pinch-analysis Documentation

Release 0.1.0

Andrew Hoetker, Emma Holle, James Taylor, Steve Wilson

Nov 11, 2019

CONTENTS:

1 pinch-analysis **1**

1.1 Getting Started . . . 1

1.2 Usage . . . 1

1.3 Testing . . . 2

1.4 Authors . . . 2

2 Modules **3**

2.1 Analysis module . . . 3

2.2 Intake module . . . 4

2.3 Plots module . . . 5

2.4 Report module . . . 6

3 Indices and tables **7**

Python Module Index **9**

Index **11**

CHAPTER

ONE

PINCH-ANALYSIS

Pinch-point technique for heat integration analysis in chemical plants.

1.1 Getting Started

The package `pinch-analysis` is defined by `setup.py`. To install, clone this repository, and run the following installation commands:

```
pip install -r requirements.txt
pip install -e .
```

1.2 Usage

After installation, the package provided the `pinch-analysis` command:

```
Usage: pinch-analysis [OPTIONS] DATAFILE REPORTFILE [OUTPUTDIR]

  Run script for `pinch-analysis`.

Options:
  --unitsystem TEXT  Unit system used in tables and graphs.
  -a                 Save all generated tables and figures to disk.
  --help             Show this message and exit.
```

For example, the following command will parse data from *design_table.xlsx*, format the output in English units, and save both the report and all included figures and tables to the *report* folder.

```
pinch-analysis -a --unitsystem english design_table.xlsx report.pdf report/
```

1.2.1 Input Data Format

Data is provided to the program as either a CSV file, or a sheet in an Excel workbook. An example of how the data must be formatted is given below. The units given in the second row can be written in any way such that Pint is able to parse them to valid quantities.

Stream	#	Supply Temperature	Target Temperature	Heat Capacity Flowrate
		degC	degC	kW/K
Compressor 1 out	3	159.2	45	101.2
Compressor 2 out	5	206.1	45	102
Reactor 1 out	12	240	45	219.5
Reactor 2 out	20	240	45	215.7
Mixed absorber effluent	29	52.1	45	597
Cold reactor 1 feed	10	106.7	240	216
Cold reactor 2 feed	16	30.3	240	215

1.3 Testing

Unit tests are provided in the `tests` directory. Running tests requires the `pytest` package.

1.4 Authors

Name	Contact	Github
Andrew Hoetker	ahoetker@asu.edu	ahoetker
Emma Holle	eholle@asu.edu	eholle123
James Taylor	jetayl14@asu.edu	notthesinger

CHAPTER

TWO

MODULES

2.1 Analysis module

`pinch.analysis.`**`classify_streams`**(*df: pandas.core.frame.DataFrame*) → pandas.core.series.Series

Classify each stream in a table as either HOT or COLD.

Parameters **`df`** – DataFrame containing stream data.

Returns Series corresponding to each stream.

`pinch.analysis.`**`composite_plots`**(*cold_df: pandas.core.frame.DataFrame*, *hot_df: pandas.core.frame.DataFrame*, *unit_system: str*, *outputdir: pathlib.Path*) → Dict[str, pathlib.Path]

Generate the following plots from the input data:

- Cold composite
- Hot composite
- Combined composite
- Grand composite

Parameters

- **`cold_df`** – DataFrame containing input data for cold streams.
- **`hot_df`** – DataFrame containing input data for hot streams.
- **`unit_system`** – Units for physical quantities.
- **`outputdir`** – Path to output folder for plots to be saved.

Returns None

`pinch.analysis.`**`cumulative_stream_enthalpy`**(*df: pandas.core.frame.DataFrame*, *unit_system: str*) → pandas.core.series.Series

Calculate cumulative stream relative enthalpies for all hot or cold streams.

Parameters

- **`df`** – DataFrame containing input data.
- **`unit_system`** – Units for physical quantities.

Returns Series containing cumulative enthalpies `pinch.analysis.`**`overall_analysis`**(*df: pandas.core.frame.DataFrame*, *unit_system: str*, *outputdir: pathlib.Path*) → Dict

Main function for performing analysis on input data.

Parameters

- **`df`** – DataFrame containing input data.
- **`unit_system`** – Units for physical quantities.
- **`outputdir`** – Path to output folder for plots to be saved.

Returns None

`pinch.analysis.`**`prob_design_table`**(*df: pandas.core.frame.DataFrame*, *unit_system: str*, *filename: pathlib.Path*) → None

Generate the problem design table.

Parameters

- **`df`** – DataFrame containing input data.
- **`unit_system`** – Units for physical quantities.
- **`filename`** – Path to file destination for problem design table.

Returns

`pinch.analysis.`**`tables`**(*df: pandas.core.frame.DataFrame*, *unit_system: str*, *outputdir: pathlib.Path*) → Dict[str, pathlib.Path]

Generate tables to be included in the pinch analysis report.

Parameters

- **`df`** – DataFrame containing input data.
- **`unit_system`** – Units for physical quantities.
- **`outputdir`** – Path to output folder for plots to be saved.

Returns Dictionary of filenames for generated tables.

2.2 Intake module

`pinch.intake.`**`attach_units`**(*column: pandas.core.series.Series*, *units: pint.quantity.build_quantity_class.<locals>.Quantity*) → numpy.array

Attach units to a DataFrame column.

Parameters

- **`column`** – DataFrame column or Series to have units attached.
- **`units`** – pint Quantity to be attached.

Returns Numpy array wrapped with the correct Quantity.

`pinch.intake.`**`df_with_units`**(*filepath: pathlib.Path*, *unit_system: str*, *sheet_name: str = None*) → pandas.core.frame.DataFrame

Take in a file containing the product design table, and produce a DataFrame with the correct units applied to relevant columns. The output DataFrame will contain only scalars, and is not labeled with units. This is because it has already been converted to the output unit system.

Parameters

- **`filepath`** – Path to data file to be read.

- **`sheet_name`** – optional sheet name for multiple-sheet Excel workbooks.

Returns DataFrame containing table data with units attached

`pinch.intake.`**`map_input_method`**(*filepath: pathlib.Path*) → Optional[Callable]

Determine which pandas method should be used to create a DataFrame.

Parameters **`filepath`** – path to data file to be read.

Returns pandas method suitable for data intake.

`pinch.intake.`**`parse_column_units`**(*filepath: pathlib.Path*, *sheet_name: str = None*) → Dict[str, pint.quantity.build_quantity_class.<locals>.Quantity]

Parse units for each column containing physical data in the data file.

Parameters

- **`filepath`** – Path to data file to be read.
- **`sheet_name`** – optional sheet name for multiple-sheet Excel workbooks.

Returns Dictionary of units for each input column

2.3 Plots module

`pinch.plots.`**`cold_composite`**(*enth: numpy.array*, *temp: numpy.array*, *show: bool = False*, *filename: pathlib.Path = None*) → None

Cold composite curve

Parameters

- **`enth`** – array of enthalpy values
- **`temp`** – array of cold temperatures
- **`show`** – display the generated plot using *pyplot.show*
- **`filename`** – file destination to save the figure

Returns None

`pinch.plots.`**`combined_composite`**(*enth_cold: numpy.array*, *enth_hot: numpy.array*, *temp_cold: numpy.array*, *temp_hot: numpy.array*, *show: bool = False*, *filename: pathlib.Path = None*) → None

Combined composite curve

Parameters

- **`enth_cold`** – array of enthalpy values corresponding to cold stream temperatures
- **`enth_hot`** – array of enthalpy values corresponding to hot stream temperatures
- **`temp_cold`** – array of cold temperatures
- **`temp_hot`** – array of hot temperatures
- **`show`** – display the generated plot using *pyplot.show*
- **`filename`** – file destination to save the figure

Returns None

`pinch.plots.`**`grand_composite`**(*enth: numpy.array*, *temp: numpy.array*, *show: bool = False*, *filename: pathlib.Path = None*) → None

Grand composite curve

Parameters

- **enth** – array of enthalpy values
- **temp** – array of temperatures
- **show** – display the generated plot using *pyplot.show*
- **filename** – file destination to save the figure

Returns None

`pinch.plots.`**`hot_composite`**`(`*enth: numpy.array*, *temp: numpy.array*, *show: bool = False*, *filename: pathlib.Path = None*`)` → None

Cold composite curve

Parameters

- **enth** – array of enthalpy values
- **temp** – array of hot temperatures
- **show** – display the generated plot using *pyplot.show*
- **filename** – file destination to save the figure

Returns None

`pinch.plots.`**`stream_matching`**`()` → None

Steam matching diagram I am still unsure how to create this diagram, so this is a pure stub with no parameters.

Returns

2.4 Report module

`pinch.report.`**`generate_report`**`(`*contents: Dict*, *filename: pathlib.Path*, *outputdir: pathlib.Path*`)` → None

Generate and save a PDF report of the pinch analysis.

Parameters

- **contents** – Dictionary containing the filenames of generated tables and figures.
- **filename** – Path to the destination where the report should be saved.
- **outputdir** – Path to the directory where tables and figures should be saved.

Returns None

CHAPTER

THREE

INDICES AND TABLES

- genindex
- modindex
- search

PYTHON MODULE INDEX

p

pinch.analysis, 3
pinch.intake, 4
pinch.plots, 5
pinch.report, 6

INDEX

A

`attach_units()` (*in module pinch.intake*), 4

C

`classify_streams()` (*in module pinch.analysis*), 3
`cold_composite()` (*in module pinch.plots*), 5
`combined_composite()` (*in module pinch.plots*), 5
`composite_plots()` (*in module pinch.analysis*), 3
`cumulative_stream_enthalpy()` (*in module pinch.analysis*), 3

D

`df_with_units()` (*in module pinch.intake*), 4

G

`generate_report()` (*in module pinch.report*), 6
`grand_composite()` (*in module pinch.plots*), 5

H

`hot_composite()` (*in module pinch.plots*), 6

M

`map_input_method()` (*in module pinch.intake*), 5

O

`overall_analysis()` (*in module pinch.analysis*), 3

P

`parse_column_units()` (*in module pinch.intake*), 5
`pinch.analysis` (*module*), 3
`pinch.intake` (*module*), 4
`pinch.plots` (*module*), 5
`pinch.report` (*module*), 6
`prob_design_table()` (*in module pinch.analysis*), 4

S

`stream_matching()` (*in module pinch.plots*), 6

T

`tables()` (*in module pinch.analysis*), 4