SWFR ENG 4TE3 (6TE3)
COMP SCI 4TE3 (6TE3)
Continuous Optimization Algorithm

Line Search

Computing and Software Chair
McMaster University
What is line search?

Minimizing an objective function

\[
\min_{x \in \mathbb{R}^n} f(x).
\]

Starting from \(x^0 (k = 0)\); general iterate \(x^k\), search direction \(d^k\).

**Find step size parameter \(\alpha\) such that \(\min \phi(\alpha) = f(x^k + \alpha d^k)\).**

A line search is a subroutine of the algorithm to determine a step size such that, at the new iterate \(x^{k+1} = x^k + \alpha d^k\), the objective has a lower value (or, in some sense, \(x^{k+1}\) is a better point).

In the subroutine of line search, we are minimizing the univariate objective function \(f(x) = \phi(\alpha)\) for \(\alpha \in [l, u]\).

**Finding the zero of a univariate function**

If \(f(x)\) is twice continuously differentiable and \(x^*\) its global minimizer, then \(f'(x^*) = 0\). If \(f(x)\) is further convex, then a global minimizer of \(f(x)\) coincides with a zero of \(f'(x)\).

Thus, we can find a solution to the optimization problem by solving \(f'(x) = 0\).
Finding an inexact zero of $f$

Solve $f(x) = 0$.

What finding a zero means? Mathematically only analytical method can find it. For instance, line, quadratic, cubic polynomial, and some simple functions such as $\sin(x)$. However, for polynomials whose order is higher than 4, it is highly difficult to get theoretically solution.

We need computationally practical iterative algorithms to find zeros. (In general it is unrealistic to find an exact zero.)

Given a small tolerance $\delta$, we aim at an algorithm providing an interval $[a, b]$ such that:

$$f(a)f(b) < 0 \text{ and } |a - b| \leq \delta$$

A zero is said to be bracketed in an interval if $f$ changes sign in the interval. An interval containing $x^*$ is called an interval of uncertainty.
Bisection method
(systematic reduction of the interval of uncertainty via function comparisons)

Input \((a,b)\) such that \(f(a)f(b) < 0\).

Evaluate \(f\) at the midpoint \((a+b)/2\) and test its sign:

1. if \(f((a+b)/2)f(b) = 0\), the midpoint is zero, terminate;
2. if \(f((a+b)/2)f(b) < 0\), then set \(a := \frac{a+b}{2}\);
3. if \(f(a)f((a+b)/2) < 0\), then set \(b := \frac{a+b}{2}\);

Repeat the above procedure until \(|a-b| \leq \delta\).

Total number of evaluations of \(f\) is about \(\log_2 \frac{b-a}{\delta}\). Setting \(b^+ - a^+ = \frac{1}{2}(b - a)\), we have linear convergence

Direct algorithm not taking account of the relative magnitudes of the values of \(f\) at various points. If \(f\) is sufficiently smooth, or well behaved, it is possible to use the derivatives of \(f\) to improve the performance of the line search.
Newton’s method

Approximate $f$ by a new function called $\hat{f}$. A good candidate is the tangent line, $\hat{f} = f(x_k) + f'(x_k)(x - x_k)$. Hence:

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}.$$

Example: $f(x) = x^2 - a$ with $a \geq 0$, Newton iterate

$$x_{k+1} = \frac{1}{2} \left( x_k + \frac{a}{x_k} \right).$$

If $a > 0$, then globally convergent and locally quadratic convergent. Why?

If $a = 0$, global linear convergent. Why?

Numerical difficulties occur when $f'(x_k)$ is very small or zero.

Newton’s method converges only locally and $f$ is sufficiently smooth.
When $f'$ is expensive, cumbersome to compute, we use another straight line that passes through the values of $f$ at the most recent iterates; in essence, the derivative $f'(x_k)$ in Newton method is replaced by the finite-difference approximation $(f_k - f_{k-1})/(x_k - x_{k-1})$, where $f_k = f(x_k)$.

Thus we get

$$x_{k+1} = x_k - \frac{x_k - x_{k-1}}{f_k - f_{k-1}} f_k.$$

If $f'(x^*) \neq 0$ and $x_0, x_1$ is sufficiently close to $x^*$, the secant method converges superlinearly with a rate $r = 1.6180$.

The method of false position: A modification of secant method. Using $x_k, x_{k-1}$, we obtain $x_{k+1}$, we can replace either $x_k$ or $x_{k-1}$ by $x_{k+1}$, depending on which function value agrees in sign with $f_{k+1}$. Improve the global convergence, but might be very slow.
The best methods available for zero-finding are the so-called safeguarded procedures. These algorithms are combinations of bisection and linear interpolation methods.

Assume that an interval of uncertainty \([a, b]\) and two 'best' points are known. Using linear interpolation, one gets a new point \(u\). We use certain requirements to keep the point \(u\) is 'good' in some sense.
A univariate convex optimization problem can be solved via finding the zero of $f'(x)$. However, methods working on the original optimization problem is more efficient since we can further use information from the objective.

A univariate function is *unimodal* in $[a,b]$ if there exists a unique $x^* \in [a,b]$ such that, given any $x_1, x_2 \in [a,b]$ for which $x_1 < x_2$:

- if $x_2 < x^*$ then $f(x_1) > f(x_2)$;
- if $x_1 > x^*$ then $f(x_1) < f(x_2)$.

Reduce the interval of uncertainty for a unimodal function.

If $f(x_1) < f(x_2)$, then reduce the interval to $[a, x_2]$, replace $b$ by $x_2$.
If $f(x_1) > f(x_2)$, then reduce the interval to $[x_1, b]$, replace $a$ by $x_1$.
If $f(x_1) = f(x_2)$, reduce the interval to $[x_1, x_2]$. 

Univariate minimization
Golden section search

We assume that the starting interval is \([a, b] = [0, 1]\). Choose \(x_1 < x_2 \in [0, 1]\); the reduced interval will be \([0, x_2]\) or \([x_1, 1]\), this implies that only one point need to be added at the next iteration. Choose \(x_2 = 1 - x_1\), otherwise the reducing rate of the interval varies.

Let \(x_1 = 1 - \tau, x_2 = \tau\), the decreasing ratio is \(\tau\) at the first iteration. Assume that \([0, \tau]\) is the new interval containing the minimizer. Since \(x_1 = 1 - \tau\) is a point in the new interval, the decreasing rate for next interval should be \((1 - \tau)/\tau\).

This should equal to the first reducing rate \(\tau\). Thus we get the following equation

\[
\tau^2 + \tau - 1 = 0,
\]

The unique solution of the above equation is

\[
\tau = \frac{2}{1 + \sqrt{5}} \approx 0.618.
\]

This is the gold section search.

Question: Why bisection is not good for optimization?
Polynomial interpolation

Approximate \( f(x) \) by a simple function whose minimum can be easily evaluated, saying a quadratic function

\[
\hat{f} = \frac{1}{2}ax^2 + bx + c, \quad \text{where } a > 0.
\]

Linear approximation is not good in such sense. Taylor-series expansion can be used, i.e.,

\[
\hat{f} = f(x_k) + f'(x_k)(x - x_k) + \frac{1}{2}f''(x_k)(x - x_k)^2.
\]

Thus, we obtain

\[
x_{k+1} = x_k - \frac{f'(x_k)}{f''(x_k)}.
\]

We can replace \( f''(x_k) \) in the above formulae by

\[
\frac{f'(x_k) - f'(x_{k-1})}{x_k - x_{k-1}}.
\]

This is the secant method, one has

\[
x_{k+1} = x_k - \frac{f'(x_k)(x_k - x_{k-1})}{f'(x_k) - f'(x_{k-1})} = \frac{f'(x_k)x_{k-1} - x_kf'(x_{k-1})}{f'(x_k) - f'(x_{k-1})}.
\]

Again, safeguard method is the best choice.
Goldstein–Armijo principle

Computing the step length

$$\min \phi(\alpha) = f(x_k + \alpha d_k), \alpha > 0,$$

where \( f'(x_k) d_k < 0 \).

The new point should decrease \( f \) 'sufficiently'.

The Goldstein-Armijo principle:

$$0 < -\mu_1 \alpha_k f'(x_k) d_k \leq f_k - f_{k+1} \leq -\mu_2 \alpha_k f'(x_k) d_k,$$

where \( 0 < \mu_1 \leq \mu_2 < 1 \).

The upper and lower bounds in the above principle ensure \( \alpha_k \) 'reasonable'.

Choose \( \alpha_0 \) and \( 0 < \rho < 1 \), \( \alpha_{k+1} = \rho \alpha_k \).

For enough large \( k \), \( \alpha_k \) satisfies the Goldstein-Armijo condition. However \( \alpha_k \) might be too small.
Goldstein–Armijo principle continued

The choice $\alpha_0$ is very important. For Newton method, to get quadratic convergence, one should try $\alpha_0 = 1$ first. However, this might requires a lot of steps to get a step length satisfying Goldstein-Armijo condition if the Newton direction is not 'good'.

One should adjust $\alpha_0$ according to the present iterate.

There are some variants of Goldstein-Armijo condition, such as Wolfe condition or strong Wolfe condition.