PySpline: A Modern, Cross-Platform Program for the Processing of Raw Averaged XAS Edge and EXAFS Data

Adam Tenderholt[*], Britt Hedman[†] and Keith O. Hodgson[*,†]

[*]*Department of Chemistry, Stanford University, Stanford CA 94305, USA*
[†]*Stanford Synchrotron Radiation Laboratory, SLAC, Stanford University, Menlo Park CA 94025, USA*

Abstract. PySpline is a modern computer program for processing raw averaged XAS and EXAFS data using an intuitive approach which allows the user to see the immediate effect of various processing parameters on the resulting k- and R-space data. The Python scripting language and Qt and Qwt widget libraries were chosen to meet the design requirement that it be cross-platform (i.e. versions for Windows, Mac OS X, and Linux). PySpline supports polynomial pre- and post-edge background subtraction, splining of the EXAFS region with a multi-segment polynomial spline, and Fast Fourier Transform (FFT) of the resulting k^3-weighted EXAFS data.

Keywords: PySpline, Data reduction, EXAFS
PACS: 01.30.Cc

INTRODUCTION

PySpline was designed with the goals that it be cross-platform and developed using tools that allow rapid development. These were chosen to allow PySpline to have the broadest impact by targeting users of the three major computing platforms (i.e. Windows, Mac OS X, and Linux) while minimizing effort.

Platform-independence is important in the scientific community because of the diverse computing preferences and needs of many scientists. Fortunately, there are numerous libraries that are both freely available (under open source licenses) and cross-platform. The Qt toolkit, which provides a common programming framework for graphical interfaces and is the base for numerous commercial and open-source projects, was chosen for its maturity, rapid development, and extensive documentation. PySpline also uses the Qwt library, which provides technical "widgets" for use in Qt-based programs.

The Python programming language was chosen to avoid the time-consuming "code, compile, relink" steps of traditional programming languages like C/C++ and for its availability on all major computing platforms. It also has numerous modules, such as Numeric, that implement useful algorithms and data structures for the manipulation of matrices.

The graphical interface of PySpline has four windows for raw data and background subtraction, normalized data and splining of the EXAFS region, k^3-weighted EXAFS data, and the Fourier transform. PySpline allows users to modify various processing parameters and see the effect on the k^3-weighted EXAFS and Fourier transform in real-time.

The raw data window allows the user to specify the order and data range for the polynomial to be used for background subtraction. As the order or range of data is modified using the mouse, the background polynomial is updated and the changes are propagated through the rest of the reduction steps.

The normalized data window shows the user the results after background subtraction, splining, and normalization. The number, order, and data range of the polynomial segments in the spline can be adjusted using the spline. Any changes immediately update the spline and normalization and are propagated through the remaining reduction steps. Additionally, as the position of the "knots" are changed in eV-space, the positions are updated in k-space in the k^3-weighted EXAFS window.

The k^3-weighted EXAFS window allows simple square "windowing" of the k-space data before Fourier processing. Changes are immediately propagated into the Fourier transform window.

At any point during the raw data reduction, the parameters and results can be saved. The file format is a text file that can be read and edited by numerous programs, for example before data analysis through least squares fitting with another program. It contains comma-separated values for the raw data, background function, normalized data, spline function, and EXAFS (not k^3-weighted) data with header information containing the title, any comments specified in an optional dialog, the value for E_0, and the background, splining, and k-windowing parameters.
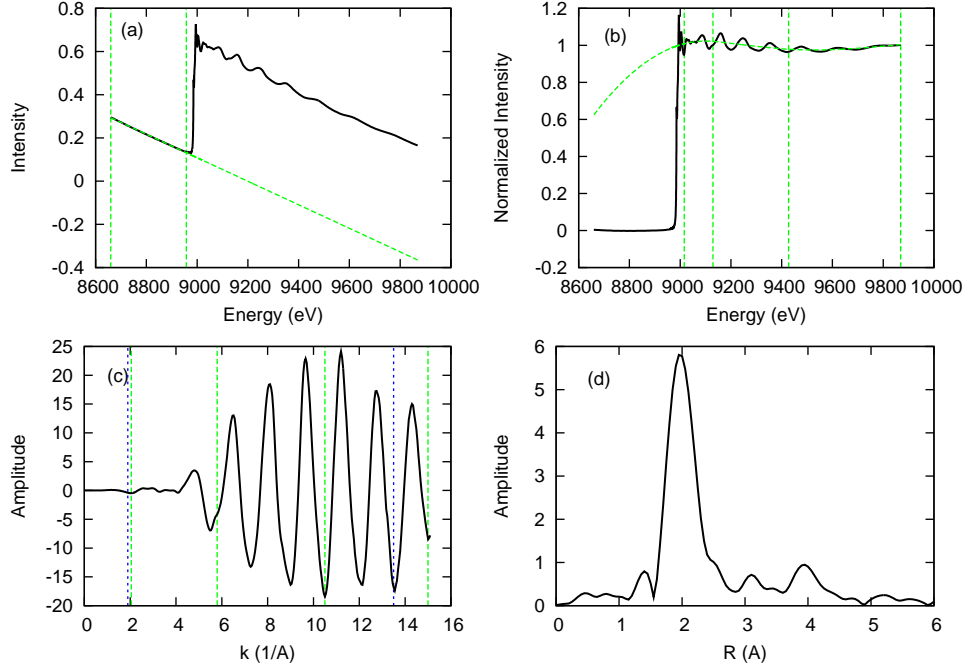
FIGURE 1. (a) Sample raw data (solid line) and a 1st-order polynomial fit to the data between the vertical lines (long dashes); (b) normalized data (solid line), and a spline with segments fitted to data between vertical lines (long dashes); (c) k^3-weighted EXAFS data (solid line) showing spline points (long dashes) and the k-window boundary (short dashes); and (d) the Fourier transform.

IMPLEMENTATION

Background Subtraction

Least-squares fitting to raw data is used to calculate coefficients of a function that models the background over a user-defined range with I data points. The general method involves the minimization of the function

$$R^2 = \sum_i^I [f(x_i, a_0, a_1, \ldots a_n) - y_i]^2 \qquad (1)$$

with respect to the coefficients $a_0, a_1, \ldots a_n$. In the case of a polynomial function, this produces a set of linear equations with the general expression

$$\frac{\partial R^2}{\partial a_j} = 2\sum_i^I x^j \left[\sum_k \left(a_k x_i^k \right) - y_i \right] = 0, \qquad (2)$$

where the function $f(x_i, a_0, a_1, \ldots a_n)$ has been replaced by a general expression for polynomials, $\sum_k(a_k x_i^k)$. PySpline uses the Numeric module to solve this set of linear equations. Example raw data and the calculated background are shown in Figure 1a.

Splining of the EXAFS Region

PySpline uses a custom algorithm to fit a multi-segment polynomial to the post-edge EXAFS region. The general methodology is to extend the linear least squares fitting to include M polynomial segments

$$F = R^2 = \sum_m^M \sum_i^I \left[\sum_k (a_{mk} x_{mi}^k - y_{mi}) \right]^2 \qquad (3)$$

with the constraints that the resulting function be continuous, $f_q(x) = f_{q+1}(x)$, and smooth, $f'_q(x) = f'_{q+1}(x)$, at the "knot" points of adjacent segments. The constraints can be written in equation form,

$$\begin{aligned} G \; = \; & \sum_q^{knots} \sum_k \lambda_q \left(a_{m'k} x_{m'q}^k - a_{mk} x_{mq}^k \right) + \\ & + \mu_q \left(a_{m'k} x_{m'q}^{k-1} - a_{mk} x_{mq}^{k-1} \right), \end{aligned} \qquad (4)$$

where λ_q and μ_q are Lagrange multipliers. Taking partial derivatives of the function $F - G$ with respect to the polynomial coefficients and the Lagrange multipliers results in a linear system of equations that can be solved with linear algebra. This approach allows the number of data points in and the order of each polynomial segment to be specified by the user.

The calculated spline function is used to normalize the background-subtracted data and extract the k^3-weighted EXAFS data. An example of normalized data and spline is shown in Figure 1b, and the resulting k^3-weighted EXAFS data are shown in Figure 1c.

Fourier Transform

PySpline applies a simple windowing function to the k^3-weighted EXAFS data before a Fourier processing. Any data outside the k-space window (short dashes, Figure 1c) is set to 0 while any data inside the window retains its value. The k-space data are then gathered into "bins" since FFT is a discrete Fourier transform method. As mentioned before, the FFT algorithm is implemented in the Numeric module. The resulting Fourier transform after applying all of the processing parameters is shown in Figure 1d.