# Homework 2

Mihai Pasnicu

September 26, 2018

**Problem 1.**

The solutions to our problem are $D_A = D_B = 0.42$

**Problem 2.**

The starting values for our problem are $p_A = 1$ and $p_B = 1$. It converges in 5 steps. The resulting equilibrium price was $p_A = p_B = 1.598942$, expected to be symmetric.

**Problem 3.**

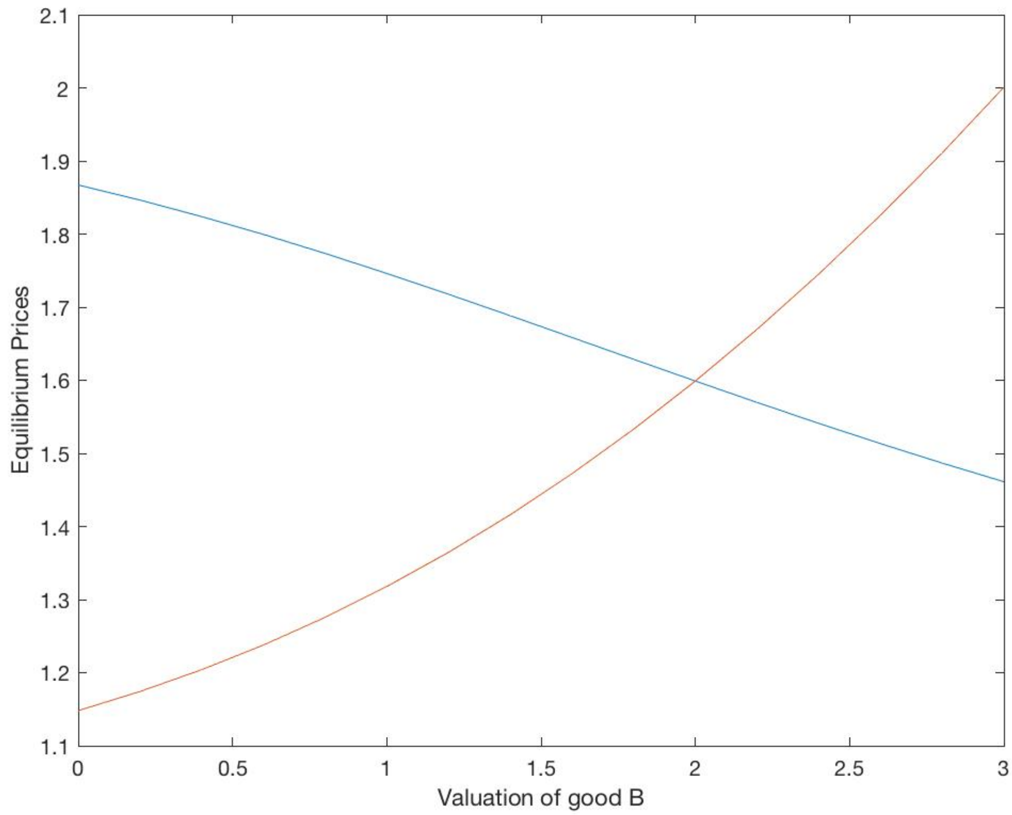
The Gauss - Siedel method converges in 6 steps. While the time I obtain always changes, after a bunch of runs, Gauss - Siedel is faster.

**Problem 4.**

Yes, it converges in 36 steps. Again, the time seems to vary, but it seems to be faster than both of the above methods.

**Problem 5.**

We chose Broyden's method and we obtained the following graph.




Code:

```
diary hw2_diary.out

%% problem 1
% just define the values

v = [2 2];
p = [1 1];

fprintf('Demand for A is %.2f and Demand for B is %.2f\n', demand(v, p), dema



%% problem 2
% keep v_a = v_b = 2
```

```
p = [1  1];
f = @(x) [x(1) - 1/( 1 - demand(v, x)); x(2) - 1/(1 - demand(fliplr(v), flipl


tic
p_sol = broyden(f, p);
toc




%% problem 3


pold = [1  1];
pnew = [2  2];

tol = 1e-8;
maxit = 100;

tic
for iter =1:maxit
    fprintf('iter %d: p(1) = %f, p(2) = %f\n', iter, pnew(1), pnew(2));
    faVal = f(pnew);
    fbVal = f(fliplr(pnew));

    if abs(max(faVal,fbVal)) < tol
        break
    else

    % updating pa
    g=@(pa) f([pa, pnew(2)]);
    gold=g(pold(1));
    gVal = g(pnew(1));
    paNew = pnew(1) - ((pnew(1) - pold(1)) / (gVal - gold) )* gVal;
    pold(1) = pnew(1);
    pnew(1) = paNew;

    % updating pb
    g=@(pb) f([pnew(1), pb]);
    gold=g(pold(2));
    gVal = g(pnew(2));
    pbNew = pnew(2) - ((pnew(2) - pold(2)) / (gVal - gold) )* gVal;
    pold(2) = pnew(2);
    pnew(2) = pbNew;
```

```
    end
end
toc

%% problem 4

p_4 = [1  1];

tol = 1e-8;
maxit = 100;

tic
for iter =1:maxit
    fprintf('iter %d: p(1) = %f, p(2) = %f\n', iter, p_4(1), p_4(2));
    faVal = f(p_4);
    fbVal = f(fliplr(p_4));

    if abs(max(faVal,fbVal)) < tol
        break
    else

    % updating pa
    p_4(1) = 1/(1 - demand(v, p_4));

    % updating pb
    p_4(2) = 1/(1 - demand(fliplr(v), fliplr(p_4)));

    end
end
toc


%%  problem 5
% we are going to use broyden for this

va = 2;
vb = 0 : 0.2 :3;
p_5 = ones(2, size(vb, 2));

for i = 1 : size((vb), 2)
    v = [va, vb(i)];
    f = @(x) [x(1) - 1/( 1 - demand(v, x)); x(2) - 1/(1 - demand(fliplr(v), f
    %p_5(:, i) = broyden(f, p_5(:, i)');
end
```

```
plot(vb,p_5(1,:),vb,p_5(2,:));
xlabel('Valuation of good B'); ylabel('Equilibrium Prices');

diary off
```