成绩

中国农业大学

课程论文

(2022-2023 学年秋季学期)

论文题目：有限元程序报告

课程名称：有限元法

任课教师：徐春晖

班　　级：工力 201

学　　号：2020310020119

姓　　名：张家瑞

目录

前言

有限元法作为求解工程数值问题的一种计算方法，对于处理力学问题中有着很重要的作用。通过连续体离散化的方式，将问题转化为方程进行求解，例如杆、梁、刚架单元中的线性方程组，传热、动力问题中的微分方程组。利用计算机我们便可以轻松求出问题的解。

衔接了上学期所学的计算方法课程，我们继续使用 FORTRAN 来完成有限元程序的编写。在前面的力学问题中，核心思想是“形函数——几何关系——物理关系——虚功原理”，通过这个思路，我们来处理杆、梁、刚架、平面单元等结构。后续又学习了传热、动力等问题的有限元求解方法，我们知道了有限元法是一种求解问题的方法，而不仅仅是只有力学问题才能用，可能是因为我们对力学问题比较熟悉才产生了这种误区。

在上这门课之前，我了解到求解 PDE 常用的三种方法：1950s 时的有限差分法、1960s 有限元法、1970s 的谱方法。当时我就对有限元法这门课充满了期待，果然通过这门课的学习，真的是收获满满，对数值计算有了更加深刻的了解，多门学科都串联在了一块：比如材料力学中的矩阵位移法、机械振动中的刚度矩阵和质量矩阵等，也学到了很多的知识。

本报告主要分为 4 个部分：首先是网格划分，有了这个程序后，后续进行求解较大的问题时，不需要一个个数节点并逐个输入电脑了；接着是 FEP1 程序，在进行了个人的解读后，尝试对其进行改写，并做了一些算例分析，最后编写了一个去除奇异性的小程序，也为后面动力部分作了准备；然后是 FEP2 程序，进行了个人解读，并添加了梁单元与刚架单元，进行了一些算例计算；最后研究了动力系统问题。纵观整个程序，就会发现它是由浅入深，通过一步步添加东西，逐步完善的，我在编写动力系统部分也深有体会，在完善的过程中也要注意好主程序与子程序的对应，一定要小心谨慎，否则一个错误就要查很久。

主要内容有：对于 FEP1 与 FEP2 均进行了个人的分析，提出了一些看法；针对学长学姐报告和 FEP 全书中的一些问题进行了指出；对 FEP1 与 FEP2 进行了改写或扩展，FEP1 中改写了 4 节点单元，FEP2 中扩展了梁单元与刚架单元；编写了基本的动力学问题程序。

第一章 单元网格划分

最开始我们介绍单元体网格划分。具体为四节点四边形单元、三节点三角形单元、八节点四边形单元、厚壁圆筒等的划分。

网格划分是我们进行有限元法学习练习的第一个程序。以平面长方形板块为例，输入对角线两端点的横、纵坐标，以及在横、纵方向划分的网格数，我们可以利用不同的划分形式将板块划分为一定的小单元，进而输出每个节点的坐标和每个小单元的信息，即节点标号。我们在有了这些信息之后，可以将这些数据作为输入，结合单元的约束、载荷、性质等信息，利用虚功原理得到待求的节点位移（这些将在后续课程的程序中实现），即为后续程序 FEP1 做准备。所以网格划分是我们学习过程中的一个最基础的程序，下面将对程序进行介绍。

1.1 四节点四边形单元

在这几个单元节点划分中，四节点四边形单元最为基础。由于这几个单元节点划分情况很类似，所以我采用了主程序与子程序，这样对于程序编写做了很大的简化，特别是部分网格划分具有相同的子程序。将输入数据文本放在与源代码同一目录下，通过以下程序来实现数据的输入与输出，这样就可以直接将复制运行结果作为 FEP1 程序的输入。

```
LOGICAL EXFIL
INQUIRE(FILE=’POUFEN.OUT’,EXIST=EXFIL) !判断文件是否存在,读入数据
IF(EXFIL)THEN
OPEN(6,FILE=’POUFEN.OUT’,STATUS=’OLD’)
ELSE
OPEN(6,FILE=’POUFEN.OUT’,STATUS=’NEW’)
END IF
OPEN(5,FILE=’FEP0.txt’,STATUS=’OLD’)
READ(5,*) X1,Y1,X2,Y2,M,N
```

我将整个程序的子程序主要分为三大部分：节点坐标的计算、单元节点的划分、初始数据节点坐标转换。下图为节点坐标运算的部分程序，我使用的方法是利用三个循环变量，借助相同行或列的纵坐标或横坐标分别相等，对节点的 X、Y 坐标进行赋值。

```
SUBROUTINE GRID(G,X1,X2,Y1,Y2,M,N) !进行节点坐标的计算
DIMENSION G(2,1000)
IJ=0
DX=(X2-X1)/M
```

```
DY=(Y2-Y1)/N
DO J=1,N+1
DO I=1,M+1
IJ=IJ+1
G(1,IJ)=X1+DX*(I-1) !对网格点的 X 赋值
G(2,IJ)=Y1+DY*(J-1) !对网格点的 Y 赋值
END DO
END DO
END
```

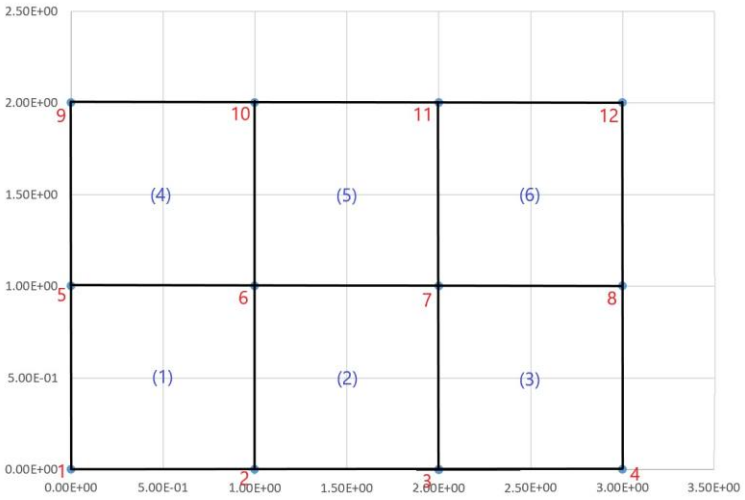
下图为各单元节点信息的计算，这个地方我认为是程序的核心点和难点，我通过利用整型数的整除运算法则和取余函数，计算出各单元的节点信息。

```
SUBROUTINE UNIT(U,M,N) !进行单元划分的计算
DIMENSION U(4,1000)
IUN=M*N
DO J=1,IUN
U(1,J)=(J-1)/M*(M+1)+MOD(J-1,M)+1
U(2,J)=(J-1)/M*(M+1)+MOD(J-1,M)+2
U(4,J)=((J-1)/M+1)*(M+1)+MOD(J-1,M)+1
U(3,J)=((J-1)/M+1)*(M+1)+MOD(J-1,M)+2
END DO
END
```

在单元信息输入的时候，考虑到输入数据时候的不确定性，例如，当我们输入坐标为(0, 3)和(3, 0)时，我们也需要使程序可以进行计算，所以我编写了下面的子程序，进行坐标的转换，原理很简单，实际上是只运用了矩形的长、宽以及坐标计算。

```
SUBROUTINE POINT(X1,X2,Y1,Y2) !考虑输入点的相对位置并做转化
IF (X1.GT.X2) THEN
X=X2,Y=Y2,X2=X1,Y2=Y1,X1=X,Y1=Y !请在真正运行时进行分行
END IF
IF (Y1.GT.Y2) THEN
Y=Y1-Y2,Y1=Y2,Y2=Y1+Y
END IF
END
```

我们选取数据进行计算，将“0，0，3，2，3，2”作为输入，运行结果如图所示。节点按照从下到上，从左到右的顺序进行编号。



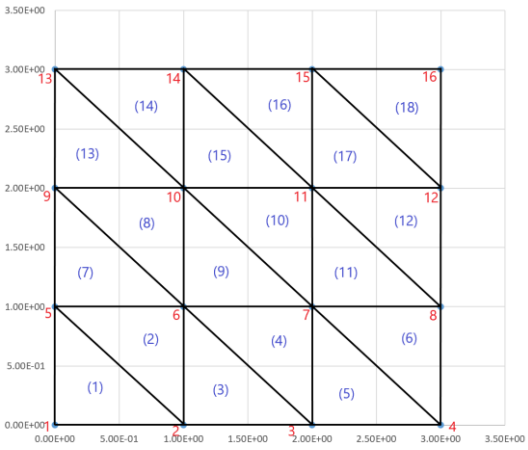
0.0000000E+00	0.0000000E+00		
1.000000	0.0000000E+00		
2.000000	0.0000000E+00		
3.000000	0.0000000E+00		
0.0000000E+00	1.000000		
1.000000	1.000000		
2.000000	1.000000		
3.000000	1.000000		
0.0000000E+00	2.000000		
1.000000	2.000000		
2.000000	2.000000		
3.000000	2.000000		
1.000000	2.000000	6.000000	5.000000
2.000000	3.000000	7.000000	6.000000
3.000000	4.000000	8.000000	7.000000
5.000000	6.000000	10.00000	9.000000
6.000000	7.000000	11.00000	10.00000
7.000000	8.000000	12.00000	11.00000

1.2 三节点三角形单元

在编写好四边形四节点单元后，三节点三角形单元就很简单了。我们只需要改动单元节点划分的程序即可。考虑到四边形单元与三角形单元的共性，所以我根据三角形单元编号的奇偶性进行了分开计算，在输出时候，依次交叉输出 U1 与 U2 中的数据即可。

```
SUBROUTINE UNIT(U1,U2,M,N) !进行单元节点的划分
DIMENSION U1(3,1000) !奇数单元的节点信息
DIMENSION U2(3,1000) !偶数单元的节点信息
IUN=M*N
DO J=1,IUN !计算奇数行的单元信息
U1(1,J)=(J-1)/M*(M+1)+MOD(J-1,M)+1
U1(2,J)=(J-1)/M*(M+1)+MOD(J-1,M)+2
U1(3,J)=((J-1)/M+1)*(M+1)+MOD(J-1,M)+1
U2(1,J)=(J-1)/M*(M+1)+MOD(J-1,M)+2
U2(2,J)=((J-1)/M+1)*(M+1)+MOD(J-1,M)+2
U2(3,J)=((J-1)/M+1)*(M+1)+MOD(J-1,M)+1
END DO
END
```

我们选取数据进行计算，将“0，0，3，3，3，3”作为输入，结果如图所示。节点按照从下到上，从左到右的顺序进行编号。




```
0.0000000E+00  0.0000000E+00
1.000000       0.0000000E+00
2.000000       0.0000000E+00
3.000000       0.0000000E+00
0.0000000E+00  1.000000
1.000000       1.000000
2.000000       1.000000
3.000000       1.000000
0.0000000E+00  2.000000
1.000000       2.000000
2.000000       2.000000
3.000000       2.000000
0.0000000E+00  3.000000
1.000000       3.000000
2.000000       3.000000
3.000000       3.000000
```

```
1.000000   2.000000   5.000000
2.000000   6.000000   5.000000
2.000000   3.000000   6.000000
3.000000   7.000000   6.000000
3.000000   4.000000   7.000000
4.000000   8.000000   7.000000
5.000000   6.000000   9.000000
6.000000   10.00000   9.000000
6.000000   7.000000   10.00000
7.000000   11.00000   10.00000
7.000000   8.000000   11.00000
8.000000   12.00000   11.00000
9.000000   10.00000   13.00000
10.00000   14.00000   13.00000
10.00000   11.00000   14.00000
11.00000   15.00000   14.00000
11.00000   12.00000   15.00000
12.00000   16.00000   15.00000
```

1.3 八节点四边形单元

对于八节点四边形单元，与四节点四边形单元相差最大的地方为它的不同奇偶行和列，拥有的节点数不同，因此这就需要我们考虑到行和列的奇偶性进行计算。

在节点坐标计算中，我分为了四次循环计算，即偶数行和奇数行的 X 和 Y 坐标计算，还是通过三个循环变量进行计算，下图中展示了偶数行网格点的 X 赋值。

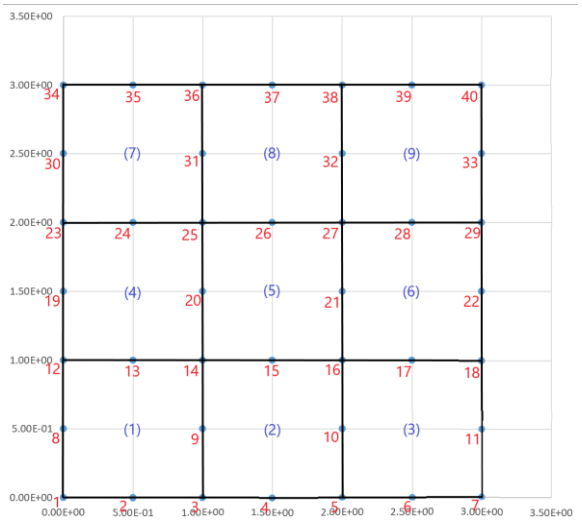
```
SUBROUTINE GRID(G,X1,X2,Y1,Y2,M,N) !进行网格划分的计算
DIMENSION G(2,1000)
IJ=0
DX=(X2-X1)/(2*M)
DO J=1,2*N+1,2 !对偶数行网格点的 X 赋值
DO I=1,2*M+1
IJ=IJ+1
G(1,IJ)=X1+DX*(I-1)
END DO
IJ=IJ+M+1
END DO
```

对于单元信息，利用整除和求余，得出每个单元的节点信息。这个地方是八节点单元的难点，需要进行仔细考虑。在此子程序中，是按照从下到上，从左到右进行依次编号的，若需要不同的节点编号方式，在程序中直接进行命名更改就可以。在括号中写入所需的判断条件、“THEN”后写出所需执行的操作即可。

```
SUBROUTINE UNIT(U,M,N) !进行单元划分的计算
DIMENSION U(8,1000)
IUN=M*N
```

```
DO J=1,IUN
U(1,J)=(J-1)/M*(3*M+2)+2*MOD(J-1,M)+1
U(2,J)=(J-1)/M*(3*M+2)+2*MOD(J-1,M)+2
U(3,J)=(J-1)/M*(3*M+2)+2*MOD(J-1,M)+3
U(7,J)=((J-1)/M+1)*(3*M+2)+2*MOD(J-1,M)+1
U(6,J)=((J-1)/M+1)*(3*M+2)+2*MOD(J-1,M)+2
U(5,J)=((J-1)/M+1)*(3*M+2)+2*MOD(J-1,M)+3
U(8,J)=((J-1)/M+1)*(2*M+1)+(J-1)/M*(M+1)+MOD(J-1,M)+1
U(4,J)=((J-1)/M+1)*(2*M+1)+(J-1)/M*(M+1)+MOD(J-1,M)+2
END DO
END
```

我们选取“0，0，3，3，3，3”作为输入，得到部分运行结果如图所示。



1.000000	2.000000	3.000000	9.000000	14.00000
13.00000	12.00000	8.000000		
3.000000	4.000000	5.000000	10.00000	16.00000
15.00000	14.00000	9.000000		
5.000000	6.000000	7.000000	11.00000	18.00000
17.00000	16.00000	10.00000		
12.00000	13.00000	14.00000	20.00000	25.00000
24.00000	23.00000	19.00000		
14.00000	15.00000	16.00000	21.00000	27.00000
26.00000	25.00000	20.00000		
16.00000	17.00000	18.00000	22.00000	29.00000
28.00000	27.00000	21.00000		
23.00000	24.00000	25.00000	31.00000	36.00000
35.00000	34.00000	30.00000		
25.00000	26.00000	27.00000	32.00000	38.00000
37.00000	36.00000	31.00000		
27.00000	28.00000	29.00000	33.00000	40.00000
39.00000	38.00000	32.00000		

1.4 厚壁圆筒单元

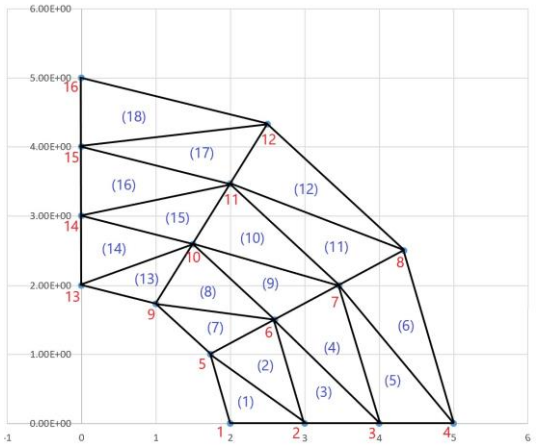
轴对称厚壁圆筒问题是我们可以得到解析解的问题之一，可以利用此模型对后续程序进行检验。我们根据轴对称性，考虑 1/4 厚壁圆筒。其实对于该模型，当进行三角形单元划分时，

与前文中的三节点三角形单元有很大的相似之处，即单元编号是一样的，我们仅需另计算出单元节点坐标即可。我们可以利用三角变换来实现节点坐标这个转换，需要注意的是 π 要进行定义，如下：

```
SUBROUTINE GRID(G,R0,R1,M,N) !进行网格划分的计算
DIMENSION G(2,1000)
IJ=0
```

```
DR=(R1-R0)/M
DT=3.1415926535/2/N
DO J=1,N+1
DO I=1,M+1
IJ=IJ+1
G(1,IJ)=(R0+DR*(I-1))*COS((J-1)*DT) !对网格点的 X 赋值
G(2,IJ)=(R0+DR*(I-1))*SIN((J-1)*DT) !对网格点的 Y 赋值
END DO
END DO
END
```

在我们输入“2，5，3，3”，依次为内半径、外半径、轴向划分数、角度划分数，得到输出结果如图 4 所示。我们可以看到，对于竖直方向上的最左端面，其横坐标都是在 10−7 量级，所以该程序的精度还是很高的。



2.000000	0.0000000E+00
3.000000	0.0000000E+00
4.000000	0.0000000E+00
5.000000	0.0000000E+00
1.732051	1.000000
2.598076	1.500000
3.464102	2.000000
4.330127	2.500000
0.9999999	1.732051
1.500000	2.598076
2.000000	3.464102
2.500000	4.330127
-8.7422777E-08	2.000000
-1.3113417E-07	3.000000
-1.7484555E-07	4.000000
-2.1855695E-07	5.000000

1.000000	2.000000	5.000000
2.000000	6.000000	5.000000
2.000000	3.000000	6.000000
3.000000	7.000000	6.000000
3.000000	4.000000	7.000000
4.000000	8.000000	7.000000
5.000000	6.000000	9.000000
6.000000	10.00000	9.000000
6.000000	7.000000	10.00000
7.000000	11.00000	10.00000
7.000000	8.000000	11.00000
8.000000	12.00000	11.00000
9.000000	10.00000	13.00000
10.00000	14.00000	13.00000
10.00000	11.00000	14.00000
11.00000	15.00000	14.00000
11.00000	12.00000	15.00000
12.00000	16.00000	15.00000

第二章 FEP1 程序的分析、改写及拓展

FEP1 是我们学习的第一个可以用于初步计算单元位移和载荷的有限元程序，以平面单元应力问题为例，我们可以通过输入材料信息、单元网格划分信息、载荷和约束信息，进而求解出每个节点的位移和每个单元网格的应力情况。当然与 FEP2 相比，FEP1 所能够解决的问题就有很大的局限性了。在编写的过程中我们也发现了在 FEP1 程序中，选取四节点单元与三节点单元相比有较大的误差，我们可以通过加密网格划分来提高精度，具体就运用到了第一次课程中输出的网格划分信息，在面对精细网格划分时，这就大大减轻了我们输入的工作量。

由于 FEP1 程序解读以及改写部分较长，所以在此我就挑选我觉得重点的部分进行阐述，在介绍的过程中，我也说明编程时候遇到的一些问题以及解决方法。限于篇幅，在此只展示出部分程序，为了缩减篇幅，部分程序采用了简化处理。

2.1 FEP1 程序的解读与分析

FEP1 程序分为了两个源程序文件进行存储，我第一次接触时不知道如何去运行，在经过摸索尝试后发现，我们在打开其中一个文件进行运行后，通过菜单栏打开另外一个文件，再进行一次运行，就可以把两个文件添加在一个“项目”中了。我认为整个程序围绕着 F = Ka 来展开了，最先利用网格划分等条件求出总刚 K，然后根据载荷、自身重力等求解出总载荷 F，再利用约束条件进行去奇异性，求解线性方程组，得到位移 a，最后利用位移与应力之间的关系，得到单元应力值。由于该程序较长，所以通过两个文件进行存储便于我们进行查阅，同时也方便我们对程序进行分类：主程序、主控程序和子程序。主程序主要是进行参数个数的读入和输出，并利用参数个数进行运行内存的预估，最后调用主控程序。

在主控程序中，对变量进行定义，再次调用读入文件，输入材料参数、约束、载荷等数据。再根据输入的信息，判断为平面应力问题还是平面应变问题，若为后者，则只需根据弹性力学知识对杨氏模量和泊松比进行变换。接下来即为程序的核心部分：单刚的计算和总刚的组集。首先遍历划分的单元，通过调用子程序 DIV 与 AXY，调出对应单元的节点信息、坐标信息、面积等。然后调用 STIFF 子程序，根据课本 75 页的单元刚度矩阵进行循环计算，得到单元刚度矩阵 Krs，然后根据单刚在总刚中的位置进行组集，此处需要注意对于平面单元，每个节点有两个自由度，所以 Krs 为二阶方阵，所以进行总刚组集时，我们要利用一个 Krs 在总刚中会占据两行两列这个特点进行组集程序的构造。在我们得到总刚 K 后，我们需要构造出载荷 F，通过在输入文档中对编号和对应载荷大小进行转化，得到载荷向量，若考虑体力，则根据密度将重力平均在每个节点上。之后，就开始求解位移 a，根据约束去除奇异性，求解出各点的位移。最后利用位移与应力的关系，求出应力。

对于子程序，主要是行使一些单一的功能：SETEM 判断预置内存是否够用；PZERO 用于矩阵清零；DIV 用于取出对应单元的节点信息；AXY 用于取出对应单元的节点坐标和计算出单元面积；STIFF 根据课本上推导的公示进行单元刚度矩阵的计算；GS1 是利用高斯消元法求解线性方程组。对于程序中的一些细节，例如桌面输出、矩阵清零等不在此阐述。

我对 FEP1 的第一印象就是程序好长，但我发现，按照“主程序——主控程序——子程序”顺序理清思路，抓住重点，整个程序就变得简单易懂了。我想，在阅读程序的过程中，构造出一个程序流程图，在上面标出各子程序的作用、各变量的意义，这样对于我们阅读程序有着很大的帮助。

2.2 FEP1 程序由三节点到四节点的改写

原 FEP1 程序是针对三角形三节点单元的，在我通读程序后，为了加强对程序的理解，我尝试对程序进行改写。主要需要改写的是两个子程序，即为刚度矩阵求解部分，其余的只需要将与四节点单元进行修正对应参数数目即可。

对于 DIV 程序，我们需要添加一个参数来与四节点匹配，注意不要有变量名重复，在此添加了“ME”来表示单元的第四个节点。

```
SUBROUTINE DIV(N,IX,ME) !取出对应单元的节点信息
COMMON/IJM/II,JJ,MM,MN !此处使用 MN 作为第四个节点，因避重复选取 MN
INTEGER IX(4,*),ME(4)
ME(1)=IX(1,N) ME(2)=IX(2,N) ME(3)=IX(3,N) ME(4)=IX(4,N) !程序运行时注意分行，此处为为减少本文篇幅
II=ME(1) JJ=ME(2) MM=ME(3) MN=ME(4) !注意分行
RETURN
END
```

对于 AXY 程序，主要是将四边形单元进行转化，利用缩放将长和宽均变换为 2，对于转化后的单元坐标，直接按照 1、-1 等进行赋值。

```
SUBROUTINE AXY(XC,B,C,DXT,DYT,AE) !取出坐标单元的信息
COMMON/IJM/II,JJ,MM,MN
DIMENSION XC(2,*),B(4),C(4)
REAL DXT,DYT
XI=XC(1,II) XJ=XC(1,JJ) XM=XC(1,MM) XN=XC(1,MN) !注意分行，下同
YI=XC(2,II) YJ=XC(2,JJ) YM=XC(2,MM) YN=XC(2,MN)
B(1)=-1 C(1)=-1 B(2)=1 C(2)=-1 B(3)=1 C(3)=1 B(4)=-1 C(4)=1
DXT=(XJ-XI)/2.0 DYT=(YM-YJ)/2.0
AE=(XJ-XI)*(YM-YJ) RETURN
```

```
END
```

接下来需要修改的是单刚计算子程序 STIFF，我们只需按照课本上 81 页的公式进行修改即可。在这里面积计算我选择了保留，因为当我们考虑体力时，需要用到面积进行计算。

```
SUBROUTINE STIFF(N,PRT,NR,NS,KRS,B,C,E0,T0,XNU,DXT,DYT,AE)
REAL KRS(2,2),B(4),C(4)
REAL DXT,DYT,XY,YX
LOGICAL PRT
BR=B(NR) BS=B(NS) CR=C(NR) CS=C(NS) !注意分行
H=(1.0-XNU)/2.0
ET=E0*T0/4.0/(1.0-XNU*XNU)
XY=DXT/DYT
YX=DYT/DXT
KRS(1,1)=ET*YX*BR*BS*(1+1.0/3.0*CR*CS)
KRS(1,1)=KRS(1,1)+ET*(H*XY*CR*CS*(1+1.0/3.0*BR*BS))
KRS(1,2)=ET*(XNU*BR*CS+H*BS*CR)
KRS(2,1)=ET*(XNU*CR*BS+H*BR*CS)
KRS(2,2)=ET*(XY*CR*CS*(1+1.0/3.0*BR*BS))
KRS(2,2)=KRS(2,2)+ET*(H*YX*BR*BS*(1+1.0/3.0*CR*CS))
IF(PRT) WRITE(*,100) N,NR,NS,KRS
100 FORMAT(' ELEM=',I3,' R=',I2,' S=',I2,2X,4E12.5)
RETURN
END
```

对于 FORTRAN 的输出命令 FORMAT，其中“I3”是指以 3 个字符的宽度来输出整数；“2X”是指输出位置向后移动 2 位；“E12.5”是指用科学计数法，以 12 个字符宽来输出浮点数，小数部分占 5 个字符宽。

在做了这些修改后，整个 FEP1 就已经修改完成了。在我修改的时候也遇到了一些问题：在第一次改写后，我发现我的刚度矩阵中出现了“NaN”，然后我修改输出程序，将 KRS 依次输出，发现问题出在了“XY”与“YX”上面，其输出结果为 0，然后我发现是前后数据类型定义的原因，我发现当长和宽之比只有定义为整型数时，才能够输出正确的结果，但这显然是不合理的。在仔细检查后，我发现问题出现在子程序调用变量名上，前后的变量名没有一一对应，导致了这个错误，现在已经解决。所以在改写程序时候，要总揽全局，后面子程序修改的变量名，在主控程序中调用时也要进行对应的修改，考虑到每个可能存在的问题，尤其是“IN”规则，我看到也有不少问我问题的同学，程序运行结果错误都是因为忘记了“IN”规则，导致一些变量定义为整型数。

个人想法：在改写的时候，我们首先要知道的就是三节点与四节点的不同之处：三节点

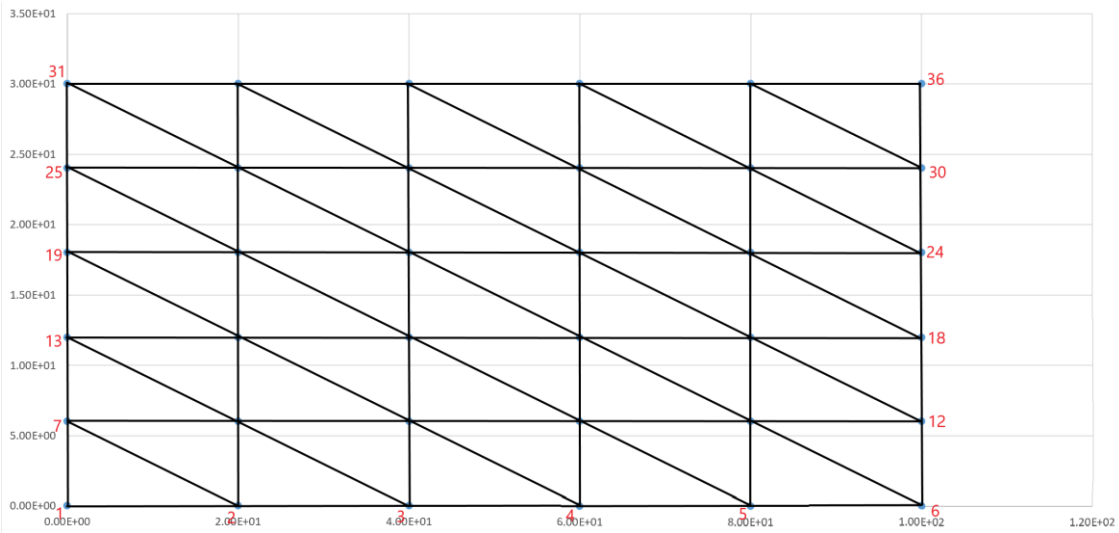
单元为三个节点信息，而四节点单元为四个，并且单刚求解公式也不同，所以我们只要抓住这些主要矛盾，进行修改即可。我们在修改时候可以灵活运用电脑快捷键，例如“Ctrl+F”，这个可以方便我们快速查找到某个变量所在的位置，也可以用来查阅变量名是否重复。在我们发现某个输出结果有问题时，我们可以调整输出数据，或者注释掉一部分内容，这样可以缩小检查空间，直至找到问题所在。

2.3 FEP1 程序算例分析

在此我考虑的研究对象为受轴向拉伸的悬臂梁。其左端固定，针对平面应力问题，材料都选取以下参数：杨氏模量为 100，泊松比为 0.3，密度为 1，厚度为 1，长度为 100，宽度为 30，杆端受到 1000N 作用力。分为两部分进行考虑：

三节点三角形单元；网格划分：长度五份，宽度五份

考虑到将杆端力均匀分布到每个节点上，我们按照（100，200，200，200，200，100）这种方式进行分配，利用第一节课编写的网格划分程序，得到输入输出文件，网格划分如图所示。



部分输入文件如下：

36, 50, 12, 6

1, 100, 0.3, 1, 1

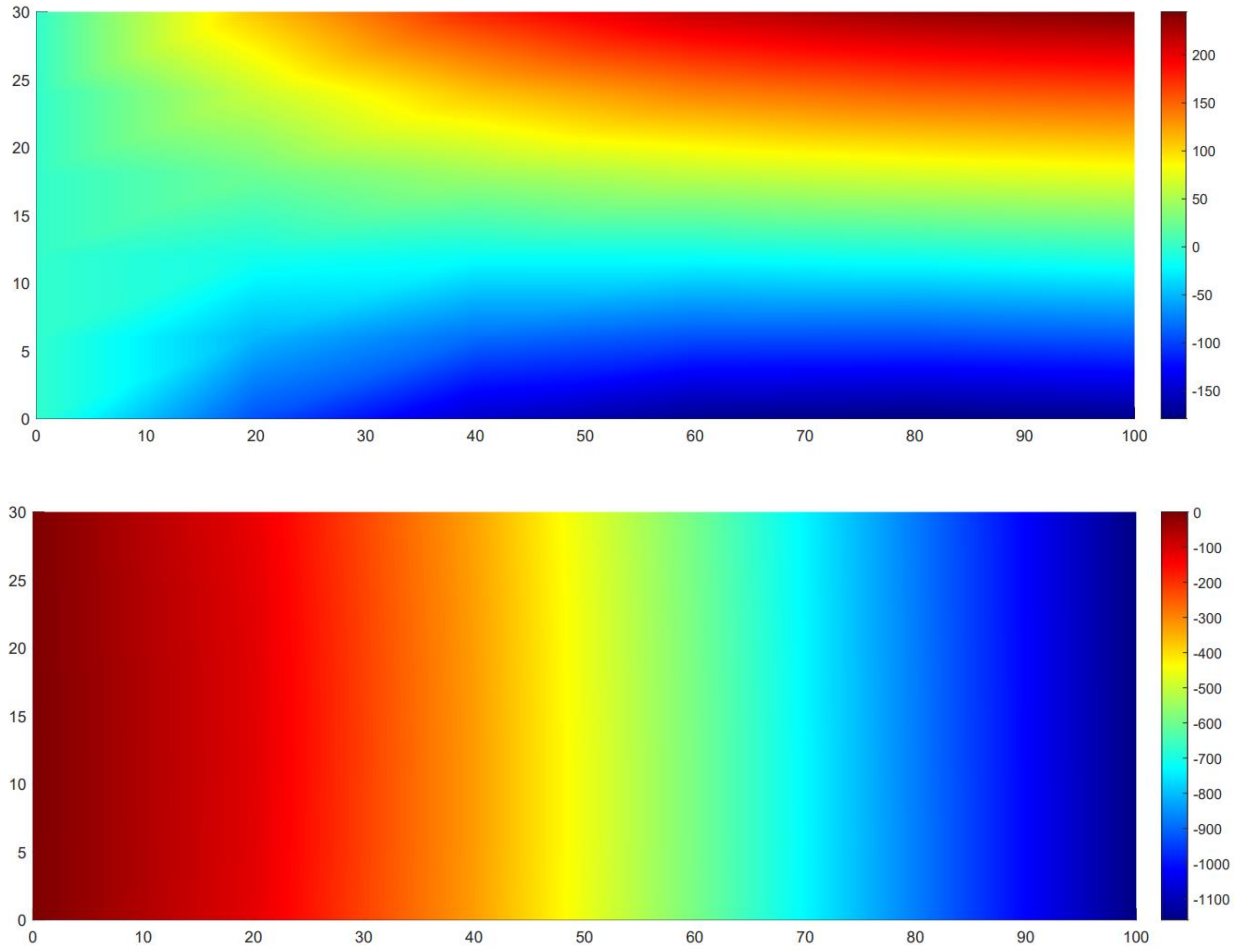
节点坐标信息

单元节点信息

1, 2, 13, 14, 25, 26, 37, 38, 49, 50, 61, 62

11, 100, 23, 200, 35, 200, 47, 200, 59, 200, 71, 100

由于我划分的网格较密，我使用第一次课的程序实现了网格信息的输入，感受到第一节课所做的工作为我带来了极大的便利，由于节点、坐标信息较多，所以不在此展示。由于结果很长，所以为缩减篇幅，在此仅展示我用 MATLAB 画的悬臂梁竖直方向与水平方向位移云图，结果如图所示。



四节点四边形单元；网格划分：长度五份，宽度五份

同上种情况，我按照同样的方法进行右端载荷分配，网格划分如图 3 所示。部分输入文件如下：

36, 50, 12, 6

1, 100, 0.3, 1, 1

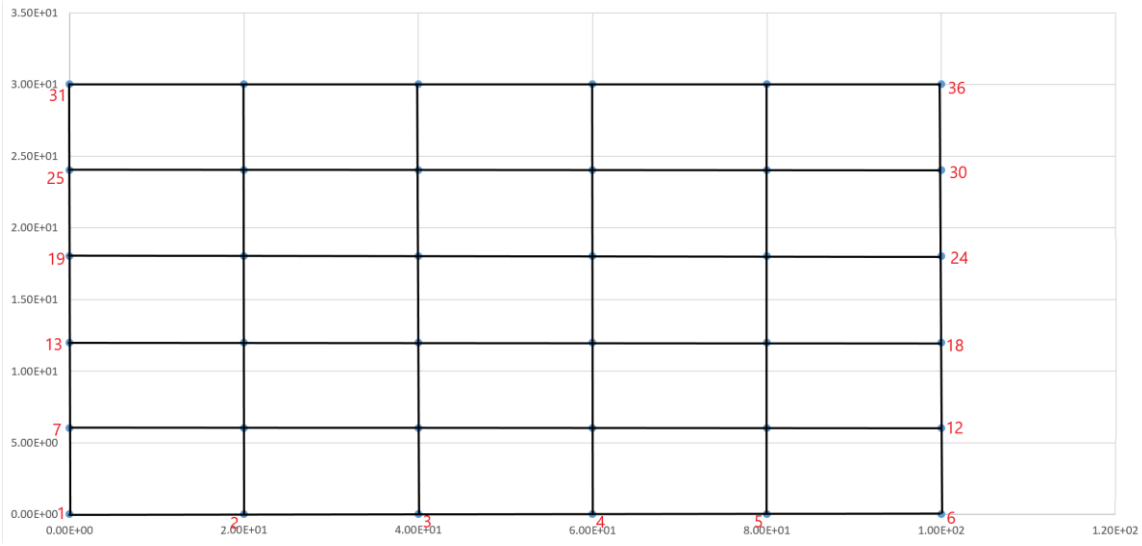
节点坐标信息

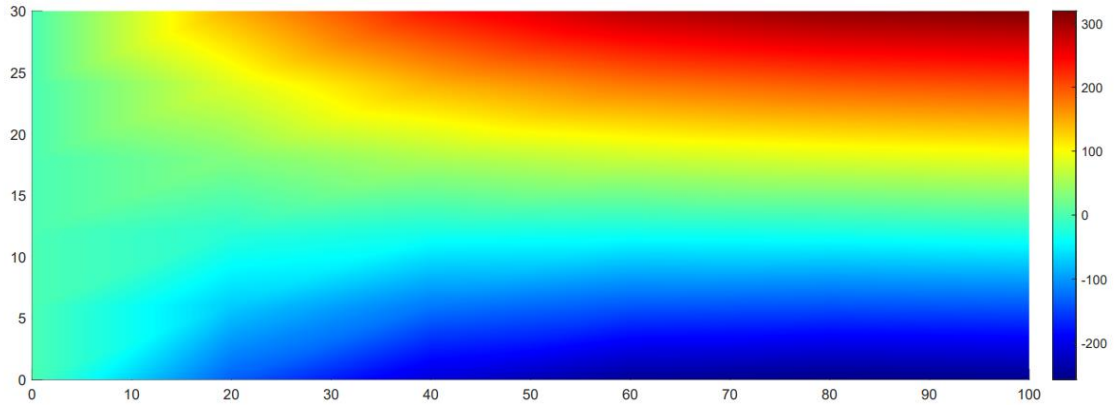
单元节点信息

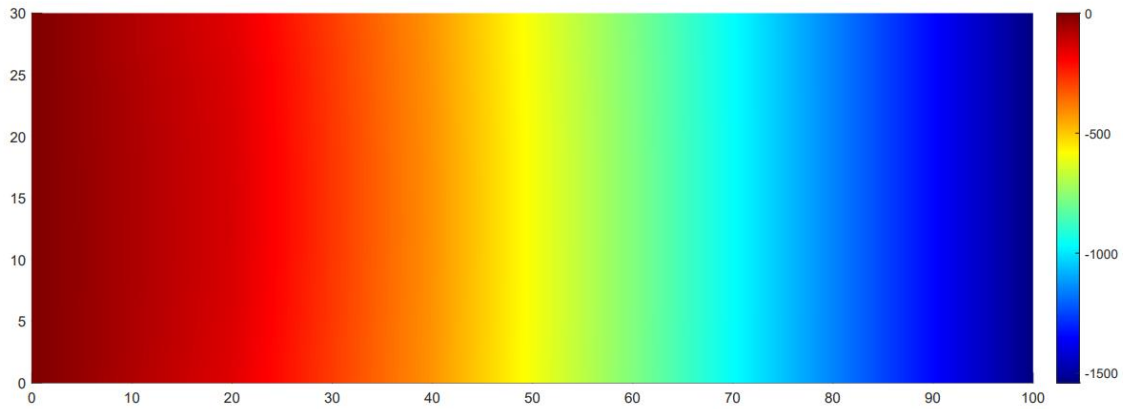
1, 2, 13, 14, 25, 26, 37, 38, 49, 50, 61, 62

11, 100, 23, 200, 35, 200, 47, 200, 59, 200, 71, 100

该输入节点信息，相对于前者，只需要对单元节点信息以及单元数进行更改即可。运行结果如图所示。







综合两次运算结果，我们发现两个结果差距较小，说明我们的改写以及计算成功。通过云图，我们可以看到该实验结果与我们预估结果很为相近：随着横坐标的增加，位移逐渐增大；由于重力的存在，在悬臂梁与支座连接处存在最大轴向应力。

在这部分中，画 MATLAB 云图花费了我一些功夫，通过网络上的搜索后，我使用的是 MATLAB 的 Patch 函数进行绘制，由于通过云图呈现结果更为清晰明了，所以不再展示数据结果。

2.4 FEP1 中去奇异化方法的改写

在 FEP1 中，我们为求解线性方程组，我们使用的是“赋 0 赋 1”法进行去除奇异性，我们还可以采用“降阶法”与“大数法”进行去奇异性。下面将展示两种方法。

降阶法

首先我们考虑降阶法，其本质是去除一个矩阵中的某行某列，我们通过一个简单程序来实现这个功能，在此参考了周天弈学长的降解法程序，但是他的程序编写不够简化，所以我对他的程序进行了改进。该程序本质上是通过已知要去除的行列信息来将矩阵往前提一位。考虑到行列循环，周学长是分别对行列进行循环来去除奇异性，我认为我们可以同时进行去除，所以只需要一次嵌套循环就可以完成，当刚度矩阵很庞大时，我的运算时间相比于周学长将会减少甚至一半。当然，我们还可以根据此程序为基础，实现把划掉的行和列移动至最后一行和列。该程序在后面的动力系统问题编写中要使用。我改写后的程序如下图所示：

```
PROGRAM JIANGJIE
DIMENSION KK(6,6),US(3) !在此定义总刚及要删除的行数列数
DO I=1,6 !输入总刚
READ(*,*) KK(I,1),KK(I,2),KK(I,3),KK(I,4),KK(I,5),KK(I,6)
END DO
DO I=1,3 !输入要删除的行数列数
READ(*,*) US(I)
END DO
DO I=1,3 !开始循环行数列数
K=US(I)
DO L=K-I+1,6-I
DO J=1,6
KK(L,J)=KK(L+1,J)
KK(J,L)=KK(J,L+1)
END DO
END DO
END DO
DO I=1,3 !输出去除奇异性后的总刚
WRITE(*,*) KK(I,1),KK(I,2),KK(I,3)
END DO
END
```

大数法

此方法相较于前者就简单很多了，我们只需将要去除奇异性的行列的主对角线上值改为一个很大的数即可。

```
400 DO 450 I=1,NU
K=US(I)
KK(K,K)=10000000000.
450 CONTINUE
```

由于大数法误差较大，直接影响到计算精度，并且运算结果不好，所以我们很少采用该方法；对于降阶法，会降低总刚阶数，运算速度上会有很大的提升，但是该方法在算法上较为复杂。所以我们在 FEP1 中采用的为改 1 改 0 法。

第三章 FEP2 程序的分析、改写及拓展

在学习并改写了 FEP1 程序后，我们不难发现该程序的局限性太大，例如：只能针对一种特定的模型进行分析、全部的单元与节点信息都需要放入输入文件、总刚度矩阵过大，能够求解规模小，计算速度慢等。因此我们需要进一步学习应用更加广泛的 FEP2，该程序适用面广，并且具有可扩充内容的接口，但其程序较长，因此我们需要理清其整体逻辑后进行精读，就可以方便我们进行理解。

由于 FEP2 程序解读以及改写部分较长，并且在老师的 FEP 全书中有逐行解释，所以在此我就挑选我觉得重点的部分进行阐述，在介绍的过程中，我也说明编程时候遇到的一些问题以及解决方法。限于篇幅，在此只展示出部分程序，为了缩减篇幅，部分程序采用了简化处理。

3.1 FEP2 程序的解读与分析

对于 FEP2 程序，该程序较长，不想让全文太长且毫无价值，在此解读部分只说明源程序位置，不放源程序，并分为以下三部分进行。

整体程序理解

FEP2 程序是全部放在一个文件中了，在我们阅读的时候，为避免前后拖动比较麻烦，我们可以把程序进行分块，模仿 FEP1 的样式，根据不同子程序的作用以及是否需要改动将该程序分为几个文件进行存放，可以方便我们进行查阅。程序的流程主要为以下部分：

（1）主程序 FEMED：进行输入输出文件的准备、调用主控程序；

（2）主控程序 PCONTR：信息的输入输出、调用子程序；

（3）内存分配计算 POINT：计算运算所需内存并进行内存分配；

（4）输入处理数据 INPUT：在该子程序中，通过进一步调用子程序，输入节点坐标 GENVEC、单元节点信息 ELDAT、材料数据 ELEMLIB、边界条件 BOUND、力和位移 GENVEC；

（5）变带宽设置 PROFIL：形成全局刚度矩阵的轮廓，形成变带宽刚度矩阵的地址；

（6）组集载荷矩阵 PLOAD；

（7）形成单刚并组集 PFORM(2)：根据功能参数 ISW=2 和单元类型 IEL，调用 ELEMLIB 中对应的功能，计算单元刚度矩阵并组集总刚 ADDSTF；

（8）三角分解 LDLT；

（9）前消回代求位移 FORBACK；

（10）输出位移 PRTDIS；

（11）计算单元内力和约束反力 PFORM(3)：根据功能参数 ISW=3 和单元类型 IEL，调用 ELEMLIB 中对应的功能，计算单元内力和约束反力 BASBLY；

（12）输出约束反力 PRTREAC。

在全局了解了整个程序之后，我发现原本很长的程序变得清晰明了，当我们想要对该程序进行拓展时，只需要抓住主要程序部分即可。对于内存分配、变带宽设置、组集载荷矩阵、三角分解、求解位移等过程，我们不需要进行改动，只用针对 ELEMLIB 进行修改即可。

所以我认为该程序中设计最为巧妙的是功能参数的引用，针对 PFORM 程序，它可以行使多种功能，当没有功能参数的时候，我们在主控程序编写时候，只能依次调用不同功能的子程序，当我们想到进行模块拓展的时候，比如加入梁单元，我们需要在子程序中的多个地方进行修改，并且还很容易出错或者遗漏，但是在引入功能参数 ISW 后，相当于把每种结构的特有部分进行打包，共同部分则公用，这样我们只需要在 ELEMLIB 中进行拓展，仿照已有的形式进行编写新的结构子程序即可。

输入信息解读

对于 FEP2，由于输入方式与 FEP1 有了很大的差别，因此在输入文件上也有所不同，并且在该程序中，使用线性插值实现的节点信息和单元信息的生成，所以输入程序本身也较为复杂，在此解读杆单元与平面单元的输入信息。

1、杆单元

（1）（主控信息）总节点数，总单元数，材料类型数，每个节点所具有的最大自由度，所计算的结构空间维数，每个单元所具有的最多节点数；

（2）（节点坐标）节点号，节点号增量，坐标；用（0,0,0,0）表示结束输入节点坐标；

（3）（单元信息）单元号，要连续生成的单元号数量，节点号增量，单元的材料号，单元格所包含的结点编号（逆时针排列）；用（0,0,0,0,0,0,0,0,0,0）表示结束输入单元节点信息；

（4）（材料信息）材料类型号，单元类型号；（此处要分行）弹性模量，截面积；

（5）（约束信息）节点号，相同约束的节点数，节点号增量，是否有横向约束，是否有纵向约束；用（0,0,0,0,0,0）表示结束输入约束信息；

（6）（载荷信息）节点号，节点号增量，横向载荷大小，纵向载荷大小；用（0,0,0,0）表示结束输入约束信息。

2、平面单元

由于杆单元和平面单元信息输入使用的是同一套子程序，因此大多数输入信息都是一致的，单元信息不相同，杆单元为两个节点信息，而平面单元为 3 8 个节点信息，这里根据实际情况进行输入即可。还有就是输入材料信息时不相同，我们也只需根据实际情况进行输入即可，在平面单元中为

（4）（材料信息）材料类型号，单元类型；（此处要分行）弹性模量，泊松比，沿一个方向的高斯积分点个数，平面应力（≠ 0）或平面应变（= 0）问题。

3、个人思考及指正

在平面单元输入节点信息时，对于六节点四边形单元，多出的两个节点是处于边上的对位位置，而我们在数单元节点编号的时候，仍需要考虑逆时针连续数，即按照八节点单元对应的顺序进行，那么此时我们在输入数据的时候，需要输入的是七个节点信息，在对应的第六个节点处写为 0 即可，在后续组集刚度矩阵和载荷向量时，编号为 0 的节点就不会再考虑了。针对输入信息时，节点坐标、单元节点、约束、载荷等信息的结尾需要一串零进行中断，这是由于这些信息我们使用的是插值输入的，只用给出需要插值点出的信息即可，显然无需给出每个节点对应的信息，因此不能利用主控信息作为循环次数，因此需要定义出一个中断指标，即一串零，需要注意的是，为保证信息输入正确，这一串零的个数是不能少于前面信息的个数的，多点的话是没有问题的。而对于材料信息，显然材料信息无法通过插值实现，因此需要输入所有材料对应的信息，所以我们在输入材料时，无需使用一串零进行截断，同时，需要注意的一点就是，在输入材料信息时候，我们需要进行分组分行处理，在材料信息部分，第一行为第一种材料的编号、类型号，第二行为该编号材料对应的材料信息，后面依次为第二种、第三种等材料的信息，这个地方在我们输入数据的时候是需要注意的。

需要注意的是，这里面的节点信息增量为相对应于前面的那个节点，两者之间的增量关系，而非全书中所说的与下个信息之间的增量关系，我认为 FEP 全书中这个地方是解释错误了，并且给的例子也是不正确的。假如给定的是第一条记录中增量 NG 非零，那么在进入判断语句后（GENV12），L 就没有一个给定的定义，显然此处是错误的。并且要想利用插值求解两个记录之间的数据，那么在输入第二个记录后，第二个记录中的 NG 会将第一个记录进行覆盖，所以我们需要给定的是相对于前者的增量关系。

重要子程序分析

对于 FEP2，其中有一些设计思路很好的子程序，我们在此挑选部分进行分析。

1、自由度分类

在 FEP2 中，有一个很重要的变量，即为 ID。是在 FEP2 中，我们是按照节点所受到的约束，将他们划分为活化自由度和约束自由度，并根据这重新赋予节点位移的 ID，也就是在原节点的基础上，按照原先的顺序，将可以移动的位移依次赋予一个正数，受到约束的方向赋予一个负数，这个过程对应的位置为 BOUND 和 PROFIL 子程序中。注意，活化自由度数加上约束自由度数是等于节点总共的自由度数的。这个设计对于我们后续进行求解有了很大的简化，在组集刚度矩阵的时候，我们根据其自由度进行不同的处理，进行分块存放，在我们求解位移时候，只需针对活化自由度部分进行求解，而在求解约束力的时候，也只需针对约束自由度部分进行求解即可。

在 FEP1 中，我们通过去除奇异性来求解出节点的位移，然后再求解约束力，但这种方式对于节点很多的时候，无论是运算速度还是存储空间上，都表现的不是很理想。而 FEP2 中的方法就有很大的优势了。

2、插值生成节点信息程序

在前面也讲到过了输入信息的方法，但为了了解为什么需要那样输入，那么我们需要进一步深入了解该子程序。在数据处理子程序 INPUT 中，我们共调用了五个子程序：读入节点坐标 GENVEC、读入单元类型 ELDAT、读入材料信息 ELEMLIB、读入约束信息 BOUN、读入载荷或位移信息 GENVEC。其中 GENVEC、ELDAT、BOUN 中使用了插值。我们以 GENVEC 为例，实现这个功能主要是利用一行信息中的第二个数据，即节点号增量实现的，通过前后两个节点的信息，利用后者的节点号增量，相对于前者进行插值处理，得到两个节点之间的节点信息。相关错误已在前面部分中指出，在此不做赘述。

3、四节点退化为三节点

在我们组集刚度矩阵的时候，前文中已经提到过，对于六节点单元中的零节点不参与计算。然而对于三节点单元，我们是利用四节点进行修正的，通过使用 SHP(I, 3) =SHP(I, 3) + SHP(I, 4)（SHAP13）进行合并，进而得到三节点结构，我认为这个地方的设计还是很巧妙的。

4、变带宽法存储

在单元、节点数很多的算例中，如果我们仍将所有程序都按照矩阵进行存储，那么这个工作量及所占用的空间是非常大的，因此我们充分利用单元刚度矩阵的特性，即通过合理的节点标号来减少刚度矩阵的带块，进而使用变带宽进行存储。通过引入一个指针向量，将矩阵对角线元素所对应的编号进行存储，具体实现是在 PROF44。

3.2 FEP2 程序梁单元的拓展

原 FEP2 程序中是有杆单元与平面单元的结构处理，在此我们进行拓展，加入梁单元。通过前面的分析，当结构改写为梁单元时，只需进行拓展一个 CASE3 即可，其余的部分无需改动。拓展后的程序如下所示：

```
SUBROUTINE BEAM(D,UL,XL,IX,S,P,NDF,NDM,ISW)
C.... TRUSS LINEAR ELASTIC ELEMENT ROUTINE
COMMON /ELDATA/ LM,N,MA,MCT,IEL,NEL,NST
DIMENSION D(10),UL(NDF,1),XL(NDM,1),IX(1),S(NST,1),P(1),DU(4)
C.... GO TO CORRECT ARRAY PROCESSOR
SELECT CASE (ISW)
C.... INPUT MATERIAL PROPERTIES
CASE(1)
READ(5,*) E,EI
D(1)=E*EI
WRITE(6,2000) E,EI
C.....CALCULATE ELEMENTAT STIFFNESS MATRIX
CASE(2,3)
```

```
DX=XL(1,2)-XL(1,1)
DY=XL(2,2)-XL(2,1)
DL=SQRT(DX*DX+DY*DY)
H2=12.0*D(1)/(DL*DL*DL)
H3=6.0*D(1)/(DL*DL)
H4=4.0*D(1)/DL
H5=2.0*D(1)/DL
S(1,1)=H2
S(2,2)=H4
S(1,2)=H3
S(2,1)=S(1,2)
S(1,3)=-H2
S(1,4)=H3
S(2,3)=-H3
S(2,4)=H5
S(3,3)=H2
S(3,4)=-H3
S(4,4)=H4
S(2,1)=S(1,2)
DO I=1,4
DO J=I+1,4
S(J,I)=S(I,J)
END DO
END DO
C.....CALCULATE ELEMENTAT NODAL FORCE
IF(ISW.EQ.3) THEN
DU(1)=UL(1,1)
DU(2)=UL(2,1)
DU(3)=UL(1,2)
DU(4)=UL(2,2)
IJK=NDF*2
DO I=1,IJK
DO J=1,IJK
P(I)=P(I)-S(I,J)*DU(J)
END DO
```

```
END DO
END IF
END SELECT
2000 FORMAT(/5X,'PLANE BEAM LINEAR ELASTIC ELEMENT'//
1 10X,7HMODULUS,E18.5/10X,4HAREA,E21.5/)
RETURN
END
```

在这个程序中，我只考虑了对于一个在节点处受到外载荷梁的结构，当然还有很多可以做的，比如梁的内力的计算、外面施加任意力系时候结构反力的计算等等。这些都是直接在该程序上面进行添加即可。

在编写这个程序的时候，虽然很简单，但是我也花了较长的时间，主要是最开始没搞清楚梁与杆的异同，在输入信息时候，我最开始假定输入的节点信息为节点的横向位移和转角，但显然这个想法是不正确的，由于在后面计算无需与初始输入信息进行过多的联系，只需要考虑梁的长度即可，所以我们在输入信息中只需输入梁节点的横、纵广义坐标位置即可，便利用这个信息进行计算梁的长度。并且在计算结构反力时候，我们需要利用单元刚度矩阵进行计算，这个与杆结构有着很大的差别，是因为杆的轴力是只沿轴向的，并且相对于杆的程序，在梁中，DU(2)需改为 DU(4)，最初我就是因为这个没有进行改正，导致不知道哪里出现了数组溢出，以至于查错浪费了很多时间，在仔细观察之后才发现了这个错误。

并且在此指出武琦学姐程序编写的错误：她在编写算例的时候，未能正确考虑力和位移之间的关系，在算例中，她把施加的 100 力写在了力矩的位置，我想她应该是按照前面杆结构的惯性思维来写了，并且得到的结果与书本上理论值相差很大。并且她在编写程序中，错误计算了梁的内力，她直接使用的是杆的内力计算公式，所以在她的结果中，没有结构反力的输出。并且她的算例与图片不匹配，按照她的算例，她是想要考虑梁的两端受到位移和转角的固定约束，但是她给出的图片里面，右端不受到转角的约束。但她的程序中还是有值得学习的地方的，比如在计算 H 时，她充分利用了 H 之间的关系，进行依次求解，这样可以简化运算。

我编写的梁单元程序算例将会在后面章节中展示。

3.3 FEP2 程序刚架结构的拓展

对于钢架结构，其本质上与杆单元、梁单元相同，我们只需要在程序中修改对应的刚度矩阵即可。我们只需要按照书本 44 页的公式，将钢架的整体坐标系下的单元刚度矩阵改写到对应的位置。另外需要注意的是，钢架的自由度要比梁的多两个，因为钢架体现在一个平面内，要有横纵两个方向的位移以及对应的转角。拓展后的程序如下所示：

```
SUBROUTINE RIGIDFRAME(D,UL,XL,IX,S,P,NDF,NDM,ISW)
 COMMON /ELDATA/ LM,N,MA,MCT,IEL,NEL,NST
 DIMENSION D(10),UL(NDF,1),XL(NDM,1),IX(1),S(NST,1),P(6),DU(6)
 SELECT CASE (ISW)
    CASE(1)
       READ(5,*) E,EA,EI
       D(1)=E*EA
   D(2)=E*EI
       WRITE(6,2000) E,EA,EI
    CASE(2,3)
       DX=XL(1,2)-XL(1,1)
       DY=XL(2,2)-XL(2,1)
       DL=SQRT(DX*DX+DY*DY)
   CC=DX/DL
   SS=DY/DL
H1=D(1)/DL
H2=12.0*D(2)/(DL*DL*DL)
H3=6.0*D(2)/(DL*DL)
H4=4.0*D(2)/DL
H5=2.0*D(2)/DL
S(1,1)=H1*CC*CC+H2*SS*SS
S(1,2)=(H1-H2)*SS*CC
S(1,3)=-H3*SS
S(1,4)=-S(1,1)
S(1,5)=-S(1,2)
S(1,6)=S(1,3)
S(2,2)=H1*SS*SS+H2*CC*CC
S(2,3)=H3*CC
S(2,4)=-S(1,2)
S(2,5)=S(2,2)
S(2,6)=S(2,3)
S(3,3)=H4
S(3,4)=-S(1,3)
S(3,5)=-S(2,3)
S(3,6)=H5
```

```
      S(4,4)=S(1,1)
      S(4,5)=S(1,2)
      S(4,6)=-S(1,3)
      S(5,5)=S(2,2)
      S(5,6)=-S(2,3)
      S(6,6)=H4
      DO I=1,6
      DO J=I+1,6
      S(J,I) = S(I,J)
      END DO
      END DO
     IF(ISW.EQ.3) THEN
      DU(1)=UL(1,1)
      DU(2)=UL(2,1)
      DU(3)=UL(3,1)
      DU(4)=UL(1,2)
      DU(5)=UL(2,1)
      DU(6)=UL(3,1)
      IJK=NDF*2 !NDF 单元最大自由度数为 2
       DO I=1,IJK
       DO J=1,IJK
       P(I)=P(I)-S(I,J)*DU(J)
       END DO
       END DO
       END IF
       END SELECT
2000    FORMAT(/5X,'PLANE RIGID FRAME LINEAR ELASTIC ELEMENT'//
        1 10X,7HMODULUS,E18.5/10X,4HAREA,E21.5/10X,7HINERTIA,E21.5)
       RETURN
       END
```

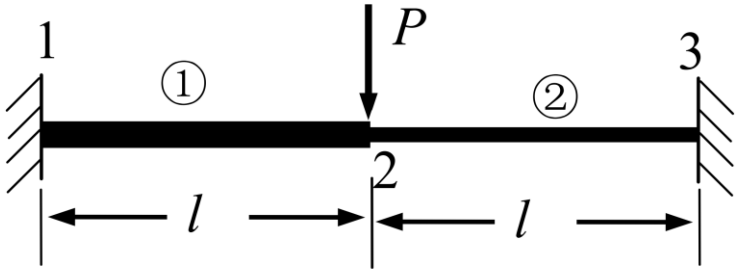
同样，在刚架单元中，我们可以添加一些考虑常规载荷等多种情况。在此需要注意的是，编写刚架单元中，由于不同于前面的杆单元和梁单元，刚架单元的节点自由度数与空间维度是不匹配的，因此需要将 GENVECW 子程序进行改动，即要将 NDM 改为 NDF，虽然平面刚架单元为一个二维问题，但是其自由度需要考虑转角，并且在外加载荷情况下，我们也要考虑作用在节点上的弯矩，因此这里必须进行改动，如下：

READ(5,*) N,NG,(XL(I),I=1,NDF)

3.4 FEP2 程序算例分析

梁单元算例分析

我们以书本上 29 页例题为算例，进行计算。



输入数据为

3,2,2,2,2,2 !主控信息

1,0,0,0 !节点信息

2,0,1,0

3,0,2,0

0,0,0,0 !节点信息输入终止

1,1,0,1,1,2 !单元信息

2,1,0,2,2,3

0,0,0,0,0,0 !单元信息输入终止

1,3 !材料 1 信息

100,2

2,3 !材料 2 信息

100,1

1,1,1,1,1 !约束信息

3,1,1,1,1

0,0,0,0,0,0 !约束信息输入终止

2,0,-100,0 !载荷信息

0,0,0,0 !载荷信息输入终止

我们要注意该例题中存在两种梁结构，因此需要考虑两种材料，它们具有不同的惯性矩。同时注意施加的 P 作用应该写在第一个位置。

最终程序的运行部分结果如下：

NODAL DISPLACEMENTS

NODE 1DISP 2DISP !节点位移结果

1 0.0000E+00 0.0000E+00

2 -0.3030E-01 -0.1515E-01

3 0.0000E+00 0.0000E+00

NODAL REACTIONS !结构反力结果

NODE 1 1DOF 54.5455

NODE 1 2DOF 30.3030

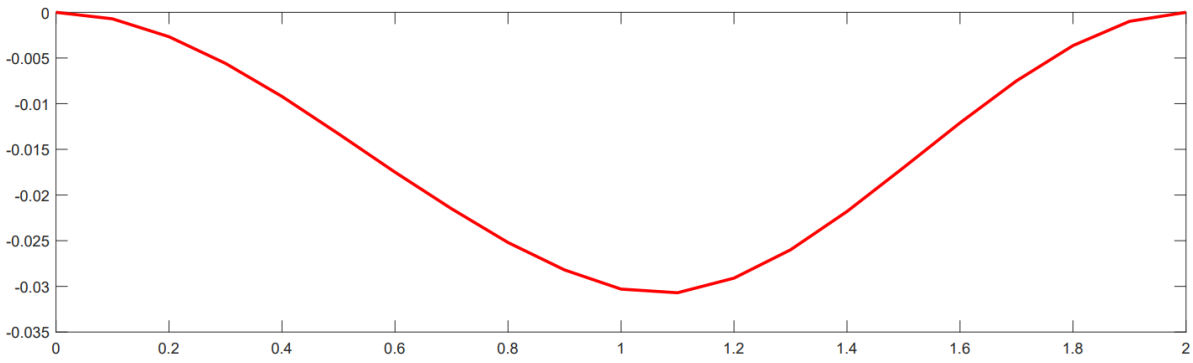
NODE 2 1DOF -100.0000

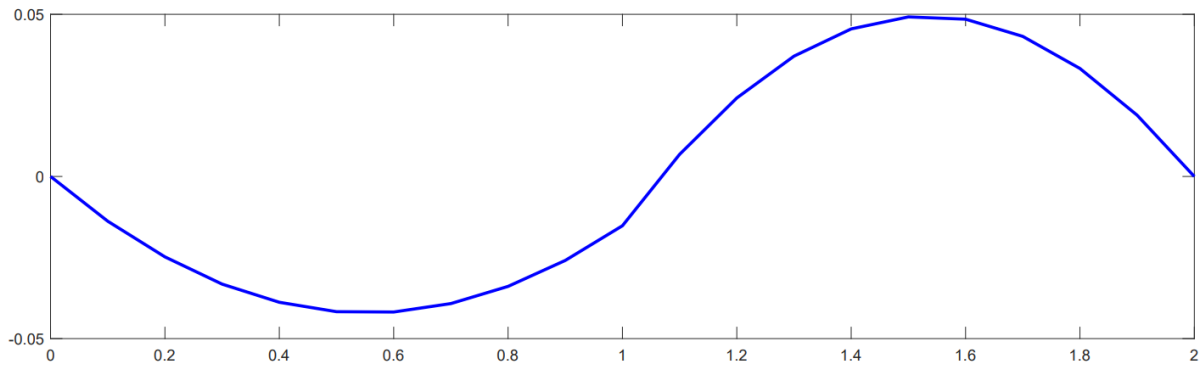
NODE 3 1DOF 45.4545

NODE 3 2DOF -21.2121

我们可以将这个结果与书本 29 页与 30 页结果比对，在 2 节点的位置，横向位移为 v2 =−0.03030，转角为 θ 2 = −0.01515，1 节点处结构反力为 Fy1 = 54.5455，M1 = 30.3030，2 节点处的结构反力为 Fy3 = 45.4545，M3 = −21.2121。与书本上的计算结果比较，可以发现完全符合理论值，所以该程序编写成功。

在编写了这个之后，我对这个结果并不满足，因为这个结果不够直观，不能以图片的形式将结果展示出来，我们不妨将该梁结构单元划分考虑的更细一些，这样可以做出精细化后的梁的每个地方的位移曲线，这样相当于求出梁的内力了，我们取 21 个节点和 20 个单元，运行结果如图。第一个为梁中竖直方向位移的图像，第二个为梁转角的图像。

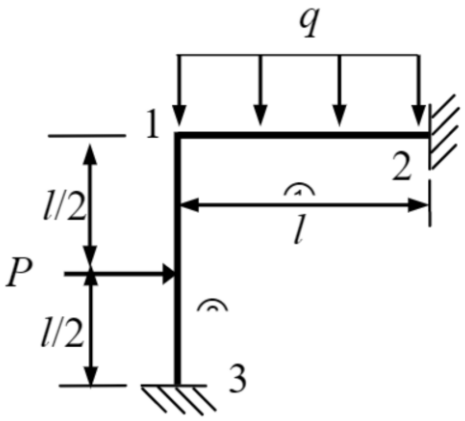




通过这个图我们发现，对于右边较细部分，梁的横向位移曲线显得更“陡”一些，这与我们的材料力学中知识吻合，并且梁的转角也是正确的，说明该程序的精度是可靠的，并且编写正确。

刚架单元算例分析

我们以书本 49 页的例题 2.15 为算例，进行计算。



需要注意的是，要将外加荷载进行等效处理，转化为作用在 1 节点上的力，经过运行后得到的部分输出结果如下：

NODAL DISPLACEMENTS

NODE	1DISP	2DISP	3DISP
1	0.1491E+01	-0.3637E+01	-0.2178E+01
2	0.0000E+00	0.0000E+00	0.0000E+00
3	0.0000E+00	0.0000E+00	0.0000E+00

经过与书本上的结果进行比对，我们发现该运算结果吻合，2、3 两节点位移为 0，表明程序正确。需要注意的是，在前面也已经提到过，因为刚架结构的节点自由度数与维数不同，因此在输出格式上也需要进行更改，确保能够将三个位移进行输出。

第四章 动力学问题

在前面的章节中，我针对于杆、梁、刚架、平面等单元进行了程序的编写与分析，学习并编写了单元网格划分程序、FEP1 和 FEP2。在对有限元有了一定的了解之后，仅满足于此是不够的，因此我尝试进行动力学问题程序的编写。

4.1 动力学问题的分析

对于动力系统的分析问题，即所受外力为变化、考虑惯性力的作用，研究结构的运动。整个流程在课本 188 页和 189 页，我们只需按照课本上的步骤进行编写即可。

首先要根据已知结构，求解出刚度矩阵、质量矩阵与阻尼矩阵；给定初始的位移和速度，利用专门的起步方法，求解出初始加速度；再给定时间步长，计算有效质量矩阵等需要的参数，最后进行迭代即可求出每个时间点的位移。还可以利用前后时间点的位移，根据需要求出速度、加速度和应力等信息。总之是化为一个二阶线性微分方程组并运用中心差分法进行求解。

在了解了基本原理之后，我考虑到 FEP2 程序比较大，修改起来比较麻烦，因此我直接以三节点三角形单元的 FEP1 为基础进行编程。

我们需要编写一个子程序来进行质量矩阵的计算，考虑到后续需要求解质量矩阵的逆矩阵，因此采用了集中质量矩阵，这样使得矩阵对角化，方便求解其逆矩阵，也大大降低了运算量，并且在书本中也有提到，这样的计算结果相比于使用协调质量矩阵不会相差太多。

在程序编写过程中，也需要考虑矩阵乘法的运算，我们可以编写一个小程序来进行矩阵乘法的计算，但考虑到我们已知矩阵的阶数，可以直接在主控程序中可以引入一个循环直接进行计算，因此就没有设计子程序。

在改程序中，也引入了一个 NJ22，为 NJ2-NU，表示活化自由度数。这是由于在动力学问题中，我们不能像之前的 FEP1 程序中，采用化 0 化 1 法来进行去除奇异性，要采用降阶法来处理，这也就用到了前面我编写的子程序。而由于后续需要进行很多的矩阵乘法运算等，所以我直接引入了 NJ22 来表示活化自由度数。需要注意的是，在我的程序中，为了表示方便，我用 WM、KK 等表示降阶前的质量矩阵、刚度矩阵，用 WMM、KKK 来表示降阶后的质量矩阵、刚度矩阵。

考虑到输入格式的简洁，我在程序中默认初始位移与速度为 0，并且施加的载荷是在 t=0 时为 0，在 t>0 时为一个常数，即为输入的荷载。并且删去了考虑重力因素作用的部分。如有需要，这些都是可以直接在程序中进行简单修改的，只需要添加一个输入窗口，将初始位移、速度输入进去即可。

我在编程中也遇到了很多问题，比如我在分配内存时候，直接把几个数组分到到同一个块上了，导致这几个数组的结果是一样的，当时也花费了很长时间查到错误。

4.2 基于 FEP1 的动力学拓展

单元质量矩阵计算子程序如下，整体与单元刚度矩阵类似，只需要按照集中质量矩阵进行编写即可：

```
SUBROUTINE WEIGHT(N,PRM,NR,NS,MRS,B,C,E0,T0,XNU,AE,RO)
   REAL MRS(2,2),AE,T0,RO
   LOGICAL PRM
IF(NR.EQ.NS)THEN
MRS(1,1)=AE*T0*RO/3.0
MRS(2,2)=AE*T0*RO/3.0
MRS(1,2)=0
MRS(2,1)=0
ELSE
MRS(1,1)=0
MRS(2,2)=0
MRS(1,2)=0
MRS(2,1)=0
END IF
   IF(PRM) WRITE(*,100) N,NR,NS,MRS
100   FORMAT(' ELEM=',I3,' R=',I2,' S=',I2,2X,4E12.5)
   RETURN
   END
```

总质量矩阵组集：

```
PRM=.FALSE.
   WRITE(*,'(A\)')' DO YOU WANT TO OUTPUT THE ELEMENT WEIGHT SUB
  1MATRIX ? (Y/N) ' !是否想要输出质量矩阵
   READ (*,'(A)') YN
   IF(YN.EQ.'Y'.OR.YN.EQ.'y') PRM=.TRUE.
   DO 210 N=1,NE !遍历单元数
   CALL DIV(N,IX,ME) !将所计算单元的节点信息抽调出来到 ME(3)
   CALL AXY(XC,B,C,AE) !将所计算单元的节点坐标信息和计算面积抽调到 B\C\AE
   DO 120 NR=1,3
```

```
      MR=ME(NR)
      DO 130 NS=1,3
      MS=ME(NS)
      CALL WEIGHT(N,PRM,NR,NS,MRS,B,C,E0,T0,XNU,AE,RO) !构造单元子矩阵 MRS
      DO 140 I=1,2
      NR2=2*(MR-1)+I
      DO 140 J=1,2
      NS2=2*(MS-1)+J
      WM(NR2,NS2)=WM(NR2,NS2)+MRS(I,J) !组集总质量矩阵
140   CONTINUE
130   CONTINUE
120   CONTINUE
210   CONTINUE
```

通过降阶法处理质量矩阵、刚度矩阵与荷载数组：

```
DO I=1,NU !开始循环行数列数，降阶处理
K=US(I)
DO L=K-I+1,NJ2-I
DO J=1,NJ2
KK(L,J)=KK(L+1,J)
WM(L,J)=WM(L+1,J)
END DO
END DO
DO M=K-I+1,NJ2-I
DO N=1,NJ2
KK(N,M)=KK(N,M+1)
WM(N,M)=WM(N,M+1)
END DO
END DO
DO L=K-I+1,NJ2-I
UU(L)=UU(L+1)
END DO
END DO
```

将降阶后的质量矩阵、刚度矩阵、荷载数组赋值到新的数组中，方便后续进行计算：

```
DO I=1,NJ22
DO J=1,NJ22
```

```
KKK(I,J)=KK(I,J)
WMM(I,J)=WM(I,J)
CCC(I,J)=AL*WMM(I,J)+BE*KKK(I,J)
END DO
END DO
DO I=1,NJ22
UUU(I)=UU(I)
END DO
```

初始加速的计算：

```
DO I=1,NJ22
UUU0(I)=UUU(I)/WMM(I,I)
END DO
```

积分常数的计算：

```
C0=1/DT/DT
C1=1/2/DT
C2=2*C0
```

计算-Δt 时刻的位移，并将用于存储每个时刻位移的矩阵 AAA 进行初始的赋值：

```
DO I=1,NJ22
UUUFD(I)=UUU0(I)/C2
AAA(I,1)=UUUFD(I)
END DO
```

Q1、Q2 等参数的计算：

```
DO I=1,NJ22
DO J=1,NJ22
WMMY(I,J)=C0*WMM(I,J)+C1*CCC(I,J)
Q1(I,J)=KKK(I,J)-C2*WMM(I,J)
Q2(I,J)=C0*WMM(I,J)-C1*CCC(I,J)
END DO
END DO
```

程序的最重要部分，进行迭代求解每个时刻的位移，我是采用矩阵 AAA 进行存储每个时刻的位移：

```
NNN=FLOOR(TIME/DT) !向下取整数循环次数
DO I=1,NNN
DO J=1,NJ22
DO K=1,NJ22
```

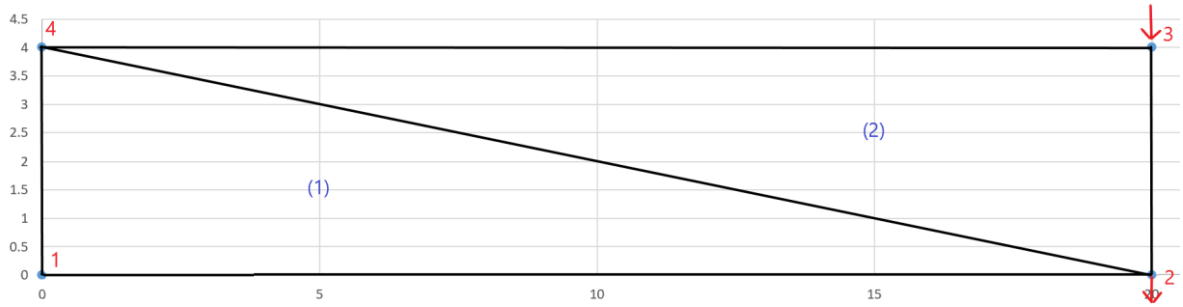
```
FD(J)=UUU(J)-Q1(K,J)*AAA(K,I+1)-Q2(K,J)*AAA(K,I)
END DO
END DO
CALL GS1(NJ22,WMMY,FD)
DO L=1,NJ22
AAA(L,I+2)=FD(L)
END DO
END DO
```

输出结果，需要注意的是要对最后的输出格式进行调整：

```
!输出位移
    WRITE(6,3100)
DO 500 J=1,NNN
    DO 480 I=1,NJ22/2
    WRITE(6,3200) J,I,AAA(2*I-1,J+2),AAA(2*I,J+2)
480     CONTINUE
500 CONTINUE
```

4.3 动力学程序算例分析

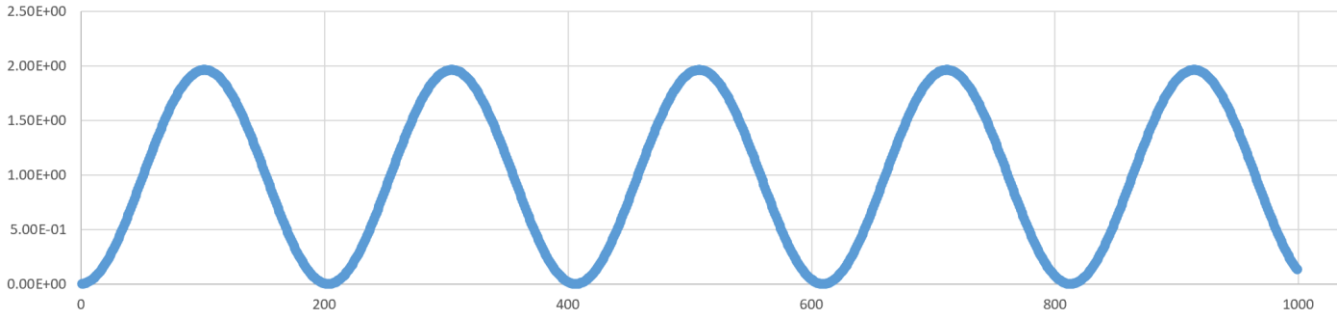
在此，考虑一个左端固定的悬臂梁，梁右端受到竖向载荷时的振动问题，划分单元如图所示。梁的长度为 20，宽和厚度均为 4。



文件的算例为：

```
4 2 4 2 !基本信息
1 2 0 1 4 0.05 0 0 50 !后四个分别为时间间隔 Δt、阻尼矩阵参数 α 与 β 、总时间 t
0,0    20,0    20,4    0,4 !节点坐标
2 4 1 !单元信息
4 2 3
1    2    7    8 !约束信息
4,-20 6,-20 !荷载信息
```

最终得到的 2 节点竖向位移随时间曲线如下：



注意此图中纵坐标为位移的绝对值；横坐标为迭代的次数，1000 对应的地方为 t=50。

而我们通过计算，也可以得到该单元的运动方程：

$$\begin{pmatrix} 106.7 & 0 & 0 & 0 \\ 0 & 106.7 & 0 & 0 \\ 0 & 0 & 53.3 & 0 \\ 0 & 0 & 0 & 53.3 \end{pmatrix}\begin{pmatrix} u_2'' \\ v_2'' \\ u_3'' \\ v_3'' \end{pmatrix}+\begin{pmatrix} 10.8 & 0 & -10 & -2 \\ 0 & 20.4 & 0 & -20 \\ -10 & 0 & 10.8 & 2 \\ -2 & -20 & 2 & 20.4 \end{pmatrix}\begin{pmatrix} u_2 \\ v_2 \\ u_3 \\ v_3 \end{pmatrix}=\begin{pmatrix} 0 \\ -20 \\ 0 \\ -20 \end{pmatrix}$$

最终解的形式也是正弦函数波，这与方程的解形式吻合，是方程的解。

总结与心得

有限元法这门课虽然结课了；但是，对它的学习不会停止。从上学期初入 FORTRAN，到这学期的灵活使用，感觉到了自己很大的进步。

在这里写报告时，我回想起来 2020 年秋季力学工程课上，徐老师给我们讲解了她的研究方向：利用有限元法，在软件里面对复杂的事物进行模拟，把复杂的事物拆分后进行简化，比如在墙上钉钉子的过程中，钉子所受的力、漏斗中物体的流下过程中所受的力、土堆坍塌的过程中各部分所受的力，离散单元法 DEM 的相关介绍等。这也许就是我第一次接触到连续体离散化的思想，当时就给刚进入大学的我不小的震撼，原来问题还能这样解决。

两年多过去了，虽然已经对有限元法有了初步的学习，但未知的还有很多，后续还要继续学习。

最后，感谢徐老师的教导！