Treatment of singular matrices in the Hybrid total FETI method

A. Markopoulos, L. Říha, T. Brzobohatý, P. Jirůtková, R. Kučera, O. Meca, and T. Kozubek

1 From FETI to HTFETI method

The FETI (Finite Element Tearing and Interconnecting) method is based on eliminating primal unknowns so that dual linear systems in terms of Lagrange multipliers are solvable by the projected conjugate gradient method (see Farhat and Roux [1994]). The projections on the kernel of $\mathbf{G}^\top$ are computed by the orthogonal projector

$$\mathbf{P} = \mathbf{I} - \mathbf{G}\left(\mathbf{G}^\top\mathbf{G}\right)^{-1}\mathbf{G}^\top. \tag{1}$$

The H(ybrid)FETI method (see Klawonn and Rheinbach [2010]) combines the classical FETI method and the FETI-DP method (see Farhat et al. [2001]) with the aim to adapt a code to parallel computer architectures. In this

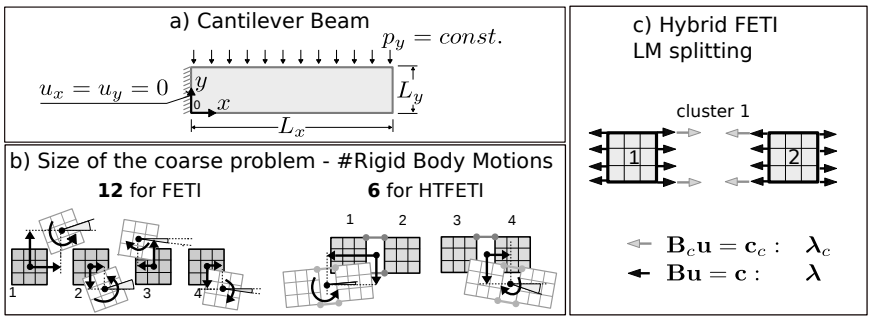

Fig. 1 Cantilever beam in 2D.

IT4Innovations National Supercomputing Centre, 17. listopadu 15/2172, Ostrava, Czech Republic. alexandros.markopoulos@vsb.cz,lubomir.riha@vsb.cz, tomas.brzobohaty@vsb.cz, pavla.jirutkova@vsb.cz, radek.kucera@vsb.cz, ondrej.meca@vsb.cz, tomas.kozubek@vsb.cz

paper, we use another variant of the Hybrid FETI method (see Brzobohatý et al.) that starts from the T(otal)FETI method (see Dostál et al. [2006]). Its implementation (HTFETI) does not differ significantly from the original approach (TFETI). In some sense, having both algorithms in one library requires just a few additions across the code of the TFETI method. Note that TFETI approach also enforces the boundary conditions by Lagrange multipliers so that stiffness matrices on all subdomains exhibit the same defect and kernel matrices may be easily assembled.

We will shortly introduce our HTFETI method for the 2-dimensional problem given by cantilever beam, see Fig.1.a. After discretization, domain decomposition, and linear algebra object assembly, the linear system reads as follows:

$$\left(\begin{array}{cccc|ccc} \mathbf{K}_1 & \mathbf{O} & \mathbf{O} & \mathbf{O} & \mathbf{B}_{c,1}^\top & \mathbf{O} & \mathbf{B}_1^\top \\ \mathbf{O} & \mathbf{K}_2 & \mathbf{O} & \mathbf{O} & \mathbf{B}_{c,2}^\top & \mathbf{O} & \mathbf{B}_2^\top \\ \mathbf{O} & \mathbf{O} & \mathbf{K}_3 & \mathbf{O} & \mathbf{O} & \mathbf{B}_{c,3}^\top & \mathbf{B}_3^\top \\ \mathbf{O} & \mathbf{O} & \mathbf{O} & \mathbf{K}_4 & \mathbf{O} & \mathbf{B}_{c,4}^\top & \mathbf{B}_4^\top \\ \hline \mathbf{B}_{c,1} & \mathbf{B}_{c,2} & \mathbf{O} & \mathbf{O} & \mathbf{O} & \mathbf{O} & \mathbf{O} \\ \mathbf{O} & \mathbf{O} & \mathbf{B}_{c,3} & \mathbf{B}_{c,4} & \mathbf{O} & \mathbf{O} & \mathbf{O} \\ \hline \mathbf{B}_1 & \mathbf{B}_2 & \mathbf{B}_3 & \mathbf{B}_4 & \mathbf{O} & \mathbf{O} & \mathbf{O} \end{array}\right) \left(\begin{array}{c} \mathbf{u}_1 \\ \mathbf{u}_2 \\ \mathbf{u}_3 \\ \mathbf{u}_4 \\ \hline \boldsymbol{\lambda}_{c,1} \\ \boldsymbol{\lambda}_{c,2} \\ \hline \boldsymbol{\lambda} \end{array}\right) = \left(\begin{array}{c} \mathbf{f}_1 \\ \mathbf{f}_2 \\ \mathbf{f}_3 \\ \mathbf{f}_4 \\ \hline \mathbf{o} \\ \mathbf{o} \\ \hline \mathbf{c} \end{array}\right). \tag{2}$$

We denote:

$$\mathbf{B}_c = \begin{pmatrix} \mathbf{B}_{c,1} & \mathbf{B}_{c,2} & \mathbf{O} & \mathbf{O} \\ \mathbf{O} & \mathbf{O} & \mathbf{B}_{c,3} & \mathbf{B}_{c,4} \end{pmatrix}, \quad \mathbf{B} = \begin{pmatrix} \mathbf{B}_1 & \mathbf{B}_2 & \mathbf{B}_3 & \mathbf{B}_4 \end{pmatrix}.$$

The matrix $\mathbf{B}_c$ is a copy of specific rows from the matrix $\mathbf{B}$ corresponding to components of $\boldsymbol{\lambda}$ acting on the corners between subdomains 1,2, and 3,4, respectively (see Fig.1.c). Although the whole matrix in (2) is singular, it beneficially affects convergence of the iterative process (Farhat and Roux [1994]). If the redundant rows of $\mathbf{B}_c$ are omitted, the primal solution components remain the same. To simplify our presentation, we permute (2) as

$$\left(\begin{array}{ccc|ccc|c} \mathbf{K}_1 & \mathbf{O} & \mathbf{B}_{c,1}^\top & \mathbf{O} & \mathbf{O} & \mathbf{O} & \mathbf{B}_1^\top \\ \mathbf{O} & \mathbf{K}_2 & \mathbf{B}_{c,2}^\top & \mathbf{O} & \mathbf{O} & \mathbf{O} & \mathbf{B}_2^\top \\ \mathbf{B}_{c,1} & \mathbf{B}_{c,2} & \mathbf{O} & \mathbf{O} & \mathbf{O} & \mathbf{O} & \mathbf{O} \\ \hline \mathbf{O} & \mathbf{O} & \mathbf{O} & \mathbf{K}_3 & \mathbf{O} & \mathbf{B}_{c,3}^\top & \mathbf{B}_3^\top \\ \mathbf{O} & \mathbf{O} & \mathbf{O} & \mathbf{O} & \mathbf{K}_4 & \mathbf{B}_{c,4}^\top & \mathbf{B}_4^\top \\ \mathbf{O} & \mathbf{O} & \mathbf{O} & \mathbf{B}_{c,3} & \mathbf{B}_{c,4} & \mathbf{O} & \mathbf{O} \\ \hline \mathbf{B}_1 & \mathbf{B}_2 & \mathbf{O} & \mathbf{B}_3 & \mathbf{B}_4 & \mathbf{O} & \mathbf{O} \end{array}\right) \left(\begin{array}{c} \mathbf{u}_1 \\ \mathbf{u}_2 \\ \boldsymbol{\lambda}_{c,1} \\ \hline \mathbf{u}_3 \\ \mathbf{u}_4 \\ \boldsymbol{\lambda}_{c,2} \\ \hline \boldsymbol{\lambda} \end{array}\right) = \left(\begin{array}{c} \mathbf{f}_1 \\ \mathbf{f}_2 \\ \mathbf{o} \\ \hline \mathbf{f}_3 \\ \mathbf{f}_4 \\ \mathbf{o} \\ \hline \mathbf{c} \end{array}\right), \tag{3}$$

and then we introduce a new notation consistently with the line partition in (3):

$$\left(\begin{array}{c|c|c} \tilde{\mathbf{K}}_1 & \mathbf{O} & \tilde{\mathbf{B}}_1^\top \\ \hline \mathbf{O} & \tilde{\mathbf{K}}_2 & \tilde{\mathbf{B}}_2^\top \\ \hline \tilde{\mathbf{B}}_1 & \tilde{\mathbf{B}}_2 & \mathbf{O} \end{array}\right) \left(\begin{array}{c} \tilde{\mathbf{u}}_1 \\ \hline \tilde{\mathbf{u}}_2 \\ \hline \tilde{\boldsymbol{\lambda}} \end{array}\right) = \left(\begin{array}{c} \tilde{\mathbf{f}}_1 \\ \hline \tilde{\mathbf{f}}_2 \\ \hline \tilde{\mathbf{c}} \end{array}\right). \tag{4}$$

Eliminating $\tilde{\mathbf{u}}_i$, $i = 1, 2$, we also eliminate the subset of dual variables $\boldsymbol{\lambda}_{c,j}$, $j = 1, 2$ related to the matrix $\mathbf{B}_c$. Therefore, the structure behaves like a problem decomposed into two clusters: the 1st and 2nd subdomains belong to the first cluster, the 3rd and 4th subdomains belong to the second cluster, see Fig.1.b. Here, $\tilde{\mathbf{K}}_1$, $\tilde{\mathbf{K}}_2$ can be interpreted as the cluster stiffness matrices with the kernels $\tilde{\mathbf{R}}_1$, $\tilde{\mathbf{R}}_2$, respectively. Denoting $\tilde{\mathbf{K}}$=diag$(\tilde{\mathbf{K}}_1, \tilde{\mathbf{K}}_2)$, $\tilde{\mathbf{B}} = (\tilde{\mathbf{B}}_1, \tilde{\mathbf{B}}_2)$, $\tilde{\mathbf{R}}^\top = (\tilde{\mathbf{R}}_1^\top, \tilde{\mathbf{R}}_2^\top)$, $\tilde{\mathbf{F}} = \tilde{\mathbf{B}}\tilde{\mathbf{K}}^+\tilde{\mathbf{B}}^\top$, $\tilde{\mathbf{G}} = -\tilde{\mathbf{B}}\tilde{\mathbf{R}}$, $\tilde{\mathbf{d}} = \tilde{\mathbf{B}}\tilde{\mathbf{K}}^+\tilde{\mathbf{f}} - \tilde{\mathbf{c}}$, and $\tilde{\mathbf{e}} = -\tilde{\mathbf{R}}^\top\tilde{\mathbf{f}}^\top$, we arrive at the Schur complement system

$$\begin{pmatrix} \tilde{\mathbf{F}} & \tilde{\mathbf{G}} \\ \tilde{\mathbf{G}}^\top & \mathbf{O} \end{pmatrix} \begin{pmatrix} \tilde{\boldsymbol{\lambda}} \\ \tilde{\boldsymbol{\alpha}} \end{pmatrix} = \begin{pmatrix} \tilde{\mathbf{d}} \\ \tilde{\mathbf{e}} \end{pmatrix} \tag{5}$$

that can be solved by the same iterative method as in the classical FETI method. The dimension of the new coarse problem $\tilde{\mathbf{G}}^\top\tilde{\mathbf{G}}$ is smaller (size = 6) compared to the FETI case. To keep optimality of the HTFETI approach, the matrix $\tilde{\mathbf{K}}$ can not be factorized directly. The implicit factorization will be demonstrated by its first block (cluster). It is obtained by solving the linear system $\tilde{\mathbf{K}}_1\tilde{\mathbf{x}}_1 = \tilde{\mathbf{b}}_1$, i.e.,

$$\begin{pmatrix} \mathbf{K}_{1:2} & \mathbf{B}_{c,1:2}^\top \\ \mathbf{B}_{c,1:2} & \mathbf{O} \end{pmatrix} \begin{pmatrix} \mathbf{x}_1 \\ \boldsymbol{\mu} \end{pmatrix} = \begin{pmatrix} \mathbf{b} \\ \mathbf{z} \end{pmatrix}, \tag{6}$$

where $\mathbf{K}_{1:2} = \text{diag}(\mathbf{K}_1, \mathbf{K}_2)$ and $\mathbf{B}_{c,1:2} = (\mathbf{B}_{c,1}, \mathbf{B}_{c,2})$. The subindex $1:2$ adverts to the first and the last ordinal number of the subdomains in the cluster. Although (6) can be interpreted as a FETI problem, we solve it by a direct solver. The respective Schur complement system reads as:

$$\begin{pmatrix} \mathbf{F}_{c,1:2} & \mathbf{G}_{c,1:2} \\ \mathbf{G}_{c,1:2}^\top & \mathbf{O} \end{pmatrix} \begin{pmatrix} \boldsymbol{\mu} \\ \boldsymbol{\beta} \end{pmatrix} = \begin{pmatrix} \mathbf{d}_{c,1:2} \\ \mathbf{e}_{c,1:2} \end{pmatrix}, \tag{7}$$

where $\mathbf{F}_{c,1:2} = \mathbf{B}_{c,1:2}\mathbf{K}_{1:2}^+\mathbf{B}_{c,1:2}^\top$, $\mathbf{G}_{c,1:2} = -\mathbf{B}_{c,1:2}\mathbf{R}_{1:2}$, $\mathbf{d}_{c,1:2} = \mathbf{B}_{c,1:2}\mathbf{K}_{1:2}^+\mathbf{b} - \mathbf{z}$, $\mathbf{e}_{c,1:2} = -\mathbf{R}_{1:2}^\top\mathbf{b}$, and $\mathbf{R}_{1:2} = \text{diag}\,(\mathbf{R}_1,\ \mathbf{R}_2)$. To obtain the vector $\tilde{\mathbf{x}}_1$, both systems (6), (7) are subsequently solved in three steps:

$$\begin{aligned} \boldsymbol{\beta} &= \mathbf{S}_{c,1:2}^+ \left(\mathbf{G}_{c,1:2}^\top\mathbf{F}_{c,1:2}^{-1}\mathbf{d}_{c,1:2} - \mathbf{e}_{c,1:2}\right), \\ \boldsymbol{\mu} &= \mathbf{F}_{c,1:2}^{-1}\left(\mathbf{d}_{c,1:2} - \mathbf{G}_{c,1:2}\boldsymbol{\beta}\right), \\ \mathbf{x} &= \mathbf{K}_{1:2}^+\left(\mathbf{b} - \mathbf{B}_{c,1:2}^\top\boldsymbol{\mu}\right) + \mathbf{R}_{1:2}\boldsymbol{\beta}, \end{aligned} \tag{8}$$

where $\mathbf{S}_{c,1:2} = \mathbf{G}_{c,1:2}^\top\mathbf{F}_{c,1:2}^{-1}\mathbf{G}_{c,1:2}$ is the singular Shur complement matrix.

The kernel $\tilde{\mathbf{R}}_1$ of $\tilde{\mathbf{K}}_1$ is the last object going to be effectively evaluated. The orthogonality condition $\tilde{\mathbf{K}}_1\tilde{\mathbf{R}}_1 = \mathbf{O}$ can be written by

$$\left(\begin{array}{c|c} \mathbf{K}_{1:2} & \mathbf{B}_{c,1:2}^{\top} \\ \hline \mathbf{B}_{c,1:2} & \mathbf{O} \end{array}\right) \left(\begin{array}{c} \mathbf{R}_{1:2} \\ \hline \mathbf{O} \end{array}\right) \mathbf{H}_{1:2} = \left(\begin{array}{c} \mathbf{O} \\ \hline \mathbf{O} \end{array}\right), \tag{9}$$

where $\tilde{\mathbf{R}}_1 = (\mathbf{R}_{1:2}^{\top},\ \mathbf{O}^{\top})^{\top}\mathbf{H}_{1:2}$. Assuming that the subdomain kernels $\mathbf{R}_1$ and $\mathbf{R}_2$ are known, it remains to determine $\mathbf{H}_{1:2}$. The first equation in (9) does not impose any condition onto $\mathbf{H}_{1:2}$. The second equation gives

$$\mathbf{B}_{c,1:2}\mathbf{R}_{1:2}\mathbf{H}_{1:2} = -\mathbf{G}_{c,1:2}\mathbf{H}_{1:2} = \mathbf{O}, \tag{10}$$

implying that $\mathbf{H}_{1:2}$ is the kernel of $\mathbf{G}_{c,1:2}$, which is not full-column rank matrix due to the absence of the Dirichlet boundary condition in $\mathbf{B}_{c,1:2}$.

Preprocessing in the HTFETI method starts in the same way as in the FETI approach preparing factors $\mathbf{K}_i$ and kernels $\mathbf{R}_i$ for each subdomain. Then, only one pair consisting of $\mathbf{F}_{c,j:k}$ and $\mathbf{S}_{c,j:k}$ is assembled and factorized on each cluster. The dimension of $\mathbf{F}_{c,1:2}$ is controlled by the number of Lagrange multipliers $\boldsymbol{\lambda}_{c,1}$ glueing the cluster subdomains. The dimension of $\mathbf{S}_{c,1:2}$ is given by the sum of defects of all matrices $\mathbf{K}_i$ belonging to a particular cluster.

2 Solving a singular system via kernel detection

This work continues with the results of Dostál et al. [2011], Brzobohatý et al. [2011], Kučera et al. [2012], Kučera et al. [2013], and it queries from work published by Suzuki and Roux [2014].

If a problem with large jumps in the material coefficients and/or with an irregular decomposition is solved by the FETI method, direct factorizations of singular symmetric stiffness matrices $\mathbf{K}_i$ can be very unstable due to unclear criteria for distinguishing null pivots. We propose a heuristic technique for detecting kernels $\mathbf{R}_i$ of symmetric positive semi-definite (SPSD) matrices utilizing direct solvers designed primarily for non-singular cases. The mesh of the subdomain, the stiffness matrix of which is assembled above, must be given by the specific graph decomposition. In the three-dimensional case, e.g., deleting any two nodes of the relevant graph does not yield two components (the resulting graph will remain connected). The analyzed matrix should be also diagonally scaled. Via fixing nodes (FNs) the goal is to find (see Dostál et al. [2011]) an appropriate set of indices s ($\text{size}(s) \geq \text{defect}(\mathbf{K}_i)$) and a complementary set of indices r characterizing the singular and non-singular part of $\mathbf{K}_i$, respectively. The original stiffness matrix $\mathbf{K}_i$ (the subindex will be omitted in the rest of this section) can be permuted by the matrix $\mathbf{Q}$ so that

$$\mathbf{QKQ}^T = \begin{pmatrix} \mathbf{K}_{rr}\ \mathbf{K}_{rs} \\ \mathbf{K}_{sr}\ \mathbf{K}_{ss} \end{pmatrix},$$

where $\mathbf{K}_{rr}$ is the well-conditioned matrix. It is sufficient to find at least 3 noncollinear nodes from the finite element mesh in the case of 3-dimensional linear elasticity. The DOFs corresponding to these nodes determine the set s. Our choice of the FNs is based on a random number generator. From mechanical point of view, the structure is sufficiently supported by those FNs against any rigid movement. As the Schur complement $\mathbf{S} = \mathbf{K}_{ss} - \mathbf{K}_{sr}\mathbf{K}_{rr}^{-1}\mathbf{K}_{rs}$ is a relatively small matrix, it can be analysed by robust algorithms for dense matrices.

Once the Schur complement is correctly defined, it is spectrally decomposed using, e.g., LAPACK to $\mathbf{U\Sigma U}^\top$. Its eigenvalues are stored in $\mathbf{\Sigma} = \mathrm{diag}(\sigma_1, \sigma_2, \cdots, \sigma_n)$ in the descending order. The k-th eigenvalue is considered to be zero, if

$$\sigma_k / \sigma_{k-1} < 10^{-4}.$$

Such information determines splitting $\mathbf{U} = (\hat{\mathbf{U}},\ \mathbf{R}_s)$ where $\mathbf{R}_s$ consists of last columns of $\mathbf{U}$ starting with the column index k, and it is already a part of the searched kernel of $\mathbf{K}$. If $\mathbf{R}_s$ is known, its supplement $\mathbf{R}_r = -\mathbf{K}_{rr}^{-1}\mathbf{K}_{rs}\mathbf{R}_s$ is obtained from

$$\begin{pmatrix} \mathbf{K}_{rr}\ \mathbf{K}_{rs} \\ \mathbf{K}_{sr}\ \mathbf{K}_{ss} \end{pmatrix} \begin{pmatrix} \mathbf{R}_r \\ \mathbf{R}_s \end{pmatrix} = \begin{pmatrix} \mathbf{O} \\ \mathbf{O} \end{pmatrix}. \tag{11}$$

As an example, a uniformly meshed cube ($L = 30$ mm, $E = 2.1 \cdot 10^5$ MPa, $\mu = 0.3$, $\rho = 7850$ kg/m^3, $g = 9.81$ m/s^2) is used with a variable number of nodes controlled by n (number of nodes in x, y, and z direction). The singular set s is selected via several DOFs belonging to randomly chosen FNs. The quality of a selection (see Fig. 2) is measured by the ratio of bad choices (collinear nodes) to all possible combinations for a given number of FNs and the size of mesh n. Probability curves for 3, 4, and 5 FNs depending

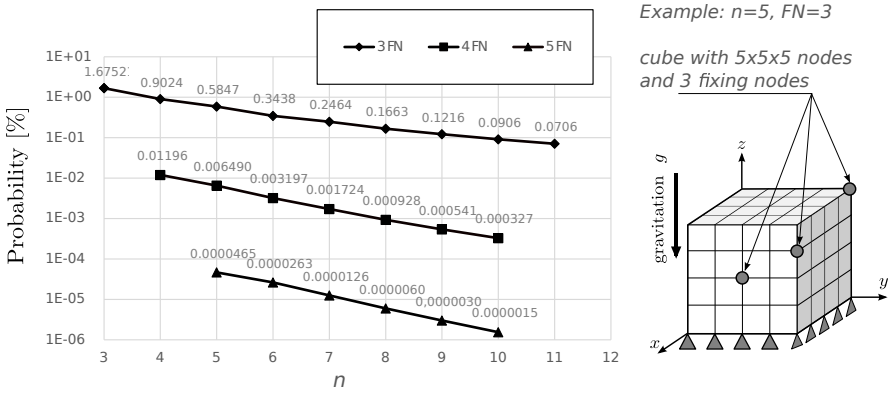

Fig. 2 Probability of collinear fixing nodes (FN).

on the mesh parameter n are shown in Fig. 2. Increasing FNs for fixed mesh (constant n) intuitively helps to ensure noncollinear nodes. For instance, for $n = 10$ with 3 FNs the probability of a bad choice is $9.068 \cdot 10^{-2}$, with 4 FNs it decreases to $3.272 \cdot 10^{-4}$, and with 5 FNs to $1.545 \cdot 10^{-6}$. Surprisingly enough, for a fixed number of FNs and a simultaneously enlarging parameter n (mesh refinement), the probability of collinear FNs decreases as well.

3 ExaScale PaRallel FETI SOlver - ESPRESO

ESPRESO is a highly efficient parallel solver which contains several FETI method based algorithms including the HTFETI method suitable for parallel machines with tens or hundreds of thousands of cores. The solver is based on a highly efficient communication layer based on MPI, and it is able to run on massively parallel machines with thousands of compute nodes and hundreds of thousands of CPU cores. ESPRESO is also being developed to support modern many-core accelerators. We are currently developing four major versions of the solver:

- **ESPRESO CPU** is a CPU version using sparse representation of system matrices;
- **ESPRESO MIC** is an Intel Xeon Phi accelerated version working with dense representation of system matrices in the form of Schur complement;
- **ESPRESO GPU** is a GPU accelerated version working with dense structures. Support for sparse structures using cuSolver is under development;
- **ESPRESO GREEN** is a power efficient version developed under the H2020 READEX project. This version is in the very early development stage.

In order to solve real engineering problems, we are developing a FEM/BEM library that enables database files from ANSYS simulation software to be imported and all inputs required by the FETI or HTFETI solver generated. In addition, we are developing an interface to ELMER that allows ESPRESO to be used as its linear solver. This integration is done through API that can be used as an interface to many other applications.

4 Numerical Experiments

Efficiency of the HTFETI method is presented in the ESPRESO library on the cube benchmark described in Sec. 2. Weak scalability of the solver, see Fig.3 left, includes matrix assembly, linear solver preprocessing (preprocessing of the TFETI and HTFETI method), and iterative solver runtime mea-

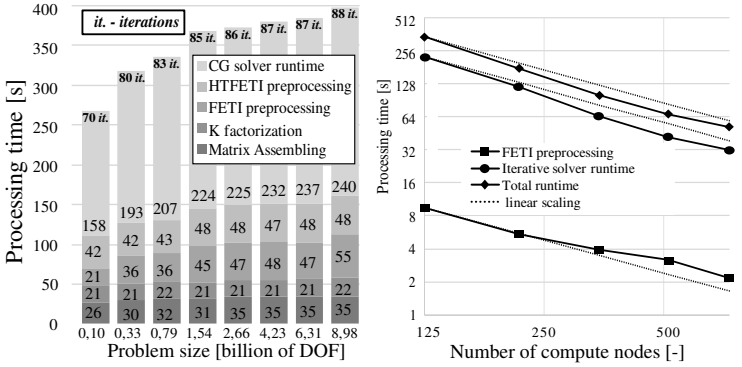

Fig. 3 Weak and strong scalability.

sured on 1 to 729 compute nodes of IT4Innovations Salomon supercomputer. Benchmark configuration: subdomain size 14,739 DOFs ($n = 17$); 1,000 subdomains per cluster; Lumped preconditioner, stopping criteria 10^{-3}. Strong scalability on 126, 216, 343, 512, and 729 compute nodes of Salomon supercomputer is seen in Fig. 3 right. The problem size is 1.5 billions of unknowns.

5 Conclusions

This paper presents the HTFETI method, an extension of FETI algorithm for problems with the larger number of subdomains to handle the coarse problem more effectively. The basic principles are explained and demonstrated on linear elasticity problem. In the second part, the methodology for factorizing SPSD matrix using robust applications, e.g., PARDISO, is shown. Efficiency is proved by the numerical test performed in ESPRESO library for almost 9 billions of unknowns.

6 Acknowledgment

This work was supported by The Ministry of Education, Youth and Sports from the National Programme of Sustainability (NPU II) project IT4Innovations excellence in science - LQ1602 and from the Large Infrastructures for Research, Experimental Development and Innovations project IT4Innovations National Supercomputing Center LM2015070.

References

T. Brzobohatý, M. Jarošová, T. Kozubek, M. Menšík, and A. Markopoulos. The hybrid total FETI method. In *Proceedings of the Third International Conference on Parallel, Distributed, Grid and Cloud Computing for Engineering*. Civil-Comp, Ltd.

T. Brzobohatý, Z. Dostál, T. Kozubek, P. Kovář, and A. Markopoulos. Cholesky decomposition with fixing nodes to stable computation of a generalized inverse of the stiffness matrix of a floating structure. *Internat. J. Numer. Methods Engrg.*, 88(5):493–509, 2011. ISSN 0029-5981. doi: 10.1002/nme.3187. URL http://dx.doi.org/10.1002/nme.3187.

Zdeněk Dostál, David Horák, and Radek Kučera. Total FETI—an easier implementable variant of the FETI method for numerical solution of elliptic PDE. *Comm. Numer. Methods Engrg.*, 22(12):1155–1162, 2006. ISSN 1069-8299. doi: 10.1002/cnm.881. URL http://dx.doi.org/10.1002/cnm.881.

Zdeněk Dostál, Tomáš Kozubek, Alexandros Markopoulos, and Martin Menšík. Cholesky decomposition of a positive semidefinite matrix with known kernel. *Appl. Math. Comput.*, 217(13):6067–6077, 2011. ISSN 0096-3003. doi: 10.1016/j.amc.2010.12.069. URL http://dx.doi.org/10.1016/j.amc.2010.12.069.

Charbel Farhat and François-Xavier Roux. Implicit parallel processing in structural mechanics. In J. Tinsley Oden, editor, *Computational Mechanics Advances*, volume 2 (1), pages 1–124. North-Holland, 1994.

Charbel Farhat, Michel Lesoinne, Patrick LeTallec, Kendall Pierson, and Daniel Rixen. FETI-DP: a dual-primal unified FETI method. I. A faster alternative to the two-level FETI method. *Internat. J. Numer. Methods Engrg.*, 50(7):1523–1544, 2001. ISSN 0029-5981. doi: 10.1002/nme.76. URL http://dx.doi.org/10.1002/nme.76.

Axel Klawonn and Oliver Rheinbach. Highly scalable parallel domain decomposition methods with an application to biomechanics. *ZAMM Z. Angew. Math. Mech.*, 90(1):5–32, 2010. ISSN 0044-2267. doi: 10.1002/zamm.200900329. URL http://dx.doi.org/10.1002/zamm.200900329.

R. Kučera, T. Kozubek, A. Markopoulos, and J. Machalová. On the Moore-Penrose inverse in solving saddle-point systems with singular diagonal blocks. *Numer. Linear Algebra Appl.*, 19(4):677–699, 2012. ISSN 1070-5325. doi: 10.1002/nla.798. URL http://dx.doi.org/10.1002/nla.798.

R. Kučera, T. Kozubek, and A. Markopoulos. On large-scale generalized inverses in solving two-by-two block linear systems. *Linear Algebra Appl.*, 438(7):3011–3029, 2013. ISSN 0024-3795. doi: 10.1016/j.laa.2012.09.027. URL http://dx.doi.org/10.1016/j.laa.2012.09.027.

A. Suzuki and F.-X. Roux. A dissection solver with kernel detection for symmetric finite element matrices on shared memory computers. *Internat. J. Numer. Methods Engrg.*, 100(2):136–164, 2014. ISSN 0029-5981. doi: 10.1002/nme.4729. URL http://dx.doi.org/10.1002/nme.4729.