

SPAMS

v1.2

Generated by Doxygen 1.9.1

1 Main Page	1
1.1 Introduction	1
1.2 Contributors	1
2 Maths and Theory	3
2.1 1 Mathematics and Algorithms	3
3 Uncertainty Quantification	5
3.1 Sensitivity Analysis	5
3.2 Uncertainty Propagation	5
4 Modules Index	7
4.1 Modules List	7
5 File Index	9
5.1 File List	9
6 Module Documentation	11
6.1 checkpointing Module Reference	11
6.1.1 Detailed Description	11
6.2 convergence_test Namespace Reference	11
6.2.1 Detailed Description	11
6.2.2 Variable Documentation	12
6.2.2.1 dt	12
6.2.2.2 dt_vals	12
6.2.2.3 final_mass_loss	12
6.2.2.4 final_voltage	12
6.2.2.5 mass_loss	12
6.2.2.6 nodenum	13
6.2.2.7 v	13
6.3 mixed_parallel_strat_test Namespace Reference	13
6.3.1 Variable Documentation	13
6.3.1.1 dt	13
6.3.1.2 end	13
6.3.1.3 label	14
6.3.1.4 n	14
6.3.1.5 nodenum	14
6.3.1.6 num_cores	14
6.3.1.7 num_nodes	14
6.3.1.8 start	14
6.3.1.9 times	15
6.4 MKL_threads_test Namespace Reference	15
6.4.1 Variable Documentation	15
6.4.1.1 dt	15

6.4.1.2 end	15
6.4.1.3 label	15
6.4.1.4 n	16
6.4.1.5 nodenum	16
6.4.1.6 num_nodes	16
6.4.1.7 num_threads	16
6.4.1.8 start	16
6.4.1.9 times	16
6.5 nc_output Module Reference	17
6.5.1 Detailed Description	17
6.5.2 Function/Subroutine Documentation	17
6.5.2.1 check()	17
6.5.2.2 output_cstorage()	17
6.6 plotter Namespace Reference	17
6.6.1 Detailed Description	18
6.6.2 Function Documentation	18
6.6.2.1 animated_conc_plot()	18
6.6.2.2 full_battery_GITT_plots()	19
6.6.2.3 gen_half_cell_voltage()	19
6.6.2.4 gen_plots()	19
6.6.2.5 get_avg_li_conc()	20
6.6.2.6 plot_halfcell_GITT_result()	20
6.6.2.7 read_input_current()	20
6.6.2.8 read_output_file()	21
6.6.2.9 voltage_current_plot()	21
6.7 read_inputs Module Reference	22
6.7.1 Detailed Description	22
6.8 sensitivity_analysis Namespace Reference	22
6.8.1 Detailed Description	22
6.8.2 Function Documentation	22
6.8.2.1 sensitivity_over_time()	23
6.8.2.2 update()	23
6.8.3 Variable Documentation	23
6.8.3.1 ani	23
6.8.3.2 ax	23
6.8.3.3 fig	23
6.8.3.4 filename	23
6.8.3.5 fps	24
6.8.3.6 scaled_sensitivities_matrix	24
6.8.3.7 writer	24
6.8.3.8 x_labels	24
6.9 Uncertainty_Propagation Namespace Reference	24

6.9.1 Detailed Description	26
6.9.2 Function Documentation	26
6.9.2.1 in_out_easy_peasy()	26
6.9.3 Variable Documentation	26
6.9.3.1 a_dist	26
6.9.3.2 a_samples	27
6.9.3.3 alpha	27
6.9.3.4 axs	27
6.9.3.5 axss	27
6.9.3.6 c0_dist	27
6.9.3.7 c0_samples	27
6.9.3.8 counter	28
6.9.3.9 cstore_neg	28
6.9.3.10 cstore_pos	28
6.9.3.11 D_dist	28
6.9.3.12 D_samples	28
6.9.3.13 dr_neg	28
6.9.3.14 dr_pos	29
6.9.3.15 dt	29
6.9.3.16 edge_conc_vals_neg	29
6.9.3.17 edge_conc_vals_pos	29
6.9.3.18 electrode_neg	29
6.9.3.19 electrode_pos	29
6.9.3.20 False	30
6.9.3.21 fig	30
6.9.3.22 figg	30
6.9.3.23 figsize	30
6.9.3.24 fixed_a	30
6.9.3.25 fixed_c0	30
6.9.3.26 fixed_D	31
6.9.3.27 fixed_L	31
6.9.3.28 fixed_R	31
6.9.3.29 full_voltage	31
6.9.3.30 i_app_data_neg	31
6.9.3.31 i_app_data_pos	31
6.9.3.32 iapp	32
6.9.3.33 L_dist	32
6.9.3.34 L_samples	32
6.9.3.35 neg_voltage	32
6.9.3.36 negparams	32
6.9.3.37 nodenum	32
6.9.3.38 nprocs	33

6.9.3.39 num_samples	33
6.9.3.40 output_filename_negative	33
6.9.3.41 output_filename_positive	33
6.9.3.42 parameters	33
6.9.3.43 params	33
6.9.3.44 pos_voltage	34
6.9.3.45 posparams	34
6.9.3.46 R_dist	34
6.9.3.47 R_neg	34
6.9.3.48 R_pos	34
6.9.3.49 R_samples	34
6.9.3.50 resol	35
6.9.3.51 sharex	35
6.9.3.52 sharey	35
6.9.3.53 time	35
6.9.3.54 time_axis	35
6.9.3.55 True	35
6.9.3.56 tsteps	36
6.9.3.57 voltage_curves	36
6.9.3.58 x	36
6.10 unit_test Namespace Reference	36
6.10.1 Function Documentation	36
6.10.1.1 run_unit_test()	36
6.11 user_input Namespace Reference	36
6.11.1 Detailed Description	37
6.11.2 Variable Documentation	38
6.11.2.1 a	38
6.11.2.2 c0	38
6.11.2.3 checkpoint	38
6.11.2.4 cmax_neg_sim	38
6.11.2.5 cmax_pos_sim	39
6.11.2.6 D	39
6.11.2.7 dt	39
6.11.2.8 electrode_charge	39
6.11.2.9 iapp	39
6.11.2.10 iapp_label	39
6.11.2.11 K_neg	40
6.11.2.12 K_pos	40
6.11.2.13 L	40
6.11.2.14 n	40
6.11.2.15 plot_params_pos	40
6.11.2.16 pos_params	40

6.11.2.17 R	41
6.11.2.18 solver_input_filename	41
6.11.2.19 tsteps	41
6.12 user_input_full_battery_GITT Namespace Reference	41
6.12.1 Detailed Description	42
6.12.2 Variable Documentation	42
6.12.2.1 a_neg	43
6.12.2.2 a_pos	43
6.12.2.3 animation_interval_time	43
6.12.2.4 c0	43
6.12.2.5 c0_neg	43
6.12.2.6 c0_pos	43
6.12.2.7 cmax_neg_sim	44
6.12.2.8 cmax_pos_sim	44
6.12.2.9 currents	44
6.12.2.10 D_neg	44
6.12.2.11 D_pos	44
6.12.2.12 dt	44
6.12.2.13 electrode_charge_neg	45
6.12.2.14 electrode_charge_pos	45
6.12.2.15 iapp_label_neg	45
6.12.2.16 iapp_label_pos	45
6.12.2.17 iapp_neg	45
6.12.2.18 iapp_pos	45
6.12.2.19 K_neg	46
6.12.2.20 K_pos	46
6.12.2.21 L_neg	46
6.12.2.22 L_pos	46
6.12.2.23 n	46
6.12.2.24 neg_params	46
6.12.2.25 nprocs	47
6.12.2.26 nsteps	47
6.12.2.27 output_filename_negative	47
6.12.2.28 output_filename_positive	47
6.12.2.29 params_neg	47
6.12.2.30 params_pos	48
6.12.2.31 plot_params_neg	48
6.12.2.32 plot_params_pos	48
6.12.2.33 pos_params	48
6.12.2.34 R_neg	48
6.12.2.35 R_pos	48
6.12.2.36 run_times	49

6.12.2.37 SparsifyAnimation	49
6.12.2.38 start_times	49
6.12.2.39 True	49
6.12.2.40 tsteps	49
6.12.2.41 wait_times	49
6.13 user_input_mod Namespace Reference	50
6.13.1 Detailed Description	50
6.13.2 Function Documentation	51
6.13.2.1 call_solver()	51
6.13.2.2 full_battery_simulation()	51
6.13.2.3 get_GITT_initial_concs()	52
6.13.2.4 GITT_full_cell()	52
6.13.2.5 GITT_half_cell()	53
6.13.2.6 iapp_constant_setup()	54
6.13.2.7 iapp_read_csv()	54
6.13.2.8 iapp_step_setup()	54
6.13.2.9 optimise_parallelism()	55
6.13.2.10 set_defaults_neg()	55
6.13.2.11 set_defaults_pos()	55
6.13.2.12 verify_iapp()	56
6.13.2.13 verify_params()	56
6.13.2.14 write_to_file()	57
6.14 user_input_parallel_simulation Namespace Reference	57
6.14.1 Detailed Description	58
6.14.2 Variable Documentation	58
6.14.2.1 a	58
6.14.2.2 Animation	58
6.14.2.3 c0	58
6.14.2.4 cmax_neg_sim	59
6.14.2.5 cmax_pos_sim	59
6.14.2.6 currents	59
6.14.2.7 D	59
6.14.2.8 dt	59
6.14.2.9 electrode	59
6.14.2.10 electrode_charge	60
6.14.2.11 iapp	60
6.14.2.12 iapp_label	60
6.14.2.13 K_neg	60
6.14.2.14 K_pos	60
6.14.2.15 L	60
6.14.2.16 n	61
6.14.2.17 neg_params	61

6.14.2.18 nprocs	61
6.14.2.19 nsteps	61
6.14.2.20 output_filename	61
6.14.2.21 params	61
6.14.2.22 plot_params_neg	62
6.14.2.23 R	62
6.14.2.24 run_times	62
6.14.2.25 SparsifyAnimation	62
6.14.2.26 start_times	62
6.14.2.27 True	62
6.14.2.28 tsteps	62
6.14.2.29 wait_times	62
7 File Documentation	63
7.1 /home/chem/msuvnj/PX915_GroupB_22-23/checkpointing.f90 File Reference	63
7.1.1 Variable Documentation	64
7.1.1.1 a_small	64
7.1.1.2 c	64
7.1.1.3 c0	64
7.1.1.4 checkpoint_name	65
7.1.1.5 cstorage	65
7.1.1.6 d	65
7.1.1.7 dimid_n	65
7.1.1.8 dimid_tsteps	65
7.1.1.9 dimids_cstorage	65
7.1.1.10 dimids_parameter	66
7.1.1.11 dir	66
7.1.1.12 dt	66
7.1.1.13 electrode_charge	66
7.1.1.14 file_ext	66
7.1.1.15 file_extension	66
7.1.1.16 file_test	67
7.1.1.17 filename	67
7.1.1.18 filename_length	67
7.1.1.19 filepath	67
7.1.1.20 freq	67
7.1.1.21 freq_in	67
7.1.1.22 freq_remainder	68
7.1.1.23 iapp	68
7.1.1.24 l	68
7.1.1.25 n	68
7.1.1.26 ncid	68

7.1.1.27 parse_idx	68
7.1.1.28 r	69
7.1.1.29 status	69
7.1.1.30 tstep	69
7.1.1.31 tstep_init	69
7.1.1.32 tstep_str	69
7.1.1.33 tsteps	69
7.1.1.34 varid	70
7.1.1.35 varid_a	70
7.1.1.36 varid_c	70
7.1.1.37 varid_cstorage	70
7.1.1.38 varid_d	70
7.1.1.39 varid_dt	70
7.1.1.40 varid_ec	71
7.1.1.41 varid_iapp	71
7.1.1.42 varid_l	71
7.1.1.43 varid_n	71
7.1.1.44 varid_r	71
7.1.1.45 varid_tstep	71
7.1.1.46 varid_tsteps	72
7.2 /home/chem/msuvnj/PX915_GroupB_22-23/convergence_test.py File Reference	72
7.3 math.md File Reference	72
7.4 uq.md File Reference	72
7.5 /home/chem/msuvnj/PX915_GroupB_22-23/finite_diff_solver.f90 File Reference	72
7.5.1 Function/Subroutine Documentation	72
7.5.1.1 main()	73
7.6 /home/chem/msuvnj/PX915_GroupB_22-23/mixed_parallel_strat_test.py File Reference	73
7.7 /home/chem/msuvnj/PX915_GroupB_22-23/MKL_threads_test.py File Reference	73
7.8 /home/chem/msuvnj/PX915_GroupB_22-23/nc_output.f90 File Reference	74
7.9 /home/chem/msuvnj/PX915_GroupB_22-23/plotter.py File Reference	74
7.10 /home/chem/msuvnj/PX915_GroupB_22-23/read_inputs.f90 File Reference	74
7.10.1 Variable Documentation	75
7.10.1.1 a_small	75
7.10.1.2 arg	75
7.10.1.3 asterix	76
7.10.1.4 c0	76
7.10.1.5 d	76
7.10.1.6 dt	76
7.10.1.7 electrode_charge	76
7.10.1.8 file_id	76
7.10.1.9 filename	77
7.10.1.10 i	77

7.10.1.11 iapp	77
7.10.1.12 invalid_param	77
7.10.1.13 ios	77
7.10.1.14 l	77
7.10.1.15 line	78
7.10.1.16 n	78
7.10.1.17 n_params	78
7.10.1.18 n_params_expected	78
7.10.1.19 name	78
7.10.1.20 num_args	78
7.10.1.21 param_read	79
7.10.1.22 parse_idx	79
7.10.1.23 parser	79
7.10.1.24 r	79
7.10.1.25 read_temp	79
7.10.1.26 tsteps	79
7.10.1.27 val	80
7.11 /home/chem/msuvnj/PX915_GroupB_22-23/sensitivity_analysis.py File Reference	80
7.12 /home/chem/msuvnj/PX915_GroupB_22-23/Uncertainty_Propagation.py File Reference	80
7.13 /home/chem/msuvnj/PX915_GroupB_22-23/unit_test.py File Reference	82
7.14 /home/chem/msuvnj/PX915_GroupB_22-23/user_input.py File Reference	82
7.15 /home/chem/msuvnj/PX915_GroupB_22-23/user_input_full_battery_GITT.py File Reference	83
7.16 /home/chem/msuvnj/PX915_GroupB_22-23/user_input_mod.py File Reference	84
7.17 /home/chem/msuvnj/PX915_GroupB_22-23/user_input_parallel_simulation.py File Reference	85
Index	87

Chapter 1

Main Page

A single particle model (SPM) by Group B for the 2023 PX915 group project.

1.1 Introduction

SPAMS models the charging and discharging of a lithium ion battery using a Crank-Nicolson semi-implicit finite difference scheme to obtain the concentration of lithium in a sphere, $c(iapp, r)$, at each time step, and includes the following features:

- Apply a constant, stepwise or custom current
- Options for parallelism
- Extend the model to a full battery
- Uncertainty quantification

1.2 Contributors

Fraser Birks, Laura Cairns, Sebastian Dooley, Arielle Fitkin, Jake Eller, and Yu Lei

HetSys CDT, University of Warwick

Chapter 2

Maths and Theory

A half single particle model by Group B for the 2023 PX915 group project.

2.1 1 Mathematics and Algorithms

A Crank-Nicolson semi-implicit finite difference scheme is used to approximate the partial differential equation (PDE) solution to obtain the concentration of lithium in the sphere, $c(i_{app}, r)$, at each time step. This is accurate to second order both spatially and temporally, i.e. $O(\Delta r^2)$ & $O(\Delta t^2)$.

A key benefit of the Crank-Nicolson scheme is that it is unconditionally stable for the spherically symmetric diffusion equation (the PDE of interest), thus not restricting the user's choice of step size spatially or temporally. Although, due to being second order, the accuracy may be impacted by a larger step size.

The implicit nature of the Crank-Nicolson scheme makes it ideal for numerically approximating solutions to both stiff and highly nonlinear diffusion problems, providing the user a strong base from which to update this solution framework.

Additionally, the Crank-Nicolson scheme is consistent meaning that 'the error of the numerical method converges to zero as the grid spacing, or time step, reduces to zero from above, under certain regularity conditions on the solution.'

The consistency and stability of the Crank-Nicolson scheme means it satisfies the Lax equivalence theorem, which states that a numerical method is convergent if and only if it is both consistent and stable.

Specific Equation:

Generally:

This is an average of standard forward and backward Euler methods.

Boundary conditions are treated through the use of ghost nodes, which assign values for the function of interest to regions just beyond the domain of the problem in an attempt to approximate the first derivatives present in the Neumann boundary conditions specified. For example,

$$\left[\frac{\partial c}{\partial r}\right]_{r=0} \approx \frac{c_1 - c_{-1}}{2\Delta r} = 0 \text{ iff } c_1 = c_{-1}.$$

The accuracy of the scheme is such that it is equivalent to the interior solved points.

Finally, a general initial condition is specified of the form

$$c = c_0 \text{ at } t=0.$$

The form of c_0 is assumed to be constant in R.

Chapter 3

Uncertainty Quantification

Below are the results of some uncertainty quantification that has been carried out for SPAMS.

3.1 Sensitivity Analysis

First order sensitivity analysis.

3.2 Uncertainty Propagation

Input Parameter Distributions

Resulting distribution of $v(t)$

Chapter 4

Modules Index

4.1 Modules List

Here is a list of all modules with brief descriptions:

checkpointing	
Module to read and write checkpoint files	11
convergence_test	
Test convergence for dt and number of nodes	11
mixed_parallel_strat_test	13
MKL_threads_test	15
nc_output	
Writes results to a netCDF file	17
plotter	
Functions for visualisation of solver outputs	17
read_inputs	
Module to read user inputs	22
sensitivity_analysis	
First order sensitivity analysis	22
Uncertainty_Propagation	
Performs uncertainty propagation	24
unit_test	36
user_input	
Sets up the user inputs and executes the solver	36
user_input_full_battery_GITT	41
user_input_mod	
Package containing the functions needed to set up the user input parameters and execute the solver	50
user_input_parallel_simulation	
Running script which exists to demonstrate how one uses the GITT function to run simulations in parallel	57

Chapter 5

File Index

5.1 File List

Here is a list of all files with brief descriptions:

/home/chem/msuvnj/PX915_GroupB_22-23/checkpointing.f90	63
/home/chem/msuvnj/PX915_GroupB_22-23/convergence_test.py	72
/home/chem/msuvnj/PX915_GroupB_22-23/finite_diff_solver.f90	72
/home/chem/msuvnj/PX915_GroupB_22-23/mixed_parallel_strat_test.py	73
/home/chem/msuvnj/PX915_GroupB_22-23/MKL_threads_test.py	73
/home/chem/msuvnj/PX915_GroupB_22-23/nc_output.f90	74
/home/chem/msuvnj/PX915_GroupB_22-23/plotter.py	74
/home/chem/msuvnj/PX915_GroupB_22-23/read_inputs.f90	74
/home/chem/msuvnj/PX915_GroupB_22-23/sensitivity_analysis.py	80
/home/chem/msuvnj/PX915_GroupB_22-23/Uncertainty_Propagation.py	80
/home/chem/msuvnj/PX915_GroupB_22-23/unit_test.py	82
/home/chem/msuvnj/PX915_GroupB_22-23/user_input.py	82
/home/chem/msuvnj/PX915_GroupB_22-23/user_input_full_battery_GITT.py	83
/home/chem/msuvnj/PX915_GroupB_22-23/user_input_mod.py	84
/home/chem/msuvnj/PX915_GroupB_22-23/user_input_parallel_simulation.py	85

Chapter 6

Module Documentation

6.1 checkpointing Module Reference

Module to read and write checkpoint files.

6.1.1 Detailed Description

Module to read and write checkpoint files.

Module contains functions that write checkpoint files and read them in as user input.

6.2 convergence_test Namespace Reference

Test convergence for dt and number of nodes.

Variables

- list `dt_vals` = [0.01,0.05,0.1,0.5,1.0,5.0,10.0]
- `final_voltage` = np.array([])
- `final_mass_loss` = np.array([])
- `v`
- `mass_loss`
- `dt`
- `nodenum`

6.2.1 Detailed Description

Test convergence for dt and number of nodes.

6.2.2 Variable Documentation

6.2.2.1 dt

dt

Definition at line 12 of file convergence_test.py.

6.2.2.2 dt_vals

```
list dt_vals = [0.01,0.05,0.1,0.5,1.0,5.0,10.0]
```

Definition at line 8 of file convergence_test.py.

6.2.2.3 final_mass_loss

```
final_mass_loss = np.array([])
```

Definition at line 10 of file convergence_test.py.

6.2.2.4 final_voltage

```
final_voltage = np.array([])
```

Definition at line 9 of file convergence_test.py.

6.2.2.5 mass_loss

mass_loss

Definition at line 12 of file convergence_test.py.

6.2.2.6 nodenum

`nodenum`

Definition at line 12 of file `convergence_test.py`.

6.2.2.7 v

`v`

Definition at line 12 of file `convergence_test.py`.

6.3 mixed_parallel_strat_test Namespace Reference

Variables

- list `num_cores` = [1,2,4,8]
- list `num_nodes` = [2000]
- `times` = `np.zeros((len(num_cores),len(num_nodes)))`
- `start` = `time.time()`
- `nodenum`
- `n`
- `dt`
- `end` = `time.time()`
- `label`

6.3.1 Variable Documentation

6.3.1.1 dt

`dt`

Definition at line 13 of file `mixed_parallel_strat_test.py`.

6.3.1.2 end

`end = time.time()`

Definition at line 14 of file `mixed_parallel_strat_test.py`.

6.3.1.3 label

```
label
```

Definition at line 19 of file mixed_parallel_strat_test.py.

6.3.1.4 n

```
n
```

Definition at line 13 of file mixed_parallel_strat_test.py.

6.3.1.5 nodenum

```
nodenum
```

Definition at line 13 of file mixed_parallel_strat_test.py.

6.3.1.6 num_cores

```
num_cores = [1,2,4,8]
```

Definition at line 7 of file mixed_parallel_strat_test.py.

6.3.1.7 num_nodes

```
list num_nodes = [2000]
```

Definition at line 8 of file mixed_parallel_strat_test.py.

6.3.1.8 start

```
start = time.time()
```

Definition at line 12 of file mixed_parallel_strat_test.py.

6.3.1.9 times

```
times = np.zeros([len(num_cores), len(num_nodes)])
```

Definition at line 9 of file mixed_parallel_strat_test.py.

6.4 MKL_threads_test Namespace Reference

Variables

- list `num_threads` = [1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,20]
- list `num_nodes` = [100,500,1000,2000]
- `times` = np.zeros([19,4])
- `start` = time.time()
- `nodenum`
- `n`
- `dt`
- `end` = time.time()
- `label`

6.4.1 Variable Documentation

6.4.1.1 dt

`dt`

Definition at line 16 of file MKL_threads_test.py.

6.4.1.2 end

```
end = time.time()
```

Definition at line 17 of file MKL_threads_test.py.

6.4.1.3 label

`label`

Definition at line 22 of file MKL_threads_test.py.

6.4.1.4 n

n

Definition at line 16 of file MKL_threads_test.py.

6.4.1.5 nodenum

nodenum

Definition at line 16 of file MKL_threads_test.py.

6.4.1.6 num_nodes

```
list num_nodes = [100,500,1000,2000]
```

Definition at line 8 of file MKL_threads_test.py.

6.4.1.7 num_threads

```
list num_threads = [1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,20]
```

Definition at line 7 of file MKL_threads_test.py.

6.4.1.8 start

```
start = time.time()
```

Definition at line 15 of file MKL_threads_test.py.

6.4.1.9 times

```
times = np.zeros([19,4])
```

Definition at line 9 of file MKL_threads_test.py.

6.5 nc_output Module Reference

Writes results to a netCDF file.

Functions/Subroutines

- subroutine [check](#) ([status](#))
- subroutine [output_cstorage](#) ([cstorage](#), [n](#), [tsteps](#), [R](#), [time_axis](#), [electrode_charge](#), [filename](#))

6.5.1 Detailed Description

Writes results to a netCDF file.

6.5.2 Function/Subroutine Documentation

6.5.2.1 [check\(\)](#)

```
subroutine nc_output::check (  
    integer, intent(in) status )
```

Definition at line 9 of file nc_output.f90.

6.5.2.2 [output_cstorage\(\)](#)

```
subroutine nc_output::output_cstorage (  
    real(real64), dimension(n, tsteps), intent(in) cstorage,  
    integer, intent(in) n,  
    integer, intent(in) tsteps,  
    real(real64), intent(in) R,  
    real(real64), dimension(tsteps), intent(in) time_axis,  
    character(len=*), intent(in) electrode_charge,  
    character(len=*), intent(in) filename )
```

Definition at line 20 of file nc_output.f90.

6.6 [plotter](#) Namespace Reference

Functions for visualisation of solver outputs.

Functions

- def `read_output_file` (filename, step_num=None)
Reads in data from user input and solver output using NetCDF.
- def `read_input_current` (filename, step_num=None)
Reads in the applied current.
- def `animated_conc_plot` (intervaltime, dr, `tsteps`, nodenum, cstore, time_axis, SaveFinalState=False, SparsifyAnimation=False)
Saves animation of lithium concentration over time.
- def `gen_half_cell_voltage` (edge_conc_vals, i_app_data, electrode, `tsteps`, pos_params=None, neg_params=None)
- def `voltage_current_plot` (electrode, cstore, time_axis, i_app_data, `tsteps`, pos_params=None, neg_params=None)
Calculates voltages and plots as a function of time.
- def `plot_halfcell_GITT_result` (filename, start_times, electrode, pos_params=None, neg_params=None, Animation=False, SaveFinalState=False, SparsifyAnimation=True, animation_interval_time=10)
- def `gen_plots` (filename, pos_params=None, neg_params=None, animation_interval_time=10, SaveFinalState=True, SparsifyAnimation=False)
- def `full_battery_GITT_plots` (filename_positive, filename_negative, start_times, pos_params=None, neg_params=None, SparsifyAnimation=True, animation_interval_time=10)
- def `get_avg_li_conc` (c_vals, r_vals)
Gets the average concentration of lithium at a given timestep using trapezium rule inside the sphere.

6.6.1 Detailed Description

Functions for visualisation of solver outputs.

(Plots are saved locally.)

Contains functions that produce:

- An animation showing the concentration of lithium as a function of time and space. (There is an additional feature for saving the concentration profile at the end of the simulation as a png.)
- Plots of both voltage and applied current over time.

6.6.2 Function Documentation

6.6.2.1 `animated_conc_plot()`

```
def plotter.animated_conc_plot (
    intervaltime,
    dr,
    tsteps,
    nodenum,
    cstore,
    time_axis,
    SaveFinalState = False,
    SparsifyAnimation = False )
```

Saves animation of lithium concentration over time.

Saves the animation 'concentration_animation.gif', that displays the evolution of lithium concentration over time. If passed the argument 'SaveFinalState = True', it saves an image 'final_state.png' of lithium concentration across the sphere at the final timestep.

Definition at line 89 of file plotter.py.

6.6.2.2 full_battery_GITT_plots()

```
def plotter.full_battery_GITT_plots (
    filename_positive,
    filename_negative,
    start_times,
    pos_params = None,
    neg_params = None,
    SparsifyAnimation = True,
    animation_interval_time = 10 )
```

Definition at line 382 of file plotter.py.

6.6.2.3 gen_half_cell_voltage()

```
def plotter.gen_half_cell_voltage (
    edge_conc_vals,
    i_app_data,
    electrode,
    tsteps,
    pos_params = None,
    neg_params = None )
```

At time = 0s, the output of `j_function` can be 0 if Li concentration at the sphere edge is 0. This results in an error as `V_function` involves division by the output of `j_function`.

Parameters

in	<i>edge_conc_vals</i>	Lithium concentration at the sphere edge for all time steps
in	<i>i_app_data</i>	
in	<i>electrode</i>	
in	<i>tsteps</i>	
in	<i>pos_params</i>	
in	<i>neg_params</i>	

Definition at line 164 of file plotter.py.

6.6.2.4 gen_plots()

```
def plotter.gen_plots (
    filename,
    pos_params = None,
    neg_params = None,
    animation_interval_time = 10,
    SaveFinalState = True,
    SparsifyAnimation = False )
```

Definition at line 374 of file plotter.py.

6.6.2.5 get_avg_li_conc()

```
def plotter.get_avg_li_conc (
    c_vals,
    r_vals )
```

Gets the average concentration of lithium at a given timestep using trapezium rule inside the sphere.

!

Uses the fact that the average concentration of lithium in a sphere at any timestep can be found from $\frac{3}{R^3} \int_0^R c(r) r^2 dr$. Note, that this seems weird. It is weird. The reason this is necessary is because the concentration values in the sphere are defined per active electrode volume, i.e $e_{act} * L * A$, not per sphere volume. Therefore, to calculate the lithium mass loss in the whole system it is necessary to first find the concentration as if each sphere was uniformly filled (that is what this function does), and then after that multiply this average concentration by the active electrode volume. Take it up with whoever came up with the single particle model, not me. !

Parameters

in	<i>c_vals</i>	Array of concentration values covering the whole sphere radius, floats. !
in	<i>r_vals</i>	Array of radius values that correspond the concentration values given in <i>c_vals</i> , floats.

Definition at line 515 of file plotter.py.

6.6.2.6 plot_halfcell_GITT_result()

```
def plotter.plot_halfcell_GITT_result (
    filename,
    start_times,
    electrode,
    pos_params = None,
    neg_params = None,
    Animation = False,
    SaveFinalState = False,
    SparsifyAnimation = True,
    animation_interval_time = 10 )
```

Definition at line 345 of file plotter.py.

6.6.2.7 read_input_current()

```
def plotter.read_input_current (
    filename,
    step_num = None )
```

Reads in the applied current.

Parameters

in	<i>filename</i>	The name of the input file, this must be a string and have max 50 characters. No file extension is required.
in	<i>step_num</i>	Specifies the step number to read for a parallel simulation (default = none for serial)

Definition at line 55 of file plotter.py.

6.6.2.8 read_output_file()

```
def plotter.read_output_file (
    filename,
    step_num = None )
```

Reads in data from user input and solver output using NetCDF.

Parameters

in	<i>filename</i>	The name of the input file, this must be a string and have max 50 characters. No file extension is required.
in	<i>step_num</i>	Specifies the step number to read for a parallel simulation (default = none for serial)

Definition at line 14 of file plotter.py.

6.6.2.9 voltage_current_plot()

```
def plotter.voltage_current_plot (
    electrode,
    cstore,
    time_axis,
    i_app_data,
    tsteps,
    pos_params = None,
    neg_params = None )
```

Calculates voltages and plots as a function of time.

Calculates the voltage at all time points and saves plots of both Li concentration at the sphere edge and voltage as a function of time. This function has different settings determined by the value of electrode.

Parameters

in	<i>electrode</i>	
in	<i>cstore</i>	
in	<i>time_axis</i>	
in	<i>i_app_data</i>	
in	<i>tsteps</i>	
in	<i>pos_params</i>	
in	<i>neg_params</i>	

Definition at line 315 of file `plotter.py`.

6.7 `read_inputs` Module Reference

Module to read user inputs.

6.7.1 Detailed Description

Module to read user inputs.

Module contains functions that read user inputs from file and command line.

6.8 `sensitivity_analysis` Namespace Reference

First order sensitivity analysis.

Functions

- def `sensitivity_over_time` ()
- def `update` (frame)

Variables

- def `scaled_sensitivities_matrix` = `sensitivity_over_time`()
- `fig`
- `ax`
- list `x_labels` = ['c0', 'D', 'R', 'a', 'L']
- `ani` = `FuncAnimation`(`fig`, `update`, frames=len(`scaled_sensitivities_matrix`), blit=True)
- `filename`
- `writer`
- `fps`

6.8.1 Detailed Description

First order sensitivity analysis.

A python file that is used to produce first order sensitivity analysis Running the file outputs a sensitivity gif.

Credit to Dr. James Kermode for some functions recycled from PX914.

6.8.2 Function Documentation

6.8.2.1 sensitivity_over_time()

```
def sensitivity_analysis.sensitivity_over_time ( )
```

Definition at line 28 of file sensitivity_analysis.py.

6.8.2.2 update()

```
def sensitivity_analysis.update (
    frame )
```

Definition at line 288 of file sensitivity_analysis.py.

6.8.3 Variable Documentation

6.8.3.1 ani

```
ani = FuncAnimation(fig, update, frames=len(scaled_sensitivities_matrix), blit=True)
```

Definition at line 317 of file sensitivity_analysis.py.

6.8.3.2 ax

```
ax
```

Definition at line 261 of file sensitivity_analysis.py.

6.8.3.3 fig

```
fig
```

Definition at line 261 of file sensitivity_analysis.py.

6.8.3.4 filename

```
filename
```

Definition at line 318 of file sensitivity_analysis.py.

6.8.3.5 `fps`

`fps`

Definition at line 318 of file `sensitivity_analysis.py`.

6.8.3.6 `scaled_sensitivities_matrix`

```
def scaled_sensitivities_matrix = sensitivity_over_time()
```

Definition at line 255 of file `sensitivity_analysis.py`.

6.8.3.7 `writer`

`writer`

Definition at line 318 of file `sensitivity_analysis.py`.

6.8.3.8 `x_labels`

```
list x_labels = ['c0', 'D', 'R', 'a', 'L']
```

Definition at line 265 of file `sensitivity_analysis.py`.

6.9 `Uncertainty_Propagation` Namespace Reference

Performs uncertainty propagation.

Functions

- def `in_out_easy_peasy` (`parameter_np_array`)

Variables

- `iapp` = `np.concatenate((np.array([20.0 for i in range(50)]), np.array([0.0 for i in range(50)])))`
*Read in applied current density from csv file iapp_filename = 'WLTP_m10.csv' iapp, iapp_label, tsteps = UI.iapp ↵
read_csv(iapp_filename)*
- `parameters` = `parameter_np_array`
END MEAN SET VALUES #####.
- `int nprocs` = 40
Call the function to perform the full battery parallel solve.
- `output_filename_positive`
- `output_filename_negative`
- `cstore_pos`
Obtaining QOI (Voltage) #.
- `tsteps`
- `nodenum`
- `R_pos`
- `time_axis`
- `dr_pos`
- `electrode_pos`
- `cstore_neg`
- `R_neg`
- `dr_neg`
- `electrode_neg`
- `i_app_data_pos` = `plotter.read_input_current(output_filename_positive)`
- `i_app_data_neg` = `plotter.read_input_current(output_filename_negative)`
- `edge_conc_vals_pos` = `cstore_pos[:, -1]`
- `edge_conc_vals_neg` = `cstore_neg[:, -1]`
- `list posparams` = `[K_pos, parameters[3], cmax_pos_sim, parameters[4]]`
- `list negparams` = `[K_neg, a_neg, cmax_neg_sim, L_neg]`
- `pos_voltage` = `plotter.gen_half_cell_voltage(edge_conc_vals_pos, i_app_data_pos, electrode_pos, tsteps, pos ↵
_params=posparams)`
- `neg_voltage` = `plotter.gen_half_cell_voltage(edge_conc_vals_neg, i_app_data_neg, electrode_neg, tsteps, neg ↵
_params=negparams)`
- `full_voltage` = `pos_voltage - neg_voltage`
- `c0_dist` = `st.norm(1000.0, 25.0)`
Define the input parameter distributions #.
- `D_dist` = `st.norm(4.0e-15, 0.5e-15)`
- `R_dist` = `st.norm(5.86e-6, 1.0e-6)`
- `a_dist` = `st.norm(3.0*0.665/5.86e-6, 1.0e5)`
- `L_dist` = `st.norm(75.6e-6, 5.0e-6)`
- `int num_samples` = 40
- `c0_samples` = `c0_dist.rvs(num_samples)`
- `D_samples` = `D_dist.rvs(num_samples)`
- `R_samples` = `R_dist.rvs(num_samples)`
- `a_samples` = `a_dist.rvs(num_samples)`
- `L_samples` = `L_dist.rvs(num_samples)`
- `fixed_c0` = `c0_dist.mean()`
- `fixed_D` = `D_dist.mean()`
- `fixed_R` = `R_dist.mean()`
- `fixed_a` = `a_dist.mean()`
- `fixed_L` = `L_dist.mean()`
- `int counter` = 0
- `dt`
- `time` = `np.linspace(0.0, dt*(tsteps-1.), tsteps)`

```

_, _, tsteps = UI.iapp_read_csv(iapp_filename)
• voltage_curves = np.zeros(tsteps)
• fig
    UQ Plot #.
• axs
• figsize
• sharex
• True
• sharey
• params = np.array([c0_samples[i], fixed_D, fixed_R, fixed_a, fixed_L])
• alpha
• figg
    Input Distributions #.
• axss
• False
• int resol = 1000
• x = np.linspace(900, 1100, resol)

```

6.9.1 Detailed Description

Performs uncertainty propagation.

Applies normal distributions to parameter inputs and produces a 5 subplot figure that shows the posterior distribution of the voltage curve. Each subplot shows the effect of altering one parameter and keeping the others constant at mean values of input distributions.

6.9.2 Function Documentation

6.9.2.1 in_out_easy_peasy()

```

def Uncertainty_Propagation.in_out_easy_peasy (
    parameter_np_array )

```

Definition at line 32 of file Uncertainty_Propagation.py.

6.9.3 Variable Documentation

6.9.3.1 a_dist

```
a_dist = st.norm(3.0*0.665/5.86e-6, 1.0e5)
```

Definition at line 109 of file Uncertainty_Propagation.py.

6.9.3.2 a_samples

```
a_samples = a_dist.rvs(num_samples)
```

Definition at line 121 of file Uncertainty_Propagation.py.

6.9.3.3 alpha

```
alpha
```

Definition at line 186 of file Uncertainty_Propagation.py.

6.9.3.4 axs

```
axs
```

Definition at line 178 of file Uncertainty_Propagation.py.

6.9.3.5 axss

```
axss
```

Definition at line 232 of file Uncertainty_Propagation.py.

6.9.3.6 c0_dist

```
c0_dist = st.norm(1000.0, 25.0)
```

Define the input parameter distributions #.

Definition at line 106 of file Uncertainty_Propagation.py.

6.9.3.7 c0_samples

```
c0_samples = c0_dist.rvs(num_samples)
```

Definition at line 118 of file Uncertainty_Propagation.py.

6.9.3.8 counter

```
int counter = 0
```

Definition at line 132 of file Uncertainty_Propagation.py.

6.9.3.9 cstore_neg

```
cstore_neg
```

Definition at line 86 of file Uncertainty_Propagation.py.

6.9.3.10 cstore_pos

```
cstore_pos
```

Obtaining QOI (Voltage) #.

Definition at line 85 of file Uncertainty_Propagation.py.

6.9.3.11 D_dist

```
D_dist = st.norm(4.0e-15, 0.5e-15)
```

Definition at line 107 of file Uncertainty_Propagation.py.

6.9.3.12 D_samples

```
D_samples = D_dist.rvs(num_samples)
```

Definition at line 119 of file Uncertainty_Propagation.py.

6.9.3.13 dr_neg

```
dr_neg
```

Definition at line 86 of file Uncertainty_Propagation.py.

6.9.3.14 dr_pos

dr_pos

Definition at line 85 of file Uncertainty_Propagation.py.

6.9.3.15 dt

dt

Definition at line 166 of file Uncertainty_Propagation.py.

6.9.3.16 edge_conc_vals_neg

```
edge_conc_vals_neg = cstore_neg[:, -1]
```

Definition at line 90 of file Uncertainty_Propagation.py.

6.9.3.17 edge_conc_vals_pos

```
edge_conc_vals_pos = cstore_pos[:, -1]
```

Definition at line 89 of file Uncertainty_Propagation.py.

6.9.3.18 electrode_neg

electrode_neg

Definition at line 86 of file Uncertainty_Propagation.py.

6.9.3.19 electrode_pos

electrode_pos

Definition at line 85 of file Uncertainty_Propagation.py.

6.9.3.20 False

`False`

Definition at line 232 of file `Uncertainty_Propagation.py`.

6.9.3.21 fig

`fig`

UQ Plot #.

Definition at line 178 of file `Uncertainty_Propagation.py`.

6.9.3.22 figg

`figg`

Input Distributions #.

Definition at line 232 of file `Uncertainty_Propagation.py`.

6.9.3.23 figsize

`figsize`

Definition at line 178 of file `Uncertainty_Propagation.py`.

6.9.3.24 fixed_a

`fixed_a = a_dist.mean()`

Definition at line 128 of file `Uncertainty_Propagation.py`.

6.9.3.25 fixed_c0

`fixed_c0 = c0_dist.mean()`

Definition at line 125 of file `Uncertainty_Propagation.py`.

6.9.3.26 fixed_D

```
fixed_D = D_dist.mean()
```

Definition at line 126 of file Uncertainty_Propagation.py.

6.9.3.27 fixed_L

```
fixed_L = L_dist.mean()
```

Definition at line 129 of file Uncertainty_Propagation.py.

6.9.3.28 fixed_R

```
fixed_R = R_dist.mean()
```

Definition at line 127 of file Uncertainty_Propagation.py.

6.9.3.29 full_voltage

```
full_voltage = pos_voltage-neg_voltage
```

Definition at line 98 of file Uncertainty_Propagation.py.

6.9.3.30 i_app_data_neg

```
i_app_data_neg = plotter.read_input_current(output_filename_negative)
```

Definition at line 88 of file Uncertainty_Propagation.py.

6.9.3.31 i_app_data_pos

```
i_app_data_pos = plotter.read_input_current(output_filename_positive)
```

Definition at line 87 of file Uncertainty_Propagation.py.

6.9.3.32 iapp

```
iapp = np.concatenate((np.array([20.0 for i in range(50)]), np.array([0.0 for i in range(50)])))
```

Read in applied current density from csv file `iapp_filename = 'WLTP_m10.csv'` `iapp`, `iapp_label`, `tsteps = UI.iapp_`↵
`read_csv(iapp_filename)`

Definition at line 53 of file `Uncertainty_Propagation.py`.

6.9.3.33 L_dist

```
L_dist = st.norm(75.6e-6, 5.0e-6)
```

Definition at line 110 of file `Uncertainty_Propagation.py`.

6.9.3.34 L_samples

```
L_samples = L_dist.rvs(num_samples)
```

Definition at line 122 of file `Uncertainty_Propagation.py`.

6.9.3.35 neg_voltage

```
neg_voltage = plotter.gen_half_cell_voltage(edge_conc_vals_neg, i_app_data_neg, electrode_neg, tsteps, neg_↵  
_params=negparams)
```

Definition at line 97 of file `Uncertainty_Propagation.py`.

6.9.3.36 negparams

```
list negparams = [K_neg, a_neg, cmax_neg_sim, L_neg]
```

Definition at line 94 of file `Uncertainty_Propagation.py`.

6.9.3.37 nodenum

```
nodenum
```

Definition at line 85 of file `Uncertainty_Propagation.py`.

6.9.3.38 nprocs

```
nprocs = 40
```

Call the function to perform the full battery parallel solve.

Definition at line 76 of file Uncertainty_Propagation.py.

6.9.3.39 num_samples

```
int num_samples = 40
```

Definition at line 115 of file Uncertainty_Propagation.py.

6.9.3.40 output_filename_negative

```
output_filename_negative
```

Definition at line 78 of file Uncertainty_Propagation.py.

6.9.3.41 output_filename_positive

```
output_filename_positive
```

Definition at line 78 of file Uncertainty_Propagation.py.

6.9.3.42 parameters

```
parameters = parameter_np_array
```

```
END MEAN SET VALUES #####.
```

Definition at line 58 of file Uncertainty_Propagation.py.

6.9.3.43 params

```
params = np.array([c0_samples[i], fixed_D, fixed_R, fixed_a, fixed_L])
```

Definition at line 184 of file Uncertainty_Propagation.py.

6.9.3.44 pos_voltage

```
pos_voltage = plotter.gen_half_cell_voltage(edge_conc_vals_pos,i_app_data_pos,electrode_pos,tsteps,pos←  
_params=posparams)
```

Definition at line 96 of file `Uncertainty_Propagation.py`.

6.9.3.45 posparams

```
list posparams = [K_pos,parameters[3],cmax_pos_sim,parameters[4]]
```

Definition at line 93 of file `Uncertainty_Propagation.py`.

6.9.3.46 R_dist

```
R_dist = st.norm(5.86e-6, 1.0e-6)
```

Definition at line 108 of file `Uncertainty_Propagation.py`.

6.9.3.47 R_neg

```
R_neg
```

Definition at line 86 of file `Uncertainty_Propagation.py`.

6.9.3.48 R_pos

```
R_pos
```

Definition at line 85 of file `Uncertainty_Propagation.py`.

6.9.3.49 R_samples

```
R_samples = R_dist.rvs(num_samples)
```

Definition at line 120 of file `Uncertainty_Propagation.py`.

6.9.3.50 resol

```
int resol = 1000
```

Definition at line 233 of file Uncertainty_Propagation.py.

6.9.3.51 sharex

```
sharex
```

Definition at line 178 of file Uncertainty_Propagation.py.

6.9.3.52 sharey

```
sharey
```

Definition at line 178 of file Uncertainty_Propagation.py.

6.9.3.53 time

```
time = np.linspace(0.0, dt*(tsteps-1.), tsteps)
```

```
_, _, tsteps = UI.iapp_read_csv(iapp_filename)
```

Definition at line 168 of file Uncertainty_Propagation.py.

6.9.3.54 time_axis

```
time_axis
```

Definition at line 85 of file Uncertainty_Propagation.py.

6.9.3.55 True

```
True
```

Definition at line 178 of file Uncertainty_Propagation.py.

6.9.3.56 tsteps

tsteps

Definition at line 85 of file Uncertainty_Propagation.py.

6.9.3.57 voltage_curves

```
def voltage_curves = np.zeros(tsteps)
```

Definition at line 171 of file Uncertainty_Propagation.py.

6.9.3.58 x

```
x = np.linspace(900, 1100, resol)
```

Definition at line 236 of file Uncertainty_Propagation.py.

6.10 unit_test Namespace Reference

Functions

- def [run_unit_test](#) (nodenum=500, dt=1.0, num_cores=1)

6.10.1 Function Documentation

6.10.1.1 run_unit_test()

```
def unit_test.run_unit_test (
    nodenum = 500,
    dt = 1.0,
    num_cores = 1 )
```

Definition at line 7 of file unit_test.py.

6.11 user_input Namespace Reference

Sets up the user inputs and executes the solver.

Variables

- bool `checkpoint` = False
- string `solver_input_filename` = 'user_input'
- `tsteps`

```
SET VALUES #####.
```
- `dt`
- `n`
- `c0`
- `D`
- `R`
- `a`
- `L`
- `iapp`
- `iapp_label`
- `electrode_charge`
- float `K_pos` = 3.42E-6
- float `K_neg` = 6.48E-7
- float `cmax_pos_sim` = 63104.00
- float `cmax_neg_sim` = 33133.00
- list `plot_params_pos` = [`K_pos`,`a`,`cmax_pos_sim`,`L`]

```
END SET VALUES #####.
```
- `pos_params`

6.11.1 Detailed Description

Sets up the user inputs and executes the solver.

This file contains all input parameters for the solver and can be changed by the user. This file and all tuneable parameters are broken down in full detail in the user tutorial (see Tutorial.ipynb).

The following options can be controlled by editing this file:

- Outputting stdout to a file.
- Changing the name of this user input file.
- Set the values for the following input parameters:
 - `tsteps`
 - `dt`
 - `n`
 - `c0`
 - `D`
 - `R`
 - `a`
 - `L`
 - `iapp`
 - `iapp_label`
- These parameters can either take default values or be entered manually.

- For iapp there are additional options for setting constant or stepwise current and reading in values from a csv file.

The user does not need to edit anything past #END SET INPUT PARAMETER VALUES

The rest of the file calls various functions to:

- Check that the filename and parameters are valid
- Write the parameters to the file
- Call the solver and plotter.

6.11.2 Variable Documentation

6.11.2.1 a

a

Definition at line 48 of file user_input.py.

6.11.2.2 c0

c0

Definition at line 48 of file user_input.py.

6.11.2.3 checkpoint

```
bool checkpoint = False
```

Definition at line 40 of file user_input.py.

6.11.2.4 cmax_neg_sim

```
float cmax_neg_sim = 33133.00
```

Definition at line 56 of file user_input.py.

6.11.2.5 cmax_pos_sim

```
float cmax_pos_sim = 63104.00
```

Definition at line 55 of file user_input.py.

6.11.2.6 D

```
D
```

Definition at line 48 of file user_input.py.

6.11.2.7 dt

```
dt
```

Definition at line 48 of file user_input.py.

6.11.2.8 electrode_charge

```
electrode_charge
```

Definition at line 48 of file user_input.py.

6.11.2.9 iapp

```
iapp
```

Definition at line 48 of file user_input.py.

6.11.2.10 iapp_label

```
iapp_label
```

Definition at line 48 of file user_input.py.

6.11.2.11 K_neg

```
float K_neg = 6.48E-7
```

Definition at line 53 of file user_input.py.

6.11.2.12 K_pos

```
float K_pos = 3.42E-6
```

Definition at line 52 of file user_input.py.

6.11.2.13 L

```
L
```

Definition at line 48 of file user_input.py.

6.11.2.14 n

```
n
```

Definition at line 48 of file user_input.py.

6.11.2.15 plot_params_pos

```
list plot_params_pos = [K_pos, a, cmax_pos_sim, L]
```

```
END SET VALUES #####.
```

Definition at line 81 of file user_input.py.

6.11.2.16 pos_params

```
pos_params
```

Definition at line 84 of file user_input.py.

6.11.2.17 R

R

Definition at line 48 of file user_input.py.

6.11.2.18 solver_input_filename

```
solver_input_filename = 'user_input'
```

Definition at line 41 of file user_input.py.

6.11.2.19 tsteps

tsteps

SET VALUES #####.

Definition at line 48 of file user_input.py.

6.12 user_input_full_battery_GITT Namespace Reference**Variables**

- string `output_filename_positive` = 'current_step_pos'
Uncomment to print stdout to file `sys.stdout = open('test.txt', 'w')`
- string `output_filename_negative` = 'current_step_neg'
- `tsteps`
Import default values from <https://doi.org/10.1149/1945-7111/ab9050> Use `UI.set_defaults_pos()` for positive electrode and `..._neg()` for negative electrode.
- `dt` = 1.0
- `n` = 1000
- `c0`
- `D_neg`
- `R_neg`
- `a_neg`
- `L_neg`
- `iapp_neg`
- `iapp_label_neg`
- `electrode_charge_neg`
- `D_pos`
- `R_pos`
- `a_pos`
- `L_pos`
- `iapp_pos`
- `iapp_label_pos`

- [electrode_charge_pos](#)
- float [c0_pos](#) = 30000.0
Set values #####.
- float [c0_neg](#) = 1.0
- float [K_pos](#) = 3.42E-6
- float [K_neg](#) = 6.48E-7
- float [cmax_pos_sim](#) = 63104.00
- float [cmax_neg_sim](#) = 33133.00
- int [nprocs](#) = 40
Check parameters are valid #####.
- int [nsteps](#) = 10
- list [currents](#) = [20.0 for i in range([nsteps](#))]
- list [start_times](#) = [2150.0*i for i in range([nsteps](#))]
- list [run_times](#) = [150.0 for i in range([nsteps](#))]
- list [wait_times](#) = [2000.0 for i in range([nsteps](#))]
- list [params_pos](#) = [[dt](#), [c0_pos](#), [D_pos](#), [R_pos](#), [a_pos](#), [L_pos](#), [electrode_charge_pos](#)]
- list [params_neg](#) = [[dt](#), [c0_neg](#), [D_neg](#), [R_neg](#), [a_neg](#), [L_neg](#), [electrode_charge_neg](#)]
- list [plot_params_pos](#) = [[K_pos](#), [a_pos](#), [cmax_pos_sim](#), [L_pos](#)]
Call the function to perform the full battery parallel solve.
- list [plot_params_neg](#) = [[K_neg](#), [a_neg](#), [cmax_neg_sim](#), [L_neg](#)]
- [pos_params](#)
- [neg_params](#)
- [SparsifyAnimation](#)
- [True](#)
- [animation_interval_time](#)

6.12.1 Detailed Description

@brief Example script for a full battery model

@details This is a running script which exists to demonstrate how one uses the GITT function to run simulations in parallel on a full battery. This simulation completes 10 current cycles on each electrode over 2 cores each - parallelising perfectly over 40 cores.

These simulations consist of applying a step function of current to a battery which turns on to a fixed value for a given time interval, then back to 0, and then on again, etc.

The ascii art below indicates schematically how a current-time step looks during this experiment.

```

I
|_____|_____|_____
|       |       |       |
|       |       |       |
|       |       |       |
|       |       |       |
|       |       |       |
|_____|_____|_____t

```

To run one of these simulations, one needs the following things:

- A vector containing the fixed values of current applied for each block of current
- A vector containing the start times of each current block
- A vector containing the duration of each current block

6.12.2 Variable Documentation

6.12.2.1 a_neg

a_neg

Definition at line 43 of file user_input_full_battery_GITT.py.

6.12.2.2 a_pos

a_pos

Definition at line 44 of file user_input_full_battery_GITT.py.

6.12.2.3 animation_interval_time

animation_interval_time

Definition at line 89 of file user_input_full_battery_GITT.py.

6.12.2.4 c0

c0

Definition at line 43 of file user_input_full_battery_GITT.py.

6.12.2.5 c0_neg

```
float c0_neg = 1.0
```

Definition at line 48 of file user_input_full_battery_GITT.py.

6.12.2.6 c0_pos

```
float c0_pos = 30000.0
```

Set values #####.

Definition at line 47 of file user_input_full_battery_GITT.py.

6.12.2.7 `cmax_neg_sim`

```
float cmax_neg_sim = 33133.00
```

Definition at line 56 of file `user_input_full_battery_GITT.py`.

6.12.2.8 `cmax_pos_sim`

```
float cmax_pos_sim = 63104.00
```

Definition at line 55 of file `user_input_full_battery_GITT.py`.

6.12.2.9 `currents`

```
list currents = [20.0 for i in range(nsteps)]
```

Definition at line 72 of file `user_input_full_battery_GITT.py`.

6.12.2.10 `D_neg`

```
D_neg
```

Definition at line 43 of file `user_input_full_battery_GITT.py`.

6.12.2.11 `D_pos`

```
D_pos
```

Definition at line 44 of file `user_input_full_battery_GITT.py`.

6.12.2.12 `dt`

```
float dt = 1.0
```

Definition at line 43 of file `user_input_full_battery_GITT.py`.

6.12.2.13 electrode_charge_neg

electrode_charge_neg

Definition at line 43 of file user_input_full_battery_GITT.py.

6.12.2.14 electrode_charge_pos

electrode_charge_pos

Definition at line 44 of file user_input_full_battery_GITT.py.

6.12.2.15 iapp_label_neg

iapp_label_neg

Definition at line 43 of file user_input_full_battery_GITT.py.

6.12.2.16 iapp_label_pos

iapp_label_pos

Definition at line 44 of file user_input_full_battery_GITT.py.

6.12.2.17 iapp_neg

iapp_neg

Definition at line 43 of file user_input_full_battery_GITT.py.

6.12.2.18 iapp_pos

iapp_pos

Definition at line 44 of file user_input_full_battery_GITT.py.

6.12.2.19 K_neg

```
float K_neg = 6.48E-7
```

Definition at line 53 of file user_input_full_battery_GITT.py.

6.12.2.20 K_pos

```
float K_pos = 3.42E-6
```

Definition at line 52 of file user_input_full_battery_GITT.py.

6.12.2.21 L_neg

```
L_neg
```

Definition at line 43 of file user_input_full_battery_GITT.py.

6.12.2.22 L_pos

```
L_pos
```

Definition at line 44 of file user_input_full_battery_GITT.py.

6.12.2.23 n

```
int n = 1000
```

Definition at line 43 of file user_input_full_battery_GITT.py.

6.12.2.24 neg_params

```
neg_params
```

Definition at line 89 of file user_input_full_battery_GITT.py.

6.12.2.25 nprocs

```
int nprocs = 40
```

Check parameters are valid #####.

Manually set up applied current and parallelisation ##### Number of parallel processors being utilised

Definition at line 66 of file user_input_full_battery_GITT.py.

6.12.2.26 nsteps

```
int nsteps = 10
```

Definition at line 69 of file user_input_full_battery_GITT.py.

6.12.2.27 output_filename_negative

```
output_filename_negative = 'current_step_neg'
```

Definition at line 39 of file user_input_full_battery_GITT.py.

6.12.2.28 output_filename_positive

```
output_filename_positive = 'current_step_pos'
```

Uncomment to print stdout to file `sys.stdout = open('test.txt', 'w')`

Call the plotting function which plots the results of the GITT test with nstep steps.

Set filename to output user input parameters ##### Set string for filename. Do not enter a file extension. Max characters = 50

Definition at line 38 of file user_input_full_battery_GITT.py.

6.12.2.29 params_neg

```
list params_neg = [dt, c0_neg, D_neg, R_neg, a_neg, L_neg, electrode_charge_neg]
```

Definition at line 79 of file user_input_full_battery_GITT.py.

6.12.2.30 params_pos

```
list params_pos = [dt, c0_pos, D_pos, R_pos, a_pos, L_pos, electrode_charge_pos]
```

Definition at line 78 of file user_input_full_battery_GITT.py.

6.12.2.31 plot_params_neg

```
plot_params_neg = [K_neg, a_neg, cmax_neg_sim, L_neg]
```

Definition at line 86 of file user_input_full_battery_GITT.py.

6.12.2.32 plot_params_pos

```
plot_params_pos = [K_pos, a_pos, cmax_pos_sim, L_pos]
```

Call the function to perform the full battery parallel solve.

Definition at line 85 of file user_input_full_battery_GITT.py.

6.12.2.33 pos_params

```
pos_params
```

Definition at line 89 of file user_input_full_battery_GITT.py.

6.12.2.34 R_neg

```
R_neg
```

Definition at line 43 of file user_input_full_battery_GITT.py.

6.12.2.35 R_pos

```
R_pos
```

Definition at line 44 of file user_input_full_battery_GITT.py.

6.12.2.36 run_times

```
list run_times = [150.0 for i in range(nsteps)]
```

Definition at line 74 of file user_input_full_battery_GITT.py.

6.12.2.37 SparsifyAnimation

```
SparsifyAnimation
```

Definition at line 89 of file user_input_full_battery_GITT.py.

6.12.2.38 start_times

```
start_times = [2150.0*i for i in range(nsteps)]
```

Definition at line 73 of file user_input_full_battery_GITT.py.

6.12.2.39 True

```
True
```

Definition at line 89 of file user_input_full_battery_GITT.py.

6.12.2.40 tsteps

```
tsteps
```

Import default values from <https://doi.org/10.1149/1945-7111/ab9050> Use `Ul.set_defaults_pos()` for positive electrode and `..._neg()` for negative electrode.

Definition at line 43 of file user_input_full_battery_GITT.py.

6.12.2.41 wait_times

```
list wait_times = [2000.0 for i in range(nsteps)]
```

Definition at line 75 of file user_input_full_battery_GITT.py.

6.13 user_input_mod Namespace Reference

Package containing the functions needed to set up the user input parameters and execute the solver.

Functions

- def `iapp_read_csv` (filename)
CURRENT DENSITY SET UP ###.
- def `iapp_constant_setup` (tsteps, iapp)
Sets up the applied current density 'iapp' as a constant valued array of length tsteps.
- def `iapp_step_setup` (tsteps, iapp_steps)
Sets up the applied current density 'iapp' as a stepped array of length tsteps.
- def `set_defaults_pos` ()
DEFAULT PARAMETERS ###.
- def `set_defaults_neg` ()
Returns default parameters for a negative electrode.
- def `verify_params` (filename, tsteps, dt, n, c0, D, R, a, L, electrode_charge)
PARAMETER VERIFICATION ###.
- def `verify_iapp` (iapp, iapp_label, tsteps)
Verifies the types and values of the applied current density array and its label.
- def `write_to_file` (filename, tsteps, dt, n, c0, D, R, a, L, iapp, iapp_label, electrode_charge)
WRITE TO FILE ###.
- def `call_solver` (filename, checkpoint, nprocs=1)
CALL SOLVER ###.
- def `get_GITT_initial_concs` (currents, run_times, c0, R, a, L, electrode_charge)
- def `GITT_half_cell` (filename, nprocs, currents, start_times, run_times, wait_times, n, params)
INITIALISE A FULL GITT TEST IN PARALLEL ###.
- def `GITT_full_cell` (filename_positive, filename_negative, nprocs, currents, start_times, run_times, wait_↵
times, n, params_pos, params_neg)
Executes the SPM solver in parallel to run a GITT full cell test over nprocs cores.
- def `full_battery_simulation` (filename_positive, filename_negative, nprocs=1)
Executes the SPM solver.
- def `optimise_parallelism` (n_procs, n_gitt_steps=1, full_battery=False)
Decides the correct number of threads to use for a given simulation.

6.13.1 Detailed Description

Package containing the functions needed to set up the user input parameters and execute the solver.

Provides options for automatically generating applied current density 'iapp', as a constant or step function, and for reading iapp from a csv file.

Provides the option to set the parameters to default values obtained from Chen et al. 2020, <https://doi.org/10.1149/1945-7111/ab9050>.

Contains additional functions for validating the types and values of the input parameters, writing the user input txt file, and executing the solver.

6.13.2 Function Documentation

6.13.2.1 call_solver()

```
def user_input_mod.call_solver (
    filename,
    checkpoint,
    nprocs = 1 )
```

CALL SOLVER ###.

Executes the SPM solver.

@details Calls the SPM solver using the subprocess package.
The filename of the user input file is passed to the solver as a command line argument.
Errors from the execution are read in and further execution prevented if necessary.
@param[in] filename: The name of the file containing the desired input, either a checkpoint file or user input
@param[in] checkpoint: Boolean indicating if a checkpoint file is used.

The function works by:

1. Validating whether the input file is a checkpoint file or user input file name.
2. Setting up the solver call line, including the file name.
3. Calling the solver.

Definition at line 413 of file user_input_mod.py.

6.13.2.2 full_battery_simulation()

```
def user_input_mod.full_battery_simulation (
    filename_positive,
    filename_negative,
    nprocs = 1 )
```

Executes the SPM solver.

@details Calls the SPM solver using the subprocess package for both the anode and cathode simultaneously
The filenames of the user input files for both the anode and cathode are passed to
instances of the solver as command line arguments. Note that if 2 processors are supplied,
both the anode and cathode will run simultaneously.
@param[in] filename_positive: The name of the positive output file, this must be a string and have max 50 char
@param[in] filename_negative: The name of the negative output file, this must be a string and have max 50 char

The function works by:

1. Setting up the solver call line, including the file name.
2. Calling the solver.

Definition at line 651 of file user_input_mod.py.

6.13.2.3 get_GITT_initial_concs()

```
def user_input_mod.get_GITT_initial_concs (
    currents,
    run_times,
    c0,
    R,
    a,
    L,
    electrode_charge )
```

@brief Calculates the initial concentrations for each step in a multi-step parallelised GITT test.

@details Computes initial flat concentrations for the initial step of a parallelised multi-step GITT test.

The formula used for this calculation is:

$C(T=t) = C_0 + \frac{i_{app} * t}{F * e_{act} * L}$, where C_0 is $C(T=0)$.

As current is defined as positive for charging, the sign of the current vector must be flipped when considering the positive electrode (cathode) which will discharge during charging.

@param[in] currents A vector containing current values to apply at each current step, must be floats and have start_times, run_times, wait_times.

@param[in] run_times A vector containing the run time of each current step, must be floats and have the same start_times, currents, wait_times.

@param[in] c0 Initial concentration, must be a float greater than or equal to 0.

@param[in] R Radius of the sphere, must be a float greater than 0.

@param[in] a Particle surface area per unit volume, must be a float greater than or equal to 0.

@param[in] L Electrode thickness, must be a float greater than or equal to 0.

@param[in] electrode_charge Electrode charge, must be a single character 'p' or 'n'.

Definition at line 475 of file user_input_mod.py.

6.13.2.4 GITT_full_cell()

```
def user_input_mod.GITT_full_cell (
    filename_positive,
    filename_negative,
    nprocs,
    currents,
    start_times,
    run_times,
    wait_times,
    n,
    params_pos,
    params_neg )
```

Executes the SPM solver in parallel to run a GITT full cell test over nprocs cores.

@details Executes the SPM solver in parallel to run a test experiment on a full cell using a galvanostatic intermittent titration technique (GITT) as described by W. Weppner and R. A. Huggins 1977 J. Electrochem. Soc. 124 1569.

Due to the equilibration time between each current step applied, it is possible to pre-compute initial concentrations of particles in the model by considering the amount of lithium removed in each step.

@param[in] filename_positive Name of user input file for the cathode, no file extension. Note that a version of the file is created for each current step applied in the GITT test.

@param[in] filename_negative Name of user input file for the anode, no file extension. Note that a version of the file is created for each current step applied in the GITT test.

each current step applied in the GITT test.

@param[in] nprocs Number of processors to parallelise current steps over.

Cannot parallelise over more processors than the number of steps applied in the GITT test.

@param[in] currents A vector containing the values of current to apply at each step, must be floats and have the same length as start_times, run_times, wait_times.

@param[in] start_times A vector containing the start times of each current step, must be floats and have the same length as currents, run_times, wait_times.

@param[in] run_times A vector containing the run time of each current step, must be floats and have the same length as start_times, currents, wait_times.

@param[in] wait_times A vector containing the wait time of each current step, must be floats and have the same length as currents, start_times, run-times.

@param[in], n: The number of nodes to use in the simulation. Must be an integer greater than 100.

@param[in] params: A vector containing the parameters for the simulation: [dt, c0, D, R, a, L]

The function works by:

1. Setting up the solver call line, including the file name.
2. Calling the solver.

Definition at line 576 of file user_input_mod.py.

6.13.2.5 GITT_half_cell()

```
def user_input_mod.GITT_half_cell (
    filename,
    nprocs,
    currents,
    start_times,
    run_times,
    wait_times,
    n,
    params )
```

INITIALISE A FULL GITT TEST IN PARALLEL ####.

Executes the SPM solver in parallel to run a GITT half cell test over nprocs cores.

@details Executes the SPM solver in parallel to run a test experiment on a half cell using a galvanostatic intermittent titration technique. See W. Weppner and R. A. Huggins 1977 J. Electrochem. Soc. 124 1569.

Due to the equilibration time between each current step applied, it is possible to pre-compute initial concentrations of particles in the model by considering the amount of lithium removed in each step.

Errors from execution are read in and further execution is prevented if necessary.

@param[in] filename Name of user input file, no file extension. Note that a version of this file is created for each step.

@param[in] nprocs Number of processors to parallelise current steps over.

Cannot parallelise over more processors than the number of steps applied in the GITT test.

@param[in] currents A vector containing the values of current to apply at each step, must be floats and have the same length as start_times, run_times, wait_times.

@param[in] start_times A vector containing the start times of each current step, must be floats and have the same length as currents, run_times, wait_times.

@param[in] run_times A vector containing the run time of each current step, must be floats and have the same length as start_times, currents, wait_times.

@param[in] wait_times A vector containing the wait time of each current step, must be floats and have the same length as currents, start_times, run-times.

@param[in], n: The number of nodes to use in the simulation. Must be an integer greater than 100.

@param[in] params: A vector containing the parameters for the simulation: [dt, c0, D, R, a, L, electrode_charge]

Definition at line 513 of file user_input_mod.py.

6.13.2.6 iapp_constant_setup()

```
def user_input_mod.iapp_constant_setup (
    tsteps,
    iapp )
```

Sets up the applied current density 'iapp' as a constant valued array of length tsteps.

```
@param[in] tsteps Number of timesteps, needs to be the length of the array iapp.
@param[in] iapp Constant value of iapp.
@result iapp_arr: An array containing applied current density.
@result iapp_label: String containing a description of iapp.
```

Function does the following:

1. Verifies that the current density, iapp, is a float.
2. Sets up an array.

Definition at line 52 of file user_input_mod.py.

6.13.2.7 iapp_read_csv()

```
def user_input_mod.iapp_read_csv (
    filename )
```

CURRENT DENSITY SET UP ###.

Reads in the applied current density 'iapp' from a csv file provided by user.

```
@param[in] filename Name of the csv file containing iapp as Time, Current density.
@result iapp_arr: Array containing applied current density.
@result iapp_label: String containing a description of iapp.
@result tsteps: Number of timesteps, found from the length of iapp array.
```

This function does the following:

1. Reads in the csv file.
2. Parses the file into array of lines.
3. Parses each line at comma.
4. Define tsteps and the iapp_label.

Definition at line 16 of file user_input_mod.py.

6.13.2.8 iapp_step_setup()

```
def user_input_mod.iapp_step_setup (
    tsteps,
    iapp_steps )
```

Sets up the applied current density 'iapp' as a stepped array of length tsteps.

```
@param[in] tsteps Number of timesteps, needs to be the length of the array iapp.
@param[in] iapp_steps 2D Array containing heights of the steps and the timesteps at which step occurs, starting from 0.
@result iapp_arr: An array containing the applied current density.
@result iapp_label: A string describing the type for iapp.
```

Function does the following:

1. Sets up the label and initialises the array.
2. Verifies that timesteps are integers and current density values are floats.
3. Verifies that the first timestep is 0.
4. Sets up an array.

Definition at line 76 of file user_input_mod.py.

6.13.2.9 optimise_parallelism()

```
def user_input_mod.optimise_parallelism (
    n_procs,
    n_gitt_steps = 1,
    full_battery = False )
```

Decides the correct number of threads to use for a given simulation.

@details Given a user supplied number of processors, as well as further details about the simulation (the number of current steps if it's a GITT test, if the simulation is a full cell or not), this function decides on the optimum number of threads to use. It never picks a number of threads greater than 4, as any larger than this and the communication overhead between threads during the matrix solve begins to dominate and the computation time increases. This function therefore allows a balanced optimum parallelisation strategy.

@param[in] n_procs: The number of processors that the user is happy for the program to exploit. The program will use a number of cores less than or equal to this number. Integer.

@param[in] n_gitt_steps: The number of GITT steps if the experiment being performed is a GITT test. 1 by default if the experiment not being a GITT test) integer.

@param[in] full_battery: Bool, True means that both sides of a battery are being simulated, whilst False means only one side.

The function works by:

1. Deciding what number of threads is best to use given the type of simulation and input parameters, up to a maximum of 4. Note that this is based off the principle that parallelisation over independent separate instances of the simulation is faster than multithreading.
2. Setting the environment variable.

Definition at line 686 of file user_input_mod.py.

6.13.2.10 set_defaults_neg()

```
def user_input_mod.set_defaults_neg ( )
```

Returns default parameters for a negative electrode.

Gives the default parameter values, obtained from Chen et al. 2020, <https://doi.org/10.1016/j.jes.2020.114919>. The simulation is set up to run for 100 timesteps of size $dt = 0.1s$, with $n = 1000$ spatial nodes. Applied current density is set up as a constant current density of value $0.73mA\ m^{-2}$.

Definition at line 174 of file user_input_mod.py.

6.13.2.11 set_defaults_pos()

```
def user_input_mod.set_defaults_pos ( )
```

DEFAULT PARAMETERS ###.

Returns default parameters for a positive electrode

Gives the default parameter values, obtained from Chen et al. 2020, <https://doi.org/10.1016/j.jes.2020.114919>. The simulation is set up to run for 100 timesteps of size $dt = 0.1s$, with $n = 1000$ spatial nodes. Applied current density is set up as a constant current density of value $0.73mA\ m^{-2}$.

Definition at line 139 of file user_input_mod.py.

6.13.2.12 verify_iapp()

```
def user_input_mod.verify_iapp (
    iapp,
    iapp_label,
    tsteps )
```

Verifies the types and values of the applied current density array and its label.

@details Verifies that the parameters output iapp and iapp_label have the correct types, valid values, and correct length. If any are found to be invalid, an error is printed and the execution will stop.
 @param[in] iapp: Applied current density, 1D array of floats of length tsteps.
 @param[in] iapp_label: Labels the type of applied current density, must be a string.
 @param[in] tsteps: Number of timesteps, in order to check iapp has the correct length.

Definition at line 323 of file user_input_mod.py.

6.13.2.13 verify_params()

```
def user_input_mod.verify_params (
    filename,
    tsteps,
    dt,
    n,
    c0,
    D,
    R,
    a,
    L,
    electrode_charge )
```

PARAMETER VERIFICATION ###.

Verifies the types and values of the input parameters.

@details Verifies that output filename, tsteps, dt, c0, D, R, a, and L have the correct type and valid values. If any are found to be invalid, an error is printed and the execution will stop.

@param[in] filename The name of the output file, this must be a string and have max 50 characters. No file extension.
 @param[in] tsteps Number of timesteps, must be an integer greater than 0.
 @param[in] dt Timestep size, must be a float greater than 0.
 @param[in] n Number of spatial nodes, must be an integer between 100 and 4000.
 @param[in] c0 Initial concentration, must be a float greater than or equal to 0.
 @param[in] D Diffusion constant, must be a float.
 @param[in] R Radius of the sphere, must be a float greater than 0.
 @param[in] a Particle surface area per unit volume, must be a float greater than or equal to 0.
 @param[in] L Electrode thickness, must be a float greater than or equal to 0.
 @param[in] electrode_charge Labels charge of the electrode, must be a string, 'p' for positive or 'n' for negative.

Definition at line 213 of file user_input_mod.py.

6.13.2.14 write_to_file()

```
def user_input_mod.write_to_file (
    filename,
    tsteps,
    dt,
    n,
    c0,
    D,
    R,
    a,
    L,
    iapp,
    iapp_label,
    electrode_charge )
```

WRITE TO FILE ###.

Writes user inputs to a txt file.

@details Writes the user inputs to a txt file named 'filename'.
Parameters are output in the format 'parameter = value'.
tsteps, dt, c0, D, R, a, L, electrode_charge are output to the top of the file, followed by an asterix line (*).
iapp_label is output below the asterix line, with iapp array the following it, written one element per line.

The function works by:
1. Setting the file name.
2. Making a list of strings to write to the file.
3. Writing to the file.

Definition at line 359 of file user_input_mod.py.

6.14 user_input_parallel_simulation Namespace Reference

a running script which exists to demonstrate how one uses the GITT function to run simulations in parallel.

Variables

- string `output_filename` = 'current_step'
- string `electrode` = 'positive'
- `tsteps`
- `dt` = 1.0
- `n`
- `c0` = 30000.0
- `D`
- `R`
- `a`
- `L`
- `iapp`
- `iapp_label`
- `electrode_charge`
- float `K_pos` = 3.42E-6
- float `K_neg` = 6.48E-7
- float `cmax_pos_sim` = 63104.00

- float `cmax_neg_sim` = 33133.00
- int `nprocs` = 5
- int `nsteps` = 5
- list `currents` = [20.0 for `i` in range(`nsteps`)]
- list `start_times` = [2150.0*`i` for `i` in range(`nsteps`)]
- list `run_times` = [150.0 for `i` in range(`nsteps`)]
- list `wait_times` = [2000.0 for `i` in range(`nsteps`)]
- list `params` = [`dt`, `c0`, `D`, `R`, `a`, `L`, `electrode_charge`]
- list `plot_params_neg` = [`K_neg`, `a`, `cmax_neg_sim`, `L`]
- `neg_params`
- `Animation`
- `True`
- `SparsifyAnimation`

6.14.1 Detailed Description

a running script which exists to demonstrate how one uses the GITT function to run simulations in parallel.

6.14.2 Variable Documentation

6.14.2.1 `a`

`a`

Definition at line 61 of file `user_input_parallel_simulation.py`.

6.14.2.2 `Animation`

`Animation`

Definition at line 101 of file `user_input_parallel_simulation.py`.

6.14.2.3 `c0`

`float c0 = 30000.0`

Definition at line 61 of file `user_input_parallel_simulation.py`.

6.14.2.4 cmax_neg_sim

```
float cmax_neg_sim = 33133.00
```

Definition at line 72 of file user_input_parallel_simulation.py.

6.14.2.5 cmax_pos_sim

```
float cmax_pos_sim = 63104.00
```

Definition at line 71 of file user_input_parallel_simulation.py.

6.14.2.6 currents

```
list currents = [20.0 for i in range(nsteps)]
```

Definition at line 85 of file user_input_parallel_simulation.py.

6.14.2.7 D

```
D
```

Definition at line 61 of file user_input_parallel_simulation.py.

6.14.2.8 dt

```
float dt = 1.0
```

Definition at line 61 of file user_input_parallel_simulation.py.

6.14.2.9 electrode

```
electrode = 'positive'
```

Definition at line 37 of file user_input_parallel_simulation.py.

6.14.2.10 electrode_charge

electrode_charge

Definition at line 61 of file user_input_parallel_simulation.py.

6.14.2.11 iapp

iapp

Definition at line 61 of file user_input_parallel_simulation.py.

6.14.2.12 iapp_label

iapp_label

Definition at line 61 of file user_input_parallel_simulation.py.

6.14.2.13 K_neg

```
float K_neg = 6.48E-7
```

Definition at line 69 of file user_input_parallel_simulation.py.

6.14.2.14 K_pos

```
float K_pos = 3.42E-6
```

Definition at line 68 of file user_input_parallel_simulation.py.

6.14.2.15 L

L

Definition at line 61 of file user_input_parallel_simulation.py.

6.14.2.16 n

```
n
```

Definition at line 61 of file user_input_parallel_simulation.py.

6.14.2.17 neg_params

```
neg_params
```

Definition at line 101 of file user_input_parallel_simulation.py.

6.14.2.18 nprocs

```
int nprocs = 5
```

Definition at line 80 of file user_input_parallel_simulation.py.

6.14.2.19 nsteps

```
int nsteps = 5
```

Definition at line 82 of file user_input_parallel_simulation.py.

6.14.2.20 output_filename

```
output_filename = 'current_step'
```

Definition at line 36 of file user_input_parallel_simulation.py.

6.14.2.21 params

```
list params = [dt, c0, D, R, a, L,electrode_charge]
```

Definition at line 91 of file user_input_parallel_simulation.py.

6.14.2.22 plot_params_neg

```
plot_params_neg = [K_neg, a, cmax_neg_sim, L]
```

Definition at line 98 of file user_input_parallel_simulation.py.

6.14.2.23 R

```
R
```

Definition at line 61 of file user_input_parallel_simulation.py.

6.14.2.24 run_times

```
list run_times = [150.0 for i in range(nsteps)]
```

Definition at line 87 of file user_input_parallel_simulation.py.

6.14.2.25 SparsifyAnimation

```
SparsifyAnimation
```

Definition at line 101 of file user_input_parallel_simulation.py.

6.14.2.26 start_times

```
start_times = [2150.0*i for i in range(nsteps)]
```

Definition at line 86 of file user_input_parallel_simulation.py.

6.14.2.27 True

```
True
```

Definition at line 101 of file user_input_parallel_simulation.py.

6.14.2.28 tsteps

```
tsteps
```

Definition at line 61 of file user_input_parallel_simulation.py.

6.14.2.29 wait_times

```
list wait_times = [2000.0 for i in range(nsteps)]
```

Definition at line 88 of file user_input_parallel_simulation.py.

Chapter 7

File Documentation

7.1 /home/chem/msuvnj/PX915_GroupB_22-23/checkpointing.f90 File Reference

Modules

- module [checkpointing](#)
Module to read and write checkpoint files.

Variables

- integer, intent(in) [tstep](#)
- integer, intent(in) [tsteps](#)
- real(real64), intent(in) [dt](#)
- integer, intent(in) [n](#)
- real(real64), dimension([n](#)), intent(in) [c](#)
- real(real64), intent(in) [d](#)
- real(real64), intent(in) [r](#)
- real(real64), intent(in) [a_small](#)
- real(real64), intent(in) [l](#)
- real(real64), dimension([tsteps](#)), intent(in) [iapp](#)
- character(len= *), intent(in) [electrode_charge](#)
- real(real64), dimension([n](#), [tsteps](#)), intent(in) [cstorage](#)
- integer, intent(in), optional [freq_in](#)
- integer [freq](#)
- integer [freq_remainder](#)
- character(len= *), intent(inout) [filename](#)
- character(len=60) [dir](#)
- character(len=60) [checkpoint_name](#)
- character(len=120) [filepath](#)
- integer [ncid](#)
- integer [dimid_n](#)
- integer [dimid_tsteps](#)
- integer, dimension(1) [dimids_parameter](#)
- integer, dimension(2) [dimids_cstorage](#)
- integer [status](#)

- integer `varid_tstep`
- integer `varid_tsteps`
- integer `varid_dt`
- integer `varid_n`
- integer `varid_c`
- integer `varid_d`
- integer `varid_r`
- integer `varid_a`
- integer `varid_l`
- integer `varid_iapp`
- integer `varid_ec`
- integer `varid_cstorage`
- integer `filename_length`
- integer `file_ext`
- integer `file_test`
- character(len=10) `tstep_str`
- integer, intent(out) `tstep_init`
- integer `varid`
- character(len=5) `file_extension`
- integer `parse_idx`
- real(real64) `c0`

7.1.1 Variable Documentation

7.1.1.1 `a_small`

```
real(real64), intent(out) a_small
```

Definition at line 42 of file `checkpointing.f90`.

7.1.1.2 `c`

```
real(real64), dimension(:), intent(out), allocatable c
```

Definition at line 33 of file `checkpointing.f90`.

7.1.1.3 `c0`

```
real(real64) c0
```

Definition at line 298 of file `checkpointing.f90`.

7.1.1.4 checkpoint_name

```
character(len=60) checkpoint_name
```

Definition at line 66 of file checkpointing.f90.

7.1.1.5 cstorage

```
real(real64), dimension(:, :), intent(out), allocatable cstorage
```

Definition at line 54 of file checkpointing.f90.

7.1.1.6 d

```
real(real64), intent(out) d
```

Definition at line 36 of file checkpointing.f90.

7.1.1.7 dimid_n

```
integer dimid_n
```

Definition at line 68 of file checkpointing.f90.

7.1.1.8 dimid_tsteps

```
integer dimid_tsteps
```

Definition at line 68 of file checkpointing.f90.

7.1.1.9 dimids_cstorage

```
integer, dimension(2) dimids_cstorage
```

Definition at line 68 of file checkpointing.f90.

7.1.1.10 dimids_parameter

```
integer, dimension(1) dimids_parameter
```

Definition at line 68 of file checkpointing.f90.

7.1.1.11 dir

```
character(len=60) dir
```

Definition at line 66 of file checkpointing.f90.

7.1.1.12 dt

```
real(real64), intent(out) dt
```

Definition at line 27 of file checkpointing.f90.

7.1.1.13 electrode_charge

```
character(len=1), intent(out) electrode_charge
```

Definition at line 51 of file checkpointing.f90.

7.1.1.14 file_ext

```
integer file_ext
```

Definition at line 71 of file checkpointing.f90.

7.1.1.15 file_extension

```
character(len=5) file_extension
```

Definition at line 275 of file checkpointing.f90.

7.1.1.16 file_test

```
integer file_test
```

Definition at line 71 of file checkpointing.f90.

7.1.1.17 filename

```
character(len=*), intent(in) filename
```

Definition at line 65 of file checkpointing.f90.

7.1.1.18 filename_length

```
integer filename_length
```

Definition at line 71 of file checkpointing.f90.

7.1.1.19 filepath

```
character(len=120) filepath
```

Definition at line 67 of file checkpointing.f90.

7.1.1.20 freq

```
integer freq
```

Definition at line 60 of file checkpointing.f90.

7.1.1.21 freq_in

```
integer, intent(in), optional freq_in
```

Definition at line 59 of file checkpointing.f90.

7.1.1.22 freq_remainder

```
integer freq_remainder
```

Definition at line 61 of file checkpointing.f90.

7.1.1.23 iapp

```
real(real64), dimension(:), intent(out), allocatable iapp
```

Definition at line 48 of file checkpointing.f90.

7.1.1.24 l

```
real(real64), intent(out) l
```

Definition at line 45 of file checkpointing.f90.

7.1.1.25 n

```
integer, intent(out) n
```

Definition at line 30 of file checkpointing.f90.

7.1.1.26 ncid

```
integer ncid
```

Definition at line 68 of file checkpointing.f90.

7.1.1.27 parse_idx

```
integer parse_idx
```

Definition at line 278 of file checkpointing.f90.

7.1.1.28 r

```
real(real64), intent(out) r
```

Definition at line 39 of file checkpointing.f90.

7.1.1.29 status

```
integer status
```

Definition at line 68 of file checkpointing.f90.

7.1.1.30 tstep

```
integer, intent(in) tstep
```

Definition at line 21 of file checkpointing.f90.

7.1.1.31 tstep_init

```
integer, intent(out) tstep_init
```

Definition at line 166 of file checkpointing.f90.

7.1.1.32 tstep_str

```
character(len=10) tstep_str
```

Definition at line 72 of file checkpointing.f90.

7.1.1.33 tsteps

```
integer, intent(out) tsteps
```

Definition at line 24 of file checkpointing.f90.

7.1.1.34 varid

```
integer varid
```

Definition at line 202 of file checkpointing.f90.

7.1.1.35 varid_a

```
integer varid_a
```

Definition at line 70 of file checkpointing.f90.

7.1.1.36 varid_c

```
integer varid_c
```

Definition at line 69 of file checkpointing.f90.

7.1.1.37 varid_cstorage

```
integer varid_cstorage
```

Definition at line 70 of file checkpointing.f90.

7.1.1.38 varid_d

```
integer varid_d
```

Definition at line 70 of file checkpointing.f90.

7.1.1.39 varid_dt

```
integer varid_dt
```

Definition at line 69 of file checkpointing.f90.

7.1.1.40 varid_ec

```
integer varid_ec
```

Definition at line 70 of file checkpointing.f90.

7.1.1.41 varid_iapp

```
integer varid_iapp
```

Definition at line 70 of file checkpointing.f90.

7.1.1.42 varid_l

```
integer varid_l
```

Definition at line 70 of file checkpointing.f90.

7.1.1.43 varid_n

```
integer varid_n
```

Definition at line 69 of file checkpointing.f90.

7.1.1.44 varid_r

```
integer varid_r
```

Definition at line 70 of file checkpointing.f90.

7.1.1.45 varid_tstep

```
integer varid_tstep
```

Definition at line 69 of file checkpointing.f90.

7.1.1.46 `varid_tsteps`

`integer varid_tsteps`

Definition at line 69 of file `checkpointing.f90`.

7.2 `/home/chem/msuvnj/PX915_GroupB_22-23/convergence_test.py` File Reference

Namespaces

- `convergence_test`
Test convergence for dt and number of nodes.

Variables

- list `dt_vals` = [0.01,0.05,0.1,0.5,1.0,5.0,10.0]
- `final_voltage` = `np.array([])`
- `final_mass_loss` = `np.array([])`
- `v`
- `mass_loss`
- `dt`
- `nodenum`

7.3 `math.md` File Reference

7.4 `uq.md` File Reference

7.5 `/home/chem/msuvnj/PX915_GroupB_22-23/finite_diff_solver.f90` File Reference

Functions/Subroutines

- program `main`
Crank-Nicolson finite differences solver.

7.5.1 Function/Subroutine Documentation

7.5.1.1 main()

program main

Crank-Nicolson finite differences solver.

Definition at line 22 of file finite_diff_solver.f90.

7.6 /home/chem/msuvnj/PX915_GroupB_22-23/mixed_parallel_strat_test.py File Reference

Namespaces

- [mixed_parallel_strat_test](#)

Variables

- list [num_cores](#) = [1,2,4,8]
- list [num_nodes](#) = [2000]
- [times](#) = np.zeros([len(num_cores),len(num_nodes)])
- [start](#) = time.time()
- [nodenum](#)
- [n](#)
- [dt](#)
- [end](#) = time.time()
- [label](#)

7.7 /home/chem/msuvnj/PX915_GroupB_22-23/MKL_threads_test.py File Reference

Namespaces

- [MKL_threads_test](#)

Variables

- list [num_threads](#) = [1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,20]
- list [num_nodes](#) = [100,500,1000,2000]
- [times](#) = np.zeros([19,4])
- [start](#) = time.time()
- [nodenum](#)
- [n](#)
- [dt](#)
- [end](#) = time.time()
- [label](#)

7.8 /home/chem/msuvnj/PX915_GroupB_22-23/nc_output.f90 File Reference

Modules

- module `nc_output`
Writes results to a netCDF file.

Functions/Subroutines

- subroutine `check` (`status`)
- subroutine `output_cstorage` (`cstorage`, `n`, `tsteps`, `R`, `time_axis`, `electrode_charge`, `filename`)

7.9 /home/chem/msuvnj/PX915_GroupB_22-23/plotter.py File Reference

Namespaces

- `plotter`
Functions for visualisation of solver outputs.

Functions

- def `read_output_file` (`filename`, `step_num`=None)
Reads in data from user input and solver output using NetCDF.
- def `read_input_current` (`filename`, `step_num`=None)
Reads in the applied current.
- def `animated_conc_plot` (`intervaltime`, `dr`, `tsteps`, `nodenum`, `cstore`, `time_axis`, `SaveFinalState`=False, `SparsifyAnimation`=False)
Saves animation of lithium concentration over time.
- def `gen_half_cell_voltage` (`edge_conc_vals`, `i_app_data`, `electrode`, `tsteps`, `pos_params`=None, `neg_params`=None)
- def `voltage_current_plot` (`electrode`, `cstore`, `time_axis`, `i_app_data`, `tsteps`, `pos_params`=None, `neg_params`=None)
Calculates voltages and plots as a function of time.
- def `plot_halfcell_GITT_result` (`filename`, `start_times`, `electrode`, `pos_params`=None, `neg_params`=None, `Animation`=False, `SaveFinalState`=False, `SparsifyAnimation`=True, `animation_interval_time`=10)
- def `gen_plots` (`filename`, `pos_params`=None, `neg_params`=None, `animation_interval_time`=10, `SaveFinalState`=True, `SparsifyAnimation`=False)
- def `full_battery_GITT_plots` (`filename_positive`, `filename_negative`, `start_times`, `pos_params`=None, `neg_params`=None, `SparsifyAnimation`=True, `animation_interval_time`=10)
- def `get_avg_li_conc` (`c_vals`, `r_vals`)
Gets the average concentration of lithium at a given timestep using trapezium rule inside the sphere.

7.10 /home/chem/msuvnj/PX915_GroupB_22-23/read_inputs.f90 File Reference

Modules

- module `read_inputs`
Module to read user inputs.

Variables

- integer `num_args`
- integer `i`
- integer `parse_idx`
- character(len=164) `arg`
- character(len=60) `name`
- character(len=104) `val`
- character(len=104) `filename`
- integer, intent(out) `tsteps`
- real(real64), intent(out) `dt`
- integer, intent(out) `n`
- real(real64), intent(out) `c0`
- real(real64), intent(out) `d`
- real(real64), intent(out) `r`
- real(real64), intent(out) `a_small`
- real(real64), intent(out) `l`
- real(real64), dimension(:), intent(out), allocatable `iapp`
- character(len=1), intent(out) `electrode_charge`
- integer, parameter `file_id` = 1
- integer, parameter `n_params_expected` = 9
- integer `n_params` = 1
- integer `ios`
- logical `asterix` = .FALSE.
- logical `invalid_param` = .FALSE.
- character(len=`n_params_expected`) `param_read`
- character(len=30) `read_temp`
- character(len=1), parameter `parser` = '='
- integer, intent(in) `line`

7.10.1 Variable Documentation

7.10.1.1 `a_small`

```
real(real64), intent(out) a_small
```

Definition at line 112 of file `read_inputs.f90`.

7.10.1.2 `arg`

```
character(len=164) arg
```

Definition at line 31 of file `read_inputs.f90`.

7.10.1.3 asterix

```
logical asterix = .FALSE.
```

Definition at line 146 of file read_inputs.f90.

7.10.1.4 c0

```
real(real64), intent(out) c0
```

Definition at line 103 of file read_inputs.f90.

7.10.1.5 d

```
real(real64), intent(out) d
```

Definition at line 106 of file read_inputs.f90.

7.10.1.6 dt

```
real(real64), intent(out) dt
```

Definition at line 97 of file read_inputs.f90.

7.10.1.7 electrode_charge

```
character(len=1), intent(out) electrode_charge
```

Definition at line 122 of file read_inputs.f90.

7.10.1.8 file_id

```
integer, parameter file_id = 1
```

Definition at line 130 of file read_inputs.f90.

7.10.1.9 filename

```
character(len=*), intent(in) filename
```

Definition at line 40 of file read_inputs.f90.

7.10.1.10 i

```
integer i
```

Definition at line 25 of file read_inputs.f90.

7.10.1.11 iapp

```
real(real64), dimension(:), intent(out), allocatable iapp
```

Definition at line 119 of file read_inputs.f90.

7.10.1.12 invalid_param

```
logical invalid_param = .FALSE.
```

Definition at line 149 of file read_inputs.f90.

7.10.1.13 ios

```
integer ios
```

Definition at line 143 of file read_inputs.f90.

7.10.1.14 l

```
real(real64), intent(out) l
```

Definition at line 115 of file read_inputs.f90.

7.10.1.15 line

```
integer, intent(in) line
```

Definition at line 359 of file read_inputs.f90.

7.10.1.16 n

```
integer, intent(out) n
```

Definition at line 100 of file read_inputs.f90.

7.10.1.17 n_params

```
integer n_params = 1
```

Definition at line 137 of file read_inputs.f90.

7.10.1.18 n_params_expected

```
integer, parameter n_params_expected = 9
```

Definition at line 134 of file read_inputs.f90.

7.10.1.19 name

```
character(len=25), intent(in) name
```

Definition at line 34 of file read_inputs.f90.

7.10.1.20 num_args

```
integer num_args
```

Definition at line 22 of file read_inputs.f90.

7.10.1.21 param_read

```
character (len=n_params_expected) param_read
```

Definition at line 152 of file read_inputs.f90.

7.10.1.22 parse_idx

```
integer parse_idx
```

Definition at line 28 of file read_inputs.f90.

7.10.1.23 parser

```
character(len=1), parameter parser '='
```

Definition at line 159 of file read_inputs.f90.

7.10.1.24 r

```
real(real64), intent(out) r
```

Definition at line 109 of file read_inputs.f90.

7.10.1.25 read_temp

```
character(len=30) read_temp
```

Definition at line 156 of file read_inputs.f90.

7.10.1.26 tsteps

```
integer, intent(out) tsteps
```

Definition at line 94 of file read_inputs.f90.

7.10.1.27 val

```
character(len=25) val
```

Definition at line 37 of file read_inputs.f90.

7.11 /home/chem/msuvnj/PX915_GroupB_22-23/sensitivity_analysis.py File Reference

Namespaces

- [sensitivity_analysis](#)
First order sensitivity analysis.

Functions

- def [sensitivity_over_time](#) ()
- def [update](#) (frame)

Variables

- def [scaled_sensitivities_matrix](#) = sensitivity_over_time()
- [fig](#)
- [ax](#)
- list [x_labels](#) = ['c0', 'D', 'R', 'a', 'L']
- [ani](#) = FuncAnimation(fig, update, frames=len(scaled_sensitivities_matrix), blit=True)
- [filename](#)
- [writer](#)
- [fps](#)

7.12 /home/chem/msuvnj/PX915_GroupB_22-23/Uncertainty_↔ Propagation.py File Reference

Namespaces

- [Uncertainty_Propagation](#)
Performs uncertainty propagation.

Functions

- def [in_out_easy_peasy](#) (parameter_np_array)

Variables

- `iapp` = `np.concatenate((np.array([20.0 for i in range(50)]),np.array([0.0 for i in range(50)])))`
*Read in applied current density from csv file iapp_filename = 'WLTP_m10.csv' iapp, iapp_label, tsteps = UI.iapp ↵
read_csv(iapp_filename)*
- `parameters` = `parameter_np_array`
END MEAN SET VALUES #####.
- `int nprocs` = 40
Call the function to perform the full battery parallel solve.
- `output_filename_positive`
- `output_filename_negative`
- `cstore_pos`
Obtaining QOI (Voltage) #.
- `tsteps`
- `nodenum`
- `R_pos`
- `time_axis`
- `dr_pos`
- `electrode_pos`
- `cstore_neg`
- `R_neg`
- `dr_neg`
- `electrode_neg`
- `i_app_data_pos` = `plotter.read_input_current(output_filename_positive)`
- `i_app_data_neg` = `plotter.read_input_current(output_filename_negative)`
- `edge_conc_vals_pos` = `cstore_pos[:, -1]`
- `edge_conc_vals_neg` = `cstore_neg[:, -1]`
- `list posparams` = `[K_pos, parameters[3], cmax_pos_sim, parameters[4]]`
- `list negparams` = `[K_neg, a_neg, cmax_neg_sim, L_neg]`
- `pos_voltage` = `plotter.gen_half_cell_voltage(edge_conc_vals_pos, i_app_data_pos, electrode_pos, tsteps, pos ↵
_params=posparams)`
- `neg_voltage` = `plotter.gen_half_cell_voltage(edge_conc_vals_neg, i_app_data_neg, electrode_neg, tsteps, neg ↵
_params=negparams)`
- `full_voltage` = `pos_voltage - neg_voltage`
- `c0_dist` = `st.norm(1000.0, 25.0)`
Define the input parameter distributions #.
- `D_dist` = `st.norm(4.0e-15, 0.5e-15)`
- `R_dist` = `st.norm(5.86e-6, 1.0e-6)`
- `a_dist` = `st.norm(3.0*0.665/5.86e-6, 1.0e5)`
- `L_dist` = `st.norm(75.6e-6, 5.0e-6)`
- `int num_samples` = 40
- `c0_samples` = `c0_dist.rvs(num_samples)`
- `D_samples` = `D_dist.rvs(num_samples)`
- `R_samples` = `R_dist.rvs(num_samples)`
- `a_samples` = `a_dist.rvs(num_samples)`
- `L_samples` = `L_dist.rvs(num_samples)`
- `fixed_c0` = `c0_dist.mean()`
- `fixed_D` = `D_dist.mean()`
- `fixed_R` = `R_dist.mean()`
- `fixed_a` = `a_dist.mean()`
- `fixed_L` = `L_dist.mean()`
- `int counter` = 0
- `dt`
- `time` = `np.linspace(0.0, dt*(tsteps-1.), tsteps)`

```

_, _, tsteps = UI.iapp_read_csv(iapp_filename)
• voltage_curves = np.zeros(tsteps)
• fig
    UQ Plot #.
• axs
• figsize
• sharex
• True
• sharey
• params = np.array([c0_samples[i], fixed_D, fixed_R, fixed_a, fixed_L])
• alpha
• figg
    Input Distributions #.
• axss
• False
• int resol = 1000
• x = np.linspace(900, 1100, resol)

```

7.13 /home/chem/msuvnj/PX915_GroupB_22-23/unit_test.py File Reference

Namespaces

- `unit_test`

Functions

- def `run_unit_test` (nodenum=500, dt=1.0, num_cores=1)

7.14 /home/chem/msuvnj/PX915_GroupB_22-23/user_input.py File Reference

Namespaces

- `user_input`

Sets up the user inputs and executes the solver.

Variables

- bool `checkpoint` = False
- string `solver_input_filename` = 'user_input'
- `tsteps`
`SET VALUES #####.`
- `dt`
- `n`
- `c0`
- `D`
- `R`
- `a`
- `L`
- `iapp`
- `iapp_label`
- `electrode_charge`
- float `K_pos` = 3.42E-6
- float `K_neg` = 6.48E-7
- float `cmax_pos_sim` = 63104.00
- float `cmax_neg_sim` = 33133.00
- list `plot_params_pos` = [K_pos,a,cmax_pos_sim,L]
`END SET VALUES #####.`
- `pos_params`

7.15 /home/chem/msuvnj/PX915_GroupB_22-23/user_input_full_battery_GITT.py File Reference

Namespaces

- `user_input_full_battery_GITT`

Variables

- string `output_filename_positive` = 'current_step_pos'
`Uncomment to print stdout to file sys.stdout = open('test.txt', 'w')`
- string `output_filename_negative` = 'current_step_neg'
- `tsteps`
`Import default values from https://doi.org/10.1149/1945-7111/ab9050 Use UI.set_defaults_pos() for positive electrode and ..._neg() for negative electrode.`
- `dt` = 1.0
- `n` = 1000
- `c0`
- `D_neg`
- `R_neg`
- `a_neg`
- `L_neg`
- `iapp_neg`
- `iapp_label_neg`
- `electrode_charge_neg`

- [D_pos](#)
- [R_pos](#)
- [a_pos](#)
- [L_pos](#)
- [iapp_pos](#)
- [iapp_label_pos](#)
- [electrode_charge_pos](#)
- float [c0_pos](#) = 30000.0
- Set values #####.*
- float [c0_neg](#) = 1.0
- float [K_pos](#) = 3.42E-6
- float [K_neg](#) = 6.48E-7
- float [cmax_pos_sim](#) = 63104.00
- float [cmax_neg_sim](#) = 33133.00
- int [nprocs](#) = 40
- Check parameters are valid #####.*
- int [nsteps](#) = 10
- list [currents](#) = [20.0 for [i](#) in range(nsteps)]
- list [start_times](#) = [2150.0*i for [i](#) in range(nsteps)]
- list [run_times](#) = [150.0 for [i](#) in range(nsteps)]
- list [wait_times](#) = [2000.0 for [i](#) in range(nsteps)]
- list [params_pos](#) = [[dt](#), [c0_pos](#), [D_pos](#), [R_pos](#), [a_pos](#), [L_pos](#), [electrode_charge_pos](#)]
- list [params_neg](#) = [[dt](#), [c0_neg](#), [D_neg](#), [R_neg](#), [a_neg](#), [L_neg](#), [electrode_charge_neg](#)]
- list [plot_params_pos](#) = [[K_pos](#),[a_pos](#),[cmax_pos_sim](#),[L_pos](#)]
- Call the function to perform the full battery parallel solve.*
- list [plot_params_neg](#) = [[K_neg](#),[a_neg](#),[cmax_neg_sim](#),[L_neg](#)]
- [pos_params](#)
- [neg_params](#)
- [SparsifyAnimation](#)
- [True](#)
- [animation_interval_time](#)

7.16 /home/chem/msuvnj/PX915_GroupB_22-23/user_input_mod.py File Reference

Namespaces

- [user_input_mod](#)
- Package containing the functions needed to set up the user input parameters and execute the solver.*

Functions

- def [iapp_read_csv](#) ([filename](#))
- CURRENT DENSITY SET UP ###.*
- def [iapp_constant_setup](#) ([tsteps](#), [iapp](#))
- Sets up the applied current density 'iapp' as a constant valued array of length tsteps.*
- def [iapp_step_setup](#) ([tsteps](#), [iapp_steps](#))
- Sets up the applied current density 'iapp' as a stepped array of length tsteps.*
- def [set_defaults_pos](#) ()
- DEFAULT PARAMETERS ###.*

- def `set_defaults_neg` ()
Returns default parameters for a negative electrode.
- def `verify_params` (filename, tsteps, dt, n, c0, D, R, a, L, electrode_charge)
PARAMETER VERIFICATION ###.
- def `verify_iapp` (iapp, iapp_label, tsteps)
Verifies the types and values of the applied current density array and its label.
- def `write_to_file` (filename, tsteps, dt, n, c0, D, R, a, L, iapp, iapp_label, electrode_charge)
WRITE TO FILE ###.
- def `call_solver` (filename, checkpoint, nprocs=1)
CALL SOLVER ###.
- def `get_GITT_initial_concs` (currents, run_times, c0, R, a, L, electrode_charge)
- def `GITT_half_cell` (filename, nprocs, currents, start_times, run_times, wait_times, n, params)
INITIALISE A FULL GITT TEST IN PARALLEL ####.
- def `GITT_full_cell` (filename_positive, filename_negative, nprocs, currents, start_times, run_times, wait_times, n, params_pos, params_neg)
Executes the SPM solver in parallel to run a GITT full cell test over nprocs cores.
- def `full_battery_simulation` (filename_positive, filename_negative, nprocs=1)
Executes the SPM solver.
- def `optimise_parallelism` (n_procs, n_gitt_steps=1, full_battery=False)
Decides the correct number of threads to use for a given simulation.

7.17 /home/chem/msuvnj/PX915_GroupB_22-23/user_input_parallel_simulation.py File Reference

Namespaces

- `user_input_parallel_simulation`
a running script which exists to demonstrate how one uses the GITT function to run simulations in parallel.

Variables

- string `output_filename` = 'current_step'
- string `electrode` = 'positive'
- `tsteps`
- `dt` = 1.0
- `n`
- `c0` = 30000.0
- `D`
- `R`
- `a`
- `L`
- `iapp`
- `iapp_label`
- `electrode_charge`
- float `K_pos` = 3.42E-6
- float `K_neg` = 6.48E-7
- float `cmax_pos_sim` = 63104.00
- float `cmax_neg_sim` = 33133.00

- int `nprocs` = 5
- int `nsteps` = 5
- list `currents` = [20.0 for `i` in range(nsteps)]
- list `start_times` = [2150.0*i for `i` in range(nsteps)]
- list `run_times` = [150.0 for `i` in range(nsteps)]
- list `wait_times` = [2000.0 for `i` in range(nsteps)]
- list `params` = [`dt`, `c0`, `D`, `R`, `a`, `L`, `electrode_charge`]
- list `plot_params_neg` = [`K_neg`, `a`, `cmax_neg_sim`, `L`]
- `neg_params`
- `Animation`
- `True`
- `SparsifyAnimation`

Index

/home/chem/msuvnj/PX915_GroupB_22-23/MKL_threads_test.py, 18
73
Animation
/home/chem/msuvnj/PX915_GroupB_22-23/Uncertainty_Propagation.py, 58
80
animation_interval_time
/home/chem/msuvnj/PX915_GroupB_22-23/checkpointing.f90, user_input_full_battery_GITT, 43
63
arg
/home/chem/msuvnj/PX915_GroupB_22-23/convergence_test.py, read_inputs.f90, 75
72
asterix
/home/chem/msuvnj/PX915_GroupB_22-23/finite_diff_solver.f90, read_inputs.f90, 75
72
ax
/home/chem/msuvnj/PX915_GroupB_22-23/mixed_parallel_strategy.py, sensitivity_analysis, 23
73
axs
/home/chem/msuvnj/PX915_GroupB_22-23/nc_output.f90, Uncertainty_Propagation, 27
74
axss
/home/chem/msuvnj/PX915_GroupB_22-23/plotter.py, Uncertainty_Propagation, 27
74
/home/chem/msuvnj/PX915_GroupB_22-23/read_inputs.f90, 74
74
checkpointing.f90, 64
/home/chem/msuvnj/PX915_GroupB_22-23/sensitivity_analysis.py, 80
80
checkpointing.f90, 64
/home/chem/msuvnj/PX915_GroupB_22-23/unit_test.py, read_inputs.f90, 76
82
user_input, 38
/home/chem/msuvnj/PX915_GroupB_22-23/user_input.py, user_input_full_battery_GITT, 43
82
user_input_parallel_simulation, 58
/home/chem/msuvnj/PX915_GroupB_22-23/user_input_full_battery_GITT.py, 83
83
c0_dist, Uncertainty_Propagation, 27
/home/chem/msuvnj/PX915_GroupB_22-23/user_input_mod.py, 84
84
c0_neg, user_input_full_battery_GITT, 43
/home/chem/msuvnj/PX915_GroupB_22-23/user_input_parallel_simulation.py, 85
85
c0_pos, user_input_full_battery_GITT, 43
c0_samples
Uncertainty_Propagation, 27
a
user_input, 38
user_input_parallel_simulation, 58
a_dist
Uncertainty_Propagation, 26
a_neg
user_input_full_battery_GITT, 42
a_pos
user_input_full_battery_GITT, 43
a_samples
Uncertainty_Propagation, 26
a_small
checkpointing.f90, 64
read_inputs.f90, 75
alpha
Uncertainty_Propagation, 27
ani
sensitivity_analysis, 23
animated_conc_plot
call_solver
user_input_mod, 51
check
nc_output, 17
checkpoint
user_input, 38
checkpoint_name
checkpointing.f90, 64
checkpointing, 11
checkpointing.f90
a_small, 64
c, 64
c0, 64
checkpoint_name, 64
cstorage, 65
d, 65
dimid_n, 65
dimid_tsteps, 65

- dimids_cstorage, 65
- dimids_parameter, 65
- dir, 66
- dt, 66
- electrode_charge, 66
- file_ext, 66
- file_extension, 66
- file_test, 66
- filename, 67
- filename_length, 67
- filepath, 67
- freq, 67
- freq_in, 67
- freq_remainder, 67
- iapp, 68
- l, 68
- n, 68
- ncid, 68
- parse_idx, 68
- r, 68
- status, 69
- tstep, 69
- tstep_init, 69
- tstep_str, 69
- tsteps, 69
- varid, 69
- varid_a, 70
- varid_c, 70
- varid_cstorage, 70
- varid_d, 70
- varid_dt, 70
- varid_ec, 70
- varid_iapp, 71
- varid_l, 71
- varid_n, 71
- varid_r, 71
- varid_tstep, 71
- varid_tsteps, 71
- cmax_neg_sim
 - user_input, 38
 - user_input_full_battery_GITT, 43
 - user_input_parallel_simulation, 58
- cmax_pos_sim
 - user_input, 38
 - user_input_full_battery_GITT, 44
 - user_input_parallel_simulation, 59
- convergence_test, 11
 - dt, 12
 - dt_vals, 12
 - final_mass_loss, 12
 - final_voltage, 12
 - mass_loss, 12
 - nodenum, 12
 - v, 13
- counter
 - Uncertainty_Propagation, 27
- cstorage
 - checkpointing.f90, 65
- cstore_neg
 - Uncertainty_Propagation, 28
- cstore_pos
 - Uncertainty_Propagation, 28
- currents
 - user_input_full_battery_GITT, 44
 - user_input_parallel_simulation, 59
- D
 - user_input, 39
 - user_input_parallel_simulation, 59
- d
 - checkpointing.f90, 65
 - read_inputs.f90, 76
- D_dist
 - Uncertainty_Propagation, 28
- D_neg
 - user_input_full_battery_GITT, 44
- D_pos
 - user_input_full_battery_GITT, 44
- D_samples
 - Uncertainty_Propagation, 28
- dimid_n
 - checkpointing.f90, 65
- dimid_tsteps
 - checkpointing.f90, 65
- dimids_cstorage
 - checkpointing.f90, 65
- dimids_parameter
 - checkpointing.f90, 65
- dir
 - checkpointing.f90, 66
- dr_neg
 - Uncertainty_Propagation, 28
- dr_pos
 - Uncertainty_Propagation, 28
- dt
 - checkpointing.f90, 66
 - convergence_test, 12
 - mixed_parallel_strat_test, 13
 - MKL_threads_test, 15
 - read_inputs.f90, 76
 - Uncertainty_Propagation, 29
 - user_input, 39
 - user_input_full_battery_GITT, 44
 - user_input_parallel_simulation, 59
- dt_vals
 - convergence_test, 12
- edge_conc_vals_neg
 - Uncertainty_Propagation, 29
- edge_conc_vals_pos
 - Uncertainty_Propagation, 29
- electrode
 - user_input_parallel_simulation, 59
- electrode_charge
 - checkpointing.f90, 66
 - read_inputs.f90, 76
 - user_input, 39

- user_input_parallel_simulation, 59
- electrode_charge_neg
 - user_input_full_battery_GITT, 44
- electrode_charge_pos
 - user_input_full_battery_GITT, 45
- electrode_neg
 - Uncertainty_Propagation, 29
- electrode_pos
 - Uncertainty_Propagation, 29
- end
 - mixed_parallel_strat_test, 13
 - MKL_threads_test, 15
- False
 - Uncertainty_Propagation, 29
- fig
 - sensitivity_analysis, 23
 - Uncertainty_Propagation, 30
- figg
 - Uncertainty_Propagation, 30
- figsize
 - Uncertainty_Propagation, 30
- file_ext
 - checkpointing.f90, 66
- file_extension
 - checkpointing.f90, 66
- file_id
 - read_inputs.f90, 76
- file_test
 - checkpointing.f90, 66
- filename
 - checkpointing.f90, 67
 - read_inputs.f90, 76
 - sensitivity_analysis, 23
- filename_length
 - checkpointing.f90, 67
- filepath
 - checkpointing.f90, 67
- final_mass_loss
 - convergence_test, 12
- final_voltage
 - convergence_test, 12
- finite_diff_solver.f90
 - main, 72
- fixed_a
 - Uncertainty_Propagation, 30
- fixed_c0
 - Uncertainty_Propagation, 30
- fixed_D
 - Uncertainty_Propagation, 30
- fixed_L
 - Uncertainty_Propagation, 31
- fixed_R
 - Uncertainty_Propagation, 31
- fps
 - sensitivity_analysis, 23
- freq
 - checkpointing.f90, 67
- freq_in
 - checkpointing.f90, 67
- freq_remainder
 - checkpointing.f90, 67
- full_battery_GITT_plots
 - plotter, 18
- full_battery_simulation
 - user_input_mod, 51
- full_voltage
 - Uncertainty_Propagation, 31
- gen_half_cell_voltage
 - plotter, 19
- gen_plots
 - plotter, 19
- get_avg_li_conc
 - plotter, 19
- get_GITT_initial_concs
 - user_input_mod, 51
- GITT_full_cell
 - user_input_mod, 52
- GITT_half_cell
 - user_input_mod, 53
- i
 - read_inputs.f90, 77
- i_app_data_neg
 - Uncertainty_Propagation, 31
- i_app_data_pos
 - Uncertainty_Propagation, 31
- iapp
 - checkpointing.f90, 68
 - read_inputs.f90, 77
 - Uncertainty_Propagation, 31
 - user_input, 39
 - user_input_parallel_simulation, 60
- iapp_constant_setup
 - user_input_mod, 53
- iapp_label
 - user_input, 39
 - user_input_parallel_simulation, 60
- iapp_label_neg
 - user_input_full_battery_GITT, 45
- iapp_label_pos
 - user_input_full_battery_GITT, 45
- iapp_neg
 - user_input_full_battery_GITT, 45
- iapp_pos
 - user_input_full_battery_GITT, 45
- iapp_read_csv
 - user_input_mod, 54
- iapp_step_setup
 - user_input_mod, 54
- in_out_easy_peasy
 - Uncertainty_Propagation, 26
- invalid_param
 - read_inputs.f90, 77
- ios
 - read_inputs.f90, 77

- K_neg
 - user_input, 39
 - user_input_full_battery_GITT, 45
 - user_input_parallel_simulation, 60
- K_pos
 - user_input, 40
 - user_input_full_battery_GITT, 46
 - user_input_parallel_simulation, 60
- L
 - user_input, 40
 - user_input_parallel_simulation, 60
- I
 - checkpointing.f90, 68
 - read_inputs.f90, 77
- L_dist
 - Uncertainty_Propagation, 32
- L_neg
 - user_input_full_battery_GITT, 46
- L_pos
 - user_input_full_battery_GITT, 46
- L_samples
 - Uncertainty_Propagation, 32
- label
 - mixed_parallel_strat_test, 13
 - MKL_threads_test, 15
- line
 - read_inputs.f90, 77
- main
 - finite_diff_solver.f90, 72
- mass_loss
 - convergence_test, 12
- math.md, 72
- mixed_parallel_strat_test, 13
 - dt, 13
 - end, 13
 - label, 13
 - n, 14
 - nodenum, 14
 - num_cores, 14
 - num_nodes, 14
 - start, 14
 - times, 14
- MKL_threads_test, 15
 - dt, 15
 - end, 15
 - label, 15
 - n, 15
 - nodenum, 16
 - num_nodes, 16
 - num_threads, 16
 - start, 16
 - times, 16
- n
 - checkpointing.f90, 68
 - mixed_parallel_strat_test, 14
 - MKL_threads_test, 15
 - read_inputs.f90, 78
 - user_input, 40
 - user_input_full_battery_GITT, 46
 - user_input_parallel_simulation, 60
- n_params
 - read_inputs.f90, 78
- n_params_expected
 - read_inputs.f90, 78
- name
 - read_inputs.f90, 78
- nc_output, 17
 - check, 17
 - output_cstorage, 17
- ncid
 - checkpointing.f90, 68
- neg_params
 - user_input_full_battery_GITT, 46
 - user_input_parallel_simulation, 61
- neg_voltage
 - Uncertainty_Propagation, 32
- negparams
 - Uncertainty_Propagation, 32
- nodenum
 - convergence_test, 12
 - mixed_parallel_strat_test, 14
 - MKL_threads_test, 16
 - Uncertainty_Propagation, 32
- nprocs
 - Uncertainty_Propagation, 32
 - user_input_full_battery_GITT, 46
 - user_input_parallel_simulation, 61
- nsteps
 - user_input_full_battery_GITT, 47
 - user_input_parallel_simulation, 61
- num_args
 - read_inputs.f90, 78
- num_cores
 - mixed_parallel_strat_test, 14
- num_nodes
 - mixed_parallel_strat_test, 14
 - MKL_threads_test, 16
- num_samples
 - Uncertainty_Propagation, 33
- num_threads
 - MKL_threads_test, 16
- optimise_parallelism
 - user_input_mod, 54
- output_cstorage
 - nc_output, 17
- output_filename
 - user_input_parallel_simulation, 61
- output_filename_negative
 - Uncertainty_Propagation, 33
 - user_input_full_battery_GITT, 47
- output_filename_positive
 - Uncertainty_Propagation, 33
 - user_input_full_battery_GITT, 47

- param_read
 - read_inputs.f90, 78
- parameters
 - Uncertainty_Propagation, 33
- params
 - Uncertainty_Propagation, 33
 - user_input_parallel_simulation, 61
- params_neg
 - user_input_full_battery_GITT, 47
- params_pos
 - user_input_full_battery_GITT, 47
- parse_idx
 - checkpointing.f90, 68
 - read_inputs.f90, 79
- parser
 - read_inputs.f90, 79
- plot_halfcell_GITT_result
 - plotter, 20
- plot_params_neg
 - user_input_full_battery_GITT, 48
 - user_input_parallel_simulation, 61
- plot_params_pos
 - user_input, 40
 - user_input_full_battery_GITT, 48
- plotter, 17
 - animated_conc_plot, 18
 - full_battery_GITT_plots, 18
 - gen_half_cell_voltage, 19
 - gen_plots, 19
 - get_avg_li_conc, 19
 - plot_halfcell_GITT_result, 20
 - read_input_current, 20
 - read_output_file, 21
 - voltage_current_plot, 21
- pos_params
 - user_input, 40
 - user_input_full_battery_GITT, 48
- pos_voltage
 - Uncertainty_Propagation, 33
- posparams
 - Uncertainty_Propagation, 34
- R
 - user_input, 40
 - user_input_parallel_simulation, 62
- r
 - checkpointing.f90, 68
 - read_inputs.f90, 79
- R_dist
 - Uncertainty_Propagation, 34
- R_neg
 - Uncertainty_Propagation, 34
 - user_input_full_battery_GITT, 48
- R_pos
 - Uncertainty_Propagation, 34
 - user_input_full_battery_GITT, 48
- R_samples
 - Uncertainty_Propagation, 34
- read_input_current
 - plotter, 20
- read_inputs, 22
 - read_inputs.f90
 - a_small, 75
 - arg, 75
 - asterix, 75
 - c0, 76
 - d, 76
 - dt, 76
 - electrode_charge, 76
 - file_id, 76
 - filename, 76
 - i, 77
 - iapp, 77
 - invalid_param, 77
 - ios, 77
 - l, 77
 - line, 77
 - n, 78
 - n_params, 78
 - n_params_expected, 78
 - name, 78
 - num_args, 78
 - param_read, 78
 - parse_idx, 79
 - parser, 79
 - r, 79
 - read_temp, 79
 - tsteps, 79
 - val, 79
 - read_output_file
 - plotter, 21
 - read_temp
 - read_inputs.f90, 79
- resol
 - Uncertainty_Propagation, 34
- run_times
 - user_input_full_battery_GITT, 48
 - user_input_parallel_simulation, 62
- run_unit_test
 - unit_test, 36
- scaled_sensitivities_matrix
 - sensitivity_analysis, 24
- sensitivity_analysis, 22
 - ani, 23
 - ax, 23
 - fig, 23
 - filename, 23
 - fps, 23
 - scaled_sensitivities_matrix, 24
 - sensitivity_over_time, 22
 - update, 23
 - writer, 24
 - x_labels, 24
- sensitivity_over_time
 - sensitivity_analysis, 22
- set_defaults_neg
 - user_input_mod, 55

- set_defaults_pos
 - user_input_mod, 55
- sharex
 - Uncertainty_Propagation, 35
- sharey
 - Uncertainty_Propagation, 35
- solver_input_filename
 - user_input, 41
- SparsifyAnimation
 - user_input_full_battery_GITT, 49
 - user_input_parallel_simulation, 62
- start
 - mixed_parallel_strat_test, 14
 - MKL_threads_test, 16
- start_times
 - user_input_full_battery_GITT, 49
 - user_input_parallel_simulation, 62
- status
 - checkpointing.f90, 69
- time
 - Uncertainty_Propagation, 35
- time_axis
 - Uncertainty_Propagation, 35
- times
 - mixed_parallel_strat_test, 14
 - MKL_threads_test, 16
- True
 - Uncertainty_Propagation, 35
 - user_input_full_battery_GITT, 49
 - user_input_parallel_simulation, 62
- tstep
 - checkpointing.f90, 69
- tstep_init
 - checkpointing.f90, 69
- tstep_str
 - checkpointing.f90, 69
- tsteps
 - checkpointing.f90, 69
 - read_inputs.f90, 79
 - Uncertainty_Propagation, 35
 - user_input, 41
 - user_input_full_battery_GITT, 49
 - user_input_parallel_simulation, 62
- Uncertainty_Propagation, 24
 - a_dist, 26
 - a_samples, 26
 - alpha, 27
 - axs, 27
 - axss, 27
 - c0_dist, 27
 - c0_samples, 27
 - counter, 27
 - cstore_neg, 28
 - cstore_pos, 28
 - D_dist, 28
 - D_samples, 28
 - dr_neg, 28
 - dr_pos, 28
 - dt, 29
 - edge_conc_vals_neg, 29
 - edge_conc_vals_pos, 29
 - electrode_neg, 29
 - electrode_pos, 29
 - False, 29
 - fig, 30
 - figg, 30
 - figsize, 30
 - fixed_a, 30
 - fixed_c0, 30
 - fixed_D, 30
 - fixed_L, 31
 - fixed_R, 31
 - full_voltage, 31
 - i_app_data_neg, 31
 - i_app_data_pos, 31
 - iapp, 31
 - in_out_easy_peasy, 26
 - L_dist, 32
 - L_samples, 32
 - neg_voltage, 32
 - negparams, 32
 - nodenum, 32
 - nprocs, 32
 - num_samples, 33
 - output_filename_negative, 33
 - output_filename_positive, 33
 - parameters, 33
 - params, 33
 - pos_voltage, 33
 - posparams, 34
 - R_dist, 34
 - R_neg, 34
 - R_pos, 34
 - R_samples, 34
 - resol, 34
 - sharex, 35
 - sharey, 35
 - time, 35
 - time_axis, 35
 - True, 35
 - tsteps, 35
 - voltage_curves, 36
 - x, 36
- unit_test, 36
 - run_unit_test, 36
- update
 - sensitivity_analysis, 23
- uq.md, 72
- user_input, 36
 - a, 38
 - c0, 38
 - checkpoint, 38
 - cmax_neg_sim, 38
 - cmax_pos_sim, 38
 - D, 39

- dt, 39
- electrode_charge, 39
- iapp, 39
- iapp_label, 39
- K_neg, 39
- K_pos, 40
- L, 40
- n, 40
- plot_params_pos, 40
- pos_params, 40
- R, 40
- solver_input_filename, 41
- tsteps, 41
- user_input_full_battery_GITT, 41
 - a_neg, 42
 - a_pos, 43
 - animation_interval_time, 43
 - c0, 43
 - c0_neg, 43
 - c0_pos, 43
 - cmax_neg_sim, 43
 - cmax_pos_sim, 44
 - currents, 44
 - D_neg, 44
 - D_pos, 44
 - dt, 44
 - electrode_charge_neg, 44
 - electrode_charge_pos, 45
 - iapp_label_neg, 45
 - iapp_label_pos, 45
 - iapp_neg, 45
 - iapp_pos, 45
 - K_neg, 45
 - K_pos, 46
 - L_neg, 46
 - L_pos, 46
 - n, 46
 - neg_params, 46
 - nprocs, 46
 - nsteps, 47
 - output_filename_negative, 47
 - output_filename_positive, 47
 - params_neg, 47
 - params_pos, 47
 - plot_params_neg, 48
 - plot_params_pos, 48
 - pos_params, 48
 - R_neg, 48
 - R_pos, 48
 - run_times, 48
 - SparsifyAnimation, 49
 - start_times, 49
 - True, 49
 - tsteps, 49
 - wait_times, 49
- user_input_mod, 50
 - call_solver, 51
 - full_battery_simulation, 51
 - get_GITT_initial_concs, 51
 - GITT_full_cell, 52
 - GITT_half_cell, 53
 - iapp_constant_setup, 53
 - iapp_read_csv, 54
 - iapp_step_setup, 54
 - optimise_parallelism, 54
 - set_defaults_neg, 55
 - set_defaults_pos, 55
 - verify_iapp, 55
 - verify_params, 56
 - write_to_file, 56
- user_input_parallel_simulation, 57
 - a, 58
 - Animation, 58
 - c0, 58
 - cmax_neg_sim, 58
 - cmax_pos_sim, 59
 - currents, 59
 - D, 59
 - dt, 59
 - electrode, 59
 - electrode_charge, 59
 - iapp, 60
 - iapp_label, 60
 - K_neg, 60
 - K_pos, 60
 - L, 60
 - n, 60
 - neg_params, 61
 - nprocs, 61
 - nsteps, 61
 - output_filename, 61
 - params, 61
 - plot_params_neg, 61
 - R, 62
 - run_times, 62
 - SparsifyAnimation, 62
 - start_times, 62
 - True, 62
 - tsteps, 62
 - wait_times, 62
- v
 - convergence_test, 13
- val
 - read_inputs.f90, 79
- varid
 - checkpointing.f90, 69
- varid_a
 - checkpointing.f90, 70
- varid_c
 - checkpointing.f90, 70
- varid_cstorage
 - checkpointing.f90, 70
- varid_d
 - checkpointing.f90, 70
- varid_dt
 - checkpointing.f90, 70

- varid_ec
 - checkpointing.f90, [70](#)
- varid_iapp
 - checkpointing.f90, [71](#)
- varid_l
 - checkpointing.f90, [71](#)
- varid_n
 - checkpointing.f90, [71](#)
- varid_r
 - checkpointing.f90, [71](#)
- varid_tstep
 - checkpointing.f90, [71](#)
- varid_tsteps
 - checkpointing.f90, [71](#)
- verify_iapp
 - user_input_mod, [55](#)
- verify_params
 - user_input_mod, [56](#)
- voltage_current_plot
 - plotter, [21](#)
- voltage_curves
 - Uncertainty_Propagation, [36](#)
- wait_times
 - user_input_full_battery_GITT, [49](#)
 - user_input_parallel_simulation, [62](#)
- write_to_file
 - user_input_mod, [56](#)
- writer
 - sensitivity_analysis, [24](#)
- x
 - Uncertainty_Propagation, [36](#)
- x_labels
 - sensitivity_analysis, [24](#)