

燃烧仿真教程：湍流燃烧模型与网格生成技术

1 燃烧仿真基础

1.1 燃烧物理学原理

燃烧是一种复杂的物理化学过程，涉及到燃料与氧化剂的化学反应、热量的产生与传递、以及流体动力学现象。在燃烧物理学中，我们关注的是燃烧的热力学和动力学特性，包括燃烧的速率、火焰的传播、以及燃烧过程中能量的转换。燃烧速率受多种因素影响，如燃料的性质、氧化剂的浓度、温度、压力以及混合物的湍流程度。

1.1.1 示例：燃烧速率计算

假设我们有一个简单的燃烧反应： $H_2 + \frac{1}{2}O_2 \rightarrow H_2O$ 。燃烧速率可以表示为：

$$r = k[H_2][O_2]$$

其中， r 是燃烧速率， k 是反应速率常数， $[H_2]$ 和 $[O_2]$ 分别是氢气和氧气的浓度。

在实际计算中，我们可以通过 Arrhenius 方程来计算反应速率常数 k ：

$$k = A \exp\left(-\frac{E_a}{RT}\right)$$

其中， A 是频率因子， E_a 是活化能， R 是理想气体常数， T 是温度。

1.2 湍流燃烧模型概述

湍流燃烧模型是用于描述在湍流环境中燃烧过程的数学模型。湍流的存在极大地增加了燃烧过程的复杂性，因为它影响了燃料与氧化剂的混合、热量的传递以及化学反应的速率。湍流燃烧模型通常分为两大类：均相燃烧模型和非均相燃烧模型。均相燃烧模型适用于气体燃料的燃烧，而非均相燃烧模型则适用于固体或液体燃料的燃烧。

1.2.1 示例：湍流燃烧模型中的混合长度理论

混合长度理论是湍流燃烧模型中常用的一种方法，它假设湍流混合是由一系列随机的混合长度引起的。混合长度 l 可以通过以下公式计算：

$$l = C_\mu k^{3/2} / \epsilon$$

其中， C_μ 是模型常数， k 是湍流动能， ϵ 是湍流动能的耗散率。

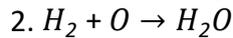
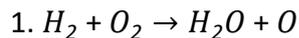
在计算燃烧速率时，混合长度理论可以用来估计燃料与氧化剂的混合程度，从而影响化学反应的速率。

1.3 化学反应动力学基础

化学反应动力学研究化学反应的速率以及反应机理。在燃烧仿真中，化学反应动力学是关键因素，因为它决定了燃烧的速率和产物的生成。化学反应动力学模型通常包括反应方程式、反应速率常数以及反应路径。

1.3.1 示例：化学反应动力学模型的构建

假设我们有一个简单的化学反应系统，包含以下反应：



我们可以构建一个化学反应动力学模型来描述这个系统。首先，我们需要定义反应速率常数 k_1 和 k_2 ，然后根据反应方程式计算反应速率。

$$r_1 = k_1[H_2][O_2]$$

$$r_2 = k_2[H_2][O]$$

其中， r_1 和 r_2 分别是两个反应的速率， $[H_2]$ 、 $[O_2]$ 和 $[O]$ 分别是氢气、氧气和氧原子的浓度。

在实际仿真中，我们通常使用数值方法来求解这些反应速率，例如使用 Runge-Kutta 方法来求解反应速率随时间的变化。

1.3.2 代码示例：使用 Runge-Kutta 方法求解化学反应速率

```
import numpy as np
from scipy.integrate import solve_ivp

# 定义反应速率常数
k1 = 1.0e6 # 反应1 的速率常数
k2 = 1.0e5 # 反应2 的速率常数

# 定义反应速率函数
def reaction_rates(t, y):
    H2, O2, O = y
    r1 = k1 * H2 * O2
    r2 = k2 * H2 * O
    return [-r1 - r2, -r1, r1 - r2]

# 定义初始条件
y0 = [1.0, 1.0, 0.0] # 初始浓度

# 定义时间范围
t_span = (0, 1.0)

# 使用 Runge-Kutta 方法求解
sol = solve_ivp(reaction_rates, t_span, y0, method='RK45', t_eval=np.linspace(0, 1, 100))
```

```
# 输出结果
print(sol.t) # 时间点
print(sol.y) # 各物质的浓度随时间的变化
```

这段代码使用了 Python 的 `scipy.integrate.solve_ivp` 函数，通过 Runge-Kutta 方法求解了上述化学反应系统的反应速率随时间的变化。`y0` 定义了初始浓度，`t_span` 定义了时间范围，`reaction_rates` 函数定义了反应速率的计算方法。

通过这个例子，我们可以看到化学反应动力学模型在燃烧仿真中的应用，以及如何使用数值方法来求解这些模型。在实际应用中，化学反应系统可能包含更多的反应和物质，因此模型的构建和求解会更加复杂。

2 湍流-化学反应相互作用

2.1 湍流对化学反应的影响

湍流环境下的燃烧过程远比层流燃烧复杂。在湍流中，流体的不规则运动导致了燃料和氧化剂的快速混合，这直接影响了化学反应的速率和模式。湍流的尺度和强度决定了反应区域的分布，从而影响燃烧效率和污染物的生成。

2.1.1 原理

湍流通过增加混合速率，使得化学反应在更短的时间内完成。此外，湍流的旋涡结构可以形成所谓的“旋涡燃烧”，在这些旋涡中，化学反应速率可能因局部条件而显著增加。湍流还通过改变火焰结构，如形成火焰皱褶或火焰尖端，来影响化学反应。

2.1.2 内容

在湍流燃烧模型中，通常使用雷诺平均纳维-斯托克斯（RANS）方程来描述流体的平均运动，同时使用概率密度函数（PDF）或混合分数（Mixture Fraction）方法来处理化学反应的不确定性。例如，PDF 方法考虑了湍流中不同化学组分的概率分布，从而更准确地预测化学反应速率。

2.2 化学反应对湍流的影响

化学反应不仅受到湍流的影响，它反过来也会影响湍流的特性。在燃烧过程中，化学反应释放的能量可以改变流体的温度和密度，从而影响湍流的结构和强度。这种相互作用在高速燃烧和爆炸中尤为显著。

2.2.1 原理

化学反应释放的热量可以导致局部温度升高，这会降低流体的密度，从而产生浮力效应，增强湍流。同时，化学反应也可能改变流体的粘度和热导率，

进一步影响湍流的动态。

2.2.2 内容

在模拟中，化学反应的热效应通过能量方程的源项来体现。例如，在 RANS 模型中，化学反应的放热率被加入到能量方程中，以反映化学反应对流体温度的影响。此外，化学反应速率的变化也会通过湍流模型中的湍动能和湍流耗散率方程来影响湍流的强度。

2.3 相互作用的数学描述

湍流和化学反应的相互作用可以通过一系列耦合的偏微分方程来描述，这些方程包括连续性方程、动量方程、能量方程以及化学物种的输运方程。

2.3.1 原理

这些方程组描述了流体的运动、能量的传递以及化学物种的扩散和反应。在湍流燃烧模型中，这些方程通常被雷诺平均或大涡模拟 (LES) 处理，以减少计算的复杂性。

2.3.2 内容

以 RANS 模型为例，其基本方程组如下：

1. 连续性方程：

$$\frac{\partial \bar{\rho}}{\partial t} + \nabla \cdot (\bar{\rho} \bar{\mathbf{u}}) = 0$$

2. 动量方程：

$$\frac{\partial (\bar{\rho} \bar{\mathbf{u}})}{\partial t} + \nabla \cdot (\bar{\rho} \bar{\mathbf{u}} \otimes \bar{\mathbf{u}}) = -\nabla \bar{p} + \nabla \cdot (\bar{\boldsymbol{\tau}})$$

3. 能量方程：

$$\frac{\partial (\bar{\rho} \bar{E})}{\partial t} + \nabla \cdot (\bar{\rho} \bar{\mathbf{u}} \bar{E}) = \nabla \cdot (\bar{\mathbf{q}}) + \bar{\rho} \bar{S}$$

4. 化学物种输运方程：

$$\frac{\partial (\bar{\rho} \bar{Y}_i)}{\partial t} + \nabla \cdot (\bar{\rho} \bar{\mathbf{u}} \bar{Y}_i) = \nabla \cdot (\bar{\mathbf{J}}_i) + \bar{\rho} \bar{R}_i$$

其中， $\bar{\rho}$ 是流体的平均密度， $\bar{\mathbf{u}}$ 是平均速度， \bar{p} 是平均压力， $\bar{\boldsymbol{\tau}}$ 是雷诺应力张量， \bar{E} 是平均总能量， $\bar{\mathbf{q}}$ 是平均热流， \bar{S} 是化学反应的放热率， \bar{Y}_i 是化学物种*i*的平均质量分数， $\bar{\mathbf{J}}_i$ 是化学物种*i*的扩散通量， \bar{R}_i 是化学物种*i*的生成率。

2.3.3 示例代码

以下是一个使用 OpenFOAM 进行湍流燃烧模拟的简化代码示例，展示了如何设置湍流模型和化学反应模型：

```

// 燃烧仿真设置
#include "fvCFD.H"

int main(int argc, char *argv[])
{
    #include "postProcess.H"

    // 读取网格
    #include "createMesh.H"

    // 定义湍流模型
    #include "createTurbulence.H"

    // 定义化学反应模型
    #include "createChemistryModel.H"

    // 定义流体属性
    #include "createThermo.H"

    // 定义状态方程
    #include "createState.H"

    // 定义湍流和化学反应的相互作用
    #include "createReaction.H"

    // 初始化时间步
    #include "readTimeControls.H"

    // 初始化场变量
    #include "initContinuityErrs.H"

    // 主循环
    while (runTime.loop())
    {
        #include "solveMomentum.H"
        #include "solveEnergy.H"
        #include "solveSpecies.H"
        #include "solveTurbulence.H"
        #include "solveReaction.H"
        #include "output.H"
    }

    return 0;
}

```

以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。如要下载或阅读全文，请访问：<https://d.book118.com/537115152013006161>