8.7 多重网格法

胡 茂 彬
http://staff.ustc.edu.cn/~humaobin/
humaobin@ustc.edu.cn

Motivation

- 误差的长波分量：在稠密网格上衰减慢
- 误差的短波分量：在稀疏网格上衰减慢

- 多重网格法：将迭代求解过程放置在不同尺度的网格上进行，促使不同波长误差分量能一致衰减

8.7.1 多重网格法的基本思想

1、迭代误差矢量的衰减过程

2、迭代求解的思路

一维稳态有源扩散问题为例

$$\frac{d^2\phi}{dx^2} = f(x)$$

- 进行中心差分离散：

$$\phi_{j+1} - 2\phi_j + \phi_{j-1} = \Delta x^2 f_j$$

- GS迭代求解：

$$\phi_{j-1}^{(n+1)} - 2\phi_j^{(n+1)} + \phi_{j+1}^{(n)} = \Delta x^2 f_j$$

误差函数的演化

$$\phi_{j-1}^{(n+1)} - 2\phi_j^{(n+1)} + \phi_{j+1}^{(n)} = \Delta x^2 f_j$$

$$\phi_{j-1} = \phi_{j-1}^{(n+1)} + \varepsilon_{j-1}^{(n+1)}$$

$$\phi_{j+1} = \phi_{j+1}^{(n)} + \varepsilon_{j+1}^{(n)}$$

$$\phi_j = \phi_j^{(n+1)} + \varepsilon_j^{(n+1)}$$

- 误差函数演化符合与迭代格式类似的规律：

$$\varepsilon_{j-1}^{(n+1)} - 2\varepsilon_j^{(n+1)} + \varepsilon_{j+1}^{(n)} = 0$$

迭代误差函数分解

放大/衰减因子

- 分量：

$$\varepsilon_j^{(n)} = e^{ikx_j}$$

$$\varepsilon_j^{(n+1)} = ge^{ikx_j}$$

$$\varepsilon_{j+1}^{(n)} = e^{ik(x_j+\Delta x)}$$

$$\varepsilon_{j-1}^{(n+1)} = ge^{ik(x_j-\Delta x)}$$

$$\varepsilon_{j-1}^{(n+1)} - 2\varepsilon_j^{(n+1)} + \varepsilon_{j+1}^{(n)} = 0$$

$$ge^{ik(x_j-\Delta x)} - 2ge^{ikx_j} + e^{ik(x_j+\Delta x)} = 0$$

衰减因子

$$g=\frac{e^{ik\Delta x}}{2-e^{-ik\Delta x}}=\frac{\cos(k\Delta x)+i\sin(k\Delta x)}{2-\cos(k\Delta x)+i\sin(k\Delta x)}=\frac{\cos\theta+i\sin\theta}{2-\cos\theta+i\sin\theta}$$

- 辐角 $\theta=k\Delta x$

$$|g|=\left|\frac{\cos\theta+i\sin\theta}{2-\cos\theta+i\sin\theta}\right|=\frac{1}{\left[\left(2-\cos\theta\right)^2+\sin^2\theta\right]^{1/2}}$$

$$\theta = k\Delta x = 2\pi\Delta x / \lambda$$

- 辐角： $\theta = \pi / 10,\ \pi / 2,\ \pi$
- 衰减因子： $|g| = 0.914,\ 1/\sqrt{5},\ 1/3$
- 5次迭代： $(|g|)^5 \approx 6.4\times10^{-1},\ 1.8\times10^{-2},\ 4.1\times10^{-3}$

$$\theta = k\Delta x = 2\pi\Delta x / \lambda$$

- 高频分量(波长较短)，衰减较快
- 低频分量(波长较长)，衰减很慢

多重网格法

- 先在细网格上迭代，把短波分量衰减掉
- 转到较粗网格上迭代，把次短波分量衰减掉，以此类推
- 再由粗网格依次返回到各级细网格上计算

8.7.2 多重网格法的实施

两种不同的求解方法

1 修正法(Correction Scheme)：在最细网格上求解未知量 X 自身，其它网格上求解未知量的迭代误差量(修正量) ➔ 线性方程组

2 完全逼近方式(Full Approximation Scheme — FAS)：在不同疏密程度的网格上所计算和传递的都是未知量本身 ➔ 非线性方程组

网格间的数据传递

- 限定：从密网格到疏网格的数值传递

$$\boldsymbol{I}_k^{k-1}$$

- 延拓(插值)：从疏网格到密网格的数值传递

$$\boldsymbol{I}_{k-1}^{k}$$

1 修正值法(Correction Scheme)

- 在最密网格上的迭代余量：$\boldsymbol{r}^k = \boldsymbol{b}^k - \boldsymbol{A}^k \overline{\boldsymbol{X}}^k$
- 该网格上的迭代误差：

$$\boldsymbol{\varepsilon}^k = \boldsymbol{X}^k - \overline{\boldsymbol{X}}^k$$

- 准确解满足：

$$0 = \boldsymbol{b}^k - \boldsymbol{A}^k \boldsymbol{X}^k$$

线性方程的情况

- 系数矩阵A恒定不变：

$$\underbrace{\boldsymbol{A}^k \boldsymbol{X}^k}_{=b} - \boldsymbol{A}^k \overline{\boldsymbol{X}}^k = \boldsymbol{A}^k \left(\boldsymbol{X}^k - \overline{\boldsymbol{X}}^k \right) = \boldsymbol{A}^k \boldsymbol{\varepsilon}^k$$

$$\Downarrow$$

$$\boldsymbol{A}^k \boldsymbol{\varepsilon}^k = \boldsymbol{r}^k$$

转入粗网格

$$\boldsymbol{A}^{k}\boldsymbol{\varepsilon}^{k}=\boldsymbol{r}^{k}$$

转入k-1重网格

$$\boldsymbol{A}^{k-1}\boldsymbol{\varepsilon}^{k-1}=\boldsymbol{I}_{k}^{k-1}\boldsymbol{r}^{k}$$

求出余量

$$\boldsymbol{r}^{k-1}=\boldsymbol{I}_{k}^{k-1}\boldsymbol{r}^{k}-\boldsymbol{A}^{k-1}\boldsymbol{\varepsilon}^{k-1}$$

再转入k-2重网格

$$\boldsymbol{A}^{k-2}\boldsymbol{\varepsilon}^{k-2}=\boldsymbol{I}_{k-1}^{k-2}\boldsymbol{r}^{k-1}$$

求解结果传回细网格

粗网格计算误差

$$\overline{\boldsymbol{X}}^{k}_{new} = \overline{\boldsymbol{X}}^{k}_{old} + \boldsymbol{I}^{k}_{k-1}\boldsymbol{\varepsilon}^{k-1}$$

细网格原计算值

延拓！

2 完全逼近方法

- 在 k-1 重网格上仍然求解函数 X 本身
- 方程为：

限定算子

$$A^{k-1}\overline{X}^{k-1}=b^{k-1}+I_k^{k-1}(b^k-A^k\overline{X}^k)$$

细网格方程余量

关键：此两系数

矩阵不同！

改进细网格上的解

粗网格解　　　　细网格解

$$\overline{\boldsymbol{X}}_{new}^{k} = \overline{\boldsymbol{X}}_{old}^{k} + \boldsymbol{I}_{k-1}^{k}\underbrace{(\overline{\boldsymbol{X}}^{k-1} - \boldsymbol{I}_{k}^{k-1}\overline{\boldsymbol{X}}_{old}^{k})}_{\text{粗细网格解之差异}}$$

延拓算子！

限定算子！

确定限定算子(细➔粗)

- 直接引入：

  密网格上的值 ➔ 疏网格对应位置

- 就近平均：

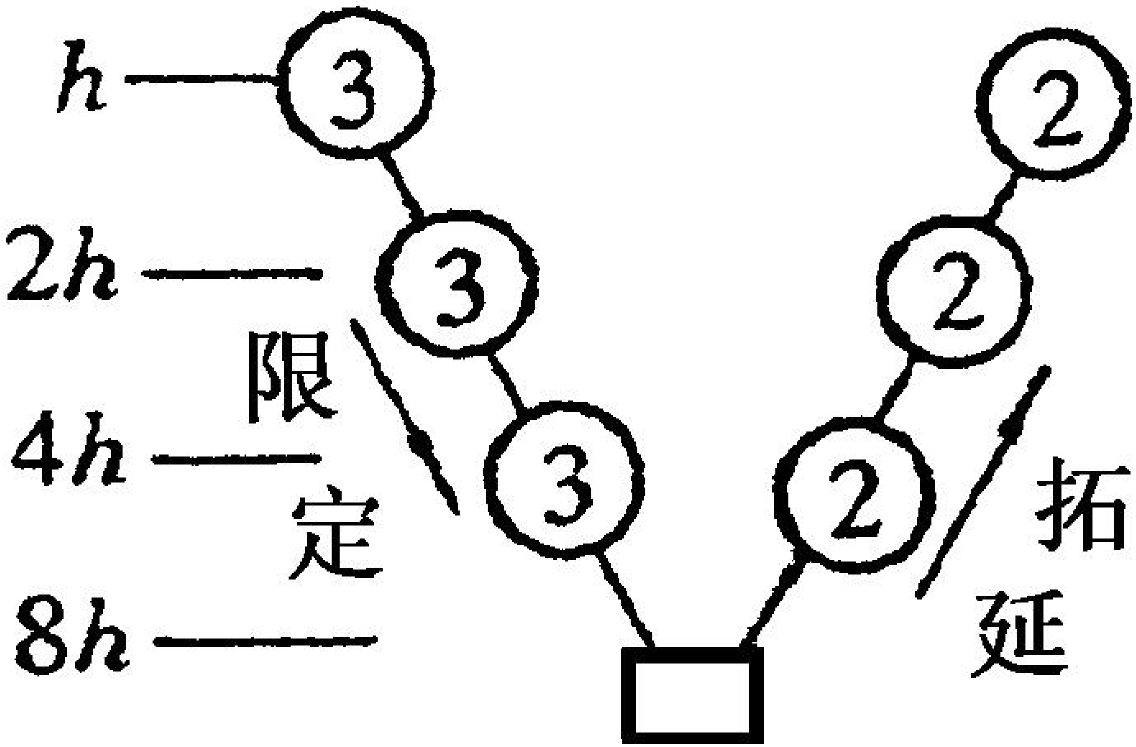
  取附近密网格节点的值作平均

确定延拓算子(粗➔细)

- 双线性插值：

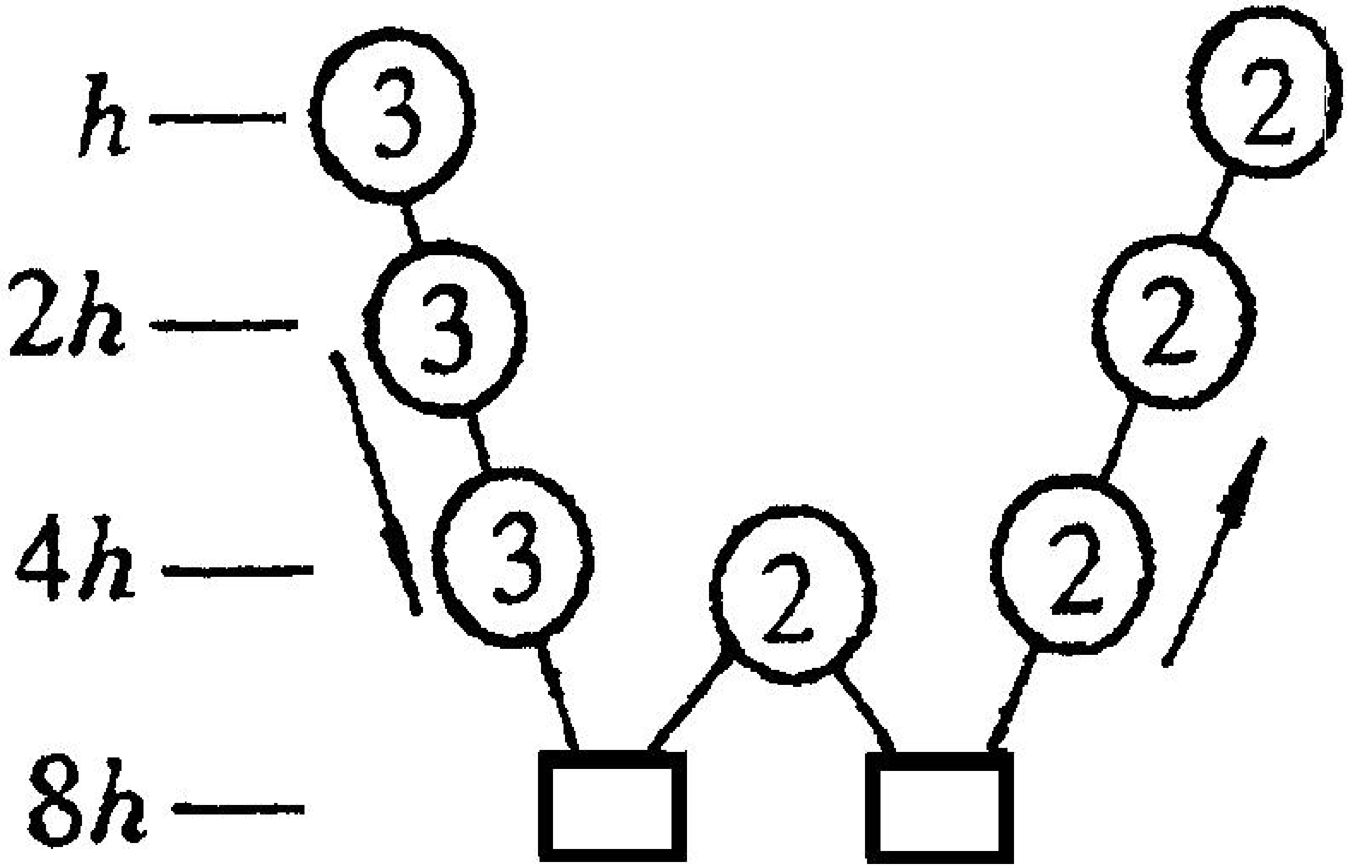
- 双二次插值：

4．不同网格间的迭代循环方式



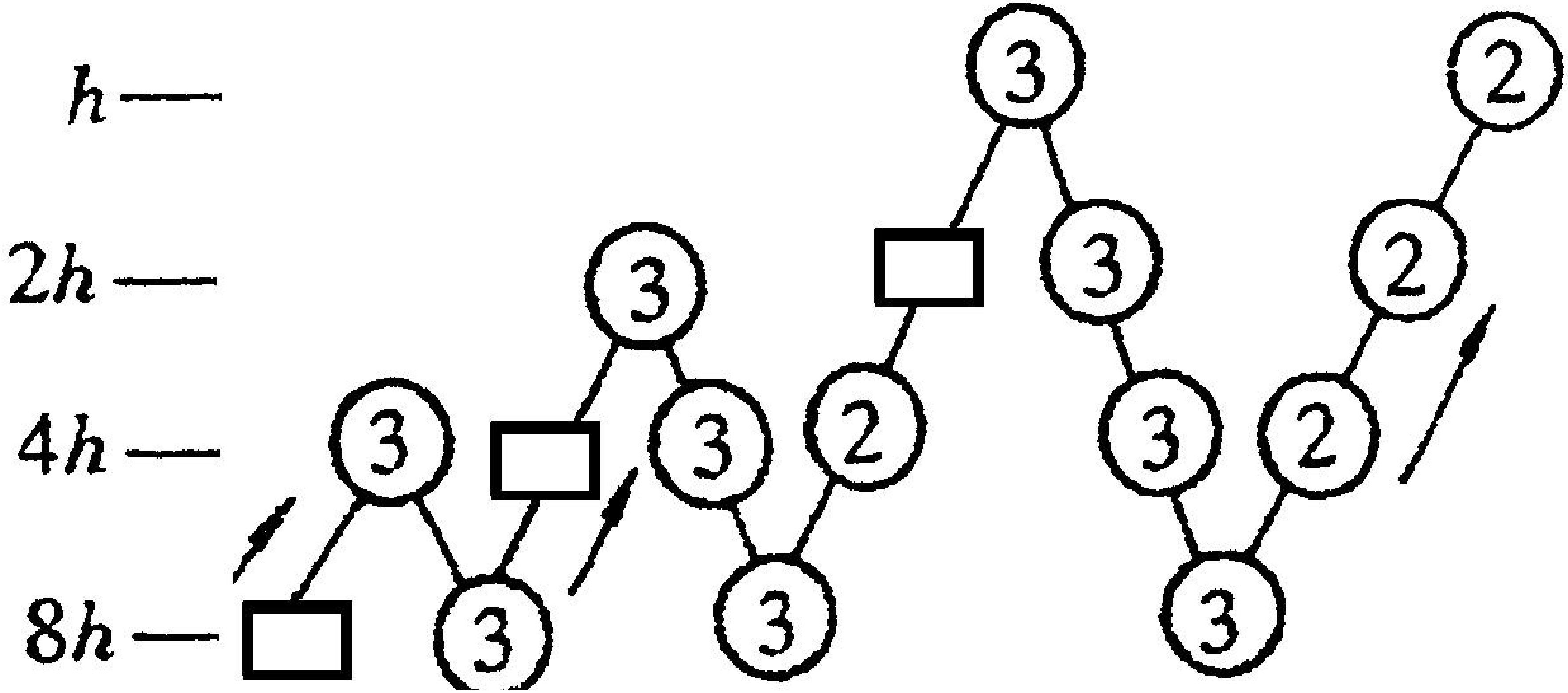
- V型循环：最常用的形式

W型循环



能更有效衰减长波分量误差

Full MultiGrid (FMG) 循环



从最疏网格开始，容易提供一个好的初场