Simultaneous perturbation gradient approximation based Metropolis adjusted Langevin Markov chain Monte Carlo for inference of ordinary differential equations

Ivan Vujačić[*1] **and Mathisca de Gunst**[1]

[1] *VU University Amsterdam*

Abstract: The problem of parameter estimation for models defined by a system of ordinary differential equations (ODEs) is considered. The most efficient way to explore the parameter space is by using derivative information. Usual approaches for obtaining the gradient in ODEs setting like solving sensitivity equations and using finite difference formulas are computationally costly and not scalable to large scale systems. In this paper we use simultaneous perturbation gradient approximation (SPGA), originally proposed in stochastic optimization literature, as a substitute for the gradient in Metropolis adjusted Langevin algorithm (MALA). The obtained algorithm, called Simultaneous Perturbation Gradient Approximation based Metropolis Adjusted Langevin Markov chain Monte Carlo (SPGA MALA), requires at most three integration of the ODE system per MCMC step, regardless of the dimension of the system. This fixed computational costs makes SPGA MALA applicable to large scale systems. On the other hand, its efficiency is comparable to that of MALA. We demonstrate its performance of via simulations.

Keywords: Metropolis adjusted Langevin Markov Chain Monte Carlo methods, Simultaneous perturbation gradient approximation, ordinary differential equations, parameter estimation.
AMS subject classifications: 60J22, 65C40, 62F15.

1 Introduction

Systems of ordinary differential equations (ODEs) are widely used in science and engineering for the mathematical modelling of various dynamic processes. We consider the system of the form

$$\begin{cases} \boldsymbol{x}'(t) = \boldsymbol{f}(\boldsymbol{x}(t), t; \boldsymbol{\theta}), \ t \in [0, T], \\ \boldsymbol{x}(0) = \boldsymbol{\xi}, \end{cases} \tag{1}$$

where $\boldsymbol{x}(t) = (x_1(t), \ldots, x_d(t))^\top \in \mathbb{R}^d$ is a state vector, $\boldsymbol{\xi}$ in $\Xi \subset \mathbb{R}^d$ is the initial condition, $\boldsymbol{\theta}$ in $\Theta \subset \mathbb{R}^p$ is a parameter and $\boldsymbol{f}$ is a known function. Given the values of $\boldsymbol{\xi}$ and $\boldsymbol{\theta}$, we denote the solution of (1) by $\boldsymbol{x}(t; \boldsymbol{\theta}, \boldsymbol{\xi})$. Let us assume that a process is modelled by the system (1) with $\boldsymbol{\xi}_0$ known and $\boldsymbol{\theta}_0$ unknown. For

[*]Corresponding author: i.vujacic@vu.nl

simplicity, assume that we have noisy observations $y_i(t_j)$, $j = 1, \ldots, n$ of all the states $x_i(t; \boldsymbol{\theta}_0, \boldsymbol{\xi}_0)$, $i = 1, \ldots, d$ at time points $t_j \in [0, T]$, $j = 1, \ldots, n$:

$$y_i(t_j) = x_i(t_j; \boldsymbol{\theta}_0, \boldsymbol{\xi}_0) + \varepsilon_i(t_j), \quad i = 1, \ldots, d; j = 1, \ldots, n,$$

where $\varepsilon_i(t_j) \sim \mathcal{N}(0, \sigma_i^2)$. The problem is to estimate $\boldsymbol{\theta}_0$ from the data $\mathbf{Y} = (y_i(t_j))_{ij}$. The methodology presented here can also be used if $\boldsymbol{\xi}_0$ is unknown and some of the states are unobserved.

In this paper, we adopt Bayesian approach to inference. For some prior density π of $\boldsymbol{\theta}$ the posterior density is

$$p(\boldsymbol{\theta}|\mathbf{Y}, \boldsymbol{\xi}_0, \boldsymbol{\sigma}) \propto \pi(\boldsymbol{\theta}) \prod_{j=1}^{d} \mathcal{N}\{\mathbf{Y}_{j,\cdot}|\mathbf{X}(\boldsymbol{\theta}, \boldsymbol{\xi}_0)_{j,\cdot}, \sigma_j \mathbf{I}_n\},$$

where $\boldsymbol{\sigma} = (\sigma_1, \ldots, \sigma_d)$, $\mathbf{X}(\boldsymbol{\theta}, \boldsymbol{\xi}_0) = (x_i(t_j; \boldsymbol{\theta}, \boldsymbol{\xi}_0))_{ij}$ and $\mathbf{I}_n$ is an identity matrix of order n. For exploring the parameter space there is an advantage in using gradient information in MCMC and optimization methods [3, 4]; for concrete examples in ODE estimation setting see [4, 6]. In the problem we consider, the gradient of the log-likelihood can be obtained by solving sensitivity equations, which are of order dp or via the finite difference formulas, which require solving the ODE system at least p times. Both approaches are computationally costly and not scalable to large scale systems.

In this paper, we avoid huge computational burden by using simultaneous perturbation stochastic approximation (SPGA), introduced by Spall [7]. To obtain SPGA, the system of the form (1) need be solved at most 2 times, regardless of the dimension of the system. By using SPGA instead of the gradient in Metropolis adjusted Langevin Markov Chain Monte Carlo (MALA) we obtain a method, which we call SPGA MALA, that can be used for large scale systems. Although there is some loss in efficiency of SPGA MALA due to using an approximation of the derivative this is outweighed by huge computational savings achieved.

The rest of the paper is organized as follows. In sections 2 and 3 reviews of MALA and SPGA are provided, respectively. Section 4 introduces the proposed method. In Section 5 we compare performance of MALA and SPGA MALA on simulated data for various models.

2 Metropolis adjusted Langevin Markov chain Monte Carlo (MALA)

For the probability density $p(\boldsymbol{\theta})$ let $\mathcal{L}(\boldsymbol{\theta}) = \log\{p(\boldsymbol{\theta})\}$ denote the log-density. The MALA proposal [4, p.130] is

$$\boldsymbol{\theta}^* = \boldsymbol{\theta}^k + \epsilon^2 \mathbf{M} \nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}^k)/2 + \epsilon \sqrt{\mathbf{M}} \boldsymbol{z}^k, \tag{2}$$

where $\boldsymbol{\theta}^k$ is the value at k-th step, $\boldsymbol{z} \sim \mathcal{N}(\boldsymbol{z}|\mathbf{0}, \mathbf{I}_p)$, $\epsilon > 0$ is the step size and $\mathbf{M}$ is the weight matrix. The proposal density and acceptance probability are

$$\begin{aligned} &q(\boldsymbol{\theta}^*|\boldsymbol{\theta}^k) = \mathcal{N}(\boldsymbol{\theta}^*|\boldsymbol{\mu}(\boldsymbol{\theta}^k, \epsilon), \epsilon^2 \mathbf{M}), \\ &\alpha = \min\{1, p(\boldsymbol{\theta}^*) q(\boldsymbol{\theta}^k|\boldsymbol{\theta}^*)/p(\boldsymbol{\theta}^k) q(\boldsymbol{\theta}^*|\boldsymbol{\theta}^k)\}, \end{aligned} \tag{3}$$

respectively, where $\boldsymbol{\mu}(\boldsymbol{\theta}^k, \epsilon) = \boldsymbol{\theta}^k + \epsilon^2 \mathbf{M} \nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}^k)/2$. The advantage of MALA over random walk Metropolis algorithm is that it uses the gradient information which leads to better exploration of the parameter space. Disadvantage is that it requires selection of the weight matrix $\mathbf{M}$. In [4], a fully automated algorithm is proposed for this but it cannot be used in our setting because it requires derivatives. For more details regarding MALA see [2, 4].

3 Simultaneous perturbation gradient approximation (SPGA)

In order to estimate partial derivatives via finite difference (FD) approximation the parameter perturbations are performed along each coordinate separately. For example, the estimate of the j-th partial derivative of $\mathcal{L}(\boldsymbol{\theta}^k)$ via the central difference formula is

$$\frac{\partial \mathcal{L}(\boldsymbol{\theta}^k)}{\partial \theta_j} \approx \frac{\mathcal{L}(\boldsymbol{\theta}^k + h\boldsymbol{e}_j) - \mathcal{L}(\boldsymbol{\theta}^k - h\boldsymbol{e}_j)}{2h},$$

where $\boldsymbol{e}_j$ is the j-th unit vector and h is sufficiently small. This requires $2p$ evaluations of $\mathcal{L}$. With simultaneous perturbation (SP), introduced by Spall [7], all elements of $\boldsymbol{\theta}^k$ are randomly perturbed together. The two sided simultaneous perturbation gradient approximation (SPGA) is

$$\widehat{\nabla}_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}^k) = \frac{\mathcal{L}(\boldsymbol{\theta}^k + h\Delta) - \mathcal{L}(\boldsymbol{\theta}^k - h\Delta)}{2h} (\Delta_1^{-1}, \Delta_2^{-1}, \ldots, \Delta_p^{-1})^\top, \tag{4}$$

where $\Delta = (\Delta_1, \Delta_2, \ldots, \Delta_p)^\top$ is usually a random vector of independent Bernoulli random variables that take values -1 and 1 with probability 0.5, although other choices are possible. Two sided SPGA requires *two* evaluations of $\mathcal{L}$ regardless of the dimension p. FD approximation is superior to SP approximation as an estimator of the gradient. However, Spall [7] showed that when used in stochastic optimization setting they achieve the same level of statistical accuracy for a given number of iterations in terms of estimation of the optimum of the objective function. In the next section, we follow the same idea but in the MCMC setting.

4 Simultaneous perturbation gradient approximation based Metropolis adjusted Langevin Markov chain Monte Carlo (SPGA MALA)

SPGA MALA proposal is obtained by substituting the gradient in MALA proposal (2) with its SPGA, defined in (4):

$$\boldsymbol{\theta}^* = \boldsymbol{\theta}^k + \epsilon^2 \mathbf{M} \widehat{\nabla}_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}^k)/2 + \epsilon \sqrt{\mathbf{M}} \boldsymbol{z}^k. \tag{5}$$

In view of the MALA proposal density in (3), we require that the density of $\boldsymbol{\theta}^*$ given $\boldsymbol{\theta}^k$ and Δ is $q(\boldsymbol{\theta}^* | \boldsymbol{\theta}^k, \Delta) = \mathcal{N}(\boldsymbol{\theta}^* | \widehat{\boldsymbol{\mu}}(\boldsymbol{\theta}^k, \epsilon, \Delta), \epsilon^2 \mathbf{M})$, where $\widehat{\boldsymbol{\mu}}(\boldsymbol{\theta}^k, \epsilon, \Delta) = \boldsymbol{\theta}^k + \epsilon^2 \mathbf{M} \widehat{\nabla}_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}^k)/2$. Since Δ can take 2^p values with equal probability it follows that

the density of $\boldsymbol{\theta}^*$ given $\boldsymbol{\theta}^k$ is the mixture density $q(\boldsymbol{\theta}^*|\boldsymbol{\theta}^k) = \frac{1}{2^p}\sum_\Delta q(\boldsymbol{\theta}^*|\boldsymbol{\theta}^k, \Delta)$. The proposal mechanism (5) with proposal density q and standard accepetance probability $\alpha = \min\{1, p(\boldsymbol{\theta}^*)q(\boldsymbol{\theta}^k|\boldsymbol{\theta}^*)/p(\boldsymbol{\theta}^k)q(\boldsymbol{\theta}^*|\boldsymbol{\theta}^k)\}$ defines a valid Markov chain; it is simply Metropolis Hastings (MH) algorithm where the proposal is a mixture density q. However, evaluating α is intractable for large p. Because of this, instead of α we use

$$\alpha_\Delta = \min\{1, p(\boldsymbol{\theta}^*)q(\boldsymbol{\theta}^k|\boldsymbol{\theta}^*, \Delta)/p(\boldsymbol{\theta}^k)q(\boldsymbol{\theta}^*|\boldsymbol{\theta}^k, \Delta)\}.$$

In other words, instead of using q which involves calculation of each $q(\boldsymbol{\theta}^*|\boldsymbol{\theta}^k, \Delta)$ for 2^p possible values of Δ, we use the acceptance ratio which involves calculation of $q(\boldsymbol{\theta}^*|\boldsymbol{\theta}^k, \Delta)$ only for the drawn value of Δ.
This algorithm defines a valid Markov chain since it can be viewed as Metropolis-Hastings-Green(MHG) algorithm [2, p.41]. MHG algorithm allows *state-dependent mixing* or *random proposals* [1], meaning that on each step the proposal distribution need not be fixed but can belong to a countable family of proposal distributions. In our case it is a finite family $\{q(\boldsymbol{\theta}^*|\boldsymbol{\theta}^k, \Delta) : \Delta = (\Delta_1, \ldots, \Delta_p), \Delta_i \in \{-1, 1\}\}$. Using the random proposal instead of the mixture proposal comes with a price. As pointed out in the discussion section of the article [1], the random proposal method is less efficient because it accepts fewer proposals. This reduces efficiency of SPGA MALA. The second reason for reduced efficiency of SPGA MALA is that instead of the gradient its approximation is used. However, it is clear that SPGA MALA will be much faster than MALA for large scale systems.

5 Numerical results

In this section we compare the desribed algorithm to MALA on simulated data generated from the following models.
Fitz Hugh Nagumo (FHN) example. Fitz-Hugh Nagumo system [4] models the behaviour of spike potentials in the giant axon of squid neurons. It has the form

$$\begin{aligned} x_1'(t) &= \theta_3\{x_1(t) - x_1(t)^3/3 + x_2(t)\}, \\ x_2'(t) &= -\tfrac{1}{\theta_3}\{x_1(t) - \theta_1 + \theta_2 x_2(t)\}. \end{aligned}$$

We have used different notation than in [4], namely (x_1, x_2) for (V, R) and $(\theta_1, \theta_2, \theta_3)$ for (a, b, c). We set $\boldsymbol{\theta} = (0.2, 0.2, 3)$ and $\boldsymbol{\xi} = (-1, 1)$.
α- pinene example. The following model describes the thermal isomerization of α-pinene [8].

$$\begin{aligned} x_1'(t) &= -(\theta_1 + \theta_2)x_1(t), \\ x_2'(t) &= \theta_1 x_1(t), \\ x_3'(t) &= \theta_2 x_1(t) - (\theta_3 + \theta_4)x_3(t) + \theta_5 x_5(t), \\ x_4'(t) &= \theta_3 x_3(t), \\ x_5'(t) &= \theta_4 x_3(t) - \theta_5 x_5(t). \end{aligned}$$

The values of the parameters that we used are $\boldsymbol{\theta} = (0.1, 0.1, 0.3, 0.1, 0.3)$ and $\boldsymbol{\xi} = (1, 0, 0, 0, 0)$.
Hockin model. In [5], a model of the extrinsic blood coagulation is developed and consists of 34 differential equations and 42 rate constants. Due to lack of space

we do not present the model but refer the reader to the aforementioned reference. We fixed 32 parameters and estimated the remaining 10. The value of the selected parameter was set to $\boldsymbol{\theta} = (0.1, 0.4, 0.1, 0.32, 0.2, 1.05, 2.4, 6, 1.8, 8.2)$.

From each of the models presented above we generated 200 data points on the interval $[0, 20]$ and added Gaussian-distributed noise with standard deviation equal to 0.5. In SPGA (see (4)) we set $h = 10e{-}5$ while the gradient in MALA is obtained by solving sensitivity equations. Ideally, the tuning parameters in MALA should be chosen in such a way that acceptance rate is between 40% and 70%. As it was pointed out in Section 2, tuning of MALA is an issue. To simplify, for both MALA and SPGA MALA we set $\mathbf{M} = \mathbf{I}_p$, $\epsilon = 0.0002p^{-1/3}$ in all the simulations. This selection achieves the desired acceptance rate in FHN model and was used in [4]. For the other two models this is not the case. However, the most important thing here is to compare the performance of these two methods for the same selection of tuning parameters. For comparing sampling efficiency we followed approach used in [4]. A single Markov chain was initialized on the true mode and 5000 posterior samples were collected. The effective sample size (ESS) for each parameter was calculated; the minimum of ESS was used to calculate the time per effectively independent sample. For each method we ran 10 simulations, using the same data set. The methods were implemented in the interpreted language MATLAB and all computations were carried out on an Intel Core i5 computer with 1.3 GHz processor speed and 4 GB of memory. The results of our simulations are presented in Table 5.

The results of FHN example demonstrate loss in efficiency of SPGA MALA with respect to MALA; see Section 4 for the discussion. In α-pinene example MALA is still faster even though the sensitivity equations are of order 25. This is because the original system and the system of sensitivity equations are both linear. The example of Hockin model show the advantage of SPGA MALA. Sensitivity equations are of order 340 and this heavily affects the computatation time of MALA. On the other hand, the computation time of SPGA MALA is much smaller compared to that of MALA, making it much better in terms of the relative speed per effectively independent sample.

Acknowledgements: This research is supported by the Dutch Technology Foundation STW, which is part of the Netherlands Organisation for Scientific Research (NWO), and which is partly funded by Ministry of Economic Affairs. Mark Girolami and Ben Calderhead are acknowledged for making their MATLAB code used in [4] freely available. We thank Itai Dattner and Bart Bakker for useful comments.

References

[1] J. Besag, P. Green, D. Higdon, and K. Mengersen. Bayesian computation and stochastic systems. *Statistical science*, pages 3–41, 1995.

[2] S. Brooks, A. Gelman, G. Jones, and X.-L. Meng. *Handbook of Markov Chain Monte Carlo*. CRC press, 2011.

Model	Sampling method	Time (s)	Mean ESS ($\boldsymbol{\theta}$)	Total time /minimum mean ESS	Relative speed
$d = 2$ $p = 3$			$(\theta_1, \theta_2, \theta_3)$		
FHN	MALA	363.6	145, 30, 109	12.12	3.4
	SPGA MALA	623.2	84 , 15, 48	41.55	1
$d = 5$ $p = 5$			$(\theta_1, \theta_2, \theta_3, \theta_4, \theta_5)$		
α-Pinene	MALA	63.7	59, 58, 17, 11, 6	10.62	2.54
	SPGA MALA	134.8	187, 96, 6, 23 , 5	26.96	1
$d = 34$ $p = 10$			$(\theta_1, \theta_2, \theta_3, \theta_4, \theta_5,$ $\theta_6, \theta_7, \theta_8, \theta_9, \theta_{10})$		
Hockin	MALA	1.03e+04	5 6 8 7 7 7 6 5 6 8	2060	1
	SPGA MALA	180.5	7 6 8 8 7 6 6 5 4 7	45.13	45.65

Table 1: Summary of results for 10 runs of the model parameter sampling schemes for different models with 5000 posterior samples.

[3] A. R. Conn, K. Scheinberg, and L. N. Vicente. *Introduction to derivative-free optimization*, volume 8. Siam, 2009.

[4] M. Girolami and B. Calderhead. Riemann manifold langevin and hamiltonian monte carlo methods. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 73(2):123–214, 2011.

[5] M. F. Hockin, K. C. Jones, S. J. Everse, and K. G. Mann. A model for the stoichiometric regulation of blood coagulation. *Journal of Biological Chemistry*, 277(21):18322–18333, 2002.

[6] A. Raue, M. Schilling, J. Bachmann, A. Matteson, M. Schelke, D. Kaschek, S. Hug, C. Kreutz, B. D. Harms, F. J. Theis, et al. Lessons learned from quantitative dynamical modeling in systems biology. *PloS one*, 8(9):e74335, 2013.

[7] J. C. Spall. Multivariate stochastic approximation using a simultaneous perturbation gradient approximation. *Automatic Control, IEEE Transactions on*, 37(3):332–341, 1992.

[8] I. B. Tjoa and L. T. Biegler. Simultaneous solution and optimization strategies for parameter estimation of differential-algebraic equation systems. *Industrial & Engineering Chemistry Research*, 30(2):376–385, 1991.