Parallel Multilevel Solvers for 3D Problems

by

Michael Jung and Michael Theß

Technische Universität Chemnitz–Zwickau

Fakultät für Mathematik

e-mail: michael.jung@mathematik.tu-chemnitz.de
michael.thess@mathematik.tu-chemnitz.de

Contents

Boundary value problems of second-order

Find $u \in \mathcal{V}_0 = \{v \in H^1(\Omega) \,:\, v = 0 \text{ on } \Gamma_1\}$ such that

$$a(u,v) = \langle F, v \rangle$$

holds for all $v \in \mathcal{V}_0$, where

- Ω is a bounded three-dimensional domain,
- $\partial\Omega = \bar{\Gamma}_1 \cup \bar{\Gamma}_2, \; \Gamma_1 \cap \Gamma_2 = \emptyset$,
- $a(.,.)$ is a symmetric, $\mathcal{V}_0$-bounded, and $\mathcal{V}_0$-elliptic bilinear form,
- $\langle .,. \rangle \,:\, \mathcal{V}_0^* \times \mathcal{V}_0 \rightarrow R^1$ is a linear form.

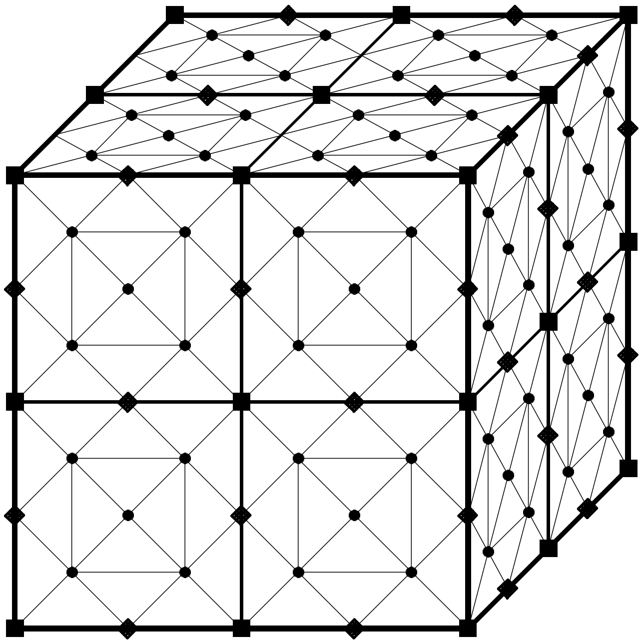
For example,

$$\begin{aligned} a(u,v) &= \int_\Omega a(x) \nabla^T v(x) \nabla u(x)\, dx \\ \langle F, v \rangle &= \int_\Omega f(x) v(x)\, dx \; + \int_{\Gamma_2} g_2(x) v(x)\, ds \end{aligned}$$

Finite element discretization

$$\overline{\Omega} = \bigcup_{i=1}^{p} \overline{\Omega}_i \,,\ \Omega_i \cap \Omega_j = \emptyset \quad \text{for} \quad i \neq j$$

Sequence of nested tetrahedral meshes $\mathcal{T}_q$



- ■ crosspoints (vertices) (V)
- ◆ edge coupling nodes (E)
- • face coupling nodes (F)
- inner nodes (I)

Finite element subspaces:

$$V_q = \text{span}\,\{\varphi_q^{(i)}\,,\ i = 1, 2, \ldots, N_q\} \qquad (\varphi_q^{(i)}\ :\ \text{piecewise linear})$$

$\Longrightarrow$ System of algebraic finite element equations

$$\begin{pmatrix} K_{q,V} & K_{q,VE} & K_{q,VF} & K_{q,VI} \\ K_{q,EV} & K_{q,E} & K_{q,EF} & K_{q,EI} \\ K_{q,FV} & K_{q,FE} & K_{q,F} & K_{q,FI} \\ K_{q,IV} & K_{q,IE} & K_{q,IF} & K_{q,I} \end{pmatrix} \begin{pmatrix} \underline{u}_{q,V} \\ \underline{u}_{q,E} \\ \underline{u}_{q,F} \\ \underline{u}_{q,I} \end{pmatrix} = \begin{pmatrix} \underline{f}_{q,V} \\ \underline{f}_{q,E} \\ \underline{f}_{q,F} \\ \underline{f}_{q,I} \end{pmatrix}$$

Distribution of vectors and matrices to the processors

$A_{q,i} \,:\, R^{N_q} \rightarrow R^{N_{q,i}}$

$(N_{q,i} \times N_q)$ boolean matrix which maps some global vector $\underline{v}_q \in R^{N_q}$ of nodal variables into the superelement vector $\underline{v}_{q,i} \in R^{N_{q,i}}$ of variables associated with the subdomain $\overline{\Omega}_i$ only.

Types of distribution of vectors to the processors

- Type I (overlapping)

  Vector $\underline{v}_q$ is stored on processor P_i as $\underline{v}_{q,i} = A_{q,i}\, \underline{v}_q$.

- Type II (adding)

  Vector $\underline{d}_q$ is stored on processor P_i as $\underline{d}_{q,i}$ such that

$$\underline{d}_q = \sum_{i=1}^{p} A_{q,i}^T\, \underline{d}_{q,i} \,.$$

Representation of the stiffness matrices:

$$K_q \;= \sum_{i=1}^{p} A_{q,i}^T K_{q,i} A_{q,i}\,, \qquad K_{q,i} \text{ is stored on processor } P_i$$

For example, within the multi-grid algorithm

- $\underline{u}_{q,k,*}$ (approximate solutions) and $\underline{\widetilde{w}}_{q,k}$ (corrections) are vectors of type I.
- $\underline{f}_q$ (right-hand side) as well as $\underline{d}_{q,k}$ (defects) are vectors of type II.

Multi-grid algorithm (q–grid algorithm)

Let an initial guess $\underline{u}_{q,k,0}$ be given.

1. Pre-smoothing

$$\underline{u}_{q,k,1} = G_q^V(\nu_{q,1}, K_q, \underline{f}_q, \underline{u}_{q,k,0})$$

2. Coarse-grid correction

   (a) Compute the defect

$$\underline{d}_{q,k} = \underline{f}_q - K_q \underline{u}_{q,k,1}\,.$$

   (b) Restrict the defect to the $(q-1)$–level mesh

$$\underline{d}_{q-1,k} = I_q^{q-1} \underline{d}_{q,k}\,.$$

   (c) Solve the coarse-grid system

$$K_{q-1} \underline{w}_{q-1,k} = \underline{d}_{q-1,k}$$

   by means of μ iteration steps of a $(q-1)$–grid algorithm (with the initial guess 0) if $q-1>1$, otherwise by means of a PCG algorithm or a direct method $\Longrightarrow \quad \underline{\widetilde{w}}_{q-1,k}\,.$

   (d) Interpolate $\underline{\widetilde{w}}_{q-1,k}$ onto the finer mesh $\mathcal{T}_q$ and correct the approximate $\underline{u}_{q,k,1}$

$$\underline{u}_{q,k,2} = \underline{u}_{q,k,1} + I_{q-1}^q \underline{\widetilde{w}}_{q-1,k}\,.$$

3. Post-smoothing

$$\underline{u}_{q,k,3} = G_q^N(\nu_{q,2}, K_q, \underline{f}_q, \underline{u}_{q,k,2})$$

Set $\quad \underline{u}_{q,k+1,0} = \underline{u}_{q,k,3}\,.$

Smoother of Gauss–Seidel type

$\underline{u}_q^{(j+1)} = G_q^{V/N}(K_q, \underline{f}_q, \underline{u}_q^{(j)})$:

(a) $K_{q,V}\underline{u}_{q,V}^{(j+1)} = \underline{f}_{q,V} - K_{q,VE}\underline{u}_{q,E}^{(j)} - K_{q,VF}\underline{u}_{q,F}^{(j)} - K_{q,VI}\underline{u}_{q,I}^{(j)}$

(b) $\underline{r}_{q,E}^{(j+1)} = \underline{f}_{q,E} - K_{q,EV}\underline{u}_{q,V}^{(j+1)} - K_{q,E}\underline{u}_{q,E}^{(j)} - K_{q,EF}\underline{u}_{q,F}^{(j)} - K_{q,EI}\underline{u}_{q,I}^{(j)}$

$$\left(D_{q,E} + L'_{q,E}\right)\underline{w}_{q,E}^{(j+1)} = \underline{r}_{q,E}^{(j+1)}$$

$$\underline{u}_{q,E}^{(j+1)} = \underline{u}_{q,E}^{(j)} + \underline{w}_{q,E}^{(j+1)}$$

(c) $\underline{r}_{q,F}^{(j+1)} = \underline{f}_{q,F} - K_{q,FV}\underline{u}_{q,V}^{(j+1)} - K_{q,FE}\underline{u}_{q,E}^{(j+1)} - K_{q,F}\underline{u}_{q,F}^{(j)} - K_{q,FI}\underline{u}_{q,I}^{(j)}$

$$\left(D_{q,F} + L'_{q,F}\right)\underline{w}_{q,F}^{(j+1)} = \underline{r}_{q,F}^{(j+1)}$$

$$\underline{u}_{q,F}^{(j+1)} = \underline{u}_{q,F}^{(j)} + \underline{w}_{q,F}^{(j+1)}$$

(d) $\underline{r}_{q,I}^{(j+1)} = \underline{f}_{q,I} - K_{IV}\underline{u}_{q,V}^{(j+1)} - K_{q,IE}\underline{u}_{q,E}^{(j+1)} - K_{q,IF}\underline{u}_{q,F}^{(j+1)} - U_{q,I}\underline{u}_{q,I}^{(j)}$

$$\left(D_{q,I} + L_{q,I}\right)\underline{u}_{q,I}^{(j+1)} = \underline{r}_{q,I}^{(j+1)}$$

$(K_{q,*} = L_{q,*} + D_{q,*} + U_{q,*} = (\text{◺}) + (\diagdown) + (\text{◹})\ (* = E,\ F \text{ or } I))$

$(D_{q,*} + L'_{q,*}) = \text{blockdiag}(D_{q,*} + L_{q,*}) \quad (* = E \text{ or } F)$

Within each iteration step:

1 crosspoint communication,
1 communication over the edge coupling nodes, and
1 communication over the face coupling nodes

Interpolation

Linear interpolation

$$I_{q-1}^{q} \; : \; R^{N_{q-1}} \; \to \; R^{N_q}$$

Mapping of a vector of overlapping type into a vector of overlapping type
$\Longrightarrow$ without communication

Restriction

$$I_q^{q-1} \; : \; R^{N_q} \; \to \; R^{N_{q-1}} \; , \; I_q^{q-1} = (I_{q-1}^{q})^T$$

Mapping of a vector of adding type into a vector of adding type
$\Longrightarrow$ without communication

Solving the coarse–grid systems $K_1 \underline{w}_{1,k} = \underline{d}_{1,k}$

- Storage of the matrix K_1 on processor P_0
- Cholesky factorization of the matrix K_1 before starting the MG algorithm
- Within each MG step:
  - All processors send their part of $\underline{d}_{1,k}$ to the processor P_0.
  - On P_0 the forward and backward substitution is performed.
  - Processor P_0 sends the solution $\underline{w}_{1,k}$ to the other processors.

Preconditioned conjugate gradient method

For all $i = 1, 2, \ldots, p$ do in parallel:

Initial step:

Determine the initial guess $\underline{\mathbf{u}}_{q,i}^{(0)}$, e.g. $\underline{\mathbf{u}}_{q,i}^{(0)} = 0$

$$\underline{r}_{q,i}^{(0)} = \underline{f}_{q,i} - K_{q,i}\underline{\mathbf{u}}_{q,i}^{(0)}$$

$$\underline{\mathbf{w}}_{q}^{(0)} = C_q^{-1} \sum_{i=1}^{p} A_{q,i}^T \underline{r}_{q,i}^{(0)}$$

$$\underline{\mathbf{s}}_{q,i}^{(0)} = \underline{\mathbf{w}}_{q,i}^{(0)}$$

$$(\underline{\mathbf{w}}_{q,i}^{(0)} = A_{q,i}\underline{\mathbf{w}}_{q}^{(0)})$$

$$\sigma^{(0)} = \sum_{i=1}^{p} (\underline{\mathbf{w}}_{q,i}^{(0)}, \underline{r}_{q,i}^{(0)})$$

Iteration steps $j = 1, 2, \ldots$

$$\underline{a}_{q,i}^{(j-1)} = K_{q,i}\underline{\mathbf{s}}_{q,i}^{(j-1)}$$

$$\delta^{(j)} = \sum_{i=1}^{p} (\underline{\mathbf{s}}_{q,i}^{(j-1)}, \underline{a}_{q,i}^{(j-1)})$$

$$\tau^{(j)} = \sigma^{(j-1)}/\delta^{(j)}$$

$$\underline{\mathbf{u}}_{q,i}^{(j)} = \underline{\mathbf{u}}_{q,i}^{(j-1)} + \tau^{(j)}\underline{\mathbf{s}}_{q,i}^{(j-1)}$$

$$\underline{r}_{q,i}^{(j)} = \underline{r}_{q,i}^{(j-1)} - \tau^{(j)}\underline{a}_{q,i}^{(j-1)}$$

$$\underline{\mathbf{w}}_{q}^{(j)} = C_q^{-1} \sum_{i=1}^{p} A_{q,i}^T \underline{r}_{q,i}^{(j)}$$

$$\sigma^{(j)} = \sum_{i=1}^{p} (\underline{\mathbf{w}}_{q,i}^{(j)}, \underline{r}_{q,i}^{(j)})$$

$$\beta^{(j)} = \sigma^{(j)}/\sigma^{(j-1)}$$

$$\underline{\mathbf{s}}_{q,i}^{(j)} = \underline{\mathbf{w}}_{q,i}^{(j)} + \beta^{(j)}\underline{\mathbf{s}}_{q,i}^{(j-1)}$$

$$\sigma^{(j)} \leq \varepsilon^2\sigma^{(0)} \rightarrow \text{STOP}$$

- $\underline{\mathbf{u}}_{q,i}^{(j)}$, $\underline{\mathbf{w}}_{q,i}^{(j)}$, and $\underline{\mathbf{s}}_{q,i}^{(j)}$ are vectors of overlapping type
- $\underline{r}_{q,i}^{(j)}$ and $\underline{a}_{q,i}^{(j)}$ are vectors of adding type
- $\sum_{i=1}^{p}$ requires communication

MG(m) preconditioner

Assumptions for $q = 2, 3, \ldots, l$:

- $K_{q-1} = I_q^{q-1} K_q I_{q-1}^q$
- $K_q (S_q^V)^{\nu_{q,1}} = ((S_q^N)^{\nu_{q,2}})^T K_q$

  S_q^V (S_q^N) is the error transmission operator of the pre-(post-) smoothing procedure
- $I_q^{q-1} = (I_{q-1}^q)^T$
- $\rho(M_q) = \|M_q\|_{K_q} \leq \eta = const. < 1$

  M_q is the error transmission operator of the q–grid method

We choose a priori the preconditioner $\widetilde{C}_q = K_q$, and we solve the systems $K_q \underline{w}_q^{(j)} = \underline{r}_q^{(j)}$ within each PCG iteration step by means of m iteration steps of a q–grid method (with the initial guess 0).

$\Longrightarrow$ This can be interpreted as the preconditioner

$$\boxed{C_q = K_q (I_q - M_q^m)^{-1}}$$

Statements:

- $C_q = K_q (I_q - M_q^m)^{-1}$ is symmetric and positive definite.
- Generalized condition number:

$$\boxed{\kappa(C_q^{-1} K_q) = (1 - \eta^m)^{-1} = O(1)}$$

BPX preconditioner

Nodal basis for the levels $q = 1, 2, \dots, l$:

$$\Phi_q = \{\varphi_q^{(1)}, \varphi_q^{(2)}, \dots, \varphi_q^{(N_q)}\}$$

Generating system:

$$\Phi_q^G = \{\varphi_1^{(1)}, \varphi_1^{(2)}, \dots, \varphi_1^{(N_1)}, \varphi_2^{(1)}, \varphi_2^{(2)}, \dots, \varphi_2^{(N_2)}, \dots, \varphi_l^{(1)}, \varphi_l^{(2)}, \dots, \varphi_l^{(N_l)}\}$$

Relation between Φ_q^G and Φ_q:

$$\boxed{\Phi_q^G = \Phi_q V_q^G}$$

with

$$V_q^G = (S_1^q \;\; S_2^q \;\; \dots \;\; S_{q-1}^q \;\; I_q)$$

$$S_k^q = I_{q-1}^q I_{q-2}^{q-1} \cdots I_k^{k+1}, \;\; k = 1, 2, \dots, q-1$$

(I_k^{k+1} are the interpolation operators used within the multi-grid algorithm.)

Preconditioner:

$$\boxed{C_q^{-1} = V_q^G D_q^{-1} (V_q^G)^T}$$

$$D_q = \mathrm{diag}\{K_1 \;\; \mathrm{diag}\{K_2\} \;\; \dots \;\; \mathrm{diag}\{K_q\}\}$$

Generalized condition number:

$$\boxed{\kappa(C_q^{-1} K_q) = O(1)}$$

Scaled efficiency

$$S(p_1, p_2) = p_1 \frac{T_{sol}(p_1)}{N(p_1)} \Big/ p_2 \frac{T_{sol}(p_2)}{N(p_2)} \quad (p_1 < p_2)$$

$N(p_i)$ is the number of unknowns of the problem solved on p_i processors $(i = 1, 2)$.

$T_{sol}(p_i)$ is the time needed for the solution of the problem.

We expect that we can solve a problem with

$$N(p_2) = c_N N(p_1) \text{ unknowns } (c_N > 1)$$

on

$$p_2 = c_p\, p_1 \text{ processors } (c_p > 1)$$

in the time

$$c_N c_p^{-1} T_{sol}(p_1).$$

The following relation holds between this ideal time and the time $T_{sol}(p_2)$

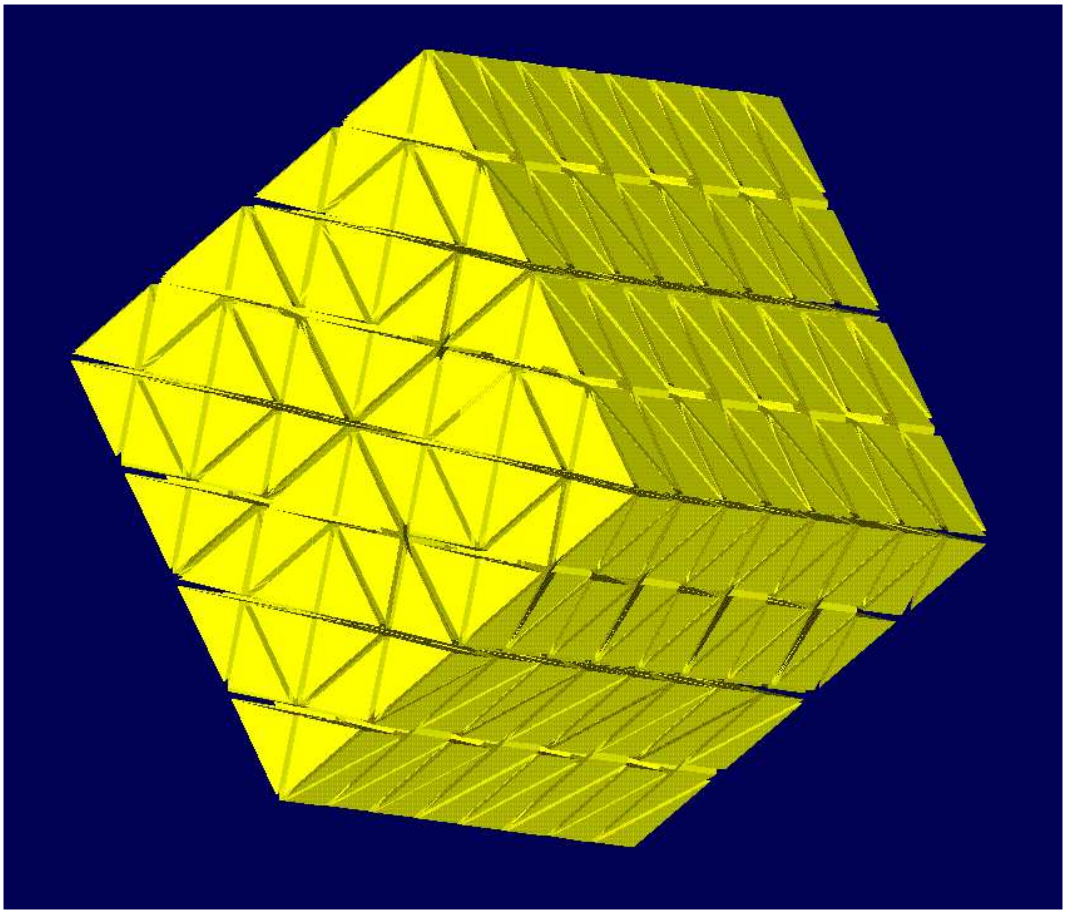
$$c_N c_p^{-1} T_{sol}(p_1) = S(p_1, p_2)\, T_{sol}(p_2).$$

Poisson's equation

Find $u \in \mathcal{V}_0$ such that

$$\int_\Omega \nabla^T v(x)\, \nabla u(x)\, dx = \int_\Omega v(x)\, dx \quad \forall\, v \in \mathcal{V}_0$$

holds with $\mathcal{V}_0 = \{u \in H^1(\Omega)\ : u = 0 \text{ on } \partial\Omega\}$.



Multi-grid algorithm

- V-cycle with 2 pre- and 2 post-smoothing steps (Gauss–Seidel)
- coarse-grid solver: Cholesky factorization
- relative accuracy: 10^{-4} , initial guess : 0
- GC/PP-128 machine

levels	2	3	4	5	S(1,p)
#unknowns	369	2465	17985	137345	
#iterations	7	7	7	7	
1 proc.	0.13	1.35			
2 proc.	0.25 (40)	1.44 (42)	6.03 (19)		0.82
4 proc.	0.28 (68)	1.03 (55)	3.50 (29)		0.70
8 proc.	0.31 (84)	0.89 (73)	2.37 (46)		0.52
16 proc.	0.38 (89)	0.96 (80)	1.99 (64)	6.47 (32)	0.73
32 proc.	0.58 (93)	1.27 (91)	2.32 (82)	5.26 (57)	0.45
64 proc.	0.73 (96)	1.49 (94)	2.46 (89)	4.56 (73)	0.26

CPU time [sec] (communication) [%]

Michael Jung and Michael Theß

MG(1)–PCG algorithm

- V-cycle with 1 pre- and 1 post-smoothing step (Gauss–Seidel)
- coarse-grid solver: Cholesky factorization
- relative accuracy: 10^{-4} , initial guess : 0
- GC/PP-128 machine

levels	2	3	4	5	S(1,p)
#unknowns	369	2465	17985	137345	
#iterations	6	6	6	6	
1 proc.	0.07	0.76			
2 proc.	0.12 (42)	0.71 (35)	3.34 (15)		0.83
4 proc.	0.13 (69)	0.50 (52)	1.91 (25)		0.73
8 proc.	0.16 (87)	0.43 (70)	1.25 (42)		0.55
16 proc.	0.20 (90)	0.47 (79)	1.02 (61)	3.57 (28)	0.74
32 proc.	0.29 (93)	0.61 (90)	1.13 (80)	2.75 (51)	0.48
64 proc.	0.36 (97)	0.72 (93)	1.19 (87)	2.28 (68)	0.29

CPU time [sec] (communication) [%]

BPX–PCG algorithm

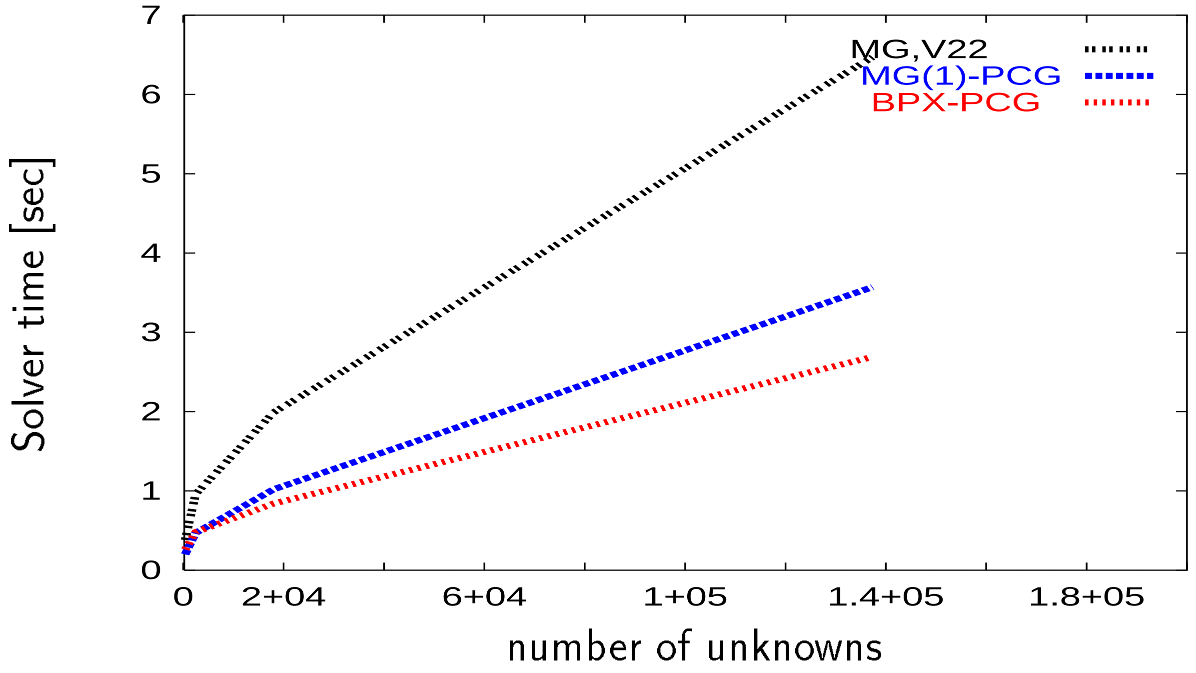
- with multilevel diagonal scaling
- coarse-grid solver: Cholesky factorization
- relative accuracy: 10^{-4} , initial guess : 0
- GC/PP-128 machine

levels	2	3	4	5	S(1,p)
#unknowns	369	2465	17985	137345	
#iterations	12	18	20	22	
1 proc.	0.11	0.46			
2 proc.	0.21 (29)	0.47 (32)	2.01 (15)		0.83
4 proc.	0.23 (43)	1.11 (19)	1.36 (27)		0.62
8 proc.	0.29 (55)	0.50 (60)	1.04 (45)		0.40
16 proc.	0.26 (81)	0.48 (83)	0.84 (67)	2.69 (40)	0.60
32 proc.	0.52 (61)	0.63 (87)	0.94 (80)	2.22 (58)	0.36
64 proc.	0.45 (91)	0.78 (91)	1.10 (87)	2.09 (75)	0.19

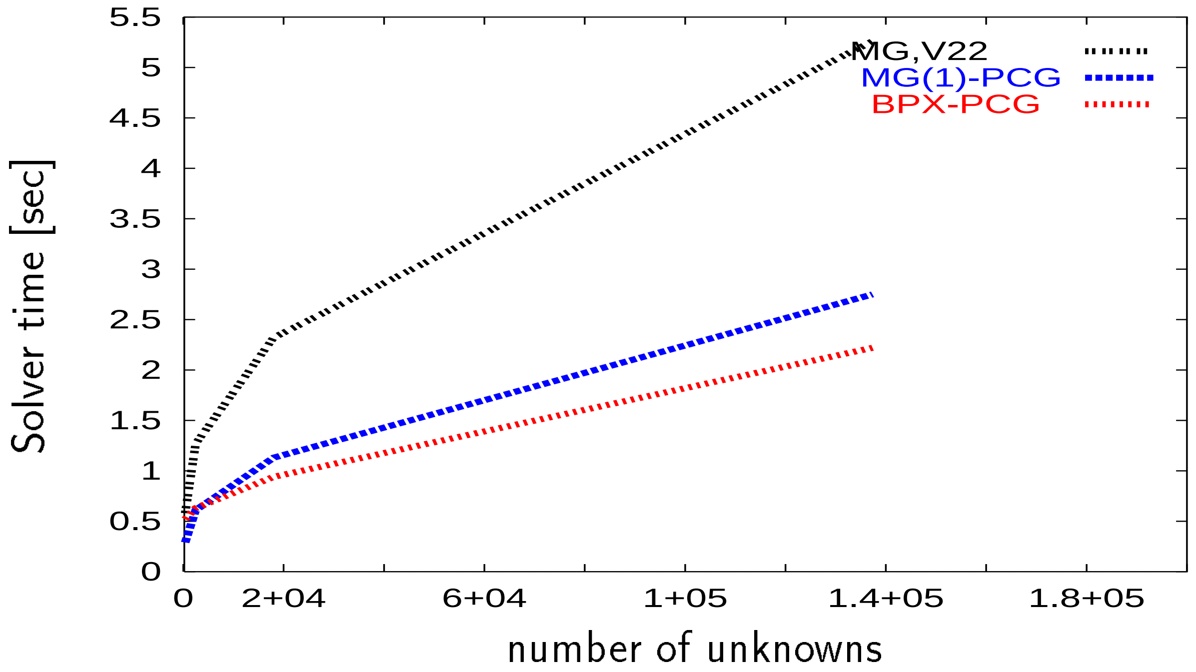
CPU time [sec] (communication) [%]

Poisson's equation

16 processors – "cube" domain



32 processors – "cube" domain



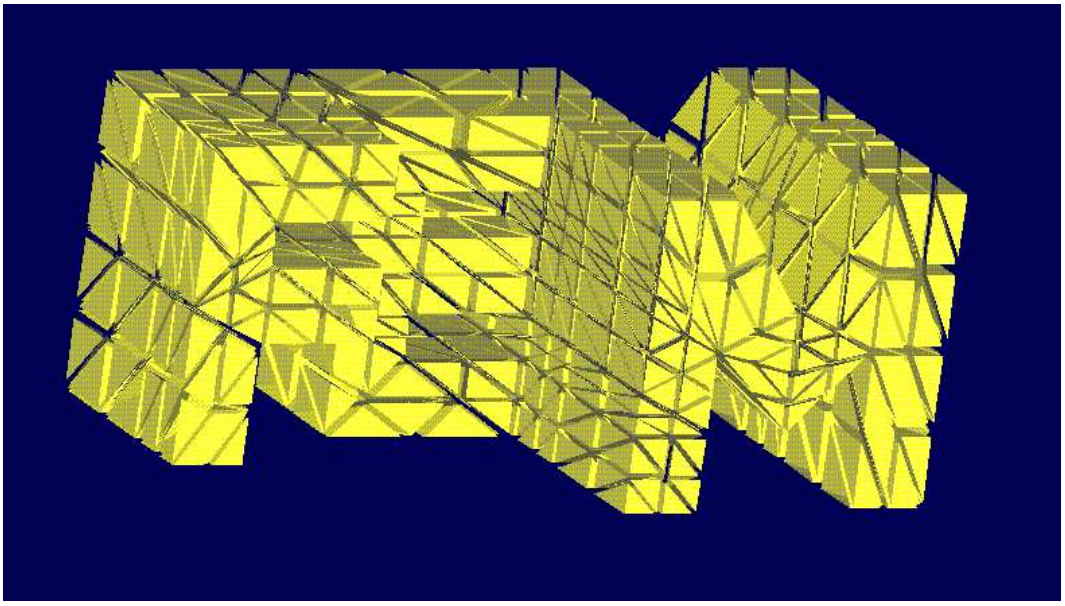
Poisson's equation

Find $u \in \mathcal{V}_0$ such that

$$\int_\Omega \nabla^T v(x)\, \nabla u(x)\, dx = \int_\Omega (-6x_3)\, v(x)\, dx \quad \forall\, v \in \mathcal{V}_0$$

holds with $\mathcal{V}_0 = \{u \in H^1(\Omega)\ :\ u = 0 \text{ on } \Gamma_D\}$.

$$\Gamma_D = \{x = (x_1, x_2, x_3)\ :\ x_3 = 0\}$$



Multi-grid algorithm

- V-cycle with 2 pre- and 2 post-smoothing steps (Gauss–Seidel)
- coarse-grid solver: Cholesky factorization
- relative accuracy: 10^{-4} , initial guess : 0
- GC/PP-128 machine

levels	2	3	4	5	S(1,p)
#unknowns	631	3848	26386	194342	
#iterations	15	15	16	18	
1 proc.	0.48	4.62			
2 proc.	0.86 (34)	4.38 (34)	19.60 (16)		0.81
4 proc.	0.94 (49)	3.68 (50)	12.68 (29)		0.62
8 proc.	1.16 (67)	3.74 (69)	10.12 (51)		0.39
16 proc.	1.30 (78)	3.75 (80)	8.70 (66)	28.93 (40)	0.50
32 proc.	1.66 (87)	3.88 (86)	7.81 (78)	20.21 (54)	0.36
64 proc.	2.11 (90)	4.44 (90)	8.26 (86)	17.80 (69)	0.20

CPU time [sec] (communication) [%]

MG(1)–PCG algorithm

- V-cycle with 1 pre- and 1 post-smoothing step (Gauss–Seidel)
- coarse-grid solver: Cholesky factorization
- relative accuracy: 10^{-4} , initial guess : 0
- GC/PP-128 machine

levels	2	3	4	5	S(1,p)
#unknowns	631	3848	26386	194342	
#iterations	9	8	9	9	
1 proc.	0.18	1.57			
2 proc.	0.28 (36)	1.34 (29)	6.84 (13)		0.79
4 proc.	0.30 (50)	1.06 (46)	4.24 (25)		0.63
8 proc.	0.36 (69)	1.05 (67)	3.22 (46)		0.42
16 proc.	0.42 (81)	1.05 (78)	2.66 (62)	8.35 (33)	0.59
32 proc.	0.54 (89)	1.10 (85)	2.37 (75)	5.80 (48)	0.43
64 proc.	0.68 (93)	1.28 (90)	2.50 (85)	4.96 (65)	0.25

CPU time [sec] (communication) [%]

BPX–PCG algorithm

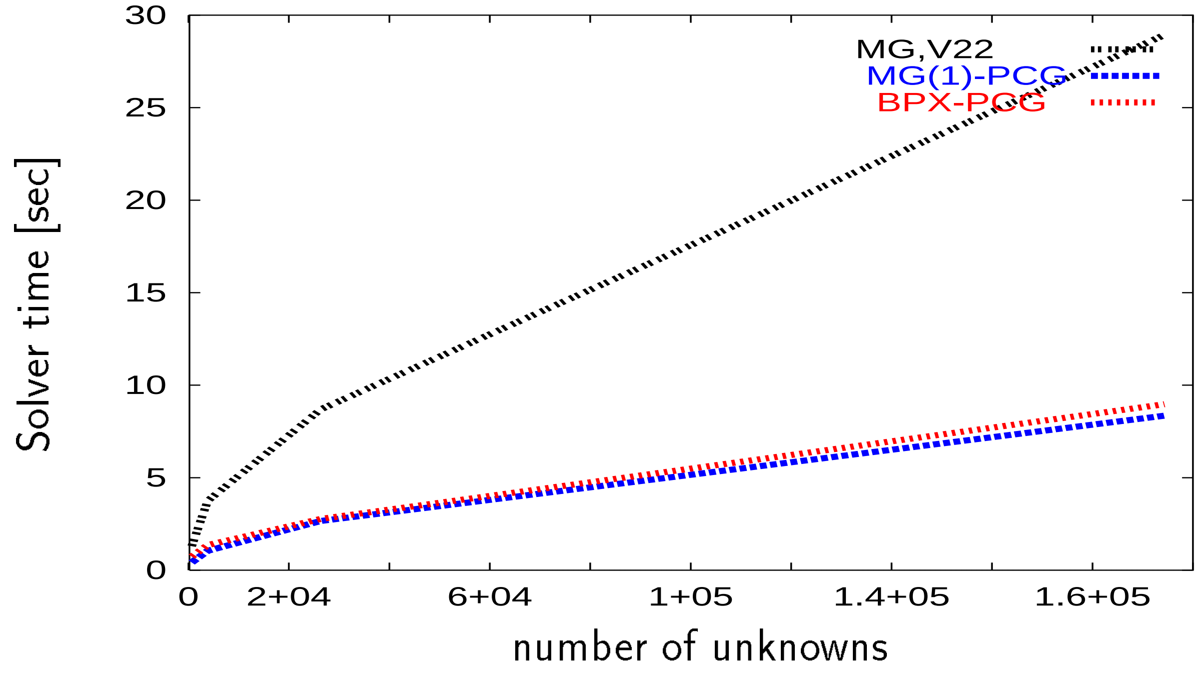
- with multilevel diagonal scaling
- coarse-grid solver: Cholesky factorization
- relative accuracy: 10^{-4} , initial guess : 0
- GC/PP-128 machine

levels	2	3	4	5	S(1,p)
#unknowns	631	3848	26386	194342	
#iterations	29	35	39	41	
1 proc.	0.32	1.30			
2 proc.	0.45 (35)	1.22 (27)	5.23 (13)		0.85
4 proc.	0.44 (66)	1.02 (53)	3.47 (31)		0.64
8 proc.	0.74 (63)	1.37 (63)	3.18 (52)		0.35
16 proc.	0.74 (85)	1.36 (82)	2.77 (72)	8.97 (47)	0.46
32 proc.	0.98 (90)	1.69 (88)	3.04 (83)	7.58 (66)	0.27
64 proc.	1.27 (91)	2.08 (90)	3.37 (88)	6.93 (76)	0.15

CPU time [sec] (communication) [%]

Poisson's equation

16 processors – "FEM" domain



32 processors – "FEM" domain