文章编号:1007-4708(2014)02-0217-06

基于开放式结构有限元系统 SiPESC.FEMS 的动力时程分析通用算法构架

陈玉震, 张 盛*, 陈飙松, 张洪武

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

摘 要:基于开放式结构有限元分析软件系统 SiPESC.FEMS,针对动力时程分析算法共性,采用 C++面向对象程序设计方法和软件设计模式,研发了一种结构动力分析通用算法构架。构架的核心思想是算法与数据模型相分离,从而实现算法通用性,整个构架由四个基本类及子类构成。本文重点阐述了基本类的抽象过程和通用接口的设计思想,给出了利用插件技术实现算法构架的步骤。利用该构架已实现了 Newmark 法、Wilson-θ 法、中心差分法及改进中心差分法,并进行数值验证。研究工作表明,算法构架适用于通用时程积分算法,可方便地进行动态扩展,具备良好的开放性和重用性。

关键词:面向对象;有限元;SiPESC.FEMS;动力时程分析;算法构架

中图分类号:O242.21 **文献标志码**:A doi:10.7511/jslx201402013

1 引 言

在工程数值模拟分析中,有限元法(FEM)已成为非常重要的工具。早期由于处理问题比较简单,有限元程序更多注重于编程技巧以及运算速度,大多采用结构化程序设计方法开发,并用 Fortran 程序设计语言编写。随着有限元应用领域的不断扩展,有限元程序处理的问题越来越复杂,为了适应新问题出现的很多新单元和新算法,对软件可扩展性、代码重用性以及可维护性提出了很高的要求,结构化程序设计方法已不能满足需求[1]。

为了解决这一问题,20 世纪 90 年代面向对象方法开始引入有限元程序,很多学者致力于这方面的研究,如 Forde 等[1-5]提出了面向对象有限元分析程序的初步构架,并设计了一些基本数据类和算法类。但针对于通用算法框架的研究很少,如动力学时程分析算法,更没有一个统一的程序构架。因此,开发一个基于面向对象具有统一接口、可扩展性强、灵活、方便的动力时程分析算法构架就变得非常重要。

收稿日期:2012-12-08;**修改稿收到日期**:2013-05-05.

基金项目:国家基础性发展规划(2010CB832704);国家高技术研究发展计划(2012AA050901);国家自然科学基金(11072050,11232003,91315302);中央高校基本科研业务费专项(DUT12ZD206)资金项目.

作者简介:陈玉震(1987-),男,博士;
张 盛*(1976-),男,博士,副教授
(E-mail: zhangs@dlut.edu.cn).

开放式结构有限元系统 SiPESC.FEMS[6]是基于工程数据库和插件技术,采用面向对象方法开发。通过设计平台和插件的标准接口及插件管理功能,实现功能的动态扩展,并通过工程数据库来实现大规模有限元模型数据的管理。此系统的特色是提供一个规范化和可扩展的设计方案,开发者可以根据自己的需求方便添加自己的计算模块。

本文基于 SiPESC.FEMS,建立面向对象动力时程分析的通用算法构架,提出算法与数据模型相分离的设计思想,重点阐述了算法类 MSolver、数据模型类 MModel、算法参数类 MParameter、算法封装类 MFemsAlgorithm 及相应子类的抽象过程及程序实现,并通过插件技术及工厂模式[7]实现整个算法流程动态搭建过程。利用这一构架实现了 Newmark 法、Wilson-θ 法、中心差分法及改进中心差分法的开发,并进行了数值对比验证。

2 算法构架的设计

动力时程分析在工程数值模拟中已广泛的应用,特别是在抗震领域,发展至今已出现了很多成熟的算法,如 Newmark 法、Wilson-θ 法、中心差分法、Houbolt 法、HHT 法以及数学家们推崇的四阶龙格-库塔法等[8-11]。

结构动力平衡方程为

$$[M]\{\ddot{x}\}+[C]\{\dot{x}\}+[K]\{\boldsymbol{x}\}=\{\boldsymbol{F}\} \quad (1)$$

式中 $[M]$,$[C]$和$[K]$分别为 n 阶质量、阻尼及刚

度矩阵，$\{\boldsymbol{F}\}$为与时间相关载荷向量，$\{\boldsymbol{x}\}$为结构位移向量。

程序主要由两个部分组成：算法和数据。Pascal之父 Niklaus Wirth 提出著名公式：

算法＋数据结构＝程序

纵观计算机发展史，算法和数据这两个主要方面一直保持不变，发展与演化的只是它们之间的关系（程序设计方法）。据此，本文设计的算法构架也主要由两个核心模块构成：算法模块和数据模块。

时程积分各算法的本质是用近似的方法求解二阶线性常微分方程（动力学平衡方程），这是共性；各算法每个时间步内对已知响应量（位移、速度和加速度）与未知响应量（位移、速度和加速度）关系的假设不同，这是异性。共性决定了各算法所处理的数据相同，数据的相关基本运算（数据生成、存取等）相同。将数据共性抽象成数据模型类 MModel，独立于算法，这样能有效减少程序代码重复度。异性决定了各算法计算流程及相应参数不同，据此，创建算法参数类 MParameter 及对应于各算法的子类、算法类 MSolver 及对应于各算法的子类。最后，创建封装类 MFemsAlgorithm 实现整个算法流程的封装。以上四个基本类及其子类构成整个算法构架。

本文提出算法构架不仅适用于动力时程分析，还适用于其他有限元分析问题。构架的搭建过程不变，首先根据新的分析问题所对应的方程形式，合理抽象数据模型类 MModel，根据算法的特点抽象 MParameter 类及 MSolver 类，最后创建封装类 MFemsAlgorithm 实现整个构架，如图 1 所示。

2.1 MModel 类及其子类

时程积分各算法的数据模型共性包括：每一个时间步内相关位移向量、速度向量和加速度向量的提取和存储，指定时间载荷向量的生成，总质量阵与指定向量的乘积计算、总阻尼阵与指定向量的乘积计算、总刚度阵与指定向量的乘积计算以及等效静力方程组求解等。据此，创建数据模型类 MModel 及其子类，将这些共性抽象成相应的接口。其中主要接口如下：

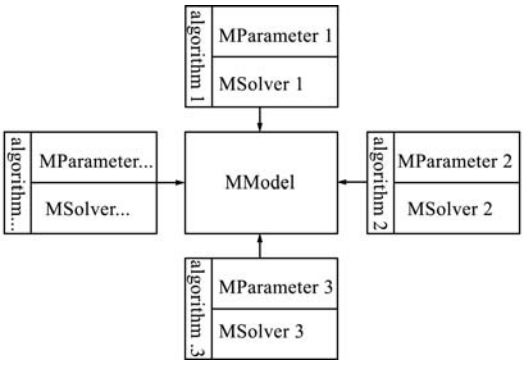

图 1 构架核心思想简图

Fig. 1 Core idea of framework

```
bool upDateModel(bool isRepeated);//数据更新
const MVector getU(int stepIndex)const;//获取指定载荷步的位移向量
void setU(int stepIndex,const MVector &u);//存储指定载荷步的位移向量
const MVector getV(int stepIndex)const;//获取指定载荷步的速度向量
void setV(int stepIndex,const MVector &v);//存储指定载荷步的速度向量
const MVector getA(int stepIndex)const;//获取指定载荷步的加速度向量
void setA(int stepIndex,const MVector &A);//存储指定载荷步的加速度向量
const MVector getR(double time)const;//获取指定时间的全局载荷向量
bool formMCK(double a1,double a2,double a3);//形成等效刚度阵
MVector calculateMU(const MVector &u);//总质量阵与指定向量的乘积计算
MVector calculateCU(const MVector &u);//总阻尼阵与指定向量的乘积计算
MVector calculateKU(const MVector &u);//总刚度阵与指定向量的乘积计算
MVector solve(const MVector &R);//每个载荷步内等效静力方程组求解
```

其中对于相关数据的处理，本文构架采用了 SiPESC.FEMS 的工程数据库管理系统，无须关心具体数据的存取过程和存放位置，按照其提供的统一接口，即可实现对数据操作，方便、简洁且安全性高。

在以上接口的实现过程中，可根据问题的复杂程度创建相应的计算功能插件，实现相应的接口，如 calculateMU ()、calculateCU ()、calculateKU () 等接口，由于一般工程问题中$[M]$，$[C]$和$[K]$矩阵维数非常大，其与向量的直接乘法运算计算量大、程序实现困难，据此本文设计 MComputeMU、MComputeCU 及 MComputeKU 插件，在单元级

别实现乘法运算，最后再累加到全局，实现总体质量、阻尼及刚度矩阵与指定向量的乘积，在实现接口时只需要调用相应插件即可，使程序更加灵活、方便；另外，当遇到新问题接口的实现发生变化时，可以利用插件机制实现动态替换，便于程序的扩展、更新、维护。

数据模型类 MModel 的接口设计具有以下优点。

(1) 独立性。内部接口只涉及模型数据的基本运算，与算法流程无关；独立于算法，不受算法的限制。

(2) 通用性。现行动力时程分析程序对于数据的处理一般是根据算法流程的需求而设计，各模块相互独立，没有形成统一的接口，不具备通用性。而本文提出的数据模型内部各接口的设计则反应了动力方程的数值特性，涵盖了动力方程有关数据的基本操作，如有限元中的单元类型、边界条件/多点约束，质量阵，阻尼形式等都是通过模型内部机制处理。在实现算法时，只要符合接口，算法与模型就可以对接，进行计算，通用性强。

2.2　MParameter 类及其子类

时程积分各算法的参数不同，每个算法都有相应的算法参数类。据此，抽象出算法参数基类 MParameter类及对应 Newmark 法、Wilson-θ 法等的子类。其中主要接口如下：

```
bool initialize (const MTimeStepInfo & stepInfo,QVector <double> paras);//初始化
const MTimeStepInfo getTimeStepInfo ( ) const;//获取时间步数据
QVector <double> getParas()const;//获取算法参数
QVector <double> getConstants()const;//获积分常数
```

2.3　MSolver 类及其子类

各时程积分算法由于近似假设不同，其在计算流程上存在差别，如中心差分法需要计算$\{\boldsymbol{X}_{-1}\}$；Wilson-θ 法需要先计算 $t_{k+\theta}$时刻的响应，然后再推出 t_{k+1}时刻的响应等。但它们也存在共性，如初始加速度向量的求解、等效刚度矩阵的形成、每个时间步内等效载荷向量的形成以及响应量的计算等，根据这些共性，抽象出 MSolver 基类。继承 MSolver 基类，根据算法异性增加相应接口，创建 Newmark、Wilson-θ 等相应算法子类。其中主要接口如下：

(1) 算法共性接口

```
bool initialize(MModel & model,const MParameter & parameter);//将参数和模型传给算法
bool calculateInitialA(const MVector & initialR);//计算初始加速度
bool formEffectiveMatrix ();//形成等效刚度矩阵
MVector calculateEffectiveLoad(int stepIndex);//计算等效载荷向量
bool calculateDisplacement (int stepIndex, const MVector R);//计算指定载荷步的位移向量
bool calculateVelocity(int stepIndex);//计算指定载荷步的速度向量
bool calculateAcceleration(int stepIndex);//计算指定载荷步的加速度向量
bool solve ();//组织以上步骤实现逐步计算
```

(2) 算法异性接口

```
MWilsonThet：
updateDisplacement ();//将{X_{k+θ}}更新为{X_{k+1}}
MCentralDifference：
calculateBackwardDisplacement ();//计算{X_{-1}}
MNewCentralDifference：
calculateZeroStepLoad ();//计算{F_0}
calculateBackwardLoad ();//计算{F_{-1}}
calculateBackwardDisplacement ();//计算{X_{-1}}
```

通过对以上接口的调用，即可实现时程积分算法的整个流程。当需要实现新算法、增加新接口时，可以通过继承 MStructuralDynamicIntegration 类，增加相应的方法，创建新的算法类，整个过程方便、简洁，代码能够得到很好的重用，可扩展性强。算法类设计的一个重要特性是通用性，算法编程只需调用数据模型对象 MModel 的接口，而无需考虑其内部如何操作，这样就解决了算法的数据依赖问题以及模型依赖问题；如果要解式(1)问题，不管其对应是动力结构有限元、非傅立叶热传导或是其他具有相似控制方程的问题，只要 MModel 接口不变，算法对象即可与之拼接求解。算法类设计另

一个特性是简便，算法是在高层次的角度实现，编程时只需关注算法主要步骤，针对具体问题计算通过 MModel 的接口调用实现，从而屏蔽了细节问题；算法的数学列式与代码的实现几乎是等价的，极大简化了编程，并有效保证了代码质量，便于代码调试和维护。

2.4 MFemsAlgorithm 类及其子类

MModel 类、MParameter 类、MSolver 类创建完毕后，搭建一个算法程序构架的基本部件都已准备完毕，进一步抽象 MFemsAlgorithm 类及其子类，来实现整个算法的封装。

主要接口如下：

bool initialize(MDataModel & model, bool isRepeated)；//初始化，将数据传入

bool start(MDataObject & data)；//启动计算流程

3 算法构架的实现

3.1 基于插件的软件设计

SiPESC 的核心思想是“微核心（平台）＋插件”，由平台定义统一插件拼装接口，由插件实现不同的功能扩展[6]。根据这一思想，本文利用平台提供的插件设计器，将动力时程分析算法构架的基本类设计成相应插件，实现与平台的动态链接。不同的算法插件，可以在平台上动态地加载与卸载，实现“即插即用”功能，使得算法构架在添加新算法时更灵活、方便。

3.2 算法构架的动态搭建

如图 2 所示，以 Newmark 法为例，构架的搭建过程可以分为以下几个步骤。

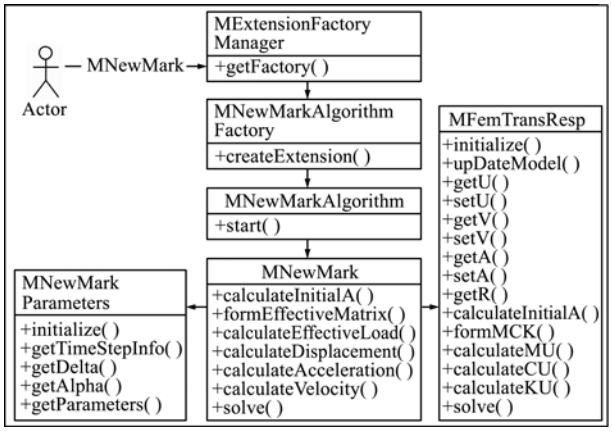

图 2 算法构架动态搭建

Fig. 2 Dynamic construction of the algorithm framework

（1）程序获取用户指定的时程积分算法名称。

（2）通过插件管理器 MExtensionManager 创建工厂管理器 MExtensionFactoryManager。

（3）工厂管理器根据用户指定算法名称，通过接口 getFactory ()创建相应算法工厂插件。

（4）工厂插件调用接口 createExtension ()创建算法插件。

（5）算法插件调用参数插件和数据模型插件，通过接口 solve ()完成循环计算。

由以上基本步骤可知，整个构架基于四个基本类，通过工厂模式[7]和插件技术动态搭建的，用户只需提供算法名称，程序即可自动实现整个流程。

研发新算法时，只需按照构架格式设计出四个基本插件，并在相应位置替换已有插件，整个算法流程即可实现。

3.3 算法构架的通用性

本文算法构架的核心思想是算法与数据模型相分离，体现了高效算法的要求，其思想同样适用于其他有限元分析问题算法，如结构自振分析、热传导分析、非线性分析以及动力非线性分析，针对这些有限元算法所处理方程的特性，抽象相应数据模型 MModel 类；根据各算法共性和异性，抽象 MParameter 类及 MSolver 类，并利用插件技术即可搭建出一个完整的不同有限元分析问题流程，进一步体现了该算法构架的良好通用性。

4 数值算例

利用本文提出的算法构架实现了 Newmark 法、Wilson-θ 法、中心差分法及改进中心差分法，并进行数值验证。

4.1 三层厂房地震载荷时程分析[12]

图 3 为三层厂房等效剪力框架，令 $k=1$ N/m，$m=1$ kg，瑞利阻尼系数为 $\alpha=0.015$，$\beta=0.02$，受水平地震加速度为 sint m/s^2。令 $T=3.25$ s，选取时间间隔为 $T/20$，Newmark 法（$\delta=0.5$，$\alpha=0.25$）、Wilson-θ 法（$\theta=1.4$）。三层厂房顶部位移响应结果列入表 1 和图 4 所示。

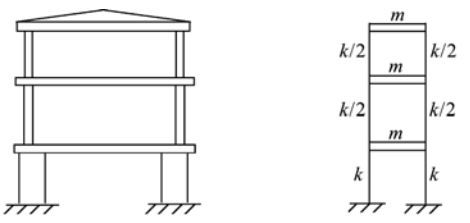

图 3 三层厂房模型

Fig. 3 Three floor building model

表 1　三自由度系统顶部位移响应

Tab. 1　Displacement of the top of the 3-dofs system

Methods	x_3/m					最大误差/%
	t=5.2 s	t=7.8 s	t=16.9 s	t=20.8 s	t=28.6 s	
Theory	−3.243	4.587	−3.420	4.559	−2.631	—
Newmark	−3.233	4.567	−3.413	4.532	−2.652	0.79
Wilson-θ	−3.238	4.567	−3.428	4.519	−2.682	1.94
Central	−3.247	4.592	−3.419	4.566	−2.616	0.58
Central_New	−3.228	4.558	−3.410	4.519	−2.663	1.19

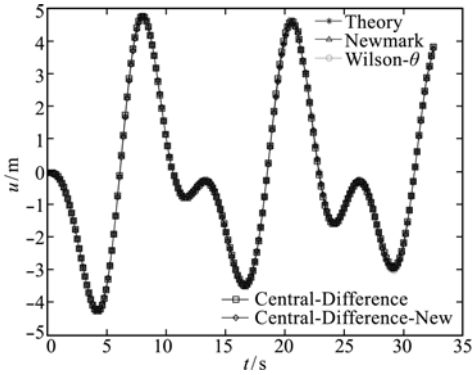

图 4　三自由度系统顶部位移时程曲线

Fig. 4　Displacement of the top of the 3-dofs system

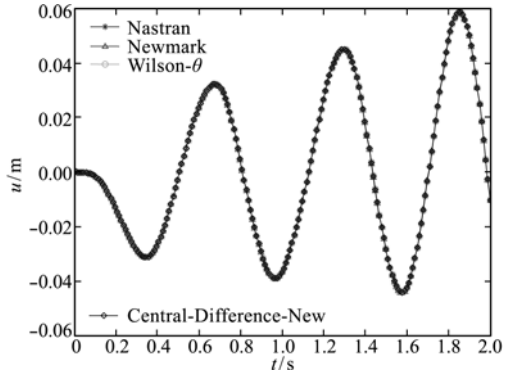

图 6　小雁塔顶部位移时程曲线（x 方向）

Fig. 6　Displacement of the top of tower(x)

由表 1 和图 4 可知，根据本文构架实现的 Newmark 法、Wilson-θ 法、中心差分法及改进中心差分法计算结果与理论解一致，针对本题选取时间间隔为 $T/20$，误差在 2%以内，验证了程序的准确性。

4.2　小雁塔受水平地震加速度时程分析

如图 5 所示小雁塔简化模型。模型经过简化，由四面体单元构成，节点总数为 3734 个，单元总数为 15272 个。材料属性：密度为 1900 kg/m^3，弹性模量为 1790 MPa，泊松比为 0.1。底部全约束，施加某一水平地震加速度载荷，取 Δt 为 0.01 s，计算 200 步，Newmark 法（δ=0.5，α=0.25）、Wilson-θ 法（θ=1.4）。节点 180 处位移结果与 Nastran 结果比较如图 6 所示。可以看出，根据本文构架实现的Newmark法、Wilson-θ 法、改进中心差分计算结果与 Nastran 结果一致，验证了程序的准确性及构架的合理性。

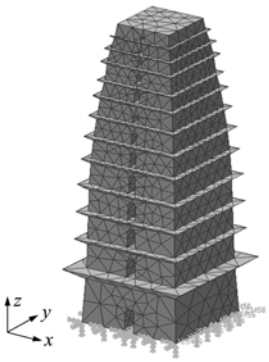

图 5　小雁塔简化模型图

Fig. 5　Xiaoyan Pagoda model

5　结　论

本文基于 SiPESC. FEMS，利用插件技术及软件设计模式，研发了一种算法与数据模型相分离的结构动力时程分析通用算法构架，并利用此构架实现了 Newmark 法、Wilson-θ 法、中心差分法及改进中心差分法。数值算例进一步验证了该设计思想的正确性。同样算法模块也应用到 SiPESC. Thermal 的瞬态热传导分析。研究工作表明，构架对于不同的时程积分算法具有很好的通用性，可方便地进行动态扩展，具备了良好的开放性和重用性。

另外，该算法构架同样适用于其他有限元分析问题，如结构自振分析、动力频率响应分析及动力非线性分析等。

参考文献（References）：

[1] Forde B W R, Foschi R O, Stiemer S F. Object-oriented finite element analysis[J]. *Computers and Structures*, 1990, **34**: 355-374.

[2] Dubois-Pelerin Y, Zimmermann T. Object-oriented finite element programming: III. An efficient implementation in C++[J]. *Computers Methods in Applied Mechanics and Engineering*, 1993, **108**: 165-183.

[3] Dolenc M. Developing extendible component-oriented finite element software[J]. *Advances in Engineering Software*,2004,**35**:703-714.

[4] 魏泳涛.面向对象的有限元程序设计方法[D].四川大学,2000.(WEI Yong-tao. Object-Oriented Approach to Finite Element Programming[D]. Sichuan University,2000.(in Chinese))

[5] 杨春峰.面向对象有限元分析程序构架设计[D].大连理工大学,2007.(YANG Chun-feng. Object-Oriented Finite Element Analysis Program Architecture Design[D]. Dalian University of Technology,2007.(in Chinese))

[6] 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))

[7] 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):46-52.(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):46-52.(in Chinese))

[8] Gamma E,Helm R,Johnson R,et al. *Design Patterns: Elements of Reusable Object-Oriented Software*[M]. Boston:Addison-Wesley,1994,11-45.

[9] Newmark N M. A method of computation for structural dynamics[J]. *Journal American Society of Civil Engineers*,1959,**85**(EM3):69-94.

[10] Wilson E L. A Computer Program for the Dynamic Stress Analysis of Underground Structures,SESM[R]. University California at:Berkeley,1968.

[11] Houbolt J C. A recurrence matrix solution for the dynamic response of elastic aircraft[J]. *Journal of Aero Science*,1950,**17**:540-550.

[12] Hilber H M,Hughes T J R. Improved numerical dissipation for time integration algorithms in structural dynamics[J]. *Earthquake Engineering and Structural Dynamics*,1977,**5**:283-292.

[13] 张亚辉,林家浩.结构动力学基础[M].大连:大连理工大学出版社,2007.(ZHANG Ya-hui,LIN Jia-hao. *Fundamentals of Structural Dynamics*[M]. Dalian:Dalian University of Technology Press,2007.(in Chinese))

Algorithm framework for time-history analysis based on open finite element system SiPESC. FEMS

CHEN Yu-zhen, ZHANG Sheng*, CHEN Biao-song, ZHANG Hong-wu

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

Abstract:Based on the open finite element system SiPESC. FEMS,a new general algorithm framework for structural time-history analysis is constructed by using C++ object-oriented programming method and software design patterns method,which considers the common features of time-history analysis algorithms. The core idea of the framework is that data model class and algorithmic class are designed individually. The whole framework includes four basic classes. The abstract process of the basic classes and the design idea of common interface are introduced,and the development process of the framework is given by employing the software plug-in technology. Furthermore,Newmark method,Wilson-θ method,central difference method and its improved method are implemented and numerical verifications are given. This investigation shows that the algorithmic framework has favorable extensibility and reusability which is applicable to the general time-history analysis.

Key words:object-oriented;finite element;SiPESC.FEMS;time-history analysis;algorithm framework