Proceedings of the Eighth International Conference on
Parallel, Distributed, GPU and Cloud Computing for Engineering
Edited by: P. Iványi, J. Kruis and B.H.V. Topping
Civil-Comp Conferences, Volume 12, Paper 2.1
Civil-Comp Press, Edinburgh, United Kingdom, 2025
ISSN: 2753-3239, doi: 10.4203/ccc.12.2.1


# Hyfeast: A Parallel Finite Element Framework for Advanced Civil Engineering Applications

**J.-R. Cho[1], K. Cho[1], J. H. Lee[2] and D. S. Rhee[3]**

**[1]Department of Structural Engineering Research, Korea Institute of Civil Engineerng and Building Technology, Goyang, Republic of Korea**
**[2]Department of Ocean Engineering, Pukyong National University, Busan, Republic of Korea**
**[3]Department of Hydro Science and Engineering Research, Korea Institute of Civil Engineerng and Building Technology, Goyang, Republic of Korea**

## Abstract

This paper presents Hyfeast, a high-performance finite element framework developed at the Korea Institute of Civil Engineering and Building Technology (KICT) to address the growing demands of advanced structural analysis in civil engineering. Hyfeast is built upon an object-oriented C++ class library (HFC) and consists of four standalone applications for structural analysis, visualization, and sectional evaluation. The framework supports a wide range of linear/nonlinear and static/dynamic simulations. In particular, it offers advanced capabilities for fluid–structure–soil interaction and digital twin–based analysis. By integrating OpenMP-based parallelism and high-performance numerical libraries such as Intel MKL and ARPACK, Hyfeast achieves superior computational performance. Validated through real-world applications including high-speed rail bridges and wastewater treatment facilities, Hyfeast proves to be a robust and extensible tool for structural engineers seeking scalable, customizable, and efficient simulation capabilities.

**Keywords:** high-performance computing, finite element framework, structural simulation, object-oriented design, visualization, digital twin

## 1 Introduction

Modern civil infrastructure demands advanced simulation capabilities that combine computational efficiency with modeling flexibility. While commercial finite element (FE) software is widely used, it often lacks extensibility and customization. In particular, academic and government institutions face limitations in implementing new elements, material models, or coupling schemes due to the closed nature of proprietary platforms.

To bridge these gaps, Hyfeast has been developed as a fully in-house framework since 2008. It is designed with modularity, high performance, and adaptability in mind, supporting a broad range of engineering problems—from traditional linear static analysis to advanced fluid–structure–soil interaction (FSSI) and digital twin–based implementations.

## 2 Architecture and Key Features of Hyfeast Framework

Hyfeast is built on the Hybrid Finite Element Class (HFC) library, written in C++ [1]. This object-oriented foundation enables modular extension and reuse across four primary executable components: hfAnalyzer, hfVisualizer, hfSectionAnalyzer, and hfSectionVisualizer, as shown in Figure 1. Among them, hfAnalyzer is the finite element solver, and hfVisualizer is its GUI-based pre-/postprocessor. hfSectionAnalyzer performs section property calculation and nonlinear section analysis, while hfSectionVisualizer serves as its GUI counterpart. hfAnalyzer and hfSectionAnalyzer operate via the command line, whereas hfVisualizer and hfSectionVisualizer are GUI applications. Data exchange between hfAnalyzer and hfVisualizer is managed through .hdb files, supporting both text and HDF5 formats.

Hyfeast integrates several external libraries: Intel oneAPI MKL for matrix computations [2], ARPACK for eigenvalue analysis [3], Qt for GUI development [4], and VTK for scientific visualization [5]. In addition, a variety of other external libraries were utilized [6–13]. Its hybrid programming model combines object-oriented design for extensibility with procedural routines optimized for performance-critical tasks. Parallel execution is enabled via OpenMP. Among its core innovations are a flexible system-wide DOF architecture—supporting elements such as 5-DOF shells—and robust constraint handling mechanisms, including multi-point constraints (MPCs) and automated elimination of over-constraints [1].

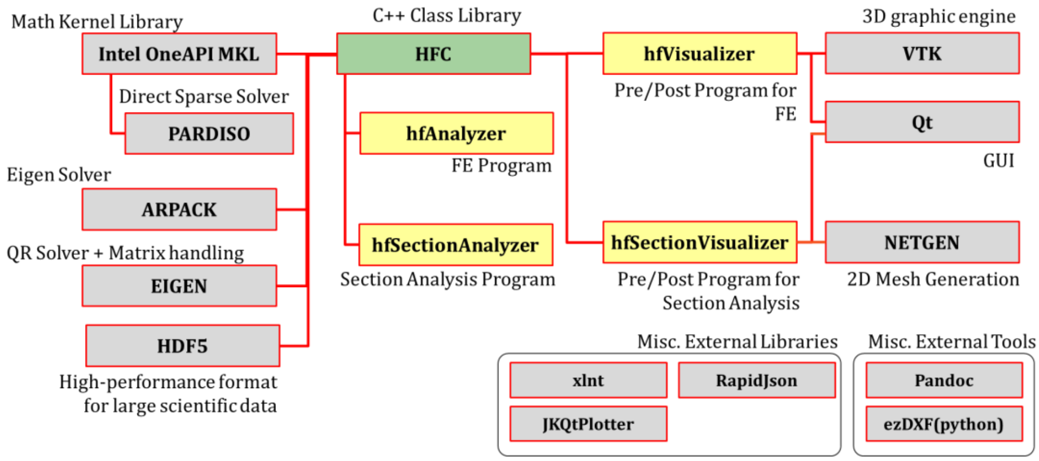



Figure 1: Framework architure of Hyfeast.

# 3 Analysis Capabilities, Elements, and Materials

Hyfeast supports a wide range of analysis types required for structural simulations, including static and dynamic analyses, modal analysis, vehicle–bridge interaction, soil–structure and fluid–structure interactions, and digital twin implementations. Nonlinear solvers include Newton–Raphson, arc-length, BFGS, and line search algorithms.

Hyfeast also offers a comprehensive library of finite elements:

- Beam: B2D2H, B3D2H, B2D2MH, B3D2MH
- Truss: T3D2, Cable
- Shell: S3F, S4F, S3, S4, CS6, CS8
- Solid: CPE3, CPE4, C3D8, …
- Acoustic Solid: AC2D3, AC2D4, AC3D8, …
- MCK Elements: PointMass, EarthSpring, Spring, MovingSpring
- Etc: EmbeddedLine

The full set of beam, shell, and solid elements available in Hyfeast is shown in Figure 2. Except for MCK elements and EmbeddedLine, most element naming conventions in Hyfeast closely follow those of ABAQUS. B2D2H and B3D2H refer to 2D and 3D beam elements, respectively, while B2D2MH and B3D2MH are the corresponding Timoshenko beam elements. S3F and S4F are flat shell elements based on section force–strain constitutive laws, whereas S3 and S4 are general 3-node and 4-node shell elements. CS6 and CS8 are continuum shell elements. MITC and EAS techniques are employed to prevent locking in shell formulations. Solid elements prefixed with 'C' (e.g., CPE3) follow standard formulations. Elements beginning with 'AC' are designed for acoustic media, with pressure as the primary field variable.

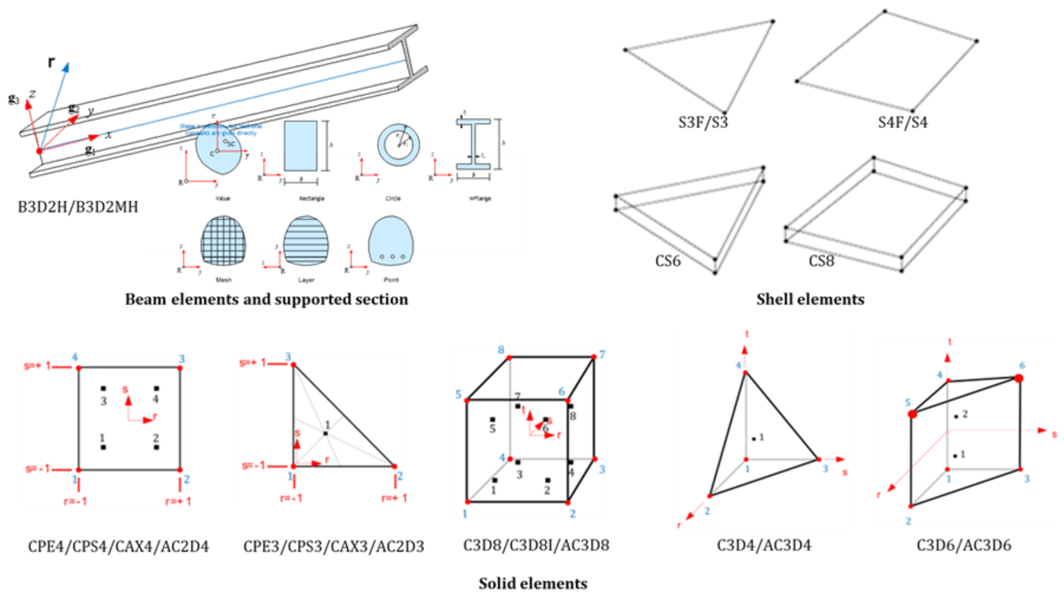



Figure 2: Beam, shell and solid elements.

Hyfeast particularly supports a diverse set of MCK elements. These include rigid arms, which are useful for modeling bridge bearings with vertical offsets. A notable element is the MovingSpring, which enables spring interactions over a predefined surface composed of beam, shell, or solid elements. This facilitates realistic vehicle–bridge interaction modeling, as illustrated in Figure 3.

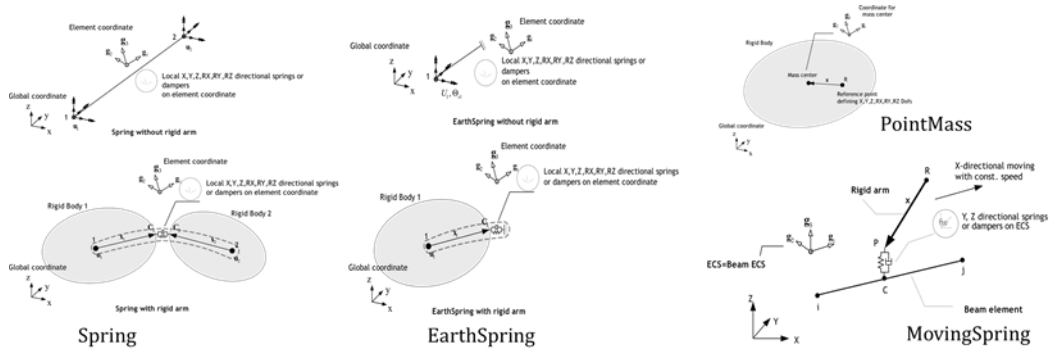



Figure 3: MCK elements

Hyfeast systematically supports 5-DOF shell elements at the system level through efficient DOF management [1]. As illustrated in Figure 4, fully formulated 5-DOF shell elements can be assembled into global models without requiring penalty-type rotational DOFs, which may otherwise degrade accuracy.

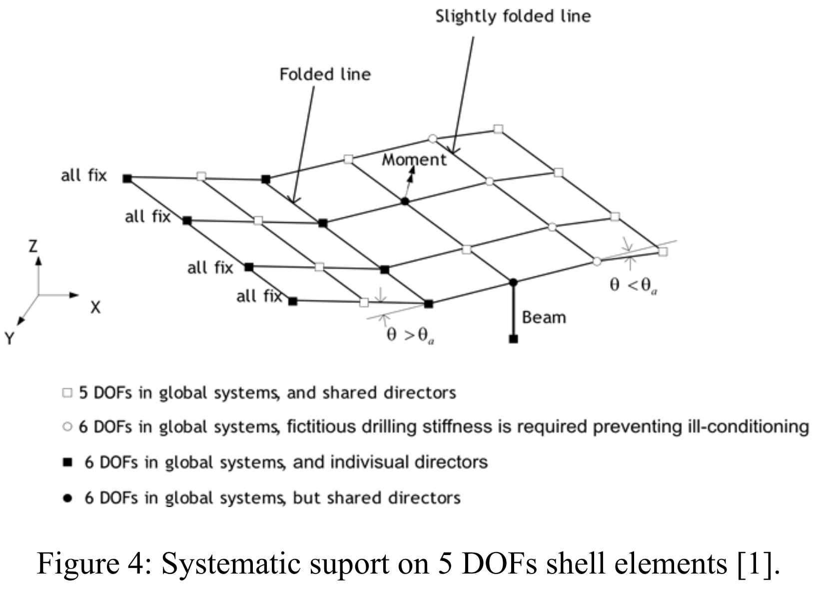



Figure 4: Systematic suport on 5 DOFs shell elements [1].

Hyfeast supports various constraint types, including Support, RigidLink, BeamLink, MPC, NodeToSurface, DistributedSpring, ViscousBoundary, AcousticImpedance, and AcousticSolidInterface. It also includes internal mechanisms for detecting and resolving overconstraints, along with smart selection of slave DOFs for efficient simulation of complex models. Figure 5 illustrates a representative overconstraint issue and common input errors associated with slave DOF selection [1].

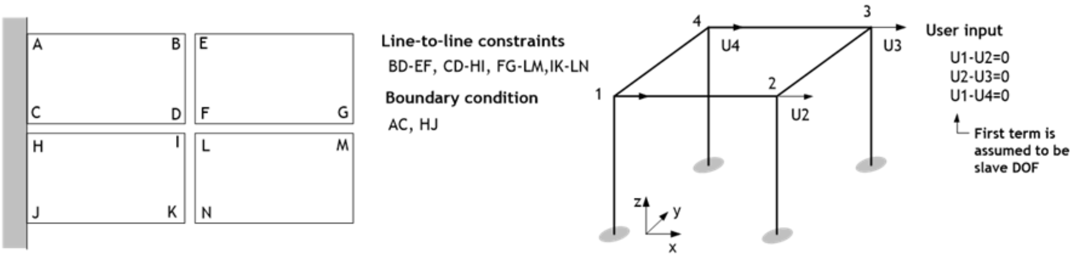
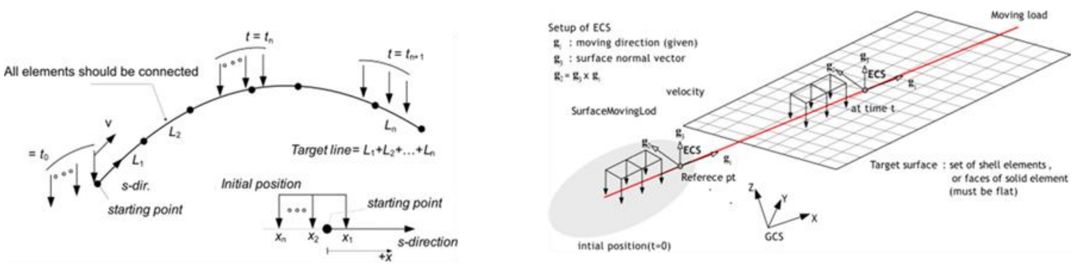



Figure 5: Overconstraint issue and input error in transformation method [1].

Hyfeast accommodates various loading conditions, including Concentric, Displacement, Gravity, Temperature, SeismicRelative, FreeFieldSeismic, LineDistributed, LineMoving, SurfaceDistributed, and SurfaceMoving loads. Among these, LineMoving and SurfaceMoving loads are particularly effective for simulating time-dependent interactions. Figure 6 illustrates the application of these moving loads. LineMoving loads enable the simulation of moving forces along a set of connected beam elements, such as trains or vehicles traveling across a bridge. SurfaceMoving loads extend this capability to flat surfaces composed of shell or solid element faces, allowing directional loading to be applied smoothly along predefined paths. These features are crucial for influence surface analysis and dynamic safety evaluation of high-speed railway bridges under moving loads. It is worth noting that MovingSpring elements operate in a similar manner—not as loads, but as springs that move over a specified surface during the simulation.




LineMoving SurfaceMoving

Figure 6: Moving loads.

Hyfeast supports various constraint types, including Support, RigidLink, BeamLink, MPC, NodeToSurface, DistributedSpring, ViscousBoundary, AcousticImpedance, and AcousticSolidInterface. It also includes internal mechanisms for detecting and resolving overconstraints, along with smart selection of slave DOFs for efficient simulation of complex models. Figure 5 illustrates a representative overconstraint issue and common input errors associated with slave DOF selection [1].

Hyfeast also supports a range of material models, as shown in Figure 7, including isotropic and orthotropic elasticity, von Mises, Tresca, and Drucker–Prager plasticity, as well as Menegotto–Pinto uniaxial models and uniaxial gap–hook models.

Despite its extensive capabilities, current limitations of Hyfeast as a finite element platform include the lack of support for multi-axial concrete material models, geometric nonlinear analysis, contact analysis, and buckling analysis.

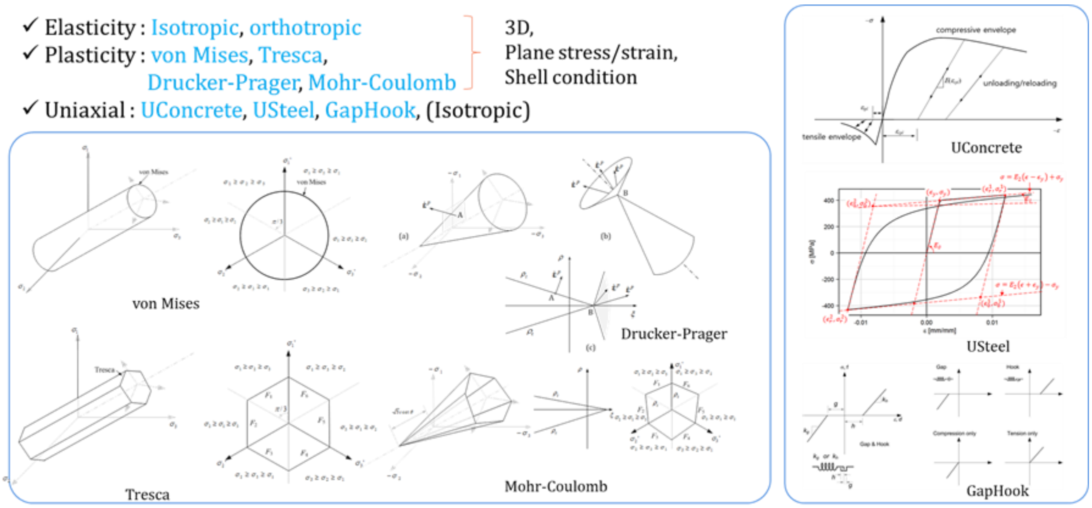



Figure 7: Material models.

## 4 Special Analysis Features

Previous studies have shown that the dynamic behaviour of liquid storage tanks is strongly influenced by the fluid-structure-soil interaction, and this phenomenon has been observed in real earthquakes. Therefore, earthquake response analysis considering the fluid-structure-soil interaction is essential to ensure the seismic safety of liquid storage tanks. In particular, when a heavy structure such as a liquid storage tank rests on a flexible ground, nonlinear behaviour of the ground can occur and significantly affect the response of the whole system. One of the factors to be considered in the analysis of the soil-structure interaction is the energy radiation into the far-field region of the ground. Therefore, this study follows the procedure described in Section 2 to analyse the earthquake response of an unanchored liquid storage tank considering the nonlinear fluid-structure-soil interaction.

### 4.1 FSSI Analysis Support

Hyfeast provides dedicated support for Fluid-Structure-Soil Interaction (FSSI) analysis. The computational model is divided into Near Field and Far Field regions. The Near Field includes water, tank, and surrounding soil. Water is modeled using acoustic solid elements with AcousticImpedance boundary conditions applied at the free surface. The interface between the water and the tank structure is defined using AcousticSolidInterface elements. The tank structure itself is modeled with shell elements, while the soil in the Near Field is modeled using solid continuum elements.

The far field is handled using a simplified perfectly matched discrete layer (PMDL) approach. A midpoint integrated element and viscous damper are used to represent wave radiation and dissipate energy effectively. The effective seismic input for the FSSI model is derived from a preliminary free field analysis and applied at the interface between the near and far field regions. Currently, more advanced implementations—including the Domain Reduction Method (DRM) and rigorous formulations of PMDL and perfectly matched layers (PML)—are under development.

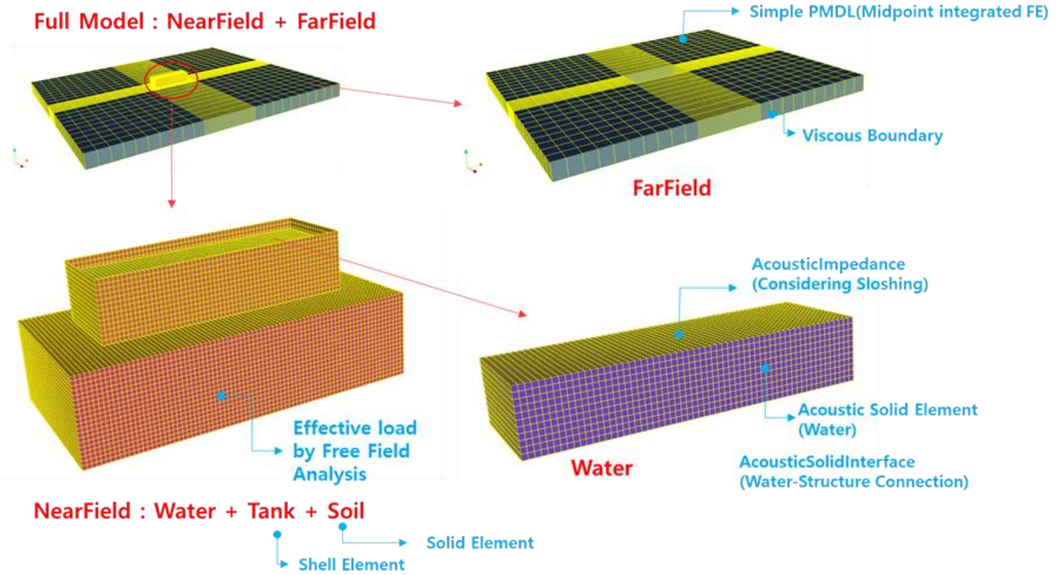



Figure 8: FSSI model.

## 4.2 Digital Twin Support Capabilities

Hyfeast provides comprehensive functionality for digital twin applications. It includes a virtual sensor system capable of reporting strain, displacement, velocity, and acceleration at arbitrary locations—independent of mesh nodes. In addition, it supports advanced tasks such as model updating, shape estimation, and performance evaluation. As illustrated in Figure 9, these capabilities allow Hyfeast to be integrated into infrastructure monitoring, structural performance assessment, and predictive maintenance workflows [14].

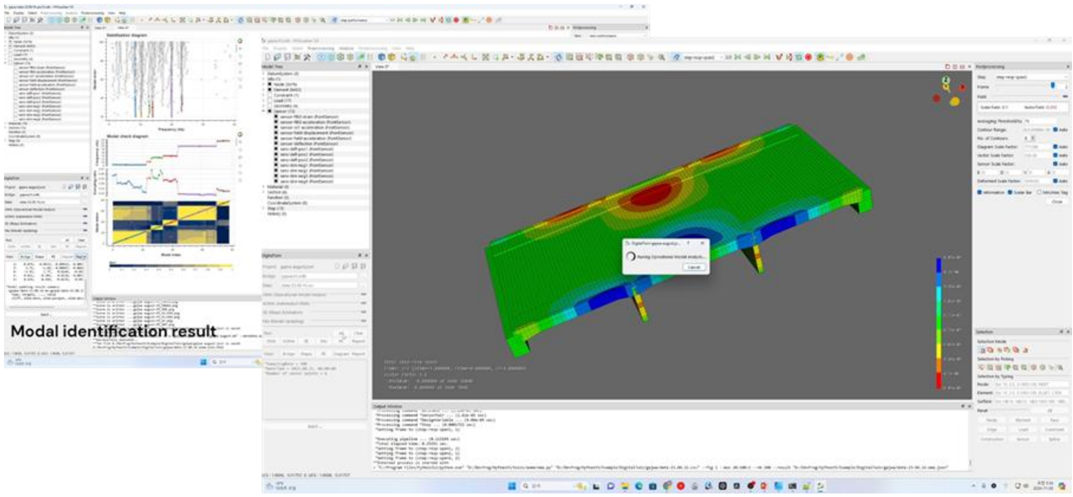
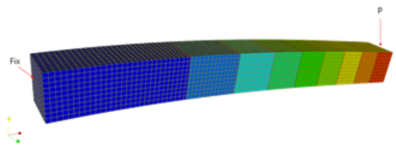



Figure 9: Digital twin for performance evalution of bridge [14].

## 5 Numerical Solvers and Parallel Performance

Hyfeast leverages Intel oneAPI MKL for low-level matrix and vector operations. For solving sparse linear systems, it employs direct solvers such as PARDISO, as well as iterative solvers including BiCG, BiCGStab, and CGS. For eigenvalue problems, both ARPACK and Subspace Iteration methods are integrated, with PARDISO and ARPACK being the default choices.

Parallelism via OpenMP is applied to element-level state updates and global matrix assembly, providing good scalability in multi-core environments. As shown in Figure 10, a performance test of Hyfeast was conducted using up to 24 CPU cores. The results show near-linear speedup up to 16 cores; beyond that point, gains diminish due to thread contention and memory bandwidth limitations. The benchmark model was a dynamic simulation involving over one million degrees of freedom with 3D solid elements [15].




Using full cores(24 cores) Intel Xeon(R) E5-2687 v4(Dual)

| Model | No. of DOFs | State Determination Time (sec) | Direct Sparse Solver Time (sec) | | | |
|---|---|---|---|---|---|---|
| | | | Analysis | Factorization | Solution | Total |
| C100x10x10 | 36,663 | 0.273514 | 0.313844 | 0.180964 | 0.0252515 | 0.52006 |
| C200x20x20 | 265,923 | 2.28899 | 1.9719 | 2.81713 | 0.291406 | 5.080436 |
| C300x30x30 | 867,783 | 6.39494 | 7.14851 | 20.8204 | 1.32534 | 29.29425 |
| C400x40x40 | 2,022,243 | 18.8086 | 24.1359 | 113.54 | 3.9931 | 141.669 |

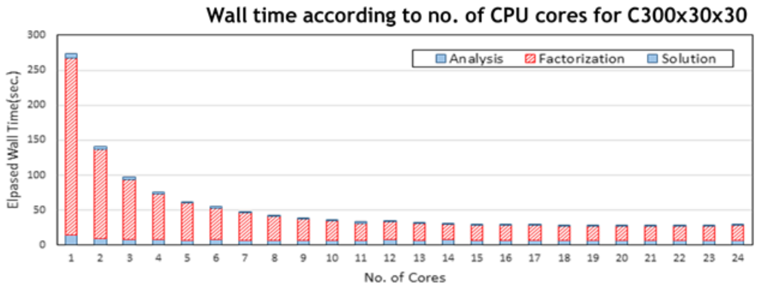
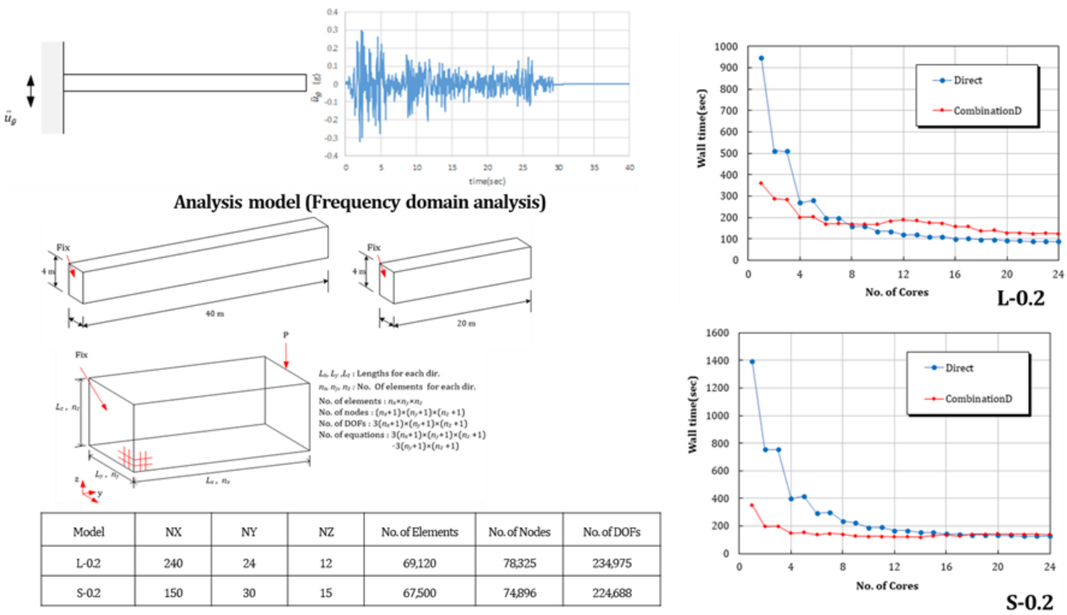



Figure 10: Scalability of Hyfeast with increasing CPU cores [15].

Hyfeast also provides an iterative–direct sparse solver, particularly effective in scenarios where the system matrix evolves gradually and must be solved repeatedly. This solver is based on PARDISO, with modifications tailored for dynamic reuse. In the original PARDISO implementation, when solving a sequence of updated linear systems, the factorized matrix from a previous direct solution is reused as a preconditioner for a Krylov iterative solver. If the solution does not converge within a preset number of iterations, PARDISO falls back to a direct solver and reuses the new factorization result as the preconditioner for subsequent systems. Hyfeast enhances this approach by dynamically estimating the time cost of the first Krylov iteration and using it to adaptively set the maximum number of iterations. This adaptive strategy improves performance and has been validated in frequency-domain analyses. As shown in Figure 11, the proposed algorithm (Combination D) was tested on a cantilever beam model and compared with a baseline direct-only approach. The results demonstrate superior efficiency with the adaptive method [16]. Additional improvements to the algorithm are currently in progress, and its applicability to nonlinear analysis is under investigation.




| Model | NX | NY | NZ | No. of Elements | No. of Nodes | No. of DOFs |
|---|---|---|---|---|---|---|
| L-0.2 | 240 | 24 | 12 | 69,120 | 78,325 | 234,975 |
| S-0.2 | 150 | 30 | 15 | 67,500 | 74,896 | 224,688 |

Figure 11: Iterative–direct sparse solver performance comparison [16].

## 6 Pre/Postprocessing with hfVisualizer

Previous studies have shown that the dynamic behaviour of liquid storage tanks is hfVisualizer provides a comprehensive graphical user interface for 3D modeling, data editing, and result visualization. As shown in Figure 12, the interface integrates multiple modeling and postprocessing features:

- Interactive GUI for geometric modeling (copy, extrude, divide)
- Table-based entity management for nodes, elements, and boundary conditions

- Postprocessing tools including deformed shape visualization, contour/diagram/vector plots, and animation
- Section view rendering for beam and shell elements, incorporating cross-sectional dimensions
- Undo/redo functionality for interactive operations
- Import/export capabilities for Hyfeast, ABAQUS, MIDAS, and GMSH file formats[17-20]

Unlike general-purpose preprocessors that support full CAD-based modeling and meshing, hfAnalyzer does not provide geometric modeling or automatic mesh generation. Instead, similar to tools such as MIDAS Civil and MIDAS GEN, Hyfeast adopts a node-and-element definition strategy in which the mesh is constructed through direct manipulation of nodal coordinates and element connectivity.

A notable feature recently added to hfVisualizer is its ability to reconstruct virtual displacement fields from dynamic pressure data obtained using acoustic solid elements. As illustrated in Figure 13, this feature enables intuitive visualization of internal fluid motion such as sloshing, and is particularly useful for analyzing hydrodynamic behavior in tanks or reservoirs.

Additionally, hfVisualizer offers smooth import of structural models created with external tools. For instance, Figure 14 shows a model imported directly from MIDAS GEN, preserving node, element, and boundary condition definitions for immediate simulation within Hyfeast.

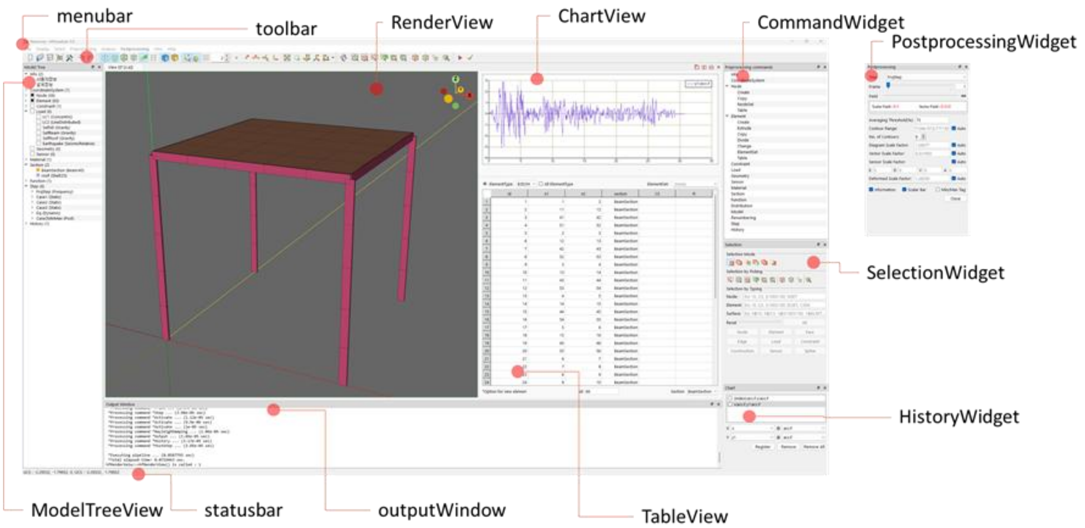



Figure 12: UI layout of hfVisualizer.

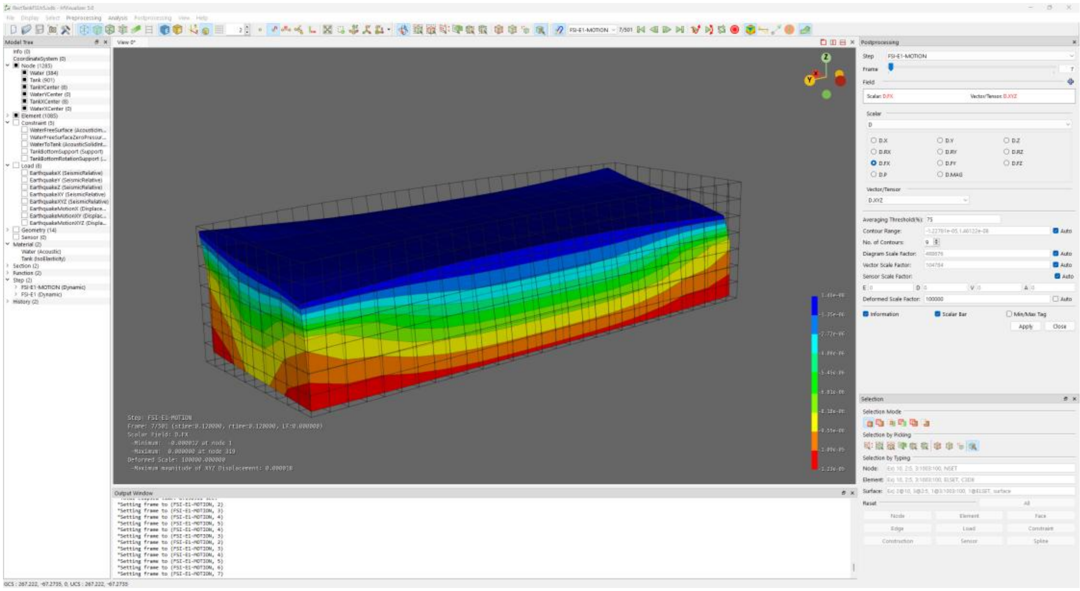


Figure 13: Virtual fluid deformation visualization using acoustic pressure data.

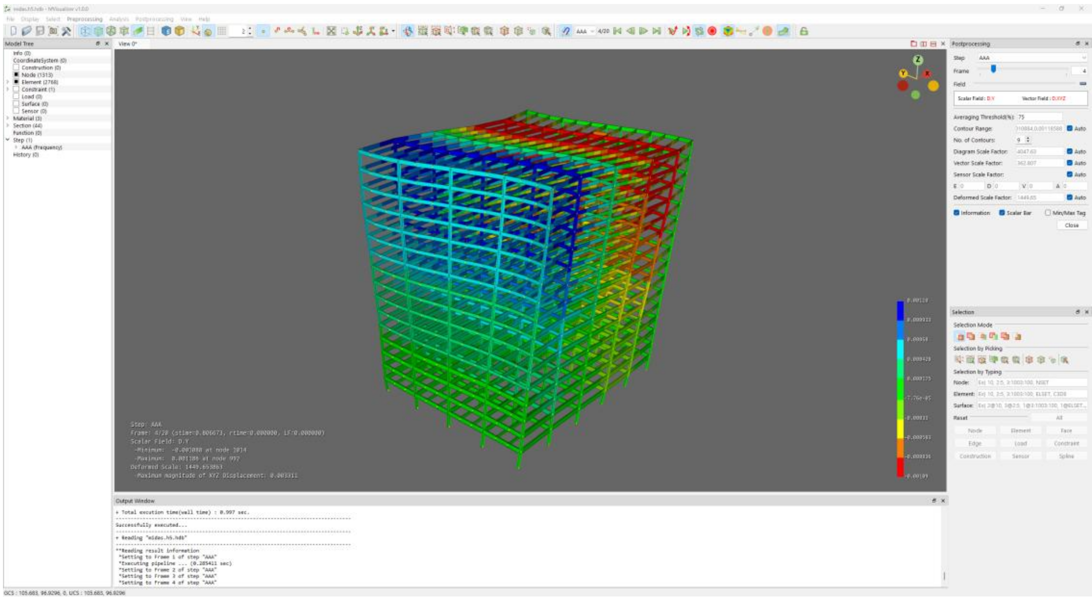


Figure 14: Model imported from MIDAS GEN.

## 7 Sectional Analysis Tools

Hyfeast includes dedicated tools for cross-sectional analysis. The hfSectionAnalyzer and hfSectionVisualizer compute geometric properties such as area, moment of inertia, torsional constant, and shear-related parameters for arbitrary cross-sectional shapes as shown in Figure 15. In particular, torsional constant and shear parameters are obtained via 2D finite element analysis. The tools also support nonlinear sectional response analyses using layered and fiber models, enabling the generation of moment–curvature relationships and axial force–moment (P–M) interaction diagrams.

To support arbitrary section modeling, hfSectionVisualizer provides a GUI for defining and editing section geometry, generating mesh, and visualizing stress and strain distributions. These tools are tightly integrated with hfAnalyzer and hfVisualizer, allowing seamless use of custom-defined sections in structural models.

## 8 Conclusions

Hyfeast is an extensible and high-performance finite element analysis platform suitable for both academic research and practical structural engineering applications. Its hybrid software architecture, parallel computing capabilities, and modular system design enable it to support a wide range of use cases—from high-speed railway dynamics to digital twin implementations.

Future development will focus on expanding Hyfeast's capabilities for architectural and civil structural design workflows, including code-based design verification. Additionally, geotechnical analysis features will be enhanced to support seepage modeling and coupled flow–stress simulations in porous media. Hyfeast will also introduce Python scripting support to enable customizable workflows.

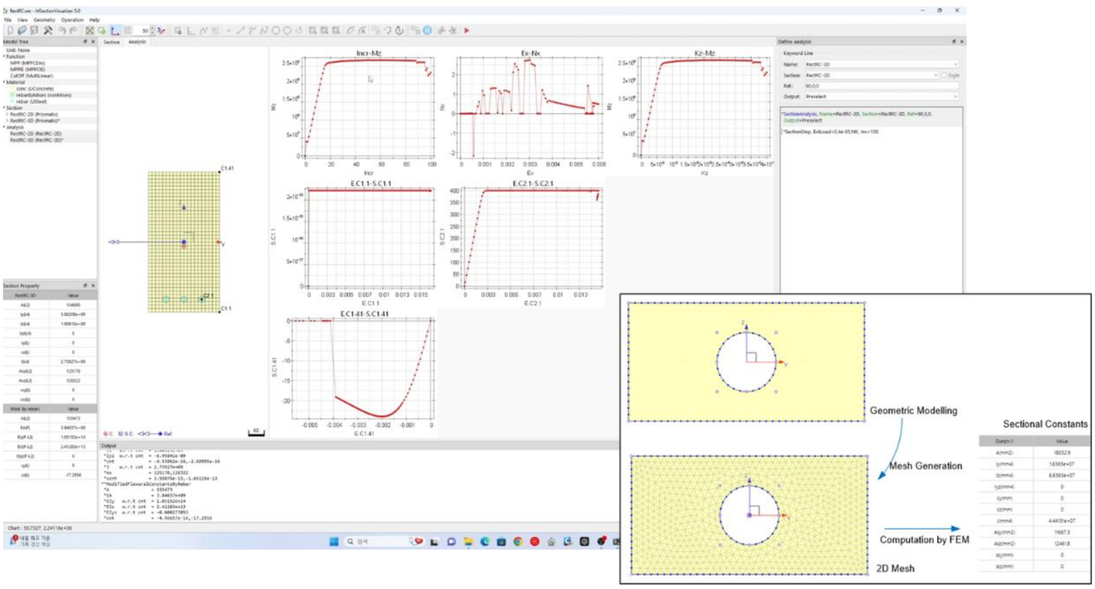



Figure 15: Section property computation and moment–curvature analysis using hfSectionAnalyzer and hfSectionVisualizer.

Although initially developed as an internal in-house project, Hyfeast is scheduled for public release this summer as a free binary distribution, aiming to support broader adoption within the engineering community.

## Acknowledgements

This work is financially supported by Korea Ministry of Environment(MOE) as 「Technology development project to optimize planning, operation, and maintenance of urban flood control facilities)(RS-2024-00397821)」.

# References

[1] Cho, J.-R., Object-oriented Finite Element Framework Using Hybrid Programming, Ph.D. Dissertation, Seoul National University, Seoul, Republic of Korea, 2009.
[2] Intel Math Kernel Library (MKL) Documentation, Intel Corporation, available at https://www.intel.com/content/www/us/en/docs/onemkl/ (accessed April 2025).
[3] ARPACK Users' Guide, Rice University CAAM, available at https://www.caam.rice.edu/software/ARPACK/ (accessed April 2025).
[4] The Qt Company, Qt 5.12.2 – patch release of Qt 5.12 LTS, released March 15 2019, available at https://www.qt.io (accessed April 2025).
[5] VTK – Visualization Toolkit, available at https://vtk.org/ (accessed April 2025).
[6] Eigen Library, available at https://eigen.tuxfamily.org/ (accessed April 2025).
[7] The HDF Group, HDF5 Reference Manual, Version 1.10…, available at https://support.hdfgroup.org/documentation/hdf5/latest/ (accessed April 2025).
[8] Schöberl, J., "NETGEN: An advancing front 2D/3D-mesh generator based on abstract rules," Computing and Visualization in Science, vol. 1, pp. 41–52, July 1997. doi:10.1007/s007910050004
[9] tfussell, "xlnt: a modern C++ library for manipulating spreadsheets," GitHub repository, first public release May 10 2017, available at https://github.com/tfussell/xlnt (accessed April 2025).
[10] Yip, M., RapidJSON: A fast JSON parser/generator for C++, Tencent, header-only, available at https://rapidjson.org/ (accessed April 2025).
[11] Kriege, J., JKQtPlotter: A Qt plotting library for Qt5/Qt6, LGPL-2.1+, GitHub Pages, available at https://jkriege2.github.io/JKQtPlotter (accessed April 2025).
[12] Pandoc User's Guide and API Manual, available at https://pandoc.org (accessed April 2025).
[13] mozman, ezdxf 1.4.2, Python package for DXF creation/manipulation (MIT License), available at https://ezdxf.readthedocs.io/ (accessed April 2025).
[14] Cho, K. and Cho, J.-R., "Stochastic Subspace Identification-based Automated Operational Modal Analysis Considering Modal Uncertainty," Applied Sciences, vol. 13, no. 22, Article 12274, 2023. doi:10.3390/app132212274
[15] Cho, J.-R. and Cho, K.-H., "Design Considerations on Large-scale Parallel Finite Element Code in Shared Memory Architecture with Multi-Core CPU," Journal of the Computational Structural Engineering Institute of Korea, vol. 30, no. 2, pp. 127–135, April 2017. doi:10.7734/COSEIK.2017.30.2.127
[16] Cho, J.-R. and Cho, K.-H., "Computational Efficiency of Frequency-Domain Analysis of Large-Scale Finite-Element Models Using an Iterative–Direct Sparse Solver Combination," Journal of the Computational Structural

Engineering Institute of Korea, vol. 32, no. 2, pp. 117–124, 2019. doi:10.7734/COSEIK.2019.32.2.117
[17] Dassault Systèmes Simulia Corp., "Abaqus FEA," available at https://en.wikipedia.org/wiki/Abaqus (accessed April 2025).
[18] MIDAS Information Technology, "Midas Civil 2020 v. 3.1 Release," available at https://www.midasoft.com/bridge-library/civil/products/midascivil (accessed April 2025).
[19] MIDAS Information Technology, "Midas GEN – Features," available at https://www.midasuser.com/en/product/gen/features (accessed April 2025).
[20] Geuzaine, C. and Remacle, J.-F., "Gmsh: A 3-D finite element mesh generator with built-in pre- and post-processing facilities," International Journal for Numerical Methods in Engineering, vol. 79, no. 11, pp. 1309–1331, 2009. doi:10.1002/nme.2579