

LDLT 分块求解计算方法在有限元分析中的编程实现

刘跃进 薛孟君

(装甲兵工程学院机械工程系 北京 100072)

摘要 针对有限元计算时遇到的大型线性方程组求解问题,提出一种解决方法,即对方程组的系数矩阵采用三角分解法,并用一维变带宽存贮,同时与分块法相结合,实现内存与外存数据的交换。这种方法节省内存,提高计算效率,且解决了内存资源不足的问题。实例表明这个算法是很有效的。

关键词 有限元法,三角分解法,分块法

中图分类号 TB12 **文献标识码** A

Program of Blocks Combining with LDLT Method for Finite Element Analysis

LIU Yue-jin XUE Meng-jun

(Department of Mechanical Engineering, Armored Force Engineering Institute, Beijing 100072, China)

Abstract The method is applied to solve the large linear equations when the finite element method is used. When blocks method is used, coefficient matrix of equations is triangulated and is stored by a dimension change bandwidth. The exchange data are carried out between EMS memory, and external memory. The method saves EMS memory, improves computational efficiency, and overcomes the problems of EMS memory. Some numerical results show that the method is more efficient than the others.

Keywords Finite element method, Triangle reduce method, Blocks method

1 引言

有限元法作为一种通用的数值法,使用方便,能获得满意的精度而广泛用于求解各种问题,其对于促进当代科学技术的发展和工程实际应用已经发挥并将继续发挥极其重要的作用^[1]。采用有限元计算时,耗费机时较多的是总体刚度矩阵的存贮和求解方程组^[2]。在对大型结构求解时,为了达到一定精度要求,往往离散模型划分的单元较多,节点和节点自由度相应很多,得到的求解方程阶数一般都很高,因而系数矩阵往往不能全部进入计算机内存。分块解法和波前法是解决计算机内存容量不够的两种常用的较好解法^[3]。

有限元法中,线性代数方程组的系数矩阵是对称的,因此可只存贮一个上三角(或下三角)矩阵。但是由于矩阵的稀疏性,仍然会发生零元素占绝大多数的情况。考虑到非零元素的分布呈带状特点,在计算机中系数矩阵的存贮一般采用一维变带宽存贮。一维变带宽存贮是最节省内存的一种存贮方法。线性代数方程组的求解分为两类:直接解法和迭代解法。直接解法以高斯消去法为基础,求解效率高。高斯消去法分为消去过程和回代求解过程两部分。三角分解法形式上看似没有消去过程,只有回代求解过程,但其在回代求解前必须由系数矩阵进行三角分解的过程,这实质就是高斯消去法中的消去过程。由于这两种解法都是用的高斯消去原理,因此它们的运算量基本相同。但在计算机上执行时,相同的系数矩阵都以一维或二维数组的形式贮存在计算机的内存中,对一个元素进行运算时,需经过从数组中取出该元素,以及运算后将该元素再送回数组中去的取数和送数过程,需要消耗一

定的机时。三角分解法运算过程中取送数的次数少,每个元素的分解都是一次完成。而采用高斯消去法时,一个元素的消元要取送 $1-D-1$ 次。因此相比高斯消去法,采用三角分解法能节省计算时间。

分块解法和波前法都是解大型线性方程组时解决计算机内存容量不够时的方法。它们的简单比较如表 1 所列。

表 1 分块解法和波前法比较

	分块解法	波前法
1	自由度集成完一批消去一批	自由度集成完一个消去一个
2	允许内存大于 $D(D+1)$ 即可求解	内存大于最大波前区就可求解,内存要求比分块解法要小
3	内外存交换次数较少	内外存交换频繁
4	无外存时也能求解	无外存时不能求解
5	程序实现简单	程序编制较复杂

分块解法在解决系数矩阵时,通常采用高斯消去法。而本文在有限元计算时,利用计算机最大内存容量,尽量减少计算机时,并提高计算效率,在采用分块解法的同时,对系数矩阵进行三角分解,且采用一维变带宽存贮,实现内存与外存数据的交换。本文在利用 VC++ 语言进行编程计算过程中,对其中一些程序设计技术进行解决^[4]。

2 程序算法

本文利用 LDLT 分块法可以在一定的内存容量下进行计算。由于 VC 基于 windows 操作系统,可充分利用计算机的整个内存,远好于在 DOS 操作系统下运行的程序只能用 640k 的内存容量,因而,本文程序采用 VC++ 6.0 进行编写,在 VC 环境中运行,最大化地利用内存。但是,当单元划

刘跃进(1977—),男,硕士,讲师,主要研究方向为结构强度及数值仿真, E-mail: lyj_gclx@sina.com; 薛孟君(1962—),男,硕士,副教授,主要研究方向为结构强度与数值仿真。

分比较密,总体刚度矩阵的存贮数据很大,采用一般正常求解方法时计算量超过整个内存时,程序就无法运行。分块法有效解决了计算量超过内存的问题,本文最大可能是在较大内存容量下将矩阵分成几块,并采用一维变带宽存贮总体刚度矩阵,利用三角分解法求解,这样既有效率又可运行程序得到结果。利用本文方法主要在编节点号时应规律,在总体刚度矩阵组合中,单元刚度矩阵投放是靠节点来进行的,本文在单元与节点的编号都较有规律,将沿 y 方向的几个列纵横分为一个小分块,并且这几个列组成的小分块的总体刚度矩阵的存贮数据可使计算机的内存容量达到饱和,然后沿 x 方向小分块的序号逐渐增加,其分块的示意图见图 1。单元块刚度矩阵在总体刚度矩阵中的排列比较有规律,其排列示意图见图 2。因此, x 方向单元划分逐渐加密,内存耗用量增加不大,程序依然可以运行。在组成总体刚度方程时,若组成一个小分块,对其进行变带宽存贮后的元素存贮到硬盘。由于相邻小分块具有重叠区,当上一分块消元后与下一分块的重叠部分应转存到两分块组成的矩阵中,并与下一分块进行叠加。最后总体刚度矩阵的数据总体存贮到硬盘。解方程时调用硬盘的数值,进行边界条件处理,并进行三角分解,最后可得到全部位移值,其计算编程顺序结构的粗框图如图 3 所示。在计算时应注意前一小分块与后一小分块的叠加部分,在叠加时相对应的节点号才进行相加。在方程消元时,总体刚度矩阵与外部载荷列阵的变化规律应保持一致。

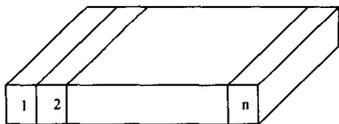


图 1 单元块的划分

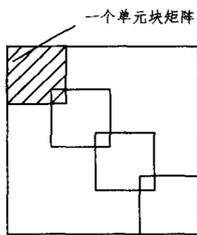


图 2 总体刚度矩阵中单元块刚度矩阵的位置排列

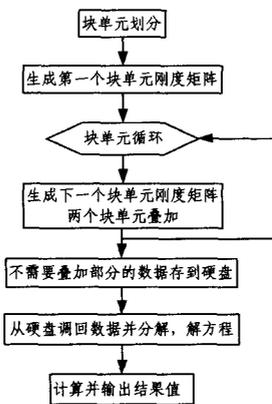


图 3 编程顺序粗框图

程序设计中,对内外存数据进行交换的语句采用流类库^[5]。使用 Ofstream 类将数据存贮到文件中。使用 ifstream 类从文件中提取数据。

给定允许内存并自动分块的程序语句如下:

```
for (i=1;i<4;i++)
```

```
{
    j=p/i;
    //p,q,r 分别为 x,y,z 方向的单元划分数目
    if (KDKM[3*(j+1)*(q+1)*(r+1)]<1000000)//所允许的内存容量
    //KDKM 为计算总体刚度矩阵主对角线上的元素在变带宽一维存贮中的位置的子程序
    {
        NT=i;//NT 为所分的块数
        break;
    }
}
```

3 算例

例 1 计算如图 4 所示结构。

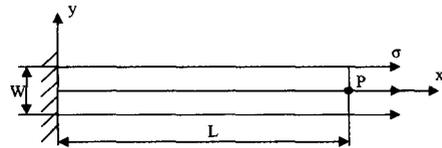


图 4 结构受力示意图

$L=5\text{mm}$, $W=4\text{mm}$, 板厚 $t=0.05\text{mm}$, 弹性模量 $E=200\text{GPa}$, 泊松比 $\mu=0.3$, 均匀分布拉应力 $\sigma=500\text{MPa}$ 。

求 P 点的位移。

用平面八节点等参元的有限元法进行计算。为了突出本文方法的特点,用 C++ 编写的有限元程序在 DOS 操作系统中运行,其仅可利用 640k 内存。在对结构进行单元划分时,为了便于将本文方法与三角解法进行比较,本文采取 y 方向单元划分固定时, x 方向单元划分不断加密。这样,在三角解法不能运行时,本文方法仍可以继续运行。

利用结构的对称性,取结构的上半部分进行单元划分。设 y 方向单元划分 10,当用三角解法计算时, x 方向单元划分最大值可到 18,而用本文方法单元划分可继续不断加密,同时运行时间也在增加,所以 x 方向单元划分应考虑计算精度与运行时间。不同计算方法的比较结果可见表 2。

表 2 不同计算方法的比较

	单元划分	计算结果	运行时间
三角解法	18×10	0.0123669mm	0.85s
本文方法	40×10	0.0123909mm	7.08s
解析计算		0.0125mm	

结束语 本文针对有限元计算时遇到的大型线性方程组求解问题,提出一种解决方法。这种 LDLT 分块解法可以解决大型线性方程组求解时遇到的内存容量不够的问题,并且可以提高计算效率。通过算例表明,这个算法是很有效的,可以为这类问题提供一种解决途径。

参考文献

- [1] 龙驭球. 有限元法概论[M]. 北京:人民教育出版社,1978:1-2
- [2] 王勖成,邵敏. 有限单元法基本原理与数值方法[M]. 北京:清华大学出版社,1988:262-292
- [3] 姜居诚. 有限单元法基础[M]. 北京:装甲兵工程学院出版社,1987:252-266
- [4] 刘跃进. 膜-基复合材料界面强度分析的影响系数法研究[D]. 北京:装甲兵工程学院,2003
- [5] 张国锋. C++ 语言及其程序设计教程[M]. 北京:电子工业出版社,1998:230-253