

```

1
2 % Programm für die Vorberechnungen der technischen Simulation der Komponenten
3
4 function [techSimuPreCalcChainVec, techSimuPreCalcCompVec, techPreCalcChainOutput,
errorInLoopsFlag] = ...
5 BgaPtg2TechPreCalc(handles,paramComp,compLoop,paramChain)
6
7 global const;
8
9 % Damit ohne GUI gearbeitet werden kann.
10 global startGuiFlag;
11
12
13 %% techChainVolFlowBG ist der vorherige Wert
14 % if(strcmp(paramComp(compLoop).sheet,'BC'))
15 %     bb = 10;
16 % end
17
18
19 % Outputs definieren
20 techSimuPreCalcChainVec = ones(8,1)*-1;
21 techSimuPreCalcCompVec = ones(10,1)*-1;
22 techPreCalcChainOutput = ones(11,1)*-1;
23 errorInLoopsFlag = false;
24 %return;
25
26
27 % Parameter (Anfang) *****
28
29 compInflRate = paramComp(compLoop).valueCompParam(find(strcmp(paramComp
(compLoop).txtCompParamInfo(:,1),'compInflRate')));
30 if isempty(compInflRate)
31     msg = 'Parameter "compInflRate" bzw. dessen Wert ist nicht definiert.';
32     msg = [char(paramChain.txtChainName), ': ', msg, ' Berechnungen für diese
Prozesskette werden unterbrochen.'];
33     if startGuiFlag == true
34         BgaPtg2UpdateMessageBox(handles,'apend',msg);
35     else
36         disp(msg);
37     end
38     errorInLoopsFlag = true; return;
39 end
40
41 compMaintFactor = paramComp(compLoop).valueCompParam(find(strcmp(paramComp
(compLoop).txtCompParamInfo(:,1),'compMaintFactor')));
42 if isempty(compMaintFactor)
43     msg = 'Parameter "compMaintFactor" bzw. dessen Wert ist nicht definiert.';
44     msg = [char(paramChain.txtChainName), ': ', msg, ' Berechnungen für diese
Prozesskette werden unterbrochen.'];
45     if startGuiFlag == true
46         BgaPtg2UpdateMessageBox(handles,'apend',msg);
47     else
48         disp(msg);
49     end
50     errorInLoopsFlag = true; return;
51 end
52
53 compUpkeepFactor = paramComp(compLoop).valueCompParam(find(strcmp(paramComp
(compLoop).txtCompParamInfo(:,1),'compUpkeepFactor')));
54 if isempty(compUpkeepFactor)
55     msg = 'Parameter "compUpkeepFactor" bzw. dessen Wert ist nicht definiert.';
56     msg = [char(paramChain.txtChainName), ': ', msg, ' Berechnungen für diese
Prozesskette werden unterbrochen.'];
57     if startGuiFlag == true
58         BgaPtg2UpdateMessageBox(handles,'apend',msg);
59     else
60         disp(msg);
61     end
62     errorInLoopsFlag = true; return;
63 end
64
65 compElecPrice = paramComp(compLoop).valueCompParam(find(strcmp(paramComp
(compLoop).txtCompParamInfo(:,1),'compElecPrice')));

```

```

66  if isempty(compElecPrice)
67      msg = 'Parameter "compElecPrice" bzw. dessen Wert ist nicht definiert.';
68      msg = [char(paramChain.txtChainName), ': ', msg, ' Berechnungen für diese
        Prozesskette werden unterbrochen.'];
69      if startGuiFlag == true
70          BgaPtg2UpdateMessageBox(handles,'apend',msg);
71      else
72          disp(msg);
73      end
74      errorInLoopsFlag = true;      return;
75  end
76
77  compp      = paramComp(compLoop).valueCompParam(find(strcmp(paramComp
        (compLoop).txtCompParamInfo(:,1),'compp')));
78  if isempty(compp)
79      msg = 'Parameter "compp" bzw. dessen Wert ist nicht definiert.';
80      msg = [char(paramChain.txtChainName), ': ', msg, ' Berechnungen für diese
        Prozesskette werden unterbrochen.'];
81      if startGuiFlag == true
82          BgaPtg2UpdateMessageBox(handles,'apend',msg);
83      else
84          disp(msg);
85      end
86      errorInLoopsFlag = true;      return;
87  end
88
89  compT      = paramComp(compLoop).valueCompParam(find(strcmp(paramComp
        (compLoop).txtCompParamInfo(:,1),'compT')));
90  if isempty(compT)
91      msg = 'Parameter "compT" bzw. dessen Wert ist nicht definiert.';
92      msg = [char(paramChain.txtChainName), ': ', msg, ' Berechnungen für diese
        Prozesskette werden unterbrochen.'];
93      if startGuiFlag == true
94          BgaPtg2UpdateMessageBox(handles,'apend',msg);
95      else
96          disp(msg);
97      end
98      errorInLoopsFlag = true;      return;
99  end
100
101  compConcBioM      = paramComp(compLoop).valueCompParam(find(strcmp(paramComp
        (compLoop).txtCompParamInfo(:,1),'compConcBioM')));
102  if isempty(compConcBioM)
103      msg = 'Parameter "compConcBioM" bzw. dessen Wert ist nicht definiert.';
104      msg = [char(paramChain.txtChainName), ': ', msg, ' Berechnungen für diese
        Prozesskette werden unterbrochen.'];
105      if startGuiFlag == true
106          BgaPtg2UpdateMessageBox(handles,'apend',msg);
107      else
108          disp(msg);
109      end
110      errorInLoopsFlag = true;      return;
111  end
112
113  compConcSNG      = paramComp(compLoop).valueCompParam(find(strcmp(paramComp
        (compLoop).txtCompParamInfo(:,1),'compConcSNG')));
114  if isempty(compConcSNG)
115      msg = 'Parameter "compConcSNG" bzw. dessen Wert ist nicht definiert.';
116      msg = [char(paramChain.txtChainName), ': ', msg, ' Berechnungen für diese
        Prozesskette werden unterbrochen.'];
117      if startGuiFlag == true
118          BgaPtg2UpdateMessageBox(handles,'apend',msg);
119      else
120          disp(msg);
121      end
122      errorInLoopsFlag = true;      return;
123  end
124
125  compConcCO2      = paramComp(compLoop).valueCompParam(find(strcmp(paramComp
        (compLoop).txtCompParamInfo(:,1),'compConcCO2')));
126  if isempty(compConcCO2)
127      msg = 'Parameter "compConcCO2" bzw. dessen Wert ist nicht definiert.';
128      msg = [char(paramChain.txtChainName), ': ', msg, ' Berechnungen für diese

```

```

        Prozesskette werden unterbrochen.'];
129  if startGuiFlag == true
130      BgaPtg2UpdateMessageBox(handles,'apend',msg);
131  else
132      disp(msg);
133  end
134  errorInLoopsFlag = true;      return;
135  end
136
137  compConcH2                = paramComp(compLoop).valueCompParam(find(strcmp(paramComp
(compLoop).txtCompParamInfo(:,1),'compConcH2')));
138  if isempty(compConcH2)
139      msg = 'Parameter "compConcH2" bzw. dessen Wert ist nicht definiert.';
140      msg = [char(paramChain.txtChainName), ': ', msg, ' Berechnungen für diese
Prozesskette werden unterbrochen.'];
141  if startGuiFlag == true
142      BgaPtg2UpdateMessageBox(handles,'apend',msg);
143  else
144      disp(msg);
145  end
146  errorInLoopsFlag = true;      return;
147  end
148
149  compConcH2S              = paramComp(compLoop).valueCompParam(find(strcmp(paramComp
(compLoop).txtCompParamInfo(:,1),'compConcH2S')));
150  if isempty(compConcH2S)
151      msg = 'Parameter "compConcH2S" bzw. dessen Wert ist nicht definiert.';
152      msg = [char(paramChain.txtChainName), ': ', msg, ' Berechnungen für diese
Prozesskette werden unterbrochen.'];
153  if startGuiFlag == true
154      BgaPtg2UpdateMessageBox(handles,'apend',msg);
155  else
156      disp(msg);
157  end
158  errorInLoopsFlag = true;      return;
159  end
160
161  compConcH2O              = paramComp(compLoop).valueCompParam(find(strcmp(paramComp
(compLoop).txtCompParamInfo(:,1),'compConcH2O')));
162  if isempty(compConcH2O)
163      msg = 'Parameter "compConcH2O" bzw. dessen Wert ist nicht definiert.';
164      msg = [char(paramChain.txtChainName), ': ', msg, ' Berechnungen für diese
Prozesskette werden unterbrochen.'];
165  if startGuiFlag == true
166      BgaPtg2UpdateMessageBox(handles,'apend',msg);
167  else
168      disp(msg);
169  end
170  errorInLoopsFlag = true;      return;
171  end
172
173  compConcTraceGas        = paramComp(compLoop).valueCompParam(find(strcmp(paramComp
(compLoop).txtCompParamInfo(:,1),'compConcTraceGas')));
174  if isempty(compConcTraceGas)
175      msg = 'Parameter "compConcTraceGas" bzw. dessen Wert ist nicht definiert.';
176      msg = [char(paramChain.txtChainName), ': ', msg, ' Berechnungen für diese
Prozesskette werden unterbrochen.'];
177  if startGuiFlag == true
178      BgaPtg2UpdateMessageBox(handles,'apend',msg);
179  else
180      disp(msg);
181  end
182  errorInLoopsFlag = true;      return;
183  end
184
185  compConcOther           = paramComp(compLoop).valueCompParam(find(strcmp(paramComp
(compLoop).txtCompParamInfo(:,1),'compConcOther')));
186  if isempty(compConcOther)
187      msg = 'Parameter "compConcOther" bzw. dessen Wert ist nicht definiert.';
188      msg = [char(paramChain.txtChainName), ': ', msg, ' Berechnungen für diese
Prozesskette werden unterbrochen.'];
189  if startGuiFlag == true
190      BgaPtg2UpdateMessageBox(handles,'apend',msg);

```

```

191     else
192         disp(msg);
193     end
194     errorInLoopsFlag = true;     return;
195 end
196
197 % compDeltap                               =
paramComp(compLoop).valueCompParam(find(strcmp(paramComp(compLoop).txtCompParamInfo(:,
1), 'compDeltap')));
198 % if isempty(compDeltap)
199 %     msg = 'Parameter "compDeltap" bzw. dessen Wert ist nicht definiert.';
200 %     msg = [char(paramChain.txtChainName), ': ', msg, ' Berechnungen für diese
Prozesskette werden unterbrochen.'];
201 %     if startGuiFlag == true
202 %         BgaPtg2UpdateMessageBox(handles, 'apend', msg);
203 %     else
204 %         disp(msg);
205 %     end
206 %     errorInLoopsFlag = true;     return;
207 % end
208
209 compLeakage                               = paramComp(compLoop).valueCompParam(find(strcmp(paramComp
(compLoop).txtCompParamInfo(:,1), 'compLeakage')));
210 if isempty(compLeakage)
211     msg = 'Parameter "compLeakage" bzw. dessen Wert ist nicht definiert.';
212     msg = [char(paramChain.txtChainName), ': ', msg, ' Berechnungen für diese
Prozesskette werden unterbrochen.'];
213     if startGuiFlag == true
214         BgaPtg2UpdateMessageBox(handles, 'apend', msg);
215     else
216         disp(msg);
217     end
218     errorInLoopsFlag = true;     return;
219 end
220
221 compNutrientVolFlowFactor                = paramComp(compLoop).valueCompParam(find(strcmp(paramComp
(compLoop).txtCompParamInfo(:,1), 'compNutrientVolFlowFactor')));
222 if isempty(compNutrientVolFlowFactor)
223     msg = 'Parameter "compNutrientVolFlowFactor" bzw. dessen Wert ist nicht
definiert.';
224     msg = [char(paramChain.txtChainName), ': ', msg, ' Berechnungen für diese
Prozesskette werden unterbrochen.'];
225     if startGuiFlag == true
226         BgaPtg2UpdateMessageBox(handles, 'apend', msg);
227     else
228         disp(msg);
229     end
230     errorInLoopsFlag = true;     return;
231 end
232
233 compOtherFactor                          = paramComp(compLoop).valueCompParam(find(strcmp(paramComp
(compLoop).txtCompParamInfo(:,1), 'compOtherFactor')));
234 if isempty(compOtherFactor)
235     msg = 'Parameter "compOtherFactor" bzw. dessen Wert ist nicht definiert.';
236     msg = [char(paramChain.txtChainName), ': ', msg, ' Berechnungen für diese
Prozesskette werden unterbrochen.'];
237     if startGuiFlag == true
238         BgaPtg2UpdateMessageBox(handles, 'apend', msg);
239     else
240         disp(msg);
241     end
242     errorInLoopsFlag = true;     return;
243 end
244
245 compConvCO2Ideal                         = paramComp(compLoop).valueCompParam(find(strcmp(paramComp
(compLoop).txtCompParamInfo(:,1), 'compConvCO2Ideal')));
246 if isempty(compConvCO2Ideal)
247     msg = 'Parameter "compConvCO2Ideal" bzw. dessen Wert ist nicht definiert.';
248     msg = [char(paramChain.txtChainName), ': ', msg, ' Berechnungen für diese
Prozesskette werden unterbrochen.'];
249     if startGuiFlag == true
250         BgaPtg2UpdateMessageBox(handles, 'apend', msg);
251     else

```

```

252         disp(msg);
253     end
254     errorInLoopsFlag = true;         return;
255 end
256
257 compConvCO2Real = paramComp(compLoop).valueCompParam(find(strcmp(paramComp
(compLoop).txtCompParamInfo(:,1),'compConvCO2Real')));
258 if isempty(compConvCO2Real)
259     msg = 'Parameter "compConvCO2Real" bzw. dessen Wert ist nicht definiert.';
260     msg = [char(paramChain.txtChainName), ': ', msg, ' Berechnungen für diese
Prozesskette werden unterbrochen.'];
261     if startGuiFlag == true
262         BgaPtg2UpdateMessageBox(handles,'apend',msg);
263     else
264         disp(msg);
265     end
266     errorInLoopsFlag = true;         return;
267 end
268
269 % Chain-Parameter lesen
270 chainPowerCHP = paramChain.valueChainParam(find(strcmp(paramChain.
txtChainParamInfo(:,1),'chainPowerCHP')));
271 if isempty(chainPowerCHP)
272     msg = 'Parameter "chainPowerCHP" bzw. dessen Wert ist nicht definiert.';
273     msg = [char(paramChain.txtChainName), ': ', msg, ' Berechnungen für diese
Prozesskette werden unterbrochen.'];
274     if startGuiFlag == true
275         BgaPtg2UpdateMessageBox(handles,'apend',msg);
276     else
277         disp(msg);
278     end
279     errorInLoopsFlag = true;         return;
280 end
281
282 chainEffCHP = paramChain.valueChainParam(find(strcmp(paramChain.
txtChainParamInfo(:,1),'chainEffCHP')));
283 if isempty(chainEffCHP)
284     msg = 'Parameter "chainEffCHP" bzw. dessen Wert ist nicht definiert.';
285     msg = [char(paramChain.txtChainName), ': ', msg, ' Berechnungen für diese
Prozesskette werden unterbrochen.'];
286     if startGuiFlag == true
287         BgaPtg2UpdateMessageBox(handles,'apend',msg);
288     else
289         disp(msg);
290     end
291     errorInLoopsFlag = true;         return;
292 end
293
294 chainEffMeth = paramChain.valueChainParam(find(strcmp(paramChain.
txtChainParamInfo(:,1),'chainEffMeth')));
295 if isempty(chainEffMeth)
296     msg = 'Parameter "chainEffMeth" bzw. dessen Wert ist nicht definiert.';
297     msg = [char(paramChain.txtChainName), ': ', msg, ' Berechnungen für diese
Prozesskette werden unterbrochen.'];
298     if startGuiFlag == true
299         BgaPtg2UpdateMessageBox(handles,'apend',msg);
300     else
301         disp(msg);
302     end
303     errorInLoopsFlag = true;         return;
304 end
305
306 chainOverscalingMeth = paramChain.valueChainParam(find(strcmp(paramChain.
txtChainParamInfo(:,1),'chainOverscalingMeth')));
307 if isempty(chainOverscalingMeth)
308     msg = 'Parameter "chainOverscalingMeth" bzw. dessen Wert ist nicht definiert.';
309     msg = [char(paramChain.txtChainName), ': ', msg, ' Berechnungen für diese
Prozesskette werden unterbrochen.'];
310     if startGuiFlag == true
311         BgaPtg2UpdateMessageBox(handles,'apend',msg);
312     else
313         disp(msg);
314     end

```

```

315     errorInLoopsFlag = true;         return;
316 end
317
318 chainEffEly = paramChain.valueChainParam(find(strcmp(paramChain.
txtChainParamInfo(:,1),'chainEffEly')));
319 if isempty(chainEffEly)
320     msg = 'Parameter "chainEffEly" bzw. dessen Wert ist nicht definiert.';
321     msg = [char(paramChain.txtChainName), ': ', msg, ' Berechnungen für diese
Prozesskette werden unterbrochen.'];
322     if startGuiFlag == true
323         BgaPtg2UpdateMessageBox(handles,'apend',msg);
324     else
325         disp(msg);
326     end
327     errorInLoopsFlag = true;         return;
328 end
329
330 techChainVolFlowBG = paramChain.valueChainParam(find(strcmp(paramChain.
txtChainParamInfo(:,1),'chainVolFlowBG')));
331 if isempty(techChainVolFlowBG)
332     msg = 'Parameter "chainVolFlowBG" bzw. dessen Wert ist nicht definiert.';
333     msg = [char(paramChain.txtChainName), ': ', msg, ' Berechnungen für diese
Prozesskette werden unterbrochen.'];
334     if startGuiFlag == true
335         BgaPtg2UpdateMessageBox(handles,'apend',msg);
336     else
337         disp(msg);
338     end
339     errorInLoopsFlag = true;         return;
340 end
341
342 chainOverscalingEly = paramChain.valueChainParam(find(strcmp(paramChain.
txtChainParamInfo(:,1),'chainOverscalingEly')));
343 if isempty(chainOverscalingEly)
344     msg = 'Parameter "chainOverscalingEly" bzw. dessen Wert ist nicht definiert.';
345     msg = [char(paramChain.txtChainName), ': ', msg, ' Berechnungen für diese
Prozesskette werden unterbrochen.'];
346     if startGuiFlag == true
347         BgaPtg2UpdateMessageBox(handles,'apend',msg);
348     else
349         disp(msg);
350     end
351     errorInLoopsFlag = true;         return;
352 end
353
354 %% Parameter (Ende) *****
355
356
357 % Konstanten lesen
358 HHV_CH4 = const.HHV_CH4;
359 density_H2 = const.density_H2;
360 convDes = const.convDes;
361 HHV_H2 = const.HHV_H2;
362 stoichiometryDes = const.stoichiometryDes;
363
364 % Umwandlungen in paramComp
365 % compInflRate = compInflRate/100;
366 % compMaintFactor = compMaintFactor/100;
367 % compUpkeepFactor = compUpkeepFactor/100;
368 % compElecPrice = compElecPrice/100;
369 comppp = comppp*10^5; % bar -> Pa
370 compT = compT + 273.15; % °C -> K
371 compConcBioM = compConcBioM/100;
372 compConcSNG = compConcSNG/100;
373 compConcCO2 = compConcCO2/100;
374 compConcH2 = compConcH2/100;
375 compConcH2S = compConcH2S/(10^6);
376 compConcH2O = compConcH2O/100;
377 compConcTraceGas = compConcTraceGas/(10^6);
378 compConcOther = compConcOther/100;
379 % compDeltaVolFlowBioM = compDeltaVolFlowBioM/100;
380 % compDeltaVolFlowSNG = compDeltaVolFlowSNG/100;
381 % compDeltaVolFlowCO2 = compDeltaVolFlowCO2/100;

```

```

382 % compDeltaVolFlowH2           = compDeltaVolFlowH2/100;
383 % compDeltaVolFlowH2S          = compDeltaVolFlowH2S/100;
384 % compDeltaVolFlowH2O          = compDeltaVolFlowH2O/100;
385 % compDeltaVolFlowTraceGas     = compDeltaVolFlowTraceGas/100;
386 % compDeltaVolFlowOther        = compDeltaVolFlowOther/100;
387 compLeakage                    = compLeakage/100;
388 compOtherFactor                = compOtherFactor/100;
389 compNutrientVolFlowFactor      = compNutrientVolFlowFactor * (10^3);
390 compConvCO2Ideal               = compConvCO2Ideal/100;
391 compConvCO2Real                = compConvCO2Real/100;
392 % compDeltap                   = compDeltap * 10^5;
393
394 % Umwandlungen in paramChain
395 % chainDiscRate                = chainDiscRate/100;
396 % chainElecPrice               = chainElecPrice/100;
397 % chainOperationFactor         = chainOperationFactor/100;
398 % chainInsurFactor             = chainInsurFactor/100;
399 % chainRenatFactor             = chainRenatFactor/100;
400 % chainTechInflRate            = chainTechInflRate/100;
401 % chainBuildInflRate           = chainBuildInflRate/100;
402 % chainInflRate                = chainInflRate/100;
403 % chainWaterPrice              = chainWaterPrice;
404 % chainBGPurchasePrice         = chainBGPurchasePrice/100;
405 % chainBMSellingPrice          = chainBMSellingPrice/100;
406 % chainHeatPurchasePrice       = chainHeatPurchasePrice/100;
407 % chainHeatSellingPrice        = chainHeatSellingPrice/100;
408 % chainO2SellingPrice          = chainO2SellingPrice/100;
409 % chainComprEff                = chainComprEff/100;
410 chainEffMeth                   = chainEffMeth/100;
411 chainEffCHP                    = chainEffCHP/100;
412 chainEffEly                    = chainEffEly/100;
413 % chainHeatUseHT               = chainHeatUseHT/100;
414 % chainHeatUseNT               = chainHeatUseNT/100;
415
416
417 % Falls keine compVolFlowBG gegeben, berechnung über Leistung BHKW
418 if techChainVolFlowBG == 0
419     if chainEffCHP == 0 || HHV_CH4 == 0 || compConcBioM+compConcSNG == 0
420         msg = 'Variable "techChainVolFlowBG" kann nicht berechnet werden. Division
421             durch null.';
422         msg = [char(paramChain.txtChainName), ': ', msg, ' Berechnungen für diese
423             Prozesskette werden unterbrochen.'];
424         if startGuiFlag == true
425             BgaPtg2UpdateMessageBox(handles,'apend',msg);
426         else
427             disp(msg);
428         end
429         errorInLoopsFlag = true;         return;
430     else
431         techChainVolFlowBG = chainPowerCHP/chainEffCHP/HHV_CH4/(compConcBioM+
432             compConcSNG);
433     end;
434
435 compVolFlowBioM                 = compConcBioM * techChainVolFlowBG;
436 compVolFlowSNG                  = compConcSNG * techChainVolFlowBG;
437 compVolFlowCO2                  = compConcCO2 * techChainVolFlowBG;
438 compVolFlowH2                   = compConcH2 * techChainVolFlowBG;
439 compVolFlowH2S                  = compConcH2S * techChainVolFlowBG;
440 compVolFlowH2O                  = compConcH2O * techChainVolFlowBG;
441 compVolFlowTraceGas             = compConcTraceGas * techChainVolFlowBG;
442 compVolFlowOther                 = compConcOther * techChainVolFlowBG;
443
444 % sÜberprüfung ob Biogas Konzentrationen 100 % ergeben
445 ConcTotal = compConcBioM + compConcSNG + compConcCO2 + compConcH2 + compConcH2S +
446             compConcH2O + compConcTraceGas + compConcOther;
447 %if ConcTotal ~= 1
448     eps = 0.0000001;
449     if 1+eps < ConcTotal && ConcTotal < 1-eps
450         msg = 'Fehler! Die Gesamtkonzentration entspricht nicht 100%!';
451         msg = [char(paramChain.txtChainName), ': ', msg, ' Berechnungen für diese
452             Prozesskette werden unterbrochen.'];

```

```

450     if startGuiFlag == true
451         BgaPtg2UpdateMessageBox(handles,'apend',msg);
452     else
453         disp(msg);
454     end
455
456     msg = [char(paramChain.txtChainName), ': Gesamt-Konzentration = ',num2str(
457         ConcTotal*100),' %'];
458     if startGuiFlag == true
459         BgaPtg2UpdateMessageBox(handles,'apend',msg);
460     else
461         disp(msg);
462     end
463     errorInLoopsFlag = true;     return;
464
465
466
467 % Volumenströme CO2, H2
468 chainVolFlowCO2     = compVolFlowCO2;
469 chainVolFlowH2      = chainVolFlowCO2 * stoichiometryDes; % Auslegung H2-Strom
470 chainVolFlowH2kg    = chainVolFlowH2 * density_H2;
471
472 % Auslegungsleistungen Methanisierung, Elektrolyse. Real anders, wenn auf
473 realFlowRate ausgelegt
474 if chainEffMeth == 0 || convDes == 0 || compConcBioM+compConcSNG == 0
475     msg = 'Variable "techChainMethPower" kann nicht berechnet werden. Division durch
476     null.';
477     msg = [char(paramChain.txtChainName), ': ', msg, ' Berechnungen für diese
478     Prozesskette werden unterbrochen.'];
479     if startGuiFlag == true
480         BgaPtg2UpdateMessageBox(handles,'apend',msg);
481     else
482         disp(msg);
483     end
484     errorInLoopsFlag = true;     return;
485
486
487
488 else
489     techChainMethPower = chainVolFlowCO2 * HHV_CH4 / chainEffMeth/convDes *
490     chainOverscalingMeth /1000;
491
492
493
494
495
496
497
498
499
500
501
502
503
504 if chainEffEly == 0
505     msg = 'Variable "techChainElyPower" kann nicht berechnet werden. Division durch
506     null.';
507     msg = [char(paramChain.txtChainName), ': ', msg, ' Berechnungen für diese
508     Prozesskette werden unterbrochen.'];
509     if startGuiFlag == true
510         BgaPtg2UpdateMessageBox(handles,'apend',msg);
511     else
512         disp(msg);
513     end
514     errorInLoopsFlag = true;     return;
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000

```



```

512         techChainpSystem = paramComp(loop).valueCompParam(find(strcmp(paramComp(
513         loop).txtCompParamInfo(:,1),'compp')) * 10^5; % Druck Methanisierung
514         break; % AK bis AX schließen sich gegenseitig aus
515     end
516 end
517
518 for loop = 1:size(paramComp,1)
519     compColInExcel = paramComp(loop).sheet;
520     if( strcmp(compColInExcel,'AI') || strcmp(compColInExcel,'AJ'))% ||
521     strcmp(compColInExcel,'BB'))
522         chainp4Compr = techChainpSystem + paramComp(loop).valueCompParam(find(strcmp(
523         paramComp(loop).txtCompParamInfo(:,1),'compDeltap')) * 10^5; % Druck
524         Methanisierung
525         %break; % AI, AJ und BB schließen sich nicht gegenseitig aus
526     else
527         chainp4Compr = techChainpSystem;
528     end
529 end
530
531 % Outputs und Rekonvertierung
532 techChainConcInputCH4 = (compConcBioM + compConcSNG) *100;
533 techChainConcInputCO2 = compConcCO2 * 100;
534 techChainConcInputH2 = compConcH2 * 100;
535 techChainConcInputH2S = compConcH2S * 10^6;
536 techChainConcInputH2O = compConcH2O * 100;
537 techChainConcInputOther = compConcOther * 100;
538 techChainConcInputTraceGas = compConcTraceGas * 10^6;
539
540 % Outputs zuweisen
541 % Chain Parameter, die weiter genutzt werden und teilweise in Outputs gehen
542 techSimuPreCalcChainVec = [techChainVolFlowBG; chainVolFlowCO2; chainVolFlowH2;
543 chainVolFlowH2kg; ...
544                             techChainMethPower; techChainElyPower; techChainpSystem;
545                             chainp4Compr];
546
547 % Paramter, die an erste Komponente gegeben werden
548 techSimuPreCalcCompVec = [compp; compT; compVolFlowBioM; compVolFlowSNG;
549 compVolFlowCO2; compVolFlowH2; ...
550                             compVolFlowH2S; compVolFlowH2O; compVolFlowTraceGas;
551                             compVolFlowOther];
552
553 % Parameter, die nur zu Outputs gehen
554 techPreCalcChainOutput = [techChainVolFlowBG; techChainConcInputCH4;
555 techChainConcInputCO2; techChainConcInputH2; ...
556                             techChainConcInputH2S; techChainConcInputH2O;
557                             techChainConcInputTraceGas; techChainConcInputOther;
558                             techChainMethPower; ...
559                             techChainElyPower; techChainpSystem];

```