16th World Congress in Computational Mechanics (WCCM)
21-26 July 2024, Vancouver, Canada
A. Korobenko, M. Laforest, S. Prudhomme, R. Vaziri (Eds)

TOTAL FINITE ELEMENT TEARING AND INTERCONNECTING METHOD FOR CONCRETE SIMULATION

Lin ZHANG[a],Xu YANG[a,b,c,*]

[a] [a]School of Civil Engineering, Harbin Institute of Technology, Harbin 150090, China
[b]Key Lab of Structures Dynamic Behavior and Control of the Ministry of Education, Harbin Institute of Technology, Harbin 150090, China
[c]Key Lab of Smart Prevention and Mitigation of Civil Engineering Disasters of the Ministry of Industry and Information Technology, Harbin Institute of Technology, Harbin 150090, China
[*] Corresponding author, mail: yangxu_civileng@hit.edu.cn

Key words: TFETI, FEM, Reliability Analysis, Monte Carlo

Abstract. Concrete structure is widely used in civil engineering. Numerical Simulation in mesoscale could provide a powerful analysis tool, which is based on the Total Finite Element Tearing and Interconnecting (TFETI) method. However, the unneglectable computational resource consumption limited the implementation of the meso-modeling method in engineering. This study investigated an efficient solution for concrete structure simulation, which was programmed using Python. A cantilever beam model with dimensions of 100mm $\times$ 50mm was simulated and studied. Then, a comparison with the ABAQUS model was given to demonstrate the accuracy and efficiency of the TFETI. Furthermore, a reliability analysis of a 300mm $\times$ 300mm concrete sample under the axial compression was performed. The results showed that the TFETI method achieved the same level of accuracy as ABAQUS. Moreover, the TFETI model solving required less computational source for the same number of elements as the one on the ABAQUS. In the reliability analysis of the axially loaded model, TFETI demonstrated superior solution speed. In conclusion, the TFETI exhibits excellent solution efficiency in finite element analysis of concrete structures, offering valuable insights and references for computational analysis of large-dimension civil engineering structures.

1 INTRODUCTION

Numerical simulation techniques are important for simulating and analyzing the mechanical behavior of structural systems under different loading conditions in civil engineering. However, as modern engineering structures become more complex and require higher levels of detail, traditional numerical algorithms face limitations. These limitations are mainly seen in dealing with large, highly detailed models, resulting in a large number of unknown variables and the need to construct high-order system matrices for solutions. This significantly increases computational costs and puts high demands on the memory and computational capabilities of computers. As a result, traditional algorithms face challenges in terms of efficiency and resource constraints when solving practical engineering problems.

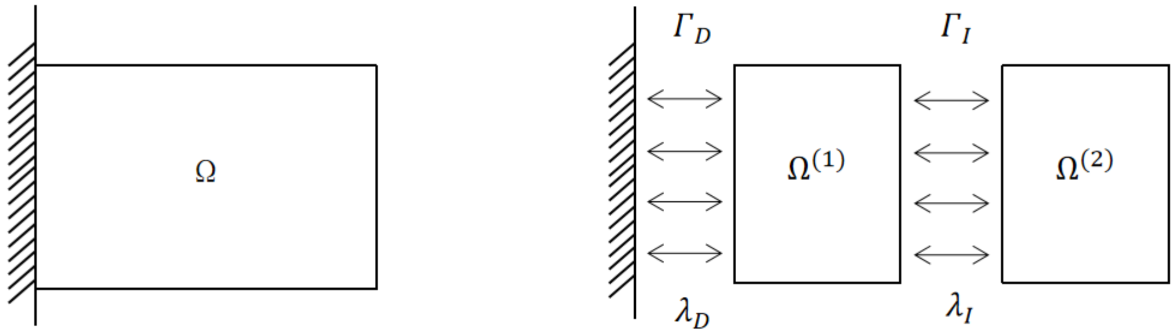
Recently, Multiscale finite element methods have gradually come to the fore as the mainstream approach for numerical simulation of large-dimension structures. This method integrates models of different scales. It employs fine-scale models at critical locations based on the characteristics of structural components or nodes and the degree of nonlinearity in the damage process. It also utilizes macroscopic models in non-critical regions to maintain computational efficiency. By significantly reducing the overall number of elements, this approach offers a more efficient and economical means for the numerical simulation of complex structures. However, as the area for studying the distribution of damage expands, the increase in the fine-scale computational region still limits further improvement in computational efficiency, presenting new challenges that need to be addressed.

The domain decomposition method is a numerical algorithm that follows the ”divide and conquer” principle[1]. It has proven to be highly advantageous for large-scale numerical computations due to its strong parallelism and efficiency. This method involves dividing the computational domain into several subdomains and conducting independent numerical calculations within each subdomain to achieve the overall computational goal. Because the computations in each subdomain are performed in parallel, the computational power of multi-core processors or distributed computing clusters can be fully utilized, significantly boosting computational speed. With its potential to enhance the speed of finite element computations, the domain decomposition method introduces a new possibility for multiscale simulations of complex structures. It is anticipated to further improve the computational efficiency of multiscale methods while maintaining computational accuracy.

The domain decomposition method can be broadly categorized into overlapping and non-overlapping types. The overlapping domain decomposition method, based on the classical Schwarz alternating algorithm, initially partitions the original solution domain S into the intersection of several subdomains, where each subdomain overlaps with others[2]. And the non-overlapping domain decomposition method also divides the original solution domain into multiple subdomains, but unlike the overlapping method, these subdomains do not overlap with each other [3]. Presently, non-overlapping domain decomposition methods mainly include the Finite Element Tearing and Interconnecting (FETI) method and the Total Finite Element Tearing and Interconnecting (TFETI) method, etc.

The FETI method breaks down the domain being studied into separate, non-overlapping subdomains. It eliminates the continuity constraints between the interfaces of each subdomain using Lagrange multipliers [4]. Compared to the Schwarz alternating method, FETI shows faster convergence and more flexibility in application. The TFETI method extends the FETI method by introducing Lagrange multiplier constraints on the interfaces between subdomains and imposing Dirichlet boundary conditions on boundary subdomains [5]. This approach addresses issues such as poor convergence of iterative solutions at the interfaces of FETI subdomains.

The analysis of reliability is essential for assessing the ability of engineering structures to perform their intended functions under specific conditions over a given period. However, the large datasets involved in reliability analysis often limit its widespread application in civil engineering. The TFETI method’s computational efficiency can potentially extend its use in reliability



(a) Domain Ω (b) subdomains $\Omega^{(1)}$ and $\Omega^{(2)}$

Figure 1: Domain dividing

analysis for engineering structures, thereby broadening its utility and laying the groundwork for reliability analysis in other engineering fields.

This study focused on a concrete cantilever beam and assessed the accuracy and efficiency of TFETI in the finite element analysis of the structure. Additionally, an identical numerical model of the cantilever beam was created using ABAQUS to compare the computational accuracy and speed of both methods. This comparison aimed to explore the potential applications of TFETI in the finite element analysis of concrete structures and provide insights for solving large-scale concrete structure finite element problems. Furthermore, the paper combined TFETI with Monte Carlo to conduct a reliability analysis of the axially loaded model, with the goal of expanding the application of TFETI in civil structural engineering.

2 METHOD

2.1 TFETI

The main steps to implement TFETI are as follows:

1. Domain partitioning: The solution domain, as shown in Figure 1, is divided into two subdomains,$\Omega^{(1)}$and$\Omega^{(2)}$. Following Zdeněk's[5] approach, Lagrange multipliers λ_I are employed to connect the subdomains. Additionally, Dirichlet boundary conditions Γ_D for the subdomains are also handled using Lagrange multipliers λ_D.

2. Establishing connectivity: According to the Galerkin method, the mixed variational equations for the two subdomains are discretized as follows[6]:

$$\begin{aligned} K^{(1)}u^{(1)} &= f^{(1)} \\ K^{(2)}u^{(2)} &= f^{(2)} \end{aligned} \tag{1}$$

$K^{(s)}$,$u^{(s)}$, and$f^{(s)}$ represent the stiffness matrix, displacement vector, and load vector of subdomain $\Omega^{(s)}$ respectively.

Displacement continuity must be ensured between subdomain $\Omega^{(1)}$ and subdomain $\Omega^{(2)}$, as depicted in Equation 2, while subdomain $\Omega^{(1)}$ must also satisfy to boundary displacement constraint conditions.

$$u^{(1)} = u^{(2)} \tag{2}$$

Let $B^{(s)}$ denote the Boolean matrix of subdomains $\Omega^{(s)}$, where $B^{(s)}$ is a matrix containing only (-1, 0, 1). The interface information for subdomains $\Omega^{(1)}$ and $\Omega^{(2)}$ can be represented by $B^{(1)}$ and $B^{(2)}$, respectively. The Boolean matrix $B^{(s)}$ of each subdomain must satisfy Equation 3.

$$B^{(1)}u^{(1)} + B^{(2)}u^{(2)} = 0 \tag{3}$$

Furthermore, as illustrated in Figure 1b, to account for the imposition of Dirichlet boundary Γ_D in subdomain $\Omega^{(1)}$, the Boolean matrix B needs to be expanded into the form depicted in Equation 4.

$$B = \begin{bmatrix} B_I \\ B_D \end{bmatrix} \tag{4}$$

B_I represents the constraint relationship between subdomains and other subdomains, while B_D signifies the constraint conditions of the Dirichlet boundary within subdomains.

Similarly, the Lagrange multiplier is also updated to reflect these constraints.

$$\lambda = \begin{bmatrix} \lambda_I \\ \lambda_D \end{bmatrix} \tag{5}$$

3. Double assembly system equation establishment: By employing the double assembly method, the stiffness equations and Boolean matrices of each subdomain are assembled into the global coefficient matrix of finite element analysis.

$$\begin{bmatrix} K^{(1)} & 0 & B^{(1)^T} \\ 0 & K^{(2)} & B^{(2)^T} \\ B^{(1)} & B^{(2)} & 0 \end{bmatrix} \begin{bmatrix} u^{(1)} \\ u^{(2)} \\ \lambda \end{bmatrix} = \begin{bmatrix} f^{(1)} \\ f^{(2)} \\ 0 \end{bmatrix} \tag{6}$$

The equation set forms a large sparse matrix. And it can be solved using the PARDISO solver to obtain u and λ[7].

2.2 Reliability calculations based on TFETI

The TFETI method is highly computationally efficient and offers excellent numerical stability. It provides a novel approach to addressing computational challenges in reliability analysis. When combined with Monte Carlo (MC), TFETI makes it feasible to significantly reduce the computational costs of reliability analysis while ensuring computational accuracy[8].

Monte Carlo (MC) is a numerical computing method based on random sampling and statistical simulation. Its core idea is to estimate solutions to mathematical problems through random sampling and to approximate real situations through numerous random experiments. The implementation steps are as follows:

1. Problem definition: Transform the problem into an integral or expected value that needs to be solved.

$$I = \int_a^b f(x)\,dx \tag{7}$$

2. Generate random samples: Generate random samples $x_1, x_2, x_3, ..., x_N$ that conform to the probability distribution p(x), where N is the sample size.

3. Evaluate the model: Substitute the random samples into the function f(x) for calculation or simulation.

$$\mathrm{I} \approx \frac{1}{\mathrm{N}} \sum_{\mathrm{i}=1}^{\mathrm{N}} \frac{\mathrm{f}(\mathrm{x_i})}{\mathrm{p}(\mathrm{x_i})} \tag{8}$$

4. Statistical analysis: Perform statistical analysis on the simulation results, such as computing the mean, etc.

$$\bar{\mathrm{I}} \approx \tfrac{\mathrm{b}-\mathrm{a}}{\mathrm{N}} \textstyle\sum_{\mathrm{i}=1}^{\mathrm{N}} \mathrm{f}(\mathrm{x_i}) \tag{9}$$

There are numerous methods for analyzing reliability in civil engineering, all of which are based on the principles of probability theory and mathematical statistics. Here is a demonstration of a simple reliability calculation formula:

$$R = \frac{N - N_t}{N} \tag{10}$$

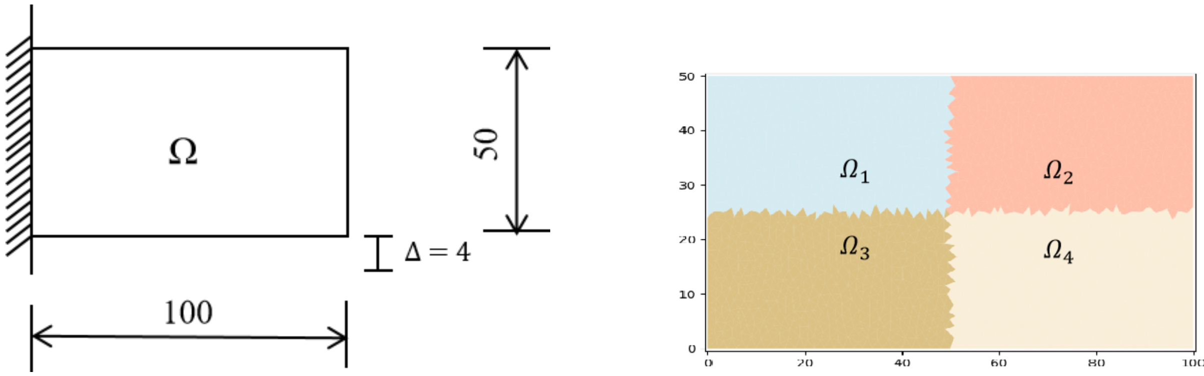
Where N represents the total number of samples, and N_t represents the number of invalid samples. The reliability R varies between 0 and 1, with higher values indicating greater structural reliability.

3 Simulation

3.1 Benchmark test

A numerical simulation analysis is conducted on a cantilever beam using TFETI to validate the accuracy of this method, and the results are compared with an identical model established in ABAQUS software to verify the accuracy and computational speed of this method. The specimen dimensions and boundary conditions are depicted in Figure 2a. The model dimensions are 100mm $\times$ 50mm, with the left side fully fixed, the top and bottom sides free, and the right side subjected to loading boundary conditions using displacement loading. Additionally, the material of the cantilever beam is chosen as C30 concrete, with an elastic modulus of 30e3 N/mm^2 and a Poisson's ratio of 0.2.

The cantilever beam model is divided into four non-overlapping subdomains, each with dimensions of 50mm $\times$ 25mm, as shown in Figure 2b. The subdomains are meshed using constant strain triangular elements, and the model is divided using the gmsh meshing software, resulting in a total of 1745 nodes and 3492 elements.



(a) Cantilever model (b) Domain decomposition

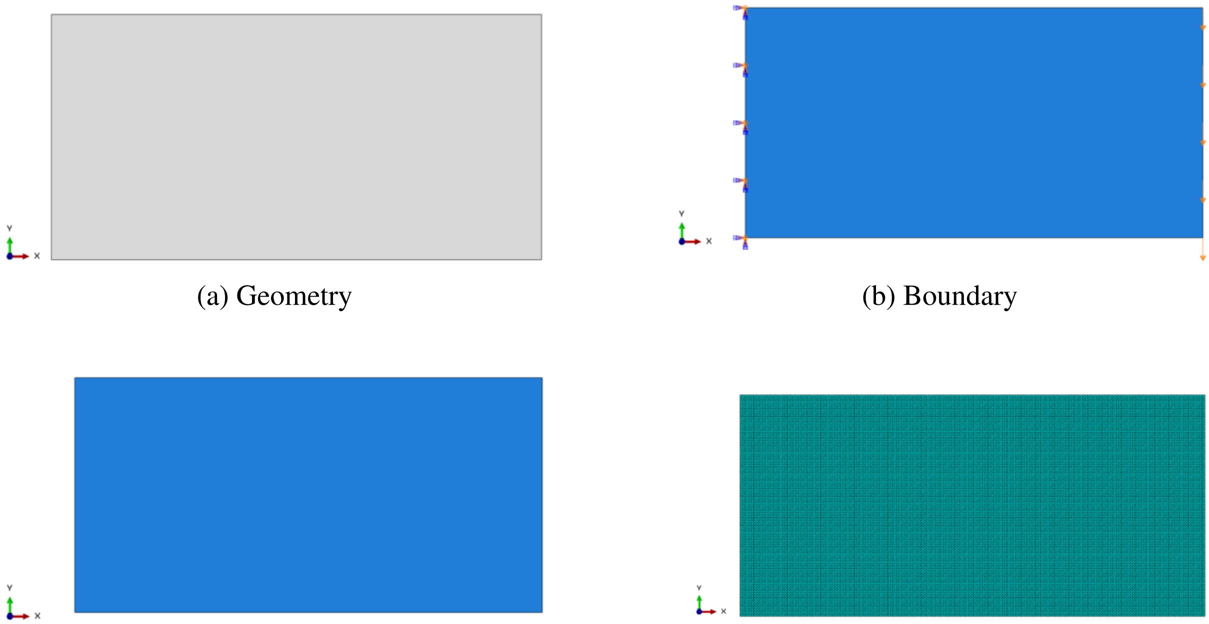
Figure 2: Cantilever model and decomposition

The cantilever beam model is created and analyzed in ABAQUS using the steps shown in Figures 5 to 8. The chosen element type is CPE3, and the global mesh division ranges from 0.55m to 1.6m. The total number of elements generated ranges from 3906 to 33124.

3.2 Reliability

In this study, an axial compression model will be utilized for reliability analysis, with specimen dimensions and boundary conditions as depicted in Figure 4a. The model size is 300mm $\times$ 300mm, with the bottom fully fixed, left and right sides free, and a force-loading boundary at the top. The material elastic modulus is 31.5e3 N/mm^2, Poisson's ratio is 0.2, axial compressive strength standard value is 23.4 N/mm^2, and shear strength standard value is 12 N/mm^2.

As shown in Figure 4b, the subdomains are meshed using constant strain triangular elements. The model is divided using the gmsh meshing software, resulting in a total of 3430 nodes and 6862 elements.



(a) Geometry (b) Boundary

(c) Load (d) Mesh

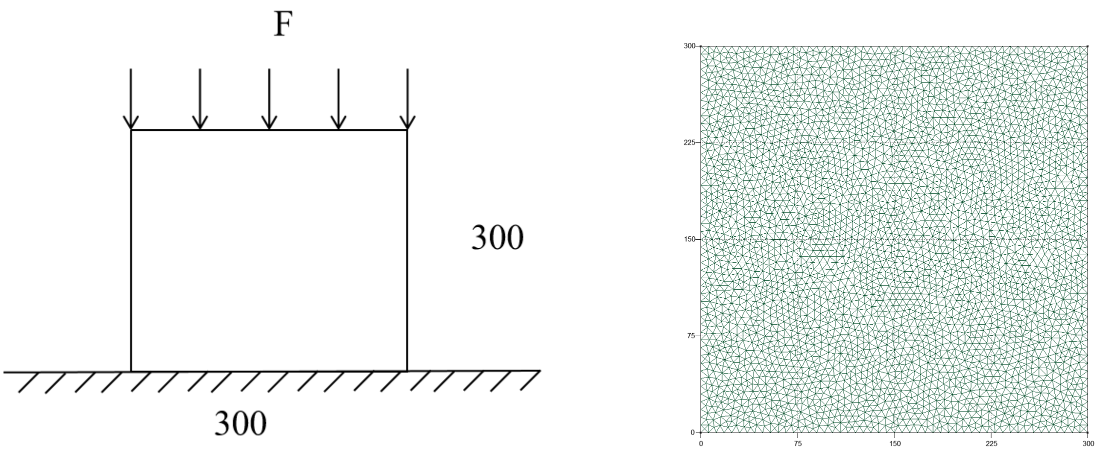
Figure 3: Modeling process(ABAQUS)

The load F varies between 45 and 63 N/m, with 18 data points evenly distributed across the load range. Each load group is labeled as F_i. For each group, the load is randomly sampled from a normal distribution with mean F_i and variance 3. Each load group undergoes 10^6 iterations, during which the maximum normal stress value of the axial compression model is calculated. This value is then compared with the axial compressive strength standard value to determine whether the model reaches failure, as shown in Equation 11.

$$\sigma_{max} \leq [\sigma] \tag{11}$$

For the same model, we calculate the maximum shear stress value of the axial compression model and compare it with the standard axial shear strength value to determine if the model fails. This comparison is shown in Equation 12. The load varies between 10.9 and 11.4 N/m, with 10 data points evenly distributed across the load spans. Each load group is denoted as F_j, and for each group, the load is randomly sampled from a normal distribution with a mean of F_j and a variance of 0.1. Each load group undergoes 10^6 iterations.

$$\tau_{max} \leq [\tau] \tag{12}$$



(a) Axial compression model

(b) Mesh

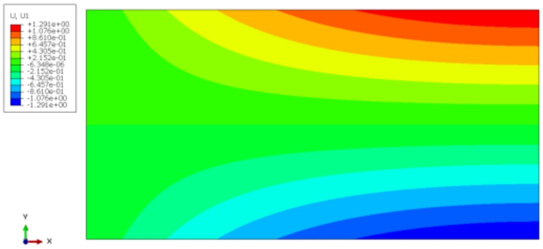
Figure 4: Geometry and mesh

4 RESULTS AND DISCUSSIONS

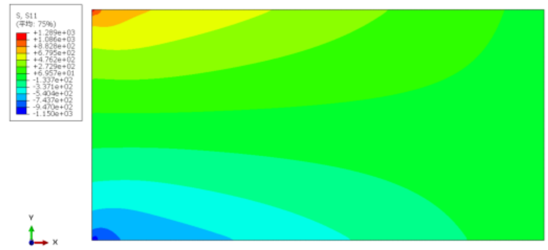
4.1 Validation

The comparison of TFETI and ABAQUS results, as depicted in Figure 5, shows that the X-displacement contour obtained using the TFETI method in Figure 5a is consistent with that from ABAQUS in Figure 5c. ABAQUS calculates the maximum displacement value as 1.298 mm, while TFETI computes it as 1.224 mm, meeting the calculation requirements. Similarly, the distribution of the first principal stress obtained using the TFETI method matches that of ABAQUS results, and the computed values are within the same order of magnitude. This indicates that TFETI's calculations meet the requirements for computational accuracy.

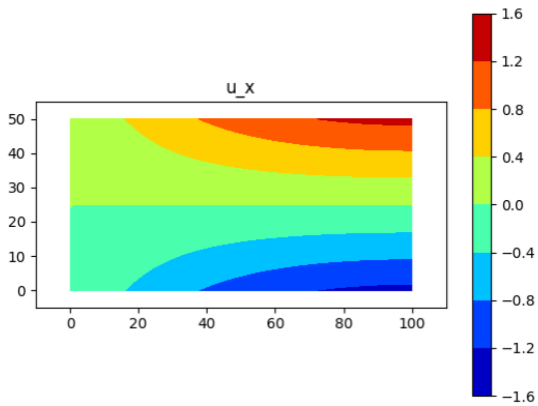
To assess the efficiency of TFETI, calculations were performed on cantilever beam models with varying numbers of elements using both TFETI and ABAQUS. The computation for both methods is illustrated in Figure 6. It is evident from the graph that when the number of elements is below 10,000, TFETI's computation time is approximately 23 seconds less than that of ABAQUS. However, as the number of elements increases, ABAQUS's computation time grows at a more stable rate, while TFETI's computation speed shows a significant increase. At 30,000 elements, TFETI's computation time is only about 6 seconds less than ABAQUS.



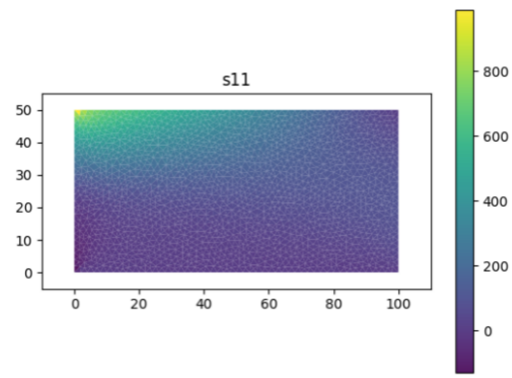
(a) ABAQUS_Ux



(b) ABAQUS_S11



(c) TFETI_Ux



(d) TFETI_S11

Figure 5: Comparison of the calculation results of ABAQUS and TFETI

The increase in computation time for TFETI can be attributed to several factors. Firstly, TFETI is implemented in Python, which may not be as optimized as ABAQUS. Additionally, TFETI's solution involves assembling large systems of equations, which demands significant computer memory. The lack of sufficient memory significantly impacts TFETI's computation speed.

In summary, disregarding differences in program optimization and hardware setups, TFETI demonstrates higher computational efficiency compared to ABAQUS. This suggests that enhancing computational efficiency through domain decomposition is a reliable approach.

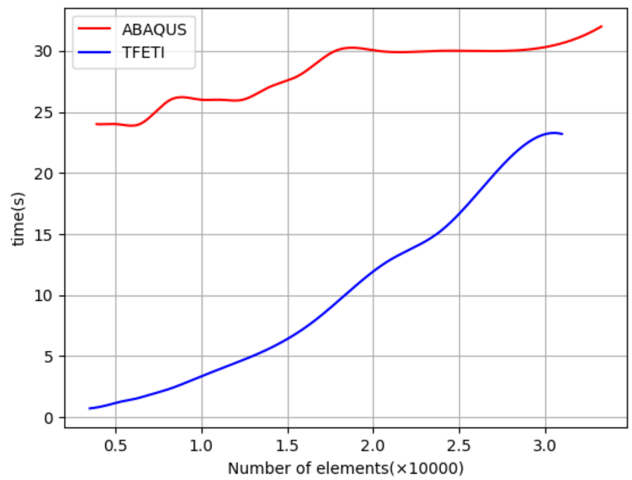


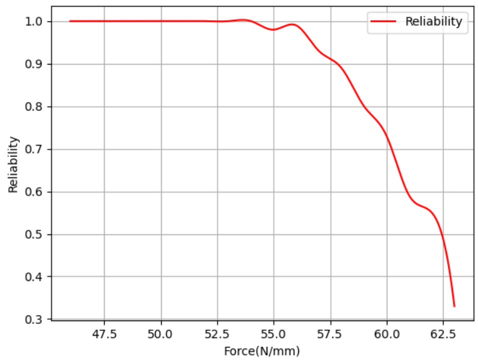
Figure 6: Comparison of the calculation time of TFETI and ABAQUS for different elements numbers

4.2 Reliability analysis

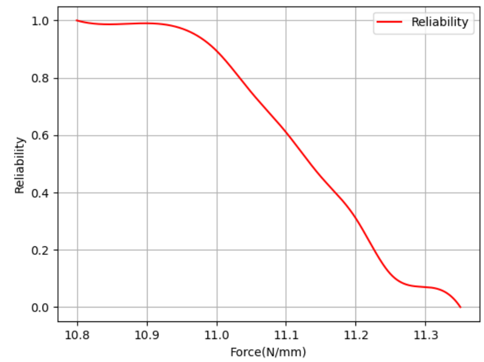
The results are calculated as shown in Figures 7a and 7b. It was observed that in the first group of reliability analyses, the axial compression model is safe when the load is 55 N/m or below, with a reliability of 1. However, as the load exceeds 55 N/m, the reliability gradually decreases, reaching 0.33 at 63 N/m. In the second group of analyses, the structural reliability is assured when the line load is less than 10.9 N/m. But as the line load exceeds 10.9 N/m, the reliability gradually decreases until reaching 0 at 11.4 N/m.

The first experiment involved calculations for 18 different load values, each with 10^6 iterations, totaling 14.2 hours. The second experiment involved calculations for 10 load values, each with 10^6 iterations, totaling 7.8 hours.

The experimental results show that conducting a reliability analysis of the axial compressive model using TFETI in this study demonstrates excellent computational efficiency. TFETI implemented in Python exhibits strong capabilities in handling multiple models, especially when dealing with a large number of model calculations. Additionally, the line load causing shear failure in the axial compressive model is significantly lower than the line load causing compressive failure.



(a) The reliability of compression bearing capacity



(b) The reliability of shear bearing capacity.

Figure 7: Reliability of the maximum normal stress and the maximum shear stress

5 CONCLUSIONS

The TFETI method was used in this study to create models for both a cantilever beam and axial compressive models. The combination of TFETI with MC was employed to perform a reliability analysis of the axial compressive model. The final results are as follows:

(1) The results of numerical calculations on identical cantilever beam models using TFETI and ABAQUS showed that the displacement and stress distributions obtained from TFETI are in good agreement with those from ABAQUS. This indicates that applying the TFETI domain decomposition method in finite element computations is rational. Furthermore, when creating cantilever beam models with different numbers of elements using TFETI and ABAQUS, it was found that the TFETI method requires significantly less computation time than ABAQUS, especially with a lower number of elements. This suggests that TFETI offers certain advantages in computational efficiency with a lower number of elements.

(2) After conducting reliability analysis on axial compression models using a combination of TFETI and MC methods, both sets of results demonstrated the high computational efficiency and accuracy of the TFETI method. This finding highlights the significant potential of TFETI in engineering applications and provides valuable references and insights for future finite element analyses.

REFERENCES

[1] Smith B F .Domain Decomposition Methods for Partial Differential Equations[J]. 1997.DOI:10.1007/978-94-011-5412-3-8.

[2] Schwarz H A. Gesammelte Mathematiche Abhandlungen[J]. Springer, Berlin, 1890, 2: 133- 143.

[3] Benamou J-D, Desprès B. A Domain Decomposition Method for the Helmholtz Equation and Related Optimal Control Problems[J]. Journal of Computational Physics, 1997, 136(1): 68–82.

[4] Charbel,Farhat,Francois-Xavier,et al.A method of finite element tearing and interconnecting and its parallel solution algorithm[J].International Journal for Numerical Methods in Engineering, 1991.DOI:10.1002/nme.1620320604.

[5] Dostal Z, Horak D, Kucera R . Total FETI-an easier implementable variant of the FETI method for numerical solution of elliptic PDE[J].Communications in numerical methods in engineering, 2006(12):22.

[6] FARHAT, CHARBEL. ”A multigrid-like semi-iterative algorithm for the massively parallel solution of large scale finite element systems.” Copper Mountain Conference on Multigrid Methods, 4th, Copper Mountain, CO. 1989.

[7] Schenk O, Grtner K . Solving Unsymmetric Sparse Systems of Linear Equations with PARDISO[J]. 2006.DOI:10.1016/j.future.2003.07.011.

[8] Eckhardt R . Stan Ulam, John von Neumann, and the Monte Carlo Method[J].los alamos science special issue, 1987.