Three-dimensional Hybrid CMFD (HCMFD) Algorithm for Efficient Pin-by-Pin Transient Analysis of PWRs

Jaeha Kim and Yonghee Kim
Korea Advanced Institute of Science and Technology (KAIST)
291 Daehak-ro, Yuseong-gu, Daejeon, Korea, 34141
**Corresponding author: yongheekim@kaist.ac.kr*

1. Introduction

Since the neutron transport theories theoretically guarantee the best estimation for reactor core analyses, they have always been one of the correct answers in reactor physics field. As the computing power has increased dramatically in last few decades, the neutron transport theories are utilized in more and more studies nowadays. However, due to the extreme computational costs required for the transport calculations, they can hardly be applied for a whole core analysis, and not to mention the whole core transient analysis, which requires much more computational cost than steady-state analysis.

In these circumstances, the HCMFD (Hybrid Coarse-Mesh Finite Difference) algorithm for an efficient 3D whole core pin-by-pin diffusion analysis [1] and the GPS (GET Plus SPH) method for the cross-section corrections in pin-wise diffusion analysis [2] have been suggested. Based on these methods, it is expected that one can get a whole-core pin-wise solution with sufficient fidelity in a reasonably short computing time.

Previously, the application of the HCMFD algorithm had been limited to steady-state analyses. In this paper, the potential performances, in the aspect of computing time, of the HCMFD algorithm for the whole core transient analysis are demonstrated with control rod movement scenario.

2. Methodology

As introduced in detail in the reference [1], the one-node and two-node CMFD methods are nonlinearly coupled for the local-global iteration in the HCMFD algorithm. The global eigenvalue problem is solved by the one-node CMFD method, which enables an efficient parallel computing by domain decomposition. The local domains, fuel assemblies composed of pin-level fine meshes in a PWR analysis, are solved in parallel by the conventional two-node CMFD method based on pin-level nodal analysis, the nodal expansion method in this work.

A transient diffusion analysis starts from an initial steady state solution prepared prior to the transient analysis, and it is based on the following time-dependent neutron diffusion equation:

$$\frac{1}{v_g}\frac{d\phi_g}{dt} = \nabla\cdot D_g\nabla\phi_g - \Sigma_{r,g}\phi_g + \sum_{g'\neq g}\Sigma_{s,g'\to g}\phi_{g'} + (1-\beta)\frac{\chi_{Pg}}{k_0}\sum_{g'=1}^{\max g}\nu_{g'}\Sigma_{f,g'}\phi_{g'} + \sum_{d=1}^{\max d}\lambda_d\chi_{d,g}C_d, \quad (1)$$

$$\frac{dC_d}{dt} = \beta_d\sum_{g'=1}^{\max g}\frac{\chi_{Pg}}{k_0}\nu_{g'}\Sigma_{f,g'}\phi_{g'} - \lambda_d C_d, \quad d=1,..,\max d, \quad (2)$$

where

k_0 = multiplication factor at the initial steady state,
d = delayed neutron precursor family index,
C_d = delayed neutron precursor density.

With the implicit Euler method applied, Eq. (3) can be derived from Eq. (1) for a current time step t_s:

$$L\phi_g(t_s) + R\phi_g(t_s) = \frac{\chi_{Pg}}{k_0}(1-\beta)F\phi_g(t_s) + S\phi_g(t_s) + \sum_{d=1}^{\max g}S_d(t_s) + \frac{\phi_g(t_{s-1})}{v_g\Delta t_s}, \quad (3)$$

where

$$\begin{aligned} L\phi_g(t) &= \nabla\cdot D_g(t)\nabla\phi_g(t), \\ R\phi_g(t) &= \left(\Sigma_{r,g}(t) + \frac{1}{v_g\Delta t_s}\right)\phi_g(t), \\ S\phi_g(t) &= \sum_{g'\neq g}\Sigma_{s,g'\to g}(t)\phi_{g'}(t), \\ F\phi_g(t) &= \sum_{g'=1}^{\max g}\nu_{g'}\Sigma_{f,g'}(t)\phi_{g'}(t), \\ S_d(t) &= \chi_{d,g}\lambda_d C_d(t). \end{aligned}$$

The current step delayed neutron precursor density $C_d(t_s)$ in Eq. (3) is expressed as

$$C_d(t_s) = C_d(t_{s-1})e^{-\lambda_d\Delta t_s} + \int_{t_{s-1}}^{t_s}\frac{\chi_{d,g}\beta_d}{k_0}F\phi_g(t')e^{-\lambda_d(t_s-t')}dt', \quad (4)$$

and the integration in Eq. (4) is linearly approximated in time as

$$F\phi_g(t) \simeq F\phi_g(t_{s-1})\frac{(t_s-t)}{\Delta t_s} + F\phi_g(t_s)\frac{(t-t_{s-1})}{\Delta t_s}. \quad (5)$$

By applying Eq. (5) into the Eq. (4), the current step delayed neutron precursor density is then expressed as

$$C_d(t_s) = C_d(t_{s-1})f_{1,d} + \frac{\chi_{d,g}\beta_d F\phi_g(t_{s-1})}{k_0}f_{2,d} + \frac{\chi_{d,g}\beta_d F\phi_g(t_s)}{k_0}f_{3,d}, \quad (6)$$

where

$$
\begin{aligned}
f_{1,d} &= e^{-\lambda_d \Delta t_s}, \\
f_{2,d} &= \frac{1-e^{-\lambda_d \Delta t_s} - \lambda_d \Delta t_s e^{-\lambda_d \Delta t_s}}{{\lambda_d}^2 \Delta t_s}, \\
f_{2,d} &= \frac{e^{-\lambda_d \Delta t_s}\left(1-e^{\lambda_d \Delta t_s} + \lambda_d \Delta t_s e^{\lambda_d \Delta t_s}\right)}{{\lambda_d}^2 \Delta t_s}.
\end{aligned}
$$

Substituting Eq. (6) into Eq. (3), following fission source iteration form for the transient fixed-source problem (TFSP) is derived as

$$
\begin{aligned}
L\phi_g^{l+1}(t_s) + R\phi_g^{l+1}(t_s) = & \frac{\chi_{Pg}}{k_0}(1-\beta) F\phi_g^{l}(t_s) + S\phi_g^{l}(t_s) \\
& + \sum_{d=1}^{\max d} S_d(t_{s-1}) f_{1,d} \\
& + \sum_{d=1}^{\max d} \frac{\lambda_d \chi_{d,g} \beta_d F\phi_g(t_{s-1})}{k_0} f_{2,d} \\
& + \sum_{d=1}^{\max d} \frac{\lambda_d \chi_{d,g} \beta_d F\phi_g^{l}(t_s)}{k_0} f_{3,d} + \frac{\phi_g(t_{s-1})}{v_g \Delta t_s},
\end{aligned} \tag{7}
$$

where l is the fission source iteration index.

In this work, the exponential transformation, Eq. (8), is also applied to improve the accuracy of the implicit Euler method. The time derivative of the neutron flux becomes as Eq. (9).

$$
\phi_g(t) = \tilde{\phi}_g(t) e^{\gamma_s \Delta t} \tag{8}
$$

$$
\frac{d\phi_g(t)}{dt} = \frac{d\tilde{\phi}_g(t)}{dt} e^{\gamma_s \Delta t} + \gamma_s e^{\gamma_s \Delta t} \tilde{\phi}_g(t) \tag{9}
$$

Finally, the TFSP equation corresponding to the exponential transformation is obtained as

$$
\begin{aligned}
L\phi_g^{l+1}(t_s) + \tilde{R}\phi_g^{l+1}(t_s) = & \frac{\chi_{Pg}}{k_0}(1-\beta) F\phi_g^{l}(t_s) + S\phi_g^{l}(t_s) \\
& + \sum_{d=1}^{\max d} S_d(t_{s-1}) f_{1,d} \\
& + \sum_{d=1}^{\max d} \frac{\lambda_d \chi_{d,g} \beta_d F\phi_g(t_{s-1})}{k_0} f_{2,d} \\
& + \sum_{d=1}^{\max d} \frac{\lambda_d \chi_{d,g} \beta_d F\phi_g^{l}(t_s)}{k_0} f_{3,d} + \frac{\phi_g(t_{s-1}) e^{\gamma_s \Delta t_s}}{v_g \Delta t_s},
\end{aligned} \tag{10}
$$

where

$$
\begin{aligned}
\tilde{R}\phi_g(t_s) &= \left(\Sigma_{r,g}(t_s) + \frac{1}{v_g \Delta t_s} + \frac{\gamma_s}{v_g}\right)\phi_g(t_s), \\
\gamma_s &= \frac{1}{\Delta t_s} \ln\left(\frac{\phi(t_{s-1})}{\phi(t_{s-2})}\right).
\end{aligned}
$$

In HCMFD algorithm, this TFSP is solved at two levels, in each local domain and in the global domain. In each time step, the regional perturbations are first treated in local TFSPs and the global constants required for the global TFSP are homogenized by appropriate weighting based on the local solutions as follows:

$$
\phi_g^{I,global} = \frac{\sum_{i \in I} \phi_g^{i,local} V^{i,local}}{\sum_{i \in I} V^{i,local}}, \tag{11}
$$

$$
\Sigma_{x,g}^{I,global} = \frac{\sum_{i \in I} \Sigma_{x,g}^{i,local} \phi_g^{i,local} V^{i,local}}{\sum_{i \in I} \phi_g^{i,local} V^{i,local}}, \tag{12}
$$

$$
\beta^{I,global} = \frac{\sum_{i \in I} \beta^{i,local} \nu\Sigma_{f,g}^{i,local} \phi_g^{i,local} V^{i,local}}{\sum_{i \in I} \nu\Sigma_{f,g}^{i,local} \phi_g^{i,local} V^{i,local}}, \tag{13}
$$

$$
C_d^{I,global} = \frac{\sum_{i \in I} C_d^{i,local} V^{i,local}}{\sum_{i \in I} V^{i,local}}, \tag{14}
$$

where I and i are the node index in global and local domains respectively.

The most notable advantage of HCMFD algorithm for a transient analysis is that the fission source iteration is mainly performed in global sense, assembly-wisely, while the local TFSPs and pin-level nodal kernels do not need to be solved frequently. Since the local detailed quantities are already converged once in the steady-state analysis, repeated updates of the local solutions are not very influential to the accuracy of the transient analysis. This will be discussed again with the numerical results in the following sections.

The overall flowchart of transient HCMFD algorithm is shown in Fig 1.

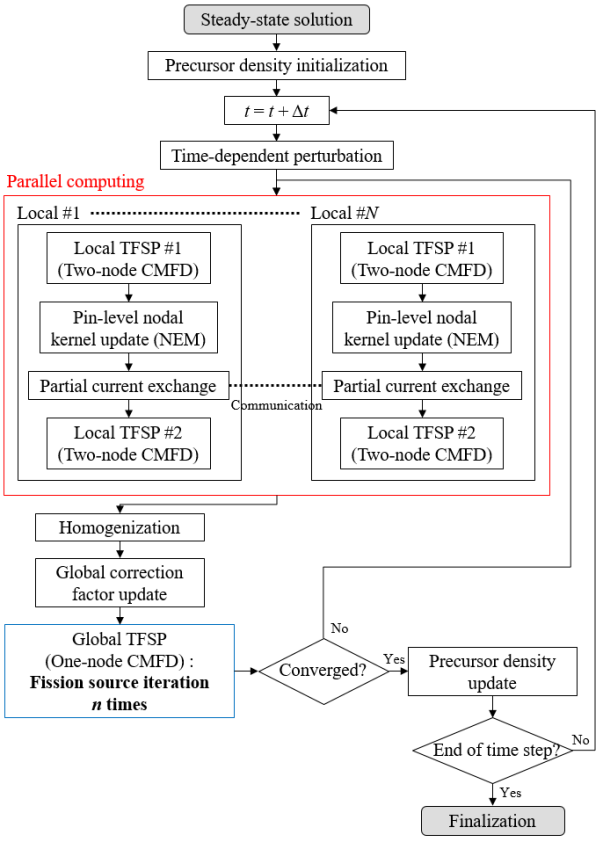

Fig. 1. Flowchart of transient HCMFD algorithm

Once the local problems are updated, the local TFSPs are solved twice, initially once and once more with the updated nodal kernel information; local correction factors and partial currents on the local boundary. It is for a better convergence between the local problems and it actually reduces possible numerical instability during the nonlinear iterations between local and global problems.

3. Numerical Results

In this work, a 3-D OPR-1000 core was treated to figure out a rough computing time for a whole core transient analysis achieved by the HCMFD algorithm. Detailed core geometry is described in Table I, and loading patterns, assembly types are shown in Figs. 2. and 3. The pin-wise two-group cross-sections of each assembly type are generated using a neutron transport code DeCART-2D [3], and a 6-group typical set of delayed neutron constants for PWR are used.

A test transient problems was solved, where the perturbation is caused by symmetric control rod movements in 8 control element assemblies (CEAs), D06, D10, F04, F12, K04, K12, M06, and M10. In the test problem, all initially inserted control rods move together as shown in Fig 4. They are initially inserted into the active core region by 100.263cm (5 axial meshes), withdrawn by 3 axial meshes in 3 seconds, then fully inserted in 1 seconds. The time-dependent position of control rods are discretized by 0.2 seconds to avoid an error caused by the time step size difference for the transient calculation.

Table I: Core geometry description

Thermal power		2815.0MWth
Radial configuration	No. of assemblies	177
	No. of reflector nodes	64
Axial configuration	Top reflector	20.95cm *1
	Active core	20.0526cm *19
	Bottom reflector	20.95cm *1
Assembly geometry		20.56cm*20.56cm (16 by 16 pins)
Total No. of coarse meshes (No. of local problems)		5,061
Total No. of fine meshes		1,295,616

The fission source convergence criterion was 10^{-7}, and the local and global problems were both solved by a BiCGstab (Biconjugate gradient stabilized) method [4]. All calculations were performed on Intel Xeon E5-2697 v3 2.60 GHz CPU with 40 physical cores. Parallel computing was performed using the OpenMP parallel algorithm [5]. All numerical results of this work are obtained by utilizing 40 cores.

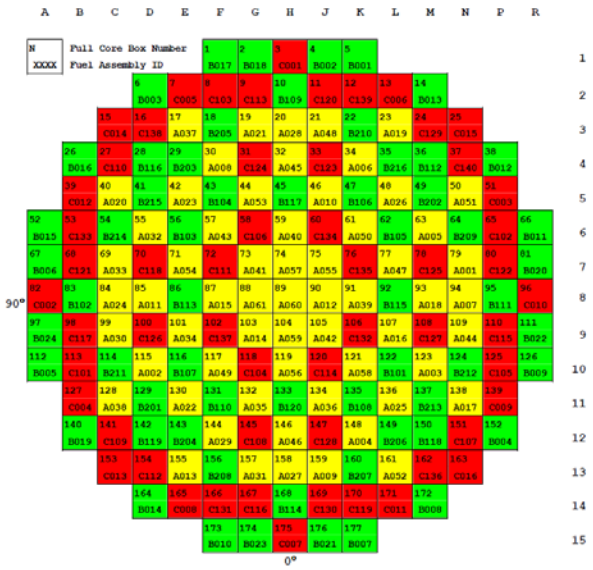

Fig. 2. Fuel loading pattern for Cycle 1

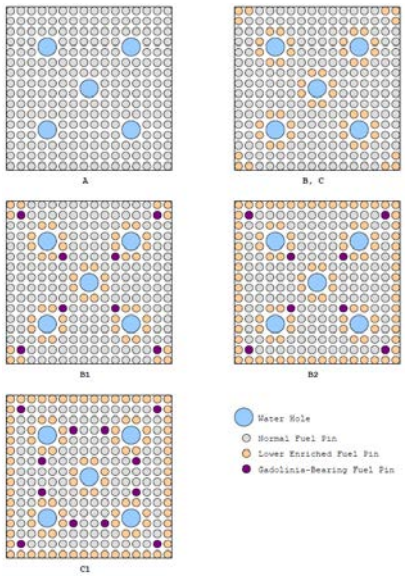

Fig. 3. Enrichment zoning pattern and burnable absorber arrangement

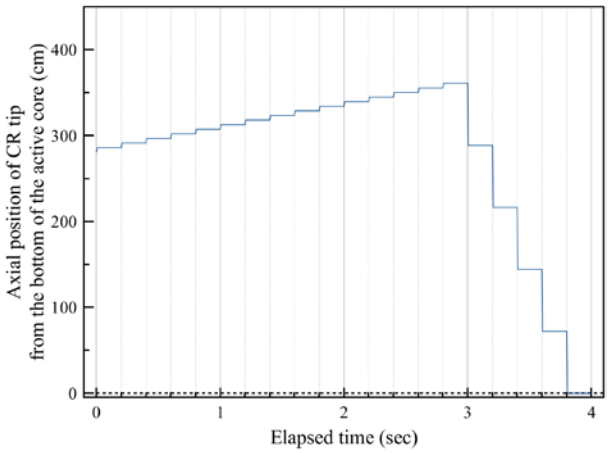

Fig. 4. Time-dependent position of control rod tip

First, the same test problem was solved with fixed time step size 0.2 seconds, and with varying the local update cycle, where it means the number of fission source iterations of global TFSP between consecutive local analyses. A large update cycle indicates a less number of local analyses in each time step. Table II shows the computing time, the average number of global fission source iteration per time step, the number of local updates per time step, and maximum power error.

It is notable that the average number of fission source iteration is not very dependent on the update cycle if the local updates are sufficiently frequent. At the same time, the computing time is highly dependent on the number of local updates. It indicates that the local calculations take major part of the computational loads, so the overall computing time can be dramatically reduced if the number of local updates can be minimized, as in the cases with update cycle larger than 50.

Table II: Numerical results depending on the cycle

Update cycle	Computing time (s)	Average No. of iterations	Average No. of local updates	Max. total power error (%)
1	3113.498	1084.1	1084.1	Ref.
2	1580.029	1084.1	542.3	0.000
5	655.329	1084.3	217.2	0.000
10	340.899	1084.6	108.8	0.000
20	189.156	1085.7	54.7	0.000
50	100.587	1110.0	22.7	0.000
100	72.499	1182.1	12.1	0.001
200	65.567	1519.7	7.8	0.007
300	71.320	1940.1	6.6	0.009
400	73.548	2163.2	5.5	0.009
500	75.792	2331.5	4.8	0.019
9999	33.155	1063.5	1.0	0.608

In the aspect of the accuracy comparing to the case with update cycle 1, only a limited level of power error was observed even with the update cycle 500, while a non-negligible error exists with only 1 local updates per time step. This trend can be seen also in the Fig. 5 that the power variations are very similar when there are more than one local updates. It shows that how frequently the local problems are updated is not a major factor that affects the accuracy of transient analysis, if they are updated more than one time. Considering the accuracy and computing time, the update cycle for the following analysis was chosen to be 100.

Table III shows the computing time with various time step sizes and with update cycle 100. Since the required number of fission source iterations are generally proportional to the power variation per time step, the average number of local updates are smaller with smaller time step size when the update cycle is fixed. The computing time for the 4-second transient is only 508 seconds with a sufficiently short time step, 0.01s.

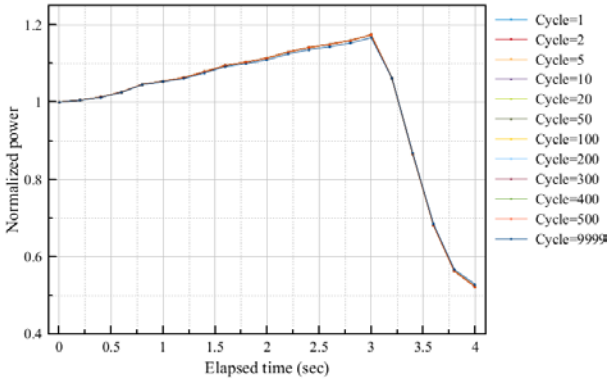

Fig. 5. Power variation during transient

Table III: Numerical results according to time step size

Time step (s)	Computing time (s)	Average No. of iterations	Average No. of local updates
0.2	72.499	1182.1	12.1
0.1	114.143	958.1	9.9
0.05	183.709	755.5	7.9
0.01	508.852	387.2	4.3

4. Conclusions

It was demonstrated that a reasonably short computing time can be achieved by the HCMFD algorithm for a 3-D whole-core pin-by-pin transient analysis of a PWR core even with a sufficiently small time step size, ~500 seconds with 0.01s time step.

ACKNOWLEDGMENTS

This work was supported by the National Research Foundation of Korea (NRF) Grant funded by the Korean Government (MSIP) (NRF-2016R1A5A1013919).

REFERENCES

[1] Jaeha Kim, and Yonghee Kim, "Development of 3-D HCMFD algorithm for efficient pin-by-pin reactor analysis," *Annals of Nuclear Energy*, vol. 127, 87-98, 2018.
[2] HwanYeal Yu, "A Combination of Generalized Equivalence Theory and Super-Homogenization for Cross-section Corrections in Pin-wise PWR Diffusion Analysis," PhD Thesis, KAIST, 2019.
[3] "DeCART 2D v1.0 User's Manual," KAERI/TR-5116/2013, Korea Atomic Energy Research Institute, 2013.
[4] H.A. Van Der Vorst, "Bi-CGSTAB: A Fast and Smoothly Converging Variant of Bi-CG for the Solution of Nonsymmetric Linear Systems", *SIAM J. Sci. and Stat. Comput.*, 13(2), 631–644, 1991.
[5] OpenMP API, "OpenMP Application Program Interface", Version 4.0, July, 2013.