Misorientation Analysis

R. Hielscher

Faculty of Mathematics,
Chemnitz University of Technology, Germany

MTEX Workshop 2015

Coordinate Transforms

Consider a Magnetite grain in cube orientation

```
CS_Mag = loadCIF('Magnetite')
O_Mag  = orientation.id(CS_Mag)
```

```
CS_Mag = crystalSymmetry (show methods, plot)
  mineral : Magnetite
  symmetry: m-3m
  a, b, c : 8.4, 8.4, 8.4

O_Mag = orientation (show methods, plot)
  size: 1 x 1
  crystal symmetry : Magnetite (m-3m)
  specimen symmetry: 1

  Bunge Euler angles in degree
  phi1   Phi  phi2  Inv.
  0      0    0     0
```

Remember, orientations convert crystal into specimen coordinates.

```
r = O_Mag * Miller(1,1,1,CS_Mag)
```

Take a Hematite grain with orientation

```
CS_Hem = loadCIF('Hematite')
```

Coordinate Transforms

Consider a Magnetite grain in cube orientation

```
CS_Mag = loadCIF('Magnetite')
O_Mag  = orientation.id(CS_Mag)
```

Remember, orientations convert crystal into specimen coordinates.

```
r = O_Mag * Miller(1,1,1,CS_Mag)
```

```
r = vector3d (show methods, plot)
 size: 1 x 1
          x         y         z
  0.119107  0.119107  0.119107
```

Take a Hematite grain with orientation

```
CS_Hem = loadCIF('Hematite')
O_Hem  = orientation('Euler',...
              135*degree,55*degree,60*degree,CS_Hem)
```

inverse orientations convert specimen into crystal coordinates

```
inv(O_Hem) * r
```

Coordinate Transforms

Consider a Magnetite grain in cube orientation

```
CS_Mag = loadCIF('Magnetite')
O_Mag  = orientation.id(CS_Mag)
```

Remember, orientations convert crystal into specimen coordinates.

```
r = O_Mag * Miller(1,1,1,CS_Mag)
```

Take a Hematite grain with orientation

```
CS_Hem = loadCIF('Hematite')
O_Hem  = orientation('Euler',...
              135*degree,55*degree,60*degree,CS_Hem)
```

```
O_Hem = orientation (show methods, plot)
  size: 1 x 1
  crystal symmetry : Hematite (-3m1, X||a*, Y||b, Z||c)
  specimen symmetry: 1

  Bunge Euler angles in degree
  phi1      Phi     phi2     Inv.
   135  54.7356       60        0
```

inverse orientations convert specimen into crystal coordinates

Coordinate Transforms

Consider a Magnetite grain in cube orientation

```
CS_Mag = loadCIF('Magnetite')
O_Mag  = orientation.id(CS_Mag)
```

Remember, orientations convert crystal into specimen coordinates.

```
r = O_Mag * Miller(1,1,1,CS_Mag)
```

Take a Hematite grain with orientation

```
CS_Hem = loadCIF('Hematite')
O_Hem  = orientation('Euler',...
           135*degree,55*degree,60*degree,CS_Hem)
```

inverse orientations convert specimen into crystal coordinates

```
inv(O_Hem) * r
```

```
ans = Miller (show methods, plot)
 mineral: Hematite (-3m1, X||a*, Y||b, Z||c)
  h 0
  k 0
  i 0
  l 1
```

Misorientations

Hence, **inv(O_Hem)** · **O_MAG** converts crystal into crystal coordinates

```
inv(O_Hem) * O_Mag * Miller(1,1,1,0,CS_Mag)
```

```
ans = Miller (show methods, plot)
 size: 1 x 1
 mineral: Hematite (-3m1, X||a*, Y||b, Z||c)
  h 0
  k 0
  i 0
  l 1
```

A misorientation transforms coordinates with respect to one crystal into coordinates with respect to another crystal.

```
Mag2Hem = inv(O_Hem) * O_Mag
```

As **O_Mag** and **O_Hem** are suspect to crystal symmetry, there are many symmetrically equivalent misorientations to **Mag2Hem**.

```
Mag2Hem.symmetrise
```

Misorientations

Hence, **inv(O_Hem)** $\cdot$ **O_MAG** converts crystal into crystal coordinates

```
inv(O_Hem) * O_Mag * Miller(1,1,1,0,CS_Mag)
```

A misorientation transforms coordinates with respect to one crystal into coordinates with respect to another crystal.

```
Mag2Hem = inv(O_Hem) * O_Mag
```

```
Mag2Hem = misorientation (show methods, plot)
  size: 1 x 1
  crystal symmetry : Magnetite (m-3m)
  crystal symmetry : Hematite (-3m1, X||a*, Y||b, Z||c)

  Bunge Euler angles in degree
  phi1  Phi phi2 Inv.
   120   55   45    0
```

As **O_Mag** and **O_Hem** are suspect to crystal symmetry, there are many symmetrically equivalent misorientations to **Mag2Hem**.

```
Mag2Hem.symmetrise
```

Misorientations

Hence, **inv(O_Hem)** · **O_MAG** converts crystal into crystal coordinates

```
inv(O_Hem) * O_Mag * Miller(1,1,1,0,CS_Mag)
```

A misorientation transforms coordinates with respect to one crystal into coordinates with respect to another crystal.

```
Mag2Hem = inv(O_Hem) * O_Mag
```

As **O_Mag** and **O_Hem** are suspect to crystal symmetry, there are many symmetrically equivalent misorientations to **Mag2Hem**.

```
Mag2Hem.symmetrise
```

```
Mag2Hem = misorientation (show methods, plot)
  size: 576 x 1
  crystal symmetry : Magnetite (m-3m)
  crystal symmetry : Hematite (-3m1, X||a*, Y||b, Z||c)
```

```
Mag2Hem.symmetrise('unique')
```

Misorientations

Hence, **inv(O_Hem)** · **O_MAG** converts crystal into crystal coordinates

```
inv(O_Hem) * O_Mag * Miller(1,1,1,0,CS_Mag)
```

A misorientation transforms coordinates with respect to one crystal into coordinates with respect to another crystal.

```
Mag2Hem = inv(O_Hem) * O_Mag
```

As **O_Mag** and **O_Hem** are suspect to crystal symmetry, there are many symmetrically equivalent misorientations to **Mag2Hem**.

```
Mag2Hem.symmetrise
```

```
Mag2Hem.symmetrise('unique')
```

```
Mag2Hem = misorientation (show methods, plot)
  size: 96 x 1
  crystal symmetry : Magnetite (m-3m)
  crystal symmetry : Hematite (-3m1, X||a*, Y||b, Z||c)
```

Phase Transitions

The phase transition from **O_Mag** to **O_Hem** is characterized by $\{111\}_m||\{0001\}_h$ and $\{\bar{1}01\}_m||\{00\bar{1}0\}_h$

```
Mag2Hem = orientation('map',...
  Miller(1,1,1,CS_Mag),Miller(0,0,0,1,CS_Hem),...
  Miller(-1,0,1,CS_Mag),Miller(1,0,-1,0,CS_Hem))
```

```
Mag2Hem = misorientation (show methods, plot)
  size: 1 x 1
  crystal symmetry : Magnetite (m-3m)
  crystal symmetry : Hematite (-3m1, X||a*, Y||b, Z||c)

  Bunge Euler angles in degree
  phi1 Phi phi2 Inv.
  120 54.7356 45 0
```

For measured hematite orientation **O_Hem** we can compute the initial magnetite orientation by

```
O_Mag = O_Hem * Mag2Hem
```

MTEX keeps track about the symmetries throughout all computations and

Phase Transitions

The phase transition from **O_Mag** to **O_Hem** is characterized by $\{111\}_m||\{0001\}_h$ and $\{\bar{1}01\}_m||\{00\bar{1}0\}_h$

```
Mag2Hem = orientation('map',...
  Miller(1,1,1,CS_Mag),Miller(0,0,0,1,CS_Hem),...
  Miller(-1,0,1,CS_Mag),Miller(1,0,-1,0,CS_Hem))
```

For measured hematite orientation **O_Hem** we can compute the initial magnetite orientation by

```
O_Mag = O_Hem * Mag2Hem
```

```
O_Mag = orientation (show methods, plot)
  size: 1 x 1
  crystal symmetry : Magnetite (m-3m)
  specimen symmetry: 1

  Bunge Euler angles in degree
     phi1      Phi     phi2     Inv.
  134.739 54.7356       45        0
```

MTEX keeps track about the symmetries throughout all computations and

Phase Transitions

The phase transition from **O_Mag** to **O_Hem** is characterized by $\{111\}_m||\{0001\}_h$ and $\{\bar{1}01\}_m||\{00\bar{1}0\}_h$

```
Mag2Hem = orientation('map',...
  Miller(1,1,1,CS_Mag),Miller(0,0,0,1,CS_Hem),...
  Miller(-1,0,1,CS_Mag),Miller(1,0,-1,0,CS_Hem))
```

For measured hematite orientation **O_Hem** we can compute the initial magnetite orientation by

```
O_Mag = O_Hem * Mag2Hem
```

MTEX keeps track about the symmetries throughout all computations and warns in case of mismatch.

Phase Transition

We should care about symmetric equivalence

```
O_Mag = O_Hem * symmetrise(Mag2Hem, 'unique')
```

```
ori_Mag = orientation (show methods, plot)
  size: 1 x 96
  crystal symmetry : Magnetite (m-3m)
  specimen symmetry: 1
```

How many are crystallographically not equivalent?

```
unique(O_Mag)
```

Pole figures of misorientations

```
plotPDF(Mag2Hem, Miller({0 0 0 1},{1 1 -2 1},CS_Hem))
plotIPDF(Mag2Hem, Miller({0 0 1},{1 1 1},CS_Mag))
```

Phase Transition

We should care about symmetric equivalence

```
O_Mag = O_Hem * symmetrise(Mag2Hem, 'unique')
```

How many are crystallographically not equivalent?

```
unique(O_Mag)
```

```
ans = orientation (show methods, plot)
  size: 1 x 2
  crystal symmetry : Magnetite (m-3m)
  specimen symmetry: 1

  Bunge Euler angles in degree
     phi1      Phi     phi2     Inv.
  314.739  125.264      225        0
  254.739  125.264      315        0
```

Pole figures of misorientations

```
plotPDF(Mag2Hem, Miller({0 0 0 1},{1 1 -2 1},CS_Hem))
plotIPDF(Mag2Hem, Miller({0 0 1},{1 1 1},CS_Mag))
```

Phase Transition

We should care about symmetric equivalence

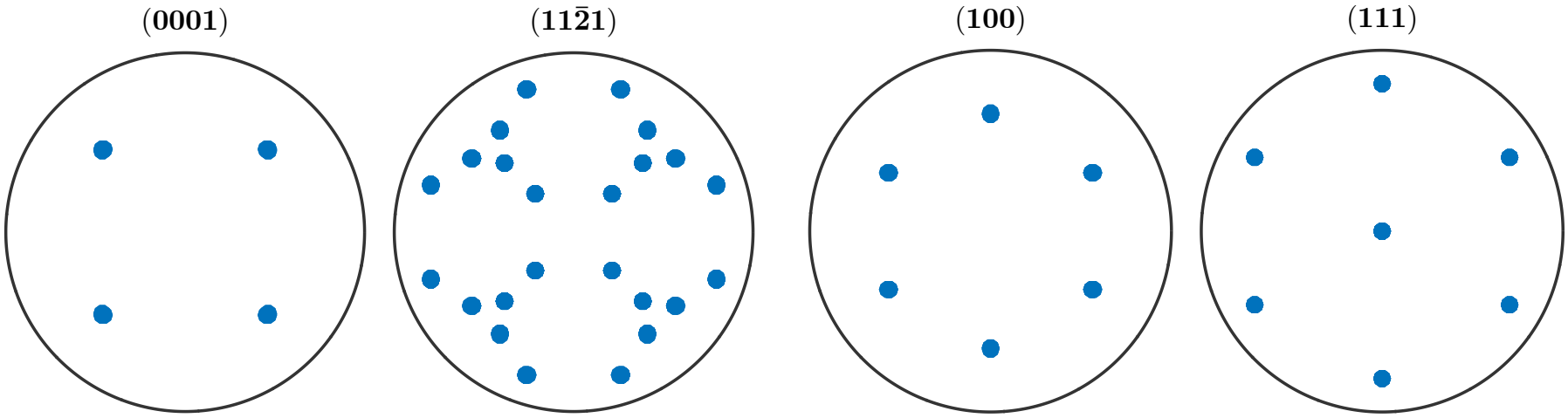
```
O_Mag = O_Hem * symmetrise(Mag2Hem, 'unique')
```

How many are crystallographically not equivalent?

```
unique(O_Mag)
```

Pole figures of misorientations

```
plotPDF(Mag2Hem, Miller({0 0 0 1},{1 1 -2 1},CS_Hem))
plotIPDF(Mag2Hem, Miller({0 0 1},{1 1 1},CS_Mag))
```



Relationship Between Lattice Planes

Problem: compute the minimal angle between two lattice planes of two different grains having different orientation and different phase.

```
m_Mag = Miller(1,0,0,cs_Magnetite);
m_Hem = Miller(1,1,-2,0,cs_Hematite);
```

```
m_Mag = Miller (show methods, plot)
 size: 1 x 1
 mineral: Magnetite (432)
  h 1
  k 0
  l 0

m_Hem = Miller (show methods, plot)
 size: 1 x 1
 mineral: Hematite (321, X||a*, Y||b, Z||c)
  h  1
  k  1
  i -2
  l  0
```

The orientation relation should be given by

Relationship Between Lattice Planes

Problem: compute the minimal angle between two lattice planes of two different grains having different orientation and different phase.

```
m_Mag = Miller(1,0,0,cs_Magnetite);
m_Hem = Miller(1,1,-2,0,cs_Hematite);
```

The orientation relation should be given by

```
Mag2Hem = orientation('map',...
   Miller(1,1,1,CS_Mag),Miller(0,0,0,1,CS_Hem),...
   Miller(-1,0,1,CS_Mag),Miller(1,0,-1,0,CS_Hem))
```

```
Mag2Hem = misorientation (show methods, plot)
  size: 1 x 1
  crystal symmetry : Magnetite (m-3m)
  crystal symmetry : Hematite (-3m1, X||a*, Y||b, Z||c)

  Bunge Euler angles in degree
  phi1 Phi phi2 Inv.
  120 54.7356 45 0
```

Relationship Between Lattice Planes

Problem: compute the minimal angle between two lattice planes of two different grains having different orientation and different phase.

```
m_Mag = Miller(1,0,0,cs_Magnetite);
m_Hem = Miller(1,1,-2,0,cs_Hematite);
```

The orientation relation should be given by

```
Mag2Hem = orientation('map',...
   Miller(1,1,1,CS_Mag),Miller(0,0,0,1,CS_Hem),...
   Miller(-1,0,1,CS_Mag),Miller(1,0,-1,0,CS_Hem))
```

The minimum angle

```
min(angle(Mag2Hem * m_Mag.symmetrise,m_Hem))/degree
```

```
  35.2644
```

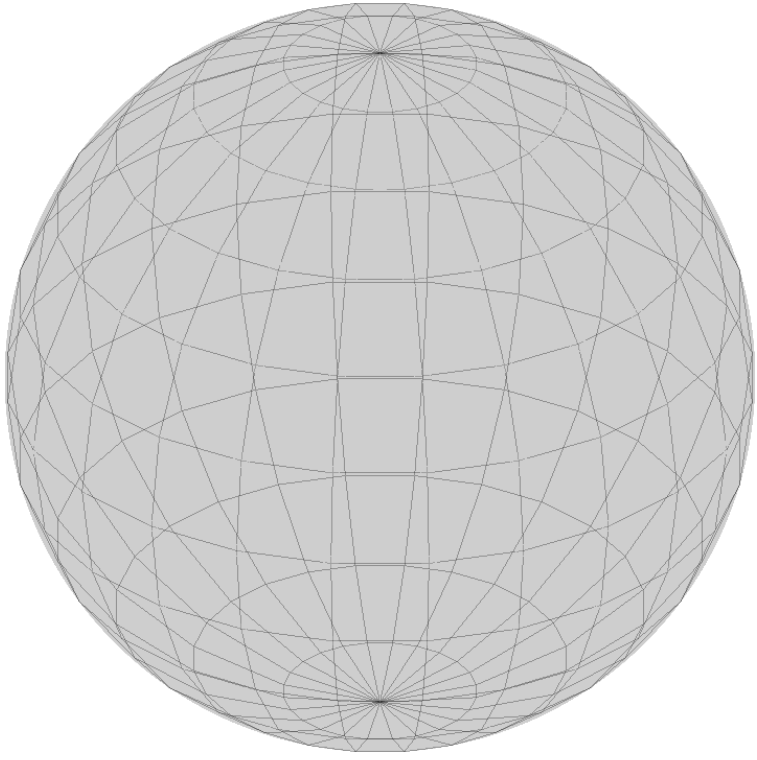
The Misorientation Space - Disjoint Symmetries

```
plot(orientationRegion)
```

```
cs3 = crystalSymmetry('3')
oR  = orientationRegion(cs3)
plot(oR,'color','r')
```

```
cs4 = crystalSymmetry('4')
oR  = orientationRegion(cs4)
plot(oR,'color','g')
```

```
oR = orientationRegion(cs3,cs4)
plot(oR,'color','r')
```



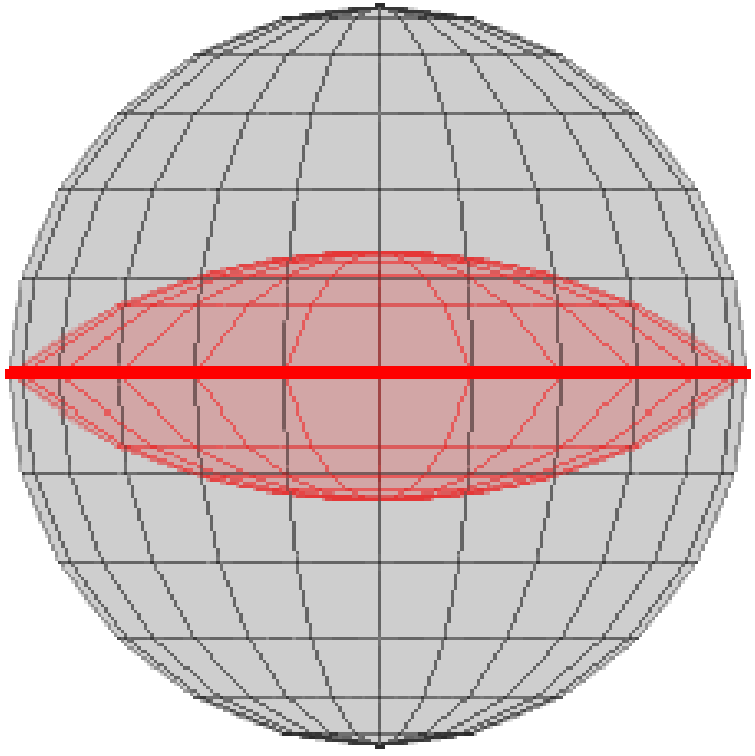
The Misorientation Space - Disjoint Symmetries

```
plot(orientationRegion)
```

```
cs3 = crystalSymmetry('3')
oR  = orientationRegion(cs3)
plot(oR,'color','r')
```

```
cs4 = crystalSymmetry('4')
oR  = orientationRegion(cs4)
plot(oR,'color','g')
```

```
oR = orientationRegion(cs3,cs4)
plot(oR,'color','r')
```



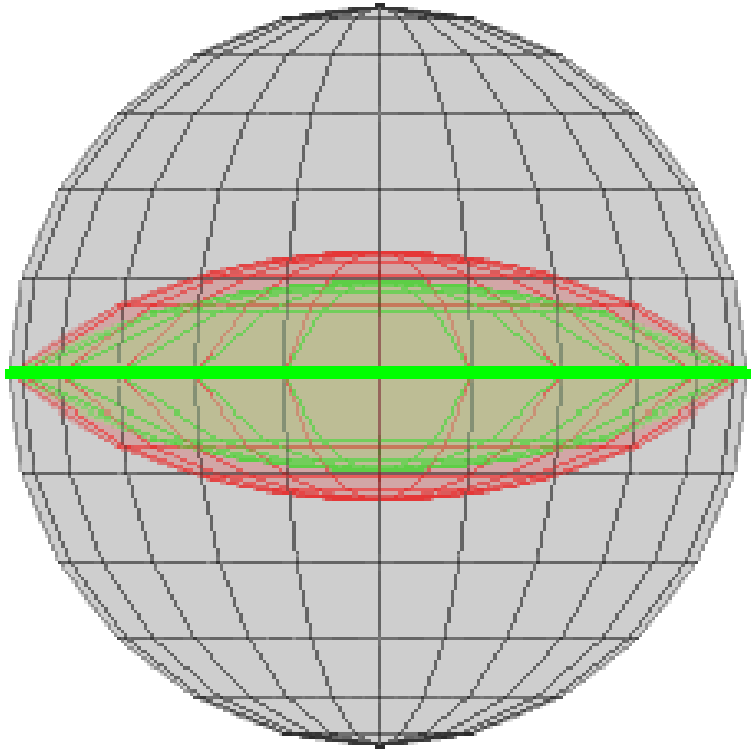
The Misorientation Space - Disjoint Symmetries

```
plot(orientationRegion)
```

```
cs3 = crystalSymmetry('3')
oR = orientationRegion(cs3)
plot(oR,'color','r')
```

```
cs4 = crystalSymmetry('4')
oR = orientationRegion(cs4)
plot(oR,'color','g')
```

```
oR = orientationRegion(cs3,cs4)
plot(oR,'color','r')
```



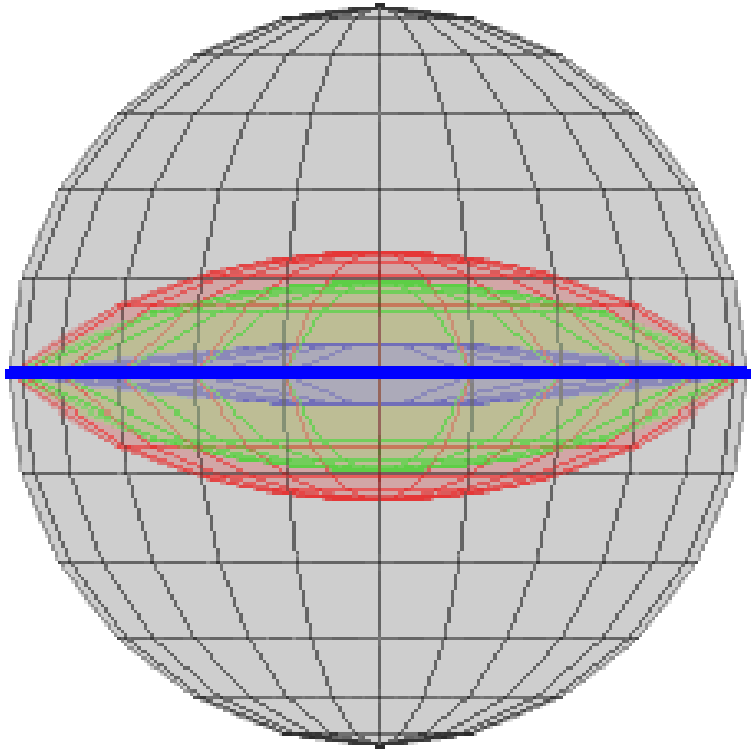
The Misorientation Space - Disjoint Symmetries

```
plot(orientationRegion)
```

```
cs3 = crystalSymmetry('3')
oR  = orientationRegion(cs3)
plot(oR,'color','r')
```

```
cs4 = crystalSymmetry('4')
oR  = orientationRegion(cs4)
plot(oR,'color','g')
```

```
oR = orientationRegion(cs3,cs4)
plot(oR,'color','r')
```



The Misorientation Space - Disjoint Symmetries

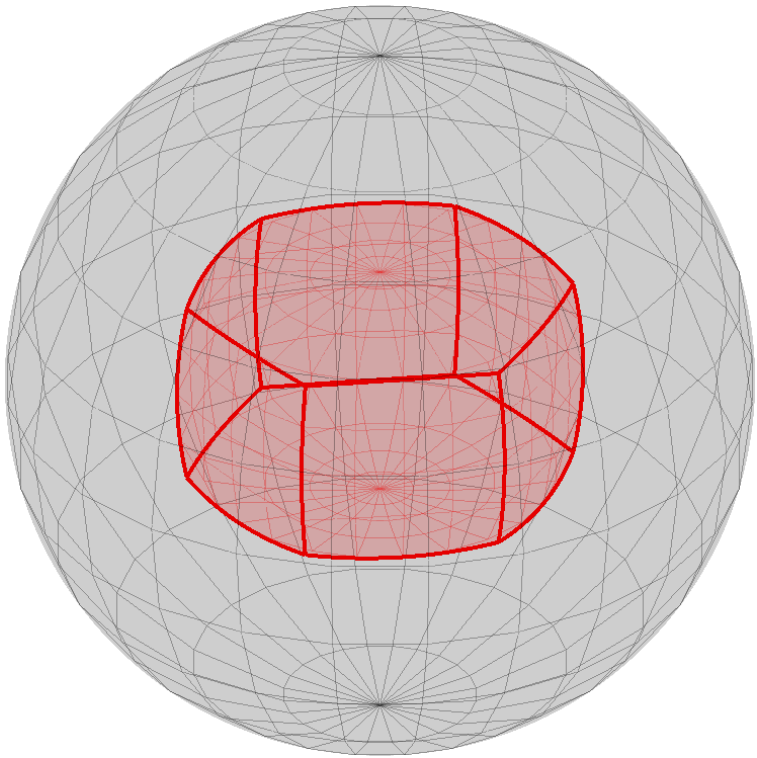
```
plot(orientationRegion)
```

```
cs3 = crystalSymmetry('3')
oR  = orientationRegion(cs3)
plot(oR,'color','r')
```

```
cs4 = crystalSymmetry('4')
oR  = orientationRegion(cs4)
plot(oR,'color','g')
```

```
oR = orientationRegion(cs3,cs4)
plot(oR,'color','r')
```

```
cs2 = crystalSymmetry('211')
oR = orientationRegion(cs3,cs2)
plot(oR,'color','r')
```



The Misorientation Space - Disjoint Symmetries

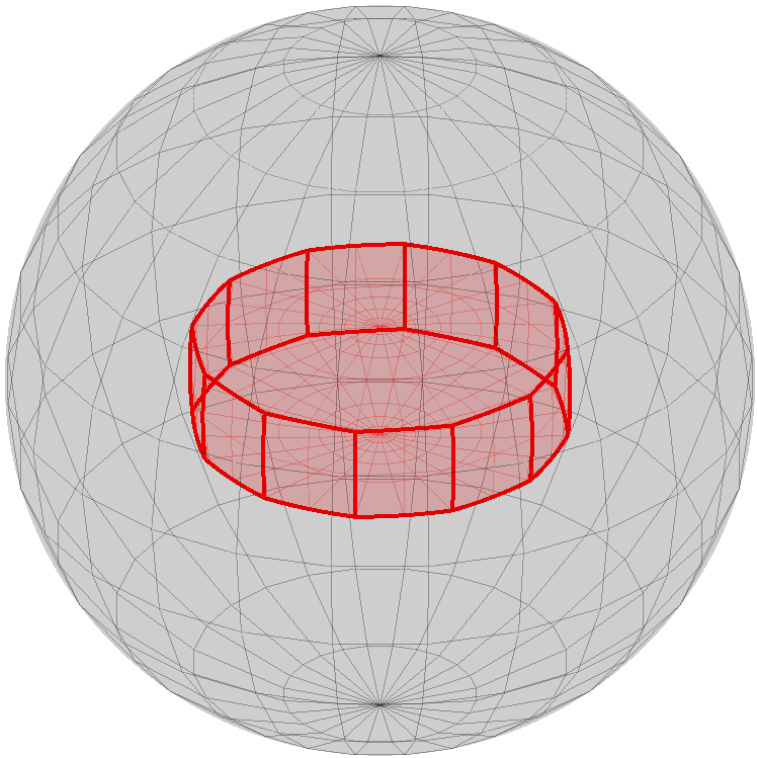
```
plot(orientationRegion)
```

```
cs3 = crystalSymmetry('3')
oR  = orientationRegion(cs3)
plot(oR,'color','r')
```

```
cs4 = crystalSymmetry('4')
oR  = orientationRegion(cs4)
plot(oR,'color','g')
```

```
oR = orientationRegion(cs3,cs4)
plot(oR,'color','r')
```

```
cs2 = crystalSymmetry('222')
oR = orientationRegion(cs3,cs2)
plot(oR,'color','r')
```



The Misorientation Space - Common Symmetries

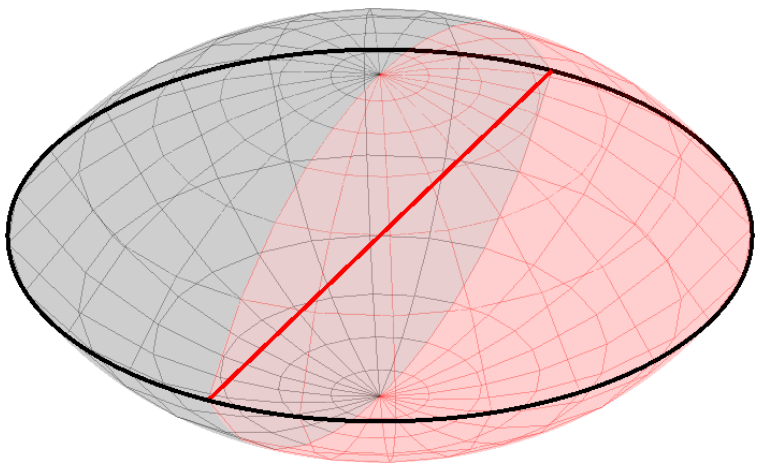
```
cs = crystalSymmetry('211')
oR = orientationRegion(cs,cs)
plot(oR,'color','r')
```

```
cs = crystalSymmetry('3')
oR = orientationRegion(cs,cs)
```

```
cs = crystalSymmetry('222')
oR = orientationRegion(cs,cs)
```

```
cs = crystalSymmetry('222')
oR = orientationRegion(cs,cs,...
    'antipodal')
```

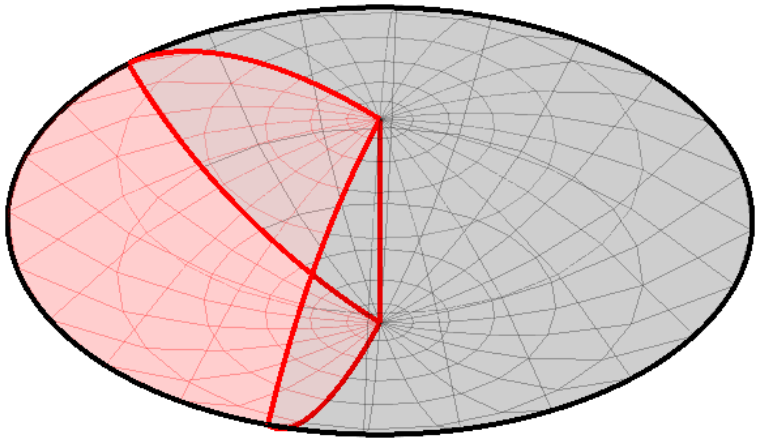
```
cs = crystalSymmetry('432')
oR = orientationRegion(cs)
```



The Misorientation Space - Common Symmetries

```
cs = crystalSymmetry('211')
oR = orientationRegion(cs,cs)
plot(oR,'color','r')
```

```
cs = crystalSymmetry('3')
oR = orientationRegion(cs,cs)
```



```
cs = crystalSymmetry('222')
oR = orientationRegion(cs,cs)
```

```
cs = crystalSymmetry('222')
oR = orientationRegion(cs,cs,...
    'antipodal')
```

```
cs = crystalSymmetry('432')
oR = orientationRegion(cs)
```

The Misorientation Space - Common Symmetries

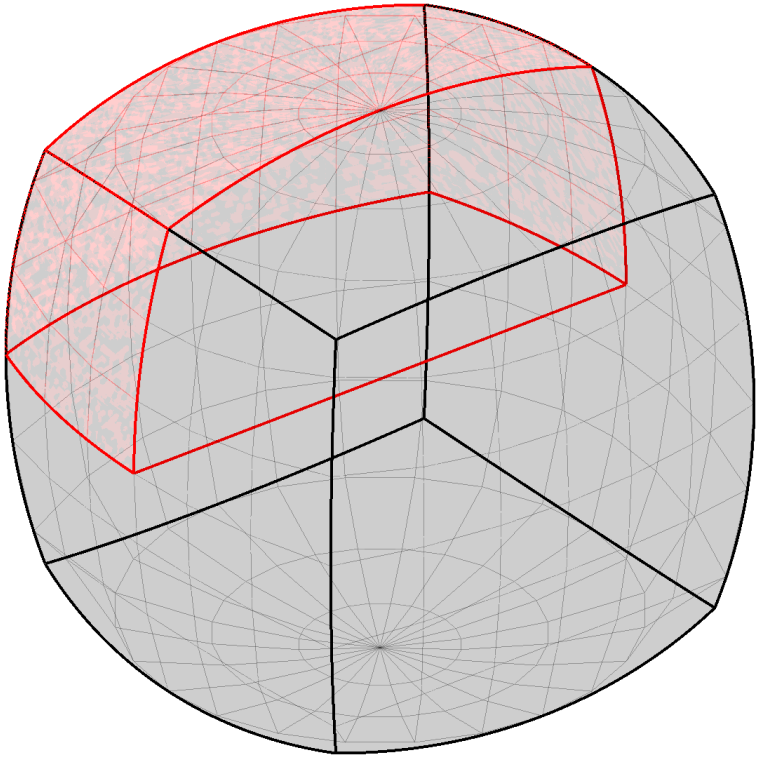
```
cs = crystalSymmetry('211')
oR = orientationRegion(cs,cs)
plot(oR,'color','r')
```

```
cs = crystalSymmetry('3')
oR = orientationRegion(cs,cs)
```

```
cs = crystalSymmetry('222')
oR = orientationRegion(cs,cs)
```

```
cs = crystalSymmetry('222')
oR = orientationRegion(cs,cs,...
    'antipodal')
```

```
cs = crystalSymmetry('432')
oR = orientationRegion(cs)
```



The Misorientation Space - Common Symmetries

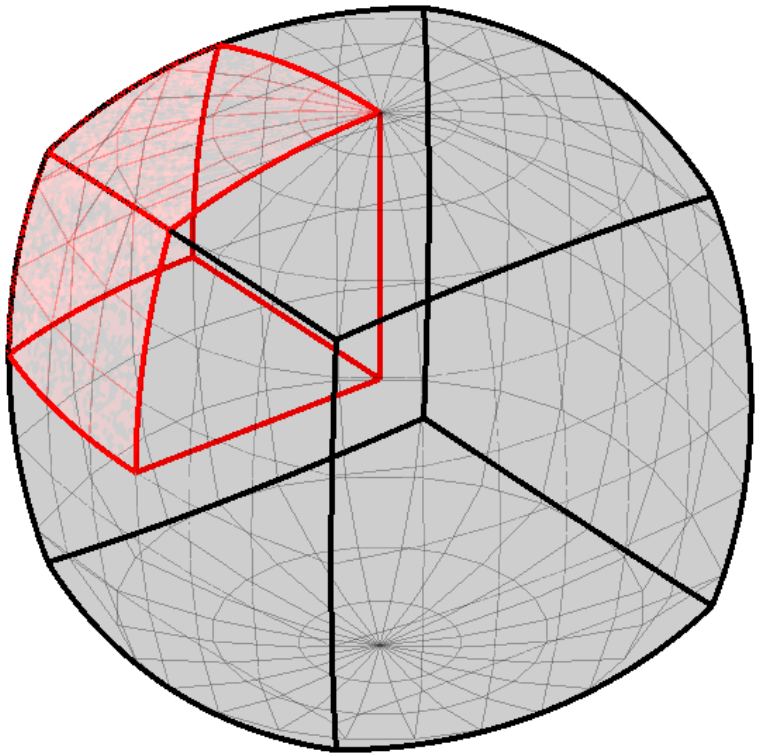
```
cs = crystalSymmetry('211')
oR = orientationRegion(cs,cs)
plot(oR,'color','r')
```

```
cs = crystalSymmetry('3')
oR = orientationRegion(cs,cs)
```

```
cs = crystalSymmetry('222')
oR = orientationRegion(cs,cs)
```

```
cs = crystalSymmetry('222')
oR = orientationRegion(cs,cs,...
    'antipodal')
```

```
cs = crystalSymmetry('432')
oR = orientationRegion(cs)
```



The Misorientation Space - Common Symmetries

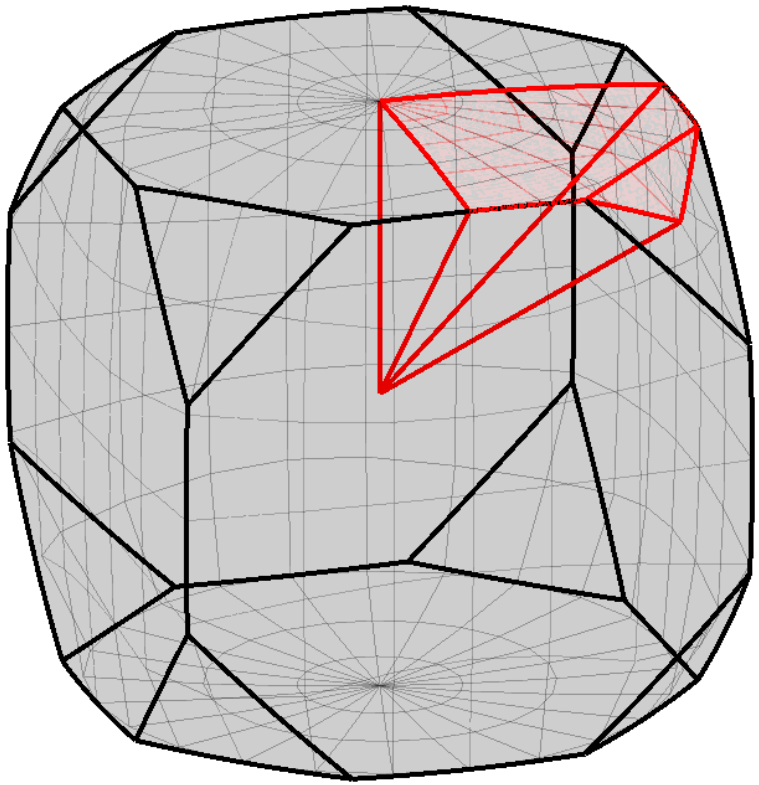
```
cs = crystalSymmetry('211')
oR = orientationRegion(cs,cs)
plot(oR,'color','r')
```

```
cs = crystalSymmetry('3')
oR = orientationRegion(cs,cs)
```

```
cs = crystalSymmetry('222')
oR = orientationRegion(cs,cs)
```

```
cs = crystalSymmetry('222')
oR = orientationRegion(cs,cs,...
      'antipodal')
```

```
cs = crystalSymmetry('432')
oR = orientationRegion(cs)
```



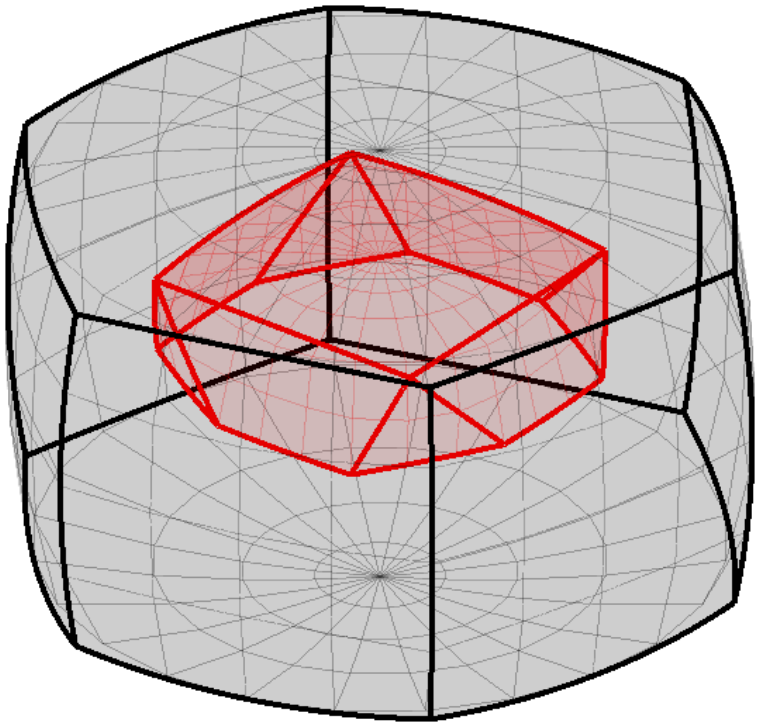
The Misorientation Space - Phase Transitions

```
cs1 = crystalSymmetry('23')
cs2 = crystalSymmetry('32')
oR = orientationRegion(cs1,cs2)
```

```
cs1 = crystalSymmetry('432')
cs2 = crystalSymmetry('321')
oR = orientationRegion(cs1,cs2)
```

```
Mag2Hem = orientation('map',...
    Miller(1,1,1,cs1), ...
    Miller(0,0,0,1, cs2) , ...
    Miller(-1,0,1, cs1), ...
    Miller(1,0,-1,0,cs2))
plot(Mag2Hem)
```

```
plotSection(Mag2Hem)
```



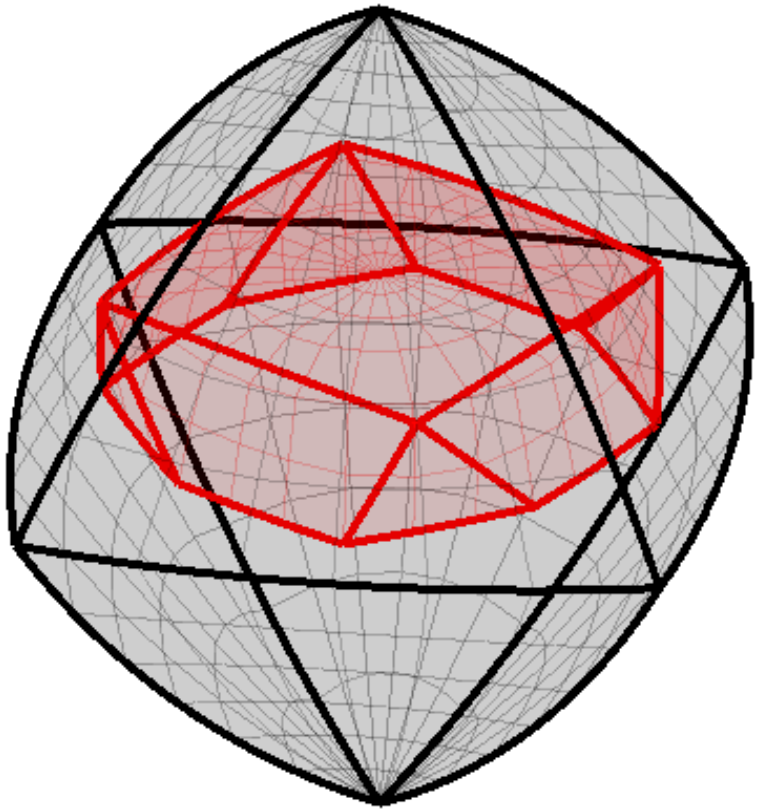
The Misorientation Space - Phase Transitions

```
cs1 = crystalSymmetry('23')
cs2 = crystalSymmetry('32')
oR = orientationRegion(cs1,cs2)
```

```
cs1 = crystalSymmetry('432')
cs2 = crystalSymmetry('321')
oR = orientationRegion(cs1,cs2)
```

```
Mag2Hem = orientation('map',...
    Miller(1,1,1,cs1), ...
    Miller(0,0,0,1, cs2) , ...
    Miller(-1,0,1, cs1), ...
    Miller(1,0,-1,0,cs2))
plot(Mag2Hem)
```

```
plotSection(Mag2Hem)
```



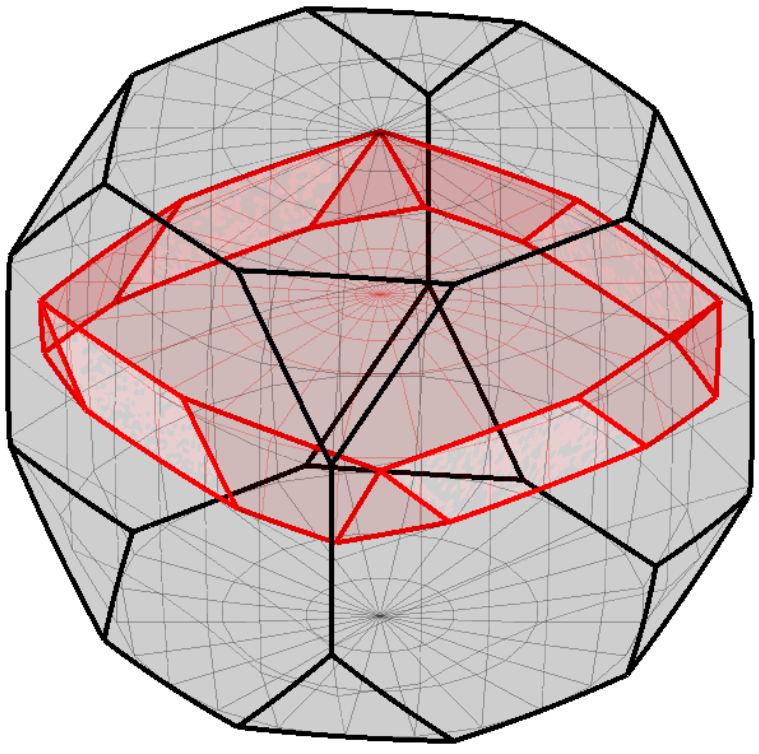
The Misorientation Space - Phase Transitions

```
cs1 = crystalSymmetry('23')
cs2 = crystalSymmetry('32')
oR = orientationRegion(cs1,cs2)
```

```
cs1 = crystalSymmetry('432')
cs2 = crystalSymmetry('321')
oR = orientationRegion(cs1,cs2)
```

```
Mag2Hem = orientation('map',...
    Miller(1,1,1,cs1), ...
    Miller(0,0,0,1, cs2) , ...
    Miller(-1,0,1, cs1), ...
    Miller(1,0,-1,0,cs2))
plot(Mag2Hem)
```

```
plotSection(Mag2Hem)
```



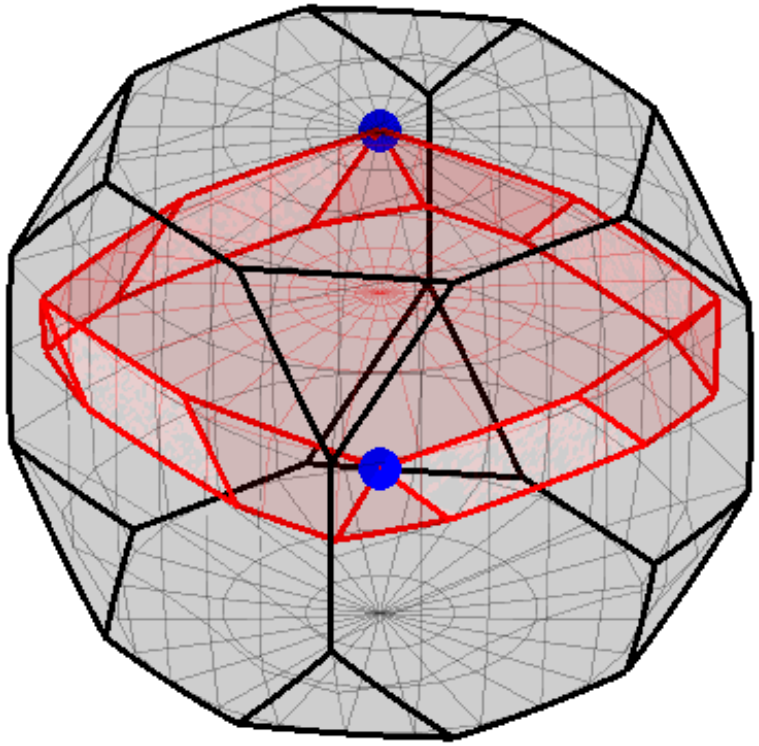
The Misorientation Space - Phase Transitions

```
cs1 = crystalSymmetry('23')
cs2 = crystalSymmetry('32')
oR = orientationRegion(cs1,cs2)
```

```
cs1 = crystalSymmetry('432')
cs2 = crystalSymmetry('321')
oR = orientationRegion(cs1,cs2)
```

```
Mag2Hem = orientation('map',...
    Miller(1,1,1,cs1), ...
    Miller(0,0,0,1, cs2) , ...
    Miller(-1,0,1, cs1), ...
    Miller(1,0,-1,0,cs2))
plot(Mag2Hem)
```

```
plotSection(Mag2Hem)
```



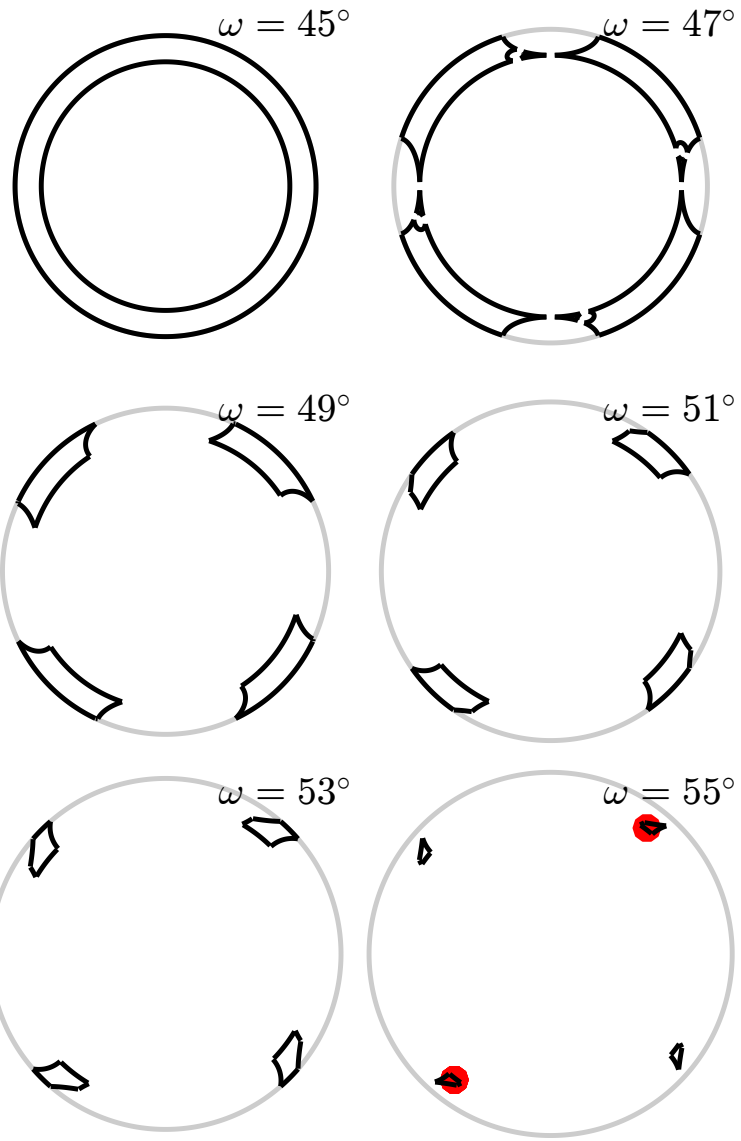
The Misorientation Space - Phase Transitions

```
cs1 = crystalSymmetry('23')
cs2 = crystalSymmetry('32')
oR = orientationRegion(cs1,cs2)
```

```
cs1 = crystalSymmetry('432')
cs2 = crystalSymmetry('321')
oR = orientationRegion(cs1,cs2)
```

```
Mag2Hem = orientation('map',...
    Miller(1,1,1,cs1), ...
    Miller(0,0,0,1, cs2) , ...
    Miller(-1,0,1, cs1), ...
    Miller(1,0,-1,0,cs2))
plot(Mag2Hem)
```

```
plotSection(Mag2Hem)
```



Austenite to Martensite transition

```
ebsd=loadEBSD('martensite.ctf')
```

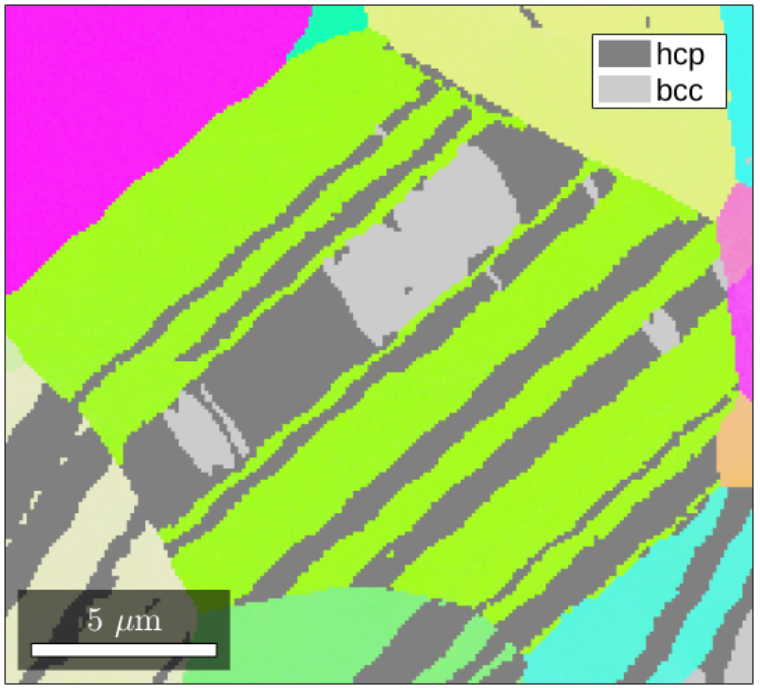
```
ebsd = EBSD (show methods, plot)

 Phase   Orientations   Mineral  Symmetry
     1     25389 (69)       fcc      m-3m
     2     1884 (5.1)       bcc      m-3m
     3      9693 (26)       hcp     6/mmm

 Properties: bands, bc, bs, error, mad
 Scan unit : um
```

```
grains=calcGrains(ebsd);
plot(grains.boundary)
gB=grains.boundary('fcc','hcp')
```

```
plotSection(gB.misorientation)
```

```
mdf=calcMDF(gB.misorientation)
plotSection(mdf)
```

```
plot(gB.misorientation,...
    mdf.eval(gB.misorientation))
```



Austenite to Martensite transition

```
ebsd=loadEBSD('martensite.ctf')
```

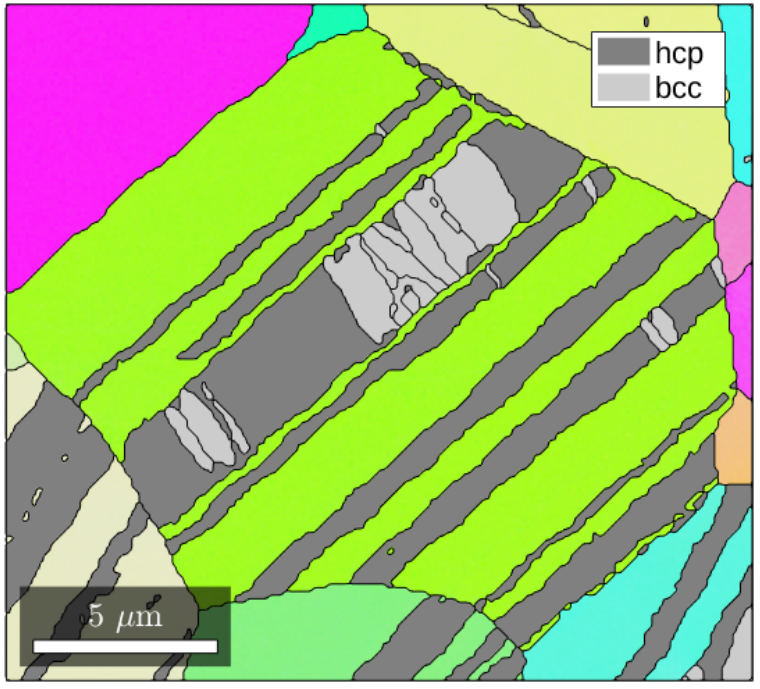
```
grains=calcGrains(ebsd);
plot(grains.boundary)
gB=grains.boundary('fcc','hcp')
```

```
gB = grainBoundary (show methods, plot)

 Segments   mineral 1   mineral 2
     3829         fcc         hcp
```

```
plotSection(gB.misorientation)
```

```
mdf=calcMDF(gB.misorientation)
plotSection(mdf)
```

```
plot(gB.misorientation,...
    mdf.eval(gB.misorientation))
```



Austenite to Martensite transition

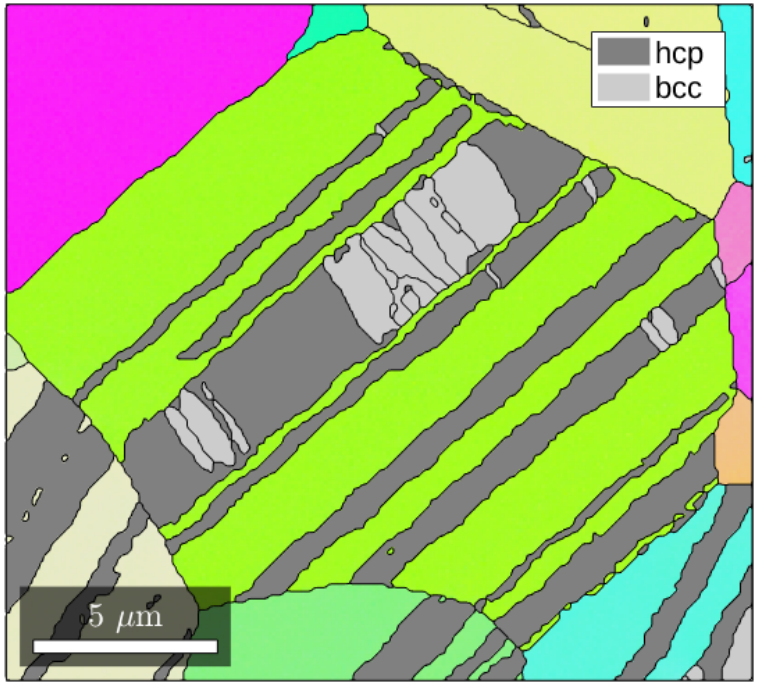
```
ebsd=loadEBSD('martensite.ctf')
```

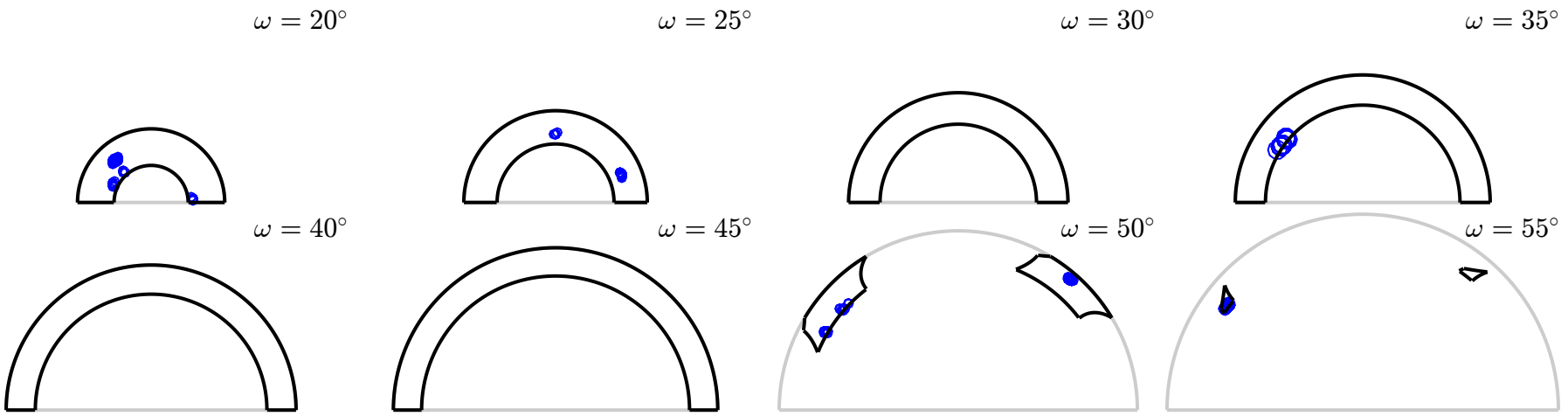
```
grains=calcGrains(ebsd);
plot(grains.boundary)
gB=grains.boundary('fcc','hcp')
```

```
plotSection(gB.misorientation)
```

```
mdf=calcMDF(gB.misorientation)
plotSection(mdf)
```

```
plot(gB.misorientation,...
    mdf.eval(gB.misorientation))
```





Austenite to Martensite transition

```
ebsd=loadEBSD('martensite.ctf')
```

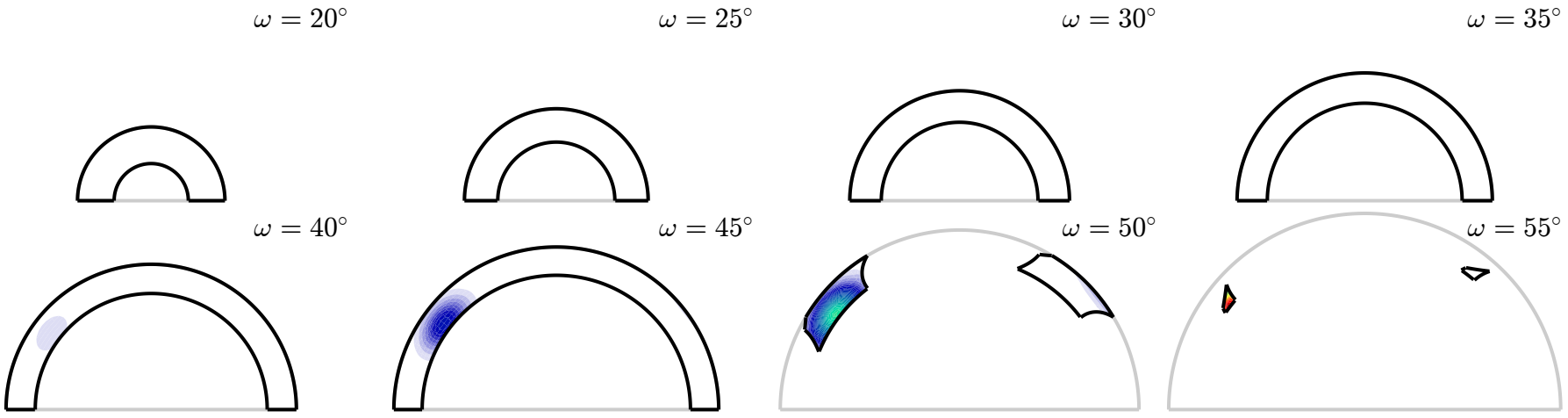
```
grains=calcGrains(ebsd);
plot(grains.boundary)
gB=grains.boundary('fcc','hcp')
```

```
plotSection(gB.misorientation)
```

```
mdf=calcMDF(gB.misorientation)
plotSection(mdf)
```

```
plot(gB.misorientation,...
    mdf.eval(gB.misorientation))
```

```
mdf = MDF
  crystal sym: fcc (m-3m)
  crystal sym: hcp (6/mmm)

  Harmonic portion:
    degree: 38
    weight: 1
```



Austenite to Martensite transition

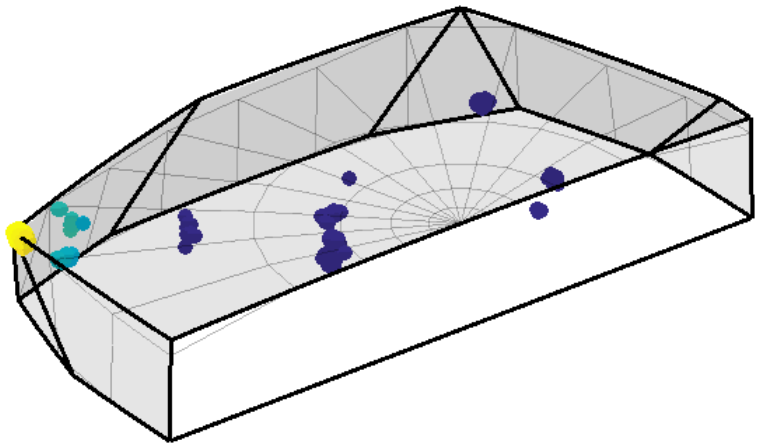
```
ebsd=loadEBSD('martensite.ctf')
```

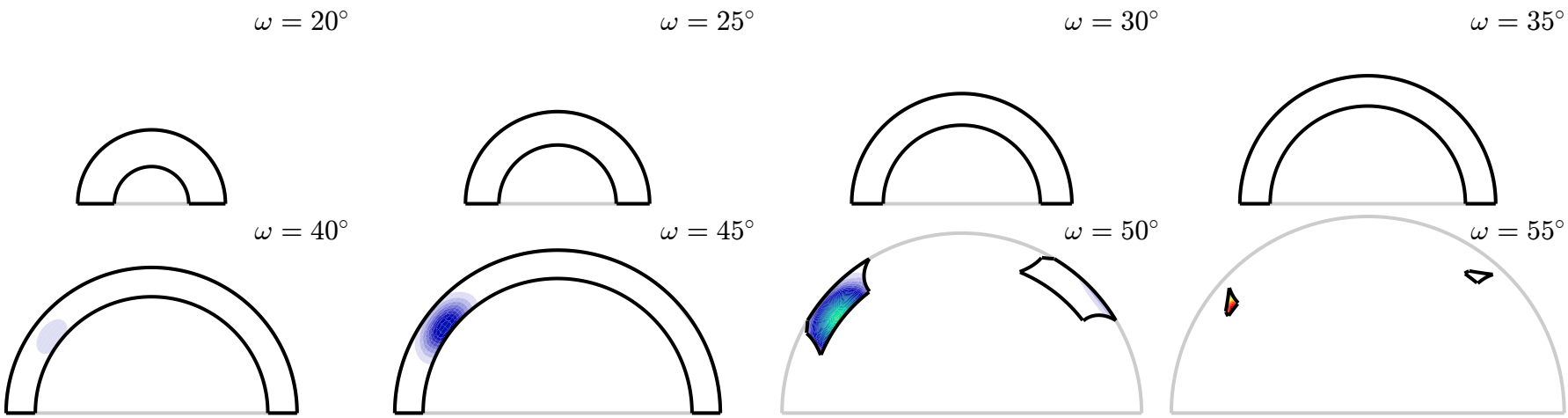
```
grains=calcGrains(ebsd);
plot(grains.boundary)
gB=grains.boundary('fcc','hcp')
```

```
plotSection(gB.misorientation)
```

```
mdf=calcMDF(gB.misorientation)
plotSection(mdf)
```

```
plot(gB.misorientation,...
    mdf.eval(gB.misorientation))
```





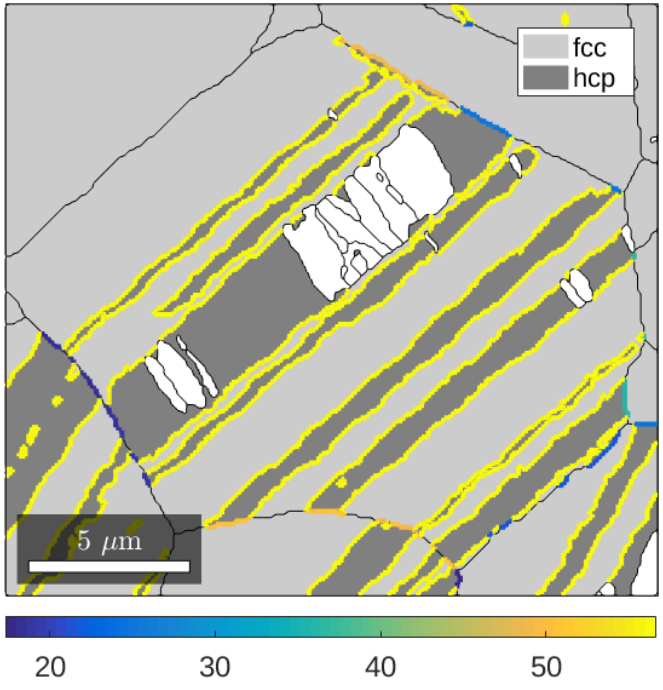
Austenite to Martensite transition

```
plot(gB,gB.misorientation.angle)
```

```
oR = fundamentalRegion(csFCC,csHCP)
mO = orientation(oR.V,csFCC,csHCP)
tB = gB(gB.isTwinning(mO,1*degree))
plot(tB,'lincecolor','r')
```

```
[mGrains,parent] = merge(grains,tB)
plot(mGrains.boundary)
```

```
wasFCC = parent(grains.phase==1);
mGrains.phase(wasFCC) = 1
```



Austenite to Martensite transition

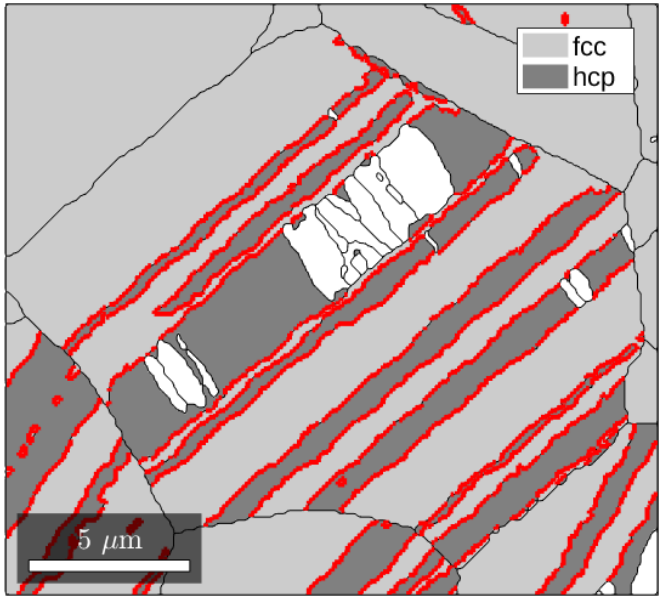
```
plot(gB,gB.misorientation.angle)
```

```
oR = fundamentalRegion(csFCC,csHCP)
mO = orientation(oR.V,csFCC,csHCP)
tB = gB(gB.isTwinning(mO,1*degree))
plot(tB,'lincecolor','r')
```

```
tB = grainBoundary (show methods, plot)

 Segments   mineral 1   mineral 2
     3283         fcc         hcp
```

```
[mGrains,parent] = merge(grains,tB)
plot(mGrains.boundary)
```

```
wasFCC = parent(grains.phase==1);
mGrains.phase(wasFCC) = 1
```



Austenite to Martensite transition

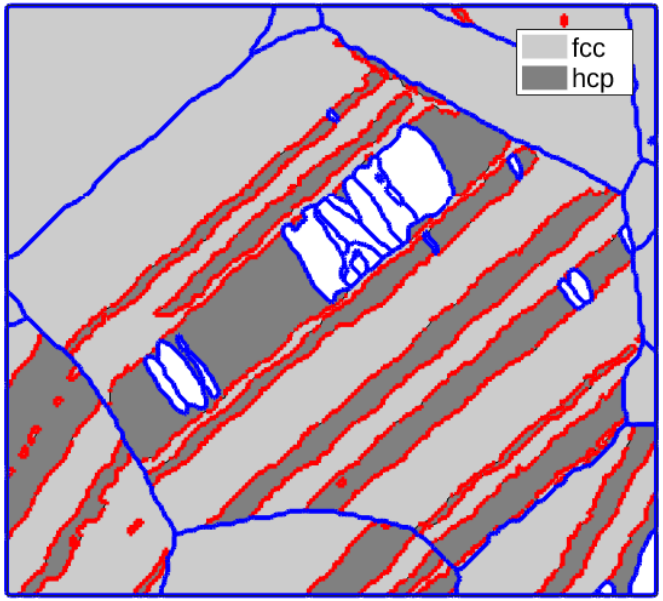
```
plot(gB,gB.misorientation.angle)
```

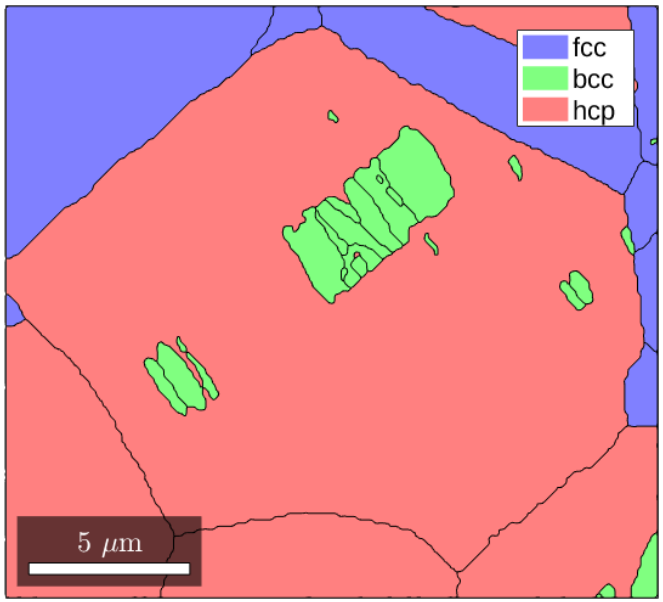
```
oR = fundamentalRegion(csFCC,csHCP)
mO = orientation(oR.V,csFCC,csHCP)
tB = gB(gB.isTwinning(mO,1*degree))
plot(tB,'lincecolor','r')
```

```
[mGrains,parent] = merge(grains,tB)
plot(mGrains.boundary)
```

```
mGrains = grain2d (show methods, plot)

 Phase   Grains   Pixels   Mineral   Symmetry
     1        8     7156       fcc       m-3m
     2       26     1884       bcc       m-3m
     3        6    27926       hcp       6/mmm
```

```
wasFCC = parent(grains.phase==1);
mGrains.phase(wasFCC) = 1
```





Austenite to Martensite transition

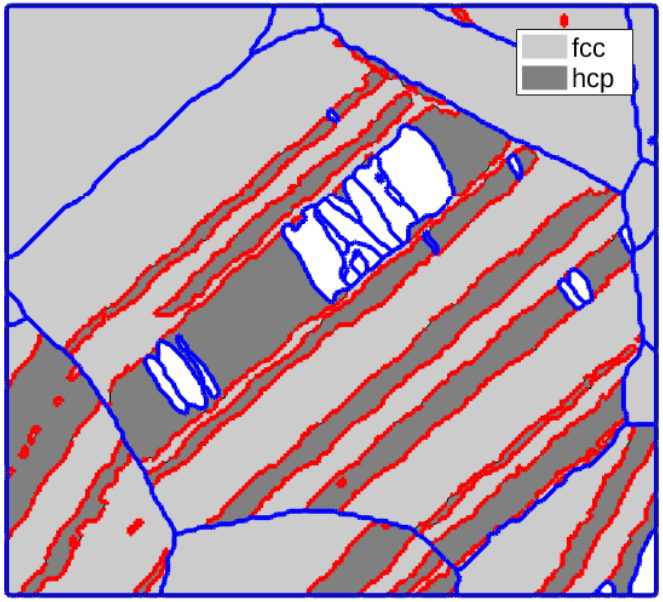
```
plot(gB,gB.misorientation.angle)
```

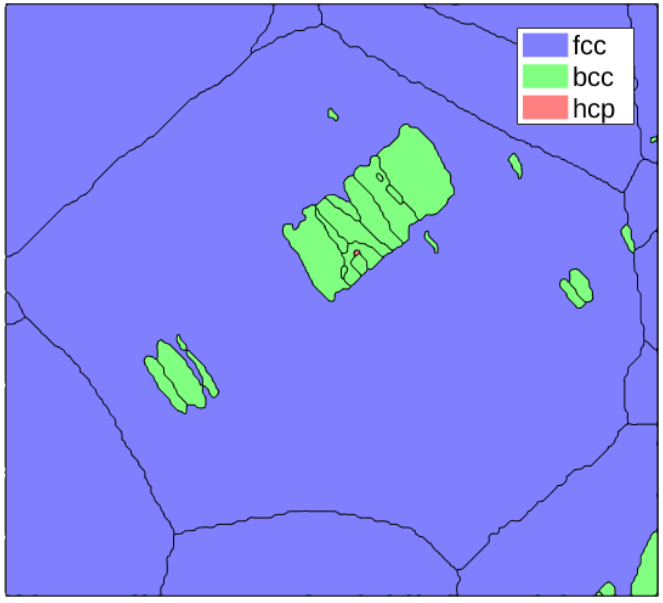
```
oR = fundamentalRegion(csFCC,csHCP)
mO = orientation(oR.V,csFCC,csHCP)
tB = gB(gB.isTwinning(mO,1*degree))
plot(tB,'lincecolor','r')
```

```
[mGrains,parent] = merge(grains,tB)
plot(mGrains.boundary)
```

```
wasFCC = parent(grains.phase==1);
mGrains.phase(wasFCC) = 1
```

```
mGrains = grain2d (show methods, plot)

 Phase   Grains   Pixels   Mineral   Symmetry
     1       13    35079       fcc       m-3m
     2       26     1884       bcc       m-3m
     3        1        3       hcp       6/mmm
```





Misorientation angel and axis

the smallest rotational angle of all symmetrically equivalent misorientations to **MO** is called misorientation angle

```
angle(O1,O2), angle(MO), angle(inv(MO))
```

for boundary misorientations we can do a simple statistics by

```
gB = grain.boundary
hist(gB('fcc','fcc').misorientation.angle ./ degree)
```

nicer plots

```
plotAngleDistribution(gB('fcc','fcc').misorientation)
```

Misorientation angel and axis

the smallest rotational angle of all symmetrically equivalent misorientations to **MO** is called misorientation angle

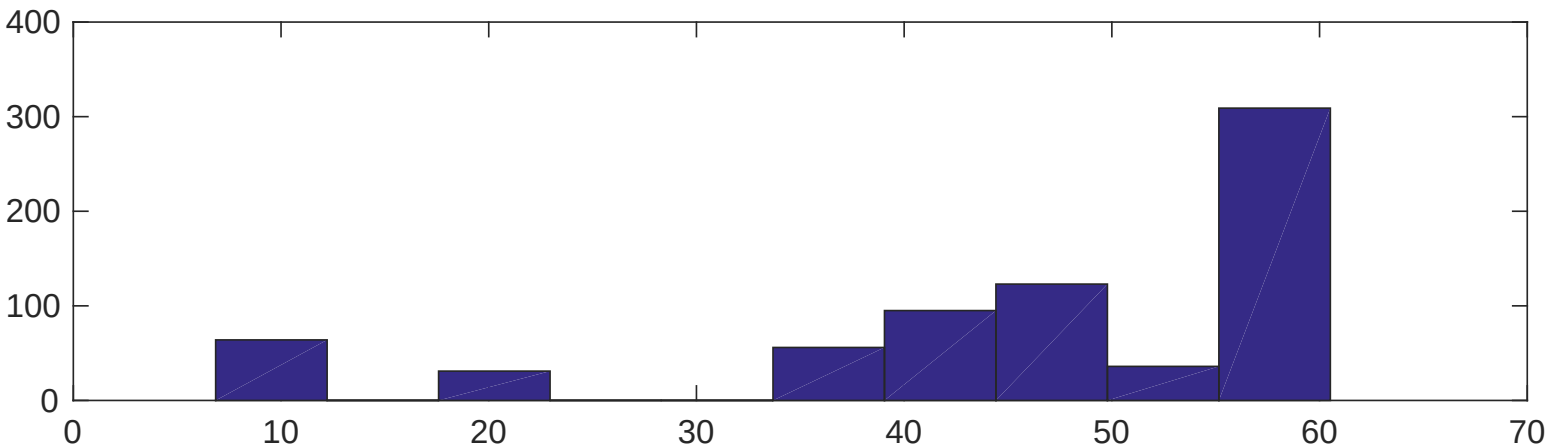
```
angle(O1,O2),  angle(MO),  angle(inv(MO))
```

for boundary misorientations we can do a simple statistics by

```
gB = grain.boundary
hist(gB('fcc','fcc').misorientation.angle ./ degree)
```

nicer plots

```
plotAngleDistribution(gB('fcc','fcc').misorientation)
```



Misorientation angel and axis

the smallest rotational angle of all symmetrically equivalent misorientations to **MO** is called misorientation angle

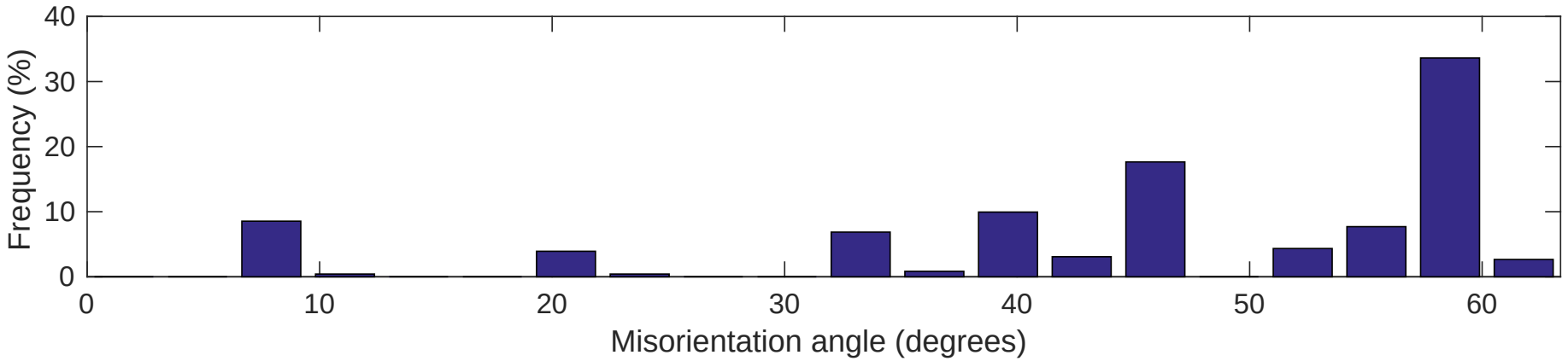
```
angle(O1,O2), angle(MO), angle(inv(MO))
```

for boundary misorientations we can do a simple statistics by

```
gB = grain.boundary
hist(gB('fcc','fcc').misorientation.angle ./ degree)
```

nicer plots

```
plotAngleDistribution(gB('fcc','fcc').misorientation)
```



Angle Distributions

The untextured, uncorrelated angle distribution

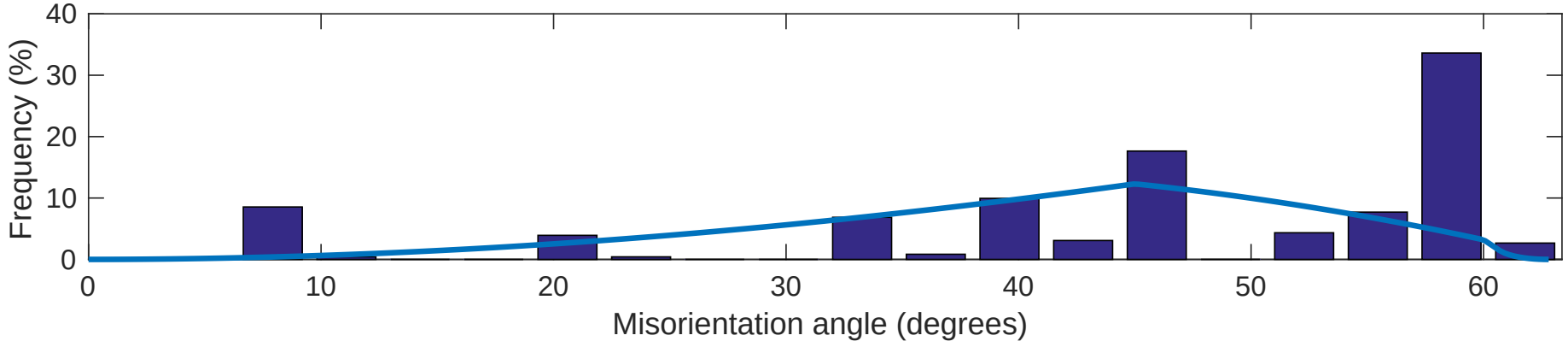
```
plotAngleDistribution(csFCC,csFCC)
```

The textured, uncorrelated angle distribution

```
odf = calcODF(ebsd('fcc').orientations);
mdf = calcMDF(odf,odf)
plotAngleDistribution(mdf)
```

The textured, correlated angle distribution

```
mdf = calcMDF(gB('fcc','fcc').misorientation)
plotAngleDistribution(mdf)
```



Angle Distributions

The untextured, uncorrelated angle distribution

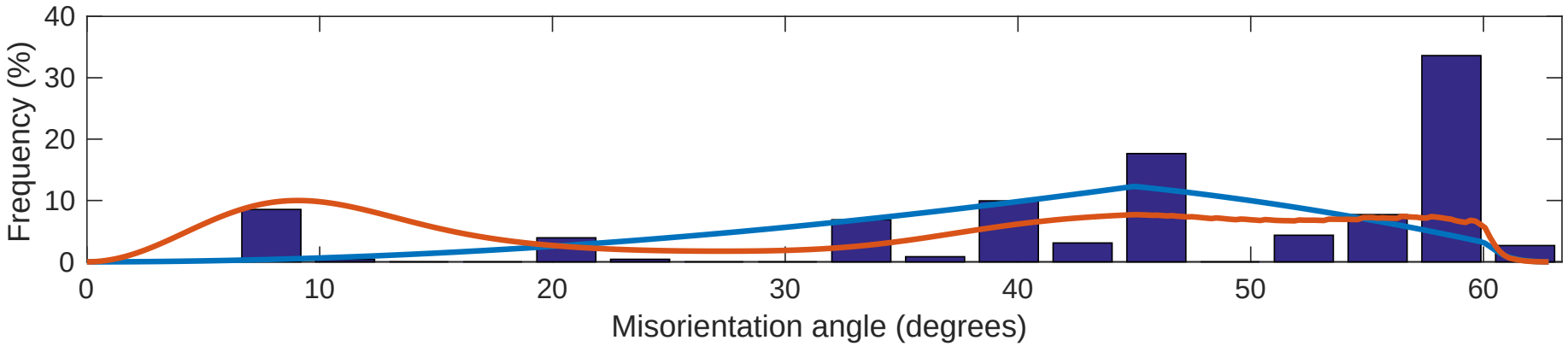
```
plotAngleDistribution(csFCC,csFCC)
```

The textured, uncorrelated angle distribution

```
odf = calcODF(ebsd('fcc').orientations);
mdf = calcMDF(odf,odf)
plotAngleDistribution(mdf)
```

The textured, correlated angle distribution

```
mdf = calcMDF(gB('fcc','fcc').misorientation)
plotAngleDistribution(mdf)
```



Angle Distributions

The untextured, uncorrelated angle distribution

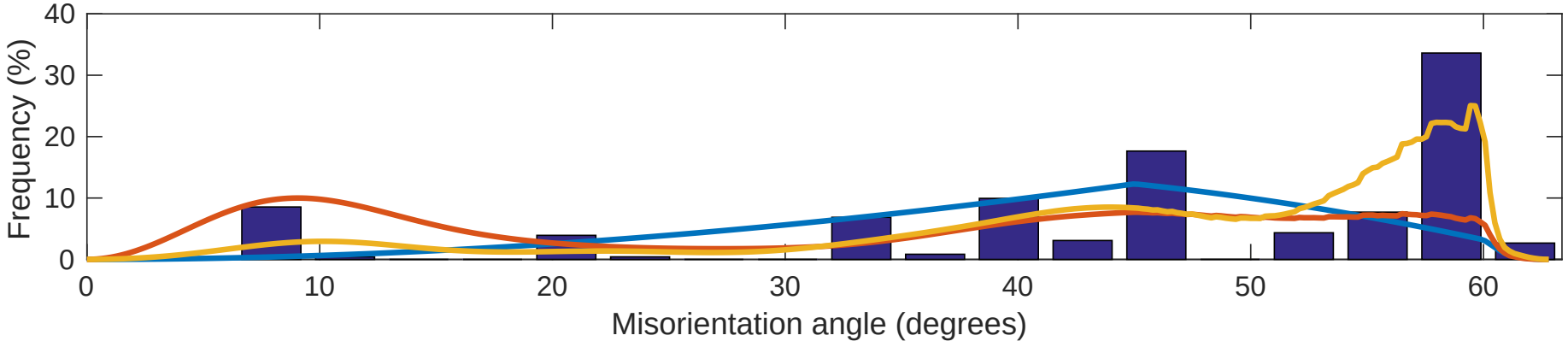
```
plotAngleDistribution(csFCC,csFCC)
```

The textured, uncorrelated angle distribution

```
odf = calcODF(ebsd('fcc').orientations);
mdf = calcMDF(odf,odf)
plotAngleDistribution(mdf)
```

The textured, correlated angle distribution

```
mdf = calcMDF(gB('fcc','fcc').misorientation)
plotAngleDistribution(mdf)
```



Misorientation axis in crystal coordinates

the misorientation realizing the minimum angle is

```
MO.project2FundamentalRegion
```

```
MO = misorientation (show methods, plot)
  size: 1 x 1
  crystal symmetry : Magnetite (m-3m)
  crystal symmetry : Hematite (-3m1)

  Bunge Euler angles in degree
     phi1      Phi     phi2     Inv.
  60.1624 54.6037 314.719        0
```

its rotational axis is the misorientation axis

```
axis(MO)
```

```
axis(inv(MO))
```

```
plotAxisDistribution(...
  gB.misorientation)
plotAxisDistribution(...
  inv(gB.misorientation))
```

Misorientation axis in crystal coordinates

the misorientation realizing the minimum angle is

```
MO.project2FundamentalRegion
```

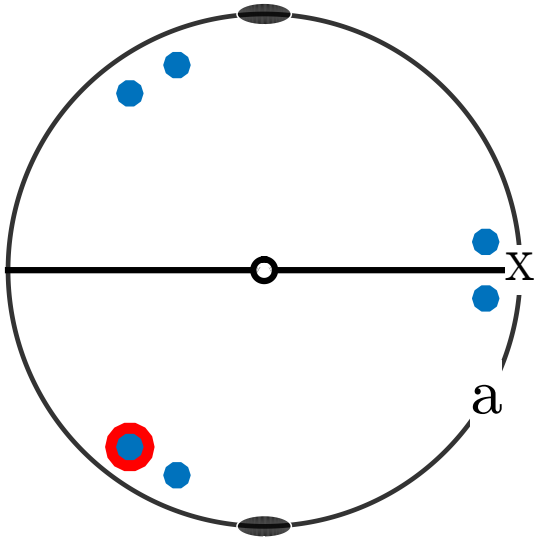
its rotational axis is the misorientation axis

```
axis(MO)
```

```
ans = Miller (show methods, plot)
 size: 1 x 1
 mineral: Hematite (-3m1)
  h   0.6191
  k   3.8882
  i  -4.5073
  l   3.3513
```

```
axis(inv(MO))
```

```
plotAxisDistribution(...
  gB.misorientation)
plotAxisDistribution(...
  inv(gB.misorientation))
```



Misorientation axis in crystal coordinates

the misorientation realizing the minimum angle is

```
MO.project2FundamentalRegion
```

its rotational axis is the misorientation axis

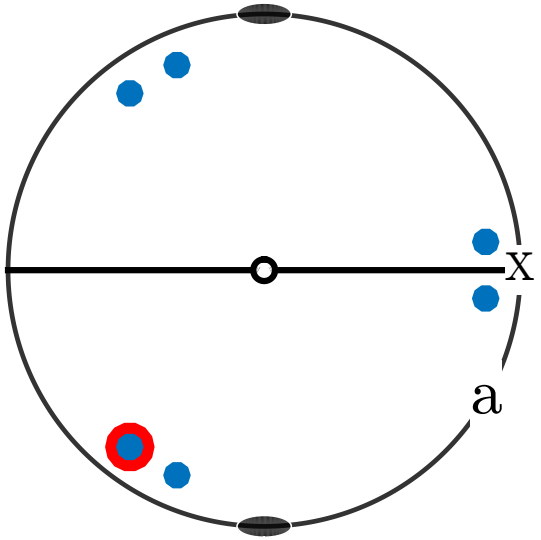
```
axis(MO)
```

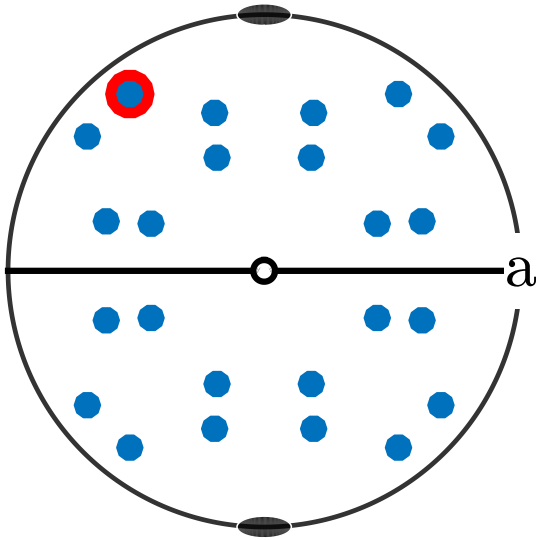
```
axis(inv(MO))
```

```
ans = Miller (show methods, plot)
 size: 1 x 1
 mineral: Magnetite (m-3m)
  h   4.9324
  k  -6.4797
  l   2.0431
```

```
plotAxisDistribution(...
  gB.misorientation)
plotAxisDistribution(...
  inv(gB.misorientation))
```

```
plotAxisDistribution(...
```





Misorientation axis in crystal coordinates

the misorientation realizing the minimum angle is

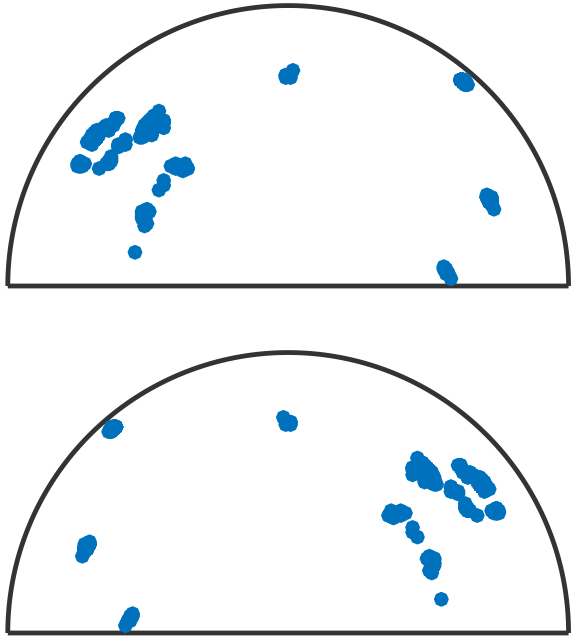
```
MO.project2FundamentalRegion
```

its rotational axis is the misorientation axis

```
axis(MO)
```

```
axis(inv(MO))
```

```
plotAxisDistribution(...
  gB.misorientation)
plotAxisDistribution(...
  inv(gB.misorientation))
```



```
plotAxisDistribution(...
  gB.misorientation,'contourf')
plotAxisDistribution(...
  inv(gB.misorientation),'contourf')
```

Misorientation axis in crystal coordinates

the misorientation realizing the minimum angle is

```
MO.project2FundamentalRegion
```

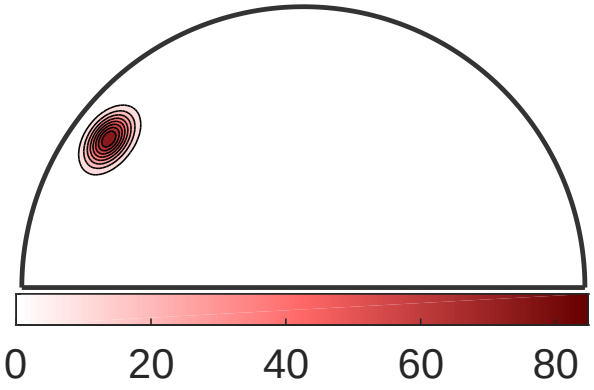
its rotational axis is the misorientation axis

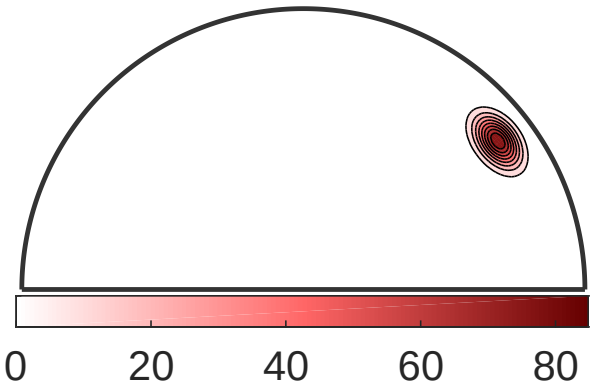
```
axis(MO)
```

```
axis(inv(MO))
```

```
plotAxisDistribution(...
  gB.misorientation)
plotAxisDistribution(...
  inv(gB.misorientation))
```

```
plotAxisDistribution(...
  gB.misorientation,'contourf')
plotAxisDistribution(...
 inv(gB.misorientation),'contourf')
```





Misorientation axis distributions

The untextured, uncorrelated axis distribution

```
plotAxisDistribution(csHCP,csFCC)
```

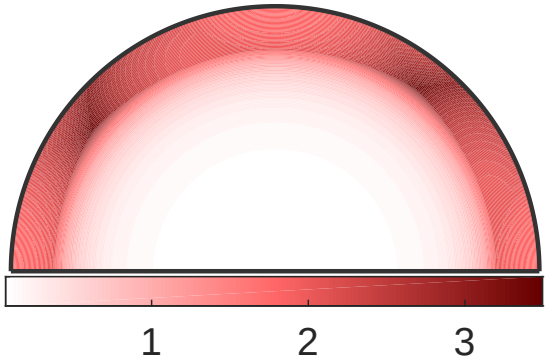
The textured, correlated axis distribution

```
mdf = calcMDF(gB.misorientations)
plotAxisDistribution(mdf)
```

```
[value,mori] = max(mdf)
plot(mori.axis)
```

The textured, uncorrelated axis distribution

```
mdf = calcMDF(odfFCC,odfHCP)
plotAxisDistribution(mdf)
```



Misorientation axis distributions

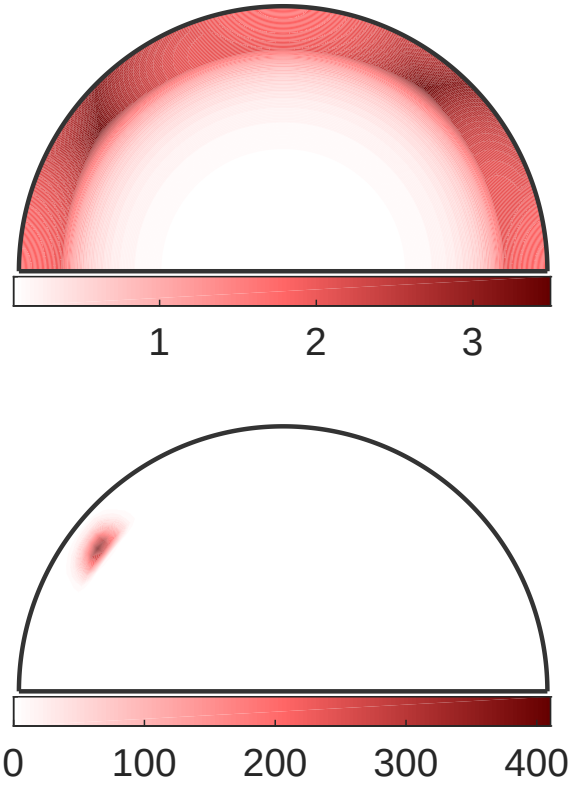
The untextured, uncorrelated axis distribution

```
plotAxisDistribution(csHCP,csFCC)
```

The textured, correlated axis distribution

```
mdf = calcMDF(gB.misorientations)
plotAxisDistribution(mdf)
```

```
mdf = MDF (show methods, plot)
  crystal symmetry : fcc (m-3m)
  crystal symmetry : hcp (6/mmm)

  Harmonic portion:
    degree: 58
    weight: 1
```




```
[value,mori] = max(mdf)
plot(mori.axis)
```

The textured, uncorrelated axis distribution

```
mdf = calcMDF(odfFCC,odfHCP)
plotAxisDistribution(mdf)
```

Misorientation axis distributions

The untextured, uncorrelated axis distribution

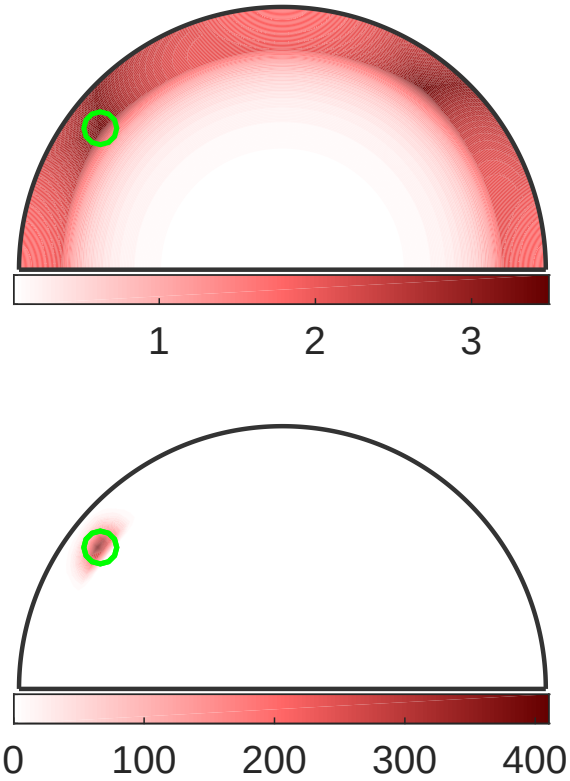
```
plotAxisDistribution(csHCP,csFCC)
```

The textured, correlated axis distribution

```
mdf = calcMDF(gB.misorientations)
plotAxisDistribution(mdf)
```

```
[value,mori] = max(mdf)
plot(mori.axis)
```

```
value =
  232.1792

mori = misorientation (show methods, plot)
  size: 1 x 1
  crystal symmetry : fcc (m-3m)
  crystal symmetry : hcp (6/mmm)

  Bunge Euler angles in degree
     phi1      Phi     phi2     Inv.
  210.936 54.7028 134.957        0
```



Misorientation axis distributions

The untextured, uncorrelated axis distribution

```
plotAxisDistribution(csHCP,csFCC)
```

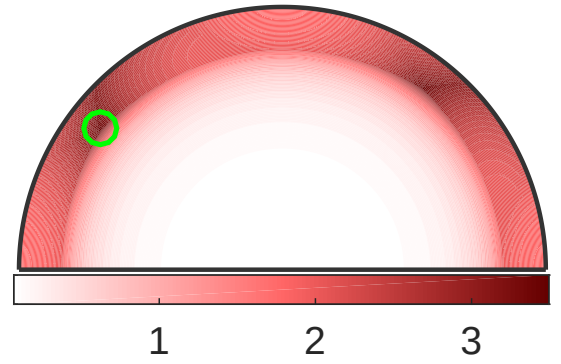
The textured, correlated axis distribution

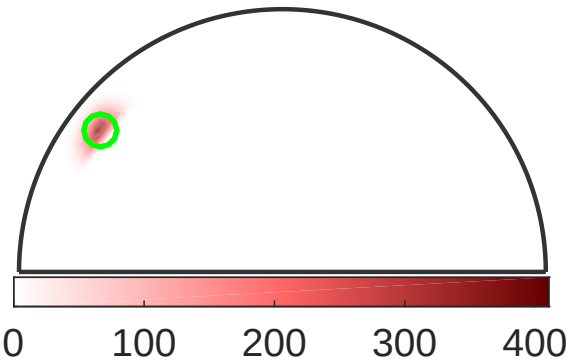
```
mdf = calcMDF(gB.misorientations)
plotAxisDistribution(mdf)
```

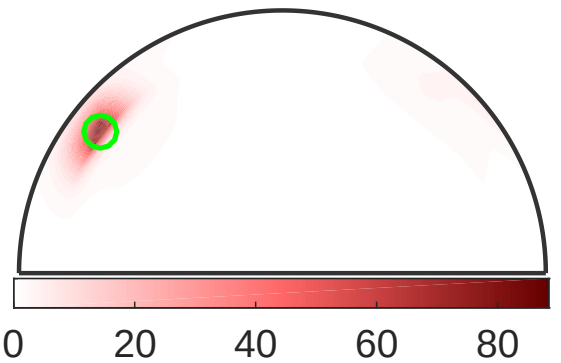
```
[value,mori] = max(mdf)
plot(mori.axis)
```

The textured, uncorrelated axis distribution

```
mdf = calcMDF(odfFCC,odfHCP)
plotAxisDistribution(mdf)
```







Misorientation axis in specimen coordinates

Surprisingly the misorientation axis in specimen coordinates is unique

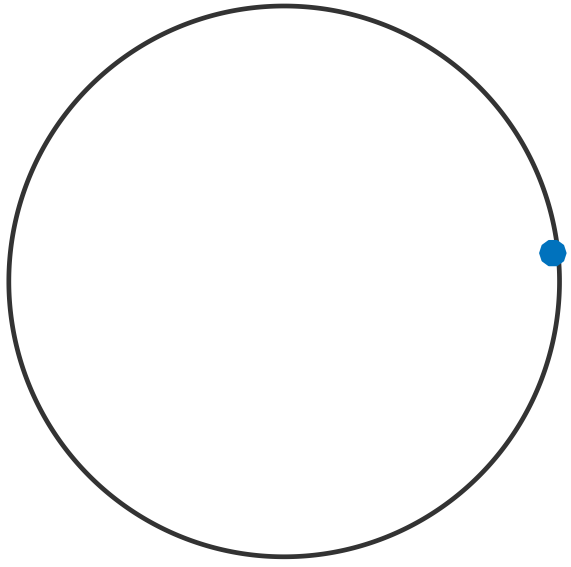
```
axis(O1,O2)
```

```
ans = vector3d (show methods, plot)
 size: 1 x 1
           x          y          z
  0.0495585   0.992437 -0.112305
```

```
[grains, ebsd.grainId] = ...
  calcGrains(ebsd)
```

```
gB = grains.boundary('fcc','hcp')
ids = gB.ebsdId;
oFCC = ebsd(ids(:,1)).orientations
oHCP = ebsd(ids(:,2)).orientations
```

```
plotAxisDistribution(oFCC,oHCP)
```



Misorientation axis in specimen coordinates

Surprisingly the misorientation axis in specimen coordinates is unique

```
axis(O1,O2)
```

```
[grains, ebsd.grainId] = ...
  calcGrains(ebsd)
```

```
gB = grains.boundary('fcc','hcp')
ids = gB.ebsdId;
oFCC = ebsd(ids(:,1)).orientations
oHCP = ebsd(ids(:,2)).orientations
```

```
plotAxisDistribution(oFCC,oHCP)
```

```
plotAxisDistribution(oFCC,oHCP,...
  'contourf')
```

Misorientation axis in specimen coordinates

Surprisingly the misorientation axis in specimen coordinates is unique

```
axis(O1,O2)
```

```
[grains, ebsd.grainId] = ...
  calcGrains(ebsd)
```

```
gB = grains.boundary('fcc','hcp')
ids = gB.ebsdId;
oFCC = ebsd(ids(:,1)).orientations
oHCP = ebsd(ids(:,2)).orientations
```

```
plotAxisDistribution(oFCC,oHCP)
```

```
plotAxisDistribution(oFCC,oHCP,...
  'contourf')
```

Misorientation axis in specimen coordinates

Surprisingly the misorientation axis in specimen coordinates is unique

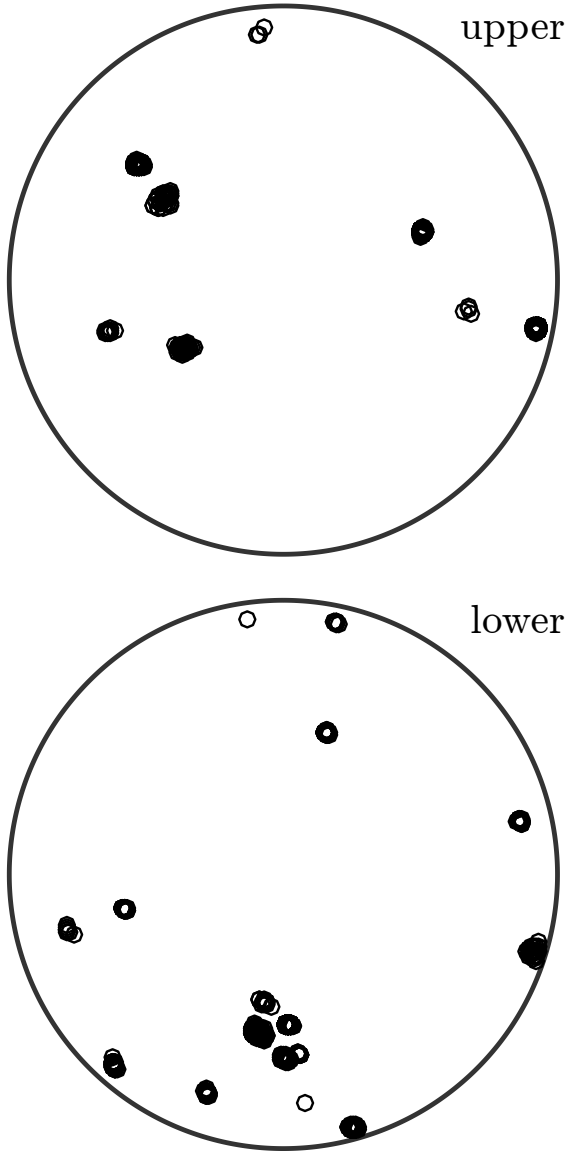
```
axis(O1,O2)
```

```
[grains, ebsd.grainId] = ...
  calcGrains(ebsd)
```

```
gB = grains.boundary('fcc','hcp')
ids = gB.ebsdId;
oFCC = ebsd(ids(:,1)).orientations
oHCP = ebsd(ids(:,2)).orientations
```

```
plotAxisDistribution(oFCC,oHCP)
```

```
plotAxisDistribution(oFCC,oHCP,...
  'contourf')
```



Misorientation axis in specimen coordinates

Surprisingly the misorientation axis in specimen coordinates is unique

```
axis(O1,O2)
```

```
[grains, ebsd.grainId] = ...
  calcGrains(ebsd)
```

```
gB = grains.boundary('fcc','hcp')
ids = gB.ebsdId;
oFCC = ebsd(ids(:,1)).orientations
oHCP = ebsd(ids(:,2)).orientations
```

```
plotAxisDistribution(oFCC,oHCP)
```

```
plotAxisDistribution(oFCC,oHCP,...
  'contourf')
```