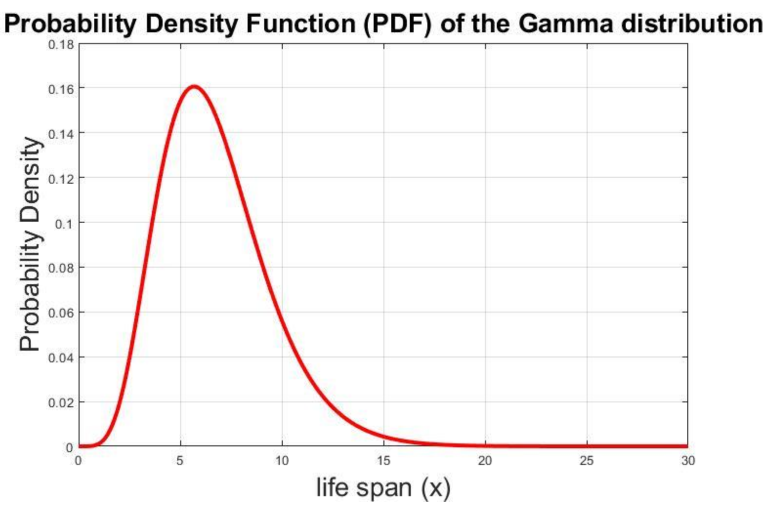
Solution for Assignment 8 (CEE 3804)

Problem 1:

Task1:

```matlab
%%%%%%%%%%%%%%Problem 1 Task 1%%%%%%%%%%%%%%%
%script to determine the Probability Density Function (PDF) of the Gamma distribution
%programmed by: Armin Zolfaghari
close all
clear
clc

%Assign desired values to alpha and beta
alpha = 6.40;
beta = 0.95;

%Define the x (life span) vector
x = 0:0.1:30;

%formula to determine Probability Density Function (PDF) of the Gamma distribution
fx = beta^alpha.* x.^(alpha-1).*exp(-beta.*x)/gamma(alpha);
%plot x vs probability
plot(x,fx, '-r','LineWidth',3)
title('Probability Density Function (PDF) of the Gamma distribution','FontSize', 20)
xlabel('life span (x)','FontSize', 20)
ylabel('Probability Density','FontSize', 20)
grid
```



Task 2:

```matlab
%%%%%%%%%%%%%%Problem 1 Task 2%%%%%%%%%%%%%%%
%call a matlab function to calculate the value of the Gamma distribution function f(x)

close all
clear
clc

%Assign desired values to alpha and beta
alpha = 6.40;
beta = 0.95;

%Define the x (life span) vector
x = 0:0.1:30;
%call function
[probability] = PDF_gamma (x,alpha,beta);

%plot x vs probability
plot(x,probability, '-r','LineWidth',3)
title('Probability Density Function (PDF) of the Gamma distribution','FontSize', 20)
xlabel('Life span (x)','FontSize', 20)
ylabel('Probability Density','FontSize', 20)
grid
```

```matlab
function [fx] = PDF_gamma(x,alpha,beta)



    fx = beta^alpha.* x.^(alpha-1).*exp(-beta.*x)/gamma(alpha);


end
```

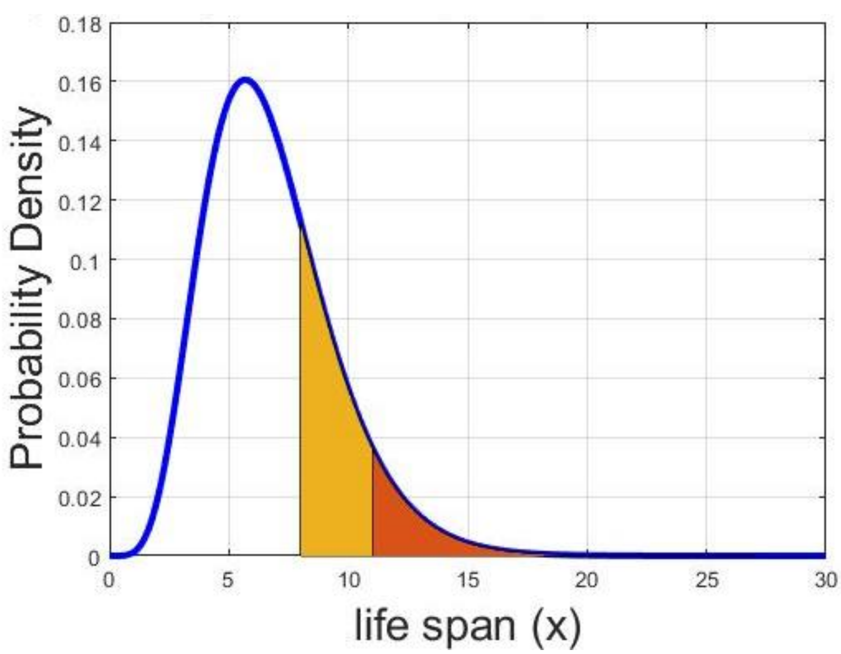
Tasks 3 and 4:

```matlab
%%%%%%%%%%%%%Problem 1 Task 3%%%%%%%%%%%%%%%
%script to determine the Probability Density Function (PDF) of the Gamma distribution
%programmed by: Armin Zolfaghari
close all
clear
clc

%Assign desired values to alpha and beta
alpha = 6.40;
beta = 0.95;

%Define the x (life span) vector
x = 0:0.1:30;

%call function
[probability] = PDF_gamma (x,alpha,beta);

%plot x vs probability
plot(x,probability, '-b','LineWidth',3)
title('Probability Density Function (PDF) of the Gamma distribution','FontSize', 20)
xlabel('life span (x)','FontSize', 20)
ylabel('Probability Density','FontSize', 20)
grid

hold on

%define the minimum value for x
lower_bound = 8;
%define the maximum value for x
upper_bound = 30;

%vector of all the numbers from minimum to maximum
life_bound = lower_bound:0.1:upper_bound;

```

Command Window

```
The probably that the asphalt life span is more than eight years is 28.17%
The probably that the asphalt life span is between eight years and eleven years is 21.19%
The median life span of asphalt is 6.4 years
fx >>
```

```
%call PDF_gamma function
[fx_area] = PDF_gamma(life_bound,alpha,beta);
%plot are from lower bound to upper bound
area (life_bound,fx_area);
%calculate the area under the curve from lower bound to upper bound
area_probability = trapz(life_bound,fx_area);
percentage = round(area_probability*100,2);
disp(['The probably that the asphalt life span is more than eight years is',' ',num2str(percentage),'%'])
hold on

%%%%%%%%%%%%%Problem 1 Task 4%%%%%%%%%%%%%%%
%define the minimum value for x
lower_bound = 8;
%define the maximum value for x
upper_bound = 11;

%vector of all numbers from minimum to maximum
life_bound = lower_bound:0.1:upper_bound;
%call PDF_gamma function
[fx_area] = PDF_gamma(life_bound,alpha,beta);
%plot from lower bound to upper bound
area (life_bound,fx_area);
%calculate the area under the curve from lower bound to upper bound
area_probability = trapz(life_bound,fx_area);
percentage = round(area_probability*100,2);
disp(['The probably that the asphalt life span is between eight years and eleven years is',' ',num2str(percentage),'%'])

%find 50 percentile

 vector = zeros(1,300);

for i = 1:300

    med = trapz(0:0.1:(i/10), PDF_gamma(0:0.1:(i/10), alpha, beta));

    vector(i) = med;

end

median = find(vector<=0.51 & vector>=0.49);
median = median/10;
disp(['The median life span of asphalt is',' ',num2str(median),' ','years'])
```



Problem 2:

Tasks 1, 2, and 3:

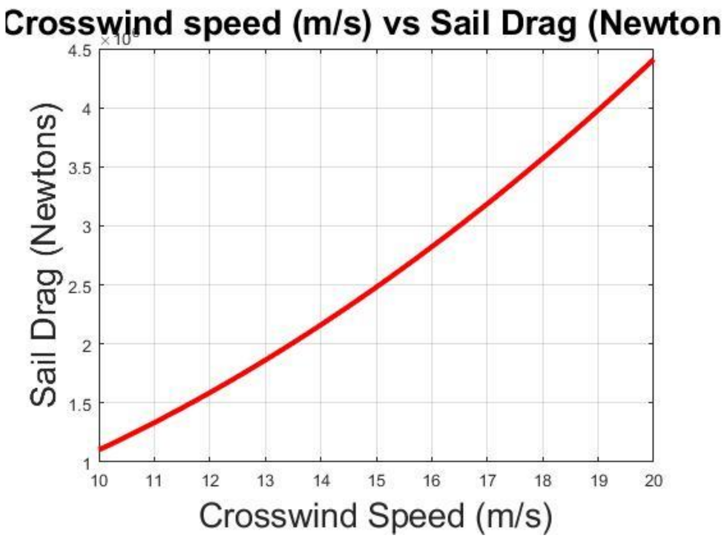
```
%%%%%%%%%%%%%Problem 2 Task 1 and 2%%%%%%%%%%%%%%
%script to calculate sail drag (Newtons) and the power to overcome sail drag (Watts)
%programmed by: Armin Zolfaghari

close all
clear
clc

%%Input variables:
%p = air density (kg/cu. meter)
%V = crosswind speed (m/s)
%S = sail area (square meters)
%C = drag coefficient (dimensionless)
p = 1.225;
V = 17;
S = (25+20)*400;
C = 1.0;

%call sail function
%outputs:
%sail_drag (newtons)
%power_overcome (watts)
[sail_drag,power_overcome] = sail(p,V,S,C);

disp(['Sail drag is ',num2str(sail_drag),' Newtons'])
disp(['Power necessary to overcome sail drag is ', num2str(power_overcome), ' Watts'])

%%%%%%%%%%%%%Problem 2 Task 3%%%%%%%%%%%%%%
%With 5 MW of power, the thrusters cannot provide the necessary power because the power to overcome sail drag is 54 MW.

%%%%%%%%%%%%%Problem 2 Task 4%%%%%%%%%%%%%%

%crosswind speed (m/s)
V = 10:0.05:20;
%call sail function
[sail_drag,power_overcome] = sail(p,V,S,C);
%plot crosswind speed vs sail drag
plot(V,sail_drag, '-r','LineWidth',3);
title('Crosswind speed (m/s) vs Sail Drag (Newtons)','FontSize', 20)
xlabel('Crosswind Speed (m/s)','FontSize', 20)
ylabel('Sail Drag (Newtons)','FontSize', 20)
grid
```


```
%%%%%%%%%%%%%Problem 2 Task 1 and 2%%%%%%%%%%%%%%
%script to calculate sail drag (Newtons) and the power to overcome sail drag (Watts)
%programmed by: Armin Zolfaghari

close all
clear
clc

%%Input variables:
%p = air density (kg/cu. meter)
%V = crosswind speed (m/s)
%S = sail area (square meters)
%C = drag coefficient (dimensionless)
p = 1.225;
V = 17;
S = (25+20)*400;
C = 1.0;

%call sail function
%outputs:
%sail_drag (newtons)
%power_overcome (watts)
[sail_drag,power_overcome] = sail(p,V,S,C);

disp(['Sail drag is ',num2str(sail_drag),' Newtons'])
disp(['Power necessary to overcome sail drag is ', num2str(power_overcome), ' Watts'])
```


Command Window

```
Sail drag is 3186225 Newtons
Power necessary to overcome sail drag is 54165825 Watts
fx >>
```

Task 4:




```
%Task 1 Create the function to calculate the sail drag
%with four inputs p,V,S,C, and two outputs Ds,Ps
function [sail_drag,power_overcome] = sail(p,V,S,C)

    sail_drag = 0.5.*p.*V.^(2).*S.*C;

    power_overcome = sail_drag.*V;


end
```

Problem 3:

Task 1:

```
%%%%%%%%%%%%%%Problem 3 Task 1%%%%%%%%%%%%%%%
%function to calcualte the basic resistant of a train and the necessary
%resistance


function [resistance,power] = train_resistance(A,B,C,v)

%inputs:
%A, B, and C are specific coefficients of the formula
%v is a vector for train speed (m/s)

%Outputs:
%resistance is basic resistance of a high-speed train
%power = necessary power in kwatts to move the train

    resistance = A + B.*v + C.*v.^(2);
    power = resistance.*v;

end
```

Task 2:

```
%%%%%%%%%%%%%Problem 3 Task 2%%%%%%%%%%%%%%

%%script to calcualte the basic resistant of a train and the necessary
%resistance

close all
clear
clc

%Determine inputs:

%A , B ,and C are train-specic coefficients
%v is the train speed (m/s)
A = 8.04000;
B = 0.12356;
C = 0.01099;
v = 20:0.1:85;

%Outputs:
%resistance is basic resistance of a high-speed train
%power = necessary power in kwatts to move the train

%call train_resistance function
[resistance,power] = train_resistance(A,B,C,v);

%plot the outputs vs speed
%plot resistance vs speed
figure
subplot(2,1,1)
plot(v,resistance, '-r','LineWidth',3)
xlabel('Train speed (m/s)','FontSize', 18)
ylabel('Produced Resistance (KNewtons)','FontSize', 18)
title('Speed (m/s) vs Produced Resistance (KNewtons)','FontSize', 16)
grid

%plot power vs speed
subplot(2,1,2)
plot(v,power, '-b','LineWidth',3)
xlabel('Train speed (m/s)','FontSize', 18)
ylabel('Power (KWatts)','FontSize', 18)
title('Speed (m/s) vs Power (KWatts)','FontSize', 16)
grid
```