

CTFS Octave Codes (1A)

Copyright (c) 2009 - 2018 Young W. Lim.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

Please send corrections (or suggestions) to youngwlim@hotmail.com.

This document was produced by using OpenOffice and Octave.

Based on
M.J. Roberts, Fundamentals of Signals and Systems

Some utility function

```
function y = u(t)
    zro = t == 0; pos = t >0; y = zro/2 + pos;
endfunction
```

```
function y = ramp(t)
    y = t.*(t >= 0);
endfunction
```

```
function y = rect(t)
    y = u(t+0.5) - u(t-0.5);
endfunction
```

```
function y = tri(t)
    y = ramp(t+1) - 2*ramp(t) + ramp(t-1);
endfunction
```

M.J. Roberts, Fundamentals of Signals and Systems

Using the DFT to approximate the CTFS (1)

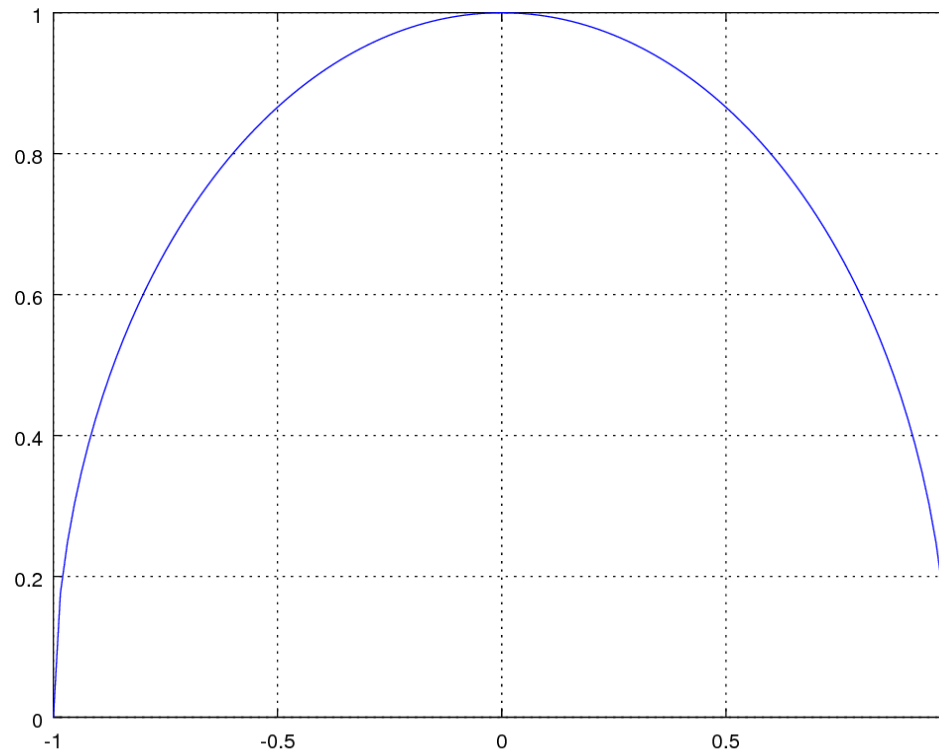
```
NF = 128;           % number of samples
T0 = 2;            % fundamental period
TF = T0;           % representation time
Ts = TF/NF;        % sampling period
Fs = 1/Ts;         % sampling frequency
n = [0:NF-1]';    % time index for sampling
t = n*Ts;          % sample times
```

M.J. Roberts, Fundamentals of Signals and Systems

Using the DFT to approximate the CTFS (1)

```
t = -1: Ts :+1;  
plot(t, x)
```

$$f(t) = \sqrt{1-t^2}, \quad (-1 < t < +1)$$

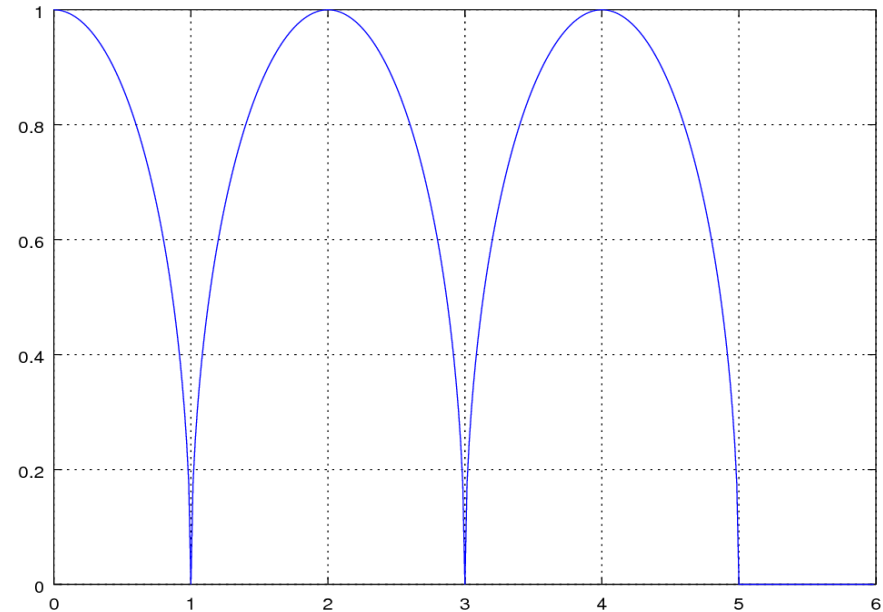


M.J. Roberts, Fundamentals of Signals and Systems

Using the DFT to approximate the CTFS (1)

$$f(t) \mathit{rect}(t/2) + f(t-2) \mathit{rect}((t-2)/2) + f(t-4) \mathit{rect}((t-4)/2)$$

```
NF=128;  
T0=2;  
TF=T0;  
Ts=TF/NF;  
Fs=1/Ts;  
n = [0:3*NF-1]';  
t = n*Ts;  
x = sqrt(1-t.^2).*rect(t/2) + ...  
    sqrt(1-(t-2).^2).*rect((t-2)/2) + ...  
    sqrt(1-(t-4).^2).*rect((t-4)/2) ;  
plot(t, x)
```



M.J. Roberts, Fundamentals of Signals and Systems

Using the DFT to approximate the CTFS (2)

```
x = sqrt(1-t.^2).*rect(t/2) + ...  
    sqrt(1-(t-2).^2).*rect((t-2)/2) + ...  
    sqrt(1-(t-4).^2).*rect((t-4)/2) ;
```

```
X = fft(x)/NF;
```

```
k = [0:NF/2-1];
```

M.J. Roberts, Fundamentals of Signals and Systems

Normalized ω_s and ω_0

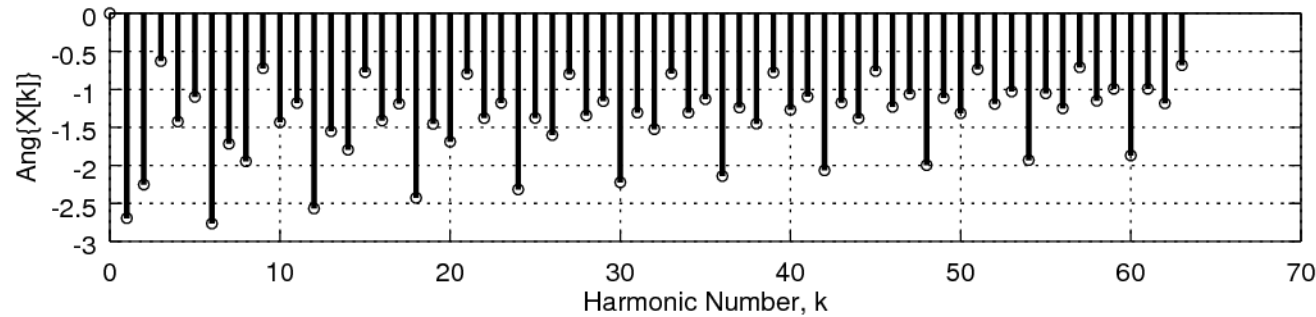
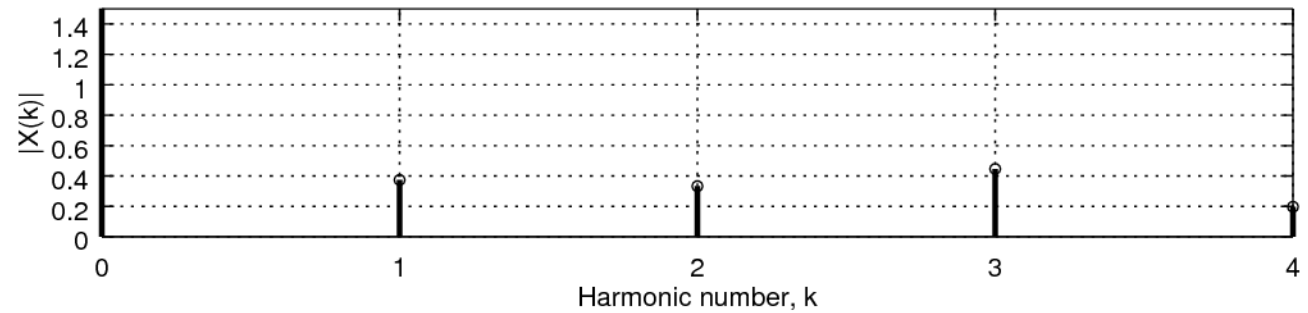
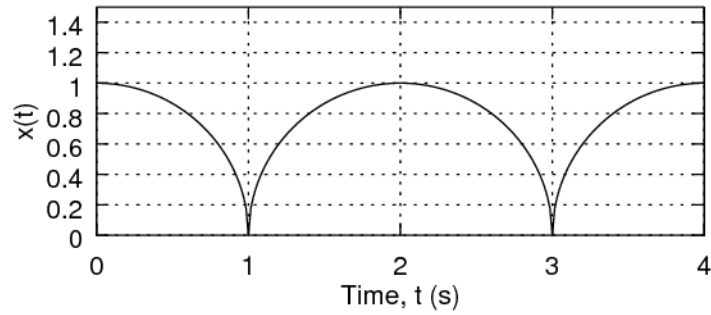
```
subplot(3,1,1);  
p = plot(t, x, 'k'); grid on;  
axis("equal"); axis([0,4,0,1.5]);  
xlabel('Time, t (s)');  
ylabel('x(t)');
```

```
subplot(3,1,2);  
p = stem(k, abs(X(1:NF/2)), 'k');  
set(p, 'LineWidth', 2, 'MarkerSize', 4); grid on;  
axis([0,4,0,1.5]);  
xlabel('Harmonic number, k');  
ylabel('|X(k)|');
```

```
subplot(3,1,3);  
p = stem(k, angle(X(1:NF/2)), 'k');  
set(p, 'LineWidth', 2, 'MarkerSize', 4); grid on;  
xlabel('Harmonic Number, k');  
ylabel('Ang{X[k]}');
```

M.J. Roberts, Fundamentals of Signals and Systems

Normalized ω_s and ω_0



of Signals and Systems

References

- [1] <http://en.wikipedia.org/>
- [2] J.H. McClellan, et al., Signal Processing First, Pearson Prentice Hall, 2003
- [3] M.J. Roberts, Fundamentals of Signals and Systems
- [4] S.J. Orfanidis, Introduction to Signal Processing
- [5] K. Shin, et al., Fundamentals of Signal Processing for Sound and Vibration Engineerings

- [6] A “graphical interpretation” of the DFT and FFT, by Steve Mann