University of California

Santa Barbara

Applications of the Minimum Sobolev Norm and Associated Fast Algorithms

A dissertation submitted in partial satisfaction

of the requirements for the degree

Doctor of Philosophy

in

Mathematics

by

Christopher Henry Gorman

Committee in charge:

Professor Shivkumar Chandrasekaran, Chair

Professor Xu Yang

Professor Hector Ceniceros

June 2019

The dissertation of Christopher Henry Gorman is approved.

Professor Xu Yang

Professor Hector Ceniceros

Professor Shivkumar Chandrasekaran, Chair

May 2019

Applications of the Minimum Sobolev Norm and Associated Fast Algorithms

Christopher Henry Gorman

Dedicated to the Blessed Holy Trinity: Father, Son, and Holy Spirit;

and my future wife: may I find you soon.

When I look at your heavens, the work of your fingers,

the moon and the stars, which you have set in place,

what is man that you are mindful of him,

and the son of man that you care for him?

— Psalm 8:3–4

For by him all things were created, in heaven and on earth, visible and invisible, whether thrones or dominions or rulers or authorities – all things were created through him and for him. And he is before all things, and in him all things hold together.

— Colossians 1:16–17

Acknowledgements

Graduate school has been a long and difficult road, and there are many people I would like to thank, not all of whom can be listed here. Without their help and support, I would not have graduated.

First and foremost, I would like to thank my doctoral committee: Shiv Chandrasekaran, Xu Yang, Hector Ceniceros, and Sherry Li. Shiv, you have been a great help through the entire process as I studied numerical linear algebra and fast algorithms, always reminding me about numerical stability and floating point arithmetic. Xu, thank you for teaching me about partial differential equations and asymptotic analysis. I am so glad to have both Shiv and Xu as doctoral advisors. Hector, thank you for teaching me about finite difference methods and pushing me to clearly articulate my results. Sherry, it was great to spend two summers working on STRUMPACK with you, Pieter, Gustavo, and Yang at Berkeley Lab. I learned a lot about computing, randomization, and software and algorithm development.

During my six years at UCSB, I have gained many friends. Those who have been with me from the beginning: Nancy, Steve, Jay, and Kyle. I have had a great time with you all. Those in Shiv's group: Kristen, Abhejit, Nithin, and Ethan. It has been great to know you while learning from Shiv. Medina, thank you for all the work you do in the Mathematics Department.

While living in Santa Barbara, I have gained many friends and mentors at Good Shepherd Lutheran Church. These include Pastor and Kitty; Ted and Andrea; and Chuck and Lois. Thank you for teaching me about life outside of school and work.

Thank you for the many friends I made in Reformed University Fellowship (RUF) at UCSB. Thank you especially to Johnathan (and Jaimeson) for leading RUF. You and Grace show God's love in Jesus to people on campus. Evan: it has been great to have

another graduate student who is going through the same process. Nathan and Stanley: it has been great to get to know you both these past few years.

Thank you to my family and especially my parents: Dad and Lin, and Mom and Paul. In particular, I must thank Vince and Megan: it has been great to get to know you and spend time with your family these years, especially Joe and Alice. Joe, you have been a great mentor and helped me early on in my doctoral program by encouraging me to recognize my dissertation is not the end, but rather the beginning, of my work.

My twin brother James: It was great going to Wabash College with you. Working on mathematics and physics homework late at night helped us succeed as undergraduates and continue on to earn our doctorates. Thank you for being my best friend.

Christopher Henry Gorman

University of California
Santa Barbara, CA 93106-3080

805-893-5306
gorman@math.ucsb.edu

Education

Ph.D. Mathematics June 2019
Emphasis: Computational Science and Engineering
Dissertation: Applications of the Minimum Sobolev Norm and Associated Fast Algorithms
Advisors: Shivkumar Chandrasekaran and Xu Yang
University of California, Santa Barbara, CA

M.A. Mathematics December 2014
University of California, Santa Barbara, CA

A.B. *Summa Cum Laude* Mathematics and Physics May 2013
Wabash College, Crawfordsville, IN

Research Experience

Graduate Intern – Systems Engineering Summer 2018
Mark Nussmeier, FLIR Systems, Inc., Goleta, CA
Performed design tests for thermal camera development

Graduate Student – Non GSRA Summer 2016/2017
Dr. Xiaoye Sherry Li, Lawrence Berkeley National Laboratory, Berkeley, CA
Assisted in the development of fast algorithms for Hierarchically Semi-Separable matrices

Research Graduate Student III/IV Summer 2014/2015
Dr. Nan Yu, Jet Propulsion Laboratory, Caltech, Pasadena, CA
2014: Performed error progation calculations and simulations for gravity gradiometer experiments
2015: Simulated atom interferometery to help development of equivalence principle test

Physics Research Assistant Summer 2012
Dr. K. Vollmayr-Lee, Bucknell University, Lewisburg, PA
Investigated structural glasses and found scaling predictions from Spin Glass theory apply to Silica

Physics Research Assistant Summer 2011
Dr. V.V. Kresin, University of Southern California, Los Angeles, CA
Studied nanoclusters and their formation while enhancing laboratory practices

Physics Research Assistant Summer 2010
Dr. M.J. Madsen, Wabash College, Crawfordsville, IN
Used Finite Element Analysis software interfaced with *Mathematica* to design compact torodial ion trap

Publications

Madsen, M.J. and Gorman, C.H., **"Compact toroidal ion-trap design and optimization,"** *Phys. Rev. A*, **82**, 043423 (2010)

K. Vollmayr-Lee, C.H. Gorman, and H.E. Castillo, **"Universal Scaling in the Strong Glass Former SiO_2,"** J. Chem. Phys. **144**, 234510 (2016) (**JCP Editors' Pick**)

P. Ghysels, X. S. Li, C. Gorman, and F. H. Rouet, **"A robust parallel preconditioner for indefinite systems using hierarchical matrices and randomized sampling,"** *2017 IEEE International Parallel and Distributed Processing Symposium (IPDPS)*, Orlando, FL, 2017, pp. 897-906.

S. Chandrasekaran, C.H. Gorman, and H.N. Mhaskar, **"Minimum Sobolev norm interpolation of scattered derivative data,"** Journal of Computational Physics **365**, pp. 149–172 (2018)

C. Gorman, G. Chávez, P. Ghysels, T. Mary, F. H. Rouet, and X. S. Li, **"Robust and Accurate Stopping Criteria for Adaptive Randomized Sampling in Matrix-free HSS Construction,"** SIAM J. Sci. Comput., 2019. To appear.

Abstract

Applications of the Minimum Sobolev Norm and Associated Fast Algorithms

by

Christopher Henry Gorman

This dissertation focuses on the development, implementation, and analysis of fast algorithms for the Minimum Sobolev norm (MSN). The MSN method obtains a unique solution from an underdetermined linear system by minimizing a derivative norm in the appropriate Hilbert space. We obtain fast algorithms by exploiting the inherent structure of the underlying system. After performing an Inverse Discrete Cosine Transform, a small number of additional operations are required. Results show the method performs as well as Chebyshev interpolation when approximating smooth functions and better than a wide variety of smooth Chebyshev filters when attempting to approximate rough functions.

One chapter is devoted to analyzing a stochastic norm estimate which is useful when computing low-rank approximations of matrices. This estimate allows us to compute approximations with relative error close to machine precision, which previously was not possible.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The major goal of this line of research is to develop high order, numerically stable fast algorithms for solving elliptic partial differential equations using the Minimum Sobolev norm (MSN). This will require much more work than can be completed in one dissertation. The present work focuses on developing fast algorithms to solve interpolation and ordinary differential equation problems using the MSN method, which will be a stepping stone to understand the structure of the matrices arising in 2D and 3D PDEs. We introduce these ideas by discussing some of the problems present in interpolation methods and how the MSN method attempts to solve them.

1.1 Lagrange Interpolation and Known Difficulties

The well-known Weierstrass Approximation Theorem, which we reproduce for completeness, says continuous functions on compact, connected intervals can be approximated arbitrarily well by polynomials:

Theorem 1.1 (Weierstrass Approximation Theorem; Theorem 7.26 in [42])
If $f \in C[a, b]$, then there exists a sequence of polynomials $\{P_n\}_{n=1}^{\infty}$ such that

$$\lim_{n\to\infty} ||f - P_n||_{\infty,[a,b]} = 0. \tag{1.1}$$

Here,

$$||g||_{\infty,[a,b]} \equiv \sup_{x\in[a,b]} |g(x)| \tag{1.2}$$

is the supremum norm. When the interval $[a, b]$ is understood, we may write $||\cdot||_\infty$ in place of $||\cdot||_{\infty,[a,b]}$. This theorem shows that the set of polynomials $\mathcal{P}$ is dense in $C[a, b]$, the space of continuous functions on $[a, b]$. We let $\mathcal{P}_n$ denote all polynomials of degree at most n. This gives rise to an important concept: degree of approximation. We also have the following theorem:

Theorem 1.2 (Best Uniform Approximation of Continuous Functions; Section 1.1 in [40]) If $f \in C[a, b]$ and $n \in \mathbb{N}_0$, then there exists a unique $q_n \in \mathcal{P}_n$ so that

$$||f - q_n||_\infty = \inf_{p\in\mathcal{P}_n} ||f - p||_\infty . \tag{1.3}$$

We set

$$E_n(f) \equiv \inf_{p\in\mathcal{P}_n} ||f - p||_\infty \tag{1.4}$$

and have

$$E_0(f) \geq E_1(f) \geq E_2(f) \geq \cdots \to 0. \tag{1.5}$$

There are many ways to prove the Weierstrass Approximation theorem. In [42, Chapter 7], Rudin convolves against a polynomial kernel. This is useful theoretically but in practice, one may only have function and derivative information at particular points. In order to reconstruct the underlying function, we want to use these function and derivative values to build an approximation, frequently chosen to be a polynomial. This is interpolation.

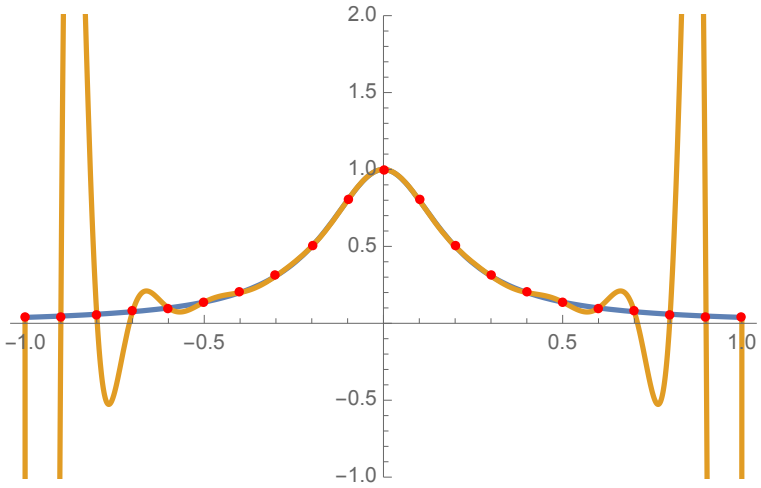


Figure 1.1: Here is an example of Lagrange interpolation of the Runge function $[1+25x^2]^{-1}$ using Lagrange interpolation with 21 equally-spaced nodes.

Interpolation Problem: Let $f \in C[-1,1]$ be continuous and specify a sequence of grid points

$$-1 \leq x_{1;n} < x_{2;n} < \cdots < x_{n;n} \leq 1. \tag{1.6}$$

for $n \in \mathbb{N}$. Determine a polynomial p_n with $\deg p_n = m(n)$ which satisfies the conditions

$$f(x_{k;n}) = p_n(x_{k;n}), \tag{1.7}$$

and ascertain under what restrictions on f and $\{x_{k;n}\}_{k=1}^{n}$ ensure

$$||f - p_n||_\infty \to 0, \quad n \to \infty. \tag{1.8}$$

A popular choice is to set $m(n) = n-1$, resulting in Lagrange interpolation. In Fig. 1.1, we see an example of Lagrange interpolation on equally-spaced nodes of the Runge function $[1+25x^2]^{-1}$ on $[-1,1]$. This function is analytic; even so, in [43] Runge proved that Lagrange interpolation diverges in this case. In particular, there are large oscillations near the boundary points.

In fact, much more is known about Lagrange interpolation, and we introduce notation which will make this discussion easier. Let

$$X = \{x_{k;n} \mid k = 1, \cdots, n; n \in \mathbb{N}\} \tag{1.9}$$

be an interpolatory matrix. Then, given $f \in C[-1, 1]$, we have the following standard definitions [46, Chapter 1]:

$$\begin{aligned} L_n(f, X, x) &= \sum_{k=1}^{n} \ell_{k,n}(X, x) f(x_{k;n}) \\ \Omega_n(X, x) &= \prod_{i=1}^{n} (x - x_{i;n}) \\ \ell_{k,n}(X, x) &= \frac{\Omega_n(X, x)}{\Omega_n'(X, x_{k;n})(x - x_{k;n})} \\ \lambda_n(X, x) &= \sum_{k=1}^{n} |\ell_{k,n}(X, x)| \\ \Lambda_n(X) &= ||\lambda_n(X, x)||_{\infty,[-1,1]}. \end{aligned} \tag{1.10}$$

Naturally, $L_n(f, X, x)$ is the Lagrange interpolating polynomial of degree at most $n - 1$ for the interpolation matrix X. The Lebesgue constants $\Lambda_n(X)$ are of critical importance, as we see

$$\begin{aligned} |L_n(f, X, x) - f(x)| &\leq |L_n(f, X, x) - q_{n-1}(x)| + |f(x) - q_{n-1}(x)| \\ &\leq |L_n(f - q_{n-1}, X, x)| + E_{n-1}(f) \\ &\leq [\Lambda_n(X) + 1] E_{n-1}(f). \end{aligned} \tag{1.11}$$

Here, $q_{n-1} \in \mathcal{P}_{n-1}$ is the minimizer in the supremum norm, and we note $L_n : \mathcal{P}_{n-1} \to \mathcal{P}_{n-1}$ is the identity map. Because $E_n(f) \to 0$, Lagrange interpolation converges when $\Lambda_n(X)E_{n-1}(f) \to 0$.

If we could find an interpolatory matrix Y so that $\Lambda_n(Y)$ is bounded, then $L_n(f, Y) \to f$ uniformly. Unfortunately, this is not the case. In [50], Vértesi references Faber (1914) as proving

$$\Lambda_n(X) > \frac{1}{8\sqrt{\pi}} \log n \tag{1.12}$$

for every X, showing $\Lambda_n(X)$ is unbounded. One popular set of interpolation nodes is the zeros of the Chebyshev polynomials:

$$\begin{aligned} T &= \{z_k^n \mid k = 1, \cdots, n; n \in \mathbb{N}\} \\ z_k^n &= \cos\left[\frac{\pi}{n}\left(n - k + \frac{1}{2}\right)\right]. \end{aligned} \tag{1.13}$$

The Chebyshev polynomials $T_n(x)$ are a set of orthogonal polynomials which will be discussed in detail in Sec. **??**. In [7], it was shown

$$\Lambda_n(T) < 8 + \frac{2}{\pi} \log n, \tag{1.14}$$

For this reason, we see that the interpolatory matrix T is close to optimal and, coupled with fast interpolation methods, gives reason for its popularity. Better bounds for Lebesgue constants can be found in [44].

We also give bounds for equally-spaced points, setting

$$E = \left\{ -1 + 2\frac{k-1}{n-1} \middle| k = 1, \cdots, n; n \in \mathbb{N} \right\}. \tag{1.15}$$

In [49], Trefethen and Weideman give the bounds

$$\frac{2^{n-2}}{n^2} < \Lambda_n(E) < \frac{2^{n+3}}{n} \tag{1.16}$$

as well as referencing the asymptotic result and some of the history of equally-spaced interpolation. Clearly, the exponential growth of $\Lambda_n(E)$ helps quantify how much worse E is when compared with T.

This divergence is not restricted to equally-spaced point distributions, though. In fact, we have the following result:

Theorem 1.3 (Theorem 4.3 in [46])

For an interpolatory matrix $X \subset [-1,1]$, there exists $h \in C[-1,1]$ so that

$$\limsup_{n\to\infty} |L_n(h,X,x)| = \infty \tag{1.17}$$

for almost every $x \in [-1,1]$.

So, there is no interpolatory matrix Y so that $||L_n(f,Y) - f||_\infty \to 0$ for all continuous functions f and the approximation error can be arbitrarily bad.

1.2 Possible Solutions to Divergence of Lagrange Interpolation

Although the previous result paints a bleak picture of Lagrange interpolation, this is true only in extreme situations. From [40, Chapter 1], we have the following theorem discussing how the degree of approximation is related to smoothness:

Theorem 1.4 (Jackson Inequality)
If $g \in C^k[-1,1]$ and $g^{(k)}$ is α-Hölder with Hölder constant L, then for $n > k$, we have

$$E_n(g) \leq \frac{c}{n^k}\left(\frac{1}{n-k}\right)^\alpha \tag{1.18}$$

with $c = 6^{k+1}e^k(1+k)^{-1}L$.

This theorem shows that if g is merely α-Hölder continuous, then $||L_n(g,T) - g||_\infty \to 0$ by Eq. (1.11). As noted above, we can have $||L_n(f,E) - f||_\infty \not\to 0$ even when f is analytic.

We previously noted $L_n(f,E)$ has large oscillations in the Runge example. Because of this, there has been interest in Hermite-Fejér interpolation. Given an interpolatory matrix X, we let $H_n(f,X,x) \in \mathcal{P}_{2n-1}$ so that

$$H_n(f, X, x_{k;n}) = f(x_{k;n})$$
$$H_n'(f, X, x_{k;n}) = 0. \tag{1.19}$$

In this case, it can be shown $||H_n(f,T) - f||_\infty \to 0$ as $n \to \infty$ for all $f \in C[-1,1]$ [46, Chapter 5]. Unfortunately, this does not hold in general; in fact, for equally-spaced nodes we have the particularly bad result

$$f(x) = x$$
$$\limsup_{n\to\infty} |H_n(f, E, x)| = \infty, \quad 0 < |x| \leq 1, \tag{1.20}$$

which is discussed in [46, Chapter 6]. Controlling the derivative of the interpolation polynomial at the Chebyshev nodes appears to give sufficient control of the polynomial in order to obtain convergence for all continuous functions. Even so, while this gives convergence in the limit, it is not useful in practice because we purposefully limit the accuracy of interpolation near, but not at, interpolation nodes.

In another direction, Bernstein polynomials give up interpolation to get overall approximation. In fact, [16, 40] use Bernstein polynomials to prove the Weierstrass Approximation Theorem. The downside is that convergence to the solution is slow:

Theorem 1.5 (Error Estimate for Berstein polynomials; Theorem 1.2 in [40])
Suppose $g \in C[0,1]$ is α-Hölder with Hölder constant L and $B_n g$ is the Berstein polynomial of degree n for g; then

$$||g - B_n g||_{\infty,[0,1]} \leq \frac{3L}{2} \frac{1}{n^{\alpha/2}}, \tag{1.21}$$

and this bound in n cannot be improved.

This precludes it from being of much use in practice, especially when f is smooth.

By relaxing the condition $\deg L_n(f, X) \leq n-1$, Erdős was able to prove in [21] that,

under some conditions on X, one could prove convergence for all continuous functions by choosing p_n so that $\deg p_n = c(X)n$, with c a constant depending only on X. The extension to all matrices X is shown in [46, Theorem 2.7]. This is important in practice, because we can not always choose the interpolation nodes. It is beneficial for a method to work well independent of node location, especially if, because of instrument specifications, data collection location cannot be modified. Unfortunately, these results require function values at arbitrary points, and this is not possible in practice.

1.3 Interpolation in Higher Dimensions

Up to this point, we have only talked about methods for approximating functions on $[a, b]$; even so, many problems in science and engineering are inherently two- and three-dimensional. A review of recent methods for multivariable polynomial interpolation can be found in [22, 23]. One challenge of interpolation in higher dimensions is choosing the correct polynomial space and point distribution. Now, the fact $\dim \mathcal{P}_{n-1} = n$ makes this easy in 1D but in higher dimensions there does not appear to be a simple way to choose a multivariable polynomial space of arbitrary dimension. Naturally, this is a topic of great interest. In [23], some standard methods discussed include tensor products of univariate polynomials, Gröbner bases, and ideal interpolation schemes.

1.4 Hermite and Birkhoff Interpolation

Hermite or Birkhoff interpolation problems involve interpolating function and derivative values. Hermite interpolation consists of interpolating function and derivative values up to a certain degree at interpolation nodes. Birkhoff interpolation is more general, allowing any combination of specified function and derivative values at nodes. Hermite interpolation is well-posed and can easily be solved in 1D. This is not the case for Birkhoff interpolation, where only certain combinations ensure a unique solution [32, 35]. The

problem is even more complicated in dimension 2 and larger; see [36, 37] for a review of these topics. An additional challenge in multidimensional interpolation comes from proving error bounds and determining sufficient conditions for convergence.

1.5 Characteristics of Good Algorithms

This dissertation focuses on the development, implementation, and analysis of fast MSN methods. The ideas behind the MSN method will be discussed in the Sec. 1.6, but here we discuss good qualities that numerical algorithms should have, especially algorithms for approximation. These are high-order convergence, low computational complexity, and numerical stability.

Given a low-order method and a high-order method of similar computational cost, a faster-converging method is more effective and useful. In practice, there is always a limit to the amount of computational resources (memory, processor speed, or bandwidth), so a high-order method would be preferred as it would lead to less work overall. As mentioned before, Bernstein polynomials converge to all continuous function but do so at a slow rate. This alone does not necessarily disqualify the algorithm, but from Thm. 1.4, we know smoother functions can have better polynomial approximations. This incentivizes developing accurate approximations and algorithms to compute them.

While some methods may be of theoretical importance, algorithms will only be of practical value if there are efficient methods to compute them. The total cost should be of reasonable size, so that both the asymptotic growth ($O(\log n)$ or $O(n^3)$) and the explicit cost ($10^6 \log n$ and $\frac{2}{3}n^3$) are important. Because computational resources are always limited, asymptotics may not be as important as the prefactor hidden by Big O notation.

Finally, numerical stability is of critical importance. Almost all algorithms are implemented on computers using floating-point arithmetic, inevitably leading to small errors.

It is necessary for practical algorithms to be immune to these changes; namely, small changes in inputs should lead to small changes in outputs. The condition number quantifies how much changes in outputs come from changes in inputs; a standard reference for numerical stability is [30].

1.6 The Minimum Sobolev Norm Method

We previously showed Lagrange interpolation does not work, for there can be large oscillations in the interpolating polynomial as seen in the Runge phenomenon, while using a polynomial of higher degree allows continuous functions to be approximated arbitrarily well. By combining these observations, the Minimum Sobolev norm (MSN) method was developed: a general method for computing approximate solutions to problems with linear constraints.

The MSN method has been used to solve problems in interpolation [10], Birkhoff interpolation [12], and partial differential equations [14], For simplicity, we assume we are performing approximations using algebraic polynomials, even though theoretical work often uses trigonometric polynomials. The main idea is this: given N linear constraints and polynomials of up to degree $M(N)$ contained in V, unknown coefficients a, correct values f, and a diagonal matrix D_s with condition number $O(M^s)$, the MSN solution solves the equation

$$\min_{Va=f} ||D_s a||_2 . \tag{1.22}$$

We choose D_s so that $||D_s a||_2$ is a Sobolev norm. This implies that we seek an approximation which satisfies the linear constraints as well as having the smallest derivative norm. Here we focus on computing the minimum 2-norm solution because this dissertation investigates efficient numerical algorithms for MSN equations and we explicitly compute LQ factorizations; methods for p-norm minimization are discussed in [10, 12]. Additionally, this description is independent of dimension and node location. The parameter s

determines which derivative of the polynomial approximation we wish to control. Larger s gives more derivative control on the approximation but leads Eq. (1.22) to have higher condition numbers. Great care is required to limit the effects of these condition numbers in order to ensure convergence to the underlying solution [12].

The technical challenge of this method is to determine the explicit form of $M(N)$ to ensure convergence to the solution. The methods in [10, 12] involve the close approximation of integral kernels by polynomials. The end result is that it is sufficient to choose $M(N) = C\eta^{-1}$, where η is the minimum separation between between interpolation nodes. Although this is a theoretically optimal result, knowing from [46] that this result cannot be improved except in the constant, it is not useful in practice because the constants from [10, 12] are difficult to explicitly compute. In practice, we have found that choosing the $M(N) = 2\pi\eta^{-d}$ is sufficient, where d is the dimension of the space. These details, along with implementation issues, will be discussed more in the next section.

One advantage of the MSN method is that we do not insist on forming a square linear system. In fact, it is necessary to take enough columns (more than twice the number of rows) in order to ensure a good approximation. Choosing the proper polynomial space was a challenge mentioned in Secs. 1.3 and 1.4.

1.7 MSN Interpolation Examples

We present some results of MSN interpolation on equally-spaced nodes in single and double precision for 1D and 2D. We do this to show that the difficulty of approximating functions on equally-spaced points arises from using suboptimal methods of interpolation rather than node location. These and similar results were published in [10, 12].

We can rewrite Eq. (1.22) as

Algorithm 1 Solve MSN System using Slow, Stable Algorithm

1: **function** SLOW_MSN_SOLVE(f,V,D_s) ▷ Solve $\min_{Va=f} ||D_s a||_2$.
2: Compute $P_1 L_1 Q_1 = V$ using an LQ factorization based on QRCP.
3: Determine permutation Π such that $Q_1 D_s^{-1}\Pi$ has decreasing column norms.
4: Compute the SVD: $U\Sigma V^* = Q_1 D_s^{-1}\Pi$; only U is stored.
5: Compute $P_2 L_2 Q_2 = U^* Q_1 D_s^{-1}\Pi$.
6: Solve $L_1 z = P_1^* f$.
7: Solve $L_2 y = P_2^* U^* z$.
8: $a = D_s^{-1}\Pi Q_2^* y$
9: **return** a
10: **end function**

$$\begin{aligned} &\min_{VD_s^{-1}x=f} ||x||_2 \\ &a = D_s^{-1}x. \end{aligned} \tag{1.23}$$

In order to compute the MSN solution, we must compute the minimum norm solution from Eq. (1.23). To do this, we must compute an LQ factorization of VD_s^{-1}, where L is a lower triangular matrix and Q is orthogonal. As previously mentioned, large s leads to greater derivative control but also gives VD_s^{-1} high condition number. Because of this, the standard pivoted LQ factorization based on QR with Column Pivoting is insufficient. A Rank-Revealing QR factorization based on [27] would be better, but an implementation is not readily available so we use another method presented here and described in [12]; see Alg. 1.

The unique feature of the algorithm may be Lines 4 and 5. Clearly, VD_s^{-1} is badly column-scaled. LQ factorizations can deal with poor row-scaling but not poor column-scaling. We compute the singular value decomposition $U\Sigma V^* = Q_1 D_s^{-1}\Pi$ in Line 4 and see $U^* Q_1 D_s^{-1}\Pi \approx \Sigma V^*$ to machine precision. This ensures we can accurately compute the pivoted LQ factorization $P_2 L_2 Q_2 = U^* Q_1 D_s^{-1}\Pi$ in Line 5. Thus, U is a preconditioner for numerical stability, showing that we can safely convert poor column-scaling to poor row-scaling. Using Alg. 1, the effective condition number of this problem appears to be that of V and not VD_s^{-1}.

Looking at the algorithm, we see two pivoted LQ factorizations and one SVD are required. Because we have N interpolation requirements and cN columns, this gives us $O(N^3)$ floating-point operations and $O(N^2)$ units of memory. At first glance, this does not seem too bad. If we are in dimension d with n^d tensor grid points, then $N = n^d$ and we require $O(n^{3d})$ flops and $O(n^{2d})$ units of memory. While these costs may be acceptable for $d = 1$ and bearable for $d = 2$, when $d = 3$ this is too great. Parallel computation would not be of much use here because the communication required for pivoted LQ and the SVD would cause the entire process to be extremely slow, although there has been recent work in reducing the communication cost in pivoted QR factorizations [17]. In order for these algorithms to be used when solving large, difficult problems, we need to investigate other methods. Similar costs arise when solving differential equations and this necessitates fast, structured algorithms. When developing fast algorithms, it is critical that we are able to convert the poor columns scaling to poor row scaling. The inherent structure of the linear system allows us to do this using careful factorizations.

We present some examples of MSN interpolation and Birkhoff interpolation. The functions we approximate are

$$\begin{aligned} f(x) &= \frac{1}{1+25x^2} \\ g(x,y) &= \frac{1}{1+25(x^2+y-0.3)^2} + \frac{1}{1+25(x+y-0.4)^2} \\ &\quad + \frac{1}{1+25(x+y^2-0.5)^2} + \frac{1}{1+25(x^2+y^2-0.25)^2} \\ h(x) &= g(x,-0.96). \end{aligned} \tag{1.24}$$

Naturally, f is the usual Runge function. Here, g is a 2D function with Runge functions on one line, one circle, and two parabolas.

We remember machine precision is $2^{-23} \approx 1.2 \times 10^{-7}$ in single precision and $2^{-52} \approx 2.2 \times 10^{-16}$ in double precision. This is the smallest relative error that we could expect for

any nonzero result. All of the plots show results for $||f-p||_\infty / ||f||_\infty$ for true function f and approximation p. The sup-norm is approximated by sampling the function at a large number of locations and taking the maximum.

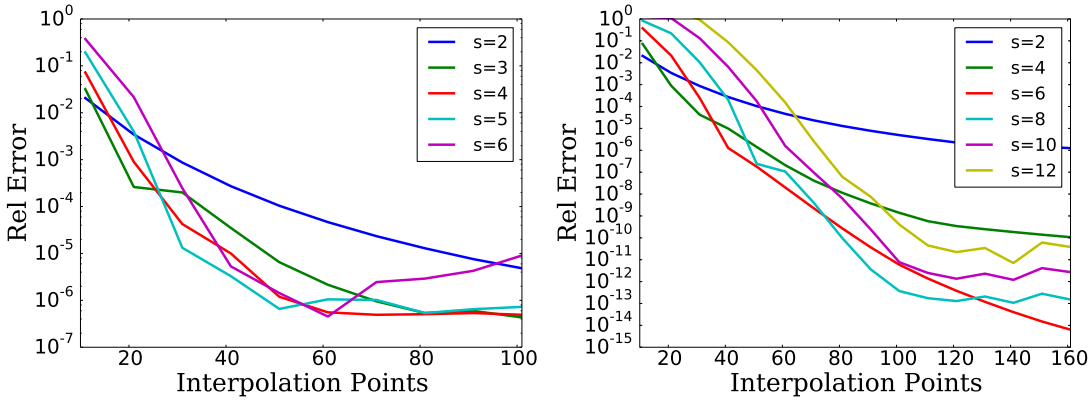
The results for interpolating f are shown in Fig. 1.2. For single precision, all error curves decay toward 10^{-7} as we increase the number of points. The main exception is for $s = 6$, which starts to increase around 60 points. We believe this occurs because of rounding error. This would also make sense given for $s \in \{3, 4, 5\}$, the error curves hover close to 10^{-6}. We see a similar results for double precision. In this case, the beginnings of the U-shaped error curve seem present for $s \in \{8, 10, 12\}$.

In Fig. 1.3, we have the results of MSN Birkhoff interpolation in 1D for h. Although the error curve for $s = 2$ in single and double precision hovers around 10^{-3}, the other error curves decay to machine precision. The beginning of a U-shaped error curve may be seen for $s \in \{4, 5\}$ in single precision and $s \in \{10, 12\}$ in double precision. The errors are low, although they may be slightly larger than those we see in regular MSN interpolation. This could stem from the fact the condition number is inherently larger for Birkhoff interpolation than for interpolation.

In Fig. 1.4, we have results for MSN Birkhoff interpolation in 2D for g from Eq. (1.24). In every case the error decreases with increasing data except for $s = 5$ with single precision. In this case, we may start to see the beginning of the effects of roundoff error. The challenge for 2D problems is the long time required to run Alg. 1.

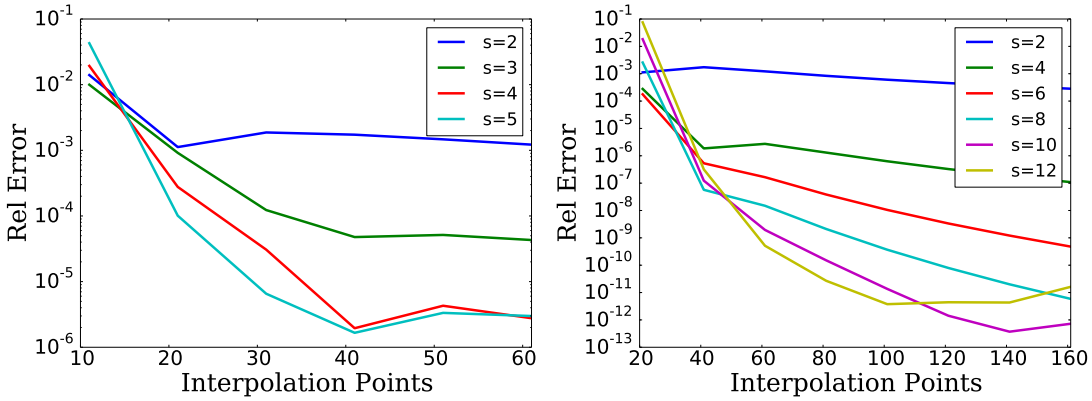
1.8 Dissertation Outline

As we noted above, the slow methods for solving problems using MSN become difficult in 2D and practically impossible in 3D due to memory requirements and flop count. With the eventual desire to use MSN to solve 3D PDEs, we will need to take advantage of *everything* we can. Keeping this in mind, the focus of this dissertation will be developing



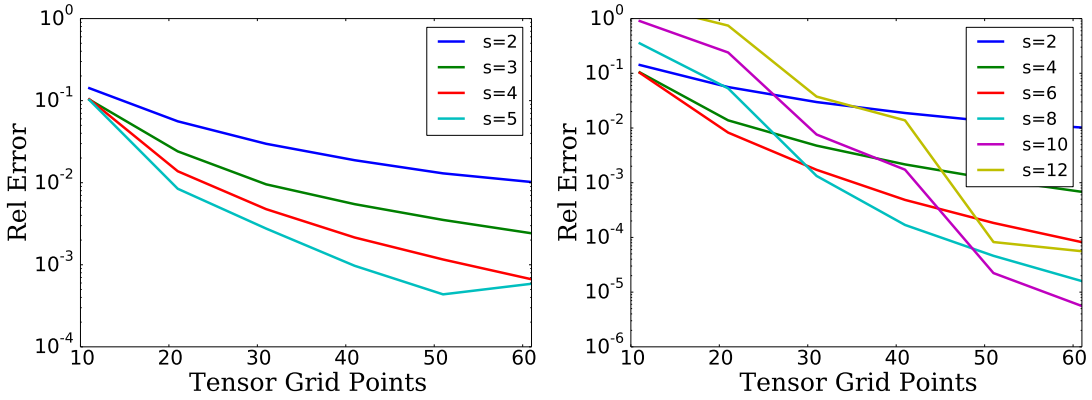
(a) Single Precision (b) Double Precision

Figure 1.2: Relative error results for MSN interpolation on equally-spaced points on the function $f(x)$ from Eq. (1.24) for various s values using single and double precision.



(a) Single Precision (b) Double Precision

Figure 1.3: Relative error results for MSN birkhoffinterpolation on equally-spaced points using both function and derivative values on the function $h(x)$ from Eq. (1.24) for various s values using single and double precision.



(a) Single Precision (b) Double Precision

Figure 1.4: Relative error results for MSN birkhoffinterpolation on equally-spaced tensor-grid points using both function and derivative values on the function $g(x)$ from Eq. (1.24) for various s values using single and double precision.

fast algorithms for solving interpolation and differential equations using the MSN method on Chebyshev nodes and express our solution in a Chebyshev polynomial basis. We review notation conventions and structured matrices in Chapter 2. In Chapter 3, we review some of the properties of Chebyshev-Vandermonde matrices which arise when developing these fast algorithms. Matrix factorizations important for interpolation problems are discussed in Chapters 4 and 5. Using these factorizations, we present examples of MSN approximation in Chapter 6. Next, we present new proofs showing that our fast methods will converge to the solution under minimal smoothness assumptions of the underlying function in Chapter 7. We investigate fast algorithms for Boundary Value Problems for ODEs in Chapter 8.

In the Chapter 9, we discuss results related to randomized low-rank approximations, unrelated to the previous work. Some of this was discussed in [26] but more details and examples will be shown here.

1.9 Algorithms Similar to the MSN Method

The ideas pursued in this dissertation are similar to those used by Chebfun [20], a software package in MATLAB [28] which attempts to have the "feel" of symbolic software with the speed of numerics. The book Approximation Theory and Approximation Practice [47] uses Chebfun to introduce the field of Approximation Theory. Here, we focus on investigating fast algorithms based on values computed on Chebyshev polynomial roots. This is similar to the fast algorithms present in Chebfun, which computes values on the Chebyshev polynomial extrema. The book Exploring ODEs [48] also uses Chebfun to introduce advanced differential equation topics. The methods in [48] are built on the work from [19, 4, 51] and are incorporated into Chebfun. Although spectral methods are well-known [5], [4] *adds* additional rows to the square linear system to impose boundary or other requirements instead of replacing rows. Naturally, this requires increasing the degree of the approximation. The work presented here does not force a square system, which allows us to add a finite number of additional requirements which do not affect the asymptotic complexity of the overall algorithm. From [48, Appendix A], it appears that Chebfun uses standard dense linear algebra algorithms to solve its ODEs. This is unfortunate, because the linear systems arising from ODEs are highly structured when approximated on Chebyshev nodes. This dissertation will show this structure and construct associated fast algorithms. The work here could be used to speedup the Chebfun ODE solver.

Chapter 2

Notation Convention, Structured Rotations, and Kronecker Products

We begin with some notation conventions before reviewing standard orthogonal matrices and looking at the Kronecker product and how it affects fast matrix-vector multiplication. One standard reference for matrix-related topics is [25]. These topics will then be used to look at matrix factorizations of Chebyshev-Vandermonde (C-V) matrices in Chapters 4 and 5.

Chapter 3

Properties of Chebyshev-Vandermonde Matrices

In this chapter we discuss properties of Chebyshev polynomials and Chebyshev-Vandermonde (C-V) matrices that we will use in later chapters to develop fast algorithms for solving interpolation and differential equations.

Chapter 4

C-V Matrices and Factorizations for 1D Interpolation

Using structured matrices from Chapter 2, we investigate Chebyshev-Vandermonde matrices in 1D and use this information to construct fast algorithms for interpolation problems. These factorizations will allow us to quickly compute the MSN solution to interpolation and differential equations (discussed later in Chapter 8).

Chapter 5

C-V Matrices and Factorizations for Interpolation in Higher Dimensions

After working through 1D interpolation in Chapter 4, we now turn our attention to interpolation problems in higher dimensions.

Chapter 6

Examples of MSN Function Interpolation

In this chapter we present results using the fast MSN algorithms. We begin by interpolating smooth functions before attempting to interpolate rough functions (discontinuous functions or functions with infinite derivatives). Unless otherwise stated, all computations will be performed in double precision. In 1D, we compare the results with Lagrange interpolation on Chebyshev nodes. In 2D, we will compare MSN with interpolation on a tensor grid of Chebyshev nodes; this is computed using the 2D IDCT. When comparing against rough functions, we use some standard filters (discussed below).

Chapter 7

Interpolation Convergence Proofs

This chapter works through the convergence proofs for MSN interpolation on Chebyshev nodes. We focus on interpolating up to degree $2n$.

Chapter 8

Fast Algorithms for ODEs

In this chapter we shift from interpolation problems to differential equations, the main goal of the dissertation. Interpolation problems look at function and derivative constraints at points of interest. Naturally, linear differential equations involve linear combinations of function and derivative values.

It is well-known that all differential equations can be rewritten to into first-order system [3, 2]. While this is not always done, especially for finite difference [34] or finite element methods [6], we do in this case. This was also done when solving partial differential equations with MSN [14].

Chapter 9

Stopping Criterion for Randomized Low-Rank Approximations

There has been work in recent years to understand structured matrices: matrices with off-diagonal blocks that are (or can be approximated as) low-rank. The goal is to develop methods which allow us to compute matrix-vector products and matrix inverses faster than standard algorithms; that is, multiplication in $O(n \log^\beta n)$ flops and inversion in $O(n^\alpha \log^\beta n)$ flops for $\alpha \in [1, 2]$ and β small. Another benefit is reduced storage requirements, frequently $O(n \log^\beta n)$. The simplest of these are banded matrices, but also include Sequentially Semi-Separable matrices [15], Hierarchically Semi-Separable (HSS) matrices [9, 11], H-matrices [29], and others. Recent work involving randomized HSS construction can be found in [38, 41, 24]. The main contribution of this chapter is related to a stochastic estimate of $||\cdot||_F$ which allows us to accurately measure the low-rank approximation and determine when it is well-approximated; portions of this material first appeared in [26]. In particular, we develop a method to compute a *relative* stopping criterion for an adaptive low-rank approximation.

Throughout this chapter we assume $A \in \mathbb{R}^{m \times n}$ with rank $r \ll \min(m, n)$. For simplicity, we will assume $m \geq n$. We let $N(0, 1)$ refer to the standard normal distribution

with mean 0 and variance 1. Finally, some of the notation in this chapter may conflict with those from previous chapters. While other adaptive randomized algorithms have frequently used an absolute tolerance ε, we will focus on allowing both an absolute tolerance ε_{abs} and a relative tolerance ε_{rel}.

Chapter 10

Conclusion

10.1 Discussion of Results and Future Directions for MSN

In this dissertation we investigated the structure of Chebyshev-Vandermonde matrices; by doing so, we were able to develop fast algorithms for solving problems in interpolation and ordinary differential equation boundary value problems. We showed examples and determined that the results were on par with many methods as well as, at times, producing more accurate approximations. Under certain circumstances, we were able to prove that our methods converge to the underlying solution.

There are multiple directions where this work could continue. First, we could look into implementing fast algorithms in 2D and 3D in order to see there are more advantages with MSN over standard Chebyshev interpolation for smooth functions or against standard filters for rough functions. Additionally, we could implement the ODE BVP fast solver using fast algorithms for SSS or HSS matrices. From there, we could look into solving 2D and 3D elliptic boundary value problems. In this case, it will be interesting to see if the methods we used in 1D variable coefficients here could be extended to 2D and 3D variable coefficients. The difficulty will come from the fact that in 2D, we will be

summing tensor products of three terms.

While interpolation on Chebyshev nodes is fast because of the DCT and IDCT, this could also hold in general, such as interpolating on Legendre nodes using Legendre polynomials as a basis and performing MSN in this basis. On the other hand, there is still structure in the Gram matrix $VD_s^{-2}V^*$, and it may be possible to invert this quickly by converting it into HSS form exactly for certain s values. Doing so would require knowledge of structured matrices and solvers, but should speed up general interpolation and make it useful. It would be interesting to see if this could be used in 2D interpolation.

10.2 Discussion of Results and Future Directions for Randomized Low-Rank Approximations

In Chapter 9 we investigated the blocked form of the low-rank fixed-precision problem in order to develop a stopping criterion useful for relative tolerances down to machine precision. We were able to prove asymptotically that this method would produce an accurate approximation to the Frobenius norm of a matrix, a critical part of the relative stopping criterion. The examples showed our method usually produces a sufficient number of random samples, though not too many, in order to reach the desired 2-norm error.

The stochastic error bound that we developed in the F-norm could be improved. First, it would be useful to determine if we can find a random variable $f(Ax)$ with $\mathbb{E}f(Ax) = ||A||_2^\alpha$ for some power α; that is, we want to find a way to combine random samplings of the range of a matrix in such a way that the expected value of the random variable is (some power of) the matrix 2-norm. Most, if not all, would prefer to bound the 2-norm rather than the F-norm, and yet accurately approximating the 2-norm is a challenge.

In our low-rank approximations, the final computation before returning our low-rank

approximation was computing a pivoted QR factorization. A BLAS-3 level QR with Column Pivoting is available in LAPACK [39, 1], yet it is well-known that this can fail for some matrices [31]; this has lead to research into better pivoting strategies [13, 8]. The best appears to be the Strong Rank-Revealing QR factorization from [27], yet there is no known efficient implementation of this algorithm. It would be useful to first develop a BLAS-2, and then BLAS-3, version. The main difficulty comes from the fact that the pivoting strategy is more involved and requires more communication. This required communication is particularly challenging in distributed memory machines and there has been work to trade communication for flops; see [17, 18, 33, 45] for some examples. Even without this, in problems where there is a hierarchical chain of QR factorizations, like those in Randomized HSS constructions [26], better pivoting at the lowest level would lead to smaller ranks overall. An implementation of SRRQR would likely require multiple passes: the first pass could either be a BLAS-3 QR (not pivoted) or BLAS-3 pivoted QR; the second pass could be a blocked version of the SRRQR algorithm acting on adjacent panels; a final BLAS-2 pass could be performed to check for any final pivots. While this would be complicated, it would be useful for the entire computing community as rank-revealing QR factorizations are essential.

Bibliography

[1] Edward Anderson et al. *LAPACK Users' guide*. Vol. 9. Siam, 1999.

[2] Uri M Ascher, Robert MM Mattheij, and Robert D Russell. *Numerical solution of boundary value problems for ordinary differential equations*. Vol. 13. Siam, 1994.

[3] Uri M Ascher and Linda R Petzold. *Computer methods for ordinary differential equations and differential-algebraic equations*. Vol. 61. Siam, 1998.

[4] Jared L Aurentz and Lloyd N Trefethen. "Block operators and spectral discretizations". In: *SIAM Review* 59.2 (2017), pp. 423–446.

[5] John P Boyd. *Chebyshev and Fourier spectral methods*. Courier Corporation, 2001.

[6] Susanne Brenner and Ridgway Scott. *The mathematical theory of finite element methods*. Vol. 15. Springer Science & Business Media, 2007.

[7] L. Brutman. "On the Lebesgue Function for Polynomial Interpolation". In: *SIAM Journal on Numerical Analysis* 15.4 (1978), pp. 694–704. DOI: 10.1137/0715046. eprint: https://doi.org/10.1137/0715046. URL: https://doi.org/10.1137/0715046.

[8] Tony F Chan. "Rank revealing QR factorizations". In: *Linear algebra and its applications* 88 (1987), pp. 67–82.

[9] S. Chandrasekaran, M. Gu, and W. Lyons. "A fast adaptive solver for hierarchically semiseparable representations". In: *CALCOLO* 42.3 (Dec. 2005), pp. 171–185. ISSN: 1126-5434.

[10] S. Chandrasekaran, K.R. Jayaraman, and H.N. Mhaskar. "Minimum Sobolev norm interpolation with trigonometric polynomials on the torus". In: *Journal of Computational Physics* 249.Supplement C (2013), pp. 96–112. ISSN: 0021-9991. DOI: https://doi.org/10.1016/j.jcp.2013.03.041. URL: http://www.sciencedirect.com/science/article/pii/S0021999113002192.

[11] Shiv Chandrasekaran, Ming Gu, and Timothy Pals. "A fast ULV decomposition solver for hierarchically semiseparable representations". In: *SIAM Journal on Matrix Analysis and Applications* 28.3 (2006), pp. 603–622.

[12] Shivkumar Chandrasekaran, CH Gorman, and Hrushikesh Narhar Mhaskar. "Minimum Sobolev norm interpolation of scattered derivative data". In: *Journal of Computational Physics* 365 (2018), pp. 149–172.

[13] Shivkumar Chandrasekaran and Ilse CF Ipsen. "On rank-revealing factorisations". In: *SIAM Journal on Matrix Analysis and Applications* 15.2 (1994), pp. 592–622.

[14] Shivkumar Chandrasekaran and Hrushikesh Narhar Mhaskar. "A minimum Sobolev norm technique for the numerical discretization of PDEs". In: *Journal of Computational Physics* 299 (2015), pp. 649–666.

[15] Shiv Chandrasekaran et al. "Some fast algorithms for sequentially semiseparable representations". In: *SIAM Journal on Matrix Analysis and Applications* 27.2 (2005), pp. 341–364.

[16] Philip J Davis. *Interpolation and approximation*. Courier Corporation, 1975.

[17] James W Demmel et al. "Communication avoiding rank revealing QR factorization with column pivoting". In: *SIAM Journal on Matrix Analysis and Applications* 36.1 (2015), pp. 55–89.

[18] Simplice Donfack, Laura Grigori, and Alok Kumar Gupta. "Adapting communiction-avoiding LU and QR factorizations to multicore architectures". In: *2010 IEEE*

International Symposium on Parallel & Distributed Processing (IPDPS). IEEE. 2010, pp. 1–10.

[19] Tobin A Driscoll and Nicholas Hale. "Rectangular spectral collocation". In: *IMA Journal of Numerical Analysis* 36.1 (2015), pp. 108–132.

[20] Tobin A Driscoll, Nicholas Hale, and Lloyd N Trefethen. *Chebfun guide*. 2014.

[21] P Erdos. "On some convergence properties of the interpolation polynomials". In: *Annals of Mathematics* (1943), pp. 330–337.

[22] Mariano Gasca and Thomas Sauer. "On the history of multivariate polynomial interpolation". In: *Numerical Analysis: Historical Developments in the 20th Century*. Elsevier, 2001, pp. 135–147.

[23] Mariano Gasca and Thomas Sauer. "Polynomial interpolation in several variables". In: *Advances in Computational Mathematics* 12.4 (2000), p. 377.

[24] Pieter Ghysels et al. "A robust parallel preconditioner for indefinite systems using hierarchical matrices and randomized sampling". In: *2017 IEEE International Parallel and Distributed Processing Symposium (IPDPS)*. IEEE. 2017, pp. 897–906.

[25] G.H. Golub and C.F. Van Loan. *Matrix Computations*. Matrix Computations. Johns Hopkins University Press, 2012. ISBN: 9781421408590. URL: `https://books.google.com/books?id=5U-l8U3P-VUC`.

[26] Christopher Gorman et al. "Matrix-free construction of HSS representation using adaptive randomized sampling". In: *CoRR* abs/1810.04125 (2018). arXiv: `1810.04125`. URL: `http://arxiv.org/abs/1810.04125`.

[27] Ming Gu and Stanley C Eisenstat. "Efficient algorithms for computing a strong rank-revealing QR factorization". In: *SIAM Journal on Scientific Computing* 17.4 (1996), pp. 848–869.

[28] MATLAB User's Guide. "The mathworks". In: *Inc., Natick, MA* 5 (1998), p. 333.

[29] Wolfgang Hackbusch. "A Sparse Matrix Arithmetic Based on H-Matrices. Part I: Introduction to H-Matrices". In: *Computing* 62.2 (1999), pp. 89–108.

[30] Nicholas J. Higham. *Accuracy and Stability of Numerical Algorithms*. Second. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 2002, pp. xxx + 680. ISBN: 0-89871-521-0.

[31] William Kahan. "Numerical linear algebra". In: *Canadian Mathematical Bulletin* 9.5 (1966), pp. 757–801.

[32] Samuel Karlin and John M Karon. "On Hermite-Birkhoff interpolation". In: *Journal of Approximation Theory* 6.1 (1972), pp. 90–115.

[33] Amal Khabou et al. "LU factorization with panel rank revealing pivoting and its communication avoiding version". In: *SIAM Journal on Matrix Analysis and Applications* 34.3 (2013), pp. 1401–1429.

[34] Randall J LeVeque. *Finite difference methods for ordinary and partial differential equations: steady-state and time-dependent problems*. Vol. 98. Siam, 2007.

[35] Georg Gunther Lorentz and KL Zeller. "Birkhoff interpolation". In: *SIAM Journal on Numerical Analysis* 8.1 (1971), pp. 43–48.

[36] RA Lorentz. "Multivariate Hermite interpolation by algebraic polynomials: A survey". In: *Journal of computational and applied mathematics* 122.1-2 (2000), pp. 167–201.

[37] Rudolph A Lorentz. *Multivariate Birkhoff Interpolation*. Springer, 1992.

[38] Per-Gunnar Martinsson. "A fast randomized algorithm for computing a hierarchically semiseparable representation of a matrix". In: *SIAM Journal on Matrix Analysis and Applications* 32.4 (2011), pp. 1251–1274.

[39] Gregorio Quintana-Ortı, Xiaobai Sun, and Christian H Bischof. "A BLAS-3 version of the QR factorization with column pivoting". In: *SIAM Journal on Scientific Computing* 19.5 (1998), pp. 1486–1494.

[40] Theodore J Rivlin. *An introduction to the approximation of functions*. Courier Corporation, 2003.

[41] François-Henry Rouet et al. "A distributed-memory package for dense hierarchically semi-separable matrix computations using randomization". In: *ACM Transactions on Mathematical Software (TOMS)* 42.4 (2016), p. 27.

[42] Walter Rudin et al. *Principles of mathematical analysis*. Vol. 3. 4.2. McGraw-hill New York, 1976.

[43] Carl Runge. "Über empirische Funktionen und die Interpolation zwischen äquidistanten Ordinaten". In: *Zeitschrift für Mathematik und Physik* 46.224-243 (1901), p. 20.

[44] Simon J Smith. "Lebesgue constants in polynomial interpolation". In: *Annales Mathematicae et Informaticae*. Vol. 33. 109-123. Eszterházy Károly College, Institute of Mathematics and Computer Science. 2006, pp. 1787–5021.

[45] Edgar Solomonik and James Demmel. "Communication-optimal parallel 2.5 D matrix multiplication and LU factorization algorithms". In: *European Conference on Parallel Processing*. Springer. 2011, pp. 90–109.

[46] József Szabados and Vertesi Peter. *Interpolation of functions*. World Scientific, 1990.

[47] Lloyd N Trefethen. *Approximation theory and approximation practice*. Vol. 128. Siam, 2013.

[48] Lloyd N Trefethen, Ásgeir Birkisson, and Tobin A Driscoll. *Exploring ODEs*. Vol. 157. SIAM, 2017.

[49] Lloyd N Trefethen and JAC Weideman. "Two results on polynomial interpolation in equally spaced points". In: *Journal of Approximation Theory* 65.3 (1991), pp. 247–260.

[50] P Vértesi. "Optimal Lebesgue constant for Lagrange interpolation". In: *SIAM Journal on Numerical Analysis* 27.5 (1990), pp. 1322–1331.

[51] Kuan Xu and Nicholas Hale. "Explicit construction of rectangular differentiation matrices". In: *IMA Journal of Numerical Analysis* 36.2 (2015), pp. 618–632.