Development and Verification of HANS-D: High-Order FEM Code for the Neutron Diffusion Equation

Nak Woong Yang, Chang Je Park*
Nuclear Engineering Dept., Sejong Univ., 209 Neungdong-ro, Gwangjin-gu, Seoul 143-747, Republic of Korea
**Corresponding author: parkcj@sejong.ac.kr*

***Keywords :** Neutron diffusion equation, Finite element method, High-order element

1. Introduction

In the contemporary nuclear industry, Small Modular Reactors (SMRs) are garnering significant attention as a viable alternative to large-scale nuclear power plants. SMRs are characterized by high neutron leakage, strong core heterogeneity, and complex geometric structures compared to conventional large reactors. These characteristics demand higher geometric flexibility from core analysis codes, thereby increasing the importance of the Finite Element Method (FEM), which possesses distinct advantages in handling unstructured meshes.

A key advantage of the FEM is its ability to flexibly control solution accuracy by adjusting the element order. However, in 3D high-order FEM analysis, the increased connectivity between nodes leads to a drastic proliferation of non-zero entries in the system matrix. This results in excessive memory consumption and computational bottlenecks during the matrix assembly phase.

To address this issue, matrix assembly efficiency was improved by applying dynamic List-of-Lists (LIL) intermediate structure followed by conversion to Compressed Sparse Row (CSR) format. Also, the performance implications of alternative static pre-allocation strategies are investigated. Based on these methodologies, this paper presents the development of HANS-D (High-order Analysis Neutronics Solver - Diffusion), a 3D neutron diffusion analysis code developed in C++, and verifies its accuracy and computational efficiency through standard benchmark problems.

2. Methods

2.1. High-order FEM formulation

The HANS-D code is based on the weak form of the steady-state 2-group neutron diffusion equation. By multiplying the governing equation by a test function ν, integrating over the entire volume, and applying the Divergence Theorem for discretization, the system results in the following generalized eigenvalue problem:

$$\boldsymbol{A}\Phi = \frac{1}{k_{eff}}\boldsymbol{F}\Phi \quad (1)$$

$$\boldsymbol{A} = \begin{bmatrix} \boldsymbol{A_{11}} & \boldsymbol{0} \\ \boldsymbol{A_{21}} & \boldsymbol{A_{22}} \end{bmatrix}, \quad \boldsymbol{F} = \begin{bmatrix} \boldsymbol{F_{11}} & \boldsymbol{F_{12}} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \quad (2)$$

Here, A is the stiffness matrix including leakage and removal terms, and F is the fission source matrix [1]. Up-scattering was not considered in this formulation.

Rectangular and hexagonal structured meshes are utilized, and Lagrange polynomials are adopted as basis functions for spatial discretization. The code implements linear (1st), quadratic (2nd), and cubic (3rd) shape functions in a generalized tensor product form, allowing for flexible expansion according to the element order.

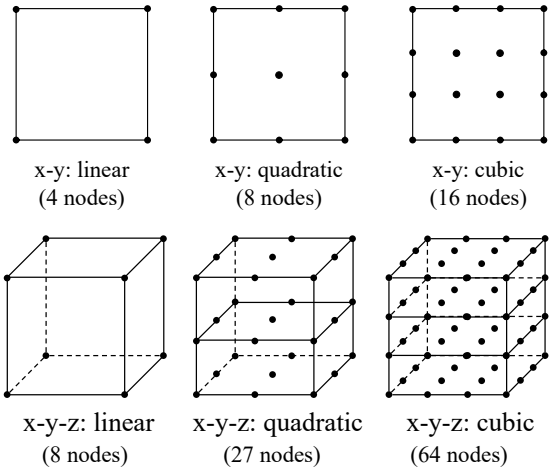

Fig. 1. Types of 2D and 3D elements.

For the internal linear system solver, the BiCGSTAB algorithm, which is suitable for asymmetric sparse matrices, was used [2]. Additionally, Jacobi and ILU factorization were applied as preconditioners to accelerate convergence speed and enhance computational efficiency.

2.2. Matrix storage strategy

The global system matrix is assembled via the standard FEM element loop. For each element, the local stiffness matrix is computed by numerical integration. When using high-order elements, the number of nodes per element increases significantly. For example, a cubic hexahedral element contains 64 nodes, resulting in a 128 x 128 local matrix per element under the 2-group formulation. This leads to a drastic increase in the number of non-zero (NNZ) entries in the global system

matrix, making efficient storage structure design essential.

During assembly, a list-of-lists (LIL) format is employed as the intermediate data structure, implemented as std::vector<std::map<int,double>> of dimension N_{dofs}. Each row is represented by a std::map that dynamically accumulates column indices and associated values as local contributions are scattered from each element. This approach requires no prior knowledge of the sparsity pattern and naturally accommodates arbitrary mesh connectivity [3]. Upon completion of the element loop, the LIL structure is converted to CSR format via a sequential traversal of each row map. The CSR format stores the matrix as three compact arrays – non-zero values, column indices, and row pointers – enabling memory-efficient calculation required by the solver.

3. Results

To verify the code, the EPRI-9R, IAEA 2D, and IAEA 3D benchmarks were selected. All calculations were using a 1/4 symmetry condition, and the error is defined as:

$$\Delta k_{eff} = k_{ref} - k_{HANS-D} \quad (3)$$

3.1. EPRI-9R core benchmark problem

Fig. 2. shows the core configuration of the EPRI-9R benchmark, and Table I. lists the material compositions used [4, 5].

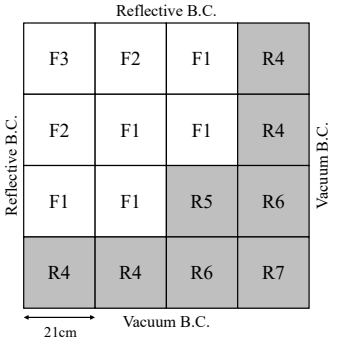

Fig. 2. Configuration of the EPRI-9R benchmark

Table I: Two-group cross-section data for EPRI-9R benchmark

ID	D_1	D_2	Σ_{s12}	Σ_{a1}	Σ_{a2}	$\nu\Sigma_{f1}$	$\nu\Sigma_{f2}$
F1	1.5133609	0.3950340	0.211256	0.01209950	0.16856040	0.00601220	0.21888080
F2	1.5133554	0.3951559	0.211252	0.00932488	0.14141880	0.00462490	0.16455390
F3	1.4656253	0.3825032	0.189507	0.01478880	0.18022310	0.00463217	0.17191960
R4	1.3509546	0.3482913	0.214356	0.00185262	0.06051837	0.0	0.0
R5	1.2278506	0.3467473	0.148120	0.00226896	0.07011230	0.0	0.0
R6	1.5565972	0.3495268	0.301634	0.00130402	0.05290274	0.0	0.0
R7	1.7	0.35	0.350000	0.00100000	0.05000000	0.0	0.0

Table II. presents the k_{eff} calculation results according to element order and mesh refinement.

Table II: The FEM solutions to the EPRI-9R benchmark

Element Type	Element/FA	k_{eff}	Δk_{eff} (pcm)	Computing time(sec)
1	1x1	0.88996	155	0.148
	3x3	0.89248	-97	0.123
	7x7	0.89223	-72	0.148
	21x21	0.89211	-60	0.511
2	1x1	0.89419	-268	0.152
	3x3	0.89233	-82	0.154
	7x7	0.89211	-60	0.367
	21x21	0.89209	-58	7.217
3	1x1	0.89288	-137	0.122
	3x3	0.89211	-60	0.243
	7x7	0.89209	-58	1.445
	21x21	0.89209	-58	36.202

Ref. keff=0.89151 (FEM, Element/FA=30x30) [5]

It was confirmed that the solution converges stably as the element order increases. However, a consistent bias of approximately 268 pcm compared to the reference solution was observed. This bias is attributed to the fact that the HANS-D code is designed based on a native 3D solver architecture. The approximations inherent in modeling axial buckling or transverse leakage for 2D problems like EPRI-9R introduce physical uncertainties compared to a full 3D analysis.

3.2. IAEA 2D core benchmark problem

Fig. 3. illustrates the configuration of the IAEA 2D benchmark, and Table III. shows the material compositions [6]. In the IAEA 2D benchmark, an axial buckling of $B^2_{z,g}=0.8*10^{-4}$ was applied for all regions and energy groups.

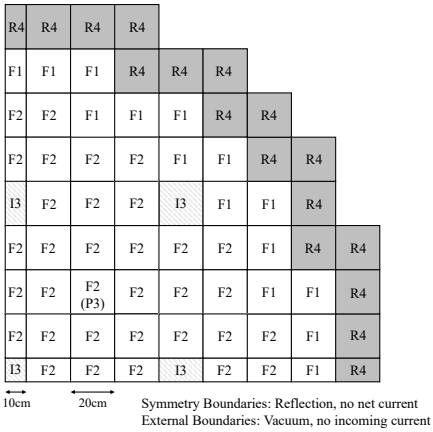

Fig. 3. Configuration of the IAEA benchmark(z=190cm)

Table III: Two-group cross-section data for IAEA benchmark

ID	D_1	D_2	Σ_{s12}	Σ_{a1}	Σ_{a2}	$\nu\Sigma_{f2}$
F1	1.5	0.4	0.02	0.01	0.08	0.135
F2	1.5	0.4	0.02	0.01	0.085	0.135
I3/P3	1.5	0.4	0.02	0.01	0.13	0.135
R4	2.0	0.3	0.04	0	0.01	0.135
R5	2.0	0.3	0.04	0	0.055	0.135

In contrast to the EPRI-9R case, the results for the IAEA 2D benchmark in Table IV. demonstrated excellent agreement with the reference solution. While slight errors were observed with linear elements on coarse meshes, the application of quadratic and cubic elements achieved high accuracy with errors less than 10 pcm.

Table IV: The FEM solutions to the IAEA 2D benchmark

Element Type	Element/FA	k_{eff}	Δk_{eff} (pcm)	Computing time(sec)
1	2x2	1.03109	-149	0.498
	4x4	1.02983	-23	0.399
	10x10	1.02961	-1	0.840
	20x20	1.02958	2	6.158
2	2x2	1.02969	-9	0.813
	4x4	1.02958	2	0.547
	10x10	1.02959	1	10.830
	20x20	1.02959	1	99.773
3	2x2	1.02969	-9	0.882
	4x4	1.02957	3	3.775
	10x10	1.02957	3	63.553
	20x20	1.02959	1	607.298

Ref. keff=1.0296 (FEM, Element type=2, Element/FA=4x4) [6]

3.3. IAEA 3D core benchmark problems

The vertical cross-section of the IAEA benchmark is shown in Fig. 4. The material compositions are consistent with Table III.

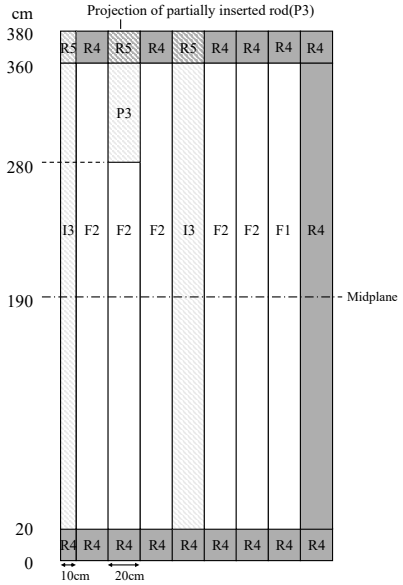

Fig. 4. Configuration of the IAEA benchmark(y=0cm)

Table V. presents the results for the 3D problem. Even with cubic elements, the maximum error was merely 6 pcm, a significant improvement over the 167 pcm error observed with linear elements. This demonstrates the code's capability to solve complex 3D problems accurately without requiring excessively fine mesh refinement.

Table V: The FEM solutions to the IAEA 3D benchmark

Element Type	Element/FA	k_{eff}	Δk_{eff} (pcm)	Computing time(sec)
1	2x2x38	1.03087	-167	1.936
	4x4x38	1.02953	-33	8.587
2	2x2x38	1.02929	-9	41.654
	4x4x38	1.02916	4	275.372
3	2x2x38	1.02914	6	581.530
	4x4x38	1.02914	6	3928.179

Ref. keff=1.0292 (FEM, Element type=2, Element/FA=2x2x13) [6]

It is noted that some mesh conditions evaluated in this study are finer than those defined in the benchmark reference, thereby encompassing a broader convergence range for a more comprehensive analysis. As observed in Table IV and Table V, quadratic elements in particular tend to exhibit smaller Δk_{eff} values under conditions comparable to those of the reference, suggesting that 2nd-order elements provide a favorable balance between accuracy and computational cost in practical applications.

3.4. Convergence and efficiency

To rigorously analyze the performance characteristics of the HANS-D code, the correlations between calculation accuracy, mesh refinement, and computational cost were investigated using the IAEA 2D benchmark as a representative case. Fig. 5. illustrates the convergence of Δk_{eff} with respect to mesh refinement.

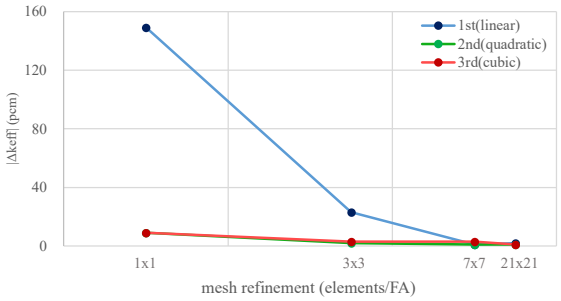

Fig. 5. Convergence of Δk_{eff} with mesh refinement for the IAEA 2D benchmark

As shown in Fig. 5., at equivalent levels of mesh refinement, high-order elements consistently achieve superior convergence of k_{eff} compared to linear elements. For example, at 4x4 elements/FA, linear, quadratic, cubic elements yield Δk_{eff} of -23.2 pcm, 2 pcm and 3 pcm, respectively – an improvement of nearly an order of magnitude under identical mesh conditions. Furthermore, while linear elements retain appreciable residual errors even on fine meshes (e.g., 20x20 elements/FA), quadratic and cubic elements achieved high accuracy with Δk_{eff} less than 10 pcm even on coarse meshes (e.g., 4x4). This confirms that high-order basis functions can resolve steep flux gradients near material interfaces with significantly fewer spatial degrees of freedom.

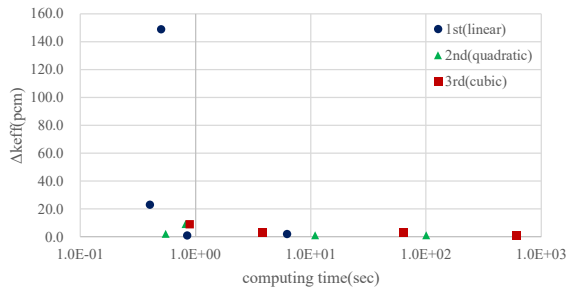

Fig. 6. Relationship between computing time and Δk_{eff} for the IAEA 2D benchmark

Fig. 6. presents the relationship between computing time and Δk_{eff} for each element order, providing a direct comparison of computational efficiency at equivalent accuracy levels. As shown in the figure, quadratic and cubic elements achieve $\Delta k_{eff} < 10$ pcm at substantially lower computing times compared to linear elements at equivalent accuracy. For instance, the quadratic element at 4x4 elements/FA yields $\Delta k_{eff} = 2$ pcm in 0.547 sec, whereas the linear element requires 20x20 elements/FA and 6.158 sec to attain comparable accuracy. This demonstrates that a coarse mesh, high-order shape function strategy is significantly more computationally efficient than a fine-mesh, linear element approach when targeting the same level of solution accuracy.

Table VI presents the matrix assembly time and peak memory usage for four storage strategies applied to the IAEA 3D benchmark (element order = 3, N_{dofs} = 522,560, N_{elem} = 9,158).

Table VI: Matrix Storage Strategy Comparison to the IAEA 3D benchmark

Matrix	Strategy	Computing time (sec)	Memory (MB)
Dense	Dynamic	-	$> 4.0 \times 10^6$
	Static	-	$> 4.0 \times 10^6$
Sparse	Dynamic	17.387	7.3×10^3
	Static	19.901	2.5×10^3

Dense matrix storage requires an estimated 4×10^6 MB for this problem, rendering it entirely impractical for large-scale 3D neutron diffusion analysis. Among sparse strategies, the Sparse/Static approach reduces peak memory by approximately 65% relative to the current Sparse/Dynamic (LIL-based) implementation. However, the sparsity pattern pre-analysis phase introduces additional overhead, resulting in a modest increase in assembly time. This trade-off suggests that the Sparse/Static strategy is particularly advantageous when memory availability is the primary constraint.

4. Conclusions

In this study, HANS-D, a 3D neutron diffusion analysis code based on the high-order finite element method, was developed to address the geometric complexities of modern reactor core analysis. The verification results through standard benchmarks confirmed that HANS-D ensures high numerical accuracy.

The comparison of matrix storage strategies reveals that sparse formats are overwhelmingly more efficient than dense storage for large-scale 3D neutron diffusion analysis. The current implementation adopts a dynamic assembly approach (LIL-to-CSR). However, the results indicate that transitioning to a static pre-allocation strategy would be necessary to address memory constraints in larger-scale problems.

Overall, it was concluded that high-order elements on coarse meshes provide more favorable trade-off between accuracy and computational cost compared to linear elements on refined meshes.

The current implementation is based on the neutron diffusion approximation, which has inherent limitations in regions where neutron transport effects are significant, such as near material interfaces and boundaries in the heterogeneous geometries of SMR. Future work will focus on extending HANS-D by incorporating the Simplified P_N (SP_N) or Even-Parity Transport formulations to improve physical fidelity while preserving the geometric flexibility of the high-order FEM.

Acknowledgement

This work was supported by Korea Institute of Energy Technology Evaluation and Planning (KETEP) grant funded by the Ministry of Trade, Industry and Energy (RS-2023-0233621)

REFERENCES

[1] J.R. Lamarsh, *Introduction to Nuclear Reactor Theory*, Addison-Wesley Publishing Co., Reading, MA, 1966.
[2] Y.Saad, “Iterative Methods for Sparse Linear Systems”, 2nd Edition, SIAM, 2003.
[3] F.G. Gustavson, "Two Fast Algorithms for Sparse Matrices: Multiplication and Permuted Transposition", ACM Trans. Math. Softw., Vol. 4, No. 3, pp. 250-269 (1978).
[4] H.S. Khalil, “The Application of Nodal Methods to PWR Analysis”, PhD. Thesis, Department of Nuclear Engineering, Massachusetts Institute of Technology, Cambridge, MA (1983).
[5] O. Abdelaziz, “The coarse mesh heterogeneous response matrix for reactor analysis”, PhD. Thesis, Department of Nuclear Engineering, The University of Michigan, 1991
[6] ANL-7416, Benchmark Problem Book, Argonne National Laboratory, 1977.