1

1.a

Given our revenue function

$$f(x,y) = -2x^2 - y^2 + xy + 8x + 3y$$

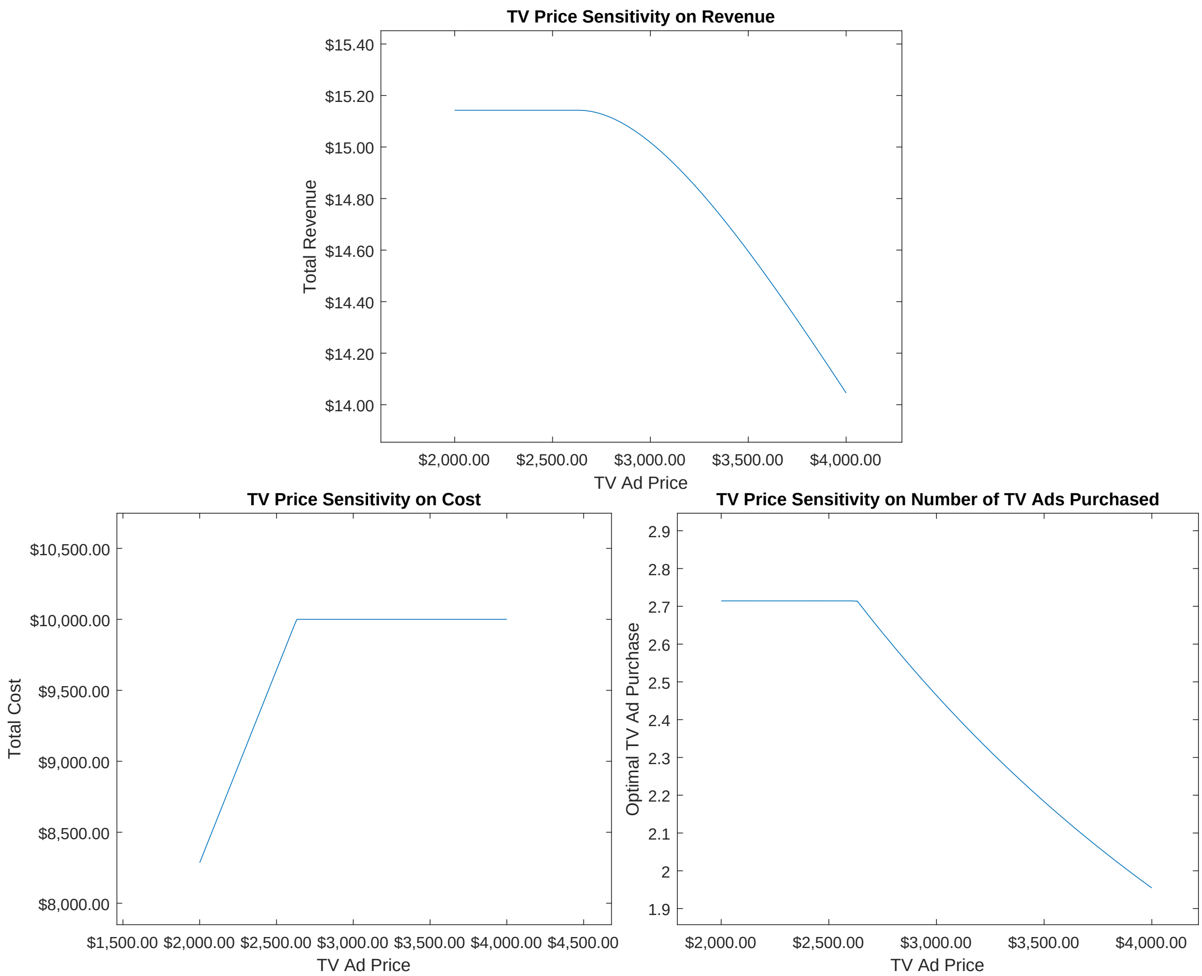
where x is a one minute TV ad costing $3,000 and y is a one minute radio ad costing $1,000, we can use the nonlinear programming problem solving *fmincon* in order to minimize this negative revenue function $-f(x,y)$ [see Matlab code below or attached separately in email]. We obtain the values $x = [2.4643; 2.6071]$. Since the company can only purchase one minute blocks, we need to round these values to the nearest integer. However, we can add an additional one or two radio ads without breaking our budget constraint ($10,000), we will need to manually test those values.

Plugging in $x1 = [2;2]$, $x2 = [2,3]$, and $x3 = [2,4]$, we obtain the following revenue, respectively: $14, $14, and $12. This means we can run either 2 TV and 2 radio ads or 2 TV or 3 radio ads for the same maximum return in revenue at $14. Revenue falls to $12 if we buy 2 TV and 4 radio ads.

1.b

In order to determine the marginal rate of return on the company's additional overall advertisement spending, we can check the values in our lambda. We see that in lambda.lower, for every dollar of additional spending in TV and radio ads, we should expect a 8.1152e-0.9 and 7.6706e-0.9 return in our revenue, respectively. Our lambda.ineqlin gives us 2.5e-04, which means for every dollar our budget increases, we should expect a 2.4e-04 return in our revenue, all else equal.

1.c



Unfortunately, I was not able to discretize these graphs into one minute blocks. However, at the TV ad price of $2,000, the company should purchase 3 TV ads and 3 radio ads for a total revenue of $15 at the cost of $9,000. As the price of TV ads increases, we find, naturally, the company would purchase fewer TV ads. The total cost also increases until the company hits their budget of $10,000 and with total revenue decreasing proportionally after TV ad prices hit around $2,600. We expect the company to purchase 3 TV ads if the price is cheaper than around $3,000 and to purchase 2 thereafter until the TV ad price hits over $5,000, where the company will only purchase 1 TV ad, as 2 at that price would already be over budget.

2

2.a

This is an infinite horizon, deterministic model with one state variable and one choice variable with a nonrenewable resource. The current reward, net profit, would be revenue minus the cost of extraction

$$f(s_t, x_t) = p(x_t)x_t - c(x_t)$$

The value of the resource stock would be:

$$V(s_t) = \max_{0 \leq x_t \leq s_t} \{p(x_t)x_t - c(x_t) + \delta V(s_t - x_t)\}$$
$$\text{s.t. } s_{t+1} = h(s_t - x_t)$$

$$0 \leq x_t \leq s_t$$
$$0 \leq s_t \leq \bar{s}; \text{ where } \bar{s} = 8$$

However, we could formulate this as a renewable resource as such:

$$f(x_t) = \int_0^{x_t} p(x_t)dx - c(x_t)$$

The value of the resource stock would be:

$$V(s_t) = \max \sum_{t=0}^{\infty} \int_0^{x_t} [p(x)dx - c(x)] + \delta V(s_{t+1})$$
$$\text{s.t. } s_{t+1} = h(s_t - x_t)$$

$$0 \leq x_t \leq s_t$$
$$0 \leq s_t \leq \bar{s}; \text{ where } \bar{s} = 8$$

2.b

Bellman's Equation:

$$V(s_t) = \max \sum_{t=0}^{\infty} \int_0^{x_t} [p(x)dx - c(x)] + \delta V(s_{t+1}), t = 0, \ldots, \infty$$

Taking the first-order condition:

$$\begin{aligned}
\frac{\partial V(s_t)}{\partial x_t} &= \frac{\partial}{\partial x_t}[p(x)dx - c(x) + \delta V(s_{t+1})] = 0 \\
&= p(x_t) - c'(x_t) - \delta \frac{\partial V}{\partial s_{t+t}} h'(.) = 0 \\
&= p(x_t) - c'(x_t) = \delta \lambda_{t+1} h'(.)
\end{aligned}$$

By the envelope theorem,

$$\frac{\partial V(s_t)}{\partial s_t} = \frac{\partial}{\partial s_t}[p(x)dx - c(x) + \delta V(s_{t+1})] = 0$$
$$= \lambda s_t = \delta \lambda_{t+1} h'(.)$$

We find that marginal revenue (current price) minus the marginal cost (the harvesting cost) equals the shadow price of leaving the resource to be harvested in the future. In other words, $\delta\lambda_{t+1}h'(.)$ is the opportunity cost of harvesting and λ_{t+1} is the shadow price of leaving the resource not extracted.

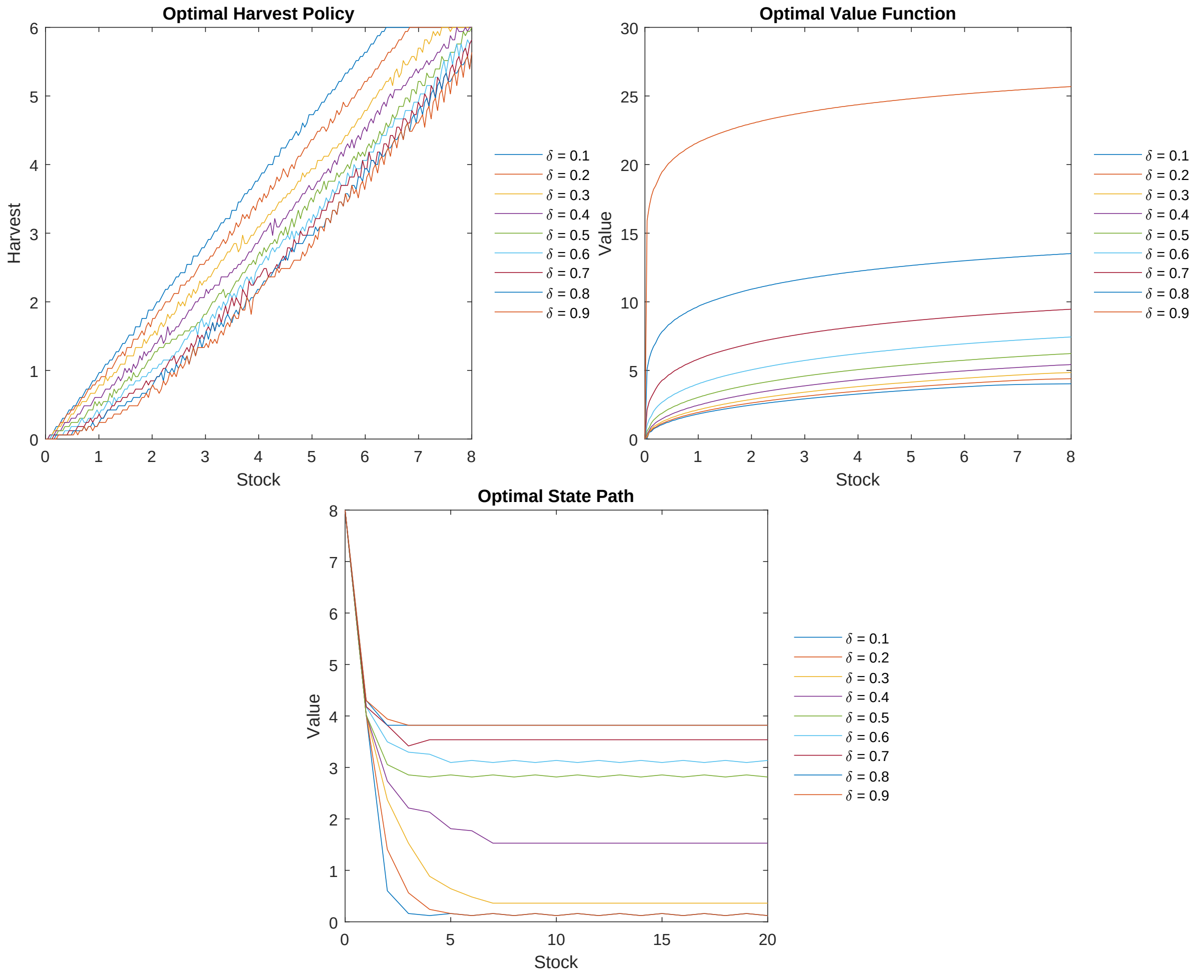
2.c

We can plug in the values from our model assumptions:

$$\frac{V(s_t)}{\partial x_t} = \frac{\partial}{\partial x_t}[p(x)dx - c(x) + \delta V(s_{t+1})]$$
$$= 0.5x_t^{-0.5} - 3.2 + s_t - x_t = 0$$

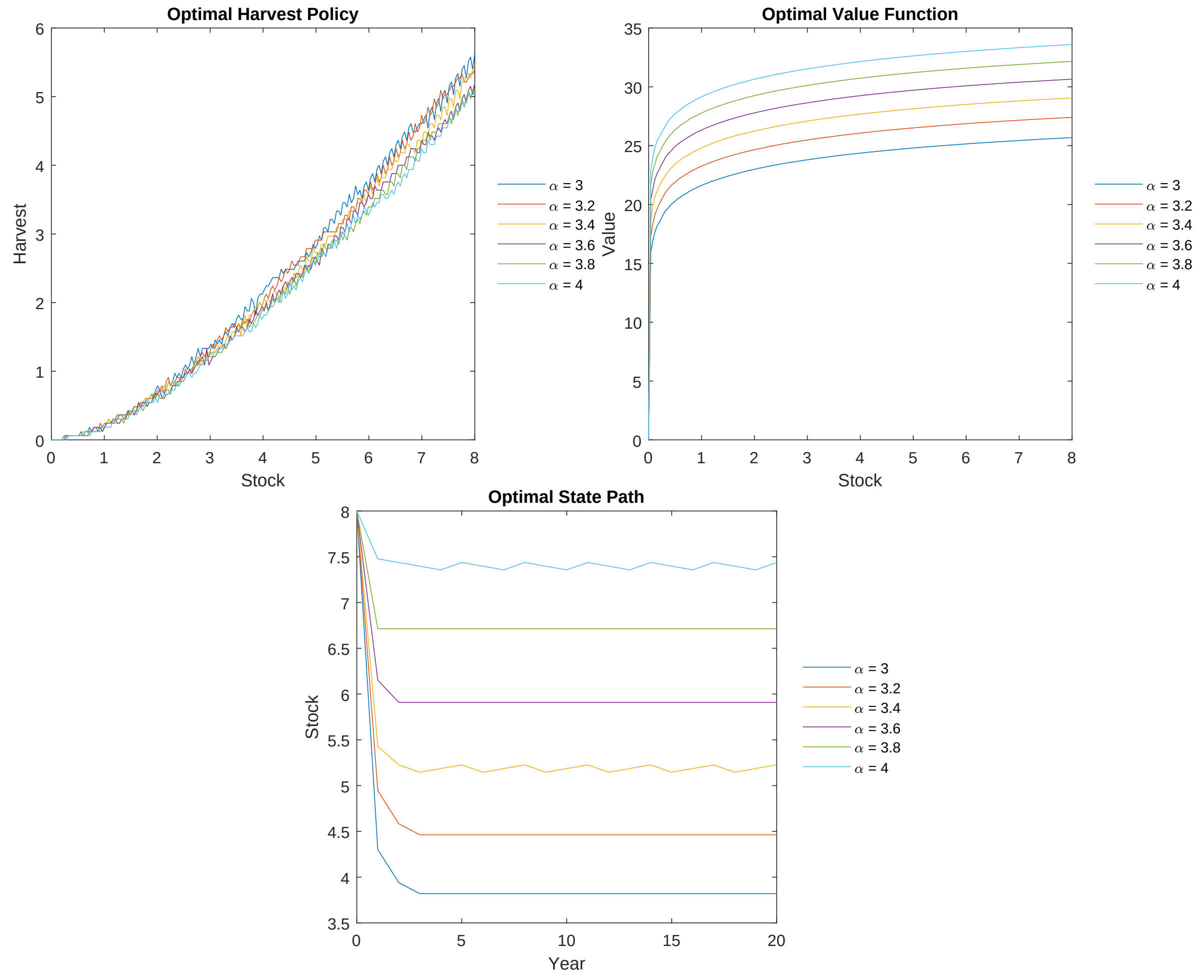
2.d

Sensitivity analysis with respect to discount factor (δ)



We see that as the discount factor increases from 0.1 to 0.9, our optimal harvesting policy decreases, the value increases, and more of the stock is left for consumption at a later time period. This makes sense since as the discount factor increases, it means one values the resource increasingly more in the future, and thus, in the steady state, more of the resource stock is saved for the next time period.

Sensitivity analysis with respect to growth factor (α)



We see that as the growth factor increases from 3 to 4, the optimal harvesting policy decreases, the value increases, and more of the stock is left for consumption at a later time period. This means that the faster the resource grows, the more that can be harvested in a shorter time frame, making it more valuable. Additionally, if the growth rate is higher or equal to consumption rate, it is no longer a scarce (or as scarce) of a resource, and in the steady state with increasing growth factor, there will be more stock in the future.

3 Matlab Code

```
Problem 1.a
clear; clc;

rev = @(x) (2)*x(1)^2+x(2)^2-(x(1)*x(2))-8*x(1)-3*x(2); % Revenue function
x0 = [2; 1]; % Initial guess
A = [3000 1000]; % Cost for TV and radio ads
b = [10000]; % Budget constraint
Aeq = [];
Beq = [];
lb = zeros(2,1); % Inequality constraints (greater than or equal to 0)

options = optimset('Algorithm','interior-point', 'Display', 'off');

[x,fval,exitflag,output,lambda,grad,hessian] = fmincon(rev, x0, A, b, Aeq, Beq,
    lb, [], [], options);

disp(x);

% Since we can floor or ceil radio without breaking our budget constraint,
```

```
% we should double check which one gives us the higher revenue
x1 = [2;2]; % 2 TV ads, 2 radio ads
x2 = [2;3]; % 2 TV ads, 3 radio ads
x3 = [2;4]; % 2 TV ads, 4 radio ads
rev1 = rev(x1);
disp(rev1);
rev2 = rev(x2);
disp(rev2)
rev3 = rev(x3);
disp(rev3)

Problem 1.c

clear; clc;

TV = 2000:4000; % Price range of TV ads
z = zeros (5,numel(TV)); % Create a zero matrix for storing results

rev = @(x) (2)*x(1)^2+x(2)^2-(x(1)*x(2))-8*x(1)-3*x(2); % Revenue function
x0 = [2; 1]; % Initial guess
b = [10000]; % Budget constraint
Aeq = [];
Beq = [];
lb = zeros(2,1); % Inequality constraints (greater than or equal to 0)
options = optimset('Algorithm','interior-point', 'Display', 'off');

for k = 1:numel(TV)
TV1 = TV(k);  % number of women viewers
A = [TV1 1000]; % Cost for TV and radio ads
[x,fval,exitflag,output,lambda,grad,hessian] = fmincon(rev, x0, A, b, Aeq, Beq,
    lb, [], [], options);

z ([1 2], k) = x; % Stores TV ads in row 1, radio ads in row 2; floors to next
    lowest integer
z (3, k) = (-2)*z(1,k)^2-z(2,k)^2+z(1,k)*z(2,k)+8*z(1,k)+3*z(2,k); % Stores total
     continuous revenue in row 3
z (4, k) = TV(k); % Stores changing constraint price of TV ads from $2000 to
    $4000 in row 4
z (5, k) = z(1,k)*z(4,k)+z(2,k)*1000; % Stores calculated total cost in terms for
     1-min block commercial (discretized)
end

plot(TV,z(3,:));
ylabel('Total Revenue')
xlabel({'TV Ad Price'})
title('TV Price Sensitivity on Revenue')
ytickformat('usd')
xtickformat('usd')
xlim([1623 4285])
ylim([13.85 15.45])

plot(TV,z(5,:));
ylabel('Total Cost')
xlabel({'TV Ad Price'})
title('TV Price Sensitivity on Cost')
ytickformat('usd')
xtickformat('usd')
xlim([1461 4682])
ylim([7849 10748])

plot(TV,z(1,:));
ylabel('Optimal TV Ad Purchase')
xlabel({'TV Ad Price'})
```

```
title('TV Price Sensitivity on Number of TV Ads Purchased')
xtickformat('usd')
xlim([1797 4217])
ylim([1.86 2.95])

Sensitivity analysis of delta (discount factor)
clear;clc;
close all

% Enter model parameters
alpha = 3.0;               % growth function parameter
beta =  1.0;               % growth function parameter
gamma = 0.5;               % demand function parameter
cost =  0.2;               % marginal cost of harvest

% Construct state space
n    = 200;                % number of states
smin = 0;                  % minimum state
sbar = 8;                  % maximum state
S    = nodeunif(n,smin,sbar); % vector of states

% Construct action space
m    = 100;                % number of actions
xmin = 0;                  % minimum action
xmax = 6;                  % maximum action
X    = nodeunif(m,xmin,xmax); % vector of actions

% Sensitivity analysis of delta
delta = 0.1:0.1:0.9;              % discount factor

% Empty matrices to store values from change values of delta
vs    = zeros(n, numel(delta));
xs     = zeros(n, numel(delta));
pstars = zeros(n, n, numel(delta));

% Construct reward function
f = zeros(n,m);
for k=1:m
f(:,k) = (X(k).^(1-gamma))/(1-gamma)-cost*X(k);
f(S<X(k),k) = -inf;
end

% Construct state transition function
g = zeros(n,m);
for i=1:n
for k=1:m
snext = alpha*(S(i)-X(k)) - 0.5*beta*(S(i)-X(k)).^2;
g(i,k) = getindex(snext,S);
end
end

% Pack model structure
for i=1:numel(delta)
clear model
model.reward   = f;
model.transfunc = g;
model.discount = delta(i);

% Solve infinite-horizon model using policy iteration
[v,x,pstar] = ddpsolve(model);
vs (:, i) = v;
xs (:, i) = X(x);
pstars (:,:,i) = pstar;
```

```
end

plot(S,xs);
title('Optimal Harvest Policy');
legend('Location',"eastoutside");
xlabel('Stock'); ylabel('Harvest');
legend("Box","off");
for i = 1:numel(delta)
legendInfo{i} = ['\delta = ' num2str(i/10)];
end
legend(legendInfo)

plot(S,vs);
title('Optimal Value Function');
xlabel('Stock'); ylabel('Value');
lgd1 = legend('Location',"eastoutside");
legend("Box","off");
legend(legendInfo)

sinit = max(S);
nyrs = 20;
spaths = zeros(nyrs+1, numel(delta));
for z=1:numel(delta)
spath(:,:,z) = ddpsimul(pstars(:,:,z),n,nyrs);
spaths(:,z) = S(spath(:,:,z));
end

plot(0:nyrs,spaths)
title('Optimal State Path')
xlabel('Year'); ylabel('Stock');
lgd1 = legend('Location',"eastoutside");
legend("Box","off");
legend(legendInfo)

Sensitivity analysis of alpha (growth factor)

clear;clc;
close all

% Enter model parameters
beta =  1.0;                % growth function parameter
delta = 0.9;                % discount factor
gamma = 0.5;                % demand function parameter
cost =  0.2;                % marginal cost of harvest

% Construct state space
n    = 200;                 % number of states
smin = 0;                   % minimum state
sbar = 8;                   % maximum state
S    = nodeunif(n,smin,sbar); % vector of states

% Construct action space
m    = 100;                 % number of actions
xmin = 0;                   % minimum action
xmax = 6;                   % maximum action
X    = nodeunif(m,xmin,xmax); % vector of actions

% Sensitivity analysis of delta
alpha = 3:0.2:4;                % growth function parameter

% Empty matrices to store values from change values of alpha
vs    = zeros(n, numel(alpha));
xs     = zeros(n, numel(alpha));
```

```
pstars = zeros(n, n, numel(alpha));

% Construct reward function
f = zeros(n,m);
for k=1:m
f(:,k) = (X(k).^(1-gamma))/(1-gamma)-cost*X(k);
f(S<X(k),k) = -inf;
end

% Construct state transition function

% Pack model structure
for l=1:numel(alpha)
g = zeros(n,m);
for i=1:n
for k=1:m
snext = alpha(l)*(S(i)-X(k)) - 0.5*beta*(S(i)-X(k)).^2;
g(i,k) = getindex(snext,S);
end
end
clear model
model.reward   = f;
model.transfunc = g;
model.discount = delta;

% Solve infinite-horizon model using policy iteration
[v,x,pstar] = ddpsolve(model);
vs (:, l) = v;
xs (:, l) = X(x);
pstars (:,:,l) = pstar;
end

plot(S,xs);
title('Optimal Harvest Policy');
legend('Location',"eastoutside");
xlabel('Stock'); ylabel('Harvest');
legend("Box","off");
for i = 1:numel(alpha)
legendInfo{i} = ['\alpha = ' num2str(3+(i-1)*0.2)];
end
legend(legendInfo)

plot(S,vs);
title('Optimal Value Function');
xlabel('Stock'); ylabel('Value');
lgd1 = legend('Location',"eastoutside");
legend("Box","off");
legend(legendInfo)

sinit = max(S);
nyrs = 20;
spaths = zeros(nyrs+1, numel(alpha));
for z=1:numel(alpha)
spath(:,:,z) = ddpsimul(pstars(:,:,z),n,nyrs);
spaths(:,z) = S(spath(:,:,z));
end

plot(0:nyrs,spaths)
title('Optimal State Path')
xlabel('Year'); ylabel('Stock');
lgd1 = legend('Location',"eastoutside");
legend("Box","off");
legend(legendInfo)
```