

PropCode1 Notes

Charles Rino

September 2010

1 Introduction

PropCode1 is a MATLAB implementation of the algorithm described in Chapter 2.2 of *The Theory of Scintillation with Applications in Remote Sensing* by Charles L. Rino, John Wiley & Sons IEEE Press, 2010. The algorithm simulates electromagnetic (EM) wave propagation in a two-dimensional medium. A two-dimensional medium has no variation outside a plane containing the EM source. Two-dimensional simulations are not computationally intensive and readily displayed. More important, they provide a usable approximations to point-to-point communication and remote sensing applications. Users may be familiar with standard propagation codes based on the parabolic wave equation, which is a special case of the forward propagation equation (FPE) described in Chapter 2 of the book cited above.

PropCode1 was developed as a tutorial introduction to the FPE and propagation phenomena that affect radio and optical propagation in structured media. Both the deterministic and random components of the structure are important, although scintillation refers to the random variation of field observables. Prior to **PropCode1** execution it is necessary to define the refractive index within the propagation space, the excitation field, the field sampling, and the incremental propagation steps. The setup utility, **SetupPropCode1**, defines these inputs and stores them in a **.mat** file accessed by a GUI when **PropCode1** is executed. Setup **.mat** file names contain the date and time of generation as a unique identifier. All subsequent code output file names include the setup date-time stamp. Display and analysis utilities use the same GUI to select appropriate input files.

To summarize the **PropCode1** execution sequence:

1. Run **SetPath4PropCode1** to put main code and support directories on the MATLAB path.
2. In the selected or copied example folder, run the **SetupPropCode1*** script to generate **.mat** files containing the variables required by the main program code (now on the MATLAB path). The names of the **.mat** files generated by the **Setup*** scripts contain the name **Setup** and a date-time stamp that uniquely identifies the setup **.mat** file.

3. In the selected or copied example folder, run `PropCode1`, which will initiate a GUI-driven file selection utility.¹ Selection of the appropriate `Setup*.mat` file will initiate execution of `PropCode1` using inputs from the selected `Setup*.mat` file. Screen output will indicate the code progress. The main code outputs include a `*.mat` file and binary data files. Each output file contains the setup time stamp, which facilitates management of the input and output associated with multiple runs in the same directory folder.
4. Run the display codes initiated in scripts with book Figure numbers to generate the book figures.

The codes have been designed for exploration, not general purpose execution. Parameter changes require editing the setup utilities, which are provided as templates. No checks are made for sample or data space-size adequacy.

2 Example Descriptions

Execution of the specific examples and the graphic outputs that will be generated are described below. First transfer the MATLAB active directory to

```
\PropCode1_Examples
```

and execute `SetPath4PropCode1` to place the `PropCode1` codes and utilities on the MATLAB path. Note that when keyboard input is requested a carriage return will select the appropriate input for the example.

2.1 Beam Propagation

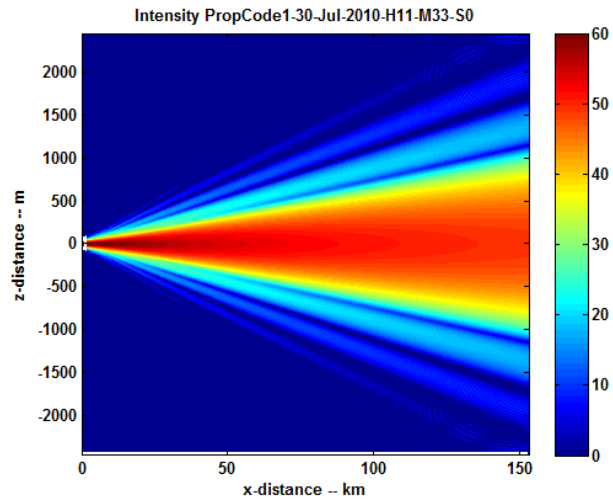
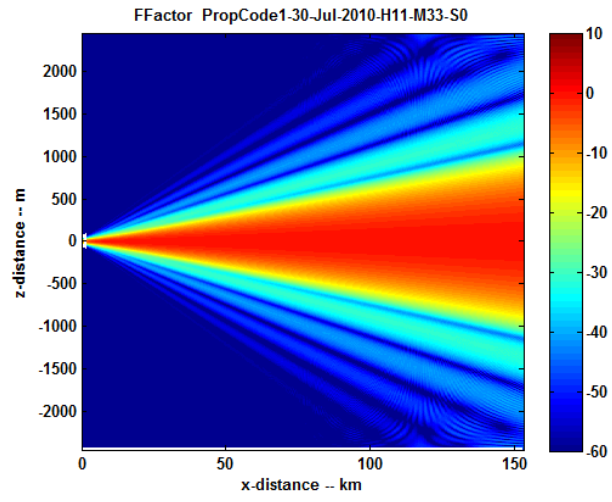
To generate the beam propagation example transfer the active MATLAB directory to

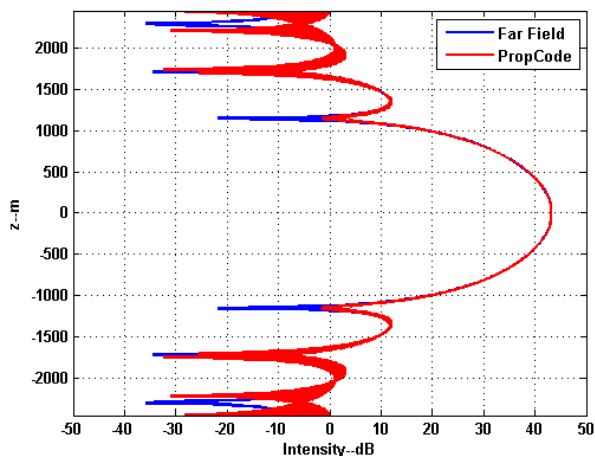
```
... \BeamPropagation.
```

Next run the script `SetupPropCodeFigBeamProp`. Next Run `PropCode1`. To display the output run `MakeFig21`. The output, Figure 2.1, is a color-coded intensity map of the output normalized to unity intensity (0 dB) on the beam axis (book Figure 2.1). Figure 2.1 shows the same display with no normalization. Because forward energy is conserved, the on-axis intensity must decrease to maintain constant total intensity. Note the much larger dynamic range needed to display the field in real intensity units.

Executing the script `MakeFig22` will produce figure 2.1 (book Figure 2.2), which is an overlay of the incident field at 153.2 km (red) on the exact far-field field computed with the DFT of the incident field (blue). As discussed in Chapter 2.2, the disparities are computation-grid edge effects. The central portion of the beam, which is usually the region of primary interest, is unaffected.

¹The GUI lists all the `*.mat` files **in its current directory**, which may not be the desired example directory. It is up to the user to select the appropriate `Setup*.mat` file generated in step 1.





2.2 Refraction

To execute the refraction example transfer the MATLAB active directory to

```
... \Refraction.
```

Next run the script `SetupPropCodeFigRefraction` to generate the input file for the first refraction example in Chapter 2.2. `PropCode1` has a provision for selecting precomputed layer profiles during the setup operation. For this example a layer of enhanced refractive index is placed just below the source. To display the profile, run the script `MakeFig23`, which will generate Figure 2.2 (book Figure 2.3).

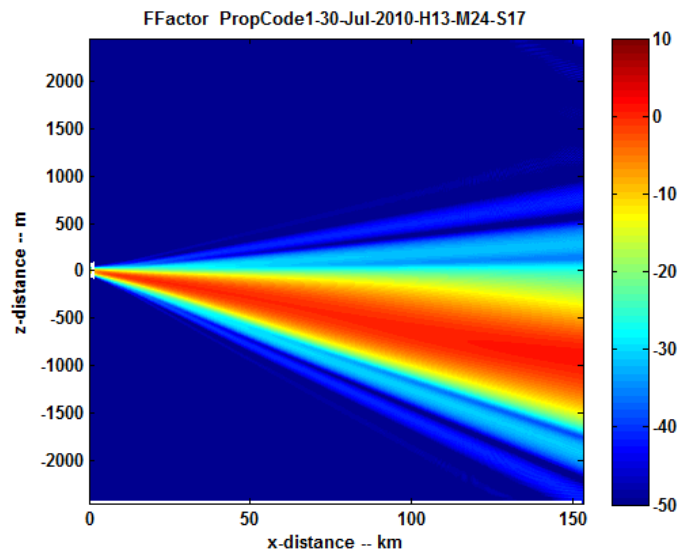
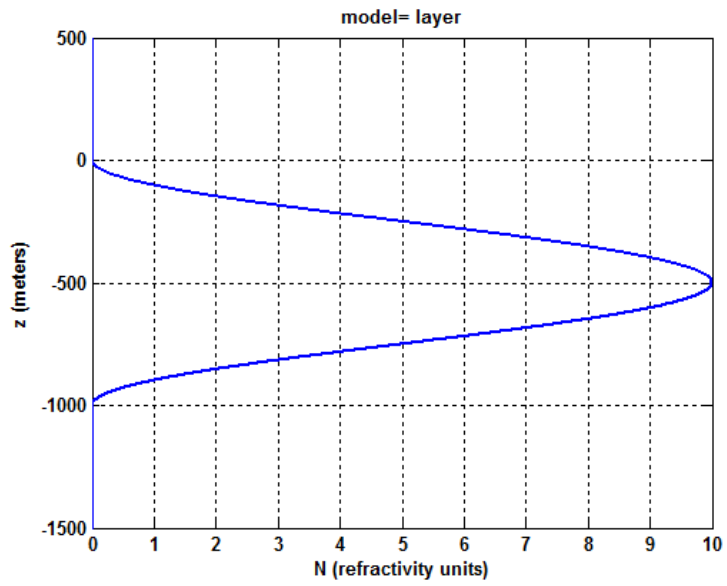
Now `PropCode1` as before to output the simulated field evolution. To display the output run the script `MakeFig24` will generate the normalized beam intensity profile shown in Figure 2.2 (book Figure 2.4). Running the script `MakeFig25` will generate the spatial Fourier transform of the field plotted against grazing angle (book Figure 2.5). These scripts set the input parameters for a standard `PropCode1` display utility.

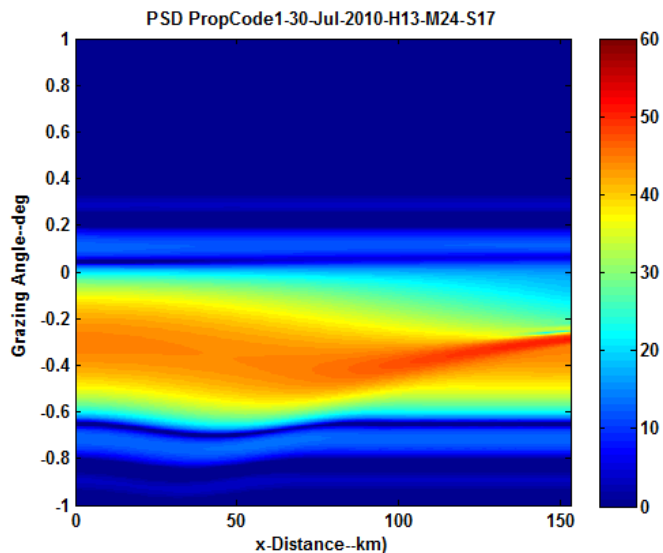
2.3 Strong Refraction

To execute the strong refraction example, transfer MATLAB active directory to the folder

```
... \StrongRefraction
```

and run the script `SetupPropCodeStrongRefraction` to generate the input file for the strong refraction example in Chapter 2.2. For this example the beam is launched along the layer axis and the refractivity has been increased to 40 refractivity units. Run `PropCode1` as before to generate the data files. Run





the script `MakeFig26` to generate Figure 2.3, (book Figure 2.6) which shows the refracted beam being redirected toward its own axis. The field concentrations act as local sources that repeat the strong focusing cycle. Run the script `MakeFig26a` to generate Figure 2.3, which shows the evolution of the spatial frequency content.

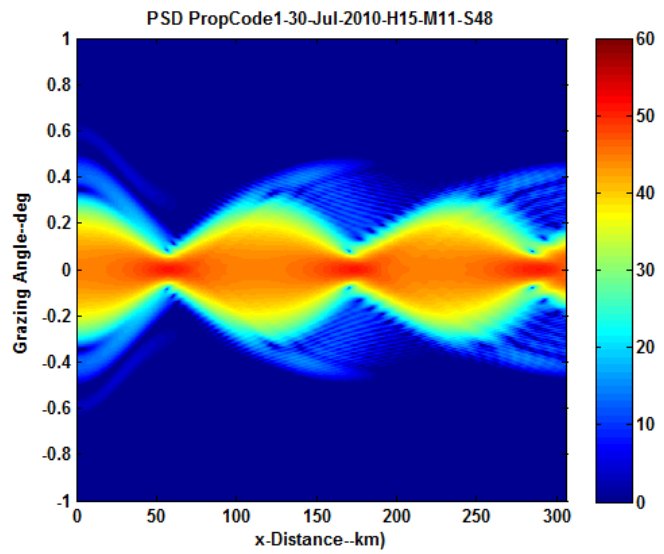
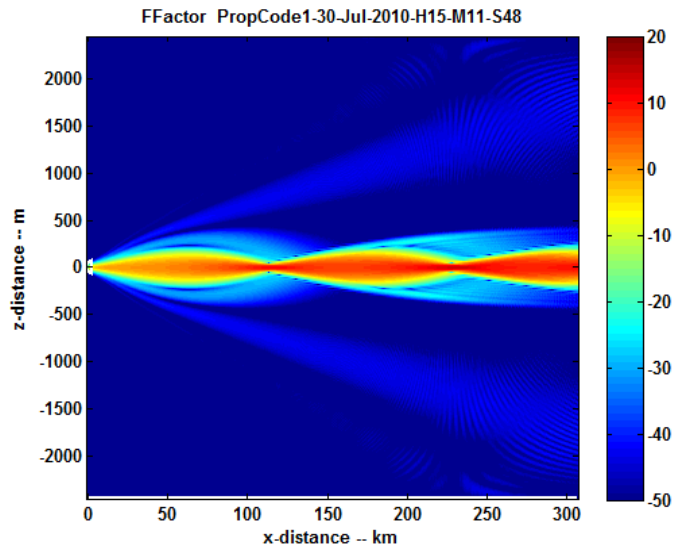
2.4 Scintillation

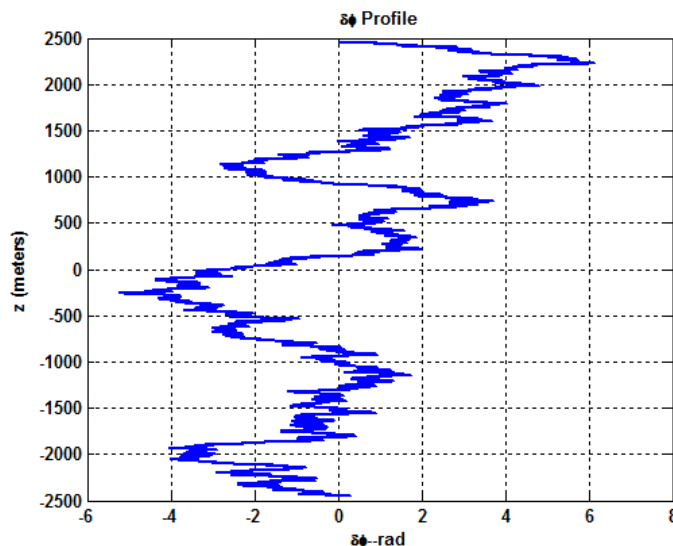
The `PropCode1` examples presented thus far have incorporated deterministic refractive index variations. The setup scripts in the folder

`... \Scintillation`

add a power-law random structure to the refractive index profile to simulate scintillation structure. Two examples are presented. The first example uses a structure confined to a single layer. Intensity structure develops as the field propagates away from the layer. The second example distributes the same average disturbance over twice the propagation distance from the single layer. The weak-scatter theory predicts statistically similar results when observed at a distance equal to the propagation distance from the center of the concentrated layer. Executing the scripts `SetupPropCode1Ex1` and `SetupPropCode1Ex2` will generate the appropriate `PropCode1` outputs as identified by time stamps. Executing the script `MakeFig29` will display the single-layer refractive index profile converted to phase units (Figure 2.4, book Figure 2.9).

Running the script `MakeFig210` will generate the unnormalized intensity versus distance display shown in Figure 2.4 (book Figure 2.10). Each slab that





contains structure is marked along the ordinate axis. Under ideal conditions a scintillation measurement would produce a scan of the field normal to the propagation direction. Close to an initiating screen the scintillation index is small.

Running the script `MakeFig11Fig12` will generate the plot of SI versus propagation distance (Figure 2.4, book Figure 2.11) and the intensity and phase at the maximum propagation distance. The phase extracted from the complex field is 2π ambiguous, but with adequate sampling the 2π jumps can be detected and removed. Figure 2.4 (book Figure 2.12) shows the intensity and phase obtained from the Example 1 realization. The fact that the phase contains a linear trend is a consequence propagating the incident field at a small angle .01 degrees from normal. As a practical matter, phase trends are invariably present because of small transmit and receive frequency offsets or gradients in the refractive index.

Executing the script `MakeFig13` will generate the intensity versus distance display for the Example 2 extended medium. As one expects the development of intensity in more gradual. Executing the script `MakeFig13Fig14` will generate the displays corresponding to Figures 2.4 and 2.4. The SI plot (Figure 2.4 book Figure 2.15) reinforces the more gradual development of intensity scintillation. The final intensity and phase structure shown in (Figure 2.4, book Figure 2.13) is nearly identical to the structure shown in Figure 2.4.

Executing the script `MakeFig16` will generate the evolution of the spatial wavenumber spectrum shown in Figure 2.4. The spectrum evolves continuously because there is no region of free-space propagation.

