

基于高效特征线追踪和实时生成的 直接三维特征线计算方法

俞荣君, 杨睿, 胡贇, 赵守智*

(中国原子能科学研究院, 北京 102413)

摘要: 直接三维特征线方法(3D MOC)具有强大的几何处理能力, 但是其对计算机的性能要求苛刻, 尤其是极大的存储规模会限制机器适用性。为此, 本文提出了一种基于层次建模树和改进R函数方法的特征线追踪方法, 并在此基础上采用特征线信息实时生成(on-the-fly generation)技术, 可有效降低计算时间和存储空间。对Takeda和C5G7算例的数值结果表明, 在特征值等问题计算上与参考解相比具有良好的计算精度, 提出的高效特征线追踪和实时生成方法能够显著降低存储空间, 在分别付出增加约45%、23%的时间成本后, 两个算例的存储空间分别最大可降低原来的99.51%、99.92%, 证明了该方法可大大提高直接3D MOC的机器适用性。

关键词: 直接三维特征线方法; 特征线追踪; 实时生成

中图分类号: TL329

文献标志码: A

文章编号: 1000-6931(2025)01-0118-09

doi: 10.7538/yzk.2024.youxian.0115

Direct 3D Method of Characteristic Based on Efficient Characteristic Tracing and On-the-fly Generation

YU Rongjun, YANG Rui, HU Yun, ZHAO Shouzhi*

(China Institute of Atomic Energy, Beijing 102413, China)

Abstract: High-precision full reactor simulation is extremely important for reactor design and safety analysis. In recent years, three-dimensional (3D) full reactor transport calculation becomes one of the key development directions of deterministic calculation in reactor physics. 3D full reactor transport calculation requires the calculation program to have strong anisotropy, fine geometric processing and large-scale parallel computing capabilities. The method of characteristics (MOC) is one of the excellent implementations of the next generation of neutronic deterministic calculations due to its powerful geometric processing capabilities and natural parallelism. However, the huge requirements of the MOC for computing time and space storage are also important factors restricting development. The main technical routes of MOC include 2D/1D method and direct 3D method. Among them, the direct 3D MOC has powerful geometric processing capabilities, but its performance requirements for computers are harsh, especially the large storage scale will limit the applicability of the machine. To this end, a

收稿日期: 2024-03-12; 修回日期: 2024-06-10

基金项目: 中核集团“青年英才”启明星项目

* 通信作者: 赵守智

characteristic tracking method based on hierarchical modeling tree and improved R-function method was proposed, and on this basis, on-the-fly generation of characteristic information technology was adopted, which could effectively reduce computation time and spatial storage. The numerical results of the Takeda and C5G7 benchmarks show that the method have good computational accuracy compared to the reference solution in the calculation of eigenvalues and other benchmarks. The proposed efficient characteristic tracking and on-the-fly generation technology can significantly reduce spatial storage. After increasing the time cost by about 45% and 23% respectively, the storage of the two benchmarks can be reduced by up to 99.51% and 99.92%, respectively. It is proved that this method can greatly improve the machine applicability of the direct 3D MOC.

Key words: direct three-dimensional method of characteristic; characteristic tracing; on-the-fly generation

近年来, 三维(3D)全堆输运计算成为反应堆物理中确定论计算的重点发展方向之一。三维全堆输运计算要求计算程序具备强各向异性、精细几何处理以及大规模并行计算能力。特征线方法(MOC)以其强大的几何处理能力和天然的可并行性成为下一代中子学确定论计算的优秀实现方案之一。

MOC的主要技术路线包括2D/1D方法和直接3D方法。2002年, CRX程序^[1-2]首次采用二维MOC(2D MOC)和1D S_N 方法相耦合(2D/1D fusion)的方式, 将整个堆芯轴向分为若干层, 对每一层进行积分可以得到源项中带有轴向泄漏项的2D MOC方程, 对每个径向网格进行轴向积分可以得到源项中带有径向泄漏项的1D S_N 方程, 两个方程轮流计算, 相互更新泄漏项。TIGER^[3]、MPACT^[4]、DeCART^[5-6]、nTRACER^[7]、NECP-X^[8]等程序均实现了2D/1D方法。2011年, Liu等^[9]开发了三维MOC(3D MOC)程序, 提出了3D模块化MOC, 该方法可以极大降低3D MOC程序的存储空间要求, 但也限制了几何适用性。后续MPACT3D^[10]、APOLLO3^[7, 11-13]、OpenMOC^[14]等程序也增加了3D MOC模块。

2D/1D方法可以提升计算速度, 但只能处理柱状体几何, 也存在不稳定性问题。直接3D MOC具备和蒙特卡罗方法相当的几何处理能力, 是一种理想的确定论解决方案。但是3D MOC程序计算量很大, 对存储空间要求也很大。

计算量大的问题可以通过粗网有限差分(CMFD)等加速方法进行解决, 但是存储空间要求会极大限制3D MOC程序在一般机器上的部署。存储空间主要来自于特征线信息的存储, 因

此采用特征线信息实时生成(on-the-fly generation)可以大大降低存储要求。但是OpenMOC^[15]等程序所采用的逐段求解特征线信息的方法效率比较低, 采用实时生成付出的时间成本较大。

本文提出一种新的基于层次建模树和改进R函数方法的特征线追踪方法, 可以将原来分别求解的特征线线段一次性完成计算, 使实时生成所占的时间成本远小于总计算时间成本, 可大大提高3D MOC程序的机器适用性。

1 特征线方法

稳态中子多群输运方程可以简写为:

$$\boldsymbol{\Omega} \cdot \nabla \phi_g(\boldsymbol{\Omega}, \mathbf{r}) + \Sigma_{t,g}(\mathbf{r})\phi_g(\boldsymbol{\Omega}, \mathbf{r}) = Q_g(\boldsymbol{\Omega}, \mathbf{r}) \quad (1)$$

式中: $\phi_g(\boldsymbol{\Omega}, \mathbf{r})$ 为角通量; $\Sigma_{t,g}(\mathbf{r})$ 为总截面; $Q_g(\boldsymbol{\Omega}, \mathbf{r})$ 为总角度源; g 为能群编号; $\boldsymbol{\Omega}$ 为立体角; \mathbf{r} 为空间位置。

边界条件可以写为:

$$\phi_g(\mathbf{r}_0, \boldsymbol{\Omega}) = \alpha(\mathbf{r}'_{\text{out}})\phi_g(\mathbf{r}'_{\text{out}}, \boldsymbol{\Omega}') \quad (2)$$

式中: $\mathbf{r}_0, \boldsymbol{\Omega}$ 为入射位置和角度; \mathbf{r}'_{out} 和 $\boldsymbol{\Omega}'$ 为出射位置和角度; α 为反照率, $0 \leq \alpha \leq 1$ 。

取 $\mathbf{r} = \mathbf{r}_0 + s\boldsymbol{\Omega}$, 利用常数变易法, 可以解得:

$$\phi_g(\mathbf{r}_0 + \boldsymbol{\Omega}s, \boldsymbol{\Omega}) = \int_{s'=0}^s Q_g(\mathbf{r}_0 + \boldsymbol{\Omega}s', \boldsymbol{\Omega}) e^{\int_{s'=s}^s -\Sigma_{t,g}(\mathbf{r}_0 + \boldsymbol{\Omega}s'') ds''} ds' + \phi_g(\mathbf{r}_0, \boldsymbol{\Omega}) e^{\int_{s'=0}^s -\Sigma_{t,g}(\mathbf{r}_0 + \boldsymbol{\Omega}s') ds'} \quad (3)$$

式中, s 为沿 $\boldsymbol{\Omega}$ 方向移动的距离。

对于某个离散角度 $\boldsymbol{\Omega}_m$, 式(3)可简写为:

$$\phi_{m,g}(s) = \int_{s'=0}^s Q_{m,g}(s') e^{\int_{s'=s}^s -\Sigma_{t,g}(s'') ds''} ds' + \phi_{m,g}(\mathbf{r}_0) e^{\int_{s'=0}^s -\Sigma_{t,g}(s') ds'} \quad (4)$$

式中: $Q_{m,g}$ 为能群 g 在离散角度 $\boldsymbol{\Omega}_m$ 上的角度源; $\phi_{m,g}$ 为能群 g 在离散角度 $\boldsymbol{\Omega}_m$ 上的中子通量。

根据平源近似假设,在每个平源近似区 F 中,式(4)变为:

$$\phi_{m,g}^F(s) = \frac{Q_{m,g}^F}{\Sigma_{t,g}^F} (1 - e^{-\Sigma_{t,g}^F s}) + \phi_{m,g}^F(\mathbf{r}_0) e^{-\Sigma_{t,g}^F s} \quad (5)$$

该线上的平均通量可写为:

$$\bar{\phi}_{m,g}^F = \frac{Q_{m,g}^F}{\Sigma_{t,g}^F} - \left(\frac{Q_{m,g}^F}{\Sigma_{t,g}^F} - \phi_{m,g}^F(\mathbf{r}_0) \right) \frac{1 - e^{-\Sigma_{t,g}^F s}}{\Sigma_{t,g}^F s} = \frac{Q_{m,g}^F - \Delta\phi_{m,g}^F/s}{\Sigma_{t,g}^F} \quad (6)$$

式中, $\Delta\phi_{m,g}^F$ 为出通量减去进通量,可表示为:

$$\Delta\phi_{m,g}^F = \phi_{m,g}^F(s) - \phi_{m,g}^F(\mathbf{r}_0) = \left[\frac{Q_{m,g}^F}{\Sigma_{t,g}^F} - \phi_{m,g}^F(\mathbf{r}_0) \right] (1 - e^{-\Sigma_{t,g}^F s}) \quad (7)$$

取求积组为角度 $\{\Omega_m\}$ 和求积系数 $\{\omega_m\}$,则平源区的通量可表示为:

$$\phi_{g',p,l}^{F,0} = \frac{1}{V_F} \sum_M \omega_m Y_{p,l}^*(\Omega_m) \iint_{S_m^F} I_m^F(\sigma) \bar{\phi}_{m,g'}^F(\sigma) d\sigma \quad (8)$$

式中: $\phi_{g',p,l}^{F,0}$ 为平源区 F 能群 g' 的通量; V_F 为平源区 F 的体积; M 为求积组的个数; $Y_{p,l}^*(\Omega_m)$ 为球谐函数; S_m^F 为平源区 F 在 Ω_m 方向的法平面上的投影; I_m^F 为从 σ 出发沿方向 Ω_m 在平源区 F 中穿行的长度; $\bar{\phi}_{m,g'}^F(\sigma)$ 为平源区 F 上 σ 的 g' 群平均通量; σ 为法平面上的 1 个小面积。

边界条件处理方面,本文采用文献 [16] 中提出的将角通量场拆成源场和边界场分离计算的方法。其源场中的特征线和其边界追踪的特征线可以实现首尾相连,由于一点接一点,不需要进行插值方法的复杂处理,也不需要引入边界误差;同时由于边界场天然有限,不要求和循环特征线一样返回原点,解除了几何限制。该方法的优点在于可以兼顾循环特征线方法的首尾相连的特性,也可以像插值、打混方法一样适用于精细几何,并且不需要存储边界角通量和进行边界迭代,理论上可以处理任意几何并用于实现特征线并行计算。

2 特征线建模和高效追踪方法

为提高特征线追踪效率,本文提出了基于层次建模树和改进 R 函数方法的特征线追踪方法,该方法可以将 1 条特征线上的所有线段一次性求出,并且不需要反复定位和边界求解。

2.1 几何模型描述

设空间范围为 $U \subset \mathbb{R}^3$, 材料可以通过标号记

为 1 个有限正整数集合 materials, 则几何模型 G 可以描述为空间范围和其上的材料映射:

$$G = (U, f: U \rightarrow \text{materials}) \quad (9)$$

最简单的几何模型是所有空间范围只有一种材料,记为:

$$G = (U, f: U \rightarrow \{m\}) = (U, m) \quad (10)$$

称这种模型为图元模型,任何复杂的堆芯模型都必须由图元模型组合而成。图元模型一般通过充实空间(Universe)和层次建模方法组成更复杂模型。假设上层模型 G_{up} 的空间范围为 U_{up} , 它由若干下层模型 $\{G_{i,down}\}$ 的并集构成,可表示为:

$$G_{up} = U_{up} \cap \left(\bigcup_{i=1}^I G_{i,down} \right) \quad (11)$$

所谓充实空间,就是要求在 U_{up} 中, $G_{i,down}$ 对应空间范围 $U_{i,down}$ 之间不能存在空隙或者冲突,即:

$$\begin{cases} \forall i, j, U_{i,down} \cap U_{j,down} = \emptyset \\ U_{up} \subset G_{up} = \bigcup_{i=1}^n U_{i,down} \end{cases} \quad (12)$$

2.2 空间范围描述

在反应堆物理中,通常采用构造实体几何(CSG)方法描述空间范围 U ,也就是通过若干基本空间范围及其交、并和补运算进行描述。

基本空间范围由多项式的不等式描述:

$$U = \{\mathbf{r} | f(\mathbf{r}) > 0\} \quad (13)$$

对于空间范围 U , 交并运算与集合论一致,但补运算 $\neg U$ 定义略有差别:

$$\neg U = \delta_{R^3}(\text{cl}(U)) \quad (14)$$

式中: δ_{R^3} 表示补集运算; $\text{cl}(U)$ 为闭包运算。闭包运算是为了保证 $\neg U$ 仍然是开区间,如 $x < 1$ 的闭包是 $x \leq 1$, 再取补集 $x > 1$ 。

CSG 方法描述简单准确,函数计算结果为正表示体内,为负表示体外。但问题在于如果计算结果为 0, 并不能够直接认为是边界点。如 $(x > 0) \cup (x < 0)$ 中的 $x = 0$ 就是聚点, $(x > 0) \cap (x < 0)$ 中的 $x = 0$ 就是孤立点。显然,这种聚点也应当被认为是空间范围内的点,而孤立点应当被认为是空间范围外的点。

严格空间范围内外的定义为:若点 \mathbf{r} , 存在 ε , 使得去心邻域 $\overset{\circ}{U}(\mathbf{r}_0, \varepsilon) \subset \delta_{R^3}(\neg U)$ 或 $\overset{\circ}{U}(\mathbf{r}_0, \varepsilon) \subset \delta_{R^3}(U)$, 则认为 \mathbf{r} 在空间范围内或外。

2.3 特征线在模型中的追踪

特征线追踪就是求解特征线线段与模型 G 的交集结果:

$$s \cap G = \bigcup (s_i, m_i), s = \{r | r = \Omega t + r_0, t \in [0, l]\} \quad (15)$$

式中: Ω 为特征线方向; r_0 为特征线线段起点; l 为特征线长度; (s_i, m_i) 表示第 i 段线段的信息及对应的材料 m_i 。对于最顶层几何, 可认为起始点在负无穷处, 长度为无穷大, 即 1 条直线。

对于图元模型(式(10)), 有:

$$s \cap G = (s \cap U, m) \quad (16)$$

对于组合模型(式(11)), 有:

$$s \cap G_{up} = \bigcup_I [(s \cap U_{up}) \cap G_{i,down}] \quad (17)$$

也就是交集后的线段在各个子模型中追踪结果的并集。

充实空间(式(12))的限定可写为:

$$\begin{cases} \forall i, j, s_{i,down} \cap s_{j,down} = \emptyset \\ s \cap U_{up} = \bigcup_{i=1}^n s_{i,down} \end{cases} \quad (18)$$

式中, $s_{i,down}$ 为线段在子模型中追踪的结果, 也就是要求所有子线段必须首尾紧密相连, 并构成与 $s \cap U_{up}$ 一样的完整线段。

2.4 特征线与空间范围的交集

为了解决 CSG 方法对边界点描述不准确的问题, 同时考虑到特征线追踪中还会出现的切点问题, 引入 R 函数:

$$R_U(r) = \begin{cases} 1 & r \in U \\ -1 & r \in \neg U \\ 0 & r \in \Gamma(U) \end{cases} \quad (19)$$

式中, Γ 为空间 U 的边界。

如前所述 $R_U(r) = 0$ 的点并不一定是边界点, 也可能是聚点或孤立点。为此定义新的 r_U 函数:

$$r_U(t) = \lim_{\Delta t \rightarrow 0} \frac{R_U(t - \Delta t) + R_U(t + \Delta t)}{2} \quad (20)$$

式中: t 为自变量; Δt 为自变量增量。

可以证明 r_U 函数的所有间断点都为零点, 间断点两端必然异号, 图像为锯齿状, 如图 1 所示。

r_U 函数保证了: 聚点处值为 1, 即被视为内点; 孤立点处值为 0, 即被视为外点; 边界点处值为 0, 即被视为边界点; 切点处值与两端同号。因此, r_U 函数可以准确表示点与空间范围的关系。

考虑到 r_U 函数的锯齿形状和有限个零点 $\{t_i | i = 1, \dots, N\}$, 记为:

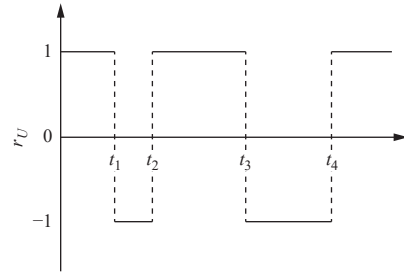


图 1 r_U 函数图像

Fig. 1 r_U function image

$$\begin{cases} t_0 = 0 \\ t_{N+1} = l \\ \delta = \lim_{\Delta t \rightarrow 0} r_U(\Delta t) \end{cases} \quad (21)$$

式中: t_0 为线段起点; t_{N+1} 为线段终点; l 为线段长度; δ 表示 t_1 点之前是否在体内。

可将 $r_U(t)$ 简记为:

$$\begin{aligned} r_U(t) &= r(\delta, \{t_i\}) \\ r(\delta, \{t_i\}) &= \begin{cases} 0 & t \in \{t_i\} \\ \delta \times (-1)^{i-1} & t \in [t_{i-1}, t_i] \end{cases} \end{aligned} \quad (22)$$

特征线线段 s 与基本空间范围(式(13))的交集为一元四次多项式的求解, 可以采用费拉里方法完成。

假设线段 s 与空间范围 U_1 的交集为 $r(\delta_1, \{t_{1,i}\})$, 与空间范围 U_2 的交集为 $r(\delta_2, \{t_{2,i}\})$, 对补运算有:

$$s \cap (\neg U_1) = r(-\delta_1, \{t_{1,i}\}) \quad (23)$$

对并运算根据定义有:

$$\begin{aligned} s \cap (U_1 \cup U_2) &= \\ &= \left(\begin{cases} \max(r_{U_1}(t), r_{U_2}(t)), r_{U_1}(t) \neq 0 \text{ 且 } r_{U_2}(t) \neq 0 \\ \lim_{\Delta t \rightarrow 0} \frac{r_{U_1 \cup U_2}(t - \Delta t) + r_{U_1 \cup U_2}(t + \Delta t)}{2}, r_{U_1}(t) = r_{U_2}(t) = 0 \end{cases} \right) \end{aligned} \quad (24)$$

故有:

$$\begin{aligned} s \cap (U_1 \cup U_2) &= \\ &= r \left(\max(\delta_1, \delta_2), \left\{ \begin{array}{l} t_i \in \{t_{1,i}\} \text{ 且 } r_{U_2}(t_i) = -1 \\ t_i \in \{t_{2,i}\} \text{ 且 } r_{U_1}(t_i) = -1 \\ t_i \in \{t_{1,i}\} \cap \{t_{2,i}\} \\ \text{且 } \lim_{\Delta t \rightarrow 0} r_{U_1}(t_i - \Delta t) = r_{U_2}(t_i - \Delta t) \end{array} \right\} \right) \end{aligned} \quad (25)$$

同理可得:

$$\begin{aligned} s \cap (U_1 \cap U_2) &= \\ &= r \left(\min(\delta_1, \delta_2), \left\{ \begin{array}{l} t_i \in \{t_{1,i}\} \text{ 且 } r_{U_2}(t_i) = 1 \\ t_i \in \{t_{2,i}\} \text{ 且 } r_{U_1}(t_i) = 1 \\ t_i \in \{t_{1,i}\} \cap \{t_{2,i}\} \\ \text{且 } \lim_{\Delta t \rightarrow 0} r_{U_1}(t_i - \Delta t) = r_{U_2}(t_i - \Delta t) \end{array} \right\} \right) \end{aligned} \quad (26)$$

根据式(23)、(25)、(26)就完成了线段和CSG描述图元模型(单个几何模型的数据模型)的交集求解,求解完成后可将求解结果转换为线段的并集:

$$s = r(\delta, \{t_i\}) = \bigcup s_i \quad (27)$$

式中, s_i 为对应 r 函数大于 0 的段。

结合式(27)和 2.3 节就完成了对于整个几何模型的特征线追踪。

3 特征线并行与实时生成计算流程

直接 3D MOC 计算程序极大的存储规模对机器性能要求极高。为了减少计算时间和降低存储要求,本文采用特征线并行技术,并在此基础上使用实时生成方法减少特征线分段信息的存储。

实时生成是指所有特征线的分段信息都是在使用前才生成,因为采用了第 2 节的交集求解算法,所以实时追踪时是先生成 1 整条特征线的分段信息,再在其上一次性计算完整条特征线,然后整条特征线的信息都被舍弃。该方法可以将存储空间减少到原先的千分之一甚至万分之一,但也会增加计算时间,因为这相当于每次内迭代中都要进行 1 次特征线追踪,但利用本文的特征线追踪算法,可有效减少实时生成中特征线追踪所带来的时间消耗。

实时生成计算流程如图 2 所示。由于特征线并行中能群循环被放在了特征线循环的内层,这样每条特征线一旦计算完成后,在本次内迭代中,该线上存储的信息就可以释放。其中的源场和边界场以及有源和无源计算是用于复杂反射边界条件追踪计算,参见文献 [16]。

4 数值验证

测试工作使用的处理器为 Intel Core i5 CPU, 4 核 8 线程,内存 8 GB,所有测试均是 8 线程下并行。计算中 k_{eff} 的收敛限为 1×10^{-6} , 网格通量的平均残差限定为 1×10^{-6} 。

4.1 Takeda 算例

Takeda 算例来自于日本京都大学的临界装置 KUCA, 是 OECD/NEA 基准例题“三维中子输运基准例题”中的模型,分为控制棒全插和全提两个算例^[7],其几何布置如图 3 所示。

首先测试了程序计算的正确性。计算条件

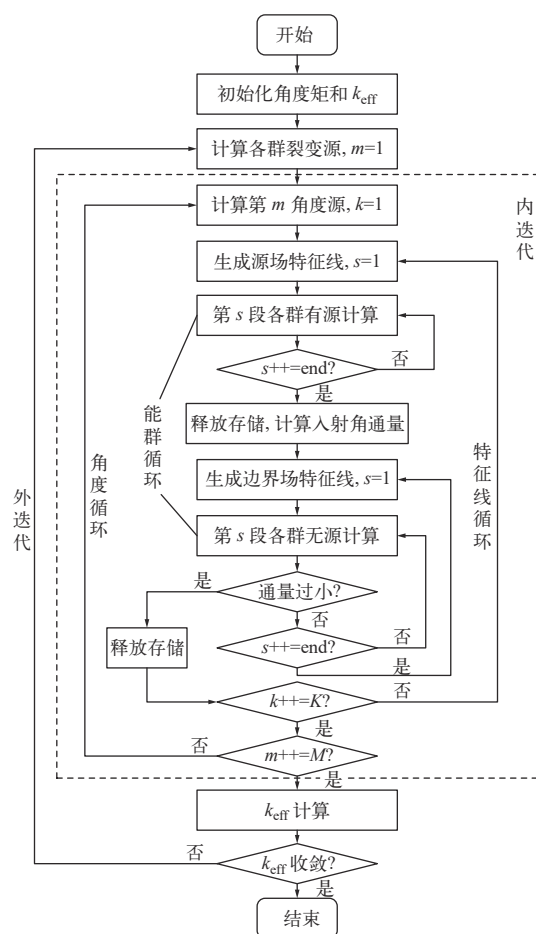


图 2 实时生成计算流程图

Fig. 2 Flowchart of on-the-fly generation method

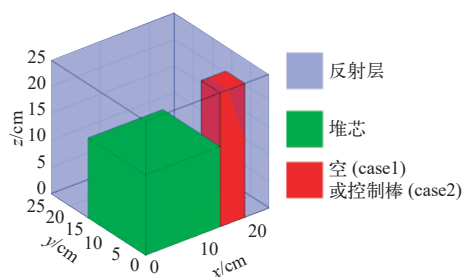


图 3 Takeda 算例

Fig. 3 Takeda benchmark

为: 网格设置为 $0.5 \text{ cm} \times 0.5 \text{ cm} \times 0.5 \text{ cm}$, 特征线密度为 $0.1 \text{ cm} \times 0.1 \text{ cm}$, 求积组为高斯-勒让德求积组, 4 个极角和 8 个方位角, 空(case1)和控制棒(case2)工况下的 k_{eff} 以及各区域的平均通量计算结果分别列于表 1~3。

从表 1~3 中可看出: 两个工况中 k_{eff} 的计算误差很小, 分别为 -20 pcm 和 17 pcm ; 两个工况下大部分区域中平均通量的相对误差均较低, 最大相对误差均出现在反射层的第 2 群中, 分

表 1 Takeda 算例的 k_{eff} 计算结果
Table 1 k_{eff} result of Takeda benchmark

对比数据	case1		case2	
	k_{eff}	误差/pcm	k_{eff}	误差/pcm
参考解	0.977 80(± 60 pcm)		0.962 40(± 60 pcm)	
计算值	0.977 60	-20	0.962 57	17

别为-3.41% 和-3.12%, 表明程序具有良好的计算精度。

为进一步检验实时生成方法对 MOC 的计算时间和空间的影响, 利用 Takeda 算例 case1, 改变不同参数进行测试。测试时采用边长为 1 cm 的网格, 极角为 4 个。

表 2 Takeda case1 算例各区域的平均通量
Table 2 Average flux of each region in Takeda case1 benchmark

对比数据	能群	空区		燃料区		反射层	
		平均通量	相对误差/%	平均通量	相对误差/%	平均通量	相对误差/%
参考解	1	$1.450 0 \times 10^{-3}$		$4.750 9 \times 10^{-3}$		$5.925 1 \times 10^{-4}$	
	2	$9.740 6 \times 10^{-4}$		$8.699 8 \times 10^{-4}$		$9.140 4 \times 10^{-4}$	
计算值	1	$1.464 3 \times 10^{-3}$	0.98	$4.724 1 \times 10^{-3}$	-0.56	$5.948 9 \times 10^{-4}$	0.40
	2	$9.537 2 \times 10^{-4}$	-2.09	$8.700 9 \times 10^{-4}$	0.01	$8.828 4 \times 10^{-4}$	-3.41

表 3 Takeda case2 算例各区域的平均通量
Table 3 Average flux of each region in Takeda case2 benchmark

对比数据	能群	控制棒		燃料区		反射层	
		平均通量	相对误差/%	平均通量	相对误差/%	平均通量	相对误差/%
参考解	1	$1.224 7 \times 10^{-3}$		$4.912 5 \times 10^{-3}$		$5.910 9 \times 10^{-4}$	
	2	$2.460 4 \times 10^{-4}$		$8.692 1 \times 10^{-4}$		$8.789 7 \times 10^{-4}$	
计算值	1	$1.224 9 \times 10^{-3}$	0.01	$4.879 4 \times 10^{-3}$	-0.67	$5.930 5 \times 10^{-4}$	0.33
	2	$2.514 9 \times 10^{-4}$	2.21	$8.693 4 \times 10^{-4}$	0.02	$8.515 1 \times 10^{-4}$	-3.12

在第 1 组测试中, 固定特征线密度为 $0.1 \text{ cm} \times 0.1 \text{ cm}$, 改变方位角数量, 测试结果列于表 4。

从表 4 可看出, 实时生成方法所需存储空间与方位角数量无关, 始终保持在 35.1 MB。在保存特征线分段信息时, 随着方位角数量增加, 存储空间呈线性增加。即使对于最少的 4 个极角、4 个方位角的情况, 实时生成方法也能减少存

储空间原先的 94.97%, 而对于极角和方位角较多的情况, 则最高可以减少原先的 99.24%。显然, 极角和方位角如果进一步增加, 减少比例还会更多。

从表 4 还可看出, 采用实时生成方法的时间增加比例始终保持在 45% 左右, 相当于 1 群左右的计算时间, 时间增加比例与方位角数量无关。

表 4 改变方位角的测试结果
Table 4 Test result of changed azimuth angle

极角数量	方位角数量	保存分段信息的存储空间/MB	实时生成的存储空间/MB	减少的空间存储比/%	保存分段信息的计算时间/s	实时生成的计算时间/s	时间增加比例/%	k_{eff} 计算偏差/pcm
4	4	698.5	35.1	94.97	1 361.52	1 940.254	42.51	-166
4	8	1 218.2	35.1	97.12	3 145.536	4 590.991	45.95	-206
4	16	2 336.8	35.1	98.50	5 414.808	7 800.432	44.06	-199
4	32	4 589.6	35.1	99.24	11 571.95	16 628.77	43.70	-205

在第2组测试中,固定极角和方位角数量为4和8个,改变特征线密度,计算结果列于表5。

从表5可看出,实时生成的存储空间同样也与特征线密度无关,始终保持在35.1 MB。在保存分段信息存储下,特征线密度和所需内存呈抛物

线关系,随着特征线密度变细,所需内存大大增加。实时生成方法减少的内存从最初的89.65%到最后的99.51%。从表5还可看出,在计算时间上,时间增加比例同样保持在45%左右,也不随特征线密度的变化而变化。

表5 改变密度测试结果
Table 5 Test result of changed density

(边长/特征线密度)/cm ⁻¹	保存分段信息的存储空间/MB	实时生成的存储空间/MB	减少的空间存储比/%	保存分段信息的计算时间/s	实时生成的计算时间/s	时间增加比例/%	k_{eff} 计算偏差/pcm
5	339	35.1	89.65	745.362	1 066.985 7	43.15	-204
10	1 218.2	35.1	97.12	3 145.536	4 584.304 2	45.74	-206
15	2 661.1	35.1	98.68	5 816.61	8 399.184 8	44.40	-207
20	4 665.3	35.1	99.25	11 759.769	16 876.444	43.51	-207
25	7 223.9	35.1	99.51	18 146.043	25 841.78	42.41	-207

总的来说,特征线分段信息存储空间与方位角数量呈正比例关系,与特征线密度呈抛物线关系。而实时生成方法的存储空间则与这些参数的设置无关,始终保持在35.1 MB。即使对于最为粗糙的参数设置,实时生成方法也可以减少89.65%的存储空间;如果要准确计算,参数需要设置为至少0.5 cm的网格边长、4个极角和8个方位角、0.1 cm×0.1 cm的特征线密度,此时可以减少99.51%以上的存储空间。

综上所述,计算时间上,实时生成方法对于Takeda算例,始终仅需增加45%左右的计算时间,但可以大量减少存储空间,显示出实时生成方法在处理3D MOC时在存储空间上具有良好效果。

4.2 C5G7算例

为了测试更加精细复杂的算例,对C5G7算例^[18]进行计算。C5G7是常用的非均匀化基准题。考虑到全堆3D计算时间过长,且C5G7算例轴向远不如径向复杂,这里将C5G7的2D算例通过补充上下全反射边界条件构造造成3D问题进行计算。C5G7算例的堆芯径向排布和边界条件如图4所示,轴向设为10 cm,分为5层,顶部和底部均设为反射边界条件。

同样地,首先对特征值等参数进行了计算,以验证程序的正确性。计算参数设置为:燃料区按照0.4、0.8、1.08 cm划分为3个环形网格、8个扇区,在反射层中的网格为0.252 cm×0.252 cm。采

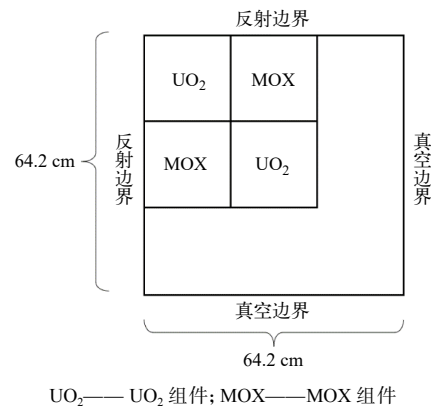


图4 C5G7算例的堆芯径向排布和边界条件

Fig. 4 Radial core arrangement and boundary condition of C5G7 benchmark

用高斯-勒让德求积组,8个极角和16个方位角,轴向高度取为1 cm,特征线密度为0.1 cm×0.1 cm。C5G7算例计算结果列于表6。表6中:AVG为算术平均值;RMS为均方根值;MRE为平均相对误差。由表6可见,C5G7算例计算结果总体上与算例参考解符合很好。

C5G7算例栅元归一化功率的分布如图5所示。C5G7算例归一化功率的相对误差如图6所示。

由图6可见,归一化功率的相对误差主要集中于栅元和慢化剂边界以及中心区域,其余区域符合很好。相对误差的最大值出现在原点处的栅元,达到3.9%。C5G7算例的计算结果进一步证明了程序在计算上的正确性。

表 6 C5G7 算例计算结果汇总
Table 6 Summary of calculation result for C5G7 benchmark

参数	参数值	
	算例参考解 (MCNP)	MOC
k_{eff}	1.186 55	1.185 19
k_{eff} 误差, pcm		-136
AVG, %	0.32	0.435
RMS, %	0.34	0.596
MRE, %	0.27	0.376
最大栅元归一化功率	2.498	2.505
最大栅元功率相对误差, %	± 0.16	0.28
最小栅元归一化功率	0.232	0.236
最小栅元功率相对误差, %	± 0.58	1.72
栅元最大相对误差, %		3.90

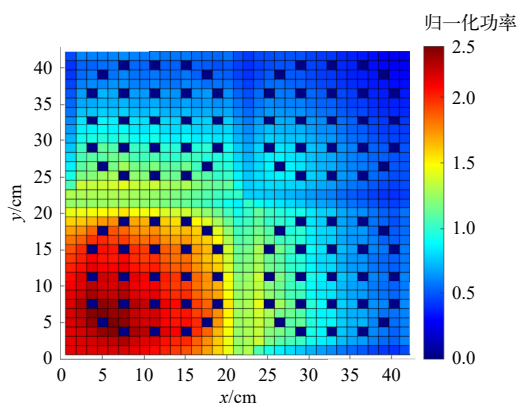


图 5 C5G7 算例栅元归一化功率的分布
Fig. 5 Pin normalized power distribution of C5G7 benchmark

需要指出的是, 上述 C5G7 算例的计算是在实时生成方法下完成的, 若要通过存储分段信息的方式完成精确计算, 需要的内存空间较大。在特征线密度为 $0.1 \text{ cm} \times 0.1 \text{ cm}$ 时, 即使只有 2 个极角、4 个方位角, 特征线密度都需要 6.94 GB 的存储空间。在该参数下的计算时间和存储空间列于表 7。

从表 7 可看出, 对于 C5G7 算例, 实时生成方法的计算时间仅增加了 22.78%, 实时生成方法的

表 7 2 个极角和 4 个方位角下的计算时间和存储空间

Table 7 Calculation time and storage requirement of 2 polar and 4 azimuth angles

求积组	保存分段信息的存储空间/MB	实时生成的存储空间/MB	减少的空间存储比/%	保存分段信息的计算时间/s	实时生成的计算时间/s	时间增加比例/%	k_{eff} 计算偏差/pcm
(2,4)	6 940	84	98.79	82 576	101 386	22.78	-434
(8,16)	111 040	84	99.92	1 215 955	1 492 950	22.78	-136

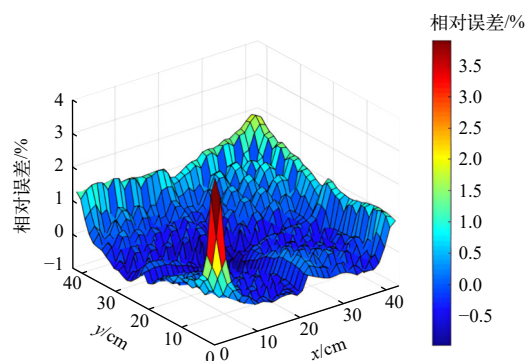


图 6 C5G7 算例的相对误差
Fig. 6 Relative error distribution of C5G7 benchmark

效果更好。这是因为 C5G7 算例虽然单根追踪时间更长, 但是由于它是 7 群问题, 而 Takeda 算例为 2 群, 所以增加的时间开销会更少。与 Takeda 算例中网格为主相比, C5G7 算例中栅格结构增多, 所以等效群数有所增加。存储规模上, 即使对于 (2, 4) 的求积组, 也可以减少 98.79% 的存储空间; 如果要准确计算, 则估计需要 108 GB 的存储空间, 但是通过实时生成方法, 能够减少 99.92% 的存储空间, 仅需不到 0.1% 的 84 MB。

5 结论

为处理直接 3D MOC 空间存储大的问题, 本文提出了一种新的特征线追踪技术以提高追踪效率、减少追踪时间, 并在此基础上实现了特征线信息的实时生成方法。首先通过 Takeda 算例、C5G7 算例在特征值、功率分布等计算结果验证了程序计算的准确性, 然后测试了高效特征线追踪和实时生成方法在 3D MOC 方法中的应用性能。结果表明, 基于本文提出的特征线追踪和实时生成方法, 分别仅需要付出增加约 45%、23% 的计算时间成本, Takeda 算例最高能够减小 99.51% 的存储空间, 而 C5G7 算例所需内存最高能减少 99.92% 的存储空间, 证明了本文提出的方法可以有效解决直接 3D MOC 的存储要求, 提高直接 3D MOC 的机器适用性。

参考文献:

- [1] CHO J Y, JOO H G, KIM K S, et al. Cell based CMFD formulation for acceleration of whole-core method of characteristics calculations[J]. *Journal of the Korean Nuclear Society*, 2002, 34(3): 250-258.
- [2] YUK S, CHO N Z. Whole-core transport solutions with 2-D/1-D fusion kernel via p-CMFD acceleration and p-CMFD embedding of nonoverlapping local/global iterations[J]. *Nuclear Science and Engineering*, 2015, 181(1): 1-16.
- [3] WU W, YU Y, LI Q, et al. Verification of the 2D/1D coupling 3D transport code TIGER with C5G7 benchmarks [J]. *International Journal of Advanced Nuclear Reactor Design and Technology*, 2019, 1: 43-50.
- [4] KELLEY B W, LARSEN E W. A consistent 2D/1D approximation to the 3D neutron transport equation[J]. *Nuclear Engineering and Design*, 2015, 295: 598-614.
- [5] CHO J Y, KIM K S, LEE C C, et al. Axial SPN and radial MOC coupled whole core transport calculation[J]. *Journal of Nuclear Science and Technology*, 2007, 44(9): 1156-1171.
- [6] JOO H G, CHO J Y, KIM K S, et al. Methods and performance of a three-dimensional whole-core transport code DeCART[C]. Chicago: [s. n.], 2004.
- [7] BAN Y S, MASIELLO E, LENAIN R, et al. Code-to-code comparisons on spatial solution capabilities and performances between nTRACER and the standalone IDT solver of APOLLO3®[J]. *Annals of Nuclear Energy*, 2018, 115: 573-594.
- [8] CHEN J, LIU Z, ZHAO C, et al. A new high-fidelity neutronics code NECP-X[J]. *Annals of Nuclear Energy*, 2018, 116: 417-428.
- [9] LIU Z, WU H, CAO L, et al. A new three-dimensional method of characteristics for the neutron transport calculation[J]. *Annals of Nuclear Energy*, 2011, 38(2/3): 447-454.
- [10] KOCHUNAS B M. A hybrid parallel algorithm for the 3-D method of characteristics solution of the Boltzmann transport equation on high performance compute clusters[R]. [S. l.]: [s. n.], 2013.
- [11] SANTANDREA S, GRAZIANO L, SCIANNANDRONE D. Accelerated polynomial axial expansions for full 3D neutron transport MOC in the APOLLO3® code system as applied to the ASTRID fast breeder reactor[J]. *Annals of Nuclear Energy*, 2018, 113: 194-236.
- [12] SCIANNANDRONE D, SANTANDREA S, SANCHEZ R. Optimized tracking strategies for step MOC calculations in extruded 3D axial geometries[J]. *Annals of Nuclear Energy*, 2016, 87: 49-60.
- [13] GRAZIANO L, SANTANDREA S, SCIANNANDRONE D. Polynomial axial expansion in the method of characteristics for neutron transport in 3D extruded geometries[J]. *EPJ Web of Conferences*, 2017, 153: 06027.
- [14] WU W, YU Y, LUO Q, et al. Verification of the 3D capability of OpenMOC with the C5G7 3D extension benchmark[J]. *Annals of Nuclear Energy*, 2020, 140: 107293.
- [15] OpenMOC Development Team. OpenMOC: A method of characteristic code for nuclear reactor physics calculations[EB/OL]. <https://github.com/mit-crpg/OpenMOC>.
- [16] 杨睿, 胡赞, 单浩栋, 等. 适用于任意几何的特征线边界条件处理方法[J]. *原子能科学技术*, 2020, 54(11): 2145-2152.
- YANG Rui, HU Yun, SHAN Haodong, et al. Boundary condition processing method for MOC calculation in arbitrary geometry[J]. *Atomic Energy Science and Technology*, 2020, 54(11): 2145-2152 (in Chinese).
- [17] TAKEDA T, IKEDA H. 3-D neutron transport benchmarks[J]. *Journal of Nuclear Science and Technology*, 1991, 28(7): 656-669.
- [18] SMITH M, LEWIS E, NA B. Benchmark on deterministic 3-D MOX fuel assembly transport calculations without spatial homogenization[J]. *Progress in Nuclear Energy*, 2004, 45: 383-393.