# A “Short” Introduction to Modeling Bacterial Gene Expression

Containing . . .

- A concise description of bacterial gene expression and its regulation
- How to use kinetics and thermodynamics to model bacterial synthetic gene networks
- A brief review of probability theory, stochastic processes, numerical integrators for stochastic differential equations, and stochastic simulation
- And ... some examples (to do).

Extracted from Howard’s Dissertation

# Contents

# List of Tables

# List of Figures

## Chapter 1

# Design of Synthetic Gene Networks

### 1.1 Introduction

The engineering of biological organisms to perform complex tasks is still in its infancy. There are numerous applications where the extreme genetic engineering of an organism can result in the insertion of a "program" of incredible usefulness. Possible applications include modifying a simple bacterium to detect small quantities of chemicals or proteins, such as TNT [2], and using gene therapy to treat a multitude of human diseases, including diabetes and cancer, by inserting the corrective DNA that produces the needed therapeutic proteins at the right times [3]. By rearranging naturally provided molecular components, such as transcription factors, mRNA hairpins, and DNA operator and promoter sites, into novel configurations, a variety of synthetic gene networks have been constructed to exhibit potentially useful dynamical or logical behaviors [4, 5]. In addition, by creating entirely new genetic components, including polydactal zinc fingers [6, 7, 8, 9, 10, 11, 12], chimeric activator or repressor fusion proteins [13, 14, 15], libraries of novel bacterial promoters with tunable basal expression [16], and RNA molecules that form small molecule-binding secondary structures [17, 18], we can extend the capabilities of our toolbox of genetic components to sensing new molecular signals and responding in with new phenotypes.

There are numerous examples of synthetic gene networks. Many of these gene networks, including transcriptionally or translationally regulated bistable switches [19, 20], transcriptional [21] or metabolically coupled [22] oscillators, combined switch-oscillators [23], cascades [24], feedback loops [25, 26], population-dependent activation in bacteria [24, 27] and yeast [28], combined metabolically coupled population-dependent activation systems [29], and light-repressed kinase-activated transcription factors [30], are prototypes of synthetic genetic programs generating increasingly complex behaviors. As more molecular components are characterized, an important goal is to identify additional gene and protein networks that exhibit new or improved behaviors.

However, it is still not well understood how the molecular interactions between DNA binding sites, RNAs, and proteins influence the dynamics of gene expression and the resulting phenotypes. In order to understand how these interactions affect the overall dynamics, one may develop mathematical models that quantitatively describe all of the significant molecular interactions in the synthetic gene network and compare the model results to experimental observations. Various types of models exist, from Boolean and graph networks to jump Markov processes. The purpose of modeling [*] is to gain insight into the key molecular interactions in the system, to connect molecular events to the observable behavior of the system, and to either study how small changes will affect the system's dynamics or predict which changes must be made in order to obtain a certain dynamical behavior.

Computational biology is, of course, a broad topic with numerous examples [31]. By using the chemical partition function, the effects of different genetic components on the regulation of transcription, including activator or repressor transcription factors, operator placement, DNA-looping, and other factors, has been studied [32, 33, 34]. Mathematical analyses of various rationally designed gene networks have also been performed, including a mixed feedback loop gene network [35] and a feed forward loop genetic network [36]. Synthetic gene networks have also been designed to study the virulence cycle of HIV [37] and to partially stop the onset of AIDS [38].

[*] My brief answer to the advantages and limitations of physical/chemical modeling is thus: An experiment shows what happens when reality exists at one of many possible configurations. A simulation models what happens when one of an infinite number of configurations is assumed to be real. Not all configurations occur in reality. And there's too many real configurations to perform each experiment. So a model without experiments can be wrong. And an experiment without a model is worthless. That's why you need both.

Genetic networks have been modeled to study the effects of stochastic resonance and noise on cellular memory [39], calcium oscillations [40], neuron dynamics [41], enzyme futile cycles [42], and quorum sensing [43]. Finally, optimization techniques can identify a synthetic gene network with certain desired behaviors, including ones using either evolutionary computation [44, 45] and simulated annealing [46]. This is a brief list of studies and is not meant to be exhaustive.

In our work, we model the interactions participating in regulated gene expression as a system of chemical and biochemical reactions and use stochastic process theory to determine the *stochastic dynamics* of the system. Stochasticity is caused by the small numbers of participating molecules, such as regulatory DNA-binding proteins, and has been experimentally observed in gene expression [47]. These mathematical models attempt to quantitatively predict the dynamics of the synthetic gene network in response to environmental and regulatory stimuli. They also present a systematic means of identifying the regulatory connections between genes that result in certain dynamical or logical behaviors, such as bistable switching or oscillations, and how the kinetics of the constituent interactions quantitatively change the behavior of the synthetic gene network.

### 1.1.1 An Overview of the Chapter

We begin by presenting a brief review of regulated gene expression. We first focus on describing the important mechanistic interactions between the gene expression machinery and its DNA and RNA binding sites in terms of the transcriptional and translational initiation, elongation, and termination processes. We continue by detailing how the DNA and RNA sequences affect the basal rates of these processes, including the sequence determinants that modulate the kinetics of the rate-limiting steps of gene expression. We then discuss how the binding of regulatory molecules, including activating or repressing transcription factors and microRNAs, can also modulate the rate of gene expression. Finally, we discuss the participation of the RNA Degradosome and the proteosomes in degrading RNA and protein molecules and how bacteria can specifically target RNA and protein molecules for faster degradation.

In the next section, we discuss how we model regulated gene expression in bacteria as a system of chemical and biochemical reactions. We present our systematic method of converting the significant molecular interactions in the transcription and translation processes of gene expression, including all protein-DNA, protein-RNA, protein-protein, and RNA-RNA interactions, into a system of chemical reactions. The kinetics and thermodynamics of these interactions are empirically measured in the laboratory and are summarized where they are available. Ordinarily, one uses a full kinetic mathematical description of the system of chemical reactions. However, in many cases, it is often more feasible to assume that the protein-DNA interactions at the promoter are in chemical equilibrium. In this case, we describe how to apply the chemical partition function as a simpler mathematical description of the mechanistic interactions involved in Holoenzyme and transcription factor assembly.

In the third and fourth sections, we present our work on designing synthetic gene networks to perform two different useful behaviors. The first example is a system of fusion proteins, called a *protein device*, which activates gene expression if and only if two different transcription factors are both present [48]. The activation of gene expression thus mimics the Boolean ”AND” logical function. The system has a number of advantages, including modularity, scalability, high fidelity, and a rapid response to its pair of inputs. By connecting together multiple protein devices, one may program bacteria to respond to a specified set of inputs with a pre-determined genetic response. The second example is a system of three genes whose repressor regulatory connections spontaneously produce long-lived oscillations [49]. We explore how the structure of the promoter region controls the period and amplitude of these oscillations.

## 1.2 An Overview of Regulated Bacterial Gene Expression

Gene expression is the mechanistic process by which a cellular organism converts the stored information in its genome into the production of functional RNA and protein molecules. These molecules regulate and catalyze all of the biochemical reactions in the cell and are responsible for metabolism and cell growth, signal transduction, cell motility, and cellular differentiation. By controlling the production of these molecules through the regulation of the gene expression mechanism, the cell can dynamically sense its environment and alter its internal behavior, including the ability to replicate itself and change its environment. In short, everything that separates a cellular organism — or, using a common definition, life — from its purely chemical counterpart, arises from the *regulated* expression of its genes.

Here, we will review the orchestrated series of catalyzed biochemical reactions that first transcribes the base pair sequence of DNA into the ribonucleic acid sequence of messenger RNA followed by the translation of that mRNA into the polyamino acid sequence of a protein. The original grand hypothesis was given the name "the Central Dogma" by Francis Crick [50]. The focus is on bacterial gene expression because, compared to eukaryotes, the biochemical machinery inside a simple bacterium, such as *Escherichia coli*, is more understood at a quantitative level. We then describe the many mechanisms by which a cellular organism can alter the *rate* of gene expression in response to intracellular signals at the molecular level. We will find that, in every step of the process from DNA to mRNA to protein, there are numerous ways for an organism to increase or decrease the final rate of production of an RNA or protein. In order to design synthetic gene networks, we must harness these mechanisms for our own purposes.

### 1.2.1 Transcription

The expression of a gene begins with transcription. Transcription is an endothermic series of biochemical reactions where the RNA polymerase, a catalytic protein made of up four subunits called $\alpha$, $\beta$, $\beta'$, and $\omega$, binds to double stranded DNA and converts its base pair sequence into an RNA transcript containing a sequence of ribonucleotides [51] complementary to the template strand of the DNA. The mechanism of transcription can be separated into three distinct steps: initiation, elongation, and termination.

### Transcriptional Initiation

To initiate transcription, RNA polymerase must bind to the beginning of a gene, called the *promoter recognition region* [52]. However, the RNA polymerase itself has only weak attractive interactions (or low affinity) for DNA and is only capable of "sliding" across DNA, looking for regions of greater affinity. Instead, another protein called a $\sigma$-factor [53] binds strongly to the promoter and uses its strong attractive interactions with RNA polymerase to form a larger multisubunit complex called the Holoenzyme [54]. The initial binding of the Holoenzyme to the promoter DNA is often the rate limiting step of transcriptional initiation [55]. The structure of the Holoenzyme forms a pair of pincers with an entrance and exit for double stranded DNA and an exit channel for single stranded RNA.

After the Holoenzyme forms and binds to promoter DNA, its pair of pincers can undergo a closed-to-open conformational change that allows double stranded DNA to enter the complex through the entrance channel, but only by unwinding into the two separate coding and template strands. The Watson-Crick hydrogen bonding between DNA base pairs (e.g. the A:T and G:C binding) is strong, but the RNA Polymerase has stronger protein-DNA interactions that facilitates the unwinding. The unwinding of the double stranded DNA is often called melting. The same Watson-Crick base pairing also exists between DNA and RNA base pairs, but it is significantly weaker and, since RNA substitutes a uridine nucleic acid for a thymine one, includes the alternate A:U base pairing. Once the template strand of the DNA is inside, it may form Watson-Crick base pairing with complementary monomeric ribonucleic triphosphates, such as ATP, GTP, CTP, and UTP, which are free floating in solution. In order for the RNA polymerase to translocate forward and begin transcription, it must break the protein-DNA interactions behind it and form new ones in front of it. This is referred to as *promoter escape*.

The RNA polymerase uses the complementary base pairing between DNA and RNA and the high energy bond in the triphosphate to catalyze the polymerization of ribonucleotides in the 5' to 3' direction. The 5' and 3' refers to the positions of the carbon backbone of the ribose sugars. For the first ten nucleotides, the efficiency of this polymerization is extremely low, resulting in the repeated release of short RNA transcripts (called *abortive initiation*). Once an RNA transcript is aborted, the RNA polymerase will backtrack and rebind at the promoter site to repeat the initiation process. After the first ten nucleotides have been polymerized, the $\sigma$-factor will partially or completely unbind from the Holoenzyme complex, leading to further conformational changes. These structural changes increase both the rate and efficiency of the polymerization reaction by tightening the protein-DNA contacts. After these structural changes, the process of transcriptional elongation begins.

### Transcriptional Elongation

During transcriptional elongation, the RNA polymerase is tightly bound to the DNA and continues adding complementary ribonucleotides onto the end of the RNA transcript at 30 to 50 nucleotides per second, depending on the concentration of the monomeric ribonucleic acid triphosphates [56]. The RNA polymerase:DNA:RNA complex is often called the transcriptional elongation complex. The complex will infrequently catalyze the addition of an incorrect ribonucleotide due to a faulty DNA:RNA base pairing, leading to an error in transcription. To correct these errors, it has an inefficient editing mechanism. The catalytic site has a constant, but relatively slow, rate of excision of existing nucleotides followed by a backtracking motion. Because movement of the nascent mRNA transcript through the exit channel is much slower when an incorrect RNA:DNA base pairing exists, the excision activity favors the removal of slow-moving incorrect ribonucleotides, but also leads to sporadic removal of correct ribonucleotides. In the end, transcriptional elongation results in fewer than 1 in $10^4$ errors, but is more error prone than DNA replication, which has fewer than 1 in $10^7$ errors.

### Transcriptional Termination

RNA polymerase will continue transcribing DNA into an RNA transcript until it disassociates from the DNA, releasing its strong protein-DNA interactions, and resulting in transcriptional termination [56]. Anything that prevents the

RNA polymerase from continuing its polymerization reaction and translocation forward will result in its disassociation. There are two general classes of terminators: (1) proteins or RNA secondary structures that prevent the forward translocation of RNA polymerase and (2) AT rich RNA sequences that weaken the DNA:RNA base pairing inside the catalytic active site of RNA polymerase [56]. The most common terminator is an RNA sequence that utilizes both mechanisms. Any AU rich mRNA sequence followed by a GC rich two-fold symmetric sequence will terminate transcription. The AU rich sequence weakens the affinity of the RNA:DNA base pairing because A:U base pairing has two hydrogen bonds while G:C base pairing has three. The GC rich two-fold symmetric sequence halts translocation of the RNA polymerase by folding into a high affinity hairpin structure. Once the RNA polymerase unbinds from the DNA it releases its synthesized RNA transcript.

### 1.2.2 Translation

Translation is an endothermic series of biochemical reactions that reads the sequence of an RNA transcript and synthesizes the corresponding primary amino acid sequence of a protein. The translation process may also be separated into the three steps of initiation, elongation, and termination. However, the only type of RNA transcript that is translated is messenger RNA, which comprises less than 10% of the total mass of RNA in a cell [57]. Messenger RNAs (mRNA) contain special sequence determinants in its 5' UnTranslated Region (UTR) that enables a ribosome to bind to it and begin translation. In addition, bacteria often organize the translation of multiple co-regulated proteins into a multi-protein gene, called an operon. The mRNA transcripts transcribed from operons contain multiple start sites for protein translation, resulting in the sequential translation of multiple different protein products.

Other types of RNA transcripts lack these sequence determinants and instead serve a variety of functional roles by binding to proteins, other RNAs, or small molecules and playing a key role in a variety of processes, including the regulation of translation (e.g. shRNAs and microRNAs [58, 59]), the catalysis of reactions (e.g. ribozymes [60]), and the sensing of the intracellular environment (e.g. aptamers [17, 18]). In this section, we focus on the roles of the ribosome, messenger RNA, ribosomal RNA, and transfer RNA in the translation process.

The ribosome is a large multisubunit macromolecular complex containing 55 different protein and RNA molecules, separated into the 50S and 30S protein subunits and 5S, 23S, and 16S RNA subunits [61]. The RNA molecules in the ribosome, called ribosomal RNA (rRNA), are responsible for binding to messenger RNA, initiating the translation process, and assisting in the catalytic polymerization of additional amino acids. Another type of RNA transcript, called transfer RNA (tRNA), is responsible for converting the nucleotide sequence in an mRNA transcript into a corresponding amino acid sequence of a protein. This transfer of information is accomplished entirely via Watson-Crick interactions between RNA molecules.

### Translational Initiation

Translation initiation begins when the smaller 30S ribosomal subunit, containing the 16S rRNA and the initiator tRNA, binds to the 5' UTR region of an mRNA transcript. Any attractive interactions that increase the affinity between the 30S subunit and 5' UTR will increase the rate of translation initiation. One such interaction is the Watson-Crick RNA:RNA base pairing that exists between the 3' end of the 16S rRNA and a special sequence, called the Shine-Dalgarno sequence [62], located in the 5' UTR. In gram positive bacteria, this interaction is the primary one responsible for translation initiation. However, in gram negative bacteria, the ribosomal S1 protein located in the 30S subunit also has strong attractive interactions with AU rich sequences in the 5' UTR [63, 64]. A second important interaction is the presence of the ribonucleotide sequence AUG that binds to the initiator tRNA and signifies the start site of protein synthesis. The total effect of these attractive interactions is the physical alignment of the mRNA within the 30S subunit, leading to the binding of the larger 50S subunit, and the initiation of translation. Lastly, there are three translation initiation proteins, named IF1, IF2, and IF3, that bind to the assembled ribosome and mediate the needed conformational changes.

Importantly, once a ribosome begins translation and translocates forward, leaving the 5’ UTR vacant, another 30S subunit may bind and initiate an additional round of translation. Multiple ribosomes may simultaneously translate the same mRNA transcript, leading to the synthesis of many proteins per single transcript. In addition, because bacteria lack a nuclear membrane, the translation of the mRNA transcript begins as soon as the 5’ UTR has been transcribed. Consequently, in bacteria, both the transcription and translation processes occur at the same time and in close proximity.

Once the larger 50S subunit binds, the mRNA transcript is securely sandwiched between the two ribosomal subunits, forming the ribosomal elongation complex [65]. Translational elongation is a highly efficient and processive series of endothermic chemical reactions that involves numerous repetitive conformational changes, Watson-Crick RNA:RNA base pairing, and catalyzed chemical reactions. It involves the orchestration of the ribosome, mRNA, and tRNA molecules.

Transfer RNAs are RNA transcripts that fold into clover leaf secondary structures and become enzymatically modified to contain covalently modified base pairs, such as psuedouridine, dihydrouridine, inosine, and $N$,$N$-dimethyl gaunine [57]. While there are only 20 natural amino acids in a protein, there are at least 31 different tRNAs in bacteria, leading to multiple tRNAs carrying the same amino acid. There are two unique interactions in a tRNA that are responsible for converting the mRNA sequence into an amino acid sequence of a protein. The first is the *anti-codon* loop, a three letter sequence of ribonucleotides, that forms selective RNA:RNA base pair hydrogen bonds with a corresponding *codon* in the mRNA transcript. The other selective feature is a short single stranded sequence in its 3’ end. This sequence binds to an aminoacyl-tRNA synthetase, an enzyme that attaches the correct amino acid to the tRNA to create an amino acyl-tRNA. Except in some simpler organisms, there is typically a single aminoacyl-tRNA synthetase for each amino acid. The synthetase uses ATP hydrolysis to create the high energy amino acyl-tRNA covalent bond, which is later used in the polymerization reaction.

### Translational Elongation

The ribosomal elongation complex contains a channel and three binding sites. The channel allows single stranded mRNA to enter and exit [57]. The binding sites, called the A, P, and E sites, are the locations in which the amino acyl-tRNAs enter the ribosome, help catalyze the polymerization reaction, and exit the ribosome, respectively. Between polymerization reactions, the mRNA transcript is locked into position so that three base pairs of its sequence are each accessible to A and P sites [66]. The elongation of the polypeptide chain then occurs in three steps:

1. A charged amino acyl-tRNA enters the A site and uses its anti-codon loop to form Watson-Crick RNA:RNA hydrogen bonds with the codon in the mRNA transcript. Amino acyl-tRNAs with the correct anti-codon:codon interaction will stably bind to the A site, thus reading the mRNA sequence. The amino acyl-tRNA in the A site is adjacent to another tRNA in the P site.

2. The amino acyl-tRNA in the P site also forms RNA:RNA base pairing with its accessible mRNA transcript codon sequence. The 50S subunit of the ribosome uses the charged amino acid and the high energy bond in the amino acyl-tRNA to catalyze a peptidyl transferase reaction, covalently adding the amino acid to the C-terminal end of the polypeptide chain. The reaction is coupled to multiple conformational changes in the ribosome, moving the amino acyl-tRNA in the A site into the P site and the now empty tRNA in the P site into the E site.

3. A second set of conformational changes in the ribosome loosens its tight grip on the mRNA transcript, pulls it forward by exactly three base pairs, and then locks it down again to prepare for around round of catalysis. The empty tRNA in the E site loses its RNA:RNA base pairing, exits the ribosome, and eventually becomes charged again by binding to an aminoacyl-tRNA synthetase.

### Translational Termination

The process of translational elongation continues until a stop codon is reached, which is either UAA, UAG, or UGA. These stop codons have no corresponding tRNAs, leading to the stalling of the processive ribosomal motion. A protein release factor can bind to stalled ribosomes and catalyze the addition of a water molecule onto the C-terminal end of the polypeptide, ending the chain and allowing the protein to exit the ribosome. Without the catalysis of additional peptide bonds, the ribosome loses affinity for the mRNA transcript and disassociates into the 30S and 50S subunits.

Importantly, the ribosomal RNAs (and not the proteins!) are more responsible for the catalysis of the peptidyl transferase reaction, the mRNA sequence recognition, and the structural core of the ribosome macromolecular structure. The proteins serve as an additional glue that keeps the core structure together, mediate the different conformational changes of the ribosome, and ensure a more processive motion. Besides ribosomal proteins, there are other proteins, called additional elongating factors, that assist in removing incorrectly added amino acids and reducing errors in protein synthesis.

### 1.2.3 The Regulation of Transcriptional Interactions

The kinetic rate of each mechanistic step in the process of transcription can be manipulated by a number of determinants: the DNA sequence of the gene, the availability of different $\sigma$-factors, and the presence or absence of additional DNA-binding proteins whose only purpose is to regulate the rate of expression of one or more genes [67, 55]. These latter molecules are called transcription factors. The rate-limiting step of the transcription process, the step with the slowest rate, is typically transcriptional initiation. Transcriptional initiation itself consists of multiple binding events and conformational changes and each of these internal steps may be targetted for regulation. Here, we focus on a number of different mechanisms of the regulation of transcriptional initiation.

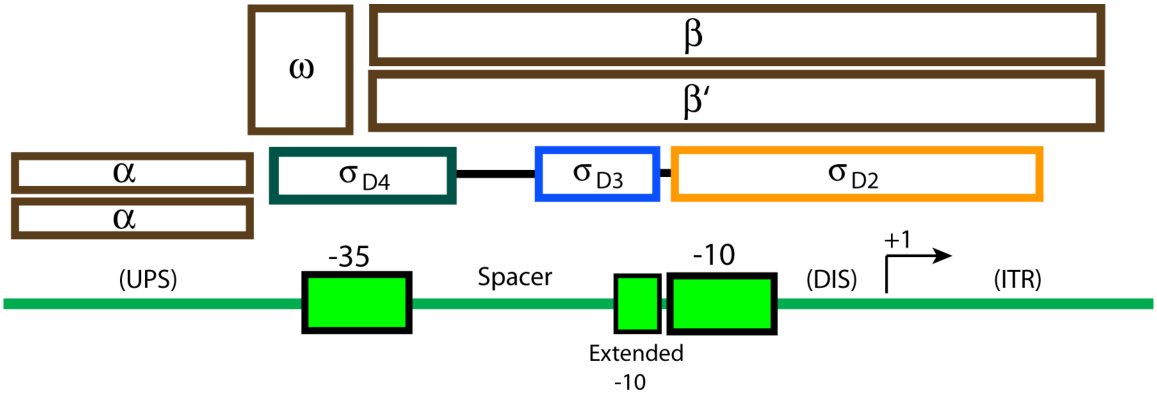

Figure 1.1: The linear alignment of the RNA polymerase subunits, the conserved protein domains of sigma factors, based on $\sigma^{70}$, and the different regions of promoter DNA are shown. The promoter DNA is subdivided into multiple genetic elements: the upstream element (UPS), the -35 and -10 hexamers, the extended -10 region, the spacer, the discriminator (DIS), and the initial transcribed region (ITR). The core RNA polymerase, composed of the ααωββ' subunits, contacts the promoter at two regions: the two α subunits can form attractive interactions with an AT rich UPS region and the ββ' crab claw wraps around the DIS and ITR regions. The $\sigma^{70}$ family of sigma factors have four conserved domains; the three that are shown have been crystallized while $\sigma_{D1}$ is disordered in solution. The $\sigma_{D2}$, $\sigma_{D3}$, and $\sigma_{D4}$ domains respectively form contacts with the -10 hexamer, extended -10, and -35 hexamer regions. The $\sigma^{70}$ housekeeping sigma factor contains all four domains while the $\sigma^{S}$ general strong response factor contains only the $\sigma_{D2}$ and $\sigma_{D3}$ domains, thus explaining why it only binds to the -10 and extended -10 regions.

Table 1.1: A list of consensus sequences for E. coli σ factors.

| Sigma Factor | -35 | Spacer Length | -10 / -10 extended |
|---|---|---|---|
| $\sigma^{70}$ | TTGACA | 15-19 bp | TATAAT [68] |
| $\sigma^{S}$ | None | | CTATACT [69] |
| $\sigma^{32}$ | TTGAAA | 12-16 bp | CCCCATTT [70] |
| $\sigma^{F}$ | TAAA | 13-17 bp | GCCGATAA [68] |

### Holoenzyme and Promoter DNA Interactions

The first and primary rate-limiting step of transcriptional initiation is the rate at which both σ-factor and RNA polymerase (the Holoenzyme) can bind to the promoter and initiate transcription. A productive initiation event requires the Holoenzyme to successfully undergo the closed-to-open conformational change, escape from the promoter region, and then successful transcribe the first 10 ribonucleotides. The sequence of the promoter recongition region is a significant determinant of the kinetic rate of each of these steps.

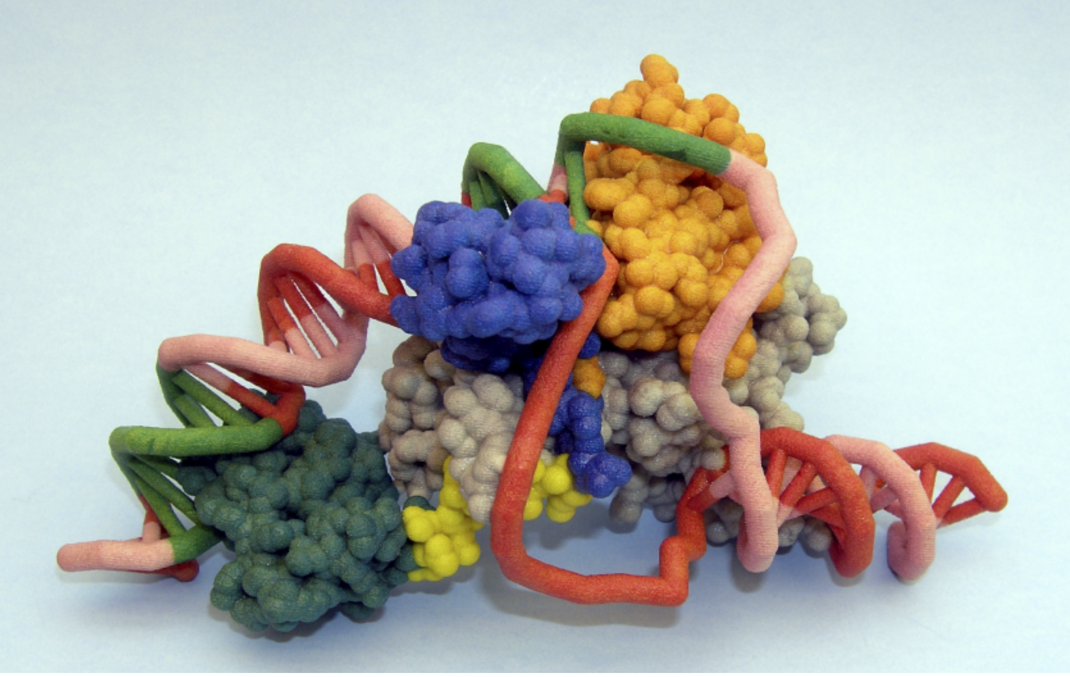

Figure 1.2: A plaster atomic model of the RNA polymerase-$\sigma$ factor (Holoenzyme) complex bound to promoter DNA is shown. The model shows how the (light green) -35 and -10 hexamers in the (red) promoter DNA contact the (orange) $\sigma_{D2}$, (blue) $\sigma_{D3}$, and (dark green) $\sigma_{D4}$ sigma factor protein domains. The (yellow) linker between the $\sigma_{D3}$ and $\sigma_{D4}$ domains and the (grey) core RNA polymerase are also shown. (Image courtesy of the Pingry school in Martinsville, NJ in coordination with Dr. Tim Herman of the Milwaukee school of Engineering.)

We will discuss the different regions of the promoter and how each one interacts with the Holoenzyme. In addition, there are multiple $\sigma$-factors, each having varying affinities to different promoter sequences [71]. By regulating the steady-state concentration of each $\sigma$-factor, bacteria can alter the expression rates of their entire genome in response to metabolic or environment stresses, such as a lack of food or heat shock.

Relative to the start of transcription (defined as +1) of a gene, a bacterial promoter is located from -60 to 20 base pairs and is separated into the promoter recognition region (PRR) from -60 to +1 bp and the initial transcribed region (ITR) from +1 to +20 bp. The basal or constitutive rate of gene expression of a bacterial promoter varies between $10^{-4}$ - 1.0 mRNA transcripts per second and depends on the sequence of both the PRR and ITR. Within this range, the basal rate is increased or decreased by the presence or absence of key sequence determinants that mediate protein-DNA interactions with the Holoenzyme. The nucleotides in the sequence determinant typically make electrostatic interactions with amino acids of the multi-domain $\sigma$-factor or the $\alpha$-subunit of RNA polymerase. The spacing of the sequence determinants also determines the relative positioning of the Holoenzyme on the promoter and incorrect spacing can lead to torsional stress and an entropic barrier.

Overall, any sequence determinant that mediates a protein-DNA interaction that increases the rate of a rate-limiting step of transcriptional initiation will result in the greater production rate of mRNA transcripts. However, too many sequence determinants with strong protein-DNA interactions will decrease the total rate of mRNA production by preventing the Holoenzyme from escaping the promoter and initiating transcription. Consequently, the rate of transcription initiation depends on the promoter sequence in a highly complicated manner, dependent on a multi-dimensional potential energy surface with multiple intermediate states of varying depth and breadth.

The bacterial promoter recognition region can be further separated into seven distinct regions, called the upstream (UPS) domain, the -35 hexamer, the spacer region, the extended -10 sequence, the -10 hexamer, and the discriminator (DIS) (see Figure 1.1) [72]. The -35 hexamer, the -10 hexamer, the extended -10, and the spacer regions make electrostatic contacts with the $\sigma$-factor. The structural differences between $\sigma$-factors determine their affinity to different sequence determinants. The "housekeeping" $\sigma^{70}$ (gene name: RpoD) factor controls global gene expression in the absence of environmental or metabolic stresses. It contains four protein domains, makes electrostatic contacts with the -35 and -10 hexamers, and requires the spacer to be between 17-19 base pairs to minimize the torsional stress between protein domains.

In *Escherichia coli*, there are five additional $\sigma$-factors of the $\sigma^{70}$ type, including $\sigma^{S}$ (RpoS), $\sigma^{32}$ (RpoH), $\sigma^{E}$ (RpoE), $\sigma^{F}$ (RpoF), and FecI, whose steady-state active concentrations are up- or down-regulated according to the cell's phase of growth or stress condition [71]. For example, the $\sigma^{S}$-factor is a global regulator of gene expression in response to any environmental or metabolic stress [69] and induces cell growth to shift into stationary phase

[73] while the $\sigma^{32}$ factor responds more specifically to the denaturement of cellular proteins, due to heat shock [70]. In Table 1.1, we list the consensus determinant sequences for four important *E. coli* $\sigma$-factors. A consensus sequence is the *most common* promoter sequence that binds to protein, but not necessarily the one that binds the strongest.

Importantly, it is not only the promoter sequence, but also the concentration of active $\sigma$-factor and the binding competition between $\sigma$-factors that determine the rate of Holoenzyme formation at the promoter. To downregulate a large number of genes with similar sequence determinants, bacteria can preferentially degrade, sequester, or stop production of the corresponding $\sigma$-factor . The sequences of most promoters also partially match multiple consensus sequences, resulting in the competitive binding of two or more $\sigma$-factors to a promoter. This combination of sequence determinants allows a bacteria to fine tune the basal regulation of genes across their entire genome according to multiple stresses and with different magnitudes of response.

In contrast, the upstream and discriminator domains do not make significant contacts with the $\sigma$-factor. An AT rich sequence in the upstream domain will make numerous attractive interactions with the $\alpha$-subunit of RNA polymerase, greatly increasing its affinity to the promoter, but also decreasing the rate of promoter escape [74]. The discriminator region modulates the initial rate of the unwinding, or melting, of double stranded DNA and also determines the starting nucleotide for transcription [75]. The length of the discriminator region is typically 6-8 base pairs with more GC content sometimes resulting in a slower closed-to-open conformational change.

Overall, the sequence of a promoter will determine the basal rate of Holoenzyme formation, promoter escape, and initiation of productive transcription. The dynamic changes in active $\sigma$-factor concentration enable the bacteria to make genome-wide changes to this basal rate. However, in order to selectively regulate single genes in response to a stimulus, the bacteria must utilize additional DNA-binding proteins, called transcription factors.

Table 1.2: A list of the thermodynamic binding free energies between the *lac*, *tet*, and *ara* transcription factors and their respective DNA operators at physiological temperature, pH, and salt concentration.

| Transcription Factor | Operator | Sequence | $\Delta G_{binding}$ kcal/mol |
|---|---|---|---|
| LacI tetramer | $O_1$ | AATTGTGAGCGGATAACAATT | $-14.12$ to $-14.90$ [78, 79] |
| LacI tetramer | $O_2$ | AAATGTGAGCGAGTAACAACC | $-13.22$ [78] |
| LacI tetramer | $O_3$ | GGCAGTGAGCGCAACGCAATT | $\approx -12$ [78] |
| LacI tetramer | Non-specific | | $-6.3$ to $-8$ [79, 80] |
| TetR dimer | $O_1$ | ACTCTATCAATGATAGAGTC | $-15.0$ [81] |
| TetR dimer | $O_2$ | TCCCTATCAGTGATAGAGA | ND |
| AraC dimer | $I_1$ | CATAGCATTTTTATCCATAA | $\approx -10.7$ [82] |
| AraC dimer | $I_2$ | AGCGGATCCTA | $\approx -8.9$ [82] |
| AraC dimer | $I_1 + I_2$ | CATAGCATTTTTATCCATAA . . .<br>. . . GATT AGCGGATCCTA | $-16$ [82] |

## Transcription Factor, Holoenzyme, and Operator DNA Interactions

A transcription factor is any DNA-binding protein that modulates the rate of production of mRNA transcripts by interacting with the Holoenzyme. Transcription factors most often regulate the rate of the transcriptional initiation by modulating the binding affinity of the Holoenzyme to promoter DNA, altering the rate of its closed-to-open conformational change, blocking its promoter escape, or a combination of these effects. The motifs that confer a DNA-binding capability to a protein domain includse $\text{Cys}_2\text{His}_2$ or $\text{Cys}_2\text{Cys}_2$ zinc fingers, steroid receptors, leucine zippers, helix-turn-helices, and helix-loop-helices [76]. Each DNA-binding protein binds to a consensus DNA sequence between 9 and 20 base pairs, called an *operator*. The DNA operator localizes the transcription factor to a spatial position adjacent to the Holoenzyme, enabling it to make attractive or repulsive electrostatic, van der Waals, or hydrophobic interactions [77].

Transcription factors may bind to DNA as a monomer or they may form multi-subunit complexes that bind to quasi-symmetric DNA sequences in a cooperative fashion. In addition, DNA-binding proteins are often composed of multiple protein domains with some providing a DNA-binding capability while others conferring a transcriptional regulatory activity. In Table 1.2, we show the DNA sequences of the operators that bind to the *lac*, *tet*, and *ara* transcription factors and their corresponding Gibbs free energies of binding. Notice how the sequences of the *lac* $O_2$ and $O_3$ operator sites deviate from the sequence of the $O_1$ site, resulting in a weaker protein-DNA interaction.

Table 1.3: A list of the thermodynamic binding free energies between the inducer-bound *lac* and *tet* repressors and their respective DNA operators.

| DNA-Binding Protein | Operator | Inducer | $\Delta G_{binding}$ kcal/mol |
|---|---|---|---|
| LacI tetramer | $O_1$ | IPTG | $-10.9$ [78] |
| TetR dimer | $O_1$ | aTC | $-11$ [81] |

In addition, note how the relatively weak protein-DNA interactions between the AraC dimer and either the $I_1$ or $I_2$ sites may be strengthened by simultaneously binding to an adjacent $I_1/I_2$ site with a 4 bp spacer. The total Gibbs free energy is, however, not simply the sum of the two because of the entropic costs of bending the protein dimer. Finally, also note that all DNA-binding proteins have the capability to bind to non-specific DNA with a relatively weak electrostatic interaction. The specificity of a transcription factor towards its operator site is always relative to its binding to non-specific DNA.

A variety of transcription factors can also bind to small molecules, called inducers, that (once bound) trigger a conformational change that either strengthens or weakens the affinity to their DNA operator. These transcription factors are often used as intracellular sensors that turn specific genes on or off in the presence or absence of a specific small molecule, such as a food source or antibiotic. In Table 1.3, we list the inducers of the *lac* and *tet* repressors with the corresponding Gibbs free energy of their DNA-binding interactions, once bound to their inducer.

The regulatory effect of the transcription factor depends on the location of its operator, the interactions that the protein forms with the Holoenzyme once it is bound, and any adjacently bound proteins that may strengthen or interfere with this interaction. A transcription factor that participates in any interaction that either increases or decreases the overall rate of mRNA production is respectively called an activator or repressor [83, 84]. The most common type of bacterial transcription factor is a repressor that binds to a promoter-overlapping operator site and uses van der Waals forces to prevent the Holoenzyme complex from binding, which is also called steric hindrance. The repressing effects of steric hindrance largely depend on the location of the operator and the extent of its overlapping with the promoter region.

Bacterial activators are less common, but many examples exist; synthetic examples of bacterial activators have also been constructed [85]. A bacterial activator possesses attractive electrostatic or hydrophobic interactions with the Holoenzyme so that, once it binds to a spatially adjacent position, it can increase the rate of Holoenzyme formation or its closed-to-open conformational change. Activators often bind to DNA operators directly upstream of the -35 or UPS element of the promoter while also forming electrostatic interactions with the C-terminus of the $\alpha$-subunit of RNA polymerase. The presence of an activator often results in a 10 to 100 fold increase in the rate of mRNA production. In addition, a second transcription factor, called a cofactor, can bind to an adjacent position relative to the first one to strengthen or weaken its activating or repressing interactions [52]. Multiple cofactors can bind to combinatorially control the regulated rate of transcriptional initiation according to the presence or absence of multiple DNA-binding proteins. However, in order to remain clear of the processive forward motion of the Holoenzyme, activating transcription factors must bind upstream of the promoter, creating a restriction on the number of activating proteins that can both bind close enough to the Holoenzyme to confer a favorable interaction. The evolution of gene regulation has overcome this constraint by utilizing long range regulatory effects, such as DNA looping.

### DNA Looping and Long-Range Effects

Like other polymers, double stranded DNA can spontaneously, sporadically, and transiently form loops with varying optimal persistence lengths. However, if one or more proteins bind to a pair of operator sites at a sufficient distance, they can stabilize the transient loop into a stable structure. This stable structure creates a three dimensional space in which additional attractive or repulsive interactions may form with the Holoenzyme at the promoter. Stable loops which prevent the Holoenzyme from binding to the promoter will result in repression while loops that place activating transcription factors in a favorable position will increase the rate of transcriptional initiation [34]. The activation or repression of transcriptional initiation depends on the loop length, the size of the loop-mediating proteins, and typically oscillates with a period of ≈11 bp [86], which corresponds to the phase of the double helix of DNA.

By using multiple DNA-binding proteins to form loops of varying lengths with different activating or repressing capabilities, bacteria can dramatically increase the combinatorially regulation of transcriptional initiation. Transcription factors may coregulate gene expression by stabilizing a DNA loop to allow another protein to confer its regulatory activity or, conversely, it may destabilize a DNA loop by sequestering the needed operator site or stabilizing another competitively forming DNA loop. The sequence of the DNA [87] and any proteins bound to it can also change the elastic properties of DNA and, consequently, change the optimal persistence length of a putative DNA loop. All of these interactions can take place hundreds to thousands of base pairs from the promoter, leading to the possibility of significant long-range effects in gene regulation. Further, these long-range effects contribute towards the organization of the entire bacterial genome into regions of sustained activation or repression of transcription [88]. These long-range effects are thought to be less significant in bacteria, but are highly important and common place in eukaryotes.

Bacteria use a diversity of mechanisms to regulate transcriptional initiation and control the production of mRNA transcripts. To gain an even finer control over the production of proteins, bacteria also regulate the translation process.

### 1.2.4 The Regulation of Translational Interactions

The rate of production of proteins through the translation of mRNA transcripts is most commonly regulated at its initiation and elongation steps. Similar to how the sequence of promoter DNA controls the basal rate of transcription, the sequence of the 5' UnTranslated Region (UTR) of an mRNA transcript controls the basal rate of translation. Sequence determinants in the 5' UTR make significant RNA-RNA and Protein-RNA interactions that control the rate of ribosome assembly and translation initiation. After initiation, the coding sequence of the mRNA transcript also plays a minor, but sometimes significant, role in the rate of translation elongation. In addition, proteins or RNA molecules can bind to the 5' UTR or coding regions of the mRNA transcript to further modulate the rate of translation. These protens and RNA molecules are called translation factors.

### mRNA Secondary Structures and mRNA:rRNA Interactions

An mRNA transcript is produced as single stranded RNA, but can use RNA:RNA Watson-Crick base pairing to form RNA duplexes. Unlike double stranded DNA, however, double stranded RNA is more forgiving of bulges, hairpins, and other mismatched base pairing. Consequently, strong RNA:RNA interactions can readily associate in solution with as few as four ribonucleotides and can extend to large stable secondary structures. Importantly, unlike protein structure, it is possible to calculate putative RNA secondary structures using a variety of modern computational methods [89]. For our purposes, we use the successor to Mfold, called UNAfold [1], to calculate the minimum free energy (MFE) of an RNA secondary structure or RNA duplex.

The Protein-RNA and RNA-RNA interactions between the 5' UTR of the mRNA transcript and the 30S ribosome subunit are the primary mechanism by which the ribosome binds to the 5' end of the mRNA transcript and initiates translation. The 30S subunit binds to the mRNA transcript at the ribosome binding site (RBS). Its footprint includes either a U rich sequence tract or the Shine-Dalgarno (SD) sequence, which is UAAGGAGG, followed by a minimum spacer of about 5 bp (measured from the end of the SD sequence), the initiation codon AUG, and about 5-10 additional base pairs (called the downstream box) [90]. The U rich sequence binds to the ribosomal S1 protein [63] while the SD sequence forms an RNA:RNA duplex with a complementary sequence located in the 3' end of the 16S rRNA [62]. Together with the initiator codon, these interactions mediate the assembly of the ribosome at the translation start site. Deviations from the SD or U rich sequences or nonoptimal spacing can result in the reduction of the basal rate of translation initiation.

However, additional RNA:RNA interactions that sequester the U rich or SD sequences will repress translation initiation. The most common example is the formation of an mRNA secondary structure within the ribosome binding

Table 1.4: A selected list of ribosome binding sites (RBSs). The DNA sequences starting with AGGA and ending with a start codon are shown. The sequences are qualitatively ranked by their translation efficiency (average proteins per mRNA transcript with all other determinants equal) with one being the most efficient. The Gibbs free energies of the mRNA folding into a secondary structure ($\Delta G_{folding}$) and the rRNA:mRNA hybridization ($\Delta G_{hybrid}$) are shown for comparison. (Calculating using UNAfold [1])

| RBS Sequence | | Ranking | $\Delta G_{folding}$ kcal/mol | $\Delta G_{hybrid}$ kcal/mol |
|---|---|---|---|---|
| AGGAGGAAAAAA | ATG | 1 | >1.5 | -14.6 |
| AGGAATTTAA | ATG | 2 | 0.1 | -10.3 |
| AGGAAACAGACC | ATG | 3 | -0.2 | -12.6 |
| AGGAAACCGGTTCG | ATG | 4 | -2.8 | -16.0 |
| AGGAAACCGGTTC | ATG | 5 | -2.5 | -13.7 |
| AGGAAACCGGTT | ATG | 6 | -0.7 | -10.7 |
| AGGACGGTTCG | ATG | 7 | -1.3 | -16.1 |
| AGGAAAGGCCTCG | ATG | 8 | -1.8 | -12.6 |
| AGGACGGCCGG | ATG | 9 | -3.1 | -11.2 |

site that occludes the SD sequence from binding to the 16S rRNA. These secondary structures often consist of hairpins of 4 or more nucleotides with Gibbs free energies of about -1 to -3 kcal/mol. As an example, in table 1.4, we present nine different ribosome binding sites (ending at the start codon) and the the Gibbs free energies of the their rRNA:mRNA and mRNA:mRNA interactions. At equilibrium, the mRNA secondary structure will rapidly fold and unfold with the probability of it existing in the folded state proportional to the Boltzmann factor, $\exp\left(-\frac{\Delta G_{folding}}{RT}\right)$. As the mRNA secondary structure grows in stability, it will more frequently exist in the folded state and sequester the SD sequence from binding to the 30S ribosome subunit [91].

Consequently, the overall basal rate of translation initiation arises from the competition between the formation of any mRNA secondary structures and the attractive protein-RNA or rRNA:mRNA interactions. However, the exact formula relating the rate of translational initiation to the sequence of the RBS is still unknown, partially due to the ribosomal S1 protein's ability to bind to U rich sequences and other less quantified variables.

### Translation Factors: Proteins and RNAs

In addition to mRNA secondary structures, both proteins [92] and other RNA molecules [59] can bind to the ribosome binding site and sequester the SD or U rich sequences. Similar to repressing transcription factors, these translation factors can bind and sterically prevent the ribosome from assembling. The most important examples are regulatory RNAs [58] whose complementary or antisense sequence to regions in the ribosome binding site cause them to form strong RNA:RNA duplexes, which includes short-hairpin RNAs and microRNAs. Because viruses often encode their genomic information as double stranded RNA, bacteria will rapidly degrade long RNA:RNA duplexes, resulting in the targetted degradation of the mRNA transcript and the cessation of translation. If the complementary sequence is short enough, the RNA:RNA duplex does not trigger targetted degradation and only sequesters the sequence from other interactions. These regulatory RNAs may be used as sequence-specific translational repressors for an arbitrary RBS sequence.

### Codon Usage, Translational Pausing, and Frameshifting

Once the ribosome initiates translation and begins the elongation process, the coding sequence of the mRNA transcript can alter the rate of elongation through its codon usage or by including long stretches of repeat Us or As.

Because there are 64 possible codons, but only 20 natural amino acids, there are redundant codons that result in the incorporation of the same amino acid into an elongating polypeptide, which has been named the *degenerate genetic code*. However, the rate in which the ribosome translates these redundant codons can vary. Through evolution, the concentration of amino acyl-tRNAs that correspond to more frequently used codons are much higher than the ones that correspond to rare codons. Consequently, the translation elongation rate of a rare codon can be much slower, and can causing translational pausing, because of the scarcity of the needed amino acyl-tRNA. The frequencies of codon usage can therefore be correlated with the rate of their translational elongation. By analyzing the frequency of codons in the genome of an organism, one can construct a codon table that estimates the translational elongation rate of each of their codons. These codon tables are publically available at many web sites (e.g. http://www.kazusa.or.jp/codon/) and differ from organism to organism.

The repeated usage of rare codons can result in longer translational pausing and possibly the dissociation of the ribosome subunits. When inserting a gene from one organism into another, it is common to *codon optimize* the mRNA coding sequence by replacing all rare codons with frequently occurring ones, resulting in an increased rate of translation [93]. However, multi-domain proteins often require translational pausing at their domain borders to allow the individual domains to correctly fold. Therefore, the repeated usage of rare codons at strategic points in the mRNA sequence can serve an important functional role.

Finally, repeated tracts of the same nucleotide followed by a stable secondary structure, such as a hairpin, can cause the ribosome to sometimes skip a nucleotide, causing a frameshift [66]. The frameshift can result in the appearance or disappearance of a downstream stop codon. Bacteria and viruses sometimes use frameshifting to vary the composition of a multi-domain protein, using the stop codon to conditionally or randomly add an additional protein domain.

### 1.2.5 Messenger RNA and Protein Degradation and Dilution

The steady-state concentrations of mRNAs and proteins are determined not only by their production rates via transcription and translation, but also by their degradation and dilution rates. By altering the degradation rate of an mRNA transcript or protein, bacteria can quickly increase or decrease the concentration of an RNA or protein without interfering with its transcription or translation process. The degradation rate of an mRNA transcript or protein has a number of sequence determinants and typically depends on the formation of secondary or tertiary structures that bind to the degradative machinery. By changing these sequence determinants, bacteria add another layer of control to the fine tuned production of mRNA and protein molecules. In addition, other protein and RNA molecules can dynamically regulate the degradation rate of a protein or RNA by binding to it and sequestering its binding site with the degradative machinery or by mediating stronger interactions between the degradative machinery and the target, resulting in increased targetted degradation.

However, when bacteria are in their exponential phase of growth, the effects of cell division can play a greater role in determining the steady-state concentration of mRNA transcripts and proteins. The rate at which bacteria replicate and distribute their cytoplasmic contents to daughter cells –called the dilution rate –is, of course, determined by the cell growth rate. The cell growth rate can be affected by a variety of different conditions, including nutrient and oxygen availability, the presence of antibiotics or other toxic chemicals, and the cell surface contact (ie. whether the bacteria are growing in solution or adhered to an agar plate). The dilution rate is often higher than the degradation rate of bacterial proteins, making it a key determinant of its steady-state concentration. Of course, the dilution rate is an approximation; cell replication is a highly orchestrated cascade of events that includes DNA replication, the separation of the bacterial chromosome, and the invagination of the plasma membrane to create two separate daughter cells. A more accurate description of the process of cell replication is as a random discrete event (discussed below).

### RNAses and the RNA Degradosome

The degradation of an mRNA transcript involves the coordinated enzymatic activities of a variety of endo and exonucleases and associated auxiliary proteins [94, 95]. Many of these proteins form a macromolecular complex, called the RNA degradosome. In general, an RNA endonuclease can attack and cut the mRNA transcript in the middle of the transcript while an RNA exonuclease binds to the tri- or mono-phosphate ends of a transcript and sequentially degrades its ends. In bacteria, the only RNAse exonucleases that have been found are the ones that bind to mono-phosphate or poly-adenylated 3' ends and degrade in a 3' $\rightarrow$ 5' direction. Endonucleases and exonucleases often coordinate their degradative activities in a “cut and chew” mechanism, where the endonuclease creates new 3' mono-phosphate ends followed by the repeated degradation of the 3' end until the exonuclease either runs out of transcript or is blocked by an mRNA secondary structure.

Importantly, the first nucleolytic attack on the mRNA transcript is typically the rate-limiting step. Any ribosome that encounters the cut will simply fall off, resulting in an unproductive translation product. However, any unaffected coding region with an intact ribosome binding site can rebind to a ribosome and lead to the productive translation of that coding region. Consequently, mRNA transcripts with multiple start sites and coding regions can be degraded in a location-specific mechanism that results in each coding region having a different effective half-life, leading to varying rates of protein production. The variation in protein production among coding regions of an operon's mRNA transcript can be used to balance their steady-state concentrations in stoichiometric proportions, which is often needed when protein subunits bind together to form multimeric complexes or when optimizing a metabolic pathway [96].

In *Escherichia coli*, the degradosome is composed of RNAse E, PNPase, and an auxilary helicase and enolase. In addition, RNAse II and III and PAP1 all play important roles in the regulated degradation of mRNA transcripts. RNAse E is an endonuclease that cuts single stranded RNA at AU rich sequences and with greater efficiency when it binds near the 5' end. The exact sequence or structural specificity of RNAse E is still uncertain . RNAse III is also an endonuclease, but binds to a double stranded RNA secondary structure called the proximal and distal boxes [97]. Other mRNA secondary structures can sequester the RNAse E and III binding sites and reduce the endonucleolytic rate of attack. Finally, the exonucleases RNAse II, PNPase, and PAP1 work together to polyadenylate the 3' ends of an mRNA transcript and sequentially degrade it in a 3' → 5' direction.

The actions of these enzymes can be blocked or enhanced by the presence of a stable mRNA secondary structure, by the binding of a complementary regulatory RNA molecule [98], or by the presence of an elongating ribosome. The 3' UTR ends of mRNA transcripts often have small secondary structures that prevent their rapid degradation by exonucleases. RNAse binding sites can also become transiently occluded by the ribosome itself, especially in efficiently translated mRNA transcripts or at the ribosome binding site [99].

### Peptide Tags and the Proteosome

The degradation of proteins in the cytoplasm requires the coordinated actions of multiple protease and substrate-binding proteins, collectively referred to as the proteosome. In bacteria, there four different families of proteins that each make up a proteosome, named ClpAP/XP, ClpYQ (HslUV), Lon, and FtsH, with each consisting of an ATPase and proteolytic domain or subunit [100, 101]. The protease subunits form a large multimeric macromolecular complex with numerous proteolytic active sites surrounding a large chamber. The ATPase domain is responsible for binding to protein targets, denaturing them into a disordered state, and translocating them inside the chamber in an ATP-dependent reaction. Once the targeted protein is inside the proteolytic chamber it is rapidly degraded into small peptides.

If a bacterial protein contains a binding site for the ATPase domain in a proteosome it will become more rapidly targeted for degradation. These binding sites are often short peptide sequences near the C- or N-terminals, enabling their easy accessibility. For example, the ssrA tag is a 11 amino acid peptide that, when fused to a protein in the cytoplasm, can bind to any of the ClpXP, ClpAP, or FtsH proteosomes to become more rapidly degraded [102]. The presence of the ssrA tag reduces the half-life of a protein by about 10 fold. Bacteria use the ssrA tag to rapidly degrade partially translated protein products whose ribosomes have stalled. The ssrA RNA, a tRNA-like RNA molecule, binds to stalled ribosomes and substitutes the RNA sequence for the ssrA tag into the stalled ribosome machinery, enabling it to finish the translation of the protein, but also including the ssrA peptide tag at the C-terminal end.

The half-lives of proteins may also be regulated by producing an adaptor protein whose purpose is to mediate stonger interactions between a targetted protein and a proteosome [100]. By dynamically producing these adaptor proteins in response to environmental or metabolic stimuli, a proteosome can more rapidly bind to and degrade a specific protein and, consequently, reduce its steady-state concentration in the cell. The classic example is the general stress response in *E. coli* where a recognition factor RssB binds to the sigma factor $\sigma^S$ for preferential degradation by the ClpXP proteosome.

## 1.3 The Modeling of Gene Networks

The goal of modeling a system of regulated genes, called a gene network, is two-fold:

1. to accurately capture the dynamics of the regulated production of RNA and protein molecules
2. to connect experimental modifications of the molecular interactions in the system to changes in the kinetic constants of the model and, consequently, the system level behavior of the model

Our goal does not require us to model each and every molecular interaction and protein conformational change in the system. We must make some simplifying assumptions, but our reason for doing so is not to make the resulting mathematical equations more readily solvable. Instead, we only make simplifying assumptions when modern techniques may not experimentally probe the difference between series of interactions, such as in the rapid and coordinated conformational changes of a large protein. Consequently, the results of these types of models are directly applicable and comparable to experimental results, which is our main goal.

We model gene networks by breaking down their protein-DNA, protein-RNA, protein-protein, and RNA-RNA molecular interactions into a system of chemical and biochemical reactions with *mass action rate laws* and whose kinetic parameters or thermodynamic free energies are largely empirically measured and taken from the literature. We treat the creation or destruction of stable non-covalent bonds, such as the formation of a protein:DNA complex, as a common biochemical reaction.

It is important to note that our model is created like an *algorithm*. Given a sequence of DNA with defined and characterized genetic components, the system of chemical reactions can be written down according to well defined set of rules, similar to a program. For a complicated gene network, the model-generating algorithm remains the same; it is applied to each and every molecular interaction in each gene in the network, including the regulatory steps in transcription, translation, mRNA and protein degradation, and enzymatic reactions that often play regulatory roles (e.g. phosphorylation or methylation). Consequently, we can detail the system of reactions for a *generalized* gene regulated by transcription or translation factors and then apply that same algorithm to all genes in the network.

The system of reactions also includes unique species for each of the DNA operators, promoters, ribosome binding sites, and other distinct regions in DNA or RNA molecules in the system. This allows us to explicitly write down the Protein-DNA, Protein-RNA, and RNA-RNA regulatory interactions between distinct binding sites without resorting to the usage of Hill kinetics and other nonsensical rate laws with associated fudge factors. Importantly, the molecular interactions between a transcription or translation factor and its DNA or RNA binding site are *context-free*; when the binding site is moved to a different location in the sequence the kinetics of the molecular interaction remain unchanged. Any cooperative binding between transcription or translation factors is explicitly included by creating attractive interactions between adjacently binding factors.

### 1.3.1 Kinetics and Equilibrium Data

Before describing the reactions in the model, we first discuss the relationships between kinetic, equilibrium, and thermodynamic data, their empirical measurement, and some necessary approximations. Every reaction in the model has a corresponding kinetic rate that describes the rate of association of its reactant molecules and the formation or destruction of any covalent bonds or stable non-covalent interactions. For biological systems, the numerous degrees of freedom of a pair of reactant molecules necessitates a largely empirical measurement of their kinetics. Consequently, it is important to only include reactions in the model whose kinetics or equilibria data may be feasibly measured.

Consider a bi-molecular reaction with a forward and backward kinetic constant, $k^f$ and $k^b$. The pair of kinetic constants also define an equilibrium disassociation constant, $K_d = k^b/k^f$, and an equilibrium association constant,

$K_a = 1/K_d$. The equilibrium constants can also be related to the Gibbs free energy of binding or reaction, $\Delta G$, between the two molecules via

$$K_d = \exp\left(\frac{\Delta G}{RT}\right) \tag{1.1}$$

and

$$K_a = \exp\left(\frac{-\Delta G}{RT}\right) \tag{1.2}$$

Using a variety of experimental techniques [103, 104], including surface plasmon resonance, electromobility gel shift assays, and fluorescence-based tagging, the kinetic and equilibrium constants of a protein binding to DNA or RNA, including Holoenzyme formation on promoter DNA and transcription factor interactions with their operator sites, can be empirically measured. Because the equilibrium constant and Gibbs free energies depend only on the relative concentrations of the bound and unbound state of the DNA or RNA molecule, it is more easily measured. Consequently, equilibrium data is more commonly available than kinetic data.

In situations where only equilibrium data is available, one can assume that the forward rate of binding of a large protein to its DNA or RNA binding site is diffusion-limited [105], use the size of the protein to calculate its forward binding kinetic, and then use the equilibrium data to calculate the backward binding kinetic. To do this, we use a first principles description of two particles diffusing in a three dimensional space. The Smoluchowski rate for a diffusion-limited reaction is

$$k^f = 4\pi D a \tag{1.3}$$

with a diffusion coefficient, $D$, and target size, $a$. The diffusion coefficient of a free particle of diameter $d$ in a homogeneous fluid undergoing a random three-dimensional walk may be calculated using Einstein's relation

$$D = \frac{k_b T}{3\pi\eta d} \tag{1.4}$$

Plugging in Eq. 1.4 into Eq. (1.3), we obtain the kinetic rate of association in terms of temperature, $T$, the viscosity of the fluid, $\eta$, and the ratio between the diameter of the protein and its target site, $a/d$, so that

$$k^f = T\,\frac{a}{d}\,\frac{4k_b}{3\eta} \tag{1.5}$$

If we assume that fluid is aqueous and the temperature is physiological so that $\eta = 1 \times 10^{-3}$ Pa s and $T = 30°C$, then the association rate is $k^f = 3.3590 \times 10^9 \frac{a}{d}$ in units of [M s]$^{-1}$. The size of the protein will typically be much larger than the size of the DNA binding site and so we may assume that $a/d \approx 0.1$, leading to an approximate association constant of $k^f \approx 10^8$ [M s]$^{-1}$. Consequently, if we lack kinetic data, but have equilibria data, then we may approximate the backward kinetic constant as $k^b = 10^8 \times K_d$.

Of course, these approximations ignore any electrostatic interactions between the protein and its target site and also any sliding of the protein along the DNA or RNA molecule that reduces the dimensionality of the random walk. These additional interactions can increase the diffusion-limited rate of association. Conversely, if the protein must bind to its target size in a specific conformation, then the association rate will accordingly decrease. We now review the systems of reactions models the each of the regulated processes in gene expression.

### 1.3.2 Regulated Transcription

The production of mRNA transcripts via transcription can be broken down into the basal transcription processes of transcriptional initiation, elongation, and termination and the regulatory interactions that modulate the rates of each these steps. The basal transcription process is modeled using five chemical reactions. For *E. coli*, we use four free kinetic parameters for the basal transcription process, controlling Holoenzyme formation and the closed-to-open conformational change. The remaining parameters have been experimentally determined. The number of chemical reactions that model the transcription factor binding interactions will, of course, depend on the number of transcription factors and DNA operators. We present a generalized reaction scheme for an arbitrary number of activating or repressing transcription factors.

### The Basal Transcription Process

The formation of the Holoenzyme complex on the promoter is modeled by a pair of reversible reactions. It is possible that either the σ-factor binds first to the promoter followed by RNA polymerase recruitment or that the σ-factor binds first to RNA polymerase to form the complex and then to promoter DNA. It is also possible that both events occur. Consequently, we assume that the ordering of these events is insigificant and that a single complex, called RNAPσ-factor, binds and unbinds to the promoter DNA in the following reaction

$$\text{RNAP-}\sigma + \text{Promoter} \rightleftarrows \text{RNAP:P}$$

with forward and reverse kinetic constants, $k^f_{RNAP}$ and $k^b_{RNAP}$. These kinetic constants will depend on the sequence determinants in the promoter DNA and the availability of the active form of the σ factor.

If there are adjacent operators near to or within the promoter, then they must also be included in the pair of reactions. These operators are represented as unique species, but are part of the same contiguous DNA molecule that includes the promoter and coding DNA sequences. By representing them as unique species, we can explicitly write down the regulatory interactions that occur when each is bound by a transcription factor. Importantly, even though there are more than two reactant "species" in the reaction, the rate law is still $2^{nd}$ order bi-molecular; the additional species can not individually diffuse in space. For example, if there are two overlapping operators within the promoter region, labeled $O_1$ and $O_2$, so that the RNA-σ complex can only bind to the promoter when these operators are unbound, then the pair of reactions is

$$\text{RNAP-}\sigma + \text{Promoter} + O_1 + O_2 \rightleftarrows \text{RNAP:P:}O_1\text{:}O_2$$

with a $2^{nd}$ order bi-molecular rate law and kinetic constant. The operators may bind to repressor transcription factors whose steric hindrance prevents the RNA polymerase from binding.

Conversely, if an adjacent (typically upstream) operator, $O_A$, binds to an activator transcription factor, $\text{TF}_A$, with attractive interactions towards the

RNA polymerase, the pair of reactions would be

$$\text{RNAP-}\sigma + \text{Promoter} + \text{TF}_A\text{:O}_1 \rightleftarrows \text{RNAP:P:TF}_A\text{:O}_1$$

with an increased $k^f_{RNAP}$ or a decreased $k^b_{RNAP}$ quantifying the attractive interactions that either recruit the RNA polymerase-σ complex or stabilize its assembly on the promoter.

Once the Holoenzyme complex has assembled on the promoter it must undergo the closed-to-open conformational change, unwinding the DNA and preparing for promoter escape. Because there are numerous steps in this process which are individually unprobable by quantitative experimentation, we assume that the conformational change is a single first order step

$$\text{RNAP:P} \rightarrow \text{RNAP*:P}$$

with a well-defined kinetic constant, $k_{init}$. If the Holoenzyme formation involved the presence of any DNA operators then they must be carried over into future reactions. For example, for the case of two overlapping operators, the conformational change must be

$$\text{RNAP:P:O}_1\text{:O}_2 \rightarrow \text{RNAP:P*:O}_1\text{:O}_2$$

After the conformational change, the Holoenzyme complex must escape the promoter region by translocating forward. In the process, the interactions between the σ factor and RNA polymerase become weaker and the σ factor may disassociate completely, but we do not track the release of σ factor. The promoter escape step is modeled as a first order reaction

$$\text{RNAP*:P} \rightarrow \text{RNAP:DNA}_1 + \text{P}$$

with a kinetic constant $k_{escape}$. In the escape reaction, the promoter "species" is regenerated, allowing another Holoenzyme to assemble there. Likewise, if there are any operators involved in the assembly reaction, they must also be regenerated. Using the same example with two overlapping operators, the promoter escape step is

$$\text{RNAP:P*:O}_1\text{:O}_2 \rightarrow \text{RNAP:DNA}_1 + \text{P} + \text{O}_1 + \text{O}_2$$

After the RNA polymerase escapes the promoter region it will begin the process of transcriptional elongation. The RNA polymerase will translocate forward and transcribe DNA to produce an mRNA transcript with a partially sequence-dependent rate, $k_{elong}$. This may be modeled by a series of $N$ first order reactions,

$$\text{RNAP:DNA}_1 \rightarrow \text{RNAP:DNA}_2 \rightarrow \ldots \rightarrow \text{RNAP:DNA}_{N-1} \rightarrow \text{RNAP + mRNA}$$

where $N$ is the number of nucleotides in the DNA coding sequence. Because $N$ is typically a large number (200 - 1000 base pairs), it becomes impractical to include each translocation step in transcriptional elongation. However, with large $N$, the delayed production of mRNA transcripts becomes important. Consequently, we need a way to more practically model transcriptional elongation while retaining the delay.

We note that, when using stochastic chemical kinetics, the reaction times of each reaction are an exponentially distributed random number. Importantly, if a series of events each has an exponentially distributed waiting time, then the amount of total time to go from beginning to end is *gamma distributed* [106]. Therefore, by assuming that the rate of elongation, $k_{elong}$, is *sequence-independent*, we can collapse the $N$ first order reactions into a single first order reaction

$$\text{RNAP:DNA}_1 \rightarrow \text{RNAP + mRNA}$$

that is $\gamma$-distributed with a single kinetic rate, $k_{elong}$, and a number of steps, $N$. The average delay time of this reaction is simply $N/k_{elong}$ while the variance of the delay time is $N/k_{elong}^2$. The rate of transcriptional elongation also depends on the growth rate of the cell [107] and varies between 30 and 70 nucleotides per second.

### Transcription Factors

The binding and unbinding of the $i^{th}$ transcription factor to its operator DNA site, $\text{O}_i$, are represented by a pair of reactions

$$\text{TF}_i + \text{O}_i \rightleftarrows \text{TF}_i\text{:O}_i$$

with forward and backward kinetic constants, $k^{f}_{TF_i}$ and $k^{b}_{TF_i}$. Additional pairs of reactions are created for each operator site in the system. If two operator sites bind to the same transcription factor then there will be two pairs of reactions.

If the transcription factor is a repressor and binds to an operator that overlaps with the promoter, then its sole regulatory effect is to prevent the RNA polymerase from binding. This is modeled by requiring that the overlapping operator take place in the initial Holoenzyme assembly reaction. If a repressor transcription factor is bound to the overlapping operator then the assembly reaction can not take place. Consequently, no additional reactions are require to model the repression of transcriptional initiation by a bound overlapping operator. Conversely, if the transcription factor is an activator, then (as shown above) pairs of additional reactions must be created to quantify the attractive interactions between a bound activator and RNA polymerase.

Cooperative binding between adjacently binding transcription factors may be modeled by treating the two transcription factors as binding to both operators and then modifying the kinetic constants when one transcription factor has already bound. The first two pairs of reactions model either transcription factor binding to the operators with base kinetic constants, $k^{f}_{TF_i}$ and $k^{b}_{TF_i}$,

$$\mathrm{TF_1 + O_1 + O_2 \rightleftarrows TF_1{:}O_1{:}O_2}$$
$$\mathrm{TF_2 + O_1 + O_2 \rightleftarrows O_1{:}TF_2{:}O_2}$$

and the second pair of reactions models the cooperative (or anti-cooperative) binding of the second transcription factor with either increased (or decreased) kinetic constants, $k^{coop}_{TF_i}$ and $k^{coop}_{TF_i}$.

$$\mathrm{TF_2 + TF_1{:}O_1{:}O_2 \rightleftarrows TF_1{:}O_1{:}TF_2O_2}$$
$$\mathrm{TF_1 + O_1{:}TF_2{:}O_2 \rightleftarrows TF_1O_1{:}TF_2{:}O_2}$$

If there are a large number of cooperatively binding transcription factors, the number of unique species and reactions can become quite large, especially if the transcription factors each possess interactions with RNA polymerase. In these cases, it is sometimes more feasible to switch to an equilibrium description of protein-DNA interactions at the promoter by using the chemical partition function.

### 1.3.3 The Chemical Partition Function and Equilibrium Holoenzyme Formation

The chemical partition function can be used to calculate the probability of finding the promoter at one of its regulatory states while the system is at chemical equilibrium [108]. It assumes that all promoter-DNA interactions between transcription factors, the Holoenzyme, and other trans- proteins are at equilibrium with their operator, promoter, and other DNA sites, respectively. Because the mathematical description uses only equilibrium data about the interactions, such as their Gibbs free energies, its application is more feasible when information about the protein-DNA interactions is limited or if the number of regulatory states is too large, making a full kinetic description unwieldly. The regulatory states of the promoter are all possible combinations of each DNA binding site being either bound or unbound, including all possible ways that two, three, or more proteins can simultaneously bind to the promoter DNA.

While this number can be quite large for a simple system, the resulting answer is an *algebraic* equation describing the probability of each of these states. Here, we use the chemical partition function to describe only the formation of the Holoenzyme on the promoter. By summing together the probabilities of all of states in which the Holoenzyme has successfully assembled on the promoter, one may calculate the probability that the promoter is in a "transcriptionally ready state". Other processes in gene expression, including the closed-to-open conformational change and transcriptional and translation elongation, are non-equilibrium processes; assuming their interactions are at chemical equilibrium would be invalid.

In order to apply the chemical partition function, one follows these steps:

1. Enumerate all of the promoter-DNA interactions in the system with their changes in Gibbs free energies, $\Delta G_{binding}$.

2. Enumerate all possible regulatory states in which the promoter may exist, including the reference state, $i = 1, \ldots, \mathcal{S}$.

Table 1.5: The enumeration and Gibbs free energies of the regulatory states of an example promoter with two operators and a single transcription factor. $N_A$: Avagadro's number. $V$: Volume.

| # | P | $O_1$ | $O_2$ | $\Delta G_i^{tot}$ | $h_i$ |
|---|---|---|---|---|---|
| 1 | — | — | — | 0 | 1 |
| 2 | RNAP | — | — | $\Delta G_{RNAP}$ | [RNAP] |
| 3 | — | $TF_1$ | — | $\Delta G_{TF}$ | $[TF_1]$ |
| 4 | — | — | $TF_1$ | $\Delta G_{TF}$ | $[TF_1]$ |
| 5 | RNAP | — | $TF_1$ | $\Delta G_{TF} + \Delta G_{RNAP} + \Delta G_{TF:RNAP}$ | $[TF_1] \times [RNAP]$ |
| 6 | RNAP | $TF_1$ | — | $\Delta G_{TF} + \Delta G_{RNAP} + \Delta G_{TF:RNAP}$ | $[TF_1] \times [RNAP]$ |
| 7 | RNAP | $TF_1$ | $TF_1$ | $2\Delta G_{TF} + \Delta G_{RNAP} + 2\Delta G_{TF:RNAP}$ | $\frac{1}{2}[TF_1] \times ([TF_1] - \frac{1}{N_A V}) \times [RNAP]$ |
| 8 | — | — | — | $\Delta G_{TF:NSDNA}$ | $[TF_1]$ |

3. Compared to the reference state, list the total change in Gibbs free energies, $\Delta G_i^{tot}$, for each regulatory state in terms of the $\Delta G$ of the participating protein-DNA interactions.

4. Determine the number of identical ways the $i^{th}$ unique regulatory state may assemble by calculating its combinatorial factor, $h_i$.

5. Invoke the ergodic theorem in terms of a canonical partition function: At equilibrium, the probability of each states is proportional to its combinatorial factor multiplied by the Boltzmann factor or, $P_i \propto h_i \exp\left(\frac{-\Delta G_i^{tot}}{RT}\right)$.

6. Determine the probability of the promoter being in a "transcriptionally ready state" by summing up the probabilities of the states in which the Holoenzyme is assembled.

For example, consider a promoter containing two adjacent operator sites, $O_1$ and $O_2$, within the promoter region, P. The two operator sites each bind to the same transcription factor, $TF_1$, with a $\Delta G_{binding} = \Delta G_{TF}$. The transcription factor also has a binding affinity to non-specific DNA with a $\Delta G_{binding} = \Delta G_{TF:NSDNA}$. The Holoenzyme complex binds to the promoter region with a $\Delta G_{binding} = \Delta G_{RNAP}$. When both the Holoenzyme complex and transcription factor are bound, they interact with one another with a $\Delta G_{binding} = \Delta G_{TF:RNAP}$, which may be either positive or negative for repulsive or attractive interactions, respectively. In this example, there are a total of eight unique regulatory states, including the reference state and a psuedo-state representing all of the non-specific binding of the transcription factor to genomic DNA. In Table 1.5, we list the regulatory states of this example promoter, their change in Gibbs free energies, and their combinatorial factors. Because the proteins diffuse in a free solution of volume $V$, the combinatorial factor has units of concentration, making it more of a “density” of states.

The probability of the promoter existing in its $i^{th}$ regulatory state existing is

$$P_i = \frac{h_i \exp\left(-\frac{\Delta G_i^{tot}}{RT}\right)}{\sum_{i \in \mathcal{S}} h_i \exp\left(-\frac{\Delta G_i^{tot}}{RT}\right)} \tag{1.6}$$

The probability of the promoter being in a “transcriptionally ready state”, $P_{init}$, is simply the sum of the probabilities in which the Holoenzyme complex is bound to the promoter, which includes states 2, 5, 6, and 7, or

$$P_{init} = P_2 + P_5 + P_6 + P_7 \tag{1.7}$$

The rate of the Holoenzyme complex undergoing the closed-to-open conformational change can be calculated by using

$$r = k_{init} \times P_{init} \tag{1.8}$$

The remaining steps in the basal transcriptional and translation process are described using a kinetic representation and proceed as normal. By using the chemical partition function, however, a promoter with numerous regulatory interactions may be practically modeled. The equilibrium assumption is more-or-less valid under most kinetic regimes.

### 1.3.4 Regulated Translation

Like transcription, the regulated process of translation can be modeled by first describing the basal translation process and then adding the regulatory interactions that alter its rates.

## The Basal Translation Process

The 30S and 50S subunits of the ribosome first bind and assemble onto the ribosome binding site (RBS) on the mRNA transcript. After the ribosome has assembled, it begins the elongation process by translocation forward and freeing up the RBS. This is modeled by two irreversible reactions

$$\text{Ribosome + mRNA}_{RBS} \rightarrow \text{Rib:mRNA}_{RBS}$$
$$\text{Rib:mRNA}_{RBS} \rightarrow \text{mRNA}_{RBS} + \text{Rib:mRNA}_1$$

with a kinetic constant for translation initiation, $k^{sl}_{init}$, and a kinetic constant for translation elongation, $k^{sl}_{elong}$. Similar to transcriptional elongation, the process of translational elongation may be modeled by a series of $A$ first order reactions, where $A$ is the number of codons in the coding sequence of the mRNA transcript. Because $A$ is typically large (67 - 667 codons), we may describe translational elongation as a single $\gamma$-distributed reaction

$$\text{Rib:mRNA}_1 \rightarrow \text{Ribosome + Protein}$$

with a sequence-independent kinetic constant $k^{sl}_{init}$.

### Translation Factors and mRNA Secondary Structures

Any mRNA secondary structures that sequester the ribosome binding site from accessibility will decrease the effective rate of translation initiation. Consequently, we can model the formation of these secondary structures by a pair of reversible reactions -mRNA secondary structure

$$\text{mRNA}_{RBS} \rightleftarrows \text{mRNA}^{folded}_{RBS}$$

whose change in Gibbs free energy may be determined by using an RNA secondary structure calculator.

Besides secondary structures, any other RNA or protein that binds to the ribosome binding site can also sequester its accessibility. For each of these non-coding RNAs (ncRNA) or translation factor proteins (SLF), we create a pair of reversible reactions

$$\text{ncRNA + mRNA}_{RBS} \rightleftarrows \text{ncRNA:mRNA}_{RBS}$$
$$\text{SLF + mRNA}_{RBS} \rightleftarrows \text{SLF:mRNA}_{RBS}$$

with appropriate forward and backward kinetic constants.

### 1.3.5 mRNA and Protein Degradation and Dilution

All molecules in the cell eventually become degraded or diluted with cell growth. The degradation of molecules involves the active participation of an enzymatic reaction, such as the RNA Degradosome or a proteosome. In addition, all molecular species in the cell become diluted with cell growth and division. We treat these two processes independently. In addition, since we have not included the changing production rates of RNA polymerase or ribosome, we also do not include their degradation or cell dilution.

We model the basal degradation rates of RNA molecules in the system using first order reactions, such as

$$\begin{gathered} \text{mRNA}_{RBS} \longrightarrow \emptyset \\ \text{ncRNA} \longrightarrow \emptyset \end{gathered}$$

whose first order kinetic constants can be quantified using half-lives. These reactions assume that the concentration of the RNA Degradosome remains constant. The basal degradation rate is sequence-dependent and is increased by the presence of RNAse binding sites and by the absence of sequestering secondary structures.

Similarly, the basal degradation rate of protein products, transcription factors, and translation factors can be modeled by using first order reactions, such as

$$\begin{gathered} \text{Protein} \longrightarrow \emptyset \\ \text{TF}_i \longrightarrow \emptyset \\ \text{SLF} \longrightarrow \emptyset \end{gathered}$$

with corresponding half-lives. Again, these reactions assume that the concentration of the proteosomes remain constant. The basal degradation rate depends on the presence or absence of peptide tags that bind to a proteosome.

The presence of adaptor proteins often increases the degradation rate of RNA or protein molecules. We can approximately model their effects by treating these adaptor proteins, called a degradation factor (DF), as an enzymatic catalyst for RNA or protein degradation, using the reactions

$$\begin{gathered} \text{DF + RNA} \rightleftarrows \text{DF:RNA} \longrightarrow \text{DF} \\ \text{DF + Protein} \rightleftarrows \text{DF:Protein} \longrightarrow \text{DF} \end{gathered}$$

The degradation factor reversibly binds to the RNA or protein molecule and shuttles it to the RNA Degradosome or proteosome for destruction.

The dilution of RNA and protein species may be modeled using two different approaches: the first approximates cell replication as a continuous process while the second describes it as a discrete event. In the first case, the volume of the bacterial cell remains constant and the dilution rate of each cytoplasmic RNA and protein species is described by a first order reaction, similar to the degradation reaction

$$\begin{gathered}\text{RNA} \rightarrow \emptyset \\ \text{Protein} \rightarrow \emptyset\end{gathered}$$

The kinetic constants of these reactions are all, however, the same: the kinetic rate of cell replication, which is $k_{cr} = \log 2/t_{\text{acr}}$, with an *average* cell replication rate of $t_{\text{acr}}$.

In the second approach, we treat cell replication as a random discrete event that occurs according to a Gaussian distribution. The contents of the cytoplasm may also be distributed to daughter cells according to a Binomial distribution. The volume of the bacterial cell exponentially grows, starting from an initial condition $V_o$, according to $V = V_o \exp\left(k_{cr}\left(t - t_o\right)\right)$, where $k_{cr}$ is the same kinetic constant described above and $t_o$ is the time of the previous cell replication event. The time of the next cell replication event is sampled from a Gaussian distribution where the mean is the average cell replication time, $t_{acr}$, and the standard deviation is empirically measured from experiments and is typically $\approx$ 5-10% of $t_{acr}$. At this time, the number of molecules of each cytoplasmic RNA and protein species can be either halved or sampled from a Binomial distribution, where the number of trials is the number of molecules of each chemical species and the probability of distribution is $1/2$. The volume may be either halved or reset to $V_0$. In a more algorithmic format, this procedure is as follows:

1. Step 1: Starting at an initial time $t_o$ and volume $V_o$, select a Gaussian random variable $t_r \sim \mathrm{N}(t_{acr}, \sqrt{0.1}\, t_{acr})$.

2. Step 2: Between $t_o$ and $t_r$, the volume increases continually according

to $V = V_o \exp(k_{cr}(t - t_o))$.

3. Step 3: At $t = t_r$, the volume is either reset to $V = V_o$ or the volume is reduced by half $V = V/2$. At the same time, the number of molecules of each cytoplasmic chemical species, which is $X_i$, is sampled from the Binomial distribution according to $X_i \sim$ Binomial$(X_i, 0.5)$. The time of the last cell replication is then reset to $t_o = t_r$ and the algorithm repeats.

Using this model, the stochasticity arising from cell replication may also be included.

### 1.3.6 Protein - Protein Interactions

Transcription factors and other regulatory proteins often interact with one another via protein-protein interactions. One common example is the multimerization of a transcription factor to a dimer or tetramer complex, which then binds to its cognate DNA operators. These multimerization reactions may be modeled using -homodimers, tetramers

$$2\ \text{TF}_1 \rightleftarrows \text{TF}_{1,(2)}$$
$$2\ \text{TF}_{1,(2)} \rightleftarrows \text{TF}_{1,(4)}$$

Similarly, if two different transcription factors bind together to form a heterodimer complex, the reactions may be straightforwardly written down as

$$\text{TF}_1 + \text{TF}_2 \rightleftarrows \text{TF}_1\text{:TF}_2$$

Finally, there are a variety of enzymes that covalently modify transcription factors to either activate or deactivate their DNA-binding capabilities. A common example are kinases and phosphatases, which can specifically bind to their target and add or remove a phosphate group. If both a kinase and phosphatase target the same protein for activation and deactivation, it is called a futile enzyme cycle. We model a futile enzyme cycle using two sets of enzymatic reactions

$$\text{Kin} + \text{TF} \rightleftarrows \text{Kin:TF} \rightarrow \text{Kin} + \text{TF}^{(p)}$$
$$\text{Phos} + \text{TF}^{(p)} \rightleftarrows \text{Phos:TF}^{(p)} \rightarrow \text{Phos} + \text{TF}$$

Futile enzyme cycles are often used to activate or deactivate a transcription factor because the concentration of active factor is very sensitive to small changes in the concentration of kinase or phosphatase enzyme, which is referred to as zero-order ultrasensitivity [109].

## Chapter 2

# Stochastic Numerical Methods

### 2.1 Introduction

As engineers and scientists study and manipulate smaller and smaller systems, the mathematics that we use to quantitatively describe these systems must change. As length and time scales decrease, previously valid approximations begin to break down, forcing us to return to a more first principles description of the system. Once we remove the mean-field and continuum approximations that are often used in a number of fields, including material science, fluid mechanics, chemical kinetics, and thermodynamics, we discover that the mathematical representation of these systems becomes probabilistic. Consequently, we must use the theory of stochastic processes to describe the dynamics of these systems and stochastic numerical methods to solve the resulting equations.

In this chapter, we review material on probability theory, stochastic processes, the simulation of jump Markov and Poisson processes, and the numerical integration of stochastic differential equations. The material in this chapter can serve as the skeleton of a curriculum for a graduate-level course on applied stochastic processes and non-equilibrium statistical mechanics.

#### 2.1.1 An Overview of the Chapter

In the first section, we present a brief introduction to probability and stochastic processes, in which we focus on teaching the fundamentals using common sense principles while avoiding the sometimes obscure route through measury theory. We first discuss the Laws of probability theory and their application in describing a special type of mathematical object called a random variable. We then list a number of commonly used random variables, their distributions, and their uses. Finally, we describe a variety of different stochastic processes, from random walks to stochastic differential equations, and use simple arguments to formulate the connections between them. We also describe the Wiener and Poisson processes, which are two especially important types of stochastic processes.

In the second section, we detail the stochastic numerical methods that simulate jump Markov processes and Poisson processes. Specifically, we describe the three original variants of the stochastic simulation algorithm and the $\tau$-leaping family of stochastic numerical methods, including the Binomial leaping methods. We conclude by explaining why these methods are deficient for our purposes.

In the next section, we describe the rigorously derived theory behind the numerical integration of Itô stochastic differential equations. Because we use this theory to construct our first hybrid stochastic method, we delve deeply into its origins. We explain why stochastic integrals are different from their Riemann counterparts and how to numerically generate or approximate their values. We then present the Itô Formula and the Itô-Taylor expansion and use them to derive a number of stochastic numerical methods, including Runge-Kutta style ones. We continue with Implicit-style stochastic numerical methods and adaptive time step schemes.

In the fourth section, we present our first new stochastic numerical: The hybrid jump/continuous Markov process stochastic simulator (HyJCMSS). It dynamically partitions an arbitrary system of chemical reactions into slow/discrete and fast/continuous reaction subnetworks, approximates the fast/continuous subsystem by describing its effects with a chemical Langevin equation, and describes the occurrences of the slow/discrete reactions with a system of differential Jump equations. Because both the chemical Langevin equation and the differential Jump equations are Itô stochastic differential equations, we use a stochastic numerical integrator to simultaneously solve the coupled equations. The solution is computed in much less computational time compared to the stochastic simulation algorithm, especially when the system contains many frequently occurring reactions. We present a rigorous measurement of the stochastic numerical method's accuracy and computational speed up using a variety of examples.

In the fifth section, we present our second newly developed stochastic numerical method: An equation-free probabilistic steady-state approximation (PSSA). It dynamically partitions an arbitrary system of chemical reactions into slow/discrete and fast/discrete reaction subnetworks, detects whether the fast/discrete subnetwork has caused the state of the system to converge to a quasi-steady state marginal distribution, extracts samples from the system at quasi-steady state, and substitutes those samples back into the calculations to dramatically speed up the simulation of both the fast and slow dynamics while retaining accuracy. When the system has a modest to significant separation of time scales, the algorithm's computational running time is much less than the stochastic simulation algorithm. It is important to note that the algorithm retains accuracy in both the fast and slow dynamics, thus sacrificing little to nothing for the computational speed up.

In the final section, we present our open-source software package, Hy3S: Hybrid Stochastic Simulation for Supercomputers. Together with a simple-to-use Matlab-driven graphical user interface, the software simulates the stochastic dynamics of an arbitrary system of chemical reactions using MPI parallelized versions of our HyJCMSS algorithm. The HyJCMSS algorithm is implemented with four different stochastic numerical integrators, including the Euler-Maruyama and Milstein stochastic numerical methods with fixed steps and with an adaptive time step scheme. The software also includes MPI parallelized versions of the Next Reaction variant of the stochastic simulation algorithm. The software package is a fast and scalable computational engine for computing the stochastic dynamics of any chemical reaction network with multiple time scales.

## 2.2 A Brief Introduction to Probability and Stochastic Processes

This section provides a brief review of the concepts and formulae behind probability theory and stochastic processes, using a descriptive approach that aims to prepare the reader for the following sections. We focus on probability, random variables, and stochastic processes while avoiding the overuse of measure theory, which is rarely taught to non-mathematicians. Accordingly, this introduction is not meant to be exhaustive. There are many texts that include similar introductions, including Gillespie's "Markov processes: Introduction to physical scientists" [110], van Kampen's "Stochastic processes in physics and chemistry" [111], and Mircea's "Stochastic Calculus: Applications in Science and Engineering" [112], but we are most indebted to Peter Kloeden and Eckhard Platen's "Numerical solution of stochastic differential equations" [113] for its clarity and breadth.

### 2.2.1 Probability

To begin applying probability theory on a system, all of the possible events and their outcomes are enumerated. An *event* is any combination of outcomes that can be recorded as having occurred or not. The rolling of a 6-sided die is a good example system. When we roll the die the possible outcomes are that either the 1, 2, 3, 4, 5, or 6 face is facing up. These outcomes are denoted by $\omega_1, \omega_2, \omega_3, \omega_4, \omega_5, \omega_6$ and the set of all possible outcomes, $\Omega = \{\omega_1, \omega_2, \omega_3, \omega_4, \omega_5, \omega_6\}$, is called the *sample space*. If the the 6-sided die is tossed $N$ times and the number of times each outcome occurs is $N_i$, then the *relative frequency* of each outcome is $f_i = N_i/N$. For a small number of rolls, this relative frequency will change. However, as the number of rolls is increased, the relative frequency will converge to a constant, called a *probability* $p_i$. The limit $\lim_{N\to\infty} f_i(N) = p_i$ is the probability of the $\omega_i$ outcome and follows certain rules, called the Laws of Probability. For a 6-sided fair die, the probability of each outcome is $p_i = 1/6$ so that each outcome has the same (uniform) probability of occurrence.

The simplest type of event is the elementary one, $A = \{\omega_i\}$, but we will often consider other events that include two or more outcomes. For example, if we are only interested in die rolls that equal 3 or less, then the events $A_1 = \{\omega_1, \omega_2, \omega_3\}$ and its complement, $A_1^c = A_2 = \{\omega_4, \omega_5, \omega_6\}$, can both be studied. Another useful event is "All possible outcomes", which is simply the sample space, $\Omega$. An event may be considered as any possible combination of the elements in the set $\Omega$. The special case $A = \Omega$ has the complement $A^c = \emptyset$, which is the event "Nothing happens".

The Laws of Probability are similar to mass, momentum, or force conservation in that they are a detailed accounting of what can go "in", what can come "out", and how much accumulates. Consequently, there are definite relationships between the probabilities of different occurrences of events. If we are interested in an event, $A$, the probability of its occurrence is $P(A) = \sum_{\omega_i \in A} p_i$. Consequently, the first Law is that "Something Must Always Happen" or

$$P(\Omega) = \sum_{\forall \omega_i \in \Omega} p_i = 1 \tag{2.1}$$

Likewise, the corollary to that Law is that "Nothing Never Happens" or $P(\emptyset) = 0$. In many systems, however, the occurrence of "nothing" is a legitimate outcome and must be included in the sample space. Mathematically, "Nothing" is whatever remains after removing all possible outcomes to an event. Also, since the event $A$ is a subset of outcomes contained in the sample space, $\Omega$, or $A \subseteq \Omega$, then $0 \leq P(A) \leq 1$ must always be true.

The second Law of Probability is that "If an Event Occurs, then its Complement Event Did Not Occur", which yields $P(A) + P(A^c) = 1$. For example, if we only record whether the die roll is even- or odd-valued, then the even-valued event is $A = \{\omega_2, \omega_4, \ldots, \omega_6\}$ with probability $P(A) = 1/2$ and its complement, $A^c$, is the odd-valued event with probability $P(A^c) = 1/2$. The outcome of the die roll, $\omega_i$, lives in either the $A$ or $A^c$ events and, consequently, both events may not simultaneously occur. If we repeat the rolling of the die, the resulting probabilities of each event will reflect the above law's consequences. We may generalize this Law by considering any two events, $A$ and $B$, such that they do not share the same outcome. The application of the second law to the events $A$ and $B$ leads to the "OR = Addition" Law or

$$P(A \bigcup B) = P(A) + P(B) \quad \text{iff } A \bigcap B = \emptyset \tag{2.2}$$

but the two events must be mutually exclusive and not share a similar outcome.

For example, if we are tracking all of the even-valued outcomes of the die roll and also the question "Was the die roll even?" then the events are $A_1 = \{\omega_2\}$, $A_2 = \{\omega_4\}$, $A_3 = \{\omega_6\}$ and $A_4 = \{\omega_2, \omega_4, \omega_6\}$. Consequently, the events are not independent because the outcomes $\omega_2$, $\omega_4$, and $\omega_6$ are all present twice. Logically, $A_1 \subset A_4$ is true and so $P(A_1 \bigcup A_4) \neq P(A_1) + P(A_4)$. This leads us to the next aspect of probability theory: the *conditional probabilities* of events whose outcomes are shared. We say that events $A_1$, $A_2$, and $A_3$ are conditioned on the event $A_4$. Once $A_4$ occurs, we know that the die roll was even-valued. Given this knowledge, the probability of the die roll yielding 2, 4, or 6 is now $1/3$ and not $1/6$. In general, the conditional probability of an event $A$, conditioned on another event $B$ with $P(B) > 0$, is

$$P(A \mid B) = \frac{P(A \bigcap B)}{P(B)} \tag{2.3}$$

Equation (2.2) can be modified to include conditional events by using

$$P(A \bigcup B) = P(A) + P(B) - P(A \mid B) \tag{2.4}$$

If the events $A$ and $B$ are independent (or lacking conditioning), such that $P(A \mid B) = P(A)$ and $P(B \mid A) = P(B)$, then Eq. (2.3) simplifies to the "AND = Multiply" Law, such that

$$P(A \bigcap B) = P(A)\, P(B) \quad \text{iff A and B are independent} \tag{2.5}$$

Equations (2.2) and (2.4) may be used to answer questions about the likelihood of different combinations of outcomes, such as "What is the probability that the die roll is either 3 or 6 for three rolls in a row?". The question is translated into an event, $A_{all}$, and can be broken up into three separate, independent events, $A_{1st}$, $A_{2nd}$, and $A_{3rd}$, each describing the outcome of a single roll, where $A_{1st} = \{\omega_3, \omega_6\}$ and so on for $A_{2nd}$ and $A_{3rd}$. The first event, $A_{1st}$, may occur when the die roll is either 3 *or* 6. By the "OR = Addition" Law, we add the probabilities of rolling a 3 or 6 together so that $P(A_{1st}) = 1/3 = 1/6 + 1/6$. Because each die roll is independent of one another and identically distributed, the probabilities of the second and third events are the same as the first, yielding $P(A_{1st}) = P(A_{2nd}) = P(A_{3rd})$. The final event occurs when $A_{1st}$ *and* $A_{2nd}$ *and* $A_{3rd}$ all occur. Again, because the rolls are independent, the "AND = Multiply" Law is used to multiply their separate probabilities together, yielding $P(A_{all}) = 1/9 = (1/3)^3$.

Using these Laws, we can tie together the sample space $\Omega$, the collection of events $\mathcal{A}$, and all of their probabilities, $\mathcal{P}$, into a triplet $(\Omega, \mathcal{A}, \mathcal{P})$ called a *probability space*, which succinctly summarizes the information in the system. The collection of events, $\mathcal{A}$, are all possible $k, j$ such that

$$\begin{aligned} &\textstyle\bigcup_{i=j}^{i=k} A_i \text{ and } \bigcap_{i=j}^{i=k} A_i \text{ are events} && \text{if } A_1, \ldots, A_n \text{ are events} \\ &P\left(\textstyle\bigcup_{i=j}^{i=k} A_i\right) = \textstyle\sum_{i=j}^{i=k} P(A_i) && \text{if } A_1, \ldots, A_n \text{ are mutually exlusive} \end{aligned} \tag{2.6}$$

The probability space is a formal structure within which mathematicians verify that the events, outcomes, and probabilities are all self-consistent with one another and that the laws of probability remain true.

The idea of a probability space is most needed when we begin to consider events composed of continuous-valued outcomes. If we allow that our die roll produces a real-valued number uniformly between zero and one, $\omega \in [0, 1]$, then the number of possible outcomes suddenly becomes infinite. The outcome $\omega = 0.14621$ is different from the outcome $\omega = 0.14622$ and so on. Because the probabilities of each outcome must add up to one and there are an infinite number of points in $[0, 1]$, then the probability of selecting a single point in $[0, 1]$ is actually zero! We sidestep these technical difficulties by always defining an event as a subinterval in $[0, 1]$, such as $A = [0.146, 0.147]$. The relative *measure* of this subinterval is $0.001 = (0.147 - 0.146)/(1 - 0)$ and so $P(A) = 0.001$. The measure of the subinterval is the relative quantity of probability compared to a total amount of probability occupying a space, which is called a measure space. Because of these technical difficulties, the formal description of probability theory is based in Measure theory, which itself is based on the integration of spaces and domains.

Armed with these Laws of Probability, we may now define a special type of variable whose values are defined on a probability space. These variables are called *random variables* and have a corresponding probability distribution.

### 2.2.2 Random Variables and Probability Distributions

The formal definition of a random variable, $X(\omega)$, is a function that, given some probability space $(\Omega, \mathcal{A}, \mathcal{P})$ will return an event, $A = \{\omega\} \subseteq \mathcal{A}$ and $\omega \in \Omega$, that occurs with probability $P(A)$. The outcome, $\omega$, is often expressed as one or more independent variables, $x$, such as time, a position, a velocity, or a number of successes. The probability of the random variable, $X$, producing a value of $x$ is then a function of $x$, or $P(x)$. The function $P(x)$ over the variable $x$ is called a *probability distribution function* (PDF). Other names include the probability density function, the probability distribution, or more casually, as "the distribution". For more clarity, we may write the PDF as $P(X = x)$ to emphasize that the random variable can take any allowable value of $x$. The allowable values of $x$ determine whether the random variable and its probability distribution are either discrete-valued, $x \in \mathcal{Z}$, or continuous-valued, $x \in \Re$.

If the allowable values of $x$ are discrete-valued, then the first Law of Probability states that the sum of their probabilities must be one, or

$$\sum_{i=-\infty}^{i=\infty} P(X = i) = 1 \tag{2.7}$$

Likewise, if $x$ is continuous valued everywhere then the sum becomes an integral,

$$\int_{-\infty}^{\infty} P(X = x)\, dx = 1 \tag{2.8}$$

Any function $f(x)$ may be a probability distribution function if (a) $f(x) \geq 0$ for all values of $x$ and (b) $P(x) = f(x)$ satisfies either Eqs. (2.7) or (2.8) if $x$ is discrete or continuous valued, respectively. These restrictions say nothing about what the PDF looks like on $x$ and so there are a great many possibilities. The PDF may be a smooth function or it may be discontinuous. It may have multiple peaks or be entirely flat. It may be symmetric or asymmetric. It could be the delta function, $\delta(x-c)$, which is equal to one only when $x = c$.

A few of the most commonly used probability distributions will be reviewed below.

There are other quantities that describe the characteristics of a random variable. Each probability distribution also has a corresponding *cumulative distribution function* (CDF), $F(x)$, which is

$$F(x) = \int_{-\infty}^{x} P(X = x')\, dx' \tag{2.9}$$

Unlike the PDF, the cumulative distribution function is always a continuous, increasing function in $x$. When given some $a$ and $b$ such that $a \leq x \leq b$, then $F(a) \leq F(x) \leq F(b)$ is always true. The *moments* of the probability distribution provide additional information. The $n^{th}$ moment of a continuous-valued PDF around a constant $c$ is

$$M_n(c) = \int_{-\infty}^{\infty} (x-c)^n\, P(X = x)dx \tag{2.10}$$

If $c = 0$ then the first moment is simply the average or mean of the random variable, or

$$\mu = \int_{-\infty}^{\infty} x\, P(X = x)dx \tag{2.11}$$

If $c = \mu$ then we call these the *central moments*. Of course, the first central moment is zero. The second central moment is called the *variance*, $\sigma^2$,

$$\sigma^2 = \int_{-\infty}^{\infty} (x-\mu)^2 P(X = x)dx \tag{2.12}$$

The square root of the variance is called the *standard deviation*. The third and fourth central moments, $\mu_3$ and $\mu_4$ are related to the *skewness* and *kurtosis* of the distribution, which are respectively $\gamma_1 = \mu_3/\sigma^3$ and $\gamma_2 = \mu_4 = \sigma^4$. The skewness measures the lopsidedness or asymmetry of a probability distribution while the kurtosis measures whether the probability distribution is tall and skinny or short and squat. The *excess kurtosis* is often used in practice, which is $\gamma_2^{ex} = \mu_4/\sigma^4 - 3$. If we do not know the PDF of a random variable, but we have collected many *samples* of its values, $X_i\ i = 1 \ldots N$, then we may calculate the mean and variance of the PDF using

$$\mu = \tfrac{1}{N} \textstyle\sum_{i=1}^{N}\, X_i \quad \sigma^2 = \tfrac{1}{N} \textstyle\sum_{i=1}^{N}\, (X_i - \mu)^2 \tag{2.13}$$

Table 2.1: Four important characteristics of five useful probability distribution functions

| Distribution | Mean $\mu$ | Variance $\sigma^2$ | Skewness $\gamma_1$ | Excess Kurtosis $\gamma_2^{ex}$ |
|---|---|---|---|---|
| Uniform, URN$(a,b)$ | $(a+b)/2$ | $(b-a)^2/12$ | 0 | $-6/5$ |
| Exponential, Exp$(\lambda)$ | $\lambda^{-1}$ | $\lambda^{-2}$ | 2 | 6 |
| Poisson, Poisson$(\lambda t)$ , | $\lambda t$ | $\lambda t$ | $(\lambda t)^{-1/2}$ | $(\lambda t)^{-1}$ |
| Gamma, Gamma$(N,\lambda)$ | $N\lambda^{-1}$ | $N\lambda^{-2}$ | $2/\sqrt{N}$ | $6/N$ |
| Gaussian, N$(\mu,\sigma^2)$ | $\mu$ | $\sigma^2$ | 0 | 0 |

Importantly, for Eq. (2.13) to be true, the samples $X_i$ must be independent (ie. not conditioned on each other) and identically distributed according to the same PDF, which is abbreviated as i.i.d. Finally, one may rewrite Eq. (2.13) in terms of an *expectation* value, which is simply $\mu = E(X) =< X >$ and $\sigma^2 = E((X-\mu)^2) =< (X-\mu)^2 >$.

### 2.2.3 Commonly Used Random Variables

There are an unlimited number of different random variables, but some have special notability because of their use in applications, such as in finance or physics. There are currently 68 commonly used probability distributions listed on Wikipedia at http://en.wikipedia.org/wiki/Probability_distribution. Because of their usage in stochastic chemical kinetics, we focus on the uniform distribution, the exponential distribution, the Poisson distribution, the Gamma distribution, and the Gaussian distribution. In Table 2.1, we list the important characteristics of these probability distributions.

### Uniform Probability Distribution Function

Consider a continuous-valued random variable, $X$, which may take any values $a < x < b$, and with a constant probability $P(X = x) = c$. By the Laws of Probability, the probability distribution function, called the uniform distribution, must be

$$P(x) = \begin{cases} 0 & x \leq a \\ \frac{1}{b-a} & a < x < b \\ 0 & x \geq b \end{cases} \tag{2.14}$$

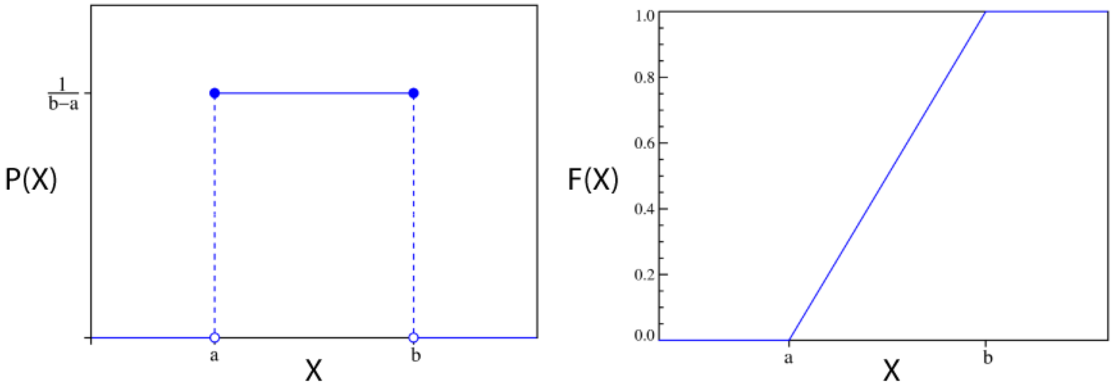

Figure 2.1: The probability distribution function, $P(X)$, and cumulative distribution function, $F(X)$, of a uniform random variable, $X \sim \mathrm{URN}(a,b)$

so that $c = 1/(b-a)$. Likewise, the cumulative distribution function is

$$F(x) = \begin{cases} 0 & x \leq a \\ \frac{x-a}{b-a} & a < x < b \\ 0 & x \geq b \end{cases} \tag{2.15}$$

The PDF and CDF of this random variable is shown in (2.1). A random variable generated with a uniform distribution on $(a,b)$ is denoted by $x \sim$ URN(a,b). If the $a,b$ are omitted then it is assumed that $a = 0$ and $b = 1$.

### Generating Samples of Non-Uniform Random Variables

Uniformly distributed random variables are the basis for the generation of all other computer generated random numbers. In conversation, a “Uniform Random Number” or URN is a uniformly distributed random number on (0,1) so that $0 < x < 1$. Using modern computers and a variety of different numerical algorithms [114, 115], one may generate a stream of uniformly distributed *psuedorandom* numbers whose properties closely mimic a “true” URN. Given such a generator, one may “stretch” the uniform distribution into any other probability distribution functions and generate samples of a random number with another distribution. In general, there are two ways to convert a URN into a random variable with a different probability distribution. The first is called the *inversion method* while the second is called the *rejection method*.

In the inversion method, we equate the cumulative distribution function of the desired random variable, $F(x)$, with a random number $u$ taken from a URN generator. We then solve for $x$ in terms of $u$, yielding

$$F(x) = u \qquad x = F^{-1}(u) \tag{2.16}$$

The cumulative distribution function is often complicated and not invertible. Consequently, the rejection method is typically used. The rejection method uses a helper distribution, $g(x)$ to generate a random variable from the distribution $P(x)$ by performing the following procedure:

1. Choose an appropriate helper distribution $g(x)$ and a bounded limit $M$
2. Generate a sample, $x \sim g(x)$, and another sample $u \sim$ URN
3. If $u < \frac{f(x)}{M\ g(x)}$ then accept $x$ as a valid sample of the distribution $P(x)$
4. Otherwise, reject $x$ and repeat the above step

The helper distribution $g(x)$ is different for each desired probability distribution $P(x)$.

The most common example of an inversion method is the random number generator for the exponential distribution [116]. Examples of the rejection method include the Box-Muller, Polar, and Ziggurat algorithms [113, 116, 117] for generating Gaussian random numbers, a variety of algorithms for generating Gamma distributed random numbers [116], and the Markov Chain Monte Carlo sampling of the Maxwell-Boltzmann distribution using the Metropolis criteria [118]. The memory and computational cost of these algorithms vary and new ones are continuing to be developed.

### Exponential Probability Distribution Function

The exponential distribution function describes the distribution of *times* between events, called waiting times, which occur with a constant rate, denoted by $\lambda$. Its random variable has non-negative, continuous values and is denoted by $x \sim \text{Exp}(\lambda)$. These random events may be the arrival of people, the radioactive decay of atoms, or the collision and reaction of two molecules.

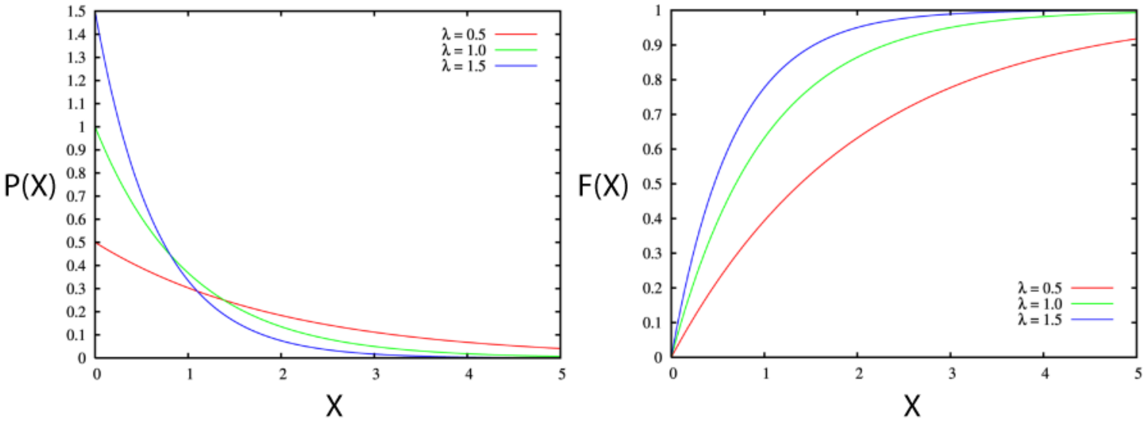

Figure 2.2: The probability distribution function, $P(X)$, and cumulative distribution function, $F(X)$, of an exponentially distributed random variable, $X \sim \text{Exp}(\lambda)$

Its probability and cumulative distribution functions, shown in Figure 2.2 are

$$P(x;\, \lambda) = \begin{cases} \lambda \exp(-\lambda x) & x \geq 0 \\ 0 & x < 0 \end{cases} \tag{2.17}$$

and

$$F(x;\, \lambda) = \begin{cases} 1 - \exp(-\lambda x) & x \geq 0 \\ 0 & x < 0 \end{cases} \tag{2.18}$$

Importantly, the exponential distribution is the only *memoryless* distribution; after having waited 10 seconds for an event to happen, the probability of it occurring in the next 10 seconds is the same as before. More formally, we say that the probability of the arrival time $T$ follows

$$P(T > t + s | T > s) = P(T > t) \quad \text{for all } t, s \geq 0 \tag{2.19}$$

In stochastic chemical kinetics, we use the exponential distribution to model the times in which molecules collide and react with one another.

### Poisson Probability Distribution Function

The Poisson probability distribution describes the distribution of the *number* of times an event occurs within a time interval, $T$, if the event occurs with a constant rate, $\lambda$. Its random variable has non-negative, discrete values and is denoted by $n \sim \text{Poisson}(\lambda t)$. Like the exponential distribution, the Poisson

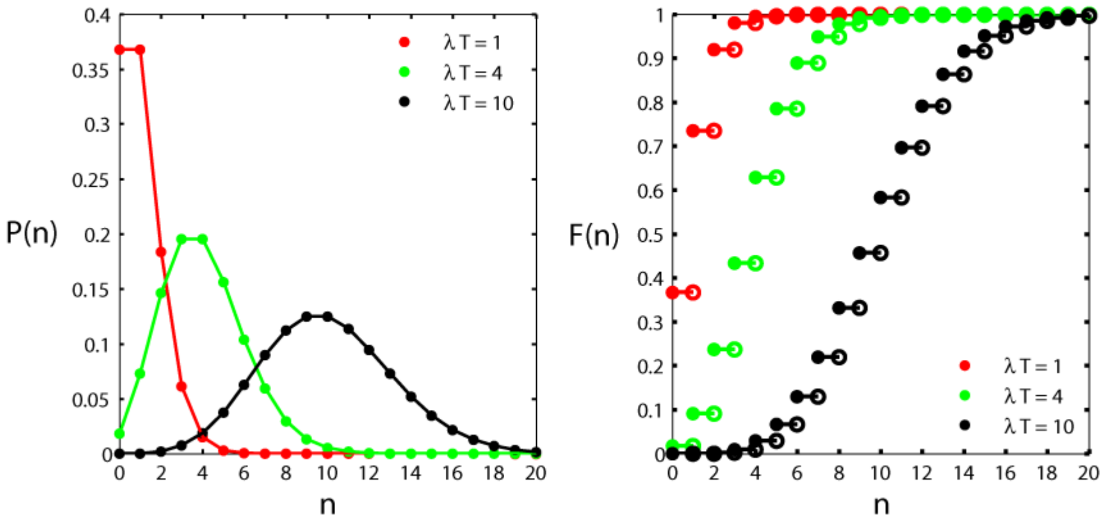

Figure 2.3: The probability distribution function, $P(n)$, and cumulative distribution function, $F(n)$, of a Poisson distributed random variable, $n \sim \text{Poisson}(\lambda\, t)$

process is often used to describe the arrival of people or radioactive decay, but now the random variable is measuring the number, $n$, of people or atoms that have arrived or decayed in a length of time, $T$. Shown in Figure 2.3, the probability and cumulative distribution functions of the Poisson random variable are

$$P(n;\, \lambda) = \begin{cases} \dfrac{\exp\left(-\lambda\, t\right)\ (\lambda\, t)^n}{n!} & x \geq 0 \\ 0 & x \leq 0 \end{cases} \tag{2.20}$$

and

$$F(n;\, \lambda\, t) = \sum_{i=0}^{i=n} P(n;\lambda\, t) \tag{2.21}$$

The Poisson distribution is the starting point for the τ-leaping family of stochastic simulation techniques [119].

### Gamma Probability Distribution Function

The Gamma probability distribution function describes the arrival *times* between an event which consists of a series of many individual events, whose waiting times are each exponentially distributed. For example, consider a Olympic hurdler who needs $x \sim$ Exp( λ ) seconds to jump over each hurdle and there are $N$ hurdles to jump before reaching the finish line. If only the

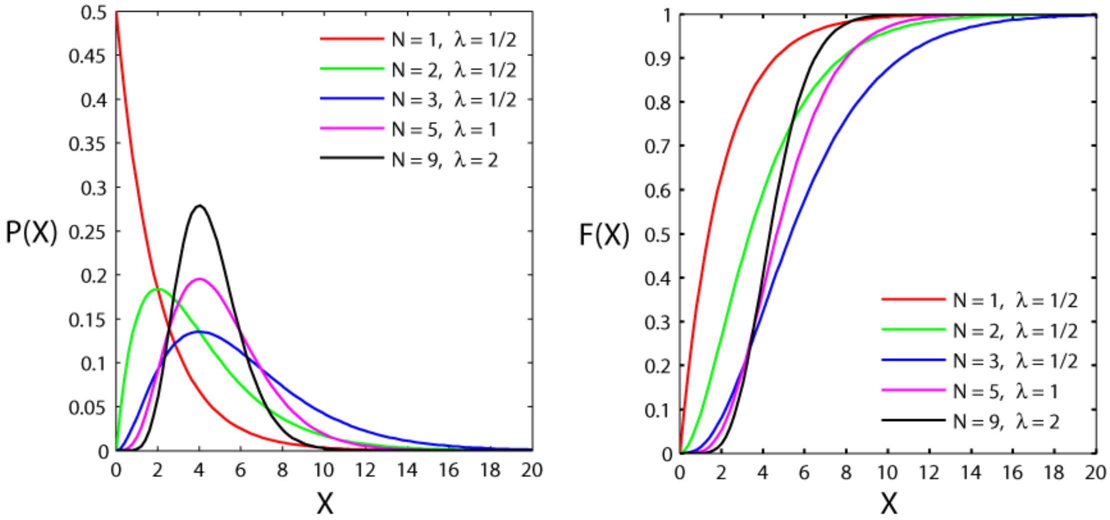

Figure 2.4: The probability distribution function, $P(X)$, and cumulative distribution function, $F(X)$, of a Gamma distributed random variable, $X \sim \text{Gamma}(N, \lambda)$

start and end times are recorded, then the the total race time, $x$, will have a Gamma distribution with rate $\lambda$ and shape parameter $N$. When $N = 1$ the Gamma distribution simplifies to the exponential distribution. A Gamma random variable has non-negative, continuous values and is denoted by $x \sim \text{Gamma}(N, \lambda)$. Shown in Figure 2.4, the probability and cumulative distribution functions of a Gamma distributed random variable are

$$P(x;\ \lambda, N) = x^{N-1}\ \frac{\lambda^N\ \exp(-\lambda x)}{(N-1)!} \quad \text{for } x > 0 \tag{2.22}$$

where $N$ is a positive integer and

$$F(x;\ \lambda, N) = \frac{\gamma(N, \lambda x)}{(N-1)!} \tag{2.23}$$

where $\gamma$ is the lower incomplete gamma function, a complicated function capable of being numerically evaluated. In biology, the arrival times of mRNA transcripts or proteins proceeding through transcriptional or translational elongation, respectively, can be described using the Gamma distribution.

### Gaussian Probability Distribution Function

Due to the Central Limit theorem, the Gaussian or normal probability distribution is the most commonly occurring distribution in Nature. The Central

Limit theorem states that the distribution of the sum of $n$ independent random variables, taken from any distribution with a finite variance, will eventually converge to the Gaussian distribution for large $n$. More formally, the theorem states that, given the sum of a sequence of i.i.d random variables, $S_n$, taken from any distribution $P(X)$, such that

$$S_n = X_1 + X_2 + \ldots + X_n \qquad X_i \sim \text{ any } P(X) \tag{2.24}$$

with a mean, $\mu = E(S_n)$, and variance $\sigma^2 = E((S_n - \mu)^2) < \infty$, then, as $n \to \infty$, the form of the original probability distribution, $P(X)$, does not matter and the distribution of $S_n$ will always converge to a Gaussian distribution with a mean of $n\mu$ and a variance of $n\sigma^2$. Consequently, if the quantity of interest has an unknown probability distribution, but it occurs very frequently, then it is valid to approximate its distribution as a Gaussian one with an empirically measured mean and variance.

A random number generated according to the Gaussian distribution has continuous values and is denoted by $x \sim \text{N}(\mu, \sigma^2)$. Shown in Figure 2.5, the probability and cumulative distribution functions of a Gaussian random variable with a mean $\mu$ and variance $\sigma^2$ are

$$P(x;\mu,\sigma) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) \tag{2.25}$$

and

$$F(x;\mu,\sigma) = \frac{1}{2}\left(1 + \text{erf}\left(\frac{x-\mu}{\sigma\sqrt{2}}\right)\right) \tag{2.26}$$

where erf(x) is the error function,

$$\text{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x \exp(-y^2)dy \tag{2.27}$$

In stochastic chemical kinetics, the occurrences of many chemical reactions in a large enough time increment can be validly approximated as a Gaussian random number.

### 2.2.4 A Brief Overview of Stochastic Processes

Consider a sequence of random variables, $X_n, X_{n-1}, X_{n-2}, \ldots, X_1$, that describes a quantity of something observed at times $t_n > t_{n-1} > t_{n-2} > \ldots > t_1$.

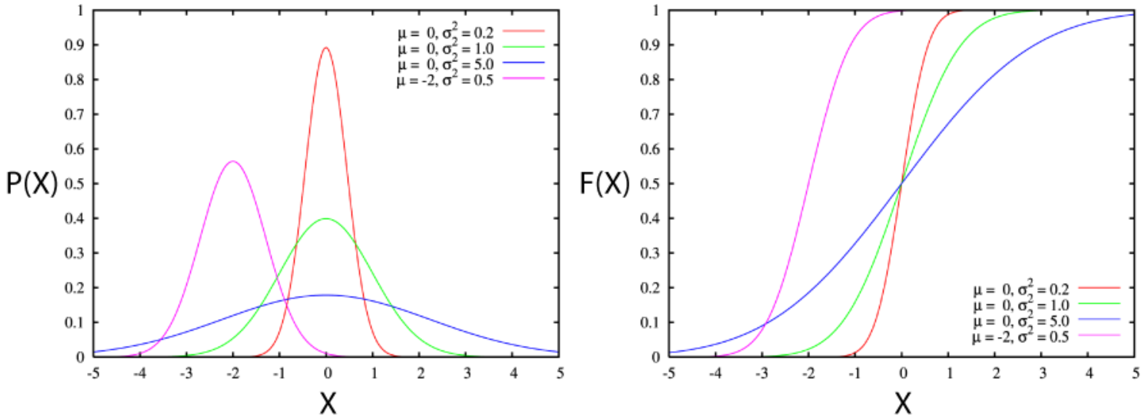

Figure 2.5: The probability distribution function, $P(X)$, and cumulative distribution function, $F(X)$, of a Gaussian distributed random variable, $X \sim \mathrm{N}(\mu, \sigma^2)$

$X_n$ is called a *stochastic process* indexed by $t$. The index $t$ can be either discrete or continuous valued, creating either a discrete-time or continuous-time stochastic process, respectively. The random variables $X_i$ may also be either discrete or continuous valued, generating a stochastic process with either discrete or continuous states. The *joint probability distribution* of a stochastic process fully describes all possible outcomes of its sequence of ordered pairs $(X_i,\ t_i)$. By plotting $X_i$ vs. $t_i$ for $i = 1 \ldots n$, one may visualize a *trajectory* of the stochastic process, which is the history of one possible set of outcomes. Besides its joint probability distribution, a stochastic process has other quantities that characterize its behavior, including its mean, variance, covariance, and moment generating function. Assuming they are well-defined, the mean and variance at any instant of time $t$ are

$$\mu(t) = E(\ X(t)\ ) \quad \sigma^2(t) = E(\ (X(t) - \mu(t))^2\ ) \tag{2.28}$$

and the covariances are

$$C(s,t) = E(\ (X(s) - \mu(s))\ (X(t) - \mu(t))\ ) \tag{2.29}$$

for any $s,t\ \in [0,T]$. Notice that the covariance simplifies to the variance when $s\ =\ t$.

A stochastic process $(X_i,\ t_i)$ can be classified according to the probability distribution of its *random increments*, $P(X_{i+1} - X_i;\ t_{i+1} - t_i)$. The distribution may be conditioned on all past and future values of the stochastic process, it may completely independent of any past or future values values, or it be conditioned on only a small subset of the past or future. The distribution may also be a function of $t$ or $X$. Depending on the conditioning and functional dependence of the distribution of the random increments, different stochastic dynamical behaviors will arise. Below, four examples of different stochastic processes are briefly reviewed, including random walks, Markov processes, Lévy processes, and stochastic differential equations.

### Random Walks

An unbiased *random walk* is a stochastic process whose random increments are independent and identically distributed according to a constant probability distribution with zero mean, or i.i.d $X_{i+1} - X_i \sim P(X)$ with $E(X) = 0$. The time step is discrete so that $t_{i-1} - t_i = 1$. The joint probability distribution of the sequence $(X_i,\, t_i)$ then greatly simplifies to

$$\begin{aligned} &P(X_n = x_n,\, X_{n-1} = x_{n-1}, \ldots,\, X_1 = x_1; t_n = \tau_n,\, t_{n-1} = \tau_{n-1}, \ldots,\, t_1 = \tau_1) \\ &\qquad = \; P(X_n = x_n; t_n = n)\; P(X_{n-1} = x_{n-1}; t_{n-1} = n-1) \;\cdots\; P(X_1 = x_1; t_1 = 1) \end{aligned} \tag{2.30}$$

For example, if $P(X_{i+1} - X_i \;=\; +1) = 1/2$ and $P(X_{i+1} - X_i \;=\; -1) = 1/2$, then a random walk starting at $X_o$ has mean $\mu = 0$ and variance $\sigma^2 = 1$. In Figure 2.6, three trajectories of the same random walk are shown. Importantly, an unbiased random walk in one or two dimensions is *recurrent*; from a starting point, the walk will eventually visit any other state in the domain within a finite (but possibly very large) time. Gambling games are often modeled using random walks, where a gambler wins or loses an amount of money with some probability. Because the random walk is recurrent, the gambler's amount of money will eventually visit zero and he will be forced to stop playing. Consequently, the unbiased random walk is often (and aptly) called the *gambler's ruin*. To compare, a stochastic process that leaves and never revisits a state again is called *transient*.

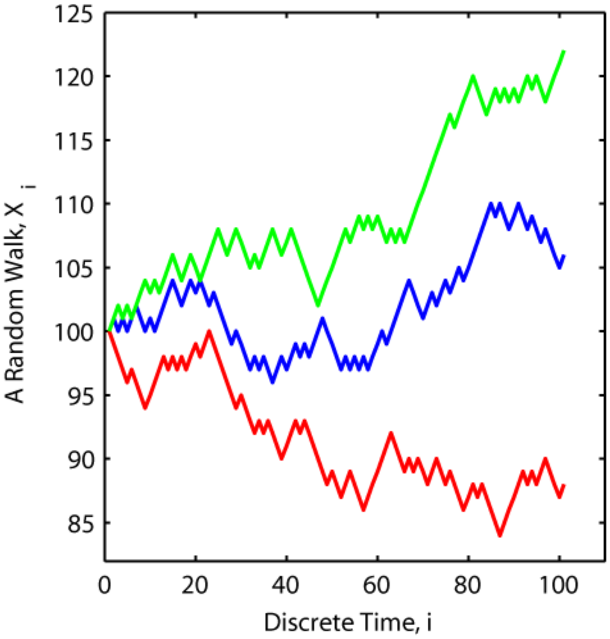

Figure 2.6: Three trajectories of the random walk called Gambler's Ruin, starting at $X_o = 100$, are shown. The red gambler is not doing so well.

### Jump Markov Processes

A stochastic process is called a memoryless or *Markov process* if the probability distribution of the increments, $P(X_{i+1} - X_i)$, has "forgotten" the past and depends only on the present value, $(X_i,\ t_i)$, such that the conditioning of the distribution simplifies to

$$\begin{aligned} P(X_{i+1} - X_i;\ t_{i+1} \mid X_i = x_i,\ X_{i-1} = x_{i-1}, \ldots,\ X_1 = x_1; t_i = \tau_i,\ t_{i-1} = \tau_{i-1}, \ldots,\ t_1 = \tau_1) \\ = P(X_{i+1} - X_i;\ t_{i+1} \mid X_i = x_i; t_i = \tau_i) \\ = P(X_{i+1}; t_{i+1} \mid X_i = x_i; t_i = \tau_i) \end{aligned} \tag{2.31}$$

The joint probability distribution of the sequence $(X_i,\ t_i)$ for $i = 0, \ldots, n$ then simplifies to

$$\begin{aligned} & P(X_n = x_n,\ X_{n-1} = x_{n-1}, \ldots,\ X_1 = x_1; t_n = \tau_n,\ t_{n-1} = \tau_{n-1}, \ldots,\ t_1 = \tau_1) \\ & \quad = P(X_n = x_n; t_n \mid X_{n-1}; t_{n-1} = \tau_{n-1})\, P(X_{n-1} = x_{n-1}; t_{n-1} \mid X_{n-2}; t_{n-2} = \tau_{n-2}) \cdots \\ & \qquad P(X_{n-2} = x_{n-2}; t_{n-2} \mid X_{n-3}; t_{n-3} = \tau_{n-3}) \cdots P(X_1 = x_1; t_1 = \tau_1 \mid X_o; t_o) \end{aligned} \tag{2.32}$$

with an initial condition $(X_o, t_o)$. A Markov process has discrete time if the time increments are restricted to integers and continuous time if they are not. We will restrict our focus to continuous time Markov processes to better understand physical systems.

If the *states* of a continuous time Markov process are discrete valued, then the process is called a *jump Markov process*. A jump Markov process will spontaneously transition from state $X_j$ to an adjacent state $X_k$ at a *jump time* $t_{i+1}$. The Markov process does not otherwise change its state in between jump times, causing the trajectory to have a discontinuous path. The derivative of the random increments, $\partial(X_{i+1} - X_i)/\partial\, t$, with respect to time is also undefined. By enumerating all possible states of the system (e.g. $\mathcal{X} = \{x_1, x_2, \ldots, x_N\}$ in one dimension), one can construct a *Markov chain* $X(t) \in \mathcal{X}$. The probability of finding the system at any state in the Markov chain at time $t$ is summarized with a *probability vector*, $P(X(t) = x_i)$. Because of the Markovian property, the probability of transitioning from one state to another adjacent state can be rewritten as a *transition matrix*. The transition matrix $\mathbb{M}(t_{i+1}; t_i)$ has elements $p^{\,k,\,j}(t_{i+1};\, t_i)$ so that

$$p^{\,k,\,j}(t_{i+1};\, t_i) = P(X(t_{i+1}) = x_k \mid X(t_i) = x_j) \tag{2.33}$$

which are the probability distributions of the frequency of transitions from state $X_j$ at time $t_i$ to $X_k$ at time $t_{i+1}$. These distributions may be any continuous valued probability distribution. However, if the transition distributions depend only on the length of the jump time $\Delta\, t = t_{i+1} - t_i$, then the Markov chain is called *homogeneous* and the transition probabilities may be rewritten as $p^{\,k,\,j}(\Delta\, t)$. Homogeneous jump Markov processes are the most common Markovian processes when modeling physical systems with discrete states, such as chemically reacting systems. Therefore, it is important to understand how these processes evolves over time.

By the first Law of Probability, a transition to another adjacent state must always occur and so the elements of the transition matrix satisfy

$$\sum_{k=1}^{n} p^{\,k,\,j}(\Delta\, t) = 1 \tag{2.34}$$

and the probability of finding the system at $X(t+\Delta\, t)$ is the matrix algebra

$$P(X(t+\Delta\, t)) = P(t)\mathbb{M}(\Delta\, t) \tag{2.35}$$

If we take the derivative of the transition probabilities with respect to time, in the limit, we obtain the instantaneous rates or *propensities* of each transition, $a^{\,k,\,j}$, which are

$$a^{\,k,\,j} = \begin{cases} \lim_{\Delta\, t \to 0} \frac{p^{\,k,\,j}(\Delta\, t)}{\Delta\, t} & j \neq k \\ \\ \lim_{\Delta\, t \to 0} \frac{p^{\,k,\,j}(\Delta\, t)}{\Delta\, t} - 1 & j = k \end{cases} \tag{2.36}$$

The transition probabilities then satisfy the following *Kolmogorov forward equation*

$$\frac{dp^{\,k,\,j}(t)}{dt} = \sum_{m=1}^{N} p^{\,m,\,j}(t)\; a^{\,k,\,m} \tag{2.37}$$

and also the *Kolmogorov backward equation*

$$\frac{dp^{\,k,\,j}(t)}{dt} = \sum_{m=1}^{N} a^{\,k,\,m}\; p^{\,k,\,m}(t) \tag{2.38}$$

for $k,\ j = 1 \ldots N$.

The *stationary solution* of a jump Markov process is a probability vector, $P^{SS}(X)$, and time-invariant transition probabilities, $\underline{p}^{\,k,\,j}$, that satisifies both

$$\frac{dp^{\,k,\,j}(t)}{dt} = \sum_{m=1}^{N} \underline{p}^{\,m,\,j}\; a^{\,k,\,m} = 0 \tag{2.39}$$

and

$$P^{SS}(X) = P^{SS}(X)\; \underline{\mathbb{M}} \tag{2.40}$$

where the transition matrix $\underline{\mathbb{M}}$ has elements $\underline{p}^{\,k,\,j}$. If the stationary solution of a jump Markov process is recurrent, then it is also *ergodic*. An ergodic stationary solution will continue to visit all of its states with a frequency that is proportional to their probability. Consequently, a jump Markov process, $X(t) \in \mathcal{X}$ with an ergodic stationary solution, $P^{SS}(X)$, will satisfy

$$\lim_{T \to \infty} \frac{1}{T} \sum_{t=1}^{T} f(X(t)) = \sum_{i=1}^{N} f(x_i) P^{SS}(x_i) \tag{2.41}$$

for any function $f(x)$. The ergodic condition will be used in the formulation of our equation-free probabilistic steady-state approximation in section **??**. If the Markov chain is both recurrent and reversible, the probabilities and transition rates satisfy a condition called *detailed balance*, which is

$$p^{\,k,\,j} P^{SS}(x_k) = p^{\,j,\,k} P^{SS}(x_j) \tag{2.42}$$

A reversible Markov chain is one in which, for each transition, there exists another transition in the opposite direction.

The transition rates of a homogenous jump Markov process often depend on the state of the system in some complicated way. In a multidimensional system with the state vector $\mathcal{X} = \{X_1, X_2, \ldots, X_N\}$, explicitly writing down the transition rates becomes unwieldly. Instead, one may collapse the matrix of transition rates, $a^{\,k,\,j}(\mathcal{X})$, into a system of $j = 1, \ldots, M$ types of transitions which alter the state of the system by a constant value, $\Delta\,\mathcal{X}_j$, with jump rates, $a_j(\mathcal{X})$. In addition, it is often more appropriate to describe the time evolution of the probability vector, $P(\mathcal{X};t)$, rather than the transition probabilities. The Kolgomorov forward equation may then be rewritten in terms of the $j$ transitions to create the *forward Master equation*, which is

$$\frac{dP(\mathcal{X};t)}{dt} = \sum_j \left[a_j(\mathcal{X}(t) - \Delta\,\mathcal{X}(t))\; P(\mathcal{X}(t) - \Delta\,\mathcal{X}(t)) - a_j(\mathcal{X}(t))\; P(\mathcal{X}(t))\right] \tag{2.43}$$

The Master equation describes the time evolution of the probability vector in terms of the transition rates of the jump Markov process. There is also a *backward Master equation*, which is

$$\frac{dP(\mathcal{X};t)}{dt} = \sum_j \left[a_j(\mathcal{X}(t) + \Delta\,\mathcal{X}(t))\; P(\mathcal{X}(t) + \Delta\,\mathcal{X}(t)) - a_j(\mathcal{X}(t))\; P(\mathcal{X}(t))\right] \tag{2.44}$$

Consider that this stochastic process is Markovian (ie. memoryless) and that the only memoryless probability distribution is the exponential distribution. Consequently, the jump times of the $k \rightarrow j$ transitions with $k \neq j$ are exponentially distributed with rate $a^{\,k,\,j}$ or $\Delta\ t \sim$ Exp$(a^{\,k,\,j})$. Due to Eq. (2.34), the jump time to any $k \neq j$ transition is also exponentially distributed, but with a rate $\sum_k a^{\,k,\,j}$. These facts are extremely important in simulating the dynamics of a homogenous jump Markov process whose transition matrix is too unwieldly to manipulate.

### Continuous Markov Processes

If the states of a continuous time Markov process are continuous valued, then the process is called a *continuous Markov process*. A continuous Markov process continuously jumps from one state to another so that its trajectory is a continuous path. The derivative of its random increments, $\partial(X_{i+1} - X_i)/\partial\, t$, with respect to time is now defined, but in a special way. A subset of continuous Markov processes are homogenous ones whose random increments are sampled from a continuous-valued Markovian probability distribution that depends only on $X_i$ and the time between observations $\Delta\, t = t_{i+1} - t_i$. These times are not jump times. One may repeatedly "zoom in" on the dynamics of a continuous Markov process, decreasing the length of time between observations, and viewing a rescaled version of the same stochastic dynamical behavior. Consequently, the dynamics of a homogenous continuous Markov process are *fractals*.

A homogenous continuous Markov process $X(t) \in \mathcal{X}$ has a conditional transition probability

$$\begin{aligned} P(s,\ x; t,\ \mathcal{X}) &= P(X(t) = \mathcal{X} \mid X(s) = x) \\ &= \int_{x \in \mathcal{X}} p(x; \Delta\, t,\ y) dy \end{aligned} \tag{2.45}$$

where $p(x; \Delta\, t,\ y)$ is called the *transition density* of the process, equivalent to the transition probabilities of a jump Markov process. A homogenous continuous Markov process is called a *diffusion* process if well-defined drift $a(x,t)$ and diffusion $D(x,t)$ coefficients exist such that

$$\begin{aligned} a(x,t) &= \lim_{|t-s| \to 0} \tfrac{1}{t-s}\, E(X(t) - X(s) \mid X(s) = x) \\ D(x,t)^2 &= \lim_{|t-s| \to 0} \tfrac{1}{t-s}\, E((X(t) - X(s))^2 \mid X(s) = x) \end{aligned} \tag{2.46}$$

Similar to the Kolmogorov forward and backward equations in Eqs. (2.37) and (2.38), the transition density $p(x; \Delta\, t,\ y)$ of a diffusion process satisfies

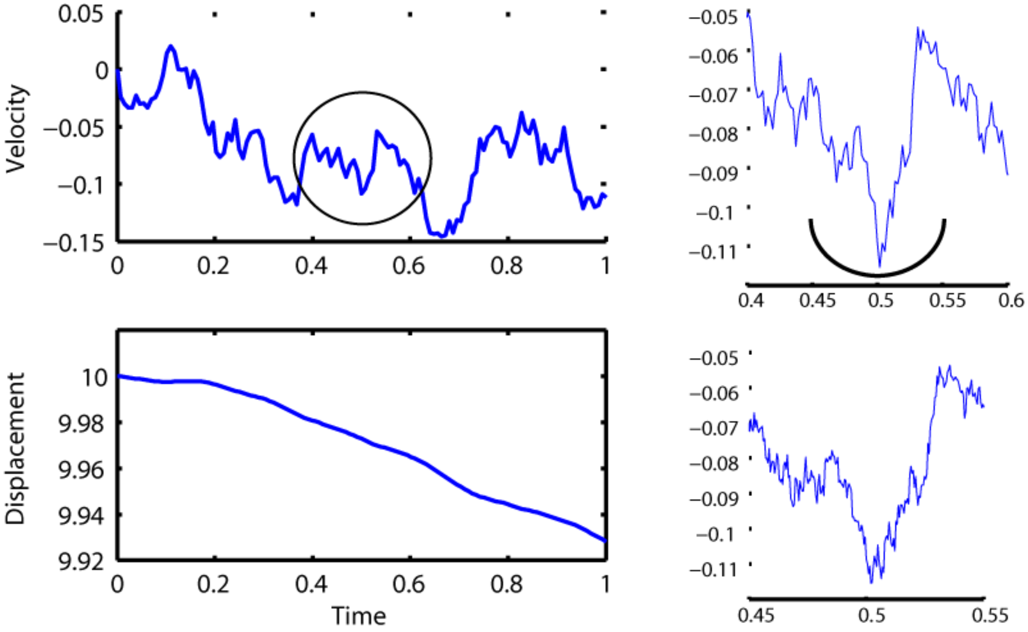

Figure 2.7: (Left) A single trajectory of the displacement and velocity of a Brownian particle is shown with a diffusion coefficient, $D = 1$, on the interval $T = [0, 1]$. (Right) The same velocity trajectory is shown on $T = [0.4, 0.6]$ and $T = [0.45, 0.55]$, showing the fractal behavior of the Wiener process.

a partial differential equation called the *Fokker-Planck equation*,

$$\frac{\partial p(x,t)}{\partial t} + \frac{\partial}{\partial x}\{a(x,t)\ p(x,t)\} - \frac{1}{2}\frac{\partial^2}{\partial x^2}\left\{D(x,t)^2\ p(x,t)\right\} = 0 \qquad (2.47)$$

A diffusion process may have a stationary solution, $p(x,t) = \underline{p}(x)$, that satisfies Eq. (2.47) when $\frac{\partial p}{\partial t} = 0$.

One of the most well-known examples of a homogenous continuous Markov process is the Brownian motion of a large particle surrounded by water molecules [120]. In a planar system, it is a four-dimensional continuous Markov process $\left(\{d_i^x, d_i^y, v_i^x, v_i^y\};\ t_i\right)$ where $d_i^x$ and $d_i^y$ are the displacement of the particle and $v_i^x$ and $v_i^y$ are the particle's instantaneous velocity in the $x$ and $y$ cartesian directions. The water molecules in the system continuously collide with the particle and exert instantaneously applied forces on it. The motion of the particle and all of the water molecules may be described by Newton's law. However, because there are so many water particles, one may approximate their applied force on the particle as a Gaussian distributed random variable. Consequently, Newton's law for only the particle's motion becomes

$$\begin{array}{ll} \frac{d\ d^x(t)}{dt} = v^x(t) & \frac{d\ d^y(t)}{dt} = v^y(t) \\ \frac{d\ v^x(t)}{dt} = \mathsf{N}(0, 4D^2\ dt) & \frac{d\ v^y(t)}{dt} = \mathsf{N}(0, 4D^2 dt) \end{array} \tag{2.48}$$

where any friction on the particle has been neglected, the average positions of the water molecules are assumed to be unchanging, and $D$ is the diffusion coefficient of the particle. The velocities of the particle in the $x$ and $y$ cartesian directions are a Gaussian distributed random variable with zero mean and $4D^2dt$ variance. Equation (2.48) is called a *Langevin equation* with additive noise. The integration of the Langevin equation describes a single trajectory of a Brownian particle in time.

The probability distribution of all possible trajectories of the Brownian particle satisfies the Fokker-Planck equation

$$\frac{\partial p(x,t)}{\partial t} - D\frac{\partial^2\ p(x,t)}{\partial x^2} = 0 \tag{2.49}$$

The solution to Eq. (2.49) is simply the Gaussian distribution with zero mean and $4D\,t$ variance and may be checked by the reader. Interestingly, a Brownian particle has no stationary solution. As time goes to infinity, the variance also goes to infinity, indicating that the Markov process is not ergodic. In Figure 2.7, the Brownian motion of a particle with $D = 1$ is shown. The fractal behavior of a continuous Markov process is illustrated by zooming in to smaller and smaller time intervals.

The reader may notice that the Langevin equation in Eq. (2.48) implies that a Gaussian random number, $\mathsf{N}(0, dt)$, is a well-defined quantity; however, when the Langevin equation was first proposed, this was not true. Since then, mathematicians have studied the derivatives and integrals of stochastic processes, including a specially defined one called a Lévy process. The integral of a Gaussian random number with a zero mean and a variance of $dt$ is now called the Wiener process, which is a type of Lévy process. The Langevin equation itself has been generalized to describe the time evolution of many different continuous valued continuous-time stochastic processes. These differential equations are called *stochastic differential equations*. These topics will be briefly discussed below.

### Lévy Processes

Consider a stochastic process, $X(t)$, with random increments $X(t_{i+1}) - X(t_i)$. If the increments are independent and their distribution depends only on the time in between the increments, $t_{i+1} - t_i$, then the stochastic process is called a *Lévy process*. The increments of a Lévy process are random variables that are decomposable into three components: a drift motion, a diffusion motion, and a jump motion. The three motions are respectively caused by a deterministic process, the *Wiener process*, and the *Poisson process*, each rescaled with parameters $\alpha$, $\sigma$, and $\lambda$. The Lévy process, $X(t)$, changes over time from $t_i$ to $t_{i+1}$ according to

$$X(t_{i+1}) = X(t_i) + \alpha(t_{i+1} - t_i) + \sigma\big(W(t_{i+1}) - W(t_i)\big) + \big(P_\lambda(t_{i+1}) - P_\lambda(t_i)\big), \tag{2.50}$$

where $\alpha$, $\sigma$, and $\lambda$ are time-independent constants, $W(t)$ is the Wiener process evaluated at $t$, and $P_\lambda(t)$ is a Poisson process with rate $\lambda$ evaluated at $t$. The Wiener and Poisson processes are both, by themselves, Lévy processes. Their increments are respectively distributed according to the Gaussian and Poisson distributions, so that

$$\begin{aligned} W(t_{i+1}) - W(t_i) &\sim \mathsf{N}(0, t_{i+1} - t_i) \\ P_\lambda(t_{i+1}) - P_\lambda(t_i) &\sim \mathsf{Poisson}\big(\lambda(t_{i+1} - t_i)\big) \end{aligned} \tag{2.51}$$

The Wiener process has mean $\mu(t) = 0$, variance $\sigma^2(t-s) = t - s$, and covariances $C_{s\,t} = \min(s,t)$. The Poisson process has mean $\mu(t) = \lambda t$, variance $\sigma^2 = \lambda t$, and covariances $C_{s\,t} = \lambda \min(s,\ t)$.

Consider what happens to the Wiener and Poisson processes when the time increment $t_{i+1} - t_i$ shrinks to zero as in

$$\begin{aligned} dW_t &= \lim_{t_{i+1}-t_i \to 0} \ \tfrac{W(t_{i+1}) - W(t_i)}{t_{i+1} - t_i} \\ dP_{\lambda\,t} &= \lim_{t_{i+1}-t_i \to 0} \ \tfrac{P_\lambda(t_{i+1}) - P_\lambda(t_i)}{t_{i+1} - t_i} \end{aligned} \tag{2.52}$$

The limits in Eq. (2.52) are the definitions of the derivatives of the Wiener and Poisson processes with respect to time and are denoted by $dW_t$ and $dP_{\lambda t}$.

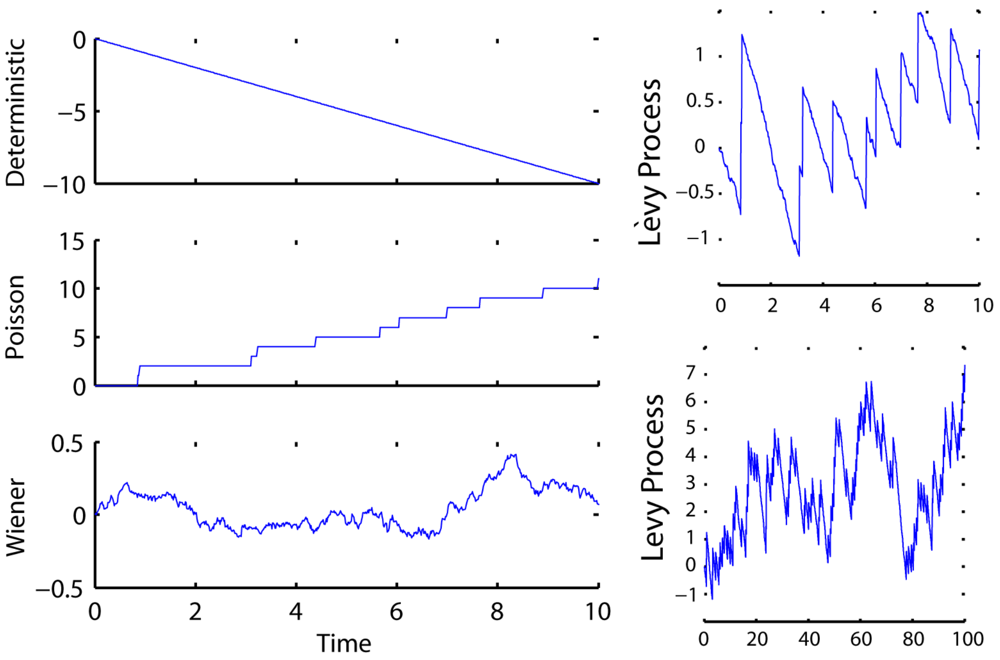

Figure 2.8: A Lévy process is the sum of a deterministic, Poisson, and Wiener process. (Left) Single trajectories of the deterministic, Poisson, and Wiener processes are shown with $\alpha = -1$ and $\sigma = \lambda = 1$ on the interval $T = [0,\, 10]$. (Right) The resulting Lévy process is shown on the interval $T = [0,\, 10]$ and $T = [0,\, 100]$.

However, the limits do not exist in the normal sense. Both the Wiener and Poisson processes are nowhere differentiable using the common definition. Instead, they are the abstract represention of a Wiener and Poisson process integrated over a $dt$ of time. The mean and variance of $dW_t$ and $dP_{\lambda t}$ are, respectively, $0$ and $1$, and $\lambda$ and $\lambda$, but they must always be integrated along some time interval to obtain a real number.

In Figure 2.8, we show how the deterministic process, the Wiener process, and the Poisson process evolve over time with $\alpha = -1$ and $\sigma = \lambda = 1$ and how they combine to form a Lévy process. A more generalized Lévy process may also be created by allowing $\alpha$, $\sigma$, and $\lambda$ to depend on $t$. An even greater generalization of these scaling functions leads us to a definition of a stochastic differential equation.

### Stochastic Differential Equations

If we consider a Lévy process, $X(t)$, with constants $\alpha$, $\sigma$, or $\lambda$ that are now functions of both $X$ and $t$, then the evolution of the stochastic process becomes

$$X(t_{i+1}) = X(t_i) + \int_{t_i}^{t_{i+1}} \alpha(X,t')\,dt' + \int_{t_i}^{t_{i+1}} \sigma(X,t')\,dW_{t'} + \int_{t_i}^{t_{i+1}} dP_{\lambda(X,t')} \tag{2.53}$$

where the integrals of $dW_t$ and $dP_{\lambda t}$ are *Lebesgue-Stieltjes* integrals that may integrate discontinuous functions and $dW_t$ and $dP_{\lambda t}$ are the derivatives of the Wiener and Poisson processes, respectively. Because the parameters depend only on the current state and time, the stochastic process is Markovian. However, the increments may no longer be independent, so it is not necessarily a Lévy process. By taking the derivative of Eq. (2.53) with respect to time, we obtain a Markovian *stochastic differential equation* or SDE

$$dX(t) = \alpha(X(t),t)\,dt + \sigma(X(t),t)\,dW_t + dP_{\lambda(X,t)} \tag{2.54}$$

that is driven by a deterministic process, a Wiener process, and a Poisson process.

A stochastic differential equation describes how some quantity, $X(t)$, changes over time while being affected by a stochastic process. The quantity $X(t)$ becomes a stochastic process and its rate of change can be decomposed into its component drift, diffusion, and jump terms, represented by the deterministic, Wiener, and Poisson processes. By setting $\sigma = \lambda = 0$, Eq. (2.54) simplifies to an ordinary differential equation, which is the familiar deterministic (non-random) process. By setting $\lambda = 0$, Eq. (2.54) simplifies to either an *Itô* or *Stratonovich* stochastic differential equation, depending on our usage of either the Itô or Stratonovich interpretation of a *stochastic integral*, such as $\int f(X(t))\,dW_t$. More generally, if $\alpha$, $\sigma$, or $\lambda$ depend on either the past or future values of $X(t)$ then the stochastic differential equation becomes Non-Markovian, such as a *stochastic delayed differential equation*. In stochastic chemical kinetics, we often use a system of Markovian Itô stochastic differential equations to describe the dynamics of a chemically reacting system. In section 2.4, the numerical solution of these stochastic differential equations will be discussed.

## 2.3 The Numerical Simulation of Jump Markov and Poisson Processes

Jump Markov and Poisson processes are often used to model dynamic systems with discrete states, such as the movement of inventory in a supply chain, the assembly of parts in a manufacturing line, or the servicing of customers. In our case, we are most interested in describing the stochastic dynamics of a chemically reacting system by modeling the occurrences of reactions as either a jump Markov or Poisson process. The discrete states of the system are the number of molecules of each chemical species and the transitions between adjacent states are the occurrences of the chemical reactions. The jump Markov process description is more exact; it treats each reaction occurrence as an exponentially distributed event with a *time independent rate*. The Poisson process description is an approximation; it bundles together multiple reaction occurrences into a single jump. The number of reaction occurrences that occur in a single jump is a Poisson distributed event with a *time-dependent rate*.

Most chemical reaction networks have non-linear reaction kinetics and a very large or possibly infinite number of discrete states. Accordingly, the equations governing the time-evolution of their probability distributions, such as the Kolgomorov and Master equations, are often intractable to solve. Instead, by generating possible trajectories of each process, we can sufficiently sample the probability distribution of the system and how it changes over time. These techniques are generally classified as kinetic Monte Carlo methods as they historically originate from the Monte Carlo sampling of equilibrium systems. Consequently, in this section, we focus on different kinetic Monte Carlo techniques for simulating jump Markov and Poisson processes.

### 2.3.1 Definitions

Consider a well-stirred, isothermal chemical reaction network with $j = 1, \ldots, M$ reactions and $i = 1, \ldots, N$ species in a volume, $V$. The state vector $X_i(t)$ is the number of molecules of the $i^{th}$ chemical species at a time $t$. The rate of the $j^{th}$ reaction is $a_j(\underline{\mathsf{X}}(t))$ and also depends on the state vector. The effect of each reaction on the state vector is determined by the stoichiometric matrix, $\nu$. The vector $\nu_j$ describes the effect of the $j^{th}$ reaction on the number of molecules of each chemical species.

The states of the system can be described as an $N$-dimensional lattice that spans all non-negative integers. At *each* state, there are $M$ transitions corresponding to the chemical reactions that may occur in the system. If the system is open (ie. if there is mass transfer across the boundaries of the system), then the total number of possible states is infinite, but many of those states will be rarely visited. For the remainder, we will assume that the system may be open. Accordingly, the detailed balance condition can not be assumed.

The transition rates of each reaction are described by a rate law, so that $a_j(\underline{\mathsf{X}}(t)) = f(\underline{\mathsf{X}}(t))$. These transition rates are also called propensities in the literature. Here, we will restrict ourselves to describing only the mass action rate laws with zero, one, two, or three reactant molecules. These rate laws are called the $0^{th}$, $1^{st}$, $2^{nd}$, and $3^{rd}$ order mass action rate laws. The rate laws may be written down in terms of the number of molecules of the reacting chemical species and with a modification to the kinetic constants, but the rate laws themselves are the same. The probabilistic interpretation of a rate law is the number of different ways the reactant molecules may collide multiplied by a probability of a collision resulting in a reaction. Therefore, the rate law for a stochastic process may be separated into a combinatorial factor and a specific kinetic constant. When writing down the rate law in terms of the number of molecules in the system, the specific kinetic constant depends on the volume of the system and is described as the mesoscopic kinetic constant. When writing down the rate law in terms of concentrations, the specific kinetic constant is volume-independent and is described as the macroscopic kinetic constant. These differences reflect nothing more than two different ways of separating the rate law into a combinatorial factor and a specific kinetic constant. The simplest way to see the connection is to write down the units for the mesoscopic and macroscopic kinetic constants for each rate law. For completeness, we list the rate laws in terms of the number of molecules in the system and four hypothetical chemical species, $A$, $B$, $C$, and $D$ and describe how the mesoscopic kinetic constant, denoted

Table 2.2: A list of the mass action rate laws for stochastic chemical kinetics. $N_A$: Avagadro's number. $V$: Volume. Molec: Molecules

| Order | Reaction | Rate Law [Molec/sec] | Units of c | Units of k | Conversion |
|---|---|---|---|---|---|
| $0^{th}$ | $\rightarrow A$ | $a = c$ | [Molec/sec] | [Molar/sec] | $c = kVN_A$ |
| $1^{st}$ | $A \rightarrow B$ | $a = c\#A$ | [1/sec] | [1/sec] | $c = k$ |
| $2^{nd}$ Bi- | $A + B \rightarrow C$ | $a = c\#A\#B$ | [Molec sec]$^{-1}$ | [Molar sec]$^{-1}$ | $c = \frac{k}{VN_A}$ |
| $2^{nd}$ Mono- | $2A \rightarrow B$ | $a = c\#A\,(\#A-1)/2$ | [Molec sec]$^{-1}$ | [Molar sec]$^{-1}$ | $c = \frac{2k}{VN_A}$ |
| $3^{rd}$ Tri- | $A + B + C \rightarrow D$ | $a = c\#A\#B\#C$ | [Molec sec]$^{-2}$ | [Molar sec]$^{-2}$ | $c = \frac{k}{(VN_A)^2}$ |
| $3^{rd}$ Bi- | $2A + B \rightarrow C$ | $a = c\#A\,(\#A-1)\,\#B/2$ | [Molec sec]$^{-2}$ | [Molar sec]$^{-2}$ | $c = \frac{2k}{(VN_A)^2}$ |
| $3^{rd}$ Mono- | $3A \rightarrow B$ | $a = c\#A\,(\#A-1)\,(\#A-2)/6$ | [Molec sec]$^{-2}$ | [Molar sec]$^{-2}$ | $c = \frac{6k}{(VN_A)^2}$ |

by $c$ is related to the macroscopic kinetic constant, denoted by $k$, with their corresponding units.

### 2.3.2 The Stochastic Simulation Algorithm

In the mid-1970s, two different kinetic Monte Carlo techniques were developed to simulate the dynamics of a jump Markov process. The first numerical scheme, the N-fold algorithm by Bortz, Kalos, and Lebowitz (BKL) [121], simulates a dynamic trajectory of an Ising model, which is a classical example of a jump Markov process describing physical systems, including simple liquids and ferromagnetism. In contrast to previous Monte Carlo algorithms that describe the equilibrium properties of a system, the N-fold algorithm introduces a method of calculating the jump times between transitions in a non-equilibrium system. The second numerical scheme is the stochastic simulation algorithm by Daniel Gillespie [122, 123]. The stochastic simulation algorithm (SSA) generates a trajectory of a chemical reaction network modeled as a non-equilibrium jump Markov process. Its recent popularity arises from the observation that gene expression in biological organisms can be accurately modeled as a jump Markov process. The SSA was first proposed in two equivalent variants, the Direct and First Reaction variants. Later, in 2000, the Next Reaction variant was created by optimizing the First Reaction variant for large reaction networks [106]. We now briefly review these three variants of the stochastic simulation algorithm.

All three variants of the stochastic simulation algorithm seek to answer these questions:

- What is the time of the next reaction occurrence?
- Which reaction occurs at that time?

These two questions describe the events in the system. The outcomes of the events are a jump time, $\tau$, and the identity of the next reaction to occur, which is $\mu$. The joint probability distribution $P(\tau, \mu)$ of these events is sampled from the distribution,

$$P(\tau, \mu)d\tau = a_\mu \exp\left(-\tau\sum_j a_j\right) \tag{2.55}$$

There are two different ways to sample $\tau$ and $\mu$ from this distribution, reflecting the differences between the Direct and First Reaction variants.

### Direct Variant

In the Direct variant, the time of the next reaction occurrence, $\tau$, is an exponentially distributed random number with rate $\sum_j a_j$. The $\tau$ is calculated according to

$$\tau = \frac{-\log(\mathsf{URN}_1)}{\sum_{j=1}^{M} a_j} + t \tag{2.56}$$

and the identity of the next reaction, $\mu$, is a uniformly distributed random number with probability density $a_\mu / \sum_j a_j$. The random variable $\mu$ is determined by finding the integer, $\mu$, that satisfies

$$\sum_{j=1}^{\mu-1} a_j < \mathsf{URN}_2 \sum_{j=1}^{M} a_j \leq \sum_{j=1}^{\mu} a_j \tag{2.57}$$

where both $\mathsf{URN}_1$ and $\mathsf{URN}_2$ are uniform random numbers on $(0, 1)$. Once the pair $(\mu, \tau)$ has been sampled, the state of the system at the next jump time is simply determined by

$$X = X + \nu_\mu \quad t = \tau \tag{2.58}$$

After the state of the system has been altered, the reaction propensities are recalculated according to their rate laws. By repeating this simple procedure, one can generate a history of the pairs $(\mu_k, \tau_k)$ for $k = 1, 2, \ldots, \mathcal{N}$ for

$\mathcal{N}$ reaction occurrences. The computational cost of one iteration of this procedure is (a) the calculation of the reaction propensities, (b) the generation of the two uniform random numbers, and (c) one division and a number of insignificant additions. Importantly, the computational cost of the Direct variant is directly proportional to the number of reaction occurrencs and directly proportional to the number of reactions in the system.

### First Reaction Variant

In the Next Reaction variant, the time of the each reaction occurrence, $\tau_j$, is an exponentially distributed random number with rate $a_j$. Now, there is a vector of reaction times, $\tau_j$ for $j = 1, \ldots, M$, which are calculated according to

$$\tau_j = \frac{-\log(\mathsf{URN}_j)}{a_j} + t \tag{2.59}$$

The identity of the next reaction occurrence, $\mu$, is the one that occurs first and may be determined by calculating the minimum value of the vector of reaction times, $\tau_j$, or

$$\tau_\mu = \min_\mu(\tau) \tag{2.60}$$

Again, after determining the pair $(\tau_\mu, \mu)$, the state and time of the system is updated according to

$$X = X + \nu_\mu \quad t = \tau_\mu \tag{2.61}$$

The reaction propensities, $a$, are recalculated according to their rate laws, and a trajectory of the state over time is generated by repeating the above procedure. The probability distribution of selecting $\tau_\mu$ in the First Reaction variant is exactly equivalent to the distribution of $\tau$ in the Direct variant. However, the computational cost of the First Reaction variant is different. Per iteration, the computational cost of the First Reaction variant consists of (a) the calculation of the reaction propensities, (b) the generation of $M$ uniform random numbers, (c) $M$ divisions and many other insignificant additions, and (d) finding the minimum value of an array of floating point numbers. Therefore, the computational cost is greater than the Direct variant. Its description illustrates how one may sample the same joint probability distribution in two different ways. By using optimizing data structures and reusing random numbers in an appropriate fashion, however, it is possible create another variant of the stochastic simulation algorithm that is most efficient for large, realistic chemical reaction networks. This variant is called the Next Reaction method.

### Next Reaction Variant

The Next Reaction variant of the stochastic simulation algorithm [106] is an optimized version of the First Reaction variant that greatly reduces its computational cost, especially when simulating the stochastic dynamics of chemical reaction networks with many reactions and species. It uses two different optimized data structures to minimize the computational cost associated with recalculating the reaction propensities and finding the minimum reaction time. The method also minimizes the generation of uniform random numbers by reusing previously generated ones whenever appropriate and valid. We briefly review these improvements.

The first optimizing data structure is a dependency graph. It minimizes the computational cost of recalculating the reaction propensities by identifying the reaction propensities which have changed since the previous iteration. By only updating the reaction propensities which have changed, one may significantly decrease the total computational time. The dependency graph can be represented as a two-dimensional list, $\mathcal{D}_{ji}$, where the $j^{th}$ row of $\mathcal{D}$ holds the list of reaction propensities which must be updated when the $j^{th}$ reaction occurs. The list of reaction propensities consists of all reactions whose rate laws depend on a reactant or product of the $j^{th}$ reaction. The dependency graph is created in the beginning of the simulation and then stored for all remaining iterations of the algorithm. Consequently, after each execution of a reaction, the list of reaction propensities that are updated is $\mathcal{D}_j$ and the length of that list is often much shorter than the total number of reactions in the system. Consequently, the computational cost of recalculating the reaction propensities is significantly reduced.

The second optimizing data structure, an indexed priority queue, is responsible for finding the minimum value of a list of numbers, performed in Eq. (2.60). An indexed priority queue is a minimum-top binary tree whose parent and children elements are each swapped until all parent elements store a lesser value than the value of their children elements. It is similar to a heapsort, but its values and dependencies are persistent and regularly updated. The advantage of using an indexed priority queue is that finding a minimum of a list of numbers takes $O(M\log M)$ swaps of elements, where $M$ is the number of reactions in the system. Other methods of finding a minimum value, such as a quicksort, asymptotically take a greater number of swaps. For our purposes, the indexed priority queue stores the reaction times for each reaction, $\tau_j$. After all of the reaction times are updated according to Eq. (2.59), the indexed priority queue is also updated. Because only a small number, $r \ll M$, reaction times require updating per iteration, the indexed priority queue requires only $O(r\log r)$ element swaps, which is significantly reduced.

Finally, whenever a reaction has not occurred, its uniform random number is reused. If the reaction propensity has not changed, the uniform random number is reused and the reaction time remains the same. If the reaction propensity has changed, the uniform random number may still be reused, but the corresponding reaction time of the $(k+1)^{th}$ iterate of the algorithm must be rescaled according to

$$\tau_j^{k+1} = \frac{a_j^k}{a_j^{k+1}}(\tau_j^k - t) + t \tag{2.62}$$

The reaction time remains distributed according to an exponential distribution, but its rate is now rescaled according to the new reaction propensity with an appropriate weight on the time that has already occurred.

The main advantages of the Next Reaction variant of the stochastic simulation algorithm are that (a) it recalculates the reaction propensities only when they have changed, (b) it minimizes the computational time required to find the minimum reaction time, and (c) asymptotically, it only requires 1 uniform random number per iteration. For systems with many loosely coupled reactions, the Next Reaction variant is the most efficient variant of the stochastic simulation algorithm. Ultimately, however, the computational cost of any variant of the stochastic simulation algorithm is directly proportional to the number of reaction occurrences. It is this drawback that has motivated the development of approximate or hybrid stochastic numerical methods.

### 2.3.3 Poisson and Binomial Leaping

Because the stochastic simulation algorithm records the occurrences of each reaction event in the history of a trajectory, its computational cost is directly proportional to the number of reaction occurrences in the system. Consequently, if there is a frequently occurring reaction, then the computational cost of generating a trajectory will significantly increase. Because of this disadvantage, the creator of the original SSA has since developed numerous approximate stochastic numerical methods with varying success. Other people have improved on these new methods or created entirely new ones, including this author. One of the first approximate stochastic methods was Gillespie's $\tau$-leaping method [119]. We briefly review it here.

The key idea behind the $\tau$-leaping family of approximate stochastic numerical methods is to execute multiple reaction occurrences as a single event instead of a executing only a single reaction per event. The time in which these reaction events occur is called the $\tau$-leap time and the number of reactions which occur in that time is Poisson distributed with rate $a_j\tau$. This leap time must be larger than the jump time of a single reaction event, but shorter than some unknown time in which the error of the Poisson approximation becomes too large. The multiple reaction occurrences are a "bundle" of reaction events that may occur anytime within the interval $[t, t+\tau]$. In order for the approximation to remain valid, the $\tau$-leap time must satisfy certain constraints: during the interval, (a) the reaction propensities must not significantly change and (b) the number of molecules of each chemical species must remain positive. If the $\tau$-leap time satisfies these constraints, then a simple numerical scheme may be derived. First, the method begins by calculating the reaction propensities of each reaction in the system. Second, an optimal $\tau$-leap time is calculated using some criteria. Third, the state of the system is updated by sampling the Poisson distributed number of reaction occurrences, for each reaction, that have occurred in the time interval

$[t, t+\tau]$, according to

$$X_i^{k+1} = X_i + \textstyle\sum_{j=1}^{M} \nu_{ji} \mathcal{K}_j(\underline{\mathrm{X}}(t)) \quad \mathcal{K}_j \sim \text{Poisson}(a_j(\underline{\mathrm{X}}(t))\,\tau) \tag{2.63}$$

Finally, the time is updated via $t = t + \tau$. The procedure is then repeated. By increasing the $\tau$-leap time, more reaction occurrences will occur in a single bundle, significantly decreasing the computational cost, but also reducing the accuracy of the approximation.

Conceptually, the $\tau$-leaping method is a valid approximation of a jump Markov process. However, in practice, the leap time rarely satisfies the above constraints. The reaction propensities of a system with a small number of molecules can dramatically change with only a few (or sometimes one) reaction occurrences. Consequently, the $\tau$-leaping time must be reduced to such a short time that only a few reaction occurrences are executed in a bundle, leading to only a small reduction in computational time. If the leap time is chosen too large, the number of molecules of species can often go negative, leading to an unrealistic and highly inaccurate trajectory of the system. The leap time must also be chosen dynamically because both the reaction propensities and the numbers of molecules of the species are changing over time, consequently changing the criteria for leap time selection.

Since the original $\tau$-leaping method was published, Daniel Gillespie has published five additional articles describing improvements to the method, including implicit versions [124, 125, 126, 127, 128]. In each publication, explicit or implicit versions of the $\tau$-leap method are proposed, each using slightly different criteria, in order to improve the robustness and accuracy of the numerical method. However, even in the latest publication in 2006, the same obstacle remains: if the leap time is too large, the approximation is invalid, and if the leap time is too small, the stochastic simulation algorithm is more efficient. Finding the correct leap time (which changes during the simulation) has proved to be very difficult. Why is this so?

Consider a stochastic differential equation driven only by a Poisson process,

$$X_i^{k+1} = X_i^k + \int_{t_i}^{t_{i+1}} dP_{\lambda(X,t')} \tag{2.64}$$

The $\tau$-leaping family of numerical schemes is, in fact, an Euler-like numerical method for solving this stochastic differential equation. Importantly, the Poisson process has a time-dependent rate $\lambda(\underline{X},t) = a_j(\underline{X}(t))\ t$. There is currently no Taylor series expansion for a Poisson driven stochastic differential equation and so it is difficult to estimate the generated error when solving this type of SDE using a specified time step. The error of this simple Euler-like scheme in terms of the time step has only been recently characterized [129] and is not first order with respect to $\tau$. In addition, no higher order numerical methods for solving a Poisson-driven SDE have been proposed. Now, compare this situation to a stochastic differential equation driven by a Wiener process. Its Taylor expansion is known and numerous higher order methods have been rigorously derived. Robust and capable numerical methods for the solution of Poisson driven SDEs will remain elusive until Poisson driven SDEs are as rigorously studied as their Wiener process counterparts. Without additional fundamental characterization of multidimensional coupled Poisson processes with time-dependent rates, further progress in developing $\tau$-leaping type methods that rely on Poisson processes will be slow.

Because of the difficulties in $\tau$-leaping and, more generally, with simulating a trajectory from a Poisson-driven SDE with time-dependent rates, two recent numerical methods have been proposed that substitute a Poisson process for a Binomial one [130, 131], although each implements the same idea in different ways. One may describe the number of reaction occurrences $n_j$ of the $j^{th}$ reaction within the interval $[t, t+\tau]$ according to the Binomial distribution, which is

$$P\left(n_j; p, n_j^{max}\right) = \frac{n_j^{max}!}{n_j!\,(n_j^{max} - n_j)!}\, p^{n_j}\,(1-p)^{n_j^{max}-n_j} \tag{2.65}$$

where $n_j^{max}$ is the maximum number of occurrences of the $j^{th}$ reaction that disallows any species from becoming negative and where $p = \min\left(a_j\tau/n_j^{max}\right)$. By using the Binomial distribution, one places an upper bound of $n_j^{max}$ occurrences on the $j^{th}$ reaction, thus preventing any reaction from occurring too many times and causing the number of molecules of any species to become negative. Consequently, by substituting the Binomial distribution for the Poisson one, the problem of negative numbers is avoided. After using Eq. (2.65) to sample the number of reaction occurrences for each reaction, $n_j$, the state and time is updated according to $X = X + \sum_j \nu_j n_j$ and $t = t + \tau$.

The main difference between the two proposed numerical methods is the way in which they sample from the multi-dimensional Binomial distribution. One method randomly selects a reaction, calculates $n_j^{max}$ and $p$, samples $n_j$, and repeats the random selection of the remaining reactions. The other method correctly samples the distribution by choosing the first reaction, calculating $n_1^{max}$ and $p$, sampling $n_1$, and then *conditioning* the number of occurrences of the second reaction on the the sampled value $n_1$. However, while the latter method's conditioning of the distribution is correct, it is difficult to implement for large, coupled reaction networks because of the dense conditioning between Binomial distributions. In addition, neither method characterizes the generated error in terms of the utilized time step and relates it to an expansion of the Binomial process in terms of leading orders. Such characterizations are necessary if the further development of leap-type methods continues beyond the prototypical stage.

### 2.3.4 Conclusions

The stochastic simulation algorithm will always remain the *exact* solution of a system of chemical reactions. Recent progress in leap-type methods have met significant obstacles in terms of both computational efficiency and accuracy. Instead, we have focused on the usage of Itô stochastic differential equations to approximate the number of reaction occurrences in a time step. This approximation is valid when both the reactant and product molecules of a reaction are above a certain threshold (typically 100 molecules) and when the reaction rate is fast enough (typically 10 molecules per second) [132]. Importantly, when both the number of reactant molecules and the reaction rate is large, the stochastic simulation algorithm becomes computationally intensive. Consequently, we use a valid approximation only in the kinetic regime where the stochastic simulation algorithm becomes the most slow, thus decreasing the computational cost of simulation while retaining accuracy. Finally, as we show in the next section, the fundamental characterization of Itô stochastic differential equations has progressed far beyond that of Poisson driven SDEs, allowing us to use rigorously derived numerical methods to efficiently and accurately compute solutions.

## 2.4 The Numerical Solution of Itô Stochastic Differential Equations

Stochastic differential equations can accurately model a wide variety of systems, including stock and derivative prices in a financial market, the Brownian dynamics of polymer molecules, the frequency of genetic alleles in a population of biological organisms, the movement of automobiles on roads, the spread of contagious diseases, and (of course) the stochastic dynamics of chemical reaction networks. Consequently, because these equations have so much utility, it is important to understand exactly how to solve these equations, including the technical details that arise in their solution. Accordingly, the numerical solution of stochastic differential equations deserves its own section.

Our primary motivation is to correctly and accurately solve the chemical Langevin equation, which describes the stochastic dynamics of a chemical system with $M$ reactions and $N$ species that satisfies certain continuity requirements. It has the form

$$dX_i = \sum_{j=1}^{M} \nu_{ji} r_j(X(t))\, dt + \sum_{j=1}^{M} \nu_{ji} \sqrt{r_j(X(t))}\, dW_j \quad i = 1,\ldots,N \tag{2.66}$$

where $X$ is the state vector of the number of molecules of each chemical species in the system, $\nu$ is the stoichiometric matrix that determines the effect of each chemical reaction on the chemical species in the system, and $r$ is the vector of reaction rates. In the previous section, we introduced the $dW$ as the differential of the Wiener process. In this section, we deepen our understanding of the Wiener process and describe, in detail, how one solves a differential equation that depends on it. Importantly, the chemical Langevin equation is one of the most difficult types of stochastic differential equations to solve; it is multi-dimensional with multiple, multiplicative, and non-communative Wiener processes. While many previously developed stochastic numerical methods treat the less difficult cases, we focus our attention on the methods that may solve the chemical Langevin equation.

In this section, we describe the technical details behind the numerical solution of Itô stochastic differential equations. We begin by stating definitions and describing the formal solution of linear SDEs. We continue presenting some examples of analytically solvables SDEs, which we later use to compare numerical and exact solutions. We then describe the difference between strong and weak solutions, the difference between Itô and Stratonovich stochastic integrals, and the methods of numerically evaluating or approximating these stochastic integrals. We conclude by detailing the different numerical schemes for solving Itô SDEs, including explicit, Runge-Kutta, implicit, and adaptive schemes.

### 2.4.1 Definitions and Formal Solutions

Consider a system with a state vector described by $N$ variables. The state vector is
$\underline{X} = \{X_1, X_2, \ldots, X_N\}$ where each variable is real-valued, or $X_i \in \Re$. The system is affected by one or more continuous-valued random processes, making the system a stochastic process. Because of the Central Limit theorem, a good model of a continuous-valued random process is the Wiener process, $W(t)$. For each random process affecting the system, an additional Wiener process is added. One may mathematically model the time evolution of a wide variety of such systems by using a Wiener-driven vector stochastic differential equation of the form

$$dX(t)_i = \alpha_i(X,t)\ dt + \sum_{j=1}^{M}\ \sigma_{i,j}(X,t)\ dW_j \tag{2.67}$$

where there are $j = 1, \ldots, M$ Wiener processes with $dW_j$ being the time derivative of the $j^{th}$ Wiener process and with state variables $i = 1, \ldots, N$. For brevity, the notational time dependence of the Wiener process has been suppressed. If both $\alpha_i$ and $\sigma_{i,j}$ (for all $i$, $j$) are linear or constant with respect to $X_t$, then the stochastic differential equation (SDE) is also linear. If $\sigma_{i,j}$ are all constant, then the SDE is said to have *additive noise*. Otherwise, if any $\sigma_{i,j}$ is a function of $X_t$, then the SDE has *multiplicative noise*. The stochastic differential equation in Eq. (2.67) is a useful notation for the integral form,

$$X(t)_i = \int_{t_o}^{t} \alpha_i(X,t')\, dt' + \int_{t_o}^{t} \sum_{j=1}^{M} \sigma_{i,j}(X,t)\, dW_j \tag{2.68}$$

where the integral of the Wiener process, $\int f(X,t)dW$, is called a *stochastic integral* and may be evaluated using either the Itô or Stratonovich definitions. In choosing a definition, the stochastic differential equation itself is classified as either an Itô or Stratonovich SDE. However, one may convert an Itô SDE to a Stratonovich one by a transformation of the drift coefficients.

Consider a scalar linear Itô SDE driven by a single Wiener process,

$$dX_t = (a_1(t)X_t + a_2(t))\ dt + (b_1(t)X_t + b_2(t))\ dW_1 \tag{2.69}$$

It has the formal solution

$$X_t = \Phi_{t,t_o}\left( X_{t_o} + \int_{t_o}^{t} (a_2(s) - b_1(s)b_2(s))\,\Phi_{s,t_o}^{-1}\, ds + \int_{t_o}^{t} b_2(s)\,\Phi_{s,t_o}^{-1}\, dW_1 \right) \tag{2.70}$$

where $\Phi_{t,t_o}$ is the fundamental solution,

$$\Phi_{t,t_o} = \exp\left( \int_{t_o}^{t} \left( a_1(s) - \frac{1}{2}b_1^2(s) \right) ds + \int_{t_o}^{t} b_1(s)\, dW_1 \right) \tag{2.71}$$

The mean $\mu(t)$ and variance $v(t)$ of the solution to Eq. (2.70) satisfies two ordinary differential equations

$$\frac{d\mu(t)}{dt} = a_1(t)\mu(t) + a_2(t) \tag{2.72}$$

and

$$\frac{dv(t)}{dt} = \left( 2a_1(t) + b_1^2(t) \right) v(t) + b_1(t)b_2(t)\mu(t) + b_1^2\mu^2(t) + b_2^2 \tag{2.73}$$

Importantly, the solution of a homogenous linear SDE with $a_1(t) \equiv a$, $b_1(t) \equiv b$, $a_2(t) = b_2(t) = 0$, and initial condition $X_{t_o} \sim \mathrm{N}(\mu_o, v_o)$ for any finite $\mu_o$, $v_o$, is also a Gaussian distribution with a mean and variance governed by Eqs. (2.72) and (2.73). Otherwise, the solution is not generally a Gaussian distribution. For a vector system of multiple linear SDEs driven by multiple Wiener processes, spectral decomposition of the drift vector and diffusion tensor can yield a system of decoupled linear SDEs driven by a single altered Wiener process whose formal solution remains Eq. (2.70).

### 2.4.2 Explicit Solutions of Some Stochastic Differential Equations

The formal solution of a linear stochastic differential equation always exists and may be evaluated in many instances. These analytic solutions are helpful when validating a stochastic numerical integrator or measuring its error with respect to the time step. In the first example, the following non-homogeneous scalar linear Itô SDE with additive noise,

$$dX_t = \left(\frac{b - X_t}{T - t}\right) dt + dW_1 \tag{2.74}$$

is satisfied by the solution

$$X_t = X_o \left(1 - \frac{t}{T}\right) + b\frac{t}{T} + (T - t)\int_0^t \frac{1}{T - s}\, dW_1 \tag{2.75}$$

In the second example, a homogenous scalar linear Itô SDE with multiplicative noise, or

$$dX_t = aX_t\, dt + bX_t\, dW \tag{2.76}$$

is satisfied by the solution

$$X_t = X_{t_o} \exp\left(\left(a - \frac{1}{2}b^2\right)t + bW(t)\right) \tag{2.77}$$

Because the formal solution of any linear SDE exists and may be evaluated in many instances, one technique for solving a non-linear system of SDEs is to convert it to a linear one. There are a variety of techniques to do this, including non-linear transformations and redefining a single non-linear SDE into a system of coupled linear SDEs. For example, to solve the following non-linear scalar Itô SDE

$$dX_t = (aX_t^n + bX_t)\; dt + cX_t\, dW_1 \tag{2.78}$$

the transformation $y = x^{1-n}$ converts the non-linear SDE into a linear one that satisfies

$$X_t = \Theta_t \left(X_o^{1-n} + a(1-n)\int_o^t \Theta_s^{n-1}\, ds\right)^{\frac{1}{1-n}} \tag{2.79}$$

with

$$\Theta_t = \exp\left(\left(b - \frac{1}{2}c^2\right)t + cW(t)\right) \tag{2.80}$$

Stochastic differential equations may also describe a complex-valued stochastic process driven by real-valued Wiener processes.

### 2.4.3 Strong and Weak Solutions

The formal solution of a general stochastic differential equation in Eq. (2.67), including the ones in Eq. (2.75), (2.77), and (2.79), may be evaluated in two different ways. The first is to generate a *trajectory* of the solution by creating a single path for each Wiener process in the system and then evaluating the state vector, $X_t$, in terms of the Wiener processes and their integrals. The second is to find a *probability distribution* $P(X_t)$ that is equivalent to the distribution of all possible trajectories of the system. The former is called a *strong solution* while the latter is called a *weak solution*.

Consider the generalized vector stochastic differential equation with multiple Wiener processes, shown in Eq. (2.67). It has a drift vector $A_i = \alpha_i$ and diffusion tensor $D^2_{i,j} = \frac{1}{2}\sum_k \sigma_{i,k}\sigma^T_{k,j}$. If both terms are bounded and the diffusion tensor is positive definite, then Eq. (2.67) has a weak solution $P(X_t, t)$ that satisfies the Fokker-Planck equation

$$\frac{\partial P(X_t,t)}{\partial t} = \left[ -\sum_{i=1}^{N} \frac{\partial}{\partial X_i} A_i(X_t,t) + \sum_{i=1}^{N} \sum_{j=1}^{M} \frac{\partial^2}{\partial X_i^2} D^2_{i,j}(X_t,t) \right] P(X_t,t) \quad (2.81)$$

which is a partial differential equation with $N$ dimensions.

It is possible that two different systems of stochastic differential equation have the same weak solution, but different strong solutions. For example, a two-dimensional system that oscillates out of phase in a clockwise direction will have a different strong solution than one that oscillates out of phase in a counter-clockwise direction. However, both of their weak solutions will be the same. The strong solution also contains more information than the weak solution. By generating many trajectories of each Wiener process and evaluating an *ensemble* of strong solutions of the stochastic differential equation, one may compute the distribution of that ensemble to obtain the weak solution. However, the opposite is not generally true. Except in certain cases, one may not sample a probability distribution of a time-dependent system to generate a strong solution.

The strong solution of a general system of SDEs typically has no formal solution. Instead, one must use a stochastic numerical integrator to simulate a trajectory of the stochastic differential equation. A stochastic numerical method generates a strong approximation to the solution of an SDE if the approximate numerically generated trajectory converges to an exact trajectory of the SDE, given that the paths of the Wiener processes are fixed and as the time step approaches zero. In other words, if the time step of a stochastic numerical integrator is $\Delta t_{SDE} = t_{i+1} - t_i$ and if the paths of all Wiener processes in the system at times $t_i$ for $i = 1, \ldots, n$ are fixed as $\{W_j(t_1), W_j(t_2), \ldots, W_j(t_n)\}$ for $j = 1, \ldots, M$, then the exact trajectory of a $d$-dimensional stochastic differential equation can be evaluated at times $t_i$ to be $\{X_k(t_1), X_k(t_2), \ldots, X_k(t_n)\}$ for $k = 1, \ldots, d$. The numerical approximation of the solution of the SDE that uses the same paths of the Wiener processes is $\{\hat{X}_k(t_1), \hat{X}_k(t_2), \ldots, \hat{X}_k(t_n)\}$ and is considered a strong approximation of the SDE if

$$\left|\hat{X}_k(t_i) - X_k(t_i)\right| \to 0 \tag{2.82}$$

as $\Delta t_{SDE} \to 0$. Consequently, a stochastic numerical method that generates a strong approximation to the solution of a stochastic differential equation will converge, in the limit, to the exact solution in a path-wise sense. In the remainder of this section, we focus on stochastic numerical methods that generate strong solutions to non-linear stochastic differential equations.

### 2.4.4 Itô and Stratonovich Stochastic Integrals

The integral of the Wiener process may be interpreted in many ways. The two most important definitions are the Itô and Stratonovich ones. These different definitions have practical consequences and are briefly reviewed.

Consider a time interval $[0, T]$ divided into $n$ equal partitions $0 = t_1^{(n)} < t_2^{(n)} < \cdots < t_n^{(n)} < t_{n+1}^{(n)} = T$. A typical Riemann integral is defined as the mean square limit of the sum of a twice differentiable and continuous function $f(t)$ evaluated at each $\xi_i^{(n)}$ over the time interval $[t_i^{(n)}, t_{i+1}^{(n)}]$, or

$$\int_0^T f(X_t, t)dt = \sum_{i=1}^{n} f\left(X_{\xi_i^{(n)}}, \xi_i^{(n)}\right)\left(t_{i+1}^{(n)} - t_i^{(n)}\right) \tag{2.83}$$

as $n \longrightarrow \infty$ and $t_{i+1}^{(n)} - t_i^{(n)} \longrightarrow 0$. The evaluation times, $\xi_i^{(n)}$, are usually taken to be $\xi_i^{(n)} = t_i^{(n)}$. Importantly, the value of a Riemann integral does not change if $\xi_i^{(n)}$ is chosen to be any other time, $t_i^{(n)} \leq \xi_i^{(n)} \leq t_{i+1}^{(n)}$. However, for a stochastic integral, the *choice of these evaluation times actually affects its value*.

Over the same time interval $[0,T]$, a stochastic integral $\int_0^T f(X_t,t)dW_t$ is also defined as the mean square limit of a function $f(t)$ evaluated at each $\xi_i^{(n)}$ over the time interval $[t_i^{(n)}, t_{i+1}^{(n)}]$, or

$$S_n = \sum_{i=1}^{n} f\left(X_{\xi_i^{(n)}}, \xi_i^{(n)}\right) \left\{W_{t_{i+1}}^{(n)} - W_{t_i}^{(n)}\right\} \tag{2.84}$$

as $n \longrightarrow \infty$ and $t_{i+1}^{(n)} - t_i^{(n)} \longrightarrow 0$. The Itô interpretation chooses the time evaluation points to be $\xi_i^{(n)} = t_i^{(n)}$ while the Stratonovich interpretation chooses the evaluation points to be $\xi_i^{(n)} = (t_i^{(n)} + t_{i+1}^{(n)})/2$. The usage of the Stratonovich definition of the stochastic integral is denoted by $\int_0^T f(X_t,t) \circ dW_t$. Importantly, the choice of $\xi_i^{(n)}$ is arbitrary, but the Itô and Stratonovich choices are the only commonly used ones because the resulting stochastic calculi have certain important behaviors.

For example, the Itô stochastic integral $\int_0^T X_t dW_t$ is

$$\int_0^T X_t dW_t = \sum_{i=1}^{n} X_{t_i} \left\{W_{t_{i+1}} - W_{t_i}\right\} = X_o \exp\left(W_T - \frac{1}{2}T\right) \tag{2.85}$$

using the same mean square limit where $X_o = X_{(t=0)}$. Conversely, the Stratonovich stochastic integral $\int_0^T X_t \circ dW_t$ is

$$\int_0^T X_t \circ dW_t = \sum_{i=1}^{n} \frac{X_{t_i} + X_{t_{i+1}}}{2} \left\{W_{t_{i+1}} - W_{t_i}\right\} = X_o \exp\left(W_T\right) \tag{2.86}$$

Consequently, the Stratonovich stochastic calculus is commonly used because it reproduces the familiar calculus described by the fundamental theorem of calculus. The Itô stochastic calculus is distinctly different. The Itô definition of the stochastic integral is commonly used because it is *non-anticipative* and a *martingale*. It is non-anticipative because its function evaluations depend only on $t_i^{(n)}$ and not any future values of the function. It is a martingale because the expectation of an Itô stochastic integral is always zero, or

$$E\left[\int_0^T f(X_t,t)dW_t\right] = 0 \tag{2.87}$$

The same expectation for a Stratonovich integral is

$$E\left[\int_0^T f(X_t,t)\circ dW_t\right] = \frac{1}{2}\int_0^T E\left[\frac{\partial f(X_t,t)}{\partial X}\right]dt \tag{2.88}$$

which is not necessarily zero. Both stochastic integrals may also have non-zero variances and other moments.

Importantly, it is straightforward to convert an Itô stochastic differential equation to a Stratonovich one. The Itô SDE

$$dX_t = a(X_t,t)\ dt + b(X_t,t)\ dW_t \tag{2.89}$$

is equivalent to the Stratonovich SDE

$$dX_t = \underline{a}(X_t,t)\ dt + b(X_t,t)\circ dW_t \tag{2.90}$$

where

$$\underline{a}(X_t,t) = a(X_t,t) - \frac{1}{2}b(X_t,t)\frac{\partial b(X_t,t)}{\partial X} \tag{2.91}$$

One may also define multiple Itô or Stratonovich stochastic integrals that integrate a function $f(X_t,t)$ over multiple Wiener processes and/or time. For example, if there are $M$ Wiener processes in the system, then any of the following single and double Itô stochastic integrals

$$\int f(X_t,t)\,ds \quad \int f(X_t,t)\,dW_{s,\,j} \quad \int\int f(X_t,t)\,dW_{z,\,j}\,ds \quad \int\int f(X_t,t)\,dW_{z,\,j_1}\,dW_{s,\,j_2} \tag{2.92}$$

may be evaluated over all unique combinations of $j_1, j_2 = 1,\ldots,M$. If integrating over two different Wiener processes, $j_1 \neq j_2$, both the Itô and Stratonovich double stochastic integrals converge to the same value. Otherwise, the remaining single and double stochastic integrals converge to different values.

The usage of either definition depends on the application. If the system of interest has reasonably smooth derivatives and is being modeled as a random process because it is more convienant to do so, then the Stratonovich definition may be more applicable because its calculus is equivalent to the calculus of ordinary differential equations. A Stratonovich stochastic differential equation is often a good model of a real random process affected by a random variable with a finite correlation, which is called colored noise. For example, in finance, Stratonovich SDEs are often used to model the dynamics of derivative and stock prices. Conversely, if the system is a physical or chemical process with uncorrelated transitions that depend only on the present time, then the Itô stochastic integral may be more appropriate. In addition, it is convienant to use the Itô definition if the martingale property is required. In stochastic chemical kinetics, the collision and reaction of molecules are modeled by using Itô stochastic differential equations because the physical quantity in question, which is the number of reaction events in a time increment, depends only on current number of molecules and has zero correlation.

If the calculus of Stratonovich stochastic integrals is the same as ordinary calculus, then what is the calculus of stochastic Itô integrals? The Itô Formula provides the answer.

### 2.4.5 The Itô Formula and Itô-Taylor Expansions

The Itô formula is a general transformation analogous to the chain rule of ordinary calculus. The key difference is that mean square limit of the integral of $(dW_t)^2$ goes to $dt$ instead of $0$ as in ordinary calculus, thereby creating additional terms of $O(dt)$. Given a $N$-dimensional stochastic differential equation with a $M$-dimensional Wiener process, written in vector form,

$$dX_t = \alpha_t \, dt + \sigma_t \, dW_t \tag{2.93}$$

and some transformation $Y_t = \mathbb{U}(t, X_t) = U(t, X_t^1, X_t^2, \ldots, X_t^N)$, the Itô Formula is an expression for $dY_t$ in terms of the time-dependent drift vector and diffusion tensor, $\alpha_t$ and $\sigma_t$, and the transformation $\mathbb{U}(t, X_t)$. In vector form, the Itô Formula is

$$dY_t = \left\{ \frac{\partial \mathbb{U}}{\partial t} + \alpha_t^T \triangledown \mathbb{U} + \frac{1}{2} \mathrm{tr} \left( \sigma_t \sigma_t^T \triangledown \left[ \triangledown \mathbb{U} \right] \right) \right\} dt + \triangledown \mathbb{U}^T \sigma_t \, dW_t \tag{2.94}$$

For example, given the scalar SDE $dX_t = f_t \, dW_t$ with a single Wiener process and the transformation $Y_t = U(t, X_t) = \exp(X_t)$, an application of the Itô Formula yields

$$dY_t = \frac{1}{2} f_t^2 Y_t \, dt + f_t Y_t \, dW_t \tag{2.95}$$

To compare, using the same transformation on the equivalent Stratonovich integral, $dX_t = f_t \circ dW_t$, yields $dY_t = f_t Y_t \circ dW_t$, reproducing the familar chain rule of ordinary calculus.

The Itô-Taylor expansion is a generalization of the deterministic Taylor expansion for Itô stochastic differential equations. It describes how a stochastic differential equation may be linearized in time with respect to a basis set of integrals. The expansion arises from the repeated application of the Itô Formula on selected terms.

Consider the scalar Itô stochastic differential equation in integral form

$$X_t = X_{t_o} + \int_{t_o}^{t} a(X_s) \, ds + \int_{t_o}^{t} b(X_s) \, dW_s \tag{2.96}$$

If the Itô Formula is applied to Eq. (2.96) with a transformation $Y_t = f(X_t)$, where $f(X_t)$ is any continuous and twice differentiable function, then the result is

$$Y_t = f(X_t) = f(X_{t_o}) + \int_{t_o}^{t} L^0 f(X_s) \, ds + \int_{t_o}^{t} L^1 f(X_s) \, dW_s \tag{2.97}$$

for $t \in [0, T]$. The $L^0$ and $L^1$ operators are defined as

$$L^0 = a \frac{\partial}{\partial x} + \frac{1}{2} b^2 \frac{\partial^2}{\partial x^2} \tag{2.98}$$

and

$$L^1 = b \frac{\partial}{\partial x} \tag{2.99}$$

If $f(x) = x$ is substituted into Eq. (2.97), then the original integral form of the stochastic differential equation in Eq. (2.96) is obtained via $L^0 f = a$ and $L^1 f = b$. The solution $X_t$ can be rewritten into a term that depends on $X_{t_o}$ and a residual that depends on time and $X_t$, or

$$X_t = X_{t_o} + R_0 \tag{2.100}$$

with

$$R_0 = \int_{t_o}^{t} a(X_s) \, ds + \int_{t_o}^{t} b(X_s) \, dW_s \tag{2.101}$$

This transformation is the $0^{th}$ order Itô-Taylor expansion of the solution $X_t$ around $X_{t_o}$, approximating the solution at time $t$ as $X_{t_o}$. The residual of this approximation, $R_0$, is of order $O(1)$ and determines the leading order of the error. In order to decrease the error, the Itô Formula must be reapplied on terms in the residual.

By applying the Itô Formula to the coefficients in the residual $R_0$, such that $f(x) = a$ and $f(x) = b$ are each substituted into Eq. (2.97), the final result,

$$X_t = X_{t_o} + a(X_{t_o}) \int_{t_o}^{t} ds + b(X_{t_o}) \int_{t_o}^{t} dW_s + R_1 \tag{2.102}$$

is the first Itô-Taylor expansion of $X_t$ around $X_{t_o}$ with a remainder, $R_1$,

$$\begin{aligned} R_1 = &\int_{t_o}^{t} \int_{t_o}^{s} L^0 a(X_z)\, dsdz + \int_{t_o}^{t} \int_{t_o}^{s} L^1 a(X_z)\, dW_z ds \\ &+ \int_{t_o}^{t} \int_{t_o}^{s} L^0 b(X_z)\, dW_s dz + \int_{t_o}^{t} \int_{t_o}^{s} L^1 b(X_z)\, dW_z dW_s \end{aligned} \tag{2.103}$$

of order $O(dt)$. By repeatedly applying the Itô Formula to the lowest order terms in the remainder, the resulting expression will be a continued expansion in terms of the multiple Itô integrals, such as $\int ds$, $\int dW_s$, and $\int\int dW_s dW_z$.

For example, by selecting the $L^1 b \propto O(dt)$ term from the remainder in Eq. (2.103) and substituting $f(x) = L^1 b$ into the Itô Formula, the result

$$X_t = X_{t_o} + a(X_{t_o}) \int_{t_o}^{t} ds + b(X_{t_o}) \int_{t_o}^{t} dW_s + L^1 b(X_{t_o}) \int_{t_o}^{t} \int_{t_o}^{s} dW_z dW_s + R_2 \tag{2.104}$$

is the second Itô-Taylor expansion of $X_t$ around $X_{t_o}$ with a remainder, $R_2$,

$$\begin{aligned} R_2 = &R_1 - \int_{t_o}^{t} \int_{t_o}^{s} L^1 b(X_z)\, dW_z dW_s \\ &+ \int_{t_o}^{t} \int_{t_o}^{s} \int_{t_o}^{z} L^0 L^1 b(X_u)\, dW_z dW_s du + \int_{t_o}^{t} \int_{t_o}^{s} \int_{t_o}^{z} L^1 L^1 b(X_u)\, dW_u dW_z dW_s \end{aligned} \tag{2.105}$$

of order $O(dt^{3/2})$. By expanding the double stochastic integral $\int\int dW_s dW_z$ of order $O(dt)$, the remainder loses its lowest order term, but gains a triple stochastic integral $\int\int\int dW_u dW_z dW_s$ of order $O(dt^{3/2})$. Notice that the $dt$

terms inside integrals contribute $O(dt)$ to the integral, while the $dW$ terms inside integrals contribute $O\left(\sqrt{dt}\right)$ to the integral. Therefore, in order to obtain an Itô-Taylor expansion that is accurate up to $O(dt^n)$, the expansion must contain a term integrating a Wiener process $2n$ times and its remainder must contain a term integrating a Wiener process by at least $2n+1$ times.

By using the Itô Formula to repeatedly expand terms inside the remainder, the Itô-Taylor expansion of $X_t$ may be continued arbitrarily long. However, the difficulty in evaluating these expansions is the presence of multiple stochastic integrals. For a system of stochastic differential equations driven by $M$ Wiener processes, the Itô-Taylor expansion will include $M+1$ single stochastic integrals, $(M+1)(M+2)/2$ double stochastic integrals, and so on, which includes the integrals containing $dt$. For example, the lowest order term in the remainder $R_1$ will be the double stochastic integrals, $\int_0^T \int_0^s dW_{s,j_1} dW_{z,j_2}$ for $j_1, j_2 = 1, \ldots, M$. In the next section, the evaluation of these multiple stochastic integrals is briefly reviewed.

### 2.4.6 Numerical Generation of Stochastic Integrals

The stochastic integrals that appear in an Itô-Taylor expansion must be numerically evaluated. The single stochastic integrals, the double stochastic integrals containing a $dt$, and the double stochastic integrals over the same Wiener process may all be represented in terms of the constituent Wiener increments and time. Consequently, it is straightforward to evaluate these integrals in terms of a Gaussian random variable and the time increment. However, the double stochastic integrals over two different Wiener processes may not be expressed in terms of their constituent Wiener processes and time. Instead, they must be numerically approximated. Because the goal is to develop strong, pathwise approximations to the solutions of a stochastic differential equation, the approximation of these multiple stochastic integrals must also be of the strong type.

Consider a stochastic differential equation driven by $j = 1, \ldots, M$ Wiener processes. On the time interval $[0, \Delta t]$ the Wiener increments are $\Delta W_{j,(\Delta t)}$ and each Wiener process may participate in one or more stochastic integrals. The values of the $M$ single Itô stochastic integrals are denoted by

$$I_{(j),\,\Delta t} = \int_0^{\Delta t} dW_{j_1} = \Delta W_{j,(\Delta t)} \tag{2.106}$$

and are evaluated as a Gaussian random number with zero mean and $\Delta t$ variance, $\Delta W_{j,(\Delta t)} \sim \mathsf{N}(0,\Delta t)$.

However, a double Itô stochastic integral on $[0,\Delta t]$, or

$$I_{(j_1,j_2),\,\Delta t} = \int_0^{\Delta t} \int_0^{\Delta t'} dW_{j_1} dW_{j_2} \tag{2.107}$$

may only be represented by its constituent Wiener processes in the case $j_1 = j_2$. Otherwise, it must be numerically approximated. In the $j_1 = j_2$ case, the double Itô stochastic integral is evaluated as

$$I_{(j_1,j_1),\,\Delta t} = \frac{1}{2}\left\{\left(\Delta W_{j,(\Delta t)}\right)^2 - \Delta t\right\} \tag{2.108}$$

Otherwise, for $j_1 \neq j_2$, the $M$-dimensional Wiener processes must be written as a componentwise Fourier expansion

$$W_{j,(t)} - \frac{t}{\Delta t} W_{j,(\Delta t)} = \frac{1}{2} a_{j,0} + \sum_{r=1}^{\infty} \left( a_{j,r} \cos\left(\frac{2r\pi t}{\Delta t}\right) + b_{j,r} \sin\left(\frac{2r\pi t}{\Delta t}\right)\right) \tag{2.109}$$

with Gaussian distributed coefficients

$$a_{j,r} = \frac{2}{\Delta t} \int_0^{\Delta t} \left(W_{j,(s)} - \frac{s}{\Delta t} W_{j,(\Delta t)}\right) \cos\left(\frac{2r\pi s}{\Delta t}\right)\, ds \tag{2.110}$$

and

$$a_{j,r} = \frac{2}{\Delta t} \int_0^{\Delta t} \left(W_{j,(s)} - \frac{s}{\Delta t} W_{j,(\Delta t)}\right) \sin\left(\frac{2r\pi s}{\Delta t}\right)\, ds \tag{2.111}$$

for $j = 1,\ldots,M$ and $r = 0,1,\ldots,\infty$. The Fourier series converges in the mean square sense and with coefficients $a_{j,r}$ and $b_{j,r} \sim \mathsf{N}(0,\Delta t/2\pi^2 r^2)$.

Because of the $1/r^2$ dependence, the series may be truncated with $p$ terms. By rearranging and using additional relationships between different types of stochastic integrals (see Kloeden and Platen [113] for details), one may numerically evaluate a double Itô stochastic integral by using

$$\begin{aligned} I_{(j_1,j_2),\Delta t} &= \Delta t \left( \frac{1}{2}\xi_{j_1}\xi_{j_2} + \sqrt{\rho_P}\left(\mu_{j_1,p}\xi_{j_2} - \mu_{j_2,p}\xi_{j_1}\right) \right) \\ &+ \frac{\Delta t}{2\pi}\sum_{r=1}^{p}\frac{1}{r}\left(\zeta_{j_1,r}\left(\sqrt{2}\xi_{j_2}+\eta_{j_2,r}\right) - \zeta_{j_2,r}\left(\sqrt{2}\xi_{j_1}+\eta_{j_1,r}\right)\right) \end{aligned} \tag{2.112}$$

where

$$\rho_p = \frac{1}{12} - \frac{1}{2\pi^2}\sum_{r=1}^{p}\frac{1}{r^2} \tag{2.113}$$

The Gaussian coefficients are sampled via $\mu_{j,p},\ \eta_{j,r},\ \zeta_{j,r} \sim \mathsf{N}(0,1)$ for $r = 1,\ldots,p$ while $\xi_j$ are related to the Wiener processes in the system via $\xi_j = \frac{1}{\sqrt{\Delta t}}\Delta W_{j,(\Delta t)}$. By increasing the number of retained terms in the expansion, the numerical error in the approximation of the double stochastic integral is reduced. A typical value of $p = 10$ ensures a reasonable approximation.

We use Eqs. (2.106), (2.108), and (2.112) to numerically evaluate the single and double Itô stochastic integrals in the following stochastic numerical integration schemes.

### 2.4.7 Itô-Taylor Explicit Numerical Schemes

By truncating an Itô-Taylor expansion at a convienant remainder, it is possible to create a "one step" explicit numerical scheme for calculating $X_{t+\Delta t}$ in terms of $X_t$ and the coefficients and stochastic integrals evaluated at $X_t$. The lowest order term in the remainder will determine the accuracy of the numerical scheme. These Itô-Taylor numerical schemes include the Euler-Maruyama, Milstein, Platen, and higher order methods. The Euler-Maruyama method is perhaps the most common one because it does not require the evaluation of double stochastic integrals. However, as we will see, the limitations of the Euler-Maruyama method can make higher-order methods more appealing. Each stochastic numerical method approximates the solution of a stochastic differential equation with different degrees of accuracy. In addition, numerical stability of the scheme is also an important topic. In the next part, we review the definitions of numerical accuracy and stability in the context of a stochastic numerical method.

### Numerical Accuracy and Stability

There are two definitions of numerical accuracy in the context of a stochastic numerical integrator: weak and strong accuracy. Weak accuracy measures the generated numerical error with respect to the probability distribution of the solution while strong accuracy measures the numerical error with respect to the trajectory of the solution.

A stochastic numerical integrator is said to have *strong accuracy* of order $\gamma$ if the pathwise strong convergence of a numerically generated trajectory, $Y(t_i)$, towards the exact trajectory, $X(t_i)$, follows

$$\varepsilon_{strong} = |Y(t_i) - X(t_i)| = K\Delta t^{\gamma} \tag{2.114}$$

where $\Delta t$ is the time step of the stochastic numerical integrator and for some $K > 0$. A stochastic numerical integrator is said to have *weak accuracy* of order $\beta$ if the expectation of some function of the approximation solution, $E\left[f(Y(t_i), t)\right]$, towards the expectation of some function of the exact solution, $E\left[f(X(t_i))\right]$, follows

$$\varepsilon_{weak} = \mid E\left[f(Y(t_i))\right] - E\left[f(X(t_i))\right] \mid = K\Delta t^{\beta} \tag{2.115}$$

for some $K > 0$.

To measure the strong accuracy of a stochastic numerical method, it is helpful to numerically solve a stochastic differential equation that has an analytical solution. We measure the strong accuracy of a stochastic numerical method by first fixing the Wiener processes in the system over a time interval $[0, T]$ with constant time increments, $\Delta t = t_{i+1} - t_i$ for $i = 1, \ldots, n$. For a single Wiener process, the path is $W(t) = \{W(t_1), W(t_2), W(t_3), \ldots, W(t_n)\}$. The exact trajectory of the solution of the SDE is then evaluated on $[0, T]$ as $X_t = \{X(t_1), X(t_2), X(t_3), \ldots, X(t_n)\}$. The numerical approximation of the solution is then calculated using a numerical scheme so that $Y(t) = \{Y(t_1), Y(t_2), Y(t_3), \ldots, Y(t_n)\}$. The numerical scheme will use the time step, $\Delta t$, and the Wiener increments, $\Delta W(t_{i+1}) = W(t_{i+1}) - W(t_i)$, in the calculations. The strong definition of the error, $\varepsilon_{strong}$, may then be measured in absolute terms via Eq. (2.114). However, because the error is only evaluated over as single path of the Wiener processes, $\varepsilon_{strong}$ is a random variable; the exact value will fluctuate according to a Gaussian distribution. By repeating the above procedure over many paths of the Wiener processes, the mean of the Gaussian distribution, $<\varepsilon_{strong}>$, may be calculated. This mean will then satisfy Eq. (2.114) with respect to the utilized time step $\Delta t$ for some $\gamma$. The variance of the Gaussian distribution weakly converges to zero in the limit of an infinite number of evaluated trajectories.

We can also measure the weak accuracy of a numerical scheme by taking the expectation of the exact solution of a stochastic differential equation and comparing it, using Eq. (2.115), to the expectation of the approximate numerical solution, $E\left[Y(t_i)\right] = \frac{1}{N}\sum_{i=1}^{N} Y(t_i)$, evaluated over $N$ different Wiener paths as described above.

The numerical stability of a stochastic numerical integrator is also important. If the numerical scheme is unstable, the propagation of initial and roundoff errors will cause the numerical error in the solution to blow up. This is especially important when solving *stiff* stochastic differential equations where there is a large time-scale separation. The following $N$-dimensional linear SDE driven by $M$ Wiener processes

$$dX_t = \mathbb{A} X_t \, dt + \sum_{j=1}^{M} \mathbb{B} X_t \, dW_{t,j} \tag{2.116}$$

is considered stiff if the minimum and maximum Lyapunov exponents of the drift and diffusion matrices, $\mathbb{A}$ and $\mathbb{B}$, are largely separated such that $\lambda_N \ll \lambda_1$ where $\lambda_1$ is the top Lyapunov exponent. Unlike ordinary differential equations, stiffness can arise due to a time-scale separation in either the drift or diffusion terms. To measure asymptoptic stochastic stability, one may substitute a "test equation" with $\mathbb{A} = \lambda_{dr}$ and $\mathbb{B} = \lambda_{du}$ into the numerical scheme and compare the generated error in the $(n+1)^{th}$ iteration versus the $n^{th}$ iteration. Asymptoptic stochastic stability is ensured if the error in the $(n)^{th}$ iterate of the solution does not grow with $n$ such that

$$\left|Y^{(n)} - X^{(n)}\right| \leq \left|g(\lambda_{dr}, \lambda_{du}, \Delta t)\right|^n \left|Y^{(0)} - X^{(0)}\right| \tag{2.117}$$

for initial conditions $Y^{(0)}$ and $X^{(0)}$. The function $g$ depends on the numerical scheme and test equation. For all Itô-Taylor explicit numerical schemes,

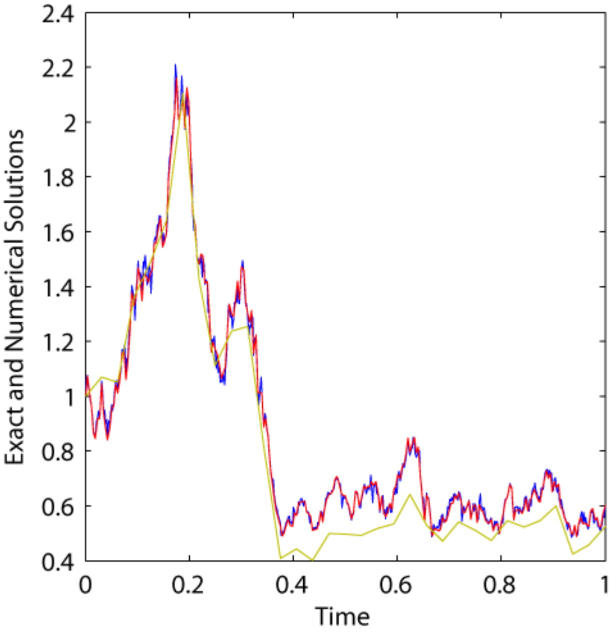

Figure 2.9: Using the Euler-Maruyama scheme, the numerical solution of the simple linear stochastic differential equation in Eq. (2.119) is calculated using a time step of (red) $\Delta t = 2^{-7}$ or (yellow) $\Delta t = 2^{-5}$ and compared to the (blue) exact solution.

asymptoptic stochastic stability is ensured if, at least, the real part of the top Lyapunov exponent satisfies Re$(\lambda_1)\Delta t < 1$.

### The Euler-Maruyama Method

The Euler-Maruyama method is an explicit stochastic numerical method resulting from the truncation of the first Itô-Taylor expansion of the solution $X_t$ around $X_{t_o}$. The lowest order term in the Euler-Maruyama method is a single stochastic integral of strong order $O\left(\sqrt{\Delta t}\right)$ and weak order $O(\Delta t)$. Therefore, the strong order of accuracy is $\gamma = 0.5$ while the weak order of accuracy is $\beta = 1.0$.

For a generalized $N$-dimensional stochastic differential equation driven by $M$ Wiener process, the Euler-Maruyama scheme for the $n^{th}$ iterate of the $k^{th}$ component of $Y$ is

$$Y_k^{(n+1)} = Y_k^{(n)} + a_k\,\Delta t + \sum_{j=1}^{M} b_{k,j}\,\Delta W_j \tag{2.118}$$

for $k = 1, \ldots, N$ and where $\Delta W_j \sim \mathrm{N}(0, \Delta t)$ for $j = 1, \ldots, M$. The $a_k$ is the drift term for the $k^{th}$ component while the $b_{k,j}$ term is the diffusion term for the effect of the $j^{th}$ Wiener process on the $k^{th}$ component. Both terms are evaluated using the current state of the system, $Y^{(n)}$.

For an example, the numerical solution of the linear stochastic differential equation

$$dX_t = aX_t\, dt + b_1\, dW_1 \tag{2.119}$$

with $a = -0.5$, $b_1 = 1$, and initial condition $X_{t_o} = Xo$ is compared to its exact solution

$$X_t = X_o \exp\left(\left(a + \frac{1}{2}b_1^2\right)t + b_1 W(t)\right) \tag{2.120}$$

using a fixed time step $\Delta t$. In Figure 2.9, we compare the exact and numerical solutions of Eq. (2.119) with time steps of $\Delta t = 2^{-7}$ and $\Delta t = 2^{-5}$, *each evaluated over the same Wiener path*. By using the same Wiener path, we may then calculate the strong pathwise accuracy of the Euler-Maruyama scheme. In Figure 2.10, we calculate the strong order of accuracy, $\gamma$, by repeating the evaluation of the numerical and exact solutions over 500 different Wiener paths, plotting the average $\log_2 \varepsilon_{strong}$ vs. $\log_2 \Delta t$, determining the slope of the linear line. The Euler-Maruyama scheme's strong order of accuracy $\gamma = 0.5$ is verified.

### The Milstein Method

The Milstein scheme is an explicit stochastic numerical method resulting from the truncation of the second Itô-Taylor expansion [133]. The lowest order term in the Milstein scheme is a double stochastic integral of both strong and weak order $O(\Delta t)$. Consequently, both the strong and weak order of accuracy is $\gamma = \beta = 1$.

For a general $N$-dimensional stochastic differential equation driven by $M$ Wiener processes, the Milstein scheme for the $n^{th}$ iterate is

$$Y_k^{(n+1)} = Y_k^{(n)} + a_k \Delta t + \sum_{j=1}^{M} b_{k,j} \Delta W_j + \sum_{j_1, j_2 = 1}^{M} \sum_{i=1}^{N} b_{i,j_1} \frac{\partial b_{k,j_2}}{\partial x_k} I_{(j_1, j_2)} \tag{2.121}$$

for $k = 1, \ldots, N$. The double stochastic integrals in Eq. (2.121) are evaluated using Eqs. (2.108) and (2.112).

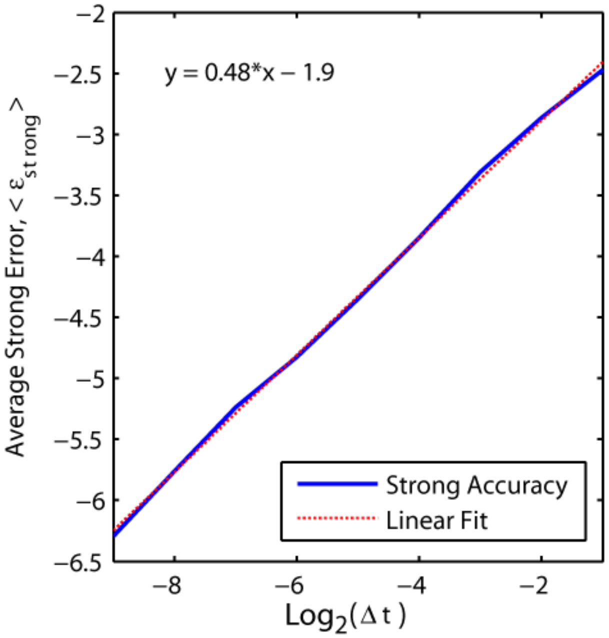

Figure 2.10: The strong order of accuracy of the Euler-Maruyama scheme is determined by calculating the slope of a linear fit of a log-log graph of the average absolute error between the numerical and exact solutions, $<\varepsilon_{strong}>$, versus the time step of the Euler-Maruyama scheme, $\Delta t$. The strong order of accuracy $\gamma = 0.5$ is verified. The average is taken over 500 different sample paths of the Wiener process.

In Figure 2.11, we show the increased accuracy of the Milstein scheme by comparing the numerical solution of the test linear SDE in Eq. (2.119) using a time step of $\Delta t = 2^{-7}$ and $\Delta t = 2^{-9}$ with its exact solution, shown in Eq. (2.120). Even at the higher time step, the Milstein scheme is capable of accurately approximating the exact solution, while the Euler-Maruyama scheme begins to show significant deviations. In Figure 2.12, we calculate the strong order of accuracy, $\gamma$, of the Milstein scheme using the previously described method. Again, the Milstein scheme's strong order of accuracy $\gamma = 1$ is verified.

By truncating Itô-Taylor expansions of the solution, it is possible to create even higher order methods, including the Platen method with $\gamma = 1.5$. However, these stochastic numerical methods require the evaluation of triple stochastic integrals. For multi-dimensional SDEs, these triple stochastic integrals must be numerically approximated and, consequently, the computational cost of generating these integrals typically offsets the advantage of a

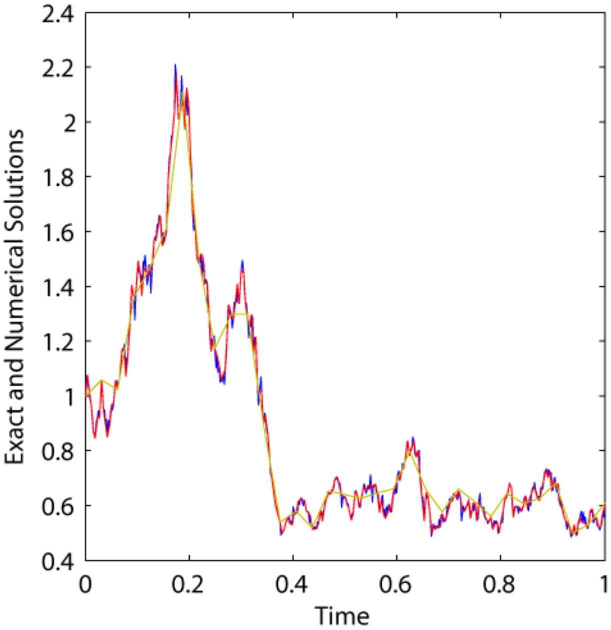

Figure 2.11: Using the Milstein scheme, the numerical solution of the simple linear stochastic differential equation in Eq. (2.119) is calculated using a time step of (red) $\Delta t = 2^{-7}$ or (yellow) $\Delta t = 2^{-5}$ and compared to the (blue) exact solution.

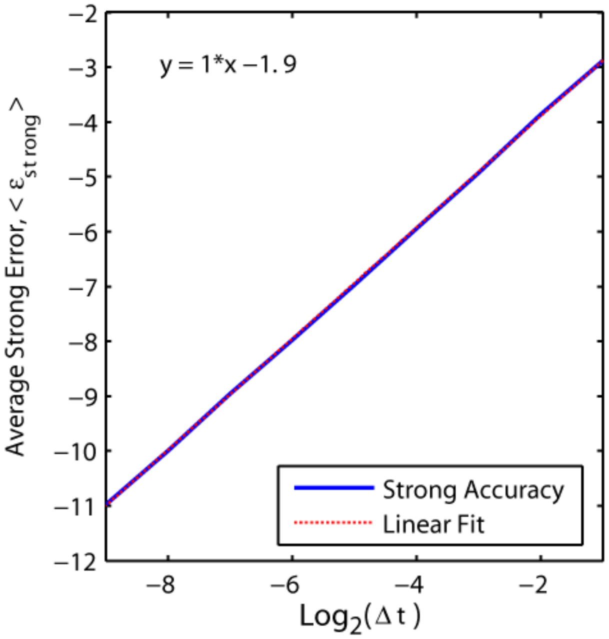

Figure 2.12: The strong order of accuracy of the Milstein scheme is determined using the same method as in Figure 2.10. The strong order of accuracy $\gamma = 1$ is verified.

larger time step.

### Stochastic Runge-Kutta Methods

A stochastic Runge-Kutta method is a derivative-free numerical scheme for solving a stochastic differential equation. Analogous to their deterministic counterparts, one may substitute a finite difference approximation for each derivative in a stochastic numerical scheme to obtain a stochastic Runge-Kutta method of the same order of accuracy. For example, by substituting the forward finite difference

$$b_{k,j}\frac{\partial b_{k,j}}{\partial x} = \frac{1}{\sqrt{\Delta t}}\left\{b_{k,j}\left(Y_k^{(n)} + a_k\Delta t + b_{k,j}\sqrt{\Delta t}\right) - b_{k,j}\left(Y_k^{(n)}\right)\right\} \tag{2.122}$$

into the Milstein scheme, the resulting stochastic numerical method

$$Y_k^{(n+1)} = Y_k^{(n)} + a_k\Delta t + \sum_{j=1}^{M} b_{k,j}\Delta W_j + \frac{1}{\sqrt{\Delta t}}\sum_{j_1,j_2=1}^{M}\left\{b_{k,j_2}\left(\Upsilon_{j_1}^{(n)}\right) - b_{k,j_2}\right\} I_{(j_1,j_2)} \tag{2.123}$$

is a Runge-Kutta method with a vector of supporting values,

$$\Upsilon_j^{(n)} = Y^{(n)} + a\,\Delta t + b_j\,\sqrt{\Delta t} \tag{2.124}$$

The $\Upsilon_k^{(n)}$ is a one step deterministic approximation of the future value of the solution. The Runge-Kutta version of the Milstein scheme has a strong and weak order of accuracy $\gamma = \beta = 1$.

### 2.4.8 Implicit Stochastic Numerical Schemes

A stochastic numerical method is implicit if the calculation of $Y^{(n+1)}$ involves the evaluation of the drift or diffusion coefficients at values of $Y^{(n+1)}$ or $t_{n+1}$. It is called semi-implicit if only the drift coefficients are evaluated at $Y^{(n+1)}$ or $t_{n+1}$. Semi-implicit methods are advantageous because they are always asymptotically stable using any time step. Consequently, for stiff stochastic differential equations, semi-implicit stochastic methods are often used.

For example, for a generalized $N$-dimensional SDE with $M$ Wiener processes, the θ-family for the semi-implicit Euler-Maruyama scheme is

$$Y_k^{(n+1)} = Y_k^{(n)} + \left\{ \theta\, a_k(Y_k^{(n+1)}, t_{n+1}) + (1-\theta)\, a_k(Y_k^{(n)}, t_n) \right\} \Delta t + \sum_{j=1}^{M} b_{k,j}\, \Delta W_j \tag{2.125}$$

for $k = 1, \ldots, N$ and where $\theta \in [0, 1]$. If $\theta = 0$ the scheme simplifies to the explicit Euler-Maruyama scheme. If $\theta = 1$ the scheme is the fully semi-implicit Euler-Maruyama scheme. If $\theta = 0.5$ the scheme is a trapezoidal-style scheme for stochastic differential equations.

Similar semi-implicit schemes exist for the Milstein and higher order methods. All of these schemes are implicit in terms of only the drift coefficients. Fully implicit schemes are implicit in terms of both the drift and diffusion terms, but often result in expressions containing terms like $\frac{1}{\Delta W_j}$. These terms are almost surely unbounded in the limit of decreasing time step and do not converge to the correct solution. However, recent research has developed a family of balanced implicit methods where the solution is implicit in terms of both the drift and diffusion coefficients while avoiding the presence of Wiener increments in the denominator [134].

### 2.4.9 Adaptive Time Step Schemes

All of the previous stochastic numerical methods have utilized a constant time step, but it is possible and frequently advantageous to vary the time step in response to changes in the stiffness of the stochastic differential equation. Modern deterministic numerical integrators commonly use a heuristic measurement to calculate an *a priori* "optimal" time step before applying the numerical scheme. For stochastic differential equations, this is often impractical because any such heuristic measurement will be a random variable and may not reflect the true error of the next iteration of the numerical scheme. Instead, it is possible to determine *a posteriori* whether the previous iteration generated too much numerical error. If it did generate too much error, it is possible to go back and reapply the numerical scheme using a reduced time step. However, when a strong solution to the SDE is desired, the reapplication of the numerical scheme must *reuse* the previously generated Wiener paths, but at intermediate time points. In order to generate an intermediate time point on the same sample paths, a Brownian bridge must be used.

Research in developing adaptive time stepping schemes for stochastic differential equations is still in its early phases. In 1997, one of the first works uses a binary Brownian bridge to develop an adaptive time stepping scheme for the Milstein scheme [135]. The same work proves, via a counter example, that an adaptive time step scheme is only gauranteed to strongly converge to the exact solution if it utilizes a stochastic numerical integrator with a strong order of accuracy $\gamma \geq 1$. More recent work has focused on developing new heuristic measurements of numerical error [136]. Finally, it is also possible to use a non-binary Brownian bridge to create an adaptive time step scheme [137], but it is more difficult to implement for a multi-dimensional stochastic differential equation.

### Brownian Bridges

Because the Wiener process is a fractal, it is possible to repeatedly "zoom" in on the sample path and always observe a continuous path with a rescaled version of the original process. A Brownian bridge is a mathematical procedure for generating sample paths with smaller and smaller time increments that are each fixed at their original end points. In this way, it is possible to determine the Wiener increments at intermediate values while reusing the same sample path.

A binary Brownian bridge is one that repeatedly halves the time increment of the sample path, generating a Brownian tree. The Brownian tree has $r = 0, 1, \ldots, R$ rows with $2^r$ elements in the $r^{th}$ row. In the beginning, the Brownian tree has a single row, called the top row. The top row of the tree contains the largest Wiener increment, $\Delta W_1^1 = W(t_f) - W(t_o)$ with an initial time step, $\Delta t = t_f - t_o$. The second row contains two smaller Wiener increments, $\Delta W_1^2 = W(t_f) - W(t_o + \frac{1}{2}\Delta t)$ and $\Delta W_2^2 = W(t_o + \frac{1}{2}\Delta t) - W(t_o)$, where the time step has been halved.

In general, there will be $2^r$ Wiener increments, each with a time step of $\frac{1}{2^r}\Delta t$ in the $r^{th}$ row. To generate the additional Wiener increments, the relationships

$$\Delta W_{2p}^{r+1} = \frac{1}{2}\Delta W_p^r + \gamma_p^{r+1} \tag{2.126}$$

and

$$\Delta W_{2p+1}^{r+1} = \frac{1}{2}\Delta W_p^r - \gamma_p^{r+1} \tag{2.127}$$

are respectively used to create the even and odd Wiener increments of the $(r+1)^{th}$ row in terms of the $r^{th}$ row with

$$\gamma_p^{r+1} \sim \mathrm{N}(0, 2^{-(r+1)}) \tag{2.128}$$

for $r = 1, \ldots, R$ rows and $p = 1, 2, \ldots, 2^r$ elements. Using the Brownian tree, one may generate the Wiener increments for a Wiener sample path for any decreasing time step so long as that time step is repeatedly halved.

### A Typical Adaptive Time Step Scheme

A typical adaptive time stepping scheme that uses a Brownian tree consists of the following steps:

1. The approximate solution, $Y^{(n+1)}$, is calculated using a stochastic numerical scheme with a strong order of accuracy, $\gamma \geq 1$ and a time step $\Delta t$.

2. Using $Y^{(n+1)}$, the measurement of the numerical error is calculated via a heuristic expression.

3. If the numerical error is greater than some tolerance, the time step is halved $\Delta t \rightarrow \Delta t/2$. If the corresponding row of the Brownian tree has not already been generated, then Eqs. (2.126) and (2.127) are used to generate it.

4. The Wiener increments from the Brownian tree are then fed into the stochastic numerical integrator with the reduced time step to produce a more accurate solution $Y^{(n+1)}$.

5. The time step selection loop repeats by going to step 2 until the generated numerical error is less than some tolerance value.

6. The procedure is repeated until a specified end time is reached: $n \rightarrow n+1$

The choice of the heuristic measurement of numerical error is arbitrary. It may be any function that measures the stiffness of the stochastic differential equation. Typically, there is a separate error measurement for the drift and diffusion terms of the SDE. Commonly, the drift or the diffusion term is, by itself, responsible for the stiffness.

# Bibliography

[1] Markham NR, Zuker M (2005) Dinamelt web server for nucleic acid melting prediction. Nucleic acids research 33:W577–81. [cited at p. v, 21, 22]

[2] Looger LL, Dwyer MA, Smith JJ, Hellinga HW (2003) Computational design of receptor and sensor proteins with novel functions. Nature 423:185–190. [cited at p. 1]

[3] Goverdhana S, Puntel M, Xiong W, Zirger JM, Barcia C, et al. (2005/8) Regulatable gene expression systems for gene therapy applications: progress and future challenges. Molecular Therapy 12:189–211. [cited at p. 1]

[4] Andrianantoandro E, Basu S, Karig DK, Weiss R (2006) Synthetic biology: new engineering rules for an emerging discipline. MolSystBiol 2:2006.0028. [cited at p. 1]

[5] Isaacs FJ, Dwyer DJ, Collins JJ (2006) Rna synthetic biology. Nature biotechnology 24:545–554. [cited at p. 1]

[6] Dhanasekaran M, Negi S, Sugiura Y (2006) Designer zinc finger proteins: tools for creating artificial dna-binding functional proteins. Accounts of Chemical Research 39:45–52. [cited at p. 1]

[7] Papworth M, Kolasinska P, Minczuk M (2006) Designer zinc-finger proteins and their applications. Gene 366:27–38. [cited at p. 1]

[8] Porteus MH (2006) Mammalian gene targeting with designed zinc finger nucleases. Molecular therapy : the journal of the American Society of Gene Therapy 13:438–446. [cited at p. 1]

[9] Durai S, Mani M, Kandavelou K, Wu J, Porteus MH, et al. (2005) Zinc finger nucleases: custom-designed molecular scissors for genome engineering of plant and mammalian cells. Nucleic acids research 33:5978–5990. [cited at p. 1]

[10] Kandavelou K, Mani M, Durai S, Chandrasegaran S (2005) ”magic” scissors for genome surgery. Nature biotechnology 23:686–687. [cited at p. 1]

[11] Klug A (2005) Towards therapeutic applications of engineered zinc finger proteins. FEBS letters 579:892–894. [cited at p. 1]

[12] Pabo CO, Peisach E, Grant RA (2001) Design and selection of novel cys2his2 zinc finger proteins. Annual Review of Biochemistry 70:313–340. [cited at p. 1]

[13] Joung JK, Ramm EI, Pabo CO (2000) A bacterial two-hybrid selection system for studying protein-dna and protein-protein interactions. Proceedings of the National Academy of Sciences of the United States of America 97:7382–7387. [cited at p. 1]

[14] Gregory BD, Deighan P, Hochschild A (2005) An artificial activator that contacts a normally occluded surface of the rna polymerase holoenzyme. Journal of Molecular Biology 353:497–506. [cited at p. 1]

[15] Gregory BD, Nickels BE, Garrity SJ, Severinova E, Minakhin L, et al. (2004) A regulator that inhibits transcription by targeting an intersubunit interaction of the rna polymerase holoenzyme. Proceedings of the National Academy of Sciences of the United States of America 101:4554–4559. [cited at p. 1]

[16] Alper H, Fischer C, Nevoigt E, Stephanopoulos G (2005) Tuning genetic control through promoter engineering. Proceedings of the National Academy of Sciences of the United States of America 102:12678–12683. [cited at p. 1]

[17] Navani NK, Li Y (2006) Nucleic acid aptamers and enzymes as sensors. Current opinion in chemical biology 10:272–281. [cited at p. 1, 8]

[18] Davidson EA, Ellington AD (2005) Engineering regulatory rnas. Trends in biotechnology 23:109–112. [cited at p. 1, 8]

[19] Gardner TS, Cantor CR, Collins JJ (2000) Construction of a genetic toggle switch in escherichia coli. Nature 403:339–342. [cited at p. 2]

[20] Isaacs FJ, Dwyer DJ, Ding C, Pervouchine DD, Cantor CR, et al. (2004) Engineered riboregulators enable post-transcriptional control of gene expression. Nature biotechnology 22:841–847. [cited at p. 2]

[21] Elowitz MB, Leibler S (2000) A synthetic oscillatory network of transcriptional regulators. Nature 403:335–338. [cited at p. 2]

[22] Fung E, Wong WW, Suen JK, Bulter T, Lee SG, et al. (2005) A synthetic gene-metabolic oscillator. Nature; Nature 435:118–122. [cited at p. 2]

[23] Atkinson MR, Savageau MA, Myers JT, Ninfa AJ (2003) Development of genetic circuitry exhibiting toggle switch or oscillatory behavior in escherichia coli. Cell 113:597–607. [cited at p. 2]

[24] You L, 3rd RSC, Weiss R, Arnold FH (2004) Programmed population control by cell-cell communication and regulated killing. Nature; Nature 428:868–871. [cited at p. 2]

[25] Becskei A, Serrano L (2000) Engineering stability in gene networks by autoregulation. Nature 405:590–593. [cited at p. 2]

[26] Maeda YT, Sano M (2006) Regulatory dynamics of synthetic gene networks with positive feedback. Journal of Molecular Biology 359:1107–1124. LR: 20061115; PUBM: Print-Electronic; DEP: 20060427; JID: 2985088R; 0 (Recombinant Fusion Proteins); 2006/03/29 [received]; 2006/03/29 [accepted]; 2006/04/27 [aheadofprint]; ppublish. [cited at p. 2]

[27] Basu S, Gerchman Y, Collins CH, Arnold FH, Weiss R (2005) A synthetic multicellular system for programmed pattern formation. Nature 434:1130–1134. [cited at p. 2]

[28] Chen MT, Weiss R (2005) Artificial cell-cell communication in yeast saccharomyces cerevisiae using signaling elements from arabidopsis

thaliana. Nature biotechnology; Nature biotechnology 23:1551–1555. [cited at p. 2]

[29] Bulter T, Lee SG, Wong WW, Fung E, Connor MR, et al. (2004) Design of artificial cell-cell communication using gene and metabolic networks. Proceedings of the National Academy of Sciences of the United States of America; Proceedings of the National Academy of Sciences of the United States of America 101:2299–2304. [cited at p. 2]

[30] Levskaya A, Chevalier AA, Tabor JJ, Simpson ZB, Lavery LA, et al. (2005) Synthetic biology: engineering escherichia coli to see light. Nature 438:441–442. [cited at p. 2]

[31] Murray JD (2004) Mathematical Biology. New York: Springer. [cited at p. 2]

[32] Buchler NE, Gerland U, Hwa T (2003) On schemes of combinatorial transcription logic. Proceedings of the National Academy of Sciences of the United States of America 100:5136–5141. [cited at p. 2]

[33] Bintu L, Buchler NE, Garcia HG, Gerland U, Hwa T, et al. (2005) Transcriptional regulation by the numbers: applications. Current Opinion in Genetics and Development 15:125–135. [cited at p. 2]

[34] Vilar JM, Saiz L (2005) Dna looping in gene regulation: from the assembly of macromolecular complexes to the control of transcriptional noise. Current Opinion in Genetics and Development 15:136–144. [cited at p. 2, 19]

[35] Francois P, Hakim V (2005) Core genetic module: the mixed feedback loop. PhysRevEStatNonlin Soft Matter Phys 72:031908. [cited at p. 2]

[36] Mangan S, Alon U (2003) Structure and function of the feed-forward loop network motif. Proceedings of the National Academy of Sciences of the United States of America 100:11980–11985. [cited at p. 2]

[37] Weinberger LS, Burnett JC, Toettcher JE, Arkin AP, Schaffer DV (2005) Stochastic gene expression in a lentiviral positive-feedback loop: Hiv-1 tat fluctuations drive phenotypic diversity. Cell 122:169–182. [cited at p. 2]

[38] Weinberger LS, Schaffer DV, Arkin AP (2003) Theoretical design of a gene therapy to prevent aids but not human immunodeficiency virus type 1 infection. Journal of virology 77:10028–10036. [cited at p. 2]

[39] Acar M, Becskei A, van Oudenaarden A (2005) Enhancement of cellular memory by reducing stochastic transitions. Nature 435:228–232. [cited at p. 3]

[40] Li H, Hou Z, Xin H (2005) Internal noise stochastic resonance for intracellular calcium oscillations in a cell system. PhysRevEStatNonlin Soft Matter Phys 71:061916. [cited at p. 3]

[41] Patel A, Kosko B (2005) Stochastic resonance in noisy spiking retinal and sensory neuron models. Neural networks : the official journal of the International Neural Network Society 18:467–478. [cited at p. 3]

[42] Samoilov M, Plyasunov S, Arkin AP (2005) Stochastic amplification and signaling in enzymatic futile cycles through noise-induced bistability with oscillations. Proceedings of the National Academy of Sciences of the United States of America 102:2310–2315. [cited at p. 3]

[43] Cox CD, Peterson GD, Allen MS, Lancaster JM, McCollum JM, et al. (2003) Analysis of noise in quorum sensing. OMICS A Journal of Integrative Biology 7:317–334. [cited at p. 3]

[44] Francois P, Hakim V (2004) Design of genetic networks with specified functions by evolution in silico. Proceedings of the National Academy of Sciences of the United States of America 101:580–585. [cited at p. 3]

[45] Hermsen R, Tans S, ten Wolde PR (2006) Transcriptional regulation by competing transcription factor modules. PLoS Computational Biology 2:e164 OP. [cited at p. 3]

[46] Tomshine J, Kaznessis YN (2006) Optimization of a stochastically simulated gene network model via simulated annealing. Biophysical journal 91:3196–3205. [cited at p. 3]

[47] Elowitz MB, Levine AJ, Siggia ED, Swain PS (2002) Stochastic gene expression in a single cell. Science 297:1183–1186. [cited at p. 3]

[48] Salis H, Kaznessis YN (2006) Computer-aided design of modular protein devices: Boolean and gene activation. Physical Biology 3:295–310. [cited at p. 4]

[49] Tuttle LM, Salis H, Tomshine J, Kaznessis YN (2005) Model-driven designs of an oscillating gene network. Biophysical journal 89:3873–3883. [cited at p. 4]

[50] CRICK FH (1958) On protein synthesis. Symposia of the Society for Experimental Biology 12:138–163. [cited at p. 5]

[51] Dai X, Rothman-Denes LB (1999) Dna structure and transcription. Current opinion in microbiology 2:126–130. [cited at p. 5]

[52] Browning DF, Busby SJ (2004) The regulation of bacterial transcription initiation. Nature reviewsMicrobiology 2:57–65. [cited at p. 6, 19]

[53] Burgess RR, Anthony L (2001) How sigma docks to rna polymerase and what sigma does. Current opinion in microbiology 4:126–131. [cited at p. 6]

[54] Borukhov S, Nudler E (2003) Rna polymerase holoenzyme: structure, function, and biological implications. Current opinion in microbiology 6:93–100. [cited at p. 6]

[55] Ptashne M, Gann A (2002) Genes and Signals. Cold Spring Harbor, N.Y: Cold Spring Harbor Laboratory Press, 208 pp. [cited at p. 6, 12]

[56] Henkin TM (2000) Transcription termination control in bacteria. Current opinion in microbiology 3:149–153. [cited at p. 7, 8]

[57] Alberts B, Johnson A, Lewis J, Raff M, Roberts K, et al. (2002) Molecular Biology of the Cell. New York: Garland Science. [cited at p. 8, 10, 11]

[58] Franch T, Gerdes K (2000) U-turns and regulatory rnas. Current opinion in microbiology 3:159–164. [cited at p. 8, 23]

[59] Masse E, Majdalani N, Gottesman S (2003) Regulatory roles for small rnas in bacteria. Current opinion in microbiology 6:120–124. [cited at p. 8, 22]

[60] Thodima V, Pirooznia M, Deng Y (2006) Riboaptdb: A comprehensive database of ribozymes and aptamers. BMC bioinformatics [computer file] 7 Suppl 2:S6. [cited at p. 8]

[61] Ramakrishnan V (2002) Ribosome structure and the mechanism of translation. Cell 108:557–572. [cited at p. 9]

[62] Shine J, Dalgarno L (1975) Determinant of cistron specificity in bacterial ribosomes. Nature 254:34–38. [cited at p. 9, 21]

[63] Boni IV, Isaeva DM, Musychenko ML, Tzareva NV (1991) Ribosome-messenger recognition: mrna target sites for ribosomal protein s1. Nucleic acids research 19:155–162. [cited at p. 9, 21]

[64] Sorensen MA, Fricke J, Pedersen S (1998) Ribosomal protein s1 is required for translation of most, if not all, natural mrnas in escherichia coli in vivo. Journal of Molecular Biology 280:561–569. [cited at p. 9]

[65] Schuwirth BS, Borovinskaya MA, Hau CW, Zhang W, Vila-Sanjurjo A, et al. (2005) Structures of the bacterial ribosome at 3.5 a resolution. Science 310:827–834. [cited at p. 10]

[66] Hansen TM, Baranov PV, Ivanov IP, Gesteland RF, Atkins JF (2003) Maintenance of the correct open reading frame by the ribosome. EMBO reports 4:499–504. [cited at p. 11, 24]

[67] McKnight SL, Yamamoto KR (1992) Transcriptional Regulation. Plainview: Cold Spring Harbor Laboratory Press. [cited at p. 12]

[68] Macnab RM (1992) Genetics and biogenesis of bacterial flagella. Annual Review of Genetics 26:131–158. [cited at p. 13]

[69] Weber H, Polen T, Heuveling J, Wendisch VF, Hengge R (2005) Genome-wide analysis of the general stress response network in escherichia coli: sigmas-dependent genes, promoters, and sigma factor selectivity. The Journal of Bacteriology 187:1591–1603. [cited at p. 13, 15]

[70] Nonaka G, Blankschien M, Herman C, Gross CA, Rhodius VA (2006) Regulon and promoter analysis of the e. coli heat-shock factor,

sigma32, reveals a multifaceted cellular response to heat stress. Genes and Development 20:1776–1789. [cited at p. 13, 16]

[71] Gruber TM, Gross CA (2003) Multiple sigma subunits and the partitioning of bacterial transcription space. Annual Review of Microbiology 57:441–466. [cited at p. 14, 15]

[72] Vo NV, Hsu LM, Kane CM, Chamberlin MJ (2003) In vitro studies of transcript initiation by escherichia coli rna polymerase. 3. influences of individual dna elements within the promoter recognition region on abortive initiation and promoter escape. Biochemistry 42:3798–3811. [cited at p. 15]

[73] Nystrom T (2004) Stationary-phase physiology. Annual Review of Microbiology 58:161–181. [cited at p. 16]

[74] Ross W, Gosink KK, Salomon J, Igarashi K, Zou C, et al. (1993) A third recognition element in bacterial promoters: Dna binding by the alpha subunit of rna polymerase. Science 262:1407–1413. [cited at p. 16]

[75] Pemberton IK, Muskhelishvili G, Travers AA, Buckle M (2000) The g+c-rich discriminator region of the tyrt promoter antagonises the formation of stable preinitiation complexes. Journal of Molecular Biology 299:859–864. [cited at p. 16]

[76] Pabo CO, Sauer RT (1992) Transcription factors: structural families and principles of dna recognition. Annual Review of Biochemistry 61:1053–1095. [cited at p. 17]

[77] Jackson DA (2003) The anatomy of transcription sites. Current opinion in cell biology 15:311–317. [cited at p. 17]

[78] Hsieh W, Whitson P, Matthews K, Wells R (1987) Influence of sequence and distance between two operators on interaction with the lac repressor. Journal of Biological Chemistry 262:14583–14591. [cited at p. 17, 18]

[79] Horton N, Lewis M, Lu P (1997/1/10) Escherichia coli lac repressor-lac operator interaction and the influence of allosteric effectors. Journal of Molecular Biology 265:1–7. [cited at p. 17]

[80] von Hippel PH, Revzin A, Gross CA, Wang AC (1974) Non-specific dna binding of genome regulating proteins as a biological control mechanism: I. the lac operon: equilibrium aspects. Proceedings of the National Academy of Sciences of the United States of America 71:4808–4812. [cited at p. 17]

[81] Lederer T, Takahashi M, Hillen W (1995/12/10) Thermodynamic analysis of tetracycline-mediated induction of tet repressor by a quantitative methylation protection assay. Analytical Biochemistry 232:190–196. [cited at p. 17, 18]

[82] Timmes A, Rodgers M, Schleif R (2004) Biochemical and physiological properties of the dna binding domain of arac protein. Journal of Molecular Biology 340:731–738. [cited at p. 17]

[83] Ptashne M, Gann A (1997) Transcriptional activation by recruitment. Nature 386:569–577. [cited at p. 18]

[84] Rojo F (2001) Mechanisms of transcriptional repression. Current opinion in microbiology 4:145–151. [cited at p. 18]

[85] Meng X, Brodsky MH, Wolfe SA (2005) A bacterial one-hybrid system for determining the dna-binding specificity of transcription factors. Nature biotechnology 23:988–994. [cited at p. 19]

[86] Muller J, Oehler S, Muller-Hill B (1996) Repression of lac promoter as a function of distance, phase, and quality of an auxiliary lac operator. Journal of Molecular Biology 257:21–29. [cited at p. 20]

[87] Coleman BD, Olson WK, Swigon D (2003) Theory of sequence-dependent dna elasticity. Journal of Chemical Physics 118:7127–7140. [cited at p. 20]

[88] Niki H, Yamaichi Y, Hiraga S (2000) Dynamic organization of chromosomal dna in escherichia coli. Genes and Development 14:212–223. [cited at p. 20]

[89] Reeder J, Hochsmann M, Rehmsmeier M, Voss B, Giegerich R (2006) Beyond mfold: recent advances in rna bioinformatics. Journal of Biotechnology 124:41–55. [cited at p. 21]

[90] Jin H, Zhao Q, de Valdivia EIG, Ardell DH, Stenstrom M, et al. (2006) Influences on gene expression in vivo by a shine-dalgarno sequence. Molecular microbiology 60:480–492. [cited at p. 21]

[91] de Smit M, van Duin J (1990) Secondary structure of the ribosome binding site determines translational efficiency: A quantitative analysis. Proceedings of the National Academy of Sciences 87:7668–7672. [cited at p. 22]

[92] Ellis JJ, Broom M, Jones S (2006) Protein-rna interactions: Structural analysis and functional classes. Proteins . [cited at p. 22]

[93] Fuglsang A (2003) Codon optimizer: a freeware tool for codon optimization. Protein expression and purification 31:247–249. [cited at p. 24]

[94] Rauhut R, Klug G (1999) mrna degradation in bacteria. FEMS microbiology reviews 23:353–370. [cited at p. 25]

[95] Regnier P, Arraiano CM (2000) Degradation of mrna in bacteria: emergence of ubiquitous features. BioEssays : news and reviews in molecular, cellular and developmental biology 22:235–244. [cited at p. 25]

[96] Pfleger BF, Pitera DJ, Smolke CD, Keasling JD (2006) Combinatorial engineering of intergenic regions in operons tunes expression of multiple genes. Nature biotechnology 24:1027–1032. [cited at p. 26]

[97] Pertzev AV, Nicholson AW (2006) Characterization of rna sequence determinants and antideterminants of processing reactivity for a minimal substrate of escherichia coli ribonuclease iii. Nucleic acids research 34:3708–3721. [cited at p. 26]

[98] Storz G, Opdyke JA, Zhang A (2004) Controlling mrna stability and translation with small, noncoding rnas. Current opinion in microbiology 7:140–144. [cited at p. 26]

[99] Laursen BS, Sorensen HP, Mortensen KK, Sperling-Petersen HU (2004) Initiation of protein synthesis in bacteria. Microbiology and Molecular Biology Reviews 69:101–123. [cited at p. 26]

[100] Jenal U, Hengge-Aronis R (2003) Regulation by proteolysis in bacterial cells. Current opinion in microbiology 6:163–172. [cited at p. 27]

[101] Gottesman S (2003) Proteolysis in bacterial regulatory circuits. Annual Review of Cell and Developmental Biology 19:565–587. [cited at p. 27]

[102] Karzai AW, Roche ED, Sauer RT (2000) The ssra smpb system for protein tagging, directed degradation and ribosome rescue. Nature structural biology 7:449–455. [cited at p. 27]

[103] Nalefski EA, Nebelitsky E, Lloyd JA, Gullans SR (2006) Single-molecule detection of transcription factor binding to dna in real time: specificity, equilibrium, and kinetic parameters. Biochemistry 45:13794–13806. [cited at p. 30]

[104] McDonnell JM (2001) Surface plasmon resonance: towards an understanding of the mechanisms of biological molecular recognition. Current opinion in chemical biology 5:572–577. [cited at p. 30]

[105] Halford SE, Marko JF (2004) How do site-specific dna-binding proteins find their targets? Nucleic acids research 32:3040–3052. [cited at p. 30]

[106] Gibson MA, Bruck J (2000) Efficient exact stochastic simulation of chemical systems with many species and many channels. Journal of Physical Chemistry A 104:1876–1889. [cited at p. 34, 74, 77]

[107] Vogel U, Jensen KF (1994) The rna chain elongation rate in escherchia coli depends on growth rate. Journal of Bacteriology 176:2807–2813. [cited at p. 34]

[108] Hill TL (2002) Thermodynamics of Small Systems. Courier Dover Publications. [cited at p. 36]

[109] Goldbeter A, Jr DEK (1981) An amplified sensitivity arising from covalent modification in biological systems. Proceedings of the National Academy of Sciences of the United States of America 78:6840–6844. [cited at p. 43]

[110] Gillespie DT (1992) Markov processes: an introduction for physical scientists. Boston: Academic Press. [cited at p. 47]

[111] van Kampen NG (1992) Stochastic processes in physics and chemistry. Amsterdam: North-Holland. [cited at p. 47]

[112] Mircea G (2002) Stochastic Calculus: Applications in Science and Engineering. Boston: Birkhauser. [cited at p. 47]

[113] Kloeden PE, Platen E (1992) Numerical solution of stochastic differential equations, volume 23; 23. Berlin: Springer-Verlag, 632 pp. [cited at p. 47, 55, 95]

[114] PARK SK, MILLER KW (1988) Random number geuerators: Good ones are hard to find. Communications of the ACM 2. [cited at p. 54]

[115] Marsaglia G (1985) A current view of random number generators. Computer Science and Statistics, Sixteenth Symposium on the Interface :310. [cited at p. 54]

[116] Devroye L (1986) Non-uniform random variate generation. New York: Springer. [cited at p. 55]

[117] Marsaglia G, Tsang WW (2000) The ziggurat method for generating random variables. Journal of Statistical Software 5:17. [cited at p. 55]

[118] Metropolis N, Rosenbluth A, Rosenbluth M, Teller A, Teller E (1953) Equation of state calculations by fast computation machines. Journal of Chemical Physics 21:10871092. [cited at p. 55]

[119] Gillespie DT (2001) Approximate accelerated stochastic simulation of chemically reacting systems. Journal of Chemical Physics 115:1716–1733. [cited at p. 57, 79]

[120] Einstein A (1905) The theory of the brownian movement. Annder Physik 17:549. [cited at p. 67]

[121] Bortz AB, Lebowitz JL, Kalos MH (1975) A new algorithm for monte carlo simulation of ising spin systems. Journal of Computational Physics 17:10–18. [cited at p. 74]

[122] Gillespie DT (1976) A general method for numerically simulating the stochastic time evolution of coupled chemical reactions. Journal of Computational Physics 22:403–434. [cited at p. 74]

[123] Gillespie DT (1977) Exact stochastic simulation of coupled chemical reactions. Journal of Physical Chemistry 81:2340–2381. [cited at p. 74]

[124] Gillespie DT, Petzold LR (2003) Improved leap-size selection for accelerated stochastic simulation. Journal of Chemical Physics 119:8229–8234. [cited at p. 80]

[125] Rathinam M, Petzold LR, Cao Y, Gillespie DT (2003) Stiffness in stochastic chemically reacting systems: The implicit tau-leaping method. Journal of Chemical Physics 119:12784–12794. [cited at p. 80]

[126] Cao Y, Petzold LR, Rathinam M, Gillespie DT (2004) The numerical stability of leaping methods for stochastic simulation of chemically reacting systems. The Journal of chemical physics 121:12169–12178. [cited at p. 80]

[127] Cao Y, Gillespie DT, Petzold LR (2006) Efficient step size selection for the tau-leaping simulation method. The Journal of chemical physics 124:044109. [cited at p. 80]

[128] Cao Y, Gillespie DT, Petzold LR (2005) Avoiding negative populations in explicit poisson tau-leaping. The Journal of chemical physics 123:054104. [cited at p. 80]

[129] Hausenblas E (2002) Error analysis for approximation of stochastic differential equations driven by poisson random measures. SIAM Journal of Numerical Analysis 40:87–113. [cited at p. 81]

[130] Chatterjee A, Vlachos DG, Katsoulakis MA (2005) Binomial distribution based tau-leap accelerated stochastic simulation. The Journal of chemical physics 122:024112. [cited at p. 81]

[131] Tian T, Burrage K (2004) Binomial leap methods for simulating stochastic chemical kinetics. The Journal of chemical physics 121:10356–10364. [cited at p. 81]

[132] Salis H, Kaznessis Y (2005) Accurate hybrid stochastic simulation of a system of coupled chemical or biochemical reactions. The Journal of Chemical Physics 122:054103. [cited at p. 82]

[133] Milstein GN (1988) The numerical integration of stochastic differential equations (Russian). Sverdlovsk: Urals University Press, 225 pp. [cited at p. 100]

[134] Milstein GN, Platen E, Schurz H (1998) Balanced implicit methods for stiff stochastic systems. SIAM Journal of Numerical Analysis 35:1010–1019. [cited at p. 104]

[135] Gaines JG, Lyons TJ (1997) Variable step size control in the numerical solution to stochastic differential equations. SIAM Journal on Applied Mathematics 57:1455–1484. [cited at p. 105]

[136] Lamba H (2003) An adaptive timestepping algorithm for stochastic differential equations. Journal of Computational and Applied Mathematics 161:417–430. [cited at p. 105]

[137] Burrage PM, Herdiana R, Burrage K (2004) Adaptive stepsize based on control theory for stochastic differential equations. Journal of Computational and Applied Mathematics 171:317–336. [cited at p. 105]