A Comparative Analysis of MCMC Algorithms for Enhanced Bayesian Logistic and Cauchit Regression Modeling

1 Data Simulation

The study simulated a dataset of 150 observations and 10 covariates as a covariate matrix X with some parameters being the linear combinations of others to induce collinearity. True beta coefficients, β, were linearly spaced from -2 to 2 across the 10 parameters, establishing the underlying true values for simulation.

1.1 Binary Outcomes Y_i from Logistic and Cauchit Regression

For the logistic model, error terms ϵ_L were sampled using rejection sampling with the Cauchy distribution as the proposal. The optimal constant M was identified through numerical optimization:

$$M = \max_{x \in [-10,10]} \left(\frac{f_L(x)}{f_C(x)} \right),$$

where $f_L(x)$ is the logistic density and $f_C(x)$ is the Cauchy density. It achieved an optimal constant $M = 1.65$ and a 57% efficiency, indicating a good balance of acceptance rate and coverage of the target distribution's support. Binary outcomes Y_L were generated on the accepted samples.

For the Cauchit model, binary outcomes Y_C were derived from a latent variable Y_i^*, such that $Y_i = I(Y_i^* > 0)$, which is defined as $Y_i^* = x_i^\top \beta + \epsilon_i$, where $x_i^\top \beta$ is the linear predictor and ϵ_i is a random variable drawn from the Cauchy distribution. The binary outcome Y_C for each observation is then determined by the indicator function.

2 Model Specification

Two priors were considered for the initial belief of the parameter β: a standard Gaussian $\beta_j \sim \mathcal{N}(0, 1^2)$ reflecting weak prior beliefs, and a Unit Information Prior (UIP) $\beta \sim \mathcal{N}_d(\mathbf{0}, n(X^\top X)^{-1})$. We then constructed four posterior models: Logistic Regression & Gaussian Prior, Logistic Regression & UIP Prior, Cauchit Regression & Gaussian Prior, and Cauchit Regression & UIP Prior. For detailed formulas of these models, see Appendix A.

3 The Metropolis-Hastings Algorithm

The Metropolis-Hastings (M-H) algorithms play a crucial role in approximating posterior distributions in Bayesian analysis. This section outlines its enhancements for increased efficiency.

3.1 Random Walk Metropolis (RWM)

The Random Walk Metropolis (RWM) algorithm, one of the most common M-H algorithms, operates by generating proposed moves from the current state and accepting these moves based on an optimal acceptance probability. Specifically, the algorithm proposes a new point Y by adding a normally distributed random increment to the current point X_i, with acceptance $\alpha = \min\left(1, \frac{\pi(Y)}{\pi(X_{i-1})}\right)$. If the move is rejected, the algorithm remains at X_i. This process is repeated for each iteration, effectively sampling from the target distribution π.

3.2 Pre-conditioned, Component-wise, and MALA Algorithms

The RWM algorithm is the basis for several advanced methods that enhance sampling:

1. **Pre-conditioned RWM**: Scales the vanilla RWM proposal distribution using the estimated covariance matrix from preliminary runs, improving convergence.

2. **Component-wise RWM**: Updates one parameter at a time, which is beneficial in high-dimensional spaces and can improve acceptance rates.

3. **Metropolis-adjusted Langevin Algorithm (MALA)**: Integrates gradient information into the proposal process, promoting more informed convergence by guiding proposals towards correct regions.

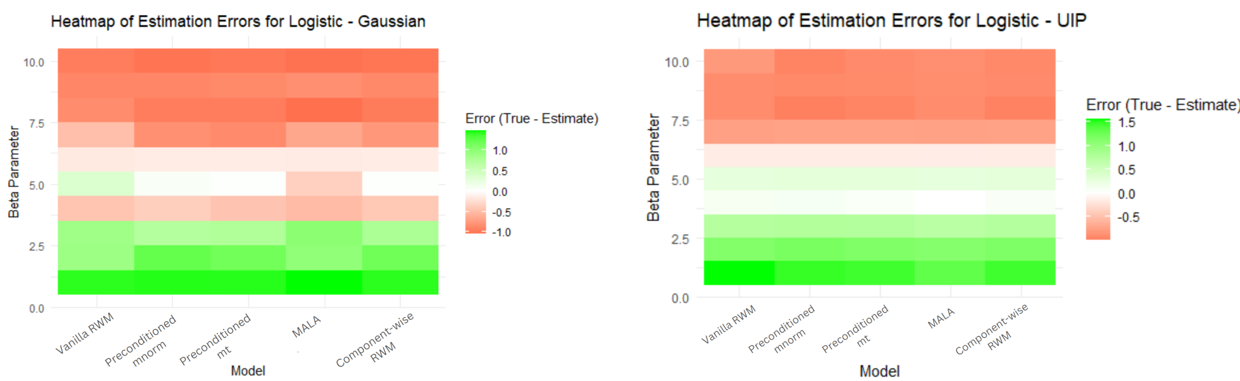
4 Performance Evaluation Metrics

We evaluated the performance of five M-H algorithms (including multivariate normal and t proposals for Pre-conditioned RWM; burn-in was applied to MALA) applied to logistic and cauchit regression models.

4.1 Logistic Regression Model Diagnostics

4.1.1 Prediction Accuracy: Brier Scores & Posterior Mean

Prediction accuracy was evaluated by comparing the posterior means to the true beta coefficients. We visualized residuals using heatmaps to quickly identify estimation errors: Logistic-Gaussian



(a) (b)

Figure 1: Heatmaps of Estimation Errors for Gaussian & UIP Prior in Logistic Regression

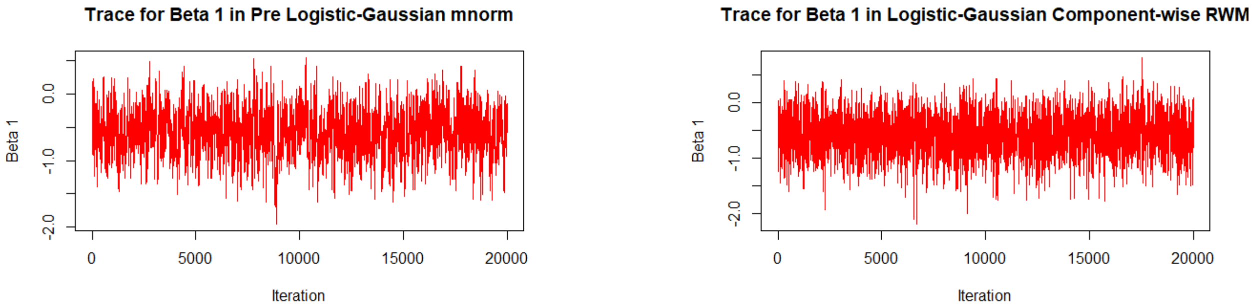
model (Figure1a) suggests greater variability across the different MCMC algorithms, with MALA showing slightly more varied error, suggesting it may be more sensitive to the specific parameters. Logistic-UIP model (Figure1b) shows generally very consistent performance across MCMCs.

The mean Brier scores were 0.04446 for Logistic-Gaussian and 0.04705 for Logistic-UIP, indicating slightly better predictive accuracy with the Gaussian prior.

4.1.2 Convergence Diagnostics: Trace Plots & Gelman-Rubin Statistics

We assessed convergence using Gelman-Rubin statistics, which compare between-chain and within-chain variances, while values close to 1.1 suggest effective convergence. Our analysis found that the Component-wise RWM algorithm achieved convergence closest to the ideal, with G-R statistics of 1.02 for the Gaussian prior and 1.00 for the UIP prior. These results were the lowest among the tested MCMC algorithms, signifying strong convergence. Detailed Gelman-Rubin statistics for all models are provided in Appendix B for reference.

We then selected two MCMCs with the lowest Gelman-Rubin statistics from each prior for trace plot analysis, focusing on Beta 1 here due to its extremity in the parameter range, making it easily assessable.



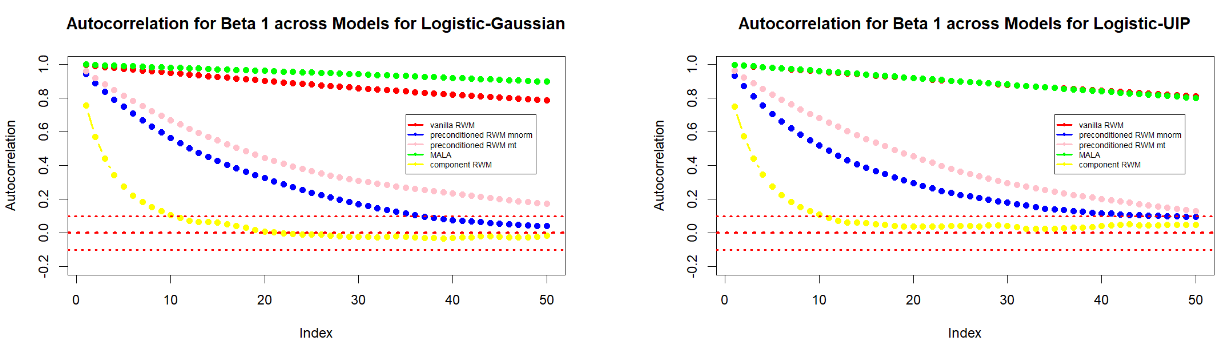
(a) (b)

Figure 2: Trace plots of selected models for Logistic-Gaussian Indicating Convergence

For Logistic-Gaussian, Figure2a showed consistent oscillations suggesting good mixing, but also indicates sharp fluctuations pointing to high autocorrelation. In contrast, Figure2b showed less variance supported by the narrower band and potentially better precision, indicating better convergence towards the correct region of the true parameter. For Logistic-UIP, we found a similar pattern where the Component-wise RWM reported a stationary range of fluctuation around the true value, suggesting a better mixing and a degree of independence.

Comparing two priors both on the Component-wise RWM, we found in the trace plots across all parameters that Gaussian priors provided more consistent convergence in most of the beta.

4.1.3 Sampling Efficiency: Auto-correlation Plots & Effect Sample Size



(a) (b)

Figure 3: Autocorrelation Plots for Logistic-Gaussian & Logistic-UIP

The Component-wise RWM has shown the fastest decay to 0 in auto-correlation in both Gaussian (3a) and UIP (3b) Prior, indicating high sampling efficiency. Heatmaps of auto-correlation across all beta parameters (see Appendix C) showed that for Gaussian prior, Component-wise RWM outperformed the pre-conditioned *mnorm* in every parameter except Beta $2, 3, 6, 7$; for UIP prior, Component-wise RWM outperformed the other one in every parameter except Beta 5 and 7.

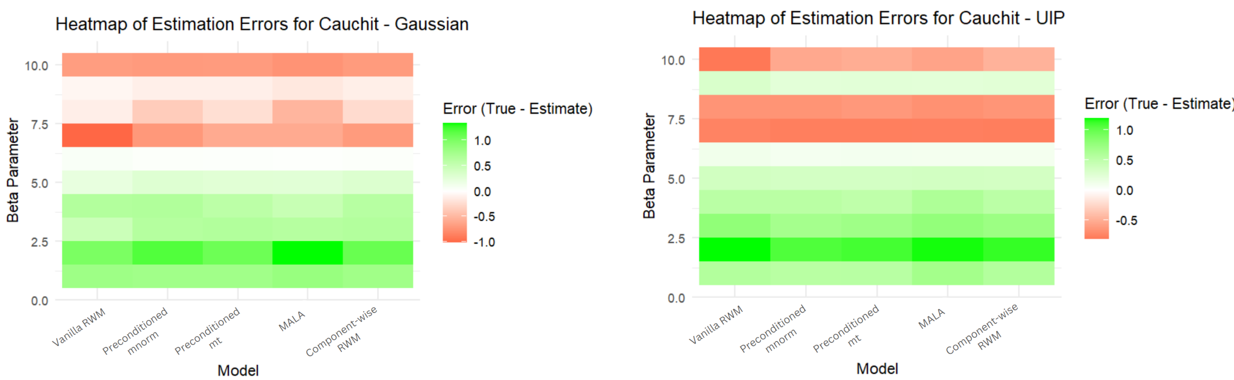
We further calculated the mean effective sample size (ESS) for the Component-wise RWM given the subtle difference in their auto-correlation plots. Gaussian prior ($ESS = 1188.01$) was found to achieve a slightly higher mean ESS than UIP ($ESS = 932.92$), suggesting it was more efficient in sampling independent draws.

4.2 Cauchit Regression Model Diagnostics

4.2.1 Prediction Accuracy: Brier Scores & Posterior Mean

Starting with the heatmaps of residuals for Cauchit-Gaussian (Figure4a), the pre-conditioned RWM with multivariate t proposal demonstrated particularly strong performance. For Cauchit - UIP (Figure4b), the performance across MCMCs was more consistent. Estimation errors seemed to be larger in UIP prior, as evidenced by much stronger red shades indicating.

This was supported by the slightly larger mean Brier score for UIP (0.0381) compared with Gaussian (0.0353), indicating that Gaussian prior offered a better prediction accuracy here.

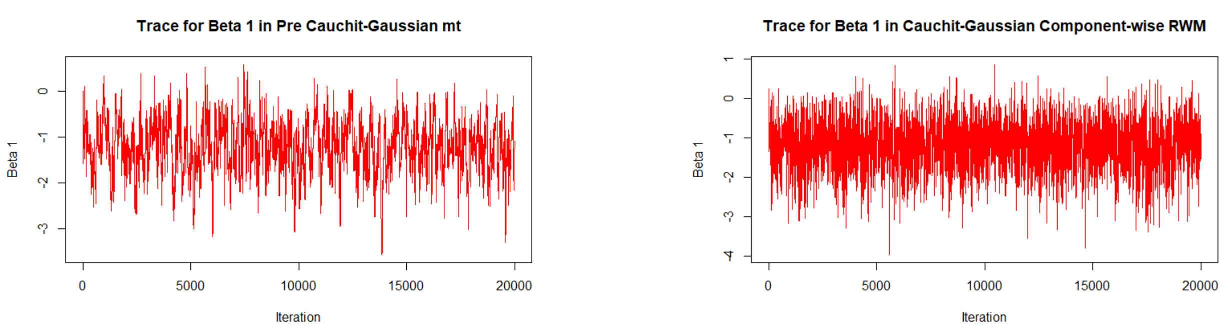


(a) (b)

Figure 4: Heatmaps of Estimation Errors for Gaussian & UIP Prior in Cauchit Regression

4.2.2 Convergence Diagnostics: Trace Plots & Gelman-Rubin statistics

Similarly, convergence for the Cauchit model was evaluated using Gelman-Rubin statistics (see Appendix B). The Component-wise RWM algorithm once again showed strong convergence, with Gelman-Rubin values of 1.00 for the Gaussian prior and 1.03 for the UIP prior, indicating it was closest to the desired value. We then selected Component-wise RWM and the algorithm with



(a) (b)

Figure 5: Trace plots of selected models for Cauchit-Gaussian Indicating Convergence

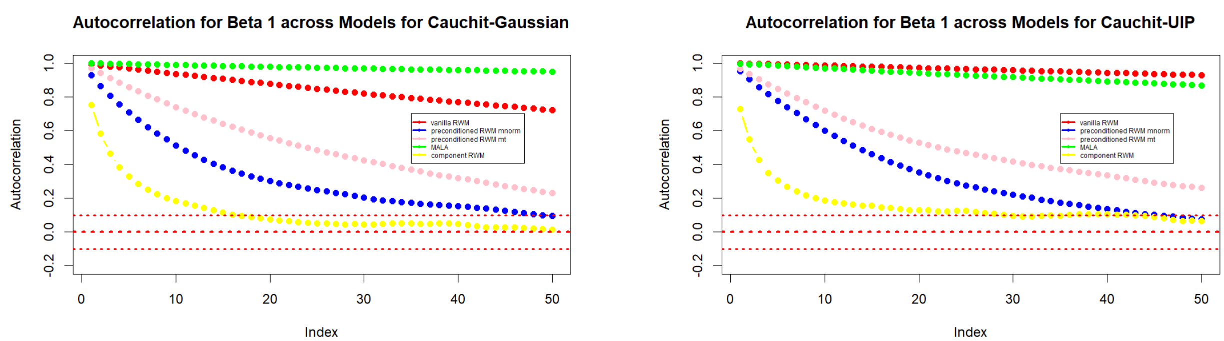
the next lowest G-R statistic, Preconditioned *mt*, for both priors. For Cauchit-Gaussian, although vanilla RWM gave the second-lowest G-R statistic, its trace plot suggested an obvious pattern for random walk rather than convergence, thus wasn't selected considering we wanted a more informative assessment.

For Cauchit-Gaussian, Figure5a exhibited a slightly larger variance in oscillation. Conversely, Figure5b oscillated more around the true value with a consistent variance, so it provided a more reliable convergence behavior. For Cauchit-UIP, the Pre-conditioned *mt* trace plot gave a higher degree of fluctuation and worse mixing, while the Component-wise RWM showed more stationarity with good convergence behavior.

Contrasting the two priors both on Component-wise RWM, by checking across all parameters, we found that Gaussian prior showed better mixing in the majority of the beta.

4.2.3 Sampling Efficiency: Auto-correlation Plots & Effect Sample Size

The Component-wise RWM demonstrated rapid auto-correlation decay for both Gaussian and UIP priors, suggesting efficient sampling. Checking the heatmaps of autocorrelation across all beta parameters (see Appendix C), we could see that for Gaussian, Component-wise approached zero quicker than Pre-conditioned RWM *mnorm* in almost all parameters; for Cauchit-UIP, however, it only outperformed Pre-conditioned RWM *mnorm* in Beta 1, 4, and 8.



(a) (b)

Figure 6: Autocorrelation Plots for Cauchit-Gaussian & Cauchit-UIP

The heatmaps showed that using the same Component-wise RWM algorithm, the auto-correlation performance in Gaussian prior much outperformed the counterpart in UIP prior. This was further supported by the mean ESS, where the Gaussian prior ($ESS = 1448.125$) has a much higher ESS value than UIP prior ($ESS = 630.473$), indicating more efficient sampling from independent draws.

5 Model Comparison and Conclusion

Performance diagnostics indicated that the **Component-wise Random Walk Metropolis** is the most efficient MCMC algorithm for logistic and cauchit regressions across all priors. The effectiveness of this algorithm can be due to its strategy of updating one parameter at a time, which reduces the complexities associated with high-dimensional parameter spaces.

Using this MCMC algorithm, the **Gaussian prior** edged out the UIP prior with better convergence, sampling efficiency, and prediction accuracy in both logistic and cauchit regression, suggesting its assumption of normality is well-suited to the structure of our synthetic data.

We also acknowledge limitations, such as the potential influence of collinearity on parameter stability and the unexplored benefits of advanced MCMC methods like pre-conditioned MALA, which may offer further improvements in future studies.

A APPENDIX A: Posterior Distribution Formulas

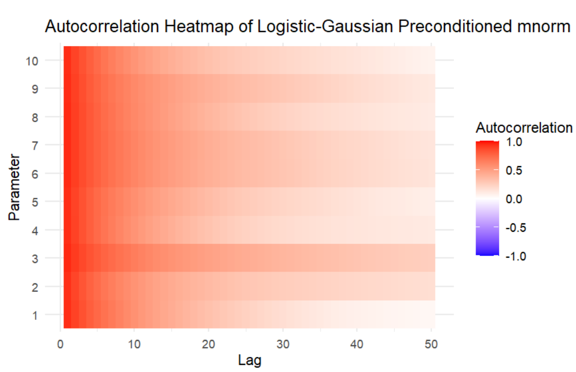
Posterior Distribution Formula
Logistic Regression with Gaussian Prior $\pi(\beta\|X,Y) \propto \left(\prod_{i=1}^n p_i^{y_i}(1-p_i)^{1-y_i}\right)\left(\prod_{j=1}^d \phi(\beta_j)\right)$, where $p_i = \frac{1}{1+e^{-x_i^T\beta}}$ and $\phi(\cdot)$ denotes the standard normal density function.
Logistic Regression with UIP Prior $\pi(\beta\|X,Y) \propto \left(\prod_{i=1}^n p_i^{y_i}(1-p_i)^{1-y_i}\right)\mathcal{N}_d\left(0, n(X^TX)^{-1}\right)$, with the same p_i as in the logistic model.
Cauchit Regression with Gaussian Prior $\pi(\beta\|X,Y) \propto \left(\prod_{i=1}^n q_i^{y_i}(1-q_i)^{1-y_i}\right)\left(\prod_{j=1}^d \phi(\beta_j)\right)$, where $q_i = \frac{1}{2} + \frac{1}{\pi}\arctan(x_i^T\beta)$.
Cauchit Regression with UIP Prior $\pi(\beta\|X,Y) \propto \left(\prod_{i=1}^n q_i^{y_i}(1-q_i)^{1-y_i}\right)\mathcal{N}_d\left(0, n(X^TX)^{-1}\right)$, using the same q_i as in the Cauchit model.

B APPENDIX B: Gelman-Rubin Statistics

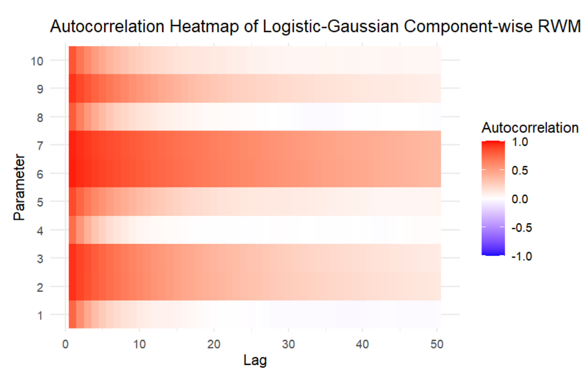
Logistic Model	**Gelman-Rubin (Gaussian)**	**Gelman-Rubin (UIP)**
Vanilla RWM	1.98	1.03
Preconditioned mnorm	1.78	1.05
Preconditioned mt	3.22	1.09
MALA	2.11	10.2
Component-wise RWM	1.02	1.00

CacModel	**Gelman-Rubin (Gaussian)**	**Gelman-Rubin (UIP)**
Vanilla RWM	1.42	1.99
Preconditioned mnorm	3.27	1.03
Preconditioned mt	1.53	1.02
MALA	1.67	1.41
Component-wise RWM	1.00	1.03

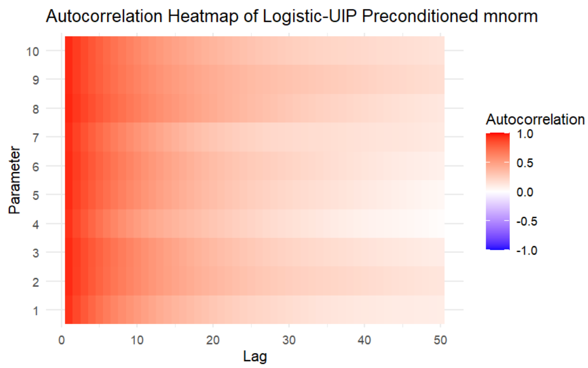
C APPENDIX C: Heatmaps for Autocorrelation across parameters



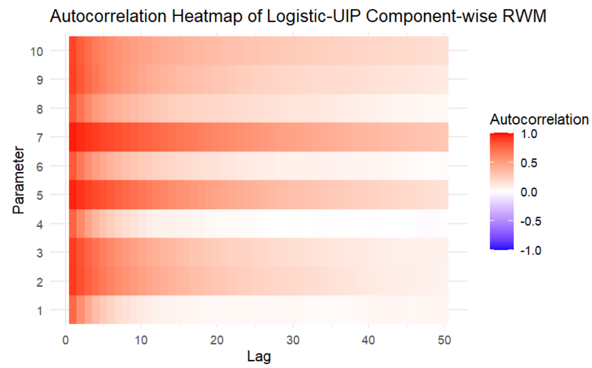
(a)



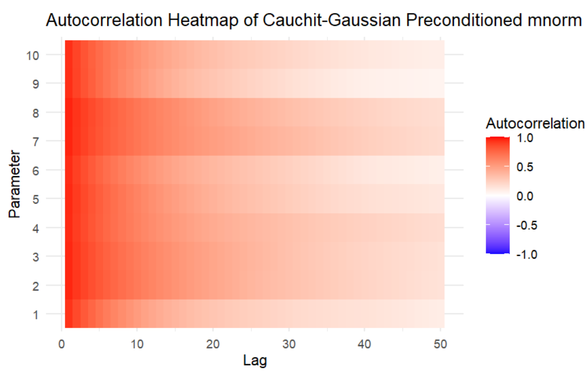
(b)


Autocorrelation Heatmap of Logistic-UIP Preconditioned mnorm
Parameter
Lag
Autocorrelation
1.0
0.5
0.0
-0.5
-1.0

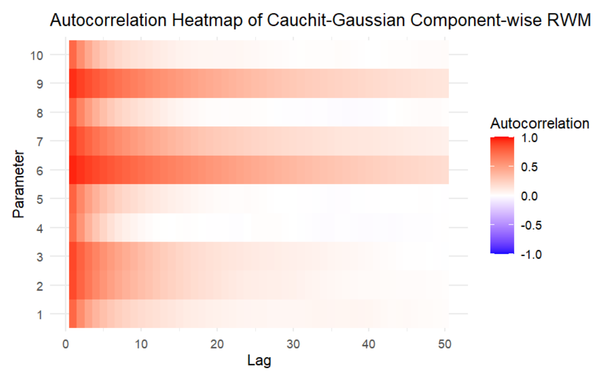
(a)


Autocorrelation Heatmap of Logistic-UIP Component-wise RWM
Parameter
Lag
Autocorrelation
1.0
0.5
0.0
-0.5
-1.0

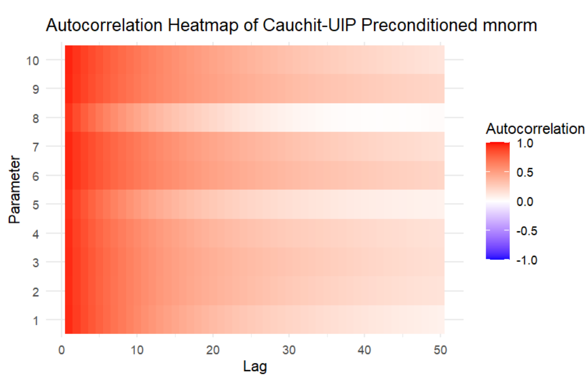
(b)


Autocorrelation Heatmap of Cauchit-Gaussian Preconditioned mnorm
Parameter
Lag
Autocorrelation
1.0
0.5
0.0
-0.5
-1.0

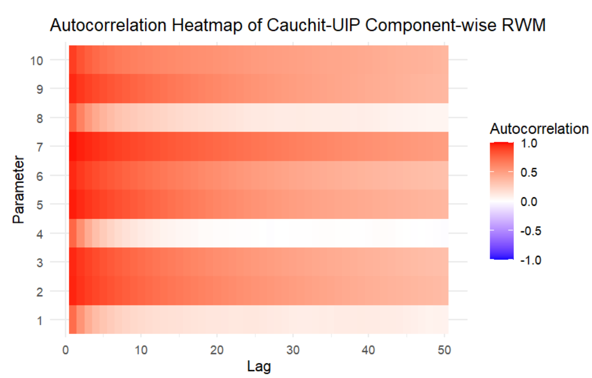
(a)


Autocorrelation Heatmap of Cauchit-Gaussian Component-wise RWM
Parameter
Lag
Autocorrelation
1.0
0.5
0.0
-0.5
-1.0

(b)


Autocorrelation Heatmap of Cauchit-UIP Preconditioned mnorm
Parameter
Lag
Autocorrelation
1.0
0.5
0.0
-0.5
-1.0

(a)


Autocorrelation Heatmap of Cauchit-UIP Component-wise RWM
Parameter
Lag
Autocorrelation
1.0
0.5
0.0
-0.5
-1.0

(b)