

Modeling Sea Surfaces

A Tutorial on Fourier Transform Techniques

Curtis D. Mobley

**Sequoia Scientific, Inc.
2700 Richards Road, Suite 107
Bellevue, WA 98005**

Version 2.0
October 2016

Abstract

The mathematics relating wind-blown sea surfaces and wave variance spectra is well known, but seldom well explained, and the devil is in the details. The goal of this tutorial is to fill in those details as needed for writing computer programs to carry out various calculations so that the results are both physically correct and computationally efficient. The tutorial first shows how measurements of wave elevation for a random sea surface lead to an elevation variance spectrum, which shows how much variance (or energy) is contained in waves of different spatial or temporal frequencies. It then shows how a variance spectrum can be used to generate random realizations of a sea surface. The basic concepts are introduced for idealized sea surfaces with just one spatial dimension. More realistic two-dimensional surfaces are then considered. After investigation of surfaces at a single instant of time, the generation of time-dependent surfaces is presented.

However, not all sea surface roughness is generated by wind. For example, turbulence induced by unstable shear flows is another process that can roughen a water surface. For such surfaces, there is no wind-dependent elevation variance spectrum, but the surface roughness properties can be described by surface elevation autocovariance functions. The Wiener-Khinchin theorem shows how surface autocovariance functions are related to surface elevation variance spectra. Using this theorem, an autocovariance can be converted to a spectral density, which can then be used for surface generation just as for wind-generated waves. The final chapter discusses how surfaces can be generated beginning with an autocovariance function.

Computational matters such as array indexing and ordering of frequency arrays are emphasized. Appendices give an overview of the computation of discrete Fourier transforms via the fast Fourier transform, and an overview of wind-dependent wave variance spectra.

Most of this material is available as web pages in the Surfaces, Level 2 section of the *Ocean Optics Web Book*, beginning at

http://www.oceanopticsbook.info/view/surfaces/level_2/modeling_sea_surfaces.

The IDL computer codes used to generate many of the figures in this report can be downloaded from the web book pages. A pdf of this report can be downloaded from the Mobley (2016) reference in the references section of the *Ocean Optics Web Book* at

<http://www.oceanopticsbook.info/view/references/publications>.

Acknowledgments

The first versions of the IDL computer codes used to generate the figures in this tutorial were developed in the course of work supported by NASA under contract NNH12CD06C to Curtis Mobley titled *Radiative Transfer Modeling for Improved Ocean Color Remote Sensing*. The writing of this tutorial was supported by my hard-working wife Ann Kruse and many unfunded weekends and evenings. Converting the hard copy tutorial to web pages for the *Ocean Optics Web Book* was supported by NASA under grant NNX14AP66G.

Permission to Reproduce

This report is Copyright © by Curtis D. Mobley, 2016. However, permission is hereby granted to reproduce this work all or in part, with appropriate acknowledgment, for non-commercial educational and research purposes.

Errata

Compared to Version 1 of this report (dated September 2014), several typographical errors have been corrected, additional explanations are given in various places, and the new Chapter 5 on autocovariance functions and Appendix A on calculation of discrete Fourier transforms have been added. Please send any comments, questions, or error identifications on the present version to curtis.mobley@sequoiasci.com. The version on the Ocean Optics Web Book will be kept up to date as material is added or errors are corrected. This version of the tutorial replaces the previous version.

Contents

List of Figures	iii
List of Tables	v
1 Mathematical Preliminaries	1
1.1 Sinusoidal Wave Representations	2
1.2 Continuous Fourier Transforms	5
1.3 Sampling	6
1.4 Discrete Fourier Transforms	8
1.4.1 Alternate Formulations	10
1.5 Parseval's Identity	12
2 One-Dimensional, Time-Independent Surfaces	13
2.1 Wave Energy	13
2.2 Wave Spectra	14
2.3 Surfaces to Variance Spectra	16
2.4 Frequency Ordering	20
2.5 Variance Spectra to Surfaces	22
2.5.1 Theory	23
2.5.2 Example: A Roundtrip Calculation	27
2.6 Numerical Resolution	29
3 Two-Dimensional, Time-Independent Surfaces	36
3.1 Surfaces to Variance Spectra	37
3.1.1 Example: A Random Sea Surface	37
3.1.2 Example: A Sea Surface of Crossing Sinusoids	39
3.2 Variance Spectra to Surfaces	42
3.2.1 Theory	42
3.2.2 Example: A Two-Dimensional Sea Surface	46
3.3 Effect of the Spreading Function	48
3.4 Resampling the DFT Grid for Ray Tracing	50
3.5 Example: Reflectance by Different Surface Models	52
3.6 Example: Tiling a Two-dimensional Surface	54

4 Two-Dimensional, Time-Dependent Surfaces	56
5 Autocovariance Functions and the Wiener-Khinchin Theorem	60
5.1 Autocovariance	61
5.2 The Wiener-Khinchin Theorem	62
5.3 The Horoshenkov Autocorrelation Function	63
5.3.1 The Horoshenkov Variance Spectrum	64
5.4 Numerical Illustration of the Wiener-Khinchin Theorem	66
5.5 Sampling Strategy and Computational Details	69
5.5.1 Idle Speculations	74
5.5.2 Lessons Learned	75
5.6 Turbulence-generated Water Surfaces	77
5.7 Two-dimensional Water Surfaces	78
A Computation of Discrete Fourier Transforms	82
A.1 Example Calculations	83
A.2 Timing	86
B Wave Spectra	90
B.1 Overview of Wave Spectra	90
B.2 The Pierson-Moskowitz Omnidirectional Gravity Wave Spectrum	95
B.3 The Elfouhaily et al. Directional Gravity-Capillary Wave Spectrum	98
B.3.1 Spreading Functions	101
References	103

List of Figures

1.1	Illustration of sampling the harmonics of a given wave and of aliasing	7
1.2	Illustration of Sampling a Sinusoid	11
2.1	A 1-D random sea surface and its Fourier transform and variance spectrum	18
2.2	A 1-D surface composed of cosine waves	21
2.3	A 1-D surface composed of sine waves	22
2.4	Example of a 1-D random sea surface generated from the Pierson-Moskowitz spectrum	28
2.5	Example sampling of elevation and slope spectra for $N = 1024$	32
2.6	Example sampling of elevation and slope spectra for $N = 65536$	32
2.7	Example 1-D surfaces and slopes created using true and adjusted variance spectra .	35
3.1	A 2-D random sea surface and its Fourier transform and two-sided variance spectrum	38
3.2	A 2-D sea surface composed of two crossing sinusoids	41
3.3	Example 2-D sea surface and underlying spectrum and Fourier amplitudes	47
3.4	Perspective view of a 2-D sea surface with $S = 2$	49
3.5	Perspective view of a 2-D sea surface with $S = 20$	50
3.6	Mapping a rectangular DFT grid to a hexagonal grid of triangles	51
3.7	A sea surface defined by a rectangular DFT grid and by the corresponding grid of triangular wave facets	53
3.8	Energy reflectance by different sea surface models	54
3.9	Example of tiling a 2-D surface	55
4.1	A time-dependent sequence of 2-D sea surfaces	58
5.1	Horoshenkov autocovariance and variance spectral density functions	65
5.2	Illustration of the Wiener-Khinchin theorem	67
5.3	Spatial correlation function of wind-generated waves	68
5.4	Same as Fig. 5.2, but for 100 surface realizations	70
5.5	Same as Fig. 5.2, but for 1000 surface realizations	71
5.6	Illustration of sampling strategy	72
5.7	Illustration of inadequate sampling	76
5.8	Example of a 1-D turbulence-generated water surface	79
5.9	Example of a 2-D turbulence-generated water surface	80
5.10	A slice through the surface of Fig. 5.9	80

5.11 A wind-generated surface for a 5 m s^{-1} wind speed	81
A.1 DFT for a 16-point surface	84
A.2 DFT for the first 14 points of the surface of Fig. A.1	85
A.3 FFT of the surface of Fig. A.2 padded with zeros to get 16 points	87
A.4 Time required to compute 1 DFT for N between 1000 and 4100.	88
A.5 Time required to compute 1000 DFTs using the IDL FFT routine for N between 1000 and 1200.	88
A.6 Time required to compute 1,000 DFTs using the IDL FFT routine for N between 1000 and 4100	89
B.1 The Pierson-Moskowitz variance spectrum as a function of wind speed, k and ω	97
B.2 The Pierson-Moskowitz slope spectrum as a function of wind speed and k	97
B.3 Omnidirectional ECKV elevation and slope spectra for fully developed seas as a function of wind speed	100
B.4 Omnidirectional ECKV elevation and slope spectra for young to fully developed seas for a wind speed of 10 m s^{-1}	101
B.5 Spreading functions for directional wave spectra	102

List of Tables

2.1	Spatial frequencies and wavelengths corresponding to peak variance for the Pierson-Moskowitz spectrum	30
3.1	Comparison of Cox-Munk mean square slopes and values for a DFT-generated 2-D surface	49
B.1	Comparison of variances computed two different ways. The values of $ \hat{z}(n) ^2$ are taken from the lower-right panel of Fig. 2.1.	92
B.2	Summary of spectral quantities.	95

CHAPTER 1

Mathematical Preliminaries

“I told her [his younger sister] to start at the beginning and read as far as you can get until you’re lost. Then start again at the beginning and keep working through until you can understand the whole book.” –Richard Feynman in *Richard Feynman: A Life in Science*

* * * * *

Wind-blown, random sea surfaces and the related concept of wave variance spectra are 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. This tutorial seeks 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 here 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 can’t figure out the details for yourself, then you are unworthy to benefit from The Master’s Wisdom. I do agree that figuring out this material from scratch, as I have had to do, is a worthwhile exercise that leads to true understanding. However, it is also very time consuming (more than three months of effort in my case for the material presented here), and time for reinventing wheels is a luxury few scientists can afford. I would argue that a lucid explanation can also lead to in-depth understanding, with time left over for application of the knowledge to solving new problems. My goal here is to explain the material at the level of detail needed to write computer programs. In particular, certain pesky special cases are considered, as is the often confusing issue of array indexing. The reader can judge whether or not my explanations are lucid.

The discussion begins with a physically and mathematically complicated but very important topic: how to describe and model wind-blown sea surfaces. The eventual goal is twofold. The first

goal is to show how to describe a time-dependent sea surface with waves of all scales propagating in any direction. The second is to show how to generate time-dependent random realizations of such surfaces so that the generated surfaces are 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 is 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 we will need. However, if you began life as a biological oceanographer, then even the idea of complex numbers may strike fear in your nonmathematical heart. We will therefore begin slowly and study various idealized cases in detail before proceeding to real ocean surfaces. We will progress 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 reach the goal of simulating time-dependent 2-D sea surfaces as are found in nature. The final chapter on the Wiener-Khinchin theorem shows how surfaces—even non-water surfaces—described by autocorrelation functions can be simulated.

1.1 Sinusoidal Wave Representations

We describe the sea surface using a Cartesian coordinate system. \mathbf{x} is a unit vector chosen to point in the downwind direction, \mathbf{z} is directed upward (from the water to the air) normal to the mean sea surface at $z = 0$, and $\mathbf{y} = \mathbf{z} \times \mathbf{x}$ is in the cross-wind direction. $z(x, y, t)$ is the surface elevation in meters at spatial location (x, y) and time t .

Jean-Baptiste Joseph Fourier (1768-1830) proved that, under very general mathematical conditions, an arbitrary function can be written as a sum of sines and cosines of different amplitudes and wavelengths. The sea surface elevation $z(x, y, t)$ is usually such a function. An exception is a sea surface with waves curling over and breaking on a beach. There can then be multiple air-water surfaces for a given (x, y, t) : the point on the surface that sees the full sky, the point above a surfer's head as the wave curls over her, and the point on the water that supports the surfboard. However, for a sea surface without breaking waves, there is only one air-water surface for a given location and time, which can be described as a sum of sinusoids.

We first define the terms and notation needed to describe propagating sinusoidal waves:

- A is the *amplitude* (in units of meters) of the wave. This is one-half the distance from the trough to the crest of a sinusoidal wave. Oceanographers often talk about the *wave height*, h , which is the distance from a wave trough to a wave crest. Wave height is what most people visualize when they talk about how “big” a wave is.

- T is the *temporal period* (seconds), usually called just the period. This is the time it takes for one wave crest to pass by a fixed point, i.e. for the argument of the sinusoid to go through 2π radians.
- $f = 1/T$ is the *temporal frequency* (1/seconds), usually called just the frequency. This is how many waves pass by a fixed point per second.
- $\omega = 2\pi/T = 2\pi f$ is the *angular frequency* (radians/second). This is the number of waves passing a fixed point in 2π seconds.
- Λ is the *wavelength*, or spatial period, (meters). This is the distance from one wave crest to the next.
- $\nu = 1/\Lambda$ is the *wavenumber* or *spatial frequency* (1/meters). This is the number of wavelengths per meter.
- $k = 2\pi/\Lambda = 2\pi\nu$ is the *angular wavenumber* or *angular spatial frequency* (radians/meter). This is the number of wavelengths in 2π meters.
- ϕ is the phase (nondimensional). This fixes the location of a wave crest relative to the origin of a coordinate system.

There is no uniformity in the literature. People often refer to “the frequency” without specifying whether they mean f or ω , and “wavenumber” or “spatial frequency” may mean either ν or k . People use $\nu, \tilde{\nu}$, or σ for the wavenumber, some use σ for ω , and so on. You just have to figure it out on a case by case basis. For pedagogic purposes, we will start with wavenumber ν as the measure of spatial frequency, and then switch to the more common angular spatial frequency k . We will use ω as the measure of temporal frequency.

Now consider a one-dimensional sea surface with elevation $z(x, t)$ over a spatial region of length L , e.g. $0 \leq x \leq L$ or $-L/2 \leq x \leq L/2$. We can write $z(x, t)$ as a sum of sinusoids, the n^{th} one of which is

$$z_n(x, t) = A_n \cos\left(\frac{2\pi nx}{L} + \phi_n \pm \frac{2\pi nt}{T}\right), \quad (1.1)$$

where $n = 0, 1, 2, \dots$ is simply an index for which sinusoid is being considered. (Note that we could just as well write a sine here.) For $n = 0$ this reduces to a constant offset $z_0(x, t) = A_0 \cos(\phi_0)$, which is usually taken to be mean sea level and set to a reference value of 0 via setting $A_0 = 0$. For the moment, we take $t = 0$, i.e., we have a snapshot of the ocean surface at time zero. This sinusoid has a wavelength of $\Lambda_n = L/n$, i.e., the cosine returns to the same value after a distance of $x = L/n$. It is a pure cosine in the chosen coordinate system if the phase is 0 or an even integer multiple of $\pm\pi$. If the phase is an odd integer multiple of $\pm\pi/2$ it is a sine. If time is included, the wave is propagating in the $+x$ direction if the time term is $-2\pi nt/T$; the cosine returns to its initial value after a time of $t = T/n$. A wave propagating in the $-x$ direction is described by a $+2\pi nt/T$ term. The physical angular frequency $\omega_n \equiv 2\pi n/T$ is always positive, but mathematically we can write just $+\omega_n t$ in the equation and then view a wave propagating in the $+x$ direction as having a negative temporal frequency.

With this interpretation of $\omega_n t$, Eq. 1.1 is conveniently written as

$$z_n(x, t) = A_n \cos(k_n x + \phi_n + \omega_n t), \quad (1.2)$$

where $k_n = 2\pi n/L = 2\pi/\Lambda_n$ is the angular wavenumber of the n^{th} wave of wavelength $\Lambda_n = L/n$. Likewise, $\omega_n = 2\pi n/T = 2\pi/T_n$ is the angular frequency of the n^{th} wave with period $T_n = T/n$. It may be intuitively easier to think of wavelengths per meter than of wavelengths per 2π meters, but the convenience of writing k rather than $2\pi/\Lambda$ or $2\pi\nu$ makes k rather than ν the spatial frequency variable used in most publications. A similar comment holds for ω vs. $2\pi/T = 2\pi f$, so ω rather than f is the common choice for the temporal frequency variable.

If we take $t = 0$, or just combine the time term with the phase, expanding the cosine in Eq. (1.2) gives an equivalent form

$$z_n(x) = a_n \cos(k_n x) + b_n \sin(k_n x), \quad (1.3)$$

where $a_n = A_n \cos \phi_n$ and $b_n = -A_n \sin \phi_n$, $n = 0, 1, 2, \dots$, with $b_0 \equiv 0$ for bookkeeping purposes. This equation can be written in terms of complex numbers:

$$z_n(x) = c_{+n} e^{ik_n x} + c_{-n} e^{-ik_n x}, \quad (1.4)$$

where $i = \sqrt{-1}$ and

$$c_{+n} = (a_n - ib_n)/2, \quad c_{-n} = (a_n + ib_n)/2, \quad \text{and} \quad c_0 = a_0/2. \quad (1.5)$$

Recalling that $e^{\pm i\theta} = \cos \theta \pm i \sin \theta$, we see that the imaginary parts of Eq. (1.4) add to zero, so that $z_n(x)$ is still a real variable even though it is written in complex form. Also note that $c_{+n}^* = c_{-n}$, where c^* denotes the complex conjugate. (Complex conjugation means replace i by $-i$ in all terms.) Although two complex numbers in general contain four independent real numbers, these c_{+n} and c_{-n} pairs contain only two independent numbers, a_n and b_n .

We thus have three ways to describe a sinusoidal wave: (1) The cosine of Eq. (1.2) defined by an amplitude and a phase; (2) the cosine and sine of Eq. (1.3) defined by two amplitudes, or (3) the complex exponentials of Eq. (1.4) defined by two complex numbers containing two independent real numbers. These equations all give the same $z_n(x)$, so we are free to choose whichever form is mathematically convenient for the problem at hand. For visualization of sea surfaces, the real forms (1.1) or (1.3) are convenient, but the calculations below are most conveniently carried out using complex exponentials.

Returning now to Fourier, linear wave theory says that we can write the sea surface elevation as a Fourier series

$$z(x) = \sum_{n=0}^{\infty} z_n(x) = \frac{a_0}{2} + \sum_{n=1}^{\infty} [a_n \cos(k_n x) + b_n \sin(k_n x)], \quad (1.6)$$

which in conjunction with Eq. (1.4) can be rewritten as

$$z(x) = \sum_{n=-\infty}^{\infty} c_n e^{ik_n x}. \quad (1.7)$$

(As previously noted, the a_0 or c_0 term is usually set to 0.) This equation is the mathematical heart of our subsequent descriptions sea surfaces.

Although the physics that leads to a given sea surface in general involves non-linear interactions between waves of different wavelengths and periods, we can write the shape of the resulting sea

surface as a linear sum of sinusoids of different frequencies. Owing to the orthogonality of sines and cosines for different n values as defined above, these wave components $z_n(x)$ are independent of each other. (Orthogonality means that $\int_0^{2\pi} \cos(mx) \cos(nx) dx = 0$ if $m \neq n$, etc.) Thus a description of a time-dependent surface based on an expansion like that of Eq. (1.6) (including the time-dependent terms as seen in Eq. (1.2)) cannot be used to predict the nonlinear evolution of a sea surface from a given initial condition. To do that, you must return to the world of hydrodynamics in all of its nonlinear glory. I lived there as a graduate student and, trust me, it isn't a pretty place. For our present purposes, we only want to model the shapes of sea surfaces, not predict their hydrodynamic development from an initial state, so linear wave representations are adequate.

1.2 Continuous Fourier Transforms

The Fourier transform is the fundamental mathematical tool for our subsequent study of sea surfaces. There are many texts on Fourier transforms. Bracewell (1986) is a standard reference, and excellent sets of lecture notes and videos of lectures can be found on the web. We will therefore state without proof the results needed for our problem.

Given a real function $z(x)$ of a continuous variable x , the Fourier transform $\hat{z}(\nu)$ of $z(x)$ is *defined* as

$$\hat{z}(\nu) \equiv \mathfrak{F}\{z(x)\} \equiv \int_{-\infty}^{\infty} z(x) e^{-i2\pi\nu x} dx. \quad (1.8)$$

The inverse Fourier transform is given by

$$z(x) = \mathfrak{F}^{-1}\{\hat{z}(\nu)\} \equiv \int_{-\infty}^{\infty} \hat{z}(\nu) e^{+i2\pi\nu x} d\nu. \quad (1.9)$$

Equations (1.8) and (1.9) are termed a Fourier transform pair. It can be shown that if we insert the $\hat{z}(\nu)$ integral of Eq. (1.8) into Eq. (1.9), we recover $z(x)$. This is known as Fourier's integral theorem. (The proof of this theorem requires a knowledge of the Dirac delta function and consideration of the conditions for convergence of the integrals. We will not worry about such matters in this tutorial.)

Understanding the units of a Fourier transform is important. In our case, $z(x)$ is sea surface elevation, and both z and x have units of meters. Equation (1.8) shows that $\hat{z}(\nu)$ thus has units of m^2 . The variance of z also has units of m^2 , which gives the first hint at a profound connection between the variance of a physical quantity and its Fourier transform. The units of m^2 in the Fourier transform also can be rewritten as $\text{m}/(1/\text{m})$, which is units of z divided by units of spatial frequency ν . The transform $\hat{z}(\nu)$ can thus be interpreted as showing “how much of z there is per unit frequency interval.” The inverse transform then has units of (z over frequency) times frequency, which returns the original units of z .

A Fourier transform is a *spectral density function*. A spectral density function is by definition a function whose integral over a given frequency interval gives the variance in the physical quantity contributed by the frequencies in the integration interval. Density functions are rather peculiar mathematical creatures compared to point functions, which simply give the value of something at a given value of the independent variable (e.g. the temperature as a function of location x and time t). The blackbody radiation function is another example of a spectral density function. The blackbody function shows how much energy is emitted (at a given temperature) per unit frequency

interval of the emitted electromagnetic radiation. The blackbody function is discussed on the *Ocean Optics Web Book* at

http://www.oceanopticsbook.info/view/light_and_radiometry/level_2/a_common_misconception. If you are not familiar with the distinction between point and density functions, especially regarding how to change variables in density functions, you may wish to take a look at that discussion.

The Fourier transfer definitions above with the 2π in the exponents are those of the “Stanford school” of Bracewell (1986) and Goodman (1996). You will see others in the literature. If we use $k = 2\pi\nu$ as the frequency variable, then Eqs. (1.8) and (1.9) become

$$\hat{z}(k) \equiv \mathfrak{F}\{z(x)\} \equiv \int_{-\infty}^{\infty} z(x)e^{-ikx}dx \quad (1.10)$$

and

$$z(x) = \mathfrak{F}^{-1}\{\hat{z}(k)\} \equiv \frac{1}{2\pi} \int_{-\infty}^{\infty} \hat{z}(k)e^{+ikx}dk. \quad (1.11)$$

This reappearance of the 2π in the second equation is required so that the inverse transform of the transform gets you back to where you started. In the $e^{\pm ikx}$ version, some people put the $1/2\pi$ in front of the other integral, and some put a $1/\sqrt{2\pi}$ in front of each integral. Some authors (e.g. Press et al., 1992) put the $+i$ in Eq. (1.8) and the $-i$ in Eq. (1.9). The choice of which sign to use on the i and where to put the 2π is almost a religion—most people stay with what they first learned, are convinced of the superiority of their definition, and are almost willing to die rather than change. Fortunately, it doesn’t matter which definitions you use, so long as you are consistent in how the transform pair is defined so that you get back to where you started if you inverse transform a transform, or vice versa.

In our work, $z(x)$ is the sea surface elevation, which is a real number. However, even though $z(x)$ is real, $\hat{z}(\nu)$ (or $\hat{z}(k)$) is complex. Expanding the complex exponential in Eq. (1.8) as the sum of a cosine and a sine, it is easy to see that $\hat{z}^*(\nu) = \hat{z}(-\nu)$. Such functions are called *Hermitian*. *A real function has a Hermitian Fourier transform. Conversely, if a function is Hermitian, it has a real inverse Fourier transform.* This will be an important constraint below when we wish to generate random realizations of a sea surface by computing the inverse Fourier transform of a complex function: we will have to conjure up a Hermitian function so that we end up with a real sea surface.

1.3 Sampling

We next consider the implications of *sampling* the sea surface at a finite number of discrete locations or times. It is always the case when we want to measure the sea surface elevation that we either make measurements at discrete locations $x_r, r = 0, 1, \dots, N - 1$ at a given time, or at discrete times $t_r, r = 0, 1, \dots, N - 1$ at a given location.

For a specific example, consider a region of sea surface $L = 10$ m long where we take $N = 8$ evenly spaced samples. This gives a measurement every $\Delta x = L/N = 0.125$ m at a given time. We are going to describe this surface as a sum of sinusoids. Figure 1.1 illustrates this, with the first few sinusoids shown as cosines. The cosines in the figure all have the same amplitude and each is offset vertically for ease of viewing. The dots show the sampled values for each cosine component.

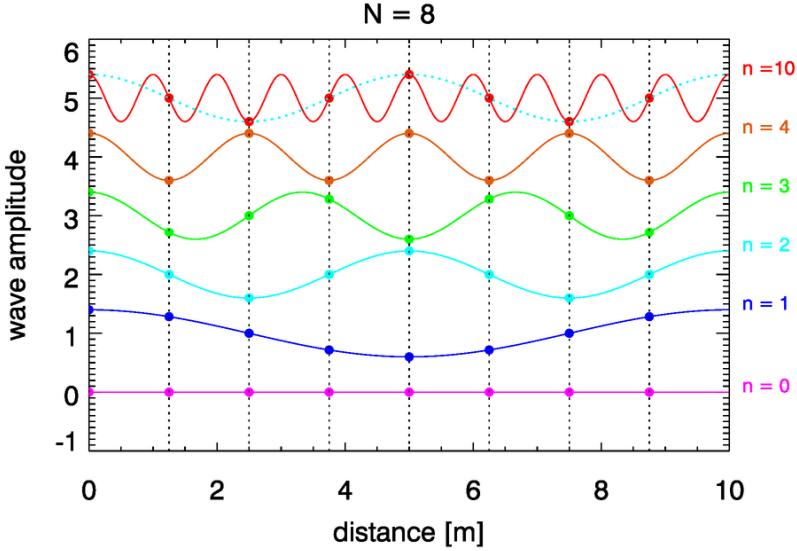


Figure 1.1: Illustration of sampling the harmonics of a given wave and of aliasing. The vertical dotted lines show the sampling locations, and the dots show the sampled values for the various cosine components.

The longest wave that can fit into a region of length L has a wavelength of $\Lambda = L$. This is called the fundamental wave or first harmonic. This wave is shown by the blue $n = 1$ curve in the figure. Note that the $n = 4$ cosine in the figure has a period of $2\Delta x$. The smallest wave than can be captured correctly in a sampling scheme has a period of twice the sampling interval. This called the two-point wave or two-point oscillation. Note that a sine term with period $2\Delta x$ would be sampled at arguments of $0, \pi, 2\pi, \dots$ where the sine is zero and is thus not detectable. The spatial frequency $1/(2\Delta x)$ (or temporal frequency $1/(2\Delta t)$ if sampling in time at a given location) is called the spatial (or temporal) *Nyquist frequency*. Waves with higher frequencies than the Nyquist frequency are still sampled, as illustrated by the dots on the wave with $n = 10$. Note, however, that the pattern of sampled points for the $n = 10$ wave is exactly the same as for the $n = 2$ wave. If all we have are the measured points, we cannot distinguish between the $n = 10$ and the $n = 2$ waves in this example. This illustrates the phenomenon of *aliasing*. In general, a wave with a frequency greater than the Nyquist frequency (i.e., a wavelength or period less that twice the sampling interval) is still sampled, but it appears as though the high frequency wave is a wave with a frequency less than the Nyquist frequency. The information about the high frequency (short wavelength) wave is added to or “aliased” into a lower frequency (longer wavelength) wave, thereby giving incorrect information about the lower frequency wave. Aliasing places a severe constraint on any sampling scheme. We can correctly sample only waves with wavelengths that lie between the size of the spatial region, L , and twice the size of the sampling interval. (In the temporal setting, the limits are the length of sampling time and twice the temporal sampling interval.)

The relation between a sampled frequency f_s greater than the Nyquist frequency, the sampling rate or Nyquist frequency f_{Ny} , and the frequency f_a receiving the aliased signal is

$$f_a = |mf_{Ny} - f_s|$$

where m is the closest integer multiple of the sampling rate to the signal being sampled. In the example of Fig. 1.1, $f_s = f_{n=10} = 10$ waves per 10 meters = 1 m^{-1} , $f_{\text{Ny}} = 0.4$, and either $m = 2$ or $m = 3$ gives $f_a = |2 \times 0.4 - 1| = 0.2$ (or $f_a = |3 \times 0.4 - 1| = 0.2$). Thus the $n = 10$ wave is aliased into the $n = 2$ wave, just as seen in the figure. Note that many other frequencies are also aliased into this frequency. For example, a wave with $f_s = 1.4$, corresponding to $n = 14$, also gives $f_a = 0.2$ with $m = 4$, and so on.

1.4 Discrete Fourier Transforms

Now suppose that we have sampled the sea surface $z(x)$ at a set of N evenly spaced points $x_r, r = 0, 1, \dots, N - 1$, in a region of size L ; $x_r = r\Delta x = rL/N$. We want to describe this sampled sea surface $z(x_r)$ as a sum of sinusoids. In general, these N values can be represented as a sum of a constant term, $N/2$ cosine terms, and $N/2 - 1$ sine terms (recall that there is no two-point sine term):

$$z(x_r) = \frac{a_0}{2} + \sum_{u=1}^{N/2} [a_u \cos(k_u x_r) + b_u \sin(k_u x_r)] , \quad (1.12)$$

with $b_{N/2} \equiv 0$. Note that the sum runs from the fundamental frequency ($u = 1, k_1 = 2\pi/L$) through the Nyquist frequency ($u = N/2, k_{N/2} = 2\pi/(2\Delta x)$), with only a cosine term for the two-point wave. This sum is equivalent to

$$z(x_r) = \sum_{u=-N/2+1}^{N/2} c_u e^{ik_u x_r} , \quad (1.13)$$

which also contains a total of N independent real and imaginary parts of the c_n coefficients. (Recall from Eq. (1.5) that $c_{-k} = c_k^*$, so these coefficients are not independent for k and $-k$ pairs.) These equations are the discrete-variable forms of Eqs. (1.6) and (1.7).

We now have a finite number N of discrete samples $z(x_r)$ of the sea surface, so we need a discrete form of the Fourier transform. The *discrete Fourier transform* (DFT) of $z(x_r)$ is defined as (Bracewell, 1986, page 358)

$$\hat{z}(\nu_u) \equiv \mathfrak{D}\{z(x_r)\} \equiv \frac{1}{N} \sum_{r=0}^{N-1} z(x_r) e^{-i2\pi\nu_u x_r} \quad \text{for } u = 0, \dots, N - 1 .$$

Recalling that $\nu_u = u/L = u/(N\Delta x)$ and $x_r = r\Delta x = rL/N$ gives $\nu_u x_r = ru/N$. It is also common to write $z(x_r) = z(r)$ and $\hat{z}(\nu_u) = \hat{z}(u)$, in which case the previous equation becomes

$$\hat{z}(u) = \mathfrak{D}\{z(r)\} = \frac{1}{N} \sum_{r=0}^{N-1} z(r) e^{-i2\pi ru/N} . \quad (1.14)$$

The corresponding inverse discrete Fourier transform is given by

$$z(r) = \mathfrak{D}^{-1}\{\hat{z}(u)\} = \sum_{u=0}^{N-1} \hat{z}(u) e^{+i2\pi ru/N} \quad \text{for } r = 0, \dots, N - 1 . \quad (1.15)$$

It is usually obvious from the context whether a continuous or a discrete Fourier transform is being used. However, for enhanced clarity, I'll use \mathfrak{F} for the continuous transform and \mathfrak{D} for the

discrete, since there are important differences. Likewise, if necessary, a subscript can be appended to show the frequency variable, e.g. $\mathfrak{F}_\nu\{z(x)\}$ as in Eq. (1.8) or $\mathfrak{F}_k\{z(x)\}$ as in Eq. (1.10).

The sums in the last two equations require computing complex exponentials (i.e., sines and cosines), multiplying by the corresponding values of $z(r)$ or $\hat{z}(u)$ and adding up the results. These equations can be evaluated for any value of N . The number of computations required to do this is of order N^2 . The computation time for a DFT thus increases very rapidly for large N .

However, a classic paper by Cooley and Tukey (1965) showed how these sums can be computed in order $N \log_2 N$ computations, if N is a power of 2. Their technique is now called the *fast Fourier transform* or FFT. The difference in computer time becomes enormous for large N . For example, if $N = 2^{12} = 4096$, then $N \log_2 N = 4096 \times 12$, and the difference in computation time is a factor of $N^2/(N \log_2 N) \approx 341$. Thus in the case of $N = 4096$, a roughly 6 minute computer run becomes a 1 second run. The development (or, perhaps, reinvention, since the basic idea goes all the way back to Gauss) of the FFT was a major advance in numerical analysis, which enables the computations on the following pages to be performed extremely efficiently. Because of the ubiquity of Fourier transforms in science, engineering, medicine, and economics, the Cooley-Tukey FFT algorithm has been called “the most important numerical algorithm of our lifetime” (Strang, 1994), and it is rated in the top ten most important numerical algorithms ever developed (Dongarra and Sullivan, 2000). Subroutines for computing FFTs and inverse FFTs are available in all computer languages commonly used in science (Fortran, C++, Python, IDL, Matlab, etc). Appendix A gives examples of DFT calculations using, and not using, the FFT and discusses the calculations when N is not a power of 2. Fortunately we do not need to concern ourselves here with the details of how the FFT algorithm actually works, any more than we need to worry about how a canned subroutine actually computes the cosine of a number. If you are interested in how the FFT works, a web search will yield many detailed explanations.

When generating sea surfaces, N can usually be chosen to be a power of 2 so that the FFT algorithm can be used to evaluate the DFT. It is thus expected that, in practice, the forward and inverse DFTs to be seen throughout the rest of this tutorial will always be evaluated via FFTs. I will thus use “DFT” for general discussions, and I’ll use “FFT” when I’m referring to actual calculations. Keep in mind that the DFT is completely general (any value of N), and that the FFT is an algorithm for efficient computation of the DFT when N is a power of 2.

The one-dimensional (1-D) equations seen above are easily extended to two or more dimensions. For two dimensions (x, y) , we can sample a region of size L_x by L_y meters over N_x points in the x direction and N_y points in the y direction, with N_x and N_y both powers of 2 so we can use FFTs. Equations (1.12) and (1.13) then become

$$\begin{aligned} z(x_r, y_s) &= \frac{a_0}{2} + \sum_{u=1}^{N_x/2} \sum_{v=1}^{N_y/2} [a_{u,v} \cos(k_u x_r + k_v y_s) + b_{u,v} \sin(k_u x_r + k_v y_s)] \\ &= \sum_{u=-N_x/2+1}^{N_x/2} \sum_{v=-N_y/2+1}^{N_y/2} c_{u,v} e^{i(k_u x_r + k_v y_s)}. \end{aligned} \quad (1.16)$$

The corresponding 2-D DFT pair is

$$\hat{z}(u, v) = \mathfrak{D}\{z(r, s)\} = \frac{1}{N_x N_y} \sum_{r=0}^{N_x-1} \sum_{s=0}^{N_y-1} z(r, s) e^{-i2\pi(ru/N_x + sv/N_y)} \quad (1.17)$$

and

$$z(r, s) = \mathfrak{D}^{-1}\{\hat{z}(u, v)\} = \sum_{u=0}^{N_x-1} \sum_{v=0}^{N_y-1} \hat{z}(u, v) e^{+i2\pi(ru/N_x + sv/N_y)}. \quad (1.18)$$

We emphasized above that the continuous function $\hat{z}(\nu)$ defined by the continuous Fourier transform (1.8) is a density function with units of $z(x)$ per spatial frequency interval, e.g., m/(1/m) if z is sea surface height in meters. However, the discrete functions $\hat{z}(u)$ and $\hat{z}(u, v)$ defined by the discrete Fourier transforms (1.14) and (1.17) have the same units as $z(r)$ and $z(r, s)$. The discrete Fourier transform is a point function that shows how much of $z(r)$ is contained in a *finite* frequency interval $\Delta\nu = 1/L$ centered at frequency $\nu_u = u/L$. Discrete Fourier transforms therefore convert point functions $z(r)$ and $z(r, s)$ to point functions $\hat{z}(u)$ and $\hat{z}(u, v)$.

It will be important below to keep notational track of continuous vs. discrete versions of various functions. Thus $\mathcal{S}(k)$ will denote a continuous function of k , $\mathcal{S}(k = k_u)$ will denote the continuous function evaluated at the discrete value k_u , and $\mathcal{S}(k_u) = \mathcal{S}(u)$ will denote a discrete point function. This notation is not standard, but it is used here because it is necessary to keep in mind that $\mathcal{S}(k = k_u)$ and $\mathcal{S}(k_u)$ have different units.

The differences in units between continuous and discrete Fourier amplitudes sometimes makes it tricky to make the transition between discrete and continuous versions of the same quantity. In particular, it will be necessary to explicitly include the frequency intervals in some of the later calculations that involve both continuous and discrete variables. For example, if we have a continuous density function and we need to convert to a corresponding discrete function at frequency ν_u , we must multiply the continuous function by the finite frequency interval, e.g.

$$\hat{z}(u) = \hat{z}(\nu = \nu_u) \Delta\nu. \quad (1.19)$$

Conversely, if we have discrete amplitudes $\hat{z}(u)$ and we wish to estimate the continuous spectral density at ν_u , then we must divide by the frequency interval:

$$\hat{z}(\nu = \nu_u) = \hat{z}(u) / \Delta\nu. \quad (1.20)$$

(If you are an optical oceanographer familiar with the scattering phase function, you can find an analogous situation in the estimation of the scattering phase function from measurements of scattered light. The scattering phase function is a measure of how much light energy is scattered from an incident direction into a particular direction, per unit solid angle; it therefore has units of 1/steradian. If you measure the scattered light using an instrument with a finite solid angle $\Delta\Omega$, then you get the total amount of energy scattered into the solid angle $\Delta\Omega$. To estimate the phase function from this measurement, you must divide the measured value by the solid angle of the instrument; this gets you back to units of 1/steradian.)

1.4.1 Alternate Formulations

As always, there are competing definitions. Equations (1.14) and (1.15) are used in Bracewell and the IDL computer language. *Numerical Recipes* (Press et al., 1992) interchanges the i and $-i$. *Numerical Recipes* and Matlab put the $1/N$ factor on the inverse transform. Matlab does not support array indices of 0, so the summation indices are shifted from 0 to $N - 1$ to 1 to N , with a corresponding shift from ru to $(r - 1)(u - 1)$ in the exponentials. As I said at the start, the devil

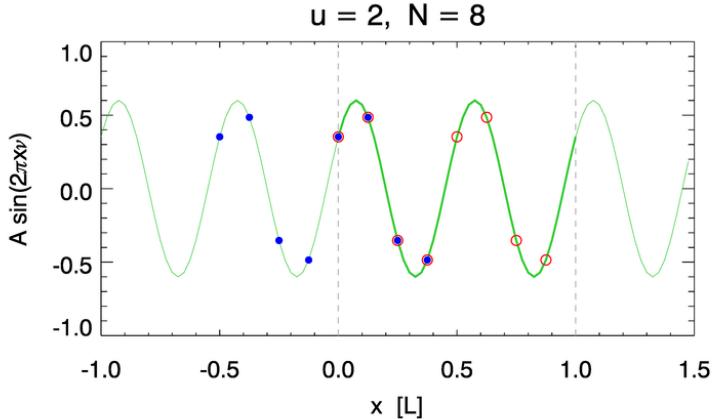


Figure 1.2: Illustration of sampling a sinusoid for different ranges of indices. The gray dashed lines show the $0 \leq x < L$ range.

is in the details, and details like where to put the $1/N$ factor and differences in array indexing in different computer languages can cause great misery when it comes time to actually write computer programs or compare results computed by different canned subroutines.

It is common in mathematics to use the letters i, j, k for dummy summation indices. However, I've already used i for $\sqrt{-1}$ and k for angular wavenumber, so the preceding equations would be hopelessly confusing if I reused i and k for indices. I will therefore use r and s for indices on spatial variables, e.g., (x_r, y_s) , and u and v for indices on frequency variables, e.g., k_u or ν_v . n and m also will be used as needed for dummy indices.

Various authors show different ranges for the summation indices in the equations of the DFT. I have chosen to let my spatial region be $0 \leq x < L$, in which case it is natural to write Eq. (1.14) with the sum running from $r = 0$ to $r = N - 1$. The thin green curve in Fig. 1.2 shows a sinusoid of amplitude $A = 0.6$ and frequency $\nu = 2/L$ (i.e, ν_u with $u = 2$), which represents either of the the sines or cosines represented by the complex exponential in the continuous Fourier transform of Eq. (1.8). The open red circles show this sinusoid sampled at $N = 8$ values $x_r = 0, \dots, (N - 1)\Delta x$, with $\Delta x = L/N$. However, you often see the DFT written with the summation indices running from $r = -N/2 + 1, \dots, N/2$ or $r = -N/2, \dots, N/2 - 1$. For example, Bracewell (1986, page 362) also writes the DFT pair as

$$\begin{aligned}\hat{z}(u) &= \frac{1}{N} \sum_{r=-N/2}^{N/2-1} z(r) e^{-i2\pi r u / N} \\ z(r) &= \sum_{u=-N/2}^{N/2-1} \hat{z}(u) e^{+i2\pi r u / N}.\end{aligned}$$

These alternate index ranges correspond to the exact same summation value. The blue dots in Fig. 1.2 show the sinusoid sampled at $N = 8$ values from $x = -L/2$ to $x = L/2 - \Delta x$. Because of the periodicity of the sinusoid, the summation of the blue dots contains the same values as for the red circles, just in a different order.

The ranges of summation indices can be confusing when comparing different texts or computer programs. Computer programs may let r and u run from 0 to $N - 1$ or from 1 to N , depending on whether the computer language allows arrays to have 0 (or negative) indices. Thus a computer program may have an N -dimensional array of frequencies labeled with indices from 0 to $N - 1$, but the frequency values in the array may run from $(-N/2 + 1)\Delta k$ to $(N/2)\Delta k$, and furthermore the order of the frequency values in the array may be different depending on the use of the array (see §2.4). It's a confusing mess, but you eventually get used to it.

1.5 Parseval's Identity

The physical and spectral variables of a continuous Fourier transform pair satisfy

$$\int_{-\infty}^{\infty} |z(x)|^2 dx = \int_{-\infty}^{\infty} |\hat{z}(\nu)|^2 d\nu , \quad (1.21)$$

which is known as Parseval's identity. The corresponding equation for the discrete Fourier transform pair defined by Eqs. (1.14) and (1.15) is

$$\sum_{r=0}^{N-1} |z(r)|^2 = N \sum_{u=0}^{N-1} |\hat{z}(u)|^2 . \quad (1.22)$$

For complex amplitudes, $|\hat{z}|^2 = \hat{z}\hat{z}^*$. The extension to the two-dimensional case is straightforward:

$$\sum_{r=0}^{N_x-1} \sum_{s=0}^{N_y-1} |z(r, s)|^2 = N_x N_y \sum_{u=0}^{N_x-1} \sum_{v=0}^{N_y-1} |\hat{z}(u, v)|^2 . \quad (1.23)$$

The discrete forms of Parseval's relations provide important checks on numerical calculations. For example, it is easy to misplace factors of N , which appear in different places depending on the exact form used for the definition of the discrete transforms. As just discussed, the essence of Parseval's relations is that the sums of the squares of the physical and spectral values is the same; the ranges of indexing labels (shown here as 0 to $N - 1$) are not important (except, perhaps, when debugging a computer program).

You can take a class in Fourier transforms and prove many more pretty theorems about their properties. Mathematicians get paid to worry about the conditions for the existence of a Fourier transform. For example, the function $z(x) = \sin(1/x)$, $0 < x \leq 1$ has an infinite number of extrema as $x \rightarrow 0$, and consequently doesn't have a Fourier transform. However, real ocean surfaces never behave like this, so we need not trouble ourselves with such matters. We have now assembled the mathematical tools needed to describe wind-blown sea surfaces, and so we return to oceanography.

CHAPTER 2

One-Dimensional, Time-Independent Surfaces

We now begin our study of sea surfaces and their Fourier transforms. For simplicity, we will introduce the basic ideas in a one-dimensional (1-D) geometry. This corresponds to making measurements of the sea surface along a line of points at a given time, which yields a set of surface elevations $z(x_r) = z(r)$, $r = 0, 1, \dots, N - 1$. The Fourier transform of $z(x_r)$ will be in terms of the spatial frequency ν_u or the angular spatial frequency k_u . Conversely, we could make measurements at a single spatial point over a set of times. This yields a set of elevations $z(t_r)$. The frequency variable for the Fourier transform of $z(t_r)$ would be either the temporal frequency f_u or angular temporal frequency ω_u . The math is the same in either case. Our primary interest is generating sea surfaces at a given time, so the discussion (until Chapter 4) will be in terms of spatial sampling at a given time, which will be taken as $t = 0$ and not shown.

2.1 Wave Energy

Consider a sinusoidal wave of amplitude A and wavenumber ν . It can be shown (e.g. Preisendorfer, 1976, Vol. VI, page 72) that the total energy per unit horizontal area of sea surface of this wave, averaged over a wavelength, is

$$\frac{\text{energy}}{\text{area}} = \frac{1}{2}\rho g A^2 + \frac{1}{4}\tau\nu^2 A^2,$$

where

$\rho \approx 10^3 \text{ kg m}^{-3}$ is the density of water,

$g \approx 9.8 \text{ m s}^{-2}$ is the acceleration of gravity,

$\tau \approx 0.072 \text{ N m}^{-1}$ (at 25 deg C) is the surface tension of water.

The $\frac{1}{2}\rho g A^2$ term is the sum of the kinetic and gravitational potential energy of the wave. (The potential energy is relative to the mean sea surface at $z = 0$.) The $\frac{1}{4}\tau\nu^2 A^2$ term is the energy required to stretch the level surface into a sinusoid, working against surface tension. It is easy to see that the units of these terms are J m^{-2} .

The variance of a sinusoidal surface is the average over one wavelength Λ of the surface elevation squared (assuming that the mean surface is at zero):

$$\text{var}\{z\} \equiv \frac{1}{\Lambda} \int_0^\Lambda \left[A \sin\left(\frac{2\pi x}{\Lambda}\right) \right]^2 dx = \frac{1}{2} A^2. \quad (2.1)$$

Thus the energy of a sinusoidal wave also can be written in terms of its variance:

$$\frac{\text{energy}}{\text{area}} = \left(\rho g + \frac{1}{2} \tau \nu^2 \right) \text{var}\{z\}. \quad (2.2)$$

2.2 Wave Spectra

Equation (2.1) shows that the variance of the surface elevation is proportional to the amplitude squared. Equation (2.2) shows that the energy per unit horizontal area of a sinusoidal wave of a given amplitude and spatial frequency is also proportional to the amplitude squared, or to the variance of the surface elevation. For a sea surface containing waves of many different amplitudes and spatial frequencies, the total variance of the sea surface is the sum of the variances of the individual waves (the variance of a sum of independent random variables is the sum of the variances of the individual variables). Likewise, the total energy of the waves is the sum of the energies of the individual waves.

We thus have the following line of reasoning:

- For a discrete sample $z(r)$ of a zero-mean sea surface, the variance is

$$\text{var}\{z\} = \frac{1}{N} \sum_{r=0}^{N-1} |z(r)|^2.$$

- Parseval's identity (1.22),

$$\sum_{r=0}^{N-1} |z(r)|^2 = N \sum_{u=0}^{N-1} |\hat{z}(u)|^2,$$

- gives

$$\text{var}\{z\} = \sum_{u=0}^{N-1} |\hat{z}(u)|^2.$$

- We therefore identify $S(u) \equiv |\hat{z}(u)|^2$ as the *discrete variance spectrum*, with units of m^2 .

Such reasoning led Schuster (1898) to the seminal observation that *Fourier transforms can be used to decompose the total variance contained in a signal into the variance contained in each frequency*. Schuster's original interest was in searching for what he called “hidden periodicities” in weather phenomena (as opposed to obvious periodicities such as daily or seasonal cycles). He soon applied the technique to looking for periodicities in earthquakes, sunspots, and other phenomena. Today, the computational speed of the FFT allows his method of analysis to be widely applied in science, engineering, economics and other fields where it is desired to know the energy or power of a signal as a function of spatial or temporal frequency. Regardless of what physical quantity is

under consideration, the essence of a spectrum is that it gives the distribution of variance in that quantity as a function of frequency.

In this tutorial we are concerned with generating sea surfaces consistent with a chosen wave spectrum. For that application, it is the roughness of the surface—that is, the variance of the surface elevation at different spatial scales—that is of interest, not the energy contained in the waves. It is thus the variance spectrum that is of interest. If we wanted to model the forces on an oil derrick or ship, then we would be interested in the energy of the waves and would work with the energy spectrum.

In our context, the discrete variance spectrum $|\hat{z}(u)|^2, u = 0, 1, \dots, N - 1$ is the variance in sea surface elevations. It common to call $|\hat{z}(u)|^2$ the discrete energy or power spectrum. (Publications often make a cryptic comment about $|\hat{z}(u)|^2$ giving energy “when suitably normalized,” but they never seem to show the variance-to-energy conversion factor $(\rho g + \frac{1}{2}\tau\nu^2)$.) This may permissible if it is relative values that are of interest, e.g. this frequency has ten times more energy than that frequency. In those cases, it is only the shape of the spectrum that is important. However, we are here interested in the variance of the sea surface, so I will correctly refer to $|\hat{z}(u)|^2$ as the discrete variance spectrum, whose units are m^2 (the units of both $|z(r)|^2$ and $|\hat{z}(u)|^2$). If you really do want to get the energy per unit area of the sea surface contained in the frequency interval $\Delta\nu$ at ν_u , multiply $|\hat{z}(u)|^2$ by $\rho g \approx 10^4 \text{ kg m}^{-2} \text{ s}^{-2}$ and you’ll have Joules per square meter. (The surface-tension term contributes a negligible part of the total variance compared to the larger gravity waves. Surface tension is significant only for a very young sea surface when only capillary waves are present.) Some writers (e.g. Holthuijsen, 2007) take care to distinguish between the *variance* spectrum $|\hat{z}(u)|^2$, which describes the *statistical* properties of the waves, and the *energy* spectrum $\rho g |\hat{z}(u)|^2$, which describes a *physical* property of the waves.

As forewarned in §1.4, the transition from discrete to continuous spectra requires care. (To complicate matters, the literature is rich with different terms and symbols used for the same quantities, and the same symbol used for different quantities.) I’ll use $\mathcal{S}(u)$ to denote the discrete variance spectrum $|\hat{z}(u)|^2, u = 0, 1, \dots, N - 1$. $\mathcal{S}(u)$ shows how much variance is contained in a *finite* frequency interval $\Delta\nu$ centered at frequency ν_u . The continuous version of this discrete point function is the *variance spectral density* $\mathcal{S}(\nu)$, which shows how much variance there is *per unit frequency interval* at frequency ν . Recalling Eq. (1.20), the conceptual relation between the discrete and continuous versions of \mathcal{S} can be written

$$\mathcal{S}(\nu) = \lim_{\Delta\nu \rightarrow 0} \frac{\mathcal{S}(u)}{\Delta\nu}. \quad (2.3)$$

In this limit process, we are thinking of taking a finer and finer resolution of the sea surface elevation $z(x_r), r = 0, 1, \dots, N - 1$ so that $N \rightarrow \infty$ for a fixed physical domain of length L . The discrete $z(x_r)$ then approaches a continuous function $z(x)$. The discrete Fourier transform of $z(x_r)$ then gives more and more frequencies ν_u with smaller and smaller frequency intervals $\Delta\nu$. An important consequence of Eq. (2.3) that the continuous variance spectral density $\mathcal{S}(\nu)$ has units of $\text{m}^2/\text{frequency}$, which in the present example is $\text{m}^2/(1/\text{m})$, whereas the discrete variance spectrum $\mathcal{S}(u)$ has units of m^2 . $\mathcal{S}(\nu)$ is called the *omnidirectional variance spectrum*. It is derived in a slightly different way in Appendix B.1, which discusses continuous wave spectra in both one and two dimensions. Those results will be cited as needed. If you are new to wave spectra, take a look at that discussion before proceeding.

Plots of the Fourier amplitude squared are also often called “power spectra” because in many fields it is power rather than energy that is of interest. An example is the power being dissipated in an electrical circuit. In that case, the power is proportional to the time-dependent electrical current squared; current replaces surface elevation in our equations and the relevant frequency is the temporal frequency. It is now common that the Fourier amplitude squared of anything is called a power spectrum, regardless of what physical variable is under consideration.

As we know from Eq. (1.11), the 1-D surface elevation $z(x)$ is related to the Fourier amplitude $\hat{z}(k)$ by

$$z(x) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \hat{z}(k) e^{ikx} dk.$$

Differentiating this equation with respect to x gives the 1-D *slope* of the sea surface as

$$\sigma(x) \equiv \frac{dz(x)}{dx} = \frac{1}{2\pi} \int_{-\infty}^{\infty} \hat{z}(k) ike^{ikx} dk.$$

This leads us to identify $ik\hat{z}(k)$ as the Fourier amplitude corresponding to the sea surface slope. This gives us two ways to study the slope statistics of random sea surfaces, given the Fourier amplitude $\hat{z}(k)$ (which we will learn to create from variance spectra in the following sections). The first way is to take the inverse Fourier transform of $\hat{z}(k)$ to obtain $z(x)$, and then to differentiate $z(x)$ to get the slope. The second way is to take the inverse transform of $ik\hat{z}(k)$ to get the slope directly, without ever creating the surface $z(x)$ itself.

It is shown in §B.1 that the variance of the sea surface slope, σ^2 , usually called the *mean square slope* or mss, is given by

$$\sigma^2 = mss = \int_0^{\infty} k^2 \mathcal{S}(k) dk,$$

where $\mathcal{S}(k)$ is the omnidirectional variance spectrum introduced above. As we saw, $\mathcal{S}(k)$ has units of $\text{m}^2/(\text{rad}/\text{m})$, so the integral above gives the mss in units of rad^2 . This is nondimensional, but the label of “rad” reminds us that we can think of the slope as an angle (from the horizontal) measured in radians. The product $k^2 \mathcal{S}(k)$ is called the *slope spectrum*.

The extensions of these spectra to two dimensions are made in §B.1, which summarizes a number of useful relations for one- and two-dimensional variance and slope spectra as needed for this tutorial.

2.3 Surfaces to Variance Spectra

We begin with a contrived example. We will comment in detail on this simple example in order to reveal the mathematical subtleties and physical characteristics of Fourier transforms and wave variance spectra derived from surface elevations. A thorough understanding of this example takes us much of the way to understanding the case of a real ocean surface.

An ad hoc wave profile is constructed using the formula

$$z(x_r) = z(r) = \sum_{j=0}^{N/2} A(j) \cos[2\pi j x_r \nu_f + \phi(j)] \quad r = 0, \dots, N-1. \quad (2.4)$$

The x_r locations are given by $r\Delta x = rL/N$, where L is the length of the sea surface region being sampled and N is the number of samples. $\nu_f = 1/L$ is the fundamental spatial frequency, that is,

the spatial frequency or wavenumber of the wave with a wavelength of L . The amplitude of the wave at the j^{th} frequency, $j = 1, 2, \dots, N/2$, is chosen to be

$$A(j) = 0.1 \exp(-3j/N),$$

and $A(0) = 0$. The phase of the j^{th} wave component is randomly distributed over $[0, 2\pi)$ using

$$\phi(j) = 2\pi\mathcal{U}$$

where \mathcal{U} is a uniform $[0, 1)$ random number. A different random number is drawn for each j value.

The upper left panel of Fig. 2.1 shows the surface generated in this manner for $L = 10$ m, $N = 16$, and a particular set of random phases. Note that N is a power of 2 as will be needed for the FFT. The thin colored lines show the $N/2 + 1 = 9$ waves for each of the frequencies. The blue line is the wave for the fundamental frequency $\nu_f = 1/L = 0.1$ m $^{-1}$; the thin black line is the two-point wave at the Nyquist frequency $\nu_{\text{Ny}} = 1/(2\Delta x) = 0.8$ m $^{-1}$; the purple line is the constant $j = 0$ wave, which is set to $z = 0$ for the mean sea surface. The black dots connected by the thick black line show the sum of the individual waves. These points represent a discrete sampling of the continuous sea surface elevation.

In this example, the sampled region of the sea surface is $L = 10$ m in length, but the $N = 16$ samples do not include the point at $x = 10$ m. This is because the surface elevation at $x = L$ is always the same as at $x = 0$ when using Fourier techniques. Resolving the surface as a sum of sinusoids that are harmonics of the fundamental frequency $\nu_f = 1/L$ gives sinusoids that always return to their initial value after distance L . Real sea surfaces are of course not periodic, but we do not know the true value at L because it was not measured by the present sampling scheme. (Likewise, we do not know the true surface elevations at points in between the sampled locations.) *When we use Fourier techniques to generate random surface realization, we are always generating a sea surface that is a periodic tiling;* the tile dimension is L . This periodicity is useful if we want to generate a visual rendering of a large region of sea surface from a smaller computed region; the edges of the small tiles will match perfectly and the larger surface will often look reasonable, if you don't look too closely. An example of a tiled two-dimension surface is given in §3.6.

We now take the sequence of the $N = 16$ real wave elevations $z(r)$ seen in Fig. 2.1 and feed them into an FFT routine. We soon get back 16 complex numbers, the $\hat{z}(\nu_u) = \hat{z}(u)$ Fourier amplitudes, at a set of 16 corresponding frequencies ν_u . The upper right panel of Fig. 2.1 plots the real part of the $\hat{z}(u)$ complex numbers, and the lower-left panel plots the imaginary part.

We note first that the FFT routine returned both negative and positive spatial frequencies: $\nu = (-\frac{N}{2} + 1)\Delta\nu = -0.7$ m $^{-1}$, ..., $-0.1, 0, 0.1, \dots, \frac{N}{2}\Delta\nu = 0.8$ m $^{-1}$, for a total of $N = 16$ discrete spatial frequencies. Note that the frequency spacing $\Delta\nu$ equals the fundamental frequency $\nu_f = 1/L$. The spatial frequency ν_u tells how many waves fit into 1 m; this is clearly a positive number with an easily understood physical meaning. Therefore the negative frequencies returned by the FFT may seem somewhat mysterious. However, these negative frequencies are simply the mathematical price we pay for the convenience of using complex numbers. Consider, for example, the representation of the cosine as a sum of complex exponentials for the u^{th} frequency:

$$\begin{aligned} \cos(2\pi\nu_u x_r) &= \frac{1}{2} (e^{i2\pi\nu_u x_r} + e^{-i2\pi\nu_u x_r}) \\ &= \frac{1}{2} (e^{i2\pi(+\nu_u)x_r} + e^{i2\pi(-\nu_u)x_r}). \end{aligned}$$

$$L = 10 \text{ m}; N = 16; \nu_f = 0.1 \text{ m}^{-1}; \nu_{Ny} = 0.8 \text{ m}^{-1}; \text{random phases}$$

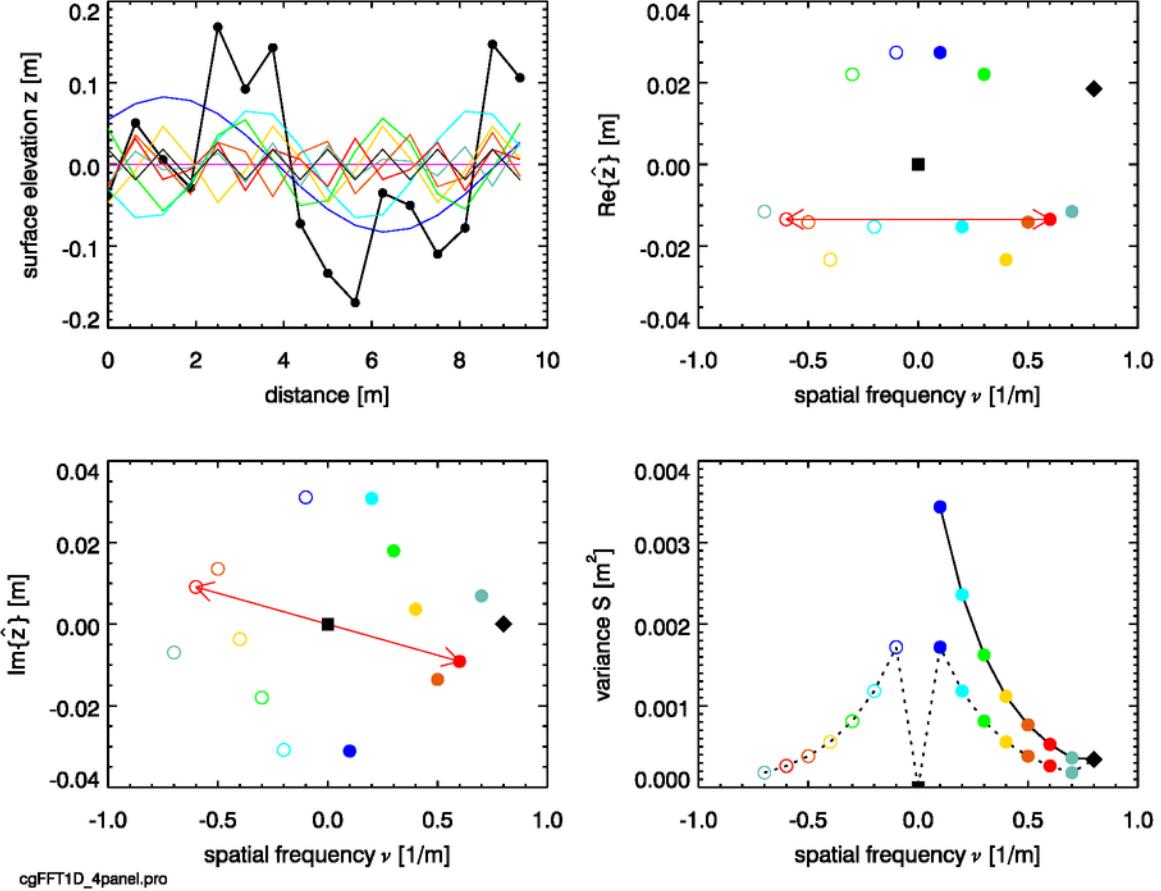


Figure 2.1: A one-dimensional random sea surface and its Fourier transform and variance spectrum. The black dots in the upper-left panel show the points $z(u)$ of the sampled sea surface. The light colored lines show the sinusoidal components used to create the surface. The upper-right panel shows the real part of the complex amplitude $\hat{z}(\nu_u)$ and the lower-left panel shows the imaginary part. The lower right panel shows the two-sided and one-sided discrete variance spectra. Compare with Figs. 2.2 and 2.3.

We can interpret the complex representation of the real cosine as having one term with a positive frequency $+\nu_u$ and one term with a negative frequency $-\nu_u$. A similar equation holds for the complex representation of $\sin(2\pi\nu_u x_r)$. The Fourier transform of a real function always contains both negative and positive frequencies, which arise from the complex exponential in the definition of the transform.

Note next that the real parts of the complex amplitudes $\hat{z}(u)$ are even functions of frequency: $\text{Re}\{\hat{z}(-\nu_u)\} = \text{Re}\{\hat{z}(+\nu_u)\}$. The imaginary parts are odd functions of frequency: $\text{Im}\{\hat{z}(-\nu_u)\} = -\text{Im}\{\hat{z}(+\nu_u)\}$. In Fig. 2.1 the positive and negative frequencies are connected by a red arrow for one particular frequency pair, $\nu_u = \pm 0.6 \text{ m}^{-1}$. If a complex function $c(\nu) = a(\nu) + ib(\nu)$ has an even real part $a(\nu)$ and an odd imaginary part $b(\nu)$, then $c^*(-\nu) = c(\nu)$, i.e. the function is

Hermitian. Thus the plots verify that the amplitudes are Hermitian, as is always the case for the Fourier transform of a real function. The Hermitian character of the complex amplitudes means that these $N = 16$ complex numbers contain only 16 independent real and imaginary numbers, not 32 as would be the case for 16 arbitrary complex numbers. In general the FFT of N real numbers (e.g., N spatial samples of a sea surface) gives back N independent numbers, so that the “information content” of the physical and Fourier representations is the same.

The positive frequency at $\frac{N}{2}\Delta\nu = 0.8 \text{ m}^{-1}$ is the Nyquist frequency. There is, however, no value for the negative of the Nyquist frequency. Note also in the lower left panel that the imaginary part of the amplitude is identically zero at the Nyquist frequency. We will explain these values below.

The lower-right panel of the figure shows the values of $|\hat{z}(\nu_u)|^2$. The values at the negative to positive frequencies are connected by the black dotted line. These points constitute the *two-sided discrete variance spectrum*,

$$\mathcal{S}_{2S}(\nu_u) = |\hat{z}(\nu_u)|^2 \quad \text{for } u = -\frac{N}{2} + 1, \dots, \frac{N}{2}. \quad (2.5)$$

“Two-sided”, denoted by the subscript 2S, refers to spectra showing both the negative and positive frequencies. The variance at zero frequency is the variance contained in the constant mean sea level. This value is zero because we have set the mean sea level to zero.

We are usually concerned only with the variance at a given magnitude of the spatial frequency, and not with whether the frequency is negative or positive. Nor is there any reason to plot the point at zero frequency, which is usually zero by choice of zero for the mean sea level. It is therefore customary to define the *one-sided variance spectrum*

$$\mathcal{S}_{1s}(\nu_u) = \mathcal{S}_{2s}(-\nu_u) + \mathcal{S}_{2s}(\nu_u), \quad (2.6)$$

for $u = 1, 2, \dots, \frac{N}{2} - 1$, and $\mathcal{S}_{1s}(\nu_{Ny}) = \mathcal{S}_{2s}(\nu_{Ny})$. Then only the positive frequencies are plotted. The points connected by the solid black line in the lower-right panel of Fig. 2.1 comprise the one-sided variance spectrum. In the present simulation, the two-sided spectrum is symmetric for positive and negative frequencies (except for the Nyquist frequency, which does not have a negative counterpart and is always a special case), and the one-sided function is simply twice the value of the two-sided function for the positive frequencies, except for the Nyquist frequency. When you read a paper and it refers to or plots “the variance (or energy) spectrum” without further comment, it is always the one-sided spectrum. We will have to account for the magnitude difference in one- and two-sided spectra in the next section.

There is an important detail to note in the computation and plotting of $\mathcal{S}(u)$, as in the lower-right panel of Fig. 2.1. The values of $\mathcal{S}(u)$ were obtained by the discrete Fourier transform of Eq. (1.14), and $\mathcal{S}(u)$ gives the *variance contained in a finite frequency interval $\Delta\nu = 1/L$ at the discrete frequency ν_u* . $\Delta\nu$ equals the fundamental frequency and is the frequency interval used in the calculations and the plot. As noted in the discussion of the discrete transform, $\mathcal{S}(u)$ is a point function. As already seen in Eqs. (1.20) and (2.3), if we wish to convert the discrete $\mathcal{S}(u)$ to an estimate of the continuous variance spectral *density* $\mathcal{S}(\nu)$, we must divide by the discrete function by the frequency interval: $\mathcal{S}(\nu) = \mathcal{S}(u)/\Delta\nu$. The units of $\mathcal{S}(\nu)$ are then $m^2/(1/m)$, as expected for a spectral density function of spatial frequency. I will strive to distinguish between a discrete variance point function and a continuous variance spectral density.

Let us return to Eq. (2.4) and set all of the phases $\phi(j)$ to zero. We are then adding together cosines to create the surface wave profile, which is seen in the upper left panel of Fig. 2.2. The FFT of this profile gives the real part of $\hat{z}(\nu_u)$ as positive numbers except for the 0 frequency, and the imaginary part is identically zero for all frequencies.

If we set all of the phases $\phi(j)$ to $\pi/2$, we are then adding together sines to create the surface wave profile, which is seen in the upper left panel of Fig. 2.3. The FFT of this profile gives the real part of $\hat{z}(\nu_u)$ identically zero and the imaginary part is nonzero except for the 0 and Nyquist frequencies.

These two figures show that the real part of the complex amplitude $\hat{z}(\nu_u)$ tells us how much of $z(x_r)$ is composed of cosine waves, and the imaginary part shows how much of $z(x_r)$ is composed of sine waves. This explains why the imaginary part of the amplitude is zero at the Nyquist frequency. The two-point wave at the Nyquist frequency is inherently a cosine wave because, as noted previously, a two-point sine wave is sampled only at its zero values. The general case of a wave component with a phase that is neither 0 (nor a multiple of 2π) nor $\pi/2$ (nor an odd integer multiple of $\pi/2$) can be written as a sum of cosine and sine waves, as in Eq. (1.3). In that case, both the real and imaginary parts of the amplitudes are nonzero (except for the special cases of the 0 and Nyquist frequencies).

Note that in each of these three simulations, which differ only by the phases of the component sinusoidal waves, the variance spectrum is exactly the same (except at the Nyquist frequency), as seen in the lower right panels of Figs. 2.1-2.3. That is to say, the variance contained in a wave does not depend on the reference coordinate system used to describe it, even though the Fourier amplitudes $\hat{z}(u)$ do depend on the coordinate system. The variance depends only on the amplitude of the wave. The variance at the Nyquist frequency is largest when cosine waves are added and is zero when sine waves are added. In the first case, we have the maximum possible amplitude of the two-point cosine wave, and in the latter case there is no two-point wave.

2.4 Frequency Ordering

It is necessary to mention a peculiarity of all FFT subroutines. The plots above show the frequencies ordered from negative to positive, as is convenient for plotting of two-sided spectra. Let Δ_f represent the fundamental frequency. If wavenumber ν is the frequency variable, then $\Delta_f = 1/L$. If angular spatial frequency k is the frequency variable, then $\Delta_f = 2\pi/L$; for temporal angular frequency ω , $\Delta_f = 2\pi/T$. In any case, the frequencies are equally spaced at intervals of Δ_f and are in the order

$$\left\{-\frac{N}{2} + 1, -\frac{N}{2} + 2, \dots, -1, 0, 1, \dots, \frac{N}{2} - 1, \frac{N}{2}\right\} \Delta_f. \quad (2.7)$$

I'll call this the "math frequency order." With the frequencies in math order, it is convenient to let the frequency index u run from $-\frac{N}{2} + 1$ to $\frac{N}{2}$, for a total of N frequencies.

However, FFT routines generally return their frequencies in the order of 0 first, then the positive frequencies, then the negative frequencies in reverse order:

$$\left\{0, 1, \dots, \frac{N}{2} - 1, \frac{N}{2}, -\frac{N}{2} + 1, -\frac{N}{2} + 2, \dots, -1, \right\} \Delta_f. \quad (2.8)$$

$$L = 10 \text{ m}; N = 16; v_f = 0.1 \text{ m}^{-1}; v_{Ny} = 0.8 \text{ m}^{-1}; \text{phases} = 0$$

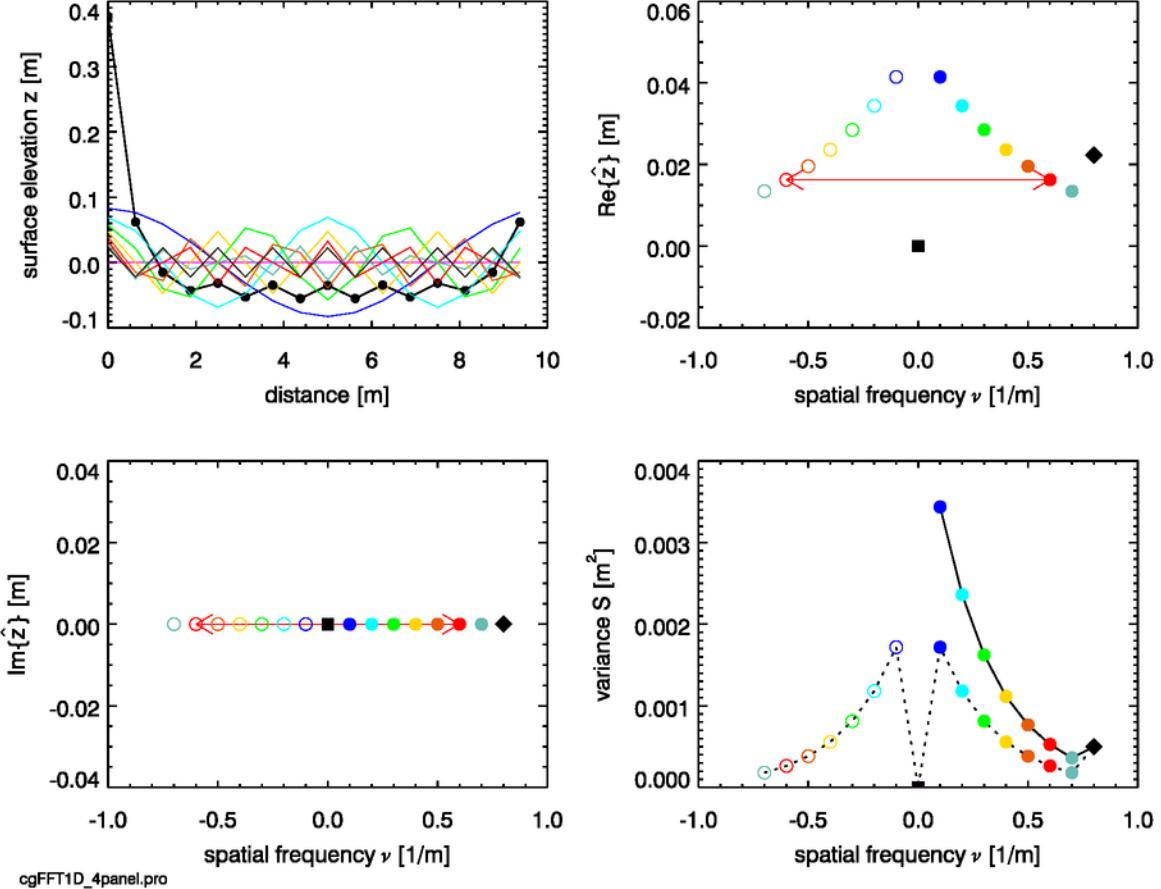


Figure 2.2: A one-dimensional surface composed of cosine waves. Compare with Figs. 2.1 and 2.3.

I'll call this the "FFT frequency order." Given the Hermitian symmetry of the amplitudes about the 0 frequency, the FFT order is convenient for creating and plotting a one-sided variance spectrum, in which case the last half of the amplitude array is ignored. With the frequencies in FFT order, it is convenient to let the frequency index u run from 0 to $N - 1$, for a total of N frequencies. This what most computer programs do when then return an array of N amplitudes corresponding to frequencies in the FFT frequency order.

Either frequency order can be obtained from the other by a repeated *circular shift*, which moves an array element off of one end of an array and copies it to the other end of the array, shifting all elements right or left by one position in the process. The detail to watch is that different computer languages define a circular shift in different ways. For example, the IDL routine SHIFT (and the Matlab routine CIRCSHIFT) moves the array elements to the *right* for a "positive" shift (a negative shift moves elements to the left), whereas the Fortran 95 CSHIFT routine moves the array elements to the *left* for a positive shift (a negative shift moves elements to the right). Thus

$$L = 10 \text{ m}; N = 16; v_f = 0.1 \text{ m}^{-1}; v_{Ny} = 0.8 \text{ m}^{-1}; \text{phases} = \pi/2$$

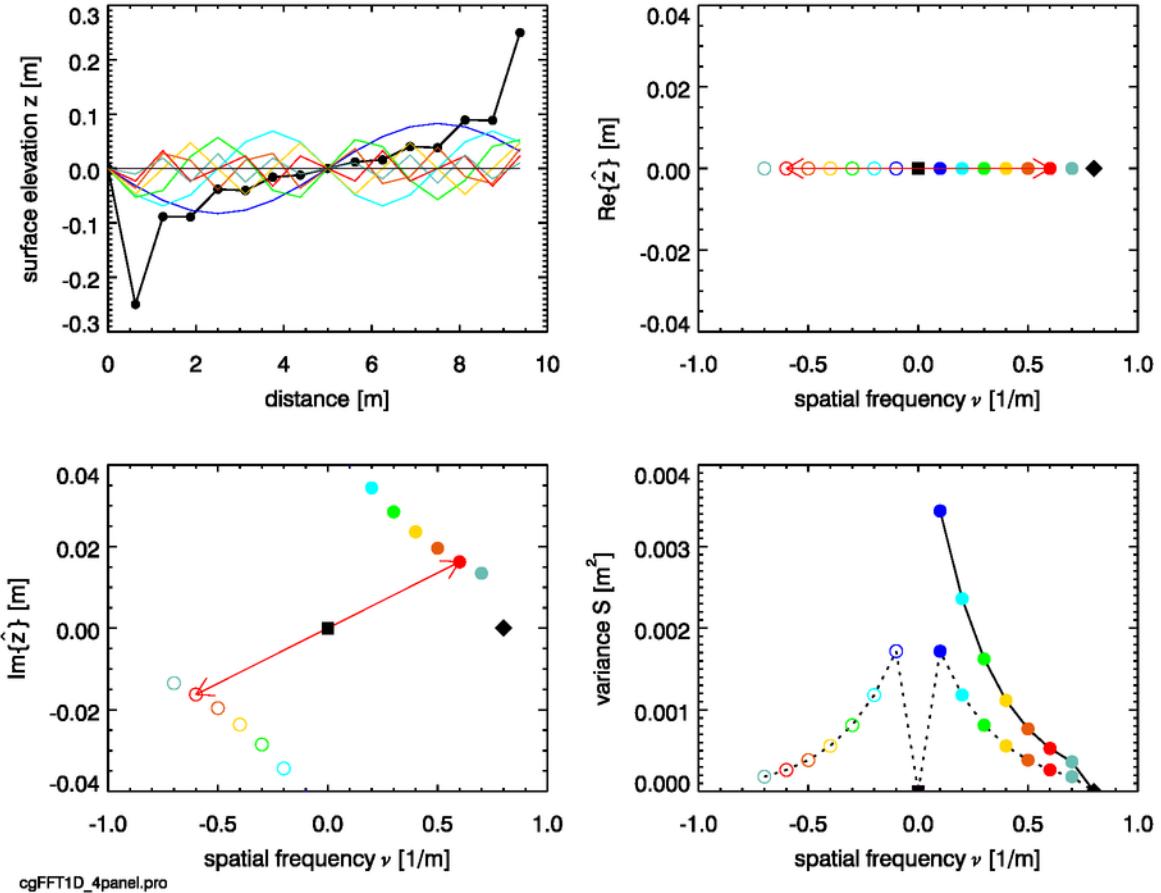


Figure 2.3: A one-dimensional surface composed of sine waves. Compare with Figs. 2.1 and 2.2.

In IDL:

```
math order = SHIFT(FFT order, N/2-1)
FFT order = SHIFT(math order, -(N/2-1))
```

In Fortran95:

```
math order = CSHIFT(FFT order, -(N/2-1))
FFT order = CSHIFT(math order, N/2-1)
```

2.5 Variance Spectra to Surfaces

The previous section showed the major features of the Fourier analysis of a sea-surface elevation record. We started with a sample of a sea surface and ended with the corresponding discrete variance spectrum (or estimate of the variance spectral density after division by the frequency interval). In

this section we learn how to go in the reverse direction: start with a variance spectrum and generate a random realization of the corresponding sea surface.

We will generate examples using two widely used variance spectra. The Pierson and Moskowitz (1964) spectrum describes fully developed gravity waves. Fully developed waves are an idealization of the sea state after the wind has been blowing for an infinitely long time over an infinite fetch. The Elfouhaily et al. (1997) spectrum describes the full range of gravity to capillary waves, including a parameter for the “age” of the waves, from young (the wind has just started blowing) to fully developed seas. The mathematical formulations of these spectra are given for reference in Appendix B.1.

2.5.1 Theory

The essence of the process is as follows:

1. **Choose the domain size.** To create a sea surface over a spatial domain at a given time, we pick the length L of the region $[0, L]$. To generate a time series at a given point, we pick the length of the time series.
2. **Choose the number of points for the DFT.** This number N is the number of frequencies at which we will sample the variance spectrum, and equals the number of samples of the sea surface that will be generated. In normal usage, pick N to be a power of 2 so that an inverse FFT routine can be used to evaluate the inverse DFT.
3. **Choose the frequency variable.** To generate a sea surface over a spatial domain at a given time, we can use either wavenumber ν or angular spatial frequency k . To generate a time series at a given point, we can use either f or ω .
4. **Choose a variance spectrum** The variance spectrum must be expressed in terms of the chosen frequency variable. §B.2 gives the Pierson-Moskowitz one-sided spectrum as a function of ν , k , or ω and shows how to change variables in a density function.
5. **Choose the wind speed.** Pick a wind speed, and perhaps other physical parameters such as the age of the waves to be generated if required by the chosen variance spectrum.
6. **Create random Hermitian amplitudes.** This is the tricky part. We must create an array of randomized discrete Hermitian Fourier amplitudes $\hat{z}(u)$, starting with the chosen continuous variance spectrum.
7. **Take the inverse DFT of the random amplitudes.** The inverse DFT converts the Fourier amplitudes to the physical wave heights.
8. **Extract the sea surface heights.** The inverse DFT of the complex amplitudes returns a complex array. The real part of this array is the random realization of the sea surface heights, and the imaginary part is zero.
9. **Check your results.** This is extremely important during code development. For example, take the DFT of the generated surface heights to see if you get back to the Fourier amplitudes and variance spectrum you started with.

We now proceed through these steps and discuss them in detail for a specific example.

Steps 1 and 2: Let us generate a sea surface over the region from $x = 0$ to $x = L = 100$ m. The longest wavelength that can be resolved is then 100 m. We use $N = 1024$, which gives a spatial grid resolution of $\Delta x = L/N = 0.0977$ m. This means that the shortest wavelength that can be resolved, the two-point wave or Nyquist wavelength, is $2\Delta x = 0.1954$ m.

Step 3: We used wavenumber ν in the previous section because of its easy interpretation. Now let's use angular spatial frequency k , which is more commonly used. The fundamental frequency is then $k_f = 2\pi/L = 0.0628$ rad m $^{-1}$. The highest frequency sampled, the Nyquist frequency, is $k_{Ny} = (N/2)k_f = 32.15$ rad m $^{-1}$. Note that $(2\pi)/k_{Ny} = 0.1954$ m which, as noted above, is the wavelength of the two-point wave.

Step 4: For this example, we use the Pierson-Moskowitz variance spectrum in terms of angular spatial frequency k . This is given by Eq. (B.11). *Note that this is a one-sided spectrum*, which is defined for positive k values.

Step 5: The wind speed at 10 m elevation is $U_{10} = 5$ m s $^{-1}$. The wind speed is the only input to the Pierson-Moskowitz spectrum.

Step 6: We now discuss in detail how to generate a set of random discrete Fourier amplitudes that are physically consistent with the chosen variance spectrum. These amplitudes must be defined for both positive and negative frequencies, and the amplitudes must be Hermitian. We first define

$$\hat{z}_o(k_u) \equiv \frac{1}{\sqrt{2}} [\rho(k_u) + i\sigma(k_u)] \sqrt{\frac{\mathcal{S}_{1s}(k=k_u)}{2} \Delta k}. \quad (2.9)$$

Here $\mathcal{S}_{1s}(k=k_u)$ denotes the continuous spectral density $\mathcal{S}_{1s}(k)$ evaluated at $k = k_u$. $\Delta k = k_f$ is the spatial frequency sampling interval. $\hat{z}_o(k_u)$ must be defined for both positive and negative discrete frequencies in order to create the Hermitian amplitudes for use in the inverse DFT. Recalling the comments of the previous section, we convert the one-sided continuous spectrum $\mathcal{S}_{1s}(k)$ into a two-sided discrete variance function by

1. dividing its magnitude by 2, assuming that $\mathcal{S}_{2s}(-k) = \mathcal{S}_{2s}(k)$;
2. multiplying the continuous spectral density by the fundamental frequency interval Δk , which gives the variance contained in a finite frequency interval at each frequency k_u .

To emphasize the discrete vs continuous functions, and for brevity of notation, let us write the frequency index u for the frequency k_u . Then Eq. (2.9) becomes

$$\hat{z}_o(u) = \frac{1}{\sqrt{2}} [\rho(u) + i\sigma(u)] \sqrt{\mathcal{S}_{2s}(u)}, \quad (2.10)$$

where $\mathcal{S}_{2s}(u)$ denotes the *two-sided* discrete variance spectrum at frequency k_u . $\hat{z}_o(u)$ can now be evaluated for both positive and negative k_u . The 0 and Nyquist frequencies are always special cases: set $\mathcal{S}_{2s}(0) = 0$ and $\mathcal{S}_{2s}(k_{Ny}) = \mathcal{S}_{1s}(k_{Ny})$. $\rho(k_u) \equiv \rho(u)$ and $\sigma(k_u) \equiv \sigma(u)$ are independent random numbers drawn from a normal distribution with zero mean and unit variance, denoted $\rho, \sigma \sim \mathcal{N}(0, 1)$. A different pair is drawn for each u value.

$\hat{z}_o(u)$ is a random variable. Let $\langle \dots \rangle$ denote the expectation of the enclosed variable. The expected value of $|\hat{z}_o(u)|^2$, $\langle \hat{z}_o(u)\hat{z}_o^*(u) \rangle$, gives back whatever variance function is used for $\mathcal{S}_{2s}(u)$:

$$\begin{aligned}\langle \hat{z}_o(u)\hat{z}_o^*(u) \rangle &= \left\langle \left\{ \frac{1}{\sqrt{2}}[\rho(u) + i\sigma(u)]\sqrt{\mathcal{S}_{2s}(u)} \right\} \left\{ \frac{1}{\sqrt{2}}[\rho(u) - i\sigma(u)]\sqrt{\mathcal{S}_{2s}(u)} \right\} \right\rangle \\ &= \frac{\mathcal{S}_{2s}(u)}{2} [\langle \rho^2 \rangle + \langle \sigma^2 \rangle] = \mathcal{S}_{2s}(u)\end{aligned}$$

because $\langle \rho^2 \rangle = \langle \sigma^2 \rangle = 1$ for $\mathcal{N}(0, 1)$ random variables. Thus $\hat{z}_o(u)$ is consistent with the chosen variance spectrum. However, $\hat{z}_o(u)$ is not Hermitian, so the inverse DFT would not give a real sea surface.

Next define $\hat{z}(u)$ as

$$\hat{z}(u) \equiv \frac{1}{\sqrt{2}} [\hat{z}_o(u) + \hat{z}_o^*(-u)]. \quad (2.11)$$

This function is clearly Hermitian, so the inverse DFT applied to $\hat{z}(u)$ will give a real-valued $z(x_r) \equiv z(r)$. Moreover, this $\hat{z}(u)$ is consistent with the variance spectrum:

$$\begin{aligned}\langle |\hat{z}(u)|^2 \rangle &= \langle \hat{z}(u)\hat{z}^*(u) \rangle \\ &= \left\langle \frac{1}{\sqrt{2}} \left[\frac{1}{\sqrt{2}}[\rho(u) + i\sigma(u)]\sqrt{\mathcal{S}_{2s}(u)} + \frac{1}{\sqrt{2}}[\rho(-u) - i\sigma(-u)]\sqrt{\mathcal{S}_{2s}(-u)} \right] \times \right. \\ &\quad \left. \frac{1}{\sqrt{2}} \left[\frac{1}{\sqrt{2}}[\rho(u) - i\sigma(u)]\sqrt{\mathcal{S}_{2s}(u)} + \frac{1}{\sqrt{2}}[\rho(-u) + i\sigma(-u)]\sqrt{\mathcal{S}_{2s}(-u)} \right] \right\rangle \\ &= \frac{1}{4} \left\langle [\rho^2(u) - i\rho(u)\sigma(u) + i\sigma(u)\rho(u) + \sigma^2(u)] \mathcal{S}_{2s}(u) + \right. \\ &\quad [\rho(u)\rho(-u) + i\rho(u)\sigma(-u) + i\sigma(u)\rho(-u) - \sigma(u)\sigma(-u)] \sqrt{\mathcal{S}_{2s}(u)} \sqrt{\mathcal{S}_{2s}(-u)} + \\ &\quad [\rho(-u)\rho(u) - i\rho(-u)\sigma(u) - i\sigma(-u)\rho(u) - \sigma(-u)\sigma(u)] \sqrt{\mathcal{S}_{2s}(-u)} \sqrt{\mathcal{S}_{2s}(u)} + \\ &\quad \left. [\rho^2(-u) + i\rho(-u)\sigma(-u) - i\sigma(-u)\rho(-u) + \sigma^2(-u)] \mathcal{S}_{2s}(-u) \right\rangle \\ &= \frac{1}{2} [\mathcal{S}_{2s}(u) + \mathcal{S}_{2s}(-u)] = \mathcal{S}_{2s}(u).\end{aligned}$$

Here we have noted that $\langle \rho(u)\rho(-u) \rangle = 0$, etc., because the random variables are uncorrelated for different u values.

Equations (2.9) and (2.11) are the key to generating random sea surfaces from variance spectra. $\hat{z}(u)$ defined by these equations contains random noise, which leads to a sea surface with random amplitudes and phases for the component waves of different frequencies. Any one of these surfaces has a variance spectrum that looks like the chosen spectrum plus random noise. However, on average over many realizations, the noise in these spectra will average out, leaving the variance spectrum. Figure 2.4 below illustrates these important ideas, but first we must complete the surface generation.

Step 7: Compute the inverse DFT of the $\hat{z}(u)$ of Eq. (2.11). The result is a complex function $Z(x)$:

$$Z(x_r) \equiv Z(r) = \mathfrak{D}^{-1}\{\hat{z}(u)\}.$$

A crucial warning to this step is that the $u = 0, \dots, N - 1$ elements of the $\hat{z}(u)$ array *must* be in the FFT frequency order given by Eq. (2.8) when using a FFT routine to evaluate the DFT. $Z(r)$ is returned with x_r values in the order from $x_0 = 0$ to $x_{N-1} = (N - 1)\Delta x$.

Step 8: Extract the surface. The inverse DFT returns a complex array $Z(x_r)$ whose real part is the surface elevations $z(x_r)$ and whose imaginary part is 0. The surface elevations are extracted as the real part of $Z(x_r)$:

$$z(x_r) = \text{Re}\{Z(x_r)\}.$$

Step 9: Check the results. There are many places along the way to lose a $\sqrt{2}$ or to mess up array indexing. At the minimum, it is worthwhile to check that the mean of the generated surface is zero, and that the imaginary part of $Z(x_r) = 0$ (to within a small amount of numerical roundoff error).

When developing computer code, or when first learning this material, it is also a good idea to take the forward DFT of $Z(x_r)$ to make sure that the input Fourier amplitudes $\hat{z}(u)$ are recovered, and that the variance spectrum corresponding to $z(x_r)$ is consistent with the one chosen in Step 4. Indeed, this check was what led to these notes.

Equations (2.9) and (2.11) are, with minor changes in notation, Eqs. (42) and (43), respectively, of Tessendorf (2004, §4.3). However, Tessendorf's version of Eq. (2.9) appears to use a one-sided variance spectrum (his example used the one-sided Phillips spectrum of his Eq. (40)) without the division by 2 seen in Eq. (2.9), which is needed to convert the one-sided spectrum to a two-sided spectrum. Nor does he show the Δk factor needed to convert a continuous spectral density to a discrete function. His version of Eq. (2.11) does not contain the overall factor of $1/\sqrt{2}$ seen above. These missing factors mean that in a round-trip calculation

$$\text{variance spectrum} \rightarrow \text{DFT}^{-1} \rightarrow \text{sea surface} \rightarrow \text{DFT} \rightarrow \text{variance spectrum},$$

you do not get back to the original variance spectrum. In other words, *the Tessendorf equations do not conserve wave variance (i.e., wave energy)*. Even if he included the Δk factor in his actual computations, the missing factors of $1/\sqrt{2}$ in his versions of our Eqs. (2.9) and (2.11) give an overall factor one-half on the amplitudes, which corresponds to a factor of four error in the variance. That is, waves generated using the Tessendorf equations have amplitudes that are too large.

Tessendorf (2004) discusses much more than just Fourier transform techniques, and his notes have been very influential in the computer graphics industry. In 2008 he deservedly received an Academy Award for Technical Achievement for showing the movie industry how to generate and render sea surfaces, as well as for his many other pioneering accomplishments in efficiently computing and rendering fluid motions into visually appealing images. (The first movie to use his techniques was *Waterworld*, followed by dozens of others including *Titanic*.) When I checked with him (personal communication) about the missing numerical factors, he readily acknowledged that Eqs. (2.9) and (2.11) are the correct ones, but noted that Hollywood doesn't care about conservation of energy. I suppose that should be no surprise, since movies seem to have no problem with rockets going faster than the speed of light, sound propagating through the vacuum of outer space, or time travel that violates causality. Tessendorf's equations are widely cited (especially in the computer graphics literature), always without comment about the missing scale factors. You can find many other mathematical and physical horrors in the computer graphics world. I am therefore sadly forced to conclude that many people have created sea surfaces without checking their calculations for internal consistency and physical correctness. That may be acceptable in a fantasy world, but such laxness is not permissible in science. I wrote this tutorial in part to make

clear exactly how surfaces should be generated so that they are physically correct and suitable for scientific work.

2.5.2 Example: A Roundtrip Calculation

Figure 2.4 shows an example of 1-D surface waves generated using the Pierson-Moskowitz spectrum for a wind speed of 5 m/s, and the recovery of the variance spectrum from the generated surface. The blue curve in the upper-left panel shows the Pierson-Moskowitz spectrum as defined by Eq. (B.11). The red dots show the frequencies at which the continuous spectrum is sampled. Those dots blur together at the higher frequencies because of the log scale, but the k_u points are equally spaced at intervals of the fundamental frequency $\Delta k = k_f = 2\pi/L = 0.0628$ rad/m. The last sampled frequency is $k_{Ny} = 32.17$ rad/m. The bottom panel shows the sea surface elevations $z(x_r)$ generated for a particular sequence of random numbers $\rho(u), \sigma(u)$.

The red line in the upper-right panel shows the function

$$\mathcal{P}(u) \equiv \frac{|\hat{z}(u)|^2}{\Delta k} = \frac{1}{\Delta k} |\mathfrak{D}\{z(r)\}|^2. \quad (2.12)$$

$\mathcal{P}(u)$ is the discrete variance function for this particular $z(x_r)$ surface. Schuster (1898) called $\mathcal{P}(u)$ a *periodogram*. The periodogram $\mathcal{P}(u)$ contains random noise because $z(x_r)$ is a random realization of the sea surface, which was generated by applying random noise to the variance spectrum. This particular $z(x_r)$ is analogous to a particular measurement of the sea surface. Had we drawn a difference sequence of random numbers for use in Eq. (2.10), we would have generated a different sea surface, and a different $\mathcal{P}(u)$. However, we can expect that if we average together many different $\mathcal{P}(u)$, corresponding to many different sets of $z(x_r)$, the noise would average out and we would be left with a curve close to the variance spectrum we started with, which is shown in blue. Numerical experimentation shows that averaging 100 $\mathcal{P}(u)$ generated from 100 independent sea surface realizations gives an average $\mathcal{P}(u)$ that is almost indistinguishable from the blue curve at the scale of this plot. Thus $\mathcal{P}(u) \equiv \mathcal{P}(k_u)$ is an approximation of the variance density spectrum $\mathcal{S}(k)$, denoted $\mathcal{P}(k_u) \doteq \mathcal{S}(k)$. This averaging processes leads to the topic of *spectrum estimation*, which considers such problems as how many sets of measurements of $z(x_r)$ are needed to estimate the variance spectrum to within certain error bounds. Fortunately, we need not pursue that here.

At the minimum, you should always check to see that Parseval's relation, Eq. (1.22), is satisfied. For the simulation of Fig. 2.4, we have

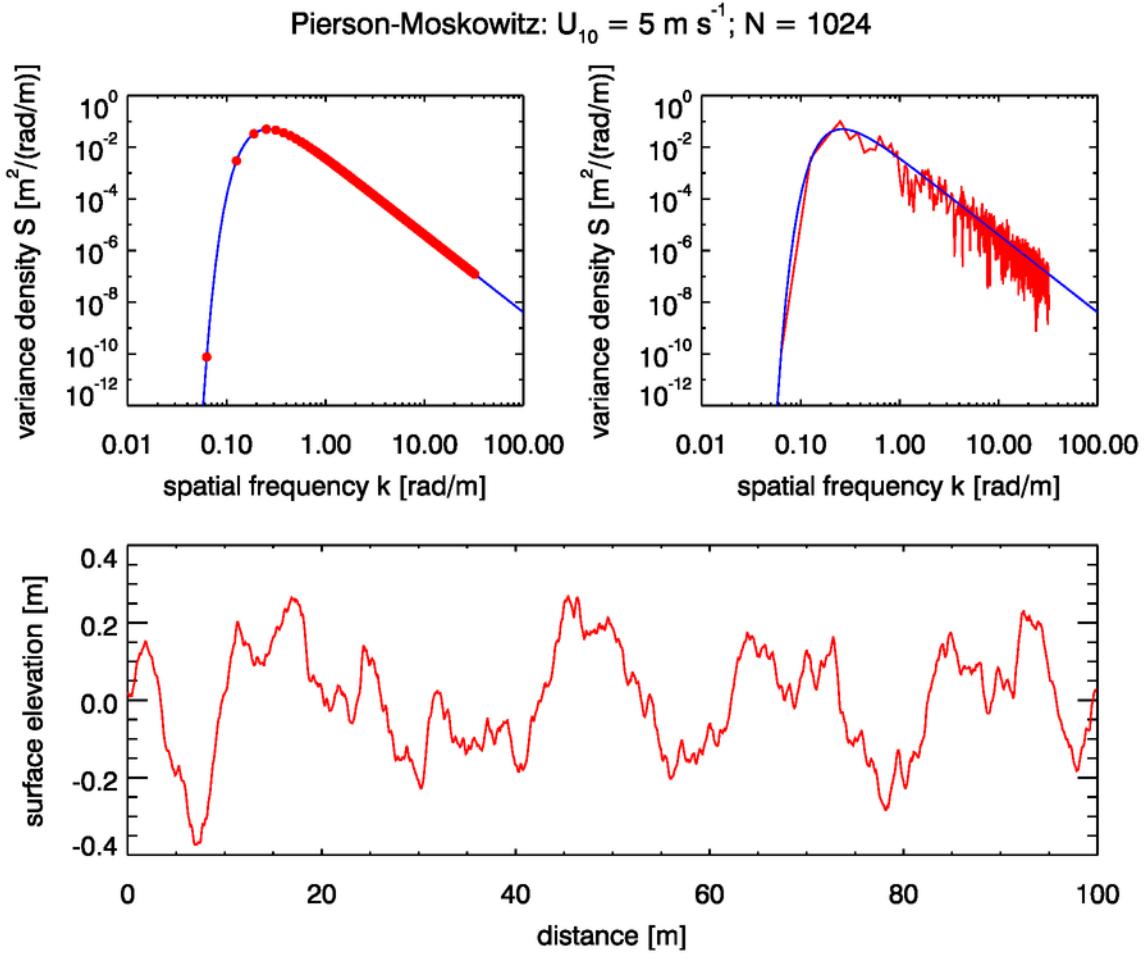
$$\sum_{r=0}^{N-1} |z(r)|^2 = N \sum_{u=0}^{N-1} |\hat{z}(u)|^2 = 19.395 \text{ m}^2.$$

There are sometimes other checks that can be made. For example, the Pierson-Moskowitz spectrum is simple enough that it can be analytically integrated over all frequencies. This gives

$$\langle z^2 \rangle = \int_0^\infty \mathcal{S}_{PM}(k) dk = 3.04 \cdot 10^{-3} \frac{U_{10}^4}{g^2}, \quad (2.13)$$

where we have recalled that the variance spectral density is related to the variance of the sea surface. The variance of the generated zero-mean sea surface can be computed from

$$var(z) = \frac{1}{N} \sum_{r=0}^N z^2(x_r) \quad (2.14)$$



`cgFFT1D_roundtrip.pro`

Figure 2.4: Example of a 1-D random sea surface generated from the Pierson-Moskowitz spectrum for $U_{10} = 5 \text{ m s}^{-1}$ (the wind speed at 10 m above mean sea level). The longest resolvable wave has wavelength $L = 100 \text{ m}$. For $N = 1024$, the two-point wave has wavelength $2L/N = 0.195 \text{ m}$. This is already less than the smallest wavelength (highest frequency) for which the Pierson-Moskowitz spectrum should be used.

and compared with the analytical expectation. For the surface seen in Fig. 2.4, Eq. (2.14) gives $\text{var}\{z\} = 0.0189 \text{ m}^2$ vs. the theoretical value of $\langle z^2 \rangle = 0.0197 \text{ m}^2$ from Eq. (2.13). This agreement to within a small amount of random noise indicates that all is probably well with the calculations. Indeed, the average $\text{var}(z) \pm$ one standard deviation for 100 independent simulations is 0.020 ± 0.007 , which agrees well with the theoretical value.

The significant wave height $H_{1/3}$ is by definition the height (trough-to-crest distance) of the highest one-third of the waves. To a good approximation, this is related to the expectation of the variance by

$$H_{1/3} = 4\sqrt{\langle z^2 \rangle}.$$

In the present example, this formula gives $H_{1/3} = 0.55 \text{ m}$. The average significant wave height for 100 simulations is 0.56 ± 0.09 (average \pm one standard deviation). If you spend enough time in a

sea kayak to develop intuition about wave heights as a function of wind speed, a half-meter height for the largest waves is about right for a 5 m s^{-1} or 10 knot wind speed.

To summarize this chapter: we have learned how to start with a wave variance spectral density function and generate random discrete Fourier amplitudes. The inverse DFT of those amplitudes gives a random realization of a sea surface that is physically consistent with the chosen variance spectrum. That is all that we can ask of the Fourier transform technique.

2.6 Numerical Resolution

The question naturally arises: “How large a region L and how many points N must be used in generating sea surface realizations?” Not surprisingly, the general answer is, “It depends on your application.” It is now computationally possible to create 2-D surfaces (as described in the next chapter) with sufficiently large N values in the x and y directions, but run times are prohibitive. Kay et al. (2011) created 2-D surfaces with 65536×65536 points in each dimension, which sampled the energy spectrum for wavelengths from 200 m gravity waves to 3 mm capillary waves. However, it took 6 hours to create just one surface on a 3 GHz computer. Many ray tracing applications use tens of thousands of surface realizations to get good statistical averages. Thus, the specific answer for brute-force scientific calculations is usually, “A bigger N than you can run on your computer.” This section shows how to finesse certain calculations so that large- N calculations can be avoided.

For creating visually appealing sea surfaces for rendering into a movie scene, $N = 2^{10} = 1024$, $2^{11} = 2048$, or perhaps at most $2^{12} = 4096$ is usually adequate. This is because the visual impression of a sea surface is, to first order, determined by the height of the waves, which in turn is governed by the largest gravity waves for the given wind speed. If you are rendering an image, you don’t need to have a finer resolution for the sea surface than maps to an image pixel. For example, if you are going to simulate what a CCD with 1024×1024 pixels would record, and you are looking down onto the sea surface from above at an elevation where one CCD pixel sees a $0.1 \text{ m} \times 0.1 \text{ m}$ patch of the sea surface, then you need to resolve the sea surface with a grid spacing of at most 0.1 m. Modeling a $100 \text{ m} \times 100 \text{ m}$ patch of sea surface with a spatial resolution of 1024×1024 gives a grid spacing of 0.1 m, which would be adequate for the CCD image simulation.

Unfortunately, for accurate scientific calculations of sea surface reflectance, N may have to be much larger. This is because the surface reflectance depends to first order on the surface wave slope, and the slope is strongly affected by the smallest waves with spatial scales down to capillary size of a few millimeters.

The k value of the peak of the Pierson-Moskowitz spectrum of Eq. (B.11) is easily obtained from setting $dS_{\text{PM}}(k)/dk = 0$ and solving for the value of k . This gives

$$k_p = \left(\frac{2\beta}{3} \right)^{0.5} \frac{g}{U_{19}^2},$$

which corresponds to a wavelength of $\Lambda_p = 2\pi/k_p$. Table 2.6 shows the values of k_p and Λ_p for a few wind speeds.

$U_{19}[\text{m/s}]$	$k_p [\text{rad/m}]$	$\Lambda_p [\text{m}]$
5	0.276	23
10	0.069	91
15	0.031	205
20	0.017	364

Table 2.1: Spatial frequencies and wavelengths corresponding to peak variance for the Pierson-Moskowitz spectrum $\mathcal{S}_{\text{PM}}(k)$.

If we use $L \approx 2\Lambda_p$ for the size of the spatial domain, we will resolve the large gravity waves, which contain most of the wave variance. (Keep in mind that density functions do not have a unique maximum: the location of the maximum of a density function depends on which frequency variable is chosen. See the discussion of this on the *Ocean Optics Web Book* page on “A Common Misconception” at

http://www.oceanopticsbook.info/view/light_and_radiometry/level_2/a_common_misconception. Differentiating Eq. (B.13) and using the dispersion relation $\omega = \sqrt{gk}$ leads to $\Lambda_p = 2\pi(1.25/\beta)^{0.25}g/U_{19}^2$, which gives 73 m at $U_{19} = 10$. However, either version of Λ_p gives adequate guidance for our present purpose.)

Now consider how many points it takes to almost fully resolve, or account for, the variances of elevation and slope spectra when they are sampled in a DFT. For a specific example, we take $U_{10} = 10$ m/s and $L = 200$ m. We use the omnidirectional, or 1-D, part of the Elfouhaily et al. (1997) spectrum presented in §B.3. This $\mathcal{S}(k)$ is given by Eq. (B.16) and the associated equations. This variance spectrum and the corresponding slope spectrum $k^2\mathcal{S}(k)$ are plotted as the green curves in Fig. B.3, and again by the blue curves in the next 3 figures.

As we have learned (c.g. §2.2 and B.1), the expected variance of the sea surface is given by

$$\langle z^2 \rangle = \int_0^\infty \mathcal{S}(k) dk, \quad (2.15)$$

and the mean-square slope is given by

$$\langle \sigma^2 \rangle = mss = \int_0^\infty k^2 \mathcal{S}(k) dk. \quad (2.16)$$

Such integrals usually must be numerically evaluated with the limits of $k = 0$ and ∞ being replaced by some small value k_0 and some large but finite value k_∞ . In the present case, I used $k_0 = 0.01$ and $k_\infty = 10^4$, with 10^6 evaluation points in between these limits. That gives an accurate evaluation of Eqs. (2.15) and (2.16) for the $U_{10} = 10$ m s⁻¹ spectra. The results are $\langle z^2 \rangle = 0.4296$ m² and $mss = 0.06011$ rad².

Now suppose we sample the spectrum using $N = 1024$ points, which leads eventually to a sea surface realization $z(x_r)$ with 1024 points. For $L = 200$ m the fundamental frequency is $k_f = 2\pi/L = 0.0314$ rad/m. This point is shown by the left-most red dot with the black center in Fig. 2.5. For this L and N , the Nyquist frequency is $k_{\text{Ny}} = 2\pi/(2L/N) = 16.085$ rad/m, which is shown by the right-most black and red dot. These two end points and the $N - 2$ red vertical lines in between show the discrete points where the variance spectrum is sampled.

The elevation and slope variances that are accounted for in a sampling scheme with N points are given by

$$z^2(N) = \int_{k_f}^{k_{Ny}} \mathcal{S}(k) dk,$$

and

$$mss(N) = \int_{k_f}^{k_{Ny}} k^2 \mathcal{S}(k) dk.$$

In the present example, these integrals are $z^2(N = 1024) = 0.4219 \text{ m}^2$ and $mss(N = 1024) = 0.02584 \text{ rad}^2$. Thus the finite range of the sampled frequencies includes the fraction

$$f_E \equiv \frac{z^2(N)}{\langle z^2 \rangle} = \frac{0.4219}{0.4296} = 0.982$$

of the total variance of the sea surface elevation. However, the corresponding fraction of the sampled slope variance is just $f_S = 0.02584/0.06011 = 0.430$. Thus $N = 1024$ is sampling 98% of the elevation variance but only 43% of the slope variance. This sampling is acceptable if we are interested only in creating a sea surface that looks realistic to the eye. The large gravity waves will be correctly simulated, and that is what counts for visual appearance.

However, if our interest is ray tracing to simulate the optical reflectance and transmission properties of the sea surface, then we *must* also sample almost all of the slope variance. This is because, to first order, reflectance and transmission of light are governed by the slope of the sea surface, not by its height. As will be seen in §3.5, if we under sample the slope variance, then the generated surface will be too smooth at the smallest spatial scales, and the computed optical properties will be incorrect. The brute-force approach to sampling more of the slope spectrum is to increase N .

Figure 2.6 shows the sampling points when $N = 2^{16} = 65536$. Now the sampled points extend into the capillary-wave spatial scale: the smallest resolvable wave is 0.006 m. (Note that the fundamental frequency is fixed by the choice of L , which also fixes the spacing of the samples, $\Delta k = k_f$.) The elevation variance integral over the sampled ranges is unchanged (we are still missing a small bit of the variance from the very longest waves with frequencies below k_f . However, the slope integral is now $mss(N = 65536) = 0.05909 \text{ rad}^2$. Thus we are now sampling a fraction $f_S = 0.05909/0.06011 = 0.983$ of the slope variance. The resolution of both the elevation and slope spectra are now acceptable for almost any application. Although using a very large N is computationally feasible in one dimension, such a large N is computationally impracticable in two dimensions, when we would need a grid of size $N \times N$. In the present example, this would require $65536^2 = 4.3$ Gbytes of storage for just one real array, as well as a corresponding increase in run time for the FFT. Another approach to resolving the slope variance is therefore needed.

As we have seen, the unsampled frequencies greater than the Nyquist frequency can account for a large part of the slope variance. These frequencies represent the smallest gravity and capillary waves. The amplitudes of these waves are small, so they contribute little to the total wave variance, but their slopes can be large. One way to account for these “missing” waves is to alias their variance into the waves of the highest frequencies. The highest frequency waves that are sampled will then contain too much variance, i.e., they will have amplitudes that are too large for their wavelengths, which will increase their slopes. One way to do this is as follows.

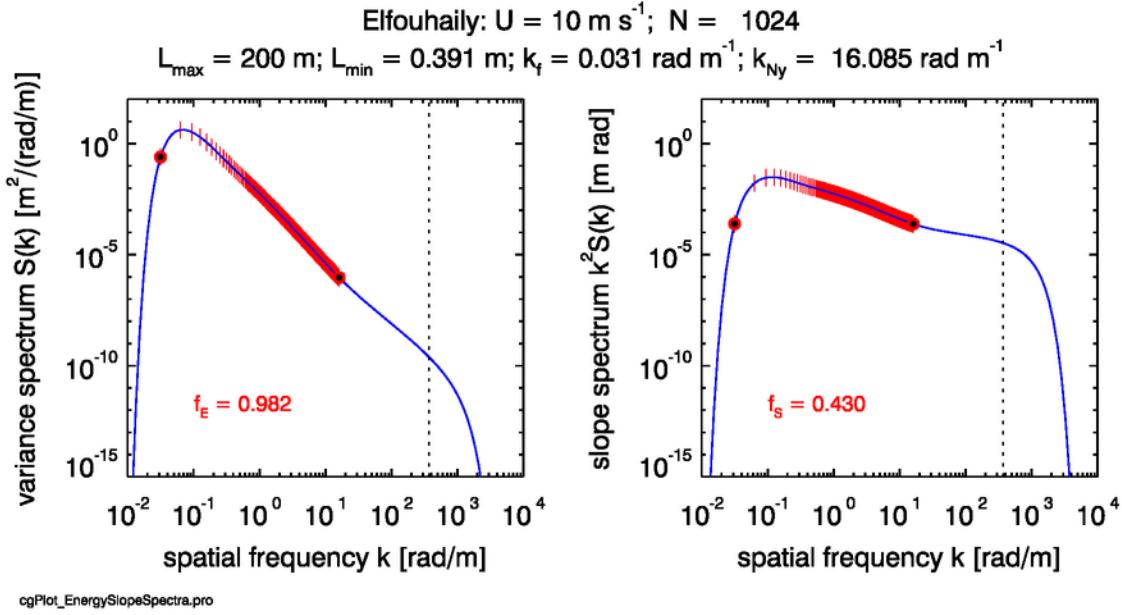


Figure 2.5: Example sampling of elevation and slope spectra for $N = 1024$. The red dots and lines show the regions of the elevation and slope spectra sampled using 1024 points. The elevation spectrum is adequately sampled, but the slope spectrum is under sampled.

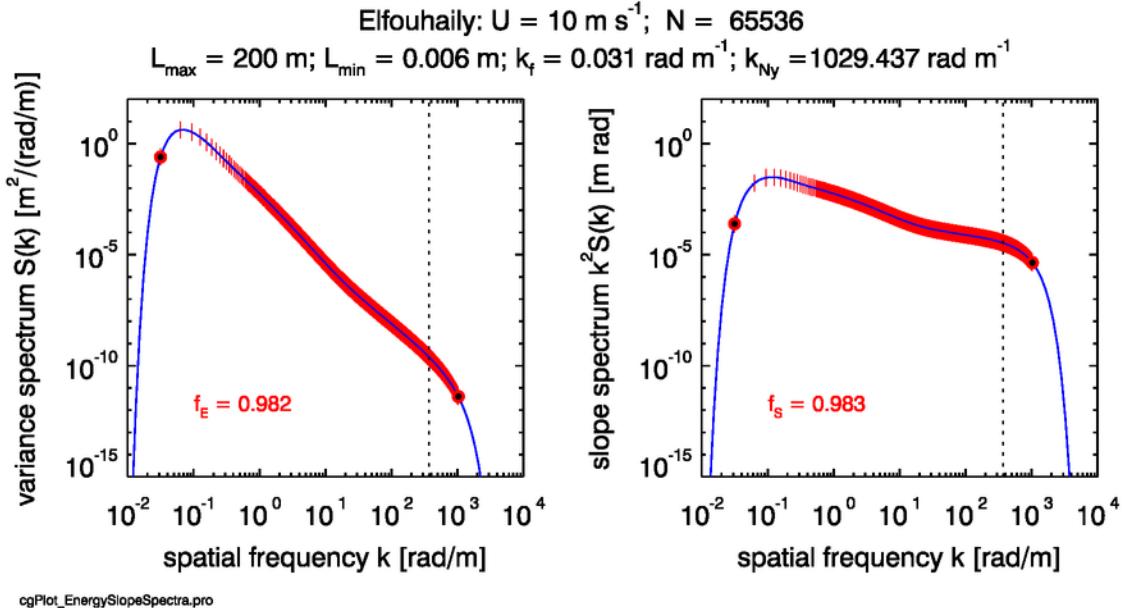


Figure 2.6: Example sampling of elevation and slope spectra for $N = 65536$. The red dots and lines show the sampled regions of the elevation and slope spectra. Both the elevation and the slope spectra are adequately sampled.

We seek an adjusted or re-scaled elevation spectrum

$$\tilde{\mathcal{S}}(k) \equiv [1 + \delta(k)]\mathcal{S}(k) \quad (2.17)$$

such that the integral of $k^2\tilde{\mathcal{S}}(k)$ over the sampled region equals the integral of the true $k^2\mathcal{S}(k)$ over the full spectral range. Then sampling the re-scaled spectrum $\tilde{\mathcal{S}}(k)$ will lead to the same slope variance as would be obtained from sampling the correct spectrum $\mathcal{S}(k)$ over the entire range of frequencies.

We can choose L so that the low frequencies are well sampled starting at the fundamental frequency $k_f = 2\pi/L$. There is thus no need to modify the low-frequency part of the variance spectrum, which if done, would adversely affect the total wave variance. We want to modify only the higher frequencies of the sampled region, which contribute the most to the wave slope but little to the wave elevation. A simple approach is to take $\delta(k)$ to be a linear function of k between the spectrum peak k_p and the highest sampled frequency k_{Ny} , and zero elsewhere:

$$\delta(k) \equiv \begin{cases} 0 & \text{if } k \leq k_p \\ \delta_{Ny} \left(\frac{k - k_p}{k_{Ny} - k_p} \right) & \text{if } k > k_p \end{cases} \quad (2.18)$$

δ_{Ny} is a parameter that depends on the spectrum (i.e., the wind speed), the size L of the spatial domain, and the number of samples N . We must determine the value of δ_{Ny} so that this $\delta(k)$ “adds back in” the unresolved slope variance. The added variance will be zero at the peak frequency k_p and largest at the Nyquist frequency. That is, the $\delta(k)$ function will make only a small change to the variance spectrum at the low frequencies, and the change will be largest near the highest sampled frequencies, which is consistent with the idea that the high frequency waves have the largest slopes. (There is nothing physical about the functional form of Eq. (2.18); it is simply a single-parameter *ad hoc* function that works well in practice. Nonlinear functions could be used, e.g. to concentrate more of the variance into the frequencies closest to the Nyquist frequency. However, those functions could have more than one free parameter to determine and, in any case, the end result is the same: a sea surface that reproduces the height and slope variances of the fully sampled spectrum.)

To determine δ_{Ny} we thus have

$$\begin{aligned} mss &= \int_0^\infty k^2 \mathcal{S}(k) dk \approx \int_{k_0}^{k_\infty} k^2 \mathcal{S}(k) dk = \int_{k_0}^{k_{Ny}} k^2 \mathcal{S}(k) dk + \int_{k_{Ny}}^{k_\infty} k^2 \mathcal{S}(k) dk \\ &\equiv \int_{k_0}^{k_{Ny}} k^2 \tilde{\mathcal{S}}(k) dk = \int_{k_0}^{k_{Ny}} k^2 \mathcal{S}(k) dk + \delta_{Ny} \int_{k_p}^{k_{Ny}} k^2 \left(\frac{k - k_p}{k_{Ny} - k_p} \right) \mathcal{S}(k) dk. \end{aligned}$$

The right-most terms of these equations give (after recalling that $\delta(k) = 0$ for $k \leq k_p$)

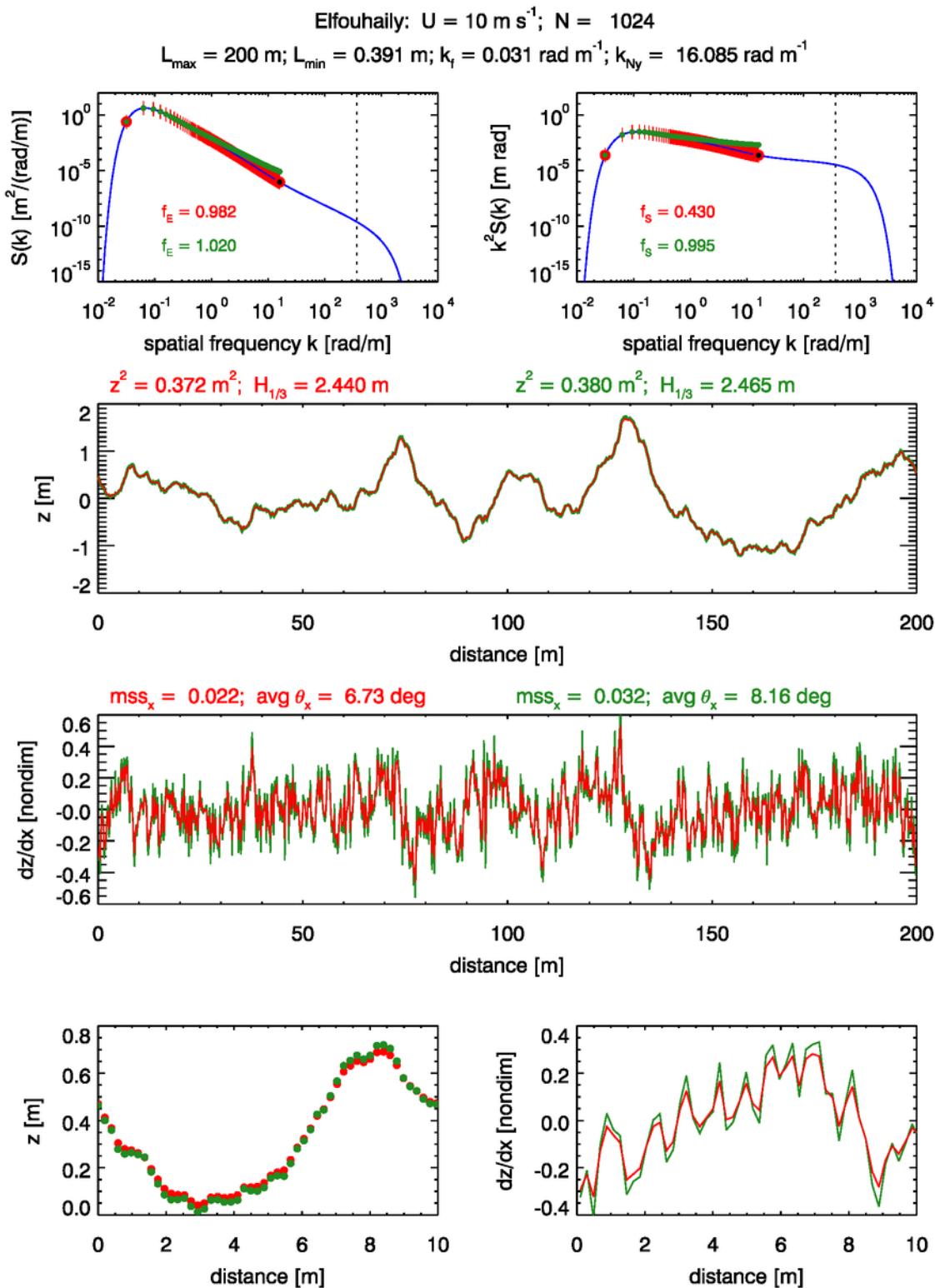
$$\delta_{Ny} = \frac{\int_{k_p}^{k_{Ny}} k^2 \mathcal{S}(k) dk}{\int_{k_p}^{k_{Ny}} k^2 \left(\frac{k - k_p}{k_{Ny} - k_p} \right) \mathcal{S}(k) dk}. \quad (2.19)$$

Figure 2.7 shows example 1-D surface realizations created with $N = 1024$ samples and both the true and scaled variance spectra. The upper two panels reproduce Fig. 2.5, except that the sampled points of the re-scaled spectra are shown in green. It is clear that the $\delta(k)$ function has added progressively more variance to the higher frequencies. The re-scaled variance spectrum does

of course contain somewhat more variance over the sampled region than does the true spectrum. As the green inset f_E number shows, this re-scaling has increased the fraction of sampled/true variance from 0.982 to 1.020. However, the f_S number in the upper-right panel shows that we are now sampling 99.5% of the slope variance, as opposed to 43% for the true spectrum. Having a bit too much total elevation variance seems to be a good tradeoff for being able to model the optically important slope statistics.

The row 2 panel shows random realizations of the 1-D surfaces generated from these two spectra (with the same set of random numbers). The surface elevations $z(x_r)$ from the true spectrum (red line) and the re-scaled spectrum (green line) are almost indistinguishable at the scale of this figure. These significant wave heights for these two surface realizations compare well with the significant wave height of $H_{1/3} = 2.48$ m for a Pierson-Moskowitz spectrum. The row 3 panel shows the surface slopes computed from finite differences of the discrete surface heights. The Cox-Munk along-wind mean square slope is given by $mss_x = 0.0316U_{12.5}$, which compares well with the value of 0.032 obtained with the re-scaled spectrum. However, the value obtained from the true spectrum is only 0.022, or 70% of the Cox-Munk value. The bottom two panels replot the 0-10 m sections of the z and dz/dx figures for better display of the details.

We thus see that when using the $\delta(k)$ correction to a variance spectrum, we can generate a surface that reproduces, to within a few percent of the theoretical values, both the surface elevation and slope statistics that would be obtained from the underlying true variance spectrum if it were sampled with enough points to fully resolve the elevation and slope variances.



cgPlot_EStilde_6panel.pro

Figure 2.7: Example 1-D surfaces and slopes created using true and adjusted variance spectra. See the text for discussion.

CHAPTER 3

Two-Dimensional, Time-Independent Surfaces

We have now had a detailed look at the one-dimensional case. The extension to two dimensions is mathematically straight forward.

Let $z(\mathbf{x}, t) = z(x, y, t)$ be the sea surface elevation in meters at point $\mathbf{x} = (x, y)$ at time t . The spatial extent of the sea surface is $0 \leq x < L_x$ and $0 \leq y < L_y$. This surface is sampled on a rectangular grid of N_x by N_y points, where both N_x and N_y are powers of 2 for the FFT. The spatial sampling points are then

$$x(r) = [0, 1, 2, \dots, N_x - 1] \frac{L_x}{N_x} = r\Delta x, r = 0, \dots, N_x - 1$$

$$y(s) = [0, 1, 2, \dots, N_y - 1] \frac{L_y}{N_y} = s\Delta y, s = 0, \dots, N_y - 1.$$

This spatial sampling frequency gives N_x and N_y spatial frequencies k_x and k_y (in math frequency order)

$$k_x(u) = [-(N_x/2 - 1), \dots, -1, 0, 1, \dots, N_x/2] \frac{2\pi}{L_x} = u\Delta k_x, u = -(N_x/2 - 1), \dots, N_x/2$$

$$k_y(v) = [-(N_y/2 - 1), \dots, -1, 0, 1, \dots, N_y/2] \frac{2\pi}{L_y} = v\Delta k_y, v = -(N_y/2 - 1), \dots, N_y/2$$

As before, the x-dimension 0 frequency is at array element $u = N_x/2 - 1$ and the Nyquist frequency is at element $u = N_x - 1$. Thus the positive and negative pairs of k_x values are related by $k_x(u) = -k_x(N_x - 2 - u)$, $u = 0, \dots, N_x - 2$, with the Nyquist frequency always being a special case because there is only a positive Nyquist frequency. Corresponding relations hold for the y direction.

Let $\mathbf{k} = (k_x, k_y)$ denote a spatial frequency vector, where k_x and k_y are frequencies in the x and y directions, respectively. For discrete values, we write $\mathbf{k}_{uv} = (k_x(u), k_y(v))$. The magnitude of \mathbf{k} is $k = \sqrt{k_x^2 + k_y^2}$. In our (x, y) coordinate system, let the wind blow in the $+x$ direction. The $-x$ direction is then upwind, and the $\pm y$ directions are the cross-wind directions. With this choice, $k_x > 0$ indicates frequencies of waves propagating more or less downwind, and $k_x < 0$ for waves propagating against the wind. $k_x = 0$ and $k_y \neq 0$ indicates a wave propagating at exactly a

cross-wind $\pm y$ direction. The angle of wave propagation relative to the downwind direction for a wave of frequency $(k_x(u), k_y(v))$ is given by

$$\varphi(\mathbf{k}_{uv}) = \varphi(u, v) = \tan^{-1} \left[\frac{k_y(v)}{k_x(u)} \right]. \quad (3.1)$$

We can thus write the 2-D surface as $z(\mathbf{x})$, its Fourier amplitude as $\hat{z}(\mathbf{k})$, and the associated variance spectrum as $\Psi(\mathbf{k})$. The discrete variance spectrum $\Psi(\mathbf{k}_{uv}) = \Psi(u, v)$ gives the variance of the wave with wavelength $2\pi/k_{uv}$ propagating in direction $\varphi(\mathbf{k}_{uv})$ relative to the downwind direction. See §B.1 for discussion of two-dimensional variance spectra.

3.1 Surfaces to Variance Spectra

Even though the mathematical transition from one to two dimensions causes no problems, it is again educational to take a careful look at a couple of contrived examples.

3.1.1 Example: A Random Sea Surface

For the first example, a sea surface area of size $L_x \times L_y = 10 \times 10$ m is sampled using $N_x = 16$ and $N_y = 8$ points in the x and y directions. Sea surface elevations $z(x, y)$ were created by drawing a $\mathcal{N}(0, 1)$ random number at each (x, y) value.

The upper-left panel of Fig. 3.1 shows a contour plot of this surface for a particular sequence of random numbers. The spatial periodicity of the Fourier representation is used to extend the contour plot to the full (L_x, L_y) range of the tile, which gives a good visual appearance. Thus the elevations at $x = L_x$ are the same as those at $x = 0$, those at $y = L_y$ equal those at $y = 0$, and the elevation at (L_x, L_y) duplicates the elevation at $(0, 0)$. The $N_x \times N_y$ grid of sample points is shown by the solid silver dots. The points at $x = L_x$ and $y = L_y$ obtained by periodicity are shown by open silver circles at the right and top of the surface plot.

The Fourier amplitudes $\hat{z}(\mathbf{k}_{uv}) = \hat{z}(k_x(u), k_y(v))$ are obtained from the 2-D DFT of $z(\mathbf{x}_{rs}) = z(x(r), y(s))$:

$$\hat{z}(\mathbf{k}_{uv}) = \mathfrak{D}\{z(\mathbf{x}_{rs})\}.$$

The usual warning on FFT frequency order applies here. The 2-D FFT gives back the amplitudes $\hat{z}(\mathbf{k}_{uv})$ with the array elements corresponding to the the FFT frequency order of Eq. (2.8). Before plotting, the $\hat{z}(\mathbf{k}_{uv})$ array elements must be shifted into the math frequency order in both the k_x and k_y array directions using the appropriate shift function for the computer language used in the plotting. For the IDL routine used to generate Fig. 3.1, the 2-D circular shift is given by the IDL command

```
realzhatplot = SHIFT(REAL_PART(zhat), Nx/2 - 1, Ny/2 - 1)
```

where $zhat$ is the complex 2-D array returned by the FFT routine, and $realzhatplot$ is the 2-D array plotted in the upper right panel of Fig. 3.1.

The upper right panel of Fig. 3.1 plots the real part of $\hat{z}(\mathbf{k}_{uv})$, and the lower left panel plots the imaginary part. The Nyquist frequency $k_x^{Ny} = 2\pi/(2\Delta x) = 5.03$ rad/m lies along the right side of the contour plot. The Nyquist frequency $k_y^{Ny} = 2\pi/(2\Delta y) = 2.51$ rad/m lies along the top of the contour plot. The white space at the left and bottom highlights that there are no

$$z(x,y) \sim N(0,1); N_x, N_y = 16, 8$$

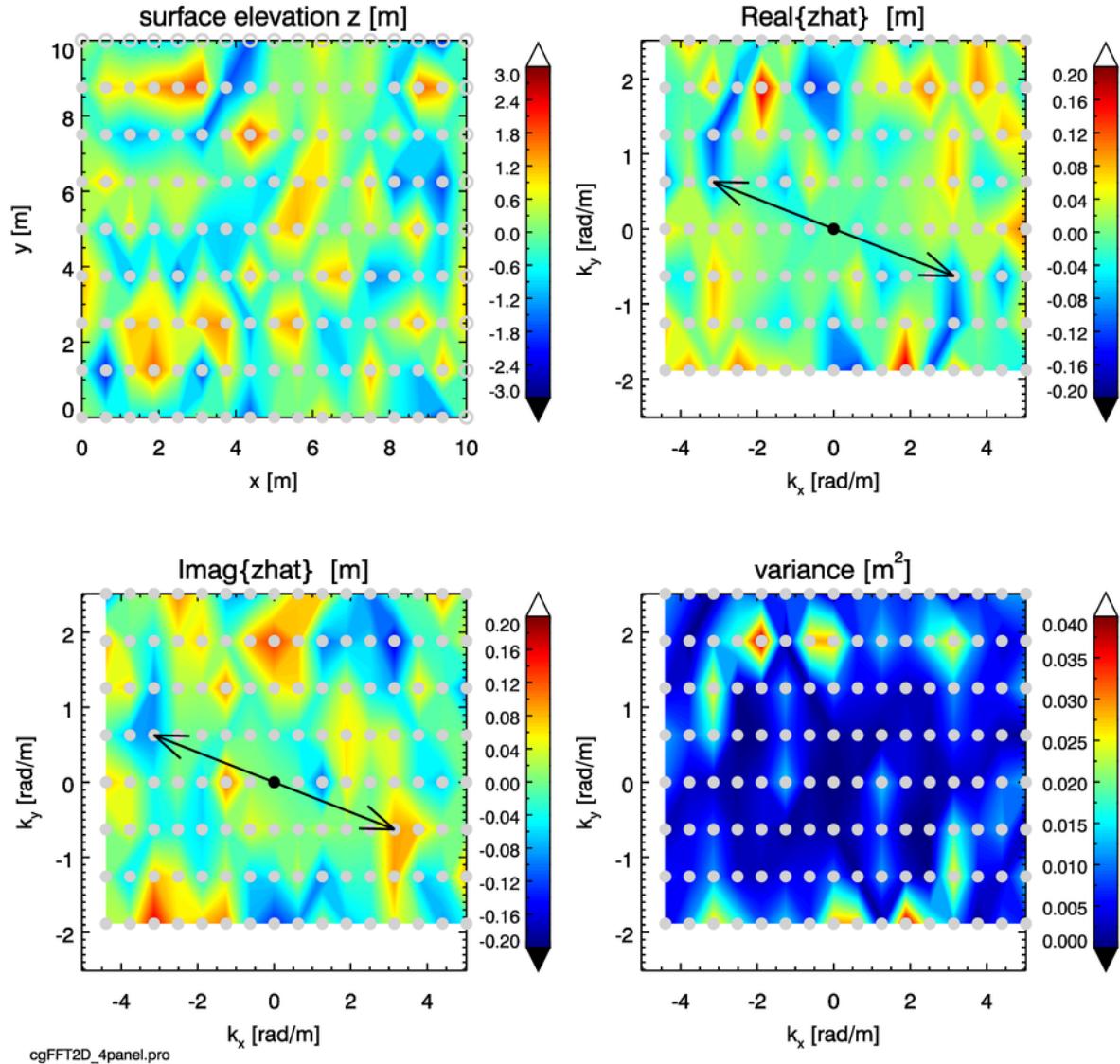


Figure 3.1: A two-dimensional random sea surface and its Fourier transform and two-sided variance spectrum. The surface elevations in the upper left panel were randomly drawn from a $\mathcal{N}(0, 1)$ distribution. The black arrows highlight a particular $\pm \mathbf{k}_{uv}$ frequency pair. The blue-blue and red-red symmetry of the real part of the Fourier amplitudes, and the red-blue symmetry of the imaginary part, shows the Hermitian nature of the amplitudes. The gray dots show the locations of the discrete values that were contoured to create the figures.

negative Nyquist frequencies. In each amplitude plot a particular pair of $\pm \mathbf{k}_{uv}$ values is indicated by the black arrows; the $(0, 0)$ frequency is shown by a black dot. Note that in the plot of the real part, $Re\{\hat{z}(-\mathbf{k}_{uv})\} = Re\{\hat{z}(+\mathbf{k}_{uv})\}$, whereas in the plot of the imaginary part, $Im\{\hat{z}(-\mathbf{k}_{uv})\} = -Im\{\hat{z}(+\mathbf{k}_{uv})\}$. The contouring is rather low quality for so few points, but it is easy to see in the digital output that when the k_x is the Nyquist frequency (the points along the right column of points in the plot), the symmetries are given by $Re\{\hat{z}(k_x^{\text{Ny}}, -ky)\} = Re\{\hat{z}(k_x^{\text{Ny}}, +ky)\}$ and $Im\{\hat{z}(k_x^{\text{Ny}}, -ky)\} = -Im\{\hat{z}(k_x^{\text{Ny}}, +ky)\}$. A corresponding relation holds for $\pm k_x$ when $k_y = k_y^{\text{Ny}}$ (the points along the top row of the plot). The point at the upper right of the plot corresponds to both k_x and k_y being at their respective Nyquist frequencies. As always, the array elements at the Nyquist frequencies must be treated as special cases when writing computer programs. These symmetries show that the 2-D amplitudes are Hermitian: $\hat{z}^*(-\mathbf{k}_{uv}) = \hat{z}(\mathbf{k}_{uv})$. The discrete 2-D variance spectrum $\Psi(\mathbf{k}_{uv}) = \Psi(u, v) = \hat{z}(u, v)\hat{z}^*(u, v)$ is contoured in the lower right panel of the figure. Note the $\Psi(-\mathbf{k}_{uv}) = \Psi(\mathbf{k}_{uv})$ symmetry.

For this example, the standard check on Parseval's relation (1.23) gives

$$\sum_{r=0}^{N_x-1} \sum_{s=0}^{N_y-1} |z(r, s)|^2 = N_x N_y \sum_{u=0}^{N_x-1} \sum_{v=0}^{N_y-1} |\hat{z}(u, v)|^2 = N_x N_y \sum_{u=0}^{N_x-1} \sum_{v=0}^{N_y-1} \Psi(u, v) = 130.90 \text{ m}^2.$$

In all of these plots, it should be remembered that the discrete values are known only at the locations of the silver dots. The contouring routine simply interpolates between these points to create a visually appealing figure. The continuous color in the plots does not imply that the values are continuous and known in between the discrete points.

3.1.2 Example: A Sea Surface of Crossing Sinusoids

For a second example, let us define a surface from a pair of crossing sinusoids as follows:

$$z(x_r, y_s) = A_1 \cos(k_{x1}x_r + k_{y1}y_s + \phi_1) + A_2 \cos(k_{x2}x_r + k_{y2}y_s + \phi_2), \quad (3.2)$$

where as usual $r = 0, \dots, N_x - 1; s = 0, \dots, N_y - 1$; and where

$A_1 = 1.0$ m is the amplitude of the first wave

$N_{x1} = 2$ is the number of wave lengths in the x direction in $[0, L_x]$ for the first wave

$k_{x1} = 2.0\pi N_{x1}/L_x = 1.257$ rad/m² is the k_x frequency of the first wave

$N_{y1} = 1$ is the number of wave lengths in the y direction in $[0, L_y]$ for the first wave

$k_{y1} = 2.0\pi N_{y1}/L_y = 0.628$ rad/m² is the k_y frequency of the first wave

$\phi_1 = 0$ is the phase of the first wave; 0 gives a cosine wave

$A_2 = 0.5$ m is the amplitude of the second wave

$N_{x2} = 4$ is the number of wave lengths in the x direction in $[0, L_x]$ for the second wave

$k_{x2} = 2.0\pi N_{x2}/L_x = 2.513$ rad/m² is the k_x frequency of the second wave

$N_{y2} = 3$ is the number of wave lengths in the y direction in $[0, L_y]$ for the second wave

$k_{y2} = -2.0\pi N_{y2}/L_y = -1.885$ rad/m² is the k_y frequency of the second wave

$\phi_2 = \pi/2$ is the phase of the second wave; $\pi/2$ gives a sine wave

The wavelength of the first wave is $\Lambda_1 = 2\pi/\sqrt{k_{x1}^2 + k_{y1}^2} = 4.47$ m, and that of the second wave is $\Lambda_2 = 2.00$ m. The direction of propagation of the first wave relative to the $+x$ axis is

$$\varphi_1 = \tan^{-1} \left(\frac{k_{y1}}{k_{x1}} \right) = 26.57 \text{ deg},$$

and the direction of propagation of the second wave relative to the $+x$ axis is

$$\varphi_2 = \tan^{-1} \left(\frac{k_{y2}}{k_{x2}} \right) = -36.87 \text{ deg}.$$

The upper left panel of Fig. 3.2 shows this surface elevation pattern when Eq. (3.2) is sampled with $N_x = N_y = 16$. The dominant red-blue pattern shows the first wave oriented with the direction of propagation along either the $+\mathbf{k}_1$ direction at $\varphi_1 = 26.57$ deg or the $-\mathbf{k}_1$ direction at $\varphi_1 = 26.57 + 180 = 206.57$ deg. We cannot of course determine the actual direction $+\mathbf{k}_1$ or $-\mathbf{k}_1$ of propagation from sea surface elevations at a single time. The dominant wave pattern is modulated by the second wave, which has one half the amplitude of the first wave.

The choice above of $\phi_1 = 0$ in Eq. (3.2) makes the dominate wave a cosine in our coordinate system, and $\phi_2 = \pi/2$ makes the second wave a sine. As we saw in the 1-D examples, variance associated with cosine waves appears in the real part of the Fourier amplitudes, and the variance in sine waves appears in the imaginary parts. We see this again here for the first (cosine) and second (sine) waves of the surface. Since each wave pattern has only one frequency, there is only one pair of points at $\pm\mathbf{k}_1$ in the plot of the real part, and one pair at $\pm\mathbf{k}_2$ in the plot of the imaginary part. The amplitudes at all other frequencies are zero. The symmetries of these points again show the Hermitian nature of the amplitudes.

The lower right panel shows the two-sided variance function $\Psi_{2s}(\mathbf{k})$ for this surface. Note that the first wave has four times the variance of the second wave because the amplitude of the first wave is twice that of the second wave. $\Psi_{2s}(-\mathbf{k}) = \Psi_{2s}(+\mathbf{k})$. Note also that you can look at a 2-D variance spectrum and see how much energy is propagating in a given $\pm\mathbf{k}$ direction.

For this simple example involving just four frequencies it is easy (from the digital output) to hand check that the right-hand side of Parseval's relation (1.23) is

$$N_x N_y \sum_u \sum_v |\hat{z}(u, v)|^2 = 16 \cdot 16 [(0.5)^2 + (0.5)^2 + (0.25)^2 + (-0.25)^2] = 160 \text{ m}^2.$$

This value agrees exactly with the corresponding sums of the surface elevations, $\sum_r \sum_s [z(r, s)]^2$, and variance values, $N_x N_y \sum_u \sum_v \Psi(u, v)$.

$z(x,y) = 2 \text{ sinusoids}; N_x, N_y = 16, 16$

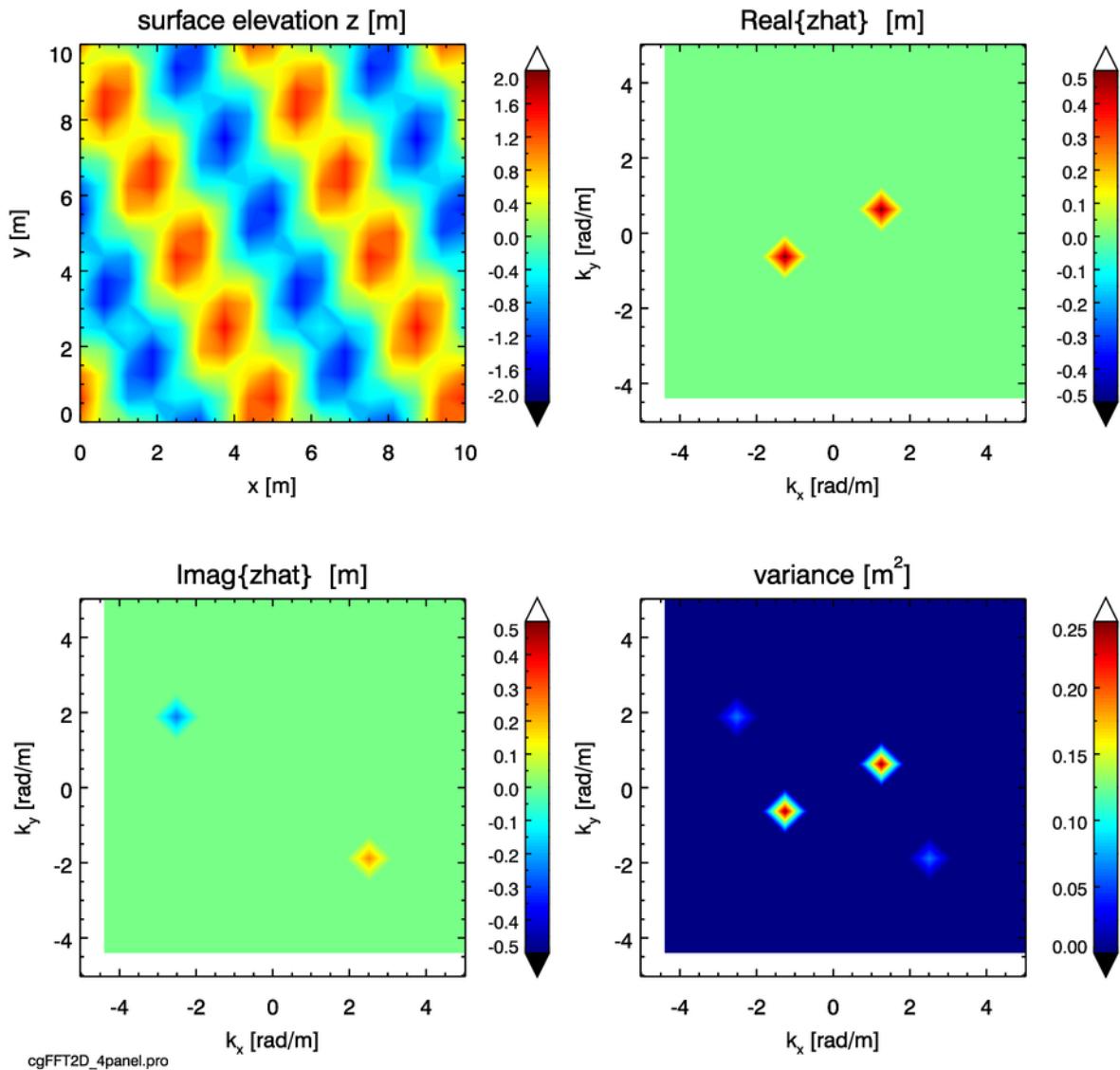


Figure 3.2: A two-dimensional sea surface composed of two crossing sinusoids, and the resulting Fourier amplitudes and variance.

3.2 Variance Spectra to Surfaces

Now, again, we face the reverse process: start with a two-dimensional, one-sided variance density spectrum and generate a random 2-D realization of a sea surface. The first task is to conjure up a two-dimensional variance density spectrum, which requires some effort.

3.2.1 Theory

Let $\Psi(\mathbf{k}) = \Psi(k_x, k_y)$ denote a 2-D elevation variance spectrum. This spectrum has units of $\text{m}^2/(\text{rad}/\text{m})^2$. By definition, the integral of $\Psi(k_x, k_y)$ over all frequencies gives the elevation variance:

$$\text{var}\{z\} = \langle z^2 \rangle = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \Psi(k_x, k_y) dk_x dk_y .$$

(See §B.1 for the precise definition of $\Psi(k_x, k_y)$ and for further details.) As in the 1-D case of §2.5, $\langle \cdot \rangle$ indicates expectation or ensemble average over many measurements of the sea surface.

In the discrete case, the \mathbf{k}_{uv} values coming out of $\hat{z} = \mathfrak{D}\{z\}$ point both “downwind” (positive $k_x(u)$ values) and “upwind” (negative $k_x(u)$ values), i.e. there are both positive and negative frequencies represented in the “two-sided” variance spectrum, denoted as $\Psi_{2s}[u, v], u = -(N_x/2 - 1), \dots, N_x/2, v = -(N_y/2 - 1), \dots, N_y/2$ as above. In general, $\Psi_{2s}(-\mathbf{k}_{uv}) \neq \Psi_{2s}(+\mathbf{k}_{uv})$ because more energy propagates downwind than upwind at a given frequency. $k_x(N_x/2)$ is the Nyquist frequency for waves propagating in the x direction; $k_y(N_y/2)$ is the Nyquist frequency for waves propagating in the y direction. $\Psi_{2s}(0, 0)$ is the variance at zero frequencies, i.e., the variance of the mean sea surface; this term is normally set to 0 so that the mean sea surface is at height 0.

The 2-D equivalents of Eqs. (2.9) and (2.10) are

$$\begin{aligned} \hat{z}_o(\mathbf{k}_{uv}) &\equiv \frac{1}{\sqrt{2}} [\rho(\mathbf{k}_{uv}) + i\sigma(\mathbf{k}_{uv})] \sqrt{\frac{\Psi_{1s}(\mathbf{k} = \mathbf{k}_{uv})}{2} \Delta k_x \Delta k_y} \\ &= \frac{1}{\sqrt{2}} [\rho(\mathbf{k}_{uv}) + i\sigma(\mathbf{k}_{uv})] \sqrt{\Psi_{2s}(\mathbf{k}_{uv})} . \end{aligned} \quad (3.3)$$

Here, as before, $\rho(\mathbf{k}_{uv})$ and $\sigma(\mathbf{k}_{uv})$ are independent $\mathcal{N}(0, 1)$ random variables, with a different random variable drawn for each \mathbf{k}_{uv} value. As in the 1-D case, the expected value of $\hat{z}_o(\mathbf{k}_{uv})$ gives back whatever variance spectrum is used for $\Psi_{2s}(\mathbf{k}_{uv})$, but is not Hermitian. As before, the notation of Eq. (3.3) distinguishes between the value of the continuous spectral density function evaluated at a discrete frequency value, $\Psi_{1s}(\mathbf{k} = \mathbf{k}_{uv})$, and the discrete variance point function, $\Psi_{2s}(\mathbf{k}_{uv})$.

We must define random Hermitian amplitudes for use in the inverse Fourier transform. Looking ahead to the next chapter, which extends the results of this chapter to time-dependent surfaces, define the time-dependent spectral amplitude

$$\hat{z}(\mathbf{k}_{uv}, t) \equiv \frac{1}{\sqrt{2}} [\hat{z}_o(\mathbf{k}_{uv}) e^{-i\omega_{uv}t} + \hat{z}_o^*(-\mathbf{k}_{uv}) e^{+i\omega_{uv}t}] . \quad (3.4)$$

This function is clearly Hermitian, so the inverse DFT applied to $\hat{z}(\mathbf{k}_{uv}, t)$ will give a real-valued $z(\mathbf{x}_{rs}, t)$. Recall from Eq. (1.1) that a function of the form $\cos(kx - \omega t)$ gives a wave propagating in the $+x$ direction. The corresponding $\hat{z}_o(\mathbf{k}_{uv}) e^{-i\omega t}$ in this equation gives a wave propagating in the $+\mathbf{k}$ direction, which is in the downwind half plane of all directions. In general the temporal angular

frequency ω_{uv} is a function of the spatial frequency k_{uv} . For example, for deep-water gravity waves, $\omega_{uv}^2 = gk_{uv}$.

For simplicity of notation, let us momentarily drop the rs and uv subscripts on the discrete variables. The $\hat{z}(\mathbf{k}, t)$ of Eq. (3.4) is also consistent with the variance spectrum:

$$\begin{aligned}
\langle |\hat{z}(\mathbf{k}, t)|^2 \rangle &= \langle \hat{z}(\mathbf{k}, t) \hat{z}^*(\mathbf{k}, t) \rangle \\
&= \left\langle \frac{1}{\sqrt{2}} \left[\frac{1}{\sqrt{2}} [\rho(\mathbf{k}) + i\sigma(\mathbf{k})] \sqrt{\Psi_{2s}(\mathbf{k})} e^{i\omega t} + \frac{1}{\sqrt{2}} [\rho(-\mathbf{k}) - i\sigma(-\mathbf{k})] \sqrt{\Psi_{2s}(-\mathbf{k})} e^{-i\omega t} \right] \times \right. \\
&\quad \left. \frac{1}{\sqrt{2}} \left[\frac{1}{\sqrt{2}} [\rho(\mathbf{k}) - i\sigma(\mathbf{k})] \sqrt{\Psi_{2s}(\mathbf{k})} e^{-i\omega t} + \frac{1}{\sqrt{2}} [\rho(-\mathbf{k}) + i\sigma(-\mathbf{k})] \sqrt{\Psi_{2s}(-\mathbf{k})} e^{i\omega t} \right] \right\rangle \\
&= \frac{1}{4} \left\langle [\rho^2(\mathbf{k}) - i\rho(\mathbf{k})\sigma(\mathbf{k}) + i\sigma(\mathbf{k})\rho(\mathbf{k}) + \sigma^2(\mathbf{k})] \Psi_{2s}(\mathbf{k}) + \right. \\
&\quad [\rho(\mathbf{k})\rho(-\mathbf{k}) + i\rho(\mathbf{k})\sigma(-\mathbf{k}) + i\sigma(\mathbf{k})\rho(-\mathbf{k}) - \sigma(\mathbf{k})\sigma(-\mathbf{k})] \sqrt{\Psi_{2s}(\mathbf{k})} \sqrt{\Psi_{2s}(-\mathbf{k})} e^{i2\omega t} + \\
&\quad [\rho(-\mathbf{k})\rho(\mathbf{k}) - i\rho(-\mathbf{k})\sigma(\mathbf{k}) - i\sigma(-\mathbf{k})\rho(\mathbf{k}) - \sigma(-\mathbf{k})\sigma(\mathbf{k})] \sqrt{\Psi_{2s}(-\mathbf{k})} \sqrt{\Psi_{2s}(\mathbf{k})} e^{-i2\omega t} + \\
&\quad \left. [\rho^2(-\mathbf{k}) + i\rho(-\mathbf{k})\sigma(-\mathbf{k}) - i\sigma(-\mathbf{k})\rho(-\mathbf{k}) + \sigma^2(-\mathbf{k})] \Psi_{2s}(-\mathbf{k}) \right\rangle \\
&= \frac{1}{2} [\Psi_{2s}(\mathbf{k}) + \Psi_{2s}(-\mathbf{k})].
\end{aligned}$$

Here we have noted that all terms like $\langle \rho(\mathbf{k})\rho(-\mathbf{k}) \rangle$ are zero because of the independence of the random variables for different \mathbf{k} values, as are terms like $\langle \rho(\mathbf{k})\sigma(\mathbf{k}) \rangle$. The remaining term is the total variance associated with waves propagating in the downwind and upwind directions at the spatial frequency of magnitude k . It should be noted that this term is independent of time even though the waves $z(x, y, t)$ depend on time. This is because the total variance (or energy) of the wave field is the same at all times, even though the exact shape of the sea surface varies with time.

If only a ‘snapshot’ of the sea surface at one time is available, it is not possible to resolve how much of the total variance is associated with waves propagating in direction \mathbf{k} compared to the opposite direction $-\mathbf{k}$. The forward DFT, $\hat{z}(\mathbf{k}, t) = \mathcal{D}\{z(x, y, t)\}$, and $\hat{z}(\mathbf{k}, t)\hat{z}^*(\mathbf{k}, t)$ then gives $\Psi_{2s}(-\mathbf{k}) = \Psi_{2s}(\mathbf{k})$, in which case the last equation reduces to

$$\langle \hat{z}(\mathbf{k}, t) \hat{z}^*(\mathbf{k}, t) \rangle = \Psi_{2s}(\mathbf{k}).$$

In any case, the amplitudes defined by Eq. (3.4) are Hermitian, so that the real part of the inverse Fourier transform $Z(\mathbf{x}, t) = \mathcal{D}^{-1}\{\hat{z}(\mathbf{k}, t)\}$ gives a real-valued sea surface $z(\mathbf{x}, t)$. That sea surface is consistent with the variance spectrum $\Psi_{2s}(\mathbf{k})$ at every time t . Wave variance (energy) is thus conserved in a “round-trip” calculation from variance spectrum to sea surface and back to variance spectrum. In the time-dependent case, if $\Psi_{2s}(\mathbf{k}) > \Psi_{2s}(-\mathbf{k})$, then more variance will be contained in waves propagating in the $+\mathbf{k}$ direction than in the $-\mathbf{k}$ direction. This is all that we can ask from Fourier transform techniques.

Although Eqs. (3.3) and (3.4) are compact representations of the random spectral amplitudes, the actual evaluation of these equations in a computer program warrants further examination. In particular, the Nyquist frequencies are always special cases because there is only a positive Nyquist frequency, $k_x^{\text{Ny}} = \frac{N_x}{2} \frac{2\pi}{L_x}$ at array element $N_x - 1$, with a corresponding equation for the Nyquist

frequency in the y direction. Writing $e^{\pm i\omega t} = \cos[\omega(k)t] \pm i \sin[\omega(k)t]$ and expanding Eq. (3.4) gives

$$\begin{aligned} 2\hat{z}(\mathbf{k}, t) &= \left[\rho(\mathbf{k})\sqrt{\Psi_{2s}(\mathbf{k})} + \rho(-\mathbf{k})\sqrt{\Psi_{2s}(-\mathbf{k})} \right] \cos[\omega(k)t] \\ &\quad - \left[\sigma(\mathbf{k})\sqrt{\Psi_{2s}(\mathbf{k})} + \sigma(-\mathbf{k})\sqrt{\Psi_{2s}(-\mathbf{k})} \right] \sin[\omega(k)t] \\ &\quad + i \left\{ \left[\rho(\mathbf{k})\sqrt{\Psi_{2s}(\mathbf{k})} - \rho(-\mathbf{k})\sqrt{\Psi_{2s}(-\mathbf{k})} \right] \sin[\omega(k)t] \right. \\ &\quad \left. + \left[\sigma(\mathbf{k})\sqrt{\Psi_{2s}(\mathbf{k})} - \sigma(-\mathbf{k})\sqrt{\Psi_{2s}(-\mathbf{k})} \right] \cos[\omega(k)t] \right\}. \end{aligned} \quad (3.5)$$

These terms are all $N_x \times N_y$ arrays, but note that terms like $\rho(\mathbf{k})\sqrt{\Psi_{2s}(\mathbf{k})}$ represent element-by-element multiplications, not matrix multiplications.

For a particular array element $\hat{z}(k_x(u), k_y(v), t) = \hat{z}(u, v, t)$, and using the previously noted indexing relation $k(u) = -k(N - 2 - u)$, $u = 0, \dots, N - 2$ for frequencies written in math order, we can write

$$\begin{aligned} &2\hat{z}(u, v, t) \\ &= \left[\rho(u, v)\sqrt{\Psi_{2s}(u, v)} + \rho(N_x - 2 - u, N_y - 2 - v)\sqrt{\Psi_{2s}(N_x - 2 - u, N_y - 2 - v)} \right] \cos[\omega(k(u, v))t] \\ &\quad - \left[\sigma(u, v)\sqrt{\Psi_{2s}(u, v)} + \sigma(N_x - 2 - u, N_y - 2 - v)\sqrt{\Psi_{2s}(N_x - 2 - u, N_y - 2 - v)} \right] \sin[\omega(k(u, v))t] \\ &\quad + i \left\{ \left[\rho(u, v)\sqrt{\Psi_{2s}(u, v)} - \rho(N_x - 2 - u, N_y - 2 - v)\sqrt{\Psi_{2s}(N_x - 2 - u, N_y - 2 - v)} \right] \sin[\omega(k(u, v))t] \right. \\ &\quad \left. + \left[\sigma(u, v)\sqrt{\Psi_{2s}(u, v)} - \sigma(N_x - 2 - u, N_y - 2 - v)\sqrt{\Psi_{2s}(N_x - 2 - u, N_y - 2 - v)} \right] \cos[\omega(k(u, v))t] \right\}. \end{aligned} \quad (3.6)$$

This equation allows for efficient computation within loops over array elements. In particular, the code can loop over the non-positive $k_x(u)$ values, $u = 0, \dots, N_x/2$, and over all $k_y(v)$ values, $v = 0, \dots, N_y - 2$ to evaluate the amplitudes for all non-Nyquist frequencies. The positive $k_x(u)$ values are then obtained from the negative $k_x(u)$ values by Hermitian symmetry. The Nyquist frequencies are evaluated by an equation of the same form, but with one or the other index held fixed (e.g., $u = N_x - 1$ while $v = 0, \dots, N_y - 2$). The amplitude $\hat{z}(k_x = 0, k_y = 0)$ at array element $(u, v) = (N_x/2 - 1, N_y/2 - 1)$ is usually set to zero, corresponding to the mean sea level being set to 0. For generation of time-independent surfaces, we can set $t = 0$ so that the cosines are one and the sines are zero, which cuts the number of terms to be evaluated in half.

If the frequencies are in the FFT order, then the last equation has the same general form, but the indexing that expresses Hermitian symmetry is different: $k(u) = -k(N - u)$, $u = 1, \dots, N - 1$,

with $k = 0$ being a special case. The equation corresponding to Eq. (3.6) is then

$$\begin{aligned}
& 2 \hat{z}(u, v, t) \\
&= \left[\rho(u, v) \sqrt{\Psi_{2s}(u, v)} + \rho(N_x - u, N_y - v) \sqrt{\Psi_{2s}(N_x - u, N_y - v)} \right] \cos[\omega(k(u, v))t] \\
&- \left[\sigma(u, v) \sqrt{\Psi_{2s}(u, v)} + \sigma(N_x - u, N_y - v) \sqrt{\Psi_{2s}(N_x - u, N_y - v)} \right] \sin[\omega(k(u, v))t] \\
&+ i \left\{ \left[\rho(u, v) \sqrt{\Psi_{2s}(u, v)} - \rho(N_x - u, N_y - v) \sqrt{\Psi_{2s}(N_x - u, N_y - v)} \right] \sin[\omega(k(u, v))t] \right. \\
&\left. + \left[\sigma(u, v) \sqrt{\Psi_{2s}(u, v)} - \sigma(N_x - u, N_y - v) \sqrt{\Psi_{2s}(N_x - u, N_y - v)} \right] \cos[\omega(k(u, v))t] \right\}. \quad (3.7)
\end{aligned}$$

However, this equation does not explicitly show the special cases. Let $\text{zhat}[u, v]$ be the array of $\hat{z}(\mathbf{k}, t) = \hat{z}(k_x(u), k_y(v), t)$ values at a particular time t . $\text{r}[u, v]$ and $\text{s}[u, v]$ are the arrays of random numbers $\rho(u, v)$ and $\sigma(u, v)$, respectively. $\text{Psiroot}[u, v]$ is $\frac{1}{2}\sqrt{\Psi_{2s}(k_x(u), k_y(v))}$ (incorporating the 2 seen on the left-hand side of Eq. (3.7)). With other obvious definitions, the pseudo code to evaluate Eq. (3.7) at a particular time t is as follows:

(1) Loop over the non-zero frequencies:

```

LOOP OVER u=1, Nx-1
LOOP OVER v=1, Ny-1
zhat[u, v] = COMPLEX(
    ( r[u, v] * Psiroot[u, v]
      + r[Nx-u, Ny-v] * Psiroot[Nx-u, Ny-v] ) * cosomegat[u, v]
    - ( s[u, v] * Psiroot[u, v]
      + s[Nx-u, Ny-v] * Psiroot[Nx-u, Ny-v] ) * sinomegat[u, v] ,
    ( r[u, v] * Psiroot[u, v]
      - r[Nx-u, Ny-v] * Psiroot[Nx-u, Ny-v] ) * sinomegat[u, v]
    + ( s[u, v] * Psiroot[u, v]
      - s[Nx-u, Ny-v] * Psiroot[Nx-u, Ny-v] ) * cosomegat[u, v] )
END v LOOP
END u LOOP

```

(2) Loop over all k_y values for the special case of $k_x = 0$ at frequency array index $u = 0$. Note that for $k_x = 0$, the $\pm k_y$ values are complex conjugates.

```

LOOP OVER v=1, Ny/2
zhat(0, v) = COMPLEX(
    ( r(0, v) * Psiroot[0, v]
      + r[0, Ny-v] * Psiroot[0, Ny-v] ) * cosomegat[0, v]
    - ( s[0, v] * Psiroot[0, v]
      + s[0, Ny-v] * Psiroot[0, Ny-v] ) * sinomegat[0, v] ,
    ( r[0, v] * Psiroot[0, v]
      - r[0, Ny-v] * Psiroot[0, Ny-v] ) * sinomegat[0, v]
    + ( s[0, v] * Psiroot[0, v]
      - s[0, Ny-v] * Psiroot[0, Ny-v] ) * cosomegat[0, v] )

```

```

        - r[0,Ny-v]*Psiroot[0,Ny-v] )*sinomegat[0,v]
        + ( s[0,v]*Psiroot[u,v]
        - s[0,Ny-v]*Psiroot[0,Ny-v] )*cosomegat[0,v] )
zhat(0,Ny-v) = CONJ( zhat(0,v) )
END u LOOP

```

(3) Loop over all k_x values for the special cases of $k_y = 0$ at frequency array index $v = 0$. Note that for $k_y = 0$, the $\pm k_x$ values are complex conjugates.

```

LOOP OVER u=1,Nx/2
zhat(u,0) = COMPLEX(
    ( r(u,0)*Psiroot[u,0]
    + r[Nx-u,0]*Psiroot[Nx-u,0] )*cosomegat[u,0]
    - ( s[u,0]*Psiroot[u,0]
    + s[Nx-u,0]*Psiroot[Nx-u,0] )*sinomegat[u,0] ,
    ( r[u,0]*Psiroot[u,0]
    - r[Nx-u,0]*Psiroot[Nx-u,0] )*sinomegat[u,0]
    + ( s[u,0]*Psiroot[u,0]
    - s[Nx-u,0]*Psiroot[Nx-u,0] )*cosomegat[u,0] )
zhat(Nx-u,0) = CONJ( zhat(u,0) )
END v LOOP

```

(4) Finally, set the $(k_x, k_y) = (0, 0)$ value to 0, which sets the mean sea level to zero:

```
zhat[0,0] = COMPLEX(0.0, 0.0)
```

Array $\text{zhat}[u,v] = \hat{z}(k_x(u), k_y(v))$ as just defined is Hermitian and has the frequencies in FFT order ready for input to an FFT routine.

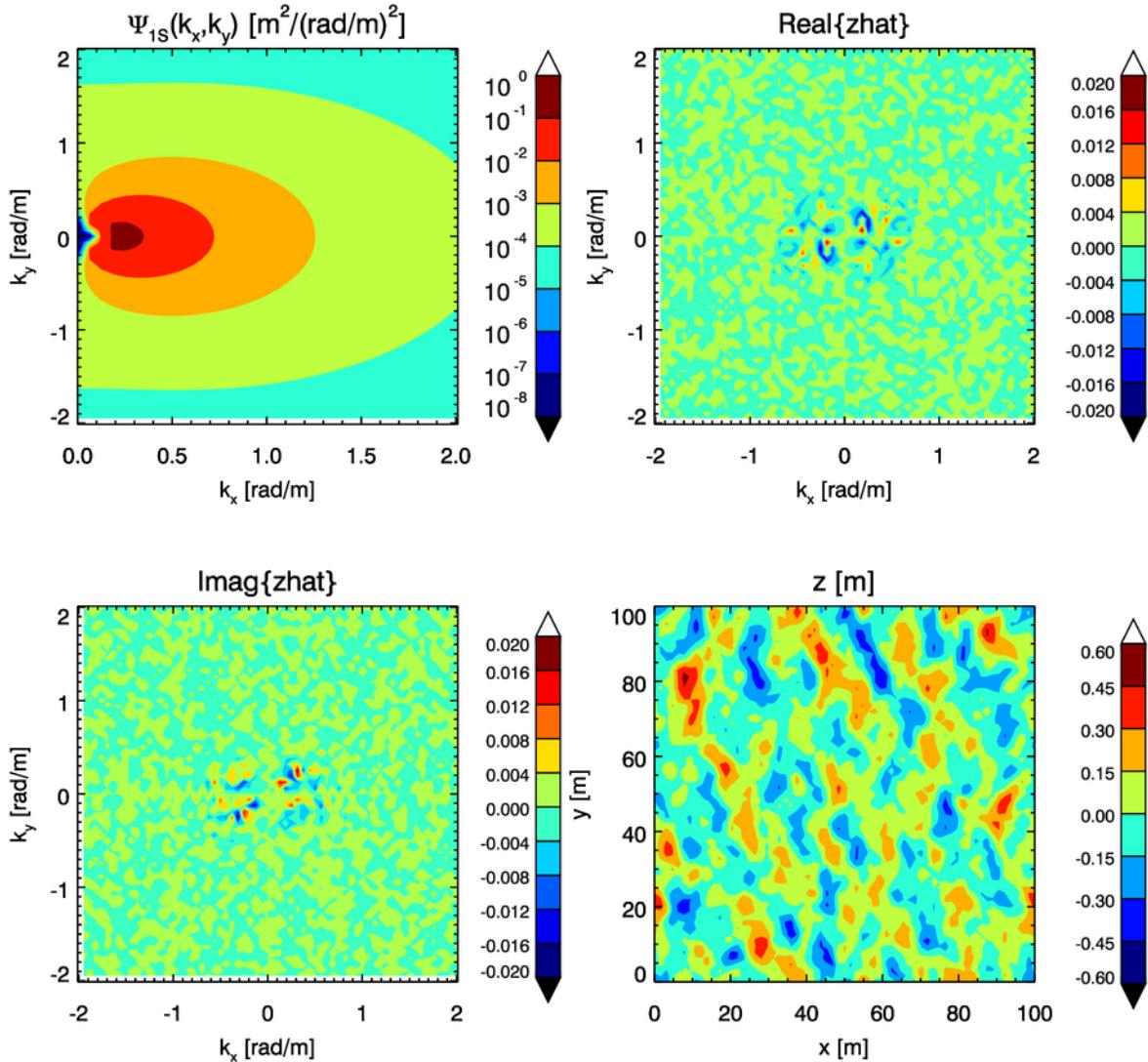
3.2.2 Example: A Two-Dimensional Sea Surface

As an example, let us use a coarse grid sampling of $N_x \times N_y = 64 \times 64$ points, which makes it easier to see certain features in the associated plots. The physical region to be simulated is $L_x \times L_y = 100 \times 100$ m. The two-dimensional, one-sided variance spectrum of Elfouhaily et al. (1997) is used; the defining equations are given in §B.3.

The upper left plot of Fig. 3.3 shows this spectrum for a wind speed of 5 m s⁻¹ and a fully developed sea state. The contours are of $\Psi_{1s}(k_x, k_y)$ evaluated at the discrete grid points; the $\frac{1}{2}\Delta k_x \Delta k_y$ factor seen in Eq. (3.3) has not yet been applied to create a discrete two-sided spectrum. The line along $k_y = 0$ corresponds to the spectrum plotted in Fig. B.3. We see in both plots that for a 5 m s⁻¹ wind, the spectrum peaks near 0.2 – 0.3 rad/m.

The plots of the real and imaginary parts of $\hat{z}(k_x(u), k_y(v))$ show that most of the variance is at low frequencies and that the amplitudes have the Hermitian symmetry illustrated previously in Figs. 3.1 and 3.2. The lower right panel of the plot shows a contour plot of the sea surface generated from the inverse FFT of the amplitudes. The significant wave height for this surface realization is 0.60 m, in good agreement with the expected value given above for the Pierson-Moskowitz spectrum, which is similar to the Elfouhaily et al. (1997) spectrum in the gravity wave region.

$$U_{10} = 5 \text{ m s}^{-1}; (L_x, L_y) = (100, 100); (N_x, N_y) = (64, 64)$$



`cgGenerate_2D_SeaSurface.pro`

Figure 3.3: Example two-dimensional sea surface generated from a 2-D, one-sided variance spectrum. The resolution is only 64×64 grid points, so as to make the features of the underlying spectrum and the Fourier amplitudes easier to see.

3.3 Effect of the Spreading Function

Figure 3.3 shows a contour plot of a two-dimensional, one-sided variance spectrum $\Psi_{1s}(k_x, k_y)$ and a contour plot of a random surface generated from that variance spectrum. A particular spreading function is implicitly contained in that two-dimensional variance spectrum. The effect on the generated sea surface of the spreading function contained within $\Psi_{1s}(k_x, k_y)$ warrants discussion.

As discussed in Appendix B, a 2-D variance spectrum is usually partitioned as (see Eq. B.6)

$$\Psi(k_x, k_y) = \frac{1}{k} \mathcal{S}(k) \Phi(k, \varphi) \equiv \Psi(k, \varphi).$$

Here $\mathcal{S}(k)$ is the omnidirectional spectrum discussed in Chapter 2, and $\Phi(k, \varphi)$ is a nondimensional spreading function, which shows how waves of different frequencies propagate (or “spread out”) relative to the downwind direction at $\varphi = 0$.

One commonly used family of spreading functions is given by the “cosine-2S” functions (Longuet-Higgins et al., 1963), which have the form

$$\Phi(k, \varphi) = C_S \cos^{2S}(\varphi/2), \quad (3.8)$$

where S is a spreading parameter that in general depends on k , wind speed, and wave age. C_S is a normalizing coefficient that gives

$$\int_0^{2\pi} \Phi(k, \varphi) d\varphi = 1 \quad (3.9)$$

for all k . The cosine-2S functions are asymmetric, with much stronger downwind than upwind propagation (see Fig. B.5). This is important for the generation of time-dependent surfaces, as will be discussed in the next chapter.

Figure 3.4 shows a surface generated with the omnidirectional variance spectrum of Elfouhaily et al. (ECKV) as described in §B.3, combined with a cosine-2S spreading function (3.8) with $S = 2$ for all k values. The wind speed is 10 m s^{-1} . The simulated region is $100 \times 100 \text{ m}$ using 512×512 grid points. Note in this figure that the mean square slopes (mss) compare well with the corresponding Cox-Munk values shown in Table 4.1. The mss values for the generated surface are computed from finite differences, e.g.

$$mss_x(r, s) = \frac{z(r+1, s) - z(r, s)}{x(r+1) - x(r)}$$

averaged over all points of the 2-D surface grid. The $\langle \theta_x \rangle$ and $\langle \theta_y \rangle$ values shown in the figure are the average angles of the surface from the horizontal in the x and y directions. These are computed from

$$\theta_x(r, s) = \tan(mss_x(r, s)),$$

etc., averaged over all points of the surface. The mss_x values of this 2-D image can be compared with those of the 1-D simulation of Fig. 2.7.

The spreading function used in Fig 3.4 was chosen (with a bit of trial and error) to give a distribution of along-wind and cross-wind slopes in close agreement with the Cox-Munk values (except for a small amount of Monte-Carlo noise). Figure 3.5 shows a surface generated with a

slope variable	DFT value	Cox-Munk formula	Cox-Munk value
mss_x	0.031	$0.0316U$	0.032
mss_y	0.021	$0.0192U$	0.019
mss	0.052	$0.001(3 + 5.12U)$	0.054

Table 3.1: [Comparison of Cox-Munk mean square slopes and values for an DFT-generated 2-D surface.

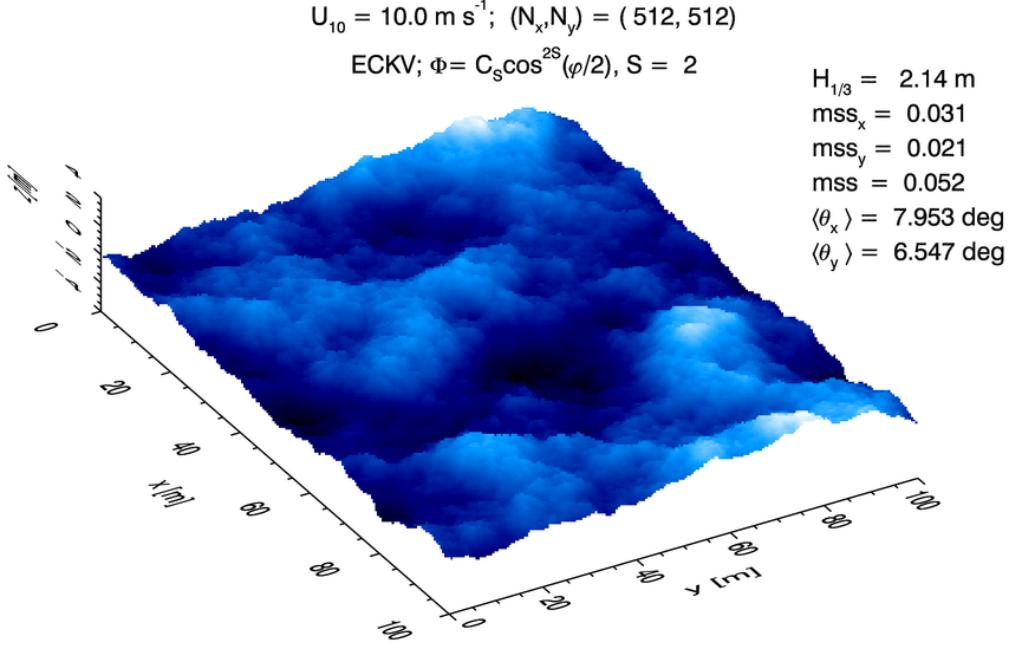


Figure 3.4: A sea surface generated with the ECKV omnidirectional spectrum and a cosine-2S spreading function with $S = 2$. Compare with Fig. 3.5.

cosine-2S spreading function with $S = 20$; all other parameters were the same as for Fig. 3.4. This S value gives wave propagation that is much more strongly in the downwind direction $\varphi = 0$, as would be expected for long-wavelength gravity waves in a mature wave field. The surface waves thus have a visually more “linear” pattern, whereas the waves of Fig. 3.4 appear more “lumpy” because waves are propagating in a wider range of angles φ from the downwind. These cosine-2S spreading functions are plotted in §B.3.1, along with other commonly used spreading functions.

As shown in §B.1, the total mean square slope depends only on the omnidirectional spectrum $\mathcal{S}(k)$. Thus the total mss is the same (except for Monte Carlo noise) in both figures, but most of the total slope is in the along-wind direction in Fig. 3.5.

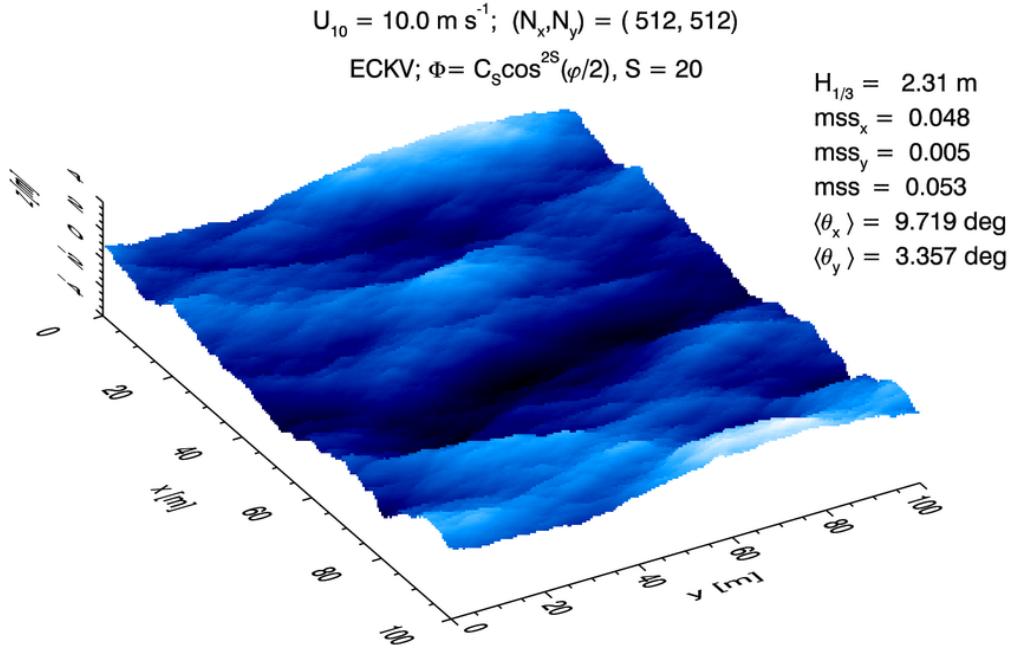


Figure 3.5: A sea surface generated with the ECKV omnidirectional spectrum and a cosine-2S spreading function with $S = 20$. Compare with Fig. 3.4.

3.4 Resampling the DFT Grid for Ray Tracing

One of the most important uses of DFT-generated sea surfaces is Monte-Carlo ray tracing for the computation of sea surface optical reflectance and transmittance. Such calculations require repeatedly finding the point of intersection of a light ray with the random sea surface and finding the angle of the ray from the normal to the surface at the point of intersection. Those calculations are most easily performed if the sea surface is constructed as an array of triangular wave facets. The three vertices of each wave facet define a plane, so that the ray tracing then amounts to finding the intersection of a line with a plane. The direction of the normal to the surface is constant within each triangular wave facet and is easily determined from the elevations at the vertices of the triangles.

Sea surfaces generated using DFT techniques are defined by a rectangular grid of surface elevations. Ray tracing with such a surface would be greatly complicated by the fact that the four points of each grid cell or wave facet do not in general define a plane, but rather a curved surface whose form depends on the surface elevations at the four points. Moreover, the local normal to the curved sea surface varies from point to point within each rectangular cell. It is therefore desirable to “resample” the rectangular DFT grid to obtain a grid of triangles. One way to do this is shown in Fig. 3.6. The blue lines of this figure show the rectangular DFT grid for the case of an 8×8 m region of water surface with the origin of the $x - y$ coordinate system placed at the center of the region. The DFT grid has $N_x = 16$ and $N_y = 8$ points. The red dots show a hexagonal grid of triangles defined using every other x grid point and every y point of the DFT grid; one of these triangles is colored in red. It is clear that this mapping requires that $N_y = N_x/2$. The number of triangle vertices defined in this way is $N_{\text{vertices}} = 3 \frac{N_y}{2} (\frac{N_y}{2} + 1) + 1$, and the number of triangular

wave facets generated from these vertices is $N_{\text{facets}} = \frac{3}{2}N_y^2$.

$$N_x = 16, N_y = 8; N_{\text{vertices}} = 61, N_{\text{facets}} = 96$$

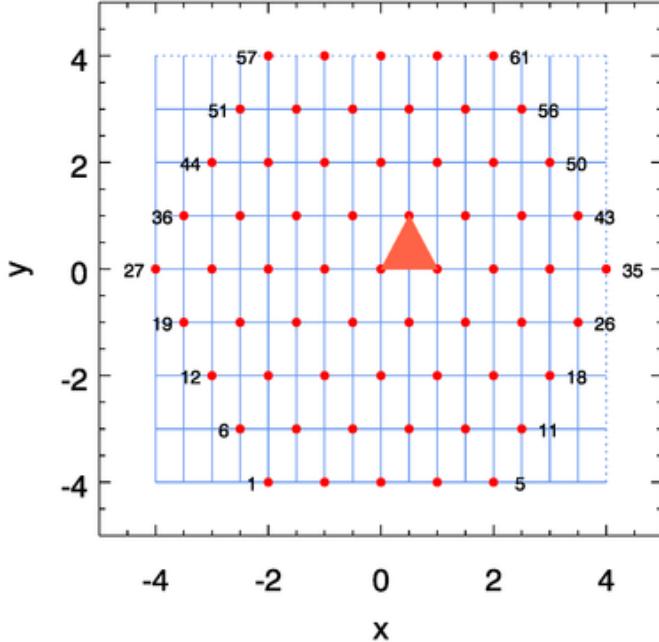


Figure 3.6: Example mapping of a rectangular computational grid used in DFT simulations of a random sea surface (blue lines) to a hexagonal grid of triangles as used in ray tracing (red dots). The DFT grid points at the right and top, shown by the dotted blue lines, are obtained by the inherent spatial periodicity of the surface as determined by DFT techniques. Thus the surface elevation of point 35 is the same as that of point 27, point 57 is the same as point 1, etc. One of the triangles generated from the DFT grid is shaded in red.

It may seem wasteful to compute the surface elevation at each of the DFT grid points (the points where the blue lines intersect) and then discard roughly half of these points when resampling the DFT grid to obtain a hexagonal grid of triangular wave facets. However the extremely fast run time of the FFT algorithm makes this option much more efficient than the regular Fourier technique given in Mobley (1994, his Eq. 4.77). That technique computes the surface elevations only at the needed triangle vertices, but requires *much* more time to evaluate the 2-D discrete Fourier transforms than the FFT technique described here. Other grid resampling schemes can be devised. For example, each DFT grid point can be used, with the result being a collection of “right-facing” and “left-facing” right triangles. However, I already had ray tracing code that uses the resampling described above to compute surface reflectance and transmittance functions (Mobley, 2014, Appendix B). A different layout requires a different formulation of the ray tracing algorithms, so I stayed with my original resampling scheme for this tutorial and subsequent work (Mobley, 2015).

Figure 3.7 shows an example of a sea surface defined on a rectangular DFT grid and the corresponding hexagonal patch of sea surface defined by a grid of triangular wave facets derived from a rectangular grid as in Fig. 3.6. The origin of the coordinate system has been placed at the center of the $L_x \times L_y = 100 \times 100$ m region. The hexagonal-grid sea surface is ready for use in ray tracing computations with existing algorithms. Although those computations are conceptually simple, the actual calculations are quite tedious. Mobley (2014, Appendix B) gives the details of a ray tracing algorithm for a sea surface defined by a hexagonal grid of triangular wave facets. (That report also gives the details of how ray tracing is performed for polarized light.)

3.5 Example: Reflectance by Different Surface Models

We have now seen how to correctly generate 2-D surface realizations. The previous chapter made two important points that underlie surface generation. First, in §2.5.1 we pointed out two errors in the widely used equations of Tessendorf (2004). His version of our Eq. (2.9) does not divide the one-sided energy spectrum by 2 to convert it to a two-sided spectrum, and his version of Eq. (2.11) does not have the overall scale factor of $\frac{1}{\sqrt{2}}$. Omission of these factors in either the 1-D equations (2.9) and (2.11), or in the 2-D equivalents (3.3) and 3.4), makes the Fourier amplitudes \hat{z} a factor-of-two too large. Consequently the wave amplitudes (surface elevations) will be too large and energy, which is proportional to wave amplitude squared, will not be conserved by a factor of four in a round-trip calculation like that of §2.5.2. Second, in §2.6 we pointed out the importance of resolving the sea surface slopes when the generated surfaces are used for optical calculations, and we presented a simple way to rescale variance spectra so that the generated surfaces adequately reproduce both the elevation and slope variances of real sea surfaces, even though the elevation variance spectrum is sampled over a limited range of spatial frequencies.

Figure 3.8 shows the consequences for optical calculations of either omitting the scale factors on the amplitude equations or failing to resolve the slope variance. Polarized ray tracing was performed as described in Mobley (2014) to compute the percentage of the total incident energy reflected into all directions for the sun at a given zenith angle in an otherwise black sky. The wind speed was 10 m s^{-1} , and the surface was modeled using $L_x = L_y = 200$ m and $N_x = 1024, N_y = 512$ as required for the resampling of the FFT grid as described in §3.4. Rays were traced for 50,000 independent surface realizations to obtain average reflectances with negligible Monte Carlo noise. The black curve shows the energy reflectance (without regard for the state of polarization) for a level sea surface; this reflectance is given by the well known Fresnel reflectance equations (e.g. Mobley, 1994, Eq. 4.14). The solid green curve shows the reflectance for DFT-generated surfaces when the correct amplitude equations (3.3) and (3.4) are used, and when the slope variance is fully resolved via the $\delta(k)$ function of Eqs. (2.17) and (2.18) for the given values of L_x, L_y, N_x and N_y . The dashed green curve shows the reflectance for surfaces generated using the well known Cox-Munk wind speed-wave slope statistics as described in Mobley (2014, §4.3). Cox-Munk surfaces model the surface slopes, but not the surface wave elevations. We see that the FFT-generated surface is in excellent agreement with the Cox-Munk surfaces. The agreement is good because both surfaces resolve the slope statistics. (Calculations not shown here indicate that Cox-Munk and FFT surfaces do differ by a few percent at large incident angles for other wind speeds.)

The red curve shows the reflectance computed when the correct amplitude equations (3.3) and

$$U = 5 \text{ m s}^{-1}; (L_x, L_y) = (100, 100); (N_x, N_y) = (32, 16)$$

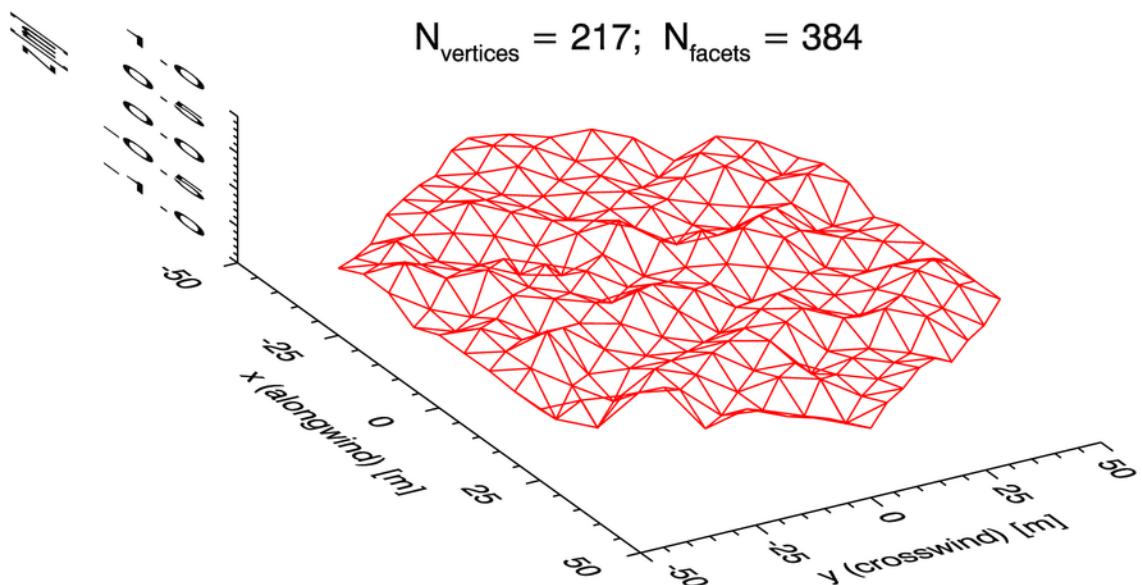
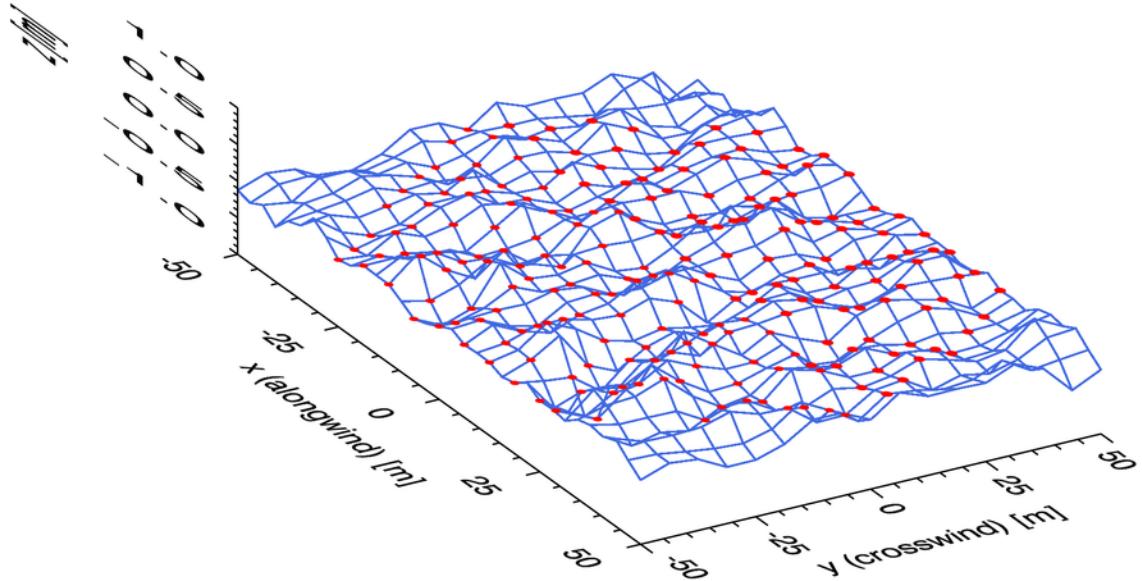


Figure 3.7: The blue lines in the top panel show a sea surface specified on a rectangular DFT grid. The red dots show the points used to define an hexagonal grid of triangles as in Fig. 3.6. The bottom figure shows the surface specified by the hexagonal grid of triangular wave facets shown by the red dots in the top figure.

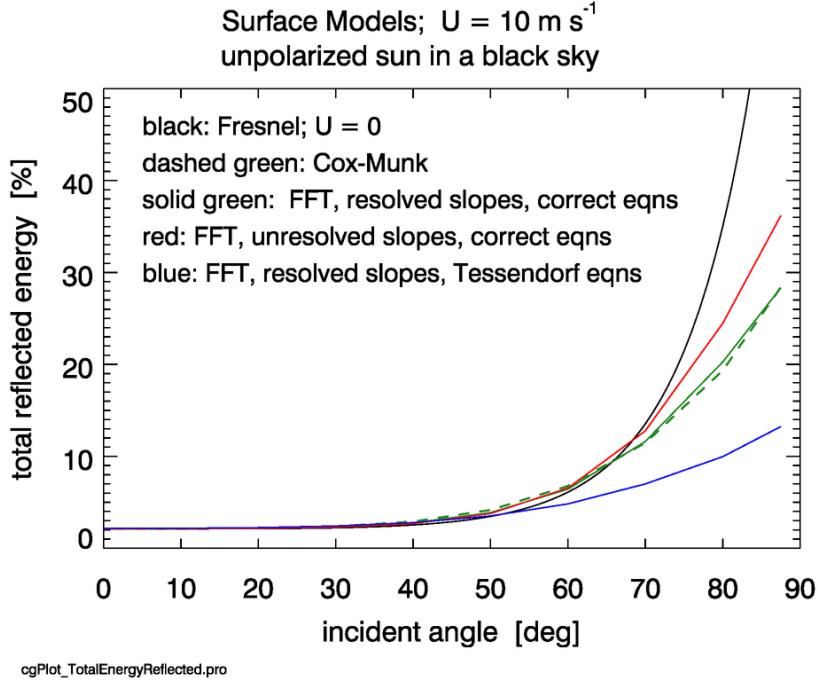


Figure 3.8: Energy reflectance by different sea surface models for a wind speed of 10 m s^{-1} .

3.4) are used, but the 2-D elevation variance spectrum is sampled only over the spatial frequency ranges determined by the values of L_x, L_y, N_x and N_y . That is, the slope variance is not fully resolved. Because the undersampled slope variance is less than the true slope variance, the generated surfaces are too smooth for the high spatial frequencies (short wavelengths), and the reflectance is too large. The difference is significant at sun angles near the horizon. The blue curve shows the reflectance when the slope variance is fully resolved, but the Tessendorf equations are used to generate the Fourier amplitudes. The surface is now too rough (the amplitudes are too large), which reduces the reflectance in the same manner as a higher wind speed does. (The reflectance at large incident angles decreases with increasing wind speed because of increased surface roughness.) Although ray tracing and optics are beyond the scope of this tutorial, Fig. 3.8 is included here to emphasize the importance of properly modeling the sea surface when doing calculations of surface optical properties.

3.6 Example: Tiling a Two-dimensional Surface

As was explained in §2.3, a Fourier-generated surface is periodic in the x and y directions. A qualitative visual impression of a large area of sea surface can be created by computing a patch of sea surface of some size $L_x \times L_y$ and then simply repeating or tiling that surface in the x and y directions. Figure 3.9 shows an example of a surface area (outlined in orange) that has been replicated in a 3×3 tile pattern to create a surface that is nine times as large as the computed surface. In a movie or video game, the sea surface would likely be just the background for something happening in the foreground, and the viewer might not notice the periodicity of the background, especially in a time-dependent image.

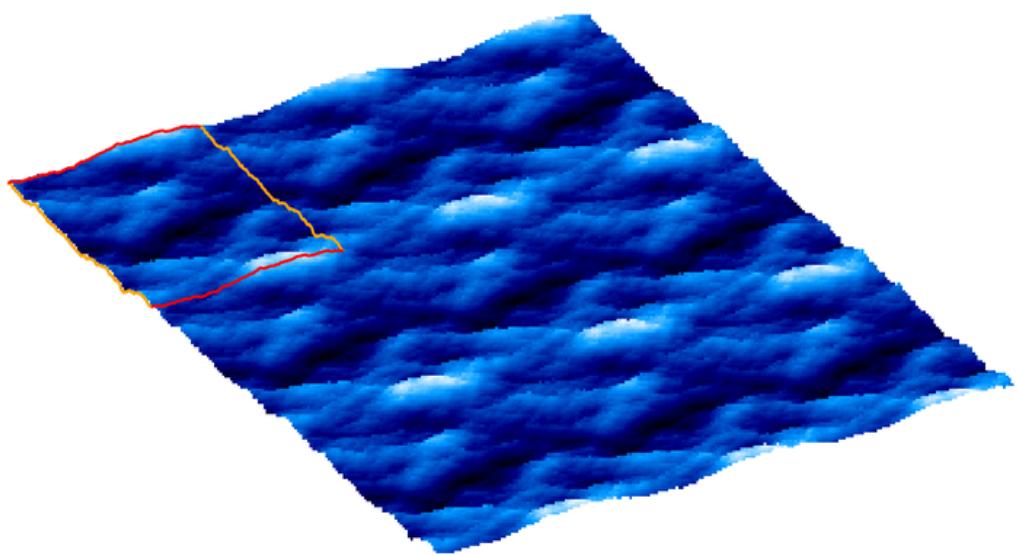


Figure 3.9: Example of a 3×3 tiling of a patch two-dimension surface to create a larger area of sea surface. The surface patch actually generated is outlined in orange.

CHAPTER 4

Two-Dimensional, Time-Dependent Surfaces

One final step remains: the addition of time dependence to a sequence of the sea surface realizations. Many scientific applications do not need time dependence. This is the case when many independent random realizations of sea surfaces are used for Monte Carlo ray tracing to compute the average optical reflectance and transmittance properties of wind-blown sea surfaces. However, time dependence is crucial for applications such as computer-generated imagery for movies.

We already have the needed theory in hand. The fundamental Eqs. (3.3) and (3.4), and their expansions such as Eq. (3.7) include time dependence. Most importantly, these equations made no simplifying assumptions about the $\pm\mathbf{k}$ symmetry of the two-sided variance spectrum $\Psi_{2s}(\mathbf{k})$.

As we have learned, the Fourier transform of a snapshot of the sea surface gives a two-sided variance spectrum with identical values for $-\mathbf{k}$ and $+\mathbf{k}$. This represents equal amounts of energy propagating in the $-\mathbf{k}$ and $+\mathbf{k}$ directions, i.e., equal amounts of energy in waves propagating upwind and downwind. Such a situation in nature gives standing waves. Here also, if $\Psi_{2s}(-\mathbf{k}) = \Psi_{2s}(\mathbf{k})$, the time-dependent surface will be standing waves composed of waves of equal energy propagating the $+\mathbf{k}$ and $-\mathbf{k}$ directions. To obtain waves propagating downwind, as is the case in a real ocean, we must use an asymmetric two-sided spectrum with $\Psi_{2s}(-\mathbf{k}) \ll \Psi_{2s}(+\mathbf{k})$, so that almost all of the energy is propagating downwind. Note, however, that we cannot simply set $\Psi_{2s}(-\mathbf{k}) = 0$, which represents no energy at all propagating upwind, because $\Psi_{2s}(-\mathbf{k}) = 0$ destroys the Hermitian property of Eq. (3.4). Thus we must conjure up an asymmetric variance spectrum that allows a nonzero (although perhaps negligible) amount of energy to propagate upwind.

An asymmetric two-sided variance spectrum can be constructed using an asymmetric spreading function $\Phi(k, \varphi)$ as described in §B.3.1. Spreading functions of the form

$$\Phi(k, \varphi) = C_s \cos^{2S(k)}(\varphi/2) \quad (4.1)$$

described there allow some energy to propagate in $-\mathbf{k}$ directions, i.e. when $|\varphi| > \pi/2$. With this choice of a spreading function, we can let the magnitude of $\Psi_{2s}(+\mathbf{k})$ equal the magnitude of the one-sided variance function $\Psi_{1s}(+\mathbf{k})$, which gives the total variance, because we assume that a negligible amount of the total energy propagates in $-\mathbf{k}$ directions. With this assumption, there is no division of $\Psi_{1s}(+\mathbf{k})$ by 2 in the first line of Eq. (3.3).

Once an asymmetric $\Psi_{2s}(\pm\mathbf{k})$ has been defined, we can evaluate Eq. (3.3) for a set of random

numbers $\rho(\mathbf{k}_{uv})$ and $\sigma(\mathbf{k}_{uv})$. *This is done only once.* Then to generate a sequence of sea surface realizations for times $t = 0, \Delta t, 2\Delta t, \dots$, we multiply the time-independent amplitudes by the time-dependent cosines and sines as shown in Eq. (3.7). We thus obtain the amplitudes $\hat{z}(k_{uv}, t)$ at the current time t . The inverse Fourier transform of this $\hat{z}(k_{uv}, t)$ gives the sea surface $z(\mathbf{x}_{rs})$ at time t .

The physics (or lack thereof) of this process is simple. We start with a realization of the sea surface at time zero. This surface contains waves of many discrete frequencies k_{uv} traveling in all directions φ_{uv} . Then to get the surface at any later time t , we simply propagate the sinusoidal waves of each frequency \mathbf{k}_{uv} in their original direction of travel through a phase change determined by the time step and the dispersion relation $\omega(k_{uv})$. For deep-water gravity waves, the dispersion relation is $\omega(k_{uv}) = \sqrt{gk_{uv}}$. It thus visually appears that the waves are propagating and the sea surface profile is changing with time. However, this Fourier transform technique is really just moving a collection of independent, non-interacting sinusoids through frequency-dependent phase angles to create a new surface realization from the sum of the phase-shifted sinusoids. In a real ocean, waves of different frequencies can interact with each other (redistribute energy among frequencies) in highly complex and nonlinear ways, so that the sea surface statistics may not be time-independent. This is, in particular, how little waves grow to big waves when the wind begins to blow over a level surface. Modeling the nonlinear evolution of a sea surface is beyond the abilities of Fourier transform techniques which are, at heart, just a mathematical artifice based on a time-independent directional variance density spectrum.

Figure 4.1 shows a sequence of surface realizations for a 10 m/s sec wind speed and a spatial grid of size $L_x \times L_y = 100 \times 100$ m, and a grid resolution of $N_x \times N_y = 256 \times 256$. A time series of images made with such a coarse grid could be useful for some non-scientific purposes.

We have seen that the spatial pattern of a sea surface generated by a Fourier transform is periodic. This is convenient for tiling a small computed region into a visually acceptable larger region. When time dependence is included and a finite-length time series of images is created, the sequence of images is not periodic in time because the various sinusoids comprising the surface do not have a common period. As pointed out by Tessendorf (2004, §4.2), this can be remedied by “quantizing” the temporal frequency $\omega(k_{uv})$ as follows.

Let T_r be the length of time over which the time sequence of surface realizations should exactly repeat. The number of frames N_t in the video loop determines the time step between frames, $\Delta t = T_r/N_t$. For a smoothly moving sea surface, N_t must be large enough that Δt is less than 0.1 s. Define $\omega_o \equiv 2\pi/T_r$. For deep-water gravity waves, the true temporal frequency $\omega(k_{uv}) = \sqrt{gk_{uv}}$ is replaced by

$$\tilde{\omega}(k_{uv}) = \left\lfloor \frac{\sqrt{gk_{uv}}}{\omega_o} \right\rfloor \omega_o, \quad (4.2)$$

where $\lfloor x \rfloor$ converts a real number x into its integer part (e.g., 15.23 is converted to 15). This operation slightly alters the temporal frequency of each wave component so that each component returns to exactly its initial shape after time T_r . A video loop can then be created from the sequence of surfaces, and the loop will match perfectly when the frame for time $T_r - \Delta t$ goes to the frame for time T_r , which is the same surface as time 0, after which the surfaces repeat. This adjustment to ω is greatest for the lowest frequencies, but the adjustment becomes smaller and smaller as the repeat time becomes larger and larger. It is thus easy to create a time-dependent small area of sea

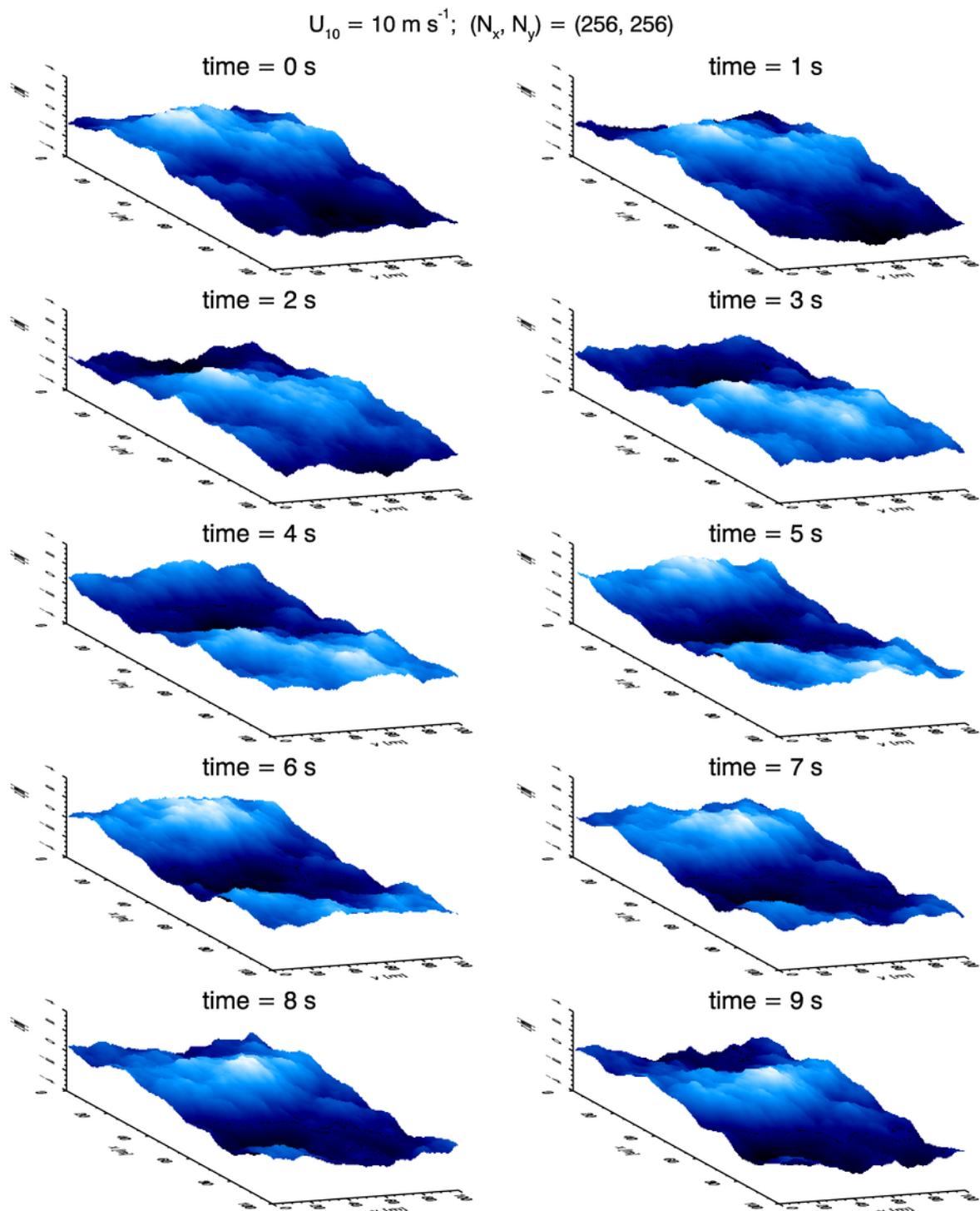


Figure 4.1: A time-dependent sequence of 2-D sea surfaces for a 10 m s^{-1} wind speed. The physical domain is $100 \times 100 \text{ m}$; the sampling is 256×256 points. The vertical scale of the plots is -3 to +3 m. The wind is blowing in the $+x$ direction, which is from the upper left to the lower right of each figure.

surface that can be tiled in both space and time to create an image of a larger sea surface over a longer time. This is good enough to fool movie-goers who have not studied this tutorial.

In order to employ a re-scaled variance spectrum as described in §2.6, determine the value of the δ_{Ny} parameter using the omnidirectional variance spectrum, as seen in Eq. (2.19). Then apply the $\delta(k)$ correction to the directional spectrum $\Psi(k_x, k_y)$ with $k = k_{uv}$ for each $(k_x(u), k_y(v))$ point of the rectangular grid.

An example of a 20-second (simulated time) video loop created using all of these tricks can be seen at http://www.oceanopticsbook.info/view/surfaces/level_2/twodimensional_timedependent_surfaces.

CHAPTER 5

Autocovariance Functions and the Wiener-Khinchin Theorem

This chapter explores an alternate path to the specification of surface roughness properties. Clearly, wind-dependent variance spectra are applicable only to surface waves that are generated by wind. Consider, however, the surface of a flowing river. The river's surface can have ripples or waves generated by turbulence resulting from unstable shear flow induced by flow over a shallow bottom, or by eddies created as the water flows around rocks in the river. These water surfaces can have different statistical properties, hence different reflectance properties, than wind-roughened water surfaces. Such surfaces can be described by their autocovariance functions.

Autocovariance functions can be converted to elevation variance spectra via the Wiener-Khinchin theorem. This chapter shows how that is done. Once a given autocovariance function has been converted to its equivalent elevation variance spectrum, the algorithms of the previous chapters are immediately applicable, even though the variance spectrum is not wind-dependent. Indeed, this conversion enables the Fourier transform methods of the previous chapters to be used to generate random realizations of any surface, not just water surfaces.

As is often the case, there is a large gap between textbook theory—usually developed for continuous variables or an infinite sample size of discrete values—and its implementation in a computer program for a finite sample size of discrete variables. In particular, careful attention¹ must be paid to sampling of an autocovariance function in order to obtain the corresponding variance spectrum, or vice versa. I find it disappointing and frustrating (but not surprising) that numerical matters such as the effects of finite sample size, maximum lag size, and exactly how to sample spectra or autocovariances (in particular, the material of §5.5) never seem to be mentioned

¹Given the care expended here on numerical details, it seems not unreasonable to ask for the same care in pronunciation. Professor “Khinchin’s” name is properly spelled Хинчин. The abominable Romanization of the Cyrillic X with a Latin Kh has been the cause of centuries of mispronunciation. The origin of the error lies perhaps in the adoption of some Greek letters for the Cyrillic alphabet, and the Cyrillic x looks similar to the Greek χ, which has a “kai” sound. In any case, this grievous orthographic error traces back at least as far as Edward Gibbon who, in his otherwise excellent history of the decline and fall of the Roman Empire, Romanized Чингис Хан as Gengis Khan rather than as Chhingis Han. The Cyrillic X is pronounced close to the ch in the Scottish “loch” or the German “doch.” However, pronunciation with a simple English H, i.e. Hinchin rather than Khinchin, will suffice unless you are dining with Russians. If you do find it necessary to say Khinchin in order not to confuse the chattering classes, at least pronounce the first guy’s name as “Veener” and not “Weener.”

in textbooks on digital signal processing or related subjects. It is left to the innocent student to spend a few weeks of unfunded time figuring out why various numerical results are not internally consistent or do not perfectly match the textbook theory.

This chapter begins with a discussion of autocovariances, and then the Wiener-Khinchin theorem is stated. The theorem is numerically illustrated first using the wind-dependent Pierson-Moskowitz elevation variance spectrum, for which certain values can be analytically calculated and used to check the numerical results. The modeling of water surfaces generated by shear-induced turbulence is then illustrated, again using analytical functions that allow for a rigorous check on the numerical results.

5.1 Autocovariance

The autocovariance $C_{zz}(\ell)$ of $z(r)$ is defined as

$$C_{zz}(\ell) \equiv \mathcal{E}\{z(r)z(r + \ell)\}, \quad (5.1)$$

where \mathcal{E} denotes statistical expectation and ℓ is the spatial lag. This definition in terms of the expectation holds for both continuous and discrete variables. For the present discussion with $z(r)$ being sea surface elevation, $C_{zz}(\ell)$ shows how strongly the sea surface elevation at one location is correlated to the elevation at a distance ℓ away. $C_{zz}(\ell)$ has units of m^2 , and $C_{zz}(\ell = 0)$ is the variance of the surface elevation. The autocovariance is an even function of the lag: $C_{zz}(-\ell) = C_{zz}(\ell)$.

Consider an infinite sample of discrete zero-mean surface elevations spaced a distance Δx apart. The autocovariance is then computed by (e.g. Proakis and Manolakis, 1996, their Eq. 2.6.3 with a minor change in notation)

$$C_{zz}(\ell) = \sum_{r=-\infty}^{+\infty} z(r)z(r + \ell), \quad \text{for } \ell = 0, \pm 1, \pm 2, \dots$$

Here ℓ is indexing the lag distance in units of the sample spacing Δx . For a finite sample of N discrete values, the same authors define the sample autocovariance by (their Eq. 2.6.11)

$$C_{zz}(\ell) = \sum_{r=0}^{N-|\ell|-1} z(r)z(r + \ell). \quad (5.2)$$

However, for a finite sample of N discrete values, perhaps with a non-zero mean, the IDL autocorrelation function (A_CORRELATE) uses

$$C_{zz}(\ell_r) = \frac{1}{N} \sum_{r=0}^{N-|\ell|-1} [z(r) - m][z(r + \ell) - m] \quad \text{for } -(N-2) \leq \ell_r \leq N-2, \quad (5.3)$$

where

$$m = \frac{1}{N} \sum_{r=0}^N z(r)$$

is the sample mean. Matlab computes the autocovariance via

$$C_{zz}(\ell) = \frac{1}{N-1} \sum_{r=0}^{N-|\ell|-1} [z(r) - m][z(r + \ell) - m].$$

Note that the lag must be less than the length of the sample. (That is, the sample locations are at $x_r = r\Delta x, r = 0, \dots, N-1$ and the allowed lag distances are $\ell_r = r\Delta x, r = 0, \dots, N-2$.) Note also the factor of $1/N$ in the IDL definition (5.3), which does not appear in Eq. (5.2), and which is a factor of $1/(N-1)$ in the Matlab version. These differences in the definitions and computations of autocovariances can make it difficult to compare textbook theory with real-world situations.

Nor is there even any concensus on the terms “autocovariance” and “autocorrelation.” Some authors (and this tutorial) define the nondimensional autocorrelation $\rho_{zz}(\ell)$ as the autocovariance normalized by the variance, i.e.

$$\rho_{zz}(\ell) \equiv \frac{C_{zz}(\ell)}{C_{zz}(0)}. \quad (5.4)$$

However, Proakis and Manolakis (1996) call the autocovariance as used here the autocorrelation, and they call the autocorrelation of Eq. (5.4) the “normalized autocorrelation.” These sorts of differences, along with the various definitions for Fourier transforms, can cause much grief when comparing the numerical outputs of different computer codes, or numerical outputs with textbook examples.

5.2 The Wiener-Khinchin Theorem

Now that the autocovariance has been defined, the Wiener-Khinchin theorem can be stated: *The Fourier transform of the autocovariance equals the variance spectral density function.* Symbolically,

$$\mathfrak{F}_\nu\{C_{zz}(\ell)\} = \mathcal{S}_{2s}(\nu). \quad (5.5)$$

Indeed, some texts *define* the spectral density as the Fourier transform of the autocovariance. The inverse is of course

$$\mathfrak{F}_\nu^{-1}\{\mathcal{S}_{2s}(\nu)\} = C_{zz}(\ell). \quad (5.6)$$

Here \mathcal{S}_{2s} is a two-sided spectral density function as discussed in §2.3.

It is important to note (as emphasized by the ν subscript on the Fourier transform operator \mathfrak{F}) that the theorem is written with the ν version of the Fourier transform (Eq. 1.8), and the density function is written in terms of the spatial frequency ν , which has units of 1/meters. (In the time domain, the conjugate variables are time t in seconds and frequency f in 1/seconds = cycles/second = Hz.) The spectral density function for the angular spatial frequency $k = 2\pi\nu$ (or angular temporal frequency $\omega = 2\pi f$ in radians per second in the time domain) can be obtained by noting that corresponding intervals of the spectral densities contain the same amount of variance:

$$\mathcal{S}_{2s}(k)dk = \mathcal{S}_{2s}(\nu)d\nu,$$

which gives

$$\mathcal{S}_{2s}(k) = \frac{1}{2\pi} \mathcal{S}_{2s}(\nu = k/2\pi). \quad (5.7)$$

Note that ℓ varies from $-\infty$ to $+\infty$ and, likewise, ν or k ranges over all negative and positive values. The variance spectrum obtained from the Fourier transform of an autocovariance function is therefore a two-sided spectrum.

Comment: In light of Eq. (5.7), the theorem stated in terms of angular spatial frequency k appears to be

$$\mathfrak{F}_k\{C_{zz}(\ell)\} = 2\pi \mathcal{S}_{2s}(k), \quad (5.8)$$

with the inverse

$$\mathfrak{F}_k^{-1}\{\mathcal{S}_{2s}(k)\} = \frac{1}{2\pi} C_{zz}(\ell). \quad (5.9)$$

I say “appears to be” because I’ve never seen the theorem written this way because the textbooks all seem to stick with x and ν (or t and f in the time domain). As Press et al. (1992, page 491) say in *Numerical Recipes*, “There are fewer factors of 2π to remember if you use the (ν or) f convention, especially when we get to discretely sampled data....” In any case, Eqs. (5.8) and (5.9) are consistent with the k spectrum of Eq. (5.17), discussed below.

The theorem is usually proved in textbooks for continuous variables x and ν . However, in numerical application to a finite number of discrete samples, discrete variables x_r and ν_u or k_u are used, and proper attention must be paid to pesky factors of N , 2π , and bandwidth, and to the array ordering required by a particular FFT routine.

The continuous-variable Fourier transform in Eq. (5.5) gives a spectral density $\mathcal{S}_{2s}(\nu)$ with units of $\text{m}^2/(1/\text{m})$. However, if the theorem is written for a DFT of discrete data $C_{zz}(\ell_r) = C_{zz}(r)$,

$$\mathfrak{D}_\nu\{C_{zz}(r)\} = \mathcal{S}_{2s}(\nu_u), \quad (5.10)$$

then the resulting discrete spectrum $\mathcal{S}_{2s}(\nu_u) = \mathcal{S}_{2s}(u)$ has units of m^2 . Just as was discussed in Eqs. (1.20) and (2.12), this discrete spectrum must be divided by the bandwidth $\Delta\nu$ in order to obtain the density at $\nu = \nu_u$. That is,

$$\mathcal{S}_{2s}(\nu = \nu_u) = \mathcal{S}_{2s}(u)/\Delta\nu. \quad (5.11)$$

5.3 The Horoshenkov Autocorrelation Function

Horoshenkov et al. (2013, Eq. 4) give an analytic formula for the autocorrelation function of surface waves generated by bottom-induced turbulence in shallow flowing water. In the notation of this note, this function is²

$$\rho_{zz}(\ell) = \exp\left(-\frac{\ell^2}{2\sigma_w^2}\right) \cos\left(\frac{2\pi}{L_o}\ell\right). \quad (5.12)$$

Here “ σ_w relates to the spatial radius of correlation (correlation length)” and “ L_o relates to the characteristic period in the surface wave pattern.” The average values for the physical conditions of the Horoshenkov et al. (2013) study are $\sigma_w = 0.22$ m and $L_o = 0.17$ m.

²Equation (5.12) is Horoshenkov’s $W(\rho)$ as shown in their abstract and in their conclusions, where it has a factor of $1/2$ in the exponential. Their Eq (4) does not have the $1/2$. This is probably a typo since Gaussians usually have the form $\exp[-x^2/(2\sigma^2)]$.

5.3.1 The Horoshenkov Variance Spectrum

Equation (5.12), when converted to an autocovariance via a factor of the variance, $\langle z^2 \rangle = C_{zz}(0)$, has the form

$$\begin{aligned} C_{zz}(\ell) &= C_{zz}(0) \exp\left(-\frac{\ell^2}{2\sigma_w^2}\right) \cos\left(\frac{2\pi}{L_o}\ell\right) \\ &= C_{zz}(0) \exp(-a^2\ell^2) \cos(q_o\ell), \end{aligned} \quad (5.13)$$

where $a = 1/(\sqrt{2}\sigma_w)$ and $q_o = 2\pi/L_o$. This function has an easily computed analytical Fourier transform.

The continuous-variable Fourier transform of this $C_{zz}(\ell)$ is given by Eq. (1.8):

$$\mathcal{S}_{2s}(\nu) = \mathfrak{F}_\nu\{C_{zz}(\ell)\} = \int_{-\infty}^{\infty} C_{zz}(\ell) e^{-i2\pi\nu\ell} d\ell. \quad (5.14)$$

Here ℓ and ν are continuous variables; $\mathcal{S}_{2s}(\nu)$ has units of m^3 , which is interpreted as $\text{m}^2/(1/\text{m})$ as explained in §1.2. Note that this variance spectral density is two-sided, i.e., $-\infty < \nu < \infty$. Expanding the complex exponential via $e^{-i\theta} = \cos\theta - i\sin\theta$ gives

$$\begin{aligned} \mathcal{S}_{2s}(\nu) &= \int_{-\infty}^{\infty} C_{zz}(0) \exp(-a^2\ell^2) \cos(q_o\ell) \cos(2\pi\nu\ell) d\ell \\ &\quad - i \int_{-\infty}^{\infty} C_{zz}(0) \exp(-a^2\ell^2) \cos(q_o\ell) \sin(2\pi\nu\ell) d\ell. \end{aligned}$$

The imaginary term is zero because the integrand is an odd function of ℓ . Using the identity

$$\cos x \cos y = \frac{1}{2} [\cos(x+y) + \cos(x-y)]$$

gives

$$\mathcal{S}_{2s}(\nu) = 2C_{zz}(0) \int_0^{\infty} \exp(-a^2\ell^2) \frac{1}{2} \{\cos[(2\pi\nu + q_o)\ell] + \cos[(2\pi\nu - q_o)\ell]\} d\ell.$$

The integral

$$\int_0^{\infty} \exp(-a^2\ell^2) \cos(b\ell) d\ell = \frac{\sqrt{\pi} \exp[-b^2/(4a^2)]}{2a}$$

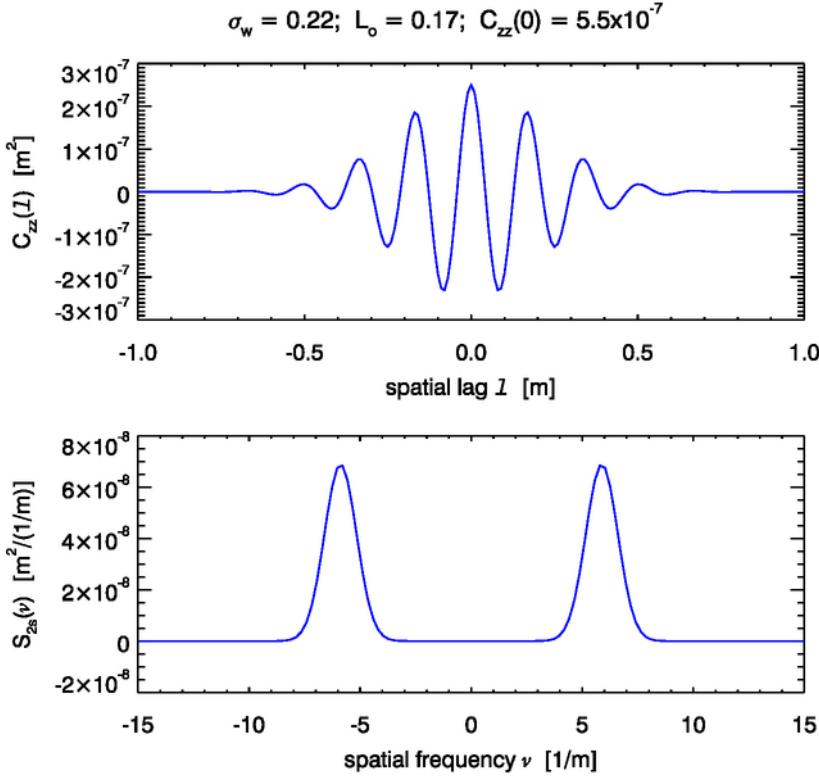
gives the Fourier transform of the $C_{zz}(\ell)$ of Eq. (5.13):

$$\mathcal{S}_{2s}(\nu) = \sqrt{\frac{\pi}{2}} \sigma_w C_{zz}(0) \left\{ \exp\left[-\frac{1}{2}(2\pi\sigma_w)^2(\nu + 1/L_o)^2\right] + \exp\left[-\frac{1}{2}(2\pi\sigma_w)^2(\nu - 1/L_o)^2\right] \right\}. \quad (5.15)$$

This variance spectrum has maxima at $\nu = \pm 1/L_o$, where the value is very close to $\sqrt{\frac{\pi}{2}} \sigma_w C_{zz}(0)$. Figure 5.1 plots this $C_{zz}(\ell)$ (Eq. 5.13) and $\mathcal{S}_{2s}(\nu)$ (Eq. 5.15) for typical values of $\sigma_w = 0.22 \text{ m}$, $L_o = 0.17 \text{ m}$, and $C_{zz}(0) = 2.5 \times 10^{-7} \text{ m}^2$. Note that the sub peaks of the autocovariance lie at integer multiples of $\pm L_o$, and that the peaks of the spectrum are at $\pm 1/L_o$.

By definition, the integral over all frequencies of an elevation variance spectral density gives the total elevation variance $\langle z^2 \rangle$:

$$\langle z^2 \rangle = \int_{-\infty}^{\infty} \mathcal{S}_{2s}(\nu) d\nu.$$



`Horoshenkov_analytic_nu.pro`

Figure 5.1: The Horoshenkov autocovariance and variance spectral density $S_{2s}(\nu)$ for typical parameter values taken from their Table 3.

This can be computed analytically for the spectrum of Eq. (5.15). The $S_{2s}(\nu)$ spectrum of Eq. (5.15) is the sum of two identical Gaussians centered at different ν values. Consider the one centered at $\nu = 1/L_o$, which involves the exponential with the $\nu - 1/L_o$ term. The total variance is then twice the integral of this Gaussian:

$$\langle z^2 \rangle = 2\sqrt{\frac{\pi}{2}}\sigma_w C_{zz}(0) \int_{-\infty}^{\infty} \exp\left[-\frac{1}{2}(2\pi\sigma_w)^2(\nu - 1/L_o)^2\right] d\nu.$$

Letting $x = \nu - 1/L_o$ gives

$$\langle z^2 \rangle = 2\sqrt{\frac{\pi}{2}}\sigma_w C_{zz}(0) \int_{-\infty}^{\infty} \exp[-c^2 x^2] dx,$$

where $c^2 = \frac{1}{2}(2\pi\sigma_w)^2$. The integral

$$\int_0^{\infty} \exp[-c^2 x^2] dx = \frac{\sqrt{\pi}}{2c}$$

then gives the final result:

$$\langle z^2 \rangle = 4\sqrt{\frac{\pi}{2}}\sigma_w C_{zz}(0) \frac{\sqrt{\pi}}{2\frac{1}{\sqrt{2}}2\pi\sigma_w} = C_{zz}(0). \quad (5.16)$$

Thus starting with a variance of $C_{zz}(0)$ in the autocovariance of Eq. (5.13), obtaining the variance spectral density from the Fourier ν -transform of the autocovariance, and then integrating the variance spectral density over ν thus gives back the variance as originally specified in the autocovariance function.

However, if the above process is naively carried through starting (as in Eq. 5.14) with the k -transform of Eq. (1.10), the end result (as in Eq. 5.16) is $2\pi C_{zz}(0)$. This extra factor of 2π is rectified by the $1/2\pi$ factor seen in Eq. (5.7). Thus the k version of the Horeshenkov spectral density is

$$\mathcal{S}_{2s}(k) = \frac{1}{2\pi} \sqrt{\frac{\pi}{2}} \sigma_w C_{zz}(0) \left\{ \exp \left[-\frac{1}{2} \sigma_w^2 (k + q_o)^2 \right] + \exp \left[-\frac{1}{2} \sigma_w^2 (k - q_o)^2 \right] \right\}. \quad (5.17)$$

Integration of this $\mathcal{S}_{2s}(k)$ over all k then results in $\langle z^2 \rangle = C_{zz}(0)$, as required. The reader may enjoy showing that the inverse k transform (1.11) of the spectral density (5.17) gives $C_{zz}(\ell)/(2\pi)$, as expected from Eq. (5.9).

5.4 Numerical Illustration of the Wiener-Khinchin Theorem

The Wiener-Khinchin theorem can be illustrated with one figure, but that figure requires a lot of discussion.

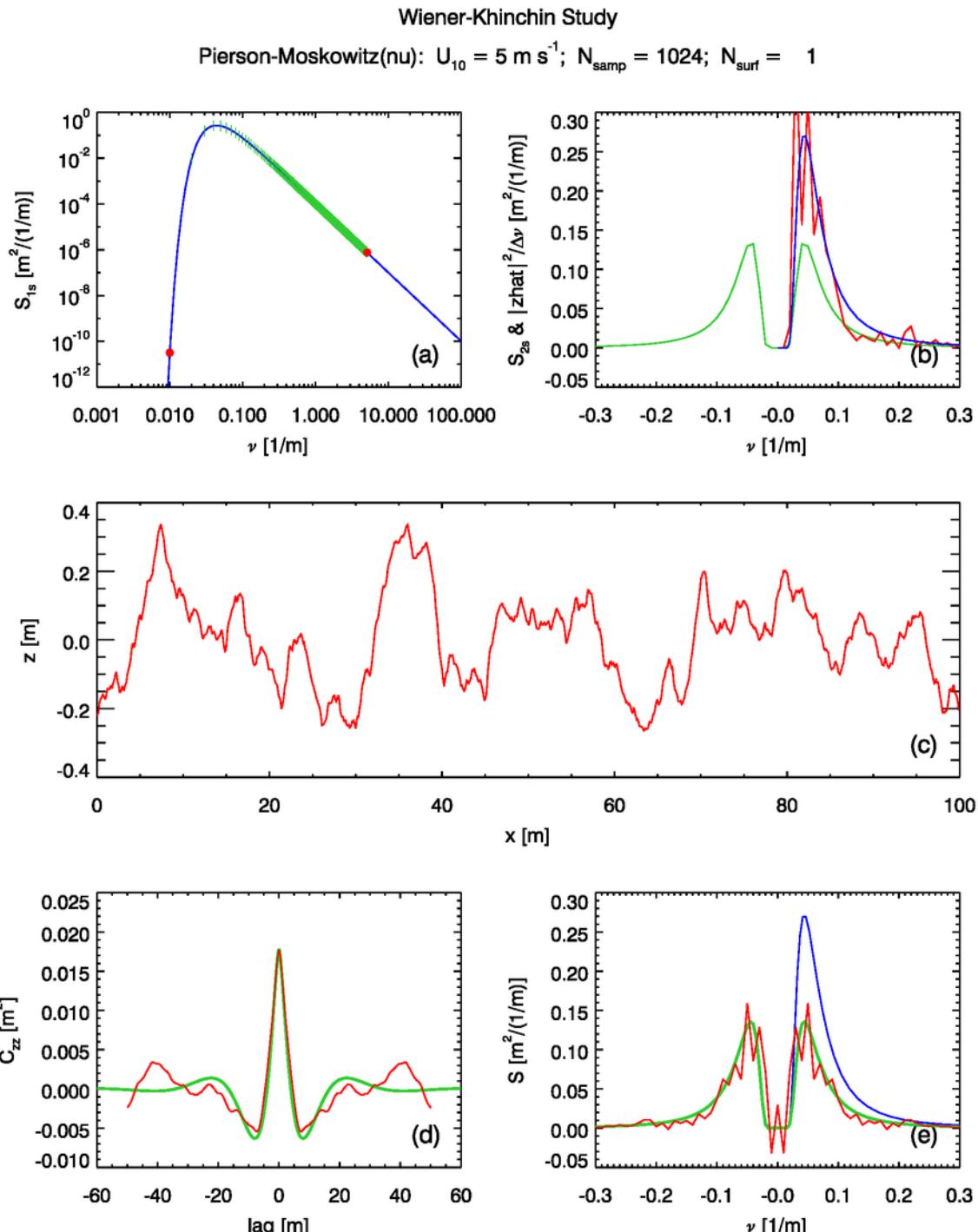
The blue curve of Panel (a) in Fig. 5.2 plots the one-sided Pierson-Moskowitz spectrum (B.12) for wind speed of $U_{10} = 5 \text{ m s}^{-1}$. Using this spectrum, surfaces are generated at $N = 1024$ points over a region of length $L = 100 \text{ m}$. Note that N is a power of 2 as required for the use of the fast Fourier transform algorithm. The spacing of these points is at intervals of $\Delta x = L/N = 0.0944 \text{ m}$. The red dot at $\nu_1 = 1/L = 0.01 \text{ 1/m}$ is the fundamental frequency. The point at $\nu_{N/2} = 1/(2\Delta x) = 5.12 \text{ 1/m}$ is the Nyquist spatial frequency. The green vertical ticks show the locations of the remaining $N - 2$ points, which are evenly spaced at intervals of $\Delta\nu = 0.01 \text{ 1/m}$.

These discrete samples of the variance spectrum are then used as described in §2.5.1 to create a random realization of the sea surface $z(r)$ at N points. One such surface, generated for a particular sequence of random variables, is shown in Fig. 5.2(c). The periodogram of this surface, computed via Eqs. (2.5, 2.6, 2.12), is shown in red Fig. 5.2(b). The blue curve in this panel is the one-sided spectrum $\mathcal{S}_{1s}(\nu)$ of Panel (a), replotted for reference. The statistical noise in the periodogram is Gaussianly distributed about the theoretical $\mathcal{S}_{1s}(\nu)$. These three panels of the figure are the essentially the same as Fig. 2.4. (The sequence of random numbers used to generate the surface is different and linear axes are used for the upper-right panel.)

Equation (5.3) applied to the $z(r)$ of Panel (c) gives the autocovariance shown in red in Panel (d). This curve contains statistical noise. To obtain a theoretical curve for comparison, the Pierson-Moskowitz spectrum was sampled at 2048 points to insure coverage of most of the spectrum. The Wiener-Khinchin theorem (5.10) was then used to obtain the autocovariance from the discretely sampled spectrum:

$$C_{zz}(\ell_r) = \mathfrak{D}_\nu^{-1}\{\mathcal{S}_{2s}(u)\} = \mathfrak{D}_\nu^{-1}\{\mathcal{S}_{2s}(\nu = \nu_u)\Delta\nu\}. \quad (5.18)$$

Here the DFT was computed via Eq. (1.15). Note that the discrete spectrum $\mathcal{S}_{2s}(u)$ was obtained by sampling the continuous spectral density at the desired ν values and then multiplying by the appropriate bandwidth. Because $\mathcal{S}_{2s}(u)$ is a real and even function of ν_u , its Fourier transform is



WK_PM2H_nu.pro

Figure 5.2: Illustration of the Wiener-Khinchin theorem for a single realization of a random sea surface.

also real and even. The result is shown as the green curve in this Panel (d). Equation (2.13) gives the total variance of $\langle z^2 \rangle = 0.0197 \text{ m}^2$ for $U_{10} = 5 \text{ m s}^{-1}$. The numerical result obtained by sampling $S_{2s}(\nu)$ and taking the inverse Fourier transform as just described gives $C_{zz}(0) = \langle z^2 \rangle = 0.0178 \text{ m}^2$.

The autocorrelation of the Pierson-Moskowitz spectrum in temporal form can be analytically computed (Latta and Bailie, 1968, Eq. 40), but the result is a formula of horrible complexity consisting of the sum of five slowly converging infinite series, the terms of which are themselves are products of infinite series. That paper plots the numerically evaluated result in terms of an unspecified but normalized temporal lag, which makes comparison with the present results for spatial lag quantitatively impossible. However, de Boer (1968, §4.3 & 4.4) obtained the spatial covariance of the Pierson-Moskowitz spectrum in the form of integrals of Bessel functions, which also require careful numerical evaluation. Figure 5.3 shows their result for the autocovariance function of waves in the down-wind direction. Their plot is in terms of a nondimensional normalized lag distance $\xi_N = (2g/U^2)\ell$. The green curve of Fig. 5.2(d) has a minimum of -0.0063 m^2 at $\ell = \pm 7.96 \text{ m}$. This translates to an autocorrelation of -0.31 at a normalized lag of $\xi_N = 6.3$. These value are in reasonable agreement with the minimum seen in Fig. 5.3, keeping in mind that the curves in that figure were themselves generated on a 1960's era computer by difficult numerical integrations of unknown accuracy. The agreements for the variance and the location and magnitude of the minimum indicate that the numerically computed $C_{zz}^{PM}(\ell)$ is probably correct for all lags. (This numerical calculation will be verified again with greater accuracy in the discussion of the Horoshenkov spectra below, for which the exact autocovariance is known.)

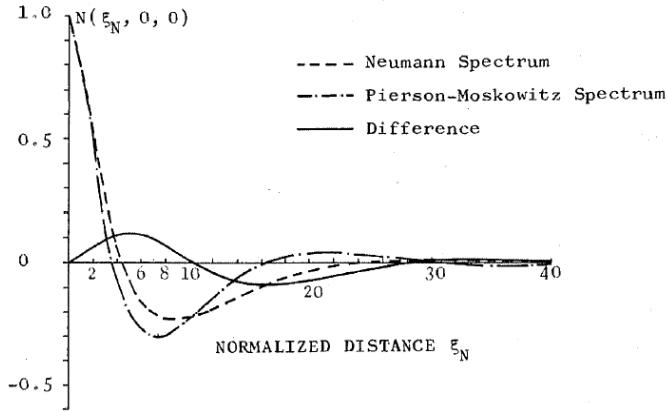


Figure 5.3: Fig. 7 from de Boer (1968): “Spatial correlation Function of Wind-Generated Ocean Surface Waves in the Down-Wind Direction.” The normalized lag distance is $\xi_N = (2g/U^2)\ell$.

Taking the DFT of the green curve in Fig. 5.2(d) should give the two-sided spectrum $S_{2s}(\nu)$ corresponding to the one-sided spectrum plotted in Panel (a). The green curve in Panel (e) of that figure shows the result (after dividing by the finite bandwidth, as mentioned previously), which is indeed one-half of the one-sided spectrum $S_{1s}(\nu)$ (shown in blue). This provides a check on the correctness of a round-trip Fourier transform.

Taking the DFT of the red curve in Panel (d) gives a sample estimate of $S_{2s}(\nu)$, which is shown in red in Panel (e). This curve has statistical noise, but it visually appears to be distributed about the theoretical value given by the green curve.

The statistical noise inherent in any single random realization of the sea surface and its autocovariance can be reduced by averaging the results from many surface realizations. Figure 5.4 is the same as Fig. 5.2, except that $N_{\text{surf}} = 100$ independent surfaces are generated. This reduces the statistical noise by a factor of $1/\sqrt{100}$. The red curve in Panel (b) shows the ensemble average periodogram for the 100 surfaces. It is clear that the average periodogram is in excellent agreement with the theoretical variance spectrum, except for a small amount of remaining statistical noise.

The red curve of Panel (d) is the average autocovariance for the 100 surfaces. This curve is much closer to the theoretical (green) curve than the autocovariance for the single surface of Fig. 5.2(d). The DFT of this average autocovariance is shown by the red curve in Panel (e). Again, this curve has much less noise and is closer to the (green) theoretical spectrum.

The statistical noise in the ensemble averages can be made arbitrarily small by averaging more and more surfaces. Figure 5.5 shows that averages for 1,000 surfaces have noise levels in the periodogram, autocovariance, and spectrum derived from the autocovariance, that are almost unnoticeable at the scale of the figures.

5.5 Sampling Strategy and Computational Details

This section shows exactly how the calculations underlying Figs. 5.2, 5.4, and 5.5 are performed. The devil is in the details, and these details are seldom discussed in the literature. Consider the case of $N = 8$, which will allow individual points to be plotted. Of course, with so few sample points, the variance spectrum is not adequately sampled and the resulting sea surface is unphysical because it has far too little variance. However, the algorithms are the same for any value of N .

Consider first the generation of the random sea surface with N points. As discussed previously, the two-sided spectrum must be sampled at exactly N spatial frequencies, which are given by Eq. (2.8). The green dots in Fig. 5.6(b) show these points for the case of $N = 8$. The frequency values, written in math order as in Eq. (2.7), are

$$\{\nu_u, u = 0, 1, \dots, N - 1\} = \left[-\frac{N}{2} + 1, \dots, -1, 0, 1, \dots, \frac{N}{2} \right] \Delta\nu, \quad (5.19)$$

which for the choice of $L = 100$ m and $N = 8$ gives

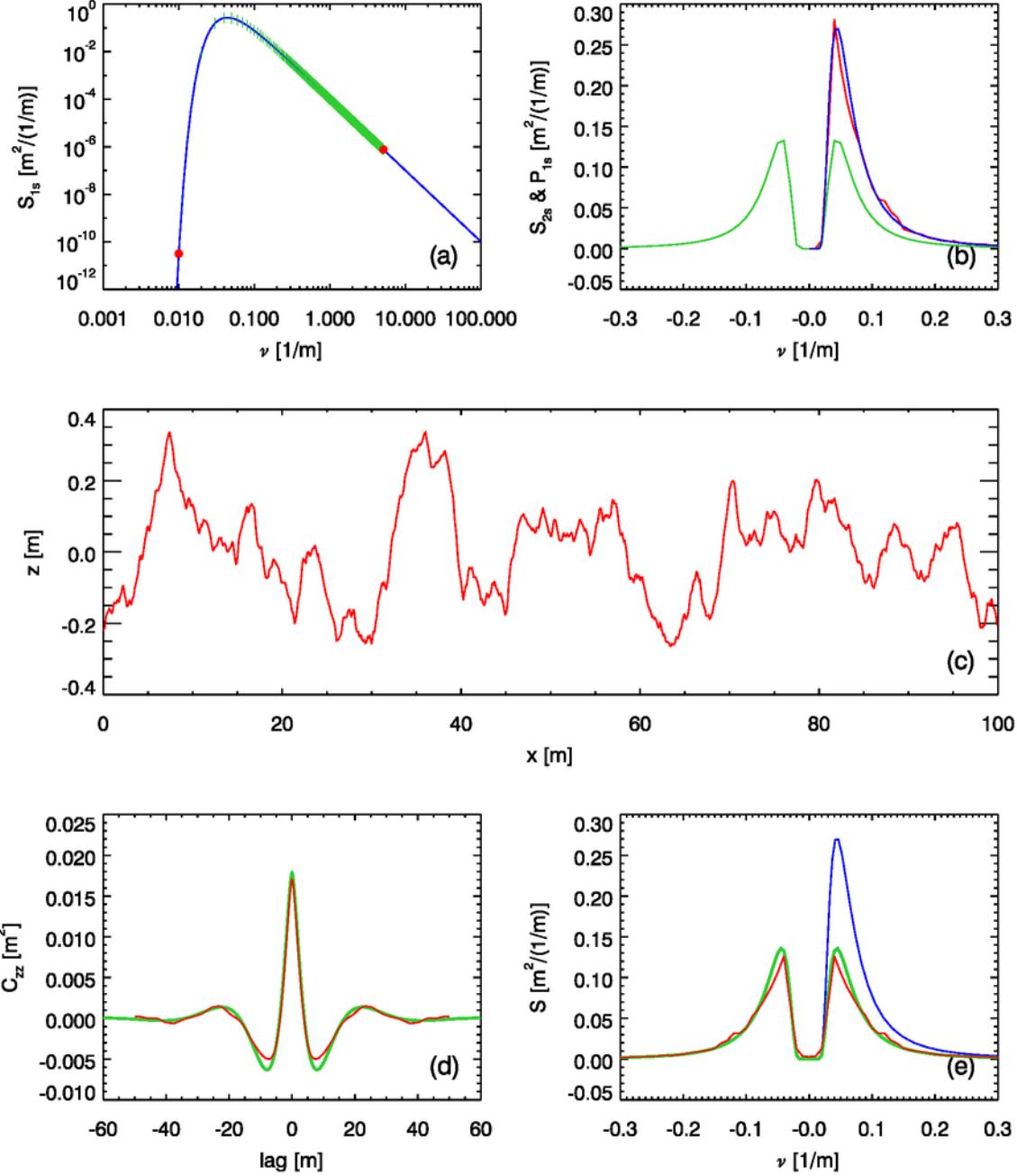
$$\{\nu_u, u = 0, 1, \dots, 7\} = [-3, -2, -1, 0, 1, 2, 3, 4] \Delta\nu, \quad (5.20)$$

where $\Delta\nu = 1/L = 0.01$ m⁻¹. Here braces $\{\dots\}$ denote a set of frequencies labeled by u values, and brackets $[\dots]$ denote an array of frequency values as shown. Note that the sampled frequencies are symmetric about $\nu = 0$, except for one “extra” point at index $u = N - 1$ or frequency $(N/2)\Delta\nu$. This value is the Nyquist frequency, which in IDL is stored as the last element of the frequency array in math order. Sampling the spectrum at *exactly* this pattern of frequencies guarantees that the spectral amplitudes generated from them are Hermitian, which in turn guarantees that the generated sea surface is real. The red dots in Panel (c) of the figure show the 8 surface elevations generated for a particular sequence of random numbers. The values are at $x_r = 0$ to $L - \Delta x$ for $r = 0$ to $N - 1$. Fourier-generated surfaces are inherently periodic, so that $z(L) = z(0)$.

Now take the inverse DFT as in Eq. (5.18) for the discrete spectrum given by the green dots in Panel (b). The result is the autocovariance values shown by the green dots in Panel (d). It

Wiener-Khinchin Study

Pierson-Moskowitz: $U_{10} = 5 \text{ m s}^{-1}$; $N_{\text{samp}} = 1024$; $N_{\text{surf}} = 100$

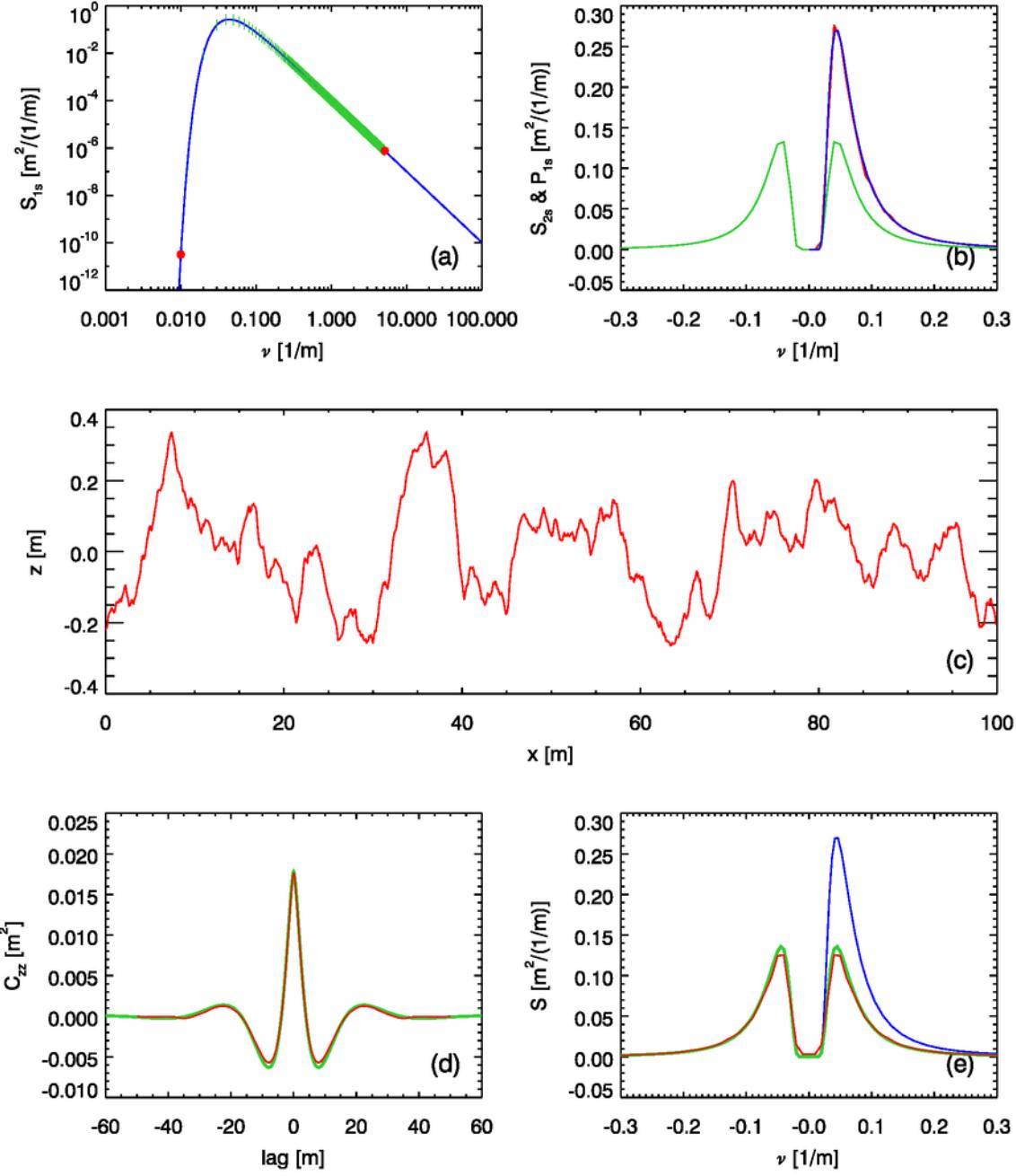


WK_PM2H_nu.pro

Figure 5.4: Same as Fig. 5.2, but for 100 surface realizations. The red curve in Panel (d) is the average of the autocovariances of the 100 surfaces. The red curve in Panel (e) is the Fourier transform of the 100-surface average of Panel (d). Only the first surface is plotted in Panel (c).

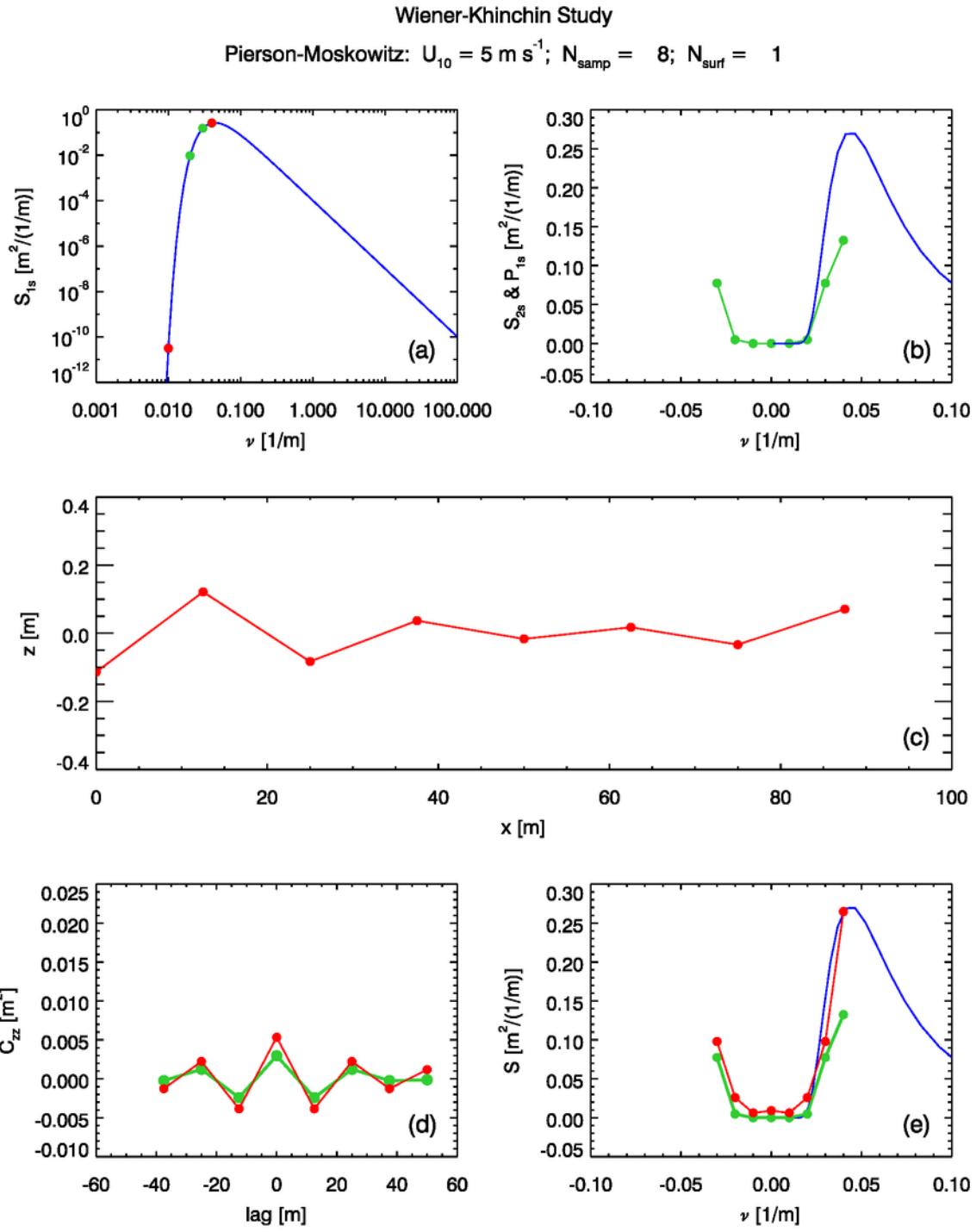
Wiener-Khinchin Study

Pierson-Moskowitz: $U_{10} = 5 \text{ m s}^{-1}$; $N_{\text{samp}} = 1024$; $N_{\text{surf}} = 1000$



WK_PM2H_nu.pro

Figure 5.5: Same as Fig. 5.2, but for 1000 surface realizations. The red curve in Panel (d) is the average of the autocovariances of the 1000 surfaces. The red curve in Panel (e) is the Fourier transform of the 1000-surface average of Panel (d). Only the first surface is plotted in Panel (c).



WK_PM2H_nu.pro

Figure 5.6: Illustration of sampling strategy for $N = 8$ sample points.

is important to note that these $N = 8$ lag values follow the same pattern (in math order) as the frequencies:

$$\{\ell_r, r = 0, 1, \dots, 7\} = [-3, -2, -1, 0, 1, 2, 3, 4]\Delta x, \quad (5.21)$$

where now $\Delta x = L/N = 12.5$ m. The lags are symmetric about $\ell = 0$, except for one “extra” point at $(N/2)\Delta x$. This is analogous to the one extra value in the frequency spectrum at the Nyquist frequency. Taking the forward DFT of these 8 autocovariance values as in Eqs. (5.10) and (5.11) gives the green points plotted in Panel (e). These values are of course exactly the 8 points of the original spectrum, as shown by the green dots in Panel (b). This is just a check on the correct implementation of the “round trip” calculation of inverse and forward Fourier transforms.

Now suppose that we wish to compute the autocovariance of the surface elevations, and from that obtain an estimate of the variance spectrum via the Wiener-Khinchin theorem. This provides a more stringent test of the calculations because of the intermediate sea surface in between the variance spectrum and the autocovariance. The crucial observation is that when calling the IDL autocovariance routine A_CORRELATE, that routine must be given an array of the requested lag indices (lags in units of Δx) as seen in Eq. (5.21). Thus for an array of surfaces,

$$\text{zsurf} = [z(r = 0), z(r = 1), \dots, z(r = 7)], \quad (5.22)$$

an array of lags

$$\text{lagindex} = [-3, -2, -1, 0, 1, 2, 3, 4], \quad (5.23)$$

must be defined. The call to the IDL routine is then

$$\text{Czz} = \text{A_CORRELATE}(\text{zsurf}, \text{lagindex}, /COVARIANCE). \quad (5.24)$$

The IDL routine then returns an array of autocovariances at the lags shown in Eq. (5.21). These values are shown by the red dots in Fig. 5.6(d). This **Czz** array returned by A_CORRELATE has the same math order as the **lagindex** array. This array must next be shifted into the FFT order via the IDL shift function:

$$\text{CzzFFT} = \text{SHIFT}(\text{Czz}, -N/2 + 1). \quad (5.25)$$

This array can now be given to the IDL FFT routine:

$$\text{S2s} = \text{FFT}(\text{CzzFFT}). \quad (5.26)$$

The resulting **S2s** array is a complex 8-element array. The real part of **S2s** is $S_{2s}(u)$, with the frequencies in FFT order. The imaginary part is zero (to within a bit of roundoff error; values are typically less than 10^{-10}). This array is shifted back to math order and divided by $\Delta\nu$ to get the array plotted as the red dots in Panel (e) of the figure:

$$\text{S2splot} = \text{REAL_PART}(\text{SHIFT}(\text{S2s}, N/2 - 1))/\Delta\nu. \quad (5.27)$$

It is always informative to take an “information count” of such operations. We started with a two-sided spectrum of 8 values. It is true that in the present time-independent simulations $S_{2s}(-\nu) = S_{2s}(+\nu)$ (except for the 0 and Nyquist frequencies, which are always special cases). However, this symmetry need not hold in general (and indeed is not the case when generating

waves that propagate downwind, as explained previously). Thus these spectrum values represent 8 independent “pieces” of information in the form of 8 real numbers.

The 8 elevations of the sea surface are likewise 8 independent pieces of information.

Finally, the 8 covariances also comprise 8 pieces of information. Similarly to the variance spectrum, there is symmetry about the 0 lag, except for the value at the largest positive lag. However, again, the fact that $C_{zz}(-\ell_r) = C_{zz}(\ell_r)$ represents two pieces of information: the value of $C_{zz}(\ell_r)$ and the fact that $C_{zz}(-\ell_r)$ has the same value.

Thus the sampled variance spectrum $S_{2s}(u)$, the generated surface $z(r)$, and the surface autocovariance $C_{zz}(r)$ all contain the same amount of information, namely 8 real numbers. The various Fourier transforms and autocorrelation function show how to convert the information from one form to another.

5.5.1 Idle Speculations

It is certainly possible to sample in different ways. For example, surface correlations can be computed for all lags from $-L + \Delta x$ to $+L - \Delta x$, which gives $2N - 1$ total $C_{zz}(\ell_r)$ values. You can then take the FFT of that covariance and get a spectrum with $2N - 1$ values. However, I can guarantee you from two weeks of misery that the spectrum so obtained does not agree with the original $S_{2s}(\nu)$ spectrum. The $N - 1$ extra points added by taking a greater range of correlations are in some way not independent of or consistent with the N independent pieces of information tallied above. That is to say, the sea surface contains only N pieces of information, and you cannot create more information simply by computing the autocovariance at more lag values. I vaguely remember reading somewhere that you should not compute autocovariances for lags greater than one-half of the data range. Note that the lag indices used above run from values of $(-N/2 + 1)\Delta x$ to $(N/2)\Delta x$, which indeed correspond to the $-L/2 + \Delta x$ to $+L/2$ data range. I suspect, but have never seen stated, that there is something going on here that is analogous to sampling at greater than the Nyquist frequency—You can do it, but it messes up the results in ways that are not immediately obvious.

Another possible way to compute the autocovariance for a given sea surface is to compute $C_{zz}(\ell_r)$ only for 0 and positive lags out to a maximum possible lag of $L - \Delta x$. This would again give N independent numbers. Autocovariances are real and even functions of the lag (symmetric about $\ell = 0$), which means that their Fourier transforms are also real and even. Since $e^{i\theta} = \cos \theta + i \sin \theta$, a Fourier transform can be written as the sum of a cosine transform plus i times a sine transform: $\mathfrak{F}\{\cdot\} = \mathfrak{C}\{\cdot\} + i\mathfrak{S}\{\cdot\}$. Here the cosine transform \mathfrak{C} is defined as in Eq. (1.8) except that $e^{-i2\pi\nu x}$ is replaced by $\cos(2\pi\nu x)$; the sine transform \mathfrak{S} is defined in the same way but with $\sin(2\pi\nu x)$ replacing $e^{-i2\pi\nu x}$. For an even function, the sine components in the Fourier transform will all be zero. Thus it seems that Eq. (5.5) could be written as $\mathfrak{C}\{C_{zz}(\ell)\} = S_{2s}(\nu)$. An example of this was seen above in the analytical computation of the Horoshenkov variance spectrum. However, there are four different algorithms for implementing the *discrete* cosine transform (DCT), which differ by how the discrete, finite- N sequence of points is assumed to be extended outside the domain for which $C_{zz}(\ell_r)$ is known. It seems that the present case of $C_{zz}(\ell_r)$, which is symmetric about $\ell_r = 0$, corresponds to the “Type I” DCT discussed at https://en.wikipedia.org/wiki/Discrete_cosine_transform or the “ y_1 ” extension seen in Fig. 2(a) of Makhoul (1980). The four different formulations of the DCT can be computed in four different ways by use of FFTs. Thus the use of a DCT in Eq. (5.5) opens a new can of worms. In any case, there is little or no penalty to be paid for sticking with a Fourier

transform evaluated by an FFT routine in order to evaluate the DFTs as needed here. As a matter of practical necessity, the internal consistency of the spectra, surfaces, and autocovariances seen in the preceding figures (and to be seen below) indicate that the sampling scheme described above is correct, even if there may be equivalent ones.

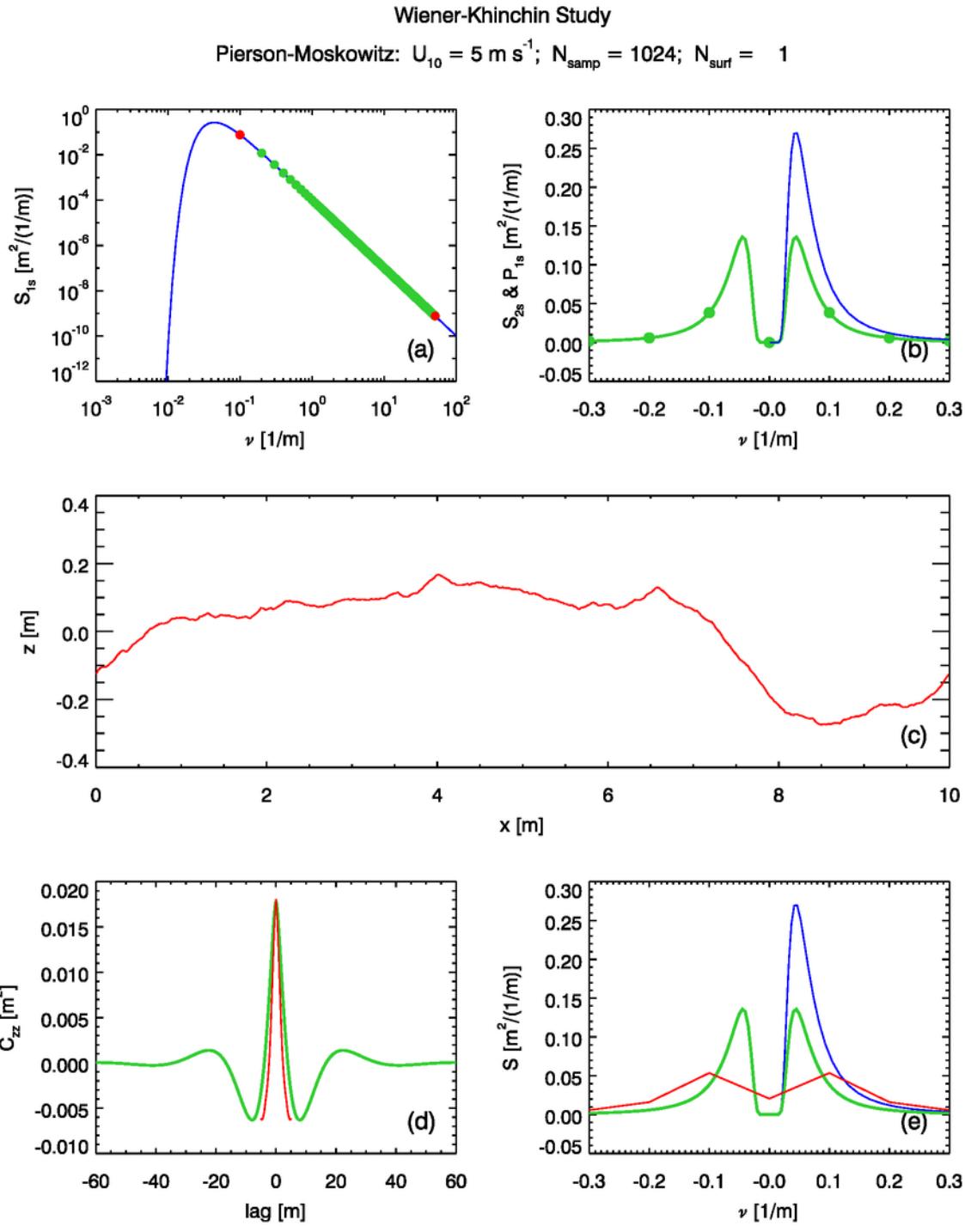
5.5.2 Lessons Learned

The preceding simulations illustrated the Wiener-Khinchin theorem starting with a variance spectrum $\mathcal{S}_{1s}(\nu)$ (the Pierson-Moskowitz spectrum) and arriving at an autocovariance $C_{zz}(\ell)$ in two ways. The first way was to construct the corresponding two-sided spectrum $\mathcal{S}_{2s}(\nu)$ and then take the inverse Fourier transform to obtain the theoretical, noise-free $C_{zz}(\ell)$ via the Wiener-Khinchin theorem. The second way was to use $\mathcal{S}_{2s}(\nu)$ to generate a large number of random sea surfaces. The autocovariance of each random surface was computed by Eq. (5.3), and then the ensemble-average autocovariance was computed as the average of the individual autocovariances.

It is important to note that the size L of the spatial region and the number of sample points N must be chosen with care. As a rule, L must be large enough to cover several wavelengths of the longest wave that contains a significant amount of the total variance. N must be large enough that the sampled points on the variance spectrum then reach far enough into the high-frequency end of the spectrum to cover the entire part of the spectrum that contributes a significant amount to the total variance. To see the effects of inadequate sampling, suppose we are concerned only with the short gravity and capillary waves, which are optically the most important because they have the highest slopes. If we are interested only in waves of wavelength ~ 1 m down to ~ 1 cm, it might then seem reasonable to let $L = 10$ m and $N = 1024$, which give $\Delta x \approx 1$ cm. The shortest resolvable wavelength is then $2\Delta x \approx 2$ cm. Figure 5.7 shows an example surface and other quantities for this case.

However, now $\Delta\nu = 1/L = 0.1$ m $^{-1}$ and the spectrum is sampled only at widely spaced points (the green dots in Panel (b)) that largely miss the peak of the variance spectrum. Consequently, the generated surface has too little variance compared to the real sea surface described by this spectrum. Also, the sample autocovariance function, shown by the red curve in Panel (d), computes the autocovariances only for lags up to $L/2 = 5$ m. This lag range does not capture the full autocovariance features of the real surface, for which the autocovariance is non-zero out to lags of ~ 40 m, as shown by the green curve in Panel (d). The spectrum estimated from the sample autocovariance (the red curve in Panel (e)) does reproduce the sampled spectrum (the green dots in Panel (b)), but this spectrum is not representative of the real sea surface.

Picking $L = 100$ m, as in the previous simulations, seems adequate for a wind speed of 5 m s $^{-1}$. This can be seen from the leftmost red point in Panel (a) of the previous plots, which is to the left of the spectrum maximum. However, $N = 1024$ then gives $\Delta\nu = 0.01$ m $^{-1}$, and the last sampled point corresponds to a shortest resolvable wavelength of $2\Delta x \approx 20$ cm. If that is not adequate resolution for the problem at hand, there are two options. One option is to increase N , which costs more computer time to evaluate the FFTs. Increasing N by a factor of 8 to $N = 8192$ then gives $2\Delta x \approx 2.4$ cm, which might be adequate for the problem at hand. The time for an FFT is proportional to $N \log_2 N$, so that increase in N comes at a factor-of-ten increase in run time, which can be prohibitive if many surfaces must be generated. The other option is to account for the unsampled variance in some other way. One technique for doing that is to adjust the spectrum



WK_PM2H_nu.pro

Figure 5.7: Example simulation with inadequate sampling of the variance spectrum.

to account for the missing variance while keeping N relatively small. One technique for doing this is described in §2.6 and in Molley (2015).

These results can be summarized as follows:

- The size of the spatial domain, L , must be large enough to cover at least several wavelengths of the wave of maximum variance. The value of L sets the fundamental frequency $\nu_1 = 1/L$, which equals the frequency interval $\Delta\nu$.
- For the given fundamental frequency ν_1 , the number of spatial samples, N , must be large enough that the highest (Nyquist) frequency, $\nu_{N/2} = (N/2)\Delta\nu$ covers the domain of the variance spectrum for which the variance is non-negligible. This highest sampled frequency must also cover the highest frequency (shortest wavelength) needed for the problem at hand. The minimum resolvable wavelength is $2\Delta x = 2L/N$.

Of course, the need for large L and large N comes as the cost of increased computer time. Experimentation is necessary to determine what values are required for a particular physical situation.

5.6 Turbulence-generated Water Surfaces

This section now considers the generation of random water surfaces beginning with the analytical autocovariance function of Horoshenkov et al. (2013), which is seen in Eq. (5.13). This particular autocovariance has an analytical spectrum, Eq. (5.15), which was computed in §5.3.1. Thus, for surface generation, the two-sided spectrum of Eq. (5.15) simply replaces the two-sided Pierson-Moskowitz spectrum used in the previous section, and the surface-generation calculations proceed as described in the previous chapters. The autocovariance function is then not needed. However, in general, if only the autocovariance is known or measured, then the needed variance spectrum must be obtained via the Wiener-Khinchin theorem. In the present study, knowing both the autocovariance and the spectral density as analytical functions provides a powerful check on the discrete numerical calculations of the same quantities.

The application of the above results is straightforward. If only the autocorrelation, rather than the autocovariance, is given, then a separate value of the surface elevation variance must be known. For the Horoshenkov study, a typical value of the surface variance is $C_{zz}(0) = 2.5 \times 10^{-7}$ m². (This is extremely small by oceanographic standards, but the surface waves in the Horoshenkov experiment had amplitudes of order 1 mm.) The previously cited parameter values of $\sigma_w = 0.22$ m and $L_o = 0.17$ m are used here. Since the characteristic spatial scales of σ_w and L_o are of order 0.2 m, a spatial region of length $L = 4$ m should be adequate to capture the spatial features of these surfaces. An N value of 1024 then gives the smallest resolvable wavelength as $2\Delta x \approx 0.8$ cm, which is the scale of capillary waves. (Capillary waves have wavelengths in the range of a few millimeters to 2 cm.)

Figure 5.8 shows an example simulation based on the Horoshenkov variance spectrum (5.15). The layout is the same as for the Pierson-Moskowitz figures. Panel (d) of the figure contains three autocovariance plots: The green curve is the inverse DFT of the sampled variance spectrum, which is shown in green in Panel (b). The red curve is the ensemble average autocovariance of 1000 water surface simulations. The purple curve is the theoretical autocovariance of Eq. (5.13). These three curves are indistinguishable at the scale of this plot. This nearly perfect agreement between

autocovariance derived in three different ways indicates that the various numerical calculations are almost without doubt being done correctly.

The red curve in Panel (e) of the plot shows the variance spectrum derived via the Wiener-Kinchin theorem as the DFT of the ensemble-average autocovariance (the red curve in Panel (d)). Again, this curve is almost indistinguishable from the theoretical autocovariance, which is shown in green. Again, this agreement indicates that the DFTs are being computed correctly.

5.7 Two-dimensional Water Surfaces

My IDL and Fortran codes for generation of two-dimensional, time-independent water surfaces are formulated using a one-sided, two-dimensional elevation variance spectrum $\Psi(k_x, k_y)$ of the form (see Eq. B.6)

$$\Psi(k_x, k_y) = \frac{1}{k} S_{1s}(k) \Phi(k, \varphi). \quad (5.28)$$

Here $S_{1s}(k)$ is a one-sided omnidirectional spectrum and $\Phi(k, \varphi)$ is a nondimensional spreading function (recall §). To generate a 2-D, time-independent surface using the Horoshenkov model, the two-sided omnidirectional spectrum of Eq. (5.17) is multiplied by 2 to obtain a one-sided spectrum, which my code evaluates only for the non-negative k_x values, i.e. for $-\pi/2 \leq \varphi \leq \pi/2$. The code then divides the result by 2 to get a two-sided spectrum and evaluates the $-k_x$ half plane of values by symmetry. Thus it is easy to replace an omnidirectional oceanographic $S_{1s}(k)$ spectrum with that of Horoshenkov. There remains only the issue of what to use for a spreading function. There is no information about the spreading functions of turbulence-generated waves in the Horoshenkov et al. (2013) paper. There is no doubt some flow-induced difference in the waves in the “down-river” vs “cross-river” directions, just as there is in the “down-wind” vs “cross-wind” directions for wind-generated waves. However, pending further information on that difference, it is probably reasonable to use a frequency-independent, isotropic spreading function, $\Phi(\nu, \varphi) = \frac{1}{2\pi}$. With that assumption, two-dimensional surfaces can be generated.

Figure 5.9 shows an example two-dimensional, turbulence-generated surface created with the $\sigma_w = 0.22$ m, $L_o = 0.17$ m and $C_{zz}(0) = 2.5 \times 10^{-7}$ m² values used for Fig. 5.8. This particular 2-D surface realization has an elevation variance of 2.48×10^{-7} m², which is close to the value of $C_{zz}(0)$ value used as input to the Horoshenkov spectrum. It is also noted that along any slice through the surface, there are about two dozen “bumps” in 4 m, just as seen in the 1-D surface realization of Fig. 5.8. Figure 5.10 shows the slice through the 2-D surface at $y = 2$. This surface is qualitatively like that of the middle panel of Fig. 5.8. These results indicate that the 2-D calculations are correct.

The visual appearance of the Horoshenkov surface is strikingly different from the wind-generated sea surface seen in Fig. 5.11, which is for a 5 m s⁻¹ wind speed. In these plots, the surfaces have a factor-of-8 difference in the scaling of the surface elevation relative to the horizontal: 0.02 m vertical to 4 m horizontal = 0.005 for the Horoshenkov surface compared 4 m to 100 m = 0.04 for the wind-blown surface. This is purely for the visual appearance of the 3D perspective plots. The Horoshenkov surface is actually quite smooth, with an average wave facet slope of only about 0.6 deg. The wind-blown surface has an average slope angle of about 3.7 deg in the along-wind direction and 2.9 deg in the cross-wind direction. (Keep in mind that for this simulation $\Delta x = 100/1024$, so the smallest resolvable wave has a wavelength of about 20 cm. Thus the smallest waves, which

Wiener-Khinchin Study of Horoshenkov Surfaces

$$\sigma_w = 0.22; L_0 = 0.17; C_{zz}(0) = 2.50e-007; N_{\text{samp}} = 1024; N_{\text{surf}} = 1000$$

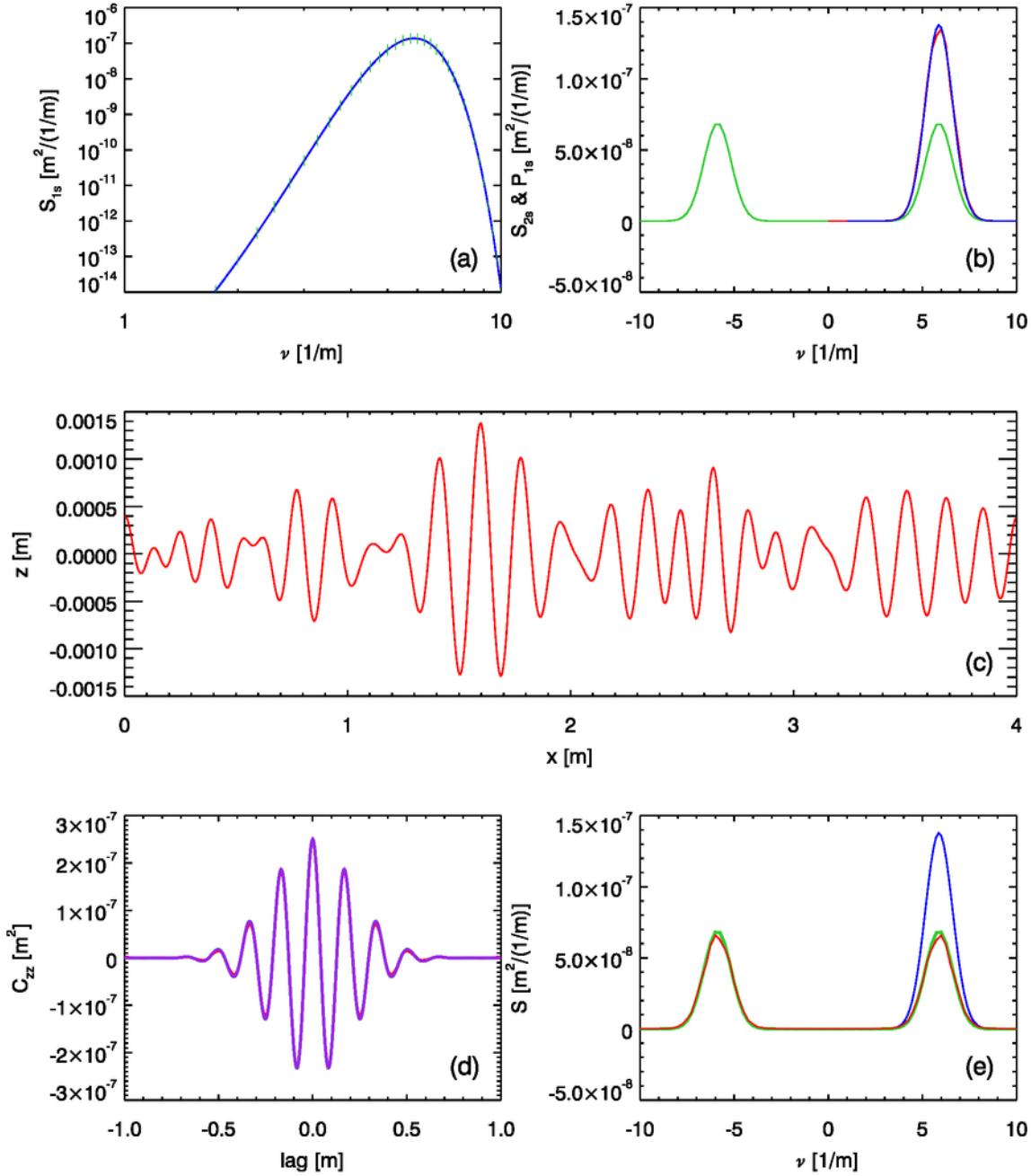


Figure 5.8: Example of a turbulence-generated water surface based on the autocovariance function of Horoshenkov et al. (2013). Compare the qualitative appearance of panel (c) with the sea surface shown in Fig. 5.2(c).

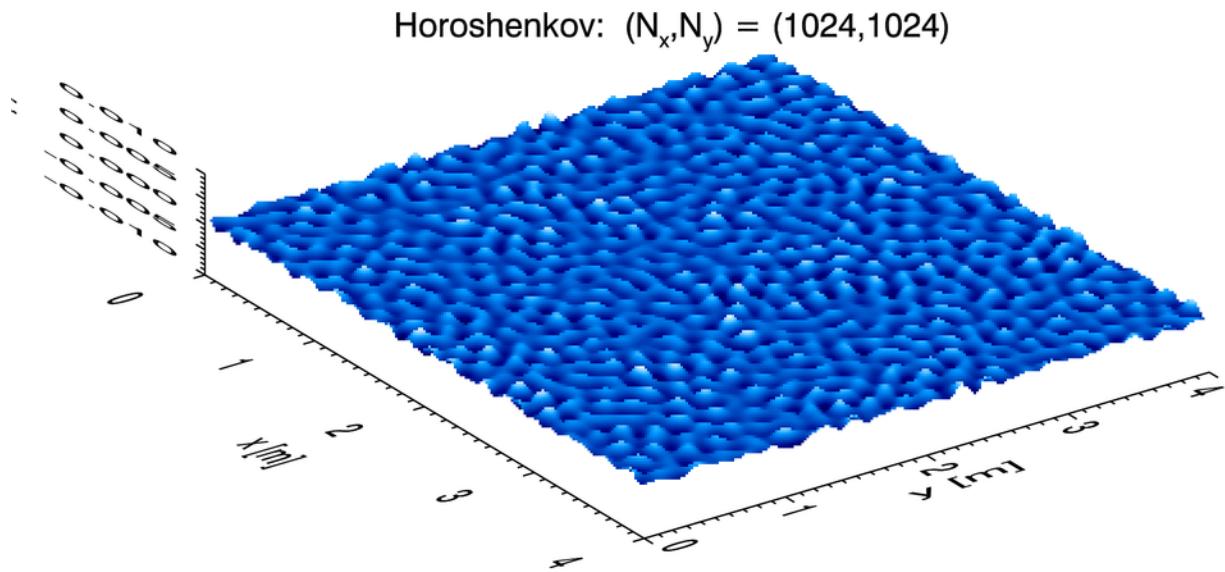


Figure 5.9: A 2-D turbulence-generated surface. White is large positive surface elevations (wave crests) and dark blue is large negative values (wave troughs).

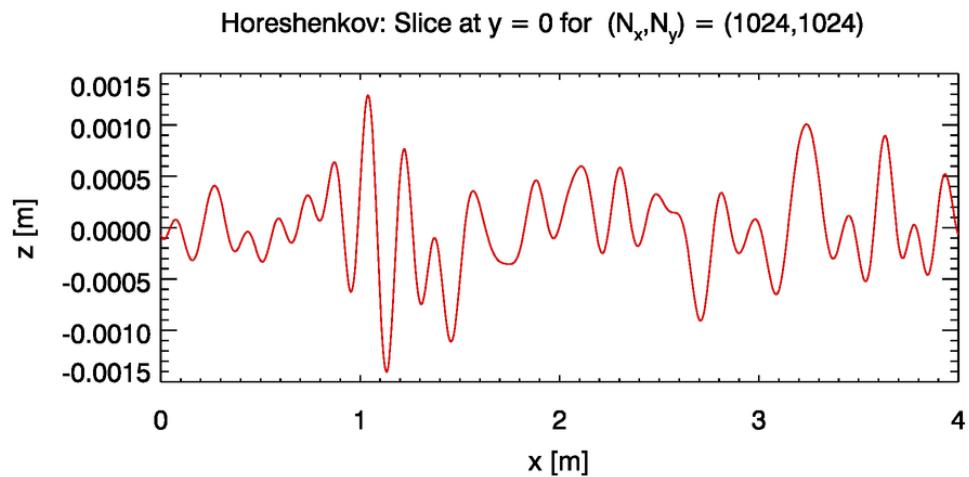


Figure 5.10: A slice through the surface of Fig. 5.9 at $y = 2$.

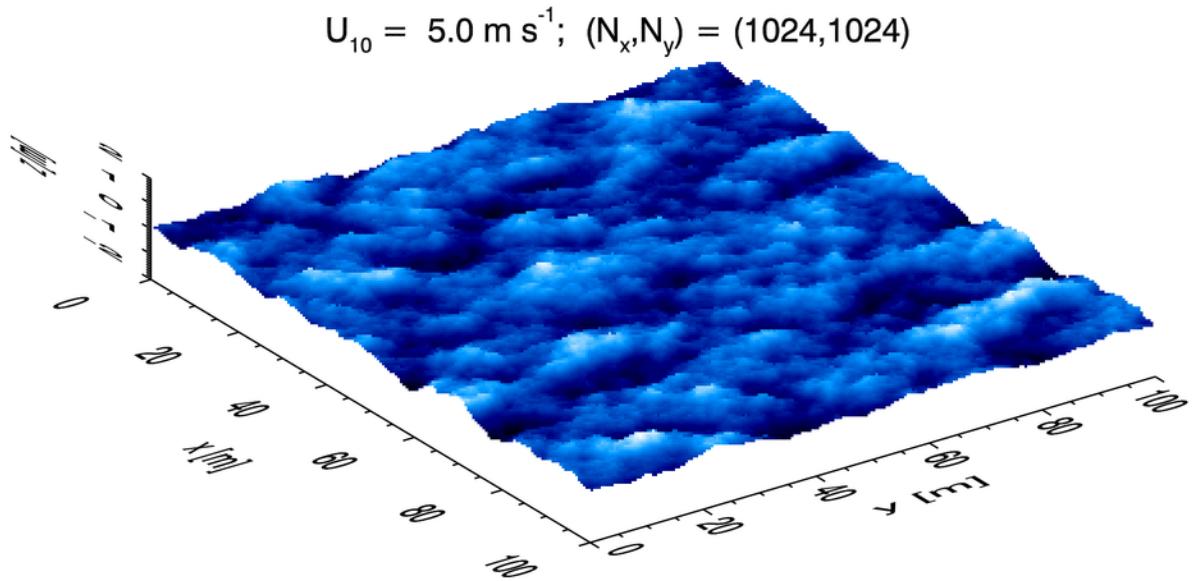


Figure 5.11: A wind-generated surface for a wind speed of 5 m s^{-1} . Compare with Fig. 5.9.

can have large slopes, are not resolved. An actual sea surface will therefore have larger average slopes.) Thus the Horoshenkov surface is smoother than the wind-blown surface. These differences highlight the original hypothesis that led to this note: turbulence-generated water surfaces may have significantly different optical reflectances than wind-generated surfaces. That hypothesis can be tested by ray tracing calculations based on surfaces like those of Figs. 5.9 and 5.11.

APPENDIX A

Computation of Discrete Fourier Transforms

As has been noted, the fast Fourier transform (FFT) is a numerical algorithm that computes the discrete Fourier transform (DFT) of a sequence of N complex numbers. The number of operations required for unoptimized evaluation of the DFT is proportional to N^2 , where an “operation” can be thought of as a multiplication followed by an addition. This is referred to as an “order N^2 ” calculation, denoted $\mathcal{O}(N^2)$. If N is a power of 2, the FFT requires only $\mathcal{O}(N \log_2 N)$ operations, which is an enormous computational savings when N is large. This appendix illustrates via numerical examples the use of the FFT algorithm for the computation of DFTs when N is not a power of 2. It is shown that modern FFT routines can compute DFTs for any N , but with an efficiency that depends strongly on N . Most importantly, it is also shown that “padding” or extending a data set with values of 0 to obtain a number of data values that is a power of two, and then using the FFT on the padded data, is absolutely wrong. This data extension simply gives an efficient calculation of a wrong answer.

When generating sea surface realizations for scientific or computer graphics purposes, the value of N usually can be chosen to be a power of two, and the FFT can be employed. However, when working with real data, the value of N is often fixed by circumstances. For a contrived example, suppose you designed an experiment to measure the sea surface elevation every Δt seconds at a given point, or every Δx meters along the surface at a given time. You wisely planned to obtain $N = 2048 = 2^{11}$ measurements so that you could use the FFT to convert those measurements to a variance spectrum as described previously. However, the instrument died after only 1975 measurements were obtained. What is to be done with this data set of 1975 values? There are several possibilities:

1. You can discard the 1975 data values, fix the instrument, and repeat the experiment to get 2048 measurements. That might or might not be possible, but in any case is a waste of data.
2. You can use only the first $1024 = 2^{10}$ data values in the FFT. That computation will be efficient, but still wastes part of the data.
3. You can do a “brute-force” calculation of the DFT for $N = 1975$. That will make full use of the data. The only penalty to be paid for not having N be a power of 2 is extra computation time.

4. You can give the array of 1975 values to an FFT routine and see what happens. This will be seen to be the best option, assuming that you have a sophisticated FFT routine such as those available in IDL, Matlab, or the Fortran 90 routines in the FFTPACK package developed at the National Center for Atmospheric Research. If that is the case, the FFT routine will default to a DFT computation, and the resulting Fourier amplitudes will be the same as those obtained from the unoptimized DFT with $N = 1975$.
5. You can pad the valid data set with zeros for values 1976, ..., 2048, and then take the FFT of the 2048 data values. It will be shown below that *this option gives incorrect results, in that the computed Fourier amplitudes are not the same as those obtained from the DFT with $N = 1975$* .

I wrote this appendix because I have occasionally heard people say that “you just pad your data with zeros to get a power of 2, and then take the FFT.” This is simply wrong because the data are sea surface elevations, and a sea surface with a stretch of zero values (a level surface) following a wavy surface simply isn’t physically realistic and isn’t the same as a surface completely covered by waves.

A.1 Example Calculations

The initial figures of this section show the results of DFT and FFT calculations for the case of $N = 16$. This is few enough points that the details can be easily plotted. Simulations with fewer points are then discussed.

An ad hoc sea-surface wave profile was constructed as in §2.3 using formula (2.4). Figure A.1 repeats Fig. 2.1 for easy reference. The upper left panel of Fig. A.1 shows the surface generated in this manner for $L = 10$ m, $N = 16$, and a particular set of random phases. The thin colored lines show the $N/2 + 1 = 9$ waves for each of the frequencies. The blue line is the wave for the fundamental frequency $\nu_f = 1/L = 0.1$ m $^{-1}$; the thin black line is the two-point wave at the Nyquist frequency $\nu_{Ny} = 1/(2\Delta x) = 0.8$ m $^{-1}$; the purple line is the constant $j = 0$ wave, which is set to $z = 0$ for the mean sea surface. The black dots connected by the thick black line show the sum of the individual waves. These points represent a discrete sampling of the continuous sea surface elevation.

I wrote a routine in IDL to evaluate the DFT Eqs. (1.14) and (1.15) without regard for any of the computational efficiencies used in the FFT. This unoptimized routine gives the “worst-case” baseline for comparison with more efficient IDL FFT calculations. The DFT of the sea surface (the dotted black line) was then computed with this DFT routine. The real and imaginary parts of the resulting Fourier amplitudes $\hat{z}(u)$ are shown in the upper-right and lower-left panels, respectively, of Fig. A.1. Each spatial frequency is identified by the color. Values for positive frequencies are shown by solid dots, and the values for negative frequencies are open circles. As discussed in §2.3, the upper-right panel shows that the real part of the amplitudes is an even function of frequency, i.e. $\hat{z}(-\nu_u) = \hat{z}(+\nu_u)$. The lower-left panel shows that the imaginary part of the amplitudes is an odd function of frequency, i.e. $\hat{z}(-\nu_u) = -\hat{z}(+\nu_u)$. This illustrates that the Fourier amplitudes of a real function are Hermitian. The amplitude at $\nu = 0$ is shown by the black square. This amplitude represents the “DC signal” and is 0 because the mean sea level was set to 0. The black

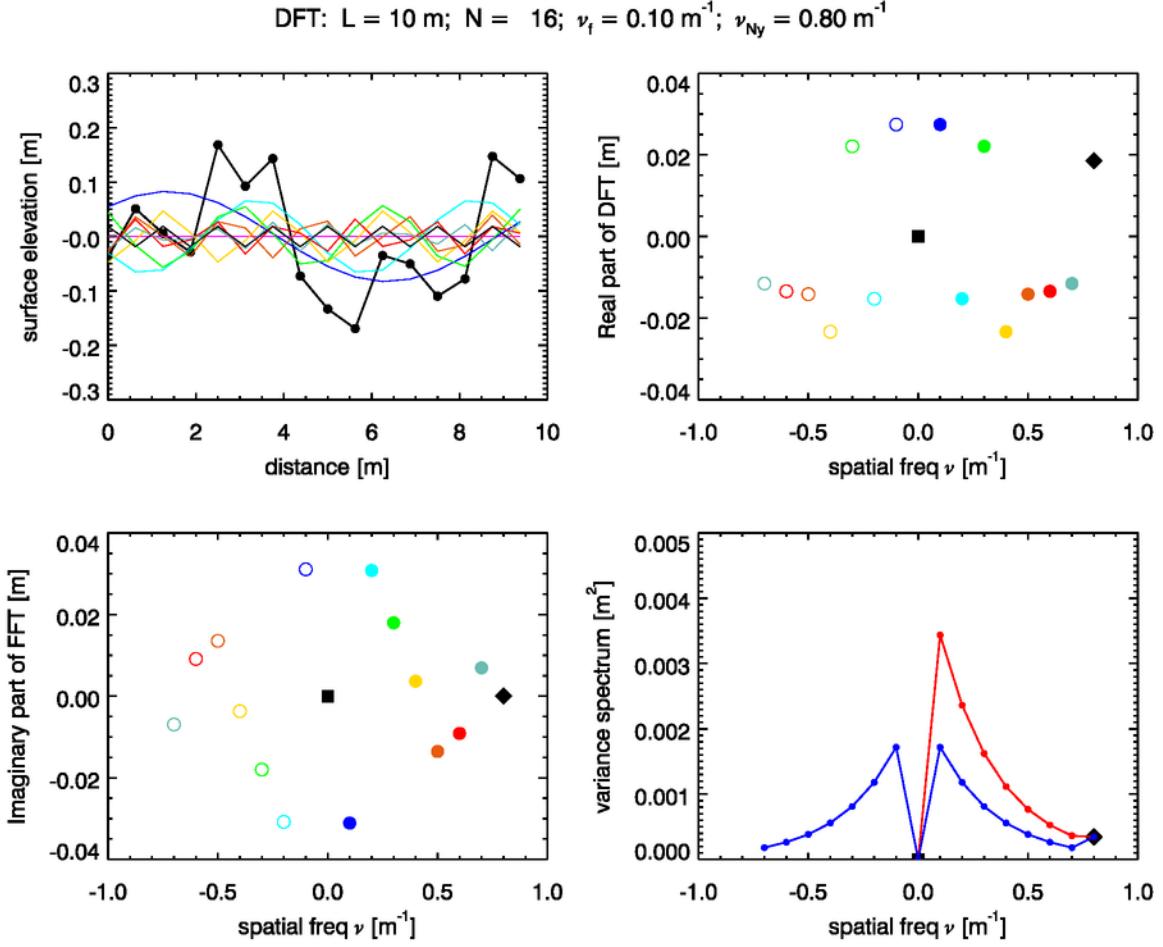


Figure A.1: Example DFT calculations. The black line with dots in the upper-left panel shows the sea surface profile; the thin lines are the component waves at the difference frequencies. The upper-right panel shows the real part of the DFT, and the lower-left panel shows the imaginary part. The lower-right panel shows the one-sided (red) and two-sided (blue) variance spectra.

diamond at $\nu = 0.8 \text{ m}^{-1}$ is the amplitude at the Nyquist frequency, which IDL stores in the array position for the largest positive frequency. This value is non-zero for the real part of $\hat{z}(\nu_{Ny})$ and zero for the imaginary part. This is because the real part of $\hat{z}(u)$ describes the cosine terms of the Fourier decomposition of the surface elevations, and the imaginary part represents the sine terms. The Nyquist frequency contains only the two-point cosine wave (there is no two-point sine wave), so the imaginary part of $\hat{z}(\nu_{Ny})$ is always zero. The two-sided variance spectrum $S_{2s}(u)$ given by Eq. (??) is shown in blue in the lower-right panel of the figure. The red dots show the corresponding one-sided spectrum S_{1s} .

When the sea surface shown in the upper left panel of Fig. A.1 is given to the IDL FFT routine, the resulting amplitudes are exactly the same as those given by my brute-force DFT routine, but the run time is much faster. The timing for these two routines will be discussed in the next section.

Now consider the case of N not a power of two. In analogy to the hypothetical experiment

with the instrument failure, the last two points of the sea surface profile were omitted. This gives a surface with $N = 14$ points, which are shown in the upper-left panel of Fig. A.2. Note first that because there are now only 14 points, there are only 14 spatial frequencies. The sampling interval Δx and frequency spacing $\Delta\nu$ remain the same, but the Nyquist frequency is now at $\frac{N}{2}\Delta\nu = 0.7 \text{ m}^{-1}$ because there are fewer points.

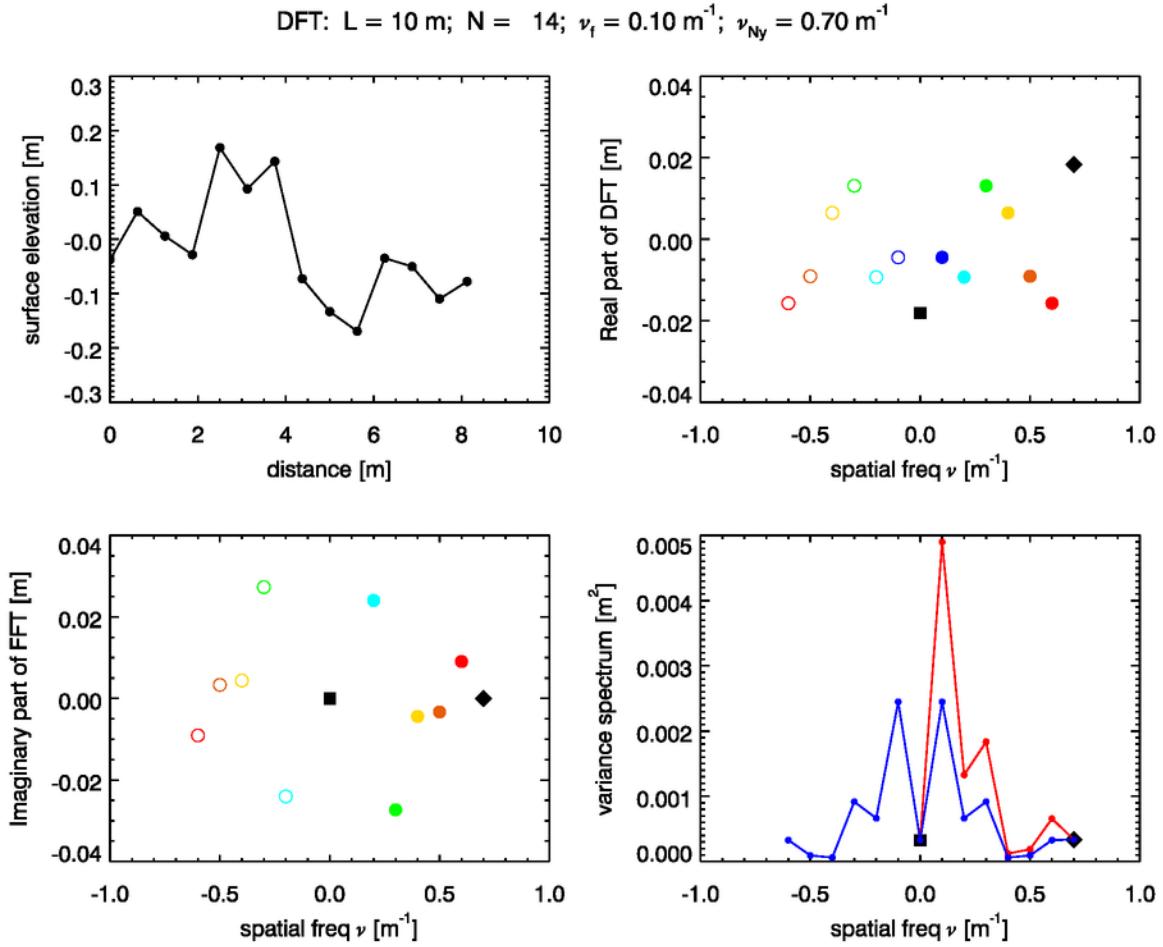


Figure A.2: DFT calculations for the first 14 points of the surface of Fig. A.1.

The amplitude and variance spectrum at $\nu = 0$ are no longer zero. This is because simply omitting the last two points has made the mean of the remaining points non-zero. In this particular case, the mean sea level is negative because the omitted points were both positive. If the 14-point surface is re-centered to give a mean sea level of zero, then the $\hat{z}(\nu = 0)$ value changes to zero without any change to the values at non-zero frequencies.

Although the 14 points of Fig. A.2 are exactly the same as the first 14 points of Fig. A.1, all amplitudes are entirely different. This is simply because these are really two different sea surfaces when periodically extended beyond the plotted points. The amplitudes for the 14-point surface fully describe all of the information contained in the 14 values of the sea level elevation, and the 14-point DFT makes full and correct use of all of the available data. However, the information

contained in the 16-point surface is not fully contained in the first 14 points of the surface. The variance spectrum for the 14 point surface seen in the lower-right panel is therefore only qualitatively similar to that of 16-point surface. The variance of the 14-point surface is distributed among fewer frequencies, with changes to each amplitude.

If these 14 points are given to the IDL FFT routine, the results are exactly the same as seen in Fig. A.2. Even though the number of points was not a power of 2, the IDL FFT routine returned the correct DFT, for reasons discussed below.

Now consider the question that led me to write this appendix: What happens if the 14-point surface is padded with zeros to get a 16 point surface? This surface is shown in Fig. A.3. The red dots in the upper-left panel show the two values that were set to zero in order to obtain 16 points for the FFT. These 16 points were then given to the IDL FFT routine. It is clear from comparison of Figs. A.1, A.2, and A.3 that the Fourier amplitudes (hence the variance spectrum) of the zero-padded surface are entirely different from the amplitudes of either the original 16-point surface or the truncated 14-point surface. A surface with some points set to zero is physically a different sea surface from either the original surface or the truncated surface, so the Fourier amplitudes are different. In other words, *a data set cannot be naively padded with zeros in order to obtain enough values for an efficient power-of-two FFT*. In any case, given the ability of most FFT routines to compute DFTs for any value of N , there is no need for padding a data set.

A.2 Timing

How the FFT algorithm works is described in numerous texts and web sites. A good explanation can be viewed at https://www.youtube.com/watch?v=EsJGuI7e_ZQ. The programming details can be quite complicated and need not be described here, but the basic idea is as follows. If N can be factored into a product of small prime numbers, then the N -point DFT can be reduced to a number of smaller DFTs, each of which can be computed in very little time. The time required to compute several small- N DFTs and then combine the results to obtain the DFT for the original large- N value is much less than the time for one large- N DFT. This process is most efficient when N is a power of 2, in which case the factorization of $N = 2^m$ is just a product of m twos. However, if N can be factored into a product of small primes, then the computations remain efficient. For example, if $N = 15$, the prime factors are 3 and 5, and two DFTs of lengths 3 and 5 can be computed and reassembled more quickly than one DFT of length 15. However, if $N = 17$, there is no prime factorization (other than 1 and 17), and the FFT algorithm is forced to do a single 17-point DFT. However, there can still be a significant speed improvement over a unoptimized DFT routine because FFT routines are written with great care to speed up the calculations. For example, these routines typically pre-compute values of $W_N = \exp(i2\pi/N)$, so that the exponentials in Eqs. (1.14) and (1.15) can be evaluated as integer powers of W_N . That is, $\exp(i2\pi ru/N) = W_N^{ru}$, with no repeated evaluation of sines and cosines needed.

The computation times required for a brute-force DFT and an FFT are compared as follows. For a given N value, a data set was generated using a Gaussian random number generator with zero mean and unit variance to generate a random $z(r)$ value at each x_r grid point. The DFT or FFT of the set of random points was then computed and timed.

Figure A.4 shows the time required to compute one DFT using the unoptimized DFT routine.

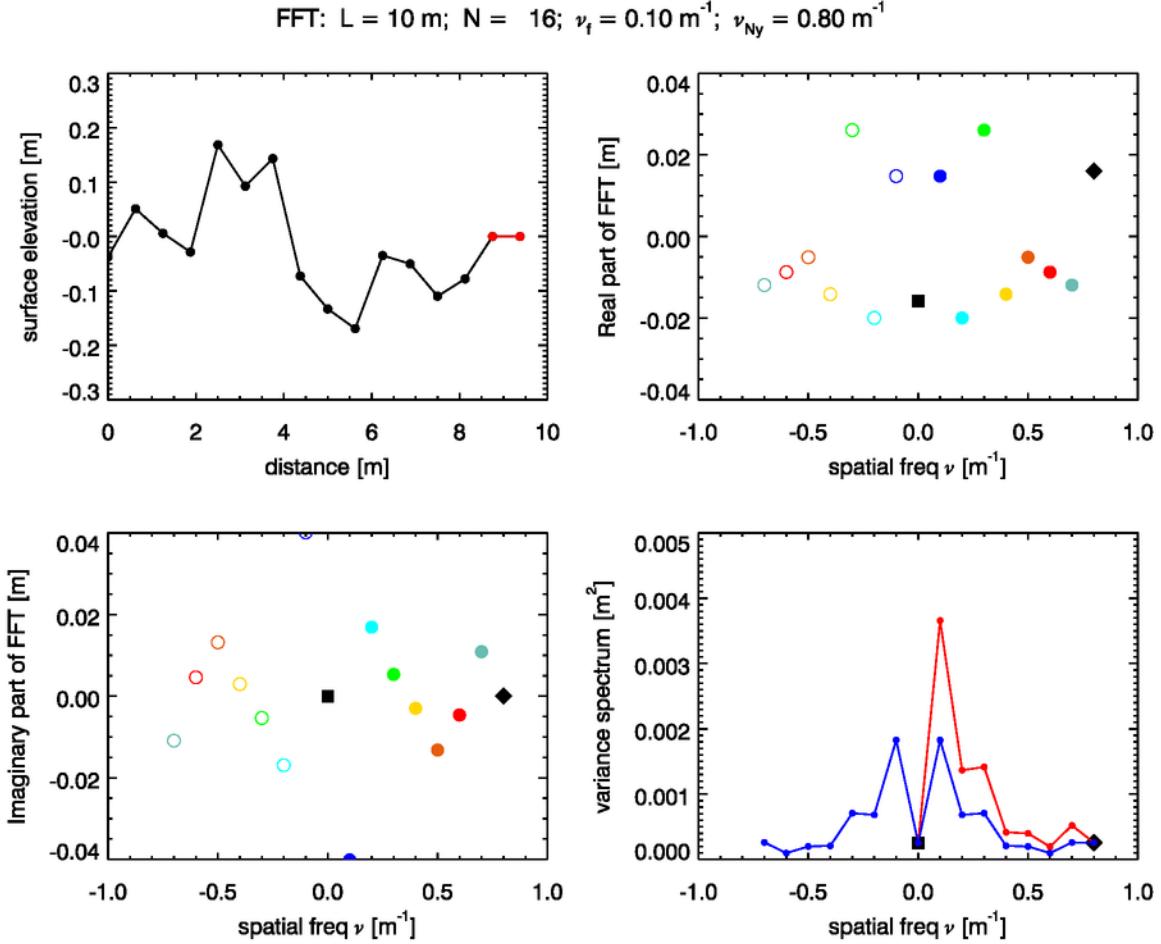


Figure A.3: FFT results for the 14-point surface of Fig. A.2 padded with zeros to get a 16-point data set. The two red points in the upper-left panel show the sea surface values that were set to zero. Note that the amplitudes for the zero-padded surface are entirely different from the amplitudes of either the original surface of Fig. A.1 or the truncated surface of Fig. A.2.

As expected, the run time increases roughly as N^2 . Note that a single DFT for $N = 4096$ requires about 1.15 seconds (on a 2.4 GHz laptop).

Figure A.5 shows the times required by the IDL FFT routine to computed 1000 transforms for N values from 1000 to 1200. The run times vary greatly with the number of points N . The red dots indicate N values that are prime numbers. The run times for prime- N FFTs are always much slower than for composite- N values. Many of the composite- N times are almost as fast as the time for the power-of-two $N = 1024$ case (0.047 sec), which is shown by the gold dot. For example, the time for $N = 1040$ is almost the same as the 1024 value. 1040 factors into $2 \times 2 \times 2 \times 2 \times 5 \times 13$, so the calculation requires four 2-point DFTs (very fast), one 5-point, and one 13-point DFT. However, $N = 1042$ factors into 2×521 , so a 521-point DFT (slow) is required, and the time for $N = 1042$ is 0.593 sec. The prime- N times are all more than one second for this range of N values.

Figure A.6 shows the times for 1000 transforms using the FFT routine, for the same range of

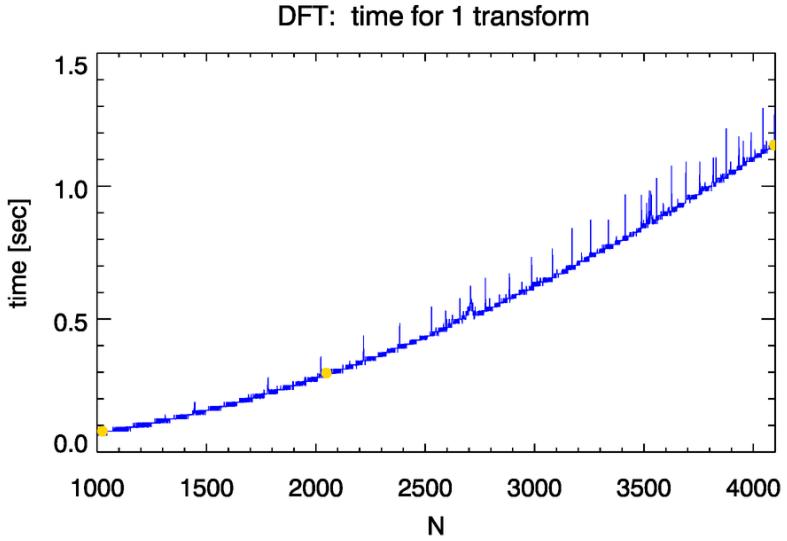


Figure A.4: Time required to compute 1 DFT using an unoptimized DFT routine for N between 1000 and 4100.

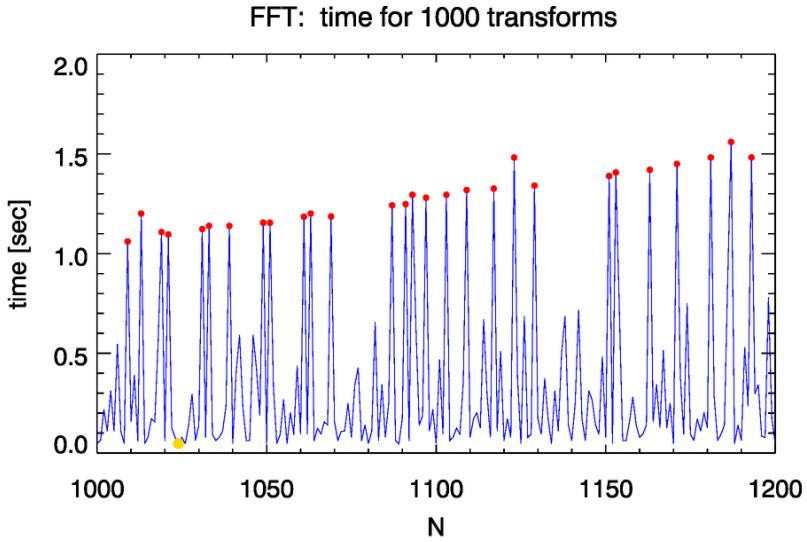


Figure A.5: Time required to compute 1000 DFTs using the IDL FFT routine. The red dots indicate values of N that are prime numbers.

N values as Fig. A.4. It is clear that the prime- N times are proportional to N^2 , but the shortest times for the composite- N values can be much less. The gold dots at $N = 1024, 2048$, and 4096 indicate the power-of-two N values. Recall from above that a single unoptimized DFT for $N = 4096$ required 1.15 sec. The FFT required only 0.17 sec for 1,000 transforms. This is an improvement in run time by a factor 6800 per transform for $N = 4096$. This decrease in run time comes from

a factor of $N^2/(N \log_2 N) = 341$ and another factor of 20 from the programming efficiency of the IDL routine, which does not simply plug numbers into sines and cosines to evaluate the complex exponentials.

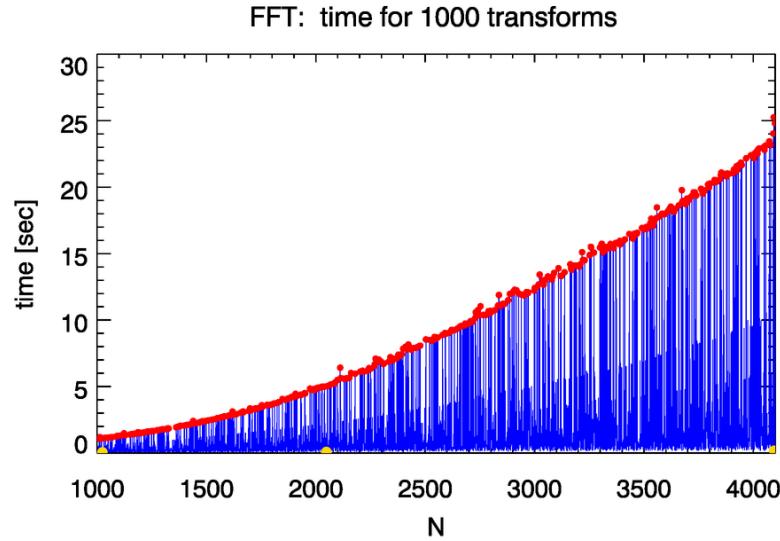


Figure A.6: Time required to compute 1000 DFTs using the IDL FFT routine for N between 1000 and 4100.

The preceding simulations support two conclusions:

1. Data should not be padded with zeros for the purpose of obtaining a number of data points equal to a power of two in order to minimize the time required to compute the DFT of the data using an FFT routine. Doing so simply gives a fast calculation of a wrong answer.
2. Commonly used FFT routines can compute the DFT for any value of N . They are extremely efficient for composite N values that have small prime numbers as factors. However, even for N values that are prime, they are still much more efficient than a naive evaluation of the DFT.

APPENDIX B

Wave Spectra

The entire business of wave spectra can be exceedingly confusing. Just as for the DFT techniques to generate sea surfaces, this is well known material. However, journal articles always assume that the reader already understands the underlying ideas and mathematics. It is therefore worthwhile to review the results needed for this tutorial. The best-written reference I've found for the development of wave spectra is Holthuijsen (2007), who is very careful in his notation and terminology. The notation used below is chosen to conform to that used in Elfouhaily et al. (1997), which will often be referred to as the ECKV spectrum, after the last letters of the authors' names.

The figures of the previous chapters were illustrated using two commonly used wave variance spectra. The one-dimensional surfaces of §2.5 used the 1-D Pierson-Moskowitz spectrum, and the two-dimensional surfaces of §3.2.2 and Chapter 4 used the 2-D spectrum of Elfouhaily et al. (1997). After the introductory overview in the next section, the remaining two sections present these two specific wave spectra for easy reference.

B.1 Overview of Wave Spectra

As we saw in §2.1 and 2.2, there is a close connection between the variance of the sea surface waves and the energy per unit area contained in the waves. For a discretely sampled sea surface with elevations $z(r)$, the discrete Fourier transform (1.14) gives the corresponding amplitudes $\hat{z}(u)$. The discrete variance spectrum is then the function $S(u) = |\hat{z}(u)|^2, u = 0, \dots, N - 1$, which is often (but not here!) loosely referred to as the discrete energy or power spectrum.

Care is required to formulate wave spectra for continuous variables. As in Eqs. (1.2) and (1.6), we can write the continuous surface as a sum of sinusoids, e.g.

$$z(x) = \sum_{n=0}^{\infty} z_n(x) = \sum_{n=0}^{\infty} A_n \cos(k_n x + \phi_n). \quad (\text{B.1})$$

As we saw in §2.1, the variance of the sinusoid with frequency k_n is $\frac{1}{2}A_n^2$. The waves of different frequencies are independent, so the total variance of the surface can be written as the sum of the

variances of the individual waves:

$$\text{var}\{z\} = \sum_{n=0}^{\infty} \frac{1}{2} A_n^2. \quad (\text{B.2})$$

Now let Δk_n be a frequency interval centered on frequency k_n , whose sinusoid has amplitude A_n . We then define

$$\mathcal{S}(k = k_n) \equiv \lim_{\Delta k_n \rightarrow 0} \frac{\frac{1}{2} A_n^2}{\Delta k_n}. \quad (\text{B.3})$$

In this definition, we keep in mind that each A_n is associated with a particular frequency k_n , and that the limit operation holds for each value of n . We are thus defining a function of the spatial frequency, which becomes a continuous function of k as the bandwidth Δk_n goes to zero.

The quantity $\mathcal{S}(k)$ is called the *omnidirectional variance spectrum*. “Omnidirectional” means that there is no reference direction (e.g., a direction relative to the wind direction) included in the quantity. This is the case if a wave record is made at a single point as a function of time: the waves go past and their elevations are recorded, but no information is obtained on the direction the waves are traveling. $\mathcal{S}(k)$ is also called the omnidirectional elevation spectrum for obvious reasons. As is to be expected, there is no uniformity of notation for this spectrum, but \mathcal{S} seems to be the most common symbol—and what is used in both Pierson and Moskowitz (1964) and Elfouhaily et al. (1997) in the next sections—so that what I will use here. (\mathcal{E} seems to be the second-most-common symbol and is used in Holthuijsen (2007) and several other papers on my desk.) The units of $\mathcal{S}(k)$ are clearly $\text{m}^2/(\text{rad}/\text{m})$. Equations (B.2) and (B.3) show that integrating the omnidirectional variance spectrum over all frequencies gives the total variance:

$$\text{var}\{z\} = \langle z^2 \rangle = \int_0^{\infty} \mathcal{S}(k) dk.$$

(The equations above are written in terms of spatial angular frequency k , as used for surface generation, but the reasoning and functional form of Eq. (B.3) are the same for any other spatial or temporal frequency variable.)

We now have two formulations of the variance at frequency k_n : $\frac{1}{2} A_n^2$ and $|\hat{z}(n)|^2$. It is wise to check that these are equivalent. In the numerical example of Fig. 2.1, sinusoids were generated using

$$A(n) = 0.1 \exp(-3n/N).$$

The resulting surface was Fourier transformed to obtain the amplitudes $\hat{z}(n)$, which are plotted in Fig. 2.1. Table B.1 shows the variance for the first few frequencies for the numerical simulation of Fig. 2.1. It is clear that the values of $\frac{1}{2} A_n^2$ equal the values of the one-sided variance spectrum obtained from the Fourier transform of the surface elevations $z(r)$.

We can now repeat the above process for two dimensions, starting with

$$z(x, y) = \sum_{n=0}^{\infty} \sum_{m=0}^{\infty} A_{n,m} \cos(k_n x + k_m y + \phi_n),$$

and ending up with a function

$$\Psi(k_x, k_y) \equiv \lim_{\Delta k_n \rightarrow 0} \lim_{\Delta k_m \rightarrow 0} \frac{\frac{1}{2} A_{n,m}^2}{\Delta k_n \Delta k_m}. \quad (\text{B.4})$$

n	$A(n) = 0.1 \exp(-3n/N)$ [m]	$\frac{1}{2}A^2(n)$ [m ²]	$\mathcal{S}_{1S}(n) = \hat{z}(n) ^2$ [m ²]
0	0	0	0
1	0.0829	0.0034	0.0034
2	0.0678	0.0034	0.0024
3	0.0570	0.0016	0.0016
\vdots	\vdots	\vdots	\vdots

Table B.1: Comparison of variances computed two different ways. The values of $|\hat{z}(n)|^2$ are taken from the lower-right panel of Fig. 2.1.

Here the x and y subscripts on k remind us that there are independent spatial frequencies associated with the x and y directions. $\Psi(k_x, k_y)$ is the *directional variance spectrum in Cartesian coordinates*. Its units are m²/(rad/m)². This spectrum is often called the “two-dimensional wavenumber spectrum,” and its arguments are often written in vector form, $\Psi(\mathbf{k})$, where $\mathbf{k} = (k_x, k_y)$ denotes the location of the frequency point in the 2-D frequency plane. This spectrum depends on direction, i.e., on the direction of the (k_x, k_y) point in a two-dimensional frequency plane. Usually, the $+x$ direction is chosen to be pointing downwind. In this case, the angle $\varphi = \tan^{-1}(k_y/k_x)$ gives the direction relative to the wind direction. At Eq. (B.4) shows, the integral of $\Psi(k_x, k_y)$ over all frequencies gives the variance of the two-dimensional surface:

$$\text{var}\{z\} = \langle z^2 \rangle = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \Psi(k_x, k_y) dk_x dk_y .$$

It is also common to define a directional spectrum in terms of polar coordinates given by the magnitude k and direction φ of the vector \mathbf{k} . These are related to k_x, k_y by

$$k = \sqrt{k_x^2 + k_y^2}$$

$$\varphi = \tan^{-1} \left(\frac{k_y}{k_x} \right)$$

and inversely by

$$k_x = k \cos \varphi$$

$$k_y = k \sin \varphi .$$

In this case we define

$$\tilde{\Psi}(k, \varphi) \equiv \lim_{\Delta k \rightarrow 0} \lim_{\Delta \varphi \rightarrow 0} \frac{\frac{1}{2}A_{n,m}^2}{\Delta k \Delta \varphi} . \quad (\text{B.5})$$

This spectrum has units of m²/[(rad/m) rad]. (The tilde notation is used here to distinguish this spectrum from the $\Psi(k, \varphi)$ spectrum of Elfouhaily, which is defined below. Some authors reserve the name “directional spectrum” for $\tilde{\Psi}(k, \varphi)$ and refer to $\Psi(k_x, k_y)$ as the wavenumber spectrum.) As before, definition (B.5) shows that integrating $\tilde{\Psi}(k, \varphi)$ over k and φ gives the variance:

$$\text{var}\{z\} = \int_0^{\infty} \int_0^{2\pi} \tilde{\Psi}(k, \varphi) dk d\varphi .$$

The directional spectrum given in §B.3 is specified in terms of polar coordinates k, φ . However, we need a spectrum in terms of Cartesian coordinates k_x, k_y for use in a rectangular DFT grid. The change of variables from polar to Cartesian coordinates is effected by the Jacobian

$$\begin{aligned}\Psi(k_x, k_y) &= \tilde{\Psi}(k, \varphi) \left| \frac{\partial(k, \varphi)}{\partial(k_x, k_y)} \right| \\ &= \tilde{\Psi}(k, \varphi) \left| \begin{array}{cc} \frac{\partial k}{\partial k_x} & \frac{\partial k}{\partial k_y} \\ \frac{\partial \varphi}{\partial k_x} & \frac{\partial \varphi}{\partial k_y} \end{array} \right| \\ &= \tilde{\Psi}(k, \varphi) \frac{1}{k}.\end{aligned}$$

Note that the $1/k$ factor converts the units of $\tilde{\Psi}(k, \varphi)$ into the units of $\Psi(k_x, k_y)$.

In Eq. (B.15) of §B.3 this last equation is partitioned as

$$\Psi(k_x, k_y) = \frac{1}{k} \mathcal{S}(k) \Phi(k, \varphi) \equiv \Psi(k, \varphi), \quad [\text{ECKV 45}] \quad (\text{B.6})$$

where $\mathcal{S}(k)$ is an omnidirectional spectrum and $\Phi(k, \varphi)$ is a nondimensional *spreading function*, which shows how waves of different frequencies propagate (or “spread out”) relative to the downwind direction at $\varphi = 0$. Labels such as [ECKV 45] refer to the corresponding equations in Elfouhaily et al. (1997). The spreading function by definition satisfies

$$\int_0^{2\pi} \Phi(k, \varphi) d\varphi = 1 \quad (\text{B.7})$$

for all k .

Equation (B.6) shows that to obtain the ECKV variance spectrum in Cartesian coordinates we need only evaluate the ECKV $\Psi(k, \varphi)$ spectrum for the corresponding values of k and φ , i.e.

$$\Psi(k_x, k_y) = \Psi\left(k = \sqrt{k_x^2 + k_y^2}, \varphi = \tan^{-1}(k_y/k_x)\right). \quad (\text{B.8})$$

Note in particular that there is no “extra” k factor involved in the conversion of $\Psi(k, \varphi)$ to $\Psi(k_x, k_y)$; both quantities have the same units.

Integration of Eq. (B.6) over the respective (k_x, k_y) and (k, φ) frequency planes gives the variance as

$$\begin{aligned}\langle z^2 \rangle &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \Psi(k_x, k_y) dk_x dk_y \\ &= \int_0^{\infty} \int_0^{2\pi} \frac{1}{k} \mathcal{S}(k) \Phi(k, \varphi) k dk d\varphi \\ &= \int_0^{\infty} \mathcal{S}(k) dk, \quad [\text{ECKV A2}] \quad (\text{B.9})\end{aligned}$$

after noting the normalization of Eq. (B.7). Thus the variance of the sea surface is still contained in the omnidirectional spectrum, even in the two-dimensional case. The omnidirectional spectrum $\mathcal{S}(k)$ is obtained from $\Psi(k, \varphi)$ via

$$\mathcal{S}(k) = \int_{-\pi}^{\pi} \Psi(k, \varphi) k d\varphi. \quad [\text{ECKV A3}]$$

Now return to Eq. (B.1) and take the derivative to get the slope of the sea surface for the n^{th} wave:

$$\frac{dz_n(x)}{dx} = -A_n k_n \sin(k_n x + \phi_n)$$

As in Eq. (2.1), the variance of this slope is

$$\text{var} \left\{ \frac{dz_n}{dx} \right\} \equiv \frac{1}{\Lambda_n} \int_0^{\Lambda_n} [A_n k_n \sin(k_n x + \phi_n)]^2 dx = \frac{1}{2} A_n^2 k_n^2.$$

A limit operation corresponding to Eq. (B.3) gives

$$\lim_{\Delta k_n \rightarrow 0} \frac{\frac{1}{2} A_n^2 k_n^2}{\Delta k_n} = k^2 \mathcal{S}(k). \quad (\text{B.10})$$

The quantity $k^2 \mathcal{S}(k)$ is the *omnidirectional slope variance spectrum*, usually called just the slope spectrum. The units of $k^2 \mathcal{S}(k)$ are m rad. Integrating the slope spectrum over all frequencies gives the total variance σ^2 of the sea surface slope:

$$\sigma^2 \equiv \text{var} \left\{ \frac{dz}{dx} \right\} = \left\langle \left(\frac{dz}{dx} \right)^2 \right\rangle = \int_0^\infty k^2 \mathcal{S}(k) dk.$$

This variance is usually called the *mean square slope* or mss. The units of mss are rad². This is a nondimensional number, but the label of rad² reminds us that we can think of the slope as an angle from the horizontal measured in radians.

The corresponding relations for two dimensions are derived in the same fashion and lead to similar results. Assuming that the wind is blowing in the $+x$ direction, the mean-square slope in the along-wind direction is given by either of

$$\begin{aligned} \left\langle \left(\frac{\partial z(x, y)}{\partial x} \right)^2 \right\rangle \equiv \sigma_x^2 \equiv \text{mss}_x &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} k_x^2 \Psi(k_x, k_y) dk_x dk_y \\ &= \int_{-\infty}^{\infty} \int_{-\pi}^{\pi} k^2 \cos^2 \varphi \Psi(k, \varphi) k dk d\varphi. \end{aligned} \quad [\text{ECKV A4}]$$

The corresponding equation for the cross-wind direction is

$$\begin{aligned} \left\langle \left(\frac{\partial z(x, y)}{\partial y} \right)^2 \right\rangle \equiv \sigma_y^2 \equiv \text{mss}_y &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} k_y^2 \Psi(k_x, k_y) dk_x dk_y \\ &= \int_{-\infty}^{\infty} \int_{-\pi}^{\pi} k^2 \sin^2 \varphi \Psi(k, \varphi) k dk d\varphi. \end{aligned} \quad [\text{ECKV A5}]$$

Recalling that variances of random variables add to get the total variance due to all variables gives the total mean square slope

$$\begin{aligned} \text{mss} = \text{mss}_x + \text{mss}_y &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} (k_x^2 + k_y^2) \Psi(k_x, k_y) dk_x dk_y \\ &= \int_{-\infty}^{\infty} \int_{-\pi}^{\pi} k^2 \Psi(k, \varphi) k dk d\varphi \\ &= \int_{-\infty}^{\infty} k^2 \mathcal{S}(k) dk. \end{aligned} \quad [\text{ECKV A6}]$$

Thus, even in the 2-D case, the total slope variance can be obtained from the 1-D omnidirectional slope spectrum.

Table B.2 summarizes the spectral quantities used in this tutorial.

Spectrum Name	Symbols	Units
1-D or omnidirectional		
variance or elevation	$\mathcal{S}(k)$	$\text{m}^2/(\text{rad}/\text{m})$
slope	$k^2 \mathcal{S}(k)$	m rad
2-D or directional		
variance or elevation	$\Psi(k_x, k_y), \Psi(k, \varphi)$	$\text{m}^2/(\text{rad}/\text{m})^2$
alongwind slope	$k_x^2 \Psi(k_x, k_y), k^2 \cos^2 \varphi \Psi(k, \varphi)$	rad^2
crosswind slope	$k_y^2 \Psi(k_x, k_y), k^2 \sin^2 \varphi \Psi(k, \varphi)$	rad^2
total slope	$(k_x^2 + k_y^2) \Psi(k_x, k_y), k^2 \Psi(k, \varphi)$	rad^2

Table B.2: Summary of spectral quantities.

B.2 The Pierson-Moskowitz Omnidirectional Gravity Wave Spectrum

The Pierson-Moskowitz spectrum (Pierson and Moskowitz, 1964) describes gravity waves in a “fully developed” sea. A fully developed sea is an idealization of the statistically steady-state wave field resulting from a steady wind blowing for an infinitely long time over an infinite fetch. (In practice, the duration and fetch required to achieve something close to a fully developed sea depend on the wind speed. A steady wind of 5 m s^{-1} blowing for 10 hours over a fetch of 60 km might be adequate; for hurricane winds of 35 m s^{-1} , a fetch of a few thousand kilometers and a duration of several days would be required. Thus it is much easier to approach a fully developed sea at low wind speeds than at high.)

The one-sided, omnidirectional Pierson-Moskowitz spectrum, formulated in terms of angular spatial frequency k , is

$$\mathcal{S}_{\text{PM}}(k) = \frac{\alpha}{2k^3} \exp \left[-\beta \left(\frac{g}{k} \right)^2 \frac{1}{U_{19}^4} \right] \quad [\text{m}^2/(\text{rad}/\text{m})], \quad (\text{B.11})$$

where

$$\alpha = 0.0081,$$

$$\beta = 0.74,$$

$g = 9.82 \text{ m s}^{-2}$ is the acceleration of gravity,

U_{19} is the wind speed in m s^{-1} at 19.5 m above the sea surface, and

k is the angular spatial frequency in rad m^{-1} .

The wind speed at 19.5 m can be obtained from the more commonly used wind at 10 m above the sea surface by the approximate formula

$$U_{19} \approx 1.026 U_{10}.$$

As has already been noted, it is often of interest to express a variance spectrum in terms of other variables, such as the wavenumber ν or the temporal angular frequency ω . To change variables

in a spectral density function, the key is to recall that a variance density function by definition expresses the variance per unit frequency interval. The variance contained in some interval dk of the spatial angular frequency equals the variance contained in the corresponding interval $d\nu$ of the wavenumber or the interval $d\omega$ of the temporal frequency. Thus we have

$$\mathcal{S}_{\text{PM}}(k)dk = \mathcal{S}_{\text{PM}}(\nu)d\nu = \mathcal{S}_{\text{PM}}(\omega)d\omega.$$

To change the variable from $k = 2\pi\nu$ to ν , the previous equation gives

$$\mathcal{S}_{\text{PM}}(\nu) = \mathcal{S}_{\text{PM}}(k)\frac{dk}{d\nu} = \mathcal{S}_{\text{PM}}(k = 2\pi\nu)2\pi,$$

which results in

$$\mathcal{S}_{\text{PM}}(\nu) = \frac{\alpha}{8\pi^2\nu^3} \exp\left[-\beta\left(\frac{g}{2\pi\nu}\right)^2 \frac{1}{U_{19}^4}\right] \quad [\text{m}^2/(1/\text{m})]. \quad (\text{B.12})$$

To change variables from spatial angular frequency k to temporal angular frequency ω , we use the dispersion relation for gravity waves in deep water,

$$\omega^2 = gk,$$

to evaluate

$$\frac{dk}{d\omega} = \frac{2\omega}{g},$$

which leads to

$$\mathcal{S}_{\text{PM}}(\omega) = \frac{\alpha g^2}{\omega^5} \exp\left[-\beta\left(\frac{g}{\omega U_{19}}\right)^4\right] \quad [\text{m}^2/(\text{rad}/\text{s})]. \quad (\text{B.13})$$

All of these formulas have units of meters squared over the appropriate frequency. (The quantities $dk/d\nu$ and $dk/d\omega$ seen in the conversions are the Jacobians for the one-dimensional changes of variables.) Figure B.1 shows the Pierson-Moskowitz spectrum as functions of k and ω for wind speeds of $U_{10} = 5, 10$, and 15 m s^{-1} .

This spectrum has withstood the test of time quite well (e.g., Alves and Banner, 2003) as a description of gravity waves in fully developed seas. However, it should not be used for high-frequency (short-wavelength) gravity waves, and certainly not for capillary waves. Likewise, it does not describe young seas, which have not had the time or fetch needed to approach the state of a well developed sea.

Figure B.2 shows the Pierson-Moskowitz slope spectra for three wind speeds. Note that the slope spectrum falls off much more slowly for high frequencies than does the elevation spectrum. That means that the higher frequencies contribute much more to the total slope variance than they do to the total elevation variance.

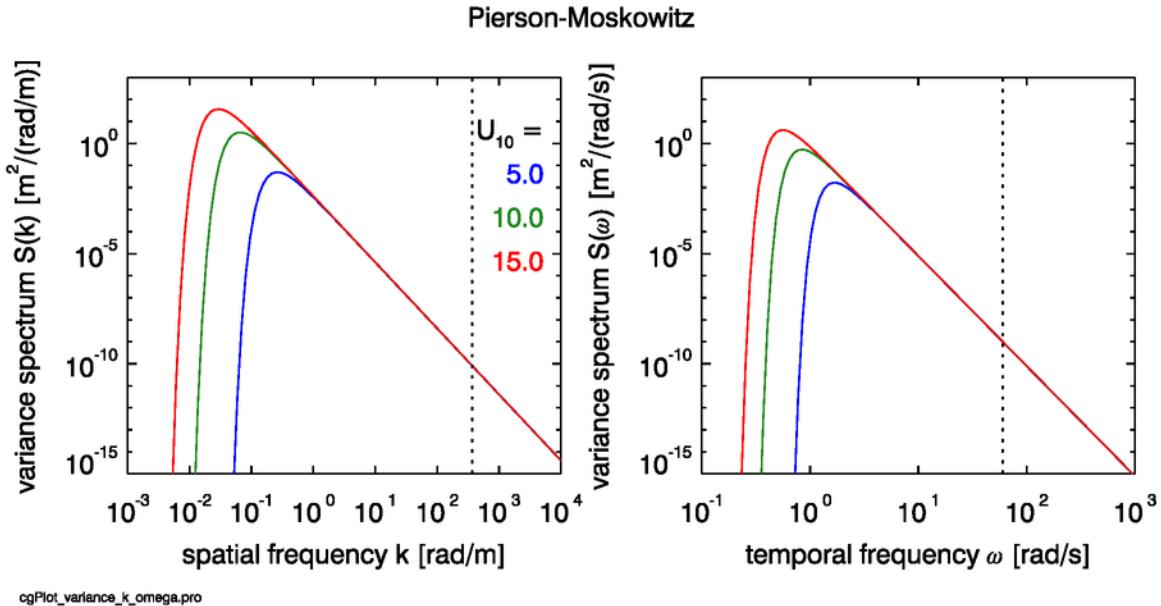


Figure B.1: The Pierson-Moskowitz variance spectrum as functions of k and ω for wind speeds of $U_{10} = 5, 10$, and 15 m s^{-1} . The vertical dotted lines at $k = 370 \text{ rad/m}$ and $\omega = 60.3 \text{ rad/s}$ show the boundary between gravity and capillary waves.

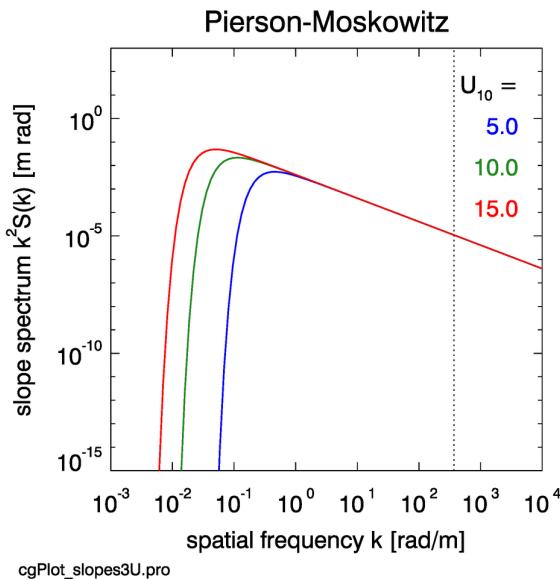


Figure B.2: The Pierson-Moskowitz slope spectrum for wind speeds of $U_{10} = 5, 10, 15 \text{ m s}^{-1}$. The plot uses the same ordinate scale as used for the elevation spectrum in the left panel of Fig. B.1 in order to highlight the slow falloff of the slope spectrum compared to the elevation spectrum. The vertical dotted line is the boundary between gravity and capillary waves.

B.3 The Elfouhaily et al. Directional Gravity-Capillary Wave Spectrum

In order to model two-dimensional sea surfaces $z(x, y)$, we need a wave variance spectrum that describes the distribution of wave variance for waves propagating in different directions.

As the examples in §3.1 showed, if we have measurements of the two-dimensional sea surface elevation at a given time, $z(\mathbf{x}_{rs}) = z(x_r, y_s) = z(r, s)$, then the two-dimensional discrete Fourier transform of Eq. (1.17) gives the two-dimensional amplitudes

$$\hat{z}(\mathbf{k}_{uv}) = \hat{z}[k_x(u), k_y(v)] = \hat{z}(u, v) = \mathfrak{F}\{z(r, s)\} = \mathfrak{F}\{z(\mathbf{x}_{rs})\}.$$

Dividing by the discrete frequency bandwidths gives an estimate (a 2-D periodogram) of the two-dimensional variance spectral density

$$P(\mathbf{k}_{uv}) \equiv \frac{|\hat{z}(\mathbf{k}_{uv})|^2}{\Delta k_u \Delta k_v} = \frac{\Psi(\mathbf{k}_{uv})}{\Delta k_u \Delta k_v}, \quad (\text{B.14})$$

just as in the 1-D case. Here I have written $z(\mathbf{x}_{rs})$, $\hat{z}(\mathbf{k}_{uv})$, and $\Psi(\mathbf{k}_{uv})$ to emphasize that these are discrete functions, whereas $\Psi(\mathbf{k})$ denotes a spectral density function of the continuous variable \mathbf{k} . Just as in the 1-D case, the 2-D periodogram is an approximation of the 2-D variance spectral density, $P(\mathbf{k}_{uv}) \doteq \Psi(\mathbf{k})$. As Eq. (1.17) shows, $\Psi(\mathbf{k}_{uv})$ has units of m^2 , whereas $P(\mathbf{k}_{uv})$ and $\Psi(\mathbf{k})$ have units of $\text{m}^2/(\text{rad}/\text{m})^2$.

Unfortunately, making measurements of 2-D sea surfaces is extremely difficult. There are very few such data sets obtained, for example, by laser reflectance measurements (e.g., Huang et al., 2000), and these do not cover the full range of spatial scales. Given the paucity of empirical 2-D wave data from which to develop 2-D variance spectra, the common procedure is to start with a 1-D or *omnidirectional* spectrum and add an angular *spreading function* to distribute the wave energy over different directions relative to the downwind direction. In nature, most waves travel more or less downwind, a small amount of energy is contained in waves propagating in nearly cross-wind directions, and almost no energy is contained in waves that by some chance (such as the breaking of a larger wave generating smaller waves in all directions) might be propagating in upwind directions. The spreading function must capture this behavior. Although omnidirectional wave spectra are well grounded in observations, the choice of a spreading function is something of a black art.

Elfouhaily et al. (1997) presented an omnidirectional variance spectrum that explicitly includes both the gravity and capillary wave scales. For brevity, I will often denote this paper and their model by “ECKV,” taken from the initials of the authors’ last names. The boundary between gravity and capillary waves is taken to be $k = \sqrt{\rho g/\tau} = 370 \text{ rad/m}$, the angular spatial frequency at which the restoring forces (which tend to bring a wave surface back to an unperturbed level surface) of gravity and surface tension are equal. The corresponding wavelength is $\Lambda = 2\pi/370 = 0.017 \text{ m}$. They then combine their omnidirectional spectrum with a spreading function to obtain their one-sided, directional variance spectrum. Using their notation, the one-sided, 2-D ECKV spectrum has the form

$$\Psi(k, \varphi) = \frac{1}{k} \mathcal{S}(k) \Phi(k, \varphi) \quad [\text{ECKV 45}]. \quad (\text{B.15})$$

Here $\mathcal{S}(k)$ is the 1-D omnidirectional spectrum with units of $\text{m}^2/(\text{rad}/\text{m})$, and $\Phi(k, \varphi)$ is a non-dimensional spreading function. $\Psi(k, \varphi)$ thus has units of $\text{m}^2/(\text{rad}/\text{m})^2$. Equation labels such as [ECKV 45] give for reference the corresponding equation in the ECKV paper.

The ECKV omnidirectional spectrum is

$$\mathcal{S}(k) = \frac{B_l + B_h}{k^3} \quad [\text{ECKV 30}], \quad (\text{B.16})$$

where B_l is the low-frequency (long gravity wave) contribution to the variance, and B_h is the high-frequency (short gravity wave to capillary wave) contribution. (The quantity $k^3 \mathcal{S}(k)$ is called the *curvature* or *saturation* spectrum and is of interest in physical oceanography because it is related to the rate of variance dissipation of the waves. Thus ECKV refer to B_l and B_h as the low and high frequency curvature spectra.) The components of the omnidirectional spectrum are given by

$$L_{PM} = \exp[-1.25(k_p/k)^2] \quad [\text{ECKV 2}]$$

$$\Gamma = \exp\left\{-\frac{1}{2\sigma^2}[(\sqrt{k/k_p} - 1)^2]\right\} \quad [\text{below ECKV 3}]$$

$$J_p = \gamma^\Gamma \quad [\text{ECKV 3}]$$

$$F_p = L_{PM} J_p \exp\{-0.3162\Omega_c(\sqrt{k/k_p} - 1)\} \quad [\text{ECKV 32}]$$

$$F_m = L_{PM} J_p \exp[-0.25(k/k_m - 1)^2] \quad [\text{ECKV 41}]$$

(Note : A typo in ECKV Eq. 41 omitted the $L_{PM} J_p$ factor in F_m)

$$B_l = 0.5\alpha_p(c_p/c)F_p \quad [\text{ECKV 31}]$$

$$B_h = 0.5\alpha_m(c_m/c)F_m \quad [\text{ECKV 40}]$$

where

$$\alpha = 0.0081,$$

$$\beta = 1.25,$$

$g = 9.82 \text{ m s}^{-2}$ is the acceleration of gravity,

U_{10} is the wind speed in m s^{-1} at 10 m above the sea surface

k is the angular spatial frequency in rad m^{-1}

Ω_c is defines the age of the waves for the given wind speed:

= 0.84 for a fully developed sea (corresponds to Pierson-Moskowitz)

= 1 for a "mature" sea [used in ECKV Fig 8a]

= 2 to 5 for a "young" sea; the maximum allowed value is 5

$Cd_{10N} = 0.00144$ is a drag coefficient [value deduced from ECKV Fig 11]

$u^* = \sqrt{Cd_{10N}}U_{10}$ is the friction velocity [using ECKV 61[

$a_o = 0.1733$ (ECKV 59)

$a_p = 4.0$

$k_m = 370.0 \text{ rad/m}$

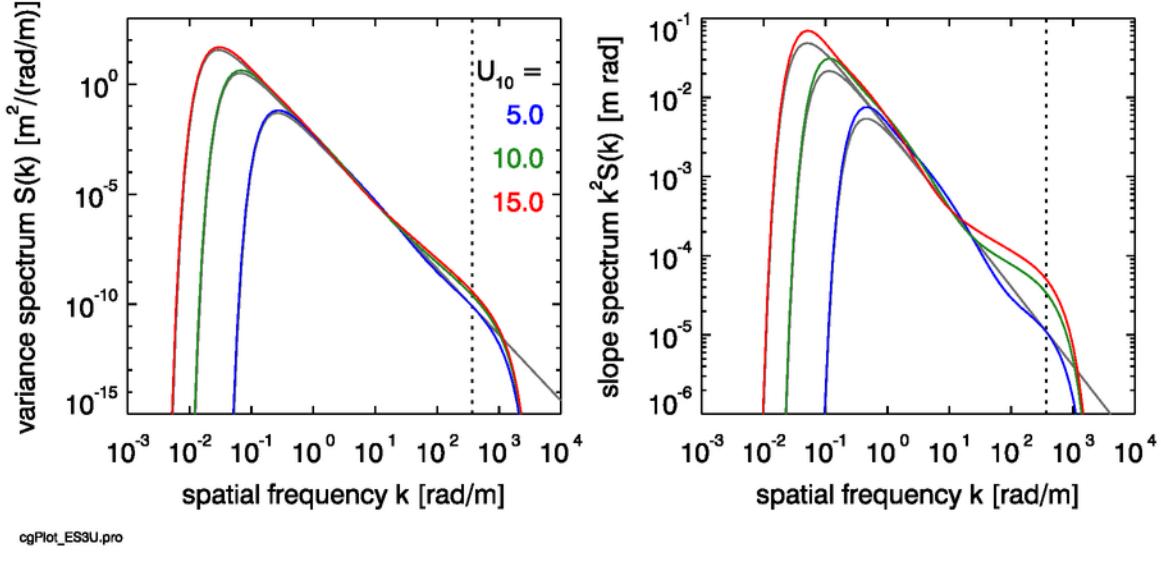


Figure B.3: The omnidirectional part \mathcal{S} of the Elfouhaily et al. (1997) elevation variance spectrum (left panel) and slope spectrum $k^2\mathcal{S}$ (right panel) for fully developed seas and wind speeds of $U_{10} = 5, 10, 15 \text{ m s}^{-1}$. The gray lines show the corresponding Pierson-Moskowitz spectra from Fig. B.2.

$c_m = 0.23 \text{ m/s}$ is the phase speed of the wave with spatial frequency k_m

$a_m = 0.13u^*/c_m$ [ECKV 59]

$\gamma = 1.7$ if $\Omega_c \leq 1$ else $\gamma = 1.7 + 6 \log_{10}(\Omega_c)$

$\sigma = 0.08(1 + 4\Omega_c^{-3})$

$\alpha_p = 0.006\Omega_c^{0.55}$ [ECKV Eq. 34]

$\alpha_m = 0.01[1 + \ln(u^*/c_m)]$ if $u^* \leq c_m$ else $\alpha_m = 0.01[1 + 3 \ln(u^*/c_m)]$ [ECKV 44]

$k_o = g/U_{10}^2$

$k_p = k_o\Omega_c^2$ is the spatial frequency of the maximum of the spectrum

$c_p = \sqrt{g/k_p}$ is the phase speed of the wave with spatial frequency k_p

$c = \sqrt{(g/k)(1 + (k/km)^2)}$ is the phase speed of the wave

At the lower frequencies, the ECKV spectrum is essentially the Pierson-Moskowitz spectrum (the L_{PM} term above) with an enhancement (the J_p term) that adds more energy to the lower frequencies. The highest frequencies have a cutoff due to viscosity damping out the smallest capillary waves. The ECKV omnidirection elevation and slope spectra are illustrated in Fig. B.3 for the case of a fully developed sea and three wind speeds. Figure B.4 shows the spectra as a function of wave age for a wind speed of $U_{10} = 10 \text{ m s}^{-1}$.

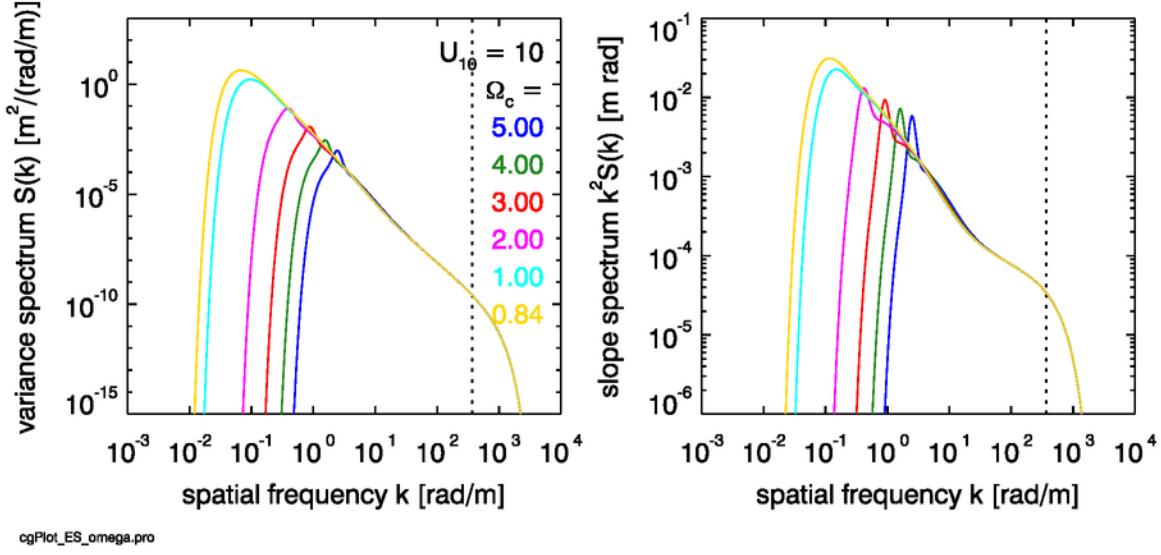


Figure B.4: The omnidirectional part of the Elfouhaily et al. (1997) elevation spectrum (left panel) and slope spectrum (right panel) for a wind speed of $U_{10} = 10$ m s⁻¹ and wave ages from very young ($\Omega_c = 5$) to mature ($\Omega_c = 1$) to fully developed ($\Omega_c = 0.84$). Compare with Fig. B.3.

B.3.1 Spreading Functions

The ECKV spreading function is given by

$$\begin{aligned}\Phi(k, \varphi) &= \frac{1}{2\pi} [1 + \Delta(k) \cos(2\varphi)] \\ &= \frac{1}{2\pi} \{1 + \tanh [a_o + a_p(c/c_p)^{2.5} + a_m(c_m/c)^{2.5}] \cos(2\varphi)\}\end{aligned}\quad (\text{B.17})$$

Note that this function is symmetric about $\varphi = \pi/2$; i.e., the function has symmetric spreading downwind and upwind. This is consistent with a symmetric variance spectrum $\Psi(-\mathbf{k}) = \Psi(\mathbf{k})$ as would be obtained from a the Fourier transform of a snapshot of a sea surface.

A commonly used family of alternate spreading functions is given by the “cosine-2S” functions (Longuet-Higgins et al., 1963), which have the form

$$\Phi(k, \varphi) = C_S \cos^{2S}(\varphi/2), \quad (\text{B.18})$$

where the normalizing coefficient is

$$C_S = \frac{1}{2\sqrt{\pi}} \frac{\Gamma(S+1)}{\Gamma(S+1/2)},$$

and S is a spreading parameter that in general depends on k , wind speed, and wave age. In this equation Γ is the customary gamma function defined by $\Gamma(p) \equiv \int_0^\infty x^{p-1} e^{-x} dx$ where $p > 0$. The cosine-2S functions are asymmetric, with much stronger downwind than upwind propagation.

The ECKV and cosine-2S spreading functions are illustrated in Fig. B.5. Both of these functions satisfy the normalization condition (B.7). Both spreading functions transition from strongly forward

peaked at low spatial frequencies (long gravity waves; the red curves) to curves with significant propagation at right angles to the wind at high frequencies (capillary waves; the blue curves). The cosine-2S curves are asymmetric in $\pm k$ and have at least a small amount of upwind propagation at all frequencies (except at exactly upwind, $\varphi = \pi$). Not surprisingly, the real ocean is more complicated than either of these models. In particular, observations of long-wave gravity waves tend to show a bimodal spreading about the downwind direction, which transitions to a more isotropic, unimodal spreading at shorter wavelengths (e.g. Heron et al., 2006). However, the simple models of Eqs. (B.17) and (B.18) are adequate for the present purpose of illustrating surface-generation techniques.

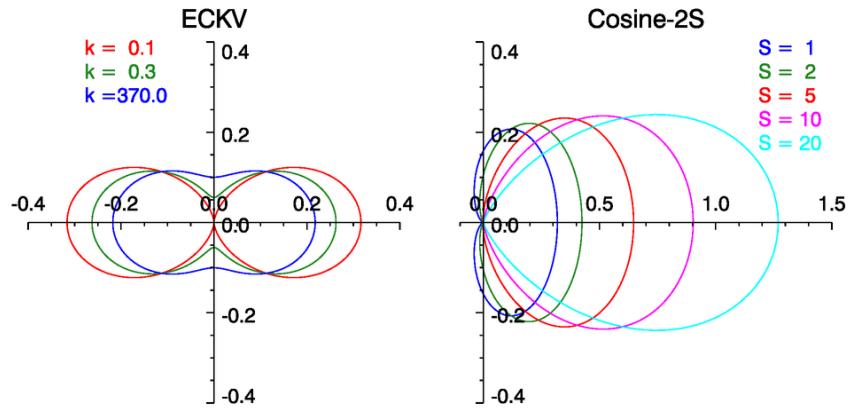


Figure B.5: Example spreading functions according to the ECKV model (left) and the cosine-2S model (right) for a wind speed of 10 m s^{-1} . Downwind is to the right, upwind is to the left.

References

- J. H. G. M. Alves and M. L. Banner. Revisiting the Pierson-Moskowitz asymptotic limits for fully developed wind waves. *J. Phys. Oceanogr.*, 33:1301–1323, 2003.
- R. R. Bracewell. *The Fourier Transform and Its Applications, Second Edition, Revised*. McGraw-Hill, 1986.
- J. W. Cooley and J. W. Tukey. An algorithm for the machine calculation of complex Fourier series. *Math. Comput.*, 19:297–301, 1965.
- J. G. de Boer. On the correlation function in time and space of wind-generated ocean waves. Technical report, SACLANT ASW Research Centre, Tech. Rept. No. 160, 1968. URL <http://oai.dtic.mil/oai/oai?verb=getRecord&metadataPrefix=html&identifier=AD0865249>.
- J. Dongarra and F. Sullivan. Guest editors introduction to the top 10 algorithms. *Computing in Science and Engineering*, 2:22–23, 2000. doi: 10.1109/MCISE.2000.814652.
- T. Elfouhaily, B. Chapron, K. Katsaros, and D. Vandemark. A unified directional spectrum for long and short wind-driven waves. *J. Geophys. Res.*, 102:15781–15796, 1997.
- J. W. Goodman. *Introduction to Fourier Optics, Second Edition*. McGraw-Hill, 1996.
- M. L. Heron, W. J. Skirving, and K. J. Michael. Short-wave ocean wave slope models for use in remote sensing data analysis. *IEEE Trans. Geosci. Rem. Sens.*, 44:1962–1973, 2006.
- L. H. Holthuijsen. *Waves in Oceanic and Coastal Waters*. Cambridge University Press, 2007.
- K. V. Horoshenkov, A. Nichols, S. J. Tait, and G. A. Maximov. The pattern of surface waves in a shallow free surface flow. *J. Geophys. Res. Earth Surf.*, 118:1864–1876, 2013.
- P. A. Huang, D. W. Wang, E. J. Walsh, W. B. Krabill, and R. N. Swift. Airborne measurements of the wavenumber spectra of ocean surface waves. Part II: Directional distribution. *J. Phys. Oceanogr.*, 30:2768–2787, 2000.
- S. Kay, J. Hedley, S. Lavender, and A. Nimmo-Smith. Light transfer at the ocean surface modeled using high resolution sea surface realizations. *Optics Express*, 19:6493–6504, 2011.
- G. E. Latta and J. A. Bailie. On the autocorrelation functions of wind generated ocean waves. *Zeit. Angew. Math. Phys.*, 19:575–586, 1968.

- M. S. Longuet-Higgins, D. E. Cartwright, and N. D. Smith. Observations of the directional spectrum of sea waves using the motions of a flotation buoy. In *Ocean Wave Spectra*, pages 111–136. Prentice Hall, 1963.
- J. Makhoul. A fast cosine transform in one and two dimensions. *IEEE Trans. Acoustics, Speech, Signal Process.*, 28(1):27–34, 1980.
- C. D. Mobley. *HydroPol Mathematical Documentation: Invariant Imbedding Theory for the Vector Radiative Transfer Equation*. Technical report, Sequoia Scientific, Inc., Bellevue, WA 98075, 2014. URL <http://www.oceanopticsbook.info/view/references/publications>.
- C. D. Mobley. Polarized reflectance and transmittance properties of windblown sea surfaces. *Appl. Optics*, 54(15):4828–4849, 2015.
- C. D. Mobley. *Light and Water: Radiative Transfer in Natural Waters*. Academic Press, 1994. URL www.oceanopticsbook.info/view/introduction/level_2/text_books_relevant_to_ocean_optics.
- W. J. Pierson and L. Moskowitz. A proposed spectral form for fully developed wind seas based on the similarity theory of S. A. Kitaigorodskii. *J. Geophys. Res.*, 69:5181–5190, 1964.
- R. W. Preisendorfer. *Hydrologic Optics*. U.S. Department of Commerce, Pacific Marine Environmental Laboratory, 1976. URL www.oceanopticsbook.info/view/introduction/level_2/text_books_relevant_to_ocean_optics. In 6 volumes; Volume VI treats surfaces.
- W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flanney. *Numerical Recipes in Fortran: The Art of Scientific Computing, Second Edition*. Cambridge Univ. Press, 1992.
- J. G. Proakis and D. G. Manolakis. *Digital Signal Processing: Principles, Algorithms, and Applications, Third Edition*. Prentice Hall, 1996.
- A. Schuster. On the investigation of hidden periodicities with application to a supposed 26 day period of meteorological phenomena. *Terrestrial Magnetism*, III:13–41, 1898.
- G. Strang. Wavelets. *Amer. Scientist*, 82:250–255, 1994.
- J. Tessendorf. *Simulating Ocean Water*. Technical report, SIGGRAPH Course Notes, 2004. URL <http://people.clemson.edu/~jtessen/reports.html>.