ElecEng 2CF3
ASSIGNMENT 5
Traveling Pulses and Waves

Mihail Georgiev, Prabhanjan Mannari, and Natalia K. Nikolova

Instructions and Objective

- Complete all three exercises.
- If you do not have MATLAB, follow UTS's instructions: https://uts.mcmaster.ca/services/computers-printers-and-software/software-licensing/matlab/. You do not need any toolboxes for this assignment.
- Submit one MATLAB source-code (`*.m`) file per exercise. Name each one `exerciseX.m` where *X* is the number of the exercise. If the current version of MATLAB cannot run the submitted file without errors, you will receive a grade of zero for the corresponding exercise. In particular, do not change the file name without checking that MATLAB can still open it, and ensure the file name extension of what you submit is not `*.txt`.
- The first line of each source-code file must be a comment with the exercise number, your full name as it appears on Avenue to Learn, your student ID number and your MacID login name.
- Submit one auto-generated PDF report per exercise with the same file name as your source-code file but with the extension `*.pdf`. The MATLAB command `publish('exerciseX.m', 'pdf')` will both create the report and name it appropriately. You do not have to write any reports manually.
- Submit exactly three `*.m` files and exactly three `*.pdf` files to the respective Dropbox on Avenue to Learn before the deadline. Do not submit files of any other sort, including archives (e.g., ZIP files).

In this assignment, you will create animated plots of one-dimensional traveling waves using MATLAB.

Theory

Consider a point which moves along the z-axis with velocity $v_p \neq 0$ starting from position $z = 0$ at time $t = 0$. Its motion is described by

$$z = v_p t \iff t - \frac{z}{v_p} = 0\,. \tag{1}$$

An entire function can move similarly. Consider an arbitrary function $f(\tau)$ and call the point $\tau = 0$ its origin. For

$$f\left(\tau = t - \frac{z}{v_p}\right), \tag{2}$$

the origin travels in the $+z$ direction as t increases if $v_p > 0$ and in the $-z$ direction if $v_p < 0$. For example, if $v_p = 2$ then $z = 10$ when $t = 5$ so that $\tau = 0$. We call $f(t - z/v_p)$ a *traveling wave* with *phase velocity* v_p, where z is *position* and t is *time*.[1] For electromagnetic waves, the phase velocity v_p relates to the *speed*

[1] Strictly speaking, any solution f of the *1-D wave equation*

$$\frac{1}{v_p^2}\frac{\partial^2 f}{\partial t^2} - \frac{\partial^2 f}{\partial z^2} = 0 \tag{3}$$

is a 1-D wave. This is a better definition than what is given above but beyond the scope of the assignment.

of light c via the *refractive index*, which is the square root of the *relative permittivity* ε_r of the medium:

$$|v_p| = \frac{c}{\sqrt{\varepsilon_r}}. \tag{4}$$

Here, we consider two specific examples of waves. The first is a traveling *Gaussian pulse* for which

$$f(\tau) = g(\tau) := Ae^{-\alpha\tau^2}, \tag{5}$$

where A is the *amplitude* and α is a *pulse-width parameter*. The second is a *sinusoidal (harmonic) wave* for which

$$f(\tau) = h(\tau) := A\cos(\omega\tau + \phi), \tag{6}$$

where $\omega \equiv 2\pi f$ is the *angular frequency* and ϕ is the *phase*. When written in terms of z and t, i.e.,

$$h\left(t - \frac{z}{v_p}\right) = A\cos\left(\omega t - \frac{\omega}{v_p}z + \phi\right) = A\cos(\omega t - \beta z + \phi), \tag{7}$$

a coefficient β termed the *phase constant* appears. Using the *wavelength* $\lambda \equiv v_p/f$ and *period* $T \equiv 1/f$, we can write it in many equivalent ways such as

$$\beta = \frac{\omega}{v_p} = \frac{2\pi f}{v_p} = \frac{2\pi}{Tv_p} = \frac{2\pi}{\lambda}. \tag{8}$$

The phase constant β is assumed positive, which implies that (7) describes a wave traveling along $+z$. The wave traveling in the opposite $-z$ direction is then described by

$$A\cos(\omega t + \beta z + \phi). \tag{9}$$

Finally, waves can superpose. The *superposition* of two waves f_1 and f_2 with velocities $v_{p,1}$ and $v_{p,2}$, respectively, is their sum:

$$f_1\left(t - \frac{z}{v_{p,1}}\right) + f_2\left(t - \frac{z}{v_{p,2}}\right). \tag{10}$$

An interesting effect is observed for otherwise-identical sinusoids that travel in opposite directions. This is due to the sum-to-product cosine identity, i.e.,

$$\cos(\mu - \nu) + \cos(\mu + \nu) \equiv 2\cos(\mu)\cos(\nu), \tag{11}$$

from which it follows that, in the zero-phase case,

$$h\left(t - \frac{z}{v_p}\right) + h\left(t + \frac{z}{v_p}\right) = A\cos(\omega t - \beta z) + A\cos(\omega t + \beta z) = A\cos(\beta z)\cos(\omega t). \tag{12}$$

This shows that the traveling behavior is in some sense lost for the total wave because there is no term of the form $(t - z/v_p)$ in the superposition. The wave instead oscillates in place due to the $\cos(\omega t)$ term and the maximum of its swing (the *envelope*) depends on the position z due to the $A\cos(\beta z)$ term. Such a wave is called a *standing wave*. If the waves have nonzero phases $\phi_{1,2} = \bar{\phi} \pm \Delta\phi/2$, the superposition is

$$A\cos\left(\omega t - \beta z + \bar{\phi} + \Delta\phi/2\right) + A\cos\left(\omega t + \beta z + \bar{\phi} - \Delta\phi/2\right) = A\cos(\beta z - \Delta\phi/2)\cos\left(\omega t + \bar{\phi}\right), \tag{13}$$

thus the standing wave is unchanged from the zero-phase case except for shifts in time and space.

1 Plot Gaussian Waves

You are given sample code for Algorithm 1 (see `exerciseX.m` in the ASSIGNMENTS folder under the Content tab on A2L) which creates a figure with a 2×1 grid of subplots. In the first subplot, an animated red line shows a Gaussian pulse

$$g(\tau) := 5e^{-\alpha\tau^2} \tag{14}$$

traveling in the $+z$ direction at speed v_p, i.e., $g(t - z/v_p)$ assuming $v_p > 0$. In the second subplot, an animated blue line also shows $g(t - z/v_p)$. Familiarize yourself with this code and make sure it runs as given before you begin to modify it. For any function you are not familiar with, its documentation can be accessed

- by searching for it via the Search Documentation field in the upper-right of the MATLAB IDE interface,
- by entering `doc func` into the Command Window, where `func` is the name of the function,[2]
- via the context (right-click) menu in the Command Window or Editor, or
- by pressing the F1 key while the text cursor is over its name in the Command Window or Editor.

For the animation, the spatial (z-axis) and temporal (t-axis) intervals are defined as

$$\begin{aligned} -\frac{3v_p}{\sqrt{2\alpha}} \leq \quad z \quad &\leq +\frac{3v_p}{\sqrt{2\alpha}} \\ -\frac{6}{\sqrt{2\alpha}} \leq \quad t \quad &\leq +\frac{6}{\sqrt{2\alpha}}. \end{aligned}$$

They specify the horizontal axis of the plot and the start and end times of the animation, respectively. Make sure you understand how these intervals are represented in the sample code.

Modify the sample code from Algorithm 1 such that:

1. The relative permittivity matches Table 1.
2. The pulse-width parameter matches Table 1.
3. There is a 3×1 grid of plots.
4. In the first subplot, an animated red line shows the Gaussian pulse traveling in the $+z$ direction with an appropriate title (no change).
5. In the second subplot, an animated blue line shows the Gaussian pulse traveling in the $-z$ direction with an appropriate title.
6. In the third subplot, an animated magenta line shows the superposition of the preceding two waves with an appropriate title.
7. The waves exactly fit into the plot window in all three cases. In particular, set the limits of the vertical axis for the superposed wave appropriately.
8. All comments properly describe the code. In particular, if you copy-paste something, change the comments, too.
9. No errors[a] of any sort are printed in the Command Window. As stated in the instructions at the start of the assignment, you will receive a grade of zero for the exercise if any errors are printed regardless of how well the code otherwise runs.

Once you are satisfied with your code, run the following command in the Command Window:

```
>> publish('exercise1.m', 'pdf')

ans =

    '/PATH/TO/html/exercise1.pdf'
```

The returned string is the path on disk to a report containing your source code and figure.
Submit to A2L your source-code file (`exercise1.m`) and this report (`exercise1.pdf`) .

[a]Warnings (printed in orange by default) are not errors (printed in red by default). You will not be penalized for warnings, but your script should ideally not emit them either.

[2]A short version of the documentation can be accessed by entering `help func`; this is often sufficient.

2 Plot Sinusoidal Waves

This time, start with a sinusoid

$$h(\tau) := 5\cos(\omega\tau) \tag{15}$$

with an amplitude of 5 and phase of 0, and plot the waves $h(t \mp z/v_p)$ traveling along the positive and negative z axes. The spatial (z-axis) and temporal (t-axis) intervals are defined as

$$\begin{aligned} -3\lambda \leq \quad & z \quad \leq +3\lambda \\ -3T \leq \quad & t \quad \leq +3T. \end{aligned}$$

They specify the horizontal axis of the plot and the start and end times of the animation, respectively. You can infer ω, λ and T from the frequency f given in Table 1. The cosine function is available in MATLAB under the name `cos`.

Modify the code from Exercise 1 such that:

1. Anything strictly related to the Gaussian pulse is removed.
2. The sinusoid's frequency matches Table 1.
3. The spatial z and temporal t limits are as given above and match the frequency from Table 1.
4. The animation has 3001 steps instead of 2001 steps.[a]
5. In the first subplot, an animated red line shows the sinusoid traveling in the $+z$ direction with an appropriate title.
6. In the second subplot, an animated blue line shows the sinusoid traveling in the $-z$ direction with an appropriate title.
7. In the third subplot, an animated magenta line shows the superposition of the preceding two waves with an appropriate title.
8. The waves exactly fit into the plot window in all three cases.
9. All comments properly describe the code.
10. No errors of any sort are printed in the Command Window.

Once you are satisfied with your code, publish an auto-generated report.
Submit to A2L both your source-code file (`exercise2.m`) and the report (`exercise2.pdf`).

[a] Note from the "animation instructions" portion of the code that the animation runs for as many steps as there are elements of the time vector `t`.

3 Formation of a Standing Wave

In the previous exercise, we saw that the superposition of the two sinusoids forms a standing wave. In this exercise, you will observe the initial formation of this standing wave. To do this, we need to see the two waves "meet" and will thus need a function that is sinusoidal for part of the domain and zero for the rest. An easy way to achieve this is with the *Heaviside unit-step function*

$$u(x) := \begin{cases} 1, & x \geq 0 \\ 0, & x < 0. \end{cases} \tag{16}$$

This *truncated sinusoid* is then

$$w(\tau) := 5\cos(\omega\tau)\, u(\omega\tau). \tag{17}$$

A sharp transition like in $u(x)$ is not realistic and leads to a poor visualization result in the animation. Thus, we replace the unit-step function with one that transitions smoothly from zero to its maximum. Here, we will use a function known as a *sigmoid*[3]:

$$s(x) := \frac{1 + \operatorname{erf}(x)}{2}, \quad \operatorname{erf}(x) := \frac{1}{\sqrt{\pi}} \int_0^x e^{-\xi^2} \mathrm{d}\xi. \tag{18}$$

The *error function* $\operatorname{erf}(x)$ is available in MATLAB under the name `erf`. The *smoothly truncated sinusoid* is thus

$$w(\tau) := 5 \cos(\omega\tau)\, s(\omega\tau). \tag{19}$$

Modify the code from Exercise 2 such that:

1. It plots smoothly truncated sinusoids $w(t \mp z/v_p)$ and their superposition instead of full sinusoids.
2. The animation runs for 50% longer. Use the temporal interval $-3T \leq t \leq +6T$.
3. The animation runs at approximately the same speed. Increase the number of steps by 50% also (from 3001 to 4501).
4. All comments properly describe the code.
5. No errors of any sort are printed in the Command Window.

Once you are satisfied with your code, publish an auto-generated report.
Submit to A2L both your source-code file (`exercise3.m`) and the report (`exercise3.pdf`).

Table 1: Assigned variations for the parameters, based on the last digit of your student number.

Last digit of student #	Frequency f for sinusoidal wave (Hz)[4]	Pulse-width 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

[3] Any roughly "S"-shaped function that somehow resembles a smooth step is a *sigmoid*. It is not a strict definition.

[4] If you want to use E notation, note that `MeP` means $M \cdot 10^P$, and thus `10eX` means $10 \cdot 10^X = 10^{X+1} \neq 10^X$; make sure to enter the parameters into your code correctly.

Algorithm 1 Sample code for Exercise 1. This code is also provided in a separate `*.m` file. Attempting to copy it from here without introducing errors is difficult and thus not advised.

```
% Exercise X - John Doe - 123456789 - doej
clear all; close all %#ok<CLALL> reset everything

% phase velocity
c = 299792458;          % speed of light
eps_r = 1.0;            % relative permittivity
vp = c / sqrt(eps_r); % phase velocity

% Gaussian pulse parameters
alpha = 1.0; A = 5;

% spatial and temporal axes
dz = (3 * vp) / sqrt(2 * alpha); z = linspace(-dz, +dz, 1001);
dt =        6 / sqrt(2 * alpha); t = linspace(-dt, +dt, 2001);

% function for a Gaussian pulse centered at the origin
gauss = @(tau) A * exp(-alpha * tau.^2);
% function for the corresponding wave over all points z at single time ti
wave = @(z, ti) gauss(ti - z / vp);

% plot specification
subplot(2, 1, 1)                                    % 2x1 grid, 1st plot
line1 = animatedline('Color', 'red');               % line in the plot
title("Gaussian pulse traveling in +z direction") % title
xlabel("z [m]"); ylabel("amplitude")                % axis labels
xlim(z([1 end])); ylim([0 A])                       % axis limits

subplot(2, 1, 2)                                    % 2x1 grid, 2nd plot
line2 = animatedline('Color', 'blue');              % line in the plot
title("Gaussian pulse traveling in +z direction") % title
xlabel("z [m]"); ylabel("amplitude")                % axis labels
xlim(z([1 end])); ylim([0 A])                       % axis limits

% animation instructions
for ti = t
    clearpoints(line1)
    clearpoints(line2)
    addpoints(line1, z, wave(+z, ti))
    addpoints(line2, z, wave(+z, ti))
    drawnow limitrate
end
```