Discrete Event Systems

Summary Chapters 6 and 7

Yannick Merkli

1 Queueing

1.1 Continuous Time Markov Chain

Definition 6.1 (Continuous Time Markov Chain, CTMC): Let S be a finite or countably infinite set of states. A Continuous Time Markov Chain (CTMC) is a continuous time stochastic process $\{X_t : t \in \mathbb{R}_{\geq 0}\}\ with\ X_t \in S$ for all t that satisfies the continuous Markov property.

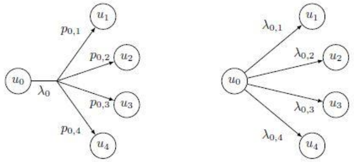
Definition 6.2 (Continuous Markov Property): A Markov chain satisfies the Markov property if the probability for the next state depends only on the current state, and not the history. Such a system is also called memoryless.

Definition 6.3 (Sojourn Time): The sojourn time T_i of state i is the time the process stays in state i.

Lemma 6.6. Let $Y_1, \dots, Y_k$ be k independent exponential random variables with corresponding parameters $\lambda_1, \dots, \lambda_k$. The random variable $Y = \min\{Y_1, \dots, Y_k\}$ is exponentially distributed with parameter $\lambda_1 + \dots + \lambda_k$.

Lemma 6.7. Let $Y_1, \dots, Y_k$ be k independent exponential random variables with corresponding parameters $\lambda_1, \dots, \lambda_k$. The probability

$$Pr[Y_1 = \min\{Y_1, \dots, Y_k\}] = \frac{\lambda}{\lambda_1 + \dots + \lambda_k}$$



$$\sum_{j \in S} p_{i,j} = 1, \quad \lambda_{i,j} = \lambda_i \cdot p_{i,j}, \quad \sum_{j \in S} \lambda_{i,j} = \lambda_i$$

Theorem 6.9. For all $i \in S$, the change in the state probability p_i is

$$\frac{d}{dt} q_i(t) = \sum_{j:j \neq i} q_j(t) \cdot \lambda_{j,i} - q_i(t) \cdot \lambda_i$$

Definition 6.10 (Stationary Distribution) For $t \to \infty$, π is a stationary distribution if for all $i \in S$,

$$\frac{d}{dt} q_i(t) = 0 = \sum_{j:j \neq i} \pi_j \cdot \lambda_{j,i} - \pi_i \cdot \lambda_i$$

- Remark: We're interested in probability distributions, therefore,
$$\sum_i \pi_i = 1, \quad and\ \pi_i \geq 0$$

Theorem 6.11 (Irreducible) A CTMC is **irreducible** if for all states I and j it holds that j is reachable from i. That is, if there exists some $t \geq 0$ such that $Pr[X_t = j | X_0 = i] > 0$

Theorem 6.12 For finite irreducible CTMCs the limits

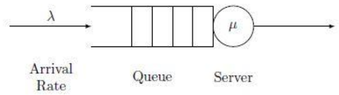
$$\pi_i := \lim_{t \to \infty} q_i(t)$$

Exists for all $i \in S$. Moreover, the entries in π are independent of q(0).

- Remark: CTMCs for which the stationary distribution exists are called **ergodic**. For finite chains this is the same as being irreducible.

1.2 Queues

Definition 6.13 A queueing system consists of a queue with one or more servers which process jobs. The queue acts as a buffer for jobs that arrived but cannot be processed yet, because the server is busy processing another job.



Definition 6.15 (Kendall's Notation). Let a and s be symbols describing the arrival and service rates, and let $m, n, j \in \mathbb{N}$. The **Kendall notation** for a queueing system Q is $a/s/m/n/j$. The symbols **a and s can be D, M, or G**, where:

- D means that the rate distribution is **degenerate**, i.e., of fixed length
- M means that the arrival/ service process is **memoryless**
- G means that the corresponding rate stems from a **generic** distribution

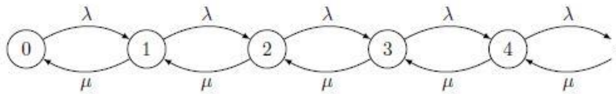
The parameter

- m is the number of **servers**
- n is the number of **places** in the system (in the queue and at servers)
- j determines the **external population** of jobs that may enter the system

1.2.1 Notations

- $\bar{N}$ the average number of jobs in the system
- $\bar{\lambda}$ the average arrival rate
- $\bar{T}$ the average response time of a job (waiting time + service time), i.e. the average time spent in the system
- $\bar{W}$ the average waiting time (time spent in the queue)
- $\bar{N}_Q$ the average number of jobs waiting in the queue

1.2.2 The *M/M/1* Queue



A CTMC modeling an M/M/1 system. In state 0 the system is empty. When the chain is in **state $i \geq 1$, then there are $i - 1$ jobs** in the queue, and one job is being served with rate μ. New jobs arrive with rate λ.

Theorem 6.17. An M/M/1 queueing system **has a stationary distribution if and only if**

$$\rho = \frac{\lambda}{\mu} < 1$$

ρ is called utilization.

In that case the stationary distribution is $\pi_k = \rho^k(1 - \rho)$.

- An M/M/1 queueing system is **stable** if $\rho = \lambda/\mu < 1$.
- The M/M/1 queue is also **ergodic** if and only if $\rho < 1$.
- The probability that the singel server is processing a job is
$$\boxed{1 - \pi_0 = \rho}$$
- The probability that no one is in the queue is
$$\boxed{\pi_0 = 1 - \rho}$$

Theorem 6.19. In expectation there are

$$N = \frac{\lambda}{\mu - \lambda} = \frac{\rho}{1 - \rho}$$

jobs in an M/M/1 system.

Remark: The variance of the number of jobs in the system is $\frac{\rho}{(1-\rho)^2}$

Using Little's law: In steady state the **average response** times is

$$\bar{T} = \frac{N}{\lambda} = \frac{1}{\mu - \lambda}$$

The **average waiting time of a job** is

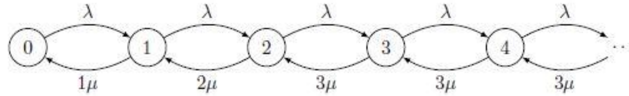
$$\bar{W} = \bar{T} - \frac{1}{\mu} = \frac{\rho}{\mu - \lambda}$$

The **average number of jobs in the queue** is

$$\bar{N}_Q = \bar{\lambda}\bar{W} = \frac{\rho^2}{1 - \rho}$$

1.2.3 The *M/M/m* Queue

What if there is a single queue for multiple servers(e.g. in a service hotline)?
In Kendall's notation, the number of servers is denoted by m.



Birth-Death process modeling an M=M=3 queueing system.

Remark: If there are less than m jobs, then the number of active servers is the number of jobs in the system. When m or more jobs are in the system all servers are active.

An M/M/m queueing system **has a stationary distribution if and only if** for the utilization

$$\rho = \frac{\lambda}{m\mu} < 1$$

Then the stationary distribution is

$$\pi_k = \begin{cases} \pi_0 \cdot \frac{(\rho m)^k}{k!}, & for\ 1 \leq k \leq m \\ \pi_0 \cdot \frac{\rho^k m^m}{m!}, & for\ k \geq m \end{cases}$$

and

$$\pi_0 = \frac{1}{\sum_{k=0}^{m-1} \frac{(\rho m)^k}{k!} + \frac{(\rho m)^m}{m!\,(1-\rho)}}$$

The probability that in the stationary distribution an arriving job has to wait in the queue is

$$P_Q = \sum_{k=m}^{\infty} \pi_k = \sum_{k=m}^{\infty} \frac{\pi_0 \rho^k m^m}{m!} = \frac{\pi_0 (\rho m)^m}{m!} \sum_{k=m}^{\infty} \rho^{k-m} = \frac{\pi_0 (\rho m)^m}{m!\,(1-\rho)}$$

Plugging in π_0 gives us the formula (known as **Erlang C Formula**)

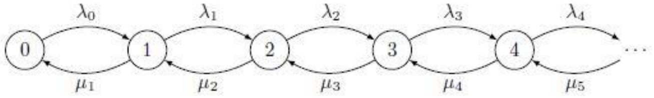
$$P_Q = \frac{(\rho m)^m / (m!\,(1-\rho))}{\sum_{k=0}^{m-1} \frac{(\rho m)^k}{k!} + \frac{(\rho m)^m}{m!\,(1-\rho)}}, \quad (for\ \rho < 1)$$

The **average number of jobs in the queue** is

$$\bar{N}_Q = P_Q \cdot \frac{\rho}{1 - \rho}$$

1.2.4 Birth-Death Processes

Our CTMC for the M/M/1 queueing system is a special case of a so-called Birth-Death Process.



We can compute the stationary distribution. We obtain

$$\pi_0 = \frac{1}{1 + \sum_{k\geq 1} \prod_{i=0}^{k-1} \frac{\lambda_i}{\mu_{i+1}}}$$

$$\pi_k = \pi_0 \cdot \prod_{i=0}^{k-1} \frac{\lambda_i}{\mu_{i+1}}$$

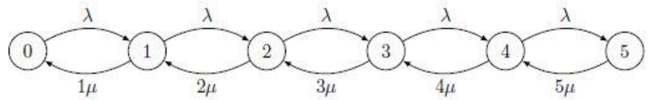
Also remember that

$$\sum_{i=0}^{n} \pi_i = 1$$

Sometimes we can find π_0 through this relation.

1.2.5 The *M/M/m/n* Queue

Often, the space in the queue is bounded, i.e., the system is M/M/m/n. Recall that n is the number of places in the system, so the maximum length of the queue is $n - m$.



The case $m = n$ is often used to model communication networks. Such a system can accommodate m simultaneous calls, and the duration of a call is distributed with $\exp(\mu)$. One can calculate that in this case

$$\pi_k = \pi_0 \cdot \left(\frac{\lambda}{\mu}\right)^k \frac{1}{k!}, 1 \leq k \leq m$$

Using $\sum_{k=0}^{m} \pi_k = 1$ yields

$$\pi_0 = \frac{1}{\sum_{k=0}^{m} \left(\frac{\lambda}{\mu}\right)^k \frac{1}{k!}}$$

The **blocking probability**, i.e. the probability that an arriving job is rejected, is

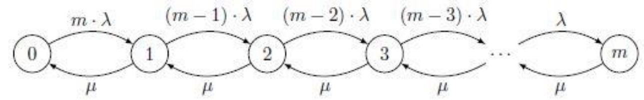
$$\pi_m = \frac{\left(\frac{\lambda}{\mu}\right)^m \frac{1}{m!}}{\sum_{k=0}^{m} \left(\frac{\lambda}{\mu}\right)^k \frac{1}{k!}}$$

This is the so-called **Erlang-B Formula**. It also holds for M/G/m/m systems where the service times are $\frac{1}{\mu}$ in expectation, regardless of their distribution.

1.2.6 The *M/M/n/m/m* Queue

Sometimes the assumption that the arrival rate is independent of the number of jobs in the system cannot be made. This cases can be modeled as an $M/M/n/m/j$ system.

Example: M/M/1/m/m queue



For M/M/1/m/m systems:

$$\pi_k = \pi_0 \cdot \prod_{i=0}^{k-1} \frac{\lambda(m-i)}{\mu}, \quad for\ 1 \leq k \leq m$$

$$\pi_0 = \frac{1}{\sum_{k=0}^{m} \left(\frac{\lambda}{\mu}\right)^k \cdot m^{\underline{k}}}$$

where $m^{\underline{k}} \coloneqq m(m-1)(m-2) \cdot \ldots \cdot (m-k+1)$

1.2.7 Little's law

The following random variables describe

- $\bar{N}$ the average number of jobs in the system
- $\bar{\lambda}$ the average arrival rate
- $\bar{T}$ the average response time of a job (waiting time + service time), i.e. the average time spent in the system

Theorem 6.21 (Little's law)

$$\boxed{\bar{N} = \bar{\lambda} \cdot \bar{T}}$$

Remark:

- Little's Law in the above form connects the random variables taking on average properties of a queueing system, and holds regardless of the probability distributions that describe the arrival and service times.
- It also holds for the expected values of $\bar{N}, \bar{\lambda}, \bar{T}$.

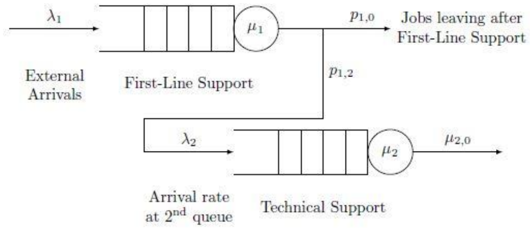
Little's law also holds for systems other than M/M/1 queues.

Applying Little's law gives us

$$\boxed{\bar{T} = \frac{\bar{N}}{\bar{\lambda}}}$$

1.3 Queueing Networks

Sometimes, systems consist of more than a single queueing system. Consider for instance a support call center where calls are handled and redirected.



Jobs arrive from the outside with rate λ_1 and enter the queue for first-line support. After the first-line support served the job, with rate μ_1, a $p_{1,0} = (1 - p_{1,2})$ fraction leave the system. A $p_{1,2}$ fraction gets redirected to technical support. Technical support serves jobs with rate μ_2. Afterwards jobs leave the system.

Theorem 6.29 (Burke's Theorem). Consider a $M/M/1$ queue with arrival rate λ and service rate μ. If the system is stable, then in the steady state the time between two departures is exponentially distributed with parameter λ.
Remarks:

- Burke's theorem also holds for the more general $M/M/m$ queues.

Definition 6.31 (Queueing Network). A queueing network is a directed graph in which nodes represent queueing systems and edges direct jobs from one queueing system towards the next one. The network is **open** if external jobs arrive and depart the network, and **closed** if jobs never enter or leave the network.

- In the following, we consider an open network containing $M/M/1$ queueing systems. Let us denote the number of queues (nodes) in the network by n.
- External arrivals come from a Poisson distribution with some rate λ_0. An external arrival joins the queueing system (node) i with probability $p_{0,i}$, with rate $\lambda_{0,i} = \lambda_0 p_{0,i}$
- The serving rate of queueing system i is μ_i. After being served at node i, a job leaves the system with probability $p_{i,0}$ and joins queueing system j with probability $p_{i,j}$.
- Due to Burke's theorem: $\lambda_i = \lambda_{0,j} + \sum_{j=1}^{n} \lambda_j \cdot p_{j,i}$
- The utilization of node i with m_i servers is

$$\rho_i = \frac{\lambda_i}{m_i \cdot \mu_i}$$

Theorem 6.32 (Jackson's Theorem). Consider an **open** queueing network with n nodes where each node $v_i, i \in \{1, \ldots, n\}$ represents an $M/M/m_i$ queueing system. If all queues v_i are stable, then the steady state of the network is

$$\pi(k_1, \ldots, k_n) = \prod_{i=1}^{n} \pi_i(k_i)$$

$\pi(k_1, \ldots, k_n)$ denotes the stationary distribution, i.e. the probability that k_i jobs are in queueing system i; and $\pi_i(k_i)$ is the probability that k_i jobs are in node v_i.

Before applying the theorem, one needs to check that each queue is stable:

$$\rho_i = \frac{\lambda_i}{m_i \cdot \mu_i} < 1$$

Little's Law also applies to networks of queueing systems as a whole.

Theorem 6.33 (Gordon-Newell). Consider a closed queueing network with total population K and n nodes, where each node $v_i, i \in \{1, \dots, n\}$, represents an $M/M/m_i/n_i$ queue. If all queues v_i are stable, then the steady state of the network is

$$\pi(k_1, \dots, k_n) = \frac{1}{G(K)} \prod_{i=1}^{n} \rho_i^{k_i}$$

where $G(K)$ is the normalizing constant

$$G(K) = \sum_{\substack{(k_1,\dots,k_n) \\ k_i \le n_i, \sum k_i = K}} \prod_{i=1}^{n} \rho_i^{k_i}$$

and the values ρ_i are obtained from the λ_i satisfying the equations

$$\lambda_i = \sum_{j=1}^{n} \lambda_j \cdot p_{i,j}$$

2 Online

In many application domains events are not Poisson distributed. Sometimes we want to study worst-case behavior. The analysis tool is often referred to as Online Theory or Online Algorithms.

2.1 Competitive Analysis

Definition 7.2 (Competitive Analysis): An online algorithm is r-competitive if for all finite input sequences I

$$\boxed{cost_{ALG}(I) \le r \cdot cost_{OPT}(I) + k}$$

for a problem instance where we think in terms of cost (i.e. the more the worse). The smaller we can make r, the better. r may be a constant or depend on the input.

If we think in terms of benefit (i.e. the more the better) we have

$$\boxed{benefit_{ALG}(I) \ge \frac{1}{r} \cdot benefit_{OPT}(I) - c}$$

Definition 7.3 (Competitive Ratio): If $k = 0$ (or $c = 0$) in Def. 7.2, then the online algorithm is called **strictly r-competitive.** In this case, the worst-case ratio between the cost of the online and the cost of the optimal offline algorithm, called **competitive ratio**, is often considered directly.
Formally, the competitive ratio for cost is defined as:

$$r = \sup_{I \in \mathcal{J}} \frac{cost_{ALG}(I)}{cost_{OPT}(I)}$$

If we consider benefit:

$$r = \inf_{I \in \mathcal{J}} \frac{benefit_{OPT}(I)}{benefit_{ALG}(I)}$$

where $\mathcal{J}$ is the set of all finite input sequences I.

2.1.1 Procedure

The competitive analysis of an algorithm ALG consists of two separate steps. First, we show that for an arbitrary problem instance, the result of ALG is asymptotically at most a factor r worse than the optimal online result. This yields an upper bound on ALG's result, that is $cost_{ALG} \le r \cdot cost_{OPT} + k$.
If the task is to show that ALG is constant-competitive for a constant r, then we are done.
If we are interested in a tight analysis, we have to show that there is a problem instance where the result of ALG is a factor r worse than the optimal online result. This gives a matching lower bound on the objective value of the algorithm,

$$cost_{ALG} \ge r \cdot cost_{OPT}$$

Naturally, the second step is easier than the first one because we just have to find a "bad instance". The first step is often much more involved. A pattern that works quite often is the following.

1. Consider an arbitrary input sequence for ALG.
2. Partition the input sequence into *suitable* parts.
3. Show that $cost_{ALG} \le r \cdot cost_{OPT}$ for each part.

The tricky part here is to find a *suitable* partition in step 2.

2.2 Ski rental

Description: We want to ski but don't know whether we should buy or rent skis.
Question: When is the best time to buy? We assume that the accident happens at the worst possible time, i.e. right after we buy skis.
The ski rental problem consists of two values:

- Input: $u \in \mathbb{R}$, the time a skier will end up skiing (i.e. u is the time the accident happens). u is chosen by an adversary. ALG doesn't know u.
- Algorithm: $z \in \mathbb{R}$, time at which the algorithm will stop renting skis and buy skis for price 1.

We can then define cost functions of the online algorithm ALG and the optimal offline algorithm OPT:

$$cost_{ALG}(u) = \begin{cases} u\,, & if\ u \le z \\ z+1\,, & if\ u > z \end{cases}$$

$$cost_{OPT}(u) = \begin{cases} u\,, & if\ u \le 1 \\ 1\,, & if\ u > 1 \end{cases} = \min(u, 1)$$

Theorem 7.4. Ski rental is *strictly 2-competitive*. The best algorithm is $z = 1$, i.e. buy skis at time $z = 1$.
Proof: Let's investigate $z = 1$ in the ski rental algorithm. Then:

$$\frac{cost_z(u)}{cost_{opt}(u)} =$$

Cases	$u \le z = 1$	$u > z = 1$
$u \le 1$	$\frac{u}{u}$	impossible
$u > 1$	impossible	$\frac{1+1}{1}$

Thus, the worst case is $u > z = 1$, and the competitive ratio is 2. Is this optimal? Turns out yes (for proof see p. 21 Chp. 7)

2.3 Randomized Ski rental

We now choose the time we buy skis at random between z_1 and z_2 (with $z_1 < z_2$) with probabilities p_1 and $p_2 = 1 - p_1$ respectively.
The adversary will still choose the worst possible input, i.e. the accident will happen at $u_1 = z_1 + \epsilon$ or $u_2 = z_2 + \epsilon$, where $\epsilon \to 0$
Let's consider $z_1 = 1/2$, $z_2 = 1$, $p_1 = 1/2$ and $p_2 = 1/2$.
The costs of ALG then are:

- $u_1 : cost_{ALG}(u_1) = p_1(z_1 + 1) + p_2 z_1 = p_1 \frac{3}{2} + (1 - p_1)\frac{1}{2} = \frac{1}{2} + p_1$
- $u_2 : cost_{ALG}(u_2) = p_1(z_1 + 1) + p_2(z_2 + 1) = p_1 \frac{3}{2} + (1 - p_1)2 = 2 - \frac{p_1}{2}$

The cost of OPT are:

- $cost_{OPT}(u_1) = 1/2$
- $cost_{OPT}(u_2) = 1$

The competitive ratios then are:

- $u_1 : \frac{cost_{ALG}}{cost_{OPT}} = \frac{1/2 + p_1}{1/2} = 2p_1 + 1$
- $u_2 : \frac{cost_{ALG}}{cost_{OPT}} = \frac{2 - p_1/2}{1} = 2 - p_1/2$

The adversary will choose the larger of the two ratios, thus to minimize we set them equal: $2p_1 + 1 = 2 - \frac{p_1}{2} \Leftrightarrow p_1 = \frac{2}{5} \Rightarrow \underline{\frac{cost_{ALG}}{cost_{OPT}} = \frac{9}{5} = 1.8}$

2.4 Lower bounds

Theorem 7.9 (Von Neumann/Yao Principle). Choose a distribution over problem instances, e.g. $d(u)$ for ski rental. If for this distribution all deterministic algorithms cost at least r, then r is a lower bound for the best possible randomized algorithm.

3 Appendix

3.1 Probability Theory

3.1.1 Exponential Distribution

A random variable Y with the cumulative distribution function (CDF)

$$F_Y(t) = \Pr[Y \le t] := \begin{cases} 1 - e^{-\lambda t}, & for\ t \ge 0 \\ 0, & otherwise \end{cases}$$

Is exponentially distributed with parameter λ, or $Y \sim \exp(\lambda)$ for short.
The corresponding probability density function (PDF) is

$$f_Y(t) = \frac{d}{dt} F_Y(t) = \lambda e^{-\lambda t}$$

- $E[Y] = \frac{1}{\lambda}$
- $Var[Y] = \frac{1}{\lambda^2}$

The exponential distribution is the continuous analogue to the discrete-time geometric distribution

3.2 Power series

- $\sum_{k=0}^{\infty} aq^k = a + aq + aq^2 + \ldots = \frac{a}{1-q}$ für $|q| < 1$
- $\sum_{k=0}^{\infty} (k+1)q^k = 1 + 2q + 3q^2 + \ldots = \frac{1}{(1-q)^2}$ für $|q| < 1$
- $\sum_{k=1}^{\infty} \frac{1}{k^2} = 1 + \frac{1}{2^2} + \frac{1}{3^2} + \ldots = \frac{\pi^2}{6}$
- $\sum_{k=0}^{\infty} \frac{1}{k!} = 1 + 1 + \frac{1}{2} + \frac{1}{6} + \ldots = e$

$$\sum_{k=0}^{n} \binom{n}{k} \cdot x^k = (1+x)^n$$

$$\sum_{k=0}^{\infty} \frac{x^k}{k!} = e^x$$

$$\sum_{k=0}^{n} q^k = \frac{1 - q^{n+1}}{1-q}, |q| < 1$$

3.3 Trigonometry

3.3.1 Functions of $\boldsymbol{\alpha \pm \beta}$, $\boldsymbol{2\alpha}$

$\sin(\alpha \pm \beta) = \sin\alpha\cos\beta \pm \cos\alpha\sin\beta$

$\sin 2\alpha = 2\sin\alpha\cos\alpha$

$\cos(\alpha \pm \beta) = \cos\alpha\cos\beta \mp \sin\alpha\sin\beta$

$\cos 2\alpha = \cos^2\alpha - \sin^2\alpha = 2\cos^2\alpha - 1 = 1 - 2\sin^2\alpha$

$\tan(\alpha \pm \beta) = \frac{\tan\alpha \pm \tan\beta}{1 \mp \tan\alpha\tan\beta}$

$\tan 2\alpha = \frac{2\tan\alpha}{1-\tan^2\alpha}$

3.3.2 Sums and Products

$$\sin\alpha + \sin\beta = 2\sin\frac{\alpha+\beta}{2}\cos\frac{\alpha-\beta}{2}$$

$$\cos\alpha + \cos\beta = 2\cos\frac{\alpha+\beta}{2}\cos\frac{\alpha-\beta}{2}$$

$$\sin\alpha - \sin\beta = 2\cos\frac{\alpha+\beta}{2}\sin\frac{\alpha-\beta}{2}$$

$$\cos\alpha - \cos\beta = -2\sin\frac{\alpha+\beta}{2}\sin\frac{\alpha-\beta}{2}$$

$$\sin\alpha\sin\beta = \tfrac{1}{2}[\cos(\alpha-\beta) - \cos(\alpha+\beta)]$$

$$\cos\alpha\cos\beta = \tfrac{1}{2}[\cos(\alpha-\beta) + \cos(\alpha+\beta)]$$

$$\sin\alpha\cos\beta = \tfrac{1}{2}[\sin(\alpha-\beta) + \sin(\alpha+\beta)]$$

3.3.3 sin/ cos/ tan values

	0	$\pi/6$	$\pi/4$	$\pi/3$	$\pi/2$
sin	0	$1/2$	$\sqrt{2}/2$	$\sqrt{3}/2$	1
cos	1	$\sqrt{3}/2$	$\sqrt{2}/2$	$1/2$	0
tan	0	$\sqrt{3}/3$	1	$\sqrt{3}$	$\to \infty$
cot	$\to \infty$	$\sqrt{3}$	1	$\sqrt{3}/3$	0

3.4 Differentials

$(f \cdot g)' = f' \cdot g + f \cdot g'$	$(f/g)' = (f'g - fg')/g^2$
$(a^x)' = \ln(a) \cdot a^x$	$(\ln(x))' = 1/x$
$\left(f(g(x))\right)' = f'(g(x)) \cdot g'(x)$	