PHYS134
Homework/Project 3
2021-02-22

Steck:

Steck 4.2 (5 pts) For a plane-wave solution of the wave equation,

$$\mathbf{E}(r,t) = \mathbf{E}_0 \cos(\mathbf{k}\cdot\mathbf{r} - \omega t + \phi_0)$$

we can write the constant $\mathbf{E}_0$ as $\hat{e}E_0$, where the unit vector $\hat{e}$ is the polarization and E_0 is a scalar constant. Show that the wave cannot be polarized along the $\mathbf{k}$ direction. Prove this directly; do not make any assumption about the form of the magnetic field.

■

Steck 4.4 (5 pts) The energy density (energy stored per unit volume) of an electromagnetic field in a simple, linear dielectric is (w, not ω):

$$w = \frac{1}{2}\left(\epsilon \mathbf{E}\cdot\mathbf{E} + \mu_0 \mathbf{H}\cdot\mathbf{H}\right)$$

for real field $\mathbf{E}$ and $\mathbf{H}$. (Remember that $\mathbf{B} = \mu\mathbf{H}$ and for non-magnetic materials, $\mu = \mu_0$, though this is not really relevant here.) Consider a plane wave in complex notation of the form

$$\mathbf{E}^{(+)}(\mathbf{r},t) = \hat{z}E_0^{(+)}e^{i(kx-\omega t)},$$

with $E_0^{(+)}$ a complex constant. There is a similar expression for $\mathbf{H}^{(+)}$ in terms of $H_0^{(+)}$

(a) Show that the time-averaged energy density is given by

$$\langle w\rangle = \frac{1}{4}\left(\epsilon E_0^2 + \mu_0 H_0^2\right),$$

where $E_0 = 2|E_0^{(+)}|$. What is the significance of the extra factor of $1/2$ in this expression? (Hint: it's not the definition of E_0.)

(b) What is the real wave corresponding to the complex wave given above? Be explicit about the parameter dependence.

(c) Calculate w, the Poynting vector $\mathbf{S}$, and the intensity I for the real plane wave of part (b).

(d) Show, for the real plane wave of part (b), that $\mathbf{S} = cw\hat{x}$. With the interpretation that the Poynting vector represents an energy flux density, argue that this means that the electromagnetic energy of a plane wave propagates with velocity c.

■

Playing with 2D arrays, plotting images, animations (5 pts) The key to plotting 2D arrays or images is the function `imshow`. It will show whatever array you give it. You often need to do bookkeeping to keep track of what the array *means* and what the x and y coordinate of each array element is. To that end, start here:

```
N = 11
axis_x = np.linspace(-5,5,N)  # 1D array of x values
axis_y = np.linspace(-5,5,N)  # 1D array of y values
X,Y = np.meshgrid(axis_x,axis_y)  # Each is a 2D array that varries along 1 dim
```

Look at each of these arrays just by evaluating each. Now try evaluating each of the following: `imshow(X)`, `imshow(Y)`, and `imshow(X+Y)`. Note that the axis tick labels act as if they are labeling the rows and columns of a matrix. First, they don't know to go from -5 to +5. Second, the y axis starts at the top and goes down like it's numbering rows of a matrix. to fix this, you need to use two options in `imshow`: `origin` to flip the image from "matrix rows" to "plot axis" and `extent` to tell it the boundaries of the array's x and y coordinates:

```
extents = (min(axis_x),max(axis_x),min(axis_y),max(axis_y))
imshow(X+Y, extent=extents, origin='lower')
```

Even this isn't quite right, since the coordinates are supposed to represent the *center* of the pixels. Extending by a half pixel on each side does the trick, but it's not necessary for large `N`:

```
dx = axis_x[1]-axis_x[0]
dy = axis_y[1]-axis_y[0]
extents = (min(axis_x)-dx/2,max(axis_x)+dx/2,min(axis_y)-dy/2,max(axis_y)+dy/2)
```

While we're at it, we usually want to show intensities in a gray-scale colormap:

```
imshow(X+Y, extent=extents, origin='lower', cmap='gray')
```

Now you can increase N to something more reasonable like 500 and plot other functions, like the amplitude of a plane wave, say `cos(2*X+4*Y)`.
To do an animation, you need to start a new notebook with `%pylab notebook` as the first cell. In `notebook` mode, remember to close each figure as you make them with the little "power off" button—if you don't, the next figure will be drawn over the previous figure.
To make an animation, you need to import the animation package, define a list of times for each frame, define a function that updates the array at each step, and call an animation function. With the relevant variables defined again in the new notebook, try animating the plane wave:

```
import matplotlib.animation
t = np.linspace(0,1,num=20)  # time steps. 20 steps from 0 to 1
omega = 2*pi*1 # in ths example, phase advances 2pi as we get to the last step

fig, ax = plt.subplots()  # make a figure that we'll modify
A = cos(2*X+4*Y)  # Need something to plot; this sets the color scale
im = ax.imshow(A, extent=extents, origin='lower', cmap='gray')

def animate(i):
    # This will get called with i going from 0 to 19
    A = cos(2*X+4*Y - omega*t[i])
    im.set_data(A)

ani = matplotlib.animation.FuncAnimation(fig, animate, frames=len(t))
```

If you want to get really fancy and show the animation with controls and a nifty slider, try:

```
from IPython.display import HTML
HTML(ani.to_jshtml())
```

Steck 5.6, modified to fit our setup with extra Python animations (10 pts) Consider a Michelson interferometer, with a 50/50 beam splitter.
Suppose a *spherically expanding* beam enters the interferometer from a focused laser spot or from a lamp with a small hole in front of it. An outgoing spherical wave has the form

$$E^{(+)}(r,t) = \frac{A}{r} e^{i(kr-\omega t)},$$

where the constant A has to do with the brightness of the source and r is the distance from the source. You could check that this satisfies the wave equation in spherical coordinates, but don't bother—you'll do that in quantum and other classes.
Assume the beam and mirrors are all ideally aligned, but that the two mirrors differ by some small amount Δz. Show that there will be a "bull's-eye" intensity pattern of concentric interference fringes on the screen, provided that the arm lengths are not equal. You'll do this two ways (and you can do them in either order): First with an analytical approximation, and then by making a 2D animation of the intensity as Δz changes a little.
The intensity I is most usefully given by equation 4.66:

$$I = \frac{2|E^{(+)}|^2}{\eta} \qquad \text{where} \qquad \eta := \frac{1}{n}\sqrt{\frac{\mu_0}{\epsilon_0}}$$

You don't need to do calculations involving all of the reflections off of each of the mirrors and beam splitter—just "unroll" the interferometer treat it as a superposition of two waves, one from each path. Call z_1 the total distance along the optical path through the interferometer reflecting off of mirror 1 and similarly for z_2. Let $z_2 = z_1 + \Delta z$.
Hints for Analytical approximation: Make the approximation that z_1 is much bigger than both Δz and the x and y coordinates. The small screen only captures light that actually makes it through the interferometer—most of the spherical wave misses the interferometer. To quote the original problem from the text book, "Try to focus only on the important stuff." This means, for example, that you can ignore the slightly different amplitudes due to the slightly farther path length, but you you need to be more careful about the phase difference.
Jupyter Exercise with Numbers and Plots: Consider the interferometer in lab with a HeNe laser. Look up the wavelength. The total path length from the source, through the mirrors of the interferometer, to the screen is about 70 cm. The screen is around 2 cm by 2 cm (1 cm on each side of the optical axis). Make an animation of the intensity on the screen as the difference in mirror positions grows from being different by 3 cm to being different by 3 cm plus an extra wavelength (the first and last frame should look the same). No need to do any Taylor expansions here.

Index of Refraction of Air (20 pts) Watch the video I made about the interferometer where I take data. When the valve is open, the incoming air molecules scatter light, and hence they increase the refractive index and increase the optical path length of that leg of the interferometer. All of the data I took is in `Lab2InterferometerData.zip` on Sakai. You can read an oscilloscope comma-separated variable (CSV) file with the following command, examples of which are in a jupyter notebook in the zip file:

```
data = loadtxt('SDS00005.csv', delimiter=',', skiprows=12)
```

The length of the inside of the vacuum vessel is 2.0165 ± 0.0005 inches long.
Since the index of refraction of air is so close to 1, you will find, report, and compare $n - 1$.
There are 9 data runs. You need to use at least 5 to get a mean and standard error for each data point. I'd start with the later runs first because I probably got better at doing the experiment. There may also be runs that you need to throw out—I didn't mess up on purpose to trick you, but I also haven't done this analysis myself yet on this data.
The first step is to count the fringes that have gone by at each mark. This should probably be done "by eye" by zooming in on the plots (use `%pylab notebook`). You should be able to estimate within a quarter of a fringe or better. The second step is to plot the number of fringes recorded since the valve was opened versus the change in chamber pressure since the valve was opened. The plot should be linear.
A least-squares fit of the data to a straight line should yield a y intercept of zero (within uncertainty), and the slope of the fit can be used to deduce a value of $n - 1$ for air at atmospheric pressure. This translation requires thinking about what's actually going on in the interferometer, what the extra optical path length does, and what the index of refraction means.
If you don't get an intercept of zero, it's possible that a perfect vacuum was not achieved each run or that the pressure gauge has a systematic shift. In this case, begin counting with the first mark.
Compare your value for $n - 1$ with the value tabulated in, for example, the CRC Handbook, the relevant page of which is included below. You will have to take into account the difference of the laboratory temperature from the temperature at which the tabulated value was measured. Convert your measured value of $n - 1$ to a value at the temperature quoted in the literature (e.g., 15°C). (The ideal gas law is adequate for this purpose.)

■

226.503 1.51941
231.288

INDEX OF REFRACTION OF AIR

Corrections for reducing wavelengths and frequencies in air (15°C, 760 mmHg) to vacuo

The indices were computed from the Cauchy formula $(n - 1)10^7 = 2726.43 + 12.288/(\lambda_2 \times 10^{-8}) + 0.3555/(\lambda^4 \times 10^{-16})$. For 0°Cand 76 cm Hg the constants of the equation become 2875.66, 13.412 and 0.3777 respectively, and for 30°C and 76 cm Hg 2589.72, 12.259 and 0.2576. Sellmeier's formula for but one absorption band closely fits the observations: $n_2 = 1 + 0.00057378\lambda^2/(\lambda^2 - 595260)$. If n — 1 were strictly proportional to the density, then $(n - 1)_o/(n - 1)t$ would equal $1 + \alpha t$ where α should be 0.00367. The following values of α were found to hold:

λ	0.85 mμ	0.75 mμ	0.65 mμ	0.55 mμ	0.45 mμ	0.35 mμ	0.25 mμ
α	0.003672	0.003674	0.003678	0.003685	0.003700	0.003738	0.003872

The indices are for dry air (0.05 ± % CO_2). Corrections to reduce to dry air the indices for moist air may be made for any wavelength by Lorenz's formula, +0.000041(m/760), where m is the vapor pressure in mm. The corresponding frequencies in waves per cm and the corrections to reduce wavelengths and frequencies in air at 15°C and 760 mmHg pressure to vacuo are given. E.G., a light wave of 5000 angstroms in dry air at 15°C, 760 mmHg becomes 5001.391 Å in vacuo; a frequency of 20,000 waves per cm correspondingly becomes 19994.44.

Wavelength, λ angstroms	Dry air (n − 1) × 10^7 15°C 760 mmHg	Vacuo correction for λ in air (nλ — λ) add	Frequency waves per cm 1/λ in air	Vacuo correction or 1/λ in air (1/nλ — 1/λ) subtract
2000	3256	.651	50,000	16.27
2100	3188	.670	47,619	15.18
2200	3132	.689	45,454	14.23
2300	3086	.710	43,478	13.41
2400	3047	.731	41,666	12.69
2500	3014	.754	40,000	12.05
2600	2986	.776	38,461	11.48
2700	2962	.800	37,037	10.97
2800	2941	.824	35,714	10.50
2900	2923	.848	34,482	10.08
3000	2907	.872	33,333	9.69
3100	2893	.897	32,258	9.33
3200	2880	.922	31,250	9.00
3300	2869	.947	30,303	8.69
3400	2859	.972	29,411	8.41
3500	2850	.998	28,571	8.14
3600	2842	1.023	27,777	7.89
3700	2835	1.049	27,027	7.66
3800	2829	1.075	26,315	7.44
3900	2823	1.101	25,641	7.24
4000	2817	1.127	25,000	7.04
4100	2812	1.153	24,390	6.86
4200	2808	1.179	23,809	6.68
4300	2803	1.205	23,255	6.52
4400	2799	1.232	22,727	6.36
4500	2796	1.258	22,222	6.21
4600	2792	1.284	21,739	6.07
4700	2789	1.311	21,276	5.93
4800	2786	1.338	20,833	5.80
4900	2784	1.364	20,406	5.68
5000	2781	1.391	20,000	5.56
5100	2779	1.417	19,607	5.45
5200	2777	1.444	19,230	5.34
5300	2775	1.471	18,867	5.23
5400	2773	1.497	18,518	5.13

Wavelength, λ angstroms	Dry air (n − 1) × 10^7 15°C 760 mmHg	Vacuo correction for λ in air (nλ — λ) add	Frequency waves per cm 1/λ in air	Vacuo correction for 1/λ in air (1/nλ — 1/λ) substract
5500	2771	1.524	18,181	5.04
5600	2769	1.551	17,857	4.94
5700	2768	1.578	17,543	4.85
5800	2766	1.604	17,241	4.77
5900	2765	1.631	16,949	4.68
6000	2763	1.658	16,666	4.60
6100	2762	1.685	16,393	4.53
6200	2761	1.712	16,129	4.45
6300	2760	1.739	15,873	4.38
6400	2759	1.766	15,625	4.31
6500	2758	1.792	15,384	4.24
6600	2757	1.819	15,151	4.18
6700	2756	1.846	14,925	4.11
6800	2755	1.873	14,705	4.05
6900	2754	1.900	14,492	3.99
7000	2753	1.927	14,285	3.93
7100	2752	1.954	14,084	3.88
7200	2751	1.981	13,888	3.82
7300	2751	2,008	13,698	3.77
7400	2750	2.035	13,513	3.72
7500	2749	2.062	13,333	3.66
7600	2749	2.089	13,157	3.62
7700	2748	2.116	12,987	3.57
7800	2748	2.143	12,820	3.52
7900	2747	2.170	12,658	3.48
8000	2746	2.197	12,500	3.43
8100	2746	2.224	12,345	3.39
8250	2745	2.265	12,121	3.33
8500	2744	2.332	11,764	3.23
8750	2743	2.400	11,428	3.13
9000	2742	2.468	11,111	3.05
9250	2741	2.536	10,810	2.96
9500	2740	2.604	10,526	2.88
9750	2740	2.671	10,256	2.81
10000	2739	2.739	10,000	2.74