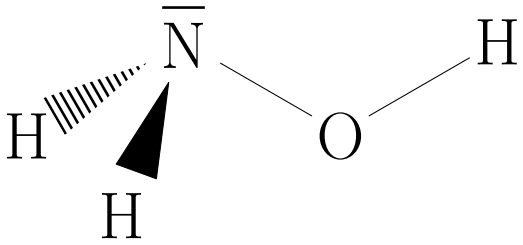
University of Applied Sciences and Arts Northwestern Switzerland
School of Life Sciences

Thermokinetic model of hydroxylamine decomposition in the catalytic oxidation of ammonia with titanium silicalite-1

Bachelor Thesis

Florian Walthert[1]



Muttenz, October 23, 2020

[1]Degree program *Molecular Life Sciences*,
Institute for Chemistry and Bioanlaytics

Principal
Dr Fabian Meemken

Expert
Dr Fabian Meemken

Supervisor
Dr Andreas Zogg, FHNW

Confidentiality status
Non-confidential

Declaration of independence
I hereby affirm that I have independently written the work presented by me, that I have given full details of the sources, Internet sources and aids used, and that I have indicated the parts of the work – including tables, maps and illustrations – which are taken from other works or the Internet in wording or sense, in any case indicating the source as borrowed.

Muttenz, October 23, 2020

Place, Date	Signature

Abstract

Hydroxylamine is a key molecule for some important chemical processes. An example is the synthesis of ε-caprolactam, which is required for the production of *Nylon*. Since hydroxylamine decomposes thermally, its transport and storage are problematic. It is also problematic for the synthesis of hydroxylamine from ammonia with titanium silicalite-1. The goal of this study was to investigate the exothermic decomposition reaction. The heat of the reaction was measured in multiple DSC experiments. During the decomposition, vast quantities of gases are released. To measure the pressure rise, an explosion reactor was employed. Three reaction hypotheses have been tested. With the use of a mathematical model one hypothesis could be proven. The material influence on the decomposition reaction was examined for PEEK, stainless steel and hastelloy C22. From the DSC measurements, a descriptive model and isoconversional kinetics were obtained.

Hydroxylamin ist ein Schlüsselmolekül für einige wichtige chemische Prozesse. Ein Beispiel ist die Synthese von ε-Caprolactam, das für die Herstellung von *Nylon* benötigt wird. Da Hydroxylamin sich thermisch zersetzt, sind sein Transport und seine Lagerung problematisch. Problematisch ist es auch für die Synthese von Hydroxylamin aus Ammoniak mit Titansilikalit-1. Das Ziel dieser Studie war es, die exotherme Zersetzungsreaktion zu untersuchen. Die Wärme der Reaktion wurde in mehreren DSC-Experimenten gemessen. Bei der Zersetzung werden große Mengen an Gasen freigesetzt. Um den Druckanstieg zu messen, wurde ein Explosionsreaktor eingesetzt. Es wurden drei Reaktionshypothesen getestet. Mit Hilfe eines mathematischen Modells konnte eine Hypothese gestützt werden. Der Materialeinfluss auf die Zersetzungsreaktion wurde für PEEK, rostfreien Stahl und Hastelloy C22 untersucht. Aus den DSC-Messungen wurden ein deskriptives Modell und eine isokonversionelle Kinetik gewonnen.

Acknowledgment

First of all, I would like to thank all the people from the industrial partner, who helped me with a lot of courteousness and spontaneity. My expert Dr Fabian Meemken assisted me throughout the project and gave me very valuable suggestions. Even in this complicated time, he organized for me to visit the industrial partners site. I really appreciate the help and expertise Dr Eberhard Irle contributed to this project. I am also very grateful for all the measurements Damian Venetz expeditiously conducted.

I gratefully acknowledge the guidance of Dr Andreas Zogg, who helped me tremendously in the first phase of the project. The extensive discussions with him shaped the direction this work went. In the matter of the DSC and all the other experimental challenges, Kai Wegmann was of great assistance. I thank Dr Christelle Jablonski for swiftly ordering all the chemicals I required. I am grateful to Gonzalo Álvarez Pérez for lending me his super quantum computer.

At last, I would like to thank my brother Mario Walthert for proofreading this work, even though he is not particularly interested in the topic.

Contents

List of Figures

List of Tables

List of Abbreviations

Abbr.	Abbreviation
APTAC	Automatic pressure tracking adiabatic calorimeter
aq.	Aqueous solution
CS	Corresponding state
CSTR	Continuously stirred tank reactor
DBD	Dielectric barrier discharge
DDBST	Dortmund Data Bank (DDBST GmbH)
DLC	Diamond like carbon
DSC	Differential scanning calorimetry
DTA	Differential thermal analysis
EP	Equivalence point
eq.	Equivalent
Exp	Experiment
EXR	Explosion reactor
FI	Flow indicator
GC	Group contribution
HA	Hydroxylamine
HC22	Hastelloy C22
HCC	Heating/cooling circuit
HP	Hydrogen peroxide
MC	*Monte Carlo*
MW	Molecular weight
NIST	National Institute of Standards and Technology
NRMSD	Normalized root-mean-square deviation
OF	Overflow, related to the *Vögtlin* flowmeters
Pos.	Position
PEEK	Polyether ether ketone
PPDS	Physical property data services (TÜV Süd)

RG	Test tube
RT	Room temperature
solu	Solution
solv	Solvent
SS	Stainless steel
t-BuOH	*tert*-butanol
TI	Temperature indicator
TS-1	Titanium silicalite-1
t_R	Retention time (chromatographic parameter)
VLE	Vapor-liquid equilibrium
VOC	Volatile Organic Compound
$wt\%$	Weight percentage

List of Symbols

A	molar *Helmholtz* free energy
A	area
A	frequency factor
α	reaction conversion (-)
c_p	molar heat capacity at constant pressure ($^{J}/_{mol*K}$)
c_V	molar heat capacity at constant volume ($^{J}/_{mol*K}$)
E_B	bond energy
φ	phi factor or thermal inertia
ϕ	heat flux (the same as $\dot{Q}$)
G	molar free enthalpy ($^{kJ}/_{mol}$)
Γ	specific gas production ($^{mol}/_{kg}$)
GP	molar gas production ($^{L}/_{mol}$)
H	molar enthalpy ($^{kJ}/_{mol}$)
H_{i,H_2O}	*Henry's law* constant for substance i in water
j	current density

k_r	rate constant of reaction r
k_1	rate constant of forward reaction
k_{-1}	rate constant of backward reaction
K_d	dissociation constant
l_B	bond length
m	mass
mw	molecular weight ($^{kg}/_{mol}$)
μ_i	chemical potential of component i ($^{kJ}/_{mol}$)
n_i	number of moles of component i (mol)
N_c	number of components
N_r	number of reactions
$\nu_{i,r}$	stoichiometric coefficient of component i in reaction r
ω	acentric factor
p	absolute pressure
p^*	vapor pressure of pure compound
p_c	critical pressure
p_r	reduced pressure (p/p_c)
p_i	partial pressure of component i
pK_a	acid dissociation constant
pK_b	base dissociation constant
P	partition coefficient
$P^{(g)}$	modified partition coefficient for the gas phase
q	molar heat ($^{kJ}/_{mol}$)
$\dot{q}$	molar heat flow ($^{kW}/_{mol}$)
Q	heat (J)
$\dot{Q}$	heat flow (W)
ρ	density ($^{kg}/_{m^3}$)
R	universal gas constant (8.314 $^{J}/_{mol*K}$)
S	molar entropy ($^{J}/_{mol*K}$)

t	time
t_{ind}	induction time
T	temperature
T_b	boiling point
T_c	critical temperature
T_{fl}	flash point
T_{fus}	fusion point
T_r	reduced temperature (T/T_c)
TMR_{ad}	Time to maximum rate in adiabatic scenario
U	molar internal energy ($^{kJ}/_{mol}$)
V	volume (m^3)
V_c	critical volume
V_r	reduced volume (V/V_c)
V_m	molar volume ($^{m^3}/_{mol}$)
ξ_r	extent of reaction r
Z	compressibility factor
Z_c	critical compressibility factor

Subscripts

0	initial value
b	boiling
B	value for bond
f	value for the formation reaction
fl	flash
fus	fusion
i	value for component i
m	molar
$n_{j\neq i}$	at constant number of moles for all components but i
p	at constant pressure

r	reduced property
r	number of the chemical reaction
R	value for the chemical reaction
sol	dissolution
sub	sublimation
sys	value for a defined system
T	at constant temperature
v	vaporization
V	at constant volume

Superscripts

(g)	value for the gaseous phase
(l)	value for the liquid phase
(s)	value for the solid phase
$*$	property for pure compound
$\ominus$	reference state

1 Introduction

1.1 Background

For the synthesis of an active ingredient, the project partner purchases hydroxylamine from a supplier in aqueous solution. In order to optimize the synthesis, however, it would be better to produce the hydroxylamine in situ. The process has to produce the free base so that no salts are formed as by-products during the oximation.

1.2 Previous work

In the preceding work [**walthert_elektrochemische_2019**] the focus was on a process that *Besson* discovered by chance[**besson_action_1911**]. With the help of an ozone generator, it was attempted to produce hydroxylamine from moist ammonia gas. For this purpose, various parameters such as the flow rate, gas composition or current frequency were varied. However, the experiment could not be reproduced. With other equipment or more drastic experimental conditions it might be possible. Also the electrolytic reduction of nitric acid did not prove successful. The probably necessary use of mercury electrodes and the resulting salt wastes make this method not very promising in the first place.

The selective oxidation of ammonia with TS-1 and hydrogen peroxide is more promising. The facts described in the literature [**roffia_cyclohexanone_1990**],[**mantegazza_selective_1994**] were more or less confirmed in the previous work. Also the ammoximation of cyclohexanone works very fast and selectively. The disadvantage of the separate hydroxylamine synthesis is the very high ammonia excess that is required. The resulting solution has a low hydroxylamine content. It still needs to be investigated how to produce a 50 % hydroxylamine solution. A distillative enrichment of the hydroxylamine would be possible, but it would be challenging from a safety point of view. There are numerous reports of accidents during the distillation of HA. Furthermore, the distillation of hydroxylamine is patented by BASF (DE19733681). The inital trials carried out during the preliminary work only showed a low yield of hydroxylamine. Most likely due to a significant decomposition of hydroxylamine in parallel to the syntheis reaction.

It can be said that the optimum conditions for the selective oxidation of ammonia still have to be found. However, the process is promising because all the reactants (ammonia and hydrogen peroxide) are very cheap and no waste products are produced. If a reasonably ammonia-free aqueous hydroxylamine solution can be produced, it can easily be fed into the existing process of the project partner.

1.3 Task

For the reasons mentioned above, the decomposition of hydroxylamine needs to be studied more closely, both computationally and experimentally. It may be possible to reduce decomposition under pressure. This also applies to the decomposition of hydrogen peroxide.

At first an extensive literature research to find existing physical and thermodynamic data for the involved substances is to be carried out. Based on the findings, calculations to roughly estimate the behavior of hydroxylamine at increased temperature need to be done. To verify those estimations, the pure hydroxylamine solution as well as mixtures resembling the reaction mixture (ammonia, titanium silicalite-1, hydrogen peroxide) are to be experimentally examined in regards to their thermodynamic behavior. Therefore, it is crucial to set up an appropriate apparatus to measure the decomposition reactions. An existing explosion reactor can be slightly modified to meet the requirements. The sample mixtures can therein be heated up in an isochoric system. The temperature and pressure can be measured. To analyze the composition of the resulting gas phase, it is necessary to take samples.

Additionally, DSC (differential scanning calorimetry) is used to measure the decomposition of hydroxylamine. The activation energies and reaction rate constants can be obtained from a model based on the evaluation of the heat flow curves. More DSC measurements are conducted to investigate the influence of other substances on the decomposition reaction.

With sufficient data with regards to pressure, temperature, gas phase composition and heat flows a thermokinetic model of the decomposition of hydroxylamine can be established. This model can be used as a foundation for the engineering of a hydroxylamine in-situ generator for the industrial partner. For example, the catalytic reaction could be carried out under pressure, to inhibit the decomposition by moving the equilibrium to the educt side. When the decomposition reactions of hydroxylamine is well understood, measurement of the catalytic oxidation can be done. The oxidation combines three reactions. The oxidation reaction to form hydroxylamine and the decomposition reactions of hydroxylamine and hydrogen peroxide. The resulting pressure rise can be solely contributed to the decomposition reactions, i.e. the conversions of the decomposition reactions can be measured from the pressure rise. Therefore, it should be possible to derive a thermokinetic model from analysis of the liquid phase only.

Furthermore, literature about the hydroxylamine decomposition is scarce. This work aims to find the existing information and expand it. Physical and thermodynamic data is crucial for the safe incorporation of a substance in any process.

2 Theoretical background

2.1 Hydroxylamine

2.1.1 General

Hydroxylamine (HA) is a derivative of ammonia and forms colorless and odorless crystal chunks at room temperature. The melting point is 33 °C. It is well soluble in water but even in dissolved form it is not stable for a long time[**wiley-vch_verlag_gmbh_&_co._kgaa_hydroxylamine_2000**]. Its basicity ($pK_S = 5.9$ [**international_congress_on_chemical_and_biological_aspects_of_vitamin**] is lower than that of ammonia ($pK_S = 9.2$ [**hall_correlation_1957**]). Resistant hydroxylamine salts are used in technology for the synthesis of oximes from carbonyls and in analytics. Hydroxylamine can react with reducing agents as well as with various oxidizing agents. This results in ammonia and water or NO_x-species and nitrates[**davis_oxidationreduction_1951**],[**raman_oxidation_2005**]. Thus, for many reactions very specific reaction conditions have to be kept and the reactions do not run uniformly. This makes the analysis of hydroxylamine very difficult. However, colorimetric methods are suitable for measuring small amounts of hydroxylamine (<0.1 g/L)[**wiley-vch_verlag_gmbh_&_co._kgaa_hydroxylamine_2000**],[**leithe_elemente_1957**] More about the synthesis and the analytics of hydroxylamine can be found in appendix A.1.3.

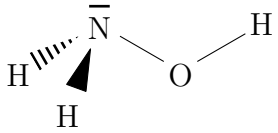


Figure 1: Structural formula of hydroxylamine

2.1.2 History and industrial importance

Hydroxylamine was first described in 1865[**lossen_ueber_1865**] and first isolated as a free base in 1881[**bruhl_ueber_1893**]. *Raschig* succeeded in the industrial production of hydroxylamine sulfate in 1908 in Germany[**seel_kinetik_1963**].

In 1938, almost 40 years after its discovery, ε-caprolactam was first used to produce a spinnable polymer called *Perlon*. A continuous phenol-based process for the synthesis of ε-caprolactam was then developed. The phenol is converted to cyclohexanone in two reaction steps. With HA-sulfate the cyclohexanone oxime is produced. Then with oleum and finally with caustic soda ε-caprolactam is synthesized. The last step is the acid-catalyzed Beckmann rearrangement from oxime to lactam. The product was used for airplane tires and parachutes in view of the Second World War[**moesdijk_catalytic_1979**],[**li_beckmann_2015**].

From 1954 onward, the by-product ammonium sulfate was also purified and sold in the form of a white solid as a fertilizer. The demand for ε-caprolactam continued to rise after 1959 and amounted to 35'000 tons per year in 1963. This also resulted in

ever larger quantities of ammonium sulfate. Research therefore concentrated on a by-product-free process. The synthesis of hydroxylamine is usually carried out at low pH values. A pH value of 4.5 is ideal for oximation, the formation of the oxime, which is why the hydroxylamine salt had to be isolated. The newly developed method allows to perform the oximation in the same vessel as the hydroxylamine synthesis by adding buffers[**moesdijk_catalytic_1979**].

To date, numerous other processes have been discovered and discarded. Most processes in the USA, Europe and Asia are based on the reduction of NO[**moesdijk_catalytic_1979**]. Today, a global production of 800'000 tons of hydroxylamine per year is assumed[**wiley-vch_verlag_gmbh_&_co._kgaa_hydroxylamine_2000**].

2.1.3 Selective oxidation with TS-1

The catalyst titanium silicalite-1 (TS-1) was discovered by ENI researchers[**taramasso_preparation_nodate**] (US Patent 4410501). It can be used for numerous reactions, such as phenol hydroxylations, olefin epoxidations, cyclohexanone ammoximations, or the reaction of secondary alcohols to ketones[**mantegazza_selective_1999**]. This catalyst has the decisive advantage of environmental compatibility, especially compared to inorganic redox reagents such as permanganates and dichromates. The cleaner catalytic variant does not produce excess environmentally harmful salts. Normally soluble metal salts are used for catalytic oxidation. The TS-1 catalyst, however, is a solid phase catalyst which acts heterogeneously. Besides the catalyst, hydrogen peroxide is used as oxidizing agent, which generates the completely harmless waste product water[**sheldon_redox_1997**].

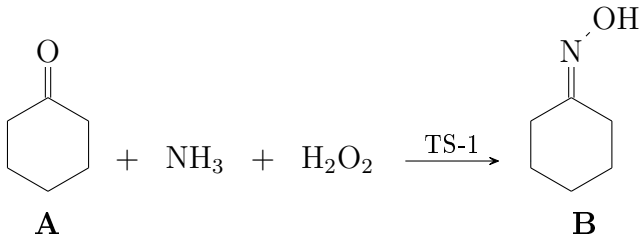
In the production of TS-1, titanium atoms are inserted into a zeolite lattice. Because this molecular sieve is clearly defined, the metal species unfolds its catalytic effect very specifically and is simultaneously immobilized[**mantegazza_selective_1999**]. For the preparation of the catalyst, ethyl orthosilicates are used, as with conventional zeolite. In addition, a stoichiometric amount of tetrabutyl titanate is used as a source of titanium. This mixture is polymerized by hydrolysis and calcination[**deniz_effect_2013**],[**jin_green_2018**]. This produces aluminium-free zeolite with regularly incorporated titanium atoms[**dal_pozzo_ts-1_2002**]. Since titanium has the same oxidation state as silicon, no charge results in the lattice due to the substitution[**clerici_liquid_2013**].

Mechanistically, it is assumed that the hydrogen peroxide tends to transfer oxygen to a nucleophilic substrate. The catalyst weakens the polarization of the peroxide bond and thus enhances its electrophilicity. Another possibility would be the formation of an oxene from the peroxide, which is able to oxidize a nucleophile[**roffia_cyclohexanone_1990**],[**schirmann_hydrogen_1981**].

In the literature there is only one example of the use of a peroxide other than hydrogen peroxide. *t*-butyl hydroperoxide with TS-1 in BF_4 is used for cyclohexanone ammoximation. However, yields are significantly lower than with hydrogen peroxide in aqueous solution[**wang_oxidation_2008**]. Elsewhere is also talk of an inertness of

t-butyl hydroperoxide to TS-1. This is regarded as proof for the formation of a TS-1$\cdot H_2O_2$ complex[**clerici_liquid_2013**].

Cyclohexanone ammoximation The reaction of cyclohexanone (**A**) to cyclohexanone oxime (**B**) with ammonia, hydrogen peroxide and TS-1 (see reaction equation (2-1)) takes place very rapidly (under 30 min) at 60-80 °C. The higher the temperature, the faster the reaction proceeds. t-BuOH/H_2O mixtures are usually named as the optimal solvent. The molar ratio $NH_3/H_2O_2 = 2$ has proved to be successful. The yields (related to hydrogen peroxide) reach up to 94 %[**dal_pozzo_ts-1_2002**],[**roffia_cyclohexanone_1990**]. The continuous large-scale implementation has also already been carried out[**trivellone_process_nodate**]. Regarding the mechanism of ammoximation of cyclohexanone (**A**) there was initially no consensus. There were two possible hypotheses. The imine route said that the ketone forms an intermediate imine which is adsorbed on the catalyst and oxidized with the peroxide to the oxime **B**. The alternative was the hydroxylamine route. Here the ammonia is first oxidized catalytically and selectively. The formed hydroxylamine then reacts with the cyclohexanone (**A**) to oxime **B**. However, a two-step synthesis proved the second route[**dal_pozzo_ts-1_2002**].



A **B** (2-1)

Hydroxylamine formation The separate synthesis of hydroxylamine by the selective oxidation of ammonia (see reaction equation (A.1-60)) achieves lower yields (based on hydrogen peroxide) of up to 64 % compared to direct further use for oxide synthesis. Optimally t-BuOH or water is used as solvent. The yield is proportional to the molar ratio NH_3/H_2O_2. This means that the greater the ammonia excess, the better the conversion of the peroxide, or the hydroxylamine yield. If too much peroxide is present, nitrogen oxides are formed from the hydroxylamine (see figure 2). At the investigated temperatures of 50 to 80 °C it was shown that higher temperatures cause a faster and better conversion, but also cause the hydroxylamine content to sink again more quickly. The catalyst should ideally have a Ti content of 2-3 % and be used in a concentration of 30 g/L[**mantegazza_selective_1994**],[**deniz_effect_2013**].

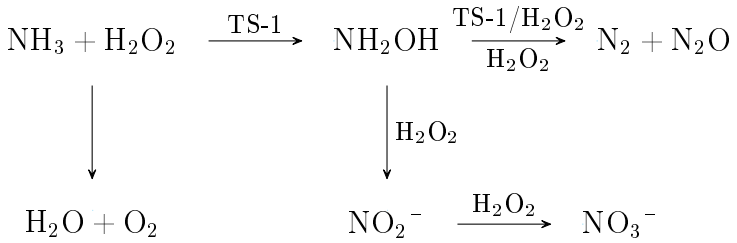


Figure 2: Reaction Network of Ammoximation

2.1.4 Decomposition

Since hydroxylamine has been used industrially, several explosion accidents happened, mostly during its distillation. It is known that hydroxylamine decomposes at elevated temperature[**cisneros_adiabatic_2002**]. Several reactions may occur during hydroxylamine decomposition. One possibility is the oxidation reaction with oxygen.

$$2\,NH_2OH^{(g)} + 0.5\,O_2 \longrightarrow 3\,H_2O^{(g)} + N_2 \tag{2-2}$$

However, a study measured the decomposition reaction in an APTAC (automatic pressure tracking adiabatic calorimeter) in vacuum and air and found no significant difference. The formed oxygen containing nitrogen compounds (N_2O and NO) are formed with the oxygen atom from within the hydroxylamine molecule[**cisneros_effect_2002**]. Another possibility is the dependence on pH[**jones_nitrogen_1973**]. Based on the two fundamental reactions (2-3f) and (2-4i) two different decomposition mechanisms were proposed[**chakrabarty_multiscale_2016**]. Their names are derived from the first reaction steps that show the pH dependance. The mechanism for the **alkaline reaction** is as follows[**wei_reaction_2004**]:

$$NH_2OH + OH^- \longrightarrow NH_2O^- + H_2O \qquad \Delta_R H = -109.7\ {}^{kJ}/_{mol} \tag{2-3a}$$
$$NH_2O^- + NH_2OH \longrightarrow NH_2NHOH + OH^- \qquad \Delta_R H = -14.6\ {}^{kJ}/_{mol} \tag{2-3b}$$
$$NH_2NHOH \longrightarrow HNO + NH_3 \qquad \Delta_R H = 29.7\ {}^{kJ}/_{mol} \tag{2-3c}$$
$$HNO + NH_2OH \longrightarrow NH_2O^\bullet \qquad \Delta_R H = -3.8\ {}^{kJ}/_{mol} \tag{2-3d}$$
$$2\,NH_2O^\bullet \longrightarrow N_2 + 2\,H_2O \qquad \Delta_R H = -477.8\ {}^{kJ}/_{mol} \tag{2-3e}$$
$$3\,NH_2OH \longrightarrow N_2 + NH_3 + 3\,H_2O \qquad \Delta_R H = -576.1\ {}^{kJ}/_{mol} \tag{2-3f}$$

The overall reaction is exothermal and therefore consistent with the observations of an adiabatic temperature rise. The same can be said for the mechanism in acidic solution

(**acid reaction**):

$$\mathrm{NH_3OH^+} \longrightarrow \mathrm{NH(triplet) + H_3O^+} \quad \Delta_R H = 301.7\ ^{kJ}/_{mol} \quad \text{(2-4a)}$$
$$\mathrm{H_3O^+ + NH_2OH} \longrightarrow \mathrm{NH_3OH^+ + H_2O} \quad \Delta_R H = -134.3\ ^{kJ}/_{mol} \quad \text{(2-4b)}$$
$$\mathrm{NH(triplet) + NH_2OH} \longrightarrow \mathrm{^{\bullet}NH_2 + NH_2O^{\bullet}} \quad \Delta_R H = -107.1\ ^{kJ}/_{mol} \quad \text{(2-4c)}$$
$$\mathrm{NH_2O^{\bullet} + NH_2OH} \longrightarrow \mathrm{NH_2NHOH + {}^{\bullet}OH} \quad \Delta_R H = 73.2\ ^{kJ}/_{mol} \quad \text{(2-4d)}$$
$$\mathrm{{}^{\bullet}OH + NH_2OH} \longrightarrow \mathrm{H_2O + NH_2O^{\bullet}} \quad \Delta_R H = -197.5\ ^{kJ}/_{mol} \quad \text{(2-4e)}$$
$$\mathrm{NH_2NHOH} \longrightarrow \mathrm{HNO + NH_3} \quad \Delta_R H = 29.7\ ^{kJ}/_{mol} \quad \text{(2-4f)}$$
$$\mathrm{NH_2O^{\bullet} + {}^{\bullet}NH_2} \longrightarrow \mathrm{HNO + NH_3} \quad \Delta_R H = -154.8\ ^{kJ}/_{mol} \quad \text{(2-4g)}$$
$$\mathrm{2\,HNO} \longrightarrow \mathrm{N_2O + H_2O} \quad \Delta_R H = -366.9\ ^{kJ}/_{mol} \quad \text{(2-4h)}$$
$$\mathrm{4\,NH_2OH} \longrightarrow \mathrm{N_2O + 2\,NH_3 + 3\,H_2O} \quad \Delta_R H = -556.0\ ^{kJ}/_{mol} \quad \text{(2-4i)}$$

The reaction enthalpies were calculated using quantum mechanical models[**chakrabarty_multiscale_2016**].

Cisneros et al. studied the decomposition of 50 % hydroxylamine solution. By quantifying the resulting gases and the ammonia in the solution, they were able to formulate the **empirical reaction equation** 2-5. When using $\Delta H_f^{\ominus}(NH_2OH^{(l)}) = -106.7\ ^{kJ}/_{mol}$[**reinmuth_thermochemistry_1936**] the heat of reaction is $\Delta_R H^{\ominus} = -124\ ^{kJ}/_{mol}$[**cisneros_decomposition_2003**].

$$\mathrm{157\,NH_2OH^{(l)} \longrightarrow 61\,{NH_3}^{(g)} + 35\,{N_2}^{(g)} + 12\,{N_2O}^{(g)} + 2\,{NO}^{(g)} + 143\,{H_2O}^{(g)} + {H_2}^{(g)}} \quad \text{(2-5)}$$

Influence of metals The hydroxylamine decomposition reaction is strongly accelerated by metals. By using both an inconel sheath and teflon coated thermocouple in a calorimetric measurement of the hydroxylamine decomposition, it was found that the onset temperature was lower and the self-heat rates were higher when the nickel based alloy was in direct contact with the solution. When measured in stainless steel cells the maximum self-heat rate was strongly elevated. In the liquid residue a brownish residue was observed. This could indicate the formation of iron hydroxide (FeO(OH)), which means that hydroxylamine acted as oxidation agent in the oxidation of Fe^0 to Fe^{3+} whilst itself being reduced to ammonia. In titanium measuring cells the self-heat rates of the hydroxylamine solutions were either similar (without stabilizer) or slightly higher (with stabilizer) than those observed in stainless steel cells[**cisneros_adiabatic_2002**].

Next to the above mentioned indications of heterogeneous catalysis, there are also signs for homogeneous catalysis of metal ions. A violent reaction took place at room temperature with iron ion concentration of 197 *ppm*. It can be broken down into two exothermic reactions. The iron ion presence affects gas product distribution. The relative amount of nitrous oxide produced increases[**cisneros_adiabatic_2002**].

Influence of the catalyst TS-1 *Xu et al.* studied the effec of microporous titanosilicates on the formation and decomposition of hydroxylamine. They found that out of the

four investigated titanosilicates Ti-MOR and TS-1 are extremely effective catalysts in the formation of hydroxylamine from ammonia and hydrogen peroxide. Still they exhibit low catalytic activity for the hydroxylamine decompostion[**xu_distinctions_2014**].

Stabilizer *BASF* uses tri(4-(2,3-dihydroxyphenyl)-4-oxabutyl)amine, tri(4-(2,3-dihydroxyphenyl)-4-keto-3-azabutyl)amine or salts thereof in its hydroxylamine solution. The compounds concentration should range from 10 to 20'000 *ppm*. While for 15 hours at 100 °C the hydroxylamine content of a solution without stabilizer falls from 50 % to 25 %, it only drops to 49.7 % with those stabilizers added[**weber_stabilizing_801**]. There is nothing mentioned as to how the stabilizer works mechanistically. However, *Cisneros et al.* found out that the stabilizer lowers the maximum self-heat rate significantly when measuring in stainless steel cells. The fact that in titanium measuring cells the self-heat rates similar without stabilizer and slightly higher with stabilizer than those observed in stainless steel cells indicates that the stabilizer has an effect specifically aimed at iron, but not necessarily exclusively.

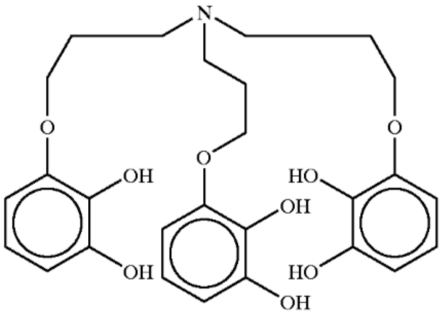


Figure 3: Stabilizer tri(4-(2,3-dihydroxyphenyl)-4-oxabutyl)amine[**weber_stabilizing_801**]

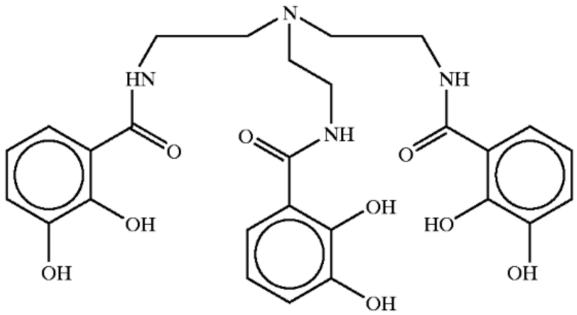


Figure 4: Stabilizer tri(4-(2,3-dihydroxyphenyl)-4-keto-3-azabutyl)amine[**weber_stabilizing_801**]

Another patent describes the storage stabilizer and antioxidant 3,4-dihydroxybenzoic acid. This compound is claimed to work at high temperatures and high hydroxylamine concentration in the presence of metal impurities such as iron[**aoki_stabilizer_2006**].

2.1.5 Substance data of hydroxylamine

Due to its potential for thermal explosions hydroxylamine is not characterized very well. Especially data that would be measured at temperatures above 60 °C is scarce. Only an extensive literature research revealed most of the required data. In this section the available information about physical and thermodynamic properties either measured or calculated are collected. In table 1 the measured data from several sources is shown. Unfortunately, most of the values only have one source. With the methods and equations described in appendix A.1.1 some physical and thermodynamic properties were calculated.

Joback and Reid *Joback* is a GC method with eleven output values[**joback_estimation_1987**]. Some of the available group contributions for an alcohol-group ($-OH$) and an amine-group ($-NH_2$) are Δ_T, Δ_p, Δ_V, Δ_{NBP}, Δ_M, Δ_H, and Δ_G. The output values obtained from *ChemCAD*, calculated with *Matlab* and with the *JRgui* are shown in table 2. The process simulation software *ChemCAD* uses the *Joback* method for hydroxylamine. The *ChemCAD*-value for the normal boiling point appears to be calculated with a different method or is empirical. *Shi and Borchardt* have written the *Joback and Reid* method into a graphical user interface program (*JRgui*) based on *Python*[**shi_jrgui_2017**] (relevant code section in appendix A.38). The input can be done manually by selecting the two relevant functional groups for hydroxylamine (input mask see appendix A.39).

Calculation of substance data of hydroxylamine The methods of appendix A.1.1 were applied to hydroxylamine where it was possible. In table 3 the results of the calculations are listed (calculations see appendix A.14). There are significant deviations (e.g the critical temperature with the *Ambrose* method). Most of the methods are applicable for organic molecules. The group contributions are therefore regressed to alcohols and organic amines, respectively.

The plot of the *Antoine* fit, different estimation and interpolation methods of the heat capacity and the results of the estimation of the free Gibbs energy can be found in the appendix A.3.

Table 1: Collection of physical and thermodynamic properties of hydroxylamine (for the symbol description see preamble v)

Quantity	Value	Unit	Annotation	Ref.
T_b	142	°C	extrapolated	[jones_nitrogen_1973]
T_{fus}	32.05	°C		[jones_nitrogen_1973]
	33.05	°C		[weiss_hazardous_1987]
p^*	0.36	*mbar*	at 0 °C	[back_determination_1965]
	7.1	*mbar*	at 32 °C	[jones_nitrogen_1973]
	13	*mbar*	at 47.2 °C	[jones_nitrogen_1973]
	53	*mbar*	at 64.6 °C	[jones_nitrogen_1973]
	133	*mbar*	at 77.5 °C	[jones_nitrogen_1973]
	533	*mbar*	at 99.2 °C	[jones_nitrogen_1973]
$\rho^{(l)}$	1.204	$^{g}/_{mL}$	at 33 °C	[jones_nitrogen_1973]
ω	0.649	-		[yaws_thermophysical_2014]
$\Delta G_f^{\ominus,(g)}$	-23.4	$^{kJ}/_{mol}$		[latimer_oxidation_1938]
ΔH_{fus}	16.5	$^{kJ}/_{mol}$	at 32 °C	[latimer_oxidation_1938]
ΔH_{sub}	64.2	$^{kJ}/_{mol}$	at 0 and 32 °C	[latimer_oxidation_1938]
ΔH_{sol}	15.9	$^{kJ}/_{mol}$		[weiss_hazardous_1987]
ΔH_v	47.7	$^{kJ}/_{mol}$		[latimer_oxidation_1938]
S_{sub}	169.2	$^{J}/_{mol*K}$	at 0 and 32 °C	[latimer_oxidation_1938]
$S^{\ominus,(g)}$	235.9	$^{J}/_{mol*K}$		[giguere_infrared_1952]
$c_p^{\ominus,(g)}$	49	$^{J}/_{mol*K}$		[giguere_infrared_1952]
V_m	27.4	$^{cm^3}/_{mol}$		[jones_nitrogen_1973]
K_D	1.07×10^{-8}	-	at 20 °C	[jones_nitrogen_1973]
$pK_a(NH_3OH^+)$	6.04	-	at 20 °C	[robinson_ionization_1961]
pK_b	8.13	-	at 20 °C	[robinson_ionization_1961]
$pH(50\ \%\ aq.)$	11	-		[cisneros_adiabatic_2002]
$l_B(N-O)$	1.46	Å		[giguere_infrared_1952]
E_B	256.7	$^{kJ}/_{mol}$	H_2N-OH	[latimer_oxidation_1938]
	213.5	$^{kJ}/_{mol}$	$HO-OH$	[latimer_oxidation_1938]
	251.2	$^{kJ}/_{mol}$	H_2N-NH_2	[latimer_oxidation_1938]
T_{fl}	129	°C	explosion	[lewis_saxs_1992]
$\Delta H_f^{\ominus,(s)}$	-114.3	$^{kJ}/_{mol}$		[jones_nitrogen_1973]
$\Delta H_f^{\ominus,(l)}$	-106.8	$^{kJ}/_{mol}$		[reinmuth_thermochemistry_1936]
	-90.9	$^{kJ}/_{mol}$		[kaye_encyclopedia_1982]
$\Delta H_f^{\ominus,(g)}$	-42.7	$^{kJ}/_{mol}$		[latimer_oxidation_1938]

Table 2: Joback output

Quantity	Unit	ChemCAD	Chemeo	Matlab	JRgui
T_c	°C	300.9	270.9	270.9	271.2
p_c	bar	137	87.5	87.5	87.5
V_c	$^{cm^3}/_{mol}$	80		83.5	83.5
T_{fus}	°C	33.1	-39.3		-39.3
T_b	°C	109.9	90.0	90.0	91.2
$c_p^{(g)}$	$^{J}/_{mol*K}$		$44.8(T_b)$		40.2
ΔH_{fus}	$^{kJ}/_{mol}$				5.041
$\Delta H_v(T_b)$	$^{kJ}/_{mol}$		42.9		42.9
$\Delta H_f^{(s)}$	$^{kJ}/_{mol}$	-114.2	-161.8		
$\Delta G_f^{(s)}$	$^{kJ}/_{mol}$	-67	-121.3		
$\Delta H_f^{\ominus,(g)}$	$^{kJ}/_{mol}$				-141.8
$\Delta G_f^{\ominus,(g)}$	$^{kJ}/_{mol}$				-121.3

Table 3: Calculated physical and thermodynamic properties of HA

Quantity	Value	Unit	Method	Eq.	Annotation
T_c	301	°C	Guldberg rule	(A.1-6)	
	355	°C	Simple correlation	(A.1-5)	calculated from $T_b = 109.9C$
	975.5	°C	Ambrose	(A.1-8)	GC for $-NH_2$ and $-OH$
	270.9	°C	Joback	(A.1-7)	
p_c	87.51	bar	Joback	(A.1-7)	
V_c	83.5	$^{cm^3}/_{mol}$	Joback	(A.1-7)	
T_b	90.96	°C	Joback	(A.1-10)	
	59	°C	Nannoolal	(A.1-7)	
$c_p^{\ominus,(g)}$	42.14	$^{J}/_{mol*K}$	Benson	(A.1-7)	GC for $OH-(C)$ and $NH_2-(C)$
	47.17	$^{J}/_{mol*K}$	Benson	(A.1-19)	GC for $OH-(O)$ and $NH_2-(N)$
	117.94	$^{J}/_{mol*K}$	Ruzicka-Domalski	(A.1-20b)	GC for $OH-(C)$ and $NH_2-(C)$
	101.25	$^{J}/_{mol*K}$	Ruzicka-Domalski	(A.1-20b)	GC for $OH-(O)$ and $NH_2-(N)$
$c_p^{\ominus,(s)}$	54.84	$^{J}/_{mol*K}$	Modified Kopp's rule	(A.1-21)	GC for H, O, N
ρ	848.6	$^{kg}/_{m^3}$	Rackett	(A.1-23)	at 100 °C
$\Delta H_f^{\ominus,(l)}$	-191.17	$^{kJ}/_{mol}$	Domalski-Hearing	(A.1-27a)	GC for N(2H,C) and O(H,C)
$\Delta H_f^{\ominus,(s)}$	-205.96	$^{kJ}/_{mol}$	Domalski-Hearing	(A.1-27a)	GC for N(2H,N) and O(H,C))
$S^{\ominus,(l)}$	115.6	$^{J}/_{mol*K}$	Domalski-Hearing	(A.1-27b)	GC for N(2H,C) and O(H,C)
$S^{\ominus,(s)}$	67.62	$^{J}/_{mol*K}$	Domalski-Hearing	(A.1-27b)	GC for N(2H,N) and O(H,C)
ΔH_v	40.5	$^{kJ}/_{mol}$	Corresponding state	(A.1-35)	
	48	$^{kJ}/_{mol}$	Vetere	(A.1-36)	
ΔH_{fus}	9.11	$^{kJ}/_{mol}$	Chickos	(A.1-37a)	

2.2 Hydrogen peroxide

2.2.1 General

Hydrogen peroxide is a colorless liquid miscible with water in all proportions. With a boiling point of 150 °C it can theoretically be separated from water by distillation. Practically, the enrichment is done by crystallization, as the peroxides melting point is -0.43 °C. Thereby a 100 % hydrogen peroxide is obtained, which is stable at room temperature. In aqueous solution it shows weak acidic activity ($pK_A = 11.75$) and can therefore form salts with various metals[**goor_hydrogen_2019**].

2.2.2 Synthesis

Historically, hydrogen peroxide was first synthesized by wet chemical processes from barium peroxide and nitric acid. With hydrochloric acid the process has been improved.

$$BaO_2 + 2\,HCl \longrightarrow BaCl_2 + H_2O_2 \tag{2-6}$$

In 1853, *Meidinger* found hydrogen peroxide during the electrolysis of aqueous sulfuric acid. Later it was found that it can be formed via the peroxodisulfuric acid. Since its discovery in 1901, hydrogen peroxide is industrially synthesized by organic autooxidation processes. Hydroquinones react quantitatively with oxygen to quinones and hydrogen peroxide. In the following decades the process was optimized by varying the hydroquinones and altering the reaction conditions. The most successful proceeding utilizing alkylated anthraquinones (see figure 5) was developed of *BASF* and carried out industrially since in 1953. Up to now the direct combination of hydrogen and oxygen proved to be fruitless on a large industrial scale[**goor_hydrogen_2019**].

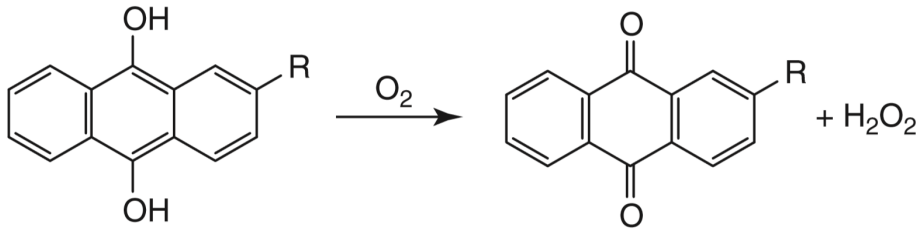


Figure 5: Industrial scale hydrogen peroxide synthesis process using 2-alkyl-9,10-anthraquinone[**goor_hydrogen_2019**]

2.2.3 Decomposition

An extremely important factor for the handling and storage of hydrogen peroxide is its disproportionation according to the decomposition reaction equations 2-7.

$$H_2O_2{}^{(g)} \longrightarrow H_2O^{(g)} + \frac{1}{2}\,O_2{}^{(g)} \qquad \Delta_R H = -105.8\ {}^{kJ}/_{mol} \tag{2-7a}$$

$$H_2O_2{}^{(l)} \longrightarrow H_2O^{(l)} + \frac{1}{2}\,O_2{}^{(g)} \qquad \Delta_R H = -98.3\ {}^{kJ}/_{mol} \tag{2-7b}$$

The reactions are highly exothermic and occur in presence of small amounts of catalysts, e.g. dissolved iron, copper, manganese, and chromium or heterogeneously by oxides and hydroxides of the same metals. Also all noble metals (platinum, osmium and silver) act as catalysts. Pure solutions without catalysts only undergo gas phase decomposition at high temperature. To prevent the decomposition with catalysts present, stabilizers like acid sodium pyrophosphate and sodium stannate are added in combination or separately[**goor_hydrogen_2019**].

The mechanism of the base-catalyzed decomposition reaction can be formulated as

$$H_2O_2 + OH^- \longrightarrow HOO^- + H_2O \tag{2-8a}$$

$$HOO^- + H_2O_2 \longrightarrow H_2O + O_2 + OH^- \tag{2-8b}$$

Since the first reaction step is characterized by the pK_A value in a aqueous solution, the decomposition is rather slow[**haas_kinetics_1960**]. It can be seen that the reaction mechanism shows the typical autocatalytic structure.

A kinetic model that accounts for contaminants like iron ions of sodium hydroxide was provided by [**cefic_sector_group_hydrogen_2012**]. The usually first order kinetics is expanded by a contamination factor k_F (equals 1 for uncontaminated product). Table 4 shows the influence of iron on this factor. An increasing amount of a contaminant increases k_F.

$$\frac{d[H_2O_2]}{dt} = k[H_2O_2] \tag{2-9a}$$

$$k = 0.81 \times 10^7 \times k_F \times e^{\frac{10357}{T}} min^{-1} \tag{2-9b}$$

Table 4: Influence of iron ion content on hydrogen peroxide decomposition kinetics[**cefic_sector_group_hydrogen_2012**]

Iron content / *ppm*	$k \times 10^8/min^{-1}$	k_F	**Comments**
0.12	0.95	1.46	Standard technical grade 70 % HP
0.37	3	4.6	
7.76	7.56	11.6	
15.52	9.42	14.4	
31.04	64.55	98.9	
38.81	912.5	1398	
46.6	6763	10366	
10	4476	6861	Unstabilized 70 % HP
20	21760	33360	

3 Experimental

3.1 Chemicals

Table 5: Chemicals

Name	CAS-nr.	Manufacturer	LOT#	MW $[g/mol]$	Purity
Ammmonium hydroxide solution	1336-21-6	Sigma-Aldrich	STBG8814	35.05	25 % (NH_3- basis)
Hydroxylamine (free base)	7803-49-8	ABCR	1410742	33.03	50% aq.
Hydroxylamine Freie Base	7803-49-8	BASF	0100292940	33.03	50% aq. with stabilizer
Hydrochloric acid 0,1 M	7647-01-0	Merck	1.09060.1000	36.46	
0.1 $^{mol}/_{L}$ Titanium siliclite-1	13463-67-7(TiO_2) 7621-86-9(SiO_2)	ACS Material	MSTS1001	-	-
Hydrogen peroxide	7722-84-1	Sigma-Aldrich	BCCC8328	34.01	30% for trace analysis

3.1.1 Titanium silicalite-1

Description:	Mesoporous Ti-Si Molecular Sieve TS-1
Manufacturer:	ACS Material
Manufacturing process:	Hydrothermal
Surface area:	360-420 $^{m^2}/_{g}$
Pore size:	0.56-0.58 nm
Comperative crystallinity:	>95 %
Na_2O content:	<0.1 wt%
Molar ratio Si/Ti:	~ 30
Cation:	H^+

3.1.2 Unstabilized hydroxylamine solution

When the unstabilized 50 % hydroxylamine solution was delivered, the bottle was bloated. Before using it, it was acidimetrically determined with 0.1 M hydrochloric acid (Exp5). A hydroxylamine content of 49.77 % and a ammonia content of 0.395 % were found (report see appendix A.42).

3.2 Equipment

3.2.1 Devices

- *Premex* explosion reactor MED 1789
- *Eurotherm 6000 series* writer/recorder
- *Bronkhorst In-Flow* flow sensor/controller
- *Vögtlin Red-y* flow meter
- *Metrohm Titrando* 809
- DSC *Mettler* DSC820
- *Mettler* high pressure crucibles gold 40 μL ME-26731

3.2.2 Software

- *MATLAB* R2019a
- *LabVIEW* 2019 Version 19.0.1f1 (64-bit)
- *RI-CAD* Ausbildungsversion *HiTec Zang GmbH* 2.2.0 (32-bit)
- *Metrohm tiamo* 1.1 - 36
- *Advanced Kinetics and Technology Solution (AKTS)* Advanced Thermokinetics Software v5.02 DB v1.5
- *STARe* software

3.2.3 *Bronkhorst* flow meter/controller

To measure and control the gas streams of the explosion reactor, a *Bronkhorst* flow meter/controller was used. A *LabVIEW* program was written to control the device and write the data continuously into a text file. The program is shown in appendix A.37.

3.3 Differential scanning calorimetry

3.3.1 Working principle

The working principle of the DSC is described in appendix A.1.4.

3.3.2 Measurements

Crucibles In this work five types of crucibles were used. C1 and C5 are the only crucibles made of steel, all the others are made of gold. C1 an C2 are 30 μL reusable crucible with screw caps. C3, C4 and C5 are 40 μL high pressure crucible that are closed by pressing. For all crucible gold rupture disks were used. The four most used types (C1 to C4) are shown in figures 6 to 9.

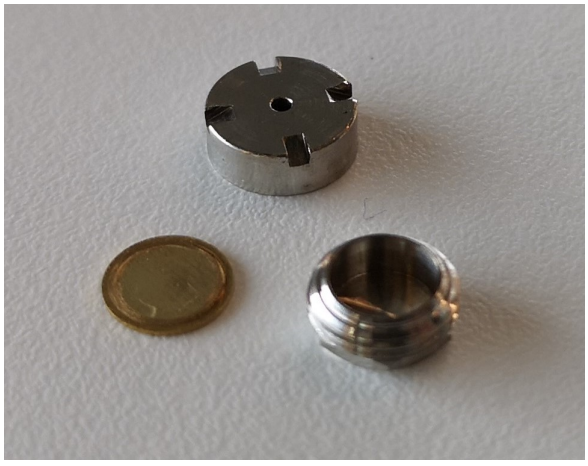


Figure 6: C1 - 30 μL reusable steel crucible

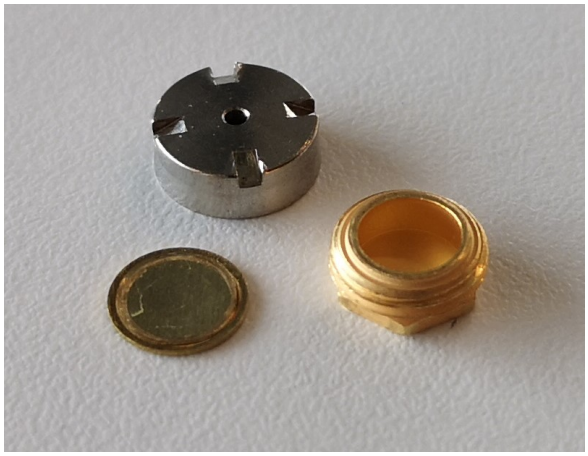


Figure 7: C2 - 30 μL reusable gold crucible

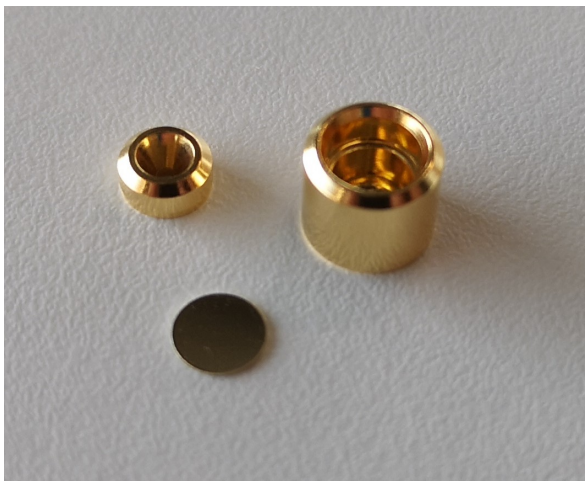


Figure 8: C3 - 40 μL high pressure gold crucible with smooth surface

Figure 9: C4 - 40 μL high pressure gold crucible with rough surface

General procedure Apart from the indicated exceptions, 9.8 uL were pipetted into the crucible defined above. To close the crucible either the provided press or, for the crucibles with screw cap, a dynamometric screwdriver (3.7 Nm) was used. The most employed methods were

- Dynamic 4 K/min from 30 to 290 °C
- Isothermal 90 °C for 24 h
- Isothermal 110 °C for 12 h
- Isothermal 130 °C for 6 h
- Isothermal 150 °C for 5 h

After each isothermal experiment a dynamic measurement of the same sample was appended to check the residual energy content. The dynamic curves were evaluated with linear baselines from visual judgment. For the isothermal curves, horizontal baselines from the last measurement point were chosen, because this resulted in the calculation of energy contents similar to those recovered in the dynamic measurements. A experiment plan of all the DSC measurements done at FHNW can be found in table 6. Additionally, DSC measurements of the stabilized hydroxylamine solution and different mixtures was conducted by the the project partner (see table 7).

Coating experiments To inhibit a potential surface catalysis of the gold crucibles, one was coated with silicone spray (SCC3 Conformal Coating *DCA*) and one with PTFE spray (*weicon*). They were let to dry over night. While the silicone coating looked uniformly distributed, the PTFE coating appeared to have numerous gaps. The crucible only had slightly white stains. To prevent the coating from sticking to the DSC sensor, the coating was polished on the bottom side using emery paper. 50 % hydroxylamine solution was measured dynamically in the two coated crucibles.

Table 6: Overview of all DSC measurements done at FHNW

Exp	Name	Methode	Crucible	Tare_start / mg	Netto_start / mg	Brutto_end / mg	Netto_end / mg	Δm / mg	q / J/g	q_rest / J/g	q_tot / J/g
Exp2	HA_dyn_4	Dyn4	C1	1436.04	11.19	1447.28	11.24	0.05	1438.8		1438.8
Exp3	HA_iso_150	Iso150	C1	1433.4	11.99	1445.4	12	0.01	1689	0	1689
	HA_iso_130	Iso130	C1	1430.4	11.52	-	-	-	2875	0	2875
	HA_iso_110	Iso110	C1	1438.1	12.16	-	-	-	2181	0	2181
	HA_iso_90	Iso90	C1	1432.54	11.28	-	-	-	1623	220	1843
	HA_iso_90_24h_2	Iso90	C1	853.4	15.11	868.44	15.04	-0.07	2685.1		2685.1
	HA_iso_90_24h_3	Iso90	C1	853.32	13.26	866.6	13.28	0.02	2459.3		2459.3
Exp4	HA_dyn_4_12mg	Dyn4	C1	857.08	11.82	869.05	11.97	0.15	2342		2342
	HA_dyn_4_7mg	Dyn4	C1	854.69	6.84	861.26	6.57	-0.27	2399		2399
Exp6	HP_dyn_4_Au	Dyn4	C2	1476.1	12.47	-			641.32		641.32
	HP_dyn_4_Steel	Dyn4	C1	1431.95	11.2	-			1026.4		1026.4
	HP_iso_40_12h_Steel	Iso40	C2	876.26	11.88	-			0	600	600
Exp7	HA_iso_40_12h	Iso40	C1	856.81	12.58	869.32	12.51	-0.07	126.87	2007.9	2134.8
Exp8	HA_iso_90_24h_2	Iso90	C2	880.64	13.81	894.45	13.81	0	1798	227	2025
	HA_iso_110_12h	Iso110	C2	883.49	13.62	897.1	13.61	-0.01	1777	154	1931
	HA_iso_150_5h	Iso150	C2	882.9	12.97	895.83	12.93	-0.04	2545	0	2545
	HA_iso_130_6h	Iso130	C2	883.42	12.93	896.31	12.89	-0.04	2109	20	2129
	HA_dyn_4	Dyn4	C2	882.86	8.07	890.89	8.03	-0.04	2457		2457
Exp9	HA_dyn_4_afterdecomp	Dyn4	C2	880.67	9.33	889.93	9.26	-0.07	112.03		112.03
Exp10	HA_dyn_4_afterdecomp	Dyn4	C2	882.8	8.07	890.98	8.18	0.11	66.52		66.52
Exp12	HA_dyn_4_afterdecomp	Dyn4	C2	1479.42	8.57	1487.97	8.55	-0.02	259.46		259.46
Exp14	HA_dyn_4_Sil	Dyn4	C3	1489.63	10.62	1498.26	8.63	-1.99	3137	0	3137
	HA_dyn_4_PTFE	Dyn4	C3	1480.58	11.37	1491.76	11.18	-0.19	2246.6	0	2246.6
	HA_dyn_4_Mettler	Dyn4	C3	1470.57	11.1	1493.19	22.62	11.5	2436.8	0	2436.8
	HA_dyn_4_Baechler	Dyn4	C4	1473.89	11.36	1485.13	11.24	-0.12	2420.9	0	2420.9
	HA_iso_110_12h	Iso110	C3	1476.09	11.74	1487.84	11.75	0.01	1927.4	292.02	2219.4
	HA_iso_90_24h	Iso90	C3	1479.79	11.63	1491.4	11.61	-0.02	1265.7	516.69	1782.4
	HA_iso_150_5h	Iso150	C3	1475.42	11.38	1486.76	11.34	-0.04	2433.7	90.4	2524.1
	HA_iso_130_6h	Iso130	C3	1482.19	10.5	1492.71	10.52	0.02	2205	61.33	2266.4
	HA_iso_150_5h_stored	Iso150	C3	1474.27	11.19	1485.74	11.47	0.28	2195.2	62.11	2257.3
	HA_dyn_4_Mettler2	Dyn4	C3	1484.32	10.46	1494.78	10.46	0	2681.9	0	2681.9
Exp17	HA_dyn_4_1	Dyn4	C3	1478.27	11.2	1489.43	11.16	-0.04	1971		1971
	HA_dyn_4_2	Dyn4	C3	1476.03	11.46	1487.5	11.47	0.01	1850.6		1850.6
	HA_dyn_4_3	Dyn4	C3	1480.2	11.19	1488.87	8.67	-2.52	1747.5		1747.5

Table 7: Overview of all DSC measurements done by Process Safety Lab of the project partner

Exp	Crucible material	Method	HA 50%	H2O2 50%	NH3 25%	TS-1
Exp16_1	Au	Iso90	12 mg	-	-	-
Exp16_2	Au	Iso110	12 mg	-	-	-
Exp16_3	Au	Iso130	12 mg	-	-	-
Exp16_4	Au	Iso150	12 mg	-	-	-
Exp13_1.1	Au	Dyn4	6 mg	-	6 mg	-
Exp13_1.2	Au	Dyn4	6 mg	-	4 mg	-
Exp13_1.3	Au	Dyn4	6 mg	-	2 mg	-
Exp13_2	Au	Dyn4	6 mg	6 mg	-	-
Exp13_3	Au	Dyn4	6 mg	-	-	6 mg
Exp13_4.1	Au	Dyn4	-	6 mg	6 mg	-
Exp13_4.2	Au	Dyn4	-	6 mg	4 mg	-
Exp13_4.3	Au	Dyn4	-	6 mg	2 mg	-
Exp13_5	Au	Dyn4	-	-	6 mg	6 mg

3.4 Explosion reactor

To verify the proposed reaction hypotheses and thermal properties of hydroxylamine, the decomposition reaction can be investigated in the explosion reactor (EXR), which holds up to 300 *bar*. The reaction mixture is prepared in the EXR. The temperature can be regulated and measured. The pressure is measured as well. When the chemical equilibrium is reached, the liquid phase can be removed by means of the immersion tube. This way neither the chemical nor the phase equilibrium can readjust. The removed liquid phase (aqueous solution) can be analyzed by acidimetric titration.

3.4.1 Apparatus

After the first experiment (Exp9), the reactor setup was extended with a pressure reduction (V9) to be able to control the gas stream when collecting the gas in the cooling trap. A scheme of the apparatus used for all the experiment apart from the first is shown in figure 10.

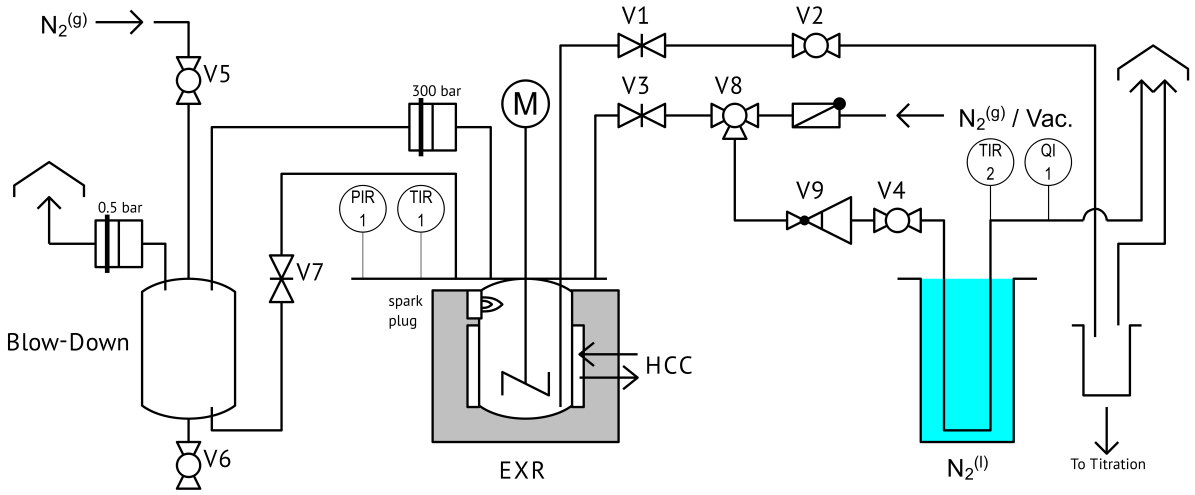


Figure 10: Scheme of the setup for the explosion reactor

The jacket of the *Premex* explosion reactor (EXR) MED 1789 is made of hastelloy C-22 (HC22) and either a borosilicate or PEEK insert can be used. The immersion tube and the thermometer protection sheath are made of stainless steel. A stirrer made of HC22, stainless steel or PEEK (see figure 16) can be used. The maximum permissible pressure is 300 *bar*. A bursting disc opens the reactor between 270 and 330 *bar*. The bursting disc of the blow down tank opens at 0.5 *bar* overpressure.

Figure 11: Picture of the whole measuring apparatus

Figure 12: Picture of the apparatus part used for the liquid collection and gas analytics

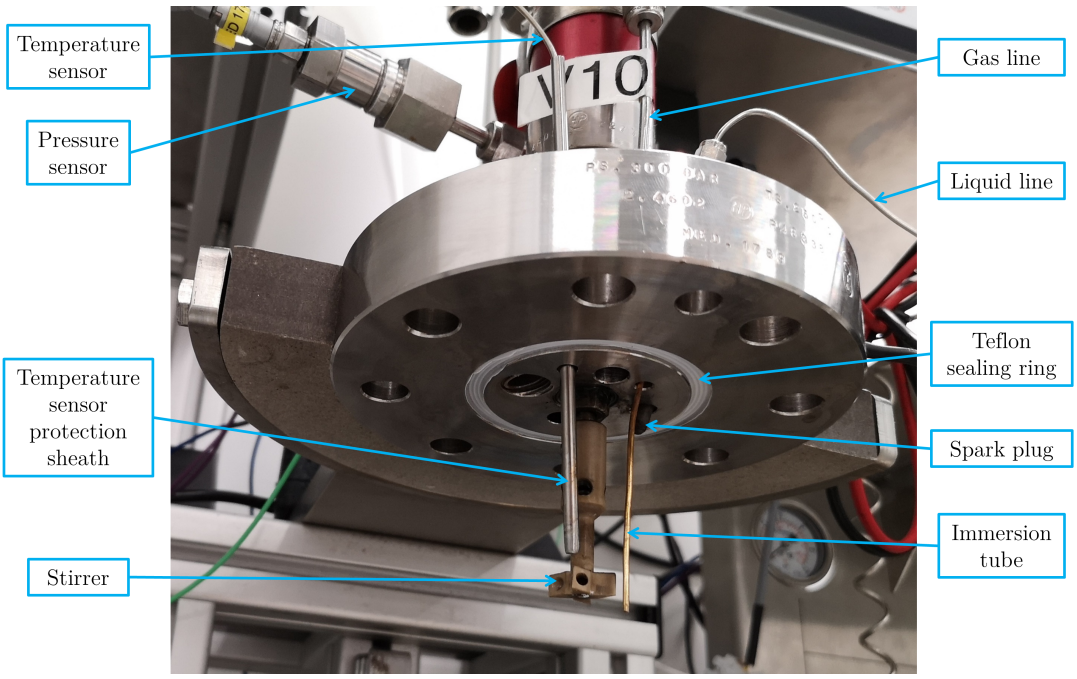


Figure 13: Picture of the explosion reactor lid

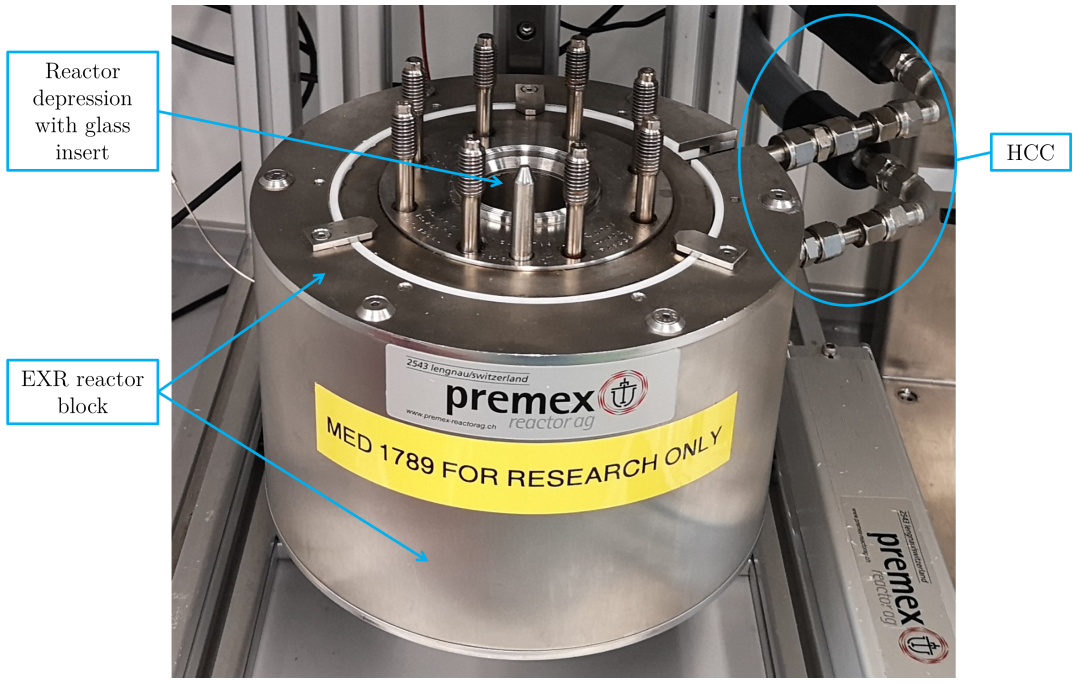


Figure 14: Picture of the explosion reactor block

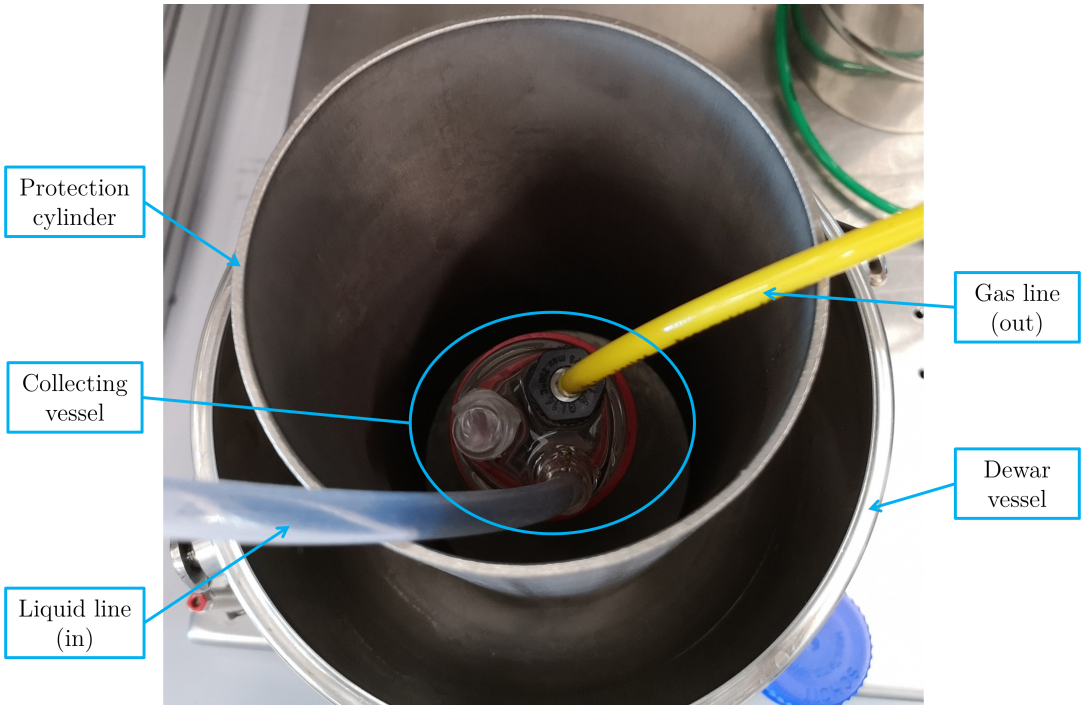


Figure 15: Picture of the apparatus part used for the liquid collection

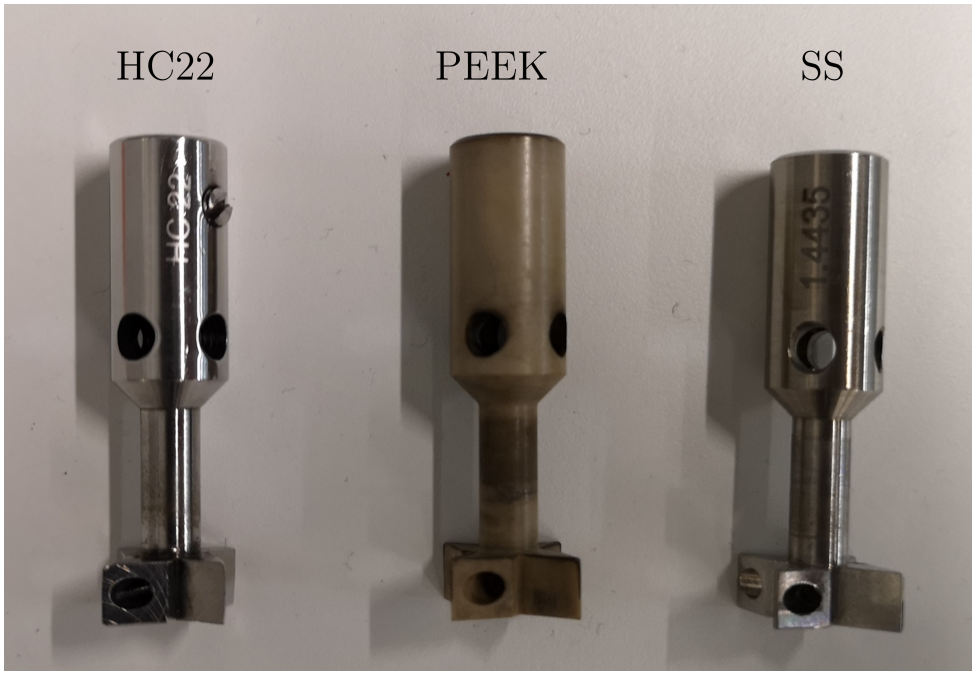


Figure 16: Picture of the stirrers for the explosion reactor made from different materials

Gilding steel parts To ensure the possibility to measure the decomposition with steel and without, the reactor parts that are not exchangeable have to be steel-free. The reactor

jacket is made of HC22. The inserts and the stirrer are exchangeable. Problematic are the protection sheath of the temperature probe and the immersion tube. They are both made of stainless steel (316L). To prevent an effect of these parts they were undertaken a galvanic gold deposition process. Two electrolytic baths were used (see figure 17). The first bath contained a 80 g/L aqueous solution of electrolytic degreasing salt type A. As anodes for this bath, two steel alloy grids were used. The second bath was a acidic pregold electrolyte bath. It contained a solution of potassium gold(I) cyanide in sulfuric acid. As anode platinized titanium electrodes were used. The cathodes were the steel tubes. Positive gold(I) ions in a $Au(CN)_2^-$ complex are attracted by the cathode and travel to a region near its surface, also known as the *Helmholtz* double layer. Within this region the complex receives an electron and breaks up, according to equation (3-1a). The gold atom deposits itself on the metal and the ligand ions are freed to the aqueous solution. The anode reaction (3-1b) releases the required electron through the electrolysis of water molecules[**wilkinson_understanding_1986**].
The cathode reaction can be formulated as follows:

$$\mathrm{AuCN} + \mathrm{e}^- \longrightarrow \mathrm{Au} + \mathrm{CN}^- \tag{3-1a}$$

The electrolysis of water serves as anode reaction.

$$2\,\mathrm{H_2O} \longrightarrow 4\,\mathrm{H}^+ + \mathrm{O_2} + \mathrm{e}^- \tag{3-1b}$$

Figure 17: Apparatus for the gilding process. On the left is the degreasing bath and on the right the pregold electrolyte bath with the protection sheath immersed. The voltage applied can be controlled with the knob on the top left of the apparatus.

Both tubes were first cleaned in isopropanol and acetone in an ultra sonic bath for 3 minutes each. Afterwards the tubes were connected to the cathode and immersed into the first bath. The tubes were cleansed without movement for 30 seconds with 0.2 A and 2.5 V. The tubes were then flushed with distilled water and then rinsed with water from the pipe. To pickle the steel, the tubes were immersed into a 10 % sulfuric acid solution for 10 seconds. Gas evolution was observed at both electrodes. The parts were again rinsed with water from the tube and flushed in distilled water. For the gilding process, the steel parts were connected to the cathode rod and immersed in the pregold electrolyte bath with slow movement perpendicular to the electrodes. To yield the best results, a current density of $j = 4 - 10\ {}^{A}/_{dm^2}$ is required. With $0.2A/4V$ for the immersion tube ($A = 0.01\ dm^2$) and $1A/5V$ for the protection sheath ($A = 0.10\ dm^2$), the gold was deposited during 10 minutes. Gas evolution was observed at both electrodes again. To produce a gold layer of 1 μm, the process has to be maintained for 1 to 3 minutes. Consequently, a layer of about 3 μm thickness is to be expected after 10 minutes. The result is shown in figure 18.

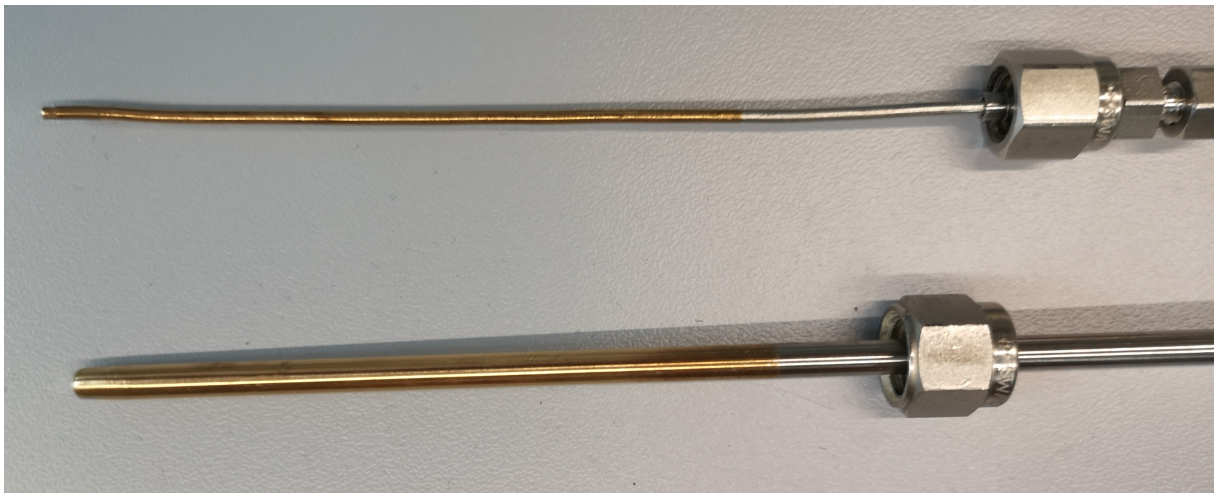


Figure 18: Gilded immersion tube (top) and protection sheath (bottom)

3.4.2 Risk analysis

For all the critical steps with the EXR, a risk analysis was conducted (see appendix A.43). It was found that the reactor can be safely operated with 80 mL filling volume and up to 80 °C process temperature.

3.4.3 Measurements

General procedure The workstation was prepared before every experiment according to the following checklist:

1. personal protective equipment is worn: laboratory coat and safety goggles, gloves only when handling liquids openly

2. emergency routes ensured
3. warning sign provided
4. collecting vessels provided
5. emergency cooling media such as ice and water are provided
6. disposal bucket with water for pre-dilution is provided
7. chapel
   (a) ventilation running
   (b) chapel window completely closable
   (c) thermostat
   (d) sufficient thermofluid
   (e) switched on
   (f) maximum temperature set at 110 °C
8. blowdown tank
   (a) inerted with nitrogen (caution: bursting disc only holds 1.5 *bar*)
   (b) valves 5, 6 and 7 closed
9. EXR
   (a) screws tightened with 90 Nm
   (b) tightness ensured (leakage rate < 10 $^{mbar}/_{h}$ at 5 *bar*)
   (c) correct bursting disc installed (300 *bar* bursting pressure)
   (d) inerted with nitrogen and valve 8 switched to the cooling trap path
   (e) maximum filled with 80 mL
   (f) valve 4 open
   (g) valves 1 to 3 and 7 closed

The glass insert was placed inside the reactor depression, the stirrer made of the desired material was attached to the motor and the teflon sealing ring was placed in the reactor lid. To close the reactor the screw-nuts were sequentially tightened with 30, 60 and 90 Nm. To check the tightness of the system, the reactor was pressed up with nitrogen to approximately 4 *bar*. The thermostat was set to 128 °C apart from Exp9 and Exp11 (PEEK stirrer and gold tube). The stirrer was set to 500 rpm. The leakage rate was calculated in a phase of thermal equilibrium over night. The reactor was opened over the immersion tube and evacuated. The outer end of the immersion tube (teflon hose connected to pipette tip) was used to transfer the sample solution into the reactor. The

vacuum pump was disconnected instantly after the solution was transferred into the reactor. When the reactor reached ambient pressure (ca. 1.1 *bar*), the valves 1, 2, 3 and 7 were closed.

After the reaction mixture was heated up to the process temperature the pressure rose with different speeds. In most cases the temperature remained practically stable between 110 and 115 °C. The end of the decomposition reaction was indicated by the stabilization of the reactor pressure which happened after approximately 10 to 20 hours. The thermostat was then switched off and the reactor was let to cool down for another 7 to 10 hours resulting in a pressure drop.

The liquid phase was discharged via the immersion tube as it was pushed out by the remaining reactor pressure. The valve was immediately closed when gas was coming out. The released solutions were all analyzed for hydroxylamine and ammonia by acidimetric titration (reports see appendix A.42). Some solutions were also measured in the DSC (plots see appendix A.35).

An overview of all the experiments conducted in the EXR can be found in table 8. In section 4.1.1 every experiment is shortly described.

3.5 Miniautoclave

To backup the results obtained from the DSC and EXR, miniautoclave measurements (see appendix A.40) of the project partners Process Safety lab were evaluated.

The miniautoclave setup consists of three steel cells placed in a ceramic oven. The temperature and pressure of each cell is measured. The inside is equipped with a glass insert and thermometer protection sheath. There is one empty reference cell and two cells for a double determination.

Table 8: Experiment plan of all the experiments conducted in the EXR

Experiment	Purpose	Insert material	Stirrer material	Surface immersion tube material	Leakage rate / mbar/h	Sample weight / g	Sample volume / mL	Process temperature	w(HA) / %	w(NH3) / %	α end	Maximal pressure / bar	Maximal temperature
Exp9	Experiment with increasing temperature (25 to 100 °C)	Glass	PEEK	Gold	11	22.4	20.0	25 - 100 °C	2.646	8.8675	0.9471	27.72	99.1 °C
Exp10	First isoperibolic experiment (at 110 °C)	Glass	PEEK	Gold	<0.1	32.1	28.6	110 °C	2.1615	8.393	0.9568	46.18	116.13 °C
Exp11	Verification of ammonia phase equilibrium	Glass	PEEK	Gold	<0.1	25.2	28.0	25 - 100 °C	-	-	-	9.34	115.82 °C
Exp12	Experiment with steel stirrer	Glass	Steel	Gold	<0.1	32.6	29.1	110 °C	3.547		0.9291	58.85	221.69 °C
Exp15	Reproduction of Exp10	Glass	PEEK	Silicon	3	32.5	29.0	110 °C	1.2765		0.9745	47.52	118.07 °C
Exp16	Experiment with HC22 stirrer	Glass	HC22	Steel	<0.1	32.6	29.1	110 °C	2.0485	9.853	0.959	47.73	120.96 °C
Exp18	Determination of reactor volume	Glass	PEEK	Steel	-	0.0	0.0	25 °C	-	-	-	4.48	24.88 °C
Exp19	Reference experiment to Exp20	Glass	PEEK	Steel	<0.1	31.1	27.8	110 °C	2.84		0.9432	43.46	118.29 °C
Exp20	Experiment with catalyst TS-1	Glass	PEEK	Steel	<0.1	32.5	29.0	110 °C	0.833	8.53	0.9833	50.35	114.39 °C

4 Results

4.1 Explosion reactor

To model the phases and the reaction equilibrium in the EXR the equations from appendix A.2 were used. In the following the situation specific parameters are mentioned. In table 9 the substances involved are listed. Figure 19 shows a scheme of the phase and the reaction equilibria that were considered in the model.

Table 9: Substances present in the explosion reactor

i	**Formula**	**Name**
1	NH_2OH	hydroxylamine
2	NH_3	ammonia
3	N_2	nitrogen
4	N_2O	nitrous oxide
5	NO	nitric oxide
6	H_2O	water
7	H_2	hydrogen

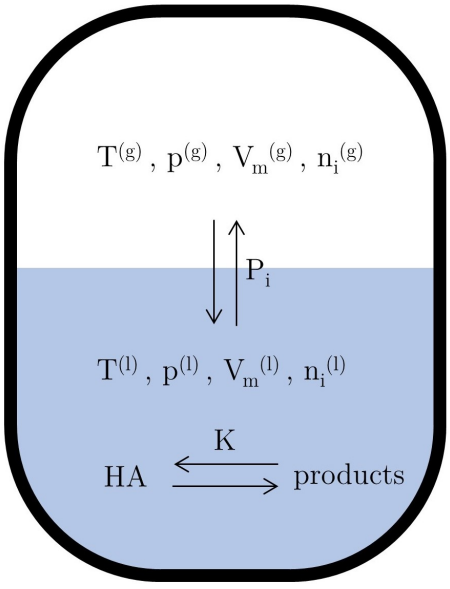


Figure 19: Scheme of the physical properties and relevant equilibria in the EXR

The stoichiometric coefficients $\nu_{i,r}$ for the proposed reactions r in table 10 are derived from the reaction equations.

Table 10: Proposed reaction hypotheses

Abbr.	r	**Eq.**	**Description**
R1	1	(2-5)	empirical reaction
R2	2	(2-3)	alkaline reaction
R3	3	(2-4)	acid reaction

$$\nu_{i,r} = \begin{bmatrix} -157 & 61 & 35 & 12 & 2 & 143 & 1 \\ -3 & 1 & 1 & 0 & 0 & 3 & 0 \\ -4 & 2 & 0 & 1 & 0 & 3 & 0 \end{bmatrix} \tag{4-1}$$

Results In order to find the thermodynamic equilibrium, the equilibrium constant K_V (see figures 20, 21 and 22) and the equilibrium term $\sum_{i=1}^{N_C} \nu_{i,r}\mu_i$ (see figures 23, 24 and 25) were calculated as a functions of the temperature and the reaction conversion α for the three reaction hypotheses. The equilibrium is found where K_V equals 1 and the equilibrium term (A.2-34) equals 0. Additionally, the total pressure was modeled (see figures 26, 27 and 28). The phase compositions and partial pressure of every component can be calculated (for the code see appendix A.18).

The same calculations were also done, by considering an initial mixture of 15 mL 50 % hydroxylamine solution and 15 mL 25 % ammonia solution (see appendix A.12).

Equilibrium constant K_V for the three reaction hypotheses:

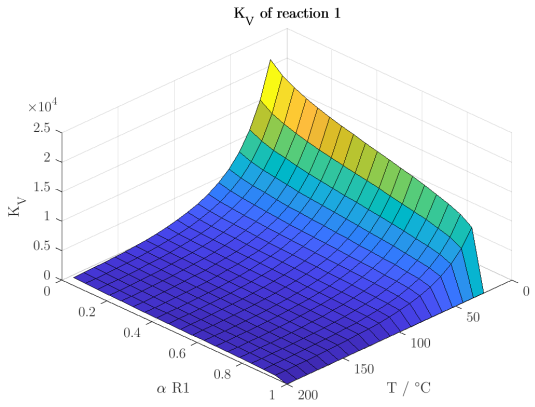


Figure 20: The equilibrium constant K_V as a function of temperature and the reaction conversion of the empiric reaction hypothesis (R1). The plot shows a strong temperature dependence of the reaction equilibrium (code see appendix A.18).

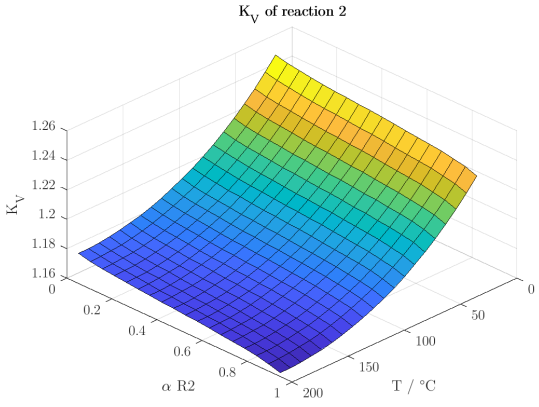


Figure 21: The equilibrium constant K_V as a function of temperature and the reaction conversion of the alkaline reaction hypothesis (R2). The alkaline reaction shows a weaker temperature dependence than the empiric reaction (code see appendix A.18).

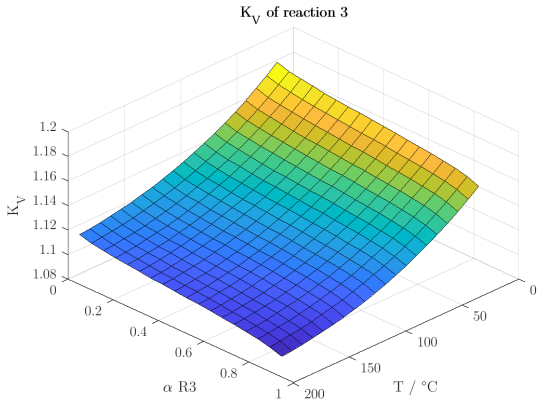


Figure 22: The equilibrium constant K_V as a function of temperature and the reaction conversion of the acid reaction hypothesis (R3). The acid reaction shows the weakest temperature dependence of all the hypotheses (code see appendix A.18).

Equilibrium term $\sum_i \nu_i \mu_i$ for the three reaction hypotheses:

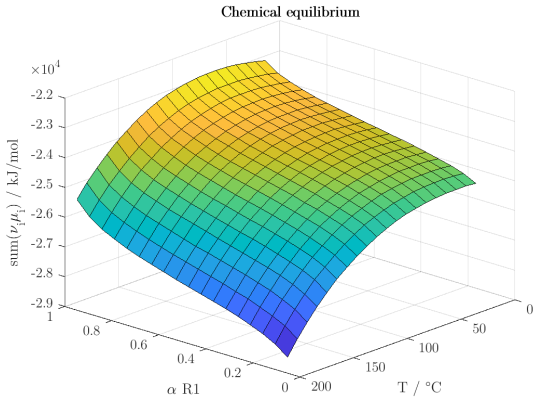


Figure 23: The equilibrium term (A.2-34) as a function of temperature and the reaction conversion of the empiric reaction hypothesis (R1). The calculated surface does not cross 0 on the z axis (equilibrium state). The term has the lowest values in the empiric hypothesis of all the hypotheses (code see appendix A.18).

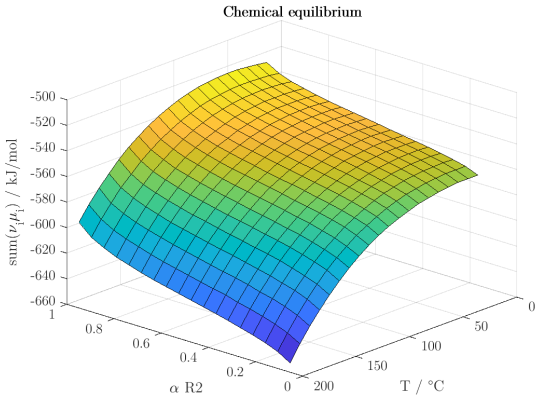


Figure 24: The equilibrium term (A.2-34) as a function of temperature and the reaction conversion of the alkaline reaction hypothesis (R2). The calculated surface does not cross 0 on the z axis (equilibrium state) (code see appendix A.18).

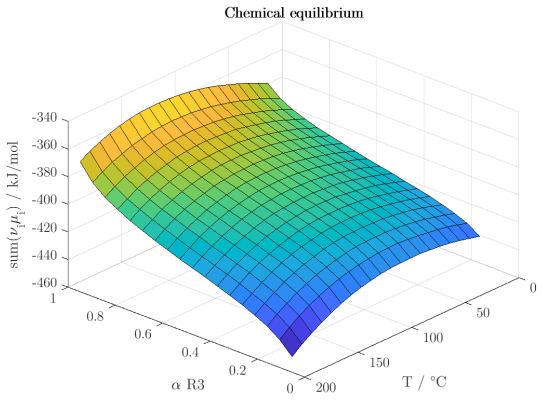


Figure 25: The equilibrium term (A.2-34) as a function of temperature and the reaction conversion of the acid reaction hypothesis (R3). The calculated surface does not cross 0 on the z axis (equilibrium state). The term has the highest values in the acid hypothesis of all the hypotheses (code see appendix A.18).

When calculating the equilibrium term in the range of $\alpha = 1 - 10^5$ to $\alpha = 1$ the surface gets steeper and the equilibrium state can be found. These results show that with the prevalent model an reaction conversion of 1 can be expected at all temperatures from 25 to 200 °C.

Total pressure for the three reaction hypotheses:

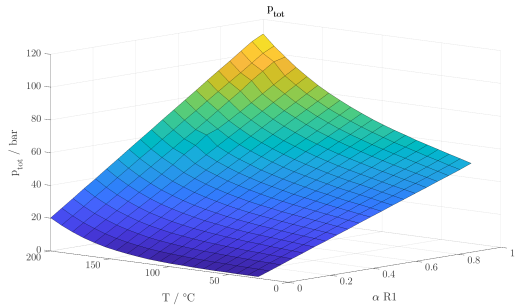


Figure 26: Total pressure as a function of temperature and the reaction conversion of the empiric reaction hypothesis (R1). The pressure rises with temperature and linearly with the reaction conversion (code see appendix A.18).

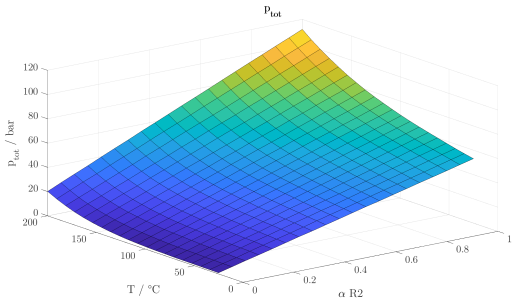


Figure 27: Total pressure as a function of temperature and the reaction conversion of the alkaline reaction hypothesis (R2). The pressure rises with temperature and linearly with the reaction conversion (code see appendix A.18).

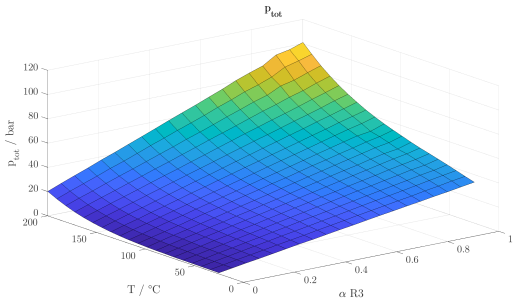


Figure 28: Total pressure as a function of temperature and the reaction conversion of the acid reaction hypothesis (R3). The pressure rises with temperature and linearly with the reaction conversion (code see appendix A.18).

4.1.1 Experiment curves

Exp18 - Determination of the reactor volume To accurately model the pressure changes in the EXR, its total volume has to be known. The apparatus documentation by *Premex* indicates a volume of 120 mL. The usage of the glass insert and tubes to the valves may change this volume. Therefore, the reactor was closed and pressed up to 4 *bar* with nitrogen. The volume of the gas flowing out again was measured with the *Bronkhorst* flow meter.

The volume of the reactor V was calculated by comparing the state before (state 1) and after (state 2) the emptying of the reactor according to equation

$$V = \int_{\text{state 1}}^{\text{state 2}} \dot{V} dt \; \frac{p_2 T_1}{p_1 T_2 - p_2 T_1} \tag{4-2}$$

The average result from the double determination was 111.05 mL (calculation see appendix A.15).

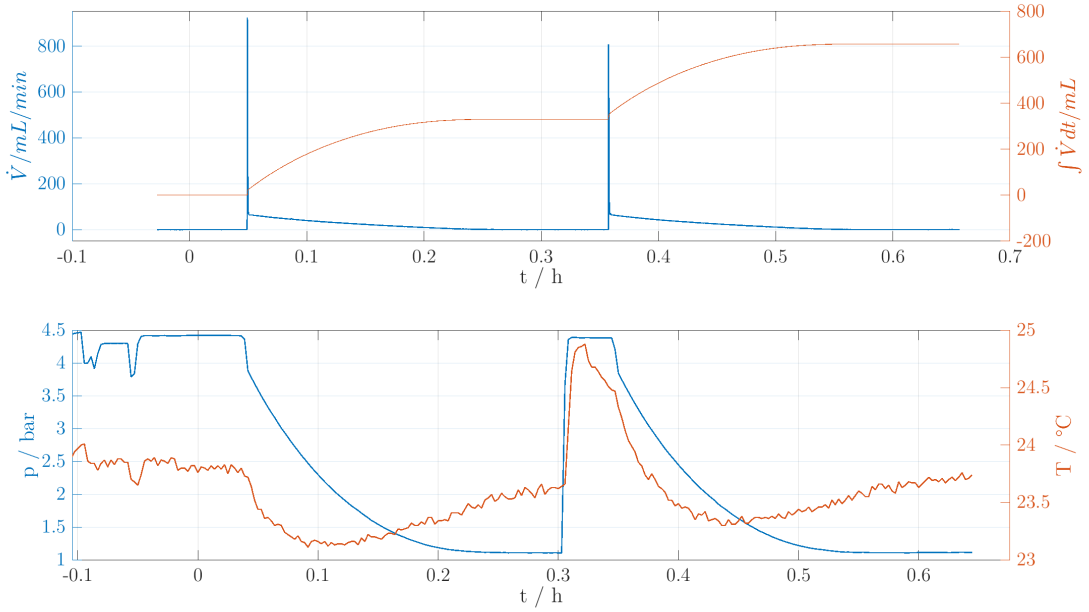


Figure 29: Double determination of the reactor volume. Upper plot: **Blue** Measured volume flow **Red** Integrated volume flow curve. Lower plot: **Blue** Measured pressure in the EXR. **Red** Measured temperature in the EXR (code see appendix A.15).

Exp11 - Verification of the *Henry* coefficient of ammonia To verify the *Henry* coefficient of ammonia found in the literature [**sander_compilation_2015**], an experiment in the EXR was conducted solely with an 25.5 g of an 25 % ammonia solution. The glass insert and the PEEK stirrer were used. After filling and closing, the reactor was heated up to 110 °C and then let cool down to room temperature. The gold coating of the thermometer protection sheath was affected by this measurement. Some steel spots became visible.

In figure 30 the experiment curve and the modeled curve of the pressure in the reactor are shown (calculations see appendix A.21 and A.30).

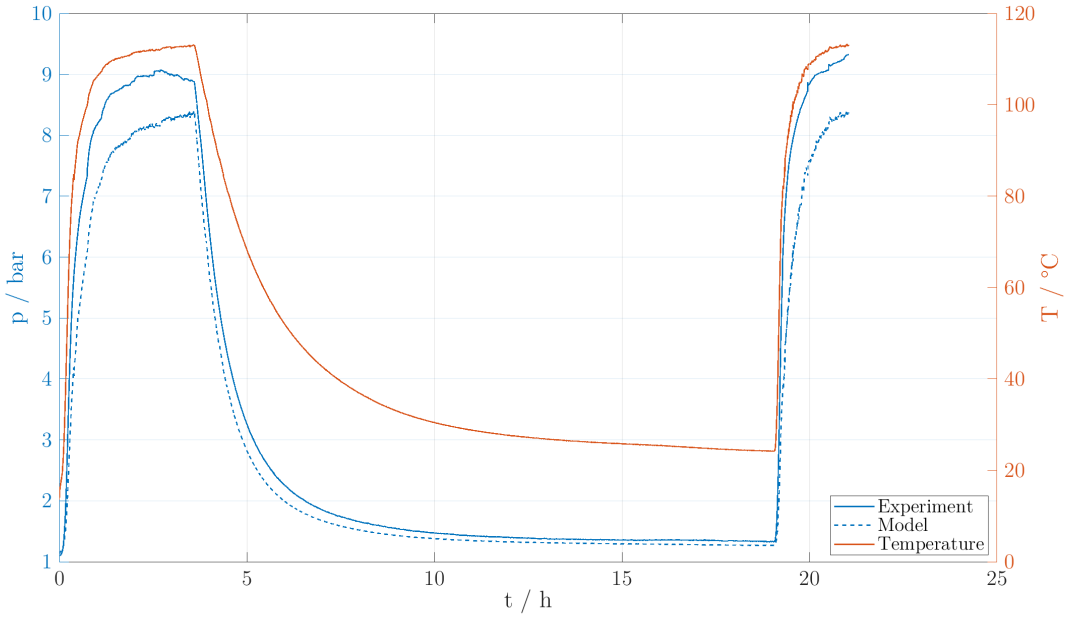


Figure 30: Exp11 - Ammonia phase equilibrium. **Blue curve** Measured pressure **Blue dotted** Pressure modeled from the temperature measurement with *Henry*'s law **Red** Measured temperature. The modeled curve fits the best during the phase of cooling down (4 - 19 h). This plot shows that the temperature-dependant *Henry* coefficient can be used to accurately model the partial pressure of ammonia in the explosion reactor.

The modeled curve fits the experimental data much better when the reactor is let to cool down and the temperature changes are slower. When heating up the, temperature probe reacts too slow, thus the simulated pressure (based on the temperature) reacts too slow as well.

Exp9 - First EXR experiment with the hydroxylamine solution The solution was prepared in the cold reactor. While heating up, the spark plug was used to try to ignite the gas mixture several times. No abrupt pressure rise was observed. Therefore, it can be assumed that no oxidizing gases are formed in the decomposition of hydroxylamine.

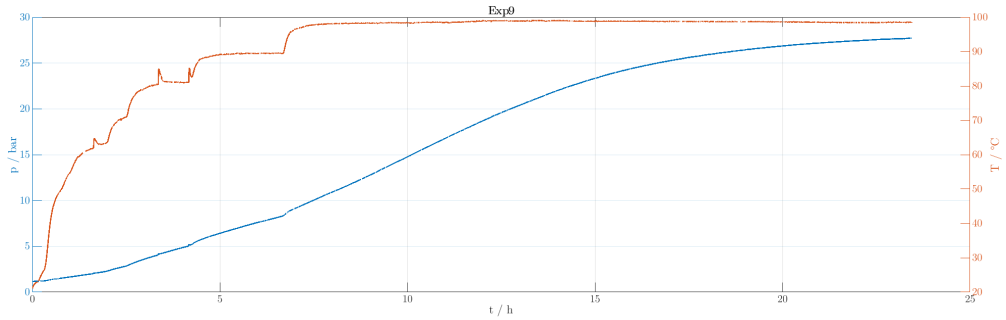


Figure 31: Experiment curves of experiment with PEEK stirrer and gilded tubes (Exp9). **Blue** Measured pressure in *bar*. **Red** Measured temperature in °C. The heated up stepwise. The gas phase was not ignitable at any time.

Exp10 - Isoperibolic experiment with PEEK stirrer The decomposition reaction was isoperibolically measured at 110 °C. A temperature peak simultaneous to a pressure change rate peak occurred after 5 h. After 9 h a jump of the temperature signal could be observed.

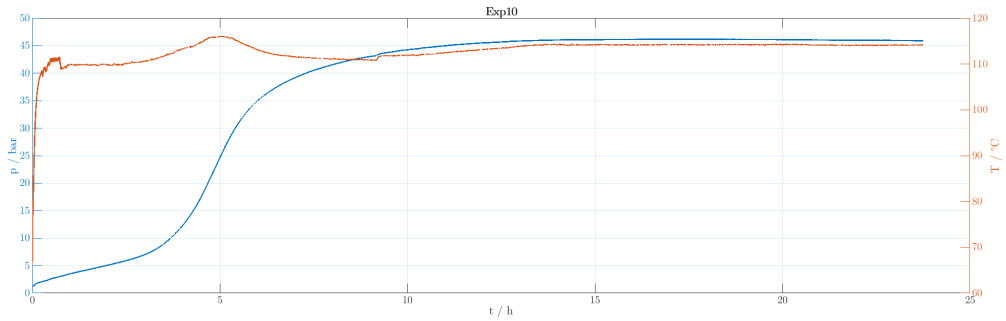


Figure 32: Experiment curves of experiment with PEEK stirrer and gilded tubes (Exp10). **Blue** Measured pressure in *bar*. **Red** Measured temperature in °C. The jacket temperature was held constantly at 110 °C. During the strongest pressure rise (5 h), the temperature in the reactor was the highest as well. The pressure stabilized after 15 hours.

Exp12 - Isoperibolic experiment with steel stirrer To investigate the catalytic effect of stainless steel on the decomposition reaction, the SS stirrer was employed. After 100 minutes, a sharp temperature and pressure rise occurred. With 10 s resolution, a peak of the pressure change rate of 2.64 bar/s and temperature change rate of 5.42 K/s were observed.

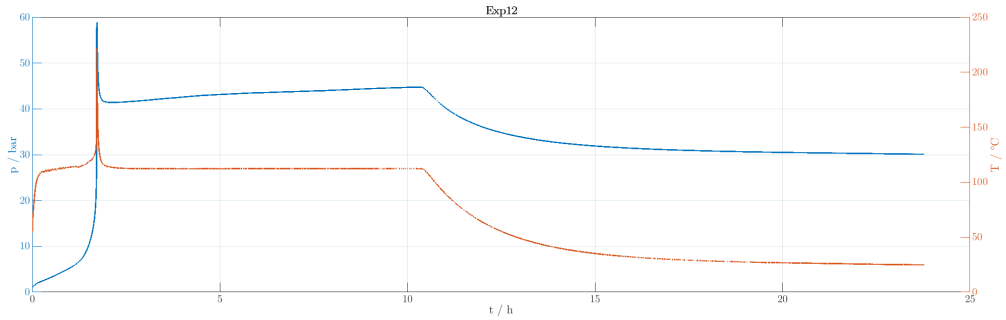


Figure 33: Experiment curves of experiment with steel stirrer and gilded tube (Exp12). **Blue** Measured pressure in *bar*. **Red** Measured temperature in °C. After 100 minutes a violent pressure and temperature rise occurred. When the mixture was cooled to 110 °C again, the decomposition proceeded. After 10 hours the thermostat was turned off and the mixture cooled down.

Exp15 - Isoperibolic experiment with PEEK stirrer After the experiment with the ammonia solution (Exp11) led to removal of the gold coating of the protection sheath, the isoperibolic experiment with the PEEK stirrer (Exp10) was reproduced, this time with a silicone coating of the protection sheath. Additionally, the reactor was let to cool down, after the pressure signal stabilized. This way, more information on the gas phase composition can be gathered, because the gases have different temperature-solubility behavior.

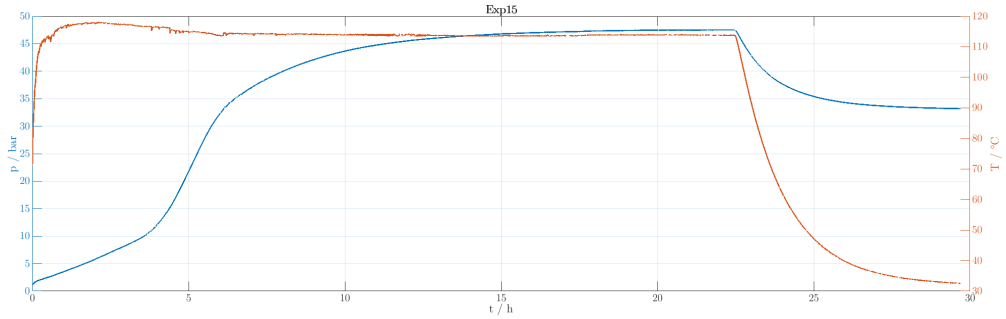


Figure 34: Experiment curves of experiment with PEEK stirrer and silicone coated tube (Exp15). **Blue** Measured pressure in *bar*. **Red** Measured temperature in °C. The reaction mixture showed the highest self-heating rate in the first phase of the experiment (1-3 hours), while the strongest pressure rise occurred only after 4 hours. After 23 hours the thermostat was turned off and the mixture cooled down.

Exp16 - Isoperibolic experiment with HC22 stirrer To test the decomposition reactor with another material, the experiment with steel stirrer (Exp15) was reproduced with a HC22 stirrer. The gas phase was analyzed after the experiment (see appendix A.11).

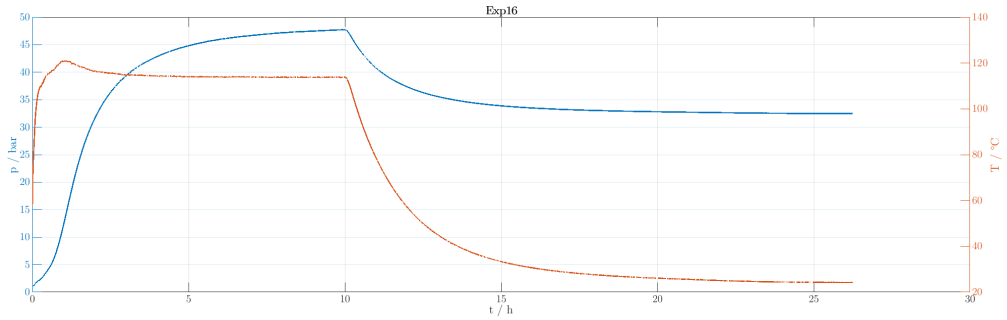


Figure 35: Experiment curves of experiment with HC22 stirrer and silicone coated tube (Exp16). **Blue** Measured pressure in *bar*. **Red** Measured temperature in °C. The fastest pressure rise coincided with the highest self-heating rate after 1 hour. After 10 hours the thermostat was turned off and the mixture cooled down.

Exp19 - Isoperibolic experiment with PEEK stirrer After the experiment with the HC22 stirrer (Exp16) it became apparent, that the silicone coating of the protection sheath disappeared. To set a new reference measurement for the following experiment, another reproduction of the first isoperibolic experiment with the steel stirrer (Exp10) was conducted.

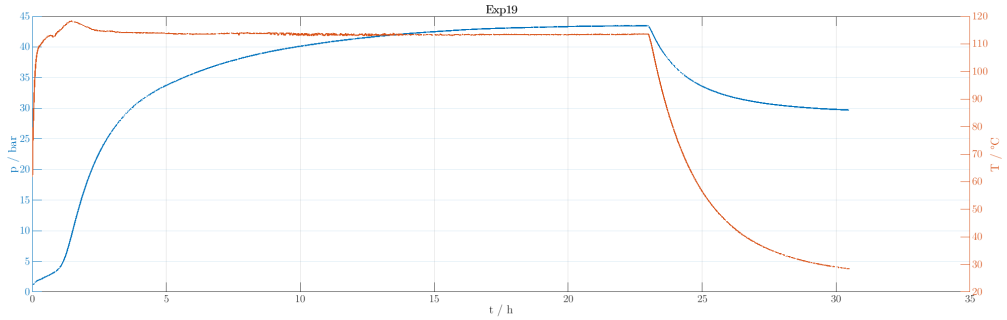


Figure 36: Experiment curves of experiment with PEEK stirrer (Exp19). **Blue** Measured pressure in *bar*. **Red** Measured temperature in °C. The fastest pressure rise coincided with the highest self-heating rate after 1 hour. After 23 hours the thermostat was turned off and the mixture cooled down.

Exp20 - Isoperibolic experiment with PEEK stirrer and TS-1 catalyst To investigate the influence of the catalyst TS-1 on the decomposition reaction, 0.9 g of the white powder was prepared inside the reactor. All the other steps were the same as in Exp19.

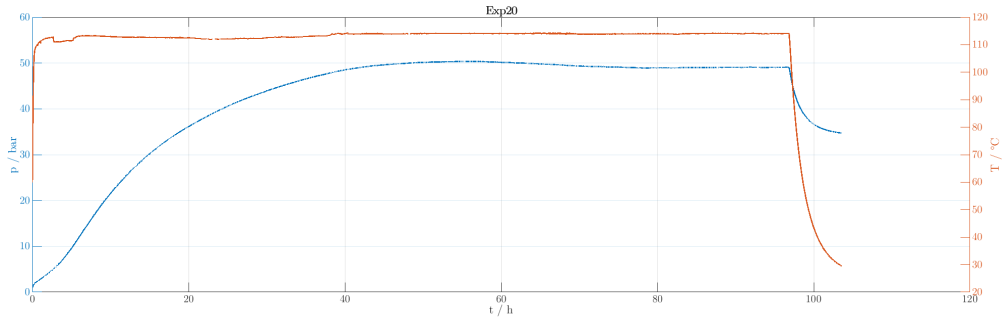


Figure 37: Experiment curves of experiment with TS-1 catalyst (Exp20). **Blue** Measured pressure in *bar*. **Red** Measured temperature in °C. There was no visible self-heating effect. The pressure rose slower than in all previous experiments and even slightly decreased after 60 hours. After 96 hours the thermostat was turned off and the mixture cooled down.

4.2 Thermodynamic modeling

4.2.1 DSC measurement

Storage experiment If DSC samples are prepared in advance and then be inserted via autosampler, the measured hydroxylamine solution must be stable at room temperature over several days. The unstabilized 50 % hydroxylamine solution was stored for three different durations in gold crucibles (C3) (for the DSC plots see appendix A.34). The energy content recovered in dynamic experiments is plotted against the storage time in figure 38.

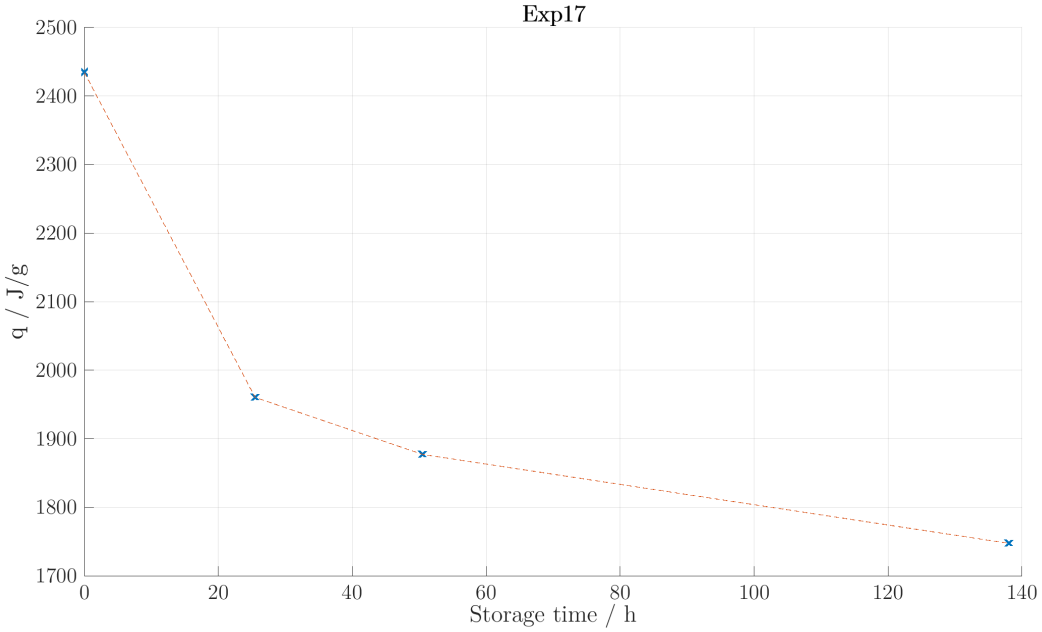


Figure 38: Plot storage time vs. energy content (DSC plots see appendix A.34). 50 % unstabilized hydroxylamine solution was stored for different times at room temperature in closed gold crucibles before measuring its decomposition enthalpy (Exp17). The energy content of the sample decreases with increased storage time.

Reaction enthalpy The reaction enthalpies can be converted from kJ/mol to J/g according to the equation

$$\frac{q}{J/g} = \frac{q}{kJ/kg} = \left| w(HA) \frac{\Delta H_r}{kJ/mol} \div \frac{MW(HA)}{kg/mol} \right| \tag{4-3}$$

To validate the three reaction hypotheses, their calculated enthalpies (see table 11). can be compared to the energy contents of the DSC measurements (shown in table 12).

Table 11: Reaction enthalpies of the three reaction hypotheses

Reaction	**Description**	**Reaction enthalpy** kJ/mol	**Energy content** J/g
R1	empiric reaction	-164.05	2483
R2	alkaline reaction	-194.43	2943
R3	acid reaction	-110.11	1667

Table 12: Average values of the energy contents of all DSC measurements of the hydroxylamine solution. q_rest is the energy recovered in the dynamic measurements after the isothermal experiments. The complete table can be found in appendix A.5.

Energy balance			
Experiment	**q / J/g**	**q_rest / J/g**	**q_tot / J/g**
Dyn4	2299		2299
Iso90	1194	657	1851
Iso110	1533	413	1946
Iso130	1984	137	2121
Iso150	2120	42	2163
SS	-	-	2167
Au	-	-	2112
All	**-**	**-**	**2128**

4.2.2 Explosion reactor measurements

The measurements from the explosion reactor give more information about which reaction hypothesis is true. In a first step, the reaction conversion α was calculated from the titration. The end pressure of every experiment could consequently be simulated for the three hypotheses. The simulated end pressures are plotted in comparison to the experimental end pressure in appendix A.6.

To model the pressure over course of the whole experiment, the reaction conversion was calculated for each moment. The function $p(T, \alpha)$ was solved for α, giving a unique solution for every p and T. In the following, all the models are applied to the experiments in the explosion reactor (see figures 39 to 44). The model curves sometimes exceed the real limit of reaction conversion $\alpha = 1$. In the upper plots the value of α_{end} measured by titration is indicated as a purple cross. To subsequently model the pressure, these curves were scaled according to α_{end}. The corrected curves are shown in the lower plots. In the first isoperibol experiment with the PEEK stirrer (Exp10) a deviation to α_{end} is to be expected because the cooling down phase was not recorded. In Exp12 (stainless steel), Exp15 (PEEK) and Exp19 (PEEK) the end point was hit perfectly by the acid model. Exp16 (HC22) and Exp20 (with TS-1) are fairly good described by the acid reaction hypothesis, with a slight deviation, however.

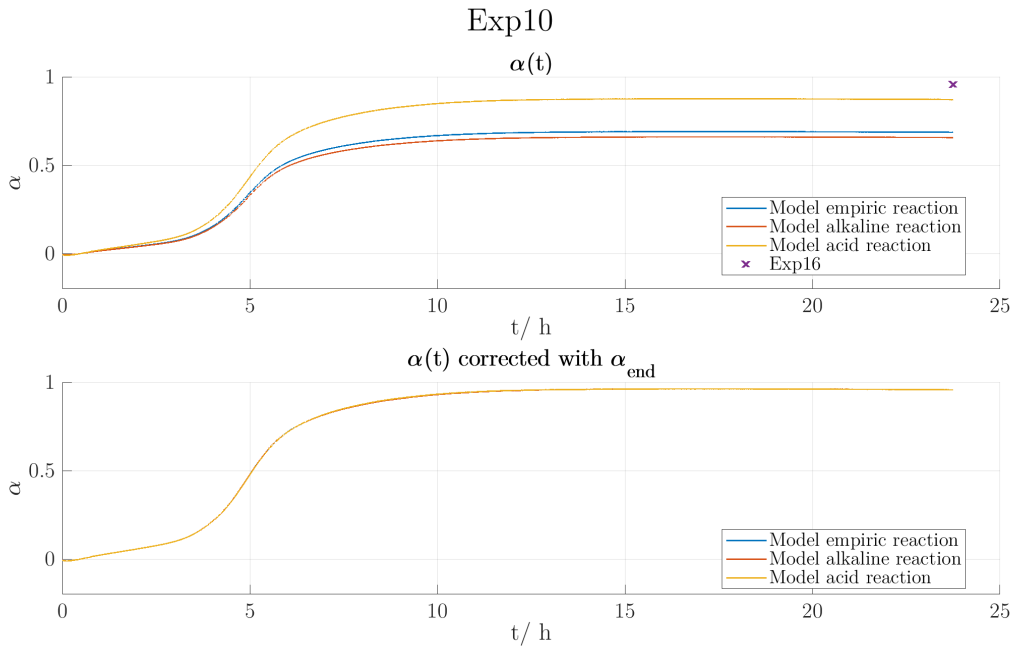


Figure 39: PEEK / Gold (Exp10) - Reaction conversion model vs. experiment. **Upper plot** The reaction conversion as a function of time compared to the value of α_{end} measured by titration (purple cross). **Lower plot** The reaction conversion as a function of time, scaled to match the experimental value of α_{end}. The deviation of all the hypotheses to α_{end} (measured of the cooled down reaction mixture) are expected, because the cooling down phase was not recorded.

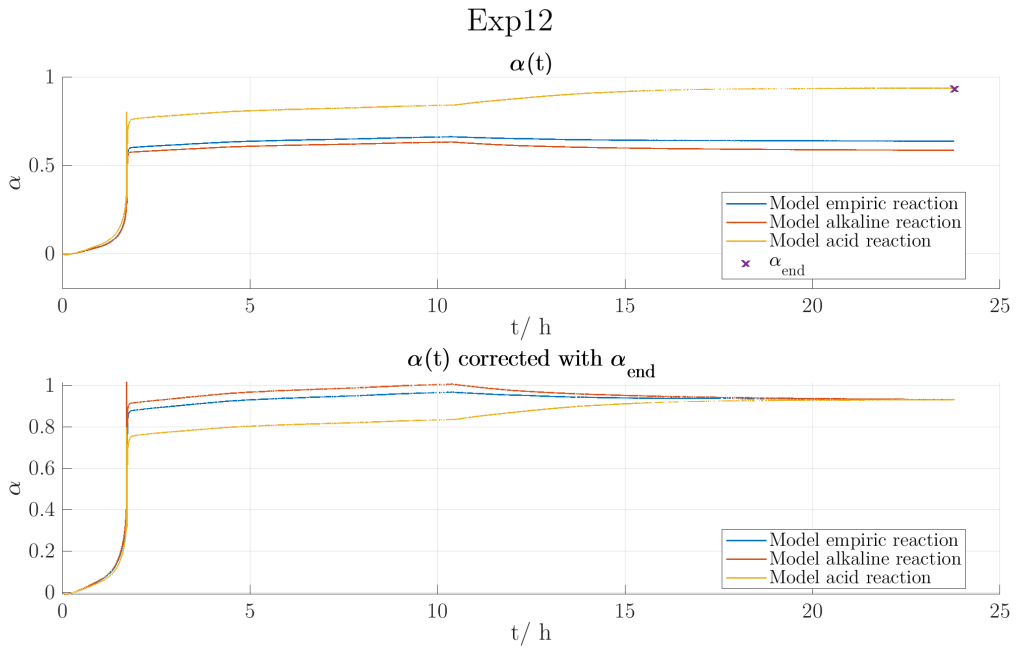


Figure 40: (Steel / Gold (Exp12) - Reaction conversion model vs. experiment. **Upper plot** The reaction conversion as a function of time compared to the value of α_{end} measured by titration (purple cross). **Lower plot** The reaction conversion as a function of time, scaled to match the experimental value of α_{end}. The acid hypothesis hits the experimental value of α_{end}. The scaled curve of the alkaline hypothesis unrealistically takes up values over 1 for the reaction conversion.

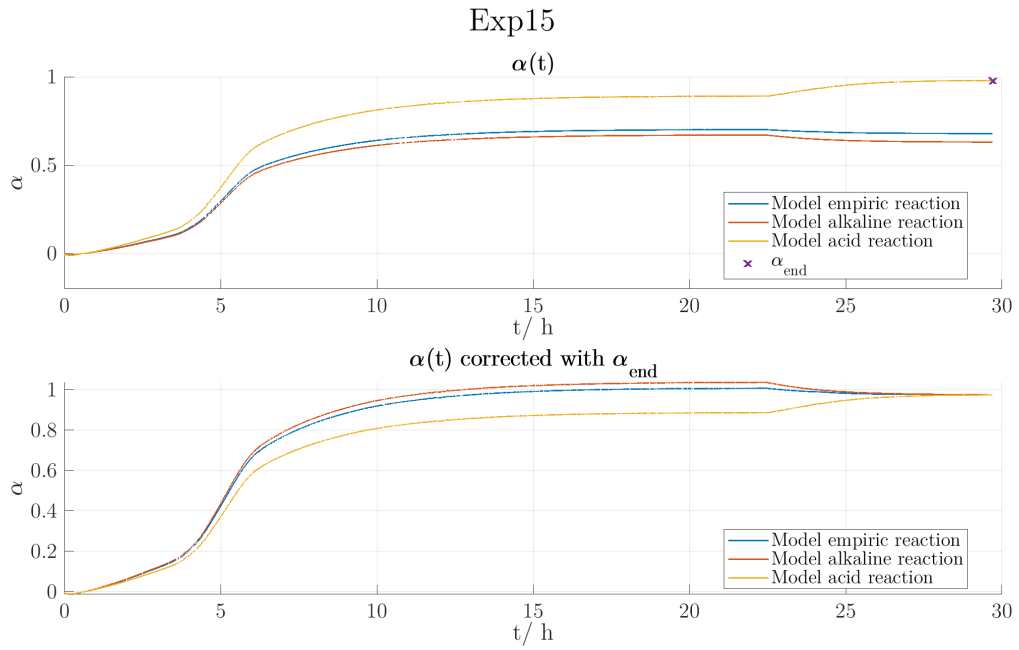


Figure 41: PEEK / Silicone (Exp15) - Reaction conversion model vs. experiment. **Upper plot** The reaction conversion as a function of time compared to the value of α_{end} measured by titration (purple cross). **Lower plot** The reaction conversion as a function of time, scaled to match the experimental value of α_{end}. The acid hypothesis hits the experimental value of α_{end}. The scaled curve of the alkaline hypothesis unrealistically takes up values over 1 for the reaction conversion. During the cool down phase, the empiric and the alkaline hypotheses suggest a shift of the equilibrium to the reactant side, while the acid hypothesis suggests a shift to the product side.

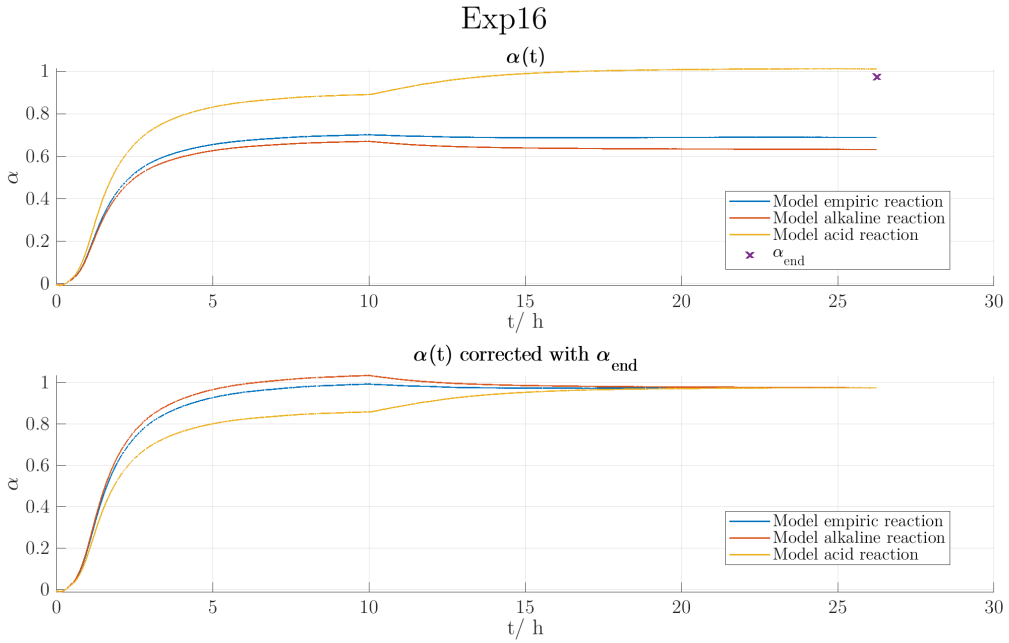


Figure 42: HC22 / Silicone (Exp16) - Reaction conversion model vs. experiment. **Upper plot** The reaction conversion as a function of time compared to the value of α_{end} measured by titration (purple cross). **Lower plot** The reaction conversion as a function of time, scaled to match the experimental value of α_{end}. The acid hypothesis almost hits the experimental value of α_{end}. The scaled curve of the alkaline hypothesis unrealistically takes up values over 1 for the reaction conversion.

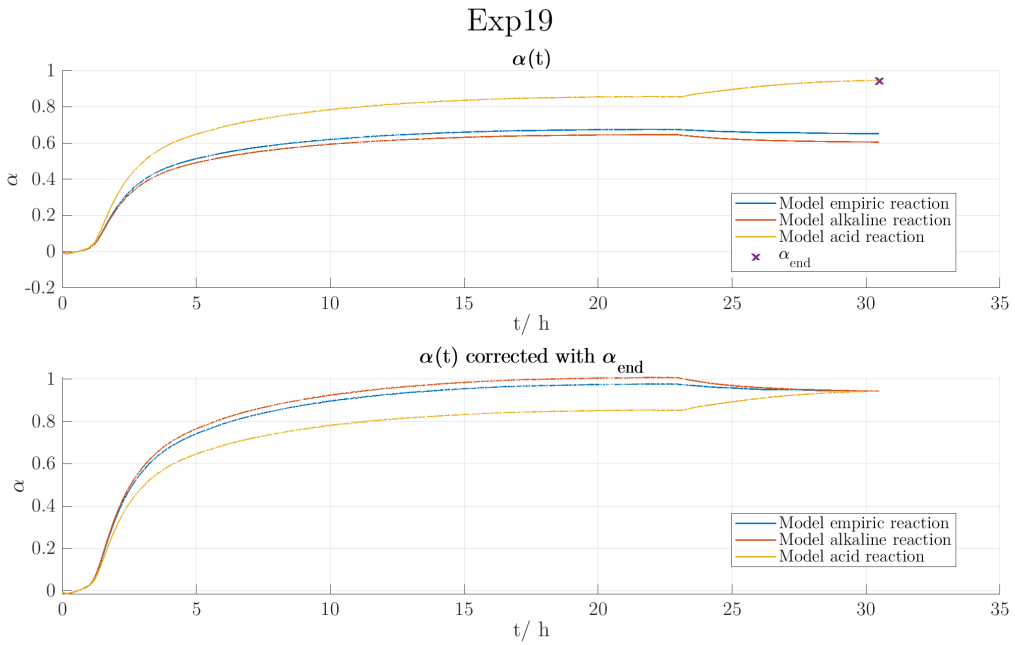


Figure 43: PEEK / steel (Exp19) - Reaction conversion model vs. experiment. **Upper plot** The reaction conversion as a function of time compared to the value of α_{end} measured by titration (purple cross). **Lower plot** The reaction conversion as a function of time, scaled to match the experimental value of α_{end}. The acid hypothesis hits the experimental value of α_{end}. The scaled curve of the alkaline hypothesis unrealistically takes up values over 1 for the reaction conversion.

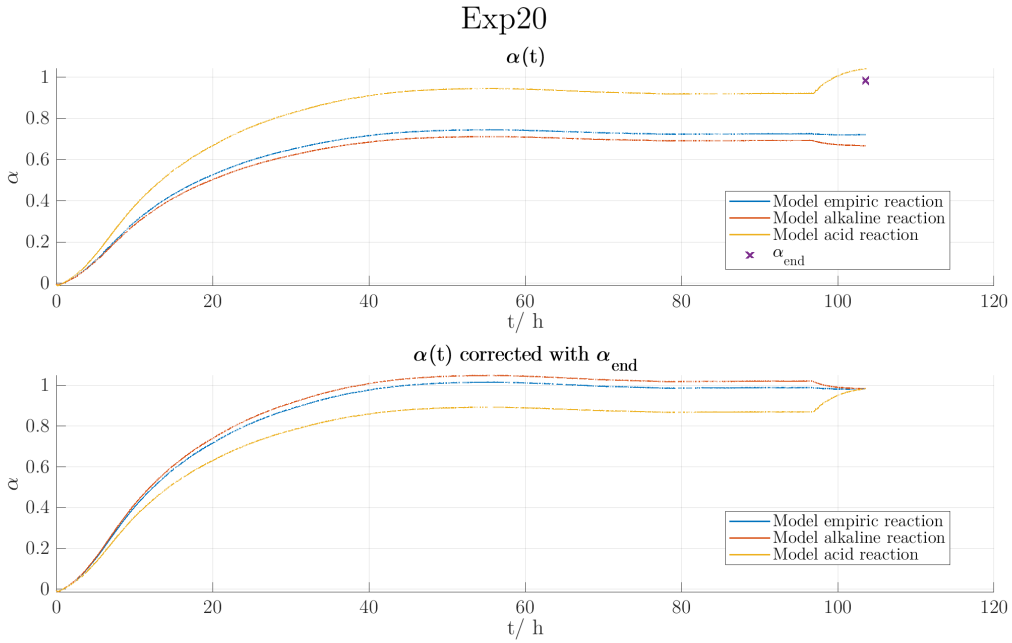


Figure 44: PEEK / steel with TS-1 catalyst (Exp20) - Reaction conversion model vs. experiment. **Upper plot** The reaction conversion as a function of time compared to the value of α_{end} measured by titration (purple cross). **Lower plot** The reaction conversion as a function of time, scaled to match the experimental value of α_{end}. The acid hypothesis curve ends closest to the experimental value of α_{end}. The scaled curve of the empiric and the alkaline hypotheses unrealistically take up values over 1 for the reaction conversion.

It is apparent that the reaction conversion changes during the cooling down of the reactor. This indicates a readjustment of the chemical equilibrium when the temperature changes. In the acid model, the equilibrium shifts to the product side when the temperature decreases. For the other two models the equilibrium shifts to the reactant side.

From the corrected curve of $\alpha(t)$ and the temperature $T(t)$, the pressure course was calculated for every model and compared to the pressure curve of the experiments (see figures 45 to 50). To quantify the goodness of the fit, the NRMSD (normalized root-mean-square deviation) was calculated according to the equations (4-5). In all the experiments, the acid reaction hypothesis describes the experimental data best. For the code, see appendix A.30.

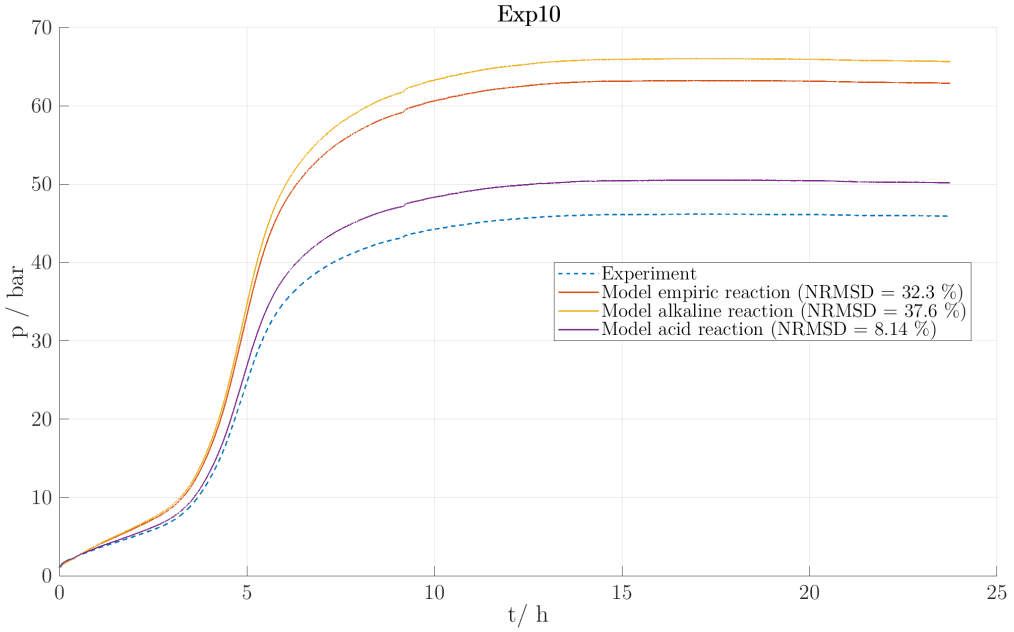


Figure 45: PEEK / gold (Exp10) - Pressure curve model vs. experiment. The pressure was modeled from the corrected reaction conversion and the temperature curve. The curve for the acid hypothesis is closest to the experimental curve.

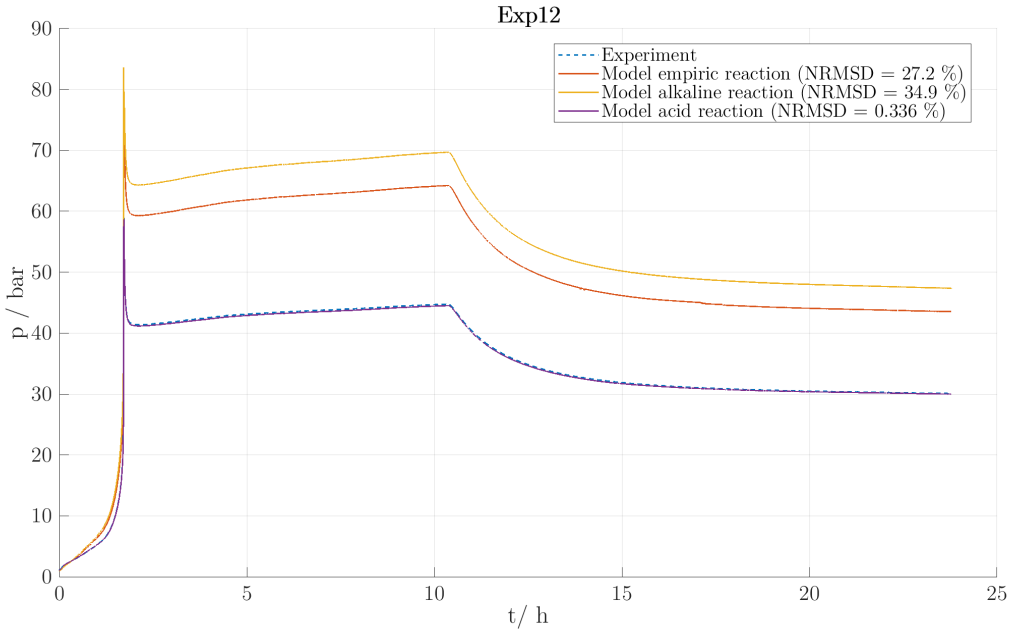


Figure 46: Steel / gold (Exp12) - Pressure curve model vs. experiment. The pressure was modeled from the corrected reaction conversion and the temperature curve. The curve for the acid hypothesis fits the experimental curve very well.

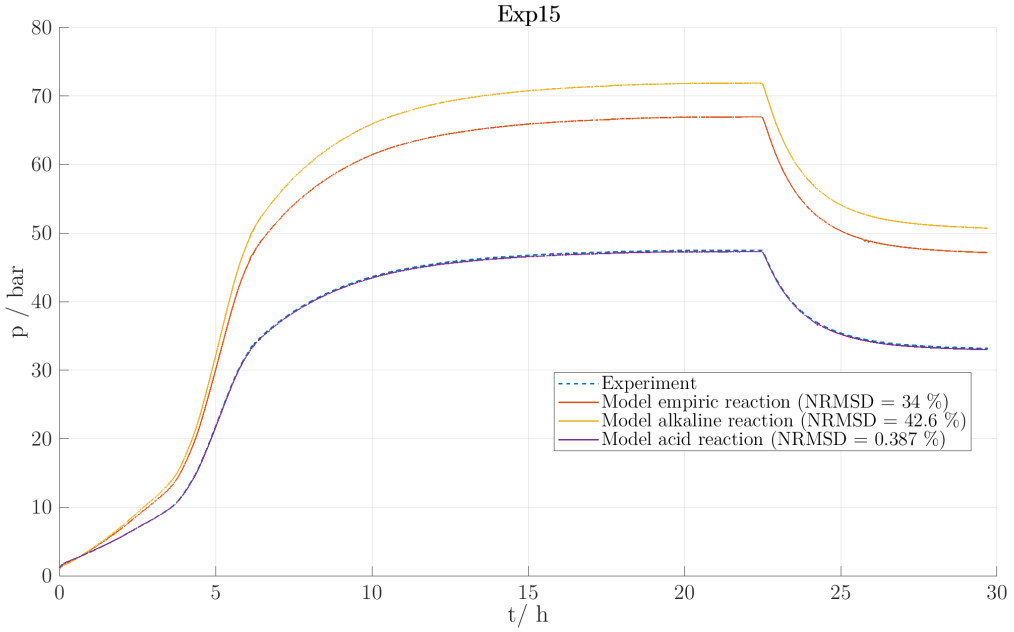


Figure 47: PEEK / silicone (Exp15) - Pressure curve model vs. experiment. The pressure was modeled from the corrected reaction conversion and the temperature curve. The curve for the acid hypothesis fits the experimental curve very well.

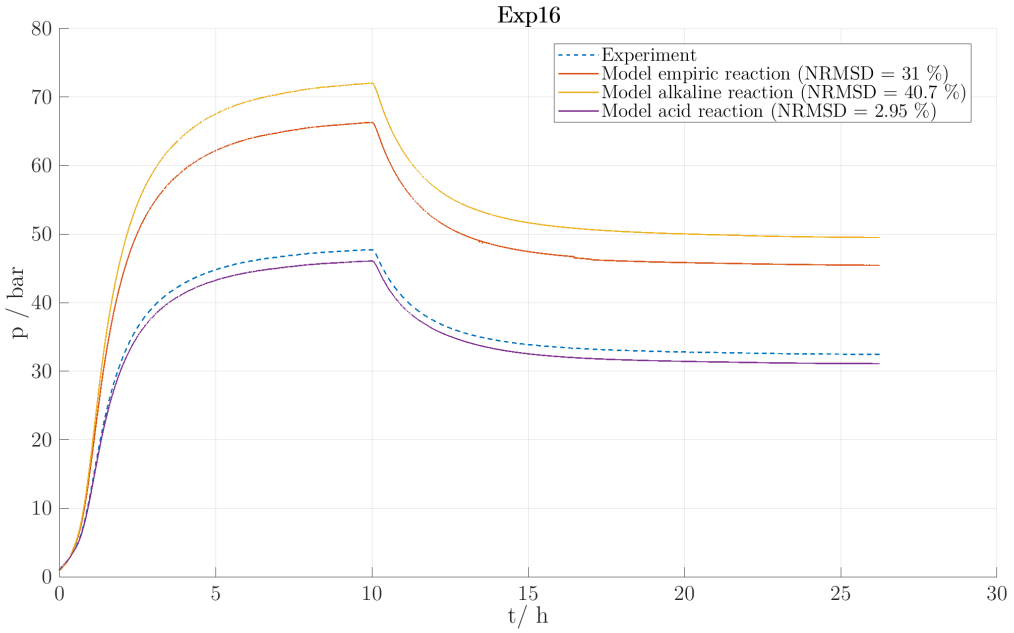


Figure 48: HC22 / silicone (Exp16) - Pressure curve model vs. experiment. The pressure was modeled from the corrected reaction conversion and the temperature curve. The curve for the acid hypothesis is closest to the experimental curve.

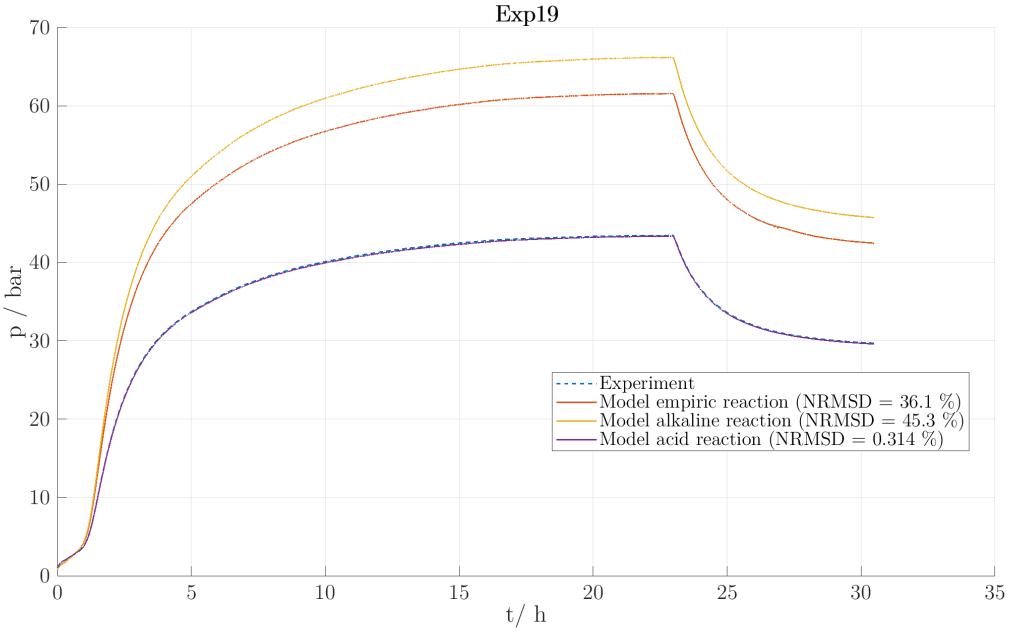


Figure 49: PEEK / steel (Exp19) - Pressure curve model vs. experiment. The pressure was modeled from the corrected reaction conversion and the temperature curve. The curve for the acid hypothesis fits the experimental curve very well.

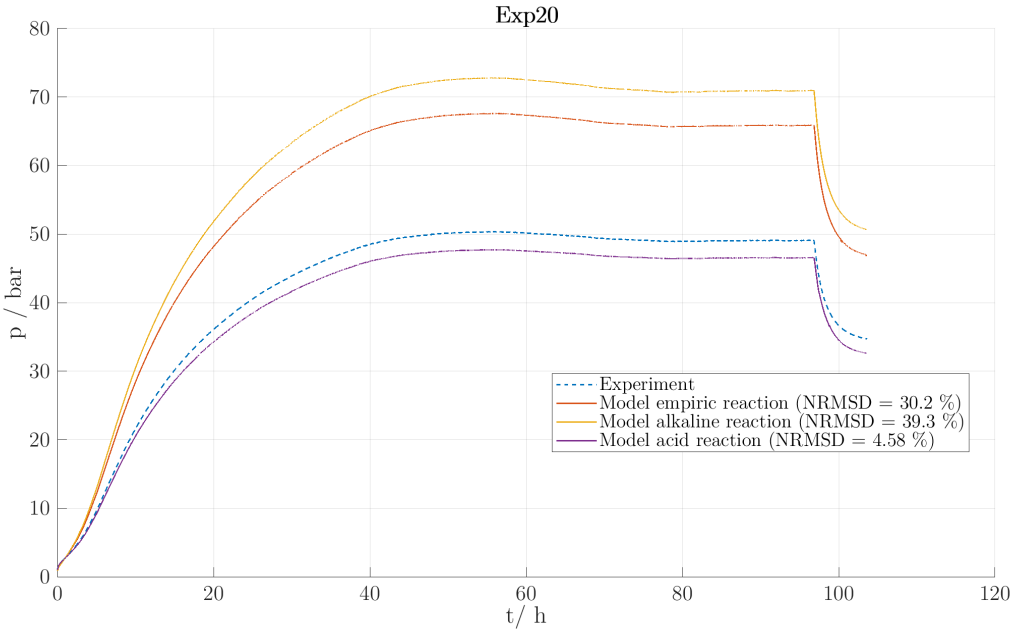


Figure 50: PEEK / steel with TS-1 catalyst (Exp20) - Pressure curve model vs. experiment. The pressure was modeled from the corrected reaction conversion and the temperature curve. The curve for the acid hypothesis is closest to the experimental curve.

4.2.3 Gas analytics

For the characterization of the gas phase, a cooling trap was used. After the gases were condensed with liquid nitrogen, they could be identified by measuring the vaporization temperature. The vapor pressure curves and the melting points of the relevant gases are shown in figure 51 (for the code see appendix A.16).

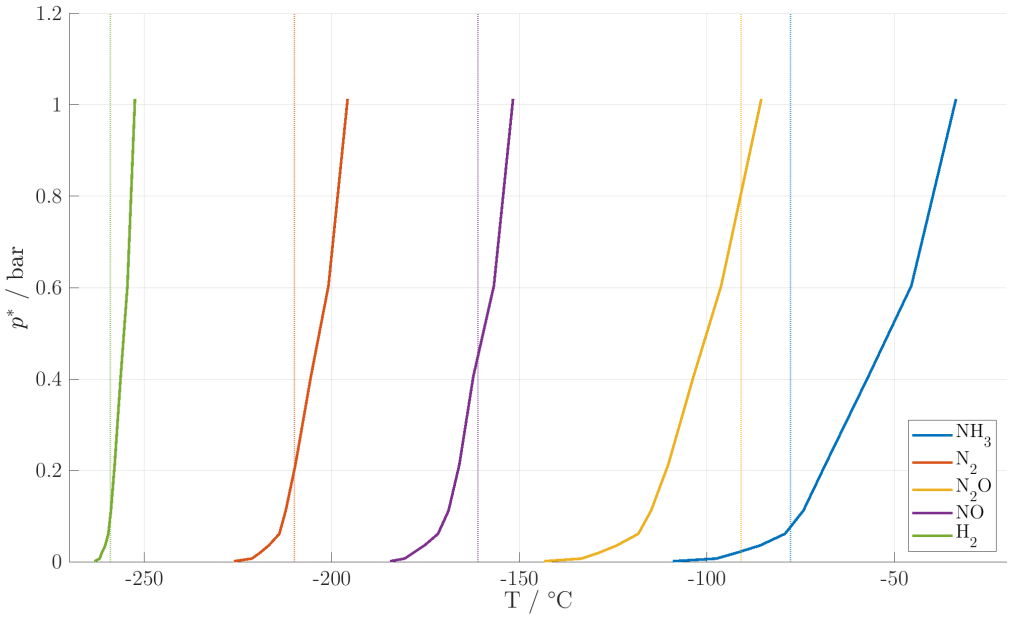


Figure 51: Vapor pressure curves (solid lines) and melting points (dotted lines) of the relevant gases. The pressure data points were obtained from tables [**perry_perrys_2008**].

For several reasons, this method was not very effective. The flow of the gas stream was either too high to be condensed or the tube bow inside the liquid nitrogen appeared to be clogged, most likely because the ammonia was crystallized at -174 °C. After continuous modification of the apparatus and the handling, the gas analytics were successful in the experiment with the HC22 stirrer (Exp16). For the results see appendix A.11.

4.3 Metal ion catalysis

Influence of the crucible material The unstabilized 50 % hydroxylamine solution was measured in three different crucibles, one thereof made from stainless steel. The stabilized solution was measured in a gold crucible. For each crucible, four isothermal (at 90 °C, 110 °C, 130 °C and 150 °C) determinations and one dynamic (4 K/min) determination were conducted. Additional dynamic experiments were included to complement the measurements. An overview of all the experiments is shown in figure 52 (for the code see appendix A.24).

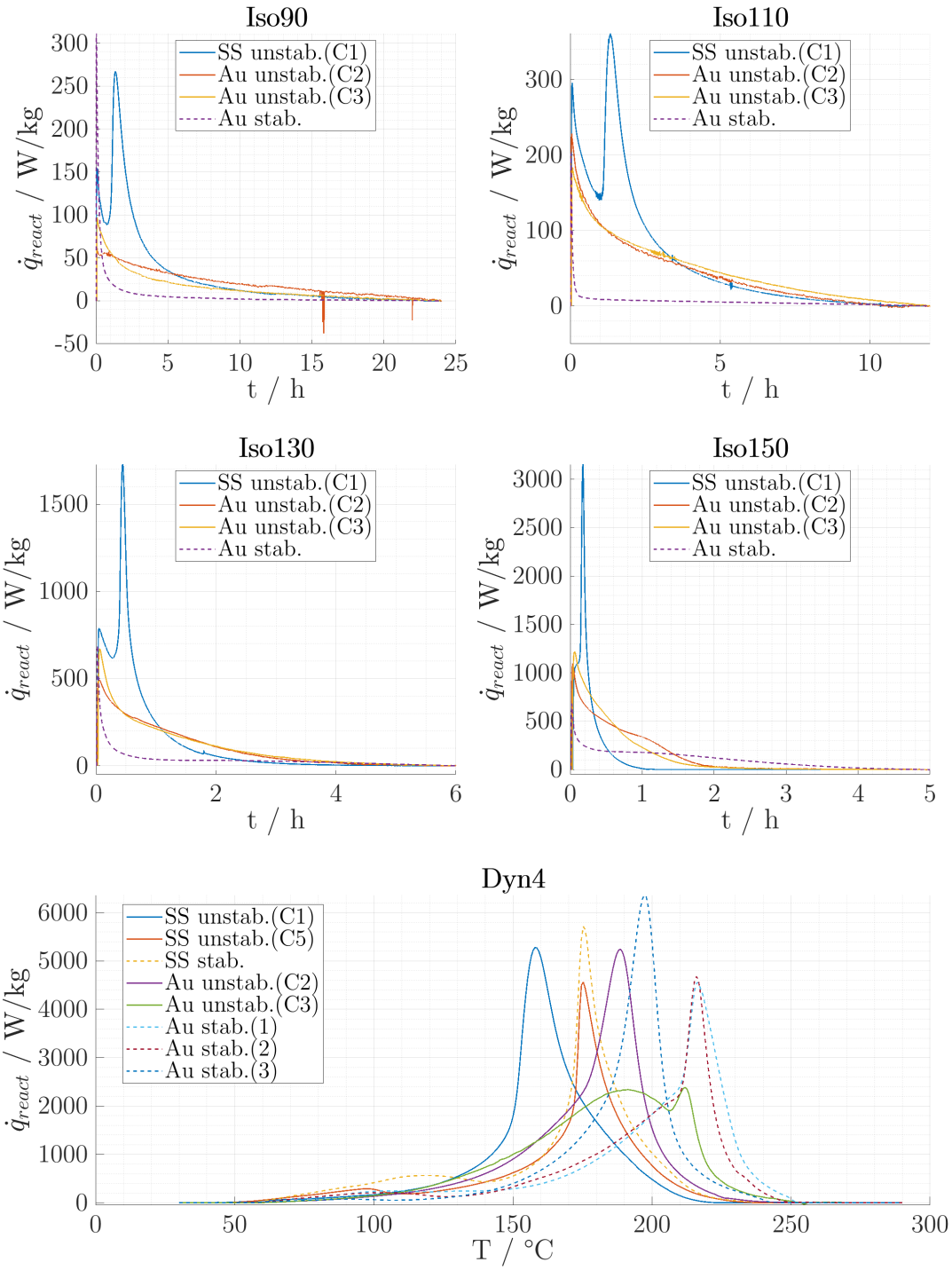


Figure 52: DSC measurements of hydroxylamine solutions in different crubicles. **Lines** Heat flow curves of the unstabilized 50 % hydroxylamine solution. **Dotted lines** Heat flow curves of the stabilized 50 % hydroxylamine solution. The isothermal DSCs show that the stabilizer inhibits the decomposition reaction significantly. In the dynamic DSCs, the peaks with stabilizer have higher onset temperatures. The measurements in stainless steel crucibles have either a higher onset temperature (dynamic) or show a distinct exothermic peak (isothermal), which suggests an autocatalytic process.

Influence of the stirrer material The ability to exchange the stirrer of the explosion reactor allows to investigate the decomposition reaction of hydroxylamine with different surface materials. The contact surface of the stirrer is significantly smaller than the one of the DSC crucibles. To adequately compare the EXR measurements, the signal of the pressure ($p(t)$) and the temperature ($T(t)$) sensors have been combined with the constant values of the reactor volume V_R, the sample mass m_S and the universal gas constant R. The obtained quantity $\Gamma(t)$ represents the specific gas production of the 50 % hydroxylamine solution. It was calculated according the equation

$$\Gamma = \frac{pV_R}{RTm_S} \tag{4-4}$$

Figure 53 shows a comparison of all the EXR experiments with 50 % hydroxylamine solution (for the code see appendix A.30).

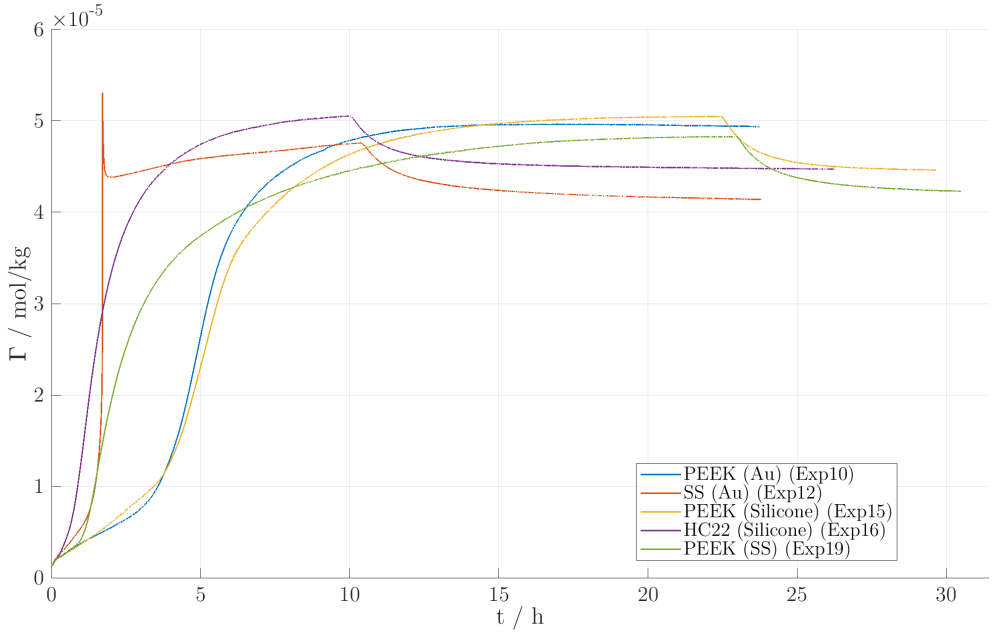


Figure 53: EXR measurements of 50 % hydroxylamine solution with different surface materials. The first mentioned material is the stirrer material, the material in parentheses is the surface material of the thermometer protection sheath. Γ is the specific gas production, which indicates the decomposition reaction. The experiment with the stainless steel stirrer (Exp12) shows a violent peak after 100 minutes. With the HC22 stirrer (Exp16) the hydroxylamine decomposed also much faster than with the PEEK stirrer. From the experiments with the PEEK stirrer, the one where the steel tube was used had the fastest decomposition.

One additional experiment was conducted to reproduce Exp12 (decomposition with stainless steel stirrer). For this purpose, the miniautoclave cells were filled with 2 g hydroxylamine solution of *BASF* and 0.2 g steel chips (V4A). An isoperibolic temperature program at 110 °C was operated (see figure 54). The initial (B_2217773) and the residual (A_3317759) solutions were analyzed by ICP-OES elementary analysis. The reports can

be found in appendix A.41. Other relevant miniautoclave measurements can be found in appendix A.40.

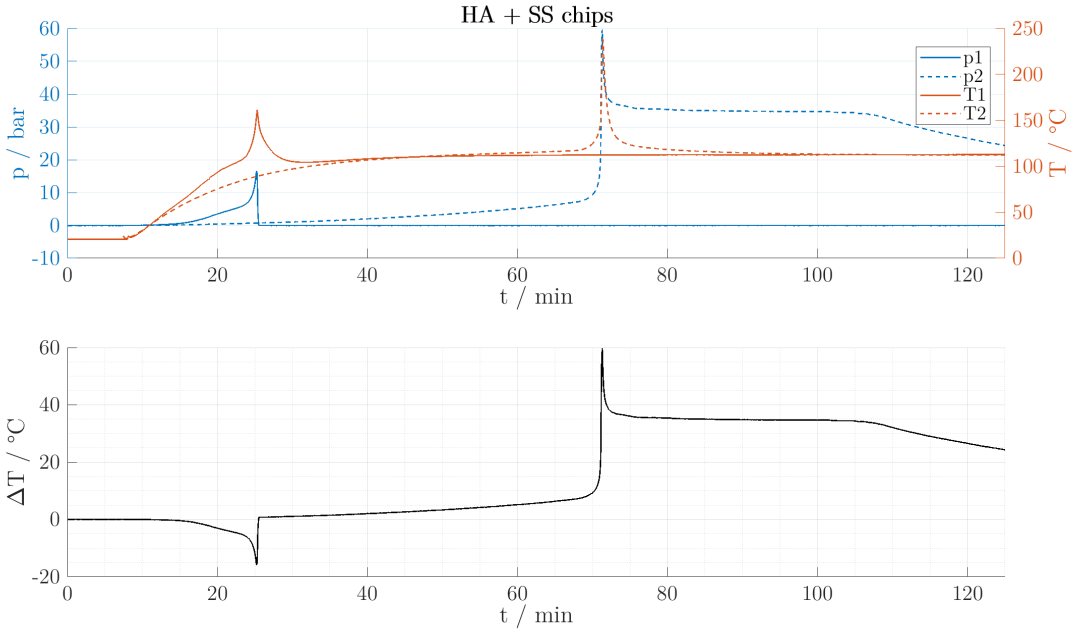


Figure 54: Experiment curves of miniautoclave experiment (double determination) of hydroxylamine solution with steel chips. **Blue** Measured pressure signals in *bar*. **Red** Measured temperature in °C. **Black** Difference of the temperature signals $T1$ and $T2$ in °C. In one case (p1/T1) the decomposition took place after 25 minutes and in the other (p2/T2) only after 70 minutes. The curve p1 indicates that the cell opened after the decomposition since the pressure curve returns to the baseline.

4.4 Thermokinetic modeling

To quantify the goodness of the fits, the NRMSD (normalized root-mean-square deviation) was calculated.

$$\mathrm{RMSD} = \sqrt{\frac{\sum_{t=1}^{T}(\hat{y}(t) - y(t))^2}{T}} \tag{4-5a}$$

$$\mathrm{NRMSD} = \frac{\mathrm{RMSD}}{\overline{y}} \tag{4-5b}$$

where a quantity $y(t)$ observed over T times is compared to an estimate $\hat{y}(t)$. Since the heat flow curves are spiky, the RMSD was normalized with the average rather than the range of the observed variable.

Descriptive model To fit the descriptive models, the `interior-point` algorithm with the `fmincon` solver was applied. A global optimization was conducted using both, `GlobalSearch` (initial estimation of the radius of a basin of attraction based on start and end point) and `MultiStart` (local optimization with multiple starting points), with

100 starting points. The parameter optimization was improved by normalizing the objective function with the average heat flow of each measurement. This helped to prevent a better fit for the dynamic measurements.

Three descriptive models were fitted to the experimental DSC curves. An n^{th} order model, where also the reaction order was fitted, delivered the best results (lowest NRMSD). A pure autocatalytic model (A.1-54b) and an autocatalytic model on top of a first order model were tried as well. The model was fitted to the measurements of C3, because this resulted in the best fit. When comparing the reaction enthalpies of those experiments (see C3 in table 11), it was chosen to exclude the isothermal experiment at 90 °C from the fit, since it has a much lower energy content. The model parameters with the best fit are as follows:

- Reaction order $n = 0.8434$
- Activation energy $E_A = 63.866\ {}^{kJ}/_{mol}$
- Rate constant $k_0 = 3.7 \times 10^{-5}\ s^{-1}$ (at $T_{ref} = 298.15\ K$)
- Average fit of the included experiments: NRMSD = 33.6 %

Isoconversional kinetics To apply the model-free kinetics to the measurements, the equation for the *Arrhenius* plot was used instead of the equation found in the literature (equation (A.1-52)).

$$\ln \dot{q} = A + \frac{E_A}{RT} \tag{4-6}$$

This equation is also used in the *AKTS* software and is a linear regression of $1/T$ and $\ln \dot{q}$. The code to calculate the fit can be found in appendix A.28. A description of the process of the fitting with the *AKTS* software can be found in appendix A.7.

Simulations In the following figures the descriptive model from C3, the isoconversional and the AKTS models of each set of measurements are used to simulate all the measurements of C1, C2, C3 and those of the stabilized hydroxylamine solution (for the code see appendix A.13 and A.24).

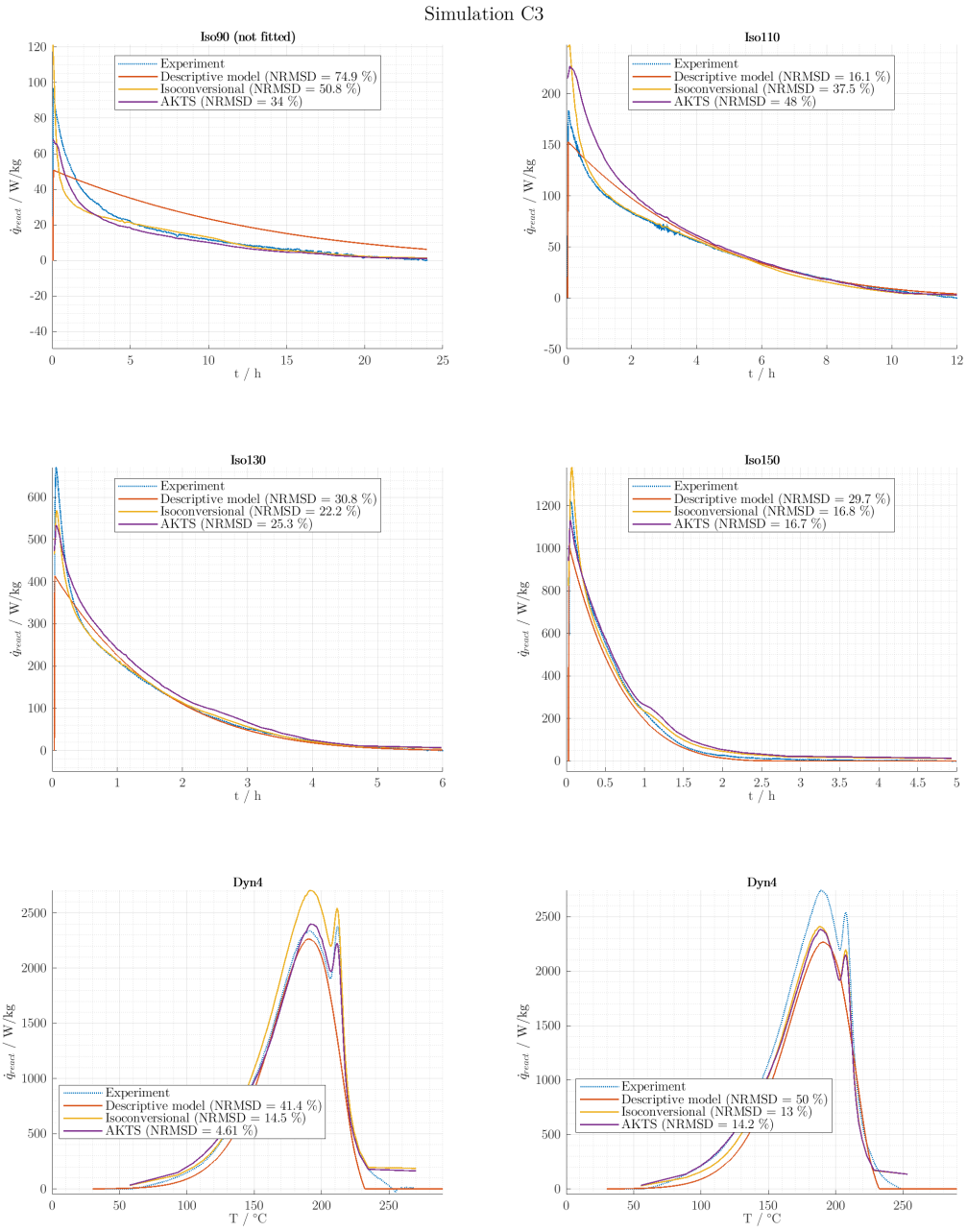


Figure 55: Kinetic models of the decomposition reaction of the unstabilized 50 % hydroxylamine solution in a gold crucible (C3). **Blue dotted lines** Heat flow curves of the unstabilized 50 % hydroxylamine solution in C3. **Lines** Kinetic models fitted to the measurements except for the isothermal determination at 90 °C. The n^{th}-order descriptive model was fitted to the isothermal measurements at 110, 130 and 150 °C as well as the two dynamic measurements with 4 K/min. The accordance with the fitted experiments is very good. The isoconversional kinetics and the AKTS curves show similar results.

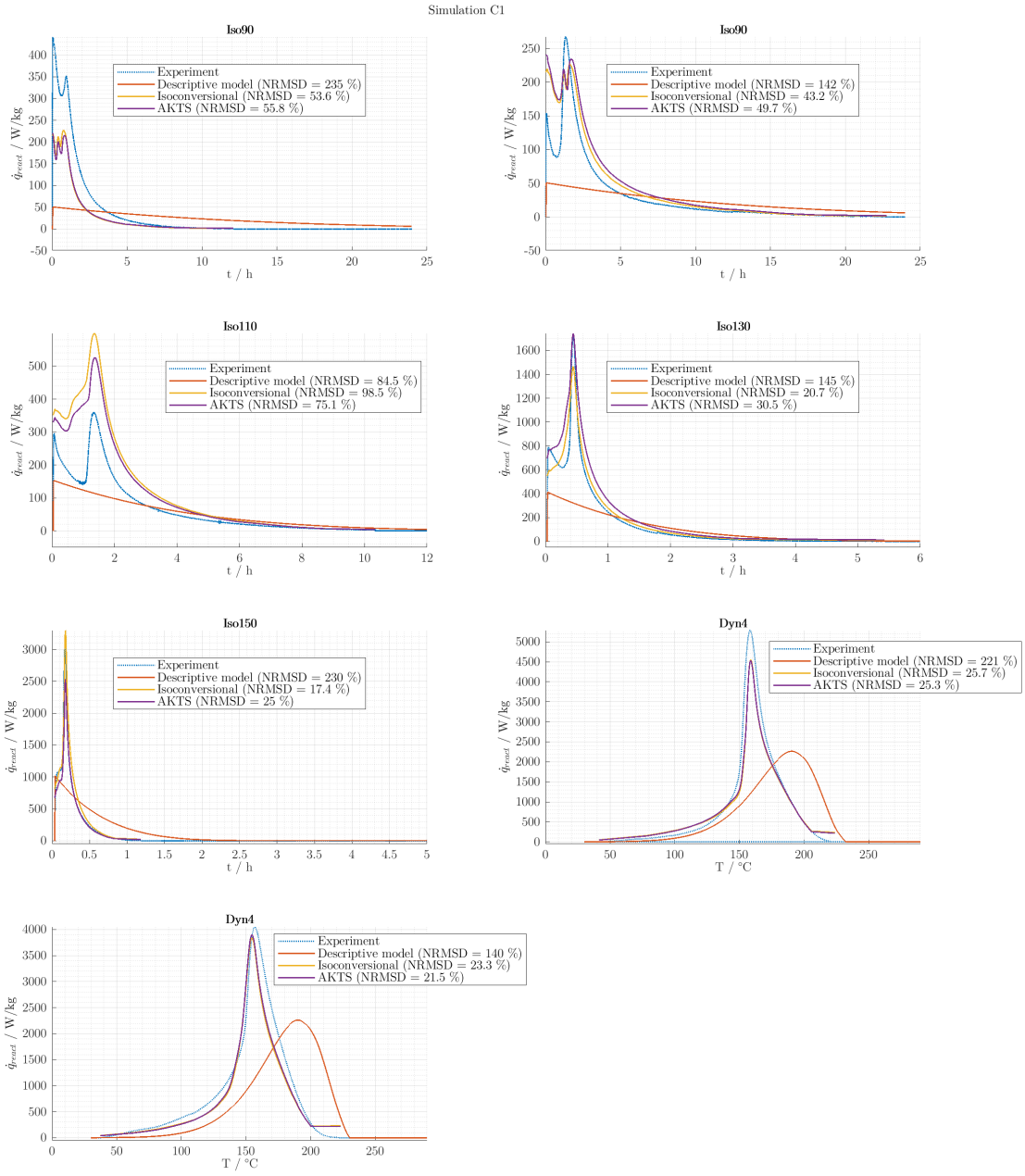


Figure 56: Kinetic models of the decomposition reaction of the unstabilized 50 % hydroxylamine solution in a stainless steel crucible (C1). **Blue dotted lines** Heat flow curves of the unstabilized 50 % hydroxylamine solution in C1. **Lines** Kinetic models fitted to the measurements in C1 (descriptive model for gold crucible C3). The isoconversional kinetics and the AKTS curves show similar results and can take the autocatalytic peak into account. However, the n^{th}-order descriptive model is a poor description of the process in the stainless steel crucible. The autocatalytic behavior cannot be represented.

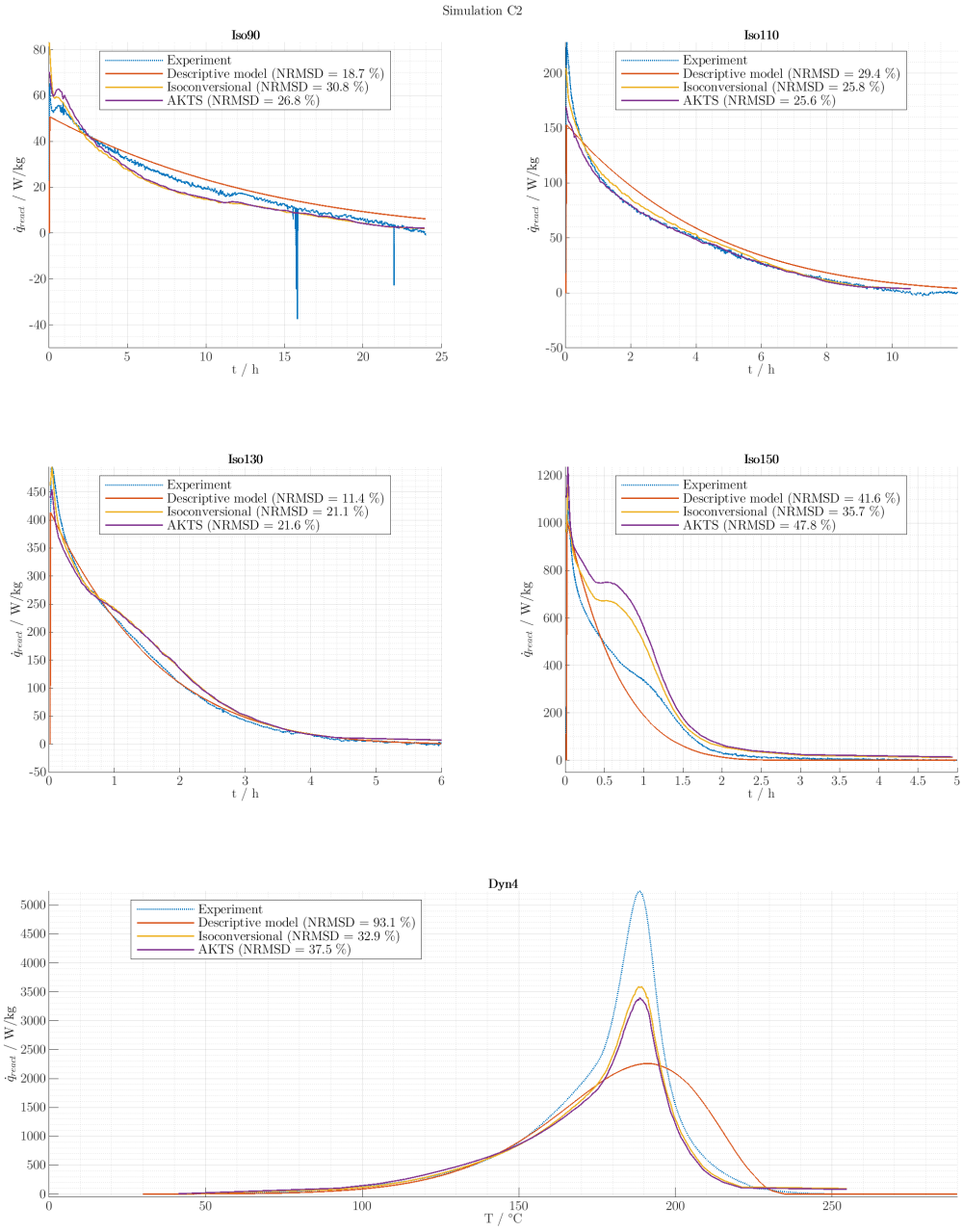


Figure 57: Kinetic models of the decomposition reaction of the unstabilized 50 % hydroxylamine solution in a gold crucible (C2). **Blue dotted lines** Heat flow curves of the unstabilized 50 % hydroxylamine solution in C2. **Lines** Kinetic models fitted to the measurements in C2 (descriptive model for another gold crucible C3). The isoconversional kinetics and the AKTS curves show similar results but have some artefacts in the simulation of an isothermal experiment at 150 °C. Except for the dynamic measurement, the n^{th}-order descriptive model fits the experimental curves well.

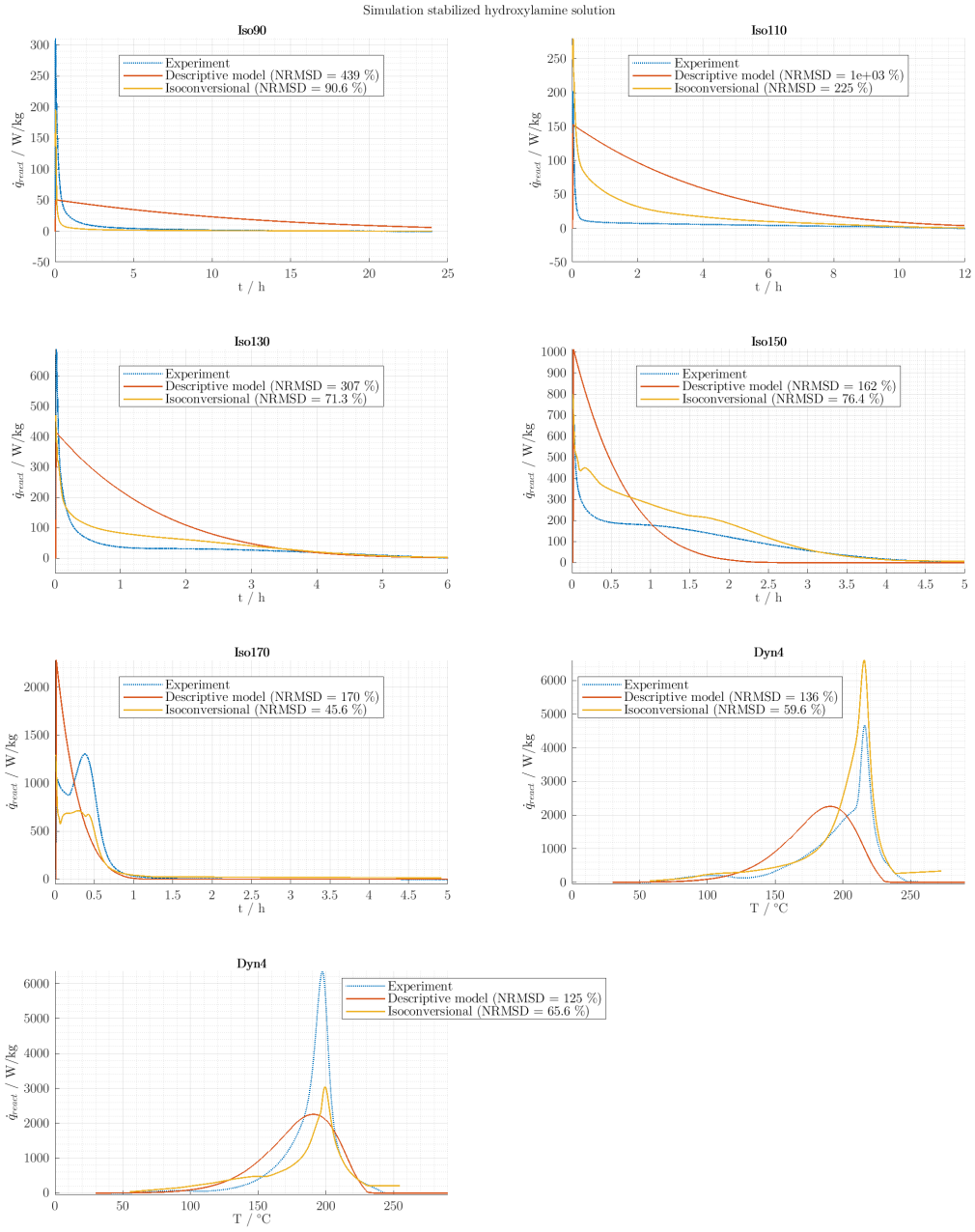


Figure 58: Kinetic models of the decomposition reaction of the stabilized 50 % hydroxylamine solution in a gold crucible. **Blue dotted lines** Heat flow curves of the stabilized 50 % hydroxylamine solution. **Lines** Kinetic models fitted to the measurements of the stabilized hydroxylamine solution (descriptive model for unstabilized hydroxylamine solution in gold crucible C3). The difference from the measurement to the n^{th}-order descriptive model fitted to the unstabilized decomposition shows the influence the stabilizer has. The curves have a much smaller integral, which indicates the inhibition of the decomposition reaction. The isoconversional kinetic fitting did not work well with these measurements and deviate significantly.

4.5 Influence of ammonia, hydrogen peroxide and TS-1 on the decomposition reactions

4.5.1 DSC measurements

In the ammoximation reaction mixture, ammonia (NH_3), hydrogen peroxide (HP) and titanium silicalite-1 (TS-1) will be present. To examine the influence of these substances on the decomposition reaction of hydroxylamine or hydrogen peroxide, several compositions were measured in the DSC with a dynamic temperature program (4 K/min).

Exp13_1 - Influence of ammonia on the hydroxylamine decomposition Three different ratios of ammonia were added to the 50 % hydroxylamine solution. The reaction enthalpies in figure 59 were normalized to the mass of the added hydroxylamine solution.

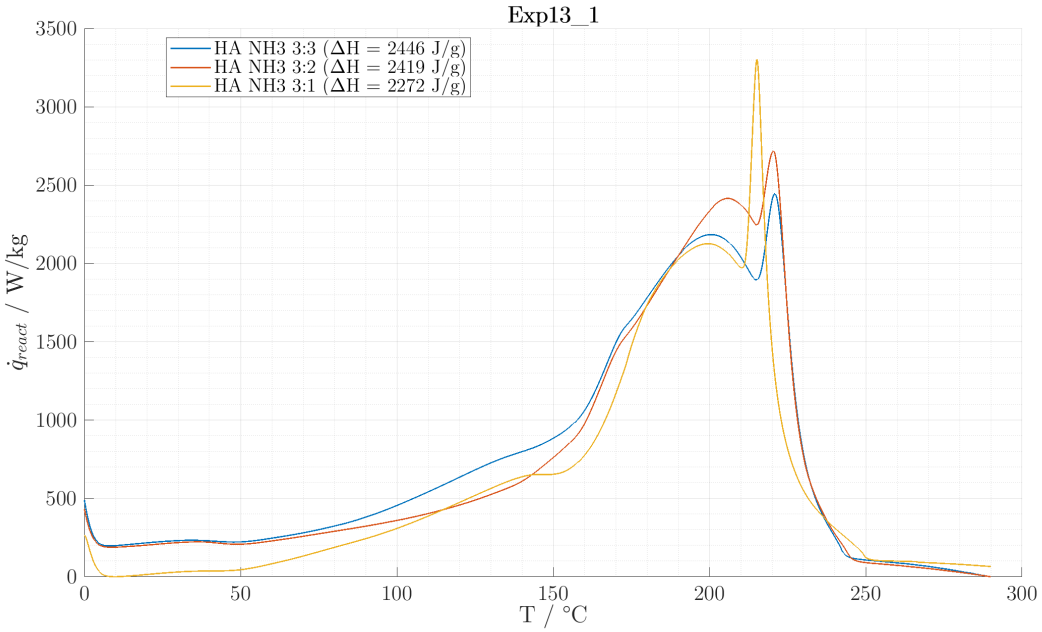


Figure 59: Experiment curves of Exp13_1. DSC measurements of 50 % hydroxylamine solution with different amounts of 25 % ammonia solution added. The heat flow was normalized to mass of the hydroxylamine solution and baseline-corrected. It can be concluded that ammonia has no significant influence on the hydroxylamine decomposition reaction.

Exp13_2 - Interaction of hydroxylamine and hydrogen peroxide decomposition In figure 60 a 1:1 mixture of a 50 % hydroxylamine solution and a 50 % hydrogen peroxide solution is compared with the measuring curves of the separate solutions. The heat flow were normalized to mass of the whole sample.

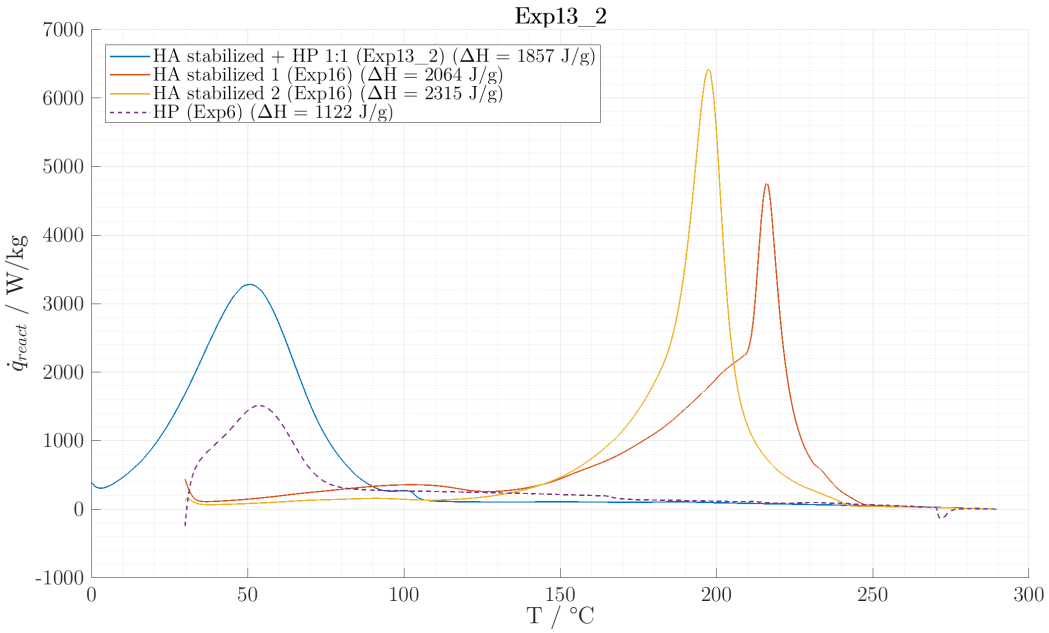


Figure 60: Experiment curves of Exp13_2. DSC measurements of a 1:1 mixture of a 50 % hydroxylamine solution and a 50 % hydrogen peroxide solution and reference experiments of the separate solutions. The heat flow was normalized to the mass of the sample. The measurements indicate that hydrogen peroxide catalyzes the decomposition reaction of hydroxylamine. The onset temperature of the hydroxylamine decomposition is shifted from about 120 °C to 5 °C when hydrogen peroxide solution is added 1:1.

Exp13_3 - Influence of TS-1 on the hydroxylamine decomposition In figure 61, a 1:1 mixture of stabilized 50 % hydroxylamine solution and TS-1 is compared with the measuring curves of the hydroxylamine solution alone. The heat flow was normalized to the mass of the hydroxylamine solution.

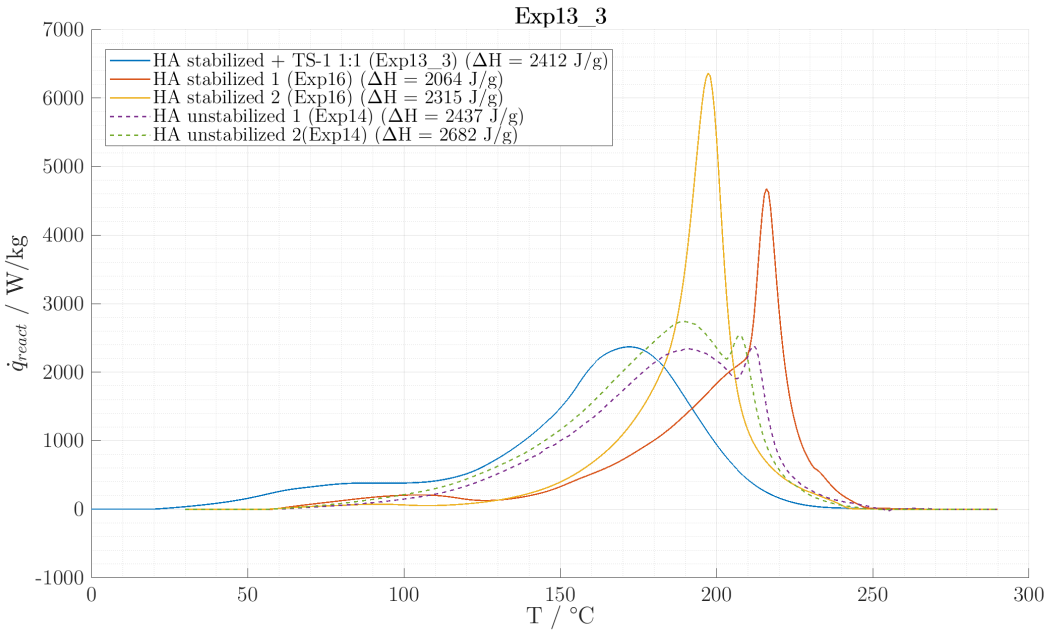


Figure 61: Experiment curves of Exp13_3. DSC measurement of a 1:1 mixture of stabilized 50 % hydroxylamine solution and TS-1. The heat flow was normalized to the mass of the hydroxylamine solution and baseline-corrected The peak temperature of the measurement with the TS-1 catalyst is lower. This indicates a weak catalytic effect of TS-1 on the hydroxylamine decomposition reaction (with stabilizer).

Exp13_4 - Influence of ammonia on the hydrogen peroxide decomposition Three different ratios of ammonia were added to the 50 % hydrogen peroxide solution. The reaction enthalpies in figure 62 were normalized to the mass of the added hydrogen peroxide solution.

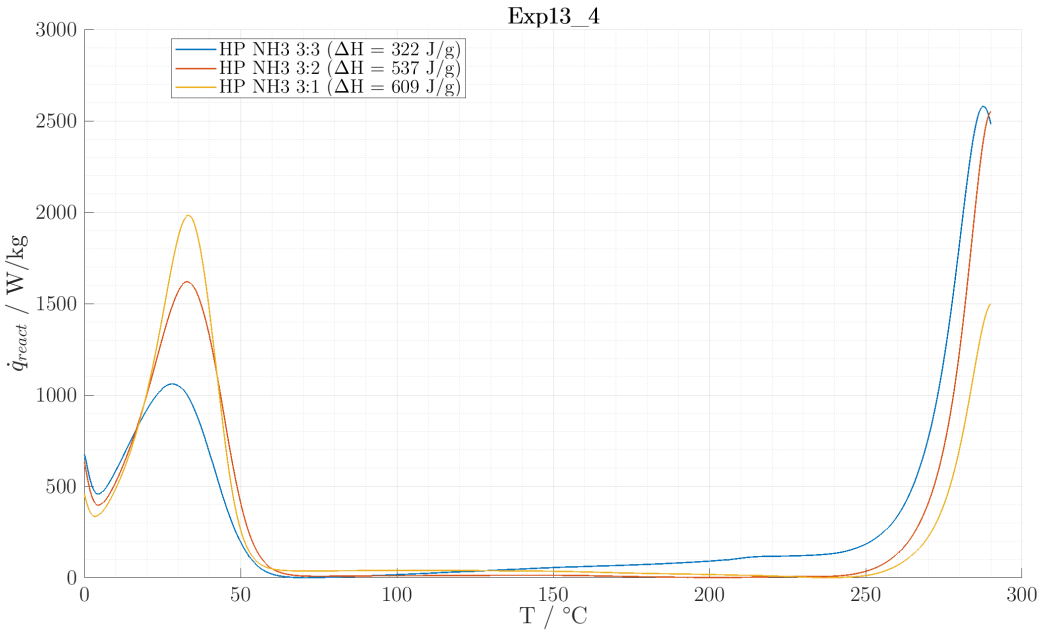


Figure 62: Experiment curves of Exp13_4. DSC measurements of 50 % hydrogen peroxide solution with different amounts of 25 % ammonia solution added. The heat flow was normalized to mass of the hydrogen peroxide solution. The more ammonia was added to the hydrogen peroxide solution, the lower was the decomposition peak at 30 °C. Compared to the measurement of the pure hydrogen peroxide solution, the peaks occur at a lower temperature (30 instead of 55 °C). Another exothermic event begins at 250 °C.

Exp13_5 - Interaction of ammonia and TS-1 In figure 63, a 1:1 mixture of a 25 % ammonia solution and TS-1 is shown. No exothermal effect is observable.

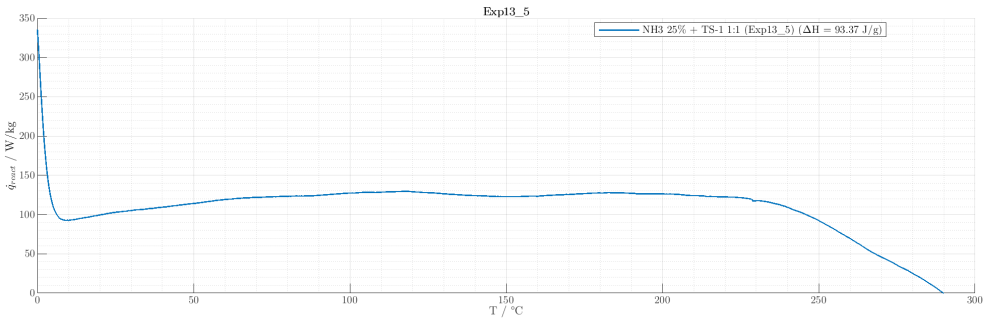


Figure 63: Experiment curves of Exp13_5. DSC measurements of 25 % hydroxylamine solution and TS-1. The mixture shows no exothermal event in the investigated temperature range.

4.5.2 EXR measurements

The influence of the catalyst TS-1 was measured in the explosion reactor by adding 0.9 g of the white powder to the 30 ml of 50 % hydroxylamine solution that was normally measured. The setup in Exp20 was exactly the same as in Exp19 (PEEK stirrer, steel thermometer protection sheath). Figure 64 shows a comparison of all the EXR experiments with 50 % hydroxylamine solution and Exp20 with added TS-1 (for the code see appendix A.30).

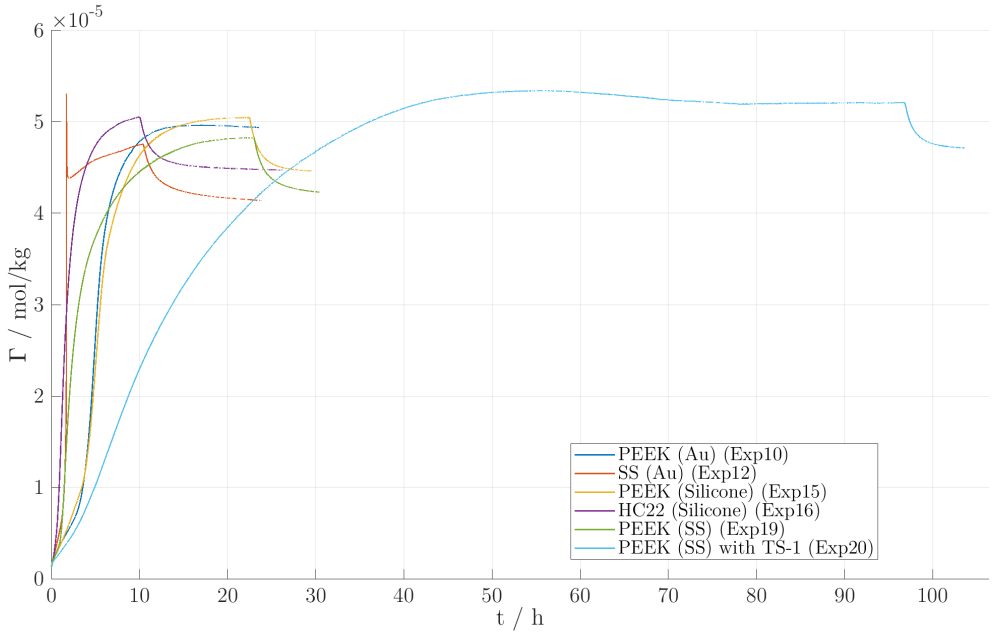


Figure 64: EXR measurement of 50 % hydroxylamine solution with TS-1. The first mentioned material is the stirrer material, the material in parentheses is the surface material of the thermometer protection sheath. The TS-1 appears to slow the decomposition reaction of hydroxylamine significantly. Instead of about 10 hours (experiments without TS-1) the equilibrium state (stable pressure) was only reached after 50 hours.

5 Discussion

Estimation methods The applied methods for the estimation of the substance data of hydroxylamine are generally not applicable for small anorganic molecules. GC methods and QSPR models are high in interpretability but low in predictability (this is a general problem of deep learning algorithms). To gain predictability, the experiments are required for every structure. Furthermore, there is no indication on the applicability of the model (e.g. coefficient of determination).

Apparatus deficiencies The gas analytics only worked in Exp16 (HC22 stirrer), but the results are still not very precise. Obviously, the cooling trap did not have enough surface to cool the gas stream. The process of heating up the tube bow led to abrupt vaporization (or sublimation) of the gases.

In the explosion reactor, the initial gas phase was supposed to be only nitrogen. However, when transferring the hydroxylamine solution into the reactor through the immersion tube, air inevitably was pulled in as well. Another problem was the immersion depth of the stirrer. The stirrer blades were fully immersed, but the stirrer did not reach to the bottom of the solution. This might have influenced especially Exp20 with the suspended catalyst. The same issue can be discussed for the temperature sensor. Only its tip dipped into the sample solution. When the reactor is not in thermal equilibrium (e.g. decomposition peak in Exp12), the measurement signal lags behind the real temperature. However, there was no obvious sign in the results that indicates this problem.

The gold coating of the immersion tube withstood all six experiments done in the explosion reactor, whereas the coating of the thermometer protection sheath failed in Exp11. The contact to 25 % ammonia solution at 110 °C and 9 bar over a period of 2 to 4 hours led the layer to drop off. As a reason, the reaction with ammonia can be ruled out because the gold coating of the immersion tube was stable. Therefore, it might have been the quality or the thickness of the layer. For a sustainable gold coating of stainless steel, the pre-gilding process is most likely not sufficient. For a resistant coating, a prior nickel layer might be advised. Another problem would be to gild the insides of a tube, because it is protected from galvanization through charge repulsion.

Thermodynamic equilibrium While there is plenty of evidence for the accuracy of the model for the phase equilibrium, the reaction equilibrium model seems to be wrong. The plots of the chemical equilibria (figures 20, 21 and 22) clearly show that the decomposition must always proceed to full extend at any reasonable temperature (25 to 200 °C). Even when a simulation was done with additional ammonia solution in the initial mixture, the equilibrium could not be shifted towards the hydroxylamine (see appendix A.12).

The results from the acidimetric determination of hydroxylamine at the end of every EXR experiment evince residual hydroxylamine even after long exposition to high temperature (over 10 h at 110 °C) and pressure stabilization. The modeled reaction conversion courses of the best fitting model (acid reaction hypothesis) also indicates a shift of the chemical equilibrium to the side product side when the temperature decreases. According

to *Le Chatelier's principle*, the heat can be viewed as a product for exothermic reactions. When the temperature decreases, one of the "products" is removed, and therefore, the equilibrium shifts to the product side. This behavior is not seen in the calculated plots. It has to be assumed, that something in the model of the chemical equilibrium is wrong. A big suspect is the standard entropy of hydroxylamine which was only calculated for the gas phase[**giguere_infrared_1952**],[**ghahremanpour_large-scale_2016**]. Another problem could be heat capacity of hydroxylamine , which was only interpolated from calculated data points.

To obtain more accurate values of the *Helmholtz* free energy, more analysis of the liquid phase has to be conducted. Another possible way would be to use the modeled reaction conversion $\alpha(t)$ in the cooling down phase of the reaction. This curve is not calculated over the reaction equilibrium (see appendix A.19), but over the phase equilibrium (see appendix A.20). For a given reaction hypothesis K_V can be calculated from α. The equation A.2-36 could be fitted to the curve and the *Helmholtz* free energy would be obtained.

Reaction route The first evidence of gas phase composition is the fact that it was not ignitable in Exp9. Therefore, there can not be huge quantities of oxidizing gases present.

Another hint gives the energy content of the DSC measurements. The average energy measured is 2182 $^J/_g$. This value lies between the values expected from the empiric (2483 $^J/_g$) and the acid (1667 $^J/_g$) reaction model. The reaction enthalpy of the alkaline reaction hypothesis predicts too much energy release. However, to trust the determination of the reaction enthalpy via DSC, the sample must be measured directly after preparation. Exp17 (figure 38) shows the problem of energy loss at room temperature. Also the sensitivity of the DSC could be problematic for the isothermal measurements at lower temperatures (90 and 110 °C). But the alkaline hypothesis can still be ruled out, since even the dynamic measurements don't indicate a energy release higher than 2299 $^J/_g$.

The first relevant result from the EXR measurements are the pressure and the hydroxylamine content at the end of each experiment. The measured pressure was compared to the one obtained with the three reaction models (see appendix A.6). It is evident that the acid model fits best in every case.

The gas analytics via cryocondensation in Exp16 is more evidence for the formation of nitrous oxide. At -150 to -120 °C, circa half the gas volume than from -120 to 23 °C flowed out of the cooling trap. This would be according to the stoichiometric ratio of nitrous oxide and ammonia in the acid reaction.

As discussed above, the thermodynamic model based on the reaction equilibrium is not accurate. However, the phase model was tested with the ammonia solution (Exp11) and described the real behavior very good. The decision as to whether to use the ideal gas law or *Henry*'s law for the gases was based on their solubility. Ammonia and nitrous oxide deviate the most from the ideal gas law and were therefore described according to *Henry*. The temperature dependence of the pressure in the steady state is mainly determined by the solubility of ammonia in water. By cooling down the reactor at the end of every experiment, the amount of ammonia in the reactor could very well be modeled. The

amount of formed ammonia differs for every reaction hypothesis. This way, the acid model could be evaluated as the most likely hypothesis for every surface material and with the catalyst TS-1(see figures 45 to 50). The term acid reaction comes from the initial reaction step in the mechanism (2-4). To assume acidic conditions are necessary for this mechanism might be wrong, because the sample solutions had pH>11 before and after the decomposition experiments. The presence of iron could be the reason for the proceeding of this reaction route, as it was mentioned by *Cisneros*[**cisneros_adiabatic_2002**].

Reaction kinetics The kinetic model obtained from the measurements in the high pressure gold crucibles (C3) can describe the measurements in the reusable gold crucibles (C2) very well. Only the dynamic measurements in C2 have spikier peaks. For the measurements in stainless steel crucibles (C1) and the measurements of the stabilized hydroxylamine solution, the n^{th} order model does not work. The isoconversional and the AKTS kinetics deliver better fits (lower NRMSD) in most of the cases. Unfortunately, without a model, only the thermal behavior of the 50 % hydroxylamine can be predicted. To model the ammoximation reaction, maybe even in loop conditions, the model-free approach is not helpful. It can not take into account concentration changes due to non-kinetic reasons.

Another possibility for the descriptive model would be the mechanism of the acid reaction (equation (2-4)) because it was identified as the most likely hypothesis. Unfortunately, it is still unclear how the iron plays into the mechanism. To establish a kinetic model that takes the iron ions (homogeneous catalysis) into account, the concentration thereof has to be determined during the reaction.

Catalysis / Reactor material By means of the DSC measurements, the influence of gold and stainless steel can be compared. The differences of the measurements in the different gold crucibles (C2, C3) are minor. It can be assumed that these differences stem from the varying quality of the gold layers. Because underneath the gold layer is stainless steel, small amounts of iron ions could find their way into the solution. A tremendous difference can be seen in the onset temperature of the dynamic measurements and in the peak form of the isothermal measurements performed in the stainless steel crucible (C1). Instead of a continuously decreasing signal in the gold crucibles, a violent second peak showed up. This second peak must stem from an autocatalytic mechanism, as it is self-accelerating. The enthalpies recovered from these measurements are comparable with the ones obtained from the measurements in the gold crucibles.

The same behavior was observed in the explosion reactor experiments (see figure 53). The effect of the surface of the thermometer protection sheath was only minor and insignificant. Much more important was the influence of the stirrer material. When the PEEK stirrer was employed, the pressure rose continuously and the exothermia led to a temperature rise of maximal 10 °C. The measurement with stainless steel stirrer (Exp12) was characterized by a thermal runaway after 100 minutes. The same happened during the miniautoclave experiment of hydroxylamine solution with steel chips. The ICP-OES report shows that the iron ion concentration is fourtheen times higher after the experi-

ment than before. This could be evidence for homogeneous catalysis by iron ions. In the isothermal DSC measurement at 110 °C in the stainless steel crucible (C1), the effect was observed as well. Additionally, Exp16 provided information about the reaction kinetics with contact to HC22. The pressure rise was much faster than with the PEEK stirrer, especially in the beginning. But no thermal runaway occurred. Interestingly, even though the stirrer material had a huge influence on the reaction kinetics, the acid reaction model seems to fit all the measurements the same.

There is additional information about the decompostion with HC22 from miniautoclave experiments in appendix A.40.

Mixture influences As expected, the DSC measurement of the mixture of a 25 % ammonia solution and TS-1 (Exp13_5) show no exothermia. In the ammoximation reaction, this component can thus be prepared together without any problem. Exp13_1 shows that ammonia has no effect on the decomposition of hydroxylamine. The reaction enthalpy and the peak form stays the same overall. Hydrogen peroxide on the other hand has a tremendous effect. The exothermal reaction peaks at 50 °C. From Exp13_4 it can be gathered that ammonia reinforces the decomposition of hydrogen peroxide. This goes along with the equation (2-8), which describes the base-catalyzed hydrogen peroxide decomposition. This behavior could also explain the early decomposition when mixed with hydroxylamine (Exp12_2). However, it looks like the hydroxylamine joins the decomposition right away because no other peak follows. The enthalpy of this peak lies between those of the pure solutions. According to the reaction network (see figure 2), this reaction would yield nitrites and nitrates. The DSC measurements also show, why the ammoximation requires a huge excess of ammonia. When hydrogen peroxide is the limiting component, it gets directly transformed into hydroxylamine and can not destroy it.

Exp13_3 indicates a catalytic effect of TS-1 on the decomposition reaction of hydroxylamine. This is contrary to the results from the measurements in the explosion reactor (see figure 64), which would indicate a stabilizing effect of TS-1. Possible reasons for the discrepancy are the use of a stirrer in the explosion reactor or the relative amounts. In the EXR 20 g/L were used, as it is optimal for the ammoximation. In the DSC experiment a 1:1 mixture was measured.

Stabilizer All the isothermal experiments yield quite different peak forms for the stabilized and the unstabilized hydroxylamine solution. The curves of the stabilized hydroxylamine decay very fast. The claims of stabilizing the solution at 100 °C made by the patent of the stabilizer [**weber_stabilizing_801**] could be true. The comparison of the dynamic DSC curves in figure 52 allows the conclusion that the stabilizer has a much bigger effect in the gold crucibles than in the stainless steel crucibles. If the stabilizer works by capturing iron ions, the ions in the gold crucible were successfully captured, while in the stainless steel crucible, the amount is too big and the stabilizer is satiated. To gain more information about the effect of the stabilizer in the hydroxylamine solution, a measurement in the explosion reactor would have been interesting.

6 Conclusion

In the present work the decomposition reaction of hydroxylamine was investigated. Of the three reaction hypotheses, the acid reaction model delivered the best fit to the explosion reaction experiments. The solution does not have to be acidic to decompose over this reaction route. This route is also favored when iron ions are present.

One of the most import lessons from this study is that metal surfaces have a huge influence on the kinetics of the decomposition reaction of hydroxylamine. Gold seems to be inert while hastelloy C22 and stainless steel accelerate the decomposition. Stainless steel can even lead to violent reactions causing thermal runaway situations. When building an in-situ generator for hydroxylamine, the selection of the reactor material is therefore one of the key aspects in optimizing the process. It is not advised to choose stainless steel. Hastelloy would be an unfavored option as well. A reactor with a gold coating could be a good solution. To create a sustainable gold layer might be tricky, as it turned out in this work. Some alternatives could be other metals like titanium or a polymer reactor made from polyurethane. Another candidate could be diamond like carbon (DLC). A dynamical DSC measurement in a DLC crucible can be found in appendix A.36. For all the alternative materials the heat transfer has to be taken into account.

Another big role plays the stabilizer. This is especially true when only small amounts of iron ions are present. In a hydroxylamine generator, the addition of a stabilizer could help to improve the yield dramatically. The optimal temperature for the ammoximation reaction is 80 °C. With a stabilizer the hydroxylamine is relatively stable at this temperature.

Most of the findings only apply to a hydroxylamine solution. Some of the substances present during the ammoximation can accelerate the decomposition of hydroxylamine. Ammonia was found to be unproblematic, apart from slightly accelerating the hydrogen peroxide decomposition. The mixture of hydroxylamine and hydrogen peroxide decomposed already at 50 °C. This might not be of concern, because in the ammoximation the hydrogen peroxide is directly transformed into hydroxylamine and is the limiting reactant. The results about TS-1 are contradictory. In the dynamic DSC measurement, it was found to slightly catalyze the hydroxylamine decomposition. In the explosion reactor experiment however, the pressure rise was by far slower, when adding TS-1 to the 50 % hydroxylamine solution.

The proposition that the chemical equilibrium can be shifted with pressure could not be fully answered. The calculations imply that it is not possible. The fact that after every explosion reactor experiment residual hydroxylamine was found and that the equilibrium shifted when cooling the reactor down indicate that the modeling of the chemical equilibrium might be inaccurate. The value which may be the problem is the standard entropy of gaseous hydroxylamine, because it was only calculated from IR spectra. More measurements are required to accurately determined this value. Then the chemical equilibrium of the decomposition reaction of hydroxylamine can be calculated.

7 Outlook

Outside the temporal frame of this work the thermodynamic model based on the reaction equilibrium was adjusted. The reaction conversions in the steady states at 110 °C (calculated with the phase equilibrium model) and room temperature (measured by titration) were used as the reference points for the chemical equilibrium. The substance property that is only based on gas phase calculations [**giguere_infrared_1952**] is the standard entropy of hydroxylamine. This value was changed until the acid reaction hypothesis fitted the reference points.

Literature [**giguere_infrared_1952**]: $S^{\ominus,(g)} = 236\ {}^{J}/_{mol*K}$ → Corrected: $S^{\ominus,(g)} = 346\ {}^{J}/_{mol*K}$ (7-1)

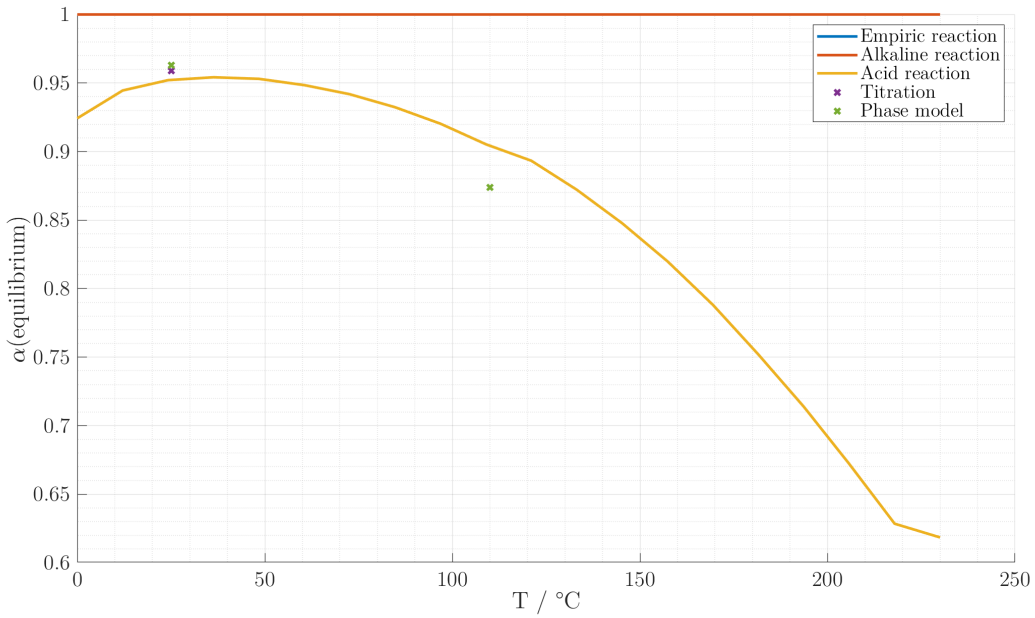


Figure 65: Chemical equilibrium as a function of temperature with the corrected standard entropy of gaseous hydroxylamine. **Purple cross** Reaction conversion measured by titration at room temperature. **Gree cross** Reaction conversion modeled from the temperature and the pressure with the phase equilibrium in the steady state. With the correction of the entropy the curve for the acid hypothesis (**yellow curve**) could be brought to the measured and the calculated reaction conversions. The curve represents the shift of the reaction equilibrium to the product side with decreasing temperature which was observed in the explosion reactor. The curves for the empiric and the alkaline hypotheses could not be moved with a reasonable adjustment of the standard entropy of gaseous hydroxylamine (code see appendix A.18).

Further insight was gained when the remaining hydroxylamine solution was destroyed. The solution was heated up to 60 to 90 °C in a beaker. The external and the internal temperatures were measured. When stainless steel chips were added no self heating effect was observed after 1 hour. The same result was obtained when adding an aqueous Fe^{3+} solution. Through addition of rust however, a different outcome resulted. The temperature in the beaker rushed from 65 to 87 °C in 5 minutes after about 30 minutes. This

rise was accompanied by intense gas evolution. With the information from *Cisneros et al.*[**cisneros_adiabatic_2002**] and this experiment it can be concluded that Fe^{2+} ions trigger the autocatalytic decomposition mechanism of hydroxylamine. In rust both Fe^{3+} and Fe^{2+} are present while in stainless steel the ions have to be produced through corrosion. Therefore, when rust is added the autocatalytic decomposition starts faster than when stainless steel is added.

The following points are advised as next steps:

- The gas resulting from the decomposition reaction of hydroxylamine should be analyzed with FT-IR to definitely prove the acid reaction hypothesis.
- More measurements (different temperatures) are to be conducted to verify the corrected entropy of hydroxylamine.
- It has to be checked if it is possible to shift the reaction equilibrium by addition of products (e.g. NH_3) (computationally and experimentally).
- Measurement of the decomposition reaction in the explosion reactor when the stabilizer is present could help to directly measure the effect of it. It should also be investigated if the stabilizer could increase the yield of the ammonia oxidation with H_2O_2 and TS-1.
- The reactor material is a crucial factor for the hydroxylamine synthesis concerning the yield and the process safety. The effect of the Fe^{2+} ions (homogeneous catalysis) has to be investigated further. It appears that rust is a big concern for process safety and should definitely be avoided.
  - It might be worthwhile to improve the gold coating technique in order to get an inert surface for the reactor.
  - Alternatively other materials like diamond-like-carbon (isothermal DSCs in DLC crucibles) could be investigated.

University of Applied Sciences and Arts Northwestern Switzerland
School of Life Sciences

Thermokinetic model of hydroxylamine decomposition in the catalytic oxidation of ammonia with titanium silicalite-1

Florian Walthert
Bachelor thesis

Principal/Expert: Dr Fabian Meemken
Supervisor: Dr Andreas Zogg, FHNW

ABSTRACT

Hydroxylamine is a key molecule for some important chemical processes. An example is the synthesis of ε-caprolactam, which is required for the production of Nylon. Since hydroxylamine decomposes thermally, its transport and storage are problematic. It is also problematic for the synthesis of hydroxylamine from ammonia with titanium silicalite-1. The goal of this study was to investigate the exothermic decomposition reaction. The heat of the reaction was measured in multiple DSC experiments. During the decomposition, vast quantities of gases are released. To measure the pressure rise, an explosion reactor was employed. Three reaction hypotheses have been tested. With the use of a mathematical model, one hypothesis could be proven. The material influence on the decomposition reaction was examined for PEEK, stainless steel and hastelloy C22. From the DSC measurements, a descriptive model and isoconversional kinetics were obtained.

INTRODUCTION

Since hydroxylamine has been used industrially, several explosion accidents happened, mostly during its distillation. It is known that hydroxylamine decomposes at elevated temperature[1]. Several reactions may occur during hydroxylamine decomposition. The reaction hypothesis can be based on the dependence on pH. The alkaline reaction

$$3\,NH_2OH \longrightarrow N_2 + NH_3 + 3\,H_2O \qquad \Delta_R H = -576.1\ {}^{kJ}/_{mol}$$

and the acid reaction

$$4\,NH_2OH \longrightarrow N_2O + 2\,NH_3 + 3\,H_2O \qquad \Delta_R H = -556.0\ {}^{kJ}/_{mol}$$

are two often proposed reactions[2]. By quantifying the resulting gases and the ammonia in the solution, a third reaction equation was formulated (empiric reaction)[1]:

$$157\,NH_2OH^{(l)} \longrightarrow 61\,NH_3{}^{(g)} + 35\,N_2{}^{(g)} + 12\,N_2O^{(g)} + 2\,NO^{(g)} + 143\,H_2O^{(g)} + H_2{}^{(g)}$$

RESULTS

The pressure curve was calculated for every model based on the temperature and compared to the experimental pressure. The acid reaction model delivered the best fit for all stirrer materials (PEEK, stainless steel, HC22).

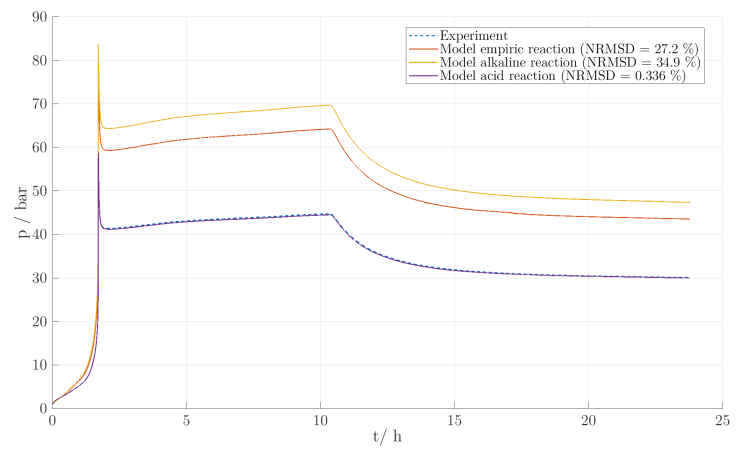


Fig. 1: The pressure curve of the hydroxylamine decomposition with a stainless steel stirrer was modeled with three reaction hypotheses

This experiment in figure 1 was conducted with a stainless steel stirrer and is characterized by the sharp pressure rise after 100 minutes. During this phase of a thermal runaway, the reaction proceeded almost adiabatically. The other experiments did not show such a violent behavior. In order to adequately compare the explosion reactor measurements, the signal of the pressure $p(t)$ and the temperature $T(t)$ sensors have been combined with the constant values of the reactor volume V_R, the sample mass m_S and the universal gas constant R. The newly obtained quantity $\Gamma(t)$ represents the specific gas production of the 50% hydroxylamine solution. It was calculated according the following equation:

$$\Gamma = \frac{pV_R}{RTm_S} \qquad [\Gamma] = {}^{mol}/_{kg}$$

The specific gas production was calculated for the experimental data of the explosion reactor experiments (see figure 2).

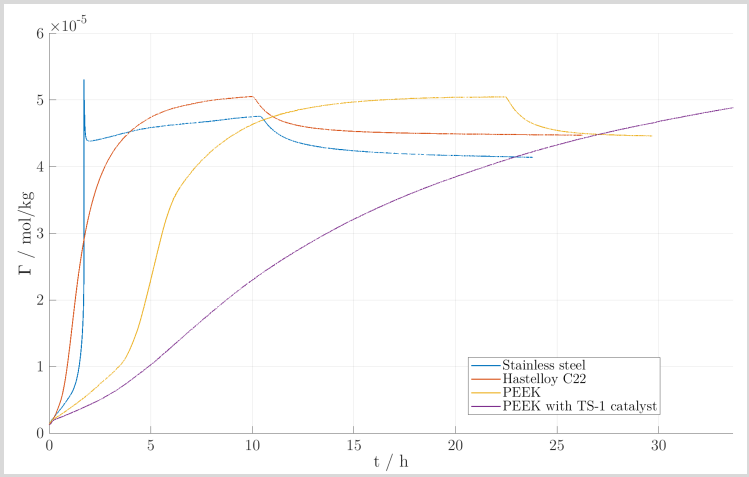


Fig. 2: Experimental curves of the hydroxylamine decomposition with different stirrer materials and with added catalyst

As already described, steel has a strong catalytic effect. Hastelloy C22 only slightly accelerates the decomposition, while PEEK can be compared to the DSC measurements in gold crucibles. The catalyst titanium silicalite-1 appears to stabilize the hydroxylamine solution. This behavior is an advantage for the synthesis of hydroxylamine from ammonia and hydrogen peroxide.

A descriptive n^{th} order kinetic was modeled to dynamic and isothermal DSC measurements of the hydroxylamine decomposition in gold crucibles. The model is represented by the following equation:

$$\frac{d[NH_2OH]}{dt} = 0.37 * 10^{-5} * e^{\frac{63870}{R*f(T)}} * [NH_2OH]^{0.84}$$

CONCLUSION

This work delivers the foundation for the engineering of a reactor to produce hydroxylamine with TS-1. The importance of the selection of the reactor material was demonstrated with measurements in the DSC and the explosion reactor. The reaction routes of all the experiments were proven to follow the acid reaction hypothesis. The obtained kinetic model can be used to predict the thermal behavior of hydroxylamine solutions, but it is only applicable without surface catalysis.

REFERENCES

[1] L.O. Cisneros, Adiabatic calorimetric studies of hydroxylamine compounds, 2002, Texas A&M University.

[2] K. Jones, Nitrogen, In: Comprehensive Inorganic Chemistry, 1973

Appendix

A.1 Addition to theoretical background

A.1.1 Prediction of physical and thermodynamic properties

Many engineering problems depend on reliable substance data. To create efficient and safe processes, the physical and thermodynamic properties of the involved chemicals have to be known. The literature is full of information about the most important substances that are used commonly. However, a huge part of this data is company property. Another portion is commercialized in databases like *DDBST* and *Detherm*. But also these sources are far from complete. Many substances are still not measured thoroughly. In such cases, there are multiple methods available to predict the physical and thermodynamic values with different uncertainties or to correlate their behavior when conditions (e.g. temperature) change.

A.1.1.1 Classification of estimation methods

To have a better overview of the estimation methods they can be divided into the following six categories. However, many of them are a combination of two categories.

Theory and empirical extension of theory Rather than just fitting empirical data in this method physical knowledge is included in the equations. This increases extrapolation capability drastically. The theory often has to be simplified with assumptions to be able to formulate a manageable equation. An example of the extension of theory is the prediction of the vapor pressure p^*. The theory provides the *Clausius-Clapeyron* equation

$$T\frac{dp^*}{dT} = \frac{\Delta H_v}{\Delta V_v} \tag{A.1-1}$$

which can be simplified for low pressures to the equation

$$\ln p^* = A - \frac{B}{T} \tag{A.1-2}$$

The empirical parameter B represents

$$B = \frac{\Delta H_v}{R\Delta Z_v} \tag{A.1-3}$$

where ΔH_v and ΔZ_v are the changes of enthalpies and compressibility factors between vapor and liquid states. In a narrow temperature range, vapor pressures can be extrapolated thanks to the inclusion of a parameter with physical meaning[**perry_perrys_2008**],[**baehr_thermodynamik_2012**].

Corresponding states (CS) The method of corresponding states is only applicable for conformal fluids. Two molecules are conformal if their intermolecular interactions remain equivalent when scaled in dimensionless form. These conformal fluids will have the reduced coordinates such as the reduced temperature ($T_r = T/T_c$) and the reduced pressure ($p_r = p/p_c$). The main accuracy limitations of this method stem from nonconformality. For example, spherical or polar molecules are not well described by scaled properties. The effect of nonconformality can be corrected by adding deviations of reference fluids to one or more reference points.

Group contributions (GC) Physical and chemical properties of substances often correlate strongly with their molecular structure. Regression of huge amounts of experimental data results in tables with the summative effect of functional or structural groups on the physical values. When using a GC method, the molecule is broken down to these groups. Every group has a certain influence to the predicted unity. The reliability of this approach is largely dependent on the quality of the regression. The more substances that were measured, the more accurate the contribution of the structural groups. Values obtained with this method should not be extrapolated.

Computational chemistry (CC) By applying the *Schrödinger* equation for the form of the wave function of atoms structural information, electron/charge information, internal vibrational modes and energies of molecules can be gained. This approach is primarily suited for the properties of gas molecules, because every molecule is calculated individually. In a chosen model chemistry, relative energies (e.g. enthalpies or entropies of reaction) are more accurately predicted than absolute ones.

Empirical QSPR correlations Opposed to the GC regressions, in quantitative structure property relationship (QSPR) correlations, not only the structural properties but also electronic descriptors are regressed. To accurately predict physical values, all the relevant descriptors are required. Therefore, this method is mainly useful for families of compounds for which the QSPR correlation was developed.

Molecular simulations Molecular simulations are primarily suited to simulate a bulk of solids or fluids. For a small amount of molecules ($\sim 10^3$) Newton's equations of motion can be applied. In doing so, the evolution (e.g. equilibrium or transport properties) of the system is predicted (molecular dynamics). Many physical properties, like vapor pressure and saturated densities, can be calculated by the potential energy between molecules. These simulations are called *Monte Carlo* simulations (MC)[**perry_perrys_2008**].

A.1.1.2 Critical properties

The critical properties of a substance are determining the phase boundaries and consist of the critical temperature T_c, critical pressure p_c and critical volume V_c. Especially T_c and p_c are very often used in thermal and volumetric property calculations with CS methods.

The critical temperature is the highest temperature at which a substance exists as a liquid. Above this temperature, no matter the pressure of the system, condensation is impossible. At the critical temperature the vapor pressure of the substance is the critical pressure. With the ideal gas constant R, one can calculate the critical volume for a given number of atoms (typically 1 mole)[**perry_perrys_2008**]. Additionally, the compressibility factor at the critical point is defined as the critical compressibility factor[**abe_critical_1989**]. It is obtained by the equation

$$Z_c = \frac{P_c V_c}{R T_c} \tag{A.1-4}$$

Methods For inorganic molecules a simple correlation to the normal boiling point is as follows[**perry_perrys_2008**]:

$$T_c = 1.64 T_b \tag{A.1-5}$$

Another approach is the earliest estimation established in 1887, called the *Guldberg* rule

$$\frac{T_b}{T_c} = \frac{2}{3} \tag{A.1-6}$$

A widely used GC method is the *Joback* method. It is mostly suited for organic compounds. The group contributions Δ_T, Δ_p and Δ_V can be taken from tables[**gmehling_chemical_2012**],[**perry_perrys_2008**].

$$\frac{T_c}{K} = \frac{T_b}{K}\left[0.0584 + 0.965 \sum \Delta_T - \left(\sum \Delta_T\right)^2\right]^{-1} \tag{A.1-7a}$$

$$\frac{p_c}{bar} = \left(0.113 + 00032 n_A - \sum \Delta_p\right)^{-2} \tag{A.1-7b}$$

$$\frac{V_c}{cm^3/mol} = 17.5 + \sum \Delta_V \tag{A.1-7c}$$

Furthermore, the critical density ρ_c can be calculated according to equation

$$\frac{\rho_c}{bar} = \left(0.113 + 0.0032\ n_A - \sum \Delta_p\right)^{-2} \tag{A.1-7d}$$

It is recommended to check the consistency of the calculated critical properties with the critical compressibility factor derived from equation(A.1-4). Z_c usually takes a value from 0.21 to 0.29[**gmehling_chemical_2012**].

The *Ambrose* method is another GC method, is applicable for organic compounds and takes alkane branches into account. The critical properties are calculated with the equations

$$T_c = T_b\left[1 + \left(1.242 + \sum \Delta_T\right)^{-1}\right] \tag{A.1-8a}$$

$$\frac{p_c}{bar} = \frac{MW}{kg/kmol}\left(0.0339 + \sum \Delta_p\right)^{-2} \tag{A.1-8b}$$

$$\frac{V_c}{cm^3/mol} = 40 + \sum \Delta_V \tag{A.1-8c}$$

The group contributions can be obtained from tables[**perry_perrys_2008**].

A.1.1.3 Normal boiling point

The normal boiling point T_b is the temperature at which the vapor pressure of a substance equals atmospheric pressure of 1.013 *bar*.

Methods With only a few vapor pressure data points available, T_b can be interpolated using equation (A.1-2). With more data points it can be calculated with vapor pressure equations (section A.1.1.4).

To estimate the normal boiling point of organic compounds without vapor pressure data the *Nannoolal* method can be used. The equation takes the structure of the molecule into account and the contribution groups are listed in tables[**perry_perrys_2008**].

$$\frac{T_b}{K} = \frac{\sum_{i=1}^{N} n_i \cdot C_i}{n^{0.6583} + 1.6868} \tag{A.1-9}$$

where:

n	=	number of hydrogen atoms
n_i	=	number of occurrences of group i
N	=	number of groups
C_i	=	group contribution

Alternatively, the *Joback* method can be used with equation

$$\frac{T_b}{K} = 198 + \sum \Delta_b \tag{A.1-10}$$

The group contributions Δ_b can be taken from tables[**gmehling_chemical_2012**].

A.1.1.4 Vapor pressure

A basic thermodynamic property of a liquid is its vapor pressure p^*, which is the pressure equilibrium of the coexisting liquid and gaseous phase at a given temperature. The vapor pressure curve connects the triple point (T_t, p_t), where all three phases exist, to the critical point (T_c, p_c), where no distinction between the liquid and gaseous phase is possible[**perry_perrys_2008**].

Methods Over a limited temperature range the *Antoine* equation

$$\log_{10} p^* = A - \frac{B}{T/K + C} \tag{A.1-11}$$

is a powerful tool to describe the correlation of temperature and liquid vapor pressure[**antoine_tensions_1888**]. The units of the parameters have to be considered. In literature also exists a version of the *Antoine* equation with the natural logarithm. To convert the parameters to fit this form, they have to be divided by ln(10). Generally the *Antoine* equation has very bad extrapolation performance[**vdi_e_v_vdi_2010**]. To correlate data from the critical point the *Wagner* equation

$$\ln p^* = \frac{a\tau + b\tau^{1.5} + c\tau^{2.5} + d\tau^5}{1-\tau} \qquad \text{where} \quad \tau \equiv 1 - T_r \tag{A.1-12}$$

or the *Riedel* equation

$$\ln \frac{p^*}{Pa} = A + \frac{B}{T/K} + C \ln \frac{T}{K} + D \left(\frac{T}{K}\right)^E \tag{A.1-13}$$

are applied[**perry_perrys_2008**].

A.1.1.5 Heat capacity

The energy required to heat a certain amount of substance by one degree Kelvin is defined as its heat capacity C. Normalized to its mass or its moles it is the specific or the molar heat capacity, respectively[**perry_perrys_2008**]. For most applications the partial derivative of enthalpy with respect to temperature

$$c_p = \left(\frac{\partial H}{\partial T}\right)_p \tag{A.1-14}$$

is used. The heat capacity c_p is defined at constant pressure and typically for 1 mole[**vdi_e_v_vdi_2010**].

Methods For gases the temperature dependence can be related to the ideal gas with the equation

$$c_p = c_p^{\ominus} - T \int_0^p \left(\frac{\partial^2 V}{\partial T^2}\right)_p dp \tag{A.1-15}$$

where the second term can be neglected for low or moderate pressure. To correlate heat capacity to temperature the *Aly-Lee* equation

$$c_p = A_0 + A_1 \left[\frac{A_2/T}{\sinh(A_2/T)}\right] + A_3 \left[\frac{A_4/T}{\cosh(A_4/T)}\right]^2 \tag{A.1-16}$$

or a polynomial equation (4^{th} order)

$$c_p = \sum_{i=0}^{4} A_i T^i \tag{A.1-17}$$

can be used. Good results are also achieved with the PPDS equation

$$\frac{c_p}{R} = B + (C-B)\left(\frac{T}{A+T}\right)^2 \left[1 - \frac{A}{A+T}\left(D + E\frac{T}{A+T} + F\left(\frac{T}{A+T}\right)^2 + G\left(\frac{T}{A+T}\right)^2\right)\right] \tag{A.1-18}$$

Computational chemistry by means of vibrational frequencies can also deliver accurate predictions of heat capacities[**vdi_e_v_vdi_2010**].

Another often deployed technique to calculate the heat capacity at standard condition with GC is the *Benson* method. It is applicable to ideal gases of organic compounds. The group contributions are summed up at multiple temperature levels according to the equation

$$c_p^{\ominus} = \sum_{i=1}^{N} n_i \cdot (c_p^{\ominus})_i \tag{A.1-19}$$

where:
n_i = number of occurrences of group i
$(c_p^{\ominus})_i$ = individual group contribution
The group contributions can be obtained from tables[**perry_perrys_2008**]. To interpolate between the discrete temperature levels equation (A.1-16) or (A.1-17) can be employed.

For liquids the more appropriate GC method is *Ruzicka-Domalski*. To find the temperature coefficients A, B and D, the group contributions a_i, b_i and d_i are summed up and applied in a cubic equation as follows.

$$A = \sum_{i=1}^{N} n_i a_i \qquad B = \sum_{i=1}^{N} n_i b_i \qquad D = \sum_{i=1}^{N} n_i d_i \tag{A.1-20a}$$

$$\frac{c_p}{R} = A + B\left(\frac{T}{100\ K}\right) + D\left(\frac{T}{100\ K}\right)^2 \tag{A.1-20b}$$

The group contributions can be obtained from tables[**perry_perrys_2008**].

For organic solids (at 298.15 K) the modified *Kopp*'s rule can be used to estimate the c_p value with equation

$$\frac{c_p}{J/(mol \cdot K)} = \sum_{E=1}^{N} n_E \Delta_E \tag{A.1-21}$$

where:
N = number of different elements in the compound
n_i = number of occurrences of element E in the compound
Δ_E = contribution of element E
The group contributions are listed in tables[**perry_perrys_2008**].

A.1.1.6 Density

For equipment design the density of a substance is a physical value with considerable demands on accuracy. It is defined by the equation

$$\frac{\rho}{kg/m^3} = \frac{m/kg}{V/m^3} \tag{A.1-22}$$

Methods The most used technique to correlate the temperature dependence of the density is the *Rackett* equation

$$\frac{\rho}{kg/m^3} = \frac{A}{B^{1+(1-\frac{T/K}{C})^D}} \tag{A.1-23}$$

A more accurate correlation for liquids delivers the PPDS equation

$$\frac{\rho_{liquid}}{kg/m^3} = \frac{\rho_c}{kg/m^3} + \left[A\left(1-T_r\right)^{0.35} + B\left(1-T_r\right)^{2/3} + C\left(1-T_r\right) + D\left(1-T_r\right)^{4/3}\right] \tag{A.1-24}$$

The critical density ρ_c can be obtained from equation (A.1-7d) via the *Joback* method[**vdi_e_v_vdi_2010**].

A.1.1.7 Thermal properties

The enthalpy change upon formation of 1 mole compound from its elements in the standard state (298.15 K and 1.013 bar) is defined as its enthalpy or heat of formation $\Delta H_f^{\ominus}$[**baehr_thermodynamik_2012**]. The temperature dependence can be written as

$$\Delta H_f(T) = \Delta H_f(T^{\ominus}) + \int_0^T c_p^{\ominus}(T)dt \tag{A.1-25}$$

The temperature and pressure dependence of the absolute entropy of a substance (typically 1 mole) in the standard state $S^{\ominus}$ for ideal gases can be calculated with equation

$$S(T,p) = S^{\ominus} + \int_{T^{\ominus}}^{T} c_p^{\ominus}(T)\frac{dT}{T} - R\,\ln\frac{p}{p^{\ominus}} \tag{A.1-26}$$

Methods A GC method for both enthalpy of formation and entropy for gases, liquids and solids (organic) is the *Domalski-Hearing* method. The group contributions $(\Delta H_f^{\ominus})_i$ and $(\Delta S^{\ominus})_i$ from tables[**perry_perrys_2008**] are implemented according to the equations

$$\frac{\Delta H_f^{\ominus}}{kJ/mol} = \sum_{i=0}^{N} n_i(\Delta H_f^{\ominus})_i \tag{A.1-27a}$$

and

$$\frac{S^{\ominus}}{J\cdot mol^{-1}K^{-1}} = \sum_{i=1}^{N} n_i(S^{\ominus})_i \tag{A.1-27b}$$

A computational way to predict thermal properties are isodesmic reactions. They are discerned as reactions in which the type and number of bounds are preserved during conversion. These *ab initio* calculations are not as accurate as GC methods for absolute enthalpies. On the other hand, for relative enthalpies (e.g. enthalpy of reaction) they are precise, provided the basis sets are accurate[**perry_perrys_2008**]. After formulating the isodesmic reaction for the formation of the wanted substance from compounds with known enthalpies, the enthalpy of reaction $\Delta_R H$ is calculated from

$$\Delta_R H = \sum \nu_i H_i \tag{A.1-28}$$

where ν_i is the stoichiometric coefficient and H_i the absolute enthalpy of i. The same result can be obtained with the enthalpies of formation $(\Delta H_f^{\ominus})_i$ according to

$$\Delta_R H = \sum \nu_i (\Delta H_f^{\ominus})_i \tag{A.1-29}$$

If the vibrational frequencies of a compound are available, its entropy can be calculated via statistical mechanics[**perry_perrys_2008**].

A.1.1.8 Latent enthalpy

The enthalpy or heat of vaporization ΔH_v is the enthalpy difference of the liquid and gaseous state of a substance at any point on the vapor pressure curve between the triple point and the critical point[**perry_perrys_2008**]. The link to the vapor pressure of a pure substance p^* is given by

$$\Delta H_v = -R\Delta Z_v \frac{d\ln p^*}{d(1/T)} = RT^2 \Delta Z_v \frac{d\ln p^*}{dT} \tag{A.1-30}$$

derived from the thermodynamic exact *Clausius-Clapeyron* equation

$$\Delta H_v = T\Delta V_v \frac{dp*}{dT} \tag{A.1-31}$$

The change of volume whilst vaporization can be calculated with the density change from liquid to gas. Most available data was not measured but correlated with this equation[**vdi_e_v_vdi_2010**]. If only one reference data point is available, the *Wagner* equation

$$\Delta H_v = \Delta H_{v,ref} \left(\frac{1 - T_r}{1 - T_{r,ref}} \right)^{0.38} \tag{A.1-32}$$

can be used. To correlate numerous data points the extended *Watson* equation

$$\Delta H_v = A(1 - T_r)^{B + CT_r + DT_r^2 + ET_r^3} \tag{A.1-33}$$

or the PPDS equation

$$\Delta H_v = RT_c(A\tau^{1/3} + B\tau^{2/3} + C\tau + D\tau^2 + E\tau^6) \qquad \text{where} \quad \tau \equiv 1 - T_r \tag{A.1-34}$$

are applied[**perry_perrys_2008**],[**vdi_e_v_vdi_2010**].

A nameless CS method to predict the enthalpy of vaporization from the acentric factor ω as a function of temperature applicable for organic compounds is the following equation:

$$\Delta H_v = RT_c\left(7.08\tau^{0.354} + 10.95\omega\tau^{0.456}\right) \tag{A.1-35}$$

where $\tau = 1 - T_r$.

Another CS method to estimate the enthalpy of vaporization at the normal boiling point is the *Vetere* method. ΔH_v is correlated to the critical properties in the following equation:

$$\frac{\Delta H_v}{RT_b} = \frac{\tau_b^{0.38}\left[\ln(p_c/bar) - 0.5313 + \frac{0.5066\ bar}{p_c T_{br}^2}\right]}{\tau_b + F(1-\tau_b^{0.38})\ln T_{br}} \tag{A.1-36}$$

where $\tau_b = 1 - T_{br}$ and $T_{br} = T_b/T_c$.

Also for the phase change from solid to liquid the enthalpy change is defined as the enthalpy or heat of fusion ΔH_{fus}. The *Chikos* method can be used to predict ΔH_{fus} with considerable uncertainty. It is calculated according to

$$\frac{\Delta H_{fus}}{J/mol} = \frac{\Delta S_{fus}}{J/(mol\cdot K)}\left(\frac{T_m}{K}\right) = (T_m/K)(a+b) \tag{A.1-37a}$$

$$a = \begin{cases} 0 & \text{no nonaromatic rings} \\ 35.19N_R + 4.289(N_{CR} - N_r) & \text{nonaromatic rings} \end{cases} \tag{A.1-37b}$$

$$b = \sum_{i=1}^{n_g} Ng_i\Delta s_i + \sum_{j=1}^{n_s} Ns_j Cs_j \Delta s_j + \sum_{k=1}^{n_f} Nf_k Ct_k \Delta s_k \tag{A.1-37c}$$

where:

Ng_i	=	number of C-H groups of type i bonded to other carbon atoms
n_g	=	number of different nonring or aromatic C-H groups bounded to other carbon atoms
Ns_j	=	number of C-H groups of type j bonded to at least one functional group or atom
n_s	=	number of different nonring or aromatic C-H groups bonded to at least one functional group or atom
Nf_k	=	number of functional groups of type k
n_f	=	number of different functional groups or atoms
t	=	total number of functional groups or atoms with the exception that F atoms count as one regardless of number of occurrences
N_R	=	number of nonaromatic rings
N_{CR}	=	number of -CH_2- groups in nonaromatic ring(s)

The group contributions Cs_j, Ct_k, Δs_i, Δs_j and Δs_k are obtained from tables[**perry_perrys_2008**].

A.1.2 Reaction kinetics

A.1.2.1 General

In the investigation of reaction kinetics, the velocity of chemical reactions is described and characterized mathematically. Depending on the reaction, this is regulated by concentrations, pressure, temperature and the presence of catalysts.

To perform a kinetic analysis, the stoichiometry of a reaction and possible side reactions must be known. The concentrations of reactants and products are measured at different times. The measurement can be carried out manometrically (for gases produced), photometrically, conductometrically or chromatographically. Real-time analysis and controlled initiation of the reaction play an important role here. If a reaction allows it, a quenching method can also be used. In this case, the reaction is stopped and the measurement is carried out without haste[**atkins_physikalische_2012**].

The speed of the reaction is usually proportional to the concentrations of the reactants to certain powers. Thus the rate law of a reaction will often be calculated according to the equation

$$r_r = k(T)[A]^n[B]^m \tag{A.1-38}$$

r_r stands for the reaction rate and $k(T)$ for the rate constant. The latter does not depend on the concentrations of the reaction partners, but it depends on the temperature.

The powers with which the concentrations enter the rate law determines the order of the reaction. If n and m correspond to 1, the reaction is of the 2nd order. The total order of a reaction is determined by the sum of the powers of all reaction partners. This value is not always an integer, as is the case with many gas phase reactions. A special case are also the zero-order reactions, whose speed depends only on k. These are mostly catalytic or enzymatic processes.

The reaction order and the rate law cannot be derived from the reaction equation. One method of determining them is the isolation method. One reactant after the other is offered in huge excess, while the others are hardly available. From the measurements of the rates of change in concentration, the powers can be determined. With an excess of substance B (approximately invariable) one thus obtains a rate law of pseudo-first order:

$$r_r = k'[A] \qquad \text{with} \qquad k' = k[B]_0 \tag{A.1-39}$$

Another method often used in conjunction with the insulation method is the initial velocity method. When considering an n-th order reaction (reduced to one reactant) the initial rates r_0 are measured at different concentrations of substance A. If the logarithms of both quantities are plotted against each other, a straight line with the slope n is obtained[**atkins_physikalische_2012**].

To understand the kinetics of a process, the integrated velocity laws often help. As the previous laws are differential equations, they can be integrated and one obtains with initial values the concentration as a function of time. The differences between the differential rate law

$$\frac{d[A]}{dt} = -k[A] \tag{A.1-40}$$

and the integral rate law

$$[A] = [A]_0 e^{-kt} \tag{A.1-41}$$

are shown in figure 66.

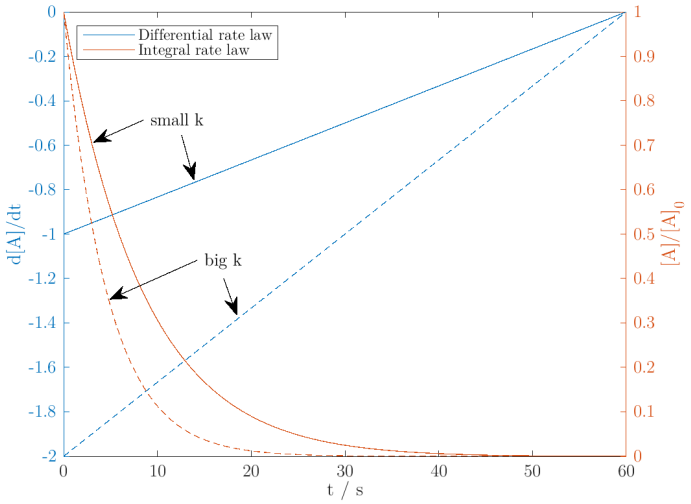


Figure 66: Visualization of differential and integral rate law

A.1.2.2 The Arrhenius parameters

Most reactions run faster at higher temperatures. To describe this relationship, the frequency factor (or pre-exponential factor) A and the activation energy E_A are needed. During the reaction along the reaction coordinate, the potential energy increases to a maximum by changing the distances and binding angles of the molecules. The activation energy is the energy needed to cross this hill and then fall into the thermodynamically more stable valley.

$$\ln k = \ln A - \frac{E_A}{RT} \qquad \Leftrightarrow \qquad k(T) = Ae^{\left(-\frac{E_A}{RT}\right)} \tag{A.1-42}$$

From this equation it can be derived that the temperature dependence of a reaction increases proportionally to its activation energy.

To model the activation energy, equation A.1-43 can be established by transformations and the introduction of a rate constant k_0 at a reference temperature T_0[**atkins_physikalische_2012**].

$$k(T) = k_0 e^{\left(\frac{E_A}{R\left(\frac{1}{T_0} - \frac{1}{T}\right)}\right)} \tag{A.1-43}$$

A.1.2.3 Kinetic modeling

To mathematically describe the reaction progression as a function of the time mainly two approaches are common. The model-free and the model-based kinetics (see figure 67). In this chapter the different strategies and their assumptions concerning the kinetics are described.

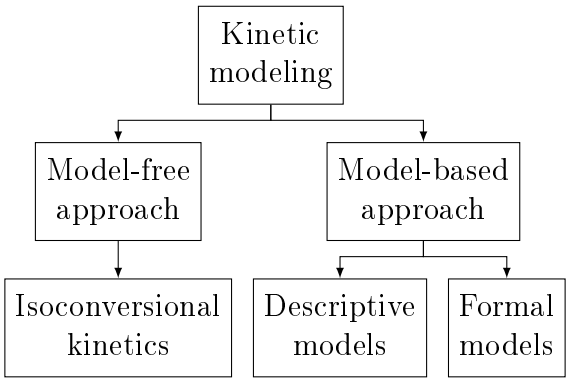


Figure 67: Hierarchy of the different approaches in kinetic modeling

Descriptive models A descriptive model is able to describe a reaction in detail, because it is formulated in terms of concentrations. The generalized law of mass action (GLMA) is assumed, i.e. the reaction rate is proportional to the product of the concentrations of the reaction partners raised to an order.

$$\frac{dc_i}{dt} = \sum_{r=1}^{N_r} \prod_{i=1}^{N_c} k_r c_i^{n_{r,i}} \tag{A.1-44}$$

The temperature dependence is taken into consideration with the *Arrhenius* parameters underlying k_r. The model is constrained by known initial conditions and the observable change rates of the specific heat q and gas production GP.

$$c_{i,0} \qquad q_0 \qquad \mathrm{GP}_0 \qquad \frac{dq}{dt} = \sum_{r=1}^{N_r} q_r r_r \qquad \frac{d\mathrm{GP}}{dt} = \sum_{r=1}^{N_r} q_r \Gamma_r \tag{A.1-45}$$

where by definition $q_0 = 0\ {}^{kJ}/_{mol}$ and $\mathrm{GP}_0 = 0\ {}^{L}/_{mol}$.

The models can be applied for batch, semi-batch and continuous stirred tank reactor (CSTR) with any desired temperature program. To use descriptive kinetics, the reaction and its participants have to be known well. Calorimetric and concentration data must be collected and the heat capacities as functions of temperature of each reaction participant is required. Once the model is complete, almost any desired prediction (e.g. TMR_{ad}, risk assessment, reactivity ratings) can be made[**kossoy_identification_2007**].

The above discussed rate law (A.1-38) describes an irreversible reaction of n-th order. However there are plenty other possibilities to describe the behavior of reactions. Some of the most used concepts are listed in table 13[**muller-erlwein_chemische_1998**].

Table 13: Typical rate laws for descriptive kinetic modeling[**muller-erlwein_chemische_1998**]

Name	**Reaction equation**	**Rate law r_r**
Irreversible reaction of n-th order	$A_1 \longrightarrow A_2\,(+...)$	$r = kc_1^n$
Isomerization equilibrium	$A_1 \longleftrightarrow A_2$	$r = k_1c_1 - k_{-1}c_2$
Irreversible bimolecular reaction	$A_1 + A_2 \longrightarrow A_3\,(+...)$	$r = kc_1c_2$
Autocatalytic reaction	$A_1 + A_2 \longrightarrow 2\,A_1 + A_3$	$r = kc_1c_2$
Consecutive reaction	$A_1 \longrightarrow A_2$	$r_1 = k_1c_1$
	$A_2 \longrightarrow A_3$	$r_2 = k_2c_2$
Parallel reaction	$A_1 \longrightarrow A_2$	$r_1 = k_1c_1$
	$A_1 \longrightarrow A_3$	$r_2 = k_2c_1$

Formal model The underlying assumption of a formal model is, that the state variables of a reaction is its degree of conversion α. This strategy is more useful when there is less information about the reaction participants. To construct such a model, only overall responses such as the heat or gas generation but not the compositions are required. For complex multi-stage reactions this approach is more suitable[**kossoy_identification_2007**].

The model consists of a set of ordinary differential equations

$$\frac{d\alpha_r}{dt} = k_r(T)f_r(\alpha_r) \qquad k_r(T) = A_{r,0}e^{-\frac{E_{A,r}}{RT}} \tag{A.1-46}$$

The model is supplemented by the initial conditions and the response quantity equations

$$\alpha_{r,0} \qquad q_0 \qquad \Gamma_0 \qquad \frac{dq}{dt} = \sum_{r=1}^{N_r} q_r r_r \qquad \frac{d\Gamma}{dt} = \sum_{r=1}^{N_r} \Gamma_r r_r \tag{A.1-47}$$

where by definition $\alpha_0 = 0$, $q_0 = 0\ {}^{kJ}/_{mol}$ and $\Gamma_0 = 0\ {}^{L}/_{mol}$. $f(\alpha)$ describes the reaction mechanism without the necessity to know all the reaction participants.

Table 14: Typical rate laws for single-stage reaction formal kinetic modeling[**kossoy_identification_2007**][**khawam_complementary_2005**]

Name	**Rate law $f(\alpha)$**
Irreversible reaction of n-th order	1
	$(1-\alpha)$
	$(1-\alpha)^2$
	$(1-\alpha)^n$
Generalized autocatalytic reaction	$(1-\alpha)^{n_1}(\alpha^{n_2} + z)$

A single-stage reaction A $\longrightarrow$ B is therefore characterized by

$$\frac{d\alpha}{dt} = k_0e^{-\frac{E_A}{RT}}f(\alpha) \tag{A.1-48}$$

Consecutive reactions with two stages of the form A $\longrightarrow$ B $\longrightarrow$ C can be modeled with equations

$$\frac{d\alpha_1}{dt} = k_{1,0}e^{-\frac{E_{A,1}}{RT}}(1-\alpha_1)^{n_1} \qquad \frac{d\alpha_2}{dt} = k_{2,0}e^{-\frac{E_{A,2}}{RT}}(\alpha_1-\alpha_2)^{n_2} \tag{A.1-49}$$

Two parallel reaction, e.g. an initiation stage A $\longrightarrow$ B and an autocatalytic stage A + B $\longrightarrow$ 2 B can be modeled according to equations

$$\frac{d\alpha}{dt} = r_1(\alpha) + r_2(\alpha) \tag{A.1-50a}$$

$$r_1(\alpha) = k_1(T)(1-\alpha)^{n_1} \tag{A.1-50b}$$

$$r_2(\alpha) = k_2(T)\alpha^{n_2}(1-\alpha)^{n_3} \tag{A.1-50c}$$

This model is very useful to describe full autocatalysis (see also section A.1.2.4). In a simple, single-stage reaction the concentrations can be directly correlated to the reaction conversion α. However, this is not possible for the more complex multi-stage models. The parameters obtained for these models can not be corresponded to the ones from descriptive models.

Since formal models need less information of the system, they also have some shortcomings. Changes due to non-kinetic reasons (e.g. feed of a reactant, vaporization of a volatile component, etc.) can not be taken into account. This means only batch reactors and not semi-batch or continuous reactors can be modeled. The kinetic model is only valid for the investigated mixture composition. Either way, formal models are of great use in thermal analysis and hazard assessment and are more flexible for complex reaction mechanisms.

Model-free approach The conventional model-fitting approach to kinetic analysis typically aims to determine the kinetic triplet of reaction consisting of the frequency factor, the activation energy and the reaction order (A, E_A, n). It assumes those quantities constant throughout the whole reaction course. This might be too simplistic for some reactions[**zhou_model-free_2003**]. Furthermore, especially non-isothermal model-fitting applies complex models like the *Coats* and *Redfern* method.

$$\ln\frac{g(\alpha)}{T^2} = \ln\left(\frac{AR}{\beta E_A}\left[1-\left(\frac{2RT_{exp}}{E_A}\right)\right]\right) - \frac{E_A}{RT} \tag{A.1-51}$$

where $g(\alpha)$ is the integrated form of $f(\alpha)$, β is the heating rate and T_{exp} is the mean experimental temperature. It was shown that the equation violated basic assumptions of linear regression and therefore, all models would result in relatively good fits[**bolton_pharmaceutical_2010**]. Thus in some cases the modelistic approach is not the most suitable.

The isoconversional methods evaluate the E_A and A, k_0 respectively at each degree of reaction conversion α. This means, the activation energy and the rate constant change

over the reaction course, but are the same for the same conversion. The underlying equation of the strategy is

$$-\ln t = \ln\left(\frac{A}{g(\alpha)}\right) - \frac{E_A}{RT} \tag{A.1-52}$$

At each α and Arrhenius plot of $-\ln t$ vs. $1/T$ can be constructed and $\frac{E_A}{R}$ is obtained as the slope. Isoconversional kinetics is an increasingly popular method to calculate the most important risk assessment indicatiors (e.g. TMR_{ad}). However, to fully understand the kinetics of a reaction, the investigation of the kinetic triplet (A, E_A and reaction order) is inevitable[**khawam_complementary_2005**].

A.1.2.4 Autocatalytic reactions

A very hazardous type of reaction kinetics is the autocatalytic decomposition. The sudden violent heat evolution of those reactions is often deemed unpredictable, because it can appear after a long time without any perceived heat emission.

A chemical reaction is called autocatalytic, when one of the products of a first reaction step is a reactant of another reaction step. Autocatalysis does not imply any form of catalytic activity in the sense of reaction mechanism, it just means the reaction is self-accelerating. This behavior leads to a rapid increase of the reaction rate, which will be higher than at the beginning, in contrast to the reaction rate in a n-th order reaction that decays over time. This acceleration leads to maximum reaction rate and then decreases again as all reactant is conversed. In thermal analysis a bell-shaped heat release curve is observed. The time that it takes the sample to reach the maximum reaction rate after it reached the initial temperature is called induction time t_{ind}. In this time period the reaction catalyst is produced. The isothermal induction time is measured typically by isothermal DSC or DTA and is a exponential function of temperature. Therefore, it can be linearly plotted as natural logarithm against the inverse absolute temperature[**stoessel_thermal_2008**].

To model the reaction rate of autocatalytic reactions, several models are applicable. The most simple one is the *Prout-Tompkins* model, since it is based only on one reaction, and therefore only on one rate equation[**prout_thermal_1944**].

$$\mathrm{A} + \mathrm{B} \longrightarrow 2\,\mathrm{B} \tag{A.1-53a}$$

$$-r_A = \frac{-dc_A}{dt} = kc_Ac_B \tag{A.1-53b}$$

In this equation one can see that the rate increases as c_B increases until c_A decreases too much. As a function of the reaction conversion (formal model), the rate can be formulated according to equation

$$\frac{d\alpha}{dt} = \dot{\alpha} = k\alpha(1-\alpha) \qquad \Leftrightarrow \qquad \alpha = \frac{1}{1+e^{-kt}} \tag{A.1-53c}$$

Another approach based on an initiation reaction and autocatalytic reaction is the *Benito-Perez* model. So the reaction rate is dependent on the rate of the two

reactions[**mata-perez_kinetic_1987**].

$$\begin{aligned} \nu_1 \mathrm{A} &\xrightarrow{\mathrm{k_1}} \nu_1 \mathrm{B} \\ \nu_2 \mathrm{A} + \nu_2 \mathrm{B} &\xrightarrow{\mathrm{k_2}} (\nu_3 + 1)\,\mathrm{B} \end{aligned} \tag{A.1-54a}$$

$$-r_A = k_1 c_A^{n_1} + k_2 c_A^{n_2} c_B^{m_2} \tag{A.1-54b}$$

in a simplified form the values of ν_1, ν_2 and ν_3 are all 1. The eight parameters for this model are the two frequency factors, two activation energies, three reaction order exponents and the initial conversion. As a specific case (all reaction orders are 1) of the formal model (A.1-50), the equation becomes

$$\frac{d\alpha}{dt} = k_1'(1-\alpha) + k_2'\alpha(1-\alpha) \tag{A.1-54c}$$

where k_1' and k_2' are the rate constant that can not be directly applied to the descriptive model above (equation (A.1-54b)).

The *Benito-Perez* model is more versatile than the *Prout-Tompkins* model, since it also takes the initiation reaction into account. This is especially important for weak autocatalytic reaction where this preceding initiation reaction is notable. Systems with a low or practically zero initiation rate (strong autocatalytic) formally approximate the *Prout-Tompkins* mechanism[**stoessel_thermal_2008**].

With the introduction of the autocatalytic factor P, the *Berlin* model is a third possibility to model self-accelerating reactions. It is a formal model and therefore not based on a reaction mechanism. The reaction rate as a function of the reaction conversion is given by

$$\frac{d\alpha}{dt} = k(1 + P\alpha)(1-\alpha) \tag{A.1-55a}$$

The bigger P is, the more important the autocatalytic character becomes. When P and k are set to

$$P = \frac{k_2'}{k_1'} \qquad k = k_1' \tag{A.1-55b}$$

the model can be transformed into the *Benito-Perez* model[**stoessel_thermal_2008**].

A.1.3 Hydroxylamine

A.1.3.1 Synthesis

Raschig Proceedings For the synthesis of HA, the Raschig process (see reaction equations (A.1-56)) is usually used in industry. Accordingly, ammonia is reacted with water and CO_2 to ammonium carbonate and then with NO and NO_2 to ammonium nitrite. With SO_2 the latter reacts to ammonium hydroxylamine disulfonate, which yields HA-sulfate and ammonium sulfate by a two-step hydrolysis[**wiley-vch_verlag_gmbh_&_co._kgaa_hydroxylamine_2000**],[**moesdijk_cataly**

$$2\,\mathrm{NH_3} + \mathrm{CO_2} + \mathrm{H_2O} \longrightarrow (\mathrm{NH_4})_2\mathrm{CO_3} + (\mathrm{NH_4HCO_3}) \tag{A.1-56a}$$

$$NO + NO_2 + (NH_4)_2CO_3 \longrightarrow 2\,NH_4NO_2 + CO_2 \quad \text{(A.1-56b)}$$

$$NH_4NO_2 + NH_4OH + 2\,SO_2 \longrightarrow NOH(SO_3NH_4)_2 \quad \text{(A.1-56c)}$$

$$NOH(SO_3NH_4)_2 + H_2O \longrightarrow NHOH(SO_3NH_4) + NH_4HSO_4 \quad \text{(A.1-56d)}$$

$$NHOH(SO_3NH_4) + 2\,H_2O \longrightarrow (NH_3OH)_2SO_4 + (NH_4)_2SO_4 \quad \text{(A.1-56e)}$$

In this thesis, three other methods for the production of hydroxylamine are discussed, which have so far only been used on a laboratory scale, but which also have the potential to become large-scale processes.

Damp ammonia oxidation by DBD For the synthesis of hydrazine, *Besson* investigated the effect of dielectrical barrier discharges (DBD) on ammonia gas. He noticed that traces of moisture in ammonia caused the formation of HA. The gas was then deliberately moistened with steam and the product was introduced in hydrochloric acid. The resulting white solid had all the properties of hydroxylamine hydrochloride[**besson_action_1911**]. Hence, *Besson* formulated the reaction equation (A.1-57).

$$NH_3 + H_2O \longrightarrow 2\,H^{\bullet} + NH_2OH \quad \text{(A.1-57)}$$

Electrolysis of nitric acid Hydroxylamine is produced electrochemically either in a nitrogen monoxide hydrogen fuel cell or in a nitrate electrolysis cell[**tafel_elektrolytische_1902**],[**lewdorowicz_synthesis_nodate**],[**danilov_modified_2012**]. The first principle is a so-called electrogenerative reduction. For this purpose, hydrogen is oxidized at the Pt anode and nitrogen oxide is reduced at the Pt or Ru cathode. An electric current is generated in the process. The electrical potential is thus generated by the chemical reaction itself (see reaction equations (A.1-58)). The hydrogen ions migrate through a H^+-permeable membrane and the electrons through the conductor[**langer_electrogenerative_1980**]. After this one-electron process, the dimerization of HNO is followed by the formation of hydroxylamine with various by-products[**janssen_reduction_1977**].
The one-step anode reaction is as follows:

$$H_2 \longrightarrow 2\,H^+ + 2\,e^- \quad \text{(A.1-58a)}$$

For the cathode a three-step reaction equation can be written.

$$NO + H^+ + e^- \longrightarrow HNO \quad \text{(A.1-58b)}$$

$$2\,HNO + e^- \longrightarrow H_2N_2O_2 \quad \text{(A.1-58c)}$$

$$H_2N_2O_2 + H^+ + e^- \longrightarrow NH_2OH + (N_2 + N_2O + N_2H_4 + NH_3) \quad \text{(A.1-58d)}$$

Since the handling of NO and H_2 gas is relatively dangerous, this paper dealt with the less explosive method of electrolysis of nitric acid. In previous experiments platinum was used as the anode and mercury as the cathode. The anolyte is located together with the

anode in a clay cell which serves as a diaphragm (see figure 68). The cathode is located in liquid form at the bottom of the cell and has been cooled with ice. The electrolyte used was 50 % sulfuric acid. Nitric acid was continuously added to the cathode. It turned out that the hydroxylamine yield is maximal at a nitric acid concentration of about 1 $^{mol}/_{L}$. At higher concentrations the hydroxylamine decomposes to nitrous oxide and other nitrogen oxides. The side reaction at the cathode is the reduction of water to hydrogen. Furthermore, nitric acid deficiency leads to the formation of an unknown nitrogen compound[**tafel_elektrolytische_1902**],[**stscherbakow_elektrolytische_1929**]. The electrode equations (A.1-59) can be derived from the available information.

Anode:

$$2\,H_2O \longrightarrow O_2 + 4\,H^+ + 4\,e^- \tag{A.1-59a}$$

Cathode:

$$HNO_3 + 2\,H^+ + 2\,e^- \longrightarrow NH_2OH + O_2 \tag{A.1-59b}$$

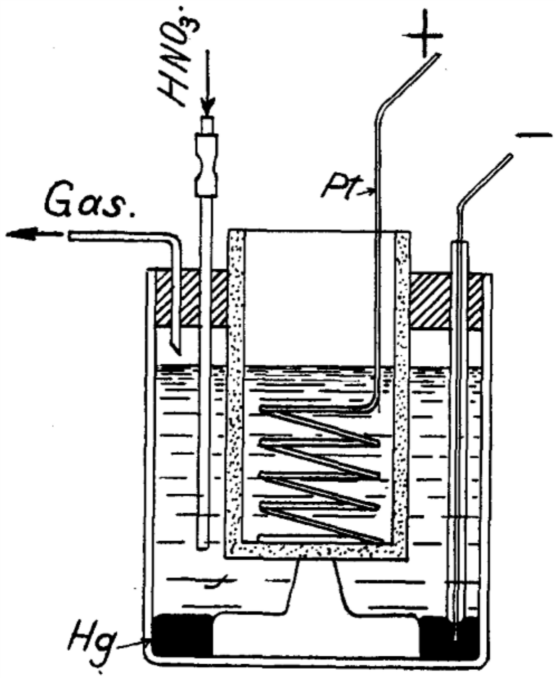


Figure 68: Electrolysis cell for hydroxylamine synthesis[**stscherbakow_elektrolytische_1929**]

Selective Oxidation of Ammonia At *FHNW* a term paper on the topic of hydroxylamine has already been written[**urso_herstellung_2015**]. A catalytic oxidation of ammonia is covered. The reaction can either be carried out directly together

with the oximation (then it is called ammoximation) or separately as a synthesis of HA[**urso_herstellung_2015**],[**dal_pozzo_ts-1_2002**].

$$NH_3 + H_2O_2 \xrightarrow{TS-1} NH_2OH + H_2O \tag{A.1-60}$$

Roffia et al. discovered in 1990 that, with the aid of the catalyst titanium-silicate-1, cyclohexanone can be selectively converted to cyclohexanone oxime[**roffia_cyclohexanone_1990**]. Further investigations revealed that the intermediate product of ammonia and hydrogen peroxide must be HA[**dal_pozzo_ts-1_2002**]. See also chapter 2.1.3.

A.1.3.2 Analytics

Acidimetric methods Due to ability of hydroxylamine to react with reducing as well as oxidizing agents, there is a wide range of redox titrations that can be carried out. However, the reactions are often unspecific. Acidimetric titrations are also possible because hydroxylamine is a weak base[**leithe_elemente_1957**]. The method of the project partner is a titration with 0.1 molar hydrochloric acid. Ammonia and hydroxylamine can be determined simultaneously (see table 15).

Table 15: Information for the acidimetric determination of HA

Substance	Potential jump	Working range
HA	pH 1 to 6	0.1 to 55 wt%
Ammonia	pH 6 to 11	0.1 to 25 wt%

Photometric Methods Colorimetric methods are also described in the literature. In the work of *A. Urso* two photometric methods are covered[**urso_herstellung_2015**]. The first procedure, which was also carried out practically, is from *Pucher* and *Day*. It takes a few minutes[**pucher_colorometric_1926**]. The second procedure of *Berg* and *Becker* takes more than an hour, but is more sensitive[**berg_neuer_1940**].

The procedure for photometry according to *Pucher* and *Day* is as follows:
4 mL of a sample containing 0.8 to 3.5 mg hydroxylamine are mixed with 2 drops of colourless benzoyl chloride, 4 mL ethanol and 2 mL sodium acetate solution (2 % in H_2O). The sample is shaken for 20 to 30 s and left to stand for 2 to 3 minutes. Add 2 mL acidic ferric chloride solution (0.5 g ferric chloride-6-hydrate + 2 mL conc. HCl in 100 mL H_2O) and dilute to 25 or 50 mL. Leave to stand for a few minutes so that the excess benzoyl chloride settles to the bottom. The violet sample can then be determined photometrically. The original sample should be neutral or slightly acidic. The accuracy is ± 2 to 6%[**leithe_elemente_1957**].

In order to find the suitable method of photometry according to *Berg* and *Becker*, a preliminary test must first be carried out:
5 mL of the neutral and unbuffered hydroxylammonium salt sample solution are mixed with 3 mL 8-oxy-quinoline solution (0.4 % in 50 % ethanol) and 3 mL buffer solution

(glycocoll sodium hydroxide solution pH 10-11.5), shaken thoroughly and measured after 10 min with a red filter. The result is evaluated according to the table 16.

Table 16: Preliminary test of the photometric determination of hydroxylamine according to *Berg* and *Becker*

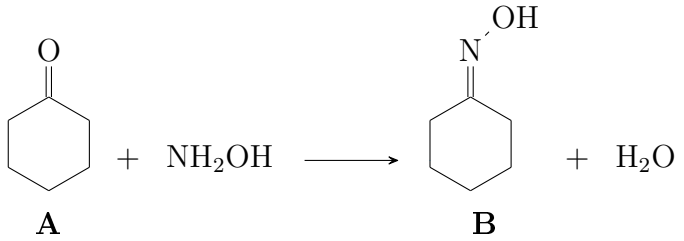
Absorbance	HA concentration	Procedure
0.4 to 0.17	20 to 100 $^{\mu g}/_{mL} mL$	1
0.11 to 0.03	2 to 10 $^{\mu g}/_{mL}$	2
<0.03	<2 $^{\mu g}/_{mL}$	3

Procedure 1 (20 to 100 $^{\mu g}/_{mL}$): As the absorbances in this concentration range change rapidly over time, the reaction is interrupted after 1 h by careful neutralization with acetic acid solution. 5 mL sample solution are mixed with 2 mL reagent solution and 3 mL buffer solution and shaken vigorously for 30 s. After standing for one hour, 0.6 mL of acetic acid (1.5 % in ethanol/H_2O 3:1) is added until the colour changes to red, filled up to 20 mL and measured using a green filter.

Procedure 2 (2 to 10 $^{\mu g}/_{mL}$): 5 mL of the sample solution is shaken vigorously in a 20 mL volumetric flask with 2 mL of 2 % 8-oxy-quinoline solution and 3 ml buffer solution for 30 s. Leave closed for 4 h, make up to 20 mL with 50 % alcohol and measure the absorbance.

Procedure 3 (0.2 to 1.0 $^{\mu g}/_{mL}$): In this low concentration range, enrichment of the reaction product by extraction with chloroform has proven to be useful. Add 2 mL 8-oxy-quinoline solution and 2 mL buffer solution to 10 mL sample solution and shake well; allow to react for 15 h. Then 3 mL chloroform is added to the bottom, 0.5 mL acetic acid solution is added and shaken well. The chloroform layer is separated and measured using the green filter[**leithe_elemente_1957**].

Gas chromatography To prove hydroxylamine formation during the selective oxidation of ammonia, *Mantegazza* and *Deniz* have made use of the oximation of cyclohexanone (**A**) (see equation (A.1-61)). The reaction proceeds overnight at RT with yields of up to 90 %. The resulting reaction mixture can be analysed in the GC and the hydroxylamine content of the original solution can be calculated stoichiometrically[**mantegazza_selective_1994**],[**deniz_effect_2013**].



(A.1-61)

A.1.4 DSC working principle

The differential scanning calorimetry is a thermal method that measures the difference in heat flux between a sample substance and a reference substance as a function of the temperature. Thereby the substances are submitted to any desired temperature program (e.g. isothermal, dynamic or modulated). The method is often applied in hazard assessment and the general investigation of thermal behavior of substances. It is also a very powerful tool in polymer characterization. In the pharmaceutical industry DSC is used for purity investigation over the melting behavior. Today, DSC is the most used thermal method[**skoog_instrumentelle_1996**].

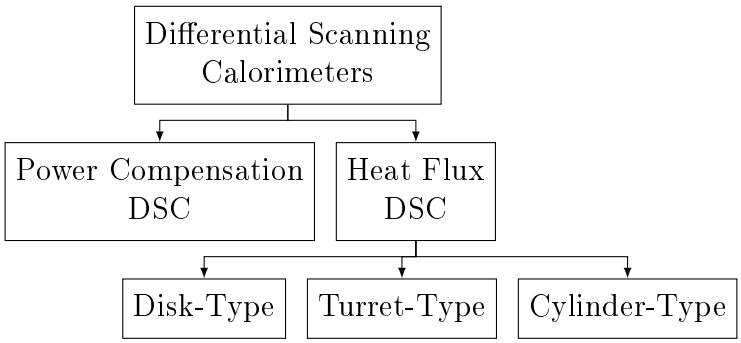


Figure 69: Overview of the existing types of differential scanning calorimeters

In power compensation methods, the sample and the reference are heated in separated ovens at equal rate or constant temperature. The difference in the required power to do so is the measuring signal.

Heat flux DSCs belong to the class of heat-exchange calorimeters. This means the heat flux over a well-defined heat conducting path with known thermal resistance is measured. The temperature difference ΔT between the sample T_S and the reference T_R is proportional to the heat flow rate.

$$\phi_{FS} - \phi_{FR} \propto -\Delta T \qquad \text{with} \quad \Delta T = T_S - T_R \tag{A.1-62a}$$

$$\phi_m = -k'\Delta T \tag{A.1-62b}$$

where k' is a factory-installed provisional calibration.

Commercial DSCs realize this measuring principle with numerous designs, including the disk-type, the turret-type and the cylinder-type calorimeters. The DSC820 from *Mettler* used in this work is of the disk-type[**hohne_differential_2010**]. The general construction is shown in figure 70.

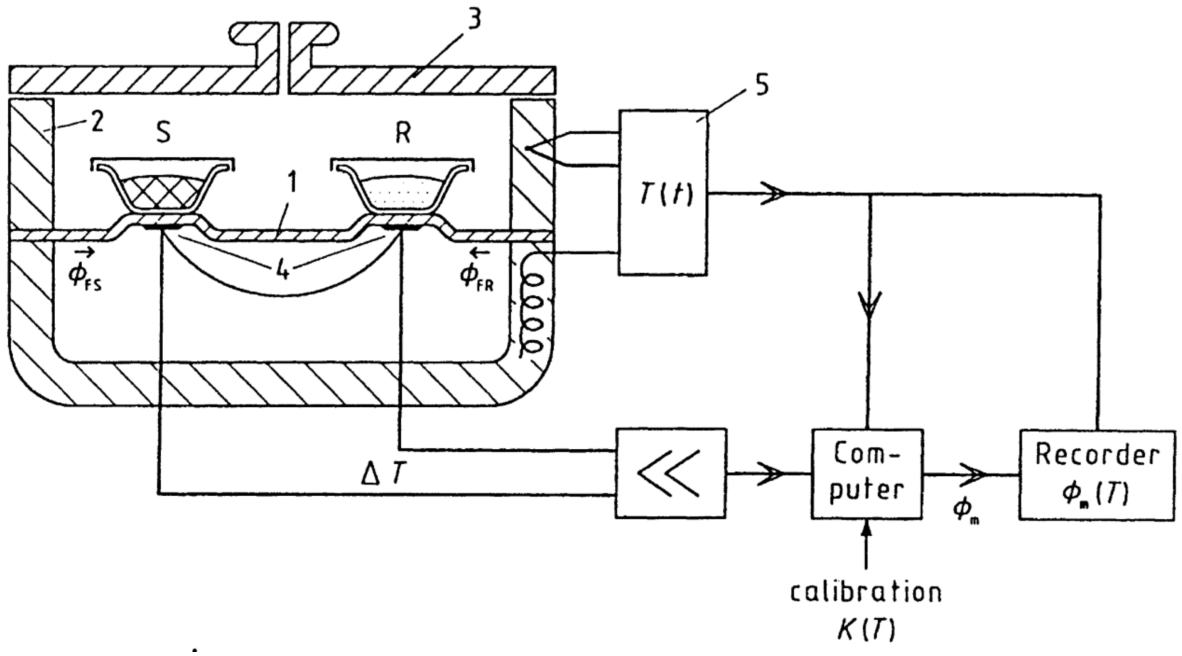


Figure 70: Schematic construction of a disk-type heat flux DSC. 1 disk, 2 furnace, 3 lid, 4 differential thermocouple(s), 5 programmer and controller, S crucible with sample substance, R crucible with reference sample substance, ϕ_{FS} heat flow rate from furnace to sample crucible, ϕ_{FR} heat flow rate from furnace to reference sample crucible, ϕ_m measured heat flow rate, $K(T)$ calibration factor[**hohne_differential_2010**].

An important correction quantity in the DSC measurement method is the so called phi factor or thermal inertia φ. It is defined by the equation

$$\varphi = \frac{C_S + C_C}{C_S} \tag{A.1-63}$$

where C_S and C_C are the heat capacities of the sample and the test cell, respectively. Since the heat capacity of the reaction vessel (more important in experimental settings than in industrial-scale) serves to lower the measured temperature of the reaction in the case of adiabatic conditions, the measured signal deviates from the real value. The phi factor can be utilized to correct the signal. The measured ΔT_{ad} and the TMR_{ad} measured must be corrected for the effect of phi [**wilcock_review_1997**].

$$\Delta T_{ad,actual} = \Delta T_{ad,measured}\varphi \tag{A.1-64}$$

$$t_{actual} = \frac{t_{measured}}{\varphi} \tag{A.1-65}$$

A.1.5 Thermocouples

To measure the temperature of the gas stream exiting the cooling trap a 0.5 mm thermocouple was used. The measuring principle of a thermocouple is based on the thermoelectric

effect. This effect was found in 1822 by *Seebeck* and is observed when two different metals touch. A thermal voltage is produced, due to the different binding energies of the electrons to the metal ions. The degree of this voltage depends on the metal and on the temperature of the contact point. To obtain a measurable current, the metal have to be connected at a second point. At this location, there is a thermal voltage as well. No current is observed, when the two junctions have the same temperature. The temperature at the sensing junction (ϑ_1) can only be measured, if the temperature at the reference junction (ϑ_2) is known. It is usually measured in a more moderate temperature range (cold-junction measurement).

$$U_{measured} = U(T_{measured}) - U(T_{ref}) = U(\vartheta_1) - U(\vartheta_2) \tag{A.1-66}$$

Originally, the reference junction was held into an ice bath and was thus fixed at 0 °C. Today, thermal resistors are used to measure the temperature ϑ_2. To inhibit additional junctions with thermoelectric effects, the wire connecting the measuring junction with the reference junction should be of the same materials as the junctions themselves (compensation line)[**weber_elektrische_1997**].

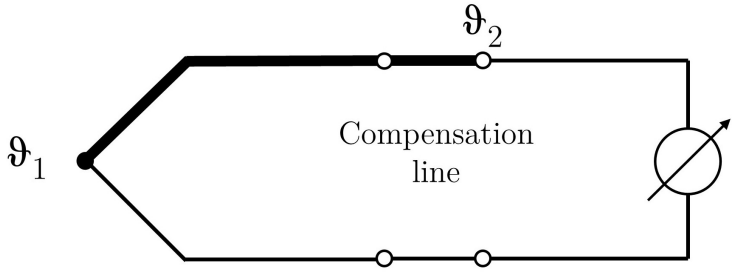


Figure 71: Circuit diagram of a conventional thermocouple

For this work thermocouples of the type K were used. They are made up from a Ni-Cr alloy connected to Ni-Al alloy[**damarla_classification_2011**]. The measuring temperature range lies between -270 °C and 1372 °C.

A.2 Thermodynamic equations for the estimation calculations

Description of the system and the mixture The situation in the explosion reactor is best described as a closed system with an aqueous liquid phase and a gas phase (see figure 72. Therefore, its total volume consists of the liquid and the gaseous volume.

$$V = V^{(l)} + V^{(g)} \tag{A.2-1}$$

The liquid phase is assumed incompressible and its volume constant. Therefore, its density remains 1120 $^{kg}/_{m^3}$, which is the density of a 50 % hydroxylamine solution. An equilibrium of the temperature and the pressure is assumed as well.

$$p^{(g)} = p^{(l)} \qquad T^{(g)} = T^{(l)} \tag{A.2-2}$$

According to the above mentioned decomposition reaction equations, the substances in table 17 could be present in the system.

Table 17: Substances present in the explosion reactor

i	**Formula**	**Name**
1	NH_2OH	hydroxylamine
2	NH_3	ammonia
3	N_2	nitrogen
4	N_2O	nitrous oxide
5	NO	nitric oxide
6	H_2O	water
7	H_2	hydrogen

The number of moles of component i in each phase is presented by equation

$$n_i = n_i^{(l)} + n_i^{(g)} \tag{A.2-3}$$

The composition of the phases can be calculated by the mole fractions

$$x_i = \frac{n_i^{(l)}}{\sum n_i^{(l)}} \qquad \text{and} \qquad y_i = \frac{n_i^{(g)}}{\sum n_i^{(g)}} \tag{A.2-4}$$

The introduction of a *partition coefficient* P allows to calculate the ratio of number of moles in the gas and the liquid phase according to

$$P_i = \frac{n_i^{(g)}}{n_i^{(l)}} \tag{A.2-5}$$

For optimization rather than P, the fraction of number of moles in the gaseous phase with respect to the total number of moles was used. This way the modified partition coefficient $P^{(g)}$ only takes up values from 0 to 1. It is calculated with equation

$$P_i^{(g)} = \frac{n_i^{(g)}}{n_i^{(g)} + n_i^{(l)}} = \frac{n_i^{(g)}}{n_i} = \frac{P_i}{P_i + 1} \tag{A.2-6}$$

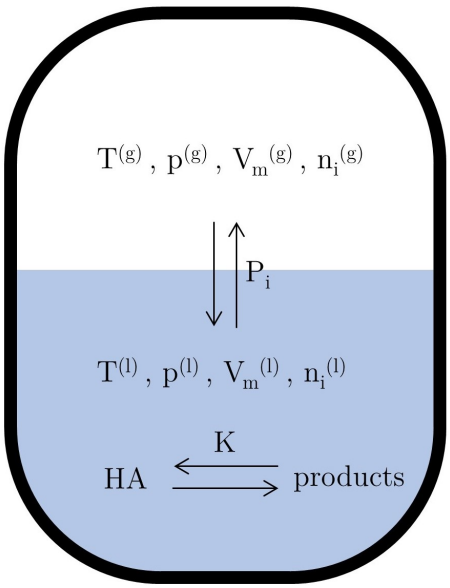


Figure 72: Scheme of the physical properties and relevant equilibria in the EXR

Thermal properties of the components The gaseous phase was assumed ideal. The liquid phase was assumed an ideal mixture of hydroxylamine and water with dissolved gases. To calculate the chemical equilibrium only the liquid phase was involved, as the gas phase contains less than only a few nanomoles of hydroxylamine. The gas solubility was described by *Henry's law* according to equation

$$p_i(T) = H_{i,H_2O}(T)x_i \tag{A.2-7}$$

Sander listed the reciprocal $H^{cp}_{i,H_2O}(T^{\ominus})$ of *Henry's law* constants at 298.15 K that connects the concentration of a compound i to the vapor pressure and their temperature dependencies $\frac{d\ln H^{cp}_{i,H_2O}}{d(1/T)}$[**sander_compilation_2015**]. One can calculate the value with following equation:

$$p_i = x_i \frac{\rho(H_2O)}{mw(H_2O)} / \left[H^{cp}_{i,H_2O}(T^{\ominus}) e^{\frac{d\ln H^{cp}_{i,H_2O}}{d(1/T)}\left(\frac{1}{T} - \frac{1}{T^{\ominus}}\right)} \right] \tag{A.2-8}$$

Plots of the vapor pressures of the present gases as a function of temperature and mole ration can be found in appendix A.4.

The vapor pressures of hydroxylamine and water were calculated with *Antoine* equation (A.1-11) and *Raoult's law*

$$p_i = p_i^* x_i \tag{A.2-9}$$

It was found that the pressure of the gases with very low solubility in water (high *Henry* coefficient), namely nitrogen, nitric oxide and hydrogen, can be calculated more accurately with the ideal gas equation

$$p_i = \frac{n_i RT}{V^{(g)}} \tag{A.2-10}$$

A thermodynamic system can either be described with enthalpy H or internal energy U. Enthalpy is more useful for systems with constant pressure, because the change of enthalpy is only dependent of temperature changes and not of volume changes, according to the equation

$$\left(\frac{\partial H}{\partial T}\right)_p = c_p(T)dT \tag{A.2-11}$$

Internal energy on the other hand is suitable for system changes with constant volume. The change of internal energy with constant volume is independent of pressure as seen in equation

$$\left(\frac{\partial U}{\partial T}\right)_V = c_V(T)dT \tag{A.2-12}$$

Therefore, internal energy is more convenient to calculate the thermal properties of the explosion reactor.

To correlate the heat capacity of each substance, apart from hydroxylamine, on the *NIST* database parameters for the *Shomate* equation

$$c_p = A + Bt + Ct^2 + Dt^3 + \frac{E}{t^2} \qquad \text{where} \quad t = \frac{T/K}{1000} \tag{A.2-13}$$

are available[**linstrom_nist_1997**]. For hydroxylamine, a 4-th order polynomial according to equation (A.1-17) was applied to calculated values of *Giguére and Liu*[**giguere_infrared_1952**]. The c_p value is converted to the c_V value according to equation

$$c_V = c_p - R \tag{A.2-14}$$

which represents the partial derivative of internal energy with respect to temperature from equation (A.2-12). The molar internal Energy at a given temperature and a reference volume can be calculated with equation

$$U(T, V_m^{\ominus}) = \Delta H_f^{\ominus} - RT^{\ominus} + \int_{T^{\ominus}}^{T} c_V(T)\ dT \tag{A.2-15}$$

The change of internal temperature is therefore the antiderivative of the *Shomate* equation (A.2-13) for c_V

$$\int c_V(T)\ dT = \left[AT + \frac{BT^2}{2*1000} + \frac{CT^3}{3*1000^2} + \frac{DT^4}{4*1000^3} - \frac{E*1000^2}{T} - RT\right]dT \tag{A.2-16}$$

The antiderivative of the 4-th order polynomial (A.1-17) for hydroxylamine is

$$\int c_V(T)\ dT = \left[A_0 T + \frac{A_1 T^2}{2} + \frac{A_2 T^3}{3} + \frac{A_3 T^4}{4} + \frac{A_4 T^5}{5} - RT\right] dT \tag{A.2-17}$$

To calculate the molar entropy at a given temperature and reference volume

$$S(T, V_m^{\ominus}) = S_f^{\ominus} + \int_{T^{\ominus}}^{T} \frac{c_V(T)}{T}\ dT \tag{A.2-18}$$

the integrated function of $\frac{c_V(T)}{T}$ is required.

$$\int \frac{c_V(T)}{T}\ dT = \left[A\ \ln T + \frac{BT}{1000} + \frac{CT^2}{2 * 1000^2} + \frac{DT^4}{3 * 1000^3} - \frac{E * 1000^2}{2 * T^2} - R\ \ln T\right] dT \tag{A.2-19}$$

and in case of hydroxylamine

$$\int \frac{c_V(T)}{T}\ dT = A_0 \ln T + A_1 T + \frac{A_2 T^2}{2} + \frac{A_3 T^3}{3} + \frac{A_4 T^4}{4} - R \ln T \tag{A.2-20}$$

To describe the chemical equilibrium, the *Helmholtz* free energy A can be calculated from internal energy and entropy according to equation

$$A(T, V_m) = U(T, V_m^{\ominus}) - TS(T, V_m^{\ominus}) - RT\ \ln \frac{V_m}{V_m^{\ominus}} \tag{A.2-21}$$

The last term is a correction for the volume. It can be modified as follows:

$$\frac{V_m}{V_m^{\ominus}} = \frac{nRT/p}{nRT/p^{\ominus}} = \frac{p^{\ominus}}{p} \tag{A.2-22}$$

In a mixture, the partial *Helmholtz* free energy of a substance i is its chemical potential

$$\mu_i = \left(\frac{\partial A}{\partial n_i}\right)_{T,V,n_{j\neq i}} \tag{A.2-23}$$

$$\mu_i(T, V_m) = A_i^*(T, V_m) + RT\ \ln x_i \tag{A.2-24}$$

To calculate the chemical potential of the gas components in the liquid equations (A.2-21), (A.2-22) and (A.2-24) can be combined to

$$\mu_i(T, V_m) = A_i^*(T, V_m^{\ominus}) + RT\ \ln \frac{p_i^*}{p^{\ominus}} + RT \ln x_i \tag{A.2-25}$$

or simplified with *Raoult's law* (A.2-9)

$$\mu_i(T, V_m) = A_i^*(T, V_m^{\ominus}) + RT\ \ln \frac{p_i}{p^{\ominus}} \tag{A.2-26}$$

The chemical potential of the ideal liquid mixtures components (hydroxylamine and water) can be described by

$$\mu_i(T, V_m) = A_i^*(T, V^{\ominus}) + RT \ \ln x_i \tag{A.2-27}$$

There is no need for a volume correction, as the liquid is assumed incompressible. To calculate the absolute *Helmholtz* free energy of the entire system, the equation

$$A_{sys}(T, V_m) = \sum_{i=1}^{N_C} \mu_i n_i \tag{A.2-28}$$

is used.

Thermodynamics Over the course of the reaction, the *Helmholtz* free energy changes according to equation

$$\Delta_R A = \left(\frac{\partial A}{\partial \xi}\right)_{T,V} \tag{A.2-29}$$

The change of the extent of reaction $d\xi$ is calculated with equation

$$d\xi = \frac{dn_i}{\nu_i} \tag{A.2-30}$$

As ξ equals 0 in the beginning of the reaction, it can be defined as

$$\xi_r = \frac{\Delta n_i}{\nu_{i,r}} \tag{A.2-31}$$

Because hydroxylamine is the limiting component in its decomposition, the extent of reaction with respect to hydroxylamine can be calculated for reaction r according to equation

$$\xi_r = \frac{n_{1,0}\alpha_r}{|\nu_{1,r}|} \tag{A.2-32}$$

The reaction conversion α_r is any number between 0 and 1. It describes the percentage of the reaction r that took place. According to equation (A.2-30), the number of moles of component i can be calculated for every reaction at a given extent of reaction with equation

$$n_i(\xi_r) = n_{i,0} + \nu_{i,r}\xi_r \tag{A.2-33}$$

The chemical equilibrium of reaction r is reached when the *Helmholtz* free energy is minimized. The following equation describes this state[**gmehling_chemical_2012**],[**atkins_physikalische_2012**]:

$$\sum_{i=1}^{N_C} \nu_{i,r}\mu_i = 0 \tag{A.2-34}$$

The equilibrium constant at constant volume is also linked to the *Helmholtz* free energy[**wiberg_inorganic_2001**], according to the equations

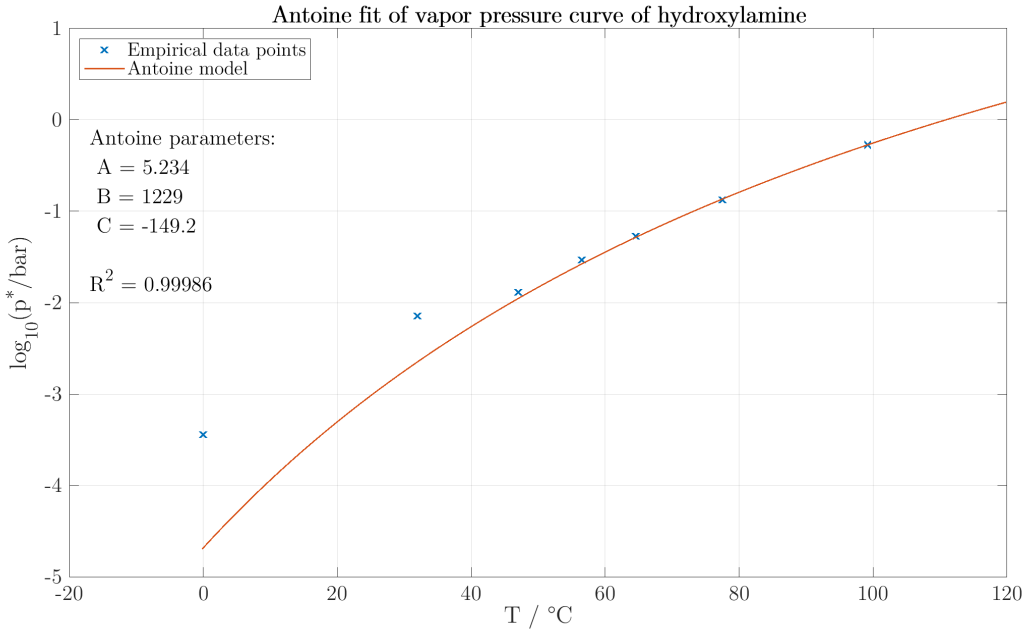
$$\Delta_R A(T, V_m) = \sum_{i=1}^{N_C} \nu_i A_i(T, V_m) \tag{A.2-35}$$

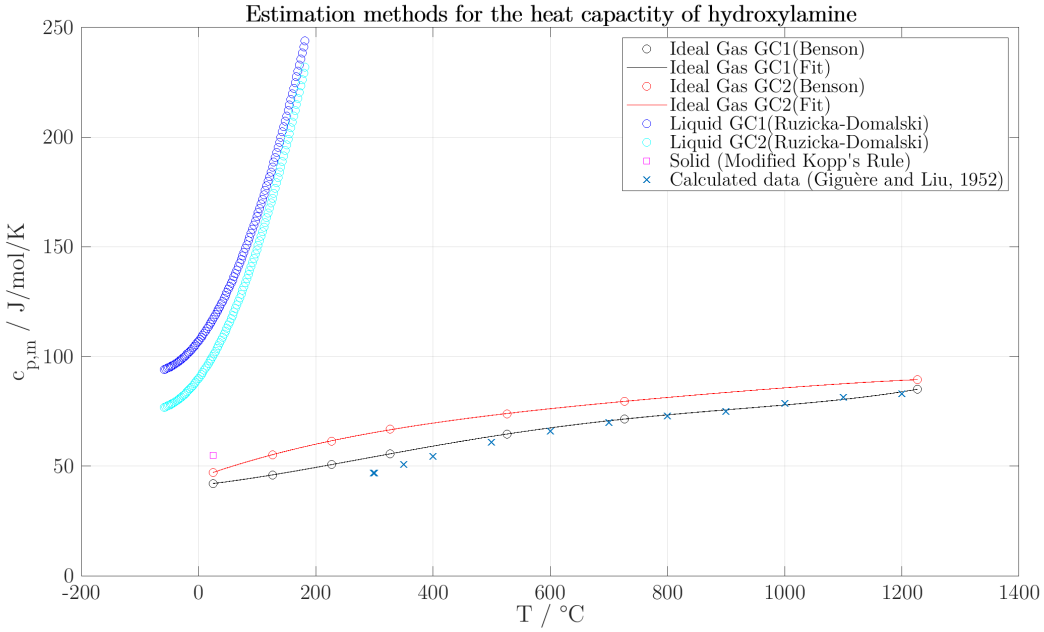
and

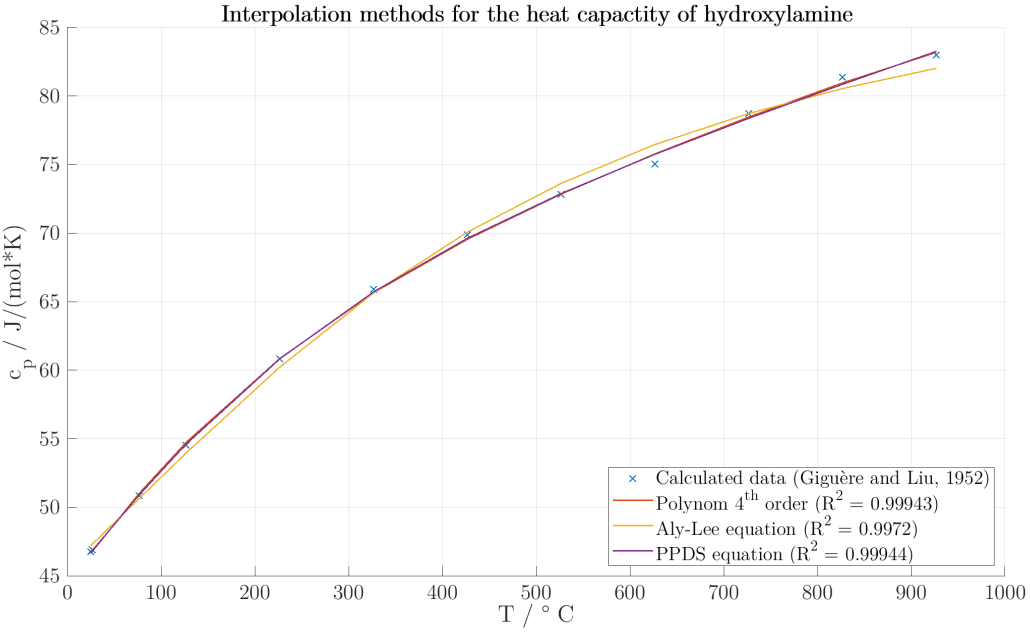
$$\Delta_R A(T, V_m) = -RT \ \ln K_V(T, V_m) \tag{A.2-36}$$

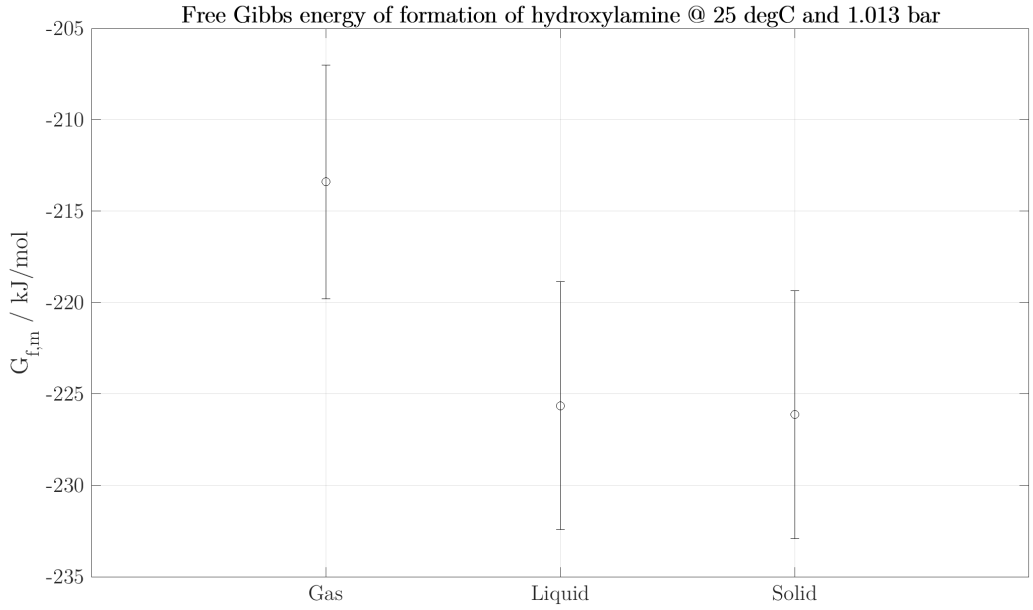
A.3 Estimation of the substance data of hydroxylamine

Plots comparing results and different methods from section A.1.1 (calculations see appendix A.14).



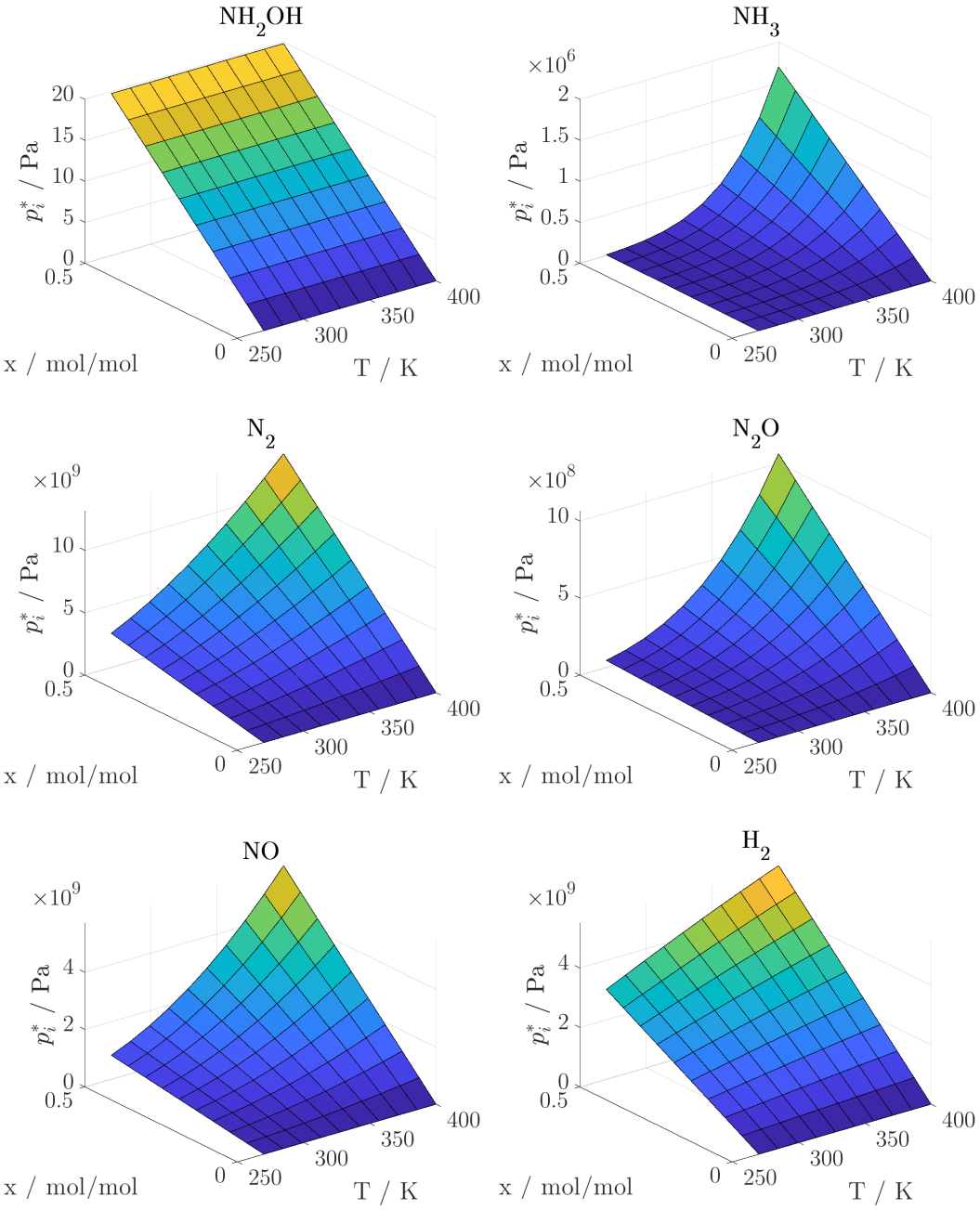



Interpolation methods for the heat capactity of hydroxylamine
c_p / J/(mol*K)
T / ° C
Calculated data (Giguère and Liu, 1952)
Polynom 4th order (R2 = 0.99943)
Aly-Lee equation (R2 = 0.9972)
PPDS equation (R2 = 0.99944)


Free Gibbs energy of formation of hydroxylamine @ 25 degC and 1.013 bar
G_f,m / kJ/mol
Gas
Liquid
Solid

A.4 Henry's law

Vapor pressures calculated for the present substances with *Henry's* law. The *Henry* coefficients and their temperature dependencies were obtained from [**sander_compilation_2015**] (for the code see A.17).



A.5 Overview of energy contents of all DSC measurements

Unstabilized HA SS C1 (Exp3)			
Exp	q / J/g	q_rest / J/g	q_tot / J/g
Dyn4_12mg	2342		2342
Dyn4_7mg	2399		2399
Iso90	1623	220	1843
Iso110	2181	0	2181
Iso130	2875	0	2875
Iso150	1689	0	1689

Unstabilized HA Au C2 (Exp8)			
Exp	q / J/g	q_rest / J/g	q_tot / J/g
Dyn4	2457		2457
Iso90_1	787	931	1718
Iso90_2	1798	227	2025
Iso110	1777	154	1931
Iso130	2109	20	2129
Iso150	2545	0	2545

Unstabilized HA Au C3 (Exp14)			
Exp	q / J/g	q_rest / J/g	q_tot / J/g
Dyn4_1	2437		2437
Dyn4_2	2682		2682
Iso90	1266	517	1782
Iso110	1927	292	2219
Iso130	2205	61	2266
Iso150	2434	90	2524

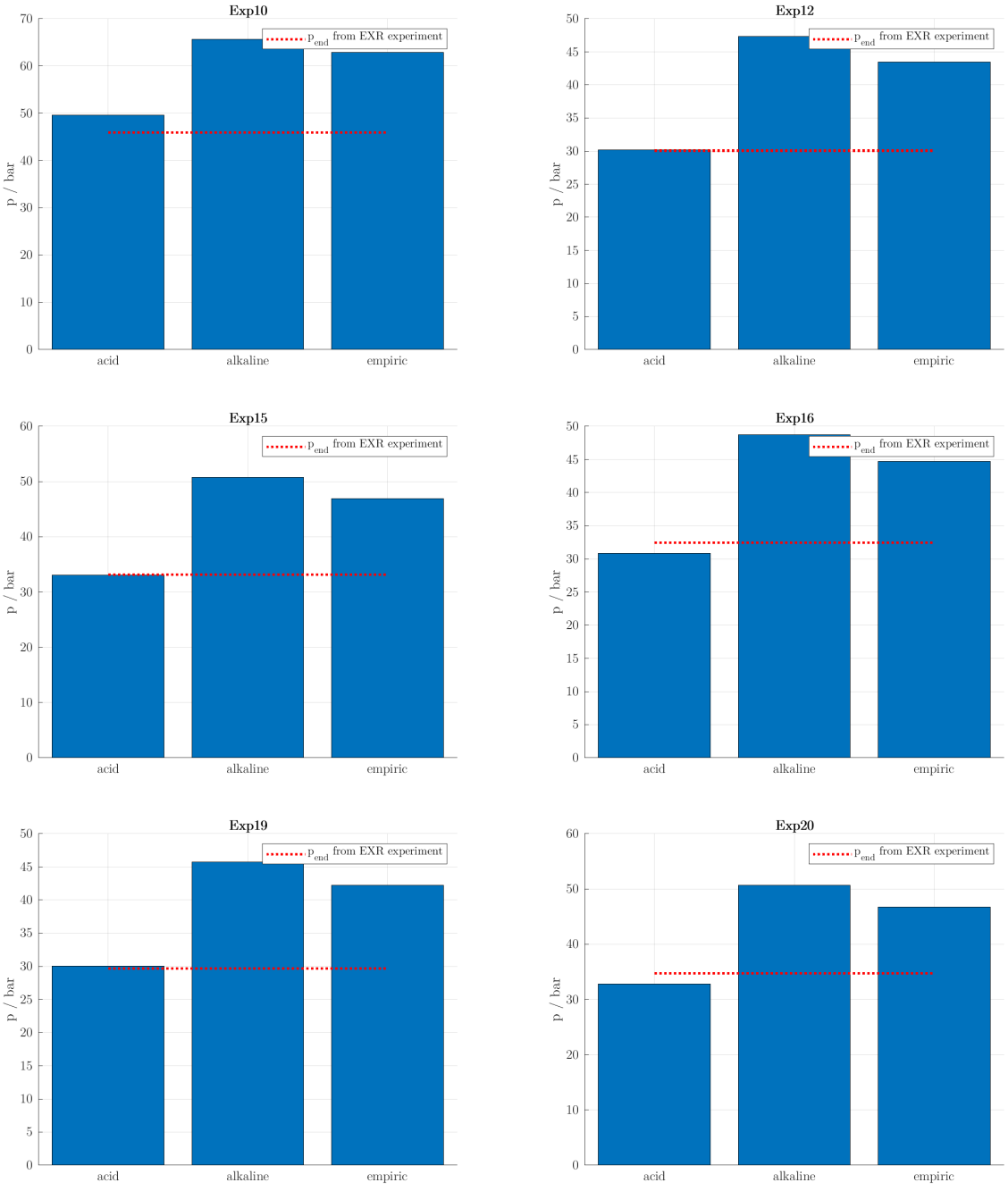
Stabilized HA Au (Exp16)			
Exp	q / J/g	q_rest / J/g	q_tot / J/g
Dyn4_1	2064		2064
Dyn4_2	2315		2315
Iso90	498	1388	1886
Iso110	248	1205	1453
Iso130	747	467	1214
Iso150	1814	78	1892
Iso170	2343	78	2421

Additional dynamic experiments			
Exp	q / J/g	q_rest / J/g	q_tot / J/g
Dyn4 SS (C5)	1764		1764
Dyn4 Au (extern)	2280		2280
Dyn4 SS (extern)	2246		2246

Average			
Exp	q / J/g	q_rest / J/g	q_tot / J/g
Dyn4	2299		2299
Iso90	1194	657	1851
Iso110	1533	413	1946
Iso130	1984	137	2121
Iso150	2120	42	2163
SS	-	-	2167
Au	-	-	2112
All	**-**	**-**	**2128**

A.6 Modeled end pressure for the experiments in the EXR

For every experiment the end pressure was calculated for the three reaction hypotheses and compared to the experimental pressure (for the code see appendix A.18).



A.7 Fitting with the AKTS software

The DSC measurements of the unstabilized hydroxylamine (done at FHNW) in the crucibles C1, C2, and C3 were evaluated with the *AKTS* software. The evaluation follows the steps listed below.

1. Custom integration or automatic baselines
2. Kinetics (define the underlying equations or relations that describe the behavior of the considered system)
3. Baseline optimization (automatic adjustment of the tangents of all baselines)
4. Simulation (calculate and display reaction progress)
5. Optimization (optimization for final adjustment of the kinetic parameters at the early stage of the reaction)
6. Prediction (based on the retrieved model parameters, it allows to foresee the behavior of the considered system under user specified conditions)

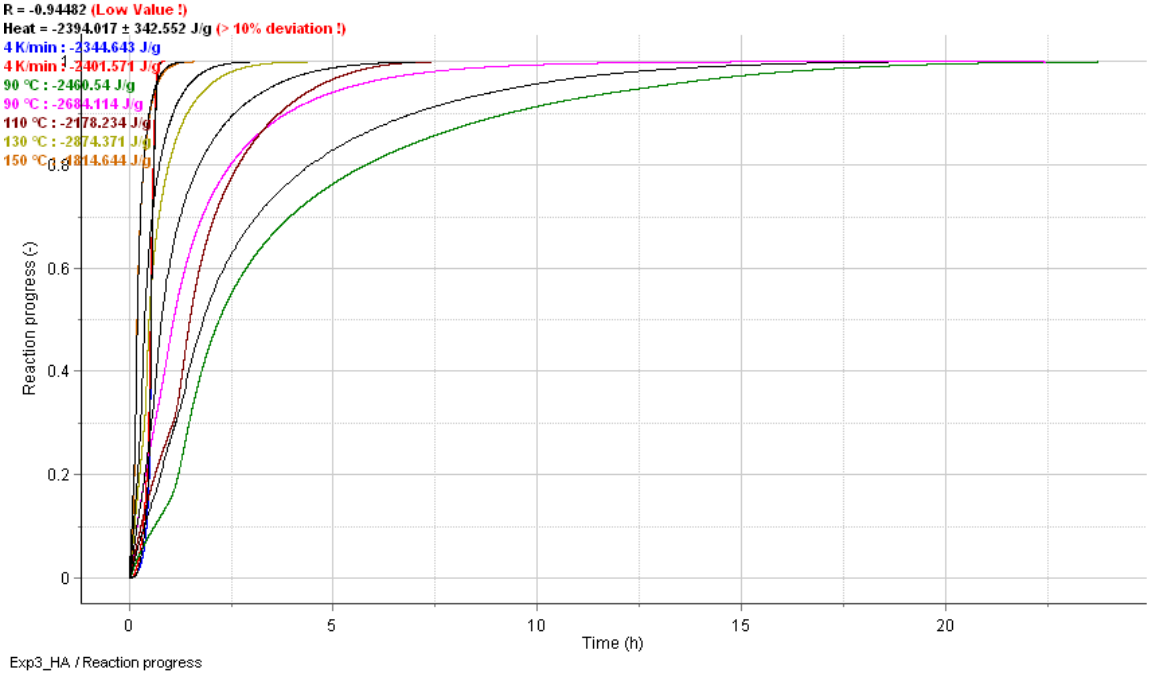
For the isothermal measurements, the automatic baseline determination was not possible (only for tangential baselines), therefore the baselines were defined manually, according to the evaluation of the measurements in the *STARe* software as a horizontal line from the last point. The step three was not possible either since not minimal two tangential baselines were applied before. For the fifth step, 100 iterations were the default number of iterations for the adjusting optimization. However, there was no apparent change of the model.

After the isoconversional model was established, the software allows the user to simulate a huge variety of scenarios, e.g. isothermal, dynamic and adiabatic.

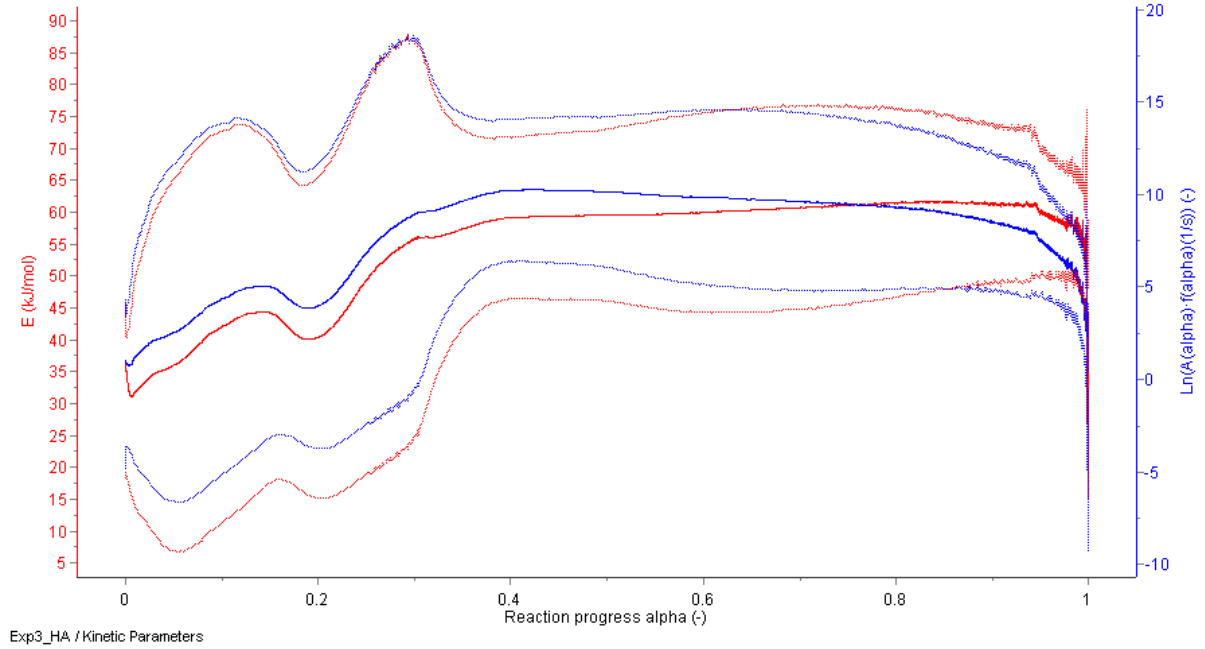
The results of the isoconversional fitting and the simulations of the TMR_{ad} and of a dynamic scenario (2.4 K/min) for the three crucible types are shown in appendices A.8 (C1), A.9 (C2) and A.10 (C3).

A.8 Evaluation of C1 measurements with AKTS software

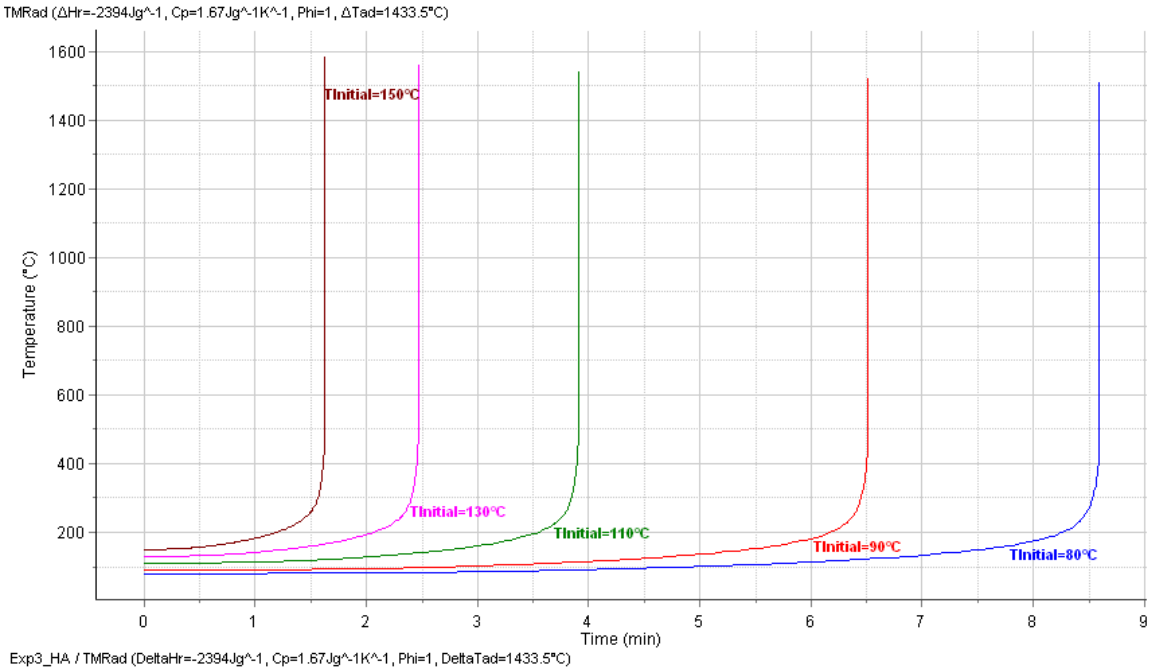
Isoconversional kinetics of crucible C1 done with the AKTS software.



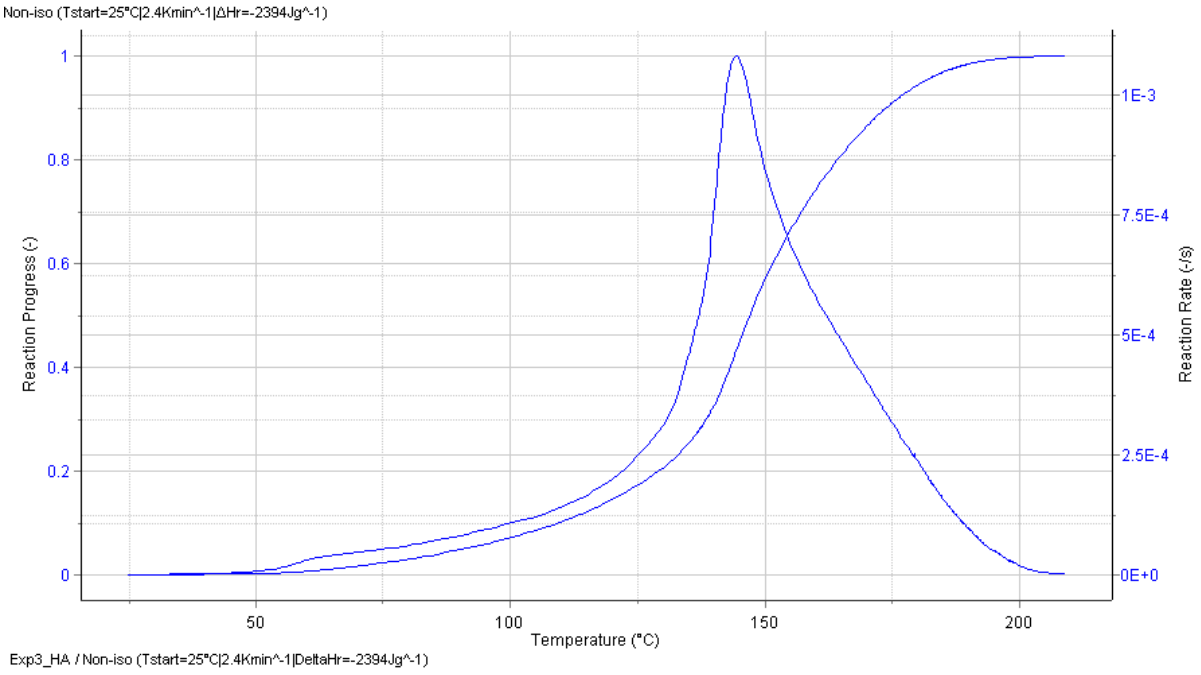
Kinetic parameters E_A (red) and $\ln(A)$ (blue) in dependence of the reaction progress α.



Simulation of time to maximal rate under adiabatic conditions at various initial temperatures.

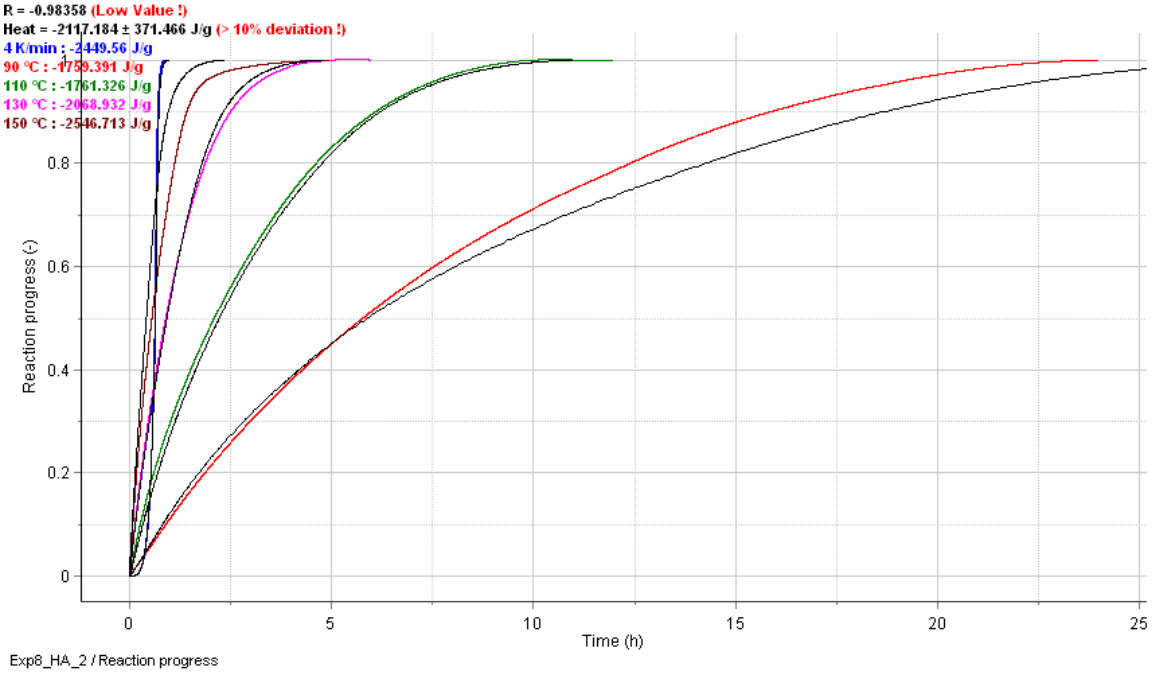


Simulation of dynamic scenario with a heating ramp of 2.4 K/min.

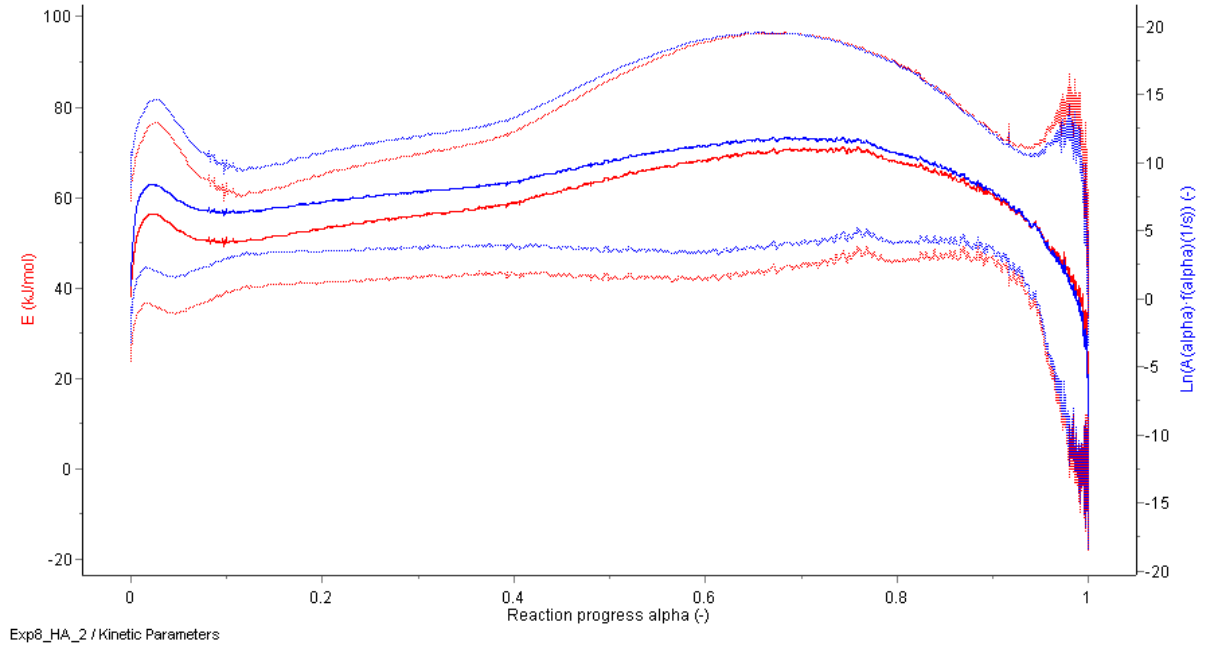


A.9 Evaluation of C2 measurements with AKTS software

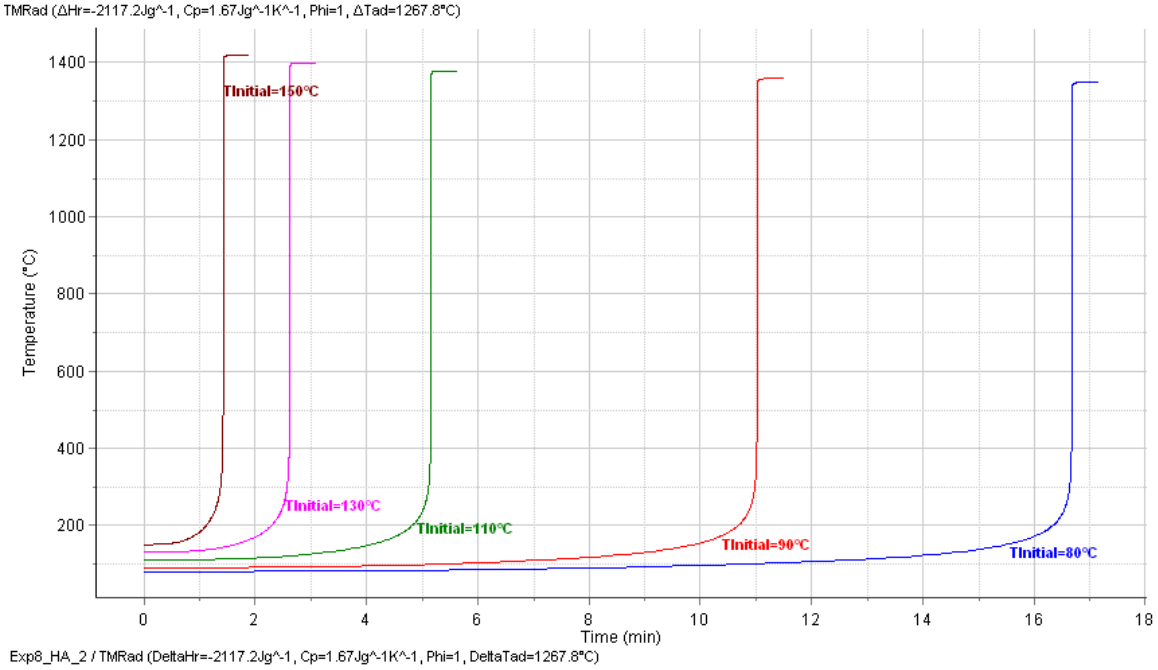
Isoconversional kinetics of crucible C2 done with the AKTS software.



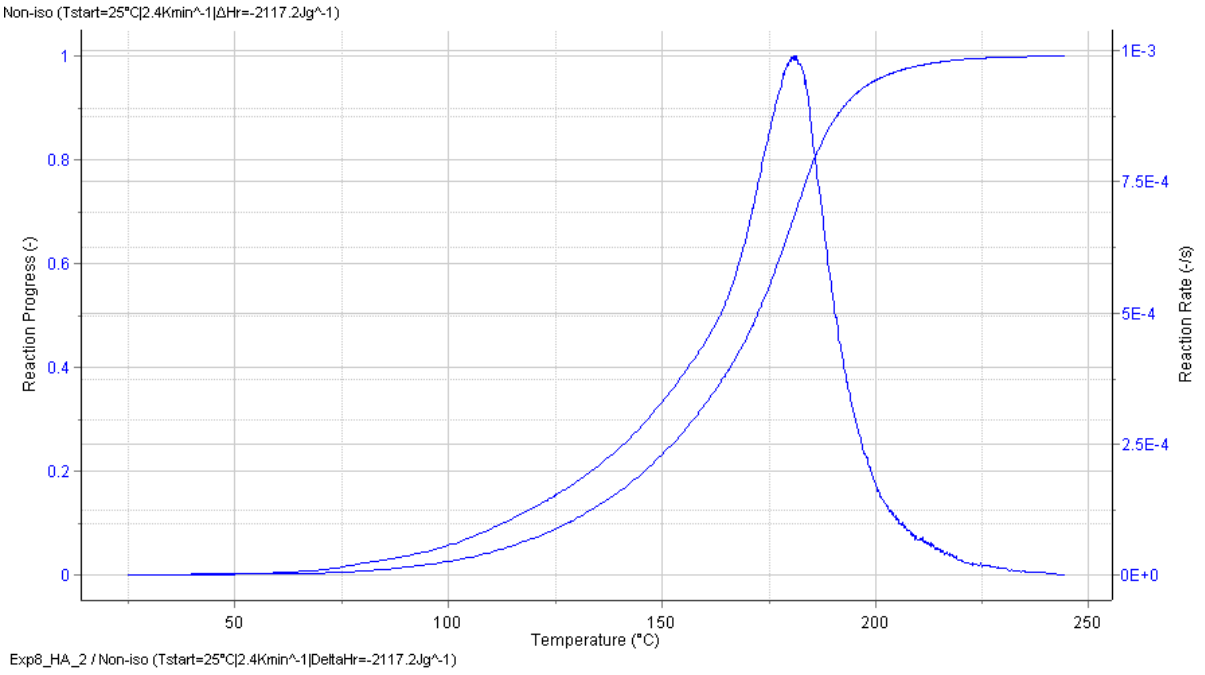
Kinetic parameters E_A (red) and $\ln(A)$ (blue) in dependence of the reaction progress α.



Simulation of time to maximal rate under adiabatic conditions at various initial temperatures.

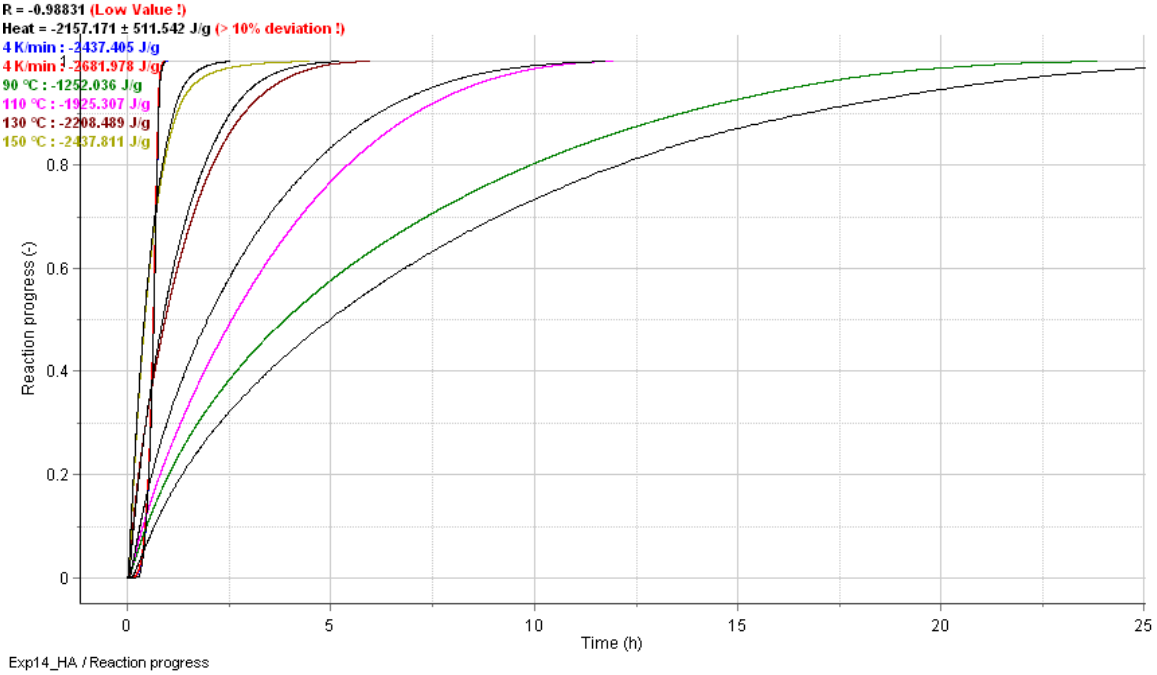


Simulation of dynamic scenario with a heating ramp of 2.4 K/min.

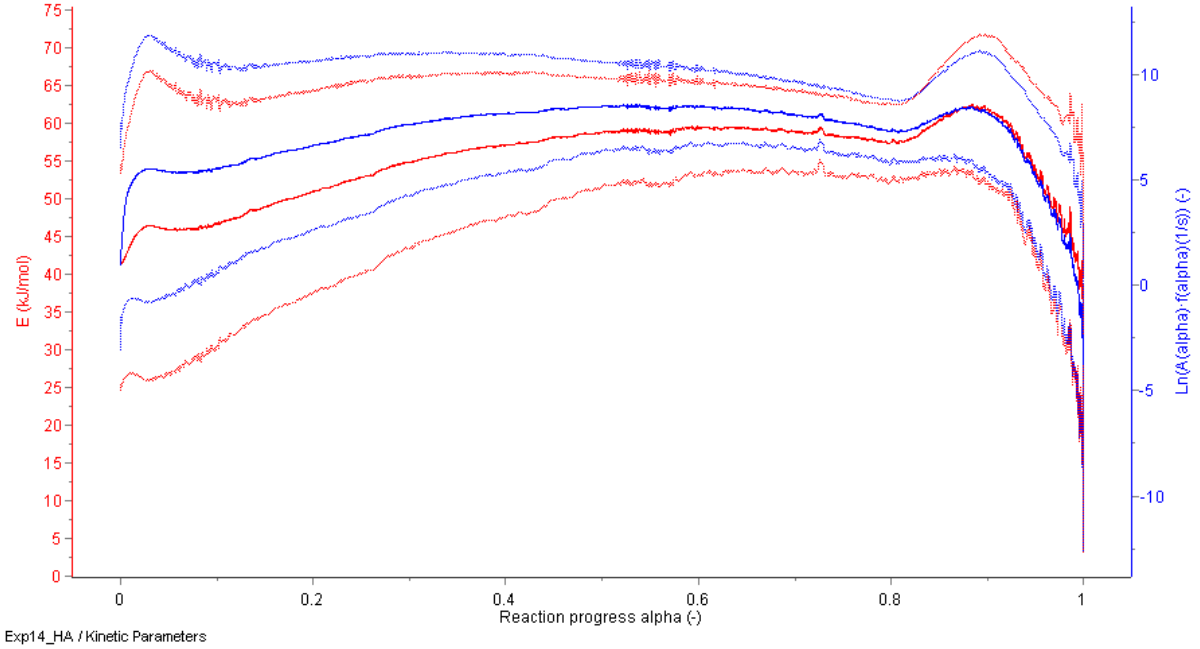


A.10 Evaluation of C3 measurements with AKTS software

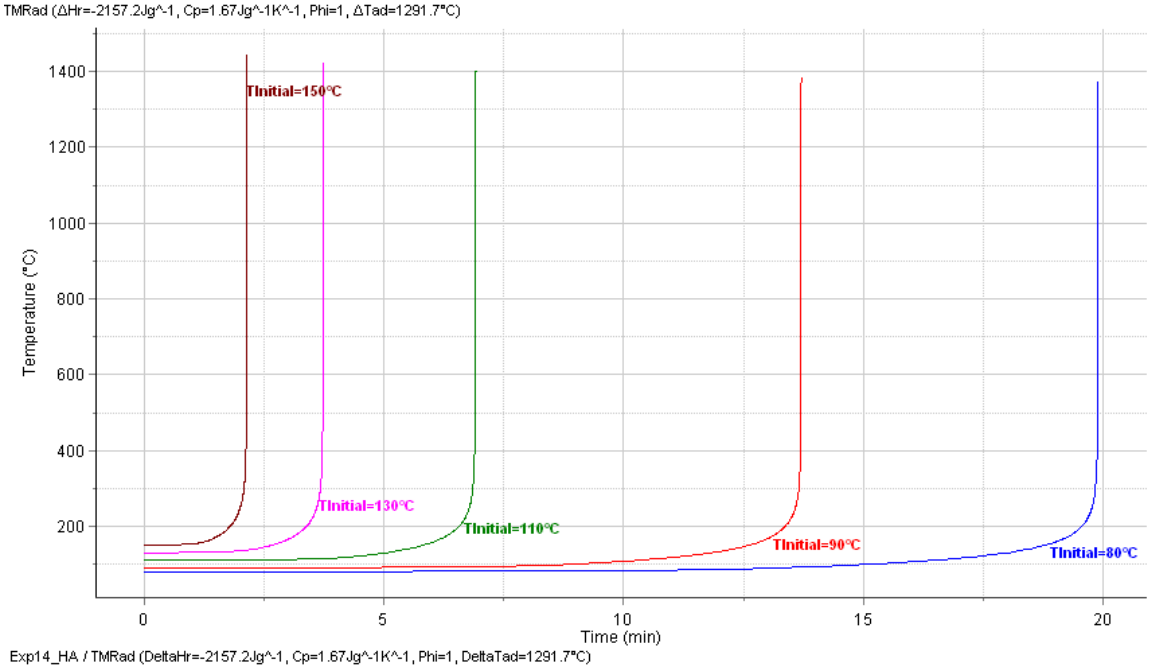
Isoconversional kinetics of crucible C3 done with the AKTS software.



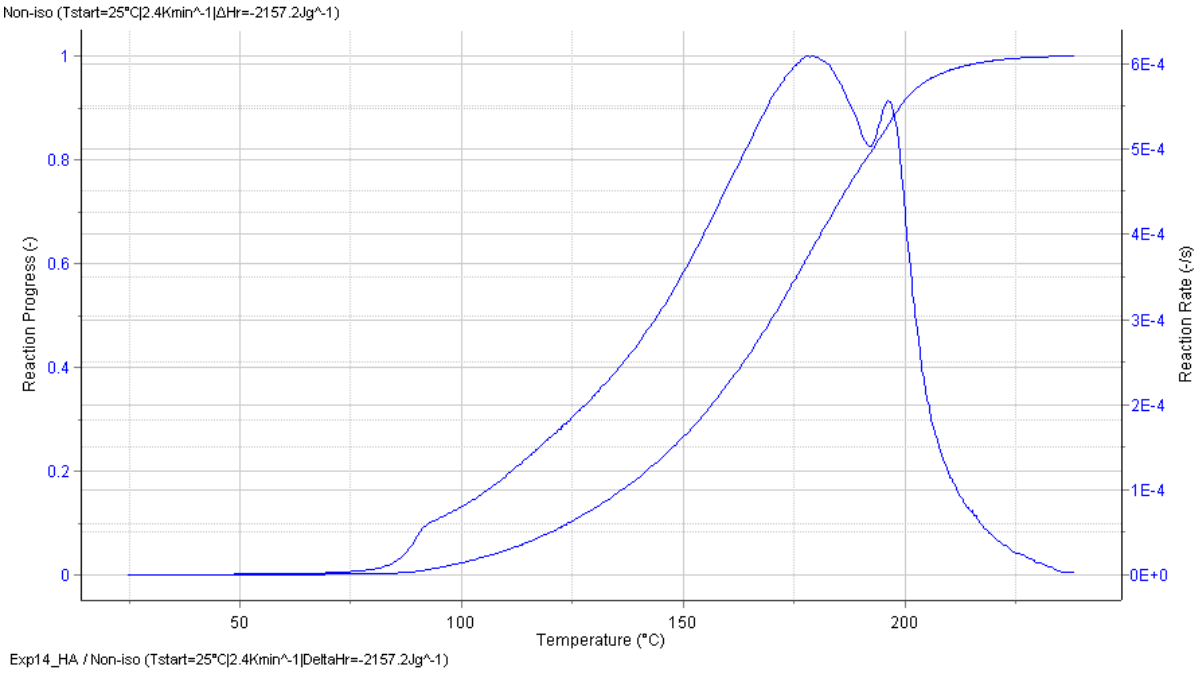
Kinetic parameters E_A (red) and $\ln(A)$ (blue) in dependence of the reaction progress α.



Simulation of time to maximal rate under adiabatic conditions at various initial temperatures.

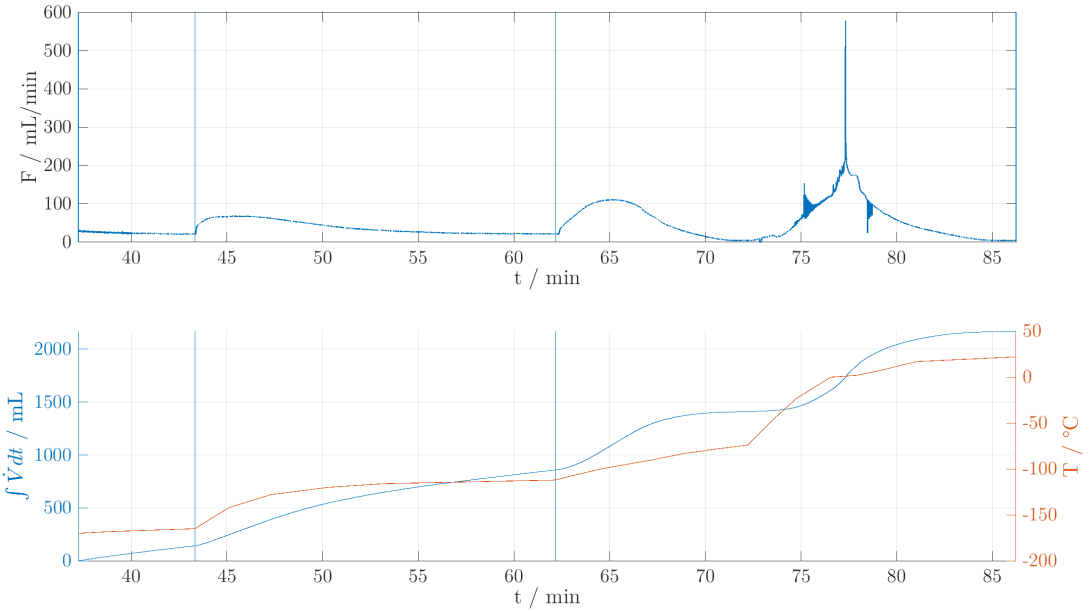


Simulation of dynamic scenario with a heating ramp of 2.4 K/min.



A.11 Exp16 - Gas analytics

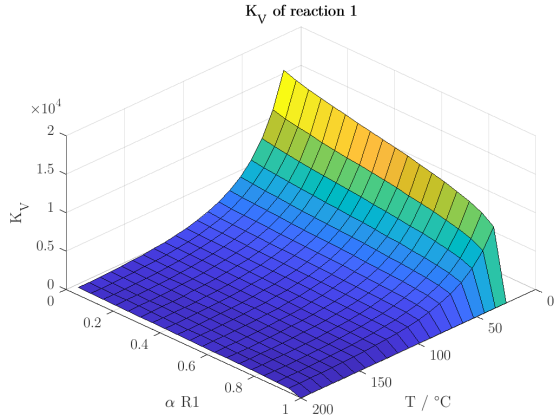
Results of the gas analytics of Exp16. The first figure is the measured volume flow. The second figure shows the integrated volume flow signal and the temperature measured at the bottom of the cooling trap.



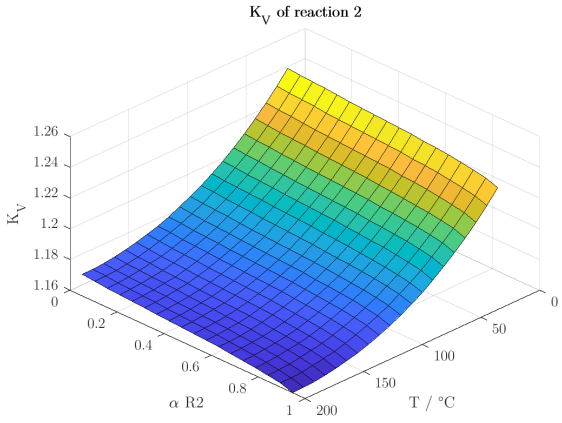
The total volume of the analyzed gas was 2170 mL. The volume ratios of the sectors defined according to the measured temperature are 0.10 to 0.55 to 1 (for the code see appendix A.11).

A.12 Chemical equilibrium with more ammonia

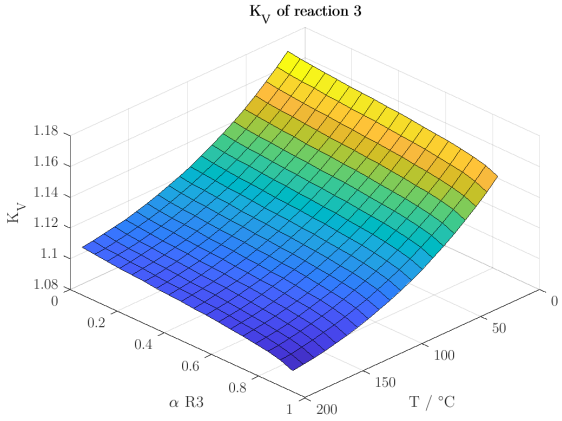
The thermodynamic calculations of the equilibrium constant were repeated for a mixture of 15 mL 50 % hydroxylamine solution and 15 mL 25 % ammonia solution.



Reaction hypothesis 1:

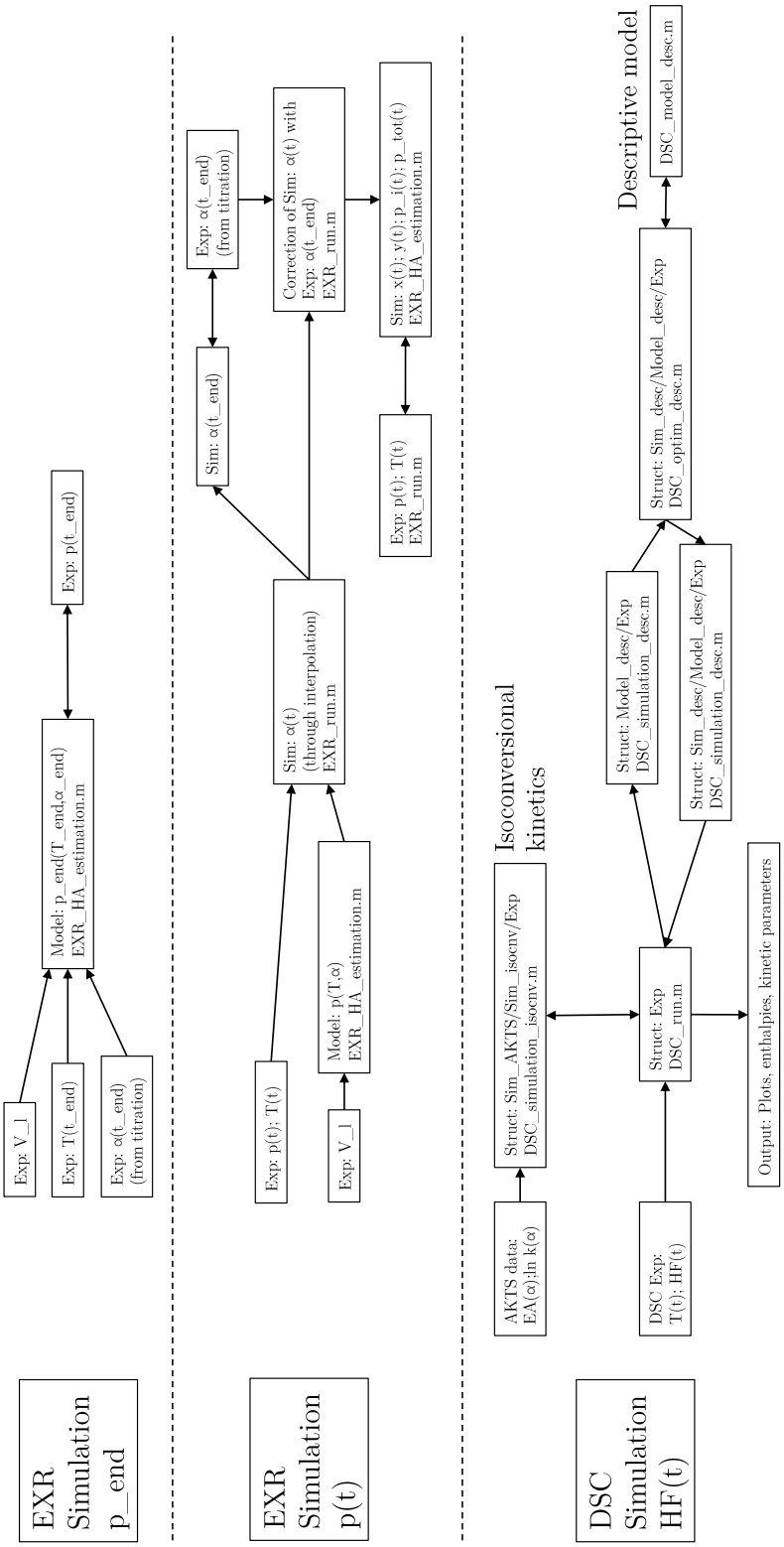


Reaction hypothesis 2:



Reaction hypothesis 3:

A.13 Calculation tree



A.14 estimation_HA_sd.m

Matlab code to calculate the estimation methods of the substance data of hydroxylamine.

```
% ------------------------------estimation_HA_sd.m------------------------------
% Florian Walthert
% 04.07.20

% Estimation calculation of the substance data of hydroxylamine
close all; clear all;
format short

%% Physical constant
R= 8.314;

%% Literature values for boiling and melting point
T_b_lit= 109.85; % degC, [Joback/ChemCAD]
T_m_lit= 33.5; % degC, [Detherm and others]

%% Critical Properties

% Guldberg rule [Gmehling p.66]
T_c = 3/2*(T_b_lit+273.15)-273.15;% degC
disp(append("Guldberg: T_c = ", num2str(T_c, 3), " degC"));

% Simple Correlation (applicable for inorganic compounds) [Perry p.2-468]
T_c = 1.64*(T_b_lit+273.15)-273.15;% degC
disp(append("Simple correlation: T_c = ", num2str(T_c, 3), " degC"));

% Ambrose (applicable for organic compounds) Perry p.2-468
% GC for NH2- and -OH
deltaT = [0.208; 0.138];
n = ((T_b_lit+273.15)-314)/19.2; % Correction for alcohols
factor = sum(1+(1.242+deltaT+[0; -0.138+(0.87-0.11*n+0.003*n^2)]).^(-1));
T_c = (T_b_lit+273.15)*factor-273.15;% degC
disp(append("Ambrose: T_c = ", num2str(T_c, 4), " degC"));

disp("p_c and V_c are obtain by simple Extrapolation!!!")

% Joback (applicable for organic compounds) [Perry p.2-470]
% GC for -OH and -NH2 : dT, dp, dV
GC = [0.0741, 0.0112, 28, 92.88; 0.0243, 0.0109, 38, 73.23];
n_A = 5; % Number of atoms
[T_b, T_c, p_c, V_c] = joback(n_A, GC);
disp(append(append("Joback: T_b = ", num2str(T_b-273.15, 4), " degC"), newline, ...
    append("Joback: T_c = ", num2str(T_c, 4), " degC"), newline, ...
    append("Joback: p_c = ", num2str(p_c, 4), " bar"), newline, ...
    append("Joback: V_c = ", num2str(V_c, 4), " cm^3/mol"))));
% Check consistency (0.21 < Z_c < 0.29)
Z_c = p_c*10^5 * V_c*10^-6 / ( R * (T_c+273.15))

%% Normal boiling point

%Nannoolal method (applicable for organic compounds) [Perry p. 2-473]

% GC for NH2-C and HO-C(chain<5C)
n=2; %Number of nonhydrogen atoms
ni=[1;1];Ci=[321.1759; 488.0819];
T_b = nannoolal(n, ni, Ci)-273.15;
disp(append("Nannoolal : T_b = ", num2str(T_b,3), "degC (Lit: ", num2str(T_b_lit,3), " degC
    (Joback))"));

%% Vapor pressure
% Data from [Pubchem]
T = [0, 32, 47, 56.5, 64.6, 77.5, 99.2] + 273.15; % K
pi = [0.36, 7.1, 13, 29.33093, 53, 133, 533]*10^-3; % bar
Ant = lsqcurvefit(@Antoine, [0,0,0], T, pi, [0, 0, -270], [Inf, Inf, 0])
trange = linspace(273, 393, 100);
chi = corrcoef(pi, Antoine(Ant, T));

figure(1)
plot(T-273.15,log10(pi), "x", "LineWidth", 2)
hold on
plot(trange-273.15, log10(Antoine(Ant, trange)), "LineWidth", 1)
xlabel("T / degC");ylabel("log_{10}(p^\ast/bar)");
legend("Empirical data points", "Antoine model", "Location", "northwest")
text(-17,-1, append("Antoine parameters:\newline A = ",num2str(Ant(1),4), ...
    "\newline B = ", num2str(Ant(2),4), "\newline C = ", num2str(Ant(3),4), ...
    "\newline\newlineR^2 = ", num2str(chi(1,2)^2,5)), "FontName", "LM Roman 12");
title("Antoine fit of vapor pressure curve of hydroxylamine");
grid on
set(gca, "FontName", "LM Roman 12");

clearvars trange

%% Heat capacity

% Ideal Gas: Benson and CHETAH (applicable for organic compounds) [Perry
```

```
84  % p.2-490]
85  Ttab = [298, 400, 500, 600, 800, 1000, 1500]; % K
86  % GC for OH-(C) and NH2-(C) for Ttab
87  GC1 = [18.2, 18.63, 20.18, 21.89, 25.2, 27.67, 33.65; 23.94, 27.25, 30.64, 33.78, 39.39, 43.83,
        51.4];
88  % GC for OH-(O) and NH2-(N) for Ttab
89  GC2 = [21.64, 24.24, 26.29, 27.88, 29.93, 31.44, 34.2; 25.53, 30.98, 35.16, 38.93, 43.95, 48.14,
        55.25];
90  n = [1; 1];
91  c_p_m_g1 = sum(n .* GC1); % J/mol/K
92  c_p_m_g2 = sum(n .* GC2); % J/mol/K
93  disp(append("Benson: OH-(C) and NH2-(C) cp_0 = ", num2str(c_p_m_g1(1)),...
94       " J/mol/K, OH-(O) and NH2-(N) cp_0 = ", num2str(c_p_m_g2(1)), " J/mol/K"));
95  % Ideal Gas: Interpolation [Perry p.2-490 (eq.2-46)]
96  trange = linspace(min(Ttab), max(Ttab), 100); % K
97  c_p_m_g_fit1 = polyval(polyfit(Ttab, c_p_m_g1, 4), trange);
98  c_p_m_g_fit2 = polyval(polyfit(Ttab, c_p_m_g2, 4), trange);
99
100 figure(2)
101 plot(Ttab-273.15, c_p_m_g1, "ok")
102 hold on; grid on
103 plot(trange-273.15, c_p_m_g_fit1, "k")
104 plot(Ttab-273.15, c_p_m_g2, "or")
105 plot(trange-273.15, c_p_m_g_fit2, "r")
106 xlabel("T / degC"); ylabel("c_{p,m} / J/mol/K");
107 title("Estimation methods for the heat capactity of hydroxylamine")
108
109 % Liquid: Ruzicka-Domalski [Perry p.2-495]
110 % GC for N(2H,C)(185-455K) and O(H,C)(155-505K)
111 GC1 = [8.2758, -.18365, .035272; 12.952, -10.145, 2.6261];
112 % GC for N(2H,N)(215-465K) and O(H,C)(155-505K)
113 GC2 = [6.8050, -0.72563, 0.15634; 12.952, -10.145, 2.6261];
114 ni = [1; 1];
115 trange = linspace(215, 455, 100); % K
116 c_p_m_l1 = ruzicka(trange, ni, GC1);
117 c_p_m_l2 = ruzicka(trange, ni, GC2);
118 disp(append("Ruzicka-Domalski: OH-(C) and NH2-(C) cp_o = ",
        num2str(c_p_m_l1(find(trange>298.15,1,"first"))),...
119      " J/mol/K, OH-(O) and NH2-(N) cp_0 = ", num2str(c_p_m_l2(find(trange>298.15,1,"first"))), "
           J/mol/K"));
120
121 figure(2)
122 plot(trange-273.15, c_p_m_l1, "ob")
123 plot(trange-273.15, c_p_m_l2, "oc")
124 xlabel("T / degC"); ylabel("c_{p,m} / J/mol/K");
125
126 % Interpolation of calculated heat capacity (IR) [Giguere]
127 T = [298.16, 300, 350, 400, 500, 600, 700, 800, 900, 1000, 1100, 1200]; % K
128 c_p_0 = [11.17, 11.20, 12.14, 13.02, 14.53, 15.74, 16.69, 17.39, ...
        17.92, 18.80, 19.43, 19.82]*4.187; % J/mol/K
129 c_p_polypar = polyfit(T, c_p_0, 4)
130 c_p_polyfit = polyval(c_p_polypar, T);
131 fun_alylee = @(x, T) x(1)+x(2)*(x(3)./T./sinh(x(3)./T))+x(4)*(x(5)./T./cosh(x(5)./T)).^2;
132 fun_ppds = @(x, T) x(2)+(x(3)-x(2)) * (T./(x(1)+T)).^2 .* ...
        (1-(x(1)./(x(1)+T)) .* (x(4)+x(5) .* (T./(x(1)+T)) + ...
        x(6)*(T./(x(1)+T)).^2+x(7)*(T./(x(1)+T)).^2));
133
134 par_alylee = lsqcurvefit(fun_alylee, [45.03, 45.63, 1378, -12.75,126.3], T, c_p_0, [0, 0, 0, -Inf,
        0],[]);
135 par_ppds = lsqcurvefit(fun_ppds, zeros(1,7), T, c_p_0, -100*ones(1,7), 1000*ones(1,7));
136
137 corr_polyfit = corrcoef(c_p_0, c_p_polyfit);
138 corr_alylee = corrcoef(c_p_0, fun_alylee(par_alylee, T));
139 corr_ppds = corrcoef(c_p_0, fun_ppds(par_ppds, T));
140
141 figure(3)
142 hold on; grid on
143 plot(T-273.15, c_p_0, "x", "LineWidth", 1)
144 plot(T-273.15, c_p_polyfit, "LineWidth", 1)
145 plot(T-273.15, fun_alylee(par_alylee, T), "LineWidth", 1)
146 plot(T-273.15, fun_ppds(par_ppds, T), "LineWidth", 1)
147 xlabel("T / deg C");ylabel("c_p / J/(mol*K)");
148 title("Interpolation methods for the heat capactity of hydroxylamine")
149 legend("Calculated data (Giguere and Liu, 1952)", ...
150      append("Polynom 4^{th} order (R^2 = ", num2str(corr_polyfit(1,2)^2,5), ")"), ...
151      append("Aly-Lee equation (R^2 = ", num2str(corr_alylee(1,2)^2,5), ")"), ...
152      append("PPDS equation (R^2 = ", num2str(corr_ppds(1,2)^2,5), ")"), ...
153      "Location", "southeast");
154 set(gca, "FontName", "LM Roman 12");
155
156 % Solid @298K: Modified Kopp's Rule [Perry p.2-497]
157 % GC for H, O, N
158 GC = [7.56, 13.42, 18.74]; ni = [3, 1, 1];
159 c_p_m_s = sum(ni .* GC);
160 disp(append("Modified Kopp's Rule : c_p_m_s = ", num2str(c_p_m_s,4), ...
161      "J/mol/K (Lit: 71.8 J/mol/K (Joback))"));
162 figure(2)
163 plot(298-273.15, c_p_m_s, "sm")
164 set(gca, 'FontName', 'LM Roman 12');
165
```

```
169 % Calculated data (IR) [Giguere]
170 plot(T, c_p_0, "x", "LineWidth", 1)
171 hold off
172 legend("Ideal Gas GC1(Benson)", "Ideal Gas GC1(Fit)", "Ideal Gas GC2(Benson)", ...
173     "Ideal Gas GC2(Fit)", "Liquid GC1(Ruzicka-Domalski)", ...
174     "Liquid GC2(Ruzicka-Domalski)", "Solid (Modified Kopp's Rule)", ...
175     "Calculated data (Giguere and Liu, 1952)");
176
177 %% Density
178 % Liquid: Rackett method [Perry p.2-503]
179 Z_c = p_c * V_c / (R * (T_c+273.15)); % Critical compressibility factor
180 T = 100 ;% degC
181 rho = 1 / (R * (T_c + 273.15) / (p_c * 10^5) * Z_c^(1+(1-(T+273.15)/(T_c+273.15))^(2/7)));
182 disp(append("Rackett : rho = ", num2str(rho,4), "kg/m^3 (@", num2str(T,3), " degC)"));
183
184 %% Enthalpy of Formation and Entropy -> Gibbs free energy
185
186 % Domalski-Hearing [Perry p.2-278]
187 % GC for O-(H)(C) and N-(2H)(C) (first, amino acids) as deltaH_f_g, S_g,
188 % deltaH_f_l, S_l, deltaH_f_s, S_s
189 GC = [-159.33, 121.5, -191.5, 43.89, -199.66, 28.62; 19.25, 124.4, 0.33, 71.71, -6.3, 39.00];
190 ni= [1; 1];
191 deltaH_f_g = sum(ni .* GC(:,1)); % kJ/mol
192 S_g = sum(ni .* GC(:,2)); % J/mol/K
193 deltaH_f_l = sum(ni .* GC(:,3)); % kJ/mol
194 S_l = sum(ni .* GC(:,4)); % J/mol/K
195 deltaH_f_s = sum(ni .* GC(:,5)); % kJ/mol
196 S_s = sum(ni .* GC(:,6)); % J/mol/K
197 disp("Domalski-Hearing method");
198 disp(append("deltaH_f_l = ", num2str(deltaH_f_l), " kJ/mol"));
199 disp(append("deltaH_f_s = ", num2str(deltaH_f_s), " kJ/mol"));
200 disp(append("S_l = ", num2str(S_l), " J/mol/K"));
201 disp(append("S_s = ", num2str(S_s), " J/mol/K"));
202
203 T= 25 + 273.15;
204 deltaG_g = deltaH_f_g - T * S_g/1000; % kJ/mol
205 deltaG_l = deltaH_f_l - T * S_l/1000; % kJ/mol
206 deltaG_s = deltaH_f_s - T * S_s/1000; % kJ/mol
207 deltaG = [deltaG_g, deltaG_l, deltaG_s];
208
209 figure(4)
210 errorbar(deltaG, 0.03*deltaG, "ok")
211 grid on
212 ylabel("G_{f,m} / kJ/mol")
213 title(append("Free Gibbs energy of formation of hydroxylamine @ ", num2str(T-273.15, 3), " degC
        and 1.013 bar"))
214 set(gca, 'XTick', 1:3, 'XTickLabel', ["Gas", "Liquid", "Solid"], "Xlim", [0,4], 'FontNAme', 'LM␣
        Roman␣12');
215
216 %% Heat of Vaporization
217
218 % CS method [Perry p.2-487] (applicable for organic compounds, and also possible
219 % as a f(T))
220 omega = 0.694; % Acentric factor
221 T_r = (T_b_lit+273.15)/(T_c+273.15); tau = 1-T_r;
222 deltaH_v1 = R * (T_c+273.15) * (7.08*tau^0.354 + 10.95*omega*tau^0.456);
223 disp(append("CS: deltaH_v = ", num2str(deltaH_v1/1000,3), " kJ/mol (Lit: 42.9 kJ/mol
        (Joback/ChemCAD))"));
224
225 % Vetere method [Perry p.2-487] (applicalbe only at normal bp)
226 T_b_r = (T_b_lit+273.15)/(T_c+273.15); tau_b = 1-T_b_r; F=1; % [Perry]
227 deltaH_v2 = R * (T_b_lit+273.15) * tau_b^0.38 * (log(p_c)-0.513+0.5066/(p_c*T_b_r^2)) / ...
228     (tau_b+F*(1-tau_b^0.38)*log(T_b_r));
229 disp(append("Vetere: deltaH_v = ", num2str(deltaH_v2/1000,3), " kJ/mol (Lit: 42.9 kJ/mol
        (Joback/ChemCAD))"));
230
231 %% Heat of Fusion
232
233 % Chickos method [Perry p.2-487]
234 a = 0; % [Perry]
235 % GC for -OH and -NH2 as C1/2/3/4, deltaS
236 GC = [1, 12.6, 18.9, 26.4, 1.13; 1, 1, 0, 0, 15.48]; k= 2; % Number of fuctional groups
237 b = 0 + 0 + sum(GC(:,k) .* GC(:,5));
238 deltaH_m = (T_m_lit+273.15) * (a+b);
239 disp(append("Chickos: deltaH_m = ", num2str(deltaH_m/1000,3), " kJ/mol (Lit: -161 kJ/mol
        (Joback/Chemeo))"));
240
241 %% Functions
242
243 % Joback
244 function [T_b, T_c, p_c, V_c] = joback(n_A, GC, T_b)
245 if(~exist('T_b','var'))
246     T_b = 198 + sum(GC(:, 4)); %K
247 else
248     T_b = T_b+273.15;
249 end
250
251 T_c = T_b * (0.584 + 0.965 * sum(GC(:,1))-sum(GC(:,1))^2)^(-1) - 273.15; % degC
252 p_c = (0.113 + 0.0032 * n_A - sum(GC(:,2)))^(-2); % bar
253 V_c = 17.5 + sum(GC(:, 3)); % cm3/mol
```

```
254
255 end
256
257 % Nannoolal
258 function T_b = nannoolal(n, ni, Ci)
259 T_b = sum(ni.*Ci)./(n^0.6583+1.6868) + 84.3395;
260 end
261
262 % Ruzicka
263 function c_p_m = ruzicka(T, ni, GC)
264 R = 8.314;
265 c_p_m = R * (sum(ni .* GC(:,1)) + sum(ni .* GC(:,2)) .* (T/100) + sum(ni .* GC(:,3)) .*
        (T/100).^2); % J/mol/K
266 end
267
268 % ------------------------------------------------------------------------
```

A.15 reactor_volume.m

Matlab code to calculate the reactor volume.

```
1  % ------------------------------reactor_volume.m-------------------------------
2  % Florian Walthert
3  % 26.07.20
4
5  clear all; close all;
6
7  %% Load data from Bronkhorst flow meter/controller
8  % Import from txt-File
9  filename = "Exp18_N2";
10
11 opts = delimitedTextImportOptions("NumVariables", 5);
12 opts.DataLines = [1, Inf];
13 opts.Delimiter = ";";
14 opts.VariableNames = ["i", "t", "F", "T", "Valve"];
15 opts.SelectedVariableNames = ["i", "t", "F", "T", "Valve"];
16 opts.VariableTypes = ["string", "string", "string", "string", "string"];
17 opts = setvaropts(opts, [1, 2, 3, 4, 5], "WhitespaceRule", "preserve");
18 opts = setvaropts(opts, [1, 2, 3, 4, 5], "EmptyFieldRule", "auto");
19 opts.ExtraColumnsRule = "ignore";
20 opts.EmptyLineRule = "read";
21 tbl = readtable(append('Data\Bronkhorst\', filename, '.txt'), opts);
22
23 % Convert to output type
24 tbl.i = str2double(strrep(tbl.i,',','.'));
25 tbl.t = str2double(strrep(tbl.t,',','.'));
26 tbl.T = str2double(strrep(tbl.T,',','.'));
27 tbl.F = str2double(strrep(tbl.F,',','.'));
28 tbl.Valve = str2double(strrep(tbl.Valve,',','.'));
29
30 FC.t = tbl.t; % s
31 FC.T = tbl.T; % degC
32 FC.F = tbl.F; % mL/min
33 FC.Valve = tbl.Valve; % %
34
35 % Clear temporary variables
36 clear opts tbl
37
38 % Baseline correction of the volume flow signal
39 FC.ind_bl = [1:find(FC.t>=0.445*3600,1,'first'),
        find(FC.t>=0.669*3600,1,'first'):find(FC.t>=0.754*3600,1,'first'),
        find(FC.t>=0.978*3600,1,'first'):length(FC.t)]';
40 FC.bl = mean(FC.F(FC.ind_bl));
41 FC.F = FC.F - FC.bl;
42
43 %% Load data from Eurotherm recorder (EXR)
44
45 % Import from csv-File
46 filename = "Exp18_1";
47 opts = delimitedTextImportOptions("NumVariables", 7);
48 opts.DataLines = [14, Inf];
49 opts.Delimiter = ";";
50 opts.VariableNames = ["t", "Var2", "Tr", "p", "pl", "Var6", "Var7"];
51 opts.SelectedVariableNames = ["t", "Tr", "p", "pl"];
52 opts.VariableTypes = ["string", "string", "string", "string", "string", "string", "string"];
53 opts = setvaropts(opts, [1, 2, 3, 4, 5, 6, 7], "WhitespaceRule", "preserve");
54 opts = setvaropts(opts, [1, 2, 3, 4, 5, 6, 7], "EmptyFieldRule", "auto");
55 opts.ExtraColumnsRule = "ignore";
56 opts.EmptyLineRule = "read";
57 tbl = readtable(append('Data\Eurotherm\', filename, '.csv'), opts);
58
59 % Convert to output type
60 EXR.t0 = tbl.t(1);
61 EXR.t = (datenum(datetime(tbl.t) - datetime(EXR.t0))) * 24 *3600; %s
```

```
62  EXR.Tr = smoothdata(str2double(strrep(tbl.Tr,',','.')), 'gaussian', 1); % degC
63  EXR.p = smoothdata(str2double(strrep(tbl.p,',','.')), 'gaussian', 1); % bar
64
65  % Indices for state 1 and 2
66  % First determination
67  ind_11 = find(EXR.t>=21470,1,'first'):find(EXR.t>=21700,1,'first');
68  ind_12 = find(EXR.t>=22570,1,'first'):find(EXR.t>=22660,1,'first');
69  % Second determination
70  ind_21 = find(EXR.t>=22760,1,'first'):find(EXR.t>=22820,1,'first');
71  ind_22 = find(EXR.t>=23660,1,'first'):find(EXR.t>=23790,1,'first');
72
73  % Pressures and temperatures in the two states
74  p1 = [mean(EXR.p(ind_11)), mean(EXR.p(ind_21))]; % bar
75  p2 = [mean(EXR.p(ind_12)), mean(EXR.p(ind_22))]; % bar
76  T1 = [mean(EXR.Tr(ind_11)), mean(EXR.Tr(ind_21))] + 273.15; % K
77  T2 = [mean(EXR.Tr(ind_12)), mean(EXR.Tr(ind_22))] + 273.15; % K
78
79  % Clear temporary variables
80  clear opts tbl
81
82  %% Calculation of reactor volume
83
84  % Calculation
85  FC.int = cumtrapz(FC.t, FC.F/60); % mL
86
87  V_g = [FC.int(find(FC.t>=0.7*3600,1,'first')), FC.int(end) -
        FC.int(find(FC.t>=0.7*3600,1,'first'))]; % mL
88
89  for i = 1:length(V_g)
90      V_i(i) = V_g(i)*p2(i)*T1(i) / (p1(i)*T2(i)-p2(i)*T1(i)); % mL
91  end
92
93  V_r = mean(V_i); % mL
94
95  % Display result
96  disp(append('Reactor volume: V_r = ', num2str(V_r), ' mL'));
97
98  %% Plot measuring signals and calculations
99  figure
100
101 subplot(2,1,1)
102
103 yyaxis right
104 plot(FC.t/3600-0.4, FC.int)
105 ylabel("$\int \dot{V} dt / mL$", 'Interpreter', 'latex')
106
107 yyaxis left
108 plot(FC.t/3600-0.4, FC.F, 'LineWidth', 1)
109 xlabel('t / h'); ylabel("$\dot{V} / mL/min$", 'Interpreter', 'latex')
110
111 grid on
112 set(gca, 'FontName', 'LM Roman 12')
113
114 subplot(2,1,2)
115
116 yyaxis right
117 plot(EXR.t/3600-6, EXR.Tr, 'LineWidth', 1)
118 xlabel('t / h'); ylabel('T / degC')
119
120 yyaxis left
121 plot(EXR.t/3600-6, EXR.p, 'LineWidth', 1)
122 xlabel('t / h'); ylabel('p / bar')
123 set(gca, 'Xlim', [5.8955, 6.6760]-6);
124
125 grid on
126 set(gca, 'FontName', 'LM Roman 12')
127
128 % ------------------------------------------------------------------------
```

A.16 vaporpressure_gases.m

Matlab code to plot the vapor pressure curves and the melting points of the gases present in the cooling trap.

```
1  % ------------------------vaporpressure_gases.m-------------------------
2  % Florian Walthert
3  % 17.6.20
4
5  % Plot the vapor pressure curves and the melting points of the gases present
6  % in the cooling trap
7  clear all; close all;
8
9  %% Vapor pressure data from Perry p.2-61
```

```
10
11  % 1: NH3
12  % 2: N2
13  % 3: N2O
14  % 4: NO
15  % 5: H2
16
17  p_i = [1, 5, 10, 20, 40, 60, 100, 200, 400, 760] * 1.013 / 760;
18
19  T = [
20       -109.1, -97.5, -91.9, -85.8, -79.2, -74.3, -68.4, -57.0, -45.5, -33.6;
21       -226.1, -221.3, -219.1, -216.8, -214.0, -212.3, -209.7, -205.6, -200.9, -195.8;
22       -143.4, -133.4, -128.7, -124.0, -118.3, -114.9, -110.3, -103.6, -96.2, -85.5;
23       -184.5, -180.6, -178.2, -175.3, -171.7, -168.9, -166.0, -162.3, -156.8, -151.7;
24       -263.3, -261.9, -261.3, -260.4, -259.6, -258.9, -257.9, -256.3, -254.5, -252.5
25       ];
26
27  T_m = [-77.7; -210.0; -90.9; -161; -259.1];
28
29  %% Plots
30
31  figure
32  hold on
33  p1 = plot(T, smooth(p_i), "LineWidth", 2)
34  grid on
35  xlabel("T / degC")
36  ylabel("$p^\ast$ / bar", 'Interpreter', 'latex')
37
38  col = get(gca, "colororder")
39  for i = 1:length(T_m)
40      p2 = xline(T_m(i), ":", "Color", col(i,:), "Linewidth", 1)
41  end
42  legend(p1, ["NH_3", "N_2", "N_2O", "NO", "H_2"], "Location", "southeast")
43  set(gca, "Xlim", [-270,-20], "FontName", "LM Roman 12");
44  % title("Vapor Pressure Curves and Melting Points of relevant gases")
45
46  % ------------------------------------------------------------------------
```

A.17 henry_coefficients.m

Matlab code to the partial pressures calculated with *Henry*'s law.

```
1   % ------------------------------henry_coefficients.m-------------------------------
2   % Florian Walthert
3   % 15.05.20
4
5   close all;clear all;
6
7   % Plot the partial pressures calculated with Henry's law
8   %% Basic Data
9   R = 8.314; % J/mol/K
10  T_0 = 298.15; % K
11  p_0 = 1.013 * 10^5; % Pa
12  %% Substance Data
13  mw = [
14       33.03,
15       17,
16       28,
17       44,
18       30,
19       18,
20       2
21       ] * 10^-3; %kg/mol
22
23  rho_H2O = 1000; %kg/m^3
24  V = 0.15 * 10^-3; %m^3
25  V_l = 0.02 * 10^-3; %m^3
26  V_g = V - V_l; %m^3
27  w_HA = 0.5;
28
29  par_henry = [
30       1.40E+03, 0;
31       5.90E-01, 4200;
32       6.40E-06, 1300;
33       2.40E-04, 2600;
34       1.90E-05, 1600;
35       nan, nan;
36       7.70E-06, 500
37       ];
38
39  subname =["NH_2OH", "NH_3", "N_2", "N_2O", "NO", "H_2O", "H_2"];
40  ind = 1:length(subname);
41  ind(6) = [];
```

```
42
43  %% Calculation of the partial pressure
44  T= linspace(270, 400, 10);
45  x = linspace(0, 0.5, 10);
46
47  for i = 1:length(ind)
48      for s = 1:length(T)
49          p_i(:,s,i) = henry(x', par_henry(ind(i),:), T(s));
50      end
51  end
52
53  %% Plot
54  figure
55  for i = 1:length(ind)
56      subplot(3,2,i)
57      surf(T, x, p_i(:,:,i))
58      xlabel("T / K");ylabel(texlabel("x / mol/mol")); zlabel("$p_i^\ast$ / Pa", 'Interpreter',
            'latex')
59      title(subname(ind(i)))
60      set(gca, 'FontName', 'LM Roman 12')
61  end
62
63  % ------------------------------------------------------------------------
```

A.18 EXR_HA_estimation.m

Script to model the hydroxylamine decomposition reaction in the EXR.

```
1   % ------------------------------HA_EXR_estimation.m-------------------------------
2   % Florian Walthert
3   % 04.07.20
4
5   % Script to model the hydroxylamine decomposition reaction in the EXR
6   clear all;close all;
7   tic
8   %% General
9   % R1: 157 NH2OH <-> 61 NH3 + 35 N2 + 12 N2O + 2 NO + 143 H2O + H2
10  % R2: 3 NH2OH <-> N2 + NH3 + 3H2O
11  % R3: 4 NH2OH <-> N2O + 2NH3 + 3H2O
12
13  % 1: NH2OH
14  % 2: NH3
15  % 3: N2
16  % 4: N2O
17  % 5: NO
18  % 6: H2O
19  % 7: H2
20  subs = ["NH_2OH"; "NH_3"; "N_2"; "N_2O"; "NO"; "H_2O"; "H_2"];
21
22  %% Basic Data
23  R = 8.314; % J/mol/K
24  T_0 = 293.15; % K (NIST uses this temperature as a reference)
25  p_0 = 1.013 * 10^5; % Pa
26  % format long
27
28  %% Substance Data
29  % General
30
31  sd.mw = [
32      33.03,
33      17,
34      28,
35      44,
36      30,
37      18,
38      2
39      ] * 10^-3; %kg/mol
40
41  % Thermodynamic
42
43  sd.G_f_0 = [
44      nan, %NH2OH [kJ/mol]
45      nan, %NH3 [kJ/mol]
46      -57.128, %N2 [kJ/mol]
47      nan, %N2O [kJ/mol]
48      27.41, %NO [kJ/mol]
49      298.13, %H2O [kJ/mol]
50      -38.962 %H2 [kJ/mol]
51      ];
52
53  sd.Cp_0 = [
54      49, % [J/(mol*K)] Estimation
55      nan, % [J/(mol*K)]
56      28.0134*1.0396, % [J/(mol*K)]
```

```
57        38.6, % [J/(mol*K)]
58        30.0061*0.9965, % [J/(mol*K)]
59        18.0153*1.8646, % [J/(mol*K)]
60        2.01588*14.304 % [J/(mol*K)]
61        ];
62
63    sd.Cp_0_poly = [
64        -4.01033196349413e-11, 1.57562557145388e-07,...
65        -0.000242012285748035, 0.197695530171796, 5.33391908027044
66        ];
67
68    sd.deltaH_f_0 = [
69        -106.7, % [kJ/mol]
70        -45.9, % [kJ/mol] g / aq [Vanderzee]
71        0, % [kJ/mol]
72        82.05, % [kJ/mol]
73        90.25, % [kJ/mol]
74        -285.83,%-241.83,% [kJ/mol]
75        0 % [kJ/mol]
76        ];
77
78    sd.S_f_0 = [
79        236; % [J/(mol*K)] [Ghahremanpour] %calculated from deltaG: 236;
80        192.77, % [J/(mol*K)] g/ aq [Vanderzee]
81        191.609, % [J/(mol*K)]
82        219.96, % [J/(mol*K)]
83        210.76, % [J/(mol*K)]
84        69.95;%188.835, % [J/(mol*K)] liquid;gas
85        130.680 % [J/(mol*K)]
86        ];
87
88    sd.T_c = [
89        271+273.5;
90        405.65;
91        126.2;
92        309.57;
93        180.15;
94        647.096;
95        33.19
96        ]; % [Perry p.2-138]
97
98    sd.deltaH_v = [
99        47.7;
100       0;
101       0;
102       0;
103       0;
104       285.83-241.83;
105       0
106       ]; %kJ/mol
107
108   % Correlation parameters
109
110   sd.par_H_v = [
111       nan, nan, nan, nan;
112       3.1523, 0.3914, -.2289, .2309;
113       0.74905, .40406, -.317, .27343;
114       2.3215, .384, 0, 0;
115       2.131, .4056, 0, 0;
116       5.2053, 0.3199, -.212, .25795;
117       .10127, .698, -1.817, 1.447
118       ]; % [Perry p.2-150]
119
120   sd.par_shomate1 = [
121       52.02427, 49.77119, -15.37599, 1.921168, 0.189174, -53.30667, 203.8591, -45.89806; %NH3
122       28.98641, 1.853978, -9.647459, 16.63537, 0.000117, -8.671914, 226.4168, 0; %N2
123       27.67988, 51.14898, -30.64454, 6.847911, -0.157906, 71.24934, 238.6164, 82.04824; %N2O
124       23.83491, 12.58878, -1.139011, -1.497459, 0.214194, 83.35783, 237.1219, 90.29114; %NO
125       -203.606, 1523.29, -3196.413, 2474.455, 3.855326, -256.5478, -488.7163, -285.8304; %H2O(l)
126       33.066178, -11.363417, 11.432816, -2.772874, -0.158558, -9.980797, 172.707974, 0 %H2
127       ]; % Liquid water
128
129   sd.par_shomate2 = [
130       52.02427, 49.77119, -15.37599, 1.921168, 0.189174, -53.30667, 203.8591, -45.89806; %NH3
              298-1400 K
131       28.98641, 1.853978, -9.647459, 16.63537, 0.000117, -8.671914, 226.4168, 0; %N2 100 - 500 K
132       27.67988, 51.14898, -30.64454, 6.847911, -0.157906, 71.24934, 238.6164, 82.04824; %N2O 298 -
              1400 K
133       23.83491, 12.58878, -1.139011, -1.497459, 0.214194, 83.35783, 237.1219, 90.29114; %NO 298
              -1200 K
134       30.092, 6.832514, 6.793435, -2.53448, 0.082139, -250.881, 223.3967, -241.8264; %H2O(g) 500 -
              1700 K
135       33.066178, -11.363417, 11.432816, -2.772874, -0.158558, -9.980797, 172.707974, 0 %H2 298 -
              1000 K
136       ]; % Gaseous water
137
138   sd.par_henry = [
139       1.40E+03, 0;
140       5.90E-01, 4200;
141       6.40E-06, 1300;
```

```
142         2.40E-04, 2600;
143         1.90E-05, 1600;
144         nan, nan;
145         7.70E-06, 500
146         ];
147
148     sd.par_ant = [
149         5.2343, 1.2289e+03, -149.2262; % Regression of experimental data (estimations_physical_data.m)
150         nan, nan, nan;
151         nan, nan, nan;
152         nan, nan, nan;
153         nan, nan, nan;
154         4.6543, 1435.264, -64.848; % [NIST]
155         nan, nan, nan
156         ];
157
158     % Gibbs free energy of formation
159
160     sd.G_f_0([1,2,4]) = sd.deltaH_f_0([1,2,4]) - T_0 * sd.S_f_0([1,2,4]) / 1000;
161
162     %% Reaction Data
163     % Reaction description
164     nu_1 = [-157; 61; 35; 12; 2; 143; 1]; % Empiric reaction formula
165     nu_2 = [-3;1;1;0;0;3;0]; % In alkaline solution
166     nu_3 = [-4;2;0;1;0;3;0]; % In acid solution
167     % nu_4 = 0.9 * nu_2 + 0.1 * nu_3; % Weighted sum of nu_2 and nu_3
168     rxn = ["empiric", "alkaline", "acid";%, "combination"];
169
170     nu_i = [nu_1, nu_2, nu_3];%, nu_4];
171
172     deltaH_r_0 = sum(nu_i.*sd.deltaH_f_0)./-nu_i(1,:);
173     deltaH_r_0 = - 124; %kJ/mol
174     delta_R_G_0 = sum(nu_1 .* sd.G_f_0);
175
176     % Initial mixture
177     w_HA = 0.5;
178     rho_RM = 1120; %kg/m^3
179
180     p = p_0;
181     % Reactor volume (measured)
182     V = 0.1110528 * 10^-3; %m^3
183
184     if(true) % For switch 1, 2 (isothermal and adiabatic) and 3
185         V_l =  0.03e-3; %m^3
186         experiment = "Test"
187
188         % Choose from experiments:
189         %       Exp10: 0.0286429 -
190         %       Exp12: 0.0291071 -
191         %       Exp15: 0.0290179 -
192         %       Exp16: 0.0291071 -
193         %       Exp19: 0.0277679 -
194         %       Exp20: 0.0290179 -
195     end
196
197     if(false) %For switch 2 (isothermal screening)
198         V_l = linspace(0.01, 0.1, 10) * 1e-3; %m^3
199     end
200
201     V_g = V - V_l; %m^3
202
203     n_N2_t0 = V_g*p_0/(R*298.15);
204
205     n_t0 = [
206         V_l*rho_RM*w_HA/sd.mw(1);
207         zeros(size(V_l));
208         n_N2_t0;
209         zeros(size(V_l));
210         zeros(size(V_l));
211         V_l*rho_RM*(1-w_HA)/sd.mw(6);
212         zeros(size(V_l))]; %mol
213
214
215     %% Model
216
217     % 1: Phase equilibrium only
218     % 2: Complete decomposition scenario
219     % 3: Thermodynamic calculations for different reaction equations
220     switch 3
221         case 1
222             % Mode
223             % 1: Reaction conversion dependant simulation
224             % 2: Temperature dependant simulation
225             % 3: Create matrix p(T,alpha) for simulation in EXR_run.m
226             % 4: Simulation of experiment with alpha(t) from EXR_run.m
227             mode = 1;
228
229             %% Solver settings
230             x0 = [0; 0; 1; 1; 1; 0; 1];
231             lb = zeros(size(x0));
```

```
232         ub = ones(size(x0));
233
234         options = optimoptions('lsqnonlin', ...
235             'Display','iter');%, ...
236         % "PlotFcn", 'optimplotx');
237
238         switch mode
239             case [1,2]
240                 %% Experiment specification
241                 nu = nu_1;
242                 ind = (nu~=0);
243
244                 switch mode
245                     case 1
246                         T = 110 + 273.15; % K
247                         alpha = linspace(0,1,10);
248                     case 2
249                         T = linspace(0, 200, 100) + 273.15;% K
250                         alpha = 1 -  2.0485/50;
251
252                 end
253
254                 %% Phase equilibrium for specified experiment
255
256                 for i = 1:max([length(alpha), length(T)])
257
258                     switch mode
259                         case 1
260                             unk = lsqnonlin(@equil_phase, x0, lb, ub, options, alpha(i), T, V_g,
                                    n_t0, R, nu, sd);
261                             [funval, eq] = equil_phase(unk, alpha(i), T, V_g, n_t0, R, nu, sd);
262                         case 2
263                             unk = lsqnonlin(@equil_phase, x0, lb, ub, options, alpha, T(i), V_g,
                                    n_t0, R, nu, sd);
264                             [funval, eq] = equil_phase(unk, alpha, T(i), V_g, n_t0, R, nu, sd);
265
266                     end
267                     n_i(:,i) = eq.n;
268                     n_l_i(:,i) = eq.n_l;
269                     n_g_i(:,i) = eq.n_g;
270                     x_i(:,i) = eq.x;
271                     y_i(:,i) = eq.y;
272                     p_tot_i(i) = eq.p_tot;
273                     P_g_i(:,i) = eq.P_g;
274                     p_i_i(:,i) = eq.p_i;
275
276
277                 end
278
279                 %% Plots
280                 switch mode
281                     case 1
282                         % Total pressure
283                         figure
284                         plot(alpha, p_tot_i*10^-5, "o-", "LineWidth", 1)
285                         xlabel("\alpha");ylabel("p_{tot} / bar");
286                         grid on
287                         title("p_{tot}")
288                         set(gca, "FontName", "LM Roman 12")
289
290                         % x_i
291                         figure
292                         plot(alpha, x_i(ind,:), "LineWidth", 1);
293                         xlabel("\alpha");ylabel("x_i");
294                         grid on
295                         title("x_i^{(l)}")
296                         legend(subs(ind), "Location", "best")
297                         set(gca, "FontName", "LM Roman 12")
298
299                         % y_i
300                         figure
301                         plot(alpha, y_i(ind,:), "LineWidth", 1);
302                         xlabel("\alpha");ylabel("y_i");
303                         grid on
304                         title("y_i^{(g)}")
305                         legend(subs(ind), "Location", "best")
306                         set(gca, "FontName", "LM Roman 12")
307
308                         % n_i
309                         figure
310                         plot(alpha, n_i(ind,:), "LineWidth", 1);
311                         xlabel("\alpha");ylabel("n_{i}^{(l,g)} / mol");
312                         grid on
313                         title("n_i^{l,g}")
314                         legend(subs(ind), "Location", "best")
315                         set(gca, "FontName", "LM Roman 12")
316
317                         % n_i^l
318                         figure
319                         plot(alpha, n_l_i(ind,:), "LineWidth", 1);
```

```
320                     xlabel("\alpha");ylabel("n_i^{(l)} / mol");
321                     grid on
322                     title("n_i^{(l)}")
323                     legend(subs(ind), "Location", "best")
324                     set(gca, "FontName", "LM Roman 12")
325
326                     % n_i^g
327                     figure
328                     plot(alpha, n_g_i(ind,:), "LineWidth", 1);
329                     xlabel("\alpha");ylabel("n_{i}^{(g)} / mol");
330                     grid on
331                     title("n_i^{(g)}")
332                     legend(subs(ind), "Location", "best")
333                     set(gca, "FontName", "LM Roman 12")
334
335                     % P_i^g
336                     figure
337                     plot(alpha, P_g_i(ind,:), "LineWidth", 1);
338                     xlabel("\alpha");ylabel("P_i^{(g)}");
339                     grid on
340                     title("P_i^{(g)}")
341                     legend(subs(ind), "Location", "best")
342                     set(gca, "FontName", "LM Roman 12")
343
344                     % p_i
345                     figure
346                     plot(alpha, p_i_i(ind,:)*10^-5, "LineWidth", 1);
347                     xlabel("\alpha");ylabel("p_i / bar");
348                     grid on
349                     title("p_i")
350                     legend(subs(ind), "Location", "best")
351                     set(gca, "FontName", "LM Roman 12")
352
353                     % Display reaction specification
354                     disp(append("Experiment temperature: T = ", num2str(T-273.15), ...
355                         " degC; Start volume: V_l = ", num2str(V_l*1e6, 3), " mL"));
356
357                 case 2
358                     % Total pressure
359                     figure
360                     plot(T-273.15, p_tot_i*10^-5, "o-", "LineWidth", 1)
361                     xlabel("T / degC");ylabel("p_{tot} / bar");
362                     grid on
363                     title("p_{tot}")
364                     set(gca, "FontName", "LM Roman 12")
365
366                     % x_i
367                     figure
368                     plot(T-273.15, x_i(ind,:), "LineWidth", 1);
369                     xlabel("T / degC");ylabel("x_i");
370                     grid on
371                     title("x_i^{(l)}")
372                     legend(subs(ind), "Location", "best")
373                     set(gca, "FontName", "LM Roman 12")
374
375                     % y_i
376                     figure
377                     plot(T-273.15, y_i(ind,:), "LineWidth", 1);
378                     xlabel("T / degC");ylabel("y_i");
379                     grid on
380                     title("y_i^{(g)}")
381                     legend(subs(ind), "Location", "best")
382                     set(gca, "FontName", "LM Roman 12")
383
384                     % n_i
385                     figure
386                     plot(T-273.15, n_i(ind,:), "LineWidth", 1);
387                     xlabel("T / degC");ylabel("n_{i}^{(l,g)} / mol");
388                     grid on
389                     title("n_i^{l,g}")
390                     legend(subs(ind), "Location", "best")
391                     set(gca, "FontName", "LM Roman 12")
392
393                     % n_i^l
394                     figure
395                     plot(T-273.15, n_l_i(ind,:), "LineWidth", 1);
396                     xlabel("T / degC");ylabel("n_i^{(l)} / mol");
397                     grid on
398                     title("n_i^{(l)}")
399                     legend(subs(ind), "Location", "best")
400                     set(gca, "FontName", "LM Roman 12")
401
402                     % n_i^g
403                     figure
404                     plot(T-273.15, n_g_i(ind,:), "LineWidth", 1);
405                     xlabel("T / degC");ylabel("n_{i}^{(g)} / mol");
406                     grid on
407                     title("n_i^{(g)}")
408                     legend(subs(ind), "Location", "best")
409                     set(gca, "FontName", "LM Roman 12")
```

```
410
411                         % P_i^g
412                         figure
413                         plot(T-273.15, P_g_i(ind,:), "LineWidth", 1);
414                         xlabel("T / degC");ylabel("P_i^{(g)}");
415                         grid on
416                         title("P_i^{(g)}")
417                         legend(subs(ind), "Location", "best")
418                         set(gca, "FontName", "LM Roman 12")
419
420                         % p_i
421                         figure
422                         plot(T-273.15, p_i_i(ind,:)*10^-5, "LineWidth", 1);
423                         xlabel("T / degC");ylabel("p_i / bar");
424                         grid on
425                         title("p_i")
426                         legend(subs(ind), "Location", "best")
427                         set(gca, "FontName", "LM Roman 12")
428
429                         % Display reaction specification
430                         disp(append("Experiment reaction conversion: X = ", num2str(alpha), ...
431                             "; Start volume: V_l = ", num2str(V_l*1e6, 3), " mL"));
432
433
434                     end
435
436
437                 case 3
438
439                     T = linspace(0, 230, 50) + 273.15;% K
440                     alpha = linspace(0, 1, 50);
441
442                     for r = 1:length(rxn)
443
444                         for i = 1:length(T)
445                             for ii = 1:length(alpha)
446
447                                 unk = lsqnonlin(@equil_phase, x0, lb, ub, options, alpha(ii), T(i),
                                        V_g, n_t0, R, nu_i(:,r), sd);
448                                 [funval, eq] = equil_phase(unk, alpha(ii), T(i), V_g, n_t0, R,
                                        nu_i(:,r), sd);
449                                 p_tot_i(ii,i) = eq.p_tot;
450
451                             end
452                         end
453
454
455
456                         %% Export matrix p(T, alpha) for simulation in EXR_run.m
457                         Model(r).rxn = rxn(r);
458                         Model(r).T = T - 273.15; % degC
459                         Model(r).alpha = alpha;
460                         Model(r).p_tot = p_tot_i * 1e-5; % bar
461
462                         figure
463                         surf(Model(r).T, Model(r).alpha, Model(r).p_tot);
464                         xlabel("T / degC"); ylabel("\alpha"); zlabel("p_{tot} / bar");
465                         title(append("Resulting pressure (", rxn(r), " model)"));
466                     end
467
468                     %                   save(append("EXR/Data/", experiment, "_Sim.mat"), "Model");
469                 case 4
470
471                     load(append("EXR/Data/", experiment, "_Sim.mat"));
472
473                     options = optimoptions('lsqnonlin', ...
474                         'Display','off');
475
476
477
478                     for r = 1:length(rxn)
479
480                         T = Sim(r).T + 273.15;% K
481                         alpha = Sim(r).alpha_corr';
482
483                         for i = 1:length(alpha)
484                             unk = lsqnonlin(@equil_phase, x0, lb, ub, options, alpha(i), T(i), V_g,
                                    n_t0, R, nu_i(:,r), sd);
485                             [funval, eq] = equil_phase(unk, alpha(i), T(i), V_g, n_t0, R, nu_i(:,r),
                                    sd);
486
487
488                             Sim(r).n_i(:,i) = eq.n;
489                             Sim(r).n_l_i(:,i) = eq.n_l;
490                             Sim(r).n_g_i(:,i) = eq.n_g;
491                             Sim(r).x_i(:,i) = eq.x;
492                             Sim(r).y_i(:,i) = eq.y;
493                             Sim(r).p_tot(i) = eq.p_tot * 1e-5;
494                             Sim(r).P_g_i(:,i) = eq.P_g;
495                             Sim(r).p_i_i(:,i) = eq.p_i;
```

```
496
497                         disp(append(num2str(((r-1)*length(alpha)+i)/(length(rxn)*length(alpha))*100),
                                " %"));
498                     end
499
500                 end
501                 %                     save(append("EXR/Data/", experiment, "_Sim.mat"), "Model",
                        "Sim");
502
503                 %% Plots
504                 % Total pressure
505                 figure; hold on
506                 for r = 1:length(rxn)
507                     plot(Sim(r).t/3600, Sim(r).p_tot, "LineWidth", 1)
508                 end
509                 xlabel("t / h");ylabel("p_{tot} / bar");
510                 legend(rxn)
511                 grid on
512                 title("p_{tot}")
513                 set(gca, "FontName", "LM Roman 12")
514
515                 % x_i
516                 figure
517                 plot(alpha, x_i(ind,:), "LineWidth", 1);
518                 xlabel("\alpha");ylabel("x_i");
519                 grid on
520                 title("x_i^{(l)}")
521                 legend(subs(ind), "Location", "best")
522                 set(gca, "FontName", "LM Roman 12")
523
524                 % y_i
525                 figure
526                 plot(alpha, y_i(ind,:), "LineWidth", 1);
527                 xlabel("\alpha");ylabel("y_i");
528                 grid on
529                 title("y_i^{(g)}")
530                 legend(subs(ind), "Location", "best")
531                 set(gca, "FontName", "LM Roman 12")
532
533                 % n_i
534                 figure
535                 plot(alpha, n_i(ind,:), "LineWidth", 1);
536                 xlabel("\alpha");ylabel("n_{i}^{(l,g)} / mol");
537                 grid on
538                 title("n_i^{l,g}")
539                 legend(subs(ind), "Location", "best")
540                 set(gca, "FontName", "LM Roman 12")
541
542                 % n_i^l
543                 figure
544                 plot(alpha, n_l_i(ind,:), "LineWidth", 1);
545                 xlabel("\alpha");ylabel("n_i^{(l)} / mol");
546                 grid on
547                 title("n_i^{(l)}")
548                 legend(subs(ind), "Location", "best")
549                 set(gca, "FontName", "LM Roman 12")
550
551                 % n_i^g
552                 figure
553                 plot(alpha, n_g_i(ind,:), "LineWidth", 1);
554                 xlabel("\alpha");ylabel("n_{i}^{(g)} / mol");
555                 grid on
556                 title("n_i^{(g)}")
557                 legend(subs(ind), "Location", "best")
558                 set(gca, "FontName", "LM Roman 12")
559
560                 % P_i^g
561                 figure
562                 plot(alpha, P_g_i(ind,:), "LineWidth", 1);
563                 xlabel("\alpha");ylabel("P_i^{(g)}");
564                 grid on
565                 title("P_i^{(g)}")
566                 legend(subs(ind), "Location", "best")
567                 set(gca, "FontName", "LM Roman 12")
568
569                 % p_i
570                 figure
571                 plot(alpha, p_i_i(ind,:)*10^-5, "LineWidth", 1);
572                 xlabel("\alpha");ylabel("p_i / bar");
573                 grid on
574                 title("p_i")
575                 legend(subs(ind), "Location", "best")
576                 set(gca, "FontName", "LM Roman 12")
577
578                 % Display reaction specification
579                 disp(append("Experiment temperature: T = ", num2str(T-273.15), ...
580                     " degC; Start volume: V_l = ", num2str(V_l*1e6, 3), " mL"));
581
582
583
```

```
584             end
585
586
587
588         case 2
589             %% Choose reaction
590             nu = nu_3;
591             T_S = 150+273.15; %K start temperature
592             T = linspace(60, 250, 10) + 273.15;%T_S;%
593
594             %% Isothermal decomposition
595
596             x0 = [0;0;1;1;1;0;1];
597             lb = zeros(size(x0));
598             ub = ones(size(x0));
599
600             options = optimoptions('lsqnonlin', ...
601                 'Display','iter', ...
602                 "PlotFcn", 'optimplotx', ...
603                 "MaxFunctionEvaluations", 5*10^4, ...
604                 "MaxIterations", 10^3);
605
606             % Resulting pressure as a function of temperature with 0% and 100% reaction
607             % conversion
608             alpha=[0,1];
609
610             for i = 1:length(T)
611                 for ii = 1:length(V_l)
612
613                     % Gas decomposition ----------------------------------------
614
615                     % Substance in gas phase n_g_t0 at specified temperature
616                     P_g = lsqnonlin(@equil_phase, x0, lb, ub, options, alpha(1), T(i), V_g(ii),
                            n_t0(:,ii), R, nu, sd);
617                     [funval, eq] = equil_phase(P_g, alpha(1), T(i), V_g(ii), n_t0(:,ii), R, nu, sd);
618                     n_g_t0 = eq.n_g;
619
620                     % Pressure after decomposition of n_g_t0
621                     P_g = ub; % all molecules are in the gaseous phase
622                     [funval, eq] = equil_phase(P_g, alpha(2), T(i), V_g(ii), n_g_t0, R, nu, sd);
623                     p_tot_g(ii) = sum(eq.n * R * T(i) / V_g(ii));
624
625                     % Liquid decomposition -------------------------------------
626
627                     % Pressure after decomposition of n_t0
628                     P_g = lsqnonlin(@equil_phase, x0, lb, ub, options, alpha(2), T(i), V_g(ii),
                            n_t0(:,ii), R, nu, sd);
629                     [funval, eq] = equil_phase(P_g, alpha(2), T(i), V_g(ii), n_t0(:,ii), R, nu, sd);
630                     p_tot_l(ii) = eq.p_tot;
631
632                 end
633                 m_p_tot_g(i, :) = p_tot_g;
634                 m_p_tot_l(i, :) = p_tot_l;
635             end
636             close all;
637
638             if(~all(size(V_l)==1)) % Isothermal screening
639                 figure
640                 contourf(V_l * 1e6, T-273.15, m_p_tot_g * 1e-5);
641                 xlabel("Filling Volume / mL"); ylabel("T / degC"); zlabel("p_{tot} / bar");
642                 %         colormap([0 1 0; 1 0 0]);
643                 %         set(gca, 'clim', [299 301]); % or any centered-on-threshold limits
644                 title("Isothermal gas phase decomposition")
645                 grid on
646                 set(gca, "FontName", "LM Roman 12")
647
648                 figure
649                 contourf(V_l * 1e6, T-273.15, m_p_tot_l * 1e-5, ':', 'Levels', 10, 'LineWidth', 0.5);
650                 set(gca, 'Xlim', [20, max(V_l * 1e6)], 'Ylim', [min(T-273.15), 200]);
651                 xlabel("Filling volume / mL"); ylabel("T / degC"); zlabel("p_{tot} / bar");
652                 colormap([0 1 0; 1 0 0]);
653                 set(gca, 'clim', [269 271]); % or any centered-on-0 limits
654                 title("Isothermal liquid phase decomposition")
655                 grid on
656                 labels = {'<270bar','>270bar'};
657                 lcolorbar(labels,'fontweight','bold', 'FontName', 'LM Roman 12');
658                 set(gca, "FontName", "LM Roman 12")
659
660
661                 figure
662                 surf(V_l * 1e6, T-273.15, m_p_tot_g * 1e-5);
663                 xlabel("V / mL"); ylabel("T / degC"); zlabel("p_{tot} / bar");
664                 title("Isothermal gas phase decomposition")
665
666                 figure
667                 surf(V_l * 1e6, T-273.15, m_p_tot_l * 1e-5);
668                 xlabel("V / mL"); ylabel("T / degC"); zlabel("p_{tot} / bar");
669                 title("Isothermal liquid phase decomposition")
670             end
671
```

```
672             if(all(size(V_l)==1)) % Adiabatic for defined experiment
673
674                 %% Adiabatic decomposition
675                 p_E_iso_g = interp1(T, m_p_tot_g, T_S);
676                 p_E_iso_l = interp1(T, m_p_tot_l, T_S);
677
678                 % Antiderivative of heat capacity
679                 Cv_int = @(T) [(sd.Cp_0_poly(1).*T.^4./5 + sd.Cp_0_poly(2).*T.^3./4 +
                        sd.Cp_0_poly(3).*T.^2./3 + sd.Cp_0_poly(4).*T./2 + sd.Cp_0_poly(5) - R) .* T;...
680                     sd.par_shomate2(:,4).*T.^4./4000000000+sd.par_shomate2(:,3).*T.^3./3000000+sd.par_shomate2(:,2).*T.^
                            - R.*T];
681
682                 alpha = [0,1]; % Reaction conversion S->E
683
684                 % Complete liquid decomposition ----------------------------------
685
686                 % End temperature after reaction (constant internal energy)
687                 [T_E_l, dU_V] = fsolve(@dU_V_ad, T_S, [], T_S, R, T_0, alpha, nu, n_t0, sd, Cv_int);
688
689                 % End state
690                 p_E_l = p_E_iso_l * T_E_l / T_S;
691
692                 % Complete gas decomposition -------------------------------------
693
694                 % End temperature after reaction (constant internal energy)
695                 % Because there are only very few atoms in the gas phase, the
696                 % solver needs to be more sensible
697                 options = optimoptions('fsolve',...
698                     'OptimalityTolerance', 1e-10,...
699                     'FunctionTolerance', 1e-10);
700                 [T_E_g, dU_V] = fsolve(@dU_V_ad, T_S, options, T_S, R, T_0, ...
701                     alpha, nu, n_g_t0, sd, Cv_int);
702
703                 % End state
704                 p_E_g = p_E_iso_g * T_E_g / T_S;
705
706                 %% Display results
707                 disp(append("Start temperature: T_S = ", num2str(T_S-273.15), ...
708                     " degC; Start volume: V_l = ", num2str(V_l*1e6, 3), " mL"));
709                 disp(append("Conditions after isothermal decomposition of gas phase: T = ", ...
710                     num2str(T_S-273.15, 3), " degC, p = ", num2str(p_E_iso_g*10^-5, 4), " bar"));
711                 disp(append("Conditions after isothermal decomposition of liquid phase: T = ", ...
712                     num2str(T_S-273.15, 4), " degC, p = ", num2str(p_E_iso_l*10^-5, 4), " bar"));
713                 disp(append("Conditions after adiabatic decomposition of gas phase: T = ", ...
714                     num2str(T_E_g-273.15, 5), " degC, p = ", num2str(p_E_g*10^-5, 4), " bar"));
715                 disp(append("Conditions after adiabatic decomposition of liquid phase: T = ", ...
716                     num2str(T_E_l-273.15, 4), " degC, p = ", num2str(p_E_l*10^-5, 4), " bar"));
717
718             end
719
720         case 3
721
722             if(true) % For end pressure of distinct experiment
723                 T = 110 +273.15; % K
724                 alpha = 1-0.833/50;
725             end
726
727             if(false) % For condition screening (risk analysis)
728                 T = linspace(25, 200, 10) + 273.15; % K
729                 alpha = linspace(0,1, 10);%logspace(-10^-13,0,100);
730             end
731
732             % Antiderivative of heat capacity
733             Cv_int = @(T) [(sd.Cp_0_poly(1).*T.^4./5 + sd.Cp_0_poly(2).*T.^3./4 +
                    sd.Cp_0_poly(3).*T.^2./3 + sd.Cp_0_poly(4).*T./2 + sd.Cp_0_poly(5) - R) .* T;...
734                 sd.par_shomate2(:,4).*T.^4./4000000000+sd.par_shomate2(:,3).*T.^3./3000000+sd.par_shomate2(:,2).*T.^2./2
                        - R.*T];
735             % Antiderivative of heat capacity/T
736             Cv_T_int = @(T) [sd.Cp_0_poly(1).*T.^4./4 + sd.Cp_0_poly(2).*T.^3./3 +
                    sd.Cp_0_poly(3).*T.^2./2 + sd.Cp_0_poly(4).*T + (sd.Cp_0_poly(5)-R).*log(T);...
737                 sd.par_shomate2(:,4).*T.^4./(3*10^9) + sd.par_shomate2(:,3).*T.^2./(2*10^6) +
                        sd.par_shomate2(:,2).*T./1000 - sd.par_shomate2(:,5)*1000^2./(2.*T.^2) +
                        (sd.par_shomate2(:,1)-R).*log(T)];
738
739             close all;
740
741             for r = 1:size(nu_i,2)
742
743                 for i = 1:length(T)
744                     % Optimization (not yet prepared)
745                     x0 = 1;
746                     lb = zeros(size(x0));
747                     ub = ones(size(x0));
748
749                     if false
750                         unk = lsqnonlin(@equil_reac, x0, lb, ub, [], T(i), V_g, n_t0, R, T_0, p_0,
                                nu_i(:,r), sd, Cv_int, Cv_T_int)
751                         alpha = unk;
752                     end
753
```

```
754                 % Screening
755                 for s = 1:length(alpha)
756
757                     [funval, req, eq] = equil_reac(alpha(s), T(i), V_g, n_t0, R, T_0, p_0,
                            nu_i(:,r), sd, Cv_int, Cv_T_int);
758
759                     p_tot_i(:,s) = eq.p_tot;
760                     chemeq_l_i(s) = req.chemeq;
761                     K_V_s(s) = req.K_V;
762                     % Ky_l_i(s) = req.Ky;
763                     G_sys_s(s) = req.G_sys;
764                     A_sys_s(s) = req.A_sys;
765
766                 end
767
768                 matchemeq(i,:) = chemeq_l_i;
769                 matK_V(i,:) = K_V_s;
770                 p_tot_si(i,:) = p_tot_i;
771                 matG_sys(i,:) = G_sys_s;
772                 matA_sys(i,:) = A_sys_s;
773
774             end
775             if(all(size(T)==[1, 1]) && all(size(alpha)==[1, 1]) && all(size(V)==[1, 1]))
776                 p_end(r) = p_tot_si;
777             end
778
779             if(any(size(T)>[1, 1]) && any(size(alpha)>[1, 1]) && any(size(V)==[1, 1]))
780
781                 % Chemical equilibrium
782                 figure
783                 surf(alpha, T-273.15, matchemeq)
784                 xlabel(append("\alpha R", num2str(r,1)));
785                 ylabel("T / degC");
786                 zlabel("sum(\nu_i\mu_i) / kJ/mol");
787                 title("Chemical equilibrium")
788                 set(gca, "FontName", "LM Roman 12")
789
790                 % K_V
791                 figure
792                 surf(alpha, T-273.15, matK_V);
793                 xlabel(append("\alpha R", num2str(r,1)));
794                 ylabel("T / degC");
795                 zlabel("K_V");
796                 title(append("K_V of reaction ", num2str(r)));
797                 set(gca, "FontName", "LM Roman 12", 'Ydir', 'reverse');
798                 view(45,45);
799                 % set(h1, 'Ydir', 'reverse')
800
801                 % % Contour K_V (limit 1)
802                 % figure
803                 % contourf(alpha, T-273.15, matK_V);
804                 % xlabel(append("alpha_", num2str(r,1)));
805                 % ylabel("T / degC");
806                 % zlabel("K_V");
807                 % colormap([1 1 0; 0 1 0]);
808                 % set(gca, 'clim', [0.99 1.01]); % or any centered-on-threshold limits
809                 % title(append("K_V of reaction ", num2str(r)))
810                 % grid on
811                 % labels = {'<1','>1'};
812                 % lcolorbar(labels,'fontweight','bold', 'FontName', 'LM Roman 12');
813                 % set(gca, "FontName", "LM Roman 12")
814
815                 % % K_y
816                 % figure
817                 % surf(alpha, T-273.15, matKy)
818                 % xlabel(append("alpha_", num2str(r,1)));
819                 % ylabel("T / degC");
820                 % zlabel("Ky^{(l)}");
821                 % title("Ky")
822                 % set(gca, "FontName", "LM Roman 12")
823
824                 % p_tot
825                 figure
826                 surf(alpha, T-273.15, p_tot_si*10^-5)
827                 xlabel(append("\alpha R", num2str(r,1)));
828                 ylabel("T / degC");
829                 zlabel("p_{tot} / bar");
830                 title("p_{tot}")
831                 set(gca, "FontName", "LM Roman 12")
832
833                 % % G_sys
834                 % figure
835                 % surf(alpha, T-273.15, matG_i)
836                 % xlabel(append("alpha_", num2str(r,1)));
837                 % ylabel("T / degC");
838                 % zlabel("G^{(l)} / kJ");
839                 % set(gca, "FontName", "LM Roman 12")
840
841                 % A_sys
842                 figure
```

```
843                     surf(alpha, T-273.15, matA_sys)
844                     xlabel(append("\alpha R", num2str(r,1)));
845                     ylabel("T / degC");
846                     zlabel("A_{sys} / kJ");
847                     title("Helmholtz energy of the system");
848                     set(gca, "FontName", "LM Roman 12")
849                 end
850
851             end
852
853             % Display experiment specification
854             disp(append("Start volume: V_l = ", num2str(V_l*1e6, 3), " mL"));
855
856             % Calculate end preassure of specified experiment
857             if(all(size(T)==[1, 1]) && all(size(alpha)==[1, 1]) && all(size(V)==[1, 1]))
858                 figure
859                 hold on
860                 bar(categorical({'empiric','alkaline', 'acid'}), p_end*1e-5)
861                 ylabel('p / bar')
862                 grid on
863                 if true % Compare with experiment end pressure
864                     h = plot(get(gca, 'Xlim'), [34.7123, 34.7123], ':r', 'LineWidth', 2);
865                     legend(h, 'p_{end} from EXR experiment')
866                 end
867                 title(experiment);
868                 set(gca, 'FontName', 'LM Roman 12');
869
870                 % Display experiment specification
871                 disp(append(experiment, " - Start volume: V_l = ", num2str(V_l*1e6, 3), " mL;
                        Temperature: T = ", num2str(T-273.15, 3), " degC; Reaction conversion: \alpha =
                        ", num2str(alpha, 3)));
872
873
874             end
875
876     end
877     toc
878
879     % ------------------------------------------------------------------------
```

A.19 equil_reac.m

Function to calculate the reaction equilibrium of the EXR.

```
 1   % ------------------------------------equil_reac.m-----------------------------------
 2   % Florian Walthert
 3   % 04.07.20
 4
 5   % Function to calculate the reaction equilibrium of the EXR
 6   function [funval, req, eq] = equil_reac(unk, T, V_g, n_t0, R, T_0, p_0, nu, sd, Cv_int, Cv_T_int)
 7
 8   alpha = unk;
 9   % Phase equilibrium ------------------------------------------------------------
10   x0 = [0.01;0.01;.99;.99;.99;0.01;.99];
11   lb = zeros(size(x0));
12   ub = ones(size(x0));
13
14   options = [];
15
16   P_g = lsqnonlin(@equil_phase, x0, lb, ub, options, alpha, T, V_g, n_t0, R, nu, sd);
17   [funval, eq] = equil_phase(P_g, alpha, T, V_g, n_t0, R, nu, sd);
18
19   % Gibb free energy -------------------------------------------------------------
20   % Thermodynamic properities of the pure substances
21   Cp = [sd.Cp_0(1); shomate(T, sd.par_shomate1)]; % short-cut
22   deltaH_f = sd.deltaH_f_0 + Cp*(T - T_0)/1000; % [kJ/mol]
23   S_f = sd.S_f_0 + Cp*log(T/T_0); % [J/(mol*K)]
24   G_f = deltaH_f - T*(S_f)/1000; % [kJ/mol]
25
26   % Thermodynamic properties of the mixture
27   mu_G = G_f + R*T*log(eq.p_i/p_0)/1000; % [kJ/mol] % baehr p.267 [Gmehling p.156 4.54]
28   if(alpha~=0 && alpha~=1)
29       mu_G(eq.p_i==0) = 0;
30   end
31   mu_G([1,6]) = G_f([1,6])+ R * T * log(eq.x([1,6]))/1000; % [kJ/mol] [Gmehling p.161 4.84]
32
33   req.G_sys = sum(mu_G.*eq.n); % kJ
34   req.K_p = exp(-sum(G_f.*nu) / (R * T));
35
36   % Helmholtz free energy --------------------------------------------------------
37
38   % Thermodynamic properities of the pure substances
39   U_f = sd.deltaH_f_0 - R * T_0 / 1000 + Cv_int(T) / 1000 - Cv_int(T_0) / 1000; % [Gmehling p.334]
```

```
40  S_f = sd.S_f_0 + (Cv_T_int(T) - Cv_T_int(T_0)); % [J/(mol*K)]
41  A_f = U_f - T*(S_f)/1000; % [kJ/mol]
42
43  % Thermodynamic properties of the mixture
44  % Chemical potential for gaseous components
45  mu_A = A_f + R*T*log(eq.p_i/p_0)/1000; % [kJ/mol] % [Baehr p.267] [Gmehling p.156 4.54]
46  if(alpha~=0 && alpha~=1)
47      mu_A(eq.p_i==0) = 0;
48  end
49  % Chemical potential for hydroxylamine and water
50  mu_A([1,6]) = G_f([1,6])+ R * T * log(eq.x([1,6]))/1000; % [kJ/mol] [Gmehling p.161 4.84]
51
52  req.A_sys = sum(mu_A.*eq.n);
53  req.chemeq = sum(mu_A.*nu); % kJ/mol
54  req.K_V = exp(-req.chemeq ./ (R.*T));
55
56
57  funval = [...
58      1-req.K_V
59      ];
60
61  end
62
63  % ------------------------------------------------------------------------
```

A.20 equil_phase.m

Function to calculate the phase equilibrium of the EXR.

```
1   % ------------------------------------equil_phase.m-----------------------------------
2   % Florian Walthert
3   % 04.07.20
4
5   % Function to calculate the phase equilibrium of the EXR
6   function [funval, eq] = equil_phase(unk, alpha, T, V_g, n_t0, R, nu, sd)
7
8   % Partition coefficients for the gas phase
9   eq.P_g = unk;
10
11  % Reaction conversion
12  xi = -1/nu(1)*alpha*n_t0(1); %mol
13  eq.n = n_t0 + nu.*xi; %mol
14
15  % Phase balance
16  eq.n_g = eq.n .* eq.P_g;
17  eq.n_l = eq.n - eq.n_g;
18
19  eq.y = eq.n_g ./ sum(eq.n_g);
20  eq.x = eq.n_l ./ sum(eq.n_l);
21
22  % Partial pressures
23  eq.p_i = henry(eq.x, sd.par_henry, T);
24  eq.p_i([1,6],:) = Antoine(sd.par_ant([1,6],:), T)*10^5 .* eq.x([1,6],:);
25
26  % Ideal gas law (no solubility) gives best results for N2, NO and H2
27  eq.p_i([3,5,7],:) = eq.n([3,5,7],:) * R * T / V_g;
28  eq.n_g([3,5,7],:) = eq.n([3,5,7],:);
29  eq.n_l([3,5,7],:) = 0;
30  eq.x([3,5,7],:) = 0;
31  eq.P_g([3,5,7],:) = 1;
32
33  eq.p_tot = sum(eq.p_i);
34
35  y_targ = eq.p_i ./ eq.p_tot;
36
37  funval = [
38      eq.y-y_targ;
39      ];
40
41  end
42
43  % ------------------------------------------------------------------------
```

A.21 EXR_NH3_estimation.m

Script to model the ammonia phase equilibrium in the EXR.

```
% ----------------------------------HA_EXR_estimation.m-----------------------------------
% Florian Walthert
% 04.07.20

% Script to model the ammonia phase equilibrium in the EXR
clear all;close all;

subs = ["NH_2OH"; "NH_3"; "N_2"; "N_2O"; "NO"; "H_2O"; "H_2"];

%% Basic Data
R = 8.314; % J/mol/K
T_0 = 293.15; % K (NIST)
p_0 = 1.013 * 10^5; % Pa
% format long

%% Substance Data
% General

sd.mw = [
    33.03,
    17,
    28,
    44,
    30,
    18,
    2
    ] * 10^-3; %kg/mol

% Thermodynamic

sd.G_f_0 = [
    nan, %NH2OH [kJ/mol]
    nan, %NH3 [kJ/mol]
    -57.128, %N2 [kJ/mol]
    nan, %N2O [kJ/mol]
    27.41, %NO [kJ/mol]
    298.13, %H2O [kJ/mol]
    -38.962 %H2 [kJ/mol]
    ];

sd.Cp_0 = [
    49, % [J/(mol*K)] Estimation
    nan, % [J/(mol*K)]
    28.0134*1.0396, % [J/(mol*K)]
    38.6, % [J/(mol*K)]
    30.0061*0.9965, % [J/(mol*K)]
    18.0153*1.8646, % [J/(mol*K)]
    2.01588*14.304 % [J/(mol*K)]
    ];

sd.Cp_0_poly = [
    -4.010331963494 13e-11, 1.575625571453 88e-07,...
    -0.000242012285748035, 0.197695530171796, 5.333919080270 44
    ];

sd.deltaH_f_0 = [
    -106.7, % [kJ/mol]
    -45.9,% -45.9-35.34 % [kJ/mol] g / aq [Vanderzee]
    0, % [kJ/mol]
    82.05, % [kJ/mol]
    90.25, % [kJ/mol]
    -285.83,%-241.83,% [kJ/mol]
    0 % [kJ/mol]
    ];

sd.S_f_0 = [
    236; % [J/(mol*K)] [Giguere] %calculated from deltaG: 236;
    192.77, % [J/(mol*K)] g/ aq [Vanderzee]
    191.609, % [J/(mol*K)]
    219.96, % [J/(mol*K)]
    210.76, % [J/(mol*K)]
    69.95;%188.835, % [J/(mol*K)] liquid;gas
    130.680 % [J/(mol*K)]
    ];

sd.T_c = [
    271+273.5;
    405.65;
    126.2;
    309.57;
    180.15;
    647.096;
    33.19
    ]; % [Perry p.2-138]

sd.deltaH_v = [
    47.7;
    0;
    0;
```

```
90        0;
91        0;
92        285.83-241.83;
93        0
94        ]; %kJ/mol
95
96
97    % Correlation parameters
98
99    sd.par_H_v = [
100       nan, nan, nan, nan;
101       3.1523, 0.3914, -.2289, .2309;
102       0.74905, .40406, -.317, .27343;
103       2.3215, .384, 0, 0;
104       2.131, .4056, 0, 0;
105       5.2053, 0.3199, -.212, .25795;
106       .10127, .698, -1.817, 1.447
107       ]; % [Perry p.2-150]
108
109   sd.par_shomate1 = [
110       52.02427, 49.77119, -15.37599, 1.921168, 0.189174, -53.30667, 203.8591, -45.89806; %NH3
111       28.98641, 1.853978, -9.647459, 16.63537, 0.000117, -8.671914, 226.4168, 0; %N2
112       27.67988, 51.14898, -30.64454, 6.847911, -0.157906, 71.24934, 238.6164, 82.04824; %N2O
113       23.83491, 12.58878, -1.139011, -1.497459, 0.214194, 83.35783, 237.1219, 90.29114; %NO
114       -203.606, 1523.29, -3196.413, 2474.455, 3.855326, -256.5478, -488.7163, -285.8304; %H2O(l)
115       33.066178, -11.363417, 11.432816, -2.772874, -0.158558, -9.980797, 172.707974, 0 %H2
116       ]; % Liquid water
117
118   sd.par_shomate2 = [
119       52.02427, 49.77119, -15.37599, 1.921168, 0.189174, -53.30667, 203.8591, -45.89806; %NH3
              298-1400 K
120       28.98641, 1.853978, -9.647459, 16.63537, 0.000117, -8.671914, 226.4168, 0; %N2 100 - 500 K
121       27.67988, 51.14898, -30.64454, 6.847911, -0.157906, 71.24934, 238.6164, 82.04824; %N2O 298 -
              1400 K
122       23.83491, 12.58878, -1.139011, -1.497459, 0.214194, 83.35783, 237.1219, 90.29114; %NO 298
              -1200 K
123       30.092, 6.832514, 6.793435, -2.53448, 0.082139, -250.881, 223.3967, -241.8264; %H2O(g) 500 -
              1700 K
124       33.066178, -11.363417, 11.432816, -2.772874, -0.158558, -9.980797, 172.707974, 0 %H2 298 -
              1000 K
125       ]; % Gaseous water
126
127   sd.par_henry = [
128       1.40E+03, 0;
129       5.90E-01, 4200;
130       6.40E-06, 1300;
131       2.40E-04, 2600;
132       1.90E-05, 1600;
133       nan, nan;
134       7.70E-06, 500
135       ];
136
137   sd.par_ant = [
138       5.2343, 1.2289e+03, -149.2262; % Regression of experimental data (estimations_physical_data.m)
139       nan, nan, nan;
140       nan, nan, nan;
141       nan, nan, nan;
142       nan, nan, nan;
143       4.6543, 1435.264, -64.848; % [NIST]
144       nan, nan, nan
145       ];
146
147   % Gibbs free energy of formation
148
149   sd.G_f_0([1,2,4]) = sd.deltaH_f_0([1,2,4]) - T_0 * sd.S_f_0([1,2,4]) / 1000;
150
151   %% Reaction Data
152   % Reaction description
153   nu_1 = [-157; 61; 35; 12; 2; 143; 1]; % Empiric reaction formula
154
155   nu_i = [nu_1, nu_2, nu_3, nu_4];
156
157   deltaH_r_0 = sum(nu_1.*sd.deltaH_f_0)/-nu_1(1);
158   deltaH_r_0 = - 124; %kJ/mol
159   delta_R_G_0 = sum(nu_1 .* sd.G_f_0);
160
161   % Initial mixture
162   w_NH3 = 0.25;
163   rho_RM = 900; %kg/m^3
164
165   p = p_0;
166
167   % Reactor volume (measured)
168   V = 0.1110528 * 10^-3; %m^3
169
170   V_l =  0.0252/rho_RM; %m^3
171
172   V_g = V - V_l; %m^3
173
174   n_N2_t0 = V_g*p_0/(R*298.15);
```

```
175
176 n_t0 = [
177     zeros(size(V_l));
178     V_l*rho_RM*w_NH3/sd.mw(2);
179     n_N2_t0;
180     zeros(size(V_l));
181     zeros(size(V_l));
182     V_l*rho_RM*(1-w_NH3)/sd.mw(6);
183     zeros(size(V_l))]; %mol
184
185
186 %% Experiment specification
187 nu = nu_1;
188 ind = (nu~=0);
189 T = linspace(0, 150, 100) + 273.15;%T_0;
190 alpha = 0;
191
192 %% Phase equilibrium for specified experiment
193 x0 = [0;0;1;1;1;0;1];
194 lb = zeros(size(x0));
195 ub = ones(size(x0));
196
197 options = optimoptions('lsqnonlin', ...
198     'Display','iter', ...
199     "PlotFcn", 'optimplotx');
200
201 for i = 1:length(T)
202     unk = lsqnonlin(@equil_phase, x0, lb, ub, options, alpha, T(i), V_g, n_t0, R, nu, sd);
203     [funval, eq] = equil_phase(unk, alpha, T(i), V_g, n_t0, R, nu, sd);
204     n_i(:,i) = eq.n;
205     n_l_i(:,i) = eq.n_l;
206     n_g_i(:,i) = eq.n_g;
207     x_i(:,i) = eq.x;
208     y_i(:,i) = eq.y;
209     p_tot(i) = eq.p_tot;
210     P_g_i(:,i) = eq.P_g;
211     p_i_i(:,i) = eq.p_i;
212
213 end
214 close all
215
216 Sim.p_tot = p_tot;
217 Sim.T = T;
218 Sim.p_i = p_i_i;
219 Sim.x_i = x_i([2,3,6],:);
220 Sim.y_i = y_i([2,3,6],:);
221
222 save("EXR/Data/Exp11_Sim.mat", "Sim");
223
224 % Total pressure
225 figure
226 plot(T-273.15, p_tot*10^-5, "o-", "LineWidth", 1)
227 xlabel("T / degC");ylabel("p_{tot} / bar");
228 title("p_{tot}")
229 set(gca, "FontName", "LM Roman 12")
230
231 % x_i
232 figure
233 plot(T-273.15, x_i(ind,:), "LineWidth", 1);
234 xlabel("T / degC");ylabel("x_i");
235 title("x_i^{(l)}")
236 legend(subs(ind), "Location", "best")
237 set(gca, "FontName", "LM Roman 12")
238
239 % y_i
240 figure
241 plot(T-273.15, y_i(ind,:), "LineWidth", 1);
242 xlabel("T / degC");ylabel("y_i");
243 title("y_i^{(g)}")
244 legend(subs(ind), "Location", "best")
245 set(gca, "FontName", "LM Roman 12")
246
247 % n_i
248 figure
249 plot(T-273.15, n_i(ind,:), "LineWidth", 1);
250 xlabel("T / degC");ylabel("n_{i}^{(l,g)} / mol");
251 title("n_i^{l,g}")
252 legend(subs(ind), "Location", "best")
253 set(gca, "FontName", "LM Roman 12")
254
255 % n_i^l
256 figure
257 plot(T-273.15, n_l_i(ind,:), "LineWidth", 1);
258 xlabel("T / degC");ylabel("n_i^{(l)} / mol");
259 title("n_i^{(l)}")
260 legend(subs(ind), "Location", "best")
261 set(gca, "FontName", "LM Roman 12")
262
263 % n_i^g
264 figure
```

```
265 plot(T-273.15, n_g_i(ind,:), "LineWidth", 1);
266 xlabel("T / degC");ylabel("n_{i}^{(g)} / mol");
267 title("n_i^{(g)}")
268 legend(subs(ind), "Location", "best")
269 set(gca, "FontName", "LM Roman 12")
270
271 % P_i^g
272 figure
273 plot(T-273.15, P_g_i(ind,:), "LineWidth", 1);
274 xlabel("T / degC");ylabel("P_i^{(g)}");
275 title("P_i^{(g)}")
276 legend(subs(ind), "Location", "best")
277 set(gca, "FontName", "LM Roman 12")
278
279 % p_i
280 figure
281 plot(T-273.15, p_i_i(ind,:)*10^-5, "LineWidth", 1);
282 xlabel("T / degC");ylabel("p_i / bar");
283 title("p_i")
284 legend(subs(ind), "Location", "best")
285 set(gca, "FontName", "LM Roman 12")
286
287 % Display reaction specification
288 disp(append("Experiment temperature: T = ", num2str(T-273.15), ...
289     " degC; Start volume: V_l = ", num2str(V_l*1e6, 3), " mL"));
290
291 % ------------------------------------------------------------------------
```

A.22 read_FC.m

Read data from txt-file of Bronkhorst recording.

```
1  % ------------------------------------read_FC.m-------------------------------------
2  % Florian Walthert
3  % 17.6.20
4
5  % Read data from txt-file of Bronkhorst recording
6  clear all; close all;
7  %% File selecetion
8
9  run = 2;
10
11 switch run
12     case 1
13         filename = [...
14             "Exp12_ct_2";
15             "Exp12_ct_3";
16             "Exp12_ct_4";
17             "Exp12_ct_5"
18             ];
19     case 2
20         filename = [...
21             "Exp16_ct_1";
22             "Exp16_ct_2";
23             "Exp16_ct_3";
24             "Exp16_ct_4"
25             ];
26
27 end
28
29 %% Switch time option
30 % 1: absolute; 2: relative
31 sw = 2;
32 %% Load data from txt-File
33 % Setup the Import Options
34 for i = 1:length(filename)
35
36     opts = delimitedTextImportOptions("NumVariables", 5);
37     opts.DataLines = [1, Inf];
38     opts.Delimiter = ";";
39     opts.VariableNames = ["i", "t", "F", "T", "Valve"];
40     opts.SelectedVariableNames = ["i", "t", "F", "T", "Valve"];
41     opts.VariableTypes = ["string", "string", "string", "string", "string"];
42     opts = setvaropts(opts, [1, 2, 3, 4, 5], "WhitespaceRule", "preserve");
43     opts = setvaropts(opts, [1, 2, 3, 4, 5], "EmptyFieldRule", "auto");
44     opts.ExtraColumnsRule = "ignore";
45     opts.EmptyLineRule = "read";
46     tbl = readtable(append('Data\Bronkhorst\', filename(i), '.txt'), opts);
47     % Convert to output type
48     tbl.i = str2double(strrep(tbl.i,',','.'));
49     tbl.t = str2double(strrep(tbl.t,',','.'));
50     tbl.T = str2double(strrep(tbl.T,',','.'));
51     tbl.F = str2double(strrep(tbl.F,',','.'));
52     tbl.Valve = str2double(strrep(tbl.Valve,',','.'));
```

```
53
54
55          if(i == 1)
56              t0 = tbl.t(1);
57              switch sw
58                  case 1
59                      t = datetime(tbl.t,'ConvertFrom','datenum');
60                  case 2
61                      t = (tbl.t); %s
62              end
63              T = smoothdata(tbl.T, 'gaussian', 1);
64              F = smoothdata(tbl.F, 'gaussian', 1);
65              Valve = smoothdata(tbl.Valve, 'gaussian', 1);
66
67          else
68              t0 = tbl.t(1)
69              switch sw
70                  case 1
71                      t = [t; datetime(tbl.t,'ConvertFrom','datenum')];
72                  case 2
73                      t = [t; tbl.t-tbl.t(1)+t(end)]; %s
74              end
75              T = [T; smoothdata(tbl.T, 'gaussian', 1)];
76              F = [F; smoothdata(tbl.F, 'gaussian', 1)];
77              Valve = [Valve; smoothdata(tbl.Valve, 'gaussian', 1)];
78
79
80          end
81          % Clear temporary variables
82          clear opts tbl
83      end
84
85      %% Integration
86      int = cumtrapz(t, F/60); % mL
87      int = int - int(find(t>=2234,1,'first'));
88
89      Vtot = int(end);
90      disp(append("Total volume: ", num2str(Vtot), " mL"));
91
92      %% Fractionation
93      if run==1
94          t_S = [t(1)/3600, 49.6416, 49.6466,
                49.6799,49.7132,49.7465,49.8131,49.8964,49.9359,49.9527,50.0192,50.1525,50.2025,50.2858,50.7692,t(end)/3600
                *3600;
95          T_S = [-170,-174,-140,-129,-118,-111,-103,-88,-68,-61,-54,-47,-20,1,23,23];
96
97          bord = [49.6416, 49.7132, 49.8964]*3600;%50.2025
98          ind_bord = [1, find(t>=bord(1),1,'first'), find(t>=bord(2),1,'first'),
                find(t>=bord(3),1,'first'), length(t)];
99          for i= 1:length(ind_bord)-1
100             V(i) = int(ind_bord(i+1))-int(ind_bord(i));
101         end
102         V/min(V)
103     end
104
105     if run==2
106         t_S = [2234, 2345, 2600, 2641, 2707, 2837, 3017, 3189, 3730, 3785, 3872, 4033, 4135, 4239,
                4333, 4419, 4486, 4593, 4674, 4757, 4861, 5167, t(end)]; % s
107         T_S = [-170, -168, -165, -156, -142, -128, -120, -116, -112, -107, -100, -90, -83, -78, -74,
                -44, -23, 0, 2, 8, 17, 22, 22]; % degC
108
109         bord = [2234, 2600, 3730, t(end)]; % s % [2234, 2600, 3730, 4333]; % s
110         ind_bord = [find(t>=bord(1),1,'first'), find(t>=bord(2),1,'first'),
                find(t>=bord(3),1,'first'), length(t)];
111         for i= 1:length(ind_bord)-1
112             V(i) = int(ind_bord(i+1))-int(ind_bord(i));
113         end
114         disp(append("Volume fractions: ", num2str(V/V(3))));
115     end
116
117
118     %% Plot
119     figure
120     switch sw
121         case 1
122
123             yyaxis right
124             plot(t/60, T, 'LineWidth', 1)
125             xlabel('t'); ylabel('T / degC')
126
127             yyaxis left
128             plot(t/60, F, 'LineWidth', 1)
129             xlabel('t'); ylabel('F / mL/min')
130
131             grid on
132             set(gca, 'FontName', 'LM Roman 12')
133
134
135         case 2
136             subplot(2,1,1)
```

```
137
138             if false % Plot valve position in %
139                 yyaxis right
140                 plot(t/60, Valve, 'LineWidth', 1)
141                 xlabel('t / min'); ylabel('Valve / %')
142
143                 yyaxis left
144             end
145             plot(t/60, F, 'LineWidth', 1)
146             xlabel('t / min'); ylabel('F / mL/min')
147
148             ylims = get(gca,'Ylim');
149             for i = 1:length(bord)
150                 line([bord(i), bord(i)]/60,ylims)
151             end
152
153             xlim([bord(1)/60, bord(end)/60]);
154             grid on
155             set(gca, 'FontName', 'LM Roman 12')
156
157             subplot(2,1,2)
158             yyaxis left
159             hold on
160             plot(t/60, int)
161             ylims = get(gca,'Ylim');
162             for i = 1:length(bord)
163                 line([bord(i), bord(i)]/60,ylims)
164             end
165             ylim([0, ylims(2)])
166             ylabel("$\int \dot{V} dt$ / mL", "Interpreter", "latex");
167
168             yyaxis right
169             plot(t_S/60,T_S)
170             ylabel("T / degC")
171             xlabel("t / min")
172
173             xlim([bord(1)/60, bord(end)/60]);
174
175             grid on
176             set(gca, 'FontName', 'LM Roman 12')
177
178     end
179     sgtitle("Flow controller read out", 'FontName', 'LM Roman 12');
180
181     % ------------------------------------------------------------------------
```

A.23 read_MA.m

Read out miniautoclave data.

```
1   % ------------------------------------read_MA.m-------------------------------------
2   % Florian Walthert
3   % 10.8.2020
4
5   % Read out miniautoclave data
6   close all; clear all;
7   %% Import data
8   load("Data/HA_SS.mat");
9   t = t_raw*60; % s
10  p1 = p1_raw; % bar
11  p2 = p2_raw; % bar
12  T1 = T1_raw; % degC
13  T2 = T2_raw; % degC
14  deltaT = deltaT_raw; % degC
15
16  %% Plot data
17
18  figure
19
20  subplot(2,1,1)
21  yyaxis left
22  grid on; grid minor; hold on;
23  plot(t/60, p1, 'Linewidth' ,1);
24  plot(t/60, p2, 'Linewidth' ,1);
25  ylabel("p / bar");
26
27  yyaxis right
28  grid on; grid minor; hold on;
29  plot(t/60, T1, 'Linewidth' ,1);
30  plot(t/60, T2, 'Linewidth' ,1);
31  ylabel("T / degC");
32
33  xlabel('t / min');
34  xlim([0,125]);
```

```
35  legend("p1", "p2", "T1", "T2",  'Location', 'best');
36  title("HA + SS chips")
37  set(gca, 'FontName', 'LM Roman 12');
38
39
40  subplot(2,1,2)
41  grid on; grid minor; hold on;
42  plot(t/60, deltaT, 'k', 'Linewidth' ,1);
43  ylabel("\DeltaT / degC");
44  xlabel('t / min');
45  xlim([0,125]);
46  set(gca, 'FontName', 'LM Roman 12');
47
48  % ----------------------------------------------------------------------
```

A.24 DSC_run.m

Run file for DSC modeling and simulation of measurments.

```
1   % ----------------------------------DSC_run.m-------------------------------------
2   % Florian Walthert
3   % 26.5.20
4
5   % Run file for DSC modeling and simulation of measurments
6   clear all; close all
7   tic
8
9   %% Options
10  % Plot experiment curves
11  plexp = 0;
12  % Plot model curves
13  plmod = 0;
14  % Plot Exp13
15  plExp13 = 0;
16
17  %% Crucible material for kinetic modeling
18  % 1: C1 Steel crucible (7 experiments)
19  % 2: C2 Gold crucible (reusable) (5 experiments)
20  % 3: C3 Gold crucible (6 experiments)
21  % 4: Gold crucible , stabilized HA (7 experiments) (no AKTS)
22  sw = 3;
23
24  % Experiments included in the fit
25  fit_exp = 1:6;
26
27  %% Load data - ExpSt
28
29  experiment = [...
30      "steel iso90 1";
31      "steel iso90 2";
32      "steel iso110";
33      "steel iso130";
34      "steel iso150";
35      "steel dyn4 1";
36      "steel dyn4 2"
37      ];
38  filename = [
39      "Exp3_HA_iso_90_24h_2";
40      "Exp3_HA_iso_90_24h_3";
41      "Exp3_HA_iso_110_12h";
42      "Exp3_HA_iso_130_6h";
43      "Exp3_HA_iso_150_5h";
44      "Exp4_HA_dyn_4_12mg";
45      "Exp4_HA_dyn_4_7mg"
46      ];
47  Tr_ramp = [0; 0; 0; 0; 0; 4; 4] / 60; % K/s
48  m = [...
49      15.1100;
50      13.2600;
51      12.16;
52      11.52;
53      11.99;
54      11.82;
55      6.84
56      ] * 1e-6; % kg
57
58  for i = 1:length(experiment)
59      ExpSt(i).filename = filename(i);
60      ExpSt(i).experiment = experiment(i);
61      ExpSt(i).Tr_ramp = Tr_ramp(i);
62      ExpSt(i).m = m(i);
63
64      load(append("Data/", ExpSt(i).filename, ".mat"));
65      ExpSt(i).t_raw = t_raw; % s
```

```
66        ExpSt(i).Tr_raw = Tr_raw; % degC
67        ExpSt(i).HF_raw = (HF_raw-HF_raw(end)) * 10^-3 / ExpSt(i).m; % W/kg
68        if(i ==6 || i==7)
69            ExpSt(i).HF_raw = (HF_raw-HF_raw(end)) * 10^3; % W/kg
70        end
71
72        if(any(i == 1:5))
73            ind = find(ExpSt(i).HF_raw>=0, 1, 'first');
74            ExpSt(i).t(1) = interp1(ExpSt(i).HF_raw([ind-1,ind]), ExpSt(i).t_raw([ind-1,ind]), 0);
75            ExpSt(i).Tr(1) = interp1(ExpSt(i).t_raw([ind-1,ind]), ExpSt(i).Tr_raw([ind-1,ind]),
                  ExpSt(i).t(1));
76            ExpSt(i).HF(1) = interp1(ExpSt(i).t_raw([ind-1,ind]), ExpSt(i).HF_raw([ind-1,ind]),
                  ExpSt(i).t(1));
77
78            ExpSt(i).t = [ExpSt(i).t(1); downsample(t_raw(ind:end), 10)]; % s
79            ExpSt(i).Tr = [ExpSt(i).Tr(1); downsample(Tr_raw(ind:end), 10)]; % degC
80            ExpSt(i).HF = [ExpSt(i).HF(1); downsample(ExpSt(i).HF_raw(ind:end), 10)]; % W/kg
81
82            ind = find(ExpSt(i).HF<=0,2,'first');
83            ExpSt(i).HF(ind(2):end) = 0;
84
85        elseif(i==6 || i==7) % Dynamic measurements
86
87            if(i==6)
88                ind = [find(t_raw >= 3*60, 1, 'first'): find(t_raw >= 48.3333*60, 1, 'first')];
89            elseif(i==7)
90                ind = [find(t_raw >= 2*60, 1, 'first'): find(t_raw >= 48.3333*60, 1, 'first')];
91            end
92
93            ExpSt(i).t = downsample(ExpSt(i).t_raw, 10); % s
94            ExpSt(i).Tr = downsample(ExpSt(i).Tr_raw, 10); % degC
95            ExpSt(i).HF_bl = interp1([ExpSt(i).t_raw(1:ind(1)-1); ExpSt(i).t_raw(ind(end)+1:end)],
                  [ExpSt(i).HF_raw(1:ind(1)-1); ExpSt(i).HF_raw(ind(end)+1:end)], ExpSt(i).t); % W/kg
96            ExpSt(i).HF = downsample(ExpSt(i).HF_raw, 10) - ExpSt(i).HF_bl; % W/kg
97
98        end
99        %     end
100
101       clearvars HF_raw t_raw Tr_raw
102
103       ExpSt(i).dH = sum(trapz(ExpSt(i).t, ExpSt(i).HF)) *1e-3; % J/g
104       % Plot experiment curves
105       if plexp
106
107           figure
108           subplot(2, 1, 1);
109           plot(ExpSt(i).t / 60, ExpSt(i).Tr)
110           xlabel('t / min');
111           ylabel('T / degC');
112           title(append('Experiment ', ExpSt(i).experiment, ' - Temperature program'));
113           grid on
114
115           subplot(2, 1, 2);
116           plot(ExpSt(i).t / 60, ExpSt(i).HF)
117           xlabel('t / min');
118           ylabel('$\dot{q}$ W/kg', 'interpreter', 'Latex');
119           ylim([-50, Inf]);
120           if(i==4)
121               ylim([-1000,Inf]);
122           end
123           title(append('Experiment ', ExpSt(i).experiment, ' - Specific HF'));
124           grid on
125
126       end
127   end
128
129   %% Load data - ExpAu1
130
131   experiment = [...
132       "gold C2 iso90";
133       "gold C2 iso110";
134       "gold C2 iso130";
135       "gold C2 iso150";
136       "gold C2 dyn4"
137       ];
138   filename = [
139       "Exp8_HA_iso_90_24h_2";
140       "Exp8_HA_iso_110_12h";
141       "Exp8_HA_iso_130_6h";
142       "Exp8_HA_iso_150_5h";
143       "Exp8_HA_dyn_4";
144       ];
145   Tr_ramp = [0; 0; 0; 0; 4] / 60; % K/s
146   m = [...
147       13.81;
148       13.62;
149       12.93;
150       12.97;
151       17.62
152       ] * 1e-6; % kg
```

```
153
154 for i = 1:length(experiment)
155     ExpAu1(i).filename = filename(i);
156     ExpAu1(i).experiment = experiment(i);
157     ExpAu1(i).Tr_ramp = Tr_ramp(i);
158     ExpAu1(i).m = m(i);
159
160     load(append("Data/", ExpAu1(i).filename, ".mat"));
161     ExpAu1(i).t_raw = t_raw;
162     ExpAu1(i).Tr_raw = Tr_raw;
163     ExpAu1(i).HF_raw = (HF_raw-HF_raw(end)) * 10^-3 / ExpAu1(i).m;
164
165
166     if(any(i == 1:4)) % Isothermal mesurements
167
168         ind = find(ExpAu1(i).HF_raw>=0, 1, 'first');
169         ExpAu1(i).t(1) = interp1(ExpAu1(i).HF_raw([ind-1,ind]), ExpAu1(i).t_raw([ind-1,ind]), 0);
170         ExpAu1(i).Tr(1) = interp1(ExpAu1(i).t_raw([ind-1,ind]), ExpAu1(i).Tr_raw([ind-1,ind]),
                ExpAu1(i).t(1));
171         ExpAu1(i).HF(1) = interp1(ExpAu1(i).t_raw([ind-1,ind]), ExpAu1(i).HF_raw([ind-1,ind]),
                ExpAu1(i).t(1));
172
173         ExpAu1(i).t = [ExpAu1(i).t(1); downsample(t_raw(ind:end), 10)]; % s
174         ExpAu1(i).Tr = [ExpAu1(i).Tr(1); downsample(Tr_raw(ind:end), 10)]; % degC
175         ExpAu1(i).HF_bl = interp1([ExpAu1(i).t(1), ExpAu1(i).t_raw(end)], [0,0], ExpAu1(i).t);
176         ExpAu1(i).HF = [ExpAu1(i).HF(1); downsample(ExpAu1(i).HF_raw(ind:end), 10)]; % W/kg
177
178
179     elseif(i==5) % Dynamic measurements
180
181         ind = [find(t_raw >= 3*60, 1, 'first'): find(t_raw >= 56*60, 1, 'first')];
182
183         ExpAu1(i).t = downsample(ExpAu1(i).t_raw, 10); % s
184         ExpAu1(i).Tr = downsample(ExpAu1(i).Tr_raw, 10); % degC
185         ExpAu1(i).HF_bl = interp1([ExpAu1(i).t_raw(1:ind(1)-1); ExpAu1(i).t_raw(ind(end)+1:end)],
                [ExpAu1(i).HF_raw(1:ind(1)-1); ExpAu1(i).HF_raw(ind(end)+1:end)], ExpAu1(i).t); % W/kg
186         ExpAu1(i).HF = downsample(ExpAu1(i).HF_raw, 10) - ExpAu1(i).HF_bl; % W/kg
187
188     end
189
190     clearvars HF_raw t_raw Tr_raw
191
192     ExpAu1(i).dH = sum(trapz(ExpAu1(i).t, ExpAu1(i).HF)) *1e-3; % J/g
193     % Plot experiment curves
194     if plexp
195
196         figure
197         subplot(2, 1, 1);
198         plot(ExpAu1(i).t / 60, ExpAu1(i).Tr)
199         xlabel('t / min');
200         ylabel('T / degC');
201         title(append('Experiment ', ExpAu1(i).experiment, ' - Temperature program'));
202         grid on
203
204         subplot(2, 1, 2);
205         hold on
206         plot(ExpAu1(i).t_raw / 60, ExpAu1(i).HF_raw)
207         plot(ExpAu1(i).t / 60, ExpAu1(i).HF_bl)
208         plot(ExpAu1(i).t / 60, ExpAu1(i).HF)
209         xlabel('t / min');
210         ylabel('$\dot{q}$ W/kg', 'interpreter', 'Latex');
211         ylim([-50, Inf]);
212         legend("Raw signal", "Baseline", "Corrected signal")
213         title(append('Experiment ', ExpAu1(i).experiment, ' - Specific HF'));
214         grid on
215
216     end
217 end
218
219 %% Load data - ExpAu2
220
221 experiment = [...
222     "gold C3 iso90";
223     "gold C3 iso110";
224     "gold C3 iso130";
225     "gold C3 iso150";
226     "gold C3 dyn4 1";
227     "gold C3 dyn4 2"
228     ];
229 filename = [
230     "Exp14_HA_iso_90_24h";
231     "Exp14_HA_iso_110_12h";
232     "Exp14_HA_iso_130_6h";
233     "Exp14_HA_iso_150_5h";
234     "Exp14_HA_dyn_4_Mettler";
235     "Exp14_HA_dyn_4_Mettler2"
236     ];
237 Tr_ramp = [0; 0; 0; 0; 4; 4] / 60; % K/s
238 m = [...
239     11.63;
```

```
240         11.74;
241         10.50;
242         11.38;
243         11.10;
244         10.46
245         ] * 1e-6; % kg
246
247     for i = 1:length(experiment)
248         ExpAu2(i).filename = filename(i);
249         ExpAu2(i).experiment = experiment(i);
250         ExpAu2(i).Tr_ramp = Tr_ramp(i);
251         ExpAu2(i).m = m(i);
252
253         load(append("Data/", ExpAu2(i).filename, ".mat"));
254         ExpAu2(i).t_raw = t_raw;
255         ExpAu2(i).Tr_raw = Tr_raw;
256         ExpAu2(i).HF_raw = (HF_raw-HF_raw(end)) * 10^-3 / ExpAu2(i).m;
257
258
259         if(any(i == 1:4)) % Isothermal mesurements
260
261             ind = find(ExpAu2(i).HF_raw>=0, 1, 'first');
262             ExpAu2(i).t(1) = interp1(ExpAu2(i).HF_raw([ind-1,ind]), ExpAu2(i).t_raw([ind-1,ind]), 0);
263             ExpAu2(i).Tr(1) = interp1(ExpAu2(i).t_raw([ind-1,ind]), ExpAu2(i).Tr_raw([ind-1,ind]),
                    ExpAu2(i).t(1));
264             ExpAu2(i).HF(1) = interp1(ExpAu2(i).t_raw([ind-1,ind]), ExpAu2(i).HF_raw([ind-1,ind]),
                    ExpAu2(i).t(1));
265
266             ExpAu2(i).t = [ExpAu2(i).t(1); downsample(t_raw(ind:end), 10)]; % s
267             ExpAu2(i).Tr = [ExpAu2(i).Tr(1); downsample(Tr_raw(ind:end), 10)]; % degC
268             ExpAu2(i).HF_bl = interp1([ExpAu2(i).t(1), ExpAu2(i).t_raw(end)], [0,0], ExpAu2(i).t);
269             ExpAu2(i).HF = [ExpAu2(i).HF(1); downsample(ExpAu2(i).HF_raw(ind:end), 10)]; % W/kg
270
271         elseif(i==5 || i==6) % Dynamic measurements
272
273             if(i==5)
274                 ind = [find(t_raw >= 7*60, 1, 'first'): find(t_raw >= 60.5*60, 1, 'first')];
275             elseif(i==6)
276                 ind = [find(t_raw >= 6.5*60, 1, 'first'): find(t_raw >= 56*60, 1, 'first')];
277             end
278
279             ExpAu2(i).t = downsample(ExpAu2(i).t_raw, 10); % s
280             ExpAu2(i).Tr = downsample(ExpAu2(i).Tr_raw, 10); % degC
281             ExpAu2(i).HF_bl = interp1([ExpAu2(i).t_raw(1:ind(1)-1); ExpAu2(i).t_raw(ind(end)+1:end)],
                    [ExpAu2(i).HF_raw(1:ind(1)-1); ExpAu2(i).HF_raw(ind(end)+1:end)], ExpAu2(i).t); % W/kg
282             ExpAu2(i).HF = downsample(ExpAu2(i).HF_raw, 10) - ExpAu2(i).HF_bl; % W/kg
283
284         end
285
286         clearvars HF_raw t_raw Tr_raw
287
288         ExpAu2(i).dH = sum(trapz(ExpAu2(i).t, ExpAu2(i).HF)) *1e-3; % J/g
289         % Plot experiment curves
290         if plexp
291
292             figure
293             subplot(2, 1, 1);
294             plot(ExpAu2(i).t / 60, ExpAu2(i).Tr)
295             xlabel('t / min');
296             ylabel('T / degC');
297             title(append('Experiment ', ExpAu2(i).experiment, ' - Temperature program'));
298             grid on
299
300             subplot(2, 1, 2);
301             hold on
302             plot(ExpAu2(i).t_raw / 60, ExpAu2(i).HF_raw)
303             plot(ExpAu2(i).t / 60, ExpAu2(i).HF_bl)
304             plot(ExpAu2(i).t / 60, ExpAu2(i).HF)
305             xlabel('t / min');
306             ylabel('$\dot{q}$ W/kg', 'interpreter', 'Latex');
307             ylim([-50, Inf]);
308             legend("Raw signal", "Baseline", "Corrected signal")
309             title(append('Experiment ', ExpAu2(i).experiment, ' - Specific HF'));
310             grid on
311
312         end
313     end
314
315     %% Load data - ExpAu3
316
317     load("Data/partner/Exp16.mat");
318     ExpAu3 = Exp;
319     clearvars Exp
320
321     for i = 1:length(ExpAu3)
322
323         ExpAu3(i).HF_raw = (ExpAu3(i).HF_raw-ExpAu3(i).HF_raw(end)) * 10^-3 / ExpAu3(i).m; % W/kg
324
325
326         if(any(i == 1:5)) % Isothermal mesurements
```

```
327
328             ind = find(ExpAu3(i).HF_raw>=0, 1, 'first');
329             ExpAu3(i).t(1) = interp1(ExpAu3(i).HF_raw([ind-1,ind]), ExpAu3(i).t_raw([ind-1,ind]), 0);
330             ExpAu3(i).Tr(1) = interp1(ExpAu3(i).t_raw([ind-1,ind]), ExpAu3(i).Tr_raw([ind-1,ind]),
                    ExpAu3(i).t(1));
331             ExpAu3(i).HF(1) = interp1(ExpAu3(i).t_raw([ind-1,ind]), ExpAu3(i).HF_raw([ind-1,ind]),
                    ExpAu3(i).t(1));
332
333             ExpAu3(i).t = [ExpAu3(i).t(1); downsample(ExpAu3(i).t_raw(ind:end), 10)]; % s
334             ExpAu3(i).Tr = [ExpAu3(i).Tr(1); downsample(ExpAu3(i).Tr_raw(ind:end), 10)]; % degC
335             ExpAu3(i).HF_bl = interp1([ExpAu3(i).t(1), ExpAu3(i).t_raw(end)], [0,0], ExpAu3(i).t);
336             ExpAu3(i).HF = [ExpAu3(i).HF(1); downsample(ExpAu3(i).HF_raw(ind:end), 10)]; % W/kg
337
338         elseif(i==6 || i==7) % Dynamic measurements
339
340             if(i==6)
341                 ind = [find(ExpAu3(i).t_raw >= 7*60, 1, 'first'): find(ExpAu3(i).t_raw >= 60.5*60, 1,
                        'first')];
342             elseif(i==7)
343                 ind = [find(ExpAu3(i).t_raw >= 6.5*60, 1, 'first'): find(ExpAu3(i).t_raw >= 56*60, 1,
                        'first')];
344             end
345
346             ExpAu3(i).t = downsample(ExpAu3(i).t_raw, 10); % s
347             ExpAu3(i).Tr = downsample(ExpAu3(i).Tr_raw, 10); % degC
348             ExpAu3(i).HF_bl = interp1([ExpAu3(i).t_raw(1:ind(1)-1); ExpAu3(i).t_raw(ind(end)+1:end)],
                    [ExpAu3(i).HF_raw(1:ind(1)-1); ExpAu3(i).HF_raw(ind(end)+1:end)], ExpAu3(i).t); % W/kg
349             ExpAu3(i).HF = downsample(ExpAu3(i).HF_raw, 10) - ExpAu3(i).HF_bl; % W/kg
350
351         end
352
353
354         ExpAu3(i).dH = sum(trapz(ExpAu3(i).t, ExpAu3(i).HF)) *1e-3; % J/g
355         % Plot experiment curves
356         if plexp
357
358             figure
359             subplot(2, 1, 1);
360             plot(ExpAu3(i).t / 60, ExpAu3(i).Tr)
361             xlabel('t / min');
362             ylabel('T / degC');
363             title(append('Experiment ', ExpAu3(i).experiment, ' - Temperature program'));
364             grid on
365
366             subplot(2, 1, 2);
367             hold on
368             plot(ExpAu3(i).t_raw / 60, ExpAu3(i).HF_raw)
369             plot(ExpAu3(i).t / 60, ExpAu3(i).HF_bl)
370             plot(ExpAu3(i).t / 60, ExpAu3(i).HF)
371             xlabel('t / min');
372             ylabel('$\dot{q}$ W/kg', 'interpreter', 'Latex');
373             ylim([-50, Inf]);
374             legend("Raw signal", "Baseline", "Corrected signal")
375             title(append('Experiment ', ExpAu3(i).experiment, ' - Specific HF'));
376             grid on
377
378         end
379     end
380
381     %% Load data - ExpAu3_afteriso
382
383     load("Data/partner/Exp16_afteriso.mat");
384     ExpAu3_afteriso = Exp;
385     clearvars Exp
386
387     for i = 1:length(ExpAu3_afteriso)
388
389         ExpAu3_afteriso(i).HF_raw = (ExpAu3_afteriso(i).HF_raw-ExpAu3_afteriso(i).HF_raw(end)) * 10^-3
                / ExpAu3_afteriso(i).m; % W/kg
390
391
392
393
394         ind = [find(ExpAu3_afteriso(i).t_raw >= 7*60, 1, 'first'): find(ExpAu3_afteriso(i).t_raw >=
                60.5*60, 1, 'first')];
395
396         ExpAu3_afteriso(i).t = downsample(ExpAu3_afteriso(i).t_raw, 10); % s
397         ExpAu3_afteriso(i).Tr = downsample(ExpAu3_afteriso(i).Tr_raw, 10); % degC
398         ExpAu3_afteriso(i).HF_bl = interp1([ExpAu3_afteriso(i).t_raw(1:ind(1)-1);
                ExpAu3_afteriso(i).t_raw(ind(end)+1:end)], [ExpAu3_afteriso(i).HF_raw(1:ind(1)-1);
                ExpAu3_afteriso(i).HF_raw(ind(end)+1:end)], ExpAu3_afteriso(i).t); % W/kg
399         ExpAu3_afteriso(i).HF = downsample(ExpAu3_afteriso(i).HF_raw, 10) - ExpAu3_afteriso(i).HF_bl;
                % W/kg
400         ExpAu3_afteriso(i).dH = sum(trapz(ExpAu3_afteriso(i).t, ExpAu3_afteriso(i).HF)) *1e-3; % J/g
401         % Plot experiment curves
402         if plexp
403
404             figure
405             subplot(2, 1, 1);
406             plot(ExpAu3_afteriso(i).t / 60, ExpAu3_afteriso(i).Tr)
```

```
407             xlabel('t / min');
408             ylabel('T / degC');
409             title(append('Experiment ', ExpAu3_afteriso(i).experiment, ' - Temperature program'));
410             grid on
411
412             subplot(2, 1, 2);
413             hold on
414             plot(ExpAu3_afteriso(i).t_raw / 60, ExpAu3_afteriso(i).HF_raw)
415             plot(ExpAu3_afteriso(i).t / 60, ExpAu3_afteriso(i).HF_bl)
416             plot(ExpAu3_afteriso(i).t / 60, ExpAu3_afteriso(i).HF)
417             xlabel('t / min');
418             ylabel('$\dot{q}$ W/kg', 'interpreter', 'Latex');
419             ylim([-50, Inf]);
420             legend("Raw signal", "Baseline", "Corrected signal")
421             title(append('Experiment ', ExpAu3_afteriso(i).experiment, ' - Specific HF'));
422             grid on
423
424         end
425     end
426
427     %% Load data - ExpAdd
428
429     experiment = [...
430         "gold dyn4 first meas.";
431         "gold dyn4 stab.";
432         "steel dyn4 stab."
433         ];
434     filename = [
435         "Exp2_HA_dyn_4";
436         "HA_gold";
437         "HA_steel"
438         ];
439     Tr_ramp = [4; 4; 4] / 60; % K/s
440     m = [...
441         11.1900;
442         11.2120;
443         11.0440
444         ] * 1e-6; % kg
445
446     for i = 1:length(experiment)
447         ExpAdd(i).filename = filename(i);
448         ExpAdd(i).experiment = experiment(i);
449         ExpAdd(i).Tr_ramp = Tr_ramp(i);
450         ExpAdd(i).m = m(i);
451
452         load(append("Data/", ExpAdd(i).filename, ".mat"));
453         ExpAdd(i).t_raw = t_raw;
454         ExpAdd(i).Tr_raw = Tr_raw;
455         ExpAdd(i).HF_raw = (HF_raw-HF_raw(end)) * 10^-3 / ExpAdd(i).m;
456
457         if(i==1)
458             ind = [find(t_raw >= 4*60, 1, 'first'): find(t_raw >= 52*60, 1, 'first')];
459         elseif(i==2)
460             ind = [find(t_raw >= 6.5*60, 1, 'first'): find(t_raw >= 56*60, 1, 'first')];
461         elseif(i==3)
462             ind = [find(t_raw >= 6.5*60, 1, 'first'): find(t_raw >= 56*60, 1, 'first')];
463         end
464
465         ExpAdd(i).t = downsample(ExpAdd(i).t_raw, 10); % s
466         ExpAdd(i).Tr = downsample(ExpAdd(i).Tr_raw, 10); % degC
467         ExpAdd(i).HF_bl = interp1([ExpAdd(i).t_raw(1:ind(1)-1); ExpAdd(i).t_raw(ind(end)+1:end)],
                [ExpAdd(i).HF_raw(1:ind(1)-1); ExpAdd(i).HF_raw(ind(end)+1:end)], ExpAdd(i).t); % W/kg
468         ExpAdd(i).HF = downsample(ExpAdd(i).HF_raw, 10) - ExpAdd(i).HF_bl; % W/kg
469
470         clearvars HF_raw t_raw Tr_raw
471
472         ExpAdd(i).dH = sum(trapz(ExpAdd(i).t, ExpAdd(i).HF)) *1e-3; % J/g
473         % Plot experiment curves
474         if plexp
475
476             figure
477             subplot(2, 1, 1);
478             plot(ExpAdd(i).t / 60, ExpAdd(i).Tr)
479             xlabel('t / min');
480             ylabel('T / degC');
481             title(append('Experiment ', ExpAdd(i).experiment, ' - Temperature program'));
482             grid on
483
484             subplot(2, 1, 2);
485             hold on
486             plot(ExpAdd(i).t_raw / 60, ExpAdd(i).HF_raw)
487             plot(ExpAdd(i).t / 60, ExpAdd(i).HF_bl)
488             plot(ExpAdd(i).t / 60, ExpAdd(i).HF)
489             xlabel('t / min');
490             ylabel('$\dot{q}$ W/kg', 'interpreter', 'Latex');
491             ylim([-50, Inf]);
492             legend("Raw signal", "Baseline", "Corrected signal")
493             title(append('Experiment ', ExpAdd(i).experiment, ' - Specific HF'));
494             grid on
495
```

```
496         end
497     end
498
499     %% Load data - Exp13
500
501     load("Data/partner/Exp13.mat");
502     Exp13 = Exp;
503     clearvars Exp
504
505     for i = 1:length(Exp13)
506
507         Exp13(i).HF_raw = (Exp13(i).HF_raw-Exp13(i).HF_raw(end)) * 10^-3 / Exp13(i).m; % W/kg
508         if i==4
509             Exp13(i).HF_raw = Exp13(i).HF_raw/2;% Normalized to the total sample mass
510         end
511
512         if any(i==[1:3,5])
513             ind = [find(Exp13(i).t_raw >= 5*60, 1, 'first'): find(Exp13(i).t_raw >= 63*60, 1,
                    'first')];
514         elseif i==4
515             ind = [find(Exp13(i).t_raw >= 0.6833*60, 1, 'first'): find(Exp13(i).t_raw >= 29*60, 1,
                    'first')];
516         elseif any(i==6:8)
517             ind = [find(Exp13(i).t_raw >= 1*60, 1, 'first'): find(Exp13(i).t_raw >= 14.4*60, 1,
                    'first')];
518         elseif i==9
519             ind = [10:10];
520         end
521
522         Exp13(i).t = downsample(Exp13(i).t_raw, 10); % s
523         Exp13(i).Tr = downsample(Exp13(i).Tr_raw, 10); % degC
524         Exp13(i).HF_bl = interp1([Exp13(i).t_raw(1:ind(1)-1); Exp13(i).t_raw(ind(end)+1:end)],
                [Exp13(i).HF_raw(1:ind(1)-1); Exp13(i).HF_raw(ind(end)+1:end)], Exp13(i).t); % W/kg
525         Exp13(i).HF = downsample(Exp13(i).HF_raw, 10) - Exp13(i).HF_bl; % W/kg
526
527         Exp13(i).dH = sum(trapz(Exp13(i).t, Exp13(i).HF)) *1e-3; % J/g
528         % Plot experiment curves
529         if plexp
530
531             figure
532             subplot(2, 1, 1);
533             plot(Exp13(i).t / 60, Exp13(i).Tr)
534             xlabel('t / min');
535             ylabel('T / degC');
536             title(append('Experiment ', Exp13(i).experiment, ' - Temperature program'));
537             grid on
538
539             subplot(2, 1, 2);
540             hold on
541             plot(Exp13(i).t_raw / 60, Exp13(i).HF_raw)
542             plot(Exp13(i).t / 60, Exp13(i).HF_bl)
543             plot(Exp13(i).t / 60, Exp13(i).HF)
544             xlabel('t / min');
545             ylabel('$\dot{q}$ W/kg', 'interpreter', 'Latex');
546             ylim([-50, Inf]);
547             legend("Raw signal", "Baseline", "Corrected signal")
548             title(append('Experiment ', Exp13(i).experiment, ' - Specific HF'));
549             grid on
550
551         end
552     end
553
554     %% Load data - Exp6
555
556     load("Data/Exp6.mat");
557     Exp6 = Exp;
558     clearvars Exp
559
560     for i = 1:length(Exp6)
561
562         Exp6(i).HF_raw = (Exp6(i).HF_raw-Exp6(i).HF_raw(end)) * 10^3; % W/kg
563
564         if i==1
565             ind = [find(Exp6(i).t_raw >= 6.7*60, 1, 'first'): find(Exp6(i).t_raw >= 29*60, 1,
                    'first')];
566         elseif i==2
567             ind = [2: find(Exp6(i).t_raw >= 35*60, 1, 'first')];
568         end
569
570         Exp6(i).t = downsample(Exp6(i).t_raw, 10); % s
571         Exp6(i).Tr = downsample(Exp6(i).Tr_raw, 10); % degC
572         Exp6(i).HF_bl = interp1([Exp6(i).t_raw(1:ind(1)-1); Exp6(i).t_raw(ind(end)+1:end)],
                [Exp6(i).HF_raw(1:ind(1)-1); Exp6(i).HF_raw(ind(end)+1:end)], Exp6(i).t); % W/kg
573         Exp6(i).HF = downsample(Exp6(i).HF_raw, 10) - Exp6(i).HF_bl; % W/kg
574
575         Exp6(i).dH = sum(trapz(Exp6(i).t, Exp6(i).HF)) *1e-3; % J/g
576         % Plot experiment curves
577         if plexp
578
579             figure
```

```
580             subplot(2, 1, 1);
581             plot(Exp6(i).t / 60, Exp6(i).Tr)
582             xlabel('t / min');
583             ylabel('T / degC');
584             if i==1
585                 title(append('Experiment HP SS - Temperature program'));
586             else
587                 title(append('Experiment HP Au - Temperature program'));
588             end
589             grid on
590 
591             subplot(2, 1, 2);
592             hold on
593             plot(Exp6(i).t_raw / 60, Exp6(i).HF_raw)
594             plot(Exp6(i).t / 60, Exp6(i).HF_bl)
595             plot(Exp6(i).t / 60, Exp6(i).HF)
596             xlabel('t / min');
597             ylabel('$\dot{q}$ W/kg', 'interpreter', 'Latex');
598             ylim([-50, Inf]);
599             legend("Raw signal", "Baseline", "Corrected signal")
600             if i==1
601                 title(append('Experiment HP SS - Specific HF'));
602             else
603                 title(append('Experiment HP Au - Specific HF'));
604             end
605             grid on
606 
607         end
608     end
609 
610     %% Plot Exp13
611 
612     if plExp13
613         %% Exp13_1 - HA and NH3
614 
615         figure
616         grid on; grid minor; hold on;
617         for i = 1:3
618 
619             plot(Exp13(i).Tr_raw, Exp13(i).HF_raw-min(Exp13(i).HF_raw), 'Linewidth' ,1);
620             leg(i) = append(Exp13(i).experiment, " (\DeltaH = ", num2str(Exp13(i).dH, 4), " J/g)");
621 
622         end
623 
624         xlabel('T / degC')
625         ylabel('$\dot{q}_{react}$ / W/kg', 'interpreter', 'Latex');
626         legend(leg, 'Location', 'best')
627         clearvars leg
628 
629         title("Exp13\_1")
630         set(gca, 'FontName', 'LM Roman 12');
631 
632         %% Exp13_2 - HA and HP
633 
634         figure
635         grid on; grid minor; hold on;
636         plot(Exp13(4).Tr_raw, Exp13(4).HF_raw, 'Linewidth' ,1);
637         leg(1) = append("HA stabilized + HP 1:1 (Exp13\_2) (\DeltaH = ", num2str(Exp13(4).dH, 4), "
                J/g)");
638 
639         % Stabilized HA
640         plot(ExpAu3(6).Tr_raw, ExpAu3(6).HF_raw, 'Linewidth' ,1);%-min(ExpAu2(5).HF_raw(1:cut))
641         leg(2) = append("HA stabilized 1 (Exp16) (\DeltaH = ", num2str(ExpAu3(6).dH, 4), " J/g)");
642         plot(ExpAu3(7).Tr_raw, ExpAu3(7).HF_raw, 'Linewidth' ,1);%-min(ExpAu2(5).HF_raw(1:cut))
643         leg(3) = append("HA stabilized 2 (Exp16) (\DeltaH = ", num2str(ExpAu3(7).dH, 4), " J/g)");
644 
645         % HP
646         plot(Exp6(2).Tr_raw, Exp6(2).HF_raw, '--', 'Linewidth' ,1);%-min(ExpAu2(5).HF_raw(1:cut))
647         leg(4) = append("HP (Exp6) (\DeltaH = ", num2str(Exp6(2).dH, 4), " J/g)");
648 
649         xlabel('T / degC')
650         ylabel('$\dot{q}_{react}$ / W/kg', 'interpreter', 'Latex');
651         legend(leg, 'Location', 'best')
652         clearvars leg
653 
654 
655         title("Exp13\_2")
656         set(gca, 'FontName', 'LM Roman 12');
657 
658         %% Exp13_3 - HA and TS-1
659 
660         figure
661         grid on; grid minor; hold on;
662         plot(Exp13(5).Tr, Exp13(5).HF, 'Linewidth' ,1);
663         leg(1) = append("HA stabilized + TS-1 1:1 (Exp13\_3) (\DeltaH = ", num2str(Exp13(5).dH, 4), "
                J/g)");
664 
665         % Stabilized HA
666         plot(ExpAu3(6).Tr, ExpAu3(6).HF, 'Linewidth' ,1);
667         leg(2) = append("HA stabilized 1 (Exp16) (\DeltaH = ", num2str(ExpAu3(6).dH, 4), " J/g)");
```

```
668         plot(ExpAu3(7).Tr, ExpAu3(7).HF, 'Linewidth',1);
669         leg(3) = append("HA stabilized 2 (Exp16) (\DeltaH = ", num2str(ExpAu3(7).dH, 4), " J/g)");
670         % Unstabilized HA
671         cut = find(ExpAu2(5).t>=65*60,1,'first');
672         plot(ExpAu2(5).Tr(1:cut), ExpAu2(5).HF(1:cut), '--', 'Linewidth',1);
673         leg(4) = append("HA unstabilized 1 (Exp14) (\DeltaH = ", num2str(ExpAu2(5).dH, 4), " J/g)");
674         plot(ExpAu2(6).Tr(1:cut), ExpAu2(6).HF(1:cut), '--', 'Linewidth',1);
675         leg(5) = append("HA unstabilized 2(Exp14) (\DeltaH = ", num2str(ExpAu2(6).dH, 4), " J/g)");
676
677         xlabel('T / degC')
678         ylabel('$\dot{q}_{react}$ / W/kg', 'interpreter', 'Latex');
679         legend(leg, 'Location', 'best')
680         clearvars leg
681
682         title("Exp13\_3")
683         set(gca, 'FontName', 'LM Roman 12');
684
685         %% Exp13_4 - HP and NH3
686
687         figure
688         grid on; grid minor; hold on;
689         for i = 6:8
690
691             plot(Exp13(i).Tr_raw, Exp13(i).HF_raw-min(Exp13(i).HF_raw), 'Linewidth',1);
692             leg(i) = append(Exp13(i).experiment, " (\DeltaH = ", num2str(Exp13(i).dH, 3), " J/g)");
693         end
694
695         xlabel('T / degC')
696         ylabel('$\dot{q}_{react}$ W/kg', 'interpreter', 'Latex');
697         legend(leg(6:8), 'Location', 'best')
698
699
700         title("Exp13\_4")
701         set(gca, 'FontName', 'LM Roman 12');
702
703         %% Exp13_5 - NH3 and TS-1
704
705         figure
706         grid on; grid minor; hold on;
707         plot(Exp13(9).Tr_raw, Exp13(9).HF_raw, 'Linewidth',1);
708         leg(1) = append("NH3 25% + TS-1 1:1 (Exp13\_5) (\DeltaH = ", num2str(Exp13(9).dH, 4), " J/g)");
709
710         xlabel('T / degC')
711         ylabel('$\dot{q}_{react}$ / W/kg', 'interpreter', 'Latex');
712         legend(leg, 'Location', 'best')
713         clearvars leg
714
715         title("Exp13\_5")
716         set(gca, 'FontName', 'LM Roman 12');
717
718     end
719
720     %% Overlay measurements
721     if false
722         figure
723         titles = ["Iso90", "Iso110", "Iso130", "Iso150", "Dyn4"];
724         for i = 1:5
725             if any( i == 1:4)
726                 subplot(ceil(length(ExpAu1)/2), 2, i)
727                 grid on; grid minor; hold on;
728
729                 plot(ExpSt(i+1).t / 3600, ExpSt(i+1).HF, 'linewidth',1);
730                 plot(ExpAu1(i).t / 3600, ExpAu1(i).HF, 'linewidth',1);
731                 plot(ExpAu2(i).t / 3600, ExpAu2(i).HF, 'linewidth',1);
732                 plot(ExpAu3(i).t / 3600, ExpAu3(i).HF, '--', 'linewidth',1);
733             else
734                 subplot(ceil(length(ExpAu1)/2), 2, [i:i+1])
735                 grid on; grid minor; hold on;
736
737                 cut = find(ExpSt(i).t>=65*60,1,'first');
738                 plot(ExpSt(6).Tr(1:cut), ExpSt(6).HF(1:cut), 'Linewidth',1);
739
740                 plot(ExpAdd(1).Tr, ExpAdd(1).HF, 'linewidth',1);
741
742                 plot(ExpAdd(3).Tr, ExpAdd(3).HF, '--', 'Linewidth',1)
743
744                 plot(ExpAu1(i).Tr, ExpAu1(i).HF, 'linewidth',1);
745
746                 cut = find(ExpAu2(i).t>=65*60,1,'first');
747                 plot(ExpAu2(i).Tr(1:cut), ExpAu2(i).HF(1:cut), 'linewidth',1);
748
749                 plot(ExpAdd(2).Tr, ExpAdd(2).HF, '--', 'linewidth',1);
750
751                 plot(ExpAu3(6).Tr, ExpAu3(6).HF, '--', 'linewidth',1);
752
753                 plot(ExpAu3(7).Tr, ExpAu3(7).HF, '--', 'linewidth',1);
754             end
755
756             ylim([-50,Inf]);
757             if any(i == 1:4)
```

```
758                 xlabel('t / h');
759
760                 legend('SS unstab.(C1)', 'Au unstab.(C2)','Au unstab.(C3)','Au stab.', 'Location',
                        'best');
761             else
762                 xlabel('T / degC')
763                 legend('SS unstab.(C1)', 'SS unstab.(C5)', 'SS stab.', 'Au unstab.(C2)', 'Au
                        unstab.(C3)', 'Au stab.(1)', 'Au stab.(2)', 'Au stab.(3)')
764             end
765             ylabel('$\dot{q}_{react}$ / W/kg', 'interpreter', 'Latex');
766             title(titles(i));
767
768             set(gca, 'FontName', 'LM Roman 12');
769         end
770     end
771
772     %% Model parameters
773
774     switch sw
775         case 1
776             Exp = ExpSt;
777         case 2
778             Exp = ExpAu1;
779         case 3
780             Exp = ExpAu2;
781         case 4
782             Exp = ExpAu3;
783     end
784
785     % Model-base fitting
786
787     [Model_desc, Sim_desc] = DSC_simulation_desc(Exp, fit_exp);
788
789     % Model-free fitting
790
791     [Sim_isocnv, AKTS, Exp] = DSC_simulation_isocnv(Exp, fit_exp, sw);
792
793     %% Plot model curves
794     if plmod
795         for i = 1:length(Exp)
796             figure
797
798             subplot(2, 1, 1);
799             plot(Sim_desc(i).t / 60, Sim_desc(i).Tr_t)
800             xlabel('time [min]');
801             ylabel('[degC]');
802             title(append('Model desc ', Model_desc(i).experiment, ' - Temperature program'));
803             grid on
804
805             subplot(2, 1, 2);
806             plot(Sim_desc(i).t / 60, Sim_desc(i).Qf_react_t / Exp(i).m)
807             xlabel('time [min]');
808             ylabel('$\dot{q}_{react}$ W/kg', 'interpreter', 'Latex');
809             title(append('Model desc ', Model_desc(i).experiment, ' - Specific HF'));
810             grid on
811
812         end
813     end
814
815     %% Calculate NRMSD
816     for i = 1:length(Exp)
817         HF_exp = interp1(Exp(i).t, Exp(i).HF, Sim_desc(i).t);
818         Sim_desc(i).NRMSD = sqrt(mean((HF_exp'- Sim_desc(i).Qf_react_t / Exp(i).m).^2)) /
                mean(HF_exp);%(max(HF_exp)-min(HF_exp));
819         NRMSD_desc(i) = Sim_desc(i).NRMSD;
820         NRMSD_isocnv(i) = Sim_isocnv(i).NRMSD;
821         if ~isempty(AKTS)
822             NRMSD_AKTS(i) = AKTS(sw).Sim(i).NRMSD;
823         end
824         clearvars HF_exp
825     end
826     NRMSD_fit = mean(NRMSD_desc(fit_exp));
827     NRMSD_test = mean(NRMSD_desc(setdiff(1:length(Exp),fit_exp)));
828
829     %% Plot model and experiment curves
830     figure('Name', 'Descriptive')
831     for i = 1:length(Exp)
832
833         subplot(ceil(length(Exp)/2), 2, i)
834         grid on; grid minor; hold on;
835         plot(Exp(i).t / 3600, Exp(i).HF, ':', 'linewidth' ,2);
836         plot(Sim_desc(i).t / 3600 , Sim_desc(i).HF, 'linewidth' ,2);
837         if Exp(i).Tr_ramp>0
838             xlim([0,65/60]);
839         end
840         ylim([-50,Inf]);
841         xlabel('t / h');
842         ylabel('$\dot{q}_{react}$ / W/kg', 'interpreter', 'Latex');
843         legend('Experiment', append('Model (NRMSD = ', num2str(Sim_desc(i).NRMSD*100, '%.3g'), " %)"),
                'Location', 'best');
```

```
844         title(append('Model desc ', Exp(i).experiment, ' - Specific HF'));
845         if(~any(i==fit_exp))
846             title(append('Model desc ', Exp(i).experiment, ' - Specific HF', " (not fitted)"));
847         end
848         set(gca, 'FontName', 'LM Roman 12');
849     end
850
851     %% Display descriptive model parameters
852
853     disp("E_A = ");
854     disp(Model_desc(1).EA);
855     disp("k_0 = ");
856     disp(Model_desc(1).k_0);
857     disp("n = ");
858     disp(Model_desc(1).n);
859     disp(append("Fit: NRMSD = ", num2str(NRMSD_fit*100, '%.3g'), " %"));
860     disp(append("Test: NRMSD = ", num2str(NRMSD_test*100, '%.3g'), " %"));
861     disp(append("Isoconversional: NRMSD = ", num2str(mean(NRMSD_isocnv)*100, '%.3g'), " %"));
862     if ~isempty(AKTS)
863         disp(append("AKTS: NRMSD = ", num2str(mean(NRMSD_AKTS)*100, '%.3g'), " %"));
864     end
865
866
867     %% Plot model and experiment and isocnv curves
868     figure('Name', 'Overview')
869     titles = ["Iso90", "Iso110", "Iso130", "Iso150", "Dyn4"];
870     switch sw
871         case 1
872             for i = 1:length(Exp)
873                 subplot(ceil(length(Exp)/2), 2, i)
874                 grid on; grid minor; hold on;
875                 if any(i==1:5)
876                     plot(Exp(i).t / 3600, Exp(i).HF, ':', 'linewidth',1);
877                     plot(Sim_desc(i).t / 3600 , Sim_desc(i).HF, 'linewidth',1);
878                     plot(Sim_isocnv(i).t_a / 3600 , Sim_isocnv(i).HF_sim, 'linewidth',1);
879                     plot(AKTS(sw).Sim(i).t / 3600 , AKTS(sw).Sim(i).HF, 'linewidth',1);
880                     xlabel('t / h');
881                 else
882                     cut = find(ExpSt(i).t>=65*60,1,'first');
883                     plot(Exp(i).Tr(1:cut), Exp(i).HF(1:cut), ':', 'linewidth',1);
884                     plot(Sim_desc(i).Tr_t, Sim_desc(i).HF, 'linewidth',1);
885                     plot(Sim_isocnv(i).T_a, Sim_isocnv(i).HF_sim, 'linewidth',1);
886                     plot(AKTS(sw).Sim(i).T, AKTS(sw).Sim(i).HF, 'linewidth',1);
887                     xlabel('T / degC');
888                     xlim([0,290]);
889                 end
890                 ylim([-50,Inf]);
891
892                 ylabel('$\dot{q}_{react}$ / W/kg', 'interpreter', 'Latex');
893                 legend(...
894                     'Experiment',...
895                     append('Descriptive model (NRMSD = ', num2str(Sim_desc(i).NRMSD*100, '%.3g'), "
                            %)"),...
896                     append('Isoconversional (NRMSD = ', num2str(Sim_isocnv(i).NRMSD*100, '%.3g'), "
                            %)"),...
897                     append('AKTS (NRMSD = ', num2str(AKTS(sw).Sim(i).NRMSD*100, '%.3g'), " %)"),...
898                     'Location', 'best');
899                 if any(i==1:2)
900                     title(titles(1));
901                 else
902                     title(titles(min([5,i-1])));
903                 end
904                 if(~any(i==fit_exp))
905                     if any(i==1:2)
906                         title(append(titles(1), " (not fitted)"));
907                     else
908                         title(append(titles(min([5,i-1])), " (not fitted)"));
909                     end
910                 end
911                 set(gca, 'FontName', 'LM Roman 12');
912             end
913         case 2
914             for i = 1:length(Exp)
915                 if any(i==1:4)
916                     subplot(ceil(length(Exp)/2), 2, i)
917                     grid on; grid minor; hold on;
918                     plot(Exp(i).t / 3600, Exp(i).HF, ':', 'linewidth',1);
919                     plot(Sim_desc(i).t / 3600 , Sim_desc(i).HF, 'linewidth',1);
920                     plot(Sim_isocnv(i).t_a / 3600 , Sim_isocnv(i).HF_sim, 'linewidth',1);
921                     plot(AKTS(sw).Sim(i).t / 3600 , AKTS(sw).Sim(i).HF, 'linewidth',1);
922                     xlabel('t / h');
923                 else
924                     subplot(ceil(length(Exp)/2), 2, 5:6)
925                     grid on; grid minor; hold on;
926                     cut = find(ExpSt(i).t>=65*60,1,'first');
927                     plot(Exp(i).Tr(1:cut), Exp(i).HF(1:cut), ':', 'linewidth',1);
928                     plot(Sim_desc(i).Tr_t, Sim_desc(i).HF, 'linewidth',1);
929                     plot(Sim_isocnv(i).T_a, Sim_isocnv(i).HF_sim, 'linewidth',1);
930                     plot(AKTS(sw).Sim(i).T, AKTS(sw).Sim(i).HF, 'linewidth',1);
931                     xlabel('T / degC');
```

```
932                 xlim([0,290]);
933             end
934             ylim([-50,Inf]);
935 
936             ylabel('$\dot{q}_{react}$ / W/kg', 'interpreter', 'Latex');
937             legend(...
938                 'Experiment',...
939                 append('Descriptive model (NRMSD = ', num2str(Sim_desc(i).NRMSD*100, '%.3g'), "
                        %)"),...
940                 append('Isoconversional (NRMSD = ', num2str(Sim_isocnv(i).NRMSD*100, '%.3g'), "
                        %)"),...
941                 append('AKTS (NRMSD = ', num2str(AKTS(sw).Sim(i).NRMSD*100, '%.3g'), " %)"),...
942                 'Location', 'best');
943             title(titles(min([5,i])));
944             if(~any(i==fit_exp))
945                 title(append(titles(min([5,i])), " (not fitted)"));
946             end
947             set(gca, 'FontName', 'LM Roman 12');
948         end
949     case 3
950         for i = 1:length(Exp)
951             subplot(ceil(length(Exp)/2), 2, i)
952             grid on; grid minor; hold on;
953             if any(i==1:4)
954                 plot(Exp(i).t / 3600, Exp(i).HF, ':', 'linewidth',1);
955                 plot(Sim_desc(i).t / 3600 , Sim_desc(i).HF, 'linewidth',1);
956                 plot(Sim_isocnv(i).t_a / 3600 , Sim_isocnv(i).HF_sim, 'linewidth',1);
957                 plot(AKTS(sw).Sim(i).t / 3600 , AKTS(sw).Sim(i).HF, 'linewidth',1);
958                 xlabel('t / h');
959             else
960                 cut = find(ExpSt(i).t>=65*60,1,'first');
961                 plot(Exp(i).Tr(1:cut), Exp(i).HF(1:cut), ':', 'linewidth',1);
962                 plot(Sim_desc(i).Tr_t, Sim_desc(i).HF, 'linewidth',1);
963                 plot(Sim_isocnv(i).T_a , Sim_isocnv(i).HF_sim, 'linewidth',1);
964                 plot(AKTS(sw).Sim(i).T , AKTS(sw).Sim(i).HF, 'linewidth',1);
965                 xlabel('T / degC');
966                 xlim([0,290]);
967             end
968             ylim([-50,Inf]);
969 
970             ylabel('$\dot{q}_{react}$ / W/kg', 'interpreter', 'Latex');
971             legend(...
972                 'Experiment',...
973                 append('Descriptive model (NRMSD = ', num2str(Sim_desc(i).NRMSD*100, '%.3g'), "
                        %)"),...
974                 append('Isoconversional (NRMSD = ', num2str(Sim_isocnv(i).NRMSD*100, '%.3g'), "
                        %)"),...
975                 append('AKTS (NRMSD = ', num2str(AKTS(sw).Sim(i).NRMSD*100, '%.3g'), " %)"),...
976                 'Location', 'best');
977             title(titles(min([5,i])));
978             if(~any(i==fit_exp))
979                 title(append(titles(min([5,i])), " (not fitted)"));
980             end
981             set(gca, 'FontName', 'LM Roman 12');
982         end
983     case 4
984         for i = 1:length(Exp)
985             subplot(ceil(length(Exp)/2), 2, i)
986             grid on; grid minor; hold on;
987             if any(i==1:5)
988                 plot(Exp(i).t / 3600, Exp(i).HF, ':', 'linewidth',1);
989                 plot(Sim_desc(i).t / 3600 , Sim_desc(i).HF, 'linewidth',1);
990                 plot(Sim_isocnv(i).t_a / 3600 , Sim_isocnv(i).HF_sim, 'linewidth',1);
991                 xlabel('t / h');
992             else
993                 cut = find(ExpSt(i).t>=65*60,1,'first');
994                 plot(Exp(i).Tr(1:cut), Exp(i).HF(1:cut), ':', 'linewidth',1);
995                 plot(Sim_desc(i).Tr_t, Sim_desc(i).HF, 'linewidth',1);
996                 plot(Sim_isocnv(i).T_a, Sim_isocnv(i).HF_sim, 'linewidth',1);
997                 xlabel('T / degC');
998                 xlim([0,290]);
999             end
1000            ylim([-50,Inf]);
1001 
1002            ylabel('$\dot{q}_{react}$ / W/kg', 'interpreter', 'Latex');
1003            legend(...
1004                'Experiment',...
1005                append('Descriptive model (NRMSD = ', num2str(Sim_desc(i).NRMSD*100, '%.3g'), "
                        %)"),...
1006                append('Isoconversional (NRMSD = ', num2str(Sim_isocnv(i).NRMSD*100, '%.3g'), "
                        %)"),...
1007                'Location', 'best');
1008            if any(i==1:4)
1009                title(titles(i));
1010            elseif i==5
1011                title("Iso170");
1012            else
1013                title(titles(5));
1014            end
1015            if(~any(i==fit_exp))
```

```
1016                    if any(i==1:4)
1017                        title(append(titles(i), " (not fitted)"));
1018                    elseif i==5
1019                        title(append("Iso170", " (not fitted)"));
1020                    else
1021                        title(append(titles(5), " (not fitted)"));
1022                    end
1023                end
1024                set(gca, 'FontName', 'LM Roman 12');
1025            end
1026    end
1027    if any(sw==1:3)
1028        sgtitle(append("Simulation C", num2str(sw)), 'FontName', 'LM Roman 12');
1029    else
1030        sgtitle("Simulation stabilized hydroxylamine solution", 'FontName', 'LM Roman 12');
1031    end
1032
1033    toc
1034
1035    % ------------------------------------------------------------------------
```

A.25 DSC_simulation_desc.m

Function to simulate DSC measurements with a descriptive model.

```
1   % ------------------------DSC_simulation_desc.m-----------------------------
2   % Florian Walthert
3   % 13.07.20
4
5   % Function to simulate DSC measurements with a descriptive model
6   function [Model_desc, Sim_desc] = DSC_simulation_desc(Exp, fit_exp)
7
8   %% Descriptive Model
9
10  for i = 1:length(Exp)
11      Model_desc(i).experiment = Exp(i).experiment;
12      % 1: n^th order model
13      % 2: Autocatalysis
14      % 3: 1^st order + autocatalysis
15      switch 1
16          case 1 % n^th order model
17
18              Model_desc(i).EA = [63.866; 0; 0] * 1000; % kJ/mol
19              Model_desc(i).n = [ ...
20                  0.8434,0,0; ...
21                  0,0,0; ...
22                  0,0,0  ...
23                  ]; % reaction order
24              Model_desc(i).k_0 = [0.37e-6; 0; 0];
25
26              Model_desc(i).DRH = [-2128]; % [J/g] = [kJ/kg] taken from the DSC measurements
27
28              % stoichiometric coefficients of the components
29              Model_desc(i).nu_i = [ ...
30                  -1,0,0; ...
31                  0,0,0; ...
32                  0,0,0  ...
33                  ]; % [-]
34
35              Kin_par0 = [Model_desc(i).EA; Model_desc(i).k_0; Model_desc(i).n(1,1)];
36              lb = [5e4; 0; 0; 1e-7; 0; 0; 0.75];
37              ub = [7e4; 0; 0; 1e-5; 0; 0; 0.9];
38
39          case 2 % Autocatalysis
40
41              Model_desc(i).EA = [50; 50; 0] * 1000; % [J/mol]
42              Model_desc(i).n = [ ...
43                  1,0,0; ...
44                  1,1,0; ...
45                  0,0,0  ...
46                  ]; % reaction order of A
47
48              Model_desc(i).k_0 = [1e-3; 1e-3; 0];
49
50              Model_desc(i).DRH = [-2128]; % [J/g] = [kJ/kg] taken from the DSC measurements
51
52              % stoichiometric coefficients of the components
53              Model_desc(i).nu_i = [ ...
54                  -1,1,0; ...
55                  -1,1,0; ...
56                  0,0,0  ...
57                  ]; % [-]
58
59              Kin_par0 = [Model_desc(i).EA; Model_desc(i).k_0];% Model_desc(1).n];
```

```
60                lb = [1e3; 1e3; 0; 1e-10; 1e-10; 0];% 1e-3; 1e-3];
61                ub = [2e5; 2e5; 0; 1e-2; 1e-2; 0]%; 3; 3];
62
63            case 3 % 1^st order + autocatalysis
64
65                Model_desc(i).EA = [66.163; 100; 40] * 1000; % [J/mol]
66                Model_desc(i).n = [ ...
67                    1,0,0; ...
68                    1,0,0; ...
69                    1,1,0 ...
70                    ]; % reaction order of A
71    %             Model_desc(i).m = [0;0]; % reaction order of B
72                Model_desc(i).k_0 = [1.747e-7; 1e-12; 1e-6];
73
74                Model_desc(i).DRH = [-2128]; % [J/g] = [kJ/kg] taken from the DSC measurements
75
76                % stoichiometric coefficients of the components
77                Model_desc(i).nu_i = [ ...
78                    -1,1,0; ...
79                    -1,1,0; ...
80                    -1,1,0 ...
81                    ]; % [-]
82
83                Kin_par0 = [Model_desc(i).EA; Model_desc(i).k_0];
84                lb = [66.163e3; 1e5; 1e3; 1.747e-7; 1e-14; 1e-8];
85                ub = [66.163e3; 2e5; 1e5; 1.747e-7; 1e-5; 1e-2];
86
87        end
88
89        % Reference temperature
90        Model_desc(i).T_0 = 25 * ones(size(Model_desc(i).k_0));    % [degC]
91
92        % Molecular weights of each component (given in the exercise)
93        Model_desc(i).mw_i = [33.03, 33.03, 33.03] * 10^-3; %kg/mol]
94
95        Model_desc(i).m_RM = Exp(i).m;
96
97        % Density of the reaction mass - assumed to be constant
98        Model_desc(i).rho_RM = 1120; % [kg/m3] HA 50 %
99
100       w_HA = 0.5;
101
102       n_HA_0 = Exp(i).m * w_HA / Model_desc(i).mw_i(1); % [mol]
103       n_H2O_0 = Exp(i).m * (1-w_HA) / Model_desc(i).mw_i(2); % [mol]
104       Model_desc(i).n_i_0 = [n_HA_0; 0; 0];
105       Model_desc(i).V_RM = Model_desc(i).m_RM / Model_desc(i).rho_RM * 1000;    % [l]
106
107       % Decomposition enthalpy based on the DSC measurement (integral)
108
109       Model_desc(i).DRH_m = Model_desc(i).DRH .* Model_desc(i).m_RM * 1000 /
              Model_desc(i).n_i_0(1);% [J/mol]
110       deltaH_r_0 = - 557/4; %kJ/mol
111       exp_H = 0.5/Model_desc(i).mw_i(1) * deltaH_r_0; %J/g
112
113       Model_desc(i).experiment = Exp(i).experiment;
114       Model_desc(i).Tr_ramp = Exp(i).Tr_ramp;
115
116       % Initial concentrations [mol/l]
117       Model_desc(i).c_i_0 = Model_desc(i).n_i_0 / Model_desc(i).V_RM; % [mol/l]
118
119
120   end
121
122
123
124   options = [];%optimoptions('fmincon', ...
125   %     'Display', 'iter',...
126   %     'PlotFcn', 'optimplotfval'...
127   %     );
128
129   switch 1
130
131       case 1 % No optimization
132
133           Kin_par = Kin_par0;
134
135       case 2 % Local optimization
136           Kin_par = fmincon(@DSC_optim_desc, Kin_par0, [], [], [], [], lb, ub, [], options,
                  Exp(fit_exp), Model_desc(fit_exp))
137
138       case 3 % Global optimization
139           f = @(Kin_par)DSC_optim_desc(Kin_par, Exp(fit_exp), Model_desc(fit_exp));
140           problem = createOptimProblem('fmincon','objective',f,'x0',Kin_par0,'lb',lb,...
141               'ub',ub,'options',optimset('Algorithm','SQP','Disp','iter'));
142           gs = GlobalSearch;
143           Kin_par = run(gs,problem);
144
145       case 4 % MultiStart
146           f = @(Kin_par)DSC_optim_desc(Kin_par, Exp(fit_exp), Model_desc(fit_exp));
147
```

```
148             rng default % For reproducibility
149
150             problem = createOptimProblem('fmincon','objective',f,'x0',Kin_par0,'lb',lb,...
151                 'ub',ub,'options',optimset('Algorithm','SQP'));%,'Disp','iter', 'PlotFcn',
                        'optimplotfunccount'));
152             ms = MultiStart;
153
154             % Starting points
155             Kin_par = run(ms,problem, 100);
156
157     end
158
159     [~, Sim_desc, Model_desc] = DSC_optim_desc(Kin_par, Exp, Model_desc);
160
161
162     figure('Name','Concetrations')
163     for i = 1:length(Exp)
164
165         subplot(ceil(length(Exp)/2), 2, i)
166         plot(Sim_desc(i).t/3600, Sim_desc(i).c_i_t(1,:), 'k-', 'linewidth', 2);
167         hold on
168         plot(Sim_desc(i).t/3600, Sim_desc(i).c_i_t(2,:), 'k', 'linewidth', 0.5);
169         plot(Sim_desc(i).t/3600, Sim_desc(i).c_i_t(3,:), 'k:', 'linewidth', 2);
170         grid on; grid minor
171         xlabel('t / h]');
172         ylabel('Concentrations [mol/l]');
173         title(append('Concentrations ', Model_desc(i).experiment));
174         legend('1: c_{NH2OH}', '2: c_{P1}', '3: c_{P2}');
175
176     end
177
178     figure('Name','Rates')
179     for i = 1:length(Exp)
180
181         subplot(ceil(length(Exp)/2), 2, i)
182         plot(Sim_desc(i).t/3600, Sim_desc(i).r_t, 'linewidth', 2);
183         xlabel('t / h]');
184         ylabel('Reaction rate / *mol/l/s*');
185         title(append('Reaction rates ', Model_desc(i).experiment));
186         legend('r_1', 'r_2');
187
188     end
189
190     end
191
192     % ------------------------------------------------------------------------------
```

A.26 DSC_optim_desc.m

Optimization function for descriptive models.

```
1   % ------------------------------DSC_optim_desc.m-----------------------------------
2   % Florian Walthert
3   % 04.07.20
4
5   % Optimization function for descriptive models
6   function [funval, Sim_desc, Model_desc] = DSC_optim_desc(x, Exp, Model_desc)
7
8   % Physical constants
9   R = 8.314; % [J/(mol*K)]
10
11  for i = 1:length(Exp)
12
13      Model_desc(i).EA = x([1,2,3]);%([1,2]);
14
15      Model_desc(i).k_0 = x([4,5,6]);
16
17      Model_desc(i).n(1,1) = x(7);
18
19
20      % Initial values of the time dependant variables
21      y0 = [Exp(i).Tr(1); Model_desc(i).n_i_0];% 0];
22
23      % Conditions for the simulation
24      n_t = 1000;     % [-] number of time points of the results
25      t_0 = Exp(i).t(1);
26      t_end = Exp(i).t(end); % [s] simulation time
27      tspan = linspace(t_0, t_end, n_t); % [s] time points at which a result is calculated
28
29      % Precision of the numerical integration
30      %       options = odeset('AbsTol', 1e-13);%odeset('AbsTol', 1e-10); % Absolute tolerance
31      %       options = odeset(options, 'RelTol', 1e-6);%odeset(options, 'RelTol', 1e-6); % Relative
            tolerance
```

```
32
33      % call the ode-solver
34      Sim_desc(i).experiment = Model_desc(i).experiment;
35      [Sim_desc(i).t, Sim_desc(i).y] = ode23s(@DSC_model_desc, tspan, y0, [], ...
            Model_desc(i));
36
37      %% Display results
38      % Results of the numerical integration
39      Sim_desc(i).y(Sim_desc(i).y<0) = 0;
40      Sim_desc(i).Tr_t = Sim_desc(i).y(:, 1)';  % [degC]
41      Sim_desc(i).c_i_t = Sim_desc(i).y(:, 2:end)' / Model_desc(i).V_RM; % [mol/l]
42
43      % Exercise 5:
44
45      % Reaction rate
46      Sim_desc(i).k_t = Model_desc(i).k_0 .* exp(Model_desc(i).EA ./ R .* (1./(Model_desc(i).T_0 +
            273.15) - 1./(Sim_desc(i).Tr_t + 273.15)));
47      Sim_desc(i).r_t = Sim_desc(i).k_t .*
            Sim_desc(i).c_i_t(1,:).^Model_desc(i).n(:,1).*Sim_desc(i).c_i_t(2,:).^Model_desc(i).n(:,2);
            % [mol/(s*l)]
48
49      % corrections for 0'th order rate laws
50
51      Sim_desc(i).r_t((Sim_desc(i).c_i_t(1) <= 0) | (Sim_desc(i).c_i_t(2) <= 0)) = 0; % [mol/(s*l)]
52
53
54      % Conversion rate of each component
55      for ii = 1:length(Model_desc(i).nu_i)
56          Sim_desc(i).r_i_t(ii,:) = sum(Model_desc(i).nu_i(:,ii) .* Sim_desc(i).r_t);% [mol/(s*l)]
57      end
58
59
60
61      % Heat flow due to decomposition reaction
62      Sim_desc(i).Qf_react_t = Sim_desc(i).r_i_t(1,:) .* Model_desc(i).V_RM .* Model_desc(i).DRH_m ;
            % [W]
63
64      Sim_desc(i).HF = Sim_desc(i).Qf_react_t ./ Exp(i).m; % W/kg
65
66      Sim_HF(i,:) = Sim_desc(i).HF; % W/kg
67      Exp_HF(i,:) = interp1(Exp(i).t, Exp(i).HF, Sim_desc(i).t); % W/kg
68
69      error(i) = sum(((Sim_HF(i,:) - Exp_HF(i,:))./mean(Exp_HF(i,:))).^2);
70
71  funval= sum(error);
72
73  end
74
75  % ------------------------------------------------------------------------------
76
```

A.27 DSC_model_desc.m

Differential equations for descriptive models.

```
1   % ------------------------------DSC_model_desc.m-------------------------------
2   % Florian Walthert
3   % 04.07.20
4
5   % Differential equations for descriptive models
6   function dy_dt = DSC_model_desc(t, y, Model_desc)
7
8   % Physical constants
9   R = 8.314; % [J/(mol*K)]
10
11  %% Actual values for the time dependent variables
12  % All values within the vector "y" below zero are set to zero
13  y(y<0) = 0;
14
15  T_R = y(1);          % [degC]
16  n_i = y(2:end);  % [mol]
17
18  %% Calculation of the reaction volume and the concentrations
19  c_i = n_i ./ Model_desc.V_RM; % [mol/l] concentrations of each component
20
21  %% Moleflow and Mole flow balance
22  % Reaction rate
23  k = Model_desc.k_0 .* exp(Model_desc.EA ./ R .* (1./(Model_desc.T_0 + 273.15) - 1./(T_R +
        273.15))); % [1/s]
24  r = k.*c_i(1).^Model_desc.n(:,1).*c_i(2).^Model_desc.n(:,2).*c_i(3).^Model_desc.n(:,3); %
        [mol/(s*l)]
25  r((n_i(1) <= 0) ) = 0; % [mol/(s*l)]
26
27  % Conversion rate of component "i"
28  r_i = sum(Model_desc.nu_i .* r); % [mol/(s*l)] eq. (5.1) ... (5.4)
```

```
29
30  % mole flow balance according
31  dn_i_dt = r_i'*Model_desc.V_RM; % [mol/s]
32
33  %% Heat flow balance (short cut!)
34  dT_R_dt = Model_desc.Tr_ramp;      % [degC/s]
35
36  %% Return the derivatives of "y"
37  dy_dt = [dT_R_dt; dn_i_dt];
38
39  end
40
41  % ------------------------------------------------------------------------
```

A.28 DSC_simulation_isocnv.m

Script to simulate measurements with isoconversional kinetics (calculation + import from AKTS software).

```
1   % ------------------------DSC_simulation_isocnv.m-------------------------
2   % Florian Walthert
3   % 04.07.20
4
5   % Script to simulate measurements with isoconversional kinetics
6   % (calculation + import from AKTS software)
7   function [Sim_isocnv, AKTS, Exp] = DSC_simulation_isocnv(Exp, fit_exp, sw)
8
9   % Physical constants
10  R = 8.314; % [J/(mol*K)]
11
12  for i = 1:length(Exp)
13
14      % Integration of heat flow delivers enthalpy
15      Exp(i).dH_c = cumtrapz(Exp(i).t, Exp(i).HF); % J/kg
16      Exp(i).dH_c_max = max(Exp(i).dH_c); % J/kg
17      % Sink enthalpies into vector
18      dH_v(i) = Exp(i).dH_c_max; % J/kg
19
20      % Calculate thermal conversion
21      Exp(i).alpha = Exp(i).dH_c / Exp(i).dH_c_max;
22  end
23
24  %% Plot reaction conversion as a function of time for every experiment
25  figure
26
27  hold on
28  for i = 1:length(Exp)
29      plot(Exp(i).t/3600, Exp(i).alpha, 'Linewidth', 1);
30  end
31  set(gca, 'YLim', [0 1]);
32  legend(Exp.experiment);
33  ylabel('\alpha');
34  xlabel('t / h');
35  grid on
36  set(gca, 'FontName', 'LM Roman 12');
37
38  %% Interpolate the different DSC measurements based on alpha
39  pts = 101;
40
41  for i = 1:length(Exp)
42      Sim_isocnv(i).alpha = linspace(0, 1, pts)';
43
44      start = find(Exp(i).dH_c > 0);
45      start = max([1,start(1)-1]);
46      stop = find(Exp(i).dH_c == Exp(i).dH_c_max, 1, 'first');
47
48      Sim_isocnv(i).t_a = interp1(Exp(i).alpha(start:stop), Exp(i).t(start:stop),
            Sim_isocnv(i).alpha); % s
49      Sim_isocnv(i).T_a = interp1(Exp(i).alpha(start:stop), Exp(i).Tr(start:stop),
            Sim_isocnv(i).alpha); % degC
50      Sim_isocnv(i).q_a = interp1(Exp(i).alpha(start:stop), Exp(i).HF(start:stop),
            Sim_isocnv(i).alpha); % W/kg
51
52  end
53
54  %% Determine EA(alpha) and k(alpha) by linear regression
55
56  figure
57  for i = 2:pts-1
58
59      % Preallocation for improved performance
60      T = zeros(1, length(fit_exp)); % degC
61      q = zeros(1, length(fit_exp)); % W/kg
```

```
62
63      % Only defined experiments included in fitting
64      for ii= 1:length(fit_exp)
65          T(ii) = Sim_isocnv(fit_exp(ii)).T_a(i);
66          q(ii) = Sim_isocnv(fit_exp(ii)).q_a(i);
67      end
68      p = polyfit(1000./(273.15 + T), log(abs(q)), 1);
69
70      % Calculate coefficient of determination
71      Corr = corrcoef(1000./(273.15 + T), log(abs(q)));
72      Sim_isocnv(1).R2(i) = Corr(2,1)^2;
73
74      if true % Continuous arrhenius plot
75          x_data = [1000./(273.15 + T)];
76          y_data = [log(abs(q))];
77          p1 = plot(x_data, y_data, 'x');
78          hold on
79          p2 = plot([min(x_data), max(x_data)], [p(1)*min(x_data)+p(2), p(1)*max(x_data)+p(2)], 'r');
80          drawnow;
81          title('Arrhenius Plot');
82          delete(p1);
83          delete(p2);
84      end
85
86      % Sink results into struct Sim_isocnv
87      for ii= 1:length(Exp)
88          Sim_isocnv(ii).EA(i)= abs(p(1)*R);
89          Sim_isocnv(ii).ln_k(i)= p(2);
90      end
91  end
92  close; % Close Arrhenius plot
93
94  for i = 1:length(Exp)
95
96      % Calculate EA/ln(k) (alpha = 0)
97      Sim_isocnv(i).EA(1)= interp1(Sim_isocnv(i).alpha(2:end-1), Sim_isocnv(i).EA(2:end), 0,
            'linear', 'extrap');
98      Sim_isocnv(i).ln_k(1)= interp1(Sim_isocnv(i).alpha(2:end-1), Sim_isocnv(i).ln_k(2:end), 0,
            'linear', 'extrap');
99
100     %Calculate EA/ln(k) (alpha = 1)
101     Sim_isocnv(i).EA(pts) = interp1(Sim_isocnv(i).alpha(1:end-1), Sim_isocnv(i).EA, 1, 'linear',
            'extrap');
102     Sim_isocnv(i).ln_k(pts) = interp1(Sim_isocnv(i).alpha(1:end-1), Sim_isocnv(i).ln_k, 1,
            'linear', 'extrap');
103
104     % Calculate heat flow from kinetic parameters
105     Sim_isocnv(i).HF_sim = exp(Sim_isocnv(i).ln_k).*exp(-Sim_isocnv(i).EA.*1000./R./(273.15 +
            Sim_isocnv(i).T_a'));
106
107 end
108
109 %% Include evaluation done with AKTS software
110 if sw ~= 4 % No evaluation for ExpAu3
111     load("Data\AKTS\AKTS.mat")
112
113     % Enthalpies
114     switch sw
115         case 1
116             AKTS(sw).Sim(1).dH = 2460.540;
117             AKTS(sw).Sim(2).dH = 2684.114;
118             AKTS(sw).Sim(3).dH = 2178.234;
119             AKTS(sw).Sim(4).dH = 2874.371;
120             AKTS(sw).Sim(5).dH = 1814.644;
121             AKTS(sw).Sim(6).dH = 2344.643;
122             AKTS(sw).Sim(7).dH = 2401.571;
123         case 2
124             AKTS(sw).Sim(1).dH = 1759.391;
125             AKTS(sw).Sim(2).dH = 1761.326;
126             AKTS(sw).Sim(3).dH = 2068.932;
127             AKTS(sw).Sim(4).dH = 2546.713;
128             AKTS(sw).Sim(5).dH = 2449.560;
129         case 3
130             AKTS(sw).Sim(1).dH = 1252.036;
131             AKTS(sw).Sim(2).dH = 1925.307;
132             AKTS(sw).Sim(3).dH = 2208.489;
133             AKTS(sw).Sim(4).dH = 2437.811;
134             AKTS(sw).Sim(5).dH = 2437.405;
135             AKTS(sw).Sim(6).dH = 2681.978;
136     end
137
138     % EA/ln(k)
139     for i = 1:length(Exp)
140         % Calculate EA/ln(k)
141         AKTS(sw).Sim(i).alpha = Sim_isocnv(i).alpha;
142         AKTS(sw).Sim(i).EA = interp1(AKTS(sw).alpha_raw, AKTS(sw).EA_raw, AKTS(sw).Sim(i).alpha);
                % kJ/mol
143         AKTS(sw).Sim(i).ln_k = interp1(AKTS(sw).alpha_raw, AKTS(sw).ln_k_raw,
                AKTS(sw).Sim(i).alpha);
144
```

```
145             % Calculate EA/ln(k) (alpha = 0)
146             AKTS(sw).Sim(i).EA(1)= interp1(AKTS(sw).Sim(i).alpha(2:end-1),
                    AKTS(sw).Sim(i).EA(2:end-1), 0, 'linear', 'extrap');
147             AKTS(sw).Sim(i).ln_k(1)= interp1(AKTS(sw).Sim(i).alpha(2:end-1),
                    AKTS(sw).Sim(i).ln_k(2:end-1), 0, 'linear', 'extrap');
148
149             % Calculate EA/ln(k) (alpha = 1)
150             AKTS(sw).Sim(i).EA(pts) = interp1(AKTS(sw).Sim(i).alpha(1:end-1),
                    AKTS(sw).Sim(i).EA(1:end-1), 1, 'linear', 'extrap');
151             AKTS(sw).Sim(i).ln_k(pts) = interp1(AKTS(sw).Sim(i).alpha(1:end-1),
                    AKTS(sw).Sim(i).ln_k(1:end-1), 1, 'linear', 'extrap');
152
153             % Calculate t, T and heat flow
154             start = find(Exp(i).dH_c > 0);
155             start = start(1)-1;% is equal to the start (1)!!!!!!!!!!!!!!!
156             stop = find(Exp(i).dH_c == Exp(i).dH_c_max);%(bl_type));
157
158             AKTS(sw).Sim(i).t = interp1(Exp(i).alpha(start:stop), Exp(i).t(start:stop),
                    AKTS(sw).Sim(i).alpha); % s
159             AKTS(sw).Sim(i).T = interp1(Exp(i).alpha(start:stop), Exp(i).Tr(start:stop),
                    AKTS(sw).Sim(i).alpha); % degC
160             AKTS(sw).Sim(i).HF = exp(AKTS(sw).Sim(i).ln_k).*exp(-AKTS(sw).Sim(i).EA.*1000./R./(273.15
                    + AKTS(sw).Sim(i).T)) * AKTS(sw).Sim(i).dH * 1000;%   W/kg
161
162         end
163     end
164
165     %% Calculation - Plot EA, ln(k) and R^2 as a function of alpha
166
167     figure
168
169     yyaxis left
170     plot(Sim_isocnv(1).alpha, Sim_isocnv(1).EA, 'Linewidth', 1);
171     hold on
172     plot(Sim_isocnv(1).alpha, Sim_isocnv(1).ln_k, '--', 'Linewidth', 1);
173     ylabel('E_A / kJ/mol, ln(k)');
174     grid on
175
176     yyaxis right
177     plot(Sim_isocnv(1).alpha(2:end), Sim_isocnv(1).R2, 'Linewidth', 1);
178     set(gca,'ylim', [0 2]);
179     ylabel("R^2");
180
181     xlabel('\alpha');
182     legend('E_A', 'ln(k)', 'R^2', 'Location', 'southwest');
183     title('Isoconversional kinetics - Kinetic parameters');
184     set(gca, 'FontName', 'LM Roman 12');
185
186     %% AKTS - Plot EA, ln(k) and R^2 as a function of alpha
187     if exist("AKTS", "var")
188         figure
189
190         plot(AKTS(sw).alpha_raw, AKTS(sw).EA_raw, 'Linewidth', 1);
191         hold on
192         plot(AKTS(sw).alpha_raw, AKTS(sw).ln_k_raw, '--', 'Linewidth', 1);
193         xlabel('\alpha'); ylabel('E_A / kJ/mol, ln(k)');
194         grid on
195         legend('E_A', 'ln(k)', 'Location', 'southwest');
196         title('AKTS kinetics - Kinetic parameters');
197         set(gca, 'FontName', 'LM Roman 12');
198     end
199
200     %% Plot EA, ln(k) for every experiment
201
202     if false
203         figure
204
205         hold on
206         for i = 1:length(Exp)
207             p(1) = plot(Sim_isocnv(i).t_a/3600, Sim_isocnv(i).EA, '-');
208             p(2) = plot(Sim_isocnv(i).t_a/3600, Sim_isocnv(i).ln_k, '--');
209             leg([2*i-1,2*i]) = [append(Exp(i).experiment, " E_A"), append(Exp(i).experiment, "
                        ln(k)")];
210         end
211
212         ylabel('E_A / kJ/mol, ln(k)');
213         xlabel('t / h');
214         legend(leg);
215         title('Isoconversional kinetics - Kinetic parameters');
216         set(gca, 'FontName', 'LM Roman 12');
217     end
218
219     %% Calculate NRMSD
220     for i = 1:length(Exp)
221         HF_exp = interp1(Exp(i).t, Exp(i).HF, Sim_isocnv(i).t_a);
222         Sim_isocnv(i).NRMSD = sqrt(mean((HF_exp- Sim_isocnv(i).HF_sim').^2)) / mean(HF_exp);
223         if exist("AKTS", "var")
224             AKTS(sw).Sim(i).NRMSD = sqrt(mean((HF_exp- AKTS(sw).Sim(i).HF).^2)) / mean(HF_exp);
225         end
226     end
```

```
227
228 %% Plot isoconversional kinetics and experiment curves
229
230 figure('Name', "Isoconversional kinetics")
231 for i = 1:length(Exp)
232     subplot(ceil(length(Exp)/2), 2, i)
233     grid on; grid minor; hold on;
234     plot(Exp(i).t/3600, Exp(i).HF, 'k', 'Linewidth', 1);
235     plot(Sim_isocnv(i).t_a/3600, Sim_isocnv(i).HF_sim, 'r', 'Linewidth', 1)
236     title(Exp(i).experiment);
237     ylabel('$\dot{q}$ / W/kg', 'Interpreter', 'latex');
238     xlabel('t / h');
239     legend('Experiment', append('Model (NRMSD = ', num2str(Sim_isocnv(i).NRMSD*100, '%.3g'), "
            %"), 'Location', 'best');
240     if(~any(i==fit_exp))
241         title(append(Exp(i).experiment, " (not fitted)"));
242     end
243     set(gca, 'FontName', 'LM Roman 12');
244 end
245
246 %% Plot AKTS and experiment curves
247 if exist("AKTS", "var")
248     figure('Name', "AKTS kinetics")
249     for i = 1:length(Exp)
250         subplot(ceil(length(Exp)/2), 2, i)
251         grid on; grid minor; hold on;
252         plot(Exp(i).t/3600, Exp(i).HF, 'k', 'Linewidth', 1);
253         plot(AKTS(sw).Sim(i).t/3600, AKTS(sw).Sim(i).HF, 'r', 'Linewidth', 1)
254         title(Exp(i).experiment);
255         ylabel('$\dot{q}$ / W/kg', 'Interpreter', 'latex');
256         xlabel('t / h');
257         legend('Experiment', append('Model (NRMSD = ', num2str(AKTS(sw).Sim(i).NRMSD*100, '%.3g'),
                " %"), 'Location', 'best');
258         if(~any(i==fit_exp))
259             title(append(Exp(i).experiment, " (not fitted)"));
260         end
261         set(gca, 'FontName', 'LM Roman 12');
262     end
263 end
264
265 %% Ensure complete output
266 if ~exist("AKTS", "var")
267     AKTS = [];
268 end
269
270 end
271
272 % ------------------------------------------------------------------------
```

A.29 importDSC_partner.m

Script to import the DSC measurements done at the project partners Process Safety lab.

```
1  % ------------------------------importDSC_partner.m-------------------------------
2  % Florian Walthert
3  % 04.07.20
4
5  % Script to import the DSC measurements done at partner PSL
6  clear all;
7
8  filename = [...
9      "Exp13_1.1-6mg HA 50% + 6mg NH3 25%";
10     "Exp13_1.2-6mg HA 50% + 4mg NH3 25%";
11     "Exp13_1.3-6mg HA 50% + 2mg NH3 25%";
12     "Exp13_2-6mg HA 50% + 6mg H2O2 50%";
13     "Exp13_3-6mg HA 50% + 6mg TS-1";
14     "Exp13_4.1-6mg H2O2 50% + 6mg NH3 25%";
15     "Exp13_4.2-6mg H2O2 50% + 4mg NH3 25%";
16     "Exp13_4.3-6mg H2O2 50% + 2mg NH3 25%";
17     "Exp13_5-6mg NH3 25% + 6mg TS-1"
18     ];
19
20 experiment = [...
21     "HA NH3 3:3";
22     "HA NH3 3:2";
23     "HA NH3 3:1";
24     "HA HP 1:1";
25     "HA TS-1 1:1";
26     "HP NH3 3:3";
27     "HP NH3 3:2";
28     "HP NH3 3:1";
29     "NH3 TS-1 1:1"
30     ];
```

```
31
32  m = [...
33       6.06;
34       7.346;
35       9.016;
36       6;
37       6;
38       5.859;
39       7.542;
40       8.905;
41       6
42       ] * 1e-6; % mg
43
44  Tr_ramp = [4;4;4;4;4;4;4;4] / 60; % K/s
45
46  for i = 1:length(filename)
47       [Exp(i).t_raw, Exp(i).Ts_raw, Exp(i).Tr_raw, Exp(i).HF_raw] = importfile(filename(i));
48       Exp(i).m = m(i);
49       Exp(i).Tr_ramp = Tr_ramp(i);
50       Exp(i).filename = filename(i);
51       Exp(i).experiment = experiment(i);
52
53       % Remove NaN's
54       rm_ind = unique([find(isnan(Exp(i).t_raw)==1); find(isnan(Exp(i).Ts_raw)==1);
             find(isnan(Exp(i).Tr_raw)==1); find(isnan(Exp(i).HF_raw)==1)]);
55       Exp(i).t_raw(rm_ind) = [];
56       Exp(i).Ts_raw(rm_ind) = [];
57       Exp(i).Tr_raw(rm_ind) = [];
58       Exp(i).HF_raw(rm_ind) = [];
59  end
60
61  save(append("Data/partner/Exp13.mat"), "Exp")
62
63  % ------------------------------------------------------------------------------
```

A.30 EXR_run.m

Script to evaluate and model the measurements of the EXR.

```
1   % ------------------------------------EXR_run.m-----------------------------------------
2   % Florian Walthert
3   % 17.6.20
4
5   % Script to evaluate and model the measurements of the EXR
6   close all;clear all;
7
8   %% Experiment selection
9   % experiment = [...
10  %       1:"test";
11  %       2:"Exp9";
12  %       3:"Exp10";
13  %       4:"Exp11"; Simulation available
14  %       5:"Exp12"; Simulation available
15  %       6:"Exp15"; Simulation available
16  %       7:"Exp16";
17  %       8:"Exp18";
18  %       9:"Exp19";
19  %       10:"Exp20"
20  %       ];
21
22  % Experiments to include
23  run = [3:7,9,10];
24
25  % Options
26  pltexp = 0; % Plot experiment curves
27
28  %% Load EXR data from csv-File
29  Exp = read_EXR(run);
30
31  %% Plot experiment curves
32  if pltexp
33       for i = run
34            figure
35
36            yyaxis right
37
38            plot(Exp(i).t/3600, Exp(i).Tr, 'LineWidth', 1)
39            xlabel('t / h'); ylabel('T / degC')
40
41
42            yyaxis left
43            plot(Exp(i).t/3600, Exp(i).p, 'LineWidth', 1)
44            xlabel('t / h'); ylabel('p / bar')
45            grid on
```

```
46
47          title(Exp(i).experiment)
48          set(gca, 'FontName', 'LM Roman 12')
49
50      end
51  end
52
53  %% Further calculations
54  for i = run
55      % Calculate jacket temperature
56      if ~isempty(Exp(i).T_j_window)
57          ind =
                (find(Exp(i).t>=Exp(i).T_j_window(1)*3600,1,'first'):find(Exp(i).t>=Exp(i).T_j_window(2)*3600,1,'first'
58          Exp(i).T_j = mean(Exp(i).Tr(ind), 'omitnan'); % degC
59      else
60          Exp(i).T_j = [];
61      end
62      % Calculate derivatives
63      Exp(i).dpdt = diff(Exp(i).p)./diff(Exp(i).t);
64      Exp(i).dTdt = diff(Exp(i).Tr)./diff(Exp(i).t);
65
66  end
67
68  %% Verification of Henry's law with ammonia solution in EXR
69  if ~isempty(Exp(4).experiment)
70      clearvars Model Sim
71      load("Data/Exp11_Sim.mat")
72      Exp(4).Sim.T = Sim.T; % degC
73      Exp(4).Sim.p = interp1(Sim.T'-273.15, Sim.p_tot*1e-5, Exp(4).Tr); % bar
74
75      figure('Name', 'Ammonia')
76
77      yyaxis right
78      plot(Exp(4).t/3600, Exp(4).Tr, 'LineWidth', 1)
79      xlabel('t / h'); ylabel('T / degC')
80
81      yyaxis left
82      plot(Exp(4).t/3600, Exp(4).p, 'LineWidth', 1)
83      hold on
84      plot(Exp(4).t/3600, Exp(4).Sim.p, 'LineWidth', 1)
85      xlabel('t / h'); ylabel('p / bar')
86      grid on
87      set(gca, 'FontName', 'LM Roman 12')
88      legend("Experiment", "Model", "Temperature")
89
90  end
91  run(find(run==4)) = [];
92
93  %% Comparison of all experiments (specific gas production)
94  R = 8.314; % J/mol/K
95  % Reactor volume (measured)
96  V = 0.1110528 * 10^-3; %m^3
97
98  if true
99      figure
100     hold on
101     for i = 1:length(run)
102
103             plot(Exp(run(i)).t/3600, Exp(run(i)).p ./ (Exp(run(i)).Tr+273.15) *  V / R /
                    Exp(run(i)).m, 'LineWidth', 1)
104             leg(i) = Exp(run(i)).experiment;
105
106     end
107     xlabel('t / h'); ylabel('\Gamma / mol/kg')
108     grid on;
109
110     legend("PEEK (Au) (Exp10)","SS (Au) (Exp12)","PEEK (Silicone) (Exp15)","HC22 (Silicone)
            (Exp16)","PEEK (SS) (Exp19)", "PEEK (SS) with TS-1 (Exp20)", 'Location', 'best')
111     % legend("Stainless steel","Hastelloy C22","PEEK","PEEK with TS-1 catalyst", 'Location',
            'best')%
112     set(gca, 'FontName', 'LM Roman 12')
113 end
114
115 %% Kalorimetric modeling
116 if false
117 % Jacket temperature with thermostat temperature of 128 degC
118 for i = 1:length(Exp)
119     if ~isempty(Exp(i).T_j) && i~=4
120
121         Exp(i).delta_T = Exp(i).Tr - Exp(i).T_j;
122         Exp(i).delta_T(Exp(i).delta_T<0) = 0;
123
124         delta_Hr = -2128e3; % J/kg
125
126         % Integral of Delta_T must be equal to delta_Hr
127         Exp(i).UA = fsolve(@UA_optim, 120, [], delta_Hr, Exp(i));
128         [~, Exp(i).HF_rj] = UA_optim(Exp(i).UA, delta_Hr, Exp(i));
129
130         % Plot HF from reactor to jacket as a function of time
131         figure
```

```
132
133             yyaxis right
134             plot(Exp(i).t/3600, Exp(i).HF_rj, 'LineWidth', 1)
135             xlabel('t / h'); ylabel('$\dot{q}$ / W/kg', 'Interpreter', 'latex')
136
137             yyaxis left
138             plot(Exp(i).t/60/60, Exp(i).p, 'LineWidth', 1)
139             xlabel('t / h'); ylabel('p / bar')
140
141             grid on
142             set(gca, 'FontName', 'LM Roman 12')
143             sgtitle(append("Modeled heat flow from ", Exp(i).experiment), 'FontName', 'LM Roman 12');
144         end
145     end
146     end
147
148     %% Thermodynamic modeling (Model EXR_HA_estimation.m)
149     for i = run
150
151         % Interpolation of simulation according to EXR experiment
152
153         if i~=4
154             clearvars Model Sim
155
156             load(append("Data/", Exp(i).experiment, "_Sim.mat"));
157             Exp(i).Model = Model;
158
159             figure
160             for r = 1:length(Model)-1
161
162                 if ~exist('Sim', 'var') % Calculate alpha(t)
163                     % Create struct Sim_alpha with alpha(t) based on experimental
164                     % data
165                     Sim(r).rxn = Model(r).rxn;
166                     Sim(r).t= Exp(i).t;
167                     Sim(r).T= Exp(i).Tr;
168
169                     Z = interp2(Model(r).T, Model(r).alpha, Model(r).p_tot, Exp(i).Tr,
                            Model(r).alpha); % bar
170                     for ii = 1:length(Z)
171                         Sim(r).alpha(ii) = interp1(Z(:,ii), Model(r).alpha, Exp(i).p(ii), 'linear',
                                'extrap');
172                     end
173                 end
174
175                 % Correct alpha according to acidimetric determination of end
176                 % point
177                 if ~isempty(Exp(i).alpha_end)
178                     Sim(r).alpha_corr = Sim(r).alpha * Exp(i).alpha_end / Sim(r).alpha(end);
179                 end
180
181                 % Plot alpha(t)
182                 if ~isempty(Exp(i).alpha_end) % With alpha_corr
183
184                     subplot(2,1,1)
185                     hold on
186                     plot(Sim(r).t/3600, Sim(r).alpha, 'LineWidth', 1)
187
188                     subplot(2,1,2)
189                     hold on
190                     plot(Sim(r).t/3600, Sim(r).alpha_corr, 'LineWidth', 1)
191
192
193                 else % Without alpha_corr
194                     hold on
195                     plot(Sim(r).t/3600, Sim(r).alpha, 'LineWidth', 1)
196                 end
197
198                 leg(r) = append("Model ", Model(r).rxn, " reaction");
199
200             end
201             % Finish plot
202             if ~isempty(Exp(i).alpha_end) % With alpha_corr
203
204                 subplot(2,1,1)
205                 grid on
206                 plot(Sim(r).t(end)/3600, Exp(i).alpha_end, 'x', 'LineWidth', 2)
207                 xlabel("t/ h"); ylabel("\alpha");
208                 legend([leg, "\alpha_{end}"], 'Location', 'best');
209                 title("\alpha(t)");
210                 set(gca, 'FontName', 'LM Roman 12');
211
212                 subplot(2,1,2)
213                 grid on
214                 xlabel("t/ h"); ylabel("\alpha");
215                 legend(leg, 'Location', 'best');
216                 clearvars leg
217                 title("\alpha(t) corrected with \alpha_{end}")
218                 set(gca, 'FontName', 'LM Roman 12');
219
```

```
220             else  % Without alpha_corr
221                 grid on
222                 xlabel("t/ h"); ylabel("\alpha");
223                 legend(leg, 'Location', 'best');
224                 clearvars leg
225                 set(gca, 'FontName', 'LM Roman 12');
226             end
227
228             sgtitle(Exp(i).experiment, 'FontName', 'LM Roman 12')
229
230             save(append("Data/", Exp(i).experiment, "_Sim.mat"), "Model", "Sim");
231
232             %% Calculate NRMSD and plot of finished simulation
233             if isfield(Sim, "p_tot")
234
235                 figure; hold on; grid on;
236                 plot(Exp(i).t/3600, Exp(i).p, '--', 'LineWidth', 1)
237                 for r=1:length(Sim)-1
238                     plot(Sim(r).t/3600, Sim(r).p_tot, 'LineWidth', 1)
239                     Sim(r).NRMSD = sqrt(mean((Exp(i).p- Sim(r).p_tot').^2)) /
                            (max(Exp(i).p)-min(Exp(i).p));
240                     leg(r) = append("Model ", Model(r).rxn, " reaction (NRMSD = ",
                            num2str(Sim(r).NRMSD*100,'%.3g'), " %)");
241
242                 end
243                 xlabel("t/ h"); ylabel("p / bar");
244                 legend(["Experiment", leg], 'Location', 'best');
245                 clearvars leg
246                 title(Exp(i).experiment);
247                 set(gca, 'FontName', 'LM Roman 12');
248
249
250             end
251             Exp(i).Sim = Sim;
252         end
253     end
254
255     %% Thermokinetic modeling
256     for i = run
257         if i~=4 && isfield(Exp(i).Sim(1), "p_tot")
258
259             Exp(i).q = cumtrapz(Exp(i).t, Exp(i).HF_rj);
260             figure
261             subplot(2,1,1)
262             hold on
263             plot(Exp(i).t/3600, Exp(i).q, 'LineWidth', 1)
264             subplot(2,1,2)
265             hold on
266             plot(Exp(i).t/3600, Exp(i).HF_rj, 'LineWidth', 1)
267
268             for r = 1:length(Exp(i).Sim)
269                 % Calculate derivatives
270                 Exp(i).Sim(r).dalpha_dt = diff(smoothdata(Exp(i).Sim(r).alpha, 'gaussian', 100)) ./
                        diff(Exp(i).Sim(r).t');
271                 Exp(i).Sim(r).dalpha_corr_dt = diff(smoothdata(Exp(i).Sim(r).alpha_corr, 'gaussian',
                        100)) ./ diff(Exp(i).Sim(r).t');
272
273                 % Integral curves
274                 Exp(i).Sim(r).dH = delta_Hr * Exp(i).Sim(r).alpha * Exp(i).Sim(r).n_l_i(1,1);
275                 Exp(i).Sim(r).dH_corr = delta_Hr * Exp(i).Sim(r).alpha_corr * Exp(i).Sim(r).n_l_i(1,1);
276
277                 % Differential curves
278                 Exp(i).Sim(r).HF = delta_Hr * Exp(i).Sim(r).dalpha_dt * Exp(i).Sim(r).n_l_i(1,1);
279                 Exp(i).Sim(r).HF_corr = delta_Hr * Exp(i).Sim(r).dalpha_corr_dt *
                        Exp(i).Sim(r).n_l_i(1,1);
280
281                 % Plot
282                 subplot(2,1,1)
283                 plot(Exp(i).Sim(r).t/3600, Exp(i).Sim(r).dH, 'LineWidth', 1)
284                 plot(Exp(i).Sim(r).t/3600, Exp(i).Sim(r).dH_corr, 'LineWidth', 1)
285
286                 subplot(2,1,2)
287                 plot(Exp(i).Sim(r).t(2:end)/3600, Exp(i).Sim(r).HF, 'LineWidth', 1)
288                 plot(Exp(i).Sim(r).t(2:end)/3600, Exp(i).Sim(r).HF_corr, 'LineWidth', 1)
289
290                 leg(2*r-1) = append("Model ", Exp(i).Sim(r).rxn, " reaction");
291                 leg(2*r) = append("Model ", Exp(i).Sim(r).rxn, " reaction (corrected)");
292             end
293
294             % Finish the plot
295             subplot(2,1,1)
296             grid on
297             xlabel("t/ h"); ylabel("q");
298             legend(["q from temperature measurement", leg], 'Location', 'best');
299             title("Reaction heat");
300             set(gca, 'FontName', 'LM Roman 12');
301
302             subplot(2,1,2)
303             grid on
304             xlabel('t / h'); ylabel('$\dot{q}$ / W/kg', 'Interpreter', 'latex')
```

```
305             legend(["HF_{rj} from temperature measurement", leg], 'Location', 'best');
306             clearvars leg
307             title("Reaction heat flow")
308             set(gca, 'FontName', 'LM Roman 12');
309
310             sgtitle(Exp(i).experiment, 'FontName', 'LM Roman 12')
311         end
312     end
313
314
315     %% Check models
316     if false
317     for i = 1:length(Exp)
318         if ~isempty(Exp(i).Sim) && i~=4
319
320             figure
321
322             plot(Exp(i).t/3600, Exp(i).p, 'LineWidth', 1)
323             hold on
324             plot(Exp(i).t/3600, Exp(i).Sim.p_1, 'LineWidth', 1)
325             plot(Exp(i).t/3600, Exp(i).Sim.p_2, 'LineWidth', 1)
326             plot(Exp(i).t/3600, Exp(i).Sim.p_3, 'LineWidth', 1)
327
328             xlabel('t / h'); ylabel('p / bar')
329             grid on
330
331             legend("Experiment", "Model empiric", "Model alkaline", "Model acid")
332             sgtitle(append("Model curves from ", Exp(i).experiment), 'FontName', 'LM Roman 12');
333             set(gca, 'FontName', 'LM Roman 12')
334
335         end
336     end
337     end
338
339     % ------------------------------------------------------------------------
```

A.31 read_EXR.m

Function to read data from csv-file of EXR recording.

```
1   % ------------------------------read_EXR.m-----------------------------------
2   % Florian Walthert
3   % 17.6.20
4
5   % Function to read data from csv-file of EXR recording
6   function Exp = read_EXR(run)
7
8   for i = run
9       %% File selecetion + sample mass
10
11      switch i
12          case 1
13              Exp(i).experiment = "test";
14              filename = [...
15                  "tightness2"
16                  ];
17              Exp(i).m = []; % kg
18              window = [0, 30.2972]; % h
19              Exp(i).T_j_window = []; % h
20          case 2
21              Exp(i).experiment = "Exp9";
22              filename = [...
23                  "Exp9_1";
24                  "Exp9_2";
25                  "Exp9_3"
26                  ];
27              Exp(i).m = 22.4 * 1e-3; % kg
28              window = [23.3194, 46.7722]; % h
29              Exp(i).T_j_window = []; % h
30          case 3
31              Exp(i).experiment = "Exp10";
32              filename = [...
33                  "Exp10_1";
34                  "Exp10_2";
35                  "Exp10_3"
36                  ];
37              Exp(i).m = 32.08 * 1e-3; % kg
38              window = [23.2611, 47.01]; % h
39              Exp(i).T_j_window = [1.07, 2.36]; % h
40              Exp(i).alpha_end = 0.9568;
41          case 4
42              Exp(i).experiment = "Exp11";
43              filename = [...
44                  "Exp11_1";
```

```
45                      "Exp11_2"
46                      ];
47                  Exp(i).m = 25.2 * 1e-3; % kg
48                  window = [4.5528, 25.6]; % h
49                  Exp(i).T_j_window = []; % h
50              case 5
51                  Exp(i).experiment = "Exp12";
52  %                 load("Data/Eurotherm/Sim_Exp12.mat")
53                  filename = [...
54                      "Exp12_1";
55                      "Exp12_2";
56                      "Exp12_3"
57                      ];
58                  Exp(i).m = 32.6 * 1e-3; % kg
59                  window = [18.0111, 41.8]; % h
60                  Exp(i).T_j_window = [8.7, 10.28]; % h
61                  Exp(i).alpha_end = 0.9291;
62              case 6
63                  Exp(i).experiment = "Exp15";
64  %                 load("Data/Eurotherm/Sim_Exp15.mat")
65                  filename = [...
66                      "Exp15_1";
67                      "Exp15_2";
68                      "Exp15_3"
69                      ];
70                  Exp(i).m = 32.5 * 1e-3; % kg
71                  window = [28.1222, 57.815]; % h
72                  Exp(i).T_j_window = [18.17, 22.35]; % h
73                  Exp(i).alpha_end = 0.9745;
74              case 7
75                  Exp(i).experiment = "Exp16";
76  %                 load("Data/Eurotherm/Sim_Exp16.mat")
77  %                 load("Data/Eurotherm/Exp16_Sim_alpha.mat")
78                  filename = [...
79                      "Exp16_1";
80                      "Exp16_2";
81                      "Exp16_3"
82                      ];
83                  Exp(i).m = 32.6 * 1e-3; % kg
84                  window = [17.15, 43.4]; % h
85                  Exp(i).T_j_window = [8.0, 9.9]; % h
86                  Exp(i).alpha_end = 0.9745;
87              case 8
88                  Exp(i).experiment = "Exp18";
89                  filename = [...
90                      "Exp18_1"
91                      ];
92                  Exp.m = []; % kg
93                  Exp(i).T_j_window = []; % h
94              case 9
95                  Exp(i).experiment = "Exp19";
96                  load("Data/Eurotherm/Sim_Exp19.mat")
97                  filename = [...
98                      % "Exp19_00";
99                      % "Exp19_0";
100                     "Exp19_1";
101                     "Exp19_2"
102                     ];
103                 Exp(i).m = 31.1 * 1e-3; % kg
104                 window = [2.4083, 32.8956]; % h
105                 Exp(i).T_j_window = [20.75, 22.46]; % h
106                 Exp(i).alpha_end = 0.9432;
107             case 10
108                 Exp(i).experiment = "Exp20";
109                 %                   load("Data/Eurotherm/Sim_Exp20.mat")
110                 filename = [...
111                     "Exp20_3";
112                     "Exp20_4";
113                     "Exp20_5";
114                     "Exp20_6";
115                     "Exp20_7";
116                     "Exp20_8"
117                     ];
118                 Exp(i).m = 32.5 * 1e-3; % kg
119                 window = [4.1583, 107.7472];%[2.4083, 32.8956]; % h
120                 Exp(i).T_j_window = [23.1639, 24.6722]; % h
121                 Exp(i).alpha_end = 0.9833;
122     end
123
124     %% Import data
125     % Setup the Import Options
126     for ii = 1:length(filename)
127
128         opts = delimitedTextImportOptions("NumVariables", 7);
129         opts.DataLines = [14, Inf];
130         opts.Delimiter = ";";
131         opts.VariableNames = ["t", "Var2", "Tr", "p", "pl", "Var6", "Var7"];
132         opts.SelectedVariableNames = ["t", "Tr", "p", "pl"];
133         opts.VariableTypes = ["string", "string", "string", "string", "string", "string",
                "string"];
```

```
134             opts = setvaropts(opts, [1, 2, 3, 4, 5, 6, 7], "WhitespaceRule", "preserve");
135             opts = setvaropts(opts, [1, 2, 3, 4, 5, 6, 7], "EmptyFieldRule", "auto");
136             opts.ExtraColumnsRule = "ignore";
137             opts.EmptyLineRule = "read";
138             tbl = readtable(append('Data\Eurotherm\', filename(ii), '.csv'), opts);
139 
140             % Convert to output type
141             if(ii == 1)
142                 t0 = tbl.t(1);
143 
144 
145                 Exp(i).t_raw = (datenum(datetime(tbl.t)  - datetime(t0))) * 24 *3600; %s
146 
147                 Exp(i).Tr_raw = smoothdata(str2double(strrep(tbl.Tr,',','.')), 'gaussian', 1);
148                 Exp(i).p_raw = smoothdata(str2double(strrep(tbl.p,',','.')), 'gaussian', 1);
149 
150             else
151 
152                 Exp(i).t_raw = [Exp(i).t_raw; (datenum(datetime(tbl.t)  - datetime(t0))) * 24 *3600];
                        %s
153 
154                 Exp(i).Tr_raw = [Exp(i).Tr_raw; smoothdata(str2double(strrep(tbl.Tr,',','.')),
                        'gaussian', 1)];
155                 Exp(i).p_raw = [Exp(i).p_raw; smoothdata(str2double(strrep(tbl.p,',','.')),
                        'gaussian', 1)];
156                 % pl = tbl.pl;
157 
158             end
159             % Clear temporary variables
160             clear opts tbl
161 
162         end
163 
164         %% Determination of experiment window
165         if false
166             figure
167 
168             yyaxis right
169             plot(Exp(i).t_raw/3600, Exp(i).Tr_raw, 'LineWidth', 1)
170             xlabel('t / h'); ylabel('T / degC')
171 
172             yyaxis left
173             plot(Exp(i).t_raw/60/60, Exp(i).p_raw, 'LineWidth', 1)
174             xlabel('t / h'); ylabel('p / bar')
175 
176 
177             grid on
178             set(gca, 'FontName', 'LM Roman 12')
179 
180         end
181 
182         %% Process data
183 
184         % Experiment window
185         ind =
                (find(Exp(i).t_raw>=window(1)*3600,1,'first'):find(Exp(i).t_raw>=window(2)*3600,1,'first'))';
186 
187         Exp(i).t = Exp(i).t_raw(ind)-Exp(i).t_raw(ind(1));
188         Exp(i).Tr = Exp(i).Tr_raw(ind);
189         Exp(i).p = Exp(i).p_raw(ind);
190 
191         % Remove NaN's
192         rm = union(union(find(isnan(Exp(i).t)), find(isnan(Exp(i).Tr))), find(isnan(Exp(i).p)));
193         Exp(i).Tr(rm) = [];
194         Exp(i).p(rm) = [];
195         Exp(i).t(rm) = [];
196 
197     end
198 
199     clearvars Sim
200     end
201 
202     % ------------------------------------------------------------------------------
```

A.32 UA_optim.m

Function to optimize the heat transfer coefficient.

```
1   % ------------------------------------equil_phase.m-------------------------------------
2   % Florian Walthert
3   % 03.08.20
4   
5   % Function to optimize the heat transfer coefficient
```

```
6   function [funval, HF_rj] = UA_optim(UA, delta_Hr, Exp)
7
8   HF_rj = UA * Exp.delta_T ./ Exp.m; % W/kg
9
10  Q_cum_t = cumtrapz(Exp.t(Exp.delta_T>0),HF_rj(Exp.delta_T>0)); % J/kg
11  Q_end = Q_cum_t(end); % J/kg
12
13  funval = (Q_end - delta_Hr); % J/kg
14
15  end
16
17  % ------------------------------------------------------------------------
```

A.33 Functions used in other scripts

Functions that were used in several other scripts.

```
1  % ------------------------------------Antoine.m-------------------------------------
2  % Florian Walthert
3  % 04.07.20
4
5  function p = Antoine(Ant, T_K)
6
7  p = 10 .^(Ant(:,1) - Ant(:,2) ./ (Ant(:,3) + T_K));
8
9  end
10
11 % ----------------------------------------------------------------------------------
```

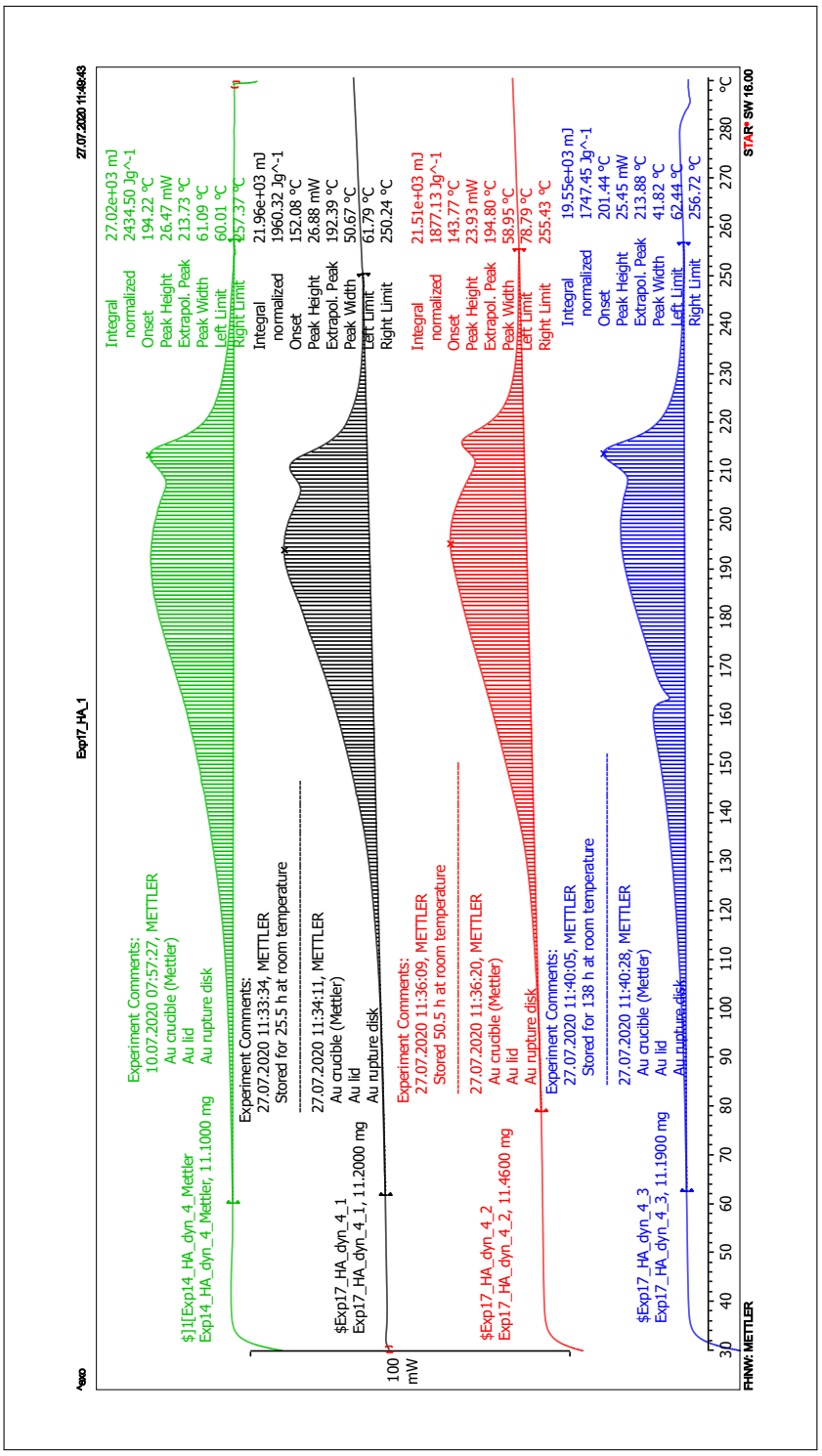
```
1  % ------------------------------------henry.m---------------------------------------
2  % Florian Walthert
3  % 04.07.20
4
5  function pi = henry(x, par, T)
6  if(~exist('T', 'var'))
7      T = 298.15; %K
8  end
9  pi = x ./ (0.018/997 * par(:,1) .* exp(par(:,2) .* (1./T - 1./298.15))); % [Sander]
10 end
11
12 % ----------------------------------------------------------------------------------
```

```
1  % ------------------------------------shomate.m-------------------------------------
2  % Florian Walthert
3  % 04.07.20
4
5  function [c_p, deltaH_ref_0, S] = shomate(T_K, par)
6  % if(~exist('comp', 'var'))
7      comp = 1:size(par,1);
8  % end
9  t=T_K./1000;
10 A = par(comp,1);B=par(comp,2);C=par(comp,3);D=par(comp,4);E=par(comp,5);
11 F=par(comp,6);G=par(comp,7);H=par(comp,8);
12 c_p = A + B.*t+ C.*t.^2 + D.*t.^3 + E./t.^2;
13 deltaH_ref_0 = A.*t + B.*t.^2./2 + C.*t.^3./3 + D.*t.^4./4 - E./t + F - H;
14 S = A*log(t) + B*t + C*t.^2./2 + D*t.^3./3 - E./(2*t.^2) + G;
15 end
16
17 % ----------------------------------------------------------------------------------
```

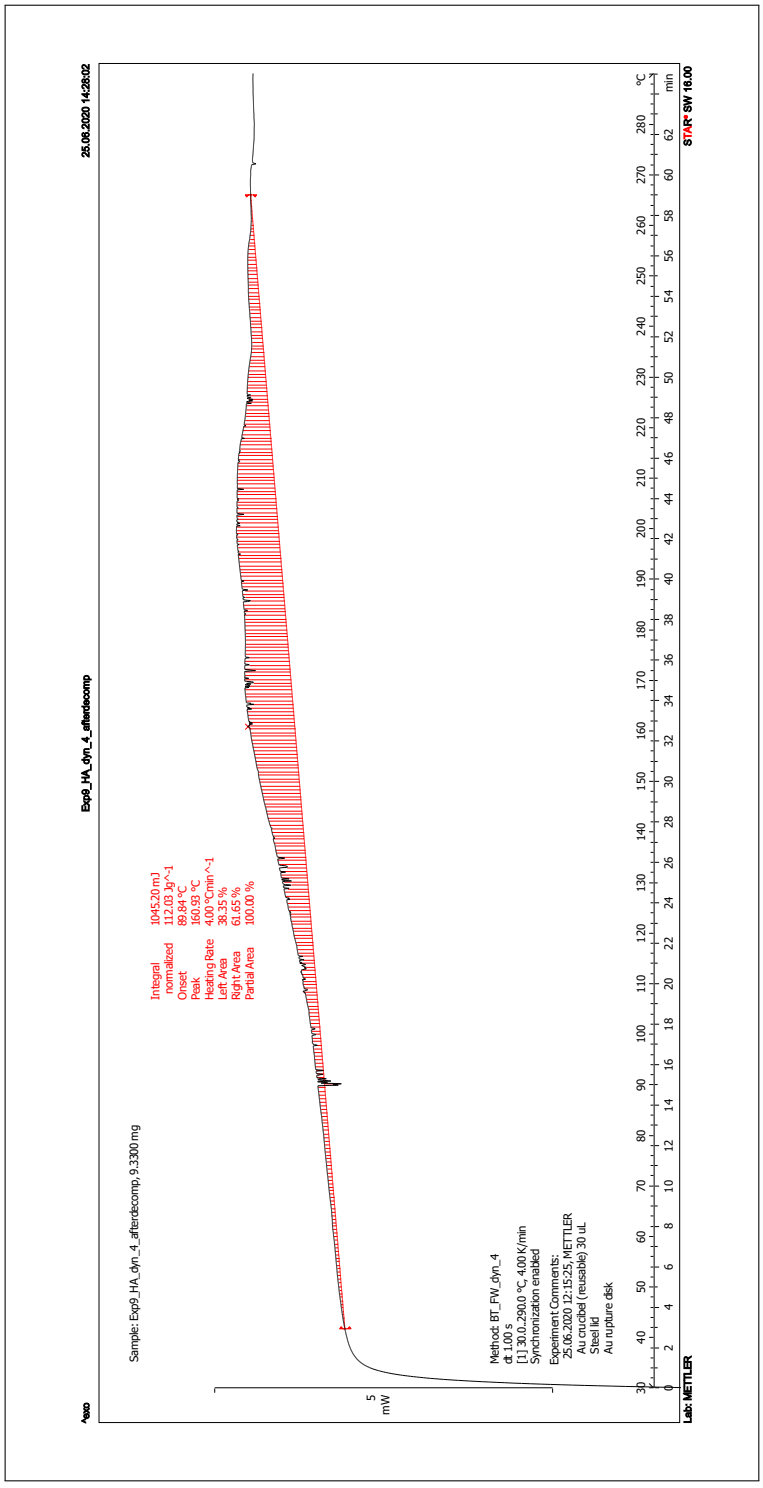
```
1  % ------------------------------------dU_V_ad.m-------------------------------------
2  % Florian Walthert
3  % 04.07.20
4
5  function dU_v = dU_V_ad(T_E, T_S, R, T_0, alpha, nu, n_t0, sd, Cv_int)
6
7  U_S = U_V(T_S, alpha(1), R, T_0, nu, n_t0, sd, Cv_int);
8  U_E = U_V(T_E, alpha(2), R, T_0, nu, n_t0, sd, Cv_int);
9
10 dU_v = U_E - U_S; % J/mol
11 end
12
13 % ----------------------------------------------------------------------------------
```

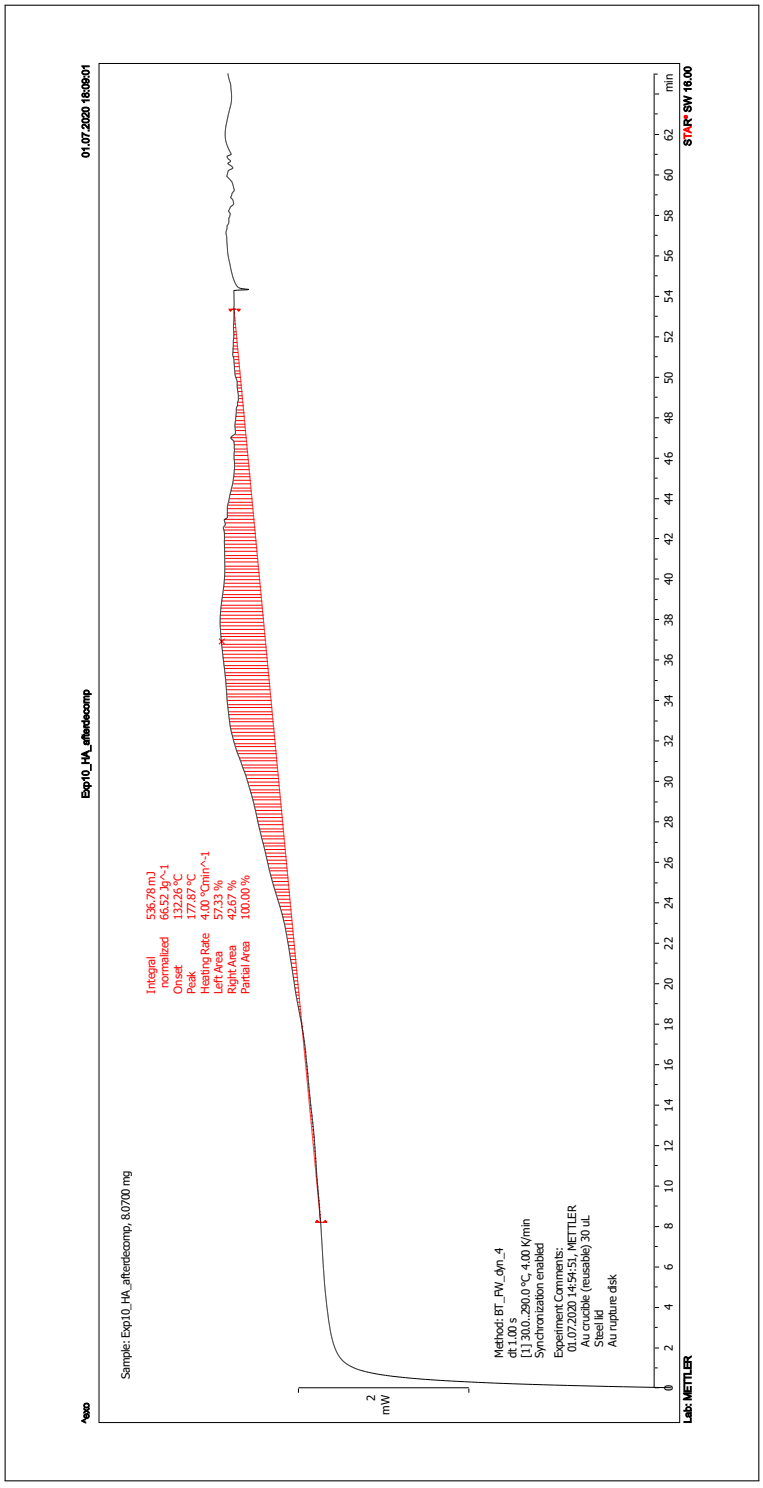
```
1  % ------------------------------------U_V.m-----------------------------------------
2  % Florian Walthert
3  % 04.07.20
4
5  function U = U_V(T, alpha, R, T_0, nu, n_t0, sd, Cv_int)
6
7  z = -1/nu(1)*alpha*n_t0(1); %mol
8  n = n_t0 + nu.*z; %mol
9
10 U_f = sd.deltaH_f_0 - R * T_0/1000 + Cv_int(T)/1000 - Cv_int(T_0)/1000; % [Gmehling p.102/334]
11 U = sum(U_f .* n); % kJ
12
13 end
14
15 % ----------------------------------------------------------------------------------
```

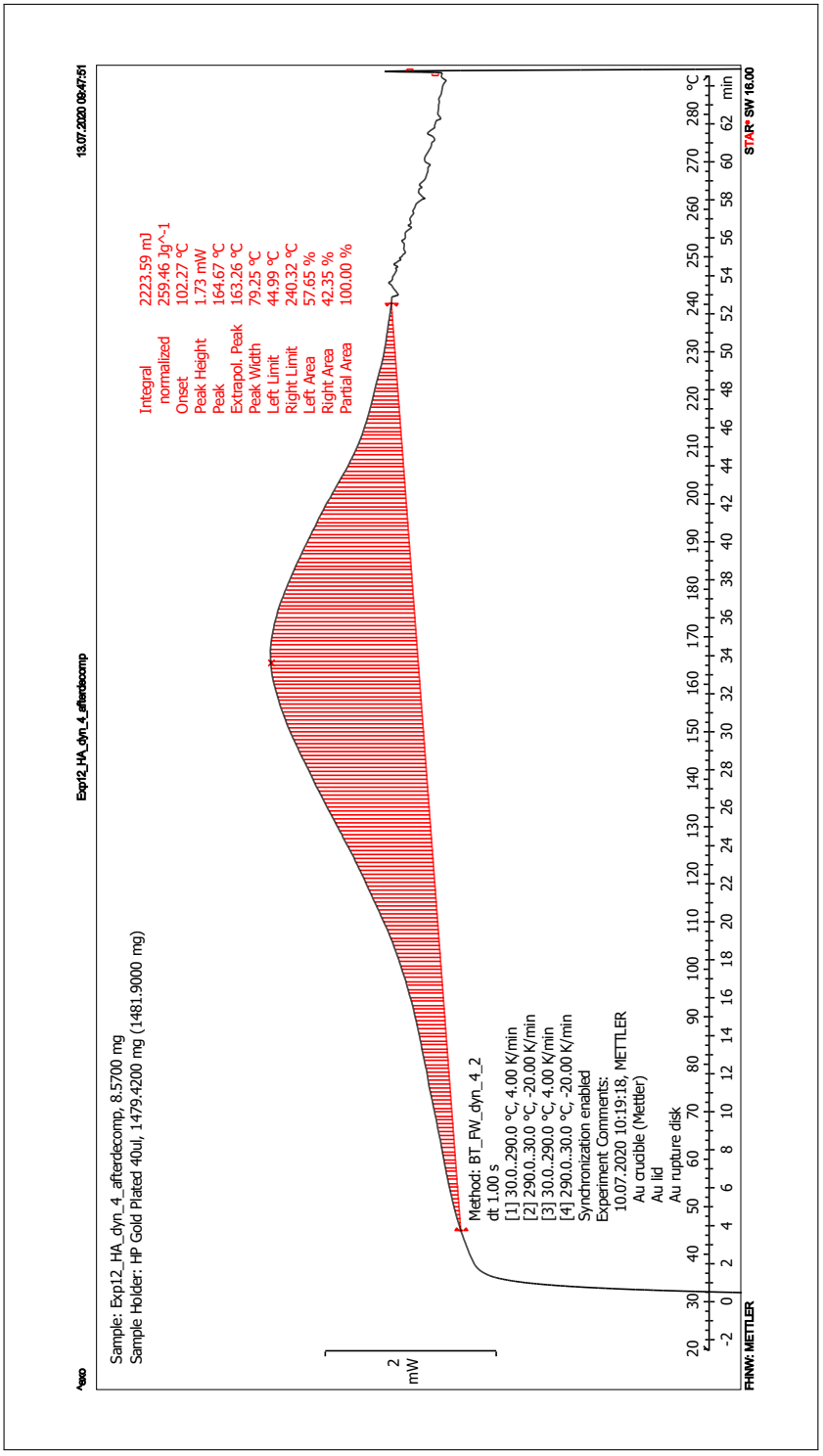
A.34 DSC Exp17 - Storage time experiment



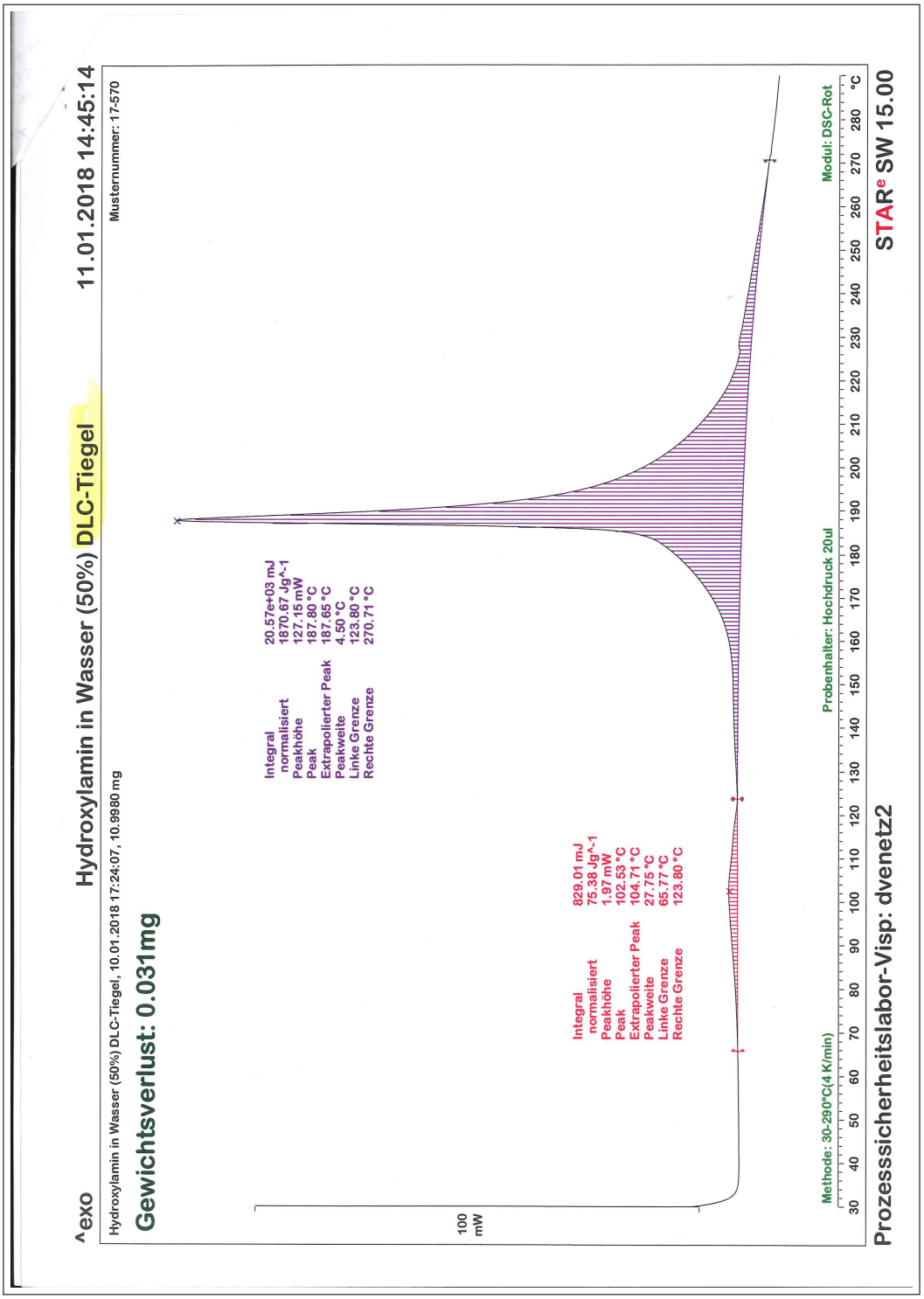
A.35 DSC Exp9/Exp10/Exp12 after EXR experiment



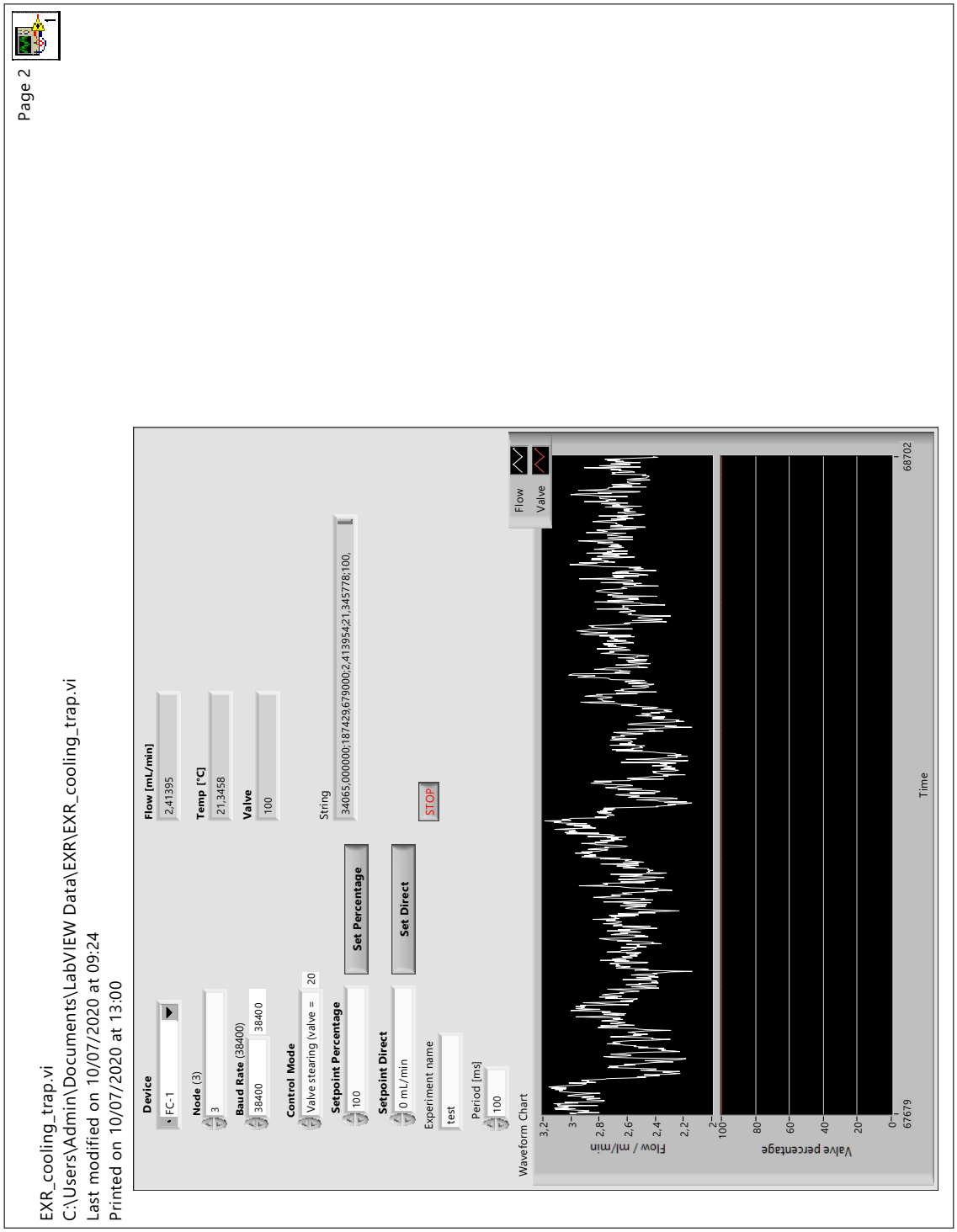

Exp10_HA_afterdecomp
01.07.2020 16:09:01
Sample: Exp10_HA_afterdecomp, 8.0700 mg
Integral 536.78 mJ
normalized 66.52 Jg^-1
Onset 132.26 °C
Peak 177.87 °C
Heating Rate 4.00 °Cmin^-1
Left Area 57.33 %
Right Area 42.67 %
Partial Area 100.00 %
Method: BT_FW_dyn_4
dt 1.00 s
[1] 30.0...290.0 °C, 4.00 K/min
Synchronization enabled
Experiment Comments:
01.07.2020 14:54:51, METTLER
Al crucible (reusable) 30 uL
Steel lid
Au rupture disk
2 mW
min
Lab: METTLER
STAR° SW 16.00

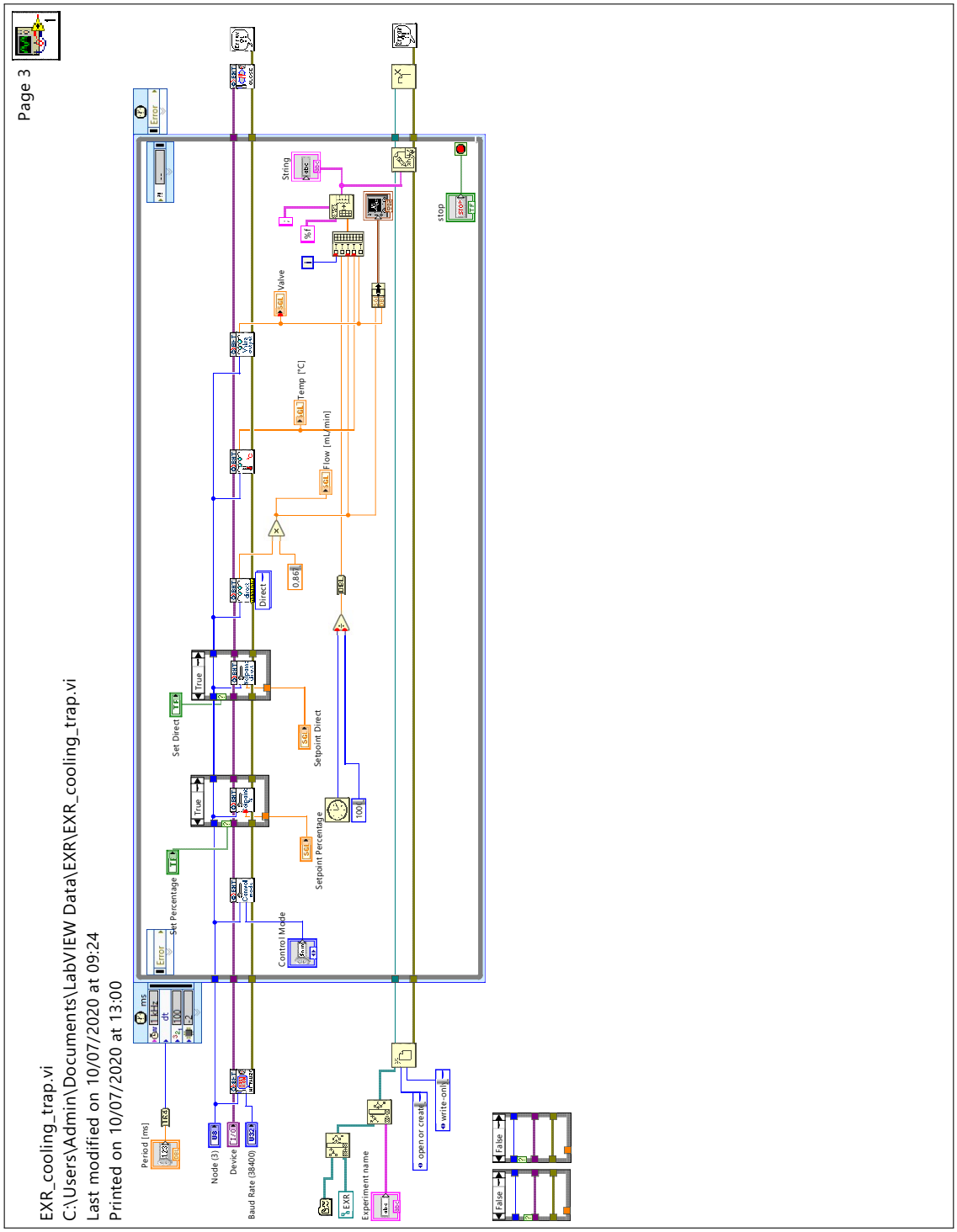

Exo
Exp12_HA_dyn_4_afterdecomp
13.07.2020 08:47:51
Sample: Exp12_HA_dyn_4_afterdecomp, 8.5700 mg
Sample Holder: HP Gold Plated 40ul, 1479.4200 mg (1481.9000 mg)
Integral 2223.59 mJ
normalized 259.46 Jg^-1
Onset 102.27 °C
Peak Height 1.73 mW
Peak 164.67 °C
Extrapol. Peak 163.26 °C
Peak Width 79.25 °C
Left Limit 44.99 °C
Right Limit 240.32 °C
Left Area 57.65 %
Right Area 42.35 %
Partial Area 100.00 %
2 mW
Method: BT_FW_dyn_4_2
dt 1.00 s
[1] 30.0..290.0 °C, 4.00 K/min
[2] 290.0..30.0 °C, -20.00 K/min
[3] 30.0..290.0 °C, 4.00 K/min
[4] 290.0..30.0 °C, -20.00 K/min
Synchronization enabled
Experiment Comments:
10.07.2020 10:19:18, METTLER
Au crucible (Mettler)
Au lid
Au rupture disk
FHNW: METTLER
STAR° SW 16.00
°C
min

A.36 DSC hydroxylamine solution in DLC crucible



A.37 LabVIEW program for Bronkhorst flow controller



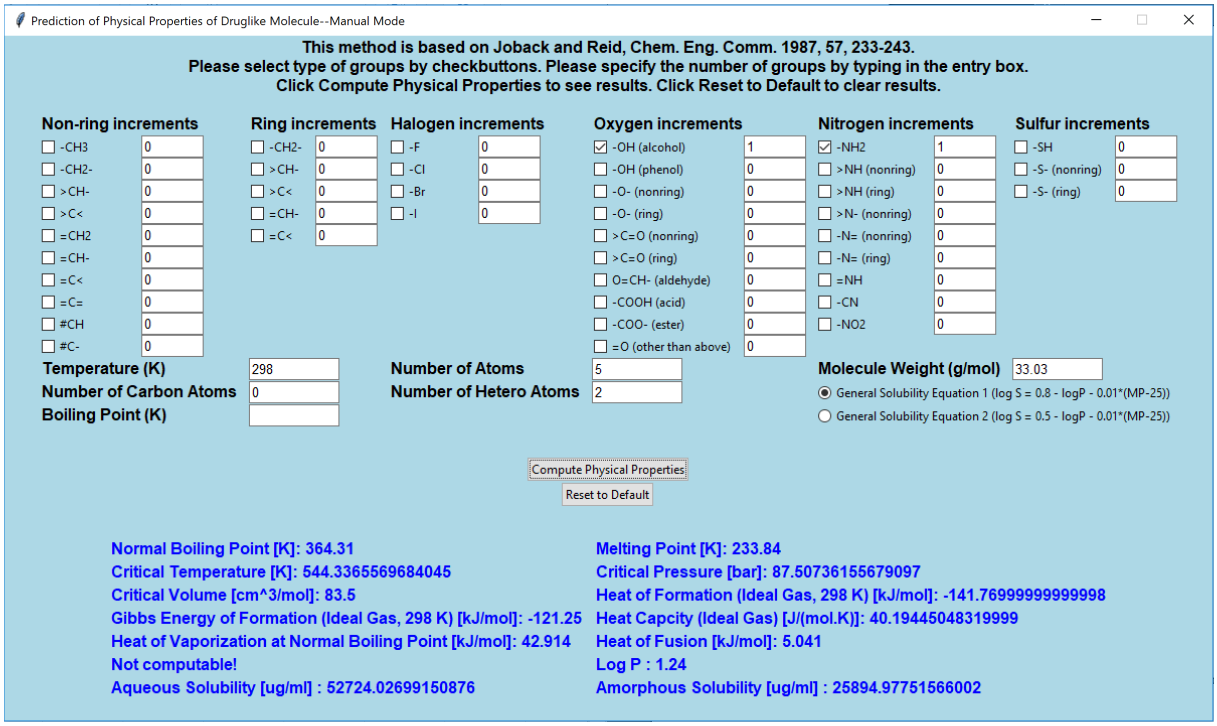

Page 3
EXR_cooling_trap.vi
C:\Users\Admin\Documents\LabVIEW Data\EXR\EXR_cooling_trap.vi
Last modified on 10/07/2020 at 09:24
Printed on 10/07/2020 at 13:00
Error
String
stop
Valve
Temp [°C]
Flow [mL/min]
0.864
Direct
True
Set Direct
Setpoint Direct
Set Percentage
Setpoint Percentage
100
Control Mode
Period [ms]
Node (3)
Device
Baud Rate (38400)
Experiment name
open or create
write-only
False

A.38 Relevant python code section in JRgui

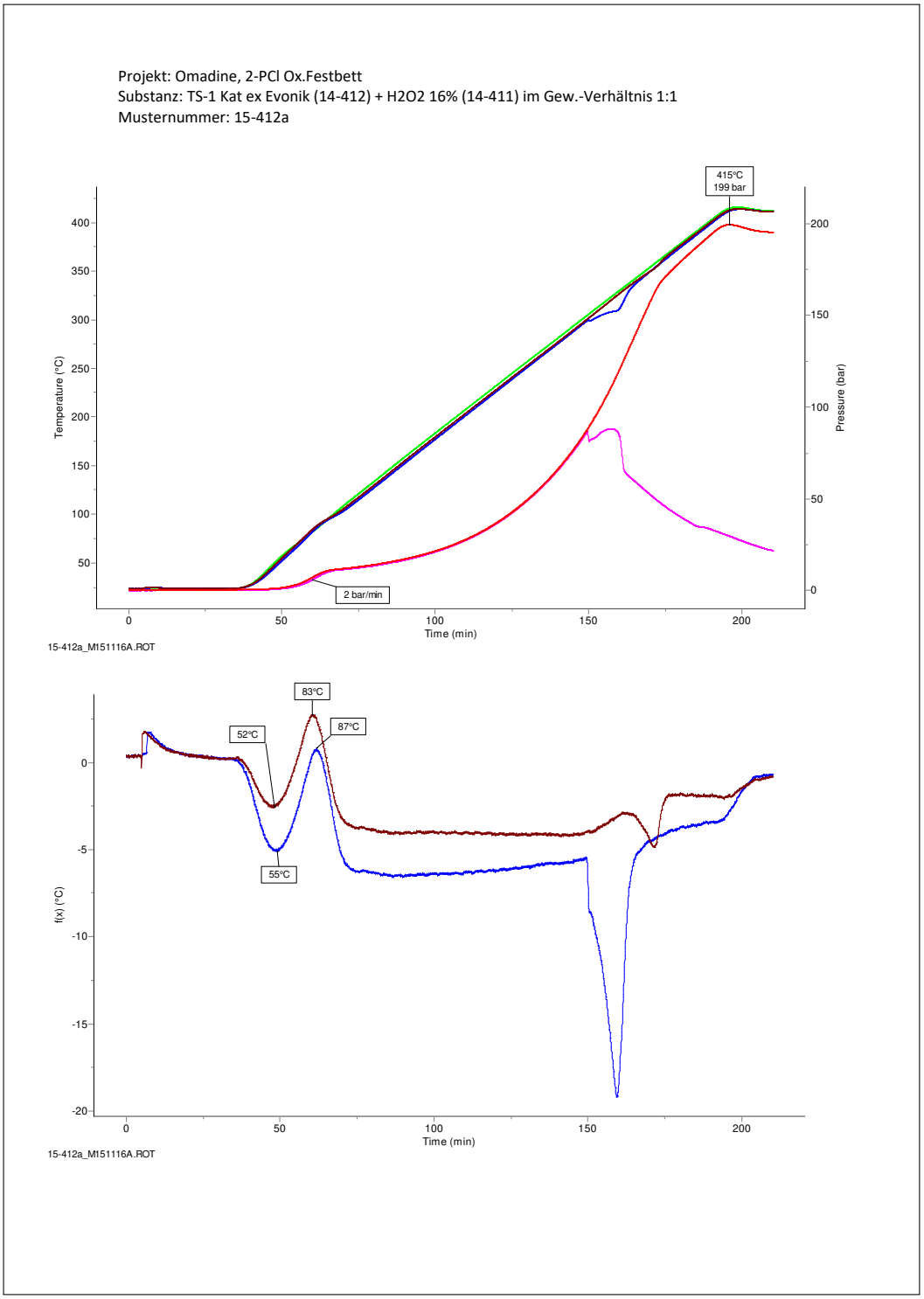
Python code to calculate substance properties via the *Joback* method (by *Shi and Borchardt*[**shi_jrgui_2017**]).

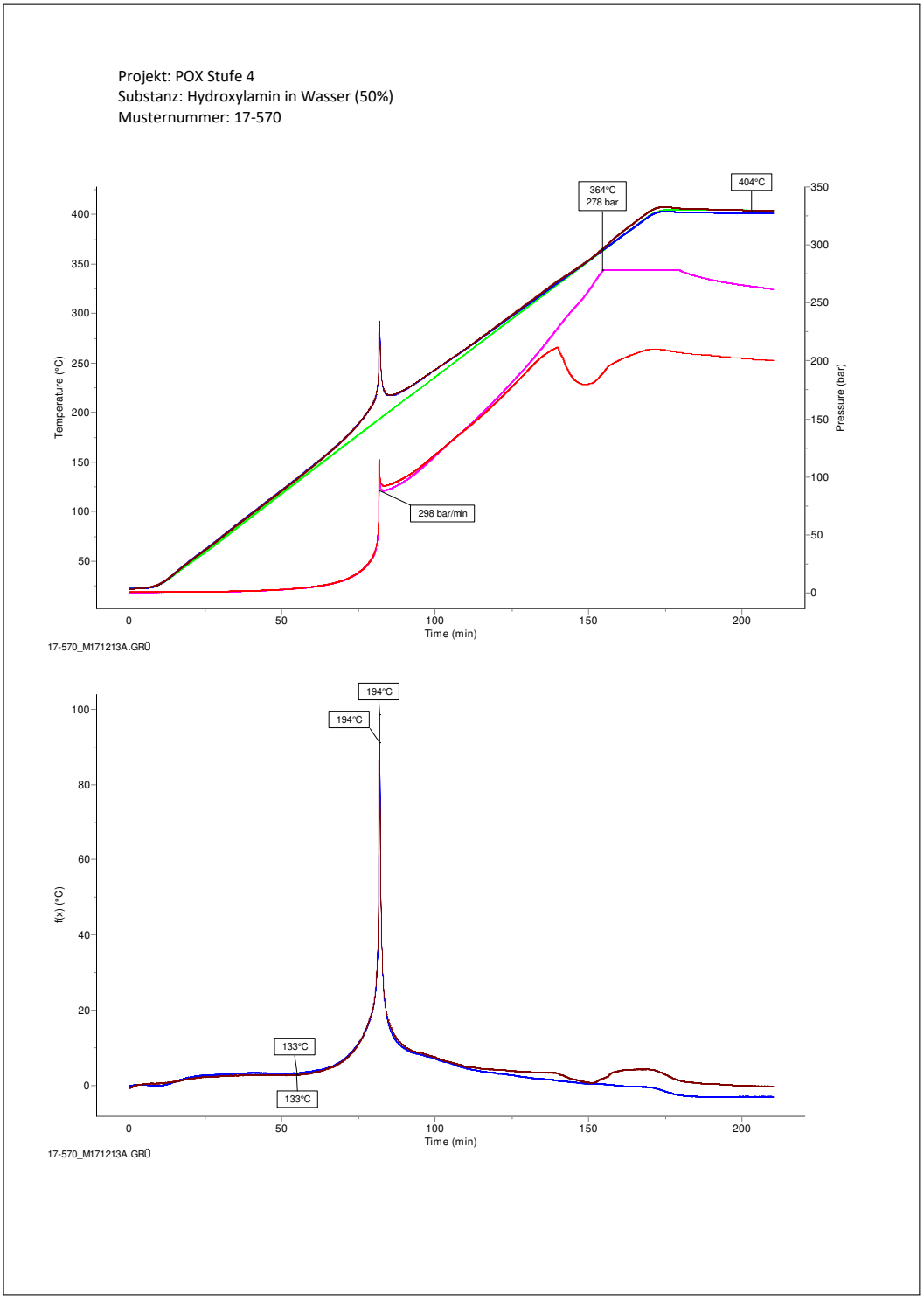
```
###normal boiling point
if self.BoilingPoint.get()== "":
        try:
                self.L1a.configure(text = "Normal Boiling Point [K]: {0}".format(198.2 + sum(Tb)))
        except:
                self.L1a.configure(text = "Not computable!")
else:
        self.L1a.configure(text = "Normal Boiling Point [K]: {0}"\
        .format(float(self.BoilingPoint.get())))
###melting point
try:
        self.L2a.configure(text = "Melting Point [K]: {0}".format(122.5 + sum(Tm)))
except:
        self.L2a.configure(text = "Not computable!")
###Critical Temperature
if self.BoilingPoint.get()== "":
        try:
                nom = sum(Tb) + 198.2
                denom = 0.584 + 0.965*sum(Tc) - sum(Tc)**2
                value = nom/denom
                self.L3a.configure(text = "Critical Temperature [K]: {0}".format(value))
        except:
                self.L3a.configure(text = "Not computable!", font = ('Arial', 12, 'bold'))
else:
        try:
                nom = float(self.BoilingPoint.get())
                denom = 0.584 + 0.965*sum(Tc) - sum(Tc)**2
                value = nom/denom
                self.L3a.configure(text = "Critical Temperature [K]: {0}".format(value))
        except:
                self.L3a.configure(text = "Not computable!", font = ('Arial', 12, 'bold'))
###Critical Pressure
try:
        denom = (0.113 + 0.0032*float(self.NoA) - sum(Pc))**2
        value = 1./denom
        self.L4a.configure(text = "Critical Pressure [bar]: {0}".format(value))
except:
        self.L4a.configure(text = "Not computable!")
###Critical Volume
try:
        self.L5a.configure(text = "Critical Volume [cm^3/mol]: {0}".format(17.5 + sum(Vc)))
except:
        self.L5a.configure(text = "Not computable!")
###Enthalpy of formation, ideal gas at 298 K
try:
        self.L6a.configure(text = "Heat of Formation \
        (Ideal Gas, 298 K) [kJ/mol]: {0}".format(68.29 + sum(Hfor)))
except:
        self.L6a.configure(text = "Not computable!")
####Gibbs energy off formation, ideal gas, unit fugacity, at 298 K
try:
        self.L7a.configure(text = "Gibbs Energy of Formation (Ideal Gas, 298 K) [kJ/mol]: \
        {0}".format(53.88 + sum(Gf)))
except:
        self.L7a.configure(text = "Not computable!")
###Heat capacity, ideal gas
try:
        term_1 = sum(Cpa) - 37.93
        term_2 = (sum(Cpb) + 0.210)*float(self._temperature.get())
        term_3 = (sum(Cpc) - 3.91*10**(-4))*float(self._temperature.get())**2
        term_4 = (sum(Cpd) + 2.06*10**(-7))*float(self._temperature.get())**3
        value = term_1 + term_2 + term_3 + term_4
        self.L8a.configure(text = "Heat Capcity (Ideal Gas) [J/(mol.K)]: {0}"\
        .format(value))
except:
        self.L8a.configure(text = "Not computable!")
###Enthalpy of vaporization at Tb
try:
        self.L9a.configure(text = "Heat of Vaporization at \
        Normal Boiling Point [kJ/mol]: {0}".format(15.30 + sum(Hvap)))
except:
        self.L9a.configure(text = "Not computable!")
###Enthalpy of fusion
try:
        self.L10a.configure(text = "Heat of Fusion [kJ/mol]: {0}".format(-0.88 + sum(Hfus)))
except:
        self.L10a.configure(text = "Not computable!")
```

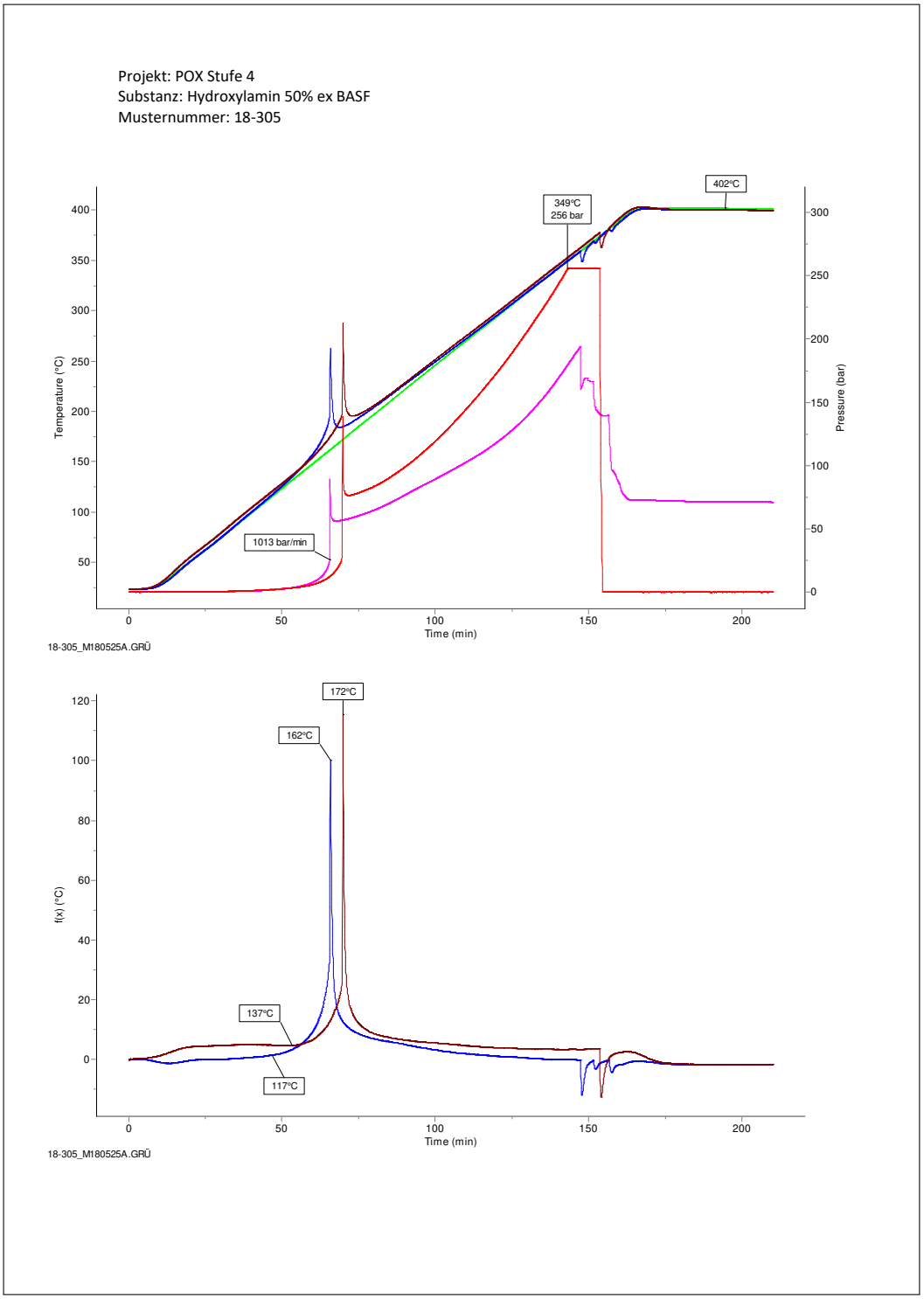
A.39 JRgui input mask

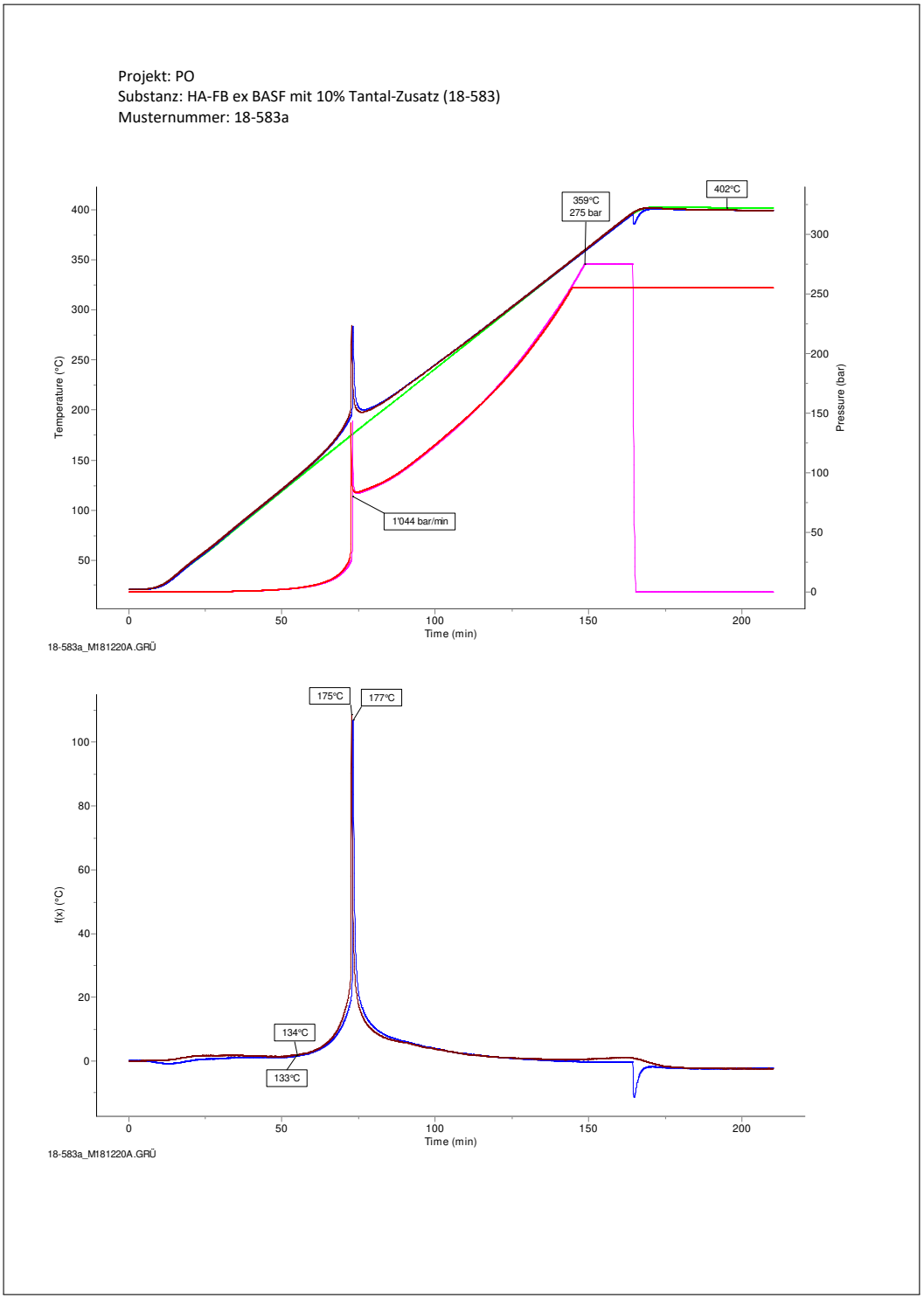


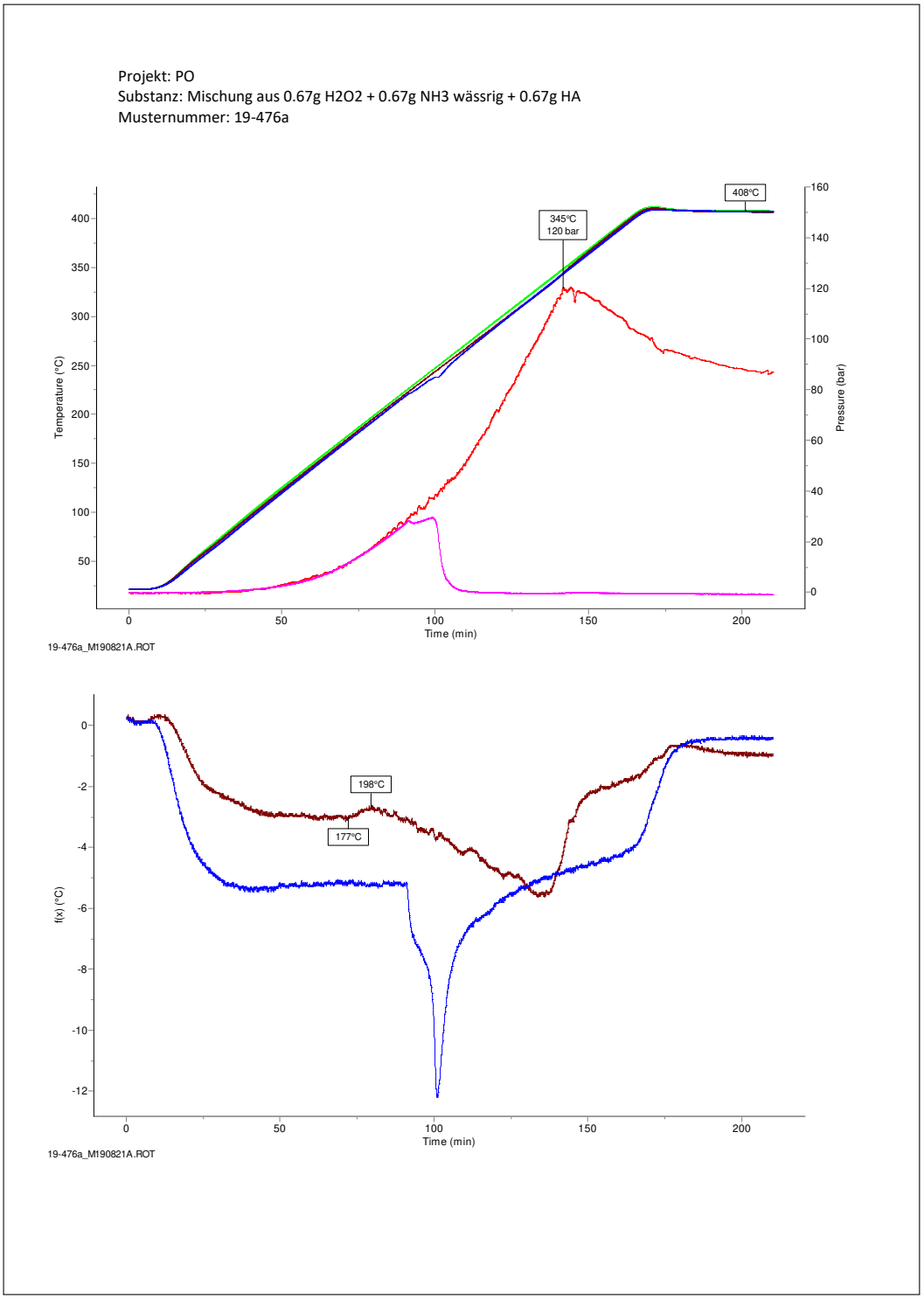
A.40 Miniautoclave measurements



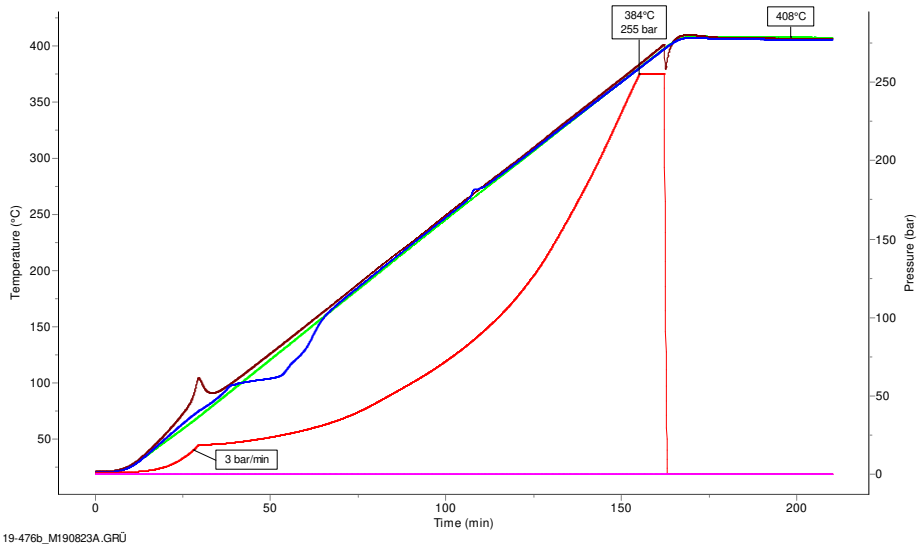

Projekt: POX Stufe 4
Substanz: Hydroxylamin in Wasser (50%)
Musternummer: 17-570
364°C
278 bar
404°C
298 bar/min
Temperature (°C)
Pressure (bar)
Time (min)
17-570_M171213A.GRÜ
194°C
194°C
133°C
133°C
f(x) (°C)
Time (min)
17-570_M171213A.GRÜ

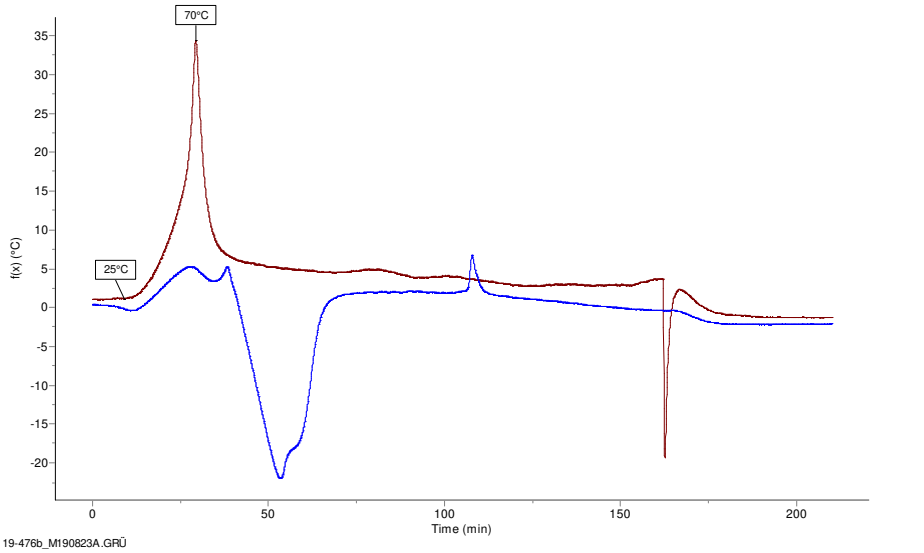

Projekt: POX Stufe 4
Substanz: Hydroxylamin 50% ex BASF
Musternummer: 18-305
402°C
349°C
256 bar
1013 bar/min
Temperature (°C)
Pressure (bar)
Time (min)
18-305_M180525A.GRÜ
172°C
162°C
137°C
117°C
f(x) (°C)
Time (min)
18-305_M180525A.GRÜ

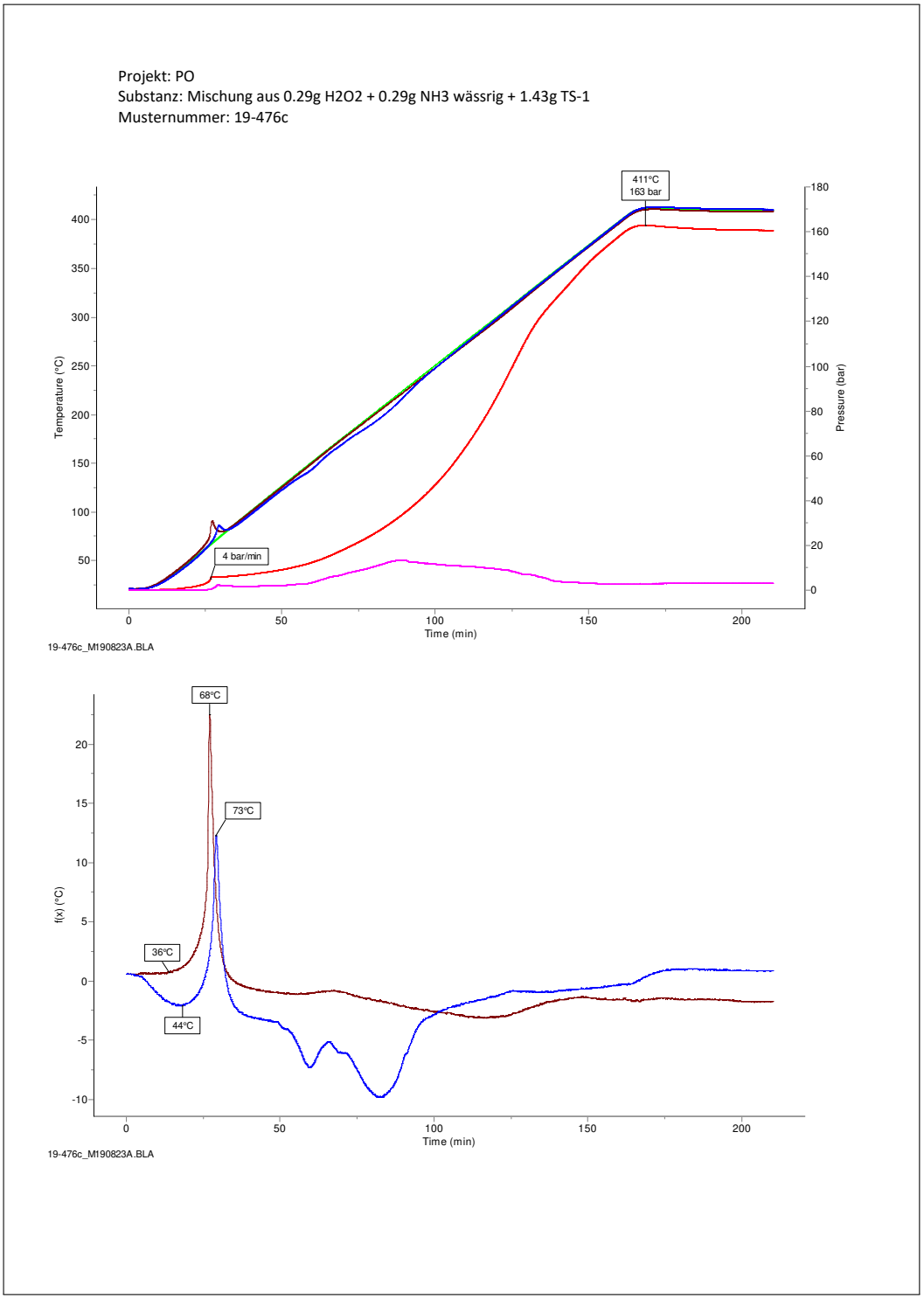

Projekt: PO
Substanz: HA-FB ex BASF mit 10% Tantal-Zusatz (18-583)
Musternummer: 18-583a
359°C
275 bar
402°C
1'044 bar/min
Temperature (°C)
Pressure (bar)
Time (min)
18-583a_M181220A.GRÜ
175°C
177°C
134°C
133°C
f(x) (°C)
Time (min)
18-583a_M181220A.GRÜ


Projekt: PO
Substanz: Mischung aus 0.67g H2O2 + 0.67g NH3 wässrig + 0.67g HA
Musternummer: 19-476a
408°C
345°C
120 bar
Temperature (°C)
Pressure (bar)
Time (min)
19-476a_M190821A.ROT
198°C
177°C
f(x) (°C)
Time (min)
19-476a_M190821A.ROT

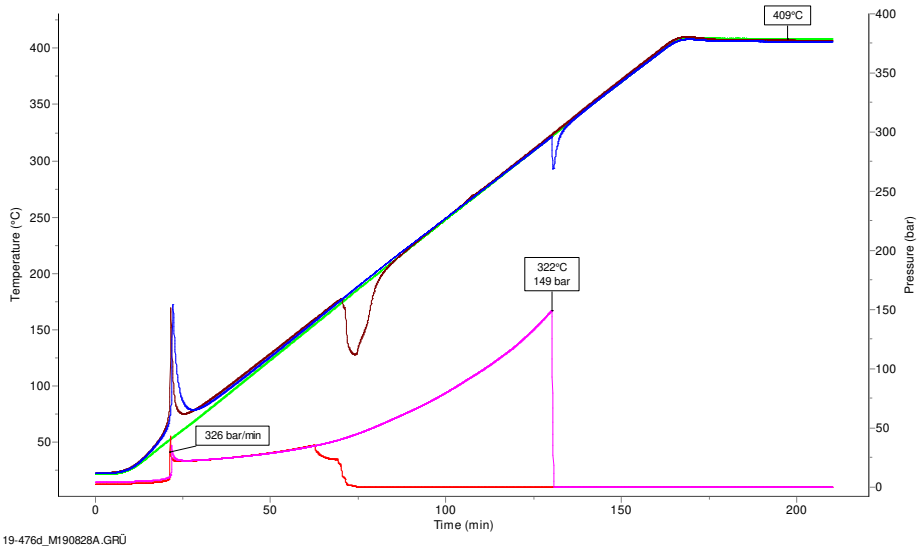
Projekt: PO
Substanz: Mischung aus 0.80g H2O2 + 0.80g NH3 wässrig + 0.40g TS-1
Musternummer: 19-476b

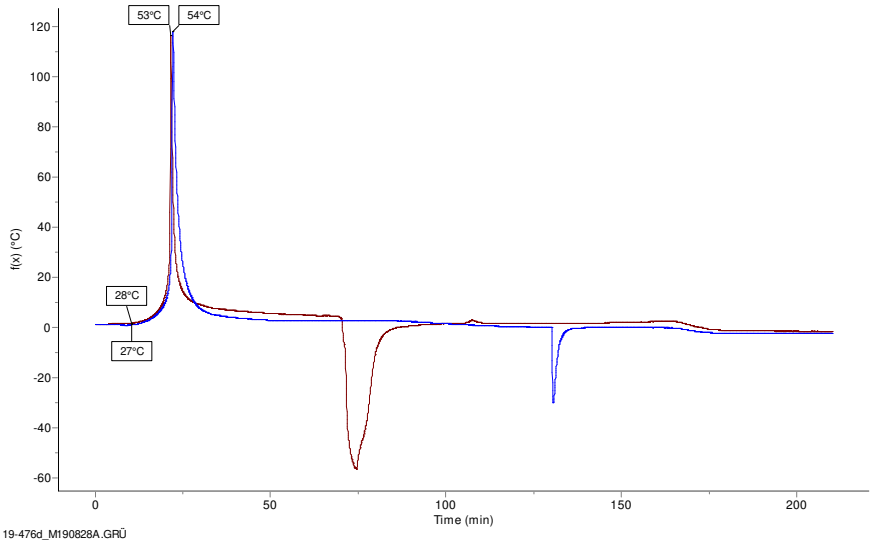


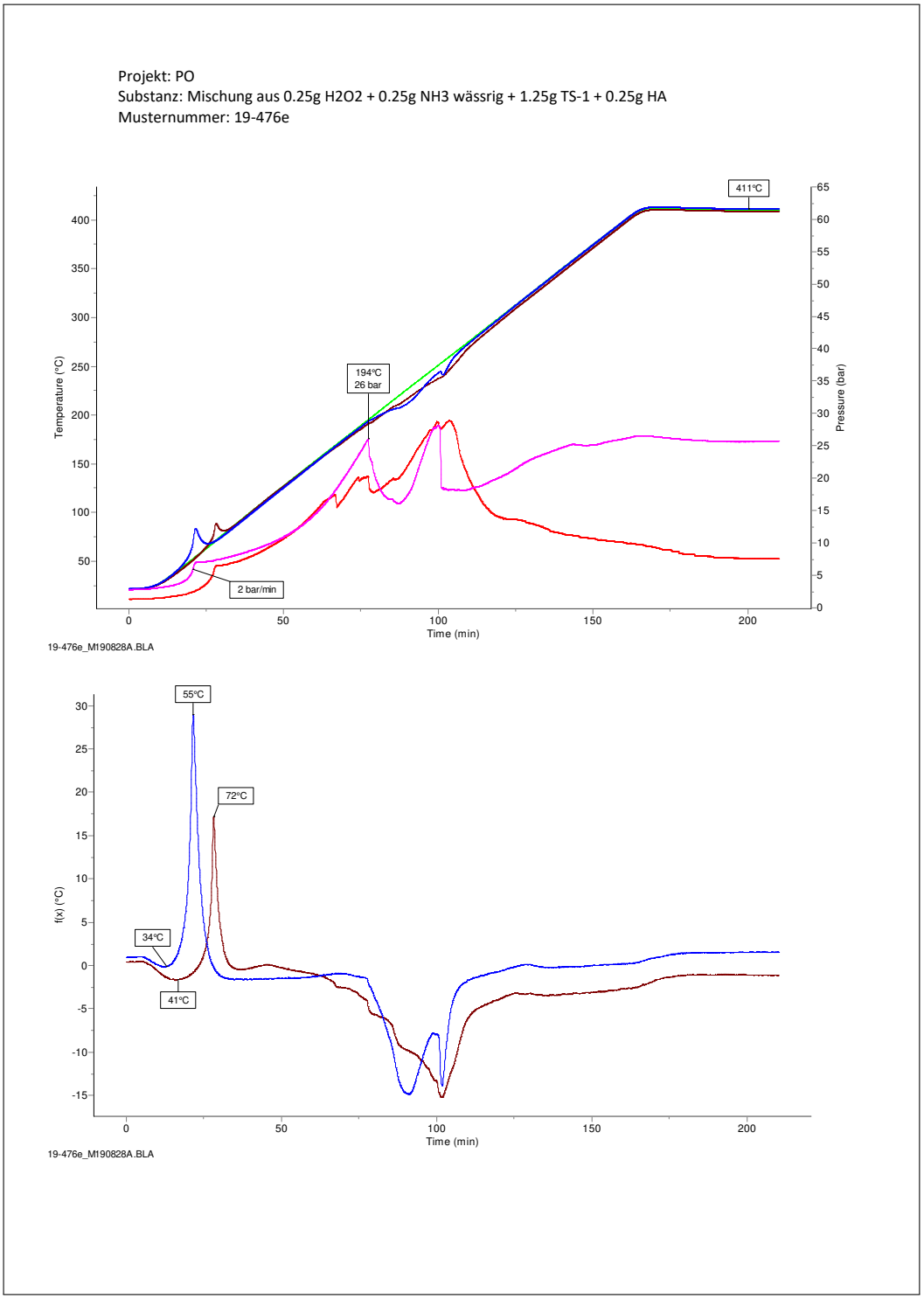



Projekt: PO
Substanz: Mischung aus 0.29g H2O2 + 0.29g NH3 wässrig + 1.43g TS-1
Musternummer: 19-476c
411°C
163 bar
4 bar/min
Temperature (°C)
Pressure (bar)
Time (min)
19-476c_M190823A.BLA
68°C
73°C
36°C
44°C
f(x) (°C)
Time (min)
19-476c_M190823A.BLA

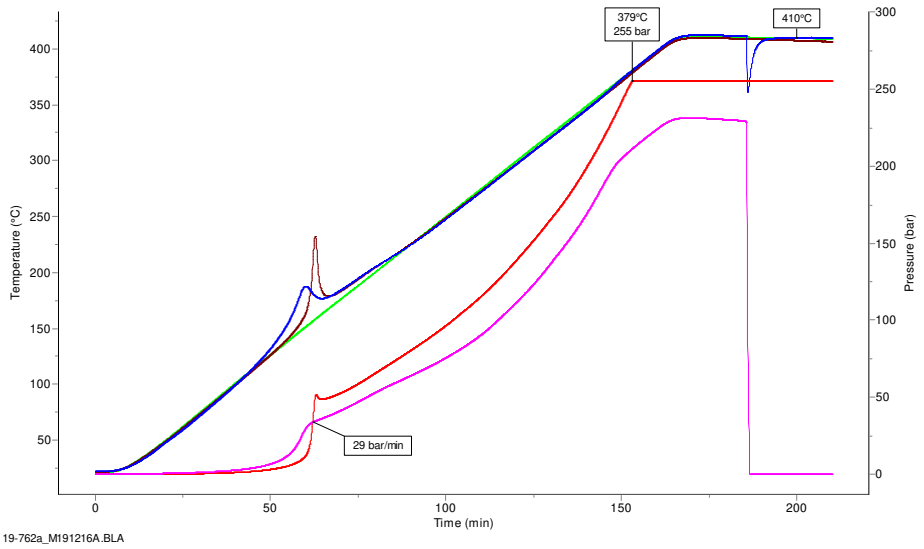
Projekt: PO
Substanz: Mischung aus 0.57g H2O2 + 0.57g NH3 wässrig + 0.29g TS-1 + 0.57g HA
Musternummer: 19-476d

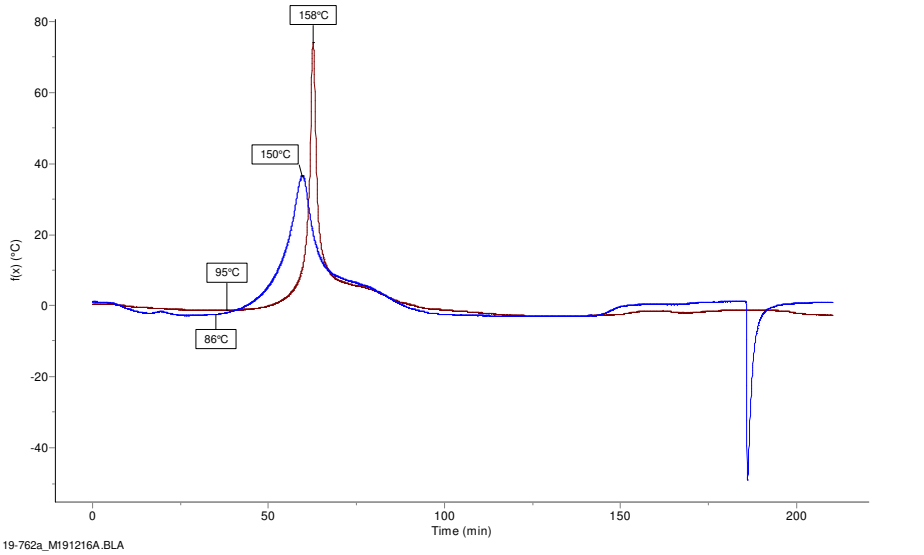


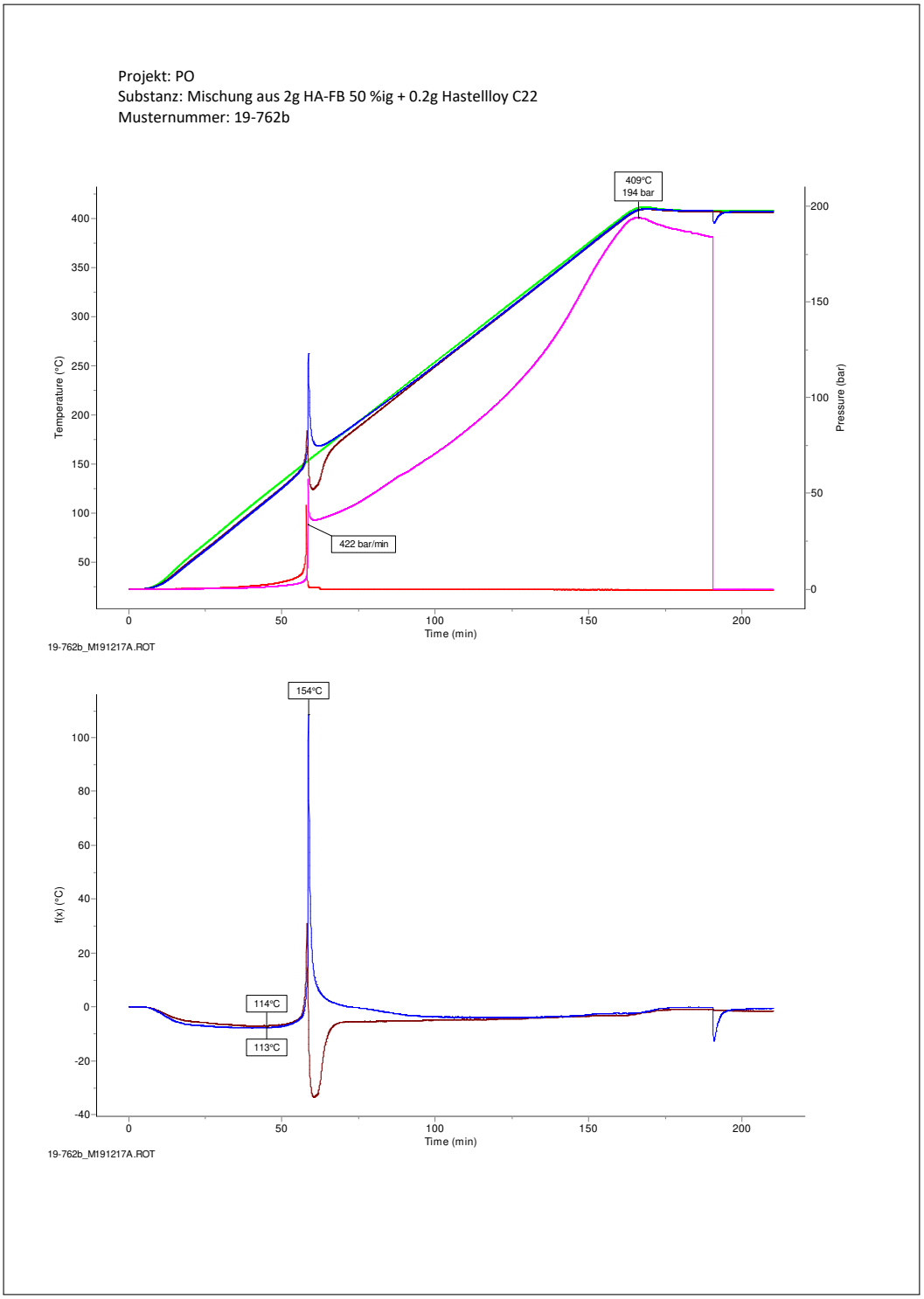



Projekt: PO
Substanz: Mischung aus 0.25g H2O2 + 0.25g NH3 wässrig + 1.25g TS-1 + 0.25g HA
Musternummer: 19-476e
411°C
194°C
26 bar
2 bar/min
Temperature (°C)
Pressure (bar)
Time (min)
19-476e_M190828A.BLA
55°C
72°C
34°C
41°C
f(x) (°C)
Time (min)
19-476e_M190828A.BLA

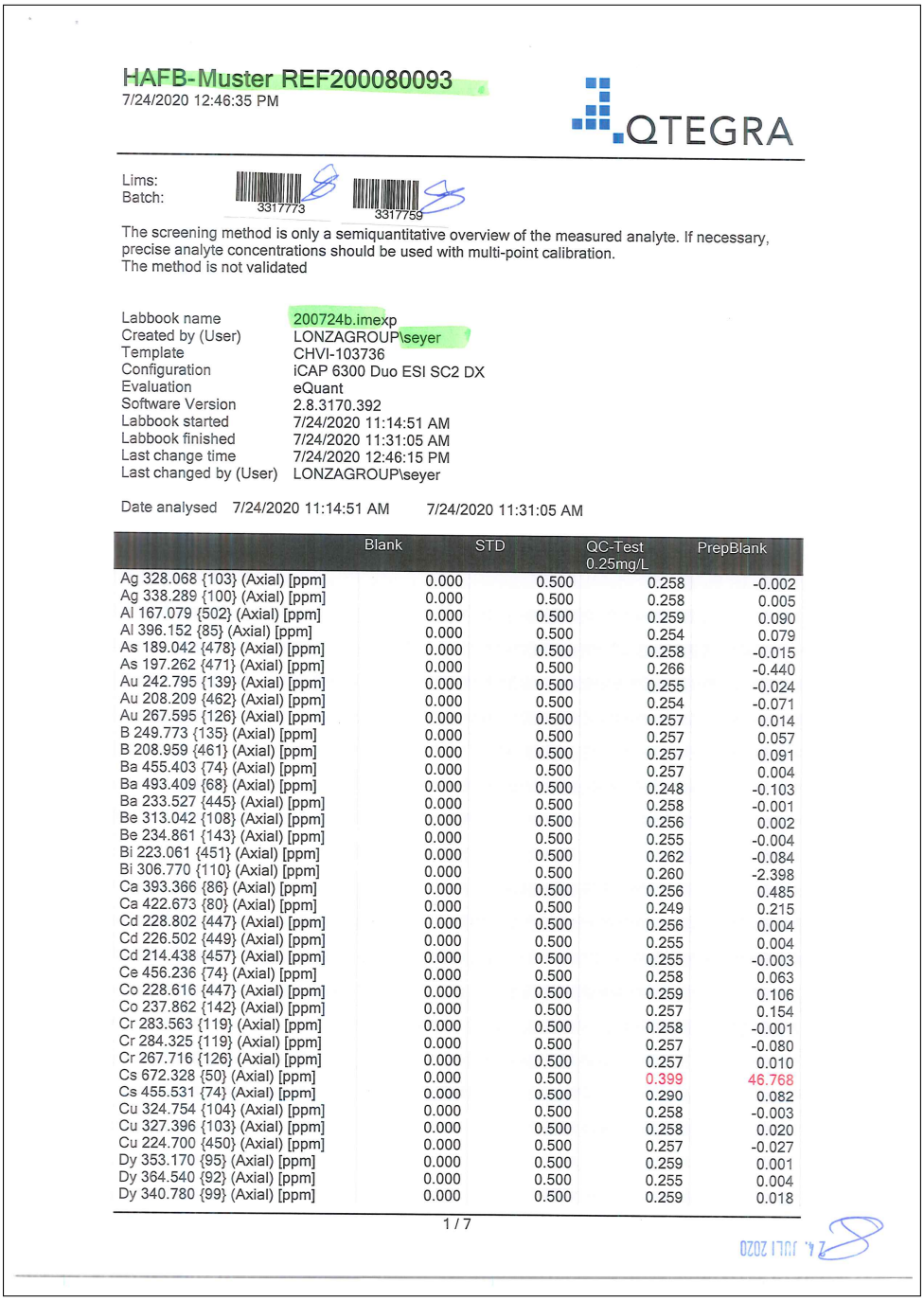
Projekt: PO
Substanz: Mischung aus 1.5g HA-FB 50 %ig + 0.5g Katalyt TS-1 + 0.2g Hastellloy C22
Musternummer: 19-762a






Projekt: PO
Substanz: Mischung aus 2g HA-FB 50 %ig + 0.2g Hastellloy C22
Musternummer: 19-762b
409°C
194 bar
422 bar/min
Temperature (°C)
Pressure (bar)
Time (min)
19-762b_M191217A.ROT
154°C
114°C
113°C
f(x) (°C)
Time (min)
19-762b_M191217A.ROT

A.41 Report from ICP-OES analysis of miniautoclave solution



HAFB-Muster REF200080093
7/24/2020 12:46:35 PM

QTEGRA

Lims: 3317773
Batch: 3317759

The screening method is only a semiquantitative overview of the measured analyte. If necessary, precise analyte concentrations should be used with multi-point calibration.
The method is not validated

Labbook name	200724b.imexp
Created by (User)	LONZAGROUP\seyer
Template	CHVI-103736
Configuration	iCAP 6300 Duo ESI SC2 DX
Evaluation	eQuant
Software Version	2.8.3170.392
Labbook started	7/24/2020 11:14:51 AM
Labbook finished	7/24/2020 11:31:05 AM
Last change time	7/24/2020 12:46:15 PM
Last changed by (User)	LONZAGROUP\seyer

Date analysed 7/24/2020 11:14:51 AM 7/24/2020 11:31:05 AM

	Blank	STD	QC-Test 0.25mg/L	PrepBlank
Ag 328.068 {103} (Axial) [ppm]	0.000	0.500	0.258	-0.002
Ag 338.289 {100} (Axial) [ppm]	0.000	0.500	0.258	0.005
Al 167.079 {502} (Axial) [ppm]	0.000	0.500	0.259	0.090
Al 396.152 {85} (Axial) [ppm]	0.000	0.500	0.254	0.079
As 189.042 {478} (Axial) [ppm]	0.000	0.500	0.258	-0.015
As 197.262 {471} (Axial) [ppm]	0.000	0.500	0.266	-0.440
Au 242.795 {139} (Axial) [ppm]	0.000	0.500	0.255	-0.024
Au 208.209 {462} (Axial) [ppm]	0.000	0.500	0.254	-0.071
Au 267.595 {126} (Axial) [ppm]	0.000	0.500	0.257	0.014
B 249.773 {135} (Axial) [ppm]	0.000	0.500	0.257	0.057
B 208.959 {461} (Axial) [ppm]	0.000	0.500	0.257	0.091
Ba 455.403 {74} (Axial) [ppm]	0.000	0.500	0.257	0.004
Ba 493.409 {68} (Axial) [ppm]	0.000	0.500	0.248	-0.103
Ba 233.527 {445} (Axial) [ppm]	0.000	0.500	0.258	-0.001
Be 313.042 {108} (Axial) [ppm]	0.000	0.500	0.256	0.002
Be 234.861 {143} (Axial) [ppm]	0.000	0.500	0.255	-0.004
Bi 223.061 {451} (Axial) [ppm]	0.000	0.500	0.262	-0.084
Bi 306.770 {110} (Axial) [ppm]	0.000	0.500	0.260	-2.398
Ca 393.366 {86} (Axial) [ppm]	0.000	0.500	0.256	0.485
Ca 422.673 {80} (Axial) [ppm]	0.000	0.500	0.249	0.215
Cd 228.802 {447} (Axial) [ppm]	0.000	0.500	0.256	0.004
Cd 226.502 {449} (Axial) [ppm]	0.000	0.500	0.255	0.004
Cd 214.438 {457} (Axial) [ppm]	0.000	0.500	0.255	-0.003
Ce 456.236 {74} (Axial) [ppm]	0.000	0.500	0.258	0.063
Co 228.616 {447} (Axial) [ppm]	0.000	0.500	0.259	0.106
Co 237.862 {142} (Axial) [ppm]	0.000	0.500	0.257	0.154
Cr 283.563 {119} (Axial) [ppm]	0.000	0.500	0.258	-0.001
Cr 284.325 {119} (Axial) [ppm]	0.000	0.500	0.257	-0.080
Cr 267.716 {126} (Axial) [ppm]	0.000	0.500	0.257	0.010
Cs 672.328 {50} (Axial) [ppm]	0.000	0.500	0.399	46.768
Cs 455.531 {74} (Axial) [ppm]	0.000	0.500	0.290	0.082
Cu 324.754 {104} (Axial) [ppm]	0.000	0.500	0.258	-0.003
Cu 327.396 {103} (Axial) [ppm]	0.000	0.500	0.258	0.020
Cu 224.700 {450} (Axial) [ppm]	0.000	0.500	0.257	-0.027
Dy 353.170 {95} (Axial) [ppm]	0.000	0.500	0.259	0.001
Dy 364.540 {92} (Axial) [ppm]	0.000	0.500	0.255	0.004
Dy 340.780 {99} (Axial) [ppm]	0.000	0.500	0.259	0.018

1 / 7

24. JULI 2020

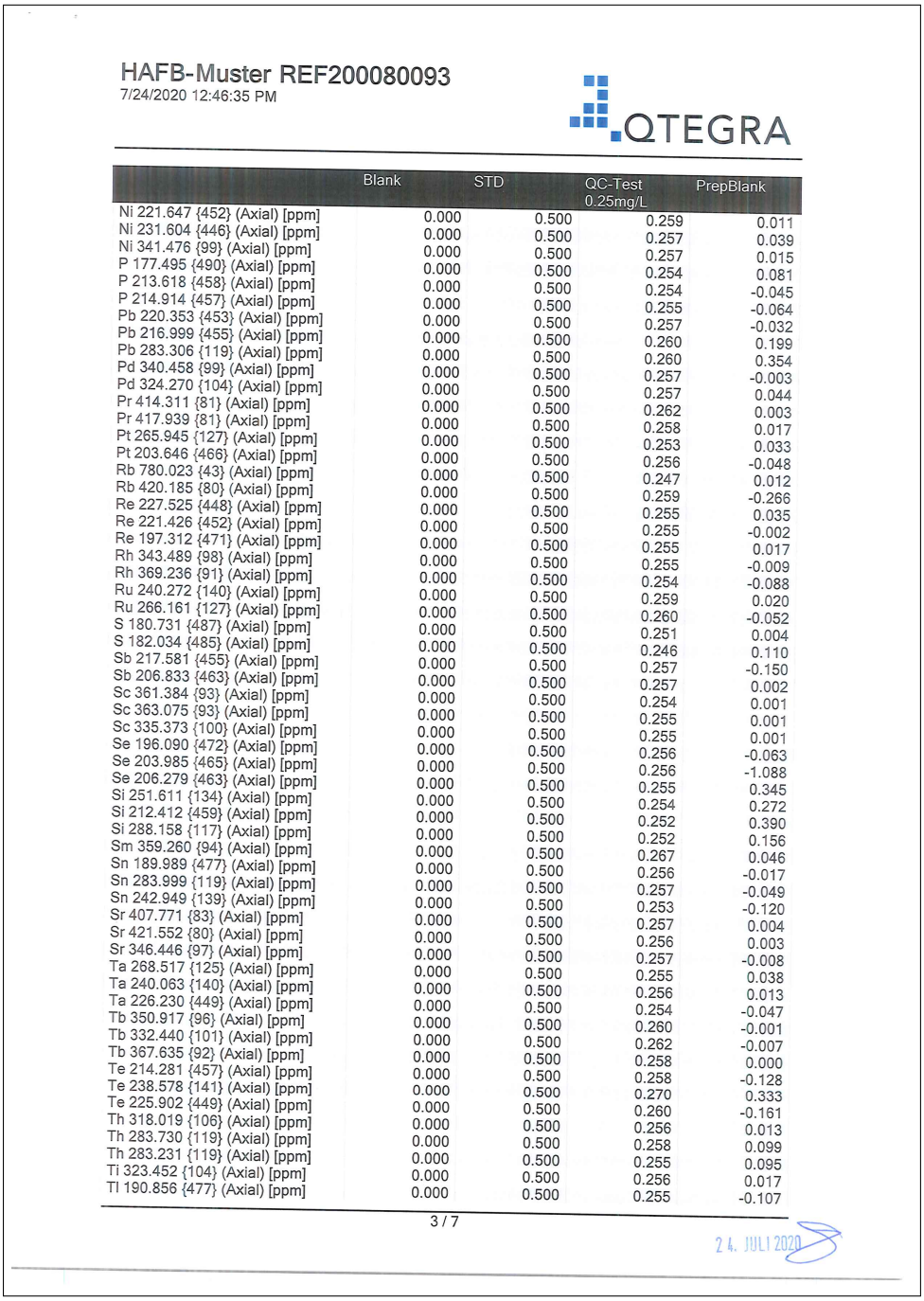
HAFB-Muster REF200080093
7/24/2020 12:46:35 PM

QTEGRA

	Blank	STD	QC-Test 0.25mg/L	PrepBlank
Er 337.271 {100} (Axial) [ppm]	0.000	0.500	0.258	-0.001
Er 326.478 {103} (Axial) [ppm]	0.000	0.500	0.259	0.066
Er 323.058 {104} (Axial) [ppm]	0.000	0.500	0.259	-0.003
Eu 381.967 {88} (Axial) [ppm]	0.000	0.500	0.258	-0.001
Eu 420.505 {80} (Axial) [ppm]	0.000	0.500	0.257	-0.001
Eu 412.970 {82} (Axial) [ppm]	0.000	0.500	0.258	0.004
Fe 259.940 {130} (Axial) [ppm]	0.000	0.500	0.258	0.073
Fe 238.204 {141} (Axial) [ppm]	0.000	0.500	0.256	0.071
Fe 239.562 {141} (Axial) [ppm]	0.000	0.500	0.258	0.080
Ga 294.364 {114} (Axial) [ppm]	0.000	0.500	0.255	-0.007
Ga 417.206 {81} (Axial) [ppm]	0.000	0.500	0.254	-0.003
Ga 287.424 {117} (Axial) [ppm]	0.000	0.500	0.256	0.018
Gd 342.247 {98} (Axial) [ppm]	0.000	0.500	0.260	-0.005
Gd 336.223 {100} (Axial) [ppm]	0.000	0.500	0.259	0.112
Gd 335.047 {101} (Axial) [ppm]	0.000	0.500	0.258	-0.014
Ge 265.118 {127} (Axial) [ppm]	0.000	0.500	0.260	0.058
Ge 219.871 {453} (Axial) [ppm]	0.000	0.500	0.256	-0.273
Ge 206.866 {463} (Axial) [ppm]	0.000	0.500	0.260	-0.122
Hf 339.980 {99} (Axial) [ppm]	0.000	0.500	0.256	0.064
Hf 263.871 {128} (Axial) [ppm]	0.000	0.500	0.257	-0.017
Hf 232.247 {445} (Axial) [ppm]	0.000	0.500	0.256	0.015
Hg 194.227 {474} (Axial) [ppm]	0.000	0.500	0.256	0.053
Hg 184.950 {482} (Axial) [ppm]	0.000	0.500	0.256	0.024
Hg 253.652 {133} (Axial) [ppm]	0.000	0.500	0.259	-0.098
Ho 345.600 {98} (Axial) [ppm]	0.000	0.500	0.260	0.001
Ho 389.102 {87} (Axial) [ppm]	0.000	0.500	0.264	-0.037
In 325.609 {103} (Axial) [ppm]	0.000	0.500	0.255	0.262
In 230.606 {446} (Axial) [ppm]	0.000	0.500	0.259	0.077
In 303.936 {111} (Axial) [ppm]	0.000	0.500	0.258	-0.097
Ir 224.268 {450} (Axial) [ppm]	0.000	0.500	0.256	0.025
Ir 212.681 {458} (Axial) [ppm]	0.000	0.500	0.256	-0.001
Ir 205.222 {464} (Axial) [ppm]	0.000	0.500	0.263	-0.151
K 766.490 {44} (Axial) [ppm]	0.000	0.500	0.243	0.010
K 769.896 {44} (Axial) [ppm]	0.000	0.500	0.245	0.037
La 412.323 {82} (Axial) [ppm]	0.000	0.500	0.259	0.000
La 333.749 {101} (Axial) [ppm]	0.000	0.500	0.259	0.002
La 379.478 {89} (Axial) [ppm]	0.000	0.500	0.257	0.003
Li 670.784 {50} (Axial) [ppm]	0.000	0.500	0.243	0.000
Li 460.286 {73} (Axial) [ppm]	0.000	0.500	0.246	-0.172
Lu 261.542 {129} (Axial) [ppm]	0.000	0.500	0.255	0.002
Lu 291.139 {116} (Axial) [ppm]	0.000	0.500	0.257	0.004
Lu 307.760 {109} (Axial) [ppm]	0.000	0.500	0.256	0.087
Mg 279.553 {121} (Axial) [ppm]	0.000	0.500	0.256	0.123
Mg 280.270 {120} (Axial) [ppm]	0.000	0.500	0.255	0.122
Mg 285.213 {118} (Axial) [ppm]	0.000	0.500	0.255	0.082
Mn 257.610 {131} (Axial) [ppm]	0.000	0.500	0.256	0.005
Mn 259.373 {130} (Axial) [ppm]	0.000	0.500	0.255	0.000
Mn 260.569 {129} (Axial) [ppm]	0.000	0.500	0.257	0.004
Mo 202.030 {467} (Axial) [ppm]	0.000	0.500	0.254	0.011
Mo 281.615 {120} (Axial) [ppm]	0.000	0.500	0.257	-0.021
Mo 204.598 {465} (Axial) [ppm]	0.000	0.500	0.255	0.008
Na 589.592 {57} (Axial) [ppm]	0.000	0.500	0.244	1.006
Na 818.326 {41} (Axial) [ppm]	0.000	0.500	0.243	0.587
Na 330.237 {102} (Axial) [ppm]	0.000	0.500	0.259	0.036
Nb 316.340 {107} (Axial) [ppm]	0.000	0.500	0.256	0.008
Nd 401.225 {84} (Axial) [ppm]	0.000	0.500	0.258	0.009
Nd 430.358 {78} (Axial) [ppm]	0.000	0.500	0.261	-0.027
Nd 406.109 {83} (Axial) [ppm]	0.000	0.500	0.259	-0.021

2 / 7

24. JULI 2020



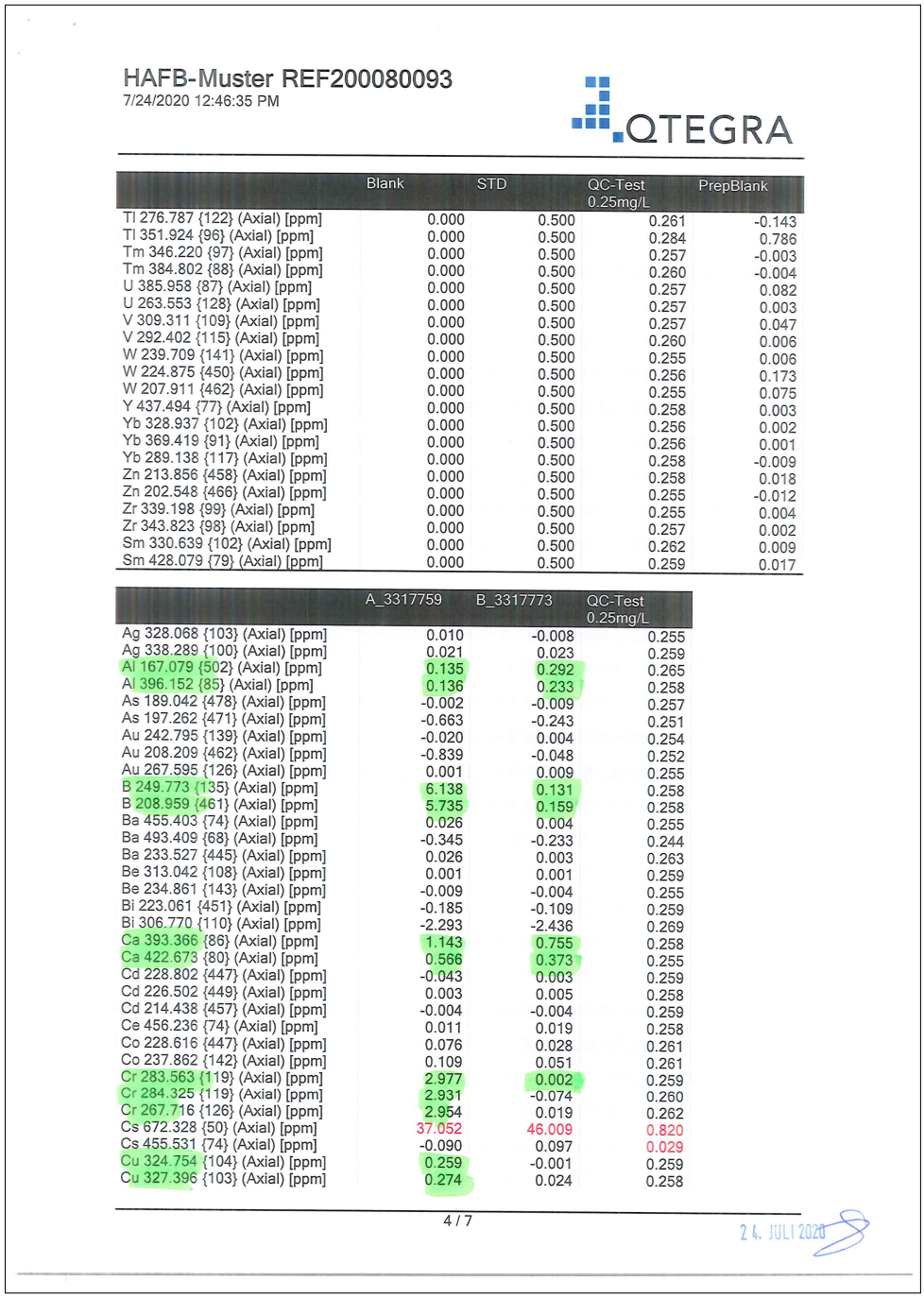
HAFB-Muster REF200080093
7/24/2020 12:46:35 PM

QTEGRA

	Blank	STD	QC-Test 0.25mg/L	PrepBlank
Ni 221.647 {452} (Axial) [ppm]	0.000	0.500	0.259	0.011
Ni 231.604 {446} (Axial) [ppm]	0.000	0.500	0.257	0.039
Ni 341.476 {99} (Axial) [ppm]	0.000	0.500	0.257	0.015
P 177.495 {490} (Axial) [ppm]	0.000	0.500	0.254	0.081
P 213.618 {458} (Axial) [ppm]	0.000	0.500	0.254	-0.045
P 214.914 {457} (Axial) [ppm]	0.000	0.500	0.255	-0.064
Pb 220.353 {453} (Axial) [ppm]	0.000	0.500	0.257	-0.032
Pb 216.999 {455} (Axial) [ppm]	0.000	0.500	0.260	0.199
Pb 283.306 {119} (Axial) [ppm]	0.000	0.500	0.260	0.354
Pd 340.458 {99} (Axial) [ppm]	0.000	0.500	0.257	-0.003
Pd 324.270 {104} (Axial) [ppm]	0.000	0.500	0.257	0.044
Pr 414.311 {81} (Axial) [ppm]	0.000	0.500	0.262	0.003
Pr 417.939 {81} (Axial) [ppm]	0.000	0.500	0.258	0.017
Pt 265.945 {127} (Axial) [ppm]	0.000	0.500	0.253	0.033
Pt 203.646 {466} (Axial) [ppm]	0.000	0.500	0.256	-0.048
Rb 780.023 {43} (Axial) [ppm]	0.000	0.500	0.247	0.012
Rb 420.185 {80} (Axial) [ppm]	0.000	0.500	0.259	-0.266
Re 227.525 {448} (Axial) [ppm]	0.000	0.500	0.255	0.035
Re 221.426 {452} (Axial) [ppm]	0.000	0.500	0.255	-0.002
Re 197.312 {471} (Axial) [ppm]	0.000	0.500	0.255	0.017
Rh 343.489 {98} (Axial) [ppm]	0.000	0.500	0.255	-0.009
Rh 369.236 {91} (Axial) [ppm]	0.000	0.500	0.254	-0.088
Ru 240.272 {140} (Axial) [ppm]	0.000	0.500	0.259	0.020
Ru 266.161 {127} (Axial) [ppm]	0.000	0.500	0.260	-0.052
S 180.731 {487} (Axial) [ppm]	0.000	0.500	0.251	0.004
S 182.034 {485} (Axial) [ppm]	0.000	0.500	0.246	0.110
Sb 217.581 {455} (Axial) [ppm]	0.000	0.500	0.257	-0.150
Sb 206.833 {463} (Axial) [ppm]	0.000	0.500	0.257	0.002
Sc 361.384 {93} (Axial) [ppm]	0.000	0.500	0.254	0.001
Sc 363.075 {93} (Axial) [ppm]	0.000	0.500	0.255	0.001
Sc 335.373 {100} (Axial) [ppm]	0.000	0.500	0.255	0.001
Se 196.090 {472} (Axial) [ppm]	0.000	0.500	0.256	-0.063
Se 203.985 {465} (Axial) [ppm]	0.000	0.500	0.256	-1.088
Se 206.279 {463} (Axial) [ppm]	0.000	0.500	0.255	0.345
Si 251.611 {134} (Axial) [ppm]	0.000	0.500	0.254	0.272
Si 212.412 {459} (Axial) [ppm]	0.000	0.500	0.252	0.390
Si 288.158 {117} (Axial) [ppm]	0.000	0.500	0.252	0.156
Sm 359.260 {94} (Axial) [ppm]	0.000	0.500	0.267	0.046
Sn 189.989 {477} (Axial) [ppm]	0.000	0.500	0.256	-0.017
Sn 283.999 {119} (Axial) [ppm]	0.000	0.500	0.257	-0.049
Sn 242.949 {139} (Axial) [ppm]	0.000	0.500	0.253	-0.120
Sr 407.771 {83} (Axial) [ppm]	0.000	0.500	0.257	0.004
Sr 421.552 {80} (Axial) [ppm]	0.000	0.500	0.256	0.003
Sr 346.446 {97} (Axial) [ppm]	0.000	0.500	0.257	-0.008
Ta 268.517 {125} (Axial) [ppm]	0.000	0.500	0.255	0.038
Ta 240.063 {140} (Axial) [ppm]	0.000	0.500	0.256	0.013
Ta 226.230 {449} (Axial) [ppm]	0.000	0.500	0.254	-0.047
Tb 350.917 {96} (Axial) [ppm]	0.000	0.500	0.260	-0.001
Tb 332.440 {101} (Axial) [ppm]	0.000	0.500	0.262	-0.007
Tb 367.635 {92} (Axial) [ppm]	0.000	0.500	0.258	0.000
Te 214.281 {457} (Axial) [ppm]	0.000	0.500	0.258	-0.128
Te 238.578 {141} (Axial) [ppm]	0.000	0.500	0.270	0.333
Te 225.902 {449} (Axial) [ppm]	0.000	0.500	0.260	-0.161
Th 318.019 {106} (Axial) [ppm]	0.000	0.500	0.256	0.013
Th 283.730 {119} (Axial) [ppm]	0.000	0.500	0.258	0.099
Th 283.231 {119} (Axial) [ppm]	0.000	0.500	0.255	0.095
Ti 323.452 {104} (Axial) [ppm]	0.000	0.500	0.256	0.017
Tl 190.856 {477} (Axial) [ppm]	0.000	0.500	0.255	-0.107

3 / 7

24. JULI 2020



HAFB-Muster REF200080093
7/24/2020 12:46:35 PM

QTEGRA

	Blank	STD	QC-Test 0.25mg/L	PrepBlank
Tl 276.787 {122} (Axial) [ppm]	0.000	0.500	0.261	-0.143
Tl 351.924 {96} (Axial) [ppm]	0.000	0.500	0.284	0.786
Tm 346.220 {97} (Axial) [ppm]	0.000	0.500	0.257	-0.003
Tm 384.802 {88} (Axial) [ppm]	0.000	0.500	0.260	-0.004
U 385.958 {87} (Axial) [ppm]	0.000	0.500	0.257	0.082
U 263.553 {128} (Axial) [ppm]	0.000	0.500	0.257	0.003
V 309.311 {109} (Axial) [ppm]	0.000	0.500	0.257	0.047
V 292.402 {115} (Axial) [ppm]	0.000	0.500	0.260	0.006
W 239.709 {141} (Axial) [ppm]	0.000	0.500	0.255	0.006
W 224.875 {450} (Axial) [ppm]	0.000	0.500	0.256	0.173
W 207.911 {462} (Axial) [ppm]	0.000	0.500	0.255	0.075
Y 437.494 {77} (Axial) [ppm]	0.000	0.500	0.258	0.003
Yb 328.937 {102} (Axial) [ppm]	0.000	0.500	0.256	0.002
Yb 369.419 {91} (Axial) [ppm]	0.000	0.500	0.256	0.001
Yb 289.138 {117} (Axial) [ppm]	0.000	0.500	0.258	-0.009
Zn 213.856 {458} (Axial) [ppm]	0.000	0.500	0.258	0.018
Zn 202.548 {466} (Axial) [ppm]	0.000	0.500	0.255	-0.012
Zr 339.198 {99} (Axial) [ppm]	0.000	0.500	0.255	0.004
Zr 343.823 {98} (Axial) [ppm]	0.000	0.500	0.257	0.002
Sm 330.639 {102} (Axial) [ppm]	0.000	0.500	0.262	0.009
Sm 428.079 {79} (Axial) [ppm]	0.000	0.500	0.259	0.017

	A_3317759	B_3317773	QC-Test 0.25mg/L
Ag 328.068 {103} (Axial) [ppm]	0.010	-0.008	0.255
Ag 338.289 {100} (Axial) [ppm]	0.021	0.023	0.259
Al 167.079 {502} (Axial) [ppm]	0.135	0.292	0.265
Al 396.152 {85} (Axial) [ppm]	0.136	0.233	0.258
As 189.042 {478} (Axial) [ppm]	-0.002	-0.009	0.257
As 197.262 {471} (Axial) [ppm]	-0.663	-0.243	0.251
Au 242.795 {139} (Axial) [ppm]	-0.020	0.004	0.254
Au 208.209 {462} (Axial) [ppm]	-0.839	-0.048	0.252
Au 267.595 {126} (Axial) [ppm]	0.001	0.009	0.255
B 249.773 {135} (Axial) [ppm]	6.138	0.131	0.258
B 208.959 {461} (Axial) [ppm]	5.735	0.159	0.258
Ba 455.403 {74} (Axial) [ppm]	0.026	0.004	0.255
Ba 493.409 {68} (Axial) [ppm]	-0.345	-0.233	0.244
Ba 233.527 {445} (Axial) [ppm]	0.026	0.003	0.263
Be 313.042 {108} (Axial) [ppm]	0.001	0.001	0.259
Be 234.861 {143} (Axial) [ppm]	-0.009	-0.004	0.255
Bi 223.061 {451} (Axial) [ppm]	-0.185	-0.109	0.259
Bi 306.770 {110} (Axial) [ppm]	-2.293	-2.436	0.269
Ca 393.366 {86} (Axial) [ppm]	1.143	0.755	0.258
Ca 422.673 {80} (Axial) [ppm]	0.566	0.373	0.255
Cd 228.802 {447} (Axial) [ppm]	-0.043	0.003	0.259
Cd 226.502 {449} (Axial) [ppm]	0.003	0.005	0.258
Cd 214.438 {457} (Axial) [ppm]	-0.004	-0.004	0.259
Ce 456.236 {74} (Axial) [ppm]	0.011	0.019	0.258
Co 228.616 {447} (Axial) [ppm]	0.076	0.028	0.261
Co 237.862 {142} (Axial) [ppm]	0.109	0.051	0.261
Cr 283.563 {119} (Axial) [ppm]	2.977	0.002	0.259
Cr 284.325 {119} (Axial) [ppm]	2.931	-0.074	0.260
Cr 267.716 {126} (Axial) [ppm]	2.954	0.019	0.262
Cs 672.328 {50} (Axial) [ppm]	37.052	46.009	0.820
Cs 455.531 {74} (Axial) [ppm]	-0.090	0.097	0.029
Cu 324.754 {104} (Axial) [ppm]	0.259	-0.001	0.259
Cu 327.396 {103} (Axial) [ppm]	0.274	0.024	0.258

4 / 7

24. JULI 2020

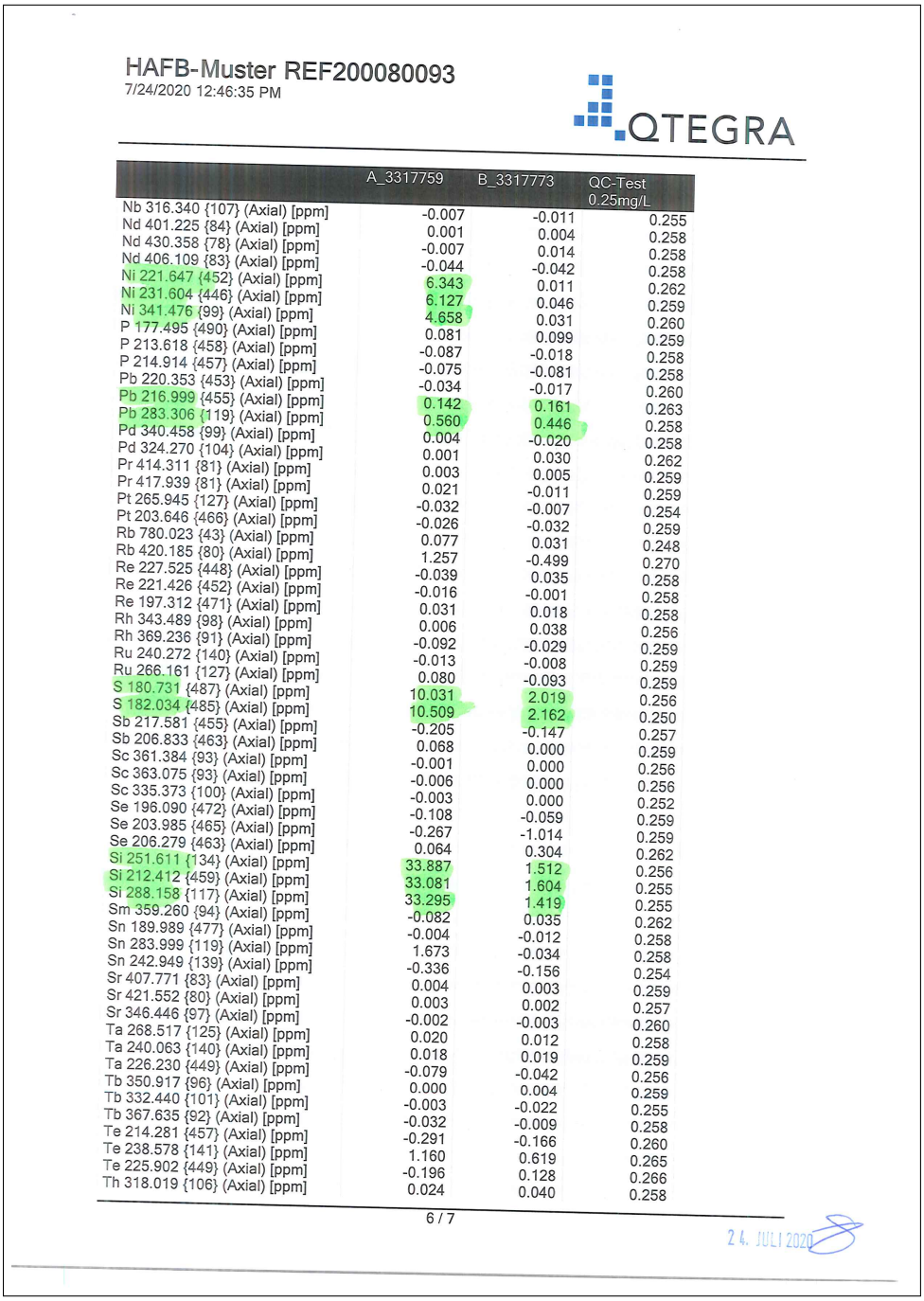
HAFB-Muster REF200080093
7/24/2020 12:46:35 PM

QTEGRA

	A_3317759	B_3317773	QC-Test 0.25mg/L
Cu 224.700 {450} (Axial) [ppm]	0.212	-0.024	0.260
Dy 353.170 {95} (Axial) [ppm]	0.000	-0.002	0.258
Dy 364.540 {92} (Axial) [ppm]	0.001	-0.001	0.257
Dy 340.780 {99} (Axial) [ppm]	0.014	0.006	0.259
Er 337.271 {100} (Axial) [ppm]	0.027	-0.002	0.255
Er 326.478 {103} (Axial) [ppm]	0.040	0.057	0.256
Er 323.058 {104} (Axial) [ppm]	0.001	-0.001	0.258
Eu 381.967 {88} (Axial) [ppm]	-0.014	0.000	0.255
Eu 420.505 {80} (Axial) [ppm]	-0.002	-0.002	0.255
Eu 412.970 {82} (Axial) [ppm]	0.002	0.002	0.257
Fe 259.940 {130} (Axial) [ppm]	1.464	0.101	0.259
Fe 238.204 {141} (Axial) [ppm]	1.339	0.088	0.260
Fe 239.562 {141} (Axial) [ppm]	1.398	0.092	0.258
Ga 294.364 {114} (Axial) [ppm]	-0.010	-0.018	0.260
Ga 417.206 {81} (Axial) [ppm]	0.012	0.000	0.261
Ga 287.424 {117} (Axial) [ppm]	0.004	0.021	0.259
Gd 342.247 {98} (Axial) [ppm]	-0.009	-0.007	0.256
Gd 336.223 {100} (Axial) [ppm]	0.111	0.104	0.261
Gd 335.047 {101} (Axial) [ppm]	-0.007	-0.013	0.259
Ge 265.118 {127} (Axial) [ppm]	0.060	0.080	0.260
Ge 219.871 {453} (Axial) [ppm]	-0.344	-0.230	0.259
Ge 206.866 {463} (Axial) [ppm]	-0.259	-0.133	0.258
Hf 339.980 {99} (Axial) [ppm]	0.099	0.062	0.256
Hf 263.871 {128} (Axial) [ppm]	0.271	-0.012	0.256
Hf 232.247 {445} (Axial) [ppm]	0.031	0.010	0.256
Hg 194.227 {474} (Axial) [ppm]	0.016	0.037	0.259
Hg 184.950 {482} (Axial) [ppm]	0.007	0.005	0.258
Hg 253.652 {133} (Axial) [ppm]	-0.133	-0.138	0.255
Ho 345.600 {98} (Axial) [ppm]	0.002	0.000	0.260
Ho 389.102 {87} (Axial) [ppm]	-0.127	-0.052	0.257
In 325.609 {103} (Axial) [ppm]	0.388	0.264	0.256
In 230.606 {446} (Axial) [ppm]	-0.031	0.055	0.261
In 303.936 {111} (Axial) [ppm]	-0.211	-0.028	0.253
Ir 224.268 {450} (Axial) [ppm]	0.026	0.020	0.256
Ir 212.681 {458} (Axial) [ppm]	0.000	-0.005	0.256
Ir 205.222 {464} (Axial) [ppm]	-0.343	-0.409	0.263
K 766.490 {44} (Axial) [ppm]	30.929	0.083	0.249
K 769.896 {44} (Axial) [ppm]	31.066	0.091	0.248
La 412.323 {82} (Axial) [ppm]	0.001	-0.005	0.255
La 333.749 {101} (Axial) [ppm]	0.002	0.001	0.259
La 379.478 {89} (Axial) [ppm]	0.004	0.002	0.257
Li 670.784 {50} (Axial) [ppm]	0.015	0.000	0.248
Li 460.286 {73} (Axial) [ppm]	-0.169	-0.025	0.258
Lu 261.542 {129} (Axial) [ppm]	0.001	0.001	0.258
Lu 291.139 {116} (Axial) [ppm]	-0.156	0.005	0.257
Lu 307.760 {109} (Axial) [ppm]	0.093	0.092	0.259
Mg 279.553 {121} (Axial) [ppm]	0.143	0.133	0.260
Mg 280.270 {120} (Axial) [ppm]	0.143	0.132	0.260
Mg 285.213 {118} (Axial) [ppm]	0.094	0.094	0.257
Mn 257.610 {131} (Axial) [ppm]	2.861	0.004	0.258
Mn 259.373 {130} (Axial) [ppm]	2.840	0.001	0.260
Mn 260.569 {129} (Axial) [ppm]	2.882	0.004	0.260
Mo 202.030 {467} (Axial) [ppm]	1.432	0.011	0.259
Mo 281.615 {120} (Axial) [ppm]	1.292	-0.009	0.260
Mo 204.598 {465} (Axial) [ppm]	1.425	0.006	0.257
Na 589.592 {57} (Axial) [ppm]	22.572	3.805	0.250
Na 818.326 {41} (Axial) [ppm]	21.101	3.358	0.260
Na 330.237 {102} (Axial) [ppm]	0.202	0.037	0.259

5 / 7

24. JULI 2020



HAFB-Muster REF200080093
7/24/2020 12:46:35 PM

QTEGRA

	A_3317759	B_3317773	QC-Test 0.25mg/L
Nb 316.340 {107} (Axial) [ppm]	-0.007	-0.011	0.255
Nd 401.225 {84} (Axial) [ppm]	0.001	0.004	0.258
Nd 430.358 {78} (Axial) [ppm]	-0.007	0.014	0.258
Nd 406.109 {83} (Axial) [ppm]	-0.044	-0.042	0.258
Ni 221.647 {452} (Axial) [ppm]	6.343	0.011	0.262
Ni 231.604 {446} (Axial) [ppm]	6.127	0.046	0.259
Ni 341.476 {99} (Axial) [ppm]	4.658	0.031	0.260
P 177.495 {490} (Axial) [ppm]	0.081	0.099	0.259
P 213.618 {458} (Axial) [ppm]	-0.087	-0.018	0.258
P 214.914 {457} (Axial) [ppm]	-0.075	-0.081	0.258
Pb 220.353 {453} (Axial) [ppm]	-0.034	-0.017	0.260
Pb 216.999 {455} (Axial) [ppm]	0.142	0.161	0.263
Pb 283.306 {119} (Axial) [ppm]	0.560	0.446	0.258
Pd 340.458 {99} (Axial) [ppm]	0.004	-0.020	0.258
Pd 324.270 {104} (Axial) [ppm]	0.001	0.030	0.262
Pr 414.311 {81} (Axial) [ppm]	0.003	0.005	0.259
Pr 417.939 {81} (Axial) [ppm]	0.021	-0.011	0.259
Pt 265.945 {127} (Axial) [ppm]	-0.032	-0.007	0.254
Pt 203.646 {466} (Axial) [ppm]	-0.026	-0.032	0.259
Rb 780.023 {43} (Axial) [ppm]	0.077	0.031	0.248
Rb 420.185 {80} (Axial) [ppm]	1.257	-0.499	0.270
Re 227.525 {448} (Axial) [ppm]	-0.039	0.035	0.258
Re 221.426 {452} (Axial) [ppm]	-0.016	-0.001	0.258
Re 197.312 {471} (Axial) [ppm]	0.031	0.018	0.258
Rh 343.489 {98} (Axial) [ppm]	0.006	0.038	0.256
Rh 369.236 {91} (Axial) [ppm]	-0.092	-0.029	0.259
Ru 240.272 {140} (Axial) [ppm]	-0.013	-0.008	0.259
Ru 266.161 {127} (Axial) [ppm]	0.080	-0.093	0.259
S 180.731 {487} (Axial) [ppm]	10.031	2.019	0.256
S 182.034 {485} (Axial) [ppm]	10.509	2.162	0.250
Sb 217.581 {455} (Axial) [ppm]	-0.205	-0.147	0.257
Sb 206.833 {463} (Axial) [ppm]	0.068	0.000	0.259
Sc 361.384 {93} (Axial) [ppm]	-0.001	0.000	0.256
Sc 363.075 {93} (Axial) [ppm]	-0.006	0.000	0.256
Sc 335.373 {100} (Axial) [ppm]	-0.003	0.000	0.252
Se 196.090 {472} (Axial) [ppm]	-0.108	-0.059	0.259
Se 203.985 {465} (Axial) [ppm]	-0.267	-1.014	0.259
Se 206.279 {463} (Axial) [ppm]	0.064	0.304	0.262
Si 251.611 {134} (Axial) [ppm]	33.887	1.512	0.256
Si 212.412 {459} (Axial) [ppm]	33.081	1.604	0.255
Si 288.158 {117} (Axial) [ppm]	33.295	1.419	0.255
Sm 359.260 {94} (Axial) [ppm]	-0.082	0.035	0.262
Sn 189.989 {477} (Axial) [ppm]	-0.004	-0.012	0.258
Sn 283.999 {119} (Axial) [ppm]	1.673	-0.034	0.258
Sn 242.949 {139} (Axial) [ppm]	-0.336	-0.156	0.254
Sr 407.771 {83} (Axial) [ppm]	0.004	0.003	0.259
Sr 421.552 {80} (Axial) [ppm]	0.003	0.002	0.257
Sr 346.446 {97} (Axial) [ppm]	-0.002	-0.003	0.260
Ta 268.517 {125} (Axial) [ppm]	0.020	0.012	0.258
Ta 240.063 {140} (Axial) [ppm]	0.018	0.019	0.259
Ta 226.230 {449} (Axial) [ppm]	-0.079	-0.042	0.256
Tb 350.917 {96} (Axial) [ppm]	0.000	0.004	0.259
Tb 332.440 {101} (Axial) [ppm]	-0.003	-0.022	0.255
Tb 367.635 {92} (Axial) [ppm]	-0.032	-0.009	0.258
Te 214.281 {457} (Axial) [ppm]	-0.291	-0.166	0.260
Te 238.578 {141} (Axial) [ppm]	1.160	0.619	0.265
Te 225.902 {449} (Axial) [ppm]	-0.196	0.128	0.266
Th 318.019 {106} (Axial) [ppm]	0.024	0.040	0.258

6 / 7

24. JULI 2020

HAFB-Muster REF200080093
7/24/2020 12:46:35 PM

QTEGRA

	A_3317759	B_3317773	QC-Test 0.25mg/L
Th 283.730 {119} (Axial) [ppm]	0.075	0.116	0.258
Th 283.231 {119} (Axial) [ppm]	0.066	0.075	0.253
Ti 323.452 {104} (Axial) [ppm]	0.028	0.015	0.257
Tl 190.856 {477} (Axial) [ppm]	-0.186	-0.130	0.263
Tl 276.787 {122} (Axial) [ppm]	-0.107	-0.222	0.262
Tl 351.924 {96} (Axial) [ppm]	0.701	1.418	0.285
Tm 346.220 {97} (Axial) [ppm]	-0.135	0.002	0.258
Tm 384.802 {88} (Axial) [ppm]	-0.003	-0.004	0.260
U 385.958 {87} (Axial) [ppm]	0.151	0.103	0.257
U 263.553 {128} (Axial) [ppm]	0.003	-0.010	0.258
V 309.311 {109} (Axial) [ppm]	0.077	0.045	0.257
V 292.402 {115} (Axial) [ppm]	0.010	0.020	0.258
W 239.709 {141} (Axial) [ppm]	0.060	0.049	0.257
W 224.875 {450} (Axial) [ppm]	0.119	0.157	0.255
W 207.911 {462} (Axial) [ppm]	0.117	0.072	0.256
Y 437.494 {77} (Axial) [ppm]	0.001	-0.001	0.254
Yb 328.937 {102} (Axial) [ppm]	0.001	0.001	0.257
Yb 369.419 {91} (Axial) [ppm]	0.000	0.000	0.257
Yb 289.138 {117} (Axial) [ppm]	0.001	-0.007	0.259
Zn 213.856 {458} (Axial) [ppm]	4.225	0.027	0.263
Zn 202.548 {466} (Axial) [ppm]	4.502	-0.002	0.258
Zr 339.198 {99} (Axial) [ppm]	0.003	0.002	0.257
Zr 343.823 {98} (Axial) [ppm]	0.008	0.013	0.257
Sm 330.639 {102} (Axial) [ppm]	-0.140	-0.017	0.257
Sm 428.079 {79} (Axial) [ppm]	-0.016	-0.031	0.257

7 / 7

24. JULI 2020

A.42 Reports from acidimetric determination of hydroxylamine and ammonia

tiamo™ titration and more

Lizenz-ID	1798754	Programmversion tiamo 1.1 - 36
Client-Name	mu15ro52804	
Anwender	user	2020-06-09 15:34:11 UTC+2

Resultatreport

Bestimmung

Methode	Hydroxylamin+NH3
Speicherdatum Methode	2020-06-09 15:29:21 UTC+2
Methodenversion	8
Methodenstatus	original
Bestimmungsstart	2020-06-09 15:29:42 UTC+2
Bestimmungsstatus	original
Bestimmungsversion	1
Probennummer	2
Anwender (voller Name)	
Anwender (Kurzname)	FWalthert

Probedaten

ID1	HA 50 % 50 uL
Einmass	56 mg

Endpunkte

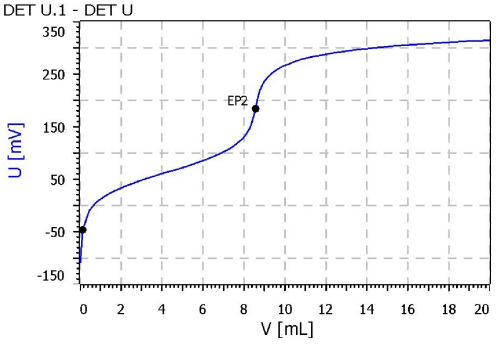
DET U **DET U.1**

EP1	-46.9	mV	0.1300	mL
EP2	184.2	mV	8.5682	mL

Resultate

Ammoniak EP1	**0.395**	**%**
Hydroxylamin E2	**49.770**	**%**

DET U.1 - DET U



Seite 1 von 1

Metrohm

tiamo™ titration and more

Lizenz-ID	1798754	Programmversion	tiamo 1.1 - 36
Client-Name	mu15ro52804		
Anwender	user	2020-06-25	11:09:01 UTC+2

Resultatreport

Bestimmung

Methode	Hydroxylamin+NH3
Speicherdatum Methode	2020-06-09 15:29:21 UTC+2
Methodenversion	8
Methodenstatus	original
Bestimmungsstart	2020-06-25 10:29:46 UTC+2
Bestimmungsstatus	original
Bestimmungsversion	1
Probennummer	2
Anwender (voller Name)	
Anwender (Kurzname)	FWalthert

Probedaten

ID1	Exp9_HA_1	
Einmass	100	µL

Endpunkte

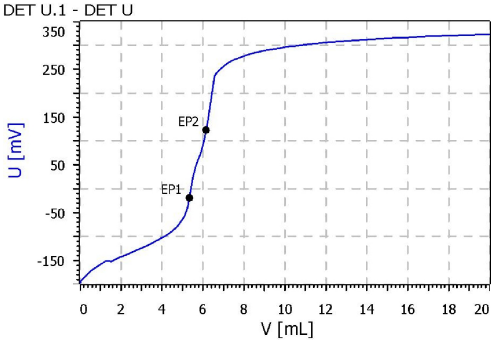
DET U **DET U.1**

EP1	-18.8	mV	5.3439	mL
EP2	123.2	mV	6.1615	mL

Resultate

Ammoniak EP1	**9.101**	**%**
Hydroxylamin E2	**2.700**	**%**

DET U.1 - DET U



Seite 1 von 1

Metrohm

tiamo™ titration and more

Lizenz-ID	1798754	Programmversion	tiamo 1.1 - 36
Client-Name	mu15ro52804		
Anwender	user	2020-06-25	11:08:57 UTC+2

Resultatreport

Bestimmung

Methode	Hydroxylamin+NH3
Speicherdatum Methode	2020-06-25 10:58:22 UTC+2
Methodenversion	9
Methodenstatus	original
Bestimmungsstart	2020-06-25 11:01:29 UTC+2
Bestimmungsstatus	original
Bestimmungsversion	1
Probennummer	4
Anwender (voller Name)	
Anwender (Kurzname)	FWalthert

Probedaten

ID1	Exp9_HA_2
Einmass	100 µL

Endpunkte

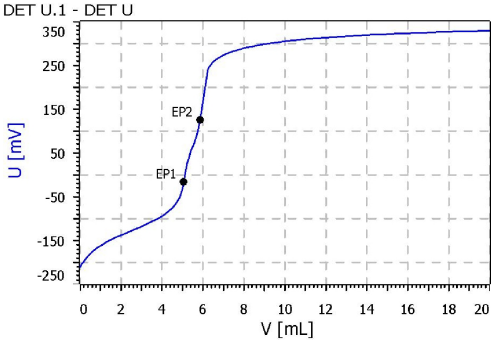
DET U **DET U.1**

EP1	-16.3	mV	5.0694	mL
EP2	125.0	mV	5.8541	mL

Resultate

Ammoniak EP1	**8.634**	**%**
Hydroxylamin E2	**2.592**	**%**

DET U.1 - DET U



Seite 1 von 1

Metrohm

tiamo™ titration and more

Lizenz-ID	1798754	Programmversion	tiamo 1.1 - 36
Client-Name	mu15ro52804		
Anwender	user		2020-07-01 10:43:34 UTC+2

Resultatreport

Bestimmung

Methode	Hydroxylamin+NH3
Speicherdatum Methode	2020-07-01 10:25:00 UTC+2
Methodenversion	10
Methodenstatus	original
Bestimmungsstart	2020-07-01 10:27:08 UTC+2
Bestimmungsstatus	original
Bestimmungsversion	1
Probennummer	2
Anwender (voller Name)	
Anwender (Kurzname)	FWalthert

Probedaten

ID1	Exp10_HA_1
Einmass	100 µL

Endpunkte

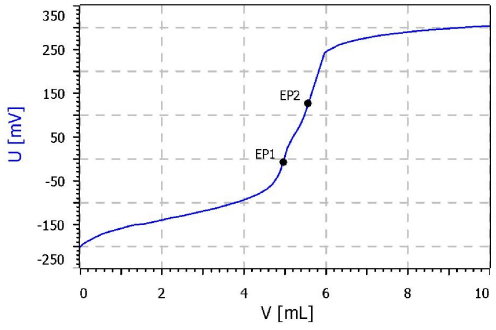
DET U **DET U.1**

EP1	-8.1	mV	4.9604	mL
EP2	126.0	mV	5.5632	mL

Resultate

Ammoniak EP1	**8.448**	**%**
Hydroxylamin E2	**1.991**	**%**

DET U.1 - DET U



Seite 1 von 1

Metrohm

tiamo™ titration and more

Lizenz-ID	1798754	Programmversion tiamo 1.1 - 36
Client-Name	mu15ro52804	
Anwender	user	2020-07-01 10:43:30 UTC+2

Resultatreport

Bestimmung

Methode	Hydroxylamin+NH3
Speicherdatum Methode	2020-07-01 10:25:00 UTC+2
Methodenversion	10
Methodenstatus	original
Bestimmungsstart	2020-07-01 10:33:15 UTC+2
Bestimmungsstatus	original
Bestimmungsversion	1
Probennummer	3
Anwender (voller Name)	
Anwender (Kurzname)	FWalthert

Probedaten

ID1	Exp10_HA_2
Einmass	100 µL

Endpunkte

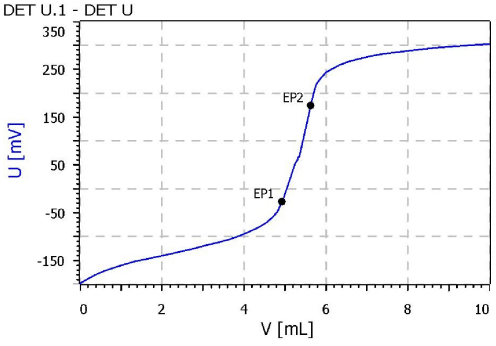
DET U **DET U.1**

EP1	-27.1	mV	4.9252	mL
EP2	173.7	mV	5.6312	mL

Resultate

Ammoniak EP1	**8.388**	**%**
Hydroxylamin E2	**2.332**	**%**

DET U.1 - DET U



Seite 1 von 1

Metrohm

tiamo™ titration and more

Lizenz-ID	1798754	Programmversion	tiamo 1.1 - 36
Client-Name	mu15ro52804		
Anwender	user		2020-07-10 10:00:14 UTC+2

Resultatreport

Bestimmung

Methode	Hydroxylamin+NH3
Speicherdatum Methode	2020-07-01 10:25:00 UTC+2
Methodenversion	10
Methodenstatus	original
Bestimmungsstart	2020-07-10 09:45:31 UTC+2
Bestimmungsstatus	original
Bestimmungsversion	1
Probennummer	1
Anwender (voller Name)	
Anwender (Kurzname)	FWalthert

Probedaten

ID1	Exp12_HA_afterdecomp
Einmass	100 µL

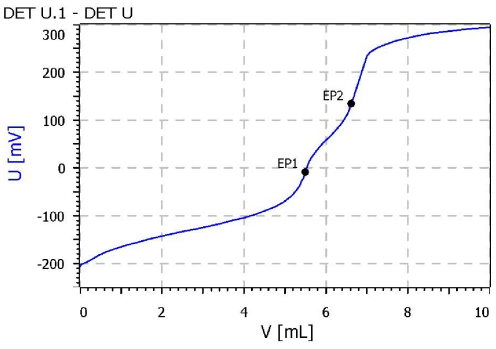
Endpunkte

DET U **DET U.1**

EP1	-8.4	mV	5.5009	mL
EP2	134.1	mV	6.6136	mL

Resultate

Ammoniak EP1	**9.369**	**%**
Hydroxylamin E2	**3.675**	**%**



Seite 1 von 1

Metrohm

tiamo™ titration and more

Lizenz-ID	1798754	Programmversion	tiamo 1.1 - 36
Client-Name	mu15ro52804		
Anwender	user		2020-07-10 10:00:10 UTC+2

Resultatreport

Bestimmung

Methode	Hydroxylamin+NH3
Speicherdatum Methode	2020-07-01 10:25:00 UTC+2
Methodenversion	10
Methodenstatus	original
Bestimmungsstart	2020-07-10 09:52:10 UTC+2
Bestimmungsstatus	original
Bestimmungsversion	1
Probennummer	2
Anwender (voller Name)	
Anwender (Kurzname)	FWalthert

Probedaten

ID1	Exp12_HA_afterdecomp2
Einmass	100 µL

Endpunkte

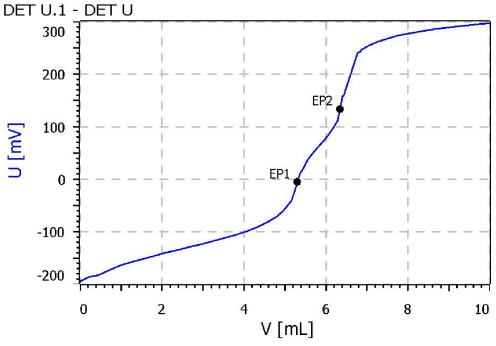
DET U **DET U.1**

EP1	-5.5	mV	5.3036	mL
EP2	133.5	mV	6.3463	mL

Resultate

Ammoniak EP1	**9.033 %**
Hydroxylamin E2	**3.444 %**

DET U.1 - DET U



Seite 1 von 1

Metrohm

tiamo™ titration and more

Lizenz-ID	1798754	Programmversion	tiamo 1.1 - 36
Client-Name	mu15ro52804		
Anwender	user		2020-07-14 17:12:39 UTC+2

Resultatreport

Bestimmung

Methode	Hydroxylamin+NH3
Speicherdatum Methode	2020-07-01 10:25:00 UTC+2
Methodenversion	10
Methodenstatus	original
Bestimmungsstart	2020-07-14 17:07:45 UTC+2
Bestimmungsstatus	original
Bestimmungsversion	1
Probennummer	3
Anwender (voller Name)	
Anwender (Kurzname)	FWalthert

Probedaten

ID1	Exp15_HA_1	
Einmass	100	µL

Endpunkte

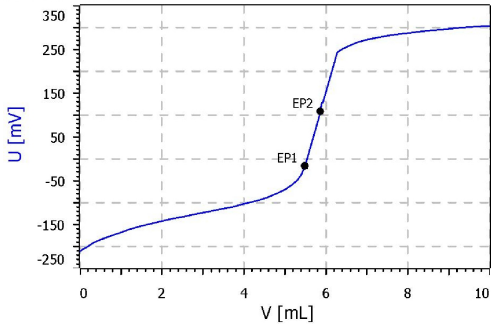
DET U **DET U.1**

EP1	-16.4	mV	5.4891	mL
EP2	108.0	mV	5.8681	mL

Resultate

Ammoniak EP1	**9.349**	**%**
Hydroxylamin E2	**1.252**	**%**

DET U.1 - DET U



Seite 1 von 1

Metrohm

tiamo™ titration and more

Lizenz-ID	1798754	Programmversion	tiamo 1.1 - 36
Client-Name	mu15ro52804		
Anwender	user		2020-07-14 17:12:43 UTC+2

Resultatreport

Bestimmung

Methode	Hydroxylamin+NH3
Speicherdatum Methode	2020-07-01 10:25:00 UTC+2
Methodenversion	10
Methodenstatus	original
Bestimmungsstart	2020-07-14 17:02:15 UTC+2
Bestimmungsstatus	original
Bestimmungsversion	1
Probennummer	2
Anwender (voller Name)	
Anwender (Kurzname)	FWalthert

Probedaten

ID1	Exp15_HA_2
Einmass	100 µL

Endpunkte

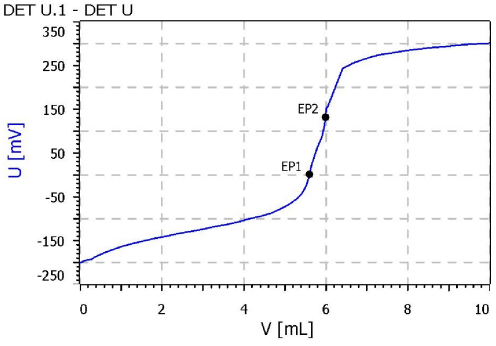
DET U **DET U.1**

EP1	0.9	mV	5.6028	mL
EP2	131.7	mV	5.9967	mL

Resultate

Ammoniak EP1	**9.542**	**%**
Hydroxylamin E2	**1.301**	**%**

DET U.1 - DET U



Seite 1 von 1

Metrohm

tiamo™ titration and more

Lizenz-ID	1798754	Programmversion	tiamo 1.1 - 36
Client-Name	mu15ro52804		
Anwender	user		2020-07-22 12:46:58 UTC+2

Resultatreport

Bestimmung

Methode	Hydroxylamin+NH3
Speicherdatum Methode	2020-07-01 10:25:00 UTC+2
Methodenversion	10
Methodenstatus	original
Bestimmungsstart	2020-07-22 12:32:58 UTC+2
Bestimmungsstatus	original
Bestimmungsversion	1
Probennummer	1
Anwender (voller Name)	
Anwender (Kurzname)	FWalthert

Probedaten

ID1	Exp16_HA_1	
Einmass	100	µL

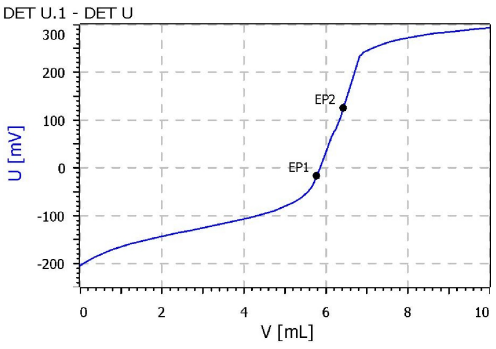
Endpunkte

DET U **DET U.1**

EP1	-16.2	mV	5.7774	mL
EP2	125.7	mV	6.4183	mL

Resultate

Ammoniak EP1	**9.839**	**%**
Hydroxylamin E2	**2.117**	**%**



Seite 1 von 1

Metrohm

tiamo™ titration and more

Lizenz-ID	1798754	Programmversion	tiamo 1.1 - 36
Client-Name	mu15ro52804		
Anwender	user		2020-07-22 12:46:54 UTC+2

Resultatreport

Bestimmung

Methode	Hydroxylamin+NH3
Speicherdatum Methode	2020-07-01 10:25:00 UTC+2
Methodenversion	10
Methodenstatus	original
Bestimmungsstart	2020-07-22 12:38:36 UTC+2
Bestimmungsstatus	original
Bestimmungsversion	1
Probennummer	2
Anwender (voller Name)	
Anwender (Kurzname)	FWalthert

Probedaten

ID1	Exp16_HA_2
Einmass	100 µL

Endpunkte

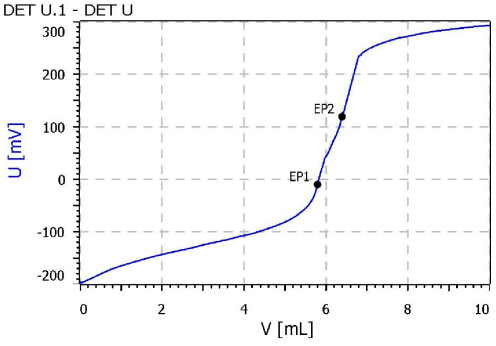
DET U **DET U.1**

EP1	-10.3	mV	5.7938	mL
EP2	118.5	mV	6.3934	mL

Resultate

Ammoniak EP1	**9.867 %**
Hydroxylamin E2	**1.980 %**

DET U.1 - DET U



Seite 1 von 1

Metrohm

tiamo™ titration and more

Lizenz-ID	1798754	Programmversion	tiamo 1.1 - 36
Client-Name	mu15ro52804		
Anwender	user		2020-07-28 19:26:48 UTC+2

Resultatreport

Bestimmung

Methode	Hydroxylamin+NH3
Speicherdatum Methode	2020-07-01 10:25:00 UTC+2
Methodenversion	10
Methodenstatus	original
Bestimmungsstart	2020-07-28 19:13:26 UTC+2
Bestimmungsstatus	original
Bestimmungsversion	1
Probennummer	1
Anwender (voller Name)	
Anwender (Kurzname)	FWalthert

Probedaten

ID1	Exp19_HA_1
Einmass	100 µL

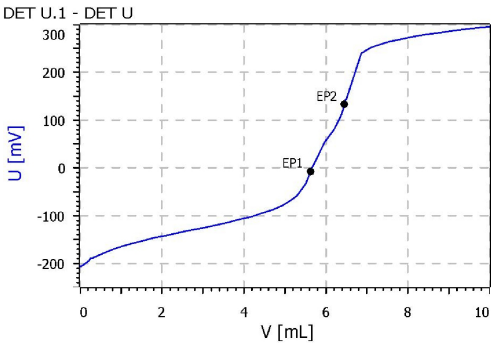
Endpunkte

DET U **DET U.1**

EP1	-7.4	mV	5.6242	mL
EP2	133.2	mV	6.4537	mL

Resultate

Ammoniak EP1	**9.579**	**%**
Hydroxylamin E2	**2.740**	**%**



Seite 1 von 1

Metrohm

tiamo™ titration and more

Lizenz-ID	1798754	Programmversion	tiamo 1.1 - 36
Client-Name	mu15ro52804		
Anwender	user		2020-07-28 19:26:44 UTC+2

Resultatreport

Bestimmung

Methode	Hydroxylamin+NH3
Speicherdatum Methode	2020-07-01 10:25:00 UTC+2
Methodenversion	10
Methodenstatus	original
Bestimmungsstart	2020-07-28 19:20:18 UTC+2
Bestimmungsstatus	original
Bestimmungsversion	1
Probennummer	2
Anwender (voller Name)	
Anwender (Kurzname)	FWalthert

Probedaten

ID1	Exp19_HA_2	
Einmass	100	µL

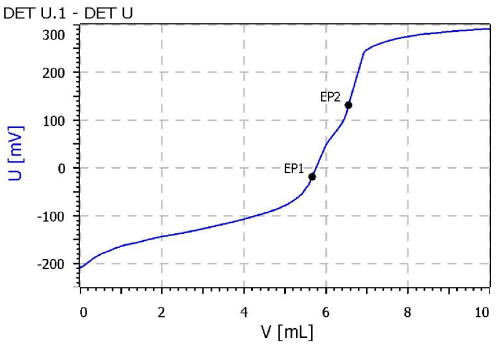
Endpunkte

DET U **DET U.1**

EP1	-18.4	mV	5.6618	mL
EP2	131.1	mV	6.5520	mL

Resultate

Ammoniak EP1	**9.643**	**%**
Hydroxylamin E2	**2.940**	**%**



Seite 1 von 1

Metrohm

tiamo™ titration and more

Lizenz-ID	1798754	Programmversion	tiamo 1.1 - 36
Client-Name	mu15ro52804		
Anwender	user		2020-08-03 18:33:56 UTC+2

Resultatreport

Bestimmung

Methode	Hydroxylamin+NH3
Speicherdatum Methode	2020-07-01 10:25:00 UTC+2
Methodenversion	10
Methodenstatus	original
Bestimmungsstart	2020-08-03 18:17:12 UTC+2
Bestimmungsstatus	original
Bestimmungsversion	1
Probennummer	1
Anwender (voller Name)	
Anwender (Kurzname)	FWalthert

Probedaten

ID1	Exp20_HA
Einmass	100 µL

Endpunkte

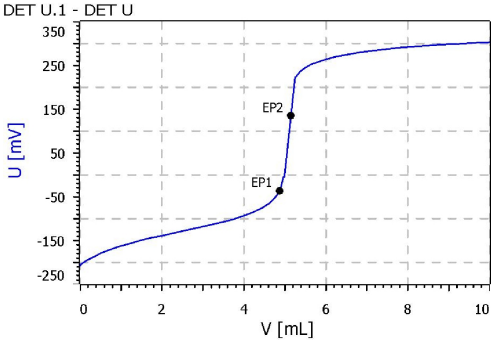
DET U **DET U.1**

EP1	-36.5	mV	4.8734	mL
EP2	134.7	mV	5.1454	mL

Resultate

Ammoniak EP1	**8.300**	**%**
Hydroxylamin E2	**0.898**	**%**

DET U.1 - DET U



Metrohm

tiamo™ titration and more

Lizenz-ID	1798754	Programmversion tiamo 1.1 - 36
Client-Name	mu15ro52804	
Anwender	user	2020-08-03 18:33:51 UTC+2

Resultatreport

Bestimmung

Methode	Hydroxylamin+NH3
Speicherdatum Methode	2020-07-01 10:25:00 UTC+2
Methodenversion	10
Methodenstatus	original
Bestimmungsstart	2020-08-03 18:28:24 UTC+2
Bestimmungsstatus	original
Bestimmungsversion	1
Probennummer	3
Anwender (voller Name)	
Anwender (Kurzname)	FWalthert

Probedaten

ID1	Exp20_HA_1
Einmass	100 µL

Endpunkte

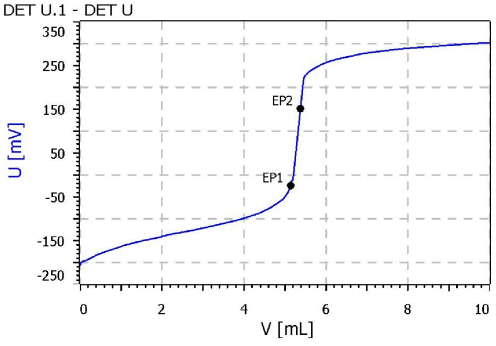
DET U **DET U.1**

EP1	-24.6	mV	5.1437	mL
EP2	150.6	mV	5.3763	mL

Resultate

Ammoniak EP1	**8.760**	**%**
Hydroxylamin E2	**0.768**	**%**

DET U.1 - DET U



Seite 1 von 1

Metrohm

A.43 Risk analysis for EXR experiments

Risikoanalyse für die Hydroxylamin-Zersetzung im EXR

Projekt:

Thermokinetic model of hydroxylamine decomposition in the catalytic oxidation of ammonia with titanium silicalite-1

Bachelor Thesis

Auftraggeber: Dr. F. Meemken (LONZA)

Betreuer: Prof. Dr. A. Zogg

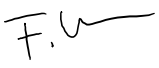
Fachhochschule Nordwestschweiz, Standort Muttenz

Hochschule für Life Sciences

MLS17

23.06.2020

Walthert Florian

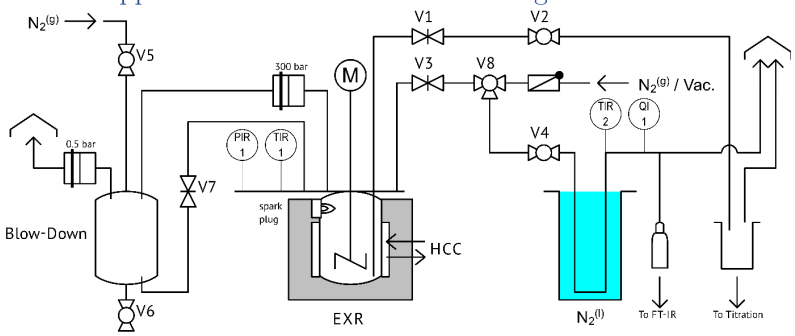


1

Inhalt

	Name	Datum	Unterschrift
Verfahrensgeber (z.B. Labormitarbeiter oder Student)			
Projektleiter (z.B. Vorgesetzter oder Modulverantwortlicher)			
Sicherheitsbeauftragter			

1. Laborapparatur und Versuchsbeschreibung



Der Mantel des *Premex* Explosionsreaktors (EXR) MED 1789 besteht aus Hastelloy C-22 und es kann wahlweise ein Borsilikat- oder PEEK-Einsatz verwendet werden. Das Tauchrohr sowie der Temperaturfühler bestehen aus Stahl. Es kann ein Rührer aus Stahl oder PEEK eingesetzt werden. Der maximal zulässige Druck ist 300 bar. Eine Berstscheibe öffnet zwischen 270 und 330 bar den Reaktor. Die Berstscheibe des Blowdown-Tanks öffnet bei 0.5 bar Überdruck.

Der gewünschte Reaktoreinsatz wird im EXR positioniert. Der gewünschte Rührer wird angebracht. Der Deckel wird mit 90 Nm angeschraubt. Der Blowdown-Tank wird vor dem Experiment mit Stickstoff inertisiert (durch Öffnen der Ventile 5 und 6). Anschliessend wird die 50 %-ige Hydroxylamin-Lösung durch das Tauchrohr mit Vakuum in den EXR eingesogen. Der EXR wird verschlossen durch das Umstellen von Ventil 8 und Schliessen der Ventile 1 bis 3 und 7. Das Ventil 4 muss offen bleiben. Der Reaktor wurde entweder bereits vorgeheizt oder wird nun für die Zersetzungsreaktion aufgeheizt. Durch die Zersetzung wird ein Druckanstieg erwartet. Die flüssige Phase aus dem EXR kann nach dem Zersetzungsexperiment zwecks Analyse durch Öffnen der Ventile V1 und V2 via ein Tauchrohr vorsichtig in ein Auffanggefäss aus Glas abgelassen werden. In diesem Auffanggefäss ist Wasser oder Eis zur Kühlung der Flüssigkeit vorgelegt. Das Auffanggefäss wird nur in der Kapelle geöffnet und wird zusätzlich mit einem Stahlrohr ummantelt und steht in einem Dewar. Die Zersetzungsgase werden entweder durch Kryokondensation oder direkt in der IR-Gaszelle gesammelt. Beim Einsatz der Kühlfalle wird zuerst der Stickstoff diese passieren ohne zu kondensieren. Die kondensierten Gase sammeln sich im Rohrbogen in der Kühlfalle an. Wenn im EXR der Umgebungsdruck erreicht ist, wird das Ventil 4 geschlossen. Der Rohrbogen wird langsam aus dem flüssigen Stickstoff gehoben, um die Gase sequentiell wieder zu verdampfen. Dabei werden jeweils die Gastemperatur und der akkumulierte Durchfluss gemessen.

2. Sichere Prozessbedingungen

Parameter	Min	Max
Reaktortemperatur	2 °C	110 °C
Reaktionsdruck	Normaldruck	270 bar
Füllvolumen	0 mL	80 mL
Versuchszeit	frei	frei

3. Gefahrensuche Produktespezifisch

Reagens	**CAS**	**1.1 Gesundheits-Gefahren**		**1.2 Physikalische Gefahren**							**Gefahren Nummer**
		Akut	Chronisch	Mp [°C]	Bp [°C]	Lager-Temperatur	Brandfördernd oder Pyrophor	Explosions-grenzen	Inkompatible Chemikalien	Katalytische Zersetzung	
NH_3	7664-41-7	Kat.2 **1**	14 mg/m³	-77.7	-33	RT, gut belüftet	nein	15 bis 28% **2**	Säuren, Luft und Oxidationsmittel	-	**1, 2**
NO	10102-43-9	Kat.1* **1**	6.3 mg/m³	-164	-152	RT	Brandfördernd	-	-	-	**1**
N_2O	10024-97-2	-	180 mg/m³	−90,8	−88,5	RT	Brandfördernd	-	-		
				−88,5							
H_2	1333-74-0	-	-	-259,2	-253	RT	Nein	4-77%	Luft, Oxidationsmittel		
Hydroxylamin, 50% in H2O	7803-49-8	Kat. 4	0.003 mg/m³ CMR **3**	33	58	2-8 °C, Frost vermeiden **4**	nein	unbekannt	Starke Red.mittel, Säuren, Basen, Metalle	-	**3, 4**

Quellen: [1]–[10]

* Reiz- und lungenschädigende Wirkung sind vorwiegend verunreinigungsbedingt (NO2)

5

4. Gefahrensuche Prozessspezifisch

4.1. Mögliche Reaktionsgleichungen

Die thermische Zersetzung von Hydroxylamin ist nicht genau bekannt. In der Literatur werden einige Reaktionsmechanismen vorgeschlagen. Die bedeutendsten drei Reaktionsgleichung werden in diesem Abschnitt behandelt. Die erste betrachtete Reaktionsgleichung wurde aufgrund analytischer Messungen der Zersetzungsgase formuliert. Die zweiten und die dritte Reaktion beruhen auf vermuteten Reaktionsmechanismen.

Reaktion 1:

$$157\,NH_2OH^{(l)} \longrightarrow 61\,NH_3{}^{(g)} + 35\,N_2{}^{(g)} + 12\,N_2O^{(g)} + 2\,NO^{(g)} + 143\,H_2O^{(g)} + H_2{}^{(g)}$$

Reaktion 2 (basisch):

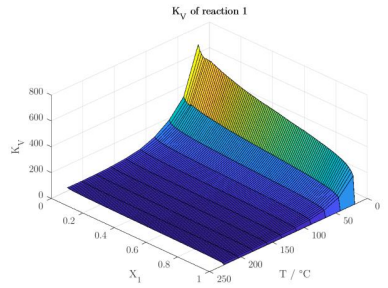
$$3\,NH_2OH \longrightarrow N_2 + NH_3 + 3\,H_2O$$

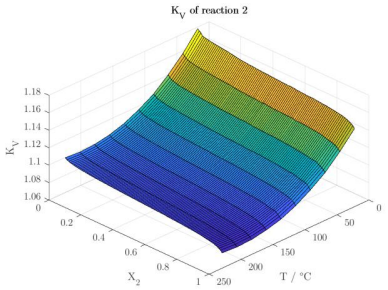
Reaktion 3 (sauer):

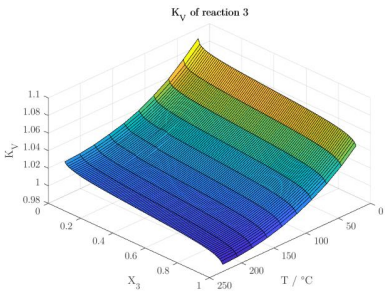
$$4\,NH_2OH \longrightarrow N_2O + 2\,NH_3 + 3\,H_2O$$

4.2. Gleichgewicht der Zersetzungsreaktionen

Situation:	Die Zersetzungsreaktionen laufen teilweise ab.
Gefahr:	Der Druck im EXR könnte über 300 bar ansteigen.
Diskussion:	Da bei den drei vorgeschlagenen Reaktionsgleichungen aus einer wässrigen Lösung Gase entstehen, wird der Druck durch die Zersetzung ansteigen. Das Gleichgewicht dieser Reaktionen entscheidet welche Gasmenge freigesetzt wird. Die Gleichgewichts-Konstante bei konstantem Volumen wurde als Funktion der Temperatur und des Reaktionsumsatzes über die Helmholtz-Energie für jede Reaktionsgleichung berechnet. Ein Wert über 1 bedeutet, dass die Reaktion mehr Gase produziert (Produktseite) lauft und ein Wert von 0 bis 1 bedeutet, dass sie hin zum Hydroxylamin strebt.







Die Reaktionen 1 und 2 erreichen das Gleichgewicht erst bei einem Umsatz von 1-10^{12} bis 1. Die dritte Reaktion hat einen Bereich mit einem $K_V < 1$ bei tieferem Umsatz, aber bei einer Temperatur von über 130 °C. Daher kann man vereinfacht sagen, dass das Reaktionsgleichgewicht bei den Versuchsbedingungen immer bei vollem Reaktionsumsatz liegt. Für die folgenden Berechnungen wurde immer von der kompletten Umsetzung ausgegangen.

4.3. Isotherme Zersetzung

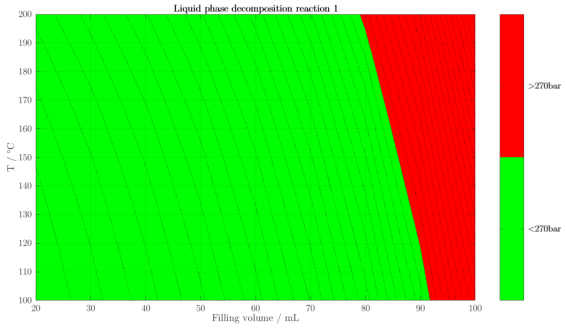
Situation: Die Zersetzung vom Hydroxylamin in der Gasphase oder in der Flüssigphase findet komplett statt. Die Temperatur kann abgeführt werden.

Gefahr: Der Druck im EXR könnte über 300 bar ansteigen.

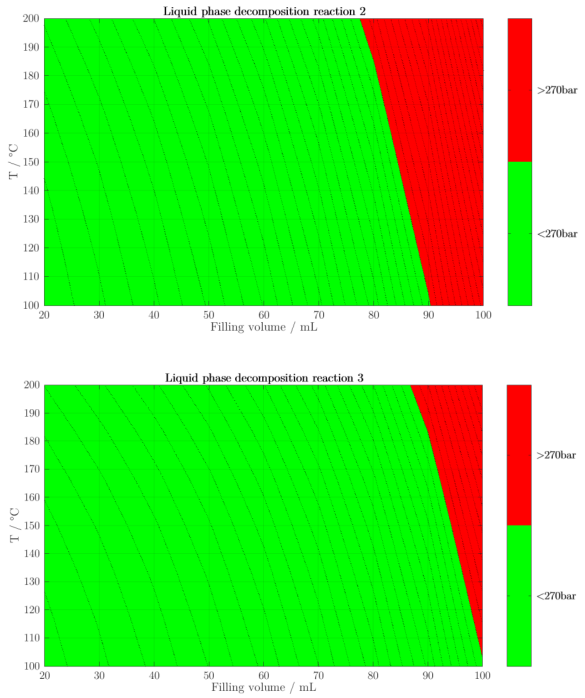
Diskussion: Der isotherme Druckanstieg wurde für die drei Reaktionen berechnet. Dabei wurde einerseits von der Zersetzung der Gasphase, sowie der Zersetzung der Flüssigphase (komplette Reaktionsmasse) ausgegangen.

Die Berechnungen für die Gasphasenzersetzungen ergaben bei allen Drücken bei 100 mL Füllmengen und 250 °C Reaktortemperatur einen maximalen Druck von 75 bar. Der EXR könnte daher einer Gasphasen-Detonation standhalten.

Anders sieht es aus bei den resultierenden Drücken im Falle einer Zersetzung der kompletten flüssigen Phase. Bei einem Füllvolumen von 100 mL und einer Reaktortemperatur von 250°C entstehen für die Reaktionen Drücke von 546, 563 und 458 bar. Daher wurde der entstehende Druck durch isotherme Zersetzungen bei verschiedenen Füllvolumen und Reaktionstemperaturen ausgerechnet und in den folgenden drei Kontour-Plots des Druckes dargestellt. Als Grenzwert für sichere Bedingungen wurden 90% des maximalen Betriebsdruckes gewählt. Die rote Fläche stellt den Bereich dar, in dem der Grenzwert überschritten wird und daher unzulässig ist.



8



Die Reaktion 2 hat den grössten unzulässigen Bereich. Aufgrund des Diagramms werden als sichere Extrembedingungen ein Füllvolumen von 80mL und einer Reaktortemperatur von 180 °C festgelegt.

9

4.4. Adiabate Zersetzung

Situation: Die Zersetzung vom Hydroxylamin in der Gasphase oder in der Flüssigphase findet komplett statt. Das System ist isoliert.

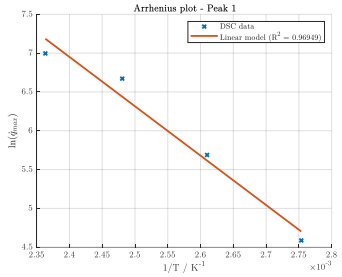
Gefahr: Der Druck im EXR könnte über 300 bar ansteigen.

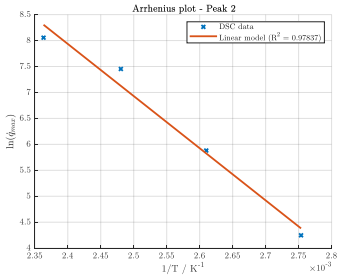
Diskussion: Ausgehend von den maximal zulässigen Bedingungen für den Fall einer isothermen Zersetzung (80 mL Füllvolumen und 180 °C Reaktortemperatur) wurde für die drei Reaktionen jeweils die adiabate Zersetzung der Gasphase und der Flüssigphase (komplette Reaktionsmasse) berechnet:

Reaktion 1:	Gasphase	T = 1496 °C, p = 63.6 bar
	Flüssigphase	T = 1384 °C, p = 933 bar
Reaktion 2:	Gasphase	T = 1675 °C, p = 71.1 bar
	Flüssigphase	T = 1569 °C, p = 1076 bar
Reaktion 3:	Gasphase	T = 1109 °C, p = 47.7 bar
	Flüssigphase	T = 1013 °C, p = 594 bar

Auch im adiabaten Fall sollten die Gasphasenzersetzungen also keine Gefahr darstellen. Die Flüssigphasenzersetzungen entwickeln jedoch Drücke die im EXR nicht mehr gehalten werden können. Es muss daher eine genügende Kühlung sichergestellt werden.**5**

Die maximale Reaktionsleistung wurde aus den isothermen DSC-Messungen bei 90, 110, 130 und 150 °C in Stahltiegeln gelesen. Folgende Arrhenius-Plots wurden für die beiden auftretenden Peaks erstellt:



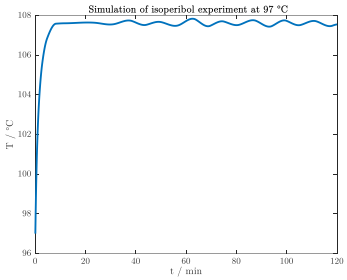


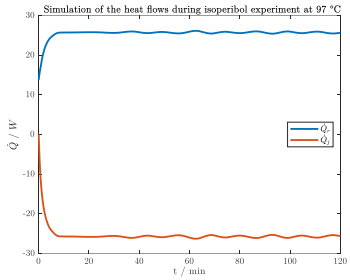
Die lineare Regression wurde verwendet, um die maximale Reaktionsleistung als Funktion der Temperatur zu berechnen.

Die Kühlung erfolgt durch den Mantel des EXR mit dem Thermostat. Der thermische Widerstand des Mantels wird als limitierender Faktor

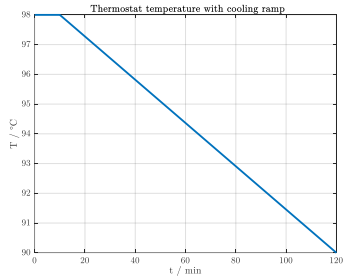
für die Kühlleistung betrachtet. Der Wärmedurchgangskoeffizient (k-Wert) kann berechnet werden durch die Wärmeleitfähigkeit der beiden Wände (Borsilikat und HC22) und dem Wärmeüberganskoeffizient von Wasser. Der k-Wert des EXR beträgt $279\,W/(m^2K)$.

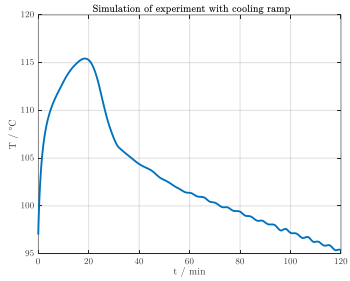
Unter der Annahme einer konstanten maximalen Reaktionsleistung (Reaktion nullter Ordnung) bei einem Füllvolumen von 80 mL ist die Reaktortemperatur (isoperibol) von 97 °C die maximale Temperatur, die nicht zu einer Runaway-Situation führt (bei 98°C nach 16 min).



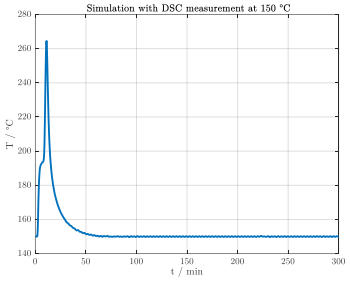


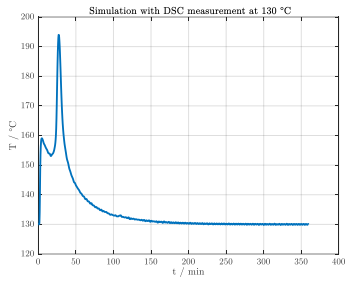
Wenn allerdings der Temperaturanstieg rechtzeitig erkannt wird, könnte der Sollwert des Thermostaten gesenkt werden. Bei einer Reaktortemperatur von 98 °C erfolgt der Runaway nach 16 min. Wenn aber nach 10 min auf 90 °C gekühlt wird kann die Reaktion unter Kontrolle gebracht werden.

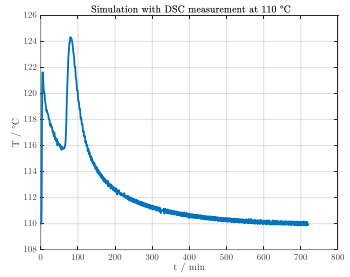


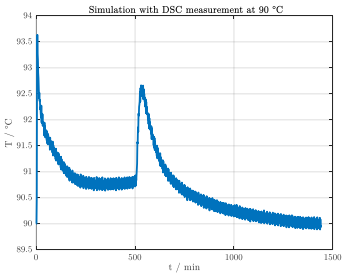


Da eine konstante maximale Reaktionsleistung jedoch unrealistisch ist, aber ein geeignetes kinetisches Modell noch fehlt, wurden die tatsächlichen Reaktionsleistungen aus den isothermen DSC-Messungen verwendet. Keine der Temperaturen (90-150 °C) ergab eine Runaway-Situation. Die Temperaturen stiegen jedoch kurzeitig extrem stark an.









Wenn man die maximale Reaktortemperatur für die isotherme Zersetzungen von 180 °C annimmt, steigt der Druck der Simulationen bei 130 und 150 °C zu stark an. Die maximale Reaktortemperatur, die also gefahren werden darf, ist 110 °C, bei der ein maximaler Druckanstieg auf 135 bar nach 100 min zu erwarten ist.**5**

Alle vorangehenden Berechnungen sind eher konservativ und die DSC-Messungen wurden in Stahltiegeln gemacht. Stahl scheint die Zersetzung zu katalysieren. Daher ist zu erwarten, dass die realen Drücke und Temperaturen tiefer sind. Die Experimente werden so geplant, dass zuerst kleine Volumina und tiefe Temperaturen gewählt werden. Wenn sich die Bedingungen als unkritisch herausstellen, werden die Bedingungen angepasst. Durch ein vorsichtiges Herantasten an das reale Verhalten, kann eine Runaway-Situation vermieden werden.

4.5. Zersetzung ausserhalb des EXR

Situation: Hydroxylamin zersetzt sich in einer der Leitungen zu den Auffanggefässen

Gefahr: Die Leitungen könnten durch den Druck bersten.

Diskussion: Alle Leitungen sind entweder Swagelok-Rohre (316L) oder Teflon-Schläuche. Die Zersetzung der Gasphase wurde im Abschnitt 4.4 bereits besprochen. Im Fall einer gesättigten Gasphase kann höchstens ein Druck von ca. 70 bar entstehen. Der Teflonschlauch, der die flüssige Phase ableitet ist hin zum Auffanggefäss immer offen und würde bei einem Überdruck nur aufreissen.**6**

Die Zersetzung im Auffanggefäss aus Glas ist möglich. Die Lösung kühlt jedoch rasch auf Raumtemperatur ab, was die autokatalytische Reaktion verzögern sollte.**7**

4.6. Zersetzung der Lösung ausserhalb der Apparatur

Situation: Die Probelösung ist noch nicht vollständig zersetzt, aber durch die Messung thermisch gealtert. Sie wird nach dem Versuch durch eine acidimetrische Titration analysiert.

Gefahr: Die autokatalytische Zersetzung startet nach der Messung.

Diskussion: Auch die abgekühlte Lösung hat das Potenzial zu einer spontanen Zersetzung. Es ist jedoch davon auszugehen, dass der HA-Gehalt wesentlich tiefer ist als in der gekauften Lösung. Allerdings ist die Probe durch die Versuche im EXR bereits thermisch gealtert und könnte dadurch bereits die TMRad erreichen. Es ist fraglich, ob die Reaktionsleistung bei RT zu einer Runaway-Situation führen würde.**8**

5. Gefahr-Massnahmenliste Produktspezifisch

Gefahrennr.	Szenario (Gefahr, Ursache, Auswirkung)	Risiko		Massnahmen	Risiko	
		W	T		W	T
1	NH_3 und NO sind akut stark toxisch und ätzend auf der Haut, in den Atemwegen und in den Augen. Ammoniak und NO können während dem Ablassen der flüssigen bzw. gasförmigen Phase aus dem Reaktor austreten.	C	III	IST: • Dichtigkeitstest der Apparatur vor dem Versuch. Toxische Gase treten nur an den dafür vorgesehen Stellen aus. • Inhalativ: Die Abluft aus dem Auffanggefäss und der IR-Gaszelle wird direkt in die Quellenabluft geleitet. • Schutz von Haut und Augen: Die persönliche Sicherheitsausrüstung wird getragen.	C	IV
2	Innerhalb der Apparatur entsteht ein explosives Luft – Ammoniak Gemisch. Ausgehend von Normaldruck wäre ein maximaler Explosionsdruck von ca. 10 bar zu erwarten. Im EXR könnte ein grundsätzlich höherer Ausgangsdruck entstehen wenn z.B. Druckluft in den EXR geleitet wird.	C	III	IST: • An die Apparatur ist keine Druckluft direkt angeschlossen. • Der Blow-Down-Tank und der EXR sind für Explosionen mit einem Ausgangsdruck von 1 bar als Druckstossfest zu betrachten (Auslegungsdruck: 6 bzw. 300 bar). • Der Restliche Versuchsaufbau ist mit Ausnahme des Glasauffanggefässes für Explosionen mit einem Ausgangsdruck von 1 bar als Druckstossfest zu betrachten SOLL: • Die Apparatur (inkl. Blow-Down-Tank) wird vor jeder Messung mit Stickstoff inertisiert, damit kein explosives Gemisch mit Luft als Oxidationsmittel entstehen kann	E	IV
3	Hydroxylamin ist chronisch inhalativ toxisch. HA-Dämpfe können während dem Ablassen der flüssigen bzw. gasförmigen Phase aus dem Reaktor austreten. Aufgrund des tiefen Dampfdruckes ist eine Gesundheitsgefährdend durch Dämpfe erst bei erhöhten Temperauren zu erwarten.	D	III	IST: • Dichtigkeitstest der Apparatur vor dem Versuch. Toxische Gase treten nur an den dafür vorgesehen Stellen aus. • Inhalativ: Die Abluft aus dem Auffanggefäss und der IR-Gaszelle wird direkt in die Quellenabluft geleitet. • Im Rahmen der Bachelorarbeit werden nur wenig Versuche innerhalb einer Zeitspanne von 3 Monaten durchgeführt.	E	IV

Gefahrennr.	Szenario (Gefahr, Ursache, Auswirkung)	Risiko		Massnahmen	Risiko	
		W	T		W	T
4	Infolge einer Fehlhandhabung gelangt noch nicht zersetzte Hydroxylamin-Lösung in die Kühlfalle. Hydroxylamin kristallisiert und zersetzt explosionsartig. Die Apparatur birst.	D	II	IST: • Sowohl das Rohr der Kühlfalle als auch der Dewar sind aus Stahl gebaut. • Der Reaktor selber kann nicht unter 2 °C gekühlt werden (Thermostat kann nicht Kühlen und ist an die 2 ... 8 °C Kühlschiene angeschlossen). SOLL: • Der EXR darf nur über die Kühlfalle geöffnet werden sofern der Reaktordruck ± 30 % dem vorher berechneten Zersetzungsdruck entspricht. In diesem Fall kann davon ausgegangen werden, dass keine signifikante Mengen an zersetzbarem Hydroxylamin mehr vorhanden sind.	E	II

15

6. Gefahr-Massnahmeliste Prozessspezifisch

Gefahrennr.	Szenario (Gefahr, Ursache, Auswirkung)	Risiko		Massnahmen	Risiko	
		W	T		W	T
5	Während einem Zersetzungsversuch fällt der Kühlkreislauf des EXR aus. Dadurch kann sich der EXR aufgrund der restlichen Zersetzungswärme adiabat aufheizen. Der Druck im Reaktor steigt über 300 bar. Die Berstscheibe öffnet. Die Zersetzungsgeschwindigkeit ist bei 300 bar jedoch so hoch, dass der Druck nicht mehr entlastet werden kann und die Temperatur steigt weiter an. Bersten des EXR.	C	II	IST: • Ein vorher inertisierter Blow-Down-Tank ist angeschlossen. Es kann jedoch nicht davon ausgegangen werden, dass die Dimensionierung der Berstscheibe des EXR für dieses Szenario ausreichend ist. • Der Blow-Down-Tank ist wiederum mit einer Berstscheibe abgesichert. Der Entlastungsquerschnitt ist ausreichend. SOLL: • Mindestens jede halbe Stunde muss die Innentemperatur des Reaktors vor Ort überprüft werden. Das Experiment dar nur fortgesetzt werden, wenn die Abweichung der Innentemperatur < 30 % bezüglich des vorher berechneten Wertes liegt. • Gleichzeitig muss die Funktionsfähigkeit des Kühlkreislaufes überprüft werden. • Wird festgestellt, dass der Kühlkreislauf ausgefallen ist oder die Reaktortemperatur zu hohe Werte annimmt, muss der Vorgesetzte informiert werden und gleichzeitig der Reaktor via V7 langsam manuell in den Blow-Down-Tank entlastet werden. Die Reaktortemperatur mittels Siedekühlung gsenkt werden. V6 des Blow-Down-Tanks bleibt während diesem manuellen Entlasten geschlossen. Das Experiment darf nur zusammen mit dem Vorgesetzen weitergeführt werden.	E	II
5.1	Fehlhandhabung Thermostat: Beim Thermostat des EXR wird eine zu hohe Temperatur eingestellt. Dadurch ereignet sich ein Runaway analog zu 5.	C	II	IST: • Beim Thermostat wird die Maximaltemperatur von 110 °C eingestellt.	E	II

16

Gefahrennr.	Szenario (Gefahr, Ursache, Auswirkung)	Risiko		Massnahmen	Risiko	
		W	T		W	T
6	Eine Zersetzung von noch in der flüssigen Phase vorhandenem Hydroxylamin ereignet sich in einer Leitung EXR -> Auffanggefäss bzw. EXR -> IR-Gazelle. Aufgrund der Zersetzung des Hydroxylamins steigt der Druck steigt über 10 bar und der Teflon-Schlauch reisst auf. Austritt von toxischen Gasen.	C	III	IST: • Gegen Spritzer: Die persönliche Sicherheitsausrüstung wird getragen SOLL: • Der EXR darf nur geöffnet werden sofern der Reaktordruck ± 30 % dem vorher berechneten Zersetzungsdruck entspricht. In diesem Fall kann davon ausgegangen werden, dass keine signifikante Mengen an zersetzbarem Hydroxylamin mehr vorhanden sind.	E	III
7	Eine Zersetzung von noch in der flüssigen Phase vorhandenem Hydroxylamin ereignet sich im Auffanggefäss. Das Auffanggefäss explodiert.	C	II	IST: • Das Auffanggefäss wird mit einem Stahlrohr ummantelt. Ein zusätzliches Kühlgefäss (Dewar) welcher das Auffanggefäss umgibt, ist ebenfalls aus Stahl gefertigt. SOLL: • Der EXR darf nur geöffnet werden sofern der Reaktordruck ± 30 % dem vorher berechneten Zersetzungsdruck entspricht. In diesem Fall kann davon ausgegangen werden, dass keine signifikante Mengen an zersetzbarem Hydroxylamin mehr vorhanden sind.	D	III
8	Die Lösung aus dem EXR oder aus dem Auffanggefäss zersetzt sich ausserhalb der Laborapparatur nach der Messung. Giftige Gase gelangen ins Labor.	D	II	SOLL: • Die flüssige Phase im Auffanggefäss wird kühl gelagert und nicht verschlossen. Die Lösung wird, so schnell wie möglich analysiert. Für Analyse-Proben werden Kunststoff-Probeflaschen verwendet. • Flüssige Abfälle werden so schnell wie möglich mit viel Wasser vorverdünnt (1:10) und anschliessend bei laufendem Wasserhahn kanalisiert (Achtung: NICHT über die PTC-Neutralisations-Anlage).	E	II

Gefahrennr.	Szenario (Gefahr, Ursache, Auswirkung)	Risiko		Massnahmen	Risiko	
		W	T		W	T
9	Havarie: • Beim Befüllen oder Entleeren (durch Tauchrohr) des Reaktors wird Hydroxylamin-Lösung oder zersetzte Reaktionslösung verschüttet. • Die Reaktionslösung wird zu schnell über das Tauchrohr abgelassen.	B	III	IST: • Die Arbeiten werden in einer belüfteten Kapelle durchgeführt. SOLL: • Für kritische Schritte werden Auffangwannen vorbereitet. • Die verschüttete Lösung wird mit Wasser verdünnt. • Werden grössere Mengen (> 10 ml) Hydroxylamin-Lösung verschüttet, muss die Kapellenscheibe heruntergefahren werden. Im Anschluss wird ein Warnschild aufgestellt und der Vorgesetzte informiert. Das Experiment darf nur zusammen mit dem Vorgesetzten fortgesetzt werden.	B	IV
10	Unerwartete Versuchsergebnisse: Der Druck steigt nicht wie erwartet an ODER der Druck steigt zu stark an. Das Experiment weicht ± 30 % von den Erwartungen ab.	C	III	SOLL: • Die Kapellenscheibe wird heruntergefahren. • Die Reaktortemperierung wird laufen gelassen. • Ein Warnschild wird aufgestellt. • Der Vorgesetzte wird informiert. • Das Experiment darf nur zusammen mit dem Vorgesetzten fortgesetzt werden.	C	IV
11	Der Blowdown-Tank wurde nicht mit Stickstoff inertisiert. Beim Entlasten des EXR (via Berstscheibe oder manuell) kann eine explosive Atmosphäre entstehen. Explosion in der Gasphase Explosion. Die Berstscheibe des Blowdown-Tanks öffnet.	D	IV	IST: • Der Blow-Down-Tank ist für Explosionen ausgehend von Normaldruck druckstossfest gebaut. SOLL: • Der Vorgesetzte wird informiert. Das Experiment darf nur zusammen mit dem Vorgesetzten fortgesetzt werden.	D	IV
12	Die Ventile 1 bis 3 werden zu Beginn nicht verschlossen. Es baut sich kein Druck im EXR auf.	C	IV	-	-	-

Gefahrennr.	Szenario (Gefahr, Ursache, Auswirkung)	Risiko		Massnahmen	Risiko	
		W	T		W	T
13	Das Ventil 3 ist offen, aber das Ventil 4 ist geschlossen. Der Teflonschlauch platzt durch den Druckaufbau währen dem Zersetzungsversuch.	C	III	IST: • Die Arbeiten werden in einer belüfteten Kapelle durchgeführt. SOLL: • Der Teflon-Schlauch wird durch einen druckfesten Schlauch ersetzt.	C	IV
14	Der Reaktor ist beim Befüllen bereits aufgeheizt. Die Zersetzung startet bereits während dem Einziehen der Hydroxylamin-Lösung. Die Lösung kann nicht mehr eingezogen werden.	B	IV	-	-	-
15	V8 falsch gestellt: Während dem Ablassen der Gasphase aus dem Reaktor gelangt das Gasgemisch irrtümlich in die Gasdosiereinheit. Manometer an der Gasdosierstation zeigt eine Fehlhandhabung an. Es ist jedoch nicht mit einem Stoffaustritt zu rechnen.	B	IV	SOLL: • Nach einer solchen Fehlhandhabung muss die Gasdosierstation mit Stickstoff gespült werden.	B	IV

FHNW HLS Risikoanalyse EXR HA F. Walthert

Gefahrennr.	Szenario (Gefahr, Ursache, Auswirkung)	Risiko		Massnahmen	Risiko	
		W	T		W	T
16	Die Zersetzung der Hydroxylamin-Lösung verläuft nicht gemäss der in der Literatur vorgeschlagenen Stöchiometrie und es bildet sich eine noch explosive Gasatmosphäre. Bei einem Reaktorinnendruck ab ca. 6 bara muss damit gerechnet werden, dass eine nicht näher spezifizierte Gasphasenexplosion zu einem Explosionsdruck von > 300 bar führt. Die Druckstossfestigkeit des Reaktors ist nicht mehr gegeben -> Bersten des Reaktors.	E	I	SOLL: • Bei einem ersten Versuch werden 20 ml 50% ige Hydroxylamin-Lösung bei Raumtemperatur eingefüllt. Im Anschluss wird der Thermostat des EXR auf 60 °C eingestellt. Es wird ein maximaler Zersetzungsdruck von 20 bar erwartet. Sobald der Reaktordruck 2 bara erreicht hat, wird versucht eine erste Zündung der Gasphase auszulösen (200 W, 15 Sekunden). So lange keine Zündung gemessen wird, kann der Reaktordruck schrittweise auf 5 bara erhöht werden. Wobei jeweils bei 4 und bei 5 bara zwei weitere Zündungsversuche unternommen werden. Im Anschluss werden die Gasphase und die flüssige Phase des Reaktors analysiert. • Bei einem zweiten Versuch werden 20 ml 50% ige Hydroxylamin-Lösung bei Raumtemperatur eingefüllt. Im Anschluss wird der Thermostat des EXR auf 60 °C eingestellt. Sobald der Reaktordruck 5 bara erreicht hat, werden die Gasphase und die flüssige Phase des Reaktors analysiert.	F	I

Anmerkungen

Vor dem Beginn eines Experimentes muss der «Anhang: Checkliste für » beachtet werden.

20

7. Anhang: Risikomatrix

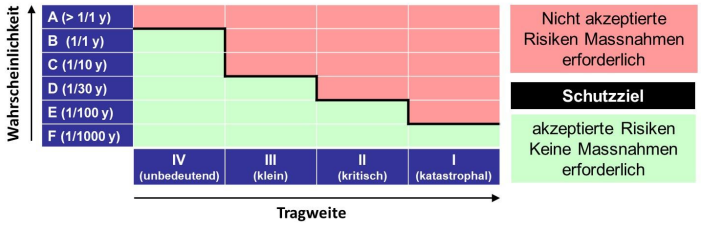
Vorschlag für die Bewertung der Tragweite an der FHNW

	IV (unbedeutend)	III (klein)	II (kritisch)	I (katastrophal)
Personenschaden	Keine Auswirkung	Leichtverletzte (erste Hilfe, Arbeitsausfall <= 3 Tage	Schwerverletzte (medizinische Behandlung; Arbeitsausfall > 3 Tage)	Tote / bleibende Behinderung oder Gesundheitsschäden, Evakuierung des Gebäudes und der Umgebung
Umweltschaden	Sofffreisetzung am Standort der Anlage (regulärer Entsorgungsweg)	Stofffreisetzung im Gebäude bzw. Gebäudeabschnitt	Stofffreisetzung ausserhalb des Gebäudes: Reversible Kurzzeitschäden	Stofffreisetzung ausserhalb des Gebäudes: Irreversible Langzeitschäden (z.B. Fischsterben)
Finanzieller Sachschaden	< 5000 CHF	< 25 kCHF	< 500 kCHF	< 1 Mio. CHF
Betriebsunterbruch	< 1 Woche	< 1 Monat	< 3 Monate	> 3 Monate
Image-Schaden	Kenntnis innerhalb der Arbeitsgruppe	Kenntnis innerhalb der HLS	Regionale Berichterstattung	Überregionale Berichterstattung

Vorschlag für die Bewertung der Wahrscheinlichkeit an der FHNW

A	B	C	D	E	F
> 1 pro Jahr	1 pro Jahr	1 pro 10 Jahre	1 pro 30 Jahre	1 pro 100 Jahre	1 pro 1000 Jahre

Vorschlag für das Schutzziel an der FHNW



21

8. Anhang: Checkliste für Versuche

1. Persönliche Schutzausrüstung wird getragen: Labormantel und Schutzbrille, Handschuhe nur bei offener Handhabung von Flüssigkeiten.
2. Notwege sichergestellt
3. Warnschild bereitgestellt
4. Auffanggefässe bereitgestellt
5. Notkühlmedien wie Eis und Wasser sind bereitgestellt
6. Entsorgungseimer mit Wasser zur Vorverdünnung ist bereitgestellt
7. Kapelle
    a. Lüftung läuft
    b. Kapellenscheibe komplett schliessbar
8. Thermostat
    a. Genügend Thermofluid
    b. Eingeschaltet
    c. Maximaltemperatur von 110 °C eingestellt
9. Blowdown-Tank
    a. Mit Stickstoff inertisiert.
    b. Ventile 5, 6 und 7 geschlossen
10. EXR
    a. Dichtigkeit sichergestellt. Leckrate < 10 mbar/h bei 5 bara.
    b. Richtige Berstscheibe eingebaut (300 bar Berstdruck)
    c. Mit Stickstoff inertisiert und Ventil 8 wieder nach Kühlfalle geöffnet.
    d. Maximal mit 80 mL befüllt
    e. Ventil 4 offen
    f. Ventile 1 bis 3 und 7 geschlossen

9. Entsorgung

Flüssige Abfälle werden so schnell wie möglich mit viel Wasser vorverdünnt (1:10) und anschliessend bei laufendem Wasserhahn kanalisiert (Achtung: NICHT über die PTC-Neutralisations-Anlage).

10. Literatur

[1] PanGas, „Sicherheitsdatenblatt Ammoniak, wasserfrei“. Okt. 31, 2018.
[2] AirLiquide, „Safety Data Sheet Ammonia“. 7 2018.
[3] Charkit, „Safety Data Sheet Hydroxylamine Free Base“. Feb. 18, 2016.
[4] Sigma-Aldrich, „Sicherheitsdatenblatt Hydroxylamin (50%ige Lösung in Wasser) zur Synthese“. Jan. 17, 2019.
[5] Roth, „Sicherheitsdatenblatt Ammoniaklösung ROTIPURAN® 30%“. Feb. 28, 2019.
[6] LONZA, „Sicherheitslabor PSL data sheet: 2208/2695“. Mai 08, 2019.
[7] PanGas, „Sichereheitsdatenblatt Wasserstoff“. Apr. 23, 2020, Zugegriffen: Dez. 06, 2020. [Online]. Verfügbar unter: https://www.pangas.ch/de/images/pangas_sdb_wasserstoff-60_d_tcm553-115273.pdf.
[8] Praxair Inc., „Nitric oxide“. Dez. 18, 2019, Zugegriffen: Dez. 06, 2020. [Online]. Verfügbar unter: https://www.praxair.com/-/media/corporate/praxairus/documents/sds/nitric-oxide-no-safety-data-sheet-sds-p4632.pdf?la=en&rev=f25f21243c5f497285925dcb647bf82c.
[9] AirLiquide, „Stickstoffmonoxid N25“, *AirLiquide*. https://produkte.airliquide.de/loesungen/produkte/gasekatalog/stoffe/stickstoffmonoxidn25.html#stickstoffmonoxidn25-sicherheit (zugegriffen Dez. 06, 2020).
[10] AirLiquide, „Safety Data Sheet Nitrous oxide“. Sep. 18, 2018, Zugegriffen: Dez. 06, 2020. [Online]. Verfügbar unter: http://alsafetydatasheets.com/download/se/Dinitrogenoxid-SE_ENG.pdf.