Tutorial 4: Calculations for Pinkerton 2015

James Foadi
email j.foadi@bath.ac.uk

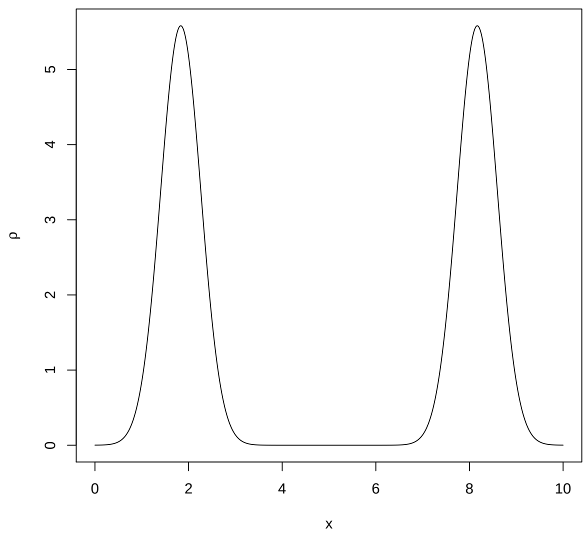
July, 2019

Prior to the publication of reference [**crn**] there have been at least two excellent introductions to 1D structural crystallography (see [**stoutjens**],[**pinkerton**]). Citation [**pinkerton**] by A. A. Pinkerton is an article in which a few important topics in structural crystallography are illustrated using a simple $P\bar{1}$ 1D structure made of two carbon atoms. In this tutorial we will reproduce Pinkerton's examples using tools from `crone`.

1 Pinkerton's structure

This structure is available as internal data "pinkerton2015". It is a $P\bar{1}$ structure made of two carbon atoms. The B factor is not given in reference [**pinkerton**] and, in line with the way B factors are generated in `crone`, it has been assigned value 13.333 $\AA^2$. Data are generated in the following code snippet.

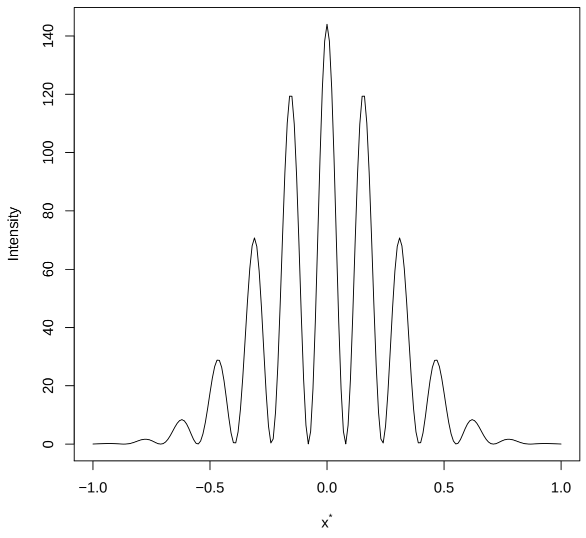
```
library(crone)
sdata <- load_structure("pinkerton2015")
rtmp <- structure_gauss(sdata,N=1000)
plot(rtmp$x,rtmp$rr,type="l",xlab="x",ylab=expression(rho))
```



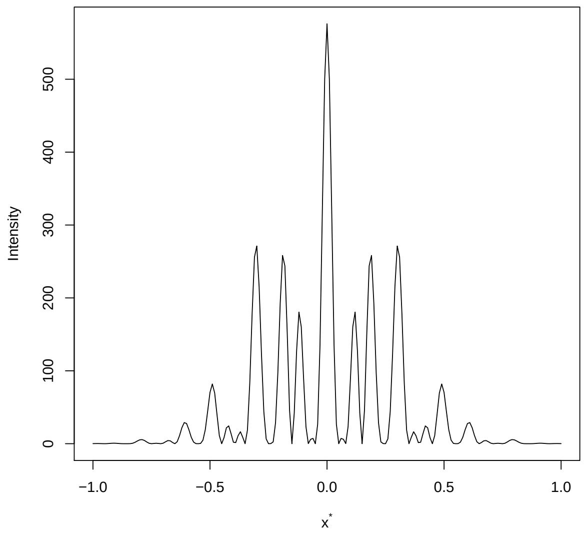
2 Indexing and cell's length determination

The unit cell length is unknown before diffraction data are collected. It can be calculated using the pattern created by the diffracted intensities. In 1D crystallography this means, essentially, the distance between the diffraction peaks.
In `crone` there exist a function to simulate 1D diffraction patterns starting from the 1D atomic structure. The function is called `diffraction`. It takes in the `sdata`-type list and the maximum resolution of the diffraction pattern in angstroms. It returns a named list with `xstar` as reciprocal space grid and `Imod` as diffracted intensities. One of the interesting and well-known features of a diffraction pattern is that the intensities are proportional to the square of the number of unit cells forming the crystal (equal to `Ncell`, with default value equal to 10). The following snippet demonstrates some of the possible diffraction patterns simulated with this function. More details are available in the documentation.

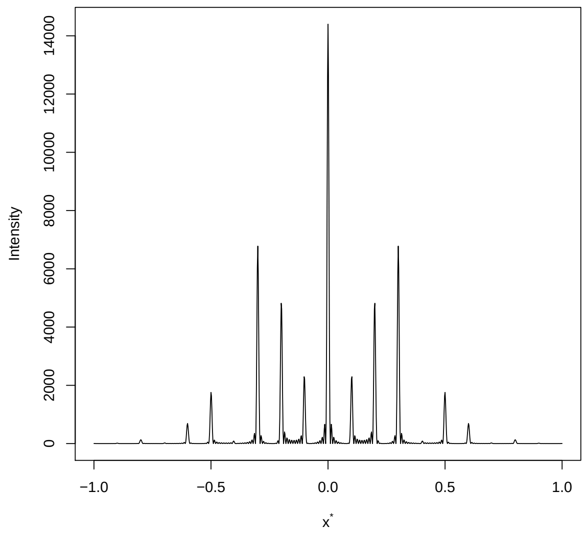
```r
# Max resolution 1 angstrom (D=1), crystal formed of only
# one unit cell (Ncell=1)
ltmp <- diffraction(sdata,D=1,Ncell=1)
plot(ltmp$xstar,ltmp$Imod,type="l",
     xlab=expression(paste("x"^"*")),ylab="Intensity")
```



```
# Max resolution 1 angstrom, crystal formed by
# two unit cells (Ncell=2)
ltmp <- diffraction(sdata,D=1,Ncell=2)
plot(ltmp$xstar,ltmp$Imod,type="l",
     xlab=expression(paste("x"^"*")),ylab="Intensity")
```



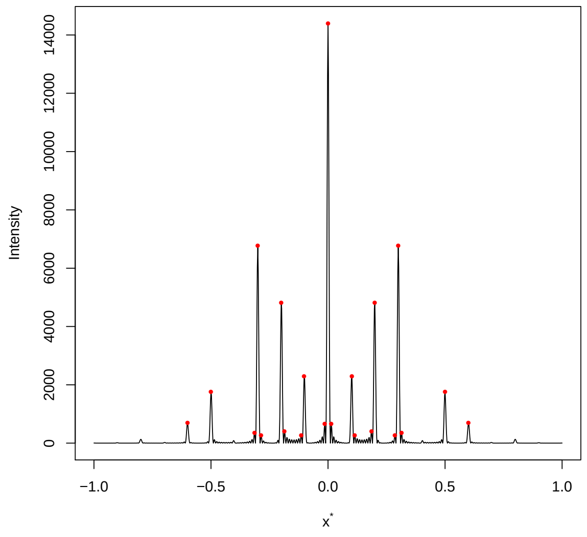
```r
# Max resolution 1 angstrom, crystal formed by
# 10 unit cells (Ncell=10, default value). The number of reciprocal
# space grid points is inreased to 1001 (n=500 -> 2*n+1=1001;
# default value is n=100 -> 2*n+1=201)
ltmp <- diffraction(sdata,D=1,n=500)
plot(ltmp$xstar,ltmp$Imod,type="l",
     xlab=expression(paste("x"^"*")),ylab="Intensity")
```



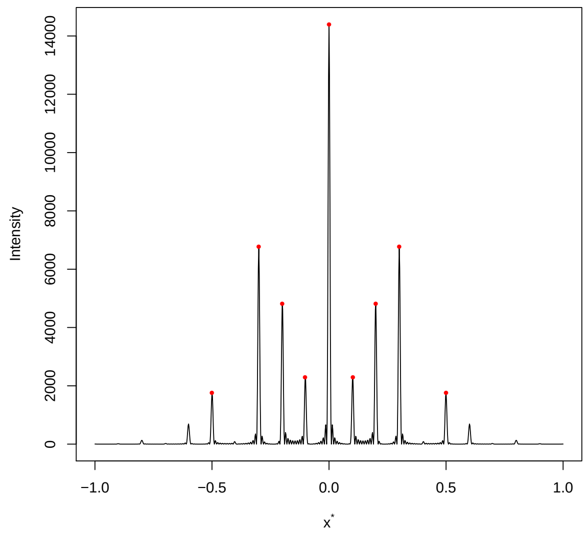
The interference effect for 10 unit cells is strong enough to make the diffraction pattern appear very similar to a diffraction pattern from an infinite crystal. It is also easy to see that the distance between adjacent peaks is around 0.1, which is the reciprocal of $1/a$, a being $10\mathring{A}$ for this structure. From the practical point of view, one of the first tasks to be carried out in structural crystallography is the determination of the diffraction spots' position (in 1D the spots are the peaks' maxima) and the accurate calculation of the unit cell length.

In order to find the maxima's position we can use the function `local_maxima` and, among all peaks found, filter those higher than a given threshold. The value of this threshold is obviously key to the determination of the appropriate diffraction geometry and correct cell's length.

```
# Find all peaks
idx <- local_maxima(ltmp$Imod)

# Mean and standard deviation of electron density
M <- mean(ltmp$Imod)
S <- sd(ltmp$Imod)

# Threshold (1st attempt)
Thr <- M + 0*S
Thr

## [1] 236.5709

# Peaks (spots) selection
idx <- local_maxima(ltmp$Imod)
jdx <- which(ltmp$Imod[idx] > Thr)
idx <- idx[jdx]  # New index of selected peaks
length(idx)

## [1] 21

# Too many peaks (some should be considered as noise)
plot(ltmp$xstar,ltmp$Imod,type="l",
     xlab=expression(paste("x"^"*")),ylab="Intensity")
points(ltmp$xstar[idx],ltmp$Imod[idx],pch=16,cex=0.65,col=2)
```



```
# Threshold (2nd attempt)
Thr <- M + 1*S
Thr

## [1] 1313.187

# Peaks (spots) selection
idx <- local_maxima(ltmp$Imod)
jdx <- which(ltmp$Imod[idx] > Thr)
idx <- idx[jdx]  # New index of selected peaks
length(idx)

## [1] 9

# Some peaks have been missed
plot(ltmp$xstar,ltmp$Imod,type="l",
     xlab=expression(paste("x"^"*")),ylab="Intensity")
points(ltmp$xstar[idx],ltmp$Imod[idx],pch=16,cex=0.65,col=2)
```

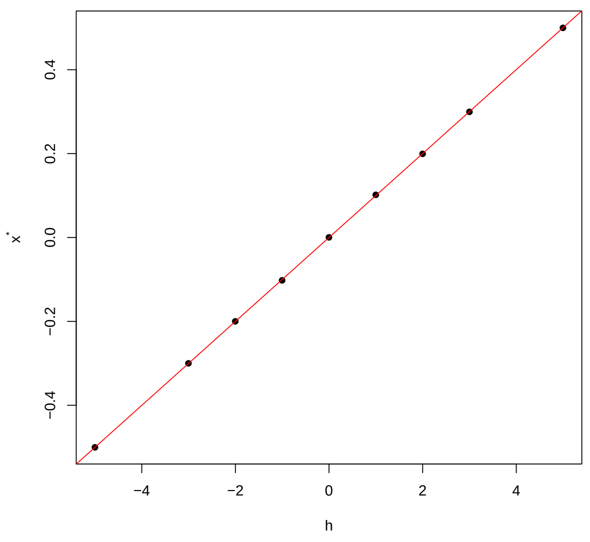


When the threshold is increased, some of the weaker diffreaction maxima are missed. When the number of maxima found is sufficient for a correct indexing, then one can proceed to cell-length determination. In this specific case we found 9 peaks; one is the peak corresponding to $h = 0$. The other 8 peaks are symmetric with respect to $x^* = 0$; half of them correspond to negative Miller indices and half to positive Miller indices. There is also a gap (see diffraction picture) between the third and the fifth diffraction maximum because the fourth is a weak diffraction spot and has a value smaller than the threshold adopted. Therefore we have the following indices $-5, -3, -2, -1, 0, 1, 2, 3, 5$ corresponding to the x^* values of the centre of the peaks in reciprocal space. The relation is the one defining the crystallographic resolution $d = 1/d^*$ of a diffraction spot; for 1D crystallography,

$$a^*h = d^*$$

The above relation is graphically described by a straight line passing through the origin. This line can be found using the least squares procedure from which the slope a^* is extracted. The unit cell's length is simply $a = 1/a^*$.

```
# Points for the plot
x <- c(-5,-3,-2,-1,0,1,2,3,5)
y <- ltmp$xstar[idx]
plot(x,y,pch=16,xlab=expression(h),
     ylab=expression(paste("x"^"*")))

# Least squares
model <- lm(y ~ 0+x)  # Origin included
smdl <- summary(model)
smdl

##
## Call:
## lm(formula = y ~ 0 + x)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -0.0019487 -0.0001538  0.0000000  0.0001538  0.0019487
##
## Coefficients:
##    Estimate Std. Error t value Pr(>|t|)
## x 0.1000513  0.0001118   895.2   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.0009871 on 8 degrees of freedom
## Multiple R-squared:      1,Adjusted R-squared:      1
## F-statistic: 8.013e+05 on 1 and 8 DF,  p-value: < 2.2e-16

# Fit
abline(model,col=2)
```



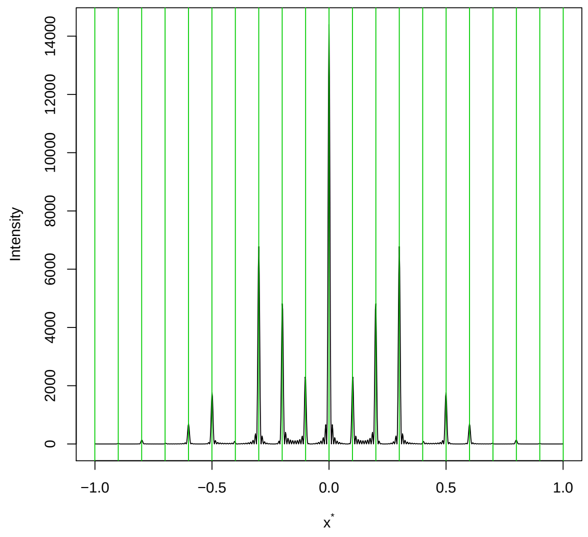
```
# Unit cell length (approximately 10)
a = 1/smdl$coefficients[1]
a

## [1] 9.994874
```

From the procedure carried out above it is quite clear that the spots selection is an essential and key part of cell determination. The threshold adopted (and thus the inclusion of noise rather than the signal) is important for both the correct indexing and unit cell determination.

Once the unit cell's length is found, the reciprocal lattice can be re-created and the intensities at the lattice points can be extracted. This process is normally known as *data integration*. We can create the reciprocal lattice as repetition of equally-spaced points of width $a^* = 1/a$, where $a \approx 9.995$.

```
# Lattice
hidx <- -10:10
astar <- smdl$coefficients[1]
L <- astar * hidx

# Lattice overlapped to diffraction pattern
```

```
plot(ltmp$xstar,ltmp$Imod,type="l",
     xlab=expression(paste("x"^"*")),ylab="Intensity")
abline(v=L,col=3)
```

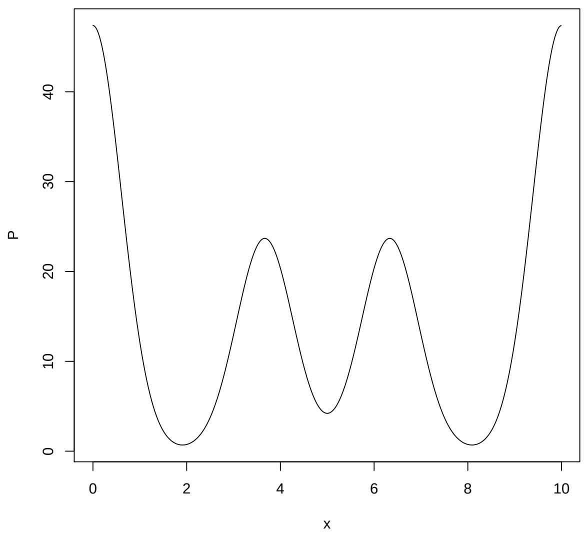


The integrated intensities can be found as areas under the curves centred at the specified lattice points. This operation can be carried out in many possible ways. For example, one can simply decide a symmetric interval centred at the specific lattice point and add up all intensities of the grid points included within the interval. Or, a gaussian curve centred at the lattice point could be fitted to the local set of diffracted intensities and the integrated intensity calculated as area of the gaussian curve found. Such methods are certainly here implementable, but they require a considerable amount of additional code and are, thus, skipped.

3 The Patterson function

The *Patterson function* is calculated as Fourier synthesis with structure factors having squared moduli, $|F_h|^2$, as amplitudes and zeros as phases, $\varphi_h = 0$. The corresponding density is symmetric with respect to the origin, has a huge peak at the origin and the peaks' position is quantitatively equivalent to an inter-atomic distance, i.e. the distance between two atomic peaks in the unit cell. Let's explore the Patterson function for the pinkerton2015 structure.

```
# Miller indices
hidx <- 0:20

# Structure factors
ftmp <- strufac(hidx,sdata)

# What's in the structure factor list
names(ftmp)

## [1] "Fmod" "Fpha"

# Amplitudes and phases for the Patterson
Pmod <- ftmp$Fmod^2
Ppha <- rep(0,times=length(hidx))

# Patterson as Fourier synthesis
rtmp <- fousynth(sdata$a,Pmod,Ppha,hidx,N=1000)
plot(rtmp$x,rtmp$rr,type="l",xlab="x",ylab="P")
```



The huge peak at the origin is self-evident. Let's next measure the position of the smaller peaks; this should correspond to the two distances between the two carbons, i.e. between the carbon and its symmetry-equivalent on one side and the other with respect to the origin). We can carry out the calculation using the function to find maxima.

```
# Generate the symmetry-equivalent
sdata2 <- expand_to_cell(sdata)
sdata2$x0

## [1] 1.833 8.167

# Smaller inter-atomic distance
sdata2$x0[1]-(-sdata2$x0[1])

## [1] 3.666

# Larger inter-atomic distance
sdata2$x0[2]-sdata2$x0[1]

## [1] 6.334

# Peaks position
idx <- local_maxima(rtmp$rr)
rtmp$x[idx]

## [1] 0.00 3.67 6.33
```

Thus, the position of the second and third Patterson peak coincide with the two inter-atomic distances previously calculated.

4 Structure factors and Fourier synthesis

In reference [**pinkerton**] two sections are devoted to the computation of the structure factors and the Fourier synthesis. With `crone` these tasks are accompished by the two functions `strufac` and `fousynth` which we have used previously.

5 Direct Methods

Direct methods are defined as a certain number of probabilistic and statistical procedures to find the phases of the Fourier synthesis, starting from its amplitudes. This is, obviously, equivalent to calculate the electron density and it is, thus, one of the methods to solve crystallographic structures, i.e. to determine the atomic positions in the unit cell.

5.1 Choice of cell origin and value of phases

For the $P\bar{1}$ symmetry group the origin can be placed either at $x = 0$ or at $x = a/2$. The choice has ripercussions on the values that the phase of certain reflections can take. For the structure we are using right now, for instance, the phases change from 0^o to 180^o for certain reflections, but not for others.

```
# Structure factors for structure with origin at x=0
hidx = 1:10
ftmp <- strufac(hidx=hidx,sdata=sdata)

# Change of origin at x=a/2=5
sdata2 <- sdata
sdata2$x0 <- sdata$x0-5+10   # Shifted back inside cell
sdata2$x0

## [1] 6.833

# Structure factors for structure with origin at x=a/2
ftmp2 <- strufac(hidx=hidx,sdata=sdata2)
```

While the position of the two carbon atoms with the origin at 0 is 1.833 and 8.167, the position of the same atoms when the origin is placed at $a/2 = 5$ is 6.833 and 3.167. A comparison of two sets of phases presents an interesting feature.

```
# Phases for structure with origin at x=0
ftmp$Fpha

##  [1]   0 180 180 180   0   0 180 180 180   0

# Phases for structure with origin at x=a/2
ftmp2$Fpha

##  [1] 180 180   0 180 180   0   0 180   0   0
```

For some of the reflections ($h = 1, 3, 5, 7, 9$) the phase changes, while for others ($h = 2, 4, 6, 8, 10$) the phase remains the same. This suggests that in this case fixing the phase of an odd reflection ($h = 1, 3, 5, 7, 9$) is equivalent to fixing the origin in the cell. For instance, if $\varphi_1 = 0$ the origin is fixed at $x = 0$, while if $\varphi_1 = 180^o$, the origin is placed at $x = a/2$.

5.2 Normalised structure factors

One of the notable features of a structure factor F_h is that its amplitude $|F_h|$ decreases with increasing value of the Miller index h. This is a useful feature for the determination of the B-factors, but it is not a desirable feature for the initial determination of the atomic positions as atoms with smaller atomic number Z or higher B-factor B are less detectable than the rest of the atoms. The goal of direct methods is to determine atomic positions; therefore such methods work effectively with a density showing pronunced and non-broadening peaks. Such a density can be obtained as Fourier synthesis of structure factors E_h having same phases as the standard structure factors F_h, but different amplitude according to the following expression,

$$E_h \equiv \frac{F_h}{\sqrt{\sum_{j=1}^N f_j^2(h)}} \qquad \Rightarrow \qquad |E_h| \equiv \frac{|F_h|}{\sqrt{\sum_{j=1}^N f_j^2(h)}}$$

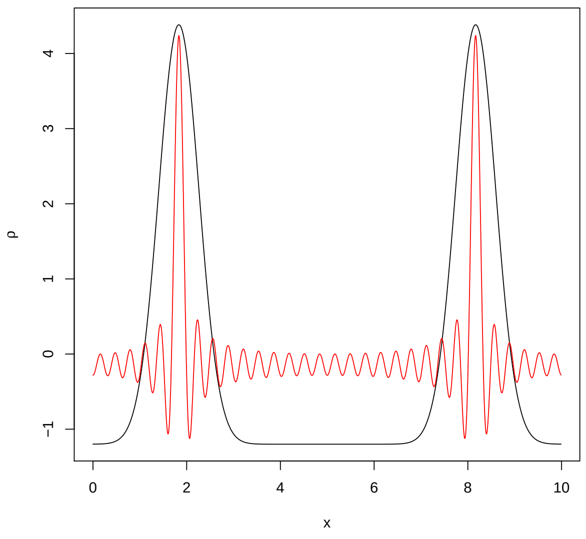
These modified structure factors are called *normalised structure factors*. In the following we carry out a comparison of amplitudes for standard and normalised structure factors. The scattering factors, $f_j(h)$, are calculated using the `crone` function `scafac`. As this particular structure has two identical carbon atoms, the bottom part of the above expression is,

$$\sqrt{\sum_{j=1}^N f_j^2(h)} = \sqrt{f_C^2 + f_C^2} = \sqrt{2} f_C$$

```
# Standard structure factors
hidx <- 1:30
ftmp <- strufac(hidx=hidx,sdata=sdata)
FF <- ftmp$Fmod

# Vectors of sums of scattering factors (this structure is
# made of two carbon atoms)
ff <- sqrt(2)*scafac(h=hidx,sdata$a,sdata$Z,sdata$occ,sdata$B)

# Normalised structure factors
EE <- FF/ff

# Display
for (i in hidx) {
  line <- sprintf("%8.3f  %8.3f\n",FF[i],EE[i])
  cat(line)
}

##    4.709     0.575
##    6.941     0.946
##    8.234     1.345
##    0.707     0.149
##    4.192     1.224
##    2.631     1.145
##    0.419     0.292
##    1.150     1.383
##    0.374     0.833
##    0.159     0.705
##    0.148     1.407
##    0.020     0.440
##    0.019     1.048
##    0.009     1.294
##    0.000     0.004
##    0.001     1.290
##    0.000     1.054
##    0.000     0.432
##    0.000     1.406
##    0.000     0.712
##    0.000     0.826
##    0.000     1.385
##    0.000     0.301
##    0.000     1.140
##    0.000     1.228
##    0.000     0.140
##    0.000     1.343
##    0.000     0.952
##    0.000     0.567
##    0.000     1.414
```

There's an evident change in value and, more specifically, there is no decay with increasing h. But perhaps the best way to understand normalised structure factors is to compare the density of Fourier synthesis using the same phases, but $|F_h|$ and $|E_h|$ in turn.

```
# Density for standard structure factors
rtmp <- fousynth(a=sdata$a,Fmod=FF,Fpha=ftmp$Fpha,
                 hidx=hidx,N=1000)

# Density for normalised structure factors
ntmp <- fousynth(a=sdata$a,Fmod=EE,Fpha=ftmp$Fpha,
                 hidx=hidx,N=1000)

# Graphical comparison
plot(rtmp$x,rtmp$rr,type="l",xlab="x",ylab=expression(rho))
points(ntmp$x,ntmp$rr,type="l",col=2)
```



From the plot one can see clearly that the density corresponding to normalised structure factors has narrower peaks than the density corresponding to standard structure factors. This more pronounced *atomicity* of the normalised-structure-factors density is key to an effective application of direct methods.

To conclude this subsection, it is worth stressing that for equal atoms, as for the case we are treating, the expression of the normalised structure factors is very simple,

$$E_h = \frac{1}{\sqrt{N}} \sum_{j=1}^{N} \exp\left(2\pi i h \frac{x_j}{a}\right)$$

with N the number of atoms in the cell. The formula above highlights the importance of the atomic positions, x_j, over the atomic density as the last one is represented by the constant quantity $1/\sqrt{N}$. For this reason it is often remarked that a density structure calculated using normalised structure factors is equivalent to the density of a structure made of point-like atoms.

5.3 Signs from Σ_1 and Σ_2 relationships

Direct methods are, essentially, probabilistic methods. The probability of the value of certain combinations of phases can be calculated under very simple and general assumptions, notably that the electron density is positive and that high density is only found around atomic centres, while everywhere else the density is very low or zero (atomicity). In the case of the structure under investigation, phases are restricted to 0^o and 180^o so that we can talk of the "sign" of the structure factors ($+$ or $-$). For centrosymmetric structures or, more specifically, for the 1D and $P\bar{1}$ structure we are studying, the two important probability relationships found in direct methods are the Σ_1 and Σ_2 relationship:

$$\Sigma_1 \text{ relationship}: \quad P_+ = \frac{1}{2} + \frac{1}{2}\tanh\left(\frac{|E_h E_h E_{2h}|}{\sqrt{2}}\right)$$

$$\Sigma_2 \text{ relationship}: \quad P_+ = \frac{1}{2} + \frac{1}{2}\tanh\left(\frac{|E_h E_k E_{h-k}|}{\sqrt{2}}\right)$$

These two relationships return the probability that the specific product considered ($E_h E_h E_{2h}$ for Σ_1 and $E_h E_k E_{h-k}$ for Σ_2) has a positive sign. In the following snippet we calculate some of these probabilities, considering $h, k, h-k$ varying between 1 and 12, as reported in reference [**pinkerton**].

```
# Normalised structure factors used
for (h in 1:12) {
  line <- sprintf("%5d   %10.3f\n",h,EE[h])
  cat(line)
}

##     1        0.575
##     2        0.946
##     3        1.345
##     4        0.149
##     5        1.224
##     6        1.145
##     7        0.292
##     8        1.383
##     9        0.833
##    10        0.705
##    11        1.407
##    12        0.440

# Sigma_1 relationships
hMat <- matrix(c(1:6,1:6,2,4,6,8,10,12),nrow=6,ncol=3)
colnames(hMat) <- c("h","h","2h")
PS1plus <- rep(0,length=6)
for (i in 1:6) {
  PS1plus[i] <- 0.5+0.5*tanh(EE[hMat[i,1]]*
                EE[hMat[i,2]]*EE[hMat[i,3]]/sqrt(2))
}
S1_table <- cbind(hMat,PS1plus)
idx <- order(S1_table[,4],decreasing=TRUE)
print(S1_table[idx,])

##      h h 2h   PS1plus
## [1,] 3 3  6 0.9493537
## [2,] 5 5 10 0.8164902
## [3,] 6 6 12 0.6935482
## [4,] 1 1  2 0.6089737
## [5,] 2 2  4 0.5469912
## [6,] 4 4  8 0.5108529
```

There are two large P_+ probabilities among the 6 Σ_1 relationships; we can use these together with the Σ_2 relationships, to phase the structure.

```
# Sigma_2 relationships (66 found!)
hMat <- matrix(ncol=3)
for (h in 1:11) {
  for (k in (h+1):12) {
    ll <- h-k
    if (ll >= -12 & ll <= 12) {
      hMat <- rbind(hMat,matrix(c(h,k,ll),nrow=1))
    }
  }
}
hMat <- hMat[-1,]
```

```
colnames(hMat) <- c("h","k","h-k")
PS2plus <- rep(0,length=length(hMat[,1]))
for (i in 1:length(hMat[,1])) {
  PS2plus[i] <- 0.5+0.5*tanh(EE[abs(hMat[i,1])]*
              EE[abs(hMat[i,2])]*EE[abs(hMat[i,3])]/sqrt(2))
}
S2_table <- cbind(hMat,PS2plus)
idx <- order(S2_table[,4],decreasing=TRUE)
print(S2_table[idx,])
```

```
##          h  k h-k   PS2plus
##  [1,]    3 11  -8 0.9758972
##  [2,]    8 11  -3 0.9758972
##  [3,]    5  8  -3 0.9615844
##  [4,]    3  8  -5 0.9615844
##  [5,]    3  6  -3 0.9493537
##  [6,]    5 11  -6 0.9420614
##  [7,]    6 11  -5 0.9420614
##  [8,]    2  5  -3 0.9004808
##  [9,]    3  5  -2 0.9004808
## [10,]    2  8  -6 0.8926617
## [11,]    6  8  -2 0.8926617
## [12,]    3  9  -6 0.8600532
## [13,]    6  9  -3 0.8600532
## [14,]    2 11  -9 0.8275417
## [15,]    9 11  -2 0.8275417
## [16,]    5 10  -5 0.8164902
## [17,]    2 10  -8 0.7863718
## [18,]    8 10  -2 0.7863718
## [19,]    1  6  -5 0.7578214
## [20,]    5  6  -1 0.7578214
## [21,]    1  3  -2 0.7379951
## [22,]    2  3  -1 0.7379951
## [23,]    1  9  -8 0.7186811
## [24,]    8  9  -1 0.7186811
## [25,]    6 12  -6 0.6935482
## [26,]    1 11 -10 0.6913958
## [27,]   10 11  -1 0.6913958
## [28,]    3 12  -9 0.6678048
## [29,]    9 12  -3 0.6678048
## [30,]    1 12 -11 0.6234514
## [31,] 11 12  -1 0.6234514
## [32,]  2  7  -5 0.6172966
## [33,]  5  7  -2 0.6172966
## [34,]  1 10  -9 0.6172452
## [35,]  9 10  -1 0.6172452
## [36,]  1  2  -1 0.6089737
## [37,]  2 12 -10 0.6022951
## [38,] 10 12  -2 0.6022951
## [39,]  3 10  -7 0.5966191
## [40,]  7 10  -3 0.5966191
## [41,]  7  8  -1 0.5814245
## [42,]  1  8  -7 0.5814245
## [43,]  2  9  -7 0.5806702
## [44,]  7  9  -2 0.5806702
## [45,]  1  7  -6 0.5676199
## [46,]  6  7  -1 0.5676199
## [47,]  2  6  -4 0.5568153
## [48,]  4  6  -2 0.5568153
## [49,]  5 12  -7 0.5554217
## [50,]  7 12  -5 0.5554217
## [51,]  4  9  -5 0.5535335
## [52,]  5  9  -4 0.5535335
## [53,]  2  4  -2 0.5469912
## [54,]  6 10  -4 0.5424016
## [55,]  4 10  -6 0.5424016
## [56,]  1  4  -3 0.5406943
## [57,]  3  4  -1 0.5406943
## [58,]  1  5  -4 0.5370400
## [59,]  4  5  -1 0.5370400
## [60,]  4 12  -8 0.5320379
## [61,]  8 12  -4 0.5320379
## [62,]  4 11  -7 0.5216273
## [63,]  7 11  -4 0.5216273
## [64,]  3  7  -4 0.5206824
## [65,]  4  7  -3 0.5206824
## [66,]  4  8  -4 0.5108529
```

The values in the above table are close but not identical to the values presented in reference [**pinkerton**] probably due to round off errors in computing `tanh` and because the normalised structure factors are slightly different from those in the reference. Some of the triplet combinations return the same value of P_+ because the indices are the same, only appearing in different order. We have, though, a sufficient number of Σ_1 and Σ_2 combinations to attempt phasing. At this point we could follow exactly the choice made in the reference and proceed with phasing. We will, instead, try a different avenue. The important step in direct methods is to work initially with large normalised structure factors. In this specific case these correspond to Miller indices 3,5,6,8,11. To fix the origin we need to pick (see earlier discussion) a reflection with odd h and assign it positive sign. Due to the two top lines of the Σ_1 relationships, the best choice falls on either $h = 3$ or $h = 5$; let's pick $h = 5$. Sign assignment for the Σ_1 relationships results, then, as follows:

$$\Sigma_1 : \qquad \begin{array}{ccc} 3 & 3 & 6^+ \\ 5^+ & 5^+ & 10^+ \end{array}$$

With these choice, the Σ_2 set of strongest relationships (with $P_+ > 0.9$) are:

$$\Sigma_2 : \qquad \begin{array}{ccc} 3 & 11 & 8 \\ 5^+ & 8 & 3 \\ 5^+ & 11 & 6^+ \\ 2 & 5^+ & 3 \end{array}$$

In the above relations, a sign is assigned only when it is the only possible choice. For example, the second relations of the Σ_1 group has the signs product $(+)(+)(?) = +$, therefore the sign of E_{10} must be positive because the sign of the full triplet is positive (high probability of being positive). For the first relation we follow a similar reasoning; whether the sign of E_3 is positive or negative, the sign of their product is always positive so that the sign of E_6 must also be positive. All these signs are reported in the Σ_2 list of relationships. In the third relationship we are forced to accept the sign of E_{11} as positive. The updated set of relationships is, thus,

$$\Sigma_1 : \qquad \begin{array}{ccc} 3 & 3 & 6^+ \\ 5^+ & 5^+ & 10^+ \end{array}$$

$$\Sigma_2 : \qquad \begin{array}{ccc} 3 & 11^+ & 8 \\ 5^+ & 8 & 3 \\ 5^+ & 11^+ & 6^+ \\ 2 & 5^+ & 3 \end{array}$$

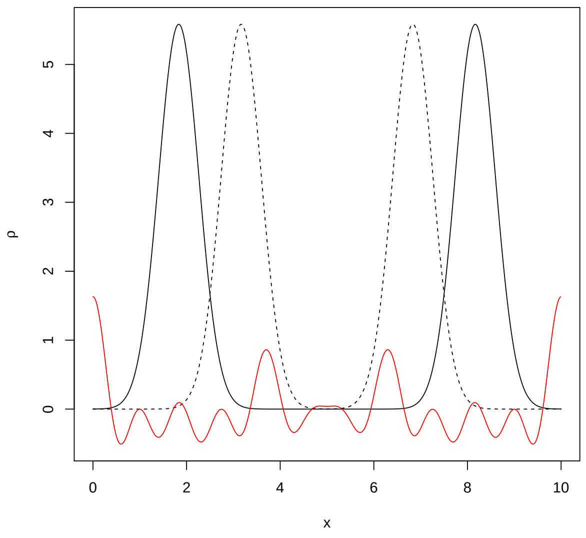
We face now two choices, according to whether the sign "+" or "-" is assigned to E_3. Let's start with the positive sign. The completed relationships are, in this case,

$$\Sigma_1 : \qquad \begin{array}{ccc} 3^+ & 3^+ & 6^+ \\ 5^+ & 5^+ & 10^+ \end{array}$$

$$\Sigma_2 : \qquad \begin{array}{rrr} 3^+ & 11^+ & 8^+ \\ 5^+ & 8^+ & 3^+ \\ 5^+ & 11^+ & 6^+ \\ 2^+ & 5^+ & 3^+ \end{array}$$

What density corresponds to this choice? Remember that we only have full values for the normalised factors with indices $h = 2, 3, 5, 6, 8, 10, 11$.

```
# Availabl set of known reflections
hidx <- c(2,3,5,6,8,10,11)
Fmod <- EE[hidx]
Fpha <- rep(0,length=length(Fmod))  # All signs are +
rtmp_test <- fousynth(sdata$a,Fmod=Fmod,
                      Fpha=Fpha,hidx=hidx,N=1000)

# Comparison: some peaks should match the two peaks for the
# structure with the origin at x=0 (rtmp) or the one with the
# origin at x=a/2 (rtmp2)
rtmp <- structure_gauss(sdata=sdata,N=1000)
rtmp2 <- structure_gauss(sdata=sdata2,N=1000)
ym <- min(rtmp$rr,rtmp2$rr,rtmp_test$rr)
yM <- max(rtmp$rr,rtmp2$rr,rtmp_test$rr)
plot(rtmp$x,rtmp$rr,type="l",xlab="x",ylab=expression(rho),
     ylim=c(ym,yM))
points(rtmp2$x,rtmp2$rr,type="l",lty=2)
points(rtmp_test$x,rtmp_test$rr,type="l",col=2)
```



Let's now consider the density corresponding to the other choice (E_3 negative). The relationships become, in this case,

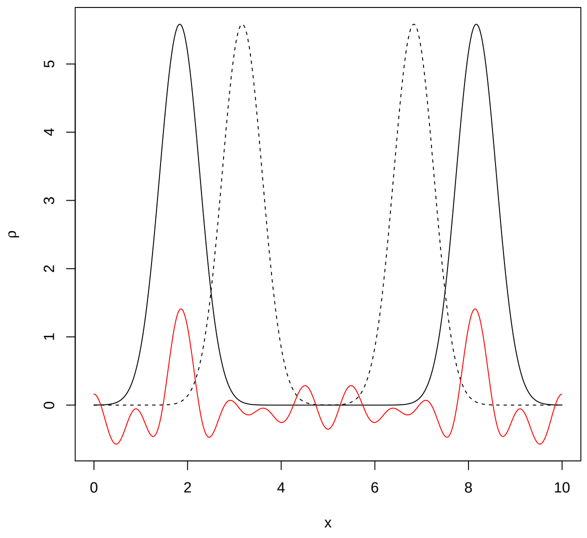
$$\begin{array}{lccc}
\Sigma_1 : & 3^- & 3^- & 6^+ \\
 & 5^+ & 5^+ & 10^+
\end{array}$$

$$\begin{array}{lccc}
\Sigma_2 : & 3^- & 11^+ & 8^- \\
 & 5^+ & 8^- & 3^- \\
 & 5^+ & 11^+ & 6^+ \\
 & 2^- & 5^+ & 3^-
\end{array}$$

The corresponding density is:

```
# Available set of known reflections
hidx <- c(2,3,5,6,8,10,11)
Fmod <- EE[hidx]
Fpha <- c(180,180,0,0,180,0,0)  # + or - signs
rtmp_test <- fousynth(sdata$a,Fmod=Fmod,
                      Fpha=Fpha,hidx=hidx,N=1000)

# Comparison
rtmp <- structure_gauss(sdata=sdata,N=1000)
rtmp2 <- structure_gauss(sdata=sdata2,N=1000)
ym <- min(rtmp$rr,rtmp2$rr,rtmp_test$rr)
```

```
yM <- max(rtmp$rr,rtmp2$rr,rtmp_test$rr)
plot(rtmp$x,rtmp$rr,type="l",xlab="x",ylab=expression(rho),
     ylim=c(ym,yM))
points(rtmp2$x,rtmp2$rr,type="l",lty=2)
points(rtmp_test$x,rtmp_test$rr,type="l",col=2)
```



The second choice quite obviously delivers a density map for normalised structure factors with peaks at the correct locations. The other solution (with 3 peaks and all phases equal to 0) is known as *uranium solution* in direct methods and it does not correspond to any real solution. Only further stereo-chemical interpretation of the peaks found and subsequent refinement can insure, in general and when multiple solutions are present, which of the solutions is acceptable.

6 Refinement

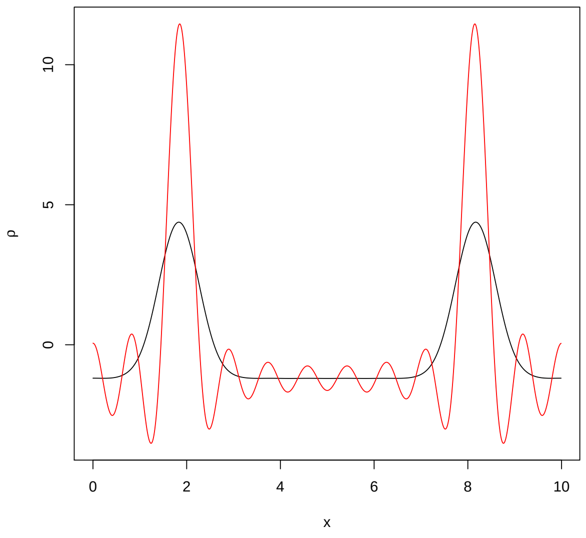
With the direct methods just used we have an approximate knowledge of the position, x_C, of the carbon atom. This is given as one of the tallest peaks of the *E-map*, the electron density map calculated using normalised structure factors. Using function `local_maxima` on the correct density, `rtmp_test$rr`, we can extract x_C.

```
idx <- local_maxima(rtmp_test$rr)
for (i in idx) {
  print(rtmp_test$rr[i])
}

## [1] 0.1613148
## [1] -0.05485548
## [1] 1.410237
## [1] 0.07034668
## [1] -0.04464219
## [1] 0.2852082
## [1] 0.2852082
## [1] -0.04464219
## [1] 0.07034668
## [1] 1.410237
## [1] -0.05485548

# The approximate carbon position corresponds to idx[3]
rtmp_test$rr[idx[3]]

## [1] 1.410237

rtmp_test$x[idx[3]]

## [1] 1.86
```

Thus, the initial (not accurate) position of the carbon atom is 1.86. Furthermore we don't know at this stage what the B_C value is. The initial density map using this position and $B_c = 0$ does not match well with the correct map.

```
# Correct density
rtmp <- fousynth(a=sdata$a,Fmod=ftmp$Fmod[1:12],
                 Fpha=ftmp$Fpha[1:12],hidx=1:12,N=1000)


# Initial (approximate) density
sdata0 <- sdata
sdata0$x0 <- rtmp_test$x[idx[3]]
sdata0$B <- 0
```

```
ftmp0 <- strufac(hidx=1:12,sdata=sdata0)
rtmp0 <- fousynth(a=sdata0$a,Fmod=ftmp0$Fmod,
                  Fpha=ftmp0$Fpha,hidx=1:12,N=1000)

# Compare plots
ym <- min(rtmp0$rr)
yM <- max(rtmp0$rr)
plot(rtmp$x,rtmp$rr,type="l",xlab="x",ylab=expression(rho),
     ylim=c(ym,yM))
points(rtmp0$x,rtmp0$rr,type="l",col=2)
```



We need, thus, to change x_C and B_C so to make the two densities match as much as possible. This can be only done making use of the only experimental data available, the observed amplitudes $|F_{\mathrm{obs}}|$. Such a procedure goes under the name of *crystallographic refinement* and it is described as follows. The expression for the calculated structure factor of ”pinkerton2015” is,

$$F_h = 2f_C(h)\exp\left(-\frac{h^2}{4a^2}B_C\right)\cos\left(2\pi h\frac{x_C}{a}\right)$$

where C refers to the carbon atom. The refinement mostly used in crystallography seeks a linear change to the calculated structure factors. The refined position of the carbon atom is $x_C + \Delta X$ and its refined B-factor is $B_C + \Delta B$. The linear change in the amplitude of the calculated structure factor is,

$$|F_h| \quad \rightarrow \quad |F_h| + \Delta|F_h|$$

$$\Delta|F_h| = \frac{\partial F_h}{\partial x_C}\mathrm{sign}(F_h)\Delta x_C + \frac{\partial F_h}{\partial B_C}\mathrm{sign}(F_h)\Delta B_C$$

The equation describing the refinement is, therefore,

$$|F_{\mathrm{obs}}(h)| = |F_h| + \Delta|F_h|$$

or,

$$\Delta|F_h| = |F_{\mathrm{obs}}(h)| - |F_h|$$

The quantity $\Delta|F_h|$ is a linear combination of Δx_C and ΔB_C. When the relation is applied to all the observed reflections, the following matrix equation is obtained,

$$\begin{pmatrix} \partial F_{h_1}/\partial x_C \ \mathrm{sign}(F_{h_1}) & \partial F_{h_1}/\partial B_C \ \mathrm{sign}(F_{h_1}) \\ \partial F_{h_2}/\partial x_C \ \mathrm{sign}(F_{h_2}) & \partial F_{h_2}/\partial B_C \ \mathrm{sign}(F_{h_2}) \\ \dots & \dots \\ \partial F_{h_{12}}/\partial x_C \ \mathrm{sign}(F_{h_{12}}) & \partial F_{h_{12}}/\partial B_C \ \mathrm{sign}(F_{h_{12}}) \end{pmatrix} \begin{pmatrix} \Delta x_C \\ \Delta B_C \end{pmatrix} = \begin{pmatrix} |F_{\mathrm{obs}}(h_1)| - |F_{h_1}| \\ |F_{\mathrm{obs}}(h_2)| - |F_{h_2}| \\ \dots \\ |F_{\mathrm{obs}}(h_{12})| - |F_{h_{12}}| \end{pmatrix}$$

As the number of observation is larger than the number of parameters, the above matrix equation can be solved using least squares. The matrix equation above is essentially a regression equation of the type,

$$y = \beta_1 x_1 + \beta_2 x_2$$

where β_1 and β_2 are the regression coefficients to be determined; once these are found we set $\Delta x_C = \beta_1$ and $\Delta B_C = \beta_2$. The elements of the matrix above can be given explicitly once both partial derivatives are calculated. These are,

$$\frac{\partial F_h}{\partial x_C} = -\frac{4\pi h}{a} f_C(h) \exp\left(-\frac{h^2}{4a^2}B_C\right) \sin\left(2\pi h \frac{x_C}{a}\right)$$

$$\frac{\partial F_h}{\partial B_C} = -\frac{h^2}{2a^2} f_C(h) \exp\left(-\frac{h^2}{4a^2}B_C\right) \cos\left(2\pi h \frac{x_C}{a}\right)$$

Let's, next, apply the above formulas to determine the improved x_C and B_C.

```
# Preliminaries...
# 1) Miller indices
hidx <- 1:12
```

```r
# 2) Observed amplitudes
Fo <- ftmp$Fmod[1:12]

# 3) Initial xc and Bc
xc <- sdata0$x0
Bc <- 0

# 4) Structure to be updated
sdata0 <- sdata

# 5) Calculated structure factors
sdata0$x0 <- xc
sdata0$B <- Bc
ftmp0 <- strufac(hidx=1:12,sdata=sdata0)
Fc <- ftmp0$Fmod
Sns <- cos(ftmp0$Fpha*pi/180)

# Initial residual
RR <- 100*sum(abs(Fo-Fc))/sum(Fo)
RR
```

```
## [1] 153.2508
```

```r
# One cycle of refinement ...
# Scattering factors
ff <- scafac(h=hidx,a=sdata$a,Zj=6,occj=1,Bj=Bc)

# y part of the regression
y <- Fo-Fc

# x1 part of the regression
x1 <- (-4*pi/a)*hidx*ff*exp(-hidx^2*Bc/(4*a^2))*
  sin(2*pi*hidx*xc/a)*Sns

# x2 part of the regression
x2 <- (-hidx^2/(2*a^2))*ff*exp(-hidx^2*Bc/(4*a^2))*
  cos(2*pi*hidx*xc/a)*Sns

# Least-Squares regression
model <- lm(y~0+x1+x2)
smm <- summary(model)
```

```
# Update model
xc <- xc+smm$coefficients[1,1]
Bc <- Bc+smm$coefficients[2,1]
xc

## [1] 1.85325

Bc

## [1] 4.364311

sdata0$x0 <- xc
sdata0$B <- Bc
ftmp0 <- strufac(hidx=1:12,sdata=sdata0)
Fc <- ftmp0$Fmod
Sns <- cos(ftmp0$Fpha*pi/180)

# Updated residual
RR <- 100*sum(abs(Fo-Fc))/sum(Fo)
RR

## [1] 66.4034
```

Each cycle of refinement should bring the values of x_C and B_C closer to their ideal values $x_C = 1.833$ and $B_C = 13.333$. As the variation of $|F_h|$ is not exactly linear in x_C and B_C, the regression will not converge to the ideal values straight away. Furthermore, as the difference $\Delta|F_h|$ is, in fact, a nonlinear function of x_C and B_C, the convergence to the ideal values could be oscillatory and it might not be clear when the refinement cycles should stop. For this reason, a *residual factor* is usually calculated at each cycle in the following way:

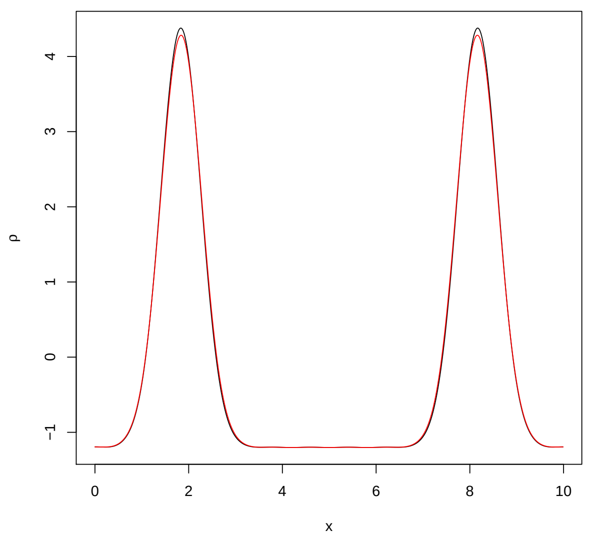
$$R \equiv 100\frac{\sum_h ||F_{\mathrm{obs}}(h)| - |F_h||}{\sum_h |F_{\mathrm{obs}}(h)|}$$

The high value of R in the initial cycle above is an indication of how far the calculated structure factors are from the observed data. The value of the residual should be small when the atomic positions and B factors are close to their ideal value. After the first cycle of refinement the residual clearly drops. But it's still far from a small value as x_c and B_C are not close enough to their ideal value. Let's carry out another refinement cycle.

```
# One cycle of refinement ...
# Scattering factors
ff <- scafac(h=hidx,a=sdata$a,Zj=6,occj=1,Bj=Bc)

# y part of the regression
y <- Fo-Fc

# x1 part of the regression
x1 <- (-4*pi/a)*hidx*ff*exp(-hidx^2*Bc/(4*a^2))*
  sin(2*pi*hidx*xc/a)*Sns

# x2 part of the regression
x2 <- (-hidx^2/(2*a^2))*ff*exp(-hidx^2*Bc/(4*a^2))*
  cos(2*pi*hidx*xc/a)*Sns

# Least-Squares regression
model <- lm(y~0+x1+x2)
smm <- summary(model)

# Update model
xc <- xc+smm$coefficients[1,1]
Bc <- Bc+smm$coefficients[2,1]
xc

## [1] 1.839266

Bc

## [1] 13.84426

sdata0$x0 <- xc
sdata0$B <- Bc
ftmp0 <- strufac(hidx=1:12,sdata=sdata0)
Fc <- ftmp0$Fmod
Sns <- cos(ftmp0$Fpha*pi/180)

# Updated residual
RR <- 100*sum(abs(Fo-Fc))/sum(Fo)
RR

## [1] 2.574678
```

The residual has now dropped below the 5% threshold. In crystallography this threshold is considered good enough for achieving accuracy. This is, then, the final comparison between ideal and calculated electron densities.

```r
# True density
ftmpT <- strufac(hidx=1:12,sdata=sdata)
rtmpT <- fousynth(a=sdata$a,Fmod=ftmpT$Fmod,Fpha=ftmpT$Fpha,
                  hidx=1:12,N=1000)

# Calculated (final) density
rtmpC <- fousynth(a=sdata$a,Fmod=ftmp0$Fmod,Fpha=ftmp0$Fpha,
                  hidx=1:12,N=1000)

# Compare densities
ym <- min(rtmpT$rr,rtmpC$rr)
yM <- max(rtmpT$rr,rtmpC$rr)
plot(rtmpT$x,rtmpT$rr,type="l",xlab="x",ylab=expression(rho),
     ylim=c(ym,yM))
points(rtmpC$x,rtmpC$rr,type="l",col=2)
```



The matching is not complete, but it is very good. In real applications one can never know the correct phases (or signs) and some errors are to be expected in the final electron density.