	1142.13

Table 5: *strip* $80 \times 80 \times 80$ $\omega = 0.3$

Search direct. used	Inner stopping tolerance	Number of outer iterations	Av. number of inner iterations	Solution time (in sec.) total execution	solution	inner solver
Solution method: *gcgls* Preconditioner: dbilu						
1	fixed	48	1.00000	633.75	535.76	104.95
1	0.001	14	5.08889	631.28	540.05	409.55
1	0.0005	12	5.87179	540.56	451.72	342.35
1	0.0001	11	7.77778	657.20	567.30	459.94
3	fixed	48	1.00000	681.44	588.70	135.30
3	0.001	14	5.11111	605.76	515.31	377.03
3	0.0005	12	5.89744	605.06	518.10	392.99
3	0.0001	11	7.77778	693.25	601.27	484.28
Solution method: *gcgls* Preconditioner: zebra						
1	fixed	112	1.00000	1360.39	1266.39	252.16
1	0.001	46	4.52482	1561.05	1465.56	1053.89
1	0.0005	36	5.69369	1402.16	1308.39	981.23
1	0.0001	16	10.31373	1118.47	1029.94	878.86
3	fixed	112	1.00000	1391.74	1286.44	238.73
3	0.001	39	4.71667	1412.31	1310.35	917.40
3	0.0005	33	5.84314	1356.81	1257.91	940.26
3	0.0001	16	10.15686	1071.63	981.78	821.98

Table 6: *strip* $80 \times 80 \times 80$ $\omega = 0.9$

ω	Inner stopping tolerance	Number of outer iterations	Av. number of inner iterations	Solution time (in sec.) total execution	solution	inner solver
Solution method: *pcg* Preconditioner: dbilu						
0.1	fixed	142	1.00/ 1.00/ 1.00	400.36	334.29	19.34
0.1	0.001	21	12.90/ 9.29/ 14.57	182.31	116.22	72.62
0.1	0.0001	20	12.65/ 9.55/ 16.45	178.51	112.22	71.46
0.1	1e-05	20	17.70/ 10.00/ 15.60	187.05	121.13	79.46
0.2	fixed	143	1.00/ 1.00/ 1.00	395.07	327.53	22.17
0.2	0.001	27	12.89/ 9.56/ 12.30	209.00	143.34	87.68
0.2	0.0001	20	12.30/ 7.50/ 12.70	167.72	101.45	60.68
0.2	1e-05	20	14.40/ 9.75/ 14.30	178.62	113.15	71.44
0.3	fixed	161	1.00/ 1.00/ 1.00	427.41	358.76	20.63
0.3	0.001	38	12.47/ 8.50/ 12.24	263.16	195.96	118.74
0.3	0.0001	20	11.75/ 7.80/ 11.75	165.72	99.66	58.74
0.3	1e-05	20	12.60/ 9.45/ 13.45	173.81	108.19	66.42
0.4	fixed	157	1.00/ 1.00/ 1.00	419.39	353.50	20.51
0.4	0.001	38	11.79/ 7.29/ 10.37	250.13	184.78	107.24
0.4	0.0001	26	13.31/ 10.00/ 12.31	205.99	139.70	86.41
0.4	1e-05	20	13.15/ 10.00/ 12.50	175.58	109.34	67.57
0.9	fixed	147	1.00/ 1.00/ 1.00	396.56	330.73	19.98
0.9	0.001	46	10.70/ 7.17/ 9.61	279.07	213.09	121.22
0.9	0.0001	29	13.66/ 9.31/ 11.17	218.13	152.91	93.18
0.9	1e-05	20	12.00/ 9.60/ 11.60	170.42	103.78	62.68
Solution method: *gcgls* Preconditioner: zebra						
0.1	fixed	110	1.00/ 1.00/ 1.00	313.24	247.41	14.23
0.1	0.001	21	16.86/ 8.81/ 16.48	193.22	126.31	80.30
0.1	0.0001	20	18.90/ 10.35/ 17.20	194.03	128.05	83.89
0.1	1e-05	20	20.35/ 13.10/ 18.45	201.91	136.28	92.38
0.2	fixed	106	1.00/ 1.00/ 1.00	337.32	237.32	13.70
0.2	0.001	21	9.33/ 7.62/ 10.52	166.86	101.27	55.27
0.2	0.0001	20	12.60/ 7.85/ 13.70	174.05	107.74	63.69
0.2	1e-05	20	14.70/ 8.75/ 14.00	179.93	113.02	68.99
0.3	fixed	122	1.00/ 1.00/ 1.00	339.61	273.56	15.71
0.3	0.001	22	9.50/ 7.23/ 10.55	171.72	105.72	57.58
0.3	0.0001	20	9.65/ 8.30/ 12.10	168.92	101.17	56.97
0.3	1e-05	20	11.85/ 9.65/ 14.30	201.74	135.89	91.86
0.4	fixed	136	1.00/ 1.00/ 1.00	370.51	304.51	17.52
0.4	0.001	27	6.63/ 7.15/ 10.33	188.40	122.02	63.51
0.4	0.0001	20	9.00/ 8.85/ 12.55	168.06	101.71	57.58
0.4	1e-05	20	10.75/ 10.35/ 14.10	174.63	108.99	65.20
0.9	fixed	181	1.00/ 1.00/ 1.00	471.32	405.18	23.28
0.9	0.001	40	7.72/ 8.50/ 10.10	302.17	208.15	102.62
0.9	0.0001	20	9.25/ 10.30/ 11.25	169.34	102.57	58.30
0.9	1e-05	20	12.30/ 13.20/ 12.70	180.30	113.94	70.04

Table 7: *most* $32 \times 32 \times 32$

Solution method: *gcgls* Preconditioner: dbilu							
Search direct. used	ω	Inner stopping tolerance	Number of outer iterations	Av. number of inner iterations	Solution time (in sec.) total execution	solution	inner solver
1	0.1	fixed	217	1.00/ 1.00/ 1.00	2477.03	1947.97	198.47
1	0.1	0.001	29	18.45/ 11.66/ 20.76	1810.74	1287.32	1046.55
1	0.2	fixed	221	1.00/ 1.00/ 1.00	2482.22	1945.41	198.00
1	0.2	0.001	37	13.78/ 8.78/ 15.84	1846.88	1319.77	1019.15
1	0.3	fixed	215	1.00/ 1.00/ 1.00	2432.19	1904.14	191.23
1	0.3	0.001	51	12.96/ 8.51/ 11.71	2176.26	1648.36	1235.04
1	0.4	fixed	228	1.00/ 1.00/ 1.00	2528.10	1990.82	200.24
1	0.4	0.001	51	13.22/ 8.61/ 12.78	2333.88	1811.81	1255.93

Table 8: *most* $63 \times 63 \times 63$

Solution method: *pcg* Preconditioner: *amli*						
Size	Inner stopping tolerance	Number of outer iterations	Av. number of inner iterations	Solution time (in sec.) total execution	solution	inner solver
32^3	fixed	77	1/ 1/ 1	339.24	272.65	107.84
32^3	0.001	39	7/ 6/ 7	515.09	449.76	363.97
32^3	0.0001	26	7/ 6/ 7	695.61	630.05	572.03
32^3	1.e-05	21	9/ 7/ 9	740.77	674.64	627.28
63^3	fixed	48	1.0/ 1.0/ 1.0	1630.15	1104.69	711.02
63^3	0.001	20	6.10/ 4.65/ 6.00	3184.56	2661.99	2495.24
63^3	0.0001	18	6.94/ 5.50/ 7.28	3116.06	2592.22	2443.64

Table 9: *most*

5 Numerical experiments on the Copernicus HIPERGEOS set of nonlinear benchmark problems

The nonlinear solver reported here is analogous to that developed by CICT. The difference is in the linear solution method - in this case it is the inner-outer variable preconditioning using D-MBILU and D-SAMLI preconditioners, developed at KUN. Some examples of the convergence behaviour of the nonlinear solver is given in Table 10.

Prec.	Stop.crit. (outer lin.)	Nonlin. iter.	Average outer iter. (lin.)	Average inner iter. (lin.)
D-MBILU	absol.10^{-6}	20	16	11/13/15-38/33/16
D-SAMLI	absol.10^{-6}	20	14	6/5/6-12/10/12

Table 10: MOST-32, 1 pile, 1 soil layer, $\|nr^{(i)}\| < 5\,10^{-4}$

Problem size	Number of outer iterations	Computing time in seconds
280	23	0.3
1088	25	1.0
4288	27	5.7
17024	29	30.4
67840	31	143.9

Table 11: The WALL problem, I-AMLI preconditioner Q^B_{vs}, $\varepsilon_{inner} = 10^{-1}$

6 Fixed versus variable block-diagonal preconditioning for the SDC ordering

As can be seen from the tests for MOST, the tested fixed preconditioners perform somewhat worse than the corresponding inner-outer solution method. The effect is more clearly seen for very large-scale problems. The major reason is that the fixed preconditioner causes an increase of the iterations and since they are quite expensive due to the many more nonzero elements in the off-diagonal blocks of A, this increases the total solution time. Note, however, that both preconditioners (D-MBILU and D-SAMLI) are not optimal with respect to convergence rate. For D-MBILU this is known from the theory for ILU methods. Although the D-SAMLI methods are (nearly) optimal with respect to both convergence rate and computational complexity, this particular version faces difficulties to handle anisotropies.

This section shows that the situation is reversed if an optimal and robust with respect to anisotropies preconditioning method is used - I-AMLI. Tables 11, 12, 13 and 14 illustrate the above conclusion. (Here the preconditioners Q^B_{vs} and Q^B_{fx} are generated by the operator $\widetilde{\mathcal{D}}_B$.)

The I–AMLI preconditioning technique is a robust multilevel algorithm suggested in [4]. This method is a black–box solver based on a sequence of nested CG–like iterations, which makes it attractive for industrial use. It works on a hierarchical sequence of finite element meshes to solve the problem with an arithmetic cost, nearly proportional to the dimension of the arising algebraic system. The method imposes no restrictions on the choice of the coarse mesh and, therefore, any suitable meshing procedure can be chosen to discretize the problem efficiently. This issue becomes especially important, when the shape of the domain is complicated and the material coefficients E and ν have discontinuities with big jumps. The number of user–supplied adjustments in the method is also small. The only "free" parameters, which may have to be tuned in the solver are the following: a) a set of "regular" levels $k_1, \ldots, k_r$ used to stabilize the overall condition number of the preconditioned system and b) a set of stopping tolerances $\varepsilon_1, \ldots, \varepsilon_r$ on each of the regular levels. Note also, that the I–AMLI method is not sensitive with respect to the above parameters (cf. [4]). In all tests the accuracy for ε_i is fixed and the asymptotic formulae from [2] is used to get a structure of the regular levels, which is sufficiently close to an optimal one.

Problem size	Number of outer iterations	Computing time in seconds
280	20	0.4
1088	23	1.4
4288	27	9.6
17024	29	49.0
67840	30	224.6

Table 12: The WALL problem, I-AMLI preconditioner Q^B_{vs}, $\varepsilon_{inner} = 10^{-2}$

Problem size	Number of outer iterations	Computing time in seconds
280	20	0.5
1088	24	2.2
4288	27	13.4
17024	29	68.1
67840	30	312.3

Table 13: The WALL problem, I-AMLI preconditioner Q^B_{vs}, $\varepsilon_{inner} = 10^{-3}$

Problem size	Number of outer iterations	Computing time in seconds
280	34	0.1
1088	65	0.6
4288	63	3.1
17024	68	14.0
67840	70	65.1
270848	69	293.2

Table 14: The WALL problem, fixed I-AMLI preconditioner Q^B_{fx}

7 Elasticity solver based on a saddle point formulation

Using a mixed variables formulation of the elasticity equation is one way to avoid the so-called "locking" phenomenon (see, for instance, [6]) offering an algorithm which is applicable even in the case of almost incompressible media, when the contractivity (Poisson) ratio ν tends to $1/2$. The method is based on the idea to introduce an extra variable, the *hydrostatic pressure* which enables us to rewrite the elasticity problem as an incompressible Stokes problem. To balance the discretization errors for the displacements and pressure, in this case the displacement field $\mathbf{u}$ is discretized by piecewise–quadratic functions, while pressure p will be discretized by piecewise linears.

The standard finite–element discretization procedure of Stokes equations in linear elasticity reads as follows:

$$\begin{bmatrix} A^h & B^h \\ {B^h}^T & -M^h \end{bmatrix} \begin{bmatrix} \mathbf{u}^h \\ p^h \end{bmatrix} = \begin{bmatrix} \mathbf{g}^h \\ 0 \end{bmatrix}. \tag{7.1}$$

The latter system is further reduced to the pressure variable and solved iteratively using the *pcg* method. A lumped mass matrix for the pressure is used as a preconditioner for the (outer) Schur complement CG iterations.

To perform a matrix-vector multiplication with the arising Schur complement system requires to solve systems with the matrix A^h which is done by an (inner) I-AMLI preconditioned conjugate gradient method.

The detailed description on the I-AMLI preconditioner for the matrix A^h, which includes special robust treatment of the anisotropy is found in [5]. As it turns out, the imposed restrictions on the meshing procedure is very weak and can be easily fulfilled in the majority of practical cases.

As shown in [5], with a proper choice of the meshing procedure, used for the discretization, the global condition number of the I-AMLI–preconditioned system can be made arbitrarily close to optimal, i.e. bounded with respect to mesh step size, mesh regularity and problem coefficients. Moreover, by balancing the distance between the regularization points in the multilevel structure, the cost per I–AMLI iteration step can be made nearly proportional to the number of degrees of freedom on the finest mesh (see [2] and [4] for more details).

Tables 15 and 16 show that the performance of the algorithm indeed depends very mildly on the Poisson ratio ν_s and computing time is almost proportional to the problem size.

8 Conclusions

- **Re: Linear solvers - D-MBILU and D-SAMLI preconditioning**
  For the considered class of problems, represented by the benchmark tests from GEAM and MOST, the D-MBILU preconditioning technique shows both efficiency (high performance) and robustness for middle- and large-sized problems. Without using any special optimization techniques, the implementation keeps simple and exploits completely the advantages of the internal structure of the problems and proves to be superior in most of the cases.

  Both D-SAMLI and I-AMLI are more general and are based on more involved preconditioning techniques. The particular version of the multiplicative AMLI method, D-SAMLI, however, is not suited for problems with anisotropies and shows unfavorable convergence rates for the GEAM benchmarks. It handles efficiently problems with large coefficient jumps, such as the MOST benchmarks, and although the method is computationally more expensive than D-MBILU, it shows stable and scalable performance.

Problem size	$\nu_s = 0.3$	$\nu_s = 0.4$	$\nu_s = 0.45$	$\nu_s = 0.49$	$\nu_s = 0.499$	$\nu_s = 0.4999$
280	2.1	2.6	2.7	2.9	3.1	3.1
1088	13.4	16.1	17.1	21.0	21.4	20.8
4288	70.1	83.3	96.9	111.3	115.1	113.2
17024	265.6	350.6	396.9	466.3	490.8	486.0
67840	1250.9	1667.9	1838.9	2087.3	2146.8	2112.7

Table 15: Computing time vs. problem size for the different values of Poisson ratio ν_s

Problem size	$\nu_s = 0.3$	$\nu_s = 0.4$	$\nu_s = 0.45$	$\nu_s = 0.49$	$\nu_s = 0.499$	$\nu_s = 0.4999$
280	17	22	23	27	27	27
1088	19	23	26	32	32	32
4288	19	24	28	32	33	33
17024	17	24	28	31	33	33
67840	17	24	27	31	32	32

Table 16: Number of outer Schur complement iterations vs. problem size for the different values of Poisson ratio ν_s

The experiments with the I-AMLI preconditioning technique show very good performance and scalability of the latter method. It is more general and designed for highly unstructured locally refined meshes which makes it a desirable choice for many practical implementations.

- **Re: Nonlinear solvers**
  The work on implementing solution methods for the nonlinear problems defined from MOST XXI was done in collaboration with the partner group in CICT. All conclusions and the detailed analysis of the results of the experiments are presented in the report *Tests of two- and multilevel patched refinement FEM code for nonlinear multi-layer interaction*, A. Georgiev, S. Margenov, P. S. Vassilevski and L. Zikatanov. The multilevel preconditioning technique (*amli*) used for the linear part of the nonlinear solver, proves to be robust with respect to the strong discontinuities, typical for the MOST benchmarks, which results in few average inner iterations (see Table 35). Further efforts will focus on increasing the performance of the nonlinear solver itself.

- **Re: Fixed versus variable preconditioning**
  The general conclusion is that the fixed preconditioning techniques using (nearly) optimal preconditioners are to be recommended for practical use. The variable (inner-outer type) preconditioning has, however, its place in the gallery of iterative solution methods, especially when the preconditioner is both robust and simple to implement. The advantages become substantial when coarse grained parallel approach is applicable (as observed in the experiments by IGAS) since the required communications needed for the outer solver are minor compared to the workload performed per inner solve.

- **Re: Saddle point formulation**
  An efficient inner/outer–type method has been constructed for solving the problems of linear isotropic elasticity. The presented algorithm is essentially a black–box solver, which requires only a few user–supplied adjustments and can be constructed automatically starting from any

given initial mesh triangulation. The method shows a good performance in the whole range of material coefficients E and ν with no "locking" and, therefore, is attractive for industrial use. It can be efficiently applied even in the difficult case of nearly incompressible media, which frequently arises in practice for certain rubber–like materials, for instance.

Another important feature of the algorithm is that it imposes only very weak restrictions on the meshing procedure. This allows us to discretize the problem efficiently using a moderate number of refinement steps and resolve the solution with a reasonable arithmetic cost.

9 Ongoing work

Parallel implementation of a 3D-elasticity solver based on the SDC-ordering using the I-AMLI preconditioner and a version of a multiplicative version of D-SAMLI.

References

[1] O. Axelsson, On iterative solvers in structural mechanics. In: *Proceedings of the Conference on Preconditioned Iterative Solution Methods for Large Scale Problems in Scientific Computations "PRISM-97"*, University of Nijmegen, May 27–29, 1997, pp.55–62.

[2] O. Axelsson, Stabilization of AMLI methods; additive methods. In: *Proceedings of the Conference on Algebraic Multilevel Iteration Methods with Applications*, University of Nijmegen, June 13–15, 1996, pp.49–62.

[3] O. Axelsson and M. Neytcheva, Scalable algorithms for the solution of Navier's equations of elasticity, *J. Comp. Appl. Math.* **63**, 1995, pp.149–178.

[4] O. Axelsson and A. Padiy, On the additive version of the algebraic multilevel iteration method for anisotropic elliptic problems, Technical report No 9711, University of Nijmegen, 1997, to appear in SIAM J. Sci. Comp.

[5] O.Axelsson and A. Padiy, On a robust and scalable Linear Elasticity solver based on a saddle point formulation. Report No. 9723, October 1997, Department of Mathematics, University of Nijmegen, The Netherlands.

[6] D. Braess, *Finite elements. Theory, fast solvers, and applications in solid mechanics*, Cambridge Univ. Press, 1997.

[7] A. Georgiev, A. Baltov and S. Margenov, HIPERGEOS benchmark problems related to bridge engineering applications, project report COP-94-0820-MOST-4.

[8] M. Monga Made, B. Polman, Efficient Planewise like preconditioners to cope with 3D problems. Report 987, Department of Mathematics, University of Utrecht.

Appendix

The Appendix contains more experimental data for the various benchmark problems and the various preconditioning techniques and solution methods used in the tests.

Search direct. used	Inner stopping tolerance	Number of outer iterations	Av. number of inner iterations	Solution time (in sec.) total execution	solution	inner solver
Solution method: *gcgls* Preconditioner: dbilu						
1	fixed	47	1.00000	68.23	57.32	10.41
1	0.001	11	6.52778	58.59	49.30	37.58
1	0.0005	11	7.16667	62.14	52.98	41.27
1	0.0001	11	9.72222	77.09	67.49	55.76
3	fixed	47	1.00000	69.18	60.18	10.39
3	0.001	11	6.38889	58.37	49.16	36.85
3	0.0005	11	7.41667	64.47	55.20	42.87
3	0.0001	11	9.77778	77.43	68.30	56.01
Solution method: *gcgls* Preconditioner: zebra						
Search direct. used	Inner stopping tolerance	Number of outer iterations	Av. number of inner iterations	Solution time (in sec.) total execution	solution	inner solver
1	fixed	104	1.00000	135.46	126.05	23.38
1	0.001	12	11.4872	92.37	82.92	69.94
1	0.0005	12	14.8462	112.43	102.66	89.69
1	0.0001	11	21.1944	137.45	128.18	116.19
3	fixed	104	1.00000	141.98	132.56	23.37
3	0.001	13	11.4048	98.89	89.37	74.99
3	0.0005	11	14.9722	104.71	95.23	82.91
3	0.0001	11	20.4722	133.89	124.13	111.81

Table 17: *stope* $40 \times 40 \times 40$ $\omega = 0.3$

Search direct. used	Inner stopping tolerance	Number of outer iterations	Av. number of inner iterations	Solution time (in sec.) total execution	solution	inner solver
Solution method: *gcgls* Preconditioner: dbilu						
1	fixed	55	1	76.65	65.70	12.78
1	0.001	15	5	70.49	60.03	44.57
1	0.0005	13	6	72.78	60.81	47.54
1	0.0001	11	9	77.80	67.13	55.83
3	fixed	55	1	80.74	69.66	12.81
3	0.001	15	5	72.15	61.02	44.71
3	0.0005	13	6	72.92	61.90	47.92
3	0.0001	12	9	85.82	75.14	62.00
Solution method: *gcgls* Preconditioner: zebra						
1	fixed	100	1	130.61	119.35	23.62
1	0.001	28	5	124.23	112.62	84.46
1	0.0005	26	6	132.24	121.00	94.93
1	0.0001	12	12	111.33	100.33	87.85
3	fixed	100	1	138.58	127.01	23.77
3	0.001	28	5	124.14	113.30	83.57
3	0.0005	22	6	115.33	104.46	80.96
3	0.0001	12	13	113.38	102.13	88.88

Table 18: *stope* $40 \times 40 \times 40$ $\omega = 0.9$

Search direct. used	Inner stopping tolerance	Number of outer iterations	Av. number of inner iterations	Solution time (in sec.) total execution	solution	inner solver
Solution method: *gcgls* Preconditioner: dbilu						
1	fixed	68	1.00000	867.09	790.77	146.83
1	0.001	19	5.63333	962.60	888.74	689.45
1	0.0005	13	7.61905	839.00	760.95	621.78
1	0.0001	12	11.7179	1080.40	1002.16	872.82
3	fixed	68	1.00000	901.79	828.58	146.79
3	0.001	17	5.96296	911.96	835.79	647.77
3	0.0005	15	7.35417	941.96	865.22	698.30
3	0.0001	12	11.7179	1083.85	1008.46	873.26
Solution method: *gcgls* Preconditioner: zebra						
1	fixed	182	1.00000	2147.09	2070.01	394.31
1	0.001	57	5.50000	2535.83	2460.11	1901.19
1	0.0005	30	9.36559	2081.53	2000.64	1702.44
1	0.0001	13	23.1905	2133.29	2057.29	1922.97
3	fixed	182	1.00000	2246.94	2171.41	393.95
3	0.001	44	6.09630	2178.23	2099.65	1642.52
3	0.0005	28	9.35632	1937.07	1861.52	1567.51
3	0.0001	13	23.5000	2149.93	2069.44	1928.67

Table 19: *stope* $80 \times 80 \times 80$ $\omega = 0.3$

Search direct. used	Inner stopping tolerance	Number of outer iterations	Av. number of inner iterations	Solution time (in sec.) total execution	solution	inner solver
Solution method: *gcgls* Preconditioner: dbilu						
1	fixed	82	1.00000	1010.74	915.53	179.01
1	0.001	28	4	1021.18	933.91	686.32
1	0.0005	23	5	978.41	888.63	683.90
1	0.0001	14	9	950.00	860.95	734.26
3	fixed	82	1	1034.42	949.17	194.67
3	0.001	27	4	1015.23	929.21	672.34
3	0.0005	22	5	962.42	872.64	661.95
3	0.0001	14	9	958.60	868.60	731.57
Solution method: *gcgls* Preconditioner: zebra						
1	fixed	160	1	1861.84	1774.15	429.57
1	0.001	53	4	1822.01	1734.47	1275.44
1	0.0005	60	4	2272.49	2182.71	1664.08
1	0.0001	31	8	1767.08	1679.44	1410.54
3	fixed	160	1	1960.10	1872.84	429.68
3	0.001	51	4	1794.92	1706.83	1235.25
3	0.0005	43	5	1749.64	1662.23	1265.65
3	0.0001	30	8	1783.97	1694.29	1413.99

Table 20: *stope* $80 \times 80 \times 80$ $\omega = 0.9$

Search direct. used	Inner stopping tolerance	Number of outer iterations	Av. number of inner iterations	Solution time (in sec.) total execution	solution	inner solver
Solution method: *gcgls* Preconditioner: dbilu						
1	fixed	43	1.00000	86.40	70.34	12.76
1	0.001	14	5.71111	92.52	77.24	57.66
1	0.0005	14	6.53333	100.64	85.70	65.78
1	0.0001	13	8.02381	114.16	98.44	79.02
3	fixed	43	1.00000	94.91	78.93	13.38
3	0.001	14	5.68889	99.39	83.18	61.11
3	0.0005	13	6.50000	102.52	86.62	65.60
3	0.0001	13	8.07143	116.80	100.11	79.35
Solution method: *gcgls* Preconditioner: zebra						
1	fixed	52	1.00000	105.78	89.77	16.27
1	0.001	16	5.39216	107.46	90.27	66.32
1	0.0005	15	6.08333	107.75	91.91	69.62
1	0.0001	13	8.16667	114.65	99.60	80.17
3	fixed	52	1.00000	110.17	94.99	16.33
3	0.001	15	5.35417	100.53	84.90	61.44
3	0.0005	14	6.11111	104.72	88.97	66.67
3	0.0001	13	8.14286	118.75	102.60	81.79

Table 21: *gate1* $102 \times 26 \times 33$ $\omega = 0.9$

Search direct. used	Inner stopping tolerance	Number of outer iterations	Av. number of inner iterations	Solution time (in sec.) total execution	solution	inner solver
Solution method: *gcgls* Preconditioner: dbilu						
1	fixed	54	1.00000	220.76	188.08	34.12
1	0.001	15	6.00000	216.43	184.99	139.69
1	0.0005	15	6.72917	229.01	197.95	153.32
1	0.0001	14	8.15556	248.44	217.66	175.25
3	fixed	54	1.00000	229.06	199.03	34.40
3	0.001	15	5.93750	214.06	183.87	136.43
3	0.0005	15	6.66667	226.89	197.04	150.42
3	0.0001	14	8.13333	243.45	213.86	170.24
Solution method: *gcgls* Preconditioner: zebra						
1	fixed	69	1.00000	264.31	233.77	43.76
1	0.001	16	5.74510	213.42	183.13	137.30
1	0.0005	15	6.72917	226.31	196.44	153.27
1	0.0001	15	8.31250	264.11	234.19	190.55
3	fixed	69	1.00000	274.99	245.11	43.44
3	0.001	16	5.74510	218.48	187.39	138.32
3	0.0005	15	6.66667	227.33	196.91	150.84
3	0.0001	15	8.29167	260.56	230.89	184.97

Table 22: *gate2* $202 \times 26 \times 33$ $\omega = 0.9$

Search direct. used	Inner stopping tolerance	Number of outer iterations	Av. number of inner iterations	Solution time (in sec.) total execution	solution	inner solver
Solution method: *gcgls* Preconditioner: dbilu						
1	fixed	72	1.00000	412.75	360.95	68.04
1	0.001	14	10.08889	414.75	366.86	307.12
1	0.0005	14	11.3333	451.23	403.72	343.94
1	0.0001	14	15.3111	560.99	515.38	456.34
3	fixed	72	1.00000	422.42	375.78	67.55
3	0.001	14	9.51111	401.38	353.53	290.54
3	0.0005	14	11.0889	451.71	404.76	340.47
3	0.0001	14	15.2222	572.03	520.80	456.62
Solution method: *gcgls* Preconditioner: zebra						
1	fixed	79	1.00000	444.65	394.91	75.27
1	0.001	14	11.0222	438.21	392.40	332.63
1	0.0005	14	12.2667	476.14	429.03	369.19
1	0.0001	14	15.9778	592.98	546.80	485.82
3	fixed	79	1.00000	473.24	424.46	76.90
3	0.001	14	10.46667	432.70	383.19	319.43
3	0.0005	14	12.3111	485.21	435.32	371.69
3	0.0001	14	15.9333	594.01	542.47	478.44

Table 23: *gate3* $302 \times 26 \times 33$ $\omega = 0.3$

Search direct. used	Inner stopping tolerance	Number of outer iterations	Av. number of inner iterations	Solution time (in sec.) total execution	solution	inner solver
Solution method: *gcgls* Preconditioner: dbilu						
1	fixed	65	1.00000	388.67	339.50	62.49
1	0.001	15	6.72917	345.89	299.79	232.65
1	0.0005	14	7.88889	362.88	315.71	252.87
1	0.0001	14	10.8667	450.63	404.34	342.05
3	fixed	65	1.00000	397.79	350.98	62.44
3	0.001	14	6.48889	313.50	267.28	203.38
3	0.0005	14	7.80000	352.28	307.33	243.39
3	0.0001	14	10.8667	447.91	400.91	336.58
Solution method: *gcgls* Preconditioner: zebra						
1	fixed	83	1.00000	464.30	417.96	78.73
1	0.001	16	6.31373	334.46	288.18	220.43
1	0.0005	15	7.08333	337.77	292.09	229.01
1	0.0001	14	11.4222	446.98	399.58	340.47
3	fixed	83	1.00000	478.42	432.59	78.79
3	0.001	16	6.03922	327.54	281.43	210.37
3	0.0005	15	7.77083	363.19	318.63	251.77
3	0.0001	14	11.4222	449.89	405.50	342.91

Table 24: *gate3* $302 \times 26 \times 33$ $\omega = 0.9$

ω	Inner stopping tolerance	Number of outer iterations	Av. number of inner iterations	Solution time (in sec.) total execution	solution	inner solver
Solution method: *gcgls* Preconditioner: dbilu						
0.2	fixed	110	1.00/ 1.00/ 1.00	319.70	253.72	14.27
0.2	0.001	16	6.25/ 3.62/ 7.38	133.88	66.26	29.00
0.2	0.0001	16	9.25/ 5.31/ 11.12	144.45	77.34	40.21
0.2	1e-05	15	10.87/ 7.33/ 12.60	145.19	78.67	43.60
0.3	fixed	111	1.00/ 1.00/ 1.00	351.47	248.88	14.52
0.3	0.001	17	5.65/ 3.82/ 6.94	135.87	69.02	29.63
0.3	0.0001	16	8.31/ 5.81/ 8.75	142.33	76.11	36.14
0.3	1e-05	15	10.67/ 8.13/ 10.60	144.69	77.66	42.45
0.4	fixed	111	1.00/ 1.00/ 1.00	312.74	246.58	14.19
0.4	0.001	17	5.41/ 3.88/ 6.18	133.87	67.35	27.92
0.4	0.0001	15	7.73/ 5.67/ 8.60	135.83	68.03	33.12
0.4	1e-05	15	11.27/ 8.60/ 9.87	148.34	80.85	44.18
0.5	fixed	103	1.00/ 1.00/ 1.00	304.32	237.30	13.64
0.5	0.001	17	5.47/ 4.12/ 6.18	134.50	67.89	28.36
0.5	0.0001	16	8.38/ 5.88/ 9.06	141.40	73.92	36.68
0.5	1e-05	15	11.67/ 8.27/ 10.93	146.29	79.07	44.08
0.9	fixed	108	1.00/ 1.00/ 1.00	327.19	261.30	14.37
0.9	0.001	18	5.61/ 3.94/ 6.06	138.91	71.75	30.14
0.9	0.0001	16	8.81/ 6.06/ 8.25	140.53	73.89	36.55
0.9	1e-05	15	11.00/ 7.93/ 10.73	142.68	77.13	42.02
Solution method: *gcgls* Preconditioner: zebra						
0.2	fixed	75	1.00/ 1.00/ 1.00	242.55	176.66	9.67
0.2	0.001	16	4.38/ 3.94/ 7.50	128.73	62.92	26.28
0.2	0.0001	16	8.12/ 5.56/ 11.88	142.91	76.49	39.47
0.2	1e-05	15	10.93/ 6.93/ 13.33	143.75	77.82	43.26
0.3	fixed	86	1.00/ 1.00/ 1.00	262.79	195.81	10.93
0.3	0.001	16	3.81/ 4.69/ 7.00	128.52	62.85	25.96
0.3	0.0001	16	6.00/ 6.50/ 10.62	139.47	72.41	35.70
0.3	1e-05	15	9.47/ 7.67/ 12.20	141.35	75.41	40.98
0.4	fixed	94	1.00/ 1.00/ 1.00	279.15	213.46	11.97
0.4	0.001	16	4.50/ 4.88/ 6.19	129.18	62.62	25.94
0.4	0.0001	16	6.44/ 7.31/ 9.94	140.59	73.11	36.40
0.4	1e-05	15	8.20/ 9.07/ 11.33	142.27	74.60	39.99
0.5	fixed	101	1.00/ 1.00/ 1.00	295.98	230.17	12.82
0.5	0.001	16	4.69/ 5.19/ 6.25	128.99	63.41	26.71
0.5	0.0001	16	6.81/ 7.62/ 10.25	140.41	74.35	37.69
0.5	1e-05	15	9.07/ 9.67/ 11.53	145.59	76.53	42.03
0.9	fixed	119	1.00/ 1.00/ 1.00	336.53	269.92	15.16
0.9	0.001	19	4.05/ 5.21/ 5.79	139.89	73.37	30.09
0.9	0.0001	16	7.56/ 9.12/ 10.31	143.85	77.35	40.59
0.9	1e-05	15	10.60/ 12.13/ 12.67	150.15	83.85	48.98

Table 25: *most* $32 \times 32 \times 32$, one search direction

ω	Inner stopping tolerance	Number of outer iterations	Av. number of inner iterations	Solution time (in sec.) total execution	solution	inner solver
Solution method: *gcgls* Preconditioner: dbilu						
0.2	fixed	110	1.00/ 1.00/ 1.00	320.35	252.09	14.04
0.2	0.001	16	5.94/ 3.62/ 6.75	133.38	65.02	27.49
0.2	0.0001	15	8.60/ 5.33/ 10.33	139.11	70.92	35.52
0.2	1e-05	15	11.00/ 7.40/ 12.07	146.17	78.46	43.26
0.3	fixed	111	1.00/ 1.00/ 1.00	319.83	253.65	14.17
0.3	0.001	16	5.56/ 3.88/ 5.94	131.90	63.82	26.22
0.3	0.0001	15	8.53/ 5.80/ 8.87	138.70	69.60	34.24
0.3	1e-05	15	10.73/ 8.13/ 11.33	144.42	77.40	42.54
0.4	fixed	111	1.00/ 1.00/ 1.00	322.38	255.58	14.27
0.4	0.001	17	5.41/ 3.82/ 5.65	133.27	66.94	27.23
0.4	0.0001	15	7.47/ 5.67/ 8.67	135.72	67.82	32.53
0.4	1e-05	15	11.73/ 8.73/ 10.60	144.97	78.47	43.60
0.5	fixed	103	1.00/ 1.00/ 1.00	308.46	240.41	13.28
0.5	0.001	17	4.82/ 4.00/ 5.71	134.09	66.43	26.71
0.5	0.0001	15	7.53/ 5.73/ 8.53	171.28	67.88	32.55
0.5	1e-05	15	11.67/ 8.47/ 10.27	143.00	76.99	42.34
0.9	fixed	108	1.00/ 1.00/ 1.00	319.67	251.82	13.89
0.9	0.001	17	5.06/ 4.24/ 5.53	133.49	67.00	27.17
0.9	0.0001	16	8.06/ 6.06/ 8.06	141.79	72.67	35.20
0.9	1e-05	15	11.00/ 7.93/ 9.60	141.16	74.78	40.05
Solution method: *gcgls* Preconditioner: zebra						
0.2	fixed	75	1.00/ 1.00/ 1.00	249.13	180.35	9.73
0.2	0.001	16	4.31/ 3.81/ 6.25	128.21	62.06	24.71
0.2	0.0001	15	8.07/ 5.53/ 10.73	137.46	69.78	35.05
0.2	1e-05	15	11.07/ 6.93/ 12.87	145.14	78.93	43.31
0.3	fixed	86	1.00/ 1.00/ 1.00	275.25	206.49	11.14
0.3	0.001	16	3.75/ 4.62/ 6.44	130.30	63.91	25.68
0.3	0.0001	16	6.06/ 6.50/ 9.94	138.87	71.87	35.02
0.3	1e-05	15	9.40/ 7.93/ 12.80	143.16	77.47	42.36
0.4	fixed	94	1.00/ 1.00/ 1.00	291.14	224.74	12.19
0.4	0.001	16	4.38/ 4.88/ 5.81	131.51	63.08	25.60
0.4	0.0001	16	6.44/ 7.31/ 9.56	140.16	73.69	36.38
0.4	1e-05	15	8.27/ 9.07/ 11.87	142.59	76.27	41.23
0.5	fixed	101	1.00/ 1.00/ 1.00	307.94	241.70	13.10
0.5	0.001	16	4.69/ 4.88/ 5.12	163.85	61.67	24.88
0.5	0.0001	15	6.73/ 7.47/ 9.27	135.29	69.34	34.17
0.5	1e-05	15	9.07/ 9.67/ 12.07	144.44	78.24	43.11
0.9	fixed	119	1.00/ 1.00/ 1.00	349.46	281.65	15.48
0.9	0.001	18	4.11/ 4.78/ 4.83	134.71	67.86	26.56
0.9	0.0001	16	7.69/ 9.12/ 8.06	143.28	75.58	38.29
0.9	1e-05	15	10.53/ 12.13/ 10.93	149.80	83.26	47.34

Table 26: *most* $32 \times 32 \times 32$, three search directions