CS 450: Numerical Analysis\textsuperscript{1}  
Numerical Optimization  

University of Illinois at Urbana-Champaign

\textsuperscript{1}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: $f, g, h$ are affine functions

Quadratic $\Rightarrow f$ is quadratic, $g, h$ are linear

$H_f(x) = H_f(x') \quad \frac{1}{2} x^T A x - b^T x$
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:

$$\mathcal{L}(2) = \{ x : f(x) = 2 \}$$

$$\mathcal{S}(2) = \{ x : f(x) \leq 2 \}$$

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

Need to find $z$ s.t. $\mathcal{S}(z)$ has finite size and includes its own boundary.
Optimality Conditions

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

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

\[
\text{If } \frac{df}{dx_i}(x) < 0 \quad \text{then} \quad x + \delta x \rightarrow f(x + \delta x) < f(x)
\]

\[
\text{If } \frac{df}{dx_i}(x) > 0 \quad x - \delta x
\]

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

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

$$H_f(x) = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \cdots & \frac{\partial^2 f}{\partial x_i \partial x_j} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_j x_1} & \cdots & \frac{\partial^2 f}{\partial x_j \partial x_n} \end{bmatrix}$$

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

$$f(x^*) = f(x^*) + \nabla f(x^*)(x - x^*) + \frac{1}{2} (x - x^*)^T H_f(x^*) (x - x^*) \geq 0$$

$$x^T H_f(x^*) x \geq 0$$

If $\exists \gamma > 0$ such that $x^T H_f(x^*) x < 0$,
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:

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

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

$$\nabla L(x^*, \lambda) = \begin{bmatrix} \nabla f(x^*) + J_g(x^*)\lambda \\ g(x^*) \end{bmatrix} = 0$$
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 local minimum (\(f\) is unimodal on the interval), 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:
Newton’s Method for Optimization

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

- The new approximate guess will be given by \( x_{k+1} - x_k = -f'(x_k)/f''(x_k) \):
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:
Convergence of Steepest Descent

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

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

- The *heavy ball method*, which uses constant $\alpha_k = \alpha$ and $\beta_k = \beta$, achieves better convergence than steepest descent:
Conjugate Gradient Method

The conjugate gradient method is capable of making the optimal choice of $\alpha_k$ and $\beta_k$ at each iteration of an extrapolation method:

*Parallel tangents* implementation of the method proceeds as follows
Krylov Optimization

- Conjugate Gradient finds the minimizer of \( f(x) = \frac{1}{2} x^T A x + c^T x \) within the Krylov subspace of \( A \):
Newton’s Method

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

- The Levenberg-Marquardt method incorporates Tykhonov regularization into the linear least squares problems within the Gauss-Newton method.
Constrained Optimization Problems

- We now return to the general case of constrained optimization problems:

- Generally, we will seek to reduce constrained optimization problems to a series of unconstrained optimization problems:
  - *sequential quadratic programming*:
  - *penalty-based methods*:
  - *active set methods*:
Sequential Quadratic Programming

- **Sequential quadratic programming** (SQP) corresponds to using Newton’s method to solve the equality constrained optimality conditions, by finding critical points of the Lagrangian function $\mathcal{L}(x, \lambda) = f(x) + \lambda^T g(x)$,

At each iteration, SQP computes

\[
\begin{bmatrix}
    x_{k+1} \\
    \lambda_{k+1}
\end{bmatrix} = \begin{bmatrix}
    x_k \\
    \lambda_k
\end{bmatrix} + \begin{bmatrix}
    s_k \\
    \delta_k
\end{bmatrix}
\]

by solving
Inequality Constrained Optimality Conditions

- The *Karush-Kuhn-Tucker (KKT)* conditions hold for local minima of a problem with equality and inequality constraints, the key conditions are:

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

- Alternatively, 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

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