LA-UR-17-30460
Approved for public release; distribution is unlimited.

Title:	FLIP for FLAG model visualization
Author(s):	Wooten, Hasani Omar
Intended for:	Report
Issued:	2017-11-15

FLIP for FLAG model visualization

By

H. Omar Wooten
X Theoretical Design-Primary Physics (XTD-PRI)

Abstract

A graphical user interface has been developed for FLAG users. FLIP (FLAG Input deck Parser) provides users with an organized view of FLAG models and a means for efficiently and easily navigating and editing nodes, parameters, and variables.

Introduction

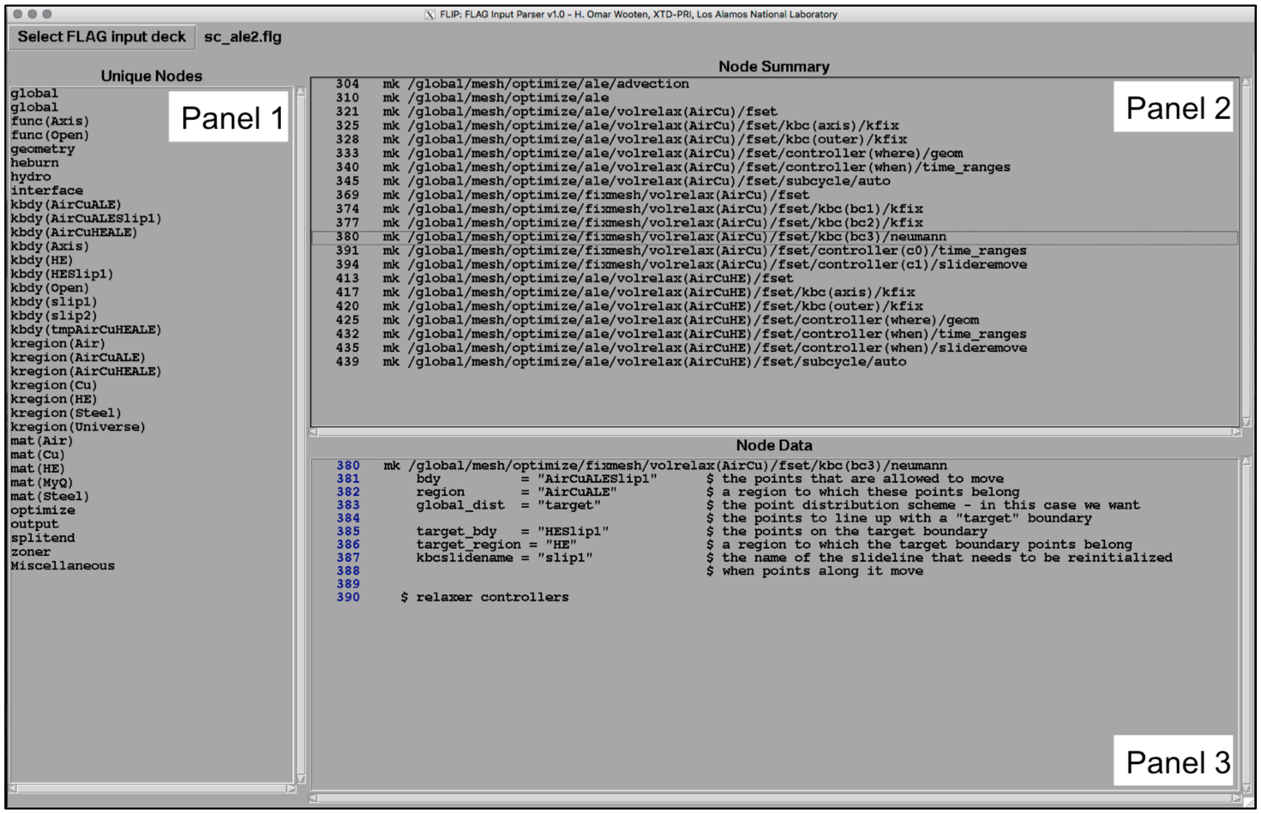
The FLAG hydrodynamics code[1] performs calculations based on models that are described in a hierarchical database-structured input deck. The code offers users some methods to make input decks more readable such as linking to separate files that include a set of parameters ("include" files), and statements to relieve the user from repeatedly typing increasingly long node locations ("mk +" and "cd" statements). Nonetheless, the free-form nature of a FLAG model's input deck can lead to models that are difficult to decipher and debug. For example, unlike some codes that require materials be described together and within the same section of the input deck, FLAG allows the user to describe materials anywhere within the input deck. Furthermore the length of the input deck scales with the complexity of the model, and as sections of one model are re-used in others, keeping track of the entire model can be challenging.

A new software application has been developed to provide FLAG users with a visual environment for more efficiently managing models. FLIP (FLAG Input deck Parser) is a graphical user interface that displays the model database in a well-organized and easy-to-navigate environment. Figure 1 shows FLIP's presentation of the Shape Charge Example available in FLAG's online documentation.

Methods and Materials

FLIP parses the model database hierarchy, imports the contents of "include" files, extracts each node and its associated sub-nodes, variables, parameters, and presents an organized view of the master model across three interactive panels.

Panel 1 displays an alphabetized list of the 1st, 2nd, 3rd, and 4th level nodes of the model database. This list is convenient for verifying, for example, that the expected numbers of boundaries ("kbdy"), regions ("kregions"), and materials ("mats") and their names, are included in the model as expected.



```
global
global
func(Axis)
func(Open)
geometry
heburn
hydro
interface
kbdy(AirCuALE)
kbdy(AirCuALESlip1)
kbdy(AirCuHEALE)
kbdy(Axis)
kbdy(HE)
kbdy(HESlip1)
kbdy(Open)
kbdy(slip1)
kbdy(slip2)
kbdy(tmpAirCuHEALE)
kregion(Air)
kregion(AirCuALE)
kregion(AirCuHEALE)
kregion(Cu)
kregion(HE)
kregion(Steel)
kregion(Universe)
mat(Air)
mat(Cu)
mat(HE)
mat(MyQ)
mat(Steel)
optimize
output
splitend
zoner
Miscellaneous
```

```
304   mk /global/mesh/optimize/ale/advection
310   mk /global/mesh/optimize/ale
321   mk /global/mesh/optimize/ale/volrelax(AirCu)/fset
325   mk /global/mesh/optimize/ale/volrelax(AirCu)/fset/kbc(axis)/kfix
328   mk /global/mesh/optimize/ale/volrelax(AirCu)/fset/kbc(outer)/kfix
333   mk /global/mesh/optimize/ale/volrelax(AirCu)/fset/controller(where)/geom
340   mk /global/mesh/optimize/ale/volrelax(AirCu)/fset/controller(when)/time_ranges
345   mk /global/mesh/optimize/ale/volrelax(AirCu)/fset/subcycle/auto
369   mk /global/mesh/optimize/fixmesh/volrelax(AirCu)/fset
374   mk /global/mesh/optimize/fixmesh/volrelax(AirCu)/fset/kbc(bc1)/kfix
377   mk /global/mesh/optimize/fixmesh/volrelax(AirCu)/fset/kbc(bc2)/kfix
380   mk /global/mesh/optimize/fixmesh/volrelax(AirCu)/fset/kbc(bc3)/neumann
391   mk /global/mesh/optimize/fixmesh/volrelax(AirCu)/fset/controller(c0)/time_ranges
394   mk /global/mesh/optimize/fixmesh/volrelax(AirCu)/fset/controller(c1)/slideremove
413   mk /global/mesh/optimize/ale/volrelax(AirCuHE)/fset
417   mk /global/mesh/optimize/ale/volrelax(AirCuHE)/fset/kbc(axis)/kfix
420   mk /global/mesh/optimize/ale/volrelax(AirCuHE)/fset/kbc(outer)/kfix
425   mk /global/mesh/optimize/ale/volrelax(AirCuHE)/fset/controller(where)/geom
432   mk /global/mesh/optimize/ale/volrelax(AirCuHE)/fset/controller(when)/time_ranges
435   mk /global/mesh/optimize/ale/volrelax(AirCuHE)/fset/controller(when)/slideremove
439   mk /global/mesh/optimize/ale/volrelax(AirCuHE)/fset/subcycle/auto
```

```
380   mk /global/mesh/optimize/fixmesh/volrelax(AirCu)/fset/kbc(bc3)/neumann
381       bdy          = "AirCuALESlip1"      $ the points that are allowed to move
382       region       = "AirCuALE"           $ a region to which these points belong
383       global_dist  = "target"             $ the point distribution scheme - in this case we want
384                                           $ the points to line up with a "target" boundary
385       target_bdy   = "HESlip1"            $ the points on the target boundary
386       target_region = "HE"                $ a region to which the target boundary points belong
387       kbcslidename = "slip1"              $ the name of the slideline that needs to be reinitialized
388                                           $ when points along it move
389
390     $ relaxer controllers
```

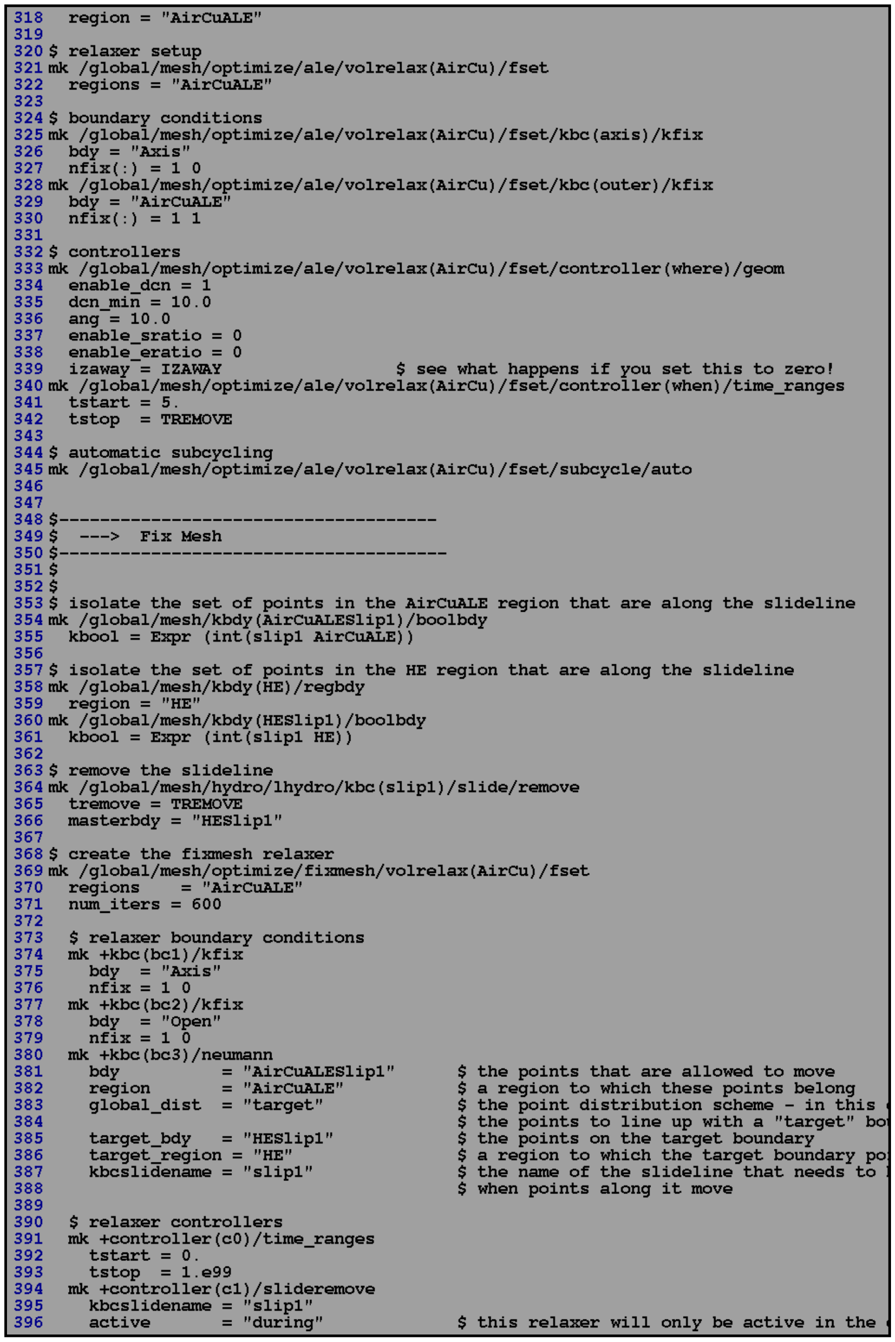
Figure 1. FLIP's user interface showing the Shaped Charge Example FLAG model.

Clicking on a node in Panel 1 produces an organized list in Panel 2 of all (if any) sub-nodes under the selected node. Clicking any sub-node in Panel 2 displays parameters and variables associated with the selected sub-node in Panel 3. For user convenience, the line numbers are displayed in Panels 2 and 3, indicating where those nodes and parameters are located in the master model input deck.

Thus, in Figure 1, the "optimize" 3rd level node has been selected. Its sub-nodes are shown in the Panel 2, and the "Neumann" boundary condition for the "fixmesh" volume relaxer is selected, with its associated parameters presented in Panel 3.

The Shape Charge Example shown in Figure 1 is 467 lines long. To demonstrate the efficiencies offered by FLIP in visualizing a model, the input deck lines containing the same nodes and data displayed in Figure 1 are shown in Figure 2.

FLIP was developed in the scripting language Python Anaconda[2] (ver 2.7.11), using the Tkinter GUI module.



```
 region = "AirCuALE"

$ relaxer setup
mk /global/mesh/optimize/ale/volrelax(AirCu)/fset
  regions = "AirCuALE"

$ boundary conditions
mk /global/mesh/optimize/ale/volrelax(AirCu)/fset/kbc(axis)/kfix
  bdy = "Axis"
  nfix(:) = 1 0
mk /global/mesh/optimize/ale/volrelax(AirCu)/fset/kbc(outer)/kfix
  bdy = "AirCuALE"
  nfix(:) = 1 1

$ controllers
mk /global/mesh/optimize/ale/volrelax(AirCu)/fset/controller(where)/geom
  enable_dcn = 1
  dcn_min = 10.0
  ang = 10.0
  enable_sratio = 0
  enable_eratio = 0
  izaway = IZAWAY                  $ see what happens if you set this to zero!
mk /global/mesh/optimize/ale/volrelax(AirCu)/fset/controller(when)/time_ranges
  tstart = 5.
  tstop  = TREMOVE

$ automatic subcycling
mk /global/mesh/optimize/ale/volrelax(AirCu)/fset/subcycle/auto


$-------------------------------------
$  --->  Fix Mesh
$--------------------------------------
$
$
$ isolate the set of points in the AirCuALE region that are along the slideline
mk /global/mesh/kbdy(AirCuALESlip1)/boolbdy
  kbool = Expr (int(slip1 AirCuALE))

$ isolate the set of points in the HE region that are along the slideline
mk /global/mesh/kbdy(HE)/regbdy
  region = "HE"
mk /global/mesh/kbdy(HESlip1)/boolbdy
  kbool = Expr (int(slip1 HE))

$ remove the slideline
mk /global/mesh/hydro/lhydro/kbc(slip1)/slide/remove
  tremove = TREMOVE
  masterbdy = "HESlip1"

$ create the fixmesh relaxer
mk /global/mesh/optimize/fixmesh/volrelax(AirCu)/fset
  regions    = "AirCuALE"
  num_iters = 600

  $ relaxer boundary conditions
  mk +kbc(bc1)/kfix
    bdy  = "Axis"
    nfix = 1 0
  mk +kbc(bc2)/kfix
    bdy  = "Open"
    nfix = 1 0
  mk +kbc(bc3)/neumann
    bdy           = "AirCuALESlip1"      $ the points that are allowed to move
    region        = "AirCuALE"           $ a region to which these points belong
    global_dist   = "target"             $ the point distribution scheme - in this
                                         $ the points to line up with a "target" bo
    target_bdy    = "HESlip1"            $ the points on the target boundary
    target_region = "HE"                 $ a region to which the target boundary po
    kbcslidename = "slip1"               $ the name of the slideline that needs to
                                         $ when points along it move

  $ relaxer controllers
  mk +controller(c0)/time_ranges
    tstart = 0.
    tstop  = 1.e99
  mk +controller(c1)/slideremove
    kbcslidename = "slip1"
    active        = "during"             $ this relaxer will only be active in the
```

Figure 2. Selected lines from the Shaped Charge Example input deck.

Key features of FLIP

FLIP allows the user to easily and efficiently navigate their models using a point-and-click approach, rather than relying on a text editor.

A built-in online help menu is accessible by the keyboard command Control-H. Table I provides a complete list of keyboard shortcuts presently available in FLIP.

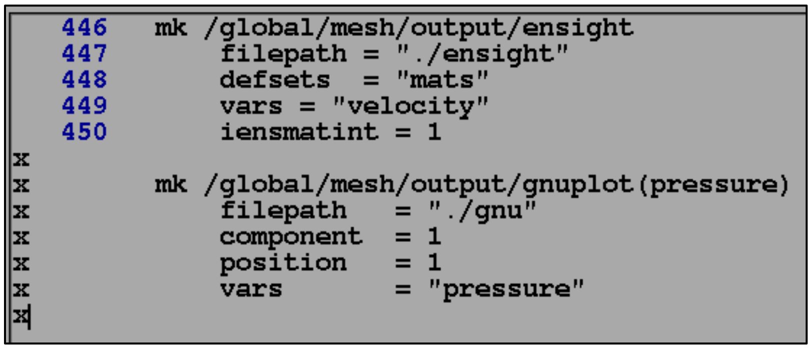
Table I. FLIP's keyboard shortcuts

Keyboard shortcut	Description
Control+H	Print the online help in lower-right panel
Control+F	Open file browser
Control+C	Clear node data from all panels
Control+P	Print the entire input deck in lower-right panel
Control+Z	Commit edits (to be done after editing each node)
Control-S	Save all committed edits to a modified file
Left/Right Arrows	Navigate between the panels
Escape	Quit FLIP

FLIP handles both FLAG "mk +", "cd" shortcut commands, and automatically imports the contents of "include" files to produce a master model. This model's input deck may be viewed it its entirety by using the Control+P shortcut.

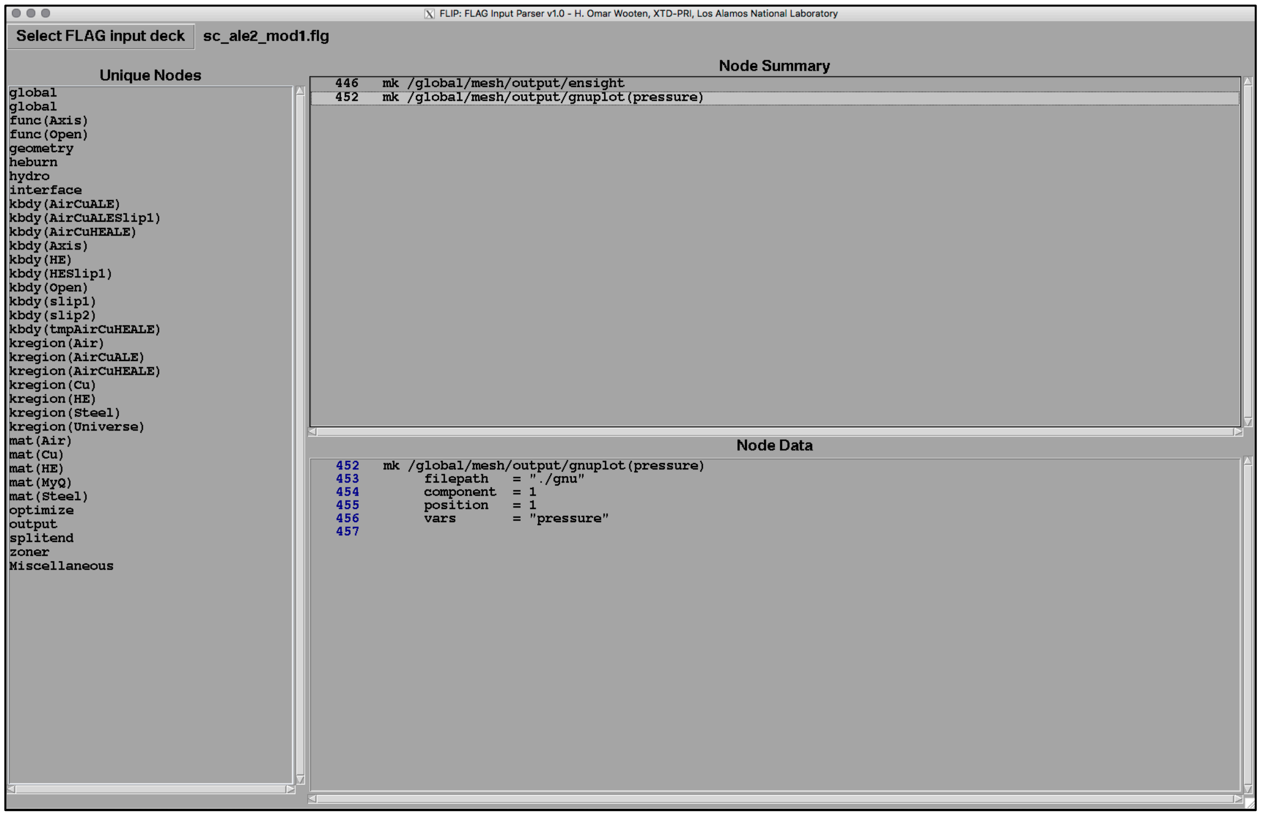
FLIP also provides an editing capability. Nodes, parameters, and variables that are displayed in Panel 3 may be directly edited by the user. Upon making any changes, the keyboard shortcut Control+Z commits those edits to a new master model. This process may be repeated until all of the desired edits are complete. Control-S saves all committed changes to a modified file, with "_modX" appended to it's filename, where X is an increasing integer representing increasing versions of modified files. FLIP does not overwrite the original file.

In addition to editing variables and parameters, FLIP also allows the user to add additional nodes and associated parameters and variables. An example is shown in Figure 2, where we add a GNU Plot output node and a desired file path in the lower right panel. The "x" indicates to FLIP that this line is to be added to a new master model. After committing the changes and saving the model to a new mod file, Figure 3 shows the new output node and it's content within the model database.



```
446    mk /global/mesh/output/ensight
447        filepath = "./ensight"
448        defsets  = "mats"
449        vars = "velocity"
450        iensmatint = 1
x
x      mk /global/mesh/output/gnuplot(pressure)
x          filepath    = "./gnu"
x          component   = 1
x          position    = 1
x          vars        = "pressure"
x
```

Figure 3. Demonstration of FLIP's editing capability. In this example, a gnuplot output node and event scheduler command are added. The "x" in the first column is required when adding new lines, and node/parameters must begin after the 8th column. When committed and saved, these changes appear in a modified master model input deck (shown in Figure 4).


FLIP: FLAG Input Parser v1.0 - H. Omar Wooten, XTD-PRI, Los Alamos National Laboratory
Select FLAG input deck
sc_ale2_mod1.flg
Unique Nodes
Node Summary
Node Data

```
global
global
func(Axis)
func(Open)
geometry
heburn
hydro
interface
kbdy(AirCuALE)
kbdy(AirCuALESlip1)
kbdy(AirCuHEALE)
kbdy(Axis)
kbdy(HE)
kbdy(HESlip1)
kbdy(Open)
kbdy(slip1)
kbdy(slip2)
kbdy(tmpAirCuHEALE)
kregion(Air)
kregion(AirCuALE)
kregion(AirCuHEALE)
kregion(Cu)
kregion(HE)
kregion(Steel)
kregion(Universe)
mat(Air)
mat(Cu)
mat(HE)
mat(MyQ)
mat(Steel)
optimize
output
splitend
zoner
Miscellaneous
```

```
446   mk /global/mesh/output/ensight
452   mk /global/mesh/output/gnuplot(pressure)
```

```
452   mk /global/mesh/output/gnuplot(pressure)
453        filepath    = "./gnu"
454        component   = 1
455        position    = 1
456        vars        = "pressure"
457
```

Figure 4. The modified master model now includes the nodes and parameters for the gnuplot node.

Four color display schemes are available for FLIP users, as described in Table II. The syntax for launching FLIP with, for example, color scheme 3 is `flip.py -c 3`.

Table II. Color schemes available in FLIP.

Color scheme	Background	Foreground	Line Numbers
1	Dark grey	Black	Dark blue
2	Light grey	Black	Dark blue
3	Black	Grey	Yellow
4	White	Black	Grey

Conclusions

FLIP provides FLAG users with a means of efficiently reviewing, debugging, and editing their models with an organized, easily navigable user interface. FLIP is available on the High Performance Computing (HPC) machines in the following directory: `/yellow/users/hasani/public/flip`.

Acknowledgements

The authors would like to acknowledge and thank the beta test users James Hill (XCP-3), David Becker (XTD-PRI), Von Whitley (XTD-SS), Eugene Dougherty (XTD-PRI), and Lori Pritchett-Sheats (XTD-PRI) for feedback and helpful suggestions.

References

1. Hill, J., "FLAG User's Manual," LA-CP-16-20334, Los Alamos National Laboratory report, 2017.

2. www.python.org