Factorization of cross spectra

Jon Claerbout, jon/res/xkolmog/[1]

keywords: *factoring, spectra, crosscorrelation, 3-D*

A solved problem is the factorization of a positive real autospectrum into a minimum-phase wavelet and its adjoint. The most practical method is that of Kolmogoroff. Here I extend the Kolmogoroff method to cross-spectra.

This problem arises in the extrapolation of 3-D wavefields where we need to factor an operator like $i - \nabla^2$. We get a band matrix to solve. In principle, we factor it into lower and upper triangular band matrices which we then backsolve. Except at the ends, this is equivalent to a filter problem where the two backsubstitutions are polynomial divisions, one causal, the other anticausal. Although $-\nabla^2$ is an autocorrelation, $i - \nabla^2$ is not, so we need two different minimum-phase filters whereas the Kolmogoroff method gives us the same one for both the causal and anticausal operations.

Level-phase functions

I define a "level-phase function" to be one for which the phase of the Fourier transform at minus the Nyquist frequency is the same as the phase at plus Nyquist frequency.

An example of a function that is not "level-phase" is the delayed impulse $Z = e^{i\omega\Delta t}$. The phase of this function is $\omega\Delta t$ so at the minus Nyquist frequency it is $-\pi\Delta t$ while at the plus Nyquist frequency it is $\pi\Delta t$. (The complex function $Z = e^{i\omega\Delta t}$ has a real part that is a cosine and an imaginary part that is a sine; in the complex plane this function is a circle, looping around the origin as ω increases. The phase is the arctangent of the ratio of the imaginary to the real part, and it steadily increases. The average phase is not level but tilted.)

Examples of functions that are level-phase are those causal wavelets with a causal inverse, known in the geophysical world as "minimum-phase wavelets."

Unlike a minimum-phase wavelet, a level-phase function need not be causal. I have been noticing level-phase functions for some time, but only recently recognized their essential features. Consider for example $i - \nabla^2$. In one dimension, it is $i - \partial_{xx}$. Expressing it in finite differences, it is an autocorrelation function $(-1, 2, -1)$ with the imaginary number i added to the zero lag getting $(-1, 2 + i, -1)$. Expressing $i - \partial_{xx}$ as a Z transform, it is $i + (2 - \cos(k_x\Delta x))/\Delta x^2$. This function is the positive

[1]**email:** jon@sep.stanford.edu, http://sepwww.stanford.edu/sep/jon/

imaginary constant i plus a positive real spectrum. Thus for all real values of the frequency k_x, its phase angle stays in the upper right quarter of the complex plane so it cannot wrap around the origin, as does the phase of $e^{ik_x \Delta x}$. Thus $i - \partial_{xx}$ is level-phase and using the logic of the helix, the operator $i - \nabla^2$ is also level-phase.

Let us add a causal wavelet to an anticausal wavelet. Using Z-transform polynomials in positive powers of Z the sum can be denoted as

$$U(Z) = \overline{B}(1/Z) + A(Z) \tag{1}$$

Let us exponentiate this sum:

$$X(Z) = e^{U(Z)} = e^{\overline{B}(1/Z)+A(Z)} \tag{2}$$

The heart of the matter is that the phase of $X(Z)$ is the imaginary part of $U(Z)$, and the imaginary part of $U(Z)$ is a convergent sum of sines (and maybe cosines). A sum of sines is periodic with one period going from minus to plus Nyquist. As a phase, it fits the definition of "level phase." Thus an arbitrary function $U(Z)$ always constructs a level-phase function $X(Z)$.

Although u_t is an arbitrary time function from which we could always construct another time function x_t, the reverse is not true. There exist time functions x_t for which there is no corresponding u_t. The example that we have seen is $X(Z) = Z$.

The reason we cannot always construct a $U(Z)$ from any possible $X(Z)$ is that we cannot always take logarithms. When poles and zeros are in the wrong place in the complex plane, the power series for logarithm diverges.

There is no requirement on $X(Z)$ other than that it be level-phase. This is so because convergent Fourier sums can represent almost any analytic function. Since they are periodic, the one thing they cannot make is a function whose value at minus Nyquist differs from that at plus Nyquist. In summary, for x_t to be represented as an exponential with (2), the necessary and sufficient condition is that it be a level-phase function.

For a while I mistakenly thought that $X(Z)$ could be taken to be an arbitrary crosscorrelation function. Now we see that this is not so because Z is a crosscorrelation function (a white signal crosscorrelated with itself delayed). Evidently, any crosscorrelation function can be shifted to become a level-phase function. Thus we now have a representation for any crosscorrelation function in terms of two minimum-phase wavelets and a delay.

The exponential of a causal is causal.

Begin with a causal filter response c_t and its associated $C(Z)$. The Z-transform $C(Z)$ could be evaluated, giving a complex value for each real ω. This complex value could be exponentiated to get another value, say

$$B(Z(\omega)) \quad = \quad e^{C(Z(\omega))} \tag{3}$$

Next, we could inverse transform $B(Z(\omega))$ back to b_t. We will prove the amazing fact that b_t must be causal too.

First notice that if $C(Z)$ has no negative powers of Z, then $C(Z)^2$ does not either. Likewise for the third power or any positive integer power, or sum of positive integer powers. Now recall the basic power-series definition of the exponential function:

$$e^x \quad = \quad 1 + x + \frac{x^2}{2} + \frac{x^3}{2 \cdot 3} + \frac{x^4}{2 \cdot 3 \cdot 4} + \frac{x^5}{2 \cdot 3 \cdot 4 \cdot 5} + \cdots \tag{4}$$

Including equation (3) gives the exponential of a causal:

$$B(Z) \quad = \quad e^{C(Z)} \quad = \quad 1 + C(Z) + \frac{C(Z)^2}{2} + \frac{C(Z)^3}{2 \cdot 3} + \frac{C(Z)^4}{2 \cdot 3 \cdot 4} + \cdots \tag{5}$$

Each term in the infinite series corresponds to a causal response, so the sum, b_t, is causal. The factorials in the denominators assure us that the power series always converges, i.e., it is finite for any finite x.) The inverse wavelet to $B(Z)$ is also causal because it is $e^{-C(Z)}$. (Unfortunately the words "minimum phase" distract people from the equivalent property of genuine interest, that the causal wavelet has a causal inverse so we can use feedback filters.)

Putting one polynomial into another or one infinite series into another is an onerous task, even if it does lead to a wavelet that is exactly causal. In practice we do operations that are conceptually the same, but for speed we do them with discrete Fourier transforms. The disadvantage is periodicity, i.e., negative times are represented computationally like negative frequencies. Negative times are the last half of the elements of a vector, so there can be some blurring of late times into negative ones. The subroutine we will use for Fourier transformation is `fts()`

```
subroutine fts( signi, nx, rr )
#   complex fourier transform.  if (signi>0) scale = 1; else scale=1/nx
#
#                      nx          signi*2*pi*i*(j-1)*(k-1)/nx
#   rr(k)   =  scale * sum rr(j) * e
#                     j=1             for k=1,2,...,nx=2**integer
#
integer nx, i, j, k, m, istep, pad2
real    signi, arg
complex rr(nx), cmplx, cw, cdel, ct
if( nx != pad2(nx) )    call erexit('fts: nx not a power of 2')
if( signi < 0.)
        do i= 1, nx
                rr(i) = rr(i) / nx
j = 1;  k = 1
do i= 1, nx {
        if (i<=j) { ct = rr(j); rr(j) = rr(i); rr(i) = ct }
        m = nx/2
        while (j>m && m>1) { j = j-m; m = m/2 }         # "&&" means .AND.
        j = j+m
        }
```

```
repeat {
        istep = 2*k;   cw = 1.;   arg = signi*3.14159265/k
        cdel = cmplx( cos(arg), sin(arg))
        do m= 1, k {
                do i= m, nx, istep
                        { ct=cw*rr(i+k);  rr(i+k)=rr(i)-ct;  rr(i)=rr(i)+ct }
                cw = cw * cdel
                }
        k = istep
        if(k>=nx) break
        }
return; end

integer function pad2( n )
integer n
pad2 = 1
while( pad2 < n )
        pad2 = pad2 * 2
return; end
```

Code for crosscorrelation factorization

The code below will factor the polynomial $(2/Z + 7 + 3Z)$ into its two factors $(3 + 1/Z)(2 + Z)$ except for a scale factor which is indeterminate. Choosing to set $u_0 = 0$ leads to the interesting factorization $(2/Z+7+3Z)/6 = (1+1/(3Z))(1+Z/2)$ where the scale is chosen so the coefficient of Z^0 is 1 (a convenience when you have factored a band matrix into two parts, and plan polynomial division for back substitution). Being based on fast FT, the work space size, `n`, must be a power of 2. Periodicity of FT requires that one side of the crosscorrelation is at the beginning of `rr(:)` while the other side is at the end. A test case for $(2/Z + 7 + 3Z)$ is:

```
rr( n)  =  2.       # first negative lag
rr( 1)  =  7.       # zero lag
rr( 2)  =  3.       # first positive lag
```

```
subroutine crossfac( n, rr, aa, bb)  # Crosscorrelation factorization.
integer i,          n                # input: rr= crosscorrelation (destroyed)
complex rr(n), aa(n), bb(n)          # output: aa and bb are minimum phase.
call fts(  1., n, rr)                         # make spectrum
rr = clog( rr )                               # log spectrum
call fts( -1., n, rr)                         # back into time domain.
rr(1) = 0.                                    # Normalize coef of Z^0
aa = rr
bb = rr
do i= 2, n/2 {  bb(i)     = 0.                # Erase positive times.
                aa(n-i+2) = 0.                # Erase negative times.
                }                             # Halve both zero and Nyquist.
aa(1) = aa(1)/2.;    aa(1+n/2) = aa(1+n/2)/2.
bb(1) = bb(1)/2.;    bb(1+n/2) = bb(1+n/2)/2.
call fts( +1., n, aa)                         # back to frequency
```

```
call fts( +1., n, bb)
bb = cexp(  conjg( bb))                          # conjugate reverses time.
aa = cexp(         aa )
call fts( -1., n, aa)                            # minimum-phase wavelet.
call fts( -1., n, bb)                            # MP wavelet for anticausal.
return; end
```

To avoid any ambiguity, here is my test case for complex-valued coefficients:

$$\begin{aligned} && e^A &= 2+Z && (6)\\ && e^B &= 3+iZ && (7)\\ R &= & e^U &= e^{\overline{B}(1/Z)+A(Z)} &= -2i/Z+(6-i)+3Z \quad & (8) \end{aligned}$$

with an output of

$$e^A = 1+Z/2 \tag{9}$$

$$e^B = 1+iZ/3 \tag{10}$$

For this case I found the errors are less than 1% when `n=8`. Generally, the larger the dynamic range of $|R|$ on the unit circle the greater the need for a higher `n`.