Real Interpolation with Complex Variables

Harvey Barnhard

June 13, 2019

Contents

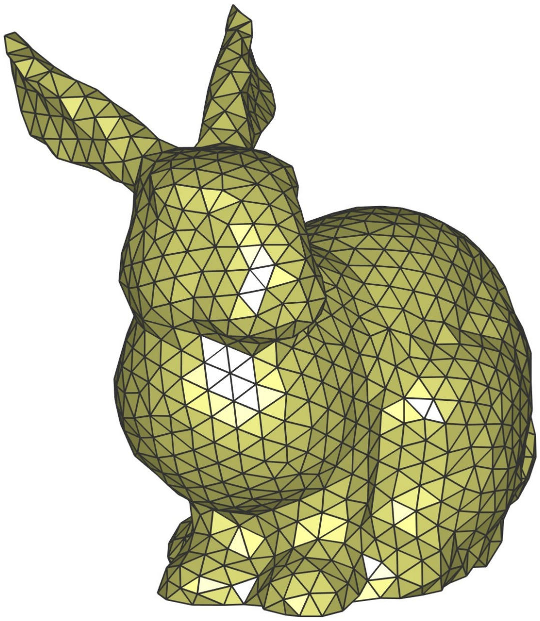

Figure 1: Triangular mesh on bunny (https://sites.google.com/site/dengwirda/jigsaw)

Introduction

In this paper, I motivate the use of complex analysis in prediction methods commonly used in electrical engineering, 3D modeling, and geostatistics. I hope to show the utility and relative simplicity of a method obscured by a turgid title. In particular, I aim to explain the "Complex variable moving Kriging interpolation for boundary meshless method" studied by Sanshan Tu and his co-authors (2016). For readers with little background in numerical analysis, the title of that paper will likely instill some convex combination of boredom and intimidation. The goal of these papers is to assuage some of those feelings. The components of the title can be broken down as follows:

- **Complex variable**: Working through the problem in the complex plane instead of curvilinear coordinates
- **Kriging interpolation**: Creating a predictor function by *interpolating* through the data points rather than creating a simpler fit that best *estimates* the data points
- **Boundary**: We only observe data points on the boundary of a region
- **Meshless**: Instead of partitioning the region into a collection of simple shapes (a mesh) to peform predictions, consider the whole region and perform predictions using a set of points in that region. See Figure 1 for an example of a mesh over a 3D bunny

To best explain the Complex Variable Moving Kriging Interpolation method (CVKMI), I build up from simpler prediction methods. I assume some familiarity of real and complex analysis as well as some linear algebra. After introducing CVKMI, I provide an example of how the method can be applied. I then compare the analytical and numerical results over a range of Kriging parameters.

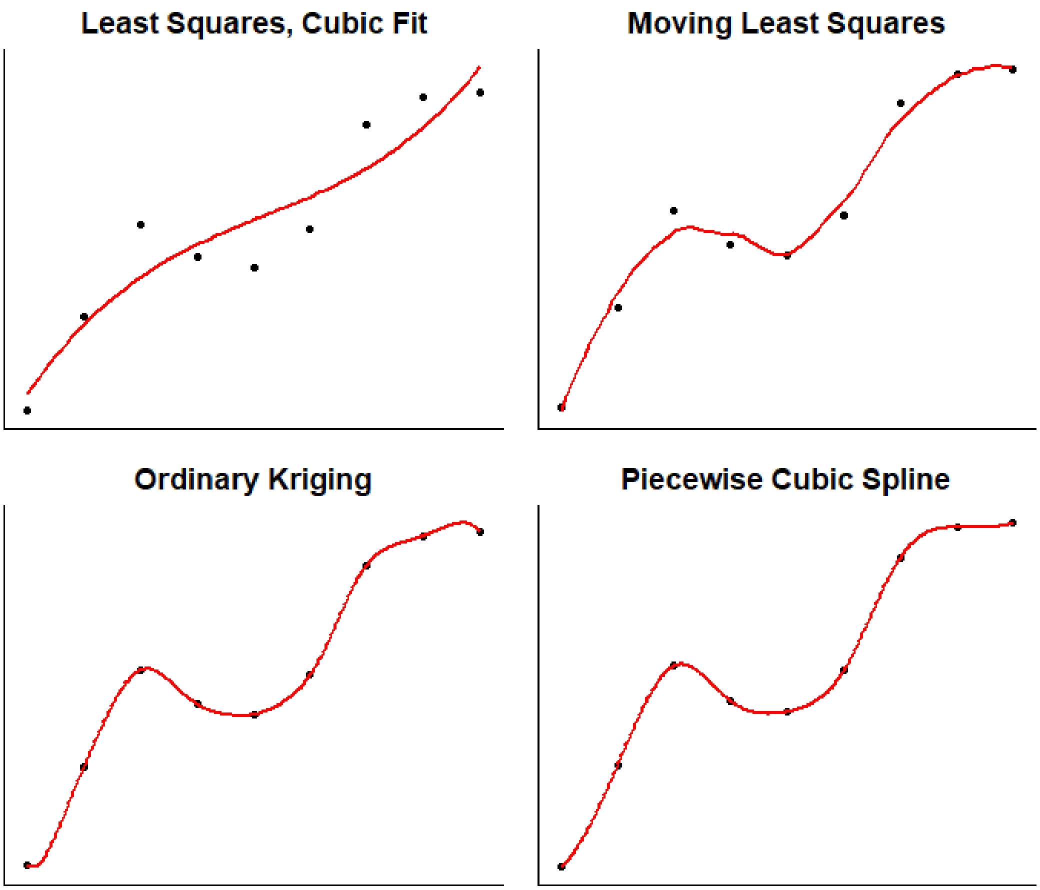

Figure 2: prediction Methods

Prediction Methods

In this section, I characterize some common prediction methods. In particular, I consider methods where we only observe a finite number of values within the two-dimensional plane and we want to estimate values within some compact set containing those points. Figure 2 provides four examples of univariate prediction methods applied to a dataset of nine points generated as follows:

$$y_i = 0.14x_i^5 - 8(x_i - 3)^4 - 2(x_i + 2.5)^2 + \varepsilon(x_i)$$

where $\varepsilon(x) = \varepsilon_i \sim N(0, 40)$ for $i = 1, \ldots, 9$. In the subsections below, I will discuss least squares, moving least squares, and Kriging. The prediction method discussed in this paper, Complex Variable Moving Kriging Interpolation, is most similar to the ordinary Kriging method. In Figure 2 I also included a plot of a piecewise cubic spline as an alternative interpolation method to Kriging. Splines are often visually more appealing, but lack some of the nice statistical properties of Kriging. In basic terms, splines are smooth but Kriging better reflects real processes, producing better predictions.

We assume a data-generating process (DGP) where the inputs are in the two-dimensional plane (x, y) and the output is real-valued $u(x, y)$. Moreover, suppose that we only observe n datapoints $u_1 = u(z_1), \ldots, u_n = u_n(z_n)$. For ease of notation, we make use of the following shorthand: $z = (x, y)$. The DGP is given by

$$u(z) = \sum_{j=1}^{m} a_j p_j(z) + \varepsilon(z) = p^{\top}(z)a + \varepsilon(z)$$

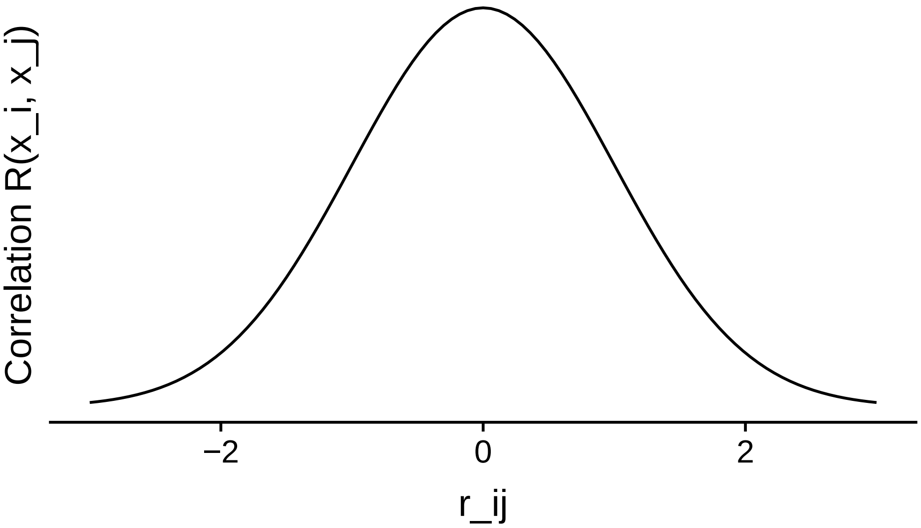

Figure 3: Correlation function over distance between points

That is, we assume u is linear in p_j where p_j are known functions of z, usually taken to be a basis of monomials. We aim to estimate the coefficients a_j.

The use of $z = (x, y)$ hints at the CVKMI approach that is the focus of this paper. But for now we can just consider z as a vector in $\mathbb{R}^2$.

The error terms $\varepsilon(z)$ describe the difference between the linear part of the DGP and any unexplained non-linear departures. We assume the following autocorrelative structure between errors:

$$u(z) = \begin{cases} \mathbb{E}[\varepsilon(z) \mid z] = 0 \\ \text{Var}(\varepsilon(z) \mid z) = \sigma^2 \\ \text{Cov}(\varepsilon(z_i), \varepsilon(z_j)) = \sigma^2 \mathcal{R}(x_i, x_j) \end{cases}$$

The first assumption states that the errors are centered around zero. The second assumption states that the errors are homoskedastic with respect to z. That is, the errors have common variance σ^2. The third assumption states that two points anywhere in the plane covary with each other by a pre-described function $\mathcal{R}$ scaled by the common variance. The function $\mathcal{R}$ is assumed to be known and given by

$$\mathcal{R} : \mathbb{R}^d \times \mathbb{R}^d \to [-1, 1] \qquad \mathcal{R}(x_i, x_j) = e^{-\theta r_{ij}^2}$$

where $r_{ij} = ||x_i - x_j||_2$ is the Euclidean norm and $\theta > 0$ is the Kurtosis parameter. In other words, smaller values of θ result in heavier tailed distributions. So smaller values of θ result in higher levels of correlation between farther away points. Large values of θ result in lower levels of correlation between farther away points. In Figure 3, I plot $\mathcal{R}$ with $\theta = 1$ against r_{ij}. Note that $\mathcal{R}$ is symmetric in its arguments. In essence, we are assuming Gaussian spatial correlation.

For simplicity, in all of the following prediction methods, we assume that $p^\top = \{p_0, \ldots, p_m\}$ are known monomial basis functions. For example, a quadratic monomial basis in one dimension is given by

$$p^\top(x) = [1, x, x^2] \qquad m = 3$$

And a quadratic monomial basis in two dimensions is given by

$$p^\top(x, y) = [1, x, y, xy, x^2, y^2] \qquad m = 6$$

Ordinary Least Squares (OLS)

I start with the simplest, and arguably most powerful, prediction technique—Ordinary Least Squares. Ordinary Least Squares prediction, often called linear regression, provide the "best" linear fit of the data. In this case, the fit will be linear in the monomial basis functions. Define the following $n \times m$ matrix that provides the value of the basis functions at each of the n observed points:

$$P = \begin{bmatrix} p_1(z_1) & \cdots & p_m(z_1) \\ p_1(z_2) & \cdots & p_m(z_2) \\ \vdots & \ddots & \vdots \\ p_1(z_n) & \cdots & p_m(z_n) \end{bmatrix}$$

This definition corresponds to the $n \times 1$ vector that provides the value of the basis functions at the unobserved points:

$$p^\top(z) = [p_1(z), p_2(z), \dots, p_m(z)]$$

Next, define the $n \times 1$ vector of observed values of u as

$$U^\top = [u_1, \dots, u_n]$$

The goal is to find $\hat{a} = [\hat{a}_1, \dots, \hat{a}_m]^\top$ such that we minimize the mean-square-error betwen the observed and predicted values.

$$\hat{a} = \arg\min_{a \in \mathbb{R}^m} MSE(u_i) = \arg\min_{a \in \mathbb{R}^m} \sum_{i=1}^{n} (u_i - p^\top(z_i)a)^2 = \arg\min_{a \in \mathbb{R}^m} ||U - Pa||_2^2$$

This is where the term "least squares" comes in since we are *minimizing* the sum of the *squared* residuals. The least-squares loss function is also used in the estimation of the other predictive methods. The solution to this problem is given by

$$\hat{a} = (P^\top P)^{-1} P^\top U$$

where we have assumed the invertibility of $P^\top P$. So the estimator is given by

$$\hat{u}(z) = p^\top(z)\hat{a}$$

Building off this OLS approach, the following methods use modified versions of this estimator.

Weighted Least Squares (WLS)

Weighted least squares is nearly the same as OLS except the errors are weighted in accordance to how far the datapoints are from the mean. The weighted least-squares approach improves upon ordinary least squares by placing more "weight" on values we believe more (e.g. values near the mean). This technique is useful if there are outlier values that we don't want to effect the fit. The minimization is thus

$$\hat{a} = \arg\min_{a \in \mathbb{R}^m} \sum_{i=1}^{n} w(d_i)(u_i - p^\top(z_i)a)^2$$

where $d_i = ||\bar{z} - z_i||$ and w is a pre-determined weight function. Here $\bar{z}$ does not represent the conjugate of z (remember we are still considering $z \in \mathbb{R}^2$), but rather the mean value of z_i for $i = 1, \dots, n$

$$\bar{z} = \frac{1}{n} \sum_{i=1}^{n} z_i$$

A typical choice of the weight function is Gaussian with a smoothing parameter $\theta > 0$:

$$w(d) = e^{-\theta d^2}$$

The solution to the minimization problem is given by

$$\hat{a} = (P^\top W P)^{-1} P^\top W U$$

where P and U were defined previously, and W is a diagonal $n \times n$ matrix of the weights at the data points

$$W = \begin{bmatrix} w(d_1) & 0 & \cdots & 0 \\ 0 & w(d_2) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & w(d_n) \end{bmatrix}$$

In the case of heteroskedastic errors with known variance $\text{Var}(\epsilon_i) = \sigma_i^2$, the weight matrix can be given by

$$W = \begin{bmatrix} \frac{1}{\sigma_1^2} & 0 & \cdots & 0 \\ 0 & \frac{1}{\sigma_2^2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \frac{1}{\sigma_n^2} \end{bmatrix}$$

where errors are assumed to be independent, but not necessarily identically distributed. In this situation, we give more weight to values with lower variance since we "believe" the values at those points more.

Moving Least Squares (MLS)

Moving least squares is an extension of weighted least squares where instead of considering the distance between each observed data point and the mean data point, we consider the distance between each observed datapoint and *every other* data point. In other words, the weight function in WLS takes on only finitely many values while the weight function in MLS is continuous, and takes on a potentially unique value for each $z \in \Omega$ where Ω is the region of interest.

$$w^{WLS} \in \{w(||\bar{z} - z_1||), \ldots, w(||\bar{z} - z_n||)\} \qquad w^{MLS} \in \{w(||z - z_1||), \ldots, w(||z - z_n||) : z \in \mathbb{R}^2\}$$

For each $z \in \mathbb{R}^2$, the minmization problem is given as

$$\hat{a}(z) = \arg\min_{a \in \mathbb{R}^m} \sum_{i=1}^{n} w(||z - z_i||)(u_i - p^\top(z_i)a)^2$$

The same weight functions w are considered as in WLS. Note that now the estimated coefficients $\hat{a}$ are a *function* of z rather than uniform across z as in OLS and WLS. The solution to the least squares minimization problem is given by

$$\hat{a}(z) = (P^\top W(z) P)^{-1} P^\top W(z) U$$

where P and U were defined previously, and $W(z)$ is a diagonal $n \times n$ matrix of the weights at the data points

$$W(z) = \begin{bmatrix} w(||z - z_1||) & 0 & \cdots & 0 \\ 0 & w(||z - z_2||) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & w(||z - z_n||) \end{bmatrix}$$

In this regard, MLS can be considered a localized WLS prediction of u. The method of Moving Least Squares produces a much better fit, with the drawback that this process ends up being more coputationally intense and statistically obtuse. For each $\hat{u}(z)$ we want to predict with Moving Least Squares, we must compute a different $W(z)$ and invert the $P^\top W(z) P$ matrix, or solve the corresponding system of linear equations for $\hat{a}(z)$.

The method of moving least squares is very flexible depending on which weight function is chosen. For the Gaussian weight function, MLS performs an estimation not unlike that seen in Figure 1. Weight functions with compact support may also be chosen to ease computation or better fit the correlation structure. For most weight functions, MLS still estimates–rather than interpolates–through datapoints. However, If particular weight functions are selected selected, then MLS very nearly interpolates through the data points. In particular, consider the following weight function:

$$w(d) = \frac{1}{d^2 + \delta}$$

As $\delta \to 0$, the MLS prediction scheme interpolates through the data points. Note that when $\delta \approx 0$, $w(d) \to \infty$ when $d \to 0$. So when minimization is performed the residuals at the data points will be forced to 0 since otherwise the weighted mean square error we are trying to minimize would explode to infinity. Explosions to infinity are bad in minimization problems.

Kriging Interpolation (KI)

We now depart from a basic least-squares minimization scheme to Kriging. The method is named after Danie Gerhardus Krige, a South African statistician and mining engineer who developed this technique in the 1950s. His aim was to predict the average grade of gold in a mining block by "nodes" of core-sample assays distributed across the mining block. Prior to Krige, the prediction approach used by mining engineers was coarse. The mining engineers simply took the sample mean of gold grade across the various nodes, and treated the sample mean as the predictor of gold grade. The old approach is analogous to the ordinary least squares approach to prediction, where averages dominate and localized peculiarities are ignored. The Kriging approach improves upon the least-squares approach by taking advantage of the spatially-correlated structure of the datapoints.

South Africa was a natural location for the Kriging interpolation method to develop. Then a British colony, South Africa was—and remains to this day—one of the largest gold and diamond exporters. Mining is a luctrative industry, and better predictive techniques can heavily reduce costs by reducing the number of core samples taken and improving mine placement. The technique of Kriging was refined heavily by Georges Matheron, a French mathematician, in the 1960s, and further development of the technique led to its use in other fields such as metereology and forestry. For more information on the history of Kriging, see "The origins of kriging" published in the journal Mathematical Geology in 1990.

In practice, especially in 3D modelling, Kriging has greater predictive accuracy than weighted least squares, which in turn has greater predictive accuracy than OLS. However, Kriging often imposes more structure on the stochastic process underlying the observed data.

I now cover some of the mathematical properties of Kriging (Ordinary Kriging, technically). We assume that $u(z_1), \dots, u(z_n)$ are realizations of a stochastic process at known locations $z_1, \dots, z_n$. We want a predictor of the form

$$\sum_{i=1}^{n} a_i u(z_i) \qquad \text{such that} \qquad \sum_{i=1}^{n} a_i = 1$$

so that our estimates $\hat{a}_i$ minimize the mean-squared prediction error. We define the following $n \times n$ symmetric matrix with 1 on the diagonal elements.

$$R = \begin{bmatrix} 1 & \mathcal{R}(z_1, z_2) & \cdots & \mathcal{R}(z_1, z_n) \\ \mathcal{R}(z_2, z_1) & 1 & \cdots & \mathcal{R}(z_2, z_n) \\ \vdots & \ddots & \cdots & \vdots \\ \mathcal{R}(z_n, z_1) & \mathcal{R}(z_n, z_2) & \cdots & 1 \end{bmatrix}$$

where $\mathcal{R}$ was the correlation function defined at the beginning of this section. This matrix gives the correlation between all of the two observed points. Note that R has ones on the diagonal because each datapoint will be perfectly correlated with itself.

Similarly, we define the $1 \times n$ vector of correlation between any arbitrary point z and our data points:

$$r(z) = [\mathcal{R}(z, z_1), \mathcal{R}(z, z_2), \ldots, \mathcal{R}(z, z_n)]^\top$$

Then the solution to the minimization problem leads to the following optimal predictor for $z \in \Omega$:

$$\hat{u}(z) = r^\top(z) R^{-1} U + (1 - r^\top(z) R^{-1} 1_n)(1_n^\top R^{-1} 1_n)^{-1} 1_n^\top R^{-1} U$$

Where 1_n is an $n \times 1$ vector of ones. For more mathetmatical details and intuition on this formula, see Cressie (1990).

The biggest difference between Kriging and the previous methods is that its shape function interpolates through the datapoints rather than merely estimating the datapoints. I delve further into this interpolation property when discussing the Kronecker delta property in the CVKMI subsection. Compared to moving-least squares (MLS) or weighted-least-squares (WLS), however, the Kriging method is computationally more intense.

Moving Kriging Interpolation (MKI)

Moving Kriging interpolation is to Kriging interpolation as moving least squares is to weighted least squares. Instead of fitting the Kriging model over the entire domain Ω at once, estimate the Kriging model by fitting the model on $\Omega_z \subseteq \Omega$ where Ω_z is a neighborhood around z. That is, Ω_z will only contain some of the n nodes.

Regular MKI makes use of a curvilinear coordinate γ when considering boundary condition problems. For example, if Ω is the domain under consideration and $u(\partial\Omega)$ satisfies some condition, then $u(\gamma)$ is the value of u evaluated at γ, a local coordinate on Ω. A common choice of coordinate system is to let γ be the arc-length along $\partial\Omega$ from the evaluation point z_i to an arbitrary bounday point $z \in \partial\Omega$.

Complex Variable Moving Kriging Interpolation (CVMKI)

Instead of considering a curvilinear coordinate γ, we consider the cartesian coordinate system as the complex plane: $z = x + iy$. The result is simplified computations, cleaner formulas, and some more intuition adopted from complex analysis.

We want to find a prediction of $u(z)$, denoted by $\hat{u}$ such that $\hat{u}$ minimizes the mean-square error between the observed points and constrained to the observ This problem can be characterized as solving the equality-constrained optimization problem,

$$\hat{\Phi}(z) = \arg\min_{\Phi} \int (\hat{u}(z) - u(z))^2 dz \text{such that} \quad \Phi(z)P - p(z) = 0$$

The solution to this optimization problem is given by the following complex shape function

$$\hat{\Phi}(z) = p^\top(z) \left(P^H R^{-1} P\right)^{-1} P^H R^{-1} + r^\top(z) R^{-1} [I - P \left(P^H R^{-1} P\right)^{-1} P^H R^{-1}]$$

Making use of the following shorthand,

$$A = \left(P^H R^{-1} P\right)^{-1} P^H R^{-1} \quad \text{and} \quad B = R^{-1}(I - PA)$$

we can simplify this result to $\hat{\Phi}(z) = p^\top(z) A + r^\top(z) B$. Note that

$$\hat{\beta}^{GLS} = \left(P^H R^{-1} P\right)^{-1} P^H R^{-1} U$$

is the generalized-least-squares estimator you'll find in the back of an introductory econometrics textbook. The shape function Φ is superior to previously studied shape functions since it posseses the Kronecker delta property. That is, this shape function interpolates. That is, $\phi_i(z_j) = \delta_{ij}$. This property is important for boundary problems where we want the boundary conditions to hold with strict equality. In other words, local approximators based off the shape function Φ will interpolate through z_j for $j = 1, \ldots, n$. To show this property, let $\phi_i(z_j)$ be the (i,j)-th coordinate of $[\phi_i(z_j)]$, then

$$[\phi_i(z_j)] = PA + RB = PA + I - PA = I$$

The global shape function Φ also posesses the partition of unity property:

$$\sum_{i=1}^{n} \phi_i(z) = 1$$

In this case, the partition of unity property essentially states that any function generated by the selected basis of monomials can be reproduced exactly by the shape function and an appropriate choice of coefficients.

When someone presents you with a real problem, it's impolite to provide a complex answer, so we instead consider the real part of the shape function $\Psi(z) = \Re\Phi(z)$ and implement the following local approximation to $u(z)$:

$$\hat{u}(z) = \Psi(z)U = \sum_{i=1}^{n} \psi_i(z)u(z_i)$$

Since Φ is analytic with respect to z in each dimension, then Φ is analytic itself. Therefore, Ψ must be harmonic in each dimension. On first thought, one might think that taking the real part of the complex shape function would remove the partition of unity and Kronecker delta property, but that is not the case—both properties still hold:

$$\sum_{i=1}^{n} \phi_i(z) = \sum_{i=1}^{n} \Re\phi_i(z) = \Re\sum_{i=1}^{n} \phi_i(z) = \Re\{1 + 0i\} = 1$$

$$\psi_i(z_j) = \Re\phi_i(z_j) = \delta_{ij}$$

Solving a Boundary Problem

We now put these results to practice. Consider the following Dirichlet problem on the unit disk Ω:

$$\text{Find } u : \mathbb{C} \to \mathbb{C} \text{ such that} \quad \begin{cases} \Delta u(z) = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 0 & z \in \Omega \\ u(z) = x = \cos\phi & z \in \partial\Omega \end{cases}$$

Analytical results

The analytic solution to this problem is given by

$$u(re^{i\phi}) = r\cos\phi \qquad r \in [0,1] \quad \text{and} \quad \phi \in [0, 2\pi)$$

The proof of this solution is mostly pain-free. Consider the "guess" function $\tilde{u}(re^{i\phi}) = r\cos\phi$. This function is harmonic on the unit disk since $\tilde{u}$ is the real part of the complex identity function $z \mapsto z$ since

$$z = re^{i\phi} = r\cos\phi + ir\sin\phi$$

Moreover, $\tilde{u}(re^{i\phi}) = \cos\phi$ for $r = 1$ so $\tilde{u}$ also fulfills the boundary condition. By the uniqueness of solutions to the Dirichlet Problem, we know that $u = \tilde{u}$ on the unit disk, so our guess was correct.

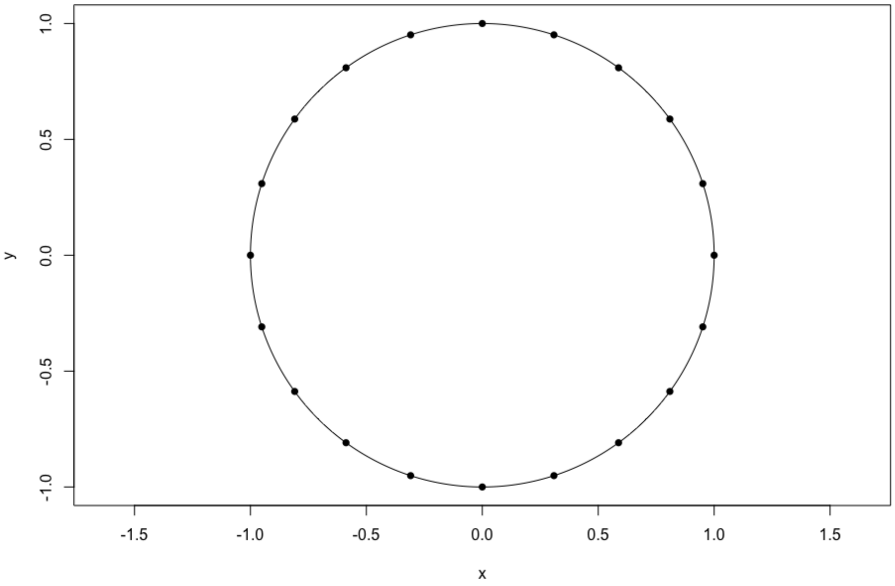

Figure 4: Nodes on the Boundary of Unit Disk

Numerical Results

Now suppose that we only observe finitely many values of u located at equidistant points on the unit circle. Figure 4 shows the disk of radius one centered at the origin with 20 nodes distributed equidistant around the unit circle. That is, we only observe points z_k points where

$$z_k = e^{i\theta_k} \qquad \theta_k = 2\pi - \frac{2\pi}{k} \qquad k = 1, \dots, 20$$

The next two plots observe the performance of CVKMI on the unit disk. Figure 5 only considers $z = x + 0i$ for $x \in [-1, 1]$, while Figure 6 shows the boundary of the unit disk, $|z| = 1$. To show the dependence of CVKMI on the initial parameters, I show the prediction with two different correlation function parameters (θ). The black curves represents the analytic solution of the boundary problem solved in the prior section. The red curves represent the CVKMI solution with $\theta = 1.5$ and the blue curves represent the CVKMI solution with $\theta = 2.5$. For the basis functions, I have selected a quadratic basis in one complex dimension:

$$p^\top(z) = [1, z, z^2]$$

I performed small-scale tests using a larger basis (up to z^7), and predicitve accuracy improved, but not enough to warrant the increase in computation time.

We see that CVKMI predicts rather well for points $z = x + 0i$, especially when the correlation function parameter is set to $\theta = 1.5$. Predictive accuracy declines as θ increases to $\theta = 2.5$. Figure 6 shows that CVKMI fails to adhere to the boundary condition set in the problem, except at the interpolation points. Note the oscillatory nature of the Kriging estimate and how it is highly sensitive to changes in the correlation function parameter θ. In Figure 6, we see that the Kriging estimate performs quite well when $\arg(z) \approx \pi k$ for $k \in \mathbb{Z}$. In other words, the estimate is good when the imaginary part of z is small relative to the real part of z. This is likely an artifact of the shape function used for this estimate being defined as the real part of the complex shape function.

Next I consider the behavior of CVKMI by varying the number of nodes and computing the relative error

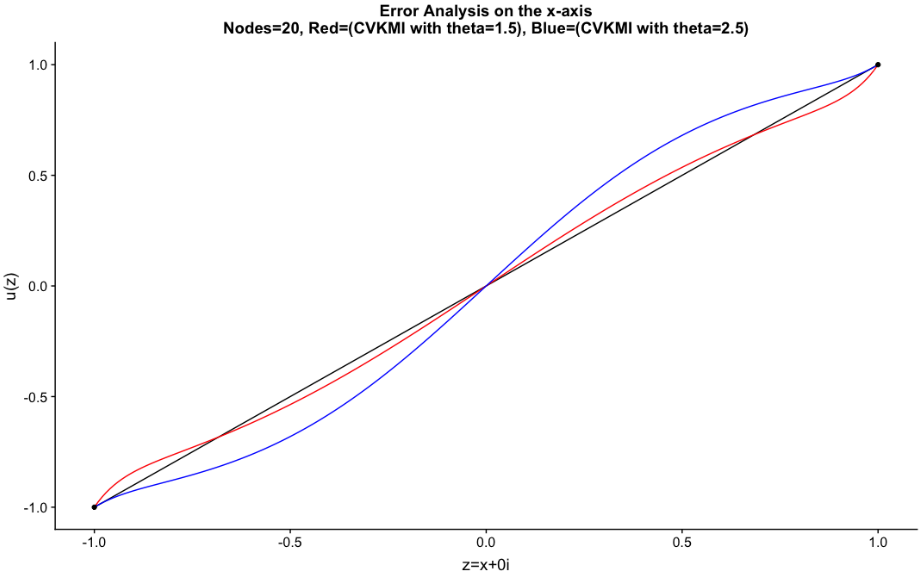

Figure 5: Error Analysis on x-axis

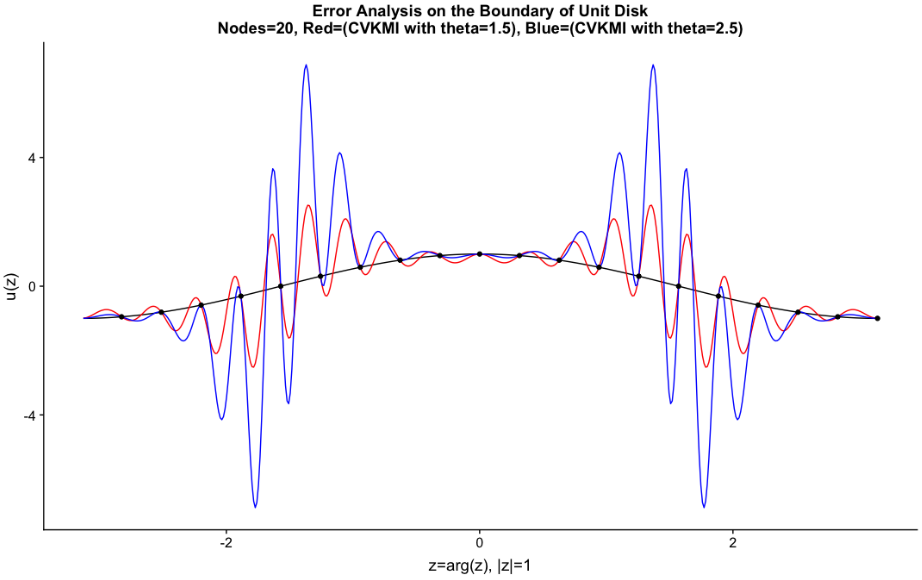

Figure 6: Error Analysis on Boundary

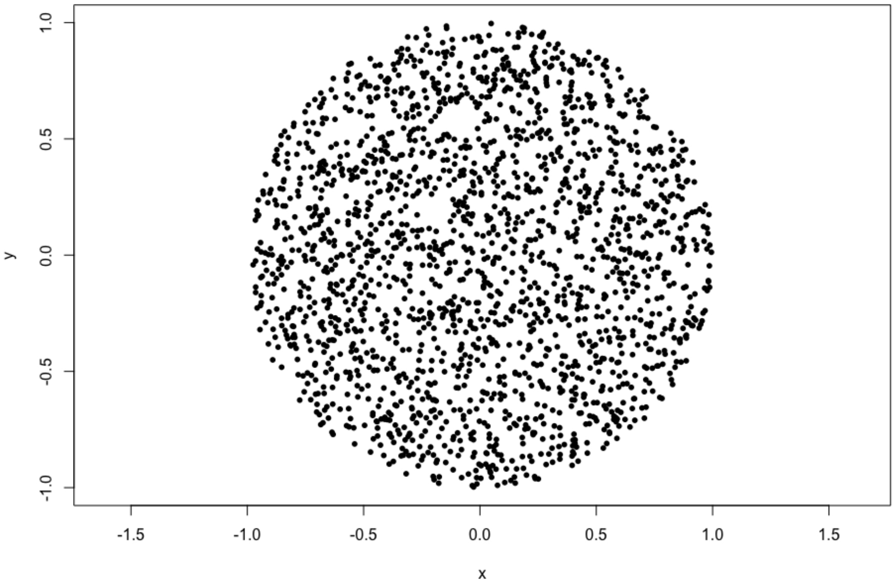

Figure 7: Sampling Points

over a set of $n = 2000$ sampling points in the unit disk. The relative error is defined as

$$e = \sqrt{\sum_{i=1}^{n}(u_i^a - u_i^n)^2 / \sum_{i=1}^{n}(u_i^a)^2}$$

The values u_i^a and u_i^n are the analytical and numerical solutions at the ith sampling point. The sampling points are visualized in Figure 7.

In Figure 8, I report the relative error of the CVKMI predicitons at the sampling points over a grid of θ values. The second plot of Figure 7 is a close-up of the first plot. We see that the relative-error is nearly convex as a function of θ for $n = 20$ nodes with the global minimum occurring somewhere between $\theta = 1$ and $\theta = 1.2$. A similar picture emerges for different values of n.

Next I observe the relative error of the CVKMI prediction method and an increasing number of nodes. For ease of analysis, the nodes remain equidistant on the unit circle for all numbers of nodes. Since the results above show that the relative error is highly dependent on the value of θ, I performed the following process to compute the relative error:

1. Select a number of nodes n, and place these nodes uniformly around the unit circle
2. Create a grid of θ values contained in $[0.01, 5]$
3. For each value of θ in this grid, calculate the relative error over all sampling points
4. Select $\hat{\theta}_n$ that minimizes the relative error for n nodes, and report that relative error.

The results in Figure 9 are surprising. We should expect that the relative error would be monotone decreasing as the number of nodes increases. We see that the relative error decreases on average until the number of nodes equals seventeen, then the relative error skyrockets. The increase in relative error after $n = 17$ nodes can be explained by a number of factors:

1. When I increased the number of nodes on the unit circle, I did not increase the number of sub-boundaries over which the CVMKI interpolation is locallized. Tu and his coauthors restrict the number of nodes per sub-boundary by splitting up $\partial\Omega$ into sub-boundaries. That is, $\partial\Omega = \Gamma_1 + \cdots + \Gamma_d$.

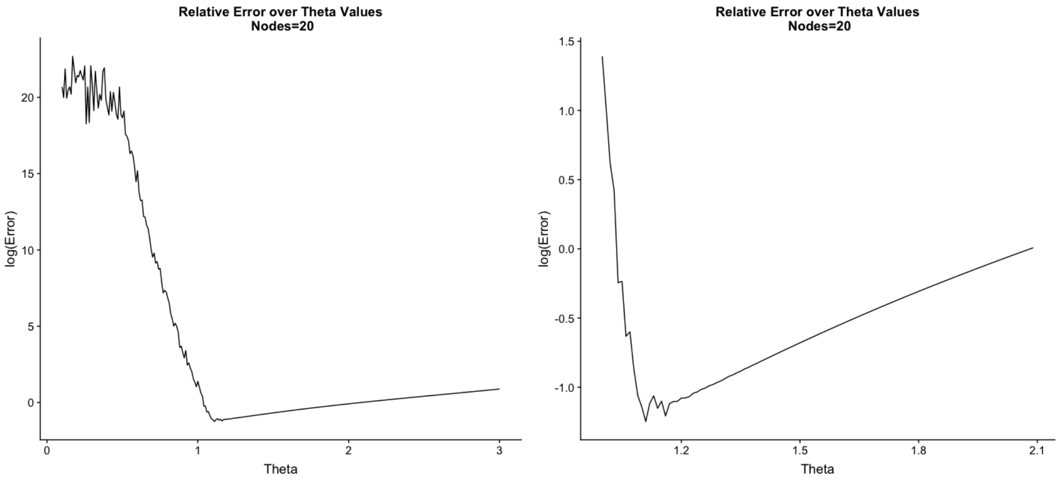

Figure 8: Relative Error over theta

2. I used the same grid of θ values to search for a global minimum of the relative error, regardless of the number of nodes. A more refined optimization strategy such as steepest-descent or Newton's method would likely produce better results when there is a greater number of nodes.
3. The correlation function under consideration does not have compact support. Perhaps if a correlation function with compact support (with a support that contains fewer nodes) would provide more accurate predictions.
4. A greater number of nodes could introduce more floating point errors in the various calculations, but I find this to be the least likely explanation.

Because errors of the predictor become more extreme as the imaginary part of z grows away from 0, I believe the first explanation is most likely. If I were to instead implement CVKMI and locallize on the sub-boundaries, and then center each sub-boundary to the origin, then the imaginary part of the locallized z would all be small. I believe this modification would considerably improve the predictive accuracy of the CVKMI estimator.

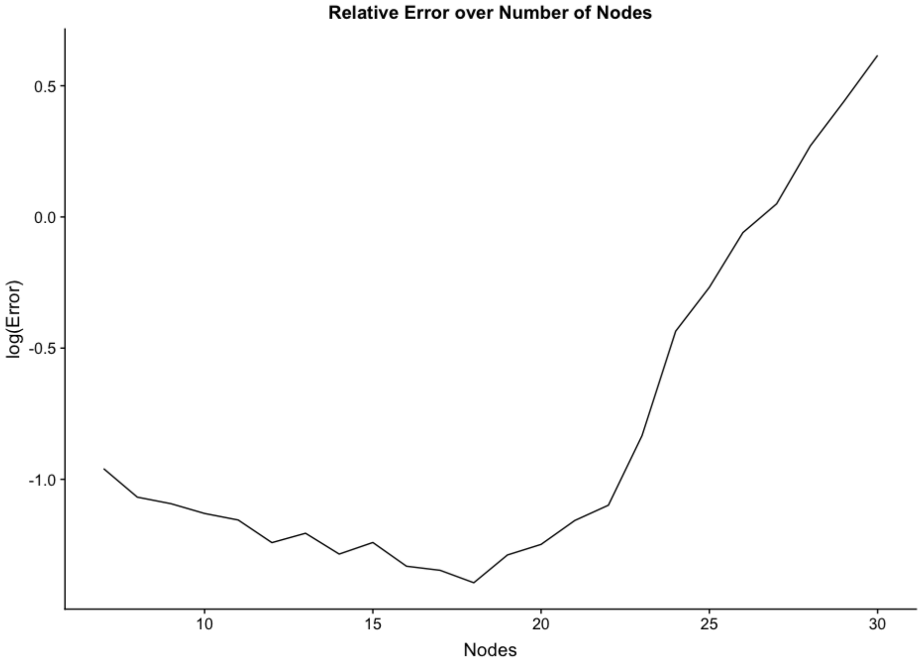

Figure 9: Relative Error over number of nodes

References

Belytschko, T., Y. Y. Lu, and L. Gu. 1994. "Element-free Galerkin methods." *International Journal for Numerical Methods in Engineering* 37 (2): 229–56. https://doi.org/10.1002/nme.1620370205.

Cressie, Noel. 1990. "The origins of kriging." *Mathematical Geology* 22 (3): 239–52. https://doi.org/10.1007/BF00889887.

Mukherjee, Yu Xie, and Subrata Mukherjee. 1997. "The Boundary Node Method for Potential Problems." *International Journal for Numerical Methods in Engineering.* https://doi.org/10.1002/(SICI)1097-0207(19970315)40:5<797::AID-NME89>3.0.CO;2-.

Nealen, Andrew. n.d. "An As-Short-As-Possible Introduction to the Least Squares, Weighted Least Squares and Moving Least Squares Methods for Scattered Data Approximation and Interpolation." http://math.nist.gov/tnt/.

Tu, Sanshan, Leilei Dong, Hongqi Yang, and Yi Huang. 2016. "Complex variable moving Kriging interpolation for boundary meshless method." *Engineering Analysis with Boundary Elements* 65 (April): 72–78. https://doi.org/10.1016/J.ENGANABOUND.2016.01.003.