

1 Multivariate Interpolation on Non-Rectilinear Grids

This section presents alternative interpolation methods for non-rectilinear grids. First, I present the relatively simple case of fast warped interpolation on a curvilinear grid, which improves upon the interpolation in White (2015). Then, I present a machine learning approach to interpolation on unstructured grids based on Gaussian Process Regression as presented in Scheidegger and Bilonis (2019).

1.1 Warped Grid Interpolation (WGI)

Assume we have a set of points indexed by (i, j) in two-dimensional space for which we have corresponding functional values in a third dimension, such that $f(x_{ij}, y_{ij}) = z_{ij}$. In practice, we are interested in cases where the z_{ij} are difficult to compute and $f(x_{ij}, y_{ij})$ is unknown, so we are unable to compute them at other values of x and y — which is why we want to interpolate¹.

These (x_{ij}, y_{ij}) points however are not evenly spaced and do not form a rectilinear grid which would make it easy to interpolate the function off the grid. Nevertheless, these points do have a regular structure as we will see.

In Figure 3, we can see the true function in 3D space, along with the points for which we actually know the value of the function.

The underlying regular structure comes from the points' position in the matrix, the (i, j) coordinates. If we join the points along every row and every column, we can see that the resulting grid is regular and piecewise affine (curvilinear).

In Figure 4 we see the values of the function at their index coordinate points in the matrix. We can see that there exists a mapping between the curvilinear grid and the index coordinates of the matrix.

The objective is to be able to interpolate the value of the function at any point off the grid, where presumably we are only interested in points internal to the curvilinear space and not outside the boundaries. For example, we can imagine that we want an approximation to the function at the point $(x, y) = (3, 5)$ pictured Figure 5. If we could find the corresponding point in the coordinate grid, interpolation would be straightforward.

We can find where the x-coordinate of the point of interest intersects with the index-coordinates of the matrix.

This is similar to assuming that we have 3 linear interpolators formed by connecting the points on the green lines in the x-direction, and for each interpolator we can approximate the corresponding y and z values using the grid data.

¹For this illustration, we generate z 's arbitrarily using the function

$$f(x, y) = (xy)^{1/4}.$$

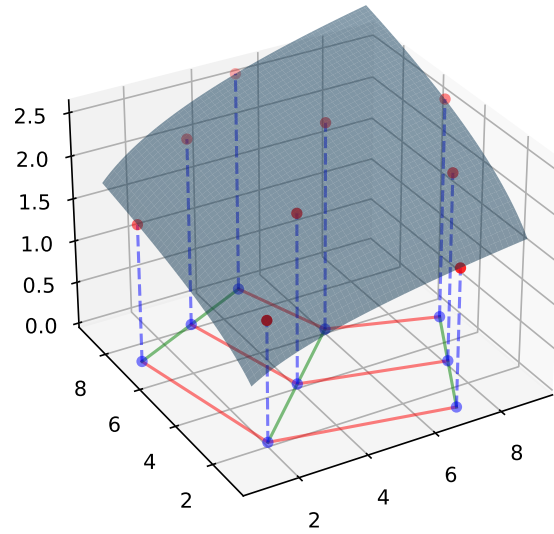


Figure 1 True function and curvilinear grid of points for which we know the value of the function.

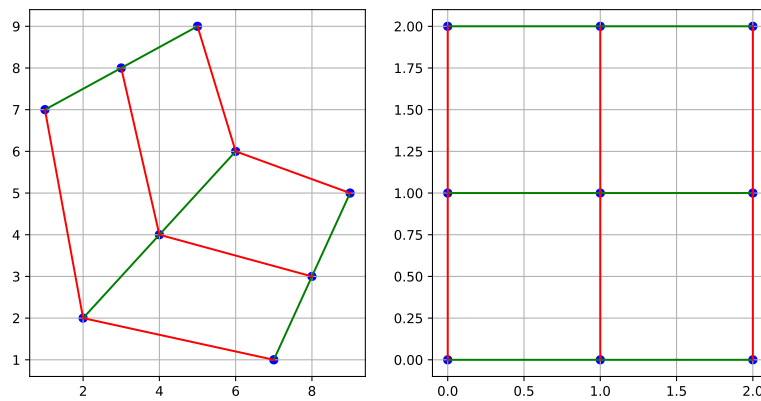


Figure 2 Homotopy between the curvilinear grid and the index coordinates of the matrix.

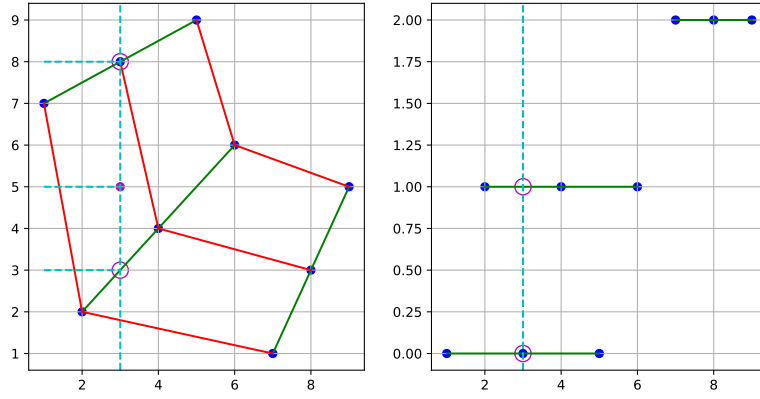


Figure 3 The method consist of extending the loci of points in the x dimension to find the corresponding crossing points in the y dimension.

Now, for each circle in Figure 5, we have a corresponding pair (y, z) , and we can interpolate in the y -direction to find the corresponding z -value for the point's y -coordinate².

1.2 Unstructured Grids

Unstructured interpolation arises in many dynamic programming applications when using the endogenous grid method because the first order conditions might be highly non-linear and non-monotonic, or because boundary constraints induce kinks in the policy and value functions. In these cases, the grid points generated by the EGM step are not evenly spaced, leading to the need for curvilinear interpolation. We saw in the previous subsection an approach to curvilinear interpolation based on White (2015) that is incapable of interpolation on structured grids. A similar approach was presented in Ludwig and Schön (2018) which used Delaunay interpolation. However, this approach is not well suited for our purposes because triangulation can be computationally intensive and slow, often offsetting the efficiency gains from the Endogenous Grid Method.

As an alternative to these methods, I introduce the use of Gaussian Process Regression (GPR) along with the Endogenous Grid Method. GPR is computationally efficient, and tools exist to easily parallelize and take advantage of hardware such as Graphics Processing Units (GPU)³.

1.2.1 Gaussian Process Regression

A Gaussian Process is an infinite dimensional random process for which every subset of random variables is jointly Gaussian or has a multivariate normal distribution.

²For more examples of the Warped Grid Interpolation method in action, see the github branch [alanlujan91/HARK@map_coordinates](#).

³Gardner, Pleiss, Bindel, Weinberger, and Wilson (2018)

$$\begin{aligned} \mathbf{X} &\sim \mathcal{N}(\mu, \Sigma) \quad \text{s.t.} \quad x_i \sim \mathcal{N}(\mu_i, \sigma_{ii}) \\ \text{and} \quad \sigma_{ij} &= \mathbb{E}[(x_i - \mu_i)(x_j - \mu_j)] \quad \forall i, j \in \{1, \dots, n\}. \end{aligned} \quad (1)$$

where

$$\mathbf{X} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \quad \mu = \begin{bmatrix} \mu_1 \\ \mu_2 \\ \vdots \\ \mu_n \end{bmatrix} \quad \Sigma = \begin{bmatrix} \sigma_{11} & \sigma_{12} & \cdots & \sigma_{1n} \\ \sigma_{21} & \sigma_{22} & \cdots & \sigma_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{n1} & \sigma_{n2} & \cdots & \sigma_{nn} \end{bmatrix}. \quad (2)$$

Being infinitely dimensional, a Gaussian Process can be used to represent a probability distribution over the space of functions in n dimensions. Thus, a Gaussian Process Regression is used to find the best fit function to a set of data points.

$$\mathbb{P}(\mathbf{f}|\mathbf{X}) = \mathcal{N}(\mathbf{f}|\mathbf{m}, \mathbf{K}) \quad (3)$$

where \mathbf{f} is the vector of function values at the points \mathbf{X} , \mathbf{m} is the mean of the function, and \mathbf{K} is a kernel function that describes the covariance between the function values at different points.

A standard kernel function is the squared exponential kernel, or the radial basis function kernel, which is defined as

$$k(\mathbf{x}_i, \mathbf{x}_j) = \sigma_f^2 \exp \left(-\frac{1}{2l^2} (\mathbf{x}_i - \mathbf{x}_j)' (\mathbf{x}_i - \mathbf{x}_j) \right). \quad (4)$$

Using GPR to interpolate a function f , we can both predict the value of the function at a point \mathbf{x}_* and the uncertainty in the prediction, which provides useful information as to the accuracy of the approximation.

References

- GARDNER, JACOB R, GEOFF PLEISS, DAVID BINDEL, KILIAN Q WEINBERGER, AND ANDREW GORDON WILSON (2018): “GPpyTorch: Blackbox Matrix-Matrix Gaussian Process Inference with GPU Acceleration,” in *Advances in Neural Information Processing Systems*.
- LUDWIG, ALEXANDER, AND MATTHIAS SCHÖN (2018): “Endogenous Grids in Higher Dimensions: Delaunay Interpolation and Hybrid Methods,” *Computational Economics*, 51(3), 463–492.
- SCHEIDEGGER, SIMON, AND ILIAS BILIONIS (2019): “Machine learning for high-dimensional dynamic stochastic economies,” *Journal of computational science*, 33, 68–82.

WHITE, MATTHEW N (2015): “The method of endogenous gridpoints in theory and practice,” *Journal of economic dynamics & control*, 60, 26–41.