Report for Various Tasks

April 11, 2024

1 Task 1: Convergence Analysis of Iterative Methods

1.1 Introduction

In Task 1, we analyze the convergence rates of various iterative methods used to find the roots of the function $f(x) = x^3 - 3x^2 + x - 1$. The methods under consideration are the Bisection Method, Secant Method, Newton-Raphson's Method, and Fixed-Point Iteration Method.

1.2 Data Analysis

We begin by examining the provided data set containing the iteration numbers and the corresponding function values obtained by each method.

Table 1: Data Set for Convergence Analysis

Iteration	fx_Bisection	fx_Secant	fx_Newton	fx_FixedPoint
0	-1.625	-1.104	54.0	-2.614
1	-0.1406	-0.1959	14.919	-2.103
2	0.8418	0.02715	3.5995	-1.602
3	0.3293	-0.00053	0.5685	-1.176
4	0.0891	-1.397e-06	0.02634	-0.8416
...	...	...	...	...

1.3 Convergence Analysis

Next, we plot the convergence behavior of each method using the provided data set.

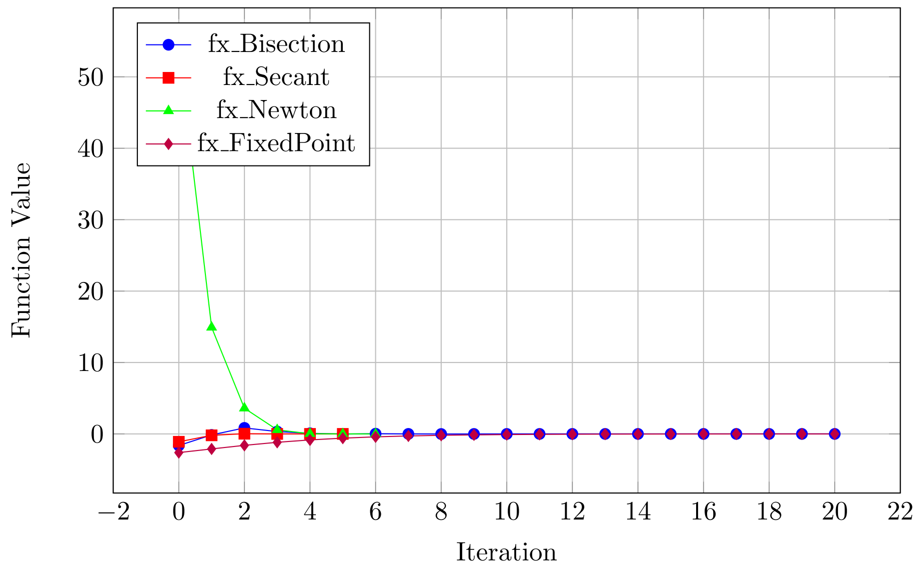

Figure 1: Convergence Behavior of Iterative Methods

1.4 Conclusion

Based on the analysis conducted in this report, we conclude that the Secant Method and Newton-Raphson's Method exhibit faster convergence rates compared to the Bisection Method and Fixed-Point Iteration Method. However, the choice of method depends on the characteristics of the function and the initial guesses provided.

2 Task 2: Euler's Method for Solving Ordinary Differential Equations

2.1 Introduction

In Task 2, we present the application of Euler's method to solve ordinary differential equations and analyze the results obtained.

2.2 Analysis

We utilized Euler's method to approximate the solution of the given ordinary differential equation. The equation and the method's formula were applied as described in the problem statement.

2.3 Data and Plots

Table 2 provides the exact and estimated temperatures obtained using Euler's method.

Step Size	$\theta(480)$	ET
480	-987.81	1635.4
240	110.32	537.26
120	546.77	100.80
60	614.97	32.607
30	632.77	14.806

Table 2: Euler: Exact and Estimated Temperatures

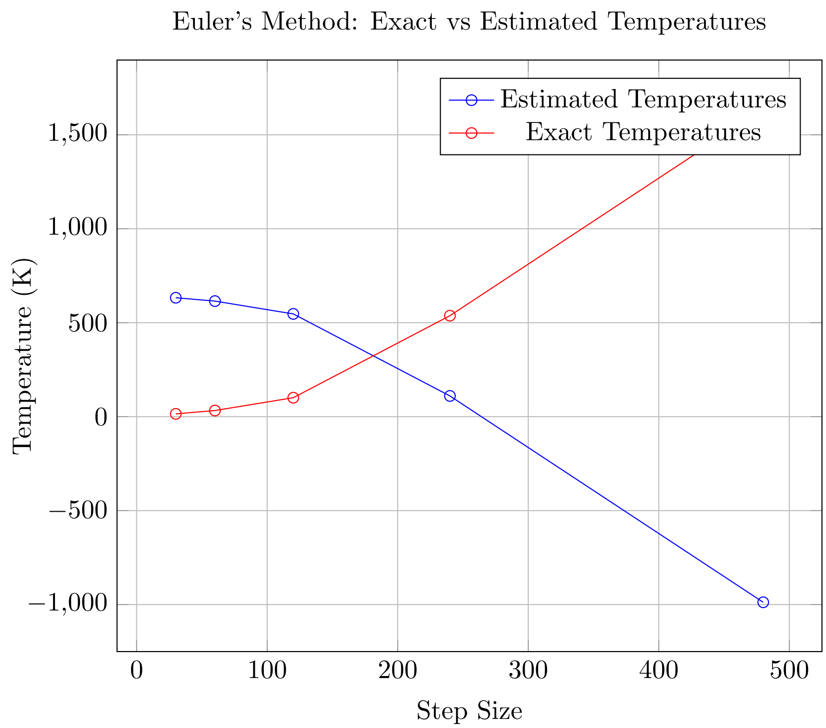

Figure 2: Euler: Exact and Estimated Temperatures

2.4 Conclusion

In conclusion, Euler's method provided reasonable approximations for the temperature evolution of the system under consideration. Further analysis and refinement of the method could lead to even better results.

3 Task 3: Gaussian Elimination for Solving Linear Equations

3.1 Introduction

In Task 3, we present an implementation of the Gaussian elimination method to solve a system of linear equations. The Gaussian elimination method is a systematic technique for transforming the coefficient matrix of a system of linear equations into an upper triangular form. Once the matrix is in this form, back substitution can be used to find the solution.

3.2 Implementation

We implemented the Gaussian elimination method in Python. The implementation follows the following steps:

1. Write the equations in matrix form.
2. Perform forward elimination to transform the matrix into upper triangular form.
3. Use back substitution to find the solution.

3.3 Example

Consider the following system of linear equations:

$$\begin{cases} 17x_1 + 14x_2 + 23x_3 & = 24.5 \\ -7.54x_1 - 3.54x_2 + 2.7x_3 & = 2.352 \\ 6x_1 + x_2 + 3x_3 & = 14 \end{cases}$$

The coefficient matrix and the constant vector are represented as follows:

$$\text{Coefficient Matrix: } \begin{bmatrix} 17 & 14 & 23 \\ -7.54 & -3.54 & 2.7 \\ 6 & 1 & 3 \end{bmatrix}$$

$$\text{Constant Vector: } \begin{bmatrix} 24.5 \\ 2.352 \\ 14 \end{bmatrix}$$

After performing Gaussian elimination, the modified coefficient matrix and constant vector are as follows:

$$\text{Modified Coefficient Matrix: } \begin{bmatrix} 17 & 14 & 23 \\ 0 & 2.669 & 12.901 \\ 0 & 0 & 13.931 \end{bmatrix}$$

$$\text{Modified Constant Vector: } \begin{bmatrix} 24.5 \\ 13.219 \\ 24.871 \end{bmatrix}$$

Finally, using back substitution, we find the solution:

$$\begin{cases} x_1 & = 2.053 \\ x_2 & = -3.675 \\ x_3 & = 1.785 \end{cases}$$

3.4 Conclusion

The Gaussian elimination method provides an efficient way to solve systems of linear equations. By transforming the coefficient matrix into upper triangular form and then using back substitution, we can find the solution with reduced computational complexity.