Overthrust Migration by Depth Extrapolation

Jon F. Claerbout

It is generally believed that migration by downward continuation is limited to dips less than 90. Proper handling of evanescent energy overcomes this limitation. At the present time, most migration-by-depth-extrapolation programs ignore or set to zero the energy that turns evanescent. The proper thing to do with energy becoming evanescent at depth z is to save it for a second pass *upward*. The upward pass begins from the bottom of the section with a zero downgoing wave. As the downgoing wave is propagated upward, the saved evanescent energy is reintroduced. As usual, the images are withdrawn from the wave at time $t = 0$.

To illustrate the concept, a program will be sketched that makes two images, first the conventional image, and second the underside image, presumably the underside of overthrust. The images may be viewed separately or summed.

We will begin with the simplifying restriction on the velocity that $dv / dz \geq 0$. (This assumption prevents internal "surface" waves. I do not believe that this restriction is a serious limitation. Anyway, we can return to investigate it later.) Because of this assumption, evanescent energy can be stored "in place" and ignored until the return pass. It is worth noting that the second pass is cheaper than the first pass because the region in which evanescence never occurred, $|k| < |\omega| / v(\tau_{max})$, need not be processed.

```
# first pass of conventional phase-shift migration.
P(ω,k_x) = FT[u(t,x)]
For τ = Δτ, 2Δτ, ···, τ_max {
        For all k_x {
                Uimage(k_x,τ) = 0.
                For all ω > |k| v(τ) {
                        C = exp( -i ω Δτ sqrt(1 - v(τ)^2 k_x^2 / ω^2) )
                        P(ω,k_x) = P(ω,k_x) * C
                        Uimage(k_x,τ) = Uimage(k_x,τ) + P(ω,k_x)
                        }
                }
        uimage(x,τ) = FT[Uimage(k_x,τ)]
        }

# Second pass for underthrust image.
For τ = τ_max, τ_max-Δτ, τ_max-2Δτ, ···, 0 {
        For all k_x {
                Dimage(k_x,τ) = 0.
                For ω = |k| v(τ) to ω = |k| v(τ_max) {
                        # The wave changes direction but so does Δτ
                        C = exp( -i ω Δτ sqrt(1 - v(τ)^2 k_x^2 / ω^2) )
                        P(ω,k_x) = P(ω,k_x) * C
                        Dimage(k_x,τ) = Dimage(k_x,τ) + P(ω,k_x)
                        }
                }
        dimage(x,τ) = FT[Dimage(k_x,τ)]
        }
```