Cheatsheet!!

- Algorithms:
  - Steepest descent: $x_{i+1} = x_i - \nabla f(x_i) \cdot \frac{\nabla f(x_i)^\top \nabla f(x_i)}{\nabla f(x_i)^\top H(x_i) \nabla f(x_i)}$.
  - Newton's method: $x_{i+1} = x_i - H(x_i)^{-1} \nabla f(x_i)$.
  - Modified Newton: Modify the Hessian matrix such that $\nabla f^\top \cdot (-H^{-1} \cdot \nabla f) < 0$ (descent direction) & make H well-conditioned. The original Hessian matrix may have negative eigenvalues, which will result in ascent direction.

    One can perform eigen decomposition and modified the eigenvalues directly, however, computing eigenvalues is costly.

    Another method is called shift modification, $\hat{H} = H + \mu I$, this will add μ to each of the eigenvalues, however, it's hard to determine μ.

    We then turn to the so called $LDL^\top$ decomposition, which decompose H into $LDL^\top$, L is a lower triangular matrix with 1 as diagonal elements, and D is a diagonal matrix.

    We only need to modify the D in order to modify the original matrix (L remains the same), also, calculating the inverse of the $LDL^\top$ decomposition is quite efficent since calculating the inverse of triangular matrix and diagonal matrix can be very efficient.
  - Conjugate Gradient Method: reduce the number of steps for each dimension.
    * Given x_0. Let $k = 0$, $r_0 = g - H_0 x_0$, and $d_0 = r_0$.
      Repeat until $\|r_k\| \leq \epsilon$
      - $\alpha_k = \frac{d_k^\top r_k}{d_k^\top H_k d_k}$
      - $x_{k+1} = x_k + \alpha_k d_k$
      - $r_{k+1} = r_k - \alpha_k H_k d_k$
      - $\beta_k = \frac{r_{k+1}^\top r_{k+1}}{r_k^\top r_k}$ (Fletcher-Reaves)
      - $d_{k+1} = r_{k+1} + \beta_k d_k$
      - $k = k + 1$
      - Evaluate H_k
    * It turns out CG is a fairly good approach to solve a linear system $Ax = b$.
  - Quasi-Newton:
    * We can calcutate the vector $H \cdot d = \lim_{h \to 0} \frac{g(x + h \cdot d) - g(x)}{h} = \begin{bmatrix} \nabla g_1 \\ \nabla g_2 \\ \vdots \\ \nabla g_k \end{bmatrix} \cdot d$.

      This is the so called Hessian-free calculation.
    * However, to use Newton's method, we need to calculate $H^{-1} \cdot d$, not $H \cdot d$.
    * To solve this, use Quasi-Newton's method which assumes that there's no significant differences between two consecutive Hessian matrices.
    * Let B_k denotes the approximation of H_k.
    * The secant condition is given by $H(x_{k+1}) \cdot (x_{k+1} - x_k) = g(x_{k+1}) - g(x_k)$, which is denoted as $B_{k+1} \cdot d_k = y_k$.
    * And the Sherman-Morrism-Woodburg formula gives:

$$A = B + ab^\top \implies A^{-1} = B^{-1} - \frac{B^{-1} a b^\top B^{-1}}{1 + b^\top B^{-1} a}$$

    * With SR1 update, we have:

$$B_{k+1} = B_k + \frac{(y_k - B_k d_k)(y_k - B_k d_k)^\top}{(y_k - B_k d_k)^\top d_k}$$

(originally, $B_{k+1} = B_k + \sigma_k u_k u_k^\top$ with $u_k u_k^\top$ a symmetric rank-one matrix).
Also,

$$B_{k+1}^{-1} = B_k^{-1} + \frac{(d_k - B_k^{-1} y_k)(d_k - B_k^{-1} y_k)^\top}{(d_k - B_k^{-1} y_k)^\top y_k}$$

* With BFGS update, we have:

$$B_{k+1}^{-1} = (I - \rho_k d_k y_k^\top) B_k^{-1} (I - \rho_k y_k d_k^\top) + \rho_k d_k d_k^\top$$

where $\rho_k = \frac{1}{y_k^\top d_k}$.
Note that BFGS is desinged to satisfy curvature condition: $d_k^\top y_k > 0$, or in other form: $d_k^\top B_k d_k > 0$.
BFGS is a **rank 2** update: $B_{k+1} = B_k + uu^\top + vv^\top$.

* Sometime we need to deal with the situations that the denominator is zero, in such case, do not update B_i (i.e. $B_{i+1} = B_i$).

- Convergence rate test: $\lim\limits_{i \to \infty} \frac{\|x_i - x^*\|}{\|x_{i-1} - x^*\|} = c = \begin{cases} 0 & superlinear \\ 0.5 & linear \\ 1 & sublinear \end{cases}$
- Quadratic convergence: $\lim\limits_{i \to \infty} \frac{\|x_i - x^*\|}{\|x_{i-1} - x^*\|^2} = c,\ c > 0$.
- Convergence of CG: For any $x \in R^n$, if A has m distinct eigenvalues, CG will terminate at the solution at most m iterations. Moreover, if A has its eigenvalues $\lambda_1 \leq \lambda_2 \leq ... \leq \lambda_n$ then $||x_{i+1} - x^*||_A^2 \leq (\frac{\lambda_{n-i} - \lambda_1}{\lambda_{n-i} + \lambda_1})^2 ||x_i - x^*||_A^2$ (superlinear convergence).
- Condition number $\kappa(A)$ of a matrix A is defined by $\kappa(A) = \|A\| \cdot \left\|A^{-1}\right\|$.
  - Consider a linear system $Ax = b$, with x as its solution.
    * An error term E is introduced if we use some numeric method to solve the system and results in an approximate solution y.

$$\begin{aligned} Ax &= (A + E)y \\ &= Ay + Ey \end{aligned}$$

$$\begin{aligned} A(x - y) &= Ey \\ x - y &= A^{-1}Ey \\ \|x - y\| &= \left\|A^{-1}Ey\right\| \leq \left\|A^{-1}\right\| \cdot \|E\| \cdot \|y\| \\ \frac{\|x-y\|}{\|y\|} &\leq \left\|A^{-1}\right\| \cdot \|E\| = \frac{\left\|A^{-1}\right\| \cdot \|E\|}{\|A\|} \cdot \|A\| \\ \frac{\|x-y\|}{\|y\|} &\leq (\left\|A^{-1}\right\| \cdot \|A\|) \cdot \frac{\|E\|}{\|A\|} \end{aligned}$$

    * $\frac{\|E\|}{\|A\|}$ is the relative error introduced by using numeric methods, and $\left\|A^{-1}\right\| \cdot \|A\|$ is the condition number which determines the condition of the linear system (or more specifically, the condition of A).
  - If $\kappa(A)$ is small, the matrix A is called well-conditioned (the linear system $Ax = b$ can be solved stably).
  - If $\kappa(A)$ is large, the matrix A is called ill-conditioned.
  - If A is symmetric, $\kappa(A) = \left|\frac{\lambda_{max}}{\lambda_{min}}\right|$.
- Backtracking line search: attempt to find step length given the function of step length $\phi(\alpha) = f(x_k + \alpha p_k)$:
  - Wolfe conditions:
    * Sufficeint decrease condition: $\phi(\alpha) \leq \phi(0) + c_1 \alpha \phi'(0)$.
    * Curvature condition: $\phi'(\alpha) \geq c_2 \phi'(0)$.
  - Goldstien condition:
    * $f(x_k) + (1 - c)\alpha g_k^\top d_k \leq f(x_k + \alpha d_k) \leq f(x_k) + c\alpha g_k^\top d_k$
  - Theory: If d_i is a descent directionm and α_i satisfies Wolfe conditions, f is bounded below ($|x^*|$ is not equal to ∞) and is a C^1 function, also ∇f is Lipschitz continuous, then $\sum\limits_{i \geq 0} cos^2 \theta_i ||\nabla f_i||^2 < \infty$ (θ_i is the angle between g_i and d_i).