Mastery Coursework - Computational Linear Algebra

Marwan Riach, CID: 01349928

January 2021

1 Introduction

Fitting polynomials given a set of data points has widespread use from modelling environmental data to image enlargement. It is therefore important that across the entire domain over which we are interpolating, the algorithm is stable. This can be achieved two-fold: firstly, by introducing points that are not equally spaced out (e.g. Chebyshev points) and secondly by coupling the Vandermonde construction with Arnoldi orthogonalization.

We will explain the concept of the Vandermonde-Arnoldi idea, inspired by Prof Nick Trefethen's paper[1], and explore two of his examples alongside extensions to the underlying ideas through means of interpolating the temperature. We will also briefly touch upon Runge's phenomenon which partly explains the issues associated with polynomial interpolation using high degree polynomials.

2 Background

Given a set of data points, f, which corresponds to a set of points x, we seek a polynomial, $p(x)$, of degree n which approximates the relationship between the two vectors. This can be expressed as a linear system of equations in the following way:

$$\begin{pmatrix} 1 & x_1 & x_1^2 & \dots & x_1^n \\ 1 & x_2 & x_2^2 & \dots & x_2^n \\ 1 & x_3 & x_3^2 & \dots & x_3^n \\ \vdots & & & & \vdots \\ 1 & x_m & x_m^3 & \dots & x_m^n \end{pmatrix} \begin{pmatrix} c_0 \\ c_1 \\ c_2 \\ \vdots \\ c_n \end{pmatrix} \approx \begin{pmatrix} f_0 \\ f_1 \\ f_2 \\ \vdots \\ f_m \end{pmatrix},$$

$$Ac \approx f$$

$$p(x) = \sum_{k=0}^{n} c_k x^k$$

The coefficients are chosen such that $\|Ac - f\|^2$ is minimised and this is achieved by using a least squares algorithm (such as Householder). Indeed, if A is square, then the inverse is taken and we know A to be of full rank given the distinction of the points x_j.

The issue with this set up of equations arises when the matrix A is ill-conditioned. This means that a small change in the c_j values will culminate in large changes in f which is to be avoided when carrying out polynomial interpolation. The degree to which a matrix is ill-conditioned is quantifiable by means of the condition number; a lower condition number means that the matrix is ill-conditioned.

$$\kappa(A) = \frac{\sigma_{max}(A)}{\sigma_{min}(A)} = \|A\|\|A^{-1}\|$$

The condition number can either be expressed as the ratio of A's largest and smallest singular value or as the product of the operator norm of A and A^{-1}. Singular values are non-negative real numbers and are only ever equal to zero if the matrix with which we are working is not of full rank. This therefore means that all singular values that we will be dealing with in this problem are positive. As a consequence, this constrains the minimum value of $\kappa(A)$ to 1. Thus, if we can find a system of equations such that $\kappa(M) = 1$ for some matrix M, then we will have a very well conditioned problem.

2.1 Runge's Phenomenon

A possible manifestation of an ill-conditioned matrix is known as Runge's phenomenon. This phenomenon is a problem of oscillation at the peripheries of an interval that occurs when using polynomial interpolation with polynomials of high degree. This is the analogue of Gibbs phenomenon in Fourier series approximations.

This phenomenon is particularly apparent in functions of the following form[2]:

$$f(x) = \frac{1}{1+kx^2}, \; k > 0, \; x \in [-1, 1]$$

The interpolation error increases (without bound) when the degree of the interpolation polynomial is increased.

The error between the function (of the form above) and the interpolating polynomial of degree n is given and bounded by[3]:

$$\|f(x) - p(x)\| \leq \| \frac{f^{(n+1)}(\xi)}{(n+1)!} \prod_{i=0}^{n} (x - x_i) \| \tag{1}$$

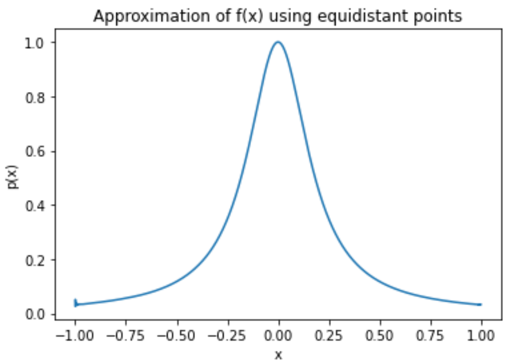
for some $\xi \in [-1, 1]$. In the case where $n+1 < m$, inequality (1) holds for any subset of size n of the m data points.

This therefore means that we seek to minimise the product term on the RHS of the inequality in (1). Choosing equidistant points means that $\frac{f^{(n+1)}(\xi)}{(n+1)!} \prod_{i=0}^{n} (x - x_i)$ behaves very differently for a different set of points x_i and will blow up at the edges as n is increased.

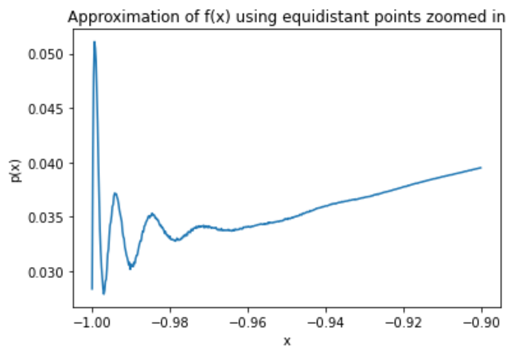
This effect can be mitigated by using Chebyshev nodes which are defined as follows:

$$x_i = \cos(i\pi/n), \; i \in \{0, 1, \ldots, n\}$$

Figures 1 and 2 show how using Chebyshev points improves the effect of Runge's phenomenon for the function $f(x) = \frac{1}{1+30x^2}$. The householder least squares algorithm was used to find the coefficients, c.

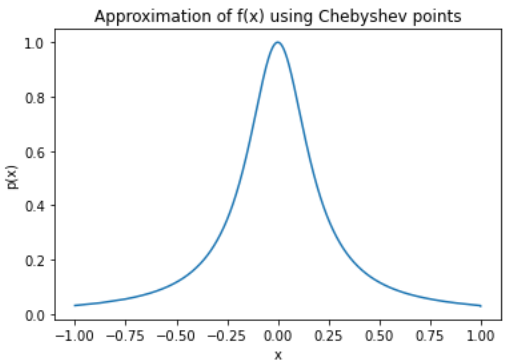


(a) Polynomial of degree 300

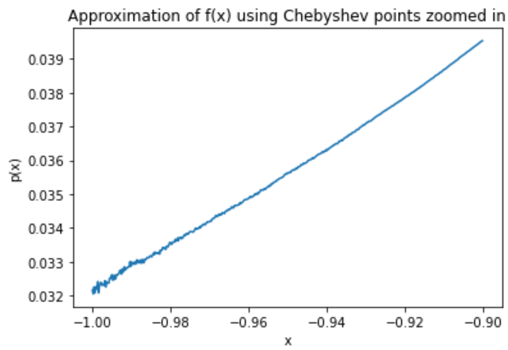


(b) Polynomial of degree 300 zoomed in at left boundary

Figure 1: Plots using 1000 equidistant data points



(a) Polynomial of degree 300



(b) Polynomial of degree 300 zoomed in at left boundary

Figure 2: Plots using 1000 Chebyshev data points

Runge's phenomenon is far more defined when equidistant data points are used as compared with Chebyshev points (as can be seen from figures 1 and 2). However, there is still a noticeable degree of perturbation at the peripheries even when Chebyshev polynomials are used, suggesting scope for improvement.

2.2 Vandermonde-Arnoldi

The next improvement comes in the form of the Vandermonde-Arnoldi algorithm. The point here is that even when the data points used are Chebyshev, the matrix A is still poorly conditioned, and so we want to find a system of equations $Md = f$ where M is optimally conditioned.

The span of the matrix A is equal to the Krylov subspace generated by the diagonal m by m matrix X (where $X_{i,i} = x_i$) and the vector b of ones:

$$\mathcal{K}_{n+1}(X, b) = \text{span}\{b, Xb, X^2b, \ldots, X^nb\} = span\{\begin{pmatrix}1\\1\\\vdots\\1\end{pmatrix}, \begin{pmatrix}x_1\\x_2\\\vdots\\x_m\end{pmatrix}, \begin{pmatrix}x_1^2\\x_2^2\\\vdots\\x_m^2\end{pmatrix}, \ldots, \begin{pmatrix}x_1^n\\x_2^n\\\vdots\\x_m^n\end{pmatrix}\}$$

Therefore, A is in fact a Krylov matrix taking the following form:

$$A = K_{n+1} = \begin{bmatrix} b & Xb & X^2b & \cdots & X^nb \end{bmatrix} = Q_{n+1}R_{n+1}$$

A has thus been expressed in a QR factorisation format with Q being orthogonal and R being upper triangular. Q_{n+1} is obtained by completing n steps of the Arnoldi iteration which returns an upper Hessenberg matrix, H_n, with which X is similar by orthogonal similarity transforms satisfying:

$$XQ_n = Q_{n+1}H_n$$

with Q_i denoting the first i columns of Q and H_i denoting the $(i + 1) \times i$ upper left hand section of H.

A and R_{n+1} are ill conditioned, and so we want to avoid directly dealing with these matrices. The Vandermonde-Arnoldi algorithm avoids these ill conditioned matrices by setting $d = R_{n+1}c$. This culminates in a new problem:

$$\begin{aligned} Ac &\approx f \\ \Rightarrow Q_{n+1}R_{n+1}c &\approx f \\ \Rightarrow Q_{n+1}d &\approx f \end{aligned}$$

This new problem is much better conditioned, as we are now dealing with an orthogonal matrix. The condition number of all orthogonal matrices is 1, which is the minimum condition number that can be obtained (as previously discussed).

To see this, we will consider the definition of the condition number in terms of its operator norm and work out the operator norm of an orthogonal matrix from its original definition:

$$\|Q\| = \sup_{\mathbf{x}\neq\mathbf{0}} \frac{\|Q\mathbf{x}\|}{\|\mathbf{x}\|} = \sup_{\mathbf{x}\neq\mathbf{0}} \frac{\|\mathbf{x}\|}{\|\mathbf{x}\|} = 1$$

Orthogonal matrices form a group and hence it follows that $\|Q^{-1}\|$ is also equal to 1 by the inverse axiom of group theory. Ergo, $\kappa(Q) = \|Q\|\|Q^{-1}\| = 1$ rendering the new problem to solve for d extremely well conditioned.

With d having been obtained through a least squares method (such as Householder), it remains to fit the polynomial to a set of given points, s, to obtain an output, y. The relationship between s and y is as follows:

$$\begin{pmatrix} 1 & s_1 & s_1^2 & \dots & s_1^n \\ 1 & s_2 & s_2^2 & \dots & s_2^n \\ 1 & s_3 & s_3^2 & \dots & s_3^n \\ \vdots & & & & \vdots \\ 1 & s_M & s_M^3 & \dots & s_M^n \end{pmatrix} \begin{pmatrix} c_0 \\ c_1 \\ c_2 \\ \vdots \\ c_n \end{pmatrix} = \begin{pmatrix} y_0 \\ y_1 \\ y_2 \\ \vdots \\ y_M \end{pmatrix}$$
$$Bc = y$$

This works in terms of c however we want to express this in terms of d since that is the solution we have been working with.

The matrix B has a very similar form to A and thus we can write it as a Krylov matrix as follows:

$$B = K'_{n+1} = \begin{bmatrix} b & Sb & S^2b & \cdots & S^nb \end{bmatrix}$$

with S being a diagonal M by M matrix (where $S_{i,i} = s_i$) and b is the vector of ones.

If we can find a matrix W such that:

$$SW_n = W_{n+1}H_n \tag{2}$$

with H_n being fixed from the previous step using the X matrix.

If S satisfies (2), then this implies that B can be written as follows and implies the following (the same R_{n+1} is used from the previous step):

$$\begin{aligned} B &= W_{n+1}R_{n+1} \\ \Rightarrow Bc &= W_{n+1}R_{n+1}c \\ \Rightarrow Bc &= W_{n+1}d = y \end{aligned}$$

Thus, the Vandermonde-Arnoldi algorithm ensures that the ill conditioned matrices A and R_{n+1} are never utilised in the obtaining of y. W_{n+1} is constructed by setting its first columns to be ones and thereafter it is constructed using H_{n+1} such that (2) is satisfied; W is not necessarily orthogonal, albeit approximately so[1].

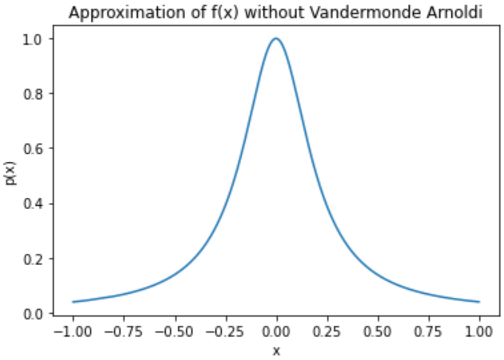
3 Examples

3.1 Example 1 from Trefethen's Paper

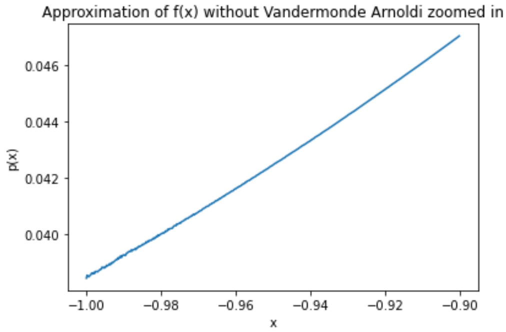
To see the improvement of the Vandermonde-Arnoldi algorithm on producing accurate interpolation, we will look at the Runge function. This a special case of the aforementioned $f(x) = \frac{1}{1+kx^2}$ when $k = 25$.

Chebyshev points will be used for the construction of both polynomials, one of which will use Vandermonde-Arnoldi and the other will use standard least squares.

Figures 3 and 4 demonstrate that Runge's phenomenon is further mitigated when using the Vandermonde Arnoldi method. Specifically, the minor oscillations in figure 3b disappear in figure 4b effectively eliminating Runge's phenomenon.

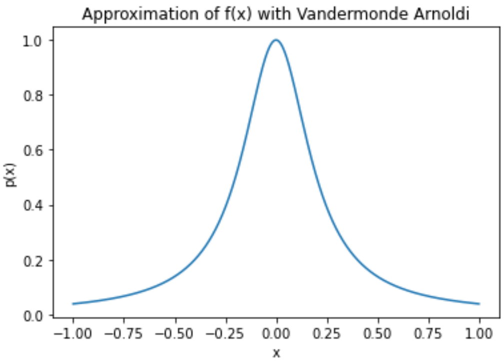


(a) Polynomial of degree 100

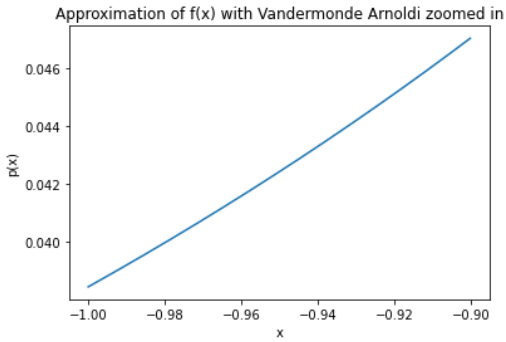


(b) Polynomial of degree 100 zoomed in at left boundary

Figure 3: Plots using 1000 Chebyshev data points and least squares method



(a) Polynomial of degree 100



(b) Polynomial of degree 100 zoomed in at left boundary

Figure 4: Plots using 1000 Chebyshev data points and Vandermonde Arnoldi algorithm

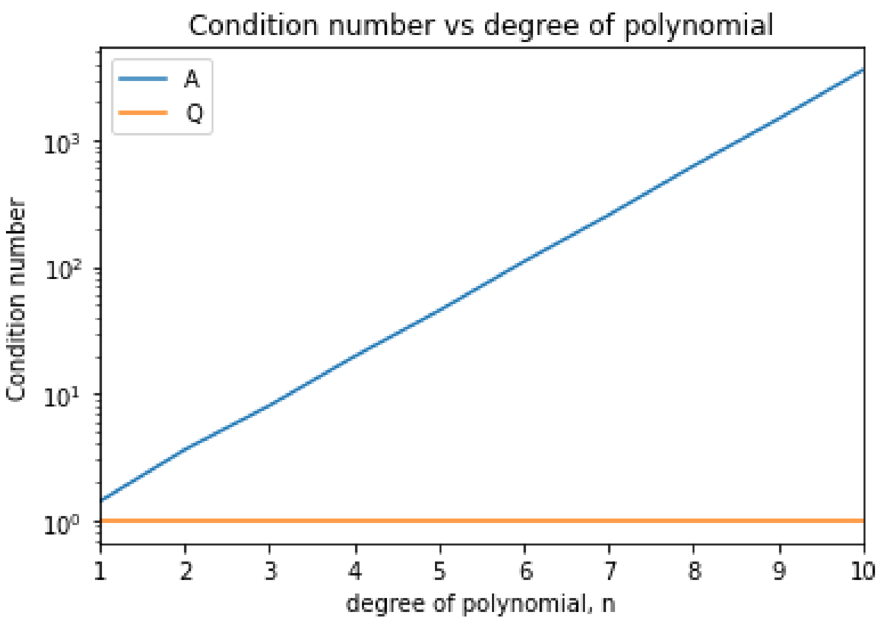


Figure 5: Condition number of Q_{n+1} and A vs the degree of the interpolating polynomial

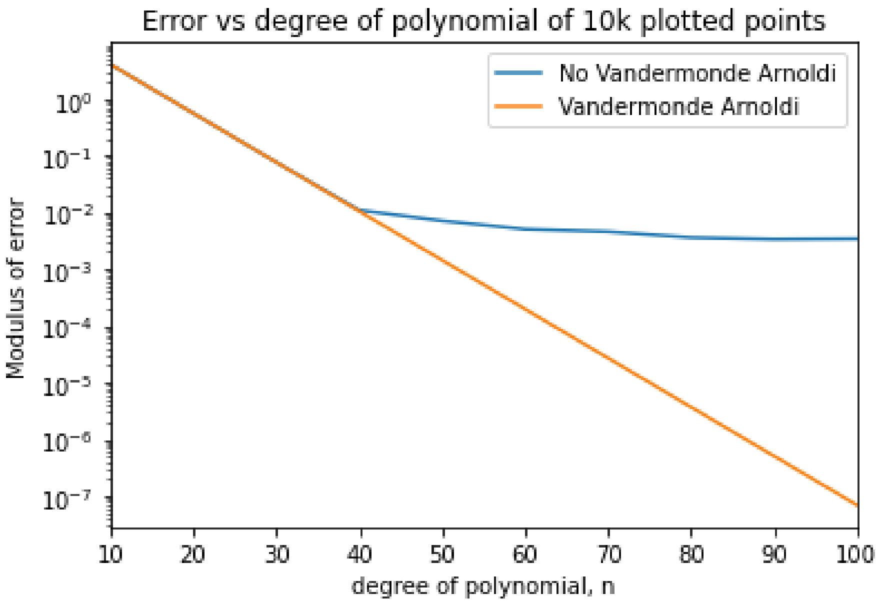


Figure 6: Error of interpolating polynomial against the degree of the polynomial for the two methods

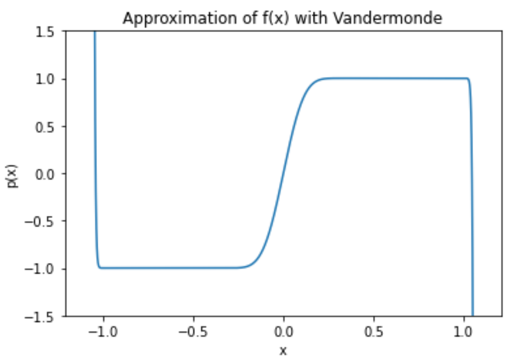
Figure 5 shows that A is exponentially ill conditioned as the degree of the interpolating polynomial is increased, as demonstrated by the linear trajectory on a log-scale. Furthermore, one can also see that Q has constant condition number 1, corroborating the proof earlier on.

Figure 6 shows us how the error decreases with the increasing polynomial degree for both methods, with the Vandermonde Arnoldi far surpassing the other method in precision beyond a degree of approximately 40.

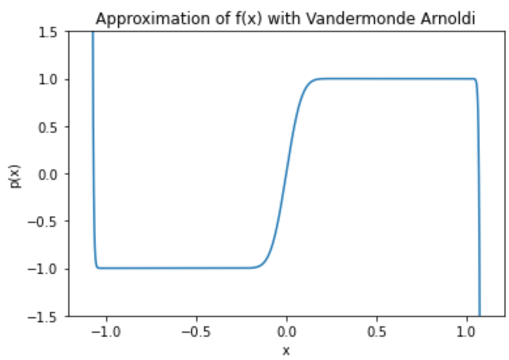
Trefethen's paper[1] discusses how the FLOP complexity of Vandermonde-Arnoldi is greater than the complexity of Vandermonde alone. This would therefore suggest that for polynomials of degree less than 40, the method without Vandermonde-Arnoldi would be preferable given its similarity of precision and improved speed ($O(Mn) + O(mn^2)$ vs $O(Mn^2) + O(mn^2)$ for Vandermonde and Vandermonde Arnoldi respectively).

3.2 Example 2 from Trefethen's Paper

Here we observe the *sign* function when it is approximated using the Vandermonde and Vandermonde Arnoldi methods on the domain [-1,1] with the errors being measured in each half of the domain, omitting the central region where the spike occurs. The data points are 500 equispaced points each in the two intervals [-1, -1/3] and [1/3,1] akin to Trefethen's report.



(a) Polynomial of degree 75 using Vandermonde method



(b) Polynomial of degree 75 using Arnoldi Vandermonde method

Figure 7: Plots of sign function approximation using two different methods

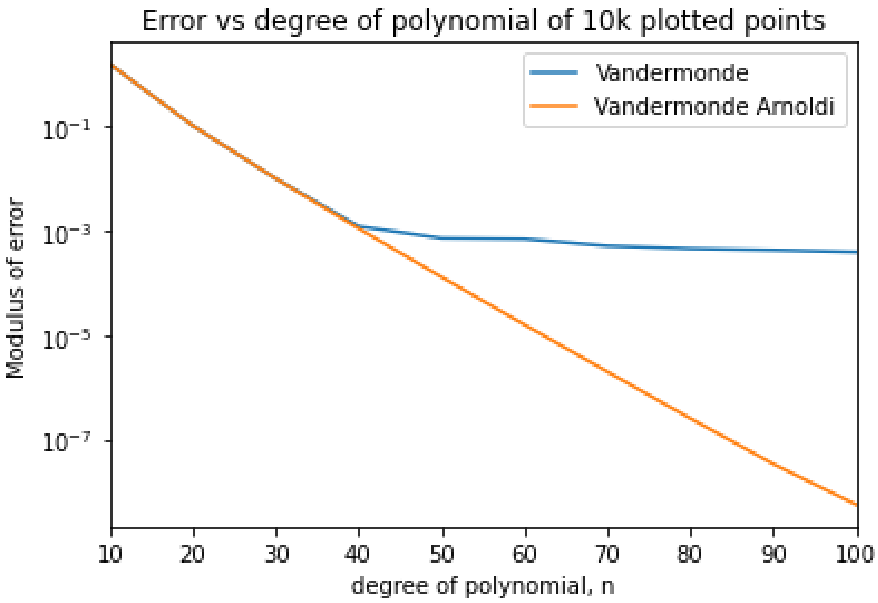


Figure 8: Error of interpolating polynomial against the degree of the polynomial for the two methods of approxiamting the sign function

Figure 7a as compared with figure 7b spikes at a lower value of x making it a less effective replica of the sign function. This is supported by figure 8 showing yet again Vandermonde-Arnoldi's superiority for polynomial approximations where the degree exceeds ~ 40.

3.3 Weather Forecasting Example

The data used to interpolate the hourly 2m elevated temperature in London from January 1st 00:00 hrs to January 11th 23:00 hrs was obtained from meteoblue's interactive database[4].

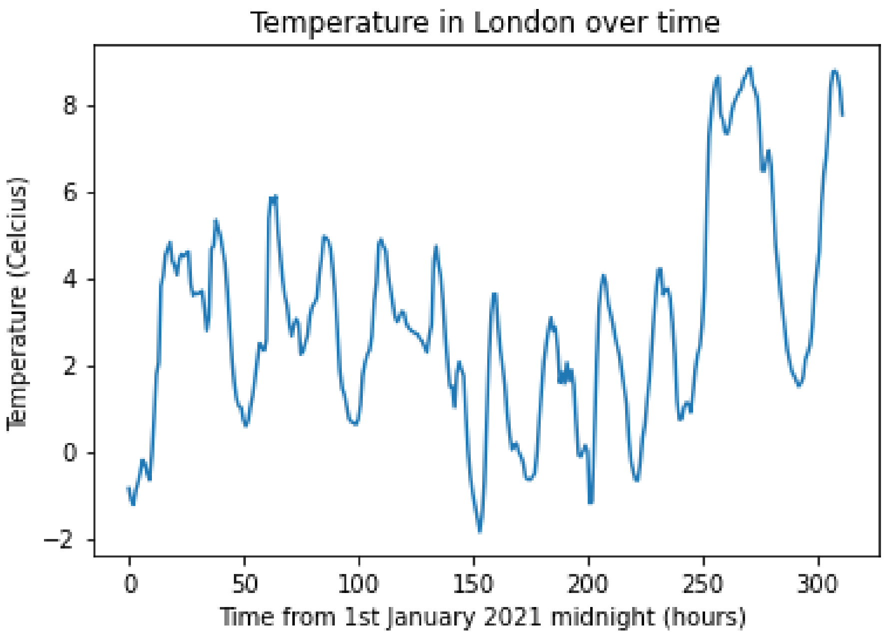
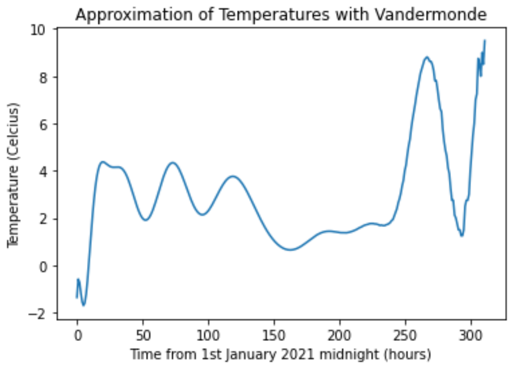
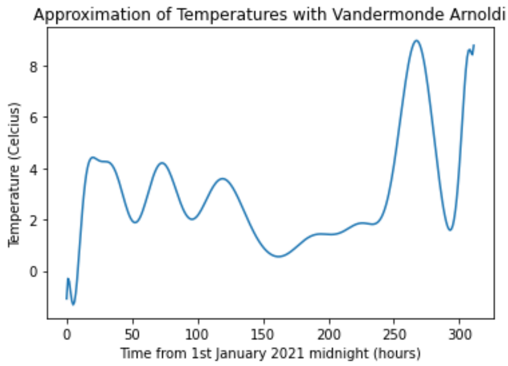


Figure 9: Temperature in London from January 1st 00:00 hrs to January 11th 23:00 hrs (Celcius) at 2m above sea level

We will attempt to find a polynomial that can interpolate the time series of temperature in figure 9. In an ideal world, we would ensure that the points are not uniformly apart to mitigate Runge's phenomenon, but practically, it makes sense to use data points that are evenly spaced out to capture the volatility of the temperature over time. The data points that are used therefore extract every other point from the time series, thus halving the amount of data relative to the original data set.



(a) Polynomial of degree 25 using Vandermonde method



(b) Polynomial of degree 25 using Vandermonde Arnoldi

Figure 10: Approximation of temperature over time using different methods and data from every other hour as opposed to every hour

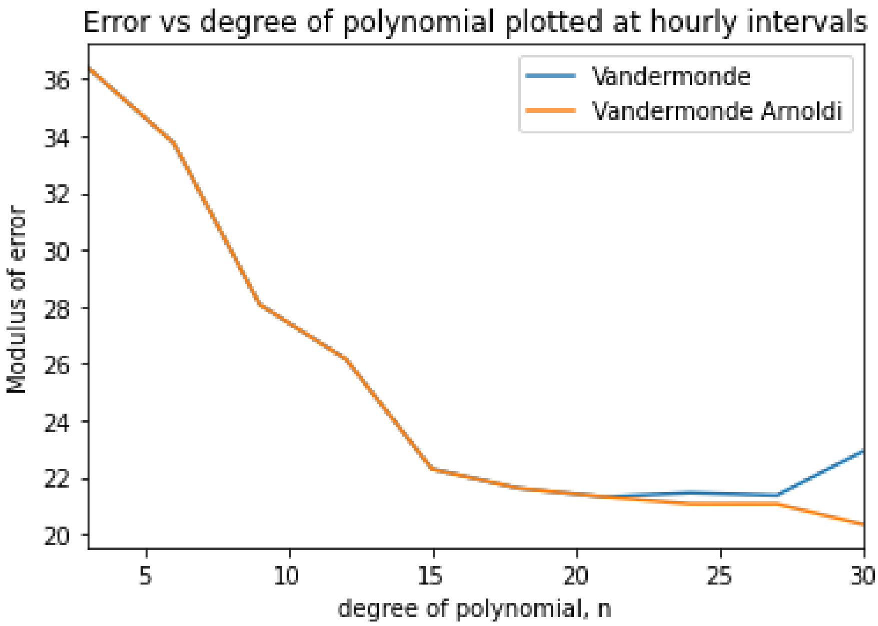


Figure 11: Error of temperature approximation vs true temperature vs degree of polynomial for two different methods

Even with this practical example, Runge's phenomenon can be seen in figure 10a during the final hours; this is mitigated by the Vandermonde Arnoldi algorithm as it cannot be seen in figure 10b. Both figures ape the true time series reasonably well with all of the prominent peaks and troughs being accounted for in the plots.

Figure 11 shows us that the Vandermonde Arnoldi always performs as well as the Vandermonde algorithm, however due to the volatile nature of temperature, it is difficult to obtain a small error, even with frequent intervals.

For the most part, a polynomial of degree ~ 25 which uses the Vandermonde Arnoldi is appropriate for fitting this time series with fewer data points. This could therefore be used in industry so as to reduce the amount of time the temperature needs to be recorded, potentially saving weather companies manpower.

4 Discussion

When used as a method to compute a least squares solution to a linear system of equations, the Vandermonde-Arnoldi method is more accurate and less sensitive to perturbations in the data points than the Vandermonde method. With this said, the Vandermonde-Arnoldi method takes longer to execute than its stripped back counterpart and therefore when interpolating a polynomial of low degree, the Vandermonde method is preferable given its similarity of accuracy for low degree polynomials. For polynomials of high degree and improved precision, turning to the Vandermonde-Arnoldi method is well advised.

References

[1] Vandermonde with Arnoldi by Lloyd N. Trefethen
`https://people.maths.ox.ac.uk/trefethen/vandermonde.pdf`

[2] Theory behind Runge's phenomenon
`http://www.tlu.ee/ tonu/Arvmeet/Runge'sphenomenon.pdf`

[3] Errors in polynomial interpolation
`https://math.okstate.edu/people/binegar/4513-F98/4513-l16.pdf`

[4] Temperature data source
`https://www.meteoblue.com/en/weather/archive/export/`