National Conference with International Participation

ENGINEERING MECHANICS 2008

Svratka, Czech Republic, May 12 – 15, 2008

SCALABILITY OF DOMAIN DECOMPOSITION METHODS

J. Brož*, J. Kruis**

Summary: *Scalability of the FETI method depends on applied preconditioner. There are two basic preconditioners for the FETI method – the Dirichlet preconditioner and the lumped preconditioner. The Dirichlet preconditioner is mathematically optimal but its time requirements are greater than requirements of the lumped preconditioner. This paper deals with implementation of the lumped preconditioner into an open source computer code. The implementation is based on the MPI library and is intended for parallel processing. Behaviour of the FETI method is shown on several two dimensional and three dimensional numerical examples. The test have been computed on a cluster of PCs.*

1. Introduction

Large scale problems are in the centre of attention of scientific and engineering community at this time. Very complex and detailed models of whole structures are used in numerical analysis. The most spread numerical method seems to be the finite element method. It converts the original problem described by the system of partial differential equations or by the minimisation of suitable functional to the solution of a system of algebraic equations. The number of equations depends on the applied mesh of finite elements. The finer mesh is used, the more equations are generated.

There are basically two groups of methods for the solution of system of algebraic equations. The direct methods are based on the Gaussian elimination. The second group of the methods contains iterative methods. The conjugate gradient method or the GMRES method are examples of iterative methods. The advantage of the direct method consists in the fact, that the number of arithmetic operations and storage requirements are known in advance. The disadvantage of the direct method resides in larger storage requirements in comparison to the iterative methods. The disadvantage of iterative methods rests in the unknown number of iterations necessary for an acceptable error. There are problems, where the number of iterations is very high and a preconditioning has to be used.

At this time, domain decomposition methods are very popular for solution of large systems of algebraic equations on parallel computers. There are many variants of the domain decomposition methods. This paper concentrates only on nonoverlapping methods, especially

[*] Ing. Jaroslav Brož.: Department of Mechanics, Faculty of Civil Engineering, Czech Technical University in Prague, Thákurova 7; 16629 Prague; tel.: +420224355417, fax.: +420224310775; e-mail: broz@cml.fsv.cvut.cz

[**] doc. Ing. Jaroslav Kruis, Ph. D.: Department of Mechanics, Faculty of Civil Engineering, Czech Technical University in Prague, Thákurova 7; 16629 Prague; tel.: +420224354580, fax.: +42024310775; e-mail: jk@cml.fsv.cvut.cz

on the FETI (Finite Element Tearing and Interconnecting) method introduced by Farhat & Roux (1991).

2. Finite Element Tearing and Interconnecting method

2.1. Introduction to the FETI method

The FETI method is based on a decomposition of the original domain into smaller subdomains, where the continuity is enforced by Lagrange multipliers. The original unknowns are eliminated and the coarse problem which contains the dual variables is obtained. The matrices of the subdomains are symmetric, positive definite or semidefinite with respect to the constraints applied to a particular subdomain. Unsupported or partially supported subdomain leads to a singular matrix and inverse of the matrix have to be replaced by a pseudoinverse matrix. The coarse problem is not positive definite. Hence the classical conjugate gradient method for solution of the problem cannot be used. The modified conjugate gradient method has to be applied for solution. More details about FETI method can be found in reference Farhat & Roux (1994).

2.2. Preconditioning of the FETI method

This paper deals with scalability of the FETI method with suitable preconditioners. In the literature (e.g. Farhat & Roux (1994)), there are two basic preconditioners for the FETI method. The optimal preconditioner which is called Dirichlet preconditioner and the economical preconditioner which is called lumped preconditioner.

In order to achieve parallel scalability, the preconditioners are necessary. The economical lumped preconditioner is based on matrix-vector multiplication. Submatrix defined by interface unknowns is selected and interface parts of subdomain vectors are multiplied by this matrix

$$\left(\bar{F}_I^{L}\right)^{-1} = \sum_{s=1}^{n} B^{(s)} \begin{bmatrix} 0 & 0 \\ 0 & K_{bb}^{(s)} \end{bmatrix} B^{(s)T} \quad (1),$$

where n denotes the number of subdomains, $B^{(s)}$ denotes a Boolean matrix of the s-th subdomain.

The Dirichlet preconditioner is more complicated because it is based on the Schur complements. Preliminary step therefore contains computation of the Schur complements which may be time and memory consuming. Ordering of nodes and unknowns is not arbitrary and is similar to the ordering used in the substructuring method. During the final phase of the FETI method, interface parts of vectors are multiplied by the Schur complements in each iteration

$$\left(\bar{F}_I^{D}\right)^{-1} = \sum_{s=1}^{n} B^{(s)} \begin{bmatrix} 0 & 0 \\ 0 & K_{bb}^{(s)} - K_{ib}^{(s)T} K_{ii}^{(s)-1} K_{ib}^{(s)} \end{bmatrix} B^{(s)T} \quad (2),$$

where n denotes the number of subdomains, $B^{(s)}$ denotes a Boolean matrix of the s-th subdomain and $K_{bb}^{(s)} - K_{ib}^{(s)T} K_{ii}^{(s)-1} K_{ib}^{(s)}$ is the Schur complement of subdomain s.

2.3. Implementation of preconditioners into SIFEL code

The SIFEL code is developed at the Department of Mechanics of Civil Engineering Faculty of the Czech Technical University (see reference SIFEL). It is an open source code for mechanical, transport and coupled problems written in C language. The code works on a single-processor as well as multiprocessor computer. The parallel version of the code is based on the MPI library and distributed memory architecture is considered.

The FETI method has been implemented earlier but without preconditioners. The lumped preconditioner has been implemented recently. The lumped version has been selected beacuse it is easier in comparison with the Dirichlet preconditioner. Multiplicity of the interface degrees of freedom has to be obtained because it is used in the scaling matrix. The multiplicity of interface DOF is the number of subdomains which share the DOF. In order to obtain an efficient algorithm, the nodal multiplicity, which describes the number of subdomains sharing the node, is assembled first and the DOF multiplicity in the second step. All vectors used on subdomains are rearranged because the components are split to internal or interface components.

3. Numerical Experiments

3.1. Simple 2D example

A rectangular domain was chosen for a numerical testing of the lumped preconditioner of the FETI method in two dimensions. The domain was discretized by the quadrilateral finite elements with two DOFs in each node. The software T3D (see T3D), which is developed at the Department of Mechanics at the FCE CTU in Prague, was used for discretization. Six different sizes of the mesh were used (50 x 50 elements, 100 x 100 elements, 150 x 150 elements, 200 x 200 elements, 250 x 250 elements and 300 x 300 elements). The software t3d2psifel (see T3D2PSIFEL) was used as a mesh decomposer for dividing the finite element mesh into several parts. Each finite element mesh was divided into 2 – 20 parts. Dirichlet boundary conditions were defined at nodes with the x-coordinate equal to zero. Prescribed loads were modelled by nodal forces located at nodes with maximum value of x-coordinate. Linear elasticity and plane stress were considered for testing.

Results obtained from numerical experiments are summarized in graphs on Fig. 1 - Fig. 10 and Tab. 1 – Tab. 5. The number of iterations with respect to the number of nodes is shown on Fig. 1, Fig. 3, Fig. 5, Fig. 7 and Fig. 9. The time of whole solution with respect to the number of nodes is shown on Fig. 2, Fig. 4, Fig. 6, Fig. 8 and Fig. 10. The following notation is used in tables:

- NN denotes the number of nodes
- NE denotes the number of elements
- NDOF denotes the number of degrees of freedom
- NME denotes the number of matrix entries
- ITER denotes the number of iterations

- TIME denotes the time of whole solution of problem

The results obtained from numerical examples clearly show that the lumped preconditioner reduces the number of iterations in comparison with the FETI method without preconditioner. Simultaneously, the preconditioned FETI method takes shorter time of solution than unpreconditioned method which is not always true. The number of iterations grows only slightly with respect to the number of unknowns. The scalability of the method can be proved with the help of data summarized in Tab. 1 – 5.

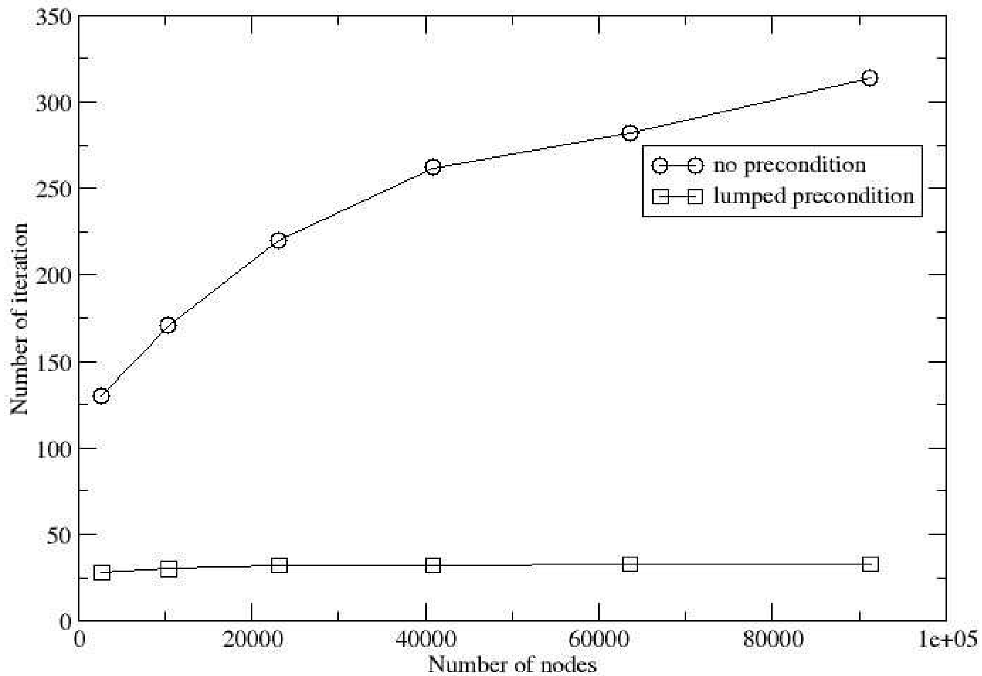

Fig. 1: Number of iterations for 4 subdomains

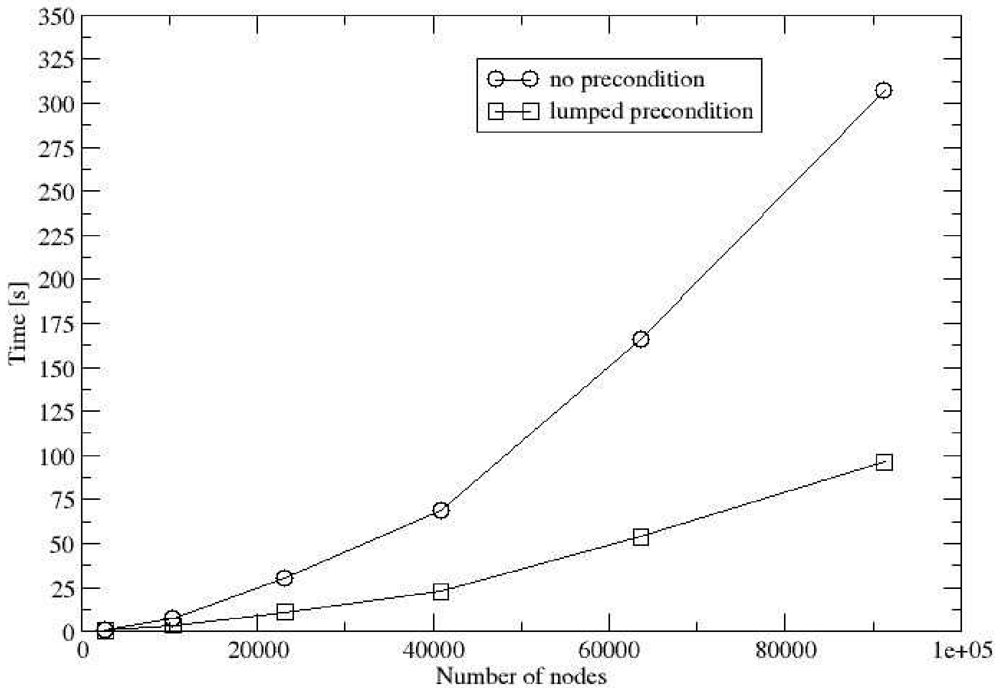

Fig. 2: Time of solution for 4 subdomains

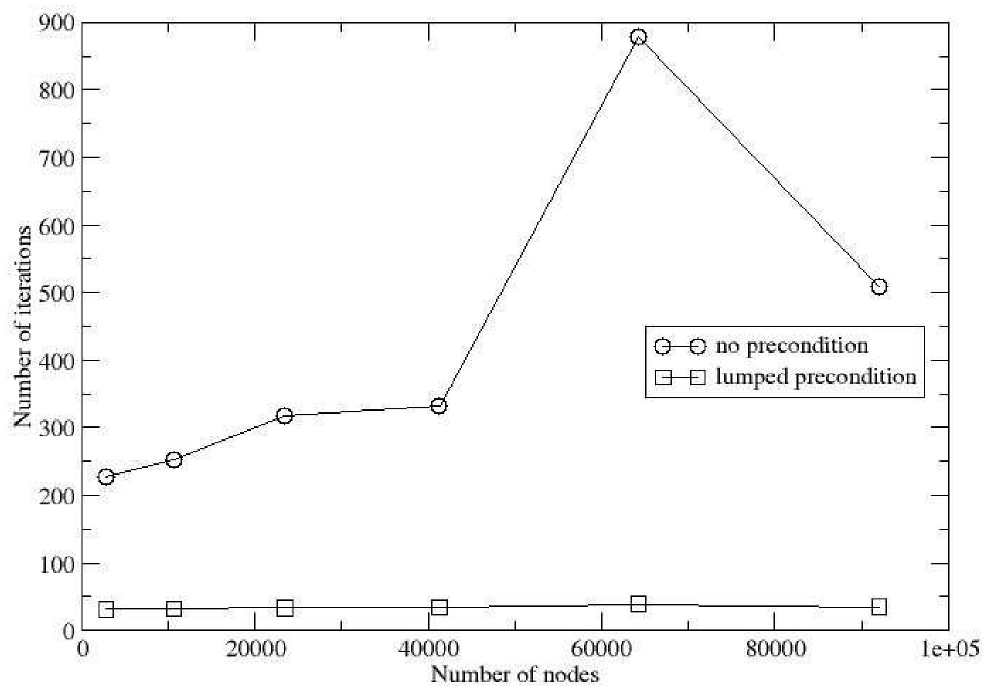

Fig. 3: Number of iterations for 8 subdomains

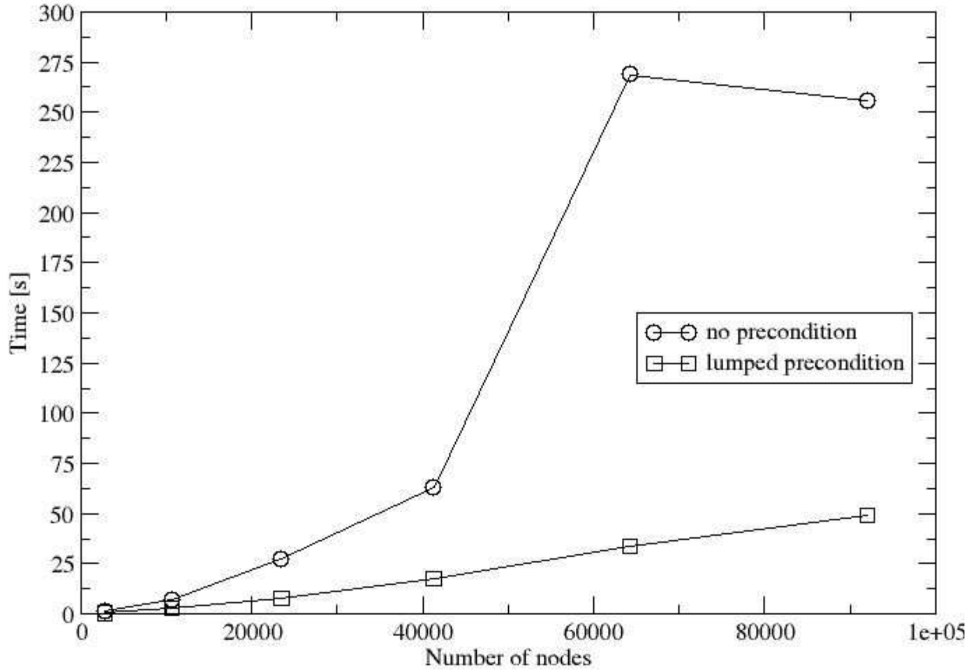

Fig. 4: Time of solution for 8 subdomains

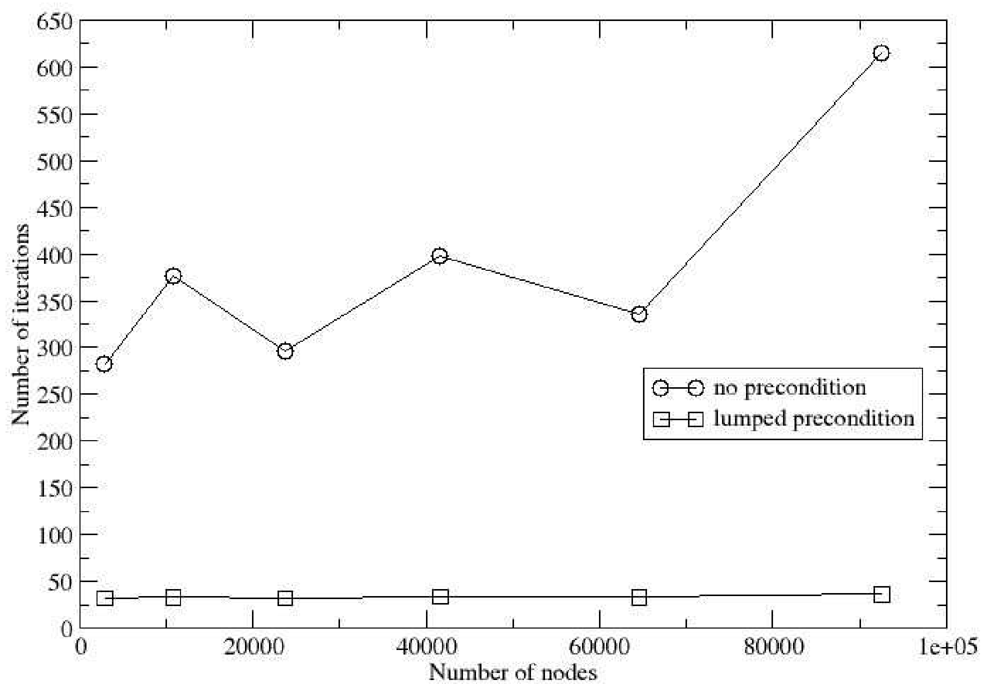

Fig. 5: Number of iterations for 12 subdomains

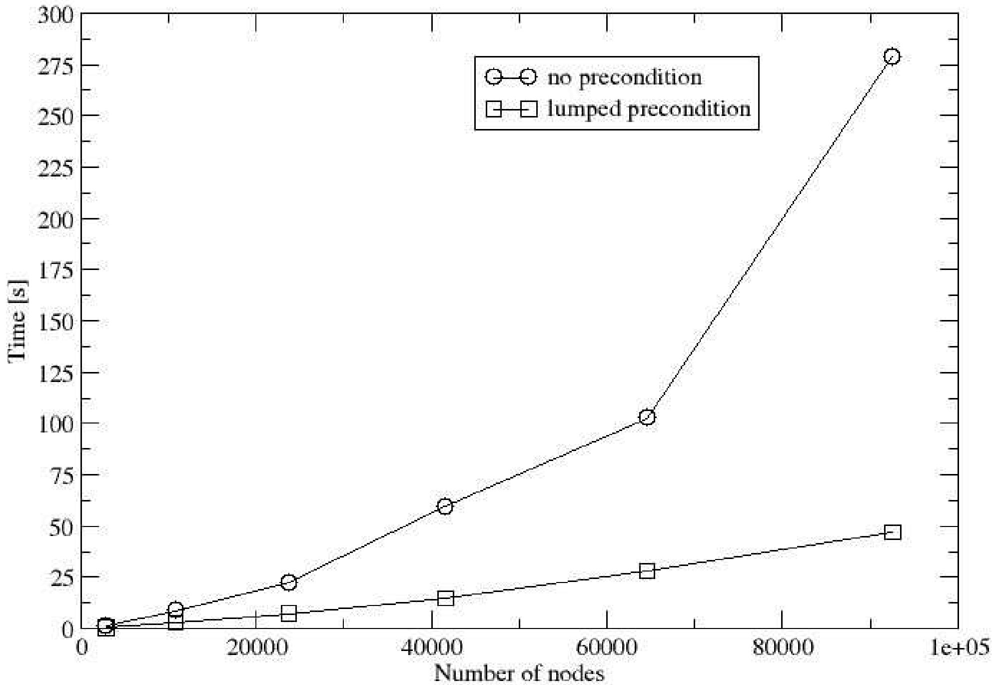

Fig. 6: Time of solution for 12 subdomains

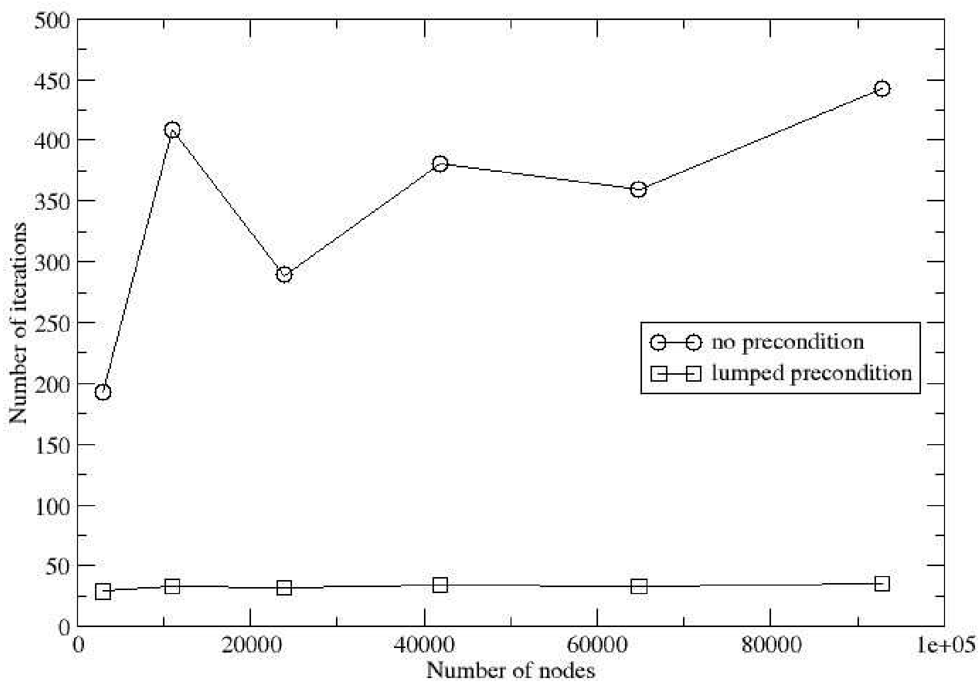

Fig. 7: Number of iterations for 16 subdomains

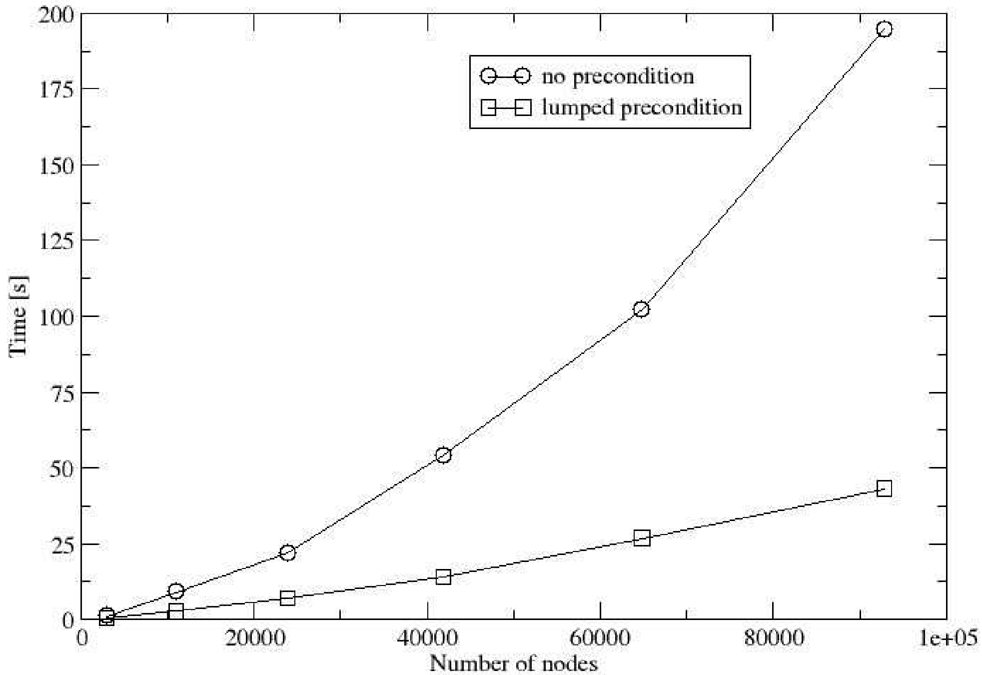

Fig. 8: Time of solution for 16 subdomains

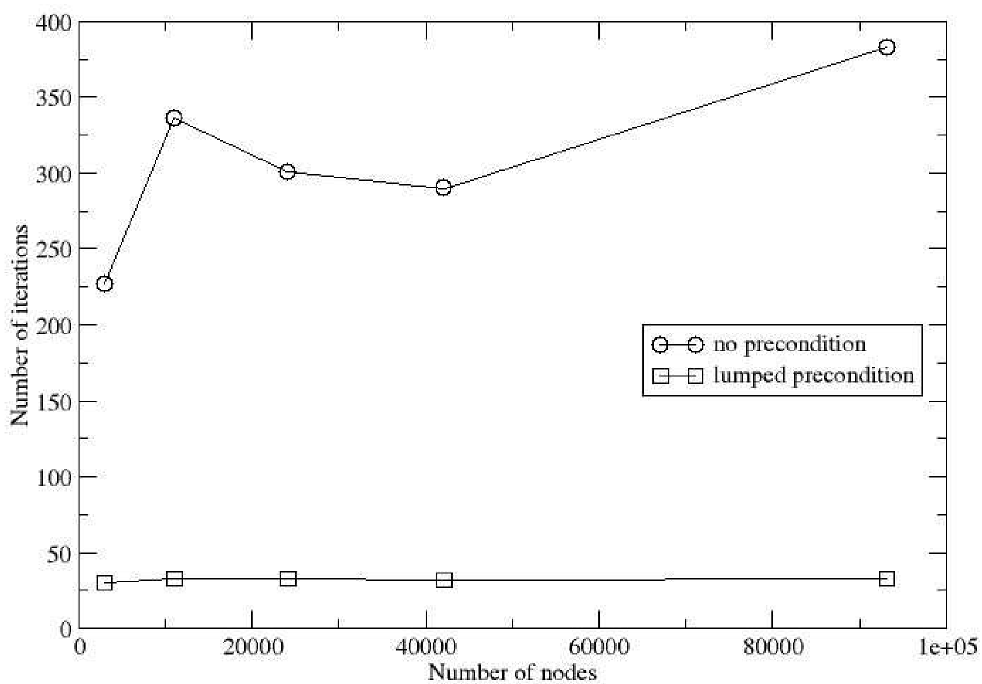

Fig. 9 : Number of iterations for 20 subdomains

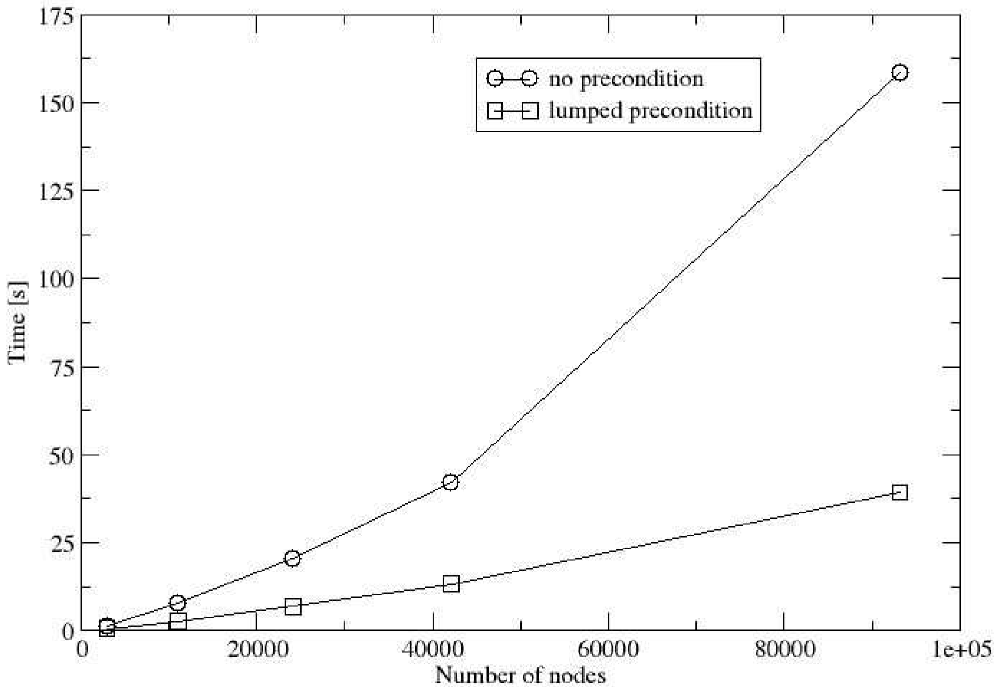

Fig. 10: Time of solution for 20 subdomains

Tab. 1: Decomposition of the problem into 4 subdomains

NN	NE	NDOF	NME	No precond.		Lumped precond.	
				ITER	TIME [s]	ITER	TIME [s]
10420	10000	20636	191010	171	7.610	30	3.400
23146	22500	45988	429142	220	30.020	32	11.090
40900	40000	81396	762470	262	68.710	32	22.680
63589	62500	126674	1190343	282	166.050	33	53.710
91306	90000	182008	1713412	314	307.540	33	96.200

Tab. 2 : Decomposition of the problem into 8 subdomains

NN	NE	NDOF	NME	No precond.		Lumped precond.	
				ITER	TIME [s]	ITER	TIME [s]
10627	10000	21048	192436	253	7.400	32	2.900
23463	22500	46620	431338	317	27.140	33	8.050
41294	40000	82182	765205	333	63.310	34	17.740
64238	62500	127968	1194856	879	268.630	39	33.830
92000	90000	183394	1718247	509	255.740	35	49.310

Tab. 3 : Decomposition of the problem into 12subdomains

NN	NE	NDOF	NME	No precond.		Lumped precond.	
				ITER	TIME [s]	ITER	TIME [s]
10823	10000	21438	193785	377	8.880	33	2.770
23748	22500	47190	433317	296	22.800	32	7.400
41608	40000	82808	767380	398	59.390	34	14.720
64611	62500	128716	1197458	335	102.700	33	28.190
92591	90000	184574	1722361	615	278.700	36	46.970

Tab. 4 : Decomposition of the problem into 16 subdomains

NN	NE	NDOF	NME	No precond.		Lumped precond.	
				ITER	TIME [s]	ITER	TIME [s]
23897	22500	47486	434337	289	21.820	32	7.210
41886	40000	83364	769310	381	54.310	34	14.120
64832	62500	129156	1198982	360	102.150	33	26.930
92853	90000	185098	1724179	442	194.550	35	43.270

Tab. 5 : Decomposition of the problem into 20 subdomains

NN	NE	NDOF	NME	No precond.		Lumped precond.	
				ITER	TIME [s]	ITER	TIME [s]
11042	10000	21874	195279	336	7.850	33	2.700
24102	22500	47896	435756	301	20.500	33	6.900
42127	40000	83846	770981	290	42.160	32	13.340
93196	90000	185784	1726564	383	158.680	33	39.300

3.2. Simple 3D example

A cube domain was chosen for a numerical testing of the lumped preconditioner in three dimensions. The domain was discretized by the hexahedral finite element with three DOFs in each node. The mesh generator T3D was used for discretization. Five different meshes were used (10x10x10 elements, 15x15x15 elements, 20x20x20 elements, 25x25x25 elements and 30x30x30 elements). The software t3d2psifel was used for dividing the finite element mesh into 2-20 parts. Dirichlet boundary conditions were defined at nodes with the z-coordinate equal to zero. Prescribed loads were modelled by nodal forces located on the top of cube domain. Linear elasticity was considered for testing.

The results obtained from these numerical experiments are summarized in graphs on Fig. 11 - Fig. 20 and Tab. 6 - Tab. 10. The number of iterations with respect to the number of nodes is shown on Fig. 11, Fig. 13, Fig. 15, Fig. 17 and Fig. 19. The time of whole solution with respect to the number of nodes is shown on Fig. 12, Fig. 14, Fig. 16, Fig. 18 and Fig. 20. The same notation as in section 3.1 is used here.

The results of numerical testing of the lumped preconditioner in three dimensions are similar to the results in two dimensions. The FETI method with the lumped preconditioner is more efficient than the method without preconditioner (see Fig. 1, Fig. 3, Fig. 5, Fig. 7 and Fig. 9). The preconditioner leads to smaller time requirements (see Fig. 2, Fig. 4, Fig. 6, Fig. 8 and Fig. 10).

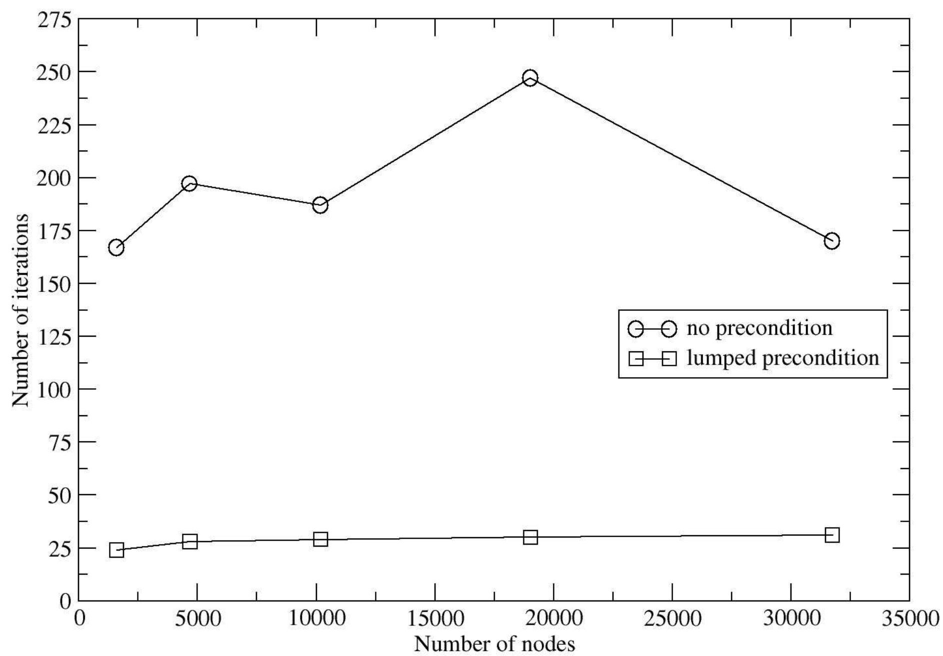

Fig. 11: Number of iterations for 4 subdomains

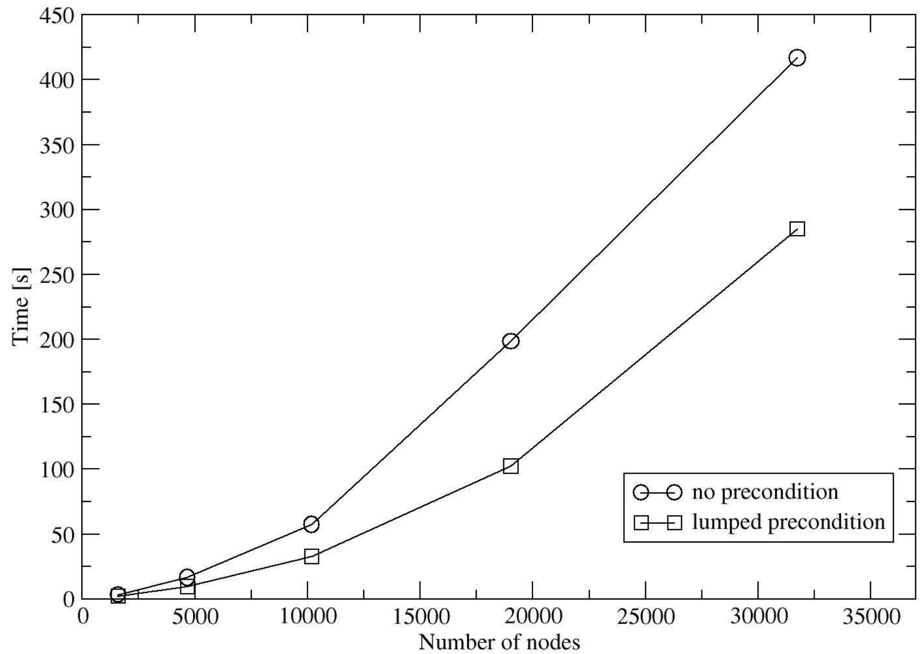

Fig. 12: Time of solution for 4 subdomains

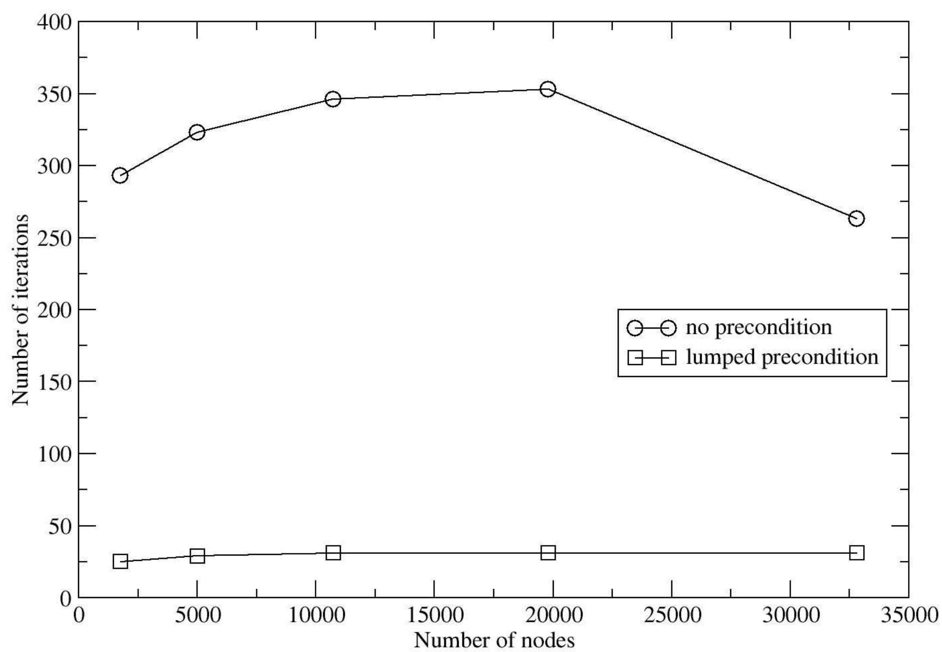

Fig. 13: Number of iterations for 8 subdomains

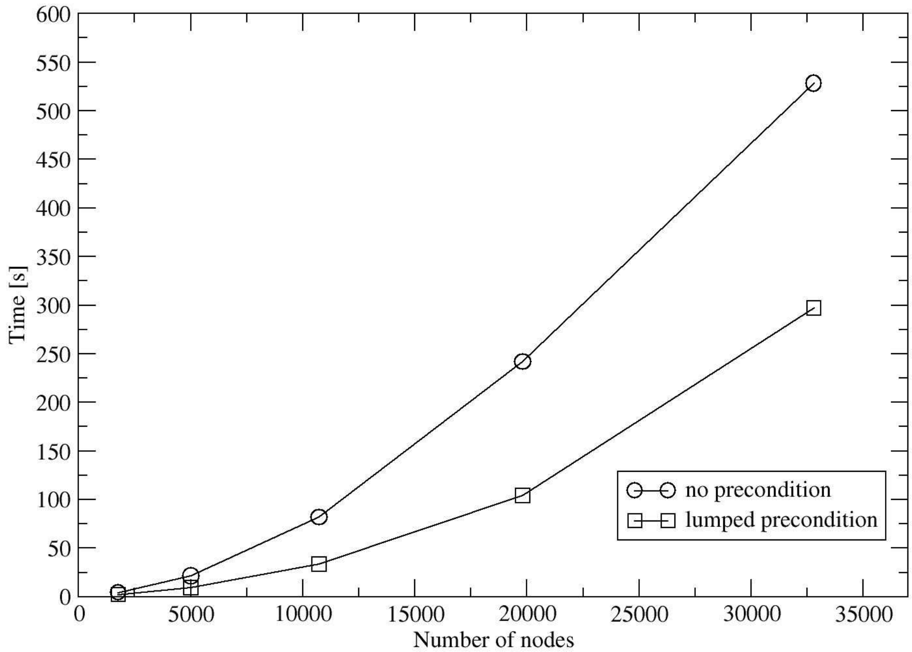

Fig. 14: Time of solution for 8 subdomains

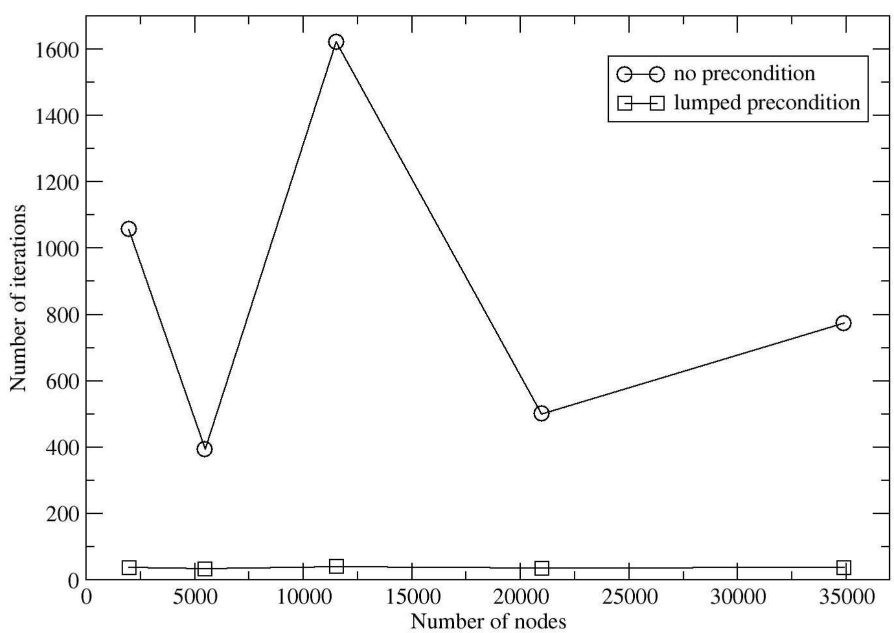

Fig. 15: Number of iterations for 12 subdomains

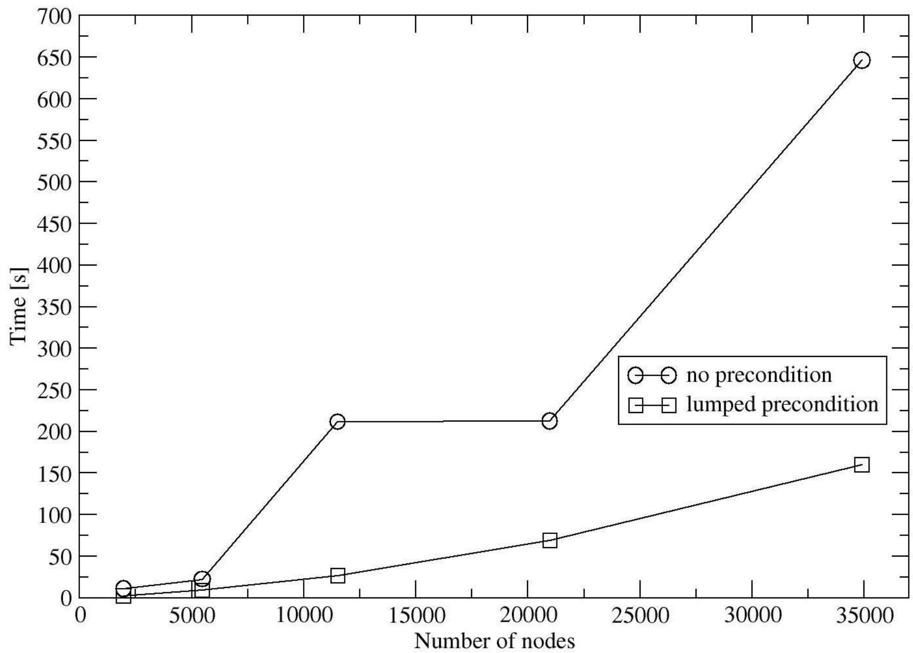

Fig. 16: Time of solution for 12 subdomains

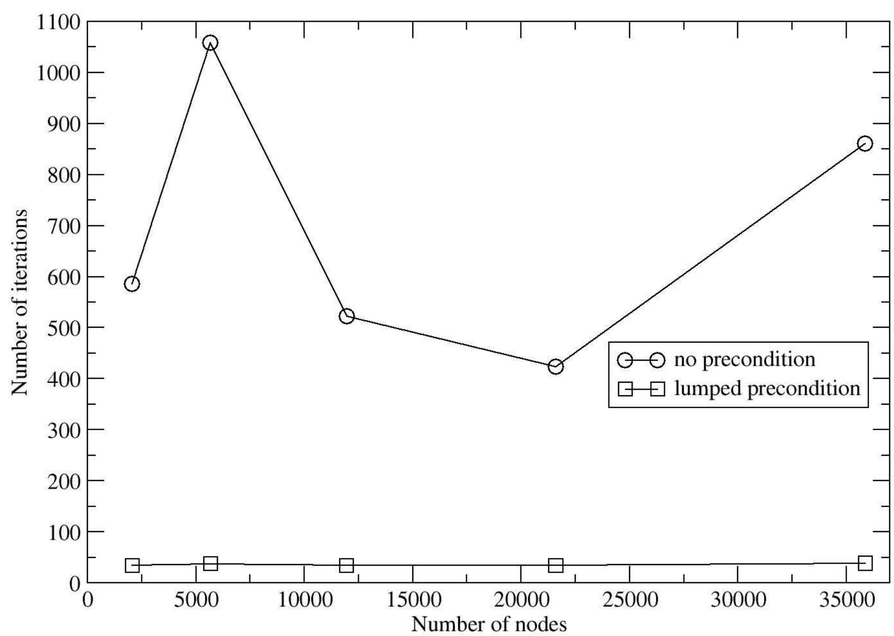

Fig. 17: Number of iterations for 16 subdomains

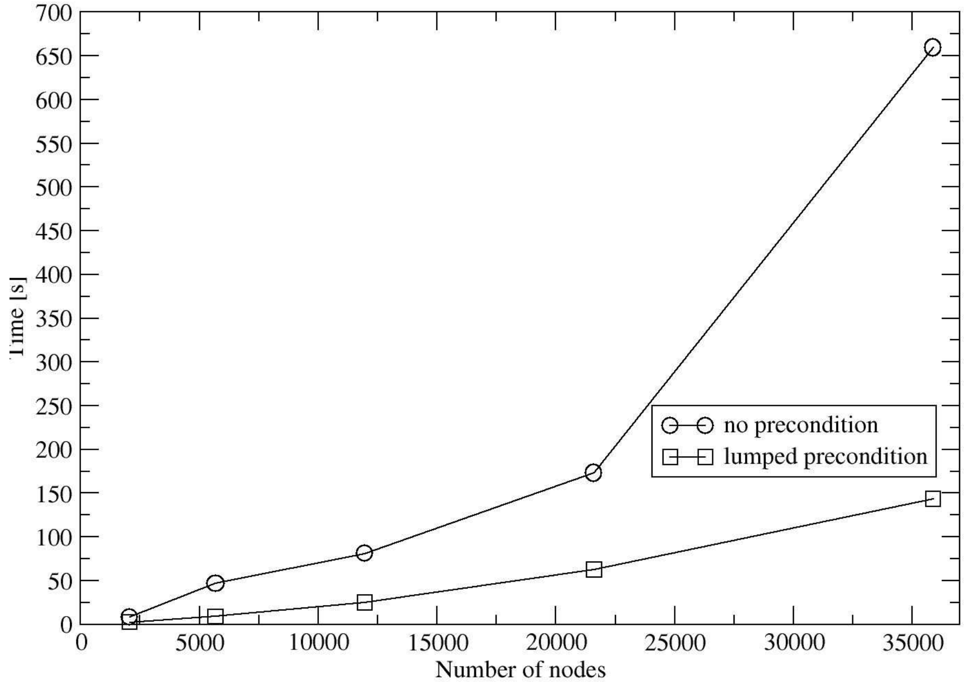

Fig. 18: Time of solution for 16 subdomains

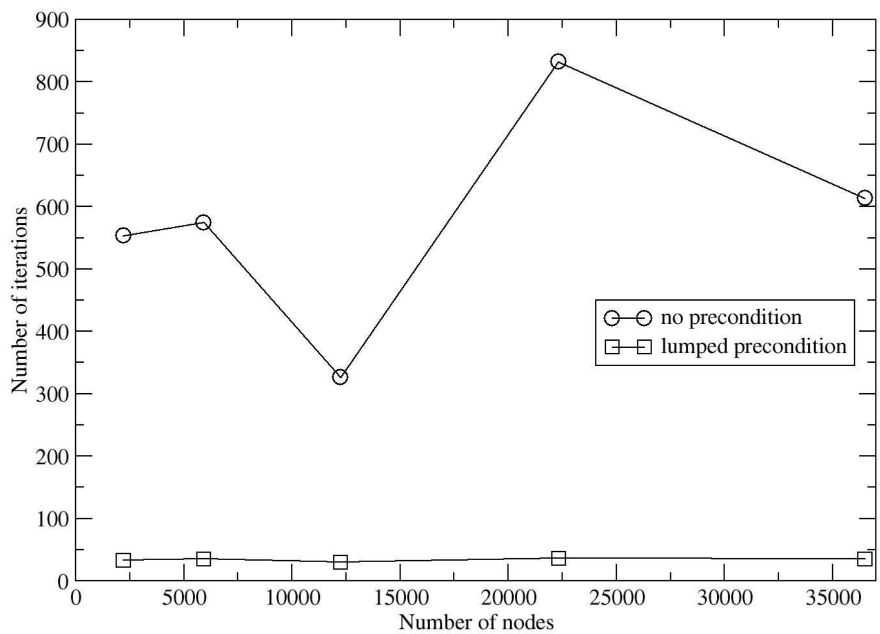

Fig. 19: Number of iterations for 20 subdomains

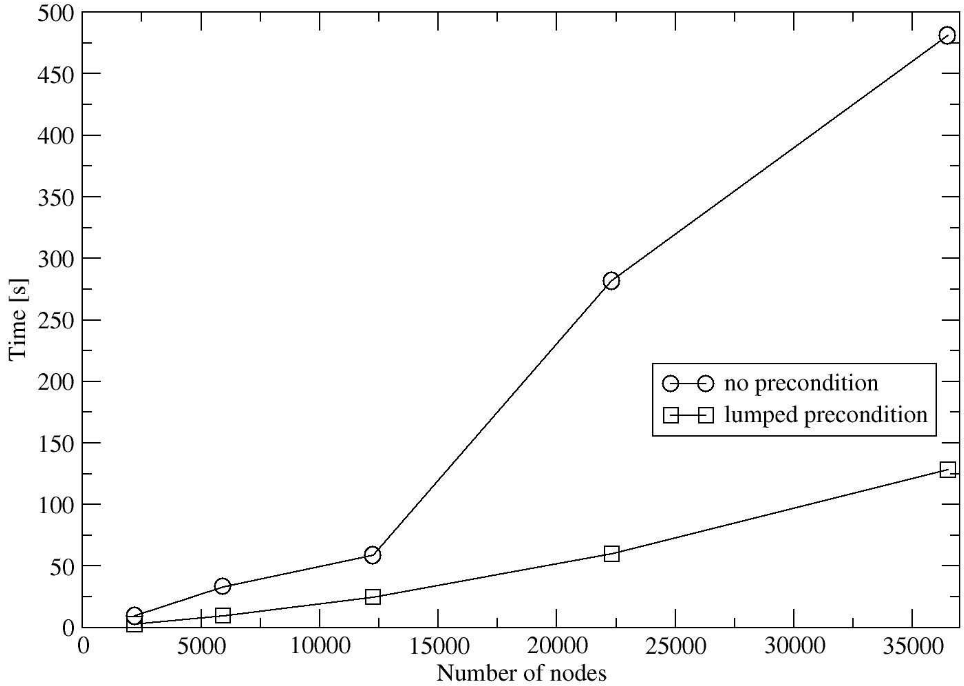

Fig. 20: Time of solution for 20 subdomains

Tab. 6: Decomposition of the problem into 4 subdomains

NN	NE	NDOF	NME	No precond.		Lumped precond.	
				ITER	TIME [s]	ITER	TIME [s]
4678	3375	13218	436710	197	16.590	28	9.470
10194	8000	29196	1019880	187	57.440	29	32.610
19026	15625	54957	1978746	247	198.380	30	102.230
31744	27000	92256	3399114	170	416.750	31	284.930

Tab. 7 : Decomposition of the problem into 8 subdomains

NN	NE	NDOF	NME	No precond.		Lumped precond.	
				ITER	TIME [s]	ITER	TIME [s]
5010	3375	14157	447984	323	21.390	29	9.160
10730	8000	30738	1039056	346	81.670	31	33.420
19810	15625	57225	2007546	353	241.610	31	104.110
32800	27000	95328	3438738	263	528.280	31	297.070

Tab. 8 : Decomposition of the problem into 12 subdomains

NN	NE	NDOF	NME	No precond.		Lumped precond.	
				ITER	TIME [s]	ITER	TIME [s]
5476	3375	15522	464493	393	22.010	33	8.870
11525	8000	33060	1068153	1621	211.240	40	26.630
20985	15625	60711	2052237	500	212.280	35	68.720
34902	27000	101562	3520200	773	646.050	37	159.990

Tab. 9 : Decomposition of the problem into 16 subdomains

NN	NE	NDOF	NME	No precond.		Lumped precond.	
				ITER	TIME [s]	ITER	TIME [s]
5675	3375	16071	470712	1058	46.590	37	8.900
11963	8000	34359	1083981	522	80.860	34	24.920
21603	15625	62520	2074800	423	173.170	34	62.530
35878	27000	104382	3555999	860	659.240	38	143.480

Tab. 10 : Decomposition of the problem into 20 subdomains

NN	NE	NDOF	NME	No precond.		Lumped precond.	
				ITER	TIME [s]	ITER	TIME [s]
5930	3375	16863	480009	574	33.060	35	9.620
12255	8000	35238	1094469	326	58.600	30	24.750
22322	15625	64641	2101056	832	281.790	36	59.6400
36492	27000	106278	3579537	613	481.000	35	28.090

4. Conclusions

The FETI method with the lumped preconditioner reduces the number if iterations as well as it reduces the time requirements in comparison with the FETI without preconditioner. The numerical examples reveal that the number of iterations during solution of the coarse problem grows very slowly with respect to the number of unknowns.

In the future, the Dirichlet preconditioner is planned to be implemented. The Dirichlet preconditioner is mathematically optimal and it should be better than the lumped preconditioner. It means that it leads to the smaller number of iterations in comparison with the lumped preconditioner. On the other hand, time requirements of the Dirichlet preconditioner are greater than the time requirements of the lumped preconditioner because it assembles and computes the Schur complements which is more complicated than matrix-vector multiplication used in the lumped preconditioner.

Acknowledgement

Financial support for this work was provided by project CTU 0805511 of the Czech Technical University in Prague. The financial support is gratefully acknowledged.

References

Farhat, C. & Roux, F. X., (1994) Implicit parallel processing in structural mechanics, Comput, Mech. Adv. pp. 2 1 - 124.

Farhat, C. & Roux, F. X., (1991) A Method of Finite Element Tearing and Interconnecting and its Parallel Solution Algorithm, International Journal for Numerical Methods in Engineering 32, pp. 1205-1227.

SIFEL code web pages http://mech.fsv.cvut.cz/~sifel

T3D web pages http://mech.fsv.cvut.cz/~dr

T3D2PSIFEL web pages http://mech.fsv.cvut.cz/~dr