We reviewed gridding algorithms discussed in earlier lectures. For gridding with Green’s functions we produced this table of Green’s functions for the various geometries and tension (regularized splines means all higher order derivatives exist and that space is still Cartesian):

<table>
<thead>
<tr>
<th>Dimension</th>
<th>No tension</th>
<th>Tension p</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-D Cartesian</td>
<td>( r^2 )</td>
<td>( e^{-pr} + pr )</td>
</tr>
<tr>
<td>2-D Cartesian</td>
<td>( r^2 \log r )</td>
<td>( K_0(pr) + \log(pr) )</td>
</tr>
<tr>
<td>3-D Cartesian</td>
<td>( r )</td>
<td>( \left( e^{-pr} - 1 \right)/pr + 1 )</td>
</tr>
<tr>
<td>2D spherical</td>
<td>dilog( (\sin^2 \frac{\theta}{2}) )</td>
<td>( aP_v(-\cos \theta) - b \log(1 - \cos \theta) )</td>
</tr>
<tr>
<td>2-D regularized</td>
<td>( \log\left( \frac{pr}{2} \right) )</td>
<td>( E_1\left( \frac{pr}{2} \right) )</td>
</tr>
<tr>
<td>3-D regularized</td>
<td>( \frac{1}{pr} ) erf( \left( \frac{\pi}{2} \right) )</td>
<td>( -\frac{1}{\sqrt{\pi}} )</td>
</tr>
</tbody>
</table>

Table 1. List of Green’s functions for splines with or without tension in various geometries.

I have updated the routines used in Lab 4 to include the two regularized splines in tension functions listed in the table. Here, \( K_0 \) is the modified Bessel function of the second kind and order zero, dilog is the dilogarithm, \( P_v \) is the Legendre function, \( E_1 \) is the exponential integral, and erf is the error function.

**PROJECTION ONTO CONVEX SETS (POCS)**

We will end our discussion of gridding methods by looking at a recent method called POCS (Projection onto Convex Sets). POCS may be thought of as an inversion technique for finding an unknown function that satisfies certain known properties. Let us call the function \( z(x,y) \). It is only known at the \( n \) discrete points \((x_i, y_i)\) where it equals \( z_i \). The interpolation problem is to “fill in” the function between these points in a sensible way. To do this sensibly means the user knows that the function has certain other properties, such as positivity, smoothness, as well.

The notion that a function has a given property (e.g. smoothness) is the same as saying that it belongs to a set of functions with that property (e.g., all functions that are everywhere positive). Other sets may be identified. The solution to the gridding or interpolation problem then becomes to find the area in function space that represents the intersection of all the sets we want our function to belong to.

Finding this area can be done by “projecting your current guess of the function orthogonally onto the surfaces of the sets”. We will see what this cryptic statement means. Repeating this process will lead to convergence provided all sets are convex, meaning if \( z_1(x,y) \) and \( z_2(x,y) \) are elements of the set, then \( z(x,y) = \alpha z_1(x,y) + (1-\alpha)z_2(x,y) \) is also an element, for all \( 0 \leq \alpha \leq 1 \).
The above figure is a geometrical visualization of how we could iteratively find the intersection of all applicable sets of functions, which then would contain all functions that have the desired characteristics. Any one of these functions is acceptable as a solution since we do not have any further requirements that we could use in choosing among the remaining functions.

The initial guess for the solution \( z(x,y) = \) white noise or \( z(x,y) = 0 \) are two possible candidates) is improved upon by projecting this solution orthogonally onto the surfaces of all the convex sets chosen as constraints. Consider the initial guess picked above. Here we have the sets \( A, B, C \). Let \( A \) be the set of all functions that go through our data constrains, \( B \) the set of all functions that are positive everywhere, and \( C \) the set of functions with minimum curvature. Assume we first project it onto \( A \). This simply means resetting all node values in \( z(x,y) \) that are near a data constraint to that value. From this new solution \( z(x,y) \) we project onto \( B \), which means we reset all negative values to zero, and finally project the solution onto set \( C \) which means that the power spectrum must fall off rapidly with wave number (since smoothness means little power at short wavelengths). We now have a solution that is inside all sets. Often several more iterations will be needed before we reach the intersection. These paths will always converge on the intersection provided all sets are convex.

There are many convex sets that are relevant to the interpolation problem. We will look at several and see what the orthogonal projection means in each instance.

- **Set P** (for “Point”). \( z(x,y) \) must equal \( z_i \) at the \( n \) given data points \((x_i, y_i)\). The projection means simply to reset \( z(x,y) \) to \( z_i \) at those points (remember \( z(x,y) \) is approximated by a matrix).

- **Set B** (for “Bounded”). Function is bounded by two known surfaces \( L(x,y) < z(x,y) < U(x,y) \). Projecting onto \( B \) means to truncate points that exceed the lower/upper bounds by replacing them with the lower/upper limit at that point.

- **Set F** (for “Fault”). Function has a jump discontinuity of size \( z(x^+,y^+) - z(x^-,y^-) = h(x,y) \) across a known boundary \( c(x,y) = 0 \) that divides it into 2 regions, \( R_1 \) and \( R_2 \). Project by adding \( h(x,y)/2 \) to \( z(x,y) \) just inside \( R_1 \) and subtract \( h(x,y)/2 \) from \( z(x,y) \) just inside \( R_2 \).

- **Set H** (for "Higher than"). Function satisfies \( z(x_1, y_1) \geq z(x_2, y_2) \) for two points 1 and 2. Project by setting \( z(x_1, y_1) = z(x_2, y_2) = \text{<their mean>} \) only if \( z(x_1, y_1 < z(x_2, y_2)) \).
• Set \( \mathbf{R} \) (for “River”). Function should monotonically increase along the curve \([x(s), y(s)]\). Treat by using \( \mathbf{H} \) on all points along the river in sequence.

• Set \( \mathbf{M} \) (for “Mean”). Function’s mean should equal \( m \). Project by removing mean and adding \( m \). Often, \( m \) will be zero.

• Set \( \mathbf{E} \) (for “Energy”). Functions energy, \( E = \int z^2(x,y) \, dS \) should be less than \( E_0 \). Project by normalizing \( E \) to \( E_0 \).

• Set \( \mathbf{K} \) (for “Wave number K”). The Fourier transform, \( Z(k_x, k_y) \), is given at discrete wave numbers. Project by taking transform, reset the amplitude spectrum where specified, and transform back.

• Set \( \mathbf{S} \) (for “Spectrum”). The power spectrum is known a priori. Project by computing power spectrum and reset the amplitude spectrum at wave numbers where the power exceeds the specified value, and transform back (do not alter phase spectrum).

Once the user has decided which sets to use, we start with an initial guess (could be white noise) and iteratively apply the various projections. We stop when no significant change can be detected. This means that the solution must belong in each of the sets and thus has the desired properties.

POCS’ strength is the ability to include complicated constraints to the problem. It can therefore be used to interpolate empty regions in a way that looks “realistic”, e.g., it has texture similar to those regions that are well constrained, or have certain statistical properties. Its main use will be situations that call for rough interpolations since simpler schemes for smooth interpolation already exists.

**GRIDDING REFERENCES**


Additional references on kriging is given in Davis.