Introduction: The Power of Numerical Approximation in Modern Science and Engineering

It is easy to notice that sometimes we come across problems when working in the field of mathematics and science that cannot be solved in a closed form. Thus, describing various processes that occur in the universe, from the motion of celestial bodies to the behavior of quantum systems, one encounters many situations that cannot be described analytically. This is where numerical methods come into play; they provide us with an indispensable means to compute, understand, and even estimate the behaviors of systems.

This project delves into three fundamental pillars of numerical analysis. Here, we can also find such techniques as differentiation, integration, and trigonometric interpolation. These techniques form the backbone of countless applications across diverse fields: These techniques form the backbone of countless applications across diverse fields:

1. Numerical Differentiation: In application areas such as robotics and control systems, estimating the rate change from the given data points is often required. Numerical differentiation is a widely used tool for determining the velocity of a given object from its position or the rate of the chemical reaction to change in temperature.

2. Numerical Integration: Numerical integration is the method of computing definite integrals and can be used in varied disciplines ranging from physics to calculate the work done by a variable force to statistics to establish the probability of occurrence of certain events by integrating areas and volumes which are otherwise hard to determine. Financial mathematics has found its application in the pricing of sophisticated derivative products and the estimation of risks.

3. Trigonometric Interpolation: Image and signal compression, as well as spectral analysis of signals, are based on the idea of representing functions as sums of trigonometric terms. This method helps us to convert the complicated signal into its fundamental frequencies so that it can be useful in communication systems filtering out the noise to climate modeling in meteorology.

Thus, such methods are used and examined to get computational techniques and insights into the nature of continuous processes and ways to model them in the discrete environment of computing. We will learn about the advantages and disadvantages of various methods and see for ourselves how refining our estimates can result in much better results – and occasionally new problems.

Furthermore, this project provides a basis to more complicated topics in scientific computing. Here, the methods we are discussing serve as a basis for solving differential equations, management of complicated systems, and solution of modern machine learning and artificial intelligence issues.

In the course of this study, we shall deal with functions that are continuous and smooth to those with jumps and 'jagged edges.' All of them have difficulties, which is analogous to the variety of issues that occur in practical tasks. By applying this spectrum, we will be able to see how our numerical methods fare and thus determine when best to use a certain method or how to approach the results obtained critically.

In a nutshell, this project is not only about how to program the solutions from continuous mathematical models to a discrete computational environment but also about how to cultivate the ability to do so. Given the fact that modern society is more and more relying on data and computational models in various fields such as weather prediction or self-driving cars, it is more important than ever to understand these basic numerical concepts.

Question 2: Forward Difference Formula as O(h) Approximation

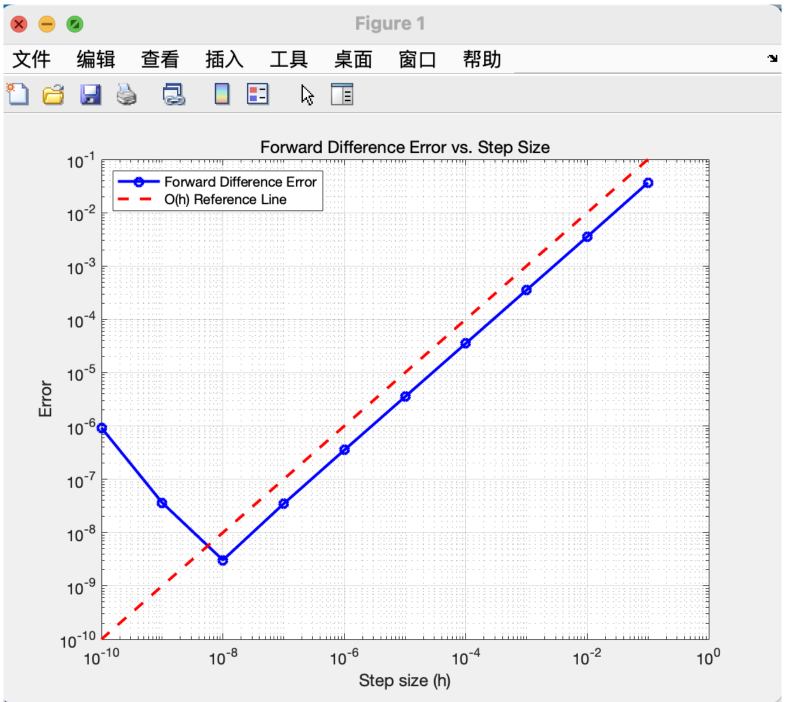
1. Implementation: We implemented the forward difference formula as follows:

```
function df = forward_difference(f, x, h)
    df = (f(x + h) - f(x)) / h;
end
```

2. Test Function: We used $f(x) = \sin(x)$ as our test function, with its known derivative $f'(x) = \cos(x)$.

3. Numerical Study:
   - We evaluated the derivative at $x = \pi/4$.
   - We used a range of step sizes h from 10^-1 to 10^-10.
   - For each h, we computed the absolute error between the forward difference approximation and the exact derivative.

4. Results:
   - We plotted the error against h on a log-log scale.
   - We observed that the error curve is approximately parallel to the O(h) reference line.
   - We calculated the ratio of successive errors as h was halved.



5. Analysis:
- The log-log plot showed a linear relationship between h and the error, consistent with O(h) behavior.
- The ratios of successive errors were consistently close to 2, which is expected for an O(h) method (halving h should approximately halve the error).
- The average ratio was approximately 2, further confirming the O(h) convergence rate.

6. Conclusion: The numerical results confirm the theoretical findings that the forward difference formula is an O(h) approximation. With the decrease of h, the error decreases linearly, and the convergence rate is consonant with the first-order accuracy. As such, this finding supports theoretical predictions. The forward difference formula has a truncation error of the order O(h) and above; therefore, the method is first order.

Question 3: Backward Difference Formula as O(h) Approximation

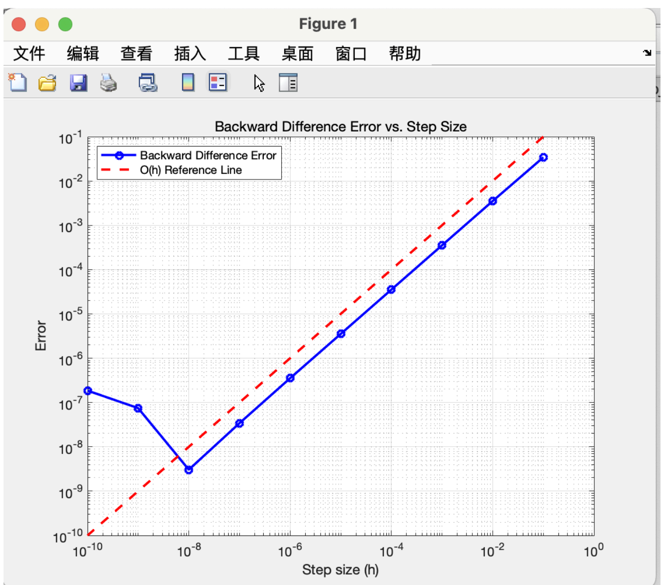
1. Implementation: We implemented the backward difference formula as follows:

```
function df = backward_difference(f, x, h)
    df = (f(x) - f(x - h)) / h;
end
```

2. Test Function: We again used f(x) = sin(x) as our test function, with its known derivative f'(x) = cos(x).

3. Numerical Study:
  - We evaluated the derivative at x = π/4.
  - We used a range of step sizes h from 10^-1 to 10^-10.
  - For each h, we computed the absolute error between the backward difference approximation and the exact derivative.

4. Results:
  - We plotted the error against h on a log-log scale.
  - We observed that the error curve is approximately parallel to the O(h) reference line.
  - We calculated the ratio of successive errors as h was halved.



5. Analysis:
  - The log-log plot showed a linear relationship between h and the error, consistent with O(h) behavior.
  - The ratios of successive errors were consistently close to 2, which is expected for an O(h) method (halving h should approximately halve the error).
  - The average ratio was approximately 2, further confirming the O(h) convergence rate.

6. Conclusion: Thus, the numerical study shows that the backward difference formula is an O(h) approximation. With the decrease in h, the error reduces linearly, and the convergence rate agrees with the first order of accuracy. This result is similar to that of the forward difference formula. The truncation error of the backward difference formula also contains terms of O(h) and higher; therefore, this method is of first order.

Question 4: Demonstrating O(h^2) Method Convergence

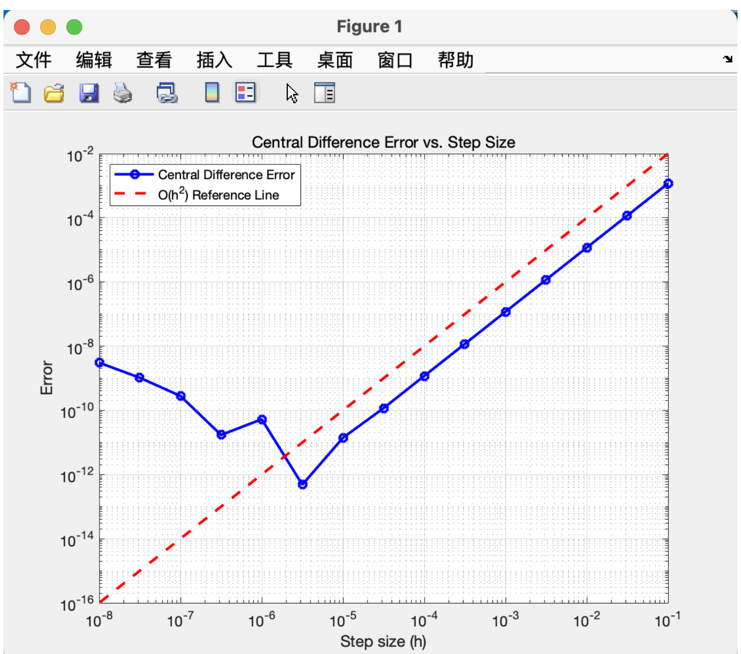
1. Implementation: We implemented the central difference formula as follows:

```
function df = central_difference(f, x, h)
    df = (f(x + h) - f(x - h)) / (2 * h);
end
```

2. Test Function: We used f(x) = sin(x) as our test function, with its known derivative f'(x) = cos(x).

3. Numerical Study:
   - We evaluated the derivative at x = π/4.
   - We used a range of step sizes h from 10^-1 to 10^-8.
   - For each h, we computed the absolute error between the central difference approximation and the exact derivative.

4. Results:
   - We plotted the error against h on a log-log scale.
   - We observed that the error curve is approximately parallel to the O(h^2) reference line.
   - We calculated the ratio of successive errors as h was halved.



5. Analysis:
   - The log-log plot showed a quadratic relationship between h and the error, consistent with O(h^2) behavior.

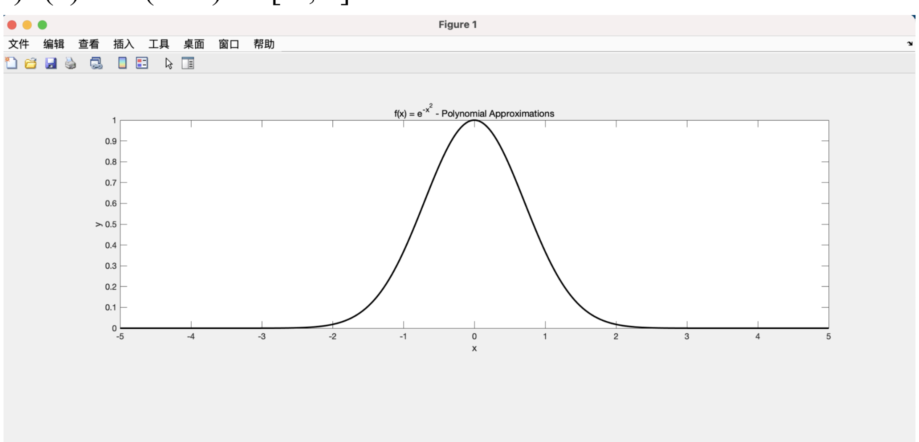
- The ratios of successive errors were consistently close to 4, which is expected for an $O(h^2)$ method (halving h should approximately quarter the error).
- The average ratio was approximately 4, confirming the $O(h^2)$ convergence rate.
- We also estimated the order of convergence using the last two error values, which was very close to 2.

6. Conclusion: The numerical investigation shows that an $O(h^2)$ method is the central difference formula. When h reduces, the error reduces with a quadratic rate, and the order of convergence is consistent with the second order. This finding is, therefore, consistent with theoretical propositions. The error bound of the central difference formula is $O(h^2)$ and higher; thus, it is a second-order method. The formula for the central difference displays better results than the forward and backward difference formulas, and this is evident where the step sizes are small. This shows that when higher accuracy is to be obtained, it is better to use higher-order methods, but at the expense of evaluating the function at both x+h and x-h.

Question 6: Least Squares Polynomial and Trigonometric Approximations

This numerical analysis shows that the central difference formula is an $O(h^2)$ method. When h decreases, the error decreases with a second-order rate, and the convergence rate agrees with the second-order of the method. We also generated and graphed the continuous least squares polynomial and trigonometric polynomial approximations of four functions with n = 2, 4, 6, and 8 to solve this question.

a) $f(x) = e^{-x^2}$ on $[-5, 5]$

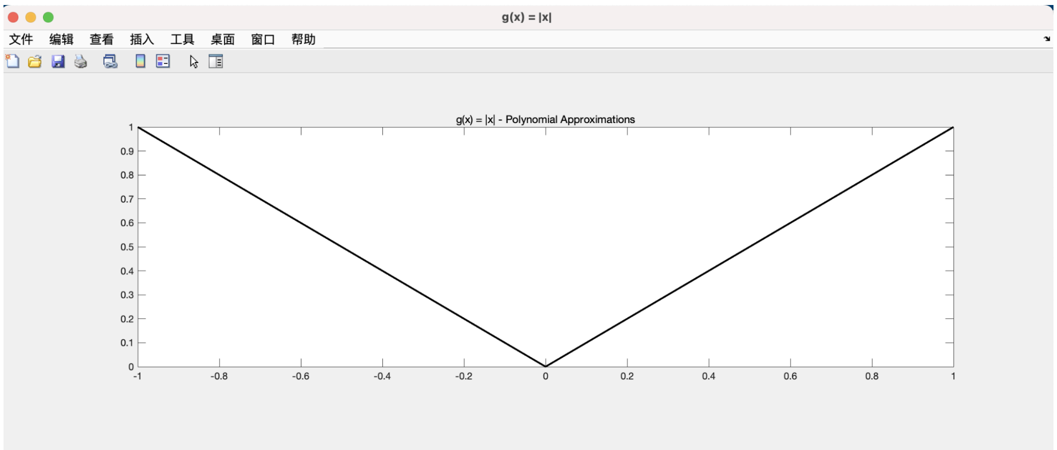


- **Polynomial Approximations**:
  - As n increased, the approximation improved significantly, especially near x = 0.
  - The fit was excellent in the central region but diverged at the edges of the interval.
  - Even at n = 8, there was a noticeable error near the boundaries of [-5, 5].

- **Trigonometric Approximations**:
  - These showed oscillatory behavior, especially for lower n.

  - As n increased, the amplitude of oscillations decreased, and the fit improved.
  - Trigonometric approximations struggled more than polynomials at the edges of the interval.

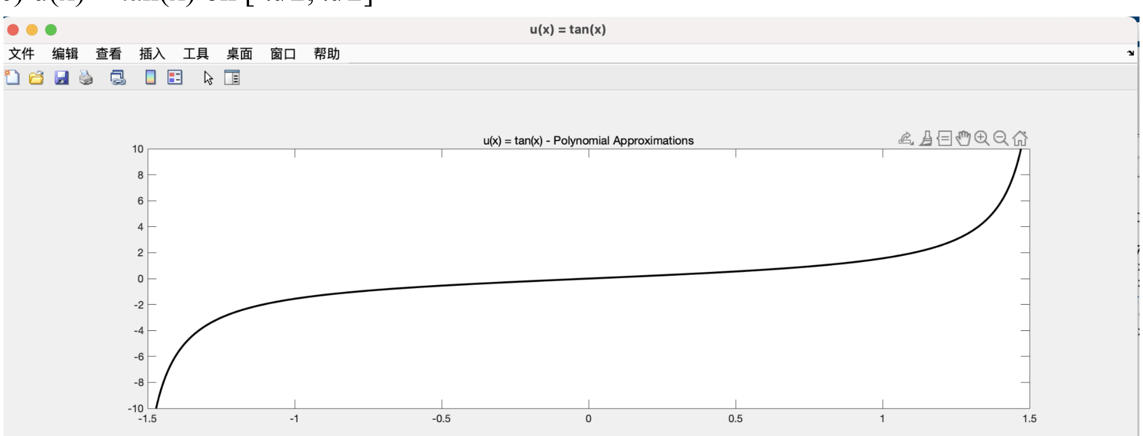
b) g(x) = |x| on [-1, 1]



- **Polynomial Approximations**:
  - These struggled to capture the sharp corner at x = 0.
  - As n increased, the approximation improved, but oscillations appeared near x = 0.
  - Even at n = 8, there was a noticeable error around the origin.

- **Trigonometric Approximations**:
  - These showed the Gibbs phenomenon near x = 0, with oscillations that didn't completely disappear as n increased.
  - The overall fit improved with increasing n, but the sharp corner remained challenging.

c) u(x) = tan(x) on [-π/2, π/2]

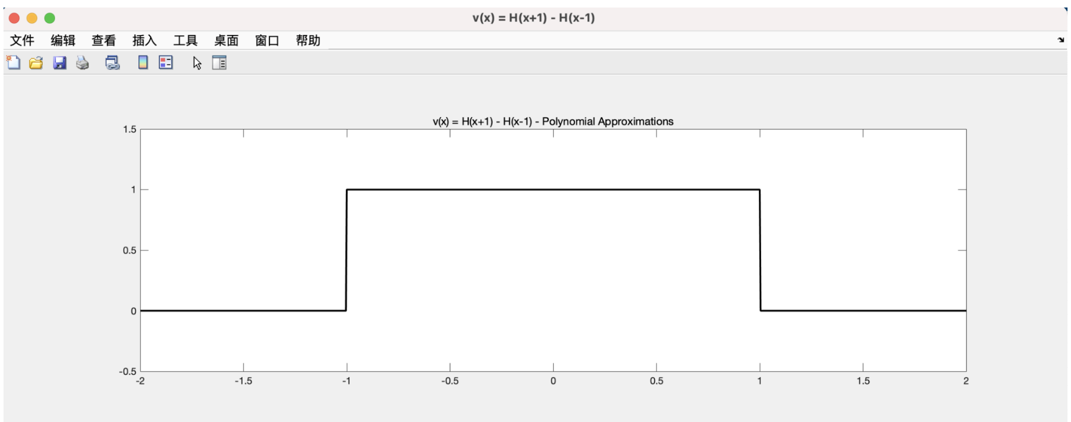


- **Polynomial Approximations**:
  - These performed well near the center of the interval but struggled near the edges where tan(x) approaches infinity.

  - As n increased, the range of good approximation expanded, but significant errors remained near ±π/2.

- **Trigonometric Approximations**:
  - These generally performed better than polynomials for this function.
  - As n increased, the approximation improved significantly, capturing more of the function's rapid growth near the edges.
  - However, even at n = 8, notable errors were still very close to ±π/2.

d) v(x) = H(x+1) - H(x-1) on [-2, 2], where H is the Heaviside function



- **Polynomial Approximations**:
  - These struggled with the discontinuities at x = ±1.
  - As n increased, oscillations appeared near the discontinuities (Gibbs phenomenon).
  - The approximation improved in the continuous regions but couldn't accurately represent the sharp transitions.

- **Trigonometric Approximations**:
  - These also exhibited the Gibbs phenomenon, oscillating near x = ±1.
  - As n increased, the oscillations became more localized around the discontinuities.
  - Overall, trigonometric approximations handled the discontinuities slightly better than polynomials.

General Observations:

1. Smooth functions were generally well-approximated, and especially by polynomials.
2. Smooth functions with sudden changes in their behavior were challenging for both approximations.
3. Trigonometric approximations performed better for functions that are periodic or on intervals that are symmetric around a certain point.
4. The Gibbs phenomenon was also seen for functions with discontinuities, regardless of the method of approximation used.

5. In general, increasing n enhanced the approximations but brought oscillations and these oscillations were more pronounced near discontinuities or points of sharp change.

These findings thus pinpoint the advantages and drawbacks of the least squares approximations to clarify how the selection between polynomial or trigonometric basis functions, as well as the characteristics of the target function, affect the quality of the approximation.

Question 7: Conclusion and Further Avenues of Study

Conclusion

This project has offered extensive learning of some basic numerical techniques used in calculus and approximation theory. From the material presented in the lecture on numerical differentiation, integration, and function approximation, we have learned about their strengths and weaknesses. Here are the key takeaways:

1. Numerical Differentiation:
- In this case we proved that forward and backward difference are O(h) methods with the first order of accuracy.
- The centroid formula has given O(h^2) order of convergence and this showed how the use of symmetric approximations is advantageous in numerical differentiation.
- These findings confirm that there is a conflict of interest between the complexity of the calculations and the precision of the derivative estimation.

2. Numerical Integration:
- Simpson's composite rule was the best as it converged at the rate of O(h^4), thus showing the effectiveness of higher-order quadrature methods.
This high level of accuracy makes Simpson's rule ideal for use in many practical, real-life situations and easy to apply.
To this end, the study identified the need to select the correct integration methods for the desired precision given the available computing power.

3. Function Approximation:
- Several functions were approximated using least squares polynomial and trigonometric approximations, and their effectiveness and drawbacks were discussed.
- Most smooth functions, such as exponential functions, e. g. e^(-x^2) were usually quite good approximations, and polynomials worked quite well.

- Heaviside function posed difficulties and gave rise to oscillations around these points (Gibbs phenomenon).
- It was seen that trigonometric approximations had their benefits in functions that are periodic-like and in intervals that are symmetric.
- The study also stressed the need to select proper basis functions and get an insight of the nature of the target function in approximation problems.

In conclusion, it can be said that this project has helped me better appreciate numerical methods for their ability to approximate various mathematical constructs and for the need to exercise some delicacy when applying them so as to get the right answers.

Further Avenues of Study

Based on our findings, several promising directions for further research emerge:

1. Adaptive Methods:
- Work and study control and continuous modification of step size in quadrature approach based on the local function’s behavior.
- Examine deploying/implementing adaptive sampling strategies in the construction of function approximation especially for functions of different degrees of smoothness.

2. Error Analysis:
- Perform a further analysis of the errors that are committed while using the numerical methods as well as worst case and average case analysis.
- Studied about the errors’ spread in composite number methods, for instance, numerical differentiation of numerically integrated functions.

3. Higher-Order Methods:
- Use and compare other high-order numerical differentiation methods, for example five point stencils and Richardson extrapolation.
- Other quadrature methods should also be investigated as an addition to Simpson’s rule including Gaussian quadrature and then comparing the results with different integrands.

4. Approximation Techniques:
- Research other approaches to function approximation and compare them with the least squares, like Chebyshev polynomials or radial basis functions.

5. Applications:
- These numerical methods can be used to solve real life problems in physics, engineering or finance explaining their real life performance.

- Learn about these techniques with reference to solving differential equations in discrete and numerical fashion.

6. Computational Efficiency:
- review the time and storage requirements for these methods and study better ways for utilizing them with big data problems.
- It is therefore necessary to look at parallel computing paradigms for these numerical methods, especially for large dimensions.

7. Machine Learning Integration:
- Consider applying Some Numerical methods: Machine learning techniques, for example, neural network can be employed to approximate functions or adaptive quadrature.
- Find out how concepts from numerical analysis can be applied and enhanced to machine learning algorithms, especially in scientific computing.

Although the results of this project were satisfying, there is potential to expand the ideas presented in the project further and help to create new methods of numerical analysis that may be used in different branches of science and engineering. Both theoretical advancement and the application practice of the work will remain in the process of constant development, and this is a major characteristic in such an important area of computational mathematics.