D priedas. Ekserginių rodiklių skaičiavimas

Ši programa parašyta programiniam paketui Matlab, skirta sistemos eksserginių rodiklių skaičiavimui

```
function pab = Input1(Vt,A,N)
%""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""
%                   Įvesties duomenys
%""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""
[ta,troom,G,SK_gain,t1,t2,M1,t3,RH,ts,t6,t7,M2,t10,t11,tp,M4,t8,t9,M3]     =im-
portfile1('ResultOutput.csv',1, 43803);

% Vt = 2;                                    % talpos tūris m3
% A=10;
ro = 1000;                                       % šilumnešio tankis
[kg/m3]
cp = 4.187;                                  % šilumnešio specifinė
savitoji šiluma [kJ/(kg*n)]
cp_oro = 1.005;                                     % [kJ/(kg K)] oro
savitoji specifinė šiluma
L_tiek = 200;
L_sal = 200;
t1v = ta;
t3v = troom;
eta_he = 0.9;
eta_aux = 1.0;
PPDa = 10;                                   % aktualus PPD, kuris
yra II kategorijos (=10 %)
CLO =1;
MET = 70;
VEL = 0.15;

%""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""
%                   Skaičiavimo pradžia
%""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""
[PMV_val, PPD_val] = PMV (troom,ts, CLO, MET,VEL,RH);

% saulės kolektorių srautai
[EG_plus, L_solar, eta_sol] = EG (A, G, SK_gain, t1, t2, M1, ta, ro, cp);
[EM1, LM1] = E_M1 (M1, ro, ta, troom);

% Akumuliacin4s talpos
```

```
[eta_t, L_accum] = etat (t1, t2, t3, t6, t7, t8, t9, t10, t11, M1, M2, M3, M4, Vt, ta, cp, ro);

% išorinio šaltinio srautai
[Ex_plus, L_aux, Q_aux, eta_auxx]= Ex (t6, t7, ta, M2, eta_aux, cp);
[EM2, LM2] = E_M2 (M2, ro, ta, troom);

% vėdinimo įrenginio eksergijos srautai
[E1v, E4v, Ev1, Lk, Lv1, eta_v] = E_1v (t8, t9, ta, troom, M3, ro, cp, cp_oro, L_tiek, L_sal, eta_he);
Ev2 = Ev1;
Lv2 = Lv1;
L_sil = Lvhe (t8, t9, ta, troom, M3, ro, cp, cp_oro, L_tiek, L_sal, eta_he);        % šilumokaičio eksergijos nuostoliai
[EM3, LM3] = E_M3 (M3, ro, ta, troom);

% paviršinio šildymo srautai
[Eh, Lh] =E_h (cp, t10, t11, tp, M4, ta);                                % grindinio šildymo suteiktas eksergijos srautas patalpai, bet iš sistemos išeina ribų
[EM4, LM4] = E_M4 (M4, ro, ta, troom);                                   % cirkuliacinios siurblio eksergijos nuostoliai

% saulės frakcija
SK_gain_size=size(SK_gain);
for i=1:1:SK_gain_size
   if SK_gain(i,1) > 0 && Q_aux(i,1) > 0
    f_sol = sum(SK_gain)./sum(Q_aux);
   else
      f_sol = 0;
   end
end

% ekserginis efektyvumas kiekvienam mėnesiui
% ////////////////////Mėnesinis//////////////////////////////////////////////////

load('time.mat')
   eta_sol(eta_sol==0)=NaN;
   eta_t(eta_t==0)=NaN;
   eta_auxx(eta_auxx==0)=NaN;
   eta_v(eta_v==0)=NaN;
```

```
for k=1:1:12
   EG_sum (k,1) =sum(EG_plus(time(k,1):time(k,2)));
   L_solar_sum(k,1) =sum(L_solar(time(k,1):time(k,2)));
   EM1_sum(k,1) =sum(EM1(time(k,1):time(k,2)));
   LM1_sum(k,1) =sum(LM1(time(k,1):time(k,2)));
   L_accum_sum(k,1) =sum(L_accum(time(k,1):time(k,2)));
   Ex_plus_sum(k,1) =sum(Ex_plus(time(k,1):time(k,2)));
   L_aux_sum(k,1) =sum(L_aux(time(k,1):time(k,2)));
   EM2_sum(k,1) =sum(EM2(time(k,1):time(k,2)));
   LM2_sum(k,1) =sum(LM2(time(k,1):time(k,2)));
   LM2_sum(k,1) =sum(LM2(time(k,1):time(k,2)));
   Q_aux_sum(k,1) =sum(Q_aux(time(k,1):time(k,2)));
   Ev1_sum(k,1) =sum(Ev1(time(k,1):time(k,2)));
   Ev2_sum(k,1) =sum(Ev2(time(k,1):time(k,2)));
   E1v_sum(k,1) =sum(E1v(time(k,1):time(k,2)));
   EM3_sum(k,1) =sum(EM3(time(k,1):time(k,2)));
   E4v_sum(k,1) =sum(E4v(time(k,1):time(k,2)));
   L_sil_sum(k,1) =sum(L_sil(time(k,1):time(k,2)));
   Lk_sum(k,1) =sum(Lk(time(k,1):time(k,2)));
   Lv1_sum(k,1) =sum(Lv1(time(k,1):time(k,2)));
   Lv2_sum(k,1) =sum(Lv2(time(k,1):time(k,2)));
   LM3_sum(k,1) =sum(LM3(time(k,1):time(k,2)));
   Eh_sum(k,1) =sum(Eh(time(k,1):time(k,2)));
   Lh_sum(k,1) =sum(Lh(time(k,1):time(k,2)));
   EM4_sum(k,1) =sum(EM4(time(k,1):time(k,2)));
   LM4_sum(k,1) =sum(LM4(time(k,1):time(k,2)));
   SK_gain_sum(k,1) =sum(SK_gain(time(k,1):time(k,2)));
   Q_aux_sum(k,1) =sum(Q_aux(time(k,1):time(k,2)));
   PMV_mean(k,1) =mean(PMV_val(time(k,1):time(k,2)));
   PPD_mean(k,1) =mean(PPD_val(time(k,1):time(k,2)));
   eta_sol_sum(k,1) = nanmean(eta_sol(time(k,1):time(k,2)));
   eta_t_sum(k,1) = nanmean(eta_t(time(k,1):time(k,2)));
   eta_auxx_sum(k,1) = nanmean(eta_auxx(time(k,1):time(k,2)));
   eta_v_sum(k,1) = nanmean(eta_v(time(k,1):time(k,2)));
end

for k=1:1:12
   Wf(k,1) = PPD_mean(k,1)/PPDa;
   Wf_sum(k,1) = Wf(k,1)*(time(k,2)-time(k,1));
   h_PPD(k,1)  = ((time(k,2)-time(k,1))-Wf_sum(k,1))/24;
end
```

```
   % mėn saulės frakcija
for k=1:1:12
   if SK_gain_sum(k,1) > 0 && Q_aux_sum(k,1) > 0
    f_sol(k,1) = SK_gain_sum(k,1)/Q_aux_sum(k,1);
   else
    f_sol(k,1) = 0;
   end
end

% mėn Eksergijos kiekis
for k=1:1:12
   E_con(k,1) =
EG_sum(k,1)+EM1_sum(k,1)+EM2_sum(k,1)+EM3_sum(k,1)+EM4_sum(k,1)
+Ev1_sum(k,1)+Ev2_sum(k,1)+Ex_plus_sum(k,1);
   L_loss(k,1) =
L_solar_sum(k,1)+LM1_sum(k,1)+LM2_sum(k,1)+LM3_sum(k,1)+LM4_sum(
k,1)+Lv1_sum(k,1)+Lv2_sum(k,1)+L_aux_sum(k,1)+L_sil_sum(k,1)+Lk_sum(
k,1);
end

% L_accum_sum(k,1);
%+E1v_sum(k,1)
%E4v_sum(k,1)
% mėn efektyvumas
for k=1:1:12
if 1-L_loss(k,1)/E_con(k,1) >0;
   eta(k,1) = 1-L_loss(k,1)/E_con(k,1);
else
   eta(k,1) = 0;
end
end

%>>>>>>>>>>>>>>>>>>>>>>>Sezoniniai paramet-
rai<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
% šildymo sezonas nuo 10 mėn. iki 5 mėn. (5 mėnuo neiskaitomas)

% saulės frakcija

   SK_gain_sez = sum(SK_gain_sum(1:4))+sum(SK_gain_sum(10:12));
   Q_aux_sez = sum(Q_aux_sum(1:4))+sum(Q_aux_sum(10:12));
```

```
    f_sol_sez = SK_gain_sez/Q_aux_sez;

% Sezoninis eksergijos kiekiai
    E_con_sez = sum(E_con(1:4))+sum(E_con(10:12));
    L_loss_sez = sum(L_loss(1:4))+sum(L_loss(10:12));

% sezoninis ekserginis efektyvumas
if 1-L_loss_sez/E_con_sez >0
    eta_sez = 1-L_loss_sez/E_con_sez;
else
    eta_sez = 0;
end

%""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""
%                  Duomenų įrašymas
%""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""

count = N;

xlsfilename = 'D:\knygos\Juozo Bielskaus studijos\Doktorantūra\4 metai\mattrn-sys\Rezulatas122223.xlsx'; %failo issaugijimo vieta
E=actxserver ('Excel.Application');
File=xlsfilename;

if ~exist(File,'file')
    ExcelWorkbook = E.workbooks.Add;
    ExcelWorkbook.SaveAs(File);
    ExcelWorkbook.Close(false);
end

invoke(E.Workbooks,'Open',File);

xlswrite1(xlsfilename, EG_sum','EG_sum',num2str(count));
xlswrite1(xlsfilename, f_sol','f_sol',num2str(count));
xlswrite1(xlsfilename, E_con','E_con',num2str(count));
xlswrite1(xlsfilename, L_loss','L_loss',num2str(count));
xlswrite1(xlsfilename, eta','eta',num2str(count));
xlswrite1(xlsfilename, eta_sol_sum','eta_sol_sum',num2str(count));
xlswrite1(xlsfilename, eta_t_sum','eta_t_sum',num2str(count));
xlswrite1(xlsfilename, eta_auxx_sum','eta_auxx_sum',num2str(count));
xlswrite1(xlsfilename, eta_v_sum','eta_v_sum',num2str(count));
```

```
xlswrite1(xlsfilename, eta_t_sum','eta_t_sum',num2str(count));
xlswrite1(xlsfilename, Q_aux_sum','Q_aux_sum',num2str(count));
xlswrite1(xlsfilename, SK_gain_sum','SK_gain_sum',num2str(count));
xlswrite1(xlsfilename, SK_gain_sez','SK_gain_sez',num2str(count));
xlswrite1(xlsfilename, Q_aux_sez',' Q_aux_sez',num2str(count));
xlswrite1(xlsfilename, f_sol_sez','f_sol_sez',num2str(count));
xlswrite1(xlsfilename, E_con_sez',' E_con_sez',num2str(count));
xlswrite1(xlsfilename, L_loss_sez',' L_loss_sez',num2str(count));
xlswrite1(xlsfilename, eta_sez',' eta_sez',num2str(count));
xlswrite1(xlsfilename, PPD_mean','PPD_mean',num2str(count));
xlswrite1(xlsfilename, Wf','Wf',num2str(count));
xlswrite1(xlsfilename, Wf_sum','Wf_sum',num2str(count));
xlswrite1(xlsfilename, h_PPD','h_PPD',num2str(count));

invoke(E.ActiveWorkbook,'Save');
E.Quit
E.delete
clear E ExcelWorkbook xlsfilename File

pab=1;
end
```