Oak Ridge National Laboratory

SCALE Users' Group Tutorials

ORIGEN: Activation Analysis of the Silene Experiment

S. W. D. Hart
W. A. Wieselquist
G. Ilas

Monday, August 27, 2018

ORNL is managed by UT-Battelle, LLC for the US Department of Energy

U.S. DEPARTMENT OF ENERGY

Why Do We Need COUPLE?

COUPLE creates ORIGEN libraries (f33) from fundamental data sources (and transport flux), performing two major functions

1) Converting fundamental decay data into a decay-only f33
2) Updating/adding reaction transition coefficients on an existing f33

$$\frac{d\vec{N}}{dt} = \left(\boldsymbol{A}_{\sigma,n}\mathit{\Phi}_n + \boldsymbol{A}_{\lambda}\right)\vec{N}(t) + \vec{S}_n,$$

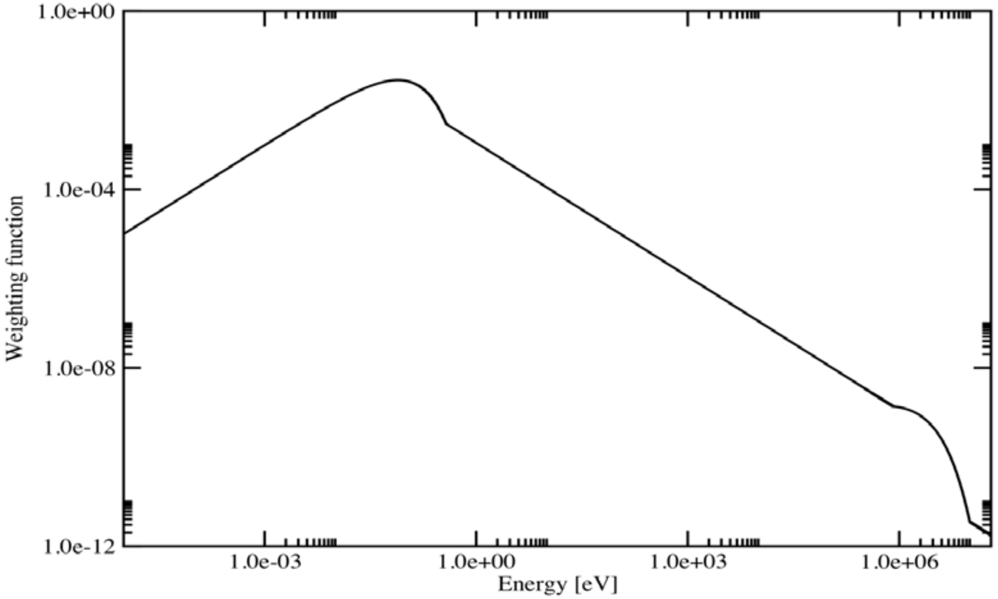
over time step $t_{n-1} \le t \le t_n$.

$$a_{ij} = \begin{cases} l_{ij}\lambda_j + f_{ij}\sigma_j\mathit{\Phi} & i \neq j \\ -\lambda_i - \sigma_i\mathit{\Phi} & otherwise \end{cases}$$

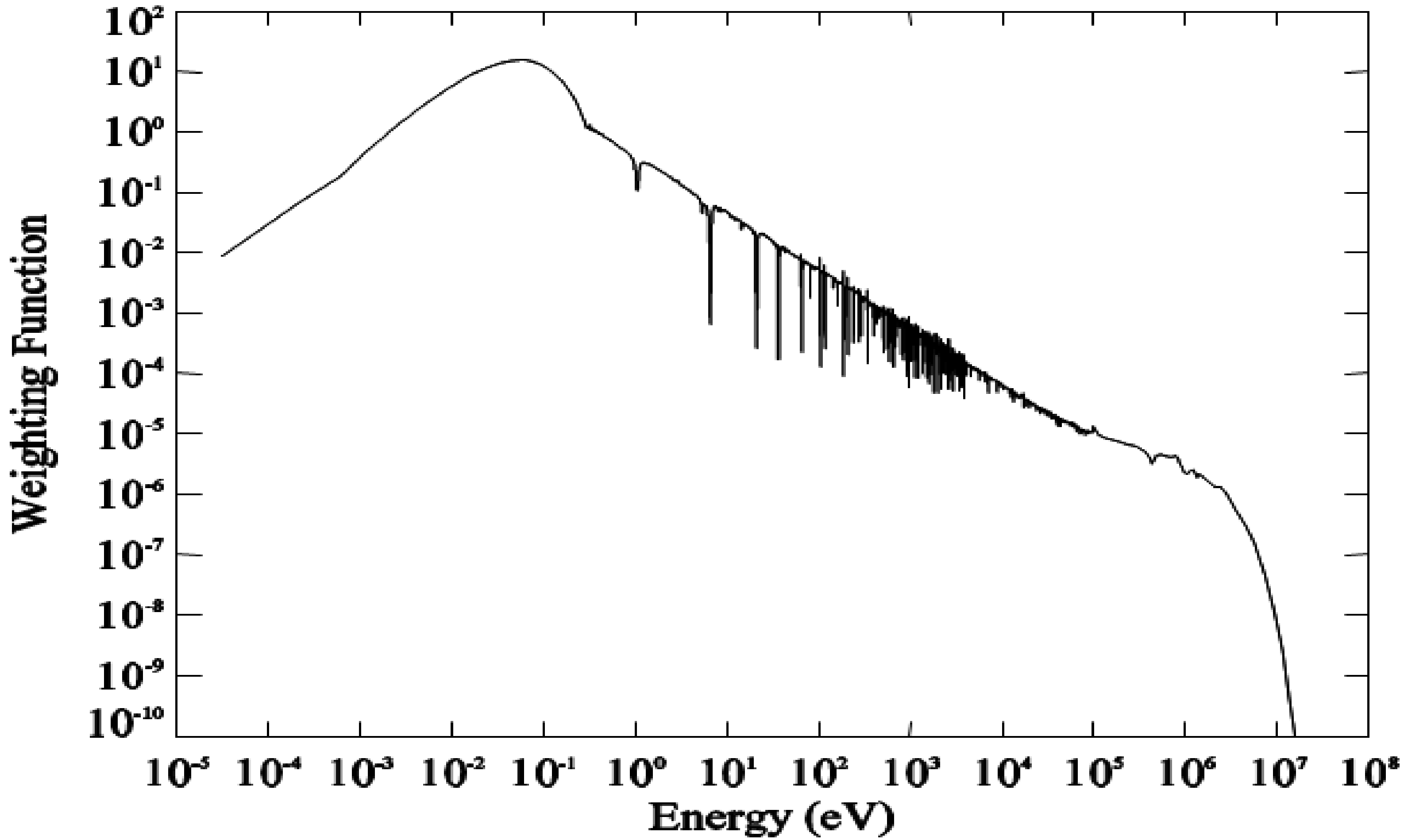
Decay and reaction pieces are stored separately in f33→later in ORIGEN, flux magnitude folded in

Shipped JEFF Libraries

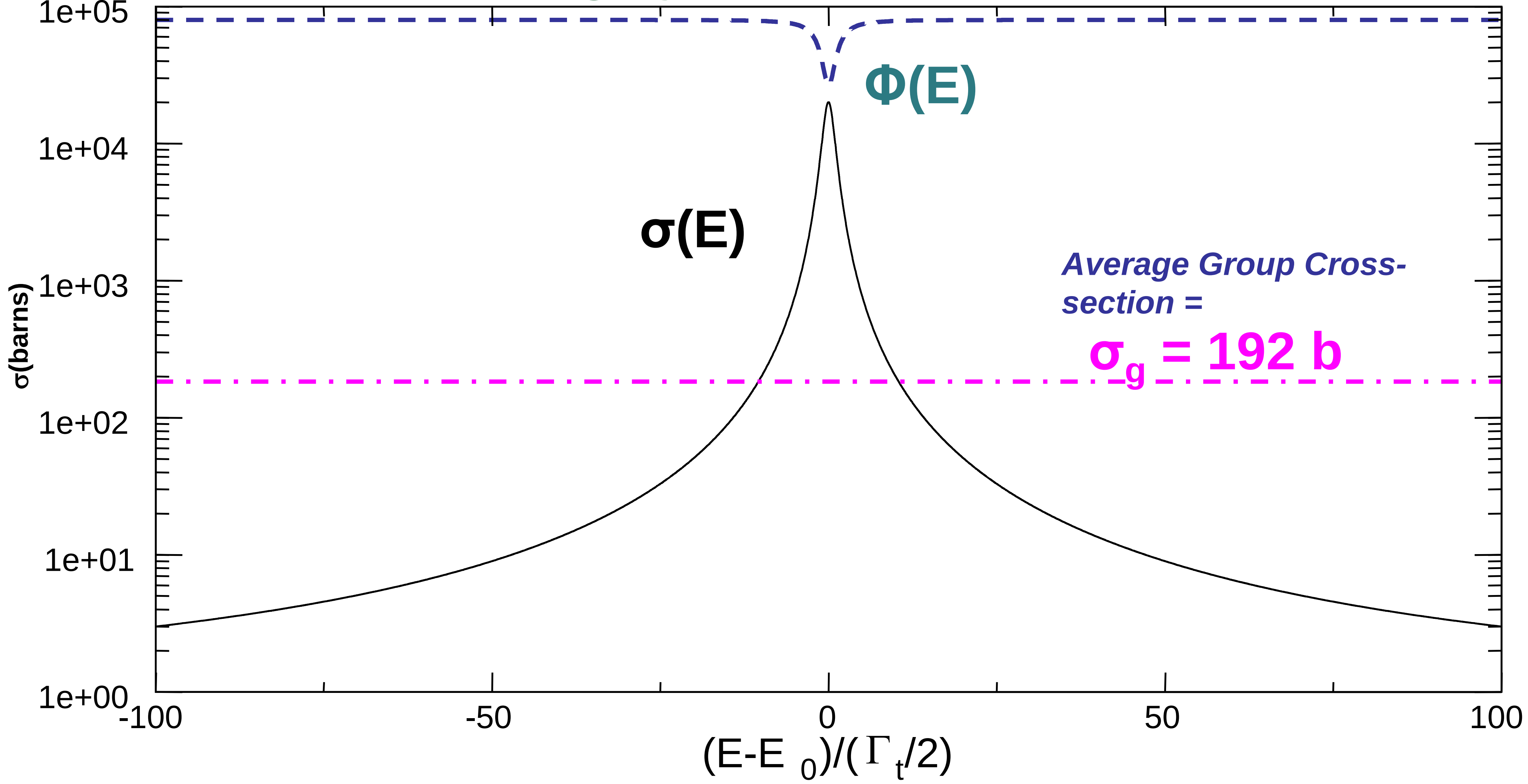
- Infinite dilution (no energy self-shielding)
- **Thermal Maxwellian + 1/E + Fission spectrum** used to create various multigroup libraries (238, 252, 56, etc.)



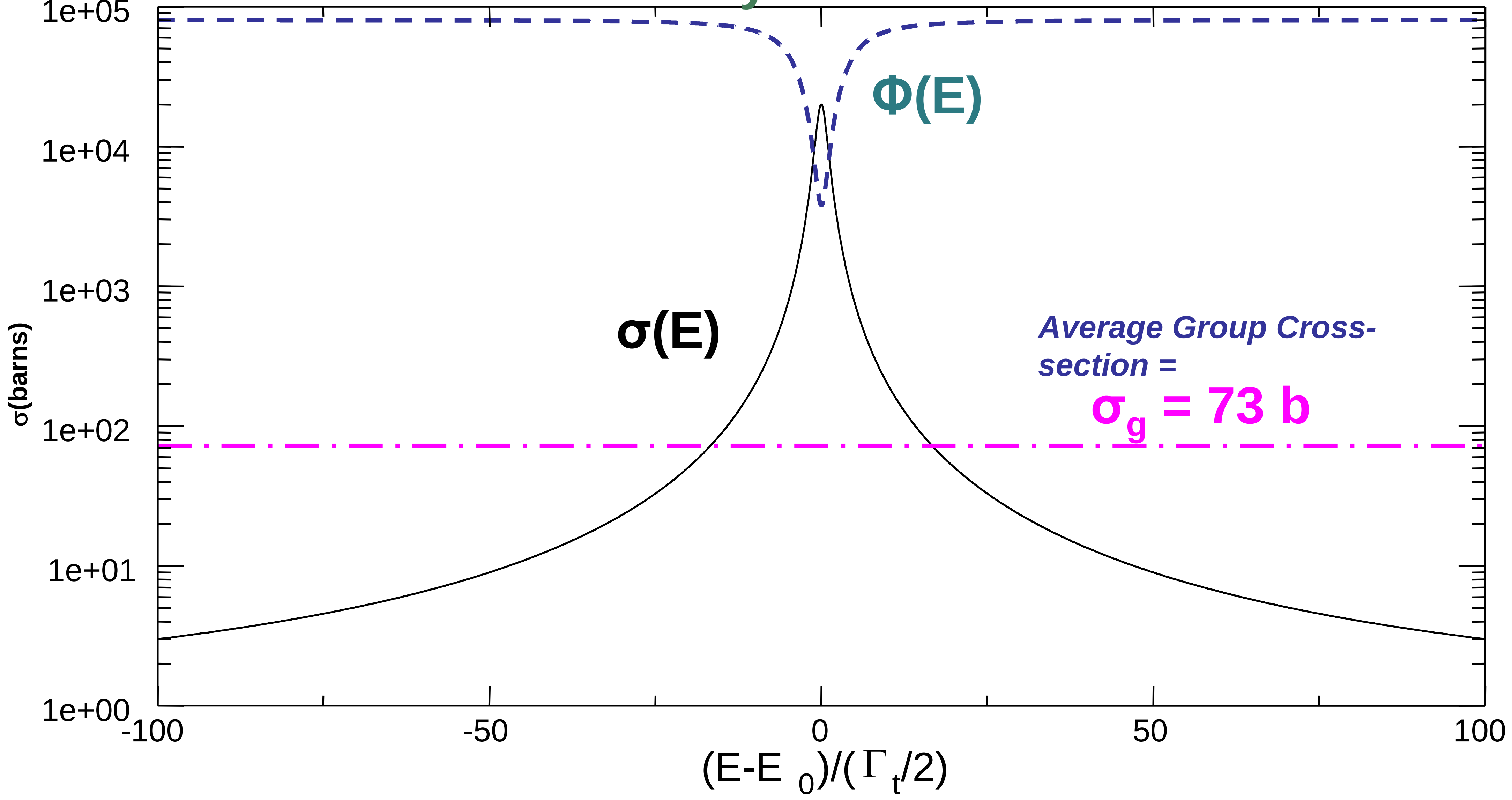
Importance of Self-Shielding



Highly Dilute Absorber



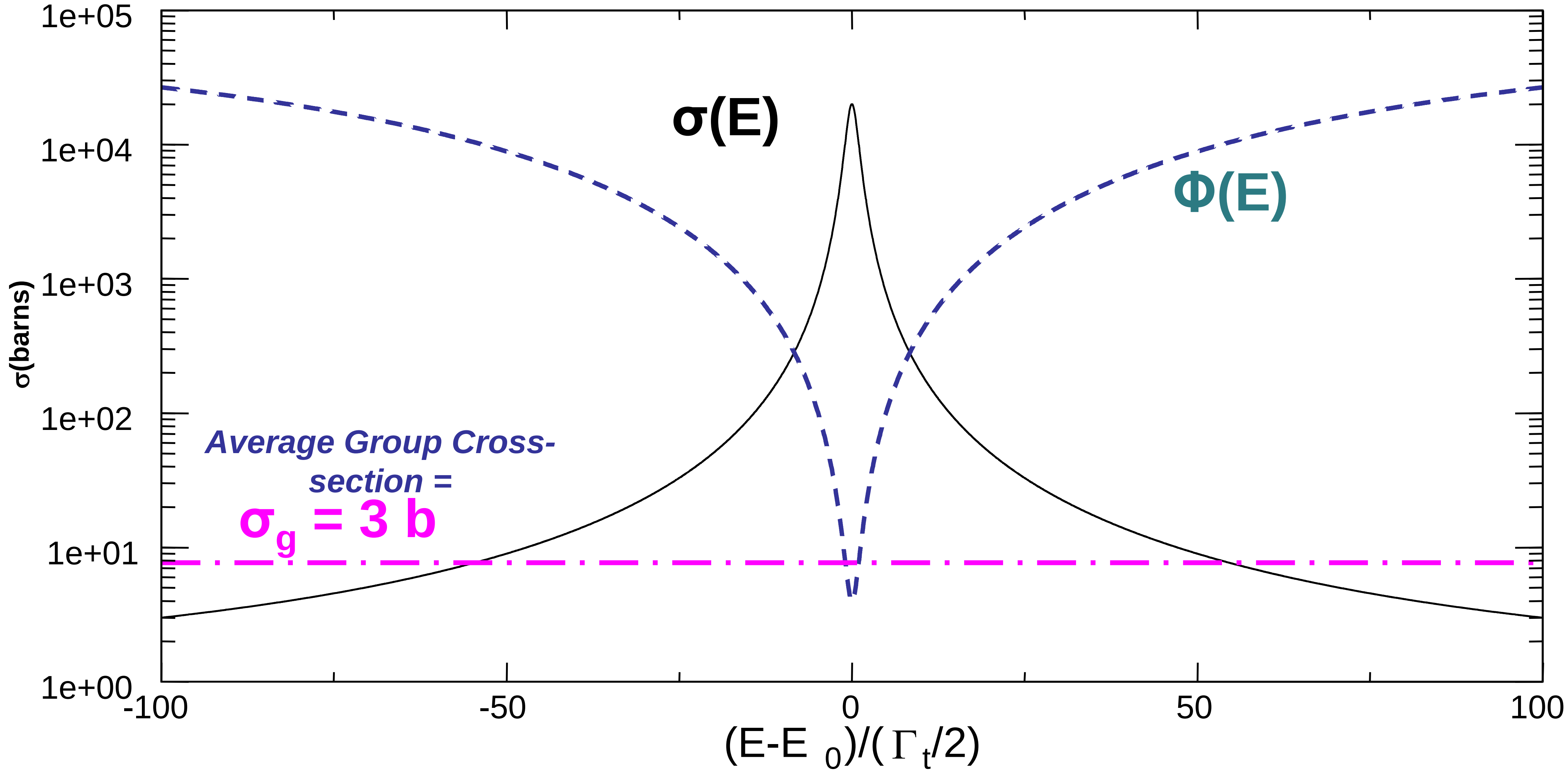
Moderately Dilute Absorber



Average Group Cross-section =

σ_g = 73 b

Undiluted Absorber (i.e., pure absorber)



One Group Cross Sections

reaction
rate

$$R = N\int_0^\infty \sigma(E)\Phi(\mathrm{E})dE$$

number density *N*

$$\approx N\sum_g \sigma^g\Phi^g$$

multi-group
approximation

$$= \frac{N\Phi\sum_g \sigma^g w^g}{\sum_g w^g}$$

multi-group
weighting
function w^g

flux magnitude
Φ (n/cm^{2}s)

$$= N\Phi\sigma$$

COUPLE Input Format (FIDO)

- Define neutron flux spectrum to weight cross sections
- Specify fission nuclides with yields
- Print all weighted cross section values
- User can specify their own cross section values via input

COUPLE Input Cont’d

The first blank line terminates the title input.

0$$ Array – Logical Unit Assignments

1.	NOUT	Printed output unit number (6).
2.	LIBDEC	ORIGEN Decay Resource {if LBIN = 1} (1$ array) (27).
3.	JD	ORIGEN Reaction Resource MG XS library (80=238-group) {always required} (80).
4.	ND	Input ORIGEN Library {if LBIN = 0} (21).
5.	LD	(optional) Self-Shielded AMPX Library (0).
6.	MD	Output ORIGEN Library (33).

COUPLE Input Cont’d

1$ Array – Control Constants [19 entries]

1. LBIN 1/0 – Input library in formatted / binary mode (0).
2. PRT 1/0 – Suppress informational output / print informational output (0).
3. LBUP 1/0 – Update with user input cross sections (Data Block 8) / no user update (0).

12.IDREF – Nuclide ID number of the isotope in AMPX library on unit LD that contains the neutron flux

13.NZONE – The zone of the nuclides with which to add/update the library (0).

14.IEDOU 1/0 – Request edit of data on the output working library on MD (0).

15.NFISW – Number of fissile nuclides in output library entered in 7$ array {JADD = 1} (-1 automatically adds 30 nuclides).

18.NWGT – Defining source of weighting flux spectrum data (0).
/0/N= AMPX working library on IDREF/ input N-group flux in 9* array

T – Data Block 1 terminator

COUPLE Input Cont’d

7$ Array – Nuclide ID numbers for nuclides with fission yields [NFISW entries].

9* Array – User input cross section weighting function [NWGT entries].

T – Data Block 2 terminator

DONE

COUPLE Input Simple Example

```
=couple
ORIGEN Decay Library Revision 3
Prepared by Johnny B. Good on 2015-10-18

1$$ a1 1 e
1t
3$$ a1 0 e
3t
done
end

'Save a copy of the ORIGEN library.
=shell
cp ft33f001 "${OUTDIR}/my_dk.f33"
end
```

COUPLE Input – Things to Remember

- **Formatting**
  - COUPLE uses FIDO→max. line length of 72
  - "done" is required to terminate
  - empty line after title is required to terminate title
- **Group Structures**
  - Reaction Resource (JEFF) and optional AMPX library must have same group structure
  - ${DATA}/origen_filenames gives mapping of "unit number" to name
- **Output**
  - ORIGEN library (f33) is output to working directory as ft33f001
  - Usually a good idea to rename:

```
=shell
cp ft33f001 "${OUTDIR}/my.f33”
end
```

```
=couple

0$$ a2 27 a6 25 e
1$$ 1 0 0 e t
3$$ 0 e 3t
done
end
```

no title still needs a blank/empty line!

ORIGEN Input

- between `=origen` and `end` in SCALE
- syntax is Scale Object Notation (SON)
  - key/value pairs, e.g. `print_xs=yes`
  - arrays, e.g. `power=[40 45 60 20]`
  - blocks, e.g. `lib{ file="end7dec" pos=1 }`
- for backwards compatibility, can still read SCALE 6.1 FIDO syntax

Overview

```
'SCALE comment
=origen

     % ORIGEN comment %

bounds{ … }
solver{ … }
options{ … }

case(A){
    time=[31 365] % days
    …
}

case(B){
    …
}

% more cases?

end
```

- comments
  - inside ORIGEN, SON uses **%** *(anywhere)*
  - outside ORIGEN, SCALE uses **'** *(beginning of line only)*
- "root" level
  - *one of*: bounds, solver, options
  - *sequence of*: case
- main problem specification in cases
  - can have optional id, case(ID)
- hierarchical

Input Component: case

```
=origen
…
case(ID){
    title="my title"

    lib{ … }
    mat{ … }
    time{ … }
    flux{ … } % or power{ … }

    print{ … }
    save{ … }

    gamma{ … }
    neutron{ … }
}
…
end
```

- setup
  - lib – load transition matrix data from library
  - mat – define initial composition
- operating history
  - time – define time steps
  - flux/power – define flux or power level
- archival
  - print – results to output file
  - save – results to f71 binary file
- emission calcs
  - gamma (beta), neutron (alpha)

Input Component: lib

```
=origen
…
case(initial){
    lib{ file="my.f33" pos=1 }
    mat{ … }
    …
}
case(continuation){
    lib{ pos=2 }
    …
}
…
end
```

- library files can contain multiple (usually burnup-dependent) "positions"
- default is pos=1 (will always exist on any library)
- file="end7dec" is a special decay library
- ORIGEN libraries can be created by
  - COUPLE
  - ARP
- position/data set fixed for each case

Input component: mat

```
% from iso
mat{
    iso=[ u235=1.0 u238=9.0 ] %id(sublib)=amount
    units=GRAMS               %units in iso array
}

% from iso with number density input
mat{
    iso=[ u235=1e-2 u238=1e-1 ] %id(sublib)=amount
    units=ATOMS-PER-BARN-CM     %units in iso array
    volume=200                  %cm^3
}

% from position on f71 file
mat{
    load{ file="origen.f71" pos=11 }
}

% from previous case
mat{
    previous=4 %step index from previous case
}
```

- iso array
  - list of id=amount
  - units in:
    - GRAMS
    - MOLES or GRAM-ATOMS
    - ATOMS-PER-BARN-CM
    - CURIES
  - ATOMS-PER-BARN-CM requires volume
- load from position on the binary concentrations file f71
  - not the same as library (f33)!
- load from previous case

Input Component: time

```
% simple decay case (two steps 0→31 and 31→365 days)
time = [ 31 365 ]

% flux irradiation (decay if flux=0)
time = [ 31   365  396 ]
flux = [ 2e14 1e14 0 ]

% power irradiation (decay if power=0)
time = [ 31 365 396 ]
power= [ 50 45  0 ] %50 MW, 45 MW, then decay

% changing units using time block
time{
    t = [ 5 15 300 ]
    units = HOURS
    % available units:
    % SECONDS, MINUTES, HOURS, DAYS, YEARS, CUSTOM
}

% custom units
time{
    t = [ 1 2 3 ]
    units = CUSTOM
    custom_name = "MONTH"
    custom_length = 2678400 %seconds per "MONTH"
}
```

- operating history
  - **time – define time steps**
  - flux/power – define flux or power level
- time can be

  a simple array [] of time steps with default units of days

  OR

  a block with a t=[] array and units=
  - SECONDS
  - MINUTES
  - HOURS
  - DAYS
  - YEARS
  - CUSTOM!

Input Component: flux/power

```
% simple decay case (two steps 0→31 and 31→365 days)
time = [ 31 365 ]

% flux irradiation (decay if flux=0)
time = [ 31   365  396 ]
flux = [ 2e14 1e14 0 ]

% power irradiation (decay if power=0)
time = [ 31 365 396 ]
power= [ 50 45  0 ] %50 MW, 45 MW, then decay

% changing units using time block
time{
    t = [ 5 15 300 ]
    units = HOURS
    % available units:
    % SECONDS, MINUTES, HOURS, DAYS, YEARS, CUSTOM
}

% custom units
time{
    t = [ 1 2 3 ]
    units = CUSTOM
    custom_name = "MONTH"
    custom_length = 2678400 %seconds per "MONTH"
}
```

- operating history
  - time – define time steps
  - **flux/power – define flux or power level**
- either flux or power (not both)
- total flux (neutrons/cm^{2}s) is required to solve depletion equations
- power (MW) can be provided
  - converted internally to flux
- must have same number of time and flux/power entries

Note: for fuel, it is convenient to specify 1e6 total grams of heavy metal so power is MW/MTIHM

Input Component: save

```
case{
    mat{ … }
    time=[ 1 10 100 1000 ] % 4 steps
    save{
        file="my.f71"      % file name
        steps=[0 2 4]      % save initial (0) isotopics
                           % end-of-step 2 (10 days)
                           % end-of-step 4 (1000 days)
    }
}
```

```
=origen
case{
    % use ENDF/VII-based decay library
    lib{ file="end7dec" }
    % create a material with 1 gram U-238
    mat{
        units=grams
        iso=[u238=1.0]
    }
    time=[20L 1.0 1e9] %default units are days

    % save to f71
    save{
        file="u238.f71"
        steps=ALL
    } %save=yes is shortcut for ALL, ${OUTPUTBASENAME}.f71
}
end
```

- save block requires
  - file name, file="xyz.f71"
  - array of step indices, steps=[]
- each value in the time array indicates a "step" from the previous time
  - the first step is from 0 days to 1 day
  - second is from 1 day to 10 days
  - etc.
- using log interpolation L and linear interpolation `I` is convenient for time and steps, respectively
  - for *x*L A B logarithmic spacing, total number of steps is x+2

Input Component: save (2)

```
=origen
case(irrad){
    lib{
        file="mylib" pos=1
    }
    % 1 MT of enriched uranium
    mat{
        units=GRAMS
        iso=[u234=356 u235=40000 u236=184 u238=959460]
    }
    time=[ 50 100 150 200 ] %time in default days
    power=[ 15 15 15 15 ] %power in MW
}
case(decay){
    time{
        units=YEARS
        start=0
        t=[0.1 0.3 0.9 1 2 3 4 5]
    }
    save{ file="discharge" steps=[0 LAST] }
    %only save begin and end in this case
}
end

'must copy back f71 explicitly because doesn’t match *.f71
=shell
cp discharge ${INPDIR}/discharge
end
```

- f71 file is ORIGEN's "results" file
  - binary ☹
  - can post-process with OPUS ☺
  - initialize ORIGEN material with it ☺
  - view in Fulcrum ☺☺
  - also called "concentrations" file
  - contains concentrations *and* spectra
- SCALE rule: all files with extension ".f71" or ".f33" will be copied out of the working directory back to the user directory
- to avoid automatic copy back of either type of file, can use filenames ft71f001 and ft33f001 (legacy SCALE names)
- shortcuts
  - LAST for last step index
  - steps=ALL for everything

Tutorial Problem – SILENE Experiment

- CEA Valduc experimental facility, France
- Critical pulse 93% ^{235}U uranyl nitrate solution
- Threshold activation foil measurements

Measurements

- Aluminum holders have 3 foil cavities containing Fe, Ni, and Mg foils with > 99% elemental purity
  - Nickel, 1 mm thick, 2.807 g
  - Iron, 3 mm thick, 7.292 g
  - Magnesium, 2 mm thick, 1.089 g
- Activation from fast (*n,p*) reactions were measured
- Measured activities per unit mass and relative uncertainties (2 sigma)
  - ^{54}Fe(n,p)^{54}Mn, ^{54}Mn activity 0.1961 Bq/g ± 4.1%
  - ^{58}Ni(n,p)^{58}Co, ^{58}Co activity 12.99 Bq/g ± 3.2%
  - ^{24}Mg(n,p)^{24}Na, ^{24}Na activity 59.10 Bq/g ± 4.1%

Description

Assumptions

- neutron energy spectrum provided
- natural isotopic abundance of foil elements
- total fluence of 1.6351×10^{11} n/cm^2

Goals

- calculate the specific activity (Bq/g) for each foil
- observe time-dependence post-irradiation
- compare to experiment plus uncertainty

Procedure

- Begin with silene.starter.inp
- Copy/paste in flux.txt to finish COUPLE library setup.
- Create an ORIGEN activation input with 3 cases, 1 for each sample
  - Note that starter has outline and follow-on OPUS post-processing
  - Make 'iron' case work first
  - Then copy/paste/modify for the 'nickel' and 'magnesium' cases
  - Uncomment the OPUS inputs for 'nickel' and 'magnesium'
- *Alternatively, you could open each .f71 in Fulcrum and use OPUS F71 Viewer*