pinch-analysis Documentation

Release 0.1.0

Andrew Hoetker, Emma Holle, James Taylor, Steve Wilson

Nov 11, 2019

CONTENTS:

CHAPTER

ONE

PINCH-ANALYSIS

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

1.1 Getting Started

No package is provided. No docker image is provided. Simply clone https://github.com/ahoetker/pinch-analysis.git, and run `pip install -r requirements.txt` to create the Python environment.

1.2 Usage

In this stage of development, the only target is a run script. Run `python main.py`.

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 Temperatur e	Target Temperatur e	Heat Capacity Flowrate	Enthalpy
		degC	degC	kW/K	MJ/hour
Compressor 1 out	3	159.2	45	101.2	41605.3
Compressor 2 out	5	206.1	45	102	59155.9
Reactor 1 out	1 2	240	45	219.5	15408 9
Reactor 2 out	2 0	240	45	215.7	15142 1.4
Mixed absorber effluent	2 9	52.1	45	597	15259.3
Cold reactor 1 feed	1 0	106.7	240	216	-1036 54.1
Cold reactor 2 feed	1 6	30.3	240	215	-1623 07.8

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

1.4.1 plots

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

1.4.2 intake

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

CHAPTER

TWO

INDICES AND TABLES

- genindex
- modindex
- search

PYTHON MODULE INDEX

p

INDEX