ChemicalReactorNetwork

* Inlet\_id: np array
  + i.a. zeros array, same as grid structure
  + array is filled so that the cells that connect to an inlet are marked with the number of the inlet they are connected to – algorithm checks if cell is within an inlet connection area and if so assigns number to Inlet\_id5[i,j]
* Outlet\_id: np array
  + i.a. zeros array, same as grid structure
* reactors: list
  + i.a. empty list []
* mixers: list
  + i.a. empty list []
* splitters: list
  + i.a. empty list []
* crn\_inlets: list
  + i.a. empty list []
* mass\_flows: array (n x n, where n is number of clusters)
  + given as input in init, is the matrix describing massflow between clusters (from input\_parser function)

Algorithm for generation of mass flows between clusters matrix

mz and my are generated 🡪 describe the massflow through each cell in y or z direction

1. Area of cells in y or z direction in matrix Az or Ay ate calculated
2. Mass flow through each cell calculated with area, velocity and density – each property is cell specific
3. A zero-matrix with n X n is generated, where n is the total number of clusters: mass\_flows
4. Mz and my are looked at cellwise: If the value is positive the cell next to the current cell in the positive direction (y or z) is checked to see whether the cluster assignment is the same. If the value is negative this is done with the cell next to the current cell in the negative direction. If adjacent cell is not within the same cluster the mass flow value is added to the mass\_flows matrix at the position (i,j). (i is the cluster the current cell is assigned to and j is the cluster the adjacent cell is assigned to)

Algorithm for connection of reactors

1. mass\_flows array/ matrix is created which describes the mass flow from one cluster to another (value at position i,j in matrix describes mass flow from reactor / cluster I to reactor / cluster j)
2. connections between reactors are simplified:
   * + if the massflow between two clusters is below 10% of the maximum massflow between any clusters then this connection is checked again
     + starting with connection between reactors with lowest massflow: connection between the reactors i and j is removed and mass flow value is set to zero in mass\_flows array (copy of mass\_flows array)
     + if any of the following conditions are true then reactor i and j are reconnected:
       - rector i is not connected to an output and none of its following connected reactors are connected to an output
       - reactor i does not have any outlets
       - rector j is not connected to an input and none of its preceding connected reactors are connected to an output
       - reactor j does not have any inlets
3. The next lowest massflow value in mass\_flows matrix is found and this connection is checked according to step 2
4. This is continued until the lowest non-zero massflow in matrix is equal to or bigger than 10% of max massflow

Clustering Algorithm

* In CRNBuilder.py the cluster\_data\_linkage() function is imported which returns the cluster\_id 🡪 np array showing which cell is assigned to which cluster

ClusterMap is initialized with one cluster per cell

Clustering of cold cells

* A threshold temperature is given in the input file, is case specific
* A loop is run for up to as many times as number of cells
* Loop ends when no cluster with max temp below the threshold temp has a neighbouring cluster with max temp below the threshold temp
* In the loop each cluster is looked at and if the max temp of this cluster is below the threshold temp then a target is set for this cluster
  + The target is the first cluster in the neighbours list that has a max temp below the threshold temp
  + The original cluster and the target cluster are merged
  + Repeated until there not any target is found for any cluster, then overall loop is ended

General Clustering based on temperatures

* Loop is created that runs maximum max\_iter times, where max\_iter is the number of cells, loop also ends when the number of clusters has reached the number of objective reactors
* For each non-empty cluster the neighbouring cluster with the minimum distance is found and set as target of current cluster – the distances found for each cluster are put into an array
* The indices for values in this array are found for the following values:
  + If number of clusters larger than 10\* objective number of reactors
    - All values where distance is larger than (1+ε)\*min value in the array
  + Else
    - Minimum value in array
* The clusters for which the indices were found are merged with their target (target cluster is cleared and cells added to origin cluster)

🡪 cluster\_id is genenerated

Generating CRN Outlets

In the input file the outlets existing for the case must be described:

e.g.

@Outlets  
outlet: north, 0, 90e-3  
//

* + - * + North describes which side of grid the outlet is at
        + 0 and 90e-30 state the coordinates where outlet starts and ends, these are y or z coordinates depending on which side outlet is on
        + The list outlets is generated with input parser:

Outlets = [ [outlet\_name, outlet\_data] , …]

Outlet data = [ int between 0 and 3 depending on location of outlet , coordinate1, coordinate2 ]