

```

1
2 % Programm für die technische Simulation der Prozesskette
3 % Momentan nur Zuweisung, diese Werte müssen aber irgendwann in dieser
4 % Funktion berechnet werden
5
6 function [techSimuChainVec, techSimuOutputVec, errorInLoopsFlag] = ...
7     BgaPtg2TechSimuChain(handles,paramChain,paramComp,techSimuCompVec,
8         techSimuCompValueVecAll,techPreCalcChainOutputHelpAll,techCompSpecialOutput,
9         techSimuPreCalcChainVec)
10
11 global const;
12
13 % Damit ohne GUI gearbeitet werden kann.
14 global startGuiFlag;
15
16 % Outputs definieren
17 techSimuChainVec = ones(7,1)*-1;
18 techSimuOutputVec = ones(41,1)*-1;
19 errorInLoopsFlag = false;
20
21 %% Parameter (Anfang) *****
22 % Projektlaufzeit [a]
23 chainPeriod = paramChain.valueChainParam(find(strcmp(paramChain.txtChainParamInfo(:,1),
24     'chainPeriod')));
25 % if strcmp(char(paramChain.txtChainName),'Prozesskette 2')
26 %     chainPeriod = [];
27 % end
28
29 if isempty(chainPeriod)
30     msg = 'Parameter "chainPeriod" bzw. dessen Wert ist nicht definiert.';
31     msg = [char(paramChain.txtChainName), ': ', msg, ' Berechnungen für diese
32     Prozesskette werden unterbrochen.'];
33     if startGuiFlag == true
34         BgaPtg2UpdateMessageBox(handles,'apend',msg);
35     else
36         disp(msg);
37     end
38     % errorldg(msg);
39     errorInLoopsFlag = true;    return;
40 end
41
42 % Kalkulationszinsatz, discount rate [%]
43 chainDiscRate = paramChain.valueChainParam(find(strcmp(paramChain.txtChainParamInfo(:,
44     1),'chainDiscRate')));
45 if isempty(chainDiscRate)
46     msg = 'Parameter "chainDiscRate" bzw. dessen Wert ist nicht definiert.';
47     msg = [char(paramChain.txtChainName), ': ', msg, ' Berechnungen für diese
48     Prozesskette werden unterbrochen.'];
49     if startGuiFlag == true
50         BgaPtg2UpdateMessageBox(handles,'apend',msg);
51     else
52         disp(msg);
53     end
54     errorInLoopsFlag = true;    return;
55 end
56
57 % Aktivkohlepreis [€/kg] Aktivkohle
58 %chainActCPrice =
59 paramChain.valueChainParam(find(strcmp(paramChain.txtChainParamInfo(:,1),'chainActCPri
60 ce')));
61
62 % Strompreis [ct/kWh]
63 chainElecPrice = paramChain.valueChainParam(find(strcmp(paramChain.txtChainParamInfo
64     (:,1),'chainElecPrice')));
65 if isempty(chainElecPrice)
66     msg = 'Parameter "chainElecPrice" bzw. dessen Wert ist nicht definiert.';
67     msg = [char(paramChain.txtChainName), ': ', msg, ' Berechnungen für diese
68     Prozesskette werden unterbrochen.'];
69     if startGuiFlag == true

```

```

64         BgaPtg2UpdateMessageBox(handles,'apend',msg);
65     else
66         disp(msg);
67     end
68     errorInLoopsFlag = true;     return;
69 end
70
71 % Biogas Einkaufspreis [ct/kWh]
72 chainBGPurchasePrice = paramChain.valueChainParam(find(strcmp(paramChain.
txtChainParamInfo(:,1),'chainBGPurchasePrice')));
73 if isempty(chainBGPurchasePrice)
74     msg = 'Parameter "chainBGPurchasePrice" 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
85 % Biogas Verkaufspreis, muss negativ sein
86 % Um EingabeFehler zu vermeiden, z.B. Eingabe von Neg. Werten, wird AbsolutWert geholt
87 % Ist das kompatibel mit Matlab oder darf das hier nicht sein?
88 % : wird negativ in exceldatei angegeben oder positiv?
89 % index scheint falsch zu sein. korrigiert
90 % Eingabe muss positiv sein [ct/kWh]
91 chainBMSellingPrice = paramChain.valueChainParam(find(strcmp(paramChain.
txtChainParamInfo(:,1),'chainBMSellingPrice')));
92 if isempty(chainBMSellingPrice)
93     msg = 'Parameter "chainBMSellingPrice" bzw. dessen Wert ist nicht definiert.';
94     msg = [char(paramChain.txtChainName), ': ', msg, ' Berechnungen für diese
Prozesskette werden unterbrochen.'];
95     if startGuiFlag == true
96         BgaPtg2UpdateMessageBox(handles,'apend',msg);
97     else
98         disp(msg);
99     end
100    errorInLoopsFlag = true;     return;
101 end
102
103 % Wärme Einkaufspreis in [ct/kWh]
104 chainHeatPurchasePrice = paramChain.valueChainParam(find(strcmp(paramChain.
txtChainParamInfo(:,1),'chainHeatPurchasePrice')));
105 if isempty(chainHeatPurchasePrice)
106     msg = 'Parameter "chainHeatPurchasePrice" bzw. dessen Wert ist nicht definiert.';
107     msg = [char(paramChain.txtChainName), ': ', msg, ' Berechnungen für diese
Prozesskette werden unterbrochen.'];
108     if startGuiFlag == true
109         BgaPtg2UpdateMessageBox(handles,'apend',msg);
110     else
111         disp(msg);
112     end
113     errorInLoopsFlag = true;     return;
114 end
115
116 %
117 % Wärme Verkaufspreis in €/kWh [ct/kWh]
118 chainHeatSellingPrice = paramChain.valueChainParam(find(strcmp(paramChain.
txtChainParamInfo(:,1),'chainHeatSellingPrice')));
119 if isempty(chainHeatSellingPrice)
120     msg = 'Parameter "chainHeatSellingPrice" bzw. dessen Wert ist nicht definiert.';
121     msg = [char(paramChain.txtChainName), ': ', msg, ' Berechnungen für diese
Prozesskette werden unterbrochen.'];
122     if startGuiFlag == true
123         BgaPtg2UpdateMessageBox(handles,'apend',msg);
124     else
125         disp(msg);
126     end
127     errorInLoopsFlag = true;     return;
128 end

```

```

129
130 % Investkosten in Bauliche Anlagen, Container, Integration
131 % Fixwert [€]
132 chainBuildInvest = paramChain.valueChainParam(find(strcmp(paramChain.txtChainParamInfo
(:,1),'chainBuildInvest')));
133 if isempty(chainBuildInvest)
134     msg = 'Parameter "chainBuildInvest" 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 % Investkosten Mess-/RegelTech, Elektroinstallation, Sicherheitstechnik, Leitwarte
145 % Fixwert [€]
146 chainTechInvest = paramChain.valueChainParam(find(strcmp(paramChain.txtChainParamInfo
(:,1),'chainTechInvest')));
147 if isempty(chainTechInvest)
148     msg = 'Parameter "chainTechInvest" bzw. dessen Wert ist nicht definiert.';
149     msg = [char(paramChain.txtChainName), ': ', msg, ' Berechnungen für diese
Prozesskette werden unterbrochen.'];
150     if startGuiFlag == true
151         BgaPtg2UpdateMessageBox(handles,'apend',msg);
152     else
153         disp(msg);
154     end
155     errorInLoopsFlag = true;     return;
156 end
157
158 % Einmalige Kosten für Planung, Genehmigung, Fixwert [€]
159 chainPlanningApproval = paramChain.valueChainParam(find(strcmp(paramChain.
txtChainParamInfo(:,1),'chainPlanningApproval')));
160 if isempty(chainPlanningApproval)
161     msg = 'Parameter "chainPlanningApproval" bzw. dessen Wert ist nicht definiert.';
162     msg = [char(paramChain.txtChainName), ': ', msg, ' Berechnungen für diese
Prozesskette werden unterbrochen.'];
163     if startGuiFlag == true
164         BgaPtg2UpdateMessageBox(handles,'apend',msg);
165     else
166         disp(msg);
167     end
168     errorInLoopsFlag = true;     return;
169 end
170
171 % Betriebskosten-Faktor für die Technik [%]
172 chainOperationFactor = paramChain.valueChainParam(find(strcmp(paramChain.
txtChainParamInfo(:,1),'chainOperationFactor')));
173 if isempty(chainOperationFactor)
174     msg = 'Parameter "chainOperationFactor" bzw. dessen Wert ist nicht definiert.';
175     msg = [char(paramChain.txtChainName), ': ', msg, ' Berechnungen für diese
Prozesskette werden unterbrochen.'];
176     if startGuiFlag == true
177         BgaPtg2UpdateMessageBox(handles,'apend',msg);
178     else
179         disp(msg);
180     end
181     errorInLoopsFlag = true;     return;
182 end
183
184 % Versicherungsfaktor der Gesamtanlage [%] % of CAPEX
185 chainInsurFactor = paramChain.valueChainParam(find(strcmp(paramChain.txtChainParamInfo
(:,1),'chainInsurFactor')));
186 if isempty(chainInsurFactor)
187     msg = 'Parameter "chainInsurFactor" 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 % Rückbau, Rücklagenfaktor der Gesamtanlage, en: renaturation reserve [%] #of CAPEX
198 chainRenatFactor = paramChain.valueChainParam(find(strcmp(paramChain.txtChainParamInfo
199 (:,1),'chainRenatFactor')));
200 if isempty(chainRenatFactor)
201     msg = 'Parameter "chainRenatFactor" bzw. dessen Wert ist nicht definiert.';
202     msg = [char(paramChain.txtChainName), ': ', msg, ' Berechnungen für diese
203     Prozesskette werden unterbrochen.'];
204     if startGuiFlag == true
205         BgaPtg2UpdateMessageBox(handles,'apend',msg);
206     else
207         disp(msg);
208     end
209     errorInLoopsFlag = true;         return;
210 end
211
212 % Personalkosten der Anlage, Fixwert €
213 chainCostPers = paramChain.valueChainParam(find(strcmp(paramChain.txtChainParamInfo(:,
214 1),'chainCostPers')));
215 if isempty(chainCostPers)
216     msg = 'Parameter "chainCostPers" bzw. dessen Wert ist nicht definiert.';
217     msg = [char(paramChain.txtChainName), ': ', msg, ' Berechnungen für diese
218     Prozesskette werden unterbrochen.'];
219     if startGuiFlag == true
220         BgaPtg2UpdateMessageBox(handles,'apend',msg);
221     else
222         disp(msg);
223     end
224     errorInLoopsFlag = true;         return;
225 end
226
227 % Sonstige Kosten, fix €
228 chainCostOther = paramChain.valueChainParam(find(strcmp(paramChain.txtChainParamInfo
229 (:,1),'chainCostOther')));
230 if isempty(chainCostOther)
231     msg = 'Parameter "chainCostOther" bzw. dessen Wert ist nicht definiert.';
232     msg = [char(paramChain.txtChainName), ': ', msg, ' Berechnungen für diese
233     Prozesskette werden unterbrochen.'];
234     if startGuiFlag == true
235         BgaPtg2UpdateMessageBox(handles,'apend',msg);
236     else
237         disp(msg);
238     end
239     errorInLoopsFlag = true;         return;
240 end
241
242 % Werte für ersetzbare technische Bauteile auf Prozesskettenebene
243 chainTechInflRate = paramChain.valueChainParam(find(strcmp(paramChain.
244 txtChainParamInfo(:,1),'chainTechInflRate')));
245 if isempty(chainTechInflRate)
246     msg = 'Parameter "chainTechInflRate" bzw. dessen Wert ist nicht definiert.';
247     msg = [char(paramChain.txtChainName), ': ', msg, ' Berechnungen für diese
248     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 chainTechLifetimeHours = paramChain.valueChainParam(find(strcmp(paramChain.
258 txtChainParamInfo(:,1),'chainTechLifetimeHours')));
259 if isempty(chainTechLifetimeHours)
260     msg = 'Parameter "chainTechLifetimeHours" bzw. dessen Wert ist nicht definiert.';
261     msg = [char(paramChain.txtChainName), ': ', msg, ' Berechnungen für diese
262     Prozesskette werden unterbrochen.'];
263     if startGuiFlag == true
264         BgaPtg2UpdateMessageBox(handles,'apend',msg);

```

```

255     else
256         disp(msg);
257     end
258     errorInLoopsFlag = true;     return;
259 end
260
261 % Werte für ersetzbare bauliche Anlagen auf Prozesskettenebene
262 chainBuildInflRate = paramChain.valueChainParam(find(strcmp(paramChain.
txtChainParamInfo(:,1),'chainBuildInflRate')));
263 if isempty(chainBuildInflRate)
264     msg = 'Parameter "chainBuildInflRate" bzw. dessen Wert ist nicht definiert.';
265     msg = [char(paramChain.txtChainName), ': ', msg, ' Berechnungen für diese
Prozesskette werden unterbrochen.'];
266     if startGuiFlag == true
267         BgaPtg2UpdateMessageBox(handles,'apend',msg);
268     else
269         disp(msg);
270     end
271     errorInLoopsFlag = true;     return;
272 end
273
274 chainBuildLifetimeHours = paramChain.valueChainParam(find(strcmp(paramChain.
txtChainParamInfo(:,1),'chainBuildLifetimeHours')));
275 if isempty(chainBuildLifetimeHours)
276     msg = 'Parameter "chainBuildLifetimeHours" bzw. dessen Wert ist nicht definiert.';
277     msg = [char(paramChain.txtChainName), ': ', msg, ' Berechnungen für diese
Prozesskette werden unterbrochen.'];
278     if startGuiFlag == true
279         BgaPtg2UpdateMessageBox(handles,'apend',msg);
280     else
281         disp(msg);
282     end
283     errorInLoopsFlag = true;     return;
284 end
285
286 % chainInflRate
287 chainInflRate = paramChain.valueChainParam(find(strcmp(paramChain.txtChainParamInfo(:,
1),'chainInflRate')));
288 if isempty(chainInflRate)
289     msg = 'Parameter "chainInflRate" bzw. dessen Wert ist nicht definiert.';
290     msg = [char(paramChain.txtChainName), ': ', msg, ' Berechnungen für diese
Prozesskette werden unterbrochen.'];
291     if startGuiFlag == true
292         BgaPtg2UpdateMessageBox(handles,'apend',msg);
293     else
294         disp(msg);
295     end
296     errorInLoopsFlag = true;     return;
297 end
298
299 % Wasserpreis
300 %chainWaterPrice =
paramChain.valueChainParam(find(strcmp(paramChain.txtChainParamInfo(:,1),'chainWaterPr
ice')));
301
302 % Preis für Nährmedien
303 %chainNutrientPrice =
paramChain.valueChainParam(find(strcmp(paramChain.txtChainParamInfo(:,1),'chainNutrien
tPrice')));
304
305 % Aktivkohlepreis [€/kg] Aktivkohle
306 % chainActCPrice =
paramChain.valueChainParam(find(strcmp(paramChain.txtChainParamInfo(:,1),'chainActCPri
ce')));
307
308
309 % Wasserpreis
310 chainWaterPrice = paramChain.valueChainParam(find(strcmp(paramChain.
txtChainParamInfo(:,1),'chainWaterPrice')));
311 if isempty(chainWaterPrice)
312     msg = 'Parameter "chainWaterPrice" bzw. dessen Wert ist nicht definiert.';
313     msg = [char(paramChain.txtChainName), ': ', msg, ' Berechnungen für diese
Prozesskette werden unterbrochen.'];

```

```

314     if startGuiFlag == true
315         BgaPtg2UpdateMessageBox(handles,'apend',msg);
316     else
317         disp(msg);
318     end
319     errorInLoopsFlag = true;     return;
320 end
321
322 % Preis für Nährmedien
323 %chainNutrientPrice           =
paramChain.valueChainParam(find(strcmp(paramChain.txtChainParamInfo(:,1),'chainNutrientPrice')));
324
325 % Verkaufserlös Sauerstoff
326 % chainO2Revenue             =
paramChain.valueChainParam(find(strcmp(paramChain.txtChainParamInfo(:,1),'chainO2Revenue')));
327 % if isempty(chainO2Revenue)
328 %     msg = 'Parameter "chainO2Revenue" bzw. dessen Wert ist nicht definiert.';
329 %     msg = [char(paramChain.txtChainName), ': ', msg, ' Berechnungen für diese
Prozesskette werden unterbrochen.'];
330 %     if startGuiFlag == true
331 %         BgaPtg2UpdateMessageBox(handles,'apend',msg);
332 %     else
333 %         disp(msg);
334 %     end
335 %     errorInLoopsFlag = true;     return;
336 % end
337
338
339 % Personalkosten für Ely, benötigt für Korrektur LCOH kein Output aus TSimu
340 % chainElyPers                =
paramChain.valueChainParam(find(strcmp(paramChain.txtChainParamInfo(:,1),'chainElyPers')));
341
342 % Ab Hier Technische Parameter, Outputs aus TSimu
343 techChainFLH                  = paramChain.valueChainParam(find(strcmp(paramChain.
txtChainParamInfo(:,1),'chainFLH'))); % Nachher anpassen
344 if isempty(techChainFLH)
345     msg = 'Parameter "chainFLH" bzw. dessen Wert ist nicht definiert.';
346     msg = [char(paramChain.txtChainName), ': ', msg, ' Berechnungen für diese
Prozesskette werden unterbrochen.'];
347     if startGuiFlag == true
348         BgaPtg2UpdateMessageBox(handles,'apend',msg);
349     else
350         disp(msg);
351     end
352     errorInLoopsFlag = true;     return;
353 end
354
355 chainElecSpecTech             = paramChain.valueChainParam(find(strcmp(paramChain.
txtChainParamInfo(:,1),'chainElecSpecTech')));
356 if isempty(chainElecSpecTech)
357     msg = 'Parameter "chainElecSpecTech" bzw. dessen Wert ist nicht definiert.';
358     msg = [char(paramChain.txtChainName), ': ', msg, ' Berechnungen für diese
Prozesskette werden unterbrochen.'];
359     if startGuiFlag == true
360         BgaPtg2UpdateMessageBox(handles,'apend',msg);
361     else
362         disp(msg);
363     end
364     errorInLoopsFlag = true;     return;
365 end
366
367 chainHeatSpecTech             = paramChain.valueChainParam(find(strcmp(paramChain.
txtChainParamInfo(:,1),'chainHeatSpecTech')));
368 if isempty(chainHeatSpecTech)
369     msg = 'Parameter "chainHeatSpecTech" bzw. dessen Wert ist nicht definiert.';
370     msg = [char(paramChain.txtChainName), ': ', msg, ' Berechnungen für diese
Prozesskette werden unterbrochen.'];
371     if startGuiFlag == true
372         BgaPtg2UpdateMessageBox(handles,'apend',msg);
373     else

```

```

374         disp(msg);
375     end
376     errorInLoopsFlag = true;     return;
377 end
378
379 chainElecSpecBuild           = paramChain.valueChainParam(find(strcmp(paramChain.
txtChainParamInfo(:,1),'chainElecSpecBuild')));
380 if isempty(chainElecSpecBuild)
381     msg = 'Parameter "chainElecSpecBuild" bzw. dessen Wert ist nicht definiert.';
382     msg = [char(paramChain.txtChainName), ': ', msg, ' Berechnungen für diese
Prozesskette werden unterbrochen.'];
383     if startGuiFlag == true
384         BgaPtg2UpdateMessageBox(handles,'apend',msg);
385     else
386         disp(msg);
387     end
388     errorInLoopsFlag = true;     return;
389 end
390
391 chainHeatSpecBuild           = paramChain.valueChainParam(find(strcmp(paramChain.
txtChainParamInfo(:,1),'chainHeatSpecBuild')));
392 if isempty(chainHeatSpecBuild)
393     msg = 'Parameter "chainHeatSpecBuild" bzw. dessen Wert ist nicht definiert.';
394     msg = [char(paramChain.txtChainName), ': ', msg, ' Berechnungen für diese
Prozesskette werden unterbrochen.'];
395     if startGuiFlag == true
396         BgaPtg2UpdateMessageBox(handles,'apend',msg);
397     else
398         disp(msg);
399     end
400     errorInLoopsFlag = true;     return;
401 end
402
403 chainComprEff                = paramChain.valueChainParam(find(strcmp(paramChain.
txtChainParamInfo(:,1),'chainComprEff')));
404 if isempty(chainComprEff)
405     msg = 'Parameter "chainComprEff" bzw. dessen Wert ist nicht definiert.';
406     msg = [char(paramChain.txtChainName), ': ', msg, ' Berechnungen für diese
Prozesskette werden unterbrochen.'];
407     if startGuiFlag == true
408         BgaPtg2UpdateMessageBox(handles,'apend',msg);
409     else
410         disp(msg);
411     end
412     errorInLoopsFlag = true;     return;
413 end
414
415 % chainHeatUseHT              =
paramChain.valueChainParam(find(strcmp(paramChain.txtChainParamInfo(:,1),'chainHeatUse
HT')));
416 % chainHeatUseNT              =
paramChain.valueChainParam(find(strcmp(paramChain.txtChainParamInfo(:,1),'chainHeatUse
NT')));
417 chainEffMeth                 = paramChain.valueChainParam(find(strcmp(paramChain.
txtChainParamInfo(:,1),'chainEffMeth')));
418 if isempty(chainEffMeth)
419     msg = 'Parameter "chainEffMeth" bzw. dessen Wert ist nicht definiert.';
420     msg = [char(paramChain.txtChainName), ': ', msg, ' Berechnungen für diese
Prozesskette werden unterbrochen.'];
421     if startGuiFlag == true
422         BgaPtg2UpdateMessageBox(handles,'apend',msg);
423     else
424         disp(msg);
425     end
426     errorInLoopsFlag = true;     return;
427 end
428
429 % chainStoichiometry          =
paramChain.valueChainParam(find(strcmp(paramChain.txtChainParamInfo(:,1),'chainStoichi
ometry')));
430 % chainOverscalingMeth        =
paramChain.valueChainParam(find(strcmp(paramChain.txtChainParamInfo(:,1),'chainOversca
lingMeth')));

```



```

431 % chainOverscalingEly          =
paramChain.valueChainParam(find(strcmp(paramChain.txtChainParamInfo(:,1),'chainOversca
lingEly')));
432 % chainPowerCHP              =
paramChain.valueChainParam(find(strcmp(paramChain.txtChainParamInfo(:,1),'chainPowerCH
P')));
433 chainEffCHP                  = paramChain.valueChainParam(find(strcmp(paramChain.
txtChainParamInfo(:,1),'chainEffCHP')));
434 if isempty(chainEffCHP)
435     msg = 'Parameter "chainEffCHP" bzw. dessen Wert ist nicht definiert.';
436     msg = [char(paramChain.txtChainName), ': ', msg, ' Berechnungen für diese
Prozesskette werden unterbrochen.'];
437     if startGuiFlag == true
438         BgaPtg2UpdateMessageBox(handles,'apend',msg);
439     else
440         disp(msg);
441     end
442     errorInLoopsFlag = true;      return;
443 end
444
445 chainEffEly                    = paramChain.valueChainParam(find(strcmp(paramChain.
txtChainParamInfo(:,1),'chainEffEly')));
446 if isempty(chainEffEly)
447     msg = 'Parameter "chainEffEly" bzw. dessen Wert ist nicht definiert.';
448     msg = [char(paramChain.txtChainName), ': ', msg, ' Berechnungen für diese
Prozesskette werden unterbrochen.'];
449     if startGuiFlag == true
450         BgaPtg2UpdateMessageBox(handles,'apend',msg);
451     else
452         disp(msg);
453     end
454     errorInLoopsFlag = true;      return;
455 end
456
457 % techChainVolFlowBG          =
paramChain.valueChainParam(find(strcmp(paramChain.txtChainParamInfo(:,1),'chainVolFlow
BG')));
458
459 chainElyFLH                    = paramChain.valueChainParam(find(strcmp(paramChain.
txtChainParamInfo(:,1),'chainElyFLH')));
460 if isempty(chainElyFLH)
461     msg = 'Parameter "chainElyFLH" bzw. dessen Wert ist nicht definiert.';
462     msg = [char(paramChain.txtChainName), ': ', msg, ' Berechnungen für diese
Prozesskette werden unterbrochen.'];
463     if startGuiFlag == true
464         BgaPtg2UpdateMessageBox(handles,'apend',msg);
465     else
466         disp(msg);
467     end
468     errorInLoopsFlag = true;      return;
469 end
470
471
472 %% Parameter (Ende) *****
473
474 % Konstanten lesen
475 HHV_CH4                        = const.HHV_CH4;
476 HHV_H2                          = const.HHV_H2;
477 HHV_CO2                         = const.HHV_CO2;
478 HHV_H2S                         = const.HHV_H2S;
479 HHV_H2O                         = const.HHV_H2O;
480 HHV_Other                       = const.HHV_Other;
481 HHV_TraceGas                   = const.HHV_TraceGas;
482 density_Air                    = const.density_Air;
483 density_H2                      = const.density_H2;
484
485 % techChainVolFlowBG = techSimuPreCalcChainVec(1);
486 % chainVolFlowCO2   = techSimuPreCalcChainVec(2);
487 % chainVolFlowH2    = techSimuPreCalcChainVec(3);
488 chainVolFlowH2kg    = techSimuPreCalcChainVec(4);
489 %techChainMethPower = techSimuPreCalcChainVec(5);
490 % techChainElyPower = techSimuPreCalcChainVec(6);
491 %techChainpSystem   = techSimuPreCalcChainVec(7);

```



```

492 %chainp4Compr          = techSimuPreCalcChainVec(8);
493
494
495
496 % Einheiten Konvertierungen
497 chainDiscRate          = chainDiscRate/100;
498 chainElecPrice         = chainElecPrice/100;
499 chainOperationFactor   = chainOperationFactor/100;
500 chainInsurFactor       = chainInsurFactor/100;
501 chainRenatFactor       = chainRenatFactor/100;
502 chainTechInflRate      = chainTechInflRate/100;
503 chainBuildInflRate     = chainBuildInflRate/100;
504 chainInflRate          = chainInflRate/100;
505 chainWaterPrice        = chainWaterPrice;
506 chainBGPurchasePrice    = chainBGPurchasePrice/100;
507 chainBMSellingPrice     = chainBMSellingPrice/100;
508 chainHeatPurchasePrice  = chainHeatPurchasePrice/100;
509 chainHeatSellingPrice   = chainHeatSellingPrice/100;
510 %chainO2SellingPrice    = chainO2SellingPrice/100;
511 chainComprEff          = chainComprEff/100;
512 chainEffMeth           = chainEffMeth/100;
513 chainEffCHP            = chainEffCHP/100;
514 chainEffEly           = chainEffEly/100;
515 % chainHeatUseHT       = chainHeatUseHT/100;
516 % chainHeatUseNT       = chainHeatUseNT/100;
517
518
519
520 % Anfang des Programms
521 % compp                 = techSimuCompValueVecAll(end,1);
522 % compT                 = techSimuCompValueVecAll(end,2);
523 compVolFlowBioM         = techSimuCompValueVecAll(end,3);
524 compVolFlowSNG          = techSimuCompValueVecAll(end,4);
525 % compVolFlowCO2        = techSimuCompValueVecAll(end,5);
526 % compVolFlowH2         = techSimuCompValueVecAll(end,6);
527 % compVolFlowH2S        = techSimuCompValueVecAll(end,7);
528 % compVolFlowH2O        = techSimuCompValueVecAll(end,8);
529 % compVolFlowTraceGas   = techSimuCompValueVecAll(end,9);
530 % compVolFlowOther      = techSimuCompValueVecAll(end,10);
531 compVolFlowPG           = techSimuCompValueVecAll(end,11);
532 compConcBioM            = techSimuCompValueVecAll(end,12);
533 compConcSNG             = techSimuCompValueVecAll(end,13);
534 compConcCO2             = techSimuCompValueVecAll(end,14);
535 compConcH2              = techSimuCompValueVecAll(end,15);
536 compConcH2S             = techSimuCompValueVecAll(end,16);
537 compConcH2O             = techSimuCompValueVecAll(end,17);
538 compConcTraceGas        = techSimuCompValueVecAll(end,18);
539 compConcOther           = techSimuCompValueVecAll(end,19);
540 compDensity_PG          = techSimuCompValueVecAll(end,20);
541
542 % techCompSpecialOutput zuweisen
543 % techChainConvCO2Ideal = techCompSpecialOutput(1);
544 % techChainConvCO2Real  = techCompSpecialOutput(2);
545 % techChainComprPower   = techCompSpecialOutput(3);
546 % techChainNutrientDemand = techCompSpecialOutput(4);
547 techChainHeatHT_ideal   = techCompSpecialOutput(5);
548 techChainHeatNT_ideal   = techCompSpecialOutput(6);
549 techChainHeatHT_real    = techCompSpecialOutput(7);
550 techChainHeatNT_real    = techCompSpecialOutput(8);
551
552
553
554 % Reiner CH4-Strom, der aus Biogas weitergeleitet wird.
555 techChainVolFlowBioM    = compVolFlowBioM;
556 techChainVolFlowSNG     = compVolFlowSNG;
557 % chainVolFlowCO2       = compVolFlowCO2;
558 % chainVolFlowH2        = compVolFlowH2;
559 % chainVolFlowH2S       = compVolFlowH2S;
560 % chainVolFlowH2O       = compVolFlowH2O;
561 % chainVolFlowTraceGas  = compVolFlowTraceGas;
562 % chainVolFlowOther     = compVolFlowOther;
563 techChainVolFlowPG      = compVolFlowPG;
564

```

```

565 % : Hier noch keine Konvertierung
566 chainConcOutputBioM           = compConcBioM;
567 chainConcOutputSNG           = compConcSNG;
568 techChainConcOutputCH4       = chainConcOutputBioM + chainConcOutputSNG;
569 techChainConcOutputCO2       = compConcCO2;
570 techChainConcOutputH2        = compConcH2;
571 techChainConcOutputH2S       = compConcH2S;
572 techChainConcOutputH2O       = compConcH2O;
573 techChainConcOutputTraceGas   = compConcTraceGas;
574 techChainConcOutputOther     = compConcOther;
575
576 if density_Air == 0
577     msg = 'Konstante "density_Air" ist gleich null. Division durch null.';
578     msg = [char(paramChain.txtChainName), ': ', msg, ' Berechnungen für diese
579           Prozesskette werden unterbrochen.'];
580     if startGuiFlag == true
581         BgaPtg2UpdateMessageBox(handles,'append',msg);
582     else
583         disp(msg);
584     end
585     errorInLoopsFlag = true;    return;
586 else
587     techChainRelDensityPG = compDensity_PG/density_Air;
588 end;
589
590
591 % GasQualität für Einspeisequalität DVGW
592 % techChainHHV_PG = techChainConcInputCH4 * HHV_CH4 + techChainConcInputH2 * HHV_H2
593 + ...
594 % techChainConcInputCO2 * HHV_CO2 + techChainConcInputH2O *
595 HHV_H2O + ...
596 % techChainConcInputH2S * HHV_H2S + techChainConcInputTraceGas *
597 HHV_TraceGas + techChainConcInputOther * HHV_Other;
598 techChainHHV_PG = techChainConcOutputCH4 * HHV_CH4 + techChainConcOutputH2 * HHV_H2
599 + ...
600 techChainConcOutputCO2 * HHV_CO2 + techChainConcOutputH2O *
601 HHV_H2O + ...
602 techChainConcOutputH2S * HHV_H2S + techChainConcOutputTraceGas *
603 HHV_TraceGas + techChainConcOutputOther * HHV_Other;
604
605 chainConcOutputBioM           = chainConcOutputBioM * 100;
606 chainConcOutputSNG           = chainConcOutputSNG * 100;
607 techChainConcOutputCH4       = techChainConcOutputCH4 * 100;
608 techChainConcOutputCO2       = techChainConcOutputCO2 * 100;
609 techChainConcOutputH2        = techChainConcOutputH2 * 100;
610 techChainConcOutputH2S       = techChainConcOutputH2S * 10^6;
611 techChainConcOutputH2O       = techChainConcOutputH2O * 100;
612 techChainConcOutputTraceGas   = techChainConcOutputTraceGas * 10^6;
613 techChainConcOutputOther     = techChainConcOutputOther * 100;
614 % oben wird berechnet.
615 %techChainRelDensityPG       = compDensity_PG/density_Air;
616
617 % techChainVolFlowBG         = techPreCalcChainOutputHelpAll(1);
618 % techChainConcInputCH4     = techPreCalcChainOutputHelpAll(2);
619 % techChainConcInputCO2     = techPreCalcChainOutputHelpAll(3);
620 % techChainConcInputH2      = techPreCalcChainOutputHelpAll(4);
621 % techChainConcInputH2S     = techPreCalcChainOutputHelpAll(5);
622 % techChainConcInputH2O     = techPreCalcChainOutputHelpAll(6);
623 % techChainConcInputOther    = techPreCalcChainOutputHelpAll(7);
624 % techChainConcInputTraceGas = techPreCalcChainOutputHelpAll(8);
625 techChainMethPower           = techPreCalcChainOutputHelpAll(9);
626 % techChainElyPower         = techPreCalcChainOutputHelpAll(10);
627 % techChainpSystem          = techPreCalcChainOutputHelpAll(11);
628
629 if techChainRelDensityPG == 0
630     msg = 'Variable "techChainRelDensityPG" ist gleich null. Division durch null.';
631     msg = [char(paramChain.txtChainName), ': ', msg, ' Berechnungen für diese
632           Prozesskette werden unterbrochen.'];
633     if startGuiFlag == true

```

```

630         BgaPtg2UpdateMessageBox(handles,'apend',msg);
631     else
632         disp(msg);
633     end
634     errorInLoopsFlag = true;     return;
635
636 else
637     techChainWobIdx = techChainHHV_PG/(techChainRelDensityPG^0.5);
638 end;
639
640
641 % Strom
642 techChainElecDemandTech = chainElecSpecTech * techChainMethPower * techChainFLH;
643 techChainElecDemandBuild = chainElecSpecBuild * techChainMethPower * techChainFLH;
644
645 techChainElecDemand = techChainElecDemandTech + techChainElecDemandBuild;
646
647 % Wärme
648 techChainHeatDemandTech = chainHeatSpecTech * techChainMethPower * techChainFLH;
649 techChainHeatDemandBuild = chainHeatSpecBuild * techChainMethPower * techChainFLH;
650
651 techChainHeatDemand = techChainHeatDemandTech + techChainHeatDemandBuild;
652
653 % Hier soll FLH Methanisierung genutzt werden
654 techCompFLHMeth = 0;
655 for loop = 1:size(paramComp,1)
656     compColInExcel = paramComp(loop).sheet;
657     if(strcmp(compColInExcel,'AK') || strcmp(compColInExcel,'AL') || strcmp(
658         compColInExcel,'AM') || ...
659         strcmp(compColInExcel,'AN') || strcmp(compColInExcel,'AO') || strcmp(
660         compColInExcel,'AP') || ...
661         strcmp(compColInExcel,'AQ') || strcmp(compColInExcel,'AR') || strcmp(
662         compColInExcel,'AS') || ...
663         strcmp(compColInExcel,'AT') || strcmp(compColInExcel,'AU') || strcmp(
664         compColInExcel,'AV') || ...
665         strcmp(compColInExcel,'AW') || strcmp(compColInExcel,'AX'))
666         techCompFLHMeth = paramComp(loop).valueCompParam(find(strcmp(paramComp(loop)
667             ).txtCompParamInfo(:,1),'techCompFLH')));
668
669         if isempty(chainEffEly)
670             msg = 'Parameter "chainEffEly" bzw. dessen Wert ist nicht definiert.';
671
672             msg = [char(paramChain.txtChainName), ': ', msg, ' Berechnungen für
673                 diese Prozesskette werden unterbrochen.'];
674             if startGuiFlag == true
675                 BgaPtg2UpdateMessageBox(handles,'apend',msg);
676             else
677                 disp(msg);
678             end
679             errorInLoopsFlag = true;     return;
680         end
681         break; % AK bis AX schließen sich gegenseitig aus
682     end
683 end
684
685 % Met % : Soll FLH Methanisierung sein, dafür Abfrage aus Matrix nötig
686 techChainEnergySNG = techChainVolFlowSNG * HHV_CH4 * techCompFLHMeth;
687
688 % Hier: WSimu rechnet Kosten auf techChainEnergySNG!
689
690 % chainBioMethaneSurplus in WSimu
691 techChainEnergyBioM = techChainVolFlowBioM * HHV_CH4 * techCompFLHMeth; %Meth % FLH
692 s.o.
693
694 techChainEnergyCH4 = techChainEnergySNG + techChainEnergyBioM;
695
696
697 % techChainVolFlowSNGMix: Aufteilung welcher Gasmix zu SNG gehört, welcher zu
698 "Biomethan". Beide sind zusammen im Endgas, BioM allein rauszurechnen, "verunreinigt"
699 SNG

```

```

693 % nicht endültig definiert, daher = 0
694 techChainVolFlowSNGMix = 0; % Platzhalter, da nicht endgültig definiert
695 techChainVolFlowBioMMix = 0; % Platzhalter, da nicht endgültig definiert
696 % techChainVolFlowSNGMix = techChainVolFlowPG - techChainVolFlowBioM; % Platzhalter,
da nicht endgültig definiert
697 % techChainVolFlowBioMMix = techChainVolFlowPG - techChainVolFlowSNG ;% Platzhalter,
da nicht endgültig definiert
698
699 %Summenbildung für Gesamtstrom aller Komponenten
700 techChainElecDemandCompTotal = sum(techSimuCompVec(:,3));
701
702 techChainElecDemandTotal = techChainElecDemandCompTotal + techChainElecDemand;
703
704 techChainElecDemandMethTotal = techChainElecDemandTotal - techSimuCompVec(1,3); %
Gesamt minus Elektrolyse- Strom (nur wenn Elektrolyse gerechnet!)
705
706 %Summenbildung für GesamtWärme aller Komponenten
707 techChainHeatDemandCompTotal = sum(techSimuCompVec(:,6));
708
709 techChainHeatDemandTotal = techChainHeatDemandCompTotal + techChainHeatDemand;
710
711 chainEnergyDemandElecHeatTotal = techChainElecDemandTotal + techChainHeatDemandTotal;
712
713 techChainHeatTotal_ideal = techChainHeatHT_ideal + techChainHeatNT_ideal;
714 %techChainHeatHT_ideal
715 techChainHeatTotal_real = techChainHeatHT_real + techChainHeatNT_real;
716
717 techChainEnergyDemandElecHeatTotal = techChainElecDemandTotal +
techChainHeatDemandTotal;
718
719 if density_H2 == 0
720     msg = 'Variable "chainEnergyH2" ist gleich null. Division durch null.';
721     msg = [char(paramChain.txtChainName), ': ', msg, ' Berechnungen für diese
Prozesskette werden unterbrochen.'];
722     if startGuiFlag == true
723         BgaPtg2UpdateMessageBox(handles,'apend',msg);
724     else
725         disp(msg);
726     end
727     errorInLoopsFlag = true;     return;
728
729 else
730     chainEnergyH2 = chainVolFlowH2kg/density_H2 * HHV_H2 * chainElyFLH;
731 end;
732
733
734 % Wirkungsgrade machen keinen Sinn, wenn Strominput für Ely nicht bekannt, d.h. = Null
735 % : if/else & =0
736 if techSimuCompVec(1,3) == 0
737     techChainEffElec = 0;
738     techChainEffTotal = 0;
739     techChainEffHeatTotal = 0;
740     techChainEffMeth_elec = 0;
741     techChainEffMeth_total = 0;
742     techChainEffMeth_HeatTotal = 0;
743 else
744
745
746     if techChainElecDemandTotal == 0
747         msg = 'Variable "techChainElecDemandTotal" ist gleich null. Division durch
null.';
748         msg = [char(paramChain.txtChainName), ': ', msg, ' Berechnungen für diese
Prozesskette werden unterbrochen.'];
749         if startGuiFlag == true
750             BgaPtg2UpdateMessageBox(handles,'apend',msg);
751         else
752             disp(msg);
753         end
754         errorInLoopsFlag = true;     return;
755
756     else
757         techChainEffElec = techChainEnergySNG/techChainElecDemandTotal; % korrekte
Division, sonst grober Fehler da kein Strombedarf

```

```

758     end
759
760
761
762     if (techChainElecDemandTotal + techChainHeatDemandTotal) == 0
763         msg = 'Variable "techChainElecDemandTotal + techChainHeatDemandTotal" ist
764             gleich null. Division durch null.';
765         msg = [char(paramChain.txtChainName), ': ', msg, ' Berechnungen für diese
766             Prozesskette werden unterbrochen.'];
767         if startGuiFlag == true
768             BgaPtg2UpdateMessageBox(handles,'apend',msg);
769         else
770             disp(msg);
771         end
772         errorInLoopsFlag = true;     return;
773
774     else
775         techChainEffTotal = techChainEnergySNG/(techChainElecDemandTotal +
776             techChainHeatDemandTotal);
777     end
778
779     if (techChainElecDemandTotal + techChainHeatDemandTotal) == 0
780         msg = 'Variable "techChainElecDemandTotal + techChainHeatDemandTotal" ist
781             gleich null. Division durch null.';
782         msg = [char(paramChain.txtChainName), ': ', msg, ' Berechnungen für diese
783             Prozesskette werden unterbrochen.'];
784         if startGuiFlag == true
785             BgaPtg2UpdateMessageBox(handles,'apend',msg);
786         else
787             disp(msg);
788         end
789         errorInLoopsFlag = true;     return;
790
791     else
792         techChainEffHeatTotal = (techChainEnergySNG + techChainHeatTotal_ideal)/(
793             techChainElecDemandTotal + techChainHeatDemandTotal);
794     end
795
796     % wenn kein Strom ermittelt für
797
798     if (techChainElecDemandMethTotal + chainEnergyH2) == 0
799         msg = 'Variable "techChainElecDemandMethTotal + chainEnergyH2" ist gleich
800             null. Division durch null.';
801         msg = [char(paramChain.txtChainName), ': ', msg, ' Berechnungen für diese
802             Prozesskette werden unterbrochen.'];
803         if startGuiFlag == true
804             BgaPtg2UpdateMessageBox(handles,'apend',msg);
805         else
806             disp(msg);
807         end
808         errorInLoopsFlag = true;     return;
809
810     else
811         techChainEffMeth_elec = techChainEnergySNG/(techChainElecDemandMethTotal +
812             chainEnergyH2); %
813     end
814
815     if (techChainElecDemandMethTotal + techChainHeatDemandTotal + chainEnergyH2) == 0
816         msg = 'Variable "techChainElecDemandMethTotal + techChainHeatDemandTotal +
817             chainEnergyH2" ist gleich null. Division durch null.';
818         msg = [char(paramChain.txtChainName), ': ', msg, ' Berechnungen für diese
819             Prozesskette werden unterbrochen.'];
820         if startGuiFlag == true
821             BgaPtg2UpdateMessageBox(handles,'apend',msg);
822         else
823             disp(msg);
824         end
825         errorInLoopsFlag = true;     return;
826
827     else
828         techChainEffMeth_total = techChainEnergySNG/(techChainElecDemandMethTotal +

```

```

    techChainHeatDemandTotal + chainEnergyH2); %
820 end
821
822
823 if (techChainElecDemandMethTotal + techChainHeatDemandTotal + chainEnergyH2) == 0
824     msg = 'Variable "techChainElecDemandMethTotal + techChainHeatDemandTotal +
chainEnergyH2" ist gleich null. Division durch null.';
825     msg = [char(paramChain.txtChainName), ': ', msg, ' Berechnungen für diese
Prozesskette werden unterbrochen.'];
826     if startGuiFlag == true
827         BgaPtg2UpdateMessageBox(handles,'apend',msg);
828     else
829         disp(msg);
830     end
831     errorInLoopsFlag = true;     return;
832
833 else
834     techChainEffMeth_HeatTotal = (techChainEnergySNG + techChainHeatTotal_ideal) /
(techChainElecDemandMethTotal + techChainHeatDemandTotal + chainEnergyH2);
835 end
836 end
837
838
839
840
841 % Umwandlung für Results
842 techChainEffElec           = techChainEffElec * 100;
843 techChainEffTotal         = techChainEffTotal * 100;
844 techChainEffHeatTotal     = techChainEffHeatTotal * 100;
845 techChainEffMeth_elec     = techChainEffMeth_elec * 100;
846 techChainEffMeth_total    = techChainEffMeth_total * 100;
847 techChainEffMeth_HeatTotal = techChainEffMeth_HeatTotal * 100;
848
849
850 techChainH2kg = chainVolFlowH2kg * chainElyFLH;
851
852 techChainPlatzhalter2     = 0;
853 techChainPlatzhalter3    = 0;
854 techChainPlatzhalter4    = 0;
855
856
857 techSimuChainVec = [techChainFLH; techChainEnergySNG; techChainHeatDemand;
techChainHeatTotal_real;
858                     techChainVolFlowPG; techChainEnergyBioM; techChainElecDemand];
859
860
861 % Übergaben für Results
862 techSimuOutputVec = [techChainEnergyBioM; techChainEnergyCH4; techChainVolFlowBioMMix;
techChainVolFlowSNG; ...
863                     techChainConcOutputCH4; techChainConcOutputCO2;
techChainConcOutputH2; techChainConcOutputH2S;
techChainConcOutputH2O; techChainConcOutputTraceGas;
techChainConcOutputOther; ...
864                     techChainHHV_PG; techChainRelDensityPG; techChainWobIdx;
techChainVolFlowSNGMix; techChainElecDemandTech;
techChainElecDemandBuild; techChainHeatDemandTech;
techChainHeatDemandBuild; ...
865                     techChainElecDemandTotal; techChainElecDemandCompTotal;
techChainElecDemandMethTotal; techChainHeatDemandTotal;
techChainHeatDemandCompTotal; techChainHeatTotal_ideal;
techChainHeatTotal_real; techChainHeatHT_ideal;
techChainHeatHT_real; techChainHeatNT_ideal; techChainHeatNT_real
; techChainEnergyDemandElecHeatTotal; ...
866                     techChainEffElec; techChainEffTotal; techChainEffHeatTotal;
techChainEffMeth_elec; techChainEffMeth_total;
techChainEffMeth_HeatTotal; ...
867                     techChainH2kg; techChainPlatzhalter2; techChainPlatzhalter3;
techChainPlatzhalter4];
868

```