文章编号: 1000_0887(2002) 12_1283_06

面向对象有限元分析程序设计及其 VC++* 实现

马永其 冯 伟

(上海大学, 上海市应用数学和力学研究所, 上海 200072)

(张禄坤推荐)

摘要: 采用面向对象的方法确定了有限元分析过程的对象、对象间的关系及类库, 并使用 VC++ 语言, 利用其 MFC 类库实现了有限元分析类库及相应窗口图形化界面的程序体序• 程序系统具有良好的可靠性, 可再用性和可扩充性• 为进一步开发大型、通用、功能性强的面向对象的有限元分析软件提供参考•

关 键 词: 面向对象; 有限元; 程序; 设计; VC++
中图分类号: TP311; O241 文献标识码: A

引 言

有限元分析程序涉及力学、应用数学和计算机科学 3 个不同学科的理论和方法, 因而其编制工作十分复杂, 且程序庞大易错• 因此, 有限元软件的开发一直寻求从方法上得以改善• 面向对象程序设计方法的出现, 给有限元软件的开发提供了新的途径• 国外学者于 1990 年开始应用面向对象方法进行有限元分析程序的设计[1,2]• 经过 10 余年的发展, 目前已编制出能够进行非线性及流体问题有限元分析, 并与 CAD 或数据库等其它软件集成的有限元分析程序[3,4,5], 但使用 VC++ 语言的少见报道• 国内在此领域的研究目前正在起步, 仅有个别文献涉足[6,7]• 本文将通过研究设计及编程实践, 讨论应用面向对象的程序设计方法进行有限元程序设计的基本思想, 及采用面向对象的 VC++ 语言进行有限元分析程序编制的基本方法• 希望引起更多国内学者的兴趣和参与, 进而开发出大型、通用、功能性强的面向对象的有限元工具软件•

1 面向对象方法及 VC++ 语言

面向对象程序设计方法是计算机程序开发方法的一种变革• 其主要特征是: 信息隐藏性; 便于在概念上体现并行和分布式结构; 便于软件的演化和扩充; 增加程序设计的灵活性和可理解性[8]• 面向对象程序设计过程在形式上为认定类和组织类之间关系的过程, 而实质为数据抽象和代码重用• 通常归结为 3 个步骤: 识别对象; 识别对象间的关系; 识别类•

* 收稿日期: 2001_10_12; 修订日期: 2002_05_21
作者简介: 马永其(1966—), 男, 宁夏人, 博士(E_mail: mayongqi@sohu.com)•

Microsoft Visual C++ 是建立在 Windows95 和 WindowsNT32 位应用程序上的强有力的复杂开发工具, 是一个 Windows 集成开发环境(IDE: integrated development environment)[9]• 其最大的优点是节约时间, 不必编译、链接便可看到应用程序的式样, 可用键盘或鼠标来完成可视化设计及应用程序的许多编程工作• 因此它允许在较短的时间内开发很复杂的 Windows 程序• 面向对象的编程基础是类, Visual C++ 中的 MFC(Microsoft Foundation Class) 类库提供了强大的面向对象编程接口, 简化了可在屏幕上创建和操纵窗口、处理绘图、使程序连接到帮助文件、便利线程、管理内存等的复杂的 API(Application Programming Interface) 函数调用, 使应用程序的编制可以用众多的可视化控件来实现• 本文的程序是在 Visual C++ 5.0 环境中开发编制的•

2 程序的设计

在有限元分析过程中, 主要应用了结构、载荷、节点、单元、自由度、矩阵、材料、高斯积分点、边界条件、求解和辅助计算等物理概念• 因此, 根据面向对象程序设计方法可确定有限元分析过程的对象为: 结构对象、载荷对象、节点对象、单元对象、自由度对象、矩阵对象、材料对象、高斯积分点对象、边界条件对象、求解对象和辅助计算对象等•

单元是有限元分析过程中最为重要的物理概念, 它是整个有限元分析过程的核心, 贯穿于整个过程的始终• 因此, 在标识对象间关联的过程中, 单元对象确定为最关键和重要的对象• 其余对象, 根据其与单元对象的关系可基本分为两类对象, 一类是服务于单元对象的对象, 另一类是被单元对象服务的对象• 在两类对象的每一类中, 根据所起作用的不同又可分为包容和利用两种• 其中载荷对象、节点对象、材料对象、矩阵对象、高斯积分点对象是服务于单元对象的对象, 而载荷对象、节点对象和材料对象是被单元对象包容的对象, 矩阵对象、高斯积分点对象是被单元对象利用的对象• 结构对象、求解对象和辅助计算对象是被单元对象服务的对象, 结构对象包容单元对象, 求解对象和辅助计算对象利用单元对象• 边界条件对象、自由度对象间接作用于单元对象, 边界条件对象被结构对象包容并被单元对象间接利用• 自由度对象被节点对象包容并被单元对象间接包容•

根据确定的有限元分析过程的对象和所标识的对象间的关联, 便形成了一个由单元类、节点类、自由度类、载荷类、材料类、边界条件类、结构类、求解类以及矩阵类和高斯节点类等组成的有限元分析类库•

具体设计以单元类为例, 单元类要实现的任务是: 1) 单元材料系数矩阵的计算、位移_应变矩阵的计算、单元刚度矩阵的计算和单元载荷列阵的计算; 2) 单元内节点的管理• 其成员数据主要有载荷、材料物性、高斯积分点、单元应变矩阵对象数组、单元刚度矩阵、单元载荷列阵和节点对象数组等• 在对具体结构进行有限元分析时, 将采用具体单元类型, 如一维单元、二维单元、三维单元等• 因此, 视单元类为虚基类, 采用 protected 关键字以便其子类能够存取有关数据, 采用 virtual 关键字以实现多态性• 这样就构筑了类之间的层次和体系结构, 形成了继承关系• 单元一旦确定后, 利用单元的自然坐标可直接建立不同类型单元的插值函数, 并可计算出相应的 Jacobi 矩阵与其行列式• 因此, 具体单元类除了继承单元类的所有数据成员和数据函数外, 还要增加两个成员数据即插值函数矩阵和 Jacobi 矩阵与其行列式, 成员函数也需增加操作这两个成员数据的相关函数, 以便计算具体单元的刚度矩阵•

设计程序的系统控制如图 1 所示• 结构类接受用户界面输入的材料、载荷、边界条件和节点位置等信息后, 发送消息并构造具体单元类、节点类、自由度类• 然后由结构类、具体单元

类、节点类和自由度类向求解类通讯并发送消息• 最后由结构类、具体单元类、节点类和求解类向辅助计算类通讯并发送消息• 从而完成一个结构的有限元分析过程•

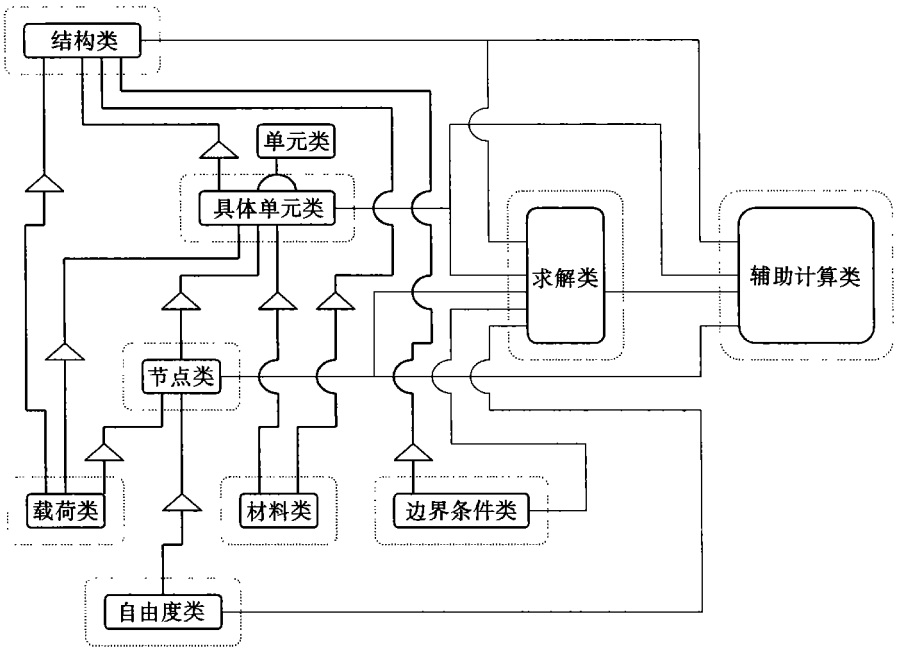

图 1 类之间的关系及程序控制

3 系统的实现

根据以上有限元分析过程的面向对象程序设计, 用 VC++ 语言具体描述类库, 以单元类为例, 它是 Visual C++ 的 MFC 类库中的 CObject 类的继承类, 具体表示如下:

```
class CElement: public CObject
{
protected:
    int noelement;              //element number
    int numnodes;               //number of element nodes
    CMaterial* elmaterial;      //pointer to element material
    CLoad* elload;              //pointer to element load
    CGausspoint* gausspoint;    //pointer to gausspoint class
    CMatrix* Dmat;              //pointer to element constitutive matrix
    CObArray BmatArray;         //array of element strain_displacement matrix
    CMatrix* kemat;             //pointer to element stiffness matrix
    CMatrix* fevec;             //pointer to element load matrix
    int* lmarray;               //pointer to equation number of dof of element's node
    CObArray nodearray;         //array of element nodes
public:
    CElement();                 //constructor
    CElement(int);              //constructor
    virtual~ CElement();        //destructor
```

```
    virtual void setDmat()= 0;                    // sets the element constitutive matrix
    virtual void setBmatArray()= 0;
    virtual void setkemat()= 0;
    virtual void setfevec()= 0;
    virtual void setnodearray()= 0;
    virtual CMatrix* getDmat()= 0;                // returns the element constitutive matrix
    virtual CObArray* getBmatArray()= 0;
    virtual CMatrix* getkemat()= 0;
    virtual CMatrix* getfevec()= 0;
    Virtual CObArray* getnodearray()= 0;
    // …(other member data and functions)
};
```

利用 Visual C++ 的 App Wizard 工具生成多文档界面(MDI)有限元分析系统的 WINDWS 应用程序框架,共生成 6 个类: About 对话框的对话框类(CAboutDlg)、整个应用程序的 CWinApp 类(CMyApp)、文档类(CMyDos)、视图类(CMyView)、框架类(CMainFrame)和子框架类(CChildFrame)• 加入有限元分析过程类库,并将有限元分析过程归纳、抽象、定义为一个问题类,由有限元分析的结构类、求解类和应力类组合而成,其成员数据为结构类对象、求解类对象和应力类对象• 将问题类作为 WINDOWS 应用程序框架中文档类的公共类型属性变量,加入其中,具体问题类和文档类属性变量描述如下:

```
class CProblem: public CObject
{
private:
    CString name;                    // string of problem name
    CStructure* structure;           // pointer to structure
    CLinearsolver* solver;           // pointer to solver
    CStress* stress;                 // pointer to stress
public:
    CProblem();                      // constructor
    CProblem(CString);               // constructor
    virtual~ CProblem():             // destructor

    void setT(CString);              // sets the problem name
    CString getT();                  // returns the problem name
    void setstructure(CStructure* );  // sets the structure
    CStructure* getstructure();      // returns the structure
    void setsolver(CLinearsolver* );  // sets the solver
    CLinearsolver* getsolver();      // returns the solver
    void setstress(CStress* );       // set the stress
    CStress* getstress();            // returns the stress
    // … (other member functions)
};
class CMyDoc: public CDocument
{
```

```
protected: // create from serialization only
    CMyDoc();
    DECLARE_ DYNCREATE(CMyDoc)

// Attributes
public:
    CProblem m_ problem;                    // attribute of CMyDoc problem

    // ... (other member date and functions)
};
```

利用 WINDOWS 操作系统的消息循环机制, 建立起相应的系统通讯体系, 其作用方式通过图 2 所示•

由 MFC 类库中提供的丰富的 for WINDOWS 的窗口、菜单、对话框、属性表等类库, 通过继承、修改和扩充编制所需的人机对话界面, 如数据和分析等菜单项, 结构对话框类(CStructure-Dialog) 等窗口图形化界面•

当用户要完成对一个结构进行有限元应力分析的计算过程时, 只要将鼠标点到菜单(界面层) 上, 即发出消息将“菜单对话”激活, 菜单对象产生反应• 通过完成对话框界面的人机交互后, 发出消息至 WINDOWS 操作系统的消息队列中参加消息循环• “问题对象”(有限元分析层) 获此消息后, 产生相应的计算动作, 完成有限元分析的计算过程•

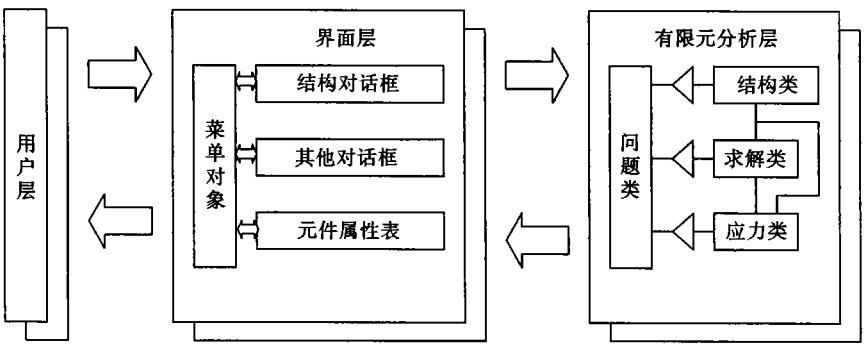

图 2 系统消息流图

4 小 结

通过对有限元分析过程进行面向对象的程序设计, 确立了相应的对象及对象间的关系, 并建立了相应的类库, 说明面向对象方法是一种易于用来进行科学和工程计算程序设计的强有力的程序设计方法• 采用 VC++ 语言描述了有限元分析类库, 通过继承、修改和扩充, 利用 MFC 类库方便地建立了面向对象的有限元分析及相应窗口图形化界面的程序体系, 即保证了有限元分析类库的灵活使用, 又保证了系统操作的清晰简便, 从而提高了程序的可靠性, 可再用性和可扩充性•

本文的开发实践将为进一步开发大型、通用、功能性强的面向对象的有限元分析软件提供有价值参考•

[参 考 文 献]

[1] Forde B W R, Foschi R O, Stiemer S F. Object_oriented finite element analysis[J]. Computers and Structures, 1990, **34**(3): 355—374.

[2] Fenves G F. Object_oriented programming for engineering software development[J]. Engineering With Computers, 1990, **6**(1): 1—15.

[3] Archer G C, Fenves G, Thewalt C. A new object_oriented finite element analysis[J]. Computers and Structures, 1999, **70**(1): 63—75.

[4] Robert Ian Mackie. An object_oriented approach to calculation control in finite element programs[J]. Computers and Structures, 2000, **77**(5): 461—474.

[5] YU Li_chao, Kumar Ashok V. An object_oriented modular framework for implementing the finite element method[J]. Computers and Structures, 2001, **79**(9): 919—928.

[6] 孔祥安. C++ 语言与面向对象有限元程序设计[M]. 成都: 西南交通大学出版社, 1995.

[7] 马永其, 陈罕, 李斯特. 采用面向对象方法的有限元程序[J]. 北京化工大学学报, 2000, **27**(3): 51—55.

[8] 宛延凯. C++ 语言和面向对象程序设计[M]. (第二版). 北京: 清华大学出版社, 1998.

[9] Kate Gregory. Visual C++ 5 开发使用手册[M]. 康博创作室 译, 文珍 审校. 北京: 机械工业出版社, 1998.

Object_Oriented Finite Element Analysis and Programming in VC++

MA Yong_qi FENG Wei

(Shanghai Institute of Applied Mathematics and Mechanics,
Shanghai University Shanghai 200072, P R China)

Abstract: The design of finite element analysis program using object_oriented programming(OOP) techniques is presented. The objects, classes and the subclasses used in the programming are explained. The system of classes library of finite element analysis program and Windows_type Graphical User Interfaces by VC++ and its MFC are developed. The reliability, reusability and extensibility of program are enhanced. It is a reference to develop the large_scale, versatile and powerful systems of object_oriented finite element software.

Key words: object_oriented programming; finite element method; program; design; VC++