Octave

Note: *OctCarb*.m means *Fit-routine*.m

In this section, an instruction for instll *Octave* under Windows, macOS and Linux. In addition, a an example script for *Octave* (*OctCarb*) for the refinement of measured WAXS/WANS data using the model of Ruland and Smarsly (2002).

All content in this repository is manly based on the following publications:

- Ruland, W. and Smarsly, B. M. (2002), X-ray scattering of non-graphitic carbon: an improved method of evaluation. J. Appl. Cryst., 35, 624-633, doi:10.1107/S0021889802011007
- Faber, K., Badaczewski, F., Oschatz, M., Mondin, G., Nickel, W., Kaskel, S., Smarsly, B. M. (2014), In-Depth Investigation of the Carbon Microstructure of Silicon Carbide-Derived Carbons by Wide-Angle X-ray Scattering, J. Phys. Chem. C., 118, 29, 15705-15715, doi:10.1021/jp502832x
- Pfaff, T., Simmermacher, M. & Smarsly, B. M. (2018), *CarbX*: a program for the evaluation of wide-angle X-ray scattering data of non-graphitic carbons, J. Appl. Cryst., 51, 219-229, doi:10.1107/S1600576718000195
- Osswald, O., Smarsly, B. M. (2022), J. Appl. Cryst., in preparation

Fit-routine.m

The file *Fit-routine.m* contains an *Octave*-script (*OctCarb*) for the refinement of measured WAXS/WANS data using the model of Ruland and Smarsly (2002). The input parameters are descriped in the file itself and in https://github.com/osswaldo/ngcs/README.md.

Output of *Fit-routine.m*

During the whole refinement, *Octave* shows a plot with the actual state of refinement (if acivated). After every refinement step, *Octave* will return the refined values, the reason of the refinement stop, a matrix containing the refined values and the standard deviations. After the whole refinmenent, *Octave* will also return all refined parameters and the resulting microstructure parameters including the mathematically calculated erros. In addition, the used reinfment script, the plots (if activated), the refined parameters and an table containing all parameters and the refined iObs-values and a file containing the output-matrices and other useful values will be exported.

Console output In addition to the informative outputs such as the start and end time, the current refinement step, etc., any errors and, after each refinement, the reason for the end of the refinement, the refined values and an associated matrix are output.

Reasons for the end of the refinement

- *Canceled*: Canceled by user
- *Maximum number of iterations.*: The maximum number of iterations has been arrived. You have to increase the parameter *maxIter*.
- *Change in refinement parameters too small.*: Obvious
- *Change in calculated function too small.*: Obvious, see *tolFun*

Refined values and matrices The refined parameters and their values will be output after every refinement step by their name. The corresponding matrix also contains the values (1st column) and the error of it (2nd column, 1-sigma neighborhood of normal distribution). The 3rd and 4th columns contains the 2-sigma and 3-sigma neighborhoods of the normal distributions, respectiveley). An 1-sigma neighborhood of *10* means, that the error could not be calculated.

General in- and Output parameters for the refinement-script (*OctCarb*)

In general, the input parameters for *iObs* and the associated *.oct* files are the same. Further information about all parameters can be found in the pulblications on top.

Input parameters

iObs

Number	Parameter	Description
1	cno	Concentration of unorganized carbon
2	mu	Shape factor for gamma function to calculate N, Lc and kapc
3	beta	Shape factor for gamma function to calculate N, Lc and kapc
4	a3	Average layer distance
5	da3	Minimal layer distance
6	sig3	Standard deviation of a3
7	u3	Thermal motion
8	eta	Homogeneity of the stacks
9	nu	Shape factor for gamma function to calculate La, lm and kapa
10	alpha	Shape factor for gamma function to calculate La, lm and kapa
11	lcc	Average C-C bond length
12	sig1	Disorder of the layers (i.e. stress and strain)
13	q	Preferred orientation
14	cH	Concentration of unorganized hydrogen
15	cN	Concentration of unorganized nitrogen
16	cO	Concentration of unorganized oxygen
17	cS	Concentration of unorganized sulfur
18	dan	Anisotropy of atomic form factor of carbon
19	k	Normalization constant for log10(k * Ie.u. + const1) + const2

Number	Parameter	Description
20	const1	Constant shift for log10(k * Ie.u. + const1) + const2
21	const2	Non-constant (linear) shift for log10(k * Ie.u. + const1) + const2
22	useQ	Switch for additional parameter for incoherent background
23	b	Additional parameter for incoherent background
24	useA	Switch for additional absorption parameters
25	density	Density of the sample in g/cm^3
26	sampleThickness	Thickness of the sample in cm
27	transmission	Transmisison geometry (if false, reflection geometry is assumed)
28	absorptionCorrection	Additional correction factor for the calculation absorption coefficient
29	useP	Switch for addiational polarization correction
30	polarizedBeam	Is the beam polarized?
31	polarizationDegree	Polarization direction of beam in °
32	useGradient	Switch for additional exponential damping
33	g	Factor for exponential damping of the scattering intensity with Ie.u.' = exp
34	useCorrAutoColl	Switch for additional conversion from fixed irradiated length to fixed slit
35	par_r	Radius of the goniometer (in cm; fixed due to experiment)
36	par_delta	Divergance angle (in °; to choose by user)
37	par_l	Irradiated length (in cm; fixed during measurement)
38	radiationType	Type of radiation (0 = X-ray, 1 = neutron)
39	wavelength	Wavelength
40	S	Vector, which contains the scattering vector values (s = sin( ) /)
41	coh	Switch for calculating coherent scattering
42	inc	Switch for calculating incoherent scattering

Relationship to microstructure parameters

Parameter	Description	Refined/c
a3	Average layer distance	refined
a3min	Minimal layer distance	refined
da3	(= a3 - a3 min) Difference between average and minimal layer distance	calculated
sig3	Disorder of the stacks (standard deviation of a3)	refined
Lc	(= N * a3) Average stack height	calculated
N	(= (mu+1)/beta) Average number of graphene layers per stack	calculated
Nm	(= mu/beta)Average number of graphene layers per stack	calculated
kapc	(= 1/mu) Polydispersity of stack height	calculated
kapr	(= 3 * Pi^2 * (1/nu + 1)/32 - 1) Polydispersity of the radius of the graphene layers	calculated
eps3	(= da3/a3min) Disorder of stacks due to local strains	calculated
eta	Homogeneity of the stacks	refined
q	Preferred orientation	refined
lcc	Average C-C bond length	refined
sig1	Disorder of the layers (i.e. stress and strain)	refined
La	(=(nu+1)/alpha) Average graphene layer size	calculated
lm	(=nu/alpha) Average chord length	calculated

Parameter	Description	Refined/c
kapa	(=1/nu)Polydispersity of chord length	calculated
eps1	Disorder of graphene layers due to local strains (currently not implemented)	refined
cH	Concentration of unorganized hydrogen	known fro
cN	Concentration of unorganized nitrogen	known fro
cO	Concentration of unorganized oxygen	known fro
cS	Concentration of unorganized sulfur	known fro
dan	Anisotropy of atomic form factor of carbon	refined
k	Normalization constant for log10(k * Ie.u. + const1) + const2	refined
const1	Constant shift for log10(k * Ie.u. + const1) + const2	refined
const2	Non-constant (linear) shift for log10(k * Ie.u. + const1) + const2	refined

Output parameters

iObs return a matrix that contains the x/y values of the calculations.

iObs

- First column: Scattering vector values (s in Å^-1)
- Second column: Oberseved Intensity (*iObs*)

File output All files will be saved in the directory <*fitPath*>/<*filename*>

Plots The plots of every refinement step and the error. For last step (refinement of all parameters together) a plot of the error (relative) and a logarthmic plot will be saved.

output_*.txt This file contains all refined and constant microstructure parameters.

output_*.csv In this file, all refined and constant microparameters are saved. An 1-sigma error of *10* means, that the error could not be calculated. In addition, the measured intensity (*I*), the refined intensity (*Ifit*) and the absolute (*errorAbs*) and relative)*errorRel*) error will be saved. This file can be opened directly with a table calculation program.

data_*.mat Contains additional matrices/vectors, which can be used for manual debugging:

- tolFun
- maxIter
- paramn
- paramFinished
- param1
- convergence1

- outp1
- result1
- stdabw1
- mat1
- param2
- convergence2
- outp2
- result2
- stdabw2
- mat2
- param3
- convergence3
- outp3
- result3
- stdabw3
- mat3
- param4
- convergence4
- outp4
- result4
- stdabw4
- mat4
- param5
- convergence5
- outp5
- result5
- stdabw5
- mat5
- rQuadratFit
- chiQuadratFit

Common warnings/errors during the refinement and how to fix them

There are several errors, which can occur during the refinement process. The most common and their solutions are written down here:

warning: lower and upper bounds identical for some parameters, fixing the respective parameters

As the warning says, for one or more parameters, the upper and lower bound are identical. You can check the parameters and ignore the warning.

warning: matrix singular to machine precision, rcond = nan

Some values in the matrix are too close to 0 and your PC cannot calculate it. It is not a big problem, the PC will use 0 instead and forward the refinement. In general, this warning can be ignored.

some lower bounds larger than upper bounds

You set some lower bound to a higher value than the upper bound. You have to check and correct the parameters.

no free parameters

The lower and upper bounds of all parameters are fixed to a certain value, so there is nothing which can be refined. At least one parameter must be free.

plt2vv: vector lengths must match

This error typical occurs, if a parameter is changed, which influences the raw data, e.g., measFile, type, nStart, nEnd, nSkip. In the most of the cases, the error can resolved by restarting Octave.

svd: cannot take SVD of matrix containing Inf or NaN values

There are a lot of reasons, why this error is shown. It might be, that the starting parameters are too "bad", the amount of measurement is to low, i.e. nSkip is to high or there is to less RAM available. First, you can ignore this error,* Octave* will try to calculate the Matrix again (10 times). If the script stops, the start parameters should be refined manually.

error: Too many errors when refine

This is not a specific error, it occurs, if something goes wrong. There should be another error or warning in the output and this is the reason, why the script stops. In principle, all manually changed parameters should be checked.

The values of the error Matrix are 10, 20 and 30

These values are not the real errors, but something like a placeholder or default value. A standard deviation of 10 means, that the error cannot be calculated correctly. This is a very common mistake when trying to refine too many parameters at the same time.

Installation

In principle,* Octave* is available for Microsoft Windows, Aplle macOS, GNU/Linux ad BSD systems. While a precompiled and executable* Octave* installer is available for Windows,* Octave* must be compiled on the system

itself for other systems. Independent of the system, some additional plugins for* Octave* have to be installed manually.

Install* Octave* under Windows (tested on Windows 10 64 bit)

For Windows, an executable * .exe file can be downloaded directly: https://www.gnu.org/software/octave/download

Update* Octave* under Windows Run 'Install* Octave* under Windows' again.

Install* Octave* under macOS (tested on macOS Catalina 10.15.7)

Octave for macOS can be installed using a package manager like (Homebrew, MacPorts and Spack). In addition, a launcher app using AppleScript can also be created. The most easiest way is to use Homebrew (https://brew.sh/), so only this way will be described.

Install* Octave* using Homebrew The actual instruction is available under https://wiki.octave.org/Octave_for_macOS 0. Information: The entire process can take up to a few hours, but the actual working time in which YOU have to do something is only a few minutes.

1. Install Xcode via the App Store
2. Open a terminal and execute the following commands:

```
/bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install
brew update
brew install octave
```

Further tips In addition,* Octave* can also be compiled against gnuplot instead of qt. This result might perform better, so it is recommended to run also the following commands: 0. Information: The entire process can take up to a few hours, but the actual working time in which YOU have to do something is only a few minutes.

1. Open the file ~/.octaverc and paste the in:

```
setenv('GNUTERM','qt')
graphics_toolkit("gnuplot")
```

2. Open a terminal and execute the following commands:

```
sudo chown -R `whoami` /usr/local/share/ghostscript
brew link --overwrite ghostscript
brew install octave
# If install returns an error about not having a formular for octave, use the following com
brew tap --repair
```

Update* Octave* under macOS using Homebrew In addition,* Octave* can also be compiled against gnuplot instead of qt. This result might perform better, so it is recommended to run also the following commands: 0. Information: The entire process can take up to a few hours, but the actual working time in which YOU have to do something is only a few minutes.

1. Open a terminal and execute the following commands:

```
brew update && brew upgrade octave
```

Install* Octave* under Linux

There are many different Linux distributions, so it is impossible to write one guide for all of them. Therefore, only the most relevant and widely used Linux distributions are considered. In principle,* Octave* can be installed with a precompiled package or by compiling on the operating system itself. Since the compilation is sometimes very complicated and requires a high level of prior knowledge, this procedure is NOT described here. The corresponding instructions can be found here: https://wiki.octave.org/Category:Installation

Debian and Debian-based (such as Ubuntu) (tested on raspbian, which is based on Debian buster) Link: https://wiki.octave.org/Octave_for_Debian_systems

1. Open a terminal and execude the following commands:

```
sudo apt-get install octave
sudo apt-get install octave-io octave-statistics octave-structs octave-optim
sudo apt-get install liboctave-dev
# Optional
sudo apt-get install octave-doc octave-info octave-htmldoc
```

Red Hat Enterprise/CentOS Link: https://wiki.octave.org/Octave_for_Red_Hat_Linux_systems

1. Open a terminal and execude the following commands:

```
sudo yum install epel-release
sudo yum install octave
yum install octave-devel
```

Arch Linux (not tested) Link: https://wiki.octave.org/Octave_for_Arch_Linux

1. Open a terminal and execude the following command:

```
sudo pacman -S octave
```

Gentoo (not tested) Link: https://wiki.octave.org/Octave_for_GNU/Linux#Gentoo

1. Open a terminal and execude the following command:

```
sudo emerge --ask sci-mathematics/octave
```

Fedora (not tested) Link: https://wiki.octave.org/Octave_for_Red_Hat_Linux_systems

1. Open a terminal and execude the following command:

```
sudo dnf install octave
sudo dnf install octave-devel
```

openSUSE (not tested) Link: https://wiki.octave.org/Octave_for_openSUSE

1. Open a terminal and execude the following command:

```
sudo zypper install octave
sudo zypper install octave-devel
# The following steps are optional:
# Install OpenBLAS https://www.openblas.net/
# Afterwards, open a terminal and execude the following commands and choose OpenBLAS:
/usr/sbin/update-alternatives --config libblas.so.3
/usr/sbin/update-alternatives --config liblapack.so.3
```

Install* Octave* under BSD (not tested)

For FreeBSD and OpenBSD, precompiled packages* Octave* packages are available (https://wiki.octave.org/Octave_for_other_Unix_systems):

FreeBSD (not tested) Open a terminal and execude the following command:

```
pkg_add -r octave
```

OpenBSD (not tested) Open a terminal and execude the following command:

```
pkg_add octave
```

Important: Install additional packages/plugins (optim)

In order to use the script to refine wide-angle X-ray/neutron scattering (WAXS/WANS) data of NGCs using the Octave-script in this repository, some additional plugins must be installed. *optim* is used to perform the mathimatical refinement/minimalization operations. *optim* itself need *struct* and *statistics* as dependencies and *statistis* in turn uses *io*. If an error occurs during the installation of *optim*, it is usually because the dependencies mentioned are missing. These must then also be installed. Basically, either the automatic package installation program from Octave, which downloads and installs the package, or the manual package installation program, in which you have to download the packages manually, can be used for the installation * optim *. In principle, manual installation should only be used if the automatic installation program has failed.

Automatic installation of *optim*

Installation of *optim* The *optim* package is available under https://octave.sourceforge.io/optim/ To install *optim* either open the Octave-GUI or the Octave-CLI and execute the following command:

```
pkg install -forge optim
```

If the installation failed due to missing dependencies, you have to install *struct*, *statistics* and *io* **before** the installation of *optim*. Aftwerwards, either open the Octave-GUI or the Octave-CLI and execute the following command again:

```
pkg install -forge optim
```

Installation of *struct* To install *struct* either open the Octave-GUI or the Octave-CLI and execute the following command:

```
pkg install -struct optim
```

Installation of *statistics* To install *statistics*, you first have to install *io*. To do this, either open the Octave-GUI or the Octave-CLI and execute the following commands:

```
pkg install -forge io
pkg install -forge statistics
```

Manual installation of *optim*

If the automatic installation failed, you can also install *optim* and its dependenceis maunal.

1. Download *optim* and the needed dependecies (*structs*, *statistics* and *io*):

```
https://octave.sourceforge.io/optim/
https://octave.sourceforge.io/struct/index.html
https://octave.sourceforge.io/statistics/index.html
https://octave.sourceforge.io/io/index.html
```

2. Install first *io*, then *structs* and *statistics* and last *optim*. To do this, either open the Octave-GUI or the Octave-CLI and execute the following commands. <*path_to_file*> is the path, there the previous downloaded files are saved. The quotes (') are only necessarry, if the path or file name contains spaces. <*version*> is the version of the downloaded package, i.e. the filename.

```
pkg install '<*path_to_file*>/io-<*version*>.tar.gz'
pkg install '<*path_to_file*>/struct-<*version*>.tar.gz'
pkg install '<*path_to_file*>/statistics-<*version*>.tar.gz'
pkg install '<*path_to_file*>/optim-<*version*>.tar.gz'
```