```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% AAE590 Stochastic Control
% Machine Replacement Dynamic Programming
% Author: Travis Hastreiter
% Created On: 16 September, 2025
% Description: Solves machine replacement problem using dynamic
% programming.
% Most Recent Change: 16 September, 2025
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Example 2.1

```
N = 3;
R = 3.5;
T = 15;
theta = 0.4;

% Terminal cost
C_T = zeros(1, N);

[J_t, mu_t, J_t_cost] = dp_machine_replacement(N, R, T, theta, C_T)
```

```
J_t = 3×16
15.7205    14.7205    13.7205    12.7205    11.7205    10.7205    9.7207     8.7 ···
15.7205    14.7205    13.7205    12.7205    11.7205    10.7205    9.7205     8.7
13.2205    12.2205    11.2205    10.2205    9.2205     8.2205     7.2205     6.2
mu_t = 3×15
1    1    1    1    1    1    1    1    1    1    1    1    1    2
2    2    1    2    2    1    2    2    1    2    2    1    2    2    2
2    2    2    2    2    2    2    2    2    2    2    2    2    2    2
J_t_cost =
J_t_cost(:,:,1) =

   15.7205   14.7205   13.7205   12.7205   11.7205   10.7205    9.7207    8.7211    7.7202    6.7336    5.7560    4.
   15.7205   14.7205   13.7205   12.7205   11.7205   10.7205    9.7207    8.7211    7.7202    6.7336    5.7560    4.
   15.7205   14.7205   13.7205   12.7205   11.7205   10.7205    9.7207    8.7211    7.7202    6.7336    5.7560    4.


J_t_cost(:,:,2) =

   16.7205   15.7205   14.7205   13.7205   12.7205   11.7207   10.7211    9.7202    8.7336    7.7560    6.7000    5.
   15.7205   14.7205   13.7205   12.7205   11.7205   10.7206    9.7205    8.7202    7.7269    6.7224    5.7000    4.
   13.2205   12.2205   11.2205   10.2205    9.2205    8.2205    7.2205    6.2207    5.2211    4.2202    3.2336    2.
```

Exercise 2.2

```
N = 10;
R = 15;
T = 15;
theta = 0.4;

% Terminal cost
C_T = zeros(1, N);
```

```
[J_t, mu_t, J_t_cost] = dp_machine_replacement(N, R, T, theta, C_T)
```

```
J_t = 10×16
42.2577    39.6327    37.0098    34.3647    31.6652    28.8957    26.0976    23. ···
42.2577    39.6327    37.0098    34.3647    31.6652    28.8957    26.0976    23.
42.2577    39.6327    37.0098    34.3647    31.6652    28.8957    26.0976    23.
42.2577    39.6327    37.0098    34.3647    31.6652    28.8957    26.0976    23.
42.2577    39.6327    37.0098    34.3647    31.6652    28.8957    26.0976    23.
42.2577    39.6327    37.0098    34.3647    31.6652    28.8957    26.0976    23.
42.2577    39.6327    37.0098    34.3647    31.6652    28.8957    26.0976    23.
40.6536    38.0008    35.3282    32.6371    29.9518    27.3225    24.8058    22.
36.5094    33.8484    31.1952    28.5672    25.9774    23.4140    20.8195    18.
29.8940    27.2577    24.6327    22.0098    19.3647    16.6652    13.8957    11.
mu_t = 10×15
1    1    1    1    1    1    1    1    1    1    1    1    1    1    2
1    1    1    1    1    1    1    1    1    1    1    1    1    1    2
1    1    1    1    1    1    1    1    1    1    1    1    1    2    2
1    1    1    1    1    1    1    1    1    1    1    1    1    2    2
1    1    1    1    1    1    1    1    1    1    1    1    1    2    2
1    1    1    1    1    1    1    1    1    1    1    1    2    2    2
1    1    1    1    1    1    1    1    1    1    1    2    2    2    2
2    2    2    2    2    2    2    2    2    2    2    2    2    2    2
2    2    2    2    2    2    2    2    2    2    2    2    2    2    2
2    2    2    2    2    2    2    2    2    2    2    2    2    2    2
J_t_cost = 
J_t_cost(:,:,1) =

   42.2577   39.6327   37.0098   34.3647   31.6652   28.8957   26.0976   23.4000   21.0000   19.0000   17.4000   16.
   42.2577   39.6327   37.0098   34.3647   31.6652   28.8957   26.0976   23.4000   21.0000   19.0000   17.4000   16.
   42.2577   39.6327   37.0098   34.3647   31.6652   28.8957   26.0976   23.4000   21.0000   19.0000   17.4000   16.
   42.2577   39.6327   37.0098   34.3647   31.6652   28.8957   26.0976   23.4000   21.0000   19.0000   17.4000   16.
   42.2577   39.6327   37.0098   34.3647   31.6652   28.8957   26.0976   23.4000   21.0000   19.0000   17.4000   16.
   42.2577   39.6327   37.0098   34.3647   31.6652   28.8957   26.0976   23.4000   21.0000   19.0000   17.4000   16.
   42.2577   39.6327   37.0098   34.3647   31.6652   28.8957   26.0976   23.4000   21.0000   19.0000   17.4000   16.
   42.2577   39.6327   37.0098   34.3647   31.6652   28.8957   26.0976   23.4000   21.0000   19.0000   17.4000   16.
   42.2577   39.6327   37.0098   34.3647   31.6652   28.8957   26.0976   23.4000   21.0000   19.0000   17.4000   16.
   42.2577   39.6327   37.0098   34.3647   31.6652   28.8957   26.0976   23.4000   21.0000   19.0000   17.4000   16.


J_t_cost(:,:,2) =

   48.6327   46.0098   43.3647   40.6652   37.8957   35.0976   32.4000   30.0000   28.0000   26.4000   25.2000   24.
   47.6327   45.0098   42.3647   39.6652   36.8957   34.0976   31.4000   29.0000   27.0000   25.4000   24.2000   23.
   46.6327   44.0098   41.3647   38.6652   35.8957   33.0976   30.4000   28.0000   26.0000   24.4000   23.2000   22.
   45.6327   43.0098   40.3647   37.6652   34.8957   32.0976   29.4000   27.0000   25.0000   23.4000   22.2000   21.
   44.6327   42.0098   39.3647   36.6652   33.8957   31.0976   28.4000   26.0000   24.0000   22.4000   21.2000   20.
   43.6327   41.0098   38.3647   35.6652   32.8957   30.0976   27.4000   25.0000   23.0000   21.4000   20.2000   18.
   42.6327   40.0098   37.3647   34.6652   31.8957   29.0976   26.4000   24.0000   22.0000   20.4000   18.1200   14.
   40.6536   38.0008   35.3282   32.6371   29.9518   27.3225   24.8058   22.4096   20.0160   17.3600   14.0000   10.
   36.5094   33.8484   31.1952   28.5672   25.9774   23.4140   20.8195   18.0928   15.1440   12.0000    9.0000    6.
   29.8940   27.2577   24.6327   22.0098   19.3647   16.6652   13.8957   11.0976    8.4000    6.0000    4.0000    2.
```

```
function [J_t, mu_t, J_t_cost] = dp_machine_replacement(N, R, T, theta, C_T)
    % Running cost
    C_t = zeros([1, N, 2]);
    C_t(:, :, 1) = R; % Replacing cost
    C_t(:, :, 2) = N - (1 : N); % Use cost

    % State transition kernal
    tau = zeros([N, N, 2]);
    tau(:, :, 1) = [zeros(N - 1, N); ones([1, N])];
```

```
    tau(1 : N, 1 : N, 2) = diag([1, (1 - theta) * ones(1, N - 1)]);
    tau(((N + 1) : (N + 1) : N ^ 2) + N ^ 2) = theta;

    J_t_cost = zeros([N, T, 2]);
    J_t = zeros([N, T + 1]);
    mu_t = zeros([N, T]);

    x_i = @(i) [zeros([i - 1, 1]); 1; zeros([N - i, 1])];

    % Algorithm 1
    J_t(:, T + 1) = C_T;

    % Dynamic programming
    for t = T - 1 : -1 : 0
        for i = 1 : N
            J_t_cost(i, t + 1, :) = pagemtimes(C_t, x_i(i)) + sum(J_t(:, t + 2) .*
pagemtimes(tau, x_i(i)), 1);

            [J_t(i, t + 1), mu_t(i, t + 1)] = min(J_t_cost(i, t + 1, :), [], 3);
        end
    end
end
```