# LA-UR-23-30353

**Approved for public release; distribution is unlimited.**

| | |
|---|---|
| **Title:** | easy_pert: A Python tool for using the PERT card to compute fixed source sensitivities to nuclear data |
| **Author(s):** | Clark, Alexander Rich |
| **Intended for:** | 2023 MCNP User Symposium, 2023-09-18/2023-09-21 (Los Alamos, New Mexico, United States) |
| **Issued:** | 2023-10-09 (rev.1) |

# easy_pert: A Python tool for using the PERT card to compute fixed source sensitivities to nuclear data

Alexander R. Clark
2023 MCNP User Symposium

Monday, September 18th, 2023
LA-UR-23-30353

# Overview

- Motivation
- Command-line interface examples
- Computing sensitivities with the PERT card
- Writing a PERT card
- easy_pert examples
- Demonstration of sensitivity analysis
- Conclusions
- Future work and outlook

# Making sensitivity analysis via the PERT card accessible

- The PERT card is an efficient way to compute first- and second-order fixed-source sensitivities to nuclear data
  - Addition of fictitious material means no change to original material specifications
  - Modest (10%-20%) increase in run time with PERT cards for as many tallies, cells, reactions, energy bins, and methods as desired
- Several aspects of using the PERT card can be tedious and error-prone
  - Defining the fictitious material
    - Perturbed nuclide atom/weight fraction
    - Perturbed density
  - Writing the PERT card(s) to an input deck
  - Parsing the unperturbed and perturbed tally results
  - Combining the unperturbed and perturbed tally results to calculate sensitivities and their statistical uncertainties
- easy_pert abstracts all these steps behind a command-line interface
  - Does the heavy lifting for the various steps
  - Requires relatively few command-line inputs from the user
  - Provides results in a JSON format

# Command-line interface examples

```
● arclark@[redacted]:~/git_repos/easy_pert% python -m easy_pert -h
usage: easy_pert.py [-h] {write,parse,combine,sensitivity,plot} ...

================================================================================
"Easy PERT" is a Python tool designed to make the MCNP code PERT card more accessible. The typical workflow is:

        1. Writing all required PERT card entries to an existing input deck ("write")
        2. performing the MCNP code calculation
        3. parsing the output MCTAL file ("parse")

Additional utility functions, such as:

        * Combining JSON files ("combine")
        * Computing sensitivities ("sensitivity")
        * Plotting sensitivities ("plot")

are also available.

For more information regarding each command, type "python -m easy_pert <command> -h."

"Easy PERT" computes sensitivities using the procedure described in:
J. A. Favorite, "Using the MCNP Taylor series perturbation feature (efficiently) for shielding problems",
EPJ Web of Conferences 153, 06030 (2017), ICRS-13 & RPSD-2016, DOI: 10.1051/epjconf/20171530
================================================================================

options:
  -h, --help            show this help message and exit

commands:
  {write,parse,combine,sensitivity,plot}
    write               Tool for writing PERT cards to an existing MCNP code input deck.
    parse               Tool for parsing PERT card results from MCNP code MCTAL files.
    combine             Tool for combining JSON files.
    sensitivity         Computes sensitivities via the MCNP code PERT card method.
    plot                Plots sensitivities computed via the MCNP code PERT card method as a function of pbin.
```

# Command-line interface examples

```
arclark@[redacted]:~/projects/easy_pert_runs/simple_berp% python -m easy_pert write -h
usage: easy_pert.py write [-h] -e E -ccn CCN [CCN ...] -zaid ZAID -rxn RXN [-units {eV,keV,MeV}] [-fmcn FMCN] [-p P]
                          [-methods {0,1,-1,2,-2,3,-3} [{0,1,-1,2,-2,3,-3} ...]] -i I -mcn MCN

Tool for writing PERT cards to an existing MCNP code input deck.

options:
  -h, --help            show this help message and exit
  -units {eV,keV,MeV}   Group structure units.
  -fmcn FMCN            Material card number for the fictitious material defining the PERT card perturbation. Must not match an
                        existing material card number.
  -p P                  Relative perturbation size. The default size is recommended but another value can be chosen at the user's
                        discretion.
  -methods {0,1,-1,2,-2,3,-3} [{0,1,-1,2,-2,3,-3} ...]
                        Desired PERT card method(s). For example, "-m 1 -1" will perform the first- and second-order perturbations
                        and return the difference in the unperturbed tally and the perturbed tally, respectively, in the MCTAL
                        file. Please see the MCNP manual for more detail. Choosing "0" for the `parse` option will result in only
                        parsing the unperturbed tally.

Required arguments:
  -e E                  Group structure path.
  -ccn CCN [CCN ...]    Cell card number(s) of the cell(s) for which the PERT cards apply. All cells must contain the same
                        material and have the same density.
  -zaid ZAID            Complete ZAID for which the PERT cards apply (e.g. "94239.00c").
  -rxn RXN              MT number for which the PERT cards apply.
  -i I                  MCNP input deck path.
  -mcn MCN              Material card number of the material for which the PERT cards apply.
```

# Computing sensitivities with the PERT card

- PERT card method estimates the change in a tally due to a perturbation in the nuclear data via the differential operator method
- Jeffrey A. Favorite demonstrated how sensitivities can be computed
  - Expand a tally, $c(\sigma_x)$, that depends on some reaction cross section, $\sigma_x$, in a Taylor series

$$c(\sigma_x) = c(\sigma_{x,0}) + \left.\frac{dc}{d\sigma_x}\right|_{\sigma_{x,0}} \Delta\sigma_x + \frac{1}{2}\left.\frac{d^2c}{d\sigma_x^2}\right|_{\sigma_{x,0}} (\Delta\sigma_x)^2 + \cdots$$

$c(\sigma_{x,0})$ is the tally calculated with unperturbed nuclear data, $\sigma_{x,0}$
$\Delta\sigma_x \equiv \sigma_x - \sigma_{x,0}$

  - Use the chain rule, $p_x \equiv \frac{\Delta\sigma_x}{\sigma_{x,0}}$, and relative sensitivity definition to express the first-order sensitivity

$$\Delta c_1 \equiv \left.\frac{dc}{d\sigma_x}\right|_{\sigma_{x,0}} \Delta\sigma_x = \left.\frac{dc}{dp_x}\right|_{p_x=0} p_x \rightarrow S_{c,\sigma_x} \equiv \frac{\sigma_{x,0}}{c(\sigma_{x,0})}\left.\frac{dc}{d\sigma_x}\right|_{\sigma_{x,0}} = \frac{\Delta c_1}{p_{x,r}c(\sigma_{x,0})}$$

  - Use linear propagation of uncertainties to compute the sensitivity statistical uncertainty (uncorrelated)

$$\sigma^2_{S_{c,\sigma_x}} = S^2_{c,\sigma_x}\left[\left(\frac{\sigma_{c(\sigma_{x,0})}}{c(\sigma_{x,0})}\right)^2 + \left(\frac{\sigma_{c_1}}{c_1}\right)^2\right]$$



Jeffrey A. Favorite, *"Using the MCNP Taylor series perturbation feature (efficiently) for shielding problems"*, *EPJ Web of Conferences 153, 06030 (2017)*

# Writing a PERT card

- Define the fictitious material by multiplying the nuclide amount by $1 + p_{x,r}$

  m100 1001 2 8016 1

  m9999 1001 4 8016 1

- Define a fictitious material density, $\rho'$, to counteract the renormalization of the fictitious material atom/weight fractions

  $$\rho' = \frac{\sum_i f_i'}{\sum_i f_{i,0}} \rho_0$$

  $\sum_i f_i'$ and $\sum_i f_{i,0}$ are the sum of the fictitious and original material atom/weight fractions, respectively

  $\rho_0$ is the unperturbed material density

- For cell 1 filled with material 100 at density 1 g/cc, the PERT card is written as "pert1:n cel=1 mat=9999 … den= 5/3*(-1.0) …"

# easy_pert examples – write

```
simple BeRP ball model to demonstrate using the PERT card.
c ==============================================================================
c Cell cards.
c ==============================================================================
1    100   -19.    -10    imp:n=1
1000 0             +10    imp:n=0

c ==============================================================================
c Surface cards.
c ==============================================================================
10  so  3.8

c ==============================================================================
c Data cards.
c ==============================================================================
mode n
nps 1e+06
sdef
      pos=0. 0. 0.
      rad=d1
      erg=d2
si1 0. 3.8
sp1 -21 2
sp2 -3 0.794930 4.689270 $ Pu-240 Watt fission parameters.
prdmp 2j 1
f101:n   10
fc101 Neutron current tally across surface 10.
e101  1e-11 11log 10. 15. 20.
m100  94239.00c   -0.94
      94240.00c   -0.06
rand gen=2 seed=19073486328125
print
```


```
rand gen=2 seed=19073486328125
print
```

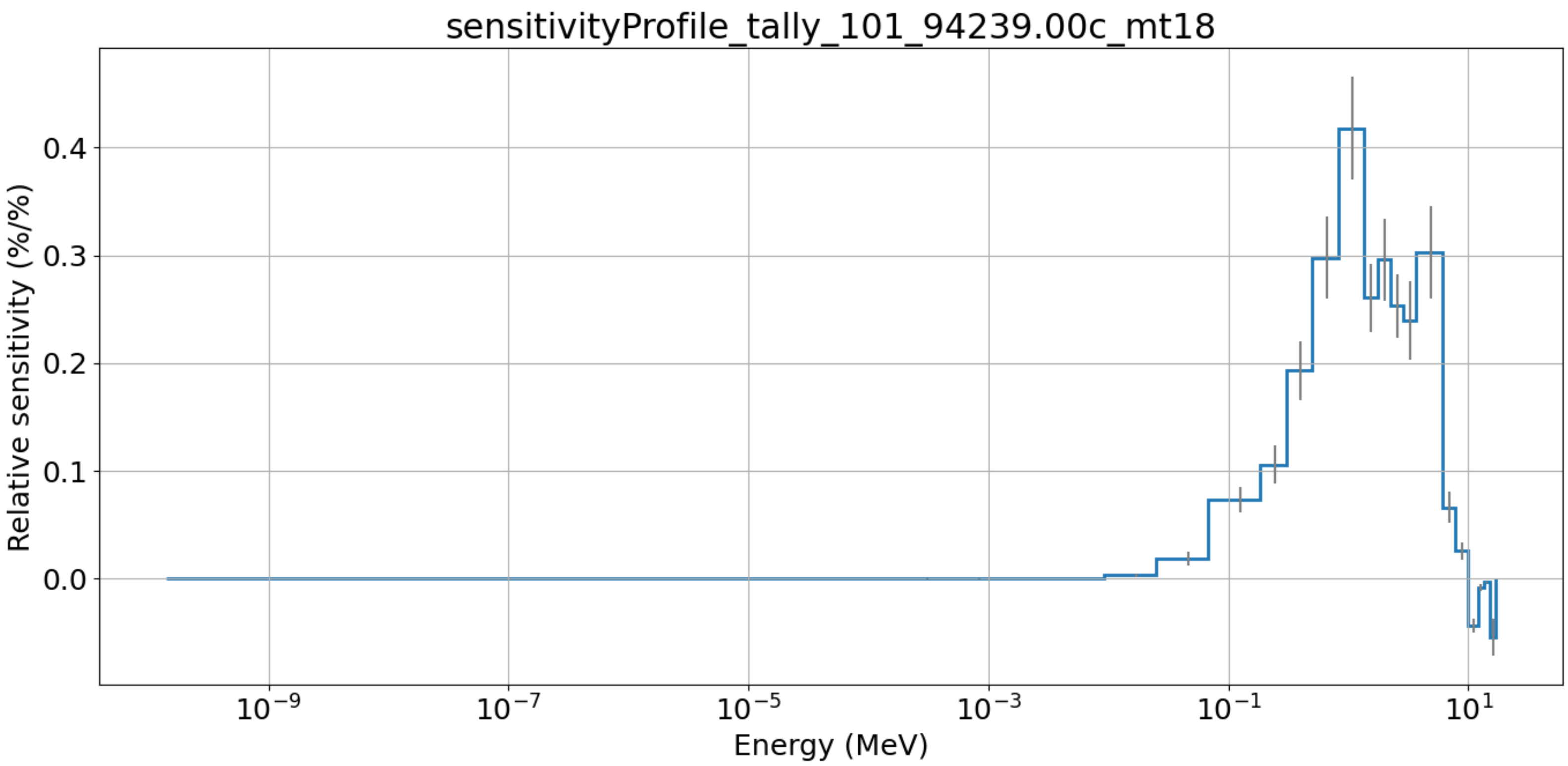

```
c fictitious material for use with the PERT card
c using p=1.0, zaid=94239.00c
m9999
      94239.00c -1.88000000e+00
      94240.00c -6.00000000e-02
pert1:n cell=1 mat=9999 rho=-3.68600000e+01 rxn=18 erg=1.3900e-10, 1.5200e-07 method=2
pert2:n cell=1 mat=9999 rho=-3.68600000e+01 rxn=18 erg=1.5200e-07, 4.1400e-07 method=2
pert3:n cell=1 mat=9999 rho=-3.68600000e+01 rxn=18 erg=4.1400e-07, 1.1300e-06 method=2
pert4:n cell=1 mat=9999 rho=-3.68600000e+01 rxn=18 erg=1.1300e-06, 3.0600e-06 method=2
pert5:n cell=1 mat=9999 rho=-3.68600000e+01 rxn=18 erg=3.0600e-06, 8.3200e-06 method=2
pert6:n cell=1 mat=9999 rho=-3.68600000e+01 rxn=18 erg=8.3200e-06, 2.2600e-05 method=2
pert7:n cell=1 mat=9999 rho=-3.68600000e+01 rxn=18 erg=2.2600e-05, 6.1400e-05 method=2
pert8:n cell=1 mat=9999 rho=-3.68600000e+01 rxn=18 erg=6.1400e-05, 1.6700e-04 method=2
pert9:n cell=1 mat=9999 rho=-3.68600000e+01 rxn=18 erg=1.6700e-04, 4.5400e-04 method=2
pert10:n cell=1 mat=9999 rho=-3.68600000e+01 rxn=18 erg=4.5400e-04, 1.2350e-03 method=2
pert11:n cell=1 mat=9999 rho=-3.68600000e+01 rxn=18 erg=1.2350e-03, 3.3500e-03 method=2
pert12:n cell=1 mat=9999 rho=-3.68600000e+01 rxn=18 erg=3.3500e-03, 9.1200e-03 method=2
pert13:n cell=1 mat=9999 rho=-3.68600000e+01 rxn=18 erg=9.1200e-03, 2.4800e-02 method=2
pert14:n cell=1 mat=9999 rho=-3.68600000e+01 rxn=18 erg=2.4800e-02, 6.7600e-02 method=2
pert15:n cell=1 mat=9999 rho=-3.68600000e+01 rxn=18 erg=6.7600e-02, 1.8400e-01 method=2
pert16:n cell=1 mat=9999 rho=-3.68600000e+01 rxn=18 erg=1.8400e-01, 3.0300e-01 method=2
pert17:n cell=1 mat=9999 rho=-3.68600000e+01 rxn=18 erg=3.0300e-01, 5.0000e-01 method=2
pert18:n cell=1 mat=9999 rho=-3.68600000e+01 rxn=18 erg=5.0000e-01, 8.2300e-01 method=2
pert19:n cell=1 mat=9999 rho=-3.68600000e+01 rxn=18 erg=8.2300e-01, 1.3530e+00 method=2
pert20:n cell=1 mat=9999 rho=-3.68600000e+01 rxn=18 erg=1.3530e+00, 1.7380e+00 method=2
pert21:n cell=1 mat=9999 rho=-3.68600000e+01 rxn=18 erg=1.7380e+00, 2.2320e+00 method=2
pert22:n cell=1 mat=9999 rho=-3.68600000e+01 rxn=18 erg=2.2320e+00, 2.8650e+00 method=2
pert23:n cell=1 mat=9999 rho=-3.68600000e+01 rxn=18 erg=2.8650e+00, 3.6800e+00 method=2
pert24:n cell=1 mat=9999 rho=-3.68600000e+01 rxn=18 erg=3.6800e+00, 6.0700e+00 method=2
pert25:n cell=1 mat=9999 rho=-3.68600000e+01 rxn=18 erg=6.0700e+00, 7.7900e+00 method=2
pert26:n cell=1 mat=9999 rho=-3.68600000e+01 rxn=18 erg=7.7900e+00, 1.0000e+01 method=2
pert27:n cell=1 mat=9999 rho=-3.68600000e+01 rxn=18 erg=1.0000e+01, 1.2000e+01 method=2
pert28:n cell=1 mat=9999 rho=-3.68600000e+01 rxn=18 erg=1.2000e+01, 1.3500e+01 method=2
pert29:n cell=1 mat=9999 rho=-3.68600000e+01 rxn=18 erg=1.3500e+01, 1.5000e+01 method=2
pert30:n cell=1 mat=9999 rho=-3.68600000e+01 rxn=18 erg=1.5000e+01, 1.7000e+01 method=2
```

# easy_pert examples – parse

```
[
  {
    "data": {
      "values": [⋯
      ],
      "uncertainties": [⋯
      ],
      "structure": [⋯
      ],
      "units": {⋯
      }
    },
    "attributes": {
      "comments": "        Neutron current tally across surface 10.
      "tally_number": "101",
      "perturbed": "False"
    }
  },
  {
    "data": {
      "values": [⋯
      ],
      "uncertainties": [⋯
      ],
      "structure": [⋯
      ],
      "units": {⋯
      }
    },
    "attributes": {
      "comments": "        Neutron current tally across surface 10.
      "tally_number": "101",
      "perturbed": "True",
      "cell": "[1]",
      "mat": "9999",
      "rxn": "18",
      "method": "[2]",
      "relative_perturbation": "1.0",
      "zaid": "94239.00c"
    }
  }
]
```

Los Alamos
NATIONAL LABORATORY

# easy_pert examples – plot




Los Alamos NATIONAL LABORATORY

# Sensitivity analysis demonstration

# Conclusions

- easy_pert provides a command line interface to simplify the various steps in writing and parsing PERT cards
  - Abstracts various tasks away from the user
  - Provides outputs in JSON format
- Implements Jeffrey A. Favorite’s method for performing sensitivity analysis and computing the sensitivity statistical uncertainties

# Future work and outlook

- Implement a Python API in addition to the command-line interface
- Allow second-order sensitivity analysis and exact uncertainty calculation
- Add capability to write RAND card seeds to generate uncorrelated PERT card results