

基于弹簧近似的非结构化网格自适应处理方法

孙旭¹, 张家忠¹, 黄科峰²

(1. 西安交通大学能源与动力工程学院, 710049, 西安; 2. 苏州热工研究院, 215004, 江苏苏州)

摘要: 首先对弹簧近似法进行了说明, 并对节点弹簧法和边弹簧法进行了对比. 在边弹簧法的基础上, 从调节弹簧平衡长度的角度出发, 提出了一种非结构化网格的自适应处理方法, 并结合有限单元法, 给出了网格自适应处理的流程. 最后, 对一维对流扩散方程、二维线性抛物型方程以及欧拉方程进行了求解. 计算结果表明: 文中提出的网格自适应处理方法可以使几个问题的网格节点布置更为合理, 有效地降低数值误差, 提高计算精度, 特别对于算例中所涉及的二维非结构化网格, 自适应化过程中并没有出现网格重叠现象.

关键词: 弹簧近似法; 非结构化网格; 自适应

中图分类号: O241.82 **文献标志码:** A **文章编号:** 0253-987X(2010)09-0104-05

Spring Analogy-Based Adaptive Method for Unstructured Grids

SUN Xu¹, ZHANG Jiazhong¹, HUANG Kefeng²

(1. School of Energy and Power Engineering, Xi'an Jiaotong University, Xi'an 710049, China;

2. Suzhou Nuclear Power Research Institute Co. Ltd., Suzhou, Jiangsu 215004, China)

Abstract: Based on the spring analogy method, an adaptive technique for unstructured grids is proposed. The spring analogy method is introduced and the details about vertex spring and segment spring are discussed. Then, a new method of grid adaptation based on segment spring analogy and the corresponding adaptive procedure are proposed and applied to the problems governed by the one-dimensional convection-diffusion equation, the two-dimensional linear parabolic equation, and the Euler equations. A better distribution of the grid nodes is obtained, and the numerical errors of these problems are significantly decreased. Especially, for the two-dimensional unstructured grids concerned in these problems, overlapping of the grid elements is avoided.

Keywords: spring analogy method; unstructured grid; adaptation

弹簧近似法是一种基于弹簧近似的网格调整方法, 其主要思想是将计算网格等价于弹簧系统(如图1所示), 利用弹性力调整网格节点的位置, 以适应网格边界及内部节点的运动. 该方法具有明确的物理含义, 简单有效, 已经被广泛应用于网格自适应问题及动边界问题.

根据节点间弹簧的设定方式, 弹簧近似法可以分为节点弹簧法和边弹簧法两类^[1]. 在节点弹簧法中, 弹簧的平衡长度为0, 弹簧始终处于拉伸的状态; 在边弹簧法中, 弹簧的平衡长度等于初始时刻弹

簧自身的长度, 除非两端节点发生相对运动, 否则弹簧不会发生拉伸或压缩. 节点弹簧法的思想最早由Gnoffo提出^[2], 后经过数位学者的改进^[3-5], 现在已经发展为一种重要的二维、三维结构化网格自适应处理方法. 然而, 对于非结构化网格, 采用节点弹簧法进行自适应处理时常常导致单元的重叠(如图2所示), 极大地限制了其在非结构化网格中的应用^[6-7]. 与节点弹簧法不同, 边弹簧法主要用于非结构化网格. 该方法首先由Batina提出^[8], 用来生成三角形非结构化动网格, 以求解扭转振动翼型的绕

流问题.随后,其他学者对其进行了改进^[1,9],分别提出了扭转弹簧法和半扭转弹簧法,解决了网格重叠问题.近年来,扭转弹簧法和半扭转弹簧法均被应用到了三维问题中^[10-11].同时,针对三维非结构化网格,还发展了 ball-vertex^[12]和 OST^[13]等边弹簧近似方法.

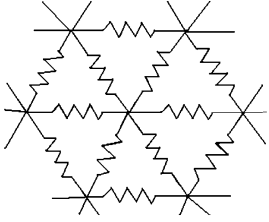


图1 计算网格的弹簧近似

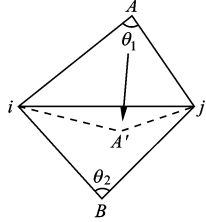


图2 网格的重叠

与节点弹簧相比,边弹簧更加适用于非结构化网格的调节,目前不仅已开发出多种方法来解决单元重叠问题,而且还形成了多种三维非结构化网格的调整方法.为了利用边弹簧法的优势,本文在此提出一种基于边弹簧近似的非结构化网格自适应处理方法.

1 网格自适应处理方法

在网格的自适应化处理中,有两个环节至关重要,即误差指针的设定和网格调整.误差指针是已有数值结果的误差表示方式,可以采用局部截断误差、关键变量的一阶或二阶导数等.网格调整则是根据误差指针对初始网格进行的调整,目前主要有3种方式:对局部网格进一步剖分(h型方法);对已有网格的节点进行重新分布(r型方法);增加部分单元的插值节点,提高单元上近似函数的次数(p型方法).本文所提出的基于边弹簧近似的非结构化网格自适应方法就是一种r型方法,此类方法可以在网格节点和单元数不变、函数场求解计算量不变的前提下,最大程度地提高计算精度.下面将结合有限单元法,给出本文方法的具体实施过程.

1.1 误差指针

根据 Diaz 等人^[14]的描述,在有限单元法中,单元 K 上的误差估计可以表示为

$$\|u - u_h\|_{m,K} \leq C h_K^{k+1-m} |u|_{k+1,K}, \quad 0 \leq m \leq k \quad (1)$$

式中: u 、 u_h 分别表示精确解和数值解; m 为非负整数; k 表示有限元插值函数的最高次数; C 为任意常数; h_K 为单元 K 的特征长度; $\|\cdot\|_{m,\Omega}$ 和 $|\cdot|_{m,\Omega}$ 为 Sobolev 空间中的范数,对于二维问题分别定义为

$$\|v\|_{m,\Omega} = \left\{ \int_{\Omega} \sum_{\alpha_1+\alpha_2 \leq m} \left[\frac{\partial^{(\alpha_1+\alpha_2)} v}{\partial x_1^{\alpha_1} \partial x_2^{\alpha_2}} \right]^2 d\Omega \right\}^{1/2} \quad (2)$$

$$|v|_{m,\Omega} = \left\{ \int_{\Omega} \sum_{\alpha_1+\alpha_2 = m} \left[\frac{\partial^{(\alpha_1+\alpha_2)} v}{\partial x_1^{\alpha_1} \partial x_2^{\alpha_2}} \right]^2 d\Omega \right\}^{1/2} \quad (3)$$

其中 α_1 和 α_2 均为非负整数,当 $\alpha_2=0$ 时,可以得到一维情况下的表达式.

考虑到式(1)中不等式右端项与精确解 u 相关,不能利用已有的数值计算结果进行表示,Diaz 等人对其进行了处理,提出了一种简单实用的误差指针 f_K ,即

$$\begin{aligned} 1D: \quad f_K &= h_K^\beta |u_h|_{k+1,K}^2, \quad \beta = 2k \\ 2D: \quad f_K &= A_K^\beta |u_h|_{k+1,K}^2, \quad \beta = k \end{aligned} \quad (4)$$

式中: A_K 表示单元 K 的面积.对于本文所采用的线性插值基函数而言,式(4)可以进一步表示为

$$\begin{aligned} 1D: \quad f_K &= h_K^2 \int_{\Omega_K} \left(\frac{\partial^2 u_h}{\partial x^2} \right)^2 dx \\ 2D: \quad f_K &= A_K \int_{\Omega_K} \left[\left(\frac{\partial^2 u_h}{\partial x_1^2} \right)^2 + \right. \\ &\quad \left. 2 \left(\frac{\partial^2 u_h}{\partial x_1 \partial x_2} \right)^2 + \left(\frac{\partial^2 u_h}{\partial x_2^2} \right)^2 \right] d\Omega \end{aligned} \quad (5)$$

采用线性插值基函数时,式(5)中任意网格节点 I 处待求变量的二阶导数是无法直接表示的,本文采用文献[5]中的近似方式进行计算.

进一步,为了衡量不同网格单元上误差指针的相对大小,定义新的参数

$$r_K = f_K / \bar{f}_K \quad (6)$$

式中: r_K 称为单元 K 上的相对误差指针; $\bar{f}_K$ 为所有网格单元上误差指针的平均值.

1.2 自适应流程

本文所进行的网格自适应处理是通过使误差指针在所有单元上均匀分布来实现的^[6].式(5)表明,单元 K 上的误差指针是与待求函数的二阶导数以及单元的尺寸相关的,因此可以通过调整单元的尺寸实现单元上的误差控制,达到使误差指针均匀分布的目的.采用边弹簧法时,单元尺寸的调整可以通过增加或减少单元各边弹簧的平衡长度来实现.因此,基于边弹簧近似的网格自适应处理可以按照如下步骤进行:

- (1)采用有限单元法,在初始网格上对已有问题进行计算,得到初步的数值结果;
- (2)对已有数值结果进行误差估计,计算单元上的误差指针 f_K 和 r_K ,找出“坏单元”(相对误差指针 r_K 超出限定范围 r_0 的单元);
- (3)采用边弹簧近似方法,得到与初始网格相对

应的弹簧系统;

(4)减少“坏单元”上边弹簧的平衡长度;

(5)在弹性力的作用下调整网格节点的位置,得到新的网格;

(6)通过插值,根据初始网格上的数值结果得到新网格上待求量的分布;

(7)采用步骤(2)的方法,对新网格上的函数场进行误差估计,如果仍存在“坏单元”,则重复步骤(3)~(7);

(8)采用有限单元法,在所得网格上重新对已有问题进行计算。

上述步骤(3)~(5)中的弹簧近似法与传统的边弹簧方法有所不同,这里作进一步说明。在步骤(3)中,根据传统的边弹簧近似方法,计算网格被等效为由相互连接的弹簧所组成的系统,此时所有弹簧均没有被拉伸或压缩,弹簧系统处于平衡状态。在步骤(4)中,部分弹簧的平衡长度被减小,根据虎克定律,此时任意节点 i 上所受到的弹性合力可以表示为

$$\mathbf{F}_i = \sum_{j=1}^{N_i} \alpha_{ij} [\mathbf{x}_j - \mathbf{x}_i - \beta(\mathbf{x}_j - \mathbf{x}_i)] \quad (7)$$

式中: N_i 表示与节点 i 相连的节点数; $\mathbf{x}_i$ 为节点 i 处的坐标; β 为控制参数,对于“坏单元”上的弹簧, $0 < \beta < 1.0$,对于其余的弹簧, $\beta = 1.0$;弹簧刚度 α_{ij} 采用如下定义^[11]

$$\left. \begin{aligned} \text{1D: } \alpha_{ij} &= \frac{1}{l_{ij}} \\ \text{2D: } \alpha_{ij} &= \frac{1}{l_{ij}^2} + k \left(\frac{1}{\sin^2 \theta_1} + \frac{1}{\sin^2 \theta_2} \right) \end{aligned} \right\} \quad (8)$$

其中 l_{ij} 表示节点 i, j 间的距离, k 为控制参数,这里取 $k = 1.0$, θ_1 和 θ_2 为边 $i-j$ 所对的三角形顶角,如图2所示。由于部分弹簧上的受力不再为0,弹簧系统的受力平衡将遭到破坏,故在弹性力的作用下,部分网格节点将发生移动,直到弹簧系统达到新的平衡状态。根据受力平衡准则,由式(7)可得

$$\mathbf{x}_i^{k+1} = \frac{\sum_{j=1}^{N_i} \alpha_{ij}^k [\mathbf{x}_j^k - \beta(\mathbf{x}_j - \mathbf{x}_i)]}{\sum_{j=1}^{N_i} \alpha_{ij}^k} \quad (9)$$

考虑到 α_{ij}^k 与网格节点的坐标相关,故式(9)是一个非线性迭代过程。为简化起见,本文在迭代过程中始终采用根据初始网格计算所得的刚度 α_{ij}^0 。通过式(9)进行迭代计算后,便可获得新的平衡位置,即新的网格。

在上述自适应处理过程中, r_0 和 β 是两个非常

重要的控制参数,其中 r_0 是自适应迭代过程的收敛指标。在理想情况下,应取 $r_0 = 1.0$,即要求误差指针在所有单元上均匀分布。但是,在实际情况下这个条件往往不能达到。本文中 r_0 是参考求解区域的最大相对误差指针 $r_{K, \max}$ 设定的,网格自适应处理的效果便是将 $r_{K, \max}$ 降低到允许的范围之内。参数 β 用来控制每次迭代中“坏单元”上弹簧平衡长度减小的程度,它直接决定着该自适应过程的收敛速度。

2 计算实例及分析

为了检验上述网格自适应处理方法的可行性,下面对一维对流扩散方程、二维线性抛物型方程以及超声速圆柱绕流问题进行计算。

2.1 一维对流扩散方程

定常一维对流扩散方程具有如下形式

$$\left. \begin{aligned} U \frac{du}{dx} - \epsilon \frac{d^2 u}{dx^2} &= 0, \quad x \in (0, 1) \\ u(0) &= 0, \quad u(1) = 1 \end{aligned} \right\} \quad (10)$$

式中: U, ϵ 分别为对流和扩散参数。该问题具有解析解: $u(x) = \frac{(1 - e^{Ux/\epsilon})}{(1 - e^{U/\epsilon})}$ 。

采用标准 Galerkin 有限单元法分别对 $U = 1.0, \epsilon = 0.0625$ 和 $U = 1.0, \epsilon = 0.02$ 两种情况进行计算,结果见图3、图4。在这两种情况下,初始网格均是对 $(0, 1)$ 区间10等分得到的,自适应网格是在 $r_0 = 1.05, \beta = 0.99$ 的情况下生成的。从图中可以看出,与初始网格相比,自适应网格在 u 变化剧烈处分配了更多的节点,数值计算结果与解析解也更加符合。特别是对于 $U = 1.0, \epsilon = 0.02$ 的情况,在进行自适应处理之后,初始网格上的数值振荡消失了,由此显示出计算网格节点布置的重要性。

表1给出了不同范数下的计算误差。从表中可以看出,当 $U = 1.0, \epsilon = 0.0625$ 时,自适应网格对应的计算误差的 ∞ 范数和 L^2 范数均比初始网格时降低了1个数量级,分别为原来的6.01%和12.02%,而在 $U = 1.0, \epsilon = 0.02$ 时则降低了2个数量级。

表1 一维对流扩散问题的计算误差

条件	$\ u - u_h\ _{\infty}$		$\ u - u_h\ _{L^2}$	
	均分网格	自适应网格	均分网格	自适应网格
$U = 1.0$ $\epsilon = 0.0625$	9.08×10^{-2}	5.46×10^{-3}	3.02×10^{-2}	3.63×10^{-3}
$U = 1.0$ $\epsilon = 0.02$	4.36×10^{-1}	6.64×10^{-3}	1.52×10^{-1}	3.51×10^{-3}

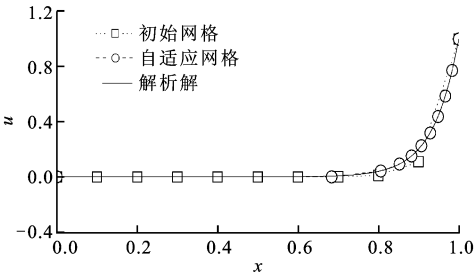


图 3 $U=1.0, \epsilon=0.0625$ 时的计算结果

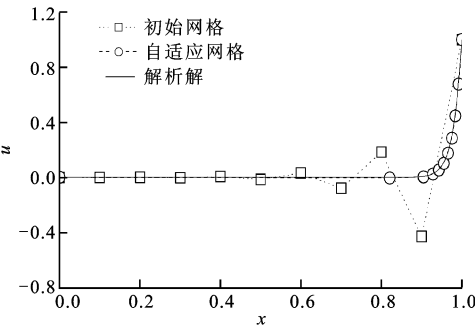


图 4 $U=1.0, \epsilon=0.02$ 时的计算结果

2.2 二维线性抛物型方程

计算如下二维线性抛物型方程

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + f(t, x, y) \tag{11}$$

求解域为 $(0,1) \times (0,1)$ 的二维矩形区域, $f(t, x, y)$ 和边界条件均根据该问题的解析解式(12)来确定

$$u(t, x, y) = \tanh[15(x - 0.5)] \tanh[15(y - 0.5)] \tag{12}$$

时间上采用欧拉法,空间上采用标准 Galerkin 有限单元法,对方程(11)进行计算. 图 5 所示为初始计算网格,由 1 800 个三角形单元和 961 个节点组成. 由初始网格计算的最大相对误差指针 $r_{K, \max} = 6.01$. 图 6 所示为 $\beta=0.8, r_0=2.0$ 时得到的自适应网格. 表 2 给出了网格自适应处理前后所对应的计算误差. 对网格进行自适应处理后,计算误差的 ∞ 范数和 L^2 范数分别降低到了原值的 39.16% 和

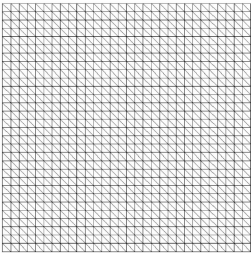


图 5 二维线性抛物型方程初始网格

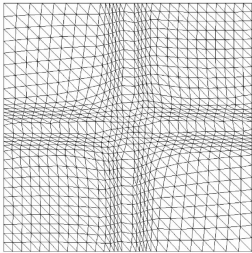


图 6 二维线性抛物型方程自适应网格

45.67%.

表 2 二维线性抛物型问题计算误差

网格	$\ u - u_h\ _{\infty}$	$\ u - u_h\ _{L^2}$
初始网格	2.03×10^{-2}	2.54×10^{-3}
自适应网格	7.95×10^{-3}	1.16×10^{-3}

2.3 超声速圆柱绕流

应用本文所提出的网格自适应化方法,对超声速圆柱绕流问题进行了计算. 来流马赫数 $Ma = 3.0$,不考虑流体的黏性,忽略体积力和外部导入的热量,此绕流问题的控制方程(欧拉方程)具有如下形式

$$\left. \begin{aligned} \frac{\partial \rho}{\partial t} + \frac{\partial U_i}{\partial x_i} &= 0 \\ \frac{\partial U_i}{\partial t} + \frac{\partial}{\partial x_j} (u_j U_i) &= - \frac{\partial p}{\partial x_i} \\ \frac{\partial (\rho E)}{\partial t} + \frac{\partial}{\partial x_j} (u_j \rho E) &= - \frac{\partial}{\partial x_j} (u_j p) \end{aligned} \right\} \tag{13}$$

式中: ρ 为流体密度; p 为压强; u_j 为流动速度; E 为能量; $U_i = \rho u_i$.

采用 CBS 有限单元法^[15] 对此问题进行求解,初始网格由 Delaunay 方法生成,如图 7 所示,共含有 4 212 个三角形单元,2 191 个节点. 图 8 给出了初始网格上计算得到的流场马赫数分布,从中可以看出:在圆柱体上游,流体的马赫数变化剧烈,产生了激波. 根据初始网格算得的最大相对误差指针 $r_{K, \max} = 52.44$. 取 $\beta=0.8, r_0=5.0$ 进行自适应处理,所得自适应网格如图 9 所示,可见在圆柱上游流动参数变化最为强烈、相对误差指针最大的激波面上,弹簧收缩最为明显,形成了更为紧凑的网格. 图10

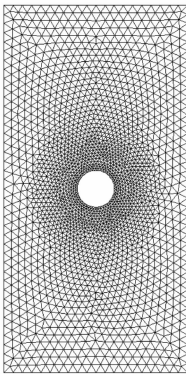


图 7 超声速圆柱绕流问题的初始计算网格

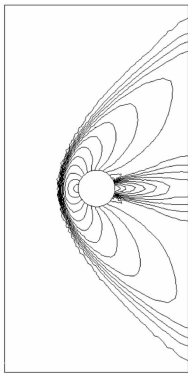


图 8 超声速圆柱绕流问题的马赫数分布(初始网格)

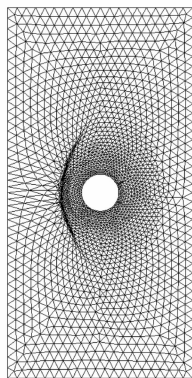


图9 超声速圆柱绕流问题的自适应网格

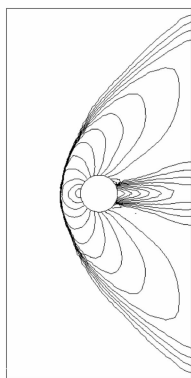


图10 超声速圆柱绕流问题的马赫数分布(自适应网格)

为自适应网格上计算所得流场的马赫数分布,与图8中的计算结果相比,圆柱上游的马赫数等值线更为集中,激波的宽度变小,使激波得到了更好的描述。

3 结 论

本文提出了一种基于边弹簧近似的 r 型非结构化网格自适应处理方法,并采用该方法对一维对流扩散方程、二维线性抛物型方程以及欧拉方程进行了求解。计算结果表明,采用本文方法进行网格自适应处理,可以使初始网格的节点布置更为合理,有效降低数值误差,提高计算精度。特别是对于算例中所涉及的二维非结构化网格,自适应化过程中并没有出现网格重叠现象,进一步说明了边弹簧方法对于非结构化网格的优越性。

需要说明的是,在几个算例中,最大相对误差指针所能减小到的程度(即 r_0)均与参数 β 的选取有关,对于参数 r_0 与 β 的内在联系及其选取方式,仍需做进一步的研究。另外,本文的算例只是初步检验了所提出方法的可行性,为了验证基于边弹簧近似的自适应方法对于二维、三维非结构化网格的普遍适用性,需要对更多的问题进行计算。

参考文献:

- [1] BLOM F J. Considerations on the spring analogy [J]. *Int J Numer Meth Fluids*, 2000, 32(6): 647-648.
- [2] GNOFFO P A. A finite-volume adaptive grid algorithm applied to planetary entry flowfields [J]. *AIAA J*, 1983, 21(9): 1249-1253
- [3] NAKAHASHI K, DEIWERT G S. A practical adaptive grid method for complex fluid flow problems [C] // *Proceedings of the Ninth International Conference on Numerical Methods in Fluid Dynamics*. Berlin,

Germany: Springer-Verlag, 1985: 422-426.

- [4] NAKAHASHI K, DEIVERT G S. Three-dimensional adaptive grid method [J]. *AIAA J*, 1986, 24(6): 948-954.
- [5] AIT-ALI-YAHIA D, HABASHI W G, TAM A, et al. A directionally-adaptive finite element method for high-speed flows, AIAA Paper 96-2553 [R]. Reston, VA, USA: AIAA, 1996.
- [6] LHNER R, MORGAN K, ZIENKIEWICZ O C. Adaptive grid refinement for the compressible Euler equation [M] // *Accuracy Estimates and Adaptive Refinements in Finite Element Computation*. New York, USA: John Wiley & Sons Ltd., 1986: 281-297.
- [7] PALMERIO B. A two dimensional FEM adaptive moving node method for steady Euler flow simulations [J]. *Comput Methods Appl Mech Engrg*, 1988, 71(3): 315-340.
- [8] BATINA J T. Unsteady Euler airfoil solutions using unstructured dynamic meshes [J]. *AIAA J*, 1990, 28(8): 1381-1388.
- [9] FARHAT C, DEGAND C, KOOBUS B, et al. Torsional springs for two-dimensional dynamic unstructured fluid meshes [J]. *Comput Methods Appl Mech Engrg*, 1998, 163(1/4): 231-245.
- [10] DEGAND C, FARHAT C. A three-dimensional torsional spring analogy method for unstructured dynamic meshes [J]. *Comput Struct*, 2002, 80(3/4): 305-316.
- [11] ZENG D, ETHIER C R. A semi-torsional spring analogy model for updating unstructured meshes in 3D moving domains [J]. *Finite Elem Anal Des*, 2005, 41(11/12): 1118-1139.
- [12] BOTTASSO C L, DETOMI D, SERRA R. The ball-vertex method: a new simple spring analogy method for unstructured dynamic meshes [J]. *Comput Methods Appl Mech Engrg*, 2005, 194(39/41): 4244-4264.
- [13] MARKOU G A, MOUROUTIS Z S, CHAMPIS D C, et al. The ortho-semi-torsional (OST) spring analogy method for 3D mesh moving boundary problems [J]. *Comput Methods Appl Mech Engrg*, 2007, 196(4/6): 747-765.
- [14] DIAZ A R, KIKUCHI N, TAYLOR J E. A method of grid optimization for finite element methods [J]. *Comput Methods Appl Mech Engrg*, 1983, 41(1): 29-45.
- [15] ZIENKIEWICZ O C, TAYLOR R L. The finite element method, vol 3: fluid dynamics [M]. 5th ed. Oxford, UK: Elsevier Butterworth-Heinemann, 2005.

(编辑 葛赵青)