

# Efficient Precision Dosing Under Estimated Uncertainties via Koopman Expectations of Bayesian Posteriors with Pumas

Chris Rackauckas<sup>1</sup>, Vaibhav Dixit<sup>2</sup>, Adam R. Gerlach<sup>3</sup>, and Vijay Ivaturi<sup>4</sup>

<sup>1</sup>*Massachusetts Institute of Technology and Pumas-AI Inc.*

<sup>2</sup>*Pumas-AI Inc.*

<sup>3</sup>*United States Air Force Research Laboratory*

<sup>4</sup>*University of Maryland, Baltimore, School of Pharmacy and Pumas-AI Inc.*

## Abstract

Personalized precision dosing is about mathematically determining effective dosing strategies that optimize the probability of containing a patient’s outcome within a therapeutic window. However, the common Monte Carlo approach for generating patient statistics is computationally expensive because thousands of simulations need to be computed. In this manuscript we describe a new method which utilizes the Koopman operator to perform a direct computation of expected patient outcomes with respect to quantified uncertainties of Bayesian posteriors in a nonlinear mixed effect model framework. We detail how quantities such as the probability of being within the therapeutic window can be calculated with a choice of loss function on the Koopman expectation. We demonstrate a high performance parallelized implementation of this methodology in Pumas<sup>®</sup> and showcase the ability to accelerate the computation of these expectations by 50x-200x over Monte Carlo. We showcase how dosing can be optimized with respect to probabilistic statements respecting variable uncertainties using the Koopman operator. We end by demonstrating an end-to-end workflow, from estimating variable uncertainties via Bayesian estimation to optimizing a dose with respect to parametric uncertainty.

## 1 Introduction: The Importance of Precision Dosing

Quantitative clinical pharmacology model-based dose selection in the wider context of model-informed drug development is widely used by the pharmaceutical industry. However, the applications of such technologies have had little to no impact in the clinic till about a few years ago. The challenge for model-based dosing lies in the idea that drug development programs are often incentivized to devise dosing recommendations in the product label primarily for ease of prescribing. This extends to the dosing guidelines for special populations where the label specifies dosing based on certain cut-points for prognostic factors such as age, weight, kidney function or disease status. Historically, the primary reason for such simplified label recommendations was to facilitate approval for the average patient. This, however, is no longer the case as advances in quantitative clinical sciences (*e.g.*, pharmacometrics) and technology enable an individualized approach to drug therapy. For much therapeutics, this signifies an important paradigm shift from a predefined dose to a more tailored and personalized dose aimed to increase efficacy and reduce toxicity.

For effective care, dosing for each patient can be individualized to achieve some desired target goal, such as serum concentrations, or an effect such as bacterial kill or longitudinal glucose measurements over time. Frequent observations of these targets at optimal times will facilitate dose adjustment as needed. However, it is also important that these adjustments are made in real time to avoid delays in decision making that can lead to unintended consequences with regards to safety or loss of efficacy. Target goals are usually defined by therapeutic windows, especially for drugs with a narrow safety margin. For such drugs, therapeutic drug monitoring (TDM) is used to measure *e.g.*, serum concentrations frequently. Based on the observed concentrations and prior clinical experience these drugs are then classified into sub-therapeutic, therapeutic

or toxic levels. Clinical and therapeutic committees in hospital management meet and develop guidelines as to what the acceptable ranges of drug concentration are, which are subsequently used by clinicians to adjust doses to achieve those ranges. Over the years, therapeutic windows have been defined for multiple drugs. While the list of narrow therapeutic index differs across regulatory bodies, a comprehensive list of narrow therapeutic index drugs can be found in DrugBank Online Database\*. Precision dosing has seen a widespread implementation in the areas of infectious disease and transplant, while more recently application in oncology and epilepsy have also surfaced [9].

However, adequately modeling the probability that a drug concentration remains within the therapeutic window remains a challenge. Monte Carlo methods compute the probability by simply computing millions of samples and checking the percentage of results which lie within in the window. Given that model simulations can be computationally time consuming and have slow convergence, it can be impractical to achieve accurate probabilistic estimates through this classical approach.

To address this issue, we describe a mathematical formalism and a new method to facilitate precision dosing. We utilize the Koopman expectation for computing expectations and probabilities of model quantities with respect to estimated parametric uncertainties. We show how to phrase the probability of internal drug concentrations being within the therapeutic window as a quantity that can be calculated via the Koopman expectation. We demonstrate how the Koopman expectation allows for 50x-200x more efficient estimation of these probabilistic quantities compared to Monte Carlo which enables optimization of dosing schemes with respect to uncertainty in less than a second. Additionally, we showcase how the quadrature approach to the calculation of the Koopman expectation yields informative error bounds on the probabilistic estimates, improving their ability to be used and trusted in practice. Together this demonstrates a practical approach for real-time decision making under uncertainty that can be applied to precision dosing scenarios.

## 2 Background: Bayesian Estimation of Nonlinear Mixed Effects Models

Dynamics of drug concentrations (pharmacokinetics, PK) and patient outcomes (pharmacodynamics, PD) on patient populations are commonly modeled via nonlinear mixed effects models (NLME) [4]. The dynamics of the PK/PD system are defined in terms of an ordinary differential equation

$$\dot{u}_i = f(u_i, D_i, \eta_i, \theta, Z_i) \quad (1)$$

where  $D_i$  is the dosage regimen,  $\theta$  are the fixed effects,  $\eta_i$  are the random effects for patient  $i$  and  $Z_i$  are the covariates of patient  $i$ , known measurable quantities of a patient such as their age, weight, or sex. For this model, we assume that  $\eta_i \sim N(0, \Omega)$ , and thus the expected patient outcome given only the prior known information  $Z_i$  is the dynamics predicted by  $\theta$  with  $\eta_i = 0$ .

Given observations  $d$  of the underlying dynamical system  $u$ , fixed effects  $\theta$  and random effects variance  $\Omega$ , the population likelihood  $L$  is given by

$$L(\Omega, \theta, d) \sim \prod_i L_i(u_i, d_i)$$

where  $L_i$  is the likelihood for individual  $i$  which is a chosen functional form. The likelihood is the probability of seeing the data  $d_i$  given a distribution around the predicted value. For example, a common choice of  $L_i$  is the normal distribution centered around the subject's predictions, *i.e.*,  $L_i \sim N(u_i, \sigma)$  for some fixed variance  $\sigma$ . Note that with this choice of likelihood, choosing  $\theta$  and  $\eta_i$  which maximize  $L_i$  is equivalent to choosing the parameters which minimize  $\|u_i - d_i\|$ .

Since we wish to capture the uncertainty in the parameters, instead of performing a deterministic maximum likelihood procedure, we resort to a Bayesian estimation of the posterior distributions for  $\theta$  and  $\eta_i$  given

---

\*<https://go.drugbank.com/categories/DBCAT003972>

assumed prior distributions [3, 22, 24, 18, 27]. Common probabilistic programming languages, such as Stan [7] or Turing [10], can be utilized to determine these posterior distributions using Markov Chain Monte Carlo (MCMC) techniques such as Hamiltonian Monte Carlo [1, 2, 15]. We will demonstrate how estimated distributions of NLME parameters computed using the Pumas<sup>®</sup> software [20] can be reinterpreted as probability distributions via a kernel density estimate (KDE) [16] and used within a Koopman expectation framework to greatly accelerate probabilistic estimates and optimization of clinical choices with respect to uncertainty.

## 3 Uncertainty-Aware Personalized Precision Dosing via the Koopman Expectation

### 3.1 The Frobenius-Perron and Koopman Operators

The mathematical background of the problem and the Koopman expectation is explained in detail in [11]. This is adapted here specifically for pharmacology applications. Assume that Bayesian estimation has given sufficient estimates of posterior distributions for  $\theta$  and  $\eta_i$  and we wish to answer the following question: what is the optimal dosage regimen  $D_i$  for patient  $i$ ? For the rest of this manuscript we will be focusing on this single patient  $i$  and thus the subscript will be dropped.

To motivate the application of the Koopman operator for choosing the optimal dose, we first introduce the Frobenius-Perron (FP) operator on dynamical uncertainty distributions [17]. Define  $S(x) = u(T)$  where  $u(0) = x$ . In other words,  $S(x)$  is the solution of the dynamical system where the initial condition is  $x$ . We wish to compute probabilistic statements on patient outcomes with respect to uncertainty of the initial condition. First, we note how uncertainties in parameters ( $\eta$  and  $\theta$ ) are incorporated into this calculation. For the ordinary differential equation, we can use the extended system:

$$\dot{u} = f(u, D, \eta, \theta, Z) \quad (2)$$

$$\dot{\eta}_i = 0 \quad (3)$$

$$\dot{\theta} = 0 \quad (4)$$

where  $\eta(0) = \eta$  and  $\theta(0) = \theta$ , and thus the parametric uncertainties can be treated as uncertainties in the initial condition on this extended system. Thus without loss of generality, we model the uncertainty via the initial conditions in a probability space  $(\Pi, \mathcal{A}, \mu)$  where  $\Pi = \mathbb{R}^n$ . For a given dynamical system  $S: \mathbb{R}^n \rightarrow \mathbb{R}^n$ , its associated FP operator,  $P_S$ , is defined such that the following equality is satisfied:

$$\int_A P_S f(x) \mu(dx) = \int_{S^{-1}(A)} f(x) \mu(dx), \quad \forall A \in \mathcal{A} \quad (5)$$

where  $S^{-1}(A)$  is the counter-image of  $A$  and  $f$  is the posterior distribution given by the Bayesian estimation [17]. This equivalence is depicted in Figure 1. In terms of precision dosing, the uncertainty in the fixed and random effects, *i.e.*,  $\theta$  and  $\eta$  according to the probability distribution  $f(x)$ , induces a probability distribution on the solution of the dynamics  $u(t)$ . This pushforward of the probability distribution through the dynamical system is the FP operator, *i.e.*,  $P_S f(x) = \mathbb{P}[u(t, \theta, \eta_i) = x]$ .

If  $S$  is both measurable and nonsingular, then  $P_S$  is uniquely defined by Eq. 5 [17]. Monte Carlo estimation of probabilistic quantities of the dynamical system's solution thus correspond to approximating the pushed forward probability mass  $P_S f$  by direct sampling of initial conditions from the probability distribution  $f$ .

In contrast, the Koopman operator  $U_S$  is defined as

$$U_S g(x) = g(S(x)) \quad (6)$$

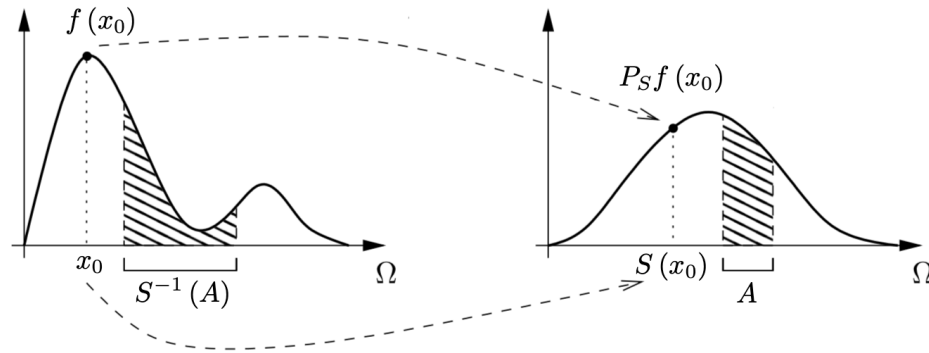


Figure 1: Graphical representation of Eq. 5, where shaded regions have equal area. (Figure adapted from [26])

where  $g(x)$  denotes the observed quantities of the solution to the dynamical system (which we will later call the cost function). Note that  $U_S$  is adjoint to the FP operator,  $P_S$ , i.e.,

$$\langle P_S f, g \rangle = \langle f, U_S g \rangle \quad (7)$$

where  $\langle \cdot, \cdot \rangle$  is the inner product. Eq. 7 can be rewritten as

$$\mathbb{E}[g(X) \mid X \sim P_S f] = \mathbb{E}[U_S g(X) \mid X \sim f] \quad (8)$$

We refer to the left- and right-hand sides of Eq. 8 as the *Frobenius-Perron* (FP) and *Koopman Expectations*, respectively. Figure 2 provides a 1D illustration of Eq. 8. The top row represents the FP Expectation while the bottom row represents the Koopman Expectation. For the top row, the PDF  $f$  (dashed line) is pushed to the right through the system dynamics via  $P_S$  and an inner product is taken with  $g$  (solid line). The expected value,  $\mathbb{E}[g \mid P_S f]$ , is represented by the area of the shaded region. Conversely, on the bottom row, the function  $g$  is pulled to the left through the system dynamics via  $U_S$ . The expected value,  $\mathbb{E}[U_S g \mid f]$ , is represented by the area of the shaded region. The areas of the two shaded regions are equal.

### 3.2 Precision Dosing as Optimization Under Uncertainty

Precision dosing under uncertainty can be phrased as a problem of optimal decision making under uncertainty. Concretely, the optimal dose is one which has the highest expectation of good patient outcomes, *e.g.*, being within a therapeutic window. In other words, we wish to find the dosing schedule  $D$  which optimizes the expectation with respect to a cost function on the solution of the dynamical system. If we let  $g(u(T))$  be the cost associated with a given patient outcome, this corresponds to finding  $D^*$  such that

$$D^* = \arg \min_D \mathbb{E}[g(x) \mid x \sim P_S f] \quad (9)$$

Thus the common way to perform dosing optimization with respect to parametric uncertainty is to utilize a Monte Carlo estimation of  $P_S f$  in order to evaluate the expectation, *i.e.*,

1. Sample parameters  $\theta$  and  $\eta_i$  from the uncertainty distribution  $f$
2. Solve the dynamical system to compute  $S(x)$  for each set of parameters
3. Compute  $g(S(x))$  on each set of parameters, and take the discrete average

This procedure is computationally expensive since it requires the solution of many differential equations. However, using the relationship of Equation 8, we see that the argument of the optimization problem is

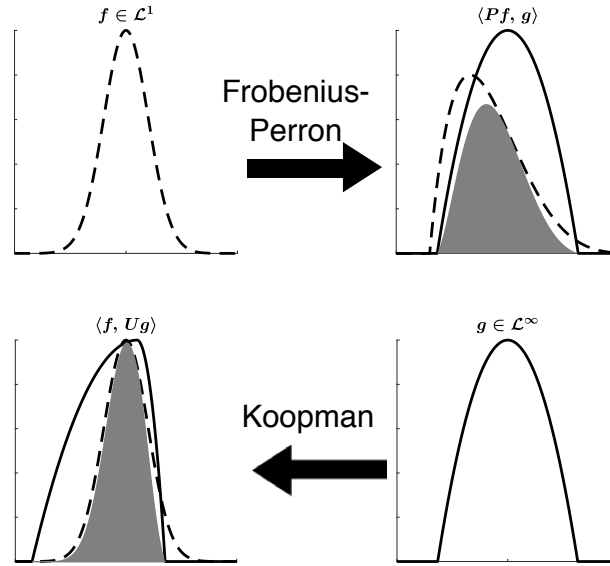


Figure 2: Illustration of the FP and Koopman operator adjoint property. The inner products, represented by the area of the filled regions, are equivalent. (Figure reproduced from [19])

simply the Frobenius-Perron expectation, and thus it can equivalently be rephrased in terms of the Koopman expectation [11]:

$$D^* = \arg \min_D \mathbb{E}[U_S g(x) | x \sim f] \quad (10)$$

Notice from Figure 2 that the Koopman operator pulls back to the original probability distribution. Thus we can explicitly represent the calculation as:

$$\mathbb{E}[U_S g(X) | X \sim f] = \int_{\Pi} U_S g(x) f(x) dx \quad (11)$$

In other words, this means the desired expectation can be calculated via a multidimensional quadrature where  $U_S g(x)$  is the solution of the dynamical system at parameters determined by the quadrature procedure and  $f(x)$  is the evaluation of the probability distribution at the quadrature points.

### 3.3 Representation of Therapeutic Windows in Cost Functions

Clinical experience via TDM establish safety guidelines known as the therapeutic window. For example, for a given drug the area under the curve (the total concentration or AUC) of the drug over 24 hour periods may be known to be safe when it is between 200 and 400 mg.hr/L. Given variability in drug disposition and elimination, the goal is to optimize the probability that the dose will be in the therapeutic window with respect to the uncertainty in the patient-specific effects.

To calculate this probability, let  $g(X) = \chi_W(X)$  where  $\chi_W$  is an indicator function for the therapeutic window  $W$ , *i.e.*,  $g$  is a function defined as 1 if the solution to the differential equation with parameters  $(\theta, \eta, D, Z)$  falls in the therapeutic window, and 0 otherwise. With this definition of the cost function, we note that the expectation of a characteristic function is the probability that the event occurs, and as such this definition of  $g$  gives rise to:

$$D^* = \arg \min_D \mathbb{P}[u(t) \in W] \quad (12)$$

Therefore under this choice of  $g$  the optimal dosing regimen  $D^*$  computed by the optimization over the Koopman expectation is the dosing regimen that has the maximal probability of the patient's outcomes to be in the therapeutic window. We note in passing that this formulation is slightly more general, as it allows for a quantification of “badness”, *i.e.*, the cost could be a distance from the therapeutic window which encapsulates the idea that being close to the therapeutic window might be sufficient while being far away incurs a greater risk on the patient. Generalizations to this computation which include higher order statistics and process noise are given in [11].

## 4 Efficient Computation of the Koopman Expectation in Pumas<sup>®</sup>

### 4.1 Description of the Implementation and Features

To demonstrate the utility of this method for performing dosage optimization under uncertainty, we created a high-performance implementation of the Koopman expectation in Pumas<sup>®</sup>. Equation 11 was implemented by utilizing DifferentialEquations.jl [21] to calculate  $U_{Sg}$  given a Pumas<sup>®</sup> [20] specification of a dynamical system and the multidimensional integral was calculated using Quadrature.jl, a wrapper library over common quadrature methods such as Cuba [13] and Cubature [12, 6]. This integration implementation allows for a batch solve that parallelizes the computation over the quadrature points, allowing for multithreaded, distributed, and GPU acceleration of the quadrature. The quadrature techniques were made differentiable in order to allow automatic differentiation use for optimization under uncertainty.

### 4.2 Calculating Probabilities With Respect to Theophylline Dosing

The following code demonstrates the use of the Koopman expectation for the calculation of the probability that the AUC will be below 300 on the Theophylline model. The nonlinear mixed effect model's definition is defined using Pumas<sup>®</sup>' standard `@modelmacro` form:

```
using Pumas, Quadrature

model = @model begin
    @param begin
         $\theta \in \text{VectorDomain}(4)$ 
    end

    @covariates sex wt etn

    @pre begin
        Ka =  $\theta[1]$ 
        CL =  $\theta[2] * ((wt/70)^{0.75}) * (\theta[4]^{\text{sex}})$ 
        V =  $\theta[3]$ 
    end

    @dynamics begin
        Depot' = -Ka*Depot
        Central' = Ka*Depot - CL*Central/V
    end

    @derived begin
        cp = Central ./ V
        nca := @nca cp
        auc = NCA.auc(nca)
```

```
end
end
```

Then to define the subject and the dosage regimen, we read in a dataset:

```
subject = read_pumas(datapath,cvs = [:sex,:wt,:etn])[1] # 1 for patient 1
```

Next we input the parameter uncertainty distributions. This would normally come from Bayesian posteriors as described in Section 2, but here we will directly prescribe distributions as an illustrative example.

```
param_dists = (theta = [Uniform(1.0,5.0),
                        Uniform(20.0,100.0),
                        Uniform(200.0,400.0),
                        Uniform(0.1,2.0)])
```

We then define the therapeutic window via the cost function  $g$  on the observables. Notice that we used the Pumas<sup>®</sup> NLME integration with NCA to compute the AUC as part of the derived variables from the model, and thus the AUC exists as one of the observed outputs of the system. We can thus define the therapeutic window via:

```
function g(obs)
    obs[:auc] < 300
end
```

Now we simply call expectation and tell it to use the KoopmanExpectation method. In there we designate that we would like to use the HCubatureJL quadrature method, which tends to be efficient for low dimensional integrands (<8 uncertain variables).

```
expectation(g,KoopmanExpectation(CubatureJLp()),model,subject,param_dists)
```

This computation will then compute the given expectation to the desired tolerance. Given its differentiability, this function can then be used inside of an optimization loop with defined dosage regimens to thus perform dosage optimization under uncertainty.

### 4.3 Efficiency and Robustness of the Koopman Expectation vs Monte Carlo

This extra formalism is useful because of the tremendous performance and feature benefits. To demonstrate this, we compared the KoopmanExpectation to the MonteCarloExpectation option with various increasing choices of imaxiters (allowed number of ODE solves) to determine the rate of convergence of the two methods for calculating the probability of patient outcomes occurring in the therapeutic window. Figure 3 demonstrates two results. One is that the Koopman method converges to give stable probability estimates of the second digit with approximately 50x less ODE solves. Secondly, by utilizing the HCubature method the Koopman calculation not only determines the probability but also gives numerical error bounds on the probability estimate, allowing the user to effectively know the uncertainty introduced by the numerical error. Such a bound is highly difficult to calculate with Monte Carlo estimates given the slow rate of convergence of the variance. Together, this demonstrates the Koopman expectation as both a method for efficiency and robustness.

While the quadrature-based Koopman significantly outperforms Monte Carlo, we note that the chosen observable, a characteristic function, was a discontinuous function. The efficiency of quadrature methods can be considerably improved when the integrands are smooth. Thus to test the efficiency in the smooth case, we used the Koopman method to estimate the expected value of the AUC via:

```
function g(obs)
    obs[:auc]
end
```



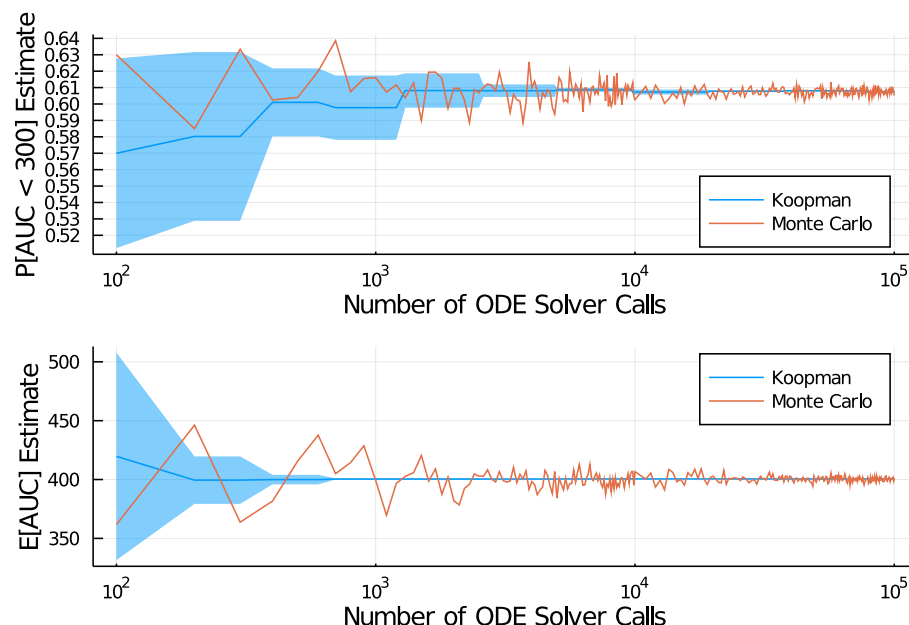


Figure 3: Convergence of the Koopman Expectation calculation via Cubature’s CubatureJLp on the integral (blue line) vs Monte Carlo (orange line), measured in terms of the number of ODE solver calls required. Note that the cubature integration method used with the Koopman Expectation comes with a free error estimation (blue region). **Top:**  $P[AUC < 300]$ . **Bottom:**  $E[AUC]$ .

In this case Figure 3 shows that it takes only 500 ODEs solves for the Koopman method to certify that  $E[AUC] \approx 400 \pm 1$ . Meanwhile the Monte Carlo approach is unable to establish a stable estimate within 1 unit with 100,000 ODE solves, with the estimate at 99500 solves giving 398.91 and 10000 giving 401.13. Therefore we see that the Koopman method in this case is effectively 200x as efficient as the classical Monte Carlo method.

#### 4.4 Accelerated Bayesian Precision Dosing Using Pumas<sup>®</sup>

In this next example we wish to use the accelerated expectation estimate to optimize a dosing scheme with respect to a probabilistic quantity. We will do so on the Theopylline model with the included example data in Pumas<sup>®</sup>. Let’s start by defining a one-compartment model and fitting it using Bayesian estimation against the example data. This looks like:

```
using Pumas, Quadrature, KernelDensity, Optim, Plots, StatsPlots
# Read the data
data = read_pumas(example_data("THEOPP"))

m_diffEq = @model begin
    @param begin
        θ ∈ VectorDomain(3,init=[30, 0.9, 1.0], lower=zeros(3), upper = [50.0, 1.0, 3.0])
        Σ ∈ PDiagDomain(init=[1.70385E-02, 8.28498E-02])
    end
    @random begin
        η ~ MvNormal(Matrix(I,2,2))
    end
    @pre begin
```



```

Vc =  $\theta$ [1]
Ke =  $\theta$ [2]
Ka =  $\theta$ [3]
CL = Ke * Vc
end
@vars begin
    conc = Central / Vc
end
@dynamics Depots1Central1
@derived begin
    dv ~ @. Normal(conc, sqrt(conc^2 *  $\Sigma$ .diag[1] +  $\Sigma$ .diag[end]) + eps())
end
end
param = init_param(m_diffEq)
ftbayes = fit(m_diffEq, data, param, Pumas.BayesMCMC(), nsamples = 1000)
chns = Pumas.Chains(ftbayes)
plot(chns) # Plot the posterior distributions.

```

A plot of the posterior distributions for the  $\theta$  terms is given in Figure 4. Given these Bayesian estimates, we can do one of two things to get uncertain parameter estimates. One way is to directly use the Kernel Density Estimate (KDE) to reinterpret the chains into probability distributions. This is done as follows:

```
posts_dists = ( $\theta$  = [kde(chns.value.data[50:end,i,1]) for i in 1:3],  $\Sigma$  = param. $\Sigma$ , )
```

This is exactly the same as how the density is shown in Figure 4. However, one can also simplify the distributions by using a maximum likelihood fit against a standard distribution. For example, we can find the best fitting normal distributions for our parameters using the `fit` overloads from `Distributions.jl` as follows:

```
posts_dists = ( $\theta$  = [fit(Normal,chns.value.data[50:end,i,1]) for i in 1:3],  $\Sigma$  = param. $\Sigma$ , )
```

We will continue the tutorial using these distributions for simplicity, though either will give similar results. Note that to stabilize the distributions we chopped off the earliest part of the chain, a procedure known as burn in to reduce the effect of the chain's random starting guess [8, 14, 25, 5, 23].

Now let's find the starter dose for which, on average given the estimated uncertainty in the population parameters, has the best chance of having an AUC around 30 in the first five hours. To do so, we make our observable be the AUC calculation using the `NCA.jl` module included in `Pumas`<sup>®</sup>:

```

function g(obs)
    ncasub = NCA.NCASubject(obs[:conc], obs.time)
    NCA.auc(ncasub, interval = (0.0, 5.0))
end

```

Next we design a new model which makes the maximum and minimum allowed values for the parameters the same as the maximum and minimum of the estimated Bayesian posteriors (to improve the Koopman calculation) and use `Optim.jl` to find the dose whose average AUC is 30.

```

m_diffEq2 = @model begin
    @param begin
         $\theta$  ∈ VectorDomain{3, lower=[minimum.([chns.value.data[50:end,i,1] for i in 1:3])...],
        upper=[maximum.([chns.value.data[50:end,i,1] for i in 1:3])...]}
         $\Sigma$  ∈ PDiagDomain(init=[1.70385E-02, 8.28498E-02])
    end
    @pre begin
        Vc =  $\theta$ [1]
    end
end

```

```

    Ke =  $\theta$ [2]
    Ka =  $\theta$ [3]
    CL = Ke * Vc
end
@vars begin
    conc = Central / Vc
end
@dynamics Depots1Central1
@derived begin
    dv ~ @. Normal(conc, sqrt(conc^2 *  $\Sigma$ .diag[1] +  $\Sigma$ .diag[end]) + eps())
end
end

function f(dose)
    subject = Subject(events=DosageRegimen(dose,time=0))
    exp_koop = expectation(g, KoopmanExpectation(HCubatureJL()),
                          m_diffeq2,subject,posts_dists,
                          iabstol=1e-2, ireltol=1e-2).u
    (exp_koop - 30)^2
end
using Optim
@time res = Optim.optimize(f,0.0,500.0,abs_tol=0.1)
@show res.minimizer, f(res.minimizer)

```

The optimization suggests that the optimal dose to choose to achieve an AUC of 30, respecting the long uncertainty tails in the parameter estimates, is approximately 311. The total time for the optimization was approximately 0.05 seconds.

For reference we can consider the time against Monte Carlo. Accurate gradients are required in order for the optimization to be stable, and thus the optimization procedure needs more digits of accuracy than required by the user. Thus an optimization of the AUC to two digits requires at least 3 stable digits. We find that needs approximately 50000 ODE solves via Monte Carlo, via the code:

```

@time mean(map(1:50000) do idx
    sample_parmas = ( $\theta$  = rand.(posts_dists. $\theta$ ),  $\Sigma$  = param. $\Sigma$ , )
    g(simobs(m_diffeq2,data[1],sample_parmas))
end)

```

which in three runs produced [36.41,36.45,36.47], showing the 3rd digit is almost stable. However, the time to calculate this estimate is approximately 1.1 seconds, meaning **the entire uncertain dosage optimization with the Koopman expectation is faster than accurately calculating the expectation using Monte Carlo at one point!** We can then see the total effect when trying to optimize a dose with respect to uncertainty:

```

function f2(dose)
    subject = Subject(events=DosageRegimen(dose,time=0))
    exp_koop = mean(map(1:50000) do idx
        sample_parmas = ( $\theta$  = rand.(posts_dists. $\theta$ ),  $\Sigma$  = param. $\Sigma$ , )
        g(simobs(m_diffeq2,data[1],sample_parmas))
    end)
    (exp_koop - 30)^2
end
@time res = Optim.optimize(f2,0.0,500.0,abs_tol=0.1)
@show res.minimizer, f2(res.minimizer)

```

This finds the optimal dose to be around 302 before the optimizer exits due to the stochastic variation of the objective function (reaching a final loss of approximately 41, as opposed to the previous  $< 1e - 8$ ). Multiple runs of the optimizer may be required as the stochastic objective can cause the optimizer to have non-deterministic behavior. In addition, it took approximately 8.5 seconds, or around 170x as long as the Koopman method while not attaining the same accuracy. Because this is done purely by reducing the number of ODE solves required to calculate the expectations, it is expected that such gaps continue to exist as models and data get more expensive to simulate and estimate.

## 5 Discussion

As personalized precision dosing becomes more commonplace, efficient and robust methods for optimizing dosing with respect to probabilistic outcomes will become more central to pharmacometric practice. As personalized computations become pervasive, more accurate models will be required, and the computations will need to migrate to the patient, living on embedded or mobile devices with little compute power. In this manuscript we demonstrated how utilizing the Koopman expectation can accelerate the computation 50x-200x over Monte Carlo while giving numerical error bounds for improved clarity of the results. This methodology is implemented with the high performance differential equation solvers of DifferentialEquations.jl within the Pumas<sup>®</sup> pharmacometrics suite so that existing models can automatically be compatible with the accelerated computation. Being a pure Julia software stack, this methodology can compile to mobile devices like ARM for direct deployment to patients. Subsequent studies will detail the effect of this methodology in real-world scenarios through its effect on the Lyv ([www.lyv.ai](http://www.lyv.ai)) dosing system.

## References

- [1] Michael Betancourt. “A conceptual introduction to Hamiltonian Monte Carlo”. In: *arXiv preprint arXiv:1701.02434* (2017).
- [2] Michael Betancourt and Mark Girolami. “Hamiltonian Monte Carlo for hierarchical models”. In: *Current trends in Bayesian methodology with applications* 79.30 (2015), pp. 2–4.
- [3] Prithwish Bhaumik and Subhashis Ghosal. “Bayesian estimation in differential equation models”. In: *arXiv preprint arXiv:1403.0609* (2014).
- [4] Peter L Bonate et al. *Pharmacokinetic-pharmacodynamic modeling and simulation*. Vol. 20. Springer, 2011.
- [5] Stephen P Brooks and Gareth O Roberts. “Assessing convergence of Markov chain Monte Carlo algorithms”. In: *Statistics and Computing* 8.4 (1998), pp. 319–335.
- [6] JM Bull and TL Freeman. “Parallel globally adaptive algorithms for multi-dimensional integration”. In: *Applied numerical mathematics* 19.1-2 (1995), pp. 3–16.
- [7] Bob Carpenter et al. “Stan: A probabilistic programming language”. In: *Journal of statistical software* 76.1 (2017).
- [8] Mary Kathryn Cowles and Bradley P Carlin. “Markov chain Monte Carlo convergence diagnostics: a comparative review”. In: *Journal of the American Statistical Association* 91.434 (1996), pp. 883–904.
- [9] Adam S Darwich et al. “Model-informed precision dosing: background, requirements, validation, implementation, and forward trajectory of individualizing drug therapy”. In: *Annual Review of Pharmacology and Toxicology* 61 (2021), pp. 225–245.
- [10] Hong Ge, Kai Xu, and Zoubin Ghahramani. “Turing: a language for flexible probabilistic inference”. In: *International Conference on Artificial Intelligence and Statistics, AISTATS 2018, 9-11 April 2018, Playa Blanca, Lanzarote, Canary Islands, Spain*. 2018, pp. 1682–1690.

- [11] Adam R Gerlach et al. “The Koopman Expectation: An Operator Theoretic Method for Efficient Analysis and Optimization of Uncertain Hybrid Dynamical Systems”. In: *arXiv preprint arXiv:2008.08737* (2020).
- [12] Ian Gladwell. “Vectorisation of one dimensional quadrature codes”. In: *Numerical integration*. Springer, 1987, pp. 231–238.
- [13] Thomas Hahn. “Cuba—a library for multidimensional numerical integration”. In: *Computer Physics Communications* 168.2 (2005), pp. 78–95.
- [14] Ghassan Hamra, Richard MacLehose, and David Richardson. “Markov chain Monte Carlo: an introduction for epidemiologists”. In: *International journal of epidemiology* 42.2 (2013), pp. 627–634.
- [15] Matthew D Hoffman and Andrew Gelman. “The No-U-Turn sampler: adaptively setting path lengths in Hamiltonian Monte Carlo.” In: *J. Mach. Learn. Res.* 15.1 (2014), pp. 1593–1623.
- [16] JooSeuk Kim and Clayton D Scott. “Robust kernel density estimation”. In: *The Journal of Machine Learning Research* 13.1 (2012), pp. 2529–2565.
- [17] Andrzej Lasota and Michael C. Mackey. *Chaos, Fractals, and Noise: Stochastic Aspects of Dynamics*. Springer Science & Business Media, Nov. 27, 2013. 481 pp. ISBN: 978-1-4612-4286-4.
- [18] Chantal Le Guellec et al. “Population pharmacokinetics and Bayesian estimation of mycophenolic acid concentrations in stable renal transplant patients”. In: *Clinical pharmacokinetics* 43.4 (2004), pp. 253–266.
- [19] Andrew Leonard. “Probabilistic Methods for Decision Making in Precision Airdrop”. GA Tech, Jan. 15, 2019. 141 pp.
- [20] Christopher Vincent Rackauckas et al. “Accelerated Predictive Healthcare Analytics with Pumas, A High Performance Pharmaceutical Modeling and Simulation Platform”. In: *bioRxiv* (2020).
- [21] Christopher Rackauckas and Qing Nie. “DifferentialEquations.jl – A Performant and Feature-Rich Ecosystem for Solving Differential Equations in Julia”. In: *J. Open Res. Softw.* 5.1 (1 May 25, 2017), p. 15. ISSN: 2049-9647. DOI: 10.5334/jors.151.
- [22] Lewis B Sheiner and Stuart L Beal. “Bayesian individualization of pharmacokinetics: simple implementation and comparison with non-Bayesian methods”. In: *Journal of pharmaceutical sciences* 71.12 (1982), pp. 1344–1348.
- [23] Sandip Sinharay. “Assessing convergence of the Markov chain Monte Carlo algorithms: A review”. In: *ETS Research Report Series* 2003.1 (2003), pp. i–52.
- [24] AH Thomson and B Whiting. “Bayesian parameter estimation and population pharmacokinetics”. In: *Clinical pharmacokinetics* 22.6 (1992), pp. 447–467.
- [25] Nils Toft et al. “Assessing the convergence of Markov Chain Monte Carlo methods: an example from evaluation of diagnostic tests in absence of a gold standard”. In: *Preventive veterinary medicine* 79.2-4 (2007), pp. 244–256.
- [26] Andrea Yeong Weiße. “Global Sensitivity Analysis of Ordinary Differential Equations: Adaptive Density Propagation Using Approximate Approximations”. In: (2009). Accepted: 2018-06-07T22:43:56Z. DOI: <http://dx.doi.org/10.17169/refubium-13784>.
- [27] Rebecca E Wrishko et al. “Vancomycin pharmacokinetics and Bayesian estimation in pediatric patients”. In: *Therapeutic drug monitoring* 22.5 (2000), pp. 522–531.

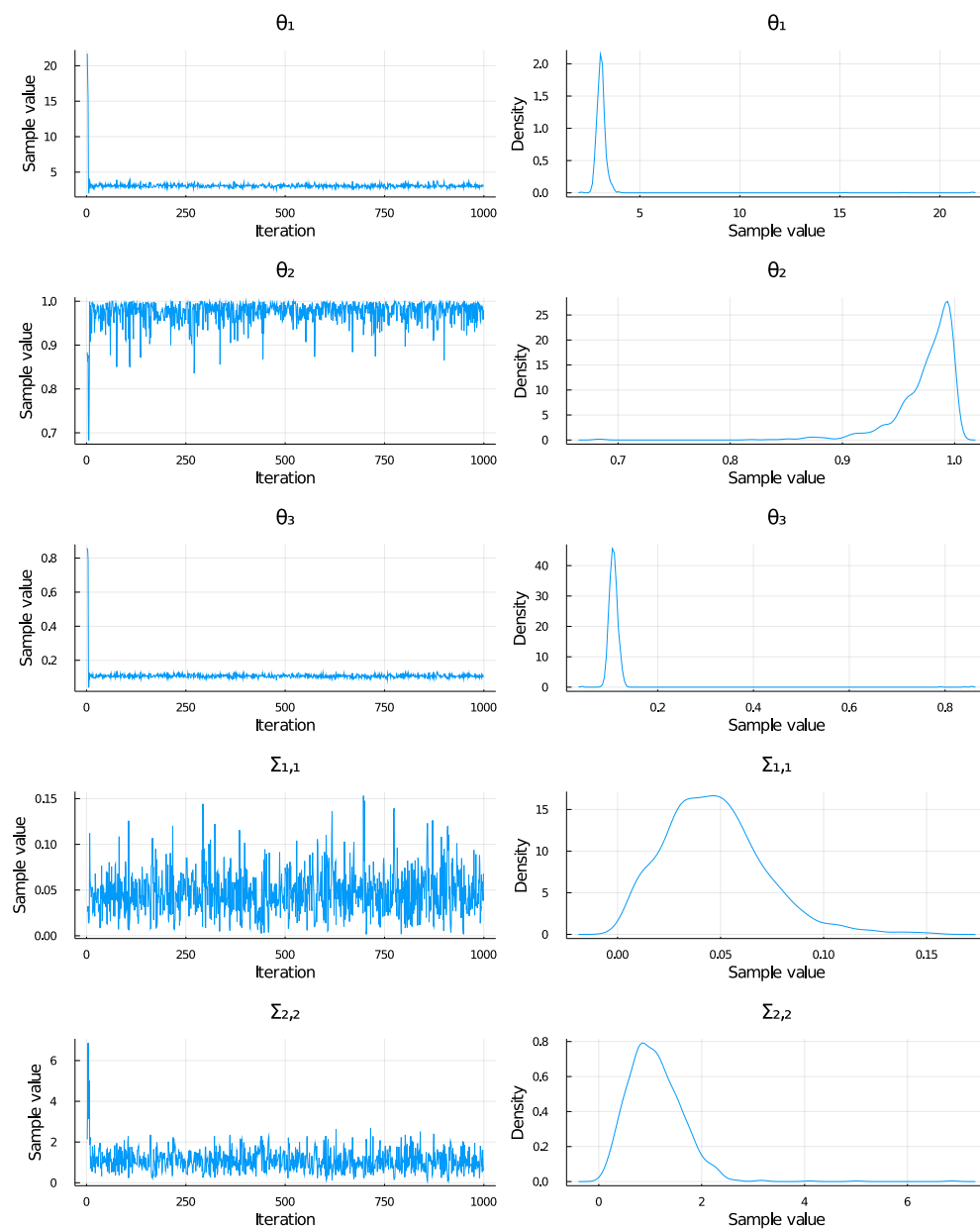


Figure 4: Posterior distributions and MCMC Chains obtained using HMC in Pumas<sup>®</sup> for bayesian estimation of parameters