Computational Economics: Problem Set 1

Yangming Bao, ID: 5601239

Cheung Ying Lun, ID: 5441897

Problem 1: Simple linear equation example

1. Substituting (1a) into (1b), we get

$$a - b \cdot q = c + d \cdot q \implies b \cdot q + d \cdot q - (a - c) = 0. \tag{1}$$

2. Solving Eq.(1) for q,

$$q = \frac{a-c}{b+d}. \tag{2}$$

Substituting into (1a),

$$p = a - b \cdot \frac{a-c}{b+d}. \tag{3}$$

3. Rearranging terms in the system,

$$\begin{cases} p + b \cdot q = a \\ p - d \cdot q = b \end{cases} \implies \underbrace{\begin{pmatrix} 1 & b \\ 1 & -d \end{pmatrix}}_{A} \underbrace{\begin{pmatrix} p \\ q \end{pmatrix}}_{x} = \underbrace{\begin{pmatrix} a \\ c \end{pmatrix}}_{y}. \tag{4}$$

Applying the LU decomposition,

(a) Define $L \cdot U = A$, where

$$L = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \qquad U = \begin{pmatrix} 1 & b \\ 1 & -d \end{pmatrix}. \tag{5}$$

Subtracting the first row from the second row,

$$L = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}, \qquad U = \begin{pmatrix} 1 & b \\ 0 & -d-b \end{pmatrix}. \tag{6}$$

(b) Solving $L \cdot b = y$,

$$\begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} b_1 \\ b_2 \end{pmatrix} = \begin{pmatrix} a \\ c \end{pmatrix} \Longrightarrow \begin{cases} b_1 = a \\ b_1 + b_2 = c \end{cases} \Longrightarrow \begin{cases} b_1 = a \\ b_2 = c - a \end{cases} \tag{7}$$

(c) Solving $U \cdot x = b$,

$$\begin{pmatrix} 1 & b \\ 0 & -d-b \end{pmatrix} \begin{pmatrix} p \\ q \end{pmatrix} = \begin{pmatrix} a \\ c-a \end{pmatrix} \Longrightarrow \begin{cases} p = a - b \cdot \dfrac{a-c}{b+d} \\ q = \dfrac{a-c}{b+d} \end{cases} . \tag{8}$$

4. Substituting $a = 3, b = 0.5, c = d = 1$, we obtain

$$\begin{cases} p^* = a - b \cdot \dfrac{a-c}{b+d} = \dfrac{4}{3} \\ q^* = \dfrac{a-c}{b+d} = \dfrac{7}{3} \end{cases} . \tag{9}$$

5. The iteration method converges with order $(\mathbf{D}, \mathbf{S})'$ but not the other way round. Convergence and non-convergence are illustrated graphically in Figure 1 and 2 respectively.

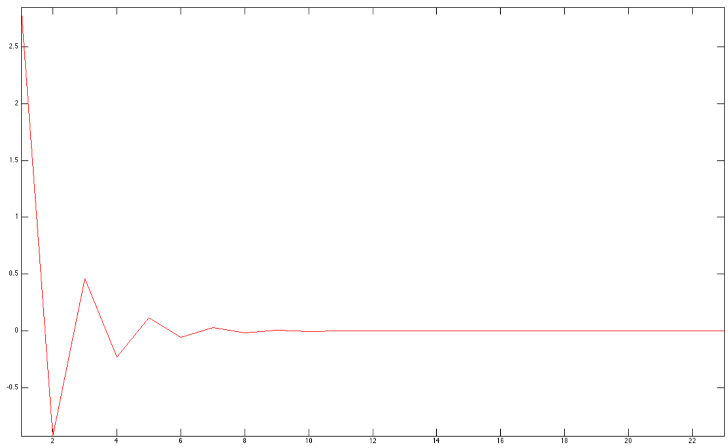

Figure 1: Convergence Case

6. The system does not converge for all of the λ chosen.

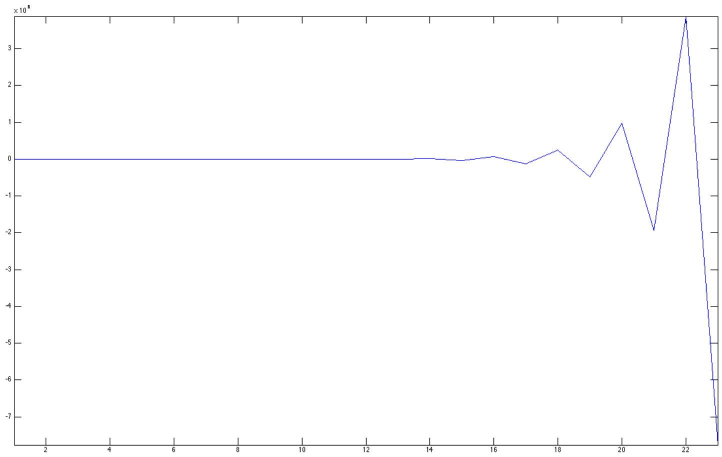

Figure 2: Non-Convergence Case

Problem 2: Determine the output gap

See `PS1_Q2.m` for the calculation results. $\log Y_{j,t}$ and $G_{j,t}$ are plotted in Figure 3 and 4 respectively.

Problem 3: Schelling's Segregation

The simulated dynamics of residential area are shown below. The white spaces represent houses occupied by white people; black spaces represent houses occupied by black; and grey spaces represent empty houses. Figure 5 to 8 show the distribution of the black and white people after 0 to 45 moves.

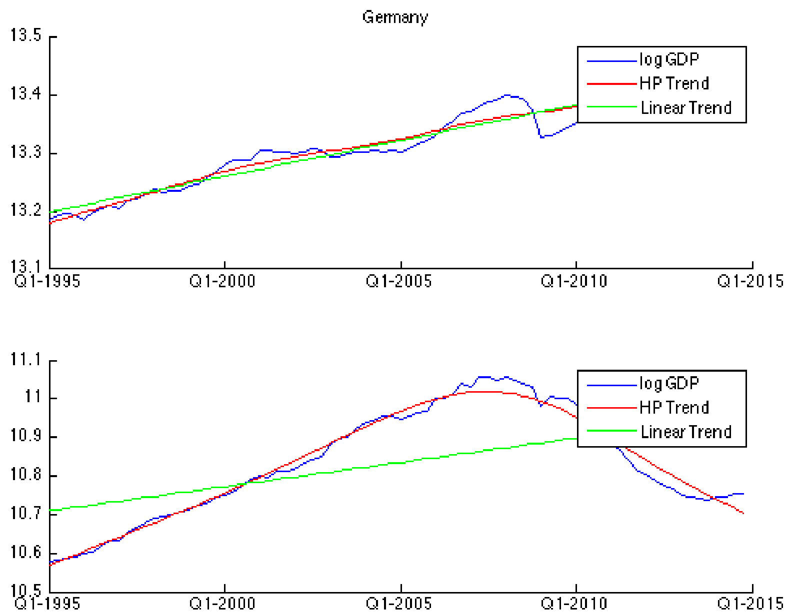

Figure 3: $\log Y_{j,t}$

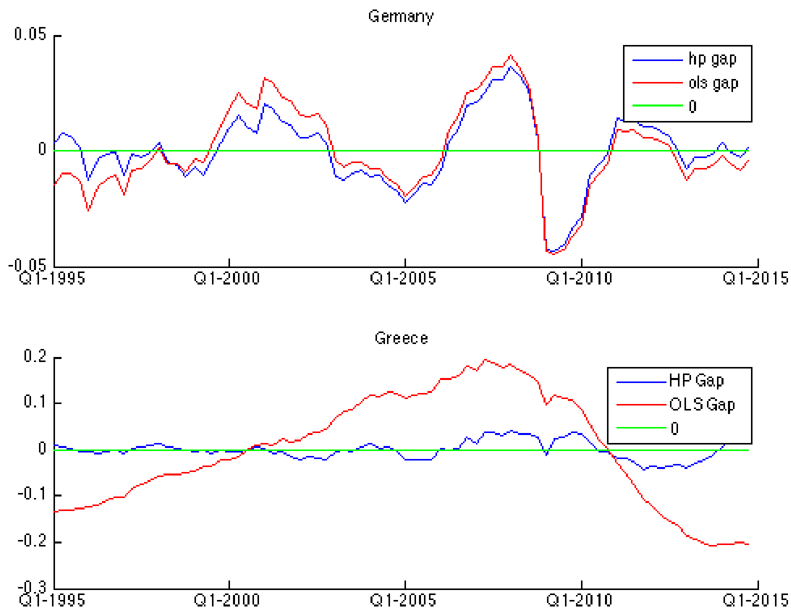

Figure 4: $G_{j,t}$

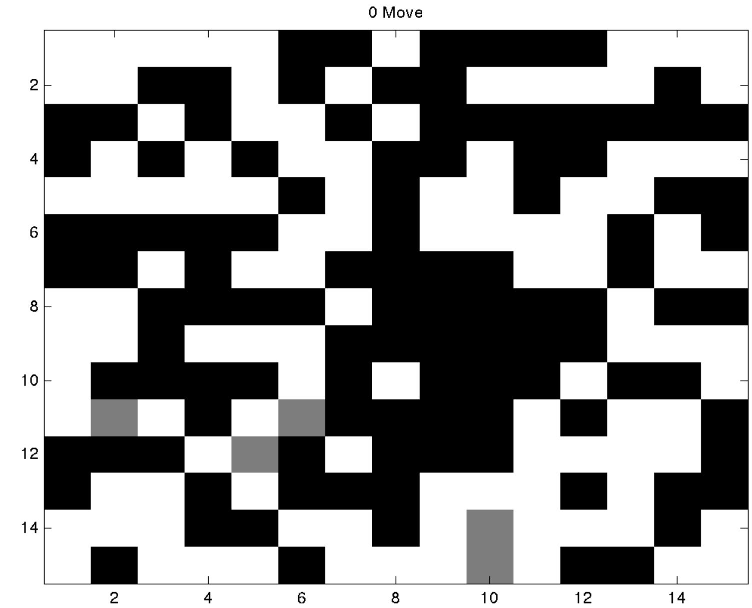

Figure 5: 0 Move

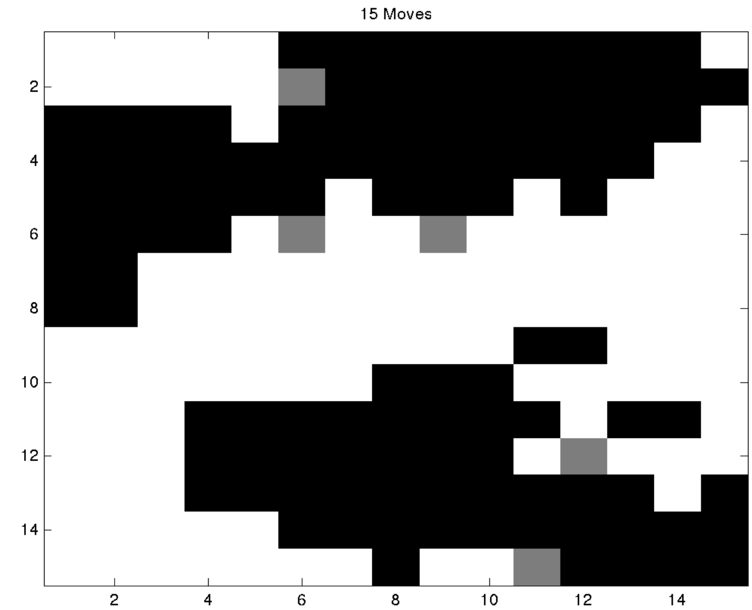

Figure 6: 15 Moves

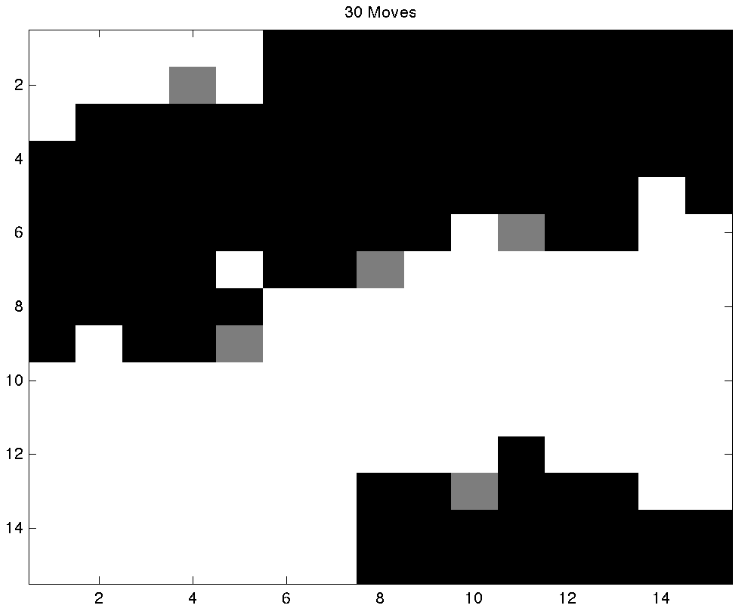

Figure 7: 30 Moves

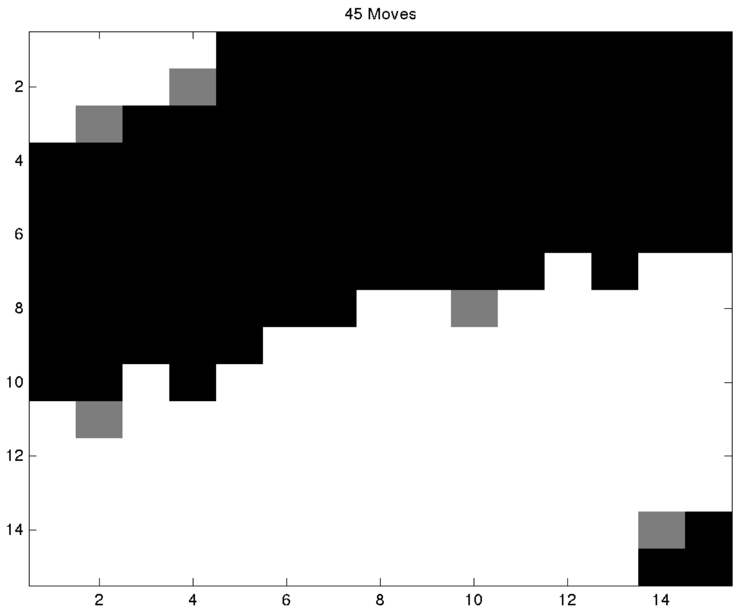

Figure 8: 45 Moves

Appendix: Code

Question 1

```
% Computational Economics
% PS1 - Q1

clear, clc
close all
format long

disp('------------------------------------------------------------')
disp('Problem 1')
disp('------------------------------------------------------------')
disp(' ')


%% 1.5

% first order
A = [1 0.5;1 -1];
y = [3 1]';
% second order
A2 = [1 -1; 1 0.5];
y2 = [1 3]';

max_it = 1000;
tol=1e-6;
rng('default');
Q = tril(A);
Q2 = tril(A2);
x_init = [0.1,0.1]';
x=x_init;

% convegence case; order 1
for it=1:max_it
    it;
    dx1(:,it)=Q\(y-A*x);
    x=x+dx1(:,it);
    if norm(dx1(:,it))<tol
    disp(['Converged at iteration #: ', num2str(it)])
```

```
    disp(['The solution vector is: ', num2str(x')])
    disp(' ')
    break
    end
    if it>=max_it, disp('No Convergence'), end
end

% nonconvergence case; order 2
x=x_init;
for it=1:max_it
    it;
    dx2(:,it)=Q2\(y2-A2*x);
    x=x+dx2(:,it);
    if norm(dx2(:,it))<tol
    disp(['Converged at iteration #: ', num2str(it)])
    disp(['The solution vector is: ', num2str(x')])
    disp(' ')
    break
    end
    if it>=max_it, disp('No Convergence'), end
end

%plot
figure1 = figure('name','convergence case');
plot(dx1(1,:),'r','linewidth',1.2);

figure2 = figure('name','nonconvergence case');
plot(dx2(1,:),'b','linewidth',1.2);


%% 1.6
lambda = linspace(0.1,0.9,9)';
telapsed = zeros(1,length(lambda));
x=x_init;
for k=1:length(lambda)
    tstart = tic;
    for it=1:max_it
        dx3=lambda(k)*Q2\(y2-A2*x);
        x=x+dx3;
        if norm(dx3)<tol
            disp(['Converged at iteration #: ', num2str(it)])
```

```
            disp(['The solution vector is: ', num2str(x')])
            disp(' ')
            break
        end
        if it>=max_it, disp('No Convergence'), end
    end
    telapsed(k) = toc(tstart);
end
disp(['fastest lambda for convergence is: ', num2str(min(telapsed))])
```

Question 2

```
% Computational Economics
% PS1 - Q2 Determine the output gap

clear, clc

%import data
[data,txt] = xlsread('data/OECD-Germany_Greece_GDP_Linux.xls');
%% 2.1

logGDP_Germany = log(data(1,:))';
logGDP_Greece = log(data(2,:))';

%% 2.2

trend_Germany = hpfilter(logGDP_Germany,1600);
trend_Greece = hpfilter(logGDP_Greece,1600);

%% 2.3

t = datenum(txt,'QQ-YYYY');
X = [ones(length(t),1) t];
beta_Germany = (X'*X)\(X'*logGDP_Germany);
beta_Greece = (X'*X)\(X'*logGDP_Greece);

logGDP_Germany_hat = X*beta_Germany;
logGDP_Greece_hat = X*beta_Greece;

%% 2.4

```

```
hp_gap_Germany =  exp(logGDP_Germany-trend_Germany);
hp_gap_Greece =  exp(logGDP_Greece-trend_Greece);
ols_gap_Germany = exp(logGDP_Germany-logGDP_Germany_hat);
ols_gap_Greece = exp(logGDP_Greece-logGDP_Greece_hat);

%% 2.5

figure(1)
subplot(2,1,1)
title('Germany')
hold on
plot(t,logGDP_Germany,'-b');
plot(t,trend_Germany,'-r');
plot(t,logGDP_Germany_hat,'-g')
hold off
datetick('x','qq-yyyy')
legend('log GDP','HP Trend','Linear Trend')


subplot(2,1,2)
hold on
plot(t,logGDP_Greece,'-b');
plot(t,trend_Greece,'-r');
plot(t,logGDP_Greece_hat,'-g')
hold off
datetick('x','qq-yyyy')
legend('log GDP','HP Trend','Linear Trend')

figure(2)
subplot(2,1,1)
title('Germany')
hold on
plot(t,log(hp_gap_Germany),'-b');
plot(t,log(ols_gap_Germany),'-r');
plot(t,zeros(length(t),1),'-g');
hold off
datetick('x','qq-yyyy')
legend('hp gap','ols gap','0')

subplot(2,1,2)
hold on
```

```
title('Greece')
plot(t,log(hp_gap_Greece),'-b');
plot(t,log(ols_gap_Greece),'-r');
plot(t,zeros(length(t),1),'-g')
hold off
datetick('x','qq-yyyy')
legend('HP Gap','OLS Gap','0')
```

Question 3

```
% Computational Economics
% PS1 - Q3 Schelling's Segregation

close all
clear, clc
rng('default')

To_Plot = [15,30,45];

%% Initialization

%  0 - non-occupied
%  1 - black people
% -1 - white people
res_area = zeros(15);
res_area(randperm(225,110)) = 1;
remaining_space = find(res_area-1);
res_area(remaining_space(randperm(115,110))) = -1;
clear remaining_space
figure('name','0 Move')
imagesc(res_area), colormap(flipud(gray));
title('0 Move')
print('PS1_Q3_0Move.png','-dpng');

%% Move!

for iMove = 1:max(To_Plot)

    % Suround the residential area with empty houses
    full_res_area = [zeros(1,17);
        zeros(15,1), res_area, zeros(15,1);
```

```
        zeros(1,17)];
    diff_color = zeros(15);

    % Count if each neighbour is of different color
    for iRow = -1:1
        for iCol = -1:1
            if ~(iRow==1 && iCol==1)
                diff_color = diff_color + ...
                    (full_res_area(2+iRow:16+iRow,2+iCol:16+iCol)==-
                        res_area);
            end
        end
    end

    clear full_res_area

    % Find empty house
    empty_house = find(res_area==0);
    diff_color(empty_house)=0;
    move = (diff_color > 8*0.35);
    empty_house = [empty_house;find(move)];

    count_moving_total = sum(sum(move));
    count_moving_white = sum(sum((res_area(move)==1)));
    count_moving_black = count_moving_total-count_moving_white;
    % Randomly assign house, first to black and then to white people
    move_perm = randperm(length(empty_house),count_moving_total);

    % Move out
    res_area(empty_house) = 0;
    % Move in
    res_area(empty_house(move_perm(1:count_moving_white))) = 1;
    res_area(empty_house(move_perm(count_moving_white+1:count_moving_total)
        )) = -1;
    if ismember(iMove,To_Plot)
        figure('name',[num2str(iMove), ' Move'])
        imagesc(res_area), colormap(flipud(gray));
        title([num2str(iMove), ' Moves'])
        print(['PS1_Q3_',num2str(iMove),'Move.png'],'-dpng');
    end
end
```