

Determining “Optimal” Shape Parameters with Precision Adapted Radial Basis Function Interpolation

Paul Mayer, Stavros Christofi Ph.D.
Western Connecticut State University, Department of Mathematics

Introduction

In Radial Basis Function (RBF) interpolation, Cross Validation techniques such as Rippa's LOOCV are often used to determine the value of the shape parameter, c .

These Cross Validation techniques only require the points of the interpolant to choose a value of c . Care must be taken, however, to ensure that any value of c considered will provide a stable interpolant. The range of acceptable values of c varies based on the type of basis function used, the number of points, their spacing, and the machine precision used in calculation.

If greater machine precision is applied to interpolation, then the range of values of c that may potentially provide good results is larger. The value of c ultimately chosen will depend on the cost function used. We propose a novel method to determine what range of c values should be examined when searching for a suitable c . This method considers the interpolation points being used as well as the available machine precision. The method works for both uniformly or non-uniformly spaced points.

Background

A Radial Basis Function (RBF) is a function of the form:

$$\phi(r, c) : \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}$$

The following are a few examples of RBFs.

| Function | Name |
|--|-----------------------|
| $\phi(r) = e^{-c^2 r^2}$ | Gaussian |
| $\phi(r) = \sqrt{c^2 + r^2}$ | multi-quadric |
| $\phi(r) = \frac{1}{\sqrt{c^2 + r^2}}$ | Inverse multi-quadric |

Typically, r is the vector norm. These RBFs can be the basis for an interpolant:

$$I_{[X,Y,c]}(\vec{x}) = a_1 \phi(|\vec{x} - \vec{x}_1|, c) + \dots + a_n \phi(|\vec{x} - \vec{x}_n|, c)$$

The $\mathbf{X} = \{\vec{x}_1 \dots \vec{x}_n\}$ are arbitrary interpolation points and $\mathbf{Y} = \{y_1 \dots y_n\}$ are the corresponding function values at those points, c is the shape parameter and \vec{x} is the point where the interpolant is being evaluated.

Building RBF Interpolants

Given arbitrary interpolation points $\mathbf{X} = \{\vec{x}_1 \dots \vec{x}_n\}$ with known values

$\mathbf{Y} = \{y_1 \dots y_n\}$, we can determine the coefficients $\vec{a} = \{a_1 \dots a_n\}$ by solving the linear system:

$$M_c \vec{a} = \mathbf{Y}$$

Here M_c is a $n \times n$ Toeplitz matrix constructed from the interpolation points. Here is an example for $n = 5$:

$$M_c = \begin{bmatrix} \phi(|\vec{0}|, c) & \phi(|\vec{x}_1 - \vec{x}_2|, c) & \phi(|\vec{x}_1 - \vec{x}_3|, c) & \phi(|\vec{x}_1 - \vec{x}_4|, c) & \phi(|\vec{x}_1 - \vec{x}_5|, c) \\ \phi(|\vec{x}_2 - \vec{x}_1|, c) & \phi(|\vec{0}|, c) & \phi(|\vec{x}_2 - \vec{x}_3|, c) & \phi(|\vec{x}_2 - \vec{x}_4|, c) & \phi(|\vec{x}_2 - \vec{x}_5|, c) \\ \phi(|\vec{x}_3 - \vec{x}_1|, c) & \phi(|\vec{x}_3 - \vec{x}_2|, c) & \phi(|\vec{0}|, c) & \phi(|\vec{x}_3 - \vec{x}_4|, c) & \phi(|\vec{x}_3 - \vec{x}_5|, c) \\ \phi(|\vec{x}_4 - \vec{x}_1|, c) & \phi(|\vec{x}_4 - \vec{x}_2|, c) & \phi(|\vec{x}_4 - \vec{x}_3|, c) & \phi(|\vec{0}|, c) & \phi(|\vec{x}_4 - \vec{x}_5|, c) \\ \phi(|\vec{x}_5 - \vec{x}_1|, c) & \phi(|\vec{x}_5 - \vec{x}_2|, c) & \phi(|\vec{x}_5 - \vec{x}_3|, c) & \phi(|\vec{x}_5 - \vec{x}_4|, c) & \phi(|\vec{0}|, c) \end{bmatrix}$$

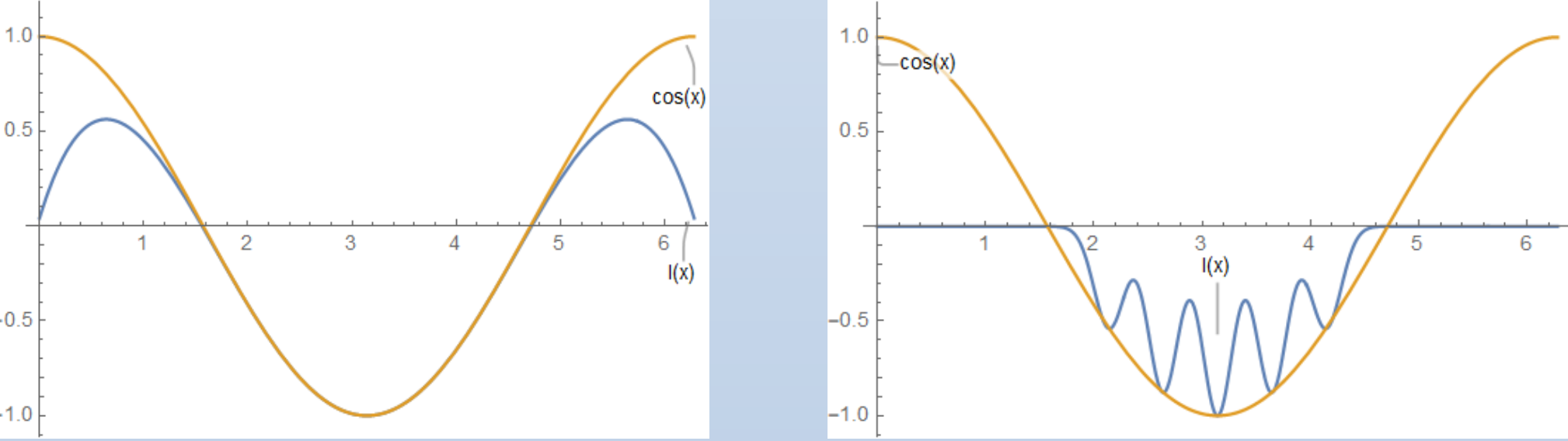
Since ϕ is a function of the squared norm of \vec{x} , the matrix is always positive definite. Inverting this matrix yields the coefficients of the interpolant.

$$\vec{a} = M_c^{-1} \mathbf{Y}$$

Since M_c depends on c , M_c^{-1} and \vec{a} also depend on c . For any given value of c the interpolant built will pass through the designated points, but each interpolant will have a different shape.

Effect of Shape Constant:

As an example, consider the test function $f(x) = \cos(x)$, we built interpolants by sampling this test function at $\{\pi - 1, \pi - \frac{1}{2}, \pi, \pi + \frac{1}{2}, \pi + 1\}$ using $c = .5$ (left) and $c = 5$ (right). Here both interpolants are plotted in blue, and the test function is plotted in yellow:



Both interpolants pass through the sampled points, but are very different functions

Shape Constant Determination

While all values of c result in valid interpolants that pass through the interpolation points, it is desirable to have a procedure for selecting c based on some criteria. This is done by defining a cost function $C(X, Y, c)$, which is then minimized. Perhaps, the most popular cost function is the **Leave One Out Cross Validation (LOOCV)** Rippa [1] and Fasshauer [2]:

$$C(X, Y, c) = \sum_{i=1}^n |y_i - I_{[X \setminus \vec{x}_i, Y \setminus y_i, c]}(\vec{x}_i)|$$

The notation $X \setminus \vec{x}_i$ refers to removing the i th point from X . The cost function C effectively returns the sum of the variances created by removing each of the n points from the set of interpolation points in turn and comparing the resulting interpolant's value with the value removed from the interpolation points. Hence **Leave One Out Cross Validation**.

This cost function is computationally expensive since it requires n Toeplitz matrices be inverted each time the cost function is evaluated for a given value of c . However, Rippa [1] devised an optimization to this process, which allows C to be evaluated with only a single matrix inversion.

Given a cost function, C , shape constant determination then becomes a matter of finding a value of c which minimizes C :

$$c_{optimal} = \underset{c>0}{\operatorname{ARGMIN}} C(X, Y, c)$$

This minimization must typically be performed using numerical methods.

Catastrophic Loss Of Precision in Radial Basis Functions

Whenever two finite precision numbers are added or subtracted there is a potential for precision loss. This occurs when the significant digits of one addend cancel the significant digits of another addend leaving fewer significant digits in the result. RBF Interpolants can exhibit this problem even when high precision arithmetic is used. To illustrate this, we took a single set of interpolation points built from the test function $\sin^7(x)$ near π of five points evenly spaced by $\frac{2\pi}{640}$. This is a place where the function is relatively flat. We created three different Gaussian interpolants at $c = 8 \times 10^{-7}$, $c = 8 \times 10^{-6}$, and $c = 8 \times 10^{-5}$. Rather than adding the terms of the interpolant together we evaluate and display the terms below:

| Shape Parameter | Terms of Interpolant | | | | |
|------------------------|--------------------------------------|--------------------------------------|--------------------------------------|--------------------------------------|--------------------------------------|
| | $a_0 \phi(\vec{x} - \vec{x}_0 , c)$ | $a_1 \phi(\vec{x} - \vec{x}_1 , c)$ | $a_2 \phi(\vec{x} - \vec{x}_2 , c)$ | $a_3 \phi(\vec{x} - \vec{x}_3 , c)$ | $a_4 \phi(\vec{x} - \vec{x}_4 , c)$ |
| $c = 8 \times 10^{-7}$ | 9.61×10^{54} | -3.84×10^{55} | 5.77×10^{55} | -3.84×10^{55} | 9.61×10^{54} |
| $c = 8 \times 10^{-6}$ | 9.61×10^{46} | -3.84×10^{47} | 5.77×10^{47} | -3.84×10^{47} | 9.61×10^{46} |
| $c = 8 \times 10^{-5}$ | 9.61×10^{38} | -3.84×10^{39} | 5.77×10^{39} | -3.84×10^{39} | 9.61×10^{38} |

In the neighborhood of π , this interpolant ought to have a value close to zero. The terms of the interpolant have wildly varying magnitudes, which must cancel to produce a value near zero. When $c = 8 \times 10^{-7}$ this stencil will require at least 54 digits of precision. This demonstrates that some stencils and values of c require more digits of precision. All values of c are valid, but not all values are available for all stencils if precision is limited. When searching for a minimum of the cost function, there must be a way to determine what values of c will work for the available precision.

Determination of c_{min} using Matrix Condition Number

To minimize the cost function C numerically, it is necessary to determine the range of values $[c_{min}, c_{max}]$ where catastrophic precision loss will not occur.

Rippa [1] suggests that the matrix condition number must satisfy:

$$\kappa(M_c) = \frac{\lambda_{max}(A)}{\lambda_{min}(A)} < \frac{1}{10d}$$

Where d is the number of decimal digits of precision being used. This might account for the ill-conditioning caused by the coefficients, a_i , but does not account for the contribution of the basis functions themselves.

Mongillo [3] suggests that the matrix condition number for Gaussian basis functions must satisfy:

$$\kappa(M_c) \leq \frac{2cnq_x \cdot e^{40.71/(qn)c^2}}{\sqrt{\pi}}$$

Where n is the number of points and q_x is the minimum spacing. However, Mongillo also states that the resulting bound is, in fact, not practical.

Our Method: Precision Loss Metric

We assume that all calculations are carried out to a precision that is accurate to d decimal digits. The precision loss is defined as:

$$PL[w_1, w_2] = f(x) = \begin{cases} 0, & w_1 = 0 \text{ or } (w_1 + w_2) = 0 \\ \left\lceil \log_{10} \left(\left| \frac{w_1}{w_1 + w_2} \right| \right) \right\rceil, & \text{else} \end{cases}$$

Note that this operation is not associative in the sense that $PL[w_1, w_2 + w_3] \neq PL[w_1 + w_2, w_3]$. Assuming we have an ordered list of terms $W = \{w_1, w_2, \dots, w_n\}$, we can apply this operator to sets of terms in the following manner:

$$PL[W] = \sum_{i=1}^{n-1} \left[PL \left[\sum_{j=1}^i w_j, w_{i+1} \right] \right]$$

This measures the number of digits of precision lost when a set of terms are added. For example:

$$PL[\{1000, -1000.2, .1\}] = 4$$

We can apply this to the terms of an interpolant $I_{[X,Y,c]}(x)$

$$W = \{a_1 \phi(|\vec{x} - \vec{x}_1|, c), \dots, a_n \phi(|\vec{x} - \vec{x}_n|, c)\}$$

If $PL[W] > d/m$ where d is the working precision and m a precision margin, then we can infer that the interpolant is using more than the available working precision for that value of c for that stencil.

Logarithmic Descent

One way to make use of the precision loss metric to guide shape constant determination is *Logarithmic Descent*. The idea is to step logarithmically from a high value of c to successively lower values of c until the interpolant has used up the available precision. For each value of c , the cost function is evaluated and the value of c that minimizes the cost function is used:

```
level=0; done=false; min_cost = Inf; min_c = 0;
For level = 0 to -100 while (not done)
  For step = 10 to 2 while (not done)
    c = (step - 1)10-level,
    <Build interpolant for stencil with shape constant, c>
    <Evaluate terms of interpolant and precision loss>
    If (precision loss > working precision(d)/margin(m))
      done=true
    else
      If (Cost(Interpolant) < min_cost)
        min_cost=cost;
        min_c=c;
      end
    end
  end
end
```

This method of search eliminates the need to set a minimum value of c apriori. Instead, the search cuts off when the precision is exhausted.

Application to ENO Type Interpolation

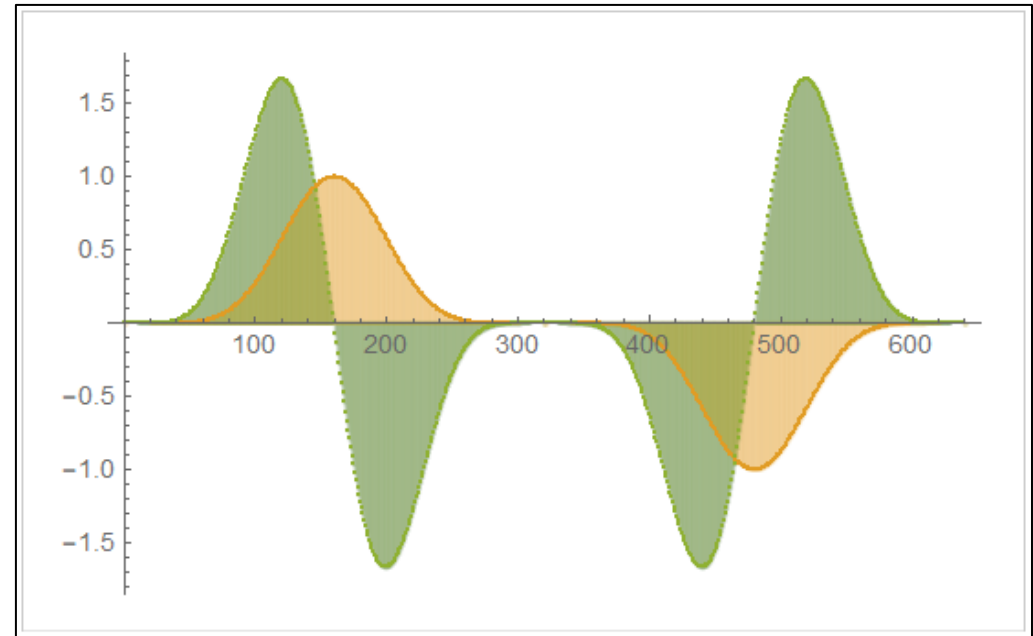
To evaluate this method of shape constant determination, we applied it to Essentially Non-Oscillatory (ENO) type interpolation. In this application, the first derivative of a data set must be reconstructed, cell by cell, by creating a stencil of points for each cell in a way that avoids discontinuities. See Shu [4]. RBF Stability is critical in this application. Moreover, we reconstructed a variety of test functions. For each cell, we determined the optimal value of c based on the interpolation points of that cell's ENO stencil. Each test function was reconstructed at multiple resolutions $\Delta x = \{\frac{2\pi}{40}, \frac{2\pi}{80}, \frac{2\pi}{160}, \frac{2\pi}{320}, \frac{2\pi}{640}\}$. The L_1 error of the first derivative was measured at the right cell boundary for the $\frac{2\pi}{640}$ resolution and compared to Newton Form Polynomial Interpolation and Trigonometric Polynomial Interpolation (Christofi [5]).

| Test Function | L_1 Error | | |
|----------------------------------|-------------|---------------|----------------------------------|
| | Polynomial | Trigonometric | Gaussian RBF cs w/ New Method |
| $\sin^7(x)$ | 1.06E-08 | 9.05E-09 | 7.22E-09 |
| $\sin(x^2)$ | 1.49E-06 | 1.42E-06 | 1.29E-06 |
| $x \cdot \sin(4x)$ | 1.01E-07 | 7.26E-08 | 2.78E-08 |
| $(1 - .9\cos(x))^{-\frac{1}{2}}$ | 1.47E-08 | 1.41E-08 | 1.38E-08 |

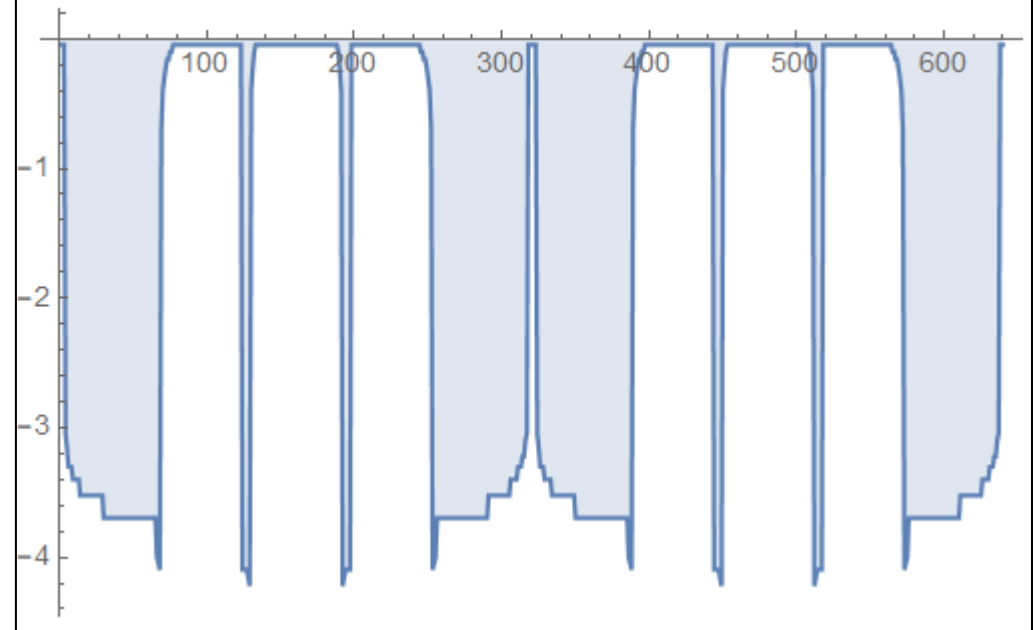
In addition to the benefit of improved L_1 accuracy, our method does not require the use of a c_{min} value.

Visualizing Precision Loss

To visualize this process, we plot a test function $y = \sin^7(x)$, and the value of the shape constant for each cell of a 640-cell ENO reconstruction as well as the precision loss of the interpolant for each such cell.

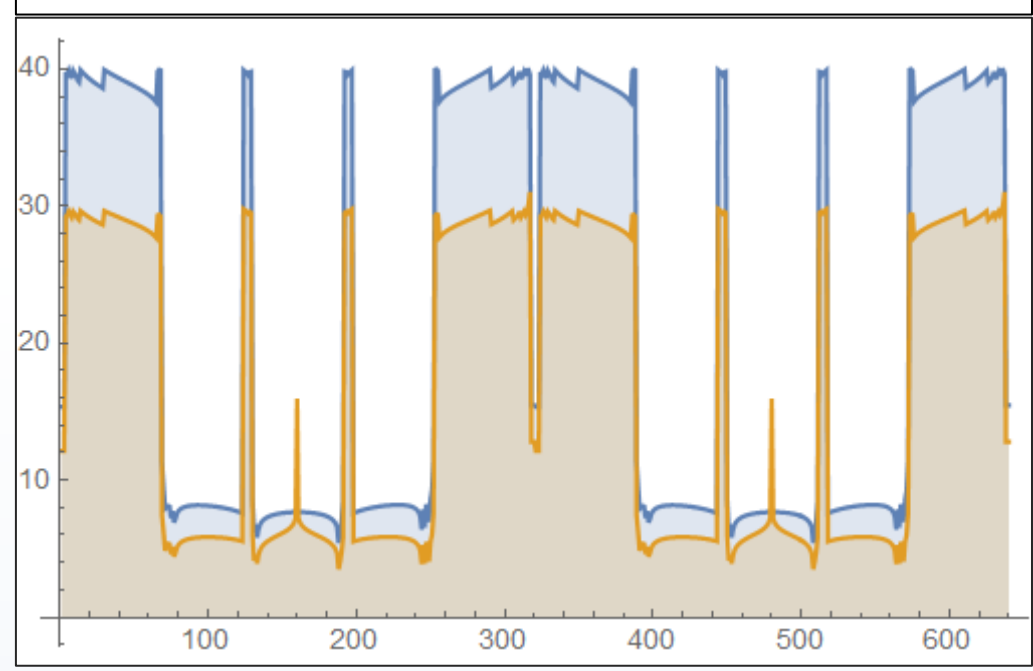


Test function in yellow and its approximated derivative in green.



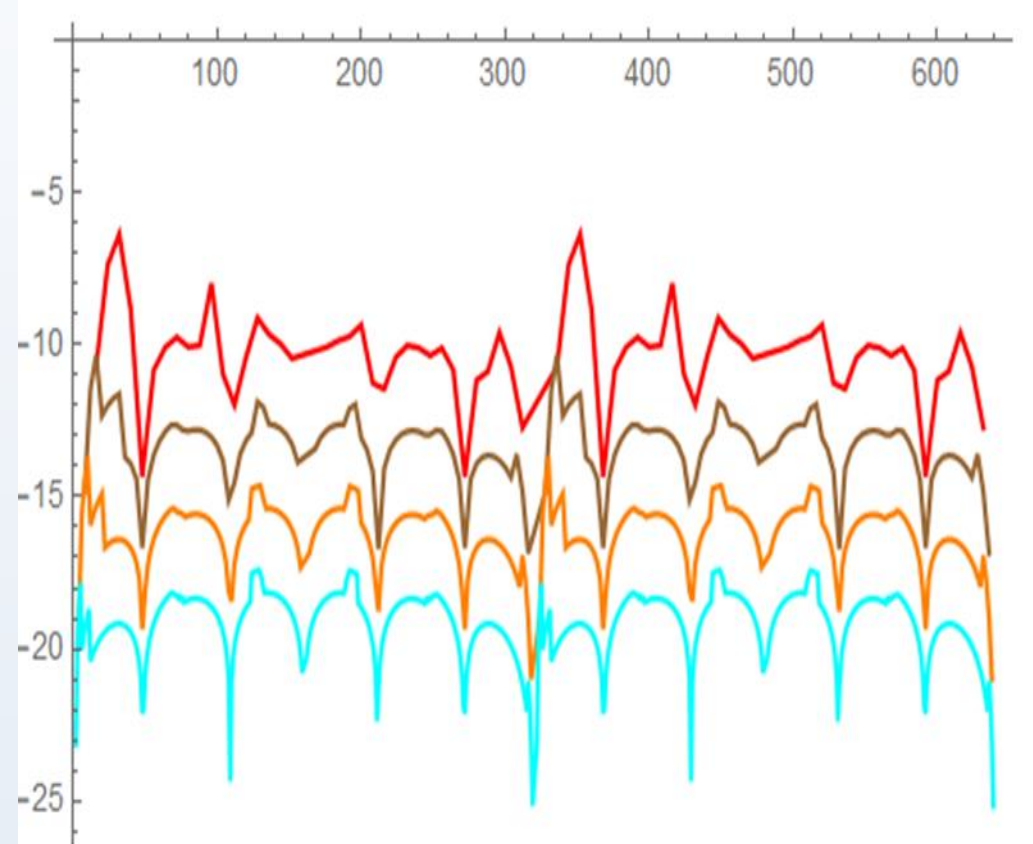
Determined Shape Parameter

The log of the shape parameter chosen for each of the 640 stencils is displayed. LOOCV was used in conjunction with logarithmic descent to choose c . Without the precision loss metric, an artificial lower bound would be needed



Precision Loss

This plot shows the precision loss of the interpolant at the chosen value of c . Note that only some cells require high precision. The loss of the interpolant is given in blue while the loss of the first derivative of the interpolant is given in yellow.

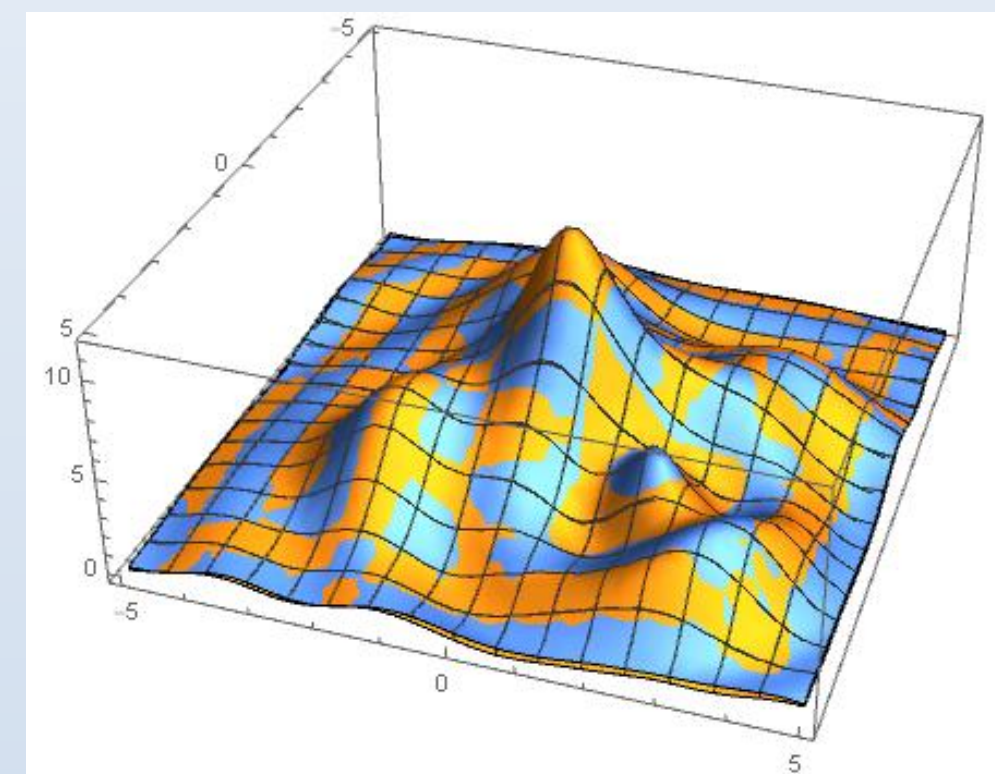


Order of Accuracy

This plot shows the $\ln L_1$ Error for each stencil reconstructed at each resolution. For $n = 5$ points in the stencil of each cell, RBF interpolation maintains consistent and correct order 4 accuracy at all resolutions.

Scattered Surface Interpolation e.g.

We also used RBF Interpolation to interpolate a test function in $\mathbb{R}^2 \rightarrow \mathbb{R}$.



$n = 169, c = .90$

We interpolated using $n = 49$ and $n = 169$ points. In both cases the precision loss metric method was used to limit the values of c considered. For $n = 49$ logarithmic descent was cut off at $c = .20$ and the value chosen was $c = .70$. For $n = 169$ points the cut off was $c = .70$ and the value of $c = .90$ was chosen.

Conclusion and Remarks

We propose and test a precision loss metric to dynamically determine the optimal value of c in RBF interpolants and apply it in the difficult setting of ENO type interpolation. Further work is needed to develop a full ENO-type scheme for numerical solution of hyperbolic conservation laws that exhibit discontinuities.

References & Acknowledgements

- [1] S Rippa, An algorithm for selecting a good value for the parameter c in radial basis function interpolation, Advances in Computational Mathematics 11, 1999.
- [2] G. Fasshauer, J. G. Zhang, On Choosing “Optimal” Shape Parameters for RBF Approximation, Numerical Algorithms 45, 345–368, 2007.
- [3] M Mongillo, Choosing a Basis Functions and Shape Parameters for Radial Basis Function Methods, SIAM, November 25th 2011
- [4] C.-W. Shu, High order weighted essentially nonoscillatory schemes for convection dominated problems, SIAM Review, v51 (2009), pp.82-126.
- [5] S. Christofi, PhD. Thesis, Building Blocks for ENO schemes, Brown University. 1996.