CritOpS Documentation

Release 2.0.0

Andrew Johnson

Apr 17, 2017

CONTENTS:

CHAPTER

ONE

INTRO

This is the documentation for *CritOpS*, a Critical Optimization Search tool for use with *NEWT'[1]. *CritOpS* is designed to iteratively modify inputs for 'NEWT* to obtain a desired eigenvalue. More documentation will be added before the final release of this code, including examples and validation testing.

1.1 Usage

CritOpS can be run from the terminal while in the directory outside the *critops* folder with the command

```
$ python critops <mainfile> <paramfile>
```

Where <*mainfile*> is a valid *NEWT* input file with some variables in place of valid values and <*paramfile*> is the file that contains limits on iteration parameters, desired k-eff, and indicates the variable to be iterated upon. See `testing/iter_tester.inp` and `testing/param_tester.txt` for one example case.

1.2 References

[1]: M. D. DeHart, and S. Bowman, "Reactor Physics Methods and Analysis Capabilities in SCALE," Nuclear Technology, Technical Paper vol. 174, no.2, pp. 196-213, 2011.

CHAPTER

TWO

ITERATOR

NRE6401 - Molten Salt Reactor

CritOpS

1. Johnson

Objective: Main file for controlling the iteration scheme

Functions:

iter_main: Landing function that drives the iteration

makefile: Write the new output file using the value from iteration _iter

update_itervar: Simple function to update the iteration variables.

parse_scale_out_eig: Read through the SCALE output file specified by _ofile and return status and eigenvalue (if present)

critops.iterator.**itermain**(*tmp_list: (<class 'list'>, <class 'tuple'>)*, *file_name: str*, *iter_vars: dict*, *kwargs: dict*)

Main function for controlling the iteration

Parameters

- **tmp_list** – List of lines from template file
- **file_name** – Name of template file
- **iter_vars** – Dictionary of iteration variables and their starting, minima, and maximum values
- **kwargs** – Additional keyword arguments

Returns k_vec: List of progression of eigenvalue through iteration procedure

Returns iter_vecs: Dictionary of iteration and their values through iteration procedure

Returns

conv_type - reason for exiting iter_main

0: Accurately converged to target eigenvalue in specified iterations

1: iter_var exceeded specified maximum twice

-1: iter_var exceeded specified minimum twice

2: Reached iteration limit without reaching target eigenvalue

-2: Previous two k are close to similar

critops.iterator.**update_itervar**(*iter_vars: dict*, *iter_vec: dict*, *kvec: (<class 'list'>*, *<class 'tuple'>)*, *ktarg: float*)

Simple function to update the iteration variables. Currently set up for a positive feedback on the variables. I.e. increasing each iteration variable will increase k

Parameters

- **iter_vars** – Dictionary of iteration variables and their minima/maxima
- **iter_vec** – Dictionary of iteration variables and their values through the iteratio procedure
- **kvec** – Vector of eigenvalues
- **ktarg** – Target eigenvalue

Returns status status = 0 if the updated value is inside the intended range status = 1 if the desired updated value is greater than the specified maximum of the parameter and the max value is used as the updated value status = -1 if the desired updated value is less than the specified maximum of the parameter and the minimum value is used as the updated value

critops.iterator.**parse_scale_out_eig**(*_ofile: str*, ***kwargs*)

Read through the SCALE output file specified by _ofile and return status and eigenvalue (if present)

Parameters **_ofile** – SCALE .out file

Returns

Status, eigenvalue

status = True if output file exists and eigenvalue was extracted status = False if output file exists but no eigenvalue was found (possible error in input file syntax) exit operation if no output file found

CHAPTER

THREE

READINPUTS

NRE6401 - Molten Salt Reactor

CritOps

Objective: Read the inputs, update global variables, and check for proper variable usage

Functions:

> check_inputs: make sure values in global_parameters are good for running read_param: Read the parameter file and update values in globalparams readmain: Main driver for reading and processing the input files

critops.readinputs.**readmain**(*tmp_file*, *param_file*, *kwargs: dict*)

Main driver for reading and processing input files.

Parameters

- **tmp_file** – Template input file
- **param_file** – Parameter file
- **kwargs** – Additional arguments - verbose (True) - status updates - output (None) - print to screen - Plus additional iteration parameters

Returns List of valid template file lines and dictionary of interation variables Updates kwargs based on values in param_file

critops.readinputs.**read_param**(*_pfile*, ***kwargs*)

Read the parameter file and update values in kwargs

Parameters **_pfile** – Parameter file

Returns iter_vars: Dictionary of iteration variables and their starting, minima, and maximum values

Returns updated keyword arguments

critops.readinputs.**check_inputs**(*temp_lines: list*, *iter_vars: dict*, ***kwargs*)

Run over the inputs and make sure things are good for operation

CHAPTER

FOUR

OUTPUTS

NRE6401 - Molten Salt Reactor

CritOpS

Objective: Functions for reading SCALE output files and writing output files

Functions:

parse_scale_output: Parse through the SCALE output file and return status

critops.outputs.**output_landing**(*iter_vecs: dict*, *k_vec: (<class 'list'>*, *<class 'tuple'>)*, *_outtype: int*, ***kwargs*)

Write the output file according to the type of output

Parameters

- **iter_vecs** – Dictionary with iteration variables and their values through the procedure
- **k_vec** – Vector of eigenvalues
- **_outtype** – Flag indicating the reason the program terminated - 0 Nothing went wrong - 1 Desired update value for iteration parameter twice exceeded the maximum value from the parameter file - -1 Desired update value for iteration parameter twice exceeded the minimum value from the parameter file - 2 Exceeded the total number of iterations allotted - -2 No excessive change in eigenvalue

Returns

CHAPTER

FIVE

INDICES AND TABLES

- genindex
- modindex
- search

PYTHON MODULE INDEX

C