**新单元添加流程**

在计算程序中，添加新单元时，需要修改以下三个项目。

**1、intelcal64主计算项目**

**2、outputintel结果输出项目**

**3、pathresult项目**

目录

[intelcal64主计算项目 4](#_Toc99920817)

[一、 普通单元 4](#_Toc99920818)

[1. 确定新单元的名称和编号 4](#_Toc99920819)

[2. 建立新单元相关的f90文件 4](#_Toc99920820)

[3. 建立新单元模块 4](#_Toc99920821)

[4. 单元刚度矩阵的生成 5](#_Toc99920822)

[5. 单元节点应力的生成 5](#_Toc99920823)

[6. 应力分片结果输出 6](#_Toc99920824)

[二、 应力杂交单元 7](#_Toc99920825)

[1. 确定新单元的名称和编号 7](#_Toc99920826)

[2. 建立新单元相关的f90文件 7](#_Toc99920827)

[3. 建立新单元模块 7](#_Toc99920828)

[4. 单元内部点信息 7](#_Toc99920829)

[5. 处理单元内部点信息 8](#_Toc99920830)

[6. 应力函数项数和P矩阵 9](#_Toc99920831)

[7. 单元刚度矩阵的生成 13](#_Toc99920832)

[8. 单元内部节点位移的生成 13](#_Toc99920833)

[9. 单元应力参数beta的生成 13](#_Toc99920834)

[10. 单元节点应力的生成 14](#_Toc99920835)

[11. 应力分片结果输出 14](#_Toc99920836)

[outputintel结果输出项目 15](#_Toc99920837)

[一、 单元通用变量 15](#_Toc99920838)

[1. 变量输出 15](#_Toc99920839)

[2. 变量读入 15](#_Toc99920840)

[二、 单元特殊变量 15](#_Toc99920841)

[1. 单元特殊变量的输出 15](#_Toc99920842)

[2. 特殊单元变量的读入 16](#_Toc99920843)

[pathresult项目 17](#_Toc99920844)

[一、 计算穿过单元各部分路径 17](#_Toc99920845)

[二、 修改在单元中某条路径上点的应力的计算 17](#_Toc99920846)

# intelcal64主计算项目

intelcal64项目为主要计算模拟程序，添加新单元时，需要修改以下部分。

## 普通单元

对于普通单元需要修改以下六个部分。

### 确定新单元的名称和编号

预添加的新单元名称为：**newElement**，

假设单元编号为：**90001**

需要在elementLib\tools\module.f90文件中的module elementType中添加：

integer(4),parameter::Type\_newElement=90001

**注意：**

**input文件中，单元类型要跟这个编号一致**

**临时生成的单元编号统一为：90000+n**

### 建立新单元相关的f90文件

在文件夹\elementLib\2D\或\elementLib\3D\中，建立新单元的文件**newElement.f90**，后续新单元相关的模块和子程序都添加到该文件中。

当单元子程序太多时，可以分到多个文件中，例如Voronoi单元、两相材料单元。

### 建立新单元模块

在文件**newElement.f90**中，添加模块module：

module ELEMENT\_newElement

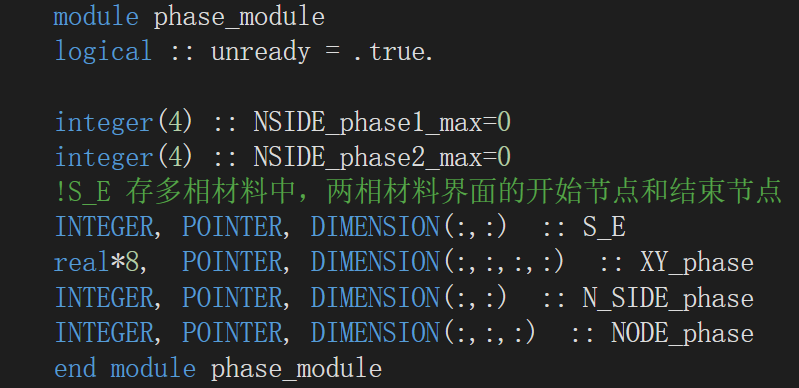
（这里添加新单元所需的相关参数和数组）

real\*8::newData(6,20)

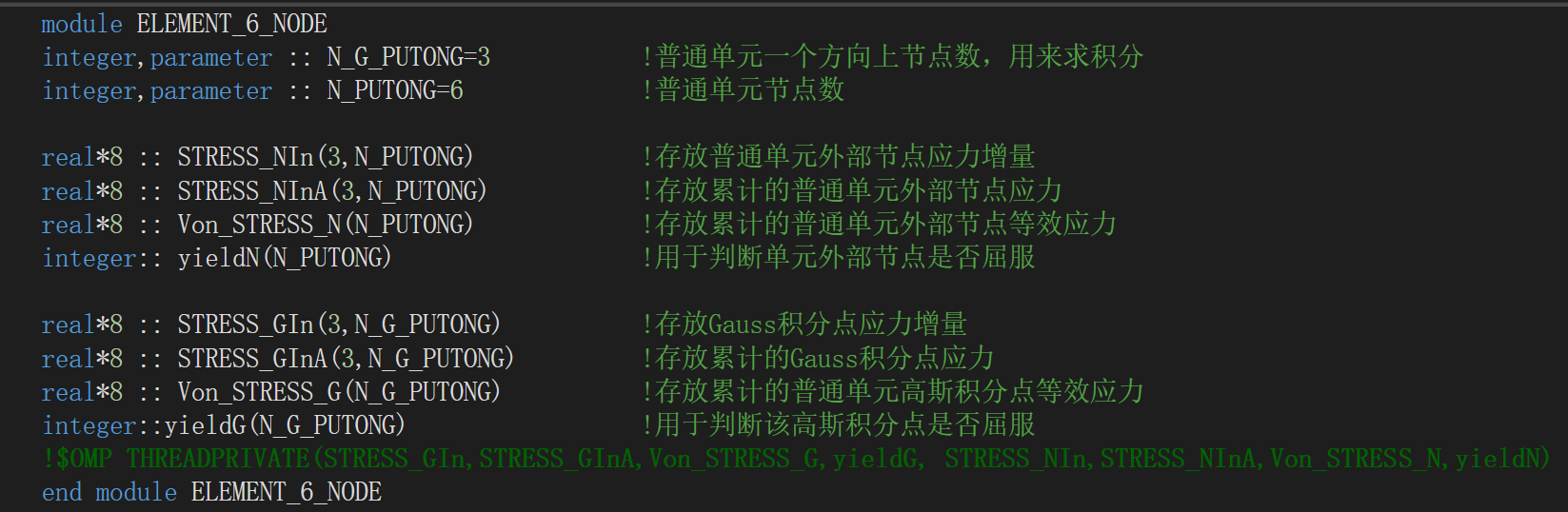
!$OMPTHREADPRIVATE( newData )

Endmodule ELEMENT\_newElement

例如：两相材料单元，模块在文件elementLib\2D\element\_2phase\Phase\_matrix.f90中。



例如：6节点普通三角形单元，模块在文件\elementLib\2D\Element\_6node.f90中。



### 单元刚度矩阵的生成

在文件newElement.f90中，添加生成单刚的子程序：

SUBROUTINE GET\_STIFF\_newElement(NUM\_ELEMENT1,……)

在文件element\_interface.f90的子程序SUBROUTINE INTERFACE\_STIFFMATRIX中添加生成单刚的子程序GET\_STIFF\_newElement的调用，形式如下：

SELECTCASE(N\_TYPE(NUM\_ELEMENT1))

case(Type\_newElement)

Call GET\_STIFF\_newElement(NUM\_ELEMENT1,……)

ENDSELECT

### 单元节点应力的生成

在文件newElement.f90中，添加计算单元节点应力的子程序：

SUBROUTINE GET\_STRESS\_newElement(NUM\_ELEMENT1,……)

在文件element\_interface.f90的子程序SUBROUTINE INTERFACE\_NODE\_STRESS中添加计算单元节点应力子程序的调用，形式如下：

ELECTCASE(N\_TYPE(NUM\_ELEMENT1))

case(Type\_newElement)

Call GET\_NODE\_STRESSnewElementNUM\_ELEMENT1,……)!（计算单元节点应力）

ENDSELECT

### 应力分片结果输出

对于需要特殊分片的单元，需要修改单元内遍历各个分片的程序。

在文件newElement.f90中，添加应力分片结果输出的子程序：

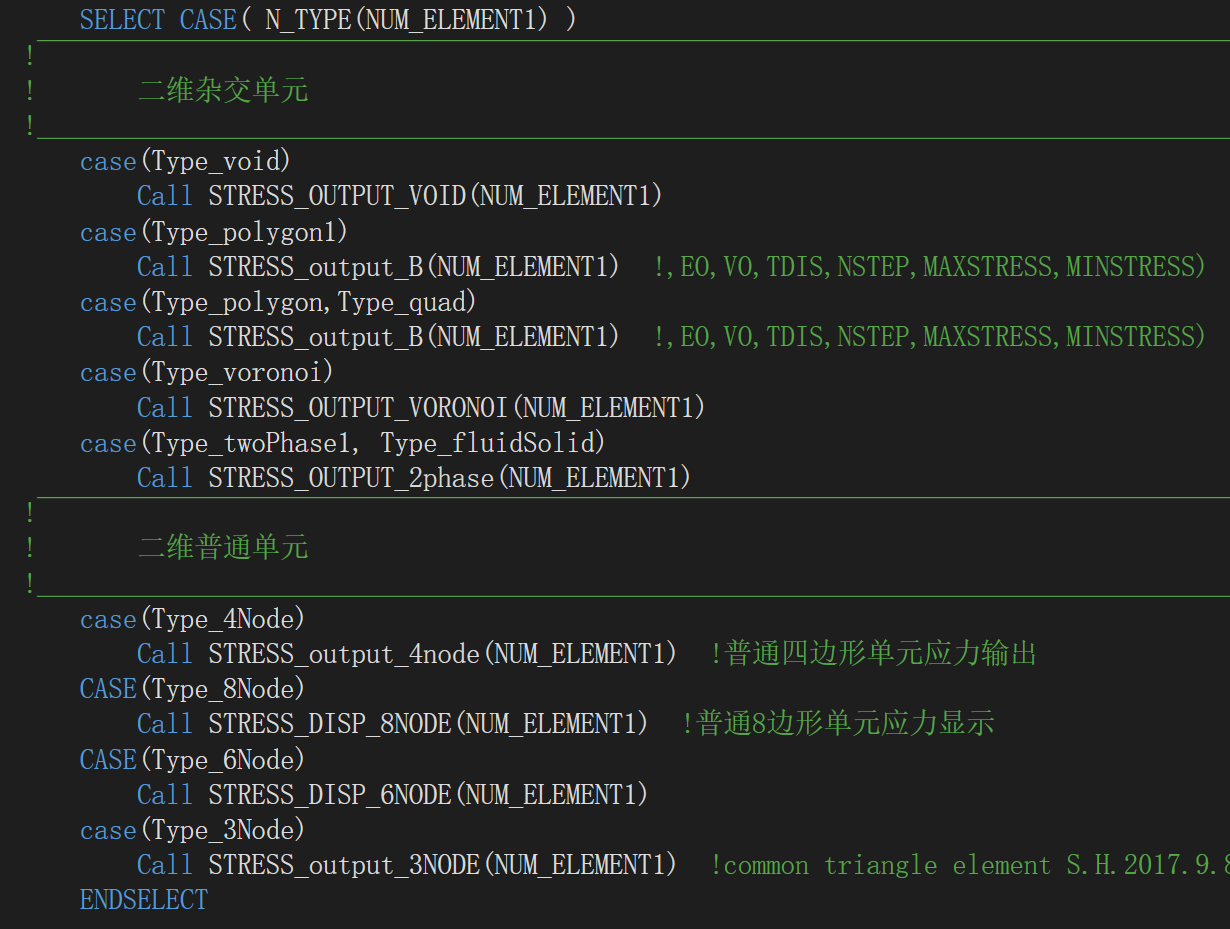
SUBROUTINE STRESS\_OUTPUT\_newElement(NUM\_ELEMENT1)

二维问题，在文件element\_interface.f90的SUBROUTINE INTERFACE\_OUTPUT\_STRESS()子程序中添加输出子程序STRESS\_OUTPUT\_newElement的调用，形式如下：

case(Type\_newElement)

Call STRESS\_OUTPUT\_newElement (NUM\_ELEMENT1)

例如：



三维问题在文件element\_interface.f90的SUBROUTINE INTERFACE\_OUTPUT\_STRESS3D子程序中添加输出子程序的调用，形式如下：

case(Type\_newElement)

Call draw\_ newElement (NUM\_ELEMENT1)

例如：



## 应力杂交单元

### 确定新单元的名称和编号

### 建立新单元相关的f90文件

### 建立新单元模块

### 单元内部点信息

对于有特殊内部点的单元，需要单独读取这些点的信息。如两相材料单元。

#### 修改input文件

单元特殊内部点的信息放在输入模型文件中的INCLUSIONS字段，每个单元根据读夹杂信息时读入的第二位参数kind\_internal\_node，决定后边的读取方式和内容。详细格式参考：[input1.4.md](https://github.com/guoronline/GPU-server/blob/main/FEM/input1.4.md)

例如：

##### 界面相信息

针对多相材料应力杂交元，输入多相材料的界面相信息时，信息类型为：7

单元编号, 7, 界面数

下面根据相数循环输入界面信息：

1行：开始节点、结束节点、内部节点数；  
2行：输内部点的坐标；  
3行：输入夹杂的中心和长短轴和角度。

例如，单元有2个相界面，就需要输入：

5, 7, 2  
123, 122, 4  
1.50479, 5.76863, 1.53433, 5.52429, 1.65512, 5.30985, 1.84878, 5.15797,   
2.13375, 5.72071, 0.630783, 0.630783, 175.643  
126, 128, 2  
1.50479, 5.76863, 1.53433, 5.52429,   
2.13375, 5.72071, 0.630783, 0.630783, 175.643

##### 多面体信息

输入三维单元的任意多面体信息时，信息类型为：**301**

单元号，301，总面数

接着指定每个面的局部连接信息。

序号，这个面上的点数，局部编号下连接信息（方向指向外）

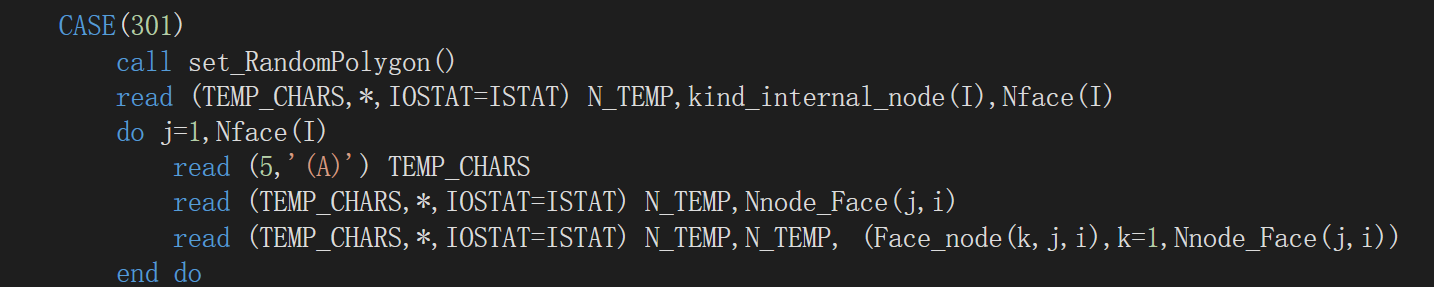
#### 修改input.f90的子程序read\_internal\_node

在文件input.f90的子程序SUBROUTINE read\_internal\_node()中添加读取特殊内部点的信息读写程序。

CASE(自己给定的内部点信息类型编号)

读取和储存特殊内部点的信息

例如：



### 处理单元内部点信息

在文件newElement.f90中，添加处理内部点的子程序：

SUBROUTINE Element\_data\_newElement(NUM\_E,XY)

在文件element\_interface.f90中，添加子程序Element\_data\_newElement的调用，形式如下：

SELECTCASE(N\_TYPE(NUM\_E))

case(Type\_newElement)

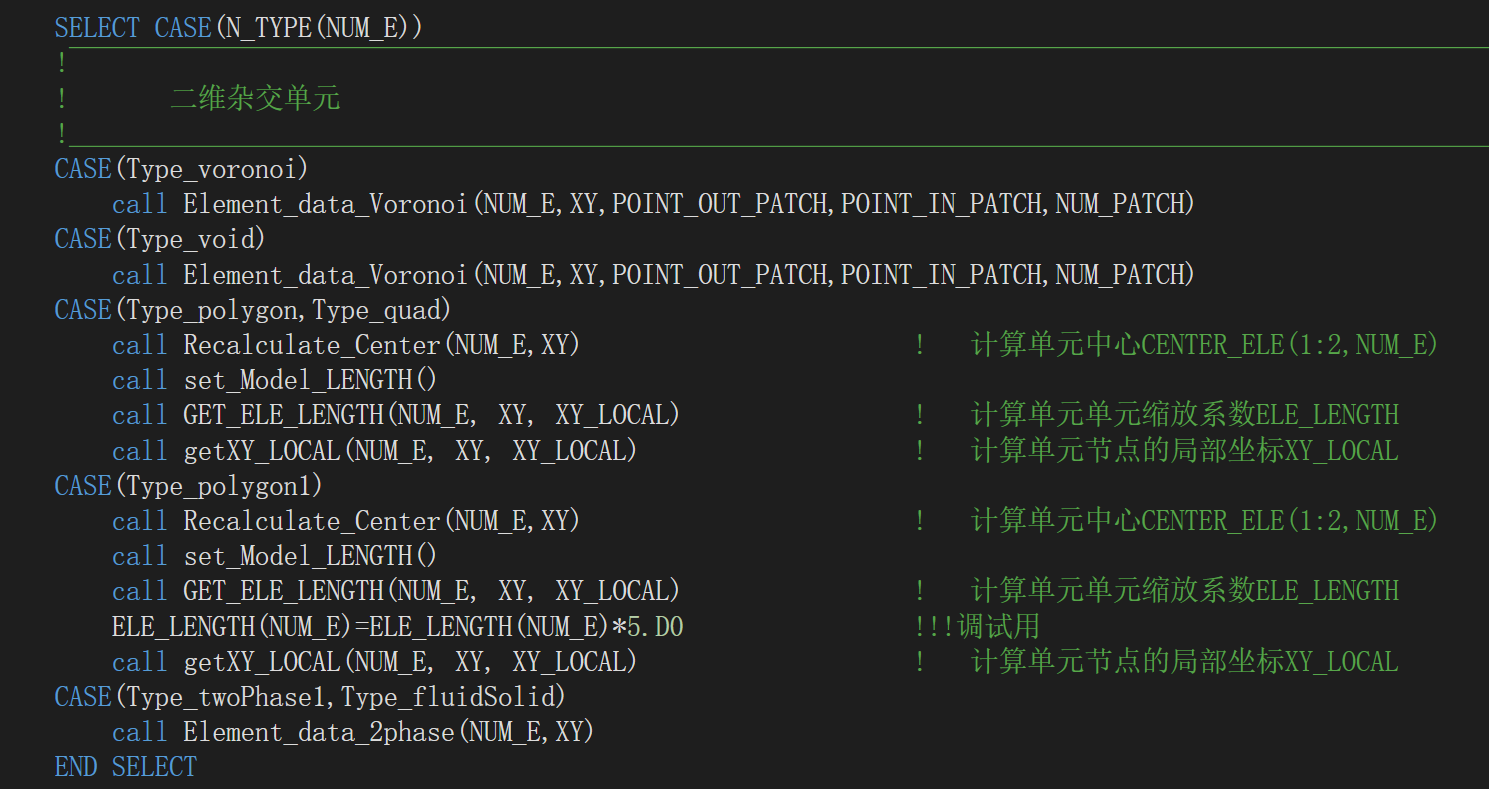
call Element\_data\_newElement(NUM\_E,XY)

ENDSELECT

例如：

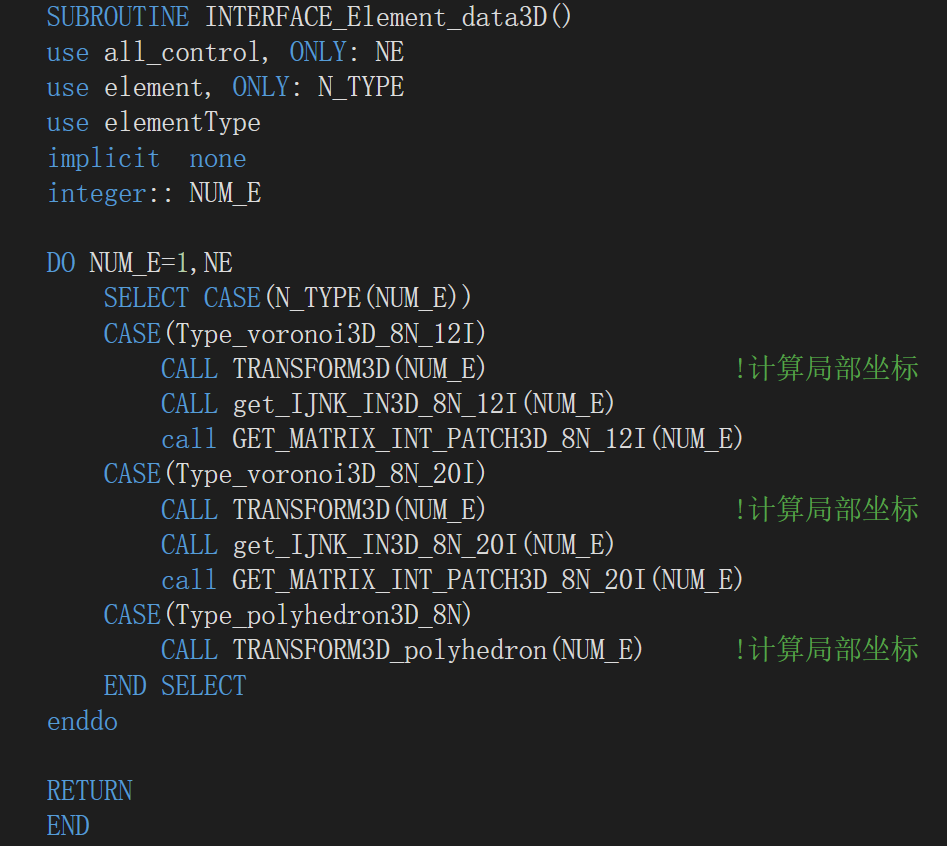
#### 二维问题

在子程序SUBROUTINE INTERFACE\_Element\_data中添加处理特殊内部点的子程序。



#### 三维问题

在子程序SUBROUTINE INTERFACE\_Element\_data3D中添加处理特殊内部点的子程序。



### 应力函数项数和P矩阵

#### 应力函数的定义：

形状相关函数：



由于在界面上，，同时，函数具有当时，的特性，它可以作为一种映射函数，被用于构造基于形状的互作用应力函数（shaped-based reciprocal stress function）。

当时, ，会导致互作用项太大；当时，会导致的高阶项（）数值太大，同时应力函数的高阶项函数进一步修改为：



其中，在0到1之间，默认。

多项式应力函数表示为

 其中

对应每一项多项式应力函数，有多个互作用应力函数项。

在夹杂外侧的部分，互作用应力函数项表示为：



当时, ；一般情况下，在单元内最边缘的点大概是椭圆半轴的5-10倍，当时, ，这样在夹杂内，当积分点远离界面时，应力函数会减小，但又不会太小。

在夹杂内部的部分，互作用应力函数项表示为：



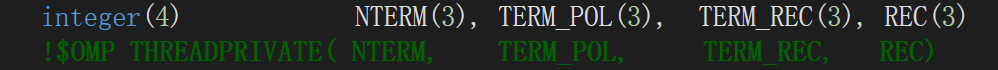
当时, ，这样在夹杂内，当积分点远离界面时，应力函数会减小，但又不会太小。

对每一个考虑互作用项的应力杂交单元，基体的应力函数为



多项式应力函数项数，互作用应力函数中多项式的项数，组数，互作用应力函数总项数为，应力函数总项数为或。

在\elementLib\tools\module.f90文件中，模块module ALL\_CONTROL 中定义了控制应力参数数量相关的四个数组。



**注：**

**1、2、3分别表示夹杂、基体和第三相的项数。**

**!$OMP：表示omp并行语句**

**THREADPRIVATE表示在omp并行计算中，括号里的变量是线程私有的。**

#### 应力函数项数的确定：

对于应力杂交元，程序在循环计算：单刚、应力参数、节点应力、输出分片应力结果时，都需要确定P、H、G矩阵的大小，所以在所有单元的循环中，对于每一个单元都需要调用子程序GET\_TERM，确定该单元的应力参数。因此，每类应力杂交元都要在其中定义其应力参数数量。

子程序GET\_TERM在文件夹\elementLib\tools\中的term.f90文件中，在子程序

SUBROUTINE GET\_TERM(NUM\_ELEMENT)

中，添加新单元的CASE：

CASE(Type\_newElement)

FRec1min = 0.5D0

FRec2min = 0.5D0

IF(IFLAG\_REC==0) THEN !夹杂和基体都不考虑互作用项

TERM\_REC(2) = 0

REC(2) = 0

TERM\_REC(1) = 0

REC(1) = 0

NTERM(2) = TERM\_POL(2)

NTERM(1) = TERM\_POL(1)

ELSE IF(IFLAG\_REC==1) THEN !夹杂不考虑互作用项,基体考虑互作用项。

NTERM(2) = TERM\_POL(2)+TERM\_REC(2)\*REC(2)

TERM\_REC(1) = 0

REC(1) = 0

NTERM(1) = TERM\_POL(1)

ELSE IF(IFLAG\_REC==3) THEN !夹杂和基体都考虑互作用项。

NTERM(2) = TERM\_POL(2)+TERM\_REC(2)\*REC(2)

NTERM(1) = TERM\_POL(1)+TERM\_REC(1)\*REC(1)

ELSE IF(IFLAG\_REC==2) THEN !夹杂考虑互作用项,基体不考虑互作用项。

TERM\_REC(2) = 0

REC(2) = 0

NTERM(2) = TERM\_POL(2)

NTERM(1) = TERM\_POL(1)+TERM\_REC(1)\*REC(1)

ELSE

NTERM(2) = TERM\_POL(2)+TERM\_REC(2)\*REC(2)

NTERM(1) = TERM\_POL(1)+TERM\_REC(1)\*REC(1)

ENDIF

其中：FLAG\_REC为互作用应力选项

FLAG\_REC==0 夹杂和基体都不考虑互作用项

IFLAG\_REC==1 夹杂不考虑互作用项,基体考虑互作用项。 11标识考虑相邻单元的夹杂

IFLAG\_REC==2 夹杂和基体都考虑互作用项。 12标识考虑相邻单元的夹杂

IFLAG\_REC==3 夹杂考虑互作用项,基体不考虑互作用项。 13标识考虑相邻单元的夹杂

FLAG\_REC的值，从input文件中，OPTION选项后的第一个数读入。

OPTION

1.互作用应力选项, 2.带宽优化, 3.宏观模量, 4.TERM参数, 5.G矩阵积分模式

#### P矩阵的确定：

在文件newElement.f90中，添加生成计算P矩阵的子程序：

SUBROUTINE GET\_P\_newElement(NUM\_ELEMENT1,X,Y,TYPE\_PHASE,TYPE\_REC,P)

二维问题，在文件夹\elementLib\2D\elementTools\中的get\_p.f90文件的子程序

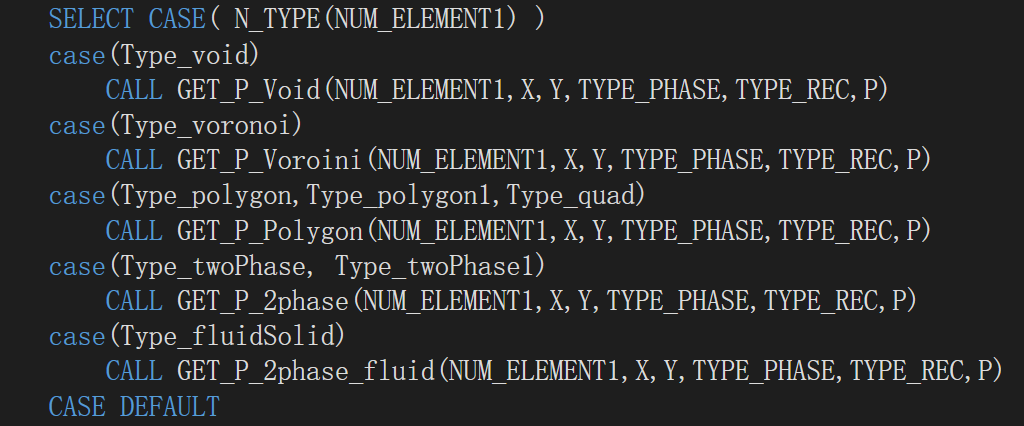
SUBROUTINE GET\_P\_ALL(NUM\_ELEMENT1,X,Y,TYPE\_PHASE,TYPE\_REC,P)

中，添加计算P矩阵的子程序GET\_P\_newElement的调用，形式如下：

case(Type\_void)

CALL GET\_P\_newElement (NUM\_ELEMENT1,X,Y,TYPE\_PHASE,TYPE\_REC,P)

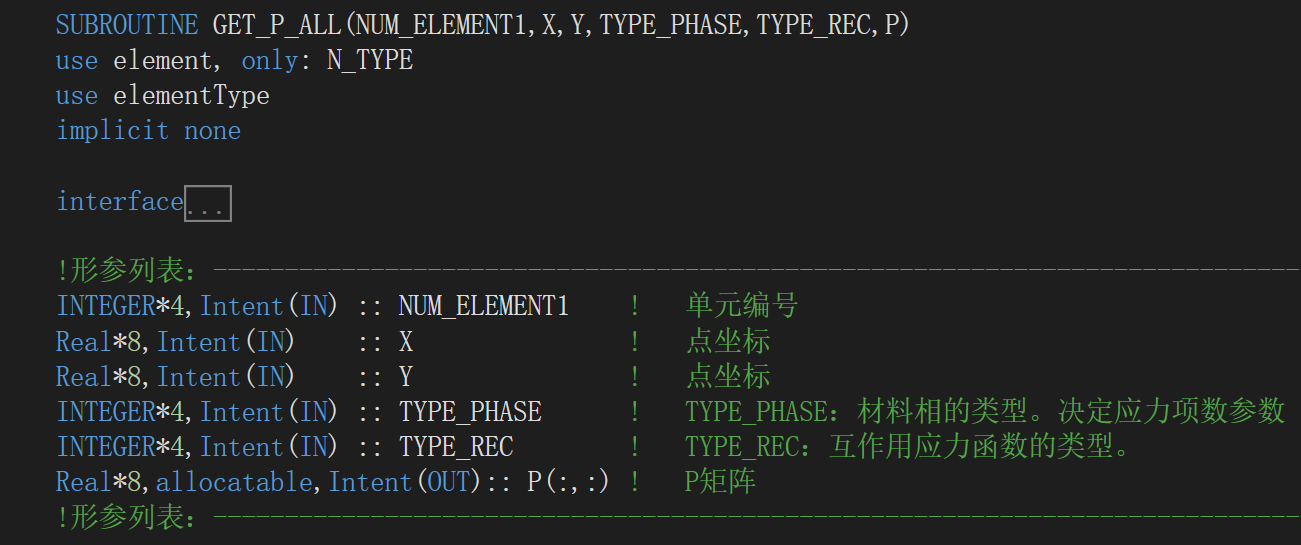
例如：



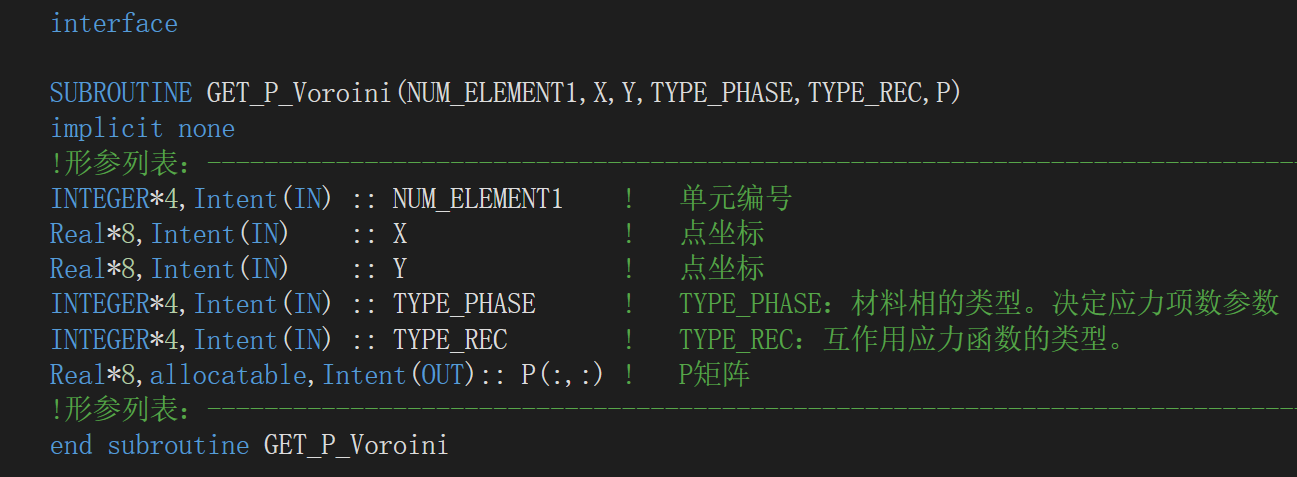
三维问题，添加到文件夹\elementLib\3D\elementTools\中的GET\_P\_3D.f90文件的子程序：

SUBROUTINE GET\_P3D\_ALL(NUM\_ELEMENT1,X,Y,TYPE\_PHASE,TYPE\_REC,P)

在SUBROUTINE GET\_P\_ALL子程序中，由于P矩阵是个可变数组：



必须要将子程序GET\_P\_newElement的定义添加到interface接口中，如下：



### 单元刚度矩阵的生成

在文件newElement.f90中，添加生成单刚的子程序：

SUBROUTINE GET\_STIFF\_newElement(NUM\_ELEMENT1,……)

在文件element\_interface.f90的子程序SUBROUTINE INTERFACE\_STIFFMATRIX中添加生成单刚的子程序GET\_STIFF\_newElement的调用，形式如下：

SELECTCASE(N\_TYPE(NUM\_ELEMENT1))

case(Type\_newElement)

Call GET\_STIFF\_newElement(NUM\_ELEMENT1,……)

ENDSELECT

### 单元内部节点位移的生成

对于含内部点的应力杂交单元，在文件newElement.f90中，添加计算内部节点位移的子程序.

总量计算时，定义内部节点位移总量的子程序：

SUBROUTINE GET\_TDIS\_IN\_newElement(NUM\_ELEMENT1,……)

增量计算时，定义内部节点位移增量的子程序：

SUBROUTINE GET\_DDIS\_IN\_newElement(NUM\_ELEMENT1,……)

在文件element\_interface.f90的子程序SUBROUTINE INTERFACE\_DIS\_IN中添加计算内部节点位移的子程序的调用，形式如下：

Call GET\_TDIS\_INnewElementNUM\_ELEMENT1,……)

Call GET\_DDIS\_INnewElementNUM\_ELEMENT1,……)

如果内部节点位移计算方式不变，也可以不修改。

### 单元应力参数beta的生成

对于应力杂交单元，在文件newElement.f90中，添加计算单元应力参数的子程序。

总量计算时，定义计算单元应力参数总量的子程序：

SUBROUTINE GET\_TSTRESS\_newElement(NUM\_ELEMENT1,……)

增量计算时，定义计算单元应力参数增量的子程序：

SUBROUTINE GET\_DSTRESS\_newElement(NUM\_ELEMENT1,……)

在文件element\_interface.f90的子程序SUBROUTINE INTERFACE\_\_STRESS中添加计算单元应力参数的子程序的调用，形式如下：

SELECTCASE(N\_TYPE(NUM\_ELEMENT1))

case(Type\_newElement)

Call GET\_TSTRESSnewElementNUM\_ELEMENT1,……)!（计算单元应力参数总量）

或Call GET\_DSTRESSnewElementNUM\_ELEMENT1,……)!（计算单元应力参数增量）

ENDSELECT

### 单元节点应力的生成

### 应力分片结果输出

# outputintel结果输出项目

项目主程序位于\for\_drawing\output\_main.f90。该项目根据主计算项目输出的位移和应力参数beta的结果，逐个单元、逐个分片的计算输出每个分片内的应力结果。每个分片中，计算MxN个点的应力值，M和N越大，输出的点数越多，图形连续性越好。M和N在计算APP中或input文件中给定。

## 单元通用变量

### 变量输出

可以在某个特殊变量的IO文件中，添加变量的输出子程序，对于新单元的特殊变量，也可以在文件newElement.f90中，添加变量输出子程序：

subroutine output\_BATE\_IN ()

在文件\for\_cal\output\_results\_for\_intelF.f90的子程序SUBROUTINE OUTPUT\_RESULTS()中添加输出每个增量步计算结果，以便于后处理绘图或输出给tecplot显示。调用形式如下：

call OUTPUT\_RESULTS\_normal()

在子程序SUBROUTINE OUTPUT\_RESULTS\_normal中进一步调用

call output\_BATE\_IN()

### 变量读入

可以在某个特殊变量的IO文件中，添加变量的输出和读入子程序，对于新单元的特殊变量，也可以在文件newElement.f90中，添加变量输出和读入的子程序：

subroutine reopen\_DIS()

subroutine reopen\_BATE\_IN(isTotal)

在文件\elementLib\tools\reopenmodel.f90的子程序SUBROUTINE INPUT\_MODOEL()中：

call reopen\_DIS()

或在子程序SUBROUTINE INPUT\_RESULT()中：

call reopen\_BATE\_IN(1)

## 单元特殊变量

### 单元特殊变量的输出

对于一些特殊单元，有一些特殊的结果变量，在结果输出项目中，计算点的应力时会用到，如裂纹尖端的k因子。这时需要额外输出。

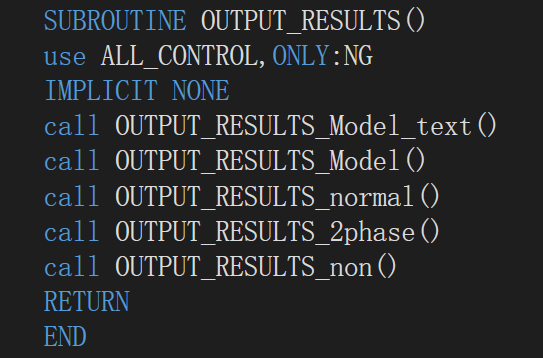
在文件newElement.f90中，添加输出特殊单元变量的子程序：

SUBROUTINE OUTPUT\_RESULTS\_newElement()

在文件output\_results\_for\_intelF.f90的子程序SUBROUTINE OUTPUT\_RESULTS()中添加输出每个增量步计算结果，以便于后处理：绘图或输出给tecplot显示。调用形式如下：

Call OUTPUT\_RESULTS\_ newElement ()

例如：对于两相材料单元，需添加特殊的结果输出子程序：



### 特殊单元变量的读入

对于一些特殊单元，有一些特殊的结果变量，在应力计算中会用到，如裂纹尖端的k因子。这时需要额外读入。

在文件newElement.f90中，添加输出特殊单元变量的子程序：

SUBROUTINE INPUT\_RESULTS\_newElement()

新单元结果输入子程序调用，可以在项目主程序位于\for\_drawing\output\_main.f90中：

CALL INPUT\_MODOEL\_text()

CALL INPUT\_MODOEL()

CALL INPUT\_MODOEL\_2phase()

CALL INPUT\_RESULT()

也可以添加到\elementLib\tools\reopenmodel.f90的SUBROUTINE INPUT\_RESULT()中：

call INPUT\_RESULTS\_newElement ()

# pathresult项目

该项目根据主计算项目输出的位移和应力参数beta的结果，计算路径穿过的单元，然后逐个计算每个穿过单元内、路径上点的应力。

## 计算穿过单元各部分路径

对于内部微结构比较复杂的单元，需要重新计算路径穿过单元各部分（基体、夹杂、空洞等）的进入点和穿出点的信息。修改位置：

1、在文件intersection\_of\_path.f90中的SUBROUTINE GET\_STRESS\_along\_path子程序中添加新的分支。

elseif(N\_TYPE(NUM\_ELEMENT\_CUR) == Type\_newElement) then

2、对于有内部微结构的单元，需要计算路径穿过单元时是否穿过微结构，需要添加路径和微结构位置关系的程序：例如：

Call intersection\_of\_path\_inclusion(…)

3、如果路径穿过微结构，则需要分段输出点应力。

## 修改在单元中某条路径上点的应力的计算

在某个单元内计算某线段（两个点之间）上的n个点的应力，调用子程序：

Call GET\_ANY\_STRESS(…)

在文件intersection\_of\_path.f90的子程序SUBROUTINE GET\_ANY\_STRESS(…)中，同样是调用：

Call GET\_POINT\_STRESS\_4out(…)