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:
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:

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

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

\[
\nabla \mathcal{L}(x^*, \lambda) = \begin{bmatrix}
\nabla f(x^*) + J_g(x^*)^T \lambda \\
g(x^*)
\end{bmatrix} = 0
\]
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 \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:
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:
Convergence of Steepest Descent

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

Given quadratic optimization problem $f(x) = \frac{1}{2}x^TAx + c^Tx$ where $A$ is symmetric positive definite, the error $e_k = x_k - x^*$ 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
Nonlinear Conjugate Gradient

Various formulations of conjugate gradient are possible for nonlinear objective functions, which differ in how they compute $\beta$ below.

Fletcher-Reeves is among the most common, computes the following at each iteration:

1. Perform 1D minimization for $\alpha$ in $f(x_k + \alpha s_k)$
2. $x_{k+1} = x_k + \alpha s_k$
3. Compute gradient $g_{k+1} = \nabla f(x_{k+1})$
4. Compute $\beta = g_{k+1}^T g_{k+1} / (g_k^T g_{k+1})$
5. $s_{k+1} = -g_{k+1} + \beta s_k$
Conjugate Gradient for Quadratic Optimization

Conjugate gradient is an optimal iterative method for quadratic optimization,
\[ f(x) = \frac{1}{2} x^T A x - b^T x \]

For such problems, it can be expressed in an efficient and succinct form, computing at each iteration

1. \( \alpha = r_k^T r_k / s_k^T A s_k \)
2. \( x_{k+1} = x_k + \alpha s_k \)
3. Compute gradient \( r_{k+1} = r_k - \alpha_k A s_k \)
4. Compute \( \beta = r_{k+1}^T r_{k+1} / (r_k^T r_k) \)
5. \( s_{k+1} = r_{k+1} + \beta s_k \)

Note that for quadratic optimization, the negative gradient \(-g\) corresponds to the residual \( r = b - A x \)
Krylov Optimization

- Conjugate Gradient finds the minimizer of \( f(x) = \frac{1}{2} x^T A x - b^T x \) 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 $h(x) \leq 0$: 