Appendix B
MATLAB Code Listings

Below are MATLAB code listings for several functions used throughout the book. They are provided here so that the reader knows exactly how to generate samples of these signals.

Listing B.1 MATLAB code for evaluating the rect function.

```
function y = rect(x, D)
% function y = rect(x, D)
    if nargin == 1, D = 1; end
    x = abs(x);
    y = double(x<D/2);
    y(x == D/2) = 0.5;
```

Listing B.2 MATLAB code for evaluating the triangle function.

```
function y = tri(t)
% function y = tri(t)
    t = abs(t);
    y = zeros(size(t));
    idx = find(t < 1.0);
    y(idx) = 1.0 - t(idx);
```

Listing B.3 MATLAB code for evaluating the circ function.

```
function z = circ(x, y, D)
% function z = circ(x, y, D)
    r = sqrt(x.^2+y.^2);
    z = double(r<D/2);
    z(r==D/2) = 0.5;
```

Listing B.4 MATLAB code for evaluating the jinc function.

```
function y = jinc(x)
% function y = jinc(x)
    y = ones(size(x));
    idx = x ~= 0;
    y(idx) = 2.0*besselj(1, pi*x(idx)) ./ (pi*x(idx));
```

Listing B.5 MATLAB code for analytically evaluating the Fresnel diffraction pattern of a square aperture.

```
function U = fresnel_prop_square_ap(x2, y2, D1, wvl, Dz)
% function U = fresnel_prop_square_ap(x2, y2, D1, wvl, Dz)

    N_F = (D1/2)^2 / (wvl * Dz); % Fresnel number
    % substitutions
    bigX = x2 / sqrt(wvl*Dz);
    bigY = y2 / sqrt(wvl*Dz);
    alpha1 = -sqrt(2) * (sqrt(N_F) + bigX);
    alpha2 = sqrt(2) * (sqrt(N_F) - bigX);
    beta1 = -sqrt(2) * (sqrt(N_F) + bigY);
    beta2 = sqrt(2) * (sqrt(N_F) - bigY);
    % Fresnel sine and cosine integrals
    ca1 = mfun('FresnelC', alpha1);
    sa1 = mfun('FresnelS', alpha1);
    ca2 = mfun('FresnelC', alpha2);
    sa2 = mfun('FresnelS', alpha2);
    cb1 = mfun('FresnelC', beta1);
    sb1 = mfun('FresnelS', beta1);
    cb2 = mfun('FresnelC', beta2);
    sb2 = mfun('FresnelS', beta2);
    % observation-plane field
    U = 1 /(2*i) *((ca2 - ca1) + i * (sa2 - sa1)) ...
        .* ((cb2 - cb1) + i * (sb2 - sb1));
```