

Technical Note

Not peer-reviewed version

Dynamic Ratio Golden Section Method for Single Variable Optimization Studies

[Wenfa Ng](#) *

Posted Date: 29 December 2025

doi: 10.20944/preprints202512.2537.v1

Keywords: dynamic ratio Golden section; fixed ratio Golden section; convergence; x2 internal point; rapidly changing function



Preprints.org is a free multidisciplinary platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This open access article is published under a [Creative Commons CC BY 4.0 license](#), which permit the free download, distribution, and reuse, provided that the author and preprint are cited in any reuse.

Disclaimer/Publisher's Note: The statements, opinions, and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions, or products referred to in the content.

Technical Note

Dynamic Ratio Golden Section Method for Single Variable Optimization Studies

Wenfa Ng

Department of Biomedical Engineering, National University of Singapore; ngwenfa771@hotmail.com

Abstract

Golden section is a proven, robust and high utility single variable optimization method of use in many engineering disciplines. In its typical implementation, a fixed ratio (also known as Golden ratio) is used to partition the left-right bracket in such a way that the two internal points are strategically placed to close-in on the minimum point. Practical usage of this fixed ratio method shows good performance in progressive movement of x_2 internal point to the minimum value. However, the fixed ratio method could not capture or follow the curve of rapidly changing function well. Herein, an improved Golden section method with dynamic ratio for partitioning the bracket is formulated and tested on a demonstrative case to optimize a quadratic function. Due to lack of computational power on a GPU budget laptop, more tests could not be conducted on this dynamic ratio Golden section method. Initial results obtained showed that the dynamic ratio does place the x_2 internal point closer to the right bracket in regions of rapidly changing value. Overall, the dynamic ratio Golden section method reached convergence to the minimum point in 3 iterations compared to 12 iterations for the conventional fixed ratio Golden section method in a demonstrative case.

Keywords: dynamic ratio Golden section; fixed ratio Golden section; convergence; x_2 internal point; rapidly changing function

Subject areas: mathematics, applied mathematics, biomedical engineering, chemical engineering

Introduction

Golden section method is an important single variable optimization method for finding the minimum or maximum of an objective function [1]. Briefly, the method relies on an initial guess for the left and right bracket of the minimum value, and progressively compares the function evaluation at two internal points to decide on which bracket to be omitted in the next iteration. Usually, the Golden ratio is used to partition the interval between the left and right bracket into the two internal points. Doing it this way allows the Golden section method to move towards the minimum point with successive iterations.

However, the minus point for using a fixed ratio Golden section method is that it may not handle rapidly changing part of the objective function well, with the attendant negative effect of missing a global minimum point. Hence, this work proposes the use of a dynamic ratio Golden section method where the partition ratio is determined by the relative ratio of the function evaluations at the left and right internal point, x_1 and x_2 . Ideally, for large relative ratio indicative of a rapidly changing function, the partition ratio, r , will shift the x_2 internal point closer to the right bracket in order to better capture the rapidly changing nature of the objective function. Noting here that the relative ratio will be recalculated at each iteration, and is likely to change with each iteration, thereby, earning the method the name of a dynamic ratio Golden section.

Algorithmic Logic of the New Method

In contrast to the standard implementation for Golden section, my fixed ratio Golden section implementation first defines as first guess for the left and right bracket (point A and B). Using the golden ratio, these two points help calculate two internal points (x_1 and x_2). Function evaluations are performed for these two internal points, and the results are compared. In the case where $f(x_1)$ value is lower than $f(x_2)$ value, the right bracket is moved to x_2 . Conversely, if $f(x_1)$ value is higher than $f(x_2)$ value, then the left bracket is moved to x_1 . And the single variable optimization proceeds until the difference between x_1 and x_2 is smaller than the set tolerance of the optimization exercise.

Enhanced dynamic ratio Golden Section method uses the relative difference between the function evaluations of x_1 and x_2 to determine the next ratio to be used in partitioning the interval. For example, if the function evaluations of x_1 and x_2 is significantly different, then the ratio to be used should help move x_2 towards the right bracket in order to track the large dynamic changes in function evaluation better. Conversely, if the function evaluation of x_1 and x_2 is not too far apart, then the standard Golden ratio should be used to partition the interval.

Source Code Implementation

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

#Defining the test function
def test_func(x):

    y = 6*(x**2) + 5*x + 10

    return y

#Implementing normal Golden sections method
r = (np.sqrt(5) - 1)/2
A = 4 #Initial guess of left bracket
B = 10 #Initial guess of right bracket

iterations = 20
tolerance = 1e-1
golden_section_result_x1 = []
golden_section_result_x2 = []
x_axis_gs_iterations = []

for h in range(1, iterations):
    x2 = r*(B-A)
    x1 = (1-r)*(B-A)

    golden_section_result_x1.append(x1)
    golden_section_result_x2.append(x2)
    x_axis_gs_iterations.append(h)
```

```
x1_eval = test_func(x1)
x2_eval = test_func(x2)

if x1_eval < x2_eval:
    B = x2

else:
    A = x1

difference = np.linalg.norm(x1_eval - x2_eval)

if difference < tolerance:
    break

#Printing key variables
print("No. of iterations of normal GS,", h)
print("Value of left internal point of normal GS, ", x1)
print("Value of right internal point of normal GS, ", x2)

#Plotting performance plot of optimization algorithm
plt.figure(1)
plt.plot(x_axis_gs_iterations, golden_section_result_x1, 'b-', label = "Variation of x1")
plt.plot(x_axis_gs_iterations, golden_section_result_x2, 'r.', label = "Variation of x2")
plt.xlabel("Number of iterations")
plt.ylabel("Value of internal point")
plt.title("Variation of x1 and x2 in fixed ratio Golden section")
plt.legend()
plt.show()

#Implementing dynamic ratio Golden Section method
r = (np.sqrt(5) - 1)/2
A = 4 #Initial guess of left bracket
B = 10 #Initial guess of right bracket

iterations = 20
tolerance = 1e-1
dynamic_ratio_golden_section_result_x1 = []
dynamic_ratio_golden_section_result_x2 = []
x_axis_dynamic_ratio_gs_iterations = []
```

```
x2 = r*(B-A)
x1 = (1-r)*(B-A)

for h in range(1, iterations):

    dynamic_ratio_golden_section_result_x1.append(x1)
    dynamic_ratio_golden_section_result_x2.append(x2)
    x_axis_dynamic_ratio_gs_iterations.append(h)

    x1_eval = test_func(x1)
    x2_eval = test_func(x2)

    if x1_eval < x2_eval:
        value_ratio = x2_eval/x1_eval

        if value_ratio < 1.5:
            B = x2
            r = (np.sqrt(5) - 1)/2
            x2 = r*(B-A)
            x1 = (1-r)*(B-A)

        if value_ratio > 1.5:
            B = x2
            r = value_ratio/(value_ratio + 1)
            x2 = r*(B-A)
            x1 = (1-r)*(B-A)

    else:
        A = x1

    difference = np.linalg.norm(x1_eval - x2_eval)

    if difference < tolerance:
        break

#Printing variables
print("No. of iterations of dynamic ratio GS,", h)
```

```

print("Value of left internal point of dynamic ratio GS, ", x1)
print("Value of right internal point of dynamic ratio GS, ", x2)

#Plotting performance plot of optimization algorithm
plt.figure(2)
plt.plot(x_axis_dynamic_ratio_gs_iterations, dynamic_ratio_golden_section_result_x1, 'b-', label =
"Variation of x1")
plt.plot(x_axis_dynamic_ratio_gs_iterations, dynamic_ratio_golden_section_result_x2, 'r.', label =
"Variation of x2")
plt.xlabel("Number of iterations")
plt.ylabel("Value of internal point")
plt.title("Variation of x1 and x2 in dynamic ratio Golden section")
plt.legend()
plt.show()

```

Results and Discussion of Demonstrative Use of the Numerical Method

Demonstrative use of the dynamic ratio Golden section method is performed on a budget Lenovo laptop with AMD Ryzen 7000 series CPU (Model: 7520U at 2.80 GHz) and AMD Radeon integrated graphics (with 2 GB video RAM), 16 GB of RAM, and 512 GB solid state drive.

Shown here is the data from demonstrative use of the fixed ratio and dynamic ratio Golden section method in finding the minimum x for solving a quadratic objective function with minimum y -axis value of 10 at $x = 0$. Hence, the minimum x_1 and x_2 should be close to zero. The issue here is how fast both algorithms get to the minimum value of zero.

In Figure 1a, it is clear that the fixed ratio Golden section method does progressively move x_2 internal value point to zero over a course of 12 iterations. But, on Figure 1b, it is shocking to see that the dynamic ratio Golden section method could move the x_2 internal point to zero in just 3 iterations. Hence, the dynamic ratio Golden section method does show initial (preliminary) promise of faster convergence to the minimum point compared to the fixed ratio Golden section method.

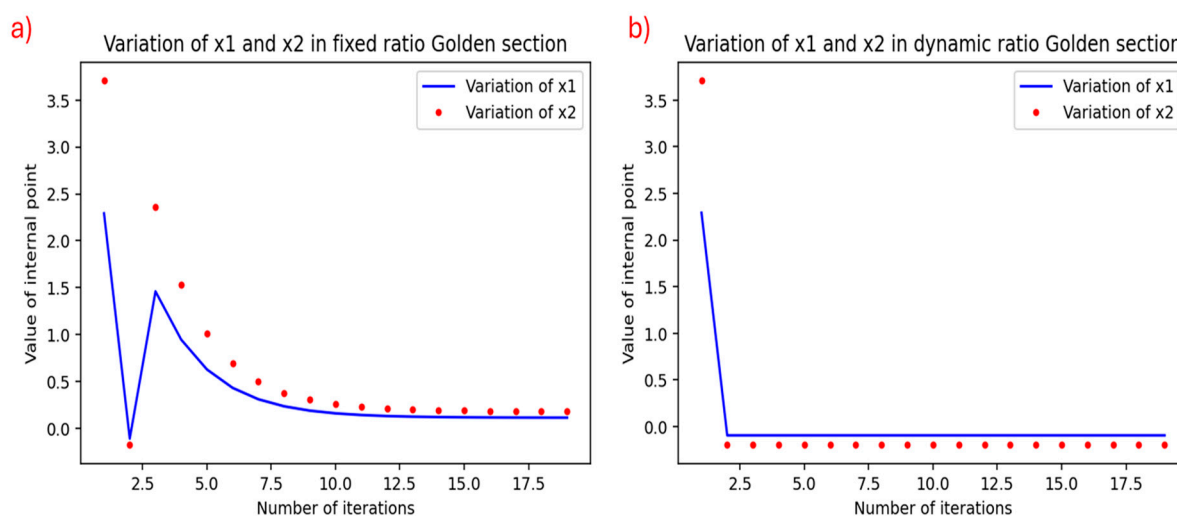


Figure 1. Data from demonstrative use of my dynamic ratio Golden section method in comparison with the fixed ratio Golden section method.

Conclusions

Numerical methods in all engineering disciplines seek to determine an approximate but fairly good estimate of the true optimum in different optimization scenarios. One good single variable optimization method is the Golden section method. In its typical implementation as a fixed ratio (i.e., Golden ratio) Golden section method, path to convergence to the minimum of maximal point is progressive. Here, a different method to implement Golden section method is described. Principally, the innovation here is the use of a dynamic ratio Golden section method in order to allow the capture or more precise following of the objective function curve, especially in regions of rapidly changing value. To this end, the dynamic ratio will partition the bracketing region and place the x_2 internal point closer to the right bracket in order to follow the rapidly changing function. Demonstrative use of this dynamic ratio Golden section method on a quadratic function with minimum y -value of 10 shows fast convergence to minimum point of zero in 3 iterations compared to 12 iterations for the conventional fixed ratio Golden section method.

Funding: No funding was used in this work.

Conflicts of interest: The author declares no conflicts of interest.

Reference

1. S. Hernandez, I. Flores, and J. Vázquez, "Improved Golden-Section Algorithm for the Multi-Item Replenishment Problem," *J. Appl. Res. Technol.*, vol. 10, June 2012.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.