
ATOMS 3.0

Bruce Ravel
Crystallography for the X-ray Absorption Spectroscopist

version 3.0alpha24
April 24, 2000

Abstract

ATOMS is a program for generating lists of atomic coordinates from crystallographic data. The primary use of ATOMS is to create input files suitable for running the *ab initio* XAFS program FEFF. For that purpose, the list is sorted by radial distance from an atom chosen as the central atom. ATOMS can also generate other sorts of lists as well as calculate estimations of the bulk absorption and density of the crystal and various corrections to EXAFS data due to experimental effects.

ATOMS is also a package containing the program ATOMS and various related materials. There are versions of the program ATOMS which run from a command line, use a graphical interface, or are served over the Web and accessed by a browser. All versions of ATOMS use the same set of object-oriented tools for crystallography, access to x-ray absorption databases, and access to structural data.

ATOMS is written entirely in [Perl](#), an object-oriented, interpreted language which can be run on a wide variety of platforms and operating systems. ATOMS is an Open Source (or Free Software or Software Libre, as you wish) product. This means that you can reuse or redistribute any or all of ATOMS in a manner consistent with the Perl Artistic License. In particular, this means that you may freely contribute to the development of ATOMS or use its programs and object-oriented tools for projects of your own.

This document is copyright © 1999,2000 Bruce Ravel.

Permission is granted to make and distribute verbatim copies of this manual provided the copyright notice and this permission notice are preserved on all copies.

Permission is granted to copy and distribute modified versions of this manual under the conditions for verbatim copying, provided that the entire resulting derived work is distributed under the terms of a permission notice identical to this one.

Permission is granted to copy and distribute translations of this manual into another language, under the above conditions for modified versions, except that this permission notice may be stated in a translation approved by the author.

Acknowledgments

A lot of people have contributed in some manner to the development of ATOMS. Without their gracious contributions of time, energy, advice, and criticism ATOMS would not be what it is now. A few people are in the Hall of Fame:

Matt Newville

Matt wrote the module ‘`CLdata.pm`’ and adapted all of the Fortran code that it uses from the original Brennan/Cowen source. He also created the file ‘`mcmaster.dat`’ and influenced much of the design of ‘`Absorption.pm`’. He was my earliest alpha tester and has always been my best, harshest, and most useful critic. He also hosts the very first site using *Web*ATOMS.

Tim Elam

During a sabbatical, Tim put together the most complete collection of edge and line energies and absorption cross sections that I know of. He graciously donated his work to the ATOMS project. It is the default in $\mathcal{T}\mathcal{K}$ ATOMS and the only choice in APT.

Shelly Kelly

Shelly was one of my earliest alpha testers, wrote code that inspired the Molecule panel in $\mathcal{T}\mathcal{K}$ ATOMS, and is working on code to transform geometrical information from ATOMS into useful instructions for FEFFIT.

Daniel Haskel

Dani kindly spent hours translating all of the language data used by ATOMS into Spanish, thus helping me introduce internationalization to the code.

Stéphane Grenier

Stéphane contributed the French translation of the language data.

Thorsten Buhrmester

Thorsten contributed the German translation of the language data.

I also would like to thank the people whose contributions, suggestions, and helpful bug reports have helped shape ATOMS: Chuck Bouldin, Hannes Fischer, Glenn “Full” Forney, Chris Glover, John Rehr, Hubert Renevier, Julia Wienold, Andrzej Wilamowski, my mom Fran Ravel, and my nephew Gabriel. Gabe, at age 5, is actually quite concerned with the symmetry properties of Legos and not at all with the symmetry properties of crystals, but it pleases me to mention him here nonetheless.

The original version of ATOMS benefited from the advice and support of Boyan Boyanovich, Julie Cross, Ed Stern, Hans Stragier, Kalle Voss, Steve Zabinsky, and Yanjun Zhang. One of the early inspirations for the original version of ATOMS was the program SEXIE by Bernhard Rupp.

Contents

1	Introduction	1
2	Installation	6
2.1	Unix Installation	6
2.1.1	Installing As Root	6
2.1.2	Installing As A Normal User	7
2.1.3	Installing CPAN Modules by Hand	7
2.1.4	Troubleshooting	8
2.2	Microsoft Windows Installation	9
2.3	Macintosh Installation	9
2.4	Installation on Other Platforms	10
3	Input Data	11
3.1	Crystallographic Data	11
3.1.1	Lattice Constants and Angles	11
3.1.2	Crystal Classes	12
3.1.3	Unique Positions	12
3.1.4	Shift Vectors	13
3.2	Space Groups	14
3.2.1	Bravais Lattice Types	14
3.2.2	Space Groups Symbols	15
3.2.3	The Symmetry Output File	15
3.2.4	Settings of Low Symmetry Groups	17
3.3	Operational Parameters	17
3.4	Dopants	17
3.4.1	Specifying Dopants	17
3.4.2	Dopants in ATOMS	18

3.4.3 Dopants in Other Calculations	19
4 ATOMS Output Files	20
5 Customization and Localization	23
6 Using the Command Line Version	27
6.1 The ATOMS Input File	27
6.2 Complete List of Keywords	28
6.3 Command Line Switches	30
7 Using the Tk Version	32
7.1 Overview of the Tk Version	32
7.2 The Crystallography Panel	33
7.3 The Program Panel	34
7.3.1 The Atoms Notecard	35
7.3.2 The Absorption Notecard	36
7.3.3 The DAFS Notecard	38
7.3.4 The Powder Notecard	40
7.3.5 The Molecule Notecard	40
7.4 The Output Display Window	41
7.5 The Space Group Browser	41
7.6 The Configuration Window	42
7.6.1 Configuring Variables	43
7.6.2 Configuring Colors	43
7.6.3 Configuring Fonts	45
8 Serving and Using the Web Version	46
8.1 Installing and Configuring WebATOMS	46
8.2 Using WebATOMS	48
9 APT.PL	50
10 Experimental Corrections	53
10.1 The Density and the Absorption Lengths	53
10.2 The Normalization Correction	55
10.2.1 The I_0 Correction	55

10.3 The Self-Absorption Correction	56
11 Contributing to the ATOMS Project	59
12 Resources and Bibliography	62
12.1 Literature References	62
12.2 Absorption Data Resources	62
12.3 Resources on the World Wide Web	63
A Frequently Asked Questions	64
A.1 General Questions	65
A.2 Installing ATOMS	67
A.3 Running ATOMS	70
A.4 Space Group Symbols	73
A.5 Output Files	74
A.6 The DAFS Program	75
B ATOMS Template Files	77
B.1 Introduction to the ATP Format	77
B.2 The Syntax of ATP Files	79
B.2.1 File Names	79
B.2.2 Syntax Overview	79
B.2.3 Types of ATP Tokens	81
B.2.4 Complete List of ATP Tokens	81
B.2.5 Lists in ATP Files	88
B.3 Editing Template Files with Emacs	91
C Space Group Symbols	93
D Perl Modules in the ATOMS package	106
D.1 ATOMS.PM	106
D.2 XTAL.PM	111
D.3 SGB.PM	121
D.4 ABSORPTION.PM	126
D.5 MCMaster.PM	132
D.6 ELAM.PM	133
D.7 HENKE.PM	137

CONTENTS

v

D.8	CHANTLER.PM	139
D.9	CL.PM	141
D.10	CLDATA.PM	144
D.11	CROMERMANN.PM	145
E	The Perl Artistic License	149

Chapter 1

Introduction

ATOMS is a crystallography program, but it was written by and is used by x-ray absorption spectroscopists. Absorption spectroscopy problems involving crystalline materials as the actual experimental subject or as model compounds for other, non-crystalline materials are quite common. ATOMS is primarily intended for use with the *ab initio* x-ray absorption theory code FEFF. To run FEFF one must supply a list of atomic coordinates describing the material of interest. Depending on the complexity of a structure, making a list of coordinates for a crystalline material can be a demanding and error-prone chore. Although capable of several other interesting calculations, the primary purpose of ATOMS is to generate lists of atomic coordinates starting from crystallographic data.

A crystalline material is described by its unit cell. A unit cell is one of seven space filling shapes which is decorated by the atoms contained in the unit cell. To completely describe a crystal, it is therefore necessary to describe the shape and contents of the unit cell. As an example, consider the rhombic form of elemental sulphur. The shape of this unit cell is such that it has sides of different lengths and all angles equal to 90 degrees. Describing the shape of unit cell thus takes 6 numbers, three of which are equal in this case. This form of sulphur has a very large unit cell, all the sides are over 10 Ångströms and the cell contains 128 atoms. If the origin of space is placed at one vertex of the unit cell, the atoms may be identified by coordinates which are fractions of the lengths of the sides. Thus it takes 384 numbers to describe the contents of the unit cell. To describe the shape *and* contents of the cell thus requires 390 numbers. Fortunately, there's a better way.

Crystal structures are described using the symmetry properties of three dimensional space. It turns out that there are 230 unique ways of combining symmetry operations in three dimensional space. These 230 symmetry rules are called *space groups*. A concise notation consisting of the symbol of the space group of the crystal, the values of the lattice constants and angles, and the fractional coordinates of the unique atomic positions can be used to describe any crystal.

In the case of sulphur, the space group symbol is F D D D. This is a symbol for an orthorhombic unit cell. By specifying an orthorhombic symbol, we need not specify the values of the angles. That the cell is orthorhombic implies that the angles are all 90 degrees. We do

need to specify the values of the lattice constants.

The 128 atoms in this unit cell may be divided into four sets of 32 atoms each. The atoms in each set are related to one another by the symmetry operations associated with the $F D D D$ space group. Because $F D D D$ denotes a particular set of symmetry operations, it is only necessary to specify one example from each set of 32 atoms. In each case, the other 31 positions can be generated from the first using the symmetry operations. This dramatically reduces the amount of information needed to describe the crystal. Instead of needing 390 numbers, we can use one symbol, $F D D D$, plus 15 numbers specifying the lattice constants and the fractional coordinates of the four representative atoms. These four representative atoms are the *unique atomic positions*.

Here is how sulphur is described using this crystallographic notation. The following lines are an input file for the ATOMS program. The lattice constants are the numbers **a**, **b**, and **c**. The unique coordinates are given in the list at the end of the example. The syntax and contents of this file will be explained fully in later chapters.

```

title Rhombic Sulphur
space   F D D D
rmax=10.0   a=10.467  b=12.870  c=24.493
core=s3
atom
!  At.type    x        y        z        tag
      S        0.8554   0.9526   0.9516   s1
      S        0.7844   0.0301   0.0763   s2
      S        0.7069   0.9795   0.0040   s3
      S        0.7862   0.9073   0.1290   s4

```

ATOMS knows how to interpret this kind of data and use it to fully describe a unit cell. It can then make various interesting kinds of calculations using that description of the unit cell, including its primary purpose — the generation of lists of atomic coordinates in a form suitable for running FEFF.

ATOMS is written entirely in the Perl programming language. Perl is an interpreted language. This means that ATOMS requires that you have a program called Perl on your computer which is capable of reading the source code for ATOMS and executing the commands contained therein. This is not a prohibitive requirement. Perl is available for virtually every kind of computer in use today and it is free of cost. You can download the Perl interpreter from the Internet or purchase it for minimal cost on a CD. See the installation instructions in Chapter 2 and the resources in Chapter 12 for details about where to find Perl.

This document describes the programs that come in the ATOMS distribution. That includes ATOMS, the program for creating atom lists from crystallographic data, and it includes several other programs. $\mathcal{T}_{\text{KATOMS}}$ is a graphical program which is used to generate atom lists and also to perform several other useful calculations using crystallographic data. DAFS is a program for approximating the energy dependence of the x-ray scattering form factor using tables of anomalous scattering coefficients. DAFS uses the same input files as ATOMS. *Web*ATOMS is a

version of ATOMS which runs as a CGI script on a web server. APT, the ATOMS Periodic Table is a graphical, periodic table interface to the x-ray absorption and anomalous scattering data used by ATOMS. TEMPLATE is a tiny script which generates empty FEFF input files useful for situations where you need to fill in atomic coordinates by hand.

Each of these programs is fairly short. For example, the first version of ATOMS mentioned above is about 200 lines long. The bulk of the functionality of all these programs is contained in reusable, object oriented code.

Free Software

ATOMS and all files distributed in the ATOMS package are free software. Or Open Source software. Or software libre. Different people like different terms. The bottom line is that ATOMS is freely usable, freely available, freely redistributable, and available for scrutiny and modification. This is a very sensible way for scientific software to be. The purpose of this software is as an aid in the performance of science. Once we make and interpret measurements, we typically publish our data and our interpretations in readily available sources. Science works best and is most believable when we fully disclose all aspects of our ideas, measurements, and interpretations in the scientific literature. It is therefore natural that the software that is used in pursuit of scientific accomplishment be fully disclosed in a similar manner.

While it is true that I want ATOMS to be freely and readily available for any use, I do need to protect certain personal interests. ATOMS is almost entirely my own work. This document carries my name next to its copyright notice. It is therefore important to me that others do not pass off ATOMS as their own work. It is also important that ATOMS not be modified in a manner not approved by me and this unapproved modification be passed off as my work. I also do not want anyone (unlikely as this may be) garnering some significant wealth by redistributing my work without making any significant enhancements. To this end I have placed the ATOMS package under the terms of the Perl Artistic License, a copy of which can be found in Appendix [E](#).

The use of the Artistic License means, in brief and legally inaccurate terms, that you can use ATOMS, you can give ATOMS to someone else, you can write new software which uses ATOMS in some integral manner, you can modify ATOMS for your own use, you can modify ATOMS and contribute the modification to its author, and (most importantly) you can read the source code and know exactly what ATOMS does.

One of the appealing aspects of free software is that users are free to fix bugs in the code. If you do so, please share your fix with me. If you do not want to or are unable to correct a problem, please send me a bug report. See FAQ [A.3.5](#) on page [70](#) for hints about filing useful bug reports.

Earlier versions of ATOMS were written in Fortran and were not free software. Specifically, they were software licensed from the University of Washington. While the current version of ATOMS bears certain similarities to the earlier versions, it is not a derived work. And while the ATOMS homepage on the World Wide Web is kindly hosted by The FEFF Project at the University of Washington, ATOMS is no longer managed by, distributed by, or in any way

associated with the University of Washington. Any questions or comments about the science or functionality of the code should be directed to its author, as should any questions concerning licensing or redistribution which are not covered in the Perl Artistic License, reprinted in Appendix E of this document.

About This Document

The chapters in the document are intended as user documentation for ATOMS and related programs and the appendices are intended as technical documentation. The first several chapters provide general information relevant to any program in the ATOMS package. These are followed by chapters which document the use of the command line, Tk, and Web versions of ATOMS. The final chapter contains bibliographic and web references relevant to ATOMS.

The first appendix is a Frequently Asked Questions (FAQ) list which may contain answers to questions not explicitly covered elsewhere in the document. Before reporting a bug it is a good idea to check there first. The second appendix is a technical reference on the syntax and creation of ATOMS template files, which are used to structure the various kinds of output files produced by ATOMS and are also partly described in Chapter 4. The third appendix is a big list of all the different ways to specify space group symbols. The fourth appendix contains the documentation embedded in each of the perl modules distributed with ATOMS. These are of interest to anyone interested in modifying ATOMS or in using the modules in projects of their own.

Most of this document is written natively in L^AT_EX and converted to postscript, pdf, or html. Several parts of this document, Chapter 9, and Appendices C and D, are generated automatically from other sources and so follow different typographic conventions. Elsewhere the typographic conventions outlined in Table 1.1 are used.

Table 1.1: Typographic conventions in this document.

font	denotes
SMALL CAPS	names of programs
<code>typewriter font</code>	contents of files
<code>'quoted typewriter font'</code>	file names
<i>slanted typewriter font</i>	commands executed at a command line

About the Author

My name is Bruce Ravel. I wrote this document and virtually all of ATOMS. I am a physicist by training and graduated from the X-Ray Spectroscopy Group at the University of Washington in Seattle, Washington, USA in the summer of 1997. My thesis topic was XAFS studies of

ferroelectric perovskites. I also developed a full-multiple scattering algorithm for computing XANES and local electronic densities of state which was eventually incorporated into FEFF8. I then spent two years as a National Research Fellow at the National Institute of Standards and Technology in Gaithersburg, MD, USA where I continued my studies in ferroelectric materials and also learned the Diffraction Anomalous Fine Structure (DAFS) technique. I am currently a visiting scientist at the Centre National de la Recherche Scientifique in Grenoble France and a staff member at the D2AM beamline at the European Synchrotron Radiation Source.

I love to receive email about ATOMS and encourage you to contact me with comments, questions, suggestions, criticisms, and bug reports. The address ravel@phys.washington.edu always forwards to me no matter where I am. My personal homepage is

<http://feff.phys.washington.edu/~ravel>

and the ATOMS homepage is

<http://feff.phys.washington.edu/~ravel/atoms/3.0.html>.

What the Future Holds

ATOMS is nowhere near finished. Quite the contrary, I have a long way to go. Matt Newville — the author of AUTOBK, FEFFIT, IFEFFIT and lots of other wonderful things — and I have grand ideas for a huge set of low-level tools for instrumenting the various aspects of XAFS data analysis. We also hope to build a variety of sophisticated applications using these tools so the absorption spectroscopist can interact with data and theory in many interesting ways. In particular we hope to provide a variety of programmatic, graphical, and network-based tools.

I will continue to develop the crystallographic capabilities of ATOMS. In the near future I hope to provide a simple powder diffraction simulation in the same spirit as the DAFS calculation presented in Section 7.3.3, that is as a simple and useful tool for planning experiments and as an aid at the beamline. I also plan to greatly expand ATOMS capabilities with regards to interpretation of structural data of molecules.

The data structures already in use in ATOMS are well suited to various real-space structural manipulation and representation chores. I hope to develop a simple ball-and-stick viewer as a graphical backend to ATOMS. This would allow the user to view and edit crystal and molecular structures in an intuitive manner before running FEFF or progressing on to data analysis.

I also plan to develop ATOMS as a structural front-end to IFEFFIT. ATOMS is capable of generating abstract data about crystal structures in a way that can be very powerful when combined with an EXAFS fitting engine like IFEFFIT. This will allow the user to more easily construct detailed fitting models wherein the measured parameters represent essential features of the local structure rather than are used to calculate on-the-fly the interatomic distances used in the EXAFS equation.

And finally, I intend to further develop ATOMS as a graphic and programmatic front-end to FEFF8 and its successors.

Chapter 2

Installation

The latest version of ATOMS can always be found at [the ATOMS home-page](#). You will also find collections of the various CPAN modules needed by ATOMS. The installation procedures are different on Unix, Windows, and Macintosh. These procedures are outlined in the following sections. The “downloads” page on the ATOMS web site should make it clear which files must be downloaded for which platform.

If you have any troubles installing or using ATOMS, please file a bug report. Problems only get fixed if I know about them! See FAQ [A.3.5](#) on page [70](#) for some hints on filing a good bug report.

2.1 Unix Installation

You can install ATOMS on your Unix computer as a normal user, in which case other users on the system will not normally have access to ATOMS; or as the root, in which case all users on the system can use ATOMS. The installation procedures are very similar and in either case are well automated.

You will need to have Perl 5.004 or later installed on your computer. You can test the version of Perl by typing `perl -v` at the command line. To use $\mathcal{T}\mathcal{K}$ ATOMS you will need version at least 4.02 of perl/Tk. Version 8.00 works fine and offers more features. See Chapter [12](#) for information on obtaining these packages.

The first time you install ATOMS on your computer, you will need to also install two other packages which supply functionality used by various parts of the ATOMS package. These are the files ‘AtomsBundle-#.##’ and ‘Tk800.014.tar.gz’ mentioned above. If you upgrade ATOMS in the future, you probably will not need to reinstall either of those packages.

2.1.1 Installing As Root

The installation instructions are the same for ‘Tk800.014.tar.gz’, ‘AtomsBundle-#.##’, and ‘Atoms-3.0alpha##.tar.gz’. They should be installed in that order. For each package execute

the following commands at the command line. `<<package>>` refers the three packages listed above and `<<directory>>` refers to the directory that is created when you unpack the archive. `%>` represents the command line prompt.

```
%> gzip -dc <<package>> | tar xvf -
%> cd <<directory>>
%> perl Makefile.PL
%> make
%> make test
%> make install
```

There is no test suite for the Atoms Bundle, so you don't need to run that step. There are test suites for Tk and for the ATOMS package. It is a good idea to run them both to be sure that the installation proceeded correctly.

The Tk installation is a bit time consuming since a very large amount of software needs to be compiled. Be patient.

After you install all three packages, you are ready to try ATOMS. The stand-alone program will be installed to a directory in the system executables path and is called `'atoms.pl'`. The Tk version is called `'tkatoms.pl'`.¹ The anomalous scattering simulation program is called `'dafs.pl'` and the absorption database interface is called `'apt.pl'`. Just rehash and try them out.

2.1.2 Installing As A Normal User

The installation of ATOMS as a normal user is almost the same as for root. Follow those instructions up to the final step for each package. You will not be able to do the `make install` step as a normal user. Instead, type `private-install` to install ATOMS into your own disk space. This step run a simple shell script that comes with the ATOMS distribution and requires that you have the PERL5LIB environment variable set. This points to the location where you keep your personal perl scripts, `$HOME/perl/` is a common choice. In CSH or TCSH, type `setenv PERL5LIB $HOME/perl`. In BASH, type `PERL5LIB=$HOME/perl`. You should put that variable definition in one of your login scripts, `'~/.cshrc'` for csh or tcsh, `'~/.profile'` for BASH.

Now rehash to refresh your execution path and you are ready to use the programs `'atoms.pl'`, `'dafs.pl'`, `'tkatoms.pl'`, and `'apt.pl'`.

2.1.3 Installing CPAN Modules by Hand

All of the stuff contained in the Atoms Bundle is taken from CPAN, which is on the web at [CPAN](#). The Bundle was assembled for convenience – it is easier to download and install just one package. Of course, you can also install all of the CPAN modules individually.

The modules from CPAN required by ATOMS are:

¹Some day, these will be called `'atoms'` and `'tkatoms'`. For now I want to avoid a name collision with any prior version of ATOMS that may be on your computer.

1. Statistics::Descriptive
2. Chemistry::Elements
3. File::Spec
4. Storable
5. Data::Dumper

These modules are needed to run $\mathcal{T}\mathcal{K}$ ATOMS:

1. Tk
2. Tk::FileDialog
3. Tk::Pod

These modules are needed to use the Elam tables of x-ray absorption data:

1. Math::Spline
2. Math::Derivative

The installation procedure is the same for each of these. Download the module from CPAN or from the ATOMS homepage, unpack it, and `cd` to its directory. Type `perl Makefile.PL` then `make`. Doing `make test` is a good idea in each case.

If you are root, type `make install`.

If you are a normal user, copy the script ‘`private-install`’ which comes with ATOMS to to the directory where you are building the CPAN module. Run `private-install` in each of the module directories. This script installs the perl module into your own disk space. Make sure that the PERL5LIB environment variable is set, as described in Sec. 2.1.2.

2.1.4 Troubleshooting

If you have any problems installing any part of ATOMS, please let me know. Streamlining installation procedures is one of the difficult parts of any software projects. Any feedback from users on this topic is extremely valuable.

1. The various modules from CPAN should install easily on your computer. A few of them require that you have a C compiler and all of them require that perl is already installed correctly on your computer. If you have problems with the CPAN module, contact your system administrator.
2. If ATOMS fails to make correctly, capture the screen output to a file and mail it to me at ravel@phys.washington.edu.
3. If ATOMS fails any of the tests during the `make test` part of the installation, mail the file ‘`test1.log`’ to me at [the same address](#).

2.2 Microsoft Windows Installation

You will need to have perl installed on your computer. It is free of cost and can be obtained from [ActiveState](#). Once you have perl installed, installing ATOMS is easy. These instructions are for using the Perl Package Manager (PPM). Probably all of this can be done using the Visual Package Manager (VPM), but I have never tried that.

First, go to the ATOMS homepage and download the latest Windows distribution. There is only one file to be downloaded: 'Atoms.ppd'. Download these all to the same folder on your computer. For the sake of this discussion, I will assume that you will download it to the folder 'C:\downloads'.

Double-click on the PPM icon which can be found in the 'bin' folder of your perl installation. This will pop open a DOS shell window and display the ppm> icon. In the PPM window enter the following three commands:

```
install Storable
install Data-Dumper
install Tk
```

These lines tell PPM to download these three packages from the [ActiveState website](#) and install them on your computer. This may take a while if you have a slow network connection or you are not in North America. Tk in particular is quite large.

If you are behind a firewall and having trouble downloading modules with PPM, there are instructions for getting through the firewall on the the ActiveState website.

Now unpack the file 'Atoms.zip' using WINZIP or a similar program. Double-click on the 'install.bat' icon that is unpacked from the zip file. This will install ATOMS on your computer.

That's it! You are ready to go. The programs installed are called 'atoms.pl', 'dafs.pl', 'tkatoms.pl' and 'apt.pl'. Probably the one you will use most often is 'tkatoms.pl', which is the GUI version of ATOMS. You may wish to place it in your taskbar's program menu or put a launcher on your desktop. Nice icons can be found on the [ATOMS website](#).

To upgrade to a new version of ATOMS it will probably only be necessary to reinstall ATOMS itself. That is, just download the latest version of 'Atoms.zip', unpack it, and double-click on 'install.bat'.

2.3 Macintosh Installation

You must have MacPerl installed on your computer. You can get the most recent version from [Prime Time Freeware](#). Just download the distribution and double-click on its icon to install. Choose "easy installation" when asked.

Note: at the time that this is being written the Macintosh installation is somewhat broken. I am eager for someone to help me streamline the installation on the Mac. Any volunteers?

Next you need to download some additional stuff which ATOMS needs, but which does not come with MacPerl. Get the file ‘`cpan-mac-0.32.sit.bin`’ from [the Atoms website](#) or from [Chris Nandor’s CPAN directory](#). Extract this archive and double click on the ‘`install.plx`’ icon.

Now download the Macintosh distribution of Atoms from the ATOMS web site. Double click on it to extract the archive, then double click on the ‘`MacInstall.plx`’ icon. That’s it.

Alas, Tk does not yet work on the Macintosh, so ‘`tkatoms.pl`’ and ‘`apt.pl`’ are not installed. However, double-clicking on the ‘`atoms.pl`’ icon will open up a file dialog which you can use to select an ATOMS input file.

2.4 Installation on Other Platforms

There is no fundamental reason that at least parts of ATOMS (particularly ‘`atoms.pl`’) will not install and run on other platforms than those listed above. It should be possible to use ATOMS and perhaps even \mathcal{T}_K ATOMS with VMS, OS/2, the Amiga and other platforms. If you are interested in helping me with these ports, please [contact me](#).

Chapter 3

Input Data

ATOMS requires three types of input data, the dimensions of the unit cell and locations of the unique atoms, the space group symbol, and several operational parameters controlling the type and features of the output data. All versions of ATOMS require the same kinds of data, although the manner in which the user supplies the data varies between the versions. The stand-alone version reads data from an input file, the Tk version obtains the data from Tk widgets, and the CGI version obtains the data from an HTML form. Each of the versions is described in its own chapter. This chapter explains the data types and what ATOMS expects for each one.

3.1 Crystallographic Data

The basic unit of crystallographic data is the description of the unit cell. A unit cell is described by its three lattice constants, its three angles, the positions of representative atoms within the unit cell, and the space group of the crystal. The symbolism used to describe space groups is explained in Section [3.2](#).

3.1.1 Lattice Constants and Angles

Lattice constants in ATOMS are specified in Ångströms and angles are specified in degrees. Please note that ATOMS *does not* understand minutes and seconds of arc at this time. If you have an angle of, say, 87 degrees and 30 minutes, you *must* specify this in the input file as 87.5 degrees.

It is only necessary to specify the lattice constants and angles which are unique for a given crystal class. For example, in a cubic space group it is only necessary to specify the length of the **a** lattice constant, because **b** and **c** must be the same as **a** in a cubic space group and all three angles must be 90 degrees. There is a priority among the lattice constants. In the cubic example, if you only explicitly set one lattice constant, it must be **a**. Explicitly setting only **b** will confuse ATOMS. There is no harm in explicitly setting all three lattice constants to the same value in the cubic example, but it is unnecessary.

3.1.2 Crystal Classes

The *class* of a crystal is determined by the shape of its unit cell. The space group is thus the decoration of atoms within a cell of a particular class. There are seven crystal classes, triclinic, monoclinic, orthorhombic, tetragonal, trigonal, hexagonal, and cubic. There is one other word used to describe a unit cell shape – rhombohedral. Rhombohedral cells are actually trigonal cells with additional symmetry.

The following table explains the shapes of the seven crystal classes in terms of the lattice constant and angles.

Table 3.1: Lattice constants for the crystal classes. See Section 3.2.4 for details about alternate settings and unique angles besides β for monoclinic groups, alternate settings of orthorhombic groups, and rotated settings of tetragonal groups. The rhombohedral settings of trigonal groups are explained below.

group	lattice constants and angles
triclinic	$a \neq b \neq c$ and $\alpha \neq \beta \neq \gamma \neq 90^\circ$
monoclinic	$a \neq b \neq c$ and $\alpha = \beta = 90^\circ, \gamma \neq 90^\circ$
orthorhombic	$a \neq b \neq c$ and $\alpha = \beta = \gamma = 90^\circ$
tetragonal	$a = b \neq c$ and $\alpha = \beta = \gamma = 90^\circ$
trigonal	$a = b \neq c$ and $\alpha = \beta = 90^\circ$ and $\gamma = 120^\circ$
hexagonal	$a = b \neq c$ and $\alpha = \beta = 90^\circ$ and $\gamma = 120^\circ$
cubic	$a = b = c$ and $\alpha = \beta = \gamma = 90^\circ$

Rhombohedral cells are a special case of the trigonal class. They can be specified using the rhombohedral parameters **a** and α or the trigonal parameters **a** and **c**. When a rhombohedral space group is specified, ATOMS will check the supplied lattice parameters and do the right thing. There is a functional relationship between the rhombohedral **a** and α and the trigonal **a** and **c**. Consult chapter ?? of the International Tables of Crystallography (see Chapter 12) for details.

3.1.3 Unique Positions

ATOMS requires a list of unique atom positions to fully describe the unit cell of a material. It is important to appreciate the difference between unique positions and the set of positions in the unit cell. In the example of sulphur given in Chapter 1, the ‘atoms.inp’ file contains a list of four unique positions, whereas the unit cell of rhombic sulphur contains 128 atoms. The relationship between the four unique positions and the 128 actual positions is conceptually simple. The space group (see Section 3.2) of the material implies a certain set of symmetry operations. These symmetry operations are applied to each of the unique positions to generate all the actual locations of atoms in the unit cell. The beauty of the notation using a space group symbol and the list of unique positions, and thus the beauty of ATOMS is that it is compact yet complete.

When structures are reported in the crystallographic literature or in a reference source like the Wyckoff tables, only the unique positions are given. And that is all the information needed by ATOMS. The manner in which the list of unique positions is entered is explained for ATOMS in Chapter 6 and for $\mathcal{T}\mathcal{K}$ ATOMS in Chapter 7.

3.1.4 Shift Vectors

If you browse through the first volume of the International Tables of Crystallography in the part where the symmetry operations for the 230 space groups are spelled out in detail, you will see that there are 25 groups for which the symmetry operations are referenced to a choice of two origins. That is, there are two choices for which center of point symmetry is placed at the position (0,0,0). In the literature, you will find that some authors use one choice of origin and some use the other.

ATOMS has a preference between the two choices for each of these 25 space groups. In each case, ATOMS expects that you give the positions referenced to the *origin at centre*.¹ The unfortunate part of this is that, while ATOMS has a preference for which origin you choose, it has no way of knowing if you have, in fact, chosen the correct origin.

While ATOMS cannot know if you have chosen the correct origin, it does recognize when you use one of the 25 space groups and issues you a warning which includes the correct value of the shift vector that you should use if your data are referenced to the wrong origin. Figure 3.1 shows an example of this for a common space group. $\mathcal{T}\mathcal{K}$ ATOMS issues a pop-up dialog with a similar message. In *Web*ATOMS, this message is written to the top of the page served upon output.

```
=====
Atoms 3.0alpha21 (linux) 1999/10/17
=====

title > alpha-tin  diamond structure

Space group "f d -3 m" appears in the International Tables with
multiple choices of origin.  If the atoms list seems wrong, you
should use a shift vector of "-0.125, -0.125, -0.125".

feff: Writing file to feff.inp
=====
```

Figure 3.1: Screen output of ATOMS for one of the 25 space groups possibly requiring a shift vector.

There are several clues that you might need to use the shift vector. The most obvious clue is usually that the interatomic distances and shell coordinations do not agree with those

¹*origin at centre*, complete with the British spelling, is the term used in the International Tables.

published in the literature. You might also find that the number ATOMS reports for the density of the material is absurdly small or absurdly large. Also the stoichiometry reported in the P1 file output option will likely be wrong.

When you see the shift vector warning and the output data are, in fact, wrong, it usually suffices to rerun ATOMS with the correct shift vector. See Chapters 6 and 7 for instructions on how to specify the shift vector in the command line and Tk versions.

Sometimes a shift vector is needed even for a space group that is not in the list of 25. A common example is SiO₂ (GeO₂ is of the same structure). A nice study of the structure can be found in *Acta Crystallographica B* 32 (1976) pages 2456–2459. In that article, the authors report the following structural data, which I reproduce as in input file for ATOMS.

```

title SiO2 alpha
title Acta Cryst. B32 (1976) 2456-2459
space P 32 2 1
rmax 8.          a 4.9134          c 5.4052
core=Si1         p1=t             ! geom=t
atom
  Si    0.46987 0.0    0.0
  O     0.4141  0.2681 0.1188

```

Reading this article carefully² you will find that the authors have chosen to place their origin at a different location than that given for space group P 32 2 1 in the International Tables. They have shifted the origin by 2/3 of a lattice constant along the \hat{z} direction. To compensate for this odd choice, it is necessary to give ATOMS the following information:

```

shift  0  0  2/3

```

The last two paragraphs are the answer to one of the most common questions I have received about ATOMS over the years. The moral of the story is, I think, to read your literature references carefully and not to trust authors to publish their data in the way that will be most convenient for your use!

3.2 Space Groups

3.2.1 Bravais Lattice Types

The Bravais type of space group specifies a particular translational symmetry. The easiest way to explain the Bravais translation is with an example. In FCC copper, you only need to specify one unique position, (0,0,0). In the copper unit cell, there are four atoms. They are at (0,0,0), (1/2,1/2,0), (1/2,0,1/2), and (0,1/2,1/2). If you examine the point symmetry operations indicated by the FCC space group, which is $Fm\bar{3}m$, you will find that each operation on the position (0,0,0) generates that same position. None of the point symmetry

²Pay particular attention to the second paragraph on page 2457.

operations generate the other four positions. This is because those other positions are not related to (0,0,0) by point symmetry, but rather by the Bravais translation. Space groups whose Hermann-Maguin symbols begin with F all have three Bravais translation vectors, $(1/2, 1/2, 0)$, $(1/2, 0, 1/2)$, and $(0, 1/2, 1/2)$ in addition to (0,0,0). So any positions generated by application of point symmetry are also translated by the Bravais vectors. In the case of an F group, the number of symmetry related positions in the unit cell is thus four times the number of positions generated by the point symmetry operations.

Table 3.2: Bravais lattice vectors. All Bravais types also have the default (0,0,0) vector.

symbol	vectors
P	none
F	$(\frac{1}{2}, \frac{1}{2}, 0)$, $(\frac{1}{2}, 0, \frac{1}{2})$, $(0, \frac{1}{2}, \frac{1}{2})$
I	$(\frac{1}{2}, \frac{1}{2}, \frac{1}{2})$
C	$(\frac{1}{2}, \frac{1}{2}, 0)$
A	$(0, \frac{1}{2}, \frac{1}{2})$
B	$(\frac{1}{2}, 0, \frac{1}{2})$
R	$(\frac{2}{3}, \frac{1}{3}, \frac{1}{3})$, $(\frac{1}{3}, \frac{2}{3}, \frac{2}{3})$

3.2.2 Space Groups Symbols

This section has not been written yet.

3.2.3 The Symmetry Output File

One of the standard output file options as described in Chapter 4 and Appendix B is a file which describes the symmetry operations of a space group. Examining one of these files will help you understand how ATOMS turns crystallographic input data into a description of the unit cell. An example of this file is shown in Figure 3.2 for the orthorhombic space group, $I m m 2$.

The Bravais vectors from Table 3.2 are shown on the sixth line and all point symmetry related positions are given in the table.

Suppose that an atom has the unique position (0.5, 0, 0.5053)³. The symmetry properties are applied to this unique position as stated in Figure 3.2. Since there are two Bravais vectors and four symmetry related positions, the one unique position yields eight possible positions. These are

(0.5, 0, 0.5053)

³The example atom positions in this section are oxygen position in the mineral hemimorphite ($Zn_4Si_2O_7(OH)_2 \cdot H_2O$), which is of the space group $I m m 2$

This symmetry file was generated by Atoms 3.0alpha21
 Atoms written by and copyright (c) Bruce Ravel, 1998, 1999

The symmetry table for space group "I m m 2" (I m m 2).

The Bravais translation vectors:

(0, 0, 0), (1/2, 1/2, 1/2)

There are 4 symmetry related positions.

x_position	y_position	z_position
x	y	z
-x	-y	z
x	-y	z
-x	y	z

Figure 3.2: Example content of a 'symmetry.dat' file.

```
(-0.5, -0, 0.5053)
( 0.5, -0, 0.5053)
(-0.5, 0, 0.5053)
( 1.0, 0.5, 1.0053)
( 0, 0.5, 1.0053)
( 1.0, 0.5, 1.0053)
( 0, 0.5, 1.0053)
```

Translating all of the positions back into the first octant so that all coordinates are between 0 inclusive and 1 exclusive, yields

```
( 0.5, 0, 0.5053)
( 0.5, 0, 0.5053)
( 0.5, 0, 0.5053)
( 0.5, 0, 0.5053)
( 0, 0.5, 0.0053)
( 0, 0.5, 0.0053)
( 0, 0.5, 0.0053)
( 0, 0.5, 0.0053)
```

Some of these positions are generated repeatedly as my example is a position of high symmetry in the x and y directions, thus the list of positions generated by this unique site is

```
( 0.5, 0, 0.5053)
( 0, 0.5, 0.0053)
```

For a position of low symmetry in this space groups, say (0.1588, 0.2075, 0.6375), the eight positions generated will be distinct.

3.2.4 Settings of Low Symmetry Groups

This section has not been written yet.

3.3 Operational Parameters

In general there are a few more pieces of information that you must feed to ATOMS in order for it to correctly build a cluster or perform some other calculation. For example, you may wish to specify a radial dimension for a cluster or specify a particular absorption edge. For many such operational parameters there are sensible default values. There are situations, though, where either it is desirable or necessary to explicitly specify non-crystallographic parameters. These operational parameters are explained for ATOMS in Chapter 6 and for $\mathcal{T}\mathcal{K}$ ATOMS in Chapter 7.

3.4 Dopants

The manner in which ATOMS handles dopants is complicated and possibly confusing. Of course, the manner in which dopants must be considered in EXAFS analysis is also complicated and possibly confusing. That vindicates the fact that dopants in ATOMS are confusing, but certainly doesn't justify it!

There is an important distinction you must keep in mind while reading this section of the document. The first is between how dopants are specified among the input data to ATOMS and how the dopant data is used by ATOMS to generate its output. In some situations you can specify dopants more thoroughly in the input data than ATOMS can use them to generate certain kinds of output data. Also keep in mind that I am not a crystallographer and I never intend that ATOMS be used to solve tough crystallography problems such as Rietveld refinements.

3.4.1 Specifying Dopants

The simplest form of dopant is a vacancy, so I will start by explaining how to tell ATOMS about a vacancy. By a vacancy I mean that a crystallographic site is incompletely occupied, i.e. that some positions in real space generated by that crystallographic site are not occupied by any atom. You tell ATOMS about this by specifying an occupancy for a site of less than 1.

In $\mathcal{T}\mathcal{K}$ ATOMS, this is done by manipulating the occupancy slider in the lower panel of the $\mathcal{T}\mathcal{K}$ ATOMS window. You can click on the slider and drag to set the value. Clicking *mouse-1* in the trough will increment or decrement the slider value by 0.01. In this way you can specify occupancies between 0 and 1.

In an input file, occupancies can be specified in the sixth column of the atoms list. As explained in Sec. 3.1.3, the first column is used for the atomic symbol, the next three for the fractional coordinates, and the fifth for the optional site tag. A number between 0 and 1 can be specified in the sixth column. If a site does not have a tag and the fifth column can be interpreted as a number, then ATOMS assumes that the number is an occupancy. Thus it is

prudent to use at least one non-numeric character in your site tags.

Now we consider a simple substitution, i.e. the replacement of a fraction of the atoms occupying a crystallographic site with an atom of another species. This is handled in a manner very similar to a vacancy. First specify an occupancy of less than 1 for the host atom. Next, create a new crystallographic site which has the same fractional coordinates as the host atom. In $\mathcal{T}_{\mathcal{K}ATOMS}$ this is done by repeating the coordinates in the subsequent row of the site table in the bottom panel. In an input file, this is done by replicating the coordinates in the next line of the atoms list. For the dopant site, specify the appropriate atomic symbol and an appropriate tag. Then specify the doping fraction using the occupancy slider in $\mathcal{T}_{\mathcal{K}ATOMS}$ or the sixth column in the input file. IN this manner you may specify any number of dopants. The occupancies of the host and its dopants must sum to a number less than 1. $ATOMS$ will flag an error if the occupancies sum to greater than 1.

$ATOMS$ uses a simple scheme to distinguish between host and dopant atoms in the case of simple substitution. Whichever atomic species is encountered first is taken as the host species. In $\mathcal{T}_{\mathcal{K}ATOMS}$ this means that the host is closer to the top of the sites table in the bottom panel. In an input file, the host is the earlier line in the atoms list. To switch host and dopant, simply change the order in which they appear.

$ATOMS$ is also able to consider a complex substitutional dopant in which the substitute atom replaces the host atom, but not in exactly the same position. This is specified in the manner described above, except that the dopant need not have the same coordinates as the host. There are two important caveats to using $ATOMS$ in this manner. The first is if you use $ATOMS$ to make an input file for $FEFF$. As explained in Sec. 3.4.2, this will result in a very unsatisfactory input file. The second is that the check made by $ATOMS$ that the occupancies sum to less than 1 will not be made correctly, thus it is easy to specify nonsensical input data.

There are, of course, other phenomena which may be lumped under the term “dopant”. For example, one might consider interstitial or grain boundary intrusions as dopants. $ATOMS$ makes no attempt to consider these sorts of phenomena in any of its calculations.

3.4.2 Dopants in atoms

The primary purpose of $ATOMS$ is to generate input data for $FEFF$. These data are finite lists of real-space atomic coordinates. For the sake of a $FEFF$ calculation there is no point in generating a list of a radial extent about the central atom that is much larger than a small multiple of the photo-electron mean free path. Typically a cluster containing all atoms no farther than 10 Å from the central atoms quite suffices. A list of that size contains no more than a few hundred atoms. In EXAFS analysis, it is rare to consider atoms farther than about 6 Å from the central atoms, i.e. no more than a few tens of atoms. While the concept of “fractional occupancy” is quite well defined in crystallography, its meaning is much more ambiguous when considering only a few tens of atoms.

Consider, for example, analyzing EXAFS data taken on metallic copper doped with 8.3% silver. Since copper is an FCC metal and the first coordination shell contains 12 atoms, it might be sensible to replace one of the first shell atoms with a silver atom. However the second

shell contains only six atoms. There is no way in the FEFF input file to replace a half an atom. Suppose that there was 10% silver in the copper. Thus 1.2% of the first coordination shell is silver, on the average, but again there is no way to specify a fractional atom in FEFF's input file. This is not say that you cannot use FEFF to analyze data taken on a doped material. Quite the opposite! FEFF is an excellent tool for doing so, but the techniques are more complicated than simply choosing atoms from the atoms list to replace with dopants. A more detailed discussion of those techniques is given elsewhere (*give the URL*).

The bottom line is that there is no simply stated algorithm for constructing a finite sized atom list in the 'feff.inp' file which properly considers dopants. Any consideration of dopants in EXAFS analysis using FEFF requires that the FEFF user expend effort that cannot be coded into ATOMS.

When ATOMS writes a list of atomic coordinates of the sort used in a 'feff.inp', it ignores the dopant atoms. All sites are occupied by the host atoms.

However, ATOMS does not completely discard the information about dopants when writing 'feff.inp' files. That information is used in a variety of ways. A dopant atom may be chosen as the central atom. In that case, the first item in the atoms list and the first item in the potentials list is the dopant atom. If an edge symbol is not specified, ATOMS will select a default edge appropriate to the dopant core. Finally the results of the absorption calculations, which may be written in the output file, will consider all dopants (not merely the central atom dopant) specified in the atoms list.

3.4.3 Dopants in Other Calculations

There are other output file types available to 'atoms.pl' or to the ATOMS notecard in $\mathcal{T}_{\mathcal{K}ATOMS}$. Most of these handle dopants in a more natural manner. For example the unit cell and P1 cell output options (see Chapter 4) report the host and all dopants along with their occupancies.

The other calculations available in $\mathcal{T}_{\mathcal{K}ATOMS}$ (and other programs in the ATOMS distribution) make use of dopants in a much more straight-forward manner than is true of the creation of 'feff.inp' files. The absorption calculations, the DAFS simulation, and the (as yet unwritten) powder diffraction simulation all make use of the dopant data exactly as stated in the sites table or atoms list. For example, the contribution of an atom to an absorption calculation is multiplied by its occupancy. Similarly, the contribution of an atom to the DAFS scattering factor is multiplied by its occupancy.

Chapter 4

ATOMS Output Files

The main purpose of ATOMS is to write input files for FEFF. However the input files for FEFF6 and FEFF8 are slightly different. Fortunately ATOMS comes with several predefined output file types, including FEFF6, FEFF8, and several others. Even better, ATOMS uses flexible template files mechanism to structure its output files. Thus you can modify the existing output files types and even create entirely new ones. Complete details for doing so are given in [Appendix B](#).

The output file types that come with the ATOMS distribution are:

feff6 This produces an input file the is suitable for running any of versions 5, 6, or 7 of FEFF.

feff8 This produces an input file the is suitable for running FEFF8.

alchemy

This produces a file of atomic coordinates in the alchemy format, and so can be used by many ball-and-stick viewers as input data.

xyz This produces a file of atomic coordinates in the xyz format, and so can be used by many ball-and-stick viewers as input data.

unit This produces a description of the unit cell which I refer to as “overfilled”. This means that atoms which are located near the walls, edges, and corners of the unit cell are copied to the other side of the cell. For example, an overfilled cell for perovskite EuTiO_3 would would have an Eu atom at each of the eight corners, an oxygen atom on each of the six walls, and a Ti atom in the center of the cube.

p1 This produces an input file suitable for running ATOMS but with a description of the unit cell using the P 1 space group and the entire contents of the unit cell. This means that all of the unique coordinates have been expanded and all lattice constants and angles are explicitly specified.

symmetry

This file type displays a list of the symmetry operations of the chosen space group. An example is shown in [Figure 3.2](#).

-
- geom** This file produces a ‘`geom.dat`’ file suitable for use with FEFF’s pathfinder. Note that this output file is only correctly made for orthogonal space groups. Unless you know exactly what this file is used for by FEFF, it is unlikely that you actually need it.
- dafs** This produces the output from the DAFS calculation made by ‘`dafs.pl`’ or the DAFS notecard in $\mathcal{T}\mathcal{K}\text{ATOMS}$.
- atoms** This rewrites the input data in the form of a proper ATOMS input file. This is primarily used by $\mathcal{T}\mathcal{K}\text{ATOMS}$ and *Web*ATOMS for saving input data.
- absorption**
This writes a report of the results from the Absorption notecard in $\mathcal{T}\mathcal{K}\text{ATOMS}$.
- molecule6**
This writes a FEFF6 input file for a molecule. It is used by the Molecule notecard in $\mathcal{T}\mathcal{K}\text{ATOMS}$.
- molecule8**
This writes a FEFF8 input file for a molecule. It is used by the Molecule notecard in $\mathcal{T}\mathcal{K}\text{ATOMS}$.
- template6**
This writes an empty template for a FEFF6 input file.
- template8**
This writes an empty template for a FEFF8 input file.
- gnxas_cry**
This writes a file containing input data for the GNXAS CRYMOL program using the “CRY” format.
- gnxas_sym**
This writes a file containing input data for the GNXAS CRYMOL program using the “SYM” format.
- formulas**
This file produces a file containing formulas for the positions of every atom in the cluster. This is a test bed for a future interface between ATOMS and IFEFFIT.
- test** This file produces a diagnostic file useful for testing ATOMS and for submitting bug reports.

The way of selecting output file types is discussed in detail for the command line, Tk, and Web versions of ATOMS in Chapters 6, 7, and 8. In brief, the output file type can be selected in an ATOMS input file using the **output** keyword. In the command-line version of ATOMS, there are command-line switches for selecting the output type. In the Tk version there are radio buttons which are used to select the file type.

Although the output file types described above are the only ones distributed with ATOMS, there actually are no hardwired output types in the ATOMS source code. All output files are

generated from external files called ATOMS template files, or atp files. ATOMS reads the appropriate atp file and using the encoding information contained in it to generate the appropriate output file.

Because all output files are generated from external files, the user can easily modify the existing output files types or even create entirely new ones. On a multi-user system, you may keep a library of modified or newly created atp files in your personal ATOMS directory.¹ This directory is searched before the system-wide directory, so when a personal atp file has the same name as a system-wide atp file, the personal one will be used by ATOMS.

The best way to modify one of the standard output file types is to copy the atp file from the ‘atp/’ directory in the ATOMS installation location to your personal ATOMS directory. On systems like the Macintosh and Windows 9x which do not have personal disk space, just edit the atp files in the ATOMS installation.

The format of the template files is described in detail in Appendix B, and Section A.5 answers some common questions about using atp files and controlling the output of ATOMS.

¹‘\$HOME/.atoms/’ on Unix, *undecided* on VMS, and *undecided* on Windows NT.

Chapter 5

Customization and Localization

There are two ways of customizing the behavior of ATOMS. One is to use the output template files described in Chapter 4 and Appendix B to control the output data from an ATOMS run. The other is to use the runtime configuration file to control how ATOMS behaves while running. The configuration options in the runtime configuration file are the topic of this chapter.

The default runtime configuration file is found in the ‘lib/’ subdirectory where ATOMS is installed. It is called ‘atomsrc’. When ATOMS starts running, one of the first things it does is to read this file and use its contents to set certain defaults. On a multiuser system such as Unix, VMS, or Windows NT, the individual user can keep a runtime configuration file which is different from the system-wide file. ATOMS first reads the system-wide file, then searches for the user’s personal file. If found, the personal file is used to override the settings from the system-wide file. On any kind of Unix machine, the personal runtime configuration file is kept as ‘\$HOME/.atoms/atomsrc’. \$HOME represents the user’s home directory. The user’s runtime configuration file is kept in the same ‘.atoms/’ directory where custom atp files (see Chapter 4) are kept. On a VMS machine, the user’s runtime configuration file is kept in *undecided?*. On a networked Windows NT machine it is kept in *undecided?*. On Macintoshes and Windows 9x machines, there is no concept of a user directory thus ATOMS does not search for a user’s runtime configuration file

The runtime configuration file uses Perl’s syntax and is interpreted by ATOMS as perl code. If you are unfamiliar with Perl’s syntax, just follow the examples given in the supplied ‘atomsrc’ file when you modify it. The best way to make a personal runtime configuration file is to copy the system-wide file to your ‘.atoms/’ directory and edit it as desired.

This list describes the things that can be customized using the runtime configuration file. See Section 7.6 for an explanation of the configuration utility in $\mathcal{T}\mathcal{K}$ ATOMS, which can be used to modify the variables on the fly.

Of all these variables only `$prefer_feff_eight` has any effect on *Web*ATOMS, although in the future cookies might be used to replicate the behavior of `$atoms_language`.

`$atoms_language`

The language of the textual data used by ATOMS. The default is English. Currently

available options include Spanish and French.

\$write_to_pwd

When this evaluates to true, it means that output files are written by default to the current working directory. If false, it means that output files are written to the location of the input file used. It defaults to 1 (true) which is the more appropriate choice for Unix systems. On the Macintosh, 0 is a more appropriate value. This is not used by $\mathcal{T}\mathcal{K}\mathcal{A}\mathcal{T}\mathcal{O}\mathcal{M}\mathcal{S}$.

\$prefer_feff_eight

If no output file type is explicitly requested, ATOMS writes an input file for FEFF. If this variable is 1, then ATOMS will write an input file for FEFF8 by default. If 0, an input file for FEFF6 will be written. The default is 0.

\$always_write_feff

Setting this to 1 makes ATOMS behave more like the old Fortran version with regard to how it chooses what output files to write. The default value, 0, tells ATOMS to only write the output files which are requested. This is not used by $\mathcal{T}\mathcal{K}\mathcal{A}\mathcal{T}\mathcal{O}\mathcal{M}\mathcal{S}$.

\$absorption_tables

This tells ATOMS which tables of x-ray absorption data to use for calculations which require them. The default is “Elam”. The other options are “McMaster”, “Henke”, “Chantler”, and (if you have installed the CLkit), “CL”.

\$dafs_default

This specifies the default table to use in a dafs calculation when the currently selected absorption resource is not one which provides anomalous scattering factors. The default is “CL” if the CLkit is installed and “Henke” if it is not.

\$default_filepath

If you want the file dialog in $\mathcal{T}\mathcal{K}\mathcal{A}\mathcal{T}\mathcal{O}\mathcal{M}\mathcal{S}$ to always display a particular directory whenever it opens up, set this variable to that directory. The default behavior of the file dialog is to display perl’s current working directory, however that behavior may be inconvenient under Windows or when launched from the desktop under Unix.

\$unused_modifier

In $\mathcal{T}\mathcal{K}\mathcal{A}\mathcal{T}\mathcal{O}\mathcal{M}\mathcal{S}$ the sites table can be navigated using the arrow keys and a modifier key. However, it is possible that your window manager uses **<Shift-arrow>** for some other purpose. You can set this to ‘Shift’, ‘Control’, ‘Alt’, or ‘Meta’, whichever works best for you.

\$display_balloons

Setting this to 1 tells $\mathcal{T}\mathcal{K}\mathcal{A}\mathcal{T}\mathcal{O}\mathcal{M}\mathcal{S}$ to display help balloons whenever the mouse lingers over a widget for 1/2 second or longer. Setting it to 0 suppresses all help balloons.

\$no_crystal_warnings

Setting this to 1 tells $\mathcal{T}\mathcal{K}\mathcal{A}\mathcal{T}\mathcal{O}\mathcal{M}\mathcal{S}$ not to display certain warning messages. Currently the messages suppressed by setting this variable to 1 are the ones about lattice parameters that do not match the space group and the one suggesting a shift vector.

\$one_frame

Setting this to 1 tells $\mathcal{T}\mathcal{K}\text{ATOMS}$ to place the program notecards and the crystallography panel in a single window. If it is set to 0, then the notecards and the crystallography panel will each get their own window. This may be helpful on small monitors or on monitors with low resolution.

\$c_??? These variables specify various colors used by $\mathcal{T}\mathcal{K}\text{ATOMS}$. The best way to understand what effect they have is to use the configuration tool in $\mathcal{T}\mathcal{K}\text{ATOMS}$, as described in Section 7.6.2. You may use RGB triplets to specify the colors or any logical name recognized by your computer.

\$f_??? These variables specify the various fonts used by $\mathcal{T}\mathcal{K}\text{ATOMS}$. The best way to understand what effect they have is to play around with their values and restart $\mathcal{T}\mathcal{K}\text{ATOMS}$. Unfortunately $\mathcal{T}\mathcal{K}\text{ATOMS}$ does not yet have a mechanism for changing the fonts on the fly. You may use X-style font names or any logical name recognized by your computer.

```
=====
Atoms 3.0alpha21 (linux) 1999/10/17
=====
titre > YBCO: Y Ba2 Cu3 O7
feff: Ecrit le fichier dans feff.inp
=====
```

Figure 5.1: Screen output of ATOMS in French.

```
=====
Atoms 3.0alpha21 (linux) 1999/10/17
=====
titulo > YBCO: Y Ba2 Cu3 O7
feff: Escribiendo archivo en feff.inp
=====
```

Figure 5.2: Screen output of ATOMS in Spanish.

The most powerful feature of the configuration file is the ability to set the language used by ATOMS during operation. This is demonstrated in Figures 5.1 and 5.2. By setting the `$atoms_language` variable to “French” or “Spanish”, virtually every text string written by ATOMS or $\mathcal{T}\mathcal{K}\text{ATOMS}$ will be written in that language. It is my intention that ATOMS be a true polyglot. Currently I only have translations into French and Spanish, however I will happily add any new language to ATOMS if I am given a translation. From the experiences of the people who made the French and Spanish translations, it takes a native speaker with a good command of English about 4 hours to translate all the text data. With the release of perl 5.6 (hopefully before Spring of 2000), Unicode will be fully supported by perl and perl/Tk. This will give

ATOMS the ability to speak virtually any language, including many Asian languages. Anyone interested in translating the text data into their native tongue should **contact me**.

Chapter 6

Using the Command Line Version

The stand-alone version of ATOMS, `atoms.pl`, is intended to be run from a command line and behaves very similarly to the Fortran version of ATOMS. Specifically, any input files that you used with the Fortran version can be used with this version.

To use this version of ATOMS, just type `atoms.pl` at the command line. If there is a file called `atoms.inp` in the current directory, it will be read and used to generate output. Alternately you specify any file name in any directory by explicitly mentioning it as an argument on the command line:

```
atoms.pl other_file.inp
```

You can read from standard input by giving `-` as the filename. This reads input until the end of the redirected file or until you type `^D`.

```
cat some_file | atoms.pl -
```

You can also use command line switches as described in Section 6.3. This example:

```
atoms.pl -pF other_file.inp
```

tells ATOMS to write a P1 file using the data in a specific file, but not to write a `feff.inp` file.

6.1 The ATOMS Input File

The input files for ATOMS is a loosely formatted flat text file. Most of the file consists of keyword/value pairs, such as

```
rmax = 5.0
```

Keywords are separated from values by whitespace, commas, and equals signs. The exact definition of what separated keywords and values is “zero or more tabs or spaces; followed by a

single comma, equals sign, tab or space; followed by zero or more tabs or spaces.” As a regular expression this is `[\t]*[\t=,][\t]*`.

The last keyword in the file must be **atoms**, which tells ATOMS that the list of unique coordinates begins on the next line. Everything after the keyword **atoms** is assumed to be part of the unique coordinates list and any keywords appearing after **atoms** will not be interpreted correctly.

Here is an example of an input file for superconducting yttrium barium copper oxide:

```

title YBCO: Y Ba2 Cu3 O7
space P M M M
rmax=5.2          a 3.823    b 3.886 c 11.681
core=cu2
atoms
  Y      0.5      0.5      0.5
  Ba     0.5      0.5      0.184
  Cu      0        0        0          cu1
  Cu      0        0      0.356      cu2
  O       0        0.5      0          o1
  O       0        0      0.158      o2
  O       0        0.5      0.379      o3
  O      0.5        0      0.377      o4

```

6.2 Complete List of Keywords

% ! *

Comment characters. Anything on a line after one of these characters will be ignored by ATOMS.

atoms This keyword marks the end of the keyword/value pairs and tells ATOMS that the list of unique atomic coordinates begins on the following line. This list must be formatted in columns. The first column contains the two letter element symbol of the atom occupying the site. The next three columns are the x , y , and z coordinates expressed as fractions of the a , b , and c axis lengths. The optional fifth column contains a tag which is used to uniquely identify the site. The tag in the fifth column is used as the value of the **core** keyword. The coordinates can be numbers or simple math expressions like $1/2$ or $0.5+0.02$.

a The length of the a lattice constant in Ångströms.

b The length of the b lattice constant in Ångströms. If not specified, this is set to be equal to a .

c The length of the c lattice constant in Ångströms. If not specified, this is set to be equal to a .

- alpha** The angle between the *b* and *c* axes in degrees. If not specified, this is set to be equal to 90 degrees.
- beta** The angle between the *a* and *c* axes in degrees. If not specified, this is set to be equal to 90 degrees, unless the space group is rhombohedral, in which case it is set equal to alpha.
- gamma** The angle between the *a* and *b* axes in degrees. If not specified, this is set to be equal to 90 degrees, unless the space group is rhombohedral, in which case it is set equal to alpha, or hexagonal or trigonal, in which case it is set to 120 degrees.
- title** User supplied title lines which may be copied to the output files. Everything on the line following this keyword is considered to be part of the title line. Keywords which appear after **title** will not be interpreted as keywords. These two words are synonyms and may be used interchangeably. **comment** is a synonym for **title**.
- rmax** The radial size of the atomic cluster written to the output files. If not specified, the default value is 7 Ångströms or 1.1 times the length of the shortest axis, whichever is longer.
- core** The tag of the central atoms. This must be specified unless the atom list only contains one atom. These two words are synonyms and may be used interchangeably. **central** is a synonym for **core**.
- edge** The symbol of the absorption edge. If not specified, this is set to K for elements lighter than lanthanum and L3 for heavier elements.
- output** Used to specify the file type and file name of an output file. This keyword may be used any number of times in an input file. The syntax is

```
output    type    filename
```

where **type** is the name of a valid atp file without the file extension. **filename** will be the name to which the output file is written. An example:

```
output    feff8    my_feff8_file.inp
```

See Chapter 4 for more details about available output file types.

- shift** This takes three numbers specifying the vector needed to shift a set of unique coordinates to the correct origin as used in the International Tables of Crystallography. The syntax is

```
shift    x    y    z
```

The shift coordinates can be numbers or simple fractions like 1/2 and 2/3. See Section 3.1.4.

nitrogen, argon, krypton

These keywords take the volume percentages in the I0 chamber of these gasses. Specifying one or more of these tells ATOMS to calculate the self-absorption and I0 corrections along with the McMaster correction. By default these are all 0 and the two additional corrections are not made.

feff, feff8, p1, geom, unit

These are boolean keywords and take either **true** or **false**. When true, they tell ATOMS to write that type of output file to the default filename. These are included for backwards compatibility to the old Fortran version of ATOMS and are considered to be obsolete. Use the **output** keyword or the command line switches (see Section 6.3) instead.

The following keywords were recognized by the Fortran version of ATOMS and are now deprecated.

```
fdat feout nepoints xanes modules message
noanomalous self i0 mcmaster dafs qvec
reflections refile egrid index corrections
```

The functionality of several of these keywords has been subsumed by the atp files (see Chapter 4 and Appendix B). The keywords having to do with DAFS are either handled by the programs written to do the DAFS simulations. A few of these options were simply discarded.

The old Fortran version of ATOMS allowed the user to specify a basis list rather than an atoms list. That is, the sodium chloride structure, for example, could be expressed as FCC with a two atom basis. I found basis lists rarely to be useful and dropped support for them in the new version of ATOMS.

6.3 Command Line Switches

There are several command line switches which can be used to control the behavior of ATOMS from the command line. Command line switches can be clustered, for example *atoms.pl -psF* is the same as *atoms.pl -p -s -F*.

- f Write an output file for FEFF6. -6 and -7 are synonyms for -f. It will be called 'feff.inp'.
- 8 Write an output file for FEFF8. It will be called 'feff.inp'.
- p Write a P1 output file. It will be called 'p1.inp'.
- u Write a unit cell output file. It will be called 'unit.dat'.
- g Write a 'geom.dat' file. It will be called 'geom.dat'.

- s Write a file listing the symmetry operations of the space group. It will be called 'symmetry.dat'.
- a Write a coordinate file in the alchemy format It will be called 'alchemy.dat'.
- x Write a coordinate file in the xyz format. It will be called 'xyz.dat'.
- O Write all output to STDOUT. You may want to use the -q switch to suppress normal run-time messages.
- t ...
- o ... These are used together to specify a file type and file name form the command line.

```
atoms.pl -t feff -o foo.inp
```

will write a FEFF6 output file and call it 'foo.inp'. This can be used also with user-defined atp files.

```
atoms.pl -t my_atp -o foobar.dat
```

will write an output file using the user-supplied 'my_atp.atp' file and call it 'foobar.dat'.

You may use the -o switch alone to redirect the output to a filename of your choice, however you should be aware that its use is ambiguous in some situations. If you are generating only one output file, then that file will be called by the value of the -o switch. If you are generating multiple output files from a single run of ATOMS, then the first file generated will be called the name given by the -o switch. The ambiguous part is that the order in which the output files are generated is unpredictable. This is due in part to the variety of ways of specifying output and in part to the data structure used in the code to store this information. If you are making many output files in a single run, it is probably a poor idea to use the -o switch.

- F Suppress the writing of an input file for FEFF.
- T Write a diagnostic file for testing ATOMS. This is also useful when reporting a bug.
- r #
Supply a value for rmax from the command line.
- q Suppress run-time messages (i.e. run in quiet mode).
- v Write version information to standard output and exit immediately.
- h Write a brief help message to standard output.

Chapter 7

Using the Tk Version

7.1 Overview of the Tk Version

$\mathcal{T}\mathcal{K}$ ATOMS is the graphical interface to the ATOMS package. It is written in perl/Tk application and includes interfaces to programs for calculating atomic clusters of the sort need by FEFF as well as several other calculations of interest to the x-ray absorption spectroscopist.

The $\mathcal{T}\mathcal{K}$ ATOMS window, shown in Figure 7.1, is split into two halves. At the top is a notebook widget, with several notecards for controlling the different programs. The bottom panel is used to specify crystallographic data, the user-supplied titles, and the absorption edge.

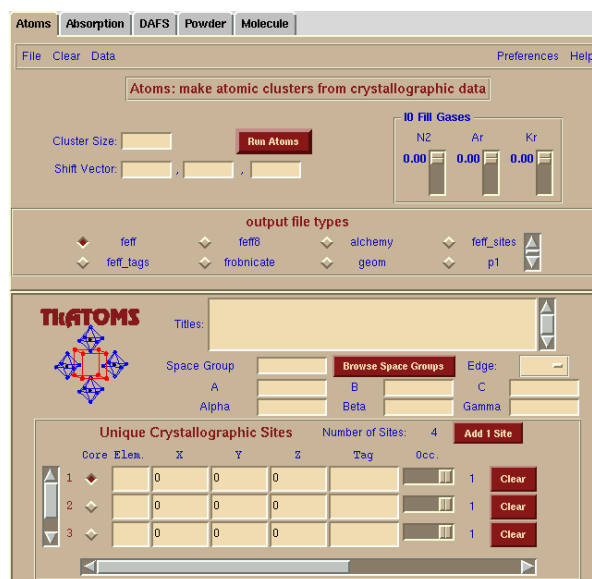


Figure 7.1: The $\mathcal{T}\mathcal{K}$ ATOMS window as it appears when first launched.

Many areas of the $\mathcal{T}\mathcal{K}$ ATOMS windows have help balloons attached. By letting the mouse

linger for a half second over any of these areas, a balloon pops up with a few words explaining the purpose of the widget near the mouse. In the case of text entry fields, only the label has a help balloon attached. This is because it's really annoying when a help balloon pops up over a text field that you want to type into. See Chapter 5 for a way of disabling the appearance of the help balloons.

Most actions in $\mathcal{T}\mathcal{K}\text{ATOMS}$ are initiated using the *mouse-1* button, although *mouse-3* is also used for some things. On a three button mouse, *mouse-1* is usually bound to the left button, *mouse-2* to the middle, and *mouse-3* to the right, although this layout is configurable. Because some people prefer to use a left-handed mouse layout, I will always refer to the mouse buttons by their numbers. On a two button mouse, as is common on Microsoft Windows systems, the left and right buttons are usually *mouse-1* and *mouse-3* respectively. *mouse-2* may be simulated using a keystroke/mouse-press combination. Sometimes pressing the chord between the mouse buttons simulates a *mouse-2* event on mice so equipped. On a one button mouse,¹ *mouse-2* and *mouse-3* are keystroke/mouse-press combinations usually involving the Apple and control keys. Some mice have more than three buttons (as does my own beloved trackball). $\mathcal{T}\mathcal{K}\text{ATOMS}$ does not use any higher-numbered buttons.

7.2 The Crystallography Panel

The bottom section of the $\mathcal{T}\mathcal{K}\text{ATOMS}$, shown in Figure 7.2, is used to enter crystallography data. These data are common to all the programs run using $\mathcal{T}\mathcal{K}\text{ATOMS}$. Consequently, this panel is visible regardless of which notecard is visible above it.

There are three sections in this panel. At the top is a text entry field for entering title lines describing your crystal. These title lines may be written to output files. Below the title box, is a grid of widgets for basic data about the unit cell. There are boxes for entering the space group symbol, the lattice constants and the lattice angles. There is also a drop down menu for choosing the absorption edge. As described in Section 3.1.1, you only need enter those lattice constants and angles which are necessary to describe your space group. See Section 7.5 for a description of what happens when you push the button labeled "Browse Space Group".

At the bottom is a scrolled table for entering the positions of the unique crystallographic sites. Initially this table has 4 lines, thus can hold data about 4 unique sites. More lines can be added to the end of the table by clicking on the button labeled "Add 1 site". The element symbol, x, y, and z coordinates, and site tag are as described in Section 3.1.3. The central atom of the cluster is chosen by clicking on the radio button next to the site of the central atom in the left-most column of the table. At the right side of the table are buttons which can be used to clear the data in that site. If you continue scrolling to the right, you will find additional columns for entering other details about a site. Currently, most of these additional columns are unused by ATOMS , but I expect that they will be used in future releases. There is no limit to the number of sites that you include in this list.

It is convenient to enter data into this table without having to shift your hands between

¹Insert derogatory remark about Macintoshes here!

TkATOMS

Titles: Testing TkAtoms 3.0alpha21
pseudo-cubic BaTiO3

Space Group: pm3m **Browse Space Groups** Edge: K

A: 3.95000 B: 3.95000 C: 3.95000
Alpha: 90.00000 Beta: 90.00000 Gamma: 90.00000

Unique Crystallographic Sites Number of Sites: 4 **Add 1 Site**

	Core Elem.	X	Y	Z	Tag	Occ.	
1	Ba	0	0	0		1	Clear
2	Ti	1/2	1/2	1/2		1	Clear
3	O	1/2	1/2	0		1	Clear

Figure 7.2: The Crystallography Panel

the keyboard and mouse. To this end, the arrow keys in combination with the shift key can be used to navigate the fields in the site list. For example *shift-right-arrow* moves the focus one entry field to the right. The other arrow keys behave similarly. All four directions wrap around the table, thus hitting *shift-right-arrow* several times will eventually return you to the same entry field. See the description for the variable `$unused_modifier` in Chapter 5 for details on how to customize the arrow key sequences.

At the right side of the table are sliders for setting the occupancy of the site. This is how dopants can be introduced into $\mathcal{T}\mathcal{K}\mathcal{A}\mathcal{T}\mathcal{O}\mathcal{M}\mathcal{S}$. Please read Section 3.4 and [my FEFF for EXAFS web page](#) for a discussion of the use of dopants in ATOMS.

7.3 The Program Panel

The upper panel of the $\mathcal{T}\mathcal{K}\mathcal{A}\mathcal{T}\mathcal{O}\mathcal{M}\mathcal{S}$ window uses a notecard metaphor to allow you to select among several different programs which use the crystallography data displayed in the lower panel. These programs are

Atoms

Generate ordered lists of atomic coordinates

Absorption

Calculate quantities of interest using tables of x-ray absorption coefficients.

DAFS Simulate the energy dependence of the anomalous scattering from a crystal using tables of normal and anomalous structure factors.

Powder

Simulate a powder diffraction scan at a specified energy.

Molecule

Convert structure data for a molecule into a FEFF input file.

7.3.1 The Atoms Notecard

This notecard contains widgets for controlling different aspects of the ATOMS program.

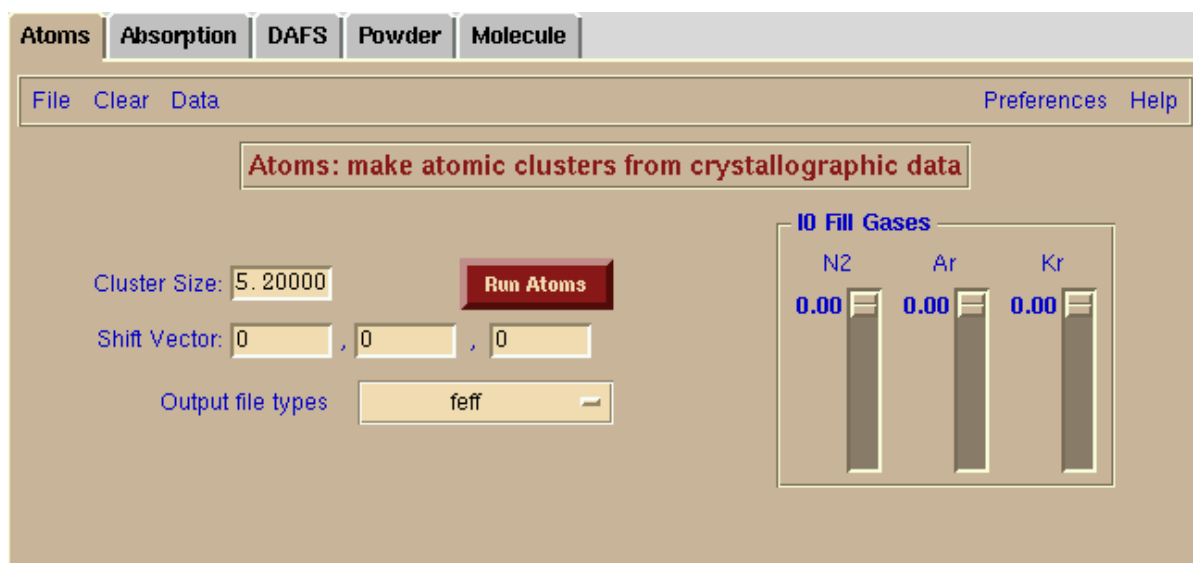


Figure 7.3: The Atoms Notecard

The Run Atoms button

This button is rather self-explanatory. Once you have filled in all the widgets with the appropriate data, press this button to calculate a list of atomic coordinates. Before actually calculating, ATOMS will verify that all the data you have entered is sensible, determine whether your chosen space group might need a shift vector, and verify that your lattice constants are appropriate to the chosen space group. If everything checks out, the list will be calculated and displayed in the output display window, described in Section 7.4.

The Cluster size entry box

If you select a type of output that generates a radially sorted cluster of atoms, then the size of the radial cluster is determined by the contents of this box. If you leave it blank, the cluster size defaults to the smaller of 7 Å and 1.1 times the length of the largest lattice constant. If you choose a really huge number, then your output file will contain a huge number of atoms. Even worse, the time and memory required to

calculate a huge list scale as the third power of this number. Usually 10 Å or less is quite enough.

The Shift Vector entry boxes

These boxes are used to enter the coordinates of a vector which is needed in certain situations. Read Section 3.1.4 for a complete discussion of the shift vector. If you use a space group which might require a shift vector, $\mathcal{T}\mathcal{K}\text{ATOMS}$ will pop up a dialog box with a message similar to that given in Figure 3.1.

Note that the contents of the shift vector entry boxes are evaluated. This means that you are allowed to type things like 1/8 rather than 0.125.

If the space group specified in the crystallography panel is one which commonly takes a shift vector, that vector can be inserted into these boxes using a function in the **Data** menu.

The I0 fill gas sliders

These are used to indicate the contents of the I0 chamber and are used by ATOMS to calculate the self-absorption and I0 corrections to the EXAFS data. These calculations are described in Section ???. The units on these sliders are percentage of the total pressure in the chamber.

The Output file types menu

$\mathcal{T}\mathcal{K}\text{ATOMS}$ allows you to select from a list of possible output files. The type of output file is selected by choosing the output file type from a drop-down menu. The list of output file types is generated at startup time by scanning the ATOMS installation directory and the users configuration directory for files which end in '.atp'. These atp, or ATOMS *template*, files are used by ATOMS to format its output files. See Chapter 4 and Appendix B.

The menubar

At the top of the notecard, you will see a menubar. Clicking *Mouse-1* on any item in the menubar will drop a menu. The **File** menu is used to load and save input files and to quit $\mathcal{T}\mathcal{K}\text{ATOMS}$. Loading and saving uses the standard `Tk::FileDialog` widget. Its use is straightforward.

The **Clear** menu is used to clear out part or all of the data in the ATOMS notecard and in the lattice panel. The **Data** menu is used to perform various data operations. The **Help** panel provides access to the on-line help system.

7.3.2 The Absorption Notecard

This notecard is used to make calculations using tables of x-ray absorption data. These are the same calculations may be written to the top of ATOMS output files except that, using this notecard, you can consider dopants and fractional occupancy. Occupancy data is not used when making atom lists using the ATOMS notecard.

Six calculations are made on this notecard. These are

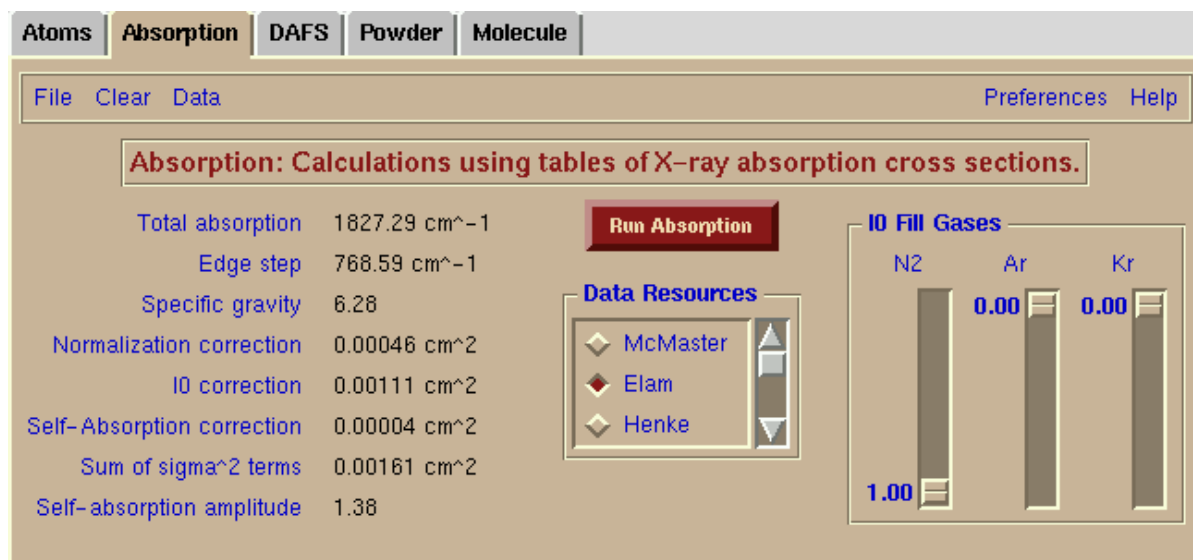


Figure 7.4: The Absorption Notecard

Total Absorption

The inverse of the total absorption is the absorption length of the sample, i.e. the distance over which incident x-rays at the chosen absorption edge attenuate e-fold.

Edge step

The change in absorption over the absorption edge energy. The inverse of this number is the sample thickness which yields a unit edge step in a transmission experiment.

Density

This is the mass of the contents of the unit cell divided by the volume of the unit cell.

Normalization correction

Normalizing EXAFS data by the value of the edge step introduces a small error into the measured $\chi(k)$. This error is proportional to k^2 , thus it behaves like an additional Debye-Waller factor.

I0 correction

In a fluorescence experiment, the energy response of the I0 detector introduces a small error into the measured $\chi(k)$. This error is proportional to k^2 , thus it too behaves like an additional Debye-Waller factor.

Self-absorption corrections

In a fluorescence experiment, the measured signal is attenuated by the energy dependence of the absorption of the sample. There are two terms, a k^2 term and an amplitude term. These are approximated with the assumptions that the sample is infinitely thick and that the measurement geometry is such that the incident and exit angles of the photons are equal.

See Chapter 10 for a discussion of how these calculations are made.

There are sliders controlling the contents of the I0 chamber on this notecard. These sliders are completely independent of the similar sliders on the ATOMS notecard. The I0 and self-absorption corrections are only calculated if one or more of the sliders is non-zero.

All other data regarding the calculation, including the identity of the central atom and the absorption edge, is taken from crystallography panel. Use the occupancy sliders to explore the effects of vacancy and doping on the numbers calculated by this notecard.

The absorption data resource (i.e. McMaster, Henke, Elam, and so on) can be chosen using the radiobuttons on the left side of the notecard. These can be changed at any time, allowing you to compare the different data resources.

The menubar on this notecard is very similar to that on the ATOMS notecard. The one interesting addition is in the Data menu. Using the **Absorption Units** entry, you can switch between units of inverse centimeters and microns for the total absorption and edge step calculations. These representations are simply reciprocals of one another and scaled appropriately.

7.3.3 The DAFS Notecard

This notecard is used to approximate the energy dependence of the amplitude of a diffraction peak as the energy is scanned through an absorption edge. The full complex scattering factor is used. Values for f_0 are taken from the Crommer-Mann tables and the anomalous corrections are taken from the Henke tables, the Chantler tables, the Brennan/Cowen tables or any other data resources provided by ‘Absorption.pm’ (See Appendix D.4).

All of the data resources in ‘Absorption.pm’ neglect solid state effects, thus there is no fine structure in the DAFS simulation. However, knowing the size and direction of the cusp for a particular reflection can be useful information when planning or interpreting an experiment. In the future I plan add the ability to use an external file in place of the absorption data resource for any atom in the unit cell as a way of introducing solid state effects to the simulation. This might be a FEFF calculation or the output of Matt Newville’s DIFFKK program.

The calculation is made over an energy range which usually includes the edge energy of one of the elements in the material. The complex scattering of all elements in the material is included in the calculation, even those with edge energies far from the chosen edge energy. Some atoms, particularly heavy atoms, have significant complex scattering components even far from their resonant energies.

There are five widgets on this notecard:

The Run Button

Press this button to start the calculation.

The Energy Grid box

The three text entry fields in this box are used to determine the energy grid of the calculation. **Emin** is the number subtracted from the chosen edge energy to determine the lower bound of the calculation. If you choose to calculate at the copper *K* edge (8979 eV) and set **emin** to 300, the calculation will start at 8679. **Emax** is a number

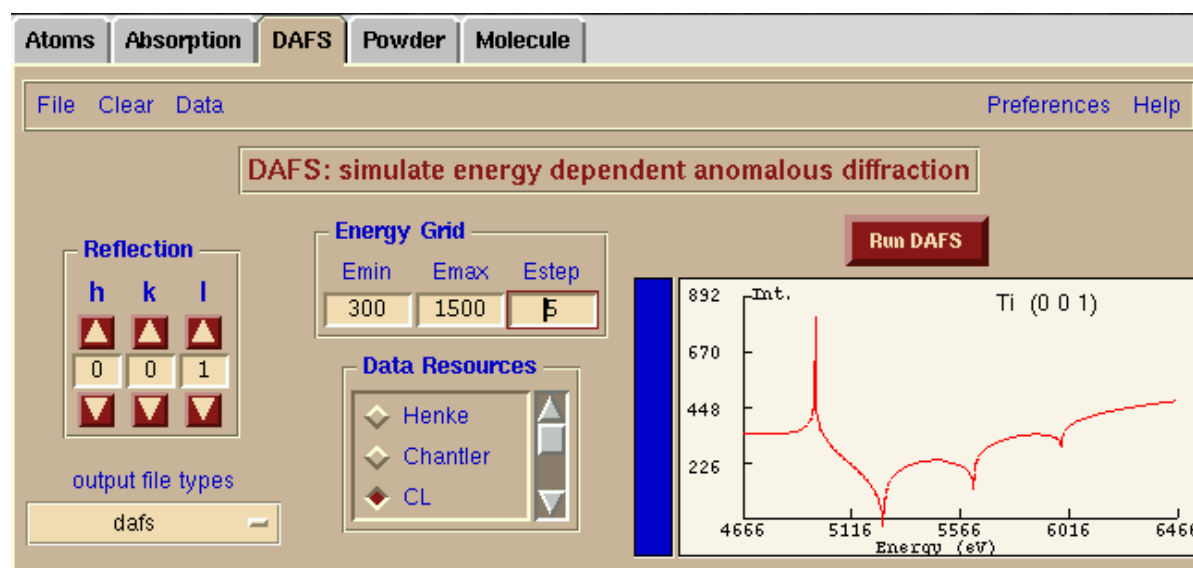


Figure 7.5: The DAFS Notecard

added to the edge energy to determine the upper bound of the calculation. In the Cu K edge example, with e_{\max} set to 500, the calculation will end at 9479. E_{step} is used to determine the grid spacing.

The Reflection box

These three fields are used to set the Miller indices of the reflection at which to calculate the DAFS spectrum. you may simply type in the fields or use the up and down arrows to increment and decrement the indices. Note that one or more of the indices must be non-0 or else $\mathcal{T}_{\text{KATOMS}}$ will signal an error.

The Data Resource box

Only those resources which provide anomalous scattering factors (i.e. not McMaster and Elam) are listed in this box. Choose a resource by clicking on it.

The Plotting canvas

The plotting canvas consists of two parts, the progress bar and the plot. As numerical calculations can be rather slow in perl, the progress bar gives you a visual cue as to the progree of the calculation. As the calculation proceeds, this area fills up from the bottom. Usually calculations are fast, but if you choose a very broad energy range or a very tight grid, it may take 10 or seconds.

When the plot finishes, a thumbnail of the calculation is displayed in the plotting space to the right of the progress bar. Do not confuse this with a proper, interactive plotting tool. It is not. The plot shown here is just a crude representation of the calculation. To examine the calculation fully you will need to save the data and plot it using your favorite plotting utility. Some day you will have the option of sending the data off to an external plotting package, such as XMgr or Matt's IFEFFIT.

The **Data** menu in the menu bar has several interesting options for handling the output of a DAFS calculation. You may save the calculation as using the format of the `'dafs.atp'` file (the one that comes with ATOMS formats the data as two columns of energy and amplitude squared). You may also output the thumbnail in the plotting canvas to a postscript file.

There is also a (highly experimental as of ATOMS 3.0alpha24) mechanism for sending the calculation to an external plotting program. This requires the use of the `'AtomsConfig.pm'` file and some voodoo.

7.3.4 The Powder Notecard

The Powder program has not yet been written as of ATOMS 3.0alpha24.

7.3.5 The Molecule Notecard

Not everything in the world is a crystal. This notecard is ATOMS's stab at accommodating the non-crystalline world. For many kinds of materials, e.g./ proteins, organo-metallics, and others, structural data is available in the form of Cartesian coordinates for the atoms in the material. The purpose of this notecard is to read in that kind of data (or to explicitly type it into the appropriate entry boxes) and convert that data into a useful FEFF input file. Please note that this is extremely fragile and incomplete as of ATOMS 3.0alpha24. When it works, it's nifty. When it doesn't (which is usual), it behaves very mysteriously.

There are four components on this notecard. The button labeled "Run Molecule" does just that, outputting the FEFF input file to the output display window (see Section 7.4). The cluster size box allows you to specify a radial cut off for the atoms in the list. The output file types menu lets you select from available output types. $\mathcal{T}_{\mathcal{K}}\text{ATOMS}$ ships with formats for turning thus kind of data into files for FEFF6 and FEFF8. See Chapter 4 and Appendix B for more details.

The big list is similar to the list in the crystallography panel except that it expects numbers to be entered as Cartesian coordinates in Ångströms. The radio buttons are used to select the central atom and the **core** and **tag** columns serve the same purposes as the similarly labeled columns in the crystallography panel.

The titles and absorption edge for the output file are taken from the crystallography panel. Other data in the crystallography panel is ignored by this notecard.

The column labeled **skip** in the atoms list contains check buttons. Any site which is checked in this column will be ignored when writing the output file. In the **Skip** menu in the menubar is a function which pops open a small window which you can use for algorithmically selecting atoms to skip. These skip rules are regular expressions matched against the element symbols or tags or else mathematical expressions matched against the coordinates.

Someday I intend for ATOMS to ship with object methods for reading in various common file formats of non-crystalline structural data, for example ShellX and Protein Data Bank files. Currently, it only knows how to read in alchemy files and a sort of generic file format that I made up. When you try to load data you be asked about what kind of file format the data is

in. Heuristics are used to try to guess this. Don't expect fireworks.

7.4 The Output Display Window

When you select an output file and run ATOMS by pressing the **Run Atoms** button, a new window appears displaying the output file. This is a simple text editing widget which allows you to alter the output before saving it to disk. An example is shown in Figure 7.6.

You may save the file to disk by pressing the **Save** button at the bottom of the screen. This pops open the same file dialog used to load and save input files. In this case you are offered a default file name which is taken from the atp file used to format the output data.

There are two other buttons at the bottom of the display frame. One dismisses the display frame. The other is labeled **Run Feff** and is active only when the output file is intended to run feff. However, pressing the **Run Feff** button doesn't do anything at this time.



```
* This feff input file was generated by TkAtoms 3.0alpha22
* Atoms written by and copyright (c) Bruce Ravel, 1998, 1999

* --- *
* total_mu = 1827.29 cm^-1, delta_mu = 768.59 cm^-1
* specific gravity = 6.285
* --- *
* Normalization correction: 0.00046 ang^2
* I0 correction: 0.00111 ang^2
* self absorption correction: 0.00004 ang^2
* amplitude factor: 1.379
* --- *
* net correction: 0.00161 ang^2
* --- *

TITLE Testing TkAtoms on pseudo-cubic BaTiO3

HOLE 1 1.0 * Ti K edge (4966.0 eV), second number is S0^2
      mphase, mpath, mfeff, mchi
CONTROL 1 1 1 1
PRINT 1 0 0 0

RMAX 7.0

*CRITERIA curved plane
*DEBYE temp debye-temp
*NLEG 8

POTENTIALS
* ipot z element
  0 22 Ti
  1 56 Ba
  2 22 Ti
  3 8 O

ATOMS
* x y z * this list contains 119 atoms
  0.00000 0.00000 0.00000 0 Ti 0.00000
  ipot tag distance
```

Figure 7.6: The Atoms output window displaying an input file for FEFF6.

7.5 The Space Group Browser

The space group browser is fairly easy to use as the status bar at the bottom of the widget always informs the user about what actions are available as the mouse passes over active regions of the widget. Three examples of these messages are shown in Figure 7.7.

The opening panel displays the seven crystal classes. This is shown in the left panel of Figure 7.7. The names of the crystal classes are active text. Clicking any mouse button on a

crystal class will display a list of all space groups in that class. If the space group text entry field is displaying a valid group symbol, then that symbol will be displayed as active text. Clicking *Mouse-1* on that symbol will jump to the panel describing that symbol.

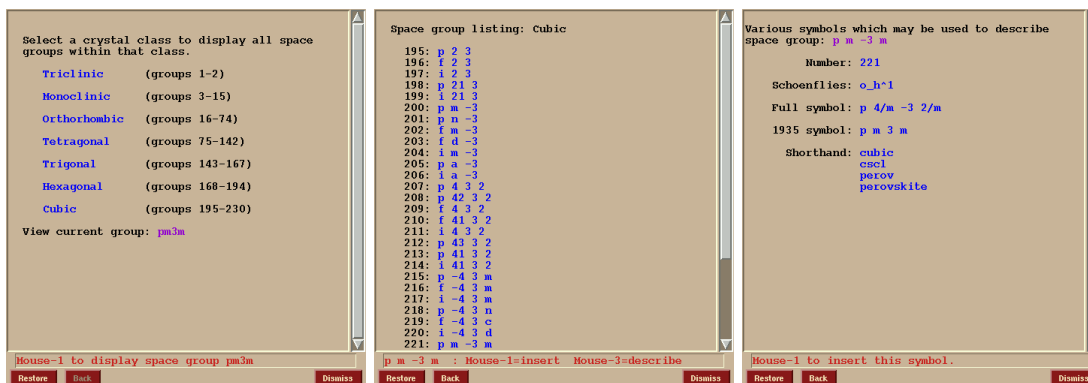


Figure 7.7: The three screens of the space group browser.

The second panel displays lists of space groups divided by crystal class. This is shown in the middle panel of Figure 7.7. The index of the group (as indexed in the International Tables of Crystallography) and the canonical (Hermann-Mauguin) space group symbol are shown. The space group symbols are active text. Clicking *mouse-1* will insert that symbol into the space group text entry field. Clicking *mouse-3* will display information about that group.

The last panel is the space group description and is shown in the right panel of Figure 7.7. It shows all symbols recognized by ATOMS as describing that group. Clicking *mouse-1* on any active text will insert that symbol into the space group text entry field.

At the bottom of the widget are three buttons. The **back** button causes the previous panel to be displayed. The **restore** button is only active if the space group text entry field contained a symbol when the browser was first invoked. If so, pressing it will restore that symbol to the space group box in the crystallography panel. The **dismiss** button hides the SGB widget.

7.6 The Configuration Window

The menubar in each notecard has a **Preferences** menu on the right hand side. This menu is used to launch the interactive configuration utility which can be used to modify variables which control the behavior of ATOMS and $\mathcal{T}_{\mathcal{K}ATOMS}$, to change the colors used by $\mathcal{T}_{\mathcal{K}ATOMS}$, and to change the fonts used by $\mathcal{T}_{\mathcal{K}ATOMS}$. The variables, colors, and fonts each have their own notecard in the configuration window. At the bottom of the window are two buttons which are there for all three notecard. The button labeled **Save Values** will cause $\mathcal{T}_{\mathcal{K}ATOMS}$ to write the current values for the variables, colors, and fonts to an 'atomsrc' file of the sort described in Chapter 5. The **Dismiss** button hides the configuration window.

There are help balloons attached to each of the variables, colors, and fonts which may be set

using this tool. That should help you make appropriate choices when configuring $\mathcal{T}\mathcal{K}\text{ATOMS}$.

7.6.1 Configuring Variables

All of the variables described in Chapter 5 can be set using the the variables configuration notecard. There are several different kinds of widgets on this notecard. The button labeled **Set Variables** is used to actually fix the values of the variables to the values indicated by the widgets to the left. This only sets them for the current $\mathcal{T}\mathcal{K}\text{ATOMS}$ session but does not save them for future sessions. The **Restore** button restores all variables to the values from the start of the session.



Figure 7.8: The variables configuration window.

The variables are set using different types of widgets depending on the nature of the variable. The first several variables are boolean and are turned on by setting the checkbox to its on state and turned off by setting the checkbox to its off state. The next several variables can take a value from a list of possible values, so pop-down menus are provided.

Some variables take files names as values. In that case, a text field is provided for typing in the file name. Also a button labeled **Browse** will open a file dialog for specifying the file name.

Variables which take arbitrary text as their values have text entry fields.

7.6.2 Configuring Colors

The color selector is a bit quirky. Unfortunately the standard color dialog that comes with perl/Tk is not well suited for use with $\mathcal{T}\mathcal{K}\text{ATOMS}$. Thus I adopted elements from it into my color configuration utility.

The basic idea is that you select a color using the sliders and/or list box on the right side of the notecard. When you have found a color that you like for a particular graphical element,

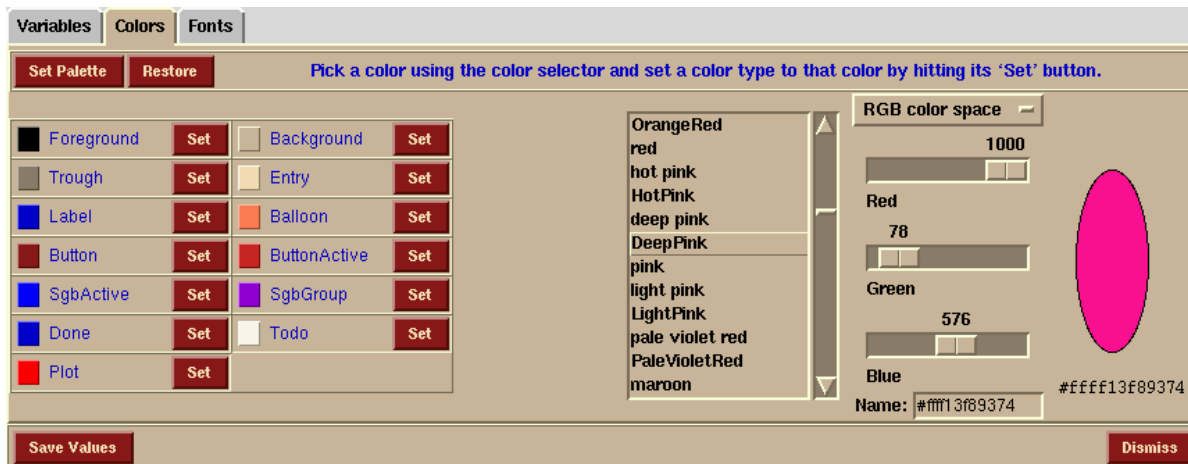


Figure 7.9: The color configuration window.

click on the **Set** button for that element. You will notice that the little color box to the left of the name of the graphical element will change to your selected color. When you have found all the colors that please you, click on the **Set Palette** button. This might take a few seconds, but when done all of your color choices will have propagated through all aspects of $\mathcal{T}_{\mathcal{K}ATOMS}$. The **Restore** button will restore $\mathcal{T}_{\mathcal{K}ATOMS}$ to the color configuration it had at the beginning of the session.

If you find a color set that you like, you should press the **Save Values** button so that next time you start $\mathcal{T}_{\mathcal{K}ATOMS}$, you will see that color set.

The color selector has several components which you can use to fine-tune all your color selections. The drop-down menu lets you select a color definition scheme. By default red-green-blue (RGB) color values are used, but two other quantizations are available.

The sliders are used to set the values of the red, blue, and green channels (if RGB is used). The best way to understand this is to slide them back and forth and watch the color displayed in the oval change.

If your system is running X Windows and so has an `rgb.txt` file that the color selector can find, it will also display the list box shown in Figure 7.9. You can scroll through this list and select named colors by double clicking *Mouse-1* on them. The oval will change to that color when after double clicking.

Once you have selected a color, click on the appropriate **Set** button as described above.

A nice alternative to the default color scheme is given in Table 7.1. To try this color scheme out, enter those values using the color configuration window or type the hex values into your `'atomsrsrc'` file.

Table 7.1: An attractive alternative color scheme.

widget	color name	RGB triplet	hex value
Foreground	black	(0, 0, 0)	#000000000000
Background	MistyRose3	(804, 718, 710)	#cdd2b7ceb5c2
Trough	MistyRose4	(545, 490, 482)	#8b857d707b64
Entry	MistyRose2	(933, 835, 824)	#eed8d5c2d2f1
Label	DarkGreen	(0, 392, 0)	#0000645a0000
Balloon	turquoise	(251, 878, 816)	#4041e0c4d0e5
Button	DarkSlateGray	(184, 310, 310)	#2f1a4f5c4f5c
ButtonActive	SlateGray	(439, 502, 565)	#7062808390a3

7.6.3 Configuring Fonts

Font configuration is not supported in ATOMS 3.0alpha24. However you may modify the font variables by editing the runtime configuration file as described in Chapter 5. See the pod documentation in ‘Tk::Font’ for more details about available font names.

Chapter 8

Serving and Using the Web Version

*Web*ATOMS is the version of ATOMS written to be a CGI application, i.e. it is served over the web and the interaction between the user and the program is via a web browser such as Netscape or Internet Explorer. In this Chapter, I explain how to install *Web*ATOMS on your own web server and give a brief description of its use by a client.

8.1 Installing and Configuring *Web*ATOMS

This section explains how to set up *Web*ATOMS to run on your web server. That means that you administer a computer which is used to serve content to the World Wide Web. If you are only interested in using *Web*ATOMS by connecting your web browser to a site that offers *Web*ATOMS, then you can skip ahead to Section 8.2.

I am assuming that you have a web server correctly configured to run CGI scripts. I am also assuming that you have perl installed on that computer and that perl is available for use by your CGI scripts. If you are using *APACHE*, you should look into installing *mod_perl*. There is a lot of overhead at start-up as *Web*ATOMS loads various modules and databases into memory. If you anticipate any significant level of traffic on your site, *mod_perl* will help *apache* and *Web*ATOMS run more efficiently. Similarly, if you are running Microsoft's *IIS*, you should look into installing PerlScript from *ActiveState*. Other web servers may offer similar abilities for embedding a perl interpreter in the running web server. I recommend you look into it.

Here are the steps to installing *Web*ATOMS on your server. Please note that I have only installed *Web*ATOMS on a Linux machine running *APACHE*. Some parts of the instructions might be different for other systems and/or servers.

1. Install the ATOMS package on the server. *Web*ATOMS is its own program and does not require either '*atoms.pl*' or '*tkatoms.pl*' to run correctly. It does, however, make use of all the same modules and databases as the other versions of ATOMS. If ATOMS and *T_K*ATOMS are working correctly on the server, then *Web*ATOMS is very likely to work also. Don't forget also to install the Atoms Bundle as described in Chapter 2.
2. You need to edit the values of four variables in the file '*atoms.cgi*', which is found in

the 'cgi' subdirectory of the ATOMS distribution. These four variables have to be set to correctly reflect the configuration of your server. They are

`$extra_INC`

If Atoms is installed somewhere outside of the default search path for perl modules, then set this variable to that location. The variable is unshifted to INC when 'atoms.cgi' starts.

`$atoms_help`

This takes the URL to the place where you will install the file 'WebAtoms_help.html'.

`$ADB_search`

The URL of the ATOMS Database search form, assuming you have a copy of the ATOMS Database on your server. If not, set this variable to 0.

`$ADB_directory`

This takes the actual location (*not* the location relative to the server's root) on the server disk where input files may be found. The purpose of this is so that 'atoms.cgi' can be on the back end of a search CGI script.

3. Put 'atoms.cgi' in a place where your web server knows to look for CGI programs. Make sure it is executable.
4. Put 'WebAtoms_help.html' in the place indicated by the `$atoms_help` variable. You can use the copy on my web page
http://feff.phys.washington.edu/~ravel/atoms/doc/WebAtoms_help.html
but I would rather that you install a copy on your own server.
5. Run 'atoms.cgi' once by hand from the command line like this:

```
atoms.cgi -install
```

This sets up the set of hard links to the script which are used to serve up the correct file names when the client clicks the "save-as" button. Note that these symbolic links have names that might conflict with files that you want to place in the same disk space as 'atoms.cgi', for example 'atoms.inp' and 'feff.inp'.

If you encounter problems, there are two things that you could include in a bug report that would greatly improve my chances of solving the problem. (1) Any relevant lines from the server log and/or error log. (2) Save the page where you entered input data as a flat text file and mail me the text file. The second thing will allow me to attempt to reproduce the problem on my own computer.

WebATOMS is designed to interact nicely with other web applications. If you have ATOMS input files on your web server that you would like WebATOMS to read, you can call WebATOMS from another CGI script or from a static page using a URL like this:

```
http://what.ever/path/to/atoms.cgi?file=foo.inp
```

In that case, the form will be filled in with data from the specific input file.

8.2 Using WebATOMS

This section documents the use of *Web*ATOMS by a client connected to a site serving *Web*ATOMS. All you need is a URL. One such location is

<http://millenia.cars.aps.anl.gov/cgi-bin/atoms/atoms.cgi>.

The use of *Web*ATOMS is very similar to the use of ATOMS and \mathcal{T}_K ATOMS. If you have read and understood Chapters 6 and 7 then *Web*ATOMS will be obvious. If not, I recommend that you read those chapters because here I will only outline the aspects of *Web*ATOMS which differ from the other versions.

Figure 8.1 shows what *Web*ATOMS looks like with Netscape. The first thing to notice is that all of the data needed by ATOMS has either a text field, a button, or a menubutton associated with it. The next thing to notice is that the label of every widget is a link. Clicking on that link will take you to the appropriate part of a web page explaining the purpose of each kind of input data. If you are confused, click on a label!

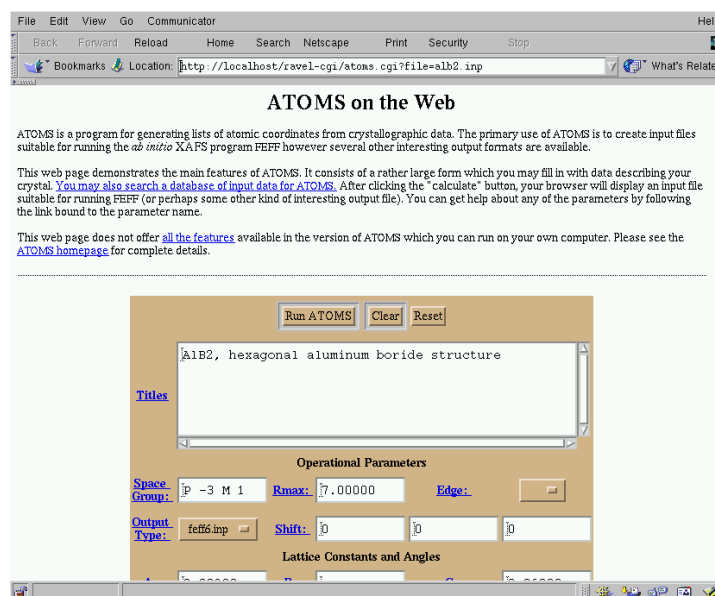


Figure 8.1: The top half of the *Web*ATOMS page as seen with Netscape, with data for AlB_2 filled in as an example.

Figure 8.2 shows the rest of the page from Figure 8.1. In particular, you see that there is a list for specifying the element symbols, tags, and coordinates of each of the atoms. If you need more rows for specifying more atoms positions, enter the number you require in the box labeled with “Redisplay with this many sites” and then click the “Do it!” button. The page will be redisplayed without losing data, but with more rows for site.

The menubutton labeled “Output type” lets you select from various different output formats. Most of the types discussed in Chapter 4 are in this menu. The menubutton labeled

“Edge” allows you to explicitly select an absorption edge. If left blank the algorithm for selecting edges described in Section 6.2 will be used.

Once all the data has been entered correctly, click on the “Run Atoms” button. If all of the data is sensible and *Web*ATOMS is able to run to completion, your output file will be presented as a plain text page suitable for downloading. If there is a problem in the data, the entry form will be redisplayed with some hints about what the problem was.

The screenshot shows a Netscape browser window with the URL `http://localhost/ravel/cgi/atoms.cgi?file=alb2.inp`. The form contains the following sections:

Space Group: `P -3 M 1` **Rmax:** `7.00000` **Edge:**

Output Type: `teff6.inp` **Shift:** `0` `0` `0`

Lattice Constants and Angles

A: `3.00900` **B:** `...` **C:** `3.26200`

Alpha: `...` **Beta:** `...` **Gamma:** `...`

Table of Crystallographic Sites

Cent.	Element	X	Y	Z	Tag
↗1	A1	0.00000	0.00000	0.00000	...
↘2	B	0.33333	0.66667	0.50000	...

Redisplay with this many sites: `2`

WebATOMS version 1.3 (1999/11/02)
 ATOMS is copyright © 1998,1999 Bruce Ravel
[ATOMS homepage](#)
[Bruce's homepage](#)

Figure 8.2: The rest of the page shown in Figure 8.1.

Finally there are buttons labeled “Clear” and “Reset”. Pressing the “Clear” button will cause a blank page to be displayed. Pressing the “reset” button will return the page to the state of its most recent display or redisplay.

Chapter 9

APT.PL

NAME

APT — The ATOMS Periodic Table

SYNOPSIS

Simple graphical interface to the Xray::Absorption package

```
apt [-v] [-h]
```

DESCRIPTION

This is a simple graphical interface to the x-ray absorption data for the elements contained in the Xray::Absorption package. When run, a window appears displaying a colorful periodic table of the elements. Clicking on an element causes data about that element to be displayed in the bottom half of the window.

All energies in this program are in eV. All wavelengths are in Angstroms. All distances are in microns or centimeters as described above. The buttons labeled "Clear", "Help", and "Exit" do exactly what their labels suggest, although the "Clear" button only clears the currently displayed notecard. These have keyboard shortcuts of control-c, control-h, and control-q respectively. Control-t is the keyboard shortcut for toggling between energy and wavelength units.

If this program is invoked with a -v switch, version information is displayed to standard output and the program quits. If -h is given, this help document is displayed to standard output and the program quits.

THE ENERGIES AND ABSORPTION LENGTHS PROGRAM

This notecard is used to display edge and line energies of the elements and to make simple calculations of absorption length and attenuation. When you click on an element in the periodic table, these data about that element are displayed in the notecard.

If an energy in eV is specified, then the absorption length of the selected element at the specified energy will also be displayed. If the selected element is a gas, the absorption length is given in centimeters, otherwise it is given in microns.

If a length is also specified in the box marks "thickness", then an attenuation factor will be given for a pure sample of that length and at the specified energy. For example, if you enter 8000 eV as the energy, and "10" as the thickness then select nickel, the attenuation will be 0.64. This means that a 10 micron nickel foil absorbs 36% of the incident beam. Again, centimeters are assumed for gases. With the thickness set to 10, selecting nitrogen gives an attenuation of 0.91 at that energy. This means that a 10 centimeter ionization chamber filled with nitrogen will absorb about 9% of the beam.

Not all of the data is available for all the elements. When data is missing, the corresponding space will be left blank. For transuranic elements, only the name and atomic number are displayed.

All data used in this program comes from the Elam data resource (see the *Xray::Absorption* manpage and the *Xray::Absorption::Elam* manpage). Of all the available xray absorption data resources, the Elam resource has by far the most complete collection of edge and line energies. The edge and line energies are organized into notecards. Click on the tabs of the notecards to see the different pages. The Siegbahn (ok...the greek letters are spelled out in English...) and IUPAC symbols for the fluorescence lines are displayed, along with the line energy and the relative intensity of the line. The relative intensity is normalized such that the sum of intensities from lines originating in the same core state sum to 1.

The weight displayed is the isotope-averaged atomic weight. The density is for the most common pure form of the element. The density displayed for carbon is the density of graphite. (Diamond has a specific gravity of about 3.1.)

The third button from the left at the bottom of the screen is used to change the units displayed in the program between eV for energy and Angstroms for wavelength. Clicking this button will toggle all parts of the program between these two units. Please note that the entry box labeled "Energy" (or "Wavelength") also toggles between the two units. When you have the program set to use wavelength, you must enter wavelength values in that box. If you enter a value that seems too small for energy units or too large for wavelength units, the program will pop open a confirmation dialog.

THE ANOMALOUS SCATTERING PROGRAM

This notecard contains a simple interface to tables of anomalous scattering factors. Simply fill in an energy range and a value for the energy step and click on an element, and the f prime

and double prime functions will be displayed in the canvas on the right. Alternatively, you can select an absorption edge and click on an element, and the anomalous scattering factors will be displayed 100 volts above and below that edge. You can even simply click on an element. If the element is lighter than cerium, the scattering factors around the K edge will be shown, otherwise the scattering factors around the L3 edge are shown.

Below the widgets allowing you to specify the energy grid and the edge are two buttons. The one labeled "Save data" allows you to save the most recent calculation to a file. By default the filename is "fpfpp_XX.dat", where XX is the symbol of the element for which the anomalous scattering was calculated. You can change the name of the output file in the file dialog.

The "Clear range" button is a bit more complicated. Because it is often useful to calculate the anomalous scattering factors of one element near the absorption edge of another element, the energy range is not cleared when you press on a new element button. Instead, the energy range used for the prior calculation is used for the new calculation. If, for example, you want the calculation made around an edge of the new element, you should hit the "Clear range" button or the "Clear" button at the bottom of the page.

The button which converts between wavelengths and energies currently does nothing when the anomalous scattering notecard is displayed.

While you certainly may specify very broad energy ranges, for example to see K and L edges for an element, do remember that perl is an interpreted language and that such a request will be rather time consuming even on a speedy computer.

TO DO

- More kinds of useful data about the elements

ACKNOWLEDGMENTS

Thanks to Stephane Grenier and Matt Newville for their suggestions and willingness to beta test.

AUTHOR

Bruce Ravel, bruce@phys.washington.edu
<http://feff.phys.washington.edu/~rael>
copyright (c) 1999 Bruce Ravel

Chapter 10

Experimental Corrections

ATOMS can perform several useful calculations using tables of x-ray absorption cross sections, including estimations of absorption length and various experimental corrections. There are several tables x-ray absorption cross sections which the user can choose from:

1. The 1999 Elam tables (the default)
2. The 1969 McMaster Tables
3. The 1993 Henke Tables
4. The 1995 Chantler Tables
5. The Brennan-Cowen implementation of the Cromer-Liberman Tables¹.

See the references in Chapter 12 and the technical documentation in Appendix D.

10.1 The Density and the Absorption Lengths

In an EXAFS experiment in transmission mode, proper sample preparation is essential for collection of high quality data. In order to make appropriate choices for the composition and form of the sample it is necessary to know both the absorption length of the sample and its edge step absorption length. ATOMS approximates both of these quantities.

The absorption length is defined as the thickness x of the sample such that the intensity of the x-rays incident upon the sample at an energy 50 eV above the absorption edge is attenuated e-fold. That is, $e^{-x\mu} = 1$ where μ is the total absorption of the sample at that energy. The sample length for unit edge step, the reciprocal of which is called $\Delta\mu$, is the thickness x such that there is an e-fold change in absorption between 50 eV below and above the edge, i.e. for $e^{-x\cdot\Delta\mu} = 1$.

ATOMS computes the free atom cross-sections for each atom specified in the input data at 50 eV above the absorption edge and for the resonant atom at 50 eV below the edge. Using the crystallographic information, it knows the size of the unit cell and how many of each species are

¹This requires installing additional software. See [the ATOMS homepage](#) for details

in the unit cell and so can correctly calculate the total and unit edge step absorption lengths for the crystal. Using the unit cell size and tabulated masses of the elements, the density of the crystal is also computed. With the absorption lengths and density of the material, proper choices for sample preparation can be made.

Here is an example from my graduate thesis of a transmission sample for lead titanate PbTiO_3 prepared using these calculations from ATOMS. Using this structural data

```

title PbTiO3 25C
title Glazer and Mabud, Acta Cryst. B34, 1065-1070 (1978)
core=ti      Space P 4 m m
a=3.905      nitrogen=1      rmax=3.6      c=4.156
atom
! At.type  x      y      z      tag
  Pb      0.0      0.0      0.0
  Ti      0.5      0.5      0.539
  O       0.5      0.5      0.1138  axial
  O       0.0      0.5      0.6169  planar

```

yields the results shown in Table 10.1.

Table 10.1: Absorption lengths and experimental corrections for lead titanate calculated using the Elam tables.

calculation	value
total μ	4850.42 cm^{-1}
$\Delta\mu$	747.42 cm^{-1}
specific gravity	7.942
Normalization correction	0.00046 \AA^2
I0 correction	0.00111 \AA^2
self absorption correction	0.00003 \AA^2
self absorption amplitude factor	1.093
net correction	0.00160 \AA^2

In this example, the absorption length is $1/4850$ cm or about 2.1 microns and the edge step absorption length is $1/747$ cm or about 13.4 microns. For my experiment on the titanium K edge of lead titanate, I chose to make a sample which was 5 microns thick, thus with total absorption of about 2.5 and an edge step of about 0.37. Knowing the appropriate thickness for the sample, the dimensions of the die used to press the sample, and the density of lead titanate, I was able to make an sample appropriate to these conditions. An example spectrum obtained with one of these samples is shown in Fig. 10.1, which has an edge step calculated by the AUTOBK program of 0.413, only 10% larger than the estimation made by ATOMS.

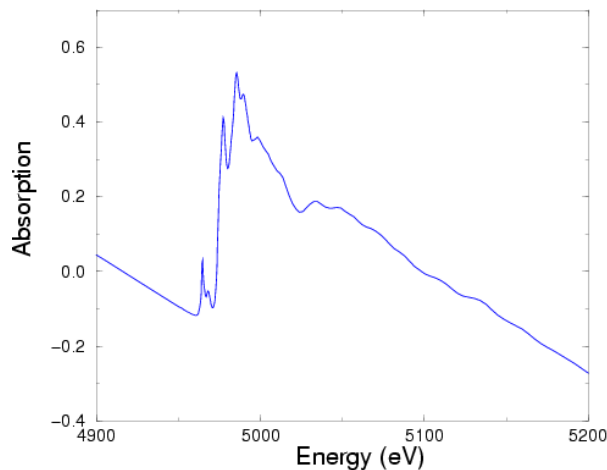


Figure 10.1: Unnormalized absorption spectrum for the titanium K edge of lead titanate at 300 K. The edge step for this scan is 0.413 which differs by only 10% from the predicted edge step of 0.37 for a 5 micron sample.

10.2 The Normalization Correction

To avoid introducing systematic errors into the amplitude of the measured $\chi(k)$, an edge-step normalization is typically used. Since the true atomic background absorption, μ_0 , has energy dependence, normalization by the edge-step introduces an energy-dependent attenuation to the amplitude of $\chi(k)$. This attenuation is small for heavy elements, but can be of the same order as thermal effects for light elements. ATOMS calculates an approximation to this attenuation called the *normalization correction*².

Using the tables of free atoms x-ray cross-sections, ATOMS evaluates the cross-section for the free central atom in a range from 50 to 300 eV above the absorption edge. It then regresses a second order polynomial in the natural log of the energy relative to the edge to the natural log of the free atom cross-section³. The linear term is an approximation to the degree of attenuation introduced by the edge-step normalization. This is intended as additive corrections to the measured σ^2 of a fit. Neglecting this correction will make the temperature dependence of σ^2 deviate from an Einstein behavior by a constant offset.

10.2.1 The I_0 Correction

In a fluorescence experiment the absorption cross section is obtained from the detected intensities on the I_0 and I_F chambers by measuring their ratio as a function of energy. This introduces an energy dependent error, usually an attenuation, into the amplitude of the measured $\chi(k)$. The secondary photon measured in the fluorescence experiment is always of the same energy.

²This correction is often called the McMaster correction in the EXAFS literature

³This is the numerical method used by McMaster et al. in their 1969 tables of cross sections. See Section 12.1

Thus there is no energy dependent part of the signal measured on I_F . There is, though, an energy response to I_0 that is neglected when I_F is normalized by the signal on I_0 . ATOMS calculates an approximation to this attenuation called the I_0 *correction*.

To enable this calculation when using ATOMS, it is necessary to specify the contents of the I_0 chamber. In ATOMS and $\mathcal{T}\mathcal{K}$ ATOMS this is enabled by specifying non-zero values for one or more of the fill gasses. From the supplied values of fill gases, ATOMS approximates the energy response of the I_0 chamber using the tables of free-atom x-ray cross sections.

ATOMS evaluates the cross-section for the gases in the I_0 chamber in a range from 50 to 300 eV above the absorption edge. It then regresses a second order polynomial in the natural log of the energy relative to the edge to the natural log of the gas cross-section. The linear term is an approximation to the degree of attenuation introduced by normalizing I_F by I_0 . These are intended as additive corrections to the measured σ^2 and fourth cumulants of a fit. Neglecting this correction will make the temperature dependence of σ^2 deviate from an Einstein behavior by a constant offset.

10.3 The Self-Absorption Correction

The second correction required for fluorescence measurements is called the *self-absorption correction*. This calculation is also enabled when fill gases are specified. Assuming⁴ equal entry and exit angles, the formula for the fluorescence signal is given by

$$\frac{I_F}{I_0} \approx \frac{\mu_c(E)}{\mu'(E_F) + \mu_b(E) + \mu_c(E)} \quad (10.1)$$

$\mu_c(E)$ is the absorption of the resonant atom, $\mu_b(E)$ is the rest of the absorption in the material, and $\mu'(E_F)$ is the absorption of the material at the fluorescence energy of the resonant atom. The self-absorption correction is due to the $\mu_c(E)$ term in the denominator of this equation. In the limit that $\mu_c(E) \gg (\mu'(E_F) + \mu_b(E))$, the self-absorption correction is enormous, canceling the oscillatory structure of the XAFS spectrum. ATOMS approximates the effect of $\mu_c(E)$ term in the denominator of Eq. (10.1) given the assumption of infinite sample thickness, which, in practice, means that the sample is very thick compared to the absorption length.

Typically, the fluorescence experiment is performed to measure χ rather than μ . The correction presented in this section is a correction to the measured χ . The signal χ is the normalized variation in I_F/I_0 , the oscillatory part of the absorption of the resonant atom. I will now derive expressions for the variation in I_F/I_0 and for the normalization term using the notation of Eq. (10.1).

⁴The correction for non-equal angles is simple, but not currently considered by ATOMS. Also, the assumption of infinite thickness discussed in the next paragraph could be corrected.

The variation in the signal, $\delta(I_F/I_0)$ is due to the variation in $\mu_c(E)$, $\delta\mu_c(E)$

$$\begin{aligned}\delta\left(\frac{I_F}{I_0}\right) &= \delta\left(\frac{\mu_c(E)}{\mu_B(E) + \mu_c(E)}\right) \\ &= \frac{\delta\mu_c(E)}{\mu_B(E) + \mu_c(E)} - \frac{\delta\mu_c(E) \cdot \mu_c(E)}{(\mu_B(E) + \mu_c(E))^2} \\ &= \frac{\delta\mu_c(E) \cdot \mu_B(E)}{(\mu_B(E) + \mu_c(E))^2}\end{aligned}\quad (10.2)$$

In this equation $\mu_B(E) = (\mu'(E_F) + \mu_b(E))$.

The normalization, Δ , is the difference in I_F/I_0 below and above the edge. The symbols $|_a$ and $|_b$ denote that the quantity is evaluated above or below the absorption edge.

$$\Delta = \frac{\mu_c(E)|_a}{\mu_c(E)|_a + \mu_B(E)} - \frac{\mu_c(E)|_b}{\mu_c(E)|_b + \mu_B(E)} \quad (10.3)$$

For a fluorescence measurement where the self-absorption correction is a significant effect, $\mu_c(E)|_a$ is a large term in Eq. (10.1), but $\mu_c(E)|_b$ is assumed to be small. I need one more bit of notation,

$$\mu_c(E)|_a = \mu_c(E)|_b + \Delta\mu_c. \quad (10.4)$$

Using the assumption that $\mu_c(E)|_b$ is small,

$$\begin{aligned}\Delta &= \frac{\mu_c(E)|_b + \Delta\mu_c}{\mu_c(E)|_a + \mu_B(E)} - \frac{\mu_c(E)|_b}{\mu_B(E)} \\ &= \frac{\Delta\mu_c}{\mu_c(E)|_a + \mu_B(E)}\end{aligned}\quad (10.5)$$

Finally I write an expression for the normalized variation in I_F/I_0 using Eqs. (10.2) and (10.5). Since this derivation is for a correction to χ , I am only concerned with normalized variation *above* the absorption edge.

$$\begin{aligned}\frac{\delta(I_F/I_0)}{\Delta} &= \frac{\delta\mu_c(E) \cdot \mu_B(E)}{(\mu_B(E) + \mu_c(E))^2} \div \frac{\Delta\mu_c}{\mu_c(E) + \mu_B(E)} \Bigg|_a \\ &= \frac{\delta\mu_c(E) \cdot \mu_B(E)}{\Delta\mu_c(\mu_c(E) + \mu_B(E))} \Bigg|_a\end{aligned}\quad (10.6)$$

The ideal measurement would be undistorted and simply expressed as $\delta\mu_c(E)/\Delta\mu_c$. The correction factor \mathcal{C}_{self} is the factor by which the measured signal must be multiplied to obtain

the ideal signal.

$$C_{self} \cdot \frac{\delta\mu_c(E) \cdot \mu_B(E)}{\Delta\mu_c(\mu_c(E) + \mu_B(E))} = \frac{\delta\mu_c(E)}{\Delta\mu_c} \quad (10.7)$$

$$\begin{aligned} C_{self} &= \frac{\mu_c(E) + \mu_B(E)}{\mu_B(E)} \\ &= 1 + \frac{\mu_c(E)}{\mu_B(E)} \end{aligned} \quad (10.8)$$

ATOMS uses the free atom cross-sections to evaluate Eq. (10.8) in the energy range from 50 to 300 eV above the absorption edge⁵. A second order polynomial in the natural log of the energy relative to the edge is regressed to the natural log of Eq. (10.8).

The amplitude factor is the exponent of the constant term in the regression. This is intended as a multiplicative correction to the amplitude of the data. If this term is neglected in a fit, the measured S_0^2 will be too small by that factor. For pure metals and K edges, this factor can be 10 or more. But one should not measure pure metals in fluorescence...

The other correction is the linear term of the regression and is expressed as a second order cumulants. This is intended as an additive corrections to the measured σ^2 of a fit. Neglecting this correction will make the temperature dependence of σ^2 deviate from an Einstein behavior by a constant offset.

⁵Care is taken in the code to avoid running into other absorption edges in the material. If there is another absorption edge within 300 eV the range is truncated appropriately.

Chapter 11

Contributing to the ATOMS Project

As mentioned in the Abstract of this document and in Frequently Asked Question [A.1.5](#), ATOMS is Open Source software. Among other things, this means that I welcome contributions to the effort. Someday ATOMS will be part of a large, sophisticated x-ray absorption spectroscopy theory and analysis package. Before that happens, lots of chores need to be finished. Eventually, I will get around to doing everything that needs to be done with ATOMS, but I am quite eager for any interested parties to step up and offer code or other useful services. I cannot offer fame, fortune, and glory to anyone who helps out, but I can offer my profound gratitude and a prominent spot for your name in the document.

Here is a list of things that remain undone as of ATOMS 3.0alpha24. They are in no particular order and they range in size from fairly quick and easy to fairly challenging. One could probably think of other interesting things to do. Virtually any contribution will be welcomed. Suggestions, comments, criticisms, and bug reports (see FAQ [A.3.5](#) on page [70](#)) are also quite welcome.

Write a good algorithm for reading arbitrary angles.

OK, this one is not really that hard, but I have been too lazy to do it. The idea is that one might write *eighty seven and one half degrees* as $87^{\circ} 30'$. What I am looking for is a fast, reliable algorithm for correctly interpreting decimal degrees or minute/seconds. The algorithm should be highly fault tolerant. $87^{\circ} 30'$, 87.5 , $87^{\circ} 30m$, $87^{\circ} 29' 60''$, $87^{\circ} 29m 60s$, and other sensible combinations, including varying amounts of whitespace, should all evaluate to the same number of radians.

Generalize the self-absorption correction

The self-absorption correction calculated in the ATOMS.PM module uses the assumptions of infinite thickness and equal entry and exit angles. This would be easy enough to modify.

Tetragonal F and C groups

The ability to interpret tetragonal crystals in their F or C settings has recently been added to ATOMS and needs to be tested. For this I need examples of crystallographic data of tetragonal crystals in those (rather than the P and I) settings.

Translation of language data

I already have a good start on the Spanish translation thanks to Dani Haskel and a mostly complete French translation thanks to Stephane Grenier. Other languages would be great. Soon (i.e. with Perl 5.6) we will have the possibility of multi-byte languages, such as Japanese, Chinese, and Korean. I estimate that a native speaker who is reasonably fluent in English could finish a translation about 6 hours spread out over a few days. Contact me for how to get started.

Translations of element names

I have already started work on the internationalization of APT, but I am lacking a way of getting element names in any language. `Chemistry::Elements` provides object methods for handling element data. It seems that a good way of internationalizing element names would be to use it. A few hours of coding plus a few hours of collecting element names in other languages.

Object methods for reading molecular structure data

By this I mean reading, for example, a BNL Protein Data Bank file, parsing out the atom coordinates and storing them in an appropriate data structure. The `Xray::File` module is a start on this, but it doesn't do much yet. I would be happy to write a test harness for anyone interested in working on this. I suspect that it would take a day or two get a single inherited method working and tested once the basic structure of `Xray::File` is clear. Others should then go faster.

Streamline the Mac distribution

The `'atoms.pl'` and `'dafs.pl'` programs (i.e. the non-Tk programs) work fine on the Mac, but there are some installation difficulties. I would love to see a streamlined pure-perl installation procedure for the Mac. This should be a day or two of work for someone who knows both Perl and the Mac pretty well.

Write a GUI for the Mac using Chris Nandor's Mac-GUI toolkit.

Unless perl/Tk ever gets ported, this is the only good way of doing a Mac GUI. This is probably a pretty big project. Perhaps it would be better to help with the Mac port of perl/Tk or wait for the holy grail of OS X.

Help me test the distribution on platforms I do not have access to.

I develop ATOMS on Linux and can get my hands on HP workstations and a Windows 95 box. With some more difficulty, I can get onto a Sun Station and an SGI machine. That leaves out a lot of important platforms! Are you using something else? Let me know how it goes.

Help me port ATOMS to VMS

The Tk part may be a bit tricky. The last time I looked, the VMS port of perl/Tk was incomplete. This could be a sizable chore, although I have been careful about things like file paths. ATOMS runs on the Mac, after all!

Port the Cromer-Liberman Kit to new platforms.

This currently works on SGI and Linux. Porting to other Unixes shouldn't be too hard. In fact, I cannot imagine it would take me more than an hour or two for any

platform if I set my mind to it. I suspect it would be possible for both Windows and the Mac, but it would require some familiarity with those development environments. You need to compile Fortran source code into a shared object and link it with some C code.

Niftier appearance of WebATOMS

It would be nice to use style sheets to make the web application look extra spiffy. A style sheet for the html version of the document would be great also.

Input file database

It would be handy to build some kind of database architecture for storing ATOMS input files in a way that ATOMS, $\mathcal{T}\mathcal{K}$ ATOMS, and *Web*ATOMS can all get at them. Using real DB concepts (B-trees, queries, and so on) seems appropriate so that the database can scale well.

Write the powder diffraction simulator

Most of the pieces are there already, but a few (such as multiplicity) are missing. I suggest writing a stand-alone version like `'dafs.pl'` and leaving me to hook into the $\mathcal{T}\mathcal{K}$ ATOMS framework. This is a few days to a few weeks worth of effort.

Write the framework of a ball-and-stick browser

I am thinking that the perl/OpenGL interface is the way to go, but its all rather opaque to me. If someone with experience in OpenGL or Mesa could help me get started, that would be splendid. This is a huge project.

Chapter 12

Resources and Bibliography

12.1 Literature References

1. T. Christiansen and N. Torkington, *The Perl Cookbook*, O'Reilly and Associates, Cambridge MA (1998)
2. J.M. Cowley *Diffraction Physics*, North-Holland, Amsterdam (1990)
3. The International Tables of Crystallography
4. L. Stein, *Official Guide to Programming with CGI.pm*, John Wiley and Sons, New York (1998)
5. L. Wall, T. Christiansen, and R. Schwartz, *Programming Perl*, O'Reilly and Associates, Cambridge MA (1996)
6. N. Walsh, *Learning Perl/Tk: Graphical User Interfaces with Perl*, O'Reilly and Associates, Cambridge MA (1999)
7. B.E. Warren, *X-ray Diffraction*, Dover Publications, New York (1990)

12.2 Absorption Data Resources

1. **The McMaster Tables**
W.H. McMaster, N. Kerr Del Grande, J.H. Mallett, J.H. Hubbell
Compilation of X-Ray Cross Sections
National Bureau of Standards
UCRL-50174 Section II Revision 1 (1969)
Available from National Technical Information Services L-3
United States Department of Commerce
2. **The Henke Tables**
B. L. Henke, E. M. Gullikson, and J. C. Davis
Atomic Data and Nuclear Data Tables **54** No. 2 (July 1993)

3. The Chantler Tables

C. T. Chantler

Theoretical Form Factor, Attenuation, and Scattering Tabulation for $Z = 1 - 92$ from $E = 1 - 10$ eV to $E = 0.4 - 1.0$ MeV

J. Phys. Chem. Ref. Data **24**, p. 71 (1995)

<http://physics.nist.gov/PhysRefData/FFast/Text/cover.html>

<http://physics.nist.gov/PhysRefData/FFast/html/form.html>

<http://optics.ph.unimelb.edu.au/~chantler/home.html>

4. The Elam Tables

W. T. Elam, unpublished

tim.elam@nrl.navy.mil

5. The Cromer-Liberman Tables

D.T. Cromer and D. Liberman

J. Chem. Phys., v. 53, p. 1891 (1970)

S. Brennan and P.L. Cowen

Rev. Sci. Instrum, **63**, p. 850 (1992)

<http://www.slac.ssrll.stanford.edu/absorb.html>

modified and adapted to perl by M. Newville.

12.3 Resources on the World Wide Web

- Perl Web Pages

1. Perl.com

<http://www.perl.com/>

2. ActiveState Perl for Windows

<http://www.activestate.com/>

3. MacPerl

<http://www.ptf.com/macperl/>

- Physics Web Pages

- The ATOMS homepage

<http://feff.phys.washington.edu/~ravel/atoms/3.0.html>

- The FEFF homepage

<http://feff.phys.washington.edu/feff/>

- The IFEFFIT homepage

<http://cars.uchicago.edu/~newville/ifeffit/>

Appendix A

Frequently Asked Questions

A.1 General Questions

- A.1.1 What is ATOMS?
- A.1.2 Who wrote ATOMS?
- A.1.3 Why was ATOMS written?
- A.1.4 Where do I find ATOMS?
- A.1.5 What does ATOMS cost?
- A.1.6 Are there other programs like ATOMS?

A.2 Installing ATOMS

- A.2.1 How do I install ATOMS?
- A.2.2 How do I find and install the necessary CPAN modules?
- A.2.3 `'check-cpan'` says 3.15 of `'Tk:Pod'` is necessary, but I can only find 0.11.
- A.2.4 Which CPAN modules are used by ATOMS?
- A.2.5 Make dies trying to make the databases
- A.2.6 The installation dies during `'make install'`
- A.2.7 `private-install` says `''Neither PERL5LIB nor PERLLIB are set.''`
- A.2.8 I get lots of messages about failed `platform tests` during `make test`.
- A.2.9 Is there an icon for my desktop/wharf/panel/whatever?
- A.2.10 How do I figure out where ATOMS is installed on my computer?

A.3 Running ATOMS

- A.3.1 I have installed ATOMS, but running `"atoms"` doesn't work.
- A.3.2 Why does ATOMS tell me `"index is no longer a keyword"`?
- A.3.3 How do I make ATOMS use a language other than English?
- A.3.4 How do I add a new language to ATOMS?
- A.3.5 I think I found a bug in ATOMS. What should I do?
- A.3.6 `TKATOMS` fails under the RedHat 5.2 Linux distribution.
- A.3.7 I get a weird error with the `"Restore"` button in the Space Group Browser
- A.3.8 `TKATOMS` hangs or crashes the second time I open help.
- A.3.9 The online help does not work on Windows.

- A.3.10 I can't see the help balloons when $\mathcal{T}\mathcal{K}\text{ATOMS}$ is on the right of my screen
- A.3.11 The previous DAFS plot did not get erased in $\mathcal{T}\mathcal{K}\text{ATOMS}$
- A.3.12 I moved the MacPerl folder on my Mac and now ATOMS doesn't work.
- A.3.13 I moved a FEFF input file from one computer to another and FEFF does not run.

A.4 Space group symbols

- A.4.1 I get incorrect output for my monoclinic or orthorhombic space group.
- A.4.2 ATOMS interpreted a Hermann-Maguin symbols without spaces incorrectly.

A.5 Output files

- A.5.1 How do I change the format of ATOMS' output files?
- A.5.2 How do I define new output formats for ATOMS?
- A.5.3 How do I change the unique potentials assignment in feff.inp files?
- A.5.4 How do I reduce the feff8 stoichiometry list by a common denominator?
- A.5.5 How do I make ATOMS write a feff8 input file by default?
- A.5.6 How do I make a ball-and-stick image of my crystal?

A.6 The DAFS program

- A.6.1 What does the DAFS program calculate?
- A.6.2 How do I look at the complex parts of the DAFS calculation?
- A.6.3 Why is the plot in $\mathcal{T}\mathcal{K}\text{ATOMS}$ so primitive?
- A.6.4 Why do Henke and Chantler produce non-smooth output?
- A.6.5 How do I install the Cromer-Lieberman tables?

A.1 General Questions

A.1.1 What is ATOMS?

ATOMS is program for interpreting crystallographic data and writing out lists of atomic coordinates and other useful kinds of output. ATOMS is also a set of object-oriented modules useful for other crystallographic and x-ray absorption calculations. ATOMS is written in perl.

A.1.2 Who wrote ATOMS?

ATOMS was written by Bruce Ravel. You can contact him by email at ravel@phys.washington.edu. ATOMS is copyright © 1999 Bruce Ravel.

A.1.3 Why was ATOMS written?

Originally ATOMS was a Fortran program written to aid in the task of writing input files for Feff, a program for *ab initio* x-ray absorption theory. As ATOMS grew and was used for more purposes, the limitations of the original Fortran implementation became annoying to the author. Consequently, ATOMS was rewritten from scratch in Perl using object oriented methods. This resulted in a code base that is easier to maintain and extend and allowed many new features to be added to ATOMS.

A.1.4 Where do I find ATOMS?

The ATOMS home page is at <http://feff.phys.washington.edu/~ravel/atoms>.

A.1.5 What does ATOMS cost?

Nothing. It's free under the terms of the Perl Artistic License. In brief, legally inaccurate terms, this means that you are free to use, modify, and redistribute ATOMS as you please, with the caveat that, if you modify the code I wrote and redistribute it, you must share your changes with me. It also means that, if you write your own code using the crystallography or x-ray absorption modules that comes with ATOMS, that code belongs to you. For more details see the Perl Artistic License at <http://feff.phys.washington.edu/~ravel/atoms/Artistic.txt>.

If you like ATOMS, send me email telling me so. Email from users is one of the great pleasures of free software. If you are so taken by ATOMS that you want to express your gratitude in a material manner, then you may make a small donation in my name to my favorite charity, The Women's Foundation of San Francisco. The Women's Foundation serves as a voice and advocate for the needs of women and girls by providing funding and resources and is a convener of groups and individuals for dialogue and collaboration regarding economic, social, and political justice.

General operating budget
The Woman's Foundation
340 Pine Street, Suite 302
San Francisco CA 94104 USA
phone: 1-415-837-1113
fax: 1-415-837-1144
pchang@twfusa.org

Many corporations make matching donations to charitable organizations.

A.1.6 Are there other programs like ATOMS?

Yes there are. The program SEXIE (<ftp://xena.llnl.gov/sexie/> Computer Physics Communications 67, (1992) 543-547 Bernhard Rupp, Bryan Smith and Joe Wong) was one of the original inspirations for ATOMS. The original version of ATOMS drew many ideas from SEXIE. Eventually, SEXIE was modified to replicate the functionality of ATOMS.

There are a huge number of programs useful to the crystallographer which read crystallographic data and write out lists of atomic coordinates. If you poke around crystallography web sites, you will find many such programs.

There are at least good three reasons to use ATOMS:

1. it is free
2. it is specifically designed to be useful to FEFF users and other XAFS types.
3. the crystallography and x-ray absorption in ATOMS is encoded in the form of reusable, object-oriented perl modules, so you can easily write new programs using them

A.2 Installing atoms

A.2.1 How do I install ATOMS?

1. Unix installation

If you have perl5 properly installed on your computer, installation is a breeze. Just download and unpack the latest distribution. In the unpacked subdirectory, do

```
perl Makefile.PL
make
make test
make install
```

That's it! If you are installing ATOMS as a normal user rather than as root, you should run the 'private-install' script that comes with ATOMS rather than doing *make install*. Rehash and you're good to go.

2. Windows 9x/NT installation.

Download the latest zip file for Windows. Unpack it using WINZIP or some similar program. Then double click on the 'install.bat' icon.

3. MacOS installation.

Download the self-extracting archive and double click on the 'MacInstall.pm' icon.

A.2.2 How do I find and install the necessary CPAN modules?

The easiest thing to do is to download and install the ATOMS Bundle. It contains everything you need. There is a Bundle for Unix which installs in the standard CPAN manner. The required CPAN modules are included in the Windows and Macintosh distributions. You can also download the modules by hand. See question [A.2.4](#).

A.2.3 'check-cpan' says 3.15 of 'Tk::Pod' is necessary, but I can only find 0.11.

Don't fret. The version number advertised by 'Tk::Pod' that 'check-cpan' reads is 3.15. The version number it is distributed under is 0.11. Isn't that annoying?

A.2.4 Which CPAN modules are used by ATOMS?

These are integral parts of ATOMS:

1. Statistics::Descriptive
2. Chemistry::Elements
3. File::Spec
4. Storable
5. Data::Dumper

These are needed by the Elam tables of absorption data:

1. Math::Spline
2. Math::Derivative

The Tk version of ATOMS requires:

1. Tk
2. Tk::File::Dialog
3. Tk::Pod

A.2.5 Make dies trying to make the databases

You have not installed the ATOMS Bundle.

A.2.6 The installation dies during *make install*

You must be root to do *make install* because it tries to write to a system directory. As a normal user, use the 'private-install' script instead.

If you are root and *make install* fails, the look at the screen messages from the failed *make install*. They will be helpful to you for diagnosing the problem or when making a bug report. Also look at the output of *perl -V* to make sure the install directory is under one of the directories in @INC.

A.2.7 'private-install' says Neither PERL5LIB nor PERLLIB are set.

When installing as a normal user, 'private-install' needs to know where to put all of the stuff that ATOMS needs. It determines this from the value of an environment variable. To let 'private-install' continue, designate a place where you want to store perl stuff. '\$HOME/perl/' is a good choice. Then, if you are a bash user put these lines in your '.profile'

```
set PERL5LIB $HOME/perl/ export PERL5LIB
```

If you are a c-shell user, put these lines in your '.cshrc'

```
setenv PERL5LIB $HOME/perl/
```

Run your '.profile' or source your '.cshrc' and you are ready to install ATOMS.

A.2.8 I get lots of messages about failed platform tests during *make test*.

Please send me a bug report. These reports are very valuable to me. Include the type of computer you are using, the version number of perl, and the file 'test1.log' which you should find in the main ATOMS after running *make test*. I might need to ask you for more information, which will involve about 5 minutes of work on your part.

One of the goals of ATOMS is to have it run identically on all platforms. This means that any number written to output files must be the same on all computers within 0.00001. In my experience with recent versions of ATOMS, most failed platform tests are due to differences in the that digit when comparing the output file generated during 'make test' with those used in the test. Probably, ATOMS is fine even if it fails some platform tests. Be sceptical and check the output carefully, but go ahead and start using ATOMS.

A.2.9 Is there an icon for my desktop/wharf/panel/whatever?

Yes there is. There is a page of icons in three sizes and five image formats on the ATOMS web page. The smallest, 16x16 pixels, is good for use in a menu. The larger sizes, 32x32 and 48x48, are good for use on the desktop. This same set of icons can be found in the 'etc/' directory of the Unix ATOMS distribution.

A.2.10 How do I figure out where ATOMS is installed on my computer?

At the command line, type

```
perl -e 'use Xray::Xtal; print $INC"Xray/Xtal.pm", $/'
```

Under MS Windows use a DOS window. Under Macintosh use the MacPerl “one-liner” option. If you have trouble, try putting the two commands in between the single apostrophes in a file and executing that file as a perl script.

A.3 Running atoms

A.3.1 I have installed ATOMS, but running *atoms* doesn't work.

Until ATOMS 3.0 is officially released, I am calling the executable ‘*atoms.pl*’ to avoid name conflict with the Fortran version. Similarly, the *dafs* and *Tk* programs are called ‘*dafs.pl*’ and ‘*tkatoms.pl*’.

A.3.2 Why does ATOMS tell me “‘index is no longer a keyword’”?

The functionality of the “index” keyword in earlier versions of ATOMS has been moved to the *atp* files. If you want to turn shell indexing on or off, do so with the <itag> and <tag> tokens in the *atp* file. The standard ‘*feff.atp*’ file uses shell indexing. You may find it useful to keep a ‘*feff_noindex.atp*’ (or some such) in your ‘*~/.atoms*’ directory.

A.3.3 How do I make ATOMS use a language other than English?

ATOMS reads runtime configuration data from a system file and from a personal file called ‘*atomsrsrc*’ which it looks for in the ‘*~/.atoms/*’ directory. One of the variables that can be set is *\$atoms_language*. Set this to your choice of language. If the language data in ATOMS has been translated into your language choice, then that will be used. If not, then ATOMS will default back to English.

A.3.4 How do I add a new language to ATOMS?

Language data does not magically appear. (Darn!) I am happy to ship any set of language data with ATOMS for which I am given a translation. If your language is not included in ATOMS, that means the translation has not yet been made. If you volunteer to make a translation, I will be grateful forever. Contact me for details.

A.3.5 I think I found a bug in ATOMS. What should I do?

Check out the questions in this FAQ. If your bug is not one of those, then submit a bug report to me at ravel@phys.washington.edu. A good bug report consists of some or all of the following information:

1. The version of ATOMS you are running.
2. The kind of computer you are using and the version of perl you are running. Running perl with its “-v” switch will tell you the version number. If the problem is with a program written with perl/Tk, the Tk version number is helpful also. You can find this out by typing

```
perl -e 'use Tk; print $Tk::VERSION,$/'
```

at the command line.

3. The input data that caused the problem.
4. The literature reference for your input data.
5. Any screen messages generated by your ATOMS run. Please copy these exactly as they appear rather than trying to paraphrase. Running ‘atoms.pl’ or ‘tkatoms.pl’ with the “-v” switch also gives information which will be useful to me.
6. The incorrect output file – assuming that the bug did not preclude your output file from being written.
7. If the FEFF input file written by ATOMS fails to run FEFF correctly, capture any screen messages written by FEFF. Make sure to tell me what version of FEFF you are running.

This URL (<http://freshmeat.net/news/2000/02/26/951627540.html>) points to an article entitled *How to Report Bugs Effectively* by Simon Tatham. It provides an excellent overview of how and why to write a bug report.

A.3.6 \mathcal{T}_{K} ATOMS fails under the RedHat 5.2 Linux distribution.

The perl distributed with RH 5.2 is severely broken. The only really good solution is to download the source code for perl from CPAN and compile up a new version. Don't fret, compiling perl is easy, although a bit time consuming. Just follow the installation instructions that come with the source code distribution.

A.3.7 I get a weird error with the Restore button in the Space Group Browser

This error probably says something like

```
Global symbol "$Restore the initial symbol: pm3m"
requires explicit package name at
/usr/lib/perl5/5.00502/i586-linux/Tk/Balloon.pm line 331.
```

I am aware of this, but don't know what causes it. It is probably a bug in the Tk help balloon code.

A.3.8 $\mathcal{T}\mathcal{K}$ ATOMS hangs or crashes the second time I open help.

Yup. It sure does. I haven't found the cause of this bug yet. I have noticed many bug reports on Usenet regarding Tk::Pod. As soon as it gets fixed, I'll include the fix in the AtomsBundle and mention it on the ATOMS webpage. It may suffice to upgrade either ATOMS or perl, but, then again, it might not.

A.3.9 The online help does not work on Windows.

Nope. It sure doesn't. This whole feature is disabled on Windows in ATOMS 3.0alpha24.

A.3.10 I can't see the help balloons when $\mathcal{T}\mathcal{K}$ ATOMS is on the right of my screen.

By default Tk's help balloons fall down and to the right of the widget they are attached to. It is a short coming of the help balloon code that this is also the only behavior. I suppose one solution is to always have $\mathcal{T}\mathcal{K}$ ATOMS on the left side of the screen, but that is surely not the answer you are looking for. In the 'etc/' directory of the ATOMS distribution is a file called 'patch.balloon'. This is a patch against the balloon code in the perl installation. Applying this patch requires that you have root access. The balloon code is the file 'Balloon.pm' which can be found in the place where Tk was installed on your computer. If you have the patch program on your computer, you can apply the patch by moving it to the same directory as 'Balloon.pm' and doing

```
patch < patch.balloon
```

This patch prevents the balloon from displaying off screen. However, it is an incomplete solution. The little arrow in the corner of the balloon identifying the widget that the balloon is attached to will likely be in the wrong place. At least you can read the balloon!

A.3.11 The previous DAFS plot did not get erased in $\mathcal{T}\mathcal{K}$ ATOMS

This is a bug that happens when you double click the "Run DAFS" button and have bad luck. It should happen very rarely. Unfortunately, without my rewriting a significant portion of $\mathcal{T}\mathcal{K}$ ATOMS, the only solution is to kill $\mathcal{T}\mathcal{K}$ ATOMS and re-launch it.

A.3.12 I moved the MacPerl folder on my Mac and now ATOMS doesn't work.

Probably you moved the folder while MacPerl was running. Quit MacPerl completely then try double clicking on the 'atoms.pl' icon again.

A.3.13 I moved a FEFF input file from one computer to another and FEFF does not run.

Unix, Microsoft, and Apple use different conventions for denoting the end of line in a text file. You must take care to transfer ATOMS' output files between platforms as ASCII files. Using ftp, be sure to specify ASCII transfer. Using a removable disk (floppy, Zip, whatever) or network file sharing, be sure to transform the text file after transferring it.

A.4 Space Group Symbols

A.4.1 I get incorrect output for my monoclinic or orthorhombic space group.

Monoclinic space groups have an inherent ambiguity regarding their setting in three dimensional space. There are 6 possible orientations of the unique and secondary axes. In the case of monoclinic and orthorhombic groups, the symbols corresponding to different settings are different from one another. The most likely cause of your problem is that you used a symbol that is correct for your group but incorrect for the setting. See Section 3.2.2 and Appendix C.

A.4.2 ATOMS interpreted a Hermann-Maguin symbols without spaces incorrectly.

If you use 1935 symbols for certain tetragonal and cubic groups, the space inserting algorithm will fail. For example, if you use the symbol p422, ATOMS assumes you mean p 4 2 2 using the modern notation, rather than p 42 2 in the 1935 notation. To resolve this ambiguity, use spaces.

The space groups for which the 1935 notation sans spaces will fail are:

number	1935 symbol	modern symbol
89	p 4 2	p 4 2 2
90	p 4 21	p 4 21 2
93	p 42 2	p 42 2 2
95	p 43 2	p 43 2 2
97	i 4 2	i 4 2 2
98	i 4 21	i 4 21 2
207	p 4 3	p 4 3 2
211	f 4 3	f 4 3 2
214	i 4 3	i 4 3 2

A.5 Output Files

A.5.1 How do I change the format of ATOMS' output files?

The output files are formatted using template files found in the 'atp/' directory underneath the installation location of ATOMS. The syntax and use of the template files (atp files) is discussed in the atp document which comes in various formats with ATOMS. To change the format of an output file, either edit the atp files which come with ATOMS or make a copy of the atp file in '~/.atoms/' and alter it as desired. On a Mac or Windows 9x, the altered atp file should be placed in the 'atp' folder in the ATOMS installation folder.

A.5.2 How do I define new output formats for ATOMS?

Make a new atp file according to your specifications and place it in '~/.atoms/'. On a Mac or Windows 9x, the new atp file should be placed in the 'atp' folder in the ATOMS installation folder.

A.5.3 How do I change the unique potentials assignment in feff.inp files?

There are three schemes for assigning unique potentials. They are assignment by atomic species, by user-supplied tag, or by crystallographic site. Change the value of the :ipots argument to the <potentials> token in 'feff.atp' to be one of species, tags, or sites. I welcome suggestions for new assignment schemes.

A.5.4 How do I reduce the FEFF8 stoichiometry list by a common denominator?

Change the :gcd argument of the <potentials> token in 'feff8.atp' from 0 to 1.

A.5.5 How do I make ATOMS write a FEFF8 input file by default?

If you set the variable \$prefer_feff_eight to 1 in the 'atomsrc' file, then the feff8 input file becomes the default output. This means that 'atoms.pl' will write a FEFF8 input file when no file type is specified and that $\mathcal{T}\mathcal{K}$ ATOMS starts up with the FEFF8 output option selected.

A.5.6 How do I make a ball-and-stick image of my crystal?

Some day I may write such a thing for ATOMS. For now, you will have to use one of ATOMS' output files with your favorite ball-and-stick plotter. ATOMS supports two common ball-and-stick file formats and you can create a suitable atp file for your favorite format.

A.6 The DAFS Program

A.6.1 What does the DAFS program calculate?

It calculates a crude approximation of a DAFS scan. Specifically it computes the energy dependent structure factor using tables of f_0 , f' , and f'' . However, it does not include solid state effects, so includes none of the fine structure. It does, however, give you a sense of the relative magnitude of the reflection and of the cusp as well as which direction the cusp points. It should not be misconstrued as an analytical tool.

A.6.2 How do I look at the complex parts of the DAFS calculation?

The 'dafs.atp' file that comes with ATOMS writes out energy versus amplitude squared. The amplitude, phase, real, and imaginary parts of the complex DAFS function are available using the <a>, <p>, <r>, and <i> tokens in the dafs list. Either edit the 'dafs.atp' file or make a new one which includes the columns that you want to look at.

A.6.3 Why is the plot in $\mathcal{T}\mathcal{K}_{\text{ATOMS}}$ so primitive?

The canvas in the DAFS notecard in $\mathcal{T}\mathcal{K}_{\text{ATOMS}}$ is only meant to be a snapshot of the calculation. As such it is a very simple plot. You can save the data and plot it using your favorite plotter. I am working on a flexible way of sending data directly from $\mathcal{T}\mathcal{K}_{\text{ATOMS}}$ to an external plotting tool.

A.6.4 Why do Henke and Chantler produce non-smooth output?

These two data resources are interpolated from a rather sparse grid. Following Chantler's suggestion of semi-log interpolation of f' results in a function that is discontinuous in first derivative at each actual grid point. A future improvement might be to do some sort of spline interpolation, but a much better solution would be to convince Henke and Chantler to release their codes on a more appropriate grid near the edge energies. Another option is to install the 'CLkit', the Cromer-Lieberman kit, which uses the Brennan/Cowen implementation as an external subroutine.

A.6.5 How do I install the Cromer-Lieberman tables?

Download the file 'CLkit_Atoms-3.0alpha##' (where ## is appropriate to your version of ATOMS) from the ATOMS homepage. This is a drop-on-top

enhancement to ATOMS. You should unpack CLkit in the same place that you unpacked the ATOMS distribution. CLkit then adds several new files to the ATOMS installation directories, and overwrites several others. Once you have unpacked CLkit, just cd into the main ATOMS installation directory and re-do the standard installation procedure:

```
perl Makefile.PL  
make
```

then

```
make install
```

or

```
private-install
```

Note that you must re-do the *perl Makefile.PL* step so that all the ‘Makefile’s get remade to allow compilation of the Cromer-Liberman code and of the wrapper used to call the Cromer-Liberman codes from perl. Once you have remade ATOMS, you can use the Cromer-Liberman tables by specifying that $\mathcal{T}_{K\text{ATOMS}}$, ‘atoms.pl’, or ‘dafs.pl’ use the “CL” tables.

Appendix B

ATOMS Template Files

B.1 Introduction to the ATP Format

ATOMS has no hard-wired output types. All output from ATOMS is controlled using files called ATOMS *template files*, or *atp files*. ATOMS comes with a variety of useful atp files. On a unix computer, the standard atp files are found in '\$PERL5LIB/Xray/atp/'. With Window 95/NT or MacOS, they are found in the 'atp' folder, which is in the 'Xray' folder under the perl's installation site. The standard atp files are:

In ATOMS you choose your output file by specifying the name of the atp file and, optionally, the filename for the output file. For example, in an ATOMS input file, a symmetry table is chosen by inserting this line

```
output  symmetry  symmetry.dat
```

The syntax of that line is

```
output  atp_type  file_name
```

and the `atp_type` is the name of the atp file without the '.atp' extension. In $\mathcal{T}\mathcal{K}$ ATOMS, the symmetry file would be chosen by clicking on its choice button and entering a filename in the space next to the button. In the web version of ATOMS, a symmetry table would be obtained by clicking on its radio button.

ATOMS scans your computer for available atp files and checks your selection against the available options. There are two sites that ATOMS searches for atp files. One is the standard location mentioned above. The other is the user's personal ATOMS configuration directory. On a unix computer, the personal configuration directory is '\$HOME/.atoms'. On Windows 95/98 and MacOS, there is no personal directory since the user may freely alter the standard atp files. On Windows NT, *the personal directory must be somewhere*.

You may create entirely new atp files and place them in your configuration directory. You may also alter standard atp files by copying them to your configuration directory and editing

Table B.1: Standard ATP files

file	purpose
feff.atp	Write an input file for FEFF6.
feff8.atp	Write an input file for FEFF8.
p1.atp	Write the entire contents of the unit cell in the P 1 space group.
unit.atp	Write the overfilled contents of the unit cell.
atoms.atp	Write out the current data as an ATOMS input file.
dafs.atp	Write out the results of a DAFS simulation.
absorption.atp	Write out the results of an absorption data calculation.
molecule6.atp	Write an input file for FEFF6 using atomic coordinates of a molecule.
molecule8.atp	Write an input file for FEFF8 using atomic coordinates of a molecule.
template6.atp	Write an empty template for a FEFF6 input file.
template8.atp	Write an empty template for a FEFF8 input file.
alchemy.atp	Write a cluster of atoms in the <i>alchemy</i> format.
xyz.atp	Write a cluster of atoms in the <i>xyz</i> format.
geom.atp	Write a 'geom.dat' file for use with FEFF's path finder.
symmetry.atp	Write a symmetry table for the current space group.
gnxas_cry	Write a file for the GNXAS CRYMOL program in the "CRY" format.
gnxas_sym	Write a file for the GNXAS CRYMOL program in the "SYM" format.
formulas.atp	Write a file containing math expressions for every atom site.
test.atp	Write an ATOMS diagnostic file.

them. ATOMS searches the personal directory before the standard directory. If, for example, 'feff.atp' is found in both places, the one from the personal directory will be used.

The next section describes the syntax of the atp file. The final section describes a simple major mode for Emacs which can be used as an aid in writing successful atp files.

B.2 The Syntax of ATP Files

B.2.1 File Names

The only hard rule about atp files is that their names end in the characters '.atp'. When ATOMS goes searching for template files, it only finds those which end in '.atp'. There are several heuristics that $\mathcal{T}\mathcal{K}\text{ATOMS}$ follows when displaying the available atp files. It is important that you are aware of these heuristics should you make your own atp files and wish to use them with $\mathcal{T}\mathcal{K}\text{ATOMS}$.

1. The atp file used for saving the crystallography data in the lower panel of $\mathcal{T}\mathcal{K}\text{ATOMS}$ is called 'atoms.atp'. You can make your own file for this purpose and put it in your '.atoms/' directory, but it must be called 'atoms.atp'.
2. Similarly, the file for saving the absorption calculation is called 'absorption.atp'.
3. The DAFS notecard in $\mathcal{T}\mathcal{K}\text{ATOMS}$ builds a menu of atp files whose names contain the characters dafs. This check is made case-insensitively. Suppose you wanted a DAFS output file with the real and imaginary parts of the form factor, you might call this 'dafs_ri.atp', 'ri_DAFS.atp', or something similar. The regular expression used to identify atp files for DAFS is /dafs/i.
4. The Molecule notecard builds a menu of atp files whose names start with the characters molec. This check is made case-insensitively. The regular expression used to identify atp files for molecules is /^molec/i.
5. Any atp file which meets one of the criteria listed above is *not* displayed in the table of available output file formats in the Atoms notecard in $\mathcal{T}\mathcal{K}\text{ATOMS}$. However any atp file can be used with the command line version and the -t and -o switches.

B.2.2 Syntax Overview

ATOMS reads atp files with a one-pass filter. This means that the atp file is read line-by-line, and at each line ATOMS follows the instructions contained on that line for generating output. The atp syntax thus consists of a set of tokens which ATOMS replaces with the indicated text. Each token consists of a tagword surrounded by angle brackets. A token might also contain arguments which are also placed within the angle brackets. Here are examples, one of a token without arguments and one of a token with arguments:

```
<rmax>
<potentials :ipots species>
```

Note that tokens with no arguments contain no white space. Arguments are always words beginning with a colon and followed by the value of the argument. Values can consist of multiple words, in which case they must be enclosed in double quotes.

A shortcoming in the current parser is that the close angle bracket > cannot be used as part of an argument value. For example, something like ">>>" is not an acceptable prefix for <id>, <corrections> or other tokens.

As ATOMS parses through the atp file, it replaces each token by the appropriate text. Any text in the atp file that is not a token is passed through unfiltered. Here is an example atp file. This describes a unit cell output file.

```
<com> This file is part of the Atoms distribution.
<com> Atoms is copyright (c) Bruce Ravel 1998.
<com> This is the atoms template file for an p1.inp file
<com> Atoms homepage: http://feff.phys.washington.edu/~ravel/atoms/
<com>
<meta :file "p1.inp file" :precision 9.5 :output "p1.inp">
<id :prefix "! ">
<titles :prefix "title = ">
space  p 1      ! original space group: <space>
a =      <a>      b =      <b>      c =      <c>
alpha = <alpha> beta = <beta> gamma = <gamma>
core = <central>      edge = <edge>  rmax = <rmax>
atoms
! elem  x          y          z          tag
<list :style unit>
  <elem> <x> <y> <z> <itag>
```

This generates output that looks something like this:

```
! This p1.inp file was generated by Atoms, version 3.0alpha21
! Atoms written by and copyright (c) Bruce Ravel, 1998
title = Copper example
space  p 1      ! original space group: F m -3 m
a =      3.810    b =      3.810    c =      3.810
alpha = 90.0     beta = 90.0     gamma = 90.0
core = Cu_1     edge = K        rmax = 5.0
atoms
! elem  x          y          z          tag
Cu      0.00000    0.00000    0.00000    Cu_1
Cu      0.00000    0.50000    0.50000    Cu_2
Cu      0.50000    0.00000    0.50000    Cu_3
```

```
Cu      0.50000      0.50000      0.00000  Cu_4
```

Note that lines beginning with the `<com>` token are treated as comments and are ignored. Normal text is passed through unaltered and tokens are replaced by appropriate values. The line after the `<list>` token is treated as the format for the list which follows.

Spaces and tabs in atp files are treated like any other text. That is, if the atp parser finds spaces in the atp file, it will write spaces to the output file. If the parser finds tabs, it will write tabs.

The parser is not very smart about knowing which tokens are reasonable for which kinds of output. For example, the `<rmin>` token is relevant to DAFS output but not for other kinds of output. Irrelevant tokens are not guaranteed to possess sensible values. It is thus the responsibility of the atp file author to write a sensible atp file. This is particularly true for atp files used for molecule output. Several tokens relevant to a crystal cell may translate to peculiar values for a molecule.

B.2.3 Types of ATP Tokens

There are five categories of tokens in the atp syntax.

1. Type 1 tokens are simply replaced by their appropriate values. These tokens never take arguments and must have no white space between the tagword and the angle brackets. If a type 1 token appears among normal text, then, in the output, the normal text will appear with the substituted text inserted for the token.
2. Type 2 tokens are replaced by a multi-line list. The following line is used to specify the format of the rows in the list. Type 2 tokens usually require arguments.
3. Type 3 tokens specify list elements. In the context of a list, they behave like type 1 tokens. That is, they are simply replaced by their substitutions. Outside the context of a list, type 3 tokens are treated as normal text, i.e. they are not substituted (unless there is a type 1 token of the same name – `<a>` is an example of this).
4. Type 4 tokens are replaced by multi-line blocks of text. Type 4 tagwords should be paced on lines by themselves and so should not be used among normal text in the manner of type 1 tokens.
5. Type 5 tokens are not translated into actual text. These are either comments or specify meta-data used by the atp parser.

B.2.4 Complete List of ATP Tokens

```
<central> ..... type 1
Replaced by the two letter element symbol of the central atom.
```

<ctag>	type 1
Replaced by the tag of the central atom.		
<edge>	type 1
Replaced by the alphanumeric symbol of the chosen absorption edge.		
<iedge>	type 1
Replaced by the index used by FEFF to denote the absorption edge.		
<eedge>	type 1
Replaced by the energy of the absorption edge in eV.		
<redge>	type 1
Replaced by the energy of the absorption edge in Rydbergs.		
<nclus>	type 1
Replaced by the number of atoms in the output atom list.		
<rmax>	type 1
Replaced by specified radial size of the output cluster.		
<rnn>	type 1
Replaced by the distance to the nearest neighbor atom.		
<rss>	type 1
Replaced by the radius of the <i>small sphere</i> . This is intended to be a modest distance suitable for calculating the self-consistent potentials in FEFF8. It is a multiplier times <rnn>. The value of the multiplier is the value of the :sphere argument to the <meta> token and defaults to 2.2.		
<a>	type 1
Replaced by the <i>a</i> lattice constant in Ångströms.		
	type 1
Replaced by the <i>b</i> lattice constant in Ångströms.		
<c>	type 1
Replaced by the <i>c</i> lattice constant in Ångströms.		

<alpha>	<i>type 1</i>
Replaced by the α lattice angle in degrees.	
<beta>	<i>type 1</i>
Replaced by the β lattice angle in degrees.	
<gamma>	<i>type 1</i>
Replaced by the γ lattice angle in degrees.	
<space>	<i>type 1</i>
Replaced by the canonicalized space group symbol.	
<given>	<i>type 1</i>
Replaced by the space group symbol as specified by the user.	
<class>	<i>type 1</i>
Replaced by crystal class of the specified space group.	
<gnclass>	<i>type 1</i>
Replaced by one letter symbol for the crystal class of the specified space group as used by GNXAS.	
<reflection>	<i>type 1</i>
Replaced by the Miller indices of the currently selected reflection, enclosed in parentheses.	
<dspacing>	<i>type 1</i>
Replaced by d-spacing of the crystal at the currently selected reflection.	
<emin>	<i>type 1</i>
Replaced by the lower bound of the energy grid in a DAFS simulation.	
<emax>	<i>type 1</i>
Replaced by the upper bound of the energy grid in a DAFS simulation.	
<estep>	<i>type 1</i>
Replaced by the spacing of the energy grid in a DAFS simulation.	
<n>	<i>type 1</i>
Replaced by a end of line character(s) appropriate to the operating system. This can also be used in atom lists (but not potential lists).	

<nabs> *type 1*
 Replaced by the number of unique sites occupied by the absorbing atom.

<abslist> *type 1*
 Replaced by a coma separated list identifying the sites occupied by the absorber species. This is useful for GNXAS output.

<gnid> *type 1*
 Replaced by a one-word identifier as needed in the second line of the GNXAS CRYMOL input data. This is the first word of the first title line. That is a lame way of doing it, but...

<fxc> *type 1*
 Replaced by the formula for the *x* component of the central atom.

<fyc> *type 1*
 Replaced by the formula for the *y* component of the central atom.

<fzc> *type 1*
 Replaced by the formula for the *z* component of the central atom.

<list> *type 2*
 Replaced by an atom list which uses the following line to format the rows of the list. **<code>** can take one argument, **:style**, which can be one of **cluster**, **neutral**, **unit**, or **overfull**. If you use an **overfull** list, then you can also use the argument **:margin** which is used to define nearness to the wall, edge, or corner of the unit cell. It defaults to 0.1. The list can contain any of **<x>**, **<y>**, **<z>**, **<r>**, **<ipot>**, **<znum>**, **<tag>**, **<itag>**, **<elem>**, **<inc>**, **<lbou>**, or **<n>**. Some other possible list items are **<valence>**, **<color>**, **<bx>**, **<by>**, **<bz>**, and **<file>**. These last six are various site attributes which will be used in future versions of ATOMS.

<potentials> *type 2*
 Replaced by the potentials list required in a FEFF input file, using the following line as the format for the list. The main argument is **:ipots** which is used to specify the assignment of unique potentials and which takes one of **species**, **tags**, or **sites**. The default is **species**. The list can contain any of **<ipot>**, **<znum>**, **<elem>**, **<l>**, or **<stoi>**. Note that the potentials list must come before the atoms list in an atp file. FEFF does not require this, but ATOMS does. This list only serves a purpose in FEFF input files. The argument **:gcd** can take 0 or 1 and tells ATOMS whether to divide a greatest common denominator out of the stoichiometry column. If the argument **:display** is 0, ATOMS will compute the potential indeces but not display the potentials list in the output file. This is necessary if the atoms list uses the potential indeces even if the potentials list is not displayed.

<dafs>	<i>type 2</i>
Replaced by the list from a DAFS simulation. It takes no arguments. The list can contain any of <e>, <ascr>, <a>, <r>, <i>, <p>, <inc>. This really must only be used with DAFS simulation output. If given cluster data, this will produce very strange output.	
<x>	<i>type 3</i>
Replaced by the x coordinate of the atom.	
<y>	<i>type 3</i>
Replaced by the y coordinate of the atom.	
<z>	<i>type 3</i>
Replaced by the z coordinate of the atom.	
<r>	<i>type 3</i>
In the context of a cluster, this is replaced by the radial distance from the central atom. In a DAFS list, it is replaced by the real part of a DAFS simulation (when in a DAFS list).	
<ipot>	<i>type 3</i>
Replaced by the potential index of the site.	
<stoi>	<i>type 3</i>
In the context of a potential list, this is replaced by the number of atoms of this potential index in the unit cell. In the context of a atom cluster list this is replaced by number of atoms in the unit cell of that unique potential. This information is used by FEFF8 to compute the Fermi energy and the self-consistent potentials. If the :gcd argument to the <potentials> token is set to 1, then the stoichiometry numbers will be reduced by a factor of their greatest common denominator.	
<znum>	<i>type 3</i>
Replaced by the elemental number of the site.	
<tag>	<i>type 3</i>
Replaced by the tag associated with the site.	
<utag>	<i>type 3</i>
Replaced by the unique tag associated with the site. In general these are the same as the tags, but If the user assigns tags that are not unique, ATOMS will append some characters to assure that every site has an individual utag. The utags are used when constructing site coordinate formulas.	

<itag>	<i>type 3</i>
Replaced by the tag associated with the site plus its cluster index.	
<elem>	<i>type 3</i>
Replaced by the two letter element symbol of the site.	
<inc>	<i>type 3</i>
Replaced by an incrementing index.	
<1bou>	<i>type 3</i>
Replaced by the one bounce flag. This is only relevant for ‘geom.dat’ files.	
<l>	<i>type 3</i>
Replaced by an angular momentum value appropriate to the element. This is 1 for Ne and lighter, 2 for Kr and lighter, and 3 for the rest of the periodic table. This is relevant for potential lists in FEFF8.	
<e>	<i>type 3</i>
Replaced by the energy in a DAFS simulation.	
<ascr>	<i>type 3</i>
Replaced by the amplitude squared of a DAFS simulation.	
<a>	<i>type 3</i>
Replaced by the amplitude of a DAFS simulation.	
<i>	<i>type 3</i>
Replaced by the imaginary part of a DAFS simulation.	
<p>	<i>type 3</i>
Replaced by the phase of a DAFS simulation.	
<valence>	<i>type 3</i>
Replaced by the formal valence of the atoms occupying a site.	
<color>	<i>type 3</i>
Replaced by the color assigned to a site.	
<fx>	<i>type 3</i>
Replaced by the formula for the x component of a site.	

<fy>	<i>type 3</i>
Replaced by the formula for the y component of a site.	
<fz>	<i>type 3</i>
Replaced by the formula for the z component of a site.	
<bx>	<i>type 3</i>
Replaced by the x component of the thermal ellipsoid.	
<by>	<i>type 3</i>
Replaced by the y component of the thermal ellipsoid.	
<bz>	<i>type 3</i>
Replaced by the z component of the thermal ellipsoid.	
<file>	<i>type 3</i>
Replaced by the name of the auxiliary file for a site.	
<id>	<i>type 4</i>
This writes out some information identifying the version of Atoms that generated the output file. The only argument is :prefix which specifies the string that is inserted before the id line (for instance, an appropriate comment character). If the prefix contains spaces, use double quotes to delimit it.	
<titles>	<i>type 4</i>
Replaced by the user-supplied title lines. The only argument is :prefix which specifies the string that is inserted before the id line (for instance, an appropriate comment character). If the prefix contains spaces, use double quotes to delimit it.	
<corrections>	<i>type 4</i>
Replaced by the results of the calculations using the McMaster tables. One optional argument is :prefix which specifies the string that is inserted before the id line (for instance an appropriate comment character). If the prefix contains spaces, use double quotes to delimit it. The other argument is :prettyline . If this is 1, then the corrections will be separated from the surrounding text by a line of dashes and asterisks. These lines are omitted if the value is 0.	
<shift>	<i>type 4</i>
Replaced by a line specifying the shift vector. This is a type 4 rather than type 1 token so that it can be omitted if the shift vector is (0,0,0). One optional argument is :prefix which specifies the string that is inserted before the word “shift” (for instance an appropriate comment character).	

<fuse> type 4
 Replaced by lines used by the **FEFF/UWXAFS System for Emacs**. The **:file** argument takes the type of the output file. Typically this will be either **feff** or **atoms**.

<resource> type 4
 Replaced by a line identifying the absorption data resource in use.

<gncell> type 4
 Replaced by lines describing the cell axes and labels in the form needed for the GNXAS CRYMOL input data.

<meta> type 5
 This line will not be passed to the output file, but it contains information relevant to the construction of the output file. The arguments are **:output** which provides the default output file name; **:precision** indicates the precision of the output floating point numbers; **:file** which specifies a character string describing the current output file; **:incbegin** which is the first number which used by the **<inc>** token; **:list** which tells the parser what kind of atom list will be used by the **<list>** token; **<:feff>** which takes the version number of FEFF that this atp file is appropriate for or 0 otherwise; **:sphere** which takes the multiplier used to compute **<rss>**; **:gnxas** which is 1 only for output files intended to be used with the GNXAS CRYMOL program; and **:occupancy** which takes 1 if the atoms list in the output file should display dopants and 0 if it should not. The **:list** argument is needed because the atp parser is a one-pass filter. Some tokens that may appear before the **<list>** token will need to know what kind of list will be used later in the file. Setting **:occupancy** to 0 is necessary for a '**feff.inp**' file, but 1 is used to save dopant information in the output file.

<com> type 5
 Anything following this token on a line will be ignored by the parser.

B.2.5 Lists in ATP Files

Atoms Lists

Atom lists are the *raison d'être* of ATOMS. Consequently ATOMS offers several kinds of atoms lists which are specified in the atp file by the **<list>** token. The **:style** argument is used to select the list type. The list types are

cluster

A spherical cluster of a specified radius centered at a chosen atom. The central atom is placed at the origin and all positions are given in Cartesian coordinates. This is the type of list used in FEFF input files.

unit The contents of the unit cell given in fractional coordinates. Every atom position is

moved to the first octant. This kind of list is appropriate for use with the $P\ 1$ space group.

overfull

The contents of the unit cell plus several other atoms given in Cartesian coordinates. This is often useful for ball-and-stick plots. Any atoms in the unit cell which are close to a wall, edge, or corner of the unit cell are copied to the opposite side(s) of the unit cell. The meaning of “close” is determined by the `:overfull` argument to the `<list>` token, which defaults to 0.1. This means that any atom within this number times the length of the **a** lattice constant away from a wall, edge, or corner in the x direction will be copied to the other side of the unit cell. Likewise for the other two directions. In the case of, for example, a perovskite, this results in a cluster containing eight corner cations, six wall anions, and a cation at the cell center.

symmetry

This is actually not a list of atom positions. Instead it is a list of math expressions denoting the symmetry operations specific to the space group which are used to generate atom positions from the unique positions. See Figure 3.2.

atoms The list of unique atom positions. This is principally used by $\mathcal{T}\mathcal{K}$ ATOMS and *Web*ATOMS for saving crystallographic data.

The line following the `<list>` token specifies the format of the list. The list can contain any the following tokens `<x>`, `<y>`, `<z>`, `<r>`, `<ipot>`, `<znum>`, `<tag>`, `<itag>`, `<elem>`, `<inc>`, or `<ibou>`. Note that FEFF requires that the first four columns be `<x>`, `<y>`, `<z>`, and `<ipot>`.

Potentials List

In the FEFF input file, the assignment of unique potentials is made in the list following the POTENTIALS card. This list is made using the `<potentials>` token in the atp file.

The `<potentials>` token takes one argument, `:ipots`. The following line in the atp file is the format of the potentials list. The list consists of any of the following tokens: `<ipot>`, `<znum>`, `<elem>`, `<l>`, or `<stoi>`. Note that FEFF6 only requires `<ipot>` and `<znum>`. `<stoi>` is optional but highly recommended in the FEFF8 input file.

The `:ipots` argument can take one of three values, **species**, **tags** or **sites**. These indicate how ATOMS should assign unique potentials from the contents of the unit cell. **species** means that each atomic species should receive its own unique potential. **tags** means that each unique atom tag should receive its own potential. **sites** means that each unique crystallographic site should receive its own potential.

Here’s an example. With this input file:

```
title YBCO: Y Ba2 Cu3 O7
space P M M M
rmax=5.2          a 3.823    b 3.886 c 11.681
```



```

core=cu2
atom
  Y      0.5      0.5      0.5
  Ba     0.5      0.5      0.184
  Cu      0        0        0      cu1
  Cu      0        0      0.356    cu2
  O        0      0.5      0        o
  O        0        0      0.158    o
  O        0      0.5      0.379    o
  O      0.5      0      0.377    o

```

Using the `species` option results in this potentials list:

```

POTENTIALS
*   ipot   Z  element
    0   29   Cu
    1   39   Y
    2   56   Ba
    3   29   Cu
    4    8    O

```

Using the `tags` option results in this potentials list:

```

POTENTIALS
*   ipot   Z  element
    0   29   Cu
    1   39   Y
    2   56   Ba
    3   29  cu1
    4   29  cu2
    5    8    o

```

Using the `sites` option results in this potentials list:

```

POTENTIALS
*   ipot   Z  element
    0   29   Cu
    1   39   Y
    2   56   Ba
    3   29  cu1
    4   29  cu2
    5    8    o
    6    8    o
    7    8    o
    8    8    o

```

Note that the `sites` option is a poor choice for this material. FEFF only allows 7 unique potentials other than the central atom. In this case the `tags` option combined with a judicious choice of site tags in the input file would be more appropriate.

DAFS List

This list is used to format the output of a DAFS simulation. The list can present any of the amplitude, amplitude squared, real part, imaginary part, and phase of the calculation as a function of energy. You can also count data points with the `<inc>` token.

If you use an `atp` file with the `<dafs>` list with cluster data from the ATOMS cluster expansion or any any other kind of atoms list data, you will get very strange results. There are some heuristics built into ATOMS to try to avoid that situation, but the best solution is just not to do it!

B.3 Editing Template Files with Emacs

In the `etc/` directory of the ATOMS distribution you will find a file called `atp.el`. This is a program for use with Emacs which defines an editing mode for creating and maintaining `atp` files.

To use `atp.el`, move it to a location where Emacs looks for lisp programs. For system-wide use, that would be the `site-lisp/` directory under the Emacs installation. For personal use, put in the place where you keep your own lisp programs. Once installed, byte-compile it. This can be done by this command at the command line:

```
emacs -batch -q -f batch-byte-compile atp.el
```

or by editing the file and doing `M-x byte-compile-file`.

You will need to put some lines in the system `site-start.el` file or in your own `.emacs` file to enable the use of `atp-mode` in Emacs:

```
(setq load-path (append (list ("\\.atp$" . atp-mode)) load-path))
(autoload 'atp-mode "atp" "Atoms template mode." t)
```

With these lines, any file with the `.atp` extension will be loaded into `atp-mode`.

Several features are offered by `atp-mode` which aid in making `atp` files which work well. Tagwords and arguments are colored by `atp-mode` using font-lock. This is useful as a sort of spell-checker. If tokens are not colored, then they are misspelled. `Atp-mode` also supplies completion and description functions for keywords and arguments. Here is a list of all the key-bindings in `atp-mode`:

The syntax checker invoked by `C-c C-c` is pretty simple minded. It checks that all words within angle brackets are actually tagwords, that tokens which take arguments have the allowed arguments, and that the argument values are sensible. The jump function moves point to the

Table B.2: Keybindings in atp-mode in Emacs

key	description
M-tab	Complete word at point.
M-?	Write a one-line description of the word at point in the echo area.
C-c C-k	Display all tagwords with short descriptions in a help buffer.
C-c C-c	Check the syntax of the current file.
C-c C-n	Jump to the next error found by the syntax checker.

beginning of the line containing the next error found by the syntax checker. I intend to keep the syntax checker up to date as tokens are added to the parser.

Appendix C

Space Group Symbols

This Appendix contains a big table of all the space group symbols recognized by ATOMS. The *index* of each space group denotes the order in which the groups appear in the International Tables of Crystallography and the *standard* symbol is the short Hermann–Maguin symbol in the standard setting of the group. The column labeled Schoenflies gives the Schoenflies symbol for each group in a form adapted to the keyboard. In the literature, you will find these symbols typeset. For example, space group #16, $P\ 2\ 2\ 2$, has Schoenflies symbol d_2^1 , which is given in Table C.1 as `d_2^1`.

The column labeled *settings* gives all the unique symbols of alternate settings of low symmetry space groups. These alternate settings are described in detail in Section 3.2.4. For monoclinic groups the full symbols are given to help avoid ambiguity and symbols are given for all three spacial settings of all three unique angles. The first three choices are for β unique, the next three for γ unique, and the last three for α unique. For orthorhombic groups, when all the alternate settings have unique symbols, they are in the order `abc`, `ba-c`, `cab`, `-cba`, `bca`, `a-cb`. For tetragonal groups, the F and C centered settings are given in this column.

The column labeled *1935/new* contains either the symbol from the 1935 edition of the International Tables when it is different from the standard symbol or the new symbol which uses *e* to denote a double glide plane.

The column labeled *nicknames* contains special strings recognized by ATOMS to denote the space groups of certain very common structures.

Except for the alternate settings of monoclinic groups, the full Hermann–Maguin symbols for the space groups do not appear in this table. If you choose to use full symbols, ATOMS will recognize them.

In print the Hermann–Maguin symbols are usually typeset, but in the table they are presented in a form accessible to the keyboard. Negated numbers such as -4 are usually typeset as $\bar{4}$, two digit numbers such as 21 as 2_1 , and two digit numbers with slashes such as $21/m$ as $2_1/m$.

Any symbol, including the indices, will be recognized by ATOMS when given as input data.

Table C.1: Space group symbols recognized by ATOMS

index	standard	schoenflies	settings	1935/new	nicknames
Triclinic groups					
1	P 1	c_1^1			
2	P -1	c_i^1			
Monoclinic groups					
3	P 2	c_2^1	p 1 2 1 p 1 1 2 p 2 1 1		
4	P 21	c_2^2	p 1 21 1 p 1 1 21 p 21 1 1		
5	C 2	c_2^3	c 1 2 1 a 1 2 1 i 1 2 1 a 1 1 2 b 1 1 2 i 1 1 2 b 2 1 1 i 2 1 1		
6	P m	c_s^1	p 1 m 1 p 1 1 m p m 1 1		
7	P c	c_s^2	p 1 c 1 p 1 n 1 p 1 a 1 p 1 1 a p 1 1 n p 1 1 b p b 1 1 p n 1 1 p c 1 1		
8	C m	c_s^3	c 1 m 1 a 1 m 1 i 1 m 1 a 1 1 m b 1 1 m i 1 1 m b m 1 1 c m 1 1 i m 1 1		

Table C.1: Space group symbols recognized by ATOMS

index	standard	schoenflies	settings	1935/new	nicknames
9	C c	c_s ⁴	c 1 c 1 a 1 n 1 i 1 a 1 a 1 1 a b 1 1 n i 1 1 b b b 1 1 c n 1 1 i c 1 1		
10	P 2/m	c_2h ¹	p 1 2/m 1 p 1 1 2/m p 2/m 1 1		
11	P 21/m	c_2h ²	p 1 21/m 1 p 1 1 21/m p 21/m 1 1		
12	C 2/m	c_2h ³	c 1 2/m 1 a 1 2/m 1 i 1 2/m 1 a 1 1 2/m b 1 1 2/m i 1 1 2/m b 2/m 1 1 c 2/m 1 1 i 2/m 1 1		
13	P 2/c	c_2h ⁴	p 1 2/c 1 p 1 2/n 1 p 1 2/a 1 p 1 1 2/a p 1 1 2/n p 1 1 2/b p 2/a 1 1 p 2/n 1 1 p 2/c 1 1		

Table C.1: Space group symbols recognized by ATOMS

index	standard	schoenflies	settings	1935/new	nicknames
14	P 21/c	c_2h^5	p 1 21/c 1 p 1 21/n 1 p 1 21/a 1 p 1 1 21/a p 1 1 21/n p 1 1 21/b p 21/a 1 1 p 21/n 1 1 p 21/c 1 1		
15	C 2/c	c_2h^4	c 1 2/c 1 a 1 2/n 1 i 1 2/a 1 a 1 1 2/a b 1 1 2/n i 1 1 2/b b 2/a 1 1 c 2/n 1 1 i 2/c 1 1		

Orthorhombic groups

16	P 2 2 2	d_2^1			
17	P 2 2 21	d_2^2	p 21 2 2 p 2 21 2		
18	P 21 21 2	d_2^3	p 2 21 21 p 21 2 21		
19	P 21 21 21	d_2^4			
20	C 2 2 21	d_2^5	a 21 2 2 b 2 21 2		
21	C 2 2 2	d_2^6	a 2 2 2 b 2 2 2		
22	F 2 2 2	d_2^7			
23	I 2 2 2	d_2^8			
24	I 21 21 21	d_2^9			
25	P m m 2	c_2v^1	p 2 m m p m 2 m		
26	P m c 21	c_2v^2	p c m 21 p 21 m a p 21 a m p b 21 m p m 21 b		

Table C.1: Space group symbols recognized by ATOMS

index	standard	schoenflies	settings	1935/new	nicknames
27	P c c 2	c _{2v} ³	p 2 a a p b 2 b		
28	P m a 2	c _{2v} ⁴	p b m 2 p 2 m b p 2 c m p m 2 a		
29	P c a 21	c _{2v} ⁵	p b c 21 p 21 a b p 21 c a p c 21 b p b 21 a		
30	P n c 2	c _{2v} ⁶	p c n 2 p 2 n a p 2 a n p b 2 n p n 2 b		
31	P m n 21	c _{2v} ⁷	p n m 21 p 21 m n p 21 n m p n 21 m p m 21 n		
32	P b a 2	c _{2v} ⁸	p 2 c b p c 2 a		
33	P n a 21	c _{2v} ⁹	p b n 21 p 21 n b p 21 c n p c 21 n p n 21 a		
34	P n n 2	c _{2v} ¹⁰	p 2 n n p n 2 n		
35	C m m 2	c _{2v} ¹¹	a 2 m m b m 2 m		
36	C m c 21	c _{2v} ¹²	c c m 21 a 21 m a a 21 a m b b 21 m b m 21 b		
37	C c c 2	c _{2v} ¹³	a 2 a a b b 2 b		

Table C.1: Space group symbols recognized by ATOMS

index	standard	schoenflies	settings	1935/new	nicknames
38	A m m 2	c _{2v} ¹⁴	b m m 2 b 2 m m c 2 m m c m 2 m a m 2 m		
39	A b m 2	c _{2v} ¹⁵	b m a 2 b 2 c m c 2 m b c m 2 a a c 2 m	a e m 2	
40	A m a 2	c _{2v} ¹⁶	b b m 2 b 2 m b c 2 c m c c m 2 a m 2 a		
41	A b a 2	c _{2v} ¹⁷	b b a 2 b 2 c b c 2 c b c c 2 a a c 2 a	a e a 2	
42	F m m 2	c _{2v} ¹⁸	f 2 m m f m 2 m		
43	F d d 2	c _{2v} ¹⁹	f 2 d d f d 2 d		
44	I m m 2	c _{2v} ²⁰	i 2 m m i m 2 m		
45	I b a 2	c _{2v} ²¹	i 2 c b i c 2 a		
46	I m a 2	c _{2v} ²²	i b m 2 i 2 m b i 2 c m i c 2 m i m 2 a		
47	P m m m	d _{2h} ¹			
48	P n n n	d _{2h} ²			
49	P c c m	d _{2h} ³	p m a a p b m b		
50	P b a n	d _{2h} ⁴	p n c b p c n a		

Table C.1: Space group symbols recognized by ATOMS

index	standard	schoenflies	settings	1935/new	nicknames
51	P m m a	d_2h^5	p m m b p b m m p c m m p m c m p m a m		
52	P n n a	d_2h^6	p n n b p b n n p c n n p n c n p n a n		
53	P m n a	d_2h^7	p n m b p b m n p c n m p n c m p m a n		
54	P c c a	d_2h^8	p c c b p b a a p c a a p b c b p b a b		
55	P b a m	d_2h^9	p m c b p c m a		
56	P c c n	d_2h^10	p n a a p b n b p b n a		
57	P b c m	d_2h^11	p c a m p m c a p m a b p b m a p c m b		
58	P n n m	d_2h^12	p m n n p n m n		
59	P m m n	d_2h^13	p n m m p m n m		
60	P b c n	d_2h^14	p c a n p n c a p n a b p b n a p c n b		
61	P b c a	d_2h^15	p c a b		

Table C.1: Space group symbols recognized by ATOMS

index	standard	schoenflies	settings	1935/new	nicknames
62	P n m a	d_2h ¹⁶	p m n b p b n m p c m n p m c n p n a m		
63	C m c m	d_2h ¹⁷	c c m m a m m a a m a m b b m m b m m b		
64	C m c a	d_2h ¹⁸	c c m b a b m a a c a m b b c m b m a b	c m c e	
65	C m m m	d_2h ¹⁹	a m m m b m m m		
66	C c c m	d_2h ²⁰	a m a a b b m b		
67	C m m a	d_2h ²¹	c m m b a b m m a c m m b m c m b m a m	c m m e	
68	C c c a	d_2h ²²	c c c b a b a a a c a a b b c b b b a b	c c c e	
69	F m m m	d_2h ²³			
70	F d d d	d_2h ²⁴			
71	I m m m	d_2h ²⁵			
72	I b a m	d_2h ²⁶	i m c b i c m a		
73	I b c a	d_2h ²⁷	i c a b		
74	I m m a	d_2h ²⁸	i m m b i b m m i c m m i m c m i m a m		

Table C.1: Space group symbols recognized by ATOMS

index	standard	schoenflies	settings	1935/new	nicknames
Tetragonal groups					
75	P 4	c ₄ ¹	c 4		
76	P 41	c ₄ ²	c 41		
77	P 42	c ₄ ³	c 42		
78	P 43	c ₄ ⁴	c 43		
79	I 4	c ₄ ⁵	f 4		
80	I 41	c ₄ ⁶	f 41		
81	P -4	s ₄ ¹	c -4		
82	I -4	s ₄ ²	f -4		
83	P 4/m	c _{4h} ¹	c 4/m		
84	P 42/m	c _{4h} ²	c 42/m		
85	P 4/n	c _{4h} ³	c 4/a		
86	P 42/n	c _{4h} ⁴	c 42/a		
87	I 4/m	c _{4h} ⁵	f 4/m		
88	I 41/a	c _{4h} ⁶	f 41/d		
89	P 4 2 2	d ₄ ¹	c 4 2 2	p 4 2	
90	P 4 21 2	d ₄ ²	c 4 21 2	p 4 21	
91	P 41 2 2	d ₄ ³	c 41 2 2	p 41 2	
92	P 41 21 2	d ₄ ⁴	c 41 2 21	p 41 21	
93	P 42 2 2	d ₄ ⁵	c 42 2 2	p 42 2	
94	P 42 21 2	d ₄ ⁶	c 42 2 21	p 42 21	
95	P 43 2 2	d ₄ ⁷	c 43 2 2	p 43 2	
96	P 43 21 2	d ₄ ⁸	c 43 2 21	p 43 21	
97	I 4 2 2	d ₄ ⁹	f 4 2 2	i 4 2	
98	I 41 2 2	d ₄ ¹⁰	f 41 2 2	p 41 2	
99	P 4 m m	c _{4v} ¹	c 4 m m		
100	P 4 b m	c _{4v} ²	c 4 m g1		
101	P 42 c m	d _{4v} ³	c 42 m c		
102	P 42 n m	c _{4v} ⁴	c 42 m g2		
103	P 4 c c	c _{4v} ⁵	c 4 c c		
104	P 4 n c	p _{4v} ⁶	c 4 c g2		
105	P 42 m c	c _{4v} ⁷	c 42 c m		
106	P 42 b c	c _{4v} ⁸	c 42 c g1		
107	I 4 m m	c _{4v} ⁹	f 4 m m		
108	I 4 c m	c _{4v} ¹⁰	f 4 m c		
109	I 41 m d	c _{4v} ¹¹	f 41 d m		
110	I 41 c d	c _{4v} ¹²	f 41 d c		
111	P -4 2 m	d _{2d} ¹	c -4 m 2		
112	P -4 2 c	d _{2d} ²	c -4 c 2		
113	P -4 21 m	d _{2d} ³	c -4 m 21		

Table C.1: Space group symbols recognized by ATOMS

index	standard	schoenflies	settings	1935/new	nicknames
114	P -4 21 c	d ₂ d ⁴	c -4 c 21		
115	P -4 m 2	d ₂ d ⁵	c -4 2 m		
116	P -4 c 2	d ₂ d ⁶	c -4 2 c		
117	P -4 b 2	d ₂ d ⁷	c -4 2 g		
118	P -4 n 2	d ₂ d ⁸	c -4 2 g		
119	I -4 m 2	d ₂ d ⁹	f -4 2 m		
120	I -4 c 2	d ₂ d ¹⁰	f -4 2 c		
121	I -4 2 m	d ₂ d ¹¹	f -4 m 2		
122	I -4 2 d	d ₂ d ¹²	f -4 d 2		
123	P 4/m m m	d ₄ h ¹	c 4/m m m		
124	P 4/m c c	d ₄ h ²	c 4/m c c		
125	P 4/n b m	d ₄ h ³	c 4/a m b		
126	P 4/n n c	d ₄ h ⁴	c 4/a c n		
127	P 4/m b m	d ₄ h ⁵	c 4/m m b		
128	P 4/m n c	d ₄ h ⁶	c 4/m c n		
129	P 4/n m m	d ₄ h ⁷	c 4/a m m		
130	P 4/n c c	d ₄ h ⁸	c 4/a c c		
131	P 42/m m c	d ₄ h ⁹	c 42/m c m		
132	P 42/m c m	d ₄ h ¹⁰	c 42/m m c		
133	P 42/n b c	d ₄ h ¹¹	c 42/a c b		
134	P 42/n n m	d ₄ h ¹²	c 42/a m n		
135	P 42/m b c	d ₄ h ¹³	c 42/m c b		
136	P 42/m n m	d ₄ h ¹⁴	c 42/m m n		
137	P 42/n m c	d ₄ h ¹⁵	c 42/a c m		
138	P 42/n c m	d ₄ h ¹⁶	c 42/a m c		
139	I 4/m m m	d ₄ h ¹⁷	f 4/m m m		
140	I 4/m c m	d ₄ h ¹⁸	f 4/m m c		
141	I 41/a m d	d ₄ h ¹⁹	f 41/d d m		
142	I 41/a c d	d ₄ h ²⁰	f 41/d d c		

Trigonal groups

143	P 3	c ₃ ¹	
144	P 31	c ₃ ²	
145	P 32	c ₃ ³	
146	R 3	c ₃ ⁴	
147	P -3	c ₃ i ¹	
148	R -3	c ₃ i ²	
149	P 3 1 2	d ₃ ¹	
150	P 3 2 1	d ₃ ²	
151	P 31 1 2	d ₃ ³	
152	P 31 2 1	d ₃ ⁴	

Table C.1: Space group symbols recognized by ATOMS

index	standard	schoenflies	settings	1935/new	nicknames
153	P 32 1 2	d ₃ ⁵			
154	P 32 2 1	d ₃ ⁶			
155	R 3 2	d ₃ ⁷			
156	P 3 m 1	c _{3v} ¹			
157	P 3 1 m	c _{3v} ²			
158	P 3 c 1	c _{3v} ³			
159	P 3 1 c	c _{3v} ⁴			
160	R 3 m	c _{3v} ⁵			
161	R 3 c	c _{3v} ⁶			
162	P -3 1 m	d _{3d} ¹			
163	P -3 1 c	d _{3d} ²			
164	P -3 m 1	d _{3d} ³			
165	P -3 c 1	d _{3d} ⁴			
166	R -3 m	d _{3d} ⁵			
167	R -3 c	d _{3d} ⁶			

Hexagonal groups

168	P 6	c ₆ ¹			
169	P 61	c ₆ ²			
170	P 65	c ₆ ³			
171	P 62	c ₆ ⁴			
172	P 64	c ₆ ⁵			
173	P 63	c ₆ ⁶			
174	P -6	c _{3h} ¹			
175	P 6/m	c _{6h} ¹			
176	P 63/m	c _{6h} ²			
177	P 6 2 2	d ₆ ¹			
178	P 61 2 2	d ₆ ²			
179	P 65 2 2	d ₆ ³			
180	P 62 2 2	d ₆ ⁴			
181	P 64 2 2	d ₆ ⁵			
182	P 63 2 2	d ₆ ⁶			
183	P 6 m m	c _{6v} ¹			
184	P 6 c c	c _{6v} ²			
185	P 63 c m	c _{6v} ³			
186	P 63 m c	c _{6v} ⁴			graphite gra
187	P -6 m 2	d _{3h} ¹			
188	P -6 c 2	d _{3h} ²			
189	P -6 2 m	d _{3h} ³			
190	P -6 2 c	d _{3h} ⁴			

Table C.1: Space group symbols recognized by ATOMS

index	standard	schoenflies	settings	1935/new	nicknames
191	P 6/m m m	d _{6h} ¹			
192	P 6/m c c	d _{6h} ²			
193	P 63/m c m	d _{6h} ³			
194	P 63/m m c	d _{6h} ⁴			hex hcp

Cubic groups

195	P 2 3	t ¹			
196	F 2 3	t ²			
197	I 2 3	t ³			
198	P 21 3	t ⁴			
199	I 21 3	t ⁵			
200	P m -3	t _h ¹		p m 3	
201	P n -3	t _h ³		p n 3	
202	F m -3	t _h ³		f m 3	
203	F d -3	t _h ⁴		f d 3	
204	I m -3	t _h ⁵		i m 3	
205	P a -3	t _h ⁶		p a 3	
206	I a -3	t _h ⁷		i a 3	
207	P 4 3 2	o ¹		p 4 3	
208	P 42 3 2	o ²		p 42 3	
209	F 4 3 2	o ³		f 4 3	
210	F 41 3 2	o ⁴		f 41 3	
211	I 4 3 2	o ⁵		i 4 3	
212	P 43 3 2	o ⁶		p 43 3	
213	P 41 3 2	o ⁷		p 41 3	
214	I 41 3 2	o ⁸		i 41 3	
215	P -4 3 m	t _d ¹			
216	F -4 3 m	t _d ²			zincblende zns
217	I -4 3 m	t _d ³			
218	P -4 3 n	t _d ⁴			
219	F -4 3 c	t _d ⁵			
220	I -4 3 d	t _d ⁶			
221	P m -3 m	o _h ¹		p m 3 m	cubic cscl perov perovskite
222	P n -3 n	o _h ²		p n 3 n	
223	P m -3 n	o _h ³		p m 3 n	
224	P n -3 m	o _h ⁴		p n 3 m	

Table C.1: Space group symbols recognized by ATOMS

index	standard	schoenflies	settings	1935/new	nicknames
225	F m -3 m	o _h ⁵		f m 3 m	fcc salt nacl
226	F m -3 c	o _h ⁶		f m 3 c	
227	F d -3 m	o _h ⁷		f d 3 m	diamond
228	F d -3 c	o _h ⁸		f d 3 c	
229	I m -3 m	o _h ⁹		i m 3 m	bcc
230	I a -3 d	o _h ¹⁰		i a 3 d	

Appendix D

Perl Modules in the ATOMS package

This appendix reproduces the documentation contained in the perl modules that come with the ATOMS distributions. This appendix is mostly useful to the person interested modifying some part of ATOMS or in using these modules to write new programs.

D.1 ATOMS.PM

NAME

Xray::Atoms — Utilities and data structures for the Atoms program

SYNOPSIS

```
use Xray::Xtal;
use Xray::Atoms qw(parse_input keyword_defaults parse_atp
                    absorption mcmaster);
```

DESCRIPTION

This module contains the utility subroutines used by the program Atoms. It defines a number of exportable routines for performing chores specific to Atoms. It also defines an object for storing input data to the various programs which use this file. For details about the crystallographic objects used by Atoms, see the *Xray::Xtal* manpage.

There are several versions of Atoms with different user interfaces all of which use routines from this module. The versions include

THE KEYWORDS OBJECT

To simplify the handling of input data, this module provides an object. It is used like this:

```
my $keywords = Xray::Atoms -> new();  
$keywords -> make('rmax' => 5.7)
```

The make method is the general way of storing data in the keywords object. In general, this method assumes that keyword takes a scalar value. The data structure is just a hash, so the keyword can be anything. Several keywords are treated specially because they are commonly used in the programs included in the Atoms package. These keywords

```
a b c alpha beta gamma argon krypton nitrogen rmax
```

are treated as numbers. In the context of this module, that This means that they are eval-ed (and so can be short expressions like $1/2$ or $3.5+0.001$) and return 0 if they cannot be interpreted as numbers.

A couple of keywords are treated specially.

```
$keywords -> make(title=>'This is a title!');
```

pushes the given string onto an anonymous array of strings. In this way, any number of title lines can be kept in the object.

The `sites` keyword is also handled specially. This takes three arguments, the three components of the shift vector. A syntax like this (or any perl-speak equivalent) works fine:

```
$keywords -> make('shift' => $x, $y, $z);
```

This stores the three values of the shift vector as an anonymous array.

METHODS AND EXPORTED SUBROUTINES

`parse_input`

This routine is used by the stand-alone version of atoms to read an atoms input file. It is typically called by

```
$keywords -> parse_input($file, 0);
```

The first argument is the name of the input file. The second argument is tells the error trapping mechnism what kind of environment you are working in. If you have written a command-line program, it should be 0. In a Tk program, it should 1, and in a CGI script it should be 2.

The contenets of the file are parsed and stored in a keywords object. The method is fairly clever about interpreting the file and offering warning and error messages. It uses a Safe compartment to evaluate unique atoms positions, thus allowing simple math expressions to be used in the atom list.

verify_keywords

This method is used to perform some sanity checks on the keyword values after reading in the input data and populating the sites and the cell. It is a good idea to call this before building a cluster or starting some other calculation.

```
$keywords -> verify_keywords(\$cell, \@sites, 0);
```

The first two arguments are to the Cell object and the list of Site objects. The final argument is the same as the second argument to the `parse_input` method.

warn_or_die

This is a subroutine rather than a method, so that it can be used even without an active keywords object. It is a wrapper for generating error and warning methods that will work in command-line, GUI, and CGI environments.

```
&warn_or_die("This is a warning message", $die, $keywords);
```

The first argument is the actual message. The `die` argument is 0, 1, or 2 as described for `parse_input`. The final argument is a keywords object and is only used in the CGI environment. In that case, there are no convenient output or error channels, so messages are stored in the keywords hash as a single string under the key "www_warn".

available_languages

This subroutine takes no arguments and returns a list of the languages for which translations of the language data exist.

parse_atp

This subroutine is the output engine. It parses the appropriate atp file, interprets its content, and generates output data. This one subroutine handles the creation of *feff.inp* and the other sorts of atoms lists generated by *atoms.pl*. It also generates reports for the Absorption notecard in TkAtoms, it generates the lists describing the DAFS calculation as a function of energy and the powder calculation as a function of angle. Essentially all output from all programs that come with the Atoms package is generated here.

```
my ($default_name, $is_feff) =  
  &parse_atp($atp, \$cell, \$keywords, \@cluster, \@neutral, \$contents);  
open (INP, ">".$default_name) or die $!;  
print INP $contents;  
close INP;
```

The arguments are

1. A string specifying the atp file to read. The actual atp file either lives in the *atp/* subdirectory of the Atoms installation or in the *.atoms/* irectory of the individual user.
2. A reference to the Call object.
3. A reference to a keywords object.
4. A reference to the list generated by the `build_cluster` subroutine, to the list containing a calculation such as the one in DAFS or Powder, or 'whatever the thing is that I did to make atoms.inp output work.'
5. A reference to a list. This is just a place holder and is not currently used for anything.
6. A scalar reference. This is assumed to be empty on entry, and is filled with contents intended for the output file on exit.

Typically the `contents` variables is the things written to the output channel. The two return values are the default name for the output file and a flag indicating whether the output is intended to run FEFF. Both of these pieces of information are read from the C<<meta>> line of the atp file.

absorption

This subroutine calculates the total absorption (or one over the e-fold absorption length), the `delta_mu` across the edge (or one over the unit edge step length) and the specific gravity of the crystal.

```
($total_absorption, $delta_mu, $specific_gravity) =
    &absorption($cell, $central, $edge);
```

The input arguments are a reference to a populated cell, the tag of the central atom, and the alphanumeric (i.e. K, L1, etc) symbol of the absorption edge.

Please note that the values returned depend on the data resource used. See the *Xray::Absorption* manpage.

mcmaster

This is called `mcmaster` for historical reasons. It calculates the normalization correcion for a given central atom.

```
$sigma_mm = &mcmaster($central, $edge);
```

It takes the central atoms tag and the alphanumeric edge symbol as arguments and returns the normalization correction in units of Angstrom squared.

Please note that the values returned depend on the data resource used. See the *Xray::Absorption* manpage.

i_zero

This calculates the correction due to the I0 fill gases in a fluorescence experiment.

```
$sigma_i0 = &i_zero($central, $edge, $nitrogen, $argon, $krypton);
```

It takes the central atoms tag, the alphanumeric edge symbol, and the volume percentages of the three gases as arguments. It assumes that any remaining volume is filled with helium and it correctly accounts for the fact that nitrogen is a diatom. It returns the I0 correction in units of Angstrom squared.

Please note that the values returned depend on the data resource used. See the *Xray::Absorption* manpage.

self

This calculates the correction due to self-absorption fluorescence experiment. It assumes that the sample is infinitely thick and that the entry and exit angles of the photons are the same.

```
$sigma_i0 = &self($central, $edge, \%cell);
```

It takes the central atoms tag, the alphanumeric edge symbol, and a reference to a populated cell. It returns a list whose zeroth element is the multiplicative amplitude correction and whose first element is the a correction in units of Angstrom squared.

Please note that the values returned depend on the data resource used. See the *Xray::Absorption* manpage.

build_cluster

This builds a cluster....

rcfile_name

This takes no arguments and returns the name of the Atoms runtime configuration file belonging to the user. This does the “right thing” on the different platforms.

rcvalues

This takes no arguments and returns a list containing the values of variables (but not of fonts or colors) read from the rc file. They are in the following order:

```
$always_write_feff $atoms_language    $write_to_pwd
$prefer_feff_eight $absorption_tables $dafs_default
$plotting_hook     $default_filepath  $display_balloons
$no_crystal_warnings $one_frame
```

New values will always be added to the end of the list to avoid beaking scripts that use this function.

`number`

This takes a text string and attempts to evaluate it as a number. It uses `eval` and so allows for simple math expressions, but it tries to be safe and not eval just any old math expression. You can use this if you want to evaluate numbers in the same manner as `Atoms`.

MORE INFORMATION

There is more information available in the `Atoms` document. There you will find complete descriptions of `atp` files, calculations using the `Xray::Absorption` package, keywords in `atoms` input files and lots of other topics.

AUTHOR

Bruce Ravel <ravel@phys.washington.edu>
Atoms URL: <http://feff.phys.washington.edu/~ravel/atoms/>

D.2 XTAL.PM

NAME

`Xray::Xtal` — A Perl extension for crystallography data classes

SYNOPSIS

```
use Xray::Xtal;

my $cell = Xray::Xtal::Cell -> new();
$cell -> make( Space_group=>"f m -3 m", A=>3.961, );

my @sites;
$sites[0] = Xray::Xtal::Site -> new();
$sites[0] -> make( X=>0.0, Y=>0.0, Z=>0.0, Element=>"Cu" );
```

```
$cell -> populate(\@sites);  
$cell -> verify_cell();
```

The preceding lines define a unit cell for FCC copper.

DESCRIPTION

Xray::Xtal is a module defining packages for unit cell and crystallographic site objects useful in a crystallography problem.

As suggested in the synopsis, these objects are closely related to one another. In fact, the method of translating a unique site to a set of symmetry-related sites requires a Cell object as an argument. Similarly, the method of populating a unit cell requires a list of Site objects as its argument. Consequently it is very unusual to use one of these classes and not the other.

In its current form, Xtal.pm is intended to solve simple crystallography problems of the sort commonly encountered by the X-ray absorption spectroscopist. Given a description of a crystal in terms of a space group symbol, axis lengths and angles, and the fractional coordinates of the unique crystallographic sites, *Xtal.pm* is used to describe the entire contents of the unit cell either in fractional or Cartesian coordinates.

To this end, two packages are included in Xtal.pm which define two data classes. One is a Cell and the other is a Site. Alone, *Xtal.pm* provides only these data classes. This, however, is the basis of several useful applications. The initial use of this module is to create lists of atomic coordinates of the sort needed for ball-and-stick figures or real-space multiple scattering calculations (which is the reason *Xtal.pm* was created). *Xtal.pm* can also be useful, for example, for powder diffraction or anomalous scattering simulations.

Xtal.pm obtains its crystallography data from a database which comes with the distribution. The space groups database is stored to disk using the **Storable** module with portable binary ordering. This choice allows both speed and networked applicability.

The following CPAN modules may not be part of a normal perl installation, but are needed by *Xtal.pm*:

```
Storable, File::Spec, Chemistry::Elements
```

THE CELL AND SITE PACKAGES.

The Cell object

A unit cell is constructed by `$cell=new();>` and the attributes of the unit cell are set using the **make** method. An example of the use of **make** to set the a lattice constant:

```
$cell -> new() -> make( A=4.0 );
```

The attributes of the Cell object are stored internally as capitalized words i.e. words that begin with an upper case letter and have lower case for the remaining letters, however the user is free to mix case in any convenient manner.

The Cell attributes that can be set using **make** are:

A The a lattice constant.

B The b lattice constant.

C The c lattice constant.

Alpha

The angle between the b and c lattice constants.

Beta

The angle between the a and c lattice constants.

Gamma

The angle between the a and b lattice constants.

Angle

This takes the value of the most recently set angle. This is only needed for the peculiar situation of a monoclinic space group with all three angles equal to 90. The function determine monoclinic will not be able to resolve the setting in that situation without a little help. The idea is that the user has to specify at least one angle in order to unambiguously determine the setting.

Space_group

A string specifying the space group of the cell. The supplied value is stored in the **Given_group** attribute and this is filled with the canonical symbol.

Occupancy

This is a boolean. If true (1) then partial occupancy of sites in this cell is allowed. If false (0) then sites may not be doped.

The bare minimum required to define a cell is the a lattice constant and the space group symbol. All other attributes have sensible defaults or are calculated quantities. Of course, any space group of lower than cubic symmetry will require that other axes and/or angles be specified.

There are several other Cell attributes. Except for the Contents attribute, these are updated every time the **make** method is called. These include:

Given_group

The space group symbol used as the argument for the **Space_group** attribute when the **make** method is called.

Setting

The setting of a low symmetry space group. See below for a discussion of low symmetry space groups.

Contents

This is an anonymous list of anonymous lists specifying the contents of the fully decoded unit cell. This attribute is set by calling the `populate` method. Each list element is itself a list containing the x, y, and z fractional coordinates of the site and a reference to the Site object which generated that site. To examine the contents of the cell, do something like this:

```
my ($contents) = $cell -> attributes("Contents");
foreach my $pos (@{$contents}) {
    printf "x=%8.5f, y=%8.5f, z=%8.5f$/",
        $$pos[0], $$pos[1], $$pos[2]
};
```

Volume

The volume of the unit cell computed from the axes and angles.

Txx The x-x element of the metric tensor computed from the axes and angles. This is used to translate from fractional to cartesian coordinates.

Tyx The y-x element of the metric tensor computed from the axes and angles.

Tyz The y-z element of the metric tensor computed from the axes and angles.

Tzx The z-x element of the metric tensor computed from the axes and angles.

Tzz The z-z element of the metric tensor computed from the axes and angles.

other metric tensor elements

The yy element of the metric tensor is unity and the other three are zero.

Xray::Xtal::Cell will only allow you to directly modify the lattice constants and angles and the space group using the `make` method. While it is common for object definitions to allow the user to define new attributes, that is not allowed for a Cell object. There truly only are 8 attributes of a Cell which the user should allowed to set.

Methods and Functions in the Cell Package

There are several other function that may be useful in your programs. Some of these are methods and some are non-exported functions. Except as noted, these subroutines are from the Xray::Xtal::Cell package.

attributes

Takes a list of attributes as its input and returns the values of the attributes as a list in the same order as the input list of attribute names. The return value is always an array, even if only one element is requested. This is the sole method of the Xray::Xtal package and is inherited by both the Site and Cell packages.

```

$cell -> Xray::Xtal::Cell::new()
      -> make("A"=>4.0, "B"=>4.3, "C"=>4.6)
($a, $b, $c) = $cell -> attributes("A", "B", "C");

$site -> Xray::Xtal::Site::new()
      -> make("X"=>0.5, "Y"=>0.25, "Z"=>0)
($x, $y, $z) = $site -> attributes("X", "Y", "Z");

print "$x, $y, $z, $a, $b, $c$"/";
      |-> 0.5, 0.25, 0, 4.0, 4.3, 4.6

```

clear

Reset a cell without destroying it.

```
$cell -> clear();
```

populate

Populate a unit cell given a list of sites. Each element of the list of sites must be a Site object. The symmetries operations implied by the space group are applied to each unique site to generate a description of the stoichiometric contents of the unit cell.

```
$cell -> populate(\@sites)
```

This fills the **Contents** attribute of the Cell with an anonymous array. Each element of the anonymous array is itself an anonymous array whose first three elements are the x, y, and z fractional coordinates of the site and whose fourth element is a reference to the Site that generated the position. This is, admittedly, a complicated data structure and requires a lot of “line-noise” style perl to dereference all its elements. It is, however, fairly efficient.

canonicalize_symbol

This takes a character string representing a space group and returns the canonical symbol for that group. See the Atoms document and the section on *INTERPRETING SPACE GROUP SYMBOLS* below for complete details about space group symbols and how they are interpreted by this function. This is not a method of the Cell class, it is just a normal function.

```
$string = Xray::Xtal::Cell::canonicalize_symbol($string);
```

For example

```

Xray::Xtal::Cell::canonicalize_symbol('pm3m')
yields "P m -3 m"

```

bravais

This takes the values of the **Space_group** and **Setting** attributes of a Cell and returns a list which specifies the Bravais translation vectors. This list is the truncation of the various Bravais translation three-vectors, thus has 0, 3, 6, or 9 elements. This is not method of the Cell class, it is just a normal function. It should rarely be necessary to call this, as the Bravais vector gets stored in the "Bravais" attribute of a cell.

```
($group, $setting) =
  $cell -> attributes("Space_group", "Setting");
@list = Xray::Xtal::Cell::bravais($group, $setting);
```

crystal_class

This returns a character string specifying the crystal class. The return value is one of "cubic", "hexagonal", "trigonal", "tetragonal", "orthorhombic", "monoclinic", or "triclinic". This is a method of the Cell class.

```
$class = $cell -> crystal_class();
```

verify_cell

Check the cell attributes. Issue warning messages if anything suspicious turns up. Exit if an error is found.

```
$cell -> verify_cell();
```

metric

Takes the three fractional coordinates and returns the cartesian coordinates of the position. The fractional coordinates need not be canonicalized into the first octant, thus this method can be used to generate the cartesian coordinates for any atom in a cluster.

```
($x,$y,$z) = $cell -> metric($xf, $yf, $zf);
```

This method is called repeatedly by the **build_cluster** function in the Xray::Atoms module. The elements of the metric tensor, i.e. the **Txx**, **Tyx**, **Tyz**, **Tzx**, and **Tz** Cell attributes, are used to make the transformation according to this formula:

$$(x \ y \ z) = \begin{vmatrix} / & T_{xx} & 0 & 0 & \backslash \\ | & T_{yx} & 1 & T_{yz} & | \\ \backslash & T_{zx} & 0 & T_{zz} & / \end{vmatrix} \begin{vmatrix} / & x_f & \backslash \\ | & y_f & | \\ \backslash & z_f & / \end{vmatrix}$$

d_spacing

Takes the Miller indices of a scattering plane and returns the d spacing of that plane in Angstroms.

```
$d = $cell -> d_spacing($h, $k, $l);
```

central

This method takes one argument, a string specifying the tag of the central atom. It returns a four element list of the central atom elemental symbol and its fractional coordinates.

```
($elem_c, $x_c, $y_c, $z_c) = $cell -> central($central_tag)
```

In many cases there is more than one choice for the central atom. Any of the various crystallographically identical positions matching the tag of the central atom can be chosen. This method returns the one closest to the center of the unit cell, i.e. the one closest to (1/2,1/2,1/2). The meaning of the word "closest" is a bit strange in this context. It is the position with the smallest value of

$$\text{sqrt}((x-0.5)^2 + (y-0.5)^2 + (z-0.5)^2)$$

This is chosen for speed and efficiency in building a spherical cluster of atoms.

overfull

This method returns the overfilled unit cell with atom positions in Cartesian coordinates. It takes an optional argument for specifying the epsilon defining which atoms should be considered close to a side, edge, or corner of the unit cell. The default for this value is 0.1 and the units are fractional cell coordinates. The return value is a list structured the same as the Contents attribute of the cell.

```
@overfull = $cell -> overfull($epsi);
```

Each element of @overfull is an anonymous array containing the x, y, and z fractional coordinates and a reference to the Site that generated the position.

warn_shift

This method returns a warning string if the space group is one for which the International Tables give two possible origin position. This string suggests a value for the shift vector. An empty string is returned if only one origin is given.

```
print $cell -> warn_shift;
```

cell_check

This method returns a warning if the cell axes and angles are not appropriate to the space group. It returns an empty string if they are appropriate.

```
print $cell -> cell_check;
```

get_symmetry_table

Given a space group symbol and an arcane string that I need to explain, return the list of symmetry operations from the space groups database. This returns an anonymous list of lists. Each list element contains three strings — the three strings which are eval-ed to generate the positions in the unit cell.

```
$positions = $cell -> get_symmetry_table;  
@first_position = @ {$positions[0]};
```

The Site object

The Site object is really just a blessed hash. It predefines several attributes of a Site, but allows the user to define new attributes on the fly. The predefined (i.e. the ones that are used in Atoms) attributes are:

Element

The two letter symbol for the chemical species.

Tag A character string identifying a unique crystallographic site.

X, Y, Z

The fractional coordinates of the sites (but see the section on *MOLECULES* below).

Bx, By, Bz

The thermal ellipsoid parameters for the site.

Valence

The formal valence for the element occupying the site.

Occupancy

The fractional occupancy of the site. This allows the user to specify dopants.

Host

This is 1 if the site is a host atom and 0 if it is a dopant.

Positions

This takes an anonymous array of symmetry equivalent sites. This is filled after calling the `populate` method.

File

The name of an external file to be used with the site.

Id A pseudo-random number assigned by the `new` method to uniquely identify the object.

Color

The color assigned to the site in a ball-and-stick image.

Methods and Functions of the Site Package

See the descriptions of `attributes` and `clear` in the Cell Package section.

clear

Reset a site without destroying it. For a site, this resets all predefined attributes to their initial values and all user-defined attributes to 0.

```
$site -> clear();
```

populate

This applied the symmetry operations implied by the space group of the Cell object pointed to by `\$cell` to a unique crystallographic site. The collection of symmetry related sites will be stored in the site object as the `Positions` attribute.

```
$site -> clear(\$cell);
```

Typically, it is not necessary to call this explicitly as the it is called for each site by the Cell populate method.

canonicalize_coordinate

This the subroutine is part of the Site package. It takes a fractional coordinate and returns it shifted into the first octant. This is not actually a method of the Site class, it is just a normal function.

```
$coord = Xray::Xtal::Site::canonicalize_coordniate($coord);
```

INTERPRETING SPACE GROUP SYMBOLS

You may specify space group symbols in a variety of ways. When presented with a possible symbol, the Xtal module will first clean the symbol up by adding and/or removing whitespace then by trying each of the following ways of interpreting the symbol.

1. A standard symbol from the 1995 edition of the International Tables for Crystallography. For example, the group for the cubic perovskite structure is $P m \bar{3} m$.
2. A Schoenflies symbol. The Schoenflies symbol for $P m \bar{3} m$ is O_h .
3. A symbol from the 1935 edition of the International tables. The 1935 symbol for $P m \bar{3} m$ is $P m \bar{3} m$.
4. The number for the entry of the space group in the International Tables. The number for $P m \bar{3} m$ is 221.
5. The full symbol. For $P m \bar{3} m$ this is $P 4/m \bar{3} 2/m$.
6. An alternative symbol. See the section on *LOW SYMMETRY SPACE GROUPS* for details.
7. A shorthand phrase, such as *fcc* for $F m \bar{3} m$. See (somewhere) for a complete list of shorthand phases.
8. A short symbol for a monoclinic space group. See *LOW SYMMETRY SPACE GROUPS* for details.

Since whitespace in the symbol is regularized, you can usually include or omit whitespace in any fashion. For example `pm3m` and `P m -3 m` both come out as `P m -3 m`, just as you would expect.

A space group is specified using the `make` method from the `Cell` class. When this is done the original symbol is stored in the `Given_group` attribute of the `Cell` object and the standard symbol interpreted from it is stored in the `Space_group` attribute. If the given symbol cannot be interpreted, the `Space_group` attribute is set to 0.

In the future, more flexibility may be added to the space group symbol interpretation scheme, including full symbols for alternate settings and more obsolete notation. I am open to suggestions.

LOW SYMMETRY SPACE GROUPS

There is a more complete discussion of low symmetry groups in the `Atoms` document. Here are the highlights:

Monoclinic Space Groups

Monoclinic groups can be quite confusing for the user. Because any of the three angles can be the unique (acute or obtuse rather than right) angle and because there are three possible settings for the cell, the standard symbol can be ambiguous. In that case, using the full symbol or the short monoclinic symbol would break the ambiguity and allow the `make` method to correctly assign the space group. An example would help illustrate this.

In "Structural phase diagram of $La(1-x)Sr(x)MnO(3+\delta)$: Relationship to magnetic and transport properties." by J.F. Mitchell et al, *Phys. Rev. B* 54, no. 9, (1996), pp./ 6172–6183 the monoclinic structure for room temperature $LnMnO_3$ is given in Table IV. The space group is given as `P 21/c` and the setting is given as `P 1 21/n 1`. Specifying the space group symbol will result in the wrong application of symmetries when the `Contents` attribute of the cell is filled. Specifying the full symbol for that setting will work correctly. The shorthand `P 21/n` will also result in a correct application of symmetries.

This section really needs to be fleshed out. Importance of symbol and angle to avoid ambiguity. Need to mention an angle even if all are 90. etc.

Orthorhombic Space Groups

You may use any setting of an orthorhombic crystal, although it is up to you to be sure that the unique coordinates are appropriate to that setting. `Atoms` has no way of checking that!

Tetragonal Space Groups

The `Cell` class does not currently handle F or C centered cells. If you have an actual literature reference where one of these centerings is used, I would love to know about it.

Rhombohedral Space Groups

You can specify either hexagonal or rhombohedral parameters for rhombohedral space groups. For the hexagonal representation, you must specify `a` and `b` and `c`, `alpha`, `beta`, and `gamma` will be set correctly by the `make` method. For the rhombohedral representation, you must specify `a` and `alpha` and the rest will be set appropriately.

BUGS

Surely there are plenty. Send me email if you find any.

AUTHOR

Bruce Ravel <bruce@phys.washington.edu>
<http://feff.phys.washington.edu/~rael>

SEE ALSO

The documentation that comes with the Atoms package.

D.3 SGB.PM

NAME

Xray::Tk::SGB — A widget for browsing space group symbols

DESCRIPTION

This widget is composite of a Text field, and Entry field, and two Buttons which supply as simple textual browsing mechanism for exploring space group symbols. In a typical use, it is hooked up to an Entry widget in a perl/Tk application so that a space group symbol can be specified in the application with the click of a mouse.

SYNOPSIS

```
use Tk;  
use Xray::Xtal;  
use Xray::Tk::SGB;
```



```
$browser = $top -> SGB(-SpaceWidget=>\$space_field);  
$browser -> configure(@sgb_args);  
$browser -> Show;
```

Configuration parameters are described below.

METHODS

The only non-standard method is **Show**, which is used to actually display the browser on your screen. **Show** always returns 0;

CONFIGURATION OPTIONS

-SpaceWidget

This is a reference to a widget, typically an Entry widget, into which the browser will insert space group symbols. If this is undefined or not a reference to a widget, then the Mouse-1 action of the browser is disabled.

-sgbActive

The color of active text in the browser.

-sgbGroup

The color used for particular emphasis of active text.

-button

Background color of the button.

-buttonActive

Active color of the buttons. Also the color of the text in the statusbar.

-buttonLabel

Color of the text on the buttons.

-buttonFont

The font to use on the button.

-sgbFont

The font to use in the text and status areas of the browser. This looks best if it is a fixed width font.

Other properties, such as **-foreground** are inherited and may be configured.

INTERNATIONALIZATION

All of the text displayed in the SGB widget can be customized using configure. The primary purpose of this is to allow the use of other languages in the SGB widget. The following list shows the configure switch and its default value. You should play around with SGB before translating or otherwise changing them so you can see what each string is used for.

- dismiss Dismiss
- back Back
- restore Restore
- Triclinic Triclinic
- Monoclinic Monoclinic
- Orthorhombic Orthorhombic
- Tetragonal Tetragonal
- Trigonal Trigonal
- Hexagonal Hexagonal
- Cubic Cubic
- MouseLInsert Mouse-1 to insert this symbol.
- MouseLDisplay Mouse-1 to display space group
- MouseADisplay Any mouse button for groups of this class.
- RestoreMsg Restore the initial symbol:
- DismissMsg Dismiss the space group browser.
- Groups groups
- MouseLR Mouse-1=insert Mouse-3=describe
- BackScreen Return to the previous screen.
- Number Number
- NumberDesc Space group index from ITC
- Schoenflies Schoenflies
- SchoenfliesDesc Schoenflies notation
- Full Full symbol

-
- FullDesc Symbol denoting complete symmetries
 - NewSymbol New symbol
 - NewSymbolDesc Symbol using \"e\" for the glide plane
 - Thirtyfive 1935 symbol
 - ThirtyfiveDesc Symbol from the 1935 edition of ITXC
 - Shorthand Shorthand
 - ShorthandDesc Special strings recognized by Atoms
 - Settings Settings
 - SettingsDesc Symbols for alternate settings
 - Short Short symbols
 - ShortDesc Short symbols for alternate settings
 - TopTop Select a crystal class to display all space groups
 within that class.
 - ViewCurrent View current group:
 - ClassTop Space group listing:
 - GroupTop Various symbols which may be used to describe space group:
 - Note Note:
 - SettingsMon The first three settings are for the b axis and beta
 angle unique. The next three are for the c axis and
 gamma angle unique. The final three are for the a
 axis and alpha angle unique. Within each group, the
 three choices are the possible choices of axes. Atoms
 has no a priori way of knowing the correct setting
 for your data.
 - SettingsOrt The five additional settings correspond to the five
 additional ways of setting a coordinate system in
 3-space. They correspond to permutations of ba-c,
 cab, -cba, bca, and a-cb away from the standard
 setting, abc. Atoms has no a priori way of knowing
 the correct setting for your data.

- SettingsTet** C and F centered tetragonal cells are related to the standard cells by a 45 degree rotation in ab plane and a doubling of the cell volume. Atoms has no a priori way of knowing the correct setting for your data.
- SettingsRho** You may use the rhombohedral parameters, in which case you must specify a and alpha. You may also use the trigonal parameters in which case you must specify a and c. The trigonal representation has three times the volume of the rhombohedral representation. Atoms has no a priori way of knowing the correct setting for your data.

USING THE SPACE GROUP BROWSER

Use of the browser is pretty straight forward because the status bar at the bottom of the widget always informs the user about what actions are available.

The opening panel displays the seven crystal classes. The names of the crystal classes are active text. Clicking any mouse button on a crystal class will display a list of all space groups in that class. If the widget specified by `-SpaceWidget` is displaying a valid group symbol, then that symbol will be displayed as active text. Clicking Mouse-1 on that symbol will jump to the panel describing that symbol.

The second panel displays lists of space groups divided by crystal class. The index of the group (as indexed in the International Tables of Crystallography) and the canonical (Hermann-Maguin) space group symbol are shown. The space group symbols are active text. Clicking Mouse-1 will insert that symbol into the widget given by `-SpaceWidget`. Clicking Mouse-3 will display information about that group.

The last panel is the space group description. It shows all symbols recognized as describing that group by the database in `Xray::Xtal`, which is used by `Xray::Tk::SGB`. Clicking Mouse-1 on any active text will insert that symbol into the widget given by `-SpaceWidget`.

At the bottom of the widget are three buttons. The "back" button causes the previous panel to be displayed. The "restore" is only active if the widget given by `-SpaceWidget` contains a symbol. If so, pressing it will restore that widget to its initial value. The "dismiss" button destroys the SGB widget.

TO DO

- The interaction of *SGB()*, *configure()*, and *Show()* is not as smooth as it should be.

Author

Bruce Ravel <ravel@phys.washington.edu> (c) 1999

```
http://feff.phys.washington.edu/~ravel/Xray/  
http://feff.phys.washington.edu/~ravel/atoms/
```

This code is distributed with Atoms and may be redistributed under the same terms as Atoms, which are the same terms as Perl itself.

D.4 ABSORPTION.PM

NAME

Xray::Absorption — obtain X-ray absorption data for the elements

SYNOPSIS

```
use Xray::Absorption;  
Xray::Absorption -> load("mcmaster");  
$xsec = Xray::Absorption->cross_section('Cu', 9000);
```

This example returns the cross section of Copper at 9000 eV using the McMaster tables.

DESCRIPTION

This module supports access to X-ray absorption data. It is designed to be a transparent interface to absorption data from a variety of sources. Currently, the only sources of data are the 1969 McMaster tables, the 1999 Elam tables, the 1993 Henke tables, and the 1995 Chantler tables. The Brennan-Cowen implementation of the Cromer-Liberman tables is available as a drop-on-top addition to this package. More resources can be added easily.

Because this is an object-oriented approach to X-ray absorption data, you must call subroutines as class methods rather than as subroutines:

```
$xsec = Xray::Absorption->cross_section('Cu', 9000);
```

is correct, but

```
$xsec = Xray::Absorption::cross_section('Cu', 9000);
```

is incorrect. Using class methods rather than a function oriented approach allows the user of the `Xray::Absorption` module to *hot swap* absorption data resources. For example

```
foreach $resource (Xray::Absorption->available) {  
  Xray::Absorption->load($resource);  
  print $resource, " : ",  
    Xray::Absorption->cross_section('Cu', 9000), $/;  
};
```

compares the cross section of copper at 9 keV as calculated from the all available data resources.

It is necessary to initialize `Xray::Absorption` to use a particular database by invoking the `load` method. This method establishes and changes inheritance.

METHODS

`current_resource`

Example:

```
$this = Xray::Absorption -> current_resource;
```

Identifies the currently selected resource.

`in_resource`

Example:

```
$is_there = Xray::Absorption -> current_resource($elem);
```

Returns true if `$elem` is tabulated in the current resource. `$elem` can be a two letter symbol, the full name of the element, or a Z number.

`get_energy`

Example:

```
$energy = $energy = Xray::Absorption -> get_energy($elem, $edge);
```

Returns the edge energy for `$elem`. `$edge` is one of K, L1, L2, L3, M1, etc. `$edge` may also be the Siegbahn or IUPAC symbol for a fluorescence line. Some data resources provide more lines than others. Some may provide no lines at all. See the documentation for each resource for which lines are available. When either `$elem` or `$edge` is an unrecognized symbol, this method returns 0.

`next_energy`

Example:

```
$next = Xray::Absorption -> next_energy($elem, $edge, @list);
```

Given a list of atomic symbols `@list`, return a list containing the element symbol, edge symbol, and energy in eV of the next highest edge energy after the `$edge` edge of `$elem`. This returns an empty list if the any argument is unrecognizable.

`cross_section`

Example:

```
$xsec = Xray::Absorption -> cross_section($elem, $edge, $mode);
```

Return the cross section in barns/atom of `$elem` at `$energy`. The optional `$mode` argument tells this method what kind of data to return. The default for all data resources is to return the cross section, however each resource has several other option. For example, the McMaster tables offer along with the absorption cross section, the coherent scattering, the incoherent scattering, or the sum of the three contributions. The allowed values for `$mode` are "photo", "coherent", "incoherent", and "full", respectively. Other resources may offer anomalous scattering factors or other data. See the documentation for the individual resource modules. If `$mode` is not given or is given incorrectly, the full cross section is returned.

If an energy is requested which is right on an edge, all data resources assume that you want the cross-section just above the edge. The granularity of the comparison between the requested energy and the edge energy is 1 millivolt, so if you want a cross-section just below an edge, you should request an energy that is more than 1 millivolt less than the value returned by `get_energy`.

`data_available`

Example:

```
$is_there = Xray::Absorption -> data_available($elem, $edge);
```

Returns true if the selected resource contains sufficient data for handling the specified element in the energy range around the specified edge. Returns false otherwise.

`available`

Example:

```
@list = Xray::Absorption -> available;
```

Returns a list of all available data resource.

`scattering`

Example:

```
@list = Xray::Absorption -> scattering;
```

Returns a list of all available data resource which contain anomalous scattering functions.

`verbose`

Example:

```
@list = Xray::Absorption -> verbose($arg);
```

Turn verbose operation on or off. If `$arg` evaluates to true, then warning messages will be printed to standard error. If `$arg` evaluates to false, then methods will silently return 0 when they encounter problems.

`get_atomic_weight`

Example:

```
$value = Xray::Absorption -> get_atomic_weight($elem);
```

Return the atomic weight of `$elem`.

`get_density`

Example:

```
$value = Xray::Absorption -> get_density($elem);
```

Return the specific gravity of the pure material of `$elem`.

`get_conversion`

Example:

```
$value = Xray::Absorption -> get_conversion($elem);
```

Return the factor for converting between barns/atom and cm squared/gram for `$elem`.

`get_Siegbahn`

Example:

```
$symbol = Xray::Absorption -> get_Siegbahn($sym);
```

Return the short Siegbahn symbol for an x-ray fluorescence line. Thus "Ka1", "Kalpha1", and "K-L3" all return "Ka1". The case of the input symbol does not matter and the symbol is returned capitalized. White space and underscores will be removed from the input symbol. The symbol "lb2,15" is translated to "lb2". This returns 0 if `$sym` is not a recognizable symbol for a line.

`get_Siegbahn_full`

Example:


```
$symbol = Xray::Absorption -> get_Siegbahn_full($sym);
```

Return the full Siegbahn symbol for an x-ray fluorescence line. Thus "Ka1", "Kalpha1", and "K-L3" all return "Kalpha1". The case of the input symbol does not matter and the symbol is returned capitalized. White space and underscores will be removed from the input symbol. This returns 0 if \$sym is not a recognizable symbol for a line.

```
get_IUPAC
```

Example:

```
$symbol = Xray::Absorption -> get_IUPAC($sym);
```

Return the IUPAC symbol for an x-ray fluorescence line. Thus "Ka1", "Kalpha1", and "K-L3" all return "K-L3". The case of the input symbol does not matter and the symbol is returned in all capitals. White space and underscores will be removed from the input symbol. This returns 0 if \$sym is not a recognizable symbol for a line.

SYMBOLS FOR FLUORESCENCE LINES

To specify fluorescence lines, Siegbahn or IUPAC symbols may be used. methods are provided for converting between these notations. The Siegbahn notations can be in the short or full forms. Here is a table of all recognized symbols:

Full Siegbahn	Short Siegbahn	IUPAC

Kalpha1	Ka1	K-L3
Kalpha2	Ka2	K-L2
Kalpha3	Ka3	K-L1
Kbeta1	Kb1	K-M3
Kbeta2	Kb2	K-N2,3
Kbeta3	Kb3	K-M2
Kbeta4	Kb4	K-N4,5
Kbeta5	Kb5	K-M4,5
Lalpha1	La1	L3-M5
Lalpha2	La2	L3-M4
Lbeta1	Lb1	L2-M4
Lbeta2	Lb2	L3-N4,5
Lbeta3	Lb3	L1-M3
Lbeta4	Lb4	L1-M2
Lbeta5	Lb5	L3-O4,5
Lbeta6	Lb6	L3-N1
Lgamma1	Lg1	L2-N4
Lgamma2	Lg2	L1-N2
Lgamma3	Lg3	L1-N3

Lgamma6	Lg6	L2-04
L1	L1	L3-M1
Lnu	Ln	L2-M1
Malpha	Ma	M5-N6,7
Mbeta	Mb	M4-N6
Mgamma	Mg	M3-N5
Mzeta	Mz	M4,5-N6,7

In addition, the symbols `Lb2,15` and `Lbeta2,15` are recognized as synonyms for `Lbeta2`. The methods which interpret these symbols will remove spaces and underscores from the input string. Thus `K_alpha_1` and `K a 1` will both be recognized as `Kalpha1`. Since hyphens are part of the IUPAC notation, `K-alpha-1` will not be recognized as `Kalpha1`. Thus use spaces or underscores if you want to make the Siegbahn notation more legible.

ABSORPTION DATA RESOURCES

Currently, `Xray::Absorption` has the McMaster, Elam, Henke, and Chantler tables as its data resources. New resources may be added over time. This section offers a few guidelines to anyone interested in supplying more resources. It does not matter how the new resource calculates the cross section. That is hidden behind the object-orientedness of the `Xray::Absorption` module. It is essential that the new resource take the namespace `Xray::Absorption::Resource`, where *Resource* is a descriptive name, like `McMaster` or `Elam`. It is essential that the new resource supply these methods

```
current_resource
in_resource
get_energy
next_energy
cross_section
```

and that they use the semantics described above. All other methods described in the last section are defined in *Absorption.pm* and do not need to be redefined in the resource modules.

New resources are welcome to define new methods particular to that data resource in addition to the 5 required methods.

UNITS

All energies returned by the methods of `Xray::Absorption` are in electron volts. All cross sections are in units of barns per atom. A conversion constant between that unit and cm squared per gram is supplied by the `get_conversion` method. Atomic weights are in atomic units. Densities are given as specific gravity (i.e. dimensionless).

BUGS AND THINGS TO DO

- It would be nice to not have to `load` if you only want to use the default resource.
- Check to be sure things are properly unloaded and overloaded when switching resources.
- Test for pathological cases, such as elements that don't exist, that have Z's in the hundreds, energies that are very low or very high, and so on.
- It would be nice to implement the list that is used by the `next_energy` method as a doubly linked list. In that way, a `previous_energy` method would be trivial.

AUTHOR

Bruce Ravel <rael@phys.washington.edu>
<http://feff.phys.washington.edu/~rael/Xray/>

D.5 MCMaster.PM

NAME

Xray::Absorption::McMaster — Perl interface to the McMaster tables

SYNOPSIS

```
use Xray::Absorption;  
Xray::Absorption -> load("mcmaster");
```

See the documentation for Xray::Absorption for details.

DESCRIPTION

This module is inherited by the Xray::Absorption module and provides access to the data contained in the 1969 McMaster tables.

The data in this module, commonly referred to as "The McMaster Tables", was originally published as

Compilation of X-Ray Cross Sections
W.H. McMaster, N. Kerr Del Grande, J.H. Mallett, J.H. Hubbell

National Bureau of Standards
UCRL-50174 Section II Revision 1
(1969)
Available from National Technical Information Services L-3
United States Department of Commerce

The data is contained in a database file called *mcmaster.db* which is generated at install time from a flat text database of the McMaster data. The data originally comes from *mucal.f*, a Fortran subroutine originally written by Dr. Pathikrit Bandhyapodhyay.

The required Chemistry::Elements module is available from CPAN in the miscellaneous modules section.

EDGE AND LINE ENERGIES

The McMaster data resource only includes K and L 1–3 edges. For light elements, it provides only a single L edge energy — that for the L1 edge. For heavier elements it provides a single M energy, the energy of the M1 edge. It only supplies four generic fluorescence line energies, Kalpha, Kbeta, Lalpha, and Lbeta. In each case the energy provided is the energy of the brightest line of that sort.

BUGS AND THINGS TO DO

- Make sure this handles fluorescence lines which are in other resources, but not in this resource in a sensible manner.
- What happens if you call `line_toggle` when this is loaded? How about when elam was loaded but then you switch to this one?

AUTHOR

Bruce Ravel, bruce@phys.washington.edu
<http://feff.phys.washington.edu/~rael/>

D.6 ELAM.PM

NAME

Xray::Absorption::Elam — Perl interface to the Elam tables

SYNOPSIS

```
use Xray::Absorption;  
Xray::Absorption -> load("elam");
```

See the documentation for Xray::Absorption for details.

DESCRIPTION

This module is inherited by the Xray::Absorption module and provides access to the data contained in the 1999 Elam tables of absorption cross-sections and line and edge energies.

The data in this module, here referred to as "The Elam Tables", will be published real soon. The compilation of data is the work of Tim Elam (tim.elam@nrl.navy.mil).

The data is contained in a database file called *elam.db* which is generated at install time from a flat text database of the Elam data. The data is stored in a Storable archive using "network" ordering. This allows speedy disk and memory access along with network and platform portability.

The required Chemistry::Elements, Math::Spline, and Math::Derivative modules are available from CPAN.

LITERATURE REFERENCES

K-shell fluorescence yield below $Z=11$ from new fits in J. H. Hubbell et. al., J. Chem. Phys. Ref. Data, Vol. 23, No. 2, 1994, pp. 339–364.

Fluorescence yields and Coster-Kronig transition rates for K and L shells Krause, J. Phys. Chem. Ref. Data, Vol. 8, No. 2, 1979, pp. 307–327. values for wK, wL2, and f23 are from Table 1. (values for light atoms in condensed matter) (note that this produces a large step in f23 values at $z=30$, see discussion in reference section 5.3 L2 Subshell and section 7 last paragraph)

Values of wL1 for $Z=85-110$ and f12 for $Z=72-96$ from Krause were modified as suggested by W. Jitschin, "Progress in Measurements of L-Subshell Fluorescence, Coster-Kronig, and Auger Values", AIP Conference Proceedings 215, X-ray and Inner-Shell Processes, Knoxville, TN, 1990. T. A. Carlson, M. O. Krause, and S. T. Manson, Eds. (American Institute of Physics, 1990).

Fluorescence yields and Coster-Kronig transition rates for M shells Eugene J. McGuire, "Atomic M-Shell Coster-Kronig, Auger, and Radiative Rates, and Fluorescence Yields for Ca-Th", Physical Review A, Vol. 5, No. 3, March 1972, pp. 1043–1047.

Fluorescence yields and Coster-Kronig transition rates for N shells Eugene J. McGuire, "Atomic N-shell Coster-Kronig, Auger, and Radiative Rates and Fluorescence Yields for $38 \leq Z \leq 103$ ", Physical Review A 9, No. 5, May 1974, pp. 1840–1851. Values for $Z=38$ to 50 were adjusted according to instructions on page 1845, at the end of Section IV.a., and the last sentence of the conclusions.

Relative emission rates, fits to low-order polynomials, low-Z extrapolations by hand and eye data from Salem, Panossian, and Krause, Atomic Data and Nuclear Data Tables Vol. 14 No.2 August 1974, pp. 92–109. M shell data is from T. P. Schreiber and A. M. Wims, X-ray Spectrometry Vol. 11, No. 2, 1982, pp. 42–45. Small, arbitrary intensities assigned to Mgamma and Mzeta lines.

Cross sections are in log cm²/gm vs log energy in eV. Berger and Hubbell above 1 keV, Henke et. al. below.

Reference: M. J. Berger and J. H. Hubbell, XCOM: Photon Cross Sections on a Personal Computer, Publication NBSIR 87–3597, National Bureau of Standards, Gaithersburg, MD, 1987. Machine-readable data from J. H. Hubbell, personal communication, Nov. 9, 1998. The data were updated as of May 7, 1998 (XCOM Version 2.1).

Reference: B. L. Henke, E. M. Gullikson, and J. C. Davis, X-ray Interactions: Photoabsorption, Scattering, Transmission, and Reflection at E=50–30000 eV, Z=1–92, Atomic and Nuclear Data Tables, July 1993, vol. 54 (no. 2): 181–342. Machine-readable data downloaded on Nov. 16, 1998 via

http://www-cxro.lbl.gov/optical_constants/intro.html

METHODS

The behaviour of the `get_energy` method in this module is a bit different from other modules used by `Xray::Absorption`. This section describes methods which behave differently for this data resource and methods offered by this module which are not available for other resources.

`get_energy`

Example:

```
$energy = Xray::Absorption -> get_energy($elem, $edge)
```

This behaves similarly to the `get_energy` method of other resources, except there are some differences regarding the syntax of specifying `$edge`. When using the Elam data resource, `$edge` can be any of K, L1-L3, M1-M5, N1-N7, O1-O7, or P1-P3. To get a fluorescence line, you may use any Siegbahn or IUPAC symbol to specify the line. See the pod in `Xray::Absorption` for details about these symbols. You may also specify a "generic" Siegbahn symbol, such as Kalpha. The energy that is returned depends on the value of an internal variable which may be set using the `line_toggle` method. If the toggle is set to "brightest", the energy of the brightest line of the class is returned. In the case of "Kalpha", the energy of the Kalpha1 line is returned. If the toggle is set to "weighted" then the intensity weighted average energy of all lines of the class is returned. "weighted" is the default.

`line_toggle`

Toggle the method of computing a generic fluorescence line between "weighted" and "brightest". This determines the response to a use of `get_energy` like this:

```
$energy = Xray::Absorption -> line_toggle("brightest");
$energy = Xray::Absorption -> get_energy("cu", "kalpha");
$energy = Xray::Absorption -> line_toggle("wieghthed");
$energy = Xray::Absorption -> get_energy("cu", "kalpha");
```

When "weighted" is selected, this returns the intensity weighted energy of the various Kalpha lines. When "brightest" is chosen, this returns the energy of the Kalpha1 line because that is the brightest Kalpha line. The default is "weighted". Case does not matter for the argument, but spelling does. If the argument is not spelled correctly then the calculation method is not toggled.

get_intesity

Example:

```
$intensity = Xray::Absorption -> get_intesity($elem, $symbol)
```

Get the relative of the line specified by \$symbol for the element \$elem. \$elem can be a two letter symbol, a full name, or a Z number. \$symbol may be either a Siegbahn or IUPAC symbol. The intensities are such that all lines of a type (e.g. all Kalpha lines) have intensities which sum to 1. If \$elem or \$symbol is not recognized, then this returns 0.

EDGE AND LINE ENERGIES

The Elam data resource provides a fairly complete set of edge and line energies. Any edge tabulated on the Gwyn William's Table of Electron Binding Energies for the Elements (that's the one published by NSLS and on the door of just about every hutch at NSLS) is in the Elam data resource. Additionally, a large but not exhaustive collection of line energies is tabulated. Every line in the table in the **SYMBOLS FOR FLUORESCENCE LINES** section of the **Absorption.pm** pod is included in the Elam tables. A reasonable value for the relative line intensity is also included in this table. See (the elam reference) for a discussion of which lines were included in the tables and how the intensities were calculated.

BUGS AND THINGS TO DO

- The "weighted" option for **get_energy** is not quite right in that it counts in lines from different edges. While that might be appropriate, there is some question as to the relative weights of lines from different edges. So a better solution would be to only use lines from the edge directly below the chosen energy.

AUTHOR

Bruce Ravel, bruce@phys.washington.edu
<http://feff.phys.washington.edu/~rael/>

D.7 HENKE.PM

NAME

Xray::Absorption::Henke — Perl interface to the Henke tables

SYNOPSIS

```
use Xray::Absorption;  
Xray::Absorption -> load("henke");
```

See the documentation for Xray::Absorption for details.

DESCRIPTION

This module is inherited by the Xray::Absorption module and provides access to the data contained in the Henke tables of anomalous scattering factors and line and edge energies.

The data in this module, referred to as "The Henke Tables", was published as

B. L. Henke, E. M. Gullikson, and J. C. Davis,
Atomic Data and Nuclear Data Tables Vol. 54 No. 2 (July 1993).

The Henke data is available on the web at http://www-cxro.lbl.gov/optical_constants/ and more information about the data can be obtained from Eric Gullikson <EMGullikson@lbl.gov>.

The data is contained in a database file called *henke.db* which is generated at install time from the flat text files of the Henke data. The data is stored in a Storable archive using "network" ordering. This allows speedy disk and memory access along with network and platform portability.

The required File::Spec, Chemistry::Elements, Storable, modules are available from CPAN.

METHODS

The behaviour of the methods in this module is a bit different from other modules used by Xray::Absorption. This section describes methods which behave differently for this data resource.

get_energy

Example:

```
$energy = Xray::Absorption -> get_energy($elem, $edge);
```


This behaves similarly to the `get_energy` method of the other resources. When using the Henke data resource, `$edge` can be any of K, L1-L3, M1-M5, N1-N7, O1-O7, or P1-P3. Line energies are not supplied with the Henke data set. The line energies from the McMaster tables are used.

`cross_section`

Example:

```
$xsec = Xray::Absorption -> cross_section($elem, $energy, $mode);
```

This behaves slightly differently from the similar method for the McMaster and Elam resources. The Henke tables are actually tables of anomalous scattering factors and do not come with coherent and incoherent scattering cross-sections. The photo-electric cross-section is calculated from the imaginary part of the anomalous scattering by the formula

$$\mu = 2 * r_e * \lambda * \text{conv} * f_2$$

where, `r_e` is the classical electron radius, `lambda` is the photon wavelength, and `conv` is a units conversion factor.

```
r_e      = 2.817938 x 10^-15 m
lambda   = 2 pi hbar c / energy
hbar*c   = 1973.27053324 eV*Angstrom
conv     = Avagadro / atomic weight
          = 6.022045e7 / weight in cgs
```

The `$mode` argument is different here than for the other resources. The options are "xsec", "f1", and "f2", telling this method to return the cross-section or the real or imaginary anomalous scattering factor, respectively.

The values for `f1` and `f2` are computed by linear interpolation of a semi-log scale. Care is taken to avoid the discontinuities at the edges.

Because the Henke tables do not include the coherent and incoherent scattering terms, the value returned by `get_energy` may be a bit smaller using the Henke tables than that from the McMaster tables.

EDGE AND LINE ENERGIES

The Henke data resource provides a fairly complete set of edge energies. Any edge tabulated on the Gwyn William's Table of Electron Binding Energies for the Elements (that's the one published by NSLS and on the door of just about every hutch at NSLS) is in the Henke data resource. The Henke data comes with the same, limited set of fluorescence energies as McMaster.

BUGS AND THINGS TO DO

- It would be nice to improve the inter-/extrapolation near absorption edges. As it stands, these tables produce really poor DAFS output.

AUTHOR

Bruce Ravel, bruce@phys.washington.edu
<http://feff.phys.washington.edu/~rael/>

D.8 CHANTLER.PM

NAME

Xray::Absorption::Chantler — Perl interface to the Chantler tables

SYNOPSIS

```
use Xray::Absorption;  
Xray::Absorption -> load("chantler");
```

See the documentation for Xray::Absorption for details.

DESCRIPTION

This module is inherited by the Xray::Absorption module and provides access to the data contained in the Chantler tables of anomalous scattering factors and line and edge energies.

The data in this module, referred to as "The Chantler Tables", was published as

C. T. Chantler
Theoretical Form Factor, Attenuation, and Scattering Tabulation
for $Z = 1 - 92$ from $E = 1 - 10$ eV to $E = 0.4 - 1.0$ MeV
J. Phys. Chem. Ref. Data 24, 71 (1995)

This can be found on the web at

<http://physics.nist.gov/PhysRefData/FFast/Text/cover.html>

The Chantler data is available on the web at

<http://physics.nist.gov/PhysRefData/FFast/html/form.html>

More information can be found on the personal web page of C.T. Chantler

<http://optics.ph.unimelb.edu.au/~chantler/home.html>

The data contained in a database file called *chantler.db* which is generated at install time from the flat text files of the Chantler data. The data is stored in a Storable archive using "network" ordering. This allows speedy disk and memory access along with network and platform portability.

The required `File::Spec`, `Chemistry::Elements`, and `Storable` modules are available from CPAN.

METHODS

The behaviour of the methods in this module is a bit different from other modules used by `Xray::Absorption`. This section describes methods which behave differently for this data resource.

`get_energy`

Example:

```
$energy = Xray::Absorption -> get_energy($elem, $edge);
```

This behaves similarly to the `get_energy` method of the other resources. When using the Chantler data resource, `$edge` can be any of K, L1-L3, M1-M5, N1-N7, O1-O5, or P1-P3. Line energies are not supplied with the Chantler data set. The line energies from the McMaster tables are used.

`cross_section`

Example:

```
$xsec = Xray::Absorption -> cross_section($elem, $energy, $mode);
```

This behaves slightly differently from the similar method for the McMaster and Elam resources. The Chantler tables contain anomalous scattering factors and the sum of the coherent and incoherent scattering cross-sections. The photo-electric cross-section is calculated from the imaginary part of the anomalous scattering by the formula

$$\mu = 2 * r_e * \lambda * \text{conv} * f_2$$

where, `r_e` is the classical electron radius, `lambda` is the photon wavelength, and `conv` is a units conversion factor.

```

r_e      = 2.817938 x 10^-15 m
lambda   = 2 pi hbar c / energy
hbar*c   = 1973.27053324 eV*Angstrom
conv     = Avagadro / atomic weight
          = 6.022045e7 / weight in cgs

```

The `$mode` argument is different here than for the other resources. The options are "xsec", "f1", "f2", "photo", and "scatter" telling this method to return the full cross-section cross-section, the real or imaginary anomalous scattering factor, just the photoelectric cross-section, or just the coherent and incoherent scattering, respectively.

The values for f1 and f2 are computed by linear interpolation of a semi-log scale, as described in the literature reference. Care is taken to avoid the discontinuities at the edges.

EDGE AND LINE ENERGIES

The Chantler data resource provides a fairly complete set of edge energies. Any edge tabulated on the Gwyn William's Table of Electron Binding Energies for the Elements (that's the one published by NSLS and on the door of just about every hutch at NSLS) is in the Chantler data resource. The Chantler data comes with the same, limited set of fluorescence energies as McMaster.

BUGS AND THINGS TO DO

- It would be nice to improve the inter-/extrapolation near absorption edges. As it stands, these tables produce really poor DAFS output.

AUTHOR

Bruce Ravel, bruce@phys.washington.edu
<http://feff.phys.washington.edu/~rael/>

D.9 CL.PM

NAME

Xray::Absorption::CL — Perl interface to the Cromer-Liberman tables

SYNOPSIS

```
use Xray::Absorption;  
Xray::Absorption -> load("cl");
```

See the documentation for Xray::Absorption for details.

DESCRIPTION

This module is inherited by the Xray::Absorption module and provides access to the data contained in the Cromer-Liberman tables of anomalous scattering factors and line and edge energies.

The data in this module and the Fortran code which it calls as a shared library, referred to as "The CL Tables", was published as

S. Brennan and P.L. Cowen, Rev. Sci. Instrum, vol 63,
p.850 (1992)

More information about these data is available on the Web at

<http://www.slac.ssrl.stanford.edu/absorb.html>.

The values for the anomalous scattering factors are calculated using subroutines written in Fortran, wrapped in a swig wrapper and accessed using the Perl DynaLoader module.

The values of edge and line energies are contained in a database file called *cl.db* which is generated at install time from the flat text files of the these data. The data is stored in a Storable archive using "network" ordering. This allows speedy disk and memory access along with network and platform portability.

The required File::Spec, Chemistry::Elements, Storable are available from CPAN.

METHODS

The behaviour of the methods in this module is a bit different from other modules used by Xray::Absorption. This section describes methods which behave differently for this data resource.

`get_energy`

Example:

```
$energy = Xray::Absorption -> get_energy($elem, $edge);
```

This behaves similarly to the `get_energy` method of the other resources. When using the CL data resource, `$edge` can be any of K, L1-L3, M1-M5, N1-N7, O1-O7, or P1-P3. Line energies are not supplied with the CL data set. The line energies from the McMaster tables are used.

`cross_section`

Example:

```
$xsec = Xray::Absorption -> cross_section($elem, $edge, $mode);
```

This behaves slightly differently from the similar method for the McMaster and Elam resources. The CL tables are actually tables of anomalous scattering factors and do not come with coherent and incoherent scattering cross-sections. The photo-electric cross-section is calculated from the imaginary part of the anomalous scattering by the formula

$$\mu = 2 * r_e * \lambda * \text{conv} * f_2$$

where, `r_e` is the classical electron radius, `lambda` is the photon wavelength, and `conv` is a units conversion factor.

```
r_e      = 2.817938 x 10^-15 m
lambda   = 2 pi hbar c / energy
hbar*c   = 1973.27053324 eV*Angstrom
conv     = Avagadro / atomic weight
          = 6.022045e7 / weight in cgs
```

The `$mode` argument is different here than for the other resources. The options are "xsec", "f1", and "f2", telling this method to return the cross-section or the real or imaginary anomalous scattering factor, respectively.

The values for `f1` and `f2` are computed by linear interpolation of a semi-log scale. Care is taken to avoid the discontinuities at the edges.

Because the CL tables do not include the coherent and incoherent scattering terms, the value returned by `get_energy` is a bit smaller using the CL tables than using the others.

EDGE AND LINE ENERGIES

The CL data resource provides a fairly complete set of edge energies. Any edge tabulated on the Gwyn William's Table of Electron Binding Energies for the Elements (that's the one published by NSLS and on the door of just about every hutch at NSLS) is in the CL data resource. The CL data comes with the same, limited set of fluorescence energies as McMaster.

BUGS AND THINGS TO DO

None that I know about...

AUTHOR

Bruce Ravel, bruce@phys.washington.edu
<http://feff.phys.washington.edu/~rael/>

D.10 CLDATA.PM

NAME

CLdata — Perl interface to Cromer-Liberman calculations of anomalous x-ray scattering factors.

SYNOPSIS

```
use Xray::Absorption::CLdata qw(fprime);
my $e_ref = [8000, 8500, 8900, 8950, 8970, 8990, 9000, 9200];
my $atom = 29;
fprime($atom, $e_ref, $f1, $f2);
for ($i = 0; $i <= $#{$e_ref}; $i++) {
    print " $e_ref->[$i]    $f1->[$i]    $f2->[$i]\n";
}
```

```
use Xray::Absorption::CLdata qw(cl_f1 cl_f2);
print " cl_f1(29,8000) = ". cl_f1(29,8000)."\n";
print " cl_f2(30,9500) = ". cl_f2(30,9500)."\n";
print " cl_f2(30,9700) = ". cl_f2(30,9700)."\n";
```

DESCRIPTION

CLdata gives access to the Cromer-Liberman data of anomalous x-ray scattering factors. The basic strategy is to specify an element by atomic number and a set of x-ray energies in eV, and to get back $f'(E)$ and $f''(E)$ — the real and imaginary parts of the anomalous scattering factor.

fprime

The fprime function provides the main interface to the Cromer-Liberman data. The call syntax is:

```
fprime($atom, $e_ref, $f1_ref, $f2_ref);
```

\$atom is the atomic number of the element of interest and **\$e_ref** is an array reference to a list of energies, both of which are inputs to *fprime*. The output parameters **\$f1_ref** and **\$f2_ref** are references to arrays of $f'(E)$ and $f''(E)$ at each of the energy values specified.

cl_f1

This returns a single value of $f'(E)$ for a given Z and energy. The syntax is

```
$f1 = cl_f1($Z, $energy);,
```

where \$energy is a scalar value.

cl_f2

This returns a single value of f'' for a given Z and energy. The syntax is

```
$f2 = cl_f2($Z, $energy);.
```

where \$energy is a scalar value.

AUTHOR

Matthew Newville — newville@cars.uchicago.edu

SEE ALSO

D.11 CROMERMANN.PM

NAME

Xray::CromerMann — Perl interface to Thomson scattering factors

SYNOPSIS

```
use Xray::CromerMann qw(get_f get_valence in_CromerMann);
$fnot = get_f($symb, $d);
```

DESCRIPTION

This module provides a functional interface to the Cromer-Mann table of coefficients for calculating the Thomson (kinematical) scattering factors of the elements and common valence

states. The coefficients are stored externally in the cromann.db database file. The coefficients are for an Aikman expansion, which is of this form:

$$f_0 = \sum_{i=1}^4 [a_i \exp(-b_i s^2)] + c$$

Thus there are 9 coefficients for each of the 213 tabulated element/valence symbols.

s is $\sin(\theta)/\lambda$. $(\lambda s)/2\pi$ is the momentum transfer. s is simply related to the crystal d-spacing by $s=1/2d$.

The data for these tables can be found in Volume C of the International Tables of Crystallography, ed. A.J.C. Wilson, published by IUCr and Kluwer Academic Publishers (1992). The table starts on page 500 and a discussion can be found on page 487.

These tables are known to be inaccurate, particularly at high angles. This module is a toy. It is suitable for a student or for the sort of quick-n-dirty crystallographic hackery that the author indulges in.

EXPORTED SUBROUTINES

get_f

This function calculates the Thomson scattering for a given symbol and d-spacing. The Thomson scattering depends only on the momentum transfer. The d-spacing of the scattering planes is a closely related quantity and is easily calculated from the crystal structure, see the *Xtal.pm* manpage.

```
$symb = "Ce3+";
$fnot = get_f($symb, $d);
```

If the symbol cannot be found in the table, `get_f` returns 0. It also returns 0 when `$symbol` consists of whitespace or is "null" or "nu". If `$symbol` is a number or the name of an element, then it assumes you want the Thomson scattering for the neutral element. The absolute value of `$d_spacing` is used by this function.

get_valence

This returns the proper element/valence symbol for use with `get_f`. `$elem` is a two-letter atomic symbol, and `$valence` is the valence of the ion. `$valence` can be an integer or a float.

```
$symbol = get_valence($elem, $valence)
```

The function takes the nearest integer to `$valence` and uses that with the element symbol to construct the element/valence symbol. See the section on *ELEMENTS AND VALENCE STATES*.

in_CromerMann

This is a test of whether a given symbol is tabulated in the Cromer-Mann table. It returns the symbol itself if found in the table or 0 if it is not in the table.

```
$symb = "Ce3+";
$is_available = in_CromerMann($symb);
```

ELEMENTS AND VALENCE STATES

The following is a list of symbols for the tabulated elements and valence states. The final two are ways of referring to an empty site (i.e. a null or blank atom).

H	H.	H1-	He	Li	Li1+	Be
Be2+	B	C	C.	N	O	O1-
F	F1-	Ne	Na	Na1+	Mg	Mg2+
Al	Al3+	Si	Si.	Si4+	S	P
Cl	Cl1-	Ar	K	K1+	Ca	Ca2+
Sc	Sc3+	Ti	Ti2+	Ti3+	Ti4+	V
V2+	V3+	V5+	Cr	Cr2+	Cr3+	Mn
Mn2+	Mn3+	Mn4+	Fe	Fe2+	Fe3+	Co
Co2+	Co3+	Ni	Ni2+	Ni3+	Cu	Cu1+
Cu2+	Zn	Zn2+	Ga	Ga3+	Ge	Ge4+
As	Se	Br	Br1-	Kr	Rb	Rb1+
Sr	Sr2+	Y	Y3+	Zr	Zr4+	Nb
Nb3+	Nb5+	Mo	Mo3+	Mo5+	Mo6+	Tc
Ru	Ru3+	Ru4+	Rh	Rh3+	Rh4+	Pd
Pd2+	Pd4+	Ag	Ag1+	Ag2+	Cd	Cd2+
In	In3+	Sn	Sn2+	Sn4+	Sb	Sb3+
Sb5+	Te	I	I1-	Xe	Cs	Cs1+
Ba	Ba2+	La	La3+	Ce	Ce3+	Ce4+
Pr	Pr3+	Pr4+	Nd	Nd3+	Pm	Pm3+
Sm	Sm3+	Eu	Eu2+	Eu3+	Gd	Gd3+
Tb	Tb3+	Dy	Dy3+	Ho	Ho3+	Er
Er3+	Tm	Tm3+	Yb	Yb2+	Yb3+	Lu
Lu3+	Hf	Hf4+	Ta	Ta5+	W	W6+
Re	Os	Os4+	Ir	Ir3+	Ir4+	Pt
Pt2+	Pt4+	Au	Au1+	Au3+	Hg	Hg1+
Hg2+	Tl	Tl1+	Tl3+	Pb	Pb2+	Pb4+
Bi	Bi3+	Bi5+	Po	At	Rn	Fr
Ra	Ra2+	Ac	Ac3+	Th	Th4+	Pa
U	U3+	U4+	U6+	Np	Np3+	Np4+
Np6+	Pu	Pu3+	Pu4+	Pu6+	Am	Cm
Bk	Cf	O2-	' '	Nu		

AUTHOR

Bruce Ravel, bruce@phys.washington.edu
<http://feff.phys.washington.edu/~ravel>

Appendix E

The Perl Artistic License

This is a verbatim copy of the Perl Artistic License, the license under which ATOMS is released. The “Copyright Holder” of ATOMS and all related files is Bruce Ravel. Some files in the ATOMS package are explicitly labeled as having been written by another person, in which case the “Copyright Holder” of the individual file is the person listed as its author.

The "Artistic License"

Preamble

The intent of this document is to state the conditions under which a Package may be copied, such that the Copyright Holder maintains some semblance of artistic control over the development of the package, while giving the users of the package the right to use and distribute the Package in a more-or-less customary fashion, plus the right to make reasonable modifications.

Definitions

"Package" refers to the collection of files distributed by the Copyright Holder, and derivatives of that collection of files created through textual modification.

"Standard Version" refers to such a Package if it has not been modified, or has been modified in accordance with the wishes of the Copyright Holder as specified below.

"Copyright Holder" is whoever is named in the copyright or

copyrights for the package.

"You" is you, if you're thinking about copying or distributing this Package.

"Reasonable copying fee" is whatever you can justify on the basis of media cost, duplication charges, time of people involved, and so on. (You will not be required to justify it to the Copyright Holder, but only to the computing community at large as a market that must bear the fee.)

"Freely Available" means that no fee is charged for the item itself, though there may be fees involved in handling the item. It also means that recipients of the item may redistribute it under the same conditions they received it.

1. You may make and give away verbatim copies of the source form of the Standard Version of this Package without restriction, provided that you duplicate all of the original copyright notices and associated disclaimers.
2. You may apply bug fixes, portability fixes and other modifications derived from the Public Domain or from the Copyright Holder. A Package modified in such a way shall still be considered the Standard Version.
3. You may otherwise modify your copy of this Package in any way, provided that you insert a prominent notice in each changed file stating how and when you changed that file, and provided that you do at least ONE of the following:
 - a. place your modifications in the Public Domain or otherwise make them Freely Available, such as by posting said modifications to Usenet or an equivalent medium, or placing the modifications on a major archive site such as uunet.uu.net, or by allowing the Copyright Holder to include your modifications in the Standard Version of the Package.
 - b. use the modified Package only within your corporation or organization.
 - c. rename any non-standard executables so the names do not conflict with standard executables, which must also be provided, and provide a separate manual page for each non-standard executable that clearly documents how it differs from the Standard Version.
 - d. make other distribution arrangements with the Copyright Holder.

4. You may distribute the programs of this Package in object code or executable form, provided that you do at least ONE of the following:
 - a. distribute a Standard Version of the executables and library files, together with instructions (in the manual page or equivalent) on where to get the Standard Version.
 - b. accompany the distribution with the machine-readable source of the Package with your modifications.
 - c. give non-standard executables non-standard names, and clearly document the differences in manual pages (or equivalent), together with instructions on where to get the Standard Version.
 - d. make other distribution arrangements with the Copyright Holder.
5. You may charge a reasonable copying fee for any distribution of this Package. You may charge any fee you choose for support of this Package. You may not charge a fee for this Package itself. However, you may distribute this Package in aggregate with other (possibly commercial) programs as part of a larger (possibly commercial) software distribution provided that you do not advertise this Package as a product of your own. You may embed this Package's interpreter within an executable of yours (by linking); this shall be construed as a mere form of aggregation, provided that the complete Standard Version of the interpreter is so embedded.
6. The scripts and library files supplied as input to or produced as output from the programs of this Package do not automatically fall under the copyright of this Package, but belong to whomever generated them, and may be sold commercially, and may be aggregated with this Package. If such scripts or library files are aggregated with this Package via the so-called "undump" or "unexec" methods of producing a binary executable image, then distribution of such an image shall neither be construed as a distribution of this Package nor shall it fall under the restrictions of Paragraphs 3 and 4, provided that you do not represent such an executable image as a Standard Version of this Package.
7. C subroutines (or comparably compiled subroutines in other languages) supplied by you and linked into this Package in order to emulate subroutines and variables of the language defined by this Package shall not be considered part of this Package, but are the equivalent of input as in Paragraph 6, provided these subroutines do not change the language in any way that would cause it to fail the regression tests for the language.

8. Aggregation of this Package with a commercial distribution is always permitted provided that the use of this Package is embedded; that is, when no overt attempt is made to make this Package's interfaces visible to the end user of the commercial distribution. Such use shall not be construed as a distribution of this Package.
9. The name of the Copyright Holder may not be used to endorse or promote products derived from this software without specific prior written permission.
10. THIS PACKAGE IS PROVIDED "AS IS" AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

The End
