Vol. 6, No. 3, 2024

Artem Mikanov[1], Ihor Farmaha[2], Krzysztof Kurzydlowski[3]

[1] Computer Design Systems Department, Lviv Polytechnic National University, Ukraine, Lviv, S. Bandery street 12, E-mail: artem.mikanov.kn.2021@lpnu.ua, ORCID 0009-0007-0643-3367,
[2] Computer Design Systems Department, Lviv Polytechnic National University, Ukraine, Lviv, S. Bandery street 12, E-mail: ihor.v.farmaha@lpnu.ua, ORCID 0009-0007-3727-2586,
[3] Department of Material and Production Engineering, Faculty of Mechanical Engineering, Bialystok University of Technology, Poland, Bialystok, Wiejska 45c street, E-mail: krzysztof.kurzydlowski@pb.edu.pl ORCID 0000-0003-3875-4820

HIGH-PERFORMANCE COMPUTING METHOD FOR INVESTIGATING HEAT CONDUCTION PROCESSES IN COMPOSITE MATERIALS

Received: November 12, 2024 / Revised: November 20, 2024 / Accepted: November 25, 2024

https://doi.org/

Abstract. This study explores an approach to modeling heat conduction in composite materials using high-performance computing methods. The solution of a complex mathematical problem is decomposed into multiple independent processes through the MPI method to maximize computational efficiency. The model structure accounts for the properties of the composite material and its components. The finite element method (FEM) is applied to study the thermal properties. Computational results demonstrate the high efficiency and accuracy of the proposed approach, confirming its relevance for both scientific research and educational purposes.

Keywords: composite materials, thermal conductivity, finite element method, high-performance computing, parallel computing, MPI, boundary conditions, temperature distribution analysis.

Introduction

Modern composite materials have found widespread applications due to their unique combination of properties, which depend on the structure of the matrix and inclusions. Calculating the thermal conductivity of such materials is a complex task that requires significant computational resources. Traditional approaches often fail to provide the necessary efficiency, complicating the analysis of multicomponent structures [5-7].

In this context, high-performance computing (HPC) opens up new opportunities [3]. Parallel programming methods, such as MPI (Message Passing Interface), optimize computational processes and significantly reduce execution time. Using the finite element method (FEM) makes it possible to account for complex geometry and material heterogeneity and adapt the model to various conditions.

Problem Statement

The primary drawback of existing methods is their limited performance, which hinders the scalability of tasks. This work proposes the use of MPI to solve thermal conductivity problems in composite materials with proportional scaling of computational nodes and tasks.

1. Develop a universal computational model that considers the specifics of composite materials.
2. Implement an efficient calculation of thermal conductivity using FEM and MPI.
3. Analyze the accuracy of the approach for different types of structures.

Main Material Presentation

A composite material consists of a matrix providing overall structure and mechanical properties and inclusions affecting physical characteristics such as thermal conductivity.

The modeling process uses a cubic structure composed of cells made from two materials with different thermal conductivities. The main stages of modeling include:

• Discretization of the domain into finite elements.

• Generation of the structure (random or gradient material distribution).

• Determination of element properties based on the material.

MPI is used to distribute tasks among computational nodes, enabling simultaneous calculation of multiple experiments.

Selection of Tools and Technologies:

FEniCSx

FEniCSx is a library for solving partial differential equations using the finite element method (FEM) with support for parallel computations [2]. It was chosen due to its ability to efficiently solve partial differential equations on complex geometries, support for various mesh element types, and boundary condition flexibility.

MPI

MPI (Message Passing Interface) is employed for parallel computations, ensuring efficient communication between processes on different computational nodes. MPI allows distributing calculations across multiple processes or computers, increasing the speed of complex computations [1].

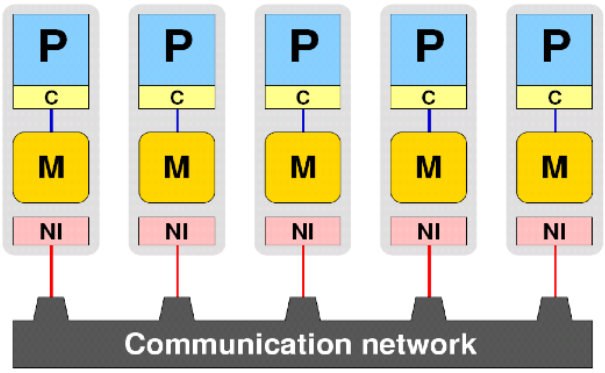

Fig. 1. The principle of MPI

Justification for choice

The choice of FEniCSx for modeling thermal conductivity is motivated by its high solution accuracy, support for parallel computations, and flexibility in task configuration. MPI ensures efficient scaling of computations across large numbers of processors, essential for handling complex models with numerous elements. These tools enable the performance and accuracy needed for solving thermal conductivity problems in composite materials, which is critical for implementing high-performance computations in scientific and engineering research.

Implementation of the Computational Process

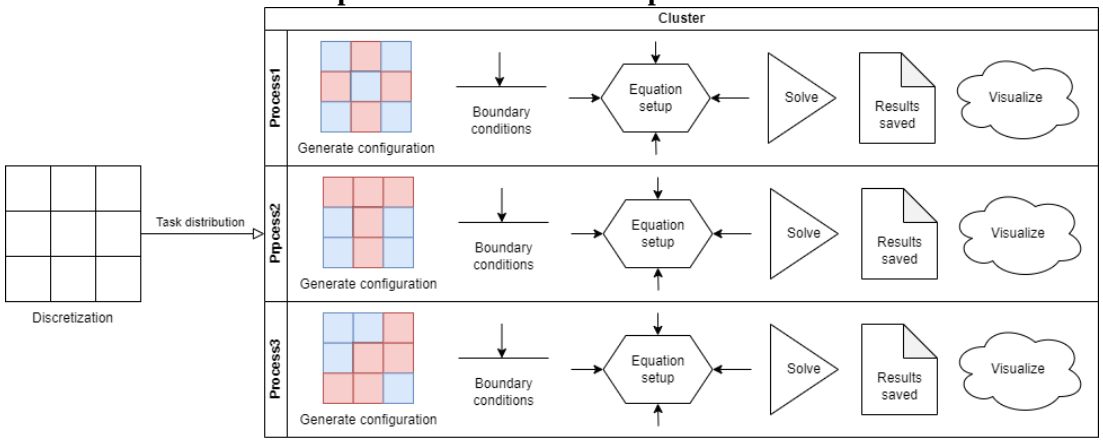

Fig. 2: Schematic representation of the algorithm

Discretization

At this stage, a finite element mesh is constructed. The discretization of the domain is performed using the create_unit_cube function from the fenicsx library [2]:

```
mesh = create_unit_cube(MPI.COMM_SELF, nx, ny, nz,
cell_type=CellType.hexahedron)
```

This function takes the dimensions of the domain for discretization, corresponding to the number of finite elements in three dimensions. It also requires the specification of the element shape, which in this case is chosen as cubic elements.

Task Distribution Among Processes

In this step, predefined experiments are distributed for execution across different processors in the cluster. Maximum computational efficiency on arbitrarily organized processors is achieved by following the "1 process - 1 processor" rule. It is assumed that the number of experiments is divisible by the number of processors to avoid idle time. Uniform task distribution is implemented as follows:

```
mesh = create_unit_cube(MPI.COMM_SELF, nx, ny, nz,
cell_type=CellType.hexahedron experiments_per_rank = num_experiments //
size
remainder = num_experiments % size
start_idx = rank * experiments_per_rank + min(rank, remainder)
end_idx = start_idx + experiments_per_rank + (1 if rank < remainder
else 0)
my_experiments = list(range(start_idx, end_idx))
```

Due to the use of MPI, the value of the rank variable will differ for each process, assigning a unique set of experiment indices to each process [1].

Structure Generation

For generating the configuration, it is assumed that each generated finite element can be assigned a different thermal conductivity value. This value determines the material classification of the element for subsequent analysis.

```
cells_1, cells_2 = [], []
for cell_index in range(c2v.num_nodes):
    if np.random.rand() < ratio:          # Material 1 with probability
of "ratio"
        cells_1.append(cell_index)
    else:                                 # Material 2 in other cases
        cells_2.append(cell_index)

cells_1 = np.array(cells_1, dtype=np.int32)
cells_2 = np.array(cells_2, dtype=np.int32)

# Filling thermal conductivity function
kappa.x.array[cells_1] = kappa1
kappa.x.array[cells_2] = kappa21, cells_2 = [], []
    for cell_index in range(c2v.num_nodes):
        if np.random.rand() < ratio:             # Material 1 with
probability of "ratio"
            cells_1.append(cell_index)
        else:                                    # Material 2 in other
cases
            cells_2.append(cell_index)

    cells_1 = np.array(cells_1, dtype=np.int32)
    cells_2 = np.array(cells_2, dtype=np.int32)
```

```
    # Filling thermal conductivity function
    kappa.x.array[cells_1] = kappa1
    kappa.x.array[cells_2] = kappa2
```

Equation Formulation

The first step in problem formulation is defining boundary conditions. This implementation supports Dirichlet, Neumann, and Robin boundary conditions [4]. The cube's faces are used as boundary regions, with predefined markers assigned to them:

```
boundary_conditions = [
    BoundaryCondition("Dirichlet", marker=1, value=t1, V=V, u=u, v=v),
    BoundaryCondition("Dirichlet", marker=2, value=t2, V=V, u=u, v=v),
    BoundaryCondition("Neumann", marker=3, value=g1, V=V, u=u, v=v),
    BoundaryCondition("Neumann", marker=6, value=g2, V=V, u=u, v=v),
    BoundaryCondition("Robin", marker=6, value=(r, s), V=V, u=u, v=v),
]
```

The weak form of the thermal conductivity problem is constructed:

```
F = kappa * inner(grad(u), grad(v)) * dx - inner(f, v) * dx
```

This formulation is based on the heat conduction equation in weak form:

$$-\kappa\nabla u \cdot \nabla v, d\Omega = \int \Omega f v, d\Omega \quad (1)$$

The previously defined boundary conditions are applied to the variational form:

```
# Separately collect Dirichlet boundary conditions
bcs = [bc.bc for bc in boundary_conditions if bc.type == "Dirichlet"]

# Apply other boundary conditions to the variational form
for bc in boundary_conditions:
    if bc.type != "Dirichlet":
        bc.apply(F)
```

Dirichlet boundary conditions are separated and applied during the mathematical solution process, as FEniCSx provides built-in tools for handling them. The mathematical formulation of Dirichlet conditions is:

$$u = \text{const on } \partial\Omega \quad (2)$$

For Neumann and Robin boundary conditions, the variational form is adjusted accordingly:

```
...
elif bc_type == "Neumann":
    self.bc = value * v * ds(marker)
elif bc_type == "Robin":
    self.bc = value[0] * inner(u - value[1], v) * ds(marker)
...
```

Neumann Boundary Conditions (based on Fourier's law):

$$-\kappa\nabla u \cdot n\hat{} = q \text{ on } \partial\Omega \quad (3)$$

Robin Boundary Conditions:

$$-\kappa\nabla u \cdot n\hat{} = h(u - T\infty) \text{ on } \partial\Omega \quad (4)$$

The resulting problem elements are represented in the form:

$$A \cdot u = L \quad (5)$$

Where A is the assembled stiffness matrix and L is the load vector, incorporating all boundary conditions.

Solution of the System of Equations

The system of equations, formulated as A·u=L [4], is solved using FEniCSx's LinearProblem class. The solver utilizes PETSc options for customizable numerical methods. The implementation is as follows:

```
def solve_problem(a, L, bcs):
```

```
    problem = LinearProblem(a, L, bcs=bcs, petsc_options={"ksp_type":
"cg", "pc_type": "lu"})
    return problem.solve()     self.bc = value
```

The function returns the solution vector u, representing the temperature distribution in the computational domain.

Solution of the System of Equations

The simulation outputs are saved in .xmdf format, which is widely used for 3D structures. The file includes the finite element mesh and computed nodal values (e.g., temperature):

```
With XDMFFile(MPI.COMM_SELF,
f"{save_path}/simulation_results_{experiment_id}.xmdf", "w") as file:
        file.write_mesh(mesh)
        file.write_function(uh)
```

The metadata of the experiment (including its ID, mesh size, fill factor, material thermal conductivities, as well as boundary conditions and their application regions) is stored in .json format. Additionally, throughout all stages, the computation time for each experiment is measured (including the time for material region determination, problem setup, solution, and data saving) and recorded in a .csv file.

The program also provides the ability to visualize the computation results as graphical images of material regions and temperature distribution. The result of each experiment is stored along with the other files.

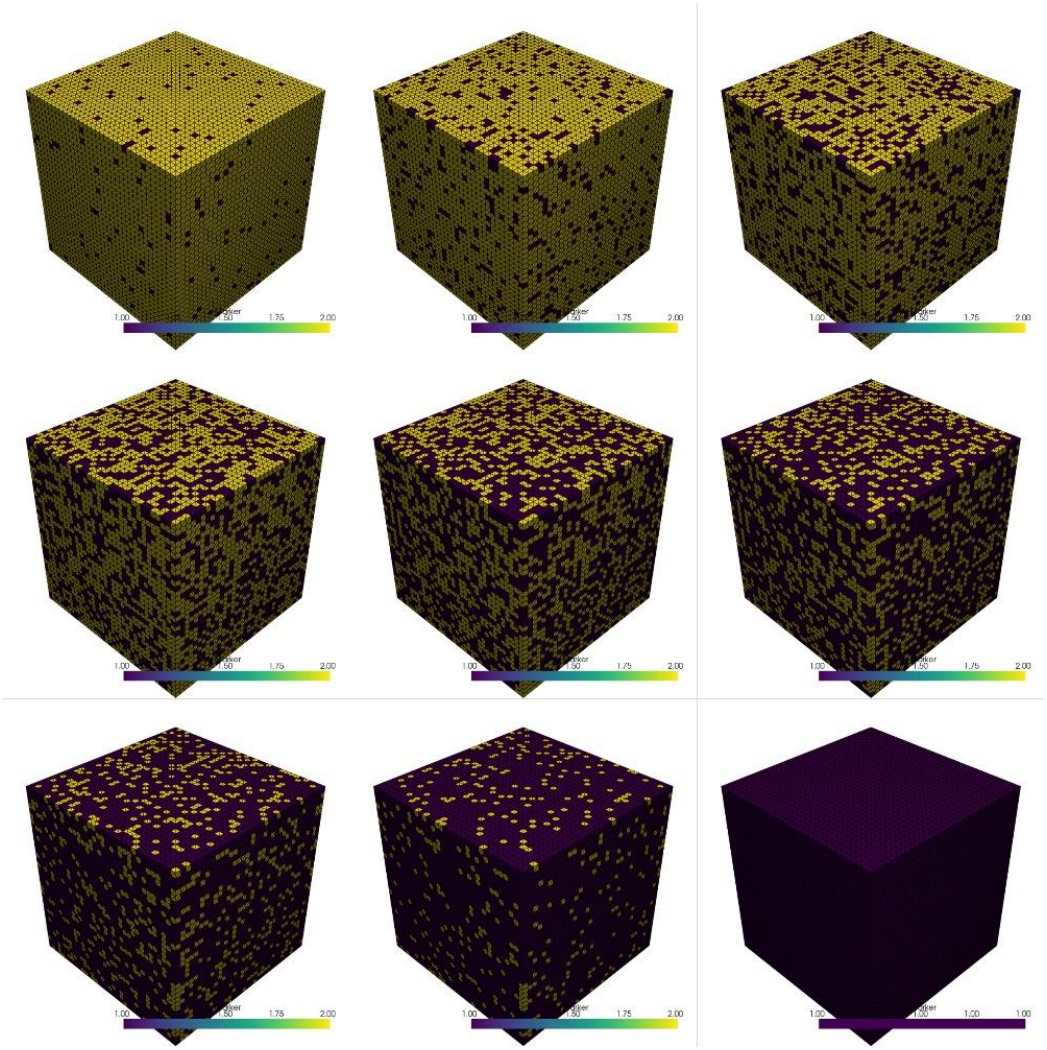

Fig.3.: Composite material configurations with different filling percentages

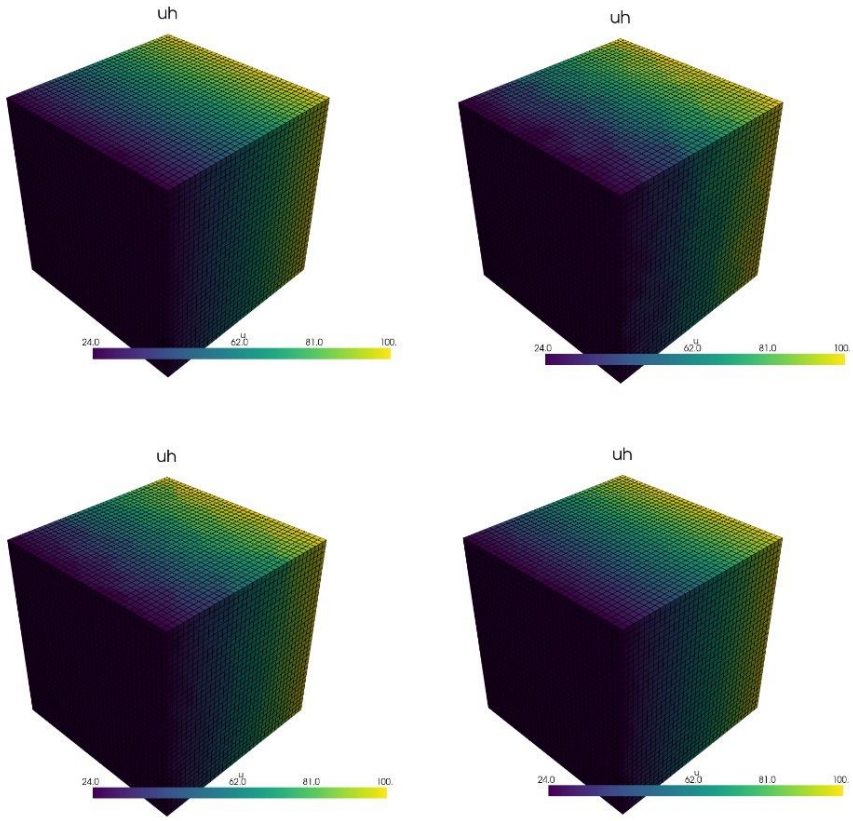

Fig. 4. Temperature distribution for material ratios (3%, 12.5%, 37.5%, 75%).

Conclusions

During the parallel execution of a set of tasks by dedicated processes, each of them measured the time spent on executing its tasks. The overall complexity of the problem was calculated as the sum of the time spent on executing all assigned tasks (conducted experiments). Additionally, a joint time measurement was taken for the entire program execution, from the moment the tasks were distributed to the process pool until the results of all distributed tasks were ready. The efficiency of the parallelization of the computational process was determined as:

$$\mathrm{Ep} = \frac{\mathrm{T1}}{p \cdot \mathrm{Tp}} \tag{6}$$

Here, T1 is the time taken to execute the task on a single processor (which corresponds to the overall complexity of the task), p is the number of processes used, and Tp is the time taken to execute the task using multiprocessing with communication through MPI.

According to the data obtained in the resulting file, the parallelization efficiency for solving this problem was 95.9% when computing 256 experiments for cubic structure models with dimensions of 20x20x20 on a cluster with a total processing power of 16 processor cores.

Furthermore, we observed that the computational time in Finite Element Analysis (FEA) increases non-linearly with the number of finite elements. For example:

- For a 20×20×20 mesh (8,000 elements), the average solve time was 2.4 seconds.
- For a 35×35×35 mesh (42,875 elements), the average solve time increased to 100 seconds.

This corresponds to a 5.36-fold increase in the number of elements, but a 41.67-fold increase in computation time. This disparity highlights the cubic or higher dependence of computation time on the number of elements, particularly when using direct solvers. Such scaling emphasizes the importance of

task distribution strategies, especially when employing heterogeneous computing systems with processors of varying capabilities. In these cases, non-uniform task allocation ensures efficient utilization of all computational resources.

References

[1] Electronic resource: Open MPI documentation. [Access mode]: https://www.open-mpi.org/doc/

[2] Electronic resource: Fenics documentation. [Access mode]: https://fenicsproject.org/documentation/

[3] Smith, J. High-Performance Computing in Materials Science. Journal of Computational Materials, 2020.

[4] Finite Element Method in Heat Transfer Analysis. Engineering Computation, 2019.

[5] Jaworski N., Lobyr M., Matviykiv O., Farmaga I., Marikutsa U. Synthesis of effective thermal characteristics of anodic alumina by the microlevel cellular composite materials model // Microtechnology and thermal problems in electronics, MICROTHERM 2017: official proceedings of international conference, June 27 –29 2017, Lodz, Poland. – 2017. – P.26–28.

[6] Jaworski N. Thermal Analysis Methods for Design of Composite Materials with Complex Structure / N. Jaworski, I. Farmaga, O. Matviykiv, M. Lobur, P. Spiewak, L. Ciupinski, K. Kurzydlowski // ECS Transactions. – 2014. – Vol. 59. – No. 1. – P. 513-523.

[7] Jaworski N. Research of composite materials optimal design task based on numerical simulation / N. Jaworski, I. Farmaga, M. Lobur, P. Spiewak // Proc. of the 8-th Int. Scientific and Technical Conference "Computer Sciences and Information Technologies (CSIT'2013)". – Lviv (Ukraine), 2013. – P. 46-48

Артем Міканов[1], Ігор Фармага[2], Кшиштоф Кужидловскі[3]

[1] Кафедра систем автоматизованого проектування, Національний університет Львівська політехніка, вул. С. Бандери 12, Львів, Україна, E-mail: artem.mikanov.kn.2021@lpnu.ua, ORCID 0009-0007-0643-3367,

[2] Кафедра систем автоматизованого проектування, Національний університет Львівська політехніка, вул. С. Бандери 12, Львів, Україна, E-mail: ihor.v.farmaha@lpnu.ua, ORCID 0009-0007-3727-2586,

[3] Кафедра матеріально-технічного забезпечення, Білостоцький технологічний університет,вул. Вєйска 45с, Білосток, Польща, E-mai: krzysztof.kurzydlowski@pb.edu.pl ORCID 0000-0003-3875-4820

ВИСОКОЕФЕКТИВНИЙ ОБЧИСЛЮВАЛЬНИЙ МЕТОД ДОСЛІДЖЕННЯ ПРОЦЕСІВ ТЕПЛОПРОВОДНОСТІ В КОМПОЗИТНИХ МАТЕРІАЛАХ

Отримано: Листопад 12, 2024 / Переглянуто: Листопад 20, 2024 / Прийнято: Листопад 25, 2024

Анотація. У цьому дослідженні досліджується підхід до моделювання теплопровідності в композитних матеріалах за допомогою високопродуктивних обчислювальних методів. Розв'язання складної математичної задачі розкладається на кілька незалежних процесів за допомогою методу MPI, щоб максимізувати обчислювальну ефективність. Структура моделі враховує властивості композиційного матеріалу та його компонентів. Для дослідження теплових властивостей застосовано метод скінченних елементів (МСЕ). Результати обчислень демонструють високу ефективність і точність запропонованого підходу, підтверджуючи його актуальність як для наукових досліджень, так і для навчальних цілей.

Ключові слова: композитні матеріали, теплопровідність, метод скінченних елементів, високопродуктивні обчислення, паралельні обчислення, MPI, граничні умови, аналіз розподілу температури.