

Computational aspects of the Rietveld Method

Needs for precise refinements and microstructural effects:
Improvement of treatment of peak shapes, Rietveld algorithm, ...

Juan Rodríguez-Carvajal
Laboratoire Léon Brillouin, (CEA-CNRS), CEA/Saclay
FRANCE



Experimental powder pattern

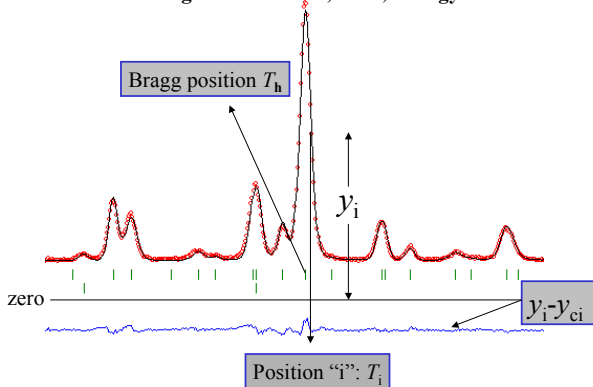
A powder diffraction pattern can be recorded in numerical form for a discrete set of scattering angles, times of flight or energies. We will refer to this scattering variable as : T . The experimental powder diffraction pattern is usually given as three arrays :

$$\{T_i, y_i, \sigma_i\}_{i=1,2,\dots,n}$$

The profile can be modelled using the calculated counts: y_{ci} at the i th step by summing the contribution from neighbouring Bragg reflections plus the background.



Powder diffraction profile:
scattering variable T : 2θ , TOF, Energy



The calculated profile of powder diffraction patterns

$$y_{ci} = \sum_{\{h\}} I_h \Omega(T_i - T_h) + b_i$$

$I_h = I_h(\beta_I)$ Contains structural information: atom positions, magnetic moments, etc

$\Omega = \Omega(x_{hi}, \beta_P)$ Contains micro-structural information: instr. resolution, defects, crystallite size...

$b_i = b_i(\beta_B)$ Background: noise, incoherent scattering diffuse scattering, ...



The calculated profile of powder diffraction patterns

$$y_{ci} = \sum_{\{h\}} I_h \Omega(T_i - T_h) + b_i$$

The symbol $\{h\}$ means that the sum is extended only to those reflections contributing to the channel " i ".

This should be taken into account (resolution function of the diffractometer and sample broadening) before doing the actual calculation of the profile intensity.

This is the reason why some Rietveld programs are run in two steps



Several phases ($\phi = 1, n_\phi$) contributing to the diffraction pattern

$$y_{ci} = \sum_{\phi} S_{\phi} \sum_{\{\phi h\}} I_{\phi, h} \Omega(T_i - T_{\phi, h}) + b_i$$

Several phases ($\phi = 1, n_\phi$) contributing to several ($p = 1, n_p$) diffraction patterns

$$y_{ci}^p = \sum_{\phi} S_{\phi}^p \sum_{\{\phi h\}} I_{\phi, h}^p \Omega^p(T_i - T_{\phi, h}) + b_i^p$$



$$y_{ci} = \sum_{\{h\}} I_h \Omega(T_i - T_h) + b_i$$

$$I_h = S \{ L p O A C F^2 \}_h$$

Integrated intensities are proportional to the square of the structure factor F .

The factors are:

Scale Factor (S), Lorentz-polarization (Lp), preferred orientation (O), absorption (A), other "corrections" (C)

The Structure Factor contains the structural parameters (isotropic case)

$$F(\mathbf{h}) = \sum_{j=1}^n O_j f_j(h) T_j \sum_s \exp \left\{ 2\pi i \left[\mathbf{h} \{ S | \mathbf{t} \}_s \mathbf{r}_j \right] \right\}$$

$$\mathbf{r}_j = (x_j, y_j, z_j) \quad (j = 1, 2, \dots, n)$$

$$T_j = \exp \left(-B_j \frac{\sin^2 \theta}{\lambda^2} \right)$$



Structural Parameters (simplest case)

$$\mathbf{r}_j = (x_j, y_j, z_j)$$

Atom positions (up to $3n$ parameters)

$$O_j$$

Occupation factors (up to $n-1$ parameters)

$$B_j$$

Isotropic displacement (temperature) factors (up to n parameters)

Structural Parameters (complex cases)

As in the simplest case plus additional (or alternative) parameters:

- Anisotropic temperature (displacement) factors
- Anharmonic temperature factors
- Special form-factors (Symmetry adapted spherical harmonics), TLS for rigid molecules, etc.
- Magnetic moments, coefficients of Fourier components of magnetic moments, basis functions, etc.



The Structure Factor in complex cases

$$F(\mathbf{h}) = \sum_{j=1}^n O_j f_j(h) T_j \sum_s g_j(\mathbf{h}_s) \exp \left\{ 2\pi i \left[\mathbf{h} \{ S | \mathbf{t} \}_s \mathbf{r}_j \right] \right\}$$

$$\mathbf{h}_s = \begin{pmatrix} h \\ k \\ l \end{pmatrix}_s = S_s^T \begin{pmatrix} h \\ k \\ l \end{pmatrix} \quad (s = 1, 2, \dots, N_G)$$

$$g_j(\mathbf{h}_s)$$

Complex form factor of object j
Anisotropic DPs
Anharmonic DPs

The peak shape function of powder diffraction patterns contains the Profile Parameters

$$\Omega(x_{hi}, \beta_p) = \Omega(T_i - T_h, \beta_p)$$

$$\Omega(x) = g(x) \otimes f(x) = \text{instrumental} \otimes \text{intrinsic profile}$$

$$\int_{-\infty}^{+\infty} \Omega(x) dx = 1$$

In most cases the observed peak shape is approximated by a linear combination of Voigt (or pseudo-Voigt) functions

$$\Omega(x) \approx \sum L(x) \otimes G(x) = \sum V(x)$$



Properties of the Voigt function

$$V(x) = V_1(x) \otimes V_2(x)$$

The Voigt function has proven to be a very good experimental approximation in many cases

$$\beta_L = \beta_{1L} + \beta_{2L} \quad \leftarrow \text{Lorentzian breadths simply have to be summed}$$

$$\beta_G^2 = \beta_{1G}^2 + \beta_{2G}^2 \quad \leftarrow \text{Gaussian breadths have to be summed quadratically}$$

$$\beta_{fL} = \beta_{hL} - \beta_{gL} \quad \leftarrow \text{Correction for instrumental broadening}$$

$$\beta_{fG}^2 = \beta_{hG}^2 - \beta_{gG}^2$$

The **Rietveld Method** consist of refining a crystal (and/or magnetic) structure by minimising the weighted squared difference between the observed and the calculated pattern against the parameter vector: β

$$\chi^2 = \sum_{i=1}^n w_i \{y_i - y_{ci}(\beta)\}^2$$

$$w_i = \frac{1}{\sigma_i^2}$$

σ_i^2 : is the variance of the "observation" y_i

However, the **Rietveld Method** can be easily extended by using, instead of the traditional χ^2 (least squares), another **Cost Function** to be minimised against the parameter vector β

$$\text{Cost} = \sum_{i=1}^n F(\{y_i - y_{ci}(\beta)\})$$

$$\text{Cost} = -\log(\text{Likelihood})$$

Least squares: Gauss-Newton (1)

Minimum necessary condition: $\frac{\partial \chi^2}{\partial \beta} = 0$

A Taylor expansion of $y_{ic}(\beta)$ around β_0 allows the application of an iterative process. The shifts to be applied to the parameters at each cycle for improving χ^2 are obtained by solving a linear system of equations (normal equations)

$$A \delta \beta_0 = b$$

$$A_{kl} = \sum_i w_i \frac{\partial y_{ic}(\beta_0)}{\partial \beta_k} \frac{\partial y_{ic}(\beta_0)}{\partial \beta_l}$$

$$b_k = \sum_i w_i (y_i - y_{ic}) \frac{\partial y_{ic}(\beta_0)}{\partial \beta_k}$$

Least squares: Gauss-Newton (2)

The shifts of the parameters obtained by solving the normal equations are added to the starting parameters giving rise to a new set

$$\beta_1 = \beta_0 + \delta \beta_0$$

The new parameters are considered as the starting ones in the next cycle and the process is repeated until a convergence criterion is satisfied. The variances of the adjusted parameters are calculated by the expression:

$$\sigma^2(\beta_k) = (A^{-1})_{kk} \chi_v^2$$

$$\chi_v^2 = \frac{\chi^2}{N - P + C}$$

Least squares: a local optimisation method

- The least squares procedure provides (when it converges) the value of the parameters constituting the local minimum closest to the starting point
- A set of good starting values for all parameters is needed
- If the initial model is bad for some reasons the LSQ procedure will not converge, it may diverge.

Needs for precise refinements and microstructural effects

Precise refinements can be done with confidence only if the intrinsic and instrumental peak shapes are properly approximated.

At present

⇒ The approximation of the **intrinsic profile** is mostly based in the **Voigt (or pseudo-Voigt) function**

⇒ The approximation of the **instrumental profile** is also based in the **Voigt function** for constant wavelength instruments

⇒ For TOF the **instrumental+intrinsic profile** is approximated by the convolution of a **Voigt function** with **back-to-back exponentials** or with the **Ikeda-Carpenter function**.



Example: General 2θ dependence of the instrumental broadening (determined by a standard sample)

$$H_{hG}^2 = (U_f + (1 - \xi_f)^2 D_{JST}^2(\mathbf{a}_D)) \tan^2 \theta + \frac{I_{fG}}{\cos^2 \theta} + H_{gG}^2$$

$$H_{hL} = (X_f + \xi_f D_{JST}(\mathbf{a}_D)) \tan \theta + \frac{[Y_f + F_f(\mathbf{a}_S)]}{\cos \theta} + H_{gL}$$

The Gaussian and Lorentzian components of the instrumental Voigt function are interpolated between empirically determined values. If needed, axial divergence is convoluted numerically with the resulting profile.



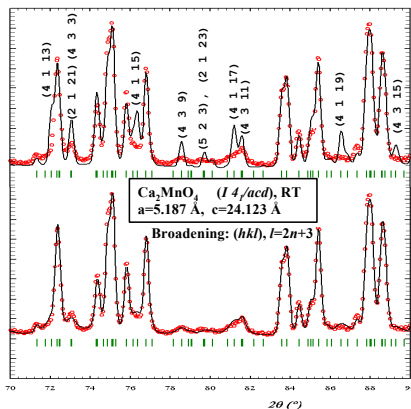
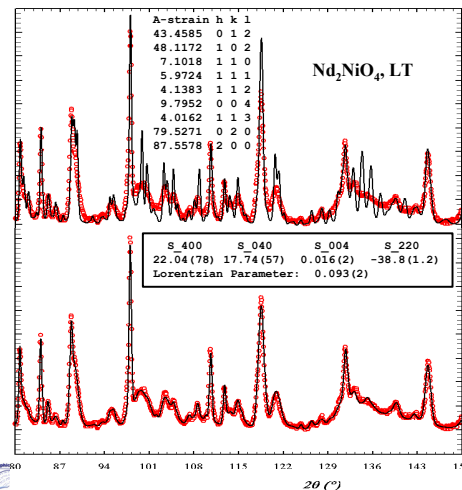
Microstructural effects and peak shapes (Rietveld)

Recent developments:

⇒ **Anisotropic peak broadening** (strain/size effects):
 quartic forms in hkl (dislocations, micro-twinning, composition fluctuations)
 spherical harmonics (complex size/microstrain effects)
 special reflections (stacking faults, antiphase domains, polytypes)

⇒ hkl -dependent shifts with respect to Bragg positions: special reflections, quartic forms, ...

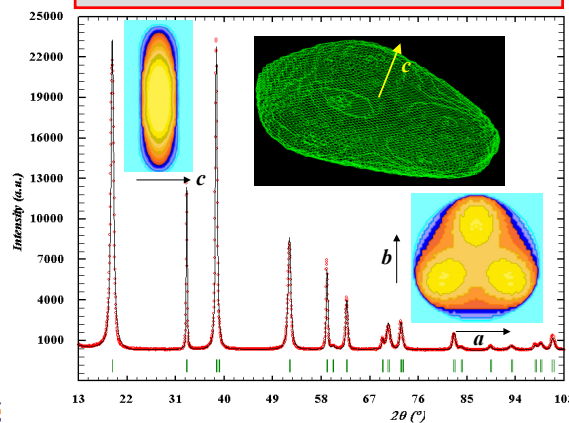
⇒ **New sample profiles**: Linear combination of pseudo-Voigt functions to mimic log-normal and gamma size distributions (Popa *et al.* *J. Appl. Cryst.* 35, 2002, 338-346)



Selective size broadening observed by neutron diffraction at room temperature (3T2, LLB) for superstructure reflections in Ca₂MnO₄. (top) Size parameter fixed to zero. (bottom) Single size parameter according to the rule (hkl), l=2n+3.



Size broadening in Ni(OH)₂



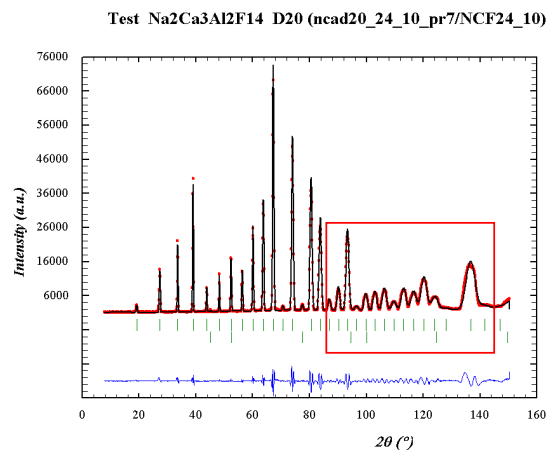
Problems when modeling the peak shape, a real case: low resolution neutron powder diffractometers

D20 at ILL:

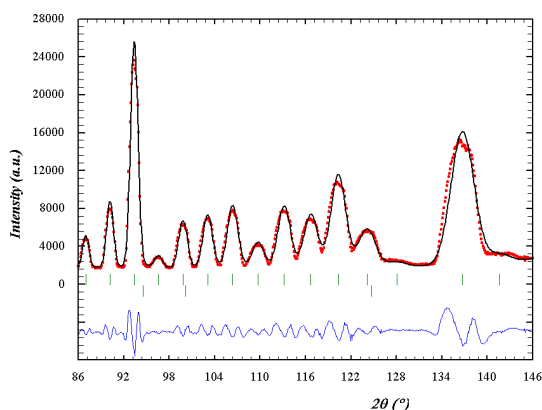
A diffraction pattern can be collected in less than a second!

Large graphite monochromator with a quite low take-off angle ($2\theta_M \approx 40^\circ$)

This implies that at high angle the peaks are broad and have a strange peak shape (that can be reproduced quite precisely by ray tracing or Monte Carlo simulations of the instrument !)



Test Na₂Ca₃Al₂F₁₄ D20 (ncad20_24_10_pr7/NCF24_10)



The inability to model peak shapes properly introduces a “systematic error” in the data treatment affecting the structural parameters and the estimation of their uncertainties

How can we solve this problem?



Future developments for Rietveld analysis: the treatment of the peak shape

Further step on complexity:

- ⇒ Fundamental parameters approach
- ⇒ Numerical instrumental profile (when needed)
- ⇒ Local convolution with analytical sample profile using FFT or interpolated direct convolution

$$\Omega(x) = FT^{-1} [G(t)F(t)] = g(x) \otimes f(x)$$

This is partially performed (with analytical functions) in the CCSL based code at ISIS and in FullProf/GSAS for the TOF case: $V(x) \otimes IK(x)$

In TOPAS the fundamental parameters approach is fully implemented



Future developments for Rietveld analysis: the treatment of the peak shape

- ⇒ Different components of both instrumental and sample profile functions are just multiplied in the Fourier space.
- ⇒ The global $G(t)$ may be provided in the instrumental resolution file in different forms, depending if it can be approximated by analytical functions or not.

$$G(t) = G_1(t)G_2(t)G_3(t) \dots$$

$$F(t) = F_1(t)F_2(t)F_3(t) \dots$$

This procedure is faster than the direct convolution using numerical integration when the number of points per profile is greater than ~ 64 .



Skeleton of the Rietveld algorithm

Calculations in a single cycle for all patterns

```
do n_pat=1,n_patt
  if(affpat(n_pat) == 0) cycle
  Select case (xunit(n_pat))
    Case("2theta")
      Call calc_pattern_2theta(n_pat)
    Case("TOF")
      Call calc_pattern_TOF(n_pat) ←
    Case("Energy")
      Call calc_pattern_Ed(n_pat)
    . . . . .
  End Select
end do
```

The core of the algorithm used in the Rietveld Method

The Rietveld algorithm: (do over points/reflections)

```
Subroutine calc_pattern_TOF(n_pat)
.....
DO i=1,npts(n_pat)
  ini=code_contribution(i,n_pat,"ini")
  fin=code_contribution(i,n_pat,"fin")
  IF(iprev <= fin) THEN
    DO j=iprev,fin
      CALL calcul_tof(j,n_pat) ←
    END DO
    iprev=MAX(iprev,fin+1)
  END IF
  CALL summat_tof(i,n_pat,ini,fin)
END DO
return
End Subroutine calc_patterns_TOF
```

$\sum_{\{h\}}$

The Rietveld algorithm: calculation for each reflection

```
SUBROUTINE calcul_tof(nn,n_pat)
.....
!Calculate contribution of micro-structure
CALL strain (n_pat,nn,h,iph,dst,ss,dvv)
CALL sizef (n_pat,nn,h,iph,dsiz,ss,dvs)
CALL shifhkl(n_pat,nn,h,iph,shv,ss,dshv)
!Calculate FWHM and so on ...
.....
!Different models to calculate the structure factors
Select case (Model_STF)
.....
case ("Magnetic_reflection_IREPS")
  call calmag_bas(n_pat,nn,iph,h,...,fnn)
.....
End Select
.....
CALL correct_tof(n_pat,nn,iph,h,fnn,ider)
!----Calculate and store part of derivatives
.....
RETURN
END SUBROUTINE calcul_tof
```

The Rietveld algorithm: (do over points/reflections)

```
Subroutine calc_patterns_TOF(n_pat)
.....
DO i=1,npts(n_pat)
  ini=code_contribution(i,n_pat,"ini")
  fin=code_contribution(i,n_pat,"fin")
  IF(iprev <= fin) THEN
    DO j=iprev,fin
      CALL calcul_tof(j,n_pat)
    END DO
    iprev=MAX(iprev,fin+1)
  END IF
  CALL summat_tof(i,n_pat,ini,fin) ←
END DO
return
End Subroutine calc_patterns_TOF
```

The Rietveld algorithm: (do over reflections, make sums, derivatives, LSQ matrix)

```
SUBROUTINE summat_tof(ipm,n_pat,ini,fin)
.....
! Calculate Ycalc and its derivative w.r.t all parameters
DO i=ini,fin
  ! Profile calculation
  Select case (nprof(n_pat))
    case ("pV-conv-exp")
      omega= tof_peak2(delta,ider)
    ..
  End Select
  omegap= scale_Lp abs*omega
  yc(ipm,n_pat)=yc(ipm,n_pat)+omegap*ff(i,n_pat)*corr(i,n_pat)
  ! Loop over MAXS parameters for completing derivatives
  j=MOD(i,MaxOVERL)+1
  DO k=1,maxs
    ..
    deriv(k)=dersto(j,k)*der*omegap+deriv(k)
  END DO
END DO
! Derivatives w.r.t. background parameters
! Construction of the Least-squares Matrix and Vector
.....
END SUBROUTINE summat_tof
```

The Rietveld algorithm: summary

Do for N_{cycles}
Do for **Patterns** \leftarrow may be done in parallel
Do for **points in Patterns**
Do for contributing reflections
Calculate broadening w.r.t to IRF
Calculate structure factors+derivatives
Sum contributions (LSQ matrix + vector)
calculate profile for current point and reflections contributing to it (convolution \Rightarrow neighbours needed) profile derivatives
End do reflections
End do **points in Patterns**
End do **Patterns**
Invert LSQ matrix and update the free parameters
Tests for convergence (if convergence is reached exit!)
End do N_{cycles}



May the Rietveld algorithm be improved?

New ideas are needed to improve the efficiency:

New data structures?

Store individual peak shapes?

Change the order of loops?

Modularise different parts of the calculations?

...



The Rietveld algorithm in a context of increasing complexity

With the forthcoming high performance instruments, and increasing complexity, **we need an improvement of the algorithms for handling Rietveld refinements if we want to preserve interactivity.**

Options:

- \Rightarrow Develop small specialized Rietveld programs
- \Rightarrow Maintain the possibility of general treatment and improve the efficiency by making strong changes on the Rietveld codes.
- \Rightarrow A combination of both ...



Conclusions and perspectives

The increasing complexity of instruments and the necessity of better refinements call for collaborative teams for improving the existing software and develop new tools.

A list of tasks and priorities is needed to undertake a rational software development.

This opens new opportunities for young people wishing to dedicate their scientific activities to
Crystallographic Computing.

