ElecEng 2FL3
Assignment 3
Travelling Pulses and Waves

Prabhanjan Mannari   Natalia K. Nikolova

The aim of assignment 3 is to generate animated plots of travelling pulses and harmonic (sinusoidal) waves using MATLAB®. Sample code is given in **Section 4** showing how to generate an animation of a Gaussian pulse travelling in the $+z$ direction.

1 Instructions

By completing this exercise you will gain experience using The Mathworks® MATLAB® software. There is no need for an analytical solution in this assignment.

A MATLAB® code must be written to generate animated plots of a Gaussian pulse and a harmonic wave (details in the next section). You must submit your MATLAB® (***.m) file through the A2L course website.

2 Problem Statement

A wave travelling along $+\hat{\mathbf{z}}$ is a function of position z and time t, which is expressed as

$$y(z,t) = y\Big(t - \frac{z}{v_p}\Big),$$

where $v_p = \frac{c}{\sqrt{\epsilon_r}}$ is the phase (or wave) velocity in the medium of relative permittivity ϵ_r. A wave travelling in the $-\hat{\mathbf{z}}$ direction can be expressed by the same formula but with a negative phase velocity $-v_p$. Any twice differentiable function $y(x)$ is a solution to the 1D wave equation provided the argument x (also referred to as *wave phase*) has the form $x \sim t \pm \frac{z}{v_p}$ or $x \sim z \pm v_p t$.

Consult the next two subsections for the mathematical descriptions of the Gaussian-pulse and the harmonic waves.

2.1 Gaussian Pulse

The Gaussian pulse is described as

$$y(t) = Ae^{-\alpha(t-t_0)^2} ,$$

where A is the amplitude, t_0 is a parameter which determines the temporal position of the pulse peak, and α is a parameter, which determines the width of the pulse. For this function to define a 1D wave, one has to replace the argument t with the argument $t \pm \frac{z}{v_p}$.

2.2 Harmonic Wave

The harmonic (or sinusoidal) wave is described as

$$y(t) = A \sin(\omega t + \phi) ,$$

where $\omega = 2\pi f$ is the angular frequency and ϕ is the initial phase. You can use a phase offset $\phi = 0$ in your code. For this function to define a 1D wave, one has to replace the argument t with the argument $t \pm \frac{z}{v_p}$. Then, the harmonic wave is expressed by

$$y(t) = A \sin(\omega t \pm \beta z + \phi) ,$$

where $\beta = \frac{\omega}{v_p}$ is the *phase constant*. The phase constant relates to the wavelength λ as $\beta = \frac{2\pi}{\lambda}$.

2.3 Problem

From the variations in the **Section 3**, calculate the required parameters for the Gaussian-pulse and harmonic waves. Use these parameters to generate animated plots similar to the ones generated by the sample code.

Your code must generate the following animation plots:

1. Gaussian pulse travelling in the $+z$ direction;
2. Gaussian pulse travelling in the $-z$ direction;
3. Superposition of the above two pulses;
4. Sinusoidal wave travelling in the $+z$ direction;
5. Sinusoidal wave travelling in the $-z$ direction;
6. Superposition of the above two waves.

Choose appropriately the spatial and temporal window sizes for your animations. You can consult the guidelines below.

1. Gaussian Pulse
   - Z window $= \left[-\frac{3v_p}{\sqrt{2\alpha}}, +\frac{3v_p}{\sqrt{2\alpha}}\right]$
   - Time window $= [t_0 - \frac{6}{\sqrt{2\alpha}}, t_0 + \frac{6}{\sqrt{2\alpha}}]$
2. Sinusoidal Wave
   - Z window $= [-3\lambda, +3\lambda]$
   - Time window $= [-\frac{3}{f}, +\frac{3}{f}]$

You can submit 2 separate MATLAB® codes, each generating one animation, or, you can have both animations generated by a single MATLAB® code. In the latter case, it is easier to generate the animations one after another using consecutive *for* loops rather than both animations at the same time.

3 Variations

Find out your assigned variation number (#) from the last (rightmost) digit of your student number, which must match the number of the variation found in the table below.

#	Frequency f for harmonic wave (Hz)	α for Gaussian pulse (s^{-2})	Relative permittivity ϵ_r
0	10^3	10^5	1
1	10^4	$2 \cdot 10^5$	2
2	10^5	$5 \cdot 10^5$	2.28
3	$5 \cdot 10^5$	10^6	3.5
4	10^6	10^7	4.9
5	10^7	10^8	1
6	$5 \cdot 10^7$	$2 \cdot 10^8$	2
7	10^8	$5 \cdot 10^8$	2.28
8	$1.5 \cdot 10^8$	10^9	3.5
9	$2 \cdot 10^8$	10^{10}	4.9

Table 1: Assigned variations of the frequencies, α and relative permittivities

4 Sample Code 1

The following is a sample code that generates a Gaussian-pulse wave travelling in the $+z$ direction.

```
%speed of light c in free space
c  = 299 792 458; %m/s
%relative permittivity of the medium epsilon_r
epsilon_r = 4.9;
%phase velocity vp
vp = c/(sqrt(epsilon_r));

%Gaussian Pulse parameters
alpha          = 10e9;
A_pulse        = 5;
t_0            = 1;
```

```
z_step_pulse   = (vp*1e-1)/sqrt(2*alpha);
z_start_pulse  = -(3*vp)/sqrt(2*alpha);
z_end_pulse    =  (3*vp)/sqrt(2*alpha);
z_window_pulse = z_start_pulse:z_step_pulse:z_end_pulse;
max_amplitude  = A_pulse;
t_step_pulse   = 1e-1*(1/sqrt(2*alpha));
t_window_pulse = t_0- 6/sqrt(2*alpha):t_step_pulse:t_0+6/sqrt(2*alpha);

for T = t_window_pulse
    figure(1);

    amplitude1 = gaussian_pulse(A_pulse,alpha,t_0,z_window_pulse,T,vp);

    %plot using axes set to fixed ranges
    subplot(2,1,1)
    plot(z_window_pulse,amplitude1)
    axis([z_start_pulse,z_end_pulse,0,max_amplitude])
    title("Gaussian pulse travelling in +z direction");
    xlabel("z(m)");
    ylabel("Amplitude")

    %plot without setting the axes ranges
    subplot(2,1,2)
    plot(z_window_pulse,amplitude1)
    title("Gaussian pulse travelling in +z direction");
    xlabel("z(m)");
    ylabel("Amplitude")
    pause(1e-3);
end

function amplitude = gaussian_pulse(A,alpha,t_0,z,t,vp)
argument           = -alpha*(t-(z/vp)-t_0).^2;
amplitude          = A*exp(argument);
end
```

Use `subplot` to plot multiple objects in a single figure. Plot the first three parts in one single figure, and the next three parts in another figure. Note that the `plot` function uses axes ranges from the minimum value in the argument to the maximum value in the argument. The minimum and maximum values are not the same at all times for a gaussian pulse in the `z_window`. It is much clearer to analyze the plot with the same axes ranges over time. Observe the difference in the two subplots generated by the sample code. Use `axis([x_lo x_hi y_lo y_hi])` to keep the axes size fixed over time.

5 Sample Code 2

The above code implements the required *function* named "gaussian_pulse" as an internal script. This may pose compatibility problems with older MATLAB® versions. Below, a code is provided wherein an alternative way of defining a function is used, namely, the "anonymous" function. It is our observation that this function definition is back compatible with all MATLAB® versions.

```
%speed of light c in free space
c  = 3e8;
%relative permittivity of the medium epsilon_r
epsilon_r = 4.9;
%phase velocity vp
vp = c/(sqrt(epsilon_r));

% function amplitude = gaussian_pulse(A,alpha,t_0,z,t,vp)
% argument           = -alpha*(t-(z/vp)-t_0).^2;
% amplitude          = A*exp(argument);
% end
gaussian_pulse = @(A,alpha,t_0,z,t,vp) A*exp(-alpha*(t-(z/vp)-t_0).^2);

%Gaussian Pulse parameters
alpha          = 10e9;
A_pulse        = 5;
t_0            = 1;
z_step_pulse   = (vp*1e-1)/sqrt(2*alpha);
z_start_pulse  = -(3*vp)/sqrt(2*alpha);
z_end_pulse    =  (3*vp)/sqrt(2*alpha);
z_window_pulse = z_start_pulse:z_step_pulse:z_end_pulse;
```

```
max_amplitude  = A_pulse;
t_step_pulse   = 1e-1*(1/sqrt(2*alpha));
t_window_pulse = t_0- 6/sqrt(2*alpha):t_step_pulse:t_0+6/sqrt(2*alpha);

for T = t_window_pulse
    figure(1);

    amplitude1 = gaussian_pulse(A_pulse,alpha,t_0,z_window_pulse,T,vp);

    %plot using axes set to fixed ranges
    subplot(2,1,1)
    plot(z_window_pulse,amplitude1)
    axis([z_start_pulse,z_end_pulse,0,max_amplitude])
    title('Gaussian pulse travelling in +z direction');
    xlabel('z(m)');
    ylabel('Amplitude')

    %plot without setting the axes ranges
    subplot(2,1,2)
    plot(z_window_pulse,amplitude1)
    title('Gaussian pulse travelling in +z direction');
    xlabel('z(m)');
    ylabel('Amplitude')
    pause(1e-3);
end
```