The Monte Carlo driven and Machine Learning enhanced Process Simulator

Jones, Mark Nicholas; Frutiger, Jérôme; Ince, Nevin Gerek; Sin, Gürkan

Published in:
Computers & Chemical Engineering

Link to article, DOI:
10.1016/j.compchemeng.2019.03.016

Publication date:
2019

Document Version
Peer reviewed version

The Monte Carlo driven and Machine Learning enhanced Process Simulator

Mark Nicholas Jones, Jérôme Frutiger, Nevin Gerek Ince, Gürkan Sin

PII: S0098-1354(19)30071-7
DOI: https://doi.org/10.1016/j.compchemeng.2019.03.016
Reference: CACE 6377

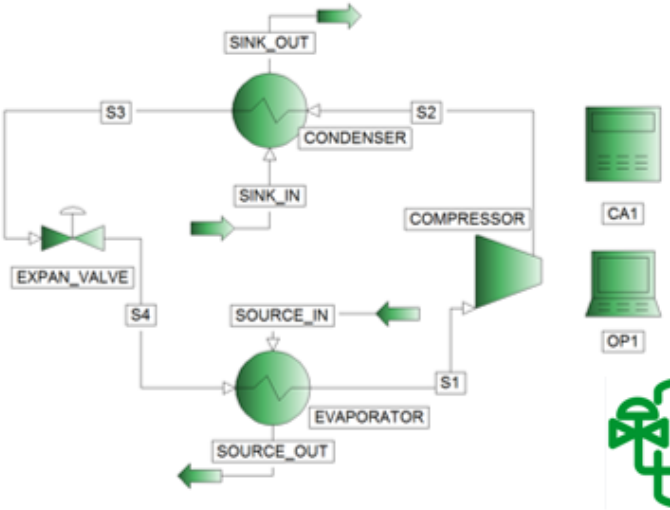
To appear in: *Computers and Chemical Engineering*

Received date: 18 January 2019
Revised date: 10 March 2019
Accepted date: 11 March 2019

Please cite this article as: Mark Nicholas Jones, Jérôme Frutiger, Nevin Gerek Ince, Gürkan Sin, The Monte Carlo driven and Machine Learning enhanced Process Simulator, *Computers and Chemical Engineering* (2019), doi: https://doi.org/10.1016/j.compchemeng.2019.03.016

SimSci PRO/II process simulator

1) Property and process model



2) Select and read properties: T_c, P_c, ω

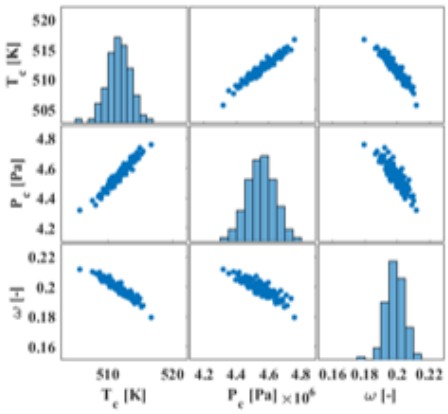
python COM

python COM

4) Evaluate property samples of T_c, P_c, ω

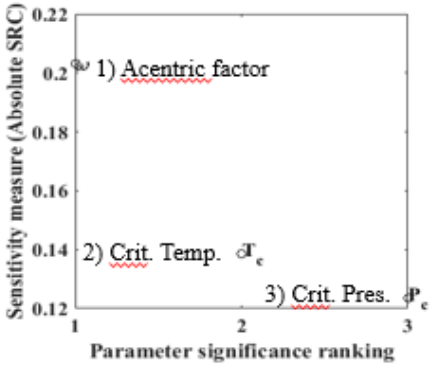
MATLAB

3) Monte Carlo based sampling within uncertainty range with/without correlation control



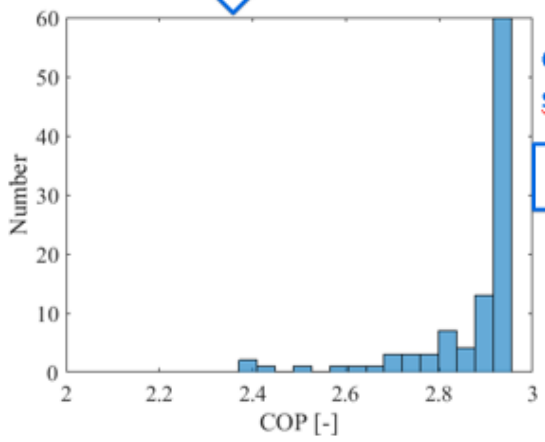
7) Sensitivity analysis of process with respect to properties

Standard regression coefficent (SRC)



5) Process model output uncertainty

python



6) Use output for sensitivity analysis

python COM

The Monte Carlo driven and Machine Learning enhanced Process Simulator

Mark Nicholas Jones[a,b], Jérôme Frutiger[b,c], Nevin Gerek Ince[d], Gürkan Sin[b,*]

[a] *Alfa Laval Copenhagen A/S, Maskinvej 5, 2860 Søborg, Denmark*
[b] *PROSYS Research Centre, Department of Chemical Engineering, Technical University of Denmark (DTU), Søltofts Plads 229, 2800 Kgs. Lyngby, Denmark*
[c] *Bayer AG, Engineering & Technology, Process Analysis & Optimization, Building E 41, 427, 51368 Leverkusen, Germany*
[d] *AVEVA, 10900 Equity Dr. Houston, TX 77041, United States*

Abstract

This study presents a methodology with tools integration to apply advanced uncertainty propagation and sensitivity analysis in connection with commercial process simulation software. The methodology was applied to two processes: a heat pump system and a molecular distillation process. The input parameters of the selected thermodynamic model, namely critical temperature, critical pressure and acentric factor, were considered as a source of uncertainty and analyzed using Monte Carlo sampling techniques. This enabled the process model output uncertainty to be described as an empirical distribution function with a 95% confidence interval. Variance-based decomposition such as the Sobol method or standard regression were used to analyse the sensitivity of the respective properties. We also show that machine learning methods such as polynomial chaos expansion (PCE) can be applied to reduce the number of necessary process simulations and obtained equivalent results in comparison with the more costly full Monte Carlo based procedure.

Keywords: Process simulation, Uncertainty, Sensitivity, Polynomial chaos

*gsi@kt.dtu.dk

1. Introduction

Process design and simulation heavily rely on the accuracy of physical and thermo-physical property data. Such data is subject to varying uncertainties depending on the quality of experimental data or regressed property models. The propagation of property uncertainties throughout the process simulation provides important insight into the reliability and sensitivity of the model and influences the design decisions to be taken [1]. There have been significant efforts to quantify uncertainties in experimental [2] and predicted property data [3]. Over the past decade a variety of studies have analysed and underlined the impact of property uncertainties in physical-property models on process design [4, 5, 6, 7]. However, property uncertainty analysis is still not widely applied in the industrial practice of chemical engineering. One possible reason is that the methods suggested by researchers are currently not available in commercial process simulators, such as PRO/II or Aspen, due to the difficulty of their implementation.

This paper presents a methodology that integrates property uncertainty analysis and sensitivity analysis in commercial process simulators. Unfortunately the above reviewed strategies and resources available for the analysis of property uncertainties do not present any code for the connection interface to commercial process simulators thereby hampering the utilization by the practicing engineer. Hence, users and developers of process models and process simulators can not use these strategies instantly. In this work the implementation of the connection interface is given in the appendix.

Mathias et al. [8, 9, 10] performed several studies performing property uncertainty analysis with Aspen Plus. The studies clearly highlighted the importance of the effect of property uncertainties in process simulators for model developers and users. The authors quantified the property uncertainty propagation of mixtures through different unit operations based on a Margules-based perturbation scheme (Margules uncertainty analysis) for analysing the effect of liquid activity coefficient perturbations on the process design. They propagated the

uncertainties through the process flowsheets using a lumped parameter subject to perturbation. This approach has large potential for further studies in the domain of more advanced uncertainty and sensitivity analysis techniques such as variance based and global sensitivity analysis.

Property models contain correlated parameters which are typically regressed from thermodynamic measurements and exhibit covariance with significant pairwise correlations [7]. However, the correlation matrix between these measured or estimate properties are rarely considered in uncertainty and sensitivity analysis studies.

In property databases, such as NIST TDE [11, 12] or the AIChE DIPPR database, the property parameters of one compound are not completely independent. If correlation is neglected the uncertainty propagation can lead to an overestimation of the output uncertainty [13]. Monte Carlo and machine learning methods have been used in a variety of studies [1, 14] and recently it has also been used for property uncertainty analysis [15, 7]. However, none of the authors implemented Monte Carlo and machine learning methods in a commercial process simulator.

In respect of integrating statistical tools in a process simulator Diwekar et al. implemented uncertainty analysis with Fortran routines for the Aspen process simulator for assigning parameter uncertainty distributions, accessing the decision variables in the simulator and collecting the evaluated data [16]. Further, Sahin and Diwekar combined their stochastic modelling tool with nonlinear optimisation under uncertainty. [17] Eslick et al. developed a framework for optimisation and quantification of uncertainty and sensitivity (FOQUS) with interfaces to Aspen Plus, Aspen Custom Modeler, gPROMS and Excel [18]. The FOQUS framework integrates the non-intrusive quantification toolkit PSUADE [19] which supports Sobol sensitivity analysis. Duong et al. have applied polynomial chaos expansion to perform sensitivity analysis of syngas and natural gas processes in Aspen HYSYS and Matlab [20].

In the current study we will present a simple methodology based on Monte Carlo sampling, machine learning and sensitivity evaluation techniques in order

to propagate and analyse property uncertainties in PRO/II. It is noted that the methodology is generic and can be readily adapted to different process simulators of interest.

The methodology is applied to a case study of a heat pump system for the recovery of heat and a molecular distillation unit to recover α-tocopherol and β-carotene. For the first case study Monte Carlo uncertainty propagation is used to analyse the output uncertainty and Monte Carlo standard regression is applied to study the global sensitivity of the properties with regression coefficients. The second case study calculates the mean and standard deviation for the property parameters, since no uncertainties have been provided with the experimental data or the software tools used. Also no covariance matrix has been documented with the predictive models reported in literature. Sobol's variance based decomposition technique was then applied to study the impact of property uncertainties on the model simulation output of interest in the molecular distillation case study.

2. Monte Carlo based uncertainty and sensitivity analysis

The strategy to apply the Monte Carlo method with process simulators is based on the methodology by Frutiger et al.[7] for property uncertainty propagation in process models. The methodology has been extended with Sobol (variance-based) sensitivity analysis (SA) in this work:

1) A process model is built in a commercial process simulation software such as PRO/II or Aspen. The property models and parameters for the uncertainty analysis need to be selected and the process variables are specified to satisfy the degrees of freedom.

2) The property uncertainty data is retrieved from databases (e.g. NIST TDE [12], AIChE DIPPR) or literature studies. Property uncertainty information needs to be estimated if not available, for example through calculation of the covariance matrix [3] or a bootstrap method [7].

3) Monte Carlo sampling technique is used to sample property values within its corresponding uncertainty range i.e. 95%-confidence interval using e.g. Matlab (2017b) or Python (3.6). Latin Hypercube Sampling (LHS) [21] or Sobol sequences [22] can be utilized for the probabilistic sampling over the components properties value space [1]. In this study the probability of uncertainty is assumed to follow a normal distribution. However, any other distribution is also possible. The rank-based method for correlation control of Iman and Conover [23] allows taking correlations between the property parameters into account. This is necessary, when parameters are not completely independent, as often is the case for property models.

4) The Monte Carlo samples are evaluated in the process model executed by the process simulator. In this work the PRO/II COM server is used, which provides read and write access to property information in PRO/II. This is done with the Python-COM interface developed by the authors.

5) Uncertainty analysis: The process model output uncertainty is quantified [7]. The Monte Carlo results provide a distribution function for the desired process model output of PRO/II. This can be analysed using mean and percentile calculations. Hence, the 95% confidence interval of the PRO/II output with respect to the corresponding input property values can be obtained.

6) Sensitivity analysis: Several sensitivity analysis methods exist which can be performed via Monte Carlo simulations or machine-learning methods. In this work we present and apply regression and variance based sensitivity analysis (SA) where the latter is either performed via Monte Carlo simulations or by first constructing a surrogate (response surface) model and then calculating the sensitivity indices. The sensitivity analysis methods applied in this work can be found in the appendix.

To this end the methodology has been presented and Table 1 summarises the

above steps. Figure 1 visualises a framework that can be established to pipeline the data flow from the sampling hypercube to the process simulation and the sensitivity analysis steps.

Table 1: Methodology for uncertainty and sensitivity analysis using a process simulator such as PRO/II

#	Step	Description	Output
1	Problem definition and analysis	Description of process and applied property models	$y = f(\theta)$
2	Define set of variables for property models	Input variables of correlations or models	$\theta = \{x_1, ..., x_M\}$
3	Retrieve uncertainty data	Mean and standard deviation	μ_i, σ_i
4	Sampling over properties' confidence intervals	LHS or Sobol random sequences to generate sampling hypercube	X_1, ..., X_k $\boldsymbol{\Omega}$
5	Monte Carlo simulations	Run process simulations and store outputs	$\boldsymbol{y}$
6	Uncertainty analysis	Mean and 95% confidence interval of output due to uncertainties in θ	$y_i \pm \sigma_i$
7	Sensitivity analysis	Retrieval of first, total and interaction sensitivity indices	$S1_i$, ST_i $S_{i,j}$

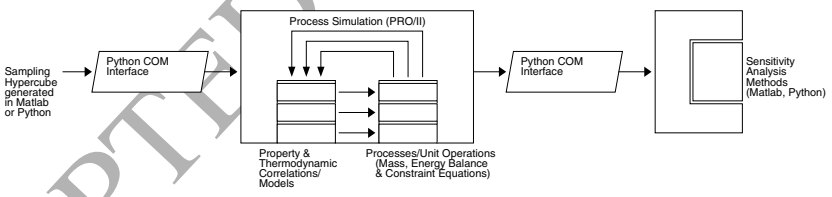


Figure 1: Framework with Python-COM-Interface to perform sensitivity analysis

3. Case studies

The methodology is demonstrated with two case studies respectively: a heat pump waste heat recovery and a molecular distillation process.

3.1. Heat recovery of organic rankine cycle

A heat pump system is used to recover heat from exhaust gas coming from a spray dryer in the food industry to preheat air before the spray dryer, utilizing secondary cycles with pressurized water [24]. Cyclopentane is used as the working fluid. Spray drying consumes a high amount of energy and is responsible for a large fraction of waste heat in the food industries [25]. Therefore, the choice of the working fluid and the assessment of the uncertainties affecting the performance of such a heat pump system is important to design an energy efficient process solution.

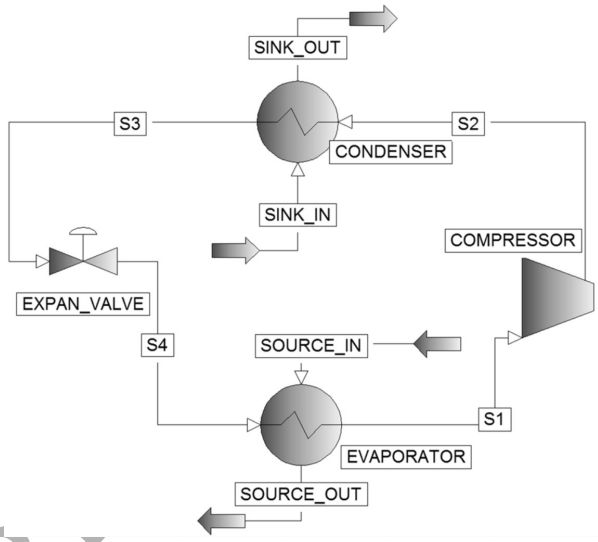


Figure 2: PRO/II flowsheet of the heat pump system

The process model performance indicator is the coefficient of performance (COP), which is the enthalpy provided by the heat source divided by the compressor work:

$$COP = \Delta H_{Source} / W_{Compressor} \tag{1}$$

The process model has been implemented in PRO/II (see Figure 2). The mass flows and relevant specifications can be found in the work of Zühlsdorf

et al. [24]. The SRK equation of state is selected as the property model for this system. The SRK parameters are the critical temperature T_C , the critical pressure P_C and the acentric factor ω. The goal of the case study is to propagate the uncertainty of T_C, P_C and ω through the SRK property model and the process model in PRO/II following the methodology described in section 2. The uncertainty range and the correlation matrix of the SRK parameters are collected from the work of Frutiger et al. [7], who obtained the correlation matrix using the bootstrap method [26] (Table 2). Advanced property analysis software, e.g. NIST TDE [11, 12], also provide the uncertainties and the correlation structures of the property models.

Table 2: Property parameter correlation matrix [26]

	T_C	P_C	ω
T_C	1.00		
P_C	-0.89	1.00	
ω	0.80	-0.48	1.00

In this work two different sampling strategies are applied: 1) Monte Carlo sampling without correlation, and 2) Monte Carlo sampling with correlation control according to Iman and Conover [23]. This allows the comparison of the influence of property parameter correlation on the process model output uncertainty. To the best of our knowledge such a comparison has to date not been reported in the literature. Figure 3 visualises the two sampling procedures. Here, only the properties of the working fluid cyclopentane are sampled and propagated. The property data set of water remains constant.

3.2. Molecular distillation

Different data sources were used to retrieve uncertainties for the critical temperature T_C, critical pressure P_C and acentric factor ω of the components for palm oil in this case study. Experimental values for T_C and P_C (without uncertainties) were taken from Diaz-Tovar [27]. The publication by Lim et

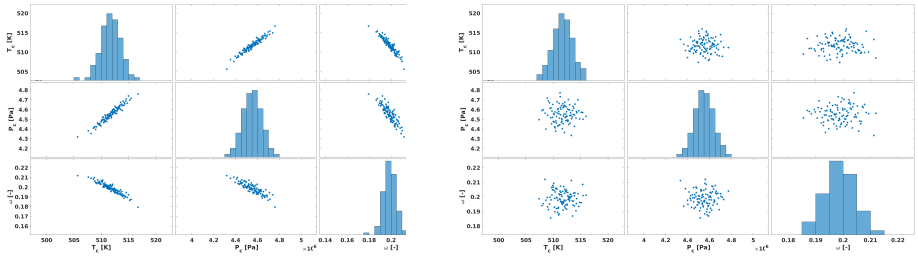


Figure 3: Illustration of samples generated by Latin Hypercube Sampling method with Iman and Conover correlation control (left) and without correlation control (right)

al. [28] applied the prediction methods by Dorn and Brunner [29] and Pitzer [30] and provided the property values for T_C, P_C and ω without uncertainties. Further, we used the property prediction tool ProPred [31] provided by the KT-Consortium software ICAS. With this application T_C, P_C and ω are predicted with the Marrero-Gani [32] or the Constantinou-Gani [33] group contribution models. No uncertainties were reported with any of these data resources or prediction models and therefore we used the experimental and predicted values to calculate the mean and standard deviation for each property as seen in Table 3.

In PRO/II 10.1 the Bio-Lib 10.1 (BIOFUELS) database provided the property values for tripalmitin and triolein, the PRO/II SIMSCI database provided the values for oleic acid and the KT-Consortium LIPIDS database contains the properties for α-tocopherol and β-carotene. The fill from structures option was selected in PRO/II to obtain values for the ideal gas enthalpy, liquid/vapour thermal conductivity, liquid/vapour viscosity and surface tension for β-carotene and α-tocopherol. The extended Antoine equation coefficients presented by Lim et al. [28] were provided to PRO/II as these correlations were also used by Tehlah et al. [34].

A molecular distillation base case design is modelled (Figure 4) and verified with the experimental data from Unnithan et al. [35] for the β-carotene recovery and the simulation performed by Tehlah et al. The molecular distillation

process consists of a short path distillator (180 °C, 0.008 mbar) with an internal condenser system which operates at 20 °C. The oil is pre-heated to 120 °C before it enters the short path distillator.

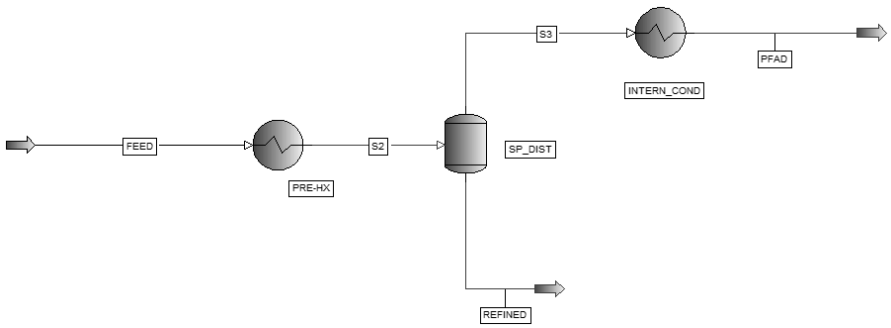


Figure 4: PRO/II flowsheet of molecular distillation process

Tehlah et al. perform their simulation with the Aspen process simulator and apply the Redlich-Kwong-Aspen EOS which is based on the SRK EOS.

In this work the SRK EOS is applied:

$$P = \frac{RT}{V_m - b} - \frac{a}{V_m(V_m + b)} \tag{2}$$

The parameters a and b are calculated with mixing rules which can be found in literature in different forms [36, 28].

These rules depend on the component fractions in the mixture, some interaction parameters and the following correction factors describing the attraction and volume of the molecules:

$$a_i = \alpha_i 0.42747 \frac{R^2 T_{C,i}^2}{P_{C,i}} \tag{3}$$

$$b_i = 0.08664 \frac{R T_{C,i}}{P_{C,i}} \tag{4}$$

Mathias [36] introduced the generalized temperature-dependent function

which improves the vapour pressure prediction:

$$\alpha_i(T) = [1 + m_i(1 - T_{r,i}^{1/2})]^2 \tag{5}$$

where m_i is a parameter for pure component i dependent on the acentric factor ω_i:

$$m_i = 0.48 + 1.574\omega_i - 0.176\omega_i^2 \tag{6}$$

and $T_{r,i}$ is the reduced temperature ($T_{r,i} = T/T_{C,i}$).

For comparison reasons we also applied the SimSci-SRK EOS, which similarly to the Aspen-SRK EOS, relies on a modification of the temperature-dependent function $\alpha_i(T)$ [37]. The K-value describing the distribution ratio of the individual components between the vapour and liquid phase will be derived from the EOS. Thus, the uncertainties in the property parameters will propagate from the EOS to the K-value calculations.

Table 3: Simulation results and comparison to experimental values from the patent by Unnithan et al. and simulation results from Tehlah et al. and this work with different equation of state models

		Tehlah et al.	**This work**	
Bottom component recovery	**Experimental**	**RK-Aspen**	**SRK**	**SRK-SimSci**
beta-carotene	95.98 %	98.96 %	95.82 %	95.81 %
alpha-tocopherol	98.54 %	not reported	40.37 %	40.30 %

Simulation Parameters: F = 2000 kg/h, T_1 = 120 °C, T_2 = 180 °C, P_2 = 0.008 mbar

The simulation results in Table 3 show the difference between the bottom component recoveries of β-carotene and α-tocopherol for the applied SRK models. Tehlah et al. did not report any values for the α-tocopherol recovery. We assume that their results for the top product recovery did not agree with the experimental results by Unnithan et al. [35]. Therefore we only investigate the effect of uncertainties on the bottom product of the molecular distillation unit. The parameters ($T_{C,i}$, $P_{C,i}$ and ω_i) are subject to the sensitivity analysis in the next sections and are overwritten with the COM-interface and the values stored

in the sampling hypercube for each simulation. The COP or β-carotene product fraction values are stored for all simulations in the output vector.

4. Results and discussion

4.1. Heat recovery of organic rankine cycle

4.1.1. Uncertainty analysis

The Monte Carlo simulation results can be illustrated by distribution plots of the COP for both sample sets (Figure 5). The larger the width of the COP output, the higher the output uncertainty. From the output distributions the 2.5% and 97.5% percentile defining the lower and upper bound of the uncertainty range (95%-confidence interval) for the process simulator output (here the COP) can be obtained (Table 4).

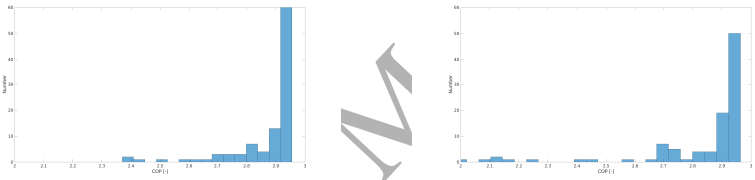


Figure 5: COP output distribution from the evaluation of the correlated property samples (left) and the uncorrelated property samples (right)

Table 4: Lower and upper bound of 95-confidence interval for COP for the correlated property samples (a) and the uncorrelated property samples

	COP mean values [-]	**COP 95% confidence interval values [-]**	
		Lower bound	**Upper bound**
Correlated samples	2.87	2.41	2.94
Uncorrelated samples	2.83	2.10	2.94

The same principle has been implemented solely in Matlab for heat pumps [25]. In this work we show and validate the analysis in connection with the PRO/II process simulator. As it can be seen in Figure 5 and Table 4, the uncertainty range of the COP is larger for the results based on uncorrelated

Monte Carlo sampling. In particular the lower bound of the uncertainty range is different. The uncorrelated samples also give COP values below 2.4 compared to the results of the correlated samples. This shows that, correlation between the property parameter samples can have a strong influence on the Monte Carlo uncertainty propagation procedures. If no or a wrong correlation is imposed on the property parameter samples, the process output uncertainty can be overestimated. This has already been discussed in detail by Frutiger et al. [3]. This implies that parameter correlation should be considered carefully in the context of property uncertainty propagation through process simulators.

4.1.2. Sensitivity analysis

The regression coefficients (SRCs) of the fluid parameters were obtained from linear least square regression using Eq. (1) (Figure 6). Correlated samples were used. The SRCs were ranked in respect to their absolute value. As to the analysis, the acentric factor ω has by far the highest influence compared to the other parameters. This is in agreement with a previous study [3]. Using a linear model between the inputs and outputs may mask some of the important properties. However, the Pearson correlation coefficient R^2 (>0.7) shows to what extend the model is linearisable. In case it is not linearisable, other statistical methods, e.g. Sobol SA or Morris screening need to be applied [1]. In the following section we therefore demonstrate how Sobol sensitivity analysis is applied to the second case study.

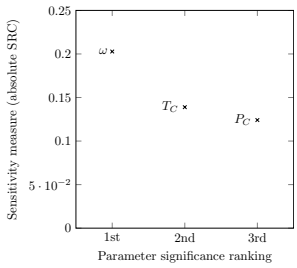


Figure 6: Standardized regression coefficients (SRCs) and ranking of SRK EOS property parameters using the correlated sample set

4.2. Molecular distillation

4.2.1. Uncertainty analysis

In respect to the vapour pressure correlations no uncertainty was provided by Lim et al. and thus the effect of uncertainties in the vapour pressure estimates couldn't be assessed in this study. The uncertainties of the critical temperature, critical pressure and acentric factor in respect to each component are calculated from the data provided by different sources as explained before. The mean and standard deviation are retrieved from the data and used for the following sensitivity analysis.

Table 5: Mean and standard deviation of properties for individual components of raw material palm oil calculated from experimental data and predicted data

	Experimental		**Prediction methods with regressed data sets**								**Mean and standard deviation ($\mu \pm \sigma$)**		
	Diaz-Tovar		**Dohrn and Brunner**		**Pritzer**	**Marrero-Gani**		**Constantinou-Gani**					
Component	T_C	P_C	T_C	P_C	ω	T_C	P_C	T_C	P_C	ω	T_C [K]	P_C [kPa]	ω [-]
Tripalmitin	1017.47	753.85	947.10	396.82	1.6500	1056.51	870	-	-	2.177	1007.03 ± 45.27	674 ± 201	1.914 ± 0.264
Triolein	1039.12	726.53	954.10	360.15	1.8004	1088.68	847	-	-	2.299	1027.3 ± 55.6	645 ± 207	2.050 ± 0.249
Oleic acid	781	1390	813.56	1250.2	0.8104	841.41	1449	-	-	1.151	811.99 ± 24.69	1363 ± 83	0.981 ± 0.170
alpha-Tocopherol	857.4	1070	936.93	838.45	1.1946	962.85	1207	-	-	-	919.06 ± 44.87	1038 ± 152	1.195 ± 0.0001
beta-Carotene	905.4	708.15	1031.1	678.41	1.6255	-	-	905.4	503	1.336	947.3 ± 59.3	630 ± 91	1.481 ± 0.145

ACCEPTED MANUSCRIPT

4.2.2. Sensitivity analysis

Six simulation runs were performed with sample sizes N of 32, 65, 98, 131, 313, 2188 and sampled values based on Saltelli's extension of the Sobol sequence [38]. These were normally distributed with the means and standard deviations for each of the 3 property parameters of each of the 5 components (Table 5). The generated sampling hypercube is used for performing Sobol sensitivity analysis and the obtained sensitivity indices $S1$ and ST are summarised in Table 6. The standard deviation is calculated with $\sigma = \sqrt{\sum_j \sigma_j^2}$. Standard deviation values with a negative $S1$ value are neglected in the summation since they have converged close to zero and are treated as non-influential parameters.

The 95% confidence interval of the β-carotene fraction at the bottom is 0.0008169 and 0.0008193 with the mean being 0.0008181 based on the simulation with a sample size of N=2188. This allows the engineer to assess if the needed purity for the β-carotene product lies within the uncertainty bounds. If this is not the case then the sensitivity analysis in this section will help to identify the properties which have to be revised through experiments or literature research to improve the output prediction of the process at hand.

As can be seen in Figure 7 from the first order sensitivity index, the critical temperature of β-carotene ($T_{C,Carotene}$), the critical pressures of tripalmitin ($P_{C,Tripalmitin}$) and triolein ($P_{C,Triolein}$) have a main effect on the β-carotene product fraction followed by $P_{C,Carotene}$ and $\omega_{Carotene}$. This shows, apart from the uncertainty of the property $T_{C,Carotene}$, that $P_{C,Tripalmitin}$ and $P_{C,Triolein}$ can be properties which have to be revised although they don't belong to the wanted product component fraction which we use as the output for the sensitivity analysis. We can also see that there is a high amount of interactions between the property parameters when comparing the values between S1 and ST for each parameter. This conclusion we can also draw because the sum of $S1_{i,j}$ doesn't add up to 1. Table 10 in the appendix presents the interaction matrix. Figure 8 visualises the summation of $T_{C,i}$, $P_{C,i}$ and ω_i, indicating which chemical species have the highest effect on the output. From these results we can conclude that

Table 6: Sobol SA of property variables used by the SRK EOS and it's effect on beta-carotene recovery with 70016 model evaluations (sample size N=2188)

Component	$S1_{j,k} \pm \sigma_{S1,j,k}$	$ST_{j,k} \pm \sigma_{ST,j,k}$
$T_{C,Tripalmitin}$	-0.000587 ± 0.004592	0.016667 ± 0.017378
$P_{C,Tripalmitin}$	0.090428 ± 0.044442	0.221925 ± 0.044754
$\omega_{Tripalmitin}$	0.002334 ± 0.007700	0.032921 ± 0.018697
$\sum_j S1_{j,Tripalmitin}$	0.092762 ± 0.045104	0.271513 ± 0.051522
$T_{C,Triolein}$	0.004282 ± 0.006385	0.014645 ± 0.015544
$P_{C,Triolein}$	0.088856 ± 0.040798	0.214896 ± 0.046402
$\omega_{Triolein}$	-0.002196 ± 0.006423	0.025806 ± 0.017340
$\sum_j S1_{j,Triolein}$	0.093138 ± 0.041295	0.255347 ± 0.051918
$T_{C,OleicAcid}$	0.002537 ± 0.006041	0.012168 ± 0.008537
$P_{C,OleicAcid}$	0.001808 ± 0.002213	0.000584 ± 0.000635
$\omega_{OleicAcid}$	-0.000080 ± 0.000146	0.000269 ± 0.000608
$\sum_j S1_{j,OleicAcid}$	0.004345 ± 0.006434	0.013021 ± 0.008582
$T_{C,Tocopherol}$	-0.000058 ± 0.000197	0.000275 ± 0.000608
$P_{C,Tocopherol}$	-0.000092 ± 0.000155	0.000270 ± 0.000608
$\omega_{Tocopherol}$	-0.000080 ± 0.000146	0.000269 ± 0.000608
$\sum_j S1_{j,Tocopherol}$		0.000814 ± 0.001053
$T_{C,Carotene}$	0.427228 ± 0.084424	0.634004 ± 0.072671
$P_{C,Carotene}$	0.025124 ± 0.027615	0.169324 ± 0.059240
$\omega_{Carotene}$	0.058044 ± 0.028075	0.185940 ± 0.033751
$\sum_j S1_{j,Carotene}$	0.510396 ± 0.093157	0.989268 ± 0.099647
$\sum_k \sum_j S1_{j,k}$	0.7006 ± 0.111621	1.529963 ± 0.123912

β-carotene has the highest effect on the β-carotene fraction at the bottom of the molecular distillation unit and also a high interaction between $T_{C,Carotene}$ and $P_{C,Carotene}$ or $\omega_{Carotene}$ can be identified although these values are highly uncertain and even a higher sample size N is needed to obtain more accurate values for the contribution of the individual interactions.

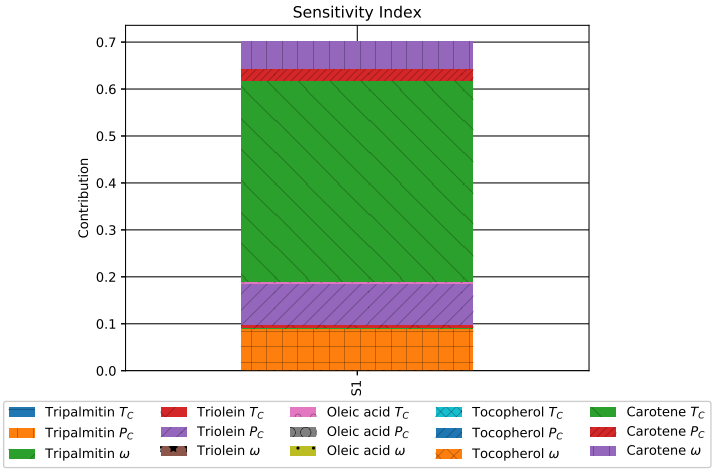


Figure 7: Sensitivity index bar plot for $S1_{i,j}$

Table 7: Comparison between Sobol SA with different sampling hypercube sizes

	Run 1	Run 2	Run 3	Run 4	Run 5	Run 6
Sample size N	32	65	98	131	313	2188
Number of evaluations (N*(2*M+2))	1024	2080	3136	4192	10016	70016
Run time [min]	69.73	143.52	226.25	298.27	865.45	6104.99

When tracking the time for evaluating all sample sets we can observe a linear increase (Figure 9, left plot) of the run time. The bottle neck is the Monte Carlo simulation step encompassing the data transmission between the COM-interface and PRO/II, populating the property variables within the process simulator, simulating the process and storing the final output vector. The sensitivity analysis is performed with the generated input-output data and takes less than 10 seconds. The Monte Carlo procedure won't be feasible for com-

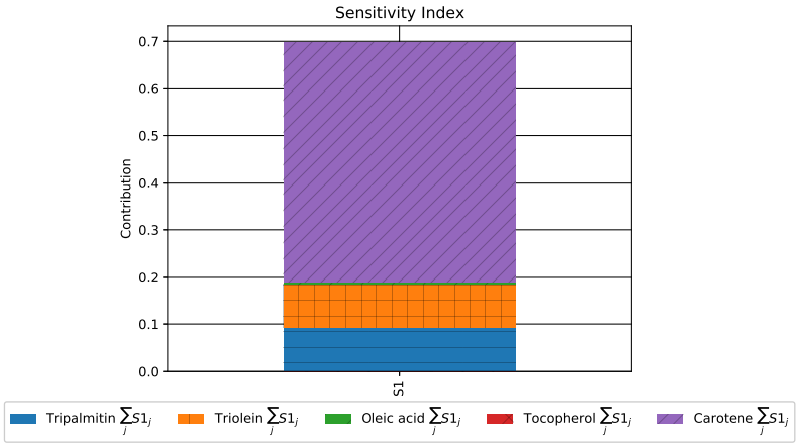


Figure 8: Sensitivity index bar plot for $\sum_j S1_{i,j}$

mercial process simulators if the evaluations are not run in parallel on multiple processors. The total time which the evaluations takes for run 6 is 6105 min (about 101 h) and even on a multicore computer with 4 processors would take too long performing the full Monte Carlo simulations if the results are needed within seconds or several minutes. This situation can of course change if computers for the general industrial user will be on the market with a high number of processors.

But since methods exist which can reduce the number of evaluations needed, we want to show that polynomial chaos expansion is a promising alternative to conduct sensitivity analysis with less computational effort and close to the results obtained with Monte Carlo based SA.

4.3. PCE based sensitivity analysis

The previous section showed that the retrieval of Sobol sensitivity indices from the full Monte Carlo approach resulted in high confidence intervals on the sensitivity indices for low sample size numbers and a high number of samples is needed to reduce these (Figure 9, right plot). We therefore applied a surrogate based sensitivity analysis where first an expansion of polynomial terms is regressed to 1024 process evaluations. Our problem at hand has a dimensionality

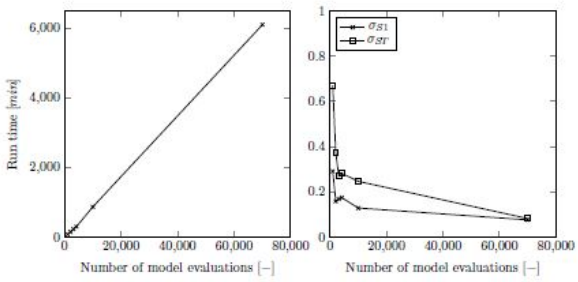


Figure 9: Run time over number of model evaluations (left plot) and standard deviation of S1 and ST for $T_{C,Carotene}$ (right plot) with increasing model evaluations for Monte Carlo based sensitivity analysis

of M=15 and the polynomial expansion we obtain fits the data with a coefficient of determination R^2=0.9148. The polynomial expansion has a degree of p=3 and 89 coefficients. In the second step the PCE is used to retrieve the sensitivity indices. Table 8 compares the obtained sensitivity indices with the previous obtained indices from the full Monte Carlo (MC) sensitivity analysis (SA). The results show that we are able to identify the same property parameters compared to MC SA with $T_{C,Carotene}$ being the most important. These results are obtained via PCE based SA with about 68-fold less model evaluations.

5. Perspective

We have shown that machine-learning based methods such as polynomial chaos expansion can reduce the computational time needed to perform sensitivity analysis. Fully data-driven methods use the input-output data and treat the process simulator as a black box. Therefore, the multiple evaluation of a process flowsheet should be performed in parallel with sequential process simulators. To this date no process simulator such as Aspen or PRO/II has been developed which supports this functionality and we are of the opinion that this will be a major improvement to adapt Monte Carlo and machine-learning based methods by commercial process simulators.

Table 8: Comparison between direct Monte Carlo based Sobol SA and PCE based Sobol SA

	PCE SA		**MC SA**	
Property	S1	ST	S1	ST
$T_{C,Tripalmitin}$	-	0.00	-	0.02
$P_{C,Tripalmitin}$	0.04	0.13	0.09	0.22
$\omega_{Tripalmitin}$	0.00	0.00	0.00	0.03
$T_{C,Triolein}$	0.00	0.00	0.00	0.01
$P_{C,Triolein}$	0.03	0.12	0.09	0.21
$\omega_{Triolein}$	0.00	0.00	-	0.03
$T_{C,OleicAcid}$	-	0.00	0.00	0.01
$P_{C,OleicAcid}$	-	-	0.00	0.00
$\omega_{OleicAcid}$	-	0.01	-	0.00
$T_{C,Tocopherol}$	-	0.00	-	0.00
$P_{C,Tocopherol}$	-	0.00	-	0.00
$\omega_{Tocopherol}$	-	0	-	0.00
$T_{C,Carotene}$	0.55	0.69	0.43	0.63
$P_{C,Carotene}$	0.08	0.16	0.03	0.17
$\omega_{Carotene}$	0.07	0.15	0.06	0.19
$\sum S_{j,k}$	0.77	1.26	0.75	1.52
Model evaluations	1024		70016	

We therefore recommend the support of parallel evaluations of the process models on multiple cores (e.g. with Co-array Fortran and/or MPI [39, 40, 41]) to reduce the computational time due to the high number of evaluations in case of the full Monte Carlo approach or to increase the speed of machine learning methods. This is necessary and important to realise the growing potential of these methods. It would be worthwile to test the performance of different machine learning techniques in respect to sensitivity analysis methods in the future, e.g. the comparison between neural networks, Gaussian process regression and polynomial chaos expansion.

6. Conclusions

Monte Carlo and machine learning methods can be integrated in commercial process simulators and applied by industrial users. Uncertainties of experimental data or estimated values should always be reported and we suggest that process simulators should take the uncertainties in properties as given and evaluate process models in respect to these uncertainty ranges. We demonstrated in this work that small changes in property uncertainties can have a major effect on the process flowsheet output.

The study presented a Monte Carlo based methodology and tools integration with scripting languages such as Python or Matlab for enabling property uncertainty and sensitivity analysis with a commercial process simulator. The framework enables process design engineers to perform robustness analysis of process design effectively and increases the value of commercial process simulators since considering uncertainty gains more and more importance in the process systems engineering community. The study shows that it is possible to use Monte Carlo techniques with commercial process simulators, which is currently not state-of-the-art in industrial practice. Further we highlight that machine learning based techniques can be applied to reduce the computational expensive full Monte Carlo approach. This was exemplified with polynomial chaos expansion. To this end, the generic nature of the framework was suc-

cessfully implemented with two case studies, namely a heat pump system with cyclopentane as a working fluid and a molecular distillation process.

The uncertainties in properties such as critical temperature $T_{C,i}$, critical pressure $P_{C,i}$ and acentric factor ω_i were propagated through the SRK EOS and the process models in PRO/II. The results show that considering correlation control in the sampling procedure influences the model output uncertainty.

The most sensitive property parameter in the heat pump system was the acentric factor of the working fluid cyclopentane. Analysis of the molecular distillation showed that the uncertainty of the recovery of β-carotene can be apportioned mostly to $T_{C,Carotene}$ as the input properties' uncertainties propagate from the SRK EOS through the calculations of the molecular distillation unit. The mostly higher values of the total order indices indicate a high degree of interaction between the parameters. This can be verified by evaluating the interaction sensitivity indices as shown in Table 10 in the appendix. The methodology we presented is also applicable to the case of analysing multiple outputs of a process.

7. Supplementary material

We provide the basic Python functions of the COM interface to connect to the PRO/II process simulator in the supplementary material. Further, we attached the input-output data to perform the sensitivity analysis and polynomial expansion with.

8. Acknowledgements

This work has received funding from the European Unions Horizon 2020 Research and Innovation Programme under the Marie Sklodowska-Curie Grant Agreement No. 675251.

References

References

[1] G. Sin, K. V. Gernaey, A. E. Lantz, Good modeling practice for pat applications: Propagation of input uncertainty and sensitivity analysis, Biotechnology Progress 25 (4) (2009) 1043–1053. doi:10.1002/btpr.166.

[2] Q. Dong, R. D. Chirico, X. Yan, X. Hong, M. Frenkel, Uncertainty reporting for experimental thermodynamic properties, Journal of Chemical & Engineering Data 50 (2) (2005) 546–550. doi:10.1021/je049673d.

[3] J. Frutiger, C. Marcarie, J. Abildskov, G. Sin, A comprehensive methodology for development, parameter estimation, and uncertainty analysis of group contribution based property modelsan application to the heat of combustion, Journal of Chemical & Engineering Data 61 (1) (2016) 602–613. doi:10.1021/acs.jced.5b00750.

[4] S. Macchietto, G. Maduabeuke, R. Szczepanski, Exact determination of process sensitivity to physical properties, Fluid Phase Equilibria 29 (1986) 59 – 67. doi:10.1016/0378-3812(86)85011-7.

[5] W. B. Whiting, Effects of uncertainties in thermodynamic data and models on process calculations, Journal of Chemical & Engineering Data 41 (5) (1996) 935–941. doi:10.1021/je9600764.

[6] S. Hajipour, M. A. Satyro, Uncertainty analysis applied to thermodynamic models and process design 1. pure components, Fluid Phase Equilibria 307 (1) (2011) 78 – 94. doi:10.1016/j.fluid.2011.05.014.

[7] J. Frutiger, I. Bell, J. P. O'Connell, K. Kroenlein, J. Abildskov, G. Sin, Uncertainty assessment of equations of state with application to an organic rankine cycle, Molecular Physics 115 (9-12) (2017) 1225–1244. doi:10.1080/00268976.2016.1275856.

[8] P. M. Mathias, Sensitivity of process design to phase equilibrium A new perturbation method based upon the margules equation, Journal of Chemical & Engineering Data 59 (4) (2014) 1006–1015. doi:10.1021/je400748p.

[9] P. M. Mathias, H. Z. Kister, Effect of phase-equilibrium uncertainties on ethyl acetate purification, Journal of Chemical & Engineering Data 62 (9) (2017) 2872–2883. doi:10.1021/acs.jced.7b00172.

[10] P. M. Mathias, H. Z. Kister, B. Parker, L. Narvaez, T. Schafer, A. Erickson, Component trapping with vaporliquid equilibrium uncertainty: Principles, design, and troubleshooting, Industrial & Engineering Chemistry Research 56 (40) (2017) 11593–11602. doi:10.1021/acs.iecr.7b02812.

[11] V. Diky, R. D. Chirico, C. D. Muzny, A. F. Kazakov, K. Kroenlein, J. W. Magee, I. Abdulagatov, J. W. Kang, R. Gani, M. Frenkel, Thermodata engine (tde): Software implementation of the dynamic data evaluation concept. 8. properties of material streams and solvent design, Journal of Chemical Information and Modeling 53 (1) (2013) 249–266. doi:10.1021/ci300470t.

[12] M. Frenkel, R. D. Chirico, V. Diky, X. Yan, Q. Dong, C. Muzny, Thermodata engine (tde): software implementation of the dynamic data evaluation concept, Journal of Chemical Information and Modeling 45 (4) (2005) 816–838. doi:10.1021/ci050067b.

[13] J. C. Helton, F. J. Davis, Latin hypercube sampling and the propagation of uncertainty in analyses of complex systems, Rel. Eng. & Sys. Safety 81 (2003) 23–69. doi:10.1016/S0951-8320(03)00058-9.

[14] M. Jones, H. Forero-Hernandez, A. Zubov, B. Sarup, G. Sin, Superstructure optimization of oleochemical processes with surrogate models, in: M. R. Eden, M. G. Ierapetritou, G. P. Towler (Eds.), 13th International Symposium on Process Systems Engineering (PSE 2018), Vol. 44 of Computer Aided Chemical Engineering, 2018, pp. 277 – 282. doi:10.1016/B978-0-444-64241-7.50041-0.

[15] S. Hajipour, M. A. Satyro, M. W. Foley, Uncertainty analysis applied to thermodynamic models and fuel properties natural gas dew points and gasoline reid vapor pressures, Energy & Fuels 28 (2) (2014) 1569–1578. doi:10.1021/ef4019838.

[16] U. Diwekar, E. Rubin, Stochastic modeling of chemical processes, Computers Chemical Engineering 15 (2) (1991) 105 – 114. doi:https://doi.org/10.1016/0098-1354(91)87009-X.

[17] K. H. Sahin, U. M. Diwekar, Better optimization of nonlinear uncertain systems (bonus): A new algorithm for stochastic programming using reweighting through kernel density estimation, Annals of Operations Research 132 (1) (2004) 47–68. doi:10.1023/B:ANOR.0000045276.18995.c8.

[18] J. C. Eslick, B. Ng, Q. Gao, C. H. Tong, N. V. Sahinidis, D. C. Miller, A framework for optimization and quantification of uncertainty and sensitivity for developing carbon capture systems, Energy Procedia 63 (2014) 1055 – 1063, 12th International Conference on Greenhouse Gas Control Technologies, GHGT-12. doi:https://doi.org/10.1016/j.egypro.2014.11.113.

[19] C. Tong, Problem Solving Environment for Uncertainty Analysis and Design Exploration, Springer International Publishing, 2016, pp. 1–37. doi:10.1007/978-3-319-11259-6_53-1.

[20] P. L. T. Duong, W. Ali, E. Kwok, M. Lee, Uncertainty quantification and global sensitivity analysis of complex chemical process using a generalized polynomial chaos approach, Computers & Chemical Engineering 90 (2016) 23 – 30. doi:https://doi.org/10.1016/j.compchemeng.2016.03.020.

[21] M. McKay, R. Beckman, W. Conover, Comparison the three methods for selecting values of input variable in the analysis of output from a computer code, Technometrics; (United States) 21:2. doi:10.1080/00401706.1979.10489755.

[22] I. M. Sobol, D. Asotsky, A. Kreinin, S. Kucherenko, Construction and comparison of high-dimensional sobol' generators, Wilmott 2011 (56) (2011) 64–79. doi:10.1002/wilm.10056.

[23] R. L. Iman, W. J. Conover, A distribution-free approach to inducing rank correlation among input variables, Communications in Statistics - Simulation and Computation 11 (3) (1982) 311–334. doi:10.1080/03610918208812265.

[24] B. Zühlsdorf, F. Bühler, R. Mancini, S. Cignitti, B. Elmegaard, High temperature heat pump integration using zeotropic working fluids for spray drying facilities, in: Proceedings of the 12th IEA Heat Pump Conference 2017, 2017.

[25] J. Frutiger, B. Zühlsdorf, B. Elmegaard, J. Abildskov, G. Sin, Reverse engineering of working fluid selection for industrial heat pump based on monte carlo sampling and uncertainty analysis, Industrial & Engineering Chemistry Research 57 (40) (2018) 13463–13477. doi:10.1021/acs.iecr.7b04607.

[26] B. Efron, Bootstrap methods: Another look at the jackknife, Ann. Statist. 7 (1) (1979) 1–26. doi:10.1214/aos/1176344552.

[27] C. Diaz Tovar, R. Gani, B. Sarup, Computer-aided modeling of lipid processing technology, Ph.D. thesis (7 2011).

[28] C. S. Lim, Z. A. Manan, M. R. Sarmidi, Simulation modeling of the phase behavior of palm oil-supercritical carbon dioxide, Journal of the American Oil Chemists' Society 80 (11) (2003) 1147–1156. doi:10.1007/s11746-003-0834-6.

[29] R. Dohrn, G. Brunner, An estimation method to calculate T_b, T_c, P_c and ω from the liquid molar volume and the vapor pressure, in: Proceedings of the 3rd International Symposium on Supercritical Fluids, 1994, pp. 214–248.

[30] K. S. Pitzer, The volumetric and thermodynamic properties of fluids. i. theoretical basis and virial coefficients1, Journal of the American Chemical Society 77 (13) (1955) 3427–3433. doi:10.1021/ja01618a001.

[31] A. S. Hukkerikar, B. Sarup, A. T. Kate, J. Abildskov, G. Sin, R. Gani, Group-contribution+ (gc+) based estimation of properties of pure components: Improved property estimation and uncertainty analysis, Fluid Phase Equilibria 321 (2012) 25 – 43. doi:https://doi.org/10.1016/j.fluid.2012.02.010.

[32] J. Marrero, R. Gani, Group-contribution based estimation of pure component properties, Fluid Phase Equilibria 183-184 (2001) 183 – 208, proceedings of the fourteenth symposium on thermophysical properties. doi:https://doi.org/10.1016/S0378-3812(01)00431-9.

[33] L. Constantinou, R. Gani, New group contribution method for estimating properties of pure compounds, AIChE Journal 40 (10) (1994) 1697–1710. doi:10.1002/aic.690401011.

[34] N. Tehlah, P. Kaewpradit, I. M. Mujtaba, Development of molecular distillation based simulation and optimization of refined palm oil process based on response surface methodology, Processes 5 (3). doi:10.3390/pr5030040.

[35] U. R. Unnithan, Refining of edible oil rich in natural carotenes and vitamin E (Aug 1999).
URL https://patents.google.com/patent/US5932261

[36] P. M. Mathias, A versatile phase equilibrium equation of state, Industrial & Engineering Chemistry Process Design and Development 22 (3) (1983) 385–391. doi:10.1021/i200022a008.

[37] P. M. Mathias, H. C. Klotz, J. M. Prausnitz, Equation-of-state mixing rules for multicomponent mixtures: the problem of invariance, Fluid Phase Equilibria 67 (1991) 31 – 44. doi:10.1016/0378-3812(91)90045-9.

[38] A. Saltelli, Making best use of model evaluations to compute sensitivity indices, Computer Physics Communications 145 (2) (2002) 280 – 297. doi:10.1016/S0010-4655(02)00280-1.

[39] S. Garain, D. S. Balsara, J. Reid, Comparing coarray fortran (caf) with mpi for several structured mesh pde applications, Journal of Computational Physics 297 (2015) 237 – 253. doi:https://doi.org/10.1016/j.jcp.2015.05.020.

[40] A. Sharma, I. Moulitsas, Mpi to coarray fortran. experiences with a cfd solver for unstructured meshes, Scientific Programming 2017 (3409647). doi:10.1155/2017/3409647.

[41] F. T. Tracy, T. C. Oppe, M. K. Corcoran, A comparison of mpi and co-array fortran for large finite element variably saturated flow simulations, Scalable Computing: Practice and Experience 19 (4) (2018) 423–432. doi:10.12694/scpe.v19i4.1468.

[42] A. Saltelli, M. Ratto, T. Andres, F. Campolongo, J. Cariboni, D. Gatelli, M. Saisana, S. Tarantola, Global Sensitivity Analysis: The Primer, Wiley, 2008.

[43] M. Jansen, Analysis of variance designs for model output, Computer Physics Communication 117 (1-2) (1999) 35 – 43. doi:10.1016/S0010-4655(98)00154-4.

[44] A. Saltelli, P. Annoni, I. Azzini, F. Campolongo, M. Ratto, S. Tarantola, Variance based sensitivity analysis of model output. design and estimator for the total sensitivity index, Computer Physics Communications 181 (2) (2010) 259 – 270. doi:10.1016/j.cpc.2009.09.018.

[45] J. Herman, W. Usher, Salib: An open-source python library for sensitivity analysis, Journal of Open Source Software 2 (9). doi:10.21105/joss.00097.

[46] A. Zubov, G. Sin, Multiscale modeling of poly(lactic acid) production: From reaction conditions to theology of polymer melt, Chemical Engineering Journal 336 (2018) 361 – 375. doi:10.1016/j.cej.2017.12.033.

[47] B. Sudret, Global sensitivity analysis using polynomial chaos expansions, Reliability Engineering and System Safety 93 (7) (2008) 964 – 979. doi:https://doi.org/10.1016/j.ress.2007.04.002.

[48] J. Feinberg, H. P. Langtangen, Chaospy: An open source tool for designing methods of uncertainty quantification, Journal of Computational Science 11 (2015) 46 – 57. doi:https://doi.org/10.1016/j.jocs.2015.08.008.

[49] D. Xiu, G. Karniadakis, The wiener–askey polynomial chaos for stochastic differential equations, SIAM Journal on Scientific Computing 24 (2) (2002) 619–644. doi:10.1137/S1064827501387826.

[50] R. B. Nelsen, An Introduction to Copulas, Lecture Notes in Statistics, Springer-Verlag, 2006. doi:10.1007/0-387-28678-0.

[51] A. Alexanderian, On spectral methods for variance based sensitivity analysis, Probab. Surveys 10 (2013) 51–68. doi:10.1214/13-PS219.

[52] G. Blatman, B. Sudret, An adaptive algorithm to build up sparse polynomial chaos expansions for stochastic finite element analysis, Probabilistic Engineering Mechanics 25 (2) (2010) 183 – 197.

[53] J. Slim, F. Rathmann, A. Nass, H. Soltner, R. Gebel, J. Pretz, D. Heberling, Polynomial chaos expansion method as a tool to evaluate and quantify field homogeneities of a novel waveguide rf wien filter, Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment 859 (2017) 52 – 62. doi:10.1016/j.nima.2017.03.040.

[54] B. Efron, T. Hastie, I. Johnstone, R. Tibshirani, Least angle regression, Ann. Statist. 32 (2) (2004) 407–499. doi:10.1214/009053604000000067.

[55] O. Garcia-Cabrejo, A. Valocchi, Global sensitivity analysis for multivariate output using polynomial chaos expansion, Reliability Engineering System Safety 126 (2014) 25 – 36. doi:https://doi.org/10.1016/j.ress.2014.01.005.

Nomenclature

$\alpha_i(T)$	Generalized temperature-dependent function for pure component i describing the attractive potential of the molecules [-]	a	Correction constant for attractive potential of molecules $[Jm^3mol^{-2}]$
β_j	Standardized regression coefficient (SRC) of the j^{th} input parameter	a_i	Attraction parameter for pure component i $[Jm^3mol^{-2}]$
ΔH_{Source}	Difference in heat enthalpies of sink and source $[J]$	b	Volume correction constant $[m^3mol^{-1}]$
ϵ_j	Error of regression model for parameter j	b_i	Volume parameter for pure component i $[m^3mol^{-1}]$
μ_i	Mean of output i	COP	Coefficient of performance [-]
$\mu_{\theta,j}$	Mean of input parameter j in θ	M	Number of input variables; Dimension of input vector [-]
ω	Acentric factor [-]	m_i	Constant for pure component i [-]
Φ	Multivariate polynomial	N	Sample size [-]
σ_i	Standard deviation of output i	P	Gas pressure $[Pa]$
$\sigma_{\theta,j}$	Standard deviation of input parameter j in θ	P_C	Critical pressure $[Pa]$
θ	Input parameter set	R	Ideal gas constant $[Jmol^{-1}K^{-1}]$
ξ	Realiziation of independent input random variable	$S1_i$	First order sensitivity index of input i
		S_i	First order sensitivity index of output i

S_{ijl}	Third order sensitivity index of output i	$y(A)$	Data matrix A of size (N/2, 2k) containing half of the samples from the matrix of random numbers
S_{ij}	Second order sensitivity index of output i	$y(A_B)$	Data matrix copied from A except i^{th} column which is copied from B
ST	Total effect sensitivity index	$y(B)$	Data matrix B of size (N/2, 2k) containing half of the samples from the matrix of random numbers
T	Temperature $[K]$	Y_i	Output i of process
T_C	Critical temperature $[K]$	α	Multi-index for PCE
T_r	Reduced temperature [-]		
$V(y)$	Variance of output		
V_m	Molar volume $[m^3 mol^{-1}]$		
$W_{Compressor}$	Compressor work $[J]$		
x_i	Input variable i		

Appendix

Sensitivity analysis with standardized regression coefficients, the Sobol method and polynomial chaos expansion

Regression analysis constructs linear regression models on the outputs obtained from the Monte Carlo procedure [1]. It is possible to obtain the standardized regression coefficient (SRC) β_j of parameter set θ_j given by:

$$\frac{Y_i - \mu_i}{\sigma_i} = \sum_{j=1}^{M} \beta_j \frac{\theta_j - \mu_{\theta,j}}{\sigma_{\theta,j}} + \epsilon_j \tag{7}$$

where Y_i is the simulator output, ϵ_j is the error of parameter j on the regression model, $\mu_{\theta,j}$, $\sigma_{\theta,j}$, μ_i and σ_i are the corresponding mean and standard deviations of the parameters and the model output. The SRC is a measure of how strongly the parameter contributes to the model output [3].

Another approach is Sobol SA which also allows to retrieve sensitivity indices. The analysis of variance (ANOVA-HDMR) decomposition gives the relation:

$$\sum_i S_i + \sum_i \sum_{j>i} S_{ij} + \sum_i \sum_{j>i} \sum_{l>j} S_{ijl} + ... + S_{123...M} = 1 \tag{8}$$

For a model with 3 input variables under study, the total effect ST_1 of input variable x_1 is analytically defined as the sum of the first order effect and higher order interactions:

$$ST_1 = S_1 + S_{12} + S_{13} + S_{123} = S1_1 + S_{12} + S_{13} + S_{123} \tag{9}$$

The first-order sensitivity index is a measure for the contribution of the i-th input parameter to the variance of the output $V(Y)$. We can describe the index in probabilistic form as:

$$S1_i = \frac{V[E(y|x_i)]}{V(y)} \tag{10}$$

where $V[E(y|x_i)]$ is the conditional variance and for $S1_i$ the following condition holds:

$$0 \leq S1_i \leq 1 \quad \wedge \quad \sum S1_i \leq 1 \tag{11}$$

$S1_1 = 1$ would imply that all variance of y is affected by x_1 and fixing it also determines y.

We generate two independent sampling matrices A and B where the row index denotes the simulation number and the column index references the input factor. $A_B^{(i)}$ is a matrix which is copied from A except the i-th column which is copied from B. With these matrices we can calculate the first-order index [42]:

$$S1_i = \frac{\frac{1}{N}\sum_{j=1}^{N} y(B)_j (y(A_B^{(i)})_j - y(A)_j)}{V(y)} \tag{12}$$

The total effect index describes the amount of interactions between the input factors. In probabilistic form, the definition of the index is:

$$ST_i = \frac{E[V(y|x_{\sim i})]}{V(y)} \tag{13}$$

$2^M - 1$ total sensitivity indices exist for a given input-output model with M number of inputs. We calculate the first order and total effect indices with A, B and $A_B^{(i)}$ [43] [44] [45]:

$$ST_i = \frac{\frac{1}{2N}\sum_{j=1}^{N} (y(A)_j - y(A_B^{(i)})_j)^2}{V(y)} \tag{14}$$

where for both indices $V(y)$ is the variance of the model output [42] [46]:

$$V(y) = \frac{1}{N}\sum_{j=1}^{N} (y(A)_j)^2 - (\frac{1}{N}\sum_{j=1}^{N} y(A)_j)^2 \tag{15}$$

2N simulations have to be performed to obtain the output y from the matrices A and B. For computing $y(A_B^{(i)})$, M times N simulations are needed. The overall model evaluations would be N(M+2) [44]. To also calculate the interaction effects the number of needed model evaluations would be N(2M+2) [45].

The other approach we want to discuss in this work is the retrieval of the sensitivity indices from a surrogate (response surface) model which we generate from the input-output data via polynomial chaos expansion (PCE). A PCE can be expressed by the following equation [47, 48]:

$$\hat{y}(\boldsymbol{\xi}) = \sum_{i \in \mathbb{N}^M}^{\infty} c_i \Phi_i(\boldsymbol{\xi}) \tag{16}$$

A truncation scheme is applied limiting the expansion order to a degree of p to make the method computational feasible. In this work we generate the polynomial basis with the Wiener-Askey scheme [49] which defines the type of univariate polynomial to select for a given standard distribution of ξ_i. If the input parameters x_i can not be interpreted via the Wiener-Askey scheme, copula theory [50] can be applied or a isoprobalistic transformation has to be performed via Rosenblatt transformation [48]. To put it simply, we derive from the independent input random vector $\boldsymbol{x}$ a set of realizations which we compute the basic random vector $\boldsymbol{\xi}$ from and for which we have a look up table to choose the univariate polynomials (e.g. Hermite polynomials for normal distributions and Legendre polynomials for uniform distributions). We can construct the M-variate orthogonal polynomial basis (multivariate polynomials $\Phi_i(\boldsymbol{\xi})$) as a tensor product of the univariate orthonormal polynomials $\phi_{\alpha_j}^{(j)}(\xi_i)$ [51]:

$$\Phi_i(\boldsymbol{\xi}) = \prod_{j=1}^{M} \phi_{\alpha_j}^{(j)}(\xi_i) \tag{17}$$

For high-dimensional problems (M $\geq$ 10) an adaptive sparse PCE can be generated to compensate the curse of dimensionality [52, 53]. These adaptive algorithms select a subset of the polynomial basis during the truncation step and then evaluate if the regression method used, e.g. least angle regression (LARS) [54], gives the minimum leave-one-out-error. The coefficients are then estimated with least squares regression to minimize the L_2-norm.

The mean μ and variance σ^2 of the model output are retrieved from the

coefficients of the individual basis functions:

$$\mu = c_0 \tag{18}$$

$$\sigma^2 = \sum_{i \neq 0} c_i^2 \Phi_i^2 = \sum_{i \neq 0} c_i^2 \tag{19}$$

The sensitivity indices are also calculated with the coefficients and the uni-/multivariate polynomials from the PCE [55]:

$$S1_i = \frac{\sum_{\alpha \in \Im_i} c_\alpha^2 \phi_\alpha^2}{\sum_{k=1}^{p} c_k^2 \Phi_k^2} \tag{20}$$

$$ST_i = \frac{\sum_{\alpha \in \jmath_i} c_\alpha^2 \phi_\alpha^2}{\sum_{k=1}^{p} c_k^2 \Phi_k^2} \tag{21}$$

Note the difference between the calculation of $S1_i$ and ST_i, namely the sets of indices $\Im_i$ and $\jmath_i$:

$$\Im_i = \left\{ \begin{array}{ll} & \alpha_k \quad \mathrm{k} = 1, ..., \mathrm{n} \quad \mathrm{k} \in (i_1, ..., i_s) \\ \boldsymbol{\alpha}: & \\ & \alpha_k \quad \mathrm{k} = 1, ..., \mathrm{n} \quad \mathrm{k} \notin (i_1, ..., i_s) \end{array} \right\} \tag{22}$$

$$\jmath_{i_1, i_2, ..., i_s} = \{\boldsymbol{\alpha} : \alpha_k > 0 \quad \forall k = 1, ..., n \quad k \in (i_1, i_2, ..., i_s)\} \tag{23}$$

The integer set of the tuple $\boldsymbol{\alpha}$ represents each term in the expansion, being the tensor product of univariate orthogonal polynomials , and is defined as follows:

$$\boldsymbol{\alpha} = (\alpha_1, ..., \alpha_n); \quad \alpha_i \geq 0; \quad \sum_{i=1}^{M} \alpha_i \leq P \tag{24}$$

See also Table 9 for more clarification on how the multi-index maps to the basis functions.

The following relation holds for the number of polynomials P of an expansion with degree p and the input vector dimensionality of M:

$$P = \left(\begin{array}{c} M + p \\ p \end{array} \right) = \frac{(M+p)!}{M!p!} \tag{25}$$

For example, the PCE of a two dimensional input vector and a polynomial degree of three leads to the number of polynomials of P=10. As mentioned before, Table 9 lists the basis functions and the corresponding multi-index for this case.

Table 9: Mapping from multiple indices to single index via multi-index for the two dimensional case and a polynomial degree of three

m	p	Basis functions	Multi-index
0	0	$\Phi_0(\xi_1,\xi_2) = 1$	$\alpha_0 = (0,0)$
1	1	$\Phi_1(\xi_1,\xi_2) = \phi_1(\xi_1)$	$\alpha_1 = (1,0)$
2	1	$\Phi_2(\xi_1,\xi_2) = \phi_1(\xi_2)$	$\alpha_2 = (0,1)$
3	2	$\Phi_3(\xi_1,\xi_2) = \phi_2(\xi_1)$	$\alpha_3 = (2,0)$
4	2	$\Phi_4(\xi_1,\xi_2) = \phi_1(\xi_1)\phi_1(\xi_2)$	$\alpha_4 = (1,1)$
5	2	$\Phi_5(\xi_1,\xi_2) = \phi_2(\xi_2)$	$\alpha_5 = (0,2)$
6	3	$\Phi_6(\xi_1,\xi_2) = \phi_3(\xi_1)$	$\alpha_6 = (3,0)$
7	2	$\Phi_7(\xi_1,\xi_2) = \phi_2(\xi_1)\phi_1(\xi_2)$	$\alpha_7 = (2,1)$
8	2	$\Phi_8(\xi_1,\xi_2) = \phi_1(\xi_1)\phi_2(\xi_2)$	$\alpha_8 = (1,2)$
9	3	$\Phi_9(\xi_1,\xi_2) = \phi_3(\xi_2)$	$\alpha_9 = (0,3)$

PRO/II-Python-Interface

The following Python script establishes a COM server connection with the process simulator PRO/II. The sampling hypercube has to be generated by the user and stored as a numpy data object depicted in the code as X.npy. For storing and loading .mat files we recommend the scipy.io class. The X.npy, X.mat, Y.npy and Y.mat files of run number 1 and 6 are attached to the supplementary material of this paper. The Python packages SALib and Chaospy were used to conduct sensitivity analysis and to generate the polynomial chaos expansion and the post-process the SA. We obtained better results (January 2019) with the UQLab Matlab package for the polynomial chaos expansion since this package has the adaptive algorithm implemented which we mention in the theoretical part of this publication.

```
import numpy as np
import win32com.client as win32
import os
import time


#Connect via COM interface
def COMconnect(db_path):
    pro2   = "Nothing"
    pro2db = "Nothing"
    pro2 = win32.Dispatch("SimSciDbs.Database.101")

    pro2.Initialize()

    pro2.SetOption("showInternalObjects","1")
    pro2.SetOption("DoublePrecision","1")

    pro2.Import(os.path.splitext(db_path)[0]+'.inp')

    pro2db = pro2.OpenDatabase(db_path)

    #Get a security license (for better performance)
    pro2.GetSecuritySeat(2)

    return pro2, pro2db


#Disconnect COM interface
def COMdisconnect(pro2, pro2db):
    #Release the security license
    pro2.ReleaseSecuritySeat()

    #Shut down the connection to the COM server
    pro2db = "Nothing"
    pro2   = "Nothing"

    return pro2, pro2db

```

```
    #Evaluate flowsheet with new input vector/matrix
    def evaluate(db_path, NOC, X):

        pro2, pro2db = COMconnect(db_path)
        pro2check    = pro2db.CheckData
        pro2db       = "Nothing"
        pro2run      = pro2.RunCalcs(db_path)
        pro2db       = pro2.OpenDatabase(db_path)


        #Provide array in which the results are stored
        Stream_Product_Bottom_arr = np.zeros(shape=(len(X),1))

        Component_Name_List = [
          "TRIPALM",
          "TRIOLEIN",
          "OLEIC",
          "A-TOCOPH",
          "B-CAROTN"]

        #Check data and run simulations
        for sim_id, sim_id_ in enumerate(Stream_Product_Bottom_arr):

            print("Simulation Number:", sim_id)

            prop_idx = 0 # Property index

            for comp_id, comp_name in enumerate(Component_Name_List):

                print("Component Name:", comp_name)

                Tc_value    = X[sim_id, prop_idx]

                #Change critical temperature in database of component i
                Comp_i = pro2db.ActivateObject("CompIn", comp_name)
                Comp_i.PutAttribute(Tc_value, "CritTempIn")
                Comp_i.Commit(True)
```

```
            Comp_i = "Nothing"

            prop_idx += 1

            Pc_value = X[sim_id, prop_idx]

            #Change critical pressure in database of component i
            Comp_i = pro2db.ActivateObject("CompIn", comp_name)
            Comp_i.PutAttribute(Pc_value, "CritPressIn")
            Comp_i.Commit(True)
            Comp_i = "Nothing"

            prop_idx += 1

            omega_value = X[sim_id, prop_idx]

            #Change acentric factor in database of component i
            Comp_i = pro2db.ActivateObject("CompIn", comp_name)
            Comp_i.PutAttribute(omega_value, "AcenFactorIn")
            Comp_i.Commit(True)
            Comp_i = "Nothing"

            prop_idx += 1

            print("Property values perturbed to...
            Tc:", Tc_value,
            "Pc:", Pc_value,
            "omega:", omega_value)

            pro2check   = pro2db.CheckData
            pro2check   = pro2db.DbsSaveDb
            print("Database saved:", pro2check)
            pro2db = "Nothing"

            print("Return Value of CheckData
            (0=no error, 1=error):", pro2check)

            #Print error messages
```

```
            nMsg = pro2.MsgCount
            if nMsg > 0:
                print(nMsg)
                for i in range(0, nMsg):
                    print("Error message:", pro2.MsgText(i))

            #Run simulation
            pro2run = pro2.RunCalcs(db_path)
            print("Return Value of RunCalc:", pro2run)
            Solutions[sim_id] = pro2run

            nMsg = pro2.MsgCount
            if nMsg > 0:
                print(nMsg)
                for i in range(0, nMsg):
                    print("Error message:", pro2.MsgText(i))

            #OUTPUT
            pro2db = pro2.OpenDatabase(db_path)

            pro2db.CalculateStreamProps("REFINED")

            #beta-carotene fraction at bottom
            Stream_Product_Bottom = pro2db.ActivateObject("Stream", "REFINED")
            Stream_Product_Bottom_arr[sim_id] =
            Stream_Product_Bottom.GetAttribute("LiquidComposition",4)

            print("Stream_Product_Bottom:", Stream_Product_Bottom)
            print("Stream Product Bottom Total Composition Value:",
            Stream_Product_Bottom_arr[sim_id])

            Stream_Product_Bottom = "Deactivate"
            Stream_Product_Bottom = "Nothing"

            np.save('Y_temp.npy', Stream_Product_Bottom_arr)
        #----End of simulation loop----

        pro2, pro2db = COMdisconnect(pro2, pro2db)
```

```
830
        return Stream_Product_Bottom_arr

    if __name__ == "__main__":

835     np.set_printoptions(threshold=np.inf)

        #Path to the ProII file
        db_path = "C:\\Users\\foo\\bar.prz"

840     #Number of components
        NOC = 5

        X = np.load("X.npy")
        print("Size of loaded sample hypercube:", X.shape)
845
        start_time = time.time()

        Y = evaluate(db_path, NOC, X)

850     #Print how much time the evaluations took
        print("--- \%s seconds ---" \% (time.time() - start_time))

        #Save results
        np.save('Y.npy', Y)
```

Table 10: Interaction sensitivity indices describing the effect of parameters between each other

Component	$T_{C,Tripalmitin}$	$P_{C,Tripalmitin}$	$\omega_{Tripalmitin}$	$T_{C,Triolein}$	$P_{C,Triolein}$	$\omega_{Triolein}$	$T_{C,OleicAcid}$	$P_{C,OleicAcid}$	$\omega_{OleicAcid}$	$T_{C,Tocopherol}$	$P_{C,Tocopherol}$	$\omega_{Tocopherol}$	$T_{C,Carotene}$	$P_{C,Carotene}$	$\omega_{Carotene}$
$T_{C,Tripalmitin}$	0	-0.001012 ± 0.010009	-0.002872 ± 0.008885	-0.000499 ± 0.006962	0.000516 ± 0.010158	-0.000480 ± 0.006792	0.000292 ± 0.007102	-0.000061 ± 0.007324	-0.000085 ± 0.007295	-0.000092 ± 0.007284	-0.000084 ± 0.007292	-0.000085 ± 0.007295	0.006562 ± 0.012055	0.002999 ± 0.008984	0.000500 ± 0.005875
$P_{C,Tripalmitin}$		0	-0.035791 ± 0.065794	-0.027521 ± 0.069174	-0.011838 ± 0.073258	-0.033325 ± 0.065804	-0.025538 ± 0.067758	-0.027382 ± 0.067525	-0.027476 ± 0.067439	-0.027459 ± 0.067440	-0.027467 ± 0.067434	-0.027476 ± 0.067439	-0.005904 ± 0.079848	-0.018482 ± 0.066615	-0.018463 ± 0.070776
$\omega_{Tripalmitin}$			0	0.003808 ± 0.014801	0.007271 ± 0.018744	0.004190 ± 0.015002	0.002263 ± 0.016664	0.003215 ± 0.015490	0.003891 ± 0.014805	0.003854 ± 0.014812	0.003890 ± 0.014807	0.003891 ± 0.014805	0.002799 ± 0.015344	0.002756 ± 0.017293	0.003664 ± 0.014012
$T_{C,Triolein}$				0	-0.000379 ± 0.013071	-0.000155 ± 0.011425	-0.002102 ± 0.010245	-0.002205 ± 0.009816	-0.002412 ± 0.009764	-0.002423 ± 0.009763	-0.002413 ± 0.009764	-0.002412 ± 0.009764	-0.008074 ± 0.013898	-0.000965 ± 0.009608	-0.002226 ± 0.010997
$P_{C,Triolein}$					0	-0.040702 ± 0.067030	-0.043563 ± 0.067023	-0.043247 ± 0.066265	-0.042160 ± 0.065215	-0.042180 ± 0.065216	-0.042148 ± 0.065215	-0.042160 ± 0.065215	-0.018967 ± 0.065284	-0.020145 ± 0.068216	-0.028444 ± 0.065761
$\omega_{Triolein}$						0	0.005894 ± 0.007906	0.003757 ± 0.00774	0.005517 ± 0.007610	0.005518 ± 0.007601	0.005516 ± 0.007608	0.005517 ± 0.007610	0.006847 ± 0.010653	0.006268 ± 0.009754	0.006506 ± 0.008606
$T_{C,OleicAcid}$							0	-0.004758 ± 0.012633	-0.004002 ± 0.011637	-0.004019 ± 0.011659	-0.004002 ± 0.011637	-0.004002 ± 0.011637	0.004438 ± 0.011010	-0.006574 ± 0.013759	-0.003803 ± 0.009392
$P_{C,OleicAcid}$								0	-0.002054 ± 0.002137	-0.002051 ± 0.002136	-0.002054 ± 0.002137	-0.002054 ± 0.002137	-0.002182 ± 0.002279	-0.002023 ± 0.002185	-0.001944 ± 0.002147
$\omega_{OleicAcid}$									0	0.000080 ± 0.000129	0.000080 ± 0.000129	0.000080 ± 0.000129	0.000080 ± 0.000129	0.000080 ± 0.000129	0.000080 ± 0.000129
$T_{C,Tocopherol}$										0	0.000030 ± 0.000232	0.000030 ± 0.000231	-0.002231 ± 0.000593	0.000006 ± 0.000248	0.000002 ± 0.000231
$P_{C,Tocopherol}$											0	0.000078 ± 0.000143	0.000062 ± 0.000174	0.000073 ± 0.000144	0.000094 ± 0.000150
$\omega_{Tocopherol}$												0	0.000080 ± 0.000129	0.000079 ± 0.000129	0.000079 ± 0.000129
$T_{C,Carotene}$													0	0.056279 ± 0.114991	0.067831 ± 0.124408
$P_{C,Carotene}$														0	0.002970 ± 0.039410
$\omega_{Carotene}$															0