Optimization

Basics and 1D Optimization

Sachin Shanbhag

Department of Scientific Computing
Florida State University,
Tallahassee, FL 32306.

References

- S. Chapra and R. Canale, Numerical Methods for Engineers
- A. Greenbaum and T. Chartier, Numerical Methods: Design, Analysis, and Computer Implementation of Algorithms
- Carnahan and Wilkes, Applied Numerical Methods, University of Michigan, Class Notes, 1996.
- Burden and Faires, Numerical Analysis, 1993
- Pal, Numerical Analysis for Scientists and Engineers, 2007.
- M. Heath, Scientific Computing: An Introductory Survey
- wikipedia.org

Contents

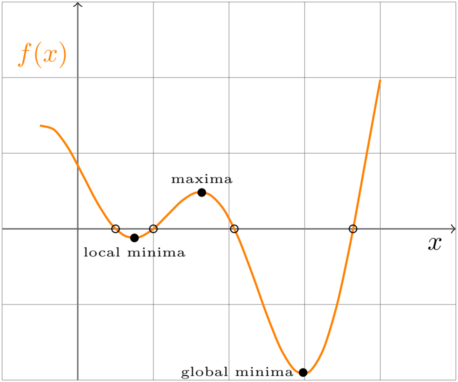
- Basics and Background
- Optimization: Single variable
  - Golden Section
  - Quadratic Interpolation
  - Newton's Method
- Multidimensional Optimization
  - Steepest Descent
  - Newton's Method
  - BFGS Method: Quasi-Newton
- Miscellaneous

Motivation

- Many application in science and engineering involve a compromise between factors that pull in opposite directions
  Example: In the design of automobiles, planes or bridges, we may want materials to be as light as possible (small thickness), and as strong as possible (large thickness).
- The minima or maxima may allow us to predict something useful
  Example: From physics, we know that systems prefer the lowest energy states (from marbles in a bowl to the protein folding problem)
- Other examples: Inventory control, optimal planning and scheduling, optimal oil pipelines/electric networks etc.

Background

In the simplest case, consider a generic nonlinear function $f(x)$



It can have multiple roots, and multiple minima and maxima. We have learned how to find the roots. In this part, we will learn how to find the *optima*.

Conditions for Optima

- For both minima and maxima, at the optima x^*,

$$f'(x^*) = 0$$

- For maxima:

$$f''(x^*) < 0$$

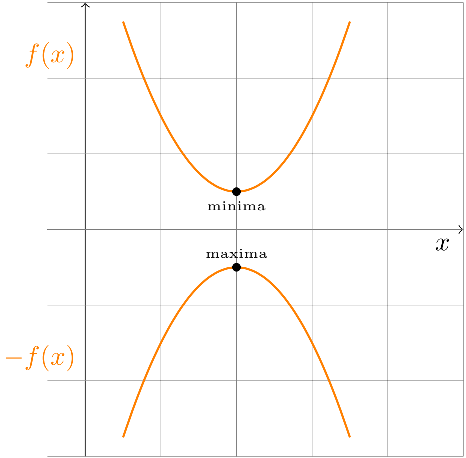
- For minima:

$$f''(x^*) > 0$$

- When $f''(x^*) = 0$, you may have an inflection point
  Example: For $f(x) = x^3$ at $x = 0$, both $f''(x = 0) = 0$ and $f''(x = 0) = 0$

Minimization/Maximization

Minimization of $f(x)$ is equivalent to the maximization of $-f(x)$.



So we don't need separate methods for minima and maxima

Local and Global Optima

- Just as a general nonlinear function can have several roots, it can also have several minima (or maxima)
- There may be several points x_i^* for which $f'(x_i^*) = 0$ and $f''(x_i^*) > 0$. That is the minimum, in general, is not unique.
- Each of these points are called *local* minima. Note that $f(x_i^*)$ are also not the same.
- The global minima x_G^* is given by the x_i^* for which $f(x_i^*)$ is the smallest
- For general complex nonlinear problems, determining the global minima is nontrivial

Dimensions

- So far we talked about minimization of $f(x)$: optimization of a function of a single variable
- In general,

  $$f(\mathbf{x}) = f(x_1, x_2, ..., x_n)$$

  can be a function of many variables.
- The optimization

  $$\min_{\mathbf{x}} f(\mathbf{x})$$

  is called multidimensional optimization
- In this class, we will consider optimization in two dimensions to introduce the idea

Constrained and Unconstrained

- So far we have considered unconstrained optimization

$$\min_x f(x)$$

- Often, we are interested in constrained optimization where we want to solve the minimization problem with some constraints placed on x
- Equality constraints

$$e_i(x) = b_i, \quad i = 1, ..., m$$

- Inequality constraints

$$g_i(x) \leq a_i, \quad i = 1, ..., p$$

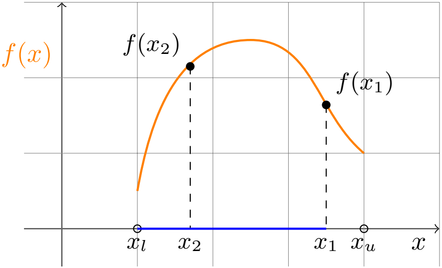
- In this class, we will only consider unconstrained optimization

1D Unconstrained Optimization

- The first method we will consider to minimize $f(x)$ is Golden-Section Search
- In many ways this is the analogous to the bisection method for finding roots of a nonlinear equation
- It begins by defining an interval given by x_l and x_u that contain a single maxima (or minima).
- The function is differentiable and *unimodal*: strictly increasing, and then strictly decreasing (or vice versa for minima).
- In bisection, we considered the function at the midpoint $x_r = 0.5(x_l + x_u)$ to determine which part of the original interval to discard
- Here we need two internal points x_1 and x_2 to determine whether a minimum occurred.

Golden Section Search

- Assume, we have bracketed a single maxima in the interval $[x_l, x_u]$, and we probe the function at two internal points x_1 and x_2 as shown below

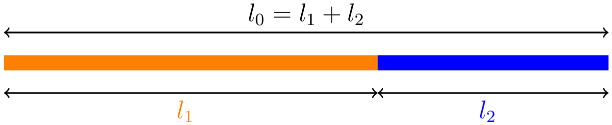


- Claim: If $f(x_2) > f(x_1)$, the maxima is confined to the interval $[x_l, x_1]$.
- Note: As yet, we don't know if it lies in $[x_l, x_2]$ or $[x_2, x_1]$

Golden Section Search

- Note: If $f(x_1) > f(x_2)$ then the maxima is confined to the interval $[x_2, x_u]$
- We have thus narrowed the interval containing the maxima from $[x_l, x_u]$ to $[x_l, x_1]$ (say)
- We can repeat this process until we have bracketed the maxima to an interval that is small enough for us
- The remaining question: how do we choose x_1 and x_2?
- One idea: Divide the interval into three equal parts with $x_2 = x_l + (1/3)(x_u - x_l)$, and $x_1 = x_u - (1/3)(x_u - x_l)$.
- This works, but we can do better!

Golden Ratio



Partition the line segment l_0 into two parts so that:

$$\frac{l_1}{l_0} = \frac{l_2}{l_1} = R$$

Divide $l_0 = l_1 + l_2$ by l_1 to obtain,

$$R = \frac{1}{1+R}$$

That is $R^2 + R - 1 = 0$, the positive root[*] of which is

$$R = \frac{\sqrt{5}-1}{2} = 0.61803$$

[*] sometimes called the conjugate of the "true" golden ratio $1/R$

Golden Ratio

- This "ratio" is called the Golden Ratio and has many interesting properties in nature, art and math[†]
- When the proportions of objects are designed according to the golden ratio, they are found to be aesthetically pleasing.
- R is also related to the Fibonacci series

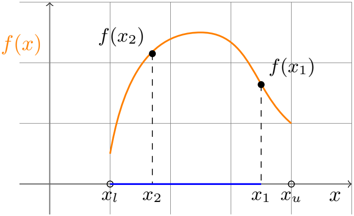
$$0, 1, 1, 2, 3, 5, 8, 13, 21, 34...$$

- The ratio of successive terms gradually approaches R $(0/1 = 0; 1/1 = 1; 1/2 = 0.5; ...21/34 = 0.617$ etc.)

[†]see `http://en.wikipedia.org/wiki/Golden_ratio`

Golden Ratio

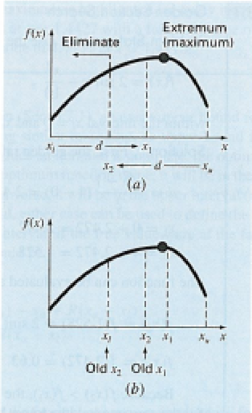
- How does it help us here?
- Key idea: Once we've determined which subinterval to throw away, we can reuse one of the points
- Hence we need to generate only one new point (and evaluate the function there) at each iteration.



- With $d = (\sqrt{5} - 1)/2(x_u - x_l)$, set $x_1 = x_l + d$ and $x_2 = x_u - d$.

Golden Section Search

- Since x_1 and x_2 are chosen according to the Golden ratio, when we discard the interval $[x_l, x_2]$, the old x_2 becomes the new x_l, and the old x_1 becomes the new x_2.
- The previous figure was not drawn to scale: Here is a picture from Chapra and Canale.
- We only need to recompute one new internal point



Algorithm for Determining Maxima

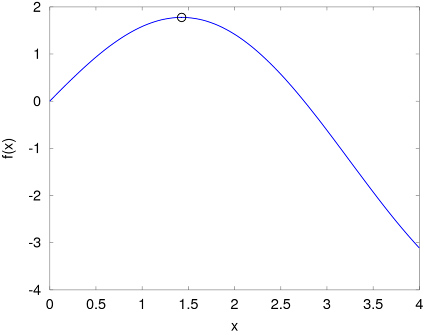
```
function xopt = golden(xl, xu, tol)

  R  = (sqrt(5) - 1)/2            % golden ratio
  d  = R * (xu - xl)

  x1 = xl + d; f1 = f(x1)
  x2 = xu - d; f2 = f(x2)

  while (xu - xl > tol)

    d = R * d                     % interval shrinks by factor R

    if (f1 > f2)
      xl = x2
      x2 = x1
      x1 = xl + d
      f2 = f1
      f1 = f(x1)
    else
      xu = x1
      x1 = x2
      x2 = xu - d
      f1 = f2
      f2 = f(x2)
    endif

  endwhile

  xopt = (xu + xl)/2

endfunction
```

Example

Find the maximum of the function

$$f(x) = 2\sin x - \frac{x^2}{10}$$

in the interval $[0, 4]$ and a tolerance of $\epsilon = 10^{-3}$



Example

n	x_l	$f(x_l)$	x_2	$f(x_2)$	x_1	$f(x_1)$	x_u	$f(x_u)$
0	0.000	0.000	1.528	1.765	2.472	0.630	4.000	-3.114
1	0.000	0.000	0.944	1.531	1.528	1.765	2.472	0.630
2	0.944	1.531	1.528	1.765	1.889	1.543	2.472	0.630
3	0.944	1.531	1.305	1.759	1.528	1.765	1.889	1.543
4	1.305	1.759	1.528	1.765	1.666	1.714	1.889	1.543
5	1.305	1.759	1.443	1.775	1.528	1.765	1.666	1.714
10	1.410	1.775	1.423	1.776	1.430	1.776	1.443	1.775
15	1.426	1.776	1.427	1.776	1.427	1.776	1.428	1.776
17	1.427	1.776	1.427	1.776	1.427	1.776	1.428	1.776

Solution at the end of 17 iterations is $x^* = 1.4273$.
True solution by solving $f'(x) = 0$ is 1.4276.

Convergence

- The interval shrinks by a factor of $R \approx 0.618$ after every iteration
- Thus after n iterations the interval size is 0.618^n
- While this is not as fast as bisection (0.5^n), it is still linear and quite fast
- To attain a tolerance level of ϵ the number of iterations required can be obtained from

$$\epsilon = R^n(x_u - x_l),$$

or

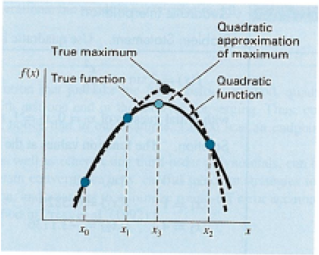
$$n = \frac{\log\left(\frac{\epsilon}{x_u - x_l}\right)}{\log R}$$

Quadratic Interpolation

- Like bisection method, golden section search uses function values only to compare them
- In non-linear equations we improved the convergence behavior by using the regula falsi method
- Here, a similar idea is used in quadratic interpolation or successive parabolic interpolation
- Key idea: Given three points (x_0, f_0), (x_1, f_1), and (x_2, f_2) which describe the interval $[x_0, x_2]$ containing the maxima of a unimodal function $f(x)$, the idea is to interpolate a parabola through the three points and use the maximum of the parabola as an estimate of the optimum
- One of the four points may then be discarded using a criterion similar to that of golden search

Quadratic Interpolation

- From Chapra and Canale:



- It can be shown using some algebraic manipulations that the given (x_0, f_0), (x_1, f_1), and (x_2, f_2), x_3 can be determined via:

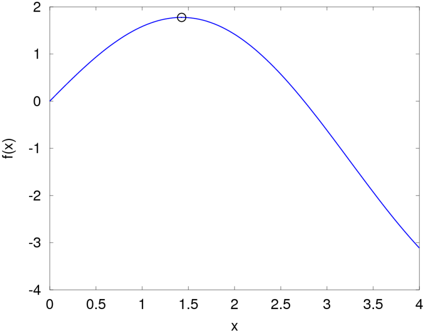
$$x_3 = \frac{f_0(x_1^2 - x_2^2) + f_1(x_2^2 - x_0^2) + f_2(x_0^2 - x_1^2)}{2f_0(x_1 - x_2) + 2f_1(x_2 - x_0) + 2f_2(x_0 - x_1)}$$

Example

Find the maximum of the function

$$f(x) = 2\sin x - \frac{x^2}{10}$$

using quadratic interpolation with $x_0 = 0$, $x_1 = 1$, and $x_2 = 4$ and a tolerance of $\epsilon = 10^{-3}$



Example

n	x_0	$f(x_0)$	x_1	$f(x_1)$	x_3	$f(x_3)$	x_2	$f(x_2)$
1	0.000	0.000	1.000	1.583	1.506	1.769	4.000	-3.114
2	1.000	1.583	1.506	1.769	1.490	1.771	4.000	-3.114
3	1.000	1.583	1.490	1.771	1.426	1.776	1.506	1.769
4	1.000	1.583	1.426	1.776	1.427	1.776	1.490	1.771
5	1.426	1.776	1.427	1.776	1.428	1.776	1.490	1.771
6	1.427	1.776	1.428	1.776	1.428	1.776	1.490	1.771
7	1.428	1.776	1.428	1.776	1.428	1.776	1.490	1.771
8	1.428	1.776	1.428	1.776	1.428	1.776	1.490	1.771

Solution at the end of 8 iterations is $x^* = 1.4276$.

When the maxima is bound reasonably tightly, the convergence is superlinear. It can be shown that the convergence rate ≈ 1.324.

Code

The central "decision-making" part of the algorithm is slightly complicated:

```
while ((x2 - x0) > tol)

  num = f0 * (x1^2 - x2^2) + f1 * (x2^2 - x0^2) + f2 * (x0^2 - x1^2);
  den = 2*f0 * (x1 - x2) + 2*f1 * (x2 - x0) + 2 * f2 * (x0 - x1);
  x3  = num/den;
  f3  = f(x3);

  if (x3 > x1)

    if(f3 > f1)
      x0 = x1; f0 = f1;
      x1 = x3; f1 = f3;
    else
      x2 = x3; f2 = f3;
    end

  else

    if(f3 > f1)
      x2 = x1; f2 = f1;
      x1 = x3; f1 = f3;
    else
      x0 = x3; f0 = f3;
    end

  end

end
```

Newton's Method

- To find the maxima of $f(x)$, find the zero of $f'(x)$
- Use usual Newton's method to find the zero of $f'(x)$

$$x_{i+1} = x_i - \frac{f'(x_i)}{f''(x_i)}$$

- The advantages (speed) and disadvantages (divergence) of this method have been described earlier
- In addition, here, one has to evaluate both $f'(x)$ and $f''(x)$.