Proceedings of the 2022 Winter Simulation Conference
B. Feng, G. Pedrielli, Y. Peng, S. Shashaani, E. Song, C.G. Corlu, L.H. Lee, E.P. Chew, T. Roeder, and P. Lendermann, eds.

COMBINING NUMERICAL LINEAR ALGEBRA WITH SIMULATION TO COMPUTE STATIONARY DISTRIBUTIONS

Zeyu Zheng

Department of Industrial Engineering
and Operations Research
University of California Berkeley
4125 Etcheverry Hall
Berkeley, CA 94720, USA

Alex Infanger

Institute for Computational and
Mathematical Engineering
Stanford University
475 Via Ortega
Stanford, CA 94305, USA

Peter W. Glynn

Department of Management Science and Engineering
Stanford University
475 Via Ortega
Stanford, CA 94305, USA

ABSTRACT

This paper introduces the first fully integrated algorithm for combining simulation with numerical linear algebra, as a means of computing stationary distributions for Markov chains and Markov jump processes. We use linear algebra to analyze the "center" of the state space, while simulation is used to estimate contributions to the steady-state from path excursions outside the "center". The method yields consistent estimators for stationary expectations, and can be viewed as an application of the variance reduction technique known as conditional Monte Carlo.

1 INTRODUCTION

This paper is concerned with the computation of stationary distributions for Markov chains and Markov jump processes taking values in a discrete state space S. Such stationary distributions are of central interest when the long-run steady-state behavior of the associated system is relevant. When the state space is of modest size, it is well known that the stationary distribution can be numerically computed as the solution to a system of linear equations; see, for example, (Heyman and Sobel 2004) and (Asmussen 2008). However, when the state space is large or countably infinite, the use of simulation is often computationally preferable.

This paper is concerned with combining the best features of numerical linear algebra (e.g. solving systems of linear equations) with simulation. In particular, numerical linear algebra can compute high accuracy solutions to linear systems of moderate size, while simulation typically computes low accuracy solutions to systems of almost arbitrarily large size. In this paper, we propose the first algorithm that computes solutions to systems of linear equations for the "center" of the state space supporting most of the equilibrium distribution, and uses simulation to explore the remaining outer state space that is of much larger magnitude. There are no ad hoc elements to the way in which we combine simulation with numerical linear algebra. Rather, we adopt a principled approach in which the combined algorithm is always guaranteed to produce an estimator that converges to the stationary quantity, regardless of how

many states the numerical linear algebraic component of the algorithm encompasses. We call our approach COSIMLA, for "COmbined SIMulation and Linear Algebra."

As we shall see in Section 2, COSIMLA can be viewed as being a powerful variance reduction technique (specifically, an instance of conditional Monte Carlo). Alternatively, one can view COSIMLA as a means of using simulation to compute those elements that can not be calculated from within the finite "truncated" sub-matrix used by the linear algebraic computation. (Kuntz, Thomas, Stan, and Barahona 2021) provide a comprehensive and accessible review of truncation-based approximation schemes for Markov jump processes with infinite or large state space. This paper paper is also connected to the literature in hybrid simulation, in that it combines simulation with the linear algebraic methods used to study one of the most import systems dynamic models, namely Markov chains; see (Brailsford, Eldabi, Kunc, Mustafee, and Osorio 2019) for a survey on hybrid simulation.

The rest of this paper is organized as follows. Section 2 describes and analyzes the COSIMLA approach for computing Markov chain stationary expectations. Section 3 describes and analyzes the COSIMLA approach for evaluating Markov chain stationary distributions. Section 4 extends COSIMLA to evaluate Markov jump process stationary distributions and expectations. Section 5 provides a brief account of numerical experiments to illustrate the analysis of COSIMLA.

2 COSIMLA FOR MARKOV CHAIN STATIONARY EXPECTATIONS

Suppose that $X = (X_n : n \geq 0)$ is an irreducible positive recurrent Markov chain with one step transition matrix $P = (P(x,y) : x,y \in S)$. Then, there exists a unique stationary distribution $\pi = (\pi(x) : x \in S)$ (encoded as a row vector) satisfying the linear system of equations

$$\pi = \pi P.$$

Consider the problem of computing the stationary expectation πr, where $r = (r(x) : x \in S)$ is a non-negative "reward" function (encoded as a column vector). For $x \in S$, let $P_x(\cdot)$ be the probability on the path-space of X conditioned on $X_0 = x$, and let $\mathbb{E}_x(\cdot)$ be its associated expectation. Fix $z \in S$ as our "regeneration" state. The theory of regenerative processes guarantees that

$$\pi r = \frac{\mathbb{E}_z \sum_{j=0}^{\tau(z)-1} r(X_j)}{\mathbb{E}_z \, \tau(z)}, \tag{1}$$

where $\tau(z) = \inf\{n \geq 1 : X_n = z\}$ is the first return time to z; see, for example, (Chung 1967).

Choose the finite subset $A \supseteq \{z\}$ of a size for which linear systems involving $|A|$ equations in $|A|$ unknowns can be tractably numerically computed. Put $T = \inf\{n \geq 0 : X_n \in A^c\}$, and note that

$$\begin{aligned}\kappa(r) \triangleq \mathbb{E}_z \sum_{j=0}^{\tau(z)-1} r(X_j) &= \mathbb{E}_z \sum_{j=0}^{(\tau(z)\wedge T)-1} r(X_j) + \mathbb{E}_z \sum_{j=T}^{\tau(z)-1} r(X_j) I(T < \tau(z)) \\ &= \mathbb{E}_z \sum_{j=0}^{(\tau(z)\wedge T)-1} r(X_j) + \mathbb{E}_z h(X_T) I(T < \tau(z)),\end{aligned}$$

where

$$h(x) = \mathbb{E}_x \sum_{j=0}^{\tau(z)-1} r(X_j)$$

for $x \in S$.

Set $A_z = A - \{z\}$, and let $B = (B(x,y) : x,y \in A_z)$ be the matrix in which $B(x,y) = P(x,y)$ for $x,y \in A_z$. Then,

$$\begin{aligned}
\mathbb{E}_z \sum_{j=0}^{(\tau(z)\wedge T)-1} r(X_j) &= r(z) + \sum_{x\in A_z} P(z,x)\, \mathbb{E}_x \sum_{j=0}^{(\tau(z)\wedge T)-1} r(X_j) \\
&= r(z) + \sum_{x\in A_z} P(z,x) \sum_{j=0}^{\infty} \sum_{y\in A_z} P_x(X_j = y, \tau(z)\wedge T > j)\cdot r(y) \\
&= r(z) + \sum_{x\in A_z} P(z,x) \sum_{j=0}^{\infty} \sum_{y\in A_z} B^j(x,y) r(y) \\
&= r(z) + \varphi_z \sum_{j=0}^{\infty} B^j r_A,
\end{aligned}$$

where $\varphi_z = (\varphi_z(x) : x \in A_z)$ is the row vector in which $\varphi_z(x) = P(z,x)$ for $x \in A_z$, and $r_A = (r_A(y) : y \in A_z)$ is the column vector for which $r_A(y) = r(y)$ for $y \in A_z$. We note that because B is a finite non-negative strictly substochastic matrix, $(I-B)^{-1}$ exists and

$$\sum_{j=0}^{\infty} B^j = (I-B)^{-1};$$

see, for example, (Kemeny and Snell 1960).

Also, we observe that for $y \in A^c$,

$$\begin{aligned}
P_z(X_T = y, T < \tau(z)) &= P(z,y) + \sum_{x\in A_z} P(z,x) \sum_{j=0}^{\infty} P_x(X_T = y, T < \tau(z)) \\
&= P(z,y) + \sum_{x\in A_z} P(z,x) \sum_{j=0}^{\infty} P_x(X_{j+1} = y, T\wedge\tau(z) > j) \\
&= P(z,y) + \sum_{x\in A_z} P(z,x) \sum_{j=0}^{\infty} \sum_{w\in A_z} P_x(X_j = w, T\wedge\tau(z) > j) P(w,y) \\
&= P(z,y) + \sum_{x\in A_z} P(z,x) \sum_{j=0}^{\infty} \sum_{w\in A_z} B^j(x,w) P(w,y) \\
&= P(z,y) + \varphi_z \sum_{j=0}^{\infty} B^j p_y \\
&= P(z,y) + \varphi_z (I-B)^{-1} p_y,
\end{aligned}$$

where $p_y = (p_y(x) : x \in A_z)$ is the column vector for which $p_y(x) = P(x,y)$ for $x \in A_z$. Hence, for $y \in A^c$,

$$P_z(X_T = y, T < \tau(z)) = P_z(T < \tau(z)) P_z(X_T = y | T < \tau(z)),$$

where

$$\begin{aligned}
P_z(T < \tau(z)) &= P_z(X_1 \in A^c) + \varphi_z (I-B)^{-1} p_{A^c}, \\
P_z(X_T = y | T < \tau(z)) &= \frac{P(z,y) + \varphi_z (I-B)^{-1} p_y}{P_z(X_1 \in A^c) + \varphi_z (I-B)^{-1} p_{A^c}},
\end{aligned}$$

and $p_{A^c} = (p_{A^c}(x) : x \in A_z)$ is the column vector in which $p_{A^c}(x) = P_x(X_1 \in A^c)$ for $x \in A_z$.

Remark 1 To enhance the above numerical computations, note that if we set $\chi_z = \varphi_z(I-B)^{-1}$, then χ_z satisfies the linear system of equations given by $\chi_z = \varphi_z + \chi_z B$. Having computed χ_z, we can then calculate $\chi_z p_y$ over all the necessary y's by performing as many inner products as there are states in A^c that are accessible in a single transition from A. So, we need not explicitly invert $I-B$.

Suppose that W is a random variable (rv) taking values in A^c, generated from the probability mass function $(P_z(X_T = y|T < \tau(z)) : y \in A^c)$. Conditional on W, we run the Markov chain X starting from W, accumulating the rewards until the chain hits $\tau(z)$, thereby generating the cumulative reward rv R. More precisely,

$$P(R \in \cdot\,|W = x) = P_x\Big(\sum_{j=0}^{\tau(z)-1} r(X_j) \in \cdot\Big),$$

for $x \in A^c$. If $(W_1,R_1),\ldots,(W_n,R_n)$ are n independent and identically distributed (iid) copies of (W,R), then our above argument establishes that if $\bar{R}_n \triangleq n^{-1}\sum_{i=1}^n R_i$, then

$$\kappa_n(r) = r(z) + \chi_z(I-B)^{-1} r_A + P_z(T < \tau(z))\bar{R}_n \tag{2}$$

is an unbiased estimator for the numerator of (1), which was note as $\kappa(r)$. If we denote the denominator of (1) by $\kappa(e)$, we observe that it is a special case of the numerator in which $r = e$, where $e(x) \equiv 1$ for each $x \in S$.

Of course, rather than implementing separate and independent simulations of the numerator and denominator, it will be more efficient to implement these simulations together. To this end, conditional on W, suppose that

$$P((R,V) \in \cdot\,|W = x) = P_x\left(\Big(\sum_{j=0}^{\tau(z)-1} r(X_j), \tau(z)\Big) \in \cdot\right)$$

for $x \in A^c$, where V denotes a rv that represents the hitting time $\tau(z)$. If $(W_1,R_1,V_1),\ldots,(W_n,R_n,V_n)$ are n iid copies of the just defined triplet (W,R,V), then we can estimate πr via

$$\alpha_n \triangleq \kappa_n(r)/\kappa_n(e), \tag{3}$$

where

$$\kappa_n(e) = 1 + \chi_z e_A + P_z(T < \tau(z))\bar{V}_n. \tag{4}$$

Here, $\bar{V}_n = n^{-1}\sum_{i=1}^n V_i$ and $e_A = (e_A : x \in A_z)$ is the column vector in which $e_A(x) = 1$ for $x \in A_z$.

The strong law of large numbers (SLLN) for iid rv's ensures that

$$\alpha_n \to \alpha \quad a.s.$$

as $n \to \infty$, so that α_n is strongly consistent for α, regardless of how accurately π is approximated over A by the linear systems involving the matrix B. Put $p = P_z(T < \tau(z))$, and note that if $Z_i = (R_i - \mathbb{E}R_1) - \alpha(V_i - \mathbb{E}V_i)$ then

$$\alpha_n - \alpha = \frac{p\bar{Z}_n}{\kappa_n(e)}$$

where $\bar{Z}_n = n^{-1}\sum_{i=1}^n Z_i$. Hence, if $\mathbb{E}Z_1^2 < \infty$, it follows that

$$n^{\frac{1}{2}}(\alpha_n - \alpha) \Rightarrow \sigma N(0,1)$$

as $n \to \infty$, where $\Rightarrow$ denotes weak convergence, $N(0,1)$ is a normal rv with mean 0 and unit variance, and $\sigma^2 \triangleq \mathbb{E} Z_1^2 \cdot (p/\kappa(e))^2$.

It follows that if we set

$$\sigma_n^2 = \Big(\frac{p}{\kappa_n(e)}\Big)^2 \frac{1}{n-1} \sum_{i=1}^{n} (R_i - \alpha_n V_i - (\bar{R}_n - \alpha_n \bar{V}_n))^2,$$

then $\sigma_n^2 \to \sigma^2$ a.s. as $n \to \infty$, and $[L_n, R_n]$ is an asymptotic $100(1-\delta)\%$ confidence interval for α (provided $\sigma^2 > 0$), where

$$L_n = \alpha_n - \frac{z\sigma_n}{\sqrt{n}},$$
$$R_n = \alpha_n - \frac{z\sigma_n}{\sqrt{n}},$$

and z is selected so that $P(-z \le N(0,1) \le z) = 1-\delta$. Hence, we have described above a complete algorithm for computing α, and assessing its accuracy (via a confidence interval).

We note that if we compute α via conventional simulation, the standard approach is to simulate X for t time units and to form the time-average

$$\alpha(t) = \frac{1}{t} \sum_{i=0}^{t-1} r(X_i).$$

It is well known that if $\mathbb{E}_z(\sum_{j=0}^{\tau(z)-1} r(X_j) - \alpha\tau(z))^2 < \infty$, then

$$t^{\frac{1}{2}}(\alpha(t) - \alpha) \Rightarrow \nu N(0,1)$$

as $t \to \infty$, where $\nu^2 = \mathbb{E}_z(\sum_{j=0}^{\tau(z)-1} r(X_j) - \alpha\tau(z))^2 / \mathbb{E}_z \tau(z)$; see (Chung 1967). Note that

$$\mathbb{E}_z\Big(\sum_{j=0}^{\tau(z)-1} r(X_j) - \alpha\tau(z) | X_j : T \ge j\Big)$$
$$\stackrel{\mathscr{D}}{=} r(z) - \alpha + \chi_z(r_A - \alpha e_A) + I(T < \tau(z)) Z_1,$$

where $\stackrel{\mathscr{D}}{=}$ means "equality in distribution", and Z_1 is independent of $I(T < \tau(z))$. Because conditional expectations always reduce variance (see, Section V.4 of (Asmussen and Glynn 2007)) and $\kappa(e) = \mathbb{E}_z \tau(z)$, we see that

$$p\frac{\mathbb{E} Z_1^2}{\kappa(e)} \le \frac{\mathbb{E}_z(\sum_{j=0}^{\tau(z)-1} r(X_j) - \alpha\tau(z))^2}{\mathbb{E}_z \tau(z)} = \nu^2. \tag{5}$$

In order to fully compare the efficiency of α_n to $\alpha(t)$, we must also take into account the fact that each of the n observations $Z_1, \ldots, Z_n$ take, on average, $\mathbb{E}(\tau(z) - T | \tau(z) > T)$ Markov chain time steps to generate. Hence, if $\alpha_{N(t)}$ is the COSIMLA estimator in which $N(t)$ is the number of Z_i's completed within a simulation budget of t Markov chain time steps, the central limit theorem (CLT) of (Glynn and Whitt 1992) implies that

$$t^{\frac{1}{2}}(\alpha_{N(t)} - \alpha) \Rightarrow \sqrt{\mathbb{E}(\tau(z) - T | \tau(z) > T)}\,\sigma N(0,1)$$

as $t \to \infty$. But (5) then shows that

$$\mathbb{E}(\tau(z) - T|\tau(z) > T)\sigma^2 = \mathbb{E}(\tau(z) - T)I(\tau(z) > T)p\frac{\mathbb{E}Z_1^2}{\kappa(e)^2}$$
$$\leq \frac{p\mathbb{E}Z_1^2}{\kappa(e)} \leq \nu^2,$$

thereby establishing that $\alpha_{N(t)}$ is a more efficient estimator for α than is $\alpha(t)$. Hence, we are justified in viewing COSIMLA as a variance reduction technique.

We now turn to the numerical linear algebra perspective on COSIMLA. Note that in truncating the transition matrix P to B, we can reasonably approximate πr via

$$\frac{r(z) + \chi_z r_A}{1 + \chi_z e_A} = \frac{\mathbb{E}_z \sum_{j=0}^{(\tau(z)\wedge T)-1} r(X_j)}{\mathbb{E}_z T \wedge \tau(z)}. \tag{6}$$

It is known (see (Seneta 1980) and (Infanger and Glynn 2022)) that the approximation (6) is identical to the unique stationary distribution of the stochastic matrix $\tilde{P} = (\tilde{P}(x,y) : x,y \in A)$, where

$$\tilde{P}(x,y) = \begin{cases} P(x,y) + \sum_{w \in A^c} P(x,w) & , y = z \\ P(x,y) & , y \neq z. \end{cases}$$

The matrix $\tilde{P}$ is known, in the literature on Markov chain numerical computation, as the truncation/augmentation of P associated with "first state" augmentation (in which z is the so-called "first state"). Hence, we can view α_n as a computational vehicle for using simulation to add information on the behavior of X outside the "truncation" set A. Whatever error arises in the approximation (6) is recovered and "fixed" by α_n.

We conclude this section with a discussion of the implementation of the simulation component of the COSIMLA algorithm. Suppose, for the sake of concreteness, that $S = \mathbb{Z}_+^d$ and that $P(x,y) > 0$ only for y's that are "neighbors" of x, namely for y's that are contained in the set $\{w \in S : |w_i - x_i| \leq 1, 1 \leq i \leq d\}$. A natural choice for A would then be a set of the form $\{(x_1, \cdots, x_d) \in S : x_1 + \cdots x_d \leq m\}$, for some $m \geq 1$. Note that $|A|$ is of order m^d, while the subset of A^c accessible from A (i.e. $\{w \in A^c : P(x,w) > 0 \text{ for some } x \in A\}$) is of order m^{d-1}. The support of $P_z(X_T \in \cdot | T < \tau(z))$ is contained in this accessibility subset, so is also of order m^{d-1} (at most).

Suppose that we use the alias method (see, (Walker 1974; Walker 1977)) to generate rv's from $P_z(X_T \in \cdot | T < \tau(z))$. The set-up phase for the alias method therefore has complexity of order $m^{d-1}\log(m)$. If the model has "random walk" structure, the expected number of transitions required to get from the accessibility subset to z is roughly of order m, so each simulation of a Z_i takes roughly order m transitions. Given that the solution of linear systems involving $I - B$ has a complexity of at least order m^{2d} (see (Raz 2002)), one can run at least m^d simulations of Z_1 without the simulation phase of COSIMLA taking more computational effort than that associated with computing the solutions to the linear system that arises here.

3 COSIMLA FOR MARKOV CHAIN STATIONARY DISTRIBUTIONS

In this section, we briefly discuss how COSIMLA can be applied to computing the stationary distribution of X over the set A, namely $\pi_A = (\pi(x) : x \in A)$, with π_A encoded as a row vector. We note that π_A is *not* the stationary distribution π conditioned on A (i.e. $(\pi(x)/\sum_{w \in A} \pi(w)) : x \in A)$; the conditioned distribution can be easily computed from π_A, but not vice versa (so π_A contains more information about π than does the conditional distribution).

We start by recognizing that (1) also applies to a representation for the entire stationary distribution π, so that

$$\pi_A(x) = \frac{\kappa(e_x)}{\kappa(e)},$$

for $x \in A_z$, where $e_x = (e_x(y) : y \in A_z)$ is the column vector in which $e_x(x) = 1$ and $e_x(y) = 0$ for $y \neq x$ and $y \in A_z$. Also, $\pi_A(z) = 1/\kappa(e)$. Then, π_A can be estimated via the row vector $\pi_n = (\pi_n(x) : x \in A)$, where

$$\pi_n(x) = \frac{\kappa_n(e_x)}{\kappa_n(e)}$$

for $x \in A_z$ and

$$\pi_n(z) = \frac{1}{\kappa_n(e)}.$$

Of course, we can simultaneously estimate the $\kappa_n(e_x)$'s, by noting that the row vector $(\kappa_n(e_x) : x \in A_z)$ can be computed as

$$(\kappa_n(e_x) : x \in A_z) \stackrel{\mathscr{D}}{=} \chi_z + p\frac{1}{n}\sum_{i=1}^{n} \Gamma_i,$$

where $(W_1, \Gamma_1), \ldots, (W_n, \Gamma_n)$ are n iid copies of (W, Γ), and

$$P(\Gamma \in \cdot | W = w) = P_w((\sum_{j=0}^{\tau(z)-1} I(X_j = x) : x \in A_z) \in \cdot).$$

In other words, the components of Γ just include counting the total number of visits made to each $x \in A_z$ along the path to z initiated from $w \in A^c$.

As in Section 2, it is evident that π_n is strongly consistent for π_A, and that it satisfies a multivariate CLT, namely

$$n^{\frac{1}{2}}(\pi_n - \pi_A) \Rightarrow \mathscr{G} \tag{7}$$

as $n \to \infty$, where $\mathscr{G} = (\mathscr{G}(x) : x \in A)$ is a Gaussian multivariate normal (row) vector. Furthermore, as in Section 2,

$$\pi_n(x) - \pi(x) = \frac{p\frac{1}{n}\sum_{i=1}^{n}(\Gamma_i(x) - \mathbb{E}\Gamma_1(x) - \pi(x))(V_i - \mathbb{E}V_1)}{\kappa_n(e)}$$

for $x \in A_z$, where $\Gamma_i(x)$ is the x'th entry of the row vector Γ_i. Also,

$$\pi_n(z) - \pi(z) = -p\pi(z)\frac{1}{n}\frac{\sum_{i=1}^{n}(V_i - \mathbb{E}V_1)}{\kappa_n(e)}.$$

One commonly used measure of the quality of π_n as a numerical approximation to π is the total variation distance $\|\pi_n - \pi_A\|_1 \triangleq \sum_{x \in A} |\pi_n(x) - \pi(x)|$. This total variation distance satisfies the limit theorem

$$n^{\frac{1}{2}} \sum_{x \in A} |\pi_n(x) - \pi(x)| \Rightarrow \sum_{x \in A} |\mathscr{G}(x)| \triangleq \|\mathscr{G}\|_1 \tag{8}$$

as $n \to \infty$, where $\mathscr{G}$ is as in (7). This result can be used to estimate $\|\pi_n - \pi_A\|$. In particular, we can estimate the covariance matrix of $\mathscr{G}$ from the simulated data, and then use Monte Carlo to compute the distribution of $\|\mathscr{G}\|_1$.

4 COSIMLA FOR MARKOV JUMP PROCESS STATIONARY DISTRIBUTIONS AND EXPECTATIONS

We now turn to briefly describing how COSIMLA can be applied to computing stationary distributions and expectations for Markov jump processes. In particular, if $X = (X(t) : t \geq 0)$ is an irreducible positive recurrent S-valued Markov jump process with rate matrix $Q = (Q(x,y) : x,y \in S)$, we can use the fact that its (unique) stationary distribution $\pi = (\pi(x) : x \in A)$ can be written in terms of the invariant measure $\tilde{\pi} = (\tilde{\pi}(x) : x \in S)$ associated with its embedded discrete time Markov chain $Y = (Y_n : n \geq 0)$. Recall that Y is an S-valued Markov chain describing the sequence of states visited by X, having a one-step transition matrix $J = (J(x,y) : x,y \in S)$ given by

$$J(x,y) = \begin{cases} Q(x,y)/\lambda(x) & , y \neq x \\ 0 & , y = x, \end{cases}$$

where $\lambda(x) \triangleq -Q(x,x)$ for $x \in S$. Then, π and $\tilde{\pi}$ are connected as follows (see for example (Norris 1997)):

$$\pi(x) = \frac{\tilde{\pi}(x)/\lambda(x)}{\sum_{y \in S} \tilde{\pi}(y)/\lambda(y)}. \tag{9}$$

We note that the positive recurrence of X implies that

$$\sum_{y \in S} \frac{\tilde{\pi}(y)}{\lambda(y)} < \infty.$$

We will also require that Y is positive recurrent, so that

$$\sum_{y \in S} \pi(y)\lambda(y) < \infty.$$

Without the positive recurrent of Y, the expected number of jumps required to compute a regenerative z-cycle of X would be infinite, so that the expected computational effort required to complete a z-cycle of X (or Y) would be infinite.

Suppose $r : S \to \mathbb{R}_+$ is a "reward" function and that we wish to compute πr. In view of (9), we can express πr as

$$\pi r = \frac{\kappa(\tilde{r})}{\kappa(\tilde{e})},$$

where $\tilde{r} = (\tilde{r}(x) : x \in S)$, $\tilde{e} = (\tilde{e}(x) : x \in S)$, $\tilde{r}(x) = r(x)/\lambda(x)$, $\tilde{e}(x) = e(x)/\lambda(x) = 1/\lambda(x)$ and

$$\kappa(\tilde{r}) = \mathbb{E}_z \sum_{j=0}^{\tau(z)-1} \tilde{r}(Y_j),$$

$$\kappa(\tilde{e}) = \mathbb{E}_z \sum_{j=0}^{\tau(z)-1} \tilde{e}(Y_j).$$

We can now compute $\kappa(\tilde{r})$ and $\kappa(\tilde{e})$ exactly as in Section 2. We note that the simulations involve Y rather than X, thereby inducing a conditional Monte Carlo variance reduction known as "discrete time conversion"; see (Fox and Glynn 1986; Fox and Glynn 1990). Similarly, π_A can be computed via $\kappa(\tilde{e}_x)/\kappa(\tilde{e})$ for $x \in A_z$, and $\lambda(z)^{-1}/\kappa(\tilde{e})$ for state z, where $\tilde{e}_x(y) = 1/\lambda(x)$ for $y = x$ and $\tilde{e}_x(y) = 0$ for $y \neq x$.

5 NUMERICAL EXAMPLE

In this section, we present a brief account of numerical experiments to demonstrate the performance of COSIMLA for computing Markov chain stationary expectations. We adopt the notation and setup of Section 2. We consider a discrete-time Markov chain on $S = \mathbb{Z}_+$, in which the state $X = (X_n : n \geq 0)$ evolves according to the stochastic recursion

$$X_{n+1} = [X_n + Z_{n+1} - 1]^+,$$

where $[x]^+ \triangleq \max(x, 0)$ and the Z_n's are iid with $P(Z_1 = 0) = q$, $P(Z_1 = 1) = 1 - p - q$, and $P(Z_1 = 2) = p$. This Markov chain describes the number-in-system process for a slotted time queue in which the system can serve 1 customer in each time slot, and Z_{n+1} represents the number of customers arriving at the beginning of slot $n+1$. The one step transition matrix P of X is a tri-diagonal matrix given by

$$P = \begin{pmatrix} 1-p & p & 0 & 0 & 0 & \cdots \\ q & 1-(p+q) & p & 0 & 0 & \cdots \\ 0 & q & 1-(p+q) & p & 0 & \cdots \\ 0 & 0 & q & 1-(p+q) & p & \\ \vdots & \vdots & & \ddots & \ddots & \ddots \end{pmatrix},$$

where $q > p > 0$ and $p + q \leq 1$. We compare two estimators to compute the stationary expectation πr, where $r(x) = x$ for $x \in S$. The first estimator is the time-average Monte Carlo simulation estimator

$$\alpha(t) = \frac{1}{t} \sum_{i=0}^{t-1} r(X_i),$$

where t is the simulation run-length and X_0 is set to be 0. The second estimator α_n is the COSIMLA estimator defined in (3). For the second estimator, we set the regeneration state z to be 0 and the finite "center" set A to be $\{0, 1, 2, \cdots, M\}$ where $M \in \mathbb{Z}_+$ is a tunable estimator parameter.

We use this birth-death Markov chain to illustrate estimator performance because it provides a closed-form $\alpha = \pi r$, so that we have an explicit true value to facilitate the evaluation of estimators, including the computation of mean squared error (MSE). In addition, we can adjust the ratio $\rho \triangleq p/q$. For the experiment setup, we fix $p + q = 1$ and consider a range of different choices of value for ρ, given from $\{0.8, 0.95, 0.99\}$. For each choice of value for ρ, we implement the two estimators with various choices of the estimator parameters. We present the mean square error and the mean wall-clock time to generate one estimator. The wall-clock time, which is sometimes referred to as elapsed real time, is the actual time taken by the computer to compute an estimator. The means are computed via 1000 replications for each estimator (except for the time-average Monte Carlo estimator's MSE and wall-clock time entry in Table 3, where 50 replications are used due to excessively long computation times). All experiments are conducted on an iMac computer with 3.6 GHz 10-Core Intel Core i9 processor and 64 GB 2667 MHz DDR4 memory.

The MSE and the mean wall-clock time to generate an estimator are reported in Tables 1, 2, 3, in correspondence to the value of ρ being 0.8, 0.95, 0.99. We have the following observations from the reported results.

- The COSIMLA estimator appears to have overall superior performance compared to the time-average Monte Carlo estimator. The COSIMLA estimator incurs much smaller expected wall-clock time to generate, compared to the time-average Monte Carlo estimator, in order to achieve a similar MSE.
- The generation of one COSIMLA estimator involves two tunable parameters M and n. For a fixed M, the larger n is, the smaller the MSE and the larger the mean wall-clock time. For a fixed n, the larger M is, the smaller the MSE and the larger the mean wall-clock time. We recall that the unbiasedness of the COSIMLA estimator is ensured regardless of the choice of M and n.

- When the ratio ρ is close to one, as reflected by $\rho = 0.99$ in Table 3, it can be computationally expensive (taking one hour or more) to generate even one copy of the time-average Monte Carlo estimator to ensure a reasonably low MSE. Table 3 suggests that users of the COSIMLA estimator may tune the flexible parameters M and n to tailor to obtain better performance on MSE and mean walk-clock time.

Table 1: Estimator mean squared error performance, with ratio $\rho = 0.8$.

Estimator\Performance	Mean squared error	Mean wall-clock time
Time-average Monte Carlo		
$\alpha(t)$: $t = 10^6$	3.21×10^{-3}	2.3 seconds
$\alpha(t)$: $t = 4 \cdot 10^6$	8.10×10^{-4}	9.1 seconds
$\alpha(t)$: $t = 16 \cdot 10^6$	2.03×10^{-4}	36.9 seconds
COSIMLA		
$M = 10, n = 1000$	4.71×10^{-3}	0.25 seconds
$M = 10, n = 4000$	1.24×10^{-3}	0.96 seconds
$M = 20, n = 250$	8.12×10^{-4}	0.11 seconds
$M = 20, n = 1000$	1.96×10^{-4}	0.46 seconds
$M = 50, n = 50$	5.16×10^{-8}	0.055 seconds
$M = 50, n = 250$	1.23×10^{-8}	0.28 seconds

Table 2: Estimator mean squared error performance, with ratio $\rho = 0.95$.

Estimator\Performance	Mean squared error	Mean wall-clock time
Time-average Monte Carlo		
$\alpha(t)$: $t = 10^6$	3.21×10^{-3}	3.5 seconds
$\alpha(t)$: $t = 4 \cdot 10^6$	7.90×10^{-4}	13.9 seconds
$\alpha(t)$: $t = 16 \cdot 10^6$	1.93×10^{-4}	56.8 seconds
COSIMLA		
$M = 100, n = 200$	9.9×10^{-3}	1.8 seconds
$M = 100, n = 1000$	2.0×10^{-3}	8.9 seconds
$M = 200, n = 5$	8.01×10^{-5}	0.095 seconds
$M = 200, n = 50$	7.98×10^{-6}	0.94 seconds
$M = 200, n = 200$	1.95×10^{-6}	3.7 seconds

Table 3: Estimator mean squared error performance, with ratio $\rho = 0.99$.

Estimator\Performance	Mean squared error	Mean wall-clock time
Time-average Monte Carlo		
$\alpha(t)$: $t = 10^9$	6.41×10^{1}	2311 seconds
$\alpha(t)$: $t = 4 \cdot 10^9$	–	>1 hour
$\alpha(t)$: $t = 16 \cdot 10^9$	–	>1 hour
COSIMLA		
$M = 200, n = 500$	7.38×10^{0}	49.0 seconds
$M = 500, n = 50$	1.11×10^{0}	11.4 seconds
$M = 500, n = 500$	1.09×10^{-1}	112 seconds
$M = 1000, n = 5$	4.77×10^{-3}	2.43 seconds
$M = 1000, n = 50$	4.81×10^{-4}	24.1 seconds

6 CONCLUSION

This paper is concerned with combining the best features of numerical linear algebra (e.g. solving systems of linear equations) with simulation to compute equilibrium distributions for Markov processes. We propose an algorithm that computes solutions to systems of linear equations for the "center" of the state space supporting most of the equilibrium distribution, and uses simulation to explore the remaining outer state space that is of much larger magnitude. We adopt a principled approach in which the combined algorithm is always guaranteed to produce an estimator that converges to the stationary quantity, regardless of how many states the numerical linear algebraic component of the algorithm encompasses. This combined approach, COSIMLA, for "COmbined SIMulation and Linear Algebra," may have the potential in broader applications that involve Markov processes. One future work is about the use of COSIMLA approach in effectively solving Markov decision processes with large or infinite size state space.

ACKNOWLEDGEMENT

The authors are grateful to the comments and suggestions from the two anonymous reviewers. The authors are grateful to the session and track co-organizers Henry Lam, Haofeng Zhang, Enlu Zhou, and Zhaolin Hu in the "Model uncertainty and robust simulation" track, for their generous support.

REFERENCES

Asmussen, S. 2008. *Applied Probability and Queues*, Volume 51. Springer New York, NY.

Asmussen, S., and P. W. Glynn. 2007. *Stochastic Simulation: Algorithms and Analysis*, Volume 57. Springer New York, NY.

Brailsford, S. C., T. Eldabi, M. Kunc, N. Mustafee, and A. F. Osorio. 2019. "Hybrid Simulation Modelling in Operational Research: A State-of-the-art Review". *European Journal of Operational Research* 278(3):721–737.

Chung, K. L. 1967. *Markov Chains*. Springer.

Fox, B. L., and P. W. Glynn. 1986. "Discrete-time Conversion For Simulating Semi-Markov Processes". *Operations Research Letters* 5(4):191–196.

Fox, B. L., and P. W. Glynn. 1990. "Discrete-time Conversion For Simulating Finite-horizon Markov Processes". *SIAM Journal on Applied Mathematics* 50(5):1457–1473.

Glynn, P. W., and W. Whitt. 1992. "The Asymptotic Efficiency Of Simulation Estimators". *Operations Research* 40(3):505–520.

Heyman, D. P., and M. J. Sobel. 2004. *Stochastic Models in Operations Research: Stochastic Optimization*, Volume 2. Courier Corporation.

Infanger, A., and P. W. Glynn. 2022. "A New Truncation Algorithm For Markov Chain Equilibrium Distributions With Computable Error Bounds". *In preparation*.

Kemeny, J. G., and J. L. Snell. 1960. *Finite Markov Chains*. Springer-Verlag, New York.

Kuntz, J., P. Thomas, G.-B. Stan, and M. Barahona. 2021. "Stationary Distributions Of Continuous-time Markov Chains: A Review Of Theory And Truncation-based Approximations". *SIAM Review* 63(1):3–64.

Norris, J. R. 1997. *Markov Chains*. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press.

Raz, R. 2002. "On the Complexity of Matrix Product". In *Proceedings of the Thiry-Fourth Annual ACM Symposium on Theory of Computing*, STOC '02, 144151. New York, NY, USA: Association for Computing Machinery.

Seneta, E. 1980. "Computing The Stationary Distribution For Infinite Markov Chains". *Linear Algebra and Its Applications* 34:259–267.

Walker, A. J. 1974. "New Fast Method For Generating Discrete Random Numbers With Arbitrary Frequency Distributions". *Electronics Letters* 10(8):127–128.

Walker, A. J. 1977. "An Efficient Method For Generating Discrete Random Variables With General Distributions". *ACM Transactions on Mathematical Software (TOMS)* 3(3):253–256.

AUTHOR BIOGRAPHIES

ZEYU ZHENG is an assistant professor in the Department of Industrial Engineering & Operations Research at University of California Berkeley. He received his Ph.D. in Management Science and Engineering from Stanford University. He has done research in simulation and non-stationary stochastic modeling and decision making. His email address is zyzheng@berkeley.edu and his homepage is http://zheng.ieor.berkeley.edu.

ALEX INFANGER is a Ph.D. candidate in the Institute for Computational and Mathematical Engineering at Stanford University. He currently works on numerical methods for Markov chains and processes. His email address is alexinf@stanford.edu and his homepage is https://stanford.edu/∼alexinf/.

PETER W. GLYNN is the Thomas Ford Professor in the Department of Management Science and Engineering (MS&E) at Stanford University. He is a Fellow of INFORMS and of the Institute of Mathematical Statistics, has been co-winner of Best Publication Awards from the INFORMS Simulation Society in 1993, 2008, and 2016, and was the co-winner of the John von Neumann Theory Prize from INFORMS in 2010. In 2012, he was elected to the National Academy of Engineering. His research interests lie in stochastic simulation, queueing theory, and statistical inference for stochastic processes. His email address is glynn@stanford.edu and his homepage is https://web.stanford.edu/∼glynn/.