CS 450: Numerical Analysis
Numerical Optimization

University of Illinois at Urbana-Champaign

---

These slides have been drafted by Edgar Solomonik as lecture templates and supplementary material for the book “Scientific Computing: An Introductory Survey” by Michael T. Heath (slides).
Numerical Optimization

- Our focus will be on continuous rather than combinatorial optimization:

\[
\min_x f(x) \quad \text{subject to} \quad g(x) = 0 \quad \text{and} \quad h(x) \leq 0
\]

- We consider linear, quadratic, and general nonlinear optimization problems:
Local Minima and Convexity

- Without knowledge of the analytical form of the function, numerical optimization methods at best achieve convergence to a *local* rather than *global* minimum:

- A set is *convex* if it includes all points on any line, while a function is (strictly) convex if its (unique) local minimum is always a global minimum:
Existence of Local Minima

- *Level sets* are all points for which $f$ has a given value, *sublevel sets* are all points for which the value of $f$ is less than a given value:

- If there exists a closed and bounded sublevel set in the domain of feasible points, then $f$ has a global minimum in that set:
Optimality Conditions

- If $x$ is an interior point in the feasible domain and is a local minima,

\[ \nabla f(x) = \left[ \frac{df}{dx_1}(x) \cdots \frac{df}{dx_n}(x) \right]^T = 0 : \]

- **Critical points** $x$ satisfy $\nabla f(x) = 0$ and can be minima, maxima, or saddle points:
Hessian Matrix

- To ascertain whether a critical point $x$, for which $\nabla f(x) = 0$, is a local minima, consider the *Hessian matrix*:

- If $x^*$ is a minima of $f$, then $H_f(x^*)$ is positive semi-definite:
Optimality on Feasible Region Border

- Given an equality constraint $g(x) = 0$, it is no longer necessarily the case that $\nabla f(x^*) = 0$. Instead, it may be that directions in which the gradient decreases lead to points outside the feasible region:

$$\exists \lambda \in \mathbb{R}^n, \quad -\nabla f(x^*) = J_g^T(x^*)\lambda$$

- Such *constrained minima* are critical points of the Lagrangian function $\mathcal{L}(x, \lambda) = f(x) + \lambda^T g(x)$, so they satisfy:

$$\nabla \mathcal{L}(x^*, \lambda) = \begin{bmatrix} \nabla f(x^*) + J_g^T(x^*)\lambda \\ \nabla g(x^*) \end{bmatrix} = 0$$
Sensitivity and Conditioning

- The condition number of solving a nonlinear equations is $1/f'(x^*)$, however for a minimizer $x^*$, we have $f'(x^*) = 0$, so conditioning of optimization is inherently bad:

- To analyze worst case error, consider how far we have to move from a root $x^*$ to perturb the function value by $\epsilon$: 
Golden Section Search

- Given bracket \([a, b]\) with a unique minimum (\(f\) is \textit{unimodal} on the interval), \textit{golden section search} considers points \(f(x_1), f(x_2)\), \(a < x_1 < x_2 < b\) and discards subinterval \([a, x_1]\) or \([x_2, b]\):

- Since one point remains in the interval, golden section search selects \(x_1\) and \(x_2\) so one of them can be effectively reused in the next iteration:

\[
\frac{(\sqrt{5} - 1)}{2} \quad \text{or} \quad (b-a) \left(\frac{\sqrt{5} - 1}{2}\right) \quad \text{is distance from} \quad a \quad \text{to} \quad x_2
\]
Newton’s Method for Optimization

- At each iteration, approximate function by quadratic and find minimum of quadratic function:

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

- The new approximate guess will be given by \( x_{k+1} - x_k = -\frac{f'(x_k)}{f''(x_k)} \):
Successive Parabolic Interpolation

- Interpolate $f$ with a quadratic function at each step and find its minima:

- The convergence rate of the resulting method is roughly 1.324
Safeguarded 1D Optimization

- Safeguarding can be done by bracketing via golden section search:

- Backtracking and step-size control:
General Multidimensional Optimization

- Direct search methods by simplex (Nelder-Mead):

- Steepest descent: find the minimizer in the direction of the negative gradient:

\[ x_{k+1} = x_k - \alpha_k \nabla f(x_k) \]

\[ \alpha_k = \min_{\alpha} f(x_k - \alpha \nabla f(x_k)) \]

via Golden Section or any 1D optimization scheme
Convergence of Steepest Descent

- Steepest descent converges linearly with a constant that can be arbitrarily close to 1:

\[ \text{in worst case } \lambda_k \geq 0, \text{ always } \lambda_k > 0 \]

how fast of changes guides converge

- Given quadratic optimization problem \( f(x) = \frac{1}{2}x^T Ax + c^T x \) where \( A \) is symmetric positive definite, the error \( e_k = x_k - x^* \) satisfies

\[
\frac{\|e_k\|}{\|e_k - 1\|_A} = \frac{\sigma_{\text{max}}(A) - \sigma_{\text{min}}(A)}{\sigma_{\text{max}}(A) + \sigma_{\text{min}}(A)} \frac{\kappa(A) - 1}{\kappa(A) + 1}
\]

\( e_k^T A e_k \), only norm if \( A \) is SPD

\( \nabla f(x) = A x + c \)

\( 0 = A x + c \)

\( h(x_k + s) = h(x_k) + \nabla h(x_k) s + \frac{1}{2} s^T H h(x_k) s + ... \)
Gradient Methods with Extrapolation

We can improve the constant in the linear rate of convergence of steepest descent by leveraging *extrapolation methods*, which consider two previous iterates (maintain *momentum* in the direction $x_k - x_{k-1}$):

$$x_{k+1} = f\left(x_k - \alpha_k \nabla f(x_k) + \beta_k (x_k - x_{k-1})\right).$$

The *heavy ball method*, which uses constant $\alpha_k = \alpha$ and $\beta_k = \beta$, achieves better convergence than steepest descent:
The conjugate gradient method is capable of making the optimal choice of $\alpha_k$ and $\beta_k$ at each iteration of an extrapolation method:

$$x_{k+1} = \underset{x}{\text{arg min}} \left( f(x_k - \alpha_k \nabla f(x_k) + \beta_k (x_k - x_{k-1})) \right)$$

parallel tangents implementation of the method proceeds as follows:

- Steepest descent from $x_k$ to generate $\tilde{x}_k$
- Minimize along the line from $x_{k-1}$ to $\tilde{x}_k$ to produce $x_{k+1}$

CG converges in $n$ steps
Conjugate Gradient as a Krylov Subspace Method

- Conjugate Gradient finds the minimizer of \( f(x) = \frac{1}{2} x^T A x + c^T x \) within the Krylov subspace of \( A \):

\[
\begin{align*}
\mathbf{x}^* &= \arg\min_{\mathbf{x} \in \mathcal{K}_k(A, c)} f(x) \\
&= \arg\min_{\mathbf{x}} \frac{1}{2} y^T Q_k^T A Q_k y + c^T Q_k y \\
&= Q_k y \\
y \in \mathbb{R}^k \\
&= \frac{1}{2} y^T T_k y + c^T Q_k y \\
&= \|c\|_2, y
\end{align*}
\]

\[
\nabla f(y) = T_k y + 11c\|\|_2 e, y = 0
\]

\[
y = -11c\|\|_2 T_k^{-1} e,
\]

\[
x^* = Q_k y = -11c\|\|_2 Q_k T_k^{-1} e,
\]
Newton’s Method

- Newton’s method in \( n \) dimensions is given by finding minima of \( n \)-dimensional quadratic approximation:

\[
\tilde{f}(x_k + \Delta x) \approx \tilde{f}(\Delta x) = f(x_k) + \Delta x^T \nabla f(x_k) + \frac{1}{2} \Delta x^T H_f(x_k) \Delta x
\]

\[
\nabla f(s) = 0 : 0 = \nabla f(x_k) + H_f(x_k) \Delta x
\]

\[
\Delta x = -H_f^{-1}(x_k) \nabla f(x_k)
\]

\[
x_{k+1} = x_k + \Delta x
\]

\[
\nabla f(x) = 0 \quad \text{equivalent Newton quadratic convergence}
\]
Quasi-Newton Methods

- **Quasi-Newton** methods compute approximations to the Hessian at each step:

- The **BFGS** method is a secant update method, similar to Broyden’s method:
Nonlinear Least Squares

- An important special case of multidimensional optimization is \textit{nonlinear least squares}, the problem of fitting a nonlinear function $f_x(t)$ so that $f_x(t_i) \approx y_i$.

- We can cast nonlinear least squares as an optimization problem and solve it by Newton’s method:
Gauss-Newton Method

- The Hessian for nonlinear least squares problems has the form:

- The *Gauss-Newton* method is Newton iteration with an approximate Hessian:
Levenberg-Marquardt Method

- The *Levenberg-Marquardt* modifies the Gauss-Newton method to use Tykhonov regularization:

  - The scalar $\mu$ controls the step size through the least squares problem:
We now return to the general case of *constrained* optimization problems:

\[
\min_x f(x) \quad \text{subject to} \quad g(x) = 0 \quad \text{and} \quad h(x) \leq 0
\]

Generally, we will seek to reduce constrained optimization problems to a series of unconstrained optimization problems:
The Lagrangian function with constraints \( g(x) = 0 \) and \( h(x) \leq 0 \) is

The Lagrangian dual problem is an unconstrained optimization problem:

\[
\max_{\lambda} q(\lambda), \quad q(\lambda) = \begin{cases} 
\min_x \mathcal{L}(x, \lambda) & \text{if } \lambda \geq 0 \\
-\infty & \text{otherwise}
\end{cases}
\]

The unconstrained optimality condition \( \nabla q(\lambda^*) = 0 \), implies
Sequential Quadratic Programming

- **Sequential quadratic programming (SQP)** reduces a nonlinear equality constrained problem to a sequence of constrained quadratic programs via a Taylor expansion of the Lagrangian function $L_f(x, \lambda) = f(x) + \lambda^T g(x)$:

- SQP ignores the constant term $L_f(x_k, \lambda_k)$ and minimizes $s$ while treating $\delta$ as a Lagrange multiplier:
Sequential Quadratic Programming

- From a different viewpoint, sequential quadratic programming corresponds to using Newton’s method to solve the nonlinear equations,

\[
\nabla \mathcal{L}(x, \lambda) = \begin{bmatrix} \nabla f(x) + J^T g(x) \lambda \\ g(x) \end{bmatrix} = 0
\]
Active Set Methods

- To use SQP for an inequality constrained optimization problem, consider at each iteration an *active set* of constraints:

- The Karush-Kuhn-Tucker (KKT) optimality conditions given the generalized Lagrangian function $\mathcal{L}(x, \mu, \nu) = f(x) + \mu^T g(x) + \nu^T h(x)$ are

\[
\begin{align*}
\nabla_x \mathcal{L}(x, \lambda) &= 0 \\
g(x) &= 0 \\
h(x) &\leq 0 \\
\nu &\geq 0 \\
\nu^T h(x) &= 0
\end{align*}
\]

at an optimal point, we must have that for either the $i$th inequality constraint is active, so $h_i(x) = 0$ or it is inactive, but its Lagrange multiplier $\nu_i = 0$. 
We can reduce constrained optimization problems to unconstrained ones by modifying the objective function. *Penalty* functions are effective for equality constraints $g(x) = 0$:

The augmented Lagrangian function provides a more numerically robust approach:
Barrier Functions

- A drawback of penalty function methods is that they can produce infeasible approximate solutions, which is problematic if the objective function is only defined in the feasible region:

- Barrier functions provide an effective way (*interior point methods*) of working with inequality constraints $h(x) \leq 0$: 