

# 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.

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

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

```
1 function y = tri(t)
2 % function y = tri(t)
3     t = abs(t);
4     y = zeros(size(t));
5     idx = find(t < 1.0);
6     y(idx) = 1.0 - t(idx);
```

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

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

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

```

1 function y = jinc(x)
2 % function y = jinc(x)
3     y = ones(size(x));
4     idx = x ~= 0;
5     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.

```

1 function U = fresnel_prop_square_ap(x2, y2, D1, wvl, Dz)
2 % function U = fresnel_prop_square_ap(x2, y2, D1, wvl, Dz)
3
4     N_F = (D1/2)^2 / (wvl * Dz); % Fresnel number
5     % substitutions
6     bigX = x2 / sqrt(wvl*Dz);
7     bigY = y2 / sqrt(wvl*Dz);
8     alpha1 = -sqrt(2) * (sqrt(N_F) + bigX);
9     alpha2 = sqrt(2) * (sqrt(N_F) - bigX);
10    beta1 = -sqrt(2) * (sqrt(N_F) + bigY);
11    beta2 = sqrt(2) * (sqrt(N_F) - bigY);
12    % Fresnel sine and cosine integrals
13    ca1 = mfun('FresnelC', alpha1);
14    sa1 = mfun('FresnelS', alpha1);
15    ca2 = mfun('FresnelC', alpha2);
16    sa2 = mfun('FresnelS', alpha2);
17    cb1 = mfun('FresnelC', beta1);
18    sb1 = mfun('FresnelS', beta1);
19    cb2 = mfun('FresnelC', beta2);
20    sb2 = mfun('FresnelS', beta2);
21    % observation-plane field
22    U = 1 / (2*i) * ((ca2 - ca1) + i * (sa2 - sa1)) ...
23        .* ((cb2 - cb1) + i * (sb2 - sb1));

```



Jason D. Schmidt,

*Numerical Simulation of Optical Wave Propagation with  
examples in MATLAB®*

SPIE Press, Bellingham, Washington (2010).

The supplemental files for this eBook are available for download by copying and pasting the following link to a browser:

[http://spie.org/Samples/Pressbook\\_Supplemental/PM199\\_sup.zip](http://spie.org/Samples/Pressbook_Supplemental/PM199_sup.zip)

A zip file containing the supplementary materials will be available from the link.