RMC PRACTICAL

Modelling the structure of vitreous SiO_2

All the files needed for this practical, including programs, are found in the file **rmca_sio2.zip**. Extract them all into the same directory. You should have the following:

confplot.exe
convol.exe
coord.exe
extract.exe
mcgr_31.exe
midpt.exe
moveout.exe
neigh.exe
random.exe
rmca.exe
rmcplot.exe
show.exe
triplets.exe
si_net.dat
sio2.dat
sio2_fq.dat
sio2_mcgr.dat
sio2_mcgr_0.dat
sio2_neutron.fq

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. Because this is a quick example and the model is small this *G(r)* will initially be used as the 'data'.
- A network of Si atoms that are 4-fold coordinated will be created using RMCA as a hard sphere Monte Carlo simulation with coordination constraints.
- The bonds in this network will be 'decorated' with O atoms to produce a network of corner sharing SiO_4 units.
- The topology of this network will be maintained while 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 <u>NOT</u> 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. The model is based on the neutron scattering data of Desa et al. This is in the file **sio2_neutron.fq**. Plot it with the program **show**.

```
File format (RAW, TEXT, DATA)? data
Full File name: sio2_neutron.fq
Blocks no: 1 to 1 exists.
> d 1
```

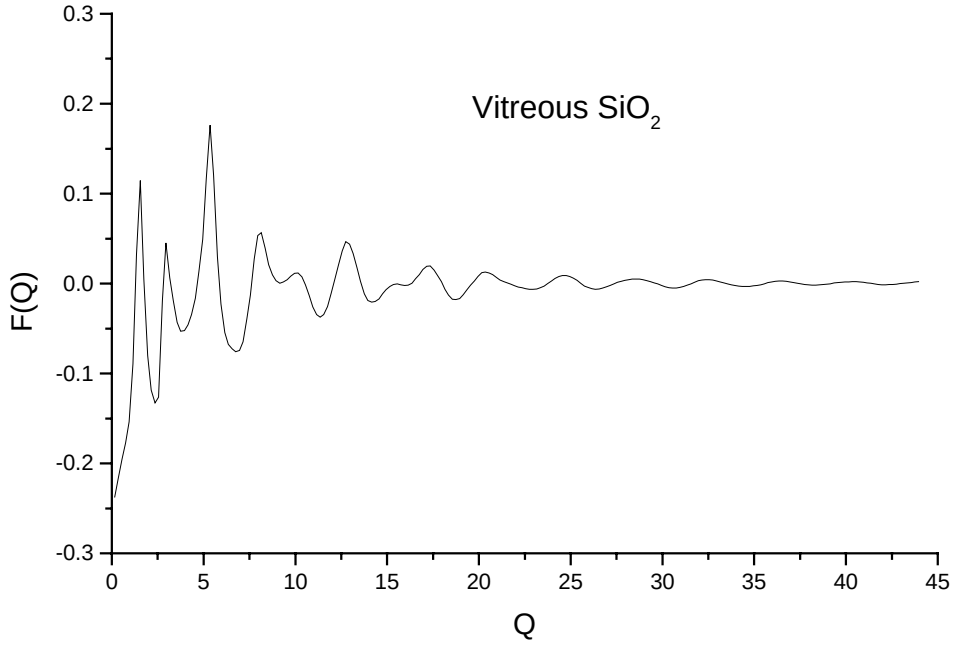
Now you should have a plot on the screen.

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 'data' in this small 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 a quick run is made to find the closest approach of atoms. The control data are in **sio2_mcgr_0.dat**. Use an editor (like Wordpad) to look at the control data.

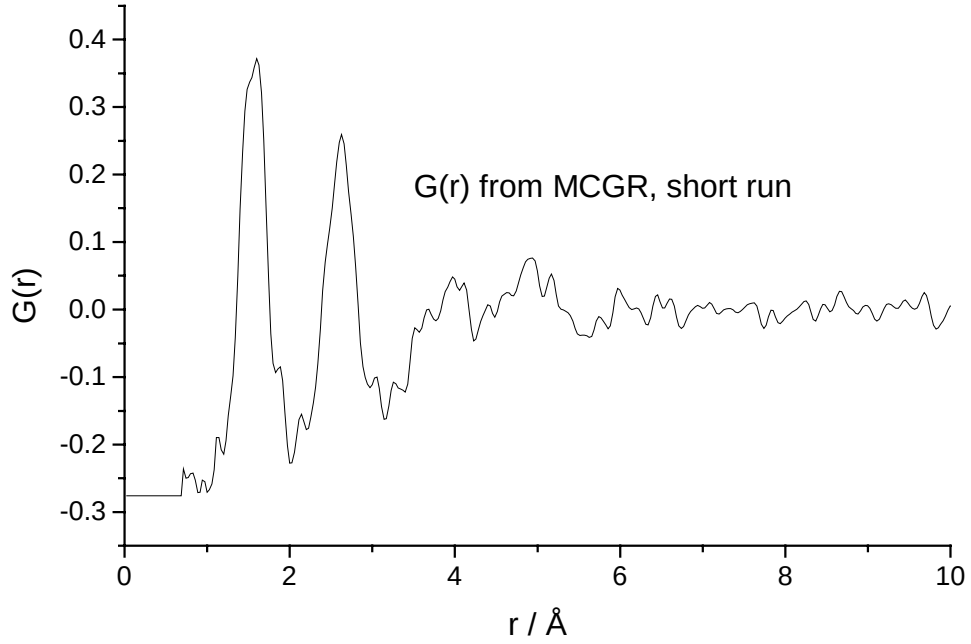
```
SiO2
.false.          ! rerun
.false.          ! plot
0.0657           ! number density
.false.          ! fitting partials
1                ! n_partials
1                ! no of zero constraints
1 0.0 0.7        ! zero constraint
0.01             ! delta
5 ,10.0          ! mr, rmax
.false.          ! save multi
.false.          ! converge only
1                ! no of positivity contraints
1 0.0 50.        ! positivity constraint
0                ! no of coordination constraints
.true.           ! smooth
3
1
0.1 50.
.false.          ! conv_resol
1000             ! step for printing
4 4              ! time limit, step_save
1 0              ! n_data sets
sio2_neutron.fq
1 220            ! points to fit
0.0              ! const to subtract
0.2759           ! coeff.
0.01             ! sigma
.false.          ! renorm
.true.           ! background
1                ! nback
.false.          ! magnetic
sio2_mcgr_0
```

Run the program **mcgr_31** and give **sio2_mcgr_0.dat** as input file. This will take 4 minutes. On the screen you can see how χ^2 decreases as the fitted *F(Q)* approaches the experimental *F(Q)*. The *r* spacing for *G(r)* will be 0.029 Å because the maximum *Q* value in the original data is 45 $Å^{-1}$ (see MCGR manual). This gives an *r* space resolution of $2\pi/45 \sim 0.15$ Å, giving 5 *r* points across the resolution width (5 points are recommended).

3. While this is running you can start making the initial configuration for the RMC model. First we make a random structure of Si 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 > 0.0219
Number of particles of type 1 > 100
Output file (no extension, will be .cfg)
si_ran
```

4. Plot the resulting *G(r)* from the MCGR run, **`sio2_mcgr_0.g`**, with the program **`show`**. You need to choose a closest approach of atoms that will be used both for a 'proper' run of MCGR and also later in RMC. A suitable value is about 1.4 Å.



5. Now do a better run of MCGR. This includes calculation to higher *r* (20 Å). We can also let the program plot the resulting *G(r)* and *F(Q)* as it is running. The control data file is in **`sio2_mcgr.dat`**.

```
SiO2
.false.          ! rerun
.true.           ! plot
5000             ! update plot
0.0657           ! number density
.false.          ! fitting partials
1                ! n_partials
1                ! no of zero constraints
1 0.0 1.4        ! zero constraint
0.01             ! delta
5 ,20.0          ! mr, rmax
.false.          ! save multi
.false.          ! converge only
1                ! no of positivity contraints
1 0.0 50.        ! positivity constraint
0                ! no of coordination constraints
.true.           ! smooth
3
1
0.1 50.
.false.          ! conv_resol
1000             ! step for printing
60 15            ! time limit, step_save
1 0              ! n_data sets
sio2_neutron.fq
1 220            ! points to fit
0.0              ! const to subtract
0.2759           ! coeff.
0.01             ! sigma
.false.          ! renorm
.true.           ! background
1                ! nback
.false.          ! magnetic
sio2_mcgr
```

This will take 60 minutes (if needed, run again setting `rerun` to `.true.`). Expected final χ^2~0.19.

6. Back to the configuration for RMC. The random Si atoms now have to be moved apart to a suitable separation. With some knowledge of the structure of silica, i.e. that it is a network of corner sharing SiO_4 tetrahedra, and from *G(r)* found in step 8 (we are jumping ahead slightly here), it can be estimated that the closest Si-Si distance is slightly above the first O-O peak, which is the second peak in *G(r)*. A value of 2.9 Å will be used (this is more easily found from X-ray data). 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.

```
 Starting configuration (no extension) ?
si_ran
 (Version 3 format configuration file)

 Output file                  : si_mov
 Closest approaches           : 1.0

   10 atoms of type 1 have too close neighbours

 Move atoms of type 1 ? (T/F) : t
 Maximum move                 : 1.0
 Max. no. of iterations       : 50000

    9 atoms of type 1 have too close neighbours after    0 iterations
    8 atoms of type 1 have too close neighbours after    0 iterations
    7 atoms of type 1 have too close neighbours after    1 iterations
    6 atoms of type 1 have too close neighbours after    3 iterations
    5 atoms of type 1 have too close neighbours after    3 iterations
    4 atoms of type 1 have too close neighbours after    3 iterations
    3 atoms of type 1 have too close neighbours after    5 iterations
    2 atoms of type 1 have too close neighbours after    7 iterations
    1 atoms of type 1 have too close neighbours after    7 iterations
    0 atoms of type 1 have too close neighbours after    7 iterations


 Re-calculate neighbours? (T/F) : f
 Change cut-offs ? (T/F)      : t
 Closest approaches           : 1.5

   26 atoms of type 1 have too close neighbours

 Move atoms of type 1 ? (T/F) : t
 Maximum move                 : 1.0
 Max. no. of iterations       : 50000
                                  .
                                  .
                                  .
```

Continue to increase the closest approaches until you reach a cut-off of 2.9 Å.

7. Now we need to make the initial Si network for the RMC model. This is done using program **rmca**, without fitting to any data but with coordination constraints. Start from the final configuration produced in step **6**.

Copy **si_mov.cfg** to **si_net.cfg**.

The RMCA data file is in **si_net.dat**. The maximum distance of 3.5 Å for Si-Si bonds is estimated from *G(r)* (actually done in the next step).

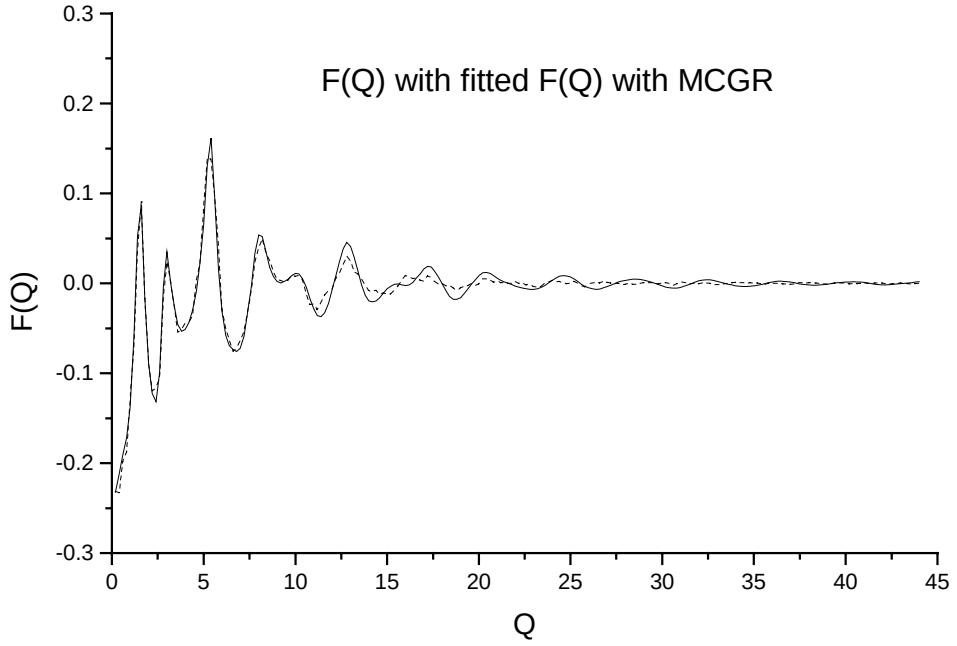
```
Making Si network for SiO2
0.0219              ! number density
2.9                 ! cut offs
0.3                 ! maximum moves
0.1                 ! r spacing
.false.             ! moveout option
0                   ! number of configurations to collect
2000                ! step for printing
15 15               ! Time limit, step for saving
0 0 0 0             ! sets of experiments
1
1 1 2.9 3.5 4 1.0 0.00001
0
.false.
```

Run **rmca** and give **si_net** as the input file name (no extension). The coordination constraint should be ~ 99% satisfied at the end. Have a look at the configuration with the program **confplot** (see point 18).

8. Look at the latest MCGR output. You can use the **`rmcplot`** program for this.

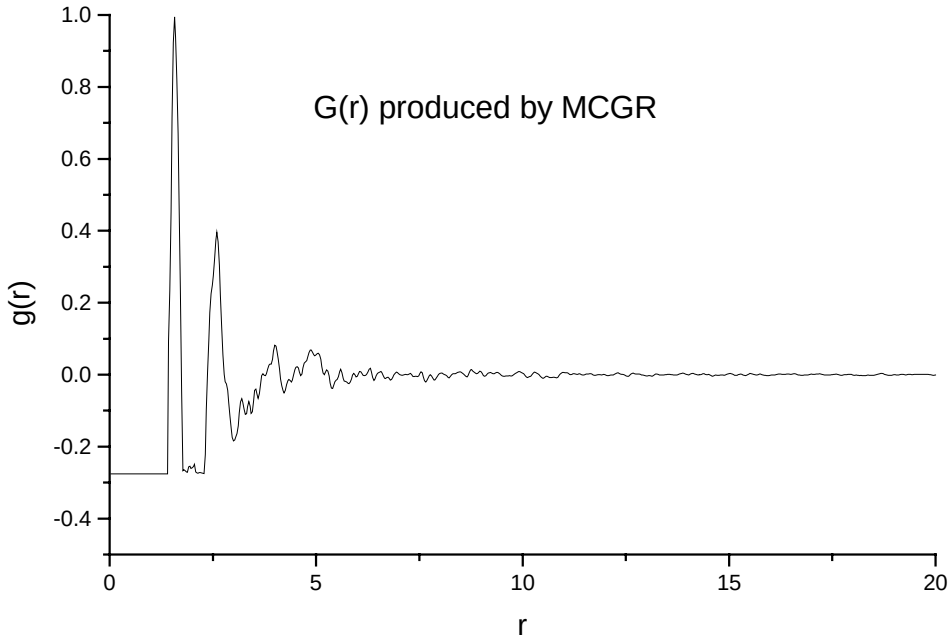
```
Graphics device/type (? to see list, default /NULL): /ws
File to plot (or RETURN to exit) > sio2_mcgr.out
Input file contains   2 groups of plots:
  Group   1 contains   1 plots of  3 curves
  Group   2 contains   1 plots of  1 curves
Plot which group (enter 0 to exit) ? 1
```

This will plot your original *F(Q)* together with the fitted *F(Q)* and a fitted background. The fit should be quite reasonable. If only a poor fit has been achieved at this stage then either there is something wrong with the parameters used in MCGR or the data contain non-negligible systematic errors. This should be checked very carefully before proceeding. Running RMCA with wrong parameters or poor data is simply a waste of time. The level of fit achieved with MCGR is a measure of the best fit that might be achieved with RMCA.



Now look at *G(r)* by continuing with **`rmcplot`**.

```
Change limits ? (T/F) > f
Input file contains   2 groups of plots:
  Group   1 contains   1 plots of  3 curves
  Group   2 contains   1 plots of  1 curves
Plot which group (enter 0 to exit) ? 2
```



9. The Si-O closest approach of 1.4 Å chosen at step **4** is obviously satisfactory. The first peak 'ends' at 1.8 Å so Si should be 4 fold coordinated to O within this distance. It should be checked that the area is reasonably consistent with such a coordination otherwise there will be an inconsistency between the constraint used in RMCA and the data which will inevitably lead to a poor fit to the data. Run **`coord`**.

```
File containing g(r): sio2_mcgr.g
Density: 0.0657
Add constant [0.]: 0.2759
Concentration [1.]: 0.6667
Coefficient [1.]: 0.1069
Write a file with cumulative coordination (y/n)? n
Select min/max r-value G=Graphically or N=Numerically: n
Minimum r-value [0.]:
Maximum r-value [Max-r]: 1.8
Coordination=    3.63
```

The result from the example shown is 3.63. This is somewhat low because the data are not yet fitted very well at high *Q*: in fact MCGR should be run for longer. The second peak in *G(r)* starts at 2.35 Å, which is then the O-O closest approach. The Si-Si peak is not obvious (it can be seen in X-ray data but starts slightly after the O-O peak at almost 2.5 Å. Constraint distances of 2.9-3.5 Å will be used for the Si network step **7**) since this range covers both the Si-Si peak and twice the Si-O peak. The latter is necessary because O atoms will be added at the centre of Si-Si bonds: the Si-O bond length will then be in the required range.

10. Now look at the data file for the proper RMC model. This is contained in **`sio2.dat`**.

```
SiO2 network (Fit to neutron G(r))
0.0657          ! number density
2.35 1.40
     2.35       ! cut offs
0.05 0.05       ! maximum move
2.855993E-2     ! r spacing
.false.         ! whether to use moveout option
0               ! number of configurations to collect
9000            ! step for printing
60 10           ! Time limit, step for saving
1 0 0 0         ! No. of g(r), neutron, X-ray, EXAFS expts
sio2_mcgr.g
1 1000          ! Range of points used
0.              ! Constant
.0191 .1069
      .1499     ! Coefficients
0.03            ! Standard deviation
.false.         ! whether to renormalise
3               ! no. of coordination constraints
1 2 0.  1.8  4 1. 0.00001
2 1 0.  1.8  2 1. 0.00001
1 2 1.8 2.2 0 1. 0.0001
0               ! no. of average coordination constraints
.false.         ! whether to use a potential
```

- The choice of closest approach constraints has been described above. Maximum moves of 0.05 Å are set for each atom. When relatively tight coordination constraints are used the maximum move needs to be approximately the same as the *r* point spacing to ensure a reasonable number of accepted moves.
- Three coordination constraints are used. Each Si has 4 O neighbours and each O has two Si neighbours within 1.8 Å: each Si should have no O neighbours between 1.8 and 2.2 Å. The aim of the third constraint is to maintain the 'gap' between the first and second peaks in *G(r)*, which should clearly be there despite a small amount of noise in the experimental *G(r)*. This is a less important constraint so it has a smaller weighting (σ is larger).

11. The distribution of neighbours in the Si network can be checked using the program **neigh**.

```
NEIGHBOURS

Configuration         : si_net
Minimum bond lengths  : 0.0
Maximum bond lengths  : 3.5
Output file           : si_net.nei

Calculating neighbours ...

Stop - Program terminated.
```

Look at the file **si_net.nei**

```
 Calculation of neighbours in si_net


 No. of atom types = 1

 Minimum bond lengths =    0.000000E+00
 Maximum bond lengths =        3.500000

 Type 1 - Type 1 neighbours:

          1         2     2.000
          4        98    98.000

 Average coordination   3.94
 ---------------------------
```

When modelling this structure 'properly' we would now attempt to remove any remaining coordination defects. If there are very few they are unlikely to disappear just from running **rmca** for longer. One possible solution is to remove them using **delatom** based on information about the coordination of defects found from **neigh** above (in the example printed there are two 1-fold coordinated Si), and then to add them into back into the configuration randomly using **addrand** or near to under-coordinated atoms using **addatom**. **rmca** should then be run again and this procedure repeated as necessary. Note that it probably becomes a waste of time after about 99.5% correct coordination; you are unlikely to get an absolutely perfect network and anyway it is probably not perfect in reality either.

12. We now add O at the mid-points of Si-Si bonds to make the SiO_2 network. Use the program **midpt**.

```
ADD ATOMS AT BOND CENTRES

Configuration                : si_net
Maximum bond lengths         : 3.5
Bond               [From,To] : 1 1
Type of atom to add          : 2

No. of atoms added =         197
Config. size       =         297

Output file                  : sio_net.cfg
```

In the printed example three O too few have been added because of the 1-fold coordinated Si. Check the neighbour distribution using **neigh**. There are now 2 types of atom in the configuration so 3 bond lengths are required. Have a look at the configuration with **confplot** (see point 18).

```
NEIGHBOURS

Configuration         : sio_net
Minimum bond lengths  : 0.0 0.0 0.0
Maximum bond lengths  : 3.5 1.75 0.0
Output file           : sio_net.nei

Calculating neighbours ...
```

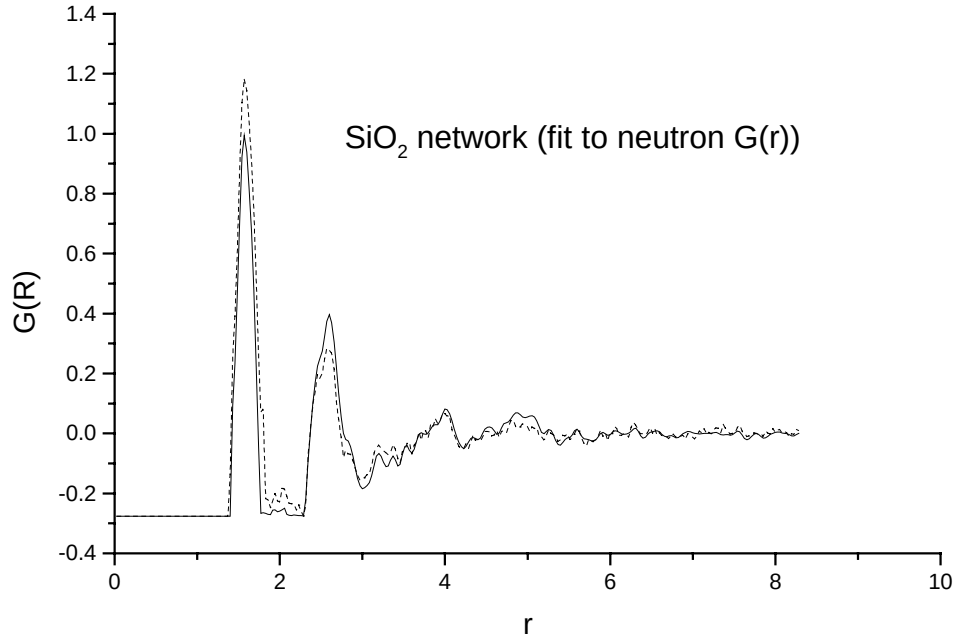
Look at the result in **sio_net.nei**.

```
 Calculation of neighbours in sio_net


 No. of atom types = 2

 Minimum  bond  lengths  =         0.000000E+00         0.000000E+00
0.000000E+00
 Maximum  bond  lengths  =            3.500000              1.750000
0.000000E+00

 Type 1 - Type 1 neighbours:

         1         2     2.000
         4        98    98.000

 Average coordination   3.94
 ---------------------------
 Type 1 - Type 2 neighbours:

         1         2     2.000
         4        98    98.000

 Average coordination   3.94
 ---------------------------
 Type 2 - Type 1 neighbours:
```

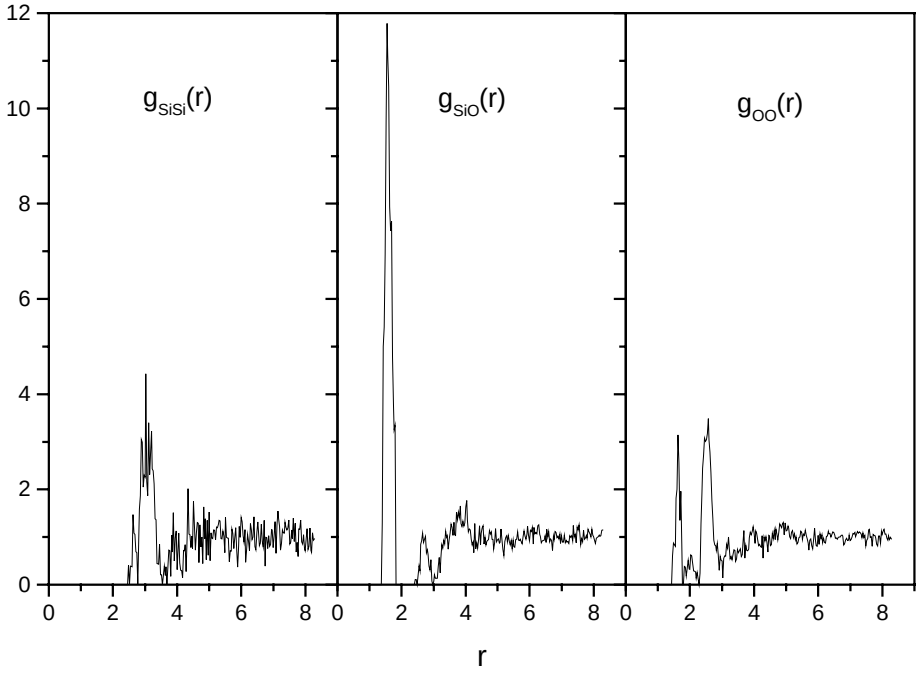
```
          2       197   100.000

 Average coordination   2.00
 ---------------------------
 Type 2 - Type 2 neighbours:

          0       197   100.000

 Average coordination    .00
 ---------------------------
```

13. Now we are ready to start RMC modelling. The RMCA data file has been discussed earlier. Copy the SiO_2 configuration (**`sio_net.cfg`**) to a file of the correct name (**`sio2.cfg`**). Run **`rmca`** and give as input file sio2 (no extension). This will take 60 minutes (It probably takes longer to get a good fit). Expected χ^2/nq ~ 8.5.

14. When the RMCA job has finished you can plot its output (RMC fit and the partial $g_{\alpha\beta}(r)$) with **`rmcplot`**.



Plot the partial $g_{\alpha\beta}(r)$.



The first peak in $g_{OO}(r)$ at 1.6 Å is from atoms which still do not satisfy the O-O closest approach constraint because RMCA has not in fact been run for long enough.

15. The resulting RMC model in **`sio2.cfg`** can be analysed with some standard programs. First look at the **`sio2.out`** file to determine the positions of the minima in $g_{\alpha\beta}(r)$. These occur at about 3.5, 1.85 and 2.85 Å. You can also have a look at the configuration with **confplot** (see point 18).

16. Look at the neighbour distribution using **`neigh`**. The Si-O 4-fold coordination has been maintained by the coordination constraints. O has approximately six O neighbours as would be expected for corner sharing tetrahedra. Si still have approximately 4 Si neighbours, but not exactly since the constraint used to create the original Si network is no longer enforced.

```
Configuration: sio2.cfg
Minimum bond lengths: 0.0 0.0 0.0
Maximum bond lengths: 3.5 1.85 2.85
Output file: sio2.nei
```

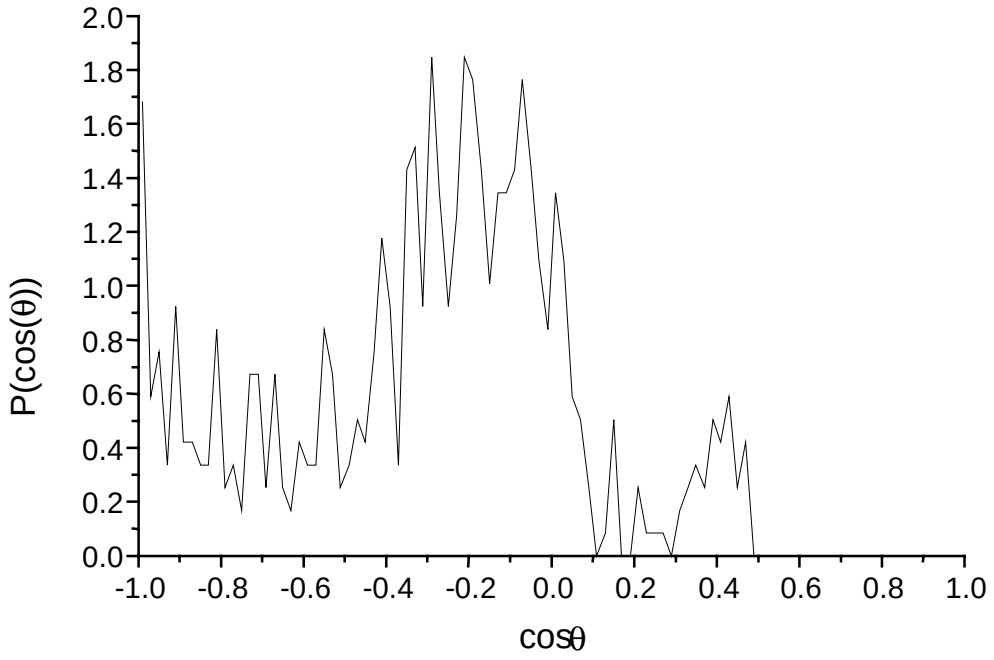
Look at the file **sio2.nei**:

```
Calculation of neighbours in sio2.cfg


 No. of atom types = 2

 Minimum  bond  lengths  =         0.000000E+00         0.000000E+00
0.000000E+00
 Maximum  bond  lengths  =            3.500000              1.850000
2.850000

 Type 1 - Type 1 neighbours:

          3         1     1.000
          4        90    90.000
          5         9     9.000

 Average coordination   4.08
 --------------------------
 Type 1 - Type 2 neighbours:

          2         1     1.000
          4        99    99.000

 Average coordination   3.98
 --------------------------
 Type 2 - Type 1 neighbours:

          2       199   100.000

 Average coordination   2.00
 --------------------------
 Type 2 - Type 2 neighbours:

          3         8     4.020
          4        50    25.126
          5        81    40.704
          6        45    22.613
          7        13     6.533
          8         2     1.005

 Average coordination   5.06
 --------------------------
```

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

```
Np. Of theta points > 100
No. of neighbours for bond ang (0 for all) > 0
Number of configurations > 1
Configuration file > sio2.cfg
Maximum r values > 3.5 1.85 2.85
Output file > sio2.tri
(A)ngle or (C)osine distribution [C] > c
```

For a two component system there are six different bond angle distributions 111, 121, 211, 212, 221 and 222. Use **extract** to extract the O-Si-O distribution then plot it with **show**.



The peak occurs close to the internal tetrahedral angle of 109^{o}.

18. Ball and stick pictures of the different configurations can also be produced with the program **confplot.exe**. See the manual for this program for instructions how to use it. Here are some examples:

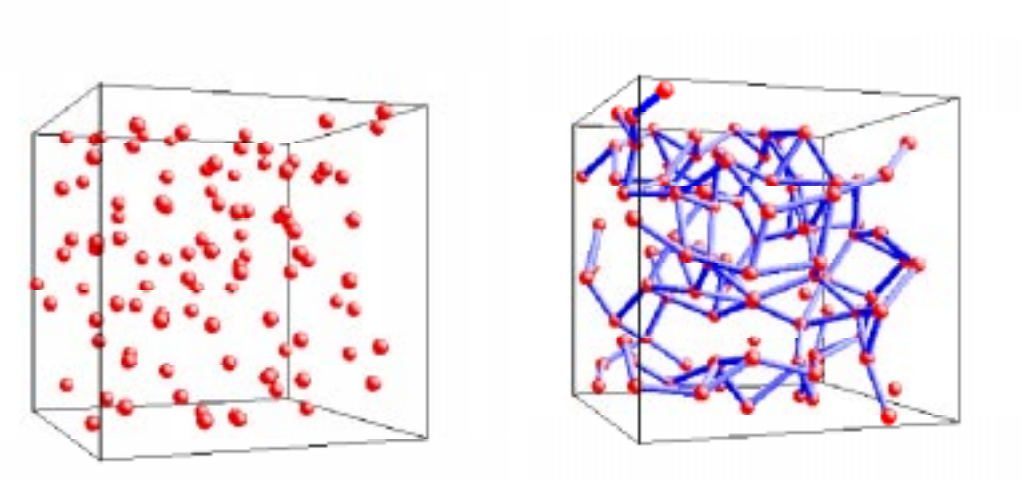


Figure 1. Left: **si_mov.cfg**, Right: **si_net.cfg.** Lines between Si-atoms indicate the tetrahedral structure.

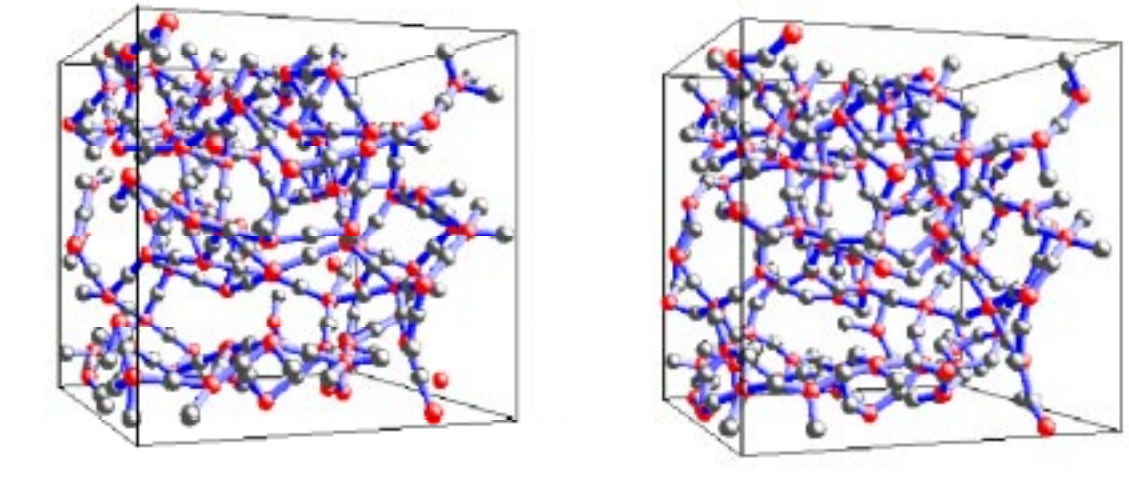


Figure 2. Left: **sio_net.cfg** Oxygen atoms are located half way between Si atoms. Right: **sio2.cfg;** the final RMC model.

19. There is also possible to run RMC by fitting to *F(Q)* instead of *G(r)*. Because of the limited size of the configuration however, we need to convolute the measured *F(Q)* in *r* space by a step function. This can be done with the program **convol** and the truncation distance to be used is the (minimum) half box length. This length can be found in the beginning of the configuration file. Run **convol**:

```
Input file          > sio2_neutron.fq
Truncation distance > 8.295106
Output file         > sio2_neutron.cfq
```

The convoluted structure factor is now in **sio2_neutron.cfq**. Copy the configuration **sio_net.cfg** to **sio2_fq.cfg** and use the control file **sio2_fq.dat** for a new run of **rmca**. Check the result as described above. The model used in this practical is a bit small in order to shorten the time needed to run RMC. For a better fit a larger model should be used. Expected χ^2/nq ~ 8.5.