Useful Formulae and Algorithms

1. **The Fibonacci numbers** satisfy the relation $F_n = F_{n-1} + F_{n-2}$ with $F_0 = 1$ and $F_1 = 1$.

2. **Fibonacci search algorithm**
   To minimise a unimodal function f over $[a, b]$ to within tolerance ϵ, where F_n is the nth Fibonacci number.

   (a) Find the smallest value of n such that $(b-a)/F_n < 2\epsilon$.

   (b) Set

   $$\begin{aligned} k &= n \\ p &= b - \frac{F_{k-1}}{F_k}(b-a) \\ q &= a + \frac{F_{k-1}}{F_k}(b-a) \end{aligned}$$

   Calculate $f(p)$ and $f(q)$.

   (c) Set $k = k - 1$.
   If $f(p) \leq f(q)$, then set

   $$\begin{aligned} b &= q \\ q &= p \\ p &= b - \frac{F_{k-1}}{F_k}(b-a) \end{aligned}$$

   Calculate $f(p)$.
   If $f(p) > f(q)$, then set

   $$\begin{aligned} a &= p \\ p &= q \\ q &= a + \frac{F_{k-1}}{F_k}(b-a) \end{aligned}$$

   Calculate $f(q)$.
   Repeat until $k = 3$.

   (d) If $f(p) \leq f(q)$, then set

   $$\begin{aligned} b &= q \\ q &= p \\ p &= b - 2\epsilon \end{aligned}$$

   Calculate $f(p)$.
   If $f(p) > f(q)$, then set

   $$\begin{aligned} a &= p \\ p &= q \\ q &= a + 2\epsilon \end{aligned}$$

   Calculate $f(q)$.

(e) If $f(p) \leq f(q)$, then $b = q$.
If $f(p) > f(q)$, then $a = p$.
The final interval is $[a, b]$. This interval has length either 2ϵ or $\alpha < 2\epsilon$.

3. **Golden section search algorithm**
To minimise a unimodal function f over $[a, b]$ to within tolerance ϵ, where $\gamma_n = F_{n-1}/F_n$.

   (a) Set
$$\begin{aligned} k &= 1 \\ p &= b - \gamma(b - a) \\ q &= a + \gamma(b - a) \end{aligned}$$
   Calculate $f(p)$ and $f(q)$.

   (b) Set $k = k + 1$.
   If $f(p) \leq f(q)$, then set
$$\begin{aligned} b &= q \\ q &= p \\ p &= b - \gamma(b - a) \end{aligned}$$
   Calculate $f(p)$.
   If $f(p) > f(q)$, then set
$$\begin{aligned} a &= p \\ p &= q \\ q &= a + \gamma(b - a) \end{aligned}$$
   Calculate $f(q)$.
   Repeat until $(b - a) < 2\epsilon$.

4. **Algorithm for the method of false position**
For an increasing, continuous function g on $[a, b]$, to find a point x^* where $|g(x^*)| < \epsilon$.

   (a) Set
$$\begin{aligned} k &= 1 \\ p &= a + \frac{(b - a)g(a)}{g(a) - g(b)} \end{aligned}$$
   Calculate $g(p)$.

   (b) Set $k = k + 1$.
   If $g(p) < 0$, then set
$$\begin{aligned} a &= p \\ p &= a + \frac{(b - a)g(a)}{g(a) - g(b)} \end{aligned}$$
   Calculate $g(p)$.
   If $g(p) > 0$, then set
$$\begin{aligned} b &= p \\ p &= a + \frac{(b - a)g(a)}{g(a) - g(b)} \end{aligned}$$

Calculate $g(p)$.
Repeat until $|g(p)| < \epsilon$.

5. **Algorithm for Newton's Method**
For an increasing, continuous function g on $\Re$ and an initial starting point a, to find a point x^* where $|g(x^*)| < \epsilon$.

(a) Set

$$\begin{aligned} k &= 1 \\ \text{if } g'(a) &< \epsilon \text{ then signal and stop.} \\ \text{else } p &= a - \frac{g(a)}{g'(a)} \end{aligned}$$

(b) Set $k = k + 1$.

$$\begin{aligned} a &= p \\ \text{if } g'(a) &< \epsilon \text{ then signal and stop.} \\ \text{else } p &= a - \frac{g(a)}{g'(a)} \end{aligned}$$

Repeat until $|g(p)| < \epsilon$.

6. **Algorithm for finding an upper bound on the location of the minimum**
For a continuous, unimodal function f on $[0, \infty)$, to find a point b such that the minimum $x_{min} < b$.

(a) Choose some small initial increment value T.

$$\begin{aligned} k &= 1 \\ p &= 0 \\ q &= T \end{aligned}$$

Calculate $f(p)$ and $f(q)$.

(b) Set $k = k + 1$.

$$\begin{aligned} p &= q \\ q &= p + 2^{k-1}T \end{aligned}$$

Calculate $f(q)$.
Repeat until $f(p) \leq f(q)$.

7. For a unimodal, continuous and differentiable function f on $[0, \infty)$, we say that the step size t satisfies the **Armijo-Goldstein condition** with weight σ if $f(t) \leq f(0) + t\sigma f'(0)$ where $\sigma \in [0, 1)$.

8. For a unimodal, continuous and differentiable function f on $[0, \infty)$, we say that the step size t satisfies the **Wolfe condition** with weight μ if $f'(t) \geq \mu f'(0)$ where $\mu \in [\sigma, 1)$.

9. **Algorithm to Find a Step Size that Satisfies The Armijo-Goldstein and Wolfe Conditions**
For a differentiable, unimodal function f on $[0, \infty)$. Input an initial step size T, a number $\sigma \in (0, 1)$ and a number $\mu \in [\sigma, 1)$.

(a) Set

$$\begin{aligned} t_{lo} &= 0 \\ t_{hi} &= \infty \\ t &= T \end{aligned}$$

(b) If $f(t) > f(0) + t\sigma f'(0)$, then

$$\begin{aligned} t_{hi} &= t \\ t &= 1/2(t_{lo} + t). \end{aligned}$$

Else if $f'(t) < \mu f'(0)$, then

$$\begin{aligned} t_{lo} &= t \\ t &= \begin{cases} 1/2(t_{lo} + t_{hi}) & \text{if } t_{hi} < \infty \\ \\ 2t & \text{otherwise} \end{cases} \end{aligned}$$

Repeat until $f(t) \leq f(0) + t\sigma f'(0)$ and $f'(t) \geq \mu f'(0)$.

10. Given $x \in \Re^n$, a vector $d \in \Re^n$ is a **descent direction** for f at x if

$$\langle \nabla f(x), d \rangle \;<\; 0.$$

11. **The steepest descent algorithm**
To minimise a unimodal function $f : \Re^N \to \Re$ to within tolerance ϵ.

   (a) Select $x^0 \in \Re^N$.
   Set $k = 0$.

   (b) Calculate $d^k = -\nabla f(x^k)$.
   If $||d^k|| < \epsilon$ then stop.

   (c) Select step length t_k either
      - by solving the single-variable minimisation problem: $\min q(t) = f(x^k + td^k)$.
      - by using our procedure for finding a step length that satisfies the Armijo-Goldstein and Wolff conditions.

   (d) Set $k = k + 1$.
   Set $x^{k+1} = x^k + t_k d^k$.
   Return to step 2.

12. **Newton's method**
To minimise a unimodal function $f : \Re^N \to \Re$ to within tolerance ϵ.

   (a) Select $x^0 \in \Re^N$.
   Set $k = 0$.

   (b) If $||\nabla f(x^k)|| < \epsilon$ then stop.
   If $\nabla^2 f(x^k)$ is positive definite, then
   Set $d^k = -\nabla^2 f(x^k)^{-1} \nabla f(x^k)$.
   Else, set $d^k = -\nabla f(x^k)$

   (c) Select step length t_k either

- by solving the single-variable minimisation problem: $\min q(t) = f(x^k + td^k)$.
- by using our procedure for finding a step length that satisfies the Armijo-Goldstein and Wolff conditions.

(d) Set $k = k + 1$.
Set $x^{k+1} = x^k + t_k d^k$.
Return to step 2.

13. **The BGFS method**
To minimise a unimodal function $f : \Re^N \to \Re$ to within tolerance ϵ.

(a) Select $x^0 \in \Re^N$.
Set $k = 0$. Set $H_0 \in \Re^{n \times n}$ to be a symmetric positive definite matrix (for example $H_0 = I$).

(b) If $||\nabla f(x^k)|| < \epsilon$ then stop.
Set $d^k = -H_k \nabla f(x^k)$.

(c) Select step length t_k either

- by solving the single-variable minimisation problem: $\min q(t) = f(x^k + td^k)$.
- by using our procedure for finding a step length that satisfies the Armijo-Goldstein and Wolff conditions.

(d) Set $x^{k+1} = x^k + t_k d^k$.
Update H_k as follows:

$$\begin{aligned}
s^k &= x^{k+1} - x^k \\
g^k &= \nabla f(x^{k+1}) - \nabla f(x^k) \\
r^k &= H_k g^k / \langle s^k, g^k \rangle \\
H_{k+1} &= H_k + \frac{1 + \langle r^k, g^k \rangle}{\langle s^k, g^k \rangle}\, s^k (s^k)^T - [s^k (r^k)^T \, + \, r^k (s^k)^T]
\end{aligned}$$

Set $k = k + 1$.
Return to step 2.

14. **The Lagrangian function** for (NLP) is

$$L(x, \lambda, \eta) \ := \ f(x) + \sum_{i=1}^{p} \lambda_i g_i(x) + \sum_{j=1}^{q} \eta_j h_j(x) \ = \ f(x) + \langle \lambda, g(x) \rangle + \langle \eta, h(x) \rangle,$$

15. **KKT Conditions** Let f, g and h be C^1 functions, and assume that one of the constraint qualifications (discussed below) on h and g holds at x^*. If x^* is a local minimum of (NLP) then there exist $\lambda^* \in \mathbb{R}^p$ and $\eta^* \in \mathbb{R}^q$ such that

KKTa. $0 = \nabla_x L(x^*, \lambda^*, \eta^*)$.

KKTb. i. $g(x^*) \leq 0$,

ii. $\lambda^* \geq 0$, and

iii. for each i, $\lambda_i^* g_i(x^*) = 0$.

KKTc. $h(x^*) = 0$.

16. **The Mangasarian-Fromovitz constraint qualification**. The equality constraint gradients $\nabla h_j(x^*)$ are linearly independent, and there exists $d \in \mathbb{R}^n$ such that

   (a) $\nabla h(x^*)^T d = 0$ and,

   (b) for all active inequality constraints, $\nabla g_i(x^*)^T d < 0$.

Text

17. **The critical cone** at (x^*, λ^*) is the set

$$\begin{aligned} \mathcal{C}(x^*, \lambda^*) \ := \ \{d \in \mathbb{R}^n \quad : \quad & \langle \nabla g_i(x^*), d \rangle \leq 0 \text{ if } g_i \text{ is active and } \lambda_i^* = 0, \\ & \langle \nabla g_i(x^*), d \rangle = 0 \text{ if } \lambda_i^* > 0, \\ & \langle \nabla h_j(x^*), d \rangle = 0, \forall j\}. \end{aligned}$$

18. **The l_2 penalty function** for (NLP) (for $\alpha > 0$) is

$$P_\alpha(x) \ = \ f(x) + \frac{\alpha}{2} \left(\sum_i [g_i(x)_+]^2 \ + \ \sum_j h_j(x)^2 \right)$$

where

$$g_i(x)_+ \ := \ \max\{g_i(x), 0\} \ = \ \begin{cases} g_i(x) & \text{if } g_i(x) > 0 \text{ (infeasible)}, \\ 0 & \text{if } g_i(x) \leq 0 \text{ (feasible)}. \end{cases}$$

19. **The log barrier penalty function** for (NLP) (for $\alpha > 0$) is

$$P_\alpha(x) \ = \ f(x) \ - \ \frac{1}{\alpha} \sum_i \log(-g_i(x)) \ + \ \frac{\alpha}{2} \sum_j h_j(x)^2,$$

20. **The exact penalty function** for (NLP) (for $\alpha > 0$) is

$$P_\alpha(x) \ = \ f(x) + \alpha \left(\|g(x)_+\|_1 + \|h(x)\|_1 \right).$$

21. **The Lagrangian dual** of (NLP) is

$$\max_{\lambda \geq 0,\, \eta} \min_x L(x, \lambda, \eta).$$

22. **The Wolfe dual** of (NLP) is

$$\begin{array}{ll} \displaystyle\max_{x,\, \lambda,\, \eta} & L(x, \lambda, \eta) \\ \text{subject to} & \lambda \geq 0, \ \nabla_x L(x, \lambda, \eta) = 0. \end{array}$$