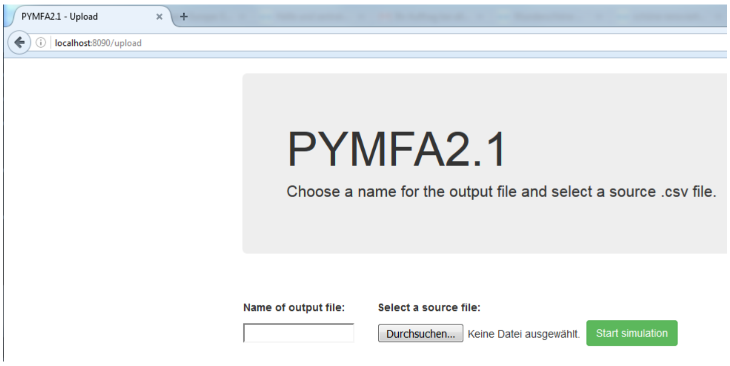
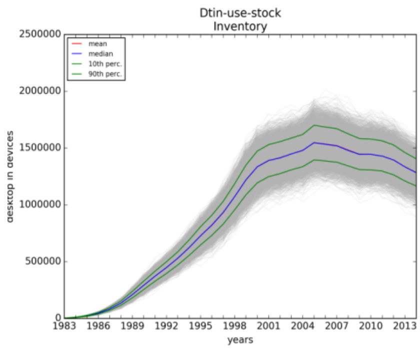
PYMFA 2.1

A probabilistic dynamic material flow analysis tool





User manual

Esther Thiébaud

Technology and Society Lab, Empa

Original Version: June 2017
Updated Version: February 2019

Table of Contents

1 Introduction

The pymfa tool is an open source tool, written in Python 3 (https://www.python.org), utilizing the numpy, scipy, and matplotlib library. The core of the tool can be used as a Python library and provides the necessary functionality to run ***deterministic or probabilistic dynamic material flow analyses (MFAs)*** and evaluate dynamic MFA with ***statistical entropy analysis (SEA)***. The tool is run through a command line interface and an interactive web application that can also be run locally. Source files containing the model descriptions can be uploaded through the web interface, upon which the tool calculates the time series of resulting stocks and flows by means of Monte Carlo simulation, visualizes them and offers the resulting data as a CSV download.

This manual provides all necessary information how to run the tool, create a source file, run probabilistic dynamic MFAs and evaluate MFAs with SEA.

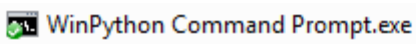
2 Prerequisites

In order to run the tool, you need:

- Python 3 installed on computer, including the numpy, scipy, and matplotlib libraries
    - For Windows users not familiar with python, we recommend the installation of WinPython-64bit-3.4.3.7, where all necessary libraries are already pre-installed
    - For Mac OS users, we recommend the installation of Anaconda distribution, where all necessary libraries are already pre-installed
- Pymfa 2.1 stored as folder on computer
    - Download code from https://bitbucket.org/Xeelk/pymfa2/src
- Web browser
- Software to create .CSV files, e.g. MS Excel, Open Office Calc etc.

3 Start of the Application

3.1 Windows

1. Go to your command line interface. e.g. 

2. Change the directory to the directory where you have stored the pymfa2.1 folder by typing: ***cd C:\Users\xxx\pymfa.2.1***
   press enter
3. Start the application by typing in your console: ***python.exe app.py***
   press enter
   the server gets started
    a. You have various options, how to start the application:
        - ***python.exe app.py***
          default to start the application for dynamic MFA simulation
        - ***python.exe app.py --plot***
          additionally, plots are created
        - ***python.exe app.py --entropy=0***

additionally to dynamic MFA, the statistical entropy analysis is executed. The intermediate results of the SEA are written in the console for all years

- ***python.exe app.py --entropy=2012***
  additionally to dynamic MFA, the statistical entropy analysis is executed. The intermediate results of the SEA are written in the console for the indicated year
- ***python.exe app.py --entropy=2012 –plot***
  additionally, plots are created, the statistical entropy analysis is executed and the intermediate results of the SEA are written in the console for the indicated year

An example is shown in Figure 1:

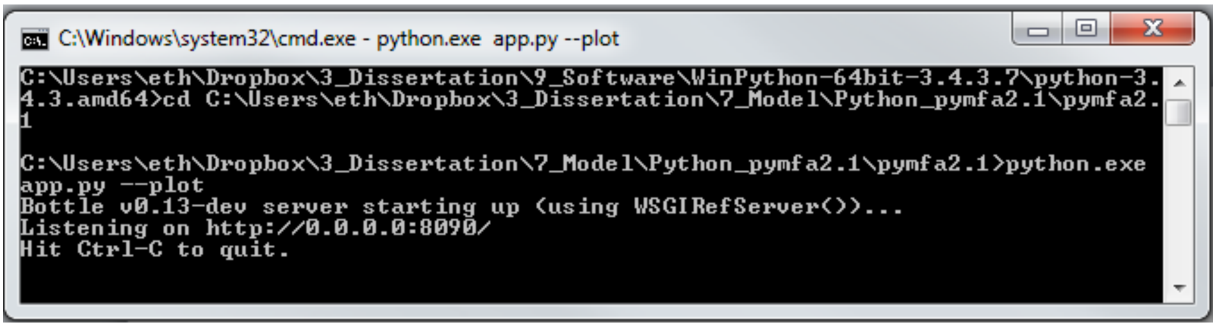


Figure 1: Command line interface

3.2 Mac OS

1. Go to your terminal 

2. Change the directory to the directory where you have stored the pymfa2.1 folder by typing: ***cd /Users/xx/xx/pymfa2.1***
   press enter
3. Start the application by typing in your console: ***python app.py***
   press enter
   the server gets started
   a. You have various options, how to start the application:
      - ***python app.py***
        default to start the application for dynamic MFA simulation
      - ***python app.py --plot***
        additionally, plots are created
      - ***python app.py --entropy=0***
        additionally to dynamic MFA, the statistical entropy analysis is executed. The intermediate results of the SEA are written in the console for all years
      - ***python app.py --entropy=2012***
        additionally to dynamic MFA, the statistical entropy analysis is executed. The intermediate results of the SEA are written in the console for the indicated year
      - ***python app.py --entropy=2012 –plot***

additionally, plots are created, the statistical entropy analysis is executed and the intermediate results of the SEA are written in the console for the indicated year

An example is shown in Figure 2:

Figure 2: Terminal

3.3 Further Steps for all Operating Systems

4. Go to your browser, insert http://localhost:8090
5. The login appears: login: admin, password: notasecret (Figure 3)

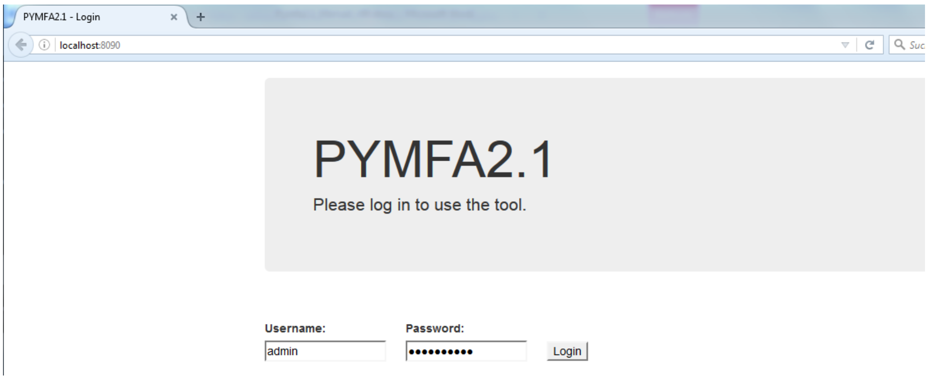


Figure 3: Login of interactive web application

6. The interactive web application appears (Figure 4)
7. Now you can search for your source CSV file, upload it and start the simulation
8. From the interface, you can download the results CSV file, the input CSV file or the plots
   The result CSV file, the input CSV file and the plots are also stored in the pymfa2.1 folder on your computer.

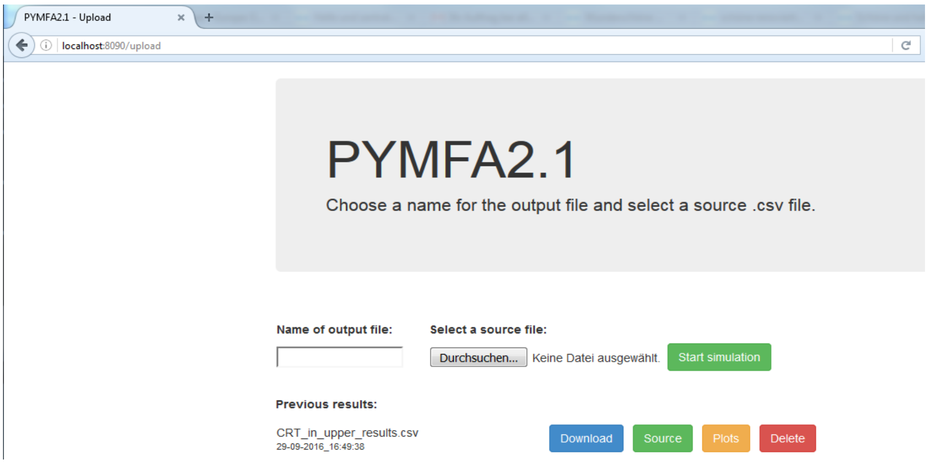


Figure 4: Interactive web application for pymfa2.1, run locally under localhost:8090.

4 Dynamic Material Flow Analysis

4.1 Background

The temporal analysis of material cycles is often based on a dynamic material flow analysis (MFA) approach. An MFA consists of processes and flows.[1] Processes may transfer, transform or stock material. If materials is transferred or transformed, the outflow is calculated from inflows and transfer coefficients (equation 1). The sum of all outflows is equal to the sum of all inflows (equation 2)

$$Outflow_j = \sum_{i=1}^{m} Inflow_i \cdot k_j \quad (1)$$

$$\sum_{i=1}^{m} Input_i = \sum_{j=1}^{p} Output_j \quad (2)$$

k_j: Transfer coefficient
m: total number of inflows
p: total number of outflows

The development of stocks and flows are calculated based on inflow or stock data and the product lifetime (equation 3 and 4). For more details, please refer to Müller et al.[2]

$$S[n] = (inflow[n] - outflow[n]) + S[n-1] \quad (3)$$

$$Outflow[n] = \sum_{m=-\infty}^{\infty} Inflow[n-m] \cdot f[m] \quad (4)$$

n: year of calculation
f(m): lifetime distribution function

Figure 5 gives an example of a basic MFA system of laptops in the use and recycling phase with a stock in the use phase. All other processes only transfer or transform material. This example will be used throughout this manual.

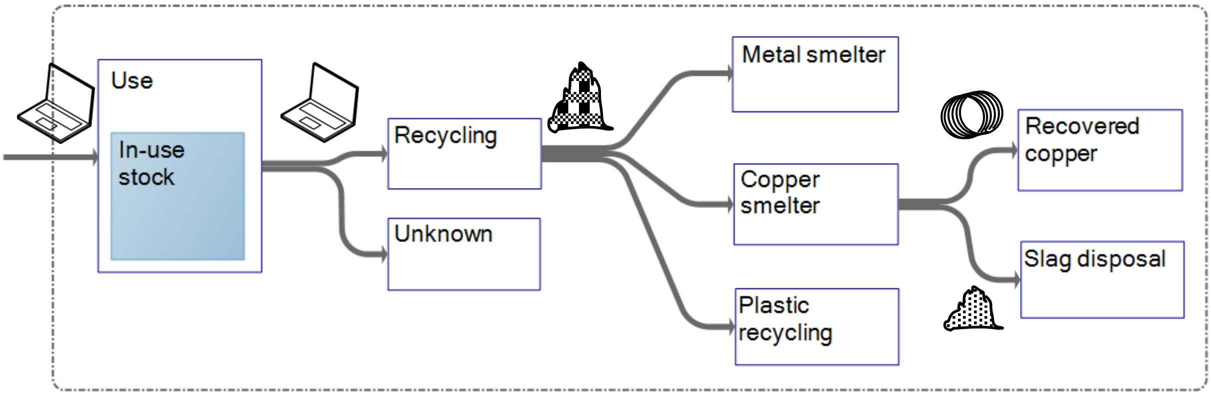


Figure 5: Example of a basic MFA system of laptops in the use and recycling phase.

In pymfa, processes are called 'nodes', flows between processes are defined as 'transfers'. Generally, every transfer has a 'source node' and a 'target node'. Every node is linked to a material and a unit as shown in Figure 6.

Figure 6: Source node, transfer and target node

There are currently five transfer types defined:

- ***Inflow:*** Inflows are used to provide the system with raw materials. This is the only transfer type that only has no source node but only a target node. Each system has to start with at least one inflow.
- ***Rate:*** A rate transfer simply takes the amount of material present at the source node, multiplies it with a transfer coefficient (TC) and forwards the resulting amount of material to the target node. The sum of all transfer coefficients of transfers leaving a given source node for a single material should be 1 if all material should be forwarded from the node. If the sum is smaller then 1, a stock is created in the source node.
- ***Fraction:*** Fraction transfers behave exactly like rate transfers, but they require different materials in the source node and destination node. This transfer type is used to split a material from a source node into multiple fractions leaving the node. For example, a laptop in a recycling process is split into iron, aluminium, copper, plastics, cables etc.
- ***Conversion:*** Conversion transfers convert the unit of the source node to a different unit of the target node, by multiplying the material present at the source node with a factor and forwarding the resulting amount of material to the target node. This makes it possible, for example, to convert 'pieces' to 'kg'.
- ***Delay:*** a delay transfer applies, when material is not directly transferred to the target node, but kept in stock for some time. For example, laptops are used for an average product lifetime of 5 years, thus within the use phase, a stock of laptops is accumulated. This means that materials which flow into a node in a particular year are distributed over several years when

leaving the node, according to the lifetime distribution. Within the source file, a stock is created.

Stocks and sinks do not have to be explicitly defined. If the outflow of a node is smaller than the inflow, a stock is created. If a node has only an inflow but no outflow, a sink is created.

4.2 Source CSV file

Within the source CSV file, all information regarding the MFA system is stored. The program only reads rows that have a value in the first column. This enables user to include comments etc. in the source file. An example of a source CSV file related to the example shown in Figure 5 is presented in Table 2.

In the first six rows, basic information is provided according to Table 1.

Table 1: Basic information provided in the source file

Row	Value column 1	Value column 2 (Example)	Comment
1	Runs	10000	Indicate the number of runs for the Monte Carlo simulation
2	Periods	10	Indicate for how many periods in the dynamic MFA should be calculated. If you leave it blank, the number of periods is read from the time series data inserted below
3	Median	yes/no	Indicate whether you want to calculate the median or not. The mean is always calculated
4	Percentile	10|90	Indicate which percentiles should be calculated (numbers between 1 and 100). Various percentiles should be separated with a '|'
5	Plots	Stock|storage	Indicate from which nodes plots should be created. For each node, plots of the inflow, outflow and stocks (if applicable) are created. Separate various nodes with a '|'
6	entropyHmax	value/blank	For more information, see Chapter 5.

Table 2: Example of source CSV file for dynamic MFA. Only the first two time periods are shown.

	A	B	C	D	E	F	G	H	I	J	K
1	Runs:	10000									
2	Periods:										
3	Median:	yes									
4	Percentiles:	10|90									
5	Plots:	In-use_stock|Recycling_kg									
6	entropyHmax:										
7											
8	**Transfer Type**	**Source Node**	**Source Material**	**Source Unit**	**Target Node**	**Target Material**	**Target Unit**	**Stages**	**Description**	**1990**	**1991**
9	Inflow				In-use_stock	Laptops	Devices		Inflow of new Laptops	stoch|normal|17226,3445	stoch|normal|27562,5512
10	Delay	In-use_stock	Laptops	Devices	Stockoutflow	Laptops	Devices		Stock outflow based on service lifetime	fix|1|1|weibull|2.1,5.9|0	fix|1|1|weibull|2.1,5.9|0
11	Rate	Stockoutflow	Laptops	Devices	Recycling	Laptops	Devices		Stock outflow to recycling	stoch|triangular|0.7,0.8,0.9|2	stoch|triangular|0.7,0.8,0.9|2
12	Rate	Stockoutflow	Laptops	Devices	Unknown	Laptops	Devices		Stock outflow to other disposal pathways	stoch|triangular|0.1,0.2,0.3|1	stoch|triangular|0.1,0.2,0.3|1
13	Conversion	Recycling	Laptops	Devices	Recycling_kg	Laptops	Kg		Conversion from devices to kg	stoch|triangular|1.8,3.1,4.8|0	stoch|triangular|1.8,3.1,4.8|0
14	Fraction	Recycling_kg	Laptops	kg	Metal_Smelter	Iron&Alu	kg		Fraction of iron and aluminium from laptops to metal smelter	stoch|uniform|0.46,0.50|0	stoch|uniform|0.46,0.50|0
15	Fraction	Recycling_kg	Laptops	kg	Copper_Smelter	Copper	kg		Fraction of copper from laptops to copper smelter	stoch|uniform|0.007,0.06|0	stoch|uniform|0.007,0.06|0
16	Fraction	Recycling_kg	Laptops	kg	Copper_Smelter	PWB	kg		Fraction of PWBs from laptops to iron smelter	stoch|uniform|0.005,0.015|0	stoch|uniform|0.005,0.015|0
17	Fraction	Recycling_kg	Laptops	kg	Plastic_Recycling	Plastics	Kg		Fraction of plastics from laptops to plastics recycling	stoch|uniform|0.17,0.47|0	stoch|uniform|0.17,0.47|0
18	Rate	Copper_Smelter	Copper	kg	Recovered_copper	Copper	Kg		Flow of recovered copper from copper smelter	fix|0.95|0	fix|0.95|0
19	Rate	Copper_Smelter	PWB	kg	Recovered_copper	PWB	Kg		Flow of recovered copper from copper smelter, from PWBs	fix|0.95|0	fix|0.95|0
20	Rate	Copper_Smelter	Copper	kg	Slag_Landfill	Copper	Kg		Flow of copper from copper smelter to landfill disposal	fix|0.05|0	fix|0.05|0
21	Rate	Copper_Smelter	PWB	kg	Slag_Landfill	PWB	Kg		Flow of copper from copper smelter to landfill disposal, from PWBs	fix|0.05|0	fix|0.05|0

After the basic information, the title line for the data table follows. From left to right, the following entries are made: transfer type, source node, source material, 'source unit', 'target node', 'target material', 'target unit', 'stages', 'description'. The stages column is only used for the SEA (Chapter 5). In the description column, every transfer can be described in more detail. In the following columns the time periods are entered ascending from left to right. The minimum number of time periods is two.

After the title line, each row represents a transfer. In the first column, the transfer type is defined as described above. The system always starts with at least one inflow. For the inflow, only a target node, target material and a target unit is defined. Below the time periods in the title line, the deterministic or probabilistic input data is entered for each time period.

In the next row, the target node of the inflow becomes the source node of the next transfer etc.

4.3 Deterministic and Probabilistic Input Data

Input data for each time period for the ***inflow, conversion, rate and fraction transfer types*** are either fix (for deterministic data input), normally distributed, uniformly distributed or triangularly distributed, according to Figure 7.

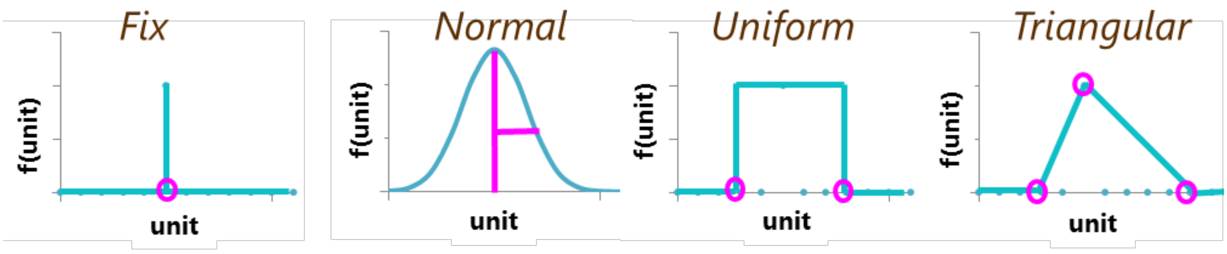


Figure 7: Possible distributions for inflow, conversion, rate and fraction transfer types

Table 3 summarizes the required parameters for each distribution and gives examples how to enter the input data for different transfer types.

Table 3: Examples of deterministic and probabilistic input data. The red value indicates the prioritization for normalization of the transfer coefficient in the Monte Carlos analysis

Transfer type	Fix	Normal	Uniform	Triangular
Required parameters	*value*	*mean, standard deviation*	*minimum, maximum*	*minimum, mode, maximum*
Inflow	fix|100	stoch|normal|100,10	stoch|uniform|50,150	stoch|triangular|50,100,150
Conversion	fix|3|**0**	stoch|normal|3,0.5|**0**	stoch|uniform|2.5, 3.5|**0**	stoch|triangular|2.5,3,3.5|**0**
Rate	fix|0.6|**0**	stoch|normal|0.6,0.1|**0**	stoch|uniform|0.5,0.7|**0**	stoch|triangular|0.5,0.6,0.7|**0**
Fraction	fix|0.6|**0**	stoch|normal|0.6,0.1|**0**	stoch|uniform|0.5,0.7|**0**	stoch|triangular|0.5,0.6,0.7|**0**

As already stated above, the sum of all TCs needs to be one. With probabilistic data input and Monte Carlo analysis that randomly selects values from each distribution in each run; the sum needs to be normalized in order to fulfill this requirement.

The last value for conversion, rate and fraction (**in red in Table 3**) gives the priority for normalization. The coefficient with the highest priority is not adjusted; the one with the lowest priority is most adjusted. If all coefficients have the same priority, all are equally adjusted.

Note: For the conversion transfer type, this prioritization would not be necessary. However, the way the tool is programmed right now, this information is required. We thus recommend using a zero (0).

For the **delay transfer type**, lifetime distribution functions can be either modeled as a fix value, a Weibull distribution, a list or a random list (Figure 8). If 'fix' is selected, the inflow is transferred to the target node after a constant lifetime. If 'rand' or 'list' are selected, for each period the share of the inflow that is transferred to the target node is selected from a list. For 'list' the rate is selected sequentially from the list, for 'rand' the rate is selected randomly from the list. If 'Weibull' is selected, the rates are selected according to a Weibull distribution function. This function is taken from scipy.stats and can be modelled with two or three parameters. Table 4 summarizes the required parameters for each distribution and gives examples for different transfer types. Additionally, the last value of the delay transfer type indicates by how many periods the first release is delayed (**indicated in red in Table 4**)

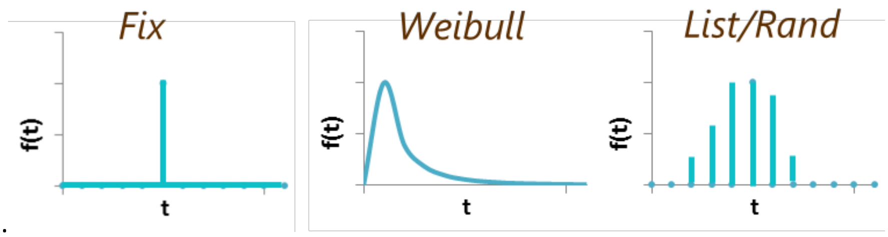


Figure 8: Possible distributions for delay transfer type

Table 4: Examples of deterministic and probabilistic input data for the delay transfer type. The values in red indicate by how many periods the first release is delayed.

Transfer type	**Fix**	**Weibull**	**List**	**Random**
Required parameters	*value*	*α: shape parameter* *β: scale parameter* *γ: location parameter*	*List of values, selected sequentially*	*List of values, selected randomly*
Delay	fix|1|0|fix|5 |**0**	fix|1|0|weibull|2.1,5.9,0.4 |**0**	fix|1|0|list|0.2,0.4,0.3 |**0**	fix|1|0|list|0.2,0.4,0.3 |**0**
		fix|1|0|weibull|2.1,5.9 |**0**		

TC and delays can be directly combined. For example:

stoch|triangular|0.1,0.5,0.7|0|weibull|2.10,5.90|0

As the setup of a source file can be a tedious task, we recommend to using scripts to transform existing data to the stochastic input data. Two examples of python scripts can be found in the appendix of this document.

After finishing the setup of the source file, it should be saved as a CSV file. From the interactive web application, the source file can be uploaded and the simulation can be run.

Note: When running the application, sometimes an error message appears that after the last time period in the CSV file, a further integer is expected.

We recommend to deleting all columns after the last time period column and save the CSV again. This usually helps to solve this problem.

4.4 Result CSV File and Plots

The results can be downloaded as a CSV file. They can be either downloaded from the interactive web application (blue button in Figure 4) or directly found in the pymfa2.1\analysis\ folder on your computer, where for every simulation, a folder including the date and the time is created.

The results file includes the mean, the median, and the percentiles for each transfer, as indicated in the basic information in the source file. An example of a results CSV file is shown in Table 5 and 6. The first part of the results file includes all transfers. In the second part, all stocks and sink are calculated. **ATTENTION:** the stocks and sinks in the result CSC file do not appear in the same order as in the input file – their order might change for every simulation.

If plots were created, the can be downloaded from the interactive web application (yellow button in Figure 4) or directly found in the pymfa2.1\analysis\ folder on your computer. The plots of a node include its inflow, stock (inventory) and outflow, if applicable (Figure 9). Each plot shows the mean, median and the percentiles as indicated in the basic information. Currently, the appearance of the plots cannot be changed. This would have to be done in the source code directly.

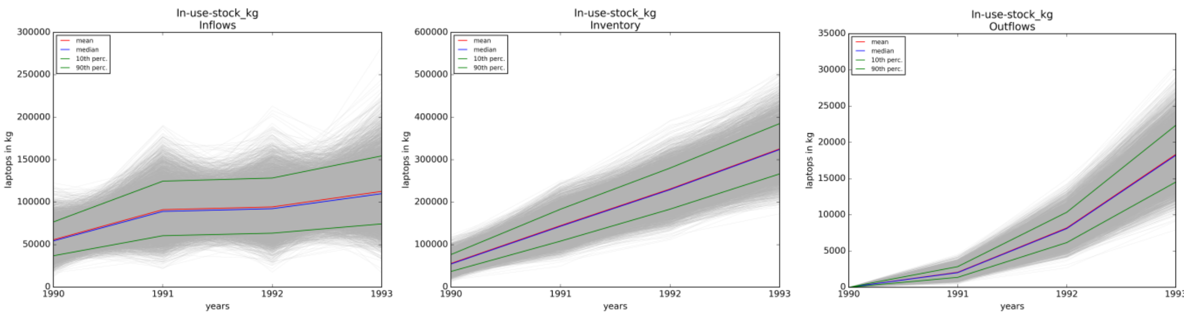


Figure 9: Example of plots

Table 5: Example of result CSV file for dynamic MFA

Runs:	10000												
Periods:	4												
Transfer Type	Source Node	Source Material	Source Unit	Destination Node	Destination Material	Destination Unit	Stages	Description		1990	1991	1992	1993
Delay	In-use_stock	Laptops	Devices	Stockoutflow	Laptops	Devices		Stock outflow based on serv	mean:	0	641	2'508	5'584
									median:	0	640	2'503	5'586
									10th perc.:	0	475	2'048	4'709
									90th perc.:	0	805	2'968	6'443
Rate	Stockoutflow	Laptops	Devices	Recycling	Laptops	Devices		Stock outflow to recycling	mean:	0	512	2'007	4'466
									median:	0	511	1'996	4'454
									10th perc.:	0	377	1'622	3'713
									90th perc.:	0	650	2'410	5'231
Rate	Stockoutflow	Laptops	Devices	Unknown	Laptops	Devices		Stock outflow to other dispo	mean:	0	128	501	1'118
									median:	0	126	492	1'106
									10th perc.:	0	82	342	777
									90th perc.:	0	179	669	1'470
Conversion	Recycling	Laptops	Devices	Recycling_kg	Laptops	Kg		Conversion from devices to	mean:	0	1'654	6'496	14'465
									median:	0	1'612	6'373	14'205
									10th perc.:	0	1'081	4'532	10'267
									90th perc.:	0	2'289	8'604	19'007
Fraction	Recycling_kg	Laptops	kg	Metal_Smelter	Iron&Alu	kg		Fraction of iron and aluminiu	mean:	0	951	3'738	8'324
									median:	0	919	3'636	8'117
									10th perc.:	0	605	2'536	5'707
									90th perc.:	0	1'340	5'061	11'231
Fraction	Recycling_kg	Laptops	kg	Copper_Smelter	Copper	kg		Fraction of copper from lap	mean:	0	66	259	579
									median:	0	61	243	547
									10th perc.:	0	23	92	200
									90th perc.:	0	116	448	995
Fraction	Recycling_kg	Laptops	kg	Copper_Smelter	PWB	kg		Fraction of PWBs from lapto	mean:	0	20	78	172
									median:	0	18	74	161
									10th perc.:	0	10	42	93
									90th perc.:	0	31	120	265
Fraction	Recycling_kg	Laptops	kg	Plastic_Recycling	Plastics	Kg		Fraction of plastics from lap	mean:	0	617	2'421	5'391
									median:	0	589	2'336	5'230
									10th perc.:	0	370	1'536	3'464
									90th perc.:	0	906	3'424	7'533
Rate	Copper_Smelte	Copper	kg	Recovered_coppe	Copper	Kg		Flow of recovered copper fr	mean:	0	62	246	550
									median:	0	58	231	520
									10th perc.:	0	21	87	190
									90th perc.:	0	110	426	945
Rate	Copper_Smelte	PWB	kg	Recovered_coppe	PWB	Kg		Flow of recovered copper fr	mean:	0	19	74	163
									median:	0	18	70	153
									10th perc.:	0	10	40	88
									90th perc.:	0	30	114	252
Rate	Copper_Smelte	Copper	kg	Slag_Landfill	Copper	Kg		Flow of copper from coppe	mean:	0	3	13	29
									median:	0	3	12	27
									10th perc.:	0	1	5	10
									90th perc.:	0	6	22	50
Rate	Copper_Smelte	PWB	kg	Slag_Landfill	PWB	Kg		Flow of copper from coppe	mean:	0	1	4	9
									median:	0	1	4	8
									10th perc.:	0	1	2	5
									90th perc.:	0	2	6	13

Note: If the inflow should appear in the results file, a **fixed rate transfer** has to be included between the inflow and the first target node. In our example, we introduce a transfer node 'laptop_sales', from which a fix transfer goes to the in-use stock. The inflow resulting from the Monte-Carlo simulation is then included in the result file.

The following source CSV file:

Transfer Type	Source Node	Source Material	Source Unit	Target Node	Target Material	Target Unit	Stages	Description	1990	1991
Inflow				Laptop_sale	Laptops	Devices		Inflow of new Laptops	stoch|normal|17226,3445	stoch|normal|27562,5512
Rate	Laptop_sale	Laptops	Devices	In-use_stock	Laptops	Devices	1	Fix rate transfer for inflow for SEA	fix|1|1	fix|1|1
Delay	In-use_stock	Laptops	Devices	Stockoutflow	Laptops	Devices	2|3|4	Stock outflow based on service lifetime	fix|1|1|weibull|2.1,5.9|0	fix|1|1|weibull|2.1,5.9|0
Rate	Stockoutflow	Laptops	Devices	Recycling	Laptops	Devices	2	Stock outflow to recycling	stoch|triangular|0.7,0.8,0.9|2	stoch|triangular|0.7,0.8,0.9|2
Rate	Stockoutflow	Laptops	Devices	Unknown	Laptops	Devices	2|3|4	Stock outflow to other disposal	stoch|triangular|0.1,0.2,0.3|1	stoch|triangular|0.1,0.2,0.3|1
Conversion	Recycling	Laptops	Devices	Recycling_kg	Laptops	Kg		Conversion from devices to kg	stoch|triangular|1.8,3.1,4.8|0	stoch|triangular|1.8,3.1,4.8|0

results in this results CSV file:

Transfer Type	Source Node	Source Material	Source Unit	Destination Node	Destination Material	Destinatio n Unit	Stages	Description		1990	1991
Rate	Laptop_sale	Laptops	Devices	In-use_stock	Laptops	Devices	1	Fix rate transfer for inflow for SEA	mean:	17238.9057	27512.4394
									median:	17203.4263	27495.241
									10th perc.:	12789.6971	20444.9983
									90th perc.:	21666.8094	34520.7697
Delay	In-use_stock	Laptops	Devices	Stockoutflow	Laptops	Devices	2|3|4	Stock outflow based on service lifetime	mean:	0	640.884011
									median:	0	639.565006
									10th perc.:	0	475.477535
									90th perc.:	0	805.498445

Table 6: Example of result CSV file, stocks and sinks

Node Type	Node Name	Material	Unit		1990	1991	1992	1993
Sink	Slag_Landfill	Copper	Kg	mean:	0	3.28858321	16.2287575	45.1710458
				median:	0	3.03208331	15.3342873	43.4782291
				10th perc.	0	1.12697668	7.54826102	23.9198198
				90th perc.	0	5.78338881	26.1457821	68.2919991
Sink	Slag_Landfill	PWB	Kg	mean:	0	0.98979267	4.89420373	13.4733099
				median:	0	0.92297956	4.67518636	13.045878
				10th perc.	0	0.50562641	2.93139702	8.79602621
				90th perc.	0	1.56281634	7.14207647	18.808938
Sink	Metal_Smelter	Iron&Alu	kg	mean:	0	951.372479	4689.34074	13013.2727
				median:	0	918.929752	4596.63381	12833.2184
				10th perc.	0	605.234568	3338.6144	9766.00734
				90th perc.	0	1339.74881	6166.91058	16552.1804
Sink	Plastic_Recyclin	Plastics	Kg	mean:	0	617.388664	3038.25599	8429.29317
				median:	0	589.284085	2945.82318	8284.39219
				10th perc.	0	369.683759	2066.33457	6079.80024
				90th perc.	0	905.849414	4141.63182	10970.5428
Stock	In-use_stock	Laptops	Devices	mean:	17238.9057	44110.461	70059.9518	98505.4359
				median:	17203.4263	44160.9177	70047.0597	98549.3894
				10th perc.	12789.6971	35763.1296	59411.8447	85231.2504
				90th perc.	21666.8094	52347.9512	80776.0444	111844.243
Sink	Unknown	Laptops	Devices	mean:	0	128.433323	629.020201	1746.91211
				median:	0	125.788453	621.117342	1730.41623
				10th perc.	0	82.3100134	450.753495	1323.54387
				90th perc.	0	178.772577	819.927925	2194.3705
Sink	Recovered_cop	Copper	Kg	mean:	0	62.483081	308.346392	858.249869
				median:	0	57.609583	291.351458	826.086352
				10th perc.	0	21.4125569	143.416959	454.476577
				90th perc.	0	109.884387	496.76986	1297.54798
Sink	Recovered_cop	PWB	Kg	mean:	0	18.8060606	92.9898709	255.992888
				median:	0	17.5366117	88.8285409	247.871682
				10th perc.	0	9.60690188	55.6965434	167.124498
				90th perc.	0	29.6935104	135.699453	357.369823

5 Statistical Entropy Analysis

5.1 Background

The SEA is a tailor-made evaluation method for MFA and can be directly applied to an MFA database.[1] It is used to measure the ability of a system to dilute or concentrate a substance and determine which processes are responsible for it. This helps to visualize and better understand the metabolism of anthropogenic systems.[3]

The SEA is connected to an MFA based on the system structure. The whole system transfers the input step by step, with each step assigned as a "stage". The MFA system (Figure 5) is thus transferred to the system depicted in Figure 10, according to the procedure described in Rechberger and Graedel.[3]

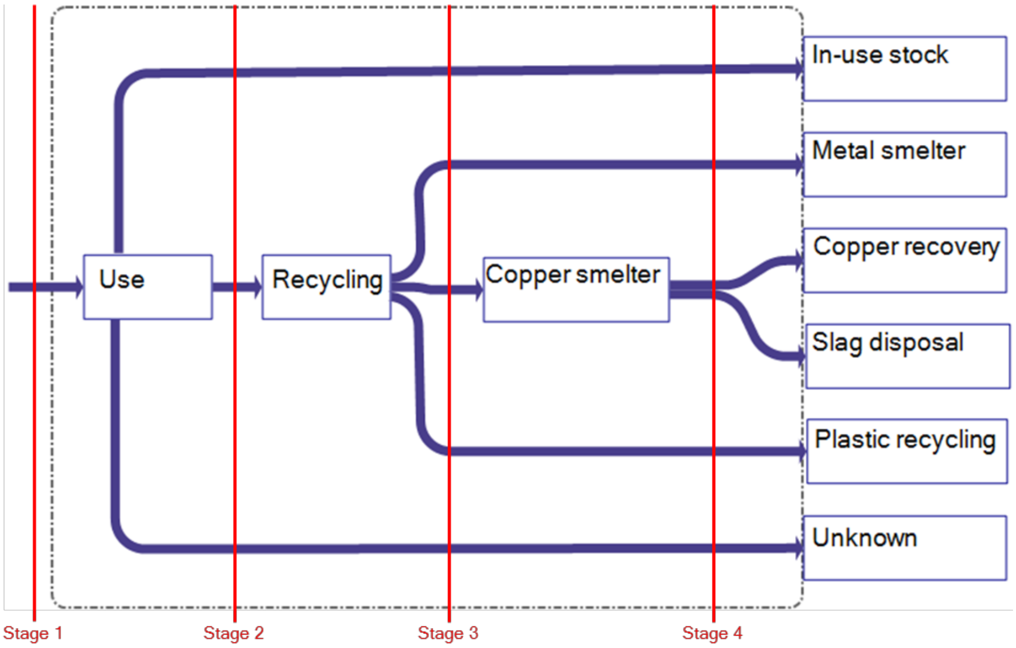


Figure 10: Allocation of the MFA system's material flows to stages.

In our example, we apply the SEA for copper in laptops and the resulting recycling flows. For every flow in the system, we need data on the material flows (laptops, metal scrap, plastics), the substance (copper) concentration in each of these material flows and the resulting substance (copper) flows. The material flow, concentrations and substance flows are related according to equation 5.

$$\dot{X}_i = \dot{M}_i \cdot c_i \quad (5)$$

$\dot{X}_i$ = Substance flow
$\dot{M}_i$ = Material flow,
c_i = concentration
i = specific flow in a stage

First, m_i for each stage is calculated (equation 6) and subsequently, the entropy $H(c_i, m_i)$ is calculated for each stage (equation 7).[3]

$$m_i = \frac{\dot{M}_i}{\sum_{i=1}^{k} \dot{X}_i} \quad (6)$$

$$H(c_i, m_i) = -\sum_{i=1}^{k} m_i \cdot c_i \cdot ld(c_i) \geq 0 \quad (7)$$

m_i = standardized mass fractions of a material set
k = total number of flows per stage
$H(c_i, m_i)$ = entropy in a specific stage
ld = logarithm to the second base

In each stage, the entropy H is compared to the entropy H_{max}. This results in the relative statistical entropy (RSE, equation 8). In industrial ecology, H_{max} often refers to the average earth crust content (equation 9). Thus, a stage with entropy H = H_{max} defines a point at which enhanced material resources no longer exist.[3]

$$RSE \equiv H/H_{max} \quad (8)$$

$$H_{max} = ld\left(\frac{1}{c_{EC}}\right) \quad (9)$$

RSE = relative statistical entropy
H_{max} = maximum statistical entropy
c_{EC} = average content of the earth crust

5.2 Source CSV file

For calculating a SEA with pymfa, you need to add additional information to the source CSV file. For an example, see Table 7.

1. You need to include a ***value for entropyHmax*** in row 6, according to Table 1. The formula for the calculation of H_{max} is given in equation 9.
2. You have to introduce a ***fix rate transfer*** between inflow and first node as shown in Chapter 4.4 below Table 5, so that the inflow is stored in the tool for further calculations.
3. After listing all transfers defining your system, you have to add ***concentration data*** for each transfer that contains the substances you are evaluating in kg/kg. Instead of a transfer type, you write 'concentration' in the first column. The information for 'source node', 'source material', 'source unit', 'target node', 'target material', 'target unit', and 'stages' should be identical to the information given above for the different transfers.
   The concentration data should be provided as deterministic values and can be varied for each time period. The tool will calculate the RSE from mean stocks and flows resulting from the probabilistic dynamic MFA and concentration data.
4. In column H, for each transfer you have to indicate through which ***stages*** it flows. If a transfer flows through various stages, the number of the stages are separated by an "|".

   If a transfer refers to the material flow and the substance flow can be calculated from the material flow and the concentration data, write only the number of the stage, e.g. **2|3**
   If a transfer refers to the substance flow and the material flow can be calculated from the substance flow and the concentration data, write an x behind the last number, e.g. **2|3|x**
   Examples for these two cases:
   The flow of laptops to the recycling plant multiplied by the copper concentration accounts for the copper flow in the flow of laptops: only numbers of stages, no x

The copper flow from laptops within the slag to a landfill is known. The concentration of copper in the slag can be estimated. However, the total amount of slag from the process (also from other material) is not known. Apply an x.

The delay transfer is a special case. In the SEA, from the delay transfer, the net flow to the stock is calculated (net flow = inflow – outflow). Thus, for a delay transfer, you have to indicate all stages through which the net flow to the stock flows. In SEA a net flow to a stock is always modeled as an output flow of the system. In our example in Figure 10, this would apply to the flow from 'use' to 'in-use stock'.

After finishing the setup of the source file for the SEA, it should be saved as a CSV file. The interactive web application has to be started by selecting one of the 3 options for the SEA calculations (Chapter 3). From the interactive web application, the source file can be uploaded and the simulation can be run.

5.3 Result CSV File

The results CSV file has the same format as the one resulting from the dynamic MFA simulation. Additionally, below the data for stocks and sinks, the RSE for each stage and year is listed (see Table 8).

Table 7: Example of source CSV file for dynamic MFA and SEA. Only the first two time periods are shown.

	A	B	C	D	E	F	G	H	I	J	K
1	Runs:	10000									
2	Periods:										
3	Median:	yes									
4	Percentiles:	10|90									
5	Plots:	In-use_stock|Recycling_kg									
6	entropyHmax:	14.02									
7											
8	**Transfer Type**	**Source Node**	**Source Material**	**Source Unit**	**Target Node**	**Target Material**	**Target Unit**	**Stages**	**Description**	**1990**	**1991**
9	Inflow				Laptop_sale	Laptops	Devices		Inflow of new Laptops	stoch|normal|17226,3445	stoch|normal|27562,5512
10	Rate	Laptop_sale	Laptops	Devices	In-use_stock	Laptops	Devices	1	Fix rate transfer for inflow for SEA	fix|1|1	fix|1|1
11	Delay	In-use_stock	Laptops	Devices	Stockoutflow	Laptops	Devices	2|3|4	Stock outflow based on service lifetime	fix|1|1|weibull|2.1,5.9|0	fix|1|1|weibull|2.1,5.9|0
12	Rate	Stockoutflow	Laptops	Devices	Recycling	Laptops	Devices	2	Stock outflow to recycling	stoch|triangular|0.7,0.8,0.9	stoch|triangular|0.7,0.8,0.
13	Rate	Stockoutflow	Laptops	Devices	Unknown	Laptops	Devices	2|3|4	Stock outflow to other disposal pathways	stoch|triangular|0.1,0.2,0.3	stoch|triangular|0.1,0.2,0.
14	Conversion	Recycling	Laptops	Devices	Recycling_kg	Laptops	Kg		Conversion from devices to kg	stoch|triangular|1.8,3.1,4.8	stoch|triangular|1.8,3.1,4.
15	Fraction	Recycling_kg	Laptops	kg	Metal_Smelter	Iron&Alu	kg	3|4	Fraction of iron and aluminium from laptops to metal smelter	stoch|uniform|0.46,0.50|0	stoch|uniform|0.46,0.50|0
16	Fraction	Recycling_kg	Laptops	kg	Copper_Smelte	Copper	kg	3	Fraction of copper from laptops to copper smelter	stoch|uniform|0.007,0.06|0	stoch|uniform|0.007,0.06|
17	Fraction	Recycling_kg	Laptops	kg	Copper_Smelte	PWB	kg	3	Fraction of PWBs from laptops to iron smelter	stoch|uniform|0.005,0.015|	stoch|uniform|0.005,0.015
18	Fraction	Recycling_kg	Laptops	kg	Plastic_Recyclin	Plastics	Kg	3|4	Fraction of plastics from laptops to plastics recycling	stoch|uniform|0.17,0.47|0	stoch|uniform|0.17,0.47|0
19	Rate	Copper_Smelter	Copper	kg	Recovered_cop	Copper	Kg	4	Flow of recovered copper from copper smelter	fix|0.95|0	fix|0.95|0
20	Rate	Copper_Smelter	PWB	kg	Recovered_cop	PWB	Kg	4	Flow of recovered copper from copper smelter, from PWBs	fix|0.95|0	fix|0.95|0
21	Rate	Copper_Smelter	Copper	kg	Slag_Landfill	Copper	Kg	4|x	Flow of copper from copper smelter to landfill	fix|0.05|0	fix|0.05|0
22	Rate	Copper_Smelter	PWB	kg	Slag_Landfill	PWB	Kg	4|x	Flow of copper from copper smelter to landfill disposal, from PWBs	fix|0.05|0	fix|0.05|0
23	Concentration	Laptop_sale	Laptops	Devices	In-use_stock	Laptops	Devices	1	Fix rate transfer for inflow for SEA	0.15	0.15
24	Concentration	In-use_stock	Laptops	Devices	Stockoutflow	Laptops	Devices	2|3|4	Stock outflow based on service lifetime	0.15	0.15
25	Concentration	Stockoutflow	Laptops	Devices	Recycling	Laptops	Devices	2	Stock outflow to recycling	0.15	0.15
26	Concentration	Stockoutflow	Laptops	Devices	Unknown	Laptops	Devices	2|3|4	Stock outflow to other disposal pathways	0.15	0.15
27	Concentration	Recycling_kg	Laptops	kg	Metal_Smelter	Iron&Alu	kg	3|4	Fraction of iron and aluminium from laptops to metal	0.01	0.01
28	Concentration	Recycling_kg	Laptops	kg	Copper_Smelte	Copper	kg	3	Fraction of copper from laptops to copper smelter	0.95	0.95
29	Concentration	Recycling_kg	Laptops	kg	Copper_Smelte	PWB	kg	3	Fraction of PWBs from laptops to copper smelter	0.19	0.19
30	Concentration	Recycling_kg	Laptops	kg	Plastic_Recyclin	Plastics	Kg	3|4	Fraction of plastics from laptops to plastics recycling	0.01	0.01
31	Concentration	Copper_Smelter	Copper	kg	Recovered_cop	Copper	Kg	4	Flow of recovered copper from copper smelter	0.99	0.99
32	Concentration	Copper_Smelter	PWB	kg	Recovered_cop	PWB	Kg	4	Flow of recovered copper from copper smelter, from	0.99	0.99
33	Concentration	Copper_Smelter	Copper	kg	Slag_Landfill	Copper	Kg	4|x	Flow of copper from copper smelter to landfill	0.00006	0.00006
34	Concentration	Copper_Smelter	Copper	kg	Slag_Landfill	PWB	Kg	4|x	Flow of copper from copper smelter to landfill	0.00006	0.00006

Table 8: Example of result CSV file for SEA

83	Sink	Recovered_cop	Copper	Kg						mean:	0	62.483081
84										median:	0	57.609583
85										10th perc.	0	21.4125569
86										90th perc.	0	109.884387
87	Sink	Recovered_cop	PWB	Kg						mean:	0	18.8060606
88										median:	0	17.5366117
89										10th perc.	0	9.60690188
90										90th perc.	0	29.6935104
91	Entropy	1990	1991	1992	1993							
92	Stage 1	0	0.195218659	0.195218659	0.195218659							
93	Stage 2	0	0.195218659	0.195218659	0.195218659							
94	Stage 3	0	0.194641909	0.192947356	0.190774648							
95	Stage 4	0	0.194565251	0.192653292	0.190260612							

6 Acknowledgments

The pymfa tool was designed, implemented and refined by Carol Alexandru, Rolf Badat and Dimitri Kohler from the University of Zurich.

7 References

(1) Brunner, P. H.; Rechberger, H. *Practical Handbook of Material Flow Analysis*; Lewis Publishers, 2004.

(2) Müller, E.; Hilty, L. M.; Widmer, R.; Schluep, M.; Faulstich, M. Modeling Metal Stocks and Flows: A Review of Dynamic Material Flow Analysis Methods. *Environ. Sci. Technol.* **2014**, *48* (4), 2102–2113.

(3) Rechberger, H., Graedel, T. E. The contemporary European copper cycle: Statistical entropy analysis. *Ecological Economics* **2002**, *42* (1–2), 59–72.

8 Appendix

8.1 Example of stochastic input generator for inflows

```
#!/usr/bin/env python3.4
import xlrd
import openpyxl
import numpy as np
from numpy import *

# Stochastic input generator for inflows of devices
file_location = 'C:/Users/xx/example.xlsx' #insert file location
device = 'Desktop' #insert device type
sheet_index_data_inflow = 0 #insert sheet index (first excel sheet has index 0)
type_values = 'normal' #insert type of values for stochastic model input: 'uniform', 'normal' or 'triangular'

output_file = 'Stochastic_Inflow.xlsx'

# open excel-file
workbook = xlrd.open_workbook(file_location)

# get sheets, index always start at 0
sheet = workbook.sheet_by_index(sheet_index_data_inflow)
```

```
#read data in the sheet
list_values = [[sheet.cell_value(r,c) for c in range (1,sheet.ncols)] for r in range (sheet.nrows)]

# create a new excel-file, prepare fix entries
wb = openpyxl.Workbook()
newsheet1 = wb.get_active_sheet()
newsheet1['A1'] = str(device)
newsheet1['A2'] = 'Year'
newsheet1['A3'] = 'Stochastic model entry'

col = 2 #Start to write values in third column (in the first column, inflow is written)
# create stochastic input values with two parameters for uniform and normal distribution
if type_values == 'uniform' or type_values == 'normal':
    for i in range (sheet.ncols-1):
        newsheet1.cell(row=2, column=col+i).value = list_values[0][i]
        newsheet1.cell(row=3, column=col+i).value
        ='stoch|'+type_values+'|'+str(round(list_values[1][i],3))+','+str(round(list_values[2][i],3))

# create stochastic input values with three parameters for triangular distribution
elif type_values == 'triangular':
    for i in range (sheet.ncols-1):
        newsheet1.cell(row=2, column=col+i).value = list_values[0][i]
```

```
        newsheet1.cell(row=3, column=col+i).value =
        'stoch|'+type_values+'|'+str(round(list_values[1][i],3))+','+str(round(list_values[2][i],3))+','
        +str(round(list_values[3][i],3))

wb.save(output_file)

print ("done")
```

8.2 Example of triangular distribution input creator for transfer coefficients

```
#!/usr/bin/env python3.4
import xlrd
import openpyxl
import numpy as np
from numpy import *

# Stochastic input generator for devices with temporally changing transfer coefficients, triangular distribution
file_location = 'C:/Users/xx/example.xlsx'
device = 'Laptop' #insert device type
sheet_index_rate_weighted = 1
sheet_index_rate_weighted_low = 2
sheet_index_rate_weighted_high = 3
## sheet_index_standard_deviation = ? #(for normal distribution)
type_values = 'triangular' #insert type of values for stochastic model input: 'uniform','normal' or 'triangular'
```

```
output_file = 'Stochastic_TK.xlsx' #insert name of output file

# open excel-file
workbook = xlrd.open_workbook(file_location)

# get sheets, index always start at 0, the weighted rates values are in two separate sheets, one for the low
# values, one for the high values
sheet1 = workbook.sheet_by_index(sheet_index_rate_weighted)
sheet2 = workbook.sheet_by_index(sheet_index_rate_weighted_low)
sheet3 = workbook.sheet_by_index(sheet_index_rate_weighted_high)
## sheet4 = workbook.sheet_by_index(sheet_index_standard_deviation)

#read data in the sheets
list_values = [[sheet1.cell_value(r,c) for c in range (2,sheet1.ncols)] for r in range (sheet1.nrows)]
list_values_low = [[sheet2.cell_value(r,c) for c in range (2,sheet2.ncols)] for r in range (sheet2.nrows)]
list_values_high = [[sheet3.cell_value(r,c) for c in range (2,sheet3.ncols)] for r in range (sheet3.nrows)]
## list_standard_deviation = [[sheet4.cell_value(r,c) for c in range (2,sheet4.ncols)] for r in range (sheet4.nrows)]

# create a new excel-file, prepare fix entries
wb = openpyxl.Workbook()
newsheet1 = wb.get_active_sheet()
newsheet1['A1'] = str(device)
newsheet1['A2'] = 'from'
```

```
newsheet1['B2'] = 'to'
newsheet1['A3'] = 'Source node 1'
newsheet1['A4'] = 'Source node 2'
# newsheet1['A5'] = 'Source node 3' etc., according to your system
newsheet1['B3'] = 'Target node 1'
newsheet1['B4'] = 'Target node 2'
# newsheet1['B5'] = 'Target node 3' etc., according to your system

col = 3 #Start to write values in third column (in the first two columns, the source and destination is written)
# create stochastic input values with two parameters for normal distribution
##if type_values == 'normal':
##    for i in range (1,34):
##        for j in range (sheet1.ncols-2):
##            newsheet1.cell(row=2, column=col+j).value = list_values[0][j]
##            newsheet1.cell(row=i+2, column=col+j).value =
##            'stoch|'+type_values+'|'+str(round(list_values[i][j],3))+','+str(round(list_standard_deviation[i][j],3))
##            +'|0'

## create stochastic input values with two parameters for uniform distribution
if type_values == 'uniform':
    for i in range (1,34):
        for j in range (sheet1.ncols-2):
            newsheet1.cell(row=2, column=col+j).value = list_values_low[0][j]
```

```
            newsheet1.cell(row=i+2, column=col+j).value =
            'stoch|'+type_values+'|'+str(round(list_values_low[i][j],3))+','+str(round(list_values_high[i][j],3))+'|0'



# create stochastic input values with three parameters for triangular distribution
elif type_values == 'triangular':
    for i in range (1,34):
        for j in range (sheet1.ncols-2):
            newsheet1.cell(row=2, column=col+j).value = list_values[0][j]
            newsheet1.cell(row=i+2, column=col+j).value =
            'stoch|'+type_values+'|'+str(round(list_values_low[i][j],5))+','+str(round(list_values[i][j],5))
            +','+str(round(list_values_high[i][j],5))+'|0'


wb.save(output_file)


print ('done')
```