LECTURE 17
TUESDAY, FEBRUARY 20, 2007

2-D FOURIER SERIES

In our study of trend surfaces we used polynomials as our basis functions to assemble the appropriate surface. This choice is of course completely arbitrary as very few data sets are expected to behave like polynomials. Because of our success in approximating 1-D time-series with sums of cosines and sines we may suspect that a 2-D extension of the Fourier Analysis will be useful in the study of 2-dimensional data. The simplest building block in the 2-D Fourier series is the product of two cosines, each a function of a separate space variable:

\[ z = A_x \cos (k_x x - \phi_x) A_y \cos (k_y y - \phi_y) \]  \hspace{1cm} (17.1)

i.e., \( A_x (k_x x - \phi_x) \) represent an oscillating carpet or cylindrical undulations aligned with the y-axis. Indeed, if \( k_y = 0 \) and \( A_y = 1 \) then that is all we have. However, for nonzero \( k_y \) the amplitude of the carpet also varies with \( y \). As we did for the 1-D Fourier series, we will replace the phase with components with zero phase:

\[
z = A_x [\cos \phi_x \cos k_x x + \sin \phi_x \sin k_x x] A_y [\cos \phi_y \cos k_y y + \sin \phi_y \sin k_y y]
\]

\[
= \alpha_x \alpha_y \cos k_x x \cos k_y y + \alpha_x \beta_y \cos k_x x \sin k_y y + \beta_x \alpha_y \sin k_x x \cos k_y y + \beta_x \beta_y \sin k_x x \sin k_y y
\]  \hspace{1cm} (17.2)

In the general case we will have a finite number of \( k_x \) and \( k_y \) wave numbers to work with (i.e., the harmonics from 0 and up to the Nyquist wave number in each direction). Thus, the 2-D Fourier series becomes

\[
z(x, y) = \sum_{i=0}^{n} \sum_{j=0}^{m} \left\{ \alpha_i \alpha_j \cos k_x x \cos k_y y + \alpha_i \beta_j \cos k_x x \sin k_y y + \beta_i \alpha_j \sin k_x x \cos k_y y + \beta_i \beta_j \sin k_x x \sin k_y y \right\}
\]  \hspace{1cm} (17.3)

For simplicity we will only consider the case where \( n \) and \( m \) are even. We rewrite this series expansion as

\[
z(x, y) = \sum_{i=0}^{n} \sum_{j=0}^{m} \left\{ a_{ij} C_i C_j^* + b_{ij} C_i S_j^* + c_{ij} S_i C_j^* + d_{ij} S_i S_j \right\}
\]  \hspace{1cm} (17.4)

where
\[
\begin{align*}
C_i &= \cos k_x = \cos \left(2\pi i x X \right) \quad S_i = \sin k_x = \sin \left(2\pi i x X \right) \quad \Rightarrow \quad k_i = 2\pi i X \\
C_j &= \cos k_y = \cos \left(2\pi j y Y \right) \quad S_j = \sin k_y = \sin \left(2\pi j y Y \right) \quad \Rightarrow \quad k_j = 2\pi j Y
\end{align*}
\]

We would like to solve for the unknown coefficients \(a_{ij}, b_{ij}, c_{ij}, d_{ij}\). Again, we will require that the Fourier Series (17.4) minimizes the misfit between the data points \(z(x_i, y_j)\) and the Fourier series. The sum of squared misfit is then

\[
E = \sum_{k=1}^{n} \sum_{l=1}^{m} \left[ \sum_{i=0}^{n/2} \sum_{j=0}^{m/2} \left( a_{ij} C_i C_j^* + b_{ij} C_i S_j^* + c_{ij} S_i C_j^* + d_{ij} S_i S_j^* \right) - z_{kl} \right]^2
\]

(17.5)

We will first find the unknowns \(a_{ij}\). Taking the partial derivative and setting it to 0 gives

\[
\frac{\partial E}{\partial a_{ij}} = 2 \sum_{k=1}^{n} \sum_{l=1}^{m} \left[ \sum_{i=0}^{n/2} \sum_{j=0}^{m/2} \left( a_{ij} C_i C_j^* + b_{ij} C_i S_j^* + c_{ij} S_i C_j^* + d_{ij} S_i S_j^* \right) - z_{kl} \right] C_i C_j^* = 0
\]

(17.6)

Reversing the order of the two double sums gives (17.7):

\[
\sum_{k=1}^{n} \sum_{l=1}^{m} z_{kl} C_i C_j^* = \sum_{i=0}^{n/2} \sum_{j=0}^{m/2} \left[ \sum_{k=1}^{n} \sum_{l=1}^{m} \left( a_{ij} C_i C_j^* + b_{ij} C_i S_j^* + c_{ij} S_i C_j^* + d_{ij} S_i S_j^* \right) - z_{kl} \right] C_i C_j^*
\]

Because of the orthogonality relationships, most of the terms in the sum vanishes. The last three terms will be sums of sines times cosines which are always zero. The first term is simply

\[
\sum_{i=0}^{n/2} \sum_{j=0}^{m/2} \left( a_{ij} C_i C_j^* + b_{ij} C_i S_j^* + c_{ij} S_i C_j^* + d_{ij} S_i S_j^* \right) = a_{ij} \frac{n}{2} \frac{m}{2}
\]

(17.8)

because the inner sum can be written

\[
a_{ij} \sum_{i=1}^{n} C_i \sum_{j=1}^{m} C_j^* C_j^*
\]

and we know the first part is non-zero \(= n/2\) only when \(i = i'\) and the second is non-zero \(= m/2\) only when \(j = j'\). Thus, \(a_{ij}^*\) is the only term that survives, scaled by \(nm/4\). Since \(i'\) and \(j'\) are just dummy subscripts, the solution for \(a_{ij}\) (and \(b, c, d\) by analogy) then becomes

\[
a_{ij} = \frac{4}{nm} \sum_{k=1}^{n} \sum_{l=1}^{m} z_{kl} C_i C_j^* \\
b_{ij} = \frac{4}{nm} \sum_{k=1}^{n} \sum_{l=1}^{m} z_{kl} C_i S_j^* \\
c_{ij} = \frac{4}{nm} \sum_{k=1}^{n} \sum_{l=1}^{m} z_{kl} S_i C_j^* \\
d_{ij} = \frac{4}{nm} \sum_{k=1}^{n} \sum_{l=1}^{m} z_{kl} S_i S_j^*
\]

(17.9)
These formulas hold if we redefine some of the (zero-frequency) coefficients associated with \( i = 0 \) and \( j = 0 \) (since \( \sum \cos 0 \cdot \cos 0 = n \), not \( n/2 \)). The power spectrum for 2-D data is simply an extension of the 1-D equation (5.109):

\[
S_{ij} = \frac{nm}{4(nm - 1)} \left( a_{ij}^2 + b_{ij}^2 + c_{ij}^2 + d_{ij}^2 \right) \approx \frac{1}{4} \left( a_{ij}^2 + b_{ij}^2 + c_{ij}^2 + d_{ij}^2 \right)
\]

(17.10)

Thus, the 2-D power spectrum can be displayed as a contour map or image. Consider the position \((i,j)\) marked in the power spectrum. Let us say that the power spectrum has the value 1 there and is zero everywhere else. What shape does the surface take on? We will see that it is an undulating cylindrical wave “propagating” in the \( \phi \) direction with wave number \( k_r \).

Let us do this in the reverse order and ask what the equations must look like. It is clear that the surface \( z(x,y) \) only depends on the distance \( d \) in the direction \( n \). Let us assume the wavelength \( \lambda \), so \( k_r = 2\pi/\lambda \). Then, given an amplitude \( A \) we find

\[
z(x,y) = A \cos (k_r d - \phi)
\]

But \( d \) is simply the projection of vector \( \mathbf{r} = (x, y) \) onto \( \mathbf{n} \), a unit vector. We write

\[
z(x,y) = A \cos \left( k_r \cdot \mathbf{n} - \phi \right) = A \cos \left( k_r \mathbf{r} \cdot \mathbf{n} - \phi \right)
\]

\[
= A \cos \left( k_r n_x x - \phi_x + k_r n_y y - \phi_y \right) = A \cos \left( k_r x - \phi_x + k_r y - \phi_y \right)
\]

Expanding this as the cosine of a sum of two angles we find

\[
z = A \cos(k_r x - \phi_x) \cos(k_r y - \phi_y) - \sin(k_r x - \phi_x) \sin(k_r y - \phi_y)
\]

which we use with basic trigonometric identities to obtain
$$z = A \left[ (\cos k_x x \cos \phi_x + \sin k_x x \sin \phi_x)(\cos k_y y \cos \phi_y + \sin k_y y \sin \phi_y) \right]$$

$$- \left( \sin k_x x \cos \phi_x - \cos k_x x \sin \phi_x \right) \left( \sin k_y y \cos \phi_y - \cos k_y y \sin \phi_y \right)$$

Let

$$C_x = \cos k_x x \; S_x = \sin k_x x \; O_x = A \cos \phi_x \; I_x = A \sin \phi_x$$

$$C_y = \cos k_y y \; S_y = \sin k_y y \; O_y = A \cos \phi_y \; I_y = A \sin \phi_y$$

Having nothing better to do, we keep at it and finally get

$$z = \left[ (C_x O_x + S_x I_x)(C_y O_y + S_y I_y) - (S_x O_x + C_x I_x)(S_y O_y + C_y I_y) \right]$$

$$= \left( O_x O_y - I_x I_y \right) C_x C_y + \left( O_x I_y + I_x O_y \right) C_x S_y + \left( I_x O_y + O_x I_y \right) S_x C_y + \left( I_x I_y - O_x O_y \right) S_x S_y$$

$$= a_{xy} C_x C_y + b_{xy} C_x S_y + c_{xy} S_x C_y + d_{xy} S_x S_y$$

Thus, the power spectrum is

$$s_{xy}^2 = \frac{1}{4} \left( a_{xy}^2 + b_{xy}^2 + c_{xy}^2 + d_{xy}^2 \right)$$

which is only non-zero for the two wave number $k_x$ and $k_y$. Therefore, we have proved that a single spike in the 2-D spectrum represents a cosine wave propagating in the $\phi = \tan^{-1}(k_y/k_x)$ direction with wave number $k_r = \sqrt{k_x^2 + k_y^2}$. This information will be useful when trying to interpret 2-D spectrums of real data.

We would like to simplify (17.3) by using complex exponentials which worked so well for the 1-D case. To do so we remember that we could express sines and cosines as

$$\cos x = \frac{1}{2} (e^{ix} + e^{-ix}) \quad \sin x = \frac{1}{2i} (e^{ix} - e^{-ix})$$

Let us use the shorthand $x = k_i x$ and $y = k_j y$ (to avoid writer’s cramp) and remember they are functions of $i$ and $j$. Substituting (17.11) into (17.3) gives

$$z(x, y) = \sum_{i=0}^{n} \sum_{j=0}^{m^2} \left\{ \frac{1}{4} \alpha \beta \left( e^{ix} + e^{-ix} \right) \left( e^{iy} + e^{-iy} \right) + \frac{1}{4i} \alpha \beta \left( e^{ix} + e^{-ix} \right) \left( e^{iy} - e^{-iy} \right) \right. \right.$$
z(x, y) = \sum_{i=-n_1}^{n_1} \sum_{j=-m_2}^{m_2} \left\{ \frac{1}{4}[\alpha, \alpha - i\alpha\beta - i\beta], \beta, \beta] e^{ix+iy} + \frac{1}{4}[\alpha, \alpha + i\alpha\beta - i\beta, \alpha + \beta, \beta] e^{-ix-iy} \right\}

In the 1-D case we found that by considering negative frequencies the $e^{-ix}$ term was unnecessary. Inspection of (17.13) reveals that if $\beta_i = -\beta_i$, $\alpha_i = \alpha_i$, $\alpha_j = \alpha_j$, and $\beta_j = -\beta_j$ we may write

$$z(x, y) = \sum_{i=-n_1}^{n_1} \sum_{j=-m_2}^{m_2} \left\{ \frac{1}{4}[\alpha, \alpha - i\alpha\beta - i\beta], \beta, \beta] e^{ik} \right\}$$

(17.14)

Because $\alpha$ is connected to the even part (cosine) and $\beta$ to the odd part (sine), we know the assumption to be true. Consider

$$b_{ij} = \alpha, \beta = \frac{4}{mn} \sum_{k=1}^{n} \sum_{l=1}^{m} z_{kl} \cos(kx) \sin(ly)$$

with $-i$ we simply get $\alpha_i \beta_j = \alpha_i \beta_i$; while $-j$ gives $\alpha_i \beta_j = -\alpha_i \beta_j$, etc. Therefore (keep in mind that $i$ in expressions is $\sqrt{-1}$ unless it appears as a subscript):

$$J_{ij} = J_i J_j = \frac{1}{2}[\alpha_i - i\beta_i, J_i = \frac{1}{2}[\alpha_i + i\beta_i, J$$

(17.15)

where

$$J_i = \frac{1}{2}[\alpha - i\beta]$$

Therefore, our final expression for the discrete 2-D Fourier series becomes

$$z(x, y) = \sum_{i=-n_2}^{n_2} \sum_{j=-m_2}^{m_2} J_{ij} e^{ikx + jly} = \sum_{i=-n_2}^{n_2} J_i e^{ikx} \sum_{j=-m_2}^{m_2} J_j e^{jly}$$

(17.16)

We solve for $J_{ij}$ by multiplying each side by a complex exponent representing a particular combination of harmonics and sum over all data points:

$$\sum_{k=1}^{n} \sum_{l=1}^{m} z_{kl} e^{-ikx} = \sum_{i=-n_2}^{n_2} \sum_{j=-m_2}^{m_2} \left[ \sum_{k=1}^{n} \sum_{l=1}^{m} J_{ij} e^{ikx} \right] e^{-ikx}$$

$$= \sum_{i=-n_2}^{n_2} \sum_{j=-m_2}^{m_2} J_{ij} \left[ \sum_{k=1}^{n} e^{ikx} \right] \left[ \sum_{l=1}^{m} e^{jly} \right]$$

The two terms in the brackets simplify according to the orthogonality identity for complex exponents of Fourier harmonics (16.8). We find

$$\sum_{k=1}^{n} \sum_{l=1}^{m} z_{kl} e^{-ikx} = J_i \eta m$$
Hence, the final expression for the discrete 2-D inverse Fourier transform is

\[ J_{ij} = \frac{1}{nm} \sum_{k=-n}^{n} \sum_{l=-m}^{m} Z_{kl} e^{-i(kx_i + ly_j)} \]  

17.17)

The power spectrum is therefore given by (remembering that power now is split between the four quadrants)

\[ s_{ij} = 4 \left( J_{ij} \cdot J_{ij}^* \right) = 4 \left( J_i \cdot J_j \cdot J_i^* \cdot J_j^* \right) = 4 \left( J_i \cdot J_i^* \cdot J_j \cdot J_j^* \right) = 4 \left( \alpha \cdot \alpha^* \cdot \beta \cdot \beta^* \right) \]

\[ = \frac{1}{4} (\alpha^2 + \beta^2)(\alpha^* \beta + \beta^* \alpha) = \frac{1}{4} (\alpha^2 \beta^2 + \beta^2 \alpha^2 + \beta^2 \alpha^* + \beta^* \alpha^2) \]

(17.18)

which is the same expression we found earlier. We can generalize (17.16) and (17.17) for the continuous case and find

\[ F(u, v) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \left( \hat{f} \right) e^{-i2\pi(ux + vy)} \, dx \, dy \]

\[ \left( \hat{f} \right) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} F(u, v) e^{i2\pi(uu + vv)} \, du \, dv \]

(17.19)

These are the continuous 2-D forward and inverse Fourier transforms, respectively.

2-D SPATIAL FILTERING

We may want to filter 2-D data for the same reasons we filtered 1-D data: To eliminate noise, isolate particular bands of wavelengths, or obtain robust trends. The simplest filter we may imagine is a moving average filter. We looked at such filters when we treated the 2-D interpolation problem. The weights here are defined as

\[ w(x, y) = \begin{cases} 
1, & \left( x - x_0 \right)^2 + \left( y - y_0 \right)^2 \leq R^2 \\
0, & \text{elsewhere} 
\end{cases} \]

(17.20)

which means that we are moving a disc of radius \( R \) from node to node (at \( x_0, y_0 \)), locating all node values inside this circle, and return their mean value. This simple filter is the 2-D equivalent of the moving average or boxcar filter we employed to filter topography in 1-D. There we found that to extract the regional topography (the trend) we were better off by using the equivalently wide median filter. The 2-D median filter is analogous to the 1-D version: All data inside the disc-radius is found and their median is returned as the filtered value. This filter will preserve step-functions but eliminates short-wavelength spikes and outliers.
Sometimes the data distribution is particularly skewed and even the median won’t work well. Consider the case of data whose histogram looks like the one above. In such cases we would rather return the mode at the values instead of the median or mean value. You may remember that a good estimator of the mode is the LMS (Least Median of Squares) estimator which minimizes

\[
\text{median}\left\{\hat{y} - \hat{f}\right\}
\]

This estimate can be found by

1. Sorting the data.
2. Find shortest half \( d_i = (z_{h+i} - z_i), \ h = n/2 + 1 \)
3. LMS = \( 1/2 \left( z_{h+i} + z_i \right) \)

Thus a mode filter can be constructed the same way you would implement a median filter except one would find the mode rather than the median. Again, such robust filters have their strength in eliminating outliers, but can also be used to clip out prominent short-wavelength signal, which would only be blurred by, say, a simple moving average process. For this reason they may be used successfully to determine robust trends instead of traditional trend surface analysis techniques.

To summarize filters based on the mean, median, and mode:

- **mean** \( \hat{x} \) value that minimizes \( \sum\limits_{i=1}^{n} \left( x_i - \hat{x} \right)^2 \)
- **median** \( \hat{x} \) value that minimizes \( \sum\limits_{i=1}^{n} | x_i - \hat{x} | \)
- **mode** \( \hat{x} \) value that minimizes \( \text{median}\left\{\hat{x} - \hat{f}\right\} \)

We found that the Gaussian filter was one of the best filters we could use on the 1-D time-series we looked at earlier. An isotropic 2-D Gaussian filter will only depend on the radius \( r \) from the node, hence we can write it as

\[
w(x, y) = e^{-\frac{1}{2\sigma^2} \left[ (x-x_0)^2 + (y-y_0)^2 \right]} = e^{-\frac{1}{2\sigma^2} (x-x_0)^2} e^{-\frac{1}{2\sigma^2} (y-y_0)^2}
\]

which is simply the product of two 1-D Gaussians. This makes it particularly easy to find the Fourier transform of the filter. Applying (17.19) we find
\[ W(u, v) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} e^{-\frac{1}{2\sigma^2}[(x-u)^2 + (y-v)^2]} dx dy = \int_{-\infty}^{\infty} e^{-\frac{1}{2\sigma^2}(x-u)^2} e^{-iux} dx \int_{-\infty}^{\infty} e^{-\frac{1}{2\sigma^2}(y-v)^2} e^{-ivy} dy \]

Because of the separation of \( x \) and \( y \) dependencies we may use our knowledge of the transform of a 1-D Gaussian to find

\[ W(u, v) = 2\pi\sigma^2 e^{-2\pi\sigma^2 u^2} e^{-2\pi\sigma^2 v^2} = 2\pi\sigma^2 e^{-2\pi\sigma^2 k_r^2} \]  

(17.22)

Thus the Fourier transform is isotropic as well and only depends on the radial wave number. Because its transform is itself a Gaussian, we know that Gaussian filtering in 2-D will not produce ripples or “ringing” either.

So the Gaussian filter in 2-D is also a Gaussian in the wave number domain. What is the spectral effect of using a spatial moving average filter? We know that in 1-D a sinc function results. In 2-D we must take advantage of circular symmetry, since our weights only depend on \( r \). The 2-D transform is

\[ F(u, v) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} (f(x, y)) e^{-iux +ivy} dx dy \]

We must convert to polar coordinates. Using

\[ x + iy = re^{i\theta} \quad u - iv = qe^{i\phi} \]

we form \((x + iy)(u - iv) = xu + yv + i(uy - vx) = rqe^{i(\theta - \phi)}\). The transform becomes

\[ F(q) = \int_{0}^{\infty} \int_{0}^{2\pi} (\mathcal{Q}) e^{-irq \cos \phi} r dr d\theta \]

since we only need the real part of the exponent. We find

\[ F(q) = \int_{0}^{\infty} (\mathcal{Q}) \left[ \int_{0}^{2\pi} e^{-irq \cos \phi} d\phi \right] r dr = \int_{0}^{\infty} (\mathcal{Q}) J_0(qr) r dr \]

(17.23)

since the inner integral is known from tables and \( J_0 \) is the zero-order Bessel function. Also,

\[ (\mathcal{Q}) = \int_{0}^{\infty} F(q) J_0(qr) r dq \]

(17.24)

The circular (cylindrical) Fourier transform is also known as the Hankel transform. Thus, the Hankel transform of our MA filter will give the isotropic 2-D equivalent:

\[ W(q) = \int_{0}^{\infty} J_0(qr) r dr = \frac{RJ_1(q)}{q} \]

(17.25)

where \( J_1 \) is the 1st order Bessel function. \( W(q) \) looks now very much like the sinc function in 1-D. However, it differ in that the zero crossings are not equidistant.
Anyway, \( w(r) \) will introduce ringing in the frequency domain.

As we have seen earlier, most filtering operations are convolutions. While slow in 1-D, they are even slower to execute for 2-D data. The convolution theorem makes 2-D filtering much easier and faster. The recipe for 2-D filtering therefore becomes:

1. Take the 2-D Fourier transform of \( f \) and \( g \) separately
2. Multiply the two transforms at all frequencies
3. Take the inverse 2-D transform of the product.

All the concerns we raised during our discussion of the 1-D transform are common to the 2-D case as well. Aliasing may occur if our sampling interval is not small enough, and leakage may occur if function \( f(x,y) \) is not periodic. Consequently, one should pay attention to picking the best Nyquist frequency and, if necessary, make sure the signal is periodic by either tapering or mirroring or both.

Isotropic band-pass filters are designed the same way we did it in 1-D. Here, the frequency-dependent filter is only a function of the radial wave number so that a 1-D (radial) filter results. Again, since the "steepness" of the filter determines how narrowly the filter can separate different wavelengths, ringing will increase in intensity as the filter approaches a gate-function.

While for the vast majority of applications it makes sense to use isotropic filters, there may be situations that call for anisotropic or direction-dependent filters. Consider the case of a device mounted on the hull of a ship that measures bathymetry in a swath across the ship track. Because the ship is bobbing up and down in the sea, the relative depth will oscillate with the heave. Thus the recorded signal will have a short wave length undulation along the direction of motion. A filter that is more restrictive in the along-track direction that across track may remove this artifact. Or consider localized magnetic anomalies superimposed on longer-wavelength undulations trending at some angle \( \theta \) with wavelength \( \lambda \). One could design a 2-D notch-filter in the wave number domain, which could be used to suppress these undulations whose power is concentrated around \((u_0, v_0)\):

\[
W(u,v) = 1 - e^{-\frac{1}{2\sigma^2}[(u-u_0)^2 + (v-v_0)^2]}
\]  

(17.26)

for some notch-width \( 6\sigma \).

Finally, it is clear that real data may contain outliers as well as undesired short-wavelength information. In such simulations a 2-step procedure may yield the best result:

1. In the space-domain, remove outliers with a robust filter (median or mode)
2. In space or wave number domain, apply conventional filter on the clean data set.

The resulting filtered data set will be smooth even if the original data was spiky.