

RMC PRACTICAL

Modelling the structure of molten Copper

All the files needed for this practical, including programs, are found in the file **rmca_cu.zip**. Extract them all into the same directory. You should have the following:

```
confplot.exe
convol.exe
mcgr_31.exe
moveout.exe
neigh.exe
random.exe
rmca.exe
rmcplot.exe
show.exe
triplets.exe
cu_mcgr.dat
cu_mcgr_0.dat
cu1.dat
cu2.dat
cusq.dat
```

The general approach in this practical is as follows:

- Various relevant distances will be determined from $G(r)$ which is obtained from the experimental $F(Q)$ using the MCGR program. Both $G(r)$ and $F(Q)$ will be used as original data in two examples.
- A configuration of Cu atoms with certain minimum separation will be created.
- The data are fitted using the RMCA program to produce the final model.

The times given in this practical, for running MCGR and RMC is real time and NOT CPU time. The times may have to be increased on a slow computer. The outputs you get, will probably not be identical to the outputs in this example.

1. Neutron scattering data of molten copper is in the file **cusq.dat**. Plot it with the program **show**.

```
File format (RAW, TEXT, DATA)? data  
Full File name: cusq.dat  
Blocks no: 1 to 1 exists.  
> d 1
```

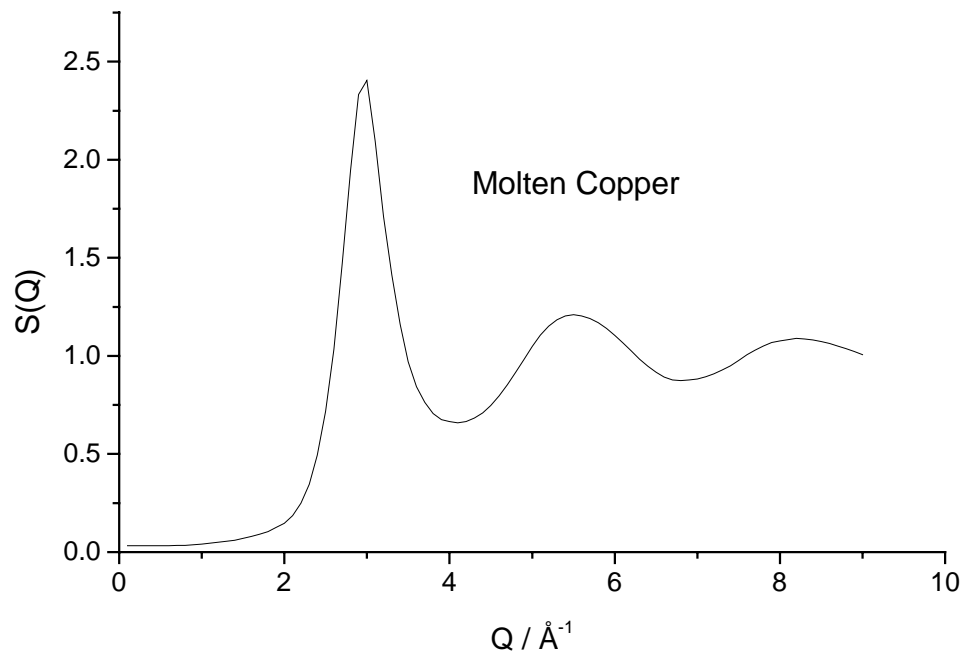
Now you should have a plot on the screen.

To plot with other limits

```
> d 1 xlow xhigh Ylow Yhigh
```

To exit

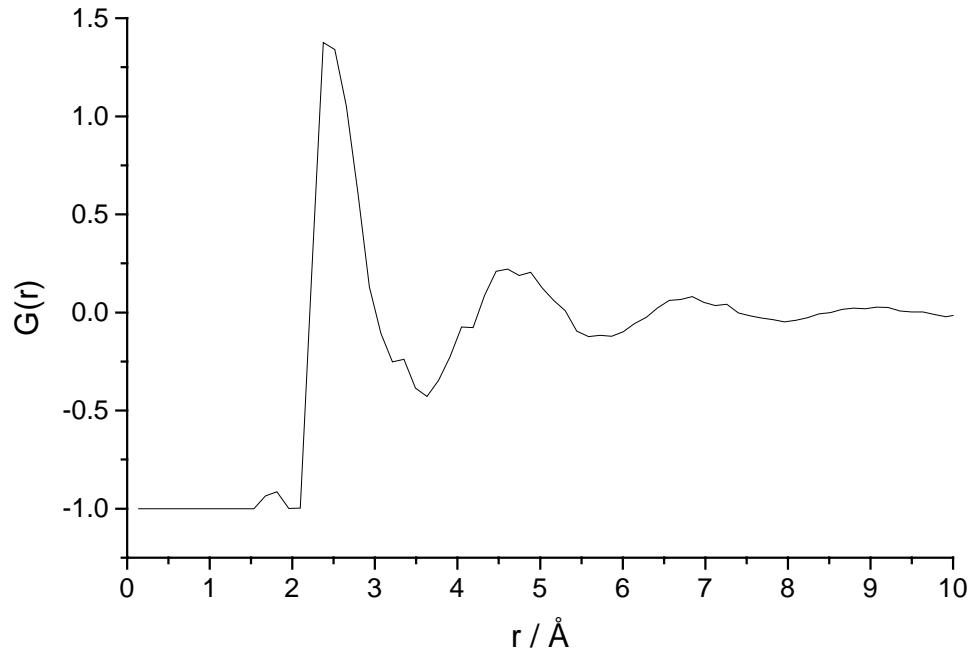
```
> e
```



2. We need to create a radial distribution function to use as the initial 'data' in this example. In practice it is always best to model $G(r)$ first anyway. We can also determine some relevant distances for creating the initial RMC configuration and make some checks on errors in the experimental data. $G(r)$ is produced using the MCGR program. First do a short run to find a good value of the closest approach value. The control data are in **cu_mcgr_0.dat**. Use an editor (like Wordpad) to look at the control data.

```
Liquid copper (MCGR control file)
.false.      ! rerun
.false.      ! plot
0.0721       ! density
.false.      ! generate partials
1            ! no of partials
1            ! no of zero constr.
1 0. 1.0
0.05         ! delta
5 20.        ! mr rmax
.false.      ! save
.false.      ! converge only
1            ! no of positivity constr.
1 0. 20.1
0            ! no of coord. constr.
.true.       ! smoothing
3            ! nsmooth
1            ! nchanges
0.2 21.      ! gau_sig,r_change
.false.      ! resolution
100          ! printing
5 5          ! time limits
1 0          ! no of data sets
cusq.dat
1 100        ! used points
1.           ! const. to subtract
1.0          ! coeff
0.005        ! sigma
.false.      ! renormalise
.true.       ! background
1            ! nparm
.false.      ! magnetic
cu_mcgr
```

We start with a closest approach of 1 Å (this is less than expected) .Run the program **mcgr_31** and give cu_mcgr_0.dat as input file. When this finish look at the produced $G(r)$ in **cu_mcgr_0.g** with the program **show**.



We can see from the plot that a closest approach of 2.0 Å is appropriate. Now lets do a real run with this value. The control data are in **cu_mcgr.dat**. First we have a look at it:

```
Liquid copper (MCGR control file)
.false.          ! rerun
.true.           ! plot
500              ! nplot
0.0721          ! density
.false.         ! generate partials
1               ! no of partials
1               ! no of zero constr.
1 0. 2.0
0.05            ! delta
5 20.          ! mr rmax
.false.         ! save
.false.         ! converge only
1               ! no of positivity constr.
1 0. 20.1
0               ! no of coord. constr.
.true.          ! smoothing
3               ! nsmooth
1               ! nchanges
0.2 21.        ! gau_sig,r_change
.false.         ! resolution
100             ! printing
15 5           ! time limits
1 0            ! no of data sets
cusq.dat
1 100          ! used points
1.             ! const. to subtract
1.0           ! coeff
0.005         ! sigma
```

```

.false.          ! renormalise
.true.           ! background
1               ! nparam
.false.         ! magnetic
cu_mcgr

```

The resolution in r depends on maximum Q value Q_{\max} in $F(Q)$, the relation is $dr=2\pi/(mr Q_{\max})$. mr gives the number of points across the resolution width (5 - 7 points recommended). In our example we get a dr of 0.1396 Å. $G(r)$ will be calculated out to $r = 20$ Å. The cut off value 2.0 Å was found in the short run previously.

Run the program **mcgr_31** and give **cu_mcgr.dat** as input file. This will take 15 minutes. On the screen you can see how χ^2 decreases as the fitted $F(Q)$ approaches the experimental $F(Q)$. You can also see how the fit of $F(Q)$ changes together with corresponding $G(r)$. Expected $\chi^2 \sim 0.7$.

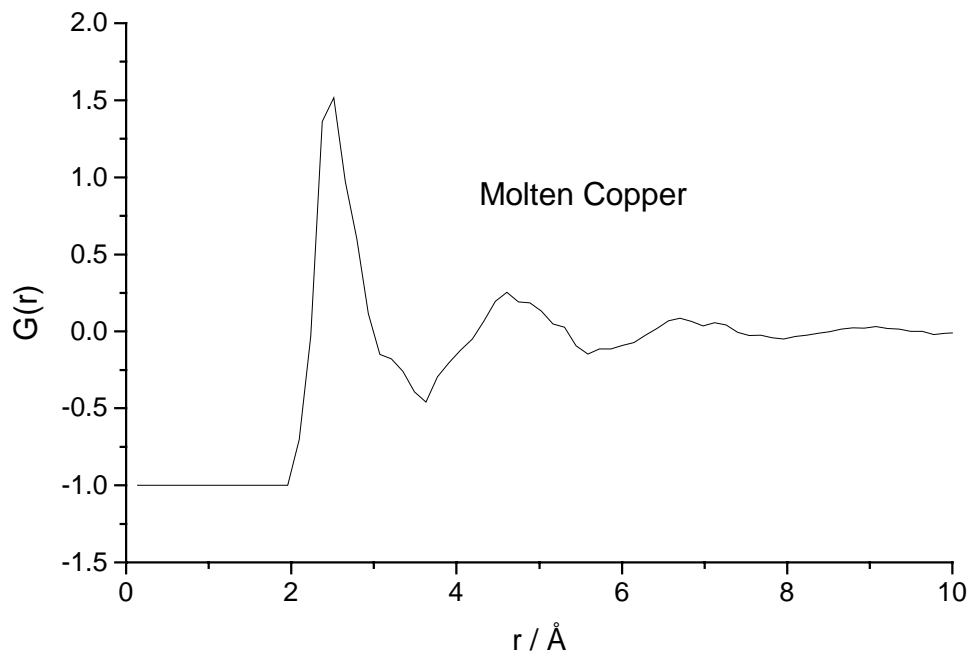
3. While this is running you can start making the initial configuration for the RMC model. First we make a random structure of Cu atoms, at the density that they will have in the final model, using the program **random**:

```

Number of Euler angles      > 0
Number of particle types    > 1
Density > 0.0721
Number of particles of type 1 > 500
Output file                 [.cfg] > cu_ran

```

4. Plot the resulting $G(r)$ from the MCGR run, **cu_mcgr.g**, with the program **show**.



5. Back to the configuration for RMC. The random Cu atoms now have to be moved apart to a suitable separation. From $G(r)$ we found that a value of 2.0 Å can be used. The most efficient way to move atoms apart, at least until the majority satisfy the closest approach constraints, is to use the **moveout** program. An example run of this program is shown below. The exact replies required to the prompts may differ slightly because the initial random configurations are themselves different.

MOVEOUT

```
Starting configuration      : cu_ran.cfg
Output file                : cu_mov.cfg
Closest approaches        : 0.5
```

22 atoms of type 1 have too close neighbours

```
Move atoms of type 1 ? (T/F) : t
Maximum move                  : 1.
Max. no. of iterations        : 50000
```

```
21 atoms of type 1 have too close neighbours after      0
iterations
20 atoms of type 1 have too close neighbours after      0
iterations
19 atoms of type 1 have too close neighbours after      0
iterations
18 atoms of type 1 have too close neighbours after      0
iterations
17 atoms of type 1 have too close neighbours after      0
iterations
16 atoms of type 1 have too close neighbours after      0
iterations
15 atoms of type 1 have too close neighbours after      0
iterations
14 atoms of type 1 have too close neighbours after      0
iterations
13 atoms of type 1 have too close neighbours after      1
iterations
12 atoms of type 1 have too close neighbours after      1
iterations
11 atoms of type 1 have too close neighbours after      1
iterations
10 atoms of type 1 have too close neighbours after      1
iterations
9 atoms of type 1 have too close neighbours after       1
iterations
8 atoms of type 1 have too close neighbours after       2
iterations
7 atoms of type 1 have too close neighbours after       2
iterations
6 atoms of type 1 have too close neighbours after       2
iterations
5 atoms of type 1 have too close neighbours after       2
iterations
4 atoms of type 1 have too close neighbours after       2
iterations
3 atoms of type 1 have too close neighbours after       2
iterations
2 atoms of type 1 have too close neighbours after       2
iterations
1 atoms of type 1 have too close neighbours after       2
iterations
```

0 atoms of type 1 have too close neighbours after 2
iterations
Re-calculate neighbours? (T/F) : **f**
Change cut-offs ? (T/F) : **t**
Closest approaches : **1.0**

138 atoms of type 1 have too close neighbours

Move atoms of type 1 ? (T/F) : **t**
Maximum move : **1.**
Max. no. of iterations : **50000**

.
.
.

Continue to increase the closest approaches until you reach a cut-off of 2.0 Å.

6. Now we are ready to model the configuration with RMC. First we run using $G(r)$ we obtained from MCGR as 'data' for the model. The control file is **cu1.dat**.

```
Molten copper at 1833K (example)
0.0721          ! number density
1.8             ! cut offs
0.1            ! maximum moves
1.396263E-01   ! r spacing
.false.        ! moveout option
0              ! number of configurations to collect
500            ! step for printing
20 5          ! time limit, time for saving
1 0 0 0       ! sets of experiments
cu_mcgr.g
1 90          ! points to use
0.            ! const. to subtract
1.            ! coeff.
0.01         ! sigma
.false.      ! renormalise
0            ! no of coord. constr.
0            ! no of av. coord. constr.
.false.     ! potential
```

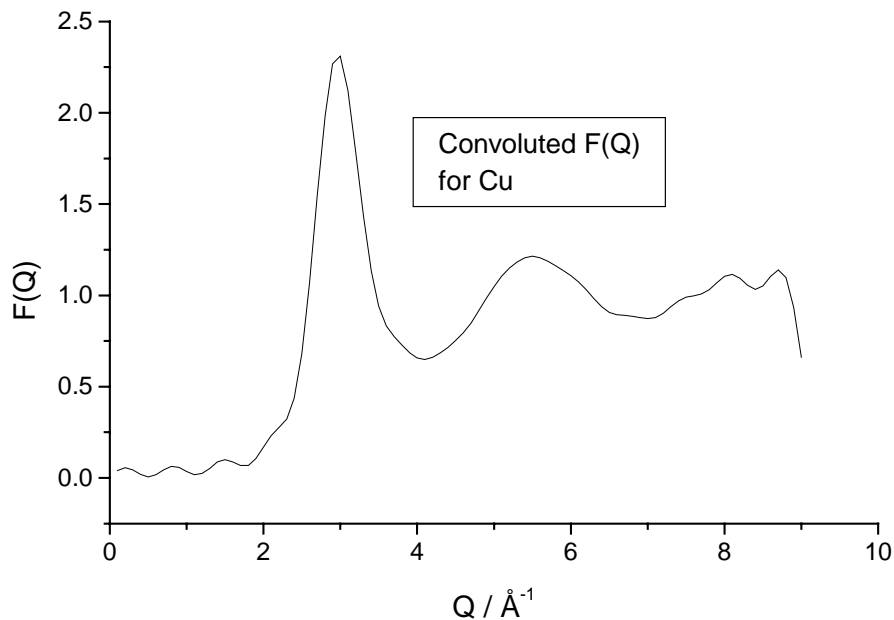
The choice of closest approach constraints has been described above. Maximum moves of 0.1 Å are set for each atom

Copy **cu_mov.cfg** to **cu1.cfg** and run **rmca**, using **cu1.dat** as control file. This will take 20 minutes. Expected $\chi^2/nq \sim 2$.

7. While this is running prepare for a RMC calculation using $F(Q)$ as data instead of $G(r)$. $F(Q)$ is measured from an "infinitely" big sample while the fit of $F(Q)$ is calculated from a limited volume. To account for this we need to use the structure factor obtained by convoluting the "measured" $G(r)$ with a step function that is unity for $r < L/2$ and zero for $r > L/2$, here L is the (minimum) box length of the configuration. Look at the file **cu_mov.cfg**, half box length is 9.534874 Å. Use the program **convol** to do this:

```
Input file          > cusq.dat
Truncation distance > 9.534874
Output file         > cusq_c.dat
```

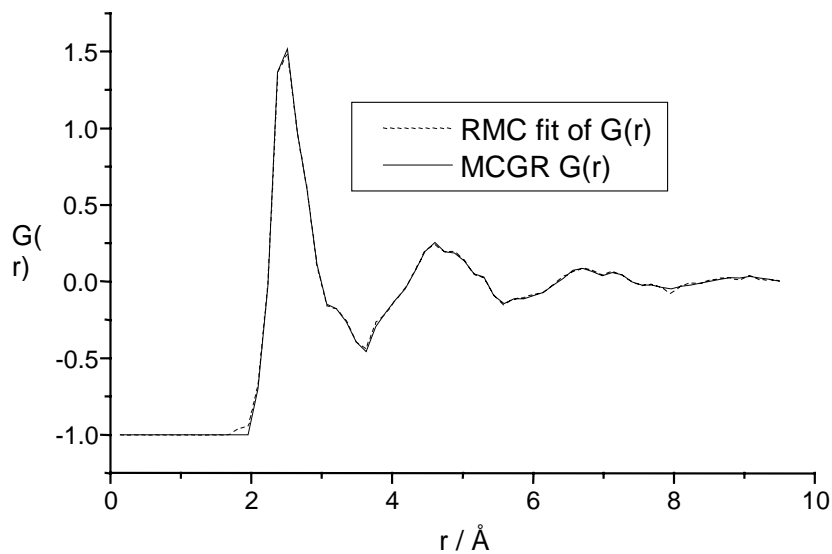
Look at **cusq_c.dat** using the program **show**:



The extra oscillations in $F(Q)$ comes from the limited configuration size. The dip at the high Q end occurs because of lack of points beyond Q_{\max} .

8. When RMCA has finished, look at the resulting fit to $G(r)$ using the program **rmcplot**:

```
Graphics device/type (? to see list, default /NULL): /ws
File to plot (or RETURN to exit) > cu1.out
Input file contains 3 groups of plots:
  Group 1 contains 1 plots of 1 curves
  Group 2 contains 1 plots of 1 curves
  Group 3 contains 1 plots of 2 curves
Plot which group (enter 0 to exit) ? 3
```



9. Now make another run using the convoluted structure factor as data to **rmca**, look at the control file **cu2.dat**.

```
Molten copper at 1833K (example)
0.0721          ! number density
1.8             ! cut offs
0.1            ! maximum moves
0.1            ! r spacing
.false.        ! moveout option
0              ! number of configurations to collect
500            ! step for printing
20 5           ! time limit, time for saving
0 1 0 0        ! sets of experiments
cusq_c.dat
1 89           ! points to use
1.             ! const. to subtract
1.            ! coeff.
0.01          ! sigma
.false.       ! renormalise
.true.        ! offset
0             ! no of coord. constr.
0             ! no of av. coord. contsr.
.false.       ! potential
```

We skip the last Q value in the fit (because of the dip described above) and only use points 1 to 89. Copy **cu_mov.cfg** to **cu2.cfg** and run **rmca**, using **cu2.dat** as control file. Expected $\chi^2/nq \sim 4.5$.

10. Look at the **cu2.out** file with **rmcplot** both at the fit and on $G(r)$ produced to see that they are reasonable.

11. The distribution of neighbours in the liquid can be checked using the program **neigh**. In $G(r)$ we see that the first minimum is at $\sim 3.4 \text{ \AA}$.

NEIGHBOURS

```
Configuration      : cu1.cfg
Minimum bond lengths : 0.0
Maximum bond lengths : 3.4
Output file        : cu1.nei
```

Look at the file **cu1.nei**.

Calculation of neighbours in **cu1.cfg**

No. of atom types = 1

```
Minimum bond lengths = 0.000000E+00
Maximum bond lengths = 3.400000
```

Type 1 - Type 1 neighbours:

7	2	.400
8	1	.200
9	13	2.600
10	50	10.000
11	115	23.000
12	138	27.600
13	99	19.800
14	63	12.600
15	17	3.400
16	2	.400

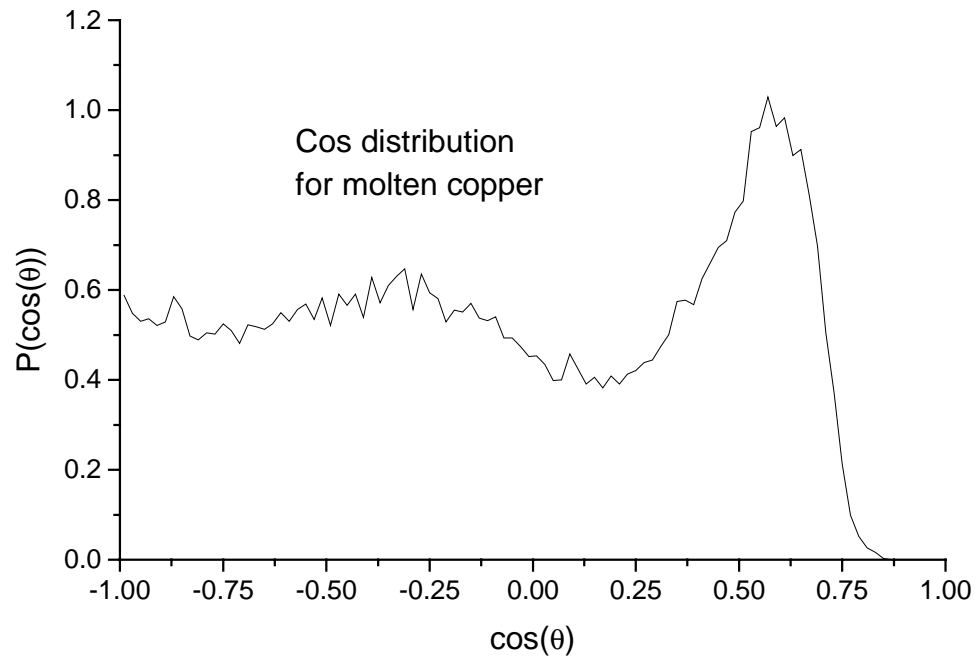
Average coordination 12.03

Most atoms have a coordination of 12 neighbours which is the coordination for close packed liquids, do the same for **cu2.cfg**.

12. Look at the bond angle distribution using the program **triplets**.

```
No. of theta pts > 100
No. of neighbours for bond ang (0 for all) > 0
Number of configurations > 1
Configuration file > cu1.cfg
Maximum r values > 3.4
Output file > cu1.tri
(A)ngle or (C)osine distribution [C] > c
```

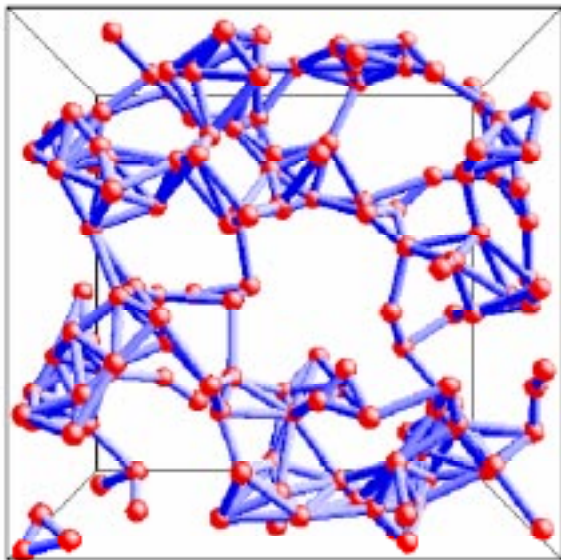
Do the same for **cu2.cfg** and plot the results with **rmcplot**:



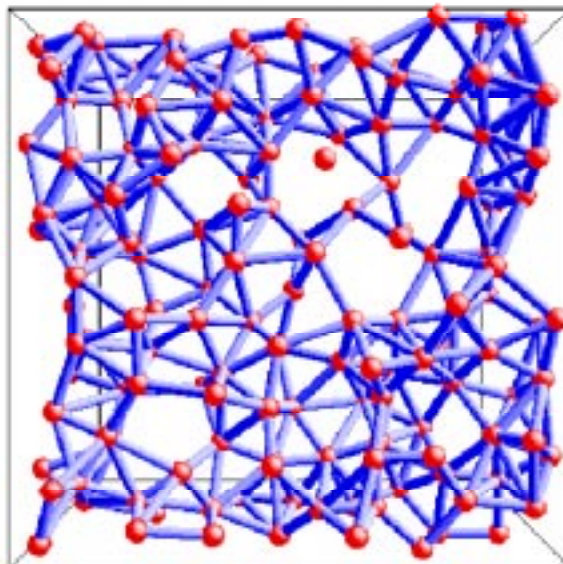
The main peak occurs around 0.6 or 50° .

To have a better model repeat step 3 with 4000 atoms, then step 5. Run RMC (step 9) but fitting directly to **cusq.dat** rather than **cusq_c.dat**.

13. Finally we can look at the distribution in three dimensions with the program **confplot**. For details how to use it, look in the manual for this program. Here are some examples in which only a thin slice has been plotted in order to see the details:



Plot of **cu_ran.cfg**: "bonds" between atoms $< 3 \text{ \AA}$ apart. The configuration is disordered.



Plot of **cu_mov.cfg**: here the atoms are more evenly spaced.

Right: **cu1.cfg**: some higher order between atoms might be seen.

