

```

1
2 % Programm für die Kostenberechnung auf Prozesskettenebene
3
4 function [costChainAnnuityTotals, costChainAnnuities, chainCapex, costInvestChainTotal
, costChainAnnuityTotal, costChainAnnuityTotalRel, costChainAnnuityHydrogen,
costHydrogenLCOH, ...
5     costCompAnnuity, costCompAnnuityRel, costChainAnnuity, errorInLoopsFlag] =
    ...
6     BgaPtg2CalcChainCost(handles,paramChain,paramComp,costCompAnnuities,
costCompInvest,techSimuCompVec,techSimuChainVec,techSimuPreCalcChainVec)
7
8 % Interne Parameter
9 global inParam;
10 global const;
11
12 % Damit ohne GUI gearbeitet werden kann.
13 global startGuiFlag;
14
15 % Outputs definieren
16 costChainAnnuityTotals = ones(6,1)*-1;
17 costChainAnnuities = ones(6,1)*-1;
18 chainCapex = -1;
19 costInvestChainTotal = -1;
20 costChainAnnuityTotal = -1;
21 costChainAnnuityTotalRel = ones(inParam.nrofCostAnnuities+1,1)*-1;
22 costChainAnnuityHydrogen = -1;
23 costHydrogenLCOH = -1;
24 costCompAnnuity = ones(size(costCompAnnuities,1),1)*-1;
25 costCompAnnuityRel = ones(size(costCompAnnuities,1),1)*-1;
26 costChainAnnuity = -1;
27 errorInLoopsFlag = false;
28 %return;
29
30
31 % Initialisierung, wird später berechnet
32 inParam.chainTechReplacement = 0;
33
34 % Konstante
35 LHV_kg_H2 = const.LHV_kg_H2;
36 HHV_kg_H2 = const.HHV_kg_H2;
37
38 % Zuweisung technischer Berechnungen
39
40 % Volllaststunden auf PK-Ebene [h]
41 % Energiemenge SNG [kWh th]
42 % Wärmebedarf auf PK-Ebene [kWh th]
43 % Wärme Überschuss auf PK-Ebene [kWh th]
44 % Biogas Bedarf auf PK-Ebene [kWh th]
45 % BioMethan Überschuss auf PK-Ebene [kWh th]
46 % Strombedarf für übergeordnete Etechnik [kWh el]
47 % Leistung der Methanisierung wird auf Komp.Ebene zur Berechnung benötigt [kW th]
48 % chainWater4H2
49 % chainElyPers
50
51 techChainFLH = techSimuChainVec(1);
52 %techChainEnergySNG = techSimuChainVec(2);
53 techChainHeatDemand = techSimuChainVec(3);
54 %techChainHeatTotal_real = techSimuChainVec(4);
55 techChainVolFlowPG = techSimuChainVec(5);
56 techChainEnergyBioM = techSimuChainVec(6);
57 techChainElecDemand = techSimuChainVec(7);
58 %techChainMethPower = techSimuChainVec(8);
59
60 % techChainVolFlowBG = techSimuPreCalcChainVec(1);
61 % chainVolFlowCO2 = techSimuPreCalcChainVec(2);
62 % chainVolFlowH2 = techSimuPreCalcChainVec(3);
63 % chainVolFlowH2kg = techSimuPreCalcChainVec(4);
64 techChainMethPower = techSimuPreCalcChainVec(5);
65 % techChainElyPower = techSimuPreCalcChainVec(6);
66 % techChainpSystem = techSimuPreCalcChainVec(7);
67
68 %% Parameter (Anfang) *****
69 % Projektlaufzeit [a]

```

```

70 chainPeriod = paramChain.valueChainParam(find(strcmp(paramChain.txtChainParamInfo(:,1), 'chainPeriod')));
71 if isempty(chainPeriod)
72     msg = 'Parameter "chainPeriod" bzw. dessen Wert ist nicht definiert.';
73     msg = [char(paramChain.txtChainName), ': ', msg, ' Berechnungen für diese Prozesskette werden unterbrochen.'];
74     if startGuiFlag == true
75         BgaPtg2UpdateMessageBox(handles, 'apend', msg);
76     else
77         disp(msg);
78     end
79     errorInLoopsFlag = true;     return;
80 end
81
82
83 % Kalkulationszinsatz, discount rate [%]
84 chainDiscRate = paramChain.valueChainParam(find(strcmp(paramChain.txtChainParamInfo(:,1), 'chainDiscRate')));
85 if isempty(chainDiscRate)
86     msg = 'Parameter "chainDiscRate" 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
97 % Aktivkohlepreis [€/kg] Aktivkohle
98 %chainActCPrice =
99 paramChain.valueChainParam(find(strcmp(paramChain.txtChainParamInfo(:,1), 'chainActCPrice')));
100
101 % Strompreis [ct/kWh]
102 chainElecPrice = paramChain.valueChainParam(find(strcmp(paramChain.txtChainParamInfo(:,1), 'chainElecPrice')));
103 if isempty(chainElecPrice)
104     msg = 'Parameter "chainElecPrice" bzw. dessen Wert ist nicht definiert.';
105     msg = [char(paramChain.txtChainName), ': ', msg, ' Berechnungen für diese Prozesskette werden unterbrochen.'];
106     if startGuiFlag == true
107         BgaPtg2UpdateMessageBox(handles, 'apend', msg);
108     else
109         disp(msg);
110     end
111     errorInLoopsFlag = true;     return;
112 end
113
114 % Biogas Einkaufspreis [ct/kWh]
115 chainBGPurchasePrice = paramChain.valueChainParam(find(strcmp(paramChain.txtChainParamInfo(:,1), 'chainBGPurchasePrice')));
116 if isempty(chainBGPurchasePrice)
117     msg = 'Parameter "chainBGPurchasePrice" bzw. dessen Wert ist nicht definiert.';
118     msg = [char(paramChain.txtChainName), ': ', msg, ' Berechnungen für diese Prozesskette werden unterbrochen.'];
119     if startGuiFlag == true
120         BgaPtg2UpdateMessageBox(handles, 'apend', msg);
121     else
122         disp(msg);
123     end
124     errorInLoopsFlag = true;     return;
125 end
126
127 % Biogas Verkaufspreis, muss negativ sein
128 % Um EingabeFehler zu vermeiden, z.B. Eingabe von Neg. Werten, wird AbsolutWert geholt
129 % Eingabe muss positiv sein [ct/kWh]
130 chainBMSellingPrice = paramChain.valueChainParam(find(strcmp(paramChain.txtChainParamInfo(:,1), 'chainBMSellingPrice')));
131 if isempty(chainBMSellingPrice)

```

```

132     msg = 'Parameter "chainBMSellingPrice" bzw. dessen Wert ist nicht definiert.';
133     msg = [char(paramChain.txtChainName), ': ', msg, ' Berechnungen für diese
        Prozesskette werden unterbrochen.'];
134     if startGuiFlag == true
135         BgaPtg2UpdateMessageBox(handles,'apend',msg);
136     else
137         disp(msg);
138     end
139     errorInLoopsFlag = true;     return;
140 end
141
142 % Wärme Einkaufspreis in [ct/kWh]
143 chainHeatPurchasePrice = paramChain.valueChainParam(find(strcmp(paramChain.
    txtChainParamInfo(:,1),'chainHeatPurchasePrice')));
144 if isempty(chainHeatPurchasePrice)
145     msg = 'Parameter "chainHeatPurchasePrice" bzw. dessen Wert ist nicht definiert.';
146     msg = [char(paramChain.txtChainName), ': ', msg, ' Berechnungen für diese
        Prozesskette werden unterbrochen.'];
147     if startGuiFlag == true
148         BgaPtg2UpdateMessageBox(handles,'apend',msg);
149     else
150         disp(msg);
151     end
152     errorInLoopsFlag = true;     return;
153 end
154
155
156 % Investkosten in Bauliche Anlagen, Container, Integration
157 % Fixwert [€]
158 chainBuildInvest = paramChain.valueChainParam(find(strcmp(paramChain.txtChainParamInfo
    (:,1),'chainBuildInvest')));
159 if isempty(chainBuildInvest)
160     msg = 'Parameter "chainBuildInvest" bzw. dessen Wert ist nicht definiert.';
161     msg = [char(paramChain.txtChainName), ': ', msg, ' Berechnungen für diese
        Prozesskette werden unterbrochen.'];
162     if startGuiFlag == true
163         BgaPtg2UpdateMessageBox(handles,'apend',msg);
164     else
165         disp(msg);
166     end
167     errorInLoopsFlag = true;     return;
168 end
169
170 % Investkosten Mess-/RegelTech, Elektroinstallation, Sicherheitstechnik, Leitwarte
171 % Fixwert [€]
172 chainTechInvest = paramChain.valueChainParam(find(strcmp(paramChain.txtChainParamInfo
    (:,1),'chainTechInvest')));
173 if isempty(chainTechInvest)
174     msg = 'Parameter "chainTechInvest" 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 % Einmalige Kosten für Planung, Genehmigung, Fixwert [€]
185 chainPlanningApproval = paramChain.valueChainParam(find(strcmp(paramChain.
    txtChainParamInfo(:,1),'chainPlanningApproval')));
186 if isempty(chainPlanningApproval)
187     msg = 'Parameter "chainPlanningApproval" bzw. dessen Wert ist nicht definiert.';
188
189     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 % Betriebskosten-Faktor für die Technik [%]
198 chainOperationFactor = paramChain.valueChainParam(find(strcmp(paramChain.
txtChainParamInfo(:,1),'chainOperationFactor')));
199 if isempty(chainOperationFactor)
200     msg = 'Parameter "chainOperationFactor" bzw. dessen Wert ist nicht definiert.';
201     msg = [char(paramChain.txtChainName), ': ', msg, ' Berechnungen für diese
Prozesskette werden unterbrochen.'];
202     if startGuiFlag == true
203         BgaPtg2UpdateMessageBox(handles,'apend',msg);
204     else
205         disp(msg);
206     end
207     errorInLoopsFlag = true;     return;
208 end
209
210 % Versicherungsfaktor der Gesamtanlage [%] % of CAPEX
211 chainInsurFactor = paramChain.valueChainParam(find(strcmp(paramChain.txtChainParamInfo
(:,1),'chainInsurFactor')));
212 if isempty(chainInsurFactor)
213     msg = 'Parameter "chainInsurFactor" bzw. dessen Wert ist nicht definiert.';
214     msg = [char(paramChain.txtChainName), ': ', msg, ' Berechnungen für diese
Prozesskette werden unterbrochen.'];
215     if startGuiFlag == true
216         BgaPtg2UpdateMessageBox(handles,'apend',msg);
217     else
218         disp(msg);
219     end
220     errorInLoopsFlag = true;     return;
221 end
222
223 % Rückbau, Rücklagenfaktor der Gesamtanlage, en: renaturation reserve [%] #of CAPEX
224 chainRenatFactor = paramChain.valueChainParam(find(strcmp(paramChain.txtChainParamInfo
(:,1),'chainRenatFactor')));
225 if isempty(chainRenatFactor)
226     msg = 'Parameter "chainRenatFactor" bzw. dessen Wert ist nicht definiert.';
227     msg = [char(paramChain.txtChainName), ': ', msg, ' Berechnungen für diese
Prozesskette werden unterbrochen.'];
228     if startGuiFlag == true
229         BgaPtg2UpdateMessageBox(handles,'apend',msg);
230     else
231         disp(msg);
232     end
233     errorInLoopsFlag = true;     return;
234 end
235
236 % Personalkosten der Anlage, Fixwert €
237 chainCostPers = paramChain.valueChainParam(find(strcmp(paramChain.txtChainParamInfo(:,
1),'chainCostPers')));
238 if isempty(chainCostPers)
239     msg = 'Parameter "chainCostPers" bzw. dessen Wert ist nicht definiert.';
240     msg = [char(paramChain.txtChainName), ': ', msg, ' Berechnungen für diese
Prozesskette werden unterbrochen.'];
241     if startGuiFlag == true
242         BgaPtg2UpdateMessageBox(handles,'apend',msg);
243     else
244         disp(msg);
245     end
246     errorInLoopsFlag = true;     return;
247 end
248
249 % Sonstige Kosten, fix €
250 chainCostOther = paramChain.valueChainParam(find(strcmp(paramChain.txtChainParamInfo
(:,1),'chainCostOther')));
251 if isempty(chainCostOther)
252     msg = 'Parameter "chainCostOther" bzw. dessen Wert ist nicht definiert.';
253     msg = [char(paramChain.txtChainName), ': ', msg, ' Berechnungen für diese
Prozesskette werden unterbrochen.'];
254     if startGuiFlag == true
255         BgaPtg2UpdateMessageBox(handles,'apend',msg);
256     else
257         disp(msg);

```

```

258     end
259     errorInLoopsFlag = true;     return;
260 end
261
262 % Werte für ersetzbare technische Bauteile auf Prozesskettenebene
263 chainTechInflRate = paramChain.valueChainParam(find(strcmp(paramChain.
txtChainParamInfo(:,1),'chainTechInflRate')));
264 if isempty(chainTechInflRate)
265     msg = 'Parameter "chainTechInflRate" bzw. dessen Wert ist nicht definiert.';
266     msg = [char(paramChain.txtChainName), ': ', msg, ' Berechnungen für diese
Prozesskette werden unterbrochen.'];
267     if startGuiFlag == true
268         BgaPtg2UpdateMessageBox(handles,'apend',msg);
269     else
270         disp(msg);
271     end
272     errorInLoopsFlag = true;     return;
273 end
274
275 chainTechLifetimeHours = paramChain.valueChainParam(find(strcmp(paramChain.
txtChainParamInfo(:,1),'chainTechLifetimeHours')));
276 if isempty(chainTechLifetimeHours)
277     msg = 'Parameter "chainTechLifetimeHours" bzw. dessen Wert ist nicht definiert.';
278
279     msg = [char(paramChain.txtChainName), ': ', msg, ' Berechnungen für diese
Prozesskette werden unterbrochen.'];
280     if startGuiFlag == true
281         BgaPtg2UpdateMessageBox(handles,'apend',msg);
282     else
283         disp(msg);
284     end
285     errorInLoopsFlag = true;     return;
286 end
287
288 % Werte für ersetzbare bauliche Anlagen auf Prozesskettenebene
289 chainBuildInflRate = paramChain.valueChainParam(find(strcmp(paramChain.
txtChainParamInfo(:,1),'chainBuildInflRate')));
290 if isempty(chainBuildInflRate)
291     msg = 'Parameter "chainBuildInflRate" bzw. dessen Wert ist nicht definiert.';
292     msg = [char(paramChain.txtChainName), ': ', msg, ' Berechnungen für diese
Prozesskette werden unterbrochen.'];
293     if startGuiFlag == true
294         BgaPtg2UpdateMessageBox(handles,'apend',msg);
295     else
296         disp(msg);
297     end
298     errorInLoopsFlag = true;     return;
299 end
300
301 chainBuildLifetimeHours = paramChain.valueChainParam(find(strcmp(paramChain.
txtChainParamInfo(:,1),'chainBuildLifetimeHours')));
302 if isempty(chainBuildLifetimeHours)
303     msg = 'Parameter "chainBuildLifetimeHours" bzw. dessen Wert ist nicht definiert.';
304
305     msg = [char(paramChain.txtChainName), ': ', msg, ' Berechnungen für diese
Prozesskette werden unterbrochen.'];
306     if startGuiFlag == true
307         BgaPtg2UpdateMessageBox(handles,'apend',msg);
308     else
309         disp(msg);
310     end
311     errorInLoopsFlag = true;     return;
312 end
313
314 % chainInflRate
315 chainInflRate = paramChain.valueChainParam(find(strcmp(paramChain.txtChainParamInfo(:,
1),'chainInflRate')));
316 if isempty(chainInflRate)
317     msg = 'Parameter "chainInflRate" bzw. dessen Wert ist nicht definiert.';
318     msg = [char(paramChain.txtChainName), ': ', msg, ' Berechnungen für diese
Prozesskette werden unterbrochen.'];
319     if startGuiFlag == true
320         BgaPtg2UpdateMessageBox(handles,'apend',msg);

```

```

319     else
320         disp(msg);
321     end
322     errorInLoopsFlag = true;     return;
323 end
324
325 % Wasserpreis
326 %chainWaterPrice =
paramChain.valueChainParam(find(strcmp(paramChain.txtChainParamInfo(:,1), 'chainWaterPr
ice')));
327
328 % Preis für Nährmedien
329 %chainNutrientPrice =
paramChain.valueChainParam(find(strcmp(paramChain.txtChainParamInfo(:,1), 'chainNutrien
tPrice')));
330
331 % Verkaufserlös Sauerstoff
332 chainO2Revenue = paramChain.valueChainParam(find(strcmp(paramChain.txtChainParamInfo
(:,1), 'chainO2Revenue')));
333 if isempty(chainO2Revenue)
334     msg = 'Parameter "chainO2Revenue" bzw. dessen Wert ist nicht definiert.';
335     msg = [char(paramChain.txtChainName), ': ', msg, ' Berechnungen für diese
Prozesskette werden unterbrochen.'];
336     if startGuiFlag == true
337         BgaPtg2UpdateMessageBox(handles, 'apend', msg);
338     else
339         disp(msg);
340     end
341     errorInLoopsFlag = true;     return;
342 end
343
344
345 % chainElyPers
346 chainElyPers = paramChain.valueChainParam(find(strcmp(paramChain.txtChainParamInfo(:,1
), 'chainElyPers')));
347 if isempty(chainElyPers)
348     msg = 'Parameter "chainElyPers" bzw. dessen Wert ist nicht definiert.';
349     msg = [char(paramChain.txtChainName), ': ', msg, ' Berechnungen für diese
Prozesskette werden unterbrochen.'];
350     if startGuiFlag == true
351         BgaPtg2UpdateMessageBox(handles, 'apend', msg);
352     else
353         disp(msg);
354     end
355     errorInLoopsFlag = true;     return;
356 end
357
358
359
360 % Parameter auf Komponentenebene
361 % Wasserstoffpreis [€/kg H2]
362 compH2Price = paramComp.valueCompParam(find(strcmp(paramComp.txtCompParamInfo(:,1),
'compH2Price')));
363 if isempty(compH2Price)
364     msg = 'Parameter "compH2Price" bzw. dessen Wert ist nicht definiert.';
365     msg = [char(paramChain.txtChainName), ': ', msg, ' Berechnungen für diese
Prozesskette werden unterbrochen.'];
366     if startGuiFlag == true
367         BgaPtg2UpdateMessageBox(handles, 'apend', msg);
368     else
369         disp(msg);
370     end
371     errorInLoopsFlag = true;     return;
372 end
373
374 compInvest1 = paramComp.valueCompParam(find(strcmp(paramComp.txtCompParamInfo(:,1),
'compInvest1')));
375 if isempty(compInvest1)
376     msg = 'Parameter "compInvest1" bzw. dessen Wert ist nicht definiert.';
377     msg = [char(paramChain.txtChainName), ': ', msg, ' Berechnungen für diese
Prozesskette werden unterbrochen.'];
378     if startGuiFlag == true
379         BgaPtg2UpdateMessageBox(handles, 'apend', msg);

```

```

380     else
381         disp(msg);
382     end
383     errorInLoopsFlag = true;     return;
384 end
385
386 %compPower =
387 paramComp.valueCompParam(find(strcmp(paramComp.txtCompParamInfo(:,1), 'compPower')));
388
389 %% Parameter (Ende) *****
390
391 % Dieser Teil ist künstlich, aber konform für Zukunft, wenn die folgenden
392 % Werte nicht mehr Parameter sind, sondern berechnet werden.
393
394 % Konvertierungen der Einheiten
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
409 % Konvertierungen der Parameter
410 % compInflRate = compInflRate/100;
411 % compMaintFactor = compMaintFactor/100;
412 % compUpkeepFactor = compUpkeepFactor/100;
413 % compElecPrice = compElecPrice/100;
414 % comppp = comppp*(10^5);
415 % compDeltap = compDeltap * (10^5);
416 % compT = compT + 273.15;
417 % compConcBioM = compConcBioM/100;
418 % compConcSNG = compConcSNG/100;
419 % compConcCO2 = compConcCO2/100;
420 % compConcH2 = compConcH2/100;
421 % compConcH2S = compConcH2S/(10^6);
422 % compConcH2O = compConcH2O/100;
423 % compConcTraceGas = compConcTraceGas/(10^6);
424 % compConcOther = compConcOther/100;
425 % compDeltaVolFlowBioM = compDeltaVolFlowBioM/100;
426 % compDeltaVolFlowSNG = compDeltaVolFlowSNG/100;
427 % compDeltaVolFlowCO2 = compDeltaVolFlowCO2/100;
428 % compDeltaVolFlowH2 = compDeltaVolFlowH2/100;
429 % compDeltaVolFlowH2S = compDeltaVolFlowH2S/100;
430 % compDeltaVolFlowH2O = compDeltaVolFlowH2O/100;
431 % compDeltaVolFlowTraceGas = compDeltaVolFlowTraceGas/100;
432 % compDeltaVolFlowOther = compDeltaVolFlowOther/100;
433 % compLeakage = compLeakage/100;
434 % compOtherFactor = compOtherFactor/100;
435 % compNutrientVolFlowFactor = compNutrientVolFlowFactor/100;
436 % compActCH2S = compActCH2S/100;
437 % compConvCO2Ideal = compConvCO2Ideal/100;
438 % compConvCO2Real = compConvCO2Real/100;
439
440
441 costChainAnnuityCapitalTotal = 0;
442 costChainAnnuityElecTotal = 0;
443 costChainAnnuityHeatTotal = 0;
444 costChainAnnuityDemandTotal = 0;
445 costChainAnnuityOperationTotal = 0;
446 costChainAnnuityOtherTotal = 0;
447
448
449 for iCnt1 = 1:inParam.nrofComponents
450     costChainAnnuityCapitalTotal = costChainAnnuityCapitalTotal + costCompAnnuities(
451         iCnt1,1);

```

```

451     costChainAnnuityElecTotal      = costChainAnnuityElecTotal + costCompAnnuities(
452     iCnt1,2); % Für Komponente BC gibt es problem
452     costChainAnnuityHeatTotal      = costChainAnnuityHeatTotal + costCompAnnuities(
453     iCnt1,3);
453     costChainAnnuityDemandTotal    = costChainAnnuityDemandTotal + costCompAnnuities(
454     iCnt1,4);
454     costChainAnnuityOperationTotal = costChainAnnuityOperationTotal +
455     costCompAnnuities(iCnt1,5);
455     costChainAnnuityOtherTotal     = costChainAnnuityOtherTotal + costCompAnnuities(
456     iCnt1,6);
456     end;
457
458
459     % Zuweisung an chainCostVec
460     costChainAnnuityTotals = [costChainAnnuityCapitalTotal; costChainAnnuityElecTotal;
461     costChainAnnuityHeatTotal; ...
462     costChainAnnuityDemandTotal; costChainAnnuityOperationTotal;
463     costChainAnnuityOtherTotal];
464
465     % Annuitätsfaktor wird für Berechnung in den Komponenten benötigt, ist für alle
466     % Komponenten gleich
467     if (((1+chainDiscRate)^chainPeriod)-1) == 0
468         msg = 'Variable "chainAnnuityFactor" kann nicht berechnet werden. Division durch
469         null.';
470         msg = [char(paramChain.txtChainName), ': ', msg, ' Berechnungen für diese
471         Prozesskette werden unterbrochen.'];
472         if startGuiFlag == true
473             BgaPtg2UpdateMessageBox(handles,'append',msg);
474         else
475             disp(msg);
476         end
477         errorInLoopsFlag = true;      return;
478     else
479         chainAnnuityFactor = (((1+chainDiscRate)^chainPeriod)*chainDiscRate) ...
480         / (((1+chainDiscRate)^chainPeriod)-1);
481     end;
482
483     % Barwertfaktor auf PK Ebene
484     if (1+chainDiscRate) == 0 || (chainDiscRate-chainInflRate) == 0
485         msg = 'Variable "chainNPVFactor" kann nicht berechnet werden. Division durch
486         null.';
487         msg = [char(paramChain.txtChainName), ': ', msg, ' Berechnungen für diese
488         Prozesskette werden unterbrochen.'];
489         if startGuiFlag == true
490             BgaPtg2UpdateMessageBox(handles,'append',msg);
491         else
492             disp(msg);
493         end
494         errorInLoopsFlag = true;      return;
495     else
496         chainNPVFactor = (1-((1+chainInflRate) / (1+chainDiscRate))^chainPeriod) ...
497         / (chainDiscRate-chainInflRate);
498     end;
499
500     % Anfangsinvestitionen auf PK Ebene
501     chainCapex1 = chainTechInvest + chainBuildInvest;
502
503     % Anfangsausgaben auf PK Ebene erhöht sich um Planungskosten
504     chainCapex = chainCapex1 + chainPlanningApproval;
505
506     % Kopie der Capex-Liste
507     %chainCapexVec = costCompInvest.copy()
508     % Hinzufügen der Capex auf PK-Ebene
509     %chainCapexVec.append(chainCapex)
510
511     % chainCapexVec = zeros(inParam.nrofComponents+1,1);
512     % for iCnt1 = 1:inParam.nrofComponents

```



```

512 % chainCapexVec(iCnt1) = costCompInvest(iCnt1);
513 % end;
514 %
515 % % Hinzufügen der Capex auf PK-Ebene
516 % chainCapexVec(inParam.nrofComponents+1) = chainCapex;
517 % chainCapexVecTotal = sum(chainCapexVec);
518 %
519
520 costInvestChainTotal = sum(costCompInvest) + chainCapex;
521
522 % Nutzungsdauer Gesamtanlagentechnik in Jahren
523 if techChainFLH == 0
524     msg = 'Variable "chainTechLifetimeYears" kann nicht berechnet werden. Division
525     durch null.';
526     msg = [char(paramChain.txtChainName), ': ', msg, ' Berechnungen für diese
527     Prozesskette werden unterbrochen.'];
528     if startGuiFlag == true
529         BgaPtg2UpdateMessageBox(handles,'append',msg);
530     else
531         disp(msg);
532     end
533     errorInLoopsFlag = true; return;
534 else
535     chainTechLifetimeYears = chainTechLifetimeHours/techChainFLH;
536 end;
537
538
539 % Ersatzhäufigkeit
540 % chainTechReplacement = math.ceil(chainPeriod/(chainTechLifetimeYears)-1);
541 if chainTechLifetimeYears == 0
542     msg = 'Variable "chainTechReplacement" kann nicht berechnet werden. Division
543     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
553     chainTechReplacement = ceil(chainPeriod/(chainTechLifetimeYears)-1);
554 end;
555
556 % Ersatzbeschaffung auf PK-Ebene
557 if chainTechReplacement == 0
558     chainTechReplNPV1 = 0;
559     chainTechReplNPV2 = 0;
560
561 elseif chainTechReplacement == 1
562
563     if ((1+chainDiscRate)^(1*chainTechLifetimeYears)) == 0
564         msg = 'Variable "chainTechReplNPV1" kann nicht berechnet werden. Division
565         durch null.';
566         msg = [char(paramChain.txtChainName), ': ', msg, ' Berechnungen für diese
567         Prozesskette werden unterbrochen.'];
568         if startGuiFlag == true
569             BgaPtg2UpdateMessageBox(handles,'append',msg);
570         else
571             disp(msg);
572         end
573         errorInLoopsFlag = true; return;
574     else
575         chainTechReplNPV1 = chainTechInvest * ((1+chainTechInflRate) ...
576         ^ (1*chainTechLifetimeYears)) / ((1+chainDiscRate)^(1*chainTechLifetimeYears));
577     end;
578
579     chainTechReplNPV2 = 0;

```

```

579
580 elseif chainTechReplacement == 2
581     chainTechReplNPV1 = 0;
582
583     if ((1+chainDiscRate)^(2*chainTechLifetimeYears)) == 0
584         msg = 'Variable "chainTechReplNPV2" kann nicht berechnet werden. Division
                    durch null.';
585         msg = [char(paramChain.txtChainName), ': ', msg, ' Berechnungen für diese
                    Prozesskette werden unterbrochen.'];
586         if startGuiFlag == true
587             BgaPtg2UpdateMessageBox(handles,'apend',msg);
588         else
589             disp(msg);
590         end
591         errorInLoopsFlag = true;         return;
592
593     else
594         chainTechReplNPV2 = chainTechInvest * ((1+chainTechInflRate)^(2*
                    chainTechLifetimeYears)) ...
595         /((1+chainDiscRate)^(2*chainTechLifetimeYears));
596     end;
597
598
599
600 else
601     % In diesem Fall wird in BgaPtg2Calc.m Fehler erzeugt und das Programm beendet.
602     chainTechReplNPV1 = 0;
603     chainTechReplNPV2 = 0;
604     inParam.chainTechReplacement = chainTechReplacement;
605     errorInLoopsFlag = true;
606     % print("Fehler! Es wird von max. 2 Ersatzbeschaffungen ausgegangen. Überprüfe
                    Lebensdauer und FLH der Gesamtanlagentechnik");
607 end;
608
609 % Restwert, residual or salvage value
610 if chainTechLifetimeYears == 0 || (1+chainDiscRate)^chainPeriod == 0
611     msg = 'Variable "chainResidualTech" kann nicht berechnet werden. Division durch
                    null.';
612     msg = [char(paramChain.txtChainName), ': ', msg, ' Berechnungen für diese
                    Prozesskette werden unterbrochen.'];
613     if startGuiFlag == true
614         BgaPtg2UpdateMessageBox(handles,'apend',msg);
615     else
616         disp(msg);
617     end
618     errorInLoopsFlag = true;         return;
619
620 else
621     chainResidualTech = chainTechInvest * ((1+chainTechInflRate)^(chainTechReplacement
                    * ...
622     chainTechLifetimeYears)) * (((chainTechReplacement+1)*chainTechLifetimeYears ...
623     - chainPeriod)/chainTechLifetimeYears) * (1/(1+chainDiscRate)^chainPeriod);
624 end;
625
626
627
628 % Nutzungsdauer Baulicher Anlagen in Jahren
629 if techChainFLH == 0
630     msg = 'Variable "chainBuildLifetimeYears" kann nicht berechnet werden. Division
                    durch null.';
631     msg = [char(paramChain.txtChainName), ': ', msg, ' Berechnungen für diese
                    Prozesskette werden unterbrochen.'];
632     if startGuiFlag == true
633         BgaPtg2UpdateMessageBox(handles,'apend',msg);
634     else
635         disp(msg);
636     end
637     errorInLoopsFlag = true;         return;
638
639 else
640     chainBuildLifetimeYears = chainBuildLifetimeHours/techChainFLH;
641 end;
642

```

```

643
644
645 % Ersatzhäufigkeit
646 % chainBuildReplacement = math.ceil(chainPeriod/(chainBuildLifetimeYears)-1);
647 if chainBuildLifetimeYears == 0
648     msg = 'Variable "chainBuildReplacement" kann nicht berechnet werden. Division
        durch null.';
649     msg = [char(paramChain.txtChainName), ': ', msg, ' Berechnungen für diese
        Prozesskette werden unterbrochen.'];
650     if startGuiFlag == true
651         BgaPtg2UpdateMessageBox(handles,'apend',msg);
652     else
653         disp(msg);
654     end
655     errorInLoopsFlag = true;     return;
656
657 else
658     chainBuildReplacement = ceil(chainPeriod/(chainBuildLifetimeYears)-1);
659 end;
660
661
662 % 2. Ersatzbeschaffung auf PK-Ebene (mit unterschiedlichen Annahmen)
663 if chainBuildReplacement == 0
664     chainBuildReplNPV1 = 0;
665     chainBuildReplNPV2 = 0;
666
667 elseif chainBuildReplacement == 1
668     if ((1+chainDiscRate)^(1*chainBuildLifetimeYears)) == 0
669         msg = 'Variable "chainBuildReplNPV1" kann nicht berechnet werden. Division
        durch null.';
670         msg = [char(paramChain.txtChainName), ': ', msg, ' Berechnungen für diese
        Prozesskette werden unterbrochen.'];
671         if startGuiFlag == true
672             BgaPtg2UpdateMessageBox(handles,'apend',msg);
673         else
674             disp(msg);
675         end
676         errorInLoopsFlag = true;     return;
677
678     else
679         chainBuildReplNPV1 = chainBuildInvest * ((1+chainBuildInflRate) ...
680             ^((1*chainBuildLifetimeYears)) / ((1+chainDiscRate)^(1*chainBuildLifetimeYears
681             )));
682     end;
683
684     chainBuildReplNPV2 = 0;
685
686 elseif chainBuildReplacement == 2
687     chainBuildReplNPV1 = 0;
688     if ((1+chainDiscRate)^(2*chainBuildLifetimeYears)) == 0
689         msg = 'Variable "chainBuildReplNPV2" kann nicht berechnet werden. Division
        durch null.';
690         msg = [char(paramChain.txtChainName), ': ', msg, ' Berechnungen für diese
        Prozesskette werden unterbrochen.'];
691         if startGuiFlag == true
692             BgaPtg2UpdateMessageBox(handles,'apend',msg);
693         else
694             disp(msg);
695         end
696         errorInLoopsFlag = true;     return;
697
698     else
699         chainBuildReplNPV2 = chainBuildInvest * ((1+chainBuildInflRate)^(2*
        chainBuildLifetimeYears)) ...
700         /((1+chainDiscRate)^(2*chainBuildLifetimeYears));
701     end;
702
703 else
704     % In diesem Fall wird in BgaPtg2Calc.m Fehler erzeugt und das Programm beendet.
705     chainTechReplNPV1 = 0;
706     chainTechReplNPV2 = 0;
707     inParam.chainTechReplacement = chainTechReplacement;
708     errorInLoopsFlag = true;

```

```

708     % print("Fehler! Es wird von max. 2 Ersatzbeschaffungen ausgegangen. Überprüfe
       Lebensdauer und FLH der Gesamtanlagentechnik");
709 end;
710
711
712 % Restwert, residual or salvage value
713 if chainBuildLifetimeYears == 0 || (1+chainDiscRate)^chainPeriod == 0
714     msg = 'Variable "chainResidualBuild" kann nicht berechnet werden. Division durch
       null.';
715     msg = [char(paramChain.txtChainName), ': ', msg, ' Berechnungen für diese
       Prozesskette werden unterbrochen.'];
716     if startGuiFlag == true
717         BgaPtg2UpdateMessageBox(handles,'apend',msg);
718     else
719         disp(msg);
720     end
721     errorInLoopsFlag = true;     return;
722
723 else
724     chainResidualBuild = chainBuildInvest * ((1+chainBuildInflRate)^(
       chainBuildReplacement* ...
725     chainBuildLifetimeYears)) * (((chainBuildReplacement+1)*chainBuildLifetimeYears
       ...
726     - chainPeriod)/chainBuildLifetimeYears) * (1/(1+chainDiscRate)^chainPeriod);
727 end;
728
729
730
731
732 % Annuität der kapitalgebundenen Kosten
733 % Gesamt-CAPEX + Barwerte der Ersatzbeschaffungen abzüglich Restwert, * AnFaktor
734 costChainAnnuityCapital = ...
735 (chainCapex + chainTechReplNPV1 + chainTechReplNPV2 + chainBuildReplNPV1 ...
736 + chainBuildReplNPV2 - chainResidualTech - chainResidualBuild) ...
737 * chainAnnuityFactor;
738
739 % Stromkosten auf PK-Ebene
740 chainCostElec = techChainElecDemand * chainElecPrice;
741
742 % Annuität Strom
743 costChainAnnuityElec = chainCostElec * chainAnnuityFactor * chainNPVFactor;
744
745 % Biogas Kosten/Erträge
746 chainCostBiogas = techChainVolFlowPG * chainBGPurchasePrice ...
747 - techChainEnergyBioM * chainBMSellingPrice;
748
749 % Annuität der Wärme
750 % costChainAnnuityHeat = (techChainHeatDemand * chainHeatPurchasePrice ...
751 % - techChainHeatTotal_real * chainHeatSellingPrice) ...
752 % * chainAnnuityFactor * chainNPVFactor;
753 costChainAnnuityHeat = techChainHeatDemand * chainHeatPurchasePrice
       ...
754     * chainAnnuityFactor * chainNPVFactor;
755
756
757 % Annuität bedarfsgebundene Kosten
758 costChainAnnuityDemand = chainCostBiogas * chainAnnuityFactor * chainNPVFactor;
759
760 % Annuität betriebsgebundene Kosten (Summe aus Komponenten + Betriebskosten\
761 % auf PK Ebene + Personalkosten)
762 costChainAnnuityOperation = (chainOperationFactor ...
763 * chainCapex1 + chainCostPers) * chainAnnuityFactor * chainNPVFactor; %
       Wartung/Instandhaltung für Tech+Build gleich, sonst extra Variable
764
765 % Annuiätät sonstiger Kosten (Versicherung, Rückstellungen etc.)
766 costChainAnnuityOther = (chainCostOther - chainO2Revenue ...
767 + (chainInsurFactor + chainRenatFactor) * costInvestChainTotal) ...
768 * chainAnnuityFactor * chainNPVFactor;
769
770 % Zusammenfassung der Annuitäten auf PK-Ebene
771 costChainAnnuities = [costChainAnnuityCapital; costChainAnnuityElec;
       costChainAnnuityHeat; ...
772 costChainAnnuityDemand; costChainAnnuityOperation; costChainAnnuityOther];

```

```

773
774 costChainAnnuity = sum(costChainAnnuities);
775
776 % Ergänzung der costChainAnnuityTotals um Annuität der PK Ebene
777
778 for iCnt1 = 1:inParam.nrOfCostAnnuities
779     costChainAnnuityTotals(iCnt1) = costChainAnnuityTotals(iCnt1) + costChainAnnuities
780     (iCnt1);
781 end;
782
783 % Gesamtannuität einer Prozesskette
784 costChainAnnuityTotal = sum(costChainAnnuityTotals);
785
786
787 % Klären
788 if strcmp(paramComp.sheet,'D') && compH2Price == 0
789     % Personalkosten für Ely
790     corrPers = chainElyPers;
791     % Versicherung für Ely
792     corrInsur = techChainMethPower*compInvest1*chainInsurFactor;
793     corrSum = (corrPers+corrInsur)*chainNPVFactor*chainAnnuityFactor;
794 else
795     corrSum = 0;
796 end;
797
798 % Annuität der Wasserstoffkosten
799 costChainAnnuityHydrogen = sum(costCompAnnuities(1,:)) + corrSum;
800
801
802 % Wasserstoffpreis LCOH
803 if HHV_kg_H2 == 0 || techChainMethPower*techChainFLH == 0
804     msg = 'Variable "costHydrogenLCOH" kann nicht berechnet werden. Division durch
805     null.';
806     msg = [char(paramChain.txtChainName), ': ', msg, ' Berechnungen für diese
807     Prozesskette werden unterbrochen.'];
808     if startGuiFlag == true
809         BgaPtg2UpdateMessageBox(handles,'apend',msg);
810     else
811         disp(msg);
812     end
813     errorInLoopsFlag = true;    return;
814 else
815     costHydrogenLCOH = costChainAnnuityHydrogen/(techChainMethPower*1000*techChainFLH/
816     HHV_kg_H2);
817 end;
818
819 for iCnt1 = 1:inParam.nrOfCostAnnuities
820     if costChainAnnuityTotal == 0
821         msg = 'Variable "costChainAnnuityTotalRel(iCnt1)" kann nicht berechnet
822         werden. Division durch null.';
823         msg = [char(paramChain.txtChainName), ': ', msg, ' Berechnungen für diese
824         Prozesskette werden unterbrochen.'];
825         if startGuiFlag == true
826             BgaPtg2UpdateMessageBox(handles,'apend',msg);
827         else
828             disp(msg);
829         end
830         errorInLoopsFlag = true;    return;
831     else
832         costChainAnnuityTotalRel(iCnt1) = costChainAnnuityTotals(iCnt1)/
833         costChainAnnuityTotal*100;
834     end;
835 end;
836
837 if costChainAnnuityTotal == 0
838     msg = 'Variable "costChainAnnuityTotalRel(7)" kann nicht berechnet werden.
839     Division durch null.';
840     msg = [char(paramChain.txtChainName), ': ', msg, ' Berechnungen für diese
841     Prozesskette werden unterbrochen.'];
842     if startGuiFlag == true

```

```

837         BgaPtg2UpdateMessageBox(handles,'apend',msg);
838     else
839         disp(msg);
840     end
841     errorInLoopsFlag = true;     return;
842
843 else
844     costChainAnnuityTotalRel(7) = costChainAnnuityHydrogen/costChainAnnuityTotal*100;
845 end;
846
847
848
849 for iCnt1 = 1:size(costCompAnnuities,1)
850     costCompAnnuity(iCnt1) = sum(costCompAnnuities(iCnt1,:));
851
852     if costChainAnnuityTotal == 0
853         msg = 'Variable "costCompAnnuityRel(iCnt1)" kann nicht berechnet werden.
854             Division durch null.';
855         msg = [char(paramChain.txtChainName), ': ', msg, ' Berechnungen für diese
856             Prozesskette werden unterbrochen.'];
857         if startGuiFlag == true
858             BgaPtg2UpdateMessageBox(handles,'apend',msg);
859         else
860             disp(msg);
861         end
862         errorInLoopsFlag = true;     return;
863     else
864         costCompAnnuityRel(iCnt1) = sum(costCompAnnuities(iCnt1,:))/
865             costChainAnnuityTotal*100;
866     end;
867
868
869
870
871
872

```