Parameter Inference in Stochastic Biological Systems via Differentiable Programming

Liam Z. Li, supervised by Dr Tom Kimpson and Prof Jennifer Flegg

2025/26 Maths and Stats Vacation Scholarships Program, The University of Melbourne

Introduction

Biological systems such as gene regulatory networks are inherently stochastic, with discrete reaction events driven by low molecule counts. Parameter inference is challenging because standard automatic differentiation[1] cannot differentiate through these discontinuous random jumps.

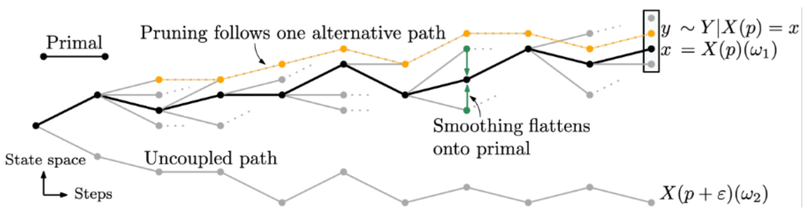

Figure 1. Illustration of gradient estimation failure in stochastic programs. Independent stochastic executions lead to divergent computation paths, making finite-difference gradients unstable. Reproduced from Arya et al. (2022) [2].

Recent work in differentiable programming addresses this gap. Arya et al. (2022) [2] introduces *stochastic automatic differentiation* (StochasticAD), which enables gradients to be computed directly through programs with discrete randomness. We apply these ideas to infer parameters in a stochastic gene expression model and evaluate its performance.

Background

Gradient Estimation in Stochastic Simulations

A common method to estimate gradients is finite-difference (FD) estimation: $\frac{\partial L}{\partial k} \approx \frac{L(k+\varepsilon)-L(k)}{\varepsilon}$, but FD gradients suffer from a bias–variance trade-off: small ε yields low bias but high variance, while large ε reduce variance but lead to high bias.

In practice, reducing this variance typically requires averaging finite-difference gradients over many Monte Carlo samples, substantially increasing the computational cost of optimisation.

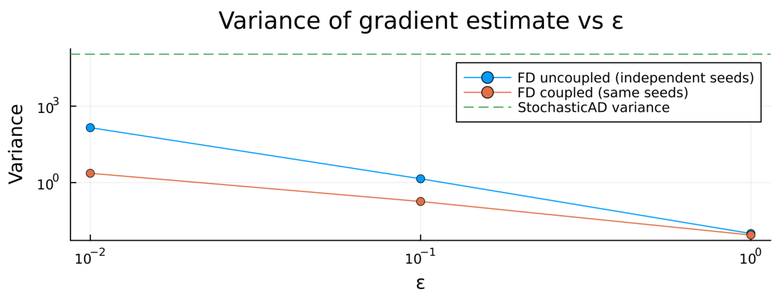

Figure 2. Variance of gradient estimates versus finite-difference step size ε. Both uncoupled and coupled finite differences exhibit rapidly increasing variance as $\varepsilon \to 0$, while StochasticAD maintains bounded variance independent of step size.

These limitations motivate the use of StochasticAD, which avoids finite-difference tuning by propagating gradients directly through stochastic simulations.

The Model

We model **Constitutive Gene Expression**, a fundamental birth-death process representing protein production and degradation. The system consists of two elementary reactions:

1. **Translation (Birth):** $\emptyset \xrightarrow{k} P$ (Production of protein P at rate k)
2. **Degradation (Death):** $P \xrightarrow{\gamma} \emptyset$ (Decay of protein P at rate γ)

At steady state, $P \sim \text{Poisson}\left(\frac{k}{\gamma}\right)$ with mean $\mu = \frac{k}{\gamma}$.

This analytic solution provides a "ground truth" to validate our computational inference pipeline.

Methodology

To make the system amenable to Automatic Differentiation (AD) while preserving stochasticity, we used a Discrete-Time τ-Leaping Approximation Algorithm.[3]

The Simulation Algorithm

Time is discretised into fixed steps of Δt. The state X_t (integer protein count) evolves as follows:

1. **Births (B_t)** The number of production events in Δt is modeled as a Poisson random variable

$$B_t \sim \text{Poisson}(k \cdot \Delta t)$$

2. **Deaths (D_t)** The number of decay events is modeled as a Binomial random variable

$$D_t \sim \text{Binomial}(X_t, 1 - e^{-\gamma \Delta t})$$

3. **Update Rule:**

$$X_{t+1} = X_t + B_t - D_t$$

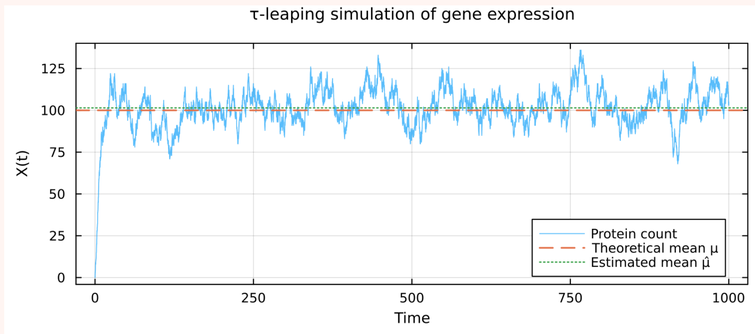

Figure 3. Discrete-time τ-leaping simulation of gene expression. Stochastic birth and death events generate noisy protein trajectories that fluctuate around a steady-state mean set by k/γ.

The Inference Problem

We assume that the observed data provide a target steady-state mean protein count μ_{obs}. We wish to infer the model parameters $\theta = (k, \gamma)$ such that simulations of the stochastic model reproduce this observed mean.

Let $\hat{\mu}(\theta)$ denote the empirical mean protein count calculated from the simulation trajectory after discarding a burn-in period T_{burn}. We formulate parameter inference as the minimisation of the loss function:

$$L(\theta) = (\hat{\mu}(\theta) - \mu_{\text{obs}})^2$$

We then use StochasticAD to compute unbiased gradients $\nabla_\theta L$ through the simulator, and we optimise $\theta = (k, \gamma)$ using the following update rule:

$$\theta_{i+1} \leftarrow \theta_i - \eta \nabla_\theta L(\theta_i)$$

where i denotes the iteration index and η is the learning rate.

Results

The Identifiability Problem

Across random initialisations, our optimisation converges to different parameter pairs (k, γ) that all reproduce the observed steady-state mean.

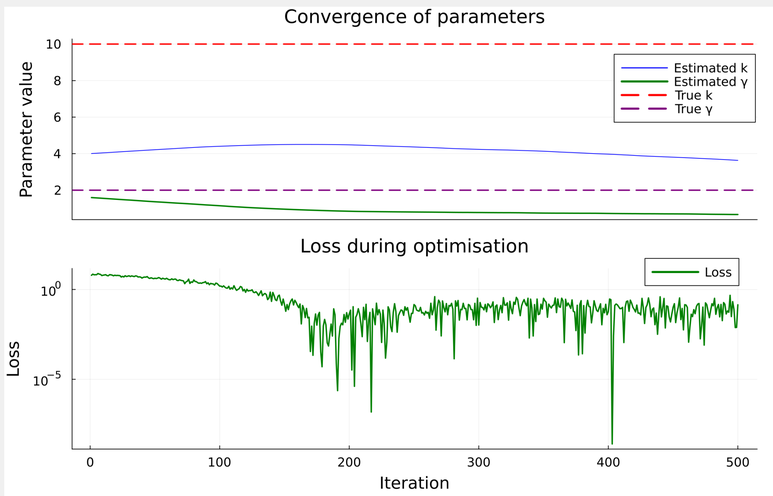

Figure 4. Top: parameter convergence trajectories during optimisation. Bottom: corresponding loss evolution. Optimisation converges to different (k, γ) pairs with identical steady-state mean $\mu = k/\gamma$.

This reflects a fundamental **parameter identifiability issue**: since the steady-state mean depends only on the ratio $\mu = k/\gamma$, there are infinitely many parameter pairs (k, γ) that yield the same observed mean. As a result, different random initialisations may lead to different (k, γ) pairs satisfying $k/\gamma = \mu_{obs}$, but not necessarily the true parameters, even though optimisation successfully minimises the loss.

Solution

In many biological systems, the degradation rate γ can be measured independently (e.g. via protein half-life experiments) or obtained from prior experimental studies. Therefore, to address this identifiability issue, we fix γ as its true value and perform univariate optimisation over k only.

Results

To evaluate the proposed approach, we first generate synthetic data from the constitutive gene expression model with known ground-truth parameters (k^*, γ^*). Then by fixing $\gamma = \gamma^*$, we inferred k by matching simulated and observed steady-state means using **stochastic gradient descent** and the **Adam optimiser** [4], both driven by gradients obtained via stochasticAD.

Across random initialisations, the optimiser consistently reduced the loss and converged to values of k close to the true parameter.

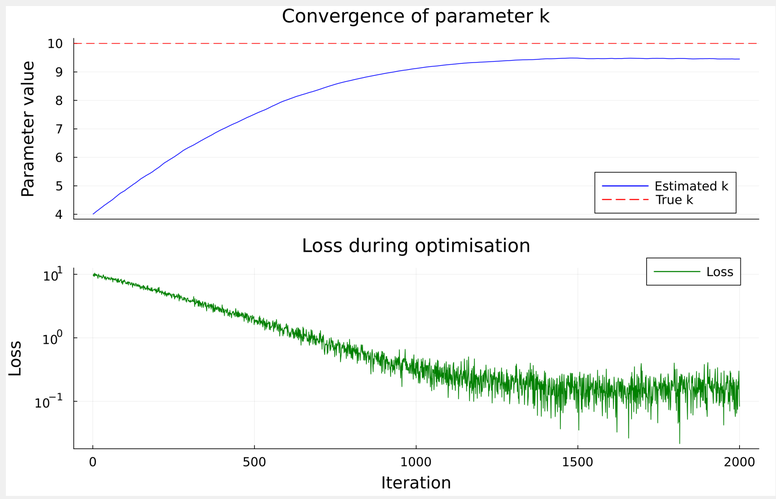

Figure 5. Top: parameter convergence trajectories during optimisation. Bottom: corresponding loss evolution. The value of k converges close to the true value of k

The optimisation exhibits smooth and stable parameter trajectories despite intrinsic stochasticity in the simulator. The loss decreases steadily before plateauing, indicating that the inferred parameter reproduces the observed steady-state behaviour.

In contrast, joint optimisation of (k, γ) (Figure 4) converges to different parameter pairs and exhibits sharp "spikes" and pronounced volatility in the loss trajectory. By fixing γ, this instability is removed, resulting in faster and more consistent convergence across random initialisations.

Overall, we show that StochasticAD enables stable, gradient-based optimisation in discrete stochastic simulators, provided the inference problem itself is identifiable.

Conclusion and Future Directions

This work illustrates how differentiable programming can bridge stochastic simulation and gradient-based inference in biological systems with discrete randomness. By embedding a stochastic model directly within an optimisation loop, we demonstrate a practical workflow for treating simulators themselves as differentiable objects, a paradigm central to modern scientific machine learning.

Future directions may include:

- **Use temporal dynamics:** Fit full time-series data rather than steady-state means to distinguish (k, γ) via relaxation timescales.
- **Incorporate variance or autocorrelation:** Higher-order statistics or temporal correlations provide additional constraints.
- **Regularisation and priors:** Include prior knowledge about biologically plausible parameter ranges
- **Multi-observable data:** If additional measurements (e.g., mRNA levels, protein degradation rates) are available, the system becomes identifiable.

References

[1] A. G. Baydin, B. A. Pearlmutter, A. A. Radul, and J. M. Siskind, "Automatic differentiation in machine learning: A survey," *Journal of Machine Learning Research*, vol. 18, no. 153, pp. 1–43, 2018.

[2] G. Arya, M. Schauer, F. Schafer, and C. Rackauckas, "Automatic differentiation of programs with discrete randomness," *Advances in Neural Information Processing Systems*, vol. 35, 2022.

[3] D. T. Gillespie, "Approximate accelerated stochastic simulation of chemically reacting systems," *The Journal of Chemical Physics*, vol. 115, no. 4, pp. 1716–1733, 2001. 10.1063/1.1378322.

[4] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *International Conference on Learning Representations*, 2015.

Contact: liamzefu.li@student.unimelb.edu.au