Efficient and Well-Conditioned Methods for Computing Frame Approximations

by

Maosheng Guo

A Thesis Presented in Partial Fulfillment
of the Requirements for the Degree
Master of Arts

Approved April 2023 by the
Graduate Supervisory Committee:

Rodrigo Platte, Chair
Malena Espanol
Rosemary Renaut

ARIZONA STATE UNIVERSITY

May 2023

ABSTRACT

This thesis addresses the problem of approximating analytic functions over general and compact multidimensional domains. Although the methods we explore can be used in complex domains, most of the tests are performed on the interval $[-1, 1]$ and the square $[-1, 1] \times [-1, 1]$. Using Fourier and polynomial frame approximations on an extended domain, well-conditioned methods can be formulated. In particular, these methods provide exponential decay of the error down to a finite but user-controlled tolerance $\epsilon > 0$. Additionally, this thesis explores two implementations of the frame approximation: a singular value decomposition (SVD)-regularized least-squares fit as described by Adcock and Shadrin in 2021, and a column and row selection method that leverages QR factorizations to reduce the data needed in the approximation. Moreover, strategies to reduce the complexity of the approximation problem by exploiting randomized linear algebra in low-rank algorithms are also explored, including the AZ algorithm described by Coppe and Huybrechs in 2020.

TABLE OF CONTENTS

LIST OF FIGURES

Chapter 1

INTRODUCTION

1.1 Motivation

Function approximation is one of the main pillars of computational mathematics. It is the foundation for developing Partial Differential Equation (PDE) solvers, optimization solvers, and data fitting techniques, to name a few. Polynomial and Fourier bases are commonly used in one-dimension and rectangular domains due to their fast convergence and robustness.

In the case of periodic functions, Fourier bases are advantageous because of their periodicity and efficiency. In this case, expansion coefficients can be computed in logarithmic time using Fast Fourier Transform algorithms (FFTs) that rely on equally spaced data points.

Polynomials are less restrictive and can be used without periodicity assumptions with similar convergence rates and fast algorithms but require non-uniform approximation points. In rectangular domains, the choices of points that allow stable approximations are well-known (e.g., tensor-product of 1-D Chebyshev points)[12]. However, polynomials fail spectacularly when the data is equispaced or randomly sampled [10]. Moreover, in the case of complex geometries, optimal nodes are not known analytically and are computationally expensive to estimate.

Traditionally, Fourier and polynomial approximations are computed by interpolating the data or, more generally, via discrete least squares. However, interpolation is very sensitive to data placement, as illustrated by the Runge phenomenon in 1-D when equispaced nodes are used (described in Chapter 2). Least squares approximation, on the other hand, is less restrictive on point placements but will require a large number of points (relative to the approximation degree) to compensate for a poor choice of sampling locations. In one-dimension, stable computation of a polynomial approximation requires $O(d^2)$ equispaced samples, where d is the approximation degree.

Frame approximation has been proposed as an alternative to these traditional techniques in recent years. For instance, Fourier frames have been studied in [8], and more recently, polynomial frames have been proposed in [1]. In particular, Fourier extensions have been extensively used to solve PDEs[4, 9].

1.2 Organization and Main Contributions

This thesis combines ideas presented in three articles: [3], [5], and [11]. The first establishes the theory of convergence for polynomial frames. The second proposes a fast and randomized algorithm for solving least squares approximations of rank-deficient matrices. The third introduces pivoted QR factorizations for estimating good sampling points for interpolation.

In Chapter 2, we review the theory presented in [3] and replicate several numerical experiments demonstrating the approximating properties of polynomial frames. In Chapter 3, the node selection algorithm from [11] is extended to frames, where we propose a column and row selection method that leverages QR factorizations to reduce the data needed in the approximation while maintaining the decay rate and conditioning established by Adcock and Shadrin [2].

Chapter 4 reviews the AZ algorithm proposed in [5]. That article focused on Fourier frames. In Chapter 5 we implement and test the algorithm for polynomial frames and demonstrate its computational efficiency with several numerical experiments. In particular, we propose a method for AZ implementation on polynomial

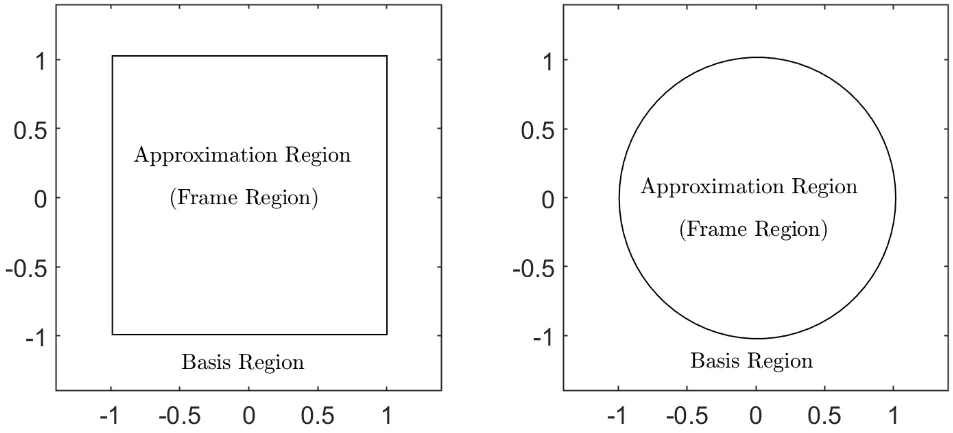

Figure 1.1: Examples of basis and frame regions generated with $\gamma = 1.4$.

frames, in which a set of composite nodes are used to ensure the method is well-conditioned.

For completeness, we include computations for Fourier frame approximations in Chapter 6. Final remarks and future work are presented in Chapter 7.

1.3 Introduction to Frames

As described by Adcock and Huybrechs[1], frames are a generalization of bases that allow for redundancy amongst the generating elements. They are usually used in the context of modern signal and image processing. However, the use of frames in function approximation is more uncommon. Thus, this paper aims to consider frames in this context. Specifically, we will examine truncated frames used in the approximation of functions on the interval $[-1, 1]$ and the square $[-1, 1] \times [-1, 1]$ using orthogonal bases on an expanded domain based on a fixed $\gamma > 1$, which determines the scaling between the basis region and the frame region, as shown in Figure 1.1. Note that when restricted to the smaller domain, the set of orthogonal basis might become ill-conditioned or even linearly dependent.

The approximation is then computed by solving a regularized least-squares problem, with user-controlled regularization parameter (tolerance) $\epsilon > 0$. Even though the frame functions will be ill-conditioned and not orthogonal on our domains of interest, in terms of motivation, we will demonstrate that there are several feasible methods from which approximations can be computed with desirable properties.

1.4 General Properties of Frames

Let $\Phi = \{\phi_n\}_{n\in\mathbb{N}}$ be an orthonormal basis element of a Hilbert space H. It then follows that Φ has two key properties. First, the direct representation of any $f \in H$ in the basis using the inner product on H is given by:

$$f = \sum_{n\in\mathbb{N}} \langle f, \phi_n \rangle \phi_n, \quad \forall f \in H,$$

where the infinite sum converges in H. Also, Φ gives rise to Parseval's identity, which indicates that the Hilbert space norm of f is the $\mathcal{L}^2$-norm of its coefficients:

$$||f||^2 = \sum_{n\in\mathbb{N}} |\langle f, \phi_n \rangle|^2, \quad \forall f \in H.$$

If the coefficients $\{\langle f, \phi_n \rangle\}_{n\in\mathbb{N}}$ are known or have been computed, we can approximate f with a finite expansion:

$$f \approx \sum_{n=1}^{N} \langle f, \phi_n \rangle \phi_n.$$

On the other hand, $\Phi = \{\phi_n\}_{n\in\mathbb{N}}$ is called a frame for H if it satisfies the frame condition

$$A||f||^2 \leq \sum_{n\in\mathbb{N}} |\langle f, \phi_n \rangle|^2 \leq B||f||^2, \quad \forall f \in H$$

for constants $A, B > 0$. The optimal A and B. i.e. the largest possible A and smallest possible B, are referred to as frame bounds.

Comparing frames and orthonormal bases, there are some key differences. The frame elements ϕ_n are not orthogonal in general. Furthermore, while the frame condition implies span(Φ) is dense in H, Φ is not necessarily a basis. Essentially, a frame is typically redundant.

With this in mind, we will first start with the 1D formulation. Let $\phi_i(x)$ be an orthonormal basis on $[-1,1]$. We then define the frame approximation on $[-\gamma,\gamma]$ as $\{\psi_i\}_{i=0}^{\infty}$, where the ith such function is given by

$$\psi_i(x) = \phi_i(x/\gamma)/\sqrt{\gamma}, \quad x \in [-\gamma,\gamma].$$

Let $m, n \geq 0$ and consider a function $f \in C([-1,1])$. Our aim is to compute a frame approximation to f of the form

$$f \approx \hat{f} = \sum_{i=0}^{n} \hat{c_i}\psi_i \in \mathbb{P}_n,$$

for suitable coefficients $\hat{c_i}$, where $\mathbb{P}_n$ denotes the space of all polynomials with degree n or less. In order to do this, we perform a least-square fit, such that

$$\hat{c} = (\hat{c}_i)_{i=0}^{n} \in \underset{c\in\mathbb{C}^{n+1}}{\operatorname{argmin}} \|Ac - b\|_2,$$

where

$$A = \alpha(m)(\psi_j(x_i))_{i,j=0}^{m,n} \in \mathbb{C}^{m\times n},$$

$$b = \alpha(m)(f(x_i))_{i=0}^{m} \in \mathbb{C}^{m},$$

and $\alpha(m)$ is a normalization factor that is included for convenience. Notice that in the frame case, A might be rank deficient, which will result in multiple solutions.

As for the 2D formulation, since we are working within the square $[-1,1]\times[-1,1]$, it is sufficient to formulate the 1D version of the A matrix first, then apply the Kronecker product to A with itself to obtain the 2D A matrix.

1.5 SVD-Regularized Frame

In frame approximation, the least-squares problem described in Section 1.2 is often ill-conditioned for large n, even when $m >> n$. However, there are a number of ways to regularize the problem, such as the ϵ-truncated SVD and column-pivoted QR decomposition. To start with the ϵ-truncated SVD decomposition, suppose that $A = U\Sigma V^*$, where $\Sigma = diag(\sigma_0, \ldots, \sigma_n) \in \mathbb{R}^{m\times n}$ is the diagonal matrix of singular values. Then $\hat{c}$ is given by

$$\hat{c} = V\Sigma^{\dagger}U^*b,$$

where $\dagger$ denotes pseudoinverse. Given $\epsilon > 0$, we define Σ^{ϵ} as

$$\Sigma^{\epsilon} = \begin{cases} \sigma_i & \sigma_i > \epsilon \\ 0 & \text{otherwise} \end{cases},$$

and similarly, we define $\Sigma^{\epsilon,\dagger}$ as

$$\Sigma^{\epsilon,\dagger} = \begin{cases} 1/\sigma_i & \sigma_i > \epsilon \\ 0 & \text{otherwise} \end{cases}.$$

We can then define the $\epsilon-$regularized approximation of $\hat{c}$ as $\hat{c}^{\epsilon} = V\Sigma^{\epsilon,\dagger}U^*b$ and the corresponding approximation of f as

$$\hat{f}^{\epsilon} = \sum_{i=0}^{n} \hat{c}_i^{\epsilon}\psi_i.$$

Thus, the overall approximation procedure can be defined as the mapping

$$\mathcal{P}_{m,n}^{\epsilon,\gamma} : C([-1,1]) \to C([-1,1]), f \to \hat{f}^{\epsilon} = \sum_{i=0}^{n} \hat{c}_i^{\epsilon}\psi_i,$$

where

$$\hat{c} = (\hat{c}_i)_{i=0}^{n} = V\Sigma^{\epsilon,\dagger}U^*b, \quad b = \alpha(m)(f(x_i))_{i=0}^{m}.$$

Chapter 2

POLYNOMIAL FRAMES

In this chapter, we review the main result presented in [3] and replicate many of its numerical results.

2.1 Introduction to Polynomial Frames

To begin, one of the main problems with the choice of equispaced points with polynomial interpolants can be demonstrated through Runge's function, which is defined as $f(x) = 1/(1 + 25x^2)$.

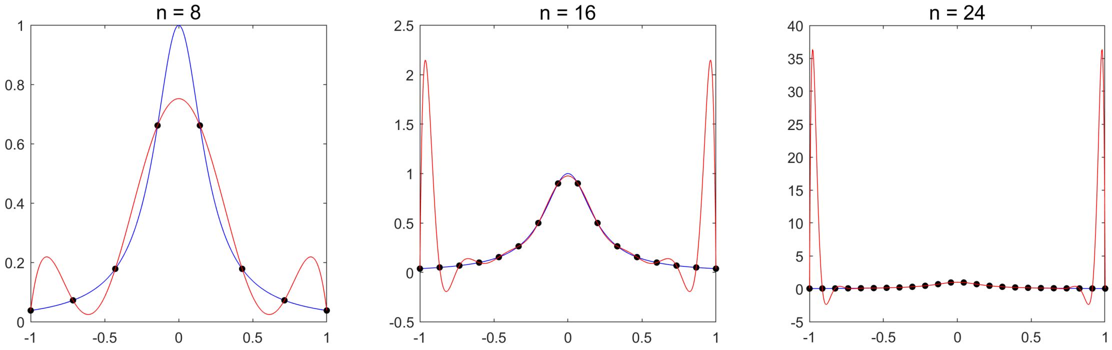

Figure 2.1: Plots of $f = 1/(1 + 25x^2)$ constructed with n equispaced nodes.

As the number of equispaced nodes increases, the interpolant becomes more accurate throughout most of the interval; however, the oscillations near the endpoints of the interval become more extreme, making the interpolant unstable. This means that even with a higher degree interpolant, the error is not necessarily small throughout the interval if the interpolation nodes are not properly clustered. This behavior is known as Runge's phenomenon. This is an illustration of a more general problem in function approximation in equally spaced points that was pointed out in [10]. The main result in that article states that under certain assumptions geometric convergence is unstable when equispaced nodes are used.

To further explore this problem, we consider families of mappings

$$\mathcal{R}_m : C([-1,1]) \to C([-1,1]),$$

where for each $m \geq 1$ and $f \in C([-1,1]), \mathcal{R}_m(f)$ depends only on the values $\{f(x_i)\}_{i=0}^m$ of f on the equispaced grid $\{x_i\}_{i=0}^m$. Then, the absolute condition number of $\mathcal{R}_m$ can be defined as

$$\kappa(\mathcal{R}_m) = \underset{f \in C([-1,1])}{sup} \underset{\delta \longrightarrow 0^+}{lim} \underset{\substack{h \in C([-1,1]) \\ 0<\|h\|_{m,\infty} \leq \delta}}{sup} \frac{\|\mathcal{R}_m(f+h) - \mathcal{R}_m(f)\|_{[-1,1],\infty}}{\|h\|_{m,\infty}},$$

which is essentially the change in the mapping as a result of small perturbations in the data.

Now we can state the impossibility theorem from [10], which generalizes Runge's phenomenon:

Theorem 2.1.1 (The impossibility theorem) *Let $E \subset \mathbb{C}$ be a compact set containing $[-1,1]$ in its interior and $\{\mathcal{R}_m\}_{m=1}^\infty$ be an approximation procedure based on equispaced grids of $m+1$ points such that, for some $C, \rho > 1$ and $1/2 < \tau \leq 1$, we have*

$$\|f - \mathcal{R}_m(f)\|_{m,\infty} \leq C\rho^{-m^\tau} \|f\|_{E,\infty}, \quad \forall m \in \mathbb{N}, \quad f \in B(E).$$

Then the condition number $\kappa(\mathcal{R}_m)$ satisfies

$$\kappa(\mathcal{R}_m) \geq \sigma^{m^{2\tau-1}},$$

for some $\sigma > 1$ and all sufficiently large m.

As shown in the impossibility theorem, the approximation problem is intrinsically difficult since any method that offers exponential rates of convergence in m for all such functions in a fixed, but arbitrary region of the complex plane must necessarily be exponentially ill-conditioned. Furthermore, the best rate of convergence achievable by a stable method is necessarily subgeometric($\tau = 1/2$).

Many methods have been proposed to overcome Runge's phenomenon by stably and accurately approximating analytic functions from equispaced nodes, but full mathematical explanations have been lacking. Thus, we seek to fully explore the impossibility theorem using polynomial frame approximation. [1]

In particular, the polynomial frame will approximate a function f on $[-1, 1]$ and $[-1, 1] \times [-1, 1]$ using orthogonal polynomials on an expanded domain based on some fixed $\gamma > 1$. The approximation is then computed by solving a regularized least-squares problem, with user-controlled regularization parameter(tolerance) $\epsilon > 0$. Note that the polynomials will be ill-conditioned and not orthogonal on the domains of interest, but they will circumvent the assumptions of the impossibility theorem. We will use orthonormal Legendre polynomials here, although other orthogonal polynomials such as Chebyshev polynomials will suffice as well.

2.2 Numerical Results

Applying the regularization methods described in Sections 1.3 and 1.4, we can estimate the maximal behavior of the polynomial frame bounded at equispaced nodes. Let's first define the quantity

$$C(m, n, \gamma, \epsilon) = \sup\{\|p\|_{[-1,1],\infty} : p \in \mathbb{P}_n, \|p\|_{m,\infty} \leq 1, \|p\|_{[-\gamma,\gamma],\infty} \leq 1/\epsilon\},$$

which provides an upper bound for the $\kappa(\mathcal{P}^{\epsilon,\gamma}_{m,n})$ and is bounded with linear oversampling between m and n[3]. With this, we will state our main result without proof:

Main result: Let $\epsilon > 0$, $\gamma \geq 1$, $c > 1$, and $m \geq n \geq 1$ be such that

$$C(m, n, \gamma, \epsilon) \leq c,$$

then the polynomial frame approximation $\mathcal{P}^{\epsilon',\gamma}_{m,n}$ with $\epsilon' = \epsilon(n+1)/\sqrt{\gamma}$ satisfies

$$\kappa(\mathcal{P}^{\epsilon',\gamma}_{m,n}) \leq c\sqrt{m+1},$$

and for any $f \in C([-1,1])$,

$$\|f - \mathcal{P}^{\epsilon',\gamma}_{m,n}(f)\|_{[-1,1],\infty} \leq 2c\sqrt{m+1}\underset{p\in\mathbb{P}_n}{inf}\{\|f-p\|_{[-1,1],\infty} + (n+1)\epsilon\|p\|_{[-\gamma,\gamma],\infty}\}.$$

We can see that the error between the function f and its frame is bounded by the greatest lower bound of the ϵ- regularized infinity norm functional and m. To how the polynomial frame behaves, we will run three different experiments.
In the first experiment, we will see how the approximation error decays with we are approximating an analytic function, which in this case is $f(x) = 1/(1+x^2)$.

We can see in Figure 2.2 that the approximation error decreases exponentially until a certain level (a fractional power of ϵ), at which it settles down and oscillates. Additionally, with a smaller ϵ, greater oversampling is needed to meet the tolerance. To further investigate this decay, we will try to approximate functions that are not sufficiently analytic.

In Figure 2.3, we see that the approximation error for the functions that are not sufficiently analytic decreases exponentially until a certain breakpoint (determined by the poles of the functions and γ), at which it decays at a superalgebraic rate.
For the last experiment in this section, we will try to approximate the oscillatory

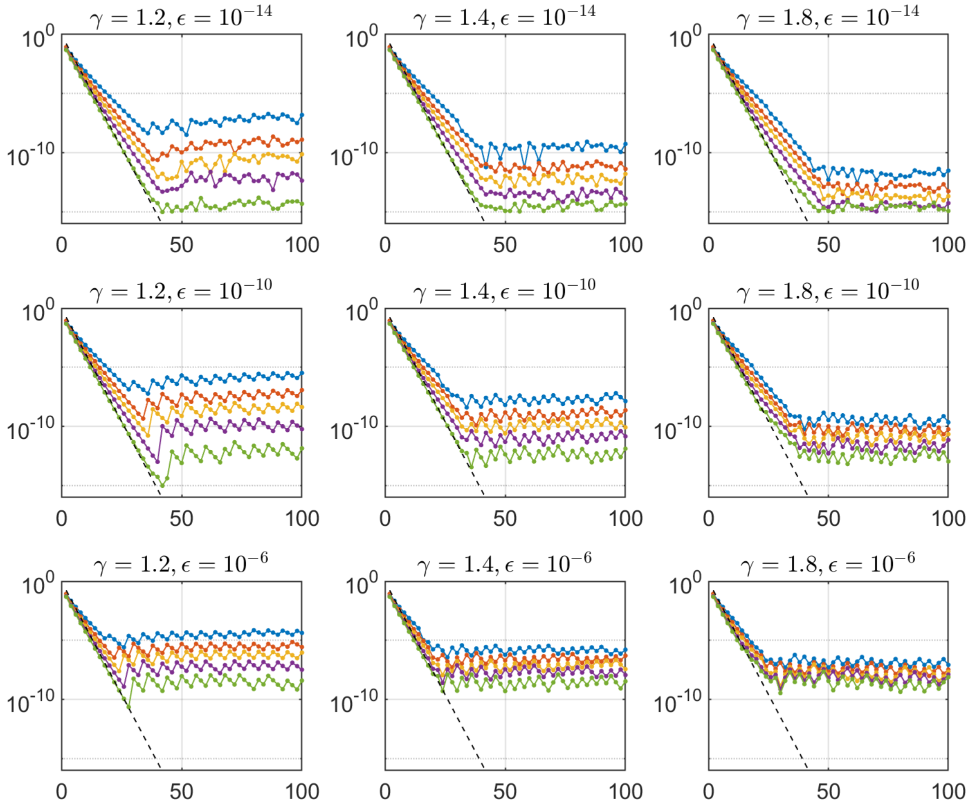

Figure 2.2: SVD-regularized frame approximation error versus n for approximating the function $f(x) = 1/(1+x^2)$ via $\mathcal{P}^{\epsilon,\gamma}_{m,n}$, where $m/n = \eta$, using various different values of η, γ and ϵ(Blue: $\eta = 1$, orange: $\eta = 1.25$, yellow: $\eta = 1.5$, purple: $\eta = 2$, green: $\eta = 4$). The dashed line shows the quantity θ^{-n}, where $\theta = \sqrt{2} + 1$.

function $f(x) = e^{i\omega\pi x}$ for various different values of ω. This function can be tough to approximate with equispaced nodes, due to its extreme growth on the imaginary axis for large ω.

As we see in Figure 2.4, the approximation error is order one until a minimum value of $\pi\gamma\omega$ (the resolution power) is met. After this point, the function begins to be resolved and the error decreases rapidly according to ϵ.

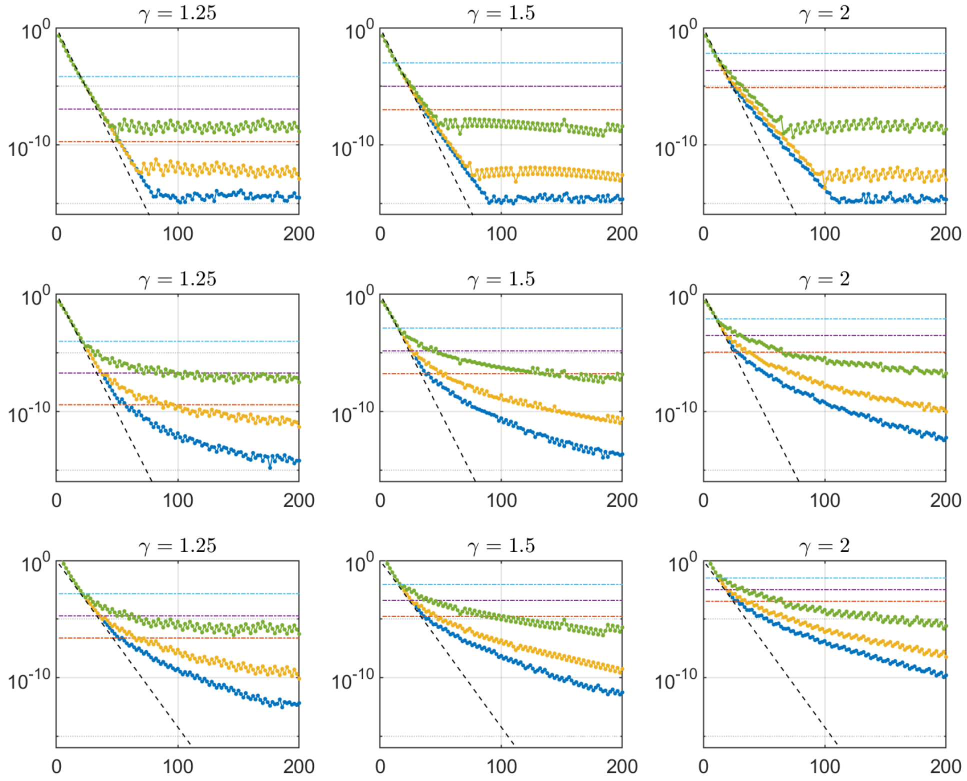

Figure 2.3: SVD-regularized frame approximation errors versus n for approximating the functions $f_1(x) = 1/(1+4x^2)$ (top-row), $f_2(x) = 1/(10-9x)$(middle-row) and $f_3(x) = 25\sqrt{(9x^2-10)}$(bottom-row) via $\mathcal{P}^{\epsilon,\gamma}_{m,n}$, where m/n =4, using different values of γ and ϵ (green: $\epsilon = 10^{-6}$, yellow $\epsilon = 10^{-10}$, blue $\epsilon = 10^{-14}$) . The dot-dashed lines show the breakpoints in each case and dashed line shows the quantity θ^{-n}.

Additionally, for 2D approximation, we have Figure 2.5. Compared to the first experiment, We can see that while the convergence is still geometric for the 2D approximations up to the user-selected tolerance ϵ, the convergence rate is generally much slower due to the added dimension and the computation time is much longer as a result.

Overall, we see that the SVD-regularized polynomial frame is very robust in terms of function approximation, although this formulation is prone to overfitting, which we will try to mitigate with the QR-Regularized Polynomial Frame.

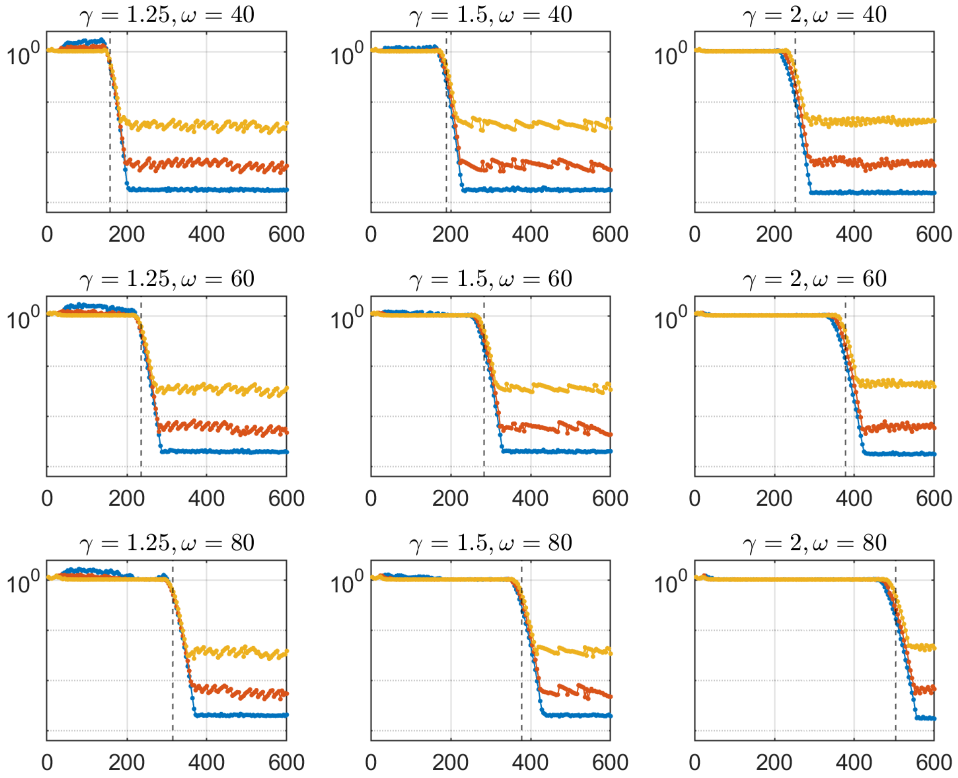

Figure 2.4: SVD-regularized frame approximation errors versus n for approximating the functions $f(x) = \exp(i\omega\pi x)$ via $\mathcal{P}_{m,n}^{\epsilon,\gamma}$, where $m/n = 4$, using various different values of γ and ϵ.(Blue: $\epsilon = 10^{-14}$, orange: $\epsilon = 10^{-10}$, yellow: $\epsilon = 10^{-6}$)

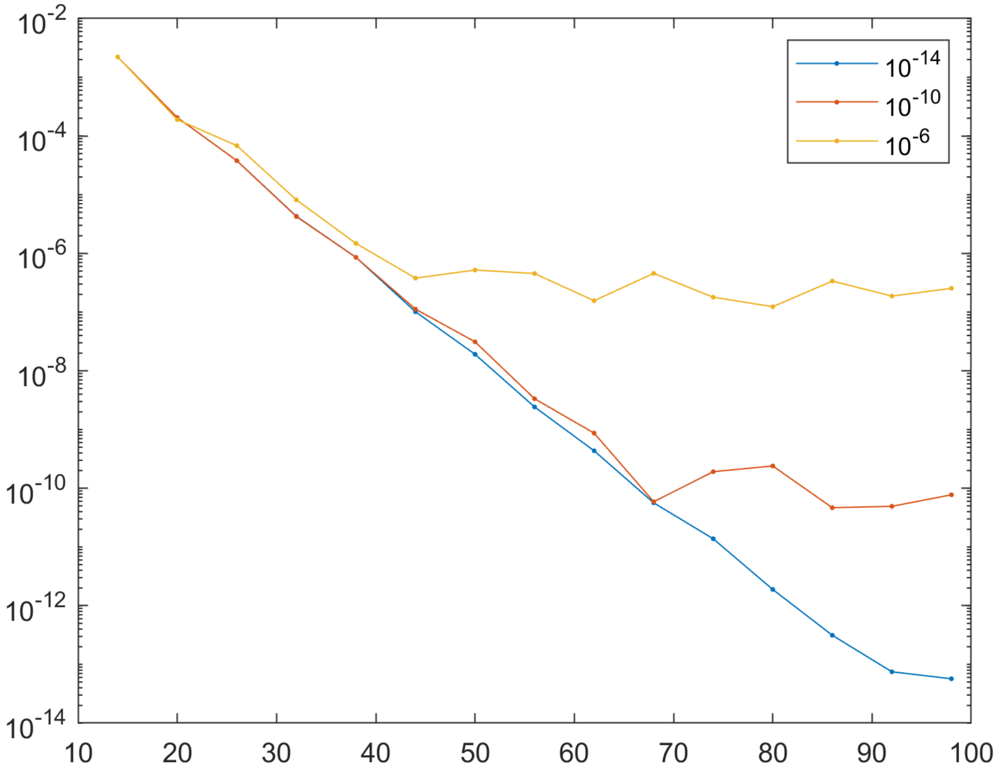

Figure 2.5: 2D SVD-regularized frame approximation error using SVD versus n for approximating the function $f(x, y) = 1/(1 + x^2 + y^2)$ via $\mathcal{P}^{\epsilon,\gamma}_{m,n}$, where $m/n = 2$, $\gamma = 1.4$ using various different values of ϵ(Blue: $\epsilon = 10^{-14}$, orange: $\epsilon = 10^{-10}$, yellow: $\epsilon = 10^{-6}$).

Chapter 3

QR-REGULARIZED FRAME

Here we address one of the main contributions of this thesis. For the original least-squares problem described in Section 1.2, another way to circumvent the ill-conditioning of A is to perform column-pivoted QR decomposition on A such that

$$AP = QR,$$

where P is the permutation matrix, Q is orthogonal and R is an upper-triangular square matrix. Then, based on the user-selected tolerance ϵ, R can be truncated via its diagonal entries, and Q can also be truncated accordingly. This helps us select certain frame polynomials while keeping the resulting submatrix well-conditioned. Defining the truncated matrices as Q_1 and R_1, we can simplify the least-squares problem even further by performing column-pivoted QR on Q_1^T, resulting in the permuted matrix $\tilde{Q}_1$. We can then truncate $\tilde{Q}_1$'s rows. Similarly to the first truncation, this is equivalent to picking certain points in the equispaced grid, but keeping the resulting submatrix well-conditioned. One example of these points is shown in Figure 3.1:

The least-squares problem is now reduced to

$$\hat{d} = \left(\hat{d}_i\right)_{i=0}^{p} \in \underset{d \in \mathbb{C}^{p+1}}{argmin} \|\tilde{Q}_1 d - \hat{b}\|_2, \quad p < n,$$

where $\hat{b}$ is obtained by permuting and truncating b similarly to Q_1. Note that $\tilde{Q}_1$ is no longer orthogonal due to the truncations.

We point out that the idea of using pivoted QR factorizations for node selection in polynomial approximations was introduced in [11] and further developed in [7]. Additionally, the points selected are akin to Fekete points, since the pivoting results

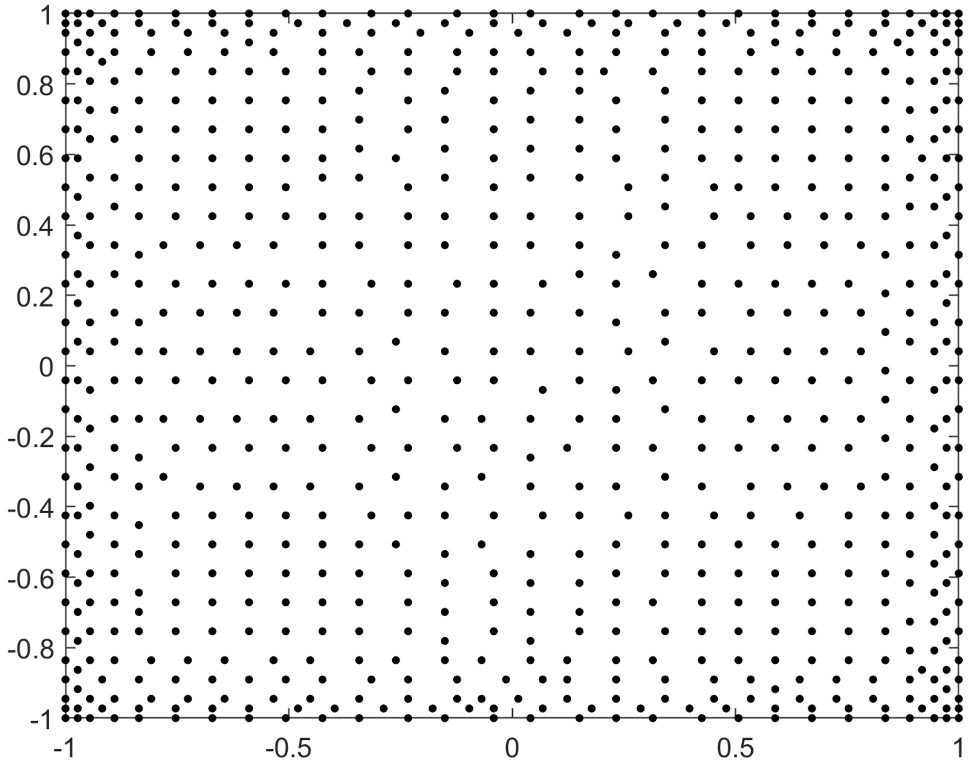

Figure 3.1: Nodes selected using pivoted QR truncated algorithm on the square $[-1, 1] \times [-1, 1]$.

in points that will maximize the determinant of the corresponding Vandermonde matrix. To extend these ideas to frames, an initial QR factorization is needed to remove the rank deficiency.

Similarly to the SVD-regularized frame, the overall approximation procedure can be defined as the mapping

$$\mathcal{P}^{\epsilon,\gamma}_{m,n} : C([-1,1]) \to C([-1,1]), f \to \hat{f}^{\epsilon} = \sum_{i=0}^{p} \hat{d}^{\epsilon}_i \tilde{\psi}_i,$$

Where $\tilde{\psi}_i$ is modified from ψ_i according to the QR decompositions of A.

Running the same three experiments as before for the QR-regularized polynomial frame, we see that the QR-regularized polynomial frame performs similarly to its SVD counterpart in terms of error decay. Additionally, due to the truncations,

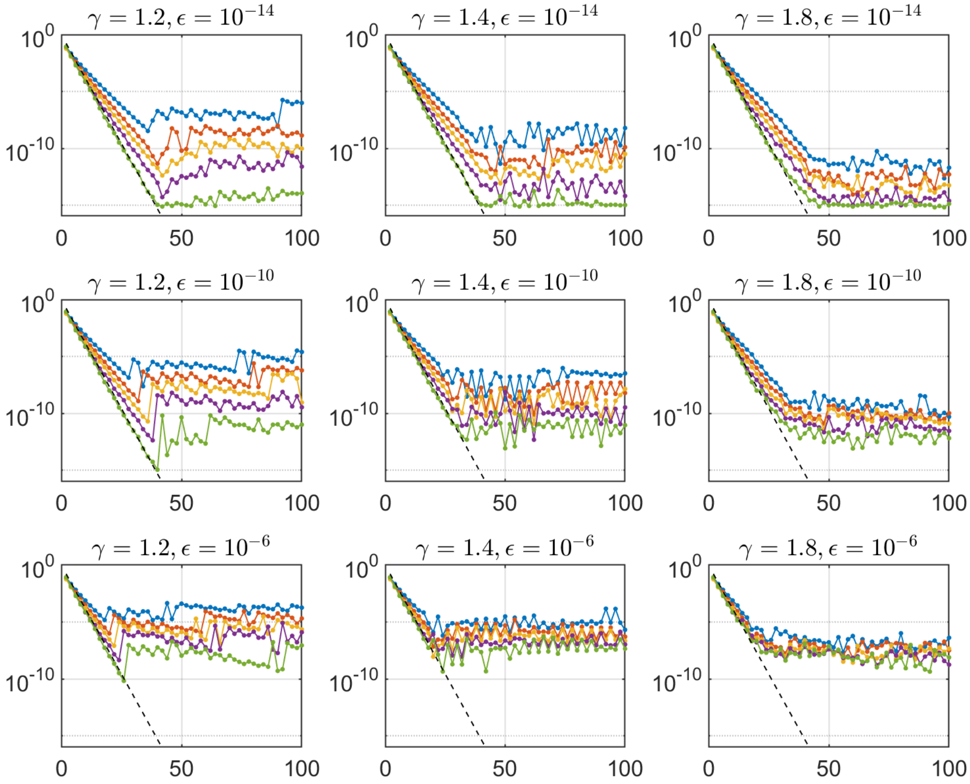

Figure 3.2: QR-regularized frame approximation error versus n for approximating the function $f(x) = 1/(1+x^2)$ via $\mathcal{P}^{\epsilon,\gamma}_{m,n}$, where $m/n = \eta$, using various different values of η, γ and ϵ(Blue: $\eta = 1$, orange: $\eta = 1.25$, yellow: $\eta = 1.5$, purple: $\eta = 2$, green: $\eta = 4$). The dashed line shows the quantity θ^{-n}, where $\theta = \sqrt{2} + 1$.

the QR-regularized polynomial frame uses much less data than the SVD-regularized polynomial frame to reach the same accuracy, as shown in Figure 3.6.

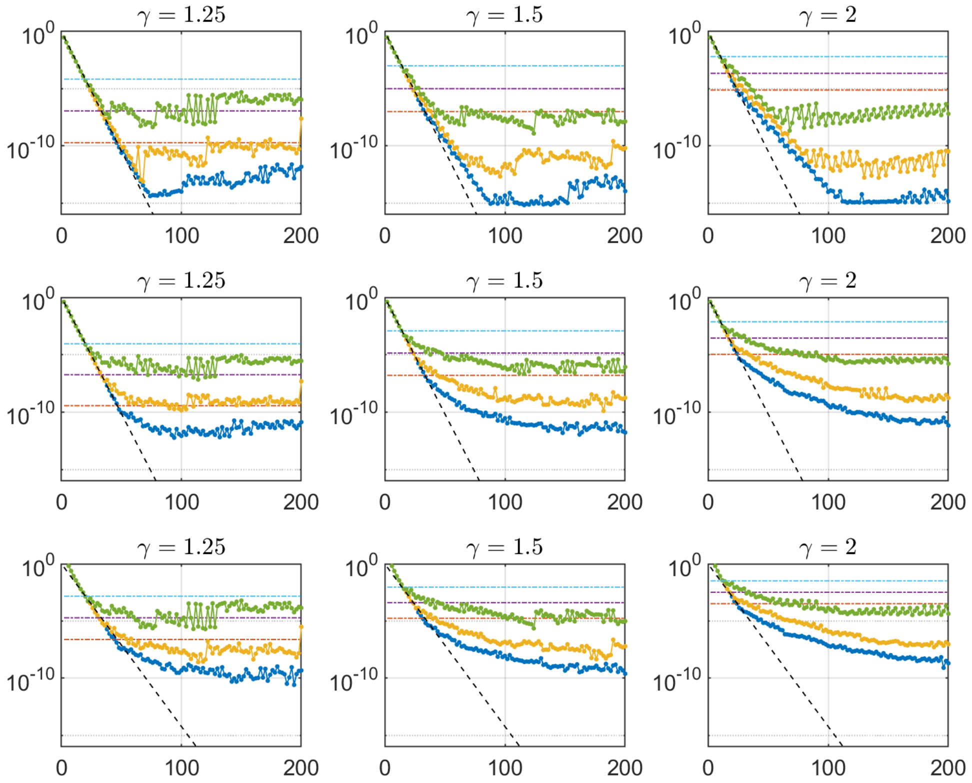

Figure 3.3: QR-regularized frame approximation errors versus n for approximating the functions $f_1(x) = 1/(1+4x^2)$ (top-row), $f_2(x) = 1/(10-9x)$(middle-row) and $f_3(x) = 25\sqrt{(9x^2-10)}$(bottom-row) via $\mathcal{P}^{\epsilon,\gamma}_{m,n}$, where m/n =4, using different values of γ and ϵ (green: $\epsilon = 10^{-14}$, yellow $\epsilon = 10^{-10}$, blue $\epsilon = 10^{-6}$) . The dot-dashed lines show the breakpoints in each case and the dashed line shows the quantity θ^{-n}.

3.1 Frames vs Standard Interpolation

In this section, we will make a comparison between the two frames and standard interpolation on equispaced nodes. In this comparison, the data is based on $f = 1/(1+25x^2) + 0.5\sin(6x)$, which is then perturbed by noise, which is randomly generated from the interval $[-0.05, 0.05]$.

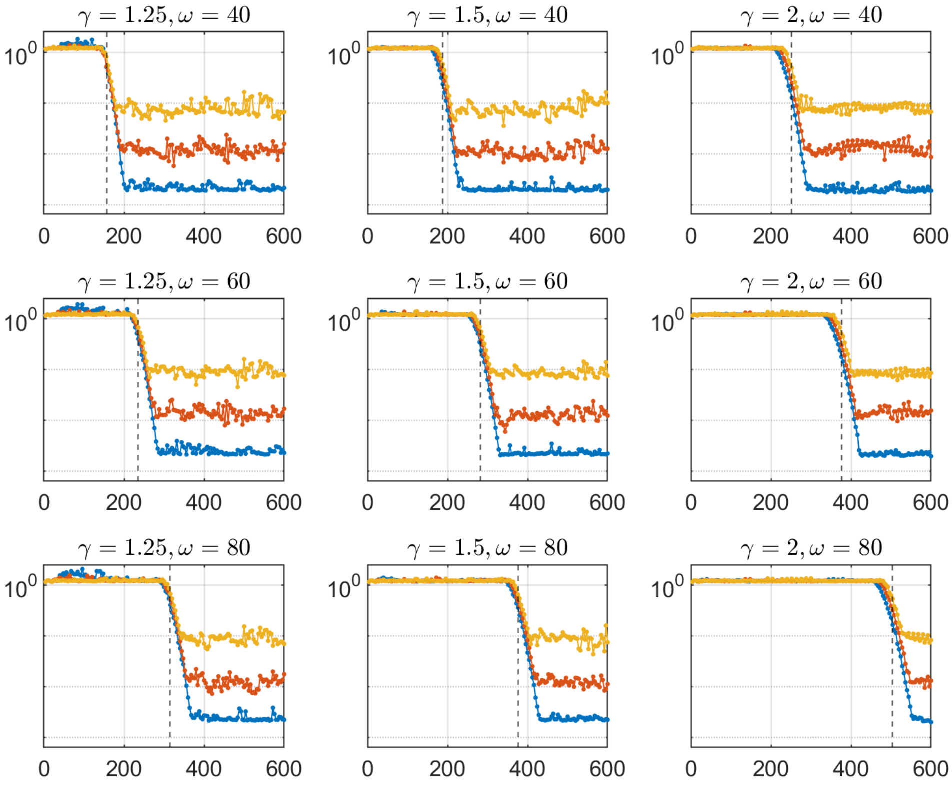

Figure 3.4: QR-regularized frame approximation errors versus n for approximating the functions $f(x) = \exp(i\omega\pi x)$ via $\mathcal{P}^{\epsilon,\gamma}_{m,n}$, where $m/n = 4$, using various different values of γ and ϵ.(Blue: $\epsilon = 10^{-14}$, orange: $\epsilon = 10^{-10}$, yellow: $\epsilon = 10^{-6}$)

As we can see in Figure 3.7, standard interpolation runs into Runge's phenomenon near the endpoints of the interval while the two polynomial frames circumvented the phenomenon due to their construction. We also see that the two frames are relatively close, even though the QR-regularized frame only used a portion of the data.

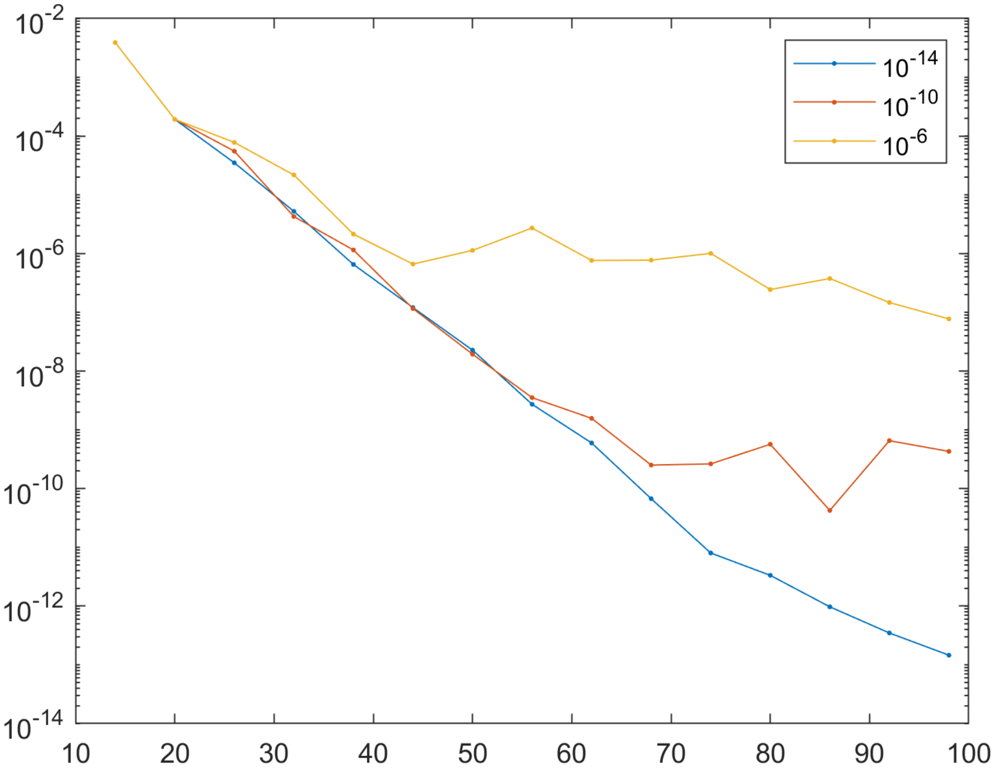

Figure 3.5: 2D QR-regularized frame approximation error using SVD versus n for approximating the function $f(x, y) = 1/(1 + x^2 + y^2)$ via $\mathcal{P}^{\epsilon,\gamma}_{m,n}$, where $m/n = 2$, $\gamma = 1.4$ using various different values of ϵ(Blue: $\epsilon = 10^{-14}$, orange: $\epsilon = 10^{-10}$, yellow: $\epsilon = 10^{-6}$).

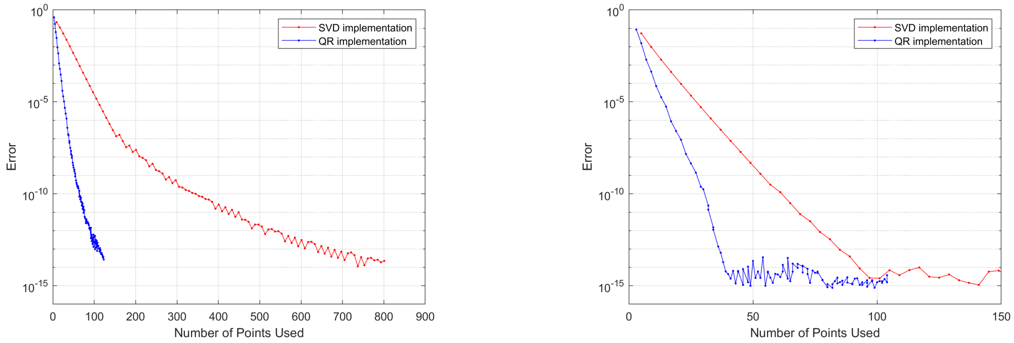

Figure 3.6: Number of points used by SVD vs QR in experiment 1(left) and experiment 2(right).

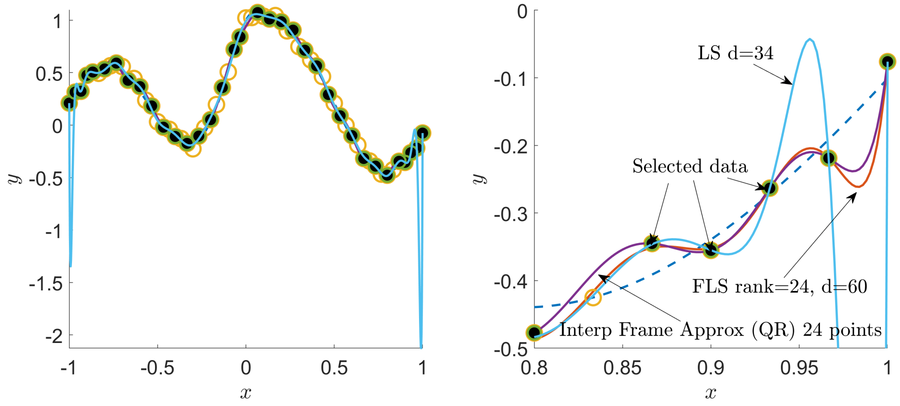

Figure 3.7: Approximation of $f = 1/(1 + 25x^2) + 0.5\sin(6x)$ using SVD-regularized frame, QR-regularized frame, and standard interpolation.

Chapter 4

AZ ALGORITHM

In this chapter, we review the main ideas presented in [5], including the general structure and the default solvers of the AZ algorithm.

4.1 General Overview

In the previous chapters, we opted for an approach that reduces the complexity of the problem through node selection. In contrast, the AZ algorithm aims to reduce the complexity by exploiting the structure of certain least squares problems. Specifically, this algorithm reduces the rank of the system, at which point a variety of solvers can be used, such as randomized QR and SVD.

To introduce the AZ algorithm, suppose that there exists a linear system $Ax = b$, $A \in \mathbb{C}^{m \times n}$, and that Z is another matrix such that $Z \in \mathbb{C}^{m \times n}$. Then, the algorithm is as follows:

---

1. Solve $(I - AZ^*)Ax_1 = (I - AZ^*)b$.
2. Compute $x_2 = Z^*(b - Ax_1)$.
3. Compute $x = x_1 + x_2$.

The main benefit of this algorithm is that for a properly chosen Z, step 1 becomes a low-rank problem, which is much more manageable than the original problem.

From the structure of the algorithm, we can make several statements regarding its accuracy, as follows:

Lemma 4.1.1 (AZ Lemma) *Let $\hat{x} = \hat{x}_1 + \hat{x}_2$ be output from the AZ algorithm. Then the final residual is equal to the residual of step 1.*

Proof Looking at the final residual, we have

$$\begin{aligned} b - A\hat{x} &= b - A\hat{x}_1 - A\hat{x}_2 \\ &= b - A\hat{x}_1 - AZ^*(b - A\hat{x}_1) \\ &= (I - AZ^*)(b - A\hat{x}_1) \end{aligned}$$

Which is the residual from step 1.

An immediate consequence of this lemma is that step 1 of the AZ algorithm determines the overall accuracy of the algorithm when it's based on the residual. Consequently, this lemma allows us to state that the AZ algorithm can find a stable least-squares fit for the approximation.

Lemma 4.1.2 (Stable Least Squares Fit) *Let $A \in \mathbb{C}^{m\times n}, b \in \mathbb{C}^m$, and suppose there exists $\tilde{x} \in \mathbb{C}^n$ such that*

$$||b - A\tilde{x}||_2 \leq \tau, \quad ||\tilde{x}||_2 \leq C$$

for $\tau, C > 0$. Then there exists a solution $\hat{x}_1$ to step 1 of the AZ algorithm such that the residual of the computed vector $\hat{x} = \hat{x}_1 + \hat{x}_2$ satisfies,

$$||b - A\tilde{x}||_2 \leq ||I - AZ^*||\tau, \quad ||\tilde{x}||_2 \leq C + ||Z^*||_2\tau$$

Proof Looking at the final residual, we have

$$
\begin{aligned}
b - A\hat{x} &= b - A\hat{x}_1 - A\hat{x}_2 \\
&= b - A\hat{x}_1 - AZ^*(b - A\hat{x}_1) \\
&= (I - AZ^*)(b - A\hat{x}_1) \\
||b - A\hat{x}||_2 &\leq ||I - AZ^*||_2\tau \\
\hat{x} &= \tilde{x} + Z^*(b - A\tilde{x}) \\
||\hat{x}||_2 &\leq C + ||Z^*||_2\tau
\end{aligned}
$$

4.2 Choosing the Z Matrix

Overall, in the AZ algorithm, Z can be chosen randomly. However, to maximize efficiency and accuracy, we have to be careful. In the context of Lemma 4.1.2, we see that Z^* needs to have a small norm for there to be a stable least squares fit. Additionally, if $A - AZ^*A$ is not low rank, step 1 of the AZ algorithm might be expensive.

We can look at two extreme and general cases of Z, which are $Z = 0$ and $Z = (A^\dagger)^*$. For $Z = 0$, solving step 1 of the AZ algorithm is the same as solving the original system, and step 2 simply returns 0, which doesn't give us any efficiency over solving the original system. On the other hand, if $Z = (A^\dagger)^*$, then step 1 of the AZ algorithm returns $x_1 = 0$ and step 2 provides the solution to the original system in terms of $A^\dagger$. While the solution might be efficient, computing $A^\dagger$ is not trivial and often not computationally feasible for large A.

For other choices of Z, part of the solution is found in step 1, and part of it is found in step 2. To find a matrix Z that will result in $A - AZ^*A$ being low rank, and Z^* having an efficient matrix-vector multiplication, we can choose Z based on some a priori information regarding the original system. The following lemma gives a general relationship between A and Z which would guarantee $A - AZ^*A$ to be numerically low rank.

Lemma 4.2.1 *Suppose that $A, Z \in \mathbb{C}^{m\times n}$ satisfy*

$$A = W + L_1 + E_1, \quad Z^* = W^\dagger + L_2 + E_2$$

where rank(L_1),rank(L_2) $\leq R$, and $||E_1||_F$, $||E_2||_F \leq \epsilon$. Here $W^\dagger$ is the Moore-Penrose pseudoinverse. Then

$$A - AZ^*A = L + E$$

where rank(L) $\leq 3R$ and

$$||E_1||_F \leq \epsilon(1 + ||I - AZ^*||_2 + ||A||_2^2) + \epsilon^2||A||_2$$

This result also holds if the norms on E, E_1, and E_2 are changed to $||.||_2$.
Practically, we will not explicitly compute W, L_1, and L_2 matrices, since it is sufficient for them to exist to guarantee the effectiveness of the algorithm.

4.3 Solvers for Numerically Low-Rank Systems

For step 1 of the AZ algorithm, suppose we found a Z such that $A - AZ^*A$ is low rank, then we can use a variety of solvers to find the solution. In this thesis, we will use randomized methods, in particular, truncated SVD and QR.
For a general system $Ax = b$, $A \in \mathbb{C}^{m\times n}$ with $m \geq n$, we can define the epsilon rank of A as $\text{rank}_\epsilon(A)$. For $\text{rank}_\epsilon(A) = r$, it means that there exists L, $E \in \mathbb{C}^{m\times n}$ such that

$$A = L + E, \quad \text{where rank}(L) = r \text{ and } ||E||_F \leq \epsilon$$

The Frobenius norm is used here to ensure that $\sum_{k>r} \sigma_k^2 \leq \epsilon^2$, where $\sigma_{r+1}, ..., \sigma_N$ are the smallest $N - r$ singular values of A. This will help us define the error bounds for the solvers later.

First, in both randomized truncated solvers, we will make use of Gaussian random matrices $\Omega \sim \mathcal{N}(0,1;\mathbb{R}^{n\times k})$, which are $n \times k$ matrices consisting of independent random variables with mean 0 and variance 1. Then, we will define $k = r + p$, where r is the epsilon rank of A and $p \geq 2$. Thus, the overall randomized truncated SVD solver can be described in algorithm 4.3.1:

Algorithm 4.3.1 *Randomized truncated SVD solver*

```
Input: $A \in \mathbb{C}^{m\times n}, b \in \mathbb{C}^m, k \in \{1, ..., n\}, \epsilon > 0$
Output: $x \in \mathbb{C}^n$ such that $Ax \approx b$
Generate $\Omega \sim \mathcal{N}(0,1;\mathbb{R}^{n\times k})$
$\tilde{A} \leftarrow A\Omega \in \mathbb{C}^{m\times k}$
Compute the SVD, $\tilde{A} = \tilde{U}\tilde{\Sigma}\tilde{V}^*$ where
    $\tilde{U}\tilde{\Sigma}\tilde{V}^* = \begin{pmatrix} \tilde{U}_1 & \tilde{U}_2 \end{pmatrix} \begin{pmatrix} \tilde{\Sigma}_1 & \\ & \tilde{\Sigma}_2 \end{pmatrix} \begin{pmatrix} \tilde{V}_1 & \tilde{V}_2 \end{pmatrix}^*,$
    with $0 \leq \tilde{\Sigma}_2 < \epsilon I \leq \tilde{\Sigma}_1$. Here $\tilde{U} \in \mathbb{C}^{m\times k}$, $\tilde{\Sigma} \in \mathbb{C}^{k\times k}$, $\tilde{V} \in \mathbb{C}^{k\times k}$, where the dimension of the blocks will be determined by the singular values.
$y \leftarrow \tilde{V}_1\tilde{\Sigma}_1^{-1}\tilde{U}_1^* b$
$x \leftarrow \Omega y$
```

As for the error bounds, we have the following theorem:

Theorem 4.3.2 (Residual bounds for randomized truncated SVD solver[1]) *Assume that A satisfies $\text{rank}_\epsilon(A) = r$, and let $x \in \mathbb{C}^n$ be the solution from algorithm*

4.1 with $k = r + k$ for $k \geq 2$. Then

$$||b - Ax||_2 \leq \inf_{v \in \mathbb{C}^n} \{||b - Av||_2 + \epsilon(1 + \kappa)||v||_2\},$$

where κ is a non-negative random variable satisfying

$$\mathbb{E}\{\kappa\} \leq 2\sqrt{\frac{r}{p-1}}, \quad \mathbb{P}\left\{\kappa > (2+u)s\sqrt{\frac{3r}{p-1}}\right\} \leq s^{-p} + e^{-\frac{u^2}{2}},$$

for any $s \geq 1$, $u \geq 0$.

Looking at theorem 4.3.2, we can see that $\kappa = \mathcal{O}(\sqrt{r})$ with a high probability that improves as p increases. While we can let p to grow linearly with respect to r to make $\mathbb{E}\{\kappa\}$ $\mathcal{O}(1)$ instead of $\mathcal{O}(\sqrt{r})$, but this is usually not necessary, and a fixed p value is sufficient.

For randomized truncated QR, the algorithm is very similar to its SVD counterpart, except that the decomposition is based on pivoted QR, as shown in algorithm 4.3.3.

Algorithm 4.3.3 *Randomized truncated pivoted QR solver*

Input: $A \in \mathbb{C}^{m \times n}, b \in \mathbb{C}^m, k \in \{1, ..., n\}, \epsilon > 0$

Output: $x \in \mathbb{C}^n$ such that $Ax \approx b$

1: Generate $\Omega \sim \mathcal{N}(0, 1; \mathbb{R}^{n \times k})$

2: $\tilde{A} \leftarrow A\Omega \in \mathbb{C}^{m \times k}$

3: Compute a column pivoted QR decomposition, $\tilde{A}\tilde{\Pi} = \tilde{Q}\tilde{R}$ where

$$\tilde{\Pi} = \begin{pmatrix} \tilde{\Pi}_1 & \tilde{\Pi}_2 \end{pmatrix}, \quad \tilde{Q} = \begin{pmatrix} \tilde{Q}_1 & \tilde{Q}_2 \end{pmatrix}, \tilde{R} = \begin{pmatrix} \tilde{R}_{11} & \tilde{R}_{12} \\ 0 & \tilde{R}_{22} \end{pmatrix},$$

with $0 \leq \text{diag}(\tilde{R}_{22}) < \epsilon I \leq \text{diag}(\tilde{R}_{11})$. Here $\tilde{\Pi} \in \mathbb{C}^{m \times k}$, $\tilde{Q} \in \mathbb{C}^{m \times k}$, $\tilde{R} \in \mathbb{C}^{k \times k}$, where the dimension of the blocks will be determined by the diagonal entries of $\tilde{R}$.

```
4: $y \leftarrow \tilde{\Pi}_1 \tilde{R}_1^{-11} \tilde{Q}_1^* b$
5: $x \leftarrow \Omega y$
```

The error bounds for algorithm 4.3.3 is shown in theorem 4.3.4:

Theorem 4.3.4 (Residual bounds for randomized truncated pivoted QR solver[1]) *Assume that A satisfies $rank_\epsilon(A) = r$, and let $x \in \mathbb{C}^n$ be the solution from algorithm 4.2 with $k = r + k$ for $k \geq 2$. Then*

$$||b - Ax||_2 \leq \underset{v \in \mathbb{C}^n}{inf} \{ ||b - Av||_2 + \epsilon(1+\kappa)||v||_2 \},$$

where κ is a non-negative random variable satisfying

$$\mathbb{E}\{\kappa\} \leq (1 + \sqrt{r+p})\sqrt{\frac{r}{p-1}}, \quad \mathbb{P}\left\{ \kappa > (1 + \sqrt{r+p} + u)s\sqrt{\frac{3r}{p-1}} \right\} \leq s^{-p} + e^{-\frac{u^2}{2}},$$

for any $s \geq 1$, $u \geq 0$.

We can see that compared to theorem 4.3.4, $\kappa = \mathcal{O}(r)$. This could be attributed to the possibility that the diagonal values of $\tilde{R}_2 2$ are not directly related to the numerical rank of $\tilde{A}$. Nonetheless, based on the discussions by Golub and Van Loan, in practice, we would still expect $\kappa = \mathcal{O}(\sqrt{r})$ similar to algorithm 4.3.1[6].

Overall, describing the computational cost for algorithms 4.3.1 and 4.3.3 step-by-step, we have:

1. Generate $n \cdot k$ Gaussian random numbers: $\mathcal{O}(n \cdot k)$

2. Apply the matrix A to k vectors: $\mathcal{O}(k \cdot m \cdot n)$

3. Compute the SVD or pivoted QR factorization of an $m \times k$ matrix: $\mathcal{O}(m \cdot k^2)$

4. Apply a sequence of matrices with one dimension smaller than k and the other smaller than m: $\mathcal{O}(r \cdot m)$

5. Apply a matrix of size $n \times k$ to a vector: $\mathcal{O}(r \cdot n)$ Thus, the total computational cost of each algorithm is

$$\mathcal{O}(kmn + r^2 m)$$

which is better than the non-randomized counterparts that require $\mathcal{O}(mn^2)$ flops, especially when $k = o(n)$. Additionally, algorithms 4.3.1 and 4.3.3 will also be more efficient if A supports fast matrix-vector multiplication.

Chapter 5

IMPLEMENTATION OF THE AZ ALGORITHM FOR POLYNOMIAL FRAMES

The algorithm presented in [5] mainly addressed Fourier frames, in this thesis we extend those results to polynomial frames.

For polynomial frames, provided that we can satisfy a discrete orthogonality condition, we can effectively apply the AZ algorithm. To ensure the stability of the solution, we will oversample the nodes such that the number of these nodes is greater than the number of basis functions by a factor of at least 2.

As for the discrete orthogonality condition, for Legendre polynomials on a full grid of L, we have

$$\sum_{l=1}^{L} w_l P_i(x_l) P_j(x_l) = h_i^2 \delta_{i-j}, \quad 1 \leq i, j \leq L-1.$$

Where w_l are the weights associated with the roots of P_i and $h_i = ||P_i||_{[-1,1]}$ is the norm of P_i. Similar to the Fourier case, there is a large $L \times L$ matrix F with entries $F_{i,j} = P_{i-1}(x_{j-1})$. It follows that $F^{-1} = DF^*W$, where W and D are diagonal matrices with entries w_i and h_i^{-2} respectively. Meanwhile, A is a submatrix of F, with columns corresponding to the basis functions and rows corresponding to the points in the domain of interest. It follows that Z is the corresponding subblock of $(F^{-1})^* = WFD$.

Examining the structure of the Z matrix, the first concern that arises is the choice of nodes. While equispaced nodes are easy to generate, it would mean that we will have to work with Newton-Cotes formulation for the weights w_l, which becomes highly unstable for large degrees due to Runge's phenomenon. To circumvent this, we propose a composite node formulation.

In this composite node formulation, we begin by generating Chebyshev nodes on the entire domain. Then, determine a subdomain such that it includes the region of interest (usually a square or rectangle), and swap the Chebyshev nodes within the subdomain with equispaced nodes. The choice of starting with Chebyshev nodes is due to their clustering near the boundary and their similarity with equispaced nodes near the center of the domain.

Orthogonalizing the polynomial basis based on the composite nodes, we can eliminate the need to form a separate weight matrix. Furthermore, with normalization, we can also eliminate the need for a separate norm matrix. These two simplifications mean that Z can simply be A, since $(F^{-1})^* = F$. In other words, A and Z can now be simultaneously computed.

5.1 Effectiveness of the AZ algorithm

To demonstrate the efficiency of the AZ algorithm, we can make a comparison between the conventional frame algorithms with the AZ-augmented frame algorithms. In this comparison, the data is based on the function $f(x, y) = 1/(1 + 25(x^2 + y^2))$, which is a challenging function for approximation in the context of polynomial approximation. Specifically, we will use the randomized SVD and pivoted QR algorithms, as well as their AZ-augmented counterparts.

In Figure 5.1, we see that for the original matrix A, the singular values cluster around $\mathcal{O}(1)$ and around machine epsilon, while for the matrix $A - AA^*A$, we see that the singular values now only cluster around the machine epsilon.

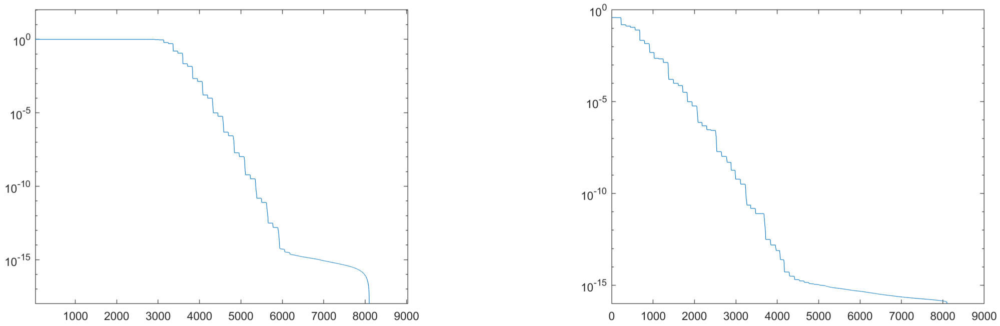

Figure 5.1: Singular values of A(left) and singular values of $A - AA^*A$(right).

This demonstrates the role of $I - AA^*$, in that it maps the cluster of singular values at $\mathcal{O}(1)$ to the region close to machine epsilon, allowing for significant rank reduction during the AZ algorithm. Thus, as we can see in Figure 5.2, the AZ-augmented algorithms are consistently faster than the non-AZ counterparts, mostly due to the rank reduction caused by the Z matrix.

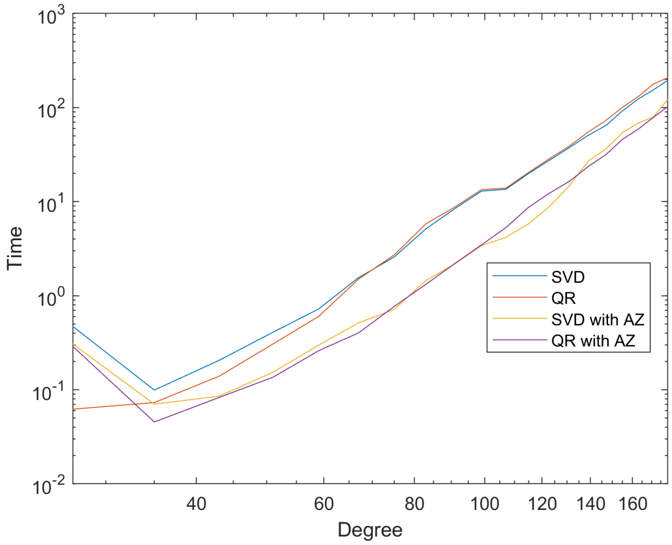

Figure 5.2: Running time in seconds versus the maximum degree of the frame extension for calculating $f(x, y) = 1/(1 + 25(x^2 + y^2))$ using randomized truncated SVD and randomized truncated pivoted QR with and without the AZ algorithm,

Chapter 6

FOURIER FRAMES

6.1 Frame Approximation

To formulate a d-dimensional Fourier frame, we first consider the d-dimensional Fourier basis on the box $[-1,1]^d$,

$$\Phi = \{\phi_n\}_{n \in \mathbb{Z}^d}, \quad \text{with } \phi_n(x) = 2^{-d/2} e^{i\pi n x}, \quad x \in [-1,1]^d.$$

Restricting this basis to a small domain $\Omega \subset [-1,1]^d$ will yield a Fourier extension frame for $L^2(\Omega)$. Consequently, any sufficiently smooth function f can be well approximated using this frame, regardless of whether it's periodic or not. Additionally, unlike polynomial frames, Fourier extensions don't suffer from the same problems on equispaced nodes. While standard Fourier approximation seems to be sufficient, there are still benefits to using the Fourier frame approximation, in that the frame functions do not need to be fully orthogonal, which offers more freedom during formulation. Even though the redundancy present in the resulting linear system will lead to ill-conditioning, similar to its polynomial counterpart, stable and efficient algorithms are possible, especially for A matrices that possess a plunge region[8].

Applying the truncated QR-regularization, we perform the same experiments with Fourier frames as in Chapters 2 and 3.

We see in Figure 6.1 that aside from some differences due to rounding errors and nonperiodicity of the function, the Fourier frame approximation behaves similarly to its polynomial counterpart, where the convergence is mostly stable and is approximately subgeometric until the breakpoint specified by ϵ. The increase in overall

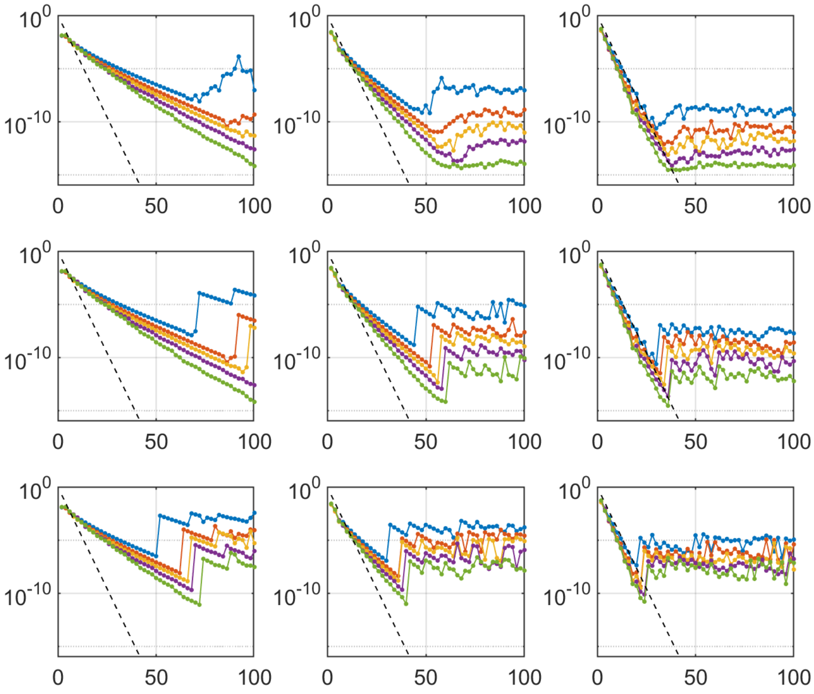

Figure 6.1: Fourier frame approximation error versus n for approximating the function $f(x) = 1/(1+x^2)$ via $\mathcal{F}^{\epsilon,\gamma}_{m,n}$, where $m/n = \eta$, using various different values of η, γ and ϵ. The values of ϵ used are $\epsilon = 10^{-14}$(top), $\epsilon = 10^{-10}$ (middle) and $\epsilon = 10^{-6}$ (bottom). The values of γ used are $\gamma = 1.2$(left), $\gamma = 1.4$ (middle) and $\gamma = 1.8$ (right). The dashed line shows the quantity θ^{-n}, where $\theta = \sqrt{2}+1$.

approximation error can be explained by the non-periodicity of $f(x) = 1/(1+x^2)$. In Figure 6.2, we can see that for the Fourier frame approximation, the function resolves much sooner than in the polynomial counterpart, due to the periodicity of $f(x) = \exp(i\omega\pi x)$. Additionally, the spikes in the plot can be explained by the truncation of certain frame functions during the approximation.

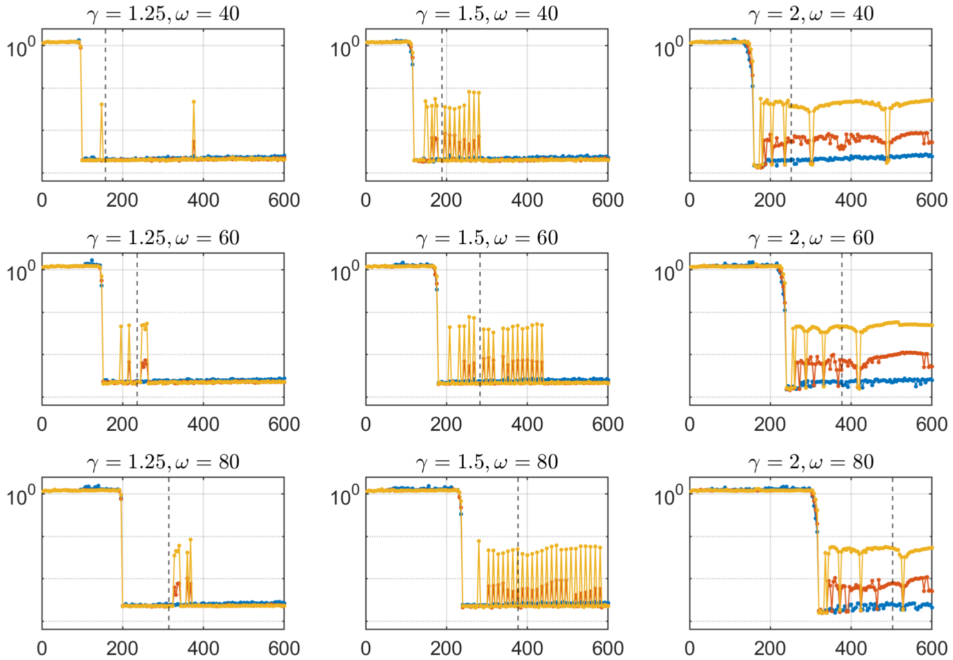

Figure 6.2: Fourier approximation errors versus n for approximating the functions $f(x) = \exp(i\omega\pi x)$ via $\mathcal{P}^{\epsilon,\gamma}_{m,n}$, where $m/n = 4$, using various different values of γ and ϵ.(Blue: $\epsilon = 10^{-14}$, orange: $\epsilon = 10^{-10}$, yellow: $\epsilon = 10^{-6}$)

6.2 AZ Implementation

To apply the AZ algorithm on Fourier frames, let's first reexamine the linear system $Ax = B$ with

$$A_{j,k} = \phi_k(x_j), \quad B_j = f(x_j), \quad j = 1, ..., n, \quad k = 1, ..., m.$$

Where $\phi_k(x_j)$ is as defined in Chapter 3. In fact, consider a set of L Fourier basis functions on a periodic equispaced grid $x_l{}_{l=1}^{L}$, we have

$$\sum_{l=1}^{L} \phi_i(x_l)\bar{\phi}_i(x_l) = L\delta_{i-j}, \quad 1 \leq i, j \leq L.$$

Thus, we can see that the full DFT matrix F of size $L \times L$ with entries $\phi_i(x_l)$ has inverse $\frac{1}{L}F^*$. The discrete approximation problem follows by choosing a subset of

$M < L$ points that belong to a subdomain Ω and a subset of $N < M$ basis functions. It follows that A is a submatrix of F similar to the polynomial counterpart, and based on the orthogonality condition, we can choose[5]

$$Z = \frac{1}{L}A$$

We can then interpret this formulation as follows: multiplying with A is an extension from N to L Fourier coefficients, followed by the DFT of length L, and followed by the restriction to M points in Ω. On the other hand, multiplying with Z^* is equivalent to extension by zero-padding in the time domain, followed by the inverse DFT, and restriction in the frequency domain. Consequently, the singular values of matrix $A - AZ^*A$ have a large cluster near machine epsilon, similar to the polynomial counterpart. This results in a smooth and periodic function, for which the discrete inverse Fourier transform gives an accurate approximation. Compared to the polynomial frames, although it will be superior in approximating periodic functions, it will be inferior in terms of approximating non-periodic functions, as shown in Section 6.1.

Chapter 7

FINAL REMARKS

Overall, we used ϵ-truncated SVD and column-pivoted QR decomposition to compute feasible approximations of the polynomial frame, which asserts stability and exponential convergence down to a finite, but user-controlled limiting accuracy. This approximation holds for analytic functions in a sufficiently large region, and for insufficient analytic functions, we have shown exponential decay down to some fractional power of ϵ, and superalgebraic decay beyond that point. Additionally, we also formulated a node-selection method by leveraging the properties of truncated QR, while maintaining the original accuracy of the frame.

On the other hand, we used the AZ algorithm to improve the speed of the frame formulation by using a composite set of nodes and corresponding orthogonalization of the polynomial basis in conjunction with randomized truncated SVD and QR decomposition, allowing us to circumvent Runge's phenomenon and efficiently tackle functions of interest in the context of polynomial approximation.

In terms of possible improvements, for the AZ algorithm, a faster rank-revealing algorithm will be very beneficial in terms of determining the optimal dimensions of the Gaussian matrices for the randomized solvers. On the other hand, using iterative solvers such as LSQR and LSMR for step 1 of the AZ algorithm might also be more efficient, making it a viable route for investigation as well.

In the future, we are interested in reconciling the two approaches to polynomial frame approximation, which are the QR-regularized node selection and the AZ algorithm. Ideally, we would want to incorporate the main advantages of each approach while avoiding as many downsides as possible, such that we can apply this combination efficiently to a variety of complicated approximation problems on more unusual multidimensional domains.

REFERENCES

[1] Adcock, B. and D. Huybrechs, “Frames and numerical approximation”, SIAM Review **61**, 3, 443–473 (2019).

[2] Adcock, B. and D. Huybrechs, “Approximating smooth, multivariate functions on irregular domains”, in “Forum of Mathematics, Sigma”, vol. 8, p. e26 (Cambridge University Press, 2020).

[3] Adcock, B. and A. Shadrin, “On the possibility of fast stable approximation of analytic functions from equispaced samples via polynomial frames”, Tech. rep., Technical Report (2021).

[4] Boyd, J. P., “A comparison of numerical algorithms for fourier extension of the first, second, and third kinds”, Journal of Computational Physics **178**, 1, 118–160 (2002).

[5] Coppé, V., D. Huybrechs, R. Matthysen and M. Webb, “The az algorithm for least squares systems with a known incomplete generalized inverse”, SIAM Journal on Matrix Analysis and Applications **41**, 3, 1237–1259 (2020).

[6] Golub, G. H. and C. F. Van Loan, *Matrix computations* (JHU press, 2013).

[7] Guo, M., “Optimal sampling for function approximation”, (2021).

[8] Huybrechs, D., “On the fourier extension of nonperiodic functions”, SIAM Journal on Numerical Analysis **47**, 6, 4326–4355 (2010).

[9] Lyon, M., “A fast algorithm for fourier continuation”, SIAM Journal on Scientific Computing **33**, 6, 3241–3260 (2011).

[10] Platte, R. B., L. N. Trefethen and A. B. Kuijlaars, “Impossibility of fast stable approximation of analytic functions from equispaced samples”, SIAM Review **53**, 2, 308–318 (2011).

[11] Sommariva, A. and M. Vianello, “Computing approximate fekete points by qr factorizations of vandermonde matrices”, Computers & Mathematics with Applications **57**, 8, 1324–1336 (2009).

[12] Trefethen, L. N., *Approximation Theory and Approximation Practice, Extended Edition* (SIAM, 2019).