From modelling to McStas and more

A short guide

E. Farhi, *ILL*

Data formats

McStas Data formats

McStas uses a number of input data formats, all text based.

- **Lau** (Single_crystal, PowderN, Isotropic_Sqw)
- **Laz** (PowderN, Isotropic_Sqw)
- **Sqw** (Isotropic_Sqw)
- **qSq** (Isotropic_Sqw) – mostly for liquids
- **nxs** (Sample_nxs) – this is ***not*** a NeXus/HDF
- **PDB** (SANSPDB)

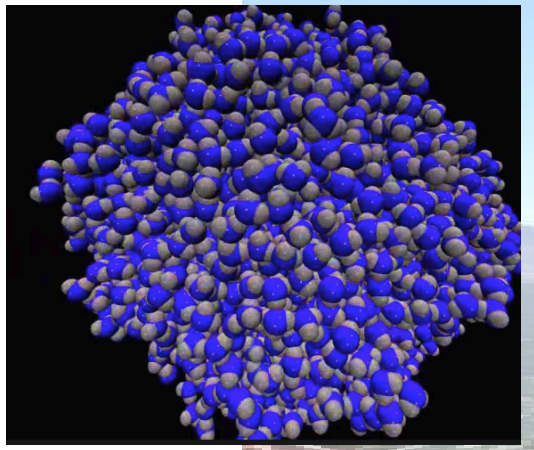
The question is how to generate these files…

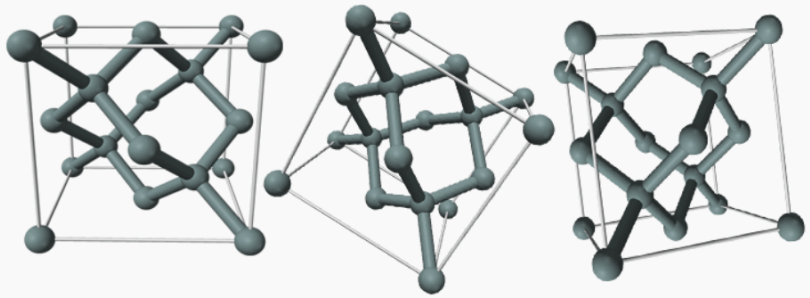
By hand ?

NEUTRONS FOR SOCIETY

Material modelling formats

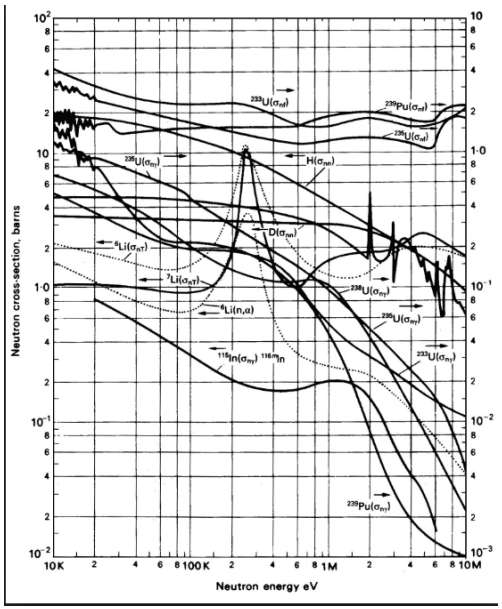
- MD (classical or DFT)
  - trajectories, many data formats
- Lattice dynamics:
  - PhonoPy FORCE_SETS
  - ...





Nuclear data base formats

- ENDF, JEFF, JENDL, …
  - all use a standard format for a set of materials. Most are monoatomic / incoherent
- ACE
  - Same as ENDF, but in binary for MCNP
- Ncrystal (2.5)
  - NCMAT,NXS,LAU



Experiments

- Intensity(d) → F(Q) → S(Q) and Laz
- Intensity(HKL) → F2(HKL) → Lau
- Intensity(theta,ToF) → S(q,w)
- Often saved as text and/or NeXus

Tools

- iFit
- Mantid
- Python/ASE (IO)
- MDANSE/nMoldyn (converters)
- Scripts of all kinds
  - OpenBABEL, ...
- cif2hkl
- bash

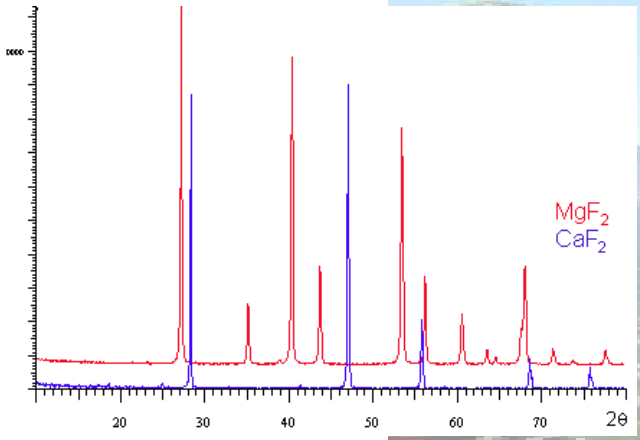
Material DataBases

- ICSD <http://icsd.ill.fr/icsd/index.html>
- COD <http://crystallography.net>
- <https://www.materialsproject.org/>
- PhononDB <http://phonondb.mtl.kyoto-u.ac.jp/>
- Nomad <https://repository.nomad-coe.eu/>

Mostly provide structures...

McStas: structure: powder

- For powders, the pure elastic diffraction is easy to prepare: LAZ
  - PowderN
  - Single_crystal(powder=1, ...)
  - Isotropic_Sqw



CIF file
FullProf/ShelX

Terminal:
```
cif2hkl --powder file.cif
```

McStas Laz file

CIF file
FullProf/ShelX

iFit:
```
cif2hkl('file.cif','','p')
```

McStas Laz file

McStas: structure: SX

- For single crystals, the pure elastic diffraction is easy to prepare: LAU
  - PowderN
  - Single_crystal
  - Isotropic_Sqw

CIF file
FullProf/ShelX

Terminal:
```
cif2hkl --xtal file.cif
```

McStas Lau file

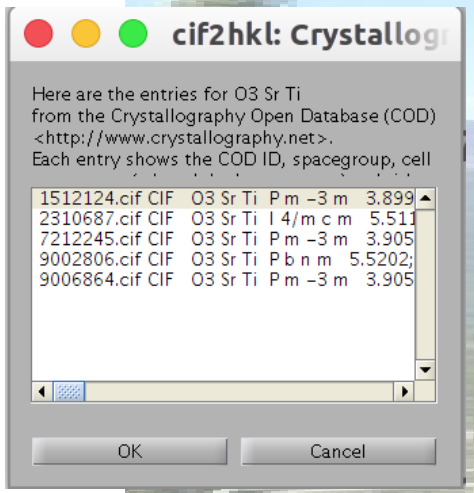
CIF file
FullProf/ShelX

iFit:
```
cif2hkl('file.cif','','x')
```

McStas Lau file

McStas: structure: COD

- cif2hkl from iFit can input chemical formula
- Requires internet connection to COD
- Hill notation: C H then alpha order



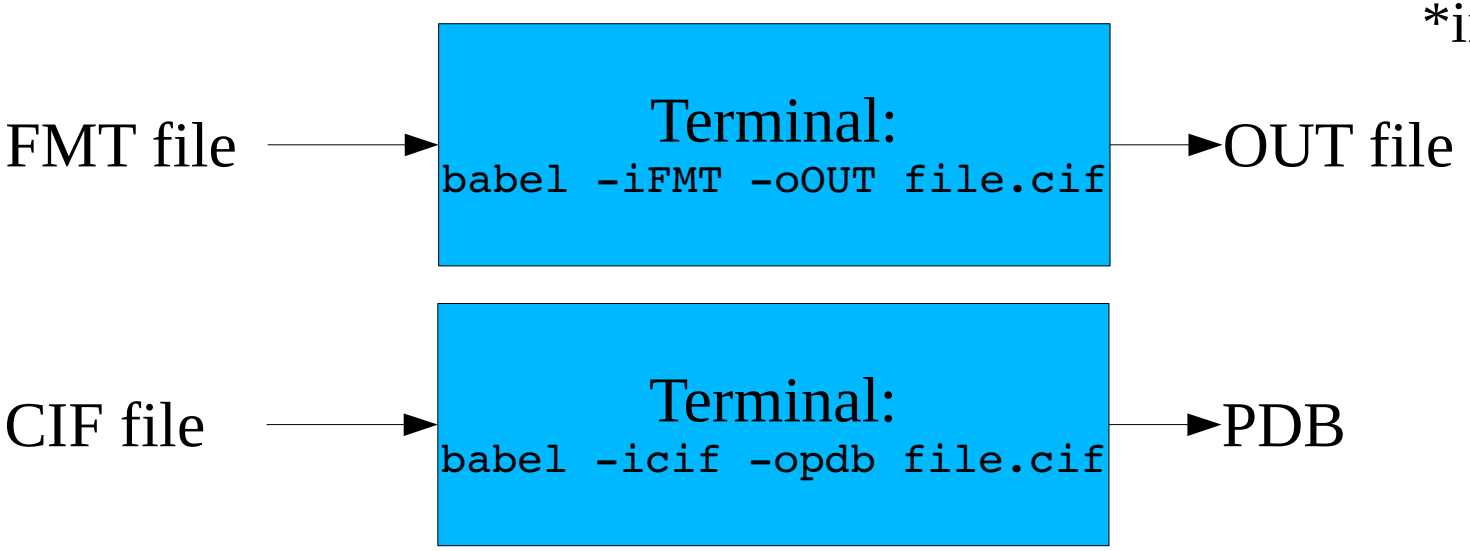
Chemical formula → iFit: `cif2hkl('cod: La Mn O3','','p')` → McStas Laz file

Chemical formula → iFit: `cif2hkl('cod: O3 Sr Ti','','x')` → McStas Lau file

McStas: structure: others

- Other McStas structure file formats
  - SANSPDB and similars: get from Data Bases
  - Sample_nxs: manual edit only (nxs, text)
  - Isotropic_Sqw: manual edit only (qSq, text)
  - NCrystal_sample*: (ncmat,nxs,lau)

*in McStas from v. 2.5



OpenBabel supports many data files

McStas: dyn from ToF spec

- Import and correct data with e.g. LAMP, Mantid
- Integrate radially (|q|) and compute $S(q,\omega)$
- Export to e.g. HDF/NeXus or other
- Import then export with iFit:
  - sqw=iData_Sqw2D('file')
  - sqw.saveas('file.sqw','mcstas')
- Can/should always edit manually (text)

```
# weight       4.002    in [g/mol]
# density      0.4784   in [g/cm^3]
# sigma_abs    0.00747  absorption scattering cross section in [barn]
# sigma_coh    1.34     coherent scattering cross section in [barn]
# sigma_inc    0        incoherent scattering cross section in [barn]
# Temperature  2        in [K]
# classical    0        experimental, contains Bose factor
```

McStas: dyn from exp. DOS

- Import and correct data with e.g. LAMP, Mantid
- Estimate g(ω) with e.g. LAMP, Mantid, MUPHOCOR
- Export to e.g. HDF/NeXus or other
- Import then export with iFit:
  - g=iData_vDOS('file')
  - inc=g.incoherent;
  - inc.saveas('inc.sqw','mcstas')
- Can/should always edit manually (text)

McStas: dyn from nuclear DB

- Import and export with iFit:
  - sqw=iData_Sqw2D('tsl.endf')
  - saveas(sqw**(1)**, 'inc.sqw','mcstas')
- Ncrystal (2.5) will provide an alternate solution
- Can/should always edit manually (text)

```
# weight       4.002    in [g/mol]
# density      0.4784   in [g/cm^3]
# sigma_abs    0.00747 absorption scattering cross section in [barn]
# sigma_coh    1.34     coherent scattering cross section in [barn]
# sigma_inc    0        incoherent scattering cross section in [barn]
# Temperature 2        in [K]
# classical    0        experimental, contains Bose factor
```

McStas: dyn from MD

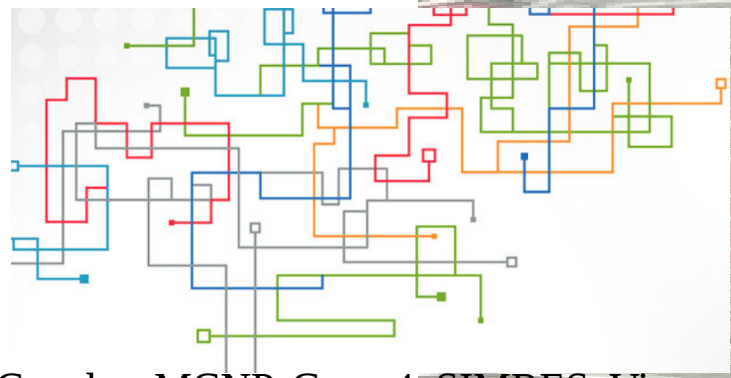
- Import trajectory into MDANSE (converters)
- Double click the trajectory to allow analysis
- Use Plugins:
  - Analysis/Scattering/DCSF and DISF
  - *Analysis/Scattering/Structure Factor*
  - *Analysis/Dynamics/Density of States*
- Import then export with iFit:
  - sqw=iData_Sqw2D('DCSF.nc')
  - sqw.saveas('coh.sqw','mcstas')
- Can/should always edit manually (text)

McStas: dyn from Lattice Dynamics

- Using PhonoPy:
  - iFit: s4d=sqw_phonons('phonopy_directory')
- Computing on the fly:
  - iFit: s4d=sqw_phonons('cod: La Mn O3')
  - p=powder(s4d)
  - d=iData_Sqw2D(iData(p, [], 0:10, 0:100))
  - d.saveas('coh.sqw','mcstas')
- Other:
  - Import in iFit, convert to Sqw2D, export.

Neutron events

- McStas handle ‘rays’ as packets of neutron events.
- Summing-up builds intensity and distributions.
- McStas can read/write ‘events’ from other software:
  - MCNP TRAC and SSW
  - TRIPOLI
  - MCPL*



*Allows interchange with many other MC codes: MCNP, Geant4, SIMRES, Vitess, ...

- McStas can generate events as well for Mantid via NeXus/HDF files.

McStas automation: bash

- Can use cif2hkl and mcrun commands:
  - **cif2hkl** ‘file’ --out ‘file.laz’ **--**powder
  - **mcrun** –dir=test **instr** Powder=file.laz
  - **mcplot**.pl -psc test

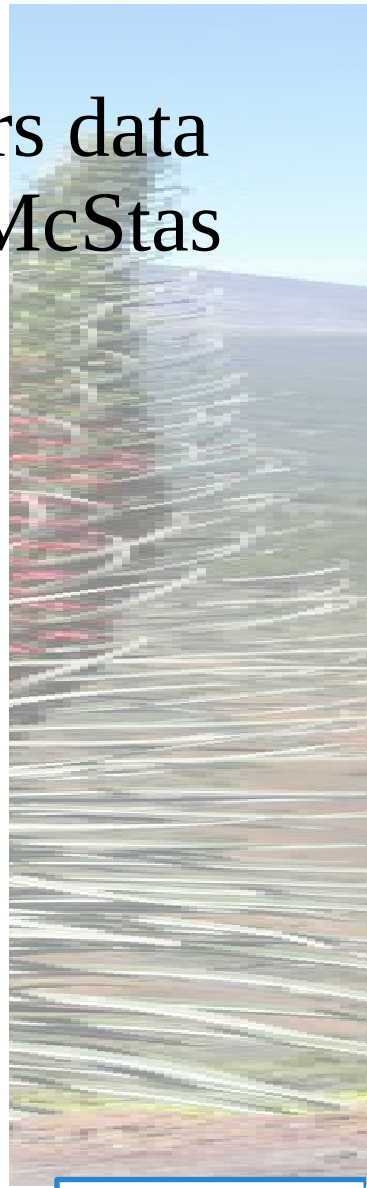
```
[~]$ bash
```

NEUTRONS FOR SOCIETY

McStas automation: iFit

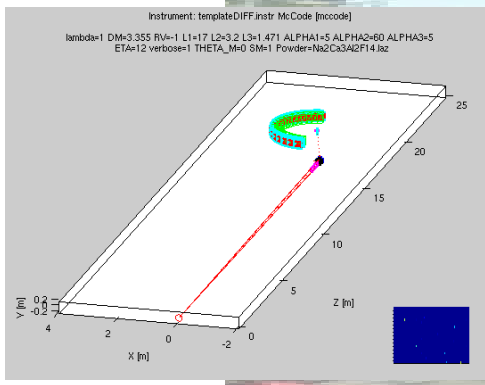
- iFit is a generic infrastructure which gathers data sets handling, fitting, S(q,w) models, and McStas hooks.

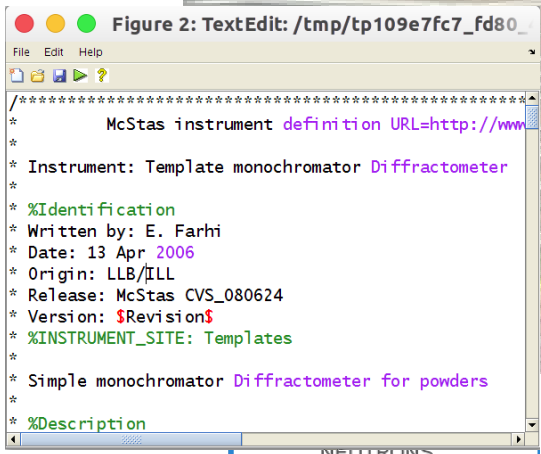
iFit



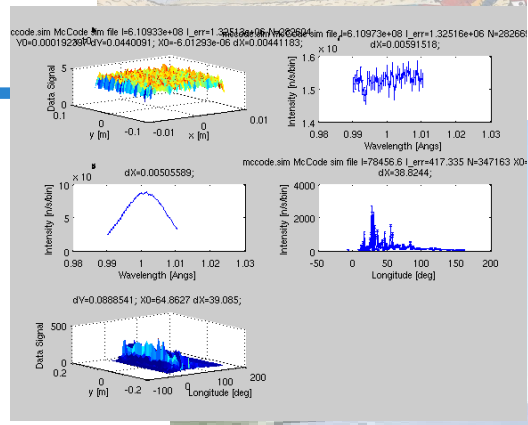
McStas iFit: build

- McStas can be controlled from within iFit.
- Open Matlab/iFit
- Create the McStas model with:
  - model = mccode('instr')
- Plot the geometry with:
  - plot(model) *% has contextual menus*
- Edit the instrument and re-compile
  - edit(model)





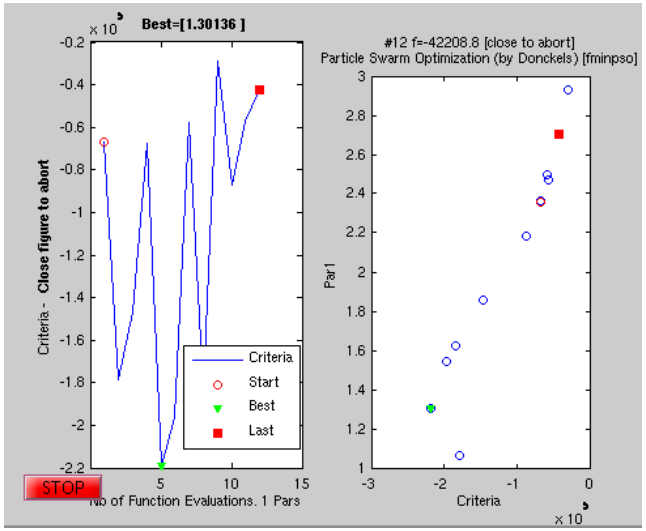
McStas iFit: eval



- Run with (default 1e6 event and pars)
  - data = iData(model, [], nan);
  - subplot(model) *% plot monitors, contextual menus*
- Specify parameters
  - data = iData(model, ‘lambda=2.36; coh=Cu.laz’)
- Do a scan:
  - data = iData(model, ‘lambda=[1.2 2.4 3.6]’)
- Change neutron events #
  - model.UserData.options.ncount = 1e7;

McStas iFit: optim

- Fix all parameters but *lambda*, Maximize model value:
  - mlock(model, ‘all’); munlock(model, ‘lambda’)
  - xlim(model, ‘lambda’, [1 3]); % bounds
  - fmax(model, ‘lambda=2.36’, ‘’, nan)
  - fmax(model, ‘lambda=2.36’, ‘OutputFcn=fminplot’, nan)



McStas iFit: advanced

- You can add McStas models:
  - model=mccode('instr1')+mccode('instr2') + ...
- For instance:
  - Instr1: structure
  - Instr2: spin-wave
  - Instr3: phonons
  - Instr4: incoherent
- The different models can also be assembled as a set of samples chosen at execution in a single McStas instrument.

Links to software

- McStas http://www.mcstas.org (includes cif2hkl)
- iFit http://ifit.mccode.org (includes cif2hkl)
- Mantid http://www.mantidproject.org
- MDANSE http://www.mdanse.org
- ASE https://wiki.fysik.dtu.dk/ase/
- Ncrystal https://github.com/mctools/ncrystal