Research Article

On Moore-Penrose Pseudoinverse Computation for Stiffness Matrices Resulting from Higher Order Approximation

Marek Klimczak and **Witold Cecot**

Cracow University of Technology, Warszawska 24, 31-155 Cracow, Poland

Correspondence should be addressed to Marek Klimczak; mklimczak@l5.pk.edu.pl

Received 5 November 2018; Accepted 22 January 2019; Published 24 February 2019

Academic Editor: David González

Copyright © 2019 Marek Klimczak and Witold Cecot. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Computing the pseudoinverse of a matrix is an essential component of many computational methods. It arises in statistics, graphics, robotics, numerical modeling, and many more areas. Therefore, it is desirable to select reliable algorithms that can perform this operation efficiently and robustly. A demanding benchmark test for the pseudoinverse computation was introduced. The stiffness matrices for higher order approximation turned out to be such tough problems and therefore can serve as good benchmarks for algorithms of the pseudoinverse computation. It was found out that only one algorithm, out of five known from literature, enabled us to obtain acceptable results for the pseudoinverse of the proposed benchmark test.

1. Introduction

The Moore-Penrose pseudoinverse arises in many real life applications. It is commonly used in graphics (image recognition [1], image restoration [2], data compression [3]), robotics [4], and statistics [5] as well as diverse branches of engineering [6, 7]. Our examination of various algorithms for the pseudoinverse was motivated by the necessity of using such an operation in the local numerical homogenization in solid mechanics. This method, proposed by Jhurani and Demkowicz (see [6–8] and also [9]), is one of the methods of the numerical upscaling, i.e., such a coarse mesh discretization that incorporates a fine mesh information. Other discretization-based homogenization methods are presented, for example, in [10–12]. One of the key steps of the local numerical homogenization is a computation of the pseudoinverse matrix. We observed that whenever the order of approximation at any level is greater than one, the behavior of the algorithms for the pseudoinverse was very poor. We have tested five approaches to computation of the pseudoinverse and concluded that the Demmel-Kahan algorithm [13] was the best one. It is worth mentioning that all the methods performed very well for many other benchmark problems [7, 8, 14–17]. Therefore, a stiffness matrix for elasticity problem

is a really demanding test for the pseudoinverse algorithm and can be used to verify the competitiveness of the newly developed methods.

In the next section the local numerical homogenization for solid mechanics is recalled. Then, several methods for the Moore-Penrose pseudoinverse computation are briefly described. Finally, numerical results are presented and discussed.

2. Local Numerical Homogenization

In this paper, we deal with the finite element (FE) analysis of heterogeneous materials. Let us assume that due to limited computational resources fine mesh solutions to problems with rapidly varying material parameters are infeasible. Instead of that, we generate a coarse mesh, which is, in fact, the finest mesh we can computationally afford (Figure 1). At this level, homogeneity of the domain is assumed and the mesh is adapted at the macroscale using automatic mesh refinement. We use for this purpose the *3Dhp* code [18]. Subsequently, the mesh is refined independently for every coarse element in order to comply with the material heterogeneity. The core of the local numerical homogenization is the computation of effective stiffness matrices for

FIGURE 1: Scheme of the local numerical homogenization routine.

the macroelements. After that, an equivalent coarse mesh problem can be solved. At the postprocessing level also a fine mesh solution for each of coarse elements can be obtained.

A mathematical formulation of the effective coarse mesh stiffness matrix computation is recapitulated below following [6, 8]. All of the quantities referring to the macroscale are marked with *hat* symbol (e.g., $\hat{\mathbf{x}}$).

Given symmetric fine mesh element stiffness matrices assembled into $\mathbf{K} \in \mathbb{R}^{N \times N}$, a nonzero fine mesh load vector $\mathbf{f} \in \mathbb{R}^N$, an interpolation matrix $\mathbf{A} \in \mathbb{R}^{N \times M}$, a positive-definite symmetric weight matrix $\mathbf{B} \in \mathbb{R}^{N \times N}$, and a dimensionless small parameter $\epsilon > 0$ find a symmetric matrix $\hat{\mathbf{K}}^\dagger \in \mathbb{R}^{M \times M}$ for which E reaches minimum, where

$$E(\hat{\mathbf{K}}^\dagger) = \frac{1}{2} \left\| (\mathbf{K}^\dagger - \mathbf{A} \hat{\mathbf{K}}^\dagger \mathbf{A}^T) \mathbf{f} \right\|_{\mathbf{B}}^2 + \frac{\epsilon}{2} \left\| \mathbf{K}^\dagger - \mathbf{A} \hat{\mathbf{K}}^\dagger \mathbf{A}^T \right\|_{\mathbf{F}, \mathbf{B}}^2 \left\| \mathbf{f} \right\|_2^2 \quad (1)$$

and $\|\mathbf{x}\|_2 = \sqrt{\text{trace}(\mathbf{x}^T \mathbf{x})}$ denotes the Euclidean norm, $\|\mathbf{x}\|_{\mathbf{B}} = \sqrt{\text{trace}(\mathbf{x}^T \mathbf{B} \mathbf{x})}$ denotes the Euclidean norm weighted with $\mathbf{B}$, and $\|\mathbf{X}\|_{\mathbf{F}, \mathbf{B}} = \sqrt{\text{trace}(\mathbf{X}^T \mathbf{B} \mathbf{X})}$ denotes the Frobenius norm weighted with $\mathbf{B}$. $\mathbf{x}$ and $\mathbf{X}$ are arbitrary column vector and matrix, respectively. The numbers of degrees of freedom for the fine and coarse meshes are denoted with N and M .

The expression for E consists of two terms. The first of them measures the error introduced to the local solution vector by the numerical homogenization, whereas the second term is a regularization one and guaranties the solution uniqueness. The matrix $\hat{\mathbf{K}}^\dagger$ that minimizes (1) is the pseudoinverse of the local effective stiffness matrix $\hat{\mathbf{K}}$. Derivation of (1) is presented in [6, 8]. Imposing the symmetry conditions on $\hat{\mathbf{K}}^\dagger$ ($\hat{\mathbf{K}}^\dagger - (\hat{\mathbf{K}}^\dagger)^T = \mathbf{0}$) into the Lagrangian, it was proved that problem (1) is equivalent to the continuous-time Lyapunov equation in a following form:

$$\mathbf{C}\mathbf{X} + \mathbf{X}\mathbf{C}^T = \mathbf{D} \quad (2)$$

with auxiliary matrices

$$\begin{aligned} \mathbf{C} &:= \mathbf{V}^{-1} \mathbf{U}, \\ \mathbf{D} &:= \mathbf{V}^{-1} (\mathbf{W} + \mathbf{W}^T) \mathbf{V}^{-1}, \\ \mathbf{U} &:= \mathbf{A}^T \mathbf{B} \mathbf{A}, \\ \mathbf{V} &:= \mathbf{A}^T (\mathbf{f} \mathbf{f}^T + \epsilon \|\mathbf{f}\|_2^2 \mathbf{I}) \mathbf{A}, \\ \mathbf{W} &:= \mathbf{A}^T \mathbf{B} \mathbf{K}^\dagger (\mathbf{f} \mathbf{f}^T + \epsilon \|\mathbf{f}\|_2^2 \mathbf{I}) \mathbf{A}. \end{aligned}$$

In (1) $\mathbf{X}^\dagger$ denotes the Moore-Penrose pseudoinverse of matrix $\mathbf{X}$ [20, 21]. In the case of the real matrices we call a matrix $\mathbf{X}^\dagger \in \mathbb{R}^{N \times M}$ a pseudoinverse of matrix $\mathbf{X} \in \mathbb{R}^{M \times N}$, if it fulfills the following conditions:

$$\mathbf{X} \mathbf{X}^\dagger \mathbf{X} = \mathbf{X} \quad (3)$$

$$\mathbf{X}^\dagger \mathbf{X} \mathbf{X}^\dagger = \mathbf{X}^\dagger \quad (4)$$

$$(\mathbf{X} \mathbf{X}^\dagger)^T = \mathbf{X} \mathbf{X}^\dagger \quad (5)$$

$$(\mathbf{X}^\dagger \mathbf{X})^T = \mathbf{X}^\dagger \mathbf{X} \quad (6)$$

A reliable computation of $\mathbf{X}^\dagger$ is the key point of the local numerical homogenization and it needs to be computed twice (for each coarse element):

- (i) for the assembled stiffness matrix $\mathbf{K} \in \mathbb{R}^{N \times N}$;
- (ii) for the pseudoinverse of the effective stiffness matrix ($\hat{\mathbf{K}}^\dagger \in \mathbb{R}^{M \times M}$, where $M \ll N$).

A comprehensive review of the theory and computational aspects of the pseudoinverse computation can be found in [14], in its updated version [19], and in the most recent book [22]. Also Jhurani [7, 8] presented several methods dedicated to the local numerical homogenization.

In the following chapters selected effective algorithms for the Moore-Penrose pseudoinverse matrix are described and numerically verified. Due to our field of interest (pseudoinverses of stiffness matrices), we limit the forthcoming discussion to real matrices ($\mathbf{A} \in \mathbb{R}^{M \times N}$). In this paper, a group of pseudoinverse computation methods based on neural networks has not been covered. This approach can be found in papers [23, 24].

3. Computing Pseudoinverse Using Singular Value Decomposition (SVD)

A well established and the most reliable algorithm for the pseudoinverse matrix computation is the one based on the singular value decomposition. Every matrix $\mathbf{A} \in \mathbb{R}^{M \times N}$ can be decomposed into $\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$, where $\mathbf{U} \in \mathbb{R}^{M \times M}$ and $\mathbf{V} \in \mathbb{R}^{N \times N}$ are orthogonal matrices and $\mathbf{\Sigma} \in \mathbb{R}^{M \times N}$ is a diagonal matrix with nonnegative entries being singular values of $\mathbf{A}$. The columns of matrices $\mathbf{U}$ and $\mathbf{V}$ are called, respectively, the left- and right-singular vectors of $\mathbf{A}$.

Using the SVD of $\mathbf{A}$, the matrix $\mathbf{A}^\dagger$ is computed as $\mathbf{A}^\dagger = \mathbf{V}\mathbf{\Sigma}^\dagger\mathbf{U}^T$, where $\mathbf{\Sigma}^\dagger$ is obtained taking the reciprocal of nonzero diagonal elements of $\mathbf{\Sigma}$ and transposing of such a matrix. Numerically, by zero one means value smaller than, e.g., $TOL = \epsilon \cdot \max(M, N) \cdot \max(\mathbf{\Sigma})$, where ϵ is a machine precision, whereas M and N are the dimensions of the matrix $\mathbf{\Sigma}$.

One of the first practical algorithms was the method proposed in 1958 by Hestenes [25], which was based on the Givens rotations of the input matrix. Later on, two fundamental SVD algorithms were presented: Golub-Kahan (1965) [26] and its modification, Golub-Reinsch (1970) [27]. They are two step algorithms based on the Householder reflections transforming the input matrix to an upper bidiagonal form at the first step and applying the modified QR algorithm to compute the SVD of this perturbed matrix subsequently. The case of very small singular values was covered by the Demmel-Kahan algorithm [13]. Other algorithms are, for instance, the one-sided Jacobi orthogonalization [28] and the divide-and-conquer methods [29]. Their idea is to modify the second step of SVD computation using other eigenvalue algorithms. The state-of-the-art review with more SVD algorithms tailored for specific matrices can be found in [30].

4. Computing Pseudoinverse by Tikhonov Regularization

Jhurani [7, 8] suggested this method for the pseudoinverse computation as it can easily account for the sparsity of the input matrix $\mathbf{A}$. Using the Tikhonov regularized matrix [31] in the limit we obtain

$$\mathbf{A}^\dagger = \lim_{\delta \rightarrow 0} (\mathbf{A}^T \mathbf{A} + \delta \mathbf{I})^{-1} \mathbf{A}^T = \lim_{\delta \rightarrow 0} \mathbf{A}^T (\mathbf{A} \mathbf{A}^T + \delta \mathbf{I})^{-1} \quad (7)$$

where $\mathbf{I}$ is the identity matrix of the size compatible with $\mathbf{A}^T \mathbf{A}$ or $\mathbf{A} \mathbf{A}^T$. Assuming δ as a small positive number one can compute an approximation of $\mathbf{A}^\dagger$. Usually, δ is assumed as the small multiple of the squared smallest nonzero singular value of $\mathbf{A}$. Nevertheless, for a good approximation of δ , singular values have to be computed, slowing down the whole routine.

5. Computing Pseudoinverse by Iterative Methods

5.1. Higher Order Iterative Methods. In [19] Ben-Israel presents a group of methods leading to the pseudoinverse computation. For a nonzero matrix $\mathbf{A} \in \mathbb{R}^{M \times N}$ with the

initial approximation $\mathbf{X}_0$ and its residual $\mathbf{R}_0$ satisfying $\mathbf{X}_0 = \mathbf{A}^T \mathbf{B} \mathbf{A}^T$ ($\mathbf{B} \in \mathbb{R}^{M \times N}$) and $\rho(\mathbf{R}_0) < 1$, respectively, for any integer $p \geq 2$ the sequence

$$\mathbf{X}_{k+1} = \mathbf{X}_k \left(\mathbf{I} + (\mathbf{I} - \mathbf{A} \mathbf{X}_k) + (\mathbf{I} - \mathbf{A} \mathbf{X}_k)^2 + \dots + (\mathbf{I} - \mathbf{A} \mathbf{X}_k)^{p-1} \right) \quad (8)$$

converges to $\mathbf{A}^\dagger$ with $k \rightarrow \infty$.

The crucial assumption is the initial approximation of $\mathbf{X}_0$. Usually, it is assumed as

$$\mathbf{X}_0 = \beta \mathbf{A}^T, \quad (9)$$

with a scalar β being bounded as follows:

$$0 < \beta < \frac{2}{\lambda_{\max}(\mathbf{A} \mathbf{A}^T)}. \quad (10)$$

It was demonstrated in [19] that the approximate optimum order p is the integer $p \geq 2$ minimizing

$$f(p) = \begin{cases} \frac{p-2+2(N/M)}{\ln p}, & p \neq 2^q, q = 1, 2, \dots \\ \frac{2(q-1+(N/M))}{q \ln 2}, & p = 2^q, q = 1, 2, \dots \end{cases} \quad (11)$$

where M and N are dimensions of matrix $\mathbf{A}$.

5.2. Iterative Methods Based on the Karush-Kuhn-Tucker Equations. Most of the numerical algorithms for the Moore-Penrose pseudoinverse are based on factorization. Despite their high reliability, they are computationally costly. Moreover, in the case of iterative methods, a wrong initial approximation deteriorates the convergence rate or even leads to the lack of convergence.

In [15] the authors proposed a novel iterative method. They formulated the pseudoinverse matrix computation problem as the matrix norm optimization problem with the Karush-Kuhn-Tucker conditions. Subsequently, the acceleration scheme was demonstrated. They proved in [15] that their method is globally convergent for arbitrary initial conditions.

Authors presented two versions of the algorithm for rectangular matrices $M \geq N$ and $M < N$, where M and N are dimensions of the input matrix $\mathbf{A}$. For the sake of brevity, only the first case is presented.

For $M \geq N$ and a given matrix $\mathbf{Y}^{(0)} \in \mathbb{R}^{N \times M}$ as well as a positive scalar ϵ we compute iteratively $k = 0, 1, \dots, L$, where L is the maximum iteration number, for $i = 1, \dots, M$

$$h_i^{(k)} = \frac{\|\mathbf{a}_i^T - \mathbf{B}_1 \mathbf{y}_i^{(k)}\|^2}{\|(\mathbf{I}_N + \mathbf{B}_1)(\mathbf{a}_i^T - \mathbf{B}_1 \mathbf{y}_i^{(k)})\|^2}, \quad (12)$$

where $\mathbf{B}_1 = (\mathbf{A}^T \mathbf{A})^2$, $\mathbf{I}_N$ is the identity matrix of dimensions $N \times N$, $\mathbf{A}^T = [\mathbf{a}_1, \dots, \mathbf{a}_M]$, and $\mathbf{Y}^{(k)} = [\mathbf{y}_1^{(k)}, \dots, \mathbf{y}_M^{(k)}]$. Then we compute $\mathbf{Y}^{(k+1)}$ as

$$\mathbf{Y}^{(k+1)} = \mathbf{Y}^{(k)} + \mathbf{W}_N (\mathbf{A}^T - \mathbf{B}_1 \mathbf{Y}^{(k)}) \mathbf{H}^{(k)}, \quad (13)$$

with $\mathbf{H}^{(k)} = \text{diag}(h_1^{(k)}, \dots, h_M^{(k)})$ and $\mathbf{W}_N = (\mathbf{I}_N + \mathbf{B}_1)$. If stopping criteria (i.e., $\|\mathbf{A}^T - \mathbf{B}_1 \mathbf{Y}^{(k+1)}\|_F^2 \leq \epsilon$ and $k \geq L$) are satisfied, then the pseudoinverse matrix is computed as follows:

$$\mathbf{A}^\dagger = \mathbf{A}^T \mathbf{A} \mathbf{Y}^{(k+1)} \quad (14)$$

Subsequently, the authors eliminated from the afore described algorithm a time step $h_i^{(k)}$, which is matrix $\mathbf{A}$ size dependent and thus convergence deteriorating in the case of large matrices. The obtained accelerated algorithm is as follows.

For $M \geq N$ and given matrix $\mathbf{Y}^{(0)} \in \mathbb{R}^{N \times M}$ as well as positive scalar ϵ we compute iteratively ($k = 0, 1, \dots, L$)

$$\mathbf{Y}^{(k+1)} = \mathbf{Y}^{(k)} + \mathbf{W}_N (\mathbf{A}^T - \mathbf{B}_1 \mathbf{Y}^{(k)}), \quad (15)$$

where $\mathbf{B}_1 = (\mathbf{A}^T \mathbf{A})^2$ and $\mathbf{W}_N = (\mathbf{I}_N + (\mathbf{A}^T \mathbf{A})^2)^{-1}$. If stopping criteria (i.e., $\|\mathbf{A}^T - \mathbf{B}_1 \mathbf{Y}^{(k+1)}\|_F^2 \leq \epsilon$ and $k \geq L$) are satisfied, then the pseudoinverse matrix is computed as follows:

$$\mathbf{A}^\dagger = \mathbf{A}^T \mathbf{A} \mathbf{Y}^{(k+1)} \quad (16)$$

The inverse matrix that has to be computed in the afore presented algorithm is evaluated only once because it does not contain any iterative term.

6. The qrginv Method

In [16] the authors proposed an efficient and elegant method (called ginv) for the pseudoinverse of matrix $\mathbf{A} \in \mathbb{R}^{M \times N}$ computation. Their concept is based on a special type of the tensor product of two vectors. However, it was limited to full rank rectangular matrices and to square matrices with at least one zero row or column (with the rest of the matrix being full rank).

This method was extended to arbitrary singular matrices in [17] constituting the qrginv method. In order to obtain this, QR factorization and the reverse order law for generalized inverse matrices were used.

Authors in [16] consider for any $\mathbf{x} \in \mathbb{R}^K$ ($N < K$) the mapping

$$\mathbf{e} \otimes \mathbf{f} : \mathbb{R}^K \longrightarrow \mathbb{R}^K \quad \text{with} \quad (\mathbf{e} \otimes \mathbf{f})(\mathbf{x}) = \langle \mathbf{x}, \mathbf{e} \rangle \mathbf{f} \quad (17)$$

with $\{\mathbf{e}_1, \dots, \mathbf{e}_N\}$ and $\{\mathbf{f}_1, \dots, \mathbf{f}_N\}$ being the sets of orthonormal and linearly independent vectors of $\mathbb{R}^K$. $\langle \cdot, \cdot \rangle$ denotes the standard inner product in $\mathbb{R}^N$. Every rank- N operator $\mathbf{A}$ can be expressed as

$$\mathbf{A} = \sum_{i=1}^N \mathbf{e}_i \otimes \mathbf{f}_i. \quad (18)$$

An operator $\mathbf{A}$ is linear and its matrix representation can be written in the following form:

$$\mathbf{A} = \begin{bmatrix} \mathbf{f}_{11} & \mathbf{f}_{21} & \dots & \mathbf{f}_{N1} & 0 & \dots & 0 \\ \mathbf{f}_{12} & \mathbf{f}_{22} & \dots & \mathbf{f}_{N2} & 0 & \dots & 0 \\ \vdots & \vdots & \dots & \vdots & \vdots & \vdots & \vdots \\ \mathbf{f}_{1K} & \mathbf{f}_{2K} & \dots & \mathbf{f}_{NK} & 0 & \dots & 0 \end{bmatrix}. \quad (19)$$

with $\{\mathbf{e}_1, \dots, \mathbf{e}_N\}$ being first N elements of the standard $\mathbb{R}^K$ basis and $\mathbf{f}_i = (\mathbf{f}_{i1}, \mathbf{f}_{i2}, \dots, \mathbf{f}_{iK})$ constituting i -th column of the matrix $\mathbf{A}$, $i = 1, 2, \dots, N$.

Recalling Theorem 3.2 from [32] the authors proposed the procedure for the Moore-Penrose inverse of the corresponding tensor-product matrix.

Theorem 1. Let H be a Hilbert space. If $\mathbf{A} = \sum_{i=1}^N \mathbf{e}_i \otimes \mathbf{f}_i$ is a rank- N operator, then its generalized inverse is also a rank- N operator and for each $\mathbf{x} \in H$ it is defined by the relation

$$\mathbf{A}^\dagger \mathbf{x} = \sum_{i=1}^N \lambda_i(\mathbf{x}) \mathbf{e}_i \quad (20)$$

where the unknown functions λ_i are the solutions of an appropriately defined $N \times N$ linear system of equations.

The system of equations resulting from the properties of pseudoinverse matrices has a following form:

$$\langle \mathbf{x}, \mathbf{f}_i \rangle = \sum_{j=1}^N \lambda_j(\mathbf{x}) \langle \mathbf{f}_i, \mathbf{f}_j \rangle, \quad i = 1, 2, \dots, N \quad (21)$$

The matrix of coefficients $\mathbf{G} \in N \times N$ is called the Gram matrix ($g_{ij} = \langle \mathbf{f}_i, \mathbf{f}_j \rangle$). For each $j = 1, 2, \dots, N$ we look for $\lambda_i(\mathbf{e}_j)$ in the following expression:

$$\mathbf{A}^\dagger \mathbf{e}_j = \sum_{i=1}^N \lambda_i(\mathbf{e}_j) \mathbf{e}_i, \quad (22)$$

where $\lambda_i(\mathbf{e}_j)$ for $i = 1, 2, \dots, N$ constitutes the i -th row of pseudoinverse matrix $\mathbf{A}^\dagger$ that has a form

$$\mathbf{A}^\dagger = \begin{bmatrix} \lambda_1(\mathbf{e}_1) & \lambda_1(\mathbf{e}_2) & \dots & \lambda_1(\mathbf{e}_K) \\ \lambda_2(\mathbf{e}_1) & \lambda_2(\mathbf{e}_2) & \dots & \lambda_2(\mathbf{e}_K) \\ \vdots & \vdots & \vdots & \vdots \\ \lambda_N(\mathbf{e}_1) & \lambda_N(\mathbf{e}_2) & \dots & \lambda_N(\mathbf{e}_K) \\ 0 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & 0 \end{bmatrix}. \quad (23)$$

A generalization of the ginv method to arbitrary singular matrices was proposed in [17]. The authors proved the following theorem.

Theorem 2. Let $\mathbf{A} = \mathbf{QR}$ be the QR factorization of $\mathbf{A}$. Then, $\mathbf{A}^\dagger = \mathbf{R}^\dagger \mathbf{Q}^*$.

Using theorem 3.3.11 from [33] they proposed the formula for $\mathbf{A}^\dagger$ using the properties of matrix $\mathbf{R}$.

Theorem 3. Let $\mathbf{A} \in \mathbb{R}^{N \times M}$ be a matrix with $\text{rank}(\mathbf{A}) = r > 0$. Then, there exist matrices $\widehat{\mathbf{A}}$, $\mathbf{Q}$, $\mathbf{R}$ such that $\widehat{\mathbf{A}}$ is obtained by $\mathbf{A}$ by permuting its columns, $\mathbf{Q} \in \mathbb{R}^{N \times N}$ is orthogonal, $\mathbf{R} = \begin{bmatrix} \mathbf{R}_{11} & \mathbf{R}_{12} \\ 0 & 0 \end{bmatrix} \in \mathbb{R}^{N \times M}$, $\mathbf{R}_{11} \in \mathbb{R}^{r \times r}$ is nonsingular and upper triangular, and $\widehat{\mathbf{A}} = \mathbf{QR}$.

Since $\widehat{\mathbf{A}} = \mathbf{A}\mathbf{P}$, where $\mathbf{P}$ is a permutation matrix from the $\mathbf{QR}$ decomposition, it can be written that $\mathbf{A}\mathbf{P} = \mathbf{Q}\mathbf{R}$. Finally, the authors observed using Theorem 2 and properties of $\mathbf{P}$ that

$$\mathbf{A}^\dagger = \mathbf{P}\mathbf{R}^\dagger\mathbf{Q}^T, \quad (24)$$

where the matrix $\mathbf{R}^\dagger$ is computed using `ginv` method. Following [34], authors compute the rank of $\mathbf{R}_{11}$ as the number of columns, in which at least one coefficient with the absolute value greater than the set tolerance exists. As in [34], it is assumed to be equal to 10^{-5} .

7. Numerical Tests

The local numerical homogenization routine uses pseudoinverses of the stiffness matrices. Thus, for our tests we have chosen several of such matrices. They are assembled in macroelement domains fine mesh matrices with no kinematic boundary conditions accounted for.

We use the standard Bubnov-Galerkin approach to solve a given boundary value problem. First, we multiply the governing equation by an arbitrary test function v and integrate both sides over the analyzed domain. Then, we integrate by parts selected left hand side terms to decrease the derivative order of the searched function and to account for the prescribed boundary conditions. As a consequence, the derivative of the test function is introduced. We assume that both the solution and test function are approximated by the same basis functions. Solution to the obtained variational formulation over the domain discretized into a number of finite elements constitutes the set of degrees of freedom (coefficients) of the assumed basis functions. Typically, polynomials of various orders are used. In this manuscript, we understand the approximation order as the order of these functions. Stiffness matrices, that are the field of our interest, are the left hand side coefficient matrices. Due to the discretization and the local basis function supports, the global stiffness matrix is the assembly of the element-wise quantities. In order to solve the resulting system of linear algebraic equation, one needs to account for the kinematic boundary conditions. However, for our purposes, we operate on pure stiffness matrices without any modification. They are singular and positive-definite. In the case of linear basis functions, these matrices are also banded. The bandwidth is the number of degrees of freedom of a single element. The rank of stiffness matrices is approximately equal (slightly smaller, obviously, due to their singularity) to the global number of degrees of freedom. The condition number depends on the approximation order, size of the element, and material parameters (e.g., for the most complex matrix analyzed in Test 5, it is equal to $6.4 \cdot 10^{13}$). Further details on the FEM and higher order elements can be found, e.g., in [18, 35].

7.1. 1D Tests. For the 1D tests we analyze a simple example of a bar of length L , the Young modulus E , and cross-sectional

area A for which the governing differential equation is the following:

$$\begin{aligned} -AEu'' &= q, \\ x &\in [0, L] \end{aligned} \quad (25)$$

7.1.1. Test 1. Two elements of equal lengths with linear shape functions were used for $L = 2$ ($AE = 1$). The ij -th entry of the element stiffness matrix is $k_{ij}^{el} = \int_{el} \varphi_i' AE \varphi_j' dx$, where φ is a given basis function. The assembled matrix is then

$$\mathbf{K} = \begin{bmatrix} 1 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 1 \end{bmatrix} \quad (26)$$

with the pseudoinverse

$$\mathbf{K}^\dagger = \begin{bmatrix} \frac{5}{9} & -\frac{1}{9} & -\frac{4}{9} \\ -\frac{1}{9} & \frac{2}{9} & -\frac{1}{9} \\ -\frac{4}{9} & -\frac{1}{9} & \frac{5}{9} \end{bmatrix} \quad (27)$$

Results for this test are shown in Table 1. For every solution obtained using methods from Sections 3 to 6 we compare CPU time and verify if the conditions for the Moore-Penrose pseudoinverse (3÷6) hold. The values shown in the tables are the averages of 100 program runs. The SVD solution was obtained using the Matlab's `pinv` function that uses the algorithm from [13]. In the case of the method presented in Section 5.2, only its accelerated version is considered in the tests.

Remarks

- (i) For Matlab's `pinv` default TOL was used.
- (ii) The parameter δ in the Tikhonov regularization was set to $1 \cdot 10^{-9}$, even though the squared minimal nonzero singular value equals $8.1079 \cdot 10^{-17}$. For such a small number pseudoinverse cannot be found, since the matrices to be inverted in (7) are singular to working precision. Thus, the next larger singular value was assumed for δ approximation.
- (iii) In Ben-Israel's algorithm $p = 3$ and $\beta = 1/9$ for the initial matrix were used. The stopping criteria were used as follows: the maximum number of iterations cannot exceed 100 or $\|\mathbf{X}_{k+1} - \mathbf{X}_k\|_F / \|\mathbf{X}_k\|_F$ is less than $1 \cdot 10^{-8}$.
- (iv) In the accelerated scheme $\mathbf{Y}_0 = 1 \cdot 10^{-9} \mathbf{K}$. The tolerance for the error measured as proposed in [15] was set to $1 \cdot 10^{-8}$.

7.1.2. Test 2. The next test is a repetition of the previous with higher approximation order, $p_1 = 2$ for the first element and

TABLE 1: CPU time and relative errors, Test 1.

Method	Time [s]	$\frac{\ KK^{\dagger}K - K\ _F}{\ K\ _F}$	$\frac{\ K^{\dagger}KK^{\dagger} - K^{\dagger}\ _F}{\ K^{\dagger}\ _F}$	$\frac{\ (KK^{\dagger})^T - KK^{\dagger}\ _F}{\ KK^{\dagger}\ _F}$	$\frac{\ (K^{\dagger}K)^T - K^{\dagger}K\ _F}{\ K^{\dagger}K\ _F}$
SVD	0.0009	$0.1994 \cdot 10^{-15}$	$0.4083 \cdot 10^{-15}$	$0.2289 \cdot 10^{-15}$	$0.3284 \cdot 10^{-15}$
[8]	0.0005	$0.1131 \cdot 10^{-06}$	$0.0377 \cdot 10^{-06}$	$0.1460 \cdot 10^{-06}$	0*
[19]	0.0012	$0.0142 \cdot 10^{-14}$	$0.1776 \cdot 10^{-14}$	$0.0324 \cdot 10^{-14}$	$0.0096 \cdot 10^{-14}$
[15]	0.0008	$0.1930 \cdot 10^{-04}$	$0.5790 \cdot 10^{-04}$	$0.8899 \cdot 10^{-15}$	$0.5382 \cdot 10^{-15}$
[17]	0.0007	$0.2538 \cdot 10^{-15}$	$0.3187 \cdot 10^{-15}$	$0.2289 \cdot 10^{-15}$	$0.7065 \cdot 10^{-15}$

*Up to the machine precision.

$p_2 = 3$ for the second one. The Peano shape functions were used. The assembled stiffness matrix is as follows:

$$\mathbf{K} = \begin{bmatrix} 1 & -1 & 0 & 0 & 0 & 0 \\ -1 & 2 & 0 & -1 & 0 & 0 \\ 0 & 0 & \frac{1}{3} & 0 & 0 & 0 \\ 0 & -1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{3} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{5} \end{bmatrix} \quad (28)$$

and its exact pseudoinverse is

$$\mathbf{K}^\dagger = \begin{bmatrix} \frac{5}{9} & -\frac{1}{9} & 0 & -\frac{4}{9} & 0 & 0 \\ -\frac{1}{9} & \frac{2}{9} & 0 & -\frac{1}{9} & 0 & 0 \\ 0 & 0 & 3 & 0 & 0 & 0 \\ -\frac{4}{9} & -\frac{1}{9} & 0 & \frac{5}{9} & 0 & 0 \\ 0 & 0 & 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 0 & 0 & 5 \end{bmatrix}. \quad (29)$$

A comparison of pseudoinverse algorithms is shown in Table 2.

7.1.3. Test 3. In this test we analyze the same domain as previously but the material is heterogeneous. The Young moduli are equal to $E_1 = 5.1 \cdot 10^2$ and $E_2 = 5.1 \cdot 10^3$. The finite element mesh consists of 10 elements with the Young modulus equal to E_1 and E_2 interchangeably. Randomly selected approximation orders (5, 8, 9, 2, 6, 5, 1, 4, 2, 8) for the elements were used. For the sake of clarity only results of pseudoinverse computations are shown in a tabular form (Table 3). The parameter δ in the Tikhonov regularization was set equal to $1.3 \cdot 10^{-3}$.

In the case of the 1D tests all of the tested algorithms provided reliable results in a short time as the matrices $\mathbf{K}$ were very small. Concluding 1D tests part,

- (i) the SVD was used with default Matlab's `pinv` tolerance with a good performance;
- (ii) the Tikhonov regularization is very sensitive to parameter δ . Looking for δ 's giving reasonable results was very time consuming. Evaluating their values, performed in terms of singular values, is very costly and rather impractical in the case of large matrices;
- (iii) the method proposed in [19] turned out to be very competitive;
- (iv) the KKT based method was always convergent but its efficiency very highly depends on the initial approximation of $\mathbf{Y}_0$;
- (v) the method `qrginv` showed the best performance. Frequently, it was even better in terms of time and accuracy than Matlab's `pinv`.

Only the SVD [13], Ben-Israel's [19] higher order iterative method and the `qrginv` algorithm [17] are considered in the next chapter for the 3D tests.

7.2. 3D Tests. For the 3D tests, we analyze a homogeneous cube of dimensions $1 \times 1 \times 1$. The whole domain was discretized with 8 equal finite elements. Linear elasticity case was considered. The Young modulus $E = 1 \cdot 10^6$ and the Poisson ratio $\nu = 0.3$.

7.2.1. Test 4a. In this test, linear shape functions were used. The matrix $\mathbf{K}$ has dimensions 81×81 . Results of Moore-Penrose pseudoinverse computations are shown in Table 4.

7.2.2. Test 4b. In the previous test, the SVD-based algorithm gave the worst results. It was due to the default *TOL* used for `pinv` Matlab's function. This parameter determines how small singular value is treated as zero. In order to obtain results with higher accuracy we proposed Algorithm 1:

Updated results for the SVD-based algorithm are shown in Table 5.

7.2.3. Test 5. In this test, the Peano shape functions of the third order were used for the approximation at each direction. The matrix $\mathbf{K}$ has dimensions 1029×1029 . We provide this matrix in the supplementary file `matrix.K.mat` that was created using Matlab. Results of pseudoinverse computations are shown in Table 6.

Looking at Tables 5 and 6, it can be observed that the method proposed in [19] did not converge. The algorithm was stopped only because of the exceeding of the maximum iteration number. Being unconditionally convergent, the method turns out to be rather impractical in real applications. The `pinv` and `qrginv` functions turned out to be very fast. Unfortunately, homogenization based on their application yields unreliable results. The higher the approximation order was, the worse the final results were. We dealt with a super problem with the pseudoinversion embedded thus. Without this super problem, one could consider the results shown in Table 5 (for the SVD and `qrginv`) as the reliable ones. Poor results of the trivial local numerical homogenization example (uniaxial test for a homogeneous material) led us to a thorough analysis. It ended with an interesting observation concerning the pseudoinverse matrix computation described below.

Let us denote pseudoinverse matrices obtained using `pinv` and `qrginv` functions, respectively, $\mathbf{K}_1^\dagger$ and $\mathbf{K}_2^\dagger$. Consequently, their pseudoinverses are $\mathbf{K}_1$ and $\mathbf{K}_2$. One may compute the following error indicators:

- (i) In Test 4b: $\|\mathbf{K} - \mathbf{K}_1\|_F / \|\mathbf{K}\|_F = 1.0434 \cdot 10^{-13}$ but $\|\mathbf{K} - \mathbf{K}_2\|_F / \|\mathbf{K}\|_F = 1$, when $\|\mathbf{K}_1 - \mathbf{K}_2\|_F / \|\mathbf{K}_1\|_F = 5.2436 \cdot 10^{-12}$.
- (ii) In Test 5: $\|\mathbf{K} - \mathbf{K}_1\|_F / \|\mathbf{K}\|_F = 1$, $\|\mathbf{K} - \mathbf{K}_2\|_F / \|\mathbf{K}\|_F = 1$ and $\|\mathbf{K}_1 - \mathbf{K}_2\|_F / \|\mathbf{K}_1\|_F = 12.7059(!)$.

Let us remind that in the 4b and 5 tests the approximation orders equal to 1 and 3, respectively. The above presented

TABLE 2: CPU time and relative errors, Test 2.

Method	Time [s]	$\frac{\ KK^\dagger K - K\ _F}{\ K\ _F}$	$\frac{\ K^\dagger KK^\dagger - K^\dagger\ _F}{\ K^\dagger\ _F}$	$\frac{\ (KK^\dagger)^T - KK^\dagger\ _F}{\ KK^\dagger\ _F}$	$\frac{\ (K^\dagger K)^T - K^\dagger K\ _F}{\ K^\dagger K\ _F}$
SVD	0.0014	$0.1968 \cdot 10^{-15}$	$0.0648 \cdot 10^{-15}$	$0.1448 \cdot 10^{-15}$	$0.2077 \cdot 10^{-15}$
[8]	0.0010	$0.1786 \cdot 10^{-05}$	$0.0096 \cdot 10^{-05}$	$0.1477 \cdot 10^{-05}$	0*
[19]	0.0022	$0.0011 \cdot 10^{-13}$	$0.1189 \cdot 10^{-13}$	$0.0016 \cdot 10^{-13}$	$0.0045 \cdot 10^{-13}$
[15]	0.0979	$0.0312 \cdot 10^{-03}$	$0.3759 \cdot 10^{-03}$	$0.0311 \cdot 10^{-13}$	$0.0348 \cdot 10^{-13}$
[17]	0.0004	$0.2505 \cdot 10^{-15}$	$0.5060 \cdot 10^{-15}$	$0.1448 \cdot 10^{-15}$	$0.4469 \cdot 10^{-15}$

*Up to the machine precision.

TABLE 3: CPU time and relative errors, Test 3.

Method	Time [s]	$\frac{\ KK^\dagger K - K\ _F}{\ K\ _F}$	$\frac{\ K^\dagger KK^\dagger - K^\dagger\ _F}{\ K^\dagger\ _F}$	$\frac{\ (KK^\dagger)^T - KK^\dagger\ _F}{\ KK^\dagger\ _F}$	$\frac{\ (K^\dagger K)^T - K^\dagger K\ _F}{\ K^\dagger K\ _F}$
SVD	0.0059	$0.0005 \cdot 10^{-11}$	$0.0434 \cdot 10^{-11}$	$0.0644 \cdot 10^{-11}$	$0.2611 \cdot 10^{-11}$
[8]	0.0025	$0.6177 \cdot 10^{-06}$	$0.0095 \cdot 10^{-00}$	$0.3379 \cdot 10^{-06}$	$0.8813 \cdot 10^{-07}$
[19]	0.0170	$0.0001 \cdot 10^{-10}$	$0.3446 \cdot 10^{-10}$	$0.0002 \cdot 10^{-10}$	$0.0150 \cdot 10^{-10}$
[15]	3.4590	$0.2453 \cdot 10^{-06}$	$0.1265 \cdot 10^{-03}$	$0.3349 \cdot 10^{-05}$	$0.0016 \cdot 10^{-00}$
[17]	0.0026	$0.0030 \cdot 10^{-12}$	$0.0538 \cdot 10^{-12}$	$0.0429 \cdot 10^{-12}$	$0.1366 \cdot 10^{-12}$

TABLE 4: CPU time and relative errors, Test 4a.

Method	Time [s]	$\frac{\ KK^TK - K\ _F}{\ K\ _F}$	$\frac{\ K^TKK^T - K^T\ _F}{\ K^T\ _F}$	$\frac{\ (KK^T)^T - KK^T\ _F}{\ KK^T\ _F}$	$\frac{\ (K^TK)^T - K^TK\ _F}{\ K^TK\ _F}$
SVD	0.0078	$0.4808 \cdot 10^{-03}$	$0.2340 \cdot 10^{-02}$	$0.4283 \cdot 10^{-02}$	$0.4977 \cdot 10^{-02}$
[19]	0.0280	$0.7028 \cdot 10^{-13}$	$0.3091 \cdot 10^{-09}$	$0.1318 \cdot 10^{-14}$	$0.1547 \cdot 10^{-14}$
[17]	0.0069	$0.7028 \cdot 10^{-13}$	$0.1572 \cdot 10^{-14}$	$0.1809 \cdot 10^{-11}$	$0.5138 \cdot 10^{-14}$

```

Set  $TOL_{pinv} = 1, TOL = 1 \cdot 10^{-6}$ 
while  $icond < 4$  do
   $icond = 0$ 
   $TOL_{pinv} = 0.1 \cdot TOL_{pinv}$ 
  if  $TOL_{pinv} < 1 \cdot 10^{-20}$  then
     $TOL_{pinv} = 1 \cdot 10^{-1}$  and  $TOL = 10 \cdot TOL$ 
  end if
   $K^\dagger = pinv(K, TOL_{pinv})$ 
  if  $\|KK^\dagger K - K\|_F / \|K\|_F < TOL$  then
     $icond = icond + 1$ 
  end if
  if  $\|K^\dagger KK^\dagger - K^\dagger\|_F / \|K^\dagger\|_F < TOL$  then
     $icond = icond + 1$ 
  end if
  if  $\|(KK^\dagger)^T - KK^\dagger\|_F / \|KK^\dagger\|_F < TOL$  then
     $icond = icond + 1$ 
  end if
  if  $\|(K^\dagger K)^T - K^\dagger K\|_F / \|K^\dagger K\|_F < TOL$  then
     $icond = icond + 1$ 
  end if
end while
return  $K^\dagger$ 

```

ALGORITHM 1: Given K compute its pseudoinverse $K^\dagger$ using pinv function.

TABLE 5: CPU time and relative errors, Test 4b.

Method	Time [s]	$\frac{\ KK^\dagger K - K\ _F}{\ K\ _F}$	$\frac{\ K^\dagger KK^\dagger - K^\dagger\ _F}{\ K^\dagger\ _F}$	$\frac{\ (KK^\dagger)^T - KK^\dagger\ _F}{\ KK^\dagger\ _F}$	$\frac{\ (K^\dagger K)^T - K^\dagger K\ _F}{\ K^\dagger K\ _F}$
SVD	0.0064	$0.7039 \cdot 10^{-13}$	$0.2704 \cdot 10^{-14}$	$0.6579 \cdot 10^{-14}$	$0.5456 \cdot 10^{-14}$

numbers practically disable the application of the local numerical homogenization with Peano higher order shape functions. As it was mentioned at the beginning of this paper, one has to compute the pseudoinverse twice. First, the input matrix is the assembled stiffness matrix. Even if it is pseudoinverted accurately, then (after some further calculations) one has to pseudoinvert the pseudoinverse of the effective macroscale stiffness matrix. So far, we have not found the algorithm giving for matrices of our interest (using Peano higher order shape functions) $(K^\dagger)^\dagger = K$. In Test 4b, matrices K_1 and K_2 have almost the same norms, however, only the error indicator $\|K - K_{computed}\|_F / \|K\|_F$ computed for K_1 is satisfactory. In Test 5, all three error indicators were prohibitively large. Only the results of the local numerical homogenization using a linear approximation and the pinv function were correct.

In order to proceed with the local numerical homogenization analysis, we applied the integrated Legendre or Lobatto polynomials (see, e.g., [35]) of higher order. Such basis functions and, consequently, the resulting stiffness matrices exhibit some superior quantities (c.f. [35]) comparing to the Peano polynomials. They enabled us to obtain afore presented error indicators ($\|K - K_{computed}\|_F / \|K\|_F$) of order $1 \cdot 10^{-12}$ for the pinv function. Thus, only the Demmel-Kahan method is reliable for pseudoinversion of stiffness matrices. Results for

Test 5 with Legendre and Lobatto shape functions are shown in Tables 7 and 8, respectively.

8. Concluding Remarks

We proposed a very demanding test for the computation of the Moore-Penrose pseudoinverse matrices. The finite element stiffness matrices resulting from higher order approximation are such difficult data for all algorithms of the pseudoinverse computation. It could be concluded that, for elasticity stiffness matrices, the Demmel-Kahan algorithm was the only suitable method for our purposes. The proposed matrices can also serve, however, as benchmarks for new algorithms for the Moore-Penrose pseudoinverse.

The difficulty of the stiffness pseudoinverse computation for the higher order approximation can be explained by the increasing condition number (the ratio of the largest and the smallest nonzero singular values) of these matrices. Stiffness matrices obtained using Peano shape functions exhibit higher condition numbers comparing to the integrated Legendre or Lobatto polynomials. Therefore, only the application of those functions allows one to obtain $(K^\dagger)^\dagger = K$ for arbitrary higher order approximation with a very high accuracy. It was obtained, however, only using the pinv function implementing algorithm presented in [13]. The remaining methods did not pass the test.

TABLE 6: CPU time and relative errors, Test 5 (Peano shape functions).

Method	Time [s]	$\frac{\ KK^\dagger K - K\ _F}{\ K\ _F}$	$\frac{\ K^\dagger KK^\dagger - K^\dagger\ _F}{\ K^\dagger\ _F}$	$\frac{\ (KK^\dagger)^T - KK^\dagger\ _F}{\ KK^\dagger\ _F}$	$\frac{\ (K^\dagger K)^T - K^\dagger K\ _F}{\ K^\dagger K\ _F}$
SVD	1.5543	$0.7596 \cdot 10^{-07}$	$0.1946 \cdot 10^{-10}$	$0.6767 \cdot 10^{-09}$	$0.2657 \cdot 10^{-10}$
[19]	175.44	$0.0022 \cdot 10^{+00}$	$0.0018 \cdot 10^{+00}$	$0.0008 \cdot 10^{+00}$	$1.4142 \cdot 10^{+00}$
[17]	1.4129	$0.0001 \cdot 10^{-09}$	$0.0004 \cdot 10^{-09}$	$0.5837 \cdot 10^{-09}$	$0.1063 \cdot 10^{-09}$

TABLE 7: CPU time and relative errors, Test 5 (Legendre shape functions).

Method	Time [s]	$\frac{\ KK^\dagger K - K\ _F}{\ K\ _F}$	$\frac{\ K^\dagger KK^\dagger - K^\dagger\ _F}{\ K^\dagger\ _F}$	$\frac{\ (KK^\dagger)^T - KK^\dagger\ _F}{\ KK^\dagger\ _F}$	$\frac{\ (K^\dagger K)^T - K^\dagger K\ _F}{\ K^\dagger K\ _F}$
SVD	1.4112	$0.0047 \cdot 10^{-11}$	$0.0344 \cdot 10^{-11}$	$0.1863 \cdot 10^{-11}$	$0.0671 \cdot 10^{-11}$
[19]	101.35	$0.0017 \cdot 10^{+00}$	$0.0018 \cdot 10^{+00}$	$0.0009 \cdot 10^{+00}$	$1.4142 \cdot 10^{+00}$
[17]	1.8285	$0.0005 \cdot 10^{-10}$	$0.0005 \cdot 10^{-10}$	$0.2127 \cdot 10^{-10}$	$0.0054 \cdot 10^{-10}$

TABLE 8: CPU time and relative errors, Test 5 (Lobatto shape functions).

Method	Time [s]	$\frac{\ KK^{\dagger}K - K\ _F}{\ K\ _F}$	$\frac{\ K^{\dagger}KK^{\dagger} - K^{\dagger}\ _F}{\ K^{\dagger}\ _F}$	$\frac{\ (KK^{\dagger})^T - KK^{\dagger}\ _F}{\ KK^{\dagger}\ _F}$	$\frac{\ (K^{\dagger}K)^T - K^{\dagger}K\ _F}{\ K^{\dagger}K\ _F}$
SVD	1.4259	$0.0357 \cdot 10^{-12}$	$0.1948 \cdot 10^{-12}$	$0.5772 \cdot 10^{-12}$	$0.2635 \cdot 10^{-12}$
[19]	107.22	$0.0013 \cdot 10^{+00}$	$0.0017 \cdot 10^{+00}$	$0.0009 \cdot 10^{+00}$	$1.4142 \cdot 10^{+00}$
[17]	1.8454	$0.0004 \cdot 10^{-10}$	$0.0005 \cdot 10^{-10}$	$0.1100 \cdot 10^{-10}$	$0.0020 \cdot 10^{-10}$

Data Availability

The authors have attached to this submission the file “matrix_K”, which stores the stiffness matrix used in Test 5.

Conflicts of Interest

No conflicts of interest can be stated.

Supplementary Materials

The Supplementary Material file stores the global stiffness matrix resulting from the linear elasticity problem. The analyzed domain (a unit cube) was discretized with 8 equal elements. Material data can be found in “Numerical Tests”. The approximation order is equal to 3 and the Peano shape functions were used. The stiffness matrix has dimensions 1029x1029. It was used for the pseudoinversion tests. (*Supplementary Materials*)

References

- [1] J. Wang, P. Guo, and X. Xin, *Chapter Review of Pseudoinverse Learning Algorithm for Multilayer Neural Networks And Applications*, Springer, Cham, pp.99–106, 2018.
- [2] S. Miljkovic, M. Miladinovic, P. S. Stanimirovic, and I. Stojanovic, “Application of the pseudoinverse computation in reconstruction of blurred images,” *Filomat*, vol. 26, pp. 453–465, 2012.
- [3] M. Lustig, D. Donoho, and J. M. Pauly, “The application of compressed sensing for rapid MR imaging,” *Magnetic Resonance in Medicine*, vol. 58, no. 6, pp. 1182–1195, 2007.
- [4] L. Sciavicco and B. Siciliano, *Modelling and Control of Robot Manipulators*, Springer-Verlag, London, 2000.
- [5] A. Albert, “Statistical applications of the pseudo inverse,” in *Generalized verses and Applications*, M. Zuhair Nashed, Ed., pp. 525–548, Academic Press, 1976.
- [6] C. Jhurani and L. Demkowicz, “Multiscale modeling using goal-oriented adaptivity and numerical homogenization. Part I: Mathematical formulation and numerical results,” *Computer Methods Applied Mechanics and Engineering*, 213–216:399–417, 2012.
- [7] C. Jhurani and L. Demkowicz, “Multiscale modeling using goal-oriented adaptivity and numerical homogenization. Part II: Algorithms for the Moore-Penrose pseudoinverse,” *Computer Methods Applied Mechanics and Engineering*, 213–216:418–426, 2012.
- [8] C. K. Jhurani, *Multiscale Modeling Using Goal-oriented Adaptivity and Numerical Homogenization (PhD Thesis)*, The University of Texas at Austin, 2009.
- [9] M. Klimczak and W. Cecot, “Application of local numerical homogenization and hp-adaptive FEM for modeling of heterogeneous viscoelastic materials,” *Engineering Transactions*, vol. 63, no. 3, pp. 317–327, 2015.
- [10] I. Babuska, G. Caloz, and J. E. Osborn, “Special finite element methods for a class of second order elliptic problems with rough coefficients,” *SIAM Journal on Numerical Analysis*, vol. 31, no. 4, pp. 945–981, 1994.
- [11] W. E. B. Engquist, X. Li, W. Ren, and E. Vanden-Eijnden, “Heterogeneous multiscale methods: A review,” *Communications in Computational Physics*, vol. 2, no. 3, pp. 367–450, 2007.
- [12] Y. Efendiev and T. Y. Hou, *Multiscale Finite Element Methods*, Springer-Verlag, New York, NY, USA, 2009.
- [13] J. Demmel and W. Kahan, “Accurate singular values of bidiagonal matrices,” *SIAM Journal on Scientific and Statistical Computing*, vol. 11, no. 5, pp. 873–912, 1990.
- [14] A. Ben-Israel and T. N. E. Greville, *Generalized Inverses: Theory and Applications*, Wiley-Interscience, 1974.
- [15] Y. Xia, T. Chen, and J. Shan, “A novel iterative method for computing generalized inverse,” *Neural Computation*, vol. 26, no. 2, pp. 449–465, 2014.
- [16] V. N. Katsikis and V. D. Pappas, “Fast computing of the Moore-Penrose inverse matrix,” *Electronic Journal of Linear Algebra*, vol. 17, pp. 637–650, 2008.
- [17] V. N. Katsikis, D. Pappas, and A. Petralias, “An improved method for the computation of the Moore-Penrose inverse matrix,” *Applied Mathematics and Computation*, vol. 217, no. 23, pp. 9828–9834, 2011.
- [18] L. Demkowicz, J. Kurtz, D. Pardo, M. Paszynski, W. Rachowicz, and A. Zdunek, “Computing with HP finite elements,” in *Frontiers: Three-Dimensional Elliptic and Maxwell Problems with Applications*, vol. 2, Chapman & Hall/CRC, 2008.
- [19] A. Ben-Israel and T. N. E. Greville, *Generalized Inverses*, Springer-Verlag, 2003.
- [20] E. Moore, “On the reciprocal of the general algebraic matrix,” *Bulletin of the American Mathematical Society*, no. 26, pp. 294–295, 1920.
- [21] R. Penrose, “On best approximate solution of linear matrix equations,” in *Proceedings of the Cambridge Philosophical Society*, vol. 51, pp. 406–413, 1956.
- [22] Y. Wei, P. S. Stanimirovic, and M. Petkovic, *Numerical and Symbolic Computations of Generalized Inverses*, World Scientific, 2018.
- [23] Y. Xia, S. Zhang, and P. S. Stanimirovic, “Neural network for computing pseudoinverses and outer inverses of complex-valued matrices,” *Applied Mathematics and Computation*, vol. 273, no. C, pp. 1107–1121, 2016.
- [24] Y. Wei, “Recurrent neural networks for computing weighted Moore-Penrose inverse,” *Applied Mathematics and Computation*, vol. 116, no. 3, pp. 279–287, 2000.
- [25] M. R. Hestenes, “Inversion of matrices by biorthogonalization and related results,” *Journal of the Society For Industrial & Applied Mathematics*, vol. 6, no. 1, pp. 51–90, 1958.
- [26] G. H. Golub and W. Kahan, “Calculating the singular values and pseudo-inverse of a matrix,” *Journal of the Society For Industrial & Applied Mathematics*, vol. 2, no. 2, pp. 205–224, 1965.
- [27] G. H. Golub and C. Reinsch, “Singular value decomposition and least squares solutions,” *Numerische Mathematik*, vol. 14, no. 5, pp. 403–420, 1970.
- [28] J. Demmel and K. Veselić, “Jacobi’s method is more accurate than QR,” *SIAM Journal on Matrix Analysis and Applications*, vol. 13, no. 4, pp. 1204–1245, 1992.
- [29] L. N. Trefethen and D. Bau, “Numerical linear algebra,” *SIAM*, 1997.
- [30] A. K. Cline and I. S. Dhillon, *Computation of the Singular Value Decomposition*, CRC Press, 2006.
- [31] A. N. Tikhonov, “Regularization of Incorrectly Posed Problems,” *Soviet Mathematics Doklady*, vol. 4, no. 6, pp. 1624–1627, 1963.
- [32] S. Karanasios and D. Pappas, “Generalized inverses and special type operator algebras,” *Facta Universitatis (NIS)*, Series Mathematics and Informatics, vol. 21, pp. 41–48, 2006.

- [33] D. Watkins, *Fundamentals of Matrix Computations*, Wiley-Interscience, 2002.
- [34] F. Toutounian and A. Ataei, "A new method for computing Moore-Penrose inverse matrices," *Journal of Computational and Applied Mathematics*, vol. 228, no. 1, pp. 412–417, 2009.
- [35] L. Demkowicz, "Computing with HP-adaptive finite elements," in *One and Two Dimensional Elliptic and Maxwell Problems*, vol. 1, Chapman & Hall/CRC, 2007.

Submit your manuscripts at
www.hindawi.com