Table of Contents

Roshan Jaiswal-Ferri

```
%Section - 01
%Aero 446 HW6: 5/27/25
```

Workspace Prep

```
%warning off
format long      %Allows for more accurate decimals
close all;       %Clears all
clear all;       %Clears Workspace
clc;             %Clears Command Window
```

Problem 1

```
d = 1;
r = d/2;
n = 0.6;
freq = 3e8; %300 mhz

Aeff = (pi*r^2)*n;
G = (4*pi*Aeff)/(freq^2);
GdB = 10*log10(G);

disp(['Gain at 300 MHz (dB): ', num2str(GdB)]);
```

```
Gain at 300 MHz (dB): -161.8179
```

Problem 2

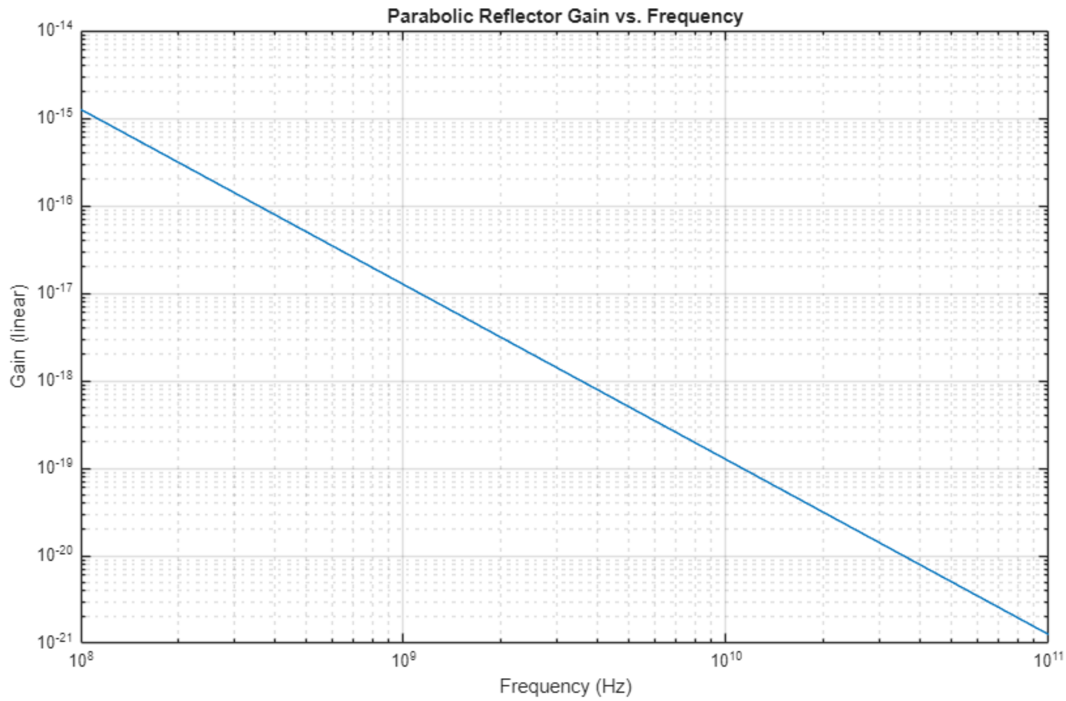
```
freq2 = 3e8*1.05;
freq3 = 3e8*0.95;

G2 = (4*pi*Aeff)/(freq2^2);
G3 = (4*pi*Aeff)/(freq3^2);

GdB2 = 10*log10(G2);
GdB3 = 10*log10(G3);

gd = GdB - GdB2; %not symmetrical
```

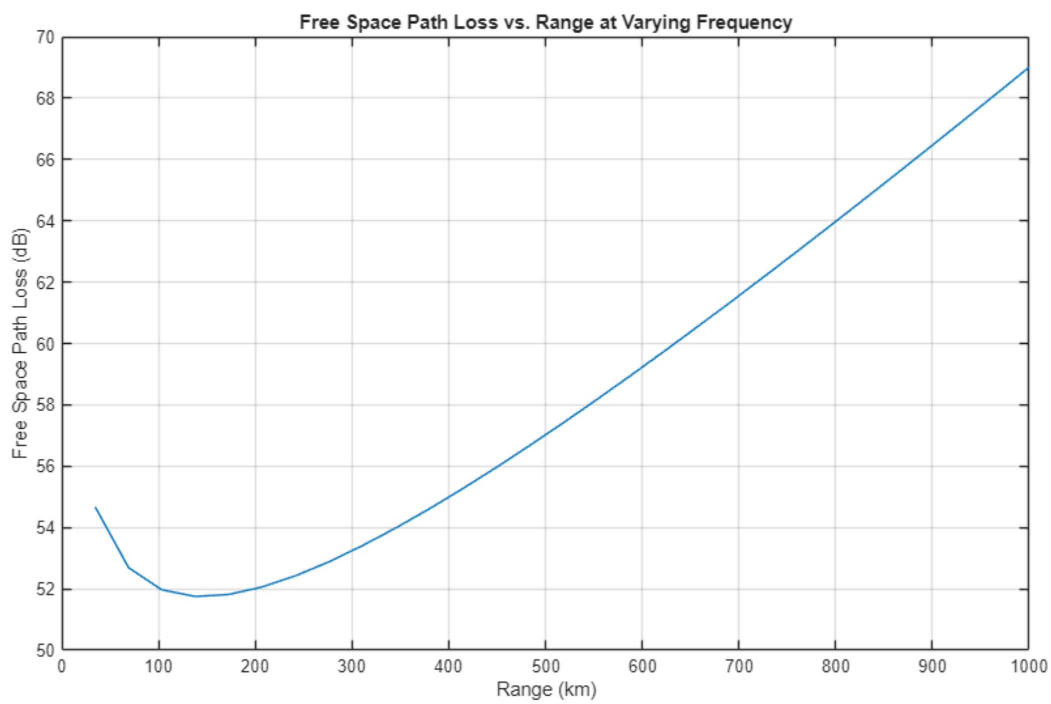
```
gd2 = GdB - GdB3;

% Since gain is proportional to 1/frequency^2, the change is not symmetric.
disp(['Gain drop (+5%): ', num2str(gd), ' dB']);
disp(['Gain drop (-5%): ', num2str(gd2), ' dB']);
```

```
Gain drop (+5%): 0.42379 dB
Gain drop (-5%): -0.44553 dB
```

Problem 3

```
f_start = 1e8;   % 100 MHz
f_end = 1e11;    % 100 GHz

num_decades = log10(f_end/f_start);
points_per_decade = 10;
num_points = num_decades * points_per_decade;

f_vector = logspace(log10(f_start), log10(f_end), num_points);

Aeff = 1;
G = (4*pi*Aeff)./(f_vector.^2);

figure('Name','Parabolic Reflector Gain vs. Frequency')
loglog(f_vector, G)
xlabel('Frequency (Hz)')
ylabel('Gain (linear)')
title('Parabolic Reflector Gain vs. Frequency')
grid on

% Gain decreases with increasing frequency squared, due to 1/f^2 dependency.

range = linspace(0,1000,length(f_vector));
inside = (f_vector./(4*pi*range));
Lfs = 10.*log10(inside);

figure('Name','Free Space Path Loss vs. Range at Varying Frequency')
plot(range, Lfs)
xlabel('Range (km)')
ylabel('Free Space Path Loss (dB)')
title('Free Space Path Loss vs. Range at Varying Frequency')
grid on

% Free space loss increases with both range and frequency (L is porportional
to f^2).
```





Problem 4

```
f = 300e6;% 300 MHz
c = 3e8;
```

```
lambda = c/f; % Wavelength in meters
beamwidth_deg = 18; % 3 dB Beamwidth in degrees
Pt_watts = 100; % watts
n = 1;

D = 70 * lambda / beamwidth_deg; % in meters
disp(['Antenna diameter: ', num2str(D), ' meters']);

% Gain
A = pi*(D/2)^2;
Aeff = n*A;
G = (4*pi*Aeff)/(lambda^2);
GdBi = 10*log10(G);

% EIRP
Pt_dBW = 10*log10(Pt_watts); % Transmit power in dBW
EIRP_dBW = Pt_dBW + GdBi;

disp(['EIRP: ', num2str(EIRP_dBW), ' dBW']);
```

```
Antenna diameter: 3.8889 meters
EIRP: 41.7395 dBW
```

Published with MATLAB® R2024b