# User's Manual
# NTP-ERSN: A new package for solving the multigroup neutron transport equation in a slab geometry, Vesrion ~~1.1~~ 1.1


*M. Lahdour*[*,~~1~~], T. El Bardouni[~~1~~], ~~E. Chakir[2]~~*

[1]*Radiations and Nuclear Systems Laboratory, University Abdelmalek Essaadi,*
*Faculty of Sciences of Tetuan, Morocco*

[2]~~*SIMO-LAB, Faculty of sciences, Ibn Tofail University, Kenitra, Morocco*~~

*Corresponding Address:*
*mlahdour@uae.ac.ma


XXXXXXXXXXXXXXXXXXXXXXXXXX
October 8, 2018

**Introduction**

Welcome to the NTP-ERSN User's Guide! This tutorial will describe the essential aspects to perform multigroup neutron transport equation physics calculations by using NTP-ERSN code, available for download at xxxxxxxxxxxxxxxxxxxxxxxxx.

## Contents


**1 Installation and Quick start** **2**

1.1 Obtaining the source . . . . . . . . . . . . . . . . . . 2

1.2 Installing Prerequisites on Ubuntu . . . . . . . . . . . 2

**2 Examples and test suite** **3**

2.1 Writing JSON Input Files . . . . . . . . . . . . . . . . 3

2.2 Generating JSON Input Files . . . . . . . . . . . . . . 3

**3 Running NTP-ERSN** **5**

3.1 Running NTP-ERSN under a GUI . . . . . . . . . . 5

3.2 Flux Visualization . . . . . . . . . . . . . . . . . . . . 6

3.3 Geometry Visualization . . . . . . . . . . . . . . . . . 7

3.4 Simple Output . . . . . . . . . . . . . . . . . . . . . . 8

# 1 Installation and Quick start

## 1.1 Obtaining the source

All NTP-ERSN source code with its GUI is hosted on GitHub and it can be downloaded for free. You can download the source code directly from GitHub or, if you have the Git version control software installed on your computer, you can use git to obtain the source code. The latter method has the benefit that it is easy to receive updates directly from the GitHub repository. GitHub has a good set of instructions for how to set up git to work with GitHub since this involves setting up ssh keys. With git installed and setup, the following command will download the full source code from the GitHub repository:

```
git clone https://github.com/mit-crgp/NTP-ERSN.git
```

*git clone https://github.com/mohamedlahdour/NTP-ERSN.git*

## 1.2 Installing Prerequisites on Ubuntu

1 untar the archive NTP-ERSN.tar

In the case where the source code was downlaod directly from GitHub, you can untar the archive and executed the installation script (NTP-ERSN-latest.sh). A second case, the source code can be downlaod by using git if you have the Git version control software installed on your computer.

The script (NTP-ERSN-latest.sh) will install NTP-ERSN with its GUI and its dependencies on the system. Install script is available for Ubuntu operating system version 17.10 and all the latest versions.

2 In your system, run the commands below.

```
cd NTP-ERSN/script
chmod +x NTP_ERSN_latest.sh
sh NTP_ERSN_latest.sh
```

This NTP-ERSN and its GUI has been released under the MIT GPL license. Since, the software is a Python-based application; it requires a Python Runtime Environment to work correctly.

3 You can started the software by typing the following command-line argument in a Linux terminal:

```
python interface.py
```

Then a Main window (GUI) of NTP-ERSN package on Ubuntu Linux machine well be displayed as in Figure 1.

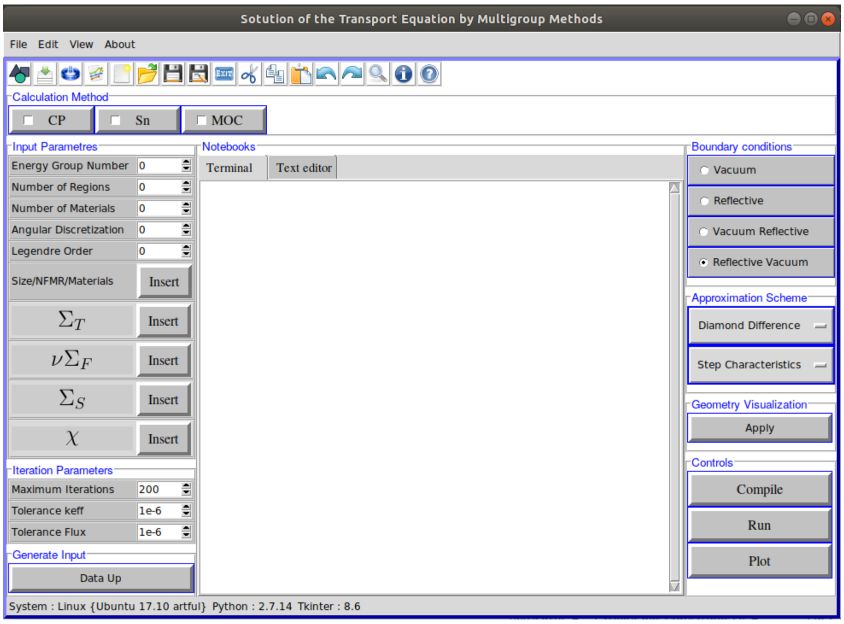


Figure 1: Main window of the GUI of the NTP-ERSN package

## 2 Examples and test suite

### 2.1 Writing JSON Input Files

The input data file must be in JSON format. The first method is to write it directly in Text editor as is shown in the figure 2 witch illustrates an example input file of a critical Benchmark taken from [Avneet Sood(2003)]. This benchmark contains one region and one energy groups.

### 2.2 Generating JSON Input Files

Another method is to use a set of buttons on the left side of the main window 3, these buttons allowing users to insert input data automatically without requiring an in-depth knowledge of JSON file syntax. Once the user click on Data Up button the input file will be automatically generated in the window Text editor. The figure 4 represents the generated input file named INPUT.JSON contains another example of a Criticality Verification Suite Problems taken from [Avneet Sood(2003)]. This benchmark contains three region and two energy groups with scattering source isotropy The parameters of this benchmark are written in INPUT.JSON input file as follows: We define the region number, material number and the scattering order of the medium where the neutron is moving. Then the size of each region (coarse mesh), within which the number of fine meshes is defined. To each region, is assigned a material index. The fission spectrum and the total, diffusion and fission cross sections have values for different materials. Each material can be placed in any region using

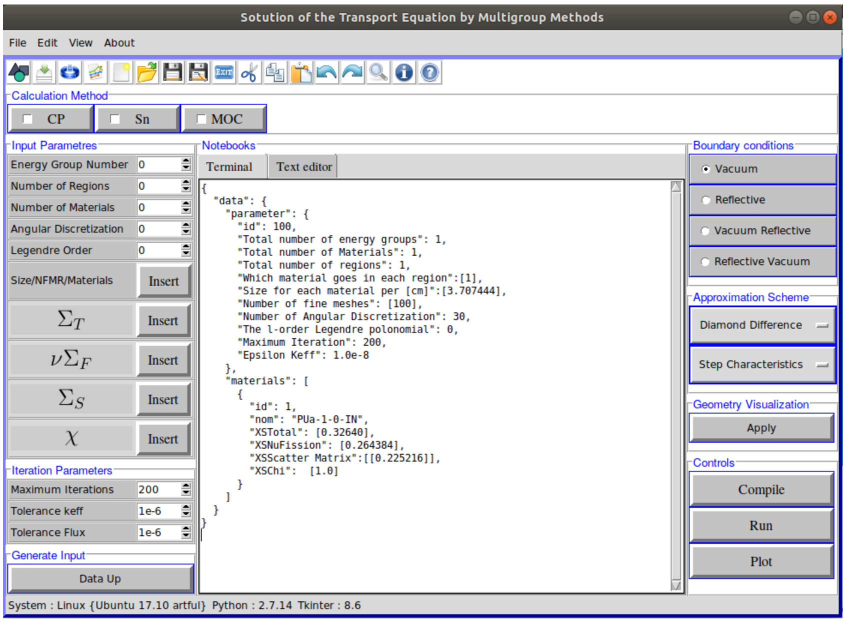


Figure 2: Setting Up Input File for slab geometry in one energy group

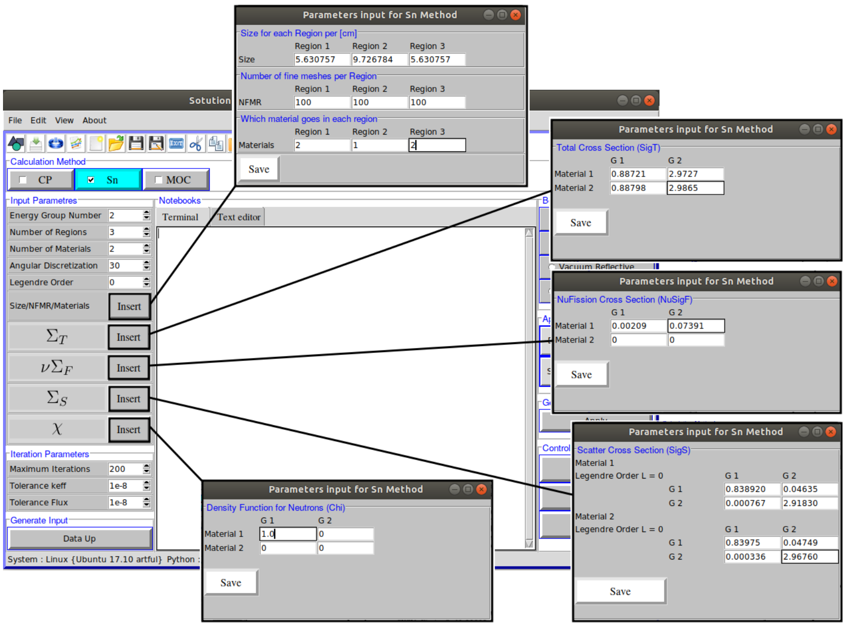


Figure 3: Example of inserting input data in GUI

the appropriate index. This input format allows one to investigate fairly easily a wide range of slab compositions.

```
{
  "data": {
    "parameter": {
      "id": 100,
      "Total number of energy groups": 2,
      "Total number of Materials": 2,
      "Total number of regions": 3,
      "Which material goes in each region": [2, 1, 2],
      "Size for each material per [cm]": [5.630757, 9.726784, 5.630757],
      "Number of fine meshes": [5, 5, 5],
      "Number of Angular Discretization": 8,
      "The l-order Legendre polonomial": 0,
      "Maximum Iteration": 200,
      "Epsilon Keff": 1e-8
    },
    "materials": [
      {
        "id": 1,
        "nom": "material 1",
        "XSTotal": [0.88721, 2.9727],
        "XSNuFission": [0.00209, 0.07391],
        "XSScatter Matrix":[[[0.838920, 0.04635],
                             [0.000767, 2.91830]]],
        "XSChi":  [1.0, 0.0]
      },
      {
        "id": 2,
        "nom": "material 2",
        "XSTotal": [0.88798, 2.9865],
        "XSNuFission": [0, 0],
        "XSScatter Matrix":[[[0.83975, 0.04749],
                             [0.000336, 2.96760]]],
        "XSChi":  [0, 0]
      }
    ]
  }
}
```

Figure 4: INPUT.JSON input file

# 3 Running NTP-ERSN

## 3.1 Running NTP-ERSN under a GUI

Once you have created your input file (*.json), it is relatively straightforward to run the code. the steps of execution are:

- Set up input file as described in Section 2.1, 2.2 or import one of the existing input files into the */tests* folder using the Open data file button on the menu bar. you will find in this folder a set of test cases named (* .json) taken from [Avneet Sood(2003)] (see figure 5).
- Select one of the CP, $S_N$ or MOC methods in Calculation Method.
- One-click on the button Compile, a Python C/API extension modules to call Fortran 90/95 modules subroutines will be created.
- Check Boundary Condition radio button.
- Check Approximation Scheme radio button for the case where $S_N$ or MOC methods are selected.
- Click Run button for solve multigroup scheme.
- When completed with criticality calculation, the software will display the message "finished running case" (see figure 6).

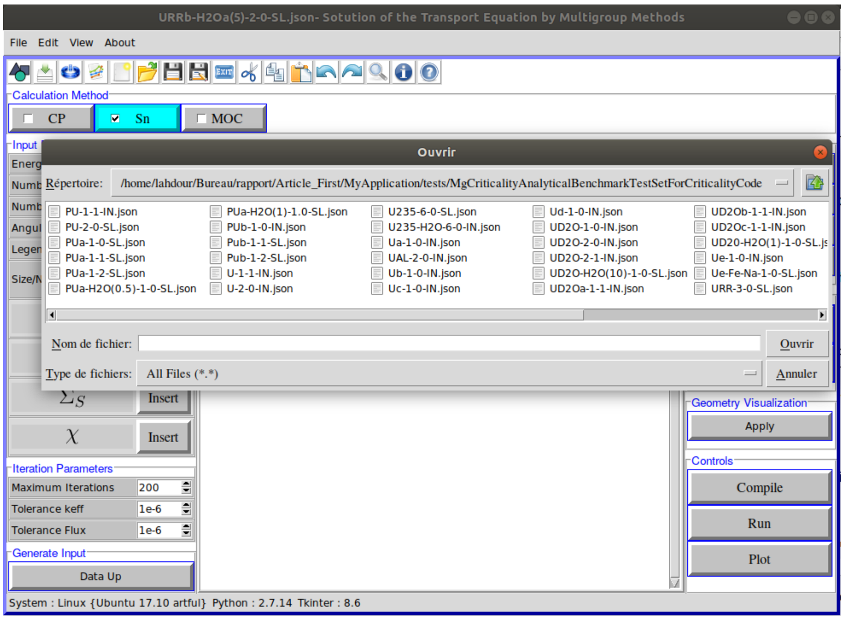


Figure 5: Import input file

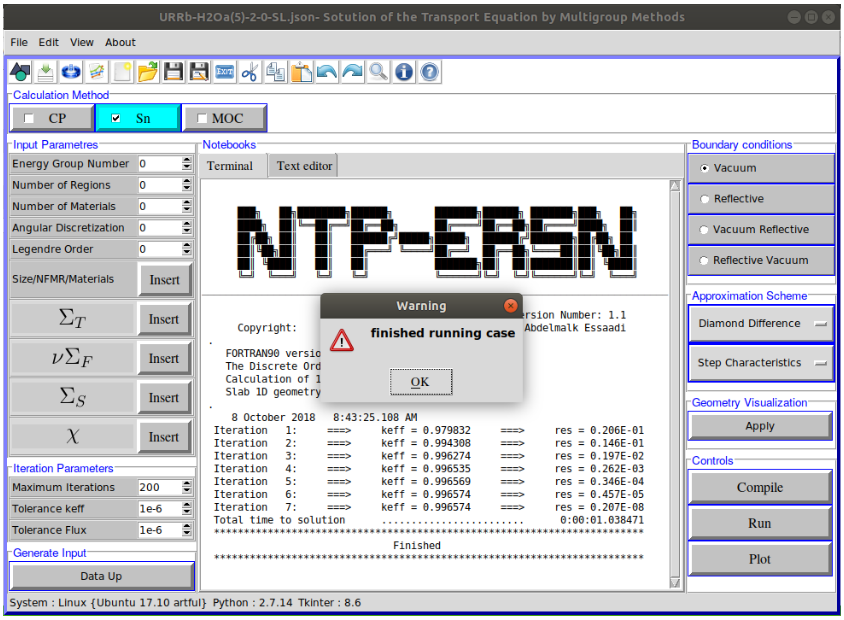


Figure 6: Finished running case

## 3.2 Flux Visualization

The Plot button refers to a set of routines to plot the scalar flux in space and in each energy group. The figure 7 shows the flux for the example input file of figure 4 with three regions and two materials after clicking on the Plot button.

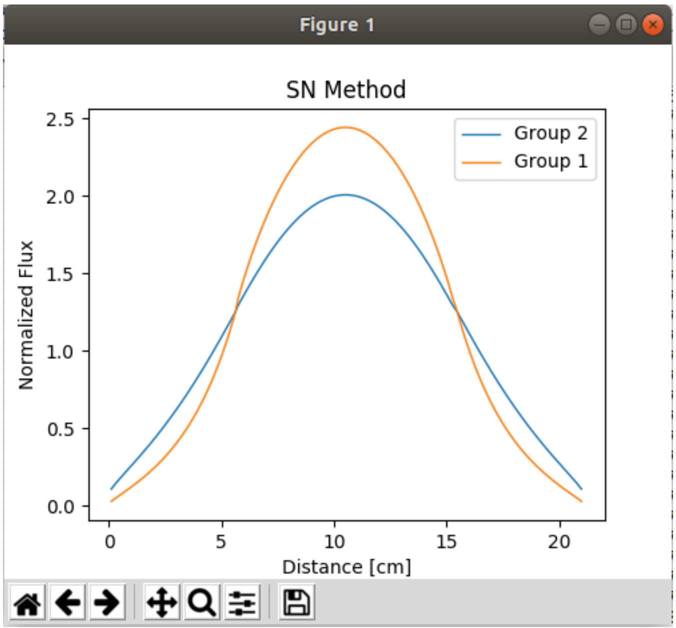


Figure 7: The fast and thermal fluxe in the URRb-H2Oa(5)-2-0-SL benchmark problem.

### 3.3 Geometry Visualization

To plot the geometry, click the Apply button.
A depiction of the geometry for the example input file of figure 4 with three regions and two materials is illustrated in Figure 8.

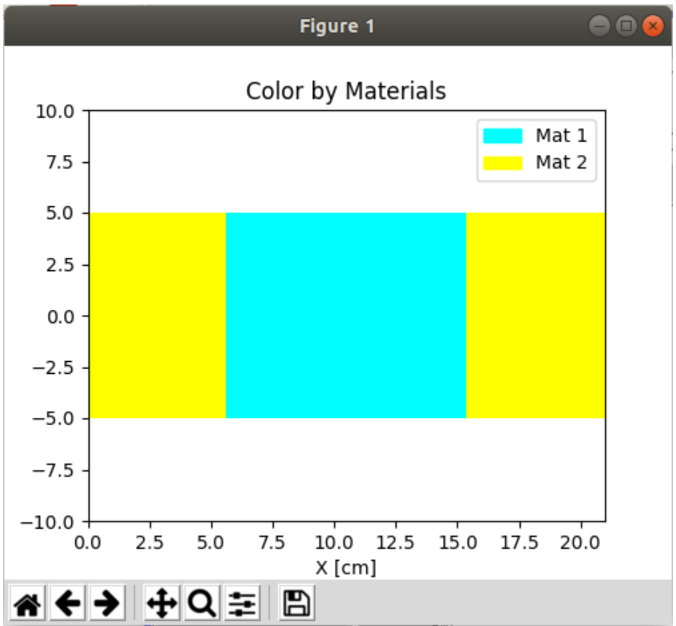


Figure 8: A slab geometry colored by material.

## 3.4 Simple Output

The following is the corresponding output to the above case. The contents of the output file are briefly described: version and run time information, input parameter - values from input, calculation run-time parametres method, scalar flux solution and output parameter - solution to transport eqaution.

```
***************************************************************************
ERSN, UNIVERSITY ABDELMALEK ESSAADI FACULTY OF SCIENCES - TETOUAN, MOROCCO
CODE DEVELOPED BY MOHAMED LAHDOUR, PHD STUDENT
NTP-ERSN:        SN DISCRETE ORDINATES METHOD
VERSION NUMBER:  1.1
VERSION DATE:    8 OTOBER 2018
RAN ON:          2018-10-07 13:44:25.78   (H/M/S)
***************************************************************************
               ____________________________________________
                  INPUT PARAMETER - VALUES FROM INPUT
               ____________________________________________

ENERGY GROUP NUMBER:                        2
REGIONS NUMBER:                             3
MATERIALS NUMBER:                           2
SIZE FOR EACH MATERIAL PER [CM]:      5.63076   9.72678   5.63076
DISCRETIZATIONS ANGULAR:                    8
ORDER LEGENDRE POLONOMIAL:                  0
TOTAL NUMBER OF FINE MESHES:               15
TOLERANCE KEFF AND FLUX:              1.0E-08
               ____________________________________________
                  CALCULATION RUN-TIME PARANETERS SN
               ____________________________________________

GAUSS LEGENDRE QUADRATURE POINTS AND WEIGHTS:

     N. GAUSS         POINTS          WEIGHTS

        1          -9.60290E-01    1.01229E-01
        2          -7.96666E-01    2.22381E-01
        3          -5.25532E-01    3.13707E-01
        4          -1.83435E-01    3.62684E-01
        5           1.83435E-01    3.62684E-01
        6           5.25532E-01    3.13707E-01
        7           7.96666E-01    2.22381E-01
        8           9.60290E-01    1.01229E-01

PSEUDO CROSS SECTIONS DATA:

      MATERIAL :  1

      GROUP        TOTAL         ABSORPTION     NU*FISSION     SCATTERING

        1       8.87210E-01    4.82900E-02    2.09000E-03    8.38920E-01
        2       2.97270E+00    5.44000E-02    7.39100E-02    2.91830E+00
      MATERIAL :  2

      GROUP        TOTAL         ABSORPTION     NU*FISSION     SCATTERING

        1       8.87980E-01    4.82300E-02    0.00000E+00    8.39750E-01
        2       2.98650E+00    1.89000E-02    0.00000E+00    2.96760E+00
               ____________________________________________
                          SCALAR FLUX SOLUTION
               ____________________________________________

FLUXES PER MESH PER ENERGY GROUP:

     M E S H       G R O U P 1     G R O U P 2

        1          1.95059E-01     7.97230E-02
        2          3.90662E-01     2.10169E-01
        3          6.03844E-01     3.85998E-01
        4          8.53613E-01     6.45659E-01
        5          1.13254E+00     1.05059E+00
        6          1.50142E+00     1.69279E+00
        7          1.85024E+00     2.24946E+00
        8          1.97499E+00     2.41389E+00
        9          1.84576E+00     2.23982E+00
       10          1.49487E+00     1.68037E+00
       11          1.12632E+00     1.04146E+00
       12          8.48570E-01     6.40041E-01
       13          6.00118E-01     3.82654E-01
       14          3.88189E-01     2.08346E-01
       15          1.93810E-01     7.90349E-02
               ____________________________________________
              OUTPUT PARAMETER - SOLUTION TO TRANSPORT EQUATION
```

```
K–EFF                      =     0.996574
N. OUTER ITERATIONS        =            7
TOTAL INNER ITERATIONS     =        60050
TOTAL EXECUTION TIME       =   0:00:01.05    (H/M/S)

******************************************************************************
```

Figure 9: OUTPUT.TXT output file

## References


[Avneet Sood(2003)] D. Kent Parsons Avneet Sood, R.Arthur Forster. Analytical benchmark test set for criticality code verification. *Progress in Nuclear Energy*, 42(1):55 – 106, 2003. ISSN 0149-1970. doi: https://doi.org/10.1016/S0149-1970(02)00098-7. URL `http://www.sciencedirect.com/science/article/pii/S0149197002000987`.