User's Guide
OpenNode: A new software for modelling 3D nuclear reactor cores by solving the multigroup Neutron Diffusion Equation in Cartesian geometry. v.1.1

H. Satti (`hsatti@uae.ac.ma`*)*
Radiations and Nuclear Systems Laboratory, Faculty of Sciences,
Abdelmalek Essaadi University, Tetouan, Morocco

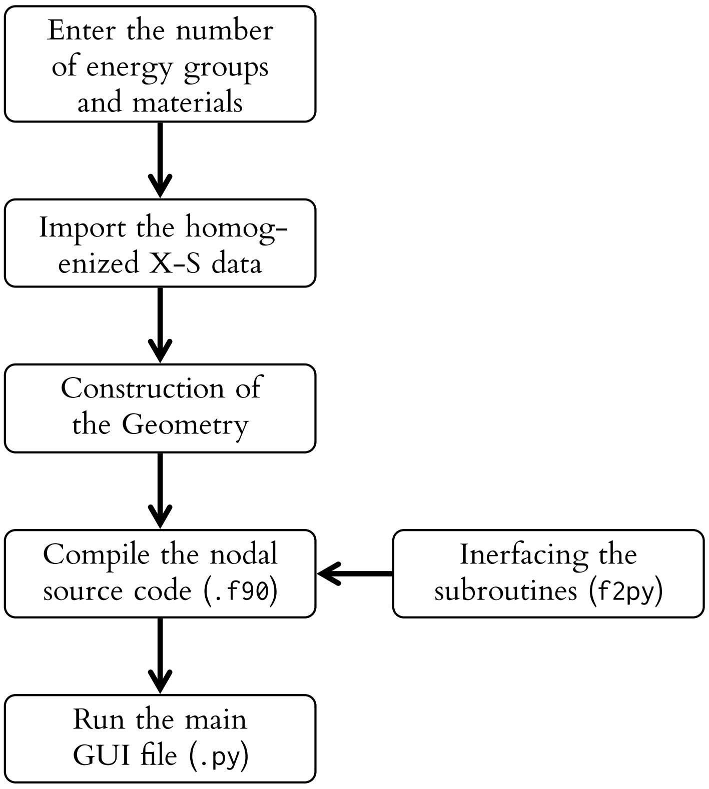
`https://github.com/HichamSatti/OpenNode`
Juin 26, 2022

Introduction

Welcome to the OpenNode User's Guide! This tutorial will describe the essential aspects to perform multigroup neutron diffusion equation physics calculations by using OpenNode code, available for download at `https://github.com/HichamSatti/OpenNode`.

Contents

Quick Guide to Workflow


Enter the number of energy groups and materials
Import the homogenized X-S data
Construction of the Geometry
Compile the nodal source code (.f90)
Inerfacing the subroutines (f2py)
Run the main GUI file (.py)

1 Installation and Quick start

1.1 Obtaining the source

All OpenNode 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/HichamSatti/OpenNode.git
```

1.2 Installing prerequisites on Windows machines

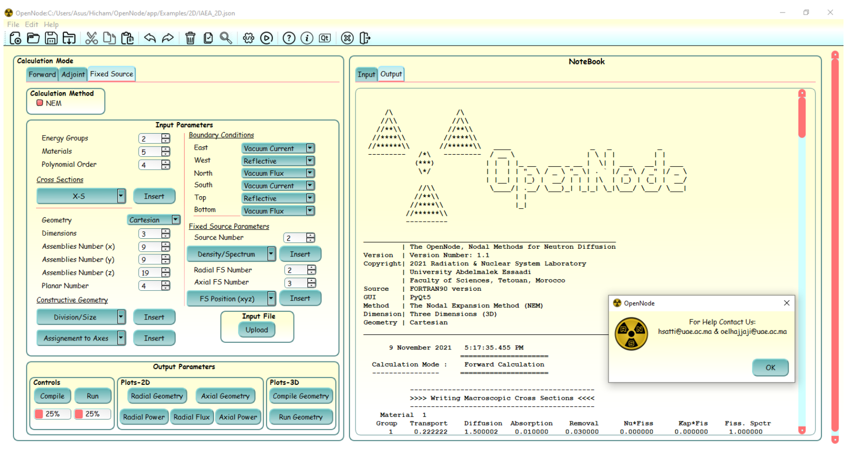
This OpenNode and its GUI has been released under the MIT license. Since, the software is a Python-based application; it requires a Python Runtime Environment to work correctly.

```
        cd  OpenNode
        cd  Script
```

1.3 Running the application

```
python  GUI.py
```

Then, a Main window (GUI) of OpenNode package on Windows machine will be displayed as in Figure 1.



```
          | The OpenNode, Nodal Methods for Neutron Diffusion
Version   | Version Number: 1.1
Copyright| 2021 Radiation & Nuclear System Laboratory
          | University Abdelmalek Essaadi
          | Faculty of Sciences, Tetouan, Morocco
Source    | FORTRAN90 version
GUI       | PyQt5
Method    | The Nodal Expansion Method (NEM)
Dimension| Three Dimensions (3D)
Geometry | Cartesian

    9 November 2021   5:17:35.455 PM
                     =====================
  Calculation Mode :  Forward Calculation
  ----------------   =====================

          ------------------------------------------
          >>>> Writing Macroscopic Cross Sections <<<<
          ------------------------------------------
   Material  1
  Group   Transport   Diffusion  Absorption   Removal    Nu*Fiss     Kap*Fis    Fiss. Spctr
    1      0.222222   1.500002   0.010000   0.030000   0.000000   0.000000    1.000000
```

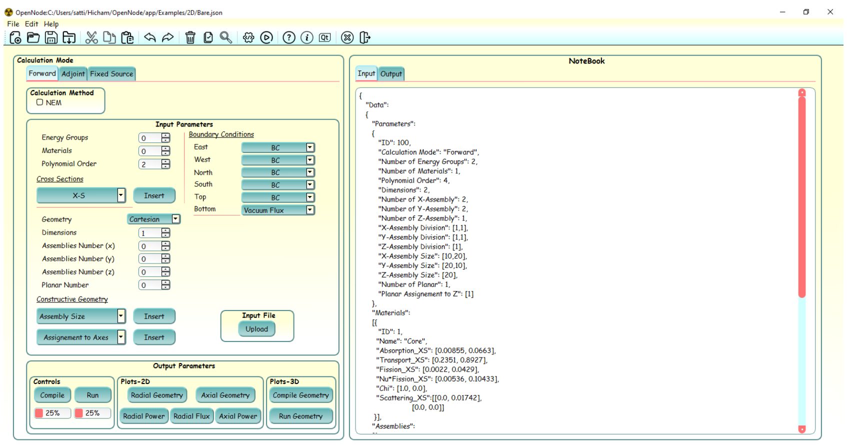
OpenNode
For Help Contact Us:
hsatti@uae.ac.ma & oelhajjaji@uae.ac.ma
OK

Figure 1: Main window of the GUI of the OpenNode code

2 Examples and tests 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 `Input` by opening a new JSON file, from examples file, as is shown in the figure 2. Here, we represent the 2D Homogeneous Bare Core Benchmark described in [1] as an example. This problem contains one material and two energy groups.



```
{
  "Data":
  {
    "Parameters":
    {
      "ID": 100,
      "Calculation Mode": "Forward",
      "Number of Energy Groups": 2,
      "Number of Materials": 1,
      "Polynomial Order": 4,
      "Dimensions": 2,
      "Number of X-Assembly": 2,
      "Number of Y-Assembly": 2,
      "Number of Z-Assembly": 1,
      "X-Assembly Division": [1,1],
      "Y-Assembly Division": [1,1],
      "Z-Assembly Division": [1],
      "X-Assembly Size": [10,20],
      "Y-Assembly Size": [20,10],
      "Z-Assembly Size": [20],
      "Number of Planar": 1,
      "Planar Assignement to Z": [1]
    },
    "Materials":
    [{
      "ID": 1,
     "Name": "Core",
     "Absorption_XS": [0.00855, 0.0663],
     "Transport_XS": [0.2351, 0.8927],
     "Fission_XS": [0.0022, 0.0429],
     "Nu*Fission_XS": [0.00536, 0.10433],
     "Chi": [1.0, 0.0],
     "Scattering_XS":[[0.0, 0.01742],
                      [0.0, 0.0]]
     }],
    "Assemblies":
```

Figure 2: Setting Up Input File for bare geometry

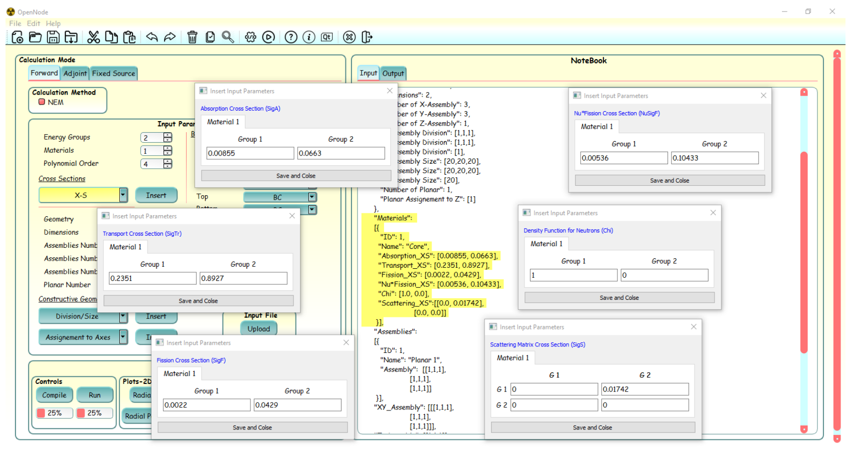
2.2 Generating JSON input files

Another method is to use a set of buttons on the left side of the main window. These buttons allow users to insert input data automatically without requiring an in-depth knowledge of JSON file syntax. Once the user clicks on Upload button the input file will be automatically generated in the window Input. The figure represents the generated input file named input.json contains a 2D Homogeneous Bare Core example taken from [1]. In order for the interface to work properly, we must follow these steps :

1. Choose the calculation mode
2. Choose the calculation method
3. Determination the expansion polynomial order
4. Insertion of the cross section data for each energy groups and materials as described in Figure 3 of :
    - Absorption
    - Transport
    - Fission
    - Nu-Fission
    - Scattering
    - Spectrum energy of neutrons
5. The core geometry is constituting by a number of assemblies distributed along the three axis considering the number of dimensions, these assemblies are characterized by their size. They are subdivided according to the mesh (node) size. Concerning the construction of the geometry we can define :

    A number of radial plans of assemblies, each assembly contains a set of materials which are placed according to their indexes along the number of X-Y assemblies, and an axial distribution of these plans along the Z assemblies Figure 4.
6. Definition of the boundary conditions.

Then finally, by clicking on Upload, two copies of the JSON input file will be automatically created, the first one will be written in the Input window and the second one in a file named "input.json" which is represented in listing 1.



Calculation Method: NEM
Energy Groups: 2
Materials: 1
Polynomial Order: 4
Cross Sections: X-S
Controls: 25% 25%

Absorption Cross Section (SigA) — Material 1

Group 1	Group 2
0.00855	0.0663

Transport Cross Section (SigTr) — Material 1

Group 1	Group 2
0.2351	0.8927

Fission Cross Section (SigF) — Material 1

Group 1	Group 2
0.0022	0.0429

Nu*Fission Cross Section (NuSigF) — Material 1

Group 1	Group 2
0.00536	0.10433

Density Function for Neutrons (Chi) — Material 1

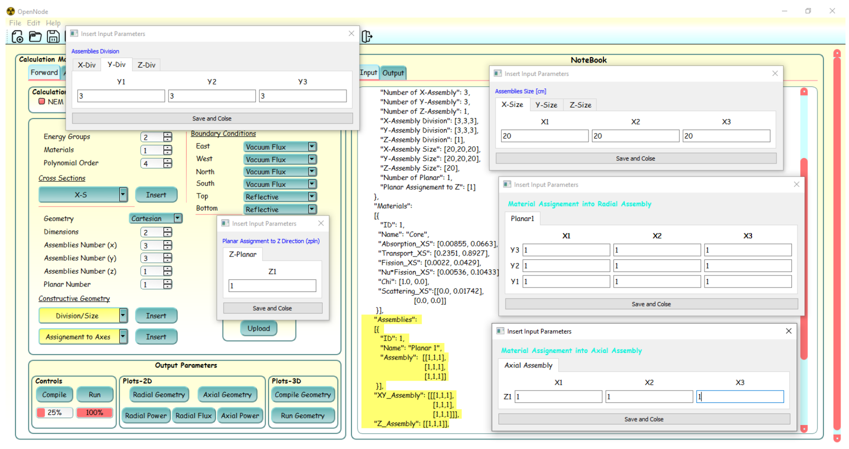
Group 1	Group 2
1	0

Scattering Matrix Cross Section (SigS) — Material 1

	G 1	G 2
G 1	0	0.01742
G 2	0	0

```
nsions": 2,
ber of X-Assembly": 3,
ber of Y-Assembly": 3,
ber of Z-Assembly": 1,
sembly Division": [1,1,1],
sembly Division": [1,1,1],
sembly Division": [1],
sembly Size": [20,20,20],
sembly Size": [20,20,20],
sembly Size": [20],
 "Number of Planar": 1,
 "Planar Assignement to Z": [1]
},
"Materials":
[{
 "ID": 1,
 "Name": "Core",
 "Absorption_XS": [0.00855, 0.0663],
 "Transport_XS": [0.2351, 0.8927],
 "Fission_XS": [0.0022, 0.0429],
 "Nu*Fission_XS": [0.00536, 0.10433],
 "Chi": [1.0, 0.0],
 "Scattering_XS":[[0.0, 0.01742],
             [0.0, 0.0]]
}],
"Assemblies":
[{
 "ID": 1,
 "Name": "Planar 1",
 "Assembly": [[1,1,1],
             [1,1,1],
             [1,1,1]]
}],
"XY_Assembly": [[[1,1,1],
             [1,1,1],
             [1,1,1]]],
```

Figure 3: Setting Up Input File for bare geometry



Calculation Method: NEM
Energy Groups: 2
Materials: 1
Polynomial Order: 4
Cross Sections: X-S
Geometry: Cartesian
Dimensions: 2
Assemblies Number (x): 3
Assemblies Number (y): 3
Assemblies Number (z): 1
Planar Number: 1
Constructive Geometry: Division/Size, Assignement to Axes
Controls: 25% 100%

Boundary Conditions

East	Vacuum Flux
West	Vacuum Flux
North	Vacuum Flux
South	Vacuum Flux
Top	Reflective
Bottom	Reflective

Assemblies Division — X-Div / Y-Div / Z-Div

Y1	Y2	Y3
3	3	3

Assemblies Size [cm] — X-Size / Y-Size / Z-Size

X1	X2	X3
20	20	20

Planar Assignment to Z Direction (zpln) — Z-Planar

Z1
1

Material Assignement into Radial Assembly — Planar1

	X1	X2	X3
Y3	1	1	1
Y2	1	1	1
Y1	1	1	1

Material Assignement into Axial Assembly — Axial Assembly

	X1	X2	X3
Z1	1	1	1

```
 "Number of X-Assembly": 3,
 "Number of Y-Assembly": 3,
 "Number of Z-Assembly": 1,
 "X-Assembly Division": [3,3,3],
 "Y-Assembly Division": [3,3,3],
 "Z-Assembly Division": [1],
 "X-Assembly Size": [20,20,20],
 "Y-Assembly Size": [20,20,20],
 "Z-Assembly Size": [20],
 "Number of Planar": 1,
 "Planar Assignement to Z": [1]
},
"Materials":
[{
 "ID": 1,
 "Name": "Core",
 "Absorption_XS": [0.00855, 0.0663],
 "Transport_XS": [0.2351, 0.8927],
 "Fission_XS": [0.0022, 0.0429],
 "Nu*Fission_XS": [0.00536, 0.10433]
 "Chi": [1.0, 0.0],
 "Scattering_XS":[[0.0, 0.01742],
             [0.0, 0.0]]
}],
"Assemblies":
[{
 "ID": 1,
 "Name": "Planar 1",
 "Assembly": [[1,1,1],
             [1,1,1],
             [1,1,1]]
}],
"XY_Assembly": [[[1,1,1],
             [1,1,1],
             [1,1,1]]],
"Z_Assembly": [[1,1,1]],
```

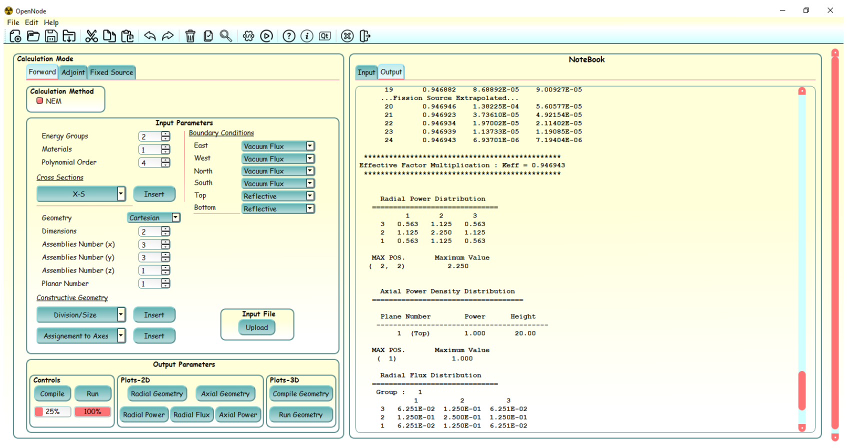
Figure 4: Setting Up Input File for bare geometry

```
{
  "Data":
  {
    "Parameters":
    {
      "ID": 100,
      "Calculation Mode": "Forward",
      "Number of Energy Groups": 2,
      "Number of Materials": 1,
      "Polynomial Order": 4,
      "Dimensions": 2,
      "Number of X-Assembly": 3,
      "Number of Y-Assembly": 3,
      "Number of Z-Assembly": 1,
      "X-Assembly Division": [3,3,3],
      "Y-Assembly Division": [3,3,3],
      "Z-Assembly Division": [1],
      "X-Assembly Size": [20,20,20],
      "Y-Assembly Size": [20,20,20],
      "Z-Assembly Size": [20],
      "Number of Planar": 1,
      "Planar Assignment to Z": [1]
    },
    "Materials":
    [{
      "ID": 1,
     "Name": "Core",
     "Absorption_XS": [0.00855, 0.0663],
     "Transport_XS": [0.2351, 0.8927],
     "Fission_XS": [0.0022, 0.0429],
     "Nu*Fission_XS": [0.00536, 0.10433],
     "Chi": [1.0, 0.0],
          "Scattering_XS":[[0.0, 0.01742],
                           [0.0, 0.0]]
        }],
       "Assemblies":
       [{
          "ID": 1,
          "Name": "Planar 1",
          "Assembly":  [[1,1,1],
                         [1,1,1],
                         [1,1,1]]
        }],
       "XY_Assembly": [[[1,1,1],
                         [1,1,1],
                         [1,1,1]]],
       "Z_Assembly": [[1,1,1]],

       "Boundary Condition":
       {
          "X_East": 0,
          "X_West": 0,
          "Y_North": 0,
          "Y_South": 0,
          "Z_Top": 2,
          "Z_Bottom": 2
       }
    }
}
```

Listing 1: JSON input

3 Running OpenNode

It also allows to compile the FORTRAN source code by clicking on Compile and to execute it on Run creating the final results. Two copies of results will be created, the first one in the Output as shown in Fig.5, the second one is a output file named "NEM.out" which is represented in listing 2. Since we are studying the 2D bare,



Calculation Method: NEM

Energy Groups: 2
Materials: 1
Polynomial Order: 4
Geometry: Cartesian
Dimensions: 2
Assemblies Number (x): 3
Assemblies Number (y): 3
Assemblies Number (z): 1
Planar Number: 1

Boundary Conditions
East: Vacuum Flux
West: Vacuum Flux
North: Vacuum Flux
South: Vacuum Flux
Top: Reflective
Bottom: Reflective

```
      19       0.946882    8.68892E-05    9.00927E-05
    ...Fission Source Extrapolated...
      20       0.946946    1.38225E-04    5.60577E-05
      21       0.946923    3.73610E-05    4.92154E-05
      22       0.946934    1.97002E-05    2.11402E-05
      23       0.946939    1.13733E-05    1.19085E-05
      24       0.946943    6.93701E-06    7.19404E-06

 ***********************************************
Effective Factor Multiplication : Keff = 0.946943
 ***********************************************


    Radial Power Distribution
  ==============================
          1       2       3
    3   0.563   1.125   0.563
    2   1.125   2.250   1.125
    1   0.563   1.125   0.563

  MAX POS.       Maximum Value
 (  2,  2)          2.250


    Axial Power Density Distribution
  ====================================

    Plane Number        Power      Height
   -----------------------------------------
        1  (Top)        1.000       20.00

  MAX POS.       Maximum Value
  (  1)             1.000

    Radial Flux Distribution
  ==============================
   Group :   1
            1           2           3
    3   6.251E-02  1.250E-01  6.251E-02
    2   1.250E-01  2.500E-01  1.250E-01
    1   6.251E-02  1.250E-01  6.251E-02
```

Figure 5: Setting Up Input File for bare geometry

```
   ____________________________________________________________
             | The OpenNode, Nodal Methods for Neutron Diffusion
   Version   | Version Number: 1.1
   Copyright| 2021 Radiation & Nuclear System Laboratory
             | University Abdelmalek Essaadi
             | Faculty of Sciences, Tetouan, Morocco
   Source    | FORTRAN90 version
   GUI       | PyQt5
   Method    | The Nodal Expansion Method (NEM)
   Dimension| Three Dimensions (3D)
```

```
     Geometry | Cartesian
     ----------------------------------------------------------
                          ====================
       Calculation Mode :    Forward Calculation
       ---------------    ====================

                -------------------------------------------
                >>>> Writing Macroscopic Cross Sections <<<<
                -------------------------------------------
         Material  1
        Group    Transport    Diffusion   Absorption    Removal      Nu*Fiss
            Kap*Fis     Fiss. Spctr
           1     0.235100    1.417836    0.008550    0.025970     0.005360
           0.002200     1.000000
           2     0.892700    0.373399    0.066300    0.066300     0.104330
           0.042900     0.000000
             --Scattering Matrix--
               G/G'          1           2
                1         0.000000    0.017420
                2         0.000000    0.000000

                ------------------------------
                >>>>>Wrting Core Geometry<<<<<
                ------------------------------
      Number of Assembly in x, y and z Directions Respectively :
                3           3           1
      Number of Nodes in x, y and z Directions Respectively :
                9           9           1

      x-Directed Nodes Divison (Delta-x)
         6.67   6.67   6.67   6.67   6.67   6.67   6.67   6.67   6.67
      y-Directed Nodes Divison (Delta-y)
         6.67   6.67   6.67   6.67   6.67   6.67   6.67   6.67   6.67
      Planar Region :  1
            1  2  3  4  5  6  7  8  9
         9  1  1  1  1  1  1  1  1  1
         8  1  1  1  1  1  1  1  1  1
         7  1  1  1  1  1  1  1  1  1
         6  1  1  1  1  1  1  1  1  1
         5  1  1  1  1  1  1  1  1  1
         4  1  1  1  1  1  1  1  1  1
         3  1  1  1  1  1  1  1  1  1
         2  1  1  1  1  1  1  1  1  1
         1  1  1  1  1  1  1  1  1  1

      Planar Region Assignment to Planes.
     ------------------------------------
       Plane Number      Planar Region     Delta-z
            1 (Top)              1           20.00

       Boundary Conditions
       ------------------
      X-Directed West   : Zero Flux
      X-Directed East   : Zero Flux
      Y-Directed North  : Zero Flux
      Y-Directed South  : Zero Flux
      Z-Directed Bottom : Reflective
      Z-Directed Top    : Reflective

      ...Core Geometry is Successfully Read...
         --------------------------------



      ===============================================================
                           Calculation Results
      ===============================================================

       Iteration     Keff      FissSrc Error   Inner Error
      ---------------------------------------------------
           1       1.332562    1.05089E+00    2.68501E+00
           2       1.027334    1.44329E+00    1.69003E+00
           3       0.918491    6.46112E-01    7.12272E-01
           4       0.896763    2.71554E-01    2.89577E-01
           5       0.907867    4.72604E-01    1.39041E-01
           6       0.914031    1.23679E-01    1.57351E-01
           7       0.925774    3.18158E-02    3.40331E-02
           8       0.933064    1.94921E-02    2.02242E-02
           9       0.937788    1.26673E-02    1.30130E-02
          10       0.946904    2.40373E-02    8.53224E-03
          11       0.943125    7.71905E-03    1.06859E-02
          12       0.944741    3.09270E-03    3.35977E-03
          13       0.945623    1.76439E-03    1.85306E-03
          14       0.946133    1.07886E-03    1.11877E-03
          15       0.946918    1.73439E-03    6.98090E-04
          16       0.946634    4.72574E-04    6.29379E-04
          17       0.946767    2.46302E-04    2.64468E-04
          18       0.946840    1.42271E-04    1.48966E-04
          19       0.946882    8.68892E-05    9.00927E-05
          20       0.946946    1.38225E-04    5.60577E-05
          21       0.946923    3.73610E-05    4.92154E-05
          22       0.946934    1.97002E-05    2.11402E-05
          23       0.946939    1.13733E-05    1.19085E-05
          24       0.946943    6.93701E-06    7.19404E-06
```

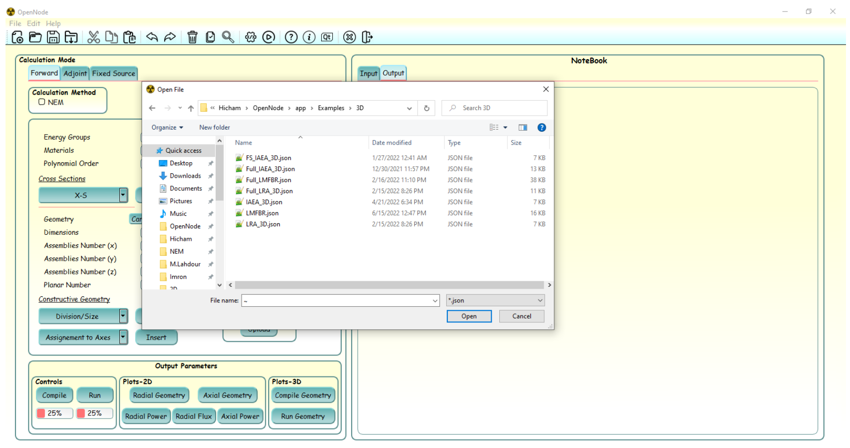
```

  ***********************************************
 Effective Factor Multiplication : Keff = 0.946943
  ***********************************************


      Radial Power Distribution
    ==============================
              1        2        3
      3    0.563    1.125    0.563
      2    1.125    2.250    1.125
      1    0.563    1.125    0.563

   MAX POS.          Maximum Value
  (  2,  2)              2.250


      Axial Power Density Distribution
    ====================================

      Plane Number           Power        Height
     -----------------------------------------

           1  (Top)          1.000        20.00

   MAX POS.          Maximum Value
    (  1)                1.000

      Radial Flux Distribution
    ==============================
     Group :    1
                  1           2           3
      3    6.251E-02   1.250E-01   6.251E-02
      2    1.250E-01   2.500E-01   1.250E-01
      1    6.251E-02   1.250E-01   6.251E-02

     Group :    2
                  1           2           3
      3    6.251E-02   1.250E-01   6.251E-02
      2    1.250E-01   2.500E-01   1.250E-01
      1    6.251E-02   1.250E-01   6.251E-02


  Total Time :     6.25000000E-02  Seconds

```

Listing 2: Output file

3.1 Running OpenNode under a GUI

Two options may be set at the top of the main AutoratingCalculator window. Once you have created your input file (*.json), it is relatively straight-forward to run the code. The steps of running the code are:

- Set up the input file as described in Sec.2.1 and Sec.2.2 or import one of the existing input files into ./app/Examples from the menu File › Open or press Ctrl + O. You will find in this folder a set of test cases named (*.json) (see Fig.6)
- Select the Forward or Adjoint mode in Calculation Mode.
- Select the NEM method in Calculation Method.
- Click on Compile or press Ctrl + L, a Python C/API extension modules to call Fortran 90/95 modules subroutines will be created. This task can be done only at the first time you use the software.
- Check Boundary Condition
- Click on Run or press Ctrl + R to execute the problem. The final results as displayed on the software shown in Fig.5



Name	Date modified	Type	Size
FS_IAEA_3D.json	1/27/2022 12:41 AM	JSON file	7 KB
Full_IAEA_3D.json	12/30/2021 11:57 PM	JSON file	13 KB
Full_LMFBR.json	2/16/2022 11:10 PM	JSON file	38 KB
Full_LRA_3D.json	2/15/2022 8:26 PM	JSON file	11 KB
IAEA_3D.json	4/21/2022 6:34 PM	JSON file	7 KB
LMFBR.json	6/15/2022 12:47 PM	JSON file	16 KB
LRA_3D.json	2/15/2022 8:26 PM	JSON file	7 KB

Figure 6: Import input file

3.2 Power visualisation

The Radial Power gives the radial distribution of neutron power, Fig.7 shows the power map obtained,

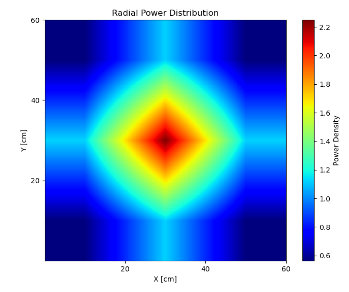
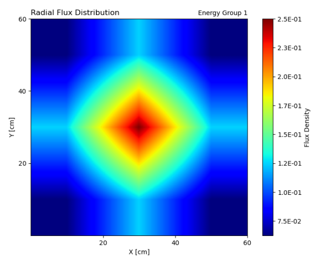


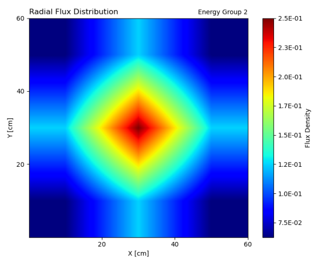
Figure 7: Radial power distribution

3.3 Flux visualisation

The Radial Flux allows to plot the nodal flux distribution in each energy group; thermal Fig.8a and fast group Fig.8b.



(a) Radial flux distribution for thermal group



(b) Radial flux distribution for fast group

Figure 8: Radial flux and power distribution for the 2-D bare

3.4 2D Geometry visualisation

To plot the studied geometry, we will limit ourselves only to the radial trace. The Radial Geometry allows to do this, the geometry is illustrated in Fig.9.

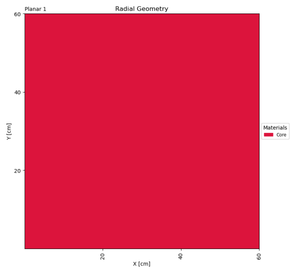


Figure 9: 2-D Geometry of the bare

3.5 3D Geometry visualisation

The Compile Geometry permits to plot figures in 3D graphics, it prepares a blender file called with the name of studied benchmark, in this case "Bare.blend", and will be executed with the Run Geometry as shown in Fig.10.

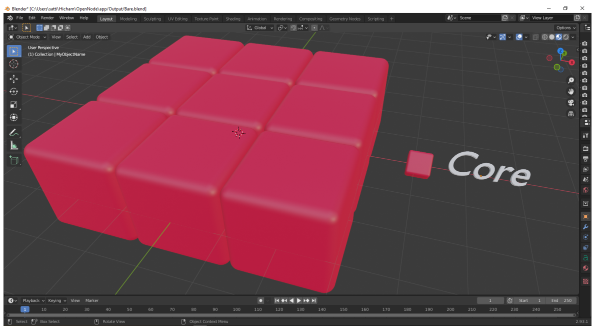


Figure 10: 3-D Geometry of the bare

4 Keyboard Shortcuts list

- New File : Ctrl + N .
- Open File : Ctrl + O .
- Save File : Ctrl + S .
- Save as : Ctrl + Alt + S .
- Cut : Ctrl + X .
- Copy : Ctrl + C .
- Paste : Ctrl + V .

- Undo : Ctrl + Z .
- Redo : Ctrl + Y .
- delete : Del .
- Select All : Ctrl + A .
- Find : Ctrl + F .
- Compile : Ctrl + L .
- Run : Ctrl + R .
- Help : F2 .
- About : F3 .
- About Qt : F4 .
- Quit : Ctrl + Q .

References

[1] M. L. Zerkle, *Development of a polynomial nodal method with flux and current discontinuity factors*. PhD thesis, Massachusetts Institute of Technology, 1992.