第20卷 第2期 力 学 进 展 Vol. 20 No. 2
1990年5月25日 ADVANCES IN MECHANICS May 25, 1990

有限元程序自动生成系统及有限元语言

梁 国 平
中国科学院数学研究所，北京（邮政编码100080）

提要 本文介绍了最近发展起来的应用软件的一个新方向——有限元语言和有限元程序自动生成系统。有限元语言是一种比目前的算法语言更高层次的语言。人们采用有限元语言和自动生成系统很容易编写有限元程序，可大大缩短有限元的编程时间，原来需要数月甚至数年才能完成的有限元程序，可望在数天内完成。最后展望了有限元语言对有限元的应用和发展所带来的影响。

关键词 有限元程序；自动生成系统；有限元语言；元件化程序设计方法；自动生成技术

1 引 言

有限元方法自从50年代末发明以来，经过30年的发展，已逐渐成熟，各个领域的工程师和科学家已普遍理解和接受有限元方法和技巧。随着计算机尤其是微型计算机的迅速发展和普及，计算机费用过高的障碍一般已不复存在，有限元方法理应被各个领域的工程师和科学家普遍采用。然而实际情况并非如此。至今有限元还只为少数工程师和科学家采用，还未普遍成为他们的有力工具。其原因很简单，主要障碍来自有限元程序的复杂性。

有限元方法的通用性造成了其应用的广泛性，但也带来了程序的复杂性及困难。任何一个功能较完善的有限元程序起码要上万条程序，按照目前编程的技术，1万条程序需要花费好几个人年的劳动才能完成。这对于一个普通的工程师和科学家来说显然是难以接受的。为了克服这一困难，只能由一部分人编制有限元程序来提供工程师和科学家们使用。这就是各种各样名目繁多的有限元程序及有限元程序系统纷纷问世的原因。

这些有限元程序虽然解决了部分工程师和科学家的某些需要，但距离满足他们大多数人的普遍需要来说相差甚远。其原因如下：首先，一个有限元程序，不管它多么庞大，通用性多强，也只能解决很少的一部分问题。因此用户往往需要修改或加进一部分程序才能达到自己的目的，而这些工作如果没有得到原程序编制者的参加一般是十分困难甚至是办不到的。第二，有限元程序并不象算法语言那样有统一文本，按某种语言写出来的程序在不同的编译器下都可运行。而不同的有限元程序要求用户填写的数据文件则完全不同。用户要理解、信任和习惯任何一个有限元程序都需要付出代价，花费许多钱和精力。这就是为什么大多数用户仍然愿意采用旧的落后的但自己熟悉的有限元程序系统，而不愿意接受新的先进的但不熟

悉的有限元程序系统的一个重要原因．第三，通用性越强的程序按理越容易满足用户的需要，然而另一方面，对于这样的程序，用户也越难理解和掌握，当然也越难修改，填写的数据也会越多．这是限制有限元通用性无限扩大的一个重要因素．程序越专用使用越方便，但所能解决的问题也就越少．

因此向工程师和科学家普及和推广有限元法的最好方法，显然不是简单地让他们去使用别人编好的有限元程序系统，而是帮助他们建立自己的有限元程序或程序系统．当然这种“帮助”要十分有效，要让他们在很短的时间内就能建立自己的有限元程序，并且很容易维护和修改．有限元自动生成系统就是为解决这一任务而发展起来的软件系统．

2 功能和特征

什么是有限元程序自动生成系统？有限元程序自动生成系统具有哪些特征？一个有限元程序自动生成系统（以下简称自动生成系统）最少应具有以下的功能和特征：

①自动生成系统面向用户的必然是一种比目前提供的各种算法语言（如 FORTRAN，ALGOL，PASCAL，BASIC 等）更高一层的有限元语言．这种语言很容易被懂得有限元方法的工程师、科学家和大学生理解与接受．

②自动生成系统首先把用户写的有限元语言“程序”翻译成某种目前常用的算法语言（如 FORTRAN）程序，然后再由该算法语言编译器译成代码程序．因此用户可以直接阅读到由自动生成系统产生的有限元程序，这将有利于用户的理解与修改．

③用户采用有限元语言写有限元“程序”的效率要比直接采用某种算法语言（如 FORTRAN）写有限元程序提高 1 个数量级以上．因此采用自动生成系统能大大提高编制有限元应用软件的生产率．

④采用自动生成系统能产生绝大部分的有限元程序．既包括线性的，也包括非线性的，以及耦合问题的有限元程序；既包含静态问题，也包含动态问题．它不仅能用于最早采用有限元方法的结构力学领域，也能用于其他采用有限元方法的任何领域（如流体力学，物理学，化学，生物学等学科）．

⑤自动生成系统允许用户同时采用有限元语言和算法语言这两种语言编写有限元程序，以便满足用户的特殊需要．就如同采用算法语言编写程序有时需要插入代码程序那样．

3 实现办法

实现办法不是唯一的，这里谈一谈我们的有限元程序自动生成系统 FEPG[1]（Finite Element Program Generator）所采用的办法．

首先是确定目标程序采用哪种算法语言．FEPG 以 FORTRAN 为其目标程序，然后就是如何把用户写的有限元语言程序翻译成目标算法语言即 FORTRAN 程序．自动生成系统本身用什么语言编写并不是一个很重要的问题，只要便于作字符处理的任何一种高级语言都可以．一般不需要与目标程序采用相同的算法语言，它们所采用的语言可以毫无关系．FEPG 采用的是 PASCAL．

众所周知，算法语言编译器把用户写的语言程序全部编译成代码程序．而自动生成系统则不同，一般来说它并不从头到尾产生全部的有限元程序．我们知道，一个完整的有限元程序有好几千行，全部生成要花许多时间．FEPG的做法是把有限元程序分成两部分：不需要经常变动的固定部分（如求解器）和经常变动的变化部分（如单元子程序），固定部分程序

用FORTRAN语言事先编好，变化部分则由自动生成系统自动产生．FEPG对于线性问题主要自动产生单元子程序，前处理程序以及很少量的主程序，其余部分都是固定的FORTRAN源程序．

采用新的程序设计方法．这种新方法的原始思想是E．L．Wilson[2]在80年代初设计SAP-80时提出来的，我们称之为“元件化”程序设计方法[3]．它与现在普遍采用的程序包（库）方法不同，它的主要模块是“元件程序”而程序包的主要模块是子程序．子程序的形式参数主要是变量和数组，而元件程序的形式参数则为文件．由于文件所包含的信息远远多于变量和数组，因此元件程序比子程序模块化程度更高，调用元件程序的“主程序”比调用子程序的主程序简单得多．因此“元件化”程序设计方法能大大降低程序设计的复杂性．

自动生成系统采用符号处理方法，具有一定的人工智能，如公式推导能力（如求任意多项式的偏导数）．

自动生成系统要尽可能采用统一的和最一般的模式处理问题．如FEPG对微分方程的描述采用了最一般的形式虚功方程（即广义解）；把单元子程序的两个主要计算步骤，形函数的计算与单刚的计算完全分开，并采用统一的参考有限元数值积分法；允许用户书写任何FORTRAN语言能接受的表达式．

自动生成系统不需要庞大的程序库（一般有限元程序系统需要庞大的单元库），但是需要一个公式库，以便节省用户寻找和书写公式的时间．对于那些还不大熟识有限元方法的用户，公式库对他们尤其重要．FEPG的公式库为用户提供了各种常用的形函数公式（包括一维、二维和三维），提供了线弹性力学在各种坐标系（包括直角坐标，极坐标，柱坐标，球坐标等八种坐标系）下的虚功方程表达式，等等．这个公式库虽然不大（只有几十K字节），但已足以产生全部线性弹性力学的常用单元．更重要的是FEPG还具有自动产生公式的功能（即公式推导能力）．这将大大扩大有限元方法的解题能力，如三维线弹性力学在球坐标系下虚功方程表达式长达近百行，由人推导这样复杂的公式将十分困难，有时甚至是不可能的．这对发展各种复杂的壳体单元可能是一条重要的途径．

4 FEPG简介

现在介绍组成FEPG的主要五个部分．

4.1 *单元子程序自动生成系统* 用户只需输入坐标变换表达式，形函数表达式，势能表达式，载荷表达式等基本公式及一些简单的规定，本系统就能自动产生用户所要求（计算单刚、单质及单元载荷）的单元子程序．位移函数个数，节点数，广义位移数，空间坐标维数，方程阶数（只要四阶以内）均不限；也不论各种表达式多么复杂，方程是否对称，本系统都能准确无误地产生用户所要求的单元子程序．并且表达式的全部符号（如坐标变量、位移函数、广义位移等等）均由用户规定，用户无须改变自己熟悉的符号．

4.2 *按表格方式输入数据的前处理器自动生成系统* 各种表格之间的关系（即树结构关系），各种表格的名字以及表格内各列数组变量名（即字段名），均由用户规定．本系统所具有的自动生成功能比一般有限元程序系统（如SAP-80）的前处理器功能要强，使用方便且容易掌握．

4.3 *内存动态分配自动生成系统* 本系统可帮助用户按静态定义数组的程序自动实现动态分配，

4.4 *批命令自动生成系统* 按照用户的要求生成有限元程序或程序系统所需要的批命令文件.

4.5 *若干元件程序及各种求解器* 求解对称和非对称代数方程组的各种变带宽求解器.

5 一个例子

本节通过一个简单例子说明如何采用FEPG规定的有限元语言编写单元子程序，本例子选自文[4]. 为了产生求解平面 Poisson 方程的双线性四边形单元而编写的有限元语言程序如下.

```
ELEM
\equation u/xx+u/yy=-2*(2-x*x-y*y)
\solution u=(1-x*x)*(1-y*y)
\boundary condition u=(1-x*x)*(1-y*y)
defi
disp u
var u1,u2,u3,u4
refc p,q
coor x,y
dord 1
node 4

shap
u=[(1-p)*(1-q)/4]u1+[(1+p)*(1-q)/4]u2
 +[(1+p)*(1+q)/4]u3+[(1-p)*(1+q)/4]u4

tran
x=[(1-p)*(1-q)/4]x(1)+[(1+p)*(1-q)/4]x(2)
 +[(1+p)*(1+q)/4]x(3)+[(1-p)*(1+q)/4]x(4)

y=[(1-p)*(1-q)/4]y(1)+[(1+p)*(1-q)/4]y(2)
 +[(1+p)*(1+q)/4]y(3)+[(1-p)*(1+q)/4]y(4)

gaus=4  -1.,-1.,1.; 1.,-1.,1.; 1.,1.,1.; -1.,1.,1.;

stif
dist=[u/x;u/x]+[u/y;u/y]

load=[u]*2.*(2.-x*x-y*y)

end
```

其中第一行 ELEM 是单元子程序名，这个名字由用户规定．第二行至第四行每行以续行符“\”开头，是注解行．从第五行至空行之间为第一段，它分别给出了未知函数名 u，广义位移变量名 u_1,u_2,u_3,u_4，参考坐标变量名 p,q，原坐标变量名 x,y，虚功方程中未知函数的最高阶微分阶数，以及单元节点数．用户在此段规定的名字将出现在以下各段的表达式中．

第二段以关键字 shap 为标识符，给出未知函数的形函数表达式（在参考坐标系下），对应于每个广义位移的基函数表达式，写在它前面的方括号内．

第三段以关键字 tran为标识符，给出原坐标变量对参考坐标变量的坐标变换表达式．其中 $x(i)$ 和 $y(i)$ 表示 i 个节点的坐标值（$i=1,2,3,4$）．

第四段以关键字 gaus 为标识符，给出数值积分点的个数，以及每个数值积分点的参考坐标系的坐标值和权系数．

第五段以 stif 关键字为标识符，在关键字 dist 后面给出虚功方程表达式．其中 [·;·] 表示两个函数的内积，前面为未知函数，后面为相应未知函数的变分．任何线性微分方程广义解（包括非对称的情况）都能用上述形式表达．

第六段以 load 关键字为标识符，给出方程的右端项．

第七段以 end 关键字为标识符，表示结束．

6 作用和影响

人们普遍理解和掌握了有限元语言和自动生成系统之后，会发现目前需要数月、数年才能编制好的有限元程序，那时只需数日即可完成；原先令人望而生畏的十分复杂的非线性耦合问题，那时轻而易举就可得到用户所需要的有限元程序．

人们采用有限元语言编写和修改有限元程序就象书写和修改有限元教科书那样方便和清晰．不同的有限元程序就如同不同的有限元教科书或不同的有限元论文，人们再也不会象现在那样厌烦程序的重复．有限元软件的可再用性和可移植性就在采用有限元语言的过程中自然而然地解决了．

有限元语言的普遍采用将促使编制有限元程序的工程师和科学家迅速增加，有限元程序的编制和使用将由同一人（或同一部分人）完成，当前普遍存在的有限元程序编制者与使用者分离的现象将逐渐减少，目前占主导地位的庞大的有限元程序系统的影响和作用将会逐渐减弱．人们将普遍采用自己编制的程序计算有限元问题．

有限元方法在有限元语言的帮助下将会在各个领域出现前所未有的广泛的普及和应用．有限元语言把有限元方法与有限元程序熔为一体．目前大学里普遍存在的，只教学生有限元方法不教学生有限元程序，学生只学方法，不编制程序和不计算有限元问题的弊病，将因有限元语言的普及而被彻底铲除．任何学过有限元语言的大学生都能轻松自如地编制和使用自己的有限元程序．只有到了那个时候，有限元才算真正成为广大工程师和科学家不可缺少的有力工具．

其他应用软件领域很可能纷纷相继推出自己的语言与自动生成系统，整个应用软件领域将出现新的飞跃，使应用软件的生产率提高到一个崭新的水平．

7 面临的问题和今后的发展

对于广大的工程师和科学家来说，有限元程序自动生成技术和有限元语言是一种新的思

想和新的技术。他们需要有一个认识、学习、理解、熟识和习惯的过程。因此普及这种新思想和新技术是当前推动有限元应用的一个重要方面。

由于有限元程序自动生成技术和思想的普及和推广，今后将会出现各种各样的有限元程序的自动生成系统及其相应的有限元语言。各种有限元语言如何发展和统一，将是一个重要的课题。

对于线性问题，一般来说用户只需要告诉自动生成系统要做么什事情就够了，不必告诉它如何做，因此线性有限元程序的自动生成比较容易统一。但对于非线性问题，只告诉自动生成系统要做什么是不够的，还必须告诉它如何做，否则就无法按用户规定的算法产生用户所需的有限元程序。因此关于非线性有限元程序的自动生成将会更加丰富多彩。大概不可能有一个系统能自动产生所有非线性问题的有限元程序，因此非线性有限元程序的自动生成将会是一个长期的研究课题。

参 考 文 献

1 梁国平. 有限元程序自动生成系统使用说明. 中国科学院数学研究所资料 (1989年9月)

2 Wilson E L. SAP-80——Structural Analysis Programs for Small or Large Computer System (1980) (手稿)

3 梁国平, 傅子智. 计算结构力学及其应用, **2**, 2 (1985) : 121—126

4 Liag Guo-Ping. An automated generation system fr finite element programs. Proc. Symp. on Scientific Software. China University of Science and Technology Press (1989) : 159—165

FINITE ELEMENT PROGRAM GENERATOR AND FINITE ELEMENT LANGUAGE

Liang Guo-ping

Institute of Mathematics, Academia Sinica

Abstract Finite element language and finite element program generator which have been developed in recent years as a new trend of developing application softwares are introduced in this article. Finite element language is in a higher level than the current high-level languages. By using this language and the generator people can develop finite element programs very easily and the coding-time can be reduced a great deal; a finite element program which used to take several months or even several years to complete can now be expected to be finished just in a few days. This article also predicts some impact on the applications and developments of FEM.

Keywords *finite element program; generator; finite element language; component programming design; automatic generation technique*