Computational Linear Algebra Mastery Material

Azmat Habibullah - 01340685

January 2021

Introduction

Frequently one seeks to approximate a function or find the closest solution to a problem. These problems are referred to as polynomial interpolation or least squares problems.

A common approach to both of these problems involves Vandermonde matrices, which are badly behaved for large degrees and thus difficult to work with. In this paper, we discuss an improved implementation based on recent work by Trefethen [1]. We demonstrate problems with the standard Vandermonde approach used in interpolation and Fourier extension, and propose some potential solutions, relating the theory to mathematical biology.

The paper is split into four section. Section 1 provides background on polynomial interpolation and least squares problems and touches on some of the issues faced when working with Vandermonde matrices. Section 2 explains the 'Vandermonde and Arnoldi' (VA) idea in [1] and discusses the theory behind the implementation. Section 3 demonstrates the benefits of this system with polynomial interpolation. Section 4 turns to Fourier extensions, linking our theory with frames and singular value decomposition. Section 5 mentions ideas for further research.

0.1 Running code

Figures in this report can be reproduced by running `main.py` (some figures have lower parameter values due to high execution time). Documentation explains how to use the code.

1 Background

1.1 Interpolation and least squares

The interpolation problem is as follows: we seek to interpolate a function $f : \mathbb{R} \to \mathbb{R}$ on an interval I (with monomials). Suppose we have $N + 1$ (distinct) data points $x_0, \ldots, x_N$, for which we know values $f_i = f(x_i)$, for $i = 0, \ldots, N$. We seek a polynomial such that $p(x_i) = f_i$.

Theorem 1. *An interpolating polynomial for $N + 1$ data points exists. This polynomial has degree N and is unique.*

Suppose instead we are interested in interpolating M data points with a degree N polynomial. It is clear there will be some error, and so we choose to minimise the squared error. This is N equations in $m = deg(p) + 1$ unknowns, which we may write as $Vc = f$, where V is the Vandermonde matrix. The condition number κ allows one to measure how bad a matrix is to work with. A large condition number indicates instability and increasing likelihood of floating-point errors, which we want to avoid.

Theorem 2. *The condition number of an $n \times n$ Vandermonde matrix is exponential in n unless its knots are more or less equally spaced on or about the unit circle C(0, 1).*

Proof. See [2] □

This has important consequences: generally, for large n, working directly with the Vandermonde matrix will lead to catastrophic floating point arithmetic errors. This paper aims to reduces such complications.

1.2 Arnoldi iteration

Consider the system $Ax = b$ and approximate the solution using the basis $(b, Ab, \dots, A^k b)$, whose span is called a *Krylov subspace*. After each iteration the Krylov subspace grows by one dimension, but later elements in this sequence get very parallel, so we must orthogonalise. Instead of doing this altogether at the end via a QR factorisation, we can orthogonalise at each iteration. Consider the transformation of A to an upper Hessenberg matrix by orthogonal similarity transforms, so $A = QHQ^*$. Denote by $\hat{Q}_n$ the first n columns of Q and by $\tilde{H}_n$ the $(n+1) \times n$ upper-left hand section of H. We have

$$A\hat{Q}_n = \hat{Q}_{n-1}\tilde{H}_n \implies Aq_n = h_{1,n}q_1 + \cdots + h_{n,n}q_n + h_{n+1,n}q_{n+1}$$

From this we see how to construct non-zero entries of the nth column of H, so we obtain the *Arnoldi iteration* algorithm to solve $Ax = b$:

$q_1 \leftarrow b/||b||$
for $k = 1, 2, \dots$ **do**
  $v \leftarrow Aq_k$
  for $j = 1$ to k **do**
    $h_{jk} \leftarrow q_j^* v$
    $v \leftarrow v - h_{jk}q_j$
  end for
  $h_{k+1,k} \leftarrow ||v||$
  $q_{k+1} \leftarrow v/||v||$
end for

2 The Vandermonde and Arnoldi idea (VA)

We aim to solve the system $Vc = f$. One could solve this problem using a QR factorisation of V, and another method proposed by Bjorck and Pereyra solves a linear Vandermonde system with a complexity of $O(n^2)$ [3]. Further improvements have been made in special cases, however, the main issue is Theorem 2: the monomial basis $\{1, x, x^2, \dots\}$ is exponentially ill-conditioned, which results in poor accuracy for large n. The key idea in this paper is to change basis to remedy this.

Instead of working directly with the monomial basis, we can equivalently think of the monomial basis as a Krylov sequence: consider it as $\{q_0, Aq_0, A^2q_0, \dots\}$, where $q_0 = (1, \dots, 1)^T$ and $X = diag(x_1, \dots, x_m)$. Instead of orthogonalising the result at the end, as in QR factorisation, we orthogonalise at each step in Arnoldi, obtaining a sequence of orthogonal vectors spanning the same space. This amounts to constructing an approximation to orthogonal polynomials at each step, rather than constructing all of the powers and then orthogonalising, resulting in a well-conditioned basis.

Applying Arnoldi iteration to the monomial basis constructs discrete orthogonal polynomials. This involves a Hessenberg matrix, which will need to be passed around our functions to interpolate and evaluate. This is because we have changed basis when interpolating and so need to change back to evaluate at our new points.

With this in mind, we can construct a superior algorithm to interpolate and evaluate the problem. We begin by applying the Arnoldi algorithm to the m-vector of ones in this basis. After n steps, we have n orthogonal vectors $q_0, \dots, q_n$ and an $(n+1) \times n$ upper-Hessenberg matrix H satisfying

$$XQ_- = QH$$

where $Q \in \mathbb{R}^{(m \times (n+1)}$ is the matrix with columns q_i, $i = 0, \dots, n$, and Q_- is Q without the final column. Now with $A = QR$ we can solve (in the least squares sense) the equivalent system $Qd = f$, where $Rd = c$, and $R = Q^T A/m$. This ends the fitting part of the process, which is simply applying Arnoldi iteration to change basis and then solving (in the least squares sense).

To evaluate points, we need to change back to the monomial basis via the matrix H. The $k+1$th column of H was used to orthogonalise Aq_k against $q_0, \dots, q_k$, so we apply the same operations, but to the evaluation points instead of the interpolation points. Writing the evaluation points as $s_1, \dots, s_M$, with $S = diag(s_1, \dots, s_M)$ this is

$$SW_- = WH$$

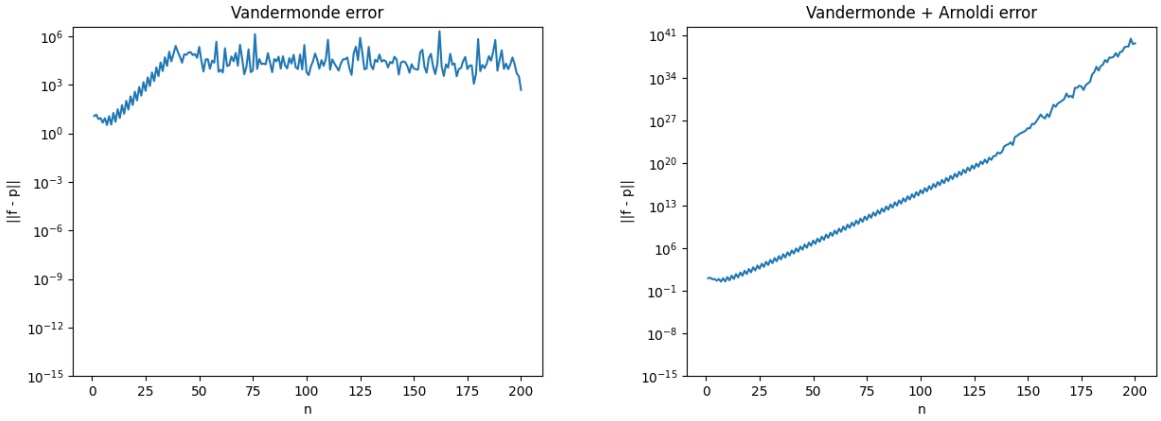

Figure 1: Errors for the Runge function with $s = 1000$ equispaced nodes on $[-1, 1]$. For all plots, the error is shown on a log scale. Exponential ill-conditioning is visible for both cases, but notice the differences in scale.

which is of the same form as for the interpolation. Finally, we have that $Wd = p(s_j)$ as desired.

The fitting part of this algorithm is $O(mn^2)$, which is the same complexity as the standard Vandermonde fitting algorithm. The evaluating part is $O(Mn^2)$, in fact slower than the standard evaluation algorithm. This paper focuses on accuracy. For discussion on efficiency see [4]. Further details on how this relates to orthogonal polynomials are provided in [5] and [6].

The algorithms for `polyfit`, `polyval`, `polyfitA` and `polyvalA` are outlined in [1] and the implementations are in the file `fit.py`. The tests for this code are in `test.pt` and verify that the fit is correct As in [1], the only non-standard feature compared to the standard Arnoldi algorithm is enforcing $|q_k| = \sqrt{m}$ instead of 1. These algorithms are easily run via other functions, as seen in `main.py`. Documentation explains how to run the code.

3 Interpolation

In this section we explore issues with interpolation, specifically choice of interpolation points and how this effects the Vandermonde and VA approaches. We end with an application in biology which can easily be generalised.

3.1 Equispaced interpolation of the Runge function

The *Runge function* $f(x) = \frac{1}{1+25x^2}$ is significant for its role in the discovery of *Runge's phenomenon*. Runge showed that if this function is interpolated at equidistant points x_i in $[-1, 1]$ with a polynomial of degree n, the interpolation oscillates towards the end of the interval, with the error increasing without bound as the degree n is increased [7]. This is because of two reasons: 1) the magnitude of the nth order derivatives of Runge's function grows quickly as n increases and 2) the equidistance between points leads to a Lebesgue constant that increases quickly when n increases.

A demonstration of just how poor the interpolation is, using both Vandermonde and VA, can be seen in Figure 1. It is clear here that the errors are catastrophic. The interpolation points were $s = 1000$ equidistant points in $[-1, 1]$. The error grows exponentially in the Vandermonde case but stabilises around order 10^5. With Arnoldi, the situation is worse still, with errors passing 10^{41} for $n = 200$.

The difference in plots can be explained by examining both of the algorithms. The VA algorithm uses modified Gram-Schmidt algorithm to produce a sequence of orthonormal vectors $q_1, q_2, \ldots$, such that for every n, the vectors $q_1, \ldots, q_n$ span the Krylov subspace K_n. The Vandermonde algorithm on the other hand uses Householder triangulation. Modified Gram-Schmidt is known to be less stable than Householder [8], which leads to the amplification of errors.

This example demonstrates the importance of picking suitable interpolation points to ensure the problem is well-conditioned. The Arnoldi algorithm has better accuracy when used properly, but misuse can still be catastrophic.

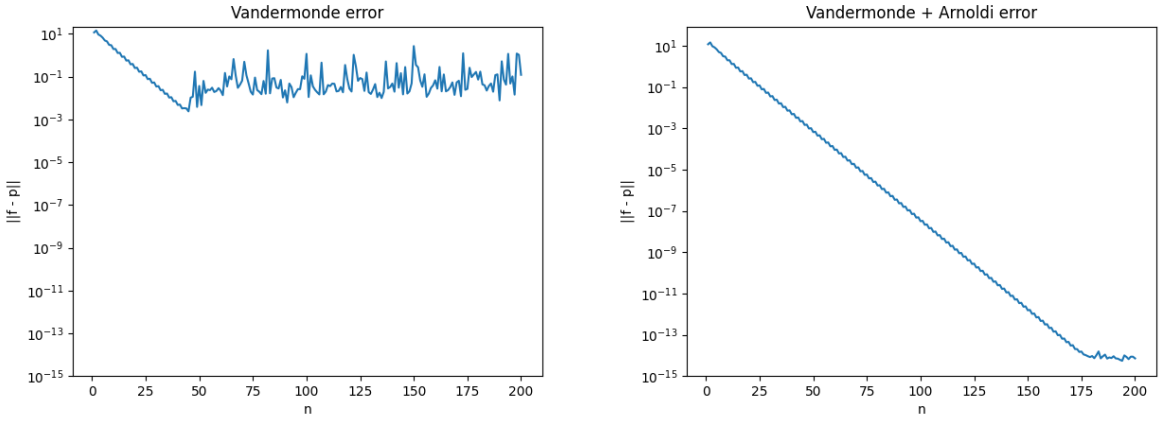

Figure 2: Errors for the Runge function $f(x) = \frac{1}{1+25x^2}$ when interpolating at Chebyshev nodes

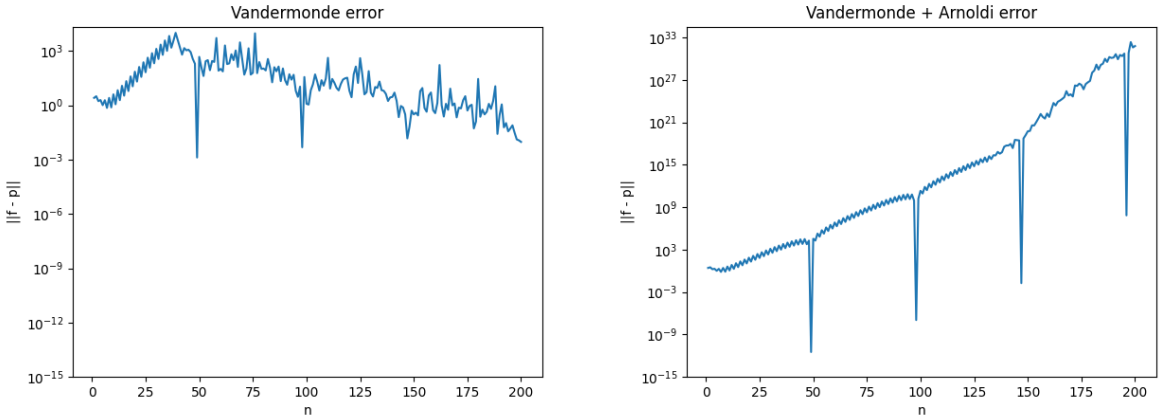

Figure 3: Errors for $f(x) = \frac{x}{a+x^2}$ with $s = 50$ for equispaced points on $[-1, 1]$

3.2 Improvement with Chebyshev nodes

A remedy is to use the *Chebyshev nodes*, roots of Chebyshev polynomials of the first kind, which have been shown to minimise the effects of Runge's phenomenon [9]. These points are $\cos(j\pi/n), 0 \leq j \leq n$. Results can be seen in Figure 2.

This is a well-conditioned problem [10]. With the Vandermonde algorithm, only 1-3 digits of accuracy are attained after around $n = 40$. With Arnoldi, we attain significantly better accuracy of up to 15 digits around $n = 175$. Both algorithms display two trends: firstly an exponential increase in accuracy until a critical value of n; secondly a range of values of $n > n_c$ for which accuracy is relatively stable with a spread. For Vandermonde this second region is centred around 10^{-1} whereas with Arnoldi this too is around 10^{-14} with a tighter spread.

3.3 Underdetermined problems

Generally one wants a high precision, that is, a large value of s, but in some cases this is not always possible, for example when sampling continuous time series. We thus briefly consider the case $s < n$, firstly equispaced points, and then Chebyshev nodes. We seek to interpolate the slightly modified function $f(x) = \frac{x}{a+x^2}$, where we take $a = 5$.

In Figure 3 we see spikes around every 50 values of n respectively. Generally, these spikes occur at integer multiples of $n = s - 1$, both the Vandermonde and VA approach. This corresponds to the situation when the interpolation points line up with the plotting points exactly, so there is a very small error.

From Figure 4 one sees that the value of n for which Vandermonde follows its initial trend is the same both for equidistant points and the Chebyshev nodes. However, this is not the case for the Arnoldi algorithm, where the critical value of n_c has been reduced considerably to just around 50, compared to around 175 for the Runge

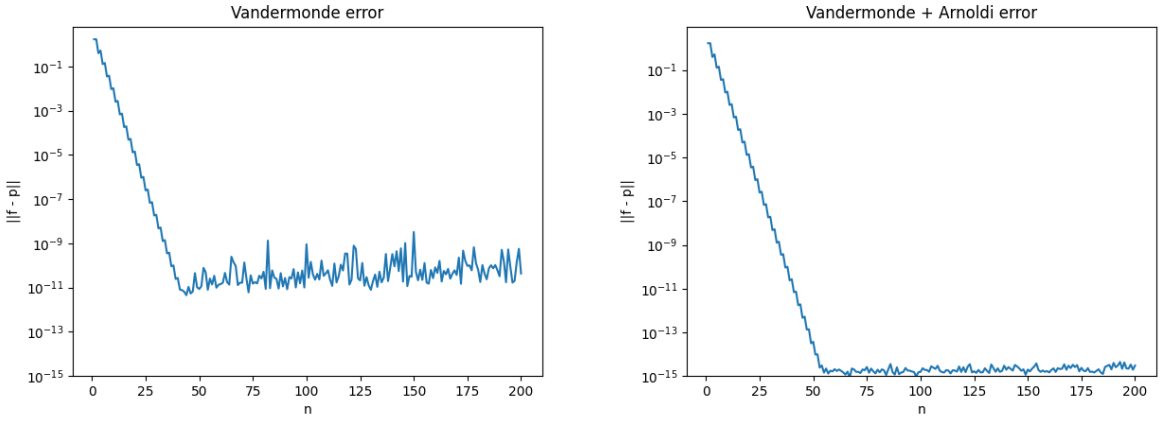

Figure 4: Errors for the $f(x) = \frac{x}{a+x^2}$ with $s = 50$ Chebyshev points on $[-1, 1]$

function (see Figure 2). Indeed one sees from the slopes of these lines that the rate of convergence is higher.

3.4 Application to mathematical biology

The theory here can be linked to *relaxation oscillators*, where a system has a *fast* one and a *slow* one. Trajectories follow a closed path in phase space and we consider time along certain trajectories on faster timescales to be approximately 0. This allows an approximation for computing the orbit period. (For further background on relaxation oscillators, see the Appendix.) Polynomial interpolation can approximate complicated curves and can lead to significantly faster computations. Here we consider a model of the cell division process [11].

$$\frac{du}{dt} = w(\alpha + u^2) - u$$
$$\frac{dw}{dt} = b(c - w(\alpha + u^2))$$

For small b, this system exhibits relation oscillations. Integrating the equation for the slow variable and using the fact that along the nullcline $w = \frac{u}{a+u^2}$ we see an expression for the period is

$$T = \frac{1}{b}\int_0^{\sqrt{a}} \frac{a - x^2}{(c - x)(a + x^2)^2} dx$$

With the VA method, accuracy is order 10^{-15}, as opposed to 10^{-9} with Vandermonde. Gaussian quadrature can then be combined with a correctly chosen interpolating polynomial to yield an exact result which has a very small error compared to the true orbit period. This is an alternative method to cubic splines, although due to the high polynomial degree and thus time required to compute it is likely not as useful.

4 Fourier extensions

We now shift our focus to extensions of Fourier series. We seek to approximate the function $f(x)$ over the interval $[-1, 1]$ using Fourier series scaled to the larger interval $[-2, 2]$. This problem generalises well to Fourier extensions in higher dimensions, referred to in the literature as *Fourier embeddings*. A change of variables shows this is a Vandermonde problem: with $z = e^{i\pi x/2}$ and $c_k = a_k + ib_k$ we have

$$f(x) \approx \sum_{k=-n}^{n} c_k' e^{ik\pi x/2} = \text{Re}\left(\sum_{k=0}^{n} c_k e^{ik\pi x/2}\right) \implies f(z) \approx \text{Re}\left(\sum_{k=0}^{n} c_k z^k\right) = \sum_{k=0}^{n} a_k \text{Re} z^k - b_k \text{Im} z^k$$

Thus we seek to find coefficients $\{a_k\}$ and $\{b_k\}$ such that we satisfy this condition in the least squares sense in m points with $m >> 2n + 1$. We have to solve

$$\text{Re}\begin{pmatrix} 1 & z_1 & z_1^2 & \dots & z_1^n \\ 1 & z_2 & z_2^2 & \dots & z_2^n \\ 1 & z_3 & z_3^2 & \dots & z_3^n \\ \vdots & \vdots & \vdots & \dots & \vdots \\ 1 & z_m & z_m^2 & \dots & z_m^n \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ \vdots \\ a_n \end{pmatrix} - \text{Im}\begin{pmatrix} 1 & z_1 & z_1^2 & \dots & z_1^n \\ 1 & z_2 & z_2^2 & \dots & z_2^n \\ 1 & z_3 & z_3^2 & \dots & z_3^n \\ \vdots & \vdots & \vdots & \dots & \vdots \\ 1 & z_m & z_m^2 & \dots & z_m^n \end{pmatrix} \begin{pmatrix} b_0 \\ b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix} \approx \begin{pmatrix} f_0 \\ f_1 \\ f_2 \\ \vdots \\ f_m \end{pmatrix}$$

To do this, we modify `polyfit` and `polyval` (and the Arnoldi counterparts) to handle the case for Fourier approximations, which is controlled through the optional argument `fourier`.

4.1 Overview of frames

The theory for such a treatment requires a generalisation of bases on a vector space to linearly dependent sets over Hilbert spaces, known as *frames*. Φ is called a *frame* for H if it satisfies the *frame condition*

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

where I is an index set. Most frames are not bases as frames are generally not ω-*independent*: there exist nonzero coefficients $\{x_n\}$ for which the sum $\sum_n x_n\phi_n$ always converges in H and satisfies $\sum_n x_n\phi_n = 0$. Bases are always ω-independent. However, $span(\Phi)$ is dense in H. By construct, frames have redundancy which gives them far greater flexibility than bases, making them easier to construct for certain problems.

For the computations later we require some more theory about frames - for a more detailed discussion on frames in Fourier extensions see [12] and [13]. For a frame Φ we have the *synthesis* operator

$$\mathcal{T} : \ell^2(I) \to H, \mathbf{y} = \{y_n\}_{n\in I} \mapsto \sum_{n\in I} y_n\phi_n$$

The *adjoint* operator is

$$\mathcal{T}^* : H \to \ell^2(I), f \mapsto \{\langle f, \phi_n\rangle\}_{n\in I}$$

The composition of these synthesis and adjoint operators is the *frame* operator

$$S = \mathcal{T}\mathcal{T}^* : H \to H, f \mapsto \sum_{n\in I}\langle f, \phi_n\rangle\phi_n$$

Finally, we have the *Gram* operator

$$\mathcal{G} = \mathcal{T}^*\mathcal{T} : \ell^2(I) \to \ell^2(I), \mathbf{x} = \{x_n\}_{n\in I} \mapsto \left\{\sum_{m\in I}\langle\phi_m, \phi_n\rangle x_m\right\}_{n\in I}$$

The Gram operator $\mathcal{G}$ is a bounded operator on $\ell^2(I)$ but not generally invertible. We can view it as the infinite matrix $G = \{\langle\phi_n, \phi_m\rangle\}_{n,m\in I}$. A frame is *tight* if $A = B$, in which case $S = A\mathcal{I}$, but $\mathcal{G}$ is not a multiple of identity, unless the frame is an orthonormal basis (for its closed linear span). A frame is *exact* if it ceases to be a frame when any one element is removed, or else it is *inexact*. A frame is *linearly independent* if every finite subset is linearly independent.

4.2 Fourier extensions

We work with $\Omega \in \mathbf{R}$, $f : \Omega \mapsto \mathbf{R}$, $f(x) = 1/(10 - 9x)$. We work with the system of functions formed by the restriction of the orthonormal Fourier basis on (-1, 1) to Ω given by

$$\mathbf{\Phi} = \{\phi_n\}_{n\in\mathbf{Z}}, \phi_n(x) = \frac{1}{\sqrt{2}}e^{\frac{in\pi x}{2}}$$

This is not an orthonormal basis when restricted to $\mathrm{L}^2(\Omega)$, but it is a tight, linearly independent frame with $A = B = 1$. Introducing the truncated frames

$$\mathbf{\Phi_N} = \{\phi_n\}_{n\in I_N}, I_n = \{n \in \mathbb{Z} : -\frac{1}{2}N \leq n < \frac{1}{2}N\}$$

then the convergence of the resulting orthogonal projections is spectral. The approximation based on this frame is the *Fourier extension*, or a *Fourier embedding* in higher dimensions.

In [14] it is shown that this frame is an example where the canonical dual frame expansion converges slowly. Since Φ is tight with $A = B = 1$ it is its own canonical dual frame, and therefore the frame coefficients are $a_n = \langle f, \phi_n\rangle$. They are precisely the Fourier coefficients of the extension $\tilde{f}$ of f by zero to $(1, 1)$:

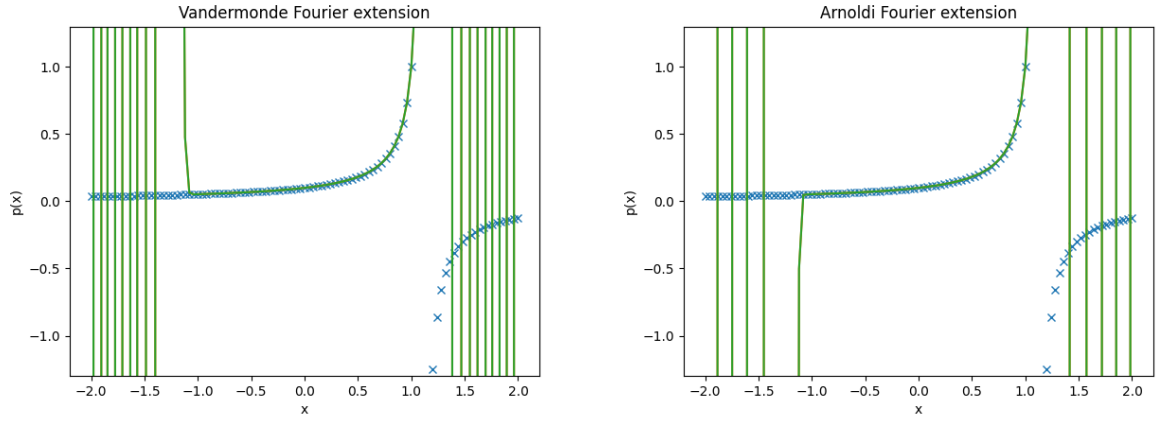

Figure 5: Degree $n = 40$ Fourier extension for $f(x) = 1/(10 - 9x)$ on $s = 50$ equispaced points on $[-1, 1]$

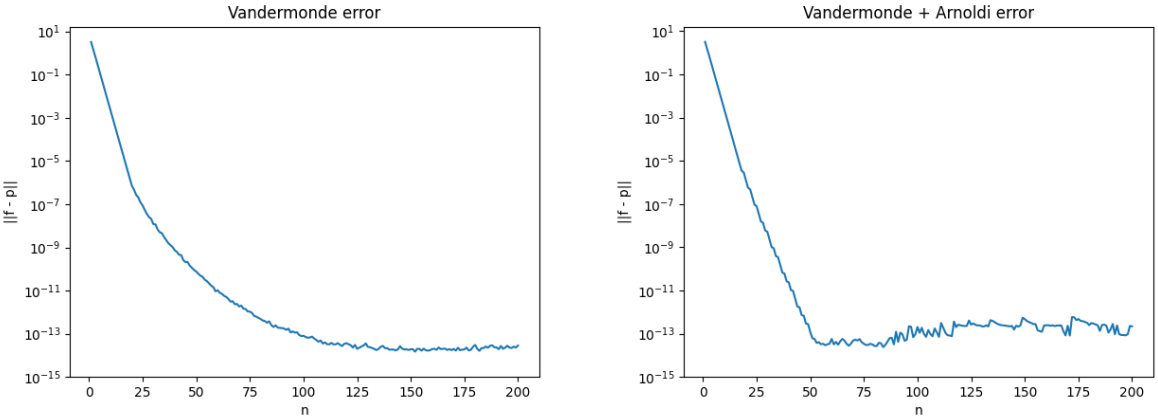

Figure 6: Fourier extension errors for Vandermonde and Arnoldi on 1000 Chebyshev points in [-1, 1] for $f(x) = 1/(10 - 9x)$

$$\langle f, \phi_n \rangle = \int_\Omega f(x)\phi_n(x)dx = \int_{(-1,1)} \tilde{f}(x)\phi_n(x)dx$$

Thus the expansion is the Fourier series of $\tilde{f}$ restricted to Ω. Unless f vanishes smoothly on the boundary $\partial\Omega$, this expansion converges slowly and suffers from Gibb's phenomenon near $\partial\Omega$. In contrast, the convergence of the best approximation $\mathcal{P}_N f$ is spectral, regardless of the shape of Ω. [14]

4.3 Vandermonde and VA

With the theory behind why this construction works, we can examine the effect of Arnoldi. Figure 5 shows the degree $n = 40$ plots for $f(x) = 1/(10 - 9x)$. We see that both are good fits, but Arnoldi has much less oscillations in the extended region.

Figure 6 shows the error plot for this extension. Despite the fact that the problem is extremely ill-conditioned (by Theorem 2, we are only on half the unit circle), we see that both algorithms provide a relatively good approximation. The theory explaining the well-behaved solution can be found in [14] and is due to properties of the frame we are working with. In both algorithms, we have exponentially improved fitting until around $n = 20$, where there is a change in behaviour. With Vandermonde we change to root exponential convergence and with Arnoldi we continue with a lower exponential rate to around machine precision. The next section investigates methods to see if we can reduce the error further.

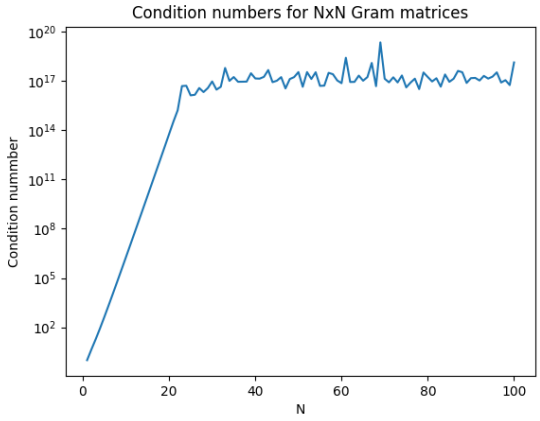

Figure 7: Gram matrix condition numbers

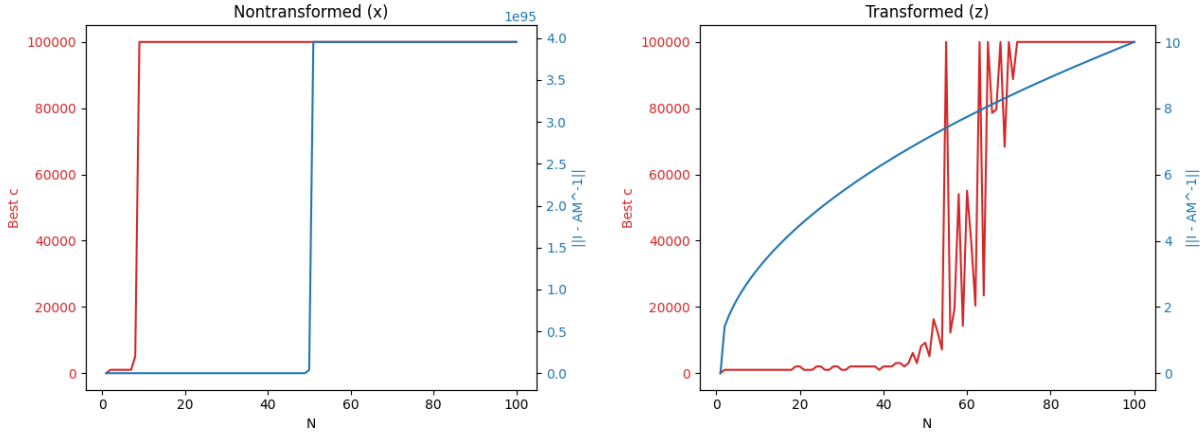

Figure 8: Distance from 1 for eigenvalues of $M^{-1}A$ (blue) and required value of c (red) for equispaced points on [-1, 1] (left) and map to unit circle (right). Required values of c are too large for this to be useful.

4.4 Error reduction

4.4.1 Preconditioning

We expect the Gram matrix G to be ill-conditioned. This is checked numerically (for finite G) in Figure 7. We calculate the entries manually first. It is clear that G is exponentially ill-conditioned to start with, as expected, and the growth slows for large N.

$$G_{nm} = \langle \phi_n, \phi_m \rangle = \frac{1}{2} \int_{-1}^{1} \exp\left(\frac{i\pi(m-n)x}{2}\right) dx = \frac{\sin\left(\frac{(m-n)\pi}{2}\right)}{\pi(m-n)} \; \forall n \neq m, \; G_{nn} = \frac{1}{2}$$

The poor conditioning of G means the system we seek to solve for Fourier extension will be ill-conditioned. Building on the idea of Question 4 in the coursework, we can seek an upper triangular conditioner $M = cU$ where U is the upper triangular part of G. If $||I - M^{-1}G|| = d < 1$ then the eigenvalues of $M^{-1}A$ are within d of 1 and we can hope for the condition number to be lower. Numerical experiments indicate that this form of preconditioner does not help in solving the problem. Figure 8 shows the optimal value of c to minimise d for both the real values and the complex ones on the unit circle. None of the values of N tested provided a sufficiently small c. The optimal value of c to minimise d grows quickly, which means that $M^{-1}A$ is has very small entries, solely because of this large scaling introduced by large c.

4.4.2 Regularisation with singular value decomposition

Since preconditioning failed, as an alternative, we try to improve the system using the truncated singular value decomposition (SVD). Since we are working in a infinite-dimensional function space, we truncate our frame Φ to Φ_N. Then the SVD $G_N = U\Sigma V^* = V\Sigma V^*$ since G_N is positive definite, and its singular values $\sigma_1, \ldots \sigma_N$

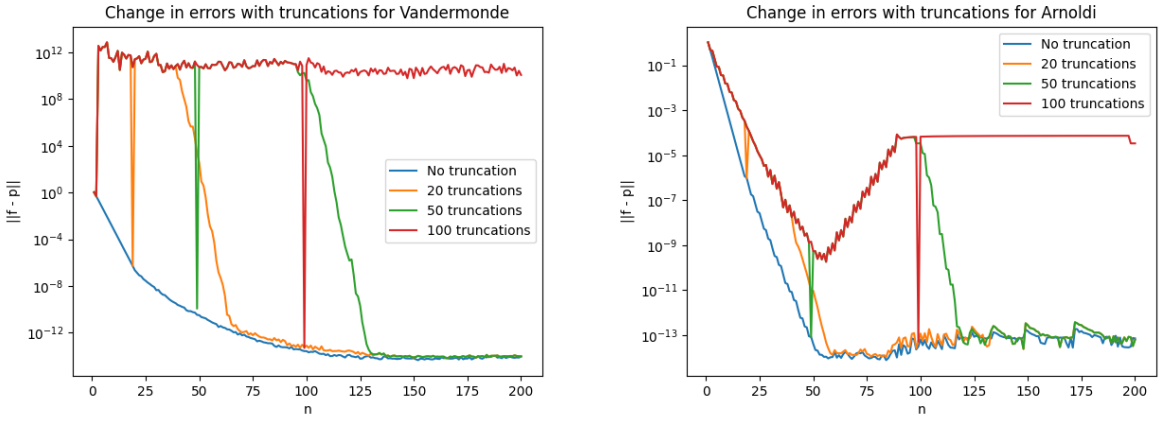

Figure 9: Error plots by regularising with simple truncated SVD. In the Arnoldi plot, the greren and blue lines overlap, and the orange and red lines overlap.

are exactly its eigenvalues. [14]

Given a tolerance $\epsilon > 0$, write Σ^ϵ as the diagonal matrix with nth entry σ_n if $\sigma_n > \epsilon$ and 0 otherwise. Define $G_N^\epsilon = V\Sigma^\epsilon V^*$. By choosing ϵ carefully, one can remove the smallest eigenvalue/s of G_N, which decreasing the condition number as the ratio of largest to smallest eigenvalue will be lower. Despite severe ill-conditioning of the Gram matrix, one still obtains convergence of the best approximation to within $\sqrt{\epsilon}$ of f. [14].

Figure 9 shows the error plots for a basic implementation of truncated SVD, with the error for Vandermonde plotted against a number of truncations. These truncations started from the smallest singular values (which are exactly the eigenvalues) and working up to larger value. As mentioned earlier, the 'spikes' are where the number of truncations is one less than the degree of the polynomial. This is due to sampling overlaps. Where the number of truncations we greater than the degree of the polynomial, sowe kept only one singular value and so we expect plots to overlap with the '100 truncation' line until n is greater than the number of truncations. The lines only diverge at degree $2n$, indicating the first n singular values are critical in the approximation.

We see that in both plots, more truncations leads to higher inital errors, as expected with the low-rank approximation. However in Vandermonde this is catastrophic for low n with errors up to 10^{12}, whereas this is not an issue with Arnoldi. Through looking at the gradient we see for higher truncation, the exponential improvement rate is higher, albeit this is only realised at higher degrees. Nevertheless, this can be used for large datasets where working with a low-rank approximation can significantly reduce computation time.

5 Conclusion

In this paper, we discussed a method of reducing error with polynomial interpolation and least squares problems by utilising Arnoldi iteration to orthogonalise at each step with Vandermonde matrices. Issues with interpolation points were considered, and we explained why the VA method can have worse errors. We then used frames to develop Fourier extensions, and investigated methods to reduce errors using preconditioning and truncated SVD.
Section 3 lends well to further investigating the differences in errors when the Arnoldi algorithm is modified to be more stable, such as through Householder triangulation. James Baglama proposed the Augmented Block Householder Arnoldi Method which can be compared to the current implementation [15]. Section 4 suggests a better preconditioner can be found, and one could investigate further the effect of other regularisation techniques on the VA approach.

6 Appendix

6.1 Relaxation oscillators in mathematical biology

For a one-dimensional system $\frac{du}{dt} = f_1(u; v)$ depending on a parameter v, call the system's *fixed points* the values u^* such that $f(u^*; v) = 0$. Fixed points are called *stable* if orbits are attracted to the fixed point, and *unstable* if orbits move away from the fixed point. Systems are said to exhibit *bistability* if for a fixed set of parameters, we

can have two stable points. This is of particular interest since it means there are multiple possibilities for steady states of a system - parameter values and initial conditions dictate which steady state will be attained. For example, in predator-prey dynamics, this could be the difference between mutual extinction and coinhabitance. Consider a one-dimensional system $\frac{du}{dt} = f_1(u; v)$. We consider the case where the system exhibits bistability for a range of parameter v. Consider turning the parameter into another variable, so that we have the system

$$\frac{du}{dt} = f_1(u, v)$$
$$\frac{dv}{dt} = \epsilon f_2(u, v)$$

for $0 < \epsilon << 1$. Then $dv/dt \approx 0$ and we consider it constant while u evolves in time and reaches its steady-state value based on $f_1(u, v) = 0$ for fixed v, and the solution follows the nullcline $f_1(u, v) = 0$. The solution moves along the nullcline until it reaches a local maximum (or minimum), at which point it leaves the nullcline and moves rapidly across phase space at a nearly constant value of v. The point at which it leaves the nullcline would correspond to the bifurcation point if v were a parameter in the one-dimensional system. When its path reaches the nullcline, it moves along it again and the process repeats. Since u evolves rapidly, we refer to it as the *fast variable*, while v is the *slow variable*.

Since certain portions of the trajectory are fast, we consider them negligible compared to the other paths. We call these *relaxation oscillations* [16] [17].

References

[1] Pablo D. Brubeck, Yuji Nakatsukasa, and Lloyd N. Trefethen. Vandermonde with arnoldi, 2019.

[2] Victor Y. Pan. How bad are vandermonde matrices?, 2015.

[3] Ake Björck and Victor Pereyra. Solution of vandermonde systems of equations. *Mathematics of Computation*, 24:893–903, 1970.

[4] Sirani M. Perera, Grigory Bonik, and Vadim Olshevsky. A fast algorithm for the inversion of quasiseparable vandermonde-like matrices, 2014.

[5] P.J. Davis. *Interpolation and Approximation*. Dover Books on Mathematics. Dover Publications, 1975.

[6] Dieter Gaier and Renate McLaughlin. *Lectures on Complex Approximation*. Birkhauser Boston Inc., USA, 1987.

[7] Carl Runge. *Über empirische Funktionen und die Interpolation zwischen äquidistanten Ordinaten*. Zeitschrift für Mathematik und Physik, 1901.

[8] Å. Björck. Numerics of gram-schmidt orthogonalization. *Linear Algebra and its Applications*, 197-198:297 – 316, 1994.

[9] Kurtis D. Fink and John H. Mathews. *Numerical Methods using MATLAB*. Prentice Hall, 1999.

[10] Lloyd N. Trefethen. *Approximation Theory and Approximation Practice (Other Titles in Applied Mathematics)*. Society for Industrial and Applied Mathematics, USA, 2012.

[11] J. J. Tyson. Modeling the cell division cycle: cdc2 and cyclin interactions. *Proc Natl Acad Sci U S A*, 88(16):7328–7332, 1991.

[12] Roel Matthysen and Daan Huybrechs. Function approximation on arbitrary domains using fourier extension frames, 2017.

[13] Ben Adcock and Daan Huybrechs. On the resolution power of fourier extensions for oscillatory functions. *Journal of Computational and Applied Mathematics*, 260:312 – 336, 2014.

[14] Ben Adcock and Daan Huybrechs. Frames and numerical approximation, 2018.

[15] James Baglama. Augmented block householder arnoldi method. *Linear Algebra and its Applications*, 429(10):2315 – 2334, 2008. Special Issue in honor of Richard S. Varga.

[16] Steven H. Strogatz. *Nonlinear Dynamics and Chaos: With Applications to Physics, Biology, Chemistry and Engineering*. Westview Press, 2000.

[17] Eric Keaveny. Mathematical biology lecture notes, 2020.