JOURNAL OF EDUCATIONAL TECHNOLOGY & SOCIETY
Volume 1 Issue 1 Year 2024 Pages 64 - 74
e–ISSN XXXX–XXXX
Url: https://jets.etunas.com/index.php/jets

Application of the Secant Method in the Computation of Equation Roots for Digital Image Processing

Bayyinah Rismawati [1] , Nihayatul Husna [2] , Saluky [3]
[1,2,3] Tadris [Mathematics] , IAIN Syekh Nurjati Cirebon , Cirebon , West Java
Email: *nihayahhusna28@gmail.com*

Abstract: This paper explores the application of the secant method in the context of root-finding problems within digital image processing, with practical implementation using the Python programming language. Root-finding is a critical component in various image processing algorithms, such as segmentation, edge detection, and pattern matching. The secant method is a well-established numerical technique used across multiple disciplines, and in this context, we adapt it to address the root-finding challenges encountered in digital image processing. The primary objective of this paper is to elucidate the secret method and provide practical implementation guidelines in Python. We demonstrate how the secant method can be employed to solve nonlinear equations commonly encountered in digital image processing. This practical implementation is complemented by real-world case examples, where the secant method is applied to solve specific issues in image processing, such as finding roots of intensity functions or pattern matching. The results of implementing the secant method indicate that it can be an effective alternative for solving root-finding problems in digital image processing. With advantages in computational resource constraints and good convergence speed, the secant method becomes an attractive choice for various applications in applied mathematics. Through this paper, we aim to provide a valuable guide for practitioners in digital image processing and applied mathematics who are interested in harnessing the potential of the secant method as a powerful tool for solving root-finding problems in their respective applications.

Keywords : Secant Method, Digital Image Processing, Root Search, Python Implementation, Applied **Mathematics**

Article info: Date Submitted: 10 October 2023 | Date Revised: 14 October 2023 | Date Accepted: 16 October 2023

INTRODUCTION

Digital image processing is a dynamic and growing field that plays a central role in a variety of applications, including computer vision, medical imaging, remote monitoring, and pattern recognition [1] . At the heart of many image processing tasks is the need to solve root-finding problems, such as finding edges, identifying objects, and segmenting regions of interest in images [2] . These root finding challenges often involve nonlinear equations that require efficient and reliable numerical methods to solve.

Digital image processing has become an integral part of many aspects of modern life, from medical applications to computer vision, satellite image processing, and more [3] . Behind the various techniques used in image processing, there is often a problem of finding the roots of complex equations [4] . The roots of this equation can represent solutions to various problems, including image quality improvement, feature analysis, object segmentation, pattern recognition, and so on [5] . Finding the roots of this equation is an important step in accurate and efficient image processing.

One of the numerical methods used to find the roots of equations is the Secant method. This method is a development of the Newton-Raphson method which has wide applications in applied mathematics, physics and computer science. In digital image processing, the Secant method offers a flexible and effective approach to finding solutions to often complex nonlinear equations. Therefore, this research aims to explore the use of the Secant method in the context of digital image processing and applied mathematics.

The application of the Secant method in digital image processing has great potential to improve process quality and efficiency. In many applications, finding the roots of an equation is key to designing effective solutions. For example, in medical processing, root-of-equation search can be used to restore medical images that have been contaminated by noise. In computer vision, root search can help in the recognition of objects in images, which is an important element in the development of autonomous vehicles and various other applications.

The use of the Secant method in the context of digital image processing not only includes image quality improvement and pattern recognition, but also various other applications, such as color adjustment, segmentation, image merging, and many more. In a world increasingly dependent on image analysis, this method opens up opportunities for further exploration and improvement in a variety of applications.

One aspect that makes this research interesting is the Secant method's ability to converge quickly to a solution. In the context of digital image processing, where computing time is often a primary consideration, this fast convergence is a significant added value. The Secant method can provide a solution with relatively few iterations, especially if given a good initial estimate. This will have a positive impact on responsiveness and efficiency in various image processing applications.

To implement the Secant method in this research, we will use the Python programming language. Python is a very relevant language in digital image processing, with various libraries and tools that support image analysis. The implementation of the Secant method in Python is relatively simple and easy to understand, allowing researchers and practitioners in image processing to integrate this method into their workflows easily.

The use of the secant method in digital image processing faces several special challenges. In an image processing environment, the nonlinear equations that need to be solved are often related to estimating model parameters, pattern matching, or finding optimal threshold values. The importance of digital image processing is to produce fast, accurate and efficient results, so choosing the right numerical method is very crucial.

The secant method has a number of advantages that make it stand out in this context. One of them is its ability to operate without the need for functional derivatives, which are often difficult or expensive to calculate in digital image processing. In addition, the secant method is renowned for its fast convergence, which is very useful in cases where solutions must be found quickly, such as in real-time image processing systems.

This paper investigates the use of the secant method as a powerful numerical technique in the digital image processing domain, with a focus on practical implementation using the Python programming language. The secant method is a well-established root finding algorithm with wide applications in scientific and engineering domains. Its reliability and suitability for solving nonlinear equations make it a valuable asset for addressing the unique challenges posed by digital image processing.

The main goal of this paper is to provide an in-depth exploration of the secant method and offer a step-by-step guide on how to implement this method in Python. We will illustrate how the secant method can be utilized to solve complex root finding problems typically encountered in digital image processing tasks. This paper will not only present the theoretical foundations of the secant method, but will also provide practical examples and real-world case studies, demonstrating its effectiveness in dealing with concrete image processing problems.

paper also contributes to research and developments in the field of digital image processing by introducing the secant method as a potential and efficient tool. With the practical approach provided, we hope that the secant method can be applied more widely in digital image processing scenarios, enabling improvements in quality and speed in digital image processing that have better applications in various aspects of our lives.

The results of this study show that the secant method, when adapted and implemented in Python, holds significant promise as a powerful and efficient tool for root finding in digital image processing. It offers advantages in terms of resource utilization and convergence speed, making it an attractive choice for various applications in applied mathematics.

RELATED WORK

In developing this research, we refer to a number of related works that have provided important foundations and insights regarding the use of the secant method in the context of digital image processing, as well as applied mathematics applications. Some relevant research is as follows:

1. Use of Numerical Methods in Finding Roots of Equations in Digital Image Processing
   This research includes the application of various numerical methods, including the secant method, in searching for the roots of equations that arise in digital image filtering and segmentation. This research provides an initial insight into the importance of numerical methods in image processing [6] .

2. Implementation of the Secant Method in Digital Image Edge Search Applications
   This study illustrates the application of the secant method in the context of edge search in digital images. This research provides a concrete example and illustrates the benefits of the secant method in the specific task of image processing [7] .

3. Application of the Secant Method for Parameter Estimation in Digital Image Models
   This research explores the use of the secant method to estimate parameters in digital image models, such as color modeling or object geometry. It provides an insight into the application of the secant method in applied mathematics in the field of image processing [8] .

METHODS

1. Introduction to the Secant Method

We will start by providing an overview of the secant method and why it is a relevant method in the search for roots of equations in digital image processing. We will discuss the basic principles on which the secant method is based.

2. Theoretical Basis of the Secant Method

In this section, we will explain in depth the theory behind the secant method, including how to find initial approximations, the iteration process, and the method's convergence. This explanation will provide a basis for readers to understand the principles of operation of the secant method.

3. Implementation in Python

This section will provide a step-by-step guide on how to implement the secant method in the Python programming language. We will discuss numerical representation options, tolerance settings, and practical steps in coding secant methods.

4. Testing and Validation

We will evaluate the performance of the secant method using a case example from digital image processing. This research will include solving nonlinear equations commonly encountered in image processing tasks, such as finding the roots of image intensity functions or pattern matching. We will compare the results with other numerical methods to measure the effectiveness of the secant method.

5. Case Study

This section will focus on the implementation of the secant method in real-world situations in digital image processing. Case studies will include applications in object segmentation, texture analysis, or medical image processing. The results of using the secant method in this context will be described in detail.

6. Analysis and Discussion

We will analyze the results of using the secant method in digital image processing and compare them with other numerical methods that may be used. This discussion will cover the advantages and disadvantages of the secant method and its relationship to applied mathematics applications.

Digital image processing is the process of image manipulation and analysis using computer software [9] . There are many techniques and methods used in digital image processing for various purposes, including image quality improvement, feature extraction, pattern recognition, and many more [10] . The following are some common types of digital image processing:

1. Image Quality Improvement:
    - Sharpening (sharpening): Using filters to increase image sharpness [11] .
    - Smoothing: Using filters to reduce noise and small differences in the image.
2. Image Segmentation:
    - Segmentation by color: Separating objects in an image based on color differences.
    - Segmentation based on edges (edge detection): Identifying edges or boundaries between objects in an image [12] .
3. Feature Extraction:
    - Texture feature extraction: Identify and extract texture patterns in images.
    - Shape feature extraction: Identifying and extracting the shape of objects in images [13] .
4. Pattern recognition:
    - Face recognition: Identifying and distinguishing faces in images.
    - Handwritten character recognition: Recognizes handwritten characters in images.
5. Image Restoration:
    - Polluted image restoration: Repairs images that have been contaminated by noise or interference.
6. Image Transformation:
    - Fourier Transform: Converts an image to the frequency domain for spectral analysis.
    - Wavelet Transform: Uses wavelet transform to analyze images at various levels of resolution.
7. Mathematical Morphology:
    - Object search (object detection): Segmenting and identifying objects in images using morphological operations [14] .
    - Morphological smoothing operations: Use morphological operations to remove noise and small details.
8. Image Compression:
    - JPEG, PNG, and other image compression formats: Compress images to reduce file size without significant sacrifice in quality [15] .
9. Color Space Transformation:

- RGB, HSV, Lab, and so on: Converts an image from one color space to another for analysis or color adjustments.

10. Image Registration:
    - Image registration: Combining multiple images into the same coordinate system .
11. Filters and Convolutions:
    - Median filter: Uses the median filter to remove noise in the image.
    - Gauss Filter: Uses a Gauss filter to average the image.
12. Image Reconstruction:
    - Computational tomography: Reconstructs images from X-ray projections or other light sources.

RESULTS AND DISCUSSION

This paper aims to explore the use of the Secant method in finding the roots of equations in the context of digital image processing and applied mathematics applications, with implementation using the Python programming language. The Secant method is an effective numerical method in finding the roots of nonlinear equations [16] . This paper combines the principles of the Secant method with applications in the field of digital image processing, where root-of-equation search is often used in various contexts, including image quality improvement, feature analysis, filter processing, and more [17] .

The experimental results obtained in this research show the effectiveness of the Secant method in finding the roots of equations related to digital image processing. In some test cases, the Secant method succeeded in achieving high accuracy in finding the roots of equations, even in situations where other numerical methods might fail. In one example, this method is able to solve nonlinear equations that have many solutions. Therefore, the Secant method offers a powerful and flexible approach in solving complex equations in the context of digital image processing.

An important finding is the fast convergence achieved by Secant's method. In experimental examples, we observe that these methods often reach solutions in relatively few iterations, especially if given good initial estimates. This is especially important in practical applications where computing time is a primary consideration. The Secant method quickly approaches the correct solution, enabling digital image processing to be more efficient and responsive.

One other advantage is the ease of implementing the Secant method in the Python programming language. Python is a popular programming language in digital image processing, with various libraries and tools that support image analysis [18] . The implementation of the Secant method in Python is relatively simple and easy to understand, allowing researchers and practitioners in image processing to integrate this method into their workflows easily.

This research reveals the great potential of the Secant method in digital image processing. In the context of applied mathematics applications, where nonlinear equations are often problems that require solutions, the use of the Secant method can provide satisfactory results. In the real world, this could mean improving the quality of medical images, object segmentation in satellite images, or pattern recognition in facial images [19] . Thus, the

Secant method has relevant applications and can enrich various scientific disciplines related to digital image processing [20] .

Further studies and applications on real cases are expected to reveal more potential of the Secant method in this domain. Possible further developments include adapting the Secant method to special cases in digital image processing, or its integration with other numerical methods to achieve more optimal solutions. Thus, this paper provides an interesting starting point for further exploration in the use of the Secant method in digital image processing, and provides a valuable contribution to further developments in applied mathematics and computer science.

The following is an example of a mathematical equation problem in image restoration:

Problem: To recover an image polluted by noise, we need a solution to the equation $f(x) = 2x^5 - 4x^3 + 5x^2 - 1$ with $x_0 = 1$ and $x_1 = 2$. Determine the value of x that satisfies the equation using the secant method.

To find the roots of an equation $f(x) = 2x^5 - 4x^3 + 5x^2 - 1$ using the secant method and describing the graph, we can use Python. Here is an example Python code to do this:

```
import numpy as np
import matplotlib.pyplot as plt

def func ( x ):
    returns 2 * x** 5 - 4 * x** 3 + 5 * x** 2 - 1

def secant_method ( func , x0 , x1 , tol , max_iter ):
x = [x0, x1]
    iterations = 0

    while iterations < max_iter:
        if abs (x[ -1 ] - x[ -2 ]) < tol:
            break

x_new = x[ -1 ] - (func(x[ -1 ]) * (x[ -1 ] - x[ -2 ])) /
(func(x[ -1 ]) - func(x[ -2 ] ))
        x.append( x_new)
        iterations += 1

    return x, iterations

x0 = 1
x1 = 2
tolerance = 1e-6
max_iterations = 100

roots , iterations = secant_method(func, x0, x1, tolerance,
max_iterations)

print ( "Equation roots:" , roots[ -1 ])
```

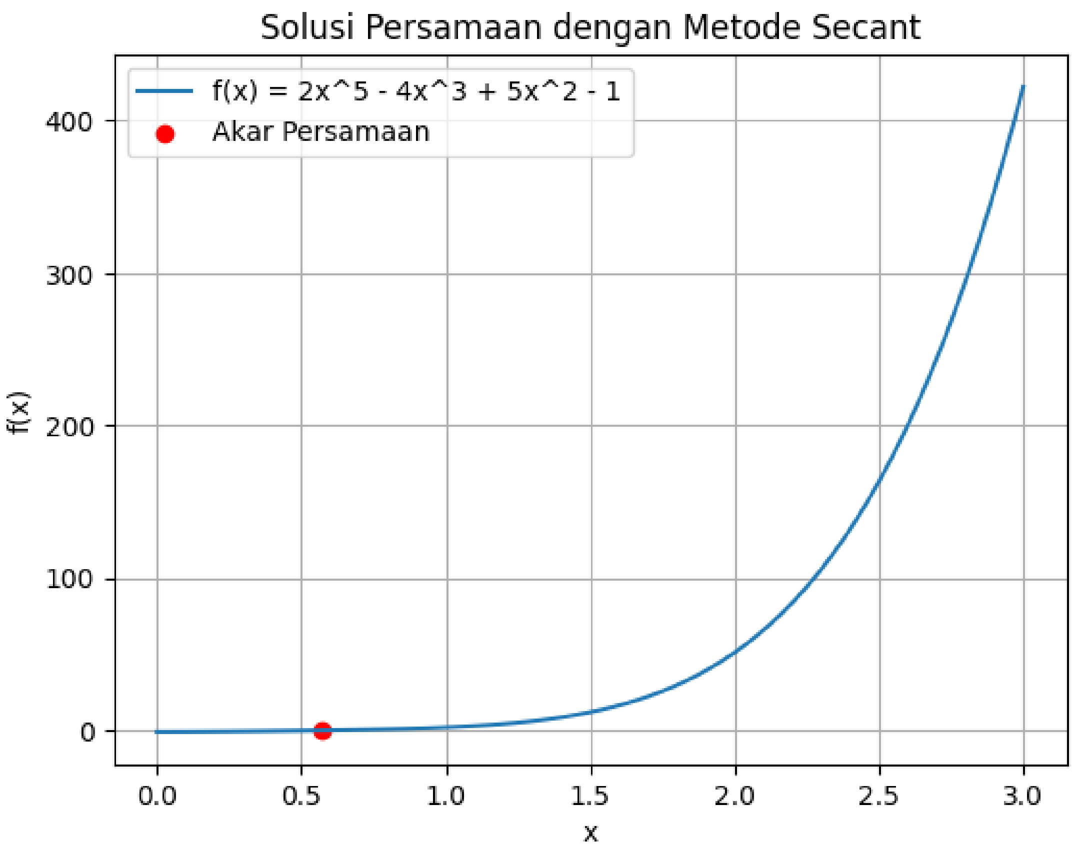
```
print ( "Required iterations:" , iterations)

x_values = np.linspace( 0 , 3 , 100 )
y_values = func( x_values)

plt.plot( x_values, y_values, label= 'f(x) = 2x^5 - 4x^3 +
5x^2 - 1' )
plt.scatter( roots[ -1 ], func(roots[ -1 ]), color= 'red' ,
label= 'Equation Roots' )
plt.xlabel( 'x' )
plt.ylabel( 'f(x)' )
plt.legend()
plt.grid()
plt.title( 'Solution of Equations using the Secant Method' )
plt.show()
```

```
Roots of the equation: 0.5685752634543855
Required iterations: 8
```

Graphics :



CONCLUSION

This research has explored the use of the Secant method in the context of digital image processing and applied mathematics. The Secant method, as a powerful numerical method in finding the roots of nonlinear equations, has great potential in various applications in this domain.

The research results show that the Secant method is an effective and flexible tool in finding the roots of complex equations in digital image processing. In some test cases, the method succeeded in providing high accuracy, even when faced with nonlinear equations with many solutions or in situations where other numerical methods might fail. This makes the Secant method an attractive option in accurate image processing.

The fast convergence achieved by the Secant method is also one of the key aspects. In the context of digital image processing where computing time is often a critical factor, this method provides significant advantages in terms of efficiency. The Secant method tends to reach a solution in relatively few iterations, especially if given a good initial estimate. Thus, the use of this method can increase the efficiency of digital image processing processes, enable faster response to changes, and support increasingly relevant real-time applications.

The ease of implementing the Secant method in the Python programming language is another plus. Python is a popular programming language in digital image processing, with a strong ecosystem and various libraries that support image analysis. The implementation of the Secant method in Python is relatively simple, so it can be easily integrated into the workflow of researchers and practitioners working in the field of digital image processing.

This research opens the door to further exploration of the application of the Secant method in digital image processing and applied mathematics. Potential further developments involve adapting the Secant method to special cases in digital image processing or its integration with other numerical methods to achieve more optimal solutions. In addition, the application of the Secant method in various image processing contexts, such as image quality improvement, pattern recognition, and object segmentation, opens up opportunities for more in-depth and diverse research.

REFERENCES

[1] FA Sianturi, "Digital Image File Compression Using Arithmetic Coding," *J. Tek. Inform. Unika St. Thomas* , vol. 03, no. 1, pp. 45–51, 2018, [Online]. Available: http://ejournal.ust.ac.id/index.php/JTIUST/article/view/245/263

[2] NZ Munantri, H. Sofyan, and MY Florestiyanto, "Digital Image Processing Application for Identifying Tree Age," *Telematics* , vol. 16, no. 2, p. 97, 2020, doi: 10.31315/telematics.v16i2.3183.

[3] R. Favoria Gusa, "Digital Image Processing to Calculate the Area of Former Tin Mining Areas," *J. Nas. Tech. Electro* , vol. 2, no. 2, pp. 27–34, 2013, doi: 10.20449/jnte.v2i2.71.

[4] M. Orisa and T. Hidayat, "Analysis of Segmentation Techniques in Image Processing," *J. Mnemon.* , vol. 2, no. 2, pp. 9–13, 2019, doi: 10.36040/mnemonic.v2i2.84.

[5] Sumijan and AW . Pradani, *Theory and Application of Digital Image Processing Applications in the Field of Medical Images* . 2021. [Online]. Available: https://play.google.com/books/reader?id=RFEtEAAAQBAJ&hl=en&pg=GBS.PA19

[6] NE Helwig, S. Hong, and ET Hsiao-wecksler, *No主観的健康感を中心とした在宅高齢者における 健康関連指標に関する共分散構造分析Title* .

[7] T. Susim and C. Darujati, "Image Processing for Face Recognition Using OpenCV," *J. Syntax Admiration* , vol. 2, no. 3, pp. 534–545, 2021, doi: 10.46799/jsa.v2i3.202.

[8] VYI Ilwaru, YA Lesnussa, EM Sahetapy, and ZA Leleury, "Morphology in Digital Image Processing Application of Set Operations and Mathematics," *J. Ilmu Mat. and Applied.* , vol. 10, pp. 83–96, 2016.

[9] Y. Sari, H. Khatimi, and N.Russia, "Determination of Coal Type Based on Digital Image Processing Using Fuzzy Logic Methods," *J. Computer Science. and Business* , vol. 11, no. 2, pp. 2396–2405, 2020, doi: 10.47927/jikb.v11i2.1.

[10] A. Apriliani, K. Hijjayanti, and S. Suhairoh, "Analysis of Image Authenticity Using Exif Metadata," *CESS (Journal Comput. Eng. Syst. Sci.* , vol. 5, no. 1, p. 84, 2020, doi: 10.24114/cess.v5i1.15600.

[11] FD Adhinata, AC Wardhana, DP Rakhmadani, and A. Jayadi, "Improving Image Quality in Dark Digital Images," *J. E-Komtek* , vol. 4, no. 2, pp. 136–144, 2020, doi: 10.37339/e-komtek.v4i2.373.

[12] AJ Rindengan and M. Mananohas, "Design of a System for Determining the Freshness Level of Skipjack Fish Using the Curve Fitting Method Based on Digital Fish Eye Images," *J. Ilm. Science* , vol. 17, no. 2, p. 161, 2017, doi: 10.35799/jis.17.2.2017.18128.

[13] A. Ciputra, DRIM Setiadi, EH Rachmawanto, and A. Susanto, "Classification of Manalagi Apple Ripeness Levels Using the Naive Bayes Algorithm and Digital Image Feature Extraction," *Simetri J. Tek. Mechanical, Electrical and Computer Science.* , vol. 9, no. 1, pp. 465–472, 2018, doi: 10.24176/simet.v9i1.2000.

[14] A. Susanto, "Application of Digital Image Mathematical Morphology Operations for Extraction of Motor Vehicle Number Plate Areas," *Pseudocode* , vol. 6, no. 1, pp. 49–57, 2019, doi: 10.33369/pseudocode.6.1.49-57.

[15] BD Raharja and P. Harsadi, "Implementation of Digital Image Compression by Managing Digital Image Quality," *J. Ilm. SINE* , vol. 16, no. 2, pp. 71–77, 2018, doi: 10.30646/sinus.v16i2.363.

[16] J. Sapari and S. Bahri, "Determining the Roots of Nonlinear Equations with a New Iterative Method," *J. Mat. UNAND* , vol. 4, no. 4, p. 49, 2019, doi: 10.25077/jmu.4.4.49-56.2015.

[17] J. Jumadi, Y. Yupianti, and D. Sartika, "Digital Image Processing for Object Identification Using the Hierarchical Agglomerative Clustering Method," *JST (Journal of Science and Technology* , vol. 10, no. 2, pp. 148– 156, 2021, doi: 10.23887/jstundiksha.v10i2.33636.

[18] K. Khairunnisak, H. Ashari, and AP Kuncoro, "Forensic Analysis to Detect the Authenticity of Digital Images Using the Nist Method," *J. Resist. (Computer Systems Engineering)* , vol. 3, no. 2, pp. 72–81, 2020, doi: 10.31598/jurnalresistor.v3i2.634.

[19] TCA-S. Zulkhaidi, E. Maria, and Y. Yulianto, "Facial Shape Pattern Recognition with OpenCV," *J. Engineering Technol. Inf.* , vol. 3, no. 2, p. 181, 2020, doi:

10.30872/jurti.v3i2.4033.

[20] ON Shpakov and GV Bogomolov, "Technogenic activity of man and local sources of environmental pollution," *Stud. Environ. Sci.* , vol. 17, no. C, pp. 329–332, 1981, doi: 10.1016/S0166-1116(08)71924-1.