Bivariate Interpolation

• Interpolation still means exact reproduction of data.

• The word bivariate refers to having two independent variables.

• Also common are the words trivariate (three independent variables), multivariate (more than one independent variable), and univariate (one independent variable).

• Much of what we’ll do in the bivariate context carries over to the more general multivariate context.

• However, there is a vast difference between univariate and multivariate contexts.

• We’ll consider five different kinds of interpolation:
  – Tensor Product
  – Polynomial
  – Transfinite
  – piecewise polynomial on triangles
  – meshless, or radial basis functions.
Tensor Product

- The phrase tensor product refers to a situation where the data lie on a rectangular grid, and we basically do univariate interpolation twice.

- Suppose we are given data

\[(x_i, y_j, f(x_i, y_j)), \quad i = 0, \ldots, m, \quad j = 0, \ldots, n\]
• We want an interpolant $p$ satisfying

$$p(x_i, y_j) = f(x_i, y_j)$$

for all $i$ and $j$.

• An important basic idea is to consider operators that do the job in one of the variables and then compose, or otherwise combine them.

• Consider polynomial interpolation. Define

$$P_x f(x, y) = \sum_{i=0}^{m} f(x_i, y)L_i(x) \quad \text{where} \quad L_i(x) = \frac{\prod_{j \neq i} (x - x_j)}{\prod_{j \neq i} (x_i - x_j)}.$$  

Clearly $P_x f$ interpolates in $x$ for all values of $y$. Similarly, we define

$$P_y f(x, y) = \sum_{j=0}^{n} f(x, y_j)M_j(x) \quad \text{where} \quad M_j(y) = \frac{\prod_{i \neq j} (y - y_i)}{\prod_{i \neq j} (y_j - y_i)}.$$
The overall interpolant is then simply the composition of those two interpolants.

\[
P f = P_x P_y f
\]

\[
= P_x \sum_{j=0}^{n} f(x, y_j) M_j(y)
\]

\[
= \sum_{i=0}^{m} \left( \sum_{j=0}^{n} f(x_i, y_j) M_j(y) \right) L_i(x)
\]

\[
= \sum_{i=0}^{m} \sum_{j=0}^{n} f(x_i, y_j) M_j(y) L_i(x)
\]

\[
= p(x, y)
\]

Clearly, \(p\) interpolates at all points \((x_i, y_j)\).
Bilinear Interpolation

• For simplicity, consider the unit square.

• Let

\[ P_x f(x, y) = (1 - x)f(0, y) + xf(1, y) \]

and

\[ P_y f(x, y) = (1 - y)f(x, 0) + yf(x, 1) \] (1)

• Clearly \( P_x \) interpolates at all points \((0, y)\) and \( P_y \) interpolates all points \((x, 0)\) and \((x, 1)\).

• The composition gives the \textbf{bilinear} interpolant to the function values at the vertices of the unit square.

\[ P f(x, y) = P_x P_y f(x, y) \]

\[ = P_x ((1 - y)f(x, 0) + yf(x, 1)) \]

\[ = (1 - x)((1 - y)f(0, 0) + yf(0, 1)) \]

\[ + x((1 - y)f(1, 0) + yf(1, 1)) \]

\[ = (1 - x)(1 - y)f(0, 0) \]

\[ + (1 - x) yf(0, 1) \]

\[ + x (1 - y)f(1, 0) \]

\[ + xyf(1, 1) \]

\[ = A + Bx + Cy + Dxy \]

• The interpolant is \textbf{bilinear} which means that setting one variable to a constant gives a function that is linear in the other variable.
• Note that the restriction of the bilinear function to a line, such as \( y = x \), in general gives a **quadratic** function.
Transfinite Interpolation

- We illustrate the idea by interpolating everywhere on the boundary of the unit square.

- This is not as far fetched as it may sound. For example, the data could come from a grid that approximate the shape of a vehicle such as a car or aircraft. The curves along the grid lines could have been designed by some univariate scheme. Or we might have a vehicle, such as a survey ship, move along a line and collect data densely along the line.

- Let $P_x$ and $P_y$ be the linear operators defined in (1).

- The following **bilinearly blended Coon’s patch** interpolates on the boundary of the unit square. It is also called the **Boolean Sum** of the operators $P_x$ and $P_y$.

\[
Qf = P_x f + P_y f - P_x P_y f \\
= (1 - x)f(0, y) + xf(1, y) \\
+ (1 - y)f(x, 0) + yf(x, 1) \\
- (1 - x)(1 - y)f(0, 0) \\
- (1 - x)yf(0, 1) \\
- x(1 - y)f(1, 0) \\
- xyf(1, 1)
\]

- Check interpolation
Exercise: try cubic Hermite interpolation on the unit square, finite and transfinite.
Scattered Data

- The phrase “scattered data” means that the data sites do not form a rectangular grid.
- So we are given data

\[(x_i, y_i, z_i), \quad i = 1, 2, \ldots, n\]

and we want an interpolant \(P\) such that

\[P(x_i, y_i) = z_i, \quad i = 1, 2, \ldots, n.\]

- The natural first step would be to attempt interpolation by a polynomial.
- The immediate stumbling block is that there is no unique polynomial associated with the number of data.
- A bivariate polynomial \(p\) of degree \(d\) is defined by

\[p(x, y) = \sum_{i+j \leq d} \alpha_{ij} x^i y^j\]
• How many parameters does a polynomial of degree $d$ have?
Linear Interpolation
Quadratic Interpolation
• Maybe we shouldn’t use polynomials. Maybe there are better functions.

• Let

\[ p(x, y) = \sum_{i=1}^{n} \alpha_i \phi_i(x, y). \]

• How do we pick the basis functions \( \phi_i \)?

\[ \hat{\phi} \]

It turns out that no matter how we pick them, there are Lagrange Interpolation problems with distinct data sites such that the Vandermonde matrix is singular.