

KAERI/RR-1799/97



KR9800545

원자력 기반 연구

# 핵자료 데이터베이스 개발

Development of Nuclear Data Base System

*R*

한국원자력연구소  
KOREA ATOMIC ENERGY RESEARCH INSTITUTE

29 - 41

## 제 출 문

소장 귀하

본 보고서를 기관고유사업 “원자력 기반연구” 과제 (단위과제 “핵자료  
데이터베이스 개발”)의 최종보고서로 제출합니다.

1998 년 1 월

과제책임자: 장 종 화

책임연구원: 김 정 도

선임연구원: 오 수 열

길 충 섭

조 영 식

박 수 진

책임기술지원: 조 인 행

감 수 위 원: 지 성 균

## 요 약 문

### I. 제 목

핵자료 데이터베이스 개발

### II. 연구의 목적 및 중요성

핵자료는 신형로의 개발, 핵연료의 개량, 방사성 폐기물 처분, 핵융합연구, 동위원소의 응용, 핵의학, 환경연구등 원자력을 이용하는 모든 분야에서 필요한 원자핵의 특성 및 반응 특성자료이다. 핵자료는 1 차적으로 실험에 의해 생산되나 실험자료의 정밀성, 범위 등에 한계가 많고 오차가 크므로 실용을 위해서는 실측된 자료를 이론모형을 통하여 보완하여 평가한다. 또한 평가된 자료를 노심 핵설계, 차폐계산 등에 사용하기 위해서는 균정수의 형태로 처리하고 적분 실험 결과에 맞도록 보정하여야 한다.

본 연구의 목적은 원자력을 이용하는 모든 분야에서 사용하는 핵자료를 이용자들이 쉽고 편리하게 이용할 수 있도록 전산 데이터베이스 형태로 보관 유지하고 국내외 자료를 수집 및 배포하여 관련 연구자가 최신 정밀 핵자료를 이용할 수 있도록 하는 것이다.

### III. 연구의 내용 및 범위

본 연구에서는 국내외 연구자들이 핵자료를 이용하기 위한 기반인 핵자료 데이터베이스를 구축하여 온라인으로 사용할수 있도록 하기 위하여 다음과 같은 연구를 수행하였다.

#### - 핵자료표 웹서버 개량

인터넷에 온라인으로 서비스되는 핵자료표와 단면적도형 작성 웹서버를

일부 개량하였음.

- 핵구조자료 DB 구축

1997 연도말의 최신 ENSDF 평가자료로 교체하였음.

- 핵반응자료 DB 구축

핵자료 측정자료인 EXFOR 데이터베이스를 도입하고, 검색시스템을 작성하였으며, 각종 평가핵자료 라이브러리를 수집하고 검색 프로그램을 작성하였음.

- 균정수 생산 및 검증

열 중성자용 MATXS 균정수 라이브러리를 생산하여 U-235 노심에 대한 검증계산을 수행하였고 소내의 요구에 따라 MCNP 용 단면적을 생산하였음.

#### IV. 연구결과의 활용에 대한 건의

본 연구결과 수집된 핵구조자료, 핵반응자료, 평가핵자료 데이터베이스는 온라인 웹서버를 통하여 전세계의 사용자에게 제공되고 있다. 또한 MATXS형 라이브러리는 DANT 계열의 중성자 수송코드의 균정수 라이브러리로 사용되며 원자로의 차폐계산, 방사선 재료 손상해석에 활용된다. 또한 MCNP용 연속에너지 라이브러리는 소내의 이용자들의 요구에 따라 계속적으로 보완되고 있다.

## SUMMARY

### I. Project Title

Development of Nuclear Data Base

### II. Objective and Importance of the Project

Nuclear data are the characteristic and interaction properties of the nuclides. These data are necessary for the fields such as the development of nuclear reactor, the upgrade of nuclear fuel, the research on fusion engineering, the application of radio-isotopes, the nuclear medical application, the study of environment, etc. Nuclear data should be measured in principle, however, the measured nuclear data are usually not complete and lack of accuracy for practical use. So the evaluation is needed with help from the nuclear physics theory. The evaluated nuclear data are further processed into the group constant libraries for specific application computer codes for nuclear design and shielding calculation. The generated group constant library is verified and benchmarked against the integral experiments.

The purpose of this study is to establish an online computerized database. The collected up-to-date and reliable nuclear data are distributed to the end users through easy and convenient web services.

### III. Scope and Contents of Project

Following scope of tasks are performed to establish the nuclear data

base which is the infra structure for the nuclear related research and industry.

- Upgrade of "Table of Nuclides" web service and addition of new feature to existing ENDF plotting services.

- Establishment of nuclear structure data base.

  - Reconstruction of nuclear srtructure (ENSDF) data base with up-to-date data.

- Establishment of nuclear reaction data base.

  - Collection of the experimental reaction database (EXFOR),

  - Collection of the evaluated nuclear reaction data libraries, and

  - Development of data retrieval programs.

- Group constant library generation and verification

  - Generation of MATXS group constant library for thermal reactor application and verification against U-235 cores, and

  - Improvement of the cross section library preparation system for the MCNP code.

#### IV. Results and Proposal for Applications

The nuclear structure, reaction and evaluated data file data bases collected during this study are ditributed through the internet online web system over the world. The MATXS-type library is useful for users of the neutron transport code system DANT, which is a powerful tool to analyses of the shielding and radiation damage of nuclear reactor. The continuous energy library for MCNP is developed to meet the users' needs in Korea.

# 목 차

제출문	
요약문	
SUMMARY	
목차	
표목차	
그림목차	
제 1 장 서론	1
제 2 장 핵자료표 웹서버 개량	3
제 3 장 핵구조자료 DB 구축	5
제 1 절 ENSDF 취득 프로그램	5
제 2 절 평가 핵구조자료 수집	7
제 4 장 핵반응자료 DB 구축	8
제 1 절 EXFOR 파일의 구조	8
제 2 절 EXFOR 수집	12
제 3 절 EXFOR 검색 프로그램	15
제 5 장 평가핵자료 DB 구축	16
제 1 절 평가핵자료 DB 구축	17
제 2 절 평가핵자료 수집	20
제 6 장 균정수 생산 및 검증	22
제 1 절 열중성자용 MATXS형 균정수 생산	22
제 2 절 MCNP 라이브러리 생산	26
제 3 절 MATXS69 라이브러리 검증	27
제 7 장 결론	36
참고문헌	37
부록 1. 평가핵자료 검색프로그램 source list	39
부록 2. MATXS69 색인	61

## 표 목 차

표 2.1	핵자료 웹서버 사용량 통계 (1997년도) . . . . .	4
표 4.1	핵반응 실측 자료 편집 기관 . . . . .	8
표 4.2	우리 나라의 핵자료 측정 기관 코드 . . . . .	12
표 4.3	우리 나라의 문헌 약어 코드 . . . . .	12
표 4.4	우리 나라에서 보고된 EXFOR 자료 . . . . .	13
표 5.1	sublibrary 번호 분류 . . . . .	16
표 5.2	ENDF-6 형식에 지정된 MF 번호 . . . . .	19
표 6.1	MATXS69 열중성자로용 라이브러리를 이용한 유효중배계수의 비교	35

## 그 립 목 차

그림 4.1	EXFOR 파일의 예 . . . . .	11
그림 4.2	EXFOR 파일 보관 디렉토리 구조 . . . . .	14
그림 4.3	색인 파일 index/ind11 의 앞부분 . . . . .	14
그림 5.1	평가핵자료 디렉토리 구조 . . . . .	17
그림 6.1	NJOY코드에서 MATXS 라이브러리 생산을 위한 자료 흐름도	29
그림 6.2	MATXS69 열중성자로용 라이브러리 생산을 위한 NJOY . . 입력 자료	30
그림 6.3	MCNP 라이브러리 생산을 위한 NJOY 입력자료 . . . . .	32
그림 6.4	MCNP 열중성자 자료 생산을 위한 NJOY 입력자료 . . . . .	33



## 제 1 장 서 론

핵자료는 신형로의 개발, 핵연료의 개량, 방사성 폐기물 처분, 핵융합연구, 동위원소의 응용, 핵의학, 환경연구등 원자력을 이용하는 모든 분야에서 필요한 원자핵의 특성 및 반응 특성자료이다. 현재까지 알려진 핵종만도 원자번호 111, 질량수 273 까지의 2930 종이다. 이 중 안정원소는 249 종, 1년 이상 반감기를 갖는 핵종이 124 종, 1 일 이상 300 종, 1 시간 이상 550 종이며, 이 중 약 800 종의 핵종의 특성이 핵을 이용하는 분야에서 중요하다. 이 보다 짧은 반감기를 갖는 핵종 중에도 단면적등 특성을 알아야하는 핵종이 다수 존재한다. 실용적인 목적에 필요한 핵자료는 핵종의 질량, 반감기 또는 불안정핵종의 붕괴에 수반되는 알파, 베타, 감마의 붕괴에너지, 감마선의 특성에너지와 분율 등의 핵구조자료와 중성자, 광자, 양성자 등의 입사입자와의 핵반응자료이다. 핵반응자료는 입사입자에 의한 핵변환 단면적뿐아니고 입사입자와의 전단면적, 산란단면적, 핵분열단면적 등과 핵반응에 수반하여 방출되는 입자 (광자, 양성자 등)의 방출각도, 방출에너지분포 등도 포함한다. 핵분열, 핵파쇄 등의 반응에서는 파편핵종의 각, 에너지분포 등의 자료도 필요하다. 이러한 핵자료는 1 차적으로 실험에 의해 생산되나 실험자료의 정밀성, 범위 등에 한계가 많고 오차가 크므로 실용을 위해서는 실측된 자료를 이론모형을 통하여 보정하여 평가한다. 여기에서 이론모형만을 사용할 수 없는 이유는 핵력에 대한 정확한 지식이 아직 없고, 핵종내의 핵자간의 현상에 대한 지식이 부족하기 때문이다. 또한 평가된 자료를 노심핵설계, 차폐계산 등에 사용하기 위해서는 균정수의 형태로 처리하고 적분 실험 결과에 맞추어 보정하여야 한다.

본 연구의 목적은 원자력을 이용하는 모든 분야에서 사용하는 핵자료를 이용자들이 쉽고 편리하게 이용할 수 있도록 전산 데이터베이스 형태로 보

관 유지하고 국내외 자료를 수집 및 배포하여 관련연구자가 최신 정밀 핵 자료를 이용할 수 있도록하는 것이다.

본 연구에서는 국내외 연구자들이 핵자료를 이용하기 위한 기반인 핵 자료 데이터베이스를 구축하여 온라인으로 사용할수 있도록 하기 위하여 다음과 같은 연구를 수행하였다.

- 핵자료표 웹서버 개량
- 핵구조자료 DB 구축
- 핵반응자료 DB 구축
- 열 중성자용 MATXS 균정수 생산
- MCNP 용 단면적 생산 체계 개량

## 제 2 장 핵자료표 웹서버 개량

핵자료 웹서버는 1996년 발표한 이후 꾸준히 전세계에서 사용하고 있다 [2.1]. 핵자료 웹서버는 핵종의 질량, 붕괴 반감기를 수록하고 있으며, 일부 핵종의 증성자 반응단면적과 핵분열 생성률도 수록하고 있다. 기본적인 자료는 핵질량 평가 네트워크, 핵붕괴 네트워크, 일본 핵자료센터, 미국 핵자료센터등에서 평가된 신뢰성있는 자료를 채택하여 제공한다. 본 연구를 통해 수집된 자료를 일반 사용자들이 용이하게 사용할 수 있도록 가공하여 인터넷 온라인 시스템을 통해 제공하고 있으며 인터넷 웹 서비스는 "http://hpngp01.kaeri.re.kr/CoN/"으로 제공하는 핵자료표와 "http://hpngp01.kaeri.re.kr/CoN/endfplot.shtml"로 제공하는 온라인 단면적 도형 시스템으로 구성되어 있다.

본 연구기간 중에는 대부분의 프로그램을 개량하였으나 특히 다음 사항이 특기할 만하다.

### - 인용문헌의 표시:

각 자료의 출처를 화면에 작은 글씨로 표시하도록 하여 자료의 신뢰성을 확보하였다.

### - 최신 붕괴자료 획득:

각 핵종의 반감기, 붕괴모드 등에 대한 최신 평가자료를 미국 BNL의 핵자료센터(NNDC)에서 NUDAT 파일로 입수하여 제공하였다.

### - 단면적 도형에서 수치자료 제공 기능 추가:

온라인 단면적 도형 작성 프로그램인 ENDFPLOT를 수정하여 GIF와 postscript 형식의 파일만을 제공하는 기존 기능에 추가하여 수치자료도 표의 형태로 제공하도록 하였다.

1997년도의 웹서버 사용량은 표 2.1과 같다. 우리 연구소를 제외한 외부의 접속중에서 실제 자료 취득을 목적으로 사용한 건수는 총 19,682 건으로 매월 1,640 건에 해당한다. 이 통계는 미국 BNL 과 영국 ISIS의 미러사이트를 제외한 값이며, 1996년의 월평균 1,283 건에 비교하여 약 28 % 증가한 수치이다[2.1].

표.2.1. 핵자료 웹서버 사용량 통계 (1997년도)

종류	1 월	2 월	3 월	4 월	5 월	6 월	7 월	8 월	9 월	10 월	11 월	12 월	년 간
핵종표	712	632	895	741	693	926	838	700	802	1494	1422	1276	11,131
단면적도형	412	634	1,092	635	796	406	707	831	738	968	622	710	8,551
합 계	1,124	1,266	1,987	1,376	1,489	1,332	1,545	1,531	1,540	2,462	2,044	1,986	19,682

### 제 3 장 핵구조자료 DB 구축

핵구조자료는 핵종의 질량, 반감기, 에너지준위, 감쇄 감마선의 에너지, 분기율 등을 수록한 자료로 핵자료 평가 자체뿐 아니고 동위원소, 핵의학 등 분야에서 이용하는 핵종붕괴에 따른 감마선발생 등을 분석하기 위한 필수자료이다. 핵구조자료는 미국 BNL이 주관하는 Nuclear Structure and Decay Data Network를 통해 질량수별로 전세계의 연구기관이 분담하여 평가하며[3.1] 이 결과는 학술지 Nuclear Data Sheets와 Nuclear Physics B에 수록된다. 수치자료는 온라인으로 제공된다. 이 자료 DB는 전년도 연구 과제인 “핵자료 체계구축”과제에서 구축하였으나 본 연구 기간에는 이 자료를 갱신하고 취득 절차를 개선하였다.

#### 제 1 절 ENSDF 취득 프로그램

Evaluated Nuclear Structure Data File (ENSDF)는 미국 BNL의 전산기 "bnlnd2.dne.bnl.gov"의 온라인 서비스에 의해 제공되나, 질량수가 300여개가 되므로 각 질량에 대한 파일을 가져오는 작업은 매우 시간이 걸린다. 취득작업을 자동화하기 위해서 tcl 언어에 기반을 둔 expect를 사용한 자동 script "nndc.exp"를 작성하였다. 이 script는 질량수를 입력하면 자동적으로 "bnlnd2.dne.bnl.gov"에 login하여 ENSDF 온라인 취득 프로그램에 연결한 후 입력으로 지정한 질량의 파일을 ENSDF형태의 파일로 받는다. 취득한 파일은 FTP를 통하여 지정된 FTP 서버로 전송된다. 전송이 완료된 후에는 BNL NNDC에 남아 있는 파일을 자동적으로 삭제한후 logout한다.

이 절차를 통해 여러 개의 질량을 한꺼번에 처리할 수 있으며, 이때는 다음과 같은 명령처럼 질량을 지정할 수 있다.

nndc.exp 10 11 12

위의 작업 예는 질량 10, 11, 12 인 3 개 파일을 가져오기 위한 것이다. 파일의 이름은 M10.TXT, M11.TXT, M12.TXT로 정해진다.

이 절차를 사용함으로써 사람이 직접 개입하지 않고 주간 또는 야간에 파일을 원격지에서 쉽게 가져올 수 있다. NNDC 서버는 FTP를 이용하여 파일을 전송해주나, 우리 연구소의 경우에는 firewall의 안쪽에 있으므로 외부에서 파일을 전송할 수는 없다. 따라서 firewall 바깥의 워크스테이션으로 파일을 전송한 후, 다시 연구소 내부의 워크스테이션으로 도입한다.

NNDC의 전산기는 VAX이므로 파일이름은 모두 대문자가 된다. 또한 ENSDF를 보관하는 전산기인 hpngp01은 unix이므로 파일은 소문자로 보관하고 있다.

ENSDF 파일은 hpngp01의 /user4/kndc/ensdf에 질량수에 따라 파일이름 "mmm"로 보관한다. 새로 입수한 파일과 기존 파일을 비교한 후 차이점이 발견되면 새 파일로 대체해야 한다. 이 절차를 자동화하기 위하여 작성한 shell script는 replace.sh이다. 질량수를 파라미터로 사용하며 새 파일로 대체되는 경우에는 "Update mmm", 기존 파일을 사용하는 경우에는 "same mmm"으로 나타난다. 단, 대체할 Mmmm.TXT 파일이 없는 경우에도 "same mmm"으로 나타난다. 사용 방법은 아래와 같다.

replace.sh 10 11 12

사용자는 작업을 시작하기 전에 Mmmm.TXT 파일이 존재하는가를 검증하는 것이 필요하다.

본 연구에 의해 자동화된 작업절차를 요약하면 다음과 같다.

- BNL NNDC에서 ENSDF 파일 취득

nndc.exp 10 11 12

- 중간 FTP서버에서 hpngp01의 /user4/kndc/ensdf 디렉토리로

파일 가져옴.

- 기존 파일을 개선

replace.sh 10 11 12

이상과 같은 작업을 완료한 후에 여러 실패요인에 의하여 생성된 부산물은 Mmmm.TXT라는 명칭으로 NNDC 서버, 중간 FTP서버, hpngp01의 디렉토리에 잔류할 수 있으므로 확인하여 삭제하는 것이 좋다.

## 제 2 절 평가 핵구조자료 수집

ENSDF 자료는 미국 BNL에서 관리하여 다른 핵자료 센타로 배포하는 자료이다. 최신 핵자료를 확보하기 위하여 BNL의 온라인 서비스를 이용하여 자료를 취득하였다. 현재 원자번호 110, 질량수 269번까지 평가되었으며 총용량은 122 MB이다.

## 제 4 장 핵반응자료 DB 구축

세계 각국의 1000여 기관에서 실측된 핵자료는 학술지 등에 발표되지만, 이를 이용하기 위해서는 그래프 등을 읽어서 수치를 재현하여야 한다. 이러한 작업을 편집이라고 하며 매우 방대하고 지루한 작업이 될 수 있다. 따라서 국제협력을 통해 작업의 중복을 피하고 신뢰성있는 자료를 확보하기 위해서 표 4.1 처럼 전세계적으로 4 군데의 편집소에서 편집하여 전산화된 수치파일의 형태로 공개된다.

표. 4.1. 핵반응 실측 자료 편집 기관

센타명	담당 지역	운영 기관	코드
NNDC	미국 및 캐나다	미국 BNL	1,6,D
NEA/DB	OECD 국가 (상기국가 제외)	OECD	2,7,8,9,E,K,O,R
CJD	독립국가연합	러시아 IPPE	4,9,A,M
NDS	전세계 (상기국가 제외)	IAEA	3,S

이 편집핵자료는 각국 및 지역의 핵자료 센타간의 정보교류를 위한 전산화된 형태의 자료이며 EXFOR (Exchange Format)[4.1] 라고 불린다. 각 핵자료 센타는 일련번호의 중복을 피하기 위해 약정된 코드를 사용하여 EXFOR entry의 일련번호를 지정한다. 자료의 형태는 교환의 편의상 전산기와 인간이 모두 편하게 인식할 수 있도록 되어 있다.

### 제 1 절 EXFOR 파일의 구조

EXFOR 파일의 구조는 80 문자길이인 ASCII 텍스트의 집합으로서 각각 entry로 불리는 단위가 기본단위이다. 각 줄은 66행까지만 사용하며



67-71행은 entry번호, 72-74는 subentry번호, 75-79는 일련번호, 80행은 변경부호를 기록한다.

각 entry는 보통 한편의 논문에서 기술한 자료를 포함한다. 즉, 핵자료가 수록된 1 편의 논문에 1 개의 entry를 작성한다. 각 entry의 내용은 서지사항과 기술정보를 기록한 문자 부분자료와 수치자료이다. 수록된 정보의 종류에 따라 식별표지어를 사용하여 문자정보와 수치정보의 내용을 구별한다. 각 부분정보는 별도사전에 수록된 부호를 이용하던지 통상적인 의미가 있는 단어를 사용하여 기록한다.

각 entry는 핵종이나 핵반응종류에 따라서 나뉜 여러개의 subentry로 구성된다. 각 entry의 첫 번째 subentry (1 번)는 문자정보와 공통사항만 기록된다. 첫 subentry의 공통자료는 이후의 모든 subentry에서도 공통자료로 간주된다.

각 subentry는 문자정보, 공통사항, 수치자료의 순으로 배열한다. Subentry에 수록된 수치자료는 표(table)의 형태로 기록하며 각 항목은 11 자씩 차지한다. 각 표의 첫 줄에 항목에 대한 코드를 기록한다. 각 subentry 또는 전체 entry에 공통되는 자료는 공통 (COMMON) 사항으로 기록하며 수치자료처럼 표의 형태로 기록한다.

그림 4.1의 예에는 entry C0070의 일부를 보여 주고 있으며 72행 이후에 나오는 일련번호는 생략하였다. 구조는 “ENTRY - ENENTRY”의 쌍으로 둘러싸여 있으며 “SUBENT - ENDSUBENT” 쌍으로 둘러싸인 3 개의 subentry로 되어 있다.

첫번째 subentry는 “BIB - ENDBIB” 쌍으로 둘러싸인 문자정보와 “NOCOMMON”으로 공통자료가 없다는 내용로 되어 있다. 문자정보 영역에는 실측기관, 참고문헌, 저자, 문헌제목, 실측시설, 계측기, 관련문헌, 오차설정 방법, 자료인증상태, 자료 등록이력 등의 정보를 EXFOR dictionary에

등재된 검색어를 사용하여 기술한다.

두번째와 세 번째 subentry는 핵반응의 종류, 시료 등에 대한 문자정보 구역과 공통자료, "DATA - ENDDATA" 쌍으로 둘러싸인 수치자료의 영역으로 구성되어있다.

```

ENTRY          C0070          960311
SUBENT         C0070001      960311
BIB            10            14
INSTITUTE      (2SWDSWR)
REFERENCE      (R, NFL-6, 79)
AUTHOR        (E. RAMSTROM)
TITLE         Excitation functions of the  $^9\text{Be}(\alpha, n)^{12}\text{C}$ , the
               $^{13}\text{C}(p, n)^{13}\text{N}$  and the  $^{13}\text{N}(n, p)^{13}\text{C}$  reactions
FACILITY       (VDG) 5.5 MeV
DETECTOR      (BF3) 20BF3 counters in 4pi geometry
REL-REF        (N., RAMSTROM+, J, NP/A, 272, 259, 76) Details of detector
              system.
ERR-ANALYS    No information
STATUS        (APRVD) Approved by author 80-May-30
              Data taken from private comm., Ramstrom, 1980
HISTORY        (800326C)SOM
              (960305A) BIB update, converted to lower case.
ENDBIB        14            0
NOCOMMON
ENDSUBENT     17            0
SUBENT        C0070002      960311
BIB           4            5
REACTION      (4-BE-9(A,N)6-C-12, ,SIG)
SAMPLE        2mm thick Be disk
ADD-RES       (STRUC) No significant effects in the reaction due to
              the 10.753 and 10.818 MeV levels of carbon-13.
HISTORY        (960311A) REACTION corrected.
ENDBIB        5            0
NOCOMMON      0            0
DATA          2            73
EN            DATA
MEV           MICRO-B
0.445         8.8
.....
0.773         176.
ENDDATA       75            0
ENDSUBENT     85            0
SUBENT        C0070003      960311
BIB           4            8
REACTION      (6-C-13(P,N)7-N-13, ,SIG)
SAMPLE        Two different targets, 41.2 microgram/cm**2, and
              60.4 microgram/cm**2.
REL-REF        (D., GIBBONS+, J, PR, 114, 571, 59) Gibbons data are 30-50x
              lower than present work
ADD-RES       (RECIP)  $^{13}\text{N}(n, p)^{13}\text{C}$  cross section calculated from
              reciprocity theorem.
ENDBIB        8            0
NOCOMMON      0            0
DATA          2            86
EN            DATA
MEV           MB
3.2391        5.9
.....
4.250         44.9
ENDDATA       88            0
ENDSUBENT     101           0
ENDENTRY      C0070

```

그림 4.1. EXFOR 파일의 예

## 제 2 절 EXFOR 수집

EXFOR 색인에 등록된 우리 나라의 핵자료 관련기관은 IAEA에 보고되어 있으며 다음 표와 같다. 또한, 문헌 색인에도 원자력학회지와 물리학회지가 등록되어 있음을 확인할 수 있다.

표 4.2. 우리 나라의 핵자료 측정 기관 코드

기관약어	기 관 명	비 고
3KORKBU	구 경북대	
3KORKNU	경북대	
3KORKOR	기 타	
3KORKSR	한국표준연구원	
3KORKUS	고려대	
3KORNSU	서울대	
3KORSEO	한국에너지연구소	구) 한국원자력연구소
3KORYON	연세대	

표 4.3. 우리나라의 문헌 약어 코드

약어	문헌 명칭
J.KNS	한국원자력학회지
J.KPS	한국물리학회지

그러나 현재까지 보고된 수치자료는 다음 표 4.4와 같은 4 건에 불과하며, 연속에너지보다는 스펙트럼에 가중 평균된 적분 단면적이다. 또한, 실험장치도 TRIGA 원자로, D-T 중성자 발생기, Cf-252 소스, 반데그라프등으로 각각 1 건씩이며 핵자료 측정에 대한 지속적이고 집중적인 연구가 전혀 없었음을 알 수 있다. 수년전 설치된 원자력병원의 50 MeV 양성자가속기를 이용한 각종 단면적 실험결과를 NEA/DB에 보고하여 타 연구자들에게 널리 인용되도록 하여야 한다.

표 4.4. 우리 나라에서 보고된 EXFOR 자료

EXFOR	핵종(반응)	장치,에너지	참고 문헌
30388	35-Br-79 34-Se-80 45-Rh-103 49-In-115 55-Cs-133	TRIGA II, thermal, 1 eV	박혜일, "Study on the isomeric ratio by thermal neutron activation," 원자력학회지 6, p.89, 1974.
30737	41-Nb-93(n, n+a)39-Y-90M 41-Nb-93(n, n)39-Y-90M 41-Nb-93(n, 2n)39-Y-90M	neutron generator 14.6 MeV	박혜일의, "Measurement of Nb-93 (n,na) Y-89m and Nb-93 (n,na) Y-90m Cross Sections for 14 MeV Neutrons," 원자력학회지 18, p.92, 1986.
30971	1-H-1(n, g)1-H-2 25-Mn-55(n, g) 25-Mn-56	Cf-252 source 1.5 MeV	P.J.Oh 외, "Determination of the Ration of the Hydrogen and Manganese Absorption Cross Section by the Manganese Bath Technique," J.Radioanal, Nucl. Chem. 139, p.37, 1990
G0010 (77010)	29-Cu-63(g, n)29-Cu-63	17.6 + 14.7 MeV gamma from Li target	성백능 외, "Cross Section Measurement for the Cu-63(g,n)Cu-62 Reaction," 물리학회지 17, p.359, 1984.

현재까지 전세계적으로 수집된 EXFOR 자료는 10,136 건, 핵반응 수치 자료 subentry는 60,572 건이며 일련번호행을 제외한 전산자료의 총량은 262 MB 정도이다. 이 자료는 linux 기반인 전산기에 보관했으며 검색의 편의상 subentry 별로 1 개의 아스키파일로 보관한다. 이 파일의 개수가 매우 많으므로 entry 번호중 처음 3 자별로 다른 디렉토리에 보관한다. 단, entry 번호가 영문자로 시작하는 경우에는 처음 2 자로 디렉토리를 작성한다. 즉, 앞에 예로든 c0070는 c0/c0070.001, c0/c0070.002, 및 c0/c0070.003 의 3 개의 파일에 분산 보관되어 있다.

각 파일의 검색의 편의를 위해서 핵반응(REACTION) 검색어를 사용해서 subentry를 다시 정리했다. 이 색인파일은 원자 번호별로 1 개의 파일을 사용하여 index 디렉토리 아래에 보관한다.

```

lui:/mirror/exfor
      nnn - nnnmm.sss
      index - indzz

```

그림 4.2. EXFOR 파일 보관 디렉토리구조

색인파일에는 그림 4.3 처럼 subentry 번호, 핵반응 검색어, 하한 에너지, 상한 에너지를 빈칸으로 구분하여 subentry 1 개당 1 줄씩 수록한다.

```

10001.014 (11-NA-23(N,0),,EN) 2.85e+03 1.39e+05
10022.003 (11-NA-23(N,2N)11-NA-22,,SIG) 0.00e+00 1.00e+20
10047.013 (11-NA-23(N,TOT),,SIG) 2.26e+06 1.49e+07
10104.002 (11-NA-23(N,EL)11-NA-23,,DA) 5.44e+06 8.52e+06
10104.003 (11-NA-23(N,EL)11-NA-23,,DA,,LEG/2L2) 5.44e+06 8.52e+06
10104.004 (11-NA-23(N,INL)11-NA-23,PAR,DA) 5.44e+06 8.52e+06
10104.005 (11-NA-23(N,INL)11-NA-23,,DA,,LEG/2L2) 5.44e+06 8.52e+06
10104.006 (11-NA-23(N,INL)11-NA-23,PAR,DA) 5.44e+06 8.52e+06
10104.007 (11-NA-23(N,INL)11-NA-23,PAR,DA) 5.44e+06 7.60e+06
10104.008 (11-NA-23(N,INL)11-NA-23,PAR,DA) 5.44e+06 8.52e+06

```

그림 4.3. 색인파일 index/ind11 의 앞부분

입수한 EXFOR 파일을 전술한 DB구조로 정리하기 위한 프로그램 extract를 개발하였다. extract는 입력파일을 읽어서 적합한 directory에 subentry 별로 보관하고, 원소별 색인파일의 내용을 작성해 준다.

### 제 3 절 EXFOR 검색 프로그램

EXFOR 파일은 핵물리 실험 및 이용자들에게 널리 사용되는 기본자료이므로 최근에 광범위하게 사용되는 인터넷의 웹서버 기술을 사용하여 검색엔진을 작성하였다. 자료의 검색은 핵종, 반응, 에너지범위에 따라 실시할 수 있다. 이 검색엔진은 원소별 색인파일을 읽어서 지정된 subentry를 발견한 후, 다시 해당 subentry에 대한 link를 제공하는 부분과 해당 subentry, 또는 entry를 click할 경우 해당 파일의 내용을 그대로 보여 주던지, 변형하여 table 형식으로 수치화된 자료를 보여 준다. Table 형식으로 수치화된 자료는 추후 graphic 소프트웨어를 이용하여 손쉽게 도형을 작성하는데 사용할 수 있다.

## 제 5 장 평가핵자료 DB 구축

원자로심 또는 핵융합 등의 수송계산 등에 필요한 핵반응자료를 핵물리 이론에 따라 준비하여 권고하는 수치가 평가핵자료이다. 평가핵자료는 IAEA의 FENDL 처럼 여러 국제기관의 협력에 의해 평가된 자료도 있지만, 미국의 ENDF/B-6, 구소련의 BROND-2, 구주공동체의 JEF-2, 일본의 JENDL-3, 중국의 CENDL-2 처럼 각국의 필요에 따라 평가한 핵자료도 있다.

예전에는 각국마다 독자적인 파일형태를 사용했지만 근래에는 ENDF-6 파일포맷을[5.1] 사용하여 전산파일 형태로 보관한다.

평가핵자료는 입사입자에 따라 다른 파일을 사용하며, 한 파일이 너무 길면 여러 개의(3-5개) 부분으로 분할하여 배포한다. 입사입자에 따른 분류는 sublibrary라고 하며 아래 표와 같이 정해져 있다.

표.5.1. sublibrary 번호 분류

sublibrary (NSUB)	입사입자 (IPART)	내 용
0	0	광핵반응 자료
1	0	광핵분열 분열 생성물
3	0	광 원자 반응 자료
4	0	붕괴 자료
5	0	즉발 핵분열 생성물
10	1	중성자 반응 자료
11	1	중성자 핵분열 생성물
12	1	열 중성자 산란 자료
10010	1001	양성자 반응 자료
10011	1001	양성자 핵분열 생성물
10020	1002	중수소 반응 자료
...	...	
20040	2004	알파 반응 자료



또한 목적에 따라 일반 평가자료와 특수목적 자료로 구별된다. 일반평가자료는 수송계산 등에 사용할 수 있기 위한 핵자료가 위주이며 전단면적, 산란단면적, 반응단면적, 핵분열단면적, 핵분열 수율 등을 위주로 구성된다. 특수목적자료는 대개 핵계측(dosimetry) 등에 사용되는 반응단면적과 이의 공분산행렬 등으로 구성된다.

## 제 1 절 평가핵자료 DB 구축

평가핵자료의 데이터베이스를 구축하기 위해서 핵자료파일을 그림 5.1처럼 디렉토리별로 구분하여 저장하였다. 중성자반응 평가파일(sublibrary 10)은 상위 디렉토리에 배치하고 기타 평가파일은 subxxx 형태의 디렉토리명칭 아래에 다시 평가핵자료 명칭을 사용한 디렉토리에 저장하였다.

endf	endf6	ENDF/B-6
	jendl3	JENDL-3
	jef2	JEF-2
	eff2	EFF-2
	la100	LANL 100 MeV 중성자 자료
	brond2	BROND-2
	sub4	방사성 붕괴자료
	endf6	ENDF/B-6
	sub11	중성자 핵분열 생성물
	jendl3	JENDL-3
	sub12	열중성자 산란법칙자료
	endf6	ENDF/B-6
	jef2	JEF-2
	sub10010	양성자반응 단면적
	endf6	ENDB/B-6
	la100	LANL 100 MeV 양성자 자료
	subdm	dosimetry 자료
	jendl3	JENDL-2
	adl3	ADL-3
	irdf90	IRDF-90

그림 5.1. 평가핵자료 디렉토리 구조

각 디렉토리에는 각 핵종의 MT번호에 따라 분리한 텍스트파일을 보관하였으며 여러 핵종이 모여있는 공개 파일도 압축파일의 형태로 보관하였다. 입수한 파일에는 많은 핵종이 들어 있으므로 이를 MT번호에 따라 핵종별로 분류하여 독립파일을 만드는 프로그램 cutendf 를 사용한다. 사용법은 독립파일을 보관할 디렉토리에서 "cutendf fname"을 치면 된다. 여기서 "fname"은 전체파일의 명칭이며 반드시 텍스트 파일이어야 한다.

평가핵자료 데이터베이스에는 핵종마다 분리한 ENDF파일 외에도 공개된 형태의 압축파일도 보관하고 있다. 이 자료를 효과적으로 검색하기 위하여, 핵종별 파일을 검색하여 ENDF 형태인 파일의 경우에는 핵종명칭, 파일명칭, 파일크기의 목록을 화면으로 보여주고 ENDF파일 검색, 파일받기를 할 수 있도록 하였다. 또한, ENDF형태가 아닌 파일은 파일명칭, 갱신일자, 파일크기의 목록을 보여준다. 이 기능은 웹서버 기술인 server side include cgi 프로그램을 이용해서 구현하였다. 프로그램 "endf"를 이러한 목적을 위해 작성하였으며, 먼저 ENDF 형태가 아닌 파일의 목록을 파일받기가 가능하도록 제시하고, ENDF 형태의 파일은 원자번호, 질량수에 따라 정렬하여 보여준다. 이 프로그램은 파일받기를 위해서는 텍스트파일을 gz 압축파일로 변환시켜주는 copy.gz와 연결시키고, ENDF 검색을 위해서는 readendf와 연결시킨다. 각 프로그램의 소스는 부록 1에 첨부하였다.

핵종별 평가핵자료에는 MF번호에 따라 표 5.2와 같은 정보를 수록하고 있다.

각 MF 번호에 따라 다른 형태의 수치를 기록하고 있으며 자세한 것은 ENDF-102[5.1]에 수록되어 있다.

복잡한 자료를 쉽게 알 수 있는 형태로 바꾸어 보기 위하여 작성한 프로그램이 READENDF이다. READENDF는 각 핵종별 평가핵자료를 읽어서

MF, MT에 따른 section 의 종류를 나열하고 웹에서 link를 제공하는 부분과 각 section의 내용을 해독하고 수치자료를 table의 형태로 제공하는 부분으로 구성되어 있다. Table 형태로 제공되는 자료는 graphic 소프트웨어 등을 이용하여 손쉽게 도형을 작성할 수 있다.

표 5.2. ENDF-6 형식에 지정된 MF번호

MF	내 용
1	일반 정보
2	공명상수
3	반응 단면적
4	방출입자의 각분포
5	방출입자의 에너지분포
6	방출입자의 각-에너지분포
7	열중성자 산란 법칙
8	방사능 및 핵분열 생성물 분율
9	방사능핵종 생산 다중성
10	방사능핵종 생산 단면적
12	광자 생산 다중성
13	광자 생산 단면적
14	광자 각분포
15	광자 에너지분포
23	광-원자 반응 단면적
27	광-원자 산란 법칙
30	공분산 자료
31	핵분열 중성자 공분산 자료
32	공명상수 공분산자료
33	반응단면적 공분산자료
34	각분포 공분산자료
35	에너지분포 공분산자료
39	방사능핵종 생산 분율 공분산자료
40	방사능핵종 생산 단면적 공분산자료

## 제 2 절 평가 핵자료 수집

전년도까지 수집 및 구축한 평가핵자료는 ENDF/B-6, JEF-2, JENDL-3.2 이다. 금년도 연구에서 수집한 자료는 러시아의 평가핵자료와 IRDF-90 이다. 예전에 발표된 러시아의 핵자료는 처리코드인 GRUCON용의 파일 형태였으나, 근년에는 ENDF-6 형태이므로 우리 연구소에서 주로 사용하는 NJOY코드로서 처리할 수 있다.

### 1. BROND-2.2

BROND-2.2는[5.2] 러시아의 IPPE, Obninsk에서 평가한 20 MeV이하의 중성자 평가핵자료집이다. H-1에서 Cm-244 까지 121개의 핵종을 포함하며 ENDF/B-6 형태로 275,000 매의 레코드로 이루어져있다.

### 2. ADL-3

ADL-3는[5.3] 러시아의 IPPE, Obninsk에서 평가한 중성자 방사화반응 단면적자료집이며 대부분의 핵종에 대한 20,049개의 반응단면적이 포함되어 있다. 문턱반응에 대한 계산은 Hauser-Feshbach 통계모형에 의한 것이며 실험자료와 systematics에 의해 보정된 것이다. 파일의 형태는 ENDF-6이다. 그러나 meta-stable 상태로의 전환은 MT번호를 300 및 600을 더붙여서 표시하였다.

### 3. MENDL-2

MENDL-2.2는[5.4] 100 MeV까지의 입사중성자에너지에 대한 방사화단면적자료집이다. 중간에너지 입사 입자에 의한 활성화와 핵종변환 연구에 사용하기 위한 것으로, 13-Al-26 부터 84-Po-210까지의 505개의 핵종에 대

한 57,500개의 문턱반응 단면적을 ALICE-IPPE코드로 계산한 것이다. 파일의 형식은 기본적으로 ENDF-6 형태이나, MT번호는 각 파일에 기술한 대로 새로 정의하여 사용하였다.

#### 4. IRDF-90

IRDF-90는[5.5] foil에 의한 원자로 dosimetry를 하기 위한 자료로서 중성자 여기단면적과 손상단면적을 포함한다. 또한 dosimetry에 필요한 공분산 (covariance) 파일도 가지고 있다. IAEA의 전략에 따라, 각국의 평가자료를 평가하였다. IRDF version 2 에 포함된 58 개의 단면적중, ENDF/B-6에서 39 개를 취하고, 14 개는 오스트리아의 IRK에서 평가하였으며, 5 개는 중국 CNDC의 전문가가 평가하였다. 이 자료는 SAND-II 형태로도[5.6] 제공되나 여기에는 공분산 파일은 포함되어있지 않다.

#### 5. 100 MeV 이하 중성자 자료 수집

대형 양자가속기를 이용한 핵종 변환장치에서 이용하기 위한 고에너지 중성자에 대한 단면적으로 미국 Los Alamos 연구소에서 잠정적으로 발표한 평가핵자료가 있다. 이 자료는 ENDF-5 형태로 되어 있으며, 1-H-1, 4-Be-9, 6-C-12, 8-O-16, 13-Al-27, 14-Si-28, 26-Fe, 74-W, 92-U-238 등 9 개 핵종에 불과하다.

## 제 6 장 균정수 생산 및 검증

금년도 연구에서는 현재 원자력계에서 사용중인 각종 수송 및 확산 코드의 다군 단면적 자료로 이용될 수 있는 다목적 라이브러리 형태인 MATXS형의 열중성자로용 균정수를 생산하였다. 이를 검증하기 위해 몇가지 소형 검증노심을 선정하여 1차원 수송코드인 ONEDANT[6.1]를 이용하여 해석하였다. 또한 소내외에서 널리 이용되고 있는 MCNP코드[6.2] 라이브러리를 이용자들의 요구에 따라 생산하여 제공하였다.

### 제 1 절 열중성자로용 MATXS형 균정수 생산

MATXS형태의 열중성자로 해석용 중성자 69군 라이브러리를 생산하였다. 처리코드로 NJOY[6.3]를 이용하였으며, 군 구조는 소내외에서 격자코드로 이용되고 있는 WIMS 라이브러리[6.4]와 동일하게 하였다. 15개 핵종에 대해 300K로 핵종에 따른 몇가지 배경단면적을 생산하였다. 그림 6.1은 MATXS 라이브러리를 생산하기 위해 이용되는 NJOY 코드에서의 자료 흐름을 나타낸다. 입사 중성자에 대한 자료 생산을 위해서는 RECONR, BROADR, UNRESR, HEATR, THERMR, GROUPT, MATXSR 모듈이, 광자자료를 위해서는 RECONR, GROUPT, GAMINR, MATXSR 모듈이 이용된다. 이번에는 중성자 자료만 시험적으로 생산하였으며, 앞으로 이용자들의 요구에 따라 핵종추가 및 광자, 하전입자 등의 자료를 추가할 수 있다. 그림 6.2에 NJOY입력 자료를 수록하였으며, 생산된 라이브러리 내용을 부록 2에 정리하였다. MATXS형식의 이해를 위해 다음에 몇가지 내용을 정리하였다. 또한 MATXS형 자료를 수송계산 코드에 적용하기 위해서는 BBC와 TRANSX 코드[6.5]가 이용되는데 이들에 대해서도 간략하게 설명하기로 한다.

## 1. MATXS 형식

MATXS형 라이브러리를 살펴보면 각 핵종에 대해 여러 가지 내용이 수록되어 있다. 그 중에서 대표적인 것들을 다음에 정리하였다.

### (A) 표준 입자 이름

Name	Particle
N	Neutron
G	Gamma
P	Proton
D	Deutron
T	Triton
H	$^3\text{He}$ Nucleus
A	Alpha

### (B) 표준 자료형태 이름

Name	Particle
NSCAT	Neutron scattering
NG	Neutron-induced gamma production
NP	Neutron-induced proton production
NR	Neutron-to recoil matrix
GSCAT	Gamma scattering
PSCAT	Photon scattering
.....	.....
NTHERM	Thermal scattering data

(C) 중성자 반응자료

Name	MT	Description
NELAS	2	Neutron elastic scattering
NNONEL	3	Neutron nonelastic(MT1-MT2)
NINEL	4	Neutron inelastic sum (MT51-91)
N2N	16	(n,2n)
N3N	17	(n,3n)
NNA	22	(n,n' $\alpha$ )
NNP	28	(n,n'p)
N01	51	(n,n1)
N02	52	(n,n2)
NCN	91	(n,n') to continuum
NCNAAA	91	(n,n') 3 $\alpha$
N01EE	51	(n,n1) ee
NABS	101	Total absorption
NG	102	Radiative capture



D) 기타 자료

Name	MT	Description
NTOT0	1	$P_0$ total cross section
NTOT1	1	$P_1$ total cross section
NWT0	1	$P_0$ weight function (flux)
NWT1	1	$P_1$ weight function (flux)
MUBAR	251	Scattering $\mu$
XI	252	Scattering $\xi$
HEAT	301	Energy-balance heat production
KERMA	443	Kinematic KERMA factor
DAME	444	Damage-energy production
FREE	221	Free-gas scattering
H2O	222	H in H <sub>2</sub> O
POLY	223	H in polyethylene(CH <sub>2</sub> ) incoherent
POLY\$	224	H in polyethylene(CH <sub>2</sub> ) coherent
BENZ	227	Benzene incoherent

2. BBC코드

이 보조코드는 MATXS형 자료에 핵종추가 및 삭제, 포함자료 열람, 형태변환(ascii ↔ binary) 등에 이용된다.

3. TRANSX코드

TRANSX 코드는 MATXS 라이브러리로부터 수송 계산 코드나 확산 코드에서 요구하는 형식의 핵자료들을 제공한다. 중성자, 광자, 하전입자,

그리고 이것들이 조합된 자료를 이 코드를 이용하여 마련할 수 있다. 핵융합로, 고속로, 열중성자로의 핵 및 차폐 계산을 위한 자료를 마련하기 위해 이 코드가 이용된다.

## 제 2 절 MCNP 라이브러리 생산

여러 가지 목적으로 소내외에서 널리 이용되고 있는 MCNP코드용 핵자료로서 상온(300K)에서 처리된 것만이 제공되어 왔다. 그러나 장수명 핵종 소멸처리, 하나로 이용 연구개발 등 분야에서 다른 온도에 대한 자료도 요청하므로 이들을 생산하여 제공하였다. 그림 6.3은 이를 위한 입력자료의 한 예이다. 여기서는 MCNP코드 이용자들이 적절한 자료를 선택하여 이용하는데 도움을 주기위해 연속에너지 자료의 내용 및구성에 대해 간략하게 설명한다.

### 1. 중성자 자료

MCNP 라이브러리는 거의 평가핵자료 그 자체이지만, 기본적인 차이는 다양한 자료를 무작위(random)로 이용할 수 있도록 배치한다는 것이다. 또한 MCNP 라이브러리는 단면적 자료들이 동일한 에너지에 대한 값들로 처리되고, 몇가지 자료들에 대해서는 동일한 확률 및 누적확률 분포로 주어진다. MCNP코드에서 입사 입자의 용이한 각분포 선정을 위해 등방산란일 때를 제외하고는 32가지의 동일한 확률분포를 갖도록 한다. 2차 생성입자 에너지분포는 누적밀도함수로 주어진다. ENDF-6 형식에서 채택되고 있는 에너지와 산란각분포(MF6)는 핵자료 평가시 적용된 이론모델에 따라 누적분포함수 및 무작위 추출방법이 적용되도록 자료가 제공된다.

## 2. 광자 생성자료

구 MCNP 광자 자료에는 30 가지의 에너지를 갖는 입사 중성자에 따라 생성되는 광자가 20 가지의 에너지로 동일한 확률을 갖도록 주어졌으나, ENDF-6 형식에서는 평가핵자료의 MF3과 MF6를 이용하여 생성광자의 에너지 및 각분포 자료를 생산할 수 있다.

## 3. 선량 단면적 자료

이 자료는 중성자의 수송 계산에는 사용될 수 없으며 자료형식 또한 다르다. 선량 평가핵자료 그대로라고 할 수 있다.

## 4. 열중성자 산란자료

대략 4eV 아래에서 핵의 열운동은 산란중성자의 에너지 이득을 가져오는데 액체나 고체 상태의 원자는 산란 중성자의 에너지와 각분포에 영향을 미친다. MCNP코드에서는 자유기체(free gas)자료와 원자의 속박효과를 고려한 자료를 이용할 수 있다. 예를 들면 물 속의 수소에 대해 4eV 이상에서는 자유기체 자료를 이용하고 그 아래 에너지에서는 열중성자 산란자료를 이용하는 것이다. 열중성자 산란이 중요한 영향을 미치는 계산에서는 반드시 두 가지 자료를 동시에 이용해야 만족한 결과를 얻을 수 있다. 그림 6.4는 MCNP 코드에서 이용되는 열중성자 산란자료를 생산하기 위한 입력자료이다.

## 제 3 절 MATXS69 라이브러리 검증[6.6]

### 1. ORNL-1, -2, -3, -4, -10

$^{235}\text{U}$  구형노심(uranylnitrate solution)이며, 반사체는 없다. 이 벤치마크 문제는  $\text{H}_2\text{O}$ 의 고속산란자료,  $^{235}\text{U}$ 와  $^1\text{H}$ 의 열중성자 포획단면적 그리고

$^{235}\text{U}$ 의 열중성자 분열단면적 및  $\nu$  데이터를 검증하는 데 유용하다. 다음에 ORNL-1, -2, -3, -4, -10의 특성을 간략히 정리했다.

	H/ $^{235}\text{U}$	Radius (Sphere)	Boron
ORNL-1	1378	34.595 cm	no
ORNL-2	1177	34.595 cm	yes
ORNL-3	1033	34.595 cm	yes
ORNL-4	972	34.595 cm	yes
ORNL-10	1835	61.011 cm	no

2. L-7, -8, -9, -10, -11

$^{235}\text{U}$  구형노심(uranyle fluoride-  $\text{UO}_2\text{F}_2$ )이며, ORNL의 보충실험이다.  $\text{H}_2\text{O}$ 의 고속산란자료,  $^{235}\text{U}$  자료 및  $^1\text{H}$ 의 열중성자 포획단면적자료 검증에 유용하다. 다음은 그 특성이다.

	H/ $^{239}\text{Pu}$	Radius (Sphere)	Reflected	Fuel Loading ( $^{235}\text{U}$ kg)
L7	76.1	11.5176 cm	yes	$2.08 \pm 0.2$
L10	126.5	11.8442 cm	yes	$1.39 \pm 0.01$
L8	1112	27.9132 cm	no	$2.13 \pm 0.2$
L11	1270	27.9132 cm	yes	$1.86 \pm 0.2$
L9	1393	34.6327 cm	no	$3.25 \pm 0.03$

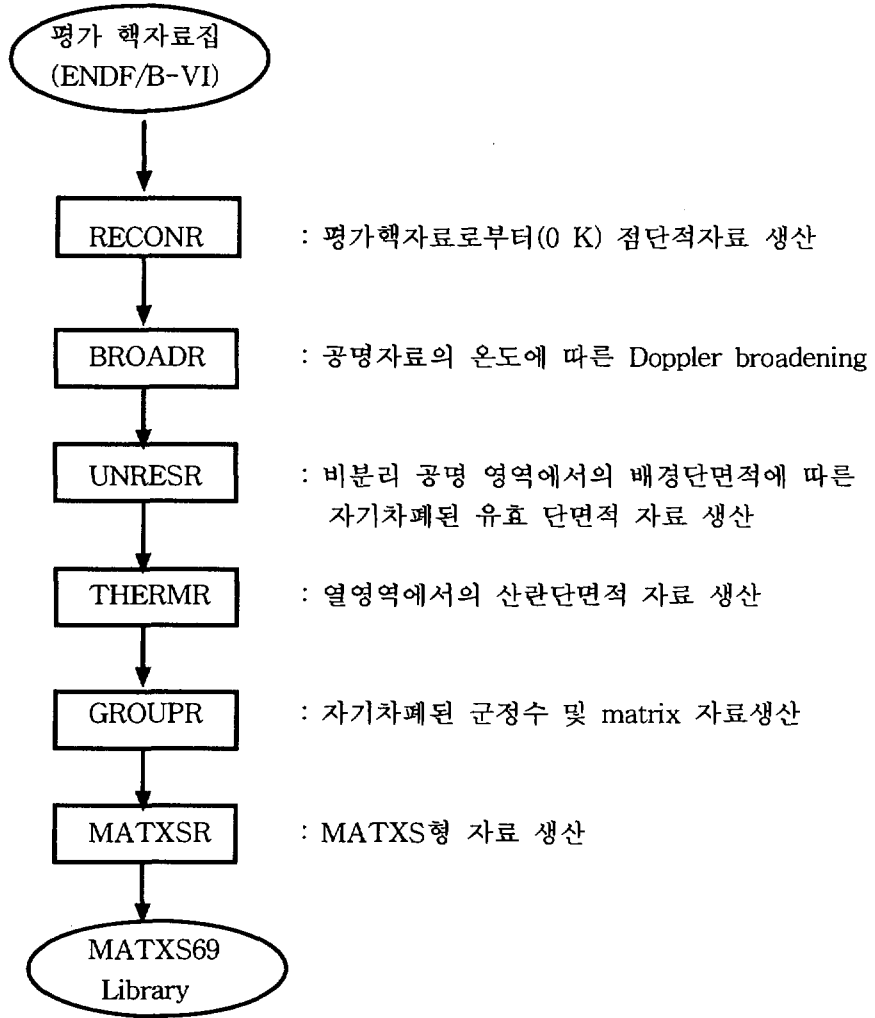


그림 6.1 NJOY코드에서 MATXS 라이브러리 생산을 위한 자료 흐름도

그림 6.2 MATXS69 열중성자로용 라이브러리 생산을 위한 NJOY 입력  
자료

```

0
6
*moder*
1 -21
*u238 */
20 9237
0/
*reconr*
-21 -22
*nuclide from endf/b-6*/
9237 1/
0.001 0. 7 /
*reconstructed from 0 K*/
0 /
*broadr*
-22 -23
9237 1 /
0.001 /
300.
0 /
*unresr*
-21 -23 -24
9237 1 8 0/
300.
1.e10 1.e4 1.e3 100. 30. 10. 3. 1.
0/
*thermr*
0 -24 -26
0 9237 8 1 1 0 1 221 0
300.
0.005 4.0
*groupr*
-21 -26 0 25
9237 9 0 -5 5 1 8 1 /
*material from endf/b-6* /
300.
1.e10 1.e4 1.e3 100. 30. 10. 3. 1.
9118.0 11.171 200000./
3 /
3 221 /
3 251 /
3 252 /

```

```
3 253 /
3 259 /
3 455/
5 455/
6 /
6 221/
0 /
0 /
*matxsr*
25 0 26/
22 *matxs u238*/
1 2 2 1/
*69 group neutron only */
*test case using endf/b-vi kaeri*/
*n* /
69 /
*nscat* *ntherm*/
1 1 /
1 1/
*u238* 9237 /
*stop*
```

그림 6.3 MCNP 라이브러리 생산을 위한 NJOY 입력자료

```
0
6
*moder*
20 -21
*reconr*
-21 -22
*nuclide from ENDF/B-VI.0*/
9237 1/
0.001 0 7 /
*reconstructed from 0 K*/
0 /
*broadr*
-21 -22 -23
9237 1 0 1 0/
0.001/
300.
0/
*heatr*
-21 -23 -24 0
9237 0 0 0 0 0
*acer*
-21 -24 0 25 26 /
1 0 1 .60c 1/
* U-238 T=300 K from ENDF/B-VI * /
92 238
9237 300. /
0.01 1 /
/
*stop*
```



그림 6.4 MCNP 열중성자 자료 생산을 위한 NJOY 입력자료

```
0
6
*moder*
30 -31
*reconr*
-31 -32
*pendf tape for endf6.3*/
125 1/
.001 /
* reconstructed from 0 K*/
0/
*broadr*
-32 -34
125 1/
.001/
300
0/
*thermr*
20 -34 -35
1 125 8 1 4 0 2 222 1
300
0.05 4.
*acer*
-31 -35 0 25 26/
2 0 1 .01t/
*H2O S(a,b) data from kaeri, 300 K*/
125 300 lwtr
1001/
222 20 0 0 1 4.0/
*stop*
```

### 3. 검증계산 및 결과

ENDF/B-VI를 이용하여 시험적으로 생산된 열 중성자로 해석용 MATXS 형식의 69군 단면적 라이브러리를 이용하여, 93w/o 농축  $U^{235}$  무반사 구형노심인 ORNL-1, -2, -3, -4, -10 과 97w/o 농축  $U^{233}$  무반사 구형노심인 ORNL-5, -6, -8, -9, -11 및 93w/o 농축  $U^{235}$  반사 및 무반사 구형노심인 L-7, -8, -9, -10, -11 을 해석하여 이용 가능성을 타진하였다. 수송계산 코드로는 ONEDANT가 이용되었으며, MATXS 라이브러리로부터 TRANSX 코드를 이용하여 필요 자료를 생산하였다. 두 번 반복하여 계산을 수행했는데, 즉 일차로 MATXS69 라이브러리로부터 TRANSX 코드를 이용하여 핵분열 스펙트럼, transport table 및 material의 자기차폐된 유효 단면적을 마련하였다. 여기서 가중함수로는 MATXS69에 있는  $P_0$  중성자속이 이용되었다. 이 자료로부터 ONEDANT로  $P_3S_{16}$  수송계산을 수행하였다. 이차로 ONEDANT에서 생산된 CCCC 형식의 중성자속을 가중함수로 하여 MATXS/TRANSX/ONEDANT 계산을 반복 수행하여 임계값을 구하였다. 표 6.1에 실험값과 계산값을 비교하여 정리하였는데, ORNL- 노심은 계산값이 모두 실험값에 비해 다소 작고, L- 노심에서는 약간 크게 나타났으나 전체적으로 실험값과 약 0.5% 이내에서 일치하고 있음을 알 수 있다. 앞으로 자료도 보완하고 좀더 많은 검증계산도 수행할 계획이다.

표 6.1. MATXS69 열중성자로용 라이브러리를 이용한  
유효중배계수의 비교

<u>Core</u>	<u>Exp.</u>	<u>Calculated</u>	<u>C/E</u>
ORNL-1	1.00026	0.99786	0.99760
ORNL-2	0.99975	0.99768	0.99793
ORNL-3	0.99994	0.99474	0.99480
ORNL-4	0.99924	0.99621	0.99697
ORNL-10	1.00031	0.99745	0.99715
ORNL-5	0.99949	0.99758	0.99809
ORNL-6	1.00009	0.99808	0.99799
ORNL-8	0.99930	0.99780	0.99850
ORNL-9	0.99942	0.99720	0.99778
ORNL-11	0.99944	0.99549	0.99605
L-7	1.0000	1.00577	1.00577
L-8	1.0004	1.00588	1.00548
L-9	1.0000	1.00245	1.00245
L-10	1.0000	1.00370	1.00370
L-11	0.9999	1.00083	1.00093

## 제 7 장 결 론

금년도 연구로 핵자료 평가 및 이용에 필수적인 핵구조자료 및 핵반응 실험자료 데이터베이스를 구축하였으며, 선진외국의 주요 평가핵자료집도 수집하였다. 수집된 자료는 핵종표와 단면적 도형작성용 온라인 웹서버로 개발하여 국내외 이용자에게 배포하고 있다.

다목적으로 이용될 수 있는 MATXS형 라이브러리의 생산 및 검증이 계속적으로 수행하여 왔다. 금년에도 열중성자로 해석용 MATXS69를 생산하여 몇 가지 벤치마크 계산을 통해 그 유용성을 확인하였다. 앞으로 이용자들의 요구에 따라 핵종 및 요구자료가 추가될 것이다. 또한 군 구조도 필요에 따라 가감하여 보다 유용한 라이브러리를 제공할 계획이다. 소내외에 MATXS형 라이브러리의 유용성을 홍보하여 보다 많이 이용되도록 해야겠다. 또한 MCNP 이용자들이 요구하는 자료가 지속적으로 제공되도록 그 라이브러리를 계속 생산할 계획이다.

## 참고 문헌

- [2.1] 장종화 등, "핵자료 체제구축," KAERI/RR-1665/96, 한국원자력연구소, 1997.
- [3.1] J.K. Tuli, "Evaluated Nuclear Structure Data File - A Manual for Preparation of Data Sets," BNL, 1987.
- [4.1] V. McLane, "EXFOR Systems Manual Nuclear Reaction Data Exchange Format," BNL-NCS-63330, BNL, 1996.
- [5.1] V.McLane et al. ed., "ENDF-102 - Data Formats and Procedures for the Evaluated Nuclear Data File ENDF-6," BNL-NCS-44945 Rev.2/97, BNL, 1997.
- [5.2] V.N. Manokhin et al., "BROND-2.2, Russian Evaluated Neutron Reaction Data Library," IAEA-NDS-90 Rev.8 (1994)
- [5.3] O.T.Grudzevich et al., "ALD-3, Neutron activation data library," IAEA-NDS-137 Rev.1 (1994)
- [5.4] Yu.N. Shubin et al., "MENDL-2, Neutron reaction data library for nuclear activation and transmutation at intermediate energies," IAEA-NDS-136 Rev.1 (1995)
- [5.5] N.P. Kocherov and P.K. McLaughlin, "The International Reactor Dosimetry File (IRDF-90 Version 2)," IAEA-NDS-141, Rev.3 (1996)
- [5.6] S. Berg and W.N. McElroy, "A Computer-Automated Iterative Method for Neutron Flux Spectra Determination by Foil Activation - Vol. II: SAND II (Spectrum Analysis by Neutron Detectors) and Associated Codes," AFWL-TR-67-41, Volume II (September 1967).

- [6.1] R.D. O'Dell, F.W. Brinkley, Jr. and D.R. Marr, "User's Manual for ONEDANT: A Code Package for Two-Dimensional Diffusion Accelerated, Neutron-Particle Transport," Los Alamos National Laboratory report LA-9184-M (February 1982).
- [6.2] J. Briesmeister (Editor), "MCNP-A General Monte Carlo Code N-Particle Transport Code Version 4A," LA-12625-M (November 1993).
- [6.3] R. E. MacFarlane, D.W. Muir, "The NJOY Nuclear Data Processing System, Version 91," LA-12740-M (Oct. 1994).
- [6.4] C.J. Taubman, "The WIMS69-group Library Tape 166259," AEEW-M-1324 (1975).
- [6.5] R. E. MacFarlane, "TRANSX 2: A Code for Interfacing MATXS Cross-Section Libraries to Nuclear Transport Codes," Los Alamos National Laboratory report LA-12312-MS (December 1993).
- [6.6] ENDF-202, Cross Section Evaluation Working Group Benchmark Specifications, BNL-19302 (1974) (Revised 11-1981).

## 부록 1. 평가해자료 검색 프로그램 source list

```
/*
** readendf.c
** Programmed by YoungSik Cho
*/
/*
** Apr.28/96 - jhchang
** added temporal file feature,
** remove conditional compilation HTML and
** introduce a variable html
*/

#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <math.h>
#include "endfutil.h"

char dataf_str[200];
char cxfile[200];
int outf;
int col,row;
int html;

void href(char *mfmt,char *desc,int lines)
{
    printf("<a href=\"/cgi-bin/readendf?%s+%s\">%s</a> ",dataf_str,mfmt,desc);
    if(lines > 0) printf("[%d lines] ",lines);
    printf("<a href=\"/cgi-bin/readendf?-ocomma+%s+%s\">C</a> ",dataf_str,mfmt);
    printf("<a href=\"/cgi-bin/readendf?-oblank+%s+%s\">B</a> ",dataf_str,mfmt);
    printf("<a href=\"/cgi-bin/readendf?-osylk+%s+%s\">S</a> ",dataf_str,mfmt);
    printf("\n");
}

void DisplayLIST(ENDF *endf,int N,int NC,char *caption,char **name)
{
    double *B;
    int i,n,m;
    if ((B=malloc(N*sizeof(double))) == NULL) {
        puts("Out of memory");
        exit(1);
    }
    ReadLIST(endf,N,B);
    if(NC < 1) NC=10;
    if(html) {
        printf("<TABLE BORDER=2><CAPTION>%s</CAPTION>\n",caption);
        if(NC < 1) NC = 10;
        printf("<TR>");
        for(n=0;n<NC;n++) {
            if(name[n] != NULL) printf("<TH>%s</TH>",name[n]);
            else printf("<TH>%d</TH>",n);
        }
    }
}
```

```

    printf("</TR>\n");
}
else {
    printf("%s\n",caption);
    for(n=0;n<NC-1;n++) printf("%s, ",name[n]);
    printf("%s\n",name[NC-1]);
}
for (i=0;i<(int)ceil((N+0.001)/NC):i++) {
    if ((m = N-i*NC) > NC ) m = NC;
    if(html) {
        printf("<TR>");
        for (n=0;n<m;n++) printf("<TD>%5le</TH>",B[i*NC+n]);
        printf("</TR>\n");
    }
    else {
        for (n=0;n<m;n++) {
            printf("%5le",B[i*NC+n]);
            if(n < m-1) {
                switch(outf) {
                    case OUT_COMMA: printf(", "); break;
                    default: printf(" ");
                }
            }
        }
        printf("\n");
    }
}
if(html) puts("</TABLE>");
free(B);
}

void DisplayTAB1(ENDF *endf,int N1,int N2,double m1,double m2)
{
    double *X,*Y;
    int i;
    if ((X=(double*)malloc(N2*sizeof(double))) == NULL) {
        puts("Out of memory");
        exit(1);
    }
    if ((Y=(double*)malloc(N2*sizeof(double))) == NULL) {
        puts("Out of memory");
        exit(1);
    }
    ReadTAB1(endf,N1,N2,NULL,NULL,X,Y);
    if (outf == OUT_SYLK) puts("ID:NONE");
    for (i=0;i<N2:i++) {
        if (outf == OUT_BLANK)
            printf("%5le %5le\n",m1*X[i],m2*Y[i]);
        else if (outf == OUT_COMMA)
            printf("%5le, %5le\n",m1*X[i],m2*Y[i]);
        else if (outf == OUT_SYLK)
            printf("C:Y%d:X%d:K%5le\nC:X%d:K%5le\n",
                row++,col,m1*X[i],col+1,m2*Y[i]);
    }
}

```



```

        else if (outf == OUT_TABLE)
            printf("<TR><TD>x.51e</TD><TD>x.51e</TD></TR>\n", m1*X[i], m2*Y[i]);
    }
    if (outf == OUT_SYLK) puts("E");
    free(X);
    free(Y);
}

int ReadContents(ENDF *endf)
{
    char HL[67];
    double C1, C2;
    int L1, L2, N1, N2, MAT, MF, MT;
    int MMF, MMT;
    int FLAG;
    int i;
    int lines;
    char mtmfstr[50], remstr[100];

    if(html) {
        printf("<TITLE>xs - Contents</TITLE>\n", cxfile);
        printf("<PRE><B><FONT size=+1>xs - Contents</FONT></B>\n\n", cxfile);
        puts("In the followings,\n"
            "C means \"Comma delimited format\",\n"
            "B means \"Blank delimited format\" and\n"
            "S means \"SYLK format (can be used by some applications such as Excel or\n"
            Grapher)");
        puts("</PRE>\n<UL>");
    }
    MMF = 0;
    MMT = 0;
    lines = 0;
    while (1) {
        ReadlnString(endf, HL, &MAT, &MF, &MT);
        if (feof(endf->fp)) break;
        if (MF == -1) break;
        else if (MF == 0) {
            if(html) if (MMF > 0) puts("</UL>");
            MMT = 0;
        }
        else if (MF > 0) {
            if (MF != MMF) {
                if (MF == 2) {
                    ReadlnString(endf, HL, &MAT, &MF, &MT);
                    ReadCONT(endf, &MAT, &MF, &MT, &C1, &C2, &FLAG, &N1, &N1, &N2);
                }
                if(html) {
                    printf("<LI>%d.  %s</LI>", MF, GetDescription(MF, 0, 0));
                    puts("<UL>");
                }
            }
            else printf("%d.  %s\n", MF, GetDescription(MF, 0, 0));
            MMF = MF;
        }
    }
}

```

```

        if (MT == 0 && MMT > 0) {
            sprintf(remstr, "%d, %d: %s", MF, MMT, GetDescription(MF, MMT, FLAG));
            if(html) {
                printf("<li>");
                sprintf(mtmfstr, "%d+%d", MF, MMT);
                href(mtmfstr, remstr, lines);
            }
            else printf("%s [%d lines]\n", remstr, lines);
            lines = 0;
        }
        if (MT > 0) {
            if (MT != MMT) MMT = MT;
            lines++;
        }
    }
}
if(html) puts("</UL>");
return 1;
}

int ReadFile1(ENDF *endf, int mt)
{
    char ins[81], HL[67];
    double C1, C2;
    int L1, L2, N1, N2, MAT, MF, MT;
    int NN1, NN2;
    int LNU;
    int idx = 1;
    double *B, *X, *Y;
    int i;
    char *name[100];

    if (!LocateFile(endf, 1)) return 0;
    if (!LocateSection(endf, mt)) return 0;

    /* read HEAD record */
    ReadCONT(endf, &MAT, &MF, &MT, &C1, &C2, &L1, &L2, &N1, &N2);
    if (MF == 0) FatalError(endf, "Unexpected FEND");
    if (MF != 1) {
        sprintf(HL, "Wrong MF %d", MF);
        FatalError(endf, HL);
    }
    if(html) {
        printf("<TITLE>%s - %s</TITLE>\n", cxfile, GetDescription(1, mt, 0));
        printf("<PRE><B><FONT size=+1>%s</FONT></B>\n\n", GetDescription(1, mt, 0));
    }
    else printf("%s\n", GetDescription(1, mt, 0));
    switch (MT) {
        case 451:
            printf("ZA=%x. Of, AWR=%x. 3le\n", C1, C2);
            ReadCONT(endf, &MAT, &MF, &MT, &C1, &C2, &L1, &L2, &N1, &N2);
            if (endf->is_endf6)
                ReadLnString(endf, HL, &MAT, &MF, &MT);
    }
}

```

```

else Warn(endf, "Not ENDF-VI format");
ReadCONT(endf, &MAT, &MF, &MT, &C1, &C2, &L1, &L2, &N1, &N2);
NN1=N1;
NN2=N2;
if(html) puts("<B><FONT size=+1>Descriptive Data</FONT></B>\n");
else puts("Descriptive Data");
for (i=0;MT && i<NN1;i++) {
    ReadlnString(endf, HL, &MAT, &MF, &MT);
    puts(HL);
}
if(html) puts("\n<B><FONT size=+1>Directory</FONT></B>\n");
else puts("Directory");
for (i=0;MT && i<NN2;i++) {
    ReadlnString(endf, HL, &MAT, &MF, &MT);
    puts(HL);
}
break;
case 452:
    if (L2==1) {
        puts("The average number of neutrons per fission are given as
coefficients for the polynomial expansion");
        ReadCONT(endf, &MAT, &MF, &MT, &C1, &C2, &L1, &L2, &N1, &N2);
        for(i=0;i<10;i++) {
            name[i] = malloc(4);
            sprintf(name[i], "C%d", i);
        }
        DisplayLIST(endf, N1, 10, "Coefficients", name);
        for(i=0;i<10;i++) free(name[i]);
    }
    else if (L2==2) {
        puts("The average number of neutrons per fission are given as a
function of energy");
        if(html) puts("</PRE><TABLE BORDER=2><TR><TH>Energy</TH>"
"<TH>Average Number of Neutrons<BR>per Fission</TH></TR>");
        ReadCONT(endf, &MAT, &MF, &MT, &C1, &C2, &L1, &L2, &N1, &N2);
        DisplayTAB1(endf, N1, N2, 1.0, 1.0);
        if(html) puts("</TABLE>");
    }
}
break;
case 455:
    LNU = L2;
    puts("Decay constans");
    ReadCONT(endf, &MAT, &MF, &MT, &C1, &C2, &L1, &L2, &N1, &N2);
    for(i=0;i<10;i++) {
        name[i] = malloc(10);
        sprintf(name[i], "Lamdaxd", i);
    }
    DisplayLIST(endf, N1, 10, "Decay Constants", name);
    for(i=0;i<10;i++) free(name[i]);
    if (LNU==1) {
        ReadCONT(endf, &MAT, &MF, &MT, &C1, &C2, &L1, &L2, &N1, &N2);
        ReadLIST(endf, 1, &C1);
        if(html) printf("<PRE>Average number of delayed neutrons per fission is

```

```

*lf</PRE>",C1);
    else printf("Average number of delayed neutrons per fission is
*lf\n",C1);
    }
    else if (LNU==2) {
        if(html) puts("<PRE>Average number of delayed neutrons per fission are
given as a function of energy</PRE>");
        else puts("Average number of delayed neutrons per fission are given as
a function of energy");
        if(html) puts("<TABLE BORDER=2><TR><TH>Energy</TH>"
" <TH>Average Number of Delayed Neutrons<BR>per
Fission</TH></TR>");
        ReadCONT(endf, &MAT, &MF, &MT, &C1, &C2, &L1, &L2, &N1, &N2);
        DisplayTAB1(endf, N1, N2, 1, 0, 1, 0);
        if(html) puts("</TABLE>");
    }
    break;
case 456:
    if (L2==1) {
        ReadCONT(endf, &MAT, &MF, &MT, &C1, &C2, &L1, &L2, &N1, &N2);
        ReadLIST(endf, 1, &C1);
        printf("Average number of prompt neutrons per fission is *lf\n",C1);
    }
    else if (L2==2) {
        puts("Average number of prompt neutrons per fission are given as a
function of energy");
        if(html) puts("</PRE><TABLE BORDER=2><TR><TH>Energy</TH>"
" <TH>Average Number of Prompt Neutrons<BR>per Fission</TH></TR>");
        ReadCONT(endf, &MAT, &MF, &MT, &C1, &C2, &L1, &L2, &N1, &N2);
        DisplayTAB1(endf, N1, N2, 1, 0, 1, 0);
        if(html) puts("</TABLE>");
    }
    if (L2==1) {
        ReadCONT(endf, &MAT, &MF, &MT, &C1, &C2, &L1, &L2, &N1, &N2);
        ReadLIST(endf, N1, NULL);
    }
    else if (L2==2) {
        ReadCONT(endf, &MAT, &MF, &MT, &C1, &C2, &L1, &L2, &N1, &N2);
        ReadTAB1(endf, N1, N2, NULL, NULL, NULL, NULL);
    }
    break;
case 458:
    ReadCONT(endf, &MAT, &MF, &MT, &C1, &C2, &L1, &L2, &N1, &N2);
    if ((B=malloc(18*sizeof(double))) == NULL) {
        puts("Out of memory");
        exit(1);
    }
    ReadLIST(endf, 18, B);
    if(html) {
        puts("</PRE><TABLE BORDER=2>");
        printf("<TR><TH>Total Energy Release per
Fission</TH><TD>*lf</TD></TR>", B[16]);
        printf("<TR><TH>Kinetic Energy of the

```

```

Fragments</TH><TD>%lf</TD></TR>", B[0]);
    printf("<TR><TH>Kinetic Energy of the Prompt Fission
Neutrons</TH><TD>%lf</TD></TR>", B[2]);
    printf("<TR><TH>Kinetic Energy of the Delayed Fission
Neutrons</TH><TD>%lf</TD></TR>", B[4]);
    printf("<TR><TH>Total Energy of Prompt Gamma
Rays</TH><TD>%lf</TD></TR>", B[6]);
    printf("<TR><TH>Total Energy of Delayed Gamma
Rays</TH><TD>%lf</TD></TR>", B[8]);
    printf("<TR><TH>Total Energy of Delayed
Betas</TH><TD>%lf</TD></TR>", B[10]);
    printf("<TR><TH>Energy Carried away by the
neutrinos</TH><TD>%lf</TD></TR>", B[12]);
    puts("</TABLE>");
}
else {
    printf("Total Energy Release per Fission : %lf\n", B[16]);
    printf("Kinetic Energy of the Fragments : %lf\n", B[0]);
    printf("Kinetic Energy of the Prompt Fission Neutrons : %lf\n", B[2]);
    printf("Kinetic Energy of the Delayed Fission Neutrons : %lf\n", B[4]);
    printf("Total Energy of Prompt Gamma Rays : %lf\n", B[6]);
    printf("Total Energy of Delayed Gamma Rays : %lf\n", B[8]);
    printf("Total Energy of Delayed Betas : %lf\n", B[10]);
    printf("Energy Carried away by the neutrinos : %lf\n", B[12]);
}
free(B);
break;
default:
    sprintf(ins, "Unknown section type: %d\n", MT);
    FatalError(endf, ins);
    break;
}
if (MT==0) Warn(endf, "Unexpected SEND record");
else CheckSEND(endf);
return 1;
}

int ReadFile2(ENDF *endf, int mt)
{
    char HL[81];
    double C1, C2;
    int L1, L2, N1, N2, MAT, MF, MT;
    int NER, LFW, LRU, LRF, NRO, NLS, NE;
    int idx = 1;
    int i, j, n, l;
    double *B;
    double EL, EH, SPI, AP;
    int NJS, L;
    double AJ;
    if (!LocateFile(endf, 2)) return 0;
    if (!LocateSection(endf, mt)) return 0;

    /* read HEAD record */

```

```

ReadCONT(endf, &MAT, &MF, &MT, &C1, &C2, &L1, &L2, &N1, &N2);
ReadCONT(endf, &MAT, &MF, &MT, &C1, &C2, &LRU, &LFW, &NER, &N2);
if (MT == 0) FatalError(endf, "Unexpected SEND");
if (MF == 0) FatalError(endf, "Unexpected FEND");
if (MF != 2) {
    sprintf(HL, "Wrong MF %d", MF);
    FatalError(endf, HL);
}
if(html) {
    printf("<TITLE>xs - 2.151: xs</TITLE>\n", cxfile, "Resonance Parameters");
    printf("<B>2.151: xs</B><p>\n", "Resonance Parameters");
}
else printf("Resonance Parameters\n");
for(n=0;n<NER;n++) {
    ReadCONT(endf, &MAT, &MF, &MT, &EL, &EH, &LRU, &LRF, &NRO, &N2);
    switch(LRU) {
        case 0: printf("Only Scattering Radius is given\n"); break;
        case 1: printf("Resolved resonance "); break;
        case 2: printf("Unresolved resonance "); break;
    }
    printf("Range= %.3le - %.3le eV: ", EL, EH);
    if (LRU == 1) {
        if (LRF == 1) puts("Single-level Breit-Wigner (SLBW)");
        else if (LRF == 2) puts("Multilevel Breit-Wigner (MLBW)");
        else if (LRF == 3) puts("Reich-Moore (RM)");
        else if (LRF == 4) puts("Adler-Adler (AA)");
        else if (LRF == 5) puts("General R-matrix (GRM)");
        else if (LRF == 6) puts("Hybrid R-function (HRF)");
    }
    else if (LRU == 2) {
        if (LRF == 1) puts("All parameters are energy-independant");
        else if (LRF == 2) puts("All parameters are energy-dependant");
    }
    if(html) printf("<br>\n");
    if (NRO == 0) puts("Scattering radius is energy-independant.");
    else if (NRO == 1) puts("Scattering radius is expressed as a table of
energy,radius pairs");
    if(html) printf("<br>\n");
    switch(LRU) {
        case 0: /* Only Sc. Rad. */
            ReadCONT(endf, &MAT, &MF, &MT, &SPI, &AP, &L1, &L2, &N1, &N2);
            printf("Target Spin=%.1lf, Scat. radius = %.4lfE-12cm\n", SPI, AP);
            break;
        case 1: /* Resolved Resonance */
            ReadCONT(endf, &MAT, &MF, &MT, &SPI, &AP, &L1, &L2, &NLS, &N2);
            printf("Target Spin=%.1lf, Scat. radius = %.4lfE-12cm\n", SPI, AP);
            if(html) printf("<br>\n");
            for(l=0;l<NLS;l++) {
                printf("Neutron Orbital Ang. Momentum %d\n", l);
                if (LRF == 1 || LRF == 2 || LRF == 3 || LRF == 4) {
                    ReadCONT(endf, &MAT, &MF, &MT, &C1, &C2, &L1, &L2, &N1, &N2);
                    if ((B=malloc(N1*sizeof(double))) == NULL) {
                        puts("Out of memory");
                    }
                }
            }
        }
    }
}

```

```

        exit(1);
    )
    ReadLIST(endf, N1, B);
    if (LRF == 1 || LRF == 2)
        if (html)
            puts("</PRE><TABLE
BORDER=2><TR><TH>Energy</TH><TH>J</TH><TH>Total Width</TH>
        <TH>Neutron Width</TH><TH>Radiation Width</TH><TH>Fission
Width</TH></TR>");
        else puts("Energy, J, Total Width, Neutron Width, Radiation Width,
Fission Width");
        else if (LRF == 3)
            if (html) puts("</PRE><TABLE BORDER=2><TR><TH>Energy</TH><TH>J</TH>
        <TH>Neutron Width</TH><TH>Radiation Width</TH>
        <TH>The First Partial<BR>Fission Width</TH>
        <TH>The Second Partial<BR>Fission Width</TH></TR>");
            else puts("Energy, J, Neutron Width, Radiation, First Fission,
        "Second Fission");
        else if (LRF == 4)
            if (html) puts("</PRE><TABLE BORDER=2><TR><TH COLSPAN=6>Background
constants for the Fission Cross Section</TH></TR>");
            else puts("Background constants for the Fission Cross Section");
            for (i=0; i<N1/6; i++) {
                if (html) printf("<TR><TD>% .4le</TD><TD>% .11f</TD><TD>% .4le</TD>\
        <TD>% .5le</TD><TD>% .4le</TD><TD>% .4le</TD></TR>\n",
                    B[6*i], B[6*i+1], B[6*i+2], B[6*i+3], B[6*i+4], B[6*i+5]);
                else printf("% .4le, % .11f, % .4le, % .4le, % .4le, % .4le\n",
                    B[6*i], B[6*i+1], B[6*i+2], B[6*i+3], B[6*i+4], B[6*i+5]);
            }
            if (html) puts("</TABLE>");
            free(B);
        )
    else puts("Currently this format is not supported");
    )
    break;
case 2: /* Unresolved Resonance */
    if (LRF == 1) {
        ReadCONT(endf, &MAT, &MF, &MT, &SPI, &AP, &L1, &L2, &NLS, &N2);
        printf("LFW=%d\n", LFW);
    }
    else if (LRF == 2) {
        ReadCONT(endf, &MAT, &MF, &MT, &SPI, &AP, &L1, &L2, &NLS, &N2);
        for (l=0; l<NLS; l++) {
            ReadCONT(endf, &MAT, &MF, &MT, &C1, &C2, &L, &L2, &NJS, &N2);
            for (j=0; j<NJS; j++) {
                ReadCONT(endf, &MAT, &MF, &MT, &AJ, &C2, &L1, &L2, &N1, &NE);
                if ((B=malloc(N1*sizeof(double))) == NULL) {
                    puts("Out of memory");
                    exit(1);
                }
            }
            ReadLIST(endf, N1, B);
            printf("L=%d, J=% .1f, ", l, j);
            printf("Degree of freedom: X=% .1f, N=% .1f, G=% .1f, F=% .1f\n",
                B[2], B[3], B[4], B[5]);
        }
    }
}

```

```

printf("ES, D, GX, GNO, GG, GF\n");
for (i=1;i<N1/6;i++) {
  if(html) printf("<TR><TD>% .4le</TD><TD>% .4lf</TD><TD>% .4le</TD>\n\
    <TD>% .5le</TD><TD>% .4le</TD><TD>% .4le</TD></TR>\n",
    B[6*i],B[6*i+1],B[6*i+2],B[6*i+3],B[6*i+4],B[6*i+5]);
  else printf("% .4le, % .4lf, % .4le, % .4le, % .4le, % .4le\n",
    B[6*i],B[6*i+1],B[6*i+2],B[6*i+3],B[6*i+4],B[6*i+5]);
} /* end of for i < N1/6 */
free(B);
} /* end of for j < NJS */
} /* end of for l < NLS */
}
else {
  printf("*** Unknown LRF=%d, LFW=%d for Un.Res ***\n",LRF,LFW);
}
break:
default:
  printf("*** Unknown format LRU=%d\n",LRU);
  break:
}
}
return 1;
}

int ReadFile3(ENDF *endf,int mt,double m1,double m2)
{
  char HL[81];
  int MMT = 0;
  int idx = 1;
  double C1,C2;
  int L1,L2,N1,N2,MAT,MF,MT;

  if (!LocateFile(endf,3)) return 0;
  if (!LocateSection(endf,mt)) return 0;

  /* read HEAD record */
  ReadlnString(endf,HL,&MAT,&MF,&MT);
  ReadCONT(endf,&MAT,&MF,&MT,&C1,&C2,&L1,&L2,&N1,&N2);
  if (MT == 0) FatalError(endf,"Unexpected SEND");
  if (MF == 0) FatalError(endf,"Unexpected FEND");
  if (MF != 3) {
    sprintf(HL,"Wrong MF %d",MF);
    FatalError(endf,HL);
  }
  if (outf == OUT_TABLE) {
    printf("<TITLE>%s - %d,%d: %s</TITLE>\n",cxfile,
      3,mt,GetDescription(3,mt,0));
    printf("<B>%d,%d: %s</B><BR><BR>",3,mt,GetDescription(3,mt,0));
    if(C2 != 0.0)
      printf("Reaction Q value is % .5le MeV<BR><BR>\n",C2*1E-6);
    if (endf->LRP == 2 &&
      (mt == 1 || mt == 2 || mt == 3 || mt == 5 || mt == 18 || mt == 102))
      puts("Resonance contributions must be computed from resonance parameters

```



```

and "
    "added to this cross section<BR><BR>");
}
else {
    if(html) {
        if (endf->LRP == 2 &&
            (mt == 1 || mt == 2 || mt == 3 || mt == 5 || mt == 18 || mt == 102))
            puts("Resonance contributions must be computed from resonance parameters
and "
                "added to this cross section");
        }
        else {
            printf("%s\n", GetDescription(3, mt, 0));
            printf("Reaction Q value is %.5le MeV\n", C2*1E-6);
            if (endf->LRP == 2 &&
                (mt == 1 || mt == 2 || mt == 3 || mt == 5 || mt == 18 || mt == 102))
                puts("Resonance contributions must be computed from resonance parameters
and\n"
                    "added to this cross section\n");
        }
    }
    if (mt >= 251 && mt <= 253) m2 = 1.0;
    if (outf == OUT_TABLE) {
        if (mt == 251)
            puts("<TABLE BORDER=2><TR><TH>Energy [MeV]</TH><TH>Average Cosine of
Scattering Angle</TH></TR>");
        else if (mt == 252)
            puts("<TABLE BORDER=2><TR><TH>Energy [MeV]</TH><TH>Average logarithmic
Energy Decrement</TH></TR>");
        else if (mt == 253)
            puts("<TABLE BORDER=2><TR><TH>Energy [MeV]</TH>"
                "<TH>The Everage of the Square of the logarithmic Energy
decrement<br>"
                "for Elastic Scattering, devided by twice the Average logarithmic
decrement<br>"
                "for Elastic Scattering of Neutrons</TH></TR>");
        else
            puts("<TABLE BORDER=2><TR><TH>Energy [MeV]</TH><TH>Cross Section
[mb]</TH></TR>");
    }
    DisplayTAB1(endf, N1, N2, m1, m2);
    if (outf == OUT_TABLE) puts("</TABLE>");
    return 1;
}

int ReadFile4(ENDF *endf, int mt, int msub1, int msub2)
{
    char HL[81];
    double C1, C2;
    int L1, L2, N1, N2, MAT, MF, MT;
    int LVT, LTT;
    char mfmtstr[50], remstr[80];
    char *name[100];

```

```

int i,NK,NM;

if (!LocateFile(endf,4)) return 0;
if (!LocateSection(endf,mt)) return 0;

/* read HEAD record */
ReadCONT(endf,&MAT,&MF,&MT,&C1,&C2,&LVT,&LTT,&N1,&N2);
if(html) {
    printf("<TITLE>%s - %d,%d: %s</TITLE>\n",cxfile,
           4,mt,GetDescription(4,mt,0));
    printf("<B>%d,%d: %s</B><br>\n",4,mt,GetDescription(4,mt,0));
}
else printf("%d,%d: %s\n",4,mt,GetDescription(4,mt,0));
if (LVT == 0) printf("Transformation matrix is not given.");
if (LTT == 0) puts("All angular distributions are isotropic");
else if (LTT == 2) puts("Normalized probability distributions are given.");
ReadCONT(endf,&MAT,&MF,&MT,&C1,&C2,&L1,&L2,&NK,&NM);
if (LVT == 1) {
    if(msub1 == 0) {
        for(i=0;i<1+NM;i++) {
            name[i] = malloc(4);
            sprintf(name[i],"V%d",i);
        }
        DisplayLIST(endf,NK,1+NM,"Transformation maxtrix elements",name);
        for(i=0;i<1+NM;i++) free(name[i]);
    }
    else {
        if(msub1 < 0) {
            if(html) {
                sprintf(mfmtstr,"4+%d+0",mt);
                href(mfmtstr,"Transformation Matrix",0);
                printf("<br>\n");
            }
            else printf("Transformation Matrix given.\n");
        }
        ReadLIST(endf,NK,NULL);
    }
}
if(msub1 < 0) {
    if (LTT == 1) {
        int NE;
        double *B,Ein,MeV;
        int i,NL;
        int j;
        ReadCONT(endf,&MAT,&MF,&MT,&C1,&C2,&L1,&L2,&N1,&NE);
        ReadTAB2(endf,N1,NULL,NULL);
        printf(" %d Legendre polynomial coefficients are given.",NE);
        if(html) printf("<br>\n");
        else printf("\n");
        if(html) {
            printf("<table border=2>\n");
            printf("<tr><th>Ein(MeV)</th><th>LC</th></tr>\n");
        }
    }
}

```

```

else printf("Ein, LC\n");
for (j=0;j<NE;j++) {
  ReadCONT(endf, &MAT, &MF, &MT, &C1, &Ein, &L1, &L2, &NL, &N2);
  MeV=Ein/1000000.0;
  if(html) printf("<tr><td>");
  if(MeV < 1.0) printf("%.5le", MeV);
  else printf("%.3f", MeV);
  if(html) printf("</td>");
  else {
    if(outf == OUT_COMMA) printf(", ");
    else printf(" ");
  }
  B = (double *) malloc(NL*sizeof(double));
  ReadLIST(endf, NL, B);
  for(i=0;i<NL;i++) {
    if(html) printf("<td>");
    printf("%.5le", B[i]);
    if(html) printf("</td>");
    else {
      if(i < NL-1) {
        if(outf == OUT_COMMA) printf(", ");
        else printf(" ");
      }
    }
  }
  if(html) printf("</tr>");
  printf("\n");
  free(B);
}
if(html) printf("</table>");
}
else if (LTT == 2) {
}
}
return 1;
}

```

```

int ReadFile5(ENDF *endf, int mt)
{
  int i;
  char HL[81];
  double C1, C2;
  int L1, L2, N1, N2, MAT, MF, MT;
  int NN1;

  if (!LocateFile(endf, 5)) return 0;
  if (!LocateSection(endf, mt)) return 0;

  /* read HEAD record */
  ReadLnString(endf, HL, &MAT, &MF, &MT);
  PushString();
  while (1) {

```

```

/* read HEAD record */
ReadCONT(endf, &MAT, &MF, &MT, &C1, &C2, &L1, &L2, &N1, &N2);
if (MF == 0) break;
if (MF != 5) {
    sprintf(HL, "Wrong MF %d", MF);
    FatalError(endf, HL);
}
NN1 = N1;
for (i=0; i<NN1; i++) {
    ReadCONT(endf, &MAT, &MF, &MT, &C1, &C2, &L1, &L2, &N1, &N2);
    if (MF == 0) FatalError(endf, "Unexpected FEND");
    if (MF != 5) {
        sprintf(HL, "Wrong MF %d", MF);
        FatalError(endf, HL);
    }
    ReadTAB1(endf, N1, N2, NULL, NULL, NULL, NULL);
    if (L2 == 1) {
        int n;
        int NN2;
        printf(" Arbitrary tabulated function for MT = %d\n", MT);
        ReadCONT(endf, &MAT, &MF, &MT, &C1, &C2, &L1, &L2, &N1, &N2);
        ReadTAB2(endf, N1, NULL, NULL);
        NN2 = N2;
        for (n=0; n<NN2; n++) {
            ReadCONT(endf, &MAT, &MF, &MT, &C1, &C2, &L1, &L2, &N1, &N2);
            ReadTAB1(endf, N1, N2, NULL, NULL, NULL, NULL);
        }
    }
}
else if (L2 == 5) {
    printf(" General evaporation spectrum for MT = %d\n", MT);
    ReadCONT(endf, &MAT, &MF, &MT, &C1, &C2, &L1, &L2, &N1, &N2);
    ReadTAB1(endf, N1, N2, NULL, NULL, NULL, NULL);
    ReadCONT(endf, &MAT, &MF, &MT, &C1, &C2, &L1, &L2, &N1, &N2);
    ReadTAB1(endf, N1, N2, NULL, NULL, NULL, NULL);
}
else if (L2 == 7) {
    printf(" Simple fission spectrum (Maxwellian) for MT = %d\n", MT);
    ReadCONT(endf, &MAT, &MF, &MT, &C1, &C2, &L1, &L2, &N1, &N2);
    ReadTAB1(endf, N1, N2, NULL, NULL, NULL, NULL);
}
else if (L2 == 9) {
    printf(" Evaporation spectrum for MT = %d\n", MT);
    ReadCONT(endf, &MAT, &MF, &MT, &C1, &C2, &L1, &L2, &N1, &N2);
    ReadTAB1(endf, N1, N2, NULL, NULL, NULL, NULL);
}
else if (L2 == 11) {
    printf(" Energy dependent Watt spectrum for MT = %d\n", MT);
    ReadCONT(endf, &MAT, &MF, &MT, &C1, &C2, &L1, &L2, &N1, &N2);
    ReadTAB1(endf, N1, N2, NULL, NULL, NULL, NULL);
    ReadCONT(endf, &MAT, &MF, &MT, &C1, &C2, &L1, &L2, &N1, &N2);
    ReadTAB1(endf, N1, N2, NULL, NULL, NULL, NULL);
}
else if (L2 == 12) {

```

```

        printf(" Energy dependent fission neutron spectrum (Madland and Nix)
for MT = %d\n",MT);
        ReadCONT(endf, &MAT, &MF, &MT, &C1, &C2, &L1, &L2, &N1, &N2);
        ReadTAB1(endf, N1, N2, NULL, NULL, NULL, NULL);
    }
    else {
        sprintf(HL, "Unknown LF %d", L2);
        Warn(endf, HL);
        while (MT != 0) ReadlnString(endf, HL, &MAT, &MF, &MT);
        PushString();
    }
}
    CheckSEND(endf);
}
    if (MF == 0) endf->cur_loc++;
    return 1;
}

```

```

int ReadFile6(ENDF *endf, int mt, int msub1, int msub2)
{
    int i;
    char HL[81];
    double ZA, AWR, C1, C2, E1, A1;
    int idum1, LCT, NK, idum2;
    int NR, NP, LANG, LEP, NE, ND, NA, NW, NEP;
    int L1, L2, N1, N2, MAT, MF, MT;
    int LAW, NN1, NRM, NMU, NRP;
    int n1, n2, n3, ipr;
    double ZAP, AWP;
    int LIP;
    char Lang[20], nucn[30], mfmtstr[50], remstr[80];
    char*name[100];

    if (!LocateFile(endf, 6)) return 0;
    if (!LocateSection(endf, mt)) return 0;

    /* read HEAD record */
    ReadCONT(endf, &MAT, &MF, &MT, &ZA, &AWR, &idum1, &LCT, &NK, &idum2);
    if (html) {
        printf("<TITLE>%s - %d.%d: %s</TITLE>\n", cxfile,
            4, mt, GetDescription(4, mt, 0));
        printf("<B>%d.%d: %s</B><br>\n", 6, mt, GetDescription(4, mt, 0));
    }
    else printf("%d.%d: %s\n", 6, mt, GetDescription(4, mt, 0));
    switch(LCT) {
        case 1: printf("LAB system"); break;
        case 2: printf("CMS angle"); break;
        case 3: printf("CMS (A <= 4), LAB ( A>4)"); break;
        default: printf("*** LCT = %d ***", LCT);
    }
}
    if (html) printf("<br>");
    printf("\n");
    for(n1=0; n1<NK; n1++) {

```

```

ReadCONT(endf, &MAT, &MF, &MT, &ZAP, &AWP, &LIP, &LAW, &NR, &NP);
if (MF == 0) break;
if (MF != 6) {
    sprintf(HL, "Wrong MF %d", MF);
    FatalError(endf, HL);
}
ReadTAB1(endf, NR, NP, NULL, NULL, NULL, NULL);
switch(LAW) {
case 0: printf("Unknown distribution %d\n", n1); break;
case 1:
    nucname(nucn, ZAP, LIP);
    if(msub1 < 0) {
        printf(" Continuum energy-angle distribution for %s.", nucn);
        if(html) printf("<br>\n");
        else printf("\n");
    }
    ReadCONT(endf, &MAT, &MF, &MT, &C1, &C2, &LANG, &LEP, &NR, &NE);
    ReadTAB2(endf, NR, NULL, NULL);
    if(msub1 < 0) {
        if(html) {
            printf("<table><tr><th>No. </th><th>Ein</th></tr>\n");
        }
        else printf("No. Ein\n");
    }
}
for(n2=0; n2<NE; n2++) {
    ReadCONT(endf, &MAT, &MF, &MT, &C1, &E1, &ND, &NA, &NW, &NEP);
    ipr = 0;
    if(msub1 < 0) {
        if(html) {
            sprintf(mfmtstr, "6+%d+%d+%d", MT, n1, n2);
            sprintf(remstr, "%d", n2);
            printf("<tr><td>");
            href(mfmtstr, remstr, 0);
            printf("</td><td>%.5le</td></tr>\n", E1);
        }
        else printf("%d, %.5le\n", n2, E1);
    }
}
else if((msub1 == n1) && (msub2 == n2)) ipr = 1;
if(ipr) {
    printf("Continuum energy-angle distribution ");
    printf("for %s, Ein=%.5le, LANG=%d\n", nucn, E1, LANG);
    name[0] = malloc(5); strcpy(name[0], "Eout");
    name[1] = malloc(5); strcpy(name[1], "B");
    for(i=0; i<NA; i++) {
        name[i] = malloc(5); sprintf(name[i], "B%d", i+1);
    }
    DisplayLIST(endf, NW, 2+NA, "Eout, B", name);
    for(i=0; i<2+NA; i++) free(name[i]);
}
else ReadLIST(endf, NW, NULL);
}
if((msub1 < 0) && html) printf("</table>\n");
break;

```

```

case 2:
    printf(" * Two-body reaction angular distribution for MT = %d\n",MT);
    break;
case 3:
    printf(" * Isotropic two-body distribution for MT = %d\n",MT);
    break;
case 4:
    printf(" * Recoil distribution of two-body reaction for MT = %d\n",MT);
    break;
case 5:
    printf(" * Charged-particle elastic scattering for MT = %d\n",MT);
    break;
case 6:
    printf(" * N-body phase-space distribution for MT = %d\n",MT);
    break;
case 7:
    printf(" Laboratory angle-energy law for MT = %d\n",MT);
    ReadCONT(endf, &MAT, &MF, &MT, &C1, &C2, &idum1, &idum2, &NR, &NE);
    ReadTAB2(endf, NR, NULL, NULL);
    if(msub1 < 0) printf("No. Energy\n");
    for(n2=0; n2<NE; n2++) {
        ReadCONT(endf, &MAT, &MF, &MT, &C1, &E1, &idum1, &idum2, &NRM, &NMU);
        if(msub1 < 0) printf(" *2d % .4le\n", n2, E1);
        else {
            if(n2 == msub1) printf("Ein = % .4le\n", E1);
        }
        ReadTAB2(endf, NRM, NULL, NULL);
        if(n2 == msub1) {
            if(msub2 < 0) printf("No. Acos\n");
        }
        for(n3=0; n3<NMU; n3++) {
            ReadCONT(endf, &MAT, &MF, &MT, &C1, &A1, &idum1, &idum2, &NRP, &NEP);
            ipr = 0;
            if(n2 == msub1) {
                if(msub2 < 0) printf(" *2d % .2f\n", n3, A1);
                else if(n3 == msub2) ipr = 1;
            }
            if(ipr) {
                printf("Acos = % .3f\n", A1);
                printf("Eout(MeV), f\n");
                DisplayTAB1(endf, NRP, NEP, 1.0e-6, 1.0);
            }
            else ReadTAB1(endf, NRP, NEP, NULL, NULL, NULL, NULL);
        }
    }
    break;
default: printf(" *** Unknown LAW=%d ***\n", LAW);
}
/*
while (MT != 0) {
    ReadlnString(endf, HL, &MAT, &MF, &MT);
    if (MF == 0) break;
    if (MF != 6) {

```

```

        sprintf(HL, "Wrong MF %d", MF);
        FatalError(endf, HL);
    }
}
ReadlnString(endf, HL, &MAT, &MF, &MT);
if (MF == 0) break;
*/
)
ReadlnString(endf, HL, &MAT, &MF, &MT);
if (MT != 0) {
    printf("*** Out of sequence MF=%d MT=%d\n", MF, MT);
}
if (MF == 0) endf->cur_loc++;
return 1;
}

int ReadFile7(ENDF *endf, int mt)
{
    int i;
    char HL[81];
    double C1, C2;
    int L1, L2, N1, N2, MAT, MF, MT;
    int NN1;

    if (!LocateFile(endf, 7)) return 0;
    if (!LocateSection(endf, mt)) return 0;

    /* read HEAD record */
    ReadCONT(endf, &MAT, &MF, &MT, &C1, &C2, &L1, &L2, &N1, &N2);
    while (1) {
        if (L1 == 0) printf(" Incoherent inelastic scattering for MT = %d\n", MT);
        else if (L1 == 1) printf(" Coherent elastic scattering for MT = %d\n", MT);
        else if (L1 == 2) printf(" Incoherent elastic scattering for MT = %d\n", MT);
        while (MT != 0) {
            ReadlnString(endf, HL, &MAT, &MF, &MT);
            if (MF == 0) break;
            if (MF != 7) {
                sprintf(HL, "Wrong MF %d", MF);
                FatalError(endf, HL);
            }
        }
        ReadCONT(endf, &MAT, &MF, &MT, &C1, &C2, &L1, &L2, &N1, &N2);
        if (MF == 0) break;
    }
    if (MF == 0) endf->cur_loc++;
    return 1;
}

int ReadFile8(ENDF *endf, int mt)
{
    int i;
    char HL[81];
    double C1, C2;

```



```

int L1,L2,N1,N2,MAT,MF,MT;
int NN1;

if (!LocateFile(endf,8)) return 0;
if (!LocateSection(endf,mt)) return 0;

/* read HEAD record */
ReadCONT(endf,&MAT,&MF,&MT,&C1,&C2,&L1,&L2,&N1,&N2);
while (1) {
    if (N2 == 0) printf(" Complete decay chain given for MT = %d\n",MT);
    else if (N2 == 1) printf(" Decay chain given for MT = 457 in MATP\n");
    while (MT != 0) {
        ReadLnString(endf,HL,&MAT,&MF,&MT);
        if (MF == 0) break;
        if (MF != 8) {
            sprintf(HL,"Wrong MF %d",MF);
            FatalError(endf,HL);
        }
    }
    ReadCONT(endf,&MAT,&MF,&MT,&C1,&C2,&L1,&L2,&N1,&N2);
    if (MF == 0) break;
}
if (MF == 0) endf->cur_loc++;
return 1;
}

int ReadFile(ENDF *endf,int mf,int mt,double m1,double m2,
            int msub1, int msub2)
{
    switch(mf) {
        case 1:
            return ReadFile1(endf,mt);
        case 2:
            return ReadFile2(endf,mt);
        case 3:
            return ReadFile3(endf,mt,m1,m2);
        case 4:
            return ReadFile4(endf,mt,msub1,msub2);
        case 5:
            return ReadFile5(endf,mt);
        case 6:
            return ReadFile6(endf,mt,msub1,msub2);
        case 7:
            return ReadFile7(endf,mt);
        case 8:
            return ReadFile8(endf,mt);
    }
    return 0;
}

int main(argc,argv)
int argc;
char **argv;

```

```

(
char *cp,**parv;
char datafn[200];
int nparv;
ENDF *endf;
int dataf=DF_PRIVATE;
double m1,m2;
int mf;
int mt;
int msub1,msub2;
int i;

html = 0;
if(getenv("REMOTE_HOST") != NULL) html = 1;
m1 = 1E-6;
m2 = 1E3;
if(html) outf = OUT_TABLE;
else outf = OUT_COMMA;
col = 1;
row = 1;

while (--argc) {
    cp = **++argv;
    if (cp[0] == '-') {
        switch (cp[1]) {
            case 'f':
            case 'F':
                if (!strcmp(&cp[2],"endf")) dataf = DF_ENDFB6;
                else if (!strcmp(&cp[2],"jendl")) dataf = DF_JENDL32;
                else if (!strcmp(&cp[2],"jef")) dataf = DF_JEF22;
                break;
            case 'o':
            case 'O':
                if (!strcmp(&cp[2],"comma")) outf = OUT_COMMA;
                else if (!strcmp(&cp[2],"blank")) outf = OUT_BLANK;
                else if (!strcmp(&cp[2],"syk")) outf = OUT_SYLK;
                else if (!strcmp(&cp[2],"table")) outf = OUT_TABLE;
                break;
            case 'e':
            case 'E':
                m1 = GetReal(&cp[2],100);
                break;
            case 'x':
            case 'X':
                m2 = GetReal(&cp[2],100);
                break;
            case 'c':
            case 'C':
                col = GetInteger(&cp[2],100);
                break;
            case 'r':
            case 'R':
                row = GetInteger(&cp[2],100);

```

```

        break;
    }
}
else {
    nparv = argc;
    parv = argv;
    break;
}
}
if(html) {
    if(outf == OUT_TABLE) {
        puts("Content-type: text/html\n\n");
        puts("<BODY LINK=#0000FF VLINK=008888 ALINK=#FF66FF>");
    }
    else puts("Content-type: text/plain\n\n");
}
if(outf != OUT_TABLE) html = 0;
if(nparv < 1) {
    puts("Usage: readendf [-fexcr] endf_file [mf_number [mt_number]]");
    puts("\t-fendf  : read ENDFB/VI file [default]");
    puts("\t-fjendl : read JENDL3.2 file");
    puts("\t-fjef   : read JEF2.2 file");
    puts("\t-ocomma : print output in comma delimited format [default]");
    puts("\t-oblank : print output in blank delimited format");
    puts("\t-osylk  : print output in SYLK format");
    puts("\t-otable : print output in HTML TABLE format");
    puts("\t-enmn   : multiply energy values by nnn [default: 1E-6]");
    puts("\t-xnnn   : multiply cross section values by nnn [default: 1E3]");
    puts("\t-cnnn   : the first column number [default:1]");
    puts("\t-rnnn   : the first row number [default:1]");
    exit(1);
}

strcpy(cxfile, parv[0]);
/* extend to private library */
switch(dataf) {
    case DF_ENDFB6:
        strcpy(datafn, ENDFB6DIR);
        sprintf(dataf_str, "-fendf");
        break;
    case DF_JENDL32:
        strcpy(datafn, JENDL32DIR);
        strcpy(dataf_str, "-fjendl");
        break;
    case DF_JEF22:
        strcpy(datafn, JEF22DIR);
        strcpy(dataf_str, "-fjef");
        break;
    default:
        datafn[0] = '\0';
        strcpy(dataf_str, cxfile);
}
if(dataf_str[0] == '@') {

```

```

    sprintf(datafn, "/user4/kndc/endif/%s", cxfile+1);
}
else strcat(datafn, cxfile);
if ((endif=OpenENDF(datafn)) == NULL) {
    printf("Cannot open ENDF file '%s'\n", datafn);
    exit(1);
}
mf = 0;
mt = -1;
msub1 = -1;
msub2 = -1;
if (nparv > 2) {
    mf = atoi(parv[1]);
    mt = atoi(parv[2]);
    if (mf != 3) {
        m1 = 1.0;
        m2 = 1.0;
    }
}
if (nparv > 3) msub1 = atoi(parv[3]);
if (nparv > 4) msub2 = atoi(parv[4]);
if (nparv > 2) ReadFile(endif, mf, mt, m1, m2, msub1, msub2);
else ReadContents(endif);
CloseENDF(endif);
return 0;
}

```

## 부록 2. MATXS69 색인

index to matxs file

```
*****
*** file matxs          *****
*** vers 69            *****
*** user matxs Kaeri   *****
*****
```

file description

69 wims group neutron only  
Kaeri test case

file data

particle	name	ngrp		
1	n	69		

data type	name	jinp	joutp
1	nscat	1	1
2	ntherm	1	1

material	name	nsub	loca
1	h1	7	0
2	b10	7	45
3	b11	7	109
4	n14	7	159
5	o16	7	217
6	f19	7	257
7	al27	7	335
8	th232	9	409
9	u233	8	483
10	u234	8	532
11	u235	8	587
12	u236	8	696
13	u238	9	793
14	pu239	8	894
15	pu240	8	997

```
-----
h1      sub 1  3.000E+02  1.000E+10  nscat
      vectors: nwt0  nwt1  ntot0  ntot1  nelas  ng      mubar  xi
              gamma  invel
      matrices: nelas
h1      sub 2  3.000E+02  1.000E+04  nscat
h1      sub 3  3.000E+02  1.000E+03  nscat
h1      sub 4  3.000E+02  1.000E+02  nscat
h1      sub 5  3.000E+02  1.000E+01  nscat
h1      sub 6  3.000E+02  1.000E+00  nscat
      vectors: nwt0  nwt1  ntot0  ntot1  nelas  ng
      matrices: nelas
h1      sub 7  3.000E+02  1.000E+10  ntherm
```

vectors: free h2o  
matrices: free h2o

---

b10 sub 1 3.000E+02 1.000E+10 nscat  
vectors: nwt0 nwt1 ntot0 ntot1 nelas ninel n01 n02  
n03 n04 n05a n06a n07 n08a n09a n10a  
n11a n12daa n13a n14daa n15p n16a n17a ng  
np nd na nt2a p00 p01g p02g p03g  
p04ee p05ee a00 a01 mubar xi gamma invel  
matrices: nelas n01 n02 n03 n04 n05a n06a n07  
n08a n09a n10a n11a n12daa n13a n14daa n15p  
n16a n17a  
b10 sub 2 3.000E+02 1.000E+04 nscat  
b10 sub 3 3.000E+02 1.000E+03 nscat  
b10 sub 4 3.000E+02 1.000E+02 nscat  
b10 sub 5 3.000E+02 1.000E+01 nscat  
b10 sub 6 3.000E+02 1.000E+00 nscat  
vectors: nwt0 nwt1 ntot0 ntot1 nelas ng  
matrices: nelas  
b10 sub 7 3.000E+02 1.000E+10 nthern  
vectors: free  
matrices: free

---

b11 sub 1 3.000E+02 1.000E+10 nscat  
vectors: nwt0 nwt1 ntot0 ntot1 nelas ninel n01 n02  
n03 n04 n05 n06 n07 n08 n09 ng  
na mubar xi gamma invel  
matrices: nelas n01 n02 n03 n04 n05 n06 n07  
n08 n09 nna  
b11 sub 2 3.000E+02 1.000E+04 nscat  
b11 sub 3 3.000E+02 1.000E+03 nscat  
b11 sub 4 3.000E+02 1.000E+02 nscat  
b11 sub 5 3.000E+02 1.000E+01 nscat  
b11 sub 6 3.000E+02 1.000E+00 nscat  
vectors: nwt0 nwt1 ntot0 ntot1 nelas ng  
matrices: nelas  
b11 sub 7 3.000E+02 1.000E+10 nthern  
vectors: free  
matrices: free

---

n14 sub 1 3.000E+02 1.000E+10 nscat  
vectors: nwt0 nwt1 ntot0 ntot1 nelas ninel n01 n02  
n03 n04 n05 n06 n07 n08 n09 n10p  
n11p n12 n13p n14p ng np nd nt  
na n2a p00 p01 p02 p03 p04 p05  
p06 d00 d01 d02 d03 t00 t01 a00  
a01 a02 a03 a04 a05 a06 a07 mubar  
xi gamma invel  
matrices: nelas n01 n02 n03 n04 n05 n06 n07  
n08 n09 n10p n11p n12 n13p n14p  
n14 sub 2 3.000E+02 1.000E+02 nscat  
n14 sub 3 3.000E+02 3.000E+01 nscat  
n14 sub 4 3.000E+02 1.000E+01 nscat

n14 sub 5 3.000E+02 1.000E+00 nscat  
n14 sub 6 3.000E+02 1.000E-01 nscat  
vectors: nwt0 nwt1 ntot0 ntot1 nelas ng  
matrices: nelas  
n14 sub 7 3.000E+02 1.000E+10 nthern  
vectors: free  
matrices: free

---

o16 sub 1 3.000E+02 1.000E+10 nscat  
vectors: nwt0 nwt1 ntot0 ntot1 nelas ninel n01ee n02  
n03 n04 n05 ng na a00 a01 a02  
a03 mubar xi gamma invel  
matrices: nelas n01ee n02 n03 n04 n05  
o16 sub 2 3.000E+02 1.000E+02 nscat  
o16 sub 3 3.000E+02 3.000E+01 nscat  
o16 sub 4 3.000E+02 1.000E+01 nscat  
o16 sub 5 3.000E+02 1.000E+00 nscat  
o16 sub 6 3.000E+02 1.000E-01 nscat  
vectors: nwt0 nwt1 ntot0 ntot1 nelas ng  
matrices: nelas  
o16 sub 7 3.000E+02 1.000E+10 nthern  
vectors: free  
matrices: free

---

f19 sub 1 3.000E+02 1.000E+10 nscat  
vectors: nwt0 nwt1 ntot0 ntot1 nelas ninel nna nnp  
n01 n02 n03 n04 n05 n06 n07 n08  
n09 n10 n11 n12 n13 n14 n15 n16  
n17 n18 n19 n20 n21 ncn ng np  
nd nt na mubar xi gamma invel  
matrices: nelas n01 n02 n03 n04 n05 n06 n07  
n08 n09 n10 n11 n12 n13 n14 n15  
n16 n17 n18 n19 n20 n21 nna nnp  
ncn  
f19 sub 2 3.000E+02 1.000E+04 nscat  
f19 sub 3 3.000E+02 1.000E+03 nscat  
f19 sub 4 3.000E+02 1.000E+02 nscat  
f19 sub 5 3.000E+02 1.000E+01 nscat  
f19 sub 6 3.000E+02 1.000E+00 nscat  
vectors: nwt0 nwt1 ntot0 ntot1 nelas ng  
matrices: nelas  
f19 sub 7 3.000E+02 1.000E+10 nthern  
vectors: free  
matrices: free

---

a127 sub 1 3.000E+02 1.000E+10 nscat  
vectors: nwt0 nwt1 ntot0 ntot1 nelas ninel n01g n02g  
n03g n04g n05g n06g n07g n08g n09g n10g  
n11g n12g n13g n14g n15g n16g n17g n18g  
n19g n20g n21g n22g ng np nd na  
mubar xi gamma invel  
matrices: nelas n01g n02g n03g n04g n05g n06g n07g  
n08g n09g n10g n11g n12g n13g n14g n15g

```

n16g  n17g  n18g  n19g  n20g  n21g  n22g
al27  sub  2  3.000E+02  1.000E+04  nscat
al27  sub  3  3.000E+02  1.000E+03  nscat
al27  sub  4  3.000E+02  1.000E+02  nscat
al27  sub  5  3.000E+02  1.000E+01  nscat
al27  sub  6  3.000E+02  1.000E+00  nscat
      vectors: nwt0  nwt1  ntot0  ntot1  nelas  ng
      matrices: nelas
al27  sub  7  3.000E+02  1.000E+10  nthern
      vectors: free
      matrices: free

-----

th232  sub  1  3.000E+02  1.000E+10  nscat
      vectors: nwt0  nwt1  ntot0  ntot1  nelas  ninel  n2n  nftot
              n01  n02  n03  n04  n05  n06  n07  n08
              n09  n10  n11  n12  n13  n14  n15  ncn
              ng  mubar  xi  gamma  inval  nudel  chid
      matrices: nelas  n2n  nftot  n01  n02  n03  n04  n05
              n06  n07  n08  n09  n10  n11  n12  n13
              n14  n15  ncn
th232  sub  2  3.000E+02  1.000E+04  nscat
th232  sub  3  3.000E+02  1.000E+03  nscat
th232  sub  4  3.000E+02  1.000E+02  nscat
th232  sub  5  3.000E+02  3.000E+01  nscat
th232  sub  6  3.000E+02  1.000E+01  nscat
th232  sub  7  3.000E+02  3.000E+00  nscat
th232  sub  8  3.000E+02  1.000E+00  nscat
      vectors: nwt0  nwt1  ntot0  ntot1  nelas  nftot  ng
      matrices: nelas
th232  sub  9  3.000E+02  1.000E+10  nthern
      vectors: free
      matrices: free

-----

u233  sub  1  3.000E+02  1.000E+10  nscat
      vectors: nwt0  nwt1  ntot0  ntot1  nelas  ninel  n2n  nftot
              n01  n02  n03  n04  ncn  ng  mubar  xi
              gamma  inval  nudel  chid
      matrices: nelas  n2n  nftot  n01  n02  n03  n04  ncn
u233  sub  2  3.000E+02  1.000E+04  nscat
u233  sub  3  3.000E+02  1.000E+03  nscat
u233  sub  4  3.000E+02  3.000E+02  nscat
u233  sub  5  3.000E+02  1.000E+02  nscat
u233  sub  6  3.000E+02  3.000E+01  nscat
u233  sub  7  3.000E+02  1.000E+01  nscat
      vectors: nwt0  nwt1  ntot0  ntot1  nelas  nftot  ng
      matrices: nelas
u233  sub  8  3.000E+02  1.000E+10  nthern
      vectors: free
      matrices: free

-----

u234  sub  1  3.000E+02  1.000E+10  nscat
      vectors: nwt0  nwt1  ntot0  ntot1  nelas  ninel  n2n  nftot
              nf  nnf  n01  n02  n03  n04  n05  n06

```



```

      ncn  ng      mubar  xi      gamma  inval  nudel  chid
matrices: nelas  n2n      nf      nnf      n01    n02    n03    n04
          n05    n06
          ncn
u234   sub  2  3.000E+02  1.000E+05  nscat
u234   sub  3  3.000E+02  1.000E+04  nscat
u234   sub  4  3.000E+02  3.000E+03  nscat
u234   sub  5  3.000E+02  1.000E+03  nscat
u234   sub  6  3.000E+02  3.000E+02  nscat
u234   sub  7  3.000E+02  1.000E+02  nscat
      vectors: nwt0  nwt1  ntot0  ntot1  nelas  nftot  nf      ng
      matrices: nelas
u234   sub  8  3.000E+02  1.000E+10  ntherm
      vectors: free
      matrices: free

```

---

```

u235   sub  1  3.000E+02  1.000E+10  nscat
      vectors: nwt0  nwt1  ntot0  ntot1  nelas  ninel  n2n    nftot
          nf      nnf      n2nf  n01    n02    n03    n04    n05
          n06    n07    n08    n09    n10    n11    n12    n13
          n14    n15    n16    n17    n18    n19    n20    n21
          n22    n23    n24    n25    n26    n27    n28    n29
          n30    n31    n32    n33    n34    ncn    ng      mubar
          xi      gamma  inval  nudel  chid
      matrices: nelas  nftot  n01    n02    n03    n04    n05    n06
          n07    n08    n09    n10    n11    n12    n13    n14
          n15    n16    n17    n18    n19    n20    n21    n22
          n23    n24    n25    n26    n27    n28    n29    n30
          n31    n32    n33    n34    n2n    ncn
u235   sub  2  3.000E+02  1.000E+04  nscat
u235   sub  3  3.000E+02  1.000E+03  nscat
u235   sub  4  3.000E+02  3.000E+02  nscat
u235   sub  5  3.000E+02  1.000E+02  nscat
u235   sub  6  3.000E+02  3.000E+01  nscat
u235   sub  7  3.000E+02  1.000E+01  nscat
      vectors: nwt0  nwt1  ntot0  ntot1  nelas  nftot  nf      ng
      matrices: nelas
u235   sub  8  3.000E+02  1.000E+10  ntherm
      vectors: free
      matrices: free

```

---

```

u236   sub  1  3.000E+02  1.000E+10  nscat
      vectors: nwt0  nwt1  ntot0  ntot1  nelas  ninel  n2n    nftot
          nf      nnf      n01    n02    n03    n04    n05    n06
          n07    n08    n09    n10    n11    n12    n13    n14
          n15    n16    n17    n18    n19    n20    n21    n22
          n23    n24    n25    n26    n27    ncn    ng      mubar
          xi      gamma  inval  nudel  chid
      matrices: nelas  n2n      nf      nnf      n01    n02    n03    n04
          n05    n06    n07    n08    n09    n10    n11    n12
          n13    n14    n15    n16    n17    n18    n19    n20
          n21    n22    n23    n24    n25    n26    n27    ncn
u236   sub  2  3.000E+02  1.000E+05  nscat
u236   sub  3  3.000E+02  1.000E+04  nscat
u236   sub  4  3.000E+02  3.000E+03  nscat
u236   sub  5  3.000E+02  1.000E+03  nscat

```

```

u236 sub 6 3.000E+02 3.000E+02 nscat
u236 sub 7 3.000E+02 1.000E+02 nscat
vectors: nwt0 nwt1 ntot0 ntot1 nelas nftot nf ng
matrices: nelas
u236 sub 8 3.000E+02 1.000E+10 nthern
vectors: free
matrices: free

-----

u238 sub 1 3.000E+02 1.000E+10 nscat
vectors: nwt0 nwt1 ntot0 ntot1 nelas ninel n2n nftot
nf nmf n01 n02 n03 n04 n05 n06
n07 n08 n09 n10 n11 n12 n13 n14
n15 n16 n17 n18 n19 n20 n21 n22
n23 n24 n25 n26 n27 ncn ng mubar
xi gamma invel nudel chid
matrices: nelas nf nmf n01 n02 n03 n04 n05
n06 n07 n08 n09 n10 n11 n12 n13
n14 n15 n16 n17 n18 n19 n20 n21
n22 n23 n24 n25 n26 n27 ncn n2n
u238 sub 2 3.000E+02 1.000E+04 nscat
u238 sub 3 3.000E+02 1.000E+03 nscat
u238 sub 4 3.000E+02 1.000E+02 nscat
u238 sub 5 3.000E+02 3.000E+01 nscat
u238 sub 6 3.000E+02 1.000E+01 nscat
u238 sub 7 3.000E+02 3.000E+00 nscat
u238 sub 8 3.000E+02 1.000E+00 nscat
vectors: nwt0 nwt1 ntot0 ntot1 nelas nftot nf ng
matrices: nelas
u238 sub 9 3.000E+02 1.000E+10 nthern
vectors: free
matrices: free

-----

pu239 sub 1 3.000E+02 1.000E+10 nscat
vectors: nwt0 nwt1 ntot0 ntot1 nelas ninel n2n nftot
nf nmf n2nf n01 n02 n03 n04 n05
n06 n07 n08 n09 n10 n11 n12 n13
n14 n15 n16 n17 n18 n19 n20 n21
n22 n23 n24 n25 n26 n27 n28 n29
n30 n31 ncn ng mubar xi gamma invel
nudel chid
matrices: nelas nftot n01 n02 n03 n04 n05 n06
n07 n08 n09 n10 n11 n12 n13 n14
n15 n16 n17 n18 n19 n20 n21 n22
n23 n24 n25 n26 n27 n28 n29 n30
n31 n2n ncn
pu239 sub 2 3.000E+02 1.000E+04 nscat
pu239 sub 3 3.000E+02 1.000E+03 nscat
pu239 sub 4 3.000E+02 3.000E+02 nscat
pu239 sub 5 3.000E+02 1.000E+02 nscat
pu239 sub 6 3.000E+02 3.000E+01 nscat
pu239 sub 7 3.000E+02 1.000E+01 nscat
vectors: nwt0 nwt1 ntot0 ntot1 nelas nftot nf ng
matrices: nelas
pu239 sub 8 3.000E+02 1.000E+10 nthern
vectors: free

```

matrices: free

---

```
pu240 sub 1 3.000E+02 1.000E+10 nscat
vectors: nwt0 nwt1 ntot0 ntot1 nelas ninel n2n nftot
         nf nnf n01 n02 n03 n04 n05 n06
         n07 n08 n09 n10 n11 n12 n13 n14
         n15 n16 n17 n18 n19 ncn ng mubar
matrices: xi gamma invel nudel chid
         nelas n2n nf nnf n01 n02 n03 n04
         n05 n06 n07 n08 n09 n10 n11 n12
         n13 n14 n15 n16 n17 n18 n19 ncn
pu240 sub 2 3.000E+02 1.000E+05 nscat
pu240 sub 3 3.000E+02 1.000E+04 nscat
pu240 sub 4 3.000E+02 3.000E+03 nscat
pu240 sub 5 3.000E+02 1.000E+03 nscat
pu240 sub 6 3.000E+02 3.000E+02 nscat
pu240 sub 7 3.000E+02 1.000E+02 nscat
vectors: nwt0 nwt1 ntot0 ntot1 nelas nftot nf ng
matrices: nelas
pu240 sub 8 3.000E+02 1.000E+10 ntherm
vectors: free
matrices: free
```

## 서 지 정 보 양 식

<b>수행기관보고서번호</b>	위탁기관보고서번호	표준보고서번호	INIS 주제코드
KAERI/RR-1799/97			
<b>제목/부제</b>	핵자료 데이터베이스 개발		
<b>연구책임자 및 부서명</b>	장종화 (핵자료평가 랩)		
<b>연구자 및 부서명</b>	장종화, 김정도, 오수열, 길충섭, 조영식, 박수진, 조인행		
<b>출판지</b>	대전	<b>발행기관</b>	한국원자력연구소
<b>발행년</b>			1998. 1
<b>페이지</b>	73 p.	<b>도표</b>	있음( v ), 없음( )
<b>크기</b>			26 Cm.
<b>참고사항</b>	'97 기관고유사업		
<b>비밀여부</b>	공개(v ), 대외비( ), __ 급비밀	<b>보고서종류</b>	연구보고서
<b>연구위탁기관</b>		<b>계약 번호</b>	
<b>초록 (15-20줄내외)</b>	<p>본 연구에서는 국내외 연구자들이 핵자료를 이용하기 위한 기반인 핵자료 데이터베이스를 구축하여 온라인으로 사용할수 있도록 하기 위하여 다음과 같은 연구를 수행하였다.</p> <ul style="list-style-type: none"> <li>- 핵자료표 웹서버 개량 인터넷에 온라인으로 서비스되는 핵자료표와 단면적도형 작성 웹서버를 일부 개량하였음.</li> <li>- 핵구조자료 DB 구축 1997 연도말의 최신 ENSDF 평가자료로 교체하였음.</li> <li>- 핵반응자료 DB 구축 핵자료 측정자료인 EXFOR 데이터베이스를 도입하고, 검색시스템을 작성하였으며, 각종 평가핵자료 라이브러리를 수집하고 검색 프로그램을 작성하였음.</li> <li>- 균정수 생산 및 검증 열 중성자용 MATXS 균정수 라이브러리를 생산하여 U-235 노심에 대한 검증계산을 수행하였고 소내외 요구에 따라 MCNP 용 단면적을 생산하였음.</li> </ul>		
<b>주제명키워드 (10단어내외)</b>	데이터베이스, 핵자료, 단면적, 인터넷, 핵자료평가, MATXS, MCNP, 균정수 검증		

## BIBLIOGRAPHIC INFORMATION SHEET

<b>Performing Org. Report No.</b>	<b>Sponsoring Org. Report No.</b>	<b>Standard Report No.</b>	<b>INIS Subject Code</b>
KAERI/RR-1799/97			
<b>Title/ Subtitle</b>	Development of Nuclear Data Base System		
<b>Project Manager and Department</b>	Jonghwa CHANG (Nuclear Data Evaluation Lab.)		
<b>Researcher and Department</b>	Jonghwa CHANG, J.D. KIM, S.Y. OH, C.S. GIL, Y.S. CHO, S.J. PARK, I.H. CHO		
<b>Pub. Place</b>	Taejon	<b>Publisher</b>	KAERI
		<b>Pub. Date</b>	Jan. 1998
<b>Page</b>	73 p.	<b>Fig. &amp; Tab.</b>	Yes( o ), No ( )
		<b>Size</b>	26 Cm.
<b>Note</b>			
<b>Classified</b>	Open(o), Restricted( ) ___ Class	<b>Report Type</b>	Research Report
<b>Sponsoring Org.</b>		<b>Contract No.</b>	
<b>Abstract (15-20 Lines)</b>	<p>Following scope of tasks are performed to establish the nuclear data base which is the infra structure for the nuclear related research and industry.</p> <ul style="list-style-type: none"> <li>- Upgrade of "Table of Nuclides" web service and addition of new feature to existing ENDF plotting services.</li> <li>- Establishment of nuclear structure data base. Reconstruction of nuclear sstructure (ENSDF) data base with up-to-date data.</li> <li>- Establishment of nuclear reaction data base. Collection of the experimental reaction database (EXFOR). Collection of the evaluated nuclear reaction data libraries. Development of data retrieval programs.</li> <li>- Group constant library generation and verification Generation of MATXS group constant library for thermal reactor application and verification against U-235 cores. Improvement of the cross section library preparation system for the MCNP code.</li> </ul>		
<b>Subject Keywords (About 10 words)</b>	database, nuclear data, cross-section, internet, nuclear data evaluation, MATXS, MCNP, group constant validation		