

three-point-minima-search

May 23, 2025

1 Brute-Force Three-Point Minima Search for Continuous Polynomial Function

Here is a continuous polynomial function in single variable x : $2x^2 + 5x - 3$

The exact coordinates of the minima (global) of this polynomial are $(-1.25, -6.125)$.

We will try to find the minima of this polynomial by normal comparison between the values of the polynomial at different points. For this:

- range of x is fixed
- *interval* is based on the *divisions* of the range
- *interval* is the size of each *division*
- values of y are calculated for each x taking three values of x for each iteration
- function is being checked from left to right

```
[57]: def func_x(x):  
  
    # give value of the function for given value of x  
  
    y = 2*(x**2) + 5*x - 3  
  
    return y  
  
[58]: def divisions_n_interval():  
  
    no_of_divisions = int(input("Enter divisions: ")) # take user input for_  
    ↪number of divisions  
    print("Number of divisions:", no_of_divisions)  
  
    # predefining these values  
  
    upper_limit = 5.0  
    lower_limit = -5.0  
  
    interval = float((upper_limit - lower_limit) / no_of_divisions)  
  
    return (lower_limit, upper_limit, interval)
```

```
[59]: def find_minima():

    x1, upper_limit, interval = divisions_n_interval()

    print(f"Lower limit = {x1}\nUpper limit = {upper_limit}\nInterval = {interval}")

    x2 = x1 + interval
    x3 = x2 + interval

    while (x3 <= upper_limit):
        y1 = func_x(x1) # getting values of y for respective x
        y2 = func_x(x2)
        y3 = func_x(x3)

        if ((y1 >= y2) and (y2 <= y3)): # minima logic
            print(f"Minima (x, y) = ({x2}, {y2})")
            return # ends function

        x2 = x1 + interval # update x1, x2, x3
        x3 = x2 + interval
        x1 = x3

    else:
        print("Minima NOT found in given range.")
        return # ends function
```

```
[60]: find_minima()
```

```
Number of divisions: 10
Lower limit = -5.0
Upper limit = 5.0
Interval = 1.0
Minima (x, y) = (0.0, -3.0)
```

```
[61]: find_minima()
```

```
Number of divisions: 500
Lower limit = -5.0
Upper limit = 5.0
Interval = 0.02
Minima (x, y) = (-1.2200000000000188, -6.123200000000002)
```

```
[62]: find_minima()
```

```
Number of divisions: 5000
Lower limit = -5.0
Upper limit = 5.0
Interval = 0.002
```

Minima $(x, y) = (-1.2500000000003297, -6.125)$