```
clearvars
close all;
startTime=clock;
displayOn=true;
writeGraphs=true;
writeResults=false;

addpath 'D:\Users\gebruiker\Documents\Technische Bestuurskunde\SEPAM jaar 2
(2016-2017)\- Master Thesis Project (Q3)\Matlab Model'

%% Parameter settings

cd 'D:\Users\gebruiker\Documents\Technische Bestuurskunde\SEPAM jaar 2 (2016-
2017)\- Master Thesis Project (Q3)\Matlab Model'

clear('c')
clear('fid')
clear('name')
cd ..

load('elecdemand_BSH2034.mat');
load('heatdemand_BSH2034.mat');
load('solarnorm_NL.mat');
load('windonnorm_NL.mat');
load('APXprice.mat');
load('elecimportprice.mat');

range=1:8760;

%% Input data

 % % % % % % % % % %
% % % Scenarios % % %
 % % % % % % % % % %

 for loopNo = 1:10                                                         %
Here choose the number of replications
    % Household efficiency list
    EnergyEfficientBuilding_full_elec =  ((1.00*0.80)+(0.00*1.00));
    EnergyEfficientBuilding_med_elec  =  ((0.50*0.80)+(0.50*1.00));        %
Here 50% of households are built energy efficient
    EnergyEfficientBuilding_none_elec =  ((0.00*0.80)+(1.00*1.00));

    EnergyEfficientBuilding_full_gas =   ((1.00*0.60)+(0.00*1.00));
    EnergyEfficientBuilding_med_gas  =   ((0.50*0.60)+(0.50*1.00));        %
Here 50% of households are built energy efficient
    EnergyEfficientBuilding_none_gas =   ((0.00*0.60)+(1.00*1.00));

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
     % % % % % % % % % % % % %
    % % % Scenario set-up % % %
     % % % % % % % % % % % % %

    % ICES TECHNOLOGIES
    IT1 = 2;                    % IT 1: Energy efficient buildings
(none/medium/full)
    IT2 = 2;                    % IT 2: Household RES
(low/medium/high)
    IT3 = 2;                    % IT 3: Community RES
(low/medium/high)
    IT4 = 2;                    % IT 4: Heat pumps
(none/medium/full)
```

```
    % UNCERTAIN FUTURE DEVELOPMENTS
    UFD1 = 2;                   % UFD 1: Electricity demand in 2034
(30% lower/stable/30% higher)
    UFD2 = 1;                   % UFD 2: ICES technology capital costs in 2034
(stable/15% lower/30% lower)
    UFD3 = 1;                   % UFD 3: APX electricity price in 2034
(stable/20% lower)
    UFD4 = 1;                   % UFD 4: Natural gas price in 2034
(stable/50% higher)


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

    % IT 1: Energy efficient buildings (None/Medium/Full)
    elecdemand = elecdemand_BSH2034;
    heatdemand = heatdemand_BSH2034;
    numberofenergyeffbuildings = 0;

    if IT1 == 1 % none
        numberofenergyeffbuildings = 0;
        elecdemand(range) = elecdemand(range)*EnergyEfficientBuilding_none_elec;
% Electricity demand is lower when more buildings are energy efficient
        heatdemand(range) = heatdemand(range)*EnergyEfficientBuilding_none_gas;
% Gas demand is lower when more buildings are energy efficient
    end
    if IT1 == 2 % medium
        numberofenergyeffbuildings = 2000;
        elecdemand(range) = elecdemand(range)*EnergyEfficientBuilding_med_elec;
% Electricity demand is lower when more buildings are energy efficient
        heatdemand(range) = heatdemand(range)*EnergyEfficientBuilding_med_gas;
% Gas demand is lower when more buildings are energy efficient
    end
    if IT1 == 3 % full
        numberofenergyeffbuildings = 4000;
        elecdemand(range) = elecdemand(range)*EnergyEfficientBuilding_full_gas;
% Electricity demand is lower when more buildings are energy efficient
        heatdemand(range) = heatdemand(range)*EnergyEfficientBuilding_full_gas;
% Gas demand is lower when more buildings are energy efficient
    end

    % IT 2: Household RES
    if IT2 == 1 % low                                                      %
Installed capacity (KW) of solar energy (maximum capacity of solar panels in BSH
2034 is 10329 kW)
        solarCap = 2000;
    end
    if IT2 == 2 % medium
        solarCap = 5000;
    end
    if IT2 == 3 % high
        solarCap = 8000;
    end

    % IT 3: Community RES
    if IT3 == 1 % low                                                      %
Installed capacity (KW) of wind energy, 1500 KW equals one wind turbine
        windonCap = 2000;
    end
    if IT3 == 2 % medium
        windonCap = 5000;
    end
    if IT3 == 3 % full
        windonCap = 8000;
```

```
    end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
     % % % % % % % % % %
    % % % Parameters % % %
     % % % % % % % % % %

    windonnorm = windonnorm_NL;                                         %
Renamed
    solarnorm = solarnorm_NL;                                           %
Renamed
    renewablenorm = windonnorm + solarnorm;

    elecexportprice = APXprice*0.001;                                   %
Hourly APX price (EUR/MWh) transform into hourly electricity price (EUR/kWh)
    stableelecexportprice = APXprice*0.001;
    elecimportprice = elecimportprice*1;                                %
Pre-created matrix with taxes over apx to form retail price
    gasprice = (0.63/36)*1000;                                          %
Average gas price (EUR/m3) of 2017 in the Netherlands times 36 because
    variablecostsolar = 0;                                              %
Variable electricity generation costs (EUR/kWh)
    variablecostwind = 0.002;                                           %
Variable electricity generation costs (EUR/kWh)
    OMWind = windonCap*5;                                               %
Average O&M costs of wind turbines per kw per year (EUR/kW/year)
    OMSolar = solarCap*6;                                               %
Average O&M costs of solar panels per kw per year (EUR/kW/year)
    CO2PricePerTon = 7;                                                 % CO2
price in EUR/ton CO2
    maxlinecap = 15000;                                                 % Max
KW over the lines for electricity to be imported or exported in an hour
    SolarEURperKW = 1769;
    WindEURperKW = 1446*0.666667;
    CapCostPerHeatPump = 9000;
    costofenergyeffbuilding = 5000;

    %% Forming renewable generation matrices

    if(displayOn==true), disp('Forming renewable generation matrices'); end

    D_wind=zeros(1,8760);
    D_solar=zeros(1,8760);

    D_wind(range) = windonnorm(range)*windonCap;                        % The
wind generation matrix every hour (BSH 2034) in KW
    D_solar(range) = solarnorm(range)*solarCap;                         % The
solar generation matrix every hour (BSH 2034) in KW

    D_renewable(range) = D_wind(range) + D_solar(range);                %
Renewable generation matrix every hour (BSH 2034) in KW

    %% Forming energy demand matrices

    if(displayOn==true), disp('Forming energy demand matrices'); end

    D_elecdemand(range) = elecdemand(range);                            % The
demand matrix every hour (BSH 2034)
    for i=1:8760
        D_elecdemand(1,i) = normrnd(1,0.25)*D_elecdemand(1,i);
    end
```

```
    D_heatdemand = heatdemand(range);                                      % The
demand matrix every hour (BSH 2034)
    for i=1:8760
        D_heatdemand(1,i) = normrnd(1,0.25)*D_heatdemand(1,i);
    end

    % IT 4: Heat pumps
    halfmatrix = ones(1,8760)*0.5;
    D_gasdemand = zeros(1,8760);
    D_heatpumpdemand = zeros(1,8760);
    CapCostHeatPumps = 0;
    OMHeatPump = 0;

    if IT4 == 1 % none
        D_gasdemand = D_heatdemand;
    end
    if IT4 == 2 % medium
        D_gasdemand(range) = D_heatdemand(range) .* halfmatrix(range);
        D_heatpumpdemand(range) = D_heatdemand(range) .* halfmatrix(range);
        CapCostHeatPumps = 2000 * CapCostPerHeatPump;
        OMHeatPump = 2000*193;
    end
    if IT4 == 3 % full
        D_heatpumpdemand = D_heatdemand;
        CapCostHeatPumps = 4000 * CapCostPerHeatPump;
        OMHeatPump = 2000*193;
    end

    D_GJheatpumptokWelec = ones(1,8760) * 54.3;
    D_heatpumpelecdemand = D_heatpumpdemand(range) .* D_GJheatpumptokWelec;
    D_elecdemand(range) = D_elecdemand(range) + D_heatpumpelecdemand(range);

    %% Uncertain future developments

    % UFD 1: Electricity demand in 2034
    D_30percentlower = ones(1,8760) * 0.70;
    D_30percenthigher = ones(1,8760) * 1.30;

    if UFD1 == 1 % 30% lower
        D_elecdemand(range) = D_elecdemand(range) .* D_30percentlower;
    end
    if UFD1 == 2 % stable
        D_elecdemand(range) = D_elecdemand(range);
    end
    if UFD1 == 3 % 30% higher
        D_elecdemand(range) = D_elecdemand(range) .* D_30percenthigher;
    end

    % UFD 2: ICES technology capital costs in 2034

    if UFD2 == 1 % stable
    SolarEURperKW = SolarEURperKW;
    WindEURperKW = WindEURperKW;
    CapCostHeatPumps = CapCostHeatPumps;
    end
    if UFD2 == 2 % 15% lower
    SolarEURperKW = SolarEURperKW*0.85;
    WindEURperKW = WindEURperKW*0.85;
    CapCostHeatPumps = CapCostHeatPumps*0.85;
    costofenergyeffbuilding = costofenergyeffbuilding*0.85;
    end
    if UFD2 == 3 % 30% lower
    SolarEURperKW = SolarEURperKW*0.7;
    WindEURperKW = WindEURperKW*0.7;
```

```
    CapCostHeatPumps = CapCostHeatPumps*0.7;
    costofenergyeffbuilding = costofenergyeffbuilding*0.7;
    end

    % UFD 3: APX electricity price in 2034
    rndfactorapx = normrnd(1.12,0.02);
    lowerelecpricefactor = windonnorm + solarnorm;
    D_onepointtwelve = ones(1,8760)*rndfactorapx;                       %
With these settings, average elecexportprice is 20.4 % lower
    D_onepointzerotwo = ones(1,8760)* normrnd(1,0.05);
    if UFD3 == 1 % stable
        %elecexportprice = elecexportprice(range) .* D_onepointtwelve;
        elecimportprice = elecimportprice * normrnd(1.00,0.05);
    end
    if UFD3 == 2 % 20% lower
        lowerelecpricefactor(lowerelecpricefactor<0.4) = 0;
        lowerelecpricefactor(range) = D_onepointtwelve -
lowerelecpricefactor(range);
        lowerelecpricefactor(lowerelecpricefactor==rndfactorapx) = 1;
        lowerelecpricefactor(lowerelecpricefactor==0) = 1;
        lowerelecpricefactor(lowerelecpricefactor==1) = 0.95;
        elecexportprice = elecexportprice(range) .* lowerelecpricefactor(range);
    end

    % UFD 4: Natural gas price in 2034
    if UFD4 == 1 % stable
    gasprice = gasprice;
    end
    if UFD4 == 2 % 50% higher
    gasprice = gasprice*1.5;
    end


    %% Exchanging electricity with the national grid

    if(displayOn==true), disp('Exchanging electricity with the national grid');
end

    D_exchanged=zeros(1,8760);

    D_exchanged(range)=D_elecdemand(range)-D_renewable(range);          % Can
have positive or negative values: when positive,
                                                                        %
this means that more electricity is imported from the
                                                                        %
national grid than given back to the national grid
    D_exchanged(D_exchanged>maxlinecap) = maxlinecap;
    D_exchanged(D_exchanged<-maxlinecap) = -maxlinecap;

    %% Key performance indicators

    if(displayOn==true), disp('Forming key performance indicators'); end

    % 1. CO2 Emission

    D_KGCO2perkWh = ones(1,8760)*0.526;

    D_KGCO2perGJheat = ones(1,8760)*1.884*36;                           %
1.884 is kg per m3 gas, so per GJ heat this is times 36

    D_CO2EmissionsImported = D_exchanged(range).*D_KGCO2perkWh(range);

    D_CO2EmissionsImported(D_CO2EmissionsImported<0) = 0;
```

```
    D_CO2EmissionsGas=D_gasdemand(range).*D_KGCO2perGJheat(range);

    D_CO2EmissionsGas(D_CO2EmissionsGas<0) = 0;

    TotalCO2FromElec = sum(sum(D_CO2EmissionsImported));

    TotalCO2FromGas = sum(sum(D_CO2EmissionsGas));

    TotalCO2Emissions(loopNo) = (TotalCO2FromElec + TotalCO2FromGas)/1000;
% Going from kg to ton: divide by 1000

    % 2. Total energy demand

    TotalElecDemand = (sum(sum(D_elecdemand)))*0.0036;                %
From kWh elec to GJ is times 0.0036 (1 kWh = 3.6 MJ)

    TotalGasDemand = (sum(sum(D_gasdemand)));                         % Is
already in GJ

    TotalEnergyDemand = TotalElecDemand + TotalGasDemand;

    TotalEnergyDemandPerHousehold(loopNo) = TotalEnergyDemand/4000;

    % 3. Cost of energy supply

    D_exported = D_exchanged(range);

    D_exported(D_exported>0) = 0;

    minus = ones(1,8760)*-1;

    D_exported = D_exported(range).*minus(range);

    D_imported = D_exchanged(range);

    D_imported(D_imported<0) = 0;

    D_ExportRevenue = D_exported(range).*elecexportprice(range);

    D_ImportCosts = D_imported(range).*elecimportprice(range);

    D_CostExchangedElec = D_ImportCosts(range) - D_ExportRevenue(range);      %
Matrices of imported and exported electricity times the APX electricity
price/retail electricity price

    TotalCostExchangedElec = sum(sum(D_CostExchangedElec));                   %
Total costs or benefits over the year 2034 from importing or exporting
electricity

    TotalVariableCostGeneration = (sum(sum(D_wind))*variablecostwind) +
(sum(sum(D_solar))*variablecostsolar); % Cost of generated electricity

    D_CostGasConsumption = D_gasdemand * gasprice;

    TotalCostGasConsumption =  sum(sum(D_CostGasConsumption));

    CO2Price(loopNo) = TotalCO2Emissions(loopNo) * CO2PricePerTon;

    CostOfEnergySupply(loopNo) = TotalCostExchangedElec +
TotalVariableCostGeneration + TotalCostGasConsumption + OMWind + OMSolar +
OMHeatPump + CO2Price(loopNo);

    CostOfEnergySupplyPerHousehold(loopNo) = CostOfEnergySupply(loopNo)/4000;
```

```
    %%%% Without ICES?
    D_CostElecConsumption = D_elecdemand.*elecimportprice(range);

    TotalCostElecConsumption =  sum(sum(D_CostElecConsumption));

    SavedCostOfEnergySupply2034 = 4156879 - CostOfEnergySupply;           %
4.156878525882619e+06 is the 2034 CostOfEnergySupply when 0 renewables, no
energy efficient buildings

    % 4. Capital costs

    CapCostSolar = solarCap*SolarEURperKW;                                 %
Capital cost of solar panel is 4600 per 2600 Wp, so 4600/2.6=1769 EUR/kWp

    CapCostWind = windonCap*WindEURperKW;                                  %
Capital cost is 1446 EUR per KW of installed wind turbine

    CapCostEnergyEfficientBuildings =
numberofenergyeffbuildings*costofenergyeffbuilding;

    TotalCapCost(loopNo) = CapCostSolar + CapCostWind + CapCostHeatPumps +
CapCostEnergyEfficientBuildings;

    CapCostHouseholdRES = CapCostSolar/4000;                                   %
Household RES capital cost per household

    CapCostCommunityRES = CapCostWind;                                         %
Total capital cost for community RES

    % 5. Payback Time
    YearsToPayback(loopNo) = TotalCapCost/SavedCostOfEnergySupply2034;

    % 6. Self-sufficiency (=percentage renewable?)

    D_PartElecDemandCovByRES = D_renewable(range)./D_elecdemand(range);

    D_PartElecDemandCovByRES(D_PartElecDemandCovByRES>1) = 1;

    PartElecDemandCovByRES = ((sum(sum(D_PartElecDemandCovByRES)))/8760);

    GJElecDemandCovByRES = PartElecDemandCovByRES * TotalElecDemand;

    GJGasDemandCovByRES = 0 * TotalGasDemand;

    PercentageEnergyDemandCovByRES(loopNo) = ((GJElecDemandCovByRES +
GJGasDemandCovByRES) / TotalEnergyDemand)*100;

    % 66. Self-consumption

    TotalGJElecExported(loopNo) = sum(sum(D_exported))*0.0036;
%From kWh to GJ

    TotalRenewableGenerated(loopNo) = sum(sum(D_renewable))*0.0036;

    SelfConsumption(loopNo) = 100-
((TotalGJElecExported(loopNo)/TotalRenewableGenerated(loopNo))*100);

    % 7. Exchange with national grid

    TotalElecExported = sum(sum(D_exported))*0.0036;

    TotalElecImported = sum(sum(D_imported))*0.0036;
```

```
    TotalGJElecExported(loopNo) = sum(sum(D_exported))*0.0036;
%From kWh to GJ

    % 8. Max Line Capacity

    MaximumLineCaps = zeros(1,2);
    MaximumLineCaps(1,1) = max(D_imported);
    MaximumLineCaps(1,2) = max(D_exported);
    MaximumLineCap(loopNo) = max(MaximumLineCaps);

    %Dec/jan/feb (winter)
    %1:1460, 8030:8760

    %Maart/april/mei (spring)
    %1461:3650

    %Juni Juli Aug (summer)
    %3651: 5840

    %Sept Okt Nov (autumn)
    %5841: 8030

    TotalGJElecExportedWinter = sum(sum(D_exported(1:1460)))*0.0036 +
sum(sum(D_exported(8031:8760)))*0.0036;
    TotalGJElecExportedSpring = sum(sum(D_exported(1461:3650)))*0.0036;
    TotalGJElecExportedSummer = sum(sum(D_exported(3651:5840)))*0.0036;
    TotalGJElecExportedAutumn = sum(sum(D_exported(5841:8030)))*0.0036;

    TotalGJElecImported = sum(sum(D_imported))*0.0036;

    TotalGJElecImportedWinter = sum(sum(D_imported(1:1460)))*0.0036 +
sum(sum(D_imported(8031:8760)))*0.0036;
    TotalGJElecImportedSpring = sum(sum(D_imported(1461:3650)))*0.0036;
    TotalGJElecImportedSummer = sum(sum(D_imported(3651:5840)))*0.0036;
    TotalGJElecImportedAutumn = sum(sum(D_imported(5841:8030)))*0.0036;

 end

if(displayOn==true), disp('Done'); end

%% Results

%CO2 emission
KPI_CO2EMISSION = mean(TotalCO2Emissions)

%Self-sufficiency
KPI_SELFSUFFICIENCY = mean(PercentageEnergyDemandCovByRES)

%Total energy demand per household [GJ/household/year]
KPI_TOTALENERGYDEMAND = mean(TotalEnergyDemandPerHousehold)

%Total renewable energy exchanged to the central grid
KPI_ENERGYEXCHANGED = mean(TotalGJElecExported)

%Self-consumption
KPI_SELFCONSUMPTION = mean(SelfConsumption)

%Maximum line capacity [kWh]
KPI_MAXLINECAP = mean(MaximumLineCap)

%Capital costs of the ICES components
KPI_TOTALCAPCOST = mean(TotalCapCost)

%Yearly cost of operation of the ICES of Buiksloterham per household
```

```
KPI_YEARLYCOSTOFOPERATION = mean(CostOfEnergySupplyPerHousehold)

%Payback time of ICES related capital costs
KPI_PAYBACKTIME = mean(YearsToPayback)

if(writeGraphs==true)

    close all;

    fig=gcf;
    figpos=get(fig,'Position');
    figpos(3)=figpos(3)*1.5;
    figpos(4)=figpos(4)*1.5;
    set(fig,'Position',figpos)

  %  p1=area([hydroROR nuclearOutput coalOutput gasOutput hydroOut windOutput
solarOutput Import LOLOutput]);
  %  set(p1(6),'FaceColor',[0 0.498039215803146 0]);
  %  set(p1(7),'FaceColor',[1 0.8 0]);
   % set(p1(8),'FaceColor',[0.847058832645416 0.160784319043159 0]);

    D_elecdemand_Daily = zeros(length(D_elecdemand)/24,1);
    D_imported_Daily = zeros(length(D_imported)/24,1);
    D_exported_Daily = zeros(length(D_exported)/24,1);
    D_solar_Daily = zeros(length(D_solar)/24,1);
    D_wind_Daily = zeros(length(D_solar)/24,1);

    for i = 0:length(D_elecdemand_Daily)-1
        D_elecdemand_Daily(i+1) = sum(D_elecdemand(i*24+1:(i+1)*24));
        D_imported_Daily(i+1) = sum(D_imported(i*24+1:(i+1)*24));
        D_exported_Daily(i+1) = sum(D_exported(i*24+1:(i+1)*24));
        D_solar_Daily(i+1) = sum(D_solar(i*24+1:(i+1)*24));
        D_wind_Daily(i+1) = sum(D_wind(i*24+1:(i+1)*24));
    end
    hold on
    demandplot=plot(D_elecdemand_Daily,'-','Color',[0 0 1],'LineWidth',1.5);
    demandplot2=plot(D_imported_Daily,'-','Color',[1 0 0],'LineWidth',1.5);
    demandplot3=plot(D_exported_Daily,'-','Color',[0 0.6 0],'LineWidth',1.5);
    %demandplot=plot(stableelecexportprice_wk26,'-','Color',[0 0
1],'LineWidth',1.5);
    %demandplot=plot(elecexportprice_wk26,'-','Color',[1 0 0],'LineWidth',1.5);
    %demandplot2=plot(D_wind,'-','Color',[1 0 0],'LineWidth',1.5);
    set(gca,'ygrid','on');
    set(gca,'xgrid','on');
    set(gca,'Ylim',[0 140000]);
    ylabel('Power (kW)')
    xlabel('Time (days)')

    %genNames{53}='Wind';
    lh=legend([],...
        'Electricity demand','Imported electricity','Exported electricity');
    set(lh,'FontSize',9);
    set(lh,'Orientation','Vertical','Location','SouthEast');

    cd 'D:\Users\gebruiker\Documents\Technische Bestuurskunde\SEPAM jaar 2
(2016-2017)\- Master Thesis Project (Q3)\Matlab Model\Figures'
    name=strcat('BSHmodelv2.eps');
    print('-depsc', name);
    cd ..
end

if(writeResults==true)
%    diary('results-3,3,3,3_2,1,1,1.txt')
```

```
  cd 'D:\Users\gebruiker\Documents\Technische Bestuurskunde\SEPAM jaar 2 (2016-
2017)\- Master Thesis Project (Q3)\Matlab Model'
   name=strcat('Results_','33332111','.mat');
   save(name,'-regexp',
'^(?!(TotalGJElecExported|YearsToPayback|PercentageElecDemandCovByRES)$).')
%    cd ..
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```