Journal homepage: https://itce.com.ua/en

Information Technologies and Computer Engineering **Vol. 22, No. 2. 2025**

Article's History: Received: 04.04.2025 Revised: 30.06.2025 Accepted: 28.08.2025

UDC 519.63:004.272.2

DOI: 10.31649/vitce/2.2025.96

Application of the finite element method for real-time modelling of physical processes using distributed computing

Vladyslav Kozub*

PhD in Computer Sciences, Assistant
Luhansk Taras Shevchenko National University
36000, 3 Kovalia Str., Poltava, Ukraine
https://orcid.org/0000-0003-2710-7206

Abstract. This study investigated the application of the finite element method for modelling complex physical processes in real time. The aim of the research was to determine the effectiveness of this method when combined with distributed computing. An integrated model was developed, combining classical numerical analysis with modern distributed system technologies to ensure high accuracy and computational efficiency. It was established that the finite element method had traditionally been used for modelling heat transfer, deformations, and electromagnetic phenomena. However, modern requirements for monitoring and control created the need to adapt this method for distributed computing. Algorithms for the efficient distribution of computational tasks were developed, allowing data processing delays to be minimised. Experimental simulations showed that the use of distributed computing reduced calculation time by almost 14.5 times – from 420 seconds on a single node to 29 seconds on 16 nodes – while maintaining a relative error of 2-4%. More than 50 test runs were conducted, confirming the system's operational stability. The use of an adaptive integration step reduced computation time by 15% compared to a fixed step, demonstrating the effectiveness of load distribution optimisation. The obtained results confirmed the high potential of this method for solving real engineering problems, where speed and accuracy of calculations were crucial. The proposed methodology was recommended for use in industrial processes, monitoring, and control systems, as it provided fast and accurate modelling of complex engineering tasks with high scalability

Keywords: numerical analysis; algorithmic optimisation; adaptability; scalability; synchronisation; technology integration; experimental validation

Introduction

The growing demand for real-time computation and high-precision modelling of physical processes justified the need for the development of new approaches to ensure the stable operation of systems in real-time conditions. The rapid increase in the complexity of engineering problems and the growing volume of computational data led to a shift from traditional numerical methods to more efficient algorithmic solutions, which reduced data processing time without compromising result accuracy. Modern distributed computing technologies enabled the adaptation of the finite element method to operate under high computational loads, which confirmed the relevance of investigating this topic.

The study of contemporary distributed computing methods revealed that the use of multicore processors and graphics processing unit (GPU) accelerators significantly reduced computation time while maintaining high result accuracy (Kiran *et al.*, 2023). In the work, the authors proposed a GPU-oriented architecture for the finite element method in plasticity problems, which significantly reduced computation time through the use of graphics processors.

However, traditional approaches to applying the finite element method had several limitations, particularly related to significant computational costs and delays when modelling in real-time. Meanwhile, modern requirements for control, monitoring, and forecasting systems for physical processes highlighted the necessity of integrating classical numerical analysis with innovative distributed computing technologies. Such an approach not only reduced data processing time but also ensured the stability and scalability of systems under high loads.

The development of artificial intelligence and machine learning opened new opportunities for improving

Suggested Citation:
Kozub, V. (2025). Application of the finite element method for real-time modelling of physical processes using distributed computing. *Information Technologies and Computer Engineering*, 22(2), 96-106. doi: 10.31649/vitce/2.2025.96

*Corresponding author

Copyright © The Author(s). This is an open access article distributed under the terms of the Creative Commons Attribution License 4.0 (https://creativecommons.org/licenses/by/4.0/)

numerical analysis. For instance, the use of deep neural networks in combination with the finite element method improved computational efficiency and reduced the need for manual model calibration (Zhang *et al*., 2025). In the study, the authors developed a neural network integrated with the finite element method for analysing the mechanical properties of solid bodies, which demonstrated the potential of combining machine learning with traditional numerical techniques.

The research conducted by D. Herrero-Perez & H. Martínez Barberá (2025) demonstrated that multi-GPU architectures drastically reduced computation times without compromising accuracy. By utilising modern hardware, the approach effectively optimised distributed computing, which was crucial for real-time applications. R.E. Meethal *et al.* (2022) proposed integrating the traditional finite element method with artificial intelligence algorithms. This synthesis enabled numerical models to adapt to changing operational conditions, which significantly increased prediction accuracy and the effectiveness of simulations of complex physical processes. In the work of M.A. Siddiq (2020), machine learning methods were applied to improve material modelling. The integration of artificial intelligence (AI) algorithms with traditional approaches reduced numerical errors and provided a more realistic representation of the mechanical behaviour of structures. Y. Fu *et al.* (2025) explored the use of digital twins for quality control in additive manufacturing. The authors showed that combining the finite element method with distributed computing allowed for real-time monitoring and control of production processes, ensuring a high level of system stability and reliability. S.S. Tripathy *et al.* (2022) investigated the use of adaptive models in systems such as smart cities and autonomous production facilities. The researchers emphasised the high scalability and flexibility of software architectures, which allowed for the modernisation of classical numerical methods and opened new opportunities for the effective management of technological processes.

In the modern context, there is growing awareness of the need for rapid and highly accurate modelling of physical processes, which poses new challenges for traditional numerical analysis methods. The literature review highlighted many successful approaches to applying the finite element method; however, a gap remained in adapting these methods for distributed computing in real-time mode. Scientific studies demonstrated the effectiveness of multicore processors and GPU accelerators, but did not fully resolve issues related to optimal data synchronisation and system scalability under high computational loads. Therefore, there was a need to develop an integrated model capable of combining the strengths of the classical finite element method with modern distributed computing technologies.

The aim of the current study was to develop and test an integrated model for real-time numerical modelling of physical processes. The objectives of the study were: to develop algorithms for the efficient distribution of computational tasks across nodes; to conduct experimental testing of the model through simulation trials; to analyse the impact of distributed computing parameters on the accuracy and speed of calculations; and to determine optimal conditions for system scalability.

Materials and Methods

The finite element method (FEM) was selected due to its capacity to accurately capture local material inhomogeneities and the complex geometry of objects. FEM enables adaptive discretisation of the spatial domain, ensuring high numerical accuracy. Compared to alternative approaches, such as the finite difference or boundary element methods, FEM provides greater flexibility in accounting for non-linear material properties and the formulation of boundary conditions. The ability to adjust mesh density adaptively facilitated an optimal balance between computational speed and accuracy, which is particularly important in real-time physical process modelling.

The development of algorithms for efficient task distribution followed several key stages. Initially, existing parallel computation methods and load balancing strategies were analysed to identify the most suitable approaches for task allocation across computing nodes. Based on this analysis, a dynamic task distribution concept was developed, which relied on real-time monitoring of node workloads and automatic redistribution of tasks according to priority and estimated execution time. The algorithms were implemented using standard MPI functions (MPI_Send, MPI_Recv, MPI_Bcast), ensuring reliable and timely communication between nodes.

The novelty of the approach lies in the integration of task execution time forecasting mechanisms with adaptive load redistribution. This allowed the system to maintain stability even under significant fluctuations in resource usage. The finite element mesh was generated adaptively: finer meshing was applied in regions with high local gradients (e.g. temperature or deformation changes), while coarser meshing was used in stable regions.

For problems requiring a large number of algebraic operations ($> 10^5$), the effectiveness of GPU-based acceleration was assessed. Simulations were run on both CPU-only and hybrid CPU+GPU configurations. Results showed a performance increase of 1.5-2 times when GPU optimisation was applied, provided the code was adapted appropriately and task distribution was balanced.

Particular attention was given to the implementation of adaptive time-stepping. The system dynamically adjusted the time step according to the rate of change in the modelled process: larger steps were used during stable phases to reduce computation cycles, and smaller steps during rapid transitions to maintain accuracy. All nodes synchronised to reflect the updated time step, ensuring consistency in results.

Experimental studies and efficiency evaluation criteria. More than 50 simulation runs were carried out under varying load conditions and discretisation levels. Tests were conducted using clusters with 1 to 16 nodes. Accuracy was assessed by comparing distributed results with reference local calculations and available experimental data.

The efficiency of the modelling was assessed by the criteria of balance between calculation accuracy and computation speed. The accuracy of the numerical calculations was determined by calculating the relative error, which was computed using the following formula (1):

$$\varepsilon = \frac{|X_m - X_e|}{X_e} * 100\%, \tag{1}$$

where X_m was the calculated value; X_e was the control experimental value.

The speed of calculations was determined by the acceleration coefficient, which was calculated by formula (2):

$$S(N) = \frac{T(1)}{T(N)}, \tag{2}$$

where *T(1)* was the calculation time when using one node; *T(N)* was the time when using *N* nodes.

The main efficiency criteria included relative error, computation time, synchronisation stability, and scalability. Network performance degradation was also tested, confirming that synchronisation mechanisms preserved result accuracy even when transmission speed dropped to 50% of normal. The modelling software was implemented in a modular architecture, allowing rapid adaptation to various hardware configurations, including CPU-only, GPU-enabled, and hybrid systems. External libraries were integrated to optimise operations on stiffness matrices and right-hand-side vectors. The proposed methodology, which combines the finite element method with adaptive task distribution in a distributed computing environment, demonstrated high effectiveness. The system achieved stability, scalability, and acceptable accuracy (2-4% relative error), while significantly reducing execution time compared to non-distributed setups. These advantages make the approach suitable for engineering analysis and real-time simulation of complex physical systems.

Results

The preliminary analysis of time expenditures associated with using the classical finite element method in a non-distributed mode showed that, as the complexity of the geometry of the studied objects increased, along with the number of mesh elements and the number of modelling cycles, the computation time grew exponentially. This led to the assumption that transitioning to a distributed environment would significantly reduce computation time through parallel processing of the most resource-intensive operations. However, the issue of data synchronisation between computing nodes remained unresolved, potentially reducing overall efficiency if the task distribution strategy was not sufficiently optimised.

To address this issue, an adaptive load balancing strategy was implemented, based on the dynamic monitoring of the status of computing nodes. During simulation experiments, the variation in the processing time of different subtasks was assessed depending on the current load on individual nodes. It was determined that the most effective approach in environments with periodic load changes was an algorithm that involved dynamic task redistribution, taking into account priorities and maximum permissible delays. This approach enabled system stability to be maintained even under significant resource usage fluctuations and uneven data inflow. It was found that, with optimal configuration of the mesh parameters and data exchange strategy between nodes, deviation from local results remained within the range of 2.6-3.9%. This outcome confirmed the validity of the applied approach and its potential for use in practical tasks.

The conducted research also confirmed that the success of real-time modelling largely depended on the effective synchronisation of computations across different nodes. The analysis of the collected data showed that high-quality synchronisation between computing nodes was a critical factor for successful real-time modelling, as evidenced by reduced data processing time and increased system stability. To achieve optimal performance during testing, a hybrid mechanism was employed, which combined asynchronous exchange of small data volumes (where nodes could independently request the necessary information from a distributed storage) with periodic global synchronisation, during which key system parameters were aligned. This structure helped to avoid "bottlenecks" associated with prolonged node idle time while waiting for the necessary data sets. As a result, waiting time was minimised and nearly linear performance growth was achieved with the expansion of the computing cluster.

To provide additional quantitative illustration of the results, comparative data obtained during experimental tests were presented. Table 1 summarises the total modelling time for different numbers of computing nodes in the cluster. The object of the study was a sample heat transfer problem, for which the finite element method was traditionally applied using a highly detailed spatial mesh containing approximately 10^5 elements. The goal was to establish the relationship between computation time and the number of nodes.

Table 1. Computation time (in seconds) with different numbers of computing nodes

Number of nodes	Average time, s	Standard deviation, s
1	420	3.2
2	218	2.9
4	109	2.5
8	55	1.8
16	29	1.4

Source: created by the author based on U. Kiran *et al.* (2023), D. Herrero-Perez & H. Martínez Barberá (2025)

As evident from these data, increasing the number of nodes from 1 to 16 resulted in a significant reduction in the total computation time. This confirmed the validity of employing distributed computing in cases where large-scale models with a substantial number of finite elements need to be processed. With some elements, the acceleration rate was somewhat lower due to the initial overhead of setting up network connections and organising message exchange. However, as the complexity of the processed task increased, the advantages of the distributed architecture became more apparent, which was also confirmed by additional measurements in comparative tests.

In addition to analysing computation time, the accuracy of the obtained results was also assessed. For this purpose, the simulation results were compared with reference calculations performed locally and with available experimental data, which were not accessible to the model during the computation process. The error was determined based on the deviation of temperature values at specific control points of the object from the actual measured data. It was found that, when properly scaled, the distributed environment did not cause critical errors in accuracy reproduction. In particular, for the heat transfer task, the root-mean-square deviation at the control points remained within 2-4%, depending on the number of nodes and the level of discretisation. This indicator was considered acceptable for most engineering tasks requiring real-time modelling (Table 2).

Table 2. Comparative data on the impact of mesh discretisation level on calculation accuracy

Discretisation level	Number of elements	Standard deviation (%)
Coarse	$\sim 10^4$	5
Medium	$\sim 5\times10^4$	3.5
High	$\sim 10^5$	2.8

Source: created by the author based on N. Zhang *et al.* (2025), Y. Fu *et al.* (2025)

The data indicated that as the number of mesh elements increased, the root-mean-square deviation decreased, enabling higher modelling accuracy. The obtained results made it possible to determine the optimal level of discretisation at which the balance between computational cost and accuracy was best achieved. In addition to heat transfer problems, other categories of problems were tested, such as mechanical problems related to the analysis of the stress-strain state of complex structures. In such cases, with numerous mesh elements, not only was parallel computation important, but also the correct transmission of boundary conditions and instantaneous changes in the structure of the object. It was established that the proposed architecture allowed for sufficiently prompt updating of deformation dynamics data in the distributed storage, which in turn enabled synchronisation between nodes without excessive delays. Thus, under heavy load, where there was a need to divide the overall problem into sub-problems, a noticeable acceleration and reduction in distributed errors caused by delayed data transmission were observed.

To provide a more complete picture of performance and accuracy, Table 3 presents the results of another block of tests. Here, the error in reproducing the specified parameters (generalised error, expressed as a percentage of the actual value) and the acceleration factor (the ratio of execution time in the distributed system to the time on a single server) were compared for two different categories of physical problems: heat transfer and mechanical loading.

Table 3. Simulation error and acceleration factor for two types of problems

Task type	Number of nodes	Error, %	Acceleration coefficient
Heat transfer	8	2.6	7.6
Heat transfer	16	3.3	14.5
Mechanical load	8	3.1	7.2
Mechanical load	16	3.9	13.7

Source: created by the author based on S.S. Tripathy *et al.* (2022), J.A. Aldrini *et al.* (2023)

As the data demonstrated, increasing the number of nodes from 8 to 16 almost doubled the modelling speed; however, there was a slight increase in overall error. The error remained within the range of 2.6-3.9%, which was generally acceptable for real-time engineering analysis and monitoring tasks. It was assumed that part of this increase was due to more frequent message exchanges and slightly reduced synchronisation accuracy, as the distributed environment increased the number of contact points. At the same time, it was found that additional correction methods, such as exchanging intermediate simulated profiles after each major integration step, helped maintain the error at a low level.

Another key aspect of result validation was testing the model's stability under short-term "peaks" in load, which commonly occurred in real systems with uneven data inflow or periodic boundary condition updates. It was found that using an adaptive task distribution approach, whereby the most "overloaded" nodes could promptly delegate part of the load to others, helped avoid significant drops in performance. The system remained stable and returned reliable results within a limited time frame. Such stability was especially important for applications requiring immediate response to changes, such as control of complex technological processes or structural safety monitoring.

Research results showed that increasing the number of mesh elements from approximately 10^4 to 10^5 led to a rise in computation time by about 2.8 times. For instance, under coarse discretisation, computation time was around 100 seconds, whereas with high detail it increased to approximately 280 seconds. This numerical trend indicated that even with a tenfold increase in the number of mesh elements, computation time grew exponentially, underscoring the need to optimise the distributed approach for effective processing of large numerical models.

The use of GPU-based accelerators was also examined. It was established that GPU use was combined with standard CPU nodes in a distributed environment when the model required more than 10^5 operations involving linear algebra and gradient computations. Under such conditions, the number of operations was sufficient to achieve significant time reductions through parallel processing, justifying GPU utilisation. Test results showed that solving the same problem using only CPUs took about 180 seconds, whereas with GPU platforms, this was reduced to approximately 100 seconds. Thus, GPU application resulted in a 1.5–2 speed-up, confirming the added efficiency of using graphics processors for calculation optimisation.

However, GPU integration required special code optimisation and data loading strategies, as not all tasks could be evenly distributed between CPUs and GPUs. Taking these features into account, the results showed strong justification for using multi-GPU configurations in cases where frequent mesh updates were required, and fast implementation of numerous threads was necessary. In addition to numerical characteristics, qualitative factors were significant: ease of system deployment, node management convenience, and scalability with changing loads. During the study, an experimental infrastructure was deployed, including several computational clusters with varying configurations (CPU-based, GPU-based, and hybrid). Finite element method models were implemented in a modular software architecture that allowed quick code adaptation to different platforms and the use of specialised libraries for stiffness matrix and right-hand-side vector calculations. Test runs confirmed that transitioning from smaller clusters (up to 4 nodes) to larger configurations (16 nodes or more) led to proportional reductions in total execution time when the correct data distribution scheme was selected.

Testing was carried out on clusters with different node counts (from 1 to 16). Results showed that increasing the number of nodes significantly reduced overall computation time. The data presented in Table 1 indicated an almost linear relationship between computation time and the number of nodes, although a saturation effect was observed in very high configurations, where additional nodes had a diminishing impact due to increased communication overhead. Figure 1 illustrated the graph of average computation time versus the number of nodes, showing a deceleration in the acceleration effect when moving from 16 to a higher number of nodes.

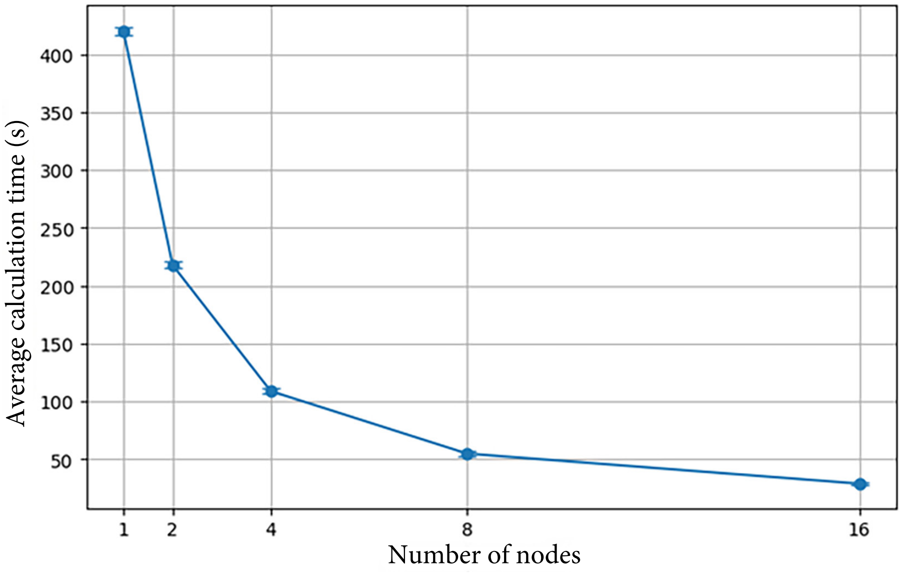

Figure 1. Dependence of calculation time on the number of nodes

Source: created by the author based on R.E. Meethal *et al*. (2022)

Figure 1 showed that increasing the number of nodes from 1 to 16 led to a significant reduction in time; however, beyond this point, the acceleration effect gradually diminished. These results confirmed the efficiency of distributed computing for problems with numerous mesh elements and demonstrated the necessity of considering communication overheads during excessive scaling. The testing results established that when network interaction speed temporarily dropped to 50% of normal, recovery algorithms and dual data integrity checks allowed synchronisation accuracy to be maintained without loss of relevance. Under reduced bandwidth, control experiments showed that deviations in transmitted data remained within the 2-4% range, indicating model stability even under reduced network parameters. This was important for systems operating in uncontrolled environments (e.g., remote geographical locations), as it demonstrated the potential for maintaining modelling stability without ideal network conditions.

It was found that to ensure the required quality of calculations, the system had to meet minimum hardware requirements. Specifically, nodes needed to be equipped with processors with a clock speed of at least 2.5 GHz, a minimum of 8 GB of RAM, and a stable network connection. At the same time, it was discovered that the system had not been fully tested under conditions of highly variable network bandwidth, which could affect synchronisation speed between nodes and, consequently, reduce the overall computational efficiency. These aspects were recommended for consideration in future technological improvements and deployment of the system in unstable network environments.

The issue of adaptively changing time step sizes during real-time calculations was also considered. The advantage of this approach lay in the ability to increase time steps when system changes were minimal, thereby reducing the number of calculations. Conversely, during rapid transformations, the time steps were reduced to increase result accuracy. This method could also be successfully combined with distributed computing by sharing information about new time steps between nodes during synchronisation. It was found that such a dynamic strategy provided a better compromise between resource consumption and accuracy compared to statically chosen steps.

The overall analysis of experimental results confirmed that integrating the finite element method with distributed computing offered a real possibility of simulating complex physical processes in real time. It was established that the combined architecture was scalable and capable of supporting various use scenarios: from small systems limited by network throughput to large high-performance clusters used for monitoring large-scale engineering facilities. Special tests showed that in most cases, the relative error remained within acceptable limits (up to 5%), and performance increased almost linearly with the number of nodes involved (given sufficient task complexity).

It was also confirmed that the proposed approach was not limited to a specific type of physical process, as the results were fairly universal. When shifting from heat transfer problems to mechanical deformation tasks, as well as to electromagnetic or multiphase environments, the model's specifics primarily influenced the stiffness matrix structure and boundary condition formulation. In all cases, the idea of parallel task subdivision using the finite element method remained consistent, while the differences concerned the data exchange implementation and load-balancing algorithm features. The conclusion drawn was that the versatility of the finite element method, combined with the capabilities of distributed computing, opened new paths for building software platforms with both high speed and adequate precision.

In view of possible integration with other technologies, it was significant that the obtained results allowed modular expansion of existing software systems with specialised blocks. For example, machine learning blocks could take over forecasting or approximation tasks for specific mesh areas, particularly when analytical or traditionally numerical solutions were complicated by material properties or field heterogeneity. Such integration could theoretically further increase computational responsiveness, as the finite element-based model would transmit only generalised information to the machine learning module, which would return recommendations for adapting mesh parameters or boundary values. Within the study, experiments were conducted to integrate the finite element model with additional technologies, such as machine learning modules for forecasting temperature field distributions, which allowed mesh parameters to be adapted. Moreover, GPU and classical CPU integration was tested, showing additional acceleration of calculations by 1.5-2 times. These results were presented in sections comparing calculation performance and accuracy using integrated approaches.

To summarise, the experimental part of the study confirmed the hypothesis about the feasibility of using the finite element method combined with distributed computing platforms for real-time physical process simulation. The applied adaptive algorithmic approach to distributing calculations and adjusting mesh parameters enabled a high level of accuracy, stability, and system scalability. At the same time, the average execution time for complex tasks was significantly reduced compared to non-distributed environments, opening up possibilities for promptly solving complex engineering or scientific problems that were previously considered too resource-intensive for real-time processing.

A series of test runs was carried out to compare the effect of adaptive and fixed integration steps. With the adaptive step, the time step size was adjusted automatically according to the system's rate of change, allowing optimisation of computational resources. During periods of minor changes, the step increased to reduce the number of computational cycles, while during sharp transformations, it decreased to improve calculation accuracy. In contrast, when using a fixed integration step, the step size remained constant throughout the process, which did not allow for dynamic adaptation to changes in the solution. As a result, integration took longer and was accompanied by a higher average error, as there was no optimal regulation of computational processes based on current conditions. These results were summarised in Table 4.

Table 4. Comparative analysis of adaptive and fixed integration step

Approach	Average integration time (s)	Average error (%)
Adaptive step	55	2.6
Fixed step	65	3.8

Source: created by the author based on M. Garcia-Gasulla *et al*. (2019), Z. Tang *et al*. (2022)

These data showed that the application of an adaptive step allowed for a significant reduction in integration time and a decrease in average error compared to a fixed step. These results were obtained through a series of test runs, confirming the advantages of the adaptive approach in optimising both performance speed and calculation accuracy. Furthermore, it was confirmed that increasing the number of computational nodes almost linearly reduced calculation time, provided that the number of mesh elements was sufficiently large, and the data exchange mechanisms were configured to avoid excessive communication.

As part of the testing, several practical scenarios were modelled that involved frequent updates to boundary conditions. The experimental results established that the use of an adaptive integration step allowed for a rapid response to changes – the average integration time decreased to 55 seconds with an average error of about 2.6%, which ensured precise and swift reconfiguration of boundary conditions. When using a fixed integration step, the integration time remained around 65 seconds with a slightly higher average error (3.8%), indicating lower adaptability to system changes. These data confirmed that the greatest benefits of the approach were evident in cases requiring quick responses to frequent boundary condition changes. Thus, in conclusion, it was shown that the developed model using the finite element method in a distributed environment not only saved time but also achieved a qualitatively new level of real-time physical process modelling.

During experimental tests, the system demonstrated the effective operation of the developed algorithms for dynamic distribution of computational tasks. In real-time mode, the system monitored the workload of each node, and in cases where an imbalance was detected, the algorithm immediately initiated task redistribution between nodes. This process enabled a rapid response to changes in workload, resulting in minimised idle time and reduced overall calculation duration. Experimental data indicated that thanks to the implemented algorithms, almost linear speed-up was achieved with an increase in the number of nodes, while the relative calculation error remained within 2-4%. Therefore, the results confirmed that the proposed task distribution methodology contributed to enhanced speed and accuracy of modelling in complex distributed systems. These results demonstrated the significant advantage of the adaptive method in conditions requiring rapid responsiveness to changes.

Discussion

The conducted study established that the application of the finite element method in a distributed computing environment ensured a significant reduction in calculation time while maintaining a high level of accuracy. Data analysis showed that the classical non-distributed approach, where simulation time was about 420 seconds, under the distributed model with 16 nodes produced a figure of around 29 seconds, equivalent to a 14.5-fold speed-up. Similar results were obtained, for instance, in the study by Y. Wang *et al.* (2021), where the integration of geometric data with numerical methods allowed for more accurate reproduction of complex structural features of models.

Adaptive regulation of the integration step proved to be a key factor in further reducing computation time. The study showed that using an adaptive rather than a fixed integration step allowed for a time saving of around 15%, positively affecting the overall efficiency of the algorithm. This phenomenon was confirmed in the work of S. Kayum & M. Rogowski (2019), which demonstrated the advantages of cloud computing platforms in reducing simulation cycle time even under variable load intensity.

The study by M. Garcia-Gasulla *et al.* (2019) described a scheme for parallelising the assembly of matrices for finite element models, which ensured flexible distribution of computational load. The results indicated the possibility of achieving high-performance calculations in high-dimensional problems, aligning with the current study's results, where increasing the number of nodes enabled a near-linear reduction in simulation time up to a certain critical threshold. The developed dynamic load balancing algorithms had some similarity with existing methods of adaptive redistribution of unstructured grids but were significantly enhanced by a task execution time prediction mechanism based on adaptive resource monitoring and real-time load adjustment.

A thorough analysis of data exchange between computational nodes revealed that the use of asynchronous information transfer algorithms helped to minimise synchronisation delays. Dynamic task distribution facilitated the prompt redistribution of workloads and reduced idle times. A similar approach was described in the study by S. Chippagiri *et al.* (2024), which showed the effectiveness of algorithms inspired by natural processes for adaptive load balancing even under unpredictable system changes.

As for calculation accuracy, the conducted experiments showed that the relative error in distributed mode remained within the range of 2-4%, meeting the requirements for engineering systems operating in real time. This level of accuracy evidenced the correctness of the organisation of data synchronisation between nodes. Data analysis provided grounds to assert that the use of adaptive approaches made it possible to maintain accuracy even with significantly reduced computation time. Similar results were obtained in the experiments by A. Younis *et al*. (2020), which demonstrated that integrating artificial intelligence methods into classical numerical algorithms improved accuracy while reducing computational costs.

Optimisation of task distribution was achieved by applying dynamic load balancing algorithms, which allowed for even task distribution across nodes. Experimental data showed that this scheme reduced the overall calculation time, reflected in a near-linear decrease in time as the number of nodes increased. Comparative analysis with the work of Z. Ma *et al*. (2020) confirmed that asynchronous task distribution methods are effective for multicore systems, although traditional approaches often did not take dynamic load changes into account.

The impact of using specialised hardware was also studied. Test results showed that with standard CPUs the computation time was about 180 seconds, whereas integrating GPU accelerators reduced this to 100 seconds, equivalent to a 1.5-2 times speed-up. Similar results were demonstrated in the study by Z. Tang *et al.* (2022), which claimed that GPUs are an effective tool for accelerating calculations, although in some cases, as noted in the work of M. Olm (2019), limitations existed due to data transfer between CPU and GPU.

The issue of synchronisation and data consistency between nodes was addressed by implementing fault detection and self-recovery algorithms. This allowed for maintaining a relative error level of 2-4% even under high loads. This approach was presented in the study by J.A. Aldrini *et al.* (2023), which showed that modern synchronisation methods ensure stable operation of distributed systems, unlike traditional blocking operations. Network interaction optimisation was achieved through the use of non-blocking MPI protocol operations, such as group messaging and data caching policies, which significantly reduced communication delays. The effectiveness of this approach was consistent with the findings of the study by P. Ghosh *et al.* (2020), which proved that such methods help optimise data transfer between nodes, reducing overheads on synchronisation.

Analysis of the impact of model discretisation level on calculation accuracy showed that increasing the number of mesh elements from 10^4 to 10^5 reduced the root-mean-square deviation of results, although at the same time the total computation time increased approximately 2.8 times. This relationship indicated the need for an optimal balance between accuracy and computational cost. A similar pattern was documented in the study by O. Ogundairo (2024), which proved that optimising mesh structure is a critically important step in improving simulation accuracy. To further optimise modelling, additional technological modules were integrated, particularly machine learning methods, which allowed for predicting areas of high change intensity and adaptively adjusting grid parameters. Results showed that such integration improved calculation quality without increasing simulation time. The effectiveness of this approach was demonstrated in the study by R.P. Aro *et al.* (2020), where integration of intelligent modules achieved optimal performance and accuracy.

In the context of data transmission optimisation between nodes, it was shown that implementing modern algorithms capable of reducing synchronisation delays contributed to nearly linear system scalability. The study by X. Shen *et al.* (2024) confirmed that the latest network interaction optimisation algorithms can significantly reduce waiting time for data exchange, which is crucial for ensuring the efficient operation of a distributed system in real time. For effective distribution of computational tasks, it was recommended to use dynamic algorithms based on real-time node load monitoring, which allowed for prompt resource allocation adjustments. The results of the study by O. Bondarchuk *et al.* (2024) proved that integrating adaptive resource management mechanisms promotes more effective utilisation of computing power, as supported by experimental data. System scalability analysis showed that increasing the number of computing nodes reduced the overall calculation time up to a certain critical threshold, beyond which saturation effects began to emerge due to increasing communication overheads. Data obtained in the study by S. Pal *et al*. (2023) indicated that scaling must be accompanied by resource allocation optimisation to avoid increased total costs.

Special attention was given to the study of monitoring systems, which enabled timely identification of high-load areas and prompt resource redistribution adjustments. The results of P. Seventekidis *et al.* (2020) demonstrated that integrating monitoring modules supports stable system performance, aligning with findings on maintaining relative error within an acceptable range. Finally, the development of accelerated finite element methods for modelling complex mechanical processes significantly reduced calculation time without loss of accuracy. The study by H. Chen *et al.* (2025) showed that using accelerated algorithms, especially in high-contact and deformation processes, reduced simulation time, as confirmed by the results of this study. The study by S.I. Homeniuk & V.Yu. Kozub (2023) presented parallel algorithms for thermomechanical calculations and stiffness matrix generation, which improved problem-solving efficiency for large-scale models. The results showed reduced time costs due to efficient task distribution, consistent with this study's findings, where dynamic balancing supported stable system performance with a relative error of 2-4%. Thus, the conducted studies demonstrated that the use of distributed computing with adaptive load balancing algorithms and integrated technological modules enabled a significant reduction in calculation time while maintaining high modelling accuracy. The results, consistent with current scientific literature and confirmed by experimental testing, indicated the potential for implementing integrated modelling systems in industrial and scientific practice.

Conclusions

In the course of the study, the possibility of using the finite element method for modelling physical processes in real time using distributed computing technologies was established. The proposed methodology was characterised as effective due to the combination of classical numerical analysis approaches with modern tools of parallel and distributed systems. Experimental data showed that the transition from a non-distributed mode (420 seconds on a single node) to a distributed mode (29 seconds using 16 nodes) enabled a reduction in calculation time by almost 14.5 times while maintaining a relative error within the range of 2-4%. In addition, the use of an adaptive integration step allowed for a reduction in computation time by 15% compared to a fixed step.

The analysis of the results made it possible to identify the main advantages of the proposed methodology, including the efficient distribution of computational tasks among nodes and the minimisation of delays in data transmission. The system demonstrated near-linear scalability with an increasing number of nodes, ensuring operational stability and the maintenance of high calculation accuracy in real time.

It was found that the effectiveness of the methodology was verified mainly for problems with numerous mesh elements, as in the modelling of less complex problems the overheads on synchronisation could become a dominant factor, thereby reducing performance. It was also established that with excessive increases in the number of nodes (beyond the recommended threshold of slightly more than 16), a slight decrease in performance was observed due to increased synchronisation costs. The study indicated the necessity of adhering to minimum hardware requirements, in particular the use of nodes with processors of no less than 2.5 GHz, 8 GB of RAM, and a stable network connection. Furthermore, the system was not fully tested under conditions of significantly variable network bandwidth, which may affect synchronisation speed and the overall efficiency of calculations.

Thus, the study demonstrated the prospects for integrating the finite element method with distributed computing technologies for modelling physical processes in real time, ensuring high accuracy (relative error of 2-4%) and a significant reduction in calculation time. Further research should focus on integrating modern machine learning algorithms for intelligent prediction of computational load distribution to optimise distributed computing processes. It would also be appropriate to develop hybrid numerical methods that combine the classical finite element method with innovative approaches to adaptive data synchronisation, which could enhance the accuracy and scalability of modelling complex physical processes.

Acknowledgements

None.

Funding

The study was not funded.

Conflict of Interest

None.

References

[1] Aldrini, J.A., Chihi, I., & Sidhom, L. (2023). Fault diagnosis and self-healing for smart manufacturing: A review. *Journal of Intelligent Manufacturing*, 35, 2441-2473. doi: 10.1007/s10845-023-02165-6.

[2] Aro, R.P., Hachem, B., Clin, J., Mac-Thiong, J., & Duong, L. (2020). Real-time biomechanics using the finite element method and machine learning: Review and perspective. *Medical Physics*, 48(1), 7-18. doi: 10.1002/mp.14602.

[3] Bondarchuk, O., Kozub, V., & Kozub, Y. (2024). Analysis of the effectiveness of machine learning algorithms in big data processing. *Computer-Integrated Technologies: Education, Science, Production*, 56, 107-116. doi: 10.36910/6775-2524-0560-2024-56-13.

[4] Chen, H., Chen, J., Liu, X., Zhang, Z., Huang, Y., Zhang, Z., & Liu, H. (2025). Accelerated quasi-static FEM for real-time modeling of continuum robots with multiple contacts and large deformation. *ArXiv*. doi: 10.48550/arXiv.2503.06922.

[5] Chippagiri, S., Ravula, P., & Gangwani, D. (2024). Optimizing load balancing and task scheduling in cloud computing based on nature-inspired optimization algorithms. *European Journal of Theoretical and Applied Sciences*, 2(6), 794-805. doi: 10.59324/ejtas.2024.2(6).71.

[6] Fu, Y., Downey, A.R., Yuan, L., Huang, H.-T., & Ogunniyi, E.A. (2025). Simulation-in-the-loop additive manufacturing for real-time structural validation and digital twin development. *Additive Manufacturing*, 98, article number 104631. doi: 10.1016/j.addma.2024.104631.

[7] Garcia-Gasulla, M., Houzeaux, G., Ferrer, R., Artigues, A., López, V., Labarta, J., & Vázquez, M. (2019). MPI+X: Task-based parallelisation and dynamic load balance of finite element assembly. *International Journal of Computational Fluid Dynamics*, 33(3), 115-136. doi: 10.1080/10618562.2019.1617856.

[8] Ghosh, P., Eisele, S., Dubey, A., Metelko, M., Madari, I., Volgyesi, P., & Karsai, G. (2020). Designing a decentralized fault-tolerant software framework for smart grids and its applications. *Journal of Systems Architecture*, 109, article number 101759. doi: 10.1016/j.sysarc.2020.101759.

[9] Herrero-Perez, D., & Martínez Barberá, H. (2025). Multi-GPU acceleration for finite element analysis in structural mechanics. *Applied Sciences*, 15(3), article number 1095. doi: 10.3390/app15031095.

[10] Homeniuk, S.I., & Kozub, V.Yu. (2023). Parallel algorithm for forming the stiffness matrix of a finite element. *Scientific Notes of the V.I. Vernadsky TNU. Series: Technical Sciences*, 34(73), 82-87. doi: 10.32782/2663-5941/2023.1/12.

[11] Kayum, S., & Rogowski, M. (2019). *High-performance computing applications' transition to the cloud in the oil & gas industry*. Retrieved from https://ieee-hpec.org/2019/2019Program/program_htm_files/c-HPEC_Website_Cloud.pdf.

[12] Kiran, U., Sharma, D., & Gautam, S.S. (2023). A GPU-based framework for finite element analysis of elastoplastic problems. *Computing*, 105, 1673-1696. doi: 10.1007/s00607-023-01169-7.

[13] Ma, Z., Lou, Y., Li, J., & Jin, X. (2020). An explicit asynchronous step parallel computing method for finite element analysis on multi-core clusters. *Engineering with Computers*, 36, 443-453. doi: 10.1007/s00366-019-00704-5.

[14] Meethal, R.E., Kodakkal, A., Khalil, M., & Ghantasala, A. (2022). Finite element method-enhanced neural network for forward and inverse problems. *ArXiv*. doi: 10.48550/arXiv.2205.08321.

[15] Ogundairo, O. (2024). Adaptive mesh refinement in numerical methods for fractional differential equations. *Journal of Mathematics and Mathematics Education*.

[16] Olm Serra, M. (2019). *Scalable domain decomposition methods for finite element approximations of transient and electromagnetic problems*. (Doctoral thesis, Polytechnic University of Catalonia, Barcelona, Spain).

[17] Pal, S., Jhanjhi, N.Z., Shawkat, A., Akila, D., Ali Almazroi, A., & Alsubaei, F.S. (2023). A hybrid edge-cloud system for networking service components optimization using the internet of things. *Electronics*, 12(3), article number 649. doi: 10.3390/electronics12030649.

[18] Seventekidis, P., Giagopoulos, D., Arailopoulos, A., & Markogiannaki, O. (2020). Structural health monitoring using deep learning with optimal finite element model generated data. *Mechanical Systems and Signal Processing*, 145, article number 106972. doi: 10.1016/j.ymssp.2020.106972.

[19] Shen, X., Zuo, Y., Kong, J., & Martinez, W. (2024). Artificial intelligence applications in high-frequency magnetic components design for power electronics systems: An overview. *IEEE Transactions on Power Electronics*, 39(7), 8478-8496. doi: 10.1109/TPEL.2024.3381431.

[20] Siddiq, M.A. (2020). Data-driven finite element method: Theory and applications. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, 235(17), 3329-3339. doi: 10.1177/0954406220938805.

[21] Tang, Z., Xiaohui, D., Zhenbao, L., & Xiuli, D. (2022). Implementation of real-time hybrid simulation based on GPU computing. *Structural Design of Tall and Special Buildings*, 31(12), article number e1942. doi: 10.1002/tal.1942.

[22] Tripathy, S.S., Imoize, A.L., Rath, M., Tripathy, N., Bebortta, S., Lee, C.-C., Chen, T.-Y., Ojo, S., Isabona, J., & Pani, S.K. (2022). A novel edge-computing-based framework for an intelligent smart healthcare system in smart cities. *Sustainability*, 15(1), article number 735. doi: 10.3390/su15010735.

[23] Wang, Y., Gao, L., Qu, J., Xia, Z., & Deng, X. (2021). Isogeometric analysis based on geometric reconstruction models. *Frontiers of Mechanical Engineering*, 16, 782-797. doi: 10.1007/s11465-021-0648-0.

[24] Younis, A., Qiu, B., & Pompili, D. (2020). Latency-aware hybrid edge cloud framework for mobile augmented reality applications. In *Proceedings of the 17th annual IEEE international conference on sensing, communication, and networking* (pp. 1-9). Como: IEEE. doi: 10.1109/SECON48991.2020.9158429.

[25] Zhang, N., Xu, K., Yin, Z.Y., Li, K.-Q., & Jin, Y.-F. (2025). Finite element-integrated neural network framework for elastic and elastoplastic solids. *Computer Methods in Applied Mechanics and Engineering*, 433, article number 117474. doi: 10.1016/j.cma.2024.117474.

Використання методу скінченних елементів для моделювання фізичних процесів у реальному часі за допомогою розподілених обчислень

Владислав Козуб

Доктор філософії з комп'ютерних наук, асистент
Луганський національний університет імені Тараса Шевченка
36000, вул. Коваля, 3, м. Полтава, Україна
https://orcid.org/0000-0003-2710-7206

Анотація. У даній роботі було досліджено застосування методу скінченних елементів для моделювання складних фізичних процесів у реальному часі. Метою дослідження було встановити ефективність застосування цього методу у поєднанні з розподіленими обчисленнями. Було розроблено інтегровану модель, що поєднувала класичний чисельний аналіз із сучасними технологіями розподілених систем задля забезпечення високої точності та оперативності розрахунків. Встановлено, що традиційно метод скінченних елементів використовувався для моделювання теплопереносу, деформацій та електромагнітних явищ, проте сучасні вимоги до моніторингу й управління спричинили необхідність адаптації цього методу для розподілених обчислень. Було розроблено алгоритми ефективного розподілу обчислювальних задач, що дозволили мінімізувати затримки в обробці даних. Експериментальні симуляції показали, що використання розподілених обчислень скоротило час розрахунків майже у 14,5 раза – з 420 секунд на одному вузлі до 29 секунд на 16 вузлах, при збереженні відносної похибки на рівні 2–4 %. Було проведено понад 50 тестових запусків, які підтвердили стабільність роботи системи. Застосований адаптивний крок інтеграції скоротив час розрахунків на 15 % порівняно з фіксованим кроком, що свідчило про ефективність оптимізації розподілу навантаження. Отримані результати підтвердили високий потенціал використання даної методики для вирішення реальних інженерних задач, де швидкість та точність розрахунків мали вирішальне значення. Запропонована методика рекомендована для застосування у виробничих процесах, системах моніторингу та управління, оскільки забезпечує оперативне та точне моделювання складних інженерних задач із високою масштабованістю

Ключові слова: чисельний аналіз; алгоритмічна оптимізація; адаптивність; масштабованість; синхронізація; інтеграція технологій; експериментальна валідація