

```

1
2 % Programm für die technische Simulation der Komponenten
3 % Momentan nur Zuweisung, diese Werte müssen aber irgendwann in dieser
4 % Funktion berechnet werden
5
6 function [techSimuCompVec, techSimuCompValueVec, techCompOutputVec,
techCompSpecOutputVec, errorInLoopsFlag] = ...
7     BgaPtg2TechSimuComp(handles,paramComp,compLoop,paramChain,techSimuPreCompValueVec,
techSimuPreCalcChainVec)
8
9 global const;
10
11 % Damit ohne GUI gearbeitet werden kann.
12 global startGuiFlag;
13
14
15 % Outputs definieren
16 techSimuCompVec           = ones(9,1)*-1;
17 techSimuCompValueVec     = ones(20,1)*-1;
18 techCompOutputVec       = ones(8,1)*-1;
19 techCompSpecOutputVec   = ones(9,1)*-1;
20 errorInLoopsFlag        = false;
21 %return;
22
23
24 % Abfrage, ob hier Kompressor
25 compColInExcel          = paramComp(compLoop).sheet;
26
27 %% Parameter (Anfang) *****
28
29 % compInflRate           =
paramComp(compLoop).valueCompParam(find(strcmp(paramComp(compLoop).txtCompParamInfo(:,
1),'compInflRate')));
30 % compInvest1           =
paramComp(compLoop).valueCompParam(find(strcmp(paramComp(compLoop).txtCompParamInfo(:,
1),'compInvest1')));
31 % compInvest2           =
paramComp(compLoop).valueCompParam(find(strcmp(paramComp(compLoop).txtCompParamInfo(:,
1),'compInvest2')));
32 % compLifetimeHours     =
paramComp(compLoop).valueCompParam(find(strcmp(paramComp(compLoop).txtCompParamInfo(:,
1),'compLifetimeHours')));
33 % compMaintFactor       =
paramComp(compLoop).valueCompParam(find(strcmp(paramComp(compLoop).txtCompParamInfo(:,
1),'compMaintFactor')));
34 % compUpkeepFactor      =
paramComp(compLoop).valueCompParam(find(strcmp(paramComp(compLoop).txtCompParamInfo(:,
1),'compUpkeepFactor')));
35 % compOtherCost         =
paramComp(compLoop).valueCompParam(find(strcmp(paramComp(compLoop).txtCompParamInfo(:,
1),'compOtherCost')));
36 % compElecPrice         =
paramComp(compLoop).valueCompParam(find(strcmp(paramComp(compLoop).txtCompParamInfo(:,
1),'compElecPrice')));
37 % compH2Price           =
paramComp(compLoop).valueCompParam(find(strcmp(paramComp(compLoop).txtCompParamInfo(:,
1),'compH2Price')));
38 % compOtherPrice        =
paramComp(compLoop).valueCompParam(find(strcmp(paramComp(compLoop).txtCompParamInfo(:,
1),'compOtherPrice')));
39 techCompFLH             = paramComp(compLoop).valueCompParam(find(strcmp(paramComp
(compLoop).txtCompParamInfo(:,1),'techCompFLH'))); %input
40 if isempty(techCompFLH)
41     msg = 'Parameter "techCompFLH" bzw. dessen Wert ist nicht definiert.';
42     msg = [char(paramChain.txtChainName), ': ', msg, ' Berechnungen für diese
Prozesskette werden unterbrochen.'];
43     if startGuiFlag == true
44         BgaPtg2UpdateMessageBox(handles,'append',msg);
45     else
46         disp(msg);
47     end
48     errorInLoopsFlag = true;     return;
49 end

```

```

50 % comppp =
paramComp(compLoop).valueCompParam(find(strcmp(paramComp(compLoop).txtCompParamInfo(:,
1), 'compp')));
51 % compT =
paramComp(compLoop).valueCompParam(find(strcmp(paramComp(compLoop).txtCompParamInfo(:,
1), 'compT')));
52 % compConcBioM =
paramComp(compLoop).valueCompParam(find(strcmp(paramComp(compLoop).txtCompParamInfo(:,
1), 'compConcBioM')));
53 % compConcSNG =
paramComp(compLoop).valueCompParam(find(strcmp(paramComp(compLoop).txtCompParamInfo(:,
1), 'compConcSNG')));
54 % compConcCO2 =
paramComp(compLoop).valueCompParam(find(strcmp(paramComp(compLoop).txtCompParamInfo(:,
1), 'compConcCO2')));
55 % compConcH2 =
paramComp(compLoop).valueCompParam(find(strcmp(paramComp(compLoop).txtCompParamInfo(:,
1), 'compConcH2')));
56 % compConcH2S =
paramComp(compLoop).valueCompParam(find(strcmp(paramComp(compLoop).txtCompParamInfo(:,
1), 'compConcH2S')));
57 % compConcH2O =
paramComp(compLoop).valueCompParam(find(strcmp(paramComp(compLoop).txtCompParamInfo(:,
1), 'compConcH2O')));
58 % compConcTraceGas =
paramComp(compLoop).valueCompParam(find(strcmp(paramComp(compLoop).txtCompParamInfo(:,
1), 'compConcTraceGas')));
59 % compConcOther =
paramComp(compLoop).valueCompParam(find(strcmp(paramComp(compLoop).txtCompParamInfo(:,
1), 'compConcOther')));
60 compDeltaVolFlowBioM = paramComp(compLoop).valueCompParam(find(strcmp(paramComp
(compLoop).txtCompParamInfo(:,1), 'compDeltaVolFlowBioM')));
61 if isempty(compDeltaVolFlowBioM)
62 msg = 'Parameter "compDeltaVolFlowBioM" bzw. dessen Wert ist nicht definiert.';
63 msg = [char(paramChain.txtChainName), ': ', msg, ' Berechnungen für diese
Prozesskette werden unterbrochen.'];
64 if startGuiFlag == true
65 BgaPtg2UpdateMessageBox(handles, 'apend', msg);
66 else
67 disp(msg);
68 end
69 errorInLoopsFlag = true; return;
70 end
71
72 compDeltaVolFlowSNG = paramComp(compLoop).valueCompParam(find(strcmp(paramComp
(compLoop).txtCompParamInfo(:,1), 'compDeltaVolFlowSNG')));
73 if isempty(compDeltaVolFlowSNG)
74 msg = 'Parameter "compDeltaVolFlowSNG" bzw. dessen Wert ist nicht definiert.';
75 msg = [char(paramChain.txtChainName), ': ', msg, ' Berechnungen für diese
Prozesskette werden unterbrochen.'];
76 if startGuiFlag == true
77 BgaPtg2UpdateMessageBox(handles, 'apend', msg);
78 else
79 disp(msg);
80 end
81 errorInLoopsFlag = true; return;
82 end
83
84 compDeltaVolFlowCO2 = paramComp(compLoop).valueCompParam(find(strcmp(paramComp
(compLoop).txtCompParamInfo(:,1), 'compDeltaVolFlowCO2')));
85 if isempty(compDeltaVolFlowCO2)
86 msg = 'Parameter "compDeltaVolFlowCO2" bzw. dessen Wert ist nicht definiert.';
87 msg = [char(paramChain.txtChainName), ': ', msg, ' Berechnungen für diese
Prozesskette werden unterbrochen.'];
88 if startGuiFlag == true
89 BgaPtg2UpdateMessageBox(handles, 'apend', msg);
90 else
91 disp(msg);
92 end
93 errorInLoopsFlag = true; return;
94 end
95
96 compDeltaVolFlowH2 = paramComp(compLoop).valueCompParam(find(strcmp(paramComp

```

```

(compLoop).txtCompParamInfo(:,1),'compDeltaVolFlowH2')));
97 if isempty(compDeltaVolFlowH2)
98     msg = 'Parameter "compDeltaVolFlowH2" bzw. dessen Wert ist nicht definiert.';
99     msg = [char(paramChain.txtChainName), ': ', msg, ' Berechnungen für diese
Prozesskette werden unterbrochen.'];
100     if startGuiFlag == true
101         BgaPtg2UpdateMessageBox(handles,'apend',msg);
102     else
103         disp(msg);
104     end
105     errorInLoopsFlag = true;     return;
106 end
107
108 compDeltaVolFlowH2S      = paramComp(compLoop).valueCompParam(find(strcmp(paramComp
(compLoop).txtCompParamInfo(:,1),'compDeltaVolFlowH2S')));
109 if isempty(compDeltaVolFlowH2S)
110     msg = 'Parameter "compDeltaVolFlowH2S" bzw. dessen Wert ist nicht definiert.';
111     msg = [char(paramChain.txtChainName), ': ', msg, ' Berechnungen für diese
Prozesskette werden unterbrochen.'];
112     if startGuiFlag == true
113         BgaPtg2UpdateMessageBox(handles,'apend',msg);
114     else
115         disp(msg);
116     end
117     errorInLoopsFlag = true;     return;
118 end
119
120 compDeltaVolFlowH2O      = paramComp(compLoop).valueCompParam(find(strcmp(paramComp
(compLoop).txtCompParamInfo(:,1),'compDeltaVolFlowH2O')));
121 if isempty(compDeltaVolFlowH2O)
122     msg = 'Parameter "compDeltaVolFlowH2O" bzw. dessen Wert ist nicht definiert.';
123     msg = [char(paramChain.txtChainName), ': ', msg, ' Berechnungen für diese
Prozesskette werden unterbrochen.'];
124     if startGuiFlag == true
125         BgaPtg2UpdateMessageBox(handles,'apend',msg);
126     else
127         disp(msg);
128     end
129     errorInLoopsFlag = true;     return;
130 end
131
132 compDeltaVolFlowTraceGas = paramComp(compLoop).valueCompParam(find(strcmp(paramComp
(compLoop).txtCompParamInfo(:,1),'compDeltaVolFlowTraceGas')));
133 if isempty(compDeltaVolFlowTraceGas)
134     msg = 'Parameter "compDeltaVolFlowTraceGas" bzw. dessen Wert ist nicht definiert.'
;
135     msg = [char(paramChain.txtChainName), ': ', msg, ' Berechnungen für diese
Prozesskette werden unterbrochen.'];
136     if startGuiFlag == true
137         BgaPtg2UpdateMessageBox(handles,'apend',msg);
138     else
139         disp(msg);
140     end
141     errorInLoopsFlag = true;     return;
142 end
143
144 compDeltaVolFlowOther    = paramComp(compLoop).valueCompParam(find(strcmp(paramComp
(compLoop).txtCompParamInfo(:,1),'compDeltaVolFlowOther')));
145 if isempty(compDeltaVolFlowOther)
146     msg = 'Parameter "compDeltaVolFlowOther" bzw. dessen Wert ist nicht definiert.';
147     msg = [char(paramChain.txtChainName), ': ', msg, ' Berechnungen für diese
Prozesskette werden unterbrochen.'];
148     if startGuiFlag == true
149         BgaPtg2UpdateMessageBox(handles,'apend',msg);
150     else
151         disp(msg);
152     end
153     errorInLoopsFlag = true;     return;
154 end
155
156 compDeltap               = paramComp(compLoop).valueCompParam(find(strcmp(paramComp
(compLoop).txtCompParamInfo(:,1),'compDeltap')));
157 if isempty(compDeltap)

```

```

158     msg = 'Parameter "compDeltap" bzw. dessen Wert ist nicht definiert.';
159     msg = [char(paramChain.txtChainName), ': ', msg, ' Berechnungen für diese
        Prozesskette werden unterbrochen.'];
160     if startGuiFlag == true
161         BgaPtg2UpdateMessageBox(handles,'apend',msg);
162     else
163         disp(msg);
164     end
165     errorInLoopsFlag = true;     return;
166 end
167
168 compLeakage          = paramComp(compLoop).valueCompParam(find(strcmp(paramComp
(compLoop).txtCompParamInfo(:,1),'compLeakage')));
169 if isempty(compLeakage)
170     msg = 'Parameter "compLeakage" bzw. dessen Wert ist nicht definiert.';
171     msg = [char(paramChain.txtChainName), ': ', msg, ' Berechnungen für diese
        Prozesskette werden unterbrochen.'];
172     if startGuiFlag == true
173         BgaPtg2UpdateMessageBox(handles,'apend',msg);
174     else
175         disp(msg);
176     end
177     errorInLoopsFlag = true;     return;
178 end
179
180 compWaterFactor     = paramComp(compLoop).valueCompParam(find(strcmp(paramComp
(compLoop).txtCompParamInfo(:,1),'compWaterFactor')));
181 if isempty(compWaterFactor)
182     msg = 'Parameter "compWaterFactor" bzw. dessen Wert ist nicht definiert.';
183     msg = [char(paramChain.txtChainName), ': ', msg, ' Berechnungen für diese
        Prozesskette werden unterbrochen.'];
184     if startGuiFlag == true
185         BgaPtg2UpdateMessageBox(handles,'apend',msg);
186     else
187         disp(msg);
188     end
189     errorInLoopsFlag = true;     return;
190 end
191
192 compActCH2S         = paramComp(compLoop).valueCompParam(find(strcmp(paramComp
(compLoop).txtCompParamInfo(:,1),'compActCH2S')));
193 if isempty(compActCH2S)
194     msg = 'Parameter "compActCH2S" bzw. dessen Wert ist nicht definiert.';
195     msg = [char(paramChain.txtChainName), ': ', msg, ' Berechnungen für diese
        Prozesskette werden unterbrochen.'];
196     if startGuiFlag == true
197         BgaPtg2UpdateMessageBox(handles,'apend',msg);
198     else
199         disp(msg);
200     end
201     errorInLoopsFlag = true;     return;
202 end
203
204 compNutrientVolFlowFactor = paramComp(compLoop).valueCompParam(find(strcmp(paramComp
(compLoop).txtCompParamInfo(:,1),'compNutrientVolFlowFactor')));
205 if isempty(compNutrientVolFlowFactor)
206     msg = 'Parameter "compNutrientVolFlowFactor" bzw. dessen Wert ist nicht
        definiert.';
207     msg = [char(paramChain.txtChainName), ': ', msg, ' Berechnungen für diese
        Prozesskette werden unterbrochen.'];
208     if startGuiFlag == true
209         BgaPtg2UpdateMessageBox(handles,'apend',msg);
210     else
211         disp(msg);
212     end
213     errorInLoopsFlag = true;     return;
214 end
215
216 % compOtherFactor          =
paramComp(compLoop).valueCompParam(find(strcmp(paramComp(compLoop).txtCompParamInfo(:,
1),'compOtherFactor')));
217 compElecSpec          = paramComp(compLoop).valueCompParam(find(strcmp(paramComp
(compLoop).txtCompParamInfo(:,1),'compElecSpec')));

```

```

218 if isempty(compElecSpec)
219     msg = 'Parameter "compElecSpec" bzw. dessen Wert ist nicht definiert.';
220     msg = [char(paramChain.txtChainName), ': ', msg, ' Berechnungen für diese
        Prozesskette werden unterbrochen.'];
221     if startGuiFlag == true
222         BgaPtg2UpdateMessageBox(handles,'apend',msg);
223     else
224         disp(msg);
225     end
226     errorInLoopsFlag = true;     return;
227 end
228
229 compHeatSpec = paramComp(compLoop).valueCompParam(find(strcmp(paramComp
(compLoop).txtCompParamInfo(:,1),'compHeatSpec')));
230 if isempty(compHeatSpec)
231     msg = 'Parameter "compHeatSpec" bzw. dessen Wert ist nicht definiert.';
232     msg = [char(paramChain.txtChainName), ': ', msg, ' Berechnungen für diese
        Prozesskette werden unterbrochen.'];
233     if startGuiFlag == true
234         BgaPtg2UpdateMessageBox(handles,'apend',msg);
235     else
236         disp(msg);
237     end
238     errorInLoopsFlag = true;     return;
239 end
240
241 % compHeatIdealHT =
paramComp(compLoop).valueCompParam(find(strcmp(paramComp(compLoop).txtCompParamInfo(:,
1),'compHeatIdealHT')));
242 % compHeatIdealNT =
paramComp(compLoop).valueCompParam(find(strcmp(paramComp(compLoop).txtCompParamInfo(:,
1),'compHeatIdealNT')));
243 compConvCO2Ideal = paramComp(compLoop).valueCompParam(find(strcmp(paramComp
(compLoop).txtCompParamInfo(:,1),'compConvCO2Ideal')));
244 if isempty(compConvCO2Ideal)
245     msg = 'Parameter "compConvCO2Ideal" bzw. dessen Wert ist nicht definiert.';
246     msg = [char(paramChain.txtChainName), ': ', msg, ' Berechnungen für diese
        Prozesskette werden unterbrochen.'];
247     if startGuiFlag == true
248         BgaPtg2UpdateMessageBox(handles,'apend',msg);
249     else
250         disp(msg);
251     end
252     errorInLoopsFlag = true;     return;
253 end
254
255 compConvCO2Real = paramComp(compLoop).valueCompParam(find(strcmp(paramComp
(compLoop).txtCompParamInfo(:,1),'compConvCO2Real')));
256 if isempty(compConvCO2Real)
257     msg = 'Parameter "compConvCO2Real" bzw. dessen Wert ist nicht definiert.';
258     msg = [char(paramChain.txtChainName), ': ', msg, ' Berechnungen für diese
        Prozesskette werden unterbrochen.'];
259     if startGuiFlag == true
260         BgaPtg2UpdateMessageBox(handles,'apend',msg);
261     else
262         disp(msg);
263     end
264     errorInLoopsFlag = true;     return;
265 end
266
267 % Chain-Parameter lesen
268 chainElyFLH = paramChain.valueChainParam(find(strcmp(paramChain.
txtChainParamInfo(:,1),'chainElyFLH')));
269 if isempty(chainElyFLH)
270     msg = 'Parameter "chainElyFLH" bzw. dessen Wert ist nicht definiert.';
271     msg = [char(paramChain.txtChainName), ': ', msg, ' Berechnungen für diese
        Prozesskette werden unterbrochen.'];
272     if startGuiFlag == true
273         BgaPtg2UpdateMessageBox(handles,'apend',msg);
274     else
275         disp(msg);
276     end
277     errorInLoopsFlag = true;     return;

```

```

278 end
279
280 chainHeatUseHT = paramChain.valueChainParam(find(strcmp(paramChain.
txtChainParamInfo(:,1),'chainHeatUseHT')));
281 if isempty(chainHeatUseHT)
282     msg = 'Parameter "chainHeatUseHT" bzw. dessen Wert ist nicht definiert.';
283     msg = [char(paramChain.txtChainName), ': ', msg, ' Berechnungen für diese
Prozesskette werden unterbrochen.'];
284     if startGuiFlag == true
285         BgaPtg2UpdateMessageBox(handles,'apend',msg);
286     else
287         disp(msg);
288     end
289     errorInLoopsFlag = true; return;
290 end
291
292 chainHeatUseNT = paramChain.valueChainParam(find(strcmp(paramChain.
txtChainParamInfo(:,1),'chainHeatUseNT')));
293 if isempty(chainHeatUseNT)
294     msg = 'Parameter "chainHeatUseNT" bzw. dessen Wert ist nicht definiert.';
295     msg = [char(paramChain.txtChainName), ': ', msg, ' Berechnungen für diese
Prozesskette werden unterbrochen.'];
296     if startGuiFlag == true
297         BgaPtg2UpdateMessageBox(handles,'apend',msg);
298     else
299         disp(msg);
300     end
301     errorInLoopsFlag = true; return;
302 end
303
304 chainComprEff = paramChain.valueChainParam(find(strcmp(paramChain.
txtChainParamInfo(:,1),'chainComprEff')));
305 if isempty(chainComprEff)
306     msg = 'Parameter "chainComprEff" bzw. dessen Wert ist nicht definiert.';
307     msg = [char(paramChain.txtChainName), ': ', msg, ' Berechnungen für diese
Prozesskette werden unterbrochen.'];
308     if startGuiFlag == true
309         BgaPtg2UpdateMessageBox(handles,'apend',msg);
310     else
311         disp(msg);
312     end
313     errorInLoopsFlag = true; return;
314 end
315
316 chainStoichiometry = paramChain.valueChainParam(find(strcmp(paramChain.
txtChainParamInfo(:,1),'chainStoichiometry')));
317 if isempty(chainStoichiometry)
318     msg = 'Parameter "chainStoichiometry" bzw. dessen Wert ist nicht definiert.';
319     msg = [char(paramChain.txtChainName), ': ', msg, ' Berechnungen für diese
Prozesskette werden unterbrochen.'];
320     if startGuiFlag == true
321         BgaPtg2UpdateMessageBox(handles,'apend',msg);
322     else
323         disp(msg);
324     end
325     errorInLoopsFlag = true; return;
326 end
327
328
329 compHeatIdealHT = 0;
330 if(strcmp(compColInExcel,'AY'))
331     compHeatIdealHT = paramComp(compLoop-1).valueCompParam(find(strcmp(paramComp(
compLoop-1).txtCompParamInfo(:,1),'compHeatIdealHT')));
332     if isempty(compHeatIdealHT)
333         msg = 'Parameter "compHeatIdealHT" bzw. dessen Wert ist nicht definiert.';
334         msg = [char(paramChain.txtChainName), ': ', msg, ' Berechnungen für diese
Prozesskette werden unterbrochen.'];
335         if startGuiFlag == true
336             BgaPtg2UpdateMessageBox(handles,'apend',msg);
337         else
338             disp(msg);
339         end
340         errorInLoopsFlag = true; return;

```



```

341     end
342 end
343
344 compHeatIdealNT = 0;
345 if(strcmp(compColInExcel,'AZ'))
346     compColVorInExcel = paramComp(compLoop-1).sheet;
347
348     if(strcmp(compColVorInExcel,'AY'))
349         compHeatIdealNT = paramComp(compLoop-2).valueCompParam(find(strcmp(paramComp(
350             compLoop-2).txtCompParamInfo(:,1),'compHeatIdealNT')));
351         % Veränderung in _v3.py
352     else
353         compHeatIdealNT = paramComp(compLoop-1).valueCompParam(find(strcmp(paramComp(
354             compLoop-1).txtCompParamInfo(:,1),'compHeatIdealNT')));
355     end
356
357     if isempty(compHeatIdealNT)
358         msg = 'Parameter "compHeatIdealNT" bzw. dessen Wert ist nicht definiert.';
359         msg = [char(paramChain.txtChainName), ': ', msg, ' Berechnungen für diese
360             Prozesskette werden unterbrochen.'];
361         if startGuiFlag == true
362             BgaPtg2UpdateMessageBox(handles,'apend',msg);
363         else
364             disp(msg);
365         end
366         errorInLoopsFlag = true;     return;
367     end
368 end
369
370 % compp                                = techSimuPreCompValueVec(1);
371 % compT                                = techSimuPreCompValueVec(2);
372 % compVolFlowBioM                      = techSimuPreCompValueVec(3);
373 % compVolFlowSNG                       = techSimuPreCompValueVec(4);
374 % compVolFlowCO2                       = techSimuPreCompValueVec(5);
375 % compVolFlowH2                        = techSimuPreCompValueVec(6);
376 % compVolFlowH2S                       = techSimuPreCompValueVec(7);
377 % compVolFlowH2O                       = techSimuPreCompValueVec(8);
378 % compVolFlowTraceGas                  = techSimuPreCompValueVec(9);
379 % compVolFlowOther                     = techSimuPreCompValueVec(10);
380
381 %% Parameter (Ende) *****
382 % Konvertierungen
383 % compInflRate                         = compInflRate/100;
384 % compMaintFactor                      = compMaintFactor/100;
385 % compUpkeepFactor                     = compUpkeepFactor/100;
386 % compElecPrice                        = compElecPrice/100;
387 % compp                                = compp*(10^5);
388 compDeltap                             = compDeltap * (10^5);
389 % compT                                = compT + 273.15;
390 % compConcBioM                         = compConcBioM/100;
391 % compConcSNG                          = compConcSNG/100;
392 % compConcCO2                          = compConcCO2/100;
393 % compConcH2                           = compConcH2/100;
394 % compConcH2S                          = compConcH2S/(10^6);
395 % compConcH2O                          = compConcH2O/100;
396 % compConcTraceGas                     = compConcTraceGas/(10^6);
397 % compConcOther                        = compConcOther/100;
398 compDeltaVolFlowBioM                   = compDeltaVolFlowBioM/100;
399 compDeltaVolFlowSNG                    = compDeltaVolFlowSNG/100;
400 compDeltaVolFlowCO2                    = compDeltaVolFlowCO2/100;
401 compDeltaVolFlowH2                     = compDeltaVolFlowH2/100;
402 compDeltaVolFlowH2S                    = compDeltaVolFlowH2S/100;
403 compDeltaVolFlowH2O                    = compDeltaVolFlowH2O/100;
404 compDeltaVolFlowTraceGas                = compDeltaVolFlowTraceGas/100;
405 compDeltaVolFlowOther                   = compDeltaVolFlowOther/100;
406 compLeakage                            = compLeakage/100;
407 % compOtherFactor                      = compOtherFactor/100;
408 compNutrientVolFlowFactor              = compNutrientVolFlowFactor/100;
409 compActCH2S                            = compActCH2S/100;
410 compConvCO2Ideal                       = compConvCO2Ideal/100;
411 compConvCO2Real                        = compConvCO2Real/100;

```

```

411 % chain Parameter
412 % chainDiscRate           = chainDiscRate/100;
413 % chainElecPrice          = chainElecPrice/100;
414 % chainOperationFactor    = chainOperationFactor/100;
415 % chainInsurFactor        = chainInsurFactor/100;
416 % chainRenatFactor        = chainRenatFactor/100;
417 % chainTechInflRate       = chainTechInflRate/100;
418 % chainBuildInflRate      = chainBuildInflRate/100;
419 % chainInflRate           = chainInflRate/100;
420 % chainWaterPrice         = chainWaterPrice;
421 % chainBGPurchasePrice    = chainBGPurchasePrice/100;
422 % chainBMSellingPrice     = chainBMSellingPrice/100;
423 % chainHeatPurchasePrice  = chainHeatPurchasePrice/100;
424 % chainHeatSellingPrice   = chainHeatSellingPrice/100;
425 % chainO2SellingPrice     = chainO2SellingPrice/100;
426 chainComprEff            = chainComprEff/100;
427 % chainEffMeth            = chainEffMeth/100;
428 % chainEffCHP             = chainEffCHP/100;
429 % chainEffEly             = chainEffEly/100;
430 chainHeatUseHT           = chainHeatUseHT/100;
431 chainHeatUseNT           = chainHeatUseNT/100;
432
433
434
435 % Konstanten lesen
436 %density_H2S              = const.density_H2S;
437
438 molarmass_CH4             = const.molarmass_CH4;
439 molarmass_CO2             = const.molarmass_CO2;
440 molarmass_H2              = const.molarmass_H2;
441 molarmass_H2S             = const.molarmass_H2S;
442 molarmass_H2O             = const.molarmass_H2O;
443 molarmass_Other           = const.molarmass_Other;
444 molarmass_TraceGas       = const.molarmass_TraceGas;
445
446 cp_CH4                    = const.cp_CH4;
447 cp_CO2                    = const.cp_CO2;
448 cp_H2                     = const.cp_H2;
449 cp_H2S                    = const.cp_H2S;
450 cp_H2O                    = const.cp_H2O;
451 cp_Other                  = const.cp_Other;
452 cp_TraceGas               = const.cp_TraceGas;
453
454 R                          = const.R;
455
456 density_CH4                = const.density_CH4;
457 density_CO2                = const.density_CO2;
458 density_H2                 = const.density_H2;
459 density_H2S                = const.density_H2S;
460 density_H2O                = const.density_H2O;
461 density_Other              = const.density_Other;
462 density_TraceGas           = const.density_TraceGas;
463
464 % techChainVolFlowBG = techSimuPreCalcChainVec(1);
465 % chainVolFlowCO2   = techSimuPreCalcChainVec(2);
466 % chainVolFlowH2    = techSimuPreCalcChainVec(3);
467 chainVolFlowH2kg    = techSimuPreCalcChainVec(4);
468 techChainMethPower  = techSimuPreCalcChainVec(5);
469 % techChainElyPower = techSimuPreCalcChainVec(6);
470 techChainpSystem    = techSimuPreCalcChainVec(7);
471 chainp4Compr        = techSimuPreCalcChainVec(8);
472
473
474 % compPower wird nur in WSimu für Berechnung Invest in Mutltiplikation mit
475 compInvest1 (spez.Invest/Leistung der Komponente) genutzt.
476 % compInvest1 wird nun aber auf MethPower und nicht auf die Leistung der einzelnen
477 Komponente bezogen
478 compPower = techChainMethPower;
479
480 % Wenn die aktuelle Komponente die Elektrolyse ist, dann nimm den chain-Parameter für
481 die VLS
482 if paramComp(1).sheet == 'D'
483     techCompFLH = chainElyFLH;

```



```

481 % Ansonsten, wenn bei techCompFLH Null steht, nimm den chain-Wert
482 elseif techCompFLH == 0
483     techCompFLH = techChainFLH;
484 % In allen anderen Fällen darf der comp Wert genommen werden
485 else
486     techCompFLH = techCompFLH;
487 end
488
489
490 % Beispiel für die Weitergabe von benötigten Größen
491 compp = techSimuPreCompValueVec (1);
492 compT = techSimuPreCompValueVec (2);
493 compVolFlowBioM = techSimuPreCompValueVec (3);
494 compVolFlowSNG = techSimuPreCompValueVec (4);
495 compVolFlowCO2 = techSimuPreCompValueVec (5);
496 compVolFlowH2 = techSimuPreCompValueVec (6);
497 compVolFlowH2S = techSimuPreCompValueVec (7);
498 compVolFlowH2O = techSimuPreCompValueVec (8);
499 compVolFlowTraceGas = techSimuPreCompValueVec (9);
500 compVolFlowOther = techSimuPreCompValueVec (10);
501
502 % Wenn Komponente Methanisierung aktiv, dann folgende Berechnungen durchführen
503 if (strcmp(compColInExcel,'AK') || strcmp(compColInExcel,'AL') || strcmp(
504     compColInExcel,'AM') || ...
505     strcmp(compColInExcel,'AN') || strcmp(compColInExcel,'AO') || strcmp(
506     compColInExcel,'AP') || ...
507     strcmp(compColInExcel,'AQ') || strcmp(compColInExcel,'AR') || strcmp(
508     compColInExcel,'AS') || ...
509     strcmp(compColInExcel,'AT') || strcmp(compColInExcel,'AU') || strcmp(
510     compColInExcel,'AV') || ...
511     strcmp(compColInExcel,'AW') || strcmp(compColInExcel,'AX'))
512
513     compVolFlowSNG = compVolFlowCO2 * compDeltaVolFlowCO2;
514     compVolFlowH2 = compVolFlowH2 + chainStoichiometry * compVolFlowCO2 * (1 -
515         compDeltaVolFlowCO2);
516 end
517
518 %compp = compp - compDeltap; % weiter nutzen
519 %compT = compT * (1-compDeltaT) % nicht benutzen
520
521 compVolFlowBioM = compVolFlowBioM * (1 - compDeltaVolFlowBioM -
522     compLeakage);
523 compVolFlowSNG = compVolFlowSNG * (1 - compDeltaVolFlowSNG -
524     compLeakage);
525 compVolFlowCO2 = compVolFlowCO2 * (1 - compDeltaVolFlowCO2 -
526     compLeakage);
527 compVolFlowH2 = compVolFlowH2 * (1 - compDeltaVolFlowH2 -
528     compLeakage);
529 compVolFlowH2S = compVolFlowH2S * (1 - compDeltaVolFlowH2S -
530     compLeakage);
531 compVolFlowH2O = compVolFlowH2O * (1 - compDeltaVolFlowH2O -
532     compLeakage);
533 compVolFlowTraceGas = compVolFlowTraceGas * (1 - compDeltaVolFlowTraceGas -
534     compLeakage);
535 compVolFlowOther = compVolFlowOther * (1 - compDeltaVolFlowOther -
536     compLeakage);
537
538 % Berechnung BG-Volumenstrom als Summe der Teilströme
539 compVolFlowPG = compVolFlowBioM + compVolFlowSNG + compVolFlowCO2 + compVolFlowH2 +
540     compVolFlowH2S + compVolFlowH2O + compVolFlowTraceGas + compVolFlowOther;
541
542 if compVolFlowPG == 0
543     msg = 'Variable "compVolFlowPG" ist gleich null. Division durch null.';
544     msg = [char(paramChain.txtChainName), ': ', msg, ' Berechnungen für diese
545         Prozesskette werden unterbrochen.'];
546     if startGuiFlag == true
547         BgaPtg2UpdateMessageBox(handles,'append',msg);
548     else
549         disp(msg);
550     end
551     errorInLoopsFlag = true; return;
552 else

```

```

539     % Berechnung Konzentrationen
540     compConcBioM      = compVolFlowBioM / compVolFlowPG;
541     compConcSNG      = compVolFlowSNG / compVolFlowPG;
542     compConcCO2      = compVolFlowCO2 / compVolFlowPG;
543     compConcH2       = compVolFlowH2 / compVolFlowPG;
544     compConcH2S      = compVolFlowH2S / compVolFlowPG;
545     compConcH2O      = compVolFlowH2O / compVolFlowPG;
546     compConcTraceGas = compVolFlowTraceGas / compVolFlowPG;
547     compConcOther    = compVolFlowOther / compVolFlowPG;
548 end;
549
550 % Outputs für Results
551 techCompVolFlowPG   = compVolFlowPG;
552 techCompConcCH4     = (compConcBioM + compConcSNG) * 100;
553 techCompConcCO2     = compConcCO2 * 100;
554 techCompConcH2      = compConcH2 * 100;
555 techCompConcH2S     = compConcH2S * 10^6;
556 techCompConcH2O     = compConcH2O * 100;
557 techCompConcTraceGas = compConcTraceGas * 10^6;
558 techCompConcOther   = compConcOther * 100;
559
560 techCompActCDemand  = compActCH2S * compVolFlowH2S * density_H2S * techCompFLH;
561 compNutrientDemand = compNutrientVolFlowFactor * compVolFlowSNG * techCompFLH;
562 compWaterDemand    = compWaterFactor * chainVolFlowH2kg * techCompFLH * density_H2O /
    1000;
563
564 % % compHeat ideal aus Methanisierung ziehen, Use aus WT
565 % % überarbeiten
566 % % Parameter lesen
567 % if(compLoop == 1)
568 %     compHeatIdealHTBefore = 0;
569 %     compHeatIdealNTVBefore = 0;
570 %
571 % else
572 %     compHeatIdealHTBefore =
paramComp(compLoop-1).valueCompParam(find(strcmp(paramComp(compLoop-1).txtCompParamInfo(:,1), 'compHeatIdealHT')));
573 %     if isempty(compHeatIdealHTBefore)
574 %         msg = 'Parameter "compHeatIdealHTBefore" bzw. dessen Wert ist nicht
definiert.';
575 %         if startGuiFlag == true
576 %             BgaPtg2UpdateMessageBox(handles, 'append', msg);
577 %         else
578 %             disp(msg);
579 %         end
580 %         errordlg(msg);
581 %     end
582 %
583 %     compHeatIdealNTVBefore =
paramComp(compLoop-1).valueCompParam(find(strcmp(paramComp(compLoop-1).txtCompParamInfo(:,1), 'compHeatIdealNT')));
584 %     if isempty(compHeatIdealNTVBefore)
585 %         msg = 'Parameter "compHeatIdealNTVBefore" bzw. dessen Wert ist nicht
definiert.';
586 %         if startGuiFlag == true
587 %             BgaPtg2UpdateMessageBox(handles, 'append', msg);
588 %         else
589 %             disp(msg);
590 %         end
591 %         errordlg(msg);
592 %     end
593 % end
594
595
596
597 compHeatHT_ideal = compHeatIdealHT * techChainMethPower * techCompFLH;
598 compHeatNT_ideal = compHeatIdealNT * techChainMethPower * techCompFLH;
599
600 compHeatHT_real = compHeatHT_ideal * chainHeatUseHT;
601 compHeatNT_real = compHeatNT_ideal * chainHeatUseNT;
602
603 % Hier checken
604 techCompHeatDemand = compHeatSpec * techChainMethPower * techCompFLH;

```

```

605
606 % Platzhalter
607 techCompOtherDemand = 0;
608
609 % geht in WSimu!
610 compHeatSurplus = compHeatHT_real + compHeatNT_real;
611
612
613 %compHeatIdea
614 %% Berechnung Kompressorarbeit und Leistung
615
616 % Masse pro mol je Gas
617 Mass_CH4      = (compConcBioM + compConcSNG) * molarmass_CH4;
618 Mass_CO2      = compConcCO2 * molarmass_CO2;
619 Mass_H2       = compConcH2 * molarmass_H2;
620 Mass_H2S      = compConcH2S * molarmass_H2S;
621 Mass_H2O      = compConcH2O * molarmass_H2O;
622 Mass_Other    = compConcOther * molarmass_Other;
623 Mass_TraceGas = compConcTraceGas * molarmass_TraceGas;
624
625 molarmass_PG = Mass_CH4 + Mass_CO2 + Mass_H2 + Mass_H2S + Mass_H2O + Mass_Other +
Mass_TraceGas;
626
627 Cp_CH4        = cp_CH4 * molarmass_CH4;
628 Cp_CO2        = cp_CO2 * molarmass_CO2;
629 Cp_H2         = cp_H2 * molarmass_H2;
630 Cp_H2S        = cp_H2S * molarmass_H2S;
631 Cp_H2O        = cp_H2O * molarmass_H2O;
632 Cp_Other      = cp_Other * molarmass_Other;
633 Cp_TraceGas   = Cp_H2S; % Hier aufpassen
634
635 % Massen-%
636 if molarmass_PG == 0
637     msg = 'Variable "molarmass_PG" ist gleich null. Division durch null.';
638     msg = [char(paramChain.txtChainName), ': ', msg, ' Berechnungen für diese
Prozesskette werden unterbrochen.'];
639     if startGuiFlag == true
640         BgaPtg2UpdateMessageBox(handles,'apend',msg);
641     else
642         disp(msg);
643     end
644     errorInLoopsFlag = true;     return;
645
646 else
647     MassProzCH4      = Mass_CH4/molarmass_PG;
648     MassProzCO2      = Mass_CO2/molarmass_PG;
649     MassProzH2       = Mass_H2/molarmass_PG;
650     MassProzH2S      = Mass_H2S/molarmass_PG;
651     MassProzH2O      = Mass_H2O/molarmass_PG;
652     MassProzOther    = Mass_Other/molarmass_PG;
653     MassProzTraceGas = Mass_TraceGas/molarmass_PG;
654 end;
655
656
657 % Kappa Gasmischung Prozessgas PG
658 % Massen-%
659 if Cp_CH4-R == 0 || Cp_CO2-R == 0 || Cp_H2-R == 0 || Cp_H2S-R == 0 || Cp_H2O-R == 0 ||
Cp_Other-R == 0 || Cp_TraceGas-R == 0
660     msg = 'Variable "Kappa_PG" kann nicht berechnet werden. Division durch null.';
661     msg = [char(paramChain.txtChainName), ': ', msg, ' Berechnungen für diese
Prozesskette werden unterbrochen.'];
662     if startGuiFlag == true
663         BgaPtg2UpdateMessageBox(handles,'apend',msg);
664     else
665         disp(msg);
666     end
667     errorInLoopsFlag = true;     return;
668
669 else
670     Kappa_PG = MassProzCH4 * Cp_CH4/(Cp_CH4-R) + MassProzCO2 * Cp_CO2/(Cp_CO2-R) + ...
671     MassProzH2 * Cp_H2/(Cp_H2-R) + MassProzH2S * Cp_H2S/(Cp_H2S-R) + ...
672     MassProzH2O * Cp_H2O/(Cp_H2O-R) + MassProzOther * Cp_Other/(Cp_Other-R)
+ MassProzTraceGas * Cp_TraceGas/(Cp_TraceGas-R);

```

```

673 end;
674
675 % Normdichte % Wird das über Mass% oder Vol% gerechnet... ? % VOL%????
676 % überarbeiten
677 compDensity_PG = (compConcBioM + compConcSNG) * density_CH4 + compConcCO2 *
density_CO2 + compConcH2 * density_H2 + ...
678 compConcH2S * density_H2S + compConcH2O * density_H2O + compConcOther
* density_Other + compConcTraceGas * density_TraceGas; % dichte
Teilströme * Mass%
679
680
681 if(strcmp(compColInExcel,'AK'))
682 compp = paramComp(compLoop).valueCompParam(find(strcmp(paramComp(compLoop).
txtCompParamInfo(:,1),'compp')) * 10^5; % Druck Methanisierung
683 end
684
685 % Berechnung KompressorArbeit. Sonderfall Membran
686 if(strcmp(compColInExcel,'AG') || strcmp(compColInExcel,'AH') || strcmp(compColInExcel
,'AX') || strcmp(compColInExcel,'BC'))
687
688 P2 = chainp4Compr;
689 P1 = compp;
690
691 if(strcmp(compColInExcel,'AX') || strcmp(compColInExcel,'BC'))
692 P2 = paramComp(compLoop).valueCompParam(find(strcmp(paramComp(compLoop).
txtCompParamInfo(:,1),'compp')) * 10^5;
693 end;
694
695 if (P2<P1)
696 msg = 'Fehler: Drücke überprüfen, Kompressor deaktivieren';
697 msg = [char(paramChain.txtChainName), ': ', msg, ' Berechnungen für diese
Prozesskette werden unterbrochen.'];
698 if startGuiFlag == true
699 BgaPtg2UpdateMessageBox(handles,'apend',msg);
700 else
701 disp(msg);
702 end
703 errorInLoopsFlag = true; return;
704 end
705
706 % if (strcmp(compColInExcel,'AX'))
707 % compp =
paramComp(compLoop).valueCompParam(find(strcmp(paramComp(compLoop).txtCompParamInfo(:,
1),'compp')));
708 % compp = compp * 10^5;
709 % print (compp, 'Membran')
710 % P1 = techChainpSystem;
711 % P2 = compp;
712 % end
713
714 compDeltap = max(P1,P2) - min(P1,P2);
715 compp = compp + compDeltap;
716
717
718
719 % Umrechnung von m3/h auf kg/s
720 MassFlow = compVolFlowPG * compDensity_PG/ 3600;
721
722 if Kappa_PG-1 == 0 || molarmass_PG == 0 || Kappa_PG == 0
723 msg = 'Variable "ComprWork" kann nicht berechnet werden. Division durch null.'
;
724 msg = [char(paramChain.txtChainName), ': ', msg, ' Berechnungen für diese
Prozesskette werden unterbrochen.'];
725 if startGuiFlag == true
726 BgaPtg2UpdateMessageBox(handles,'apend',msg);
727 else
728 disp(msg);
729 end
730 errorInLoopsFlag = true; return;
731
732 else
733 ComprWork = Kappa_PG/(Kappa_PG-1) * R / molarmass_PG * 1000 * compT * ((P2/P1
)^((Kappa_PG-1)/Kappa_PG)-1);

```

```

734     end;
735
736
737     % kW % Testwert ist noch unrealistisch, REchenfehler?
738     if chainComprEff == 0
739         msg = 'Variable "chainComprEff" ist gleich null. Division durch null.';
740         msg = [char(paramChain.txtChainName), ': ', msg, ' Berechnungen für diese
741             Prozesskette werden unterbrochen.'];
742         if startGuiFlag == true
743             BgaPtg2UpdateMessageBox(handles,'apend',msg);
744         else
745             disp(msg);
746         end
747         errorInLoopsFlag = true;     return;
748
749     else
750         compComprPower = ComprWork * MassFlow / chainComprEff / 1000;
751     end;
752
753     else
754         compComprPower = 0;
755         comppp = comppp - compDeltap;
756     end
757
758     techCompElecDemand = compElecSpec * techCompFLH * techChainMethPower + compComprPower
759     * techCompFLH;
760
761     % Als Ausgabe soll Kompressorleistung nur für Komponente Kompressor ausgegeben werden
762     und nicht bei Membranen. Strombedarf muss bei Membran dennoch berechnet werden. Nun
763     wird die Variable in jenem Fall wieder auf Null gesetzt.
764     if(strcmp(compColInExcel,'AX') || strcmp(compColInExcel,'BC'))
765         compComprPower = 0;
766     end
767
768     % Output für WSimu
769     techSimuCompVec = [compPower; techCompFLH; techCompElecDemand; techCompActCDemand;
770         compNutrientDemand; techCompHeatDemand; compHeatSurplus; compWaterDemand;
771         techCompOtherDemand];
772
773     % Weitergabe an nächste Komponente
774     techSimuCompValueVec = [compp; compT; compVolFlowBioM; compVolFlowSNG; compVolFlowCO2;
775         compVolFlowH2; compVolFlowH2S; compVolFlowH2O; ...
776         compVolFlowTraceGas; compVolFlowOther; compVolFlowPG;
777         compConcBioM; compConcSNG; compConcCO2; compConcH2; ...
778         compConcH2S; compConcH2O; compConcTraceGas; compConcOther;
779         compDensity_PG];
780
781     % Output für Results
782     techCompOutputVec = [techCompConcCH4; techCompConcCO2; techCompConcH2; techCompConcH2S
783         ; techCompConcH2O; techCompConcTraceGas; techCompConcOther; techCompVolFlowPG];
784
785     % Special Outputs für Results
786     techCompSpecOutputVec = [compConvCO2Ideal; compConvCO2Real; compComprPower;
787         compNutrientDemand; ...
788         compHeatHT_ideal; compHeatNT_ideal; compHeatHT_real;
789         compHeatNT_real; compWaterDemand];

```