PH4606 - Lecture 2

Claus-Dieter Ohl

March 22, 2016

1 Harmonic Plane Waves

In the following we will consider a constant speed of sound and start with the description of a plane wave propagation. Planar waves have a constant amplitude and phase perpendicular to the direction of propagation. Thus if the wave moves in the x-plane we have:

$$\frac{\partial^2 p}{\partial x^2} = \frac{1}{c^2}\frac{\partial^2 p}{\partial t^2} \quad . \tag{2.1}$$

The complex solution to the harmonic wave equation is

$$p = A\, e^{i(\omega t - kx)} + B\, e^{i(\omega + kx)} \quad . \tag{2.2}$$

Using the linear Euler Eq. (1.11) we can obtain the particle velocity in a plane wave as

$$\mathbf{u} = u\,\vec{e}_1 = \left[\frac{A}{\rho_0 c} e^{i(\omega t - kx)} - \frac{B}{\rho_0 c} e^{i(\omega t + kx)}\right]\vec{e}_1 \tag{2.3}$$

Please note that the constants A and B, pressures and velocities are now all complex. We can separate the wave travelling in positive x-direction from the the wave travelling in negative x-direction and designate them with the subscript $+$ and $-$, respectively.

The wave travelling in positive x-direction is

$$p_+ = Ae^{i(\omega t - kx)} \tag{2.4}$$

while in negative x-direction is

$$p_- = Be^{i(\omega t + kx)} \tag{2.5}$$

These two waves can be formally written for the other acoustic variables:

$$u_\pm = \pm\frac{p_\pm}{\rho_0 c} \tag{1}$$

$$s_\pm = \frac{p_\pm}{\rho_0 c^2} \tag{2}$$

$$\Phi_\pm = -\frac{p_\pm}{i\omega\rho_0} \tag{2.8}$$

Plane waves travelling in an arbitrary direction can be written as

$$p = Ae^{i(\omega t - k_x x - k_y y - k_z z)} \tag{2.9}$$

If we insert Eq. (2.9) into the wave equation, Eq. (1.17) we obtain the condition

$$\left(\frac{\omega}{c}\right)^2 = k_x^2 + k_y^2 + k_z^2 \quad . \tag{2.10}$$

Now we can introduce the propagation vector, $\mathbf{k}$ of the wave as

$$\mathbf{k} \equiv k_x \mathbf{e}_1 + k_y \mathbf{e}_2 + k_z \mathbf{e_3} \tag{2.11}$$

with

$$|\mathbf{k}| = \frac{\omega}{c} \quad . \tag{2.12}$$

Thus using the position vector $\mathbf{r} = x\mathbf{e}_1 + y\mathbf{e}_2 + z\mathbf{e}_3$ we can rewrite the plane wave solution, Eq. (2.9), as

$$p = Ae^{i(\omega t - \mathbf{kr})} \quad . \tag{2.13}$$

The propagation vector points in the direction of wave propagation which can be simply shown. Surfaces of constant phase are defined by the scalar product $\mathbf{k} \cdot \mathbf{r} = \text{const.}$ The vector $\nabla(\mathbf{k} \cdot \mathbf{r})$ is a vector in the normal direction to the constant phase surface. But $\mathbf{k} = \nabla(\mathbf{k} \cdot \mathbf{r})$, thus $\mathbf{k}$ is pointing normal to the plane of constant phase and thus in the direction of propagation.

1.1 Example

Let a plane wave have a constant phase parallel to the z-axis, thus the wave travels in the xy-plane as sketched in Fig. 2.1. Formally the surface of constant phase is $\mathbf{k} \cdot \mathbf{r} = 0$; thus if the phase is independent of z, $\mathbf{k}$ can only have k_x and k_y components. We can thus write Eq. (2.9) as

$$p = Ae^{i(\omega t - k_x x - k_y y)} \tag{2.14}$$

and the equation for the surface of constant phase is

$$k_x x + k_y y = \text{const.} \tag{2.15}$$

or

$$y = -\frac{k_x}{k_y} x + \text{const.} \quad . \tag{2.16}$$

Equation (2.16) is line with a slope $-k_x/k_y$. The surfaces are thus oriented as sketched in Fig. 2.1.

Figure 2.1

Now let's study this wave in the x-direction while $y = 0$

$$p = Ae^{i(\omega t - k_x x)} \tag{2.17}$$

where $k_x = 2\pi/\lambda_x$ with λ_x being the apparent wavelength in x-direction. Note that this wavelength is larger than the wavelength in $\vec{k}$-direction. We can relate λ_x with the help of Fig. 1 to $\cos\phi = \lambda/\lambda_x$ and thus $k_x = k\cos\phi$. Applying the same arguments in the y-direction gives

$$\mathbf{k} = k\cos\phi\,\mathbf{e}_1 + k\sin\phi\,\mathbf{e}_2 \quad . \tag{2.18}$$

Thus an harmonic plane wave in 2-dimensions propagating under an angle ϕ to the x-axis can be written as

$$p = Ae^{i(\omega t - kx\cos\phi - ky\sin\phi)} \tag{2.19}$$

```
## Simulation of a 2-dimensional wave
```

```
%matplotlib notebook
import math as m
import numpy #array operations
import matplotlib.pyplot as plt #plotting
from ipywidgets import widgets #for the widgets
#from ipykernel.client import display
from IPython import display #for continous display
#from PIL import Image #to export images

nimg=0

def savemyimage(visual):
  global nimg
  visual = (visual +2.)/4.
  #result = Image.fromarray((visual * 255).astype(numpy.uint8))
  #result.save('out{:03d}.bmp'.format(nimg))
  nimg=nimg+1

def plotwave(u,time,px,py,pp,pt):
  plt.figure(1)
  plt.clf()
  plt.subplot2grid((4,4),(0,0), colspan = 4, rowspan = 3)
  plt.imshow(u, origin='upper', extent=[0., 2., 0., 2.], vmax=2, vmin=-2) #plot the wave field
  plt.text(0.1,1.8,"time {0:.5f}".format(time)) #annotate the time
  plt.plot(w_probex.value,1.,'o') #position of probe
  plt.subplot2grid((4,4),(3,0), colspan = 4)
  plt.plot(pt,pp) #plot the pressure at the probe
  plt.gca().set_ylim([-2,2])
  display.clear_output(wait=True)
  display.display(plt.gcf())

def solvewave(b):
  tabs.visible=False
  #computational domain
  nx = ny = 381
  size=2. #size of the domain
  #parameters of the wave
  c = 5. #speed of sound
  l=w_wavelength.value #wavelength
  nu=c/l #frequency
  omega=nu*2.*m.pi #angular frequency
  duration=w_sourceduration.value/nu #duration of source
  #position
  emissionlength=(w_sizeemit.value/100.)*nx
  startx=int(nx/2-emissionlength/2)
  endx=int(nx/2+emissionlength/2)
```

```python

#further variables
dx = size/(nx-1)
CFL=0.1 #CFL number <1
dt = CFL*dx/c
nt=int(w_simduration.value/dt) #number of time steps
if w_position.value=='Left':
    sourcepos=0
else:
    sourcepos=int(nx/2)

#arrays for measuring the pressure at a position
pt=numpy.arange(nt+1)*dt
pp=numpy.zeros(nt+1)
px=int(w_probex.value*(nx-1)/2.)
py=int(ny/2)

#every xx times over the total nt timesteps an output should be generated
output=map(int,list(numpy.linspace(1,nt,int(nt/50))))

u  = numpy.zeros((nx,ny)) #pressure at t
un = numpy.zeros((nx,ny)) #pressure at t-dt
unn= numpy.zeros((nx,ny)) #pressure at t-2*dt

row, col = u.shape
#Assign initial conditions, for a sine wave
un[startx:endx,0]=0.    #amplitude is sin(omega*t) with t=0
unn[startx:endx,0]=1.   #velocity  is cos(omega*t) with t=0

C=c*c*dt*dt/dx/dx

plt.figure(1, figsize=(8, 8), dpi=300)

for n in range(nt+1): ##loop across number of time steps
    #this line computes the finite differences of the wave equation
    u[1:-1,1:-1]=2.*un[1:-1,1:-1]-unn[1:-1,1:-1]+C*(un[1:-1,:-2]+\
                    un[:-2,1:-1]+un[2:,1:-1]+un[1:-1,2:]-4.*un[1:-1,1:-1])

    #hard reflective boundary conditions
    u[0,:] = u[1,:]
    u[-1,:] = u[-2,:]
    u[:,0] = u[:,1]
    u[:,-1] = u[:,-2]

    #pressure source
    #if float(n*dt<duration):
    u[startx:endx,sourcepos]=m.sin(omega*n*dt)*float(n*dt<duration)

    pp[n]=u[py,px]

    #save values for the time derivative
    unn=un.copy() #n-1 time steop
    un=u.copy()   #n time step

    if (n in output):
        plotwave(u,n*dt,px,py,pp,pt)
        if (w_saveplots.value):
            savemyimage(u)

```

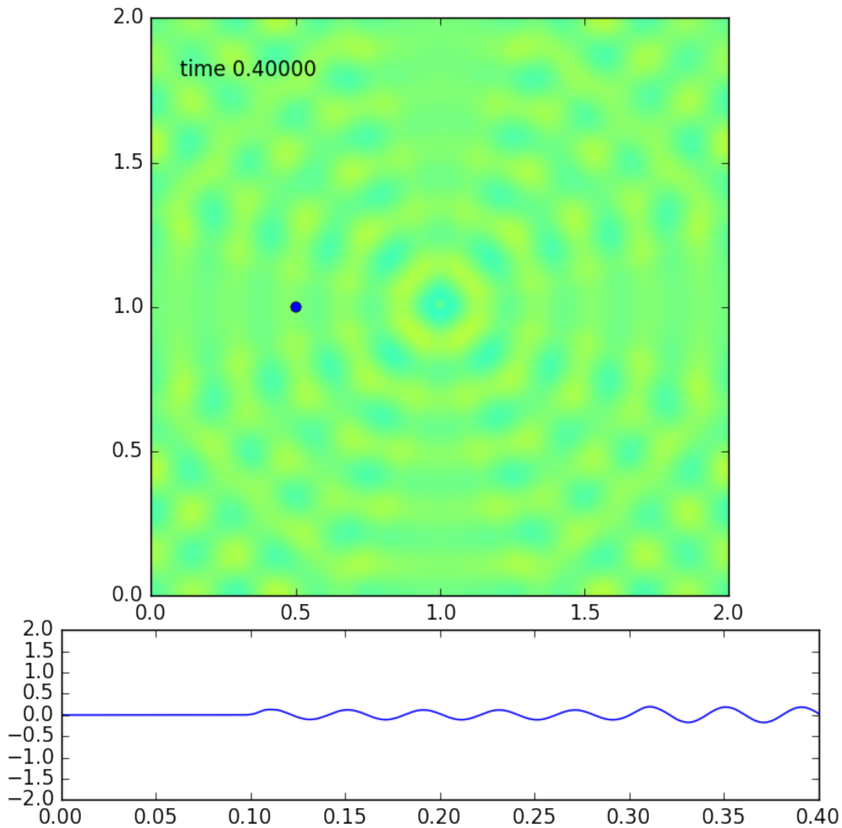

Figure 1:

```
#and plot the last figure
plotwave(u,n*dt,px,py,pp,pt)
tabs.visible=True
```