SpaceGroupIrep程序包使用说明

SpaceGroupIrep是一款用Mathematica语言开发的用于处理空间群及其不可约表示的开源程序包，它既是一个包含空间群及其不可约表示数据的数据库，又是一个可以操作这些数据的工具集。该程序包中的空间群及不可约表示的数据采用BC约定，即C. J. Bradley 和A. P. Cracknell所著《The Mathematical Theory of Symmetry in Solids》一书（之后该书简称“BC书”）中的约定。历史上，空间群及其不可约表示的数据在多本书中独立给出，但它们都使用着不同的约定，比如Kovalev约定、Zak-Casher-Gluck-Gur（ZCGG）约定，Cracknell-Davies-Miller-Love（CDML）约定，以及Bradley-Cracknell（BC）约定等。不同的约定下原胞的定义、k点（即波矢）的选择以及小群（即波矢群）不可约表示（即小表示）的命名一般是不同的。我们选择BC约定的原因在于BC书是有关空间群不可约表示最流行的一本书，它既包含系统的空间群表示理论还包含完整的不可约表示数据；还有一个很重要的原因是其他几本书都已绝版，很难找到，而BC书还在出版，很容易买到。

SpaceGroupIrep程序包的主要功能包括：获取空间群、小群、Herring小群、小陪群的中心扩展中的元素以及计算元素间的群乘；识别k点；计算任意k点的小群不可约表示（LGIR）和任意波矢星的空间群不可约表示（SGIR）并以直观的表格方式显示结果，单双值表示都支持；计算SGIR的直积分解；判定能带的LGIR；给出k点及LGIR在BCS（Bilbao Crystallographic Server）约定（BCS用的是CMDL约定，因找不到CDML书，只能间接的通过BCS了解CDML约定，于是干脆称之为BCS约定）及BC约定之间的对应。

#### 一、获取和安装

SpaceGroupIrep是一个开源程序包，源码发布在如下网址：

<https://github.com/goodluck1982/SpaceGroupIrep>

程序包的核心文件有以下四个：

* SpaceGroupIrep.wl 主文件，包含大部分函数和数据
* AbstractGroupData.wl 包含BC表5.1中的抽象群数据
* LittleGroupIrepData.wl包含BC表5.7、5.11、6.13和6.15中的不可约表示数据
* allBCSkLGdat.mx 包含BCS的不可约表示数据（从[irvsp](https://github.com/zjwang11/irvsp)的输出结果收集而得）

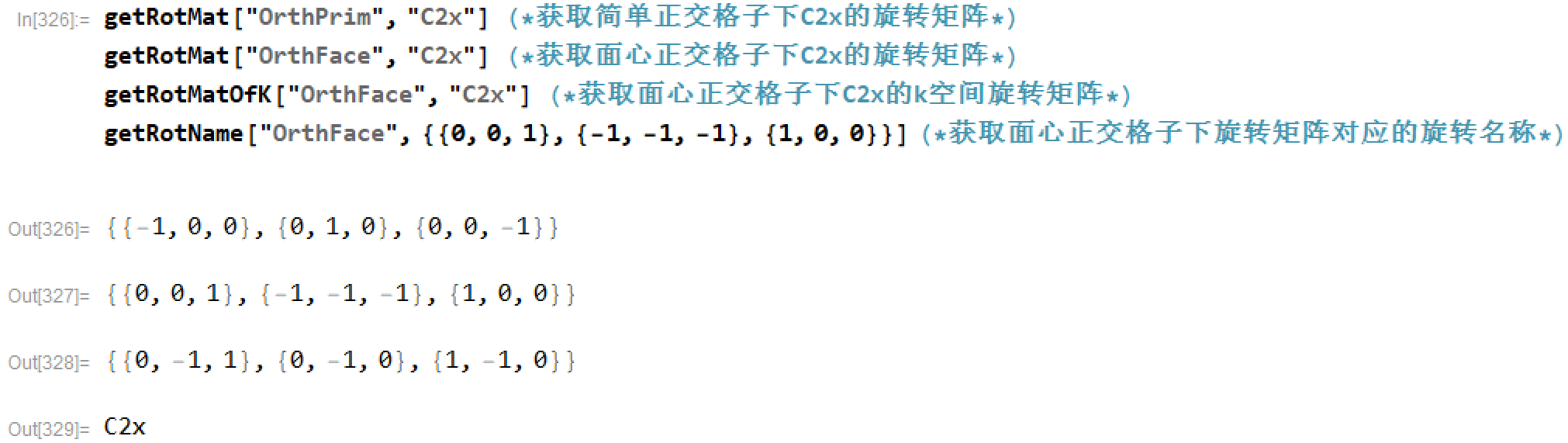
上面所说BC表5.1指的是BC书中的表5.1，其他类似。将上述四个文件放置于名为SpaceGroupIrep的文件夹下并将该文件夹放在以下任一路径下即可：

* $InstallationDirectory/AddOns/Packages/
* $InstallationDirectory/AddOns/Applications/
* $BaseDirectory/Applications/
* $UserBaseDirectory/Applications/

$InstallationDirectory是Mathematica内置变量，在Mathematica中运行即可得到其具体值，比如在我的电脑上它是C:\Program Files\Wolfram Research\Mathematica\11.2。安装完成后，只需运行命令 就可以载入该程序包，然后就可以调用程序包里的功能了。

#### 二、元素及乘法

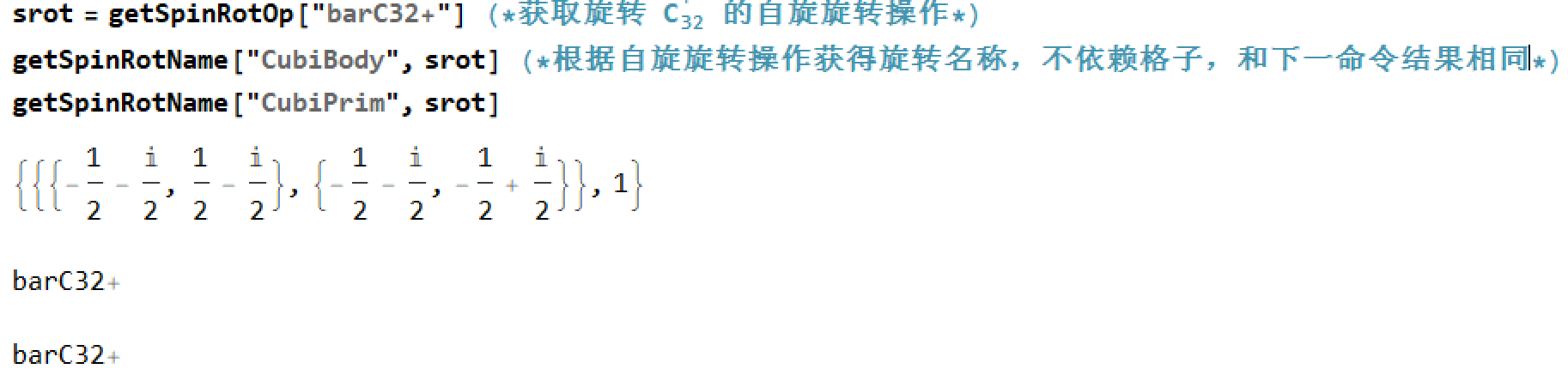
我们用Seitz符号来表示空间群元素，意为先进行旋转再进行矢量的平移。需注意实空间的旋转矩阵和矢量坐标都是以原胞基矢量为基础的数据（并非笛卡尔坐标系下的数据），而原胞基矢必须按照BC表3.1中的定义，相应倒格基矢则按BC表3.3中的定义。BC表3.1中给出了14种布拉维格子（简称格子）每一种的原胞基矢定义，既然旋转矩阵依赖于原胞基矢而原胞基矢又依赖于格子，所以旋转矩阵依赖于格子，同样的旋转在不同格子下的矩阵可能不同。我们可以用如下函数根据旋转名称（参考BC图1.2、1.3及BC表1.4）获取旋转矩阵，也可根据旋转矩阵获取旋转名称



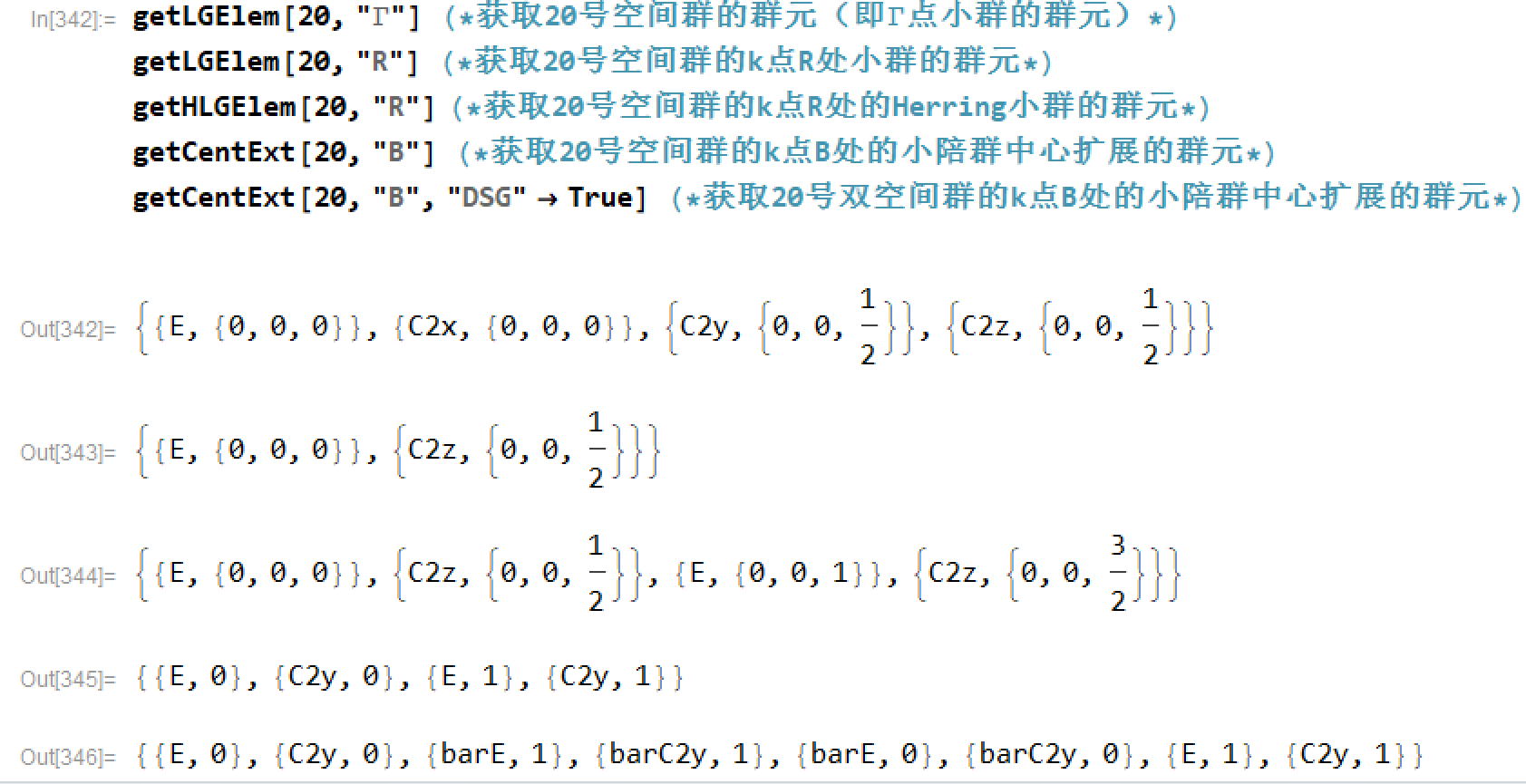
从上面例子中可以看出，我们用一个8字节的字符串来描述一种格子，所有格子及相应字符串见下表

|  |  |  |
| --- | --- | --- |
| 格子 | 字符串 | 布里渊区类型 |
| 简单三斜 | "TricPrim" |  |
| 简单单斜 | "MonoPrim" |  |
| 底心单斜 | "MonoBase" |  |
| 简单正交 | "OrthPrim" |  |
| 底心正交 | "OrthBase" | "a"、"b" |
| 体心正交 | "OrthBody" | "a"、"b"、"c" |
| 面心正交 | "OrthFace" | "a"、"b"、"c"、"d" |
| 简单四方 | "TetrPrim" |  |
| 体心四方 | "TetrBody" | "a"、"b" |
| 简单三角 | "TrigPrim" | "a"、"b" |
| 简单六角 | "HexaPrim" |  |
| 简单立方 | "CubiPrim" |  |
| 面心立方 | "CubiFace" |  |
| 体心立方 | "CubiBody" |  |

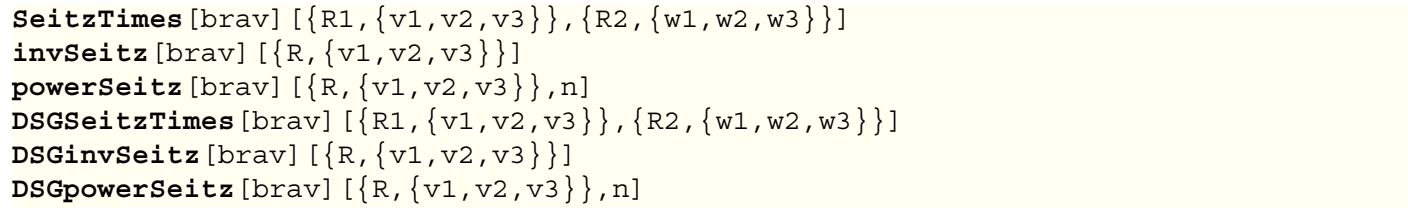
对于1/2自旋的旋转操作，我们用{srot,o3det}来描述，其中srot是一个SU(2)旋转矩阵，o3det是相应O(3)旋转矩阵的行列式（只能为1或）。同样，可以根据旋转名称获得自旋旋转操作，也可以根据自旋旋转操作获取旋转名称，由于每一个旋转名称对于所有格子都对应固定的旋转操作，前者不需要格子作为参数，但后者需要，例如：



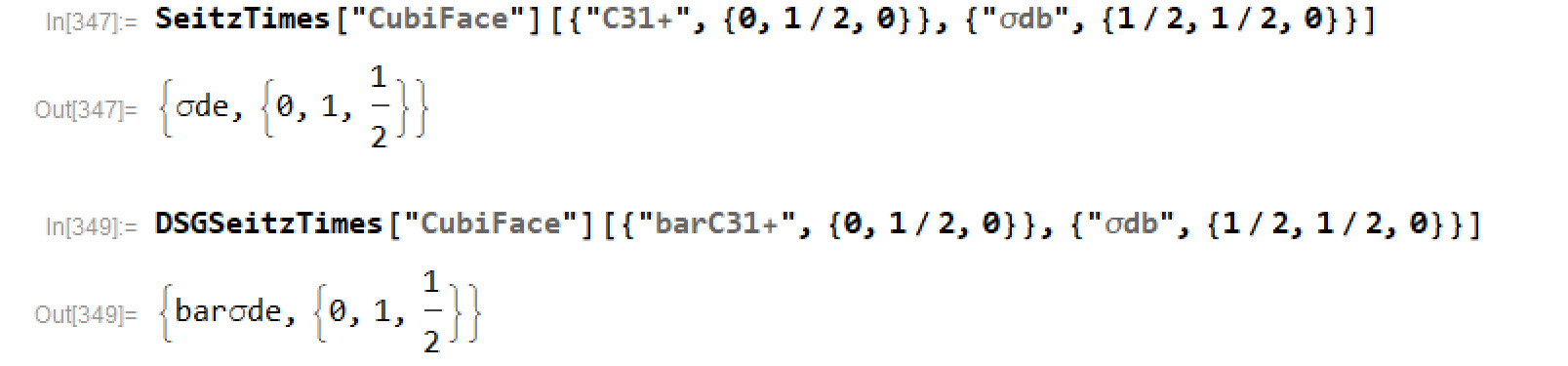
在代码中，空间群操作用{Rname,{v1,v2,v3}}来表示，其中Rname是旋转的名字字符串，{v1,v2,v3}是矢量的在原胞基矢下的坐标（或分量）。于是，可以用如下函数获取任意给定编号的空间群的群元，任意k点小群的群元，任意给定k点名称的Herring小群及小陪群的中心扩展的群元，而且都可以通过指定选项来获取相应双空间群版本的元素。注意，这里给出的小群元素实际上只是小群关于其平移群的陪集代表元。例如（关于下面各k点名称的定义见BC表3.6）：



两个空间群元素相乘，元素的逆，以及元素的乘方可通过如下函数计算



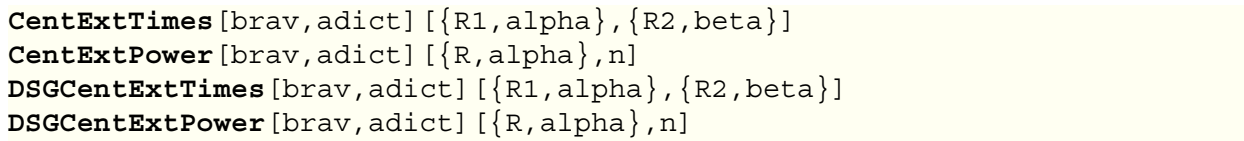
其中DSG开头的相应双空间群版本的计算函数，例如：



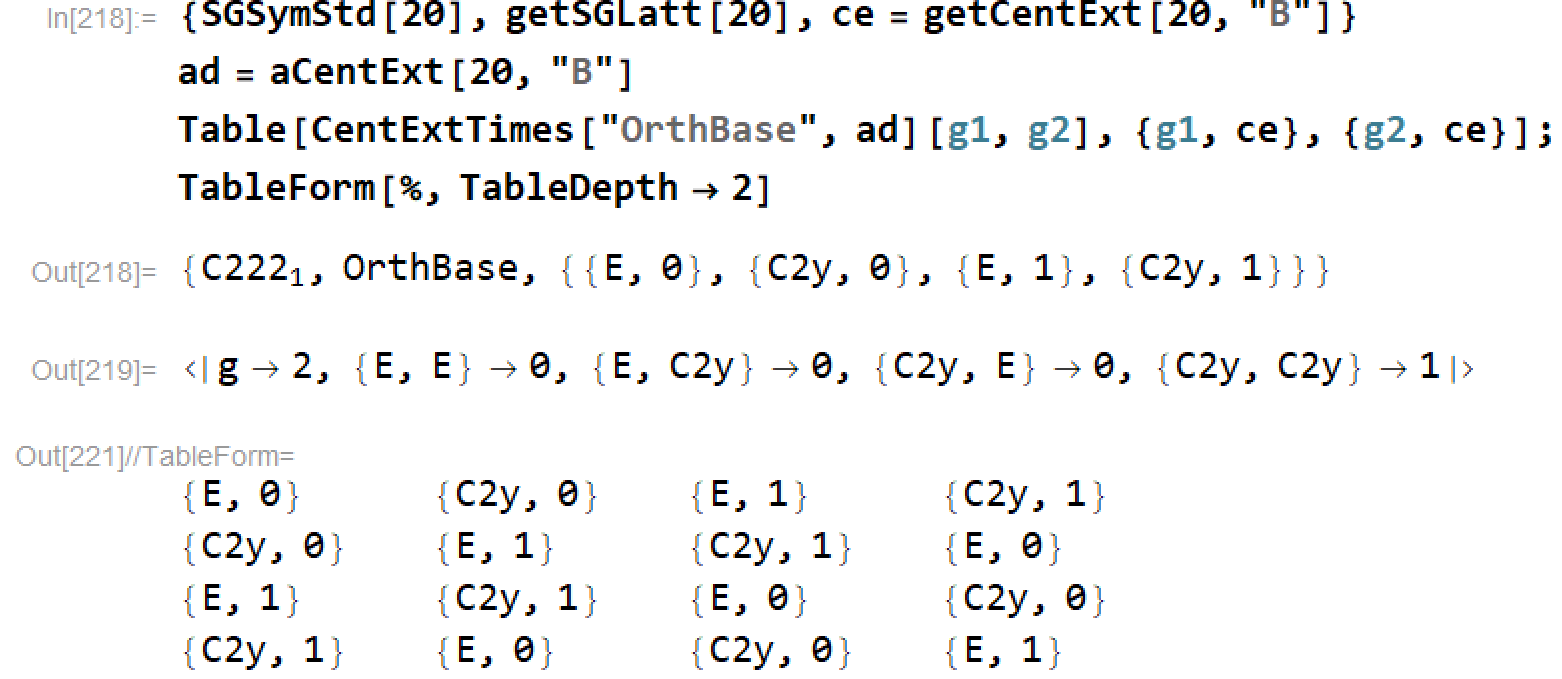
小陪群的中心扩展的元素 ( 用{Rname, alpha}来描述，其乘法规则为



其中的函数和整数可通过函数 aCentExt[sgno, kname] 获取，其中sgno是空间群号，kname是k点名称，若指定选项则得到的是相应双空间群版本，之后便可通过如下函数计算中心扩展元素间的乘法以及乘方，其中DSG开头的是相应双空间群版本



例如：



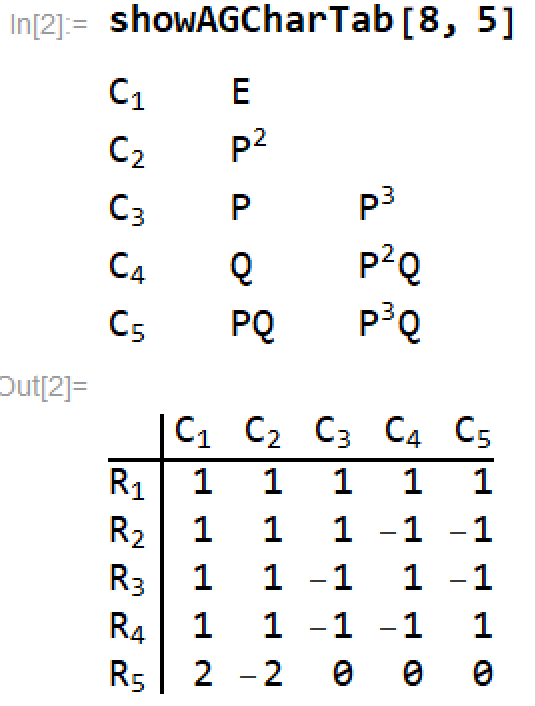
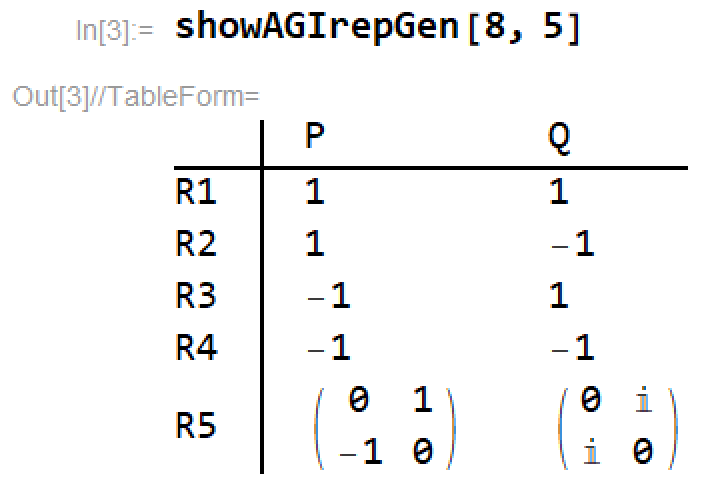
上面第一行给出20号空间群的符号、格子以及k点B处的中心扩展；第二行给出含有和整数信息的数据，用于计算中心扩展的乘法；第三、四行计算该中心扩展的乘法表并以表格方式显示。

#### 三、抽象群

BC书中通过抽象群来描述小群的不可约表示，BC表5.1中共给出了93个抽象群的特征标表和相应生成元的表示矩阵。所有抽象群的指标{m,n}可以通过allAGindex获取。BC表5.1中的抽象群数据存储在三个函数中

* AGClasses[m,n] 以生成元的幂指数方式给出抽象群的各类中元素
* AGCharTab[m,n] 给出抽象群的特征标表
* AGIrepGen[m,n] 给出抽象群的生成元在各不可约表示下的表示矩阵

可以通过showAGCharTab[m,n]和showAGIrepGen[m,n]来显示抽象群数据，例如：



上图左边给出抽象群的各类中元素及特征标表，右图给出其生成元的表示矩阵。

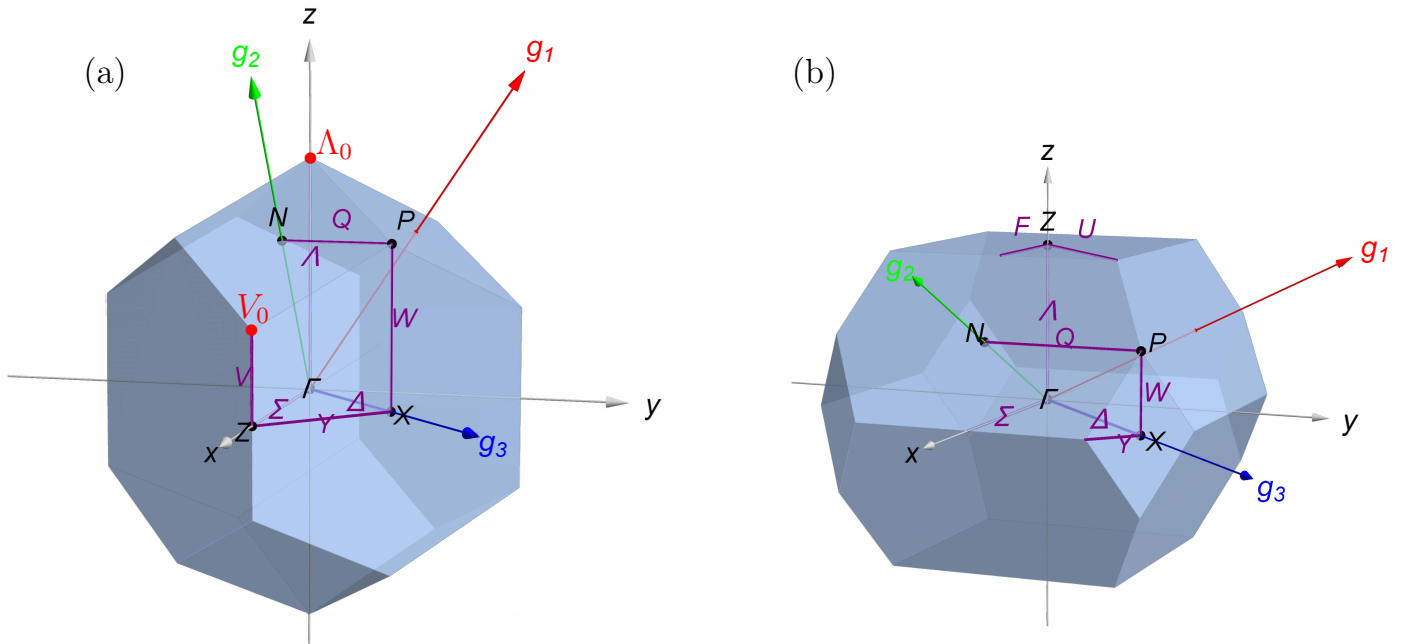
#### 四、计算并显示LGIR及SGIR表格

##### 1. 布里渊区和k点

要计算任意给定k点（或波矢星）的LGIR（SGIR），首先面临一个识别k点的问题。BC表3.6中定义了各种布里渊区下高对称点和高对称线上的k点名称，我们称之为BC标准k点，BC表5.7和6.11中给出了这些BC标准k点处的LGIR；然而若给定的k点不在这些BC标准k点之列，则无法直接根据BC表5.7和6.11得到LGIR，这时便需要找出它们与BC标准k点之间的关系，从而可以根据BC标准k点的LGIR及同构关系得到给定k点的LGIR。

我们将k点分为五种类型，任一给定 属于以下类型之一：

* **类型I**：就是BC标准k点或与BC标准k点等价，记为，其中表示等价（相差倒格矢），表示BC标准k点。
* **类型II**：与不等价但是与波矢星中一个k点等价，即存在一个元素使得，其中为所讨论的空间群。
* **类型III：**与波矢星中任意k点都不等价，但是存在一个不属于的元素满足以及。
* **类型IV**：一般k点（小陪群中仅含单位元）且不属于类型I到类型III，这样的k点我们统一命名为GP。
* **类型V**：不属于上述类型I到IV的k点，事实上这样的k点除了2号空间群外都是高对称面上的点，而对于2号空间群则为若干高对称点，这些k点不是一般点但对称性很低，而且BC书中并没有对其命名，我们统一命名为UN（即“未命名”的意思）。

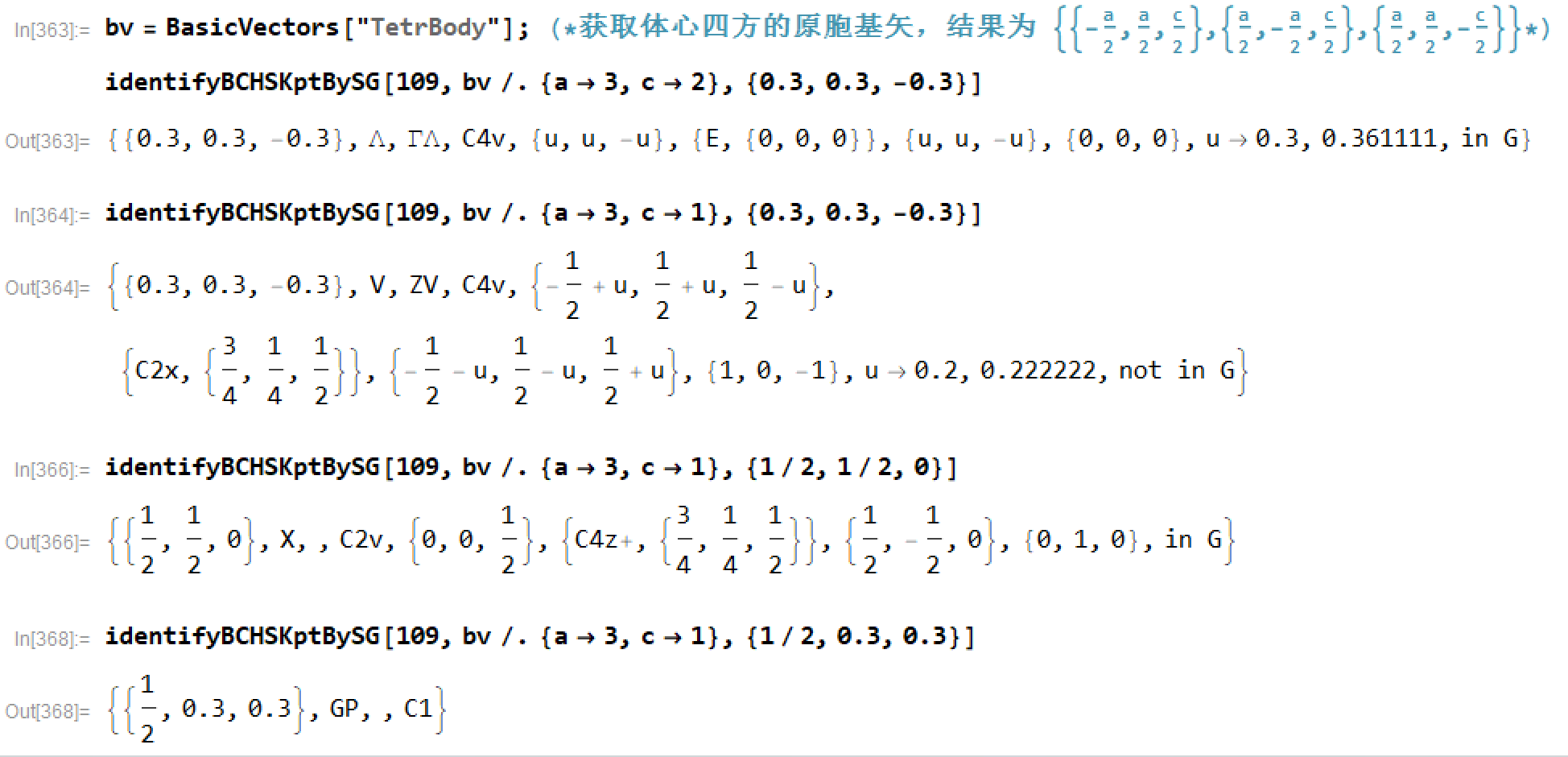


再来说下（第一）布里渊区（BZ）。对于三斜和单斜格子，由于其倒空间的魏格纳赛茨原胞的形状过于依赖具体晶格常数其比较复杂，BC书中不采用这样的BZ，而是简单的使用以点为中心的倒空间的平行六面体原胞作为BZ，而其他格子的BZ则依然使用通常的倒空间魏格纳赛茨原胞。不过，仍然有五种格子的BZ因具体的晶格常数比值不同而呈现出不止一种形状。这五种格子是：底心正交（BZ有a和b两种形状/类型），体心正交（BZ有a、b、c三种类型），面心正交（BZ有a、b、c、d四种类型），体心四方（BZ有a、b两种类型），简单三角（BZ有a、b两种类型）；BZ类型也在前面第二节的表格中列出了。这样，不同的BZ总共有22种。需要指出的是，对于上述五种多BZ的格子，有些名称的高对称线只在某种类型下出现，例如体心四方的格子当晶格常数时是上图的a类型，而时是上图的b类型，可以看出高对称线只在a中出现而b中没有，而和只在b中出现而a中没有。这种多BZ的格子还存在另外一个问题，那就是某些高对称线上的点可能被识别为具有不同名称的k点。还是以体心四方给子为例，对于a类型的BZ，当给定一个具体k点坐标为 (0.3,0.3,-0.3) 时，它符合图a中的坐标，是从到的高对称线，如果那么该点在线上且在BZ内部，但若则该点在的延长线上且已跑到BZ外部，而我们识别k点的要求是k点必须在BZ内部或边界上（不能跑到外部去），那么这时该k点便会被识别为类型III的而不是类型I的。至于到底是还是这依赖于的具体值，而该值依赖于具体晶格常数的值。在没有明确给出晶格常数（或原胞基矢）的情况下，原则上是无法准确确定该点到底是还是，但为了有一个明确的输出，我们采取的方案是给一个权宜之值1/4。

识别k点的函数为identifyBCHSKptBySG，它的输出是一个列表

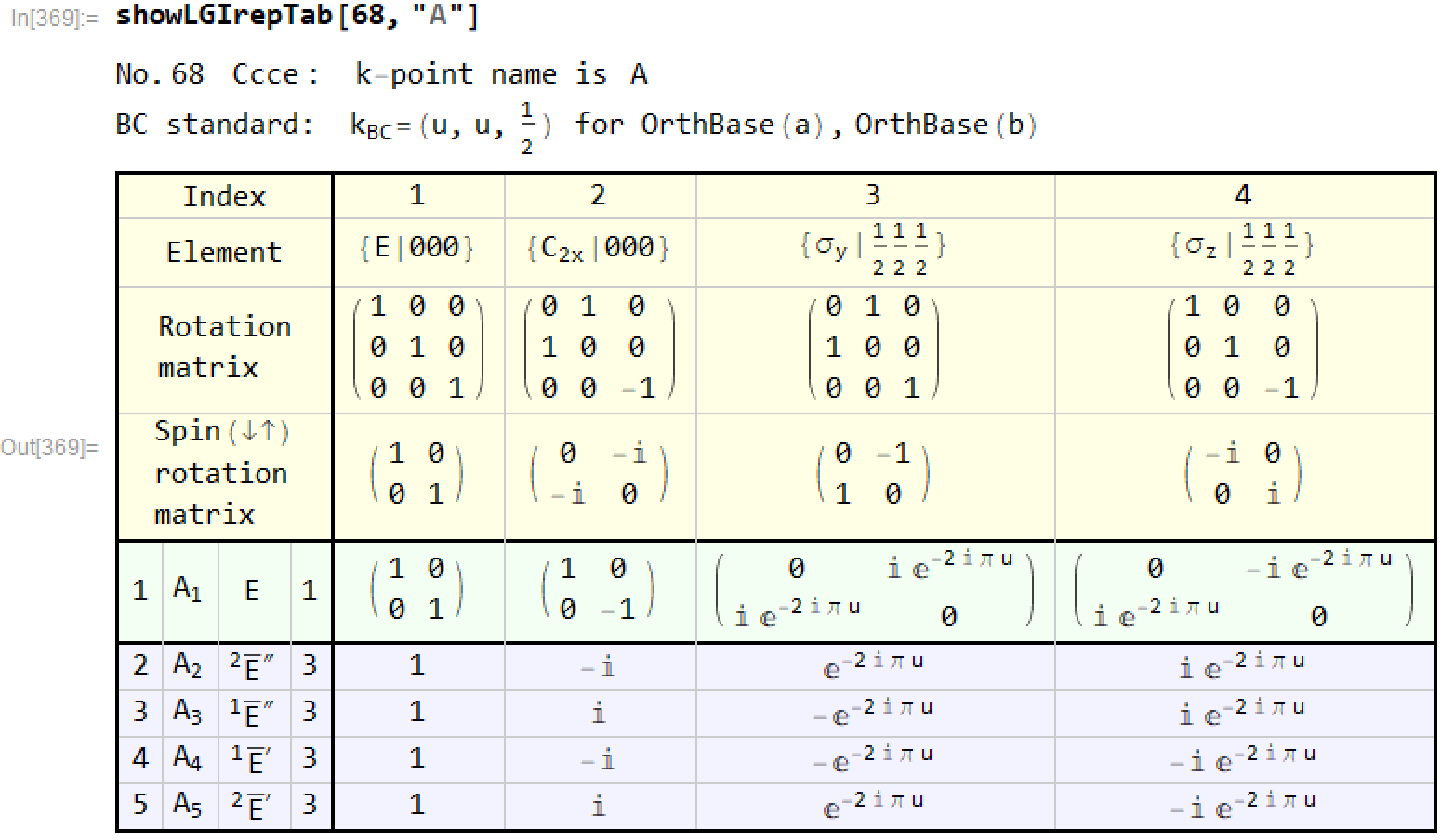


其中是输入的的k点数值坐标；kname是识别出的k点名称（注意：若是类型II或III的k点，严格说来这个名称只是与相关联的的名称，这里仅仅是“借用”相关的名称而已，如有必要时可给它一个自定义的名称与的名称加以区别）；line\_info给出高对称线的连接信息，是该k点的小陪群；是与之关联的BC标准k点的坐标，即前面k点分类中所提到的元素，用于将与关联起来（使它们的小群同构）；；是高对称线上的具体取值；是使高对称线上的k点不超出BZ的的最大取值（即前面例子中的），ifinG指出是（类型II）否（类型III）在空间群内。对于高对称点，输出结果中没有和两项；对于类型IV和类型V的k点 则只有前四项。具体例子如下（以体心正交格子的109号空间群为例）：

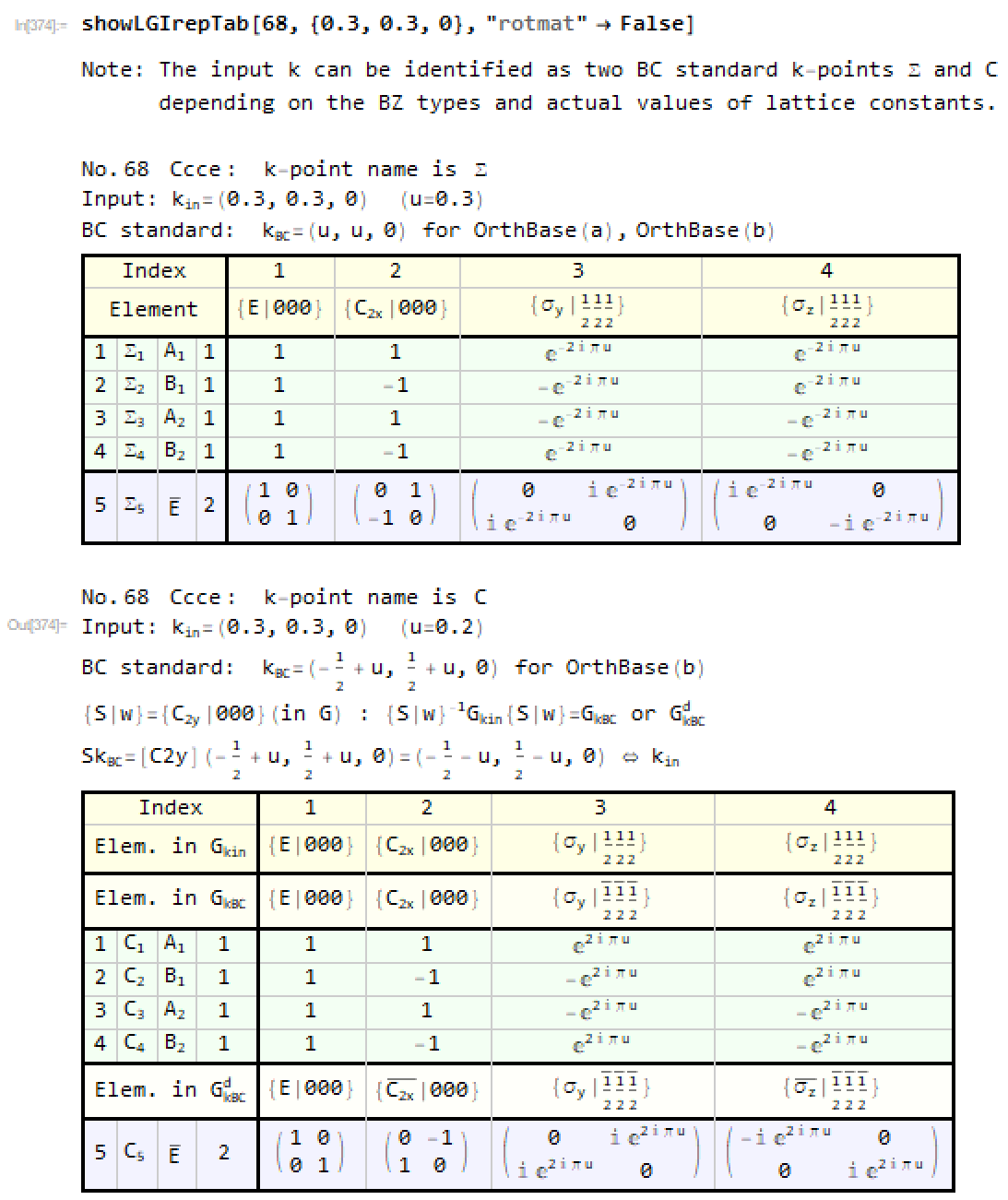


#### 2. 获取LGIR的表格

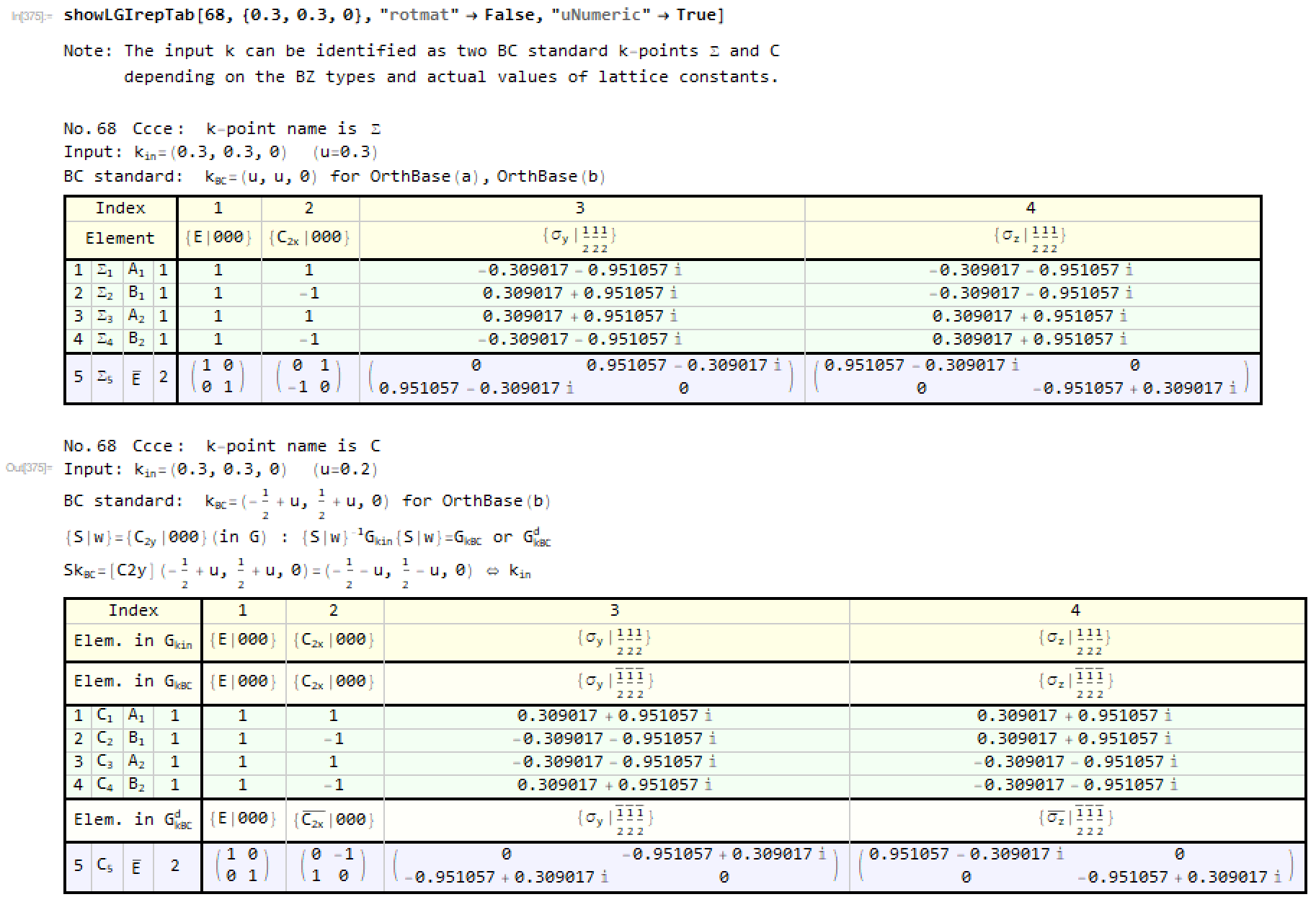
getLGIrepTab[sgno, k] 和 showLGIrepTab[sgno, k] 函数分别计算并显示LGIR表格，其中sgno是空间群号，k既可以是数值的k点坐标也可以是BC标准k点的名称。例如：68号空间群Ccce的A点小群的不可约表示表格见下图，表中浅绿色背景的是单值表示，浅蓝色背景的是双值表示。需要指出的是，对于双值表示，由于其带bar的元素的表示矩阵只是其相应不带bar元素表示矩阵的负值，即，故为节省空间，表格中只给出了不带bar元素的表示矩阵。showLGIrepTab函数有一些选项，具体可以通过Options[showLGIrepTab]来查看其默认值。比如可以不显示旋转矩阵，只显示特征标，只显示第1、2、4个元素等。



上面表格中表示的第一列是一个自定义的序号，第二列是LGIR的符号，第三列是LGIR的扩展Mulliken符号，第四列是相应SGIR的实性（可能取值1、2、3、x，具体含义参见BC书）。对于某些高对称线上的k点，给定的坐标可能被识别为两个不同的名称，这时会给出两个表格，比如还是68号空间群，(0.3,0.3,0)这一k点可以识别为或C点，见下图：

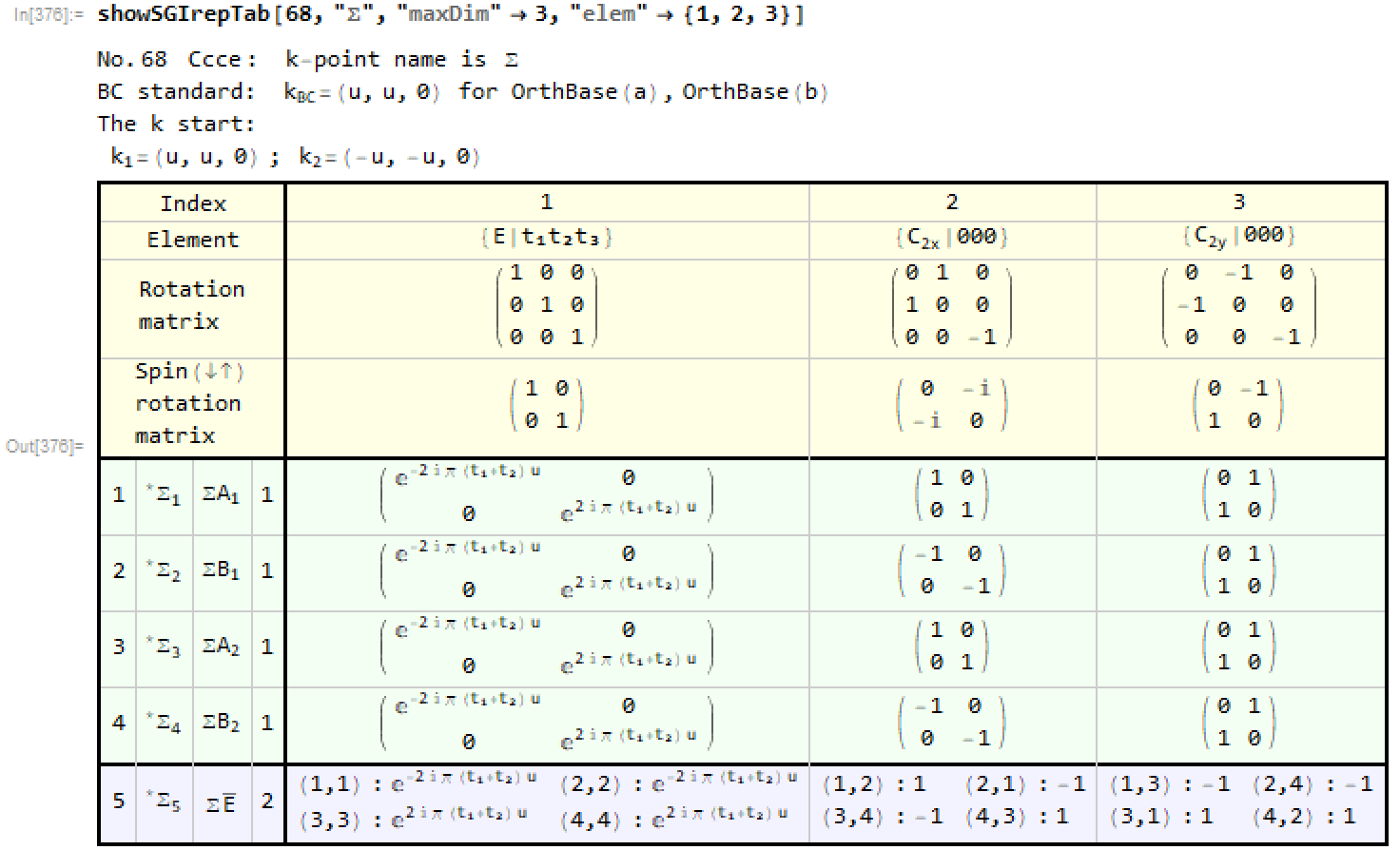


乍一看似乎识别为或C后其表示并不一样，那是因为值不一样，如果把具体的值代入，令表示数值化，则可以看出表示是一样的，只是顺序变了，即等价于，等价于，等价于，等价于，等价于，见下图。



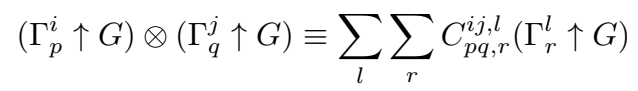
#### 3. 获取SGIR表格

有了LGIR后，构造诱导表示即可得到相应k点所在波矢星所对应的SGIR，计算SGIR通过getSGIrepTab[sgno,k]实现，其结果通过showSGIrepTab[sgno,k]来显示，例如：

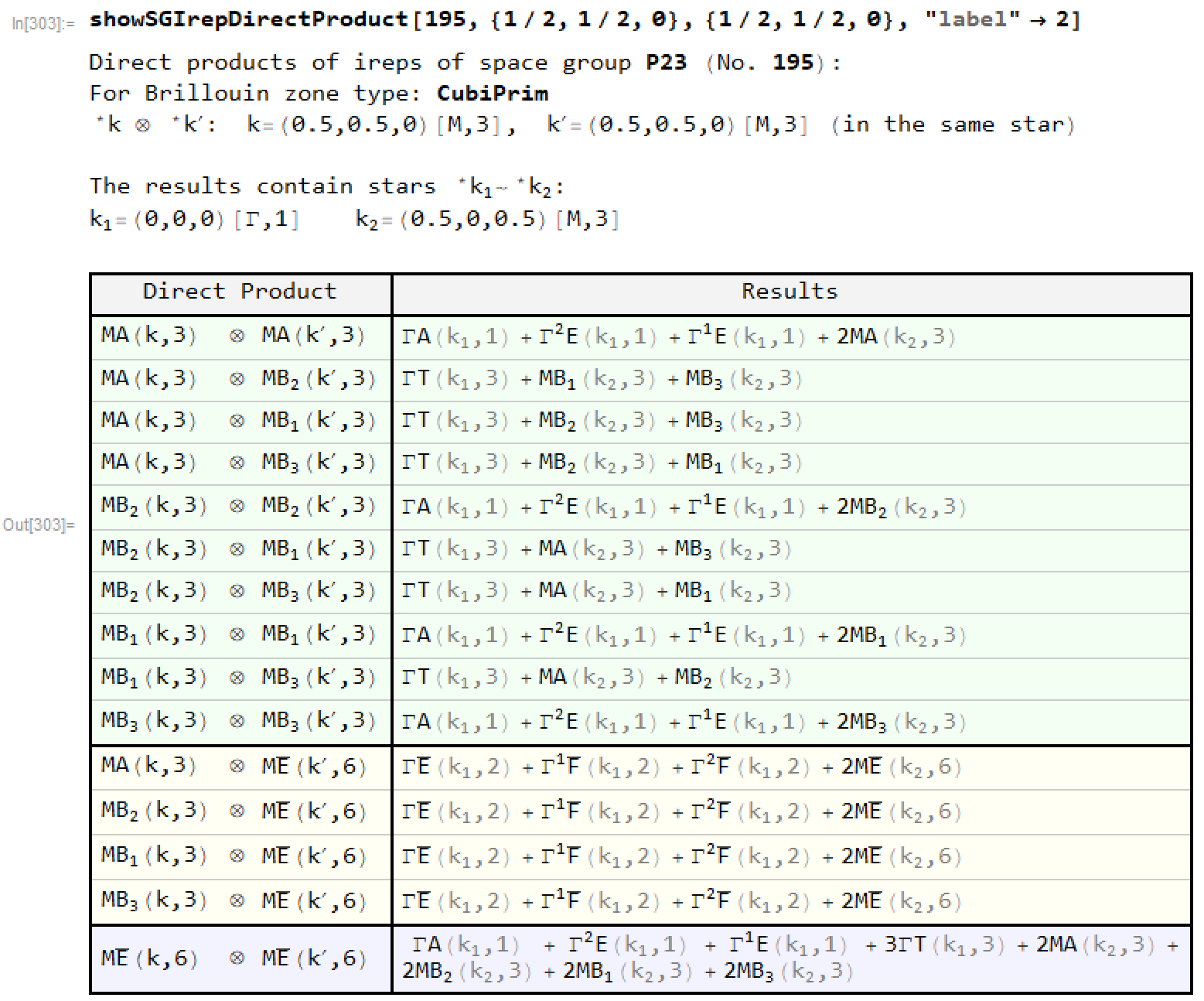
上图例子中由于空间限制只输出了前三个群元的表示矩阵，而且通过选项指定当表示矩阵维数大于3时将只显示非零矩阵元（该选项默认值为4）。与LGIR表格类似，表示的的第一列是序号，第二、三列是标记SGIR的两种不同符号，第四列是SGIR的实性。

#### 五、计算SGIR的直积分解

设和分别是空间群的属于和的LGIR，那么它们诱导得出的SGIR则分别为和，将它们的直积进行分解（约化）则有



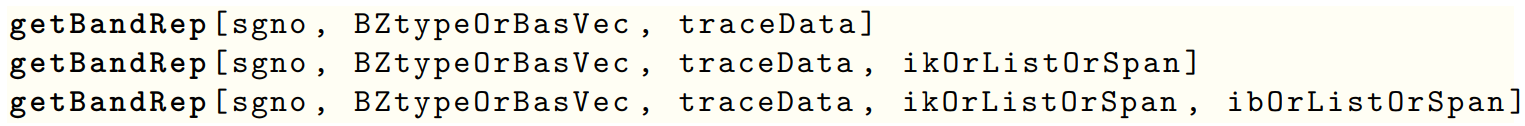
根据BC书中公式(4.7.29)可以计算出上面分解中的约化系数。实现该功能的函数为SGIrepDirectProduct[sgno,k1,k2]，而以表格方式显示分解结果的函数为showSGIrepDirectProduct[sgno,k1,k2]，其中sgno是空间群号，k1和k2分别是两个SGIR所对应的波矢星里的任一k点的数值坐标。使用BC书中的例子，即195号空间群（P23）的 波矢星和 波矢星间的SGIR直积分解如下，结果与BC书中一致：



上面图中表格上方的中括号内分别是识别出的k点名称和相应波矢星中k点个数，分解结果中的小括号内分别是所对应的波矢星代表k点和相应SGIR的维数。

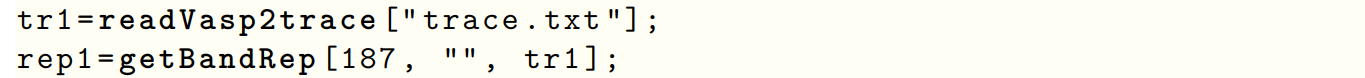
#### 六、判定能带的LGIR

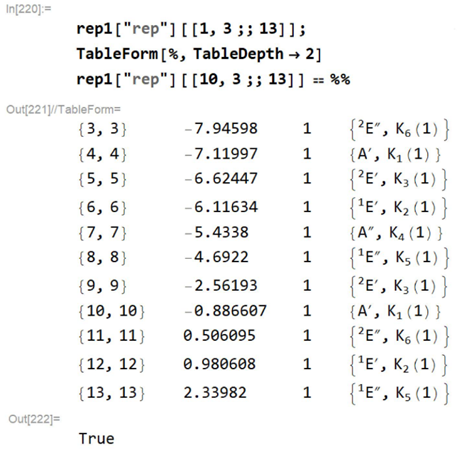
要确定能带的LGIR原则上只需要两个步骤，首先是计算小群中各元素在给定简并布洛赫态处的特征标，然后是查找相应小群的特征标表判定能级所属的LGIR。所幸的是，对于VASP计算的能带，第一个步骤已有现成的工具vasp2trace（https://www.cryst.ehu.es/html/cryst/  
topological/vasp2trace.tar.gz）可以实现，虽然该工具本是为判断能带的拓扑性质而开发，但其所作的事情恰好是这里所说的第一步，只不过默认情况下它只计算占据态的特征标，因为判断拓扑性质只需要占据态的信息即可，但这里需要计算所有能带的特征标，故而需要对其进行两处简单修改：wrtir.f90文件第30行中的nele改为ne；删掉或注释掉chrct.f90文件中的第55行，即IF(IE>nele) EXIT。VASP算出能带后，用vasp2trace工具进行后处理，会计算小群的特征标并将相关数据写入trace.txt文件中。SpaceGroupIrep程序包里提供readVasp2trace函数读取trace.txt文件中的数据，然后将其传给getBandRep函数即可判定能带的LGIR。getBandRep有三种调用方式

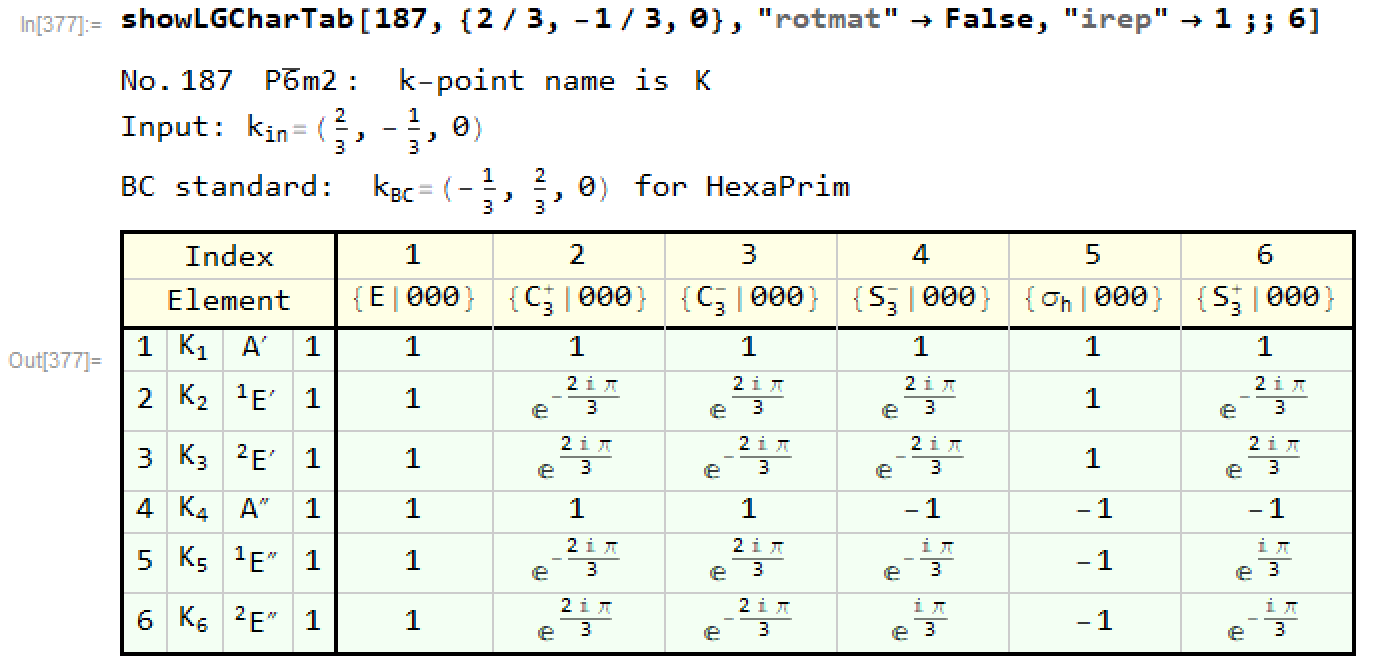


其中sgno是空间群号；BZtypeOrBasVec可以是简单的BZ类型（比如或空字符串等）也可以是数值的原胞基矢，若是BZ类型则识别出的k点可能跑到BZ外部（参见前面识别k点的讨论），若是数值的原胞基矢则不会跑到BZ外部；traceData是readVasp2trace的返回值，ikOrListOrSpan（ibOrListOrSpan）是可选参数，指定计算的k点（能带），既可以是一个数（比如第5个k点就写5），也可以是一个列表（比如{1,3,5}），还可以是一个跨度（比如2;;6）。如果不指定这两个参数则计算所有k点和能带的LGIR。

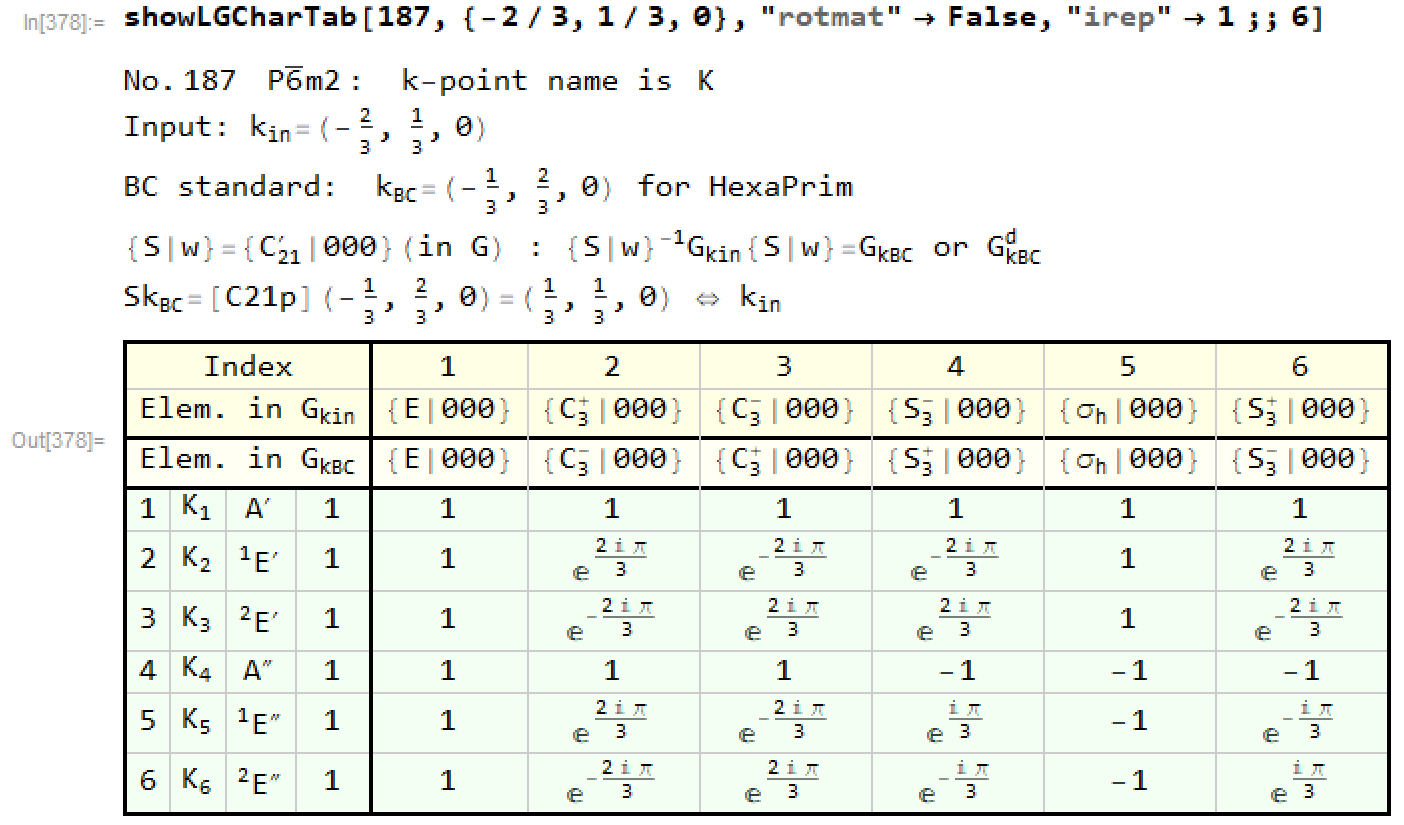
以单层MoS2体系为例，其空间群是（187号），先用VASP计算能带，然后用vasp2trace计算并得到trace.txt文件，把trace.txt文件放到工作目录下便可执行如下命令



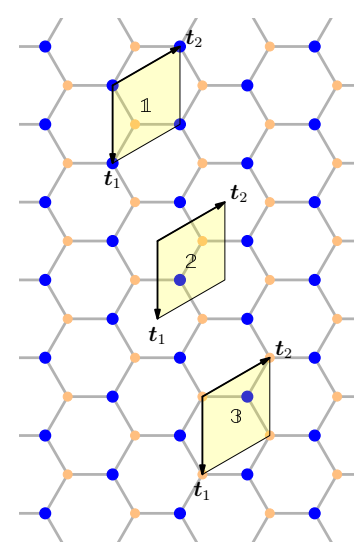
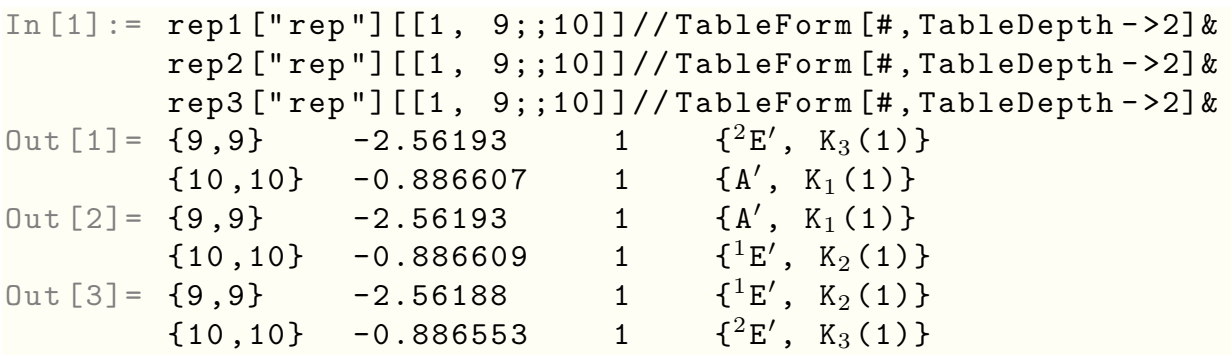
获取能带的LGIR，得到的LGIR数据便存储在rep1[]中。该例中的能带，其第1个k点是，第10个k点是，右图中给出了点处第3到13个能带的LGIR（第3到9个是价带，第10到13个是导带）。右图结果中第一列给出的是简并的能级起止编号（比如{3,5}表示第3到第5个能级简并，若是{3,3}则表示第3个能级不简并），第二列是能级（单位eV），第三列是能级简并度，第四列则是判定出的LGIR，每组中前一个是扩展的Mulliken符号，后一个是符号并在括号中标出了表示的维数。根据右图第三条命令的结果（True）可以看出，点（第10个k点）的LGIR与点的LGIR竟然完全一样！**这其实是个假象**！因为这里的点是BC标准k点，而点则不是BC标准k点，它们的特征标表并不一样。点的特征标表如下（showLGCharTab就是用了选项的showLGIrepTab）：



而点的特征标表如下：



以最高价带（第9个能带为例），虽然rep1[][[1,9]]（点）和rep1[][[10,9]]（点）给出的LGIR都是，但是其的特征标并不一样，点的是，而点的则是，二者互为复共轭，这正好与与互为时间反演相一致。不要忘了，由于不是BC标准k点，它的特征标表中的符号**仅仅是借用**相应的BC标准k点的名称而已，所以为了明确加以区分，这里的实为。

值得指出的是，能带的LGIR是什么依赖于原胞的选取。还是以MoS2为例，上图中左边给出了三种不同的原胞取法，区别仅在于原点的选取不同，因而旋转中心不同（旋转中心总是在原点处），但是三种原胞取法得出的空间群操作却是完全一样的，都符合BC约定。若是分别用这三个原胞进行VASP能带计算并通过vasp2trace得到相应trace.txt文件，再用getBandRep判定LGIR，可以看出同一能级的LGIR并不相同。以点的最高价带（第9条能带）和最低导带（第10条能带）为例，上图中右边给出了三种原胞下对应的LGIR，可以看出并不相同！所以，原则上若要问或说某个能级的LGIR是什么，应先明确指出原胞如何取。

#### 非BC原胞

前面所述用getBandRep判定LGIR的过程其实有个大前提，那就是VASP算能带时所用的原胞必须是BC原胞。所谓BC原胞指的是一方面其原胞基矢必须按BC表3.1来定义，另一方面该原胞所确定的空间群元素必须与BC表3.7中（每个空间群的第一行数据）给出的生成元生成的空间群元一致。如果仅满足原胞基矢按BC表3.1定义则未必是BC原胞。对于非BC原胞，其生成的trace.txt文件经readVasp2trace读取后不能直接给getBandRep使用，必须经过原胞转换，将非BC原胞转换成BC原胞并将相应的trace.txt数据也转成BC原胞对应的数据后才能给getBandRep使用。这一转换过程通过如下函数实现，



其中 traceData 是readVasp2trace 直接读取的非BC原胞生成的trace.txt后得到的数据，其中 P, p, stdR 则分别是第三方工具spglib将给定原胞标准化和理想化过程中给出的标准化变换矩阵、原点移动和理想化旋转矩阵。经这样转换后的trace数据便可直接传入getBandRep用于判断LGIR了。若是对spglib不熟悉，不用担心，我们还提供了自动调用外部spglib的python库并进行转换的函数：



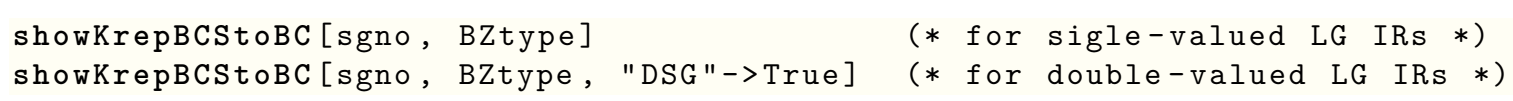
只需要给定非BC原胞的VASP结构文件和相应trace.txt文件经readVasp2trace读取得到的traceData，那么上述函数返回的结果就已是BC原胞的trace数据了，可以传给getBandRep使用了(注：1.0.4之前的版本的返回结果需要取其Key为”trace”的关联分量才是可直接传给getBandRep的值)。此外需要指出的是，**即使是非BC原胞，也必须是“原胞”而不能是“超胞”**，否则计算会出错，因为原胞和超胞的布里渊区及k点定义都是不一样的而小表示都是根据原胞定义的而不是超胞。

#### 七、k点和LGIR符号在BCS约定与BC约定间的对应

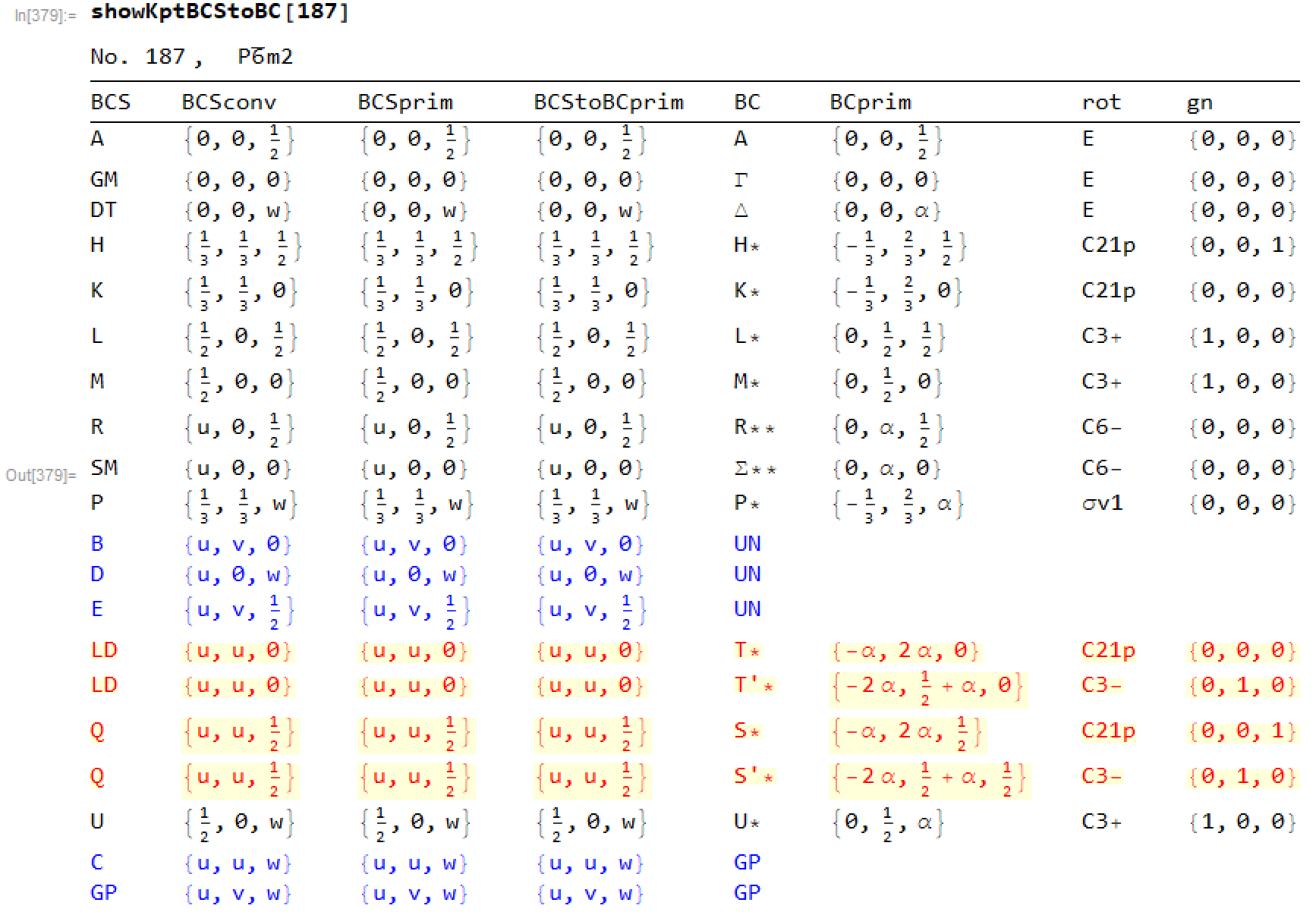
借助irvsp软件，从其输出结果中我们收集了BCS约定下k点以及LGIR的特征标表等数据，据此我们给出BCS约定下的k点和LGIR与BC约定间的对应关系。可使用如下函数给出k点间对应的表格



使用下面函数给出LGIR间对应的表格

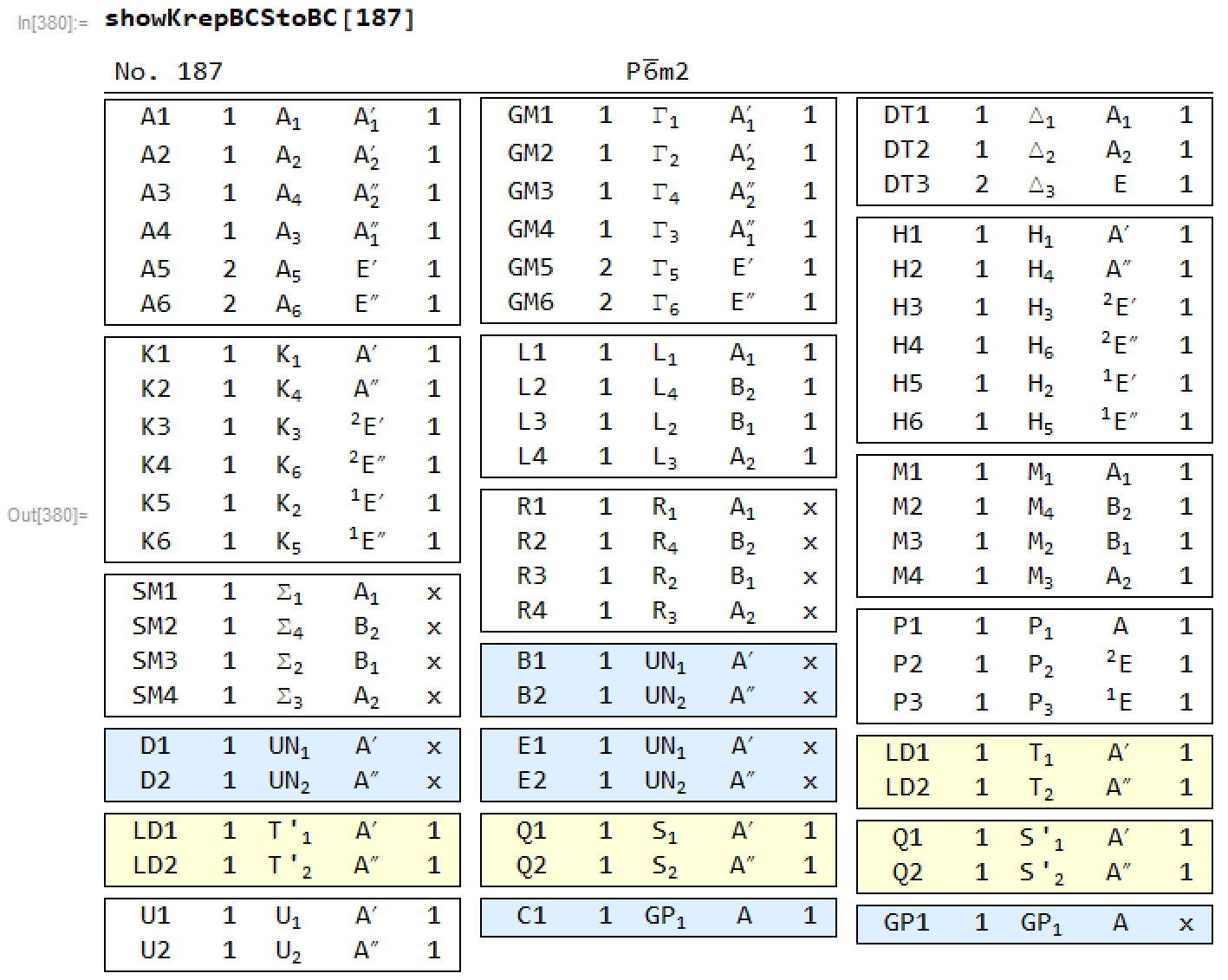


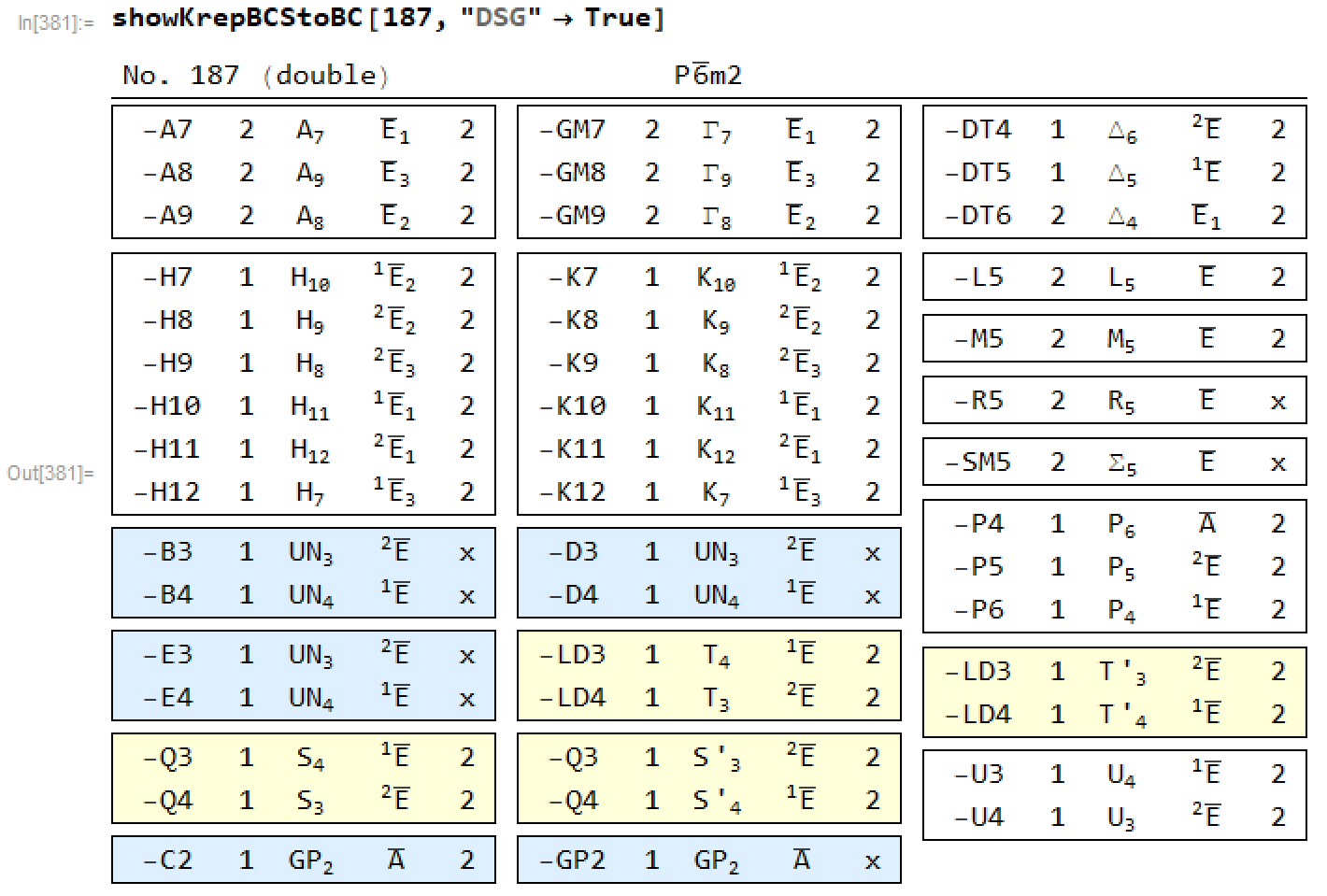
其中 BZtype 可选，若不指定则对于多BZ的格子默认取。具体例子如下：



上图给出了187号空间群BCS约定及BC约定下的k点间对应。第一列是BCS的k点名称；第二、三列都是BCS的k点坐标，只不过第二列是传统晶胞下的坐标而第三列是原胞下的坐标；第四列是BCS的原胞转换成BC原胞后得到的k点坐标，记为；第五列是对第四列识别出的BC下k点名称（一个\*是类型II的k点，两个\*\*是类型III的k点）；第六列是相应BC标准k点的坐标；第七列是中的，第八列是。上图中GP和UN用蓝色显示，BCS和BC约定填的k点名称不同的用红色显示，而一个BCS的k点被识别为两个不同名称的BC k点的情况用浅黄色背景显示。

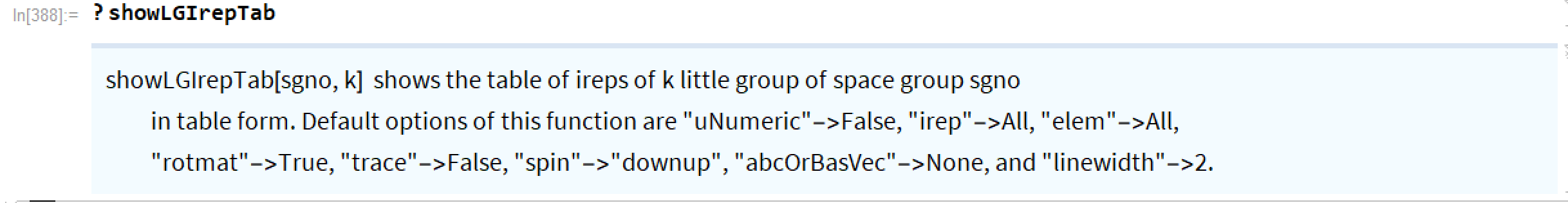
下页图中则给出了187号空间群的单值（上图）和双值（下图）LGIR在BCS约定与BC约定间的对应。第一列是BC约定下标记LGIR的符号，第二列是表示的维数，第三列是BC约定下的符号，第四列是扩展的Mulliken符号，第五列是相应SGIR的实性。类似的，浅蓝色背景对应GP和UN点，浅黄色背景则对应一个BCS k点被识别为两个BC k点的情况。



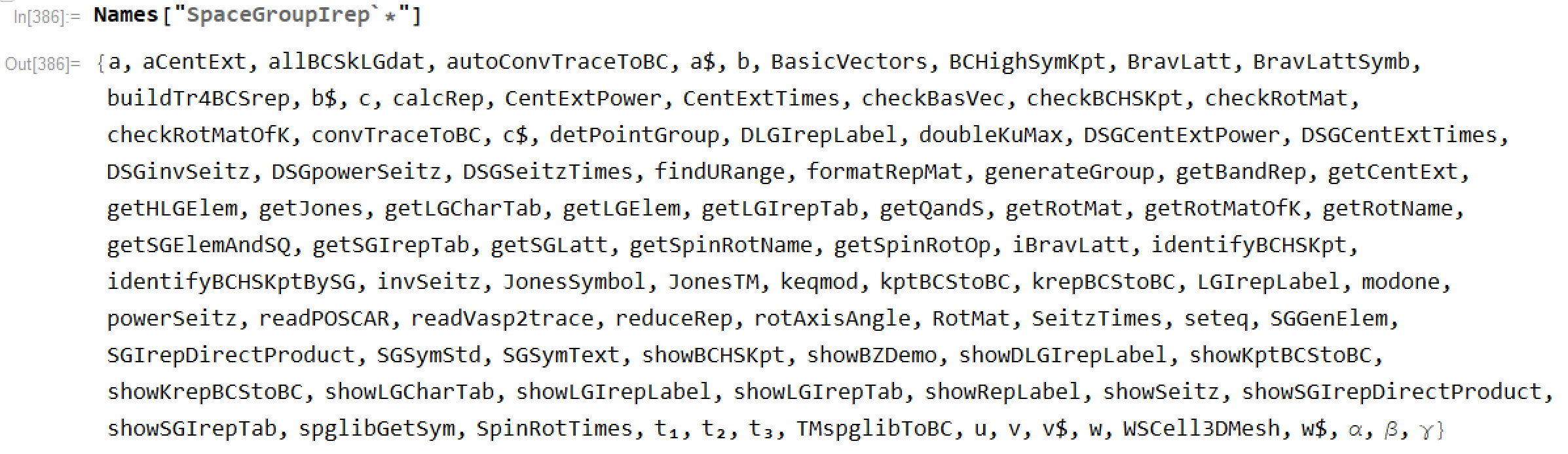


#### 八、其他

SpaceGroupIrep程序包内任意函数可通过“?”来查看其简要说明，例如



而该程序包中都包含哪些函数和变量可通过如下命令进行了解



可见除了前面介绍的主要功能外，SpaceGroupIrep程序包里 包含一些其他实用的函数，比如：**showBZDemo**可以显示可旋转的三维布里渊区及高对称点的定义，**generateGroup**可以根据生成元、乘法规则和单位元来生成整个群，**rotAxisAngle**可以给出任意O(3)旋转矩阵的行列式、转轴和转角。

关于SpaceGroupIrep程序包的理论文章发表在:

Gui-Bin Liu *et al.*, Comput. Phys. Commun. **265**, 107993 (2021)

<https://doi.org/10.1016/j.cpc.2021.107993>