

Research Article

Open Access



A domain decomposition solver for spectral stochastic finite element: an approximate sparse expansion approach

Bowen Luo¹, Wen Cao¹, Zheng Zhou², Hongzhe Dai¹

¹School of Civil Engineering, Harbin Institute of Technology, Harbin 150090, Heilongjiang, China.

²The First Company of China 8th Engineering Bureau Ltd, Jinan 250100, Shandong, China.

Correspondence to: Prof. Hongzhe Dai, School of Civil Engineering, Harbin Institute of Technology, No. 73 Huanghe Road, Harbin 150090, Heilongjiang, China. E-mail: hzdai@hit.edu.cn

How to cite this article: Luo B, Cao W, Zhou Z, Dai H. A domain decomposition solver for spectral stochastic finite element: an approximate sparse expansion approach. *Dis Prev Res* 2024;3:3. <http://dx.doi.org/10.20517/dpr.2023.39>

Received: 27 Nov 2023 **First Decision:** 2 Feb 2024 **Revised:** 4 Mar 2024 **Accepted:** 18 Mar 2024 **Published:** 29 Mar 2024

Academic Editor: George Deodatis **Copy Editor:** Fangling Lan **Production Editor:** Fangling Lan

Abstract

Spectral stochastic finite element (SSFE) has been widely used in the uncertainty quantification of real-life problems. However, the prohibitive computational burden prevents the application of the method in practical engineering systems because an enormous augmented system has to be solved. Although the domain decomposition method has been introduced to SSFE to improve the efficiency for the solution of the augmented system, there still exist significant challenges in solving the extended Schur complement (e-SC) system from domain decomposition method. In this paper, we develop an approximate sparse expansion-based domain decomposition solver to generalize the application of SSFE. An approximate sparse expansion is first presented for the subdomain-level augmented matrix so that the computational cost in each iteration of the preconditioned conjugate gradient is greatly alleviated. Based on the developed sparse expansion, we further establish an approximate sparse preconditioner to accelerate the convergence of the preconditioned conjugate gradient. The developed approximate sparse expansion-based domain decomposition solver is then incorporated in the context of SSFE. Since the difficulties of solving the e-SC system have been overcome, the developed approximate sparse expansion-based solver greatly improves the computational efficiency of the solution of the e-SC system, and thereby, the SSFE is capable of dealing with large-scale engineering systems. Two numerical examples demonstrate that the developed method can significantly enhance the efficiency for the stochastic response analysis of practical engineering systems.



© The Author(s) 2024. **Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License (<https://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, sharing, adaptation, distribution and reproduction in any medium or format, for any purpose, even commercially, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.



Keywords: Stochastic finite element, domain decomposition, approximate sparse expansion, preconditioned conjugate gradient, Schur complement

INTRODUCTION

Common geological hazards, such as debris flows, landslides, and avalanches, are usually devastating, and the timing, location, and scale of disasters are uncertain, posing a significant threat to people's lives and property^[1]. Therefore, researching the ability of structures to resist disasters is particularly important. However, uncertainty is a common occurrence in various aspects of practical engineering, including structure design, manufacturing, operation, and maintenance^[2,3], which can lead to significant deviations in the structural behavior during structural response analysis and directly result in inaccurate analysis results of the ability of the structure to withstand disasters.

So, the realistic design and analysis of these physical systems must consider uncertainties contributed by various sources such as manufacturing variability, insufficient data, unknown physics, and aging^[4-7]. The uncertainties may also significantly influence the predictive capabilities of computer simulations^[8-12]. The latest advancements in high-performance computing and sensing technology have stimulated computational simulations with extremely high resolution, providing great possibilities for integrating effective uncertainty quantification methods to achieve realistic and reliable numerical predictions^[13-18]. Nevertheless, using standard Monte Carlo simulations for uncertainty quantification in the simulation of random responses to large and complex problems may be time-consuming or impractical.

As an alternative method to Monte Carlo simulation, spectral stochastic finite element (SSFE) has been widely applied in the field of stochastic mechanics^[19-21]. Different from various methods that rely on sampling such as the so-called collocation, SSFE is intrusive^[22,23]. It estimates the random response of the structure by transforming the random stiffness equation into a coupled set of a series of deterministic equations^[4]. In the framework of SSFE, if the uncertain properties of the system follow a Gaussian distribution, then they are quantified with the Karhunen-Loeve (KL) expansion; if not, for example, the uncertain properties follow a lognormal distribution, they will be projected into a polynomial chaos (PC) basis^[24,25]. The random response of the system is usually projected into the PC basis because it is unknown and almost never a Gaussian distribution; adopting the Galerkin minimization scheme will establish an augmented system of linear algebraic equations, and only solving this augmented system will determine the coefficients of the basis and then obtain the random response of the system^[19].

However, compared to deterministic equations, the scale of the augmented ones that SSFE needs to solve will be several orders of magnitude larger, because in the analysis of complex structures, the PC expansion (PCE) items of the random response usually reach thousands or even tens of thousands. For such large-scale problems, in the past few years, the customized solution algorithm has been developing continuously, and to some extent, it has achieved successful results in solving such high computational requirements. A kind of method is directly solved by the preconditioned conjugate gradient (PCG) method, and the preconditioned matrix is established using the structure of the coefficient matrix of the augmented system, including the block diagonal preconditioner, successive symmetric over-relaxation, and so on^[4,25]. Such preconditions can effectively reduce the number of iterations of PCG, transforming what actually takes time to calculate into a series of deterministic problems, thus greatly improving the solution efficiency of the augmented system.

Nevertheless, with the increasing scale of the augmented system, the deterministic problems demanding resolution through the direct PCG method are growing, which has gradually approached the number of samples that need to be solved by the Monte Carlo method. Domain decomposition method (DDM) provides another

essential framework for developing fast and efficient solvers for such applications. The DDM is an important strategy for solving large-scale problems in computational mechanics^[26,27] and has been applied to the solution of SSFE in recent years. Using this approach, the solution of augmented systems is transformed into a series of sub-problems about each sub-domain and an extended Schur complement (e-SC) system^[28]. The solution of the e-SC system takes up most of the calculation time when the number of sub-domains is huge. Hence, the efficiency for the solution of the e-SC system is the key in this kind of method. Till now, all these methods are based on the structure of the coefficient matrix of the e-SC system, and PCG is used to solve the e-SC system, which improves the computational efficiency to a certain extent in the random response analysis of structures^[29-34]. However, the above techniques can still not effectively analyze the large-scale structures, and the efficiency is even lower than that of direct conjugate gradient (CG) method. This is because the preconditioner established from the existing methods cannot guarantee the similarity with the e-SC matrix^[35], and as a result, the iteration steps of PCG will be quite huge. In addition, since direct DDM needs to store a large-scale dense matrix in the solution process, the computational cost in each iteration will be quite large, which may further decrease the efficiency of PCG in the stochastic analysis of large-scale structures.

In order to overcome the above deficiencies of the existing methods, we develop an approximate sparse expansion-based domain decomposition solver for stochastic finite element strategy. Firstly, we develop an approximate sparse expansion for the subdomain-level augmented matrix. By utilizing the property of the subdomain-level random stiffness matrix, the subdomain-level augmented matrix can be approximated as a product of a block diagonal matrix and a sparse matrix, which is denoted as approximate sparse expansion in this study. With this expansion, we further establish an algorithm of multiplying the e-SC matrix and an arbitrary vector, which may greatly save the computational cost in each iteration in PCG. Secondly, a preconditioner for the PCG solution of the e-SC system is further developed based on the established approximate sparse expansion. Since the preconditioner is constructed as the product of a block diagonal matrix and a sparse matrix, the inverse of the preconditioner can be readily obtained. More importantly, the preconditioner is approximately equal to the e-SC matrix so that the number of iterations can be significantly reduced. As a result, our approach overcomes the difficulties of solving the e-SC system faced by traditional DDM-based SSFE by establishing the approximate sparse preconditioner, and the efficiency for the solution of e-SC system can be greatly improved compared with the traditional DDM-based SSFE. Therefore, the structural stochastic response analysis will become much more efficient.

The remainder of this paper is organized as follows: In Section 2, the primal domain decomposition for SSFE is briefly described. Then, in Section 3, a domain decomposition solver of stochastic finite element method (SFEM) based on the approximate sparse expansion approach is proposed, and the actual calculation steps are summarized. Section 4 contains numerical examples demonstrating the efficiency of the proposed method. Finally, the concluding remarks are given in Section 5.

DOMAIN DECOMPOSITION-BASED SSFE

Stochastic finite element: spectral approach

In the response analysis of a structural system with stochasticity in the model parameters, the equilibrium equation is usually expressed as

$$\mathbf{K}(\theta)\mathbf{u}(\theta) = \mathbf{f} \quad (1)$$

where $\mathbf{K}(\theta)$ is the stochastic stiffness matrix related to the stochastic process $\omega(x, \theta)$ of the system, $\mathbf{u}(\theta)$ is the random response vector, and \mathbf{f} is the external forcing vector.

In one particular formulation of the SFEM, according to the KL of the stochastic process $E(x, \theta)$, the stochastic stiffness matrix, with the KL expansion^[1], can be given as

$$\mathbf{K}(\theta) = \mathbf{K}_0 + \sum_{i=1}^N \mathbf{K}_i \xi_i \quad (2)$$

where \mathbf{K}_0 is average stiffness, ξ_i is a set of standard Gaussian random variables, and $\{\mathbf{K}_i\}$ are deterministic, which are functions of the eigenvalues and eigenfunctions; we called them stochastic part stiffness matrices in this work.

The resulting vector $\mathbf{u}(\theta)$ of Equation (1), which is expanded along a PCE, is calculated by

$$\mathbf{u}(\theta) = \sum_{j=0}^{M-1} \mathbf{u}_j \psi_j \quad (3)$$

where $\{\psi_j\}$ are multidimensional orthogonal polynomials in $\{\xi_j\}$, and each \mathbf{u}_j denotes a vector of deterministic coefficients. The number of terms M in Equation (3) depends on: (i) the number N of the uncorrelated random variables $\{\xi_j\}$ used to describe the uncertainty; and (ii) the highest degree p of the $\{\psi_j\}$ orthogonal polynomials. This term is calculated using the following formula^[25]:

$$M = (N + p)/(N!p!).$$

Substituting Equations (2) and (3) into (1) and forcing the residual to be orthogonal to the approximating space spanned by the PC $\{\psi_j\}$ yields the following system of linear equations^[36]:

$$\sum_{j=0}^{M-1} \sum_{i=0}^N \langle \xi_i \psi_j \psi_k \rangle \mathbf{K}_i \mathbf{u}_j = \langle \psi_k \mathbf{f} \rangle. \quad (4)$$

These equations can be assembled into a matrix of size $M \cdot n \times M \cdot n$:

$$\begin{bmatrix} \mathbf{K}^{00} & \mathbf{K}^{01} & \dots & \mathbf{K}^{0,M-1} \\ \mathbf{K}^{10} & \mathbf{K}^{11} & \dots & \mathbf{K}^{1,M-1} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{K}^{M-1,0} & \mathbf{K}^{M-1,1} & \dots & \mathbf{K}^{M-1,M-1} \end{bmatrix} \begin{bmatrix} \mathbf{u}_0 \\ \mathbf{u}_1 \\ \vdots \\ \mathbf{u}_{M-1} \end{bmatrix} = \begin{bmatrix} \mathbf{f}^0 \\ \mathbf{f}^1 \\ \vdots \\ \mathbf{f}^{M-1} \end{bmatrix} \quad (5)$$

where $\mathbf{K}^{(j,k)} = \sum_{i=0}^N c_{ijk} \mathbf{K}_i$, and $c_{ijk} = \langle \xi_i \psi_j \psi_k \rangle$

It is evident that the augmented system is an equation that couples the determinism of finite elements with the randomness of structures. The number of rows and columns of its coefficient matrix are the product of: (1) the spatial degrees of freedom (DOF) after meshing the structure; and (2) the number of expansion terms for

random responses using the PC method. When the structural grid is subdivided densely and multiple terms are expanded for random responses, the size of the augmented system can become quite large, which exceeds the computational capacity of current computers. Ultimately, this fact makes it impossible for SSFE to analyze the stochastic response of large-scale structures. Therefore, it is quite necessary to develop efficient algorithms for the augmented system.

Domain decomposition approach for SSFE

When dealing with large-scale problems, DDM is one of the best candidates due to its proven performance and scalability. In this method, the computational domain Ω is partitioned into N_s non-overlapping subdomains $\Omega = \bigcup_{s=1}^{N_s} \{\Omega_s\}$; the random response vector of every typical subdomain s , $\mathbf{u}^s(\theta)$, is divided into interior response vector $\mathbf{u}_I^s(\theta)$ corresponding to the interior nodes and interface response vector $\mathbf{u}_\Gamma^s(\theta)$ corresponding to interface nodes (shared by two or more adjacent subdomains) [26,27].

According to this partition, the DDM numbers, first the subdomains and last the interface, that is, let $\mathbf{u}(\theta) = \{\mathbf{u}_I^1(\theta) \quad \mathbf{u}_I^2(\theta) \cdots \mathbf{u}_I^N(\theta) \quad \mathbf{u}_\Gamma(\theta)\}$, at this time, the vector before any base after PCE of the random response vector $\mathbf{u}(\theta)$ can be expressed as $\mathbf{u}_i = \{\mathbf{u}_{i,I}^1 \quad \mathbf{u}_{i,I}^2 \cdots \mathbf{u}_{i,I}^N \quad \mathbf{u}_{i,\Gamma}\}$. The new order of the vector of random response vector leads to an "arrow" pattern of every stochastic part stiffness matrix \mathbf{K}_i in Equation (2) as:

$$\mathbf{K}_i = \left[\begin{array}{ccc|c} \mathbf{K}_{i,II}^1 & \cdots & 0 & \mathbf{K}_{i,I\Gamma}^1 (\mathbf{B}^1)^T \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \cdots & \mathbf{K}_{i,II}^{N_i} & \mathbf{K}_{i,I\Gamma}^{N_i} (\mathbf{B}^{N_i})^T \\ \hline \mathbf{B}^1 \mathbf{K}_{i,I\Gamma}^1 & \cdots & \mathbf{B}^{N_i} \mathbf{K}_{i,I\Gamma}^{N_i} & \sum_{s=1}^{N_i} \mathbf{B}^s \mathbf{K}_{i,I\Gamma}^s (\mathbf{B}^s)^T \end{array} \right] \quad (6)$$

where the restriction operators \mathbf{B}^s acts as a scatter or gather operator between global and local components of the deterministic interface solution vectors [31], expressed as $\mathbf{u}_\Gamma(\theta) = (\mathbf{B}^s)^T \mathbf{u}_\Gamma^s(\theta)$ and $\mathbf{u}_I^s(\theta) = \mathbf{B}^s \mathbf{u}_\Gamma(\theta)$.

Rearranging the augmented system Equation (5) based on the order of $\mathbf{u}(\theta)$ will lead to an "arrow" pattern of the coefficient matrix, as given in

$$\left[\begin{array}{ccc|c} \mathbf{K}_{II}^1 & \cdots & 0 & \mathbf{K}_{I\Gamma}^1 (\mathbf{B}_S^1)^T \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \cdots & \mathbf{K}_{II}^{N_i} & \mathbf{K}_{I\Gamma}^{N_i} (\mathbf{B}_S^{N_i})^T \\ \hline \mathbf{B}_S^1 \mathbf{K}_{I\Gamma}^1 & \cdots & \mathbf{B}_S^{N_i} \mathbf{K}_{I\Gamma}^{N_i} & \sum_{i=1}^{N_i} \mathbf{B}_S^i \mathbf{K}_{I\Gamma}^i (\mathbf{B}_S^i)^T \end{array} \right] \left\{ \begin{array}{c} \mathbf{u}_I^1 \\ \vdots \\ \mathbf{u}_I^{N_i} \\ \mathbf{u}_\Gamma \end{array} \right\} = \left\{ \begin{array}{c} \mathbf{f}_I^1 \\ \vdots \\ \mathbf{f}_I^{N_i} \\ \sum_{i=1}^{N_i} \mathbf{B}_S^i \mathbf{f}_\Gamma^i \end{array} \right\}. \quad (7)$$

where the expansion restriction operators \mathbf{B}_S^s is obtained by the restriction operators \mathbf{B}^s as $\mathbf{B}_S^s = \text{blockdiag}(\mathbf{B}^s, \dots, \mathbf{B}^s)$, $\mathbf{u}_I^s = \left\{ \mathbf{u}_{0,I}^s \quad \mathbf{u}_{1,I}^s \quad \cdots \quad \mathbf{u}_{M-1,I}^s \right\}^T$ for $\alpha, \beta \in \{I, \Gamma\}$, and the solution vector is split into $\mathbf{u}_\Gamma = \left\{ \mathbf{u}_{0,\Gamma} \quad \mathbf{u}_{1,\Gamma} \quad \cdots \quad \mathbf{u}_{M-1,\Gamma} \right\}^T$; $\mathbf{K}_{\alpha,\beta}^s$ is a small augmented matrix and can be expressed as a Kronecker product [37], determined by

$$\mathbf{K}_{\alpha\beta}^s = \sum_{i=0}^N \Theta(\mathbf{A}_i, \mathbf{K}_{i,\alpha\beta}^s) \quad (8)$$

where \mathbf{A}_i is a stochastic Galerkin matrix of size $M \times M$, and $\mathbf{A}_i^{j,k} = \langle \xi_i \psi_j \psi_k \rangle$; the operator $\Theta(\mathbf{A}, \mathbf{B})$ denotes

the Kronecker product of matrices \mathbf{A} and \mathbf{B} .

Then, using Gaussian elimination technique, Equation (7) becomes a global extended e-SC system, as given in

$$\bar{\mathbf{K}}\mathbf{u}_\Gamma = \bar{\mathbf{f}} \quad (9)$$

where the e-SC matrix $\bar{\mathbf{K}}$ and the e-SC loads $\bar{\mathbf{f}}$ can be computed by

$$\begin{cases} \bar{\mathbf{K}} = \sum_{S=1}^{N_S} \mathbf{B}_S^T \left(\mathbf{K}_{\Gamma\Gamma}^S - \mathbf{K}_{I\Gamma}^S \left(\mathbf{K}_{II}^S \right)^{-1} \mathbf{K}_{\Gamma I}^S \right) \mathbf{B}_S \\ \bar{\mathbf{f}} = \sum_{S=1}^{N_S} \mathbf{B}_S^T \left(\mathbf{f}_\Gamma^S - \mathbf{K}_{I\Gamma}^S \left(\mathbf{K}_{II}^S \right)^{-1} \mathbf{f}_I^S \right) \end{cases} \quad (10)$$

By solving the e-SC system Equation (9), the global interface solution coefficients \mathbf{u}_Γ are obtained; the local interior solution coefficient \mathbf{u}_I^s on each subdomain can be obtained independently as:

$$\mathbf{K}_{II}^s \mathbf{u}_I^s = (\mathbf{f}_I^s - \mathbf{K}_{I\Gamma}^s \mathbf{u}_\Gamma^s) \quad (11)$$

As given in Equation (9), the key idea of DDM is to transform the original super-large-scale augmented system into a smaller e-SC system so as to reduce the complexity for calculating the stochastic response of structures. However, even if DDM has improved the efficiency of analyzing the stochastic response of structures through the reduction of complexity, since the coefficient matrix is dense and has lost its structure similar to that of the augmented matrix, a suitable preconditioner has been lacking to date. The ultimate result is that the analysis of stochastic response in large-scale structures is not efficient enough.

APPROXIMATE SPARSE EXPANSION SOLVER FOR DDM-BASED SSFE

As mentioned earlier, the reason why the analysis of stochastic response in large-scale structures is not efficient enough is that, compared with the augmented matrix, the e-SC matrix will become quite dense, and the structure easy to establish preconditioner is lost. As a result, there is no suitable preconditioner for the e-SC system till now. If a precondition suitable for the structure of the e-SC system can be established according to the new structure of the e-SC matrix, its solving efficiency will be greatly improved, and then the efficiency of structural random response analysis will be improved. So, in Section 3.1, we first establish the approximate sparse matrix for the augmented matrix at the subdomain level, and then in Section 3.2, we further develop an approximate sparse preconditioner for the e-SC system. Since the new preconditioner is established, the computational efficiency of the e-SC system will be greatly improved; thus, the analysis of structural stochastic response will be more efficient.

Approximate sparse expansion of subdomain-level augmented matrix

In this section, we develop an approximate sparse expansion of the subdomain-level augmented matrix \mathbf{K}_{II}^s in Equation (10). Through such an expansion, the subdomain-level augmented matrix can be approximately expressed as a block diagonal matrix and a sparse matrix. This is because the random stiffness matrix at the subdomain level can be decomposed into a form of multiplication of a deterministic stiffness matrix and an approximately diagonal matrix. Since the expansion is established, the current dilemma of being unable to

establish an effective preconditioner for the e-SC system will be overcome. Additionally, the multiplication steps between the e-SC matrix and an arbitrary vector will also be optimized.

In order to establish the approximate sparse expansion, we first develop an improved mean-based approximation. Considering the original mean-based approximation $\tilde{\mathbf{K}}^{(0)} = \Theta(\mathbf{I}, \mathbf{K}_{0,II}^s)$, where $\Theta(\mathbf{A}, \mathbf{B})$ denotes the Kronecker product of matrices \mathbf{A} and \mathbf{B} , it is easy to find the inverse^[37], because it is a diagonal matrix, given as $(\tilde{\mathbf{K}}^{(0)})^{-1} = \Theta(\mathbf{I}, (\mathbf{K}_{0,II}^s)^{-1})$. However, with the increasing variance of the random field, the effect of this diagonal precondition on improving computational efficiency is getting worse. For this reason, if the subdomain-level augmented matrix is expanded based on the mean-based approximation, as $\mathbf{K}_{II}^s = \Theta(\mathbf{I}, \mathbf{K}_{0,II}^s)\mathbf{R}_{II}^s$, it is only necessary to find an approximate value $\hat{\mathbf{R}}_{II}^s$ for the relation matrix \mathbf{R}_{II}^s of \mathbf{K}_{II}^s in order to obtain a suitable approximation for \mathbf{K}_{II}^s .

In order to establish the approximate value $\hat{\mathbf{R}}_{II}^s$, we first consider the approximate sparsity of the random stiffness matrix at the subdomain level. Based on our findings about the properties of random stiffness matrices recently, the expansion term of the random stiffness matrix of the s-number subdomain \mathbf{K}_i^s ($i = 0, 1, 2, \dots, m$) can be approximated as a product form, obtained as

$$\mathbf{K}_i^s \approx \mathbf{K}_0^s \hat{\mathbf{R}}_i^s \approx \hat{\mathbf{R}}_i^s \mathbf{K}_0^s \tag{12}$$

where $\hat{\mathbf{R}}_i^s$ is a diagonal matrix, an approximate value of the relation matrix $\mathbf{R}_i^s = (\mathbf{K}_0^s)^{-1} \mathbf{K}_i^s$ for \mathbf{K}_i^s . The number of terms \mathbf{K}_i^s , m , depends on the order d of PCE of the stochastic process $E(x, \theta)$ and number N of the KL expansion of the underlying Gaussian field $\omega(x, \theta)$ expressed as $m = (N + d)! / (N!d!)$.

The value of $\hat{\mathbf{R}}_i^s$ is related to the results of KL expansion of random field at the subdomain; that is, the value of this matrix can be calculated through theoretical computation as:

$$\hat{\mathbf{R}}_i^s = f(E_i(x)) \quad x \in \Omega_s \tag{13}$$

This result further indicates that if the matrix \mathbf{K}_i^s is rewritten as a block matrix, as formulated in

$$\mathbf{K}_i^s = \begin{bmatrix} \mathbf{K}_{i,II}^s & \mathbf{K}_{i,I\Gamma}^s \\ \mathbf{K}_{i,\Gamma I}^s & \mathbf{K}_{i,\Gamma\Gamma}^s \end{bmatrix} \tag{14}$$

Then, its sub-matrices can also be expressed in this approximate form. The submatrix $\mathbf{K}_{i,II}^s$ of \mathbf{K}_i^s can be approximately expressed as

$$\mathbf{K}_{i,II}^j \approx \mathbf{K}_{0,II}^j \hat{\mathbf{R}}_{i,II}^j \tag{15}$$

It means that any matrix $\mathbf{K}_{i,II}^j$ can be expressed as the product of the mean term and the approximate diagonal matrix, and the diagonal elements of this approximate diagonal matrix can be approximately obtained through theoretical calculation.

Then, we establish the approximate relation matrix $\tilde{\mathbf{R}}_{II}^s$, considering the small augmented matrix \mathbf{K}_{II}^s according to the Kronecker product form given in Equation (8); if it is left multiplied by a block diagonal matrix $\Theta(\mathbf{I}, (\mathbf{K}_{0,II}^j)^{-1})$, its relation matrix will be established as $\mathbf{R}_{II}^s = \sum_{i=0}^N \Theta(\mathbf{A}_i, \mathbf{R}_{i,II}^s)$. In the relational matrix, each submatrix $\mathbf{R}_{i,II}^s$ will be an approximate diagonal matrix.

The submatrix of the relation matrix $\mathbf{R}_{i,II}^s$ obtained by direct calculation is dense, and a large number of elements are quite close to 0 in it; these elements should be deleted and the relation matrix should be transformed into an approximate sparse structure with only large elements. Therefore, we establish a function $\ell(\mathbf{R})$ that converts a dense relationship matrix into an approximately sparse relationship matrix. In the approximate sparse function $\ell(\mathbf{R})$, firstly, the matrix is rewritten as a set, as:

$$\Upsilon = \{(j, k, \vartheta_{jk}) \mid j, k \in [0, n]\} \tag{16}$$

where in this equation, ϑ_{jk} is the element in the matrix \mathbf{R} .

Then, according to the magnitude of these elements, they are sorted, and a certain amount of the big elements are retained. The preserved elements constitute another set $\tilde{\Upsilon}$, which can be represented as a sparse matrix to obtain the approximate sparse form of the relation matrix $\tilde{\mathbf{R}}$. Using the Frobenius norm to ensure the accuracy of results through the verification errors, that is, to let the Frobenius norm of the errors low enough as $\|\mathbf{R} - \tilde{\mathbf{R}}\| \leq \varepsilon$, where the norm can be calculated by $\|\mathbf{A}\|_F = \sqrt{\text{tr}(\mathbf{A}^T \mathbf{A})}$ and $\text{tr}(\mathbf{A}) = \sum_{i=1}^n [\mathbf{A}]_{i,i}$ is the trace of a matrix.

The above steps represent the implementation process of approximate sparsity, and we define the steps as a function, denoted as

$$\tilde{\mathbf{R}} = \ell(\mathbf{R}) \tag{17}$$

Then, the approximate sparse relationship matrix of the sub-domain augmented matrix will be established through the following steps as given in Algorithm 1 (in the APPENDIX). Now, all the elements where each node is located are defined as a set Ξ_v ; v denotes the number of one element, and let $\Omega(\Xi_v)$ be the total domain of the elements set Ξ_v .

Through Algorithm 1, a series of relation matrices of random stiffness matrices at the subdomain level are obtained as $\tilde{\mathbf{R}}_{i,II}^s$, and the approximate sparse relation matrix of the augmented matrix at this level is assembled as $\tilde{\mathbf{R}}_{II}^s = \sum_{i=0}^{m-1} \Theta(\mathbf{A}_i, \tilde{\mathbf{R}}_{i,II}^s)$. According to the approximate sparse relation matrix, the approximate sparse matrix of the augmented matrix at this level can be obtained as

$$\mathbf{M}^s = \Theta(\mathbf{I}, \mathbf{K}_{0,II}^s) \left(\sum_{i=0}^{m-1} \Theta(\mathbf{A}_i, \tilde{\mathbf{R}}_{i,II}^s) \right) \tag{18}$$

As given in Equation (18), approximate sparse expansion \mathbf{M}^s of \mathbf{K}_{II}^s is a product of a block diagonal matrix and a sparse matrix with a very small condition number; as an improvement over the mean-based approximation, its inverse matrix is easily established, and it is sufficiently approximate to \mathbf{K}_{II}^s under any conditions. Through this approximation, an approximate value of the e-SC matrix can be also derived, and its preconditioner can be also established. At the same time, a new method can be established to multiply the e-SC matrix with any

vector, and using the method, it is possible to avoid performing direct operations with relatively dense e-SC matrices.

Multiplication of e-SC matrix and vector

As given in Equation (9), the cost of solving the e-SC system by PCG depends on: (1) the computational efficiency of the multiplication of the e-SC matrix and vector; and (2) the preconditioner of the system. So, in this section, in order to improve the efficiency of solving e-SC systems, we develop a new algorithm of multiplication of the e-SC matrix and vector based on approximate sparse expansion in Equation (18). The algorithm is efficient, avoids the direct involvement of large matrices in operations, and significantly reduces the cost of stochastic response analysis of Structure.

During each iteration step in PCG, the multiplication of the e-SC matrix with a temporary vector, $\mathbf{q} = \overline{\mathbf{K}}\mathbf{p}$, will be calculated once, so its calculation time is one of the factors determining the efficiency of solving the e-SC system. The direct multiplication of the matrix and vector, however, will face the problem of too much data of the coefficient matrix of the e-SC system in the random response analysis of large structures. Even if it can be saved as a set of subdomain-level e-SC matrices $\overline{\mathbf{K}}^s = \mathbf{K}_{\Gamma\Gamma}^s - \mathbf{K}_{\Gamma I}^s (\mathbf{K}_{\Gamma\Gamma}^s)^{-1} \mathbf{K}_{\Gamma I}^s$ and Boolean matrix \mathbf{B}^s , each subdomain-level e-SC matrix remains dense and large, and such a large storage pressure will make it impossible to calculate the e-SC system.

In this work, the step of directly multiplying the e-SC matrix with the vector is transformed into a series of steps of multiplying the small-scale augmented matrix with the vector, and the coefficient matrix is the solution problem of the small-scale augmented system. By rewriting the algorithm of multiplying these small-scale augmented matrices with vectors and proposing an approximate sparse preconditioned matrix for small-scale augmented systems, the computational efficiency of the multiplication of e-SC matrices and vectors is improved.

Considering the multiplication of an e-SC matrix and a temporary vector $\mathbf{q} = \overline{\mathbf{K}}\mathbf{p}$, in most cases, this multiplication step does not multiply by directly establishing e-SC matrices; instead, the multiplication will be replaced by the following steps:

$$\begin{cases} \mathbf{p}^s = \mathbf{B}_s^s \mathbf{p} & (1) \\ \mathbf{q}^s = \overline{\mathbf{K}}^s \mathbf{p}^s & (2) \\ \mathbf{q} = \sum_{s=1}^{N_s} (\mathbf{B}_s^s)^T \mathbf{q}^s & (3) \end{cases} \quad (19)$$

In the actual calculation process, the e-SC matrix at the subdomain level $\overline{\mathbf{K}}^s$ is replaced by the formula in Equation (10) to avoid directly using the dense e-SC matrix of the subdomain level; that is, step (2) in Equation (19) is replaced by $\mathbf{q} = \sum_{s=1}^{N_s} (\mathbf{B}_s^s)^T \left((\mathbf{K}_{\Gamma\Gamma}^s - \mathbf{K}_{\Gamma I}^s (\mathbf{K}_{\Gamma\Gamma}^s)^{-1} \mathbf{K}_{\Gamma I}^s) \mathbf{B}_s^s \mathbf{p} \right)$. Therefore, it is obvious that the steps of e-SC matrix and vector multiplication are ultimately transformed into a series of small augmented matrix $(\mathbf{K}_{\Gamma\Gamma}^s, \mathbf{K}_{\Gamma I}^s \text{ and } \mathbf{K}_{I\Gamma}^s)$ and vector multiplication, and solution of a series of small augmented systems $\mathbf{K}_{II}^s \mathbf{a} = \mathbf{b}$, where \mathbf{b} is a temporary vector. According to the Kronecker product of the subdomain-level augmented matrix, a set representing all the elements of matrix $\mathbf{A}_i (i = 0, 1, 2, \dots)$ is established as λ derived in

$$\lambda = \{(i, j, k, c_{ijk}) \mid c_{ijk} \neq 0\} \quad (20)$$

Then, the multiplication of argument matrix $\hat{\mathbf{K}} = \sum_{i=0}^N \Theta(\mathbf{A}_i, \mathbf{K}_i)$ and vector \mathbf{p} can be realized by Algorithm

2 (in the APPENDIX).

The solution of a series of small-scale augmented systems in each iteration step of PCG, which are not large in scale but need to be solved many times in the iterative process, will lead to the time-consuming calculation and finally have a great impact on the computational efficiency of the e-SC system. So, in this work, PCG is used to solve the equation, and the matrix-vector multiplication step is conducted by Algorithm 2. Then, the approximate sparse matrix in Equation (18) is used as the preconditioner of the solution.

In recent years, the parallelization of DDM is usually calculations involving the interior of sub-domains. In order to calculate larger-scale structures, the strategy of multi-computer parallelization is often adopted. The computations inside each subdomain usually do not interfere with each other. Therefore, in the process of DDM, this portion of the calculations is typically parallel, and because of this, in this work, the steps of multiplying e-SC matrices and temporary vectors are finally transformed into the following parallel steps of Algorithm 3 (in the APPENDIX).

As given in Algorithm 3, by virtue of the approximate sparse expansion of the subdomain-level augmented matrix in Equation (18), we establish an algorithm of multiplying the e-SC matrix and vector for the PCG solver of e-SC system Equation (9). The algorithm can avoid direct operations on large dense matrices, improve the computation efficiency of each iteration step in PCG, and thus improve the efficiency of stochastic response analysis of structures.

Approximate sparse preconditioner of e-SC system

Once the approximate sparse expansion of the subdomain-level augmented matrix is established, as determined by Equation (18), the approximate sparse preconditioner of e-SC can be established in this section. The preconditioner is expressed in the form of multiplication of a block diagonal matrix with a sparse matrix called the approximate relation matrix. The preconditioner is quite close to the e-SC matrix, which makes the iteration of PCG converge within a relatively small number of iteration steps. Furthermore, because the inverse matrix of the approximation relation matrix can be easily established, the computational complexity of each iteration step is considerably low. These facts make the solving efficiency of the e-SC system extremely high and ultimately substantially improve the efficiency of structural stochastic response analysis.

Approximate sparse expansion of e-SC matrix

When solving e-SC systems, the more important factor determining the computational efficiency of PCG is the choice of preconditioned matrix \mathbf{M} , which is used in the solution of a temporary equation $\mathbf{M}\mathbf{z} = \mathbf{g}$, where \mathbf{g} is a temporary vector calculated in the iterative process. The choice of preconditioned matrix not only affects the calculation time of each iteration step, but also determines the number of iterations, which greatly affects the calculation time of the e-SC system. Therefore, it is required that the preconditioned matrix \mathbf{M} is close enough to the e-SC (that is, the conditional number of $\mathbf{M}^{-1}\bar{\mathbf{K}}$ is low enough), and the time-consuming for solving the temporary equation $\mathbf{M}\mathbf{z} = \mathbf{g}$ should be quite low.

So, in this section, we derive the approximate form of the e-SC matrix based on the approximate sparse augmented matrix in Equation (18). Firstly, based on the derivation process of the approximately sparse augmented matrix, the approximately sparse forms of the matrices in Equation (10) are further established. Then, the approximation values of each matrix in Equation (10) are substituted into the calculation formula of the e-SC matrix, and the approximate form of the e-SC matrix is obtained through computation.

Consider further adjusting the order of solutions in the augmented system to carry out the overall Schur complement (SC) process, expressed as

$$\begin{bmatrix} \mathbf{K}_{II} & \mathbf{K}_{I\Gamma} \\ \mathbf{K}_{\Gamma I} & \mathbf{K}_{\Gamma\Gamma} \end{bmatrix} \begin{Bmatrix} \mathbf{u}_I \\ \mathbf{u}_\Gamma \end{Bmatrix} = \begin{Bmatrix} \mathbf{P}_I \\ \mathbf{P}_\Gamma \end{Bmatrix} \tag{21}$$

Different from Equation (7), in (21), let the solution vector $\mathbf{u}_\alpha = \{ \mathbf{u}_{1,\alpha} \ \mathbf{u}_{2,\alpha} \ \cdots \ \mathbf{u}_{M-1,\alpha} \}$ and the load vector $\mathbf{P}_\alpha = \{ \mathbf{P}_{1,\alpha} \ \mathbf{P}_{2,\alpha} \ \cdots \ \mathbf{P}_{M-1,\alpha} \}$ ($\alpha, \beta \in \{I, \Gamma\}$). Then, for $i = 0, 1, 2, \dots, N$, the submatrix of (21) can be expressed as $\mathbf{K}_{II} = \Theta(\mathbf{A}_i, \mathbf{K}_{i,I})$, $\mathbf{K}_{\Gamma I} = \Theta(\mathbf{A}_i, \mathbf{K}_{i,\Gamma I})$, $\mathbf{K}_{i,\Gamma I} = (\mathbf{K}_{i,\Gamma I})^T$, and $\mathbf{K}_{\Gamma\Gamma} = \Theta(\mathbf{A}_i, \mathbf{K}_{i,\Gamma\Gamma})$, where $\mathbf{K}_{i,II} = \text{blockdiag} \left\{ \mathbf{K}_{i,II}^1 \ \mathbf{K}_{i,II}^2 \ \cdots \ \mathbf{K}_{i,II}^{N_i} \right\}$ is a block diagonal matrix, $\mathbf{K}_{i,\Gamma I} = \left\{ (\mathbf{B}^1)^T \mathbf{K}_{i,\Gamma I}^1 \ (\mathbf{B}^2)^T \mathbf{K}_{i,\Gamma I}^2 \ \cdots \ (\mathbf{B}^{N_s})^T \mathbf{K}_{i,\Gamma I}^{N_s} \right\}$, $\mathbf{K}_{i,\Gamma\Gamma} = \sum_{s=1}^{N_s} (\mathbf{B}^s)^T \mathbf{K}_{i,\Gamma\Gamma}^s \mathbf{B}^s$ is a matrix composed of submatrices and a Boolean matrix.

According to the approximate value of \mathbf{K}_{II}^s in (18), \mathbf{K}_{II} can be approximated as the value of the block diagonal matrix $(\mathbf{M}^1, \mathbf{M}^2, \dots, \mathbf{M}^{N_s})$ after the transformation order, as formulated in

$$\mathbf{K}_{II} \approx \left(\sum_{i=0}^{m-1} \Theta(\mathbf{A}_i, \hat{\mathbf{R}}_{i,II}) \right) \Theta(\mathbf{I}, \mathbf{K}_{0,II}) \tag{22}$$

where $\hat{\mathbf{R}}_{i,II} = \text{blockdiag} \left(\hat{\mathbf{R}}_{i,II}^1, \hat{\mathbf{R}}_{i,II}^2, \dots, \hat{\mathbf{R}}_{i,II}^s \right)$.

For other submatrices $\mathbf{K}_{\Gamma I} = \Theta(\mathbf{A}_i, \mathbf{K}_{i,\Gamma I})$, $\mathbf{K}_{I\Gamma} = \Theta(\mathbf{A}_i, \mathbf{K}_{i,I\Gamma})$, and $\mathbf{K}_{\Gamma\Gamma} = \Theta(\mathbf{A}_i, \mathbf{K}_{i,\Gamma\Gamma})$ in Equation (21), an approximate sparse expansion is established for the overall random stiffness matrix using a similar approach as Equation (12), denoted as

$$\mathbf{K}_i \approx \mathbf{K}_0 \hat{\mathbf{R}}_i \approx \hat{\mathbf{R}}_i \mathbf{K}_0 \tag{23}$$

After dividing the subdomains, we can expand Equation (23) to obtain the block form, expressed as

$$\begin{bmatrix} \mathbf{K}_{i,II} & \mathbf{K}_{i,I\Gamma} \\ \mathbf{K}_{i,\Gamma I} & \mathbf{K}_{i,\Gamma\Gamma} \end{bmatrix} = \begin{bmatrix} \mathbf{K}_{0,II} & \mathbf{K}_{0,I\Gamma} \\ \mathbf{K}_{0,\Gamma I} & \mathbf{K}_{0,\Gamma\Gamma} \end{bmatrix} \begin{bmatrix} \mathbf{R}_{i,II} & \mathbf{R}_{i,I\Gamma} \\ \mathbf{R}_{i,\Gamma I} & \mathbf{R}_{i,\Gamma\Gamma} \end{bmatrix} \approx \begin{bmatrix} \mathbf{K}_{0,II} & \mathbf{K}_{0,I\Gamma} \\ \mathbf{K}_{0,\Gamma I} & \mathbf{K}_{0,\Gamma\Gamma} \end{bmatrix} \begin{bmatrix} \hat{\mathbf{R}}_{i,II} & \hat{\mathbf{R}}_{i,I\Gamma} \\ \hat{\mathbf{R}}_{i,\Gamma I} & \hat{\mathbf{R}}_{i,\Gamma\Gamma} \end{bmatrix} \tag{24}$$

where $\mathbf{R}_{i,II}$, $\mathbf{R}_{i,\Gamma I}$ and $\mathbf{R}_{i,\Gamma\Gamma}$ are the real values of the submatrices of the relation matrix.

According to the diagonal dominance of $\hat{\mathbf{R}}_i$, $\hat{\mathbf{R}}_{i,I\Gamma} = 0$ and $\hat{\mathbf{R}}_{i,\Gamma I} = 0$ in Equation (24) can be known, so each submatrix of \mathbf{K}_i has an approximate value: $\mathbf{K}_{i,\Gamma I} \approx \mathbf{K}_{0,\Gamma I} \hat{\mathbf{R}}_{i,II}$, $\mathbf{K}_{i,I\Gamma} \approx \mathbf{K}_{0,I\Gamma} \mathbf{R}_{i,\Gamma\Gamma}$ and $\mathbf{K}_{i,\Gamma\Gamma} \approx \mathbf{K}_{0,\Gamma\Gamma} \mathbf{R}_{i,\Gamma\Gamma}$; then, the approximate value of the submatrices in Equation (21) can be obtained as

$$\begin{cases} \mathbf{K}_{\Gamma I} \approx \Theta(\mathbf{I}, \mathbf{K}_{0,\Gamma I} (\sum_{i=0}^{m-1} \Theta(\mathbf{A}_i, \hat{\mathbf{R}}_{i,II}))) \\ \mathbf{K}_{I\Gamma} \approx \Theta(\mathbf{I}, \mathbf{K}_{0,I\Gamma} (\sum_{i=0}^{m-1} \Theta(\mathbf{A}_i, \mathbf{R}_{i,\Gamma\Gamma}))) \\ \mathbf{K}_{\Gamma\Gamma} \approx \Theta(\mathbf{I}, \mathbf{K}_{0,\Gamma\Gamma} (\sum_{i=0}^{m-1} \Theta(\mathbf{A}_i, \mathbf{R}_{i,\Gamma\Gamma}))) \end{cases} \tag{25}$$

Then, Equations (25) and (22) are substituted into (21) and the SC process is carried out; with some simplification steps, the approximate form of the e-SC matrix can be finally obtained as

$$\bar{\mathbf{K}} \approx \Theta(\mathbf{I}, \bar{\mathbf{K}}_0) \left(\sum_{i=0}^{m-1} \Theta(\mathbf{A}_i, \mathbf{R}_{i,\Gamma}) \right) \quad (26)$$

where $\bar{\mathbf{K}}_0 = \mathbf{K}_{0,\Gamma} - \mathbf{K}_{0,\Gamma I} (\mathbf{K}_{0,II})^{-1} \mathbf{K}_{0,\Gamma}$ is the mean SC (m-SC) matrix.

According to the approximate simplification result in Equation (26), the e-SC matrix will approximate a special structure, which is a multiplication of two matrices: the one is a diagonal matrix composed of m-SC matrix $\bar{\mathbf{K}}_0 = \mathbf{K}_{0,\Gamma} - \mathbf{K}_{0,\Gamma I} (\mathbf{K}_{0,II})^{-1} \mathbf{K}_{0,\Gamma}$, and another one is a block matrix stacked by several approximate diagonal matrices $\sum_{i=0}^{m-1} \Theta(\mathbf{A}_i, \mathbf{R}_{i,\Gamma})$. Conversely, if the above-mentioned block diagonal matrix consisting of the inverse of the m-SC matrix $\bar{\mathbf{K}}_0^{-1}$ is used to multiply the e-SC matrix left, then a matrix consisting of approximate block diagonal matrices or approximate zero matrix will be obtained, which is called the relationship matrix of the e-SC matrix in this work, and the diagonal element values of these approximate diagonal submatrices are closely related to the results of KL expansion of random field.

The relation matrix of the e-SC matrix can be approximated as a sparse matrix with the original matrix, which can be used to establish the precondition for PCG to solve the e-SC system. However, different from establishing the relational matrix of a subdomain augmented matrix, it is usually not feasible to directly multiply the block diagonal matrix $\Theta(\mathbf{I}, \bar{\mathbf{K}}_0^{-1})$, because the multiplication of $\bar{\mathbf{K}}_0^{-1}$ and any submatrix $\bar{\mathbf{K}}_{jk}$ will be involved in the calculation process, and in large structures, the scale of the matrix is quite large, so direct calculation will waste a lot of time and need to store a large-scale dense matrix.

We utilize the similarity between approximate relation matrix in (26) to infer the relationships between various sub-matrices of the actual relationship matrix. Using this relationship, unnecessary calculations can be avoided, thereby improving computational efficiency.

According to Equation (26), if we calculate the theoretical values of each submatrix of the relational matrix based on Equation (26), we can obtain that $\mathbf{R}_{r,q} \approx \sum_{i=0}^{m-1} c_{irq} \mathbf{R}_{i,\Gamma}$. This means that if a series of approximate values can be calculated as $\tilde{\mathbf{R}}_{i,\Gamma} \approx \mathbf{R}_{i,\Gamma}$; then, the approximate values of all submatrices of the relation matrix can be computed. Therefore, it is only necessary to find a fraction of the values of c , such that $c_{irc} \neq 0$ and $c_{jrc} = 0$ when $j \neq i$; at the same time, establish a collection to store these c , as the eigen set \mathfrak{S} in

$$\mathfrak{S} = \{(i, r, q) \mid c_{irq} \neq 0, c_{jrc} = 0(j \neq i)\} \quad (27)$$

To avoid the direct storage of large dense matrices, we derived an approximate sparse expansion form of subdomain-level e-SC through the conclusion of Section 3.1. Each term of the calculation formula $\bar{\mathbf{K}}^s = \mathbf{K}_{\Gamma}^s - \mathbf{K}_{\Gamma I}^s (\mathbf{K}_{II}^s)^{-1} \mathbf{K}_{\Gamma I}^s$ is expanded into the Kronecker product as

$$\bar{\mathbf{K}}^s = \sum_{i=0}^N \Theta(\mathbf{A}_i, \mathbf{K}_{i,\Gamma}^s) - \sum_{i=0}^N \Theta(\mathbf{A}_i, \mathbf{K}_{i,\Gamma}^s) \left(\sum_{i=0}^N \Theta(\mathbf{A}_i, \mathbf{K}_{i,II}^s) \right)^{-1} \sum_{i=0}^N \Theta(\mathbf{A}_i, \mathbf{K}_{i,\Gamma}^s) \quad (28)$$

According to Equations (12) and (14), the sub-matrix in (28) can be approximately obtained as $\mathbf{K}_{i,\Gamma} \approx \mathbf{K}_{0,\Gamma} \hat{\mathbf{R}}_{i,\Gamma}$, $\mathbf{K}_{i,\Gamma I} \approx \mathbf{K}_{0,\Gamma I} \hat{\mathbf{R}}_{i,\Gamma}$ and $\mathbf{K}_{i,II} \approx \mathbf{K}_{0,\Gamma I} \hat{\mathbf{R}}_{i,II}$; by substituting the above approximate value into

Equation (28), the subdomain-level e-SC matrix $\bar{\mathbf{K}}^s$ will have an approximate value as

$$\bar{\mathbf{K}}^s \approx \Theta \left(\mathbf{I}, \left(\mathbf{K}_{0,\Gamma\Gamma}^s - \mathbf{K}_{0,\Gamma\Gamma}^s \left(\mathbf{K}_{0,II}^s \right)^{-1} \mathbf{K}_{0,\Gamma I}^s \right) \right) \left(\sum_{i=0}^N \Theta \left(\mathbf{A}_i, \hat{\mathbf{C}}_{i,\Gamma\Gamma}^s \right) \right) \tag{29}$$

According to Equation (29), by left-multiplying a diagonal matrix $\Theta(\mathbf{I}, \bar{\mathbf{K}}_0^s)$, where $\bar{\mathbf{K}}_0^s$ denotes the subdomain-level m-SC matrix and $\bar{\mathbf{K}}_0^s = \mathbf{K}_{0,\Gamma\Gamma}^s - \mathbf{K}_{0,\Gamma\Gamma}^s \left(\mathbf{K}_{0,II}^s \right)^{-1} \mathbf{K}_{0,\Gamma I}^s$, then approximating each submatrix with sparsity processing as $\tilde{\mathbf{R}}_{r,q}^s = \ell \left(\left(\bar{\mathbf{K}}_0^s \right)^{-1} \bar{\mathbf{K}}_{r,q}^s \right)$, every submatrix of the approximate value of subregion-level e-SC can be obtained as

$$\bar{\mathbf{K}}_{r,q}^s \approx \bar{\mathbf{K}}_0^s \tilde{\mathbf{R}}_{r,q}^s \tag{30}$$

Thus, according to Equations (26), (27) and (30), any sub-matrix of the relation matrix of an e-SC matrix can be denoted as

$$\begin{cases} \mathbf{R}_i = \left(\bar{\mathbf{K}}_0^s \right)^{-1} \left(\sum_{s=1}^{N_s} \left(\mathbf{B}^s \right)^T \bar{\mathbf{K}}_0^s \tilde{\mathbf{R}}_{r,q}^s \mathbf{B}^s \right) / c_{irq} & (i, r, q) \in \mathfrak{I}, \quad i = 0, 1, 2, \dots \\ \mathbf{R}_{r,q} \approx \sum_{i=0}^m c_{irq} \mathbf{R}_i \end{cases} \tag{31}$$

where the matrix \mathbf{R}_i is called the eigen relation matrix in this work; Equation (31) demonstrates that only few number of eigen relation matrix can express the relation matrix of the e-SC matrix.

Due to the approximate diagonal structure given Equation (31), it can be transformed into an approximately sparse form $\tilde{\mathbf{R}}_{r,q} = \ell \left(\mathbf{R}_{r,q} \right)$, and all submatrices are combined to get an approximate sparse relation matrix $\tilde{\mathbf{R}}$ for the e-SC matrix, which can be further used to obtain the approximate sparse expansion of the e-SC matrix as $\bar{\mathbf{K}} \approx \Theta \left(\mathbf{I}, \bar{\mathbf{K}}_0 \right) \tilde{\mathbf{R}}$.

The matrix $\Theta \left(\mathbf{I}, \bar{\mathbf{K}}_0 \right) \tilde{\mathbf{R}}$ is a good approximation of the e-SC matrix and the equations with this matrix as the coefficient matrix are quite easy to solve, because $\tilde{\mathbf{R}}$ is a matrix having a quite low condition number. Therefore, using this matrix as a preconditioner can greatly improve the efficiency of solving the e-SC equations.

However, another problem is, for the e-SC matrix, when many subdomains are divided, the number of global interface nodes will be quite large, which leads to the large scale of any sub-matrix of the e-SC matrix, along with the m-SC. The fact means that the necessary step of computing the \mathbf{R}_i will result in a considerable computational cost. So, there is a need for a computation process that can avoid directly computing the inverse of the m-SC matrix and the multiplication of large dense matrices.

Approximate relation matrix for e-SC matrix

In order to efficiently calculate the approximate values of the relation matrix \mathbf{R} , we further develop an algorithm based on the two-level domain decomposition. By virtue of the two-level domain decomposition, in this algorithm, the inverse of the m-SC matrix is translated into a form of Cholesky decomposition, meaning that only several small-scale matrices need to be inverse to obtain the inverse of the m-SC matrix. Meanwhile, the

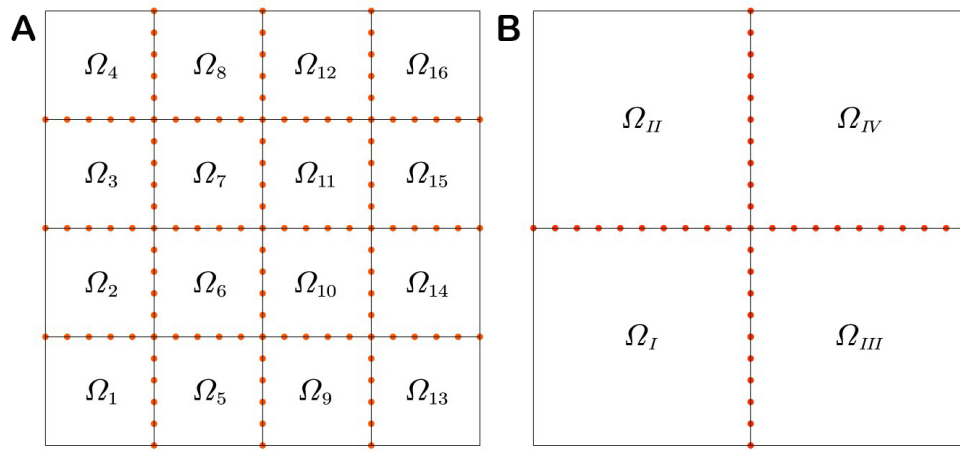


Figure 1. The two-level domain decomposition of a square plate (• (red) denotes the interface nodes, and the interior nodes are omitted in this figure). (A) the first-level domain decomposition; (B) the second-level domain decomposition.

product of the inverse matrix of m-SC and the submatrix of the e-SC matrix is transformed into a series of small-scale matrix multiplications that can be performed efficiently in parallel. Thus, the problem of obtaining the approximate relation matrix in Section 3.2.1 has been solved.

(1) Computation of the inverse of m-SC matrix

The division of two-level subdomains is based on the division of one-level subdomains, and several adjacent first-level subdomains form a second-level subdomain. The order of e-SC matrix solutions is rearranged by two-level division; then, each sub-matrix has an arrow-shaped block structure according to the element distribution law of its coefficient matrix.

On the basis of finite element mesh, the structure is divided into a certain number N_s of subdomains; each subdomain contains a certain number of elements shown in Figure 1A, which is called the first-level domain decomposition. Then, according to the first-level domain decomposition, the structure is divided into fewer subdomains; each subdomain contains a certain number of the first-level subdomains, as shown in Figure 1B.

After the structure is divided into two levels, the order of Schur complementary system solutions is adjusted according to the results of the division, so that its coefficient matrix becomes a block matrix with an arrow structure element distribution in each submatrix. In order to explain this structure more clearly, firstly, the element distribution of the mean interface stiffness matrix $\bar{\mathbf{K}}_0$ after two-level division and adjustment of the order of solutions is considered. The m-SC matrix comes from the SC process of a temporary deterministic problem, denoted as

$$\mathbf{K}_0 \mathbf{v} = \mathbf{f} \quad (32)$$

According to the deterministic domain decomposition, the SC process of Equation (32) will get the following m-SC system, as formulated in

$$\bar{\mathbf{K}}_0 \mathbf{v}_\Gamma = \bar{\mathbf{f}} \quad (33)$$

By the two-level domain decomposition, the m-SC matrix $\bar{\mathbf{K}}_0$ can be expressed by the subdomain-level m-SC matrices and Boolean matrices as $\bar{\mathbf{K}}_0 = \sum_{s=1}^{N_s} (\mathbf{B}^s)^T \bar{\mathbf{K}}_0^s \mathbf{B}^s$. It is obvious that, the matrix is a series of small-scale matrices mapped and added by Boolean matrices; therefore, there are a large number of zero elements in the matrix, and the distribution of zero elements satisfies

$$\bar{\mathbf{K}}_0^{(ij)} \begin{cases} \neq 0 & \text{if } \mathbf{v}_\Gamma^d \text{ and } \mathbf{v}_\Gamma^j \text{ in the same subdomain} \\ = 0 & \text{if } \mathbf{v}_\Gamma^i \text{ and } \mathbf{v}_\Gamma^j \text{ in different subdomains} \end{cases} \quad (34)$$

According to the distribution law of zero elements given in Equation (34) and the division of the second-level domain decomposition shown in Figure 1, we reorder the solutions of the interface \mathbf{v}_Γ and block them as

$$\mathbf{v}_\Gamma = \left\{ \mathbf{v}_\Gamma^1 \quad \mathbf{v}_\Gamma^2 \quad \dots \quad \mathbf{v}_\Gamma^{L_t} \quad \mathbf{v}_X \right\} \quad (35)$$

where \mathbf{v}_Γ^t denotes the second-level interior solution, and \mathbf{v}_X denotes the second-level interface solution.

After rearranging the order of SC matrices according to the new order of interface solution vectors derived in Equation (35), based on the law of element distribution in Equation (34), the matrix will become the following arrow shape expressed as

$$\bar{\mathbf{K}}_0 = \left[\begin{array}{ccc|c} \bar{\mathbf{K}}_{0,II}^{(1)} & \dots & 0 & \bar{\mathbf{K}}_{0,I\Gamma}^{(1)} (\mathbf{B}_L^{(1)})^T \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \dots & \bar{\mathbf{K}}_{0,II}^{(L_t)} & \bar{\mathbf{K}}_{0,I\Gamma}^{(L_t)} (\mathbf{B}_L^{(L_t)})^T \\ \hline \mathbf{B}_L^{(1)} \bar{\mathbf{K}}_{0,\Gamma I}^{(1)} & \dots & \mathbf{B}_L^{(L_t)} \bar{\mathbf{K}}_{0,\Gamma I}^{(L_t)} & \sum_{t=1}^{L_t} \mathbf{B}_L^{(t)} \bar{\mathbf{K}}_{0,\Gamma\Gamma}^{(t)} (\mathbf{B}_L^{(t)})^T \end{array} \right] \quad (36)$$

where $\mathbf{B}_L^{(t)}$ acts as a scatter or gather operator between global and local components of the deterministic second-level interface solution vectors as

$$\mathbf{v}_X^t = \mathbf{B}_L^{(t)} \mathbf{v}_X, \quad \mathbf{v}_X = (\mathbf{B}_L^{(t)})^T \mathbf{v}_X^t \quad (37)$$

The form of the block matrix given in Equation (36) is quite easy to find the inverse. Because it can be easily expanded into a simpler form by block Cholesky decomposition, denoted as

$$\bar{\mathbf{K}}_0 = \begin{bmatrix} \mathbf{I}_I & \mathbf{0} \\ \bar{\mathbf{K}}_{0,\Gamma I} \bar{\mathbf{K}}_{0,II}^{-1} & \mathbf{I}_\Gamma \end{bmatrix} \begin{bmatrix} \bar{\mathbf{K}}_{0,II} & \mathbf{0} \\ \mathbf{0} & \mathbf{S} \end{bmatrix} \begin{bmatrix} \mathbf{I}_I & \bar{\mathbf{K}}_{0,II}^{-1} \bar{\mathbf{K}}_{0,I\Gamma} \\ \mathbf{0} & \mathbf{I}_\Gamma \end{bmatrix} \quad (38)$$

where $\bar{\mathbf{K}}_{0,\Gamma I} = \left[\mathbf{B}_L^{(1)} \bar{\mathbf{K}}_{0,\Gamma I}^{(1)} \quad \mathbf{B}_L^{(2)} \bar{\mathbf{K}}_{0,\Gamma I}^{(2)} \quad \dots \quad \mathbf{B}_L^{(L_t)} \bar{\mathbf{K}}_{0,\Gamma I}^{(L_t)} \right]$, \mathbf{I}_I and \mathbf{I}_Γ denote the identity matrix, $\bar{\mathbf{K}}_{0,II} = \text{blockdiag}(\bar{\mathbf{K}}_{0,II}^{(1)}, \bar{\mathbf{K}}_{0,II}^{(2)}, \dots, \bar{\mathbf{K}}_{0,II}^{(L_t)})$ is a block diagonal matrix, $\bar{\mathbf{K}}_{0,I\Gamma} = \bar{\mathbf{K}}_{0,\Gamma I}^T$, and \mathbf{S} is the second-level SC matrix which can be obtained by

$$\mathbf{S} = \sum_{l=1}^{L_l} \mathbf{B}_L^{(l)} \left(\overline{\mathbf{K}}_{0,\Gamma\Gamma} - \overline{\mathbf{K}}_{0,\Gamma I} \left(\overline{\mathbf{K}}_{0,II} \right)^{-1} \overline{\mathbf{K}}_{0,I\Gamma} \right) \left(\mathbf{B}_L^{(l)} \right)^T \tag{39}$$

It can be easily observed that the inverse of this matrix can be expressed as

$$\overline{\mathbf{K}}_0^{-1} = \begin{bmatrix} \mathbf{I}_I & -\overline{\mathbf{K}}_{0,II}^{-1} \overline{\mathbf{K}}_{0,I\Gamma} \\ \mathbf{0} & \mathbf{I}_\Gamma \end{bmatrix} \begin{bmatrix} \overline{\mathbf{K}}_{0,II}^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{S}^{-1} \end{bmatrix} \begin{bmatrix} \mathbf{I}_I & \mathbf{0} \\ -\overline{\mathbf{K}}_{0,\Gamma I} \overline{\mathbf{K}}_{0,II}^{-1} & \mathbf{I}_\Gamma \end{bmatrix} \tag{40}$$

where only a series of internal stiffness matrices for second-level sub-domains $\overline{\mathbf{K}}_{0,II}$ and the second-level SC matrix \mathbf{S} needs to be inverted. Usually, the second-level SC matrix will be fairly small-scale, which can be directly inverted, because after two-level domain decomposition, there will be quite few second-level interface nodes. Especially for structures with many small sections, such as frame structures and power transmission tower structures, it is quite easy to find a two-level subdomain division scheme with very few second-level interface nodes.

Once the inverse of the m-SC matrix is obtained as Equation (40), the multiplication of the inverse of the m-SC matrix and a vector (or matrix) can be transformed into a more efficient form. Taking multiplication of the matrix and a vector $\mathbf{a} = \overline{\mathbf{K}}_0^{-1} \mathbf{b}$ as an example, the multiplication will be transformed into three steps: (1) the SC process of the right-hand side vector; (2) the solution of the SC system to obtain the interface solution; and (3) substituting the interface solution into the calculation of the internal solution, as determined by

$$\begin{cases} \mathbf{h}_\Gamma = \mathbf{b}_\Gamma - \overline{\mathbf{K}}_{0,\Gamma I} \overline{\mathbf{K}}_{0,II}^{-1} \mathbf{b}_I & (1) \\ \mathbf{a}_\Gamma = \mathbf{S}^{-1} \mathbf{h}_\Gamma & (2) \\ \mathbf{a}_I = \overline{\mathbf{K}}_{0,II}^{-1} (\mathbf{b}_I - \overline{\mathbf{K}}_{0,I\Gamma} \mathbf{a}_\Gamma) & (3) \end{cases} \tag{41}$$

where only multiplication of the small-scale matrix and vector and inverse of the small-scale matrix exist, and clearly, it is more efficient to multiply than computing the inverse matrix and then multiplying it with the vector directly.

(2) *Determination of relation matrix*

Based on our previous research findings, it has been concluded that the e-SC matrix can be represented in a two-level block form, where each sub-block has the same distribution of elements as the determined SC matrix, obtained as

$$\overline{\mathbf{K}} = \begin{bmatrix} \sum_{s=1}^{N_s} \mathbf{B}^s \overline{\mathbf{K}}_{0,0}^s (\mathbf{B}^s)^T & \sum_{s=1}^{N_s} \mathbf{B}^s \overline{\mathbf{K}}_{0,1}^s (\mathbf{B}^s)^T & \cdots & \sum_{s=1}^{N_s} \mathbf{B}^s \overline{\mathbf{K}}_{0,M-1}^s (\mathbf{B}^s)^T \\ \sum_{s=1}^{N_s} \mathbf{B}^s \overline{\mathbf{K}}_{1,0}^s (\mathbf{B}^s)^T & \sum_{s=1}^{N_s} \mathbf{B}^s \overline{\mathbf{K}}_{1,1}^s (\mathbf{B}^s)^T & \cdots & \sum_{s=1}^{N_s} \mathbf{B}^s \overline{\mathbf{K}}_{1,M-1}^s (\mathbf{B}^s)^T \\ \vdots & \vdots & \ddots & \vdots \\ \sum_{s=1}^{N_s} \mathbf{B}^s \overline{\mathbf{K}}_{M-1,0}^s (\mathbf{B}^s)^T & \sum_{s=1}^{N_s} \mathbf{B}^s \overline{\mathbf{K}}_{M-1,1}^s (\mathbf{B}^s)^T & \cdots & \sum_{s=1}^{N_s} \mathbf{B}^s \overline{\mathbf{K}}_{M-1,M-1}^s (\mathbf{B}^s)^T \end{bmatrix} \tag{42}$$

where each submatrix can be transformed into this arrow-shaped representation akin to Equation (36) through the two-level region decomposition, as derived in

$$\overline{\mathbf{K}}_{jk} = \left[\begin{array}{ccc|c} \overline{\mathbf{K}}_{jk,II}^{(1)} & \cdots & 0 & \overline{\mathbf{K}}_{jk,I\Gamma}^{(1)} \left(\mathbf{B}_L^{(1)}\right)^T \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \cdots & \overline{\mathbf{K}}_{jk,II}^{(L_i)} & \overline{\mathbf{K}}_{jk,I\Gamma}^{(L_i)} \left(\mathbf{B}_L^{(L_i)}\right)^T \\ \hline \mathbf{B}_L^{(1)} \overline{\mathbf{K}}_{jk,I\Gamma}^{(1)} & \cdots & \mathbf{B}_L^{(L_i)} \overline{\mathbf{K}}_{jk,I\Gamma}^{(L_i)} & \sum_{i=1}^{(L_i)} \mathbf{B}_L^{(i)} \overline{\mathbf{K}}_{jk,I\Gamma}^{(i)} \left(\mathbf{B}_L^{(i)}\right)^T \end{array} \right] \tag{43}$$

It is noteworthy that substituting matrices of the same number of rows as $\overline{\mathbf{K}}_0$ for \mathbf{a} and \mathbf{b} also yields a valid equation. Therefore, Equation (41) can also be applied to the construction of the relationship matrix $\mathbf{R}_{jk} = \overline{\mathbf{K}}_0^{-1} \overline{\mathbf{K}}_{jk}$. The obtained relation matrix through this approach will be block-structured as

$$\mathbf{R}_{jk} = \left[\begin{array}{cccc} \mathbf{R}_{jk}^{(1,1)} & \mathbf{R}_{jk}^{(1,2)} & \cdots & \mathbf{R}_{jk}^{(1,L_i+1)} \\ \mathbf{R}_{jk}^{(2,1)} & \mathbf{R}_{jk}^{(2,2)} & \cdots & \mathbf{R}_{jk}^{(2,L_i+1)} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{R}_{jk}^{(L_i+1,1)} & \mathbf{R}_{jk}^{(L_i+1,2)} & \cdots & \mathbf{R}_{jk}^{(L_i+1,L_i+1)} \end{array} \right] \tag{44}$$

Directly approximating sparseness of the relationship matrix would consume a considerable amount of time due to the lowest computational complexity of sorting, which is at least $O(n \log n)$. Therefore, it is possible to take advantage of the fact that the sub-matrices of the relationship matrix can be calculated independently and perform two rounds of sparse approximation. For any column of the relationship matrix, according to the above steps, the interface solution is obtained through steps (1) and (2) in Equation (41) first, and then this interface solution is used to determine the interior solution of the subdomain by step (3) in Equation (41).

Since the inverse of second-level SC matrix \mathbf{S} will be used repeatedly in calculations, it is advisable to compute and save the inverse \mathbf{S}^{-1} at the beginning of the analysis. Firstly, considering any column t in the first L_t columns, through the calculation of steps (1) and (2) in Equation (41), the value of the interface solution can be obtained as

$$\mathbf{R}_{jk}^{(L_i+1,t)} = \mathbf{S}^{-1} \left(\mathbf{B}_L^{(t)} \overline{\mathbf{K}}_{jk,I\Gamma}^{(t)} - \overline{\mathbf{K}}_{0,I\Gamma}^{(t)} \left(\mathbf{B}_L^{(t)}\right)^T \left(\overline{\mathbf{K}}_{0,II}^{(t)}\right)^{-1} \overline{\mathbf{K}}_{jk,II}^{(t)} \right) \tag{45}$$

where a series of matrix multiplication problems is involved, including the multiplication of Boolean matrix and other matrices. In order to improve the calculation efficiency of this step, we change the original calculation order, Using the properties of Boolean matrix, $\left(\mathbf{B}_L^{(t)}\right)^T \left(\overline{\mathbf{K}}_{0,II}^{(t)}\right)^{-1}$ is calculated first to reduce the scale of the matrix, then other steps are calculated, so that each operation is a multiplication between small-scale matrices, then the calculation time is reduced. By substituting the calculation result of Equation (45) into step (3) of (41), the rest solutions can be obtained as

$$\mathbf{R}_{jk}^{(o,t)} = \left(\overline{\mathbf{K}}_{0,II}^{(t)}\right)^{-1} \left(\delta_o^t \overline{\mathbf{K}}_{jk,II}^{(t)} - \overline{\mathbf{K}}_{0,I\Gamma}^{(L_i)} \left(\mathbf{B}_L^{(L_i)}\right)^T \mathbf{R}_{jk}^{(L_i+1,t)} \right) \tag{46}$$

Similar to formula a, there is also a Boolean matrix in the calculation of Equation (46), so although what needs to be calculated is the multiplication of $\overline{\mathbf{K}}_{0,\Gamma}^{(L_i)} (\mathbf{B}_L^{(L_i)})^T$ and $\mathbf{R}_{jk}^{(L_i+1,t)}$, in practical calculation, in order to reduce the calculation cost, we choose to calculate $(\mathbf{B}_L^{(L_i)})^T \mathbf{R}_{jk}^{(L_i+1,t)}$ first, and then multiply $\overline{\mathbf{K}}_{0,\Gamma}^{(L_i)}$ left to get the final result. For every $o = 1, 2, \dots, L_t$, $\mathbf{R}_{jk}^{(o,t)}$ can be calculated independently, so in order to reduce the storage cost as much as possible, when each $\mathbf{R}_{jk}^{(o,t)}$ is calculated, it is approximately sparse before calculating the next item $\mathbf{R}_{jk}^{(o+1,t)}$.

For the last column of the relationship matrix in Equation (44), the interface solution $\mathbf{R}_{jk}^{(L_i+1,L_i+1)}$ can be obtained by adopting the same calculation method as

$$\mathbf{R}_{jk}^{(L_i+1,L_i+1)} = \mathbf{S}^{-1} \left(\sum_{s=1}^{(L_i)} \left(\mathbf{B}_L^{(s)} \overline{\mathbf{K}}_{jk,\Gamma}^{(s)} (\mathbf{B}_L^{(s)})^T - \mathbf{K}_{0,\Gamma}^s (\mathbf{K}_{0,II}^s)^{-1} \overline{\mathbf{K}}_{jk,\Gamma}^{(s)} (\mathbf{B}_L^{(s)})^T \right) \right) \quad (47)$$

Bring the calculation results into the sub-formula (3) of Equation (41), the rest of the solution can be obtained as

$$\mathbf{R}_{jk}^{(s,L_i+1)} = \left(\overline{\mathbf{K}}_{0,II}^{(s)} \right)^{-1} \left(\overline{\mathbf{K}}_{jk,\Gamma}^{(s)} (\mathbf{B}_L^{(s)})^T - \overline{\mathbf{K}}_{0,\Gamma}^{(L_i)} (\mathbf{B}_L^{(L_i)})^T \mathbf{R}_{jk}^{(L_i+1,L_i+1)} \right) \quad (48)$$

It can be observed that in Equation (47), the same contents $\mathbf{K}_{0,\Gamma}^s (\mathbf{K}_{0,II}^s)^{-1}$ will be calculated every time the submatrix of the relationship matrix is established, so these repeated calculations can be calculated in advance to avoid unnecessary repetition.

During the calculations, according to the specified order of computation, Equations (48) and (46) can be written in an approximate sparse form first as $\tilde{\mathbf{R}}_{jk}^{(s,L_i+1)} = \ell \left(\mathbf{R}_{jk}^{(s,L_i+1)} \right)$ and $\tilde{\mathbf{R}}_{jk}^{(o,t)} = \ell \left(\mathbf{R}_{jk}^{(o,t)} \right)$; then, Equations (47) and (45) are expressed as $\tilde{\mathbf{R}}_{jk}^{(L_i+1,L_i+1)} = \ell \left(\mathbf{R}_{jk}^{(L_i+1,L_i+1)} \right)$ and $\tilde{\mathbf{R}}_{jk}^{(L_i+1,t)} = \ell \left(\mathbf{R}_{jk}^{(L_i+1,t)} \right)$. According to the above calculation results, the relation matrix can be roughly approximated as $\tilde{\mathbf{R}}'_{jk}$, as formulated in

$$\tilde{\mathbf{R}}'_{jk} = \begin{bmatrix} \tilde{\mathbf{R}}_{jk}^{(1,1)} & \tilde{\mathbf{R}}_{jk}^{(1,2)} & \dots & \tilde{\mathbf{R}}_{jk}^{(1,L_i+1)} \\ \tilde{\mathbf{R}}_{jk}^{(2,1)} & \tilde{\mathbf{R}}_{jk}^{(2,2)} & \dots & \tilde{\mathbf{R}}_{jk}^{(2,L_i+1)} \\ \vdots & \vdots & \ddots & \vdots \\ \tilde{\mathbf{R}}_{jk}^{(L_i+1,1)} & \tilde{\mathbf{R}}_{jk}^{(L_i+1,2)} & \dots & \tilde{\mathbf{R}}_{jk}^{(L_i+1,L_i+1)} \end{bmatrix} \quad (49)$$

Further approximate sparsity processing is performed on the matrix $\tilde{\mathbf{R}}'_{jk}$ to obtain the final approximate sparse relation matrix $\tilde{\mathbf{R}}_{jk} = \ell \left(\mathbf{R}'_{jk} \right)$. Assembling these submatrices results, the approximate sparse relation matrix $\tilde{\mathbf{R}}$ for e-SC.

In the calculation process, all the computations involved are operations between small matrices and the approximate sparse processing of small matrices. Since the relation matrix of each submatrix is established by calculating the same amount of content separately, in the actual calculation, this part will be calculated in

parallel, and the algorithm for finally calculating the relationship matrix of an e-SC matrix is expressed as Algorithm 4 (in the APPENDIX), where we first choose the row and column numbers of the eigen relation matrices by step 5; then, for every eigen relation matrix, which is split into a block matrix akin to Equation (44), the submatrices in the bottom row are first obtained by steps 6 and 8; using the result, the other row can be calculated by steps 9 and 11; steps 12–16 involve creating the approximate value of these submatrices, then use these submatrices to assemble the approximate eigen relation matrix by steps 17 and 18, finally create the approximate relation matrix by step 20.

Once the approximate relation matrix $\tilde{\mathbf{R}}$ of the e-SC matrix is established by Algorithm 4, it is possible to obtain the preconditioner for solving the e-SC system, expressed as

$$\mathbf{M} = \Theta \left(\mathbf{I}, \overline{\mathbf{K}}_0 \right) \tilde{\mathbf{R}} \quad (50)$$

which denotes the proposed approximate sparse preconditioner that is quite close to the e-SC matrix; therefore, using this matrix as the preconditioner for PCG will lead to convergence within a relatively small number of iterations. On the other hand, the matrix is a product of a diagonal matrix and a sparse matrix with very little condition number, which makes the solution of the preconditioning equation relatively easy. Ultimately, adopting this preconditioner can make PCG more efficient in solving the e-SC system, which greatly improves the solution efficiency of the random response of the structure.

Approximate sparse approach for DDM-based SSFE

According to the derivation of Sections 3.1 and 3.2, we develop an approximate sparse approach for DDM-based SSFE; the proposed method targets the step of solving e-SC systems with PCG. An efficient algorithm for the multiplication of an e-SC matrix and an arbitrary vector, and an approximate sparse preconditioner are established to improve the computing efficiency of PCG, thus making the structural random response more efficient. The following steps show the new computational framework based on the approximate sparse approach for DDM-based SSFE in this work:

step 1 Describe the stochastic process $E(x, \theta)$ through KL expansion (or PCE, depending on the process is Gaussian or non-Gaussian). Calculate the matrix \mathbf{A}_i according to c_{ijk} .

step 2 Mesh the structure and divide two-level subdomains, establish the first-level Boolean matrix \mathbf{B}^s and second-level Boolean matrix \mathbf{B}_L^s .

step 3 For $s = 1, 2, 3, \dots, N_s$ and $i = 0, 1, 2, \dots, N$, establish the submatrix of subdomain-level stochastic stiffness matrix \mathbf{K}_i^s , including $\mathbf{K}_{i,II}^s, \mathbf{K}_{i,I\Gamma}^s, \mathbf{K}_{i,\Gamma I}^s, \mathbf{K}_{i,\Gamma\Gamma}^s$.

step 4 Calculate approximate relation matrix $\tilde{\mathbf{R}}_{i,II}^s$ of $\mathbf{K}_{i,II}^s$ according to Algorithm 1

step 5 Calculate Second-level m-SC matrix \mathbf{S} and inverse matrix \mathbf{S}^{-1} by Equation (39).

step 6 Establish the eigen set \mathfrak{J} by Equation (27).

step 7 Calculate the approximate relation matrix $\tilde{\mathbf{R}}$ for the e-SC matrix $\overline{\mathbf{K}}$ by Algorithm 4.

step 8 Parallely calculate the interface solution \mathbf{u}_Γ according to the proposed solution for the e-SC system Algorithm 5 (in the APPENDIX).

step 9 Obtain the local interior solution \mathbf{u}_I^s by Equation (11) parallelly.

In the established algorithm, PCG in step 8 is used to solve the e-SC system; different from the traditional solution, we propose a new calculation scheme based on the approximate sparse stiffness matrix for the two most critical steps in PCG: the e-SC matrix and temporary vector multiplication and the solution of preconditioned equations, to solve the problems that the existing solution methods must save large dense matrices and their calculation efficiency are low.

In step 4, according to the derivation in Section 3.1, the approximate relation matrix of expansion term of the random stiffness matrix in subdomain $\tilde{\mathbf{R}}_{i,II}^s$ is established. Then, the multiplication of the e-SC matrix and temporary vector (steps 2 and 7 in Algorithm 5) can adopt the steps of Algorithm 3 for parallel calculation. In Algorithm 3, the direct multiplication of e-SC and vectors at the subdomain level will no longer be used, but an algorithm based on the calculation formulated in Equation (10) will be established instead. The subdomain augmented matrix and vector multiplication in Algorithm 3 is further transformed into the multiplication given in Algorithm 1; then, the preconditioner for solving any subdomain-level augmented system $\mathbf{K}_{II}^s \mathbf{a} = \mathbf{b}$ by PCG is built as $\mathbf{M}^s = \Theta(\mathbf{I}, \mathbf{K}_{0,II}^s) \tilde{\mathbf{R}}_{i,II}^s$ through the approximate relation matrix $\tilde{\mathbf{R}}_{i,II}^s$ of $\mathbf{K}_{i,II}^s$; this preconditioned matrix is quite close to the subdomain augmented matrix \mathbf{K}_{II}^s , and because it is composed of a block diagonal matrix and a sparse matrix with low condition number, the preconditioned equation $\mathbf{M}^s \mathbf{a} = \mathbf{b}$ is quite easy to solve. By this means, compared with the traditional methods, the proposed new method of multiplying e-SC matrices and temporary vectors, $\mathbf{b} = \bar{\mathbf{K}}\mathbf{a}$, does not need to save large dense matrices and has quite high computational efficiency.

On the basis of Section 3.1, after further derivation, the proximate relationship matrix of e-SC matrix $\tilde{\mathbf{R}}$ is established, and steps 5-7 show the specific calculation process. By creating the eigen set \mathfrak{J} , the number of needed multiplication steps between the inverse of the m-SC matrix and submatrix of an e-SC matrix will be reduced, then by the two-level domain decomposition given in Algorithm 4, the calculation process of every submatrix of relation matrix is transformed into a series of operations between small matrices, which greatly improves the calculation efficiency while avoiding the direct participation of large dense matrices in the operation. After the calculation, the preconditioner for solving the e-SC is obtained as $\mathbf{M} = \Theta(\mathbf{I}, \bar{\mathbf{K}}_0) \tilde{\mathbf{R}}$; this matrix ensures the similarity with the original e-SC matrix $\bar{\mathbf{K}}$, and the high calculational efficiency of solving the preconditioned equation on steps 3 and 12 of Algorithm 5, because it is composed of block diagonal matrix and a sparse matrix with low condition number.

To sum up, by rewriting the step of the e-SC matrix and the vector multiplication based on the approximate sparse random stiffness matrix, we avoid the large dense matrix directly participating in the operation and improve the calculation efficiency. At the same time, based on the approximate sparse random stiffness matrix, the relationship matrix of the e-SC matrix is further established to create a preconditioning matrix. Using this matrix as the preconditioner for solving the e-SC matrix by PCG can greatly improve the calculation efficiency by controlling the iteration steps. Finally, an efficient framework for structural random response analysis is established.

NUMERICAL EXAMPLES

To assess the computational efficiency of the proposed approximate sparse expansion-based domain decomposition solver for SFEM, we consider two numerical examples. In the two examples, the precision of the results (structural stochastic response) and the distribution of elements in the relation matrix of the e-SC matrix will be verified first, and then the convergence rate, computational time needed, and relative speedup compared with the original DDM-based SSFE will be investigated and manifested. The computational platform used for

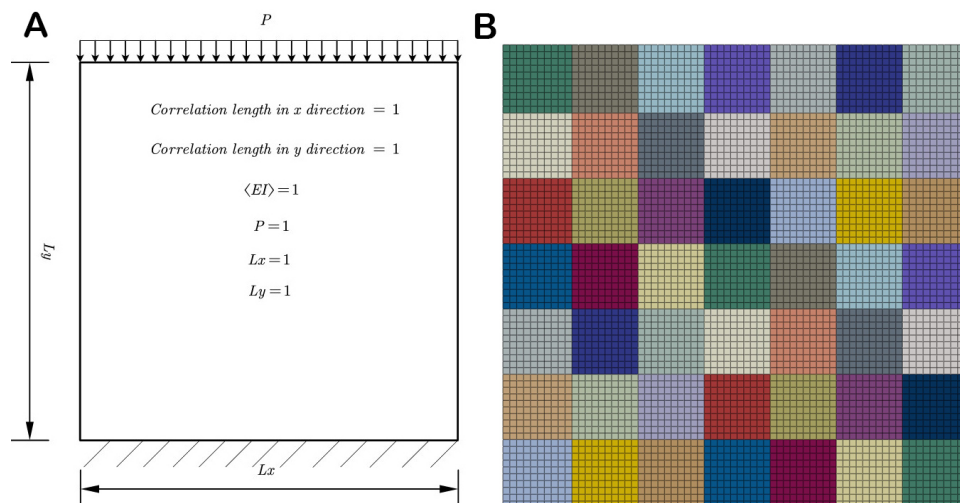


Figure 2. Problem setting(A) and typical finite element mesh with 49 subdomains (B).

this investigation is an Intel Core i9-13900 with 24 physical cores at 3.0 GHz, coupled with 128 GB of RAM. The finite element mesh of the two examples is generated using Abaqus; the finite element method (FEM) assembly procedure is adapted from the FEON framework in python.

Stochastic response of a thin square plate

As a test case, we have considered a stochastic response analysis of a thin square plate. The dimensions of this plate are $L_x = L_y = 1$; thickness is 0.1, and Poisson’s ratio is 0.3. One of its edges is a fixed end, receiving uniformly distributed load with a size of $p = 1$, as shown in Figure 2A. The elastic modulus of the thin square plate E is modeled as a Gaussian field $E(x, \theta)$ with a mean value of $\mu = 1.0$; the exponential covariance function of the field is expressed as

$$C(\mathbf{x}, \mathbf{y}) = \sigma^2 \exp\left(-\frac{|x_1 - x_2|}{b_x} - \frac{|y_1 - y_2|}{b_y}\right) \tag{51}$$

where standard deviation is $\sigma = 0.2$; the correlation lengths of x and y directions are $b_x = b_y = 1$.

The physical domain with unit-square geometry is discretized using an unstructured finite element mesh. After partitioning the mesh, the number of nodes is 5,041, and the count of Quadrilateral elements totals 4,900; the number of subdomains involves 16, 25, 36, 49, and 64; the mesh and the division of subdomains are shown in Figure 2B illustrating the situation of 49 subdomains.

The random field $E(x, \theta)$ is expressed by the KL expansion; in this example, the terms of expansion are $N = 2$, as $E(x, \theta) = E_0 + \sum_{i=1}^N E_i \xi_i$; the result is shown in the following Figure 3.

The response is expressed using sixth order PCE ($M = 28$) leading to a linear system of order 282296. In this work, the traditional DDM, the DDM based on approximate sparse expansion proposed (the PCGM is considered to have converged when the error satisfies: $\|\bar{\mathbf{K}}\mathbf{u} - \bar{\mathbf{f}}\|/\|\bar{\mathbf{f}}\| \leq 10^{-6}$) and 50,000 times of Monte Carlo method are used to calculate the random response of structures. Firstly, the result of the stochastic response calculated through the method proposed is shown in Figure 4.

In order to verify the accuracy of the calculated results more clearly, we consider the node No. 2485 (point

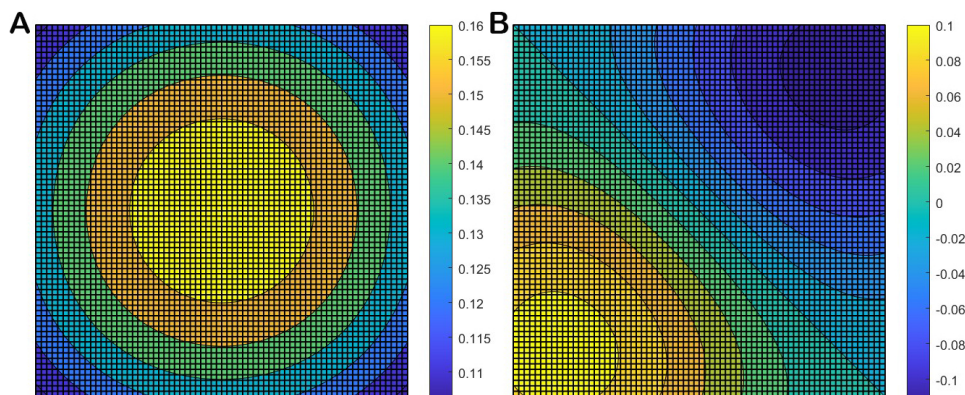


Figure 3. KL expansion result of the random field $E(x, \theta)$: Coefficients before the standard Gaussian random variables ξ_i . (A) E_1 ; (B) E_2 .

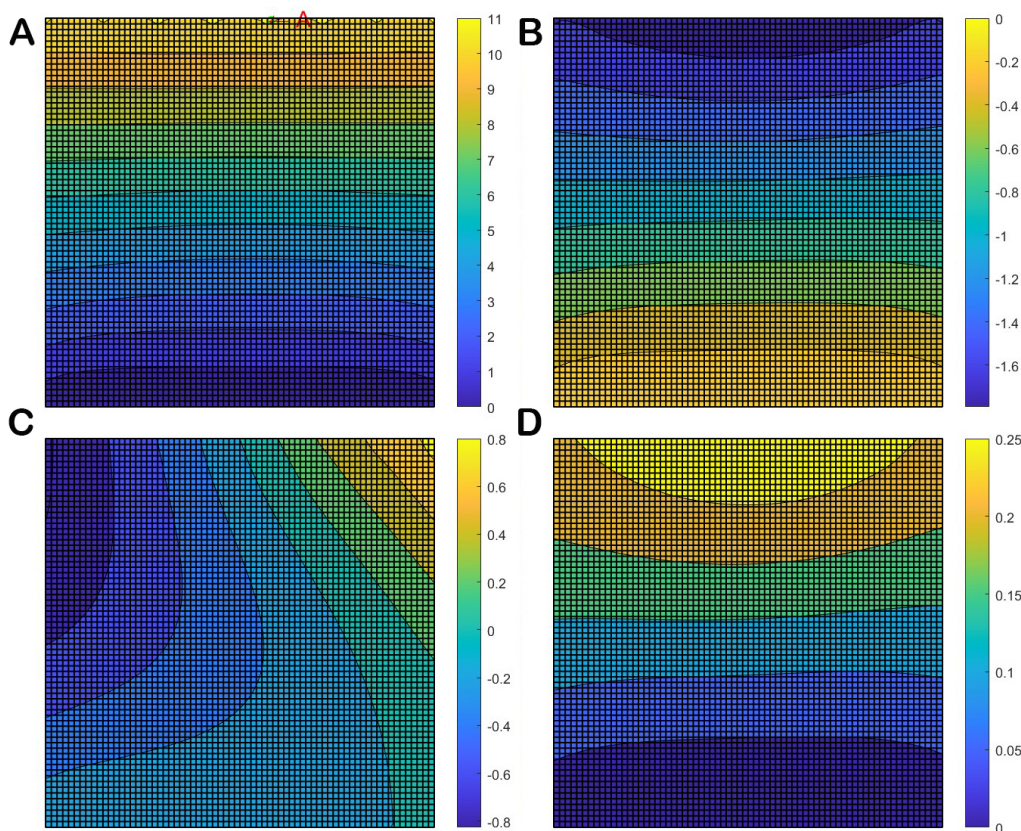


Figure 4. The solution process: Coefficients before the first 4 PC basis. (A) u_0 ; (B) u_1 ; (C) u_0 ; (D) u_1 .

A in Figure 4A), which has a large average response, and draw the probability density function (PDF) of its displacement in the y direction according to the calculation results of different methods shown as Figure 5.

From the PDF shown in the figure, it can be seen that the random response results in the y-direction of node number 2485 obtained by the proposed method are almost consistent with the traditional SSFE method. Moreover, due to the maximum mean displacement in the y-direction of this node, it can be considered that the random response results of other nodes obtained by the proposed method also meet the accuracy requirements. Therefore, it indicates that the proposed approach has the same accuracy as the traditional SSFE method and

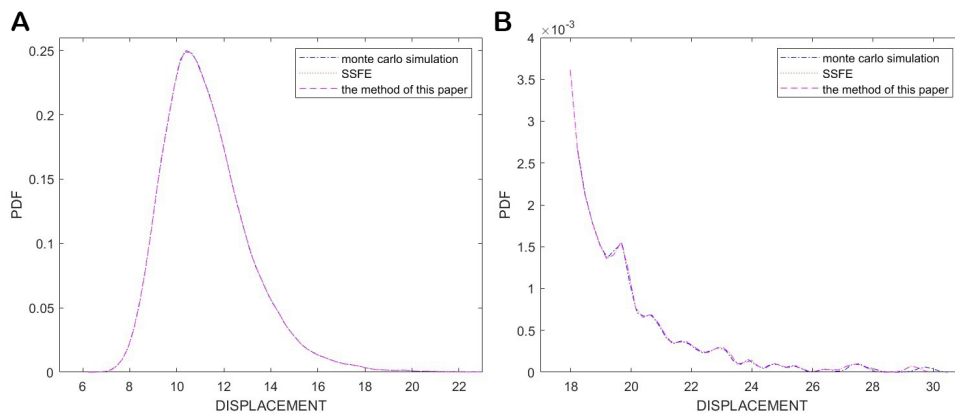


Figure 5. Stochastic response of y-direction displacement of node No. 2485: (A) the PDF and (B) tail of the PDF.

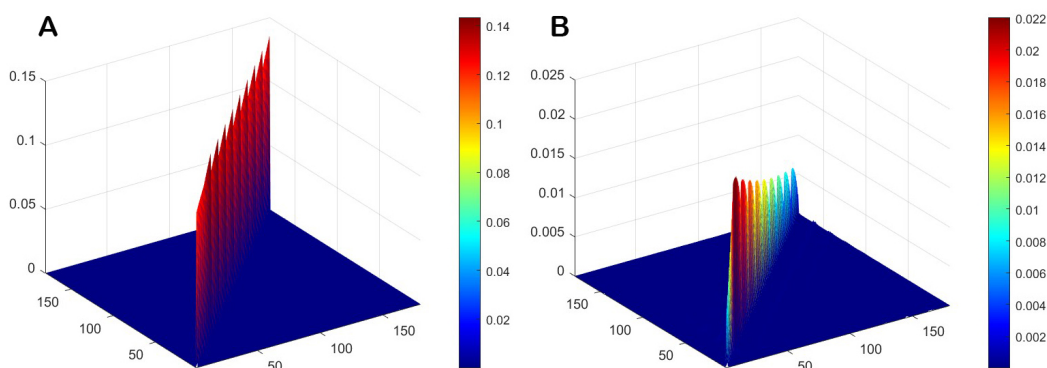


Figure 6. Element distribution of relation matrix of random partial stiffness matrix at the subdomain level: (A) $\mathbf{R}_{1,II}^1$ and (B) $\mathbf{R}_{2,II}^1$.

can meet the accuracy requirements when the response PCE order is sufficient.

Next, the distribution law of the coefficient matrix elements of the e-SC system is verified. Firstly, the element distribution of the augmented matrix at the subdomain level is verified; the relation matrix $\mathbf{R}_{i,II}^s = (\mathbf{K}_{0,II}^s)^{-1} \mathbf{K}_{i,II}^s$ is obtained by left multiplying the inverse matrix of the average stiffness matrix of the subdomain level; then, Figure 6 shows the distribution of elements of $\mathbf{R}_{1,II}^1$ and $\mathbf{R}_{2,II}^1$; in the figures, the x-axis and y-axis represent the row number and column number of the matrix elements, while the z-axis represents the absolute value of the matrix element.

Figure 6 presents that for the relation matrices of the subdomain-level stochastic stiffness matrices, the non-diagonal elements are almost zero, while the values of the diagonal elements are relatively large compared to the non-diagonal elements, which confirms our findings.

Based on the above verification results, we further verify whether the properties of the relationship matrix of the e-SC matrix are consistent with the conclusion of proof in Section 3.2. Firstly, it is necessary to verify the distribution of the elements in the relation matrix $\mathbf{R} = (\Theta(\mathbf{I}, \bar{\mathbf{K}}_0))^{-1} \bar{\mathbf{K}}$. As shown in Figure 7, we choose some of the submatrices of the relation matrix, and the distribution of their elements was charted. Similar to Figure 6, in Figure 7, the x-axis and y-axis represent the row number and column number of the matrix elements, while the z-axis represents the absolute value of the matrix element.

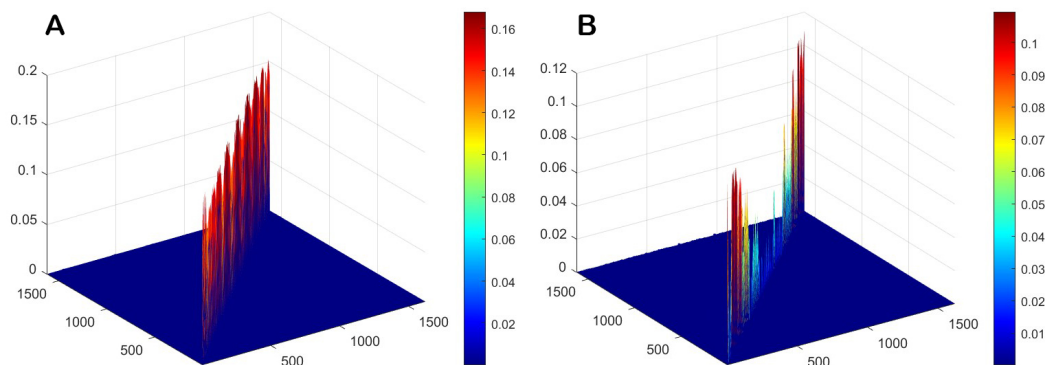


Figure 7. Element distribution of some relation matrix’s submatrices: (A) the first row, second column $R_{1,2}$ and (B) the first row, third column $R_{1,3}$.

Table 1. Statistics of computational time: DDM-based SSFE and approximate sparse expansion approach

Number of subdomains	Time of SC process	Time of solving e-SC matrix	
		Proposed preconditioner	Traditional method
16	1,202.1	258.7	1,532.7
25	400.3	342.7	2,819.3
36	318.1	536.7	5,201.9
49	338.5	744.3	7,254.2
64	542.4	1,120.6	11,533.4

It is evident that the sub-matrices of the relation matrix of the e-SC matrix also exhibit similar patterns of element distribution; that is, the non-diagonal elements are almost zero, while the values of the diagonal elements are relatively large compared to the non-diagonal elements.

Therefore, it can be said that, as demonstrated in this article, the distribution pattern of the elements in the relation matrix has been confirmed. Thus, the approximate sparse preconditioner established in Section 3.2 is approximately equal to the e-SC matrix, thus greatly improving the efficiency of solving the e-SC system.

On this basis, we study the magnitude of improvement in computational efficiency provided by the method proposed in our research. We compared the computational time of the traditional DDM-based SSFE and the proposed method, including the time of the SC process, the direct CG solver and the PCG solver with the proposed preconditioner of the e-SC system, as shown in Table 1.

As demonstrated in Table 1, it can be obviously seen that compared with the traditional DDM-based SSFE, which uses the direct CG solver to solve the e-SC system, PCG with the proposed approximate sparse preconditioner is highly efficient, and as a result, the efficiency of stochastic response analysis of structures is greatly improved. Especially when there are a large number of subdomains and the solving time of the e-SC system becomes a higher proportion, this improvement in computational efficiency will become more evident from the fact shown in Table 1.

Stochastic response of a frame structure

The second example features the stochastic response analysis of a frame structure, where we pay greater attention to its computational efficiency under different circumstances, including varying standard deviations and correlation lengths, than in Section 4.1.

As shown in the following Figure 8, the structure we consider in this example is a five-layer frame structure with a layer height of 3 m, the distance between each column is 2 m, the cross-sectional widths of the beams

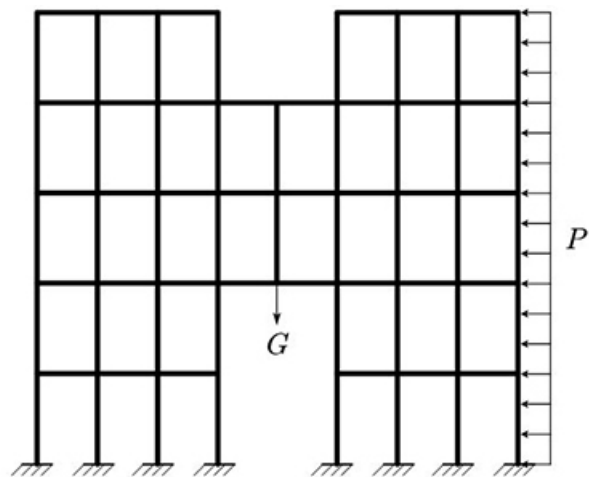


Figure 8. The geometry and subjected load of the frame structure.

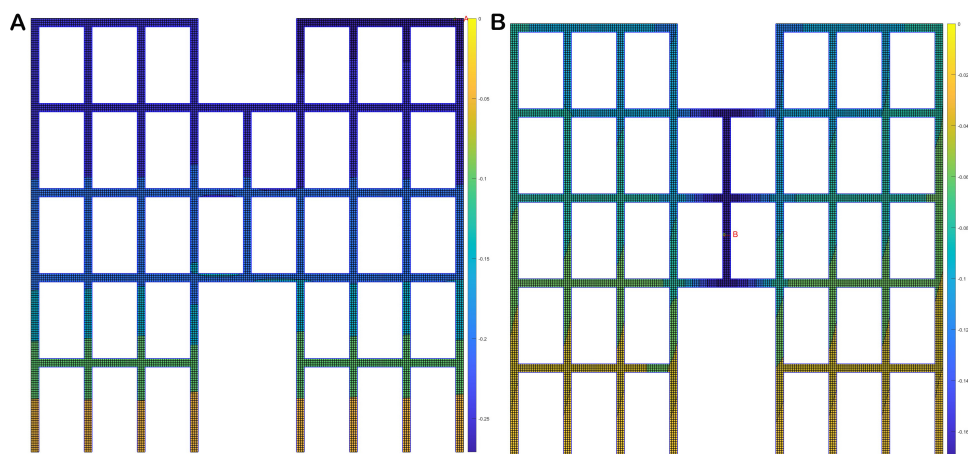


Figure 9. The coefficient before the first PC basis of solution process: (A) Component in the x-direction (B) Component in the y-direction.

and columns are 0.3 m, thickness is 0.1, and Poisson’s ratio is 0.25. The structure is subjected to the self-weight G with planar density $\rho = 7.85 \times 10^3 \text{ Kg/m}^2$ and a lateral uniformly distributed load P with $P = 700 \text{ N/m}$. The elastic modulus of the material in this frame structure is considered as a gaussian random field with a mean of $\bar{E} = 2.06 \times 10^5 \text{ Mpa}$ and a standard deviation of $\sigma = 0.2\bar{E}$, the correlation length of the field is $b_x = b_y = 15.3 \text{ m}$, and the exponential covariance function is the same as given in Equation (51). The finite element mesh of the structure consists of 9,552 elements and 11,855 nodes, and 65 subdomains are generated after subdivision.

The KL expansion is used to express the random field; in this example, the number of terms N used in the truncated KL is 2, the maximum polynomial order of the PCE is 6, the total terms of the PC basis of the solution are $M = 28$, and the augmented system to be solved will have a coefficient matrix of $663,880 \times 663,880$. Using our approximate sparse expansion-based domain decomposition solver, we get the solution of the augmented system, and Figure 9 shows the coefficients before the first PC basis of the solution process of the x-direction and y-direction.

Then, we use PDF of the y-direction displacement at point A and the x-direction displacement at point B to

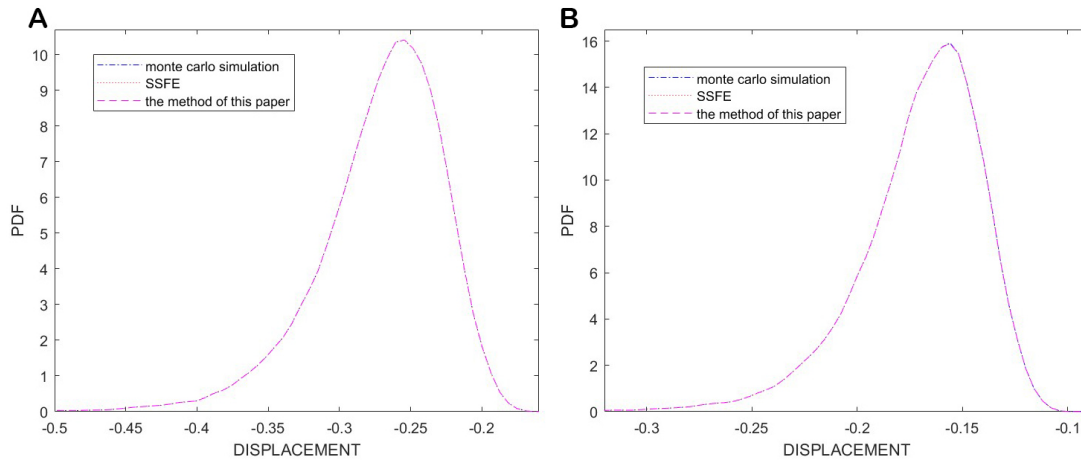


Figure 10. The PDF of (A) x-direction displacement of point A and (B) y-direction displacement of point B.

Table 2. Statistics of computational time of the different standard deviation: DDM-based SSFE and approximate sparse expansion approach

Standard deviation	Time of SC process	Time of solving e-SC matrix	
		Proposed preconditioner	Traditional method
$0.1\bar{E}$	946.4	302.1	15,893.6
$0.15\bar{E}$	953.1	539.4	18,106.9
$0.2\bar{E}$	935.4	619.8	21,011.6
$0.25\bar{E}$	940.2	703.5	21,933.9

Table 3. Statistics of computational time of the different correlation lengths: DDM-based SSFE and approximate sparse expansion approach

Correlation length	Time of SC process	Time of solving e-SC matrix	
		Proposed preconditioner	Traditional method
$0.2b$	928.6	496.4	16,555.3
$0.4b$	936.3	466.1	14,879.8
$0.6b$	938.2	562.9	19,194.6
$0.8b$	936.4	584.5	19,527.4
b	935.4	619.8	21,011.6

describe the calculated results (points A and B are shown in Figure 9; A is the point with the largest average displacement in the y-direction shown in Figure 9A, while B is the point with the largest average displacement in the x-direction shown in Figure 9B) and compare them with the results of 50000 Monte-Carlo simulations, as shown in Figure 10.

It is obvious that the solution process calculated by the proposed approach is similar to the MC method in Figure 10, indicating that the proposed one can meet the accuracy requirements when the PCE order of the response is sufficient. Then, we are concerned about the numerical scalability of the improvement of computational efficiency, that is, the computational efficiency improvement of the proposed method under different conditions (the PCGM is considered to have converged when the error satisfies: $(\|\bar{\mathbf{K}}\mathbf{u} - \bar{\mathbf{f}}\|/\|\bar{\mathbf{f}}\| \leq 10^{-8})$). So, we first consider the different standard deviation σ of the elastic modulus E , taking values of $0.1\bar{E}$, $0.15\bar{E}$, $0.2\bar{E}$ and $0.25\bar{E}$, respectively. Then, we consider the different correlation lengths of b ($b = b_x = b_y$), taking values of $0.2b$, $0.4b$, $0.6b$, $0.8b$ and b , respectively. The computation time statistics are documented in the following Table 2 and Table 3:

From the computational time presented in Table 2 and Table 3, it is evident that the proposed approximate sparse preconditioner exhibits high efficiency; moreover, the degree of improvement does not show significant variations as the standard deviation and the correlation length change, thus indicating its numerical scalability.

CONCLUSIONS

In this paper, we develop an approximate sparse expansion-based domain decomposition solver in the context of SSFE. By establishing an approximate sparse expansion of the subdomain-level augmented matrix, we transform the multiplication of the e-SC matrix and vector to the operation of a series of subdomain-level augmented matrices and vectors. The transformation can greatly decrease the computational cost in each iteration in PCG. With the approximate sparse expansion, we further develop an approximate sparse preconditioner in the context of PCG for the solution of the e-SC system. The preconditioner is established as a product of a block diagonal matrix and a sparse matrix with low condition number. The convergence of PCG can be remarkably accelerated using the preconditioner. Since the two main difficulties in the existing DDM-based SSFE are overcome, the developed solver addresses the challenge of difficulty solving the e-SC system in traditional methods, and thereby the uncertainty quantification of practical engineering is quite efficient using SSFE. Two numerical examples, including stochastic analysis of a thin square plate and a planar frame structure, have been studied to illustrate the effectiveness of the developed method. In both examples, the developed approximate sparse expansion-based domain decomposition solver enables greatly improving the efficiency of stochastic response analysis, and the improvement of efficiency is not affected by the number of subdomains and the variance of the random field.

Due to our focus on improving the computational efficiency of the proposed algorithm compared to traditional DDM-based SSFE, we only consider the simplest case, assuming that the structure is elastic and only withstands deterministic static loads.

In future work, we will further evaluate the computational performance of the algorithm under complex conditions (such as nonlinear structures subjected to random dynamic loads^[38]). Additionally, it is necessary to be aware that the proposed method has certain limitations; for example, the SC process is still inevitably required in establishing the approximate sparse preconditioner. So, in future work, we will also consider developing an approximate SC process to reduce its computational cost while meeting the accuracy requirements.

DECLARATIONS

Authors' contributions

Conceptualization, methodology, analysis and writing: Luo B
Software coding, numerical analysis and validation: Cao W
Methodology, supervision and review: Dai H, Zhou Z

Availability of data and materials

Not applicable.

Financial support and sponsorship

This research was supported by a Grant from the National Natural Science Foundation of China (Project 12272109). This support is gratefully acknowledged.

Conflicts of interest

All authors declared that there are no conflicts of interest

Ethical approval and consent to participate

Not applicable.

Consent for publication

Not applicable.

Copyright

© The Author(s) 2024.

REFERENCES

1. Cai J, Hu S, Sun F, Tang L, Fan G, Xing H. Exploring the relationship between risk perception and public disaster mitigation behavior in geological hazard emergency management: a research study in Wenchuan county. *Dis Prev Res* 2023;2:21. DOI
2. Du Y, Xu J. Structural reliability analysis with epistemic and aleatory uncertainties via AK-MCS with a new learning function. *Dis Prev Res* 2023;2:15. DOI
3. De Santis Y, Aloisio A, Pasca DP, Fragiaco M, Dombrowski F. Evaluation of the shear size effect in glued laminated timber using a stochastic FE model calibrated on 17000 glue-line tests. *Constr Build Mater* 2023;399:132488. DOI
4. Ghosh D, Avery P, Farhat C. A FETI-preconditioned conjugate gradient method for large-scale stochastic finite element problems. *Int J Numer Meth Eng* 2009;80:914-31. DOI
5. Zhang R, Dai H. Stochastic analysis of structures under limited observations using kernel density estimation and arbitrary polynomial chaos expansion. *Comput Method Appl Mech Eng* 2023;403:115689. DOI
6. Wang Y, Li J. Probabilistic analysis on stochastic failure modes of reinforced concrete beams under fatigue loading. *J Eng Mech* 2022;148:04022018. DOI
7. Zheng Z, Dai H, Beer M. Efficient structural reliability analysis via a weak-intrusive stochastic finite element method. *Probabilist Eng Mech* 2023;71:103414. DOI
8. Sousedík B, Ghanem RG, Phipps ET. Hierarchical Schur complement preconditioner for the stochastic Galerkin finite element methods. *Numer Linear Algebra Appl* 2014;21:136-51 DOI
9. Peng Y, Zhou T, Li J. Surrogate modeling immersed probability density evolution method for structural reliability analysis in high dimensions. *Mech Syst Signal Proc* 2021;152:107366. DOI
10. Dai H, Zhang R, Beer M. A new method for stochastic analysis of structures under limited observations. *Mech Syst Signal Proc* 2023;185:109730. DOI
11. Pang R, Zhou Y, Chen G, Jing M, Yang D. Stochastic mainshock-aftershock simulation and its applications in dynamic reliability of structural systems via DPIM. *J Eng Mech* 2023;149:04022096. DOI
12. He J, Chen J, Ren X, Li J. Uncertainty quantification of random fields based on spatially sparse data by synthesizing bayesian compressive sensing and stochastic harmonic function. *Mech Syst Signal Proc* 2021;153:107377. DOI
13. Pranesh S, Ghosh D. Addressing the curse of dimensionality in SSFEM using the dependence of eigenvalues in KL expansion on domain size. *Comput Method Appl Mech Eng* 2016;311:457-75. DOI
14. Meng Z, Zhao J, Chen G, Yang D. Hybrid uncertainty propagation and reliability analysis using direct probability integral method and exponential convex model. *Reliab Eng Syst Safe* 2022;228:108803. DOI
15. Kundu A, Diazdelao F, Adhikari S, Friswell M. A hybrid spectral and metamodeling approach for the stochastic finite element analysis of structural dynamic systems. *Comput Method Appl Mech Eng* 2014;270:201-19. DOI
16. Wan Z, Chen J, Li J. Probability density evolution analysis of stochastic seismic response of structures with dependent random parameters. *Probabilist Eng Mech* 2020;59:103032. DOI
17. Mouyeaux A, Carvajal C, Bressolette P, Peyras L, Breul P, Bacconnet C. Probabilistic stability analysis of an earth dam by stochastic finite element method based on field data. *Comput Geotech* 2018;101:34-47. DOI
18. Zheng Z, Dai H. Structural stochastic responses determination via a sample-based stochastic finite element method. *Comput Method Appl Mech Eng* 2021;381:113824. DOI
19. Xiu D. Numerical methods for stochastic computations: a spectral method approach. Princeton, NJ: Princeton University Press; 2010.
20. Stefanou G. The stochastic finite element method: past, present and future. *Comput Method Appl Mech Eng* 2008;198:1031-51. DOI
21. Matthies HG, Keese A. Galerkin methods for linear and nonlinear elliptic stochastic partial differential equations. *Comput Method Appl Mech Eng* 2005;194:1295-331. DOI
22. Ghanem RG, Spanos PD. Stochastic finite elements: a spectral approach. New York Berlin Heidelberg: Springer; 1991.
23. Gopalakrishnan S, Chakraborty A, Mahapatra DR. Spectral finite element method. London: Springer; 2012.
24. Dai H, Zheng Z, Ma H. An explicit method for simulating non-Gaussian and non-stationary stochastic processes by Karhunen-Loeve and polynomial chaos expansion. *Mech Syst Signal Proc* 2019;115:1-13. DOI
25. Stavroulakis G, Giovanis DG, Papadopoulos V, Papadrakakis M. A GPU domain decomposition solution for spectral stochastic finite element method. *Comput Method Appl Mech Eng* 2017;327:392-410. DOI
26. Farhat C, Wilson E, Powell G. Solution of finite element systems on concurrent processing computers. *Eng Comput* 1987;2:157-65. DOI
27. Farhat C, Lesoinne M, Pierson K. A scalable dual-primal domain decomposition method. *Numer Linear Algebra Appl* 2000;7:687-714. DOI
28. Sarkar A, Benabbou N, Ghanem R. Domain decomposition of stochastic PDEs: theoretical formulations. *Int J Numer Methods Eng* 2009;77:689-701. DOI
29. Cros JM. A preconditioner for the Schur complement domain decomposition method. 2003. Available from:<http://www.ddm.org/DD14/cros.pdf>[Last accessed on 21 Mar 2024]
30. Desai A, Khalil M, Pettit C, Poirel D, Sarkar A. Scalable domain decomposition solvers for stochastic PDEs in high performance computing.

- Comput Method Appl Mech Eng* 2018;335:194-222. [DOI](#)
31. Subber W, Sarkar A. A domain decomposition method of stochastic PDEs: An iterative solution techniques using a two-level scalable preconditioner. *J Comput Phys* 2014;257:298-317. [DOI](#)
 32. Toselli A, Widlund OB. Domain decomposition methods-algorithms and theory. In Springer series in computational mathematics, Berlin: Springer; 2005. [DOI](#)
 33. Pranesh S, Ghosh D. A FETI-DP based parallel hybrid stochastic finite element method for large stochastic systems. *Comput Struct* 2018;195:64-73. [DOI](#)
 34. Mathew TPA. Domain decomposition methods for the numerical solution of partial differential equations. Berlin: Springer; 2008. [DOI](#)
 35. Saad Y. Basic iterative methods. In: Iterative methods for sparse linear systems. 2003. [DOI](#)
 36. Babuska I, Tempone R, Zouraris GE. Galerkin finite element approximations of stochastic elliptic partial differential equations. *SIAM J Numer Anal* 2004;42:800-25. [DOI](#)
 37. Ullmann E. A kronecker product preconditioner for stochastic galerkin finite element discretizations. *SIAM J Sci Comput* 2010;32:923-46. [DOI](#)
 38. Schenk CA, Pradlwarter HJ, Schuëller GI. On the dynamic stochastic response of FE models. *Probabilist Eng Mechan* 2004;19:161-70. [DOI](#)

APPENDIX

Algorithm 1 Approximate sparse relation matrix of subdomain – level augmented matrix

```

1: Input result of KL expansion  $\mathbf{K}_{i,II}^s$  for  $i = 0, 1, 2, \dots, m$  and  $s = 1, 2, \dots, N_s$ 
2: for  $s = 1, 2, \dots, N_s$  do
3:   for  $i = 0, 1, 2, \dots, m$  do
4:     Calculate  $\mathbf{R}_{i,II}^s = (\mathbf{K}_{0,II}^s)^{-1} \mathbf{K}_{i,II}^s$ 
5:     Calculate  $\tilde{\mathbf{R}}_{i,II}^s = \ell(\mathbf{R}_{i,II}^s)$  by (17)
6:   end for
7:   Calculate  $\tilde{\mathbf{R}}_{II}^s = \sum_{i=0}^m \Theta(\mathbf{A}_i, \tilde{\mathbf{R}}_{i,II}^s)$ 
8: end for
9: Output  $\tilde{\mathbf{R}}_{II}^s$ 

```

Algorithm 2 Multiplication of argument matrix and vector

```

1: Input  $\mathbf{p}$ 
2: Initialize  $\mathbf{q}_j = \mathbf{0}$ ,  $j = 0, 1, 2, \dots, M - 1$ 
3: for  $(i, j, k, c_{ijk}) \in \tilde{\lambda}$  do
4:    $\mathbf{q}_j = \mathbf{q}_j + \mathbf{K}_i \mathbf{p}_k$ 
5:  $\mathbf{q} = \{ \mathbf{q}_0 \quad \mathbf{q}_1 \quad \dots \quad \mathbf{q}_{M-1} \}$ 
6: end for
7: Output  $\mathbf{q}$ 

```

Algorithm 3 Parallelmultiplication of e – SC matrix and vector

```

1: Input  $\mathbf{p}$ 
2: for  $s = 1, 2, 3, \dots, N_s$ , parallely do
3:   Compute  $\mathbf{p}_s = \mathbf{B}_S^s \mathbf{p}$ 
4:   Compute  $\mathbf{q}_1^s = \mathbf{K}_{\Gamma T}^s \mathbf{p}_s$  by Algorithm 2
5:   Compute  $\mathbf{q}_a^s = \mathbf{K}_{\Gamma I}^s \mathbf{p}_s$  by Algorithm 2
6:   Solve  $\mathbf{K}_{II}^s \mathbf{q}_b^s = \mathbf{q}_a^s$  by PCG, preconditioner  $\mathbf{M}^s = (\mathbf{I} \otimes \mathbf{K}_{0,II}^s) \tilde{\mathbf{R}}_{II}^s$ 
7:   Compute  $\mathbf{q}_2^s = \mathbf{K}_{\Gamma I}^s \mathbf{q}_b^s$  by Algorithm 2
8:   Compute  $\mathbf{q}^s = \mathbf{q}_1^s - \mathbf{q}_2^s$ 
9: end for
10: Compute  $\mathbf{q} = \sum_{s=1}^{N_s} (\mathbf{B}_S^s)^T \mathbf{q}^s$ 
11: Output  $\mathbf{q}$ 

```

Algorithm 4 Parallel computation for the relation matrix of e – SC matrix

- 1: Establish the second level Boolean matrix $\mathbf{B}_L^{(i)}$, for $i = 1, 2, \dots, L_i$
 - 2: Compute $\mathbf{S} = \sum_{i=1}^{L_i} \left(\overline{\mathbf{K}}_{0,\Gamma}^i - \overline{\mathbf{K}}_{0,\Gamma}^i \left(\overline{\mathbf{K}}_{0,II}^i \right)^{-1} \overline{\mathbf{K}}_{0,\Gamma}^i \right)$
 - 3: Compute \mathbf{S}^{-1}
 - 4: **for** $i = 0, 1, 2, \dots$, parallely **do**
 - 5: Choose j and k satisfy $(i, j, k) \in \mathfrak{J}$
 - 6: Compute $\mathbf{R}_{jk}^{(L_i+1, L_i+1)}$ by Equation (47)
 - 7: **for** $s = 1, 2, 3, \dots, L_s$ **do**
 - 8: Compute $\mathbf{R}_{jk}^{(L_i+1, s)}$ by Equation (45)
 - 9: Compute $\mathbf{R}_{jk}^{(s, L_i+1)}$ by Equation (48)
 - 10: **for** $t = 0, 1, 2, 3, \dots, L_s$ **do**
 - 11: Compute $\mathbf{R}_{jk}^{(t, s)}$ by Equation (46)
 - 12: Approximate $\tilde{\mathbf{R}}_{jk}^{(t, s)} = \ell \left(\mathbf{R}_{jk}^{(t, s)} \right)$ by Equation (17)
 - 13: **end for**
 - 14: Approximate $\tilde{\mathbf{R}}_{jk}^{(L_i+1, s)} = \ell \left(\mathbf{R}_{jk}^{(L_i+1, s)} \right)$ by Equation (17)
 - 15: Approximate $\tilde{\mathbf{R}}_{jk}^{(s, L_i+1)} = \ell \left(\mathbf{R}_{jk}^{(s, L_i+1)} \right)$ by Equation (17)
 - 16: Approximate $\tilde{\mathbf{R}}_{jk}^{(L_i+1, L_i+1)} = \ell \left(\mathbf{R}_{jk}^{(L_i+1, L_i+1)} \right)$ by Equation (17)
 - 17: Assembly $\tilde{\mathbf{R}}'_{jk}$ by Equation (49)
 - 18: Approximate $\tilde{\mathbf{R}}_i = \ell(\mathbf{R}_{jk})$ by Equation (17)
 - 19: **end for**
 - 20: **end for**
 - 21: Assembly $\tilde{\mathbf{R}}$ according to $\tilde{\mathbf{R}}_{jk}$ by $\sum_{i=0}^m c_{irq} \mathbf{R}_i$
 - 22: Output $\tilde{\mathbf{R}}$
-

Algorithm 5 Approximate sparse approach for the e – SC system

- 1: Input $\mathbf{u}_\Gamma = 0; \varepsilon = 1$
 - 2: Parallely compute $\mathbf{r} = \mathbf{g} - \overline{\mathbf{K}}\mathbf{u}_\Gamma$ by Algorithm 3
 - 3: Parallel Preconditioned Residual: $\mathbf{r} = \Theta(\mathbf{I}, \overline{\mathbf{K}}_0) \tilde{\mathbf{R}}\mathbf{z}$
 - 4: Compute: $\mathbf{p} = \mathbf{z}$
 - 5: Compute: $\delta = (\mathbf{r}, \mathbf{z})$
 - 6: **While** $\varepsilon \leq 10^{-8}$, **do** :
 - 7: Parallely compute $\mathbf{q} = \overline{\mathbf{K}}\mathbf{p}$ by Algorithm 3
 - 8: Compute: $\gamma = (\mathbf{q}, \mathbf{p})$
 - 9: Compute: $\alpha = \delta/\gamma$
 - 10: Update: $\mathbf{u}_\Gamma = \mathbf{u}_\Gamma + \alpha\mathbf{p}$
 - 11: Update: $\mathbf{r} = \mathbf{r} - \alpha\mathbf{q}$
 - 12: Parallel Preconditioned Residual: $\mathbf{r} = \Theta(\mathbf{I}, \overline{\mathbf{K}}_0) \tilde{\mathbf{R}}\mathbf{z}$
 - 13: Compute: $\beta = (\mathbf{r}, \mathbf{z})/\delta$
 - 14: Compute: $\delta = \beta\delta$
 - 15: Update: $\mathbf{p} = \mathbf{z} + \beta\mathbf{p}$
 - 16: $\varepsilon = (\mathbf{r}, \mathbf{r})/(\mathbf{g}, \mathbf{g})$
 - 17: **End while**
 - 18: **Output** : \mathbf{u}_Γ
-