International Conference on Mathematics, Computational Methods & Reactor Physics (M&C 2009)
Saratoga Springs, New York, May 3-7, 2009, on CD-ROM, American Nuclear Society, LaGrange Park, IL (2009)

THE LINEAR SOURCE APPROXIMATION IN THREE DIMENSION CHARACTERISTICS METHOD

X. M. Chai[*] and K. Wang
Department of Engineering Physics
Tsinghua University in China
Department of Engineering Physics, Tsinghua University, Beijing, China, 100084
chaixm@163.com; wangkan@tsinghua.edu.cn

D. Yao
China State Key Laboratory of Reactor System Design Technology
P. O. Box 622-509, Chengdu, China, 610041
d.yao@npic.ac.cn

ABSTRACT

The Method of Characteristics (MOC) which can calculate arbitrary geometry neutron transport equation becomes more and more popular and is adopted in many neutron transport codes. However, the MOC codes with Flat Source Approximation (FSA) needs large storage space and long computing time when calculating large-scale three dimensional neutron transport problems. This paper proposes a new source approximation called Linear Source Approximation (LSA). By calculating several three dimensional neutron transport problems using the codes with FSA and LSA respectively, it was shown that the LSA MOC code gets more accurate results than the FSA MOC code with the same spatial meshes, and that the LSA MOC code with large meshes can get the results with the same accuracy of the FSA MOC code with fine spatial meshes. By using large spatial meshes which means there are less meshes and ray trajectories to store in memory, the LSA MOC code can calculate the three dimensional neutron transport problems with less storage and shorter computing time than FSA MOC code with fine meshes. Therefore, it is concluded that LSA MOC with large spatial meshes is able to calculate large-scale three dimensional problems with reasonable storage space and computing time.

Key Words: MOC, Flat Source Approximation, Linear Source Approximation, neutron transport

1. INTRODUCTION

In the past years, much interest has been devoted to the development of transport schemes that can treat arbitrary geometry and avoid homogenization process. Among these methods, the Method of Characteristics (MOC) has become more and more popular in neutron transport calculation. The past researches mainly focus on two dimension MOC and several two dimensional MOC codes[1][2][3] have been developed and used widely. It is the trend to calculate the three dimensional problems using neutron transport method instead of the diffusion method. MCCG3D[4] and Dragon[5] are typical codes which adopt the three dimensional MOC. There are two main difficulties when using three dimensional MOC to calculate large-scale problems, one of which is the required large quantity of memory, and the other is longer

[*] PhD Candidate in Tsinghua University in China

computing time needed. These difficulties are caused by large quantity of spatial meshes and ray trajectories. Our work is to reduce the required number of meshes and ray trajectories and keep the high accuracy.

This paper is organized as follows. In Section 2, a brief summary of the FSA MOC is given. The LSA MOC is described in Section 3. In Section 4, the required memory and computing time when using FSA and LSA are analyzed and compared. The numerical results are shown in Section 5.

2. MOC WITH FLAT SOURCE APPROXIMATION

The MOC method solves the transport equation along straight lines at discrete number of spatial directions. These lines, known as characteristics of the system, are regarded as neutron paths along which the differential operator of the Boltzmann equation reduces to a total derivative. The geometrical domain is sub-divided into so called flat-flux zones, where the material properties are assumed to be constant. The solid angle domain is subdivided into M discrete directions. The scattering and fission sources are assumed to be isotropic. With these assumptions, the Boltzmann transport equation can be written in the following characteristics form along the direction Ω_m :

$$\frac{d\psi_m(s)}{ds} + \Sigma_t(s)\psi_m(s) = Q(s) \tag{1}$$

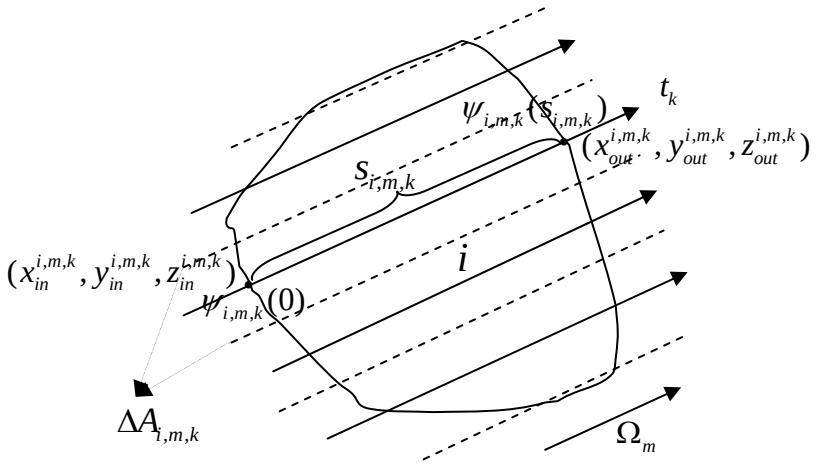

Figure 1. The MOC representation of rays

As shown in Fig.1, the angular flux $\psi_{i,m,k}(s)$ along the line segment k is then calculated by analytically integrating Eq.(1) along the characteristics, given the incoming angular flux $\psi_{i,m,k}(0)$ and the average source in zone i , Q_i

$$\psi_{i,m,k}(s) = \psi_{i,m,k}(0)e^{-\Sigma_{ti}s} + \frac{Q_i}{\Sigma_{ti}}(1-e^{-\Sigma_{ti}s}) \tag{2}$$

The outgoing angular flux $\psi_{i,m,k}(s_{i,m,k})$ can be calculated from Eq. (2). The average angular flux along the segment k is obtained by integrating Eq.(2):

$$\overline{\psi}_{i,m,k} = \frac{1}{s_{i,m,k}} \int_0^{s_{i,m,k}} \psi_{i,m,k}(s)ds = \frac{Q_i}{\Sigma_{ti}} + \frac{\Delta_{i,m,k}}{s_{i,m,k}\Sigma_{ti}} \tag{3}$$

where, $\Delta_{i,m,k} = \psi_{i,m,k}(0) - \psi_{i,m,k}(s_{i,m,k})$

The average angular flux in zone i, in direction m, $\Psi_{i,m}$, is given by:

$$\Psi_{i,m} = \frac{\sum_k \overline{\psi}_{i,m,k} \Delta A_{i,m,k} s_{i,m,k}}{\sum_k \Delta A_{i,m,k} s_{i,m,k}} \tag{4}$$

where, $\Delta A_{i,m,k}$ is the corresponding cross-sectional area in 3-D. The average scalar flux in zone i is expressed as:

$$\phi_i = \sum_m \omega_m \Psi_{i,m} \tag{5}$$

From Eq.(3)-(5), the average scalar flux in zone i is obtained as:

$$\phi_i = \frac{Q_i}{\Sigma_{ti}} + \frac{1}{\Sigma_{ti} V_i} \sum_m \omega_m \sum_k \Delta_{i,m,k} \Delta A_{i,m,k} \tag{6}$$

The numerical calculated volume $\sum_k \Delta A_{i,m,k} s_{i,m,k}$ is not equal to the real volume V_i, the track lengths are corrected as follows:

$$t_{i,m,k} = s_{i,m,k} \frac{V_i}{\sum_k s_{i,m,k} \Delta A_{i,m,k}} \tag{7}$$

The MOC assumes a known set of parallel straight lines over a number of discrete angles that cover the domain of interest along which the solution is required. Given a boundary condition and an initial neutron source distribution, the average angular flux along each segment can be determined, as well as the average angular flux in each zone i, in the direction m. The contributions to the scalar flux in each zone i, are added as soon as they are available [see Eq. (6)], but the angular fluxes are not usually saved from iteration to iteration. Each neutron path is used in both directions. When a ray hits the outer boundary, it is reflected according to the boundary condition imposed. The iterative process consists of outer ("fission source") iterations and inner ("scattering source") iterations. Convergence is checked after each outer iteration. The iterative procedure is stopped, if both the scalar flux in each zone i and the multiplication factor satisfy the convergence criteria.

3. MOC WITH LINEAR SOURCE APPROXIMATION

In FSA MOC, it is assumed that the flux and source in spatial mesh are constant. If the real physical sources vary greatly in space, small meshes must be used in FSA MOC to get the

accurate results. However, small meshes lead to the great increase of mesh number and the ray number for three dimensional problems, and thus the calculation needs large storage space and long computing time. In this section, we propose LSA MOC which brings in less error when using larger meshes to calculate the 3-D problems, so that the calculation need less storage space and computing time.

3.1. Linear Source Representation

As shown in Fig.1, we assume that the source and scalar flux in the mesh is represented:

$$Q(x,y,z) = q_a x + q_b y + q_c z + q_d \tag{8}$$

$$\phi(x,y,z) = \phi_a x + \phi_b y + \phi_c z + \phi_d \tag{9}$$

where, (x,y,z) is the global coordinate of point in the mesh. q_a, q_b, q_c, q_d and $\phi_a, \phi_b, \phi_c, \phi_d$ are the coefficients of the linear source and linear scalar flux representation.

The equation of ray trajectory t_k in direction m is:

$$\begin{aligned} x(s) &= a_m s + x_{in}^{i,m,k} \\ y(s) &= b_m s + y_{in}^{i,m,k} \\ z(s) &= c_m s + z_{in}^{i,m,k} \end{aligned} \tag{10}$$

where, (a_m, b_m, c_m) represents the direction m. s is the local coordinate of point in the mesh i. $(x_{in}^{i,m,k}, y_{in}^{i,m,k}, z_{in}^{i,m,k})$ and $(x_{out}^{i,m,k}, y_{out}^{i,m,k}, z_{out}^{i,m,k})$ are the intersection points of ray trajectory t_k and the boundary of mesh i.

Introducing Eq.(10) into Eq.(8), the source on line segment k is:

$$Q_{i,m,k}(s) = (q_a^i a_m + q_b^i b_m + q_c^i c_m)s + (q_a^i x_{in}^{i,m,k} + q_b^i y_{in}^{i,m,k} + q_c^i z_{in}^{i,m,k} + q_d^i) \tag{11}$$

3.2 Flux Calculation along Ray Trajectories

Similar with FSA MOC, the angular flux $\psi_{i,m,k}(s)$ along the line segment k is then calculated by analytically integrating Eq.(1) along the characteristics, given the incoming angular flux $\psi_{i,m,k}(0)$ and the source as shown in Eq.(11)

$$\begin{aligned} \psi_{i,m,k}(s) = [\psi_{i,m,k}(0) &- \frac{1}{\Sigma_{ti}}(q_a^i x_{in}^{i,m,k} + q_b^i y_{in}^{i,m,k} + q_c^i z_{in}^{i,m,k} + q_d^i \\ &- \frac{q_a^i a_m + q_b^i b_m + q_c^i c_m}{\Sigma_{ti}})]e^{-\Sigma_{ti}s} + \frac{q_a^i a + q_b^i b + q_c^i c}{\Sigma_{ti}} s \\ &+ \frac{1}{\Sigma_{ti}}(q_a^i x_{in}^{i,m,k} + q_b^i y_{in}^{i,m,k} + q_c^i z_{in}^{i,m,k} + q_d^i - \frac{q_a^i a_m + q_b^i b_m + q_c^i c_m}{\Sigma_{ti}}) \end{aligned} \tag{12}$$

The outgoing angular flux $\psi_{i,m,k}(s_{i,m,k})$ can be calculated from Eq.(12). The average angular flux along the segment k is obtained by integrating Eq.(12):

$$\bar{\psi}_{i,m,k} = \frac{\Delta_{i,m,k}}{s_{i,m,k}\Sigma_{ti}} + \frac{1}{\Sigma_{ti}}[\frac{1}{2}(q_a^i a_m + q_b^i b_m + q_c^i c_m)s_{i,m,k} + q_a^i x_{in}^{i,m,k} + q_b^i y_{in}^{i,m,k} + q_c^i z_{in}^{i,m,k} + q_d^i] \quad (13)$$

where $\Delta_{i,m,k} = \psi_{i,m,k}(0) - \psi_{i,m,k}(s_{i,m,k})$

The outward flux $\psi_{i,m,k}(s_{i,m,k})$ of segment k also serves as inward flux $\psi_{i,m,k'}(0)$ in the next segment k'. The fluxes along the ray trajectories can be calculated segment-by-segment by Eq.(12) .

3.3 Solving the Scalar Flux

In the FSA MOC, the scalar flux is calculated through the average equation of scalar flux after the flux calculation along the ray trajectories. However, the scalar flux in the LSA MOC need four equations to calculate the scalar flux, because there are four coefficients in scalar flux expression. We use equations of average flux, equation of flux's first order moment to plane XY, equation of flux's first order moment to plane YZ and equation of flux's first order moment to plane ZX to calculate the scalar flux.

3.3.1 Equation of average flux

The definition of scalar flux is:

$$\phi(x,y,z) = \frac{1}{4\pi}\int \psi(x,y,z,\Omega)d\Omega \quad (14)$$

Integrating the Eq.(14) over the cell volume, we get the average flux equation.

$$\frac{1}{V_i}\int \phi(x,y,z)dV = \frac{1}{4\pi}\frac{1}{V_i}\int d\Omega \int \psi(x,y,z,\Omega)dV \quad (15)$$

The volume integral of cell i in Eq.(15) is equal to the sum of volume integrals of sub-cells which are divided by line segment. Introducing Eq.(9), average flux equation becomes:

$$\sum_k \Delta A_{i,m_0,k} \int_0^{s_{i,m_0,k}} (\phi_a^i x(s) + \phi_b^i y(s) + \phi_c^i z(s) + \phi_d^i)ds = \sum_m \omega_m \sum_k \Delta A_{i,m,k} \int_0^{s_{i,m,k}} \psi_{i,m,k}(s)ds \quad (16)$$

where k is the index of ray segments which belongs to direction m (or m_0 in the left hand) and cell i. $\Delta A_{i,m,k}$ is the corresponding cross-sectional area of line segment k in three dimension. m_0 is an arbitrary direction in discrete directions. In order to make the value of left hand of Eq.(16) not vary as the selected direction m_0, we use average values of all the directions instead of one selected direction.

$$\sum_m \omega_m \sum_k \Delta A_{i,m,k} \int_0^{s_{i,m,k}} (\phi_a^i x(s) + \phi_b^i y(s) + \phi_c^i z(s) + \phi_d^i)ds = \sum_m \omega_m \sum_k \Delta A_{i,m,k} s_{i,m,k} \bar{\psi}_{i,m,k} \quad (17)$$

3.3.2 Equation of first order moment to plane YZ

Multiplying $x(s)$ and integrating over the cell volume, Eq.(14) becomes to:

$$\int \phi(x,y,z)xdV = \int xdV \frac{1}{4\pi}\int \psi(x,y,z,\Omega)d\Omega \tag{18}$$

Eq.(18) can be simplified as:

$$\sum_m \omega_m \sum_k \Delta A_{i,m,k} \int_0^{s_{i,m,k}} (\phi_a^i x(s) + \phi_b^i y(s) + \phi_c^i z(s) + \phi_d^i)x(s)ds = \sum_m \omega_m \sum_k \Delta A_{i,m,k} \int_0^{s_{i,m,k}} x(s)\psi_{i,m,k}(s)ds \tag{19}$$

3.3.3 Equation of first order moment to plane XZ and XY

Similar with Eq.(19), the equations of first order moment to plane XZ and XY are:

$$\sum_m \omega_m \sum_k \Delta A_{i,m,k} \int_0^{s_{i,m,k}} (\phi_a^i x(s) + \phi_b^i y(s) + \phi_c^i z(s) + \phi_d^i)y(s)ds = \sum_m \omega_m \sum_k \Delta A_{i,m,k} \int_0^{s_{i,m,k}} y(s)\psi_{i,m,k}(s)ds \tag{20}$$

$$\sum_m \omega_m \sum_k \Delta A_{i,m,k} \int_0^{s_{i,m,k}} (\phi_a^i x(s) + \phi_b^i y(s) + \phi_c^i z(s) + \phi_d^i)z(s)ds = \sum_m \omega_m \sum_k \Delta A_{i,m,k} \int_0^{s_{i,m,k}} z(s)\psi_{i,m,k}(s)ds \tag{21}$$

3.3.4 Summary

According to Eq.(17) and Eq.(19)-(21), four coefficients of scalar flux can be calculated by:

$$\vec{\Phi}_i = \frac{1}{\Sigma_{ti}}[B_i^{-1}(M_i\vec{Q}_i + \vec{P}_i) + \vec{Q}_i] \tag{22}$$

where $\vec{\Phi}_i = (\phi_d^i, \phi_a^i, \phi_b^i, \phi_c^i)$ is the vector expression of scalar flux in LSA. $\vec{Q}_i = (q_d^i, q_a^i, q_b^i, q_c^i)$ is the vector expression of total source in LSA. B_i and M_i are coefficient matrixes which can be calculated by geometry and material information. $\vec{P}_i = (p_d^i, p_a^i, p_b^i, p_c^i)$ and:

$$\begin{aligned}
p_d^i &= \sum_m \omega_m \sum_k \Delta A_{i,m,k}\Delta_{i,m,k} \\
p_a^i &= \sum_m \omega_m \sum_k \Delta A_{i,m,k}(x_{in}^{i,m,k}\Delta_{i,m,k} - a_m\psi_{in}^{i,m,k}e^{-\Sigma_{ti}s_{i,m,k}}s_{i,m,k} + \frac{a_m}{\Sigma_{ti}}\Delta_{i,m,k}) \\
p_b^i &= \sum_m \omega_m \sum_k \Delta A_{i,m,k}(y_{in}^{i,m,k}\Delta_{i,m,k} - b_m\psi_{in}^{i,m,k}e^{-\Sigma_{ti}s_{i,m,k}}s_{i,m,k} + \frac{b_m}{\Sigma_{ti}}\Delta_{i,m,k}) \\
p_c^i &= \sum_m \omega_m \sum_k \Delta A_{i,m,k}(z_{in}^{i,m,k}\Delta_{i,m,k} - c_m\psi_{in}^{i,m,k}e^{-\Sigma_{ti}s_{i,m,k}}s_{i,m,k} + \frac{c_m}{\Sigma_{ti}}\Delta_{i,m,k})
\end{aligned} \tag{23}$$

The calculation process of LSA MOC is similar with that of FSA MOC. Given a boundary condition and an initial neutron source distribution, the angular flux along each segment can be determined. The contributions to the vector $\vec{P}$ in each zone i, are added as soon as they are available [see Eq. (23)]. Each neutron path is used in both directions. Once the vector $\vec{P}$ in each zone i is calculated, the scalar flux can be determined by Eq.(22). The iteration process is as the same as FSA MOC.

4. ESTIMATION OF CPU TIME AND STORAGE SPACE

The purpose of adopting LSA in MOC is to reduce the storage space and computing time, because the LSA MOC can use larger meshes than FSA MOC when their results reach the same accuracy. But the LSA MOC has a disadvantage which is that the LSA MOC needs more computing time and storage than the FSA MOC when they use the same meshes. In the following analysis, we assume that the LSA MOC which uses n times larger meshes can reach the same accuracy as the FSA MOC with the fine mesh.

We now analyze the memory used by FSA MOC and LSA MOC. If the mesh size becomes n times larger in 3-D, the number of meshes is $1/n^3$ of original number. Because there are four coefficients in flux and source expression in LSA MOC, the mesh information that LSA MOC needs store is $4/n^3$ of that which FSA MOC needs, when LSA MOC use n times larger mesh than FSA MOC. Assuming the radio of rays' distance to mesh size is constant, the number of rays in LSA MOC is $1/n^2$ of that in FSA MOC, and the number of ray segments in LSA MOC is $1/n^3$ of that in FSA MOC. In the MOC calculation process, we just need to store the meshes and ray segments information, so the total memory used by LSA MOC is $O(1/n^3)$ of that used by FSA when LSA MOC using n times larger mesh.

Assuming the number of ray segments is K and the number of meshes is N, calculations along all the ray segments need $6K$ multiplication in FSA MOC and $42K$ multiplication in LSA MOC. To calculate the scalar fluxes, it needs $2N$ multiplication in FSA MOC and $33N$ multiplication in LSA MOC. If the numbers of iterations in FSA MOC and LSA MOC calculation processes are equal, LSA MOC will spend $\frac{M+42K+33N}{M+6K+2N}$ times more computing time than FSA MOC. We assume M, which is the other multiplication calculation except the calculation along the ray segments and the calculation of scalar fluxes, to be equal in FSA and LSA MOC. Finally, when the LSA MOC uses n times larger meshes, the computing time that LSA MOC spends is $\frac{M+42K/n^3+33N/n^3}{M+6K+2N}$ of that which FSA MOC spends.

5. NUMERICAL TESTS

Based on FSA and LSA MOC, we develop the three dimensional FSA and LSA MOC codes. TCM (Three dimension Characteristics Method code) is the FSA MOC code and TCM_L is the LSA MOC code. The following problems are used to test the two codes and compare the two source approximations.

5.1 BWR Problem

This problem is a two dimension homogeneous BWR cell[6]. The central homogenized fuel region is surrounded by water moderator as presented in Fig. 2. The macroscopic cross-sections of each region are given in Table I. We assume the height of the problem is 0.8cm. We use two kinds of meshes shown in Fig. 3 and Fig. 4 to compare the accuracy and efficiency of two codes. The results are compared with MCNP and SURCU[6]. Table II shows the comparison of

consumed storage and computing time and Table III shows the comparison of K-infinite and neutron flux distribution.

The results show that the LSA MOC with coarse mesh needs about 1/5 storage space and computing time compared with FSA MOC with fine mesh, and their results have similar accuracy.

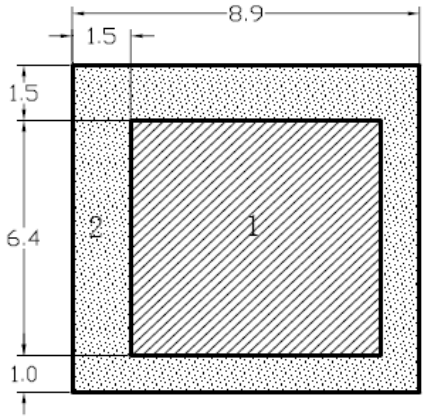

Figure 2. BWR problem geometry /cm

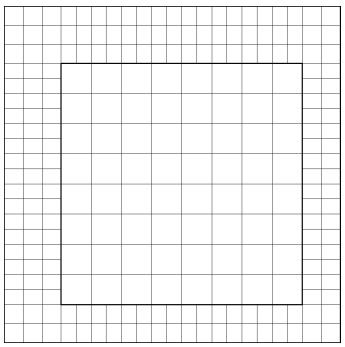

Figure 3. Fine mesh

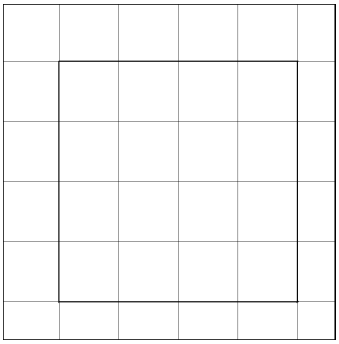

Figure 4. Coarse mesh

Table I. Macroscopic cross-sections of BWR problem

Energy group	Material	$\nu\Sigma_f^g / cm^{-1}$	Σ_s^{g-1} / cm^{-1}	Σ_s^{g-2} / cm^{-1}	Σ_t^g / cm^{-1}	χ_g
1	**1**	6.203E-3	1.780E-1	1.002E-2	1.96647E-1	1.0
	2	0.0	1.995E-1	2.188E-2	2.22064E-1	
2	**1**	1.101E-1	1.089E-3	5.255E-1	5.96159E-1	0.0
	2	0.0	1.558E-3	8.783E-1	8.87874E-1	

Table II. Storage and CPU time

	Fine mesh, S16		**Coarse mesh, S16**	
	TCM	**TCM_L**	**TCM**	**TCM_L**
Mesh number	249		36	
Ray separation (cm)	0.08		0.2	
Ray number	1186692		272762	
Track number	5930312		943412	
Outer iteration	28	28	28	28
Inner iteration	450	450	448	450

Memory (Mb)	367.75	382.98	78.24	82.38
CPU Time (s)	3060.30	4173.67	463.86	644.81

Table III. K-infinite and neutron flux distribution of BWR problem

		Fast group		Thermal group		K-infinite
		Fuel region	**Moderator region**	**Fuel region**	**Moderator region**	
MCNP		1.0	0.9249	0.3520	0.4515	1.2122 0.0001*
SURCU		1.0 (0.00%**)	0.9271 (0.24%)	0.3529 (0.25%)	0.4509 (-0.14%)	1.2127 (0.04%)
Fine mesh	**TCM**	1.0 (0.00%)	0.9264 (0.17%)	0.3534 (0.38%)	0.4484 (-0.69%)	1.2138 (0.08%)
	TCM_L	1.0 (0.00%)	0.9262 (0.14%)	0.3528 (0.21%)	0.4523 (0.16%)	1.2143 (0.06%)
Coarse mesh	**TCM**	1.0 (0.00%)	0.9274 (0.28%)	0.3550 (0.85%)	0.4390 (-2.79%)	1.2139 (0.14%)
	TCM_L	1.0 (0.00%)	0.9263 (0.16%)	0.3531 (0.32%)	0.4515 (0.003%)	1.2131 (0.07%)

* relative standard deviation； **percent error

5.2 PWR Assembly Problem

This problem is the UO2 lattice of C5G7 benchmark[7], as shown in Fig.5 and Fig.6. 7 groups macroscopic cross-sections are used in this problem. The reference results are given by MCNP code and DeCART[8](only K-Infinite provided). We calculate this problem with two kinds of meshes shown in Fig. 7 and Fig. 8.

K-Infinite and pin powers are compared in Table IV, V and VI. In Table V, AVG RMS and MRE are used to measure the pin power distribution. They are defined as $AVG = \frac{\sum_N |e_n|}{N}$, $RMS = \frac{\sqrt{\sum_N e_n^2}}{N}$ and $MRE = \frac{\sum_N |e_n| \bullet p_n}{N \bullet p_{avg}}$. N is the number of fuel pin, p_n is the pin power and e_n is the percent error.

Table VII is the comparison of storage and CPU time. From the results, we can see that the TCM_L can get better results than TCM when they use the same mesh and the TCM_L using

coarse mesh can get almost the same accurate results as TCM using fine mesh. Compared with TCM using fine mesh, TCM_L with coarse mesh can save almost 4/5 storage and CPU time.

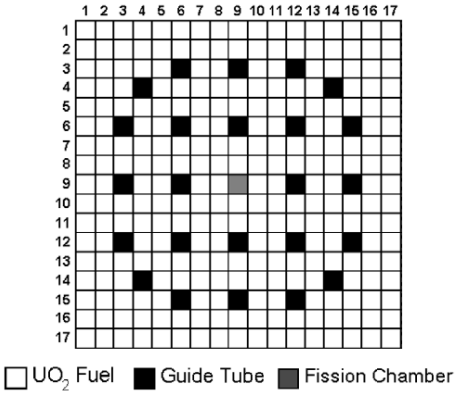

Figure 5. C5G7 UO2 assembly geometry

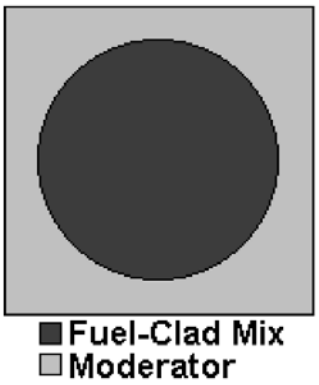

Figure 6. Cell geometry in lattice

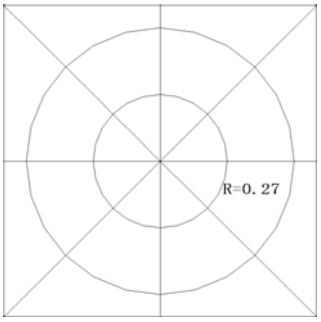

Figure 7. Fine mesh in each cell

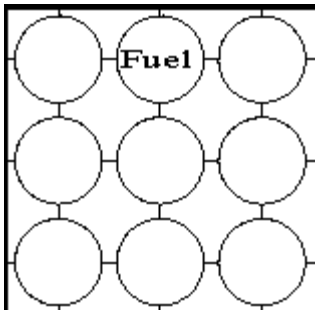

Figure 8. Coarse mesh for assembly

Table IV. K-Infinite results

	MCNP	DeCART	Fine mesh		Coarse mesh	
			TCM	TCM_L	TCM	TCM_L
K-Infinite	1.333419 0.00004*	1.33321	1.333616	1.332483	1.33388	1.332923
Relative Error	——	-0.016%	0.015%	-0.070%	0.035%	-0.037%

* relative standard deviation

Table VI. Results for pin powers

		Maximum pin power	Percent error	Minimum pin power	Percent error	Maximum percent error
MCNP		1.0776		0.9112		
Fine mesh	**TCM**	1.0763	-0.117%	0.9113	0.005%	0.291%
	TCM_L	1.0774	0.046%	0.9118	0.042%	0.140%
Coarse mesh	**TCM**	1.0692	-0.706%	0.9131	0.202%	1.268%
	TCM_L	1.0771	0.007%	0.9111	-0.070%	0.261%

Table V. Pin power distribution error measures

		AVG	RMS	MRE
Fine mesh	**TCM**	0.127%	0.009%	0.130%
	TCM_L	0.039%	0.003%	0.039%
Coarse mesh	**TCM**	0.698%	0.045%	0.710%
	TCM_L	0.080%	0.006%	0.079%

Table VII. Storage and CPU time

	Fine mesh, S8		Coarse mesh, S8	
	TCM	**TCM_L**	**TCM**	**TCM_L**
Mesh number	6936		613	
Ray separation (cm)	0.06		0.15	
Ray number	2,734,635		535,406	
Track number	25,002,282		3,907,590	
Out iteration	20	20	20	20
In iteration	312	306	312	310
Memory (Mb)	1,107.11	1,148.27	201.76	209.59
CPU Time (s)	17379.66	25267.31	2789.89	3715.09

5.3 KUCA Benchmark

This benchmark is one of set of 3-D neutron transport benchmark problems proposed by the Osaka University to NEACRP in 1991[9], which has been calculated by many participants. The core is a model of the Kyoto University Critical Assembly (KUCA). The core calculation is performed in 2 groups. Two kinds of meshes are used. Fine mesh size is 1cm*1cm*1cm and coarse mesh size is 2cm*2cm*2cm. We only compare the storage space and computing time for case 1 (the control rod position is empty).

The K-effective and rod worth comparisons are shown in Table VIII, and region average fluxes comparisons are shown in Table IX. Table X shows the comparison of storage and computing time. From Table VIII and Table IX, we can know that with the same mesh the LSA MOC method code TCM_L gets more accurate results than FSA MOC method code TCM. The TCM_L using coarse mesh spends less CPU time and memory than TCM using fine mesh, and the results of TCM_L with coarse mesh are more accurate.

Table VIII. K-effective and rod worth of KUCA

	Control rod out case	**Control rod in case**	**Control rod worth**
Monte Carlo	0.9778	0.9624	1.64E-02
	0.05%*	0.05%	0.07%
P_N	0.9766	0.9630	1.45E-02
TCM (Fine mesh)	0.9766 （0.12%**）	0.9609 （0.16%）	1.68E-02 （2.31%）
TCM_L (Fine mesh)	0.9774 (0.07%)	0.9625 (0.01%)	1.577E-02 (4.84%)
TCM (Coarse mesh)	0.9547 (2.39%)	0.9394 (2.39%)	1.701E-02 (2.64%)
TCM_L (Coarse mesh)	0.9775 (0.05%)	0.9625 (0.01%)	1.593E-02 (3.86%)

* relative standard deviation； **percent error

Table IX. Region-averaged flux of KUCA

	Energy Group	**Core**	**Reflector**	**Void**
Monte Carlo	1G	4.7509E-03	5.9251E-04	1.4500E-03
		0.10%*	0.21%	0.47%
	2G	8.6998E-04	9.1404E-04	9.7406E-04
		0.12%	0.23%	0.63%
P_N	1G	4.7472E-03	5.9439E-04	1.4096E-03
	2G	8.6452E-04	9.2059E-04	9.0113E-04
TCM (Fine mesh)	1G	4.7416E-03 (0.20%**)	5.9733E-04 (0.81%)	1.4525E-03 (0.17%)
	2G	8.7248E-04 (0.29%)	8.3091E-04 (9.09%)	9.1679E-04 (5.88%)

TCM _L (Fine mesh)	1G	4.7556E-03 (0.10%)	5.9467E-04 (0.36%)	1.4588E-03 (0.61%)
	2G	8.7204E-04 (0.24%)	9.1431E-04 (0.03%)	9.7169E-04 (0.24%)
TCM (Coarse mesh)	1G	4.8283E-03 (1.63%)	6.0873E-04 (2.74%)	1.4802E-03 (2.08%)
	2G	8.6977E-04 (0.02%)	7.8943E-04 (13.63%)	8.5464E-04 (12.26%)
TCM_L (Coarse mesh)	1G	4.7520E-03 (0.02%)	5.9487E-04 (0.40%)	1.4583E-03 (0.57%)
	2G	8.7216E-04 (0.25%)	9.1098E-04 (0.33%)	9.7048E-04 (0.37%)

* relative standard deviation； **percent error

Table X. Storage space and CPU time comparison

	Fine mesh, S8		**Coarse mesh, S8**	
	TCM	**TCM_L**	**TCM**	**TCM_L**
Mesh number	15625		2469	
Ray separation (cm)	0.1933		0.3866	
Ray number	1,034,268		257,436	
Track number	25,862,030		3,477,492	
Out iteration	36	41	33	41
In iteration	320	363	290	362
Memory (Mb)	547.85	576.76	90.70	97.09
CPU Time (s)	1853.74	6638.47	255.28	771.94

6. CONCLUSIONS

In this paper, we introduce the linear source approximation MOC. According to the analysis and numerical tests, it was shown that the LSA MOC can get more accurate results than FSA MOC with the same mesh size. By using larger meshes, the LSA MOC needs less memory and CPU time than FSA MOC when calculating three dimension problems. It's concluded that the LSA MOC which consumes less memory is suitable to calculate large three dimensional problems.

In future, some acceleration methods should be adopted in linear source approximation MOC in order to increase the calculation speed of LSA MOC. Anisotropic scattering also should be adopted in LSA MOC.

ACKNOWLEDGMENTS

This work is supported by China State Key Laboratory of Reactor System Design Technology.

REFERENCES

1. M. J. Halsall, "CACTUS, A Characteristics Solution to the Neutron Transport Equations in Complicated Geometries," *AEEW-R* 1291 (1988)
2. D. Knott, "KRAM: A Lattice Physics Code for Modeling the BWR Fuel Designs," *Doctoral Thesis*, Pennsylvania State University (1990).
3. T. Jevremovic, J. Vujic, and K. Tsuda, "ANEMONA - a neutron transport code for general geometry reactor assemblies based on the method of characteristics and R-function solid modeler," *Annals of Nuclear Energy* **Vol. 28** pp.125-152 (2001)
4. IGOR SUSLOV, "Improvement in the Long Characteristics Method and Their Efficiency for Deep Penetration Calculations," *Progress in Nuclear Energy*, **Vol. 39, No.2**, pp.233-243, (2001)
5. G.J. Wu and R.Roy, "A new characteristics algorithm for 3D transport calculations", *Annals of Nuclear Energy* **Vol. 30** pp.1-16 (2003)
6. J. Stepanek, T. Auerbach, W. Haelg, "Calculation of Four Thermal Reactor Benchmark Problems in X-Y Geometry," *EPRI NP-2855* (1983)
7. "Benchmark on Deterministic Transport Calculations Without Spatial Homogenisation - A 2-D/3-D MOX Fuel Assembly Benchmark," *Nuclear Science NEA/NSC/DOC(2003)16, OECD*, (2003)
8. Smith, M. A., Palmiotti, G., Taiwo, T. A., Lewis, E. E., Tsoulfanidis, N., "Benchmark Specification for Deterministic MOX Fuel Assembly Transport Calculations without Spatial Homogenisation (3-D extension C5G7 MOX)" , *NEA/NSC/DOC(2001)4*, (2001)
9. Takeda T and Ikeda H, "3-D Neutron Transport Benchmarks[R]," *Technical Report OECD/NEA Committee on Reactor Physics (NEACRP)*, (1991)