Table of Contents

ECE 8873 Homework 2.1

Caitlyn Caggia

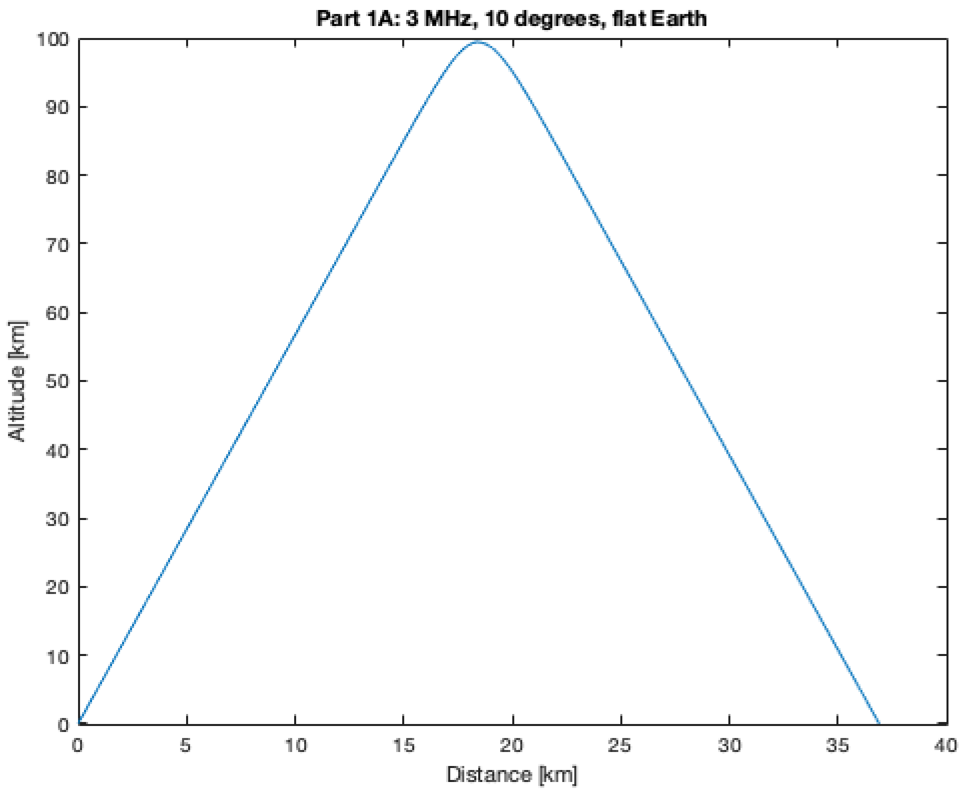
```
clear all; close all;
```

Part A

```
thetai_a = 10; % incident angle from vertical [degrees]
w_a = 2 * pi * 3e6; % propagation frequency 3 MHz [rad/sec]

[z_a, h_a, ~] = raytracer('a', thetai_a, w_a);
figure;
plot(z_a, h_a);
title('Part 1A: 3 MHz, 10 degrees, flat Earth')
xlabel('Distance [km]')
ylabel('Altitude [km]')

sprintf('1A) Highest altitude: %2.2f km.', max(h_a))
```



Part B

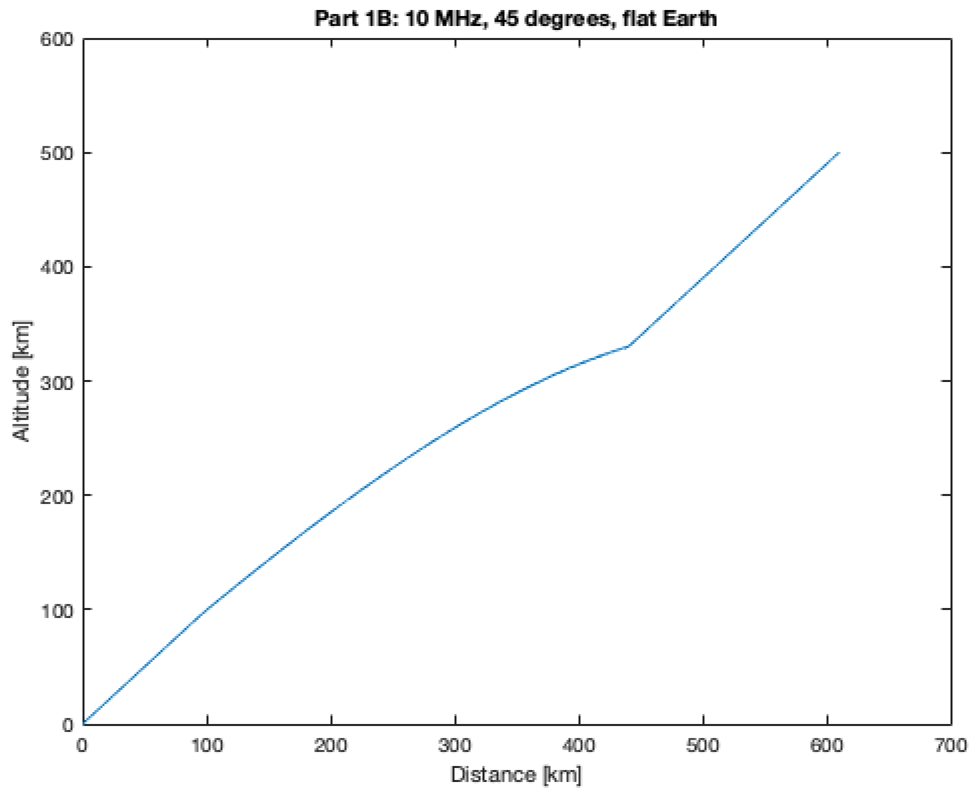
```
thetai_b = 45;
w_b = 2 * pi * 10e6;

[z_b, h_b, ~] = raytracer('b', thetai_b, w_b);
figure;
plot(z_b, h_b);
title('Part 1B: 10 MHz, 45 degrees, flat Earth')
xlabel('Distance [km]')
ylabel('Altitude [km]')

if max(h_b) > 500
    sprintf('1B) Yes, it clears the ionosphere.')
else
    sprintf('1B) No, it does not clear the ionosphere.')
end
```

```
ans =

    '1B) Yes, it clears the ionosphere.'
```



Part C

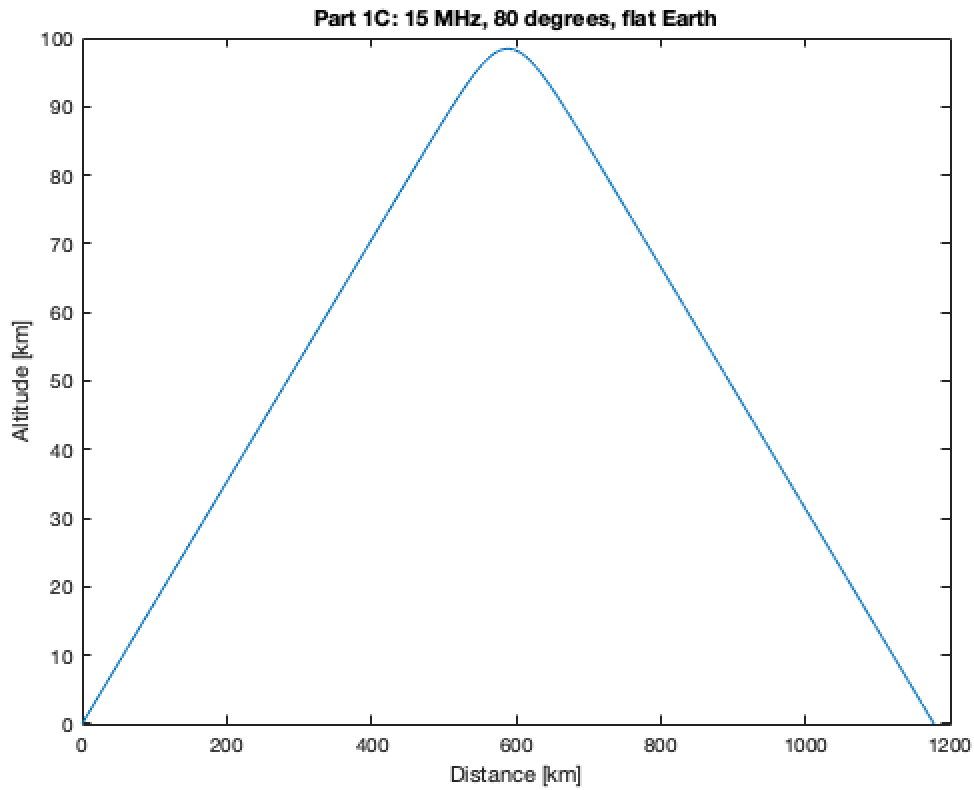
```
thetai_c = 80;
w_c = 2 * pi * 15e6;

[z_c, h_c, ~] = raytracer('c', thetai_c, w_c);
figure;
plot(z_c, h_c);
title('Part 1C: 15 MHz, 80 degrees, flat Earth')
xlabel('Distance [km]')
ylabel('Altitude [km]')

sprintf('1C) First bounce reaches %2.2f km from the source.',
 z_c(end))
```

```
ans =

    '1C) First bounce reaches 1177.96 km from the source.'
```



Part D

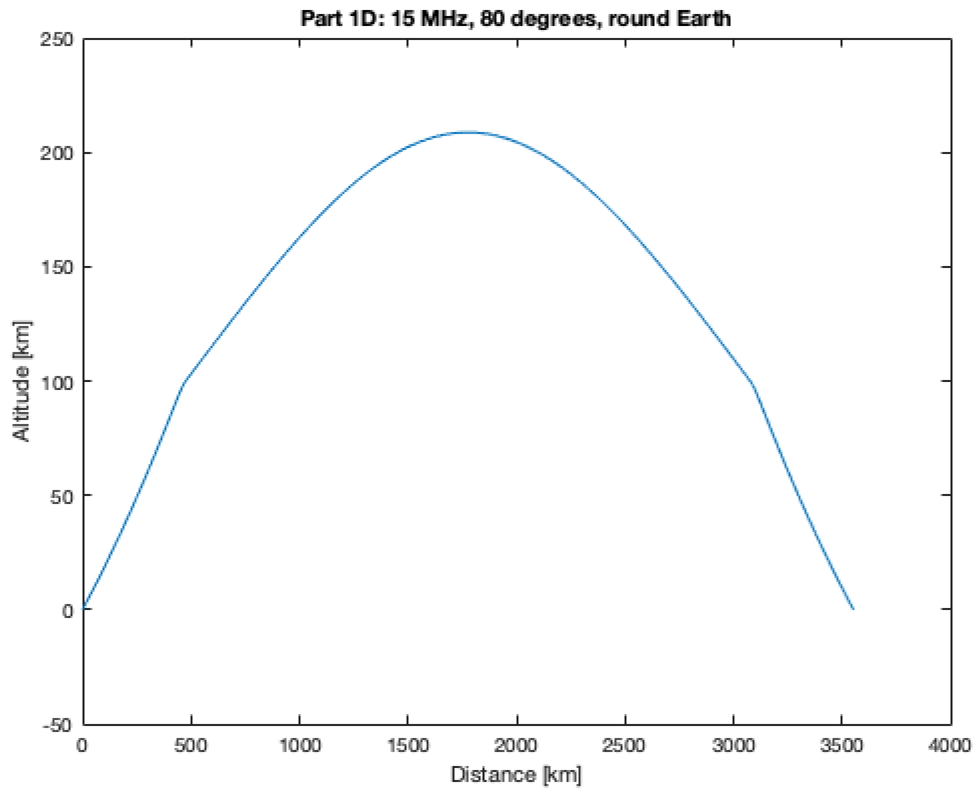
```
thetai_d = 80;
w_d = 2 * pi * 15e6;

[z_d, h_d, ~] = raytracer('d', thetai_d, w_d);
figure;
plot(z_d, h_d);
title('Part 1D: 15 MHz, 80 degrees, round Earth')
xlabel('Distance [km]')
ylabel('Altitude [km]')

sprintf('1D) Signal lands %2.2f km away.', z_d(end))
```

```
ans =

    '1D) Signal lands 3552.92 km away.'
```



Part E

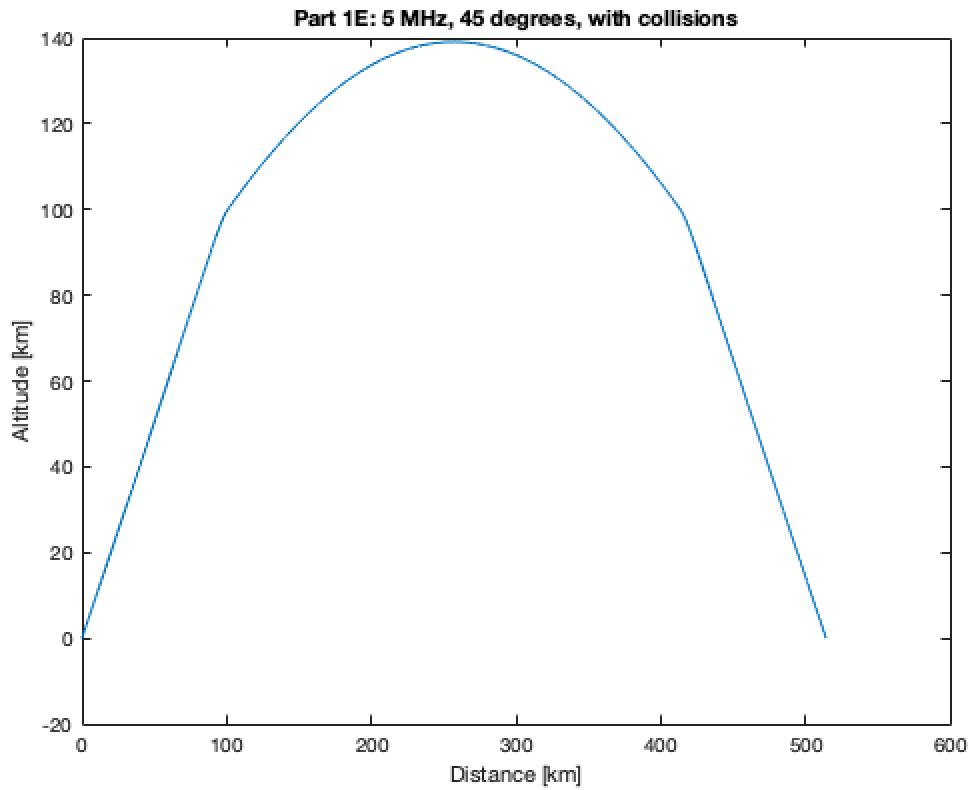
```
thetai_e = 45;
w_e = 2 * pi * 5e6;

[z_e, h_e, atten_e] = raytracer('e', thetai_e, w_e);
figure;
plot(z_e, h_e);
title('Part 1E: 5 MHz, 45 degrees, with collisions')
xlabel('Distance [km]')
ylabel('Altitude [km]')

atten_dB = 10 * log(exp(-atten_e));
sprintf('1E) Total attenuation: %2.2f dB', atten_dB)
```

```
ans =

    '1E) Total attenuation: -21.90 dB'
```



Part F

```
[z_f, h_f, ~] = raytracer('f', thetai_e, w_e);
figure;
plot(z_e, h_e);
hold on;
plot(z_f, h_f, '-*');
title('Part 1F: RHCP vs LHCP')
xlabel('Distance [km]')
ylabel('Altitude [km]')
legend('LHCP','RHCP')

sprintf('1F) LHCP goes both higher and farther than RHCP.')
```

Raytracing Helper Function

```
function [distance, height, atten] = raytracer(part, thetai, w)

% constants
delta = 0.01; % step size
muR = 1;
mu = muR * 4*pi*10^(-7);
ep0 = 8.85e-12;
radE = 6371; % radius of Earth [km]
q = 1.6e-19; % charge of an electron [C]
```

```
melec = 9.11e-31; % mass of an electron [kg]
B = 40e-6; % Earth's magnetic field [T]
theta = 90; % assuming East-West direction [degrees]

% initialize height
h = delta * tand(90 - thetai); % [km]
height = [h]; % store all h values to plot later
r0 = radE + h;
ri = r0;

% initialize electron density
Ne = 1; % [m^-3]

% initialize index of refraction
n1 = 1.0000;
sign = 1; % handle Snell's Law negative sign

% initially assume no collisions
v = 0;

% run raytracing loop
keepGoing = true;
atten = 0;

while keepGoing

    % update electron density
    if h <= 100
        Ne = 10^(0.111 * h);
    elseif 330 >= h
        Ne = 10^(0.0028*(h-100)+11.1);
    else
        Ne = (-0.0028*(h-330)+11.744);
    end

    % update collisions
    if part == 'e'
        if h < 125
            v = 10^((170-h)/15);
        else
            v = (10^((170-125)/15))*10^((125-h)/87.5);
        end
    end

    % update Appleton-Hartree Equation
    wc = q*B / melec; % gyro frequency
    wp = sqrt((Ne * q^2) / (ep0 * melec)); % plasma frequency
    X = (wp/w)^2; Y = wc/w; Z = v/w;
    a = ((Y*sind(theta))^2) / (2 - (2*X) - (2*j*Z));
    b = (Y*cosd(theta))^2;
    if part == 'f' % RHCP
        epR = 1 - X/(1 - (j*Z) - a - sqrt(b + a^2));
    else % LHCP
        epR = 1 - X/(1 - (j*Z) - a + sqrt(b + a^2));
```

```
    end

    % update index of refraction
    n2 = sqrt(muR * real(epR));

    % update angles
    if part == 'a' || part ==  'b' || part == 'c'
        thetat = asind(n1*sind(thetai)/n2); % tx angle
    else
        thetat = asind(r0*n1*sind(thetai)/(n2*ri)); % tx angle
    end
    thetac = asind(n2/n1); % critical angle
    if thetat > thetac
        sign = -sign;  % adjust for negative sign if needed
    end

    % update height
    h = sign * delta * tand(90-thetai) + h;
    height = [height h];
    r0 = ri;
    ri = radE + h;

    % update attenuation
    k = w*sqrt(mu*epR*ep0);
    alpha = -imag(k);
    atten = atten + (alpha*delta*1000); % need atten in nepers (m, not
 km)

    % update Snell's Law parameters
    thetai = thetat;
    n1 = n2;

    % check stopping condition
    if part == 'b'
        if h > 500 || h < 0
            keepGoing = false; % passed the ionosphere or hit ground
        end
    else
        if h < 0, keepGoing = false; end % finished first bounce
    end

end

distance = delta*(1:length(height)); % [km]

end
```

ans =

'1A) Highest altitude: 99.42 km.'

Published with MATLAB® R2018a