

A new three-dimensional method of characteristics for the neutron transport calculation

Zhouyu Liu, Hongchun Wu, Liangzhi Cao*, Qichang Chen, Yunzhao Li

School of Nuclear Science and Technology, Xi'an Jiaotong University, No. 28, Xianning West Road, Xi'an, Shanxi 710049, PR China

ARTICLE INFO

Article history:

Received 7 July 2010

Received in revised form 8 September 2010

Accepted 25 September 2010

Keywords:

3D MOC

Modular ray tracing technique

CMFD acceleration

ABSTRACT

The method of characteristics (MOC) is a very flexible and effective method for the neutron transport calculation in a complex geometry. It has been well developed in two-dimensional geometries but meets serious difficulty in three-dimensional geometries because of the requirements of large computer memory and long computational time. Due to the demand related to the advanced reactor design for complex geometries, an efficient and flexible three-dimensional MOC is needed. This paper presents a modular ray tracing technique to reduce the amount of the ray tracing data and consequently reduce the memory. In this method, the object geometry is dissected into many cuboid cells by a background mesh. Typical geometric cells are picked out and ray traced, and only the ray tracing data in these typical cells is stored. Furthermore, the Coarse Mesh Finite Difference (CMFD) acceleration method is employed to save computing time. Numerical results demonstrate that the modular ray tracing technique can significantly reduce the amount of ray tracing data, and the CMFD acceleration is effective in shorting the computing time.

© 2010 Elsevier Ltd. All rights reserved.

1. Introduction

The method of characteristics (MOC) is an effective tool for the neutron transport calculations. It considers only a finite number of discrete directions (with quadrature sets) and calculates mesh-average angular flux by sweeping. For a given direction, each of several parallel rays is traced to provide mesh-average ray angular flux and outgoing ray angular flux by analytic integration along the tracing ray. The ray-wise integration allows flexibility of the mesh shapes. It can take any shape and mixture of shapes as in Monte-Carlo methods (Cho, 2005).

The MOC has been extensively developed for 2D transport problems. The earliest code of the MOC for complex 2D geometries was developed by Askew (1972), and the first reactor analysis code with the MOC was CACTUS developed in (Halsall, 1980). However the MOC was not widely used at that time because of the high memory requirements and slow processor speeds. In the 1990s, the MOC became popular again for the quick development of computer technology and the need to deal with complex geometries. And a lot of codes based on the MOC were developed, such as

KRAM (Knott, 1990), CHAR (Goldberg et al., 1995), CRX (Hong and Cho, 1998), MOCC (Roy, 1998), and DECART (Cho et al., 2002).

As the development of nuclear engineering, especially the reach of the next generation fission reactors and fusion reactors is popular around the world, the geometries of new reactors become more and more complicated, such as the High Temperature Gas-cooled Reactor (HTGR) and the International Thermonuclear Experimental Reactor (ITER). To get the reliable and precise results, three-dimensional transport calculations are needed. The 3D MOC is a very good choice due to its geometric flexibility.

However, direct MOC calculations for three-dimension problems require a tremendous amount of memory (Chai et al., 2009) and very long computing time. As one solution, a 2D + 1D strategy is proposed, which combines 2D MOC for the radial calculation and 1D calculation in the axial direction. This method performs well when the geometry keeps the same in the axial direction (Lee and Cho, 2006). But it is not easy to apply it to the problems with unstructured axial geometry, such as the Test Blanket Module (TBM) of the International Thermonuclear Experimental Reactor (ITER).

Therefore, a 3D modular ray tracing technique is adopted, which can not only hold the geometry flexibility of the MOC but also reduce the tremendous memory requirement. It is an extension of the 2D modular ray tracing technique firstly proposed by Filippone et al. (1981) and widely used in 2D MOC codes (CRX,

* Corresponding author. Tel.: +86 29 82663285; fax: +86 29 82667802.

E-mail address: caolz@mail.xjtu.edu.cn (L. Cao).

DECART, et al.). For the modular ray tracing technique, the object geometry is dissected into many cuboid cells by a background mesh. Typical geometric cells are picked out and ray traced. Only in these typical cells, the ray tracing data is stored. It thus needs much less memory, nevertheless faces two difficulties. (1) The positions of tracing lines should be the same in all typical cells. (2) The tracing lines must connect at the interface.

For three-dimensional transport methods, there are several kinds of angular quadrature sets, such as level symmetric angular quadrature sets. The angles of these quadrature sets are all calculated by given formulas. However, the angles obtained in modular ray tracing technique do not conform to them, so new angular quadrature sets are also derived.

The Coarse Mesh Finite Difference (CMFD) acceleration method is popular in the fast solution of nodal diffusion equations (Turinsky et al., 1994; Smith, 1983), and widely used to accelerate the transport calculations with very good results (Smith and Rhodes, 2000). To reduce the long computing time, the CMFD is employed to acceleration the 3D MOC calculation. The results of the CMFD for 3D MOC also demonstrate that it is very effective in reducing the number of iterations and computing time.

In Section 2, we review the MOC. The detailed 3D modular technique and angular quadrature sets are introduced respectively in Sections 3 and 4. Section 5 presents the CMFD acceleration method. In Section 6, some numerical problems are tested. Finally, we summarize and discuss our results in Section 7.

2. Step characteristics

The Boltzmann transport equation describing the neutron transport behavior can be written in the following form along the path, namely the characteristics.

$$\frac{d\phi_g(s, \vec{\Omega}_m)}{ds} + \Sigma_{t,g}(s)\phi_g(s, \vec{\Omega}_m) = Q_g(s, \vec{\Omega}_m). \quad (1)$$

The source consists of the scattering and fission parts:

$$Q_g(s, \vec{\Omega}_m) = \frac{1}{4\pi} \sum_{g'} \Sigma_{s,g' \rightarrow g}(s) \bar{\phi}_{g'}(s) + \frac{\chi_g}{4\pi k_{eff}} \sum_{g'} \nu \Sigma_{f,g'}(s) \bar{\phi}_{g'}(s). \quad (2)$$

After the domain is divided into some sub-domains and both the material and the neutron source are assumed to be flat, the outgoing angular flux from sub-domain i along the path k can be given by

$$\begin{aligned} \phi_{g,i,k}^{out}(\vec{\Omega}_m) &= \phi_{g,i,k}^{in}(\vec{\Omega}_m) \exp(-\Sigma_{t,g,i} s_{i,k}) \\ &+ \frac{Q_{g,i}(\vec{\Omega}_m)}{\Sigma_{t,g,i}} [1 - \exp(-\Sigma_{t,g,i} s_{i,k})], \end{aligned} \quad (3)$$

where $s_{i,k}$ is the traveling distance in the sub-domain i of the path k ; and $\phi_{g,i,k}^{in}(\vec{\Omega}_m)$ is the incoming angular flux.

The average angular flux of the segment i, k $\bar{\phi}_{g,i,k}(\vec{\Omega}_m)$ can be obtained by integrating the Eq. (3) along the path k :

$$\bar{\phi}_{g,i,k}(\vec{\Omega}_m) = \frac{Q_{g,i}(\vec{\Omega}_m)}{\Sigma_{t,g,i}} + \frac{\phi_{g,i,k}^{in}(\vec{\Omega}_m) - \phi_{g,i,k}^{out}(\vec{\Omega}_m)}{\Sigma_{t,g,i} s_{i,k}}. \quad (4)$$

The sub-domain average angular flux $\bar{\phi}_{g,i}(\vec{\Omega}_m)$ is given by:

$$\bar{\phi}_{g,i}(\vec{\Omega}_m) = \frac{\sum_k \bar{\phi}_{g,i,k}(\vec{\Omega}_m) s_{i,k} \delta A_k}{\sum_k s_{i,k} \delta A_k}. \quad (5)$$

In addition, the traveling distance should be corrected since the numerically calculated volume $\sum_k s_{i,k} \delta A_k$ is not equal to the true volume V_i :

$$s'_{i,k} = s_{i,k} \frac{V_i}{\sum_k s_{i,k} \delta A_k}. \quad (6)$$

3. Modular ray tracing technique

The scheme of the 3D modular ray tracing technique for the MOC is shown in Fig. 1.

The modular ray tracing technique consists of two main parts: modularization and ray tracing. Modularization consists of steps 1–4 in Fig. 1. The object geometry is dissected into many cuboid cells by a background mesh. However, most of them are usually the same, so only the typical cells are considered. And the object geometry can be arrayed by these typical cells (see step 6) again. This part can be realized by commercial modeling software, such as AutoCAD (Chen et al., 2008).

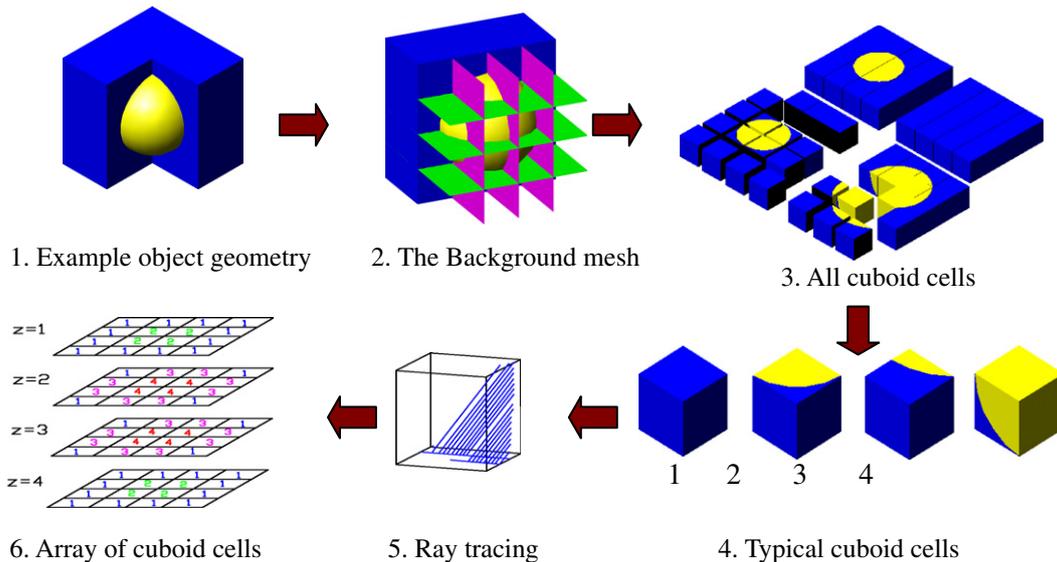


Fig. 1. Modular ray tracing process.

The second part is the 3D ray tracing (step 5). It is the key point of the modular ray tracing technique, which make sure that the segments will connect at interface between different typical cells. The following are the details of the 3D ray tracing process in direction $\vec{\Omega}(\varphi, \theta)$ for a typical cell.

3D modular ray tracing also contains two steps, dealing with the azimuthal angle and polar angle respectively.

The azimuthal angle is constrained by the requirement of modular ray tracing. The following relation is proposed by [Filippone et al. \(1981\)](#).

$$\tan(\varphi) = (\Delta y / \Delta x) \frac{nx}{ny}, \tag{7}$$

and

$$\delta_\varphi = \frac{\Delta x}{nx} \sin(\varphi) = \frac{\Delta y}{ny} \cos(\varphi), \tag{8}$$

where Δx is the length of classical cell in x direction; Δy is the length in y direction; nx is the number of points in x direction; ny is the number of points in y direction.

We extend this idea to deal with the polar angle. So the solid slope lines in [Fig. 2](#) are no longer the segments, but the cutaway

sections perpendicular to the x - y plane. The sections are arrayed by order (see [Fig. 3](#)).

With the same options, we will get the solid slope lines in [Fig. 3](#). It will lead to a modular ray tracing system with self-repeating ray distributions and returning to the same starting point after reflection. This makes 3D whole-core ray tracing possible without any approximation on the interfaces between cells (see [Fig. 4](#)) and reflective boundary faces (see [Fig. 5](#)).

Similar to step 1, the polar angle can be obtained by:

$$\Delta l_\varphi = \sum_k s_k, \tag{9}$$

$$\tan(\theta') = (\Delta z / \Delta l_\varphi) \frac{n_\varphi}{nz}, \tag{10}$$

$$\theta = \frac{\pi}{2} - \theta', \tag{11}$$

and

$$\delta_\theta = \frac{\Delta l_\varphi}{n_\varphi} \sin(\theta') = \frac{\Delta z}{nz} \cos(\theta'), \tag{12}$$

where Δl_φ is the sum length of the faces in the azimuthal angle direction; s_k is the length of the solid slope segments in [Fig. 2](#); Δz is the size of the classical cell in z direction; nz is the number of the points on Δz ; n_φ is the number of points on Δl_φ .

Then, the cross-sectional area of the direction $\vec{\Omega}(\varphi, \theta)$ is thus given by:

$$\delta A(\vec{\Omega}) = \delta_\varphi \delta_\theta. \tag{13}$$

4. Angular quadrature sets

Now, for 3D problems, there are several kinds of angular quadrature sets, such as level symmetric angular quadrature sets. The angles of these quadrature sets are all calculated by given formulas, but the angles obtained by Eqs. (7), (10), and (11) do not

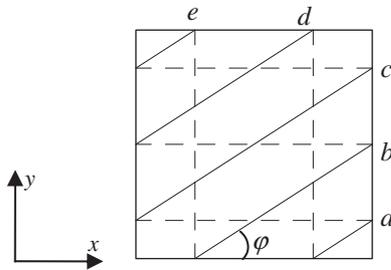


Fig. 2. Schematic diagram of step 1.

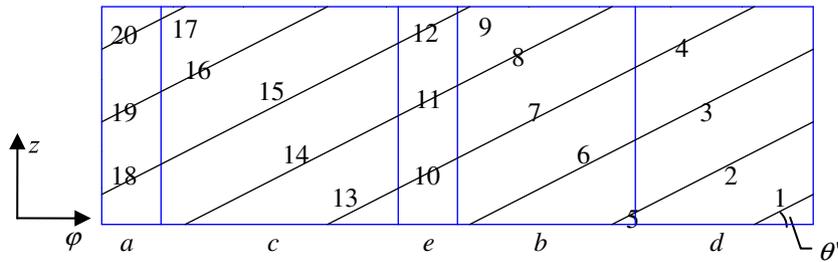


Fig. 3. Schematic diagram of polar angle.

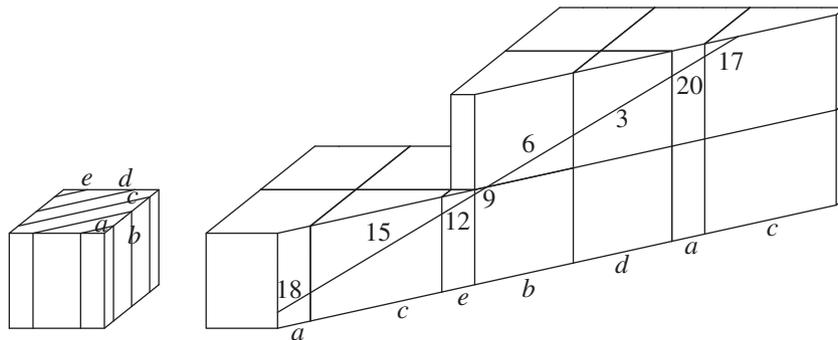


Fig. 4. Treatment of the interfaces between cells.

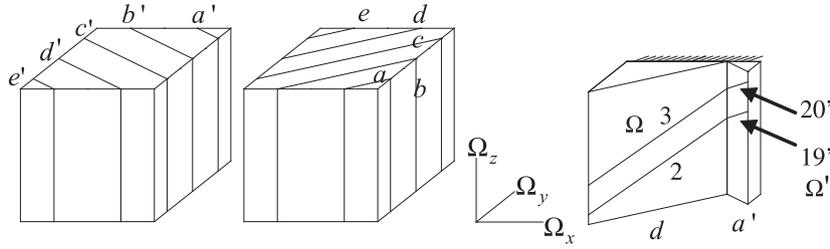


Fig. 5. Treatment of the reflective boundary faces.

conform to them. So we derive new relations to calculate the weight of directions.

The direction $\bar{\Omega}(\varphi, \theta)$ is obtained by Eqs. (7), (10), and (11); M is the number the directions in one octant; N has the same meaning of S_N order, and the relation between M and N is (see Fig. 6):

$$M = \frac{(N/2)(N/2 + 1)}{2}. \quad (14)$$

Because $\{Y_{2n}^{2m}(\varphi, \theta) : n = 0, 1, \dots, m = 0, 1, \dots, n\}$ is a complete orthogonal system in the space $[\varphi \in [0, \pi/2], \theta \in [0, \pi/2]]$, it can be used to expand the functions defined in space $[\varphi \in [0, \pi/2], \theta \in [0, \pi/2]]$.

Assuming function $f(\varphi, \theta)$ is defined in $[\varphi \in [0, \pi/2], \theta \in [0, \pi/2]]$, so it can be expanded as:

$$f(\varphi, \theta) = \sum_n \sum_{m=0}^n a_{n,m} Y_{2n}^{2m}(\varphi, \theta). \quad (15)$$

With the integral in the domain of definition written in the form of numerical integration:

$$\int_0^{\pi/2} \int_0^{\pi/2} f(\varphi, \theta) \sin \theta \, d\theta \, d\varphi = \sum_k w_k f(\theta_k, \varphi_k), \quad (16)$$

Substituting Eq. (15) into Eq. (16), and making the both sides of the equation identical, it will derive

$$\int_0^{\pi/2} \int_0^{\pi/2} Y_{2n}^{2m}(\varphi, \theta) \sin \theta \, d\theta \, d\varphi = \sum_k w_k Y_{2n}^{2m}(\theta_k, \varphi_k). \quad (17)$$

When the direction discrete order is N , the formula to calculate the weight of directions is:

$$\begin{aligned} \sum_k^M w_k Y_{2n}^{2m}(\theta_k, \varphi_k) &= \int_0^{\pi/2} \int_0^{\pi/2} Y_{2n}^{2m}(\varphi, \theta) \sin \theta \, d\theta \, d\varphi \\ (n = 0, 1, \dots, N; m = 0, 1, \dots, n; \\ M &= (N/2)(N/2 + 1)/2) \end{aligned} \quad (18)$$

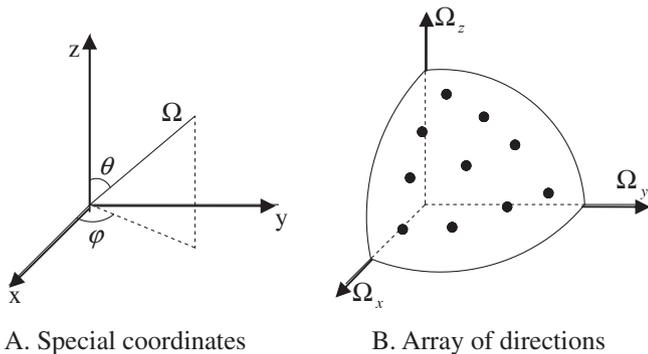


Fig. 6. Angular quadrature set in the MOC.

This new method can calculate the weight of arbitrary directions if the sum number of directions in one octant satisfies Eq. (14). The numerical results of the test problems in Section 6 prove this new method right.

5. Coarse Mesh Finite Difference (CMFD) acceleration

In the CMFD method, a current correction coefficient is introduced to preserve the interface currents between coarse mesh cells, which are got from the solutions of transport sweep.

The coarse mesh cell homogenized multi-group constants can be calculated from the heterogeneous regional cross sections and scalar fluxes:

$$\Sigma_{x,g,(i,j,k)} = \frac{\sum_{l \in (i,j,k)} V_l \Sigma_{x,g,l} \phi_{g,l}^n}{\phi_{g,(i,j,k)}^n V_{(i,j,k)}}, \quad (19)$$

and

$$\phi_{g,(i,j,k)}^n = \frac{\sum_{l \in (i,j,k)} V_l \phi_{g,l}^n}{V_{(i,j,k)}}, V_{(i,j,k)} = \sum_{l \in (i,j,k)} V_l. \quad (20)$$

where $\phi_{g,l}^n$ is the scalar flux of region l that is determined from the MOC transport calculation.

The current correction coefficient is obtained from the interface current and coarse mesh flux.

$$\hat{D}_{g,u+1/2} = \frac{J_{g,u+1/2}^n + \tilde{D}_{g,u+1/2} (\phi_{g,u+1}^n - \phi_{g,u}^n)}{\phi_{g,u+1}^n + \phi_{g,u}^n} \quad (u = x, y, z), \quad (21)$$

where

$$J_{g,u+1/2}^n = \frac{\sum_m w_m \Omega_m \varphi_{m,g,u+1/2}^n}{8}, \quad (22)$$

$$\tilde{D}_{g,u+1/2} = 2 \frac{(D_{g,u+1/2}/h_u)(D_{g,u+1/2}/h_{u+1})}{D_{g,u+1/2}/h_u + D_{g,u+1/2}/h_{u+1}}. \quad (23)$$

Here $\tilde{D}_{g,u+1/2}$ is the nodal coupling coefficient determined by the coarse mesh cell homogenized diffusion constant: $D_{g,u} = \frac{1}{3\Sigma_{t,g,u}}$, $J_{g,u+1/2}^n$ is the coarse mesh cell interface average current.

The CMFD equation for coarse mesh cell (i, j, k) is

$$J_{g,u+1/2}^{n+1/2} = -\tilde{D}_{g,u+1/2} (\phi_{g,u+1}^{n+1/2} - \phi_{g,u}^{n+1/2}) - \hat{D}_{g,u+1/2} (\phi_{g,u+1}^{n+1/2} + \phi_{g,u}^{n+1/2}), \quad (24a)$$

$$\begin{aligned} \sum_{u=i,j,k} (J_{g,u+1/2}^{n+1/2} - J_{g,u-1/2}^{n+1/2}) + V_{(i,j,k)} \left(\sum_t - \sum_s \right) \phi_{(g,i,j,k)}^{n+1/2} \\ = V_{(i,j,k)} q_{(i,j,k)} \end{aligned} \quad (24b)$$

If the vacuum boundary condition is given on $u + 1/2$, and reflective boundary condition on $u - 1/2$, the coarse mesh surface currents can be replaced by:

$$J_{g,u+1/2}^{n+1/2} = \frac{\phi_{g,(i,j,k)}^{n+1/2}}{\phi_{g,(i,j,k)}^n} J_{g,u+1/2}^n \quad (25)$$

$$J_{g,u-1/2}^{n+1/2} = 0. \quad (26)$$

After the CMFD calculation, the coarse mesh cell average scalar fluxes are obtained. Since the subsequent MOC transport calculation requires updated regional scalar fluxes and boundary incoming angular fluxes from the CMFD calculation, the regional scalar fluxes and boundary angular fluxes are updated by:

$$\phi_{g,l}^{n+1/2} = \frac{\phi_{g,(i,j,k)}^{n+1/2}}{\phi_{g,(i,j,k)}^n} \phi_{g,l}^n \quad (27)$$

$$\varphi_{g,m}^{n+1/2} = \frac{\phi_{g,(i,j,k)}^{n+1/2}}{\phi_{g,(i,j,k)}^n} \varphi_{g,m}^n \quad (28)$$

where n is the iteration index.

6. Numerical results

Based on the 3D modular characteristics method described above, a code 3D MMOC is developed. In this section, three test problems are presented. The first two problems are 2D/3D super-cell problems, and the third is a KUCA core benchmark. They are presented to show that, as the increase of dimension and size, the storage save tends to be more significant. The speedup of the CMFD acceleration is also listed in tables. In the test calculations, the directions are selected similar to S8, and the corresponding angular quadrature set is listed in Table 1.

6.1. The accuracy and storage of 3D MOC calculations

6.1.1. 2D cell problem

This is a 2D super-cell problem. It contains two materials, and reflective boundary conditions are applied in all side. The central homogenized fuel region is surrounded by water moderator as presented in Fig. 7. The 2-group cross section set and energy group structure is listed in Table 2.

The reference solution is given by MCNP. Table 3 shows the comparison of k -infinity between 3D MMOC and MCNP, and 3D MMOC performs well. Table 4 presents the region-averaged scalar flux distribution, and the differences are all less than 1%.

In this problem, there is only one typical cell. The storage of ray tracing data is 499 kB, when $N = 8$, $\delta A = 2.6E-3 \text{ cm}^2$, and the size of the cell is 0.5 cm. If we do the global ray tracing with the same condition, it will cost more than 250 (16×16) times storage than the first case (see Table 5).

Table 1
S8 angular quadrature set.

	ϕ	θ	w
1	1.3521	1.3572	1.170E-01
2	0.9420	1.3537	9.219E-02
3	0.6288	1.3537	9.219E-02
4	0.2187	1.3572	1.170E-01
5	1.3102	0.9543	9.145E-02
6	0.7854	0.9553	1.000E-01
7	0.2606	0.9543	9.145E-02
8	1.2120	0.6604	9.081E-02
9	0.3588	0.6604	9.081E-02
10	0.7854	0.3045	1.170E-01

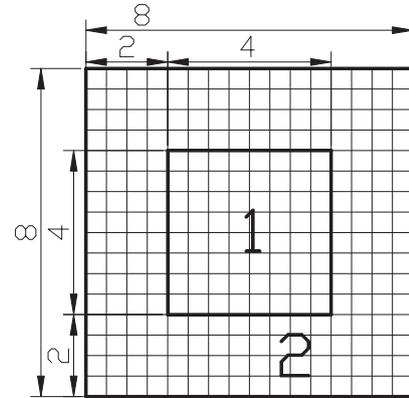


Fig. 7. 2D cell problem-geometry description.

Table 2
2D cell problem-cross section data.

Region	Group	$\nu\Sigma_f/\text{cm}^{-1}$	$\Sigma_{a-1}/\text{cm}^{-1}$	$\Sigma_{a-2}/\text{cm}^{-1}$	Σ_t/cm^{-1}	χ
1	1	6.203E-3	1.780E-1	1.002E-2	1.96647E-1	1.0
	2	1.101E-1	1.089E-3	5.255E-1	5.96159E-1	0.0
2	1	0.0	1.995E-1	2.188E-2	2.22064E-1	-
	2	0.0	1.558E-3	8.783E-1	8.87874E-1	-

Table 3
2D cell problem-comparison of k -infinity.

Code	k -effective	Difference (%)
Monte-Carlo	1.03815 ± 0.00026	-
3D MMOC	1.03741	-0.07

Table 4
2D cell problem-region-averaged scalar flux.

Method		Core	Reflector
Monte-Carlo	1G	8.25E-01 ^a	7.15E-01
	2G	0.0003	0.0003
3D MMOC	1G	5.21E-01	6.37E-01
		0.0003	0.0003
	2G	8.17E-01	7.14E-01
		5.22E-01	6.11E-01
		0.85%	-0.64%

^a Normalized by $\int \nu \Sigma_f \phi dV = 1$.

6.1.2. 3D cell problem

It is a 3D cell problem, containing two materials. Reflective boundary conditions are considered in all sides. The fuel region is in the center of water moderator as presented in Fig. 8. The size of fuel region is 4 cm, and the size of moderator is 6 cm. The 2-group cross section set of the materials is listed in Table 2.

Table 6 gives the comparison of k -infinity of 3D MMOC with MCNP, and 3D MMOC presents satisfied accuracy. Table 7 shows the scalar flux distribution of every material region. The biggest difference is -0.90%.

Table 5
2D cell problem-comparison of the storage.

Case	Storage	Ratio
Global ray tracing	120 MB	250
Modular ray tracing	499 KB	1

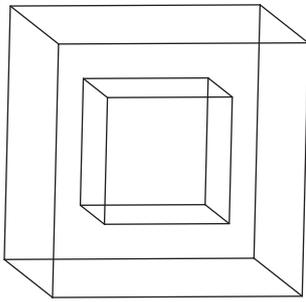


Fig. 8. 3D cell problem-geometry description.

Table 6
3D cell problem-comparison of k -infinity.

Code	k -effective	Difference (%)
Monte-Carlo	1.11707 ± 0.00027	–
3D MMOC	1.114721	0.21

The results in Tables 6–8 are got with the condition that $N = 8$, $\delta A = 2.6E-3 \text{ cm}^2$, and the size of the sub-domain is 0.5 cm. The storage of ray tracing data is 500 kB. If the global ray tracing is done with the same condition, the storage will be about 1728 ($12 \times 12 \times 12$) times larger the modular case.

6.1.3. KUCA core benchmark

This benchmark was published by Takeda and Ikeda (1991). It describes the core model of the Kyoto University Critical Assembly (KUCA). The size of the whole reactor core is $50 \text{ cm} \times 50 \text{ cm} \times 50 \text{ cm}$, with three material regions: core fuel, control rod, and reflector. Fig. 9 shows the core configuration. The 2-group cross section set and energy group structure used in calculation are given in Table 9. The following two cases are considered:

- Case 1: The control rod position is empty (void).
- Case 2: The control rod is inserted.

When calculated, the mesh size is $1 \text{ cm} \times 1 \text{ cm} \times 1 \text{ cm}$. Table 10 gives the mean values and standard deviations of k -effective and control rod worth, and the reference value is got from the TAKEDA benchmark published in 1991. The control rod worth is calculated by $(k - k')/(kk')$. The results of the MOC is calculated with the condition that: $N = 8$, $\delta A = 0.01 \text{ cm}^2$. Though the k -effective results of the MOC are smaller than the reference values, the control rod worth agrees very well with the Monte-Carlo result. The region-averaged scalar flux distribution is listed in Table 11. In this problem, the biggest difference is more than -7.38% . It is because that the fine-mesh size is 1 cm, almost as long as the mean free path. From the results of 2D/3D cell problems, we can find that if the cell size is less than $1/2$ of the mean free path, the flux difference would be much better.

Table 7
3D cell problem-region-averaged scalar flux.

Method		Core	Reflector
Monte-Carlo	1G	2.15E-01 ^a	1.99E-01
		0.0003	0.0003
	2G	1.30E-01	1.46E-01
3D MMOC		0.0003	0.0003
	1G	2.13E-01	1.99E-01
		-0.90%	0.15%
	2G	1.30E-01	1.45E-01
		0.08%	-0.44%

^a Normalized by $\int v \Sigma_f \phi dV = 1$.

Table 8
3D cell problem-comparison of the storage.

Case	Storage	Ratio
Global ray tracing	862 MB	1728
Modular ray tracing	499 KB	1

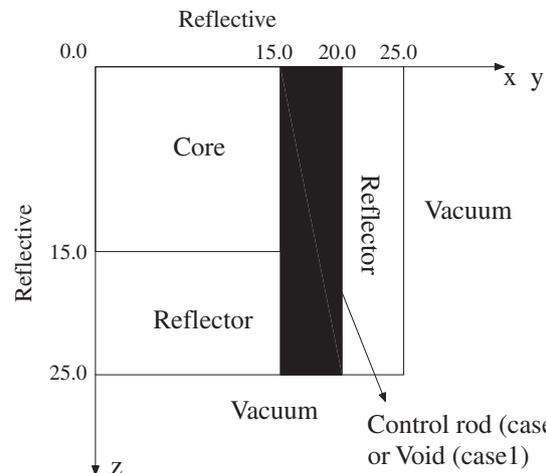
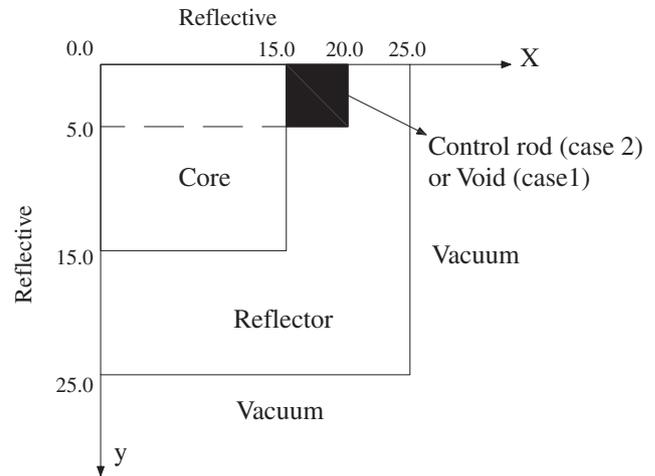


Fig. 9. KUCA core benchmark-core configuration.

Table 12 compares the storage of the ray tracing data between the global ray tracing case and modular ray tracing case. In the KUCA benchmark, the geometry is discretized into 15,625 ($25 \times 25 \times 25$) cells, and every of these cells is the same. Compared to the global ray tracing case, only the ray tracing data in one cell is stored in the modular ray tracing case. So the storage needed will be much more less. The storage of these three problems demonstrates that modular ray tracing can significantly reduce the ray tracing data when dealing with large repeated-geometry problems.

6.2. The performance of the CMFD acceleration

In this part, the test problems are the same with Section 6.1. Because the k -effective and fluxes are almost the same with that before using the CMFD acceleration, only the speedup of the CMFD is listed. In the tables, the iterations denote the number of inner iteration.

Table 9
KUCA core benchmark-two-group cross sections and energy group structure.

Region	Group	Σ_f/cm^{-1}	$\nu\Sigma_f/\text{cm}^{-1}$	$\Sigma_{1-1}/\text{cm}^{-1}$	$\Sigma_{1-2}/\text{cm}^{-1}$	χ
Core fuel	1	2.23775E-1	9.09319E-3	1.92423E-1	2.28253E-2	1.0
	2	1.03864E+0	2.90183E-1	0.0	8.80439E-1	
Control rod	1	8.52325E-2	0.0	6.77241E-2	6.45461E-5	-
	2	2.17460E-1	0.0	0.0	3.52358E-2	
Reflector	1	2.50367E-1	0.0	1.93446E-1	5.65042E-2	-
	2	1.64482E+0	0.0	0.0	1.62452 E+0	
Empty (void)	1	1.28407E-2	0.0	1.27700E-2	2.40997E-5	-
	2	1.20676E-2	0.0	0.0	1.07387E-2	

Table 10
KUCA core benchmark-average k -effective and control rod worth.

Method	Rod-out	Rod-in	Control rod worth
Monte-Carlo	0.9778 ± 0.0005	0.9624 ± 0.0005	1.64E-2 $\pm 0.07E-2$
3D MMOC	0.9764 (-0.143%)	0.9609 (-0.156%)	1.65E-2 (0.6%)

Table 11
KUCA core benchmark-region-averaged scalar flux-rod out.

Method		Core	Reflector	Void
Monte-Carlo	1G	4.78E-03 0.06%	5.97E-04 0.08%	1.45E-03 0.21%
	2G	8.78E-04 0.08%	9.20E-04 0.11%	9.77E-04 0.34%
3D MMOC	1G	4.74E-03 ^a -0.84%	5.98E-04 0.09%	1.45E-03 0.25%
	2G	8.73E-04 -0.62%	8.52E-04 -7.38%	9.34E-04 -4.38%

^a Normalized by $\int \nu \Sigma_f \phi dV = 1$.**Table 12**
KUCA core benchmark-comparison of the storage.

Case	Storage	Ratio
Global ray tracing	8437.5 MB	15,625
Modular ray tracing	540 KB	1

Table 13
2D cell problem-results of the CMFD acceleration.

Acceleration	Iterations	Computing times (s) ^a	Speedup
No	399	78.200	-
CMFD	21	4.78	16.4

^a On Core2 2.66-GHz machine.**Table 14**
3D cell problem-results of the CMFD acceleration.

Acceleration	Iterations	Computing times (s) ^a	Speedup
No	360	444.49	-
CMFD	26	34.81	12.7

^a On Core2 2.66-GHz machine.

6.2.1. 2D cell problem

The problem consists of 4×4 coarse mesh cells, in which one coarse mesh cell contains 4×4 fine mesh cells. Table 13 shows

Table 15
KUCA core benchmark-results of the CMFD acceleration.

Benchmark case	Acceleration	Iterations	Computing times (s) ^a	Speedup
Rod-out	No	433	4881	-
	CMFD	63	868	5.6
Rod-in	No	445	5081	-
	CMFD	44	603	8.4

^a On Core2 2.66-GHz machine.

the results of calculations, and it indicates that the CMFD is effective to accelerate the MOC in terms of the number of iterations and computing times.

6.2.2. 3D cell problem

This problem has $4 \times 4 \times 4$ coarse mesh cells, and a coarse mesh cell contains $3 \times 3 \times 3$ fine mesh cells. Table 14 shows the results of the CMFD. In both 2D and 3D cell problems the speedup is more than 10.

6.2.3. KUCA core benchmark

In the acceleration calculation, there are $25 \times 25 \times 25$ coarse mesh cells, and a coarse mesh cell contains one fine mesh cell. For the rod-out case, the time speedup is 5.6 while the speedup of the rod-in case is 8.4 (see Table 15). It is because that, when control rod is inserted, the flux distribution is flatter than that of rod-out case. And the CMFD will be more efficient. It is also the reason why the speedup of the first two problems is larger than that of the KUCA benchmark.

7. Conclusions

A 3D modular ray tracing technique is proposed in this paper. On one hand, compared with global ray tracing, the ray tracing data storages required by this modular technique are only 1/250 and 1/1000 in the 2D/3D test cell problems respectively. And in the KUCA benchmark, the ratio is up to 15,625. Therefore, this effect tends to be more significant while the problem scale increases, because of the appearance of more repetitious cells. On the other hand, this modular technique can avoid approximations on the interfaces and outer boundaries.

Since the present quadrature sets can not conform to the discrete angles in this modular ray tracing technique, a series of new angular quadrature sets are derived mathematically and computed numerically. The derivation process and numerical results show that it can provide correct quadrature sets of any order needed by the modular ray tracing technique.

To reduce the computing time, the CMFD acceleration method is implemented in 3D MOC calculations, and the results showed

that it is very effective in reducing the number of iterations and computing time.

In the future, we will focus on the secondary development based on AutoCAD to improve the capability of modeling complicated geometry.

Acknowledgements

This research was carried out under the financial support of the National Science Foundation of China (Approved numbers 10976021 and 10875094) and the National High Technology Research and Development Program ('863' Program) of China (Approved number 2009AA050705).

References

- Askew, J.R., 1972. A characteristics formulation of the neutron transport equation in complicated geometries. Report AEEW-R-1108, UK Atomic Energy Authority Establishment, Winfrith.
- Chai, X., Yao, D., Wang, K., 2009. The linear source approximation in three dimension characteristics method. International Conference on Mathematics, Computational Methods & Reactor Physics (M&C 2009). American Nuclear Society, Saratoga Springs, New York, May 3–7.
- Chen, Q., Wu, H., Cao, L., 2008. Auto MOC—a 2D neutron transport code for arbitrary geometry based on the method of characteristics and customization of AutoCAD. Nucl. Eng. Des. 238, 2828.
- Cho, N.Z., 2005. Fundamentals and recent developments of reactor physics methods. Nucl. Eng. Technol. 37, 25.
- Cho, J.Y., Joo, H.G., Kim, K.K., Zee, S.Q., 2002. Cell based CMFD formulation for acceleration of whole-core method of characteristics calculations. J. Korean Nucl. Soc. 34 (3), 250–258. June.
- Filippone, W.L., Woolf, S., Lavigne, R.J., 1981. Particle transport calculations with the method of streaming rays. Nucl. Sci. Eng. 77, 119.
- Goldberg, L., Vujic, J., Leonard, A., Tackowski, R., 1995. The method of characteristics in general geometry. Trans. Am. Nucl. Soc. 73, 173.
- Halsall, M.J., 1980. CACTUS, A characteristic solution to the neutron transport equation in complicated geometries. AEEW-R 1291, UK Atomic Energy Authority Establishment, Winfrith.
- Hong, S.G., Cho, N.Z., 1998. CRX: a code for rectangular and hexagonal lattices based on the method of characteristics. Annu. Nucl. Energy 25, 545.
- Knott, D., 1990. KRAM: A Lattice Physics Code for Modeling the BWR Fuel Designs. Doctoral thesis, Pennsylvania State University.
- Lee, G.S., Cho, N.Z., 2006. 2D/1D fusion method solutions of the three-dimensional transport OECD benchmark problem C5G7 MOX. Prog. Nucl. Energy 48, 410–423.
- Roy, R., 1998. The cyclic characteristics method. International Conference Physics of Nuclear Science and Technology, Long Island, 5–8 October.
- Smith, K.S., 1983. Nodal method storage reduction by nonlinear iteration. Trans. Am. Nucl. Soc. 44, 265.
- Smith, K.S., Rhodes III, J.D., 2000. CASMO characteristics method for two-dimensional PWR and BWR core calculation. Trans. Am. Nucl. Soc. 83, 294.
- Takeda, T., Ikeda, H., 1991. 3-D neutron transport benchmarks. J. Nucl. Sci. Technol. 28 (7), 656–669.
- Turinsky, P.J. et al., 1994. NESTLE: a few-group neutron diffusion equation solver. EGG-NRE-11406, Electric Power Research Center, North Carolina State University.