DOING PHYSICS WITH PYTHON

FOURIER METHODS

FOURIER TRANSFORMS

SINUSOIDAL FUNCTIONS

Ian Cooper
matlabvisualphysics@gmail.com

DOWNLOAD DIRECTORIES FOR PYTHON CODE

Google drive

GitHub

simFM003.py simFM003A.py simFM003B.py

The Codes are used to calculate the Fourier transform and the power spectral density functions by the direct integration of the Fourier integrals using Simpson's rule.

For a more in-depth discussion on the Fourier transform visit

FOURIER TRANSFORMS / GAUSSIAN FUNCTION

INTRODUCTION

Many physical processes can be described in the **time domain** by the values of a function $h(t)$ or else in the **frequency domain** $H(f)$. The function $H(f)$ is usually complex function specified by its amplitude and phase.

Fourier transform equations

(1) $$H(f) = \int_{-\infty}^{\infty} h(t) e^{i(2\pi f t)}\, dt$$

(2) $$h(t) = \int_{-\infty}^{\infty} H(f) e^{-i(2\pi f t)}\, df$$

Equation 1 is the **Fourier transform** and equation 2 gives the **inverse Fourier transform**. If t is measured in seconds, then the frequency f is measured in hertz and the angular frequency ω in radians.

The total power in a signal is the same whether we compute it in the time domain or in the frequency domain. This result is called **Parseval's theorem**

(3) $$Total\ Power = \int_{-\infty}^{\infty} |h(t)|^2\, dt = \int_{-\infty}^{\infty} |H(f)|^2\, df$$

If one wants to compute how much power there is the frequency interval between f and df, one does not usually distinguish between negative and positive values of f, but rather regards f as varying from zero frequency or DC $(f = 0)$ to $f \to \infty$. In such cases, one defines the **one-sided power spectral density** psd of the function $h(t)$ as

(4) $$psd(f) = |H(f)|^2 + |H(-f)|^2$$

If $h(t)$ is a real function, then

(5) $$psd(f) = 2|H(f)|^2 \qquad 0 \le f < \infty$$

(6) $$Total\ Power = 2\int_0^\infty |H(f)|^2\, df$$

PHASE of $H(f)$

The Fourier transform converts a time-domain signal into a complex-valued frequency spectrum. While the **magnitude** tells you the *strength* (energy or amplitude) of each frequency component, the **phase** tells you the *timing, alignment, or relative shift* of those frequency components.

Starting Position / Time Offset: The phase angle ϕ at any given frequency ω represents the horizontal shift (the starting angle or

delay) of that specific sine or cosine wave. A time shift in the original signal (e.g., delaying a pulse or audio wave by t_0) does not change the frequency magnitudes, but it causes a linear change in the phase spectrum.

Waveform Shape and Feature Alignment: Sharp features in a signal—like edges in an image or transients and spikes in audio—depend heavily on the phase. When multiple sinusoids interfere constructively or destructively at exact moments in time, the phase orchestrates *where* those peaks and spikes land.

Sine vs. Cosine Balance: The phase describes the proportion of sine and cosine waves making up that frequency (e.g., 0° corresponds purely to a cosine wave, while 90° corresponds to a sine wave).

How Phase is Calculated:

$$\phi = \tan^{-1}\left(\frac{\mathrm{Im}(H(f))}{\mathrm{Re}(H(f))}\right)$$

Image Reconstruction: In images (2D Fourier transforms), the phase contains almost all of the *structural and edge information* (where objects and boundaries are), whereas the magnitude mostly contains average energy/contrast data. If you swap the phase spectra of two different images, the resulting reconstructed image takes on the shape/structure of the phase donor.

Signal Reconstitution: Each complex exponential coefficient encapsulates both magnitude and phase; dropping the phase discards the temporal alignment required to rebuild the original waveform accurately.

Simulation 1 Sinusoidal functions smFM003.py

We can easily computer the Fourier transform of the sinusoidal function expressed in the forms

(1.1) $$h(t) = A_0 \sin\left(2\pi f_0 t + \phi_0\right)$$

(1.2) $$h(t) = A_0 \cos\left(2\pi f_0 t + \phi_0\right)$$

Figures 1.1 and 1.2 show the plots of the sine and cosine functions given by equations 1.1 and 1.2. The plots for the Fourier transform $|H(f)|$ and $psd(f)$ are identical for the sine and cosines curves. The phase plots for the sine function and cosine function shown in figure 1.3 are however different. The real and imaginary parts of the Fourier transform $H(f)$ are shown in figure 1.4.

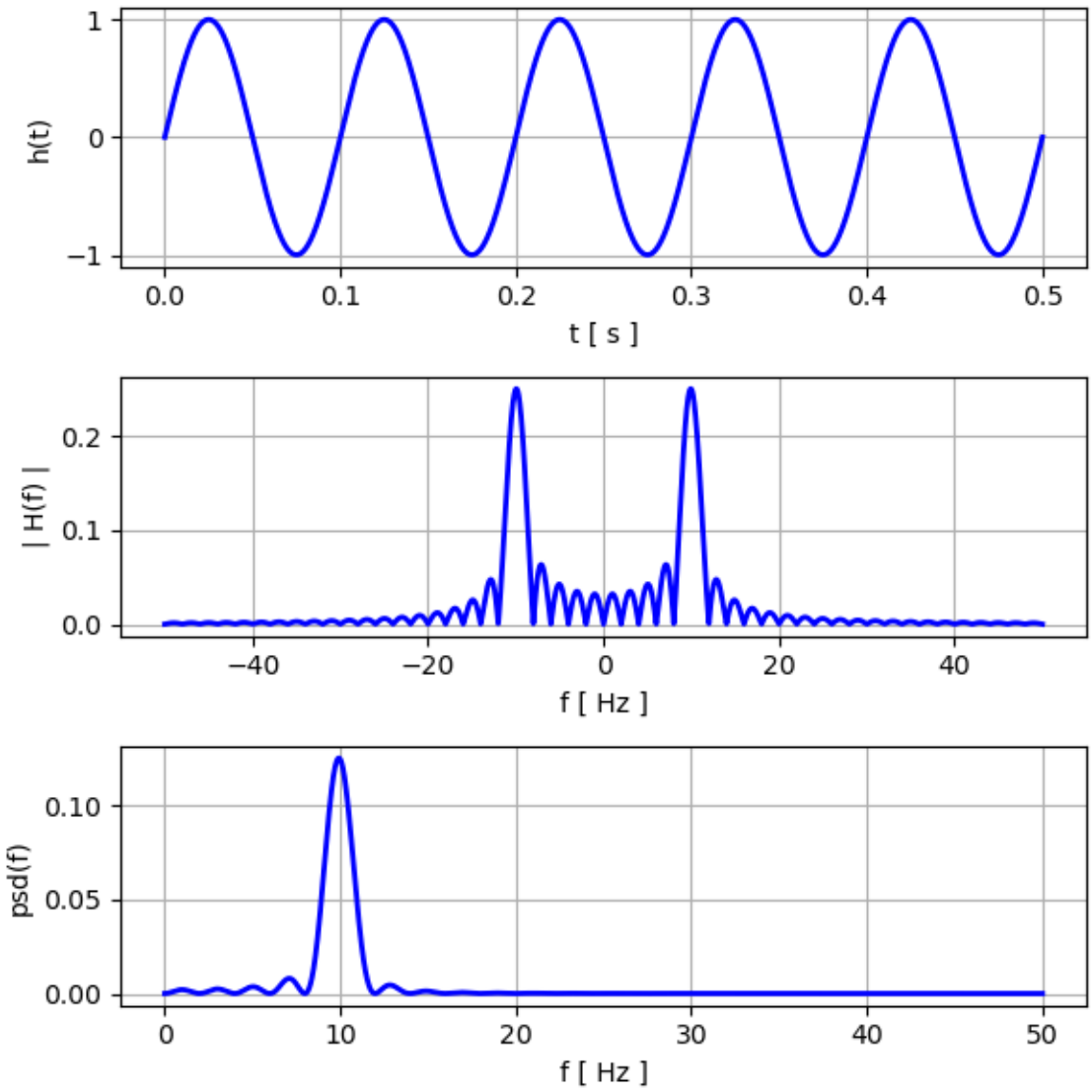


Fig. 1.1 $h(t) = A_0 \sin\left(2\pi f_0 t + \phi_0\right)$

$A_0 = 1.0 \quad f_0 = 10 \text{ Hz} \quad \phi_0 = 0 \text{ rad}$

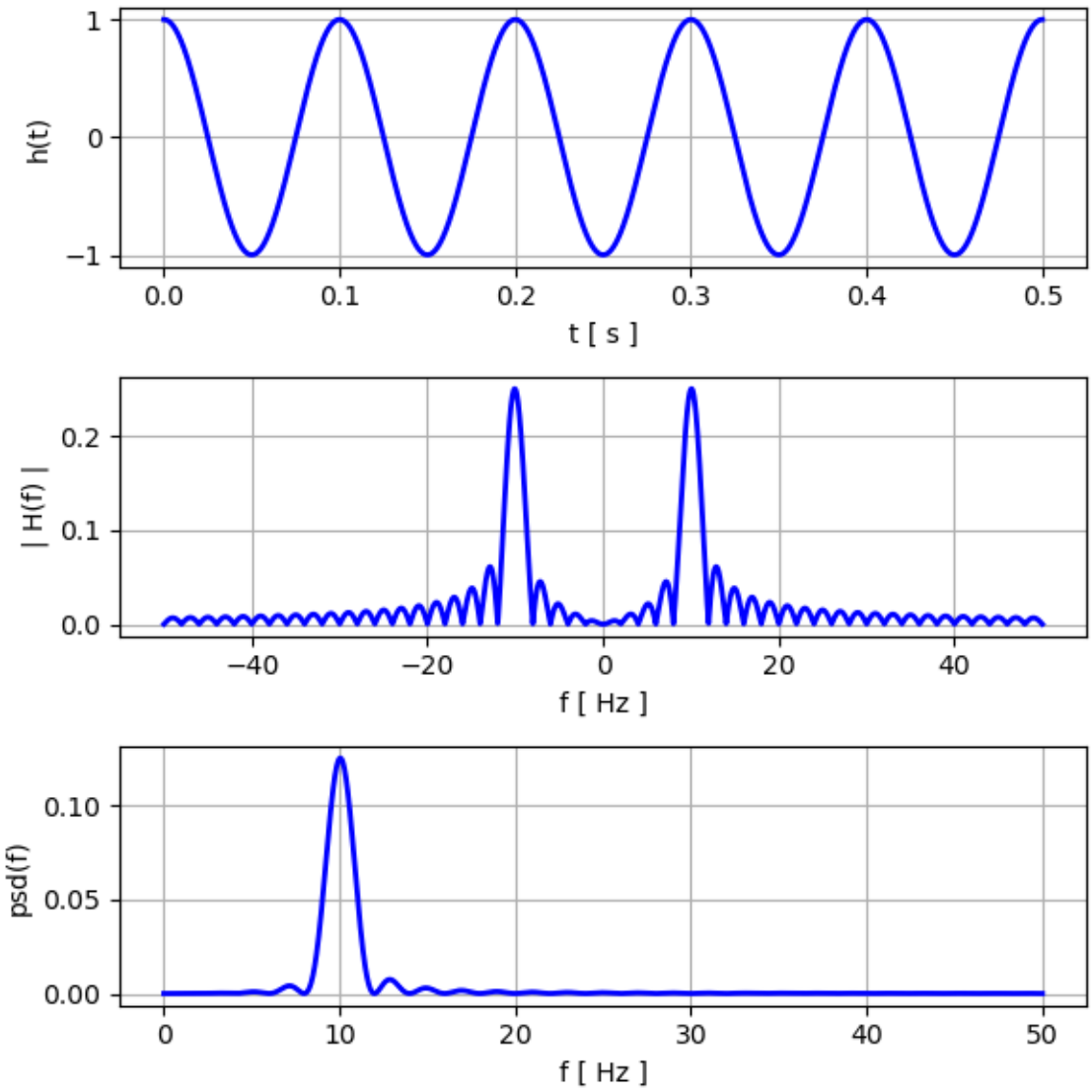


Fig. 1.2 $h(t) = A_0 \cos\left(2\pi f_0 t + \phi_0\right)$

$A_0 = 1.0 \quad f_0 = 10 \text{ Hz} \quad \phi_0 = 0 \text{ rad}$

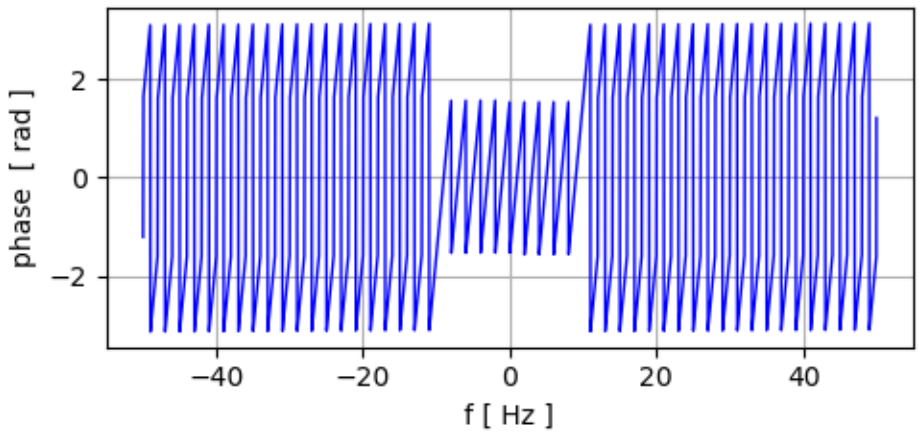


Fig. 1.3A Phase plot

$h(t) = A_0 \sin(2\pi f_0 t + \phi_0)$ $A_0 = 1.0$ $f_0 = 10$ Hz $\phi_0 = 0$ rad

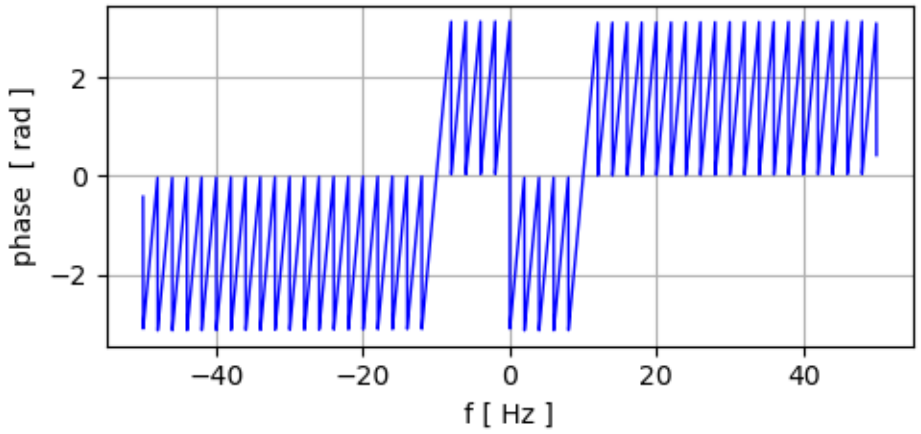


Fig. 1.3B. Phase plot

$h(t) = A_0 \cos(2\pi f_0 t + \phi_0)$ $A_0 = 1.0$ $f_0 = 10$ Hz $\phi_0 = 0$ rad

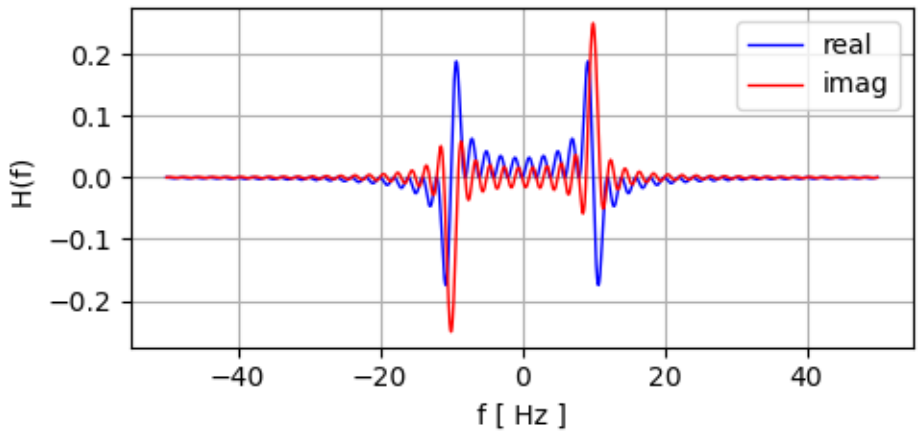


Fig. 1.4A. Real and imaginary parts of the Fourier transform.

$$h(t) = A_0 \sin\left(2\pi f_0 t + \phi_0\right) \quad A_0 = 1.0 \quad f_0 = 10 \text{ Hz} \quad \phi_0 = 0 \text{ rad}$$

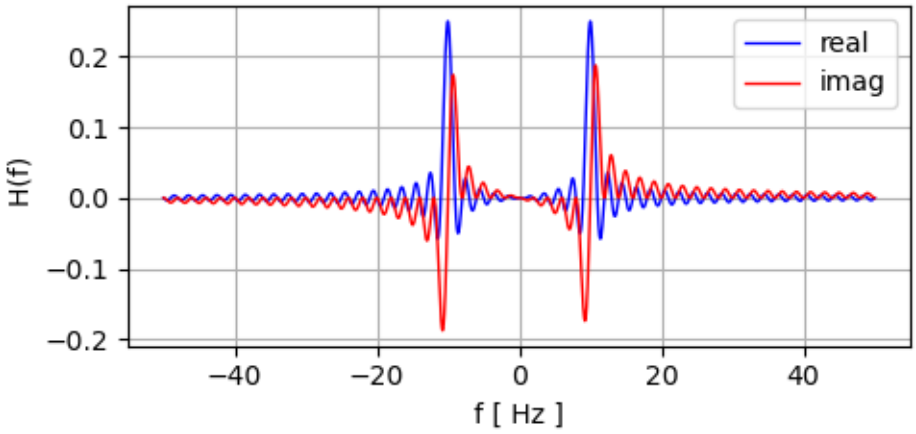


Fig. 1.4B. Real and imaginary parts of the Fourier transform.

$$h(t) = A_0 \cos\left(2\pi f_0 t + \phi_0\right) \quad A_0 = 1.0 \quad f_0 = 10 \text{ Hz} \quad \phi_0 = 0 \text{ rad}$$

Total power (equation 6)

$h(t) = A_0 \sin(2\pi f_0 t + \phi_0)$ $A_0 = 1.0$ $f_0 = 10$ Hz $\phi_0 = 0$ rad

Power (time domain) Pt = 0.250

Power (frequency domain) Pf = 0.250

$h(t) = A_0 \sin(2\pi f_0 t + \phi_0)$ $A_0 = 2.0$ $f_0 = 10$ Hz $\phi_0 = 0$ rad

Power (time domain) Pt = 1.00

Power (frequency domain) Pf = 1.00

$h(t) = A_0 \sin(2\pi f_0 t + \phi_0)$ $A_0 = 4.0$ $f_0 = 10$ Hz $\phi_0 = 0$ rad

Power (time domain) Pt = 4.00

Power (frequency domain) Pf = 4.00

The power of the function is proportional to the square of the amplitude: $P \propto A^2$

$h(t) = A_0 \sin(2\pi f_0 t + \phi_0)$ $A_0 = 1.0$ $f_0 = 5.0$ Hz $\phi_0 = 0$ rad

Power (time domain) Pt = 0.250

Power (frequency domain) Pf = 0.250

$h(t) = A_0 \sin(2\pi f_0 t + \phi_0)$ $A_0 = 1.0$ $f_0 = 20$ Hz $\phi_0 = 0$ rad

Power (time domain) Pt = 0.250

Power (frequency domain) Pf = 0.250

The power of the function is **independent** of the frequency.

Simulation 2 Superposition of sinusoidal functions

The Code **simFM003A.py** can be used for calculating the Fourier transform for any superposition of sinusoidal functions. One example is shown below.

```
#%% TIME DOMAIN
Nt = 1999
t1 = 0; t2 = 5
t = linspace(t1,t2,Nt)
A = array([1,1/3,1/5,1/7])
f0 = 5; f = array([1,3,5,7])
w = 2*pi*f
h = A[0]*sin(w[0]*t) + A[1]*sin(w[1]*t)+ A[2] *sin(w[2]*t) +
A[3]*sin(w[3]*t)

#%%  FREQUENCY DOMAIN
Nf = 3999
f1 = -12; f2 = 12
f = linspace(f1,f2,Nf)
p  = 1j*2*pi
HR = zeros(Nf); HI = HR; H = HR+HI*1j
# Fourier transform
for c in range(Nf):
    g = h*exp(p*f[c]*t)
    H[c] = simpson(real(g),t) + 1j*simpson(imag(g),t)

psd = 2*abs(H)**2           # one-sided power spectral density

phase = np.angle(H)

powerF = simpson(abs(H)**2,f)
powerT = simpson(abs(h)**2,t)
```

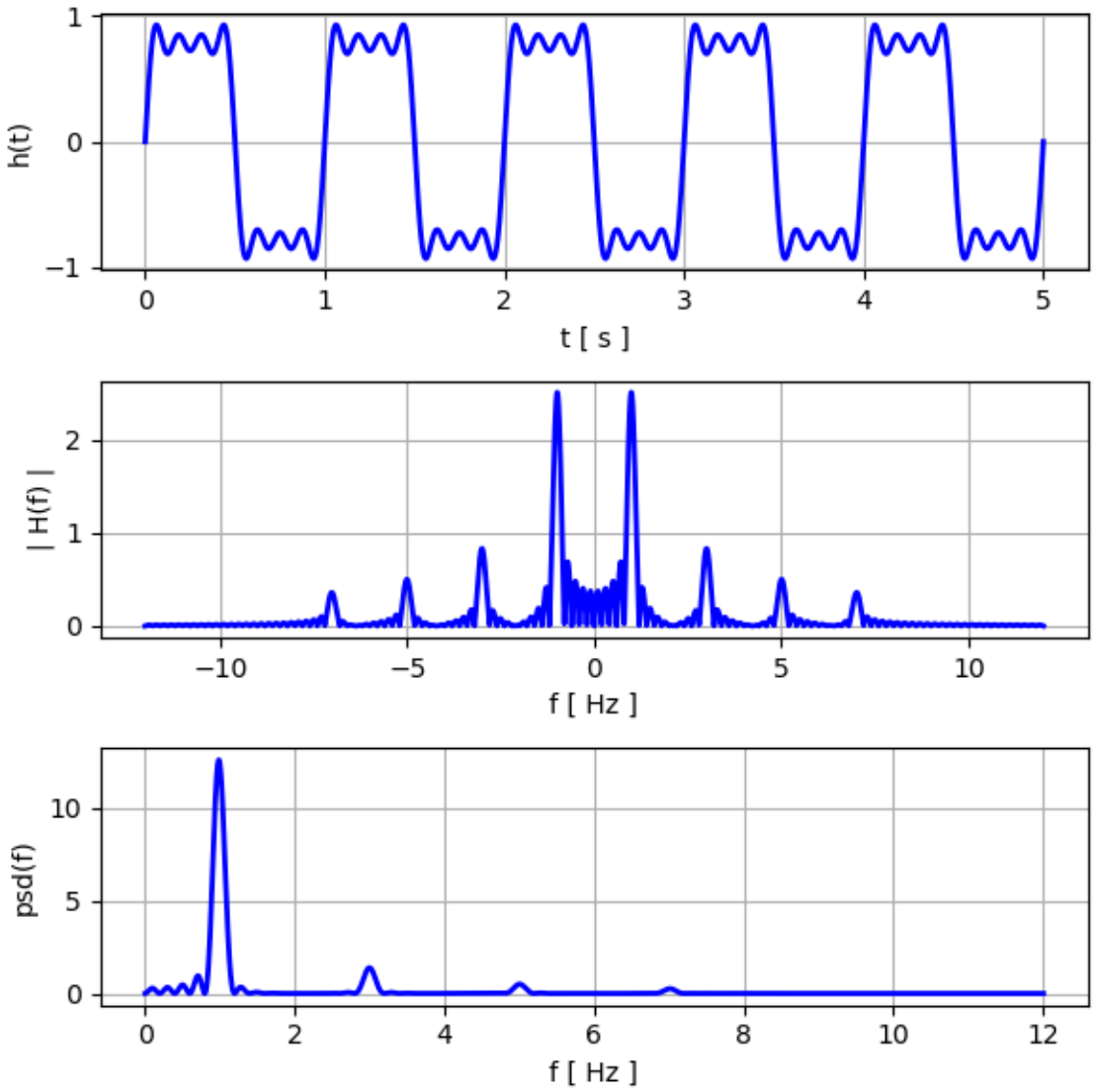


Fig. 2.1. Superposition of four sine functions to approximate a square wave. The series exhibits a non-uniformity of convergence near a discontinuity. Note, the overshoot, which is called the **Gibb's phenomena**. The greater the number of terms included in the series a better the approximation, however, the overshoot remains finite.

Simulation 3 Damped sine function simFM003B.py

The function for a damped sine wave can be expressed as

$$h(t) = A\exp(-t/\tau)\,\sin(2\pi\, f_0\, t)$$

This function might represent the displacement of a damped harmonic oscillator or the electric field in a radiated wave, or the current in an antenna.

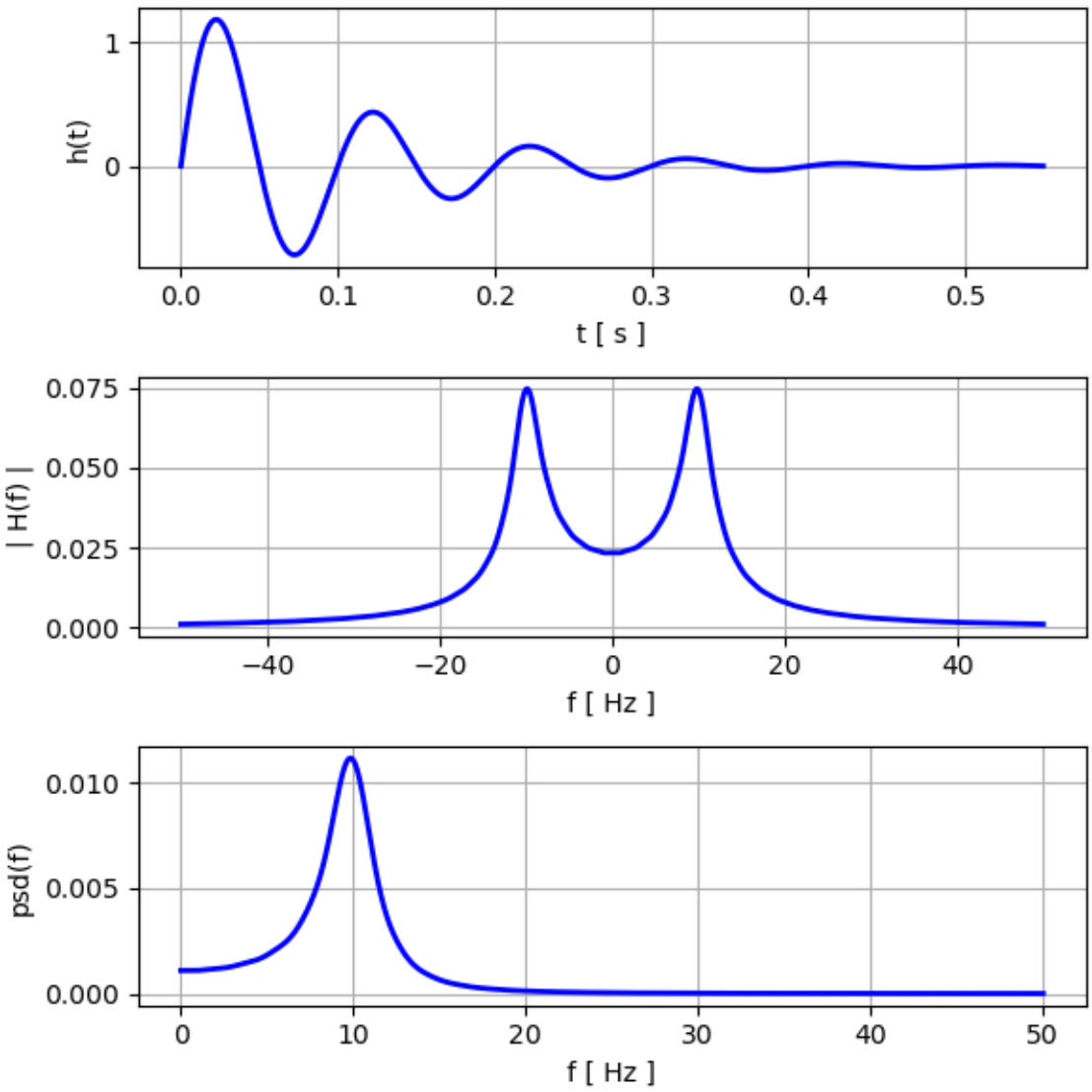


Fig. 3.1. A damped sine wave with $A = 1.5 \quad f_0 = 10 \text{ Hz} \quad \tau = 0.1 \text{ s}$

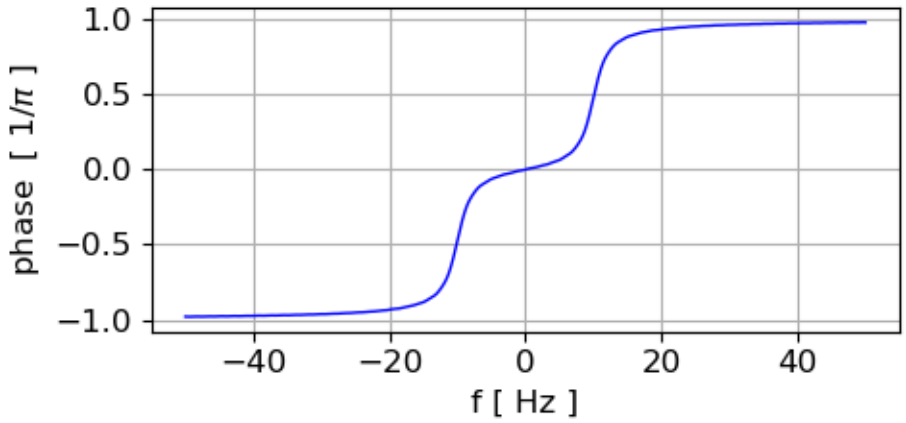


Fig. 3.2. Phase of the Fourier transform. Note: change in phase f = 0 Hz, ϕ = 0; f ~ 40 Hz, ϕ = π rad.

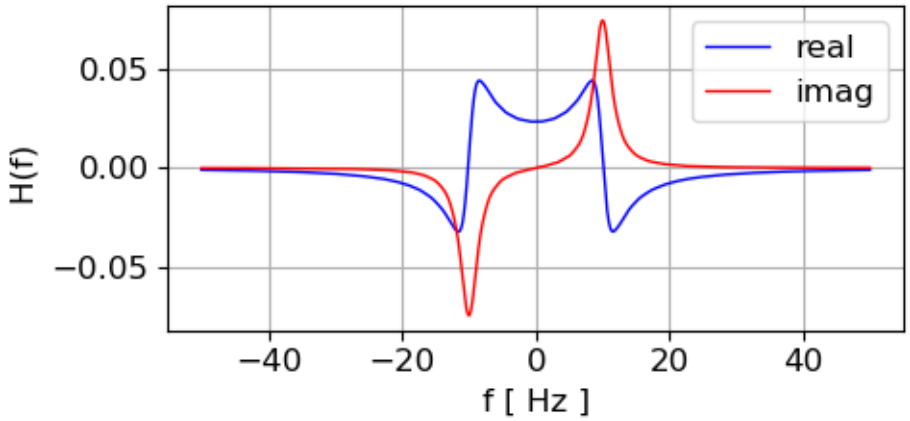


Fig. 3.3. Real and imaginary parts of the Fourier transform.