

LA-UR-06-7002

*Approved for public release;
distribution is unlimited.*

Title: The makxsf Code with Doppler Broadening

Author(s): Forrest B. Brown

Intended for: Description and Input Manual
for the makxsf Code



Los Alamos National Laboratory, an affirmative action/equal opportunity employer, is operated by the Los Alamos National Security, LLC for the National Nuclear Security Administration of the U.S. Department of Energy under contract DE-AC52-06NA25396. By acceptance of this article, the publisher recognizes that the U.S. Government retains a nonexclusive, royalty-free license to publish or reproduce the published form of this contribution, or to allow others to do so, for U.S. Government purposes. Los Alamos National Laboratory requests that the publisher identify this article as work performed under the auspices of the U.S. Department of Energy. Los Alamos National Laboratory strongly supports academic freedom and a researcher's right to publish; as an institution, however, the Laboratory does not endorse the viewpoint of a publication or guarantee its technical correctness.

The *makxf* Code with Doppler Broadening

Forrest B. Brown
X-3 Monte Carlo Codes
Los Alamos National Laboratory
fbrown@lanl.gov

ABSTRACT

The *makxf* code is a utility program for manipulating cross-section library files for the *mcnp5* Monte Carlo code. *makxf* has been used to convert ACE data files between ASCII and binary formats and to make customized libraries containing selected datasets. In 2006, *makxf* capabilities were extended to permit the creation of temperature-dependent libraries. Routines from the NJOY and DOPPLER codes were incorporated into *makxf* to provide for Doppler broadening of resolved data to any higher temperature, and for interpolating $S(\alpha,\beta)$ thermal scattering kernels and probability tables for unresolved resonance data between two bracketing temperatures.

1. INTRODUCTION

The *makxf* code is a utility program for manipulating cross-section library files for the *mcnp5* Monte Carlo code [1]. The basic functions performed by *makxf* include the following:

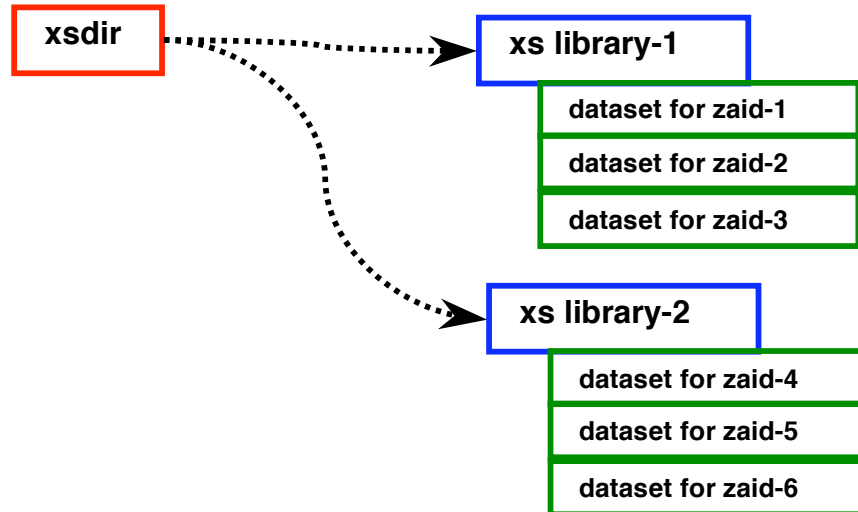
- *makxf* can be used to change the format of *mcnp5* cross-section libraries from Type-1 to Type-2, or vice versa. Type-1 data files are formatted, sequential-access, ASCII files, while Type-2 files are unformatted, direct-access, binary files.
- *makxf* can be used to copy entire libraries to new files or to copy selected nuclide data-sets to new libraries, for either Type-1 or Type-2.
- *makxf* can be used to create nuclide datasets at new temperatures, resulting in a temperature-dependent library for a specific application. Creation of a nuclide dataset at a new temperature involves 3 basic operations:
 1. Doppler broaden the resolved resonance data to a higher temperature.
 2. Interpolate any unresolved resonance probability tables to the new temperature.
 3. Interpolate $S(\alpha,\beta)$ thermal scattering kernel data to the new temperature.

The ability to create nuclide datasets at new temperatures was added to *makxf* in early 2006 by incorporating several routines from the NJOY [2] and DOPPLER [3] codes.

- Create a new *xmdir* file which includes all of the above changes

The program *makxf* is compiled appropriately for a given computer platform using the *mcnp5* *makefile* system when *mcnp5* is built. It is located in directory *MCNP5/Source/datasrc*.

Figure 1. Data Files for MCNP

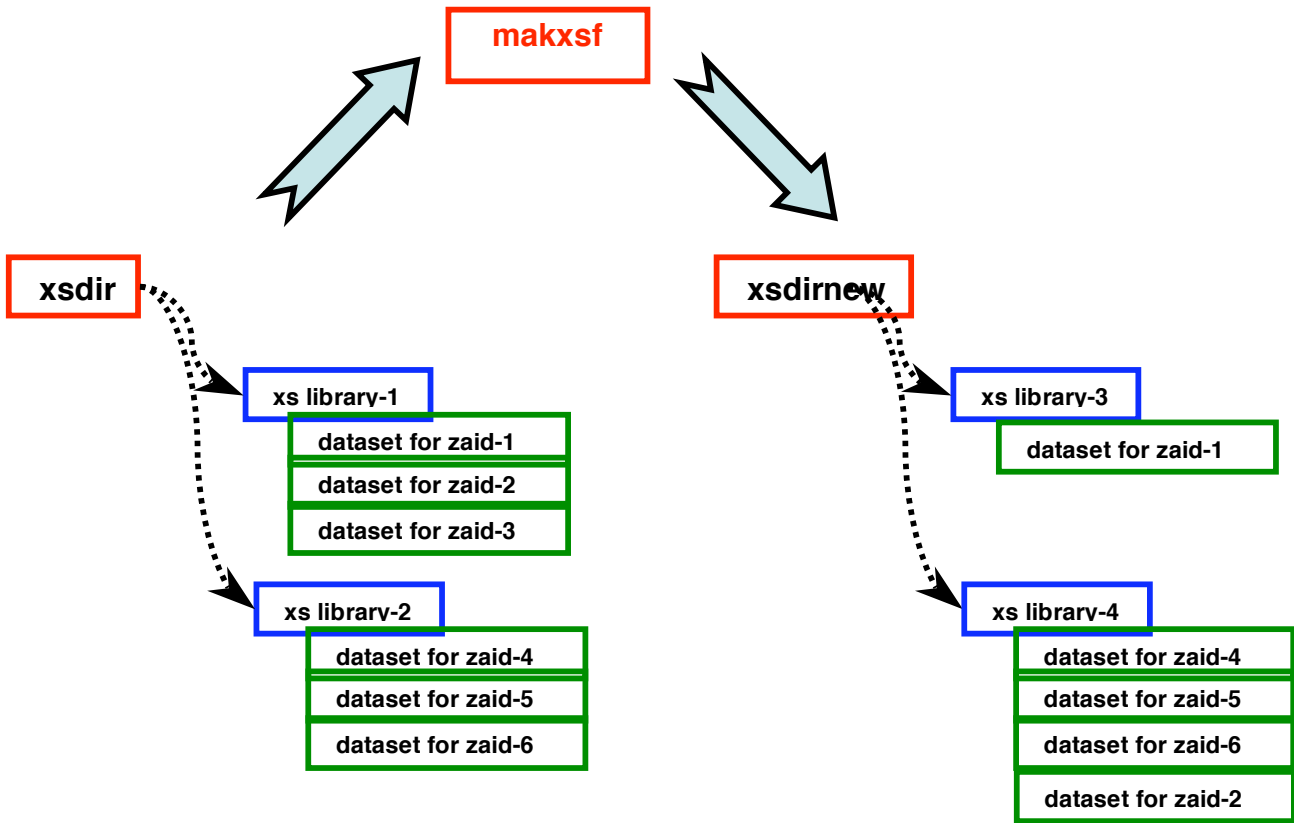


2. DATA FILES FOR *mcnp5*

The cross-section data files used by *mcnp5* are illustrated in Figure (1). The *xsdir* file is a free-format ASCII text file that provides directory information. *xsdir* contains the atomic mass ratios for nuclides and a listing of the available cross-section datasets. Information in *xsdir* for each ZAID provides the name of the library file containing the data and information on the location of the data within the file. The cross-section library files may be either Type-1 or Type-2, and may contain one or more ACE datasets. See Appendices F, G, and K in [1] for a detailed description of the ACE dataset contents and format.

Type-1 data files are formatted, sequential-access, ASCII files, while Type-2 files are unformatted, direct-access, binary files. *mcnp5* can read cross-section data from either Type-1 or Type-2 files. RSICC distributes Type-1 files with the standard *mcnp5* distribution because they are portable across platforms. For some compilers and computers, however, reading very large Type-1 the files can be slow and leads to long startup times for *mcnp5* calculations. Type-2 files provide much faster direct access to binary data, reducing the *mcnp5* startup time, but are not portable among computer systems.

Figure (2) illustrates the use of *makxsf* to create new cross-section libraries. Given some instructions contained in the *specs* file, *makxsf* reads an existing *xsdir* file to obtain information on the locations and types of existing datasets. *makxsf* then can create new cross-section library files of either Type-1 or Type-2, which may contain datasets for one or more ZAID's. After all of the new files are created, a new *xsdir* is created which provides the information needed to locate and read data for each ZAID in the new library files. If any of the new datasets are to be

Figure 2. Typical Application of *makxsf*

adjusted to a new temperature, *makxsf* invokes the Doppler broadening and interpolation routines before writing the new datasets into the library files.

3. *specs* FILE FORMAT

makxsf reads an input file named *specs* to acquire instructions for reading existing cross-section libraries and creating new ones. *specs* is an ASCII text file created by a user according to the following general rules:

- The file is free-format, with blank or tab delimiters between data fields on each line.
- Each line should be 80-characters or less in length.
- Lines ending in the characters "&" or "+" will be continued on the next line.
- Comments may be placed anywhere in the file by starting a line with one of the characters "#", "|", "\$", or "!".
- Floating-point numbers can be entered in any format, such as 1, 1., 1.0, 1e+1, 1.0E+1.
- Integer quantities should be entered as whole numbers, without decimal points or exponents.

The contents of the *specs* file are specified below in Table (1).

4. CONVERTING CROSS-SECTION FILES WITH MAKXSF

makxsf can be used to convert cross-section libraries from one format to another and to construct custom-designed cross-section libraries. *makxsf* can be used to convert Type-1 files into the faster-access Type-2 files, or to convert Type-2 cross-sections into portable Type-1 cross-sections. You can also use *makxsf* to delete cross-section datasets that are not needed and to reorganize the cross-section datasets into custom-designed cross-section libraries. The input files to *makxsf* are: one or more existing cross-section libraries; a directory file, *xmdir*, which describes the input cross-section libraries; and a file called *specs* that tells *makxsf* what it is supposed to do. The output files are: one or more new cross-section libraries, and a new directory file that describes the new cross-section libraries. The input and output cross-section libraries can be any combination of Type-1 and Type-2 files. The various types of cross-section libraries and the form and contents of the cross-section directory file are described in detail in Appendix F of [1]. The directory file *xmdir* in the *mcnp5* distribution package contains complete descriptions of all the cross-section files in that package. Printing it provides a useful reference. The sample *specs* file in the *mcnp5* data package can be used with *makxsf* to create a complete set of Type-2 files from the Type-1 files provided. The *specs* file is a formatted sequential file with records not exceeding 80 characters. The data items in each record may start in any column and are delimited by blanks or tabs. The structure of the *specs* file is given in Table (1) and an example is given in Table (2).

Table 1. *specs* File Record Structure (* = optional entry)

<i>Record</i>	<i>Contents</i>
0	datapath=path-to-directory-containing-old-xmdir*
1	Name-of-old-xmdir-file Name-of-new-xmdir-file
2	Name-of-old-xslib-file* Name-of-new-xslib-file Type Recl* Epr*
3	Path-entered-into-new-xmdir-file* (or blank line)
4+	Nuclide list, if Name-of-old-xslib-file is absent
...	Blank record

Record 0 is optional. It can be used to specify the location of the directory containing the existing (old) *xmdir* file. If Record 0 is absent, then *makxsf* will attempt to get the directory location from the environmental variable DATAPATH. If neither Record 0 nor the DATAPATH variable is present, then the current directory will be used as the location of the old *xmdir* file.

The default for *Epr* (entries per record) is 512. *Recl* (record length) will be set appropriately by *makxsf*, depending on whether your platform specifies records in terms of words (OSF and PC) or bytes (all others), resulting in 8 bytes per entry. All data is stored in double precision. You should NOT need to use these options.

Records 2 through 4+ can be repeated any number of times with data for additional new cross-section libraries. The *specs* file ends with a blank record. If “Name-of-old-xslib-file” exists on Record 2, all nuclides from that library will be converted.

Table 2. Example specs File

<i>Record</i>	<i>Contents</i>
1	# create 2 new xs library files & a new xsdir
2	#
3	# old-xsdir new-xsdir
4	xsdir1 xsdir2
5	#
6	# create xslib file el2
7	# existing-xslib new-xslib new-type
8	el1 el2 2
9	# unix path for new-xslib, to put into new-xsdir
10	/home/scratch/el2
11	# done creating first new xslib file
12	#
13	# create another xslib, with selected nuclide datasets
14	# name of new-xslib (type-2)
15	rmccsab2 2
16	# unix path to the new xslib file, to be put into new-xsdir
17	/datalib/rmccsab2
18	# list of zaid's to put into new-xslib-file
19	7015.55c
20	1001.50c
21	# done creating rmccsab2
22	[blank line]

In Table (2), the *specs* file starts with the Type-1 cross-section directory file *xsdir1*, electron library *el1*, and neutron libraries *rmccsa1* and *rmccs1*. All nuclides in the electron data file *el1* are to be converted to a Type-2 file called *el2*. Records 19-20 tell *makxsf* to search all libraries listed in *xsdir1* until it finds nuclides 7015.55c and 1001.50c (which happen to be in file *rmccsa1* and *rmccs1*, respectively) and construct a new Type-2 library *rmccsab2* consisting of only these nuclides. The new directory file *xsdir2* will tell *mcnp5* to look for the electron cross sections in */home/scratch/el2* and for the neutron cross sections in */datalib/rmccsab2*. If the type of the new cross-section file is specified to be 1 in Record 15, only the name of the new cross-section file and the 1 for the type are read in that record. If the type in Record 15 is 2, the record length and the number of entries per record can be specified, but it should not be necessary. If there is any difficulty, be sure that you compile *makxsf* with the option `-DDIRACCESS_RECL_WORD` for platforms specifying records in terms of words (OSF and PC) and without that flag for platforms specifying record lengths in bytes. Running the Type-2 tests of the test suite from the *install* script is a good way to be sure the settings are correct. The optional access route on Record 17 of the *specs* file is a concatenation of a Unix path with the library filename and becomes the fourth entry for each nuclide in the library in the *xsdir2* file. It is not necessary to generate all the cross-section files that you will ever need in one *makxsf* run. You can combine and edit directory files at any time with a text editor or with another *makxsf* run. The only requirement is that you must give *mcnp5* a directory file that points to all the cross-section files

that are needed by the current problem. If you plan to run a long series of *mcnp5* problems that all use the same small set of cross-section tables, it might be convenient to generate with *makxsf* a small special-purpose cross-section file and directory file just for your project.

5. ADJUSTING CROSS-SECTION DATA TO NEW TEMPERATURES

makxsf can be used to create nuclide datasets at new temperatures, resulting in a temperature-dependent library for a specific application. Creation of a nuclide dataset at a new temperature involves 3 basic operations:

1. Doppler broaden the resolved resonance data to a higher temperature.
2. Interpolate any unresolved resonance probability tables to the new temperature.
3. Interpolate $S(\alpha,\beta)$ thermal scattering kernel data to the new temperature.

The ability to create nuclide table-sets at new temperatures was added to *makxsf* in early 2006 by incorporating several routines from the NJOY [2] and DOPPLER [3] codes. *mcnp5* users can now conveniently create libraries at the specific temperatures required for their problems without having to learn about and run NJOY. *makxsf* and the DOPPLER code perform essentially identical temperature adjustments to data-sets. However, DOPPLER has not been officially released since its development in 2003, the documentation [3] is not publicly available, and support is uncertain. *makxsf* is intended to be maintained as a separate code, to be distributed both along with the *mcnp5* distribution and separately as a stand-alone utility code. Because [3] is not publicly available, portions of the description of the DOPPLER code are reprinted below (text surrounded by [] was copied from other parts of the document, for clarity):

... Instead of having to go back to the basic evaluated data and reprocess with NJOY, DOPPLER starts from an existing MCNP data set in ACE format. By using the same “kernel broadening” logic used by NJOY, DOPPLER is able to prepare new cross sections for any temperature greater than the temperature of the base data set with the same accuracy as would be obtained by using NJOY directly. The code cannot obtain quite as an exact solution for the thermal scattering tables or the unresolved-range probability tables. It is assumed that enough temperature points are available for those tables so that the data at the desired temperature can be obtained by interpolation. ...

A. Pointwise Doppler Broadening

Doppler broadening of pointwise cross sections in both DOPPLER and NJOY uses the accurate “kernel broadening” approach, which is described in detail in the BROADR chapter of the NJOY manual. This method works with detailed cross sections represented using piece-wise linear segments chosen to represent the base cross sections to some desired tolerance. Roughly speaking, the NJOY libraries usually represent cross sections to within about 0.1%. The broadening algorithm used in DOPPLER tries to compute the broadened cross sections to a precision of about 0.1%. There is some accumulation of errors in this process, but in general, the accuracy of the broadened cross sections is very good.

There are several physical effects of Doppler broadening that can be seen in the DOPPLER results. Low-energy $1/v$ cross sections (capture or fission) tend to be unchanged by Doppler broadening. Low-energy constant cross sections (elastic scattering) pick up a $1/v$ component. The low-energy total cross section will be a combination of these effects. Resonance peaks in the cross sections will be broadened

out, and in some cases, an additional $1/v$ component will be generated. Cross sections at energies high with respect to the energy of thermal motion or resonances that are broad with respect to the thermal energy will be unaffected by broadening. The broadening of the resonances will often mean that the resulting cross sections will be smoother than the base cross sections and require fewer energy points to represent them. DOPPLER readjusts the energy grid to account for this effect. However, the broadened version of a resonance may require fewer energy grid points to represent its shape near the peak but more points further out on the side of the resonance. DOPPLER also accounts for this effect properly. On occasion, the number of energy points required to represent the broadened cross section may increase. One example of this is when the base cross sections were constructed to an accuracy criterion that was looser than the 0.1% hard wired into DOPPLER. The code prints out the number of energy points found in the base cross section set and the new number of points generated for the broadened cross section set.

[Subroutine *broada* is used to doppler broaden the pointwise cross sections from the base ACE file read into memory by *acein*. Only reactions that show resonance effects need to be broadened. These include the normal ACE principal reactions, total (MT=1), elastic (MT=2), absorption (MT=101), and heating (MT=301). Photon production is included when present (MT=202). The code searches for other low-threshold reactions, typically finding fission (MT=18) and capture (MT=102). The maximum energy for Doppler broadening *emax* is set to the lowest of the thresholds for the high-energy cross sections, typically the threshold for the first inelastic level (MT=51), or to the beginning energy for the unresolved range probability tables when present. The code prints out the identifiers for the reactions being broadened and the maximum energy being used. The cross sections for these reactions in this energy range are then loaded into memory. An additional area of memory is assigned for the new cross sections after broadening, allowing for the possibility that there could be up to twice as many energy points in the broadened cross sections. Subroutine *broadn* is then called to take care of the actual Doppler broadening of the selected cross sections. The code prints the number of points that were generated for the new energy grid of the broadened cross sections. The next step in subroutine *broada* is to insert the new cross sections into the ACE file. This is fairly complicated because various blocks of data have to be moved to account for the changes in lengths of the cross section blocks, thresholds have to be fixed up, and pointers have to be reconstructed. Finally, the modified ACE data have to be moved down to the normal position of the ACE data in memory and the temperature field in the ACE structure has to be updated to its new value. The ACE data are now ready to be written out with *aceout*.

Subroutine *broadn* is a slightly modified version of the subroutine of the same name in the BROADR module of NJOY. Given a cross section that is described by a set of linear segments, it Doppler broadens it to any higher temperature. For details, please refer to the NJOY manual. The routines *bsigma*, *hunky*, *funky*, and *hnabb* are also fully described in the NJOY manual.]

B. Thermal Scattering Tables

At thermal energies, MCNP uses two different methods to account for the scattering of neutrons. For most materials, it attempts to construct a free-gas scattering model based on the constant elastic cross section that it finds at low energies. This model is analytic and handled internally. However, for important moderator materials like hydrogen bound in water, deuterium bound in heavy water, beryllium, beryllium oxide, graphite, zirconium hydride, polyethylene, and so on, it is important to take the binding of the material in the solid or liquid into account. MCNP libraries normally provide a set of such "Class t" files on a fairly coarse grid of temperatures. These tables provide the thermal inelastic (and sometimes elastic) cross sections tabulated versus incident energy. In the inelastic case, they also provide outgoing energy-and-angle spectra for the scattered neutrons using a set of discrete angles and secondary energies chosen to represent equally probable pieces of the angle-energy phase space.

DOPPLER can easily generate data for a temperature that lies in between two of the temperatures available in the library tables assuming that thermal cross sections are linear functions of the temperature. [Simple linear-linear interpolation is used for the inelastic cross section and all the

thermal elastic data.] However, this doesn't work well for the equally probable emission elements. [A special reciprocal interpolation law is used for the inelastic energy-angle emissions.] The extent of the data in the energy-angle phase space is inversely proportional to the probability for that element, so the interpolation is done using a reciprocal law. If you are halfway in between the grid temperatures, it behaves something like parallel resistors, i.e., $R = R_1 R_2 / (R_1 + R_2)$. Clearly, this method will be more accurate if there are a sufficient number of well chosen temperatures in the base library. Experience has shown that this method tends to produce smooth variations of k_{eff} versus temperature in practice.

C. Probability Tables

Probability tables used to calculate self-shielding effects in the unresolved energy range may be provided in the base MCNP library at several temperatures. We simply interpolate each element of the table assuming that the table elements are linear in the log of the temperature. Here again, reasonable closely spaced temperature values for the base tables will improve the accuracy of the interpolated results. The effect of the unresolved range is often less important than the resonance cross sections or the thermal scattering tables for many practical problems, and they may not be represented well in some MCNP library sets. With the current formats, giving a new temperature in the unresolved range requires giving an entire pointwise library at that temperature. Therefore, a library that gives detailed coverage of the unresolved range can be very large. However, when unresolved self-shielding is important, more closely spaced temperature values may be needed in the base library. The ENDF66 library provided with MCNP5 provides room temperature and 3000K only.

The *specs* input file for *makxsf* was described above in Table (1). To adjust a particular nuclide dataset to a different temperature, only minor changes are needed in the list of nuclides:

- If a nuclide dataset is to be copied as-is from an existing library file to a new library file with no change in temperature, then a line of the nuclide list in *specs* should just contain the ZAIID of the existing dataset:

ZAIID

- If a nuclide dataset does not contain probability tables for unresolved resonances, then it can be Doppler broadened to any higher temperature. (A second dataset at a higher temperature is not necessary, since no interpolation is needed.) In this case, the line of the nuclide list in *specs* should contain the ZAIID to be used in the new library file, the temperature of the new dataset, the ZAIID of the existing nuclide (at the lower temperature), and a 0 (which tells *makxsf* that a higher temperature dataset is not needed):

ZAIID_{new} T_{new} ZAIID_{original} 0

The temperature may be specified in MeV or in degrees-Kelvin. *makxsf* assumes that a temperature value less than 1.0 is given in MeV, and that any temperature value greater than 1.0 is degrees-Kelvin.

If the original dataset does contain probability tables for unresolved resonances, *makxsf* will perform Doppler broadening to the new temperature, but will simply copy the probability tables as-is to the new dataset, since interpolation cannot be performed without a second, higher-temperature dataset (see next case).

- If a nuclide dataset contains both resolved data and probability tables for unresolved resonances, then existing datasets are needed at temperatures that bracket the new temperature. In this case, the line of the nuclide list in specs should contain 4 entries:

$$\text{ZAID}_{\text{new}} \quad T_{\text{new}} \quad \text{ZAID}_{\text{original, lower T}} \quad \text{ZAID}_{\text{original, higher T}}$$

The 3rd entry must correspond to a dataset generated at a temperature lower than T_{new} , and the 4th entry must correspond to a dataset generated at a temperature higher than T_{new} . The 3rd and 4th entries must have the same ZA part of the ZAID, but different ID suffixes.

- To adjust the $S(\alpha,\beta)$ thermal scattering datasets, interpolation between two existing datasets must be performed. The line of the nuclide list in specs should contain 4 entries just as in the preceding case:

$$\text{ZAID}_{\text{new}} \quad T_{\text{new}} \quad \text{ZAID}_{\text{original, lower T}} \quad \text{ZAID}_{\text{original, higher T}}$$

A sample *specs* file in Table (3) illustrates the various options for copying nuclide datasets and for adjusting nuclide datasets to different temperatures.

Table 3. Example SPECS File Including Temperature Adjustments

<i>Record</i>	<i>Contents</i>
1	xmdir1 xmdir2
2	#
3	# test for new makssf
4	#
5	# old-xmdir new-xmdir
6	testdir1 testdir2
7	#
8	# copy selected xs-datasets to testlib2, type2
9	testlib2 2
10	/home/fbrown/mcnp6_new/MCNP6/Testing/xsec_data/testlib2
11	1001.60c
12	1001.50m
13	7014.50d
14	[blank line]
15	#
16	# copy some more datasets to testlib3, type 1
17	testlib3 1
18	/home/fbrown/mcnp6_new/MCNP6/Testing/xsec_data/testlib3
19	7000.01g
20	7014.50m
21	8016.40c
22	82000.50m

```

23          [blank line]
24          #
25          # copy some more datasets to testlib4, type 2
26          testlib4 2
27          /home/fbrown/mcnp6_new/MCNP6/Testing/xsec_data/testlib4
28          92235.50d
29          92000.01g
30          92235.50m
31          92238.40c
32          92238.50m
33          lwtr.01t
34          # for doppler broadening, T can be MeV or degrees-K
35          92235.00c 540 92235.69c 92235.68c
36          #
37          # if higher T does not exist or is not needed, use 0 for 4th entry
38          8016.01c 540 8016.69c 0
39          [blank line]
40          #
41          # done, create the new xsdir file & exit

```

6. TESTING

Four different types of verification calculations were performed to verify that the revisions to *makxsf* were done properly, that compatibility with previous versions was maintained, and that the temperature adjustments were performed correctly:

1. The regression tests in the *mcnp5* and *mcnp6* build systems were changed so that the testing of Type-2 cross-sections exercises more features of *makxsf*. This includes creating several different cross-section library files, some Type-1 and some Type-2. *makxsf* is now used to convert *testlib1* (Type-1) into 3 separate cross-section libraries: *testlib2* (Type-2), *testlib3* (Type-1), *testlib4* (Type-2). The results of the regression tests did not change when run with the new cross-section data (except for the lines which describe the files from which the cross-section datasets were read). The changes to the *mcnp5* build system for Type-2 testing using *makxsf* were checked into the CVS repository.
2. About a dozen different nuclide datasets from the standard *mcnp5* libraries were adjusted to different temperatures separately using first DOPPLER and then *makxsf*. Every word of these various datasets was compared to verify that *makxsf* performed the same Doppler broadening and interpolation as the DOPPLER code. The datasets from the 2 codes matched exactly in all tests, except the comparison for the zirconium-hydride thermal scattering data ($Zr/H S(\alpha,\beta)$ dataset). The differences for that one dataset were traced to a bug in DOPPLER that had been fixed in *makxsf* (a pointer was not initialized properly in DOPPLER for interpolating $S(\alpha,\beta)$ elastic scattering.)

3. Several different Doppler-broadened datasets were created using both DOPPLER and *makxsf*. Using the ^{235}U datasets from each code, various cross-sections were plotted vs energy using the *mcnp5* cross-section plotting. These visual comparisons did not reveal any discrepancies between the two codes.
4. For a realistic verification of the Doppler broadening and interpolation routines in *makxsf*, the hot Kritz benchmark problems from [4] were calculated using *makxsf* and *mcnp5*. First, *makxsf* was used to create nuclear datasets at 245 C. Basic data were taken from ENDF/B-VI Release 6 (ENDF66). Then, *mcnp5* calculations were performed for the hot 2-D benchmarks. Each calculation employed 550 generations with 10,000 neutron histories per generation. Results from first 50 generations were discarded, giving 5,000,000 active histories for each case.

The results from these calculations were compared to those from [4], where a similar procedure had been followed first with NJOY and *mcnp5*, and then with DOPPLER and *mcnp5*. All three calculation schemes (NJOY/*mcnp5*, DOPPLER/*mcnp5*, and *makxsf*/*mcnp5*) should produce the same results within statistics. (Because of round-off errors, they need not produce bit-for-bit identical results.) Results from the 3 sets of calculations are shown in Table (4). Within statistics, all results agree. This comparison verifies that using NJOY, DOPPLER, or *makxsf* to perform the temperature adjustments results in nuclide datasets that produce the same *mcnp5* results.

**Table 4. Results from Kritz Benchmark Calculations
(NJOY and DOPPLER results are from [4].)**

<i>Case</i>	<i>Library</i>	<i>keff</i>	Δk (vs NJOY)
Kritz:2-1	NJOY	0.9914 ± 0.0003	—
	DOPPLER	0.9911 ± 0.0003	-0.0003 ± 0.0004
	new <i>makxsf</i>	0.9913 ± 0.0003	-0.0001 ± 0.0004
Kritz:2-13	NJOY	0.9944 ± 0.0003	—
	DOPPLER	0.9942 ± 0.0003	-0.0002 ± 0.0004
	new <i>makxsf</i>	0.9940 ± 0.0003	-0.0004 ± 0.0004
Kritz:2-19	NJOY	1.0005 ± 0.0003	—
	DOPPLER	1.0009 ± 0.0003	0.0004 ± 0.0004
	new <i>makxsf</i>	1.0004 ± 0.0003	-0.0001 ± 0.0004

7. CONCLUSIONS

The *makxsf* code was modified to add the capability of adjusting nuclide datasets to different temperatures. All previously existing features of *makxsf* were preserved. *makxsf* now provides a very convenient way for *mcnp5* users to create problem-specific, temperature-dependent cross-section libraries for use in *mcnp5* calculations. The revised *makxsf* code was verified by direct comparison of output datasets, by visual comparison of output datasets, by the *mcnp5* regression tests, and by repeating the calculations performed in [4]. *makxsf* is included with the standard *mcnp5* distribution from RSICC and is also available as a stand-alone utility code.

REFERENCES

1. X-5 Monte Carlo Team, "MCNP – A General Monte Carlo N-Particle Transport Code, Version 5" , Los Alamos National Laboratory (2003)
LA-UR-03-1987 - Volume I: Overview and Theory,
LA-CP-03-0245 - Volume II: User's Guide,
LA-CP-03-0284 - Volume III: Developer's Guide
2. R.E. MacFarlane & D.W. Muir, "The NJOY Nuclear Data Processing System, Version 91", LA-12740-M (1994).
3. R.E. MacFarlane & P. Talou, "DOPPLER: A Utility Code for Preparing Customized Temperature-Dependent Data Libraries for the MCNP Monte Carlo Transport Code", unpublished (Oct 3, 2003)
4. R.D. Mosteller, R.E. MacFarlane, R.C. Little, M. White, "Analysis of Hot and Cold Kritz Benchmarks With MCNP5 and Temperature-specific Nuclear Data Libraries", LA-UR-03-7071 (2003).