1. A time domain real-signal `x(t)` has a Fourier Transform property of `X(ω)=X*(-ω)`. Consider the following frequency domain description of a signal `G(ω)`:

$$G(\omega)=\begin{cases} 2, 5\le|\omega|\le 10 \\ 0, \text{elsewhere} \end{cases}.$$

(a) Evaluate `g(t)` using the definition of Inverse Fourier Transformation

$$\left(g(t)=\frac{1}{2\pi}\int_{-\infty}^{\infty} G(\omega)e^{j\omega t}\,d\omega\right)$$

Plot `G(ω)`, `Re(g(t))`, and `Im(g(t))` in a 3x1 subplot for the interval `ω=-31.4:0.01:31.4` and `t=-100:0.1:100`.

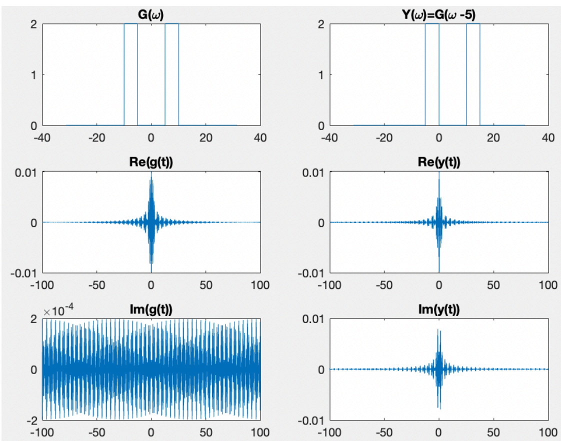
(b) Now consider `Y(ω)=G(ω-5)`. Plot `Y(ω)`, `Re(y(t))`, and `Im(y(t))` in a 3x1 subplot with the same intervals.

(c) Are `g(t)` and `y(t)` real-signal or complex signal?

Summary:

We define G(w) and g(t) as above. I use ifftshift(ifft()*dT) to do this. The calculation for Y(w) is G(w-5) and we call it to ifftshift(ifft()*dT) for y(t). We compare values of G(w) and its conjugate and do the same for Y(w) to determine that both g(t) and y(t) are real-signals (though I do see marginal imaginary components that make me question this).

Results:




```
>> lab5problem1
g(t) is real? 1
y(t) is real? 1
```

Code:

```
G = @(w) 2 * (abs(w) >= 5 & abs(w) <= 10); % define G(w)
w = -31.4:0.01:31.4; % w domain
t = -100:0.1:100; % t domain
g = @(t) ifftshift(ifft(G(t))*0.1); % define g(t)
y = @(t) ifftshift(ifft(G(t-5))*0.1); % define y(t)
subplot(3,2,1);
plot(w, G(w)); % plot G(w)
title("G(\omega)");
subplot(3,2,3);
plot(t, real(g(t))); % plot real components of g(t)
title("Re(g(t))");
subplot(3,2,5);
plot(t, imag(g(t))); % plot imaginary components of g(T)
title("Im(g(t))");
subplot(3,2,2);
plot(w, G(w-5)); % plot Y(w)
title("Y(\omega)=G(\omega -5)");
subplot(3, 2, 4);
plot(t, real(y(t))); % plot real components of y(t)
title("Re(y(t))");
subplot(3,2,6);
plot(t, imag(y(t))); % plot imaginary components of y(t)
title("Im(y(t))");
fprintf('g(t) is real? %d\ny(t) is real? %d\n', all(abs(G(w) - conj(G(-w))) < 1e-10),
all(abs(G(w-5) - conj(G(-(w-5)))) < 1e-10)); % determine if g(t) and y(t) are real
```

2. When the signal `g(t)` goes through a filter `h(t)` where the frequency domain definition of the filter is:

$$H(\omega)=\begin{cases} 5|\omega|, \ |\omega|\le 20 \\ 0, \text{elsewhere} \end{cases},$$

the results in a time domain output signal: `m(t)`.

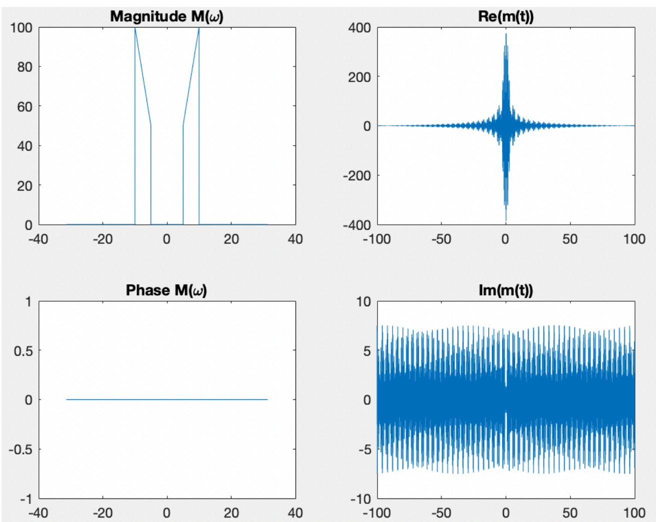
(a) Using convolution theorem, calculate the frequency domain output signal `M(ω)`. Plot the magnitude and phase of `M(ω)` in a 2x1 subplot for the interval `ω=-31.4:0.01:31.4`.

(b) Evaluate `m(t)` using the definition of Inverse Fourier Transformation. Plot `Re(m(t))` and `Im(m(t))` in a 2x1 subplot for the interval `t=-100:0.1:100`.

Summary:

We calculate M(w) as H(w)*G(w). We plot abs(M(w)) and angle(M(w)). We plot the real and imaginary components of m(t) after using ifftshift(ifft()*dT) to derive it from M(w).

Results:



Code:

```
function y = M(w) % define M(w) as H(w)*G(w)
   a = 5 * abs(w) .* (abs(w)<=20);
   b = 2 * (abs(w) >= 5 & abs(w) <= 10);
   y = a .* b;
end
w = -31.4:0.01:31.4; % domain w
subplot(2, 2, 1);
plot(w, abs(M(w))); % abs to get magnitude
title("Magnitude M(\omega)");
subplot(2, 2, 3);
plot(w, angle(M(w))); % angle to get phase
title("Phase M(\omega)");
m = ifftshift(ifft(M(t))*(t(2)-(1))); % define m(t)
subplot(2, 2, 2);
plot(t, real(m)); % plot real components of m(t)
title("Re(m(t))")
subplot(2, 2, 4);
plot(t, imag(m)); % plot imaginary components of m(t)
title("Im(m(t))");
```

3. Calculate the energy of the output signal `m(t)` for the time range `t=-100:0.1:100`. Also evaluate the energy of the output signal in frequency domain using Parseval's theorem (use the frequency range `ω=31.4:0.01:31.4`).

Summary:

We calculate energy for time domain as sum(abs(m(t))^2)*dT and for frequency domain as sum(abs(M(w))^2)*dW/2pi.

Results:

```
>> lab5problem3
Energy of m(t) is 2977.76 joules.
Energy of M(w) is 9295.98 joules.
```

Code:

```
function y = M(w) % define M(w)
   a = 5 * abs(w) .* (abs(w)<=20);
   b = 2 * (abs(w) >= 5 & abs(w) <= 10);
   y = a .* b;
end
function y = m(t) % define m(t)
   y = ifftshift(ifft(M(t))/(t(1)-t(2)));
end
w = -31.4:0.01:31.4; % domain w
t = -100:0.1:100; % domain t
E = sum(abs(m(t)).^2)*(t(2)-t(1)); % calculate energy m(t)
parsevals = sum(abs(M(w)).^2) * (w(2)-w(1))/ (2*pi); % calculate energy M(w)
fprintf("Energy of m(t) is %.2f joules.\n", E);
fprintf("Energy of M(w) is %.2f joules.\n", parsevals);
```