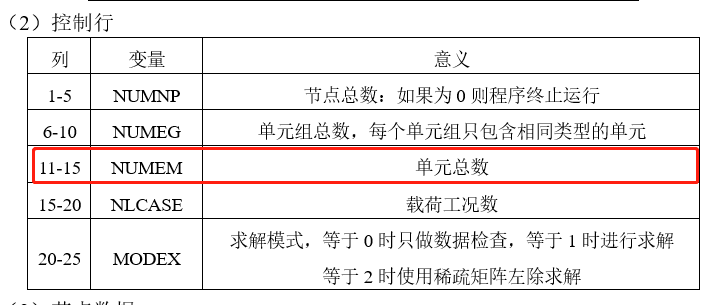
stapmat 改动 输出Tecplot可读格式

修改了输入文件的格式 增加了输入总单元个数

NUMEM



输出变形后的构型

文件名格式“anim\_XX.DAT”

1. ControlData增加变量

1）文件指针IDAT\_ANIM

2）总的单元个数 NUMEM

2. stapmat.m主函数改动：

增加一个函数WriteParasDatANIM()

cdata,IDAT\_ANIM = fopen([‘.\Data\anim\_’,fileID,’.DAT’],’w’);

WriteParasDatANIM();

%函数路径在Initiation文件夹下

function Addpath()中 增加一句

fclose(cdata.IDA\_ANIM);

3. WriteParasDatANIM()

向“anim\_XX.DAT”写入

TITLE = "Example : C3D8\_test1"

VARIABLES= "X" "Y" "Z"

4. GetStiff.m改动：

将清除sdata.X sdata.Y sdata.Z从ReadC3D8() ReadC3D20() ReadPlane() ReadTruss()内移动至Solve() WriteDis()之后

【原因】

在ReadFile()函数中 读入所有节点坐标赋值给sdata.X sdata.Y sdata.Z

在GetStiff()函数中 应该在遍历所有的单元，即遍历所有的单元组之后，再清空节点坐标数据

5. 计算总的单元个数，并检验&写入

GetStiff()改动

遍历单元组循环前，提取文件指针，初始化单元总个数

IDAT\_ANIM = cdata.IDAT\_ANIM;

NUMEM = int64(0);

循环内，读取NPAR()数组之后,累计更新NUMEM

NUMEM = NUMEM + cdata.NPAR(2);

循环外，对比读入cdata.NUMEM

if (cdata.NUMEM ~= NUMEM)

fprintf(IOUT,'\nTotal Number of elements is error!');

return

end

向“anim\_XX.DAT”写入 ZONE F=FEPOINT N = NUMNP E = NUMEM

fprintf(IDAT\_ANIM,'ZONE F=FEPOINT N = %5d E = %5d ET=BRICK C=CYAN\n',cdata.NUMNP,cdata.NUMEM);

6.计算变形后各节点坐标，并写入

在Solve()中的Fuction WriteDis()中改动

IDAT\_ANIM = cdata.IDAT\_ANIM;

X = sdata.X;Y = sdata.Y;Z = sdata.Z;

for II = 1:NUMNP

…

fprintf(IDAT\_ANIM,'%12.4E %11.4E %11.4E\n',X(II)+D(1), Y(II)+D(2), Z(II)+D(3));

…

end

7.写入单元信息 即各单元中节点编号

首先 储存单元节点编号信息

在SolutionData中增加变量

ELEII; %int, ELEII(NUMEM,NNODE), element information

NUMEGEM % int, NUMEGEM(NUMEG), Numid of elements in one element group;

在GetStiff()函数中，遍历各个单元组前，初始化

sdata.NUMEGEM = zeros(cdata.NUMEG,1,'int8');%初始化各单元组中单元个数

sdata.ELEII = zeros(cdata.NUMEM, sdata.NNODE, 'double'); % 单元内各个节点编号信息

单元组循环内赋值

sdata.NUMEGEM(N,1) = NUMEM + cdata.NPAR(2);

在ReadC3D8()的ReadElements()和ReadC3D20()的ReadElements()中更改

将遍历的单元组的编号传递下去

所以将C3D8Stiff(NUMEG\_ID) C3D20Stiff(NUMEG\_ID)

ReadC3D8(NUMEG\_ID) ReadC3D20(NUMEG\_ID)

Read Elements(NUMEG\_ID)

在读入单元组中各单元的节点编号信息后，储存起来

if (NUMEG\_ID ~= 1)

for i3 = 1:sdata.NNODE

sdata.ELEII(sdata.NUMEGEM(NUMEG\_ID-1)+N,i3) = II(i3);

end

else

for i3 = 1:sdata.NNODE

sdata.ELEII(N,i3) = II(i3);

end

end

在向“anim\_XX.DAT”写入变形后的节点坐标信息后，在Finalize()中写入单元编号信息

global sdata;

for N = 1:cdata.NUMEM

fprintf(cdata.IDAT\_ANIM,'\n');

for NN = 1:sdata.NNODE

fprintf(cdata.IDAT\_ANIM,'%7d',sdata.ELEII(cdata.NUMEM,NN));

end

fprintf(cdata.IDAT\_ANIM,'\n');

end

输出曲线结果

文件名格式 “curv\_XX.DAT”

1.ControlData增加变量

1）文件指针 IDAT\_CURV

2.stapmat.m主函数改动：

增加一个函数WriteParasDatCURV()

cdata.IDAT\_CURV = fopen(['.\Data\curv\_',fileID,'.DAT'], 'w');

WriteParasDatCURV();%创建Tecplot可读文件

function Addpath()中 增加一句

fclose(cdata.IDAT\_CURV);

3. WriteParasDatCURV()

向“curv\_XX.DAT”写入

TITLE = "Example : C3D8\_test1"

VARIABLES= "NODE\_ID" "U1" "U2" "U3" "E11" "E22" "E33" "E12" "E13" "E23""S11" "S22" "S33" "S12" "S13" "S23"

4. 单元应力局部磨平

在solutionData中增加变量

NODE\_FLAG; % int, NODE\_FLAG(NUMNP,NLCASE),In all element, the times of the node appears

数组NODE\_FLAG(NUMNP) 节点在各个单元中出现的次数

STRAIN; % double, STRAIN(NUMNP,6, NLCASE), Strain

STRESS; % double, STRESS(NUMNP,6, NLCASE), Stress

GetStress() 在遍历单元组之前，初始化变量

global sdata;

sdata.NODE\_FLAG = zeros(cdata.NUMNP,cdata.NLCASE,'int8');

sdata.STRAIN = zeros(cdata.NUMNP,6,cdata.NLCASE,'double');

sdata.STRESS = zeros(cdata.NUMNP,6,cdata.NLCASE,'double');

在C3D8Stress()

创建数组用来存储当前遍历单元的各个节点的应力应变值

strain\_n = zeros(6,ng^3);

stress\_n = zeros(6,ng^3);

创建磨平矩阵的逆

smooth\_Ni = zeros(ng^3, ng^3,'double');

smooth\_num = int8(0);

for i = 1:ng

for j = 1:ng

for k = 1:ng

smooth\_num = smooth\_num+1;

…

smooth\_Ni(smooth\_num,:) = C3D8Ni(ksi(i),eta(j),zeta(k));

…

end

end

end

使用磨平矩阵计算节点应力应变

strain\_n(1:6,:) = (smooth\_Ni \ strain\_e(1:6,:)')';

stress\_n(1:6,:) = (smooth\_Ni \ stress\_e(1:6,:)')';

找到对应单元编号

if (N ~=1)

element\_id = sdata.NUMEGEM(NG-1)+N;

else

element\_id = N;

end

找到对应节点编号，记录各个节点储存的次数(因为考虑到从各个单元应力磨平之后得到的节点的应力可能不同，所以需要做平均) 并储存应力应变值

此时的应力应变值并不正确，尚未做平均

for i = 1:sdata.NNODE

node\_id = sdata.ELEII(element\_id,i);

sdata.NODE\_FLAG(node\_id,NUM) = sdata.NODE\_FLAG(node\_id,NUM)+1;

sdata.STRAIN(node\_id,:,NUM) = sdata.STRAIN(node\_id,:,NUM)+strain\_n(:,i)';

sdata.STRESS(node\_id,:,NUM) = sdata.STRAIN(node\_id,:,NUM)+stress\_n(:,i)'; end

在C3D20Stress()

创建数组用来存储当前遍历单元的各个节点的应力应变值

strain\_n = zeros(6,ng^3);

stress\_n = zeros(6,ng^3);

创建磨平矩阵的逆

smooth\_Ni = zeros(ng^3, ng^3,'double');

smooth\_num = int8(0);

for i = 1:ng

for j = 1:ng

for k = 1:ng

smooth\_num = smooth\_num+1;

…

smooth\_Ni(smooth\_num,:) = C3D20Ni(ksi(i),eta(j),zeta(k));

…

end

end

end

使用磨平矩阵计算节点应力应变

strain\_n(1:6,:) = (smooth\_Ni \ strain\_e(1:6,:)')';

stress\_n(1:6,:) = (smooth\_Ni \ stress\_e(1:6,:)')';

找到对应单元编号

if (N ~=1)

element\_id = sdata.NUMEGEM(NG-1)+N;

else

element\_id = N;

end

找到对应节点编号，记录各个节点储存的次数(因为考虑到从各个单元应力磨平之后得到的节点的应力可能不同，所以需要做平均) 并储存应力应变值

此时的应力应变值并不正确，尚未做平均

for i = 1:sdata.NNODE

node\_id = sdata.ELEII(element\_id,i);

sdata.NODE\_FLAG(node\_id,NUM) = sdata.NODE\_FLAG(node\_id,NUM)+1;

sdata.STRAIN(node\_id,:,NUM) = sdata.STRAIN(node\_id,:,NUM)+strain\_n(:,i)';

sdata.STRESS(node\_id,:,NUM) = sdata.STRAIN(node\_id,:,NUM)+stress\_n(:,i)'; end

在GetStress中做平均

%应力磨平

for j = 1:cdata.NLCASE

for i = 1:cdata.NUMNP

if sdata.NODE\_FLAG(i) ~= 1

sdata.STRAIN(i,[1 2 3 4 5 6],j) = sdata.STRAIN(i,[1 2 3 4 5 6],j)./NODE\_FLAG(i);

sdata.STRESS(i,[1 2 3 4 5 6],j) = sdata.STRESS(i,[1 2 3 4 5 6],j)./NODE\_FLAG(i);

end

fprintf(IDAT\_CURV,'%15d%',i);

for ii = 1:3

fprintf(IDAT\_CURV,'%15.6E%',sdata.DISP(i,ii,j));

end

for ii = 1:6

fprintf(IDAT\_CURV,'%15.6E%',sdata.STRAIN(i,ii,j));

end

for ii = 1:6

fprintf(IDAT\_CURV,'%15.6E%',sdata.STRESS(i,ii,j));

end

fprintf(IDAT\_CURV,'\n');

end

fprintf(IDAT\_CURV,'\n');

end

1.节点顺序对不上，检查一下，一定要debug

