Golden Section Search for the Mode of a Function

Gellért Peresztegi-Nagy

October 16, 2016

1 Question 1

(We choose $[x, b]$ to be the new interval, but as the whole interval is symmetric we basically prove both options.)

Let $L = b - x$ and $r \cdot L = b - x$. Then $x - a = b - y = (1 - r) \cdot L$ and the section in the middle $y - x = (2r - 1) \cdot L$.

1.1 In case we put the new point to the left from y

In this case the ratio of the intervals must remain:

$$\frac{b-x}{b-a} = \frac{y-x}{b-x}$$

$$\frac{rL}{L} = \frac{(2r-1)L}{rL}$$

$$r^2 = 2r - 1$$

$$r^2 - 2r + 1 = 0$$

In this case $r = 1$ which can not be a solution

1.2 In case we put the new point to the right from y

The ratio of the intervals again must remain:

$$\frac{b-x}{b-a} = \frac{b-y}{b-x}$$

$$r = \frac{(1-r)L}{rL}$$

$$r^2 + r - 1 = 0$$

$$r_{1,2} = \frac{-1 \pm \sqrt{1 - 4 \cdot (-1)}}{2}$$

$$r_1 = -\frac{1+\sqrt{5}}{2}$$

$$r_2 = -\frac{1-\sqrt{5}}{2} = \frac{\sqrt{5}-1}{2}$$

We choose r_2 which is the golden section constant proving that the new subinterval is already divided in golden section.

(Similarly, we can deduce that if we choose $[a, y]$ to be the new interval, we need to put the new point to the left from x.)

2 Programming - Implementing the algorithm

2.1 Program

The following object-oriented program searches for the mode of the given function in a given interval.

```
d't'żusing System;

namespace GoldenSectionSearch
{
  class MainClass
  {
    // The function according to the task
    public static double F(double x)
    {
      return (-4.0) * Math.Pow(x, 4.0) + 1.0 + x + Math.Pow(x, 2.0)
    ;
    }


    public static void Main(string[] args)
    {
      GSSearch s = new GSSearch(0, 1, F, 0.0001);
      double result = s.search();

      Console.WriteLine("The maximum value of the function at {0}
    is {1}.",result, F(result));

      Console.ReadKey();
    }

    public class GSSearch
    {
      public double PRECISION;
      public static double phi = ((Math.Sqrt(5) - 1) / 2);
      Func<double, double> f;
      double start, x, y, end, fX, fY;

      public GSSearch(double start, double end, Func<double, double
    > func, double precision)
      {
        this.start = start;
        this.end = end;

        this.f = func;

        this.x = end - (end - start) * phi;
        this.y = end - (x - start);

        this.fX = f(x);
        this.fY = f(y);

        this.PRECISION = precision;
      }

      public double search()
      {
        while (2 * PRECISION <= end - start)
        {
          // In case the two function values are equal, we pick the
     interval in the middle
```

```
            if (fX == fY)
            {
              this.start = x;
              this.end = y;

              this.x = end - (end - start) * phi;
              this.y = end - (x - start);
              this.fX = f(x);
              this.fY = f(y);
            }
            // If the function value on the left is grater, the mode must be in the (start,y) interval
            else if (fX > fY)
            {
              this.end = y;

              this.y = x;
              this.fY = fX;

              this.x = start + end - y;
              this.fX = f(x);
            }
            // If the function value on the right is grater, the mode must be in the (x,end) interval
            else {
              this.start = x;

              this.x = y;
              this.fX = fY;

              this.y = end - (x - start);
              this.fY = f(y);
            }
          }
          return x;
        }
      }
    }
}
```

GoldenSectionSearch.cs

2.1.1

In case $f(x) == f(y)$ the algorithm chooses $[x, y]$ to be the new interval and gets new x and y values from the function (using 2 function evaluations).

2.1.2

It is preferable to use $b-x = y-a$ (Equation (2)) to locate the point for the second function evaluation in each new subinterval as it requires less computation ($x = a + b - y$, $y = a + b - x$).

2.1.3

If the mode's position were located at an end-point of the original interval, the algorithm would always choose the left/right interval (depending on at which end the mode is), and would shrink down the interval by the golden section ratio each step until the required precision is attained.

2.2 Question 2 - Testing the algorithm

The function $f(x) = 1 + x + x^2 - 4x^4$ is unimodal on the $[0, 1]$ interval because of the following. 1 is just a constant, x is always increasing, x^2 and $-4x^4$ are unimodal with the same maximum place. Hence, the sum of these must be unimodal so we can use the algorithm on this function and it gives the following output:
Maximum: 1,49999999791352
Precision: 0,0001
# of evaluations: 18

3 Question 3 - Computational cost

The most time consuming part of these types of algorithms is the time required to evaluate the function (to get the value of the function at a given position). In case we have an alternative algorithm in which the subdivisions are defined by $x-a = \frac{1}{3}(b-a)$, $y-a = \frac{2}{3}(b-a)$, it would shrink the interval to $\frac{2}{3}$ of the original in each step, but would require $2n$ evaluations of the function (n = number of steps). The golden section search algorithm shrinks the interval to $\frac{\sqrt{5}-1}{2}$ of the original in each step, and require $n + 1$ evaluations of the function. Hence, the golden section search method is better. (The efficiency of the golden section search method is not much less than that of the theoretical optimal method.)

4 Theoretical considerations

4.1 Question 4

Only the slope of the function determines the numerical accuracy that is attainable through the golden section search method.

4.2 Question 5

If the mode was located to some accuracy, the corresponding accuracy in the height of the mode would be the slope times the accuracy (in case the function has a slope there).

5 Application: Optimization of the gramophone design

5.1 Program - Implementing the code

We need to optimize the stylus to track so that its direction diverges as little as possible from the alignment with the groove; that is so that as the stylus moves from the outer edge of the record to the inner one, the maximum absolute value of the angle closed by the stylus and the groove is minimised.
Taking $r = 6.5$ (inner radius), $R = 16$ (outer radius), $l = 14$ (the length of the tone-arm) (all in cm), the algorithm finds the optimum values of d (the distance

of the tone-arm pivot from the centre of the record), and θ (the angle between the tone-arm the direction of the stylus).

```
d't'żusing System;

namespace GramophoneOptimization
{
  class MainClass
  {
    const double r = 6.5;
    const double R = 16;
    const double l = 24;
    const double precision = 0.000001;

    static double phi(double d, double x) {
      return Math.Asin((l * l + x * x - d * d) / (2 * l * x));
      //return Math.Acos((d * d - l * l - x * x) / (2 * l * x));
    }

    static double deltaPhi(double d)
    {
      double max = phi(d,new GSSearch(r, R, x => phi(d, x),
    precision).search());
      double min = Math.Min(phi(d, r), phi(d, R));
      return max - min;
    }

    public static void Main(string[] args)
    {
      double d_optimal = new GSSearch(Math.Max(R, l - r), l + r, d
    => -deltaPhi(d), precision).search();
      double theta = (phi(d_optimal, r) + phi(d_optimal, R)) / 2.0;

      Console.WriteLine("d = {0}", Math.Round(d_optimal, 5));
      Console.WriteLine("Theta = {0}", Math.Round(theta, 5));
    }

    public class GSSearch
    {
      public double PRECISION;
      public static double phi = ((Math.Sqrt(5) - 1) / 2);
      Func<double, double> f;
      double start, x, y, end, fX, fY;

      public GSSearch(double start, double end, Func<double, double
    > func, double precision)
      {
        this.start = start;
        this.end = end;

        this.f = func;

        this.x = end - (end - start) * phi;
        this.y = end - (x - start);

        this.fX = f(x);
        this.fY = f(y);

        this.PRECISION = precision;
      }

      public double search()
      {
```

```
58          while (2 * PRECISION <= end - start)
            {
60            // In case the two function values are equal, we pick the
      interval in the middle
              if (fX == fY)
62            {
                this.start = x;
64              this.end = y;

66              this.x = end - (end - start) * phi;
                this.y = end - (x - start);
68              this.fX = f(x);
                this.fY = f(y);
70            }
              // If the function value on the left is grater, the mode
     must be in the (start,y) interval
72            else if (fX > fY)
              {
74              this.end = y;

76              this.y = x;
                this.fY = fX;
78
                this.x = start + end - y;
80              this.fX = f(x);
              }
82            // If the function value on the right is grater, the mode
      must be in the (x,end) interval
              else {
84              this.start = x;

86              this.x = y;
                this.fX = fY;
88
                this.y = end - (x - start);
90              this.fY = f(y);
              }
92          }
            return x;
94        }
      }
96  }
}
```

GramophoneOptimization.cs

5.2 Question 6

5.2.1

In the QPS triangle the sum of two sides must be greater than the third side:

$$l + r > d \implies 30.5 > d$$

$$d + r > l \implies d > 17.5$$

Hence, in the outer iteration the algorithm uses the $[17.5, 30.5]$ interval.
In the inner iteration (inside the golden section search object) the algorithm uses the given r and R values as the ends of the interval.

5.2.2

The inner function on which we perform the golden section search is clearly unimodal on the $[r;R]$ interval. It is the arccos of a fraction with a quadratic function in the numerator and a linear function in the denominator. The arccos is decreasing, the quadratic function is unimodal and the linear function is increasing as well. Hence, the inner function must be unimodal in the on the $[r;R]$ interval.

5.2.3

Approximately $\log_{\frac{\sqrt{5}-1}{2}}(\frac{k}{R-r}) \cdot \log_{\frac{\sqrt{5}-1}{2}}(\frac{k}{2r}) \approx 96$ inner function evaluations are needed to find the length of d to one decimal place ($k = 0.1$). The required accuracy highly affects this number, the number of evaluations go up if we need more accurate results.