X光编程作业

韩炤鸿

材73

2007012017

**源码部分（Ba面缺氧）：**

#include<stdio.h>

#include<stdlib.h>

#include<math.h>

#include <conio.h>

#include <ctype.h>

#include <string.h>

#define pi 3.1415926

#include<fstream>

#include<iostream>

double cal\_dist(int r[3]);//计算晶面距离

double cal\_theta(double d);//计算衍射角theta,单位为rad

void rank(struct plane \*pl,int h,int k,int l);//生成新链表与排序

void cal\_i(struct plane \*pl);//计算衍射强度

void rel\_i(struct plane \*pl);//计算相对强度，其中包含对分辨角的计算

struct plane

{

struct plane \*next; //链表存放下一个地址

int r[3]; // r0=h,r1=k,r2=l

double dist; //晶面间距

double theta; // 衍射角theta ,单位为rad

double i; //衍射强度

int con; //控制变量 ，0=独立，1=开始， 2=结束 ,3=重复上个数据

};

main()

{

struct plane \*crystal=(struct plane\*)malloc(sizeof(struct plane));

crystal->con=1;

struct plane \*print;

FILE \*fp;//打印指针

for(int h=0;h<=6;h++)

{

for(int k=0;k<=6;k++)

{

for(int l=0;l<=6;l++)

{

if(h+k+l!=0)

rank(crystal,h,k,l);

}

}

}

cal\_i(crystal);

rel\_i(crystal);

//打印

fp=fopen("data.txt","wb");

print=crystal->next;

while(1)

{

if(print->con!=3)

fprintf(fp,"%d%d%d %f %f %f\n",print->r[0],print->r[1],print->r[2],print->i,print->dist,print->theta\*360/pi);

else

fprintf(fp,"%d%d%d\n",print->r[0],print->r[1],print->r[2]);

if(print->con==2)

break;

else

print=print->next;

}

fclose(fp);

}

double cal\_dist(int r[3])

{

double d;

d=1/sqrt(r[0]\*r[0]/15.155449+r[1]\*r[1]/136.609344+r[2]\*r[2]/14.5924);//abc的平方已算出

return d;

}

double cal\_theta(double d)

{

double theta;

theta=asin(1.54178/(2\*d)) ;

return theta;

}

void rank(struct plane \*pl,int h,int k,int l)

{

struct plane \*temp=(struct plane\*)malloc(sizeof(struct plane));//临时存放面

struct plane \*point=pl ;//指针

temp->r[0]=h; temp->r[1]=k;temp->r[2]=l;

temp->dist=cal\_dist(temp->r);

temp->theta=cal\_theta(temp->dist);

//开始排序，对角度有限制

if(temp->theta<=2.62)

{

if (point->con==1)

{temp->con=2;point->next=temp;point->con=0;}

else

{

while(1)

{

if(temp->dist<point->next->dist)

if(point->next->con==2)

{point->next->con=0;point->next->next=temp; temp->con=2;break;}

else

{point=point->next;}

else

{temp->con=0;temp->next=point->next;point->next=temp;break;}

}

}

}

}

void cal\_i(struct plane \*pl)

{

struct plane \*point=pl->next ;//指针

double P,F\_2,LP,fi[4],fj[13];

double fi\_const[4][9]=

// a1 b1 a2 b2 a3 b3 a4 b4 c

{{17.9268,1.35417,9.15310,11.2145,1.76795,22.6599,-33.108,-0.01319,40.2602}, //Y

{20.1807,3.21367,19.1136,0.283310,10.9054,20.0558,0.77634,51.7460,3.0292}, //Ba

{11.8168,3.37484,7.11181,0.244078,5.78135,7.98760,1.14523,19.8970,1.14431}, //Cu

{4.19160,12.8573,1.63969,4.17236,1.52673,47.0179,-20.307,-0.01404,21.9412}} ; //O

double atom\_pos[13][3]=

//Y

{{1.0/2,1.0/2,1.0/2},

//Ba

{1.0/2,1.0/6,1.0/2},

{1.0/2,5.0/6,1.0/2},

//Cu

{0,0,0},

{0,1.0/3,0},

{0,2.0/3,0},

//O

{1.0/2,0,0},

{1.0/2,1.0/3,0},

{1.0/2,2.0/3,0},

{0,0,1.0/2},

{0,1.0/3,1.0/2},

{0,2.0/3,1.0/2},

{0,1.0/2,0}};

while(1)

{

//计算P

if(point->r[0]\*point->r[1]\*point->r[2]==0)

if(point->r[0]+point->r[1]==0||point->r[2]+point->r[1]==0||point->r[0]+point->r[2]==0)

P=2.0;

else

P=4.0;

else

P=8.0;

//计算 LP

LP=(1+pow(cos(2\*point->theta),2))/(pow(sin(point->theta),2)\*cos(point->theta));

//计算F\_2

F\_2=0;

for(int i=0;i<=3;i++)

{

fi[i]=0;

for(int x=0;x<=3;x++)

{fi[i]=fi[i]+fi\_const[i][2\*x]\*exp(-fi\_const[i][2\*x+1]\*pow(sin(point->theta)/1.54178,2));}

fi[i]=fi[i]+fi\_const[i][8];

}

fj[0]=fi[0];

fj[1]=fj[2]=fi[1];

fj[3]=fj[4]=fj[5]=fi[2];

fj[6]=fj[7]=fj[8]=fj[9]=fj[10]=fj[11]=fj[12]=fi[3];

for(int j=0;j<=12;j++)

{

for(int k=0;k<=12;k++)

{

F\_2=F\_2+fj[j]\*fj[k]\*cos(2\*pi\*(point->r[0]\*(atom\_pos[j][0]-atom\_pos[k][0])+point->r[1]\*(atom\_pos[j][1]-atom\_pos[k][1])+point->r[2]\*(atom\_pos[j][2]-atom\_pos[k][2])));

}

}

//计算i

point->i=LP\*F\_2\*P;

if(point->con==2)

break;

else

point=point->next;

}

}

void rel\_i(struct plane \*pl)

{

struct plane \*point=pl->next ;//指针

struct plane \*temp;

double imax=0;

//检查衍射角分辨率，小于0.05度

while(1)

{

if(point->next->con==2)

break;

if(point->next->theta-point->theta<0.000436111)

point->next->con=3;

point=point->next;

}

//合并强度

point=pl->next;

while(point->con!=2)

{

temp=point->next;

while(temp->con==3)

{ point->i=point->i+temp->i;temp=temp->next; }

if(imax<point->i)

imax=point->i;

point=temp;

}

//转为相对强度

point=pl->next;

while(1)

{

point->i=point->i\*100/imax;

if(point->con==2)

break;

else

point=point->next;

}

}

**对于Cu面缺氧**

Cal\_i函数的atom\_pos数组变为：

double atom\_pos[13][3]=

**//Y**

{{1.0/2,1.0/2,1.0/2},

**//Ba**

{1.0/2,1.0/6,1.0/2},

{1.0/2,5.0/6,1.0/2},

**//Cu**

{0,0,0},

{0,1.0/3,0},

{0,2.0/3,0},

**//O**

{1.0/2,1.0/3,0},

{1.0/2,2.0/3,0},

{0,1.0/3,1.0/2},

{0,2.0/3,1.0/2},

{0,1.0/6,0},

{0,1.0/2,0},

{0,5.0/6,0}};

**Ba面缺氧 计算数据**

**hkl I d theta**

010 1.324784 11.688000 7.563455

020 0.314276 5.844000 15.160113

030 4.884373 3.896000 22.824515

100

001 3.994501 3.820000 23.284905

110 1.612623 3.693508 24.094098

011 1.539645 3.630991 24.515334

120 1.121161 3.239939 27.529209

021 1.079740 3.197494 27.901994

040 0.065710 2.922000 30.594022

130 58.399942 2.753827 32.512543

031 100.000000 2.727621 32.833702

101

111 0.208222 2.655297 33.754416

121 0.174280 2.470889 36.358261

050 0.453153 2.337600 38.510653

140

041 0.406037 2.320872 38.799293

131 14.973252 2.233874 40.374706

150 0.249212 2.004067 45.245626

051 0.346266 1.993898 45.489259

141

060 31.778371 1.948000 46.623531

200

210 0.046024 1.920056 47.343163

002 14.892703 1.910000 47.607770

012 0.043898 1.884997 48.279156

220 0.041645 1.846754 49.344837

022 0.039855 1.815495 50.252830

151 0.075173 1.774670 51.492323

160 1.237457 1.742076 52.528582

230

061 1.830278 1.735384 52.746716

201

211 1.477380 1.715540 53.404887

032

102

112 0.279173 1.696577 54.050040

221 0.259863 1.662651 55.245905

122 0.250386 1.645372 55.876571

240 0.029693 1.619970 56.831625

042 0.028696 1.598747 57.656267

161 39.082549 1.585033 58.202673

231

132 20.881714 1.569452 58.836899

250 0.024153 1.495815 62.043665

241 0.176308 1.491404 62.247584

052 0.194546 1.479059 62.825977

142

251 0.138299 1.392839 67.210347

152 0.134767 1.382633 67.773364

260 9.312279 1.376913 68.093367

062 17.979757 1.363810 68.838913

202

212 0.037402 1.354115 69.402055

222 0.035580 1.327649 70.991576

300 0.203310 1.297667 72.891098

261 0.809582 1.295335 73.043489

310 0.053007 1.289742 73.411948

162 1.164699 1.287114 73.586479

232

003 0.194002 1.273333 74.516846

320 0.049922 1.266811 74.966272

013 0.049796 1.265844 75.033471

023 0.047052 1.244143 76.576091

242 0.029820 1.235445 77.214394

330 4.581451 1.231169 77.532483

301 3.806720 1.228706 77.717065

311 0.029061 1.221973 78.226747

033 7.978132 1.210330 79.125856

103

113 0.028074 1.203804 79.640068

321 0.028000 1.202417 79.750313

340 0.040500 1.185973 81.084134

123 0.027103 1.185095 81.156832

252 0.026730 1.177653 81.778474

331 1.615054 1.171812 82.273955

043 0.038647 1.167312 82.660397

133 1.556825 1.155762 83.671493

350 0.035694 1.134571 85.602173

341 0.024652 1.132642 85.782976

053 0.058413 1.118200 87.165715

143

262 9.334688 1.116937 87.289045

351 0.022913 1.087613 90.273200

360 0.282384 1.079980 91.089953

153 0.022487 1.074743 91.660437

302 0.280366 1.073371 91.811361

312 0.061903 1.068873 92.310176

063 0.556388 1.065831 92.651214

203

213 0.061010 1.061184 93.178057

322 0.060403 1.055712 93.807824

223 0.059622 1.048301 94.677093

361 5.019606 1.039245 95.766326

332 5.872095 1.034816 96.310224

163 10.745901 1.028000 97.162240

233

342 0.056178 1.007543 99.835139

243 0.055776 1.001095 100.716156

352 0.054606 0.975452 104.425240

400 1.825112 0.973250 104.760279

410 0.064744 0.969893 105.276400

253

420 0.010308 0.960028 106.832612

004 1.810827 0.955000 107.649366

014 0.010331 0.951828 108.173249

430 0.159570 0.944234 109.455714

401 0.276206 0.943121 109.647039

024 0.010378 0.942498 109.754599

362 0.608813 0.940103 110.170801

411

263 0.558531 0.934859 111.097111

421 0.054515 0.931075 111.778912

034 0.447281 0.927541 112.426241

104

114 0.054755 0.924594 112.974089

440 0.010553 0.923377 113.202392

431 5.536462 0.916646 114.489571

124 0.055194 0.916035 114.608503

303 2.387643 0.908864 116.030921

044 0.010792 0.907748 116.256849

313 0.021645 0.906128 116.587016

134 5.639818 0.902284 117.381767

450 0.010983 0.898487 118.182595

323 0.021986 0.898068 118.271966

441 0.056692 0.897528 118.387333

333 1.334000 0.885099 121.141866

054 0.069736 0.884068 121.379126

144

451 0.059898 0.874620 123.623454

460 4.013100 0.870635 124.609928

343 0.023962 0.867852 125.314279

402 4.058959 0.867162 125.490867

412 0.024244 0.864785 126.105469

154 0.062529 0.862118 126.807048

422 0.024965 0.857770 127.979130

064 8.418558 0.857497 128.053861

204

214 0.025273 0.855072 128.725152

461 0.740107 0.848867 130.499077

224 0.026138 0.848289 130.668848

353 0.026306 0.847090 131.023136

432 0.453459 0.846448 131.214200

164 1.280980 0.837423 134.013309

234

442 0.029050 0.831326 136.036158

363 6.668771 0.823630 138.769752

244 0.031139 0.822686 139.120681

452 0.034275 0.813023 142.947218

254 0.037935 0.804936 146.552632

462 15.675036 0.792213 153.352677

264 19.355874 0.784726 158.449756

500 0.586961 0.778600 163.860259

510 0.112501 0.776878 165.763033

403 2.127260 0.773248 171.048480

520 0.291360 0.771780 174.494969

413 0.671022 0.771561 175.219067

**Cu面缺氧 计算数据**

**hkl I d theta**

010 0.000650 11.688000 7.563455

020 4.174563 5.844000 15.160113

030 5.878310 3.896000 22.824515

100

001 1.864782 3.820000 23.284905

110 0.715958 3.693508 24.094098

011 0.684871 3.630991 24.515334

120 0.505856 3.239939 27.529209

021 0.488050 3.197494 27.901994

040 0.659107 2.922000 30.594022

130 47.884717 2.753827 32.512543

031 100.000000 2.727621 32.833702

101

111 1.956344 2.655297 33.754416

121 0.015372 2.470889 36.358261

050 0.202181 2.337600 38.510653

140

041 0.194220 2.320872 38.799293

131 14.041725 2.233874 40.374706

150 0.123515 2.004067 45.245626

051 0.136843 1.993898 45.489259

141

060 29.801352 1.948000 46.623531

200

210 0.007571 1.920056 47.343163

002 13.966188 1.910000 47.607770

012 0.007528 1.884997 48.279156

220 0.282080 1.846754 49.344837

022 0.264997 1.815495 50.252830

151 0.487444 1.774670 51.492323

160 1.409535 1.742076 52.528582

230

061 1.072703 1.735384 52.746716

201

211 1.501135 1.715540 53.404887

032

102

112 0.145235 1.696577 54.050040

221 0.136059 1.662651 55.245905

122 0.131538 1.645372 55.876571

240 0.173458 1.619970 56.831625

042 0.165059 1.598747 57.656267

161 38.519199 1.585033 58.202673

231

132 17.693054 1.569452 58.836899

250 0.006598 1.495815 62.043665

241 0.095681 1.491404 62.247584

052 0.099647 1.479059 62.825977

142

251 0.076805 1.392839 67.210347

152 0.075029 1.382633 67.773364

260 8.732936 1.376913 68.093367

062 16.861188 1.363810 68.838913

202

212 0.011998 1.354115 69.402055

222 0.164479 1.327649 70.991576

300 0.134041 1.297667 72.891098

261 0.534124 1.295335 73.043489

310 0.030195 1.289742 73.411948

162 1.273799 1.287114 73.586479

232

003 0.128839 1.273333 74.516846

320 0.028601 1.266811 74.966272

013 0.028535 1.265844 75.033471

023 0.027109 1.244143 76.576091

242 0.127426 1.235445 77.214394

330 3.936278 1.231169 77.532483

301 4.269202 1.228706 77.717065

311 0.122780 1.221973 78.226747

033 7.805103 1.210330 79.125856

103

113 0.116821 1.203804 79.640068

321 0.010474 1.202417 79.750313

340 0.023671 1.185973 81.084134

123 0.010300 1.185095 81.156832

252 0.010226 1.177653 81.778474

331 1.514577 1.171812 82.273955

043 0.022690 1.167312 82.660397

133 1.459971 1.155762 83.671493

350 0.021119 1.134571 85.602173

341 0.009800 1.132642 85.782976

053 0.030081 1.118200 87.165715

143

262 8.753951 1.116937 87.289045

351 0.087011 1.087613 90.273200

360 0.198402 1.079980 91.089953

153 0.084609 1.074743 91.660437

302 0.197340 1.073371 91.811361

312 0.037165 1.068873 92.310176

063 0.392434 1.065831 92.651214

203

213 0.036688 1.061184 93.178057

322 0.036364 1.055712 93.807824

223 0.035948 1.048301 94.677093

361 5.545945 1.039245 95.766326

332 5.082620 1.034816 96.310224

163 10.479440 1.028000 97.162240

233

342 0.034137 1.007543 99.835139

243 0.033932 1.001095 100.716156

352 0.033366 0.975452 104.425240

400 1.711567 0.973250 104.760279

410 0.037858 0.969893 105.276400

253

420 0.036144 0.960028 106.832612

004 1.698171 0.955000 107.649366

014 0.004614 0.951828 108.173249

430 0.258678 0.944234 109.455714

401 0.200770 0.943121 109.647039

024 0.036079 0.942498 109.754599

362 0.436687 0.940103 110.170801

411

263 0.406709 0.934859 111.097111

421 0.033531 0.931075 111.778912

034 0.470952 0.927541 112.426241

104

114 0.033708 0.924594 112.974089

440 0.036376 0.923377 113.202392

431 4.806886 0.916646 114.489571

124 0.034015 0.916035 114.608503

303 2.619564 0.908864 116.030921

044 0.036961 0.907748 116.256849

313 0.074085 0.906128 116.587016

134 4.897914 0.902284 117.381767

450 0.005010 0.898487 118.182595

323 0.010031 0.898068 118.271966

441 0.035015 0.897528 118.387333

333 1.251008 0.885099 121.141866

054 0.041313 0.884068 121.379126

144

451 0.037085 0.874620 123.623454

460 3.763434 0.870635 124.609928

343 0.011035 0.867852 125.314279

402 3.806440 0.867162 125.490867

412 0.011174 0.864785 126.105469

154 0.038759 0.862118 126.807048

422 0.084062 0.857770 127.979130

064 7.894817 0.857497 128.053861

204

214 0.011678 0.855072 128.725152

461 0.547728 0.848867 130.499077

224 0.087782 0.848289 130.668848

353 0.088316 0.847090 131.023136

432 0.704295 0.846448 131.214200

164 1.339706 0.837423 134.013309

234

442 0.097139 0.831326 136.036158

363 7.300607 0.823630 138.769752

244 0.103919 0.822686 139.120681

452 0.015984 0.813023 142.947218

254 0.017717 0.804936 146.552632

462 14.699849 0.792213 153.352677

264 18.151692 0.784726 158.449756

500 0.438604 0.778600 163.860259

510 0.070143 0.776878 165.763033

403 1.590520 0.773248 171.048480

520 0.181708 0.771780 174.494969

413 0.418491 0.771561 175.219067