|  |
| --- |
|  |
| Optimization Labs Report |
| Submitted By  Emre Ozan Alkan  Chinmay Samant  Priyanka Phutane |
|  |
|  |
|  |
|  |

|  |
| --- |
|  |

**Function minimization in 1D**

For the problems in this section, we consider following function

Goal of this lab is to find minimum of the given function using following methods.

**Brute Force Method:**

In this method, we use matlab's min() function to find minimum of the function. First we calculate all the values of the function. Store them in an array and then find the minimum of it using min(). As the function is in 1D space, we only define values limits for *x*.

If we do not have to use in built function of matlab, we would have used the following algorithm to find the minimum.

1. Store all values of *f(x)* in an array.
2. Compare first value with the second one
3. If the first is smaller than the second, swap them.
4. If not, then proceed to the comparison with 3rd value.
5. Repeat the step 2 to 4 for all the values in the array.
6. Display the first variable which is the smallest.

**Result:**

**Min(y) = -0..7294**

**Brent's Method :**

In this method we fit a parabola to the function at three unique points. At each iteration we replace the earlier point with the new minimum of the fitted parabola.

We chose a triplet of points a, b and c such that f(c) < f(a) and f(b) > f(c). Here we determine the minimum *x* of the parabola which is the set of points. The point updates the triplet with the new *x*. The point *x* of the parabola is can be found from the equation written below,

We get the following results after keeping the same interval for our function as we has in the previous method. Following graph shows the result.

**Function minimization in 2D:**

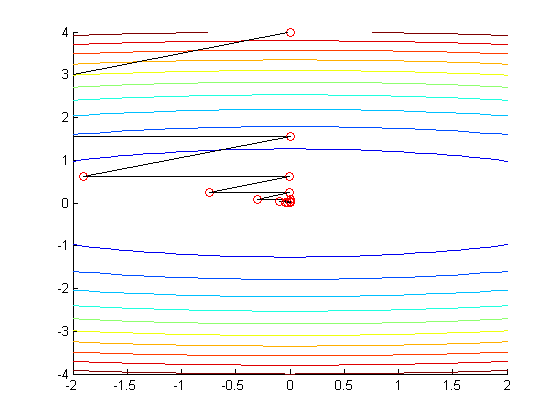
We consider the following equation.

**Arbitrary Line Search:**

In this method, first we draw our 2D function and then we search for a minimum in any arbitrary direction along the line. We create a vector to store all the co-ordinates of the line. Then using brute force algorithm, we find minimum of the line along that direction. With the minimum, we set the end point of the line.

After this, we repeat the process in another arbitrary direction. We draw a line in that direction, draw a line until we find the minimum. We keep on going in both direction until we reach the threshold value. The threshold value is the distance between starting point and end point. When it is reduced below the threshold level, we stop the iteration process.

**Result:**



**Powell's Method**

In Powell's method, we start with a point and move in a predefined direction. We define the end point of the primary line which is minimum along that line. Then instead of going in any arbitrary direction like in Arbitrary line search