Exported Code - Team #15 Submission 1

Note: The MATLAB functions (mass_function, cost_function, min_cost_function, and min_gross_mass_funtion) throw an error when being published to a PDF because the function is not being called within each MATLAB file. So, there is an error in each of these files that should be ignored. The error is not thrown when properly called.

Note: The CSV files being read in full_plot and best_combos_plot are each member's Table 5.

Page 1: mass_function (Group)

Page 2: cost_function (Group)

Page 3: min_cost_function (Group)

Page 4: min_gross_mass_function (Group)

Pages 5-6: ben_1point2a (Ben)

Pages 7-8: chris_1point2a (Chris)

Pages 9-10: sara_1point2a (Sara)

Pages 11-12: sophie_1point2a (Sophie)

Pages 13-14: spyros_1point2a (Spyros)

Pages 15-16: ben_1point3a (Ben)

Pages 17-18: chris_1point3a (Chris)

Pages 19-20: sara_1point3a (Sara)

Pages 21-22: sophie_1point3a (Sophie)

Pages 23-24: spyros_1point3a (Spyros)

Pages 25-27: full_plot (Group)

Page 28: best_combos_plot (Group)

```
function [m_in1, m_in2, m_pr1, m_pr2, m0] = mass_function(Isp1, Isp2, X, delta1, delta2)
    % Returns the inert mass of stage 1, inert mass of stage 2, propellant
    % mass of stage 1, propellant mass of stage 2, and total mass given
    % specific impulse of stage 1, specific impulse of stage 2, and delV
    % fraction (delta1, delta2 are assumed to be 0.08)

    % Set constants based on mission requirements
    deltaV_tot = 12300; % m/s
    g0 = 9.81; % m/s^2
    m_pl = 26000; % kg

    % Solve for deltaV1 and deltaV2 from given delV fraction
    deltaV1 = X*deltaV_tot;
    deltaV2 = (1-X)*deltaV_tot;

    % Solve for mass ratio of stages 1 and 2 (m_i/m_f definition)
    R1 = exp(deltaV1/(g0*Isp1));
    R2 = exp(deltaV2/(g0*Isp2));

    % If (1 - R1*delta1) or (1 - R2*delta2) is less than 0, then the design
    % is structurally impossible based on the definitions
    % for m_2 and m_1 that follow, return NaN for all outputs
    if (1 - R1*delta1) <= 0 || (1 - R2*delta2) <= 0
        m_in1 = NaN; m_in2 = NaN; m_pr1 = NaN; m_pr2 = NaN; m0 = NaN;
        return
    end

    % Solve for total masses of stage 1 and stage 2
    m_2 = R2*m_pl/(1 - R2*delta2);
    m_1 = R1*m_2/(1 - R1*delta1);

    % Solve for inert masses of stage 1 and stage 2
    m_in1 = delta1*m_1;
    m_in2 = delta2*m_2;

    % Solve for propellant masses of stage 1 and stage 2
    m_pr1 = m_1 - m_in1 - m_2;
    m_pr2 = m_2 - m_pl - m_in2;

    % Assign total mass as the total mass of stage 1 (m_1 includes m_2)
    m0 = m_1;
end
```

```
Not enough input arguments.

Error in mass_function (line 13)
    deltaV1 = X*deltaV_tot;
              ^
```

Published with MATLAB® R2024b

```
function [stageCost_nr1, stageCost_nr2] = cost_function(Isp1, Isp2, X, delta1, delta2)
    % Returns the total cost given specific impulse of stage 1,
    % specific impulse of stage 2, and delV fraction (delta1, delta2 are
    % assumed to be 0.08)

    % Call mass function to get the inert masses of stage 1 and 2 for the
    % cost equation, then add the two stage costs together
    [m_in1, m_in2, m_pr1, m_pr2, m0] = mass_function(Isp1, Isp2, X, delta1, delta2);
    stageCost_nr1 = 13.52*m_in1^0.55;
    stageCost_nr2 = 13.52*m_in2^0.55;

end
```

```
Not enough input arguments.

Error in cost_function (line 8)
    [m_in1, m_in2, m_pr1, m_pr2, m0] = mass_function(Isp1, Isp2, X, delta1, delta2);
                                                     ^^^^
```

Published with MATLAB® R2024b

```
function [min_X, min_cost] = min_cost_function(Isp1, Isp2, delta1, delta2)
    % Returns the minimum cost given specific impulse of stage 1,
    % specific impulse of stage 2, and delV fraction (delta1, delta2 are
    % assumed to be 0.08)

    % Find minimum cost by iteratively calling cost_function and comparing
    % total cost for each X
    min_cost = realmax;
    for X = 0:0.001:1
        [stageCost_nr1, stageCost_nr2] = cost_function(Isp1, Isp2, X, delta1, delta2);
        total_cost = stageCost_nr1 + stageCost_nr2;
        if total_cost < min_cost
            min_cost = total_cost;
            min_X = X;
        end
    end
end
```

```
Not enough input arguments.

Error in min_cost_function (line 10)
        [stageCost_nr1, stageCost_nr2] = cost_function(Isp1, Isp2, X, delta1, delta2);
                                                       ^^^^
```

Published with MATLAB® R2024b

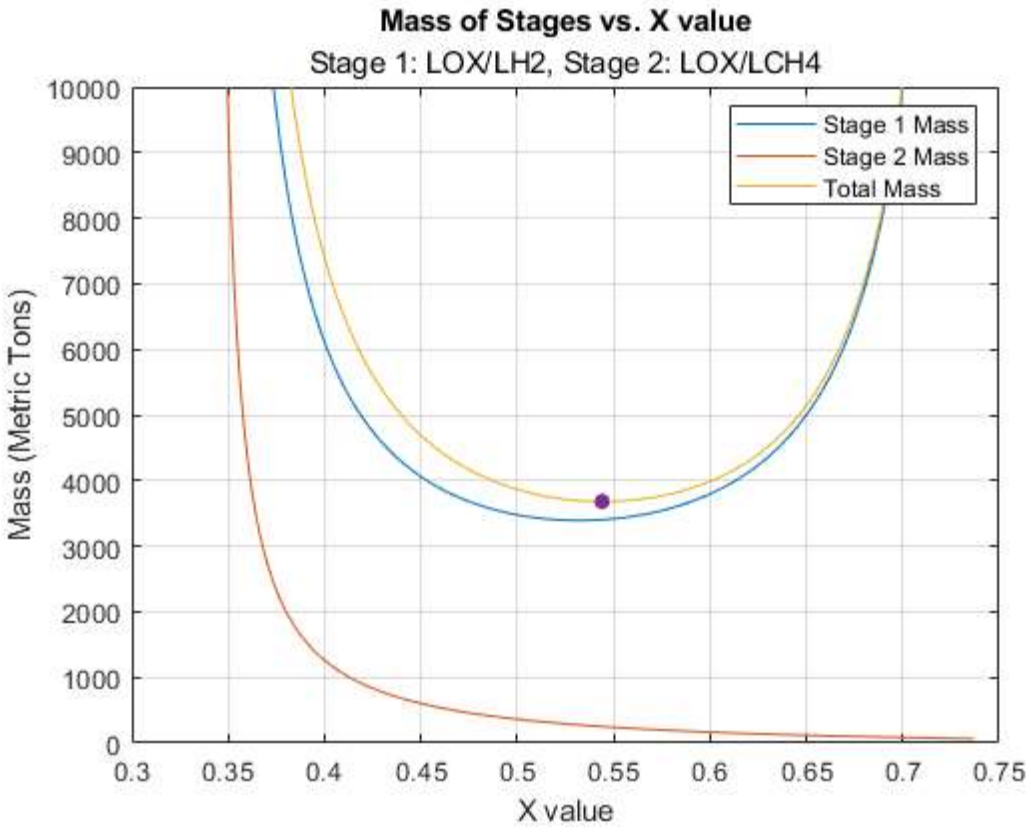
```
function [min_X, min_gross_mass] = min_gross_mass_function(Isp1, Isp2, delta1, delta2)
    % Returns the minimum gross mass given specific impulse of stage 1,
    % and specific impulse of stage 2 (delta1, delta2 are
    % assumed to be 0.08)

    % Initialize minimum gross mass
    min_gross_mass = realmax;

    % Find minimum gross mass by iteratively calling mass_function and comparing
    % gross mass for each X
    for X = 0:0.001:1
        [m_in1, m_in2, m_pr1, m_pr2, m0] = mass_function(Isp1, Isp2, X, delta1, delta2);
        if m0 < min_gross_mass
            min_gross_mass = m0;
            min_X = X;
        end
    end

end
```

```
Not enough input arguments.

Error in min_gross_mass_function (line 12)
        [m_in1, m_in2, m_pr1, m_pr2, m0] = mass_function(Isp1, Isp2, X, delta1, delta2);
                                                         ^^^^
```

Published with MATLAB® R2024b

```
% I have the LOX/LCH4 second stage row
% the first stage I am selecting is LOX/LH2 and the
% second stage is LOX/LCH4
addpath('..')
% given Constants
Delta_1 = 0.08;
Delta_2 = 0.08;
Isp_1 = 366;
Isp_2 = 327;
m_PL = 26000;

% the range of x from 0 to 1 at 0.01 intervals and the
% number of x values we will have
X = 0:0.001:1;
X_length = 1001;

% the arrays
X_array = zeros(1,X_length);
M_stage1 = zeros(1,X_length);
M_stage2 = zeros(1,X_length);
M_total = zeros(1,X_length);

% filling the arrays
S = 1;
while S <= X_length
    X_Position = X(S);
    % call mass function from 1.1
    [m_in1, m_in2, m_pr1, m_pr2, m0] = mass_function(Isp_1, Isp_2, X_Position, Delta_1, Delta_2);
    % defind and populate the arrays from these values
    M_stage1(S) = m_in1 + m_pr1;
    M_stage2(S) = m_in2 + m_pr2;
    M_total(S) = m0;
    S = S + 1;
end

% finding the minimum gross mass
Minimum_M0 = inf;
Minimum_M0X = NaN;

S = 2;
while S <= X_length
    if (M_stage1(S) > 0) && (M_stage2(S) > 0) && (M_total(S) < Minimum_M0)
        Minimum_M0 = M_total(S);
        Minimum_M0X = X(S);
    end
    S = S + 1;
end

% Converting to metric tons

M_stage1 = M_stage1 / 1000;
M_stage2 = M_stage2 / 1000;
M_total = M_total / 1000;
Minimum_M0 = Minimum_M0 / 1000;

% making the plots
```

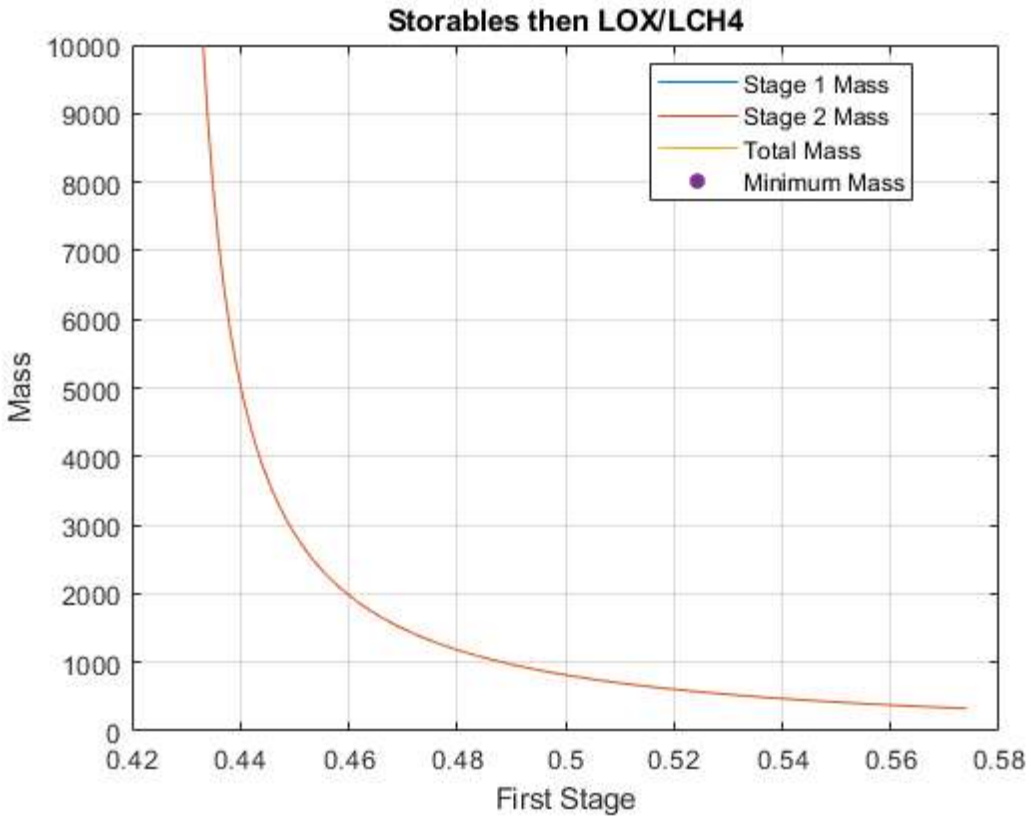
```
plot(X, M_stage1);
hold on;
plot(X, M_stage2);
plot(X, M_total);
plot(Minimum_M0X, Minimum_M0, '.', 'MarkerSize',20)
ylim([0,1e4]);
xlabel('X value');
ylabel('Mass (Metric Tons)');
title('Mass of Stages vs. X value');
subtitle('Stage 1: LOX/LH2, Stage 2: LOX/LCH4')
legend('Stage 1 Mass', 'Stage 2 Mass', 'Total Mass');
grid on;
```



Published with MATLAB® R2024b

```
% TP 1
% Chris Witherspoon
% 1.2

addpath('..')

% Delta constants
d1 = 0.08 ;
d2 = 0.08 ;

% Specific impulse (storables)
isp = 285 ;
isp_comp = 285 ; % Second

% Payload mass
mpl = 26000 ;

% X variable array
x = 0:0.001:1 ;

% Stage arrays initializations
first_stage = [] ;
second_stage = [] ;
tot_mass = [] ;

% Array population
i = 1 ; % initialize the variable to loop
while i <= length(x)
    [min1, min2, mpr1, mpr2, m0] = mass_function(isp, isp_comp, x(i), d1, d2);
    first_stage(end+1) = min1 + mpr1 ;
    second_stage(end+1) = min2 + mpr2 ;
    tot_mass(end+1) = m0 ;
    i = i + 1 ;
end

% Value placeholder for the mass
min_m0 = realmax ;
min_mox = NaN ;
% Check the min value
i = 1 ; %
while  i <= length(x)
    if first_stage(i) > 0 && second_stage(i) > 0 && tot_mass(i) < min_m0
        min_m0 = tot_mass(i) ;
        min_mox = x(i) ;
    end
    i = i + 1 ;
end

% Plotting the graph
plot(x,first_stage/1000) ;
hold on ;
grid on ;
title("Storables then LOX/LCH4")
plot(x, second_stage/1000) ;
plot(x, tot_mass/1000) ;
plot(min_mox, min_m0/1000, '.', MarkerSize = 20) ;
```

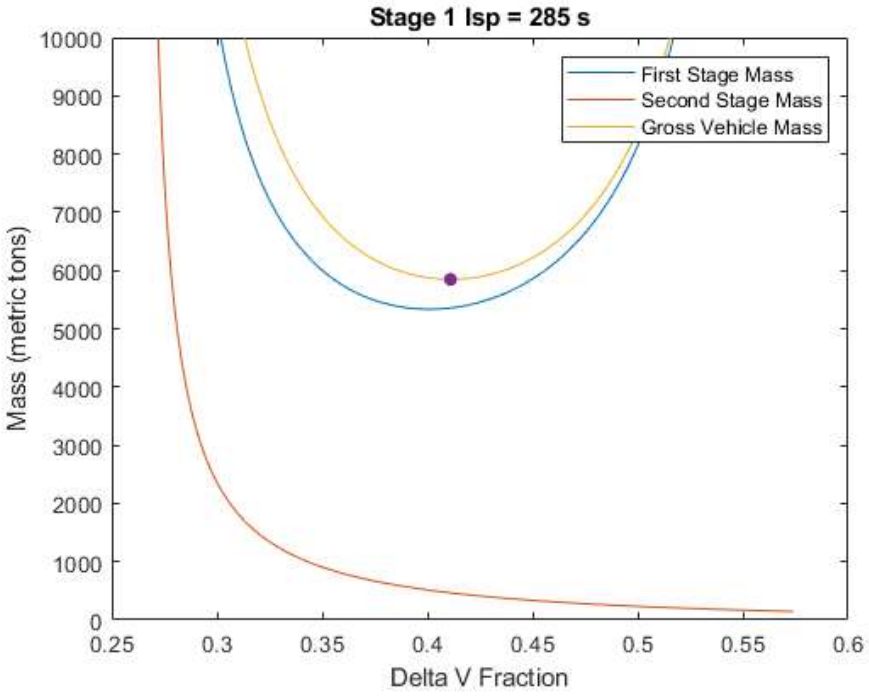
```
legend('Stage 1 Mass','Stage 2 Mass','Total Mass','Minimum Mass','Location','best');
ylim([0, 1e4]) ;
xlabel("First Stage") ;
ylabel("Mass") ;
```



Published with MATLAB® R2024b

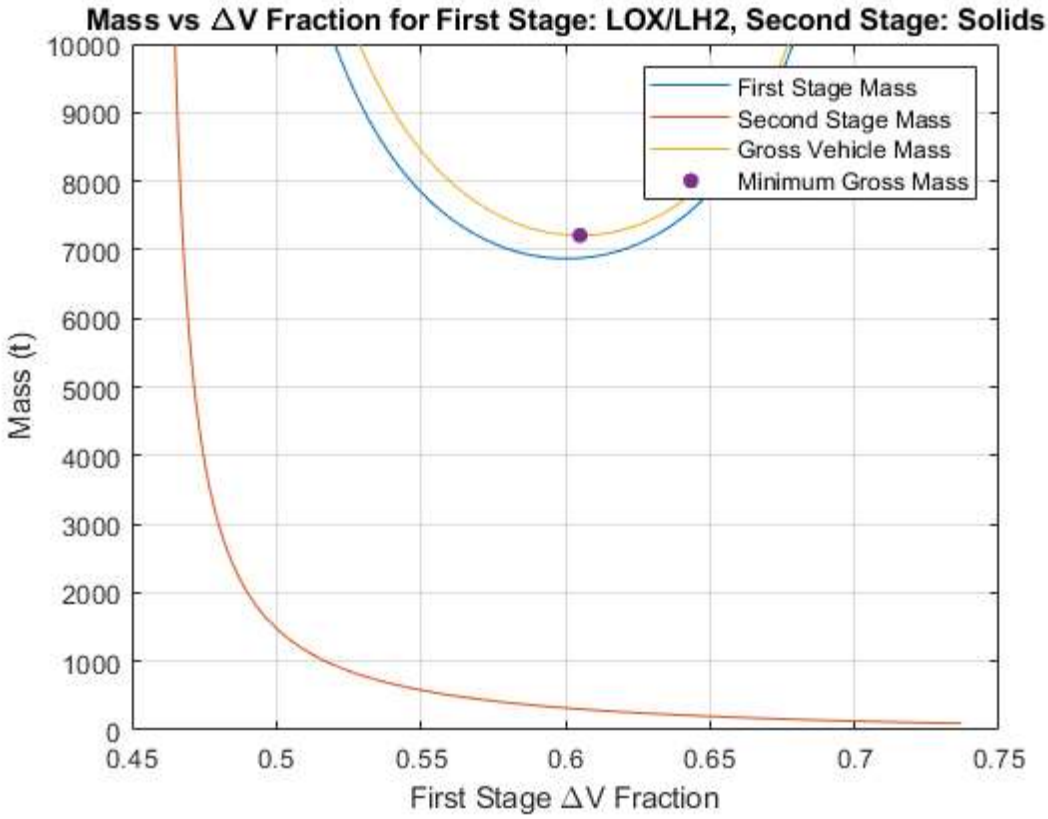
```
%Sara C 1.2a Code

%Looking at mixture of:
%Stage 2: LOX/LH2
%Varied Stage 1: LOX/LCH4, LOX/LH2, LOX/RP1, Solid, Storables

%Variables & Given Parameters
delta1 = 0.08;
delta2 = 0.08;
payload = 26000; %kg
%Specific Impulse
Isp_s2 = 366; %stage 2
%LCH4, LH2, RP1, solids, storables
Isp = [327, 366, 311, 269, 285]; %stage 1 varies


%initialize X (this is delta_V ratio/fraction)
X=0:0.001:1;

for isp = 1:length(Isp)
    %mass arrays
    m_s1 = [];
    m_s2 = [];
    m_0 = [];
    min_gross_mass = realmax;

    for i = 1:length(X)
        %calling mass function to populate mass arrays
        [m_in1, m_in2, m_pr1, m_pr2, m0] = mass_function(Isp(isp), Isp_s2, X(i), delta1, delta2);
        m_s1 = [m_s1 (m_pr1 + m_in1)];
        m_s2 = [m_s2 (m_pr2 + m_in2)];
        m_0 = [m_0 m0];

        %min gross mass function - returns the x,y minimum of the m_0 array
        if m_s1(i) > 0 && m_s2(i) > 0 && m_0(i) < min_gross_mass
            min_gross_mass = m_0(i); %convert to metric tons
            x_min = X(i);

        end
    end

    %kg to metric tons
    m_s1 = m_s1/1000;
    m_s2 = m_s2/1000;
    m_0 = m_0/1000;
    min_gross_mass = min_gross_mass/1000;

    %output - min mass and corresponding deltaV
    fprintf('Stage 1 Isp = %d s --> Optimum at X = %.3f, Min Gross Mass = %.2f metric tons\n', Isp(isp), x_min, min_gross_mass);

    %plotting
    plot(X, m_s1);
    hold on;
    plot(X, m_s2);
    plot(X, m_0);
    plot(x_min, min_gross_mass, '.', MarkerSize=20);
    ylim([0, 1e4]);
    title(sprintf('Stage 1 Isp = %d s', Isp(isp)))
    legend("First Stage Mass", "Second Stage Mass", "Gross Vehicle Mass")
    xlabel("Delta V Fraction");
    ylabel("Mass (metric tons)")
    hold off;

end
```

```
Stage 1 Isp = 327 s --> Optimum at X = 0.456, Min Gross Mass = 3680.72 metric tons
Stage 1 Isp = 366 s --> Optimum at X = 0.500, Min Gross Mass = 2582.05 metric tons
Stage 1 Isp = 311 s --> Optimum at X = 0.438, Min Gross Mass = 4337.65 metric tons
Stage 1 Isp = 269 s --> Optimum at X = 0.395, Min Gross Mass = 7209.12 metric tons
Stage 1 Isp = 285 s --> Optimum at X = 0.411, Min Gross Mass = 5848.23 metric tons
```



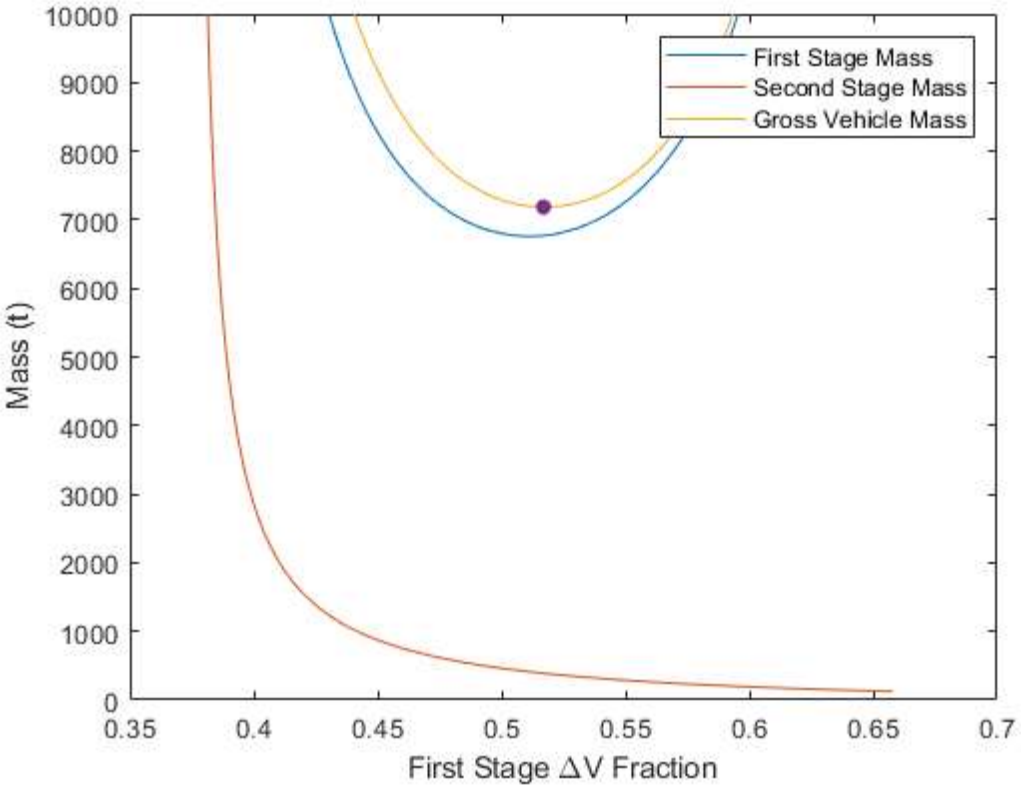
Published with MATLAB® R2024b

```
% solids row
% combo: first prop: LOX/LH2, second prop: solid
addpath('..')

% Set constants based on requirements and propellant combination
delta1 = 0.08;
delta2 = 0.08;
Isp1 = 366;
Isp2 = 269;
m_pl = 26000;

% Initialize/populate arrays
X = 0:0.001:1;
m_stage1_arr = [];
m_stage2_arr = [];
m0_arr = [];

% For each X value, call mass_function and populate the arrays for mass of
% stage 1, mass of stage 2, and total mass
for i = 1:length(X)
    [m_in1, m_in2, m_pr1, m_pr2, m0] = mass_function(Isp1, Isp2, X(i), delta1, delta2);
    m_stage1_arr(end+1) = m_in1 + m_pr1;
    m_stage2_arr(end+1) = m_in2 + m_pr2;
    m0_arr(end+1) = m0;
end

% Find the minimum gross mass, similar to min_gross_mass_function, but save
% the associated X for the minimum gross mass
min_m0 = realmax;
for i = 1:length(m0_arr)
    if m_stage1_arr(i) > 0 && m_stage2_arr(i) > 0 && m0_arr(i) < min_m0
        min_m0 = m0_arr(i);
        min_m0_X = X(i);
    end
end

% Plot mass of stage 1, mass of stage 2, and total mass in metric tons
% as a function of X. Also plot as a point the minimum gross mass
plot(X, m_stage1_arr/1000)
hold on
plot(X, m_stage2_arr/1000)
plot(X, m0_arr/1000)
plot(min_m0_X, min_m0/1000, '.', MarkerSize=20)
ylim([0, 1e4])
grid on
legend("First Stage Mass", "Second Stage Mass", "Gross Vehicle Mass", "Minimum Gross Mass")
xlabel("First Stage \DeltaV Fraction")
ylabel("Mass (t)")
title("Mass vs \DeltaV Fraction for First Stage: LOX/LH2, Second Stage: Solids")
```



Published with MATLAB® R2024b

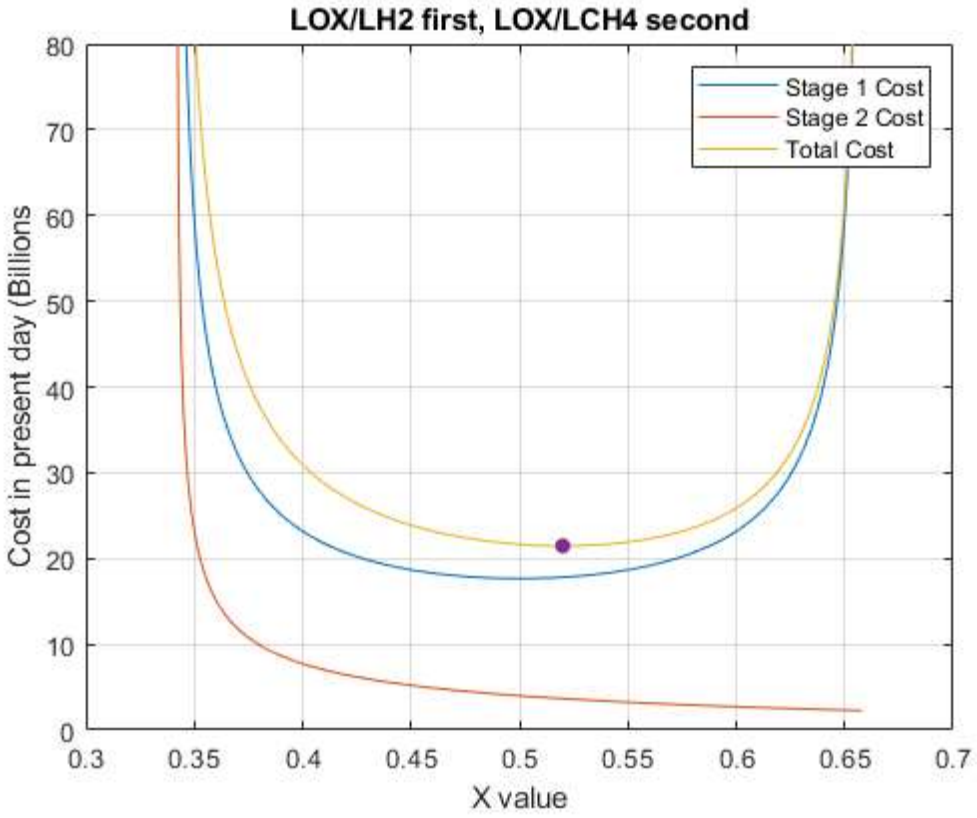
```
% LOX/RP1 row
% combo: first prop: LOX/LCH4, second prop: LOX/RPI
addpath('..')

% Set constants based on requirements and propellant combination
delta1 = 0.08;
delta2 = 0.08;
Isp1 = 327;
Isp2 = 311;
m_pl = 26000;

% Initialize/populate arrays
X = 0:0.001:1;
m_stage1_arr = [];
m_stage2_arr = [];
m0_arr = [];

% For each X value, call mass_function and populate the arrays for mass of
% stage 1, mass of stage 2, and total mass
for i = 1:length(X)
    [m_in1, m_in2, m_pr1, m_pr2, m0] = mass_function(Isp1, Isp2, X(i), delta1, delta2);
    m_stage1_arr(end+1) = m_in1 + m_pr1;
    m_stage2_arr(end+1) = m_in2 + m_pr2;
    m0_arr(end+1) = m0;
end

% Find the minimum gross mass, similar to min_gross_mass_function, but save
% the associated X for the minimum gross mass
min_m0 = realmax;
for i = 1:length(m0_arr)
    if m_stage1_arr(i) > 0 && m_stage2_arr(i) > 0 && m0_arr(i) < min_m0
        min_m0 = m0_arr(i);
        min_m0_X = X(i);
    end
end

% Plot mass of stage 1, mass of stage 2, and total mass in metric tons
% as a function of X. Also plot as a point the minimum gross mass
plot(X, m_stage1_arr/1000)
hold on
plot(X, m_stage2_arr/1000)
plot(X, m0_arr/1000)
plot(min_m0_X, min_m0/1000, '.', MarkerSize=20)
ylim([0, 1e4])
legend("First Stage Mass", "Second Stage Mass", "Gross Vehicle Mass")
xlabel("First Stage \DeltaV Fraction")
ylabel("Mass (t)")
```



Published with MATLAB® R2024b

```
% I have the LOX/LCH4 second stage row
% the first stage I am selecting is LOX/LH2 and the
% second stage is LOX/LCH4
addpath('..')
% given Constants
Delta_1 = 0.08;
Delta_2 = 0.08;
Isp_1 = 327;
Isp_2 = 327;
m_PL = 26000;

% the range of x from 0 to 1 at 0.01 intervals and the
% number of x values we will have
X = 0:0.001:1;
X_length = 1001;

% the arrays
Stage1_cost = zeros(1,X_length);
Stage2_cost = zeros(1,X_length);

% filling the arrays
S = 1;
while S <= X_length
    X_Position = X(S);
    % call cost function
    [stageCost_nr1, stageCost_nr2] = cost_function(Isp_1, Isp_2, X_Position, Delta_1, Delta_2);
    % defind and populate the arrays from these values
    Stage1_cost(S) = stageCost_nr1;
    Stage2_cost(S) = stageCost_nr2;

    S = S + 1;
end
%total it all together
Total_cost = Stage1_cost + Stage2_cost;

% finding the minimum cost
Minimum_Cost = inf;
Minimum_X = NaN;

S = 2;
while S <= X_length
    if (Stage1_cost(S) > 0) && (Stage2_cost(S) > 0) && (Total_cost(S) < Minimum_Cost)
        Minimum_Cost = Total_cost(S);
        Minimum_X = X(S);
    end
    S = S + 1;
end

% Converting to billion dollars

Stage1_cost = Stage1_cost / 1000;
Stage2_cost = Stage2_cost / 1000;
Total_cost = Total_cost / 1000;
Minimum_Cost = Minimum_Cost / 1000;

% making the plots
```

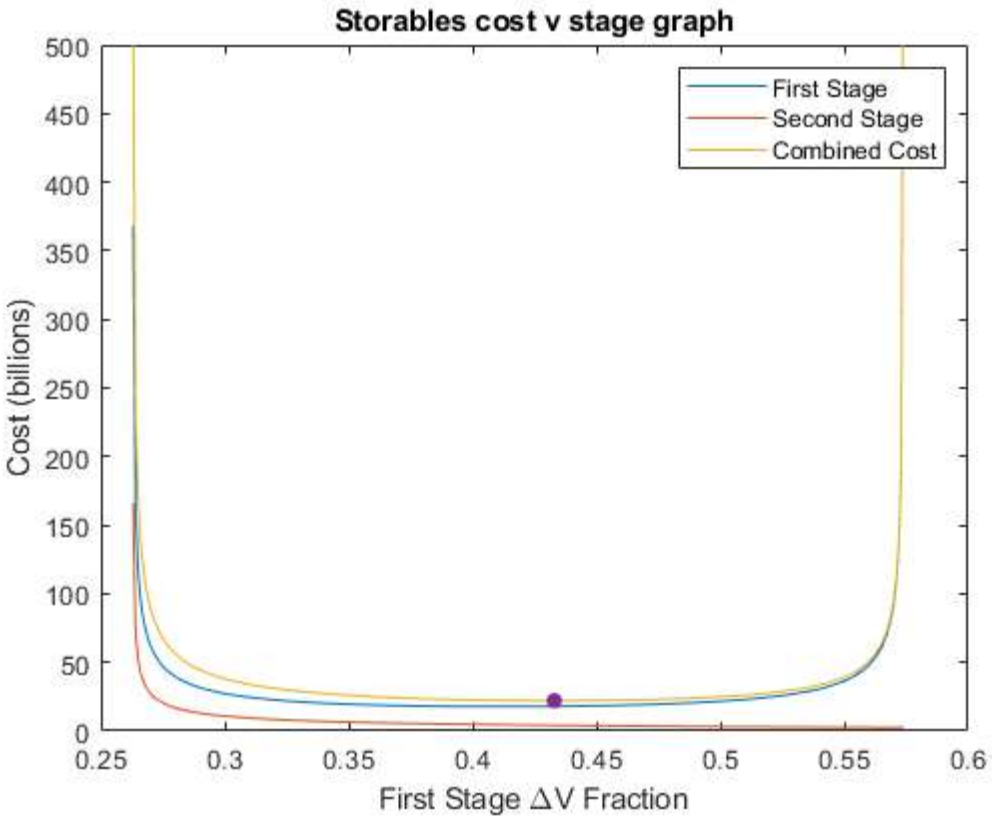
```
plot(X, Stage1_cost);
hold on;
plot(X, Stage2_cost);
plot(X, Total_cost);
plot(Minimum_X, Minimum_Cost, '.', 'MarkerSize',20)
ylim([0,80]);
xlabel('X value');
ylabel('Cost in present day (Billions');
title('LOX/LH2 first, LOX/LCH4 second');
legend('Stage 1 Cost', 'Stage 2 Cost', 'Total Cost');
grid on;
```



Published with MATLAB® R2024b

```
% TP 1
% Chris Witherspoon
% 1.3

addpath('..')

% Delta constants
d1 = 0.08 ;
d2 = 0.08 ;

% Specific impulse (storables)
isp = 285 ;
isp_comp = 366 ; % 2nd liquid

% Payload mass
mpl = 26000 ;

% Initialize the x array
x = 0:0.001:1 ;

% Create an empty array for later use
first_stage = [];
second_stage = [];

% Cost function loop
i = 1 ;
while i <= length(x)
    [first_stage_cost, second_stage_cost] = cost_function(isp, isp_comp, x(i), d1, d2);
    first_stage(end+1) = first_stage_cost;
    second_stage(end+1) = second_stage_cost;
    i = i + 1 ;
end

% Create an array for the total cost
total_cost = first_stage + second_stage ;

% Initialize the value to change for later
min = realmax;

% Find the value of the minimum cost
i = 1 ;
while i<= length(total_cost)
    if ~isnan(total_cost(i)) && total_cost(i) < min
        min = total_cost(i);
        min_cost_val = x(i);
    end
    i = i + 1 ;
end


% Plot the graph
plot(x, first_stage/1000)
hold on
plot(x, second_stage/1000)
plot(x, total_cost/1000)
plot(min_cost_val, min/1000, '.', MarkerSize=20)
```

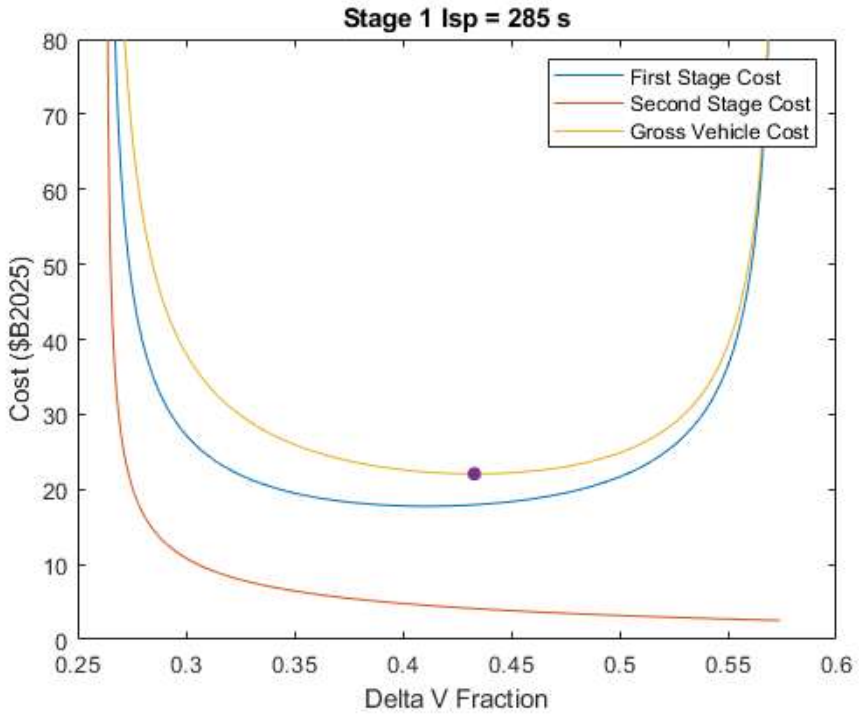
```
ylim([0, 500])
legend("First Stage", "Second Stage", "Combined Cost")
xlabel("First Stage \DeltaV Fraction")
ylabel("Cost (billions)")
title("Storables cost v stage graph") % Storables is the consistent
```



Published with MATLAB® R2024b

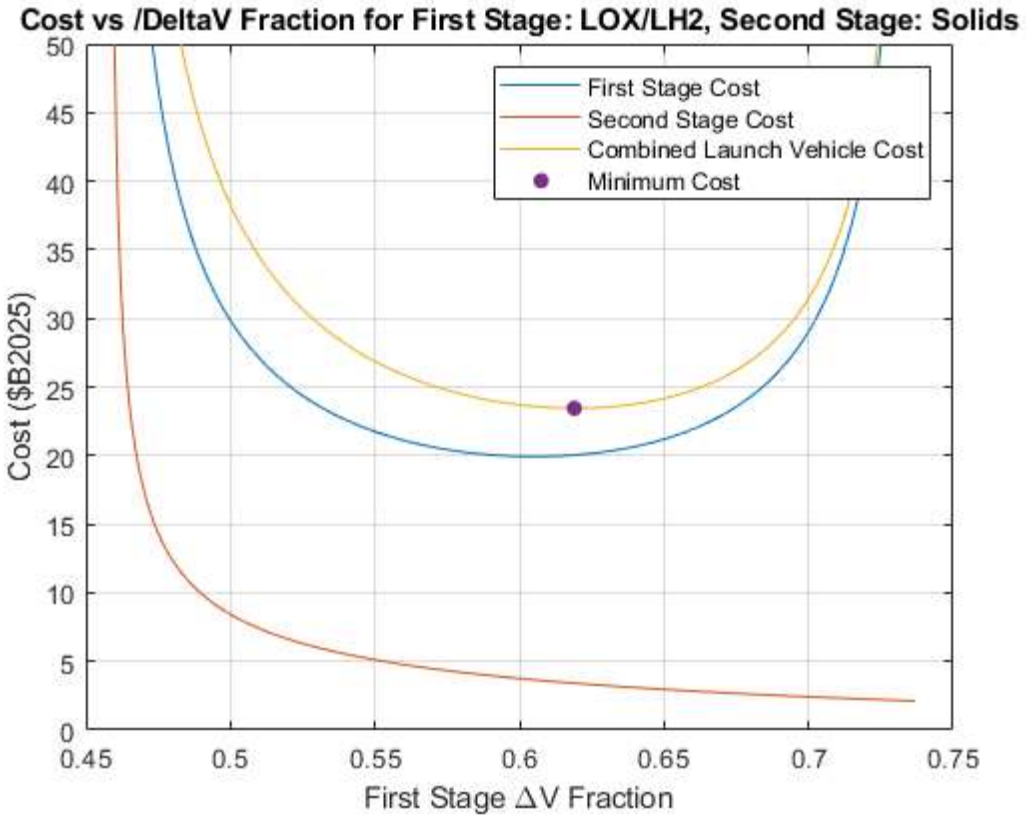
```
%Sara C 1.3a Code
%cost analysis

%Looking at mixture of:
%Stage 2: LOX/LH2
%Varied Stage 1: LOX/LCH4, LOX/LH2, LOX/RP1, Solid, Storables

%Variables & Given Parameters
delta1 = 0.08;
delta2 = 0.08;
payload = 26000; %kg
%Specific Impulse
Isp_s2 = 366; %stage 2
%LCH4, LH2, RP1, solids, storables
Isp = [327, 366, 311, 269, 285]; %stage 1 varies


%initialize X (this is delta_V ratio/fraction)
X=0:0.001:1;

for isp = 1:length(Isp)
    %mass arrays
    cost_s1 = [];
    cost_s2 = [];
    cost_tot = [];
    min_gross_cost = realmax;

    for i = 1:length(X)
        %calling mass function to populate mass arrays
        [stageCost_nr1, stageCost_nr2] = cost_function(Isp(isp), Isp_s2, X(i), delta1, delta2);
        cost_s1 = [cost_s1 stageCost_nr1];
        cost_s2 = [cost_s2 stageCost_nr2];
        cost_tot = [cost_tot (stageCost_nr1 + stageCost_nr2)];

        %min gross mass function - returns the x,y minimum of the m_0 array
        if cost_s1(i) > 0 && cost_s2(i) > 0 && cost_tot(i) < min_gross_cost
            min_gross_cost = cost_tot(i); %convert to metric tons
            x_min = X(i);
        end
    end

    %millions to billions
    cost_s1 = cost_s1/1000;
    cost_s2 = cost_s2/1000;
    cost_tot = cost_tot/1000;
    min_gross_cost = min_gross_cost/1000;

    %output - min cost and corresponding deltaV
    fprintf('Stage 1 Isp = %d s --> Optimum at X = %.3f, Min Gross Cost = %.2f $B2025 \n', Isp(isp), x_min, min_gross_cost);

    %plotting
    plot(X, cost_s1);
    hold on;
    plot(X, cost_s2);
    plot(X, cost_tot);
    plot(x_min, min_gross_cost, '.', MarkerSize=20);
    ylim([0, 80]);
    title(sprintf('Stage 1 Isp = %d s', Isp(isp)));
    legend("First Stage Cost", "Second Stage Cost", "Gross Vehicle Cost")
    xlabel("Delta V Fraction");
    ylabel("Cost ($B2025)")
    hold off;

end
```

```
Stage 1 Isp = 327 s --> Optimum at X = 0.487, Min Gross Cost = 17.30 $B2025
Stage 1 Isp = 366 s --> Optimum at X = 0.540, Min Gross Cost = 14.31 $B2025
Stage 1 Isp = 311 s --> Optimum at X = 0.466, Min Gross Cost = 18.87 $B2025
Stage 1 Isp = 269 s --> Optimum at X = 0.414, Min Gross Cost = 24.61 $B2025
Stage 1 Isp = 285 s --> Optimum at X = 0.433, Min Gross Cost = 22.07 $B2025
```



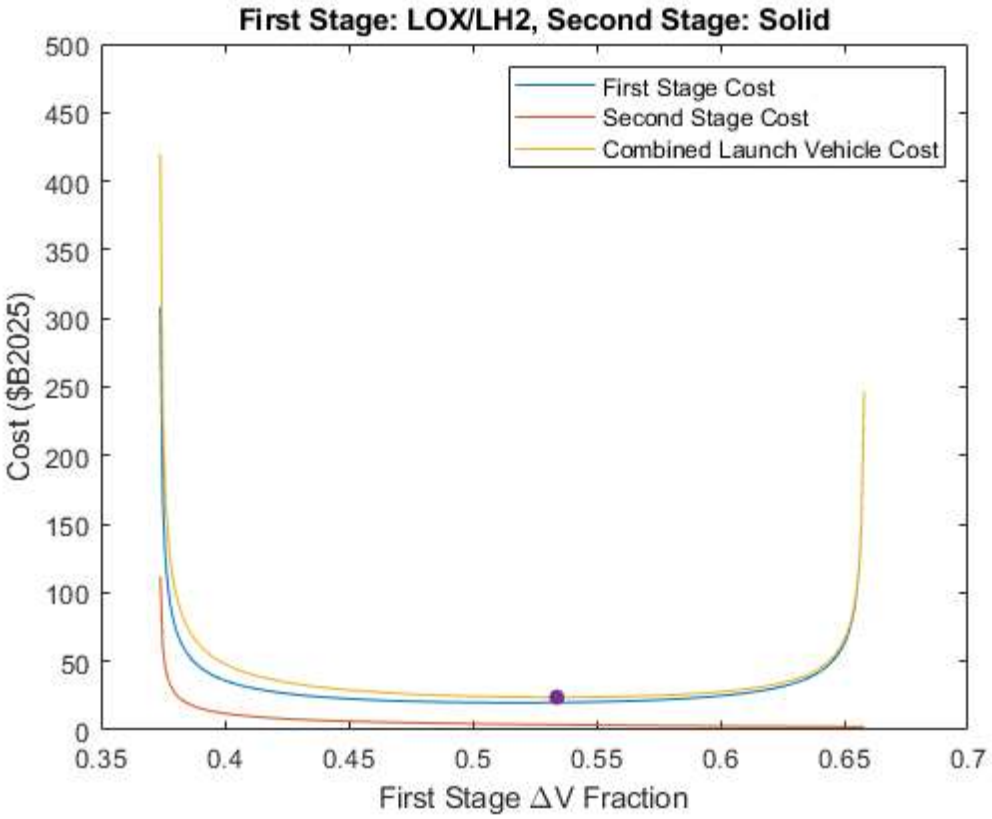
Published with MATLAB® R2024b

```
% solids row
% combo: first prop: LOX/LH2, second prop: solid
addpath('..')

% Set constants based on requirements and propellant combination
delta1 = 0.08;
delta2 = 0.08;
Isp1 = 366;
Isp2 = 269;
m_pl = 26000;

% Initialize/populate arrays
X = 0:0.001:1;
cost_stage1_arr = [];
cost_stage2_arr = [];

% For each X value, call cost_function and populate the arrays for cost of
% stage 1 and mass of stage 2
for i = 1:length(X)
    [stageCost_nr1, stageCost_nr2] = cost_function(Isp1, Isp2, X(i), delta1, delta2);
    cost_stage1_arr(end+1) = stageCost_nr1;
    cost_stage2_arr(end+1) = stageCost_nr2;
end

% Add the two arrays to get an array of the total cost
total_cost = cost_stage1_arr + cost_stage2_arr;

% Find the minimum gross mass, similar to min_cost_function, but save
% the associated X for the minimum total cost
min_cost = realmax;
for i = 1:length(total_cost)
    if ~isnan(total_cost(i)) && total_cost(i) < min_cost
        min_cost = total_cost(i);
        min_cost_X = X(i);
    end
end

% Plot mass of stage 1, mass of stage 2, and total mass in billions of dollars
% as a function of X. Also plot as a point the minimum total cost
plot(X, cost_stage1_arr/1000)
hold on
plot(X, cost_stage2_arr/1000)
plot(X, total_cost/1000)
plot(min_cost_X, min_cost/1000, '.', MarkerSize=20)
grid on
ylim([0, 50])
legend("First Stage Cost", "Second Stage Cost", "Combined Launch Vehicle Cost", "Minimum Cost")
xlabel("First Stage \DeltaV Fraction")
ylabel("Cost ($B2025)")
title("Cost vs /DeltaV Fraction for First Stage: LOX/LH2, Second Stage: Solids")
```



Published with MATLAB® R2024b

```
% LOX/RP1 row
% combo: first prop: LOX/LCH4, second prop: LOX/RPI
addpath('..')

% Set constants based on requirements and propellant combination
delta1 = 0.08;
delta2 = 0.08;
Isp1 = 327;
Isp2 = 311;
m_pl = 26000;

% Initialize/populate arrays
X = 0:0.001:1;
cost_stage1_arr = [];
cost_stage2_arr = [];

% For each X value, call cost_function and populate the arrays for cost of
% stage 1 and mass of stage 2
for i = 1:length(X)
    [stageCost_nr1, stageCost_nr2] = cost_function(Isp1, Isp2, X(i), delta1, delta2);
    cost_stage1_arr(end+1) = stageCost_nr1;
    cost_stage2_arr(end+1) = stageCost_nr2;
end

% Add the two arrays to get an array of the total cost
total_cost = cost_stage1_arr + cost_stage2_arr;

% Find the minimum gross mass, similar to min_cost_function, but save
% the associated X for the minimum total cost
min_cost = realmax;
for i = 1:length(total_cost)
    if ~isnan(total_cost(i)) && total_cost(i) < min_cost
        min_cost = total_cost(i);
        min_cost_X = X(i);
    end
end

% Plot mass of stage 1, mass of stage 2, and total mass in billions of dollars
% as a function of X. Also plot as a point the minimum total cost
plot(X, cost_stage1_arr/1000)
hold on
plot(X, cost_stage2_arr/1000)
plot(X, total_cost/1000)
plot(min_cost_X, min_cost/1000, '.', MarkerSize=20)
ylim([0, 500])
legend("First Stage Cost", "Second Stage Cost", "Combined Launch Vehicle Cost")
xlabel("First Stage \DeltaV Fraction")
ylabel("Cost ($B2025)")
title("First Stage: LOX/LH2, Second Stage: Solid")
```



Published with MATLAB® R2024b

```
solid = readmatrix('solid.csv', 'ThousandsSeparator', ',');
RP1 = readmatrix('RP1.csv', 'ThousandsSeparator', ',');
LCH4 = readmatrix('LCH4.csv', 'ThousandsSeparator', ',');
LH2 = readmatrix('LH2.csv', 'ThousandsSeparator', ',');
storable = readmatrix('storable.csv', 'ThousandsSeparator', ',');

LCH4_min_mass_MASS_solid = solid(1,1);
LH2_min_mass_MASS_solid  = solid(1,2);
RP1_min_mass_MASS_solid  = solid(1,3);
solid_min_mass_MASS_solid  = solid(1,4);
storable_min_mass_MASS_solid  = solid(1,5);

LCH4_min_mass_COST_solid  = solid(2,1);
LH2_min_mass_COST_solid  = solid(2,2);
RP1_min_mass_COST_solid  = solid(2,3);
solid_min_mass_COST_solid  = solid(2,4);
storable_min_mass_COST_solid  = solid(2,5);

LCH4_min_cost_COST_solid  = solid(3,1);
LH2_min_cost_COST_solid  = solid(3,2);
RP1_min_cost_COST_solid  = solid(3,3);
solid_min_cost_COST_solid  = solid(3,4);
storable_min_cost_COST_solid  = solid(3,5);

LCH4_min_cost_MASS_solid  = solid(4,1);
LH2_min_cost_MASS_solid  = solid(4,2);
RP1_min_cost_MASS_solid  = solid(4,3);
solid_min_cost_MASS_solid  = solid(4,4);
storable_min_cost_MASS_solid  = solid(4,5);

LCH4_min_mass_MASS_RP1 = RP1(1,1);
LH2_min_mass_MASS_RP1 = RP1(1,2);
RP1_min_mass_MASS_RP1 = RP1(1,3);
solid_min_mass_MASS_RP1 = RP1(1,4);
storable_min_mass_MASS_RP1 = RP1(1,5);

LCH4_min_mass_COST_RP1 = RP1(2,1);
LH2_min_mass_COST_RP1 = RP1(2,2);
RP1_min_mass_COST_RP1 = RP1(2,3);
solid_min_mass_COST_RP1 = RP1(2,4);
storable_min_mass_COST_RP1 = RP1(2,5);

LCH4_min_cost_COST_RP1 = RP1(3,1);
LH2_min_cost_COST_RP1 = RP1(3,2);
RP1_min_cost_COST_RP1 = RP1(3,3);
solid_min_cost_COST_RP1 = RP1(3,4);
storable_min_cost_COST_RP1 = RP1(3,5);

LCH4_min_cost_MASS_RP1 = RP1(4,1);
LH2_min_cost_MASS_RP1 = RP1(4,2);
RP1_min_cost_MASS_RP1 = RP1(4,3);
solid_min_cost_MASS_RP1 = RP1(4,4);
storable_min_cost_MASS_RP1 = RP1(4,5);

LCH4_min_mass_MASS_LCH4 = LCH4(1,1);
LH2_min_mass_MASS_LCH4 = LCH4(1,2);
RP1_min_mass_MASS_LCH4 = LCH4(1,3);
solid_min_mass_MASS_LCH4 = LCH4(1,4);
storable_min_mass_MASS_LCH4 = LCH4(1,5);

LCH4_min_mass_COST_LCH4 = LCH4(2,1);
LH2_min_mass_COST_LCH4 = LCH4(2,2);
RP1_min_mass_COST_LCH4 = LCH4(2,3);
solid_min_mass_COST_LCH4 = LCH4(2,4);
storable_min_mass_COST_LCH4 = LCH4(2,5);

LCH4_min_cost_COST_LCH4 = LCH4(3,1);
LH2_min_cost_COST_LCH4 = LCH4(3,2);
RP1_min_cost_COST_LCH4 = LCH4(3,3);
solid_min_cost_COST_LCH4 = LCH4(3,4);
storable_min_cost_COST_LCH4 = LCH4(3,5);

LCH4_min_cost_MASS_LCH4 = LCH4(4,1);
LH2_min_cost_MASS_LCH4 = LCH4(4,2);
RP1_min_cost_MASS_LCH4 = LCH4(4,3);
solid_min_cost_MASS_LCH4 = LCH4(4,4);
storable_min_cost_MASS_LCH4 = LCH4(4,5);

LCH4_min_mass_MASS_LH2 = LH2(1,1);
LH2_min_mass_MASS_LH2 = LH2(1,2);
RP1_min_mass_MASS_LH2 = LH2(1,3);
solid_min_mass_MASS_LH2 = LH2(1,4);
storable_min_mass_MASS_LH2 = LH2(1,5);

LCH4_min_mass_COST_LH2 = LH2(2,1);
```

```
LH2_min_mass_COST_LH2 = LH2(2,2);
RP1_min_mass_COST_LH2 = LH2(2,3);
solid_min_mass_COST_LH2 = LH2(2,4);
storable_min_mass_COST_LH2 = LH2(2,5);

LCH4_min_cost_COST_LH2 = LH2(3,1);
LH2_min_cost_COST_LH2 = LH2(3,2);
RP1_min_cost_COST_LH2 = LH2(3,3);
solid_min_cost_COST_LH2 = LH2(3,4);
storable_min_cost_COST_LH2 = LH2(3,5);

LCH4_min_cost_MASS_LH2 = LH2(4,1);
LH2_min_cost_MASS_LH2 = LH2(4,2);
RP1_min_cost_MASS_LH2 = LH2(4,3);
solid_min_cost_MASS_LH2 = LH2(4,4);
storable_min_cost_MASS_LH2 = LH2(4,5);

LCH4_min_mass_MASS_storable = storable(1,1);
LH2_min_mass_MASS_storable = storable(1,2);
RP1_min_mass_MASS_storable = storable(1,3);
solid_min_mass_MASS_storable = storable(1,4);
storable_min_mass_MASS_storable = storable(1,5);

LCH4_min_mass_COST_storable = storable(2,1);
LH2_min_mass_COST_storable = storable(2,2);
RP1_min_mass_COST_storable = storable(2,3);
solid_min_mass_COST_storable = storable(2,4);
storable_min_mass_COST_storable = storable(2,5);

LCH4_min_cost_COST_storable = storable(3,1);
LH2_min_cost_COST_storable = storable(3,2);
RP1_min_cost_COST_storable = storable(3,3);
solid_min_cost_COST_storable = storable(3,4);
storable_min_cost_COST_storable = storable(3,5);

LCH4_min_cost_MASS_storable = storable(4,1);
LH2_min_cost_MASS_storable = storable(4,2);
RP1_min_cost_MASS_storable = storable(4,3);
solid_min_cost_MASS_storable = storable(4,4);
storable_min_cost_MASS_storable = storable(4,5);


figure;
plot(LCH4_min_mass_MASS_solid, LCH4_min_mass_COST_solid, 'o', 'MarkerSize', 15, 'LineWidth', 2, 'Color', 'r')
hold on
plot(LCH4_min_cost_MASS_solid, LCH4_min_cost_COST_solid, 'o', 'MarkerSize', 15, 'LineWidth', 2, 'Color', 'r')

plot(LH2_min_mass_MASS_solid, LH2_min_mass_COST_solid, '*', 'MarkerSize', 15, 'LineWidth', 2, 'Color', 'r')
plot(LH2_min_cost_MASS_solid, LH2_min_cost_COST_solid, '*', 'MarkerSize', 15, 'LineWidth', 2, 'Color', 'r')

plot(RP1_min_mass_MASS_solid, RP1_min_mass_COST_solid, 'x', 'MarkerSize', 15, 'LineWidth', 2, 'Color', 'r')
plot(RP1_min_cost_MASS_solid, RP1_min_cost_COST_solid, 'x', 'MarkerSize', 15, 'LineWidth', 2, 'Color', 'r')

plot(solid_min_mass_MASS_solid, solid_min_mass_COST_solid, 'diamond', 'MarkerSize', 15, 'LineWidth', 2, 'Color', 'r')
plot(solid_min_cost_MASS_solid, solid_min_cost_COST_solid, 'diamond', 'MarkerSize', 15, 'LineWidth', 2, 'Color', 'r')

plot(storable_min_mass_MASS_solid, storable_min_mass_COST_solid, '^', 'MarkerSize', 15, 'LineWidth', 2, 'Color', 'r')
plot(storable_min_cost_MASS_solid, storable_min_cost_COST_solid, '^', 'MarkerSize', 15, 'LineWidth', 2, 'Color', 'r')


plot(LCH4_min_mass_MASS_RP1, LCH4_min_mass_COST_RP1, 'o', 'MarkerSize', 15, 'LineWidth', 2, 'Color', [0 52 0]/255)
plot(LCH4_min_cost_MASS_RP1, LCH4_min_cost_COST_RP1, 'o', 'MarkerSize', 15, 'LineWidth', 2, 'Color', [0 52 0]/255)

plot(LH2_min_mass_MASS_RP1, LH2_min_mass_COST_RP1, '*', 'MarkerSize', 15, 'LineWidth', 2, 'Color', [0 52 0]/255)
plot(LH2_min_cost_MASS_RP1, LH2_min_cost_COST_RP1, '*', 'MarkerSize', 15, 'LineWidth', 2, 'Color', [0 52 0]/255)

plot(RP1_min_mass_MASS_RP1, RP1_min_mass_COST_RP1, 'x', 'MarkerSize', 15, 'LineWidth', 2, 'Color', [0 52 0]/255)
plot(RP1_min_cost_MASS_RP1, RP1_min_cost_COST_RP1, 'x', 'MarkerSize', 15, 'LineWidth', 2, 'Color', [0 52 0]/255)

plot(solid_min_mass_MASS_RP1, solid_min_mass_COST_RP1, 'diamond', 'MarkerSize', 15, 'LineWidth', 2, 'Color', [0 52 0]/255)
plot(solid_min_cost_MASS_RP1, solid_min_cost_COST_RP1, 'diamond', 'MarkerSize', 15, 'LineWidth', 2, 'Color', [0 52 0]/255)

plot(storable_min_mass_MASS_RP1, storable_min_mass_COST_RP1, '^', 'MarkerSize', 15, 'LineWidth', 2, 'Color', [0 52 0]/255)
plot(storable_min_cost_MASS_RP1, storable_min_cost_COST_RP1, '^', 'MarkerSize', 15, 'LineWidth', 2, 'Color', [0 52 0]/255)

plot(LCH4_min_mass_MASS_LCH4, LCH4_min_mass_COST_LCH4, 'o', 'MarkerSize', 15, 'LineWidth', 2, 'Color', 'b')
plot(LCH4_min_cost_MASS_LCH4, LCH4_min_cost_COST_LCH4, 'o', 'MarkerSize', 15, 'LineWidth', 2, 'Color', 'b')

plot(LH2_min_mass_MASS_LCH4, LH2_min_mass_COST_LCH4, '*', 'MarkerSize', 15, 'LineWidth', 2, 'Color', 'b')
plot(LH2_min_cost_MASS_LCH4, LH2_min_cost_COST_LCH4, '*', 'MarkerSize', 15, 'LineWidth', 2, 'Color', 'b')

plot(RP1_min_mass_MASS_LCH4, RP1_min_mass_COST_LCH4, 'x', 'MarkerSize', 15, 'LineWidth', 2, 'Color', 'b')
plot(RP1_min_cost_MASS_LCH4, RP1_min_cost_COST_LCH4, 'x', 'MarkerSize', 15, 'LineWidth', 2, 'Color', 'b')

plot(solid_min_mass_MASS_LCH4, solid_min_mass_COST_LCH4, 'diamond', 'MarkerSize', 15, 'LineWidth', 2, 'Color', 'b')
plot(solid_min_cost_MASS_LCH4, solid_min_cost_COST_LCH4, 'diamond', 'MarkerSize', 15, 'LineWidth', 2, 'Color', 'b')
```

```
plot(storable_min_mass_MASS_LCH4, storable_min_mass_COST_LCH4, '^', 'MarkerSize', 15, 'LineWidth', 2, 'Color', 'b')
plot(storable_min_cost_MASS_LCH4, storable_min_cost_COST_LCH4, '^', 'MarkerSize', 15, 'LineWidth', 2, 'Color', 'b')

plot(LCH4_min_mass_MASS_LH2, LCH4_min_mass_COST_LH2, 'o', 'MarkerSize', 15, 'LineWidth', 2, 'Color', [204 0 204]/255)
plot(LCH4_min_cost_MASS_LH2, LCH4_min_cost_COST_LH2, 'o', 'MarkerSize', 15, 'LineWidth', 2, 'Color', [204 0 204]/255)

plot(LH2_min_mass_MASS_LH2, LH2_min_mass_COST_LH2, '*', 'MarkerSize', 15, 'LineWidth', 2, 'Color', [204 0 204]/255)
plot(LH2_min_cost_MASS_LH2, LH2_min_cost_COST_LH2, '*', 'MarkerSize', 15, 'LineWidth', 2, 'Color', [204 0 204]/255)

plot(RP1_min_mass_MASS_LH2, RP1_min_mass_COST_LH2, 'x', 'MarkerSize', 15, 'LineWidth', 2, 'Color', [204 0 204]/255)
plot(RP1_min_cost_MASS_LH2, RP1_min_cost_COST_LH2, 'x', 'MarkerSize', 15, 'LineWidth', 2, 'Color', [204 0 204]/255)

plot(solid_min_mass_MASS_LH2, solid_min_mass_COST_LH2, 'diamond', 'MarkerSize', 15, 'LineWidth', 2, 'Color', [204 0 204]/255)
plot(solid_min_cost_MASS_LH2, solid_min_cost_COST_LH2, 'diamond', 'MarkerSize', 15, 'LineWidth', 2, 'Color', [204 0 204]/255)

plot(storable_min_mass_MASS_LH2, storable_min_mass_COST_LH2, '^', 'MarkerSize', 15, 'LineWidth', 2, 'Color', [204 0 204]/255)
plot(storable_min_cost_MASS_LH2, storable_min_cost_COST_LH2, '^', 'MarkerSize', 15, 'LineWidth', 2, 'Color', [204 0 204]/255)

plot(LCH4_min_mass_MASS_storable, LCH4_min_mass_COST_storable, 'o', 'MarkerSize', 15, 'LineWidth', 2, 'Color', [255 153 51]/255)
plot(LCH4_min_cost_MASS_storable, LCH4_min_cost_COST_storable, 'o', 'MarkerSize', 15, 'LineWidth', 2, 'Color', [255 153 51]/255)

plot(LH2_min_mass_MASS_storable, LH2_min_mass_COST_storable, '*', 'MarkerSize', 15, 'LineWidth', 2, 'Color', [255 153 51]/255)
plot(LH2_min_cost_MASS_storable, LH2_min_cost_COST_storable, '*', 'MarkerSize', 15, 'LineWidth', 2, 'Color', [255 153 51]/255)

plot(RP1_min_mass_MASS_storable, RP1_min_mass_COST_storable, 'x', 'MarkerSize', 15, 'LineWidth', 2, 'Color', [255 153 51]/255)
plot(RP1_min_cost_MASS_storable, RP1_min_cost_COST_storable, 'x', 'MarkerSize', 15, 'LineWidth', 2, 'Color', [255 153 51]/255)

plot(solid_min_mass_MASS_storable, solid_min_mass_COST_storable, 'diamond', 'MarkerSize', 15, 'LineWidth', 2, 'Color', [255 153 51]/255)
plot(solid_min_cost_MASS_storable, solid_min_cost_COST_storable, 'diamond', 'MarkerSize', 15, 'LineWidth', 2, 'Color', [255 153 51]/255)

plot(storable_min_mass_MASS_storable, storable_min_mass_COST_storable, '^', 'MarkerSize', 15, 'LineWidth', 2, 'Color', [255 153 51]/255)
plot(storable_min_cost_MASS_storable, storable_min_cost_COST_storable, '^', 'MarkerSize', 15, 'LineWidth', 2, 'Color', [255 153 51]/255)

red = plot(nan, nan, 'o', 'MarkerFaceColor', 'r', 'MarkerEdgeColor', 'r', 'MarkerSize', 10);
blue = plot(nan, nan, 'o', 'MarkerFaceColor', 'b', 'MarkerEdgeColor', 'b', 'MarkerSize', 10);
green = plot(nan, nan, 'o', 'MarkerFaceColor', [0 52 0]/255, 'MarkerEdgeColor', [0 52 0]/255, 'MarkerSize', 10);
orange = plot(nan, nan, 'o', 'MarkerFaceColor', [255 153 51]/255, 'MarkerEdgeColor', [255 153 51]/255, 'MarkerSize', 10);
purple = plot(nan, nan, 'o', 'MarkerFaceColor', [204 0 204]/255, 'MarkerEdgeColor', [204 0 204]/255, 'MarkerSize', 10);

circle = plot(nan, nan, 'o', 'Color', 'k', 'MarkerSize', 10);
star = plot(nan, nan, '*', 'Color', 'k', 'MarkerSize', 10);
x = plot(nan, nan, 'x', 'Color', 'k', 'MarkerSize', 10);
diamond = plot(nan, nan, 'diamond', 'Color', 'k', 'MarkerSize', 10);
triangle = plot(nan, nan, '^', 'Color', 'k', 'MarkerSize', 10);

legend([circle, star, x, diamond, triangle, blue, purple, green, red, orange], 'First Stage: LOX/LCH4', 'First Stage: LOX/LH2', 'First Stage: LOX/RP1', 'First Stage

xlabel('Mass (t)')
ylabel('Cost ($B2025)')
title('Minimum Cost and Minimum Mass Propellant Mix Designs')
grid on
```

```
Error using readmatrix (line 171)
Unable to find or open 'solid.csv'. Check the path and filename or file permissions.

Error in full_plot (line 1)
solid = readmatrix('solid.csv', 'ThousandsSeparator', ',');
        ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
```

Published with MATLAB® R2024b

```
solid = readmatrix('solid.csv', 'ThousandsSeparator', ',');
RP1 = readmatrix('RP1.csv', 'ThousandsSeparator', ',');
LCH4 = readmatrix('LCH4.csv', 'ThousandsSeparator', ',');
LH2 = readmatrix('LH2.csv', 'ThousandsSeparator', ',');
storable = readmatrix('storable.csv', 'ThousandsSeparator', ',');

LH2_min_mass_MASS_solid  = solid(1,2);
LH2_min_mass_COST_solid  = solid(2,2);
LH2_min_cost_COST_solid  = solid(3,2);
LH2_min_cost_MASS_solid  = solid(4,2);

LH2_min_mass_MASS_RP1 = RP1(1,2);
LH2_min_mass_COST_RP1 = RP1(2,2);
LH2_min_cost_COST_RP1 = RP1(3,2);
LH2_min_cost_MASS_RP1 = RP1(4,2);

LH2_min_mass_MASS_LCH4 = LCH4(1,2);
LH2_min_mass_COST_LCH4 = LCH4(2,2);
LH2_min_cost_COST_LCH4 = LCH4(3,2);
LH2_min_cost_MASS_LCH4 = LCH4(4,2);

LH2_min_mass_MASS_LH2 = LH2(1,2);
LH2_min_mass_COST_LH2 = LH2(2,2);
LH2_min_cost_COST_LH2 = LH2(3,2);
LH2_min_cost_MASS_LH2 = LH2(4,2);

LH2_min_mass_MASS_storable = storable(1,2);
LH2_min_mass_COST_storable = storable(2,2);
LH2_min_cost_COST_storable = storable(3,2);
LH2_min_cost_MASS_storable = storable(4,2);

figure;
plot(LH2_min_mass_MASS_solid, LH2_min_mass_COST_solid, '*', 'MarkerSize', 15, 'LineWidth', 2, 'Color', 'r')
hold on
plot(LH2_min_cost_MASS_solid, LH2_min_cost_COST_solid, '*', 'MarkerSize', 15, 'LineWidth', 2, 'Color', 'r')

plot(LH2_min_mass_MASS_RP1, LH2_min_mass_COST_RP1, '*', 'MarkerSize', 15, 'LineWidth', 2, 'Color', [0 52 0]/255)
plot(LH2_min_cost_MASS_RP1, LH2_min_cost_COST_RP1, '*', 'MarkerSize', 15, 'LineWidth', 2, 'Color', [0 52 0]/255)

plot(LH2_min_mass_MASS_LCH4, LH2_min_mass_COST_LCH4, '*', 'MarkerSize', 15, 'LineWidth', 2, 'Color', 'b')
plot(LH2_min_cost_MASS_LCH4, LH2_min_cost_COST_LCH4, '*', 'MarkerSize', 15, 'LineWidth', 2, 'Color', 'b')

plot(LH2_min_mass_MASS_LH2, LH2_min_mass_COST_LH2, '*', 'MarkerSize', 15, 'LineWidth', 2, 'Color', [204 0 204]/255)
plot(LH2_min_cost_MASS_LH2, LH2_min_cost_COST_LH2, '*', 'MarkerSize', 15, 'LineWidth', 2, 'Color', [204 0 204]/255)

plot(LH2_min_mass_MASS_storable, LH2_min_mass_COST_storable, '*', 'MarkerSize', 15, 'LineWidth', 2, 'Color', [255 153 51]/255)
plot(LH2_min_cost_MASS_storable, LH2_min_cost_COST_storable, '*', 'MarkerSize', 15, 'LineWidth', 2, 'Color', [255 153 51]/255)

red = plot(nan, nan, 'o', 'MarkerFaceColor', 'r', 'MarkerEdgeColor', 'r', 'MarkerSize', 10);
blue = plot(nan, nan, 'o', 'MarkerFaceColor', 'b', 'MarkerEdgeColor', 'b', 'MarkerSize', 10);
green = plot(nan, nan, 'o', 'MarkerFaceColor', [0 52 0]/255, 'MarkerEdgeColor', [0 52 0]/255, 'MarkerSize', 10);
orange = plot(nan, nan, 'o', 'MarkerFaceColor', [255 153 51]/255, 'MarkerEdgeColor', [255 153 51]/255, 'MarkerSize', 10);
purple = plot(nan, nan, 'o', 'MarkerFaceColor', [204 0 204]/255, 'MarkerEdgeColor', [204 0 204]/255, 'MarkerSize', 10);

circle = plot(nan, nan, 'o', 'Color', 'k', 'MarkerSize', 10);
star = plot(nan, nan, '*', 'Color', 'k', 'MarkerSize', 10);
x = plot(nan, nan, 'x', 'Color', 'k', 'MarkerSize', 10);
diamond = plot(nan, nan, 'diamond', 'Color', 'k', 'MarkerSize', 10);
triangle = plot(nan, nan, '^', 'Color', 'k', 'MarkerSize', 10);

legend([circle, star, x, diamond, triangle, blue, purple, green, red, orange], 'First Stage: LOX/LCH4', 'First Stage: LOX/LH2', 'First Stage: LOX/RP1', 'First Stage

xlabel('Mass (t)')
ylabel('Cost ($B2025)')
title('Minimum Cost and Minimum Mass Propellant Mix Designs')
grid on
```

```
Error using readmatrix (line 171)
Unable to find or open 'solid.csv'. Check the path and filename or file permissions.

Error in best_combos_plot (line 1)
solid = readmatrix('solid.csv', 'ThousandsSeparator', ',');
        ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
```

Published with MATLAB® R2024b