

New Tools for the Analysis of EXAFS: The DL_EXCURV Package

S. Tomić^{a,*}, B.G. Searle^a, A. Wander^a, N.M. Harrison^{a,b},
A.J. Dent^c, J.F.W. Mosselmans^d, and J.E. Inglesfield^e

^a*Computational Science and Engineering Department, CCLRC Daresbury Laboratory,
Warrington, Cheshire WA4 4AD, UK*

^b*Department of Chemistry, Imperial College of Science, Technology and Medicine,
London SW7 2AZ, UK*

^c*Diamond Light Source Ltd., Rutherford Appleton Laboratory, Chilton, Didcot, Oxon
OX11 0QX, UK*

^d*Synchrotron Radiation Department, CCLRC Daresbury Laboratory, Warrington,
Cheshire WA4 4AD, UK*

^e*Department of Physics and Astronomy, University of Wales Cardiff, Cardiff CF24 3YB,
UK*

Abstract

This paper reports on a modular package for the analysis of extended X-ray absorption fine structure (EXAFS) data. The package, DL_EXCURV, has been developed at The CCLRC Daresbury Laboratory and is based on the EXCURV98 code. The major modifications to that package includes the removal of all system dependant directives providing complete portability across a wide range of platforms. A new graphical user interface (GUI) has also been developed providing an interface between DL_EXCURV and the DLV package. This provides interoperability between DL_EXCURV and other CCP3 codes. The GUI also greatly simplifies the set up procedures for the analysis and also provides powerful graphical tools for the visualization and analysis of the results.

Key words: EXAFS, EXCURV

PACS: 41.60.Ap, 87.64.Fb

* Corresponding author.

Email addresses: s.tomic@dl.ac.uk (S. Tomić), b.searle@dl.ac.uk (B.G. Searle).

1 Introduction

Since the origin of the extended X-ray absorption fine structure (EXAFS) oscillations was correctly explained by Sayers et al. (1970) [1], EXAFS has become the technique of choice for local structural investigations in a diverse range of material systems. This technique has also proved very useful when used *in situ* to study chemical reactions of matter under extreme conditions of temperature and pressure. The EXAFS oscillation in the absorption spectrum is caused by interference in the final state from backscattering of the outgoing photoelectron wave by neighbours of the excited atom. Hence it is extremely sensitive to nearest neighbour bond lengths.

The first program to use curved wave theory (Lee and Pendry, Ref. [2]) to analyse EXAFS, was the EXCURV code. This was originally written by Gurman et al. [3]. The code has undergone a number of developments over the last 20 years, including the incorporation of a proper treatment of multiple scattering, restrained refinement, and whole spectrum analysis. The most recent general release, EXCURV98 [5,6], introduced multiple edge refinement and a full cluster treatment of multiple scattering. Up to the end of the year 2000 over 900 scientific papers have been published that have used a version of the EXCURV code to obtain structural information from X-ray absorption spectra.

In 2002 the experimental community approached Collaborative Computational Project 3 (CCP3)[7] to discuss the possibility of establishing direct support for the EXCURV program. Previous freely available versions of EXCURV have generally only been accessible as an executable installed on a central server at The CCLRC Daresbury Laboratory. This policy was maintained as the code contained many system dependant, and hence non-portable, sections of code. However, with the rapid growth in speed and memory of PCs users desire a portable version which can be installed at their home institutions.

The DL_EXCURV package described here can be thought of as two separate parts: DL Visualize (DLV)[8] and DL_EXCURV. Together these sections provide a powerful tool designed to fit a model structure to experimental EXAFS oscillation in the X-ray absorption spectrum. However, DL_EXCURV can also be run as a command line driven package independent of DLV in a manner analogous to the EXCURV98 package.

To fulfill those requirements the old EXCURV98 package has been re-written in modular FORTRAN 90. System dependencies have been removed and the new code, DL_EXCURV v1.0, will run on platforms ranging from Windows PC systems through to UNIX workstations. In addition, the Daresbury Laboratory Visualize (DLV) package, [8] which provides a powerful environment for the visualization and manipulation of scientific data, has been interfaced to the DL_EXCURV. The DLV package also provides tools allowing the construction of graphical user

interfaces (GUI) and consequently a GUI to the DL_EXCURV package has been designed and implemented. The new code, DL_EXCURV v1.0, was placed on beta release in late 2003 and test versions, together with the DLV package were installed on end stations at the Synchrotron Radiation Source (SRS) in Daresbury in early 2004 for user testing and evaluation. The current version of the GUI (DLV v2.4) provides an interface to both single scattering analysis and full multiple scattering analysis of EXAFS. Within the single scattering mode the package reports the radius of shells of atoms surrounding the emitter, the Debye-Waller factors, and occupation numbers of the shells. In multiple scattering mode a full three dimensional structural determination is performed.

2 The DL_EXCURV Code

The EXCURV98 package has been completely rewritten using standard FORTRAN 90 and renamed DL_EXCURV v1.0. FORTRAN 90 features have been exploited to produce a new modular program design. This new structure has resulted in a package which is easy to maintain and easy to expand. Data is stored and passed between program units in self containing data modules which now replace all `Common` and `Data` blocks used in previous versions of the code. Full use of data hiding via `Public/Private` statement is implemented and all modules are written using the `Implicit None` directive. This ensures maximum flexibility and clarity in the construction of the main code, maximizes reusability of modules, and facilitates the trapping and fixing of programmer errors.

One of the major obstacles to EXCURV98 being a portable code was extensive use of system (UNIX) calls. This was the principle reason why this code has never been released to the user community and has only been accessible via central CCLRC Daresbury Laboratory servers. In the current version we have removed all system dependent routines from the code. In some cases routines have been removed from the code although the majority have been rewritten using FORTRAN 90 standard commands e.g. `date_and_time`. The system calls to the non-standard error function `erf(z)` have been replaced with a FORTRAN implementation of the complementary error function modelled by the Chebyshev approximation [9]. The non standard trigonometric functions with argument in degrees (for example `sind(a)`) have also been rewritten in the form of standard FORTRAN radian argument functions.

Another source of non-portability was the extensive use of binary formatted external data files. These have been replaced by a new I/O system based on standardized formats in a portable ASCII-based representation.

Removal of some system dependent calls results in minor changes to file handling and file name manipulation in DL_EXCURV compared to EXCURV98. All commu-

nications with the operating system in terms of file input/output must now be accomplished using explicitly the full path name of the file. The ability to use UNIX features such as (. .) and (~) has therefore been lost.

One of the future goals in the development of DL_EXCURV is to provide the ability to perform self consistent phase shifts calculations. Fixing previously existing bugs, DL_EXCURV is now able to read in potentials from external files. This new functionality of the code directly opens up the possibility of using potential files produced by a number of *ab initio* codes such as self consistent layer Korringa-Kohn-Rostoker (KKR) codes and density functional theory (DFT) codes. Currently potentials generated by both KKR and CRYSTAL [10] are being tested and evaluated.

Future extensions of software will include full dynamic memory allocation for all storage. Although not critical yet, this will remove built in pre-sets limits on any values within the calculation, providing the way towards analysis of large physical and biological systems. This will also remove the need inherent in the earlier FORTRAN 77 code of recompilation as the system size is varied.

3 The DLV Interface to DL_EXCURV

DLV has been designed to provide an integrated environment for data visualization and analysis for CCP3 codes [11]. Here we describe its extension to provide a GUI and visualization environment to the DL_EXCURV, Fig. 1. The code is fully integrated to DLV allowing users to utilize DLV's powerful tools for the creation of full 3D models of material structures which can then be used as input to DL_EXCURV, Fig. 1. Structural models can be created either via the user interface or loaded from existing files. DLV also provides a mechanism for loading files from databases such as the chemical database service (CDS) [12].

As DLV is interfaced to other codes, it provides mechanism for data passing, and hence interoperability with other CCP3 codes.

3.1 The Main Panel

The principle interface to DL_EXCURV is via the main panel, Fig 1, which provides a full range of options allowing the user to load experimental input file, set up the initial structure model, or to preview their experimental results.

Two basic mechanism are provided to define a geometry. Firstly this can be achieved by inputting an existing model into DLV [8], Fig 1. In this way sequences of EXCURV98 commands which describe the material structure, like constituents atom

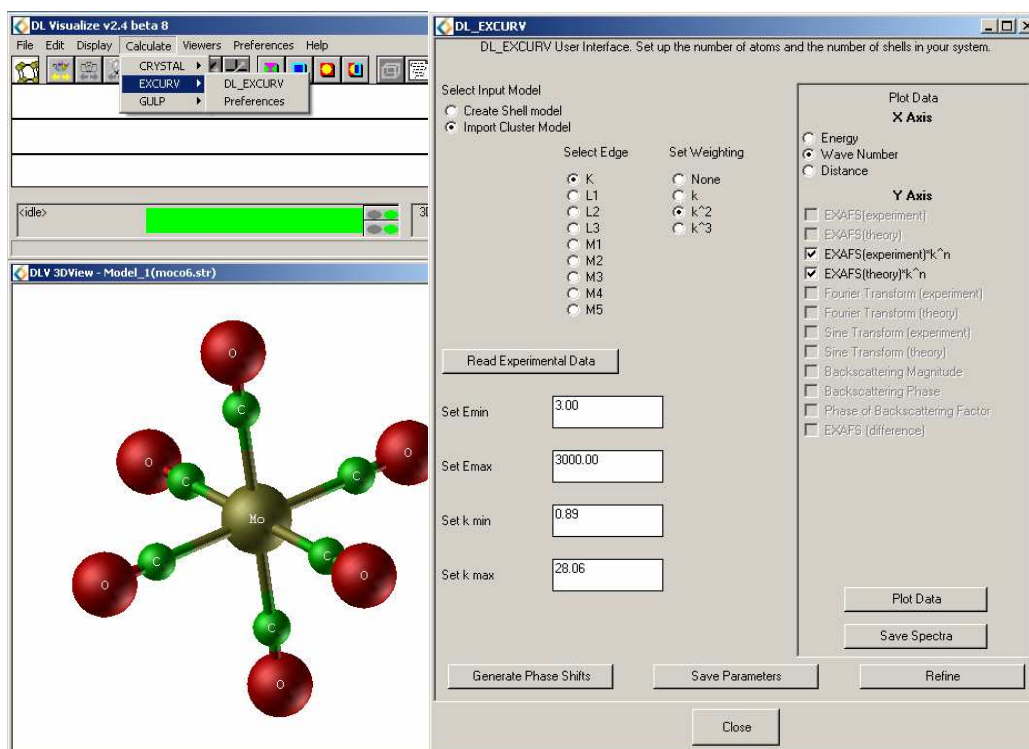


Fig. 1. DLV panel with implemented link to DL_EXCURV software in Calculate menu (upper left). Cluster model of Mo(CO)₆ displayed in DLV 3DView window (down left). The main panel of the DL_EXCURV software with the input cluster model from DLV option set (right).

types (c atom1 Mo, . . .), number of shells (c ns 2), radius of the shells (c r1 2.1, . . .) and space group symmetry (for example: sym Oh) are automatically passed from DLV to DL_EXCURV. Alternatively a shell model can be created via the main panel. The material structure is then set up by choosing a central (excited) atom type, followed by the choice of other atom types, and the number of shells in the structure at which point the potentials and phase shifts can be calculated. At this stage information on the experimental X-ray absorption edge under consideration should also be set-up.

The main panel also provides, through DLV, a full range of plotting options for displaying the level of agreement between theoretical and experimental EXAFS spectra. All graphics can be output in a variety of formats and can also be saved to disk. Results also can be saved on disk in an ascii file, for analysis in other graphical packages.

3.2 The Potentials and Phase Shifts

To proceed with the fitting procedure of the EXAFS spectra, the potentials and corresponding phase shifts must be calculated for each constituent atom of the ex-

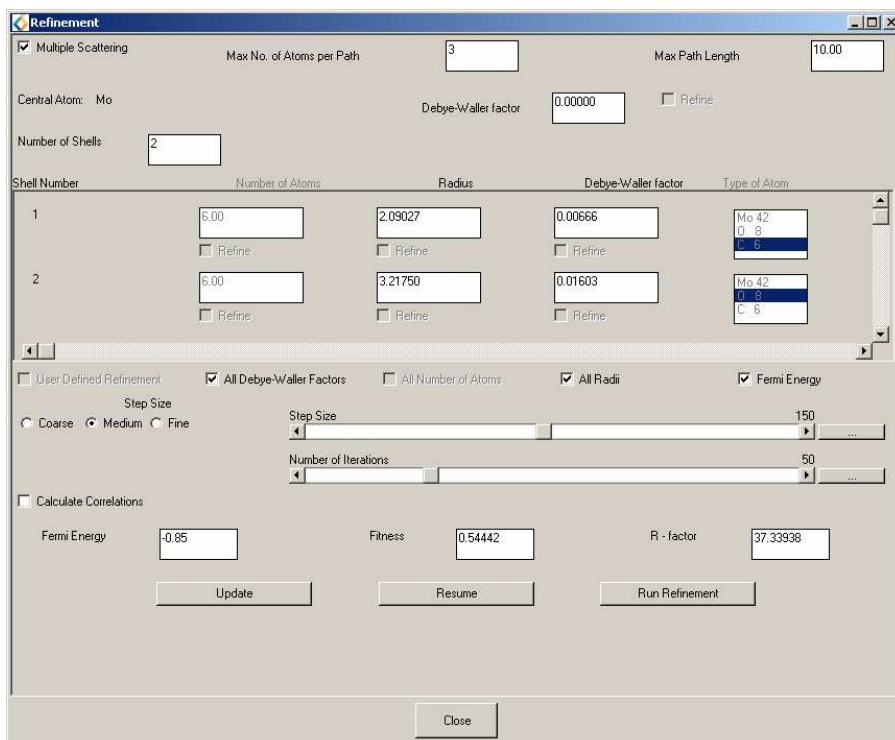


Fig. 2. The refinement panel, set for the multiple scattering of the photoelectron in $\text{Mo}(\text{CO})_6$ material loaded as an input cluster model from DLV. Refinement of the Debye-Waller factors, all radii and Fermi energy presented corresponds with command `ref` are 150 followed by `s 50` in the command line driven mode of EXCURV98.

aminated material sample. The code allows at most two atom types per cluster during potential and phase shifts calculations. If the system contains several different neighbors it is often best to choose the lightest neighbour. This restriction has been inherited from EXCURV98 but will be removed in the next version of DL_EXCURV. Independently of the input model, atomic potential calculations can be done in the presence of a core-hole (relaxed approximation) or by the $Z = Z + 1$ approximation of von Barth (sudden approximation) [13].

3.3 Refinement

The most important part of the program package, from the user perspective, is the ability to iteratively refine the theoretical EXAFS spectra in order to obtain the best fit with the experimental data and thus to solve the structure. The refinement procedure is based on least squares minimization in k space under constrain or restrain condition when well defined elements of the structure are known. The refinement panel is presented in Figure 2.

Both structural optimization via single and multiple scattering of the outgoing photo-electron is implemented and can be control through the refinement panel.

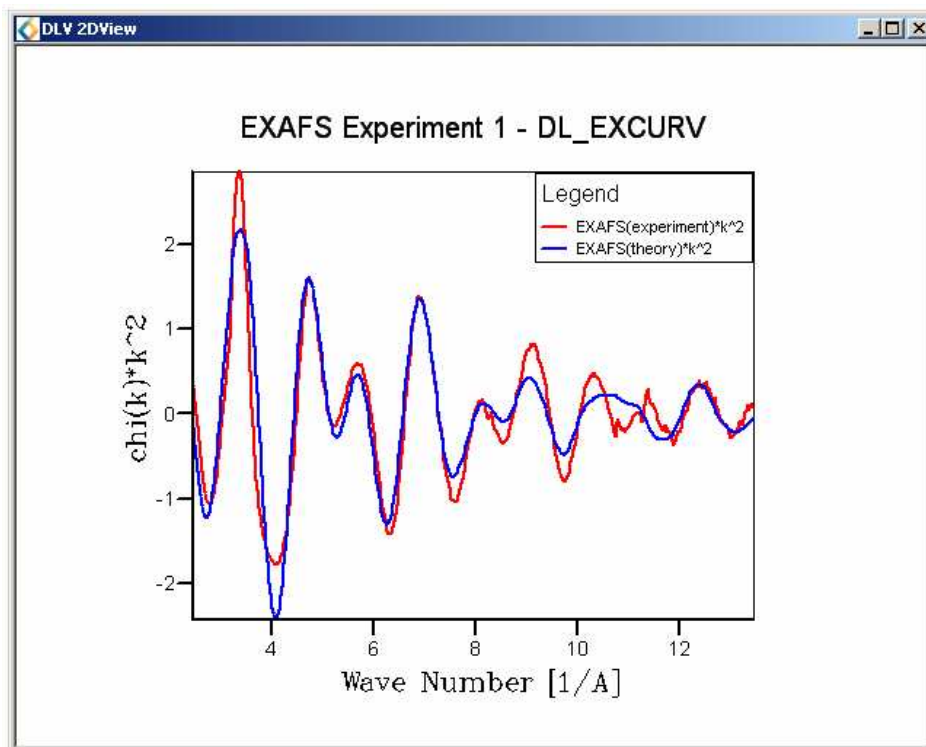


Fig. 3. Theoretical (blue line) and experimental (red line) EXAFS spectra of $\text{Mo}(\text{CO})_6$, weighted with k^2 , vs. wave number.

The parameters which can be refined are: the Debye-Waller factors, the Fermi energy, the number of atoms in each shell together with the associated shell radius (in single scattering mode), and the distance of a particular atom from the central atom. Each of those parameters can be refined either individually or by applying constrains on related physical quantities (for example: All Radii) which are refined together. The iterative procedure is controlled by two sliders which provide fine tuning of the number of iterations and the step size.

The general strategy for the fitting procedure in this mode is to minimize the resultant **R - factor** and Fit Index in each subsequent iteration cycle. Once satisfactory agreement between theory and experiment has been achieved the EXAFS spectra can be previewed from the main panel, Fig. 3

3.4 Interprocess communications

Until now, all job submission in DLV has used a batch type mechanism, where an input file with all the commands is created, the program is run, and the output files are recovered for display [8]. Interfacing to DL_EXCURV requires a more interactive approach where DLV submits commands to the program, waits for the results, and DL_EXCURV continues to run.

This method of job submission has required a new implementation of the abstract job class to support interprocess communications. On UNIX systems this uses socket pairs and on Windows it uses anonymous pipes. The communication channels are created during the creation of the DL_EXCURV process and attached to its standard input and output. To simplify the interpretation of the output by DLV, modified versions of some of the EXCURV98 commands have been implemented in DL_EXCURV e.g. PLOT.

4 Testing and Validation

The current package has been written in modular FORTRAN 90. It is closely based on the original EXCURV98 code. The program is standard conforming and has been compiled and executed on a wide variety of platforms and computer architectures. The code has been validated on PC-Linux (SuSE 8.1, 9.0, 9.1), Sun Solaris, SGI Irix, as well as PC-Microsoft Windows systems. For the Linux/UNIX systems the code has been compiled using the Portland Group Fortran compiler `pgf90`, the Sun Solaris Fortran compiler `f90`, the SGI Fortran compiler, and the Intel Fortran compiler `ifort`. For PC-Windows systems the code has been compiled with Compaq Visual Fortran 6.5.

DL_EXCURV can also be run in command mode independently of the DLV interface. In this mode DL_EXCURV retains the full set of output graphical options (both on screen and in an output PostScript file) as EXCURV98 on Linux/UNIX based systems. However, due to non-portability of the graphics libraries used within the original EXCURV98 package on MS Windows system all graphical outputs must be redirected to PostScript output files (using the command: `gset device plotter` followed by `gset plotter ps_file`).

5 Code Availability

The current release of DL_EXCURV is version 1.0 and DLV 2.4. It is available for Linux, Sun Solaris, SGI Irix, and PC-Microsoft Windows platforms. More information can be found on <http://www.cse.clrc.ac.uk/cmgi/EXCURV>. The currently release of DLV that supports DL_EXCURV can be accessed via that site or directly from <http://www.cse.clrc.ac.uk/cmgi/DLV>.

6 Conclusion

We have developed a new modular DL_EXCURV code which is interfaced with an extended version of the DLV program [8]. A graphical user interface significantly simplifies manipulation of input and output data and eases control of the refinement process of the R-factor and Fit Index between theoretical and experimental EXAFS spectra. A new development of a modular FORTRAN 90 code ensures platform for further improvement of the code functionality. The current code provides an easy way to input model parameters, to set up the fitting procedure, to control and fine tune the process of iteration and to visualize experimental and theoretical results. Currently work is in progress to develop improved methods of generating scattering potentials. Traditionally phase shifts have been generated from a superposition of neutral atomic solutions. However, for systems involving substantial charge transfer between the constituent atoms full self-consistent potentials should be utilized. Such potentials account for the effects of both charge transfer and the Ewald potential that will be significantly different at the various ionic sites within the crystal. The use of such potentials has already been shown to be of critical importance for the interpretation of low energy electron diffraction (LEED) experimental data for studies of ionic systems where strong charge transfer between the constituent atoms takes place, such as in transition metal oxides [14]. It is highly likely that similar effects will be observed in the analysis of EXAFS data from such systems. It is anticipated that this new feature will be implemented and released at the beginning of 2005.

Acknowledgements

The development of the DL_EXCURV package has been provided as part of the CCP3 flagship grant activities [7]. We are very grateful for funding by the EPSRC (UK) via the CCP3 project (GR/R68542/01). We thank John Charnock for useful discussion.

References

- [1] D.E. Sayers, E. A. Stern, and F. W. Lytle, Phys. Rev. Lett. **27**, 1204 (1971).
- [2] P.A. Lee and J.B. Pendry, Phys. Rev. B **8**, 2795 (1975).
- [3] S.J. Gurman, N. Binsted and I. Ross, J. Phys. C. **17**,143 (1984)
- [4] J.J. Rehr and R.C. Albers, Rev. Mod. Phys. **72**, 621 (2000).
- [5] N. Binsted, EXCURV98: CCLRC Daresbury Laboratory computer program.
- [6] <http://srs.dl.ac.uk/xrs/>
- [7] <http://www.ccp3.ac.uk>
- [8] B.G. Searle, Comp. Phys. Commun. **137**, 25 (2001).
- [9] W.H. Press, S.A. Teukolsky, W.T. Vetterling, and B.P. Flannery, Numerical Recipes (Fortran version), (Cambridge University Press, 1989)
- [10] V.R. Saunders, R. Dovesi, C. Roetti, R. Orlando, C. M. Zicovich-Wilson, N.M. Harrison, K. Doll, B. Civalleri, I.J. Bush, Ph. D'Arco, M. Llunell, CRYSTAL2003 User's Manual, University of Torino, Torino, 2003.
- [11] For a review see thematic issue of Computer Physics Communication **137**, No. 1, (1 June 2001).
- [12] <http://cds.dl.ac.uk/>
- [13] K. Zhang, E.A. Stern, and J.J Rehr, "Adiabatic versus sudden turn-on of multielectron effects in X-ray absorption," Springer Proc. Phys. **2**,18 (1984).
- [14] R. Lindsay, A. Wander, A. Ernst, G. Thornton, B. Montanari and N. Harrison (private communication)