

# **Analysis and evaluation of wide-angle scattering data of non-graphitic carbons using GNU *Octave* – theory and application**

## **Supporting information part A**

### ***Octave* installation and examples**

Authors

**Oliver Osswald<sup>a\*</sup> and Bernd M. Smarsly<sup>a</sup>**

<sup>a</sup>Institute of Physical Chemistry, Justus-Liebig-University Giessen, Heinrich-Buff-Ring 17, Giessen, 35392, Germany

Correspondence email: [oliver.osswald@phys.chemie.uni-giessen.de](mailto:oliver.osswald@phys.chemie.uni-giessen.de)

**Funding information** Financial support is provided by the DFG via the RTG (Research Training Group) 2204 “Substitute Materials for Sustainable Energy Technologies”.

## Supporting information part A – *Octave* installation and examples

S1. Overview of all used parameters .....	3
S2. Implementation and calculation time consumption of $\nu$ .....	7
S3. Results of the refined samples for the verification of the used software including the calculation times for the WAXS refinements .....	9
S4. Tests for fitting the (004)-region of the LSPP-1200 WANS-data .....	15
S5. Download and usage of <i>iObs</i> .....	22
S5.1. Compile of an *.oct file for using <i>iObs</i> with <i>Octave</i> .....	22
S6. Installation and updates .....	23
S6.1. Installation under <i>Microsoft Windows</i> .....	23
S6.2. Installation under <i>MacOS</i> .....	23
S6.3. Installation under <i>GNU/Linux</i> , <i>BSD</i> and other systems .....	23
S6.4. Installation of <i>optim</i> for non- <i>Windows</i> builds .....	23
S6.5. Links and updates .....	23
S7. Usage of <i>Octave</i> .....	25
S8. Example refinement .....	29
S8.1. Step 0 – initial situation .....	29
S8.2. Step 1 – path and data adjustments .....	29
S8.3. Step 2 – manual background .....	31
S8.4. Step 3a – manual fitting – interlayer .....	32
S8.5. Step 4a – manual fitting – intralayer .....	33
S8.6. Step 5a – manual fitting – all .....	34
S8.7. Step 3b – automatic fitting – initialization .....	35
S8.8. Step 4b – automatic fitting - fine adjustment .....	36
S8.9. Common warnings/errors during the refinement and how to fix them .....	36
S8.10. Initial refinement script lines 1 – 134 .....	36
S8.11. Refinement script lines 1 – 134 after the manual fitting .....	39
S9. <i>Octave</i> cannot always calculate parameter errors – what to do .....	44
S10. Whole refinement script .....	49

### S1. Overview of all used parameters

In addition to 2.2, not the microstructure parameters itself, but parameters of distribution functions are refined ( $\mu$  and  $\beta$  for interlayer parameters,  $\nu$  and  $\alpha$  for intralayer parameters) from which the microstructure parameters are calculated (equations (16) – (23)). Therefore, a brief overview of all parameters used for the refinement and the received microstructure parameters as well as some other important units is given here.

For the number deviation of the average number of layers per stacks the following equations were found (Ruland & Smarsly, 2002):

$$n(N) = \frac{\beta^\mu}{\Gamma(\mu)} N^{\mu-1} \exp(-\beta N) \quad (16)$$

$$\langle N \rangle = \frac{\mu}{\beta} \quad (17)$$

$$L_c = \bar{a}_3 \frac{\mu+1}{\beta} \quad (18)$$

$$\kappa_c = \frac{1}{\mu} \quad (19)$$

**Table S1** Overview of all used parameters for a refinement using *Octave*. The column “influence on scattering data” describes, if the parameter has a large (+), medium (o) or small (-) influence on the theoretical intensity.

Parameter	Parameter in <i>Octave</i>	Parameter used for refinement	Description	Influence on scattering data
$q$	q	Measured	Scattering vector ( $= 2 \cdot \pi \cdot s$ )	
$s$	s	Measured	Scattering vector ( $= 2 \cdot \sin(\theta) / \lambda$ )	
$\lambda$	wavelength	Given constant	Wavelength of used radiation	
$\theta$	theta	Measured	Scattering angle, half „Bragg-angle“	
$\bar{a}_3$	a3	$a_3$	Average layer distance	+
$a_{3 \text{ min}}$	a3min	$= \bar{a}_3 - da_3$	Minimal layer distance	+ (only for visible (004) reflection, else -)
$\sigma_3$	sig3	$\sigma_3$	Disorder of the stacks (standard deviation of $a_3$ )	+
$L_a$	La	$= (\nu + 1)/\alpha$	Average graphene layer size	+ (calculated parameter)
$l_m$	lm	$= \nu/\alpha$	Average chord length	+ (calculated parameter)
$\kappa_a$	kapa	$= 1/\nu$	Polydispersity of chord length	Depends a lot on the experiment, see 2.3
$l_{cc}$	lcc	$l_{cc}$	Average C-C bond length	+
$\sigma_1$	sig1	$\sigma_1$	Disorder of the layers (i.e. stress and strain)	+
$\varepsilon_1$	eps1	-	Disorder of graphene layers due to local strains	-- (not implemented)
$\kappa_r$	kapr	$= 3\pi^2 (1/\nu + 1)/32 - 1$	Polydispersity of the radius of the graphene layers	Currently not used
$N$	N	$= (\mu+1)/\beta$	Average number of graphene layers per stack	+ (calculated parameter)

$L_c$	Lc	$= \bar{a}_3 \cdot (\mu+1)/\beta = L_c \cdot \bar{a}_3$	Average stack height	+	(calculated parameter)
$\kappa_c$	kapc	$= 1/\mu$	Polydispersity of stack height	o	(calculated parameter, but generally high error)
$\varepsilon_3$	eps3	$= a_3/a_{3 \min}$	Disorder of stacks due to local strains	o	(calculated parameter, but generally high error)
$q$	q	$q$	Preferred orientation	+	
$c_H$	cH	$c_H$ (constant)	Concentration of unorganized hydrogen	-	
$c_N$	cN	$c_N$ (constant)	Concentration of unorganized nitrogen	o	(depends on value)
$c_O$	cO	$c_O$ (constant)	Concentration of unorganized oxygen	o	(depends on value)
$c_S$	cS	$c_S$ (constant)	Concentration of unorganized sulfur	+	
$\Delta_{an}$	dan	$\Delta_{an}$	Anisotropy of atomic form factor of carbon	-	
$k$	k	$k$	Normalization constant for $\log_{10}(k \cdot \text{Ie.u.} + \text{const1}) + \text{const2}$	+	
$const_1$	const1	$const_1$	Constant shift for $\log_{10}(k \cdot \text{Ie.u.} + \text{const1}) + \text{const2}$	+	
$const_2$	const2	$const_2$	Non-constant (linear) shift for $\log_{10}(k \cdot \text{Ie.u.} + \text{const1}) + \text{const2}$	+	
$g$	g	$g$	Factor for exponential damping of the scattering intensity with $\text{Ie.u.} = \exp(g \cdot s) \cdot \text{Ie.u.}$	+	
$Q$	b	$Q$	Additional parameter for incoherent background ( $Q \neq b$ , see Pfaff <i>et al.</i> , 2019)	-	
$\rho$	density	$\rho$ (constant)	Density of the sample	o	
$d$	sampleThickness	$d$ (constant)	Thickness of the sample	o	

---

$\mu_{\text{ab}}$	mue_ab	$\mu_{\text{ab}}$ (constant)	Absorption factor, calculated from $\rho$ density, sample thickness and wavelength	
-	polarizedBeam	constant	Is the beam polarized?	+
$\Phi$	polarization Degree	$\Phi$ (constant)	Polarization direction of beam in $^{\circ}$	+
$r$	par_r	$r$ (constant)	Radius of the goniometer (in cm; fixed due to experiment)	$\rho$
$\delta$	par_delta	$\delta$ (constant)	Divergence angle (in $^{\circ}$ ; to choose by user)	$\rho$
$l$	par_l	$l$ (constant)	Irradiated length (in cm; fixed during measurement)	$\rho$
$R$	R	$R$ (constant)	Parameter for position correction according to D2ttx	-- (not available)
$t$	t	$t$ (constant)	Parameter for position correction according to D2ttx	-- (not available)

---

*Influence on scattering data* is just the personal meaning of the author based on the experience of several refinements.

## S2. Implementation and calculation time consumption of $\nu$

The implementation of  $\nu$  as described in 2.3 in the main article

$$P_L(r) = \frac{1}{\Gamma(\nu+1)} [\Gamma(\nu+1, ar) - ar\Gamma(\nu, ar)] \quad (20)$$

$$\langle l \rangle = \frac{\nu}{\alpha} \quad (21)$$

$$L_a = \frac{\langle l^2 \rangle}{\langle l \rangle} = \frac{\nu+1}{\alpha} \quad (22)$$

$$\kappa_a = \frac{1}{\nu} \quad (23)$$

is used to calculate the profile shape of the intralayer reflections and therefore the intralayer parameters (e.g. the average layer extension  $L_a$ ). However,  $\nu$  is not refined, but fixed to a constant depending on the maximum measured scattering vector  $s_{\max}$  (see Table S2). Several calculations showed, that this implementation is very useful to improve the speed of the refinement, since the influence of  $\nu$  is only very small on a single reflection. Also, the parameters  $\alpha$  (for the calculation of the layer extension  $L_a$ ) and  $\sigma_1$  (stress and strain of the layers (“disorder”)) have a much higher influence on the profile shape. Hence, this usage of  $\nu$  does not degrade the physical assumptions and the resulting accuracy of the refinement. More precisely, it is not possible to determine  $\nu$  and therefore the polydispersity of the graphene layers ( $\kappa_a = 1/\nu$ ) in an experimental way.

In the following, an overview of the recommended minimal value of  $\nu$  for the maximum measured scattering vector is given. Table S2 is just a recommendation and for some samples or more special experiments, e.g. WANS with a low wavelength, it might be useful to use a larger value of  $\nu$ . In general,  $\nu < 4$  is not recommended. Additionally, the calculation time using *Octave* is about 4 times faster than using the algorithm from Pfaff *et al.* (2018) for  $\nu = 4$ .

**Table S2** Recommended value of  $\nu$  for a maximum measured scattering vector  $s_{\max}$ , regarding maximum polydispersity and the average calculation times using the algorithm from Pfaff *et al.* (2018) and the improved used in the present study, respectively. In general,  $\nu < 4$  is not recommended.

$\nu$	$s_{\max} / \text{\AA}^{-1}$	$\kappa_{c, \max}$	Duration for a single calculation	
			Pfaff <i>et. al</i> (2018)	<i>Octave</i>
4	1.4	0.250	21.6 s	5.9 s
5	1.85	0.200	30.1 s	13 s
6	2.35	0.167	39.3 s	25.2 s
7	2.8	0.143	49.9 s	44.2 s
8	3.1	0.125	61.6 s	75.1 s
9	3.5	0.111	74.5 s	114.3 s
10	3.85	0.100	88.6 s	166.7 s



### S3. Results of the refined samples for the verification of the used software including the calculation times for the WAXS refinements

**Table S3** Overview of the microstructure parameters of CTP-800 (carbonized coal tar pitch), AC Type H (activated carbon), FA-850 (carbonized furfuryl alcohol) and PIL (poly(ionic liquid) carbon fiber).

The parameters were rounded to the same number of digits as the values given in the original publications.  $\nu$  was fixed to 4, i.e.  $\kappa_a = 1/\nu$  was fixed to 0.25. In Faber *et al.* (2014) and Pfaff *et al.* (2018), the errors of the other parameters were estimated by the influence of changing the parameter on the quality of the fit. For *Octave manual* and *Octave fitted* the error was calculated inside the fitting process. For this refinement, the error is a mathematical calculated error and more accurate than in the references.

Parameter	AC (Faber <i>et al.</i> , 2014)	Type H	AC (Pfaff <i>et al.</i> , 2018)	Type H	AC ( <i>Octave manual</i> )	Type H	AC ( <i>Octave fitted</i> )	Type H
$L_a / \text{\AA}$	$32 \pm 4$		$31 \pm 3.9$		$30 \pm 7.3$		$30 \pm 7.1$	
$l_m / \text{\AA}$	$26 \pm 3.3$		$24 \pm 3$		$24 \pm 11.7$		$24 \pm 11.4$	
$\sigma_1$	$0.12 \pm 0.014$		$0.12 \pm 0.014^6$		$0.11 \pm 0.006$		$0.11 \pm 0.006$	
$\kappa_a / \text{\AA}$	0.25		0.25		0.25		0.25	
$N_m$	$1.8 \pm 0.23$		$1.8 \pm 0.23$		$1.7 \pm 0.37$		$2 \pm 0.4$	
$N$	$2.7 \pm 0.27$		$2 \pm 0.25$		$2 \pm 0.2$		$2 \pm 0.2$	
$L_c / \text{\AA}$	$7 \pm 0.9$		$7 \pm 0.9$		$7 \pm 0.6$		$7 \pm 0.6$	
$\kappa_c / \text{\AA}$	$0.15 \pm 0.02$		$0.12 \pm 0.02$		$0.15 \pm 0.02$		$0.15 \pm 0.02$	
$\bar{a}_3 / \text{\AA}$	$3.6 \pm 0.054$		$3.56 \pm 0.053$		$3.51 \pm 0.057$		$3.52 \pm 0.056$	
$\sigma_3 / \text{\AA}$	$0.57 \pm 0.057$		$0.5 \pm 0.05$		$0.45 \pm 0.058$		$0.46 \pm 0.058$	

<sup>6</sup> Not published but known due to personal contact to the author.

Parameter	CTP-800 (Faber <i>et al.</i> , 2014)	CTP-800 (Pfaff <i>et al.</i> , 2018)	CTP-800 ( <i>Octave</i> manual)	CTP-800 ( <i>Octave</i> fitted)
$L_a / \text{\AA}$	$19 \pm 2.4$	$19 \pm 2.4$	$18 \pm 1.3$	$18 \pm 1.3$
$l_m / \text{\AA}$	$16 \pm 2$	$15 \pm 1.9$	$15 \pm 2.1$	$15 \pm 2.1$
$\sigma_1$	$0.14 \pm 0.016$	$0.13 \pm 0.016^7$	$0.12 \pm 0.02$	$0.12 \pm 0.02$
$\kappa_a / \text{\AA}$	0.25	0.25	0.25	0.25
$N_m$	$2 \pm 0.25$	$2.2 \pm 0.28$	$1.5 \pm 0.59$	$1 \pm 0.6$
$N$	$3.9 \pm 0.49$	$3.8 \pm 0.48$	$3.4 \pm 0.39$	$3.4 \pm 0.39$
$L_c / \text{\AA}$	$13 \pm 1.6$	$13 \pm 1.6$	$12 \pm 0.8$	$12 \pm 0.8$
$\kappa_c / \text{\AA}$	$0.96 \pm 0.12$	$0.76 \pm 0.1$	$1.27 \pm 0.38$	$1.27 \pm 0.38$
$\bar{a}_3 / \text{\AA}$	$3.44 \pm 0.052$	$3.44 \pm 0.052$	$3.43 \pm 0.006$	$3.43 \pm 0.006$
$\sigma_3 / \text{\AA}$	$0.26 \pm 0.026$	$0.23 \pm 0.023$	$0.2 \pm 0.025$	$0.2 \pm 0.025$
Parameter	FA-850 (Faber <i>et al.</i> , 2014)	FA-850 (Pfaff <i>et al.</i> , 2018)	FA-850 ( <i>Octave</i> manual)	FA-850 ( <i>Octave</i> fitted)
$L_a / \text{\AA}$	$25 \pm 3.1$	$23 \pm 2.9$	$28 \pm 1.6$	$28 \pm 1.7$
$l_m / \text{\AA}$	$20 \pm 2.5$	$19 \pm 2.4$	$23 \pm 2.5$	$23 \pm 2.8$
$\sigma_1$	$0.13 \pm 0.016$	$0.13 \pm 0.0168$	$0.15 \pm 0.012$	$0.16 \pm 0.013$
$\kappa_a / \text{\AA}$	0.25	0.25	0.25	0.25
$N_m$	$1.5 \pm 0.19$	$1.6 \pm 0.2$	$1.6 \pm 0.28$	$2 \pm 0.7$
$N$	$2.3 \pm 0.29$	$2.3 \pm 0.29$	$2.3 \pm 0.17$	$2.8 \pm 0.4$
$L_c / \text{\AA}$	$8 \pm 1$	$8 \pm 1$	$8 \pm 0.5$	$10 \pm 1.3$
$\kappa_c / \text{\AA}$	$0.51 \pm 0.06$	$0.42 \pm 0.05$	$0.44 \pm 0.05$	$0.44 \pm 0.1$
$\bar{a}_3 / \text{\AA}$	$3.6 \pm 0.054$	$3.61 \pm 0.054$	$3.6 \pm 0.025$	$3.74 \pm 0.13$
$\sigma_3 / \text{\AA}$	$0.43 \pm 0.043$	$0.44 \pm 0.044$	$0.44 \pm 0.052$	$0.69 \pm 0.217$

<sup>7</sup> Not published but known due to personal contact to the author.<sup>8</sup> Not published but known due to personal contact to the author.

Parameter	PIL (Einert <i>et al.</i> , 2015)	PIL (Pfaff <i>et al.</i> , 2018)	PIL ( <i>Octave</i> manual)	PIL ( <i>Octave</i> fitted)
$L_a / \text{\AA}$	$20 \pm 2.5$	$20 \pm 2.5$	$22 \pm 0.5$	$22 \pm 0.5$
$l_m / \text{\AA}$	$16 \pm 1.9$	$16 \pm 2$	$18 \pm 0.8$	$18 \pm 0.9$
$\sigma_1$	0.25	$0.13 \pm 0.016^9$	$0.15 \pm 0.005$	$0.14 \pm 0.005$
$\kappa_a / \text{\AA}$	$1.7 \pm 0.21$	0.25	0.25	0.25
$N_m$	$3 \pm 0.36$	$1.7 \pm 0.21$	$1.5 \pm 0.14$	$1 \pm 0.1$
$N$	$11 \pm 1.4$	$3 \pm 0.38$	$2.6 \pm 0.09$	$2.6 \pm 0.09$
$L_c / \text{\AA}$	$11 \pm 1.4$	$11 \pm 1.4$	$9 \pm 0.3$	$9 \pm 0.2$
$\kappa_c / \text{\AA}$	$0.77 \pm 0.1$	$0.77 \pm 0.1$	$0.79 \pm 0.05$	$0.78 \pm 0.05$
$\bar{a}_3 / \text{\AA}$	$3.63 \pm 0.054$	$3.64 \pm 0.055$	$3.57 \pm 0.008$	$3.56 \pm 0.007$
$\sigma_3 / \text{\AA}$	$0.54 \pm 0.054$	$0.56 \pm 0.056$	$0.47 \pm 0.015$	$0.46 \pm 0.014$

<sup>9</sup> Not published but known due to personal contact to the author.

**Table S4** Calculation times for the different samples and a different amount of measurement points.

*HPC* means high-performance computing cluster (JustHPC at the Justus Liebig University Giessen using CentOS 7), *common PC* is a Windows computer using an Intel® Core™ i5-8400 CPU (4x 2.8 GHz) and 16 GB RAM. *Raspberry Pi* is a Raspberry PI 4B Rev. 1.1 using an ARM v7 processor (4x 1.5 GHz) and 4 GB RAM running at Raspbian 11 (bullseye). Since the Raspberry Pi uses a micro-SD card instead of a SSD, the read/write speed might also limit the calculation performance. The main influence on the calculation time has the amount of measurement points and not the processor, due to the missing availability of parallel computing. Of course, a high-performance computing cluster is faster, because on these systems a high number of different measurements can be refined simultaneously. All values are just examples and might vary due to different background processes on the different systems.

Sample	Measurement points	Calc.	time	Calc.	time	Calc.	time
		HPC		common PC		Raspberry Pi	
AC Type H	258	05:14		03:42		11:49	
CTP-800	163	01:50		02:13		04:08	
FA-850	300	04:36		07:56		05:13	
PIL	2392	22:17		15:55		50:54	

**Table S5** Overview of the microstructure parameters of H-2100/H-2800 (phenol formaldehyde resorcinol) and LSPP-1200 (low softening point pitch) measured by wide-angle neutron scattering (WANS).

The parameters were rounded to the same number of digits as the values given in the original publications.  $\nu$  was fixed to 4, i.e.  $\kappa_a = 1/\nu$  was fixed to 0.25. The error for each parameter for the references and the manual fitting is about 10-15 %.

	H-2100	H-2100	H-2800	H-2800
Parameter	Pfaff <i>et al.</i> , 2019	<i>Octave</i> automatic	Pfaff <i>et al.</i> , 2019	<i>Octave</i> automatic
$L_a / \text{\AA}$	40	$34 \pm 0.3$	51	$51 \pm 1.2$
$l_m / \text{\AA}$	32	$27 \pm 0.5$	43	$41 \pm 1.9$
$l_{cc} / \text{\AA}$	1.413	$1.418 \pm 0.0002$	1.419	$1.421 \pm 0.0002$
$\sigma_1$	0.067	$0.056 \pm 0.0025$	0.047	$0.052 \pm 0.0024$
$L_c / \text{\AA}$	14	$11 \pm 0.4$	23	$19 \pm 1$
$N_m / \text{\AA}$	3	$2 \pm 0.3$	5	$3 \pm 0.7$
$N$	4	$3 \pm 0.2$	7	$6 \pm 0.4$
$\kappa_c / \text{\AA}$	0.37	$0.43 \pm 0.029$	0.49	$0.79 \pm 0.032$
$\bar{a}_3 / \text{\AA}$	3.56	$3.49 \pm 0.005$	3.48	$3.46 \pm 0.003$
$a_{3,\min} / \text{\AA}$	3.16	$2.49 \pm 0.076$	3.3	$2.87 \pm 0.037$
$\sigma_3 / \text{\AA}$	0.44	$0.34 \pm 0.009$	0.25	$0.2 \pm 0.008$
$\eta$	0.86	$0.89 \pm 0.003$	0.93	$0.96 \pm 0.002$

**Table S6** Overview of the microstructure parameters of H-2100/H-2800 (phenol formaldehyde resorcinol) and LSPP-1200 (low softening point pitch) measured by wide-angle neutron scattering (WANS).

The parameters were rounded to the same number of digits as the values given in the original publications.  $\nu$  was fixed to 4, i.e.  $\kappa_a = 1/\nu$  was fixed to 0.25. The error for each parameter for the references and the manual fitting is about 10-15 %.

	LSPP-1200	LSPP-1200	LSPP-1200
Parameter	Pfaff <i>et al.</i> , 2019	<i>Octave</i> automatic	<i>Octave</i> manual
$L_a / \text{\AA}$	27	$23 \pm 0.1$	23
$l_m / \text{\AA}$	22	$18 \pm 0.2$	18
$l_{cc} / \text{\AA}$	1.418	$1.42 \pm 0.0003$	1.420
$\sigma_1$	0.062	$0.066 \pm 0.0037$	0.066
$L_c / \text{\AA}$	17	$31 \pm 3.3$	15
$N_m / \text{\AA}$	3	$0 \pm 0.7$	3
$N$	5	$9 \pm 1.2$	4
$\kappa_c / \text{\AA}$	0.63	$100 \pm 559.241$	0.67
$\bar{a}_3 / \text{\AA}$	3.48	$3.51 \pm 0.006$	3.51
$a_{3,\min} / \text{\AA}$	3.00	$2.51 \pm 0.1$	3.21
$\sigma_3 / \text{\AA}$	0.30	$0.45 \pm 0.009$	0.33
$\eta$	1.00	$1 \pm 0.002$	1

#### S4. Tests for fitting the (004)-region of the LSPP-1200 WANS-data

In section 5.3 of the main article, the problem of fitting the (004)-region of the WANS data of the sample LSPP-1200 was discussed. The main problem is the poor fitting of the (004)-region using the automatic fit from the provided *Octave* script. Hence, an automatic refinement using the whole data range must be adjusted manually afterwards (Figure 9 of the main article). To get deeper insights and a more detailed understanding of this issue, some other refinements were performed using only a smaller range of the measured data (Figure S1 - Figure S3).

First, only the refinement range was reduced and the resulting scattering curve was extrapolated over the entire range (red). Second, the normalization parameters were fixed and the other microstructure parameter were refined and the result extrapolated (blue). In the next step, only the interlayer parameters with (green) and without the normalization parameters (purple) were refined and the result extrapolated. Since at higher values of the scattering vector  $s$ , only intralayer reflections are visible, a smaller range is sufficient to determine the interlayer structure parameters. Data at higher  $s$ -values do not lead to more accurate results regarding the interlayer structure parameters.

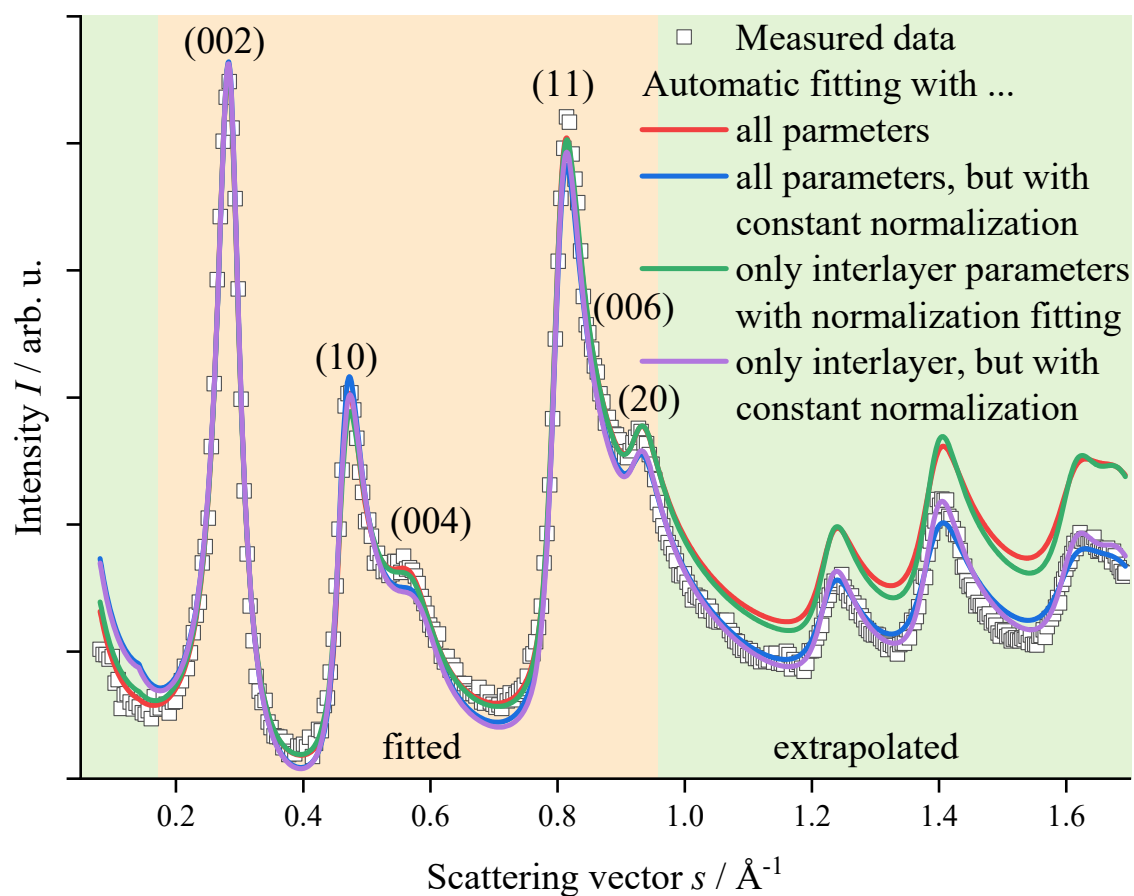
Three main conclusions can be drawn from these tests:

1. Looking at the refinements including the normalization parameters, it becomes evident that the normalization parameters were calculated wrong using only the smaller data range. On the other hand, the scattering data in this area including the (004)-region can be refined well. Overall, this indicates that the incoherent background of the scattering data was not subtracted properly.
2. Even if the measured data are hardly influenced by noise and the fitted curve lies almost perfectly on the data, the intralayer parameters cannot be determined exactly. For the higher order reflections ( $s > 1.2 \text{ \AA}^{-1}$ ), slight differences are visible between the “without normalization” and “only interlayer” fits. This means, that *Octave* tries to use the overlapping intralayer reflections as small background correction to refine all data as best as possible. This results in slightly calculated intralayer reflections and therefore other intralayer parameters.
3. Another explanation for 2 can be found by the influence of the layer disorder on the scattering data: In general, a stronger layer disorder causes broader reflections, with higher order reflections becoming broader (Ruland & Smarsly, 2002). So the used range for the refinement might be too small to determine this disorder correctly, since the influence is too small on these lower ordered reflections. On the other hand, with higher values of the scattering vector  $s$ , no more interlayer reflections are visible, so an extension of the measurement range would not solve this problem.

However, overall the main problem for the automatic refinement regarding the (004)-reflection is its shape: It is too much a shoulder rather than a clearly visible reflection and therefore, it is to “insignificant” for the *Octave* fitting routine compared to the whole scattering curve, especially to the neighboring valleys/feet which makes the manual adjustment necessary.

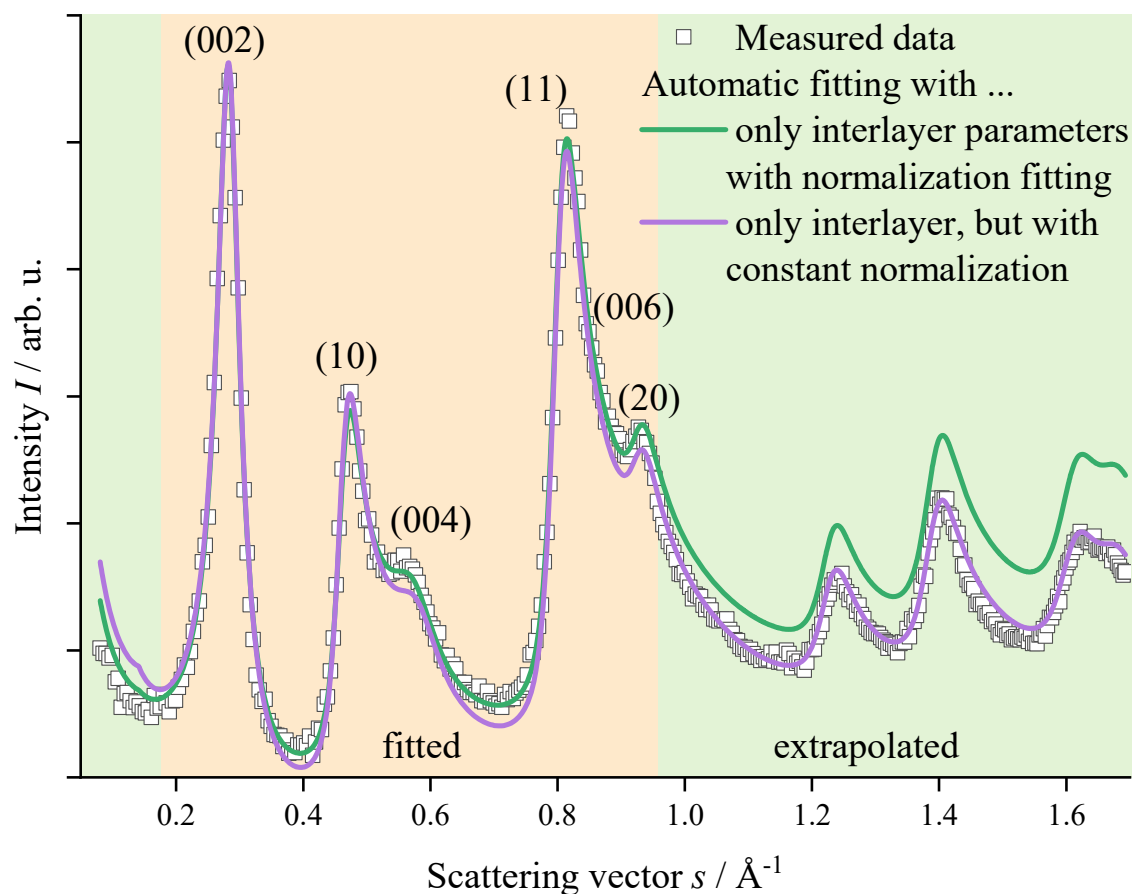
As another test, in the ranges of  $0.33 \text{ \AA}^{-1} < s < 0.44 \text{ \AA}^{-1}$  and  $0.63 \text{ \AA}^{-1} < s < 0.77 \text{ \AA}^{-1}$  not all measured points were refined. Instead, some points were left out during the refinement and extrapolated afterwards (Figure S4 - Figure S6). This test confirms that for *Octave* the feet/valleys besides the (004) and (11) reflection are more relevant in

the fitting algorithm as the (004) reflection itself. Figure S6 shows, that only every 30<sup>th</sup> point (in the  $s$ -ranges 0.33-0.44 & 0.63-0.77 Å<sup>-1</sup>) should be considered in this case, while more points would still influence the results. Nevertheless, the results from the automatic refinement are different compared to the manual refinement, especially the stacks are now significant higher ( $L_c$ ). At first glance, this sounds like a significant intrinsic flaw between these two methods. However, this difference can be explained on closer inspection of the individual parameters: With the manual fit, both the difference between  $a_3$  and  $a_{3\text{ min}}$  and the absolute value for  $\sigma_3$  are smaller. This higher order results in sharper interlayer reflections in WAXS/WANS data. To compensate for this effect, the crystallite size, i.e. the stack height, has to be smaller, which in turn leads to broader reflections. Overall, these effects balance each other out, so that both fits and both results are plausible.

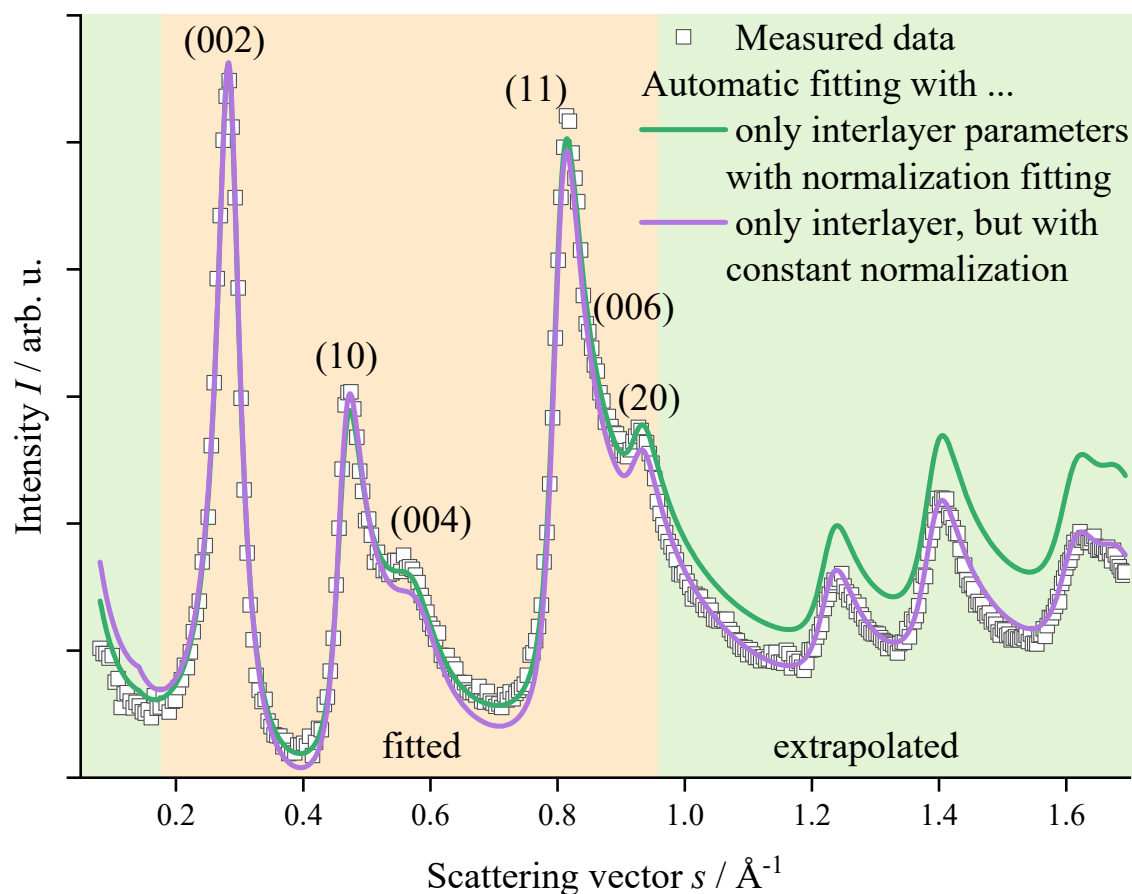


**Figure S1** Tests for the automatic *Octave* refinement with special attention to the (004)-region of WANS data from LSPP-1200 (low softening-point pitch). For all tests, only the orange area was fitted and the green areas extrapolated. Tests were performed with all parameters (red), the interlayer and intralayer parameters without the normalization (blue), the interlayer parameters with the normalization (green) and only with the interlayer parameters (purple). Only every 5<sup>th</sup> data point is shown.

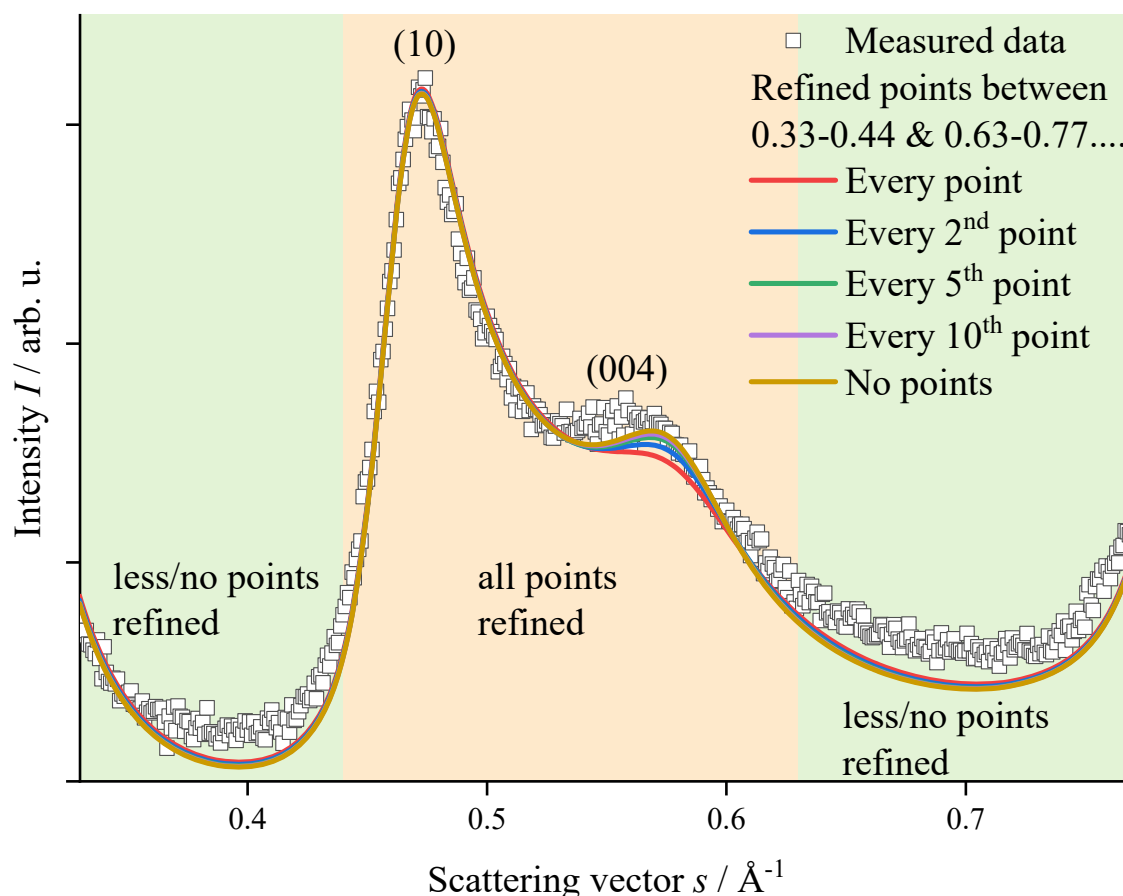




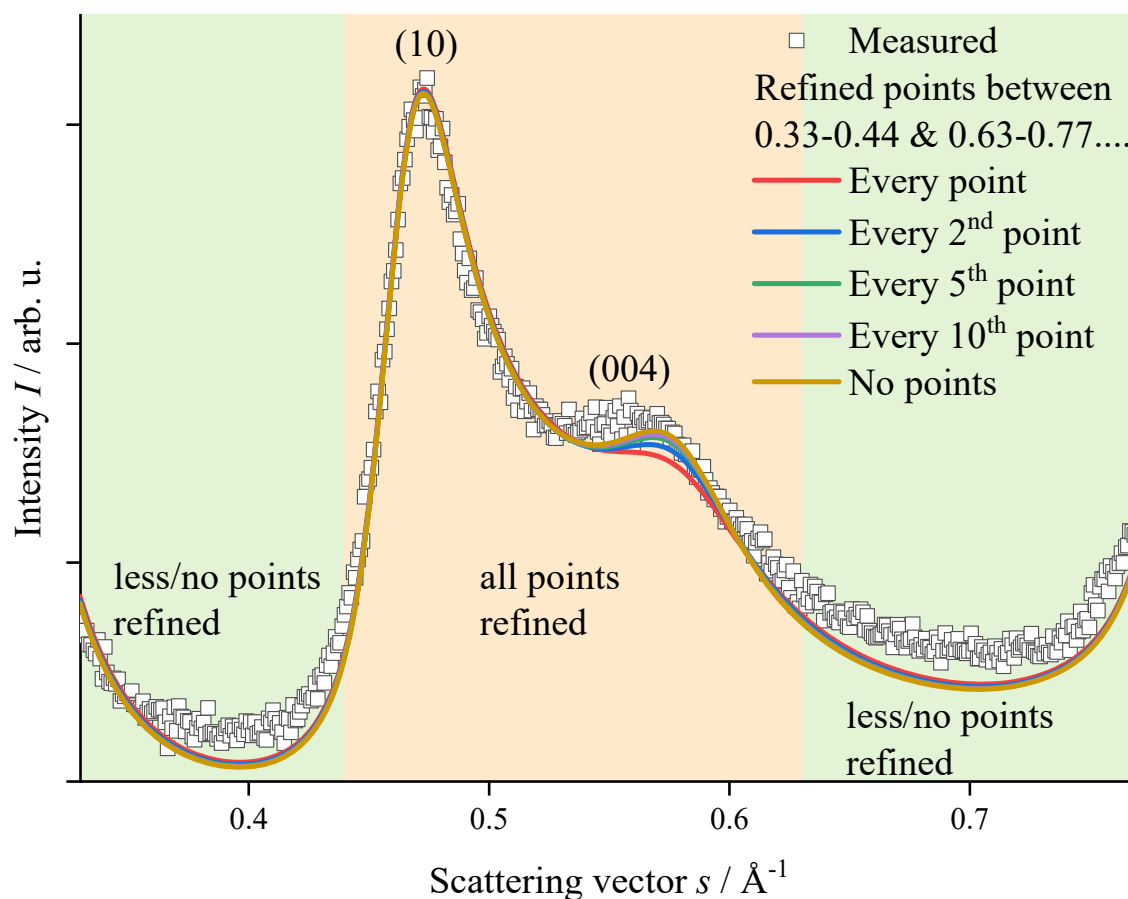
**Figure S2** Tests for the automatic *Octave* refinement with special attention to the (004)-region of WANS data from LSPP-1200 (low softening-point pitch). For all tests, only the orange area was fitted and the green parts extrapolated. Tests were performed with all parameters (red), the interlayer and intralayer parameters without the normalization (blue), the interlayer parameters with the normalization (green) and only with the interlayer parameters (purple). For the other refinements, look at Figure S1 or Table S3.



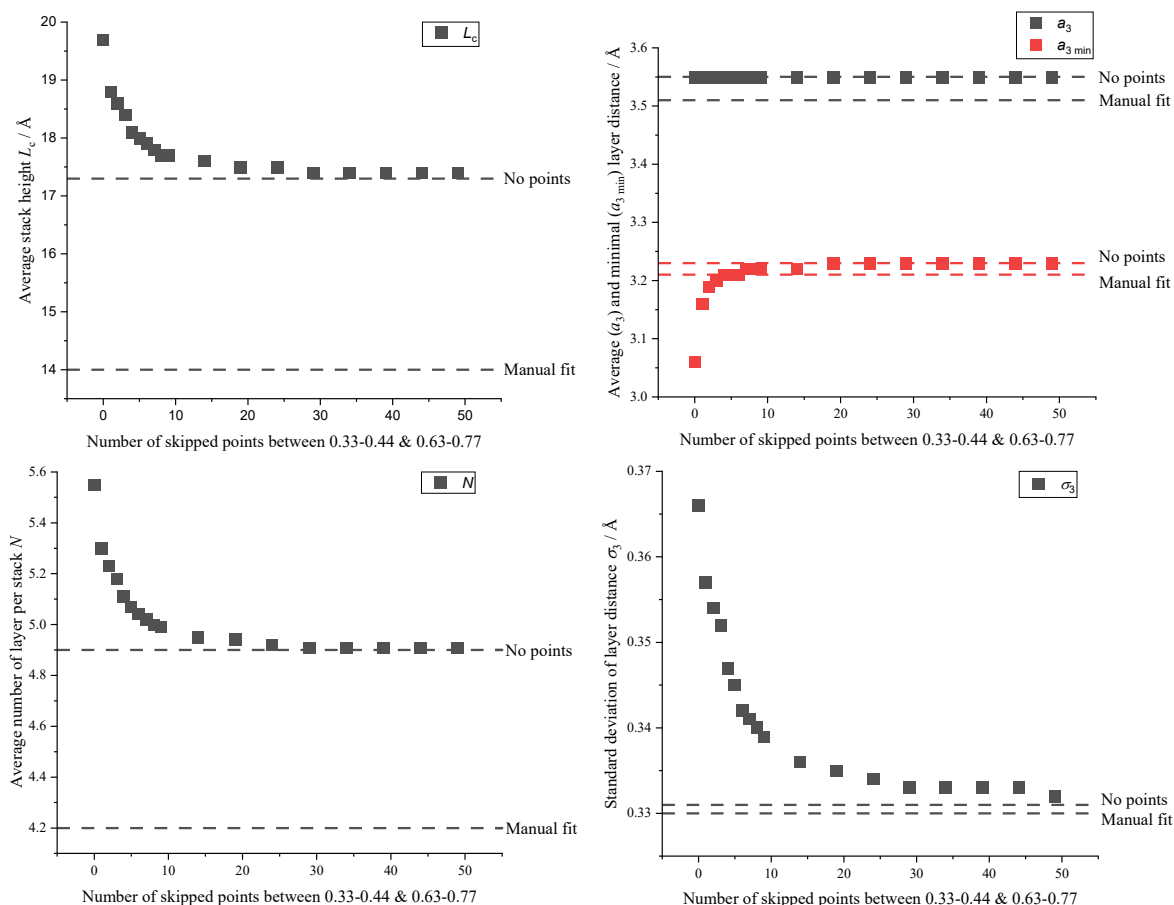
**Figure S3** Tests for the automatic *Octave* refinement with special attention to the (004)-region of WANS data from LSPP-1200 (low softening-point pitch). For all tests, only the orange area was fitted and the green parts extrapolated. Tests were performed with all parameters (red), the interlayer and intralayer parameters without the normalization (blue), the interlayer parameters with the normalization (green) and only with the interlayer parameters (purple). For the other refinements, look at Figure S1 or Figure S2.



**Figure S4** Tests for the automatic *Octave* refinement with special attention to the (004)-region of WANS data from LSPP-1200 (low softening-point pitch). For all tests, only the orange area was fitted and the green parts extrapolated. In order to determine the influence of the feet/valleys on the (004)-reflection, first all (red) and then only every 2<sup>nd</sup> (blue), every 5<sup>th</sup> (green), every 10<sup>th</sup> (purple) and at last no point (brown) were used for the refinement. For more details, take a look at Figure S5. It becomes clearly, that the high number of data points in the feet/valleys influence the refinement. Compared to the (004)-reflection, these green areas are in mathematical terms more important. Less refined points in the green areas lead to a higher accuracy in the (004)-reflection.



**Figure S5** Tests for the automatic *Octave* refinement with special attention to the (004)-region of WANS data from LSPP-1200 (low softening-point pitch). For all tests, in the orange area all points and in the green area less points were fitted. Hence, in the ranges of  $0.33 \text{ \AA}^{-1} < s < 0.44 \text{ \AA}^{-1}$  and  $0.63 \text{ \AA}^{-1} < s < 0.77 \text{ \AA}^{-1}$ , not all points were used. In order to determine the influence of the feet/valleys on the (004)-reflection, first all (red) and then only every 2<sup>nd</sup> (blue), every 5<sup>th</sup> (green), every 10<sup>th</sup> (purple) and at last no point (brown) were used for the refinement. It becomes clearly, that the high number of data points in the feet/valleys influence the refinement. Compared to the (004)-reflection, these green areas are in mathematical terms more important. Less refined points in the green areas lead to a higher accuracy in the (004)-reflection.



**Figure S6** Tests for the automatic *Octave* refinement with special attention to the (004)-region of WANS data from LSPP-1200 (low softening-point pitch). For all tests, only the green range in Figure S4 was fitted. Hence, in the range of  $0.33 \text{ \AA}^{-1} < s < 0.44 \text{ \AA}^{-1}$  and  $0.63 \text{ \AA}^{-1} < s < 0.77 \text{ \AA}^{-1}$ , not all points were used. In order to determine the influence of the feet/valleys on the (004)-reflection, some points were skipped. Evidently, a high number of data points in the feet/valleys influence the refinement. Compared to the (004)-reflection, these regions of the scattering curve (green domains in the figures above) are in mathematical terms more important. Less refined points in the green areas lead to a higher accuracy in the (004)-reflection and therefore, the interlayer microstructure parameters are different.

## S5. Download and usage of *iObs*

The currently used C++ code for calculating *iObs* including some auxiliary files for compiling an \*.oct file for *Octave* is available in the GitHub.<sup>10</sup>

### S5.1. Compilation of an \*.oct file for using *iObs* with *Octave*

An alternative way to use the calculations code in C++ is to compile it in an \*.oct file, which can be used from *Octave*. A video describing the next step is available in the GitHub<sup>11</sup> and at YouTube<sup>12,13</sup>. The basic idea is, that *Octave* uses some code, which is precompiled for different operating systems. This code can be used as an additional library for C++, so the algorithm of calculating  $I_{\text{obs}}$  can be converted in an \*.oct file on the currently used operating system. This \*.oct file in turn can be opened and used by *Octave* to calculate  $I_{\text{obs}}$  inside *Octave*. This method allows to calculate  $I_{\text{obs}}$  and therefore to refine measured WAXS/WANS data without the directly usage of any C++ code. Hence, the command *mkoctfile* implemented in *Octave* is used to perform this compilation. In principle, beside the pure calculation file, an additional file performing the in- and output operations, which is basically a “connector” between *Octave* and C++ code, is needed (*iObs.cpp*).

Since the compilation is unique for each operating system, it is highly recommended to compile the file on every different operating system. While the compilation is different for each operating system, only a brief overview of the command and its usage can be given. A more detailed instruction how to build an \*.oct file explaining these steps as well as some example files are available in the GitHub.<sup>14</sup>

For Microsoft Windows, two commands must be executed, whereas the paths must be adjusted, obviously:

```
cd C:\Octave\Octave-5.2.0\mingw64\bin
```

```
C:\Octave\Octave-5.2.0\mingw64\bin\mkoctfile -LC:\Octave\Octave-5.2.0\mingw64\lib\Octave\5.2.0 -
IC:\Octave\Octave-5.2.0\mingw64\include\Octave-5.2.0\Octave 'C:\iObsOct\iObs.cpp'
```

For MacOS and Linux, only one command must be executed:

```
MacOS: mkoctfile -I/usr/local/bin/Octave ~/iObsOct/iObs.cpp
```

```
Linux: mkoctfile -I/usr/include/Octave-5.2.0/octave ~/iObsOct/iObs.cpp
```

The exact path for the include files can be found out using the following command:

```
which Octave
```

<sup>10</sup> <https://github.com/ossvaldo/NGCs/blob/master/Octave/oct-files/Self-compilation/iObsOct/calculations.cpp>

<sup>11</sup> <https://github.com/ossvaldo/NGCs/tree/master/Instruction%20Videos>

<sup>12</sup> English: [https://www.youtube.com/playlist?list=PLTlnYDX5g1FylWfH8cSM\\_ZQUQpn3dSp6M](https://www.youtube.com/playlist?list=PLTlnYDX5g1FylWfH8cSM_ZQUQpn3dSp6M)

<sup>13</sup> German: <https://www.youtube.com/playlist?list=PLTlnYDX5g1FwaOQkLXTxrpvHW9CRUwBrk>

<sup>14</sup> <https://github.com/ossvaldo/NGCs/tree/master/Octave/oct-files>

## S6. Installation and updates

A video describing the next step is available in the GitHub<sup>15</sup> and at YouTube<sup>16, 17</sup>.

### S6.1. Installation under *Microsoft Windows*

For Microsoft Windows, an executable installation package can be downloaded directly from the developer website<sup>18</sup>. Afterwards the *optim* package including all needed dependencies can be installed using “pkg install -forge optim”.

### S6.2. Installation under *MacOS*

For MacOS, no official-maintained installer or bundle is available, but, however, there are some \*.app builds available from the developer<sup>19</sup>, for newer versions, an installation guide can be found in the developer instructions.<sup>20</sup>

### S6.3. Installation under *GNU/Linux, BSD* and other systems

For all other operating systems like *Linux*, *BSD* and distribution independent approaches like *Docker* it is impossible to give a brief installation guide due to the high number of different systems. In general, all guides can be found in the developer instructions.<sup>21</sup>

### S6.4. Installation of *optim* for non-*Windows* builds

For some operating systems/installations (if *pkg install -forge optim* fails) the *optim* package and its dependencies must be installed manually using console commands. First the *optim* package and its dependencies (*structs*, *statistics* and *io*) must be downloaded from *sourceforge*<sup>22</sup>. Second, open *Octave* and navigate to the download directory, e.g. “cd C:\Users\<Username>\Downloads“. In the last step, install *optim* and its dependencies (order as followed):

```
pkg install io-<version>.tar.gz
```

```
pkg install statistics-<version>.tar.gz
```

```
pkg install struct-<version>.tar.gz
```

```
pkg install optim-<version>.tar.gz
```

### S6.5. Links and updates

---

<sup>15</sup> <https://github.com/ossvaldo/NGCs/tree/master/Instruction%20Videos>

<sup>16</sup> English: [https://www.youtube.com/playlist?list=PLTlnYDX5g1FylWfH8cSM\\_ZQUQpn3dSp6M](https://www.youtube.com/playlist?list=PLTlnYDX5g1FylWfH8cSM_ZQUQpn3dSp6M)

<sup>17</sup> German: <https://www.youtube.com/playlist?list=PLTlnYDX5g1FwaOQkLXTxrpvHW9CRUwBrk>

<sup>18</sup> <https://www.gnu.org/software/octave/download>

<sup>19</sup> <https://octave-app.org/>

<sup>20</sup> [https://wiki.octave.org/Octave\\_for\\_MacOS](https://wiki.octave.org/Octave_for_MacOS)

<sup>21</sup> <https://wiki.octave.org/Category:Installation> and [https://wiki.octave.org/Octave\\_for\\_GNU/Linux](https://wiki.octave.org/Octave_for_GNU/Linux)

<sup>22</sup> <https://octave.sourceforge.io/packages.php>

Due to the number of different programs, extensions and scripts that are used and the resulting update frequency that can be expected from them, it does not make sense to describe the update process in this static work. Current links, installation files can be found in the file “Useful links.txt” and scripts can be found in the GitHub.<sup>23</sup>

---

<sup>23</sup> <https://github.com/ossvaldo/NGCs>



## S7. Usage of *Octave*

After loading the refinement script, only a few adjustments are necessary to perform the first fit. Stepwise, consecutive files for the next steps below can be found in the other SI files. A video explaining these steps is available in the in the GitHub<sup>24</sup> and at YouTube<sup>25, 26</sup>.

0. Download and install *Octave*, *iObs* and the refinement script (see above).
1. First, some initial parameters like the sample name, data path, radiation type, wavelength etc. must be adjusted (Figure S7). Additionally, a meaningful for the parameter *nSkip* should be set. *nSkip* = 5 means, that only every 5<sup>th</sup> point will be calculated to make the calculations much faster, less accurate at the same time. The value should be higher the more measurement points are available. Also, a constant  $\nu$  must be chosen. For a common XRD measurement using Cu-K $\alpha$  radiation in a range of  $10^\circ < 2\theta < 140^\circ$ , a value of  $\nu = 4$  is sufficient for most of the samples (see Table S2).
2. As the first real refinement step, the normalization parameters ( $k$ ,  $const_1$ ,  $const_2$ ) and the concentrations of foreign atoms ( $c_H$ ,  $c_N$ ,  $c_O$ ,  $c_S$ ) should be refined. While the last ones are known exactly from elemental analysis, the normalization parameters must be refined by hand. For the automatic adjustment, it can be done only roughly, but for the manual adjustment, it should be done more exactly. The normalization constants should be adjusted at every of the following steps.

### Manual refinement:

- 3a. Generally, the manual refinement is divided into 3 parts: refinement of the interlayer structure, refinement of the intralayer structure and a final step with all parameters including the normalization. For the interlayer parameters, first  $\mu$  or  $\beta$  can be left constant and only one of it should be varied. Also,  $a_{3\min}$  should not be lower than 3 Å, otherwise it will dominate the whole curve too much, because the set value is no longer physically meaningful.  $\sigma_3$  can be used to refine the broadness of the (002) and (004) reflections and has a high influence on the damping of the (004) reflection. If the (002) reflection is refined approximately, both  $\mu$  and  $\beta$  should be further refined. It should be noted that the stack height  $L_c \sim (\mu+1)/\beta$  should no longer vary too much. In this step,  $\eta$  (homogeneity of the stacks) can be refined by taking the left side of the (002) reflection and lower values of the scattering vector/angle into account.  $q$  (preferred orientation) should be left constant in this step, because without a refinement of the intralayer structure, it is not possible to refine this parameter meaningfully. Figure 6 in the main article shows an exemplary WAXS pattern and a manually adjusted result that contains the priority of the various regions.
- 4a. For the most samples, this step is easier to perform than the refinement of the interlayer parameters due to the lower number of parameters. Since  $\nu$  is chosen as a constant, only  $\alpha$ ,  $\sigma_1$  and  $l_{cc}$  must be refined.  $\alpha$  influences both, the height and the broadness of the (10) and the (11) reflection,  $\sigma_1$  influences more the

<sup>24</sup> <https://github.com/ossvaldo/NGCs/tree/master/Instruction%20Videos>

<sup>25</sup> English: [https://www.youtube.com/playlist?list=PLTlnYDX5g1FylWfH8cSM\\_ZQUQpn3dSp6M](https://www.youtube.com/playlist?list=PLTlnYDX5g1FylWfH8cSM_ZQUQpn3dSp6M)

<sup>26</sup> German: <https://www.youtube.com/playlist?list=PLTlnYDX5g1FwaOQkLXTxrpvHW9CRUwBrk>

(11) reflection. Using  $l_{cc}$ , the exactly position of both reflections can be refined. For some samples, the preferred orientation  $q$  must be refined at this step.

- 5a. As the last step for the manual refinement, all microstructural parameters including the normalization should be adjusted. The exact procedure depends strongly on the sample and can therefore not be described in general here. In general, the accuracy should focus on the valley between the (002) and the (10) reflection rather than on the values that lie left to the (002) reflection. For the (10) and (11) reflections it is important to assure that the ratio of the size and shape in relation to the background is correct. The (004) reflection needs to be adjusted precisely, even if it sometimes just becomes like a shoulder. The accuracy of the refinement of this reflection significantly influences the accuracy of the resulting disorder parameters of the stack structure (see Figure 6 in the main article).

#### Automatic refinement:

- 3b. For the basic automatic refinement, the default values from Table S7 can be used. These values are in principle mean values from the common range of these values. Of course, if other suitable start values are known, they should be used (i.e., for measurements from a temperature series). To start the automatic refinement, the switch *plotOnly* must be set to *false*. For some samples or measurements, the upper and lower bonds for the preferred orientation ( $q$ ) must be adjusted to a range from 0 to 1 (ub3 & lb3), otherwise, a preferred orientation will not be refined.
- 4b. As a final refinement step, the parameter *nSkip* should set to 1. Now, every measured point will be refined, and the fitting result might improve. As start values, the values from the 3<sup>rd</sup> step (3a or 3b) should be used. In addition, the calculated error for the microstructure parameters will become a little bit lower.
- 5b. As a last step, an additional manual adjustment can be performed. This step is often necessary for more disordered samples, where the (004) reflection is damped and broad.

**Table S7** Overview about the default starting values for the refinement with *Octave*.

These values are in principle mean values of the common range of the fitting values and can be regarded as recommendations. Of course, every additional information like results from an elemental analysis should be used to choose as best start values as possible. The *microstructural parameters* are partly calculated from the *parameter for Octave*.

Parameter for <i>Octave</i>	Value
$\mu$	4
$\beta$	0.5
$\bar{a}_3 / \text{\AA}$	3.5
$da_3 / \text{\AA}$	0.4

$\sigma_3 / \text{\AA}$	0.25
$u_3$	0
$\eta$	1
$\nu$	4
$\alpha$	0.2
$l_{\text{cc}} / \text{\AA}$	1.412
$\sigma_1$	0.1
$q$	0
$\Delta_{\text{an}}$	0
$k$	500
$\text{const}_1$	0
$\text{const}_2$	0

---

Microstructural parameter	Value
$L_a / \text{\AA} (= (\nu + I) / \alpha)$	25
$l_{\text{m}} / \text{\AA} (= \nu / \alpha)$	20
$\kappa_{\text{a}} / \text{\AA} (= 1/\nu)$	0.25
$l_{\text{cc}} / \text{\AA}$	1.412
$\sigma_1$	0.1
$L_{\text{c}} / \text{\AA} (= (\mu + I) / \beta \cdot \bar{a}_3 = N \cdot \bar{a}_3)$	35
$\kappa_{\text{c}} / \text{\AA} (= 1/\beta)$	0.25
$\bar{a}_3 / \text{\AA}$	3.5
$a_{3 \text{ min}} / \text{\AA}$	3.1
$\sigma_3 / \text{\AA}$	0.25
$N (= (\mu + I) / \beta = N)$	10
$u_3$	0
$\eta$	1
$q$	0
$\Delta_{\text{an}}$	0

```

cno = 0.0; #Concentration of disordered sp3 carbon
ch  = 0.0; #Concentration of hydrogen
cn  = 0.0; #Concentration of nitrogen
co  = 0.0; #Concentration of oxygen
cs  = 0.0; #Concentration of sulfur

#Name of the series and id of the sample, filename of current used file
name = "WAXS example";
global id = "WAXS example Step 1";
filename = "WAXS Step 1 - path and data adjustments.m";
path = 'D:/OneDrive/Uni/PhD/Paper Octave/Github/NGCs/examples/WAXS Steps';
cd 'D:/OneDrive/Uni/PhD/Paper Octave/Github/NGCs/examples/WAXS Steps';
#Measurement data file
measFile = 'D:/OneDrive/Uni/PhD/Paper Octave/Github/NGCs/examples/WAXS Steps/WAXS example data.xy';

neutronCorrection = false; #Corrections for Wide-Angle Neutron Scattering (WANS) experiments.
neutronCorrectionVoigt = false; #Useful for samples containing hydrogen.

#Wavelength and type of radiation (0 = X-ray, 1 = neutrons)
wavelength = 1.5418;
radiation = 0;

#Type of x-values: theta, thetaRad, twoTheta, twoThetaRad, scatS, scatQ
type = "twoTheta";

useQ      = false; #Use Additional Debye-Waller-factor
b         = 0.002; #Debye-Waller-factor Factor
useA      = true;   #Absorption correction
density   = 2.2;    #Density of sample in g/cm^3
sampleThickness = 0.3; #Sample thickness in cm
transmission = false; #Transmission geometry (if false, reflection geometry is assumed)
absorptionCorrection = 1; #Correction factor for absorption coefficient (multiplier)
useP      = true;   #Polarization correction
polarizedBeam = false; #Do you use a polarized beam?
polarizationDegree = 0; #Polarization direction of beam in °.
useCorrAutoColl = false; #Slit-correction
par_r      = 14;    #Radius of the diffractometer (Debye-Scherrer) in cm
par_delta  = 4;     #Divergence angle in ° (it is converted as if this fixed slit were inside)
par_l      = 5;     #Irradiated length in cm

```

**Figure S7** Overview about the basic instrumental parameters, which should be checked and changed for every different measurement.

## S8. Example refinement

All files described in this section are also available under in the GitHub<sup>27</sup> and in the file “Example refinement.zip”.

### S8.1. Step 0 – initial situation

First, you have to download and install *Octave* and an *iObs* file as described below. Depending on your system, you might have to compile the *iObs* file by your own. For this case, the Numerical Recipes for C must be downloaded<sup>28</sup>. Generally, the lines 89 – 134 should be checked for each measurement or measurement group. In these lines, some parameters like the polarization or absorption as well as mathematical fit parameters like the function tolerance, upper and lower bounds and the weight can be tweaked. The lines 1 – 134 of the initial (downloaded) refinement scripts (WAXS Fit-Routine-IUCr.m) are pasted below and are identical to refinement script in the zip folder (S8.10). In general, all other lines from 136 onwards should not be changed. The final modified script is also shown in S8.11.

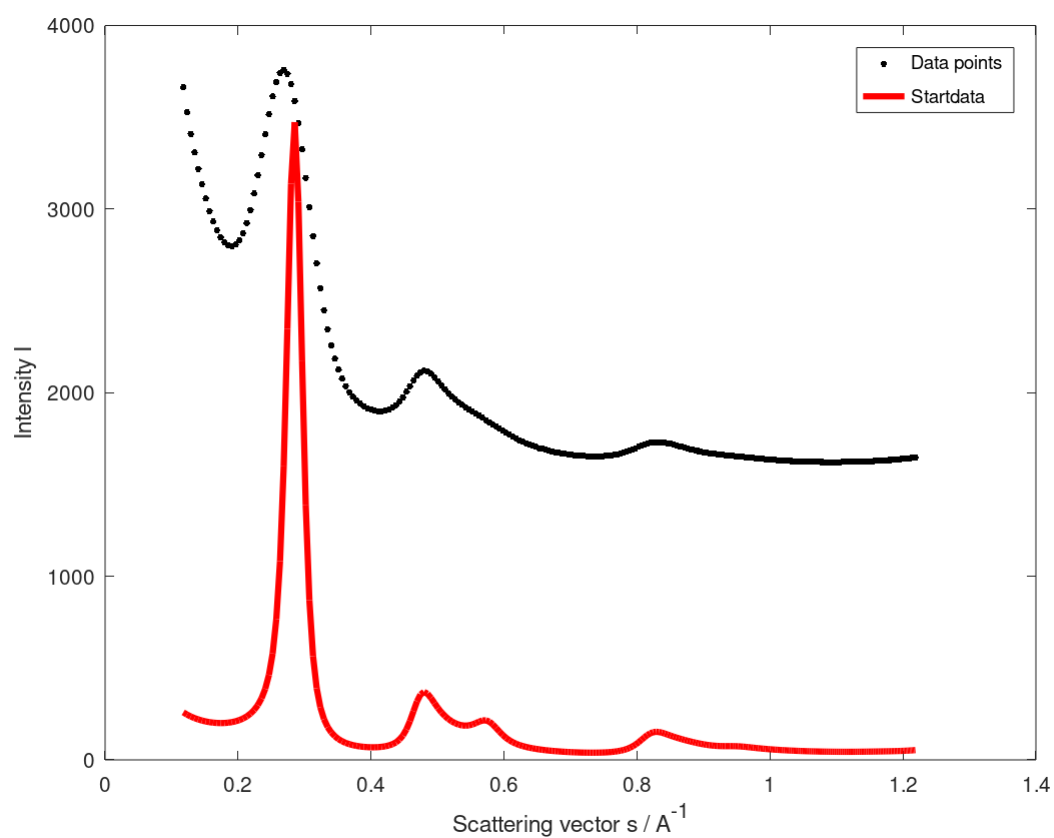
### S8.2. Step 1 – path and data adjustments

As first steps,  $\nu$  (line 14) must be tweaked. In general, for a common XRD using Copper-radiation (1.54 Å),  $\nu = 4$  is sufficient. If you use another wavelength, you can look at Table S2 to get a suitable value for  $\nu$ . Additionally, the lines 41 – 87 must be checked. “name” means the name of your sample series or the current date or whatever you want to choose. The resulting refinements will be saved in this directory. The “global id” is a unique name for your current sample or step. You can also use a consecutive number to not override old refinements. “filename” must be the name of the currently used refinement scripts and can be found on top of the editor. “<path\_to\_filename>” must be replaced by the path, where the “filename” AND the *iObs.oct* file is placed. The *iObs.oct* file must be in the same directory as the refinement script. The complete path of your measurement file (x, y without headers) is stored in the variable “measFile”. If you use neutron scattering or another wavelength, you must check the lines 54, 57, 60 and 61, otherwise you can skip these lines. If you want to plot only the coherent or incoherent scattering, you can change the values in the lines 64 and 65. Line 74 (“type”) is important if you are not using  $2\theta$  in ° as x-values. The parameters “nStart”, “nEnd”, “nSkip” and “nUp” can be used to skip some points at the beginning or the end or to plot only every  $n^{\text{th}}$  point (“nSkip”). To prevent negative values of the intensity, you can use “nUp” to add a constant background to move the WAXS/WANS pattern up.

Regarding the measurement geometry, there are some correction terms, you can use (absorption, polarization and variable slit, lines 121 – 134). These parameters are set to Bragg-Brentano geometry using an unpolarized incidence beam and absorption correction for a 3 mm graphite sample. You should check the parameters “density” and “sampleThickness” in lines 122 and 123 and change it to the values of your samples. It is not necessary to know the exact values, but you should put in the correct tendency.

<sup>27</sup> <https://github.com/ossvaldo/NGCs/tree/master/Example%20refinement/WAXS%20Steps>

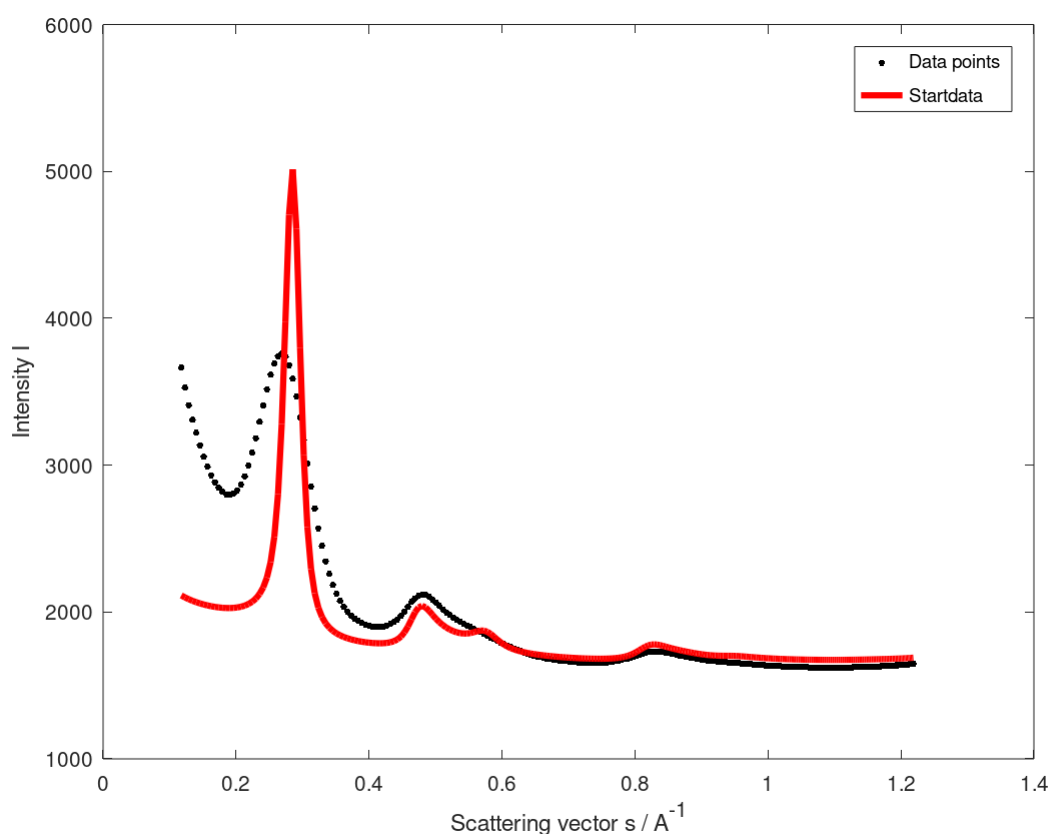
<sup>28</sup> <http://numerical.recipes/com/storefront.html>



**Figure S8** Step 1 - path and data adjustments without any refinements of the normalization or the microstructure.

### S8.3. Step 2 – manual background

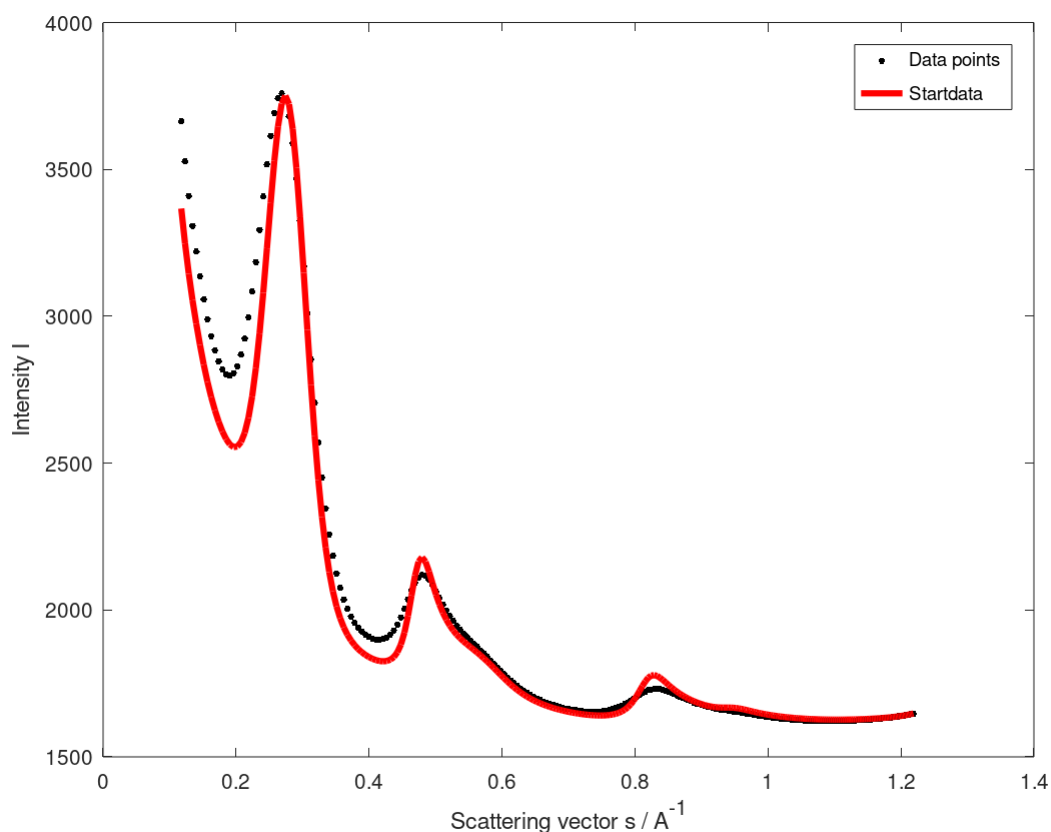
Second, the background must be corrected in order to start to refine the microstructure. “k” is a normalization constant depending on the amount of the sample, intensity of the radiation etc. and “const1” is used a constant offset of the scattering intensity caused by several effects like cosmic background radiation, incoherent scattering by your sample holder and other general measurement effects. The concentrations of foreign atoms (in atomic %) of hydrogen, nitrogen, oxygen and sulfur known from an elemental analysis can be considered in the lines 29 – 32 (parameters “cH”, “cN”, “cO” and “cS”).



**Figure S9** Step 2 – manual background. Only “k” and “const1” are refined.

### S8.4. Step 3a – manual fitting – interlayer

To start the refinement of the microstructure, you should start with the interlayer scattering, but you can also start with the intralayer scattering (Step 4a). For the interlayer scattering, the parameters “mu”, “beta”, “a3”, “da3”, “sig3” and “eta” must be refined. In addition, “q” (preferred orientation), “k” and “const1” must be refined parallel (lines 7 – 22). The important and significant parts of the WAXS data are shown in Figure 6 in the main article. “mu” and “beta” are parameters to refine the stack height ( $L_c = (\mu+1)/\beta \cdot \bar{a}_3$ ) and polydispersity ( $\kappa_c = 1/\mu$ ), “a3” is the average layer distance ( $\bar{a}_3$ ), “da3” is the difference between the average and the minimal layer distance ( $a_{3\min} = \bar{a}_3 - da_3$ ), “sig3” the standard deviation of it ( $\sigma_3$ ) and eta means the homogeneity of the stacks ( $\eta$ ).

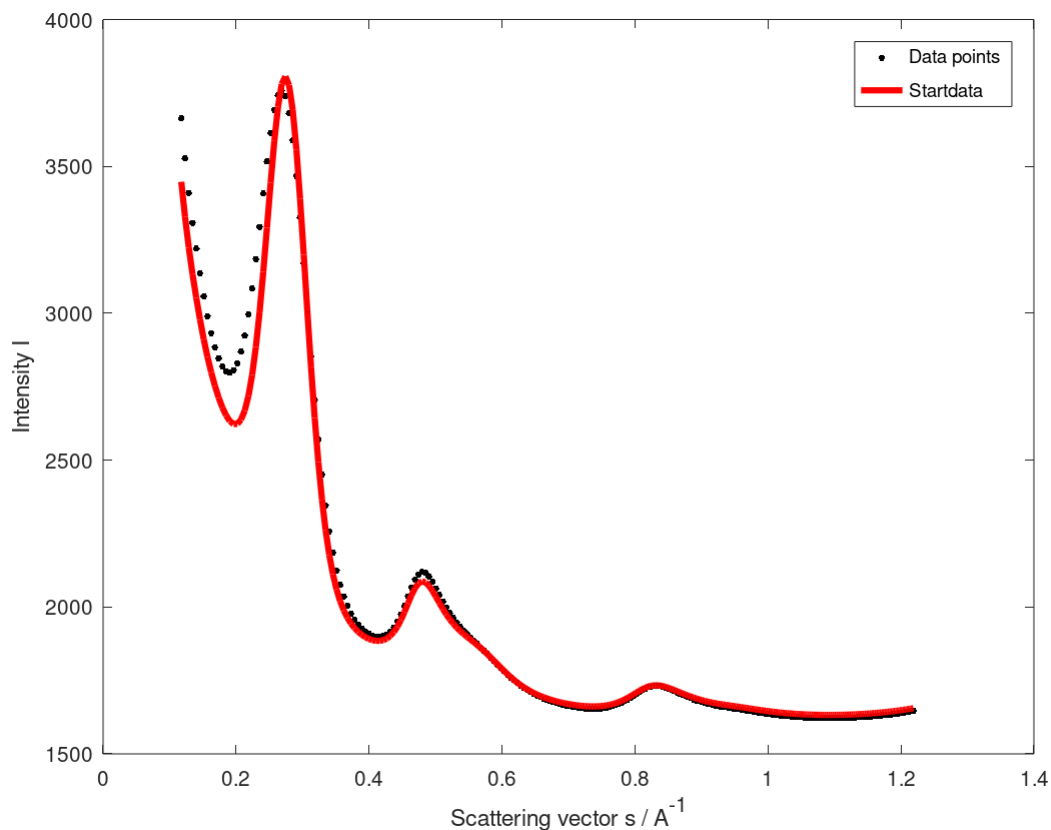


**Figure S10** Step 3a – manual fitting – interlayer. The microstructure parameters for the interlayer scattering (“mu”, “beta”, “a3”, “da3”, “sig3” and “eta”) and “q” (preferred orientation) as well as “k” and “const1” were refined.



### S8.5. Step 4a – manual fitting – intralayer

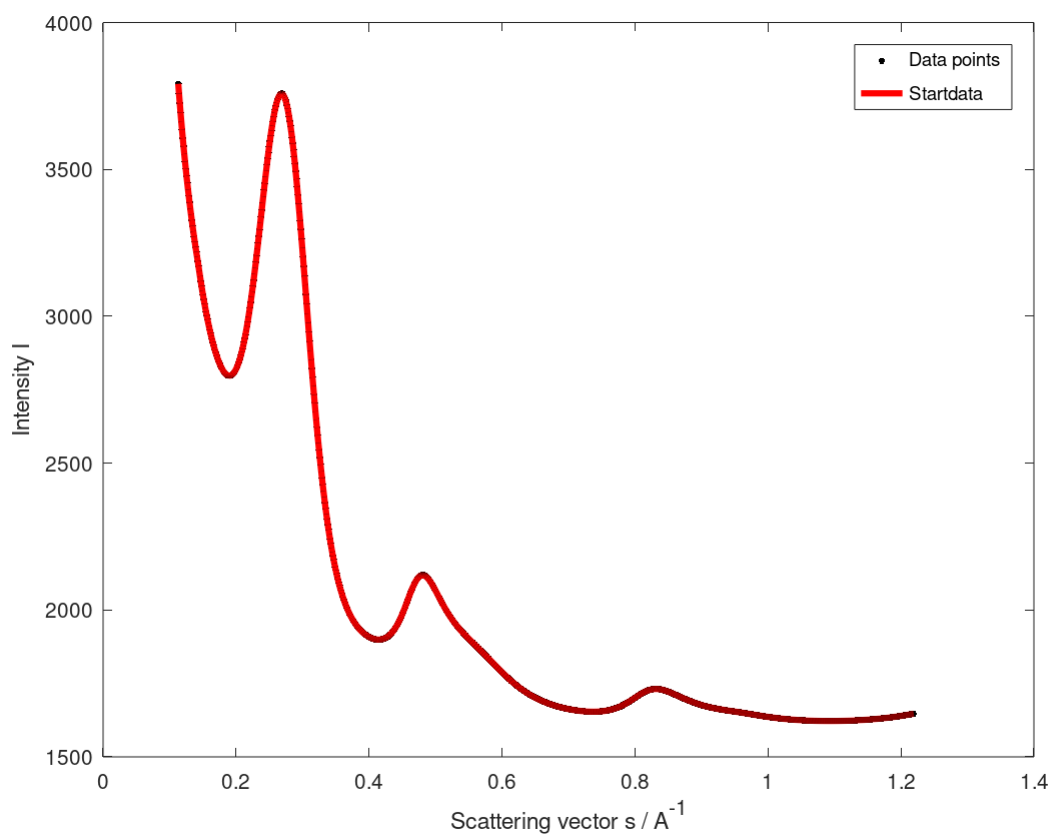
Beside the normalization, the parameters “alpha”, “lcc” and “sig1” (lines 15 – 17) must be refined in this step. Analogous to the interlayer scattering, “alpha” is used to calculate the average layer extension ( $L_a = (v+1)/\alpha$ ),  $l_{cc}$  is the average bond length and  $\sigma_1$  the disorder of the layers (i.e. stress and strain).



**Figure S11** Step 4a – manual fitting – interlayer. The microstructure parameters for the intralayer scattering (“alpha”, “lcc”, “sig1”) and “q” (preferred orientation) as well as “k” and “const1” were refined.

**S8.6. Step 5a – manual fitting – all**

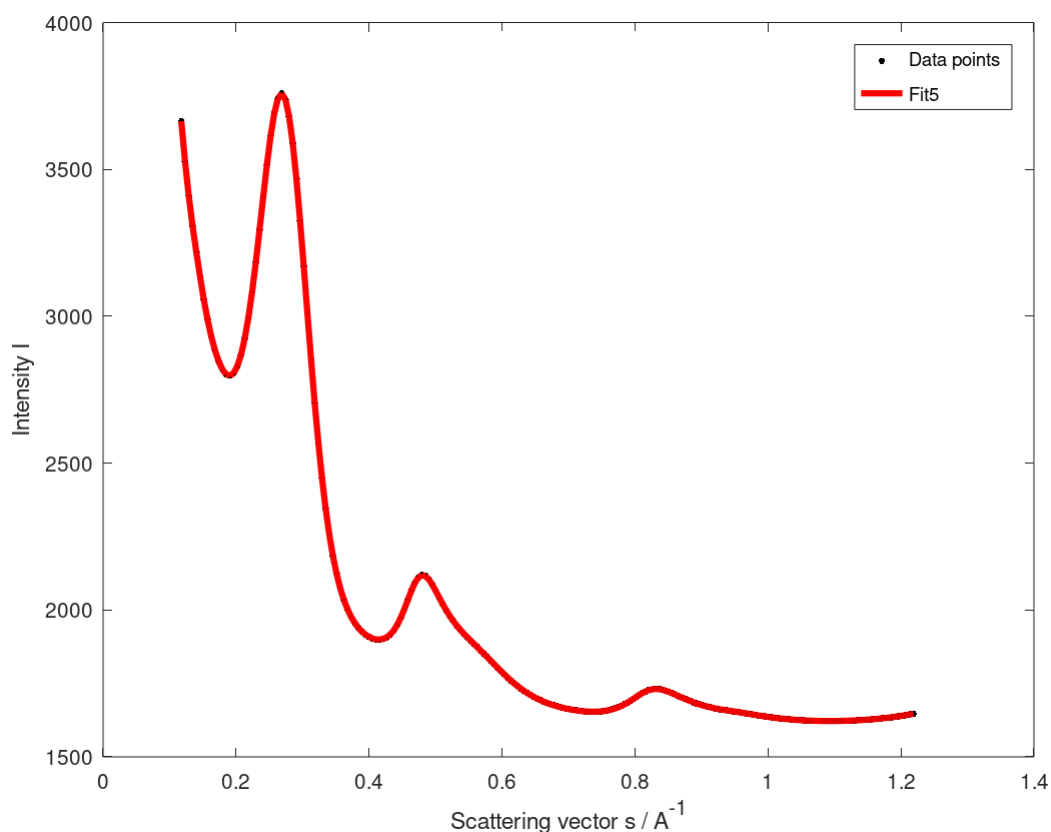
For the final step of the manual fitting, you should set “nSkip” to 0 (line 84). Afterwards, try to refine all parameters as described above. You should start with the normalization constants,  $q$  and  $\eta$ . As an alternative, you can use these parameters as new start parameters for step 2. In principle, you should repeat the steps 2 – 5 multiple times to get a result, which is as good as possible.



**Figure S12** Step 5a - manual fitting – all. In this step, all microstructure and normalization parameters were refined at once.

### S8.7. Step 3b – automatic fitting – initialization

To make things easier, you do not have to refine the scattering data manually. There is an automatically fitting routine, which can fit the scattering data and give out the microstructure data. To use this method, you need to roughly refine the background (step 2) set “plotOnly” to “false” (line 35). Basically, that is all what you have to do (of course, you have to press the “run” button on top of the *Octave* GUI). Beside this, you can set the lower and upper bonds for the different microstructure parameters (89 – 104). This might be useful, if you have a temperature series and some parameters have to increase or decrease, but in general, it is not necessary to use it.



**Figure S13** Step 3b – result of the automatic refinement. No further refinement must be done.

### S8.8. Step 4b – automatic fitting - fine adjustment

If everything works well, you do not need this step (like in this example). For other samples or insufficient start values or a too bad refinement of the normalization in step 2, you have to make a manual refinement after the automatically fit. To do so, change “plotOnly” back to “true” and perform the steps 3a – 5a. As an alternative, use the resulting parameters as new start parameters and repeat step 3b.

### S8.9. Common warnings/errors during the refinement and how to fix them

A list of common errors and their solutions can be found in the GitHub<sup>29</sup> and in the SI under Octave/README.pdf.

### S8.10. Initial refinement script lines 1 – 134

```
1 clear;
2
3 timeStart = time();
4
5 #Initial values for fitting. k & const1 should be roughly adjusted manually. nu depends on the experiment, for
  Cu-radiation, 4 is sufficient. Mostly, g can be 0 (switch with useGradient = true/false).
6 u3      = 0; #Generally not refined and not outputted
7 mu      = 4;
8 beta    = 0.5;
9 a3      = 3.5;
10 da3     = 0.4;
11 a3min   = a3-da3;
12 sig3    = 0.25;
13 eta     = 1;
14 nu      = 4;
15 alpha   = 0.2;
16 lcc     = 1.412;
17 sig1    = 0.1;
18 q       = 0;
19 dan     = 0;
20 k       = 500;
21 const1  = 0;
22 const2  = 0;
23
24 #Switch for usage of gradient g and concentrations of impurities
```

<sup>29</sup> <https://github.com/ossvaldo/NGCs>

```
25 useGradient = false;
26 g          = 0;
27
28 cno        = 0.0; #Concentration of disordered sp3 carbon
29 cH         = 0.0; #Concentration of hydrogen
30 cN         = 0.0; #Concentration of nitrogen
31 cO         = 0.0; #Concentration of oxygen
32 cS         = 0.0; #Concentration of sulfur
33
34 #Switch for show a plot using the values above. Works only, if "shouldPlot = true".
35 plotOnly = true;
36
37 #Graphical output (has to be "false" if using octave-cli). ('global' can be ignored, but must be present, it is
   necessary)
38 #Global variables can only be resetted by restarting Octave.
39 global shouldPlot = true;
40
41 #Name of the series and id of the sample
42 name = "name";
43 global id = "id";
44 #Filename and path the currently used file, must also contain iObs.oct. The path must be changed twice.
45 filename = "filename.m";
46 #The '/' symbol must be used in the paths
47
48 path = '<path_to_filename>';
49 cd '<path_to_filename>';
50
51 #Measurement data file
52 measFile = '<path_to_measurement_file>';
53
54 #Corrections for Wide-Angle Neutron Scattering (WANS) experiments, only meaningful, if radiation = 1
   (means neutrons scattering)
55 neutronCorrection = false;
56
57 #Useful for samples containing hydrogen. Using this method, a Pseudo-Voigt function will be used to fit the
   background instead of a polynomial. If false, the Placzek correction ( $a \cdot x^2 + b$ ) will be used for the
   background determination.
58
59 neutronCorrectionVoigt = false;
60
61 #Wavelength and type of radiation (0 = X-ray, 1 = neutrons)
62 wavelength = 1.5418;
```

```
61  radiation = 0;
62
63  #Calculate coherent and/or incoherent radiation. Incoherent radiation is only considered for X-ray radiation
64  coh = true;
65  inc = true;
66
67  #Type of x-values:
68  #theta = Scattering angle theta in
69  #thetaRad = Scattering angle theta in rad.
70  #twoTheta = Scattering angle 2 theta in
71  #twoThetaRad = Scattering angle 2 theta in rad.
72  #scatS = Scattering vector s = 2*sin(theta)/wavelength
73  #scatQ = Scattering vector q = 2*Pi*s
74  type = "twoTheta";
75
76  #Octave must be restarted after changing one of the below lines
77  #Skip n points at start
78  nStart = 0;
79
80  #Skip n points at the end
81  nEnd = 0;
82
83  #Calculate only every n point
84  nSkip = 1;
85
86  #Shifting the raw data up/down
87  nUp = 0;
88
89  #Lower and upper bounds for fitting parameters (can be changed, but does not have to be)
90  #mu, beta, a3, da3, sig3, eta
91  lb1 = [0.01; 0.01; 2.50; 0; 0; 0];
92  ub1 = [10; 20; 5.00; 1; 1; 1];
93
94  #nu, alpha, sig1
95  lb2 = [nu; 0.01; 0];
96  ub2 = [nu; 2; 2];
97
98  #q, dan, k, const1, const2, g
99  lb3 = [q; dan; 0.0001; -100000; const2; -1];
```

```

100  ub3 = [q; dan; 10000; +100000; const2; +1];
101
102  #lcc
103  lb4 = [1.2];
104  ub4 = [1.8];
105
106  #Function tolerance
107  tolFun = 1e-10;
108
109  #Maximal iterations per fit step
110  maxIter = 50;
111
112  #Weight of measurement points
113  #normal = 1 (every point has the same weight)
114  #weight = 1/y (default; lower intensity has lower weight)
115
116  weight = "weight";
117
118  useQ = false;           #Additional Debye-Waller-factor
119  b   = 0.002;           #Factor
120
121  useA   = true;          #Absorption correction
122  density = 2.2;          #Density of sample in g/cm^3
123  sampleThickness = 0.3;  #Sample thickness in cm
124  transmission = false;  #Transmission geometry (if false, reflection geometry is assumed)
125  absorptionCorrection = 1; #Correction factor for absorption coefficient (multiplied to theoretical
                             coefficient)
126
127  useP   = true;          #Polarization correction
128  polarizedBeam = false;  #Do you use a polarized beam?
129  polarizationDegree = 0;  #Polarization direction of beam in °.
130
131  useCorrAutoColl = false; #Slit correction
132  par_r           = 14;    #Radius of the diffractometer (Debye-Scherrer) in cm
133  par_delta       = 4;    #Divergence angle in ° (it is as if this fixed slit were inside)
134  par_l           = 5;    #Irradiated length in cm

```

#### S8.11. Refinement script lines 1 – 134 after the manual fitting

```
1 clear;
```

```
2
3  timeStart = time();
4
5  #Initial values for fitting. k & const1 should be roughly adjusted manually. nu depends on the experiment, for
   Cu-radiation, 4 is sufficient. Mostly, g can be 0 (switch with useGradient = true/false).
6  u3 = 0; #Generally not refined and not outputted
7  mu    = 8;    # Step 3a
8  beta  = 3;    # Step 3a
9  a3     = 3.5;  # Step 3a
10 da3    = 0.6;  # Step 3a
11 a3min   = a3-da3;
12 sig3    = 0.3;  # Step 3a
13 eta     = 0.87; # Step 3a
14 nu      = 4;    # Step 1
15 alpha   = 0.3;  # Step 4a
16 lcc     = 1.412; # Step 4a
17 sig1    = 0.15; # Step 4a
18 q       = 0;    # Step 3a
19 dan     = 0;
20 k       = 780;  # Step 2 # Step 3a # Step 4a
21 const1  = 1500; # Step 2 # Step 3a # Step 4a
22 const2  = 0;
23
24 #Switch for usage of gradient g and concentrations of impurities
25 useGradient = false;
26 g          = 0;
27
28 cno       = 0.0; #Concentration of disordered sp3 carbon
29 cH        = 0.0; #Concentration of hydrogen # Step 2 (known from elemental analysis)
30 cN        = 0.0; #Concentration of nitrogen # Step 2 (known from elemental analysis)
31 cO        = 0.05; #Concentration of oxygen # Step 2 (known from elemental analysis)
32 cS        = 0.02; #Concentration of sulfur # Step 2 (known from elemental analysis)
33
34 #Switch for show a plot using the values above. Works only, if "shouldPlot = true".
35 plotOnly = true;
36
37 #Graphical output (has to be "false" if using octave-cli). ('global' can be ignored, but must be present, it is
   necessary)
38 #Global variables can only be resetted by restarting Octave.
39 global shouldPlot = true;
```



```
40
41 #Name of the series and id of the sample
42 name = "Example";           # Step 1
43 global id = "Example";      # All steps
44 #Filename and path the currently used file, must also contain iObs.oct. The path must be changed twice.
45 #The '/' symbol must be used in the paths
46 filename = "Example.m";     # All steps
47 path = 'D:/Example';        # Step 1
48 cd 'D:/Example';            # Step 1
49
50 #Measurement data file
51 measFile = 'D:/Example.xy';  # Step 1
52
53 #Corrections for Wide-Angle Neutron Scattering (WANS) experiments, only meaningful, if radiation = 1
  (means neutrons scattering)
54 neutronCorrection = false;
55
56 #Useful for samples containing hydrogen. Using this method, a Pseudo-Voigt function will be used to fit the
  background instead of a polynomial. If false, the Placzek correction ( $a \cdot x^2 + b$ ) will be used for the
  background determination.
57 neutronCorrectionVoigt = false;
58
59 #Wavelength and type of radiation (0 = X-ray, 1 = neutrons)
60 wavelength = 1.5418;        # Step 1
61 radiation = 0;               # Step 1
62
63 #Calculate coherent and/or incoherent radiation. Incoherent radiation is only considered for X-ray radiation
64 coh = true;
65 inc = true;
66
67 #Type of x-values:
68 #theta = Scattering angle theta in °
69 #thetaRad = Scattering angle theta in rad.
70 #twoTheta = Scattering angle 2 theta in °
71 #twoThetaRad = Scattering angle 2 theta in rad.
72 #scatS = Scattering vector  $s = 2 \cdot \sin(\theta) / \text{wavelength}$ 
73 #scatQ = Scattering vector  $q = 2 \cdot \pi \cdot s$ 
74 type = "twoTheta";          # Step 1
75
76 #Octave must be restarted after changing one of the below lines
```

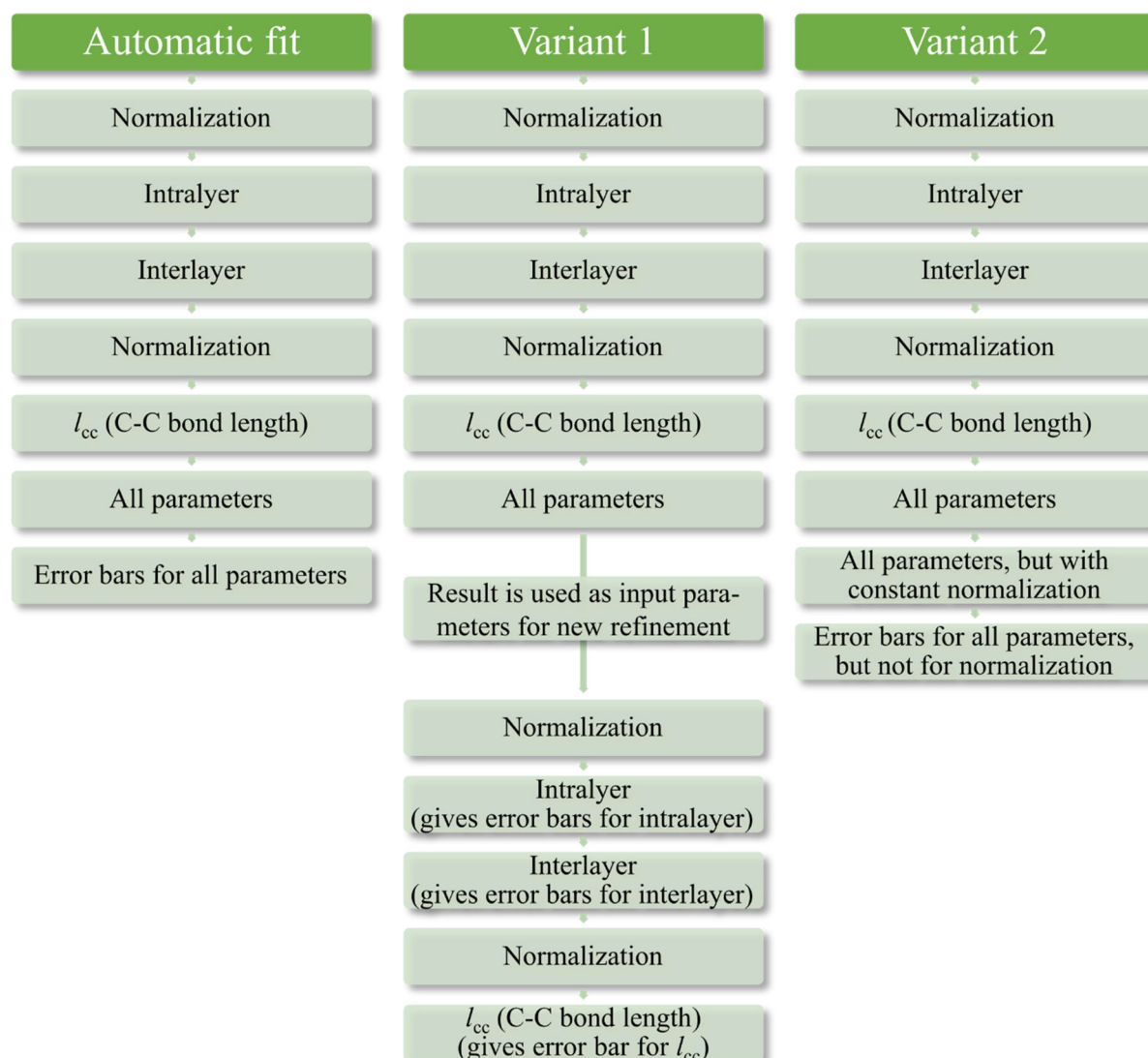
```
77  #Skip n points at start
78  nStart = 0;
79
80  #Skip n points at the end
81  nEnd = 0;
82
83  #Calculate only every n point
84  nSkip = 1;      # Step 1 # Step 4b # Step 5a
85
86  #Shifting the raw data up/down
87  nUp = 0;
88
89  #Lower and upper bounds for fitting parameters (can be changed, but does not have to be)
90  #mu, beta, a3, da3, sig3, eta
91  lb1 = [0.01; 0.01; 2.50; 0; 0; 0];
92  ub1 = [10; 20; 5.00; 1; 1; 1];
93
94  #nu, alpha, sig1
95  lb2 = [nu; 0.01; 0];
96  ub2 = [nu; 2; 2];
97
98  #q, dan, k, const1, const2, g
99  lb3 = [q; dan; 0.0001; -100000; const2; -1];
100  ub3 = [q; dan; 10000; +100000; const2; +1];
101
102  #lcc
103  lb4 = [1.2];
104  ub4 = [1.8];
105
106  #Function tolerance
107  tolFun = 1e-10;
108
109  #Maximal iterations per fit step
110  maxIter = 50;
111
112  #Weight of measurement points
113  #normal = 1 (every point has the same weight)
114  #weight = 1/y (default; lower intensity has lower weight)
115
```

```
116 weight = "weight";
117
118 useQ = false;           #Additional Debye-Waller-factor
119 b = 0.002;             #Factor
120
121 useA = true;            #Absorption correction
122 density = 2.2;          #Density of sample in g/cm^3
123 sampleThickness = 0.3;  #Sample thickness in cm
124 transmission = false;   #Transmission geometry (if false, reflection geometry is assumed)
125 absorptionCorrection = 1; #Correction factor for absorption coefficient (multiplicated to theoretical
                             coefficient)
126
127 useP = true;            #Polarization correction
128 polarizedBeam = false;  #Do you use a polarized beam?
129 polarizationDegree = 0; #Polarization direction of beam in °.
130
131 useCorrAutoColl = false; #Slit correction
132 par_r = 14;            #Radius of the diffractometer (Debye-Scherrer) in cm
133 par_delta = 4;         #Divergence angle in ° (it is as if this fixed slit were inside)
134 par_l = 5;             #Irradiated length in cm
```

**S9. Octave cannot always calculate parameter errors – what to do**

As described in chapter 5.3 in the main article, *Octave* sometimes fails to calculate the parameter errors:

“The following issue is important for determining the error range of the refinement parameters: In some cases, *Octave* is not able to calculate the correlation and covariation matrices correctly, generating “NaN” as output values for the errors. However, this does not disturb the quality of the fitting, i.e. the resulting structural parameters themselves are still reasonable. If the errors are, however, relevant, in such cases there are basically 2 ways to work around the problem, described in the SI part S9 and **Fehler! Verweisquelle konnte nicht gefunden werden.** First, the resulting parameters obtained from the fitting can be used as input parameters for another automatic refinement (variant 1 in **Fehler! Verweisquelle konnte nicht gefunden werden.**). This procedure might already allow *Octave* to calculate the matrices and therefore the errors correctly. If not, the errors from the intralayer- and interlayer-refinement (steps 1 and 2 of the refinement script) can be used as errors (see S9). Second, the “NaN” error often occurs for the normalization parameters  $k$  and  $const_l$ . Hence, the influence of the variation of these parameters on the resulting scattering curve is negligible ( $< 0.1\%$ ). Second, another possibility to avoid the NaN problem and to obtain error bars is to add an additional refinement step after all other steps, in which all microstructure parameters except for the normalization parameters are refined and therefore the errors of the parameters are calculated (variant 2 in **Fehler! Verweisquelle konnte nicht gefunden werden.**). An example for this refinement step is shown in S9.”



**Figure S14** Schematic representation for calculating the error bars for the refined parameters. If the automatic fit cannot calculate the errors, there are basically 2 options for how to proceed: Variant 1 can be used with the existing script. The result of the fit is used there as input parameters for a new fit. In this new fit, the error from the individual steps for the intralayer and interlayer parameters can be determined individually. For mathematical reasons, the error here is generally higher than if all parameters were calculated at the same time. Variant 2 describes the procedure with a modified script. In this case, after the automatic fit, a fit of intra- and interlayer parameters is performed while the now known normalization parameters are kept constant. This allows the error bars to be calculated, but it is possible that the structure parameters will change very slightly as a result.

The following script-part must be added starting at line 986 in the file “WAXS Fit-Routine-IUCr.m”.

If only these lines are added, *Octave* does not consider the results for the errors in the final results-file and the resulting plots are not saved. To do so, some other changes are necessary and already implemented in the file “WAXS Fit-Routine-IUCr with additional step all-without-normalization.m”.

```

1  #Bounds for last step (order!)
2  lb6 = [lb1; lb2(1); lb2(2); lb4; lb2(3)];
3  ub6 = [ub1; ub2(1); ub2(2); ub4; ub2(3)];
4
5  options6.AutoScaling = autoscaling;
6  options6.FunValCheck = funValCheck;
7  options6.MaxIter = maxIter;
8  options6.TolFun = tolFun;
9  options6.lbound = lb6;
10 options6.ubound = ub6;
11 options6.weights = wtWeight;
12 options6.TypicalX = typicalX;
13 options6.user_interaction = @outfun;
14
15 #All without normalization
16 fun6 = @(a, x) (fun(cno, a(1), a(2), a(3), a(4), a(5), u3, a(6), a(7), a(8), a(9), a(10), q, cH, cN, cO, cS, dan, k,
    const1, const2, useQ, b, useA, density, sampleThickness, transmission, absorptionCorrection, useP,
    polarizedBeam, polarizationDegree, useGradient, g, useCorrAutoColl, par_r, par_delta, par_l, radiation,
    wavelength, s, coh, inc));
17 \n\n\n Alles
18 settings.weights = options6.weights;
19 paramn6 = [mu; beta; a3; da3; sig3; eta; nu; alpha; lcc; sig1];
20 function result6 = result6fun(cno, mu, beta, a3, da3, sig3, u3, eta, nu, alpha, lcc, sig1, q, cH, cN, cO, cS, dan, k,
    const1, const2, useQ, b, useA, density, sampleThickness, transmission, absorptionCorrection, useP,
    polarizedBeam, polarizationDegree, useGradient, g, useCorrAutoColl, par_r, par_delta, par_l, radiation,
    wavelength, s, coh, inc, fitPath, id, x, param6, yn, settings, errorCount = 1)
21 maxerrorCount = 10;
22 result6.covp = 100*eye(length(param6));
23 try
24     result6 = curvefit_stat(@(a) (fun(cno, a(1), a(2), a(3), a(4), a(5), u3, a(6), a(7), a(8), a(9), a(10), q, cH, cN, cO,
        cS, dan, k, const1, const2, useQ, b, useA, density, sampleThickness, transmission, absorptionCorrection, useP,
        polarizedBeam, polarizationDegree, useGradient, g, useCorrAutoColl, par_r, par_delta, par_l, radiation,
        wavelength, x, coh, inc)), param6, x, yn, settings)
25 catch
26     lasterror.message
27 if errorCount < maxerrorCount

```

```

    result6 = result6fun(cno, mu, beta, a3, da3, sig3, u3, eta, nu, alpha, lcc, sig1, q, cH, cN, cO, cS, dan, k, const1,
28 const2, useQ, b, useA, density, sampleThickness, transmission, absorptionCorrection, useP, polarizedBeam,
    polarizationDegree, useGradient, g, useCorrAutoColl, par_r, par_delta, par_l, radiation, wavelength, s, coh, inc,
    x, param3, yn, settings, errorCount + 1)
29 endif
30 end_try_catch
31 endfunction
32
    function [param6, f6, cvg6, outp6, result6] = fit6(fun6, paramn6, x, yn, options6, settings, cno, mu, beta, a3, da3,
33 sig3, u3, eta, nu, alpha, lcc, sig1, q, cH, cN, cO, cS, dan, k, const1, const2, useQ, b, useA, density,
    sampleThickness, transmission, absorptionCorrection, useP, polarizedBeam, polarizationDegree, useGradient,
    g, useCorrAutoColl, par_r, par_delta, par_l, radiation, wavelength, s, coh, inc, fitPath, id, errorCount = 1)
34 maxerrorCount = 10;
35 try
36     timeStartFit = time();
37     [param6, f6, cvg6, outp6] = nonlin_curvefit(fun6, paramn6, x, yn, options6);
38     timeEndeFit = time();
39     dauerFit = timeEndeFit - timeStartFit
40 catch
41     lasterror.message
42     if errorCount < maxerrorCount
43         [param6, f6, cvg6, outp6, result6] = fit6(fun6, paramn6, x, yn, options6, settings, cno, mu, beta, a3, da3, sig3,
            u3, eta, nu, alpha, lcc, sig1, q, cH, cN, cO, cS, dan, k, const1, const2, useQ, b, useA, density, sampleThickness,
            transmission, absorptionCorrection, useP, polarizedBeam, polarizationDegree, useGradient, g,
            useCorrAutoColl, par_r, par_delta, par_l, radiation, wavelength, s, coh, inc, fitPath, id, errorCount + 1);
44     else
45         error("Too many errors when refine")
46     endif
47 end_try_catch
    result6 = result6fun(cno, mu, beta, a3, da3, sig3, u3, eta, nu, alpha, lcc, sig1, q, cH, cN, cO, cS, dan, k, const1,
48 const2, useQ, b, useA, density, sampleThickness, transmission, absorptionCorrection, useP, polarizedBeam,
    polarizationDegree, useGradient, g, useCorrAutoColl, par_r, par_delta, par_l, radiation, wavelength, s, coh, inc,
    fitPath, id, x, param6, yn, settings);
49 endfunction
50
    [param6, f6, cvg6, outp6, result6] = fit6(fun6, paramn6, x, yn, options6, settings, cno, mu, beta, a3, da3, sig3,
51 u3, eta, nu, alpha, lcc, sig1, q, cH, cN, cO, cS, dan, k, const1, const2, useQ, b, useA, density, sampleThickness,
    transmission, absorptionCorrection, useP, polarizedBeam, polarizationDegree, useGradient, g,
    useCorrAutoColl, par_r, par_delta, par_l, radiation, wavelength, s, coh, inc, fitPath, id);
52 paramn6 = param6;
53
54 mu = param6(1)
55 beta = param6(2)
56 a3 = param6(3)

```

```
57 da3 = param6(4)
58 sig3 = param6(5)
59 eta = param6(6)
60 nu = param6(7)
61 alpha = param6(8)
62 lcc = param6(9)
63 sig1 = param6(10)
64
65 convergence6 = flag(cvg6)
66 outp6 = outp6;
67 for i=1:length(param6)
68     stdabw6(i) = sqrt(result6.covp(i, i));
69     mat6(i, 1) = param6(i);
70     mat6(i, 2) = 1*stdabw6(i);
71     mat6(i, 3) = 2*stdabw6(i);
72     mat6(i, 4) = 3*stdabw6(i);
73 endfor
74 stdabw6 = stdabw6';
75 mat6 = mat6
76
77 yFit6 = fun(cno, mu, beta, a3, da3, sig3, u3, eta, nu, alpha, lcc, sig1, q, cH, cN, cO, cS, dan, k, const1, const2,
    useQ, b, useA, density, sampleThickness, transmission, absorptionCorrection, useP, polarizedBeam,
    polarizationDegree, useGradient, g, useCorrAutoColl, par_r, par_delta, par_l, radiation, wavelength, x, coh,
    inc);
78
79 if shouldPlot == true
80     plot6 = figure(6);
81     plot(x, yn, ".k;Data points;", "markersize", 10, x, yFit6, "r;Fit6;", "LineWidth", 3);
82     xlabel ("Scattering vector s / A^-1");
        ylabel ("Intensity I");
        title ("6 - All without normalization");
    endif
```



**S10. Whole refinement script**

The final refinement scripts, which can be used with *Octave* are in the SI in the directory “Example refinement”. In principle, the file “WAXS Fit-Routine-IUCr.m” should be used to perform the whole refinement. In some cases, it might happen, that the errors of the refined parameters cannot be calculated (see S9). This is often caused by the refinement of the normalization parameters (which one could not be determined so far). In this case and if the error values of the refined parameters needed to be known (and only in this case), the file “WAXS Fit-Routine-IUCr with additional step all-without-normalization.m” should be used. Hence, an additional refinement step including the intralayer and interlayer but without the normalization parameters will be performed. This will reduce a bit the quality of the whole fit, but *Octave* will calculate the parameter errors correctly.

## References

- Einert, M., Wessel, C., Badaczewski, F., Leichtweiß, T., Eufinger, C., Janek, J., Yuan, J., Antonietti, M. & Smarsly, B. M. (2015). *Macromol. Chem. Phys.* **216**, 1930–1944, doi:10.1002/macp.201500169.
- Faber, K., Badaczewski, F. M., Ruland, W. & Smarsly, B. M. (2014). *Z. anorg. allg. Chem.* **640**, 3107–3117, doi:10.1002/zaac.201400210.
- Pfaff, T., Badaczewski, F. M., Loeh, M. O., Franz, A., Hoffmann, J.-U., Reehuis, M., Zeier, W. G. & Smarsly, B. M. (2019). *J. Phys. Chem. C* **123**, 20532–20546, doi:10.1021/acs.jpcc.9b03590.
- Pfaff, T., Simmermacher, M. & Smarsly, B. M. (2018). *J Appl Crystallogr.* **51**, 219–229, doi:10.1107/S1600576718000195.
- Ruland, W. & Smarsly, B. M. (2002). *J Appl Crystallogr.* **35**, 624–633, doi:10.1107/S0021889802011007.