python-crunchflow

Release 1.0.3

Zach Perzan

May 06, 2021

CONTENTS:

1 Spatial (tec) Output 3

2 Time Series Output 7

3 Utilities 9

4 Indices and tables 11

Python Module Index 13

Index 15

python-crunchflow is an open source python package designed for processing and visualizing output from the reactive transport model CrunchFlow. This library provides a set of tools for quickly plotting time series and geochemical output produced by CrunchFlow.

This library consists of three main modules:

- `output.tec`. Class for working with CrunchFlow `spatial_profile` output, which include spatially-variable geochemical conditions and sediment properties at a given point in time.
- `output.timeseries`. Class for working with CrunchFlow `time_series` output, which consists of temporally-variable aqueous concentrations at a specific node.
- `util`. General utilities for parsing CrunchFlow input files and manipulating output.

Contents:

CHAPTER

ONE

SPATIAL (TEC) OUTPUT

output.tec.**get_tec_metadata**(*file*)

Given a crunch .tec output file, read it in and return a list of the variables (e.g., ‘X-Perm’, ‘Y-Perm’, etc) included in the output file.

Parameters

file [str] Filename to read in

Returns

columns [list, str] ordered list of variables included in filename

title [str] Value included in this file, as output by CrunchFlow

output.tec.**get_tec_output_times**(*crunch_log*, *folder='.'*)

Given a log of CrunchFlow terminal output, get the time steps associated with .tec file numbers. For example, during a model run, CrunchFlow will print progress to the screen, including blocks that look like:

```
>>> WRITING OUTPUT FILES
>>> Time (days) =  9.000E+00
>>> File number  =            2
```

This function combs through that terminal output to create a list of times at which .tec files were written.

Parameters

crunch_log [str] Filename of the crunchflow terminal output

folder [str] Folder in which ‘crunch_log’ is located.The default is the current directory

Returns

output_times [list of float] Ordered list of the output times (in CrunchFlow time units) corresponding to each .tec file number.

class output.tec.**tec**(*fileprefix*, *folder='.'*, *output_times=None*)

This is the tec class for working with CrunchFlow .tec files.

Examples

```
>>> vol = tec('volume')
>>> print(vol.columns)
>>> vol.plot('Calcite')
>>> calcite = vol.extract('Calcite', time=2)
```

Attributes

times [list of int] relative times at which .tec files were output, sorted

files [list of str] ordered list of all files matching fileprefix

columns [list of str] list of all vars included within each .tec file

nx [int] # of grid cells in the x direction

ny [int] # of grid cells in the y direction

nz [int] # of grid cells in the z direction

coords [ndarray of float] (nx*ny*nz, 2) array of x, y coordinates of each node

griddedX [ndarray of float] (ny, nx) array of x coordinates; used for plotting

griddedY [ndarray of float] (ny, nx) array of y coordinates; used for plotting

Methods

plot(variable, time)	Plot .tec output from a single time step.
plot_series(variable)	Plot .tec output for a series of time steps.
extract(variable, time)	Return the spatial profile of var at the given time.
outline(variable, value, time)	Get line segments that outline all the regions equal to the provided value.

`extract`(*var*, *time=1*)

Return the spatial profile of var at the given time.

Parameters

var [str] variable to return (e.g., ‘Porosity’)

time [int] time slice at which to return the profile

Returns

data [ndarray of float] (ny, nx) numpy array of var

`outline`(*var*, *value=None*, *time=1*)

For a given .tec file, get line segments that outline all the regions equal to the provided value. Useful for generating outlines of a areas of a model domain sharing a single attribute (e.g., permeability) and later adding the outlines to a plot of a different variable.

Parameters

var [str] variable to outline (e.g., ‘Porosity’)

value [float] value to outline. The default is least-frequent value in array

time [int] time slice at which to create an outline

Returns

segments [ndarray of float] array of (x,y) coordinate pairs for each line segment that comprises the outline

Examples

```
>>> # Get stratigraphy from permeability map
>>> perm = tec('permeability')
>>> segments = perm.outline('X-Perm')
>>>
>>> # Plot stratigraphy outlines on O2 map
>>> conc = tec('conc')
>>> fig, ax = conc.plot('O2(aq)')
>>> ax.plot(segments[:,0], segments[:,1])
```

plot (*var*, *time=None*, *plot_type='image'*, *figsize=(12, 3)*, ***kwargs*)
Plot .tec output from a single time step.

Parameters

var [str] variable to plot (e.g., ‘H+’)

time [int] which time slice to show

figsize [tuple of int] figure size; default (12, 3)

plot_type [str] whether to display an image (‘image’) or a color-filled contour plot (‘contourf’)

****kwargs** [dict] args passed on to matplotlib plot function (either contourf or imshow)

Returns

fig [pyplot object] matplotlib fig handle for the plot

ax [pyplot object] matplotlib ax handle for the plot

plot_series (*var*, *times=None*, *plot_type='image'*, *figsize=None*, ***kwargs*)
Plot .tec output for a series of time steps.

Parameters

var [str] variable to plot (e.g., ‘H+’)

times [list] time steps to plot, must be either components of self.times or self.output_times; defaults to all time steps

plot_type [str] whether to display an image (‘image’) or a contour-filled contour plot (‘contourf’)

figsize [tuple of int] figure size; default (12, 1.5 * # of timesteps)

****kwargs** [dict] args passed on to matplotlib’s imshow

Returns

fig [pyplot object] matplotlib fig handle for the plot

axes [list of pyplot objects] list of matplotlib axes handles for each subplot

CHAPTER

TWO

TIME SERIES OUTPUT

output.timeseries.**get_ts_coords**(*tsfile*)

Given a CrunchFlow time series ouput file, return the coordinate at which that time series was output.

Parameters

tsfile [str] filename containing timeseries output

Returns

coords [tuple of int] Coordinates of the form (x, y, z)

output.timeseries.**get_ts_duplicates**(*tsfile*)

Find duplicate columns in a time series file. Useful when a user specifies *time_series_print all* in CrunchFlow; the time series file includes primary species printed twice.

Parameters

tsfile [str] path to the time series file

Returns

columns [list] list of column headings without duplicates

nondup_indices [list] list of column indices without duplicates

class output.timeseries.**timeseries**(*tsfile*, *folder='.'*)

This is the timeseries class for working with CrunchFlow time series output files.

Examples

```
>>> ts = timeseries('Well1-1.txt')
>>> ts.convert_mgL()
>>> calcium = ts.df['Ca++']
>>> ts.plot('Ca++')
```

Attributes

coords [tuple of int] x, y and z coordinates of the time series

timeunit [str] time unit used in the CrunchFlow input

unit [str] Concentration units included in the file. Automatically set to the default CrunchFlow concentration units (mol/kgw)

species [list of str] list of aqueous species in the file

data [ndarray of float] Numpy array of all data. First col is the time step and remaining cols are species in the same order as self.species list

df [dataframe of float] Pandas dataframe of all data. Index is the time step and columns #are the aqueous species

Methods

convert_mgL(database='datacom.dbs', folder='.')	Convert time series concentrations from mol/kgw to mg/L (ppm).
plot(species, units='mg/L', **kwargs)	Plot the time series of one or more species.

`convert_mgL` (*database='datacom.dbs'*, *folder='.'*)

Convert time series concentrations from mol/kgw to mg/L (ppm). Note that this assumes that 1 kg water = 1 L water.

Parameters

database [str] name of the CrunchFlow database. The default is 'datacom.dbs'

folder [str] path to the database. The default is current directory.

Returns

None. Modifies timeseries object in place.

`plot` (*species*, *units='mg/L'*, ***kwargs*)

Plot the time series of one or more species.

Parameters

species [str or list of str] Either single species or list of species to be plotted

units [str] Concentration units to use for plotting. The default is 'mg/L'

****kwargs** [dict] keyword arguments passed to plt.subplots (e.g., figsize)

Returns

fig [pyplot object] figure handle for current plot

ax [pyplot object] axis handle for current plot

CHAPTER

THREE

UTILITIES

util.**correct_exponent**(*filename*, *folder='.'*, *verbose='med'*)

Correct triple digit exponents within a file. CrunchFlow has trouble outputting triple-digit exponents and omits the ‘E’. For example, ‘2.5582E-180’ prints as ‘2.5582-180’.

Parameters

filename [str] name of the file to be processed

folder [str] folder containing the file, either relative or absolute path

verbose [{‘med’, ‘high’, ‘low’}] Print each correction as it’s performed (‘high’), print total number of corrections (‘med’), or print nothing (‘low’). The default is ‘med’

Returns

None. Modifies the file in place.

util.**crunch_input_block**(*line*, *block*)

While reading a CrunchFlow input file, return the input block to which the current line belongs.

Parameters

line [str] current line in the input file

block [str] block to which the previous line belongs. Initialize as empty str

Returns

block [str] one of the default CrunchFlow blocks (e.g., “RUNTIME”) or an empty string if between blocks. If within a geochemical condition block, returns the name of the geochemical condition.

Examples

```
>>> block = ""
>>> with open(input_file, "r") as f:
>>>     for line in f:
>>>         block = crunch_input_block(line, block)
>>>
>>>         if block == "BOUNDARY_CONDITIONS":
>>>             print(line)
```

CHAPTER

FOUR

INDICES AND TABLES

- genindex
- modindex
- search

PYTHON MODULE INDEX

o

output.tec, 3
output.timeseries, 7

u

util, 9

INDEX

C

convert_mgL() (*output.timeseries.timeseries method*), 8
correct_exponent() (*in module util*), 9
crunch_input_block() (*in module util*), 9

E

extract() (*output.tec.tec method*), 4

G

get_tec_metadata() (*in module output.tec*), 3
get_tec_output_times() (*in module output.tec*), 3
get_ts_coords() (*in module output.timeseries*), 7
get_ts_duplicates() (*in module output.timeseries*), 7

M

module
    output.tec, 3
    output.timeseries, 7
    util, 9

O

outline() (*output.tec.tec method*), 4
output.tec
    module, 3
output.timeseries
    module, 7

P

plot() (*output.tec.tec method*), 5
plot() (*output.timeseries.timeseries method*), 8
plot_series() (*output.tec.tec method*), 5

T

tec (*class in output.tec*), 3
timeseries (*class in output.timeseries*), 7

U

util
    module, 9