

```

%% Name of file: "Hansen_Spulber_code_Jan_20112_1.m"

% This Matlab code generates all simulation results reported in "Does Cost
% Uncertainty in the Bertrand Model Soften Competition" by Johan N. M.
% Lagerlöf. The latest version of the paper, the code and other material
% are available at www.johanlagerlof.org.

clear all; % clean up the memory

% The following code creates Fig 2a in the paper. The code closely follows the
% one in Fibich and Gavish (2011, online appendix), appropriately adapted
% for this particular problem.

%% Construct grid for c2
h=1/1000; % set value of grid spacing h
c2=(h:h:1-h)'; % a column vector
nGrid=numel(c2); % Number of grid points = nGrid

%% Construct a finite-difference approximation operator
e=ones(nGrid,1); % 2nd order approximation matrix % e is a column vector
D=spdiags([-e 0*e e], -1:1, nGrid, nGrid)/2/h;

Dc=D;Dc_RHS=e*0; % Handle boundary c(1)=1 for Dc
Dc_RHS(end)=1/2/h;

Dp=D;Dp_RHS=e*0;
Dp_RHS(end)=1/2/h; % Handle boundary p(1)=1 for Dp
Dp(1,1:3)=[-3 4 -1]/2/h; % Handle one-sided boundary p(0)=? for Dp

%% Construct initial guess
c1=c2; % a column vector
p=(1+2*c2)/3; % a column vector

%% Specify the parameters in the distribution functions
x1=1; % Firm 1's efficiency parameter
x2=12; % Firm 2's efficiency parameter

%% Compute solution using iterations
for iter=1:50
% Calculate v^(k+1)
c_coeff=(x2.*(e-c1).*(e-2.*p+c2))./(x1.*(p-c2).*(e-c2).*(e-2*p+c1));
LHS_c=Dc+spdiags(c_coeff,0,nGrid,nGrid);
RHS_c=p.*c_coeff-Dc_RHS;
c1=LHS_c\RHS_c;

% Calculate b^(k+1)
p_coeff=(x2.*(e-p))./((e-c2).*(e-2*p+c1));
LHS_p=Dp-spdiags(p_coeff,0,nGrid,nGrid);
RHS_p=-c1.*p_coeff-Dp_RHS;
p=LHS_p\RHS_p;
end

p2 = p;

%% Below follows some code that is not adapted from Fibich and Gavish (2011).

```

```

% Construct a vector with firm 1's prices, p1. Above we treated firm 1's
% price as exogenous, and solved for the values of firm 1's cost and firm 2's price
% that were consistent with equilibrium. Now we need to "invert" these
% results. This is done by, for each c value on the grid, identify the element
% in the c1 vector that is closest in value to that grid point. The index
% of that element of the c1 vector will be denoted I. We can then pick the
% I'th element in the p vector solved for above and put that element in our
% newly constructed vector p1.

for i=1:nGrid
    Z = abs(c1 - c2(i));
    [C,I]=min(Z); % I is the index of the min value of Z. (C is the min value, but it's
not used in this code).
    p1(i) = p(I);
end

% Plot result. The following creates Fig 2a.

figure(1)
subplot(2,2,1)
plot(c2,1/3+2*c2/3,'k-.',c2,1/14+13*c2/14,'k--',c2,p1,'r:',c2,p2,'b-');
legend('p1 & p2 when x1=x2=1','p1 & p2 when x1=x2=12','p1 when x1=1 & x2=12','p2 when
x1=1 & x2=12','Location','SouthEast');
xlabel('Marginal cost');
ylabel('Price');
title('Fig. 2a: Equilibrium pricing strategies');
axis square

%% Remaining figures

% In order to create the remaining figures I start all over again by first
% clearing the memory and then finding the equilibrium prices given x1=1
% and x2 taking values from 1 to tmax=20 (using a loop).

clear all

% Specify the parameters in the distribution functions
x1=1; % Firm 1's efficiency parameter

tmax = 20; % This is the highest value of firm 2's efficiency parameter.

% The following code is identical to the one above, which was adapted from
% Fibich and Gavish (2011, online appendix), but it is put into a loop
% running from 1 to tmax.

for t = 1:tmax

    X2(t)=t;

    x2=t;

% Construct grid for c2
    h=1/1000; % set value of grid spacing h
    c2=(h:h:1-h)'; % a column vector

```

```

nGrid=numel(c2); % Number of grid points = nGrid

% Construct a finite-difference approximation operators
e=ones(nGrid,1); % 2nd order approximation matrix % e is a column vector
D=spdiags([-e 0*e e], -1:1, nGrid, nGrid)/2/h;

Dc=D;Dc_RHS=e*0; % Handle boundary c(1)=1 for Dc
Dc_RHS(end)=1/2/h;

Dp=D;Dp_RHS=e*0;
Dp_RHS(end)=1/2/h; % Handle boundary p(1)=1 for Dp
Dp(1,1:3)=[-3 4 -1]/2/h; % Handle one-sided boundary p(0)=? for Dp

% Construct initial guess
c1=c2; % a column vector
p=(1+2*c2)/3; % a column vector

% Compute solution using iterations
for iter=1:50
% Calculate v^(k+1)
c_coeff=(x2.*(e-c1).*(e-2.*p+c2))./(x1.*(p-c2).*(e-c2).*(e-2*p+c1));
LHS_c=Dc+spdiags(c_coeff,0,nGrid,nGrid);
RHS_c=p.*c_coeff-Dc_RHS;
c1=LHS_c\RHS_c;

% Calculate b^(k+1)
p_coeff=(x2.*(e-p))./((e-c2).*(e-2*p+c1));
LHS_p=Dp-spdiags(p_coeff,0,nGrid,nGrid);
RHS_p=-c1.*p_coeff-Dp_RHS;
p=LHS_p\RHS_p;
end

p2 = p;

%% All of the remaining code is not adapted from Fibich and Gavish (2011),
% but constructed by me to solve my particular problem.

% Construct vectors with the probabilities that a particular cost level has realized.

% Divide the unit line into hh=h/2 gridpoints. Then the c values in the grid used
% above will be located at the points 2, 4, 6, and so on, in this new grid.
% Approximate the probability of a c value at, say, 2 in the new grid by
% calculating the probability that c is between the points 1 and 3. Given
% the distribution function used here, this probability equals (1-a)^x -
% (1-b)^2, where a=1/hh and b=3/hh and where either x=x1 or x=x2, depending on
% which firm we consider. With this way of approximating the probabilities,
% they will add up to something smaller than one, as the space between the
% left end of the unit line and grid point 1, as well as the space between
% the right end and the last gridpoint, are not counted. To reduce the
% error created by this I normalize the level of all probabilities so that
% they add up to one.

hh = h/2; % A grid that is exactly half as the one used when finding the eq prices above.

```

```

for i=1:nGrid

    prc1(i) = (1-hh*(2*(i-1)+1))^x1 - (1-hh*(2*i+1))^x1;

    prc2(i) = (1-hh*(2*(i-1)+1))^x2 - (1-hh*(2*i+1))^x2;

end

Sumprc1 = prc1*e; % The sum of firm 1's probabilities.
Sumprc2 = prc2*e; % The sum of firm 2's probabilities.

prc1 = prc1/Sumprc1; % Normalize firm 1's probabilities so they add up to one.
prc2 = prc2/Sumprc2; % Normalize firm 2's probabilities so they add up to one.

%% Construct a vector with firm 1's prices, p1. Above we treated firm 1's
% price as exogenous, and solved for the values of firm 1's cost and firm 2's price
% that were consistent with equilibrium. Now we need to "invert" these
% results. This is done by, for each c value on the grid, identify the element
% in the c1 vector that is closest in value to that grid point. The index
% of that element of the c1 vector will be denoted I. We can then pick the
% I'th element in the p vector solved for above and put that element in our
% newly constructed vector p1.

for i=1:nGrid
    Z = abs(c1 - c2(i));
    [C,I]=min(Z); % I is the index of the min value of Z. (C is the min value, but it's ✓
not used in this code.)
    p1(i) = p(I);
end

Ep1(t) = prc1*p1';
Ep2(t) = prc2*p2';

%% Calculate expected market price

P1 = repmat(p1',1,nGrid); % An (nGrid x nGrid) matrix with each column = p1.
P2 = repmat(p2',nGrid,1); % An (nGrid x nGrid) matrix with each row = p2.
marketprice = min(P1,P2);
Emarketprice(t) = prc1 * marketprice * prc2';

%% Calculate expected industry and individual profits

c = c2;

Pratio = P2./P1;

H1 = min(ones(size(Pratio)),Pratio); % An (nGrid x nGrid) matrix with 1's where p1 is ✓
lowest and a number strictly between zero and one otherwise.
H1 = floor(H1); % As above but making the numbers that are strictly between zero and ✓
one equal to zero by elimination of the non-integer part.

half = 0.5*ones(nGrid,1);
H1 = H1 - diag(half);
H2 = ones(size(H1))-H1;

```

```

C1 = repmat(c,1,nGrid); % An (nGrid x nGrid) matrix with each column = c.
C2 = repmat(c',nGrid,1); % An (nGrid x nGrid) matrix with each row = c

Q1 = ones(size(P1))-P1;
Q2 = ones(size(P2))-P2;

Profit1 = (P1-C1).*Q1;
Profit2 = (P2-C2).*Q2;

EProfit1(t) = prc1 * (Profit1 .* H1) * prc2';
EProfit2(t) = prc1 * (Profit2 .* H2) * prc2';

EProfit(t) = EProfit1(t) + EProfit2(t);

CS1 = (Q1.^2)/2;
CS2 = (Q2.^2)/2;

ECS1(t) = prc1 * (CS1 .* H1) * prc2';
ECS2(t) = prc1 * (CS2 .* H2) * prc2';

ECS(t) = ECS1(t) + ECS2(t);

ETS(t) = EProfit(t) + ECS(t);

%% Calculate expected price under complete info

% U1, U2 and U3 give us the eq prices when firm 2 has the lowest cost (so
% the part of the matrix below the main diagonal is the relevant one).

U1 = repmat(c,1,nGrid); % An (nGrid x nGrid) matrix with each column = c.
U2 = repmat(((c+1)./2)',nGrid,1); % An (nGrid x nGrid) matrix with each row = (c+1)/2
U3 = min(U1,U2);

% U4, U5 and U6 give us the eq prices when firm 1 has the lowest cost (so
% the part of the matrix above the main diagonal is the relevant one).

U4 = repmat(c',nGrid, 1); % An (nGrid x nGrid) matrix with each row = c.
U5 = repmat((c+1)./2,1,nGrid); % An (nGrid x nGrid) matrix with each column = (c+1)/2
U6 = min(U4,U5);

% Now add the part of U3 below, and on, the main diagonal to the part of U6
% that is strictly above the main diagonal (so as to ensure that the main
% diagonal elements are not counted twice).

U = tril(U3) + triu(U6,1);

% Calculate the expected price under complete information by taking the sum
% of the sum of the elements in U weighted by the probabilities of (c1,c2).

Epcomplete(t) = prc1 * U * prc2';

%% Calculate expected profits under complete info

Q1CI = ones(size(U6)) - U6;
Q2CI = ones(size(U3)) - U3;

```

```

Profit1CI = (U6-C1).*Q1CI;
Profit2CI = (U3-C2).*Q2CI;

EProfit1CI(t) = prc1 * triu(Profit1CI) * prc2';
EProfit2CI(t) = prc1 * tril(Profit2CI,-1) * prc2';
EProfitCI(t) = EProfit1CI(t) + EProfit2CI(t);

%% Calculate expected consumer surplus under complete info

CS1CI = (Q1CI.^2)/2;
CS2CI = (Q2CI.^2)/2;
ECS1CI(t) = prc1 * triu(CS1CI) * prc2';
ECS2CI(t) = prc1 * tril(CS2CI,-1) * prc2';
ECSCI(t) = ECS1CI(t) + ECS2CI(t);
ETSCI(t) = EProfitCI(t) + ECSCI(t);

%% Calculate expected Lerner index

LI1CI = (U6-C1)./U6;
LI2CI = (U3-C2)./U3;

ELICI(t) = prc1 * triu(LI1CI) * prc2' + prc1 * tril(LI2CI,-1) * prc2';

%% Calculate probability that firm 2 has the lowest cost

S = ones(nGrid);
S = tril(S,+1);
S = S + (1/2)*eye(nGrid);

Prc2lowest(t) = prc1 * S * prc2';

Prp2lowest(t) = 1 - prc1 * H1 * prc2';

%% Calculate probaility of missallocation (the less efficient firm charges the lowest price)
Prmissall(t) = prc1 * tril(H1,-1) * prc2';
end

% Create Fig 2b
subplot(2,2,2)
plot(X2,Ep1,'r--',X2,Ep2,'b:',X2,Emarketprice,'k-');
legend('Expected firm 1 price', 'Expected firm 2 price', 'Expected market price', 'Location', 'SouthEast');
xlabel('Firm 2 ex ante efficiency, x2');
ylabel('Expected price')
title('Fig. 2b: Expected prices')
axis([1 tmax .1 .7])
axis square

% Create Fig 2c
subplot(2,2,3)
plot(X2,Prc2lowest,'r--',X2,Prp2lowest,'b:');
legend('Prob firm 2 cost is lowest', 'Prob firm 2 price is lowest', 'Location', 'SouthEast');

```

```
xlabel('Firm 2 ex ante efficiency, x2');
ylabel('Probability')
title('Fig. 2c: Prob firm 2 price, resp. cost, is lowest')
axis([1 tmax .5 1])
axis square

% Create Fig 2d
subplot(2,2,4)
plot(X2,Prmissall,'k-');
legend('Prob of ex post inefficiency','Location','SouthEast');
xlabel('Firm 2 ex ante efficiency, x2');
ylabel('Probability')
title('Fig. 2d: Prob of ex post inefficiency')
axis([1 tmax .0 .2])
axis square

% Create Fig 3a-f
figure(2)

% Create Fig 3a
subplot(2,3,1)
plot(X2,Epcomplete,'k-',X2,Emarketprice,'r:');
legend('E(p) with complete info','E(p) with incomplete info','Location','SouthEast');
xlabel('Firm 2 ex ante efficiency, x2');
ylabel('Expected price')
title('Fig. 3a: Expected prices')
axis([1 tmax .2 .7])
axis square

% Create Fig 3b
subplot(2,3,2)
plot(X2,ECSCI,'k-',X2,ECS,'r:');
legend('E(S) with complete info','E(S) with incomplete info','Location','SouthEast');
xlabel('Firm 2 ex ante efficiency, x2');
ylabel('Expected consumer surplus')
title('Fig. 3b: Expected consumer surplus')
axis square
axis([1 tmax .1 .3])

% Create Fig 3c
subplot(2,3,3)
plot(X2,ETSCI,'k-',X2,ETS,'r:');
legend('E(W) with complete info','E(W) with incomplete info','Location','SouthEast');
xlabel('Firm 2 ex ante efficiency, x2');
ylabel('Expected total surplus')
title('Fig. 3c: Expected total surplus')
axis square
axis([1 tmax .2 .45])

% Create Fig 3d
subplot(2,3,4)
plot(X2,Emarketprice./Epcomplete,'k-');
xlabel('Firm 2 ex ante efficiency, x2');
ylabel('Expected price')
title('Fig. 3d: As in Fig. 3a, but the share')
```

```
axis([1 tmax 0.6 1.3])
axis square

% Create Fig 3e
subplot(2,3,5)
plot(X2,ECS./ECSCI,'k-');
xlabel('Firm 2 ex ante efficiency, x2');
ylabel('Expected consumer surplus')
title('Fig. 3e: As in Fig. 3b, but the share')
axis([1 tmax 1 1.4])
axis square

% Create Fig 3f
subplot(2,3,6)
plot(X2,ETS./ETSCI,'k-');
xlabel('Firm 2 ex ante efficiency, x2');
ylabel('Expected total surplus')
title('Fig. 3f: As in Fig. 3c, but the share')
axis([1 tmax 1.05 1.1])
axis square

% Create Fig 4a-c
figure(3);

% Create Fig 4a
subplot(1,3,1)
plot(X2,EProfit1CI,'k-',X2,EProfit1,'r:');
legend('E({Pi}), complete info','E({Pi}), incomplete info', 'Location','NorthEast');
xlabel('Firm 2 ex ante efficiency, x2');
ylabel('Expected profits')
title('Fig. 4a: Exp firm 1 profits')
axis([1 tmax .0 .10])
axis square

% Create Fig 4b
subplot(1,3,2)
plot(X2,EProfit2CI,'k-',X2,EProfit2,'r:');
legend('E({Pi}), complete info','E({Pi}), incomplete info', 'Location','SouthEast');
xlabel('Firm 2 ex ante efficiency, x2');
ylabel('Expected profits')
title('Fig. 4b: Exp firm 2 profits')
axis([1 tmax .05 .20])
axis square

% Create Fig 4c
subplot(1,3,3)
plot(X2,EProfitCI,'k-',X2,EProfit,'r:');
legend('E({Pi}), complete info','E({Pi}), incomplete info', 'Location','SouthEast');
xlabel('Firm 2 ex ante efficiency, x2');
ylabel('Expected profits')
title('Fig. 4c: Exp aggregate profits')
axis([1 tmax .05 .20])
axis square
```