



Converting ENDF libraries into relational format*

Anatoliy G. Yuferov¹

¹ JSC “Pilot and Demonstration Center for Decommissioning of Uranium-Graphite Nuclear Reactors”, 13 Avtodoroga st., bld. 179a, Seversk, 636000 Russian Federation

Corresponding author: *Anatoliy G. Yuferov* (anatoliy.yuferov@mail.ru)

Academic editor: *Yuri Korovin* ♦ **Received** 18 September 2018 ♦ **Accepted** 19 September 2018 ♦ **Published** 17 October 2018

Citation: Yuferov AG (2018) Converting ENDF libraries into relational format. Nuclear Energy and Technology 4(1): 57–63. <https://doi.org/10.3897/nucet.4.29858>

Abstract

The article considers the issues of converting the ENDF format systems of constants to relational databases. This conversion can become one of the tools facilitating the development and operation of factual information, techniques and algorithms in the field of nuclear data and, therefore, increasing the efficiency of the corresponding computational codes. The work briefly examines an infological model of ENDF libraries. The possible structure of tables of the corresponding relational database is described. The proposed database schema and the form of tables take into account the presence of both single and multiple properties of the isotopes under consideration. Consideration is given to the difference in organizational requirements for transferring constants from relational tables to programs and performing a visual analysis of data in tables by a physicist-evaluator. The conversion algorithms and results are described for the ROSFOND-A and ENDF/B-VII.1 libraries. It is shown that performing calculations directly in the DBMS environment has its advantages in terms of simplifying programming and eliminating the need to solve a number of problems on data verification and validation. Possible approaches are indicated to ensure operation of inherited software together with nuclear data libraries in the relational format. Some terminological refinements are proposed to facilitate constructing an infological model for ENDF format. The conversion programs and the ENDF/B-VII.1 library in the relational format are available on a public site.

Keywords

ENDF libraries, conversion, relational format.

Introduction

Much attention has always been paid to the issues of organizing machine-readable data for neutronic calculations (see, for example Kolesov and Nikolaev 1972, Parker 1963, Woll 1968, Drake 1970, Nikolaev et al. 1984, Pronyaev et al. 2001, MacFarlane and Muir 1996, Larson 2007, ENDF-6 2009, Mattoon et al. 2012, Abramovich et al. 2001, Zizin et al. 1974, Sinitisa and Rineiskiy

1993, Koshcheev et al. 2000, Plyaskin and Kosilov 2002, Manturov et al. 2000, Zhuravlev et al. 2009, Koshcheev et al. 2014, Manturov and Nikolaev 2016). Data location in a library of constants (data format) significantly affects the calculation efficiency, determining the speed and, at times, the accuracy of calculations. Analyses of neutronic calculation programs show that up to 60% of the code is intended for the implementation of data management functions. The way data are organized is

* Russian text published: *Izvestiya vuzov. Yadernaya Energetika* (ISSN 0204-3327), 2017, n.4, pp. 138-150.

also important for physicists-evaluators, making possible operative sampling, taxonomy, visualization and comparison of data.

The most common text file format for evaluated nuclear data files (ENDF) became a *de facto* standard for historical reasons. It reproduces the punch-card and tape data organization, which determines the order of access to information and introduces appropriate restrictions on the style of programming and data processing. Efforts aimed at developing and standardizing nuclear data formats in traditional text file technologies do not lead to radical solutions in terms of reducing the data management cost, since the emergence of new experimental information and the needs of applied problems necessitate introducing new types of data that require format changes and new specialized software (Manturov et al. 2000, Zhuravlev et al. 2009, Koshcheev et al. 2014, Manturov and Nikolaev 2016).

The volume of factual information, techniques and algorithms in the field of nuclear data is very large now. Therefore, the task of finding means to simplify the development and operation of this information array is very urgent. Using a technology, where the search, retrieval and updating of information operations are standardized invariantly to the nature of data, would help reduce the labor of programmers, focus on the functional (applied) part of the calculation codes, and improve readability (self-documentability) and program verifiability. Currently, such a technology is provided by relational database management systems (DBMS). One of the main concepts of this approach is that the physical and logical organizations of data are separated. Centralized database management provides standard low-level data operations, eliminating the need to program these operations in specific applications.

The paper briefly describes the concept and technology as well as the programs and results of converting the ROSFOND-A and ENDF/B-VII.1 libraries to relational format. The conversion programs and ENDF/B-VII.1 library in the relational format are available on the public site: <http://178.215.91.20/nd>.

Prospects of using relational DBMS in neutronic calculations

There is a positive experience of using relational databases in libraries of nuclear constants (Boboshin et al. 1994, Boboshin et al. 1999, Boboshin et al. 1999a, Varlamov et al. 2001, Varlamov and Ishkhanov 2017). However, it seems that the DBMS technology, which provides “standard tools for solving standard tasks” of data management, should be used more widely. The metaphor of a table for representing data in a program is a very powerful tool that determines the style and effectiveness of programming. A tabular form of relational databases is much better suited to the task of organizing variant calculations (as well as verification and validation tasks (Yuferov et al. 2013b)) than traditional data structures.

Storing data in a relational database eliminates the need for an explicit allocation and checking of control information (separating records, navigation parameters, data type indicators, counters, flags, etc.). Correct data positioning is provided based on the *information principle of object descriptions*, according to which each object is assigned a table row, i.e., a unique tuple of pairs <object property name, property value>. This eliminates the task of developing special manuals for organizing and formatting data as well as saves time for studying control information and practicing its correct application in processing programs.

The structure of relational tables is determined only by the nature of specific physical information. Table columns store homogeneous data, for example, total cross-sections, and each entry (table row) is identified (in this case) by the corresponding energy value. Adding records (for example, when expanding or specifying an energy interval) is a standard operation that does not require correction or inclusion of any control information. Similarly, adding columns (for example, with data uncertainty estimations) does not necessitate changing the query modules that retrieved data from the table of the original structure.

When performing calculations directly in the DBMS environment, programming is simplified as follows:

- the stages of data input and processing are divided;
- the task of formatting input/output data is excluded;
- the validity of source data is verified outside the processing modules at the time of online input into the tables;
- data addressing is performed “by name” without monitoring their actual allocation in the external or random access memory;
- optimized search algorithms in the DBMS significantly speed up data sampling;
- sequences of keystrokes and queries can be recorded and saved for repeated execution or for insertion into the program text;
- database tables for source data and results can be used in modern programming languages based on standard data access technologies.

There are no fundamental problems and shortcomings caused by transferring libraries of constants to the environment of relational databases. Back in 2001, the report (Pronyaev et al. 2001) formulated the goals and ways of converting all libraries, maintained by world nuclear data centers, to the relational format. It was assumed that the nuclear data centers will develop standard database diagrams, the Java language will be adopted as a common programming language and in about five years the nuclear data libraries will migrate to the format of relational tables. As a result, both local and remote access was provided through a single interface to both data and processing facilities. However, for various reasons, the declared goals have not been fully achieved to date. Thus, when developing the seventh version of the ENDF/B library (2006), it was stated that the transition to relatio-

nal formats would be advisable, but the proposed changes were considered premature due to considerable labor costs (ENDF-6 2009). Therefore, ENDF format remains the main form of storing information in evaluated neutron data libraries.

Converting ENDFs to relational format represents a problem not due to the great complexity or even the complex branched structure of the data in question, but because of the many widely used programs oriented to processing data in ENDF format. Nevertheless, the objective need to provide diverse works with very large arrays of textual and numerical information on nuclear physics promotes the development of new formats designed to standardize and unify data access. In particular, the development of a unified nuclear data format (GND – Generalized Nuclear Data (Mattoon et al. 2012)) is completed based on the extensible markup language (XML). The CINDA bibliographic databases and the EXFOR experimental databases are converted to the relational format (Varlamov et al. 2001, INDC(NDS)-0614 2012). Work is underway to transfer the files of ENDF-formatted data and group constants for neutronic calculations to local and client-server DBMS (Fan et al. 2005, Alekseev et al. 2012).

Constructing an infological ENDF model

An infological model describes the subject area in some standard terms and notations for later mapping this description into a relational database schema, i.e., a list of specially structured tables and their relationships.

To describe the database schema for ENDF libraries is rather difficult due to some terminological contradictions with the standard notions of informatics. In the ENDF system, constants are grouped according to materials as text files with lines that formally refer to two structural levels. The major structural element of such a file is also called ‘file’ (in the ENDF system). Therefore, for definiteness, a text file at the library level (at the level of the operating system) will be referred to as a *material file*, and the top-level section in a material file as an *internal file*.

An internal file is treated as a section with “data of a particular class”. However, the concept of a ‘class’ (implying a fixed list of predicates, i.e., conditions and attributes characterizing a certain class) is not strict here – the first lines of the section contain some header information (in particular, the ID of an internal MF file), supplementing the main content of the section. This content is divided into sections, i.e., second-level structural elements, which (in ENDF terminology) “describe a certain type of data”. The main content of a section is also preceded by the header entries. Thus, there are actually four structural elements: the header and the first-level data, the header and the second-level data. The headers contain identifying or controlling information and some specific material pa-

rameters. In the lines of both headers and sections for data placement, ten fixed-length fields are allocated:

P1, P2, P3, P4, P5, P6, Material, File, Section, Line,

of which the last four – Material, File, Section, Line – are standard, and the first six store various information content. The semantic content of neighboring lines can be different. This is the difference between the ENDF format and the relational model, in which the meaning of all records (lines) of the table is identical and determined by the list of properties considered for a certain class of objects represented by table entries. Values of the properties of a particular object are stored in the record fields associated with the corresponding table columns.

This subject area contains three main classes: MATERIALS, INTERACTIONS (REACTIONS), and DECAYS. All other information can be treated as a set of properties of the specified entities. Allocation of other entities in the subject area, i.e., some subclasses, for example, ISOTOPES or RESONANCE INTERACTIONS, is caused by practical conveniences of operating relational tables. Thus, the resulting infological model for a file of resonance parameters includes the following entities:

- a class of materials;
- a class of isotopes;
- a class of interactions identified by an energy interval of incoming neutrons;
- a class of interactions, the determining attribute of which is the orbital angular momentum value;
- a class of interactions identified by the target nucleus spin;
- a class of resonant interactions (resonances).

The list of classes defines a database schema. Each class has a corresponding table with recorded scalar characteristics of the class instances, i.e., material, isotope, interaction, and resonance.

Organizing tables of constants in relational databases

Data are distributed throughout the tables according to the nature of their properties. A property can be *single* (scalar, e.g., a mass number) or *multiple* (e.g., a vector for counting the cross-section energy dependence). All single properties inherent in all materials can be placed in one table. Specific properties characteristic of some materials (e.g., fissile materials) are grouped into separate tables to exclude gaps in the table of general properties.

Multiple properties are expressed by functional dependencies (temperature, energy). If the values of a function are taken for all materials at the same points of the argument, it is advisable to treat these points as single properties (e.g., a cross-section in a given energy group), assigning a single column of the table to each point of

the argument (e.g., to a group). As a result, the table is filled as densely as possible. For example, such a table can store all the main group cross-sections. In this case, the first column is reserved for the material name, the second column – for the section type, and the remaining column – for the corresponding group values. Currently, the permissible number of table columns can be several thousands; therefore, such storage is possible for a multi-group presentation of constants.

The described “horizontal” arrangement of groups (each group in a separate column) is convenient for a one-time sampling of all data on the energy distribution of cross-sections – the contents of the current record is transferred to an array by one COPYTOARRAY instruction. On the other hand, according to the basic concept of arranging tables (one column for one property), “automatic” output to the graph is carried out column by column. Therefore, in order to promptly put out the energy distribution to the graph, the groups should be placed “vertically”, i.e., data of a particular group are in a separate record, and the column sample gives cross-sectional values for all the groups.

These two processing tasks are typical; therefore, it is advisable to store the tables in two forms, i.e., with vertical and horizontal records of energy dependencies. In terms of maintaining the databases and the amount of necessary software support, this approach is not burdensome.

If the sets of argument values are not the same for different materials, the functional dependence can be stored only “vertically”, allocating for it two columns: the first one stores the argument values, the second one contains the functions. If such pairs are placed contiguously, then there will be gaps in the table due to the different number of values in the sets. If they are placed consecutively, there will be no gaps, but it will be necessary to provide a field for the dependency name.

In ENDF libraries, many efforts are made to uniquely identify the material by means of four-digit numbers. This is due to the currently artificial “punch card” limitation of the recording length by 80 characters. It is advisable to use the common chemical notation to identify, for example, hafnium isomers: these are the identifiers 178Hfg, 178Hfm2 and 178Hfn or a form of Hf-178-m2 type that is more convenient for visual perception and computer processing. Adding the element symbol eliminates the need to memorize the mass numbers and charges to identify materials, especially since conditional numbering is still used for compound materials (water, zirconium hydride, etc.). All used names are stored in the reference table associated with the material identifier field. This field is filled by selecting the required value from the reference table, which excludes the possibility of typing a random incorrect name.

The total number of tables is determined by the actual number of “material-internal_file-section” combinations and the presence of single and (or) multiple properties in the section. Thus, the ROSFOND-A library contains

301 files of materials with internal files of seven types: MF=1 (general information), MF=2 (resonant parameters), MF=3 (reaction cross-sections), etc. In addition to physical data, each material file contains historical, navigational and statistical data. The practice in operating DBMS shows that such data should be stored in separate tables. In total, the ROSFOND-A library has 4664 “material-file-section” combinations, each of which can be designed as a “material-property” table. Such a maximally fractional presentation can be convenient for evaluators analyzing a particular reaction.

To support the calculation tasks, larger file-section groupings are preferred: their number in the ROSFOND-A library is 70. The table of existing “file-section” groupings is included in the database as navigation data. The main sections of these groupings combine data that are homogeneous in terms of physical meaning and can be represented by separate tables. Thus, all 25 sections of File 3 contain energy dependences of cross-sections and derived quantities. Therefore, all the information, even when specifying an individual dependence of each cross-section on energy, can be placed in one table. On the other hand, the “2-151” grouping (File 2 containing a single Section 151) includes parameters of resolved and unresolved resonances. The structure of this information is rather complicated and, therefore, it is advisable to use several tables.

In general, information of each grouping is divided into two types of tables with data on single and multiple material properties, respectively. For convenience of analyzing and selecting operational data, the tables are stored in two formats: “material-all_properties” and “property-all_materials”. For correlating with ENDF files, the name of the “property-all_materials” table is MF_MT, i.e., it is formed from the ID of the internal MF file and the identifier of the MT section. For example, the “03_001” table contains complete neutron cross-sections of all materials. Similarly, the table name of the “material-all-property” type is formed as ZZAA_MF from the ZZAA material cipher and the ID of the internal MF file. Over time, this conditional identification of tables should be replaced with meaningful reasonable naming, which will allow non-specialized users to navigate in the database schema and provide some self-documentation of processing programs.

Procedure for converting ENDF libraries to relational databases

Although ENDF format uses a large number of different flags and indicators, specific information units within sections do not have any special features. The required unit can be found only by reading all the records in the section and calculating its initial line by the indicators of lines allocated for information units. If there were individual unit characteristics, converting ENDF libraries to relational format would be a trivial task.

ENDFs can be converted to relational tables by various means. This conversion is actually a one-time operation; therefore, there is no need to design a special conversion program stored as an .EXE module. It is more appropriate to use the built-in scripting language of a given DBMS, performing the conversion step by step. It is advisable to import the source text files of a nuclear data library line-by-line into the database “as is”, and then to select step-by-step the necessary fragments of lines, creating new tables or columns for their placement. The relational tables make it possible to visually evaluate the results of each stage and decide on the path of further conversions.

The scenario for converting ENDFs to a relational database includes the following steps:

1. ENDFs are imported line-by-line into relational tables (summary or individual for each file) as text lines.

2. The tables are restructured to extract the main navigation properties {Isotope}, {File}, {Section}, {Line} from the line and six information fields provided by ENDF format, which, for convenience, are referred to simply as {1}, {2}, {3}, {4}, {5}, {6}. Thus, the stages of structural and semantic analyses of lines are divided. To visually control the correctness of conversion, six information columns are accompanied by identification columns, in the fields of which ENDF flags are written at the data semantic analysis stage.

3. From the tables obtained in Stage 2, tables are selected that combine the data of sections across all materials, for example, the “02_151” table contains data of the MT=151 section from the MF=2 file of resonance parameters.

4. Navigation information tables for sections are created and filled: for example, the “List_ER” energy table, in which all the scalar parameters (flags and physical quantities) characterizing the energy interval are stored (for example, in different representations of resonance parameters).

5. Information tables are created and filled for a section. Thus, the “SLBW_MLBW” table stores the scalar parameters of resolved resonances in the single-level and multilevel Breit-Wigner approximations. If necessary, all the scalar parameters of different representations of resonances can be placed in a single table. This possibility is limited only by the permissible number of table columns in a particular DBMS.

- 5.1. Information tables are created and filled for energy-dependent parameters. Tables of such functional dependencies have a simple structure of one or more tuples of the form

{Isotope}, {Energy Interval No.}, {Energy}, {Parameter}, {Interpolation Scheme}.

It is noteworthy that even the second of the listed steps would solve the conversion problem, which is understood as providing direct logical access to data by their names, if all information units had semantic indicators or flags

of the unit heading and end. However, the ENDF format provides only availability flags and data quantity indicators, offering only sequential access. That is why the conversion task turns out to be non-trivial, requiring that an algorithm be developed for a sequential analysis of all the ENDF lines to determine the data availability and quantity flags.

To ensure verification and control of data, the created relational tables first save all the ENDF navigation parameters. Conversion scripts use these parameters to search for information units in the summary table and generate detail tables. At the stages of further restructuring, the database schema is optimized: in particular, the flag fields are excluded. For example, it is not difficult to see that the LRU flag indicating which resonances are described in a given energy interval can be eliminated, since all situations can be indicated by the LRF flag if the set of its values is supplemented with the following values:

LRF=0, only the scattering radius is determined;

LRF=8, unresolved resonances are described in the current energy interval, with only mean fission widths depending on the energies;

LRF=9, unresolved resonances are described in the current energy interval, the mean distances between the level, the widths for competing reactions, the average reduced neutron widths, the radiation widths, and the mean fission widths depend on the energy.

A complete normalization that meets all the theoretical requirements for eliminating redundancy is not done, when operating a database, so as to avoid creating complex queries that slow down data sampling.

Materials illustrating the stages and results of implementing the scenario presented above (for ROSFOND-A and ENDF/B-VII.1-neutrons libraries) are given in (Yuferov et al. 2013b, Yuferov 2011, Yuferov et al. 2013, Yuferov et al. 2013a) and on the site: <http://178.215.91.20/nd>.

Conclusion

1. Today, relational DBMS are the only unified means for working with large amounts of data. We can talk about the emergence and formation of a new discipline, *nuclear informatics*, based on the DBMS technology. Alternative technologies should necessarily reproduce or borrow the functionality of relational DBMS. In particular, spreadsheets are limited in functions such as group search and sampling, and XML technology (Mattoon et al. 2012, Holzner 2002) can only be considered as a low-level tool for storing or sharing data.
2. There are no fundamental problems with nuclear data migration to the environment of relational databases. Placing nuclear data in a relational database can be considered as a form of presentation of the original format and applied along with the latter. Due to this,

it is possible to preserve the accumulated software tools and apply the DBMS technologies for organizing calculations in both existing and new neutronic calculation complexes. This provides additional means to verify the libraries and codes.

3. It is not definitely necessary to develop standard schemas for databases, the need for which was noted in the report (Pronyaev et al. 2001). On the one hand, the “standard” is defined by the logic of nuclear data interconnection and subordination but, on the other hand, the variety of possible tasks does not allow us to determine the only optimal DB schema. Using a relational model, it becomes possible to easily adapt the available database to any newly created problem by selecting and arranging the necessary data or by restructuring the tables. Certain standardization should be envisaged only to facilitate replication and

synchronization of databases of world nuclear data centers.

4. Also, it is not necessary to use a common programming language for nuclear data libraries (the report (Pronyaev et al. 2001) suggested the Java language). The current situation is such that the language can and should be selected “for the task” and any language can use standard technologies of local or remote access to databases.
5. The structure of tables should be dynamically optimized for a specific calculation task. The variability of the tabular layout requires comparison of possible database schemas by performing computational experiments to evaluate the efficiency of working with relational presentations of ENDFs by the criteria of visibility, convenience of analyzing, and data sampling rate for various tasks and software packages.

References

- Abramovich SN, Gorelov VP, Gorshikhin AA, Grebennikov AN, Ilin VI, Krutko NA, Farafontov GG (2001) The TENDL Neutron Data Library and the 38-group neutron constant system TENDL038. VANT. Ser. Yadernye konstanty 1: 11–26. [in Russian]
- Alekseev AV, Bnyatov AV, Kasatkin SS, Krutko NA (2012) Software complex GROUND of constant supply with nuclear physical data. The thesis of the industry seminar “Neutronics-2012”. Obninsk: 33–34. [in Russian]
- Beck B, Hedstrom GW, Hill TS, Marchetti AA, McNabb DP (2006) ASCII format specifications for the Evaluated Nuclear Data Libraries (ENDL). Technical Report UCRL-TM-218475, LLNL.
- Boboshin IN, Varlamov AV, Varlamov VV (1999) Web-server CDFE INPI MSU: databases of nuclear-physical data in hypertext representation. VANT. Ser. Yadernye konstanty (2): 99–110. [in Russian]
- Boboshin IN, Varlamov VV, Trukhanov SK (1994) NESSY. Relational database on the structure and decays of cores on IBM / PC. VANT. Ser. Yadernye konstanty (3-4): 75–81. [in Russian]
- Boboshin IN, Varlamov VV, Varlamov AV, Rudenko DS, Stepanov ME (1999a) The Centre for Photonuclear Experiments Data Nuclear Data Bases. Preprint NIIYaF MGU 99-26/584, 24 pp. [in Russian]
- Drake MS (1970) Data Formats and Procedures for the ENDF Neutron Cross Section Library. BNL-50274.
- ENDF-6 (2009) Formats Manual, Data Formats and Procedures for the Evaluated Nuclear Data File ENDF/B-VI and ENDF/B-VII, BNL-90365-2009, edited by M.W Herman and A. Trkov, revised by M.W. Herman, A. Trkov and D.A. Brown (Dec. 2011).
- Fan TS, Huang G, Feng YG, Ye WG, Liu WL, Hong YJ, Song XX, Liu TJ, Chen JX, Tang GY, Shi ZM, Guo ZY (2005) Study on Relational ENDF Databases and Online Services. Atomic Energy Science and Technology 39 (1): 28–31.
- Holzner S (2002) XSLT. The programmer’s library. Piter Publ., Saint Petersburg, 544 pp. [in Russian]
- INDC(NDS)-0614 (2012) Further Development of EXFOR Summary Report of the Consultant’s Meeting. INDC International Nuclear Data Committee. IAEA Headquarters, 6–9 March, Vienna, Austria.
- Kolesov VE, Nikolaev MN (1972) The format of the library of recommended data for the calculation of reactors. VANT. Ser. Yadernye konstanty 8(4): 3–66 [in Russian]
- Koshcheev VN, Manturov GN, Nikolaev MN, Cibulya AM (2014) Library of group constants BNAB_RF for reactor calculations and protection. Izvestiya vuzov. Yadernaya energetika 3: 83–101. [in Russian]
- Koshcheev VN, Nikolaev MN, Korchagina ZhA, Savoskina GV (2000) Library of the estimated neutron data FOND-2.2. VANT. Ser. Yadernye konstanty 2: 40–70. [in Russian]
- Larson NM (2007) SAMMY User Guidance for ENDF Formats. ENDF367, ORNL/TM2007/23, Oak Ridge National Laboratory, Oak Ridge, TN.
- MacFarlane RE, Muir DE (1996) NJOY94.61: Code System for Producing Pointwise and Multigroup Neutron and Photon Cross Sections from ENDF/B Data, PSR-355, Los Alamos National Laboratory, Los Alamos, New Mexico).
- Manturov GN, Nikolaev MN (2016) Priorities for the development of systems for the constant supply of reactor calculations and protection. Izvestiya vuzov. Yadernaya energetika 2: 133–142. [in Russian]
- Manturov GN, Nikolaev MN, Tsibulya AM (2000) CONSYST code for neutron constants preparation. Scope statement: IPPE Preprint -2828. FEI Publ., Obninsk, 41 pp. [in Russian]
- Mattoon CM, Beck BR, Patel NR, Summers NC, Hedstrom GW, Brown DA (2012) Generalized Nuclear Data: A New Structure (with Supporting Infrastructure) for Handling Nuclear Data. Nuclear Data Sheets 113(12): 3145–3171. <https://doi.org/10.1016/j.nds.2012.11.008>
- Nikolaev MN, Ryazanov BG, Savoskin MM, Cibulya AM (1984) Multigroup approximation in the theory of neutron transport. Energoatomizdat Publ., Moscow, 256 pp. [in Russian]
- Parker K (1963) The Aldermaston Nuclear Data Library. AWRE Rep. 0-70/63.
- Plyaskin VI, Kosilov RA (2002) Reference-information interactive systems of nuclear physical data for various applications of nuclear physics. Energoatomizdat Publ., Moscow, 375 pp. [in Russian]

- Pronyaev V, Winchell D, Zerkin V, Muir D, Arcilla R (2001) Requirements for the Next Generation of Nuclear Databases and Services. *J. Nucl. Sci. Technol.*, suppl. 2: 1476–1479. [Proc. Int. Conf. on Nuclear Data for Science and Technology, 7–12 October, Tsukuba, Japan]
- Sinitsa VV, Rineiskiy AA (1993) GRUCON – A Package of Applied Computer Programs, Rep. INDC(CCCP)-344, IAEA, Vienna.
- Varlamov VV, Ishkhanov BS (2017) Nuclear Databases in Scientific Research. <http://nuclphys.sinp.msu.ru/bd/index.html> [accessed 13 Feb. 2017; in Russian]
- Varlamov VV, Ivanov SV, Peskov NN, Stepanov ME (2001) A new search engine for experimental data on nuclear reactions from the international system EXFOR. Proc. of the III All-Russian Conference on Electronic Libraries RCDL'2001. *Electronic Libraries: Advanced Methods and Technologies, Digital Collections*. Petrozavodsk, 11 – 13 sent., KarNCz RAN, 166–173. [in Russian]
- Woll D (1968) Card Image Format of the Karlsruhe Evaluated Nuclear Data File KEDAK. KFK880, EANDC-E-112 «U», EUP-4160e.
- Yuferov AG (2011) Converting the ROSFOND library to a relational database: IPPE Preprint-3194. FEI Publ., Obninsk, 28 pp. [in Russian]
- Yuferov AG, Komissarov SA, Nikolaev MA (2013a) Infologic model of the file of resonance parameters: IPPE Preprint -3233. FEI Publ., Obninsk, 40 pp. [in Russian]
- Yuferov AG, Komissarov SA, Nikolaev MA (2013b) Relational model of the file of angular distributions: IPPE Preprint -3235. FEI Publ., Obninsk, 20 pp. [in Russian]
- Yuferov AG, Nikolaev MA, Komissarov SA (2013c) Verification and validation of the file of resonant parameters in the relational format: IPPE Preprint -2828. FEI Publ., Obninsk, 20 pp. [in Russian]
- Zhuravlev VI, Voronkov AV, Brykin AN, Sinicza VV, Zemskov EA, Kalchenko VV (2009) New features of GNDL are the system of constant provision of reactor calculations and protection. *VANT. Ser. Obespechenie bezopasnosti AES*, (24): 29-37. [in Russian]
- Zizin MN, Zagaczkiy BA, Temnoeva TA, Yaroslavceva LN (1974) Automation of reactor calculations. Atomizdat Publ., Moscow, 103 pp. [in Russian]