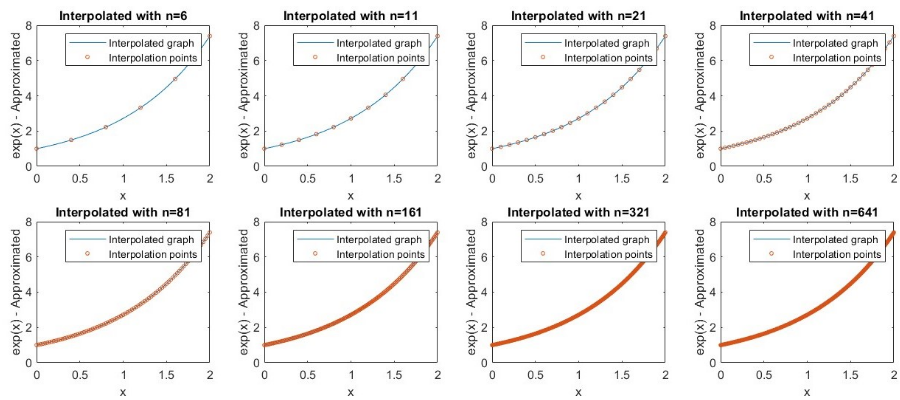
CS 6210 Assignment 1

Artem Koval

September 2024

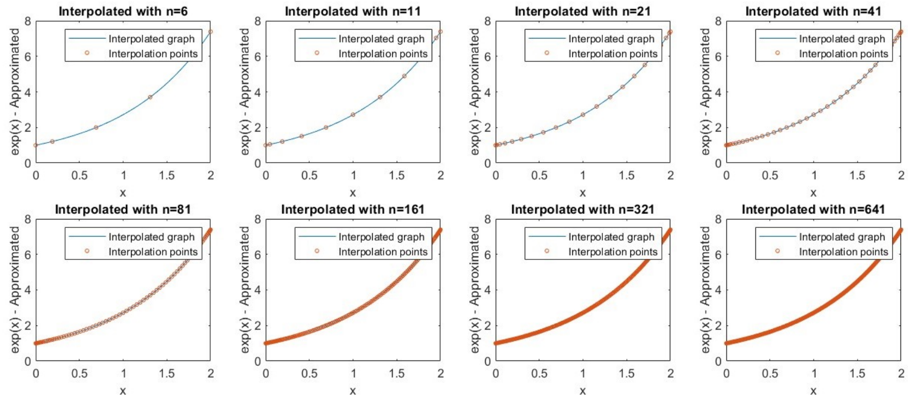
1 Interpolation using Vandermonde matrix

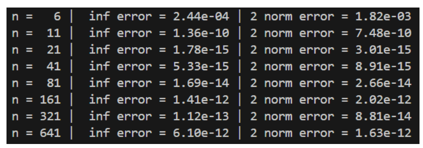
Below are the results of using the Vandermonde matrix to calculate an interpolating polynomial to exp(x) with evenly and Chebyshev spaced points, as well as the associated errors:




```
n =   6 |  inf error = 3.05e-04 | 2 norm error = 1.62e-03
n =  11 |  inf error = 6.39e-10 | 2 norm error = 1.68e-09
n =  21 |  inf error = 1.71e-13 | 2 norm error = 2.24e-13
n =  41 |  inf error = 1.93e-12 | 2 norm error = 9.47e-13
n =  81 |  inf error = 1.61e-12 | 2 norm error = 3.90e-13
n = 161 |  inf error = 9.93e-12 | 2 norm error = 1.28e-12
n = 321 |  inf error = 2.11e-11 | 2 norm error = 1.29e-12
n = 641 |  inf error = 5.83e-12 | 2 norm error = 2.51e-13
```

Figure 1: Interpolation using Vandermonde (Evenly-Spaced) Plots and Errors





```
n =   6 |  inf error = 2.44e-04 | 2 norm error = 1.82e-03
n =  11 |  inf error = 1.36e-10 | 2 norm error = 7.48e-10
n =  21 |  inf error = 1.78e-15 | 2 norm error = 3.01e-15
n =  41 |  inf error = 5.33e-15 | 2 norm error = 8.91e-15
n =  81 |  inf error = 1.69e-14 | 2 norm error = 2.66e-14
n = 161 |  inf error = 1.41e-12 | 2 norm error = 2.02e-12
n = 321 |  inf error = 1.12e-13 | 2 norm error = 8.81e-14
n = 641 |  inf error = 6.10e-12 | 2 norm error = 1.63e-12
```

Figure 2: Interpolation using Vandermonde (Chebyshev-Spaced) Plots and Errors

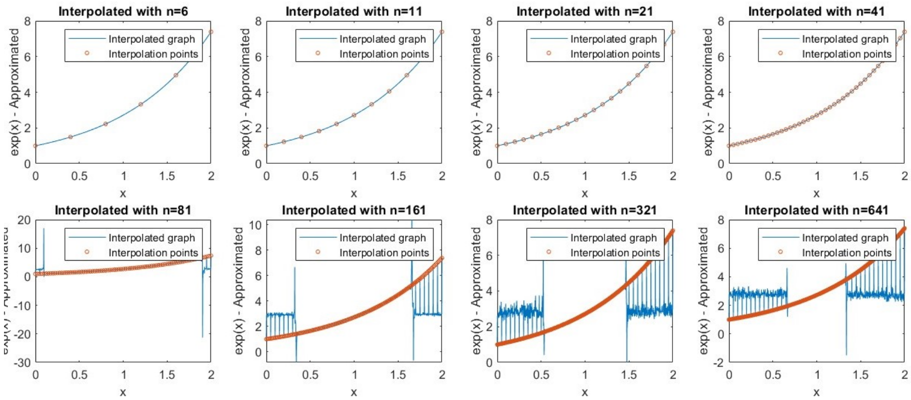
When interpolating the function $y = e^x$ on the interval [0,2] using the Vandermonde matrix, the results tend to get more accurate as the value of **n** (the number of points initially selected for interpolation) gets larger. This trend stops at about **n=21**, where the algorithm seems to reach peak accuracy, and increasing **n** at this point leads to larger errors. The results are expected and consistent with our observations in class. The Vandermonde systems seem to be hard to solve accurately as they grow in size.

When comparing the use of evenly or Chebyshev spaced points, the use of Chebyshev points provides more accurate results at both high and low values of n, with the minimum error reaching 1.78e-15 (inf norm) and 3.01e-15 (2nd norm) at **n=21**.

Overall, interpolation using the Vandermonde matrix produces valid and accurate results for smaller datasets (**n<21**), but produces less accurate results for large datasets.

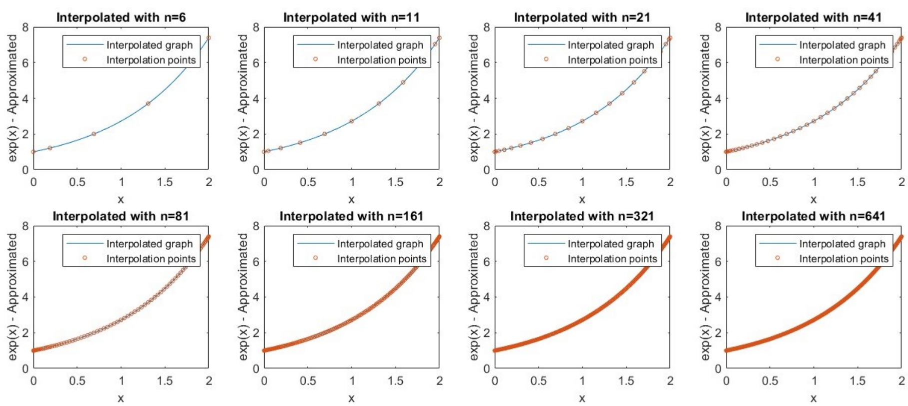
2 Lagrange Barycentric Interpolation

Below are the results of using the *bary* and *baryweights* routines to accomplish Lagrange interpolation:




```
n =   6 |  inf error = 3.05e-04 | 2 norm error = 1.62e-03
n =  11 |  inf error = 6.39e-10 | 2 norm error = 1.68e-09
n =  21 |  inf error = 1.10e-11 | 2 norm error = 9.67e-12
n =  41 |  inf error = 2.03e-06 | 2 norm error = 8.78e-07
n =  81 |  inf error = 2.80e+01 | 2 norm error = 5.15e+00
n = 161 |  inf error = 6.04e+00 | 2 norm error = 3.89e+00
n = 321 |  inf error = 4.69e+00 | 2 norm error = 3.03e+00
n = 641 |  inf error = 5.31e+00 | 2 norm error = 2.29e+00
```

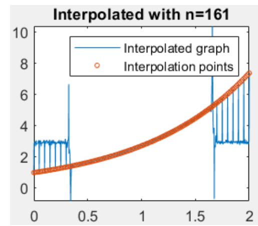
Figure 3: Barycentric Lagrange Interpolation (Evenly-Spaced) Plots and Errors




```
n =   6 |  inf error = 2.44e-04 | 2 norm error = 1.82e-03
n =  11 |  inf error = 1.36e-10 | 2 norm error = 7.48e-10
n =  21 |  inf error = 2.66e-15 | 2 norm error = 5.24e-15
n =  41 |  inf error = 5.33e-15 | 2 norm error = 8.67e-15
n =  81 |  inf error = 5.33e-15 | 2 norm error = 5.70e-15
n = 161 |  inf error = 7.99e-15 | 2 norm error = 5.52e-15
n = 321 |  inf error = 7.11e-15 | 2 norm error = 4.03e-15
n = 641 |  inf error = 7.99e-15 | 2 norm error = 2.35e-15
```

Figure 4: Barycentric Lagrange Interpolation (Chebyshev-Spaced) Plots and Errors

The Barycentric algorithm that uses evenly-spaced points (Figure 3) performed poorly at **n>21**, where significant errors appear towards the edges of the plot. This issue gets resolved when Chebyshev-spaced points are used. This phenomenon (Runge's phenomenon) is consistent with our observations in the class, and has to deal with how the equations are constructed. When using evenly-spaced points, oscillations are large towards the end of the interval because of the increasing form of the basis functions.

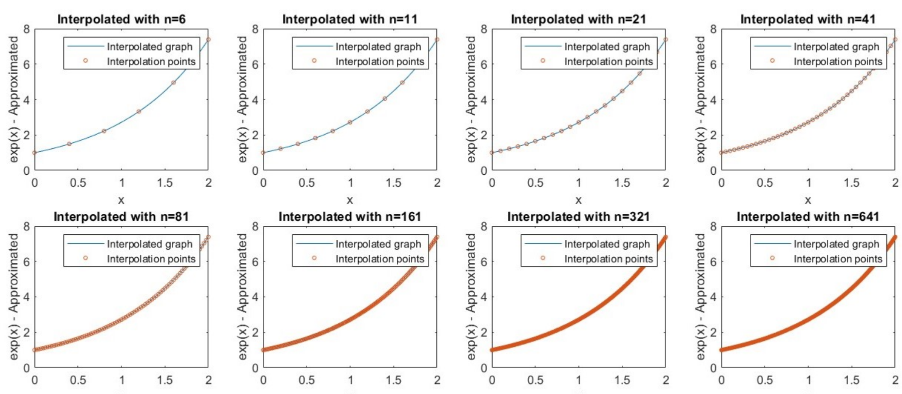


Another takeaway is that Barycentric Lagrange interpolation using Chebyshev points performed better than interpolation using the Vandermonde matrix at high **n**. Additionally, the error stays consistently low for **n<21**. Overall, Lagrange interpolation is more accurate for larger datasets, but care is needed when selecting the data points.

3 Spline Interpolation

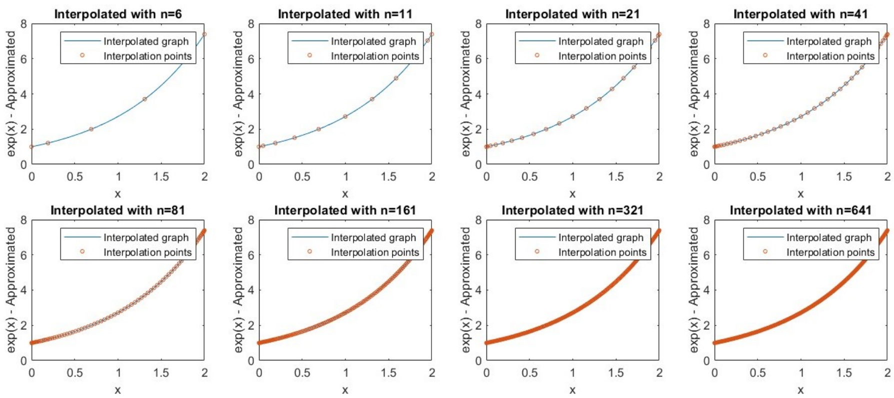
3.1 Using Spline Toolbox

Below are the results of using the MATLAB Spline toolbox with both even and Chebyshev points (6th-order splines were used in my implementation):




```
n =   6 |  inf error = 3.05e-04 | 2 norm error = 1.62e-03
n =  11 |  inf error = 3.77e-06 | 2 norm error = 8.14e-06
n =  21 |  inf error = 6.93e-08 | 2 norm error = 7.54e-08
n =  41 |  inf error = 1.18e-09 | 2 norm error = 6.45e-10
n =  81 |  inf error = 1.92e-11 | 2 norm error = 5.28e-12
n = 161 |  inf error = 3.06e-13 | 2 norm error = 4.21e-14
n = 321 |  inf error = 5.33e-15 | 2 norm error = 1.64e-15
n = 641 |  inf error = 6.22e-15 | 2 norm error = 1.07e-15
```

Figure 5: Interpolation using Spline routine (Evenly-Spaced) Plots and Errors




```
n =   6 |  inf error = 2.44e-04 | 2 norm error = 1.82e-03
n =  11 |  inf error = 6.01e-07 | 2 norm error = 2.29e-06
n =  21 |  inf error = 2.25e-09 | 2 norm error = 5.17e-09
n =  41 |  inf error = 4.14e-11 | 2 norm error = 7.10e-11
n =  81 |  inf error = 6.89e-13 | 2 norm error = 8.56e-13
n = 161 |  inf error = 1.29e-14 | 2 norm error = 9.86e-15
n = 321 |  inf error = 5.33e-15 | 2 norm error = 1.70e-15
n = 641 |  inf error = 5.33e-15 | 2 norm error = 1.25e-15
```

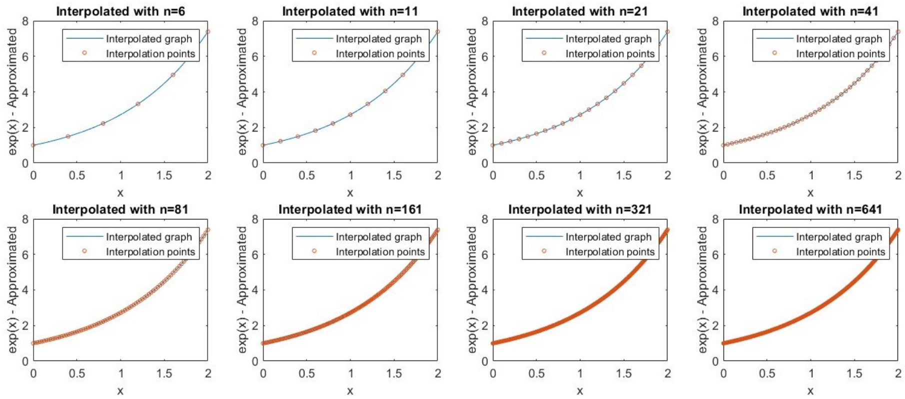
Figure 6: Interpolation using Spline routine (Chebyshev-Spaced) Plots and Errors

The spline routine produced valid results when using evenly-spaced and Chebyshev-spaced points. Importantly, as the number of points n increased, the accuracy of the result also increased, making the spline routine a great choice for interpolating large datasets.

Another key takeaway from using the spline routine is that the choice of points (evenly or Chebyshev) did not matter as much as it did for Lagrange interpolation. Both approaches produced more accurate results for higher values of **n**.

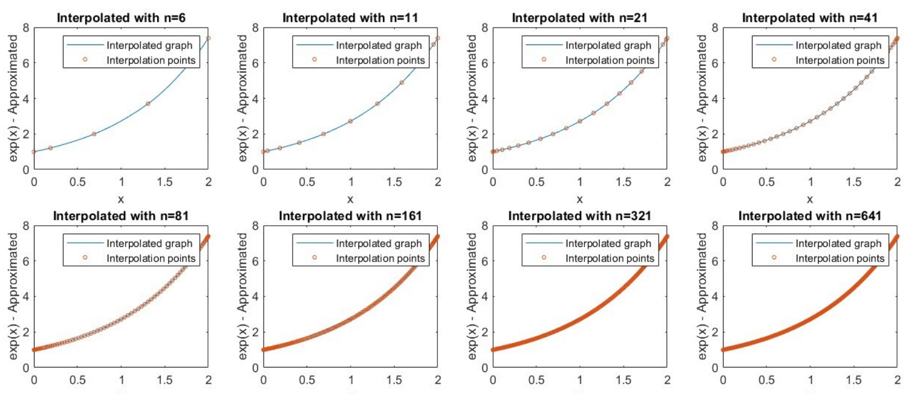
3.2 Using PCHIP Toolbox

Below are the results of using the MATLAB PCHIP toolbox with both even and Chebyshev points:




```
n =   6 |  inf error = 1.54e-02 | 2 norm error = 5.80e-02
n =  11 |  inf error = 2.23e-03 | 2 norm error = 4.42e-03
n =  21 |  inf error = 3.00e-04 | 2 norm error = 3.17e-04
n =  41 |  inf error = 3.88e-05 | 2 norm error = 2.26e-05
n =  81 |  inf error = 4.93e-06 | 2 norm error = 1.65e-06
n = 161 |  inf error = 6.22e-07 | 2 norm error = 1.27e-07
n = 321 |  inf error = 7.81e-08 | 2 norm error = 1.02e-08
n = 641 |  inf error = 3.46e-09 | 2 norm error = 8.13e-10
```

Figure 7: Interpolation using PCHIP routine (Evenly-Spaced) Plots and Errors




```
n =   6 |  inf error = 2.09e-02 | 2 norm error = 1.04e-01
n =  11 |  inf error = 2.24e-03 | 2 norm error = 8.05e-03
n =  21 |  inf error = 2.61e-04 | 2 norm error = 6.71e-04
n =  41 |  inf error = 3.08e-05 | 2 norm error = 5.85e-05
n =  81 |  inf error = 3.72e-06 | 2 norm error = 5.17e-06
n = 161 |  inf error = 4.60e-07 | 2 norm error = 4.58e-07
n = 321 |  inf error = 5.70e-08 | 2 norm error = 4.04e-08
n = 641 |  inf error = 7.09e-09 | 2 norm error = 3.56e-09
```

Figure 8: Interpolation using PCHIP routine (Chebyshev-Spaced) Plots and Errors

Interpolating using the PCHIP toolbox produced more accurate results for higher values of **n**. The PCHIP routine produced valid results for both evenly and Chebyshev spaced points; however, it is important to note that the routine produced less accurate results than the previous interpolation methods. This outcome is expected since PCHIP is designed for cases with additional constraints. PCHIP constrains the solution so that the polynomials are bounded by the data points, which causes less accurate results for smooth functions like $y = e^x$.

Another observation is that the choice of points (evenly or Chebyshev) did not significantly impact the accuracy of the PCHIP routine.

4 Measuring Time and Errors

4.1 Execution Time Comparison

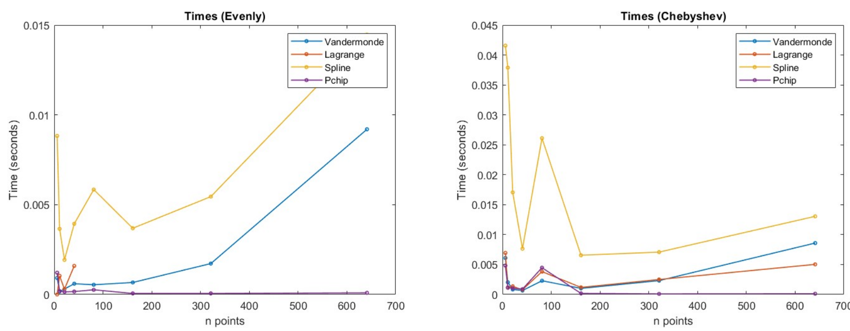


Figure 9: Execution times of interpolation methods

The four different methods shown on the plot are Vandermonde, Lagrange (using bary/baryweights), Spline, and PCHIP routines using evenly spaced and Chebyshev points. The results of measuring the execution time are seen in Figure 9 above. Please note that I excluded the plots of Lagrange interpolation with evenly-spaced points for high **n** due to Runge's phenomenon.

The figure shows that the spline routine took the longest to execute at any number of points. I found it unsurprising since I used a 6th-order spline in my program. To test my observation, I changed my program to use a 3-rd order spline instead, which shortened the execution time and reduced the accuracy of the spline routine.

Interpolation using the Vandermonde matrix and the bary/baryweights routines had similar execution times when using the Chebyshev points, which was unsurprising due to them being conceptually similar. However, the Lagrange interpolation was faster for **n>321**.

Finally, the PCHIP routine was the fastest to execute and was very impressive. The execution time stayed almost constant as the number of points increased, highlighting the routine's efficiency.

4.2 Accuracy Comparison

Below are the results of plotting the infinity errors for the four Interpolation methods. Please note that I excluded the plots of Lagrange interpolation with evenly-spaced points for high **n** due to Runge's phenomenon.

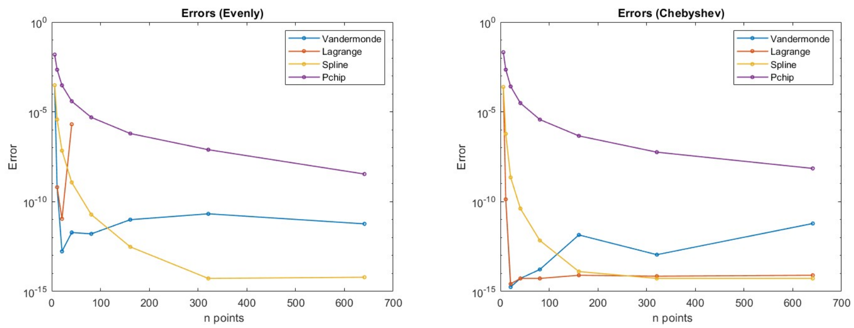


Figure 10: Errors of interpolation methods

To summarize, the PCHIP routine was the least accurate overall, followed by the interpolation using the Vandermonde matrix. The Lagrange and Spline interpolation ended up being the most accurate at high values of **n** when interpolating $y = e^x$ (when using the Chevyshev-spaced points).

5 Summary

This assignment involved the implementation and analysis of interpolating the function $y = e^x$ using the Vandermonde matrix, the Lagrange interpolation, and piecewise interpolation using the Spline and PCHIP routines. Below are the key takeaways:

- Interpolation using the Vandermonde matrix performed well when the function is smooth, and the dataset is small (below 20 points). However, there are faster, more accurate, and more versatile methods available.
- Lagrange Interpolation using bary/baryveights routines is accurate and very fast at any number of points. However, this method is very sensitive to Runge's phenomenon and requires an accurate selection of points. In real-world applications, if the dataset is limited, if the function is not smooth, or if there is no ability to accurately select starting points, Lagrange interpolation using the bary/baryweights routine may not be the best choice.
- The piecewise Interpolation using the spline routine allows trade between accuracy and speed. Using a higher-order spline increases the accuracy of the results but increases the execution time. In my opinion, spline interpolation is the most versatile overall.

- The piecewise Interpolation method using the PCHIP routine trades accuracy for the shortest execution time and has properties (such as bounding to the data points) that may make the method preferable when using real-life physical data. In our example, when interpolating $y = e^x$, PCHIP does not make much sense since it is a very smooth and predictable function.

In conclusion, I do not think there is one best method when choosing between the interpolation methods covered in this assignment. There are many factors that may make one algorithm excel over the other, such as the number of points used, the selection of points, and physical constraints.

However, when interpolating the function $y = e^x$ in this assignment, I think that Lagrange interpolation using Chebyshev points performed the best when both accuracy and speed were measured.

List of Figures