

强度计算.数值计算方法：有限元法（FEM）：有限元软件操作与实践

1 绪论

1.1 有限元法的历史与发展

有限元法（Finite Element Method, FEM）起源于 20 世纪 40 年代末，最初由工程师们在解决结构工程问题时提出。1943 年，R. Courant 在解决弹性问题时首次使用了类似于有限元法的离散化技术。然而，直到 1956 年，O.C. Zienkiewicz 和 Y.K. Cheung 在《工程计算》杂志上发表了一篇关于有限元法的文章，才真正标志着有限元法的诞生。此后，随着计算机技术的飞速发展，有限元法逐渐成为解决复杂工程问题的重要工具。

1.2 FEM 在工程中的应用

有限元法广泛应用于各种工程领域，包括但不限于：

- **结构工程**：分析桥梁、建筑、飞机等结构的强度、刚度和稳定性。
- **机械工程**：预测机械部件在不同载荷下的变形和应力分布。
- **土木工程**：评估地基的承载力和土壤的力学行为。
- **热力学和流体力学**：模拟热传导、流体流动和传热过程。
- **电磁学**：分析电磁场的分布和电磁设备的性能。

1.2.1 示例：使用 Python 进行简单梁的有限元分析

```
# 导入必要的库
import numpy as np

# 定义材料属性和截面属性
E = 200e9 # 弹性模量，单位：Pa
nu = 0.3 # 泊松比
I = 1e-4 # 截面惯性矩，单位：m^4

# 定义梁的长度和节点位置
L = 1.0 # 梁的总长度，单位：m
n_nodes = 5 # 节点数量
nodes = np.linspace(0, L, n_nodes) # 节点位置

# 定义单元和节点的连接关系
elements = np.array([[0, 1], [1, 2], [2, 3], [3, 4]]) # 单元连接
```

```

# 定义边界条件
boundary_conditions = np.array([0, 0, 1, 1, 0]) # 第一个和最后一个节点固定

# 定义载荷
loads = np.array([0, -1000, 0, 0, 0]) # 第二个节点受到向下 1000N 的力

# 计算刚度矩阵
def stiffness_matrix(E, l, L):
    k = np.array([[12, 6*L, -12, 6*L],
                  [6*L, 4*L*L, -6*L, 2*L*L],
                  [-12, -6*L, 12, -6*L],
                  [6*L, 2*L*L, -6*L, 4*L*L]]) * E/(L**3)
    return k

# 组装整体刚度矩阵
K = np.zeros((n_nodes*2, n_nodes*2))
for e in elements:
    k = stiffness_matrix(E, l, nodes[e[1]] - nodes[e[0]])
    K[2*e[0]:2*e[0]+2, 2*e[0]:2*e[0]+2] += k[:2, :2]
    K[2*e[0]:2*e[0]+2, 2*e[1]:2*e[1]+2] += k[:2, 2:]
    K[2*e[1]:2*e[1]+2, 2*e[0]:2*e[0]+2] += k[2:, :2]
    K[2*e[1]:2*e[1]+2, 2*e[1]:2*e[1]+2] += k[2:, 2:]

# 应用边界条件
K = K[np.ix_(np.where(boundary_conditions == 0)[0], np.where(boundary_conditions == 0)[0])]
F = loads[np.where(boundary_conditions == 0)[0]]

# 解线性方程组
displacements = np.linalg.solve(K, F)

# 输出位移
print("节点位移: ", displacements)

```

此代码示例展示了如何使用 Python 和 Numpy 库对一个简单的梁进行有限元分析。通过定义材料属性、节点位置、单元连接、边界条件和载荷，我们组装了整体刚度矩阵，并求解了线性方程组以得到节点位移。

1.3 有限元软件的种类与选择

有限元软件种类繁多，选择合适的软件取决于具体的应用场景和用户的需求。常见的有限元软件包括：

- **ANSYS**: 广泛应用于航空航天、汽车、电子和能源行业。
- **ABAQUS**: 在复合材料、非线性分析和疲劳分析方面表现突出。
- **NASTRAN**: 最初为 NASA 开发，特别适用于航空航天结构的分析。
- **COMSOL Multiphysics**: 适用于多物理场耦合分析，如热力学、流体力学和电磁学。
- **SolidWorks Simulation**: 集成在 SolidWorks CAD 软件中，适合进行初步的结构分

析。

选择有限元软件时，应考虑以下因素：- **功能需求**：软件是否支持所需分析类型（如线性、非线性、多物理场）。- **易用性**：软件的用户界面是否友好，是否容易学习和使用。- **成本**：软件的购买和维护成本是否在预算范围内。- **技术支持**：软件提供商是否提供良好的技术支持和培训服务。- **兼容性**：软件是否能与现有的 CAD 或 CAE 系统无缝集成。

在实际应用中，工程师和研究人员通常会根据项目需求和自身经验选择最合适的有限元软件。

2 有限元法基础

2.1 基本概念与原理

有限元法（Finite Element Method, FEM）是一种数值计算方法，用于求解复杂的工程问题，如结构分析、热传导、流体动力学等。其核心思想是将连续的结构或物理场离散成有限数量的单元，每个单元用简单的函数（如多项式）来近似表示其内部的物理量变化，从而将偏微分方程转化为代数方程组，通过求解这些方程组来获得整个结构或场的解。

2.1.1 离散化过程

1. **几何离散化**：将连续的结构或物理场划分为有限个单元，如三角形、四边形、六面体等。
2. **函数逼近**：在每个单元内，用插值函数（如线性、二次函数）来逼近物理量（如位移、温度、压力）的变化。
3. **方程组建立**：通过变分原理或加权残数法，将偏微分方程转化为单元的刚度矩阵方程，然后组合所有单元的方程，形成整个系统的方程组。
4. **求解方程组**：使用数值方法（如直接法、迭代法）求解系统方程组，得到物理量的数值解。

2.1.2 示例：一维杆件的有限元分析

假设有一根长度为 1m 的均匀杆件，两端固定，受到均匀分布的轴向力作用。杆件的弹性模量为 200GPa，截面积为 0.01m²。我们使用有限元法来计算杆件的位移。

```
# 导入必要的库
import numpy as np

# 定义材料属性和几何参数
E = 200e9 # 弹性模量, 单位: Pa
A = 0.01 # 截面积, 单位: m^2
L = 1.0 # 杆件长度, 单位: m
```

```

F = 1000 # 轴向力, 单位: N

# 定义单元数和节点数
n_elements = 10
n_nodes = n_elements + 1

# 定义节点坐标
nodes = np.linspace(0, L, n_nodes)

# 定义单元连接
elements = np.zeros((n_elements, 2), dtype=int)
for i in range(n_elements):
    elements[i] = [i, i+1]

# 计算单元刚度矩阵
k = (E * A) / L * np.array([[1, -1], [-1, 1]])

# 组合系统刚度矩阵
K = np.zeros((n_nodes, n_nodes))
for i in range(n_elements):
    K[elements[i][0]:elements[i][1]+1, elements[i][0]:elements[i][1]+1] += k

# 应用边界条件
K[0, :] = 0
K[:, 0] = 0
K[0, 0] = 1
K[-1, :] = 0
K[:, -1] = 0
K[-1, -1] = 1

# 定义力向量
F_vec = np.zeros(n_nodes)
F_vec[-1] = F

# 求解位移向量
U = np.linalg.solve(K, F_vec)

# 输出位移结果
print("节点位移: ", U)

```

2.2 弹性力学基础

弹性力学是研究弹性体在外力作用下变形和应力分布的学科。在有限元分析中，弹性力学的理论是基础，用于建立单元的刚度矩阵和应力应变关系。

2.2.1 应力应变关系

在弹性力学中，应力（ σ ）和应变（ ϵ ）之间的关系由胡克定律描述，即：

$$\sigma = E * \epsilon$$

其中， E 是弹性模量，表示材料抵抗弹性变形的能力。

2.2.2 应力和应变的计算

在有限元分析中，应力和应变可以通过单元的位移解来计算。例如，在一维杆件中，应变 ϵ 可以通过相邻节点的位移差除以单元长度来计算，而应力 σ 则可以通过应变 ϵ 和弹性模量 E 来计算。

2.3 变分原理与加权残数法

变分原理和加权残数法是有限元法中用于建立单元方程的两种主要方法。

2.3.1 变分原理

变分原理是基于能量守恒的原理，通过最小化系统总势能来建立方程。在结构分析中，总势能由内部能量（如弹性能量）和外部能量（如外力做功）组成。

2.3.2 加权残数法

加权残数法是通过最小化残差（即偏微分方程的左端减去右端）的加权平均来建立方程。权重函数的选择取决于问题的类型，常见的有伽辽金法（Galerkin Method）和最小二乘法（Least Squares Method）。

2.3.3 示例：使用伽辽金法建立单元方程

伽辽金法是一种常用的加权残数法，它使用单元的 shape 函数作为权重函数。在下面的例子中，我们使用伽辽金法来建立一维杆件的单元方程。

```
# 定义伽辽金法的权重函数
def shape_function(x, xi):
    return (1 - xi) / 2, (1 + xi) / 2

# 定义残差函数
def residual_function(u, x, E, A, F):
    du_dx = np.gradient(u, x)
    return E * A * du_dx - F

# 定义单元的伽辽金方程
def galerkin_equation(xi, x, E, A, F):
    N1, N2 = shape_function(x, xi)
```

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/497166102052006160>