Visualization and processing of computed solid-state NMR parameters: MagresView and MagresPython

Simone Sturniolo, Timothy F.G. Green, Robert M. Hanson, Miri Zilka, Keith Refson, Paul Hodgkinson, Steven P. Brown, Jonathan R. Yates

1. Introduction

First-principles calculations of NMR parameters in the solid state are a vital component of the NMR crystallography toolbox, providing a connection between the underlying atomistic structure and the observed NMR spectrum [1,2]. Such calculations can now provide all of the key NMR parameters relevant to diamagnetic solids: the magnetic shielding, electric field gradient (EFG), direct dipolar coupling and indirect spin–spin (J) coupling. Significantly, calculations provide the full tensorial information for each NMR parameter, giving the orientation with respect to the crystal lattice. There is a clear need to visualise the computed results – to assign experimental spectra, and to gain understanding about the relation between NMR parameters and atomic level structure. There are a number of programs to view crystal structures and tools have been developed to visualise spin systems e.g. SIMMOL [3] and Spinach GUI [4] or to display and manipulate NMR spectra [5–8]. However, there is still a need for a visualisation tool that has a complete understanding both of NMR parameters and crystal symmetry.

A related issue occurs when needing to process large numbers of calculated NMR parameters. For example for a system undergoing rapid dynamics it may be necessary to perform a molecular dynamics simulation and to calculate the NMR parameters as an ensemble average over the trajectory. Similarly, when studying materials with positional or compositional disorder it may be necessary to simulate multiple copies of the unit cell each containing a different configuration of atoms. An illustration of the latter case is a study of the Al/Si ordering in the Calcium Aluminosilicate Gehlenite by Florian et al.[9]. They considered 50 unit cells with differing Al/Si ordering each containing 256 atoms. From these calculations the correlation of the $^{27}$Al EFG with the number of silicon next nearest neighbours could be obtained. Extracting this information by hand from the calculation output files would be extremely tedious and prone to error. In contrast a scripted approach should be both efficient and reliable.

In this article we introduce two open source tools to address these issues. The Magres file format is a simple format for storing computed NMR parameters. MagresView is built upon the widely used Jmol crystal viewer, and provides an intuitive environment to display computed NMR parameters. It can provide simple pictorial representation of one- and two-dimensional NMR spectra as well as output a selected spin-system for exact simulations with dedicated spin-dynamics software. MagresPython provides a simple scripting environment to manipulate large numbers of computed NMR parameters to search for structural correlations.

$$\frac{\text{Keywords: Solid-state NMR}}{\text{GIPAW}}$$

$$\frac{\text{DFT}}{\text{Simulation}}$$

$$\frac{\text{Visualization}}{\text{A R T I C L E ~ I N F O}}$$

Article history:
Received 4 October 2015
Received in revised form
24 March 2016
Accepted 19 May 2016

Keywords:
Solid-state NMR
GIPAW
DFT
Simulation
Visualization

$$\frac{\text{ABSTRACT}}{\text{We introduce two open source tools to aid the processing and visualisation of ab-initio computed solid-state NMR parameters. The Magres file format for computed NMR parameters (as implemented in CA-STEP v8.0 and QuantumEspresso v5.0.0) is implemented. MagresView is built upon the widely used Jmol crystal viewer, and provides an intuitive environment to display computed NMR parameters. It can provide simple pictorial representation of one- and two-dimensional NMR spectra as well as output a selected spin-system for exact simulations with dedicated spin-dynamics software. MagresPython provides a simple scripting environment to manipulate large numbers of computed NMR parameters to search for structural correlations.}}$$

© 2016 The Authors. Published by Elsevier Inc. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).
2. Magres file format

As more computational codes incorporate the prediction of NMR parameters, a unified output format is necessary in order to create a common platform for the further development of tools. The format described here can be used as an input to tools for diverse tasks such as visualisation, interfacing to spin simulation software, and for performing the sort of flexible analysis required in the course of research. It can also be used as an archival format for preserving research products. This format is written by default by CASTEP v8.0 [10] and later and QuantumEspresso v5.0.0 [11] and later.

This unified format includes all the information produced by a first-principles calculation of NMR parameters, without redundancy, and all quantities are precisely defined theoretically and with units. The full specification can be found in the SI but here we will give a brief summary.

The format is designed with minimal complexity to be easy to understand and parse by someone of modest programming skill in most computer languages. By deliberately designing the format to be easily parsed, without compromising the information contained, it will be significantly more accessible to the average researcher. The file itself is an ASCII text file and is broken up into a number of blocks, each containing a sequence of records, one per line. Records are identified by the contents of their first column, unrecognised blocks or records can be ignored, single-line comments can be added and units are always specified – in a sense making the data self-documenting.

The first line of the file identifies it as a magres file. Additional lines starting with # are optional comments. After this the file can contain a block for meta-data describing the computational details of the calculation, e.g. the code used and its version, the methodology used, the choice of density functional, and the details of the basis set. The exact fields are not specified, but it should be sufficient to reproduce the calculation in conjunction with the atomic data. Next, the file contains a block with the unit cell lattice vectors, atomic coordinates and any symmetry operations present. These are all given in a Cartesian basis. Finally, there is a block containing the magnetic resonance parameters, usually rank-2 tensors. The tensors are also stored in a Cartesian basis. Magnetic shielding tensors are given in units of ppm, EFG tensors are given in atomic units and the J-coupling tensors are given in reduced SI units. This choice of units has been made so quantities given are independent of the isotope of the nucleus. The MagresView and MagresPython software allow a user to select a particular isotope and display parameters in commonly used systems of units. Files written in this format are given the file extension `.magres`. A sample file for quartz is shown in Fig. 1.

---

Fig. 1. Sample Magres file for Quartz generated by CASTEP. Note that for readability in this figure all values have been truncated to 4 decimal places.

Please cite this article as: S. Sturniolo, et al., Solid State Nucl. Magn. Reson. (2016), http://dx.doi.org/10.1016/j.ssnmr.2016.05.004
3. MagresView

MagresView is a user-friendly environment for the visualization and analysis of the Magres file format. The primary author is Simone Sturniolo. It is an application customized for the NMR crystallography community which operates in a web browser; it can be used over the Internet but a downloadable version for offline usage is also available. We developed MagresView to allow a smooth and intuitive interaction with calculated NMR parameters: it provides a graphical representation of the simulated data, generates human readable records (tables and graphs) from the computer readable format, and provide a pipeline for spin dynamics simulations.

MagresView is built on top of Jmol; a free, open-source molecular visualization software available in both Java and JavaScript-only versions [12]. Jmol has full 3D-crystal symmetry and periodic crystal visualization capabilities. Jmol includes basic NMR-specific functionalities as well as the ability to load and display magres files. It can be used directly in its standalone Java version, as a standalone web-based Java applet, or as a JavaScript-only web-based page component [13]. MagresView provides more advanced NMR-specific functionality and an easy to use graphical interface. Table 1 contains a rundown of the functionalities provided by each program for various NMR specific tasks.

Table 1
Summary of MagresView and Jmol NMR functionality.

| Visualization | Magnetic shielding | Electric field gradient | Indirect spin-spin (J) coupling (ISC) | Dipolar coupling | Euler angles |
|---------------|--------------------|-------------------------|--------------------------------------|-----------------|-------------|
| MagresView    | Represent tensors as ellipsoids. Calculate tensor parameters; isotropy, anisotropy and asymmetry (or span and skew). | Calculate isotropic component of ISC and convert to frequency for the given isotopes. Visualize ISC as link network. | Calculate isotropic component of ISC and convert to frequency for the given isotopes. Visualize ISC as link network. | Calculate the magnetic shielding tensor. Compute tensor parameters and quadrupole coupling given the isotope for the element. | Calculate Euler angle differences between principal frames of two picked MS or EFG tensors. Compute a table of all Euler angle differences between MS and EFG tensors on the same or different atom. Choice of Euler angle convention. |
| Jmol          | Visualize isotropy, anisotropy and asymmetry (or span and skew) with labels and/or color codes. Convert from magnetic shielding to chemical shift. | Visualize isotropy, anisotropy and asymmetry (or span and skew) with labels and/or color codes. | Represent tensors as ellipsoids. Calculate tensor parameters and quadrupole coupling given the isotope for the element. | Visualize dipole coupling constant (DCC) for arbitrary atom pairs and convert to frequency for the given isotopes. Visualize DCC as link network. | Visualize DCC with atoms from neighboring unit cells ("ghosts"). |
| MagresView    | Visualize Vzz, anisotropy and asymmetry (or span and skew) and quadrupole coupling with labels and/or color codes. | Visualize isotropy, anisotropy and asymmetry (or span and skew) and quadrupole coupling with labels and/or color codes. | Visualize isotropy, anisotropy and asymmetry (or span and skew) with labels and/or color codes. | Visualize DCC for all selected atoms in a spherical neighborhood around one. Visualize DCC with atoms from neighboring unit cells (will appear as "ghost" atoms). Compute isonuclear and total root square sum coupling. | Visualize dipole coupling constant (DCC) for arbitrary atom pairs and convert to frequency for the given isotopes. Visualize DCC as link network. |
| Jmol          | Represent tensors as ellipsoids. Calculate tensor parameters and quadrupole coupling given the isotope for the element. | Represent tensors as ellipsoids. Calculate tensor parameters and quadrupole coupling given the isotope for the element. | Represent tensors as ellipsoids. Calculate tensor parameters and quadrupole coupling given the isotope for the element. | Represent tensors as ellipsoids. Calculate tensor parameters and quadrupole coupling given the isotope for the element. | Represent tensors as ellipsoids. Calculate tensor parameters and quadrupole coupling given the isotope for the element. |
| MagresView    | Represent tensors as ellipsoids. Calculate tensor parameters and quadrupole coupling given the isotope for the element. | Represent tensors as ellipsoids. Calculate tensor parameters and quadrupole coupling given the isotope for the element. | Represent tensors as ellipsoids. Calculate tensor parameters and quadrupole coupling given the isotope for the element. | Represent tensors as ellipsoids. Calculate tensor parameters and quadrupole coupling given the isotope for the element. | Represent tensors as ellipsoids. Calculate tensor parameters and quadrupole coupling given the isotope for the element. |
| Jmol          | Represent tensors as ellipsoids. Calculate tensor parameters and quadrupole coupling given the isotope for the element. | Represent tensors as ellipsoids. Calculate tensor parameters and quadrupole coupling given the isotope for the element. | Represent tensors as ellipsoids. Calculate tensor parameters and quadrupole coupling given the isotope for the element. | Represent tensors as ellipsoids. Calculate tensor parameters and quadrupole coupling given the isotope for the element. | Represent tensors as ellipsoids. Calculate tensor parameters and quadrupole coupling given the isotope for the element. |
| MagresView    | Represent tensors as ellipsoids. Calculate tensor parameters and quadrupole coupling given the isotope for the element. | Represent tensors as ellipsoids. Calculate tensor parameters and quadrupole coupling given the isotope for the element. | Represent tensors as ellipsoids. Calculate tensor parameters and quadrupole coupling given the isotope for the element. | Represent tensors as ellipsoids. Calculate tensor parameters and quadrupole coupling given the isotope for the element. | Represent tensors as ellipsoids. Calculate tensor parameters and quadrupole coupling given the isotope for the element. |
| Jmol          | Represent tensors as ellipsoids. Calculate tensor parameters and quadrupole coupling given the isotope for the element. | Represent tensors as ellipsoids. Calculate tensor parameters and quadrupole coupling given the isotope for the element. | Represent tensors as ellipsoids. Calculate tensor parameters and quadrupole coupling given the isotope for the element. | Represent tensors as ellipsoids. Calculate tensor parameters and quadrupole coupling given the isotope for the element. | Represent tensors as ellipsoids. Calculate tensor parameters and quadrupole coupling given the isotope for the element. |

MagresView provides versatile options for the visualization of NMR ab-initio results (Fig. 2). Magnetic shielding and electric field gradient tensors can be represented graphically as ellipsoids, surrounding each atom. This represents the principal directions and values of the tensor by the principal axes and lengths of the ellipsoids (Fig. 3). In this way the relation between the principal frame of the tensor and that of the crystal structure can be clearly seen. Many calculated parameters of interest can be revealed by text labels or by color scaling on top of the 3D crystal structure (Figs. 4 and 6). By providing a suitable reference shielding, users can convert between the calculated magnetic shielding and the chemical shift. Users can also choose between various conventions for parameterising the anisotropy of NMR tensors, e.g. Haeberlen [14] (anisotropy, asymmetry) or Herzfeld–Berger [15] (span, skew). Direct dipolar couplings and indirect spin–spin couplings (ISC, J) are represented as links connecting pairs of atoms; including periodic images of atoms in an adjacent unit cell (Fig. 5). Calculation and plotting of dipolar couplings are available even without the ab-initio calculated results; dipolar couplings are determined directly from the atomic coordinates, allowing MagresView to compute them from data in the form of standard crystallographic file formats such as PDB, CIF, and SHELX, all of which Jmol can read. All the properties detailed above can be presented for all or only some of the atoms within the system.

Please cite this article as: S. Sturniolo, et al., Solid State Nucl. Magn. Reson. (2016), http://dx.doi.org/10.1016/j.ssnmr.2016.05.004
Supplementary material related to this article can be found online at http://dx.doi.org/10.1016/j.ssnmr.2016.05.004.

MagresView can visualize a predicted NMR spectrum from the calculated data: 1D spectra can be plotted as simple spectral lines with Lorentzian broadening; the 2D NMR plotting tool can generate a plot of a correlation spectrum by positioning peaks at the coordinates corresponding to the known isotropic chemical shifts (referenced and taking into account the second order quadrupolar shift if required), as well as scaling their intensity by the strength of the dipolar or J coupling for the corresponding spin pair, see Fig. 6. The predicted spectra can then be overlaid on top of an experimental spectrum, allowing easy comparison and assignment of the resonances.

For cases where a more exact match with experimental spectra is required, MagresView supports the production of spinsys files, the input format required by both SIMPSON [19] and pNMRsim [20] to define a spin system. An alternative is to write out the data in JSON format [21] which can then be post-processed by a Python script, json2sim, distributed together with MagresView. This script produces a batch of input files configured to run a full simulation with either SIMPSON [19] or SPINEVOLUTION [22].

MagresView can also be used to generate several useful outputs. Human-readable and tabulated ASCII files summarizing only selected parameters of interest can be imported into external software. The active view in MagresView can be saved to picture format (.png and .jpg). Predicted spectral plots, both 1D and 2D,
can be saved to vector graphics format (SVG), that are easily
convertible to PDF or PostScript for use in presentations and
papers.

A full description of MagresView’s functionality is found in
Table 1. The MagresView webpage [23] provides easy-to-follow
tutorials to help new users get started.

4. MagresPython

Processing NMR parameters with a GUI is not appropriate for
all systems, or problems. A programmatic approach is necessary
for efficiently implementing high-throughput approaches, e.g.
structure searching or when considering an ensemble of structures
- e.g. snapshots of a molecular dynamics simulation, or sites
within a disordered material. Implementing analysis program-
metrically also ensures reproducibility, aids rapid prototyping, and
allows publication of verifiable methodology. Hence, as a
complement to MagresView, we have developed the MagresPy-
thon library [24]. Tim Green is the primary author. This library
enables natural manipulation of NMR parameters using Python,
which is a popular programming language in the scientific
community.

The MagresPython library implements a robust parser of the
Magres file format, reading the information into native data
structures. It then provides classes to represent concepts in the
standard, such as atoms, which can have magnetic shielding ten-
sors, electric field gradients and indirect spin couplings. These
classes implement the various well-known conventions for tensor
components and can implicitly handle converting between re-
duced J-couplings and coupling frequencies, and between electric
field gradients and quadrupole couplings, using a built in library of
the gyromagnetic ratios and quadrupole moments for all known
isotopes.

The library also provides containers to represent lists of atoms,
with behaviour similar to native Python containers. A simple

![Fig. 5. $^1$H–$^1$H dipolar couplings in β-aspartyl-alanine within 3 Å of the selected atom. Couplings to atoms in neighbouring molecules are shown. (embedded video).](image)
querying system can be used to select subsets of atoms, for example by species or spatial proximity to other atoms, using periodic boundary conditions, and concisely extract properties. By providing an additional set of well-tested tools that have proved useful in real research, these features add value to the library beyond just providing a parser for the Magres format. The library has a number of unit tests and integration tests, along with tutorial notebooks for the IPython system. As a GitHub project it is open for contributions and, with time, should represent best practices in manipulation of Magres files.

Fig. 7 shows an example script which processes the Magres files from calculations on 50 disordered structures on Gahlenite [9]. The script classifies each Al site as either T1 (4 Al/Si second nearest neighbours) or T2 (3 Al/Si second nearest neighbours) and for each type of site plots a histogram of $^{27}$Al quadrupolar coupling constants, distinguishing between sites on the basis of how many Al second nearest neighbours are present. The structure is shown in MagresView in Fig. 8. The script takes a short time to execute on a standard desktop computer producing the average values and images shown in Fig. 7.

5. Conclusions

First-principles calculations of NMR parameters in the solid state can in many cases provide results with accuracy capable of assigning experimental spectra. This has enabled the community to devise increasingly sophisticated computational simulations in order to address complex materials problems. We have developed a number of freely available tools to aid the visualisation and processing of computed NMR parameters. We hope these provide a platform for future developments, including closer integration with spin-simulation codes and databases of computed and experimental solid-state NMR parameters.

---

**Fig. 6.** Experimental 2D $^{1}$H–$^{1}$H (500 MHz) double quantum MAS (35 kHz) spectrum of L-alanine [17,18]. The position of peaks from first-principles calculated data has been superimposed as orange crosses using MagresView. The $^{1}$H chemical shifts for the mobile CH$_3$ and NH$_3$ groups have been averaged before plotting using a MagresPython script (The corresponding magres file and script are given in the SI).

**Fig. 7.** A selection of an IPython notebook output, showing the MagresPython library being used to concisely extract and plot with the matplotlib library the $^{27}$Al CQ’s of an ensemble of structures from a disordered system as a histogram.
Fig. 8. (left) Structure of Gehlenite: Ca (green), Al (grey), Si (brown), O (red). (right) Al atoms colored according to magnitude of $C_Q$. The largest values of $C_Q (~12.9$ MHz) are colored red, and correspond to the T1 sites Al-Al$_2$Si, while the smallest values ($~2.6$ MHz) are colored blue and correspond to the T2 site Al-Al$_4$. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Acknowledgements

The development of MagresView is supported by the Collaborative Computational Project for NMR Crystallography funded by the EPSRC (UK) Grants EP/J010510/1 and EP/M022501/1. MZ acknowledges funding from the European Union under a Marie Curie Initial Training Network FP7-PEOPLE-2012-ITN Grant Agreement number 316630 CAS-IDP. JRY was supported by a Royal Society Research Fellowship.

Appendix A. Supplementary material

Supplementary data associated with this article can be found in the online version at http://dx.doi.org/10.1016/j.ssnmr.2016.05.004.

References

[1] C. Bonhomme, C. Gervais, F. Babonneau, C. Coelho, F. Pourpoint, T. Azaïs, et al., First-principles calculation of NMR parameters using the gauge including projector augmented wave method: a chemist's point of view, Chem. Rev. 112 (2012) 5753–5779, http://dx.doi.org/10.1021/cr300108z.
[2] T. Charpentier, The PAW/GIPAW approach for computing NMR parameters: a new dimension added to NMR study of solids, Solid State Nucl. Magn. Reson. 40 (2011) 1–20, http://dx.doi.org/10.1016/j.ssnmr.2011.04.006.
[3] M. Bak, R. Schultz, T. Vosegaard, N.C. Nielsen, Specification and visualization of anisotropic interaction tensors in polypeptides and numerical simulations in biological solid-state NMR, J. Magn. Reson. 154 (2002) 28–45, http://dx.doi.org/10.1006/jmre.2001.2454.
[4] A.G. Biternas, G.T. Charnock, I. Kuprov, A standard format and a graphical user interface for spin system specification, J. Magn. Reson. 240 (2014) 124–131, http://dx.doi.org/10.1016/j.jmr.2014.01.004.
[5] R.J. Lancashire, The SpecView Project: an Open Source Java viewer and converter for JCAMP-DX and XML spectral data files, Chem. Cent. J. 1 (2007) 31, http://dx.doi.org/10.1186/1752-133X-1-31.
[6] G.J. Sharman, Java applets for viewing one- and two-dimensional NMR spectra, Magn. Reson. Chem. 44 (2006) 1008–1012, http://dx.doi.org/10.1002/mrc.1884.
[7] N. Pellet, jsGraph and jsNMR—advanced scientific charting, Challenges 5 (2014) 294–295, http://dx.doi.org/10.3390/challe5020294.
[8] T. Vosegaard, jsNMR: an embedded platform-independent NMR spectrum viewer, Magn. Reson. Chem. 53 (2015) 285–290, http://dx.doi.org/10.1002/mrc.4195.
[9] P. Florian, E. Veron, T.E.G. Green, J.R. Yates, D. Massiot, Elucidation of the Al/Si ordering in Gehlenite Ca$_2$Al$_2$SiO$_7$ by combined 29 Si and 27 Al NMR spectroscopy/quantum chemical calculations, Chem. Mater. 24 (2012) 4068–4079, http://dx.doi.org/10.1021/cm3016955.
[10] S.J. Clark, M.D. Segall, C.J. Pickard, P.J. Hasnip, M.I.J. Probert, K. Refson, et al., First principles methods using CASTEP, Z. Krist. 220 (2005) 567–570, http://dx.doi.org/10.1524/zkri.220.5.567.65075.
[11] P. Giannozzi, S. Baroni, N. Bonini, M. Calandra, R. Car, C. Cavazzoni, et al., Quantum Espresso: a modular and open-source software project for quantum simulations of materials, J. Phys. Condens. Matter. 21 (2009) 395502, http://dx.doi.org/10.1088/0953-8984/21/39/395502.
[12] Jmol: an open-source Java viewer for chemical structures in 3D, (n.d.). www.jmol.org.
[13] (http://sourceforge.net/projects/jmol/files), (n.d.).
[14] U. Haeberlen, High Resolution NMR in Solids Selective Averaging: Supplement 1 Advances in Magnetic Resonance, Elsevier Science, 2012 (accessed 31.05.15) https://books.google.com/books?hl=en&lr=&id=–kV-5uCpByAC&pgis=–.
[15] J. Herzfeld, A.E. Berger, Sideband intensities in NMR spectra of samples spinning at the magic angle, J. Chem. Phys. 73 (1980) 6021, http://dx.doi.org/10.1063/1.440136.
[16] C.H. Görbitz, Crystal and molecular structures of the isomeric dipeptides alpha-L-aspartyl-L-alanine and beta-L-aspartyl-L-alanine, Acta Chem. Scand. B 41 (1987) 679–685, http://dx.doi.org/10.1063/1.440136 (accessed 16.09.15) http://www.ncbi.nlm.nih.gov/pubmed/3442178.
[17] S.P. Brown, Probing proton–proton proximities in the solid state, Prog. Nucl. Magn. Reson. Spectros. 50 (2007) 199–251, http://dx.doi.org/10.1016/j.pnms.2006.10.002.
[18] I. Schnell, H.W. Spiess, High-resolution 1H NMR spectroscopy in the solid state: very fast sample rotation and multiple-quantum coherences, J. Magn. Reson. 178 (2006) 153–227, http://dx.doi.org/10.1016/j.jmr.2006.04.039.
[19] M. Bak, J.T. Rasmussen, N.C. Nielsen, SIMPSON: a general simulation program for solid-state NMR spectroscopy, J. Magn. Reson. 147 (2000) 296–330, http://dx.doi.org/10.1006/jmre.2000.2179.
[20] P. Hodgkinson, pNMRsim: a general simulation program for large problems in solid-state NMR, (https://www.dur.ac.uk/paul.hodgkinson/pnMRSim).
[21] (http://json.org).
[22] M. Veshtort, R.G. Griffin, SPINEVOLUTION: a powerful tool for the simulation of solid and liquid state NMR experiments, J. Magn. Reson. 178 (2006) 248–282, http://dx.doi.org/10.1006/jmre.2005.07.018.
[23] (http://www.ccppc.ac.uk/magresview).
[24] Tim Green, (https://github.com/tfg/magres-format).