Parabolic Interpolation Optimization

Noureddin Mohammed Lutfi
0197061
Computer Engineering Department
The University of Jordan
Amman, Jordan
nor0197061@ju.edu.jo,
lutfenour@gmail.com

***Abstract*—Parabolic optimization method uses parabolic interpolation algorithm iteratively to find local extrema of a given function, this method uses Lagrange method, parabolic interpolation and bracketing method by fitting parabolas to a given function, minimizing both interval and x minimum.**

Keywords—interpolation, parabolic, optimization, Lagrange, extremum, ***(****key words****)***

I. Introduction

Successive parabolic interpolation or parabolic optimization method is an optimization algorithm that is used to find local maximum and minimum points of a given continuous unimodal function by interpolating (fitting) parabolas (2nd order polynomial) to a given function, to understand this method you need to know what polynomial interpolation is and how does this algorithm work [1].

II. Interpolation

A. Polynomial Interpolation

First, interpolation is a method of deriving a function using given discrete data points and this function will pass through these given data points which allow us to draw a function that lies on the given points and helps us to determine what the unknown values in between these points could be. By using polynomial interpolation, which means to interpolate the given data set using polynomial with the lowest possible degree that passes through the data set, the number of points we require to interpolate a unique polynomial of degree (n) on a given data set is (n+1) distinct points. For example, if we want to make linear interpolation which means that n =1 then we need two points to interpolate the data set with a polynomial of degree one (linear function), three points for a polynomial of degree two and so on. This can be widely used to predict unknown values using a given data points and required interpolation function [2, 3].

Moreover, if we use linear interpolation with two given points, a line will be interpolated between the given two points and we can predict x and y values of any point that lies on this line, but if three points were used then a parabola will be interpolated using the three points and we can predict what x and y values of any point that lies on the parabola that were interpolated [2, 3].

B. Parabolic Interpolation

As earlier mentioned, parabolic interpolation is used to fit a parabola to a given data set, but there are many ways to interpolate a polynomial such as, Vandermonde matrix, Lagrange polynomial and many others. We will simply use Lagrange polynomial method to fit the given data points to a parabola and identify the formula of the function.

C. Lagrange method

Lagrange polynomial is used to interpolate a polynomial and get the formula of the polynomial you are trying to interpolate, simply we will show the $p(x)$ which represents the formula that will interpolate a parabola using three distinct points $[x_1, x_2, x_3]$, since it is our case of study [4].

$$p(x) = \frac{(x-x_2)(x-x_3)}{(x_1-x_2)(x_1-x_3)} f(x_1) + \frac{(x-x_1)(x-x_3)}{(x_2-x_1)(x_2-x_3)} f(x_2) + \frac{(x-x_1)(x-x_2)}{(x_3-x_1)(x_3-x_2)} f(x_3)$$

This formula will return the parabola equation that satisfies that the three points $[x_1, x_2, x_3]$ will lie on it.

Given that extrema points exist only if $\frac{dp}{dx} = 0$ since we assumed that this method only works for continuous unimodal functions, so x_4 expression derived by solving $\frac{dp}{dx} = 0$ will be as the following formula shows:

$$x_4 = x_2 - \frac{1}{2} . \frac{(x2-x1)^2[f(x_2)-f(x_3)] - (x_2-x_3)^2[f(x_2)-f(x_1)]}{(x2-x1)[f(x_2)-f(x_3)] - (x_2-x_3)[f(x_2)-f(x_1)]}$$

Which indicates that x_4 is the extremum value of $p(x)$ numerically calculated and we will apply in our optimization method, we won't need to calculate Lagrange polynomial, because all what we care about is the extremum point and this will be explained in detail later [4]. Here is an example of evaluating an extremum for a certain parabola if 3 distinct points are given as follows : (-1,3). (1,3) and (2,6), we can simply calculate x_4 by substituting the three points in the previous formula:

$$x_4 = 1 - \frac{1}{2} . \frac{(1--1)^2[3-6] - (1-2)^2[3-3]}{(1--1)[3-6] - (1-2)[3-3]}$$

When we calculate this, we get zero and this means there is an extremum at $x = 0$, which is true since the right function is $y = x^2 + 2$ and this function has an extremum at $x = 0$.

The function $y = x^2 + 2$ could be evaluated using Lagrange polynomial formula, but we don't need to use it and we just explained it earlier to show how to get the x_4 formula.

III. Parabolic Optimization

Earlier, we discussed some important mathematical concepts about interpolation and Lagrange polynomial, now we will use the previous parabolic interpolation and Lagrange method to make an optimization for a given function and evaluate its extremum value.

A. Algorithm

Let's suppose that we have a function $f(x)$ and a given 3 distinct initial points x_1, x_2 and x_3 that bounds the extremum value, and we need to calculate its local minimum, this method assumes that if we interpolate a parabola using the 3 points and then calculate the minimum using x_4 equation that we explained earlier, then x_4 will be the initial extremum guess of $f(x)$, but since there will be some error because of the difference in between the function $f(x)$ extremum value and the parabola extremum value then we need to recalculate three new points iteratively while updating the x_4 guess in each iteration by excluding the right x value until we get a parabola that best describes the objective function with a close shape and the difference between the real extremum value and the parabola extremum becomes acceptable due to a given error percentage or when it exceeds the maximum number of iterations given [5].

B. Bracketing Method

Assuming that x_1 is the most left value, x_2 is in the middle and x_3 is the most right value then:

If x_4 is on the left of x_2 and $f(x_4) < f(x_2)$ then we should change the interval such that, x_3 becomes x_2 and x_2 becomes x_4 (discard upper interval), but if x_4 is on the left of x_2 and $f(x_4) > f(x_2)$ then we should replace x_1 with x_4 and replace x_4 with x_2 since x_2 has a lower $f(x)$ then it will be closer to the extremum (discard lower interval) [4].

If x_4 is on the right of x_2, and $f(x_4) < f(x_2)$ then we should change the interval such that, x_1 becomes x_2 and x_2 becomes x_4 (discard lower interval), but if x_4 is on the right of x_2 and $f(x_4) > f(x_2)$ then we should replace x_3 with x_4 and replace x_4 with x_2 since x_2 has a lower $f(x)$ then it will be closer to the extremum (discard upper interval) [4].

Example

Assume that you have a function $f(x) = \frac{x^2}{5} - 7\sin(x)$ and we need to calculate its minimum value with three initial points $x_1 = 0, x_2 = 1$ and $x_3 = 4$, number of iterations (n=50) and error percentage (eps = 0.001).

First, we evaluate $x_4 = 1.5922$ and $f(x)$ has been evaluated for all points, then and we assume x_4 our first extremum value, with $f(x_1) = 0, f(x_2) = -5.6903$, $f(x_3) = 8.1643$, f(x4) = 1.5922, now we need to discard lower interval since x_4 is on the right of x_2, and $f(x_4) > f(x_2)$, according to Table I which shows how x values change, track how value of x_4 changes and gets closer to the extremum value and note that the values are taken at the beginning of each iteration.

Look at Fig. 1 which shows how does the parabola (green) look like compared to the function $f(x)$ (red) and you can notice that there is an error in the extremum value (1.592 for parabola and 1.486 for $f(x)$ which is the right value).

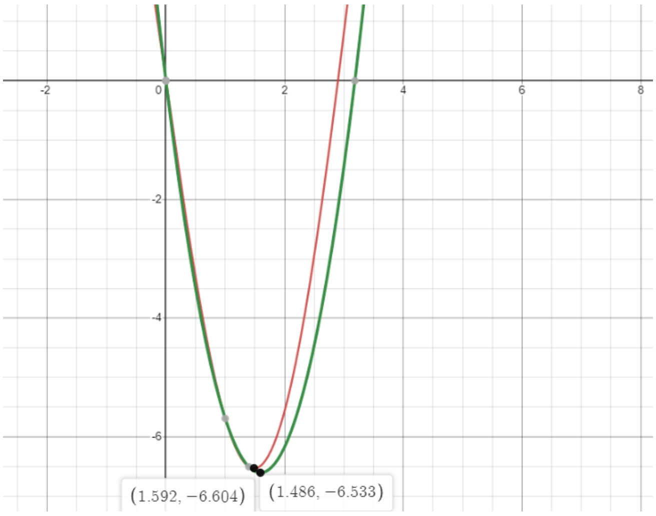

Fig 1. Interpolated Parabola and $f(x) = \frac{x^2}{5}$ - 7sin(x), n=1

In the next iteration (n=2) which Fig. 2 shows we can see how the new parabola minimum value is closer to 1.486 and we will keep fitting parabolas with new points until it converges to a certain error or number of iterations is exceeded, note also If x4 is equal to zero this method will terminate and take x4 as a guess since $\frac{0}{0}$ is undefined, so we can't calculate error between iterations.

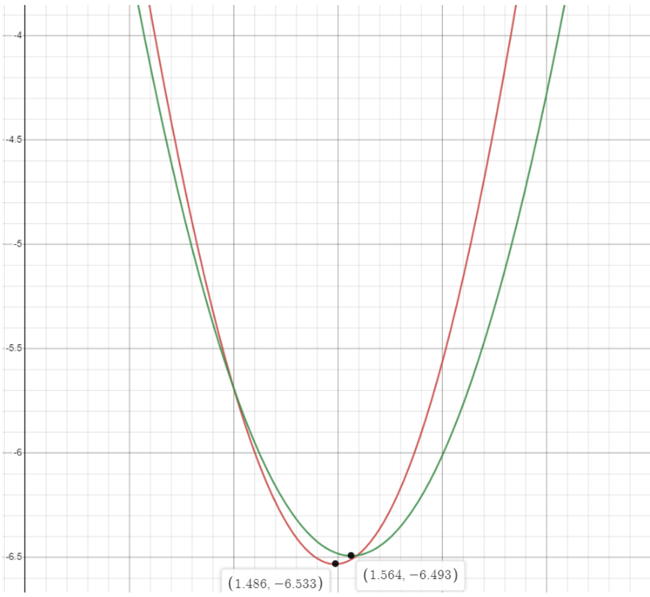

Fig 2. Interpolated Parabola and $f(x) = \frac{x^2}{5}$ - 7sin(x), n=2

TABLE I. PARABOLIC OPTIMIZATION x VALUES SUMMARIZATION

n	x_1	x_2	x_3	x_4
1	0	1	4	1.592
2	1	1.592	4	1.5639
3	1	1.5639	1.5922	1.4835
4	1	1.4835	1.5639	1.4847
5	1.4835	1.4847	1.5639	1.4858
6	1.4847	1.4858	1.5639	1.4858

TABLE II. PARABOLIC OPTIMIZATION $f(x)$ VALUES SUMMARIZATION

n	$f(x_1)$	$f(x_2)$	$f(x_3)$	$f(x_4)$
1	0	-5.6903	8.4976	-6.4914
2	-5.6903	-6.4914	8.4976	-6.5107
3	-5.6903	-6.5107	-6.4914	-6.5332
4	-5.6903	-6.5332	-6.5107	-6.5332
5	-6.5332	-6.5332	-6.5107	-6.5332
6	-6.5332	-6.5332	-6.5107	-6.5332

At the end of the optimization method, we get the results as x_4 converges to 1.4858 and $f(x_4)$ converges to -6.5332 and if we draw this function, we can see that this function minimum converges at (1.4858, -6.5332) as Fig 3 shows.

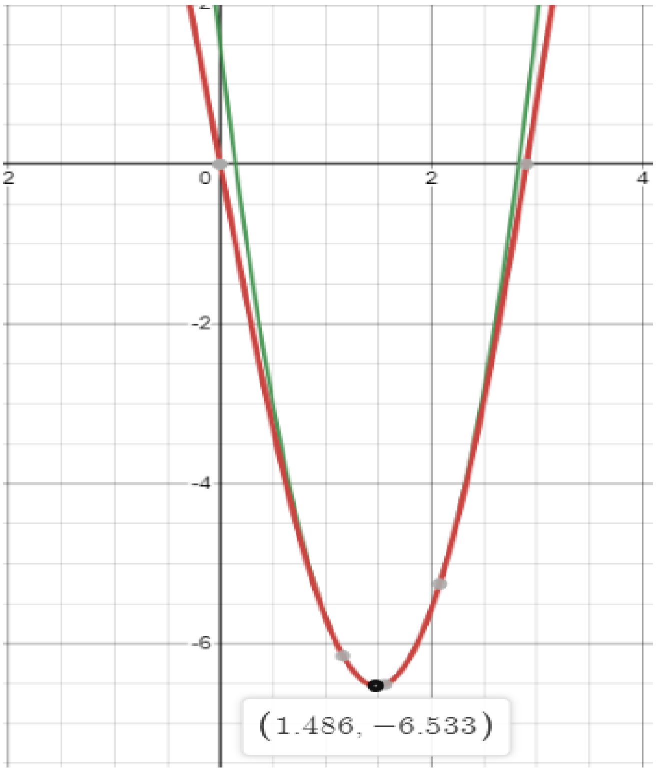

Fig 3. Interpolated Parabola and $f(x) = \frac{x^2}{5}$ - 7sin(x), n=6

IV. EXTRA

What if we want to calculate the maximum, simply you can multiply $f(x)$ by -1 and calculate its minimum then multiply the value of $f(x)$ by -1 to get the maximum extremum, look at the following figures Fig 4. and Fig 5. which shows the implementation of parabolic optimization method and the output of function (p_inter) used on the previous $f(x) = \frac{x^2}{5}$ - 7sin(x) example, in addition this method can calculate the maximum extremum too if we give the parameter max = 1 it calculates the maximum instead of minimum.

```
function [xmin,ymin] = p_inter(f,x1,x2,x3,iter,eps,max)
n=1; %number of iterations initially = 1
if(max == 1) %if you want to calculate maximum not minimum
point
  f=@(x)-(f(x)); %replace the function with f(x) * -1
end
while(true) %keep iterating until a break command
 %now we evaluate x4 which will be our extremum guess
x4 = x2 -(1/2)*(((x2-x1)^2*(f(x2)-f(x3))-(x2-x3)^2*(f(x2)-
f(x1)))/((x2-x1)*(f(x2)-f(x3))-(x2-x3)*(f(x2)-f(x1))));
fx1=f(x1);
fx2=f(x2);
fx3=f(x3);
fx4=f(x4);
if(x4<x2) %if x4 is on the left of x2
    if(f(x4)<f(x2))
       x3=x2; %replace x3 with x2
       x2=x4; %replace x2 with x4
    elseif(f(x4)>f(x2))
        x1=x4; %replace x1 with x4
        x4=x2;
    end
elseif(x4>x2) %if x4 is on the right of x2
     if(f(x4)<f(x2))
       x1=x2; %replace x1 with x2
       x2=x4; %replace x2 with x4
     elseif(f(x4)>f(x2))
        x3=x4; %replace x3 with x4
        x4=x2;
     end
end
if(n>1) %if number of iterations more than 1
    e = abs(x4-xold)/x4; %now we can evaluate eps since we
must have more than 1 iteration
    if(e<=eps || x4==0  || n>iter)
       %if we get the desired eps or x4 = 0 since it cant
be zero or it will be zero/zero
       %or if we exceed the number of desired iterations
        xmin=x4; %then xmin will be the last evaluated x4
        ymin = f(xmin);%compute y from f(xmin)
        break; %exit the loop
    end
end
xold = x4; %save the x4 value of the current iteration to
calculate eps
n = n+1; %update number of iterations
end
if(max==1) %if you want to calculate maximum not minimum
point
    ymin = ymin*-1; %then multiply y by -1
end
end
```

Fig 4. Parabolic Optimization Implementation – MATLAB CODE

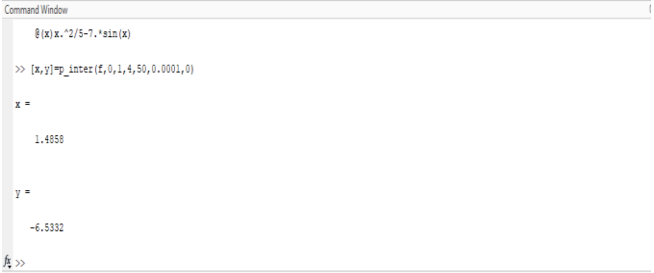

Fig 5. Output of The MATLAB Function Used in Fig 4.

Furthermore, this method is used by fminbnd in MATLAB as well as golden section search since this method has disadvantages such as, no guarantee to get the extremum point, for example if the three points are collinear then the resulting is degenerate and will be a line segment and it will fail, even this method is faster than golden-section search, but it doesn't always converge and has some drawbacks.

V. Conclusion

Finally, Parabolic optimization using the parabolic interpolation algorithm is fast and easy to implement but doesn't always converge and since it has some disadvantages, we need to combine it with golden section search algorithm, which is slower but more reliable, so we get the right answer [1].

References

[1] "Successive parabolic interpolation." wikipedia, https://en.wikipedia.org/wiki/Successive_parabolic_interpolation.

[2] "Polynomial interpolation.", wikipedia, https://en.wikipedia.org/wiki/Polynomial_interpolation.

[3] "Interpolation." byjus, https://byjus.com/maths/interpolation/.

[4] "SECTION 5: ROOT FINDING AND OPTIMIZATION.", https://web.engr.oregonstate.edu/~webbky/MAE4020_5020_files/Section%205%20Root%20Finding%20and%20Optimization.pdf.

[5] "Lecture 14: Optimization II" neuron, https://neuron.eng.wayne.edu/auth/ece3040/lectures/lecture14.pdf.

[6] "Successive parabolic interpolation" wikipedia, https://en.wikipedia.org/wiki/Successive_parabolic_interpolation.