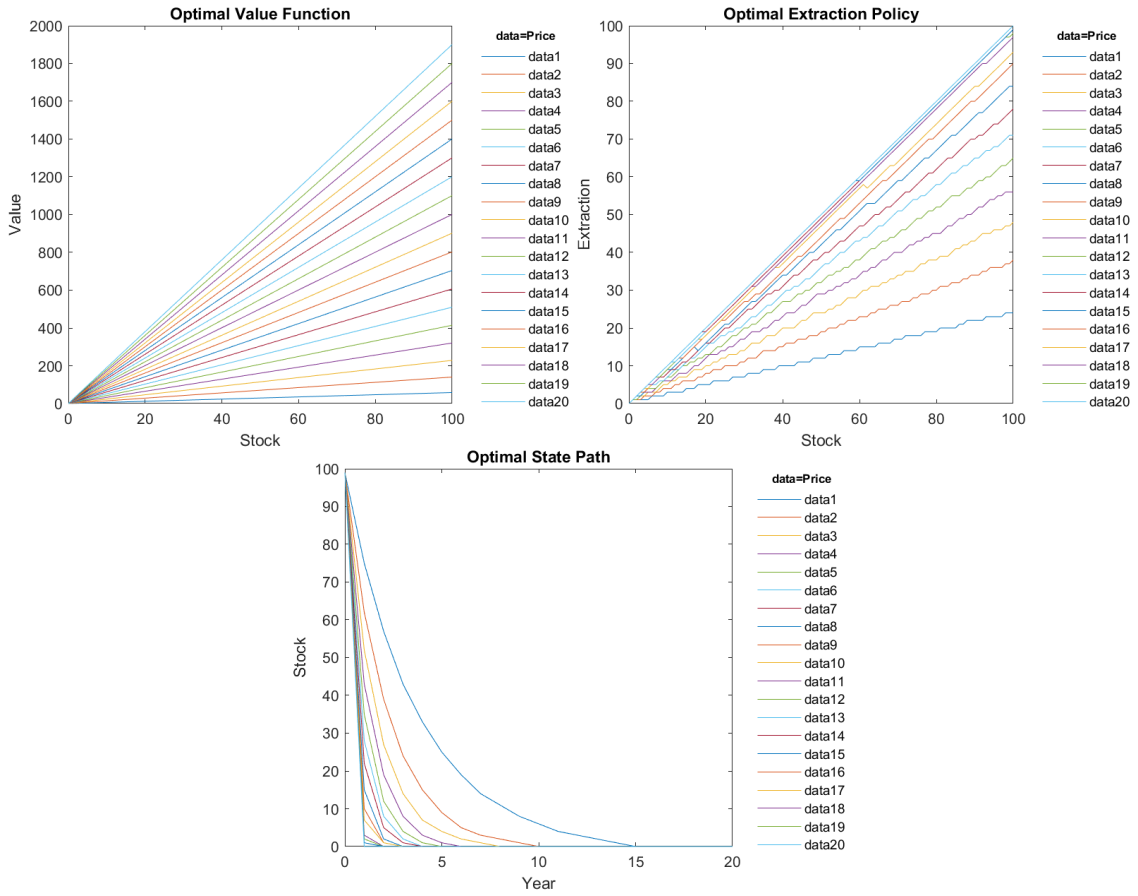
1 Deterministic Dynamic Models



From the following figures, we see as price increases, the value of our stock also increases, which should be obvious. However, we also see that as price increases, the more stock is extracted in the optimal extraction policy. At the same time, the optimal state path shows us that consumption increases over a shorter time horizon (i.e. consumption from the previous years increases).

As the price increases and approaches infinity, we notice our stock value, again, also increases proportionally, and also towards infinity. As price increases, the optimal extraction policy would be continue to extract more and more of the resource until we extract all of it—which occurs when the resource price is $13 and above. Consumption of the entire stock of resources would also be done within the first time period (the first year in this case). In fact, we see that when the resource price is $16 and above, essentially all of the resource was consumed within the first year.

See below for Matlab code.

```
clc;clear;

% Enter model parameters
price = 1:20;               % price of ore
sbar  = 100;                % initial ore stock
delta = 0.9;                % discount factor

% Construct state and action spaces
S = (0:sbar)';              % vector of states
X = (0:sbar)';              % vector of actions
n = length(S);              % number of states
m = length(X);              % number of actions

% Empty matrices to store values
vs    = zeros(sbar+1, numel(price));
xs    = zeros(sbar+1, numel(price));
```

```
pstars = zeros(sbar+1, sbar+1, numel(price));

% Vectorized version
for k=1:numel(price)
p = price(k);
[SS,XX] = gridmake(S,X);
f = (p-XX./(1+SS)).*XX;
f(XX>SS) = -inf;
f = reshape(f,n,m);
g = getindex(SS-XX,SS);
g = reshape(g,n,m);
clear SS XX

% Pack model data
clear model
model.reward   = f;
model.transfunc = g;
model.discount = delta;
model.horizon  = inf;

% Solve model using policy iteration for each price
[v,x,pstar] = ddpsolve(model);
vs(:,k)=v;
xs(:,k)=x;
pstars(:,:,k)=pstar;
end

% Plot optimal value function
plot(S,vs);
title('Optimal Value Function');
xlabel('Stock'); ylabel('Value');
lgd1 = legend('Location',"eastoutside");
legend("Box","off");
title(lgd1,'data=Price','FontSize',8);

% Plot optimal policy function
plot(S,X(xs));
title('Optimal Extraction Policy');
xlabel('Stock'); ylabel('Extraction');
lgd2=legend('Location',"eastoutside");
legend("Box","off");
title(lgd2,'data=Price','FontSize',8);

% Generate optimal path
sinit = max(S); nyrs = 20;
spaths = zeros(nyrs+1, numel(price));
for z=1:numel(price)
spath = ddpsimul(pstars(:,:,z),sinit,nyrs);
spaths(:,z) = S(spath);
end
% Plot optimal path
plot(0:nyrs,spaths);
title('Optimal State Path');
xlabel('Year'); ylabel('Stock');
lgd3=legend('Location',"eastoutside");
legend("Box","off");
title(lgd3,'data=Price','FontSize',8);
```