Abstract

This paper presents a method of producing higher order discretization weights for linear differential and integral operators using the Minimum Sobolev Norm idea [1][2] in arbitrary geometry and grid configurations. The weight computation involves solving a severely ill-conditioned weighted least-squares system. A method of solving this system to very high-accuracy is also presented, based on the theory of Vavasis et al [3]. An end-to-end planar partial differential equation solver is developed based on the described method and results are presented. Results presented include the solution error, discretization error, condition number and time taken to solve several classes of equations on various geometries. These results are then compared with those obtained using the Matlab’s FEM based PDE Solver as well as Dealii[4].

1 Keywords

PDE Solver, Numerical Discretization , Sobolev Norm, Higher Order, Finite Difference , weighted least squares , parallelism , Python
2 Introduction

This paper introduces a higher order finite difference scheme for solving two dimensional elliptic PDEs based on Minimum Sobolev Norm (MSN) interpolation\cite{1}\cite{2}. The MSN scheme is a higher order interpolation idea, that suppresses Runge Oscillations by minimizing an appropriately set up Sobolev Norm of the interpolant. We extend the idea to setup a weighted least squares problem for weights that approximate linear differential and integral operators. The ill-conditioned system that arises in the process is solved using a Complete Orthogonal Decomposition method. We present a simple, yet effective orthogonal decomposition method that we call CODA. With a method to approximate differential operators in hand, we present a local Finite Difference like method to solve planar elliptic PDEs. The method described in this paper is fairly generic, basis independent and extends to variable coefficient, higher order and higher dimensional problems easily.

The underlying solution is assumed to be discretized at prescribed grid points. At each interior grid point, we consider a local neighborhood of grid points. We then use sample values at these points to approximate the underlying differential operator using Minimum Sobolev Norm weights. We believe that the order of the method depends on the number of neighbors chosen. A sparse banded matrix corresponding to the discretization of the PDE is assembled, and solved. Appropriate balancing is applied to control the condition number of the resulting matrix. The obvious motivation for a higher order method is that it needs a coarser grid to solve the PDE at hand to a given accuracy than a lower order method. We believe that the fineness of discretization required beyond the Nyquist rate to resolve the solution underlying a PDE to a given accuracy reduces as the order of the method increases.

The question of whether a higher order method is faster to get to a given accuracy remains though, as a higher order method may lead to a significant increase in computational complexity. The simplest manifestation of this could be the fact that we have a denser system to solve, since we use more weights to approximate an operator in the vicinity of a point. We assume that the cost of weight computation is amortized by the sparse system solve cost. For regular grids, one could exploit the regularity of the stencils to avoid computations. Also the weight computations are easily parallelizable even in absence of regularity and are reusable for a given geometry/grid. Let the final sparse system to be solved be

\[ A_{N \times N} x = b, \]

where \( A \) is the sparse matrix discretizing the PDE, \( x \) is the solution vector, and \( b \) is the appropriate right hand side. If we use \( L^2 \) weights per row at maximum to approximate some operator through \( A \) then \( A \) is still sufficiently sparse as long as \( L^2 \ll N \). Assuming that we consider appropriate reordering for optimality, we believe that the time taken to solve the system is \( O(L^3 N^{1.5}) \). In the maximal case, we can have an \( N \) point neighborhood, and so each equation would have \( N \) weights, giving us the dense compute time \( O(N^3) \).
We hypothesize that the order of the method is given by $k = \frac{L+1}{2}$ for a regular grid using an $L$ point, centered, square neighborhood. Since the numerical instability in computing the weights increases with $L$, we believe that there is a maximum order beyond which the numerical accuracy begins to diminish. Also, considering the asymptotic cost of solving the final system, we see that, the system becomes more and more dense, and a sparse solver continues to lose its efficiency. Under our assumption above, we can see that the flops required to solve an $N \times N$ sparse system arising out of a $k^{th}$ order method would then be $O((2k-1)^3N^{1.5})$. The accuracy achieved with a $k^{th}$ order method with an $N^{0.5} \times N^{0.5}$ grid is $O(N^{-k/2})$. Therefore, to get to an accuracy of $\epsilon_{req}$,

$$\epsilon_{req} = N^{-k/2} \Rightarrow N = \left(\frac{1}{\epsilon_{req}}\right)^{\frac{2}{k}}.$$ 

The flops needed with a $k^{th}$ order method to get to $\epsilon_{req}$ digits of accuracy would then be

$$O((2k-1)^3(\epsilon_{req})^{-3/k}).$$

From Figure 2, it may be inferred that the computational advantage of the higher order method diminishes beyond an order of about 10.

Figure 1: FLOPS vs Order of method for $\epsilon_{req} = 10^{-16}$

Section 3, briefly introduces the MSN interpolation idea, and sets up the weighted LS problem corresponding to the approximation of linear operators. It then discusses the CODA method to solve the ill-conditioned WLS problem accurately using recursive singular value decompositions. Results indicating the success of CODA are presented. Section 5 discusses the discretization method to setup a local Finite Difference like system, and the related issues. Section 7 presents a host of results obtained using the PDE solver over various problems and geometries. Each of these problems and geometries are solved using varying neighborhood sizes corresponding to varying orders. The associated
sparse matrix bandwidth and sparse solve time are also discussed. Section 8 provides conclusions and possible extensions. The Python and Matlab based PDE Solvers will be made available through our group website [5]. The proof that the local discretization of the PDE converges is presented in Section 6.

Related closely to our work is the work of Wright and Fornberg [6] which describes a compact finite difference formulation using radial basis functions. [6] pays particular attention to the basis function, and interest is exhibited in retrieving standard FD weights. Also, [6] uses Hermite interpolation ideas in computing the weights to improve the discretization accuracy. Furthermore, their method is a “mesh free” method, where in scattered data stencils in the locality of collocation points are selected for accuracy. Our method is basis independent, and provides for the possibility of adaptive gridding and selective neighborhood selection as well. For convenience we use a square window and regular grids on arbitrary geometry to solve PDEs. Comparable Higher Order methods in the FEM paradigm are the Deal II [4] and the Hermes [7] projects.

3 MSN Discretization

The MSN interpolation scheme computes an interpolating polynomial of order higher than the number of interpolating nodes, whose sobolev norm is minimum [1][2]. Traditional lagrange interpolation uses a polynomial of order equal to the number of interpolating nodes. It is known that, except under particular circumstances, such interpolation schemes produce divergent interpolants [8][9]. The MSN scheme produces a polynomial of order much larger than the number of interpolating points. Of course, there are infinitely many such polynomials since the linear system set up to solve for the interpolant would be under-determined. We use the additional degrees of freedom to pick that interpolant whose sobolev norm is minimum. Let $V$ refer to a Vandermonde matrix with a Chebyshev basis, $f$ the $N$ vector corresponding to $N$ equi-spaced sample values, $a$ the $M \gg N$ vector of spectral coefficients, $s$ the parameter corresponding to the Sobolev space containing the interpolant, and $D_s$ an appropriate diagonal matrix. $M$ depends on the minimum spacing between the samples and the number of samples. Then, MSN interpolation corresponds to the optimization problem

$$\arg\min_{V, a, f} \|a\|_s^2 = \|D_s a\|_2^2 = \sum_{m=0}^{M-1} (\|m\|_2^2 + 1)^{s/2} |a|^2.$$  \hspace{1cm} (1)

Here $m$ is a multi-index depending on the dimensionality of the problem. Intuitively speaking, MSN finds an interpolant that has minimum energy in higher frequencies. This in turn corresponds to the smoothest interpolant in the sense of the above sobolev norm. It has been shown that there exists a unique solution to (1) [1]. It is also shown that the interpolant converges point-wise to the underlying function in the vicinity of the samples, provided the right Sobolev space is chosen. Note that this convergence result is independent of the underlying sample distribution and illustrates the local convergence property of MSN.
interpolants.

4 Stabilizing MSN Computation: CODA

While (1) is setup and solved as an under-constrained problem for which we find the minimum norm solution, a corresponding dual Weighted Least Squares (WLS) problem can be specified as follows. We compute the MSN weights \( w_l \) such that

\[
\begin{align*}
\mathbf{w}_l = \arg \min_{\mathbf{w}} & \| \mathbf{D}_{s}^{-1} (V_{N \times \Pi(M)}^T \mathbf{w} - V_{1 \times \Pi(M)}^T(\mathbf{x}_l)) \|_2, \\
\text{where } V & \text{ is the Chebyshev Vandermonde matrix at } (\mathbf{x}_i)_{i=1}^N, \text{ an } N \text{ point neighborhood of } \mathbf{x}_l.
\end{align*}
\]

Note that

\[
f(\mathbf{x}_l) = \sum_{i=1}^N w_l(i) f(\mathbf{x}_i).
\]

The MSN interpolation problem becomes ill-conditioned with increasing problem size \( N \) and the Sobolev parameter \( s \). For a local interpolation, \( N \) does not need to be too large; as mentioned in the introduction, there is no significant computational advantage beyond a 10th order method. As per [3], for a diagonal ill-conditioned positive matrix scaling an otherwise well conditioned LS system, the solution can be computed stably. We present a poor man’s complete orthogonal decomposition based on SVDs to solve such a system. Consider any general weighted LS problem of the form

\[
\arg \min_x \| W(Ax - b) \|.
\]

Let \( y \) be the computed solution, and let \( \hat{y} \) be the true solution. Then, the goal is an algorithm that achieves a floating point accuracy bounded independent of \( W \) as below.

\[
\| y - \hat{y} \| \leq \epsilon_{\text{mach}} f(A) \| b \|,
\]

where \( f(A) \) is a function of \( A \) independent of \( W \). Unpivoted QR factorization of such ill-conditioned WLS systems are seen to suffer from numerical inaccuracies[3]. Hough and Vavasis proposed the complete orthogonal decomposition idea which uses a QR factorization with column pivoting which converts the given LS problem to one that is well conditioned up to a column scaling. Traditional algorithms do well to solve such systems to good floating point accuracies, and so an error bound of the form above is achieved. The key idea here is to change the ill-conditioned row-scaling into an ill-conditioned column scaling. Instead of a QR factorization with column pivoting as proposed by Hough and Vavasis, we use the SVD. The theory by Vavasis et al. also seeks that the weights be ordered; this is taken care by an apriori permutation.

We assume in this description that the factor \( W \) has been multiplied into the equation. Consider a singular value decomposition

\[
A = U \Sigma V^H.
\]
Then,
\[ AV = U\Sigma. \]
If we now solve for the modified LS problem
\[ U\Sigma\hat{x} = \text{LS} b, \]
where \( \hat{x} = V^H x \), then a traditional LS algorithm works well under floating point errors as well. However, the SVD algorithms achieve only backward error stability. Hence, we use a recursive SVD refinement approach in order to produce a numerically accurate complete orthogonal decomposition. This SVD based technique is described below. We require a numerical threshold \( \eta = O(1) \) that specifies the refinement. We assume that \( A \) has full column rank and well conditioned. This assumption is indeed valid for the MSN method assuming the underlying grid points that produce the Chebyshev Vandermonde matrix are themselves well spaced.

We first compute the SVD
\[ A = U\Sigma V^H \]
and apply \( V \) to \( A \), to get \( \hat{A} = AV \). Let \( A_0, V_0 \) be empty matrices to accumulate results. Also, let
\[ k = \arg \min_i \sigma_i < \frac{\sigma_0}{\eta}. \]
If such a \( k \) exists, then we split \( \hat{A} \) as
\[ \hat{A} = \begin{bmatrix} A_1 & A_2 \end{bmatrix} \]
where \( A_1 \) is the first \( k - 1 \) columns of \( A \). We also split \( V \) as
\[ V = \begin{bmatrix} V_1 & V_2 \end{bmatrix} \]
where \( V_1 \) has the first \( k - 1 \) columns of \( V \). We then refine \( A_2 \) by computing its SVD again as
\[ A_2 = \hat{U}\hat{\Sigma}\hat{V}^H \]
and try to split \( A_2\hat{V}, V_2\hat{V} \) as above. We accumulate \( A_0 = \begin{bmatrix} A_0 & A_1 \end{bmatrix} \). We also accumulate \( V_o = \begin{bmatrix} V_o & V_1 \end{bmatrix} \). The iteration stops when no such \( k \) exists as described above. At the end of the iterations, we compute the orthogonal decomposition \( A_o = QR \). Now \( Q, R, V_o \) specify a complete orthogonal decomposition.

In order to test the above algorithm we assume a known random solution to the 2D MSN system of size 784 \( \times \) 49 and solve this system using the factorization developed above. The threshold \( \eta \) used for CODA refinement was 10. The algorithm was observed to be sensitive to the ordering of the weights in the ill-condition system. To this effect, permutations corresponding to ascending and descending order of weights in \( D \) were employed to test the solver. Figure 2 summarizes the results obtained with various orderings. In Figure 2, we
compare three pairs of results. A QR based least squares solver in MATLAB is taken as the reference for comparison. The first pair compares unordered weights between QR and CODA. The second pair compares them with the weights sorted in ascending order, while the third pair compares them with the weights in descending order. It is clearly evident that the CODA algorithm with descending order of weights outperforms all other methods considered. Our experiments seem to indicate that descending order works best.

Figure 2: Results of CODA and variants to solve the 2D MSN system

Figure 3: The stencils used for discretization in the vicinity of points

5 Discretization of two dimensional Elliptic PDEs using the MSN Scheme

Consider a general 2nd order elliptic PDE, given by:

\[ \nabla . A \nabla u + b^T \nabla u + cu = f \]  

We wish to solve (2) in a compact region \( \Omega \subset \mathbb{R}^2 \) subject to the boundary condition

\[ e^T \nabla u + du = g \]
on the boundary $\partial \Omega$. We consider a discretization of (2) at $N_I$ points $\{x^i_k\} \in \Omega^o$, $k = 1, 2, \ldots N_I$. We call these the interior points. We also consider a discretization of (3) on $\partial \Omega$ at $N_B$ boundary points, $\{x^b_k\}$, $k = 1, 2, \ldots N_B$. We wish to solve (2) for $u(x^i_k)$ and perhaps $u(x^b_k)$ depending on the boundary conditions.

At each $\{x^i_k\}$, we consider a neighborhood of size corresponding to some order $d$ of local approximation. For a square stencil and equi-sampling, we consider a square of side $(2d - 1)h$, where $h$ is the sampling distance. Figure (2) depicts these neighborhood stencils at a far interior stencil (not involving any boundary point) as well as in the vicinity of the boundary.

We generalize the PDE problem and construct local approximation weights in a manner below. Let $\Omega \subset \mathbb{R}^2$ be a compact region, $L$ be a differential operator on the interior of $\Omega$, $B$ be another suitable boundary operator. Then we wish to solve the boundary value problem,

$$Lu = f : \Omega^o, Bu = g : \partial \Omega \quad (4)$$

Let $C$ be a finite collection of “grid points” in $\Omega$, $\delta > 0$. Let $B(x, \delta) = \{y \in \mathbb{R}^2 : \|y - x\|_\infty \leq \delta\}$.

We fix some $x^*$, and consider the set

$$C'(x^*) = \left\{ \frac{\|x - x^*\|_\infty}{\delta} : x \in C \cap B(x^*, \delta) \right\}$$

Let $\{\phi_k\}$ be an orthonormal set on $[-1, 1]^2$, $M = M(x^*)$ be the cardinality of $\{\phi_k\}$, where $M$ is computed to be the mesh norm $[1]$ of $C'(x^*) = \{y_0, y_1, \ldots, y_{N-1}\}$. Let $N$ be the cardinality of $C'(x^*)$, $M > N$.

We find weights such that

$$w(x^*) = \arg \min_{w \in \mathbb{R}^N} \left\{ \sum_{l=1}^M (1 + \|l\|_2^2)^{-s/2} \left( \sum_{k=0}^{N-1} w_k \phi_l(y_k) - L\{\phi_l\}(0) \right)^2 \right\} \quad (5)$$

In order to discretize (2) and (3) at a point $x^*$, we place a stencil $B(x^*, (2d - 1)h)$, and at $C'(x^*)$ we evaluate the Chebyshev Vandermonde Matrix $V$. The operator $L$ is applied to $V(0)$ since $x^*$, the stencil center is mapped to the origin. Then, the above minimization problem problem becomes

$$\arg \min_{w} \|D_s \left( V^T w - L(V^T(0)) \right) \|_2$$

where $D_s$ is the diagonal matrix with weights $(1 + \|l\|_2^2)^{-s/2}$. As described in the previous section, we solve this equation using the CODA method. Here the solution $w$ specifies the linear combination of $u(C'(x^*))$ that approximates $L(u)(x^*)$.

Let $N = N_I + N_B$. Let $F_2$ be the $N_I \times N$ sparse matrix that contains the weights corresponding to $\nabla A \nabla u$. The $l^{th}$ row of $F_2$ contains the weights that
approximate $\nabla.A\nabla u(x_i)$ using $u(C(x_i))$. Let $F_{x^i}, F_{y^i}$ correspond to the $N_I \times N$ matrices that contain the weights for $\frac{\partial}{\partial x^i}, \frac{\partial}{\partial y^i}$ respectively. Then (2) becomes

$$(F_2 + B_1 F_{x^i} + B_2 F_{y^i} + I^i)u = f_{N_I}, B_i = b_i(x), x \in \{x_k^i\}. \quad (6)$$

Similarly, (3) becomes

$$(E_1 F_{x^b} + E_2 F_{y^b} + I^b)u = g_{N_B}, \quad (7)$$

where $F_{x^i}, F_{y^i}$ are $N_B \times N$ sparse matrices that represent the operators $\frac{\partial}{\partial x^i}, \frac{\partial}{\partial y^i}$ at the boundary points. Let $C^i, D^b$ represent appropriately sized diagonal matrices representing coefficients $c, d$ in (2) and (3) respectively. The grand system that we solve is now setup as follows.

$$F u = \left( F_2 + B_1 F_{x^i} + B_2 F_{y^i} + I^i \right) u = \left( f_{N_I} \right) \quad (8)$$

To reduce the condition number of $F$, we perform a row 2-norm scaling. Full scale equilibration experiments are yet to be explored since the simple scaling seems satisfactory in controlling the condition number. Let this diagonal scaling be represented by $D_1$. Then we have a system

$$D_1 F u = D_1 \left( f_{N_I} \right) \quad (9)$$

Since $f_{N_I}$ and $g_{N_B}$ are at scales $O(h^2)$ and $O(1)$ respectively, we setup and solve for two separate systems of the form:

$$D_1 F \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = D_1 \begin{bmatrix} f_{N_I} \\ 0 \end{bmatrix} \quad (10)$$

The overall solution is then reconstructed as $u = u_1 + u_2$. If the condition number induced by the operator is high, as with the Bi-Harmonic operator, or singularly perturbed operators, we need to resort to iterative refinement. The proof that the local discretization of the PDE converges with a high order follows from the fact that the local interpolant converges with a high order as well and is presented in [10].

6 Local Convergence

In this section, we present an extension to Theorem 2.2b in [1] that describes local convergence of the MSN interpolant under linear differential operators.

Let $\Pi_m$ be the class of all (univariate) algebraic polynomials of degree at most $m$, and for $a < b$, $f \in C[a,b],$

$$\text{dist} (f, \Pi_m, [a,b]) = \min_{P \in \Pi_m} \|f - P\|_{\infty;[a,b]} = \|f - B_{m.a,b}\|_{\infty;[a,b]}. \quad (11)$$

The notation $[a,b]$ shall be omitted in case of $a = -1, b = 1$. 

9
Proposition 1 (a) Let $k \geq 1$ be an integer, $m \geq k$ be an integer, and $f \in C^k[-1,1]$. Then
\[
\text{dist } (f, \Pi_m) \leq cm^{-k}\text{dist } (f^{(k)}, \Pi_{m-k}).
\] (12)
Moreover, for $x \in [-1,1],$
\[
|f^{(k)}(x) - B_m^{(k)}(f)(x)| \leq c(\sqrt{1-x^2} + 1/m)^{-k}\text{dist } f^{(k)}, \Pi_{m-k}),
\] (13)
where $c$ is a constant depending on $k$ alone.
(b) Let $P \in \Pi_m$. Then for integer $k \geq 1,$
\[
\max_{x \in [-1,1]} |(1-x^2)^{k/2}P^{(k)}(x)| \leq cm^k\|P\|_{\infty,[-1,1]}.
\] (14)

Proof. In part (a), (12) is proved in [11, Chapter 7, estimate (6.7), p. 220] (needs to be iterated $k$ times), the estimate (13) is proved in [11, Chapter 8, Corollary 4.6, p. 249–250]. Part (b) is proved as [11, Chapter 8, Theorem 7.6, p. 265].

Corollary 1 Let $r \geq 1$ be an integer, $f \in C^r[-1,1], m \geq r$, $P \in \Pi_m$, and
\[
\|f - P\|_{\infty,[-1,1]} \leq \epsilon.
\] (15)
Then for $k = 1, \cdots, r$ and $x \in [-1,1],$
\[
|\left(1-x^2\right)^{k/2}f^{(k)}(x) - P^{(k)}(x)| \leq c\left\{\text{dist } f^{(k)}, \Pi_{m-k} + m^k\epsilon\right\}.
\] (16)

Proof. We use in order, (13), (14) (with $B_m(f) - P$ in place of $P$), (11), (12), and (17) to obtain
\[
|\left(1-x^2\right)^{k/2}f^{(k)}(x) - P^{(k)}(x)|
\leq |\left(1-x^2\right)^{k/2}f^{(k)}(x) - B_m^{(k)}(f)(x)| + |\left(1-x^2\right)^{k/2}(B_m^{(k)}(f)(x) - P^{(k)}(x))|
\leq c\left\{\text{dist } f^{(k)}, \Pi_{m-k} + m^k\|B_m(f) - P\|_{\infty,[-1,1]}\right\}
\leq c_1\left\{\text{dist } f^{(k)}, \Pi_{m-k} + m^k\|f - B_m(f)\|_{\infty,[-1,1]} + m^k\|f - P\|_{\infty,[-1,1]}\right\}
\leq c_2\left\{\text{dist } f^{(k)}, \Pi_{m-k} + m^k\text{dist } f, \Pi_m\} + m^k\|f - P\|_{\infty,[-1,1]}\right\}
\leq c_3\left\{\text{dist } f^{(k)}, \Pi_{m-k} + m^k\epsilon\right\}.
\]

Let $x_0 \in \mathbb{R}, \delta > 0, I = [x_0 - \delta, x_0 + \delta].$ It is easy to verify that if $\tilde{f}(y) = f(x_0 + y\delta),$ then $\tilde{f}^{(k)}(y) = \delta^k f^{(k)}(x_0 + y\delta)$, and hence, that
\[
\text{dist } (f^{(k)}, \Pi_{m-k}; I) = \delta^k \text{dist } (\tilde{f}^{(k)}, \Pi_{m-k}).
\]
Therefore, the above corollary can be reformulated in the form
Corollary 2  Let \( x_0 \in \mathbb{R}, \delta > 0, I = [x_0 - \delta, x_0 + \delta], r \geq 1 \) be an integer, \( f \in C^r[I], m \geq r, P \in \Pi_m, \) and

\[
\|f - P\|_{\infty;I} \leq \epsilon.
\]

Then for \( k = 1, \cdots, r \) and \( x = x_0 + y\delta \in I, \)

\[
\left| (1 - y^2)^{k/2} \left( f^{(k)}(x) - P^{(k)}(x) \right) \right| \leq c \left\{ \text{dist} \left( f^{(k)}, \Pi_{m-k}, I \right) + \delta^{-k}m^k \epsilon \right\}.
\]

In particular,

\[
\left| f^{(k)}(x_0) - P^{(k)}(x_0) \right| \leq c \left\{ \text{dist} \left( f^{(k)}, \Pi_{m-k}, I \right) + \delta^{-k}m^k \epsilon \right\}.
\]

A multivariate version of Corollary 2, especially the implication that an estimate of the form (17) implies (19) is also immediate, and will be assumed now.

We now revert to the notation in the proof of Theorem 2.2(b), in [1]. A few typos are in order. One is that \( K = [x_0 - \delta, x_0 + \delta] \), and subscript \( K \) on line 3 from below on p. 20 is replaced by \([-1,1]^q\). Writing

\[
Q(x_0 + \delta y) = V_r(\tilde{f}, y) = V_r(P, y),
\]

the estimate on line 3 from below on p. 20 (as amended) becomes

\[
\|f - Q\|_{\infty,K} \leq c\delta^{s-q/p}, \|P_n^* - Q\|_{\infty,K} \leq c\delta^{s-q/p}.
\]

We now use multivariate version of Corollary 2 with \( 2r \) in place of \( m, Q \) in place of \( P, \) once with \( f \) and once with \( P_n^* \) to conclude that

\[
|D^k(f - Q)(x_0)| \leq c \left\{ \text{dist} \left( D^k, \Pi_{2r-n}^{p^*}, K \right) + \delta^{-\|k\|_1, r}\|k\|_1, \delta^{s-q/p} \right\},
\]

and

\[
|D^k(P_n^* - Q)(x_0)| \leq c \left\{ \text{dist} \left( D^k_{p^*}, \Pi_{2r-n}^{p^*}, K \right) + \delta^{-\|k\|_1, r}\|k\|_1, \delta^{s-q/p} \right\},
\]

Since \( D^k f \) and \( D^k_{p^*} \) are both in \( B_{c,s-q/p-\|k\|_1, \infty} \), we estimate their degrees of approximation as in the proof of Theorem 2.2(b), and deduce that

\[
|D^k(f - P_n^*)(x_0)| \leq c\delta^{s-q/p-\|k\|_1},
\]

where the constant \( c \) now depends also on \( r, \) which is the integer part of \( s. \)

7 Numerical Results

In order to test the above approach, we consider PDEs in various geometries. Starting with a square region, we morph the boundary points using the transformation in (23) to produce a family of boundary curves corresponding to varying
Figure 4: Various geometries generated using a transformation from the Square geometry

values of $1 \leq p \leq 20$. For our purposes we choose $p = 1, 2, 20$, corresponding to the geometries shown in Figure 7.

$$x_{\text{morphed}} = \frac{\|x\|_p}{\|x\|_2}, x \neq 0$$ (23)

We equi-sample these geometries and interior points closer to the boundary points than $0.5h$ are discarded, where $h$ is the interior sample spacing.

Results below include the maximum relative error in the solution denoted as $\|e\|_\infty = \frac{\|u - u_0\|_\infty}{\|u_0\|_\infty}$. We measure the relative residue as $\frac{\|F u - g\|_\infty}{\|g\|_\infty}$ where $F$ denotes the assembled FD matrix, prior to scaling. $g$ denotes the appropriate right hand side. The maximum number of weights used is denoted as $L^2$. To measure the condition number, we use the statistical estimation method of Laub and Kenney [12]. The condition number thus estimated is denoted as $\kappa(\hat{F})$, where $\hat{F}$ denotes the FD matrix after scaling. We compare our results with the Matlab FEM Toolbox (PDETool) [13] and Deal II [4].

7.1 Negative Definite Helmoltz

$$\nabla^2 u - u = f$$

$$\Omega : [-0.5, 0.5]^2 \Rightarrow p$$
We assume a complicated solution to the above equation, and set \( f \) using \( u \) as below.

\[
    u = \frac{1}{1 + 1000(x^2 + y - 0.3)^2} + \frac{1}{1 + 1000(x + y - 0.4)^2} + \frac{1}{1 + 1000(x + y^2 - 0.5)^2} + \frac{1}{1 + 1000(x^2 + y^2 - 0.25)^2}.
\]

Note that \( u \) has singularities very close to the real plane through the domain. We compare the result of solving this equation with PDETool. We see that our method compares very favourably against this in terms of error achieved with comparable number of triangles and samples.

The following observations are immediate from Figures 7.1 through 9. The MSNFD solution converges under the arbitrary geometries considered with equi-sampled interior points. The method achieves nearly 4 orders higher accuracy compared with the Matlab FEM method for a comparable number of samples. Also, increasing window size is observed to produce higher order of convergence. The condition number scaling seems to be similar to that of the FEM approach. There seems to exist a large constant scaling the condition number. The timing chart at the first glance appears to be in favour of the FEM approach. But we believe that this is misleading. We feel that in order for the comparison to be fair, the minimum time taken by each method to get to a given accuracy needs to be compared. The times presented here are both on comparable CPUs. We extrapolate the time taken by the FEM method to a comparable accuracy. Figure 7 depicts this comparison. This extrapolation was necessary since the FEM method could not be run in the computer with the given amount of memory.
Figure 7: Error and Condition number to equation in section 7.1, $p = 1$

Figure 8: Minimum Time taken to solve the Sparse System from equation in section 7.1 to a given accuracy

Figure 9: Error and Condition number to equation in section 7.1, $p = 2$
7.2 Negative Definite Helmoltz with Smooth Solution

We now consider the same problem as in the previous solution, but we choose the right hand side to arise from a rather smooth Runge function as the solution. This example is chosen to depict that the MSNFD method indeed gives back all 12 digits of accuracy with just a 200 \times 200 grid.

\[ \nabla^2 u - u = f \]

\[ \Omega : [-0.5, 0.5]^2 \Rightarrow p \]

\[ u = \frac{1}{1 + x^2 + y^2} \]

7.3 High-Frequency Helmoltz

We first consider a Helmoltz equation with a high wave number. Results of solving this problem with the MSNFD method and the Matlab FEM Toolbox (PDETool) are presented in Figures 5 through 8. This being a stiff problem, PDETool is seen to have a lower convergence rate compared to the MSNFD. Considering the minimum time taken to get to a given accuracy, we see that unless the expected accuracy is less than 2 digits, we see that the MSNFD outperforms the FEM approach. We extrapolate the time take by FEM to get to the given accuracy. This extrapolation was necessary since the FEM method could not be run on our computer with the given amount of memory. For \( p = 2 \) this minimum sampling rate for the solution to begin to converge is observed to be about 100 \times 100. While the condition number scales similar to the FEM method, there is a large constant scaling it.

\[ \nabla^2 u + 10000u = f \]
Time taken to solve the grand system for Problem 1, with $p = 1$

Figure 11: Time taken to solve the Sparse System from equation in section 7.1, $p = 1$

Figure 12: Geometry corresponding to equation in section 7.5, $p = 1$
Figure 13: Error and Condition number to equation in section 7.3, $p = 1$

Figure 14: Error and Condition number to equation in section 7.3, $p = 2$

Figure 15: Minimum Time taken to solve the Sparse System from equation in section 7.3 to a given accuracy
Figure 16: Error and Condition number to equation in section 7.3, $p = 20$

Table 1: Numerical results for the problem in section 7.2, $p = 20$

<table>
<thead>
<tr>
<th>$\frac{1}{n}$</th>
<th>$\frac{|u-u_0|<em>\infty}{|u_0|</em>\infty}$</th>
<th>$\frac{|F u-g|<em>\infty}{|g|</em>\infty}$</th>
<th>$L^2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>30</td>
<td>4e-09</td>
<td>8.7e-07</td>
<td>123</td>
</tr>
<tr>
<td>100</td>
<td>7e-12</td>
<td>2.6e-09</td>
<td>125</td>
</tr>
<tr>
<td>200</td>
<td>1e-11</td>
<td>5.7e-10</td>
<td>125</td>
</tr>
<tr>
<td>30</td>
<td>6e-10</td>
<td>1.2e-07</td>
<td>171</td>
</tr>
<tr>
<td>100</td>
<td>6e-12</td>
<td>2.5e-09</td>
<td>173</td>
</tr>
<tr>
<td>200</td>
<td>2e-11</td>
<td>7.1e-10</td>
<td>175</td>
</tr>
<tr>
<td>30</td>
<td>3e-10</td>
<td>5.7e-08</td>
<td>227</td>
</tr>
<tr>
<td>100</td>
<td>3e-12</td>
<td>1.6e-09</td>
<td>229</td>
</tr>
<tr>
<td>200</td>
<td>2e-11</td>
<td>1.5e-09</td>
<td>233</td>
</tr>
<tr>
<td>30</td>
<td>6e-11</td>
<td>1.0e-08</td>
<td>293</td>
</tr>
<tr>
<td>100</td>
<td>4e-12</td>
<td>8.4e-10</td>
<td>296</td>
</tr>
<tr>
<td>200</td>
<td>2e-12</td>
<td>4.1e-10</td>
<td>365</td>
</tr>
<tr>
<td>30</td>
<td>5e-12</td>
<td>8.8e-10</td>
<td>373</td>
</tr>
</tbody>
</table>
Figure 17: Error and Condition number to equation in section 7.3, $p = 1$

Figure 18: Error and Condition number to equation in section 7.3, $p = 2$

Figure 19: Minimum Time taken to solve the Sparse System from equation in section 7.3 to a given accuracy
7.4 Positive Definite Helmholtz - comparison with dealii

In this section, we compare the result of solving an example problem provided by the Deal II [4] package. It considers a positive definite Helmholtz equation with mixed boundary conditions. $f$ is obtained by using $u$ in the equation below.

$$
\Omega : [-0.5, 0.5]^2 \Rightarrow p
$$

$$
u = \frac{1}{1 + 1000(x^2 + y - 0.3)^2} + \frac{1}{1 + 1000(x + y - 0.4)^2} + \frac{1}{1 + 1000(x + y^2 - 0.5)^2} + \frac{1}{1 + 1000(x^2 + y^2 - 0.25)^2}
$$

7.5 Region with a hole

We consider a region as described below. Table 2 in Figure 11 shows that the MSNFD approach provides very good results for the considered smooth solution.
removing interior points too close to the boundary points alleviates the geometry induced ill-conditioning.

5.3. Region with a hole

boundary conditions on

Figure 5.2 below plots the maximum error in the solution as a function of the grid density. The graph provides

Table 1: Numerical results for the Multi-

Table 2: Numerical results for the problem

on a 600 × 600 grid

in section 5.3, p = 1

Figure 21: Numerical results for problems 7.6 and 7.5

Figure 22: Plot of ∥e∥∞ = ∥u−u_h∥∞ ∥u_0∥∞ Vs the grid density 1/n

21
It may also be observed that our simple heuristic of removing interior points too close to the boundary points alleviates the geometry induced ill-conditioning.

\[ \nabla^2 u - u = f \]
\[ \Omega : [-0.5, 0.5]^2 \backslash [-0.25, 0.25]^2 \Rightarrow p \]
\[ u = \frac{1}{1 + x^2 + y^2} \]

7.6 Multi Scale

We finally consider an example of a PDE whose analytic solution is not known. This is an example of a Multiscale problem as investigated by Shu et al. [14]. In their work a multiscale discontinuous Galerkin method was setup to solve this problem. A spectral method as a reference and convergence results were obtained. In our case, as in Table 1 in Figure 11, the error measured at a grid density \( i \) is the error between the solution at density \( i \) and density \( i - 1 \) when interpolated to the new set of high density grid points, relative to the solution generated by the grid at density \( i - 1 \). These errors were measured by reinterpolating on a 600x600 grid. It may be observed that there is convergence and the solution is consistent as observed to about 4 digits. Increasing rate of convergence is also observed with increasing stencil sizes.

\[ \nabla . A \nabla u = x + y \]
\[ A = \begin{bmatrix} \frac{1}{4 + x + \sin(x/0.01)} & 0 \\ 0 & \frac{1}{4 + y + \sin(y/0.01)} \end{bmatrix} \]
\[ \Omega : [-1, 1]^2 \]
\[ u = 0 : \partial \Omega \]

8 Conclusions and Extensions

A Higher Order numerical method to solve elliptic PDEs in two dimensions was presented. Comparison with Matlab FEM Toolbox (PDETool) as well as Deal II were presented. The MSNFD approach performed better and exhibited higher order of convergence than the FEM methods for problems considered. We believe that a higher order method would be very useful to get to accuracies which we believe may be hard and perhaps even impossible with lower order methods. Several additional problems and the corresponding results will be available at our website [5] in the technical report [10]. We believe adaptive gridding to be important and this currently work in progress. A comaparable Finite Element approach together with the application of MSNFD to the BiHarmonic Type equations as well as Exterior problems would be presented in our future work [15]. A similar discretization of the integral equation of the PDE and a fast solution of the resulting dense matrix is possible as well.
References


