Modeling the highly complex shapes of wind-blown, random sea surfaces is fundamental to a wide range of oceanographic problems, which include reflection and transmission of light, exchange of momentum and energy between winds and currents, loading of structures, and ship dynamics. Various mathematical techniques for modeling sea surfaces are therefore widely used in oceanography and ocean engineering. Unfortunately, sea surfaces arise from very complicated hydrodynamics, and it should be no surprise that the mathematics needed to describe them can also be rather complex. The following pages seek to explain a subset of those techniques, namely the use of Fourier transforms to go back and forth between surface realizations and wave variance spectra, at a level appropriate for students coming upon the material for the first time.

The material presented on the next 15 web pages is considered well known and can be found in abbreviated form in many texts and publications. However, "well known" often means that somewhere, sometime, someone figured something out, but the details were not given. The implication is that if you cannot figure out the details for yourself, then you are unworthy to benefit from The Master's Wisdom. It is true that figuring out this material from scratch is a worthwhile exercise that leads to true understanding. However, it is also very time consuming, and time for reinventing wheels is a luxury few scientists can afford. A lucid explanation can also lead to in-depth understanding, with time left over for application of the knowledge to solving new problems. The goal here is to explain the material at the level of detail needed to write computer programs and simulate sea surface optical properties. In particular, certain pesky special cases are considered, as is the often confusing issue of array indexing. The reader can judge whether or not these explanations are lucid.

The discussion begins with a physically and mathematically complicated but very important topic: how to mathematically describe wind-blown sea surfaces. There are various ways to do this. The first way discussed here is by use of wave variance spectra. The second way, discussed the later on, is via autocovariance functions. The two descriptions are equivalent, and it is possible to transform one description into the other. In any case, the eventual goal is twofold. The first goal is to show how to describe a time-dependent sea surface with waves of all temporal and spatial scales propagating in any direction. The second is to show how to generate time-dependent random realizations of such surfaces so that the generated surface is close to what would be observed in nature. Such generated surfaces can be used, for example, as the basis for Monte Carlo simulations of the optical reflection and transmission properties of sea surfaces, and even to simulate the visual appearance of water surfaces for use in computer-generated movie scenes.

Real sea surfaces exhibit extremely complicated shapes involving waves of many sizes traveling in all directions. Those waves arise from various nonlinear physical processes that transfer energy from the wind to the water, and between waves of different sizes. It should not come as a surprise that the mathematics needed to describe these processes and the resulting sea surfaces are also complex, both in the psychological sense and in the $\sqrt{-1}$ sense.

If you come to optical oceanography from a background in physics, electrical engineering, or numerical analysis, then you are probably already familiar with Fourier transforms, variance (or energy or power) spectra, sampling theory, and the other tools that will be needed. However, if you began life in some other field, then even the idea of complex numbers may strike fear in your nonmathematical heart. The discussion therefore begins slowly

and studies various idealized cases in detail before proceeding to real ocean surfaces. After reviewing the needed mathematics, the fundamental concepts of wave variance spectra are presented. The discussion then progresses from the simplest possible physical context of a snapshot (i.e., the surface at one instant of time) of a one-dimensional (1-D) sea surface, then a time-independent two-dimensional (2-D) surface, and finally reaches the goal of simulating time-dependent 2-D surfaces as are found in nature. Finally, the description of sea surfaces via autocovariance functions will be described, and it will be shown how to generate random sea surfaces starting with an autovariance function, rather than with a variance spectrum.

Much of this material comes from the tutorial Modeling Sea Surfaces, version 2.0.

# IDL Code for Surface Generation and Plotting

The IDL source codes that created some of the figures on the following Web Book pages and in the report *Modeling Sea Surfaces: A Tutorial on Fourier Transform Techniques* are being made publicly available for non-commercial scientific research. This is the same code package that was distributed to attendees at the short course on Sea Surface Modeling given at Ocean Optics XXIII in October 2016.

Selected figures on the following pages have notes in their captions such as "Generated by IDL routine cgGenerate_2D_SeaSurface.pro" to identify which IDL routine created that figure.

Running the IDL code requires a license for IDL. For IDL licensing information, see IDL.

This code was developed largely without funding and I (Curtis Mobley) regard it as intellectual property of considerable value. I therefore request appropriate acknowledgment for its use in any presentations, reports, publications, or other works that make use of my code or its output. Depending on the extent of use of the code or its outputs, suitable acknowledgment can range from a footnote to offer of coauthorship. For commercial use, contact curtis <dot> mobley <at> gmail <dot> com for licensing.

**Note:** My IDL code often uses extensions or wrappers to the standard IDL code, which were developed by David Fanning and distributed as his "Coyote Graphics System." The Coyote Graphics routines make it easier to do certain things like create vector eps files, manage color tables, create device-independent plots, etc. The Coyote Graphics routines replace the standard IDL calls, e.g., the IDL "Plot" command becomes "cgPlot".

The IDL directory on the download contains a subdirectory IDL/coyote, which contains the Coyote Graphics routines I use. There is possibly a more recent version available. You will need to add the ../coyote directory to the IDL search path, so that these routines will be found by IDL. See Coyote Graphics for further info.

Also see the text *Coyote's Guide to Traditional IDL Graphics* by David Fanning. This is the bible for IDL users.

**Final comment:** If you use my IDL routines to reproduce some of the figures on the following pages, you may get different realizations of the random sea surfaces. This will occur if you use a different seed for a random number generator, which gives a different sequence of random numbers, hence a different surface realization. However, even if you use exactly the same code and exactly the same seed, you may still get a different sequence of random numbers. In particular, 64 bit and 32 bit computers will generate different sequences of

pseudo random numbers for the same algorithm and seed because the algorithms are based on modular arithmetic, which gives different results for floating point variables that have different numbers of digits on 64 and 32 bit machines.

Download a zip file containing the IDL Surface Generation and Plotting Code