



高等反应堆物理 I 课程作业

中子扩散程序开发与验证

学院（部、中心）： 能动学院

班 级： B2437

学生姓名： 史锦康

学 号： 4124103139

课程教师： 李云召

2024 年 11 月

摘 要

本文推导了有限差分法下的一维中子扩散方程求解方法，并基于该方法开发了一维单群中子扩散求解程序 ONED，该程序用 MATLAB 语言编写，支持多种材料、多个分段非均匀网格输入。

ONED 程序能够正确计算所给的临界计算基准题，在合适的网格划分条件下， k_{eff} 计算偏差可控制在 1pcm 以内。验证了程序的分段无关性，从而验证了非均匀网格功能的正确性。研究了网格数和网格划分结构对计算结果的影响，得到了网格数、迭代时间、计算偏差之间的定性关系。

本文提出了一种自适应非均匀网格优化方法，经基准题验证，能有效地在不增加网格数的情况下减小计算偏差。

关 键 词：中子扩散方程；有限差分法；非均匀网格；优化方法

目 录

摘 要.....	I
1 有限差分法求解一维中子扩散方程理论推导	1
1.1 一维中子扩散方程	1
1.2 有限差分法的数值离散过程	1
1.2.1 基本思想	1
1.2.2 数值离散	1
1.2.3 区间积分	1
1.2.4 边界条件	2
1.2.5 构建线性代数系统	3
1.3 有限差分法的数值求解过程	4
1.3.1 裂变源迭代	4
1.3.2 散射源迭代	5
1.3.3 群内计算	5
2 一维中子扩散有限差分程序开发	6
2.1 概述	6
2.2 程序计算流程	6
2.3 程序迭代过程	7
2.4 程序输出文件	7
3 一维中子扩散有限差分程序验证	8
3.1 基准题	8
3.2 程序正确性验证	8
3.3 分段无关性验证	9
3.4 网格数对计算结果的影响	9
3.5 非均匀网格优化计算	10
3.5.1 非均匀网格对计算偏差的影响	10
3.5.2 自适应非均匀网格优化方法	10
4 结论与展望	12
4.1 结论	12
4.2 展望	12
参考文献	13
附 录	14

1 有限差分法求解一维中子扩散方程理论推导

本章将推导直角几何下一维中子扩散方程的有限差分数值解法。

1.1 一维中子扩散方程

多群稳态中子扩散方程如公式（1-1）所示^[1]：

$$\begin{aligned} -\nabla[D_g(r)\nabla\Phi_g(r)] + \Sigma_{r,g}(r)\Phi_g(r) &= Q_g(r) \\ Q_g(r) &= \sum_{g' \neq g} \Sigma_{s,gg'}(r)\Phi_{g'}(r) + \frac{1}{k} \chi_g(r) \sum_{g'} \nu \Sigma_{f,g'}(r)\Phi_{g'}(r) \end{aligned} \quad (1-1)$$

在一维情况下，方程变为：

$$\begin{aligned} -\frac{d}{dx}[D_g(x)\frac{d}{dx}\Phi_g(x)] + \sum_{r,g}(x)\Phi_g(x) &= Q_g(x) \\ Q_g(x) &= \sum_{g' \neq g} \Sigma_{s,gg'}(x)\Phi_{g'}(x) + \frac{1}{k} \chi_g(x) \sum_{g'} \nu \Sigma_{f,g'}(x)\Phi_{g'}(x) \end{aligned} \quad (1-2)$$

1.2 有限差分法的数值离散过程

1.2.1 基本思想

有限差分法的基本思想：在足够小的网格下，用微商代替微分，将连续型中子扩散方程转化为离散型中子扩散方程（包括边界条件）。

经过数值离散的中子扩散方程会形成线性代数系统，再通过数值求解（直接解法或迭代解法）过程，从而获得离散的中子通量密度和中子流密度。

1.2.2 数值离散

如图 1-1 所示，将求解区域分解成 N 个网格。其中各网点位置为 x_i ，各网点上的中子通量密度为 $\Phi_{i,g}$ ， $i=0 \sim N$ ，网格区间长度 $\Delta_i = x_i - x_{i-1}$ ，各网格内的材料参数为 $D_{i,g}, \Sigma_{x,g,i}$ ， $i=1 \sim N$ ，网格中心点 $x_{i-1/2} = x_i - \Delta_i / 2$ ， $i=1 \sim N$ ，连续方程被离散到单个网格上，网格间满足连续边界条件，外边界条件保持不变。

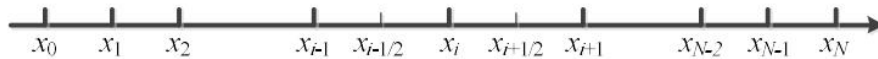


图 1-1 一维有限差分法数值离散网格

1.2.3 区间积分

对于没有处在边界区域的中间网点，对公式（1-2）在区间 $(x_{i-1/2}, x_{i+1/2})$ 内进行积分。如公式（1-3）和公式（1-4）所示

$$\begin{aligned}
 & \int_{x_{i-1/2}}^{x_{i+1/2}} -\frac{d}{dx} [D_g(x) \frac{d}{dx} \Phi_g(x)] dx \\
 &= -D_{g,i+1} \frac{d}{dx} \Phi_g(x_{i+1/2}) + D_{g,i} \frac{d}{dx} \Phi_g(x_{i-1/2}) \\
 &= -D_{g,i+1} \frac{\Phi_{g,i+1} - \Phi_{g,i}}{\Delta_{i+1}} + D_{g,i} \frac{\Phi_{g,i} - \Phi_{g,i-1}}{\Delta_i}
 \end{aligned} \tag{1-3}$$

$$\begin{aligned}
 & \int_{x_{i-1/2}}^{x_{i+1/2}} \sum_{x,g} (x) \Phi_g(x) dx \\
 &= \int_{x_{i-1/2}}^{x_i} \sum_{x,g} (x) \Phi_g(x) dx + \int_{x_i}^{x_{i+1/2}} \sum_{x,g} (x) \Phi_g(x) dx \\
 &= \sum_{x,g,i} \Phi_{g,i} \frac{\Delta_i}{2} + \sum_{x,g,i+1} \Phi_{g,i} \frac{\Delta_{i+1}}{2}
 \end{aligned} \tag{1-4}$$

将公式 (1-3) 和公式 (1-4) 代入公式 (1-2) 中得到公式 (1-5)

$$\begin{aligned}
 & -\frac{D_{g,i}}{\Delta_i} \Phi_{g,i-1} + \left(\frac{D_{g,i+1}}{\Delta_{i+1}} + \frac{D_{g,i}}{\Delta_i} + \sum_{r,g,i} \frac{\Delta_i}{2} + \sum_{r,g,i+1} \frac{\Delta_{i+1}}{2} \right) \Phi_{g,i} - \frac{D_{g,i+1}}{\Delta_{i+1}} \Phi_{g,i+1} = Q_{g,i} \\
 Q_{g,i} &= \sum_{g' \neq g} \left(\sum_{s,gg',i} \frac{\Delta_i}{2} + \sum_{s,gg',i+1} \frac{\Delta_{i+1}}{2} \right) \Phi_{g',i} + \frac{1}{k} \sum_{g'} (\chi_{g,i} \nu \sum_{r,g',i} \frac{\Delta_i}{2} + \chi_{g,i+1} \nu \sum_{r,g',i+1} \frac{\Delta_{i+1}}{2}) \Phi_{g,i}
 \end{aligned} \tag{1-5}$$

公式 (1-5) 可化简为公式 (1-6) 和公式。

$$a_i \Phi_{g,i-1} + b_i \Phi_{g,i} + c_i \Phi_{g,i+1} = Q_{g,i} \tag{1-6}$$

$$\begin{aligned}
 a_i &= -\frac{D_{g,i}}{\Delta_i} \\
 b_i &= \sum_{r,g,i} \frac{\Delta_i}{2} + \sum_{r,g,i+1} \frac{\Delta_{i+1}}{2} + \frac{D_{g,i+1}}{\Delta_{i+1}} + \frac{D_{g,i}}{\Delta_i} \\
 c_i &= -\frac{D_{g,i+1}}{\Delta_{i+1}}
 \end{aligned} \tag{1-7}$$

1.2.4 边界条件

1) 对称边界条件 (以左边界为例)

在对称边界条件下, 有如下公式 (1-8) 成立。

$$0 = \frac{d}{dx} \Phi_g(x_0) = \frac{\Phi_{g,1} - \Phi_{g,0}}{\Delta_1} \rightarrow \Phi_{g,1} = \Phi_{g,0} \tag{1-8}$$

将公式 (1-8) 代入公式 (1-6), 可得公式 (1-9)。

$$(a_1 + b_1) \Phi_{g,1} + c_1 \Phi_{g,2} = Q_{g,1} \tag{1-9}$$

2) 真空边界条件 (以右边界为例)

在真空边界条件下, 有如下公式 (1-10) 成立。

$$0 = \frac{1}{4}\Phi_g(x_s) - \frac{1}{2} \langle n_s, -D_g(r_s) \frac{d}{dx} \Phi_g(x_s) \rangle = \frac{1}{4}\Phi_{g,N} + \frac{1}{2}D_{g,N} \frac{\Phi_{g,N} - \Phi_{g,N-1}}{\Delta_N} \quad (1-10)$$

化简可得公式 (1-11)。

$$\Phi_{g,N} = \frac{2D_{g,N}}{\Delta_N + 2D_{g,N}} \Phi_{g,N-1} \quad (1-11)$$

将公式 (1-11) 代入 (1-6)，可得公式 (1-12)。

$$a_{N-1}\Phi_{g,N-2} + (b_{N-1} + c_{N-1} \frac{2D_{g,N}}{\Delta_N + 2D_{g,N}})\Phi_{g,N-1} = Q_{g,N-1} \quad (1-12)$$

3) 外推边界条件 (以右边界为例)

在外推边界条件下，有如下公式 (1-13) 成立。

$$\Phi_g(x_s) = 0 \rightarrow \Phi_{g,N} = 0 \quad (1-13)$$

将公式 (1-13) 代入 (1-6)，可得公式 (1-14)。

$$a_{N-1}\Phi_{g,N-2} + b_{N-1}\Phi_{g,N-1} = Q_{g,N-1} \quad (1-14)$$

1.2.5 构建线性代数系统

每个能群的 $N-1$ 个离散方程构成线性代数方程组，如公式 (1-15) 所示：

$$\begin{aligned} b_{l,1}\Phi_{g,1} + c_1\Phi_{g,2} &= Q_{g,1} \\ &\vdots \\ a_i\Phi_{g,i-1} + b_i\Phi_{g,i} + c_i\Phi_{g,i+1} &= Q_{g,i} \\ &\vdots \\ a_{N-1}\Phi_{g,N-2} + b_{N-1}\Phi_{g,N-1} &= Q_{g,N-1} \end{aligned} \quad (1-15)$$

将公式 (1-15) 写成矩阵形式，如公式 (1-16)、公式 (1-17)、公式 (1-18)、公式 (1-19) 所示。

$$\begin{aligned} \mathbf{M}_g \Phi_g &= \mathbf{Q}_g \\ \mathbf{Q}_g &= \sum_{g' \neq g} \Gamma_{s,gg'} \Phi_{g'} + \frac{1}{k} \sum_{g'} \Gamma_{f,gg'} \Phi_{g'} \end{aligned} \quad (1-16)$$

$$\begin{aligned} \Phi_g &= [\Phi_{g,1}, \dots, \Phi_{g,i}, \dots, \Phi_{g,N-1}]^T \\ \mathbf{Q}_g &= [Q_{g,1}, \dots, Q_{g,i}, \dots, Q_{g,N-1}]^T \end{aligned} \quad (1-17)$$

$$\begin{aligned} \Gamma_{s,gg'} &= \text{diag} \left\{ \frac{1}{2} (\Sigma_{s,gg'i} \Delta_i + \Sigma_{s,gg'i+1} \Delta_{i+1}) \right\} \\ \Gamma_{f,gg'} &= \text{diag} \left\{ \frac{1}{2} (\chi_{g,i} \nu \Sigma_{f,g'i} \Delta_i + \chi_{g,i+1} \nu \Sigma_{f,g'i+1} \Delta_{i+1}) \right\} \end{aligned} \quad (1-18)$$

$$\mathbf{M}_g = \begin{bmatrix} b_{l,1} & c_1 & & & \\ a_2 & b_2 & c_2 & & \\ & \ddots & \ddots & \ddots & \\ & & a_{N-2} & b_{N-2} & c_{N-2} \\ & & & a_{N-1} & b_{r,N-1} \end{bmatrix} \quad (1-19)$$

将所有能群写到一起。得到公式 (1-20)

$$\mathbf{M}\Phi = \mathbf{S}\Phi + \frac{1}{k}\mathbf{F}\Phi \quad (1-20)$$

其中：

$$\begin{aligned} \mathbf{M} &= \text{diag}\{\mathbf{M}_g\} \\ \mathbf{S} &= [\mathbf{S}_1^T, \dots, \mathbf{S}_g^T, \dots, \mathbf{S}_G^T]^T \\ \mathbf{F} &= [\mathbf{F}_1^T, \dots, \mathbf{F}_g^T, \dots, \mathbf{F}_G^T]^T \end{aligned} \quad (1-21)$$

散射和裂变耦合了不同的能群：

$$\begin{aligned} \mathbf{Q}_g &= \mathbf{S}_g\Phi + \frac{1}{k}\mathbf{F}_g\Phi \\ \mathbf{S}_g &= [\Gamma_{s,g1}, \dots, \Gamma_{s,gg-1}, 0, \Gamma_{s,gg+1}, \dots, \Gamma_{s,gG}] \end{aligned} \quad (1-22)$$

$$\begin{aligned} \mathbf{F}_g &= [\Gamma_{f,g1}, \dots, \Gamma_{f,gg}, \dots, \Gamma_{f,gG}] \\ \Phi &= [\Phi_1^T, \dots, \Phi_g^T, \dots, \Phi_G^T]^T \end{aligned} \quad (1-23)$$

1.3 有限差分法的数值求解过程

1.3.1 裂变源迭代

1. 给定初值： $k^{(0)}$ 和 $\Phi^{(0)}$
2. 获得裂变源初值： $f^{(1)} = F\Phi^{(0)}$
3. 已知 $k^{(0)}$ 和 $f^{(n)}$ ，求解方程更新中子通量密度 $\Phi(n)$ ：

$$(\mathbf{M} - \mathbf{S})\Phi^{(n)} = \frac{1}{k^{(n-1)}} f^{(n)} \quad (1-24)$$

4. 更新裂变源： $f^{(n+1)} = F\Phi^{(n)}$

$$5. \text{更新有效增殖因子： } k^{(n)} = \frac{f^{(n+1)}_2}{\frac{1}{k^{(n-1)}} f^{(n)}_2}$$

6. 按照以下标准判断收敛：如果同时成立则结束；否则返回步骤 4 继续

$$\left| \frac{k^{(n)} - k^{(n-1)}}{k^{(n)}} \right| \leq \varepsilon_k \quad \max \left| \frac{f_i^{(n+1)} - f_i^{(n)}}{f_i^{(n+1)}} \right| \leq \varepsilon_f \quad (1-25)$$

1.3.2 散射源迭代

扫描所有能群，求解如下线性代数系统：

$$\begin{bmatrix} \mathbf{M}_1 & & & & \\ & \ddots & & & \\ & & \mathbf{M}_g & & \\ & & & \ddots & \\ & & & & \mathbf{M}_G \end{bmatrix} \begin{bmatrix} \Phi_1 \\ \vdots \\ \Phi_g \\ \vdots \\ \Phi_G \end{bmatrix} = \begin{bmatrix} 0 & \Sigma_{s,12} & \cdots & \cdots & \Sigma_{s,1G} \\ \Sigma_{s,21} & 0 & & & \vdots \\ \vdots & & \ddots & & \vdots \\ \vdots & & & 0 & \Sigma_{s,G-1G} \\ \Sigma_{s,G1} & \cdots & \cdots & \Sigma_{s,GG-1} & 0 \end{bmatrix} \begin{bmatrix} \Phi_1 \\ \vdots \\ \Phi_g \\ \vdots \\ \Phi_G \end{bmatrix} + \begin{bmatrix} \tilde{f}_1 \\ \vdots \\ \tilde{f}_g \\ \vdots \\ \tilde{f}_G \end{bmatrix} \quad (1-26)$$

若有上散射，依次扫描包含上散射的能群，并判断有上散射的能群通量是否收敛。

1.3.3 群内计算

求解如下线性代数系统，从而获得每一能群内求解通量分布。

$$\begin{bmatrix} b_{l,1} & c_1 & & & \\ a_2 & b_2 & c_2 & & \\ & \ddots & \ddots & \ddots & \\ & & a_{N-2} & b_{N-2} & c_{N-2} \\ & & & a_{N-1} & b_{r,N-1} \end{bmatrix} \begin{bmatrix} \Phi_1 \\ \Phi_2 \\ \vdots \\ \Phi_{N-2} \\ \Phi_{N-1} \end{bmatrix} = \begin{bmatrix} Q_1 \\ Q_2 \\ \vdots \\ Q_{N-2} \\ Q_{N-1} \end{bmatrix} \quad (1-27)$$

2 一维中子扩散有限差分程序开发

2.1 概述

一维中子扩散程序 ONED 基于 MATLAB 编写，ONED 程序采用单群计算，所以只进行裂变源迭代，不进行散射迭代和群间迭代。ONED 程序支持划分非均匀网格，支持任意分段任意材料数，边界条件目前支持反射边界条件和真空边界条件，输入卡片采用 `xlsx` 表格文件，输出文件有 MATLABfigure 文件、GIF 文件、关键参数迭代过程输出表格文件。

2.2 程序计算流程

ONED 程序的计算流程如图 2-1 所示

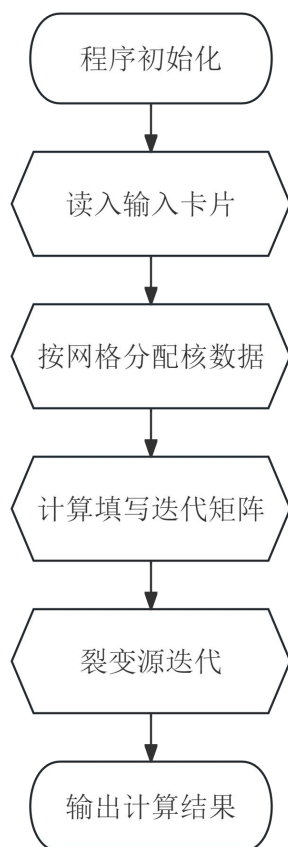


图 2-1 ONED 程序计算流程

程序输入卡片格式如图 2-2 所示，其中边界条件填 1 为反射边界条件、填 2 为真空边界条件，有效增殖因子收敛限和裂变源收敛限推荐值为 $1.0\text{E-}5$ ，最大迭代次数推荐值为 30 次，分段数和材料数必须与实际所填写的分段个数和材料个数相匹配。

分段数	材料数	最大迭代次数	有效增殖因子收敛限	裂变源收敛限	左侧边界条件	右侧边界条件
第一段长度	第二段长度	第N段长度			
第一段网格数	第二段网格数	第N段网格数			
第一段材料号	第二段材料号	第N段材料号			
材料号1	总截面	有效裂变中子数*裂变截面	散射截面			
材料号2	总截面	有效裂变中子数*裂变截面	散射截面			
.....			

图 2-2 ONED 程序输入卡片格式

2.3 程序迭代过程

1. 给定初值： $k^{(0)}$ 和 $\Phi^{(0)}$
2. 获得裂变源初值： $f^{(1)} = F\Phi^{(0)}$
3. 已知 $k^{(0)}$ 和 $f^{(n)}$ ，求解方程更新中子通量密度 $\Phi(n)$ ：

$$(M - S)\Phi^{(n)} = \frac{1}{k^{(n-1)}} f^{(n)} \quad (2-1)$$

4. 更新裂变源： $f^{(n+1)} = F\Phi^{(n)}$
5. 更新有效增殖因子： $k^{(n)} = \frac{f^{(n+1)}_2}{\frac{1}{k^{(n-1)}} f^{(n)}_2}$
6. 按照以下标准判断收敛：如果同时成立则结束；否则返回步骤 4 继续

$$\left| \frac{k^{(n)} - k^{(n-1)}}{k^{(n)}} \right| \leq \varepsilon_k \quad \max \left| \frac{f_i^{(n+1)} - f_i^{(n)}}{f_i^{(n+1)}} \right| \leq \varepsilon_f \quad (2-2)$$

2.4 程序输出文件

程序的输出如表 2-1 所示。

表 2-1 程序输出信息表

信息	格式	查看区域
迭代过程数据	数字	屏幕
迭代结果	数字	屏幕
每次迭代的 k_{eff} 数值	xlsx 表格	当前文件夹
最终中子通量分布图	figure	屏幕
中子通量迭代过程 GIF	GIF	当前文件夹

3 一维中子扩散有限差分程序验证

3.1 基准题

采用一个一维直角几何单群特征值问题作为基准题，该问题为单群问题，包含 2 种材料，右边真空边界条件，左边为全反射边界条件。该问题材料布置如下图，其中材料区 1 长为 2cm，材料区 2 长为 3cm。几何布置如图 3-1 所示：



图 3-1 一维直角几何单群特征值问题几何布置

截面参数如表 3-1 所示：

表 3-1 一维直角几何单群特征值问题截面参数

	Σ_t	$\nu\Sigma_f$	Σ_s
材料 1	1.0	1.0	0.5
材料 2	0.8	0.0	0.4

参考解如下： $n \cdot \text{cm}^{-2} \cdot \text{s}^{-1}$

表 3-2 一维直角几何单群特征值问题参考解

	参考解
k_{eff}	1.64524

3.2 程序正确性验证

利用 ONED 程序对 3.1 中的基准题进行计算，材料 1 所在区域划分为 4850 个网格，材料 2 所在区域划分为 150 个网格， k_{eff} 和裂变源迭代的收敛限全部设置为 1E-05。

迭代 12 次后得到符合收敛限的结果，数值计算结果如表 3-3 所示， k_{eff} 的数值计算结果为 1.64522，与参考解的偏差为-0.8pcm，说明数值计算的结果相当准确。

表 3-3 基准题计算结果对比

	参考解	数值解	偏差/pcm
k_{eff}	1.64524	1.64522	-0.8

该问题的中子通量分布数值计算结果如图 3-2 所示，位置 0~2cm 处为凸曲线，2~5cm 处为凹曲线，符合理论预期。

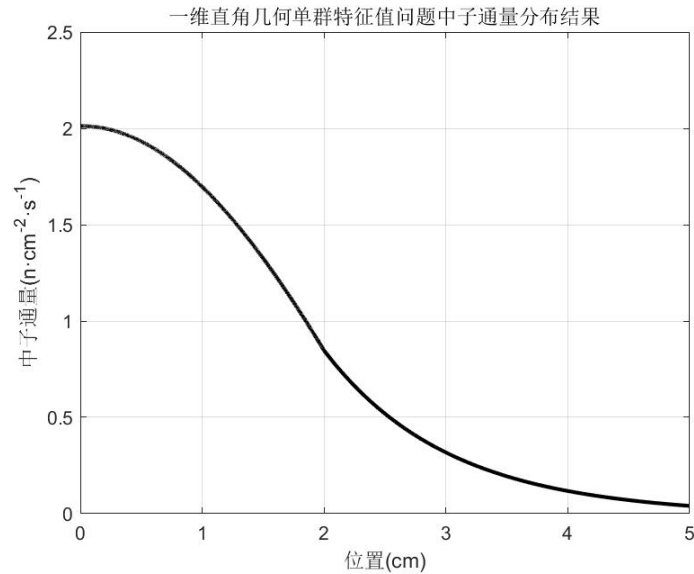


图 3-2 基准题中子通量分布数值计算结果

3.3 分段无关性验证

为了验证分段对程序计算的结果无影响，对于同样的网格密度，采用不同的分段形式来计算基准题，计算结果如表 3-4 所示。

表 3-4 分段无关性验证计算结果

分段方案 (cm)	网格分布	网格密度	k_{eff}
2+3	4500+500	1/2500+3/500	1.645185
1+1+3	2250+2250+500	1/2500+1/2500+3/500	1.645185
1+1+1.5+1.5	2250+2250+250+250	1/2500+1/2500+3/500+3/500	1.645185

从表 3-4 中可以发现，在保持相同的网格密度的前提下，对于不同的分段方案， k_{eff} 的计算结果保持一致，从而验证了分段对程序计算结果无影响。

3.4 网格数对计算结果的影响

为了研究网格数量对程序计算结果的影响，在均匀网格条件下，采用不同数量的网格对基准题进行计算，其他计算参数设置与 3.2 中相同，计算结果如表 3-5 所示。

表 3-5 不同网格数量对计算结果的影响

网格划分	k_{eff}	反应性	偏差 (pcm)	迭代历时 (s)
600+900	1.64196	0.390970	-121.5	0.59
1000+1500	1.64344	0.391522	-66.4	1.71
1200+1800	1.64381	0.391657	-52.9	2.83
2000+3000	1.64456	0.391936	-25.0	6.89

从表 3-5 中可以发现, 网格数和偏差总体呈反比, 迭代用时与网格数的平方呈正比, 迭代用时开方与偏差成反比, 这说明随着网格数目的上升, 降低单位偏差所需的计算耗时越来越多。

3.5 非均匀网格优化计算

3.5.1 非均匀网格对计算偏差的影响

由于在较大的网格数目下, 继续提升网格数不仅对偏差减小的帮助十分有限, 而且会使迭代用时呈平方倍数地增加, 所以研究如何利用非均匀网格优化网格分配从而降低计算偏差显得十分重要。本节以 3.1 中的基准题为例, 在计算网格总数保持在 5000 的条件下, 研究非均匀网格对计算偏差的影响。

考虑到在该问题中, 裂变源都集中在材料 1 所在区域, 所以采取加密材料 1 所在区域的策略进行网格分配。不同加密程度的计算结果如表 3-6 所示。

表 3-6 裂变区网格加密计算结果

网格划分		偏差 (pcm)	迭代历时(s)
材料 1 (2cm)	材料 2 (3cm)		
2000	3000	-25.0	6.89
2778	2222	-13.4	6.58
3000	2000	-11.2	6.57
4000	1000	-4.3	6.43
4500	500	-2.0	6.92
4750	250	-1.1	6.71
4850	150	-0.8	6.74
4950	50	-2.3	6.85

从表 3-6 中的结果可以看出, 对于材料 1 所在区域的网格加密可在保持迭代用时基本不变的前提下有效降低偏差。虽然材料 2 所在区域所需的网格密度远低于材料 1 所在区域, 但其网格密度过低时也会增大计算偏差。

3.5.2 自适应非均匀网格优化方法

为了进一步降低偏差, 本文提出一种针对裂变区域的自适应非均匀网格优化方法: 在裂变区内, 在一次均匀网格计算的基础上, 根据均匀网格计算所得到的裂变区通量结果来调整不同计算段的网格数量, 优化后的网格数可用公式 (3-1) 计算。

$$N_i = N_{i,uniform} \cdot (1 + \Delta w_i) \quad (3-1)$$

其中 $N_{i,uniform}$ 是第 i 个裂变源计算段采用均匀计算时的网格数, Δw_i 是第 i 个裂变源分段的网格数量权重的相对调整值, 对于内部只有一种材料的裂变源分段, Δw_i 可用公式 (3-2) 计算。

$$\Delta w_i = \left(\sqrt{\frac{\bar{\phi}_i}{\bar{\phi}}} - 1 \right) \cdot \frac{L_i}{L} \quad (3-2)$$

其中 $\bar{\phi}_i$ 是第 i 个裂变源分段的平均通量, $\bar{\phi}$ 是裂变源区域的平均通量, L_i 是第 i 个裂变源分段的长度, L 是裂变源区域的总长度。

对于内部有多种材料的裂变源分段, Δw_i 可用公式 (3-2) 计算。

$$\Delta w_i = \left(\sqrt{\frac{\bar{f}_i}{\bar{f}}} - 1 \right) \cdot \frac{L_i}{L} \quad (3-3)$$

其中 \bar{f}_i 是第 i 个裂变源分段的平均裂变源强, \bar{f} 是裂变源区域的平均裂变源强。对于某个网点 j , $f_j = \nu_j \Sigma_{t,j} \cdot \phi_j$ 。

采用上述优化方法对计算网格进行优化, 计算结果如表 3-7 所示。

表 3-7 网格优化前后基准题计算结果对比

网格划分				偏差 (pcm)
	材料 1 (1cm)	材料 1 (1cm)	材料 2 (3cm)	
优化前	2425	2425	150	-0.822
优化后	2540	2310	150	-0.053
网格划分				偏差 (pcm)
	材料 1 (1cm)	材料 1 (1cm)	材料 2 (3cm)	
优化前	2000	2000	1000	-4.3
优化后	2121	1879	1000	-3.1

从表 3-7 中可看出, 网格优化后, 计算偏差减小。说明自适应非均匀网格优化方法能有效地在不增加网格数的情况下减小偏差。

4 结论与展望

4.1 结论

基于一维中子扩散方程和有限差分计算方法所开发的一维单群中子扩散程序 ONED 能正确计算所给的基准题，在非均匀网格下，偏差可控制在 1pcm 以内。

本文提出的自适应非均匀网格优化方法能有效地在不增加网格数的情况下减小偏差。

4.2 展望

后续可为程序增加散射源迭代和群间迭代功能，从而支持多群计算。

参考文献

- [1] 吴宏春. 高等反应堆物理 [M]. 北京: 科学技术出版社, 2023: 100-120.

附 录

源码主函数：

```

clc;clear;
format long
input=readmatrix('input1.xlsx');
N_max=input(1,3);%最大迭代次数
yip_k = input(1,4);
yip_f = input(1,5);
left = input(1,6);%left boundary
right = input(1,7);%right boundary
n=input(1,1);%分段数
n_material=input(1,2);
Lx(1:n)=zeros;
Nx(1:n)=zeros;
for i=1:1:n
    Lx(1,i)=input(2,i);% x 方向长度
end
for i=1:1:n
    Nx(1,i)=input(3,i);%x 方向网格数
end
sumNx = sum(Nx);
Delta(1:sum(Nx)) =zeros;% 网格大小/网格区间长度
tep=1;
for j=1:1:n
    for i=tep:1:(tep+Nx(j)-1)
        Delta(i)= Lx(j)/Nx(j);
    end
    tep=i+1;
end
% 截面参数(假设常数)
Sigma_t_tep(1,n)=zeros;
fission_tep(1,n)=zeros;
Sigma_s_tep(1,n)=zeros;
for i=1:1:n
    index=4+input(4,i);
    Sigma_t_tep(1,i)=input(index,2);%总截面
    fission_tep(1,i)=input(index,3); %裂变截面乘有效裂变中子数
    Sigma_s_tep(1,i)=input(index,4);%散射截面
end
Sigma_a_tep= Sigma_t_tep-Sigma_s_tep;%吸收截面

```

```

Sigma_f_tep= Sigma_a_tep;% 裂变截面

phi_n=ones((sum(Nx)-1),N_max+1);%中子通量初值
M((sumNx-1):(sumNx-1)) = zeros; % 消失项矩阵
F((sumNx-1):(sumNx-1)) = zeros; % 裂变源项矩阵
S= zeros((sumNx-1),(sumNx-1)); % 散射源项矩阵
Q= zeros((sum(Nx)-1),1);
k_n(1:N_max+1)=ones; %有效增殖因子初值

D(1:sumNx)=zeros; %系数和截面分配到每个网格中
Sigma_t(1:sumNx)=zeros;
Sigma_s(1:sumNx)=zeros;
fission(1:sumNx)=zeros;
tep=1;
for j=1:1:n
    for i=tep:1:(tep+Nx(j)-1)
        fission(i)= fission_tep(1,j);
        D(i)= 1/(Sigma_t_tep(1,j)*3);
        Sigma_t(i)= Sigma_t_tep(1,j);
        Sigma_s(i)= Sigma_s_tep(1,j);
    end
    tep=i+1;
end

% xp(1:(sum(Nx)+1))=zeros; %网点位置
% for i=1:1:sum(Nx(1:1))
%     xp(i)=(i-1)*Delta(1);
% end
% for i=(sum(Nx(1:1))+1):1:sum(Nx(1:2))
%     xp(i)=xp(i-1)+Delta(2);
% end

%填迭代矩阵
%左侧边界条件
if left==1
    M(1,1) = Sigma_t(1)*Delta(1)/2 ...
        + Sigma_t(2)*Delta(2) + D(2)/Delta(2) ;
    M(1,2) = -D(2)/Delta(2);
elseif left==2
    i=1;
    a(i) = -D(i)/Delta(i);
    b(i) = Sigma_t(i)*Delta(i)/2 + ...
        Sigma_t(i+1)*Delta(i+1)/2 + ...
        D(i+1)/Delta(i+1) + D(i)/Delta(i);
    c(i) = -D(i+1)/Delta(i+1);
    M(1,1) = b(i) + c(i)*2*D(i+1)/(Delta(i+1)+2*D(i+1));
    M(1,2) = a(i);
end

```

```

%填写 M 矩阵
a(1:sumNx)=zeros;
b(1:sumNx)=zeros;
c(1:sumNx)=zeros;
for i=2:1:(sumNx-2)
    a(i) = -D(i)/Delta(i);
    b(i) = Sigma_t(i)*Delta(i)/2 + ...
          Sigma_t(i+1)*Delta(i+1)/2 + ...
          D(i+1)/Delta(i+1) + D(i)/Delta(i);
    c(i) = -D(i+1)/Delta(i+1);
    M(i,i-1) = a(i);
    M(i,i) = b(i);
    M(i,i+1) = c(i);
end

%右侧边界条件
if right==1
    M((sumNx-1),(sumNx-1)) = Sigma_t(1)*Delta(1)/2 ...
        + Sigma_t(2)*Delta(2) + D(2)/Delta(2) ;
    M((sumNx-1),(sumNx-2)) = -D(2)/Delta(2);
elseif right==2
    i=i+1;
    a(i) = -D(i)/Delta(i);
    b(i) = Sigma_t(i)*Delta(i)/2 + ...
          Sigma_t(i+1)*Delta(i+1)/2 + ...
          D(i+1)/Delta(i+1) + D(i)/Delta(i);
    c(i) = -D(i+1)/Delta(i+1);
    M((sumNx-1),(sumNx-1)) = b(i) + c(i)*2*D(i+1)/(Delta(i+1)+2*D(i+1));
    M((sumNx-1),(sumNx-2)) = a(i);
end

%填写 F 和 S 矩阵
for i=1:1:(sumNx-1)
    F(i,i) = (fission(i)*Delta(i)+fission(i+1)*Delta(i+1))/2;
end
for i=1:1:(sumNx-1)
    S(i,i) = (Sigma_s(i)*Delta(i)+Sigma_s(i+1)*Delta(i+1))/2;
end

%迭代
tic
for i=1:1:N_max
    f_n = F*phi_n(:,i);
    phi_n(:,i+1) = (M-S) \ (f_n/k_n(i));
    f_nn = F*phi_n(:,i+1);
    k_n(i+1) = norm(f_nn)/(norm(f_n)/k_n(i));
    if ((max(abs((f_nn-f_n)./f_nn)))<yip_f) &&
        (abs((k_n(i+1)-k_n(i))/k_n(i+1))<yip_k)

```

```

disp('符合收敛限的特征值结果:')
disp(k_n(i+1))
f_n=f_nn;
%phi_n=phi_nn;
disp('总迭代次数:')
disp(i)
break
end
disp(['第',num2str(i),'次迭代的特征值结果为'])
disp(k_n(i+1))
f_n=f_nn;
%phi_n=phi_nn;
end
disp('迭代用时')
toc
i_max=i;
if i==N_max
    disp('迭代次数达到最大次数未达到给定收敛标准')
end
disp('done')
DPLLOT((sumNx-1):1)=zeros;
DPLLOT(1,1)=Delta(1);
for i=2:1:(sumNx-1)
    DPLLOT(i,1)=DPLLOT(i-1,1)+Delta(i);
end
plot(DPLLOT,phi_n(:,i_max+1),'LineWidth',2,'Color','black');
grid on
xlabel("位置(cm)")
ylabel("中子通量( $n \cdot \text{cm}^{-2} \cdot \text{s}^{-1}$ )")
title("一维直角几何单群特征值问题中子通量分布结果")

gif(DPLLOT,phi_n,i_max) %画 GIF
kk=k_n(1:(i_max+1));
xlswrite('output-keff.xlsx',kk);

```

画 GIF 函数:

```

function [] = gif(DPLLOT,phi_n,i_max)

num = i_max+1;% 参数 a 的个数, 及组成 GIF 图的总图片个数
a_list = linspace(0, i_max, i_max+1);% 设置参数 a 的取值范围
x = DPLLOT;% 设置自变量 x 的范围

pic_num = 1;
for i = 1:num
    a = a_list(i);
    y = phi_n(:,i);
    figure(1);

```

```

set(figure(1), 'Color', 'white');% 设置图片窗口背景颜色为白色
plot(x, y, 'LineWidth', 2, 'Color', [0.0118, 0.0359, 0.4824], 'DisplayName',
'Number of iterations='+string(roundn(a, -2)));
% 绘制 x,y 曲线, 并设置线宽, 曲线颜色, 曲线图例名称

grid on;% 为绘图窗口加上网格
set(gca, 'FontSize', 12, 'FontName', 'Bookman Old Style');% 设置图片中字体的大小, 样式
xlabel('Position(cm)', 'FontSize', 14);% 设置 x 轴标签
xlim([min(x), max(x)]);% 设置 y 轴标签
ylim([0, 1.05*max(phi_n(:, i_max+1))]);% 设置 y 轴显示范围
ylabel('Neutron flux(n·cm-2·s-1)', 'FontSize', 14);% 设置 y 轴标签
legend('FontSize', 14, 'box', 'off');% 为图片加上图例
title('GIF: One-dimensional problem iterative process', 'FontSize', 14);% 增加图片的标题
drawnow;% 立即刷新当前绘图窗口, 这是 matlab 绘图中动态展示的关键

F = getframe(gcf); % 获取当前绘图窗口的图片
Im = frame2im(F); % 返回与电影帧相关的图片数据
[A, map] = rgb2ind(Im, 256); % 将 RGB 图片转化为索引图片
if pic_num == 1
    imwrite(A, map, 'one.gif', 'gif', 'Loopcount', Inf, 'DelayTime', 1);
    % 将第一张图片写入 one.gif 文件中, 并且将 GIF 播放次数设置成无穷, 即保存的 GIF
    图会一直动下去
else
    imwrite(A, map, 'one.gif', 'gif', 'WriteMode', 'append', 'DelayTime', 0.8);
    % 依次将其他的图片写入到 GIF 文件当中
end
pic_num = pic_num + 1;
end
end

```