

J. Synchrotron Rad. (1999). **6**, 274–275

APEX: cross-platform analysis program for EXAFS

Nicholas Dimakis^{a/} and Grant Bunker^a

^a*Department of Biological Physical and Chemical Sciences, Illinois Institute of Technology, Chicago, IL 60616 USA.
E-mail: dimakis@biosgo2.bio.aps.anl.gov*

We have developed version 1.0 of a freely available (including source code) suite of basic X-Ray Absorption Fine Structure (XAFS) data analysis programs for data reduction and single scattering analysis. This package is based on the University of Washington (UW/NRL) Fortran 77 programs that are available on the International XAFS Society (IXS) database, complemented by a graphical TCL/TK scripting language based user interface which runs virtually unchanged between platforms, using the native look and feel of the corresponding platform. The package has been tested on MacOS 8.1, Linux, IRIX, Windows95 and NT. Particular emphasis is placed on simplicity, reliability, and (sup)portability. APEX 1.0 in its current form is suitable for routine data analysis and training, and systematic improvements and extensions to the underlying codes are planned.

Keywords: EXAFS; Data Analysis.

1. Introduction

There has long been a need for a basic set of X-Ray Absorption fine Structure (XAFS) analysis programs that are freely available, including source code; that runs on all major operating systems; is readily extensible; is easy to use by beginners; and is a useful tool for experts. The user community of the Biophysics Collaborative Access Team (BioCAT) which supports this effort has very heterogeneous computing environments, and we wish to minimize the effort required to develop and support data analysis codes on multiple platforms. In addition, we wish to use a set of programs for training purposes which could be made available at no cost to the user.

The original UW/NRL programs (available on the IXS database) constitute a good, basic, robust, well-tested set of programs for single scattering data analysis upon which to build additional features. The MacXAFS package of Furenliid, Bouldin, *et al.* (1995) fills this need admirably under MacOS by providing an intelligent Hypercard stack (program and data) that gathers input from the user and coordinates execution of these Fortran programs, and other supplementary ones. Unfortunately, the Hypercard front-end is not machine-portable, its use being limited to MacOS at this time (incorporation of HyperScript into QuickTime Media Layer may allow it to be run under Windows, but not Linux). We are aware of no other package running on MacOS, Windows 95/98/NT, and Linux/Unix that is available at present.

Using our experience with MacXAFS as a guide, we have written an executive program in the TCL/TK (Tool Command Language/Tool Kit) (Ousterhout, J., 1994, Welch B., 1997) script language which has the advantages that it is free, relatively easy to understand, and runs under all major operating systems – MacOS, Linux, various Unix implementations, and Windows95/NT (VMS

can be added if there is sufficient demand) – using the native graphical interface. This approach, in contrast to the alternative of rewriting everything in Java (an approach which also has merit), allows us to reuse standard, well tested code that has already been ported to popular platforms, and to leverage the large code base of Fortran and C routines, and the efforts spent by many others in implementing TCL/TK on various platforms. This approach allows us to systematically and incrementally improve the underlying Fortran programs (or C or Java), and to keep the interface layer code base nearly identical between platforms. It also will allow us (as time permits) to restructure the Fortran programs (wrapped in C or converted using F2C) into dynamically loaded libraries, offering functions that are more tightly coupled to the APEX environment. Various light weight utility programs can be recoded directly into TCL as desired. The structure of APEX permits anyone with a modest understanding of TCL/TK to add or remove a program card or modify it to meet one's needs. This architecture offers a practical way to incrementally and systematically improve and extend the package. Once released, official versions will be made available at the BioCAT home page ¹ and mirror sites.

2. Discussion

To date, the package has been tested on a Silicon Graphics O2 under IRIX Version 4.0, Apple Macintosh PPC operating system 8.0/8.1 and PC compatibles under Windows 95 and NT. APEX will automatically identify the machine environment and will make the necessary settings without any user intervention. When APEX is called for the first time it will only ask the user to locate the executables and/or applescript directory. This information is saved in a preference file and does not need to be changed again, unless the user changes the executables and/or applescript directory location. If the machine is a Macintosh then any external program that is called from APEX is executed via an applescript rather than an EXEC command. It should be mentioned here that APEX will not run in any earlier MacOS than 8.0, not because Tcl/Tk code is not portable, but because Tcl requires a package that includes commands to execute an applescript which are not available into previous versions of MacOS. We also observe that although external program invocation is faster on UNIX and Wintel environments, because of one fewer layers of system software, it is still reasonably fast on recent model Macs, and it is expected to be very fast under MacOS 8.5 and later because of native code Applescript.

APEX has both local and on-line help using the user's choice of web browser (e.g. Netscape Navigator, MS IE, Opera). The local help is derived from similar help provided in MacXAFS for each of the programs, but online help will connect the user to an APEX web page so the user can be informed by any bug fixes and enhancements, and to download an updated version.

In case of multiple-user environments (e.g. Unix/Linux), the main program together with the executables and local help directory can be installed in shared directory. APEX will automatically set up all necessary links to a user-local working directory, thus permitting multiple users to run APEX at the same time.

APEX has been written with emphasis on simplicity and ease of use. Superfluous modes and options are minimized, and much attention is paid to error trapping at the interface layer, thus minimizing fruitless program invocations. Options that are rarely used and require more advanced XAFS knowledge are available as parameters, so as not to confuse novices. APEX also informs the user about invalid entries and guides the user to a more reasonable entry. The user has always the option to modify background colors and set them to his or her own preference. Although the standard

foreground color is black, limited foreground color modification is also possible.

At present the package handles basic operations from raw data conversion through ratio method and nonlinear least squares fitting using either empirical standards or FEFF (Rehr *et al.* 1991) standards. The core codes are based on the codes underlying the MacXAFS package, which derived from programs originally written at the University of Washington in Edward Stern's laboratory (B.A. Bunker, W.T. Elam, C.E. Bouldin, G. Bunker, E. Keller, K. Kim), and subsequently simplified and restructured at the Naval Research Laboratories by W.T. Elam and K. Kim. It does not include, but is compatible with, more recent codes (such as FEFF7, AUTOBAK and FEFFIT) which are separately licensed and therefore cannot be included in this package. However, within APEX we have included an interface to FEFF.

Future plans include systematic improvements to the underlying codes (particularly graphics, which is rather primitive at present), consolidation to a uniform application code-base, conformation to the most recent IXS Standards and Criteria guidelines, and extension to multiple scattering fitting and regularization.

3. Conclusion

APEX, version 1.0, is to our knowledge the only EXAFS data analysis package that is freely available and portable among the major operating systems, and which offers a user-friendly interface for beginners. Forthcoming versions will offer improved interactive graphics programs, native TCL code for most of the utility routines, and advanced features not presently available in any analysis programs.

This project is supported by NIH-RR08630.

References

- Biophysics Collaborative Access Team, <http://biocat1.iit.edu>
Bouldin, C.E, Elam W.T. and Furenlid L., (1995). *Physica B* **208&209** 190-192.
Ousterhout, J., (1994) *TCL and TK*, Addison-Wesley.
Rehr, J.J., De Leon, J., Zabinsky, and Albers, R.C., (1991). *it J. Amer. Chem. Soc.* **113**, 5135-5140.
Welch B., (1997). *Practical Programming in TCL and TK*, Prentice-Hall.

(Received 10 August 1998; accepted 21 December 1998)