

# 基于 SiPESC 平台的传热问题有限元分析系统

杨东生<sup>1,2</sup>, 张盛<sup>1,2</sup>, 李云鹏<sup>\*1,2</sup>, 陈飙松<sup>1,2</sup>, 张洪武<sup>1,2</sup>

(1. 大连理工大学 工业装备结构分析国家重点实验室, 辽宁 大连 116024;  
2. 大连理工大学 运载工程与力学学部 工程力学系, 辽宁 大连 116024)

**摘要:** 基于新一代集成化计算软件平台和大规模工程数据库, 研发传热问题面向对象有限元分析系统 SiPESC.THERMAL. 该系统采用插件及扩展的管理机制, 可实现计算功能的动态替换和组装, 具有一定的通用性、开放性和扩展性. 热单元设计采用 Factory、Builder 设计模式, 可方便添加不同单元, 已完成添加实际传热问题的常用单元. 热表面荷载处理采用荷载面单元处理方式, 已实现热流、对流等热边界条件的处理. 数值算例验证了程序系统的有效性, 及解决大规模工程实际问题的能力.

**关键词:** SiPESC; 传热问题; 有限元分析; 软件设计模式  
**中图分类号:** TB115.1 **文献标志码:** A

## 0 引言

工程中存在许多传热问题, 如耐火材料的热强度、极端热环境下结构热防护、激光脉冲相关的高热流及瞬态冲击、航空航天结构热变形热防护、电子电路的热疲劳、微电子封装以及制造业中焊接带来的残余应力、工业生产中的锻造成型等<sup>[1]</sup>. 近几十年来, 有限元法的发展为通过数值模拟解决这些工程问题提供了有效途径<sup>[2]</sup>. 然而, 随着模型复杂性增加, 对有限元分析程序的基本需求就是其开放性、扩展性及大规模分析能力, 而现有的有限元程序多基于结构化语言, 难以扩展和继续维护, 因而很难满足这样的需求, 使其进一步发展遇到瓶颈.

面向对象的思想克服了结构化程序语言的不足, 并为有限元程序研发提供了新的手段. 面向对象的有限元编程最早由 Fenves<sup>[3]</sup> 提出, Forde 等<sup>[4]</sup> 给出了详细的有限元程序的面向对象执行过程, 文献<sup>[5-7]</sup> 也使用面向对象的编程来改善有限元程序的灵活性. 但面向对象有限元编程的研究多集中于定义模型类, 如单元、节点、边界条件、材

料, 而很少注意和分析相关的计算任务类的定义. 在模型类和分析代码之间, 没有明显的界限, 这易使一个有限元程序变得庞大而难以维护. 另一方面, CAE 技术是国家装备制造业的核心竞争能力之一, 然而主流 CAE 软件产品都来自国外.

面向工程与科学计算的集成化软件平台 SiPESC<sup>[8-9]</sup> 就是在这样的背景和需求下提出来的. 本文在 SiPESC 的基础上, 尝试研发传热问题的面向对象有限元分析程序系统 SiPESC.THERMAL, 通过插件及扩展的管理机制, 采用面向对象的 C++ 程序设计语言、Factory 及 Builder 等软件设计模式, 以使系统具有良好的开放性、扩展性、模块化等特点.

本文重点介绍热单元计算模块的设计模式、单元数据类及单元计算实现类, 以及热荷载处理模块的设计模式、荷载数据类及荷载处理实现类, 以期为解决工程实际问题提供参考.

## 1 传热问题控制方程

Fourier 定律是传统导热理论的基础, 其推导

收稿日期: 2012-12-19; 修回日期: 2013-09-20.

基金项目: 国家自然科学基金资助项目(51021140004, 11072050); “九七三”国家重点基础研究发展计划资助项目(2010CB832704); “八六三”国家高技术研究发展计划资助项目(2012AA050901); 中央高校基本科研业务费专项资金资助项目(DUT12ZD206).

作者简介: 杨东生(1985-), 男, 博士生, E-mail: yangds@mail.dlut.edu.cn; 李云鹏\*(1970-), 男, 博士, 教授, E-mail: lyp@dlut.edu.cn.

获得 Fourier 传热控制方程<sup>[1]</sup>

$$\rho c \dot{T} - \text{div}(\mathbf{k} \cdot \text{grad } T) = Q \quad (1)$$

其中  $\mathbf{k}$  为传热系数张量,  $Q$  为热源强度,  $T$  为温度场,  $\rho$  为密度,  $c$  为比热容. 本程序考虑三类热边界条件, 即指定温度、热流和对流. 忽略时间效应, 方程(1) 退化为稳态传热控制方程, 即

$$-\text{div}(\mathbf{k} \cdot \text{grad } T) = Q \quad (2)$$

由于 Fourier 理论会导致热量传播速度无限大, 为了对瞬态热传导问题做出更合理的描述, 需要对 Fourier 定律进行修正, 本文采用 CV 波模型, 其推导获得非 Fourier 传热控制方程<sup>[1]</sup>

$$\rho c (\tau \dot{T} + T) - \text{div}(\mathbf{k} \cdot \text{grad } T) = Q \quad (3)$$

其中  $\tau$  为热流松弛时间, 其物理意义为温度场的重新建立滞后于热扰动改变的时间. 该方程为有阻尼的波动方程, 属于双曲型偏微分方程.

采用 Galerkin 离散得到稳态传热、Fourier 及非 Fourier 瞬态传热有限元半离散方程分别为

$$\mathbf{K}\boldsymbol{\phi} = \mathbf{R}, \mathbf{C}\dot{\boldsymbol{\phi}} + \mathbf{K}\boldsymbol{\phi} = \mathbf{R}, \mathbf{M}\ddot{\boldsymbol{\phi}} + \mathbf{C}\dot{\boldsymbol{\phi}} + \mathbf{K}\boldsymbol{\phi} = \mathbf{R} \quad (4)$$

其中  $\boldsymbol{\phi}$  为节点温度向量, 对应的整体矩阵分别由单元级别矩阵集合而成, 即

$$\mathbf{K} = \sum (\mathbf{k}_e + \mathbf{h}_e), \mathbf{C} = \sum \mathbf{c}_e \quad (5)$$

$$\mathbf{M} = \sum \mathbf{m}_e, \mathbf{R} = \sum (\mathbf{r}_q + \mathbf{r}_r + \mathbf{r}_h)$$

单元热传导刚度、对流刚度、比热容和热松弛矩阵以及对流、热流和热源荷载矢量分别由以下积分得到:

$$\mathbf{k}_e = \int_{\Omega_e} \mathbf{B}^T \mathbf{k} \mathbf{B} d\Omega_e \quad (6a)$$

$$\mathbf{h}_e = \int_{\Gamma_h} \mathbf{N}^T \mathbf{N} h d\Gamma_h \quad (6b)$$

$$\mathbf{c}_e = \int_{\Omega_e} \mathbf{N}^T \mathbf{N} \rho c d\Omega_e \quad (6c)$$

$$\mathbf{m}_e = \int_{\Omega_e} \mathbf{N}^T \mathbf{N} \rho c \tau d\Omega_e \quad (6d)$$

$$\mathbf{r}_h = \int_{\Gamma_h} \mathbf{N}^T h T_\infty d\Gamma_h \quad (6e)$$

$$\mathbf{r}_q = \int_{\Gamma_q} \mathbf{N}^T \hat{q} d\Gamma_q \quad (6f)$$

$$\mathbf{r}_Q = \int_{\Omega_e} \mathbf{N}^T Q d\Omega_e \quad (6g)$$

其中  $\mathbf{N}$  和  $\mathbf{B}$  分别为形函数及其梯度矩阵.

对瞬态传热方程进行时间离散, 对 Fourier

和非 Fourier 传热方程分别采用  $\theta$  积分<sup>[2]</sup> 和 Newmark 直接积分法<sup>[2]</sup> 进行计算.

## 2 程序框架

本程序系统是在 SiPESC<sup>[8]</sup> 及其开放式结构有限元分析子系统<sup>[9]</sup> 的基础上完成的. SiPESC 有大规模数值模型的管理能力、多物理场多相多尺度的集成计算能力以及系统开放性的特点.

开放式结构有限元分析子系统的数据存储基于大规模工程数据库系统, 可满足各种数据对象的设计和高效存取; 数值计算部分采用模块化设计, 提供了节点排序、约束处理、局部坐标转换、单元刚度计算、荷载计算、求解器等通用计算模块.

本程序研发在此基础上展开. 传热问题的标量场特点, 以及边界条件的多样性, 使其具有和结构有限元不同的特点. 因此从通用的标量场问题求解角度考虑, 必须设计适合标量场问题求解的计算流程, 如标量场自由度处理、标量场约束处理等, 这些流程不但适合传热问题的分析, 也适合其他标量场如渗流场等问题的处理.

考虑到传热问题的本构、荷载的特殊性, 本文增加了传热问题特有的计算流程, 如单元矩阵的计算包括单元热传导阵、单元比热容阵, 以及后处理流程单元温度梯度、单元热流及节点热流计算, 程序框架如图 1 所示.

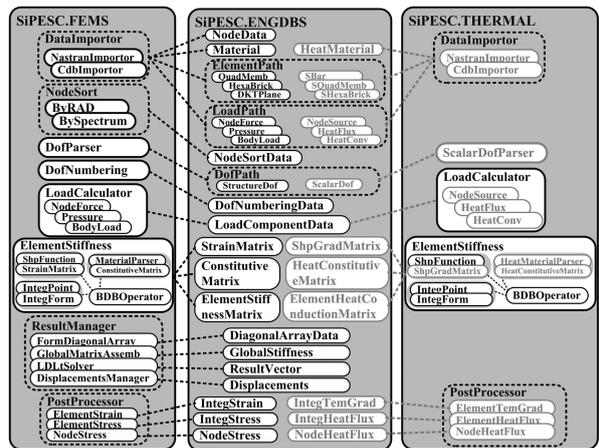


图 1 分析系统程序框架

Fig. 1 Program framework of the analysis system

各个流程之间采用数据定义标准接口, 耦合性大大降低, 各流程分别调用不同的插件完成各

自的功能,对不同问题可以通过不同的流程动态组装得到不同的分析类型. 稳态热分析的流程为节点排序、标量场自由度处理、标量场约束处理、单元热传导阵计算、荷载计算及求解. 由于定义了标准的数据接口,不同的分析类型可以共用相同的流程,体现了模块化的灵活性.

### 3 热单元和热荷载计算模块

热单元和热荷载计算是本程序系统的重要计算流程,本文将介绍其实现过程. 为了提高代码的可重用性、提高软件质量、缩短研发时间,在热单元和热荷载计算模块采用了软件设计模式中的 Factory 模式和 Builder 模式<sup>[10]</sup>,使本程序系统具有良好的开放性、扩展性、模块化等特点.

#### 3.1 Factory 模式

在有限元计算时,对不同单元需生成不同的计算模块,传统方法往往用 new 生成. 以荷载计算类为例,假设要满足节点热荷载及单元热流荷载的计算,传统的面向对象方法首先定义一个荷载计算基类,由此类分别派生出节点热荷载计算类及单元热流荷载计算类,其类图如图 2(a)所示. 则对节点热荷载的调用方式如下:

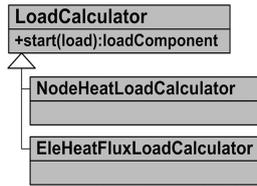
```
LoadCalculator calculator =
    new NodeHeatLoadCalculator;
loadComponent = calculator.start(load);
```

对单元热流荷载的调用方式如下:

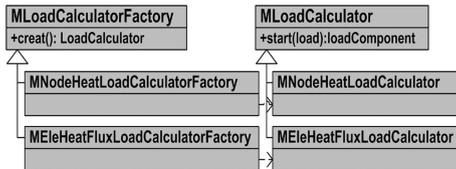
```
LoadCalculator calculator =
    new EleHeatFluxLoadCalculator;
loadComponent = calculator.start(load);
```

可以看出,对不同种类的荷载,调用方式不同,这样添加新的荷载类型时需要修改已有的调用代码,降低了程序的可维护性和可扩展性.

本文采用软件设计模式中的 Factory 模式,对象工厂首先生成父工厂类,定义通过单元类型变量创建对象的公共接口,而其子类则负责生成具体的类实例. 这样可以不用关心具体对象的实现,并可通过工厂提供的优先级方法动态替换单元计算模块,提高了系统的灵活性.



(a) 传统面向对象方法



(b) Factory 模式

图 2 荷载计算类图

Fig. 2 Class diagram for load calculation

在具体实施过程中,对不同的单元类型(或荷载、积分、形函数)定义不同的 type,在计算时由此 type 找到对应的 Factory,然后由 Factory 创建出对应的计算模块.

仍以荷载计算类为例,本文方法首先定义荷载计算基类与荷载计算工厂基类,由此派生出不同种类的荷载计算类及荷载计算工厂类. 其类图如图 2(b)所示. 对应的调用方式如下:

```
type = load.getType();
MLoadCalculatorFactory factory =
    factoryManager.getFactory(type);
MLoadCalculator calculator = factory.create();
loadComponent = calculator.start(load);
```

即首先由荷载类型找到对应的荷载计算工厂,然后创建出对应的计算类,统一完成计算. 由于调用方式统一,添加新荷载不需要修改已有的调用代码,方便维护和扩展.

#### 3.2 Builder 模式

在单元矩阵的计算过程中需要多次数值积分,如式(6)所示. 数值积分算法较稳定,但是单元类型和材料模型的多样性使稳定的数值积分算法不能重用. 本文采用软件设计中的 Builder 模式,采用封装机制来隔离出复杂对象的各个部分的变化,保持系统中稳定构建算法不随其子对象的改变而改变,使同样的构造过程可创建不同的表示. 以单元刚度阵(单元热传导阵)积分计算为例,在

单元计算前,由 MElementStiffBuilder 对象将所需对象生成,传递给单元计算类 MElementStiffCalculator,类图如图 3 所示。

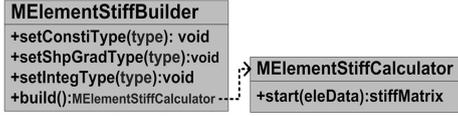


图 3 Builder 模式的单元刚度阵计算

Fig. 3 Builder pattern for element stiffness matrix calculation

这样单元计算对象就只负责计算任务,而不必判断单元的性质,对不同类型的单元积分,只需定义新的构造器,例如三维六面体单元热传导阵积分设置为

```
builder.setConstiType("heatbrickeleconstidparser");
builder.setShpGradType(HexaBrickShpGradMatrix);
builder.setIntegType(Gau3D2IntegForm);
```

三维四面体单元热传导阵积分设置为

```
builder.setConstiType("heatbrickeleconstidparser");
builder.setShpGradType(TetraBrickShpGradMatrix);
builder.setIntegType(TetraIntegForm);
```

对于不同的单元,可以重用计算模块,体现了软件设计的可重用性。

### 3.3 热单元计算模块

工程中传热问题的多样性会导致有限元分析单元类型多种多样,一些通用有限元软件甚至有上百种单元.这些单元通常还不能满足用户的需求,随着理论和算法的发展,会出现更多的单元类型.所以对于一个开放式有限元计算平台,一个重要功能是可以提供开放的接口,方便添加新的单元。

为了提高系统的灵活性,本文数据存储采用工程数据库系统,可满足各种数据对象的设计和高效存取.计算部分采用模块化设计,各个流程间采用数据定义标准接口,通过工程数据库系统完成各自功能.本文单元计算模块采用软件设计模式中的 Factory 和 Builder 模式,建立通用单元计算模块。

### (1) 单元数据类

对于不同类型的单元,首先需要定义单元数据类.单元数据类由工程数据库系统的数据基类 MDataObject 派生而来,MDataObject 类只定义数据对象 ID 的存取方法.为了保证各单元数据使用过程中的通用性,定义单元数据基类 MElementData,该基类定义的主要接口为获取单元类型、存取单元指定序号的节点 ID、获取单元节点个数、存取单元性质 ID、获取节点刚度自由度贡献描述数组.具体某种单元的数据类可由上述基类派生,以标量场单元数据为例,单元基类接口定义以及数据类之间的派生关系如图 4 所示。

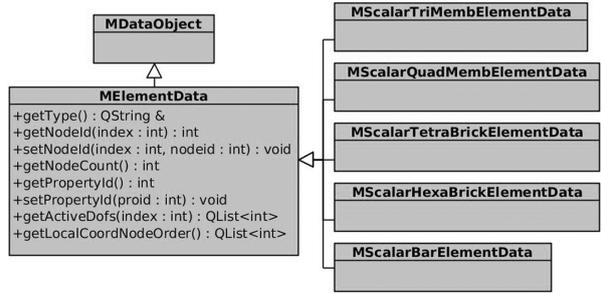


图 4 数据基类、单元数据基类及单元数据类类图  
Fig. 4 Class diagram for data base class, element data base class and element data classes

对某种单元,单元类型、单元节点个数、自由度贡献描述数组已确定,只需从数据库存取指定位置节点 ID 和单元性质 ID,这大大简化了单元数据类的设计.每个单元数据类均有唯一的类型 type 作为标识,在各计算流程中,若计算过程根据各单元不同,则通过该 type 和工厂模式可找到对应的计算模块。

### (2) 单元计算实现类

在 SiPESC 结构有限元<sup>[8]</sup>系统中,已定义了应变阵、本构解析、材料解析等基类和接口,本文的形函数梯度阵、热传导本构解析、传热材料解析均由这些基类派生而来。

SiPESC 的插件和扩展机制定义了扩展基类 MExtension,所有扩展均由此派生.为了更好地管理和分配不同层次的计算任务,在扩展基类的基础上,定义了三种基类,即任务基类 MTask、任务执行基类 MTaskExecutor 和任务管理基类

MTaskManager. 其派生关系类图如图 5 所示.

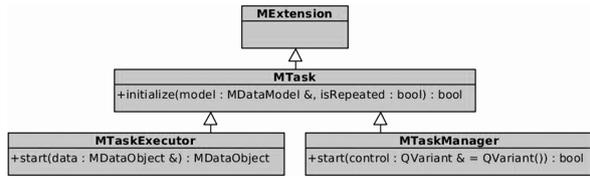


图 5 扩展基类、任务基类、任务执行基类、任务管理基类类图

Fig. 5 Class diagram for extension base class, task base class, task executor base class and task manager base class

其中 MTask 类定义的虚方法 initialize 完成数据库的初始化. MTaskExecutor 类定义的虚方法 start 完成单一任务(比如单一单元、单一荷载)的计算,其输入输出参数均为以 MDataObject 为基类的数据类. MTaskManager 类定义的虚方法 start 打开数据库,把任务分配给 MTaskExecutor,得到计算结果,并存入数据库.

单元计算模块采用 Factory 模式和 Builder 模式结合,其中 Factory 基类 MExtensionFactory 由扩展基类 MExtension 派生而来,Builder 基类 MElementBuilder 由 MTask 派生而来,图 6 给出了各个实现类之间的关系.

其中 Factory 基类 MExtensionFactory 定义的成员函数有 getType 和 createExtension,其中前者返回单元类型 type,供扩展工厂管理器查找,后者调用 MHeatEleStiffBuilder 的 3 个成员函数,完成对本构、形函数及其梯度、积分类型的设置之后,返回 MHeatEleStiffBuilder 对象.而 MHeatEleStiffBuilder 分别由 Factory 模式创建出对应的计算模块,通过调用 MHeatEleStiffCalculator 的 3 个成员函数,完成计算模块的设置,通过成员函数 build 返回设置好的 MHeatEleStiffCalculator.最后由 MHeatEleStiffCalculator 继承基类 MTaskExecutor 的 start 方法完成对式(6a)的积分计算.

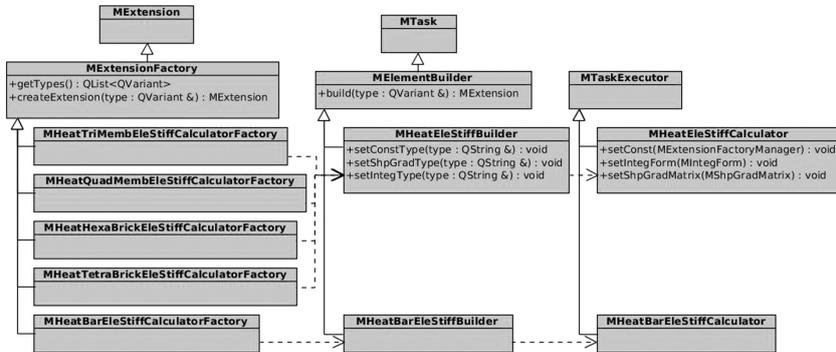


图 6 传热单元刚度阵计算类图

Fig. 6 Class diagram for thermal element stiffness matrix calculation

对传热杆单元,其单元刚度阵由下式解析给出:

$$\mathbf{K}_e = \frac{Ak}{L} \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix} \quad (7)$$

故不需要采用上述积分方法计算,但为了各个单元的统一,仍采用同样的流程,只是不用设置本构、形函数及其梯度、积分类型等信息,如图 6 所示.

值得注意的是,所有单元模块的计算对象都由 Factory 模式生成,这样各计算对象都可以灵

活地替换和组装.这样对于不同的单元,只需完成对应的本构、形函数及其梯度、积分类型等模块的定义,即可重用已有的单元刚度阵计算模块.

### 3.4 热荷载处理模块

在传热问题的有限元分析中,荷载形式多种多样.根据荷载类型不同,可以分为热源荷载、热流荷载、对流荷载等.根据施加荷载几何形式不同,可分为体荷载、面荷载、点荷载.根据施加有限元对象不同,可以分为节点荷载和单元表面荷载.

在不同荷载的处理上采用 Factory 模式,即

每种荷载均含有一个荷载类型 *type*, 根据该 *type* 找到对应的荷载计算器进行相应的处理。

对于和单元无关的荷载, 如节点热源, 处理较简单。对于和单元相关的荷载, 多样的单元类型和荷载形式使开发者的工作变得很烦琐。比如开发者若想增加一种单元, 则在完成单元计算的同时, 需要处理此种单元表面施加的不同荷载形式。若新增加一种和单元相关的荷载, 则需要对所有支持此种荷载的单元开发出对应的计算模块。这样使开发的耦合度上升, 且不利于代码的重用性。本文程序系统采用荷载面单元的处理方式, 有效地降低了两者之间的耦合度, 使开发的工作量降低。

### (1) 荷载数据类

以表面荷载(热流或对流)为例, 不同几何形式表示的表面荷载可分为面、线(需定义厚度)、点(需定义面积)3 种形式。其中每种几何形式可以有不同的类型, 比如面形式也可以是三节点三角形或六节点三角形、八节点四边形等高阶形式。

对不同的几何形式, 需定义几何表面单元数据类作为一种新的单元类型, 但是不需要定义自由度信息, 为了方便统一处理, 定义几何表面单元数据基类 *MGeoFaceData*。对应的荷载类只需定义荷载类型、荷载值和该荷载施加几何对象的 ID。

在表面荷载的处理上, 采用 *Factory* 的设计模式, 即根据几何表面单元的 *type* 找到对应的几何表面单元荷载计算器, 完成计算并返回各个节点的荷载值。

### (2) 荷载处理实现类

对于面荷载, 通常由各个节点描述一个荷载密度曲面, 即定义各个节点的荷载密度值, 曲面上的其他点由一定的插值方式得到。此种形式荷载由数值积分转化为节点荷载。

对于点形式的几何表面单元, 其节点荷载的转化可由解析形式完成, 即

$$\mathbf{F} = \mathbf{f} \cdot A \quad (8)$$

其中  $\mathbf{f}$  为表面荷载密度,  $A$  为荷载施加面积,  $\mathbf{F}$  为节点荷载向量。二节点线几何表面单元的积分也有解析解, 其荷载项和对流刚度项分别为

$$\mathbf{F} = \frac{L \cdot t}{6} \begin{pmatrix} 2f_1 + f_2 \\ f_1 + 2f_2 \end{pmatrix};$$

$$\mathbf{H} = \frac{L \cdot t}{12} \begin{pmatrix} 3f_1 + f_2 & f_1 + f_2 \\ f_1 + f_2 & f_1 + 3f_2 \end{pmatrix} \quad (9)$$

其中  $L$  为线长度,  $t$  为厚度,  $f_1$  和  $f_2$  分别为两个节点上荷载密度大小, 两点之间的荷载密度由这两个节点的值插值得到。

在程序实现上采用 *Factory* 模式, 类图如图 7 所示。由节点荷载、对流荷载、热流荷载的 *type* 找到对应的 *Factory*, 由 *Factory* 创建出对应的计算模块。其中各对应模块均有共同的基类 *MLoadCalculator*, 此基类继承于 *MTaskExecutor* 任务执行基类。

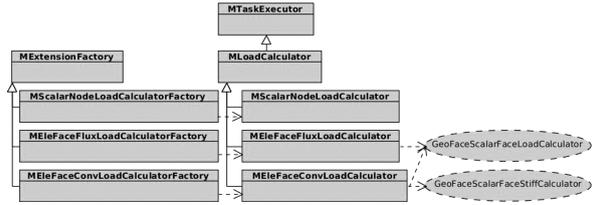


图 7 节点荷载、对流荷载和热流荷载类图  
Fig. 7 Class diagram for node load, convection load and heat flux load

其中节点荷载计算与单元无关, 直接由 *MScalarNodeLoadCalculator* 完成, 而对流荷载和热流荷载需要对单元进行面积分, 所以分别调用对应的积分计算模块完成。其中式(6e)~(6g)的荷载面积分由 *MScalarFaceLoadCalculator* 模块完成, 式(6b)中的对流刚度积分由 *MScalarFaceStiffCalculator* 模块完成。这两个模块的积分与单元刚度模块的积分类似, 采用 *Factory* 模式与 *Builder* 模式相结合的设计模式, 如图 8 所示。

本文考虑了三节点三角形、四节点四边形, 及二节点线、点等不同形式的表面单元, 对于需要数值积分的面单元, 由 *Builder* 模式、*Factory* 模式中给定的 *type* 参数创建出插值形函数对象、形函数梯度对象、积分格式对象, 传递给 *Calculator*, 统一完成计算。对于有解析解的点、线等几何形式的表面单元, 仍然采用统一的流程, 由对应的 *Calculator* 直接计算完成。

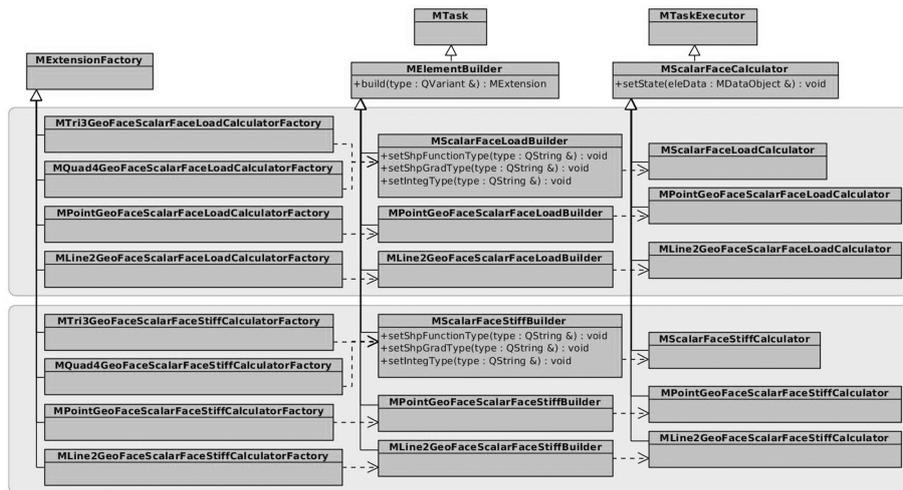


图 8 热荷载积分和热对流刚度积分类图

Fig. 8 Class diagram for heat load integration and heat convection stiffness integration

### 4 数值算例

本节给出了一维、二维、三维常用单元在指定温度、对流、热流等不同边界条件下的算例验证。

#### 4.1 一维散热片问题

矩形铝制散热片<sup>[11]</sup>, 表面温度为 100 °C, 环境温度 20 °C, 铝热导率为 168 W/(m·K), 对流传热系数为 30 W/(m<sup>2</sup>·K), 散热片长 80 mm, 宽 5 mm, 厚 1 mm. 采用一维单元建模, 划分 5 个节点, 4 个热传导单元, 4 个对流边界单元, 1 个对流边界节点(节点 5), 如图 9 所示。

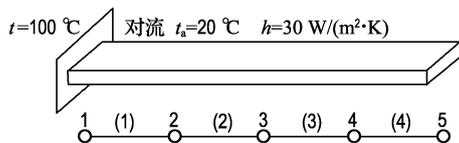


图 9 一维散热片模型与单元离散

Fig. 9 Model and mesh for one-dimension cooling plate

计算得到的结果与文献[11]中的结果比较如表 1 所示, 从表中可以看出与文献结果吻合较好。

#### 4.2 二维散热片问题

考虑电子设备中的铝制散热片<sup>[11]</sup>, 热导率为 170 W/(m·K), 基座热流密度为 1 000 W/m<sup>2</sup>, 环境温度为 20 °C, 对流传热系数为 40 W/(m<sup>2</sup>·K)。

由于对称性, 取一半进行建模, 划分为四边形网格, 共有 3 093 个节点, 2 743 个单元. 计算得到

的温度分布云图如图 10 所示。

从图中可以看出, 结构温度分布较合理, 温度峰值为 48.463 °C, 与 ANSYS 软件计算值的前 5 位完全相同。

表 1 一维散热片温度分布结果比较

Tab. 1 Temperature distribution results comparison of one-dimension cooling plate

节点	t/°C	
	SiPESC	文献[11]参考解
1	100.00	100.00
2	75.04	75.03
3	59.79	59.79
4	51.56	51.56
5	48.91	48.90

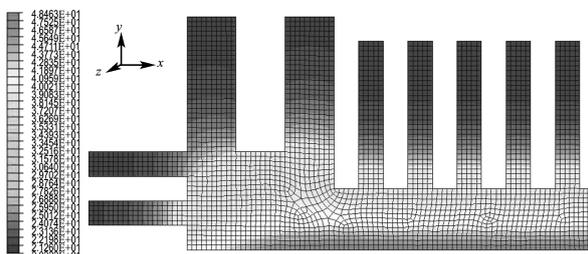


图 10 二维散热片温度分布云图

Fig. 10 Temperature contour plot for two-dimension cooling plate

#### 4.3 换热管道传热

为了提高换热管道的传热特性, 在内壁形成一系列轴向散热片, 其中材料的热导率为 400 W/

( $\text{m} \cdot \text{K}$ ), 内部流体温度  $80\text{ }^\circ\text{C}$ , 内壁对流传热系数为  $150\text{ W}/(\text{m}^2 \cdot \text{K})$ , 外部环境温度为  $15\text{ }^\circ\text{C}$ , 外壁对流传热系数  $30\text{ W}/(\text{m}^2 \cdot \text{K})$ 。

划分为三角形网格, 共有 3 632 个节点, 6 484 个单元, 计算得到的温度分布云图如图 11 所示。

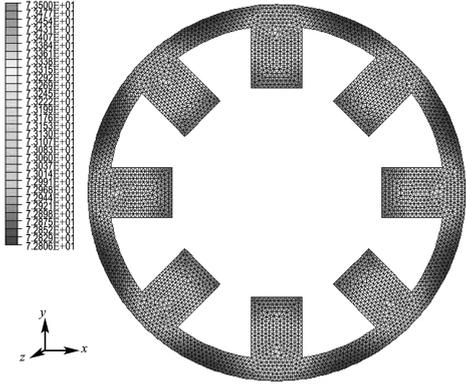


图 11 换热管道温度分布云图

Fig. 11 Temperature contour plot for heat exchange tube

从图中可以看出, 管道温度分布从  $72.806\sim 73.500\text{ }^\circ\text{C}$ , 与 ANSYS 软件计算值的前 5 位完全相同。

#### 4.4 齿轮热分析

齿轮运转过程中, 由于齿间摩擦生热会对齿轮的性能产生影响, 本算例模拟齿轮运转过程中的温度分布情况, 其中材料热导率为  $230\text{ W}/(\text{m} \cdot \text{K})$ , 内壁固定温度为  $0\text{ }^\circ\text{C}$ , 各齿缘与另一齿轮接触处给定热流密度是  $100\text{ W}/\text{m}^2$  来模拟运转过程中的摩擦生热情况。

划分六面体网格, 其中有 55 910 个节点, 46 200 个单元, 计算得到的温度分布云图如图 12 所示。

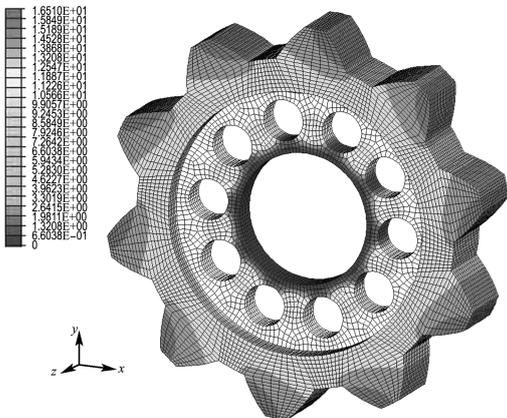


图 12 齿轮运转过程中温度分布云图

Fig. 12 Temperature contour plot for working gear

从图中可以看出, 温度峰值为  $16.510\text{ }^\circ\text{C}$ , 与 ANSYS 软件计算值的前 5 位完全相同。

## 5 结 语

本文基于 SiPESC 集成平台, 设计了传热问题有限元分析的通用程序系统, 从单元计算模块和荷载计算模块两方面介绍了程序系统的设计模式。本文程序基于 SiPESC 的插件和扩展机制, 采用 C++ 面向对象设计语言及软件设计模式中的 Factory 模式和 Builder 模式, 程序特点如下:

(1) 代码重用。提取相同算法作为通用模块, 比如对不同单元矩阵的数值积分算法进行提取, 以实现代码重用。

(2) 开放性接口。程序所有接口全部开放, 便于添加新的单元、荷载和功能。

(3) 良好的可维护性。由于采用插件和扩展机制, 以及 Factory 设计模式, 添加新的单元或荷载不需对原有代码进行修改, 程序原有功能也可由 Factory 的优先级管理实现动态替换。

本文程序系统有良好的开放性和扩展性, 可在此基础上进一步展开瞬态传热、非线性传热以及热力耦合等问题的程序研发, 也可以用于其他标量场相关问题的分析和研究。

## 参 考 文 献:

- [1] Tamma K K, Namburu R R. Computational approaches with applications to non-classical and classical thermomechanical problems [J]. *Applied Mechanics Reviews*, 1997, **50**(9):514-551.
- [2] Zienkiewicz O C, Taylor R L, Zhu J Z. *The Finite Element Method: Its Basis and Fundamentals* [M]. 6th ed. Oxford: Elsevier Butterworth-Heinemann, 2005.
- [3] Fenves G L. Object-oriented programming for engineering software development [J]. *Engineering with Computers*, 1990, **6**(1):1-15.
- [4] Forde B W R, Foschi R O, Stiemer S F. Objectoriented finite element analysis [J]. *Computers and Structures*, 1990, **34**(3):355-374.
- [5] Archer G C, Fenves G, Thewalt C. A new objectoriented finite element analysis program architecture [J]. *Computers and Structures*, 1999, **70**(1):63-75.

- [6] 张 向,许晶月,沈启戎,等. 面向对象的有限元程序设计[J]. 计算力学学报, 1999, **16**(2):216-226. ZHANG Xiang, XU Jing-yue, SHEN Qi-yu, *et al.* Object-oriented finite element programming [J]. **Chinese Journal of Computational Mechanics**, 1999, **16**(2):216-226. (in Chinese)
- [7] 马永其,冯 伟. 面向对象有限元分析程序设计及其 VC++ 实现[J]. 应用数学和力学, 2002, **23**(12):1283-1288. MA Yong-qi, FENG Wei. Object-oriented finite element analysis and programming in VC++ [J]. **Applied Mathematics and Mechanics**, 2002, **23**(12):1283-1288. (in Chinese)
- [8] 张洪武,陈飙松,李云鹏,等. 面向集成化 CAE 软件开发的 SiPESC 研发工作进展[J]. 计算机辅助工程, 2011, **20**(2):39-49. ZHANG Hong-wu, CHEN Biao-song, LI Yun-peng, *et al.* Advancement of design and implementation of SiPESC for development of integrated CAE software systems [J]. **Computer Aided Engineering**, 2011, **20**(2):39-49. (in Chinese)
- [9] 张 盛,杨东生,尹 进,等. SiPESC.FEMS 的单元计算模块设计模式[J]. 计算机辅助工程, 2011, **20**(3):49-54. ZHANG Sheng, YANG Dong-sheng, YIN Jin, *et al.* Design pattern of element computation module of SiPESC.FEMS [J]. **Computer Aided Engineering**, 2011, **20**(3):49-54. (in Chinese)
- [10] Gamma E, Helm R, Johnson R, *et al.* **Design Patterns: Elements of Reusable Object-oriented Software** [M]. Boston: Addison Wesley, 1995:11-45.
- [11] Moaveni S. 有限元分析——ANSYS 理论与应用 [M]. 3 版. 王 崧,刘丽娟,董春敏,等,译. 北京: 电子工业出版社, 2008. Moaveni S. **Finite Element Analysis — Theory and Application with ANSYS** [M]. 3rd ed. WANG Song, LIU Li-juan, DONG Chun-min, *et al.*, tran. Beijing: Publishing House of Electronics Industry, 2008. (in Chinese)

## Finite element analysis system for thermal problems based on SiPESC

YANG Dong-sheng<sup>1,2</sup>, ZHANG Sheng<sup>1,2</sup>, LI Yun-peng<sup>\*1,2</sup>, CHEN Biao-song<sup>1,2</sup>, ZHANG Hong-wu<sup>1,2</sup>

- ( 1. State Key Laboratory of Structural Analysis for Industrial Equipment, Dalian University of Technology, Dalian 116024, China;  
2. Department of Engineering Mechanics, Faculty of Vehicle Engineering and Mechanics, Dalian University of Technology, Dalian 116024, China )

**Abstract:** An object-oriented finite element analysis system SiPESC.THERMAL is developed for thermal problems based on software integration platform for engineering scientific computation (SiPESC) and engineering database. The plug-in and extension design patterns are used for the flexibility of the system which allows the analysis task to be assembled dynamically. This system has the features of generality, openness and scalability. The Factory and Builder design patterns are used for the element computation module which is convenient to the expanding of the element library. The general heat elements used in practice have been implemented in the system. The geometry face elements are defined for the flexible handling of various geometry faces and different face loads. The handler for heat flux and heat convection boundary on different boundary elements has been implemented. The system's efficiency and ability for solving engineering problems are tested by several representative numerical examples.

**Key words:** software integration platform for engineering scientific computation (SiPESC); thermal problems; FE analysis; software design patterns