

Question 1

Subsection (a)

(Part I)

We consider the same elliptic partial differential equation defined in the spatial domain between 0 and 1. Here, $k(x, \omega)$ is unknown and is the subject of Bayesian inference.

$$\frac{\partial}{\partial x} \left[k(x, \omega) \frac{\partial u(x, \omega)}{\partial x} \right] = -s(x) \quad (1)$$

The boundary conditions and source terms are described by Eq. 2,3 and 4.

$$u(x = 1, \omega) = u_r \quad (2)$$

$$k(x, \omega) \frac{\partial u}{\partial x} \Big|_{x=0} = -F \quad (3)$$

$$s(x) = \sum_{i=1}^M \frac{\theta}{\delta \sqrt{2\pi}} \exp \left[-\frac{(x - m_i)^2}{2\delta^2} \right] \quad (4)$$

where $F = -1.0$, $u_r = 1$, $M = 4$, $\delta = 0.05$, $\theta = 0.8$ and $\mathbf{m} = [0.2, 0.4, 0.6, 0.8]$.

We are interested to infer $k(x)$ from noisy observations of u at N distinct spatial locations x_i . The inverse problem can be mathematically formulated as:

$$y_i = u(x_i) + \epsilon_i \quad (5)$$

where ϵ_i are the observation errors that is independently and identically distributed based on a Gaussian distribution with zero mean and σ_ϵ^2 variance. ϵ_i is also independent of $k(x)$.

In this part, we assume that σ_ϵ^2 is a deterministic value, and is equal to 10^{-4} . To use Markov Chain Monte Carlo (MCMC) methods, we apply Bayes rule to evaluate the posterior density of $k(x)$, using the prior density and the likelihood function. The likelihood function is obtained based on actual observations between x and $u(x)$, denoted by D , which was provided in the question.

$$\pi[k(x)|D] \propto \pi[k(x)] * \pi[D|k(x)] \quad (6)$$

Given that $Y(x) \equiv \ln(k(x))$ is given as a Gaussian process, we can model $Y(x)$ with a finite Karhunen-Loeve expansion:

$$Y_d(x, \omega) = \mu_Y(x) + \sum_{i=1}^d \sqrt{\lambda_i} \psi_i(x) Z_i(\omega) \quad (7)$$

where Z_i are standard Gaussian random variables. Consequently, $Y_d \sim N(\mu_y, \sigma_y^2)$, where $\sigma_y^2 = \sum_{i=1}^d \lambda_i \psi_i(x)^2$. The eigenvalues and eigenvectors are generated using the Nystrom method, as described in the previous project submission.

Further, we leverage the bijection between $k(x)$ and \mathbf{Z} and rewrite Eq. 6 as:

$$\pi[\mathbf{Z}(x)|D] \propto \pi[\mathbf{Z}] * \pi[D|\mathbf{Z}] \quad (8)$$

Given that all the components of \mathbf{Z} are independent of each other, we can write the prior distribution as $\pi(\mathbf{z}) \sim N(0, \mathbf{I}_d)$.

To derive the likelihood distribution, we look at Eq. 5. Applying conditional probabilities: $y_i|\mathbf{Z} = u(x_i)|\mathbf{Z} + \epsilon_i$, where $u(x_i)|\mathbf{Z}$ is now non-random. Therefore, the statistical distribution of $y_i|\mathbf{Z}$ is governed by the distribution of ϵ_i . Therefore, we can write the likelihood distribution as $y_i|\mathbf{Z} \sim N(u(x_i)|\mathbf{Z}, \sigma_\epsilon^2)$. Then, $\pi[D|\mathbf{Z}]$ is given by:

$$\pi[D|\mathbf{Z}] = \prod_{i=1}^d f_{y_i|\mathbf{Z}} \quad (9)$$

$$= \prod_{i=1}^d \frac{1}{\sqrt{2\pi\sigma_\epsilon}} \exp\left[-\frac{1}{2\sigma_\epsilon^2}(y_i - u(x_i)|\mathbf{Z})^2\right] \quad (10)$$

$$\sim N(\mathbf{y}; \text{mean} = u(x_i)|\mathbf{Z}, \sigma^2 = \sigma_\epsilon^2 \mathbf{I}_n) \quad (11)$$

We use Eq. 8, allowing the posterior distribution to converge to the true distribution of $\pi(\mathbf{z})$ with the adaptive MCMC algorithm (AM) by Haario et al. [1]. In the AM algorithm, unlike conventional MCMC, the proposal considers the values of all prior steps, making it a non-Markovian process in theory. Following the implementation of the original authors, the joint distribution of the proposal is given by a multivariate normal distribution centered at the previous step:

$$q(\mathbf{l}|\mathbf{Z}^{(0)}, \dots, \mathbf{Z}^{(t-1)}) = N(\mathbf{l}; \mathbf{Z}^{(t-1)}, \mathbf{C}_t). \quad (12)$$

$$\mathbf{C}_t = S_d \text{Cov}\{\mathbf{Z}^{(0)}, \dots, \mathbf{Z}^{(t-1)}\} + S_d \epsilon \mathbf{I}_d. \quad (13)$$

$$S_d = \frac{2.4^2}{d} \quad (14)$$

We run the algorithm with $d = 10$, and at least 1500 Monte Carlo steps. To mitigate transient effects, we eliminate the samples from the first 1000 steps, and only evaluate the statistical moments based on the last 500 steps. We coin this as the "burn-in effect".

The acceptance sequence follows that of the conventional Metropolis-Hasting algorithm. A proposal \mathbf{l} is drawn from Eq. 12. Then, the acceptance ratio is calculated based on:

$$\alpha(\mathbf{l}, \mathbf{Z}^{(0)}, \dots, \mathbf{Z}^{(t-1)}) = \min\left\{1, \frac{\pi(\mathbf{l}|D) q(\mathbf{Z}^{(0)}, \dots, \mathbf{Z}^{(t-1)}|\mathbf{l})}{\pi(\mathbf{Z}^{(t-1)}|D) q(\mathbf{l}|\mathbf{Z}^{(0)}, \dots, \mathbf{Z}^{(t-1)})}\right\} \quad (15)$$

To ascertain the extent of chain mixing, the trace plot of three components for the last 500 Monte Carlo steps are plotted in Fig. 1. The acceptance probability in the last 500 steps is approximately 15 %, which corresponds to the number of movements in the \mathbf{Z} values of the three plotted components. In general, we see that the Monte Carlo algorithm is moving and mixing decently, generating the required samples from the target posterior distribution.

Similarly, the autocorrelation of the 10 components as a function of lag is plotted in Fig. 2. In general, we see that the autocorrelation for each of the components decay quickly with the lag, and hovers between the values of ± 0.3 within 300 steps. We also notice that the autocorrelation increased in magnitude when the lag tends towards its largest value for Z_5 and Z_3 . Nevertheless, the autocorrelation of the other 8 components remain within the ± 0.3 range.

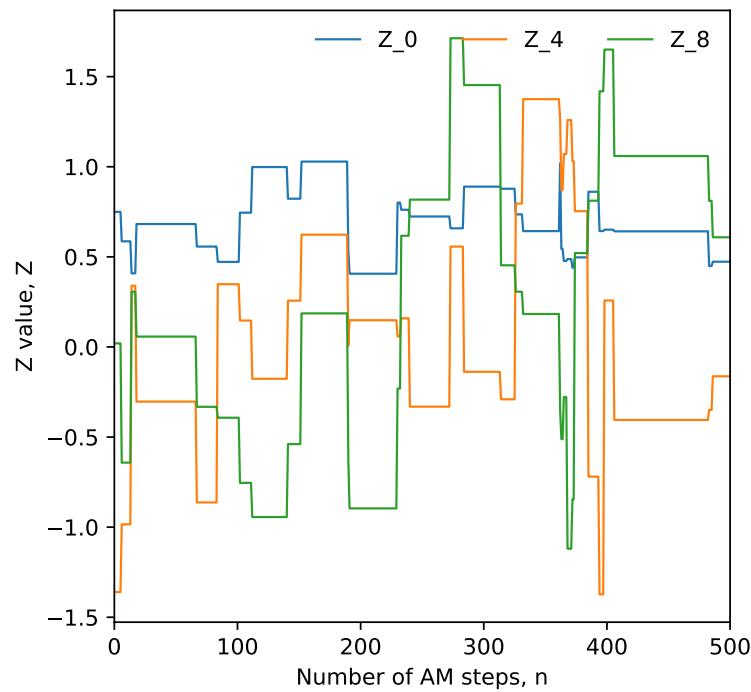


Figure 1: Trace plot for selected components of the MCMC chains.

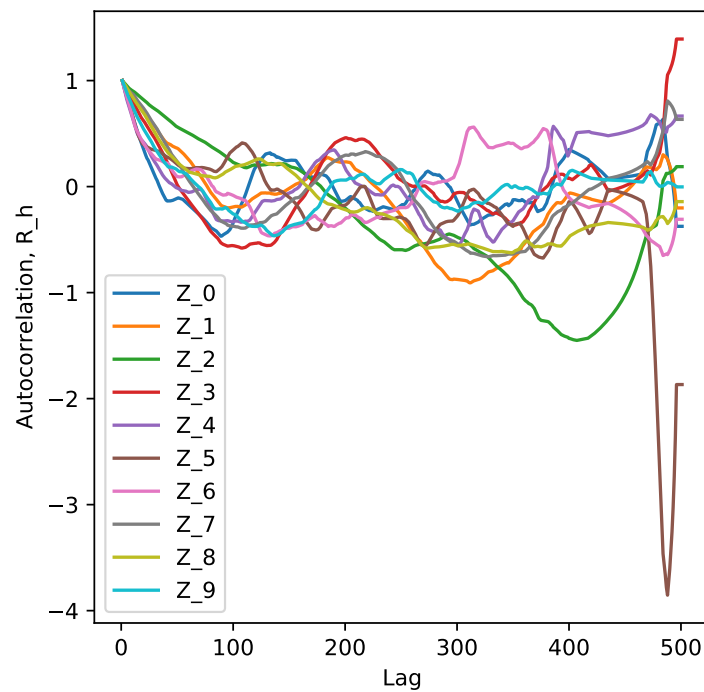


Figure 2: Autocorrelation coefficient of the 10 stochastic elements (variables of KL expansion) as a function of lag.

(Part II)

Here, following Monte Carlo simulations with the adaptive Metropolis algorithm, we are interested to visualize the posterior densities of the 10 stochastic parameters used to describe $Y(x)$. The ‘*Scattermatrix*’ diagram plotted using the Pandas package with Python is depicted in Fig. 3. The posterior probability densities of the ten coefficients of the KL expansion are described along the diagonals, while the off-diagonal plots illustrate the extent of correlation between the individual components. From inspection, there does not appear to be a very obvious trend. However, the posterior densities indicate that the coefficients either are centered at one particular value (the mode being much higher than the other values), or are more evenly distributed across two or three modes.

Based on the off-diagonal plots, there appears to be some correlation between the ten components. We see this from the way the points are distributed. For some components, such as between components 0 and 1, one component appears to increase while the other decrease, showing a negative correlation. Likewise, between components 1 and 7, a positive correlation trend is observed. The numerical results suggest that the AM algorithm inherently couples the inferred components, leading to the correlations observed in the scatter plots.

(Part III)

The plot of the marginal posterior mean and standard deviation following AM Monte Carlo simulations is depicted in Fig. 4. In the figure, the red lines indicate the true values from the provided dataset, and the blue line and markers indicate the expected value from AM Monte Carlo simulations. The dotted lines indicate the confidence intervals based on one standard deviation. To investigate if the posterior is stationary, we observe if the correlation between different pairs of x coordinates is independent of their respective vector positions, but is dependent on only their distances apart. This means that the correlation between the different components should be fairly similar. A stationary posterior will likely have off-diagonal plots that look fairly similar to each other. From inspection, we can see that the off-diagonal plots are scattered rather differently, indicating that the process is likely not stationary.

Further, we also see that the AM algorithm is very effective in inferring the log permeability field from the provided data. The posterior densities are able to reproduce the high density regions near $x = 0$ and $x = 0.8$, and the low density region near $x = 0.6$. Considering the confidence interval of one standard deviation, majority of the true values fall within this interval region, enhancing our confidence of the solution. Unfortunately, with 10 coordinate points, the AM algorithm is unable to fully replicate all the local maximums and minimums in the observed data. For instance, the posterior is missing the local maximum we see at $x = 0.8$. Instead, the posterior has a maximum value at $x = 0.9$, between the two local maximum points of $x = 0.8$ and $x = 0.95$ that is observed in the true values. Therefore, as a consequence of the lower resolution of the posterior, the local maximums and minimums are blended and spatially averaged.

(Part IV)

The covariance fields of the log permeability of the posterior and prior densities are depicted in Fig. 5 and 6, respectively. The x and y axes indicate the x -coordinate of the domain.

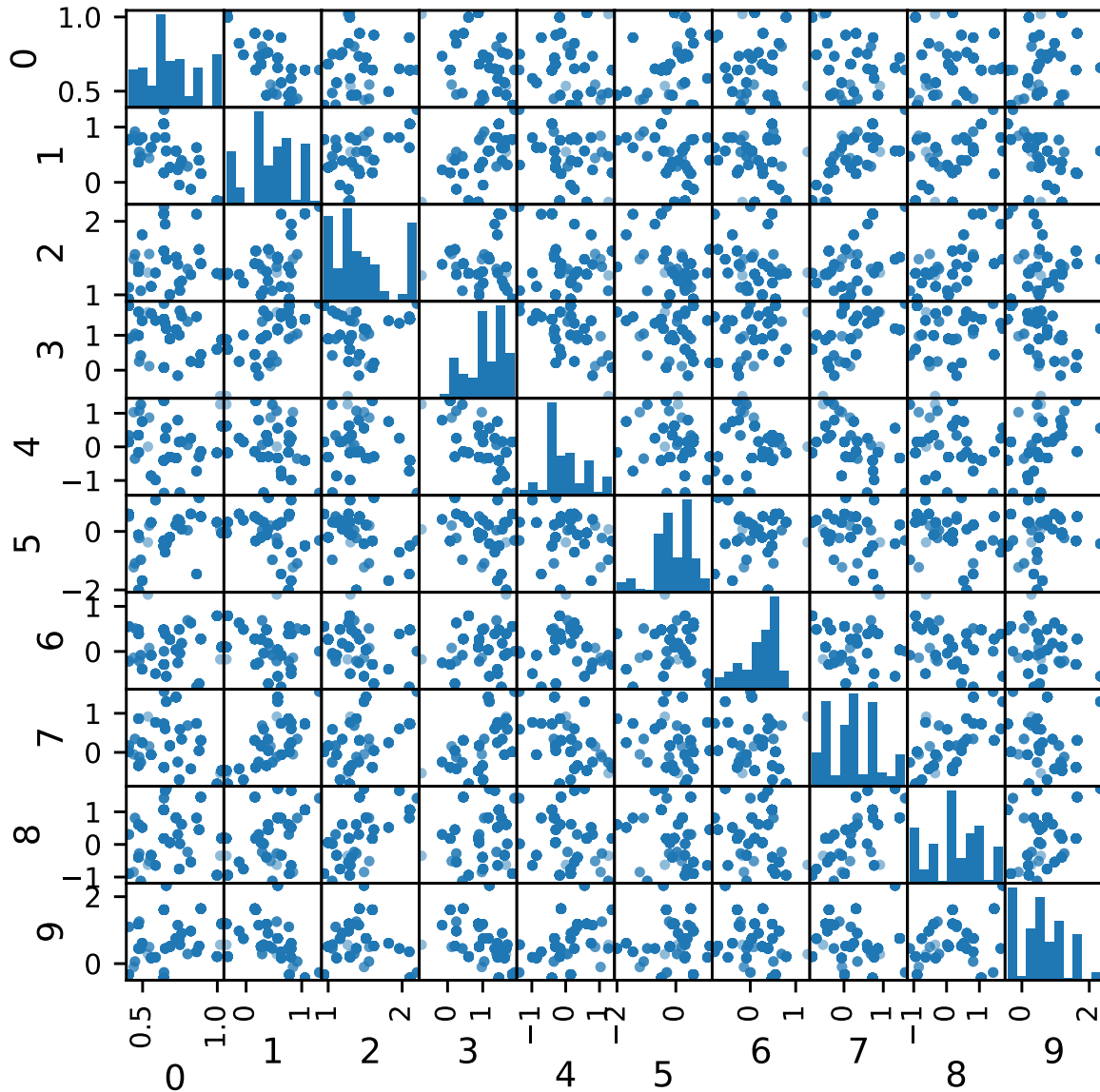


Figure 3: ‘Scattermatrix’ diagram of the 10 stochastic components. The posterior densities of the 10 respective components are depicted along the diagonal, while the off-diagonal plots illustrate the correlation between the respective components.

Based on the prior in Fig. 6, we see that the covariance is highest along the main diagonal of the covariance field. The covariance between the coordinates decay with distance, as observed by the reducing covariance values as you tend away from the main diagonal. This is mathematically coherent with the model inputs, as the covariance kernel adopted for the stationary Gaussian process prior decays exponentially with the distance.

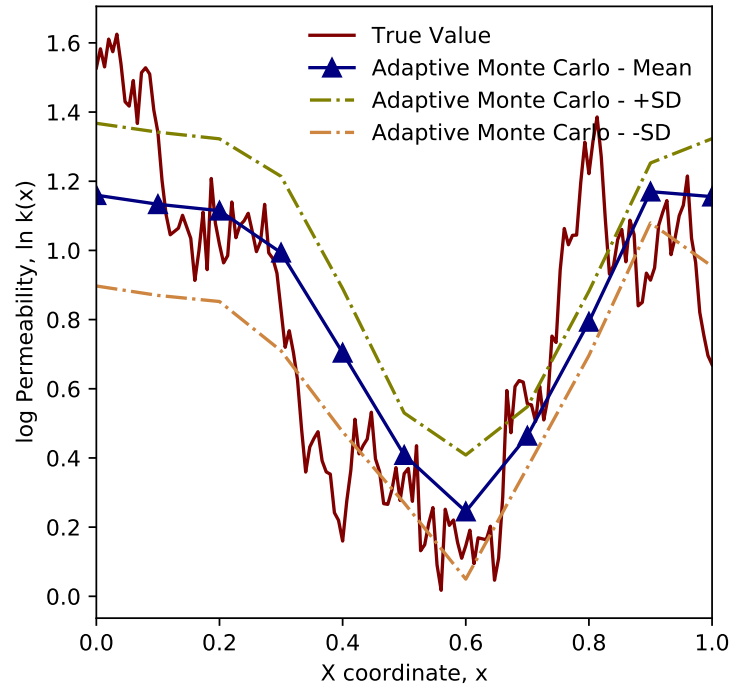


Figure 4: Plot of the marginal posterior mean and standard deviation of the log permeability as a function of the x coordinate, as compared to the true values from the provided dataset.

We now analyze the posterior covariance field. From inspection, in Fig. 5, similar to the prior covariance field, we still observe the strongest correlation to be along the main diagonal of the posterior covariance field. However, unlike the prior covariance field, the absolute magnitude of the covariance (effectively variance of the distributions of the respective x coordinates) has greatly reduced. Therefore, the data suggests that AM Monte Carlo algorithm is effective in reducing the variance of the posterior with the provided data, and the statistical inference approach is largely effective.

In the same plots, we also notice some negative correlations off the main diagonals. It is unclear to us if this is a result of numerical noise, given that the resolution of the entire x-domain is only 10 points. Nevertheless, we see that the correlation between the x coordinates decay with distance, similar to the prior. The rate of decay, on a percentage basis, is slower for the posterior field than the prior.

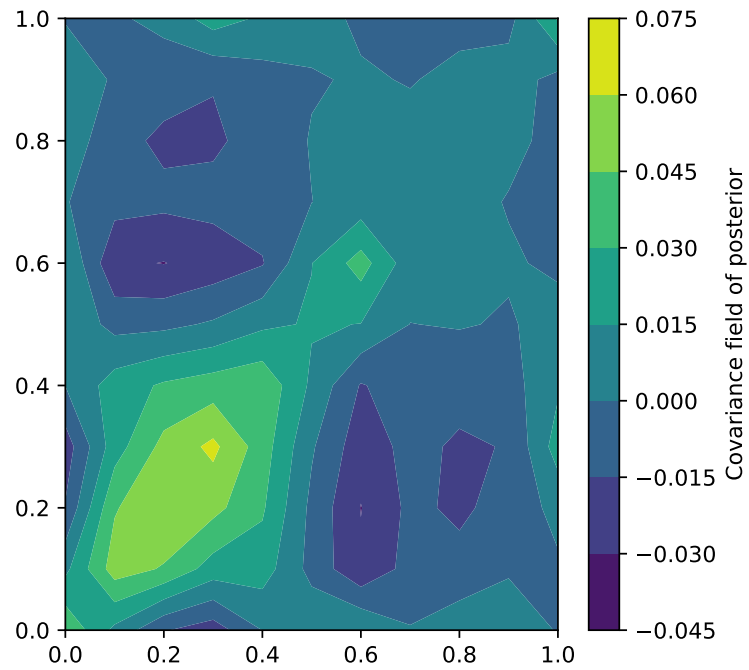


Figure 5: Plot of the posterior covariance field of the log permeability as a function of the spatial coordinate.

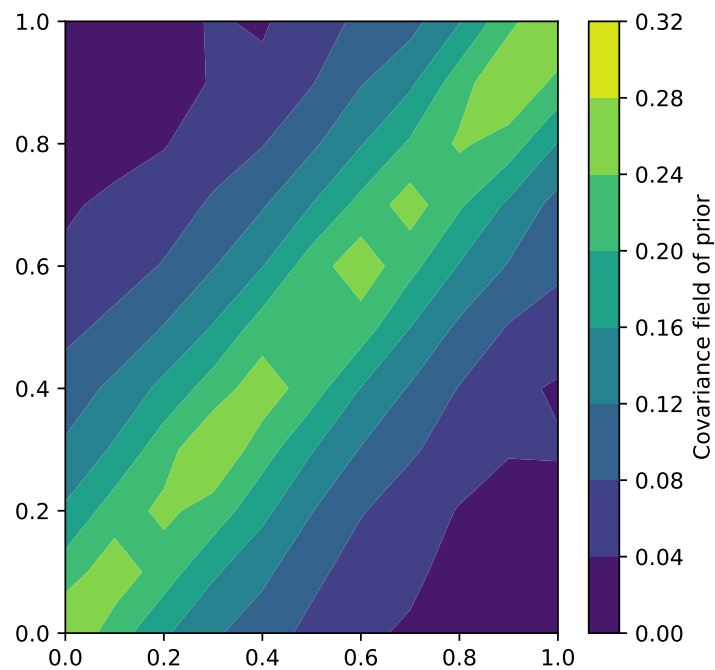


Figure 6: Plot of the prior covariance field of the log permeability as a function of the spatial coordinate.

Subsection (b)

Unlike the previous section, we do not assume that the variance of σ_ϵ^2 is known. Therefore, we have to endow it with a hyperprior and jointly infer both σ_ϵ^2 and $k(x)$ from the provided data using adaptive MC.

To allow for facile statistical inference, let us endow the hyperprior of σ_ϵ^2 as a Gamma distribution.

$$\frac{1}{\sigma_\epsilon^2} \sim \Gamma(\beta, \gamma) \quad (16)$$

Next, we revise the posterior density as:

$$\pi \left[\mathbf{Z}, \frac{1}{\sigma_\epsilon^2} | D \right] \propto \pi \left[D | \mathbf{Z}, \frac{1}{\sigma_\epsilon^2} \right] \pi \left[\mathbf{Z}, \frac{1}{\sigma_\epsilon^2} \right] \quad (17)$$

$$\propto \pi \left[D | \mathbf{Z}, \frac{1}{\sigma_\epsilon^2} \right] \pi [\mathbf{Z}] \pi \left[\frac{1}{\sigma_\epsilon^2} \right] \quad (18)$$

$$\propto \left(\frac{1}{\sigma_\epsilon^2} \right)^{\beta + N/2 - 1} \exp \{ -(\gamma + \|\mathbf{u} - \mathbf{y}\|^2 / (2\sigma_\epsilon^2)) \} \exp \{ -\|\mathbf{Z}\|^2 \} \quad (19)$$

Therefore, the posterior distribution of σ is given by:

$$\frac{1}{\sigma_\epsilon^2} | \mathbf{Z}, D \sim \Gamma \left(\beta + \frac{N}{2}, \gamma + \frac{\|\mathbf{u} - \mathbf{y}\|^2}{2} \right) \quad (20)$$

Lastly, we incorporate a hyper algorithm that does component wise Monte Carlo. In each step, we draw $\frac{1}{\sigma_\epsilon^2}$ from the gamma distribution. Based on Gibbs sampling, since we know the posterior distribution exactly (as we picked gamma as the prior), the acceptance probability for σ_ϵ^2 at each step is 1. Then, we use the updated σ_ϵ^2 and draw for the new \mathbf{Z} using the AM algorithm developed in the previous part. The acceptance/rejection of the step is based on the same equation as Metropolis-Hasting (Eq. 15). The updated \mathbf{Z} is then used to draw the next σ_ϵ^2 . and the process continues until the number of MC steps is met.

A comparison between the hyperprior and posterior of σ_ϵ^2 is depicted in Fig. 7. As expected, the prior of σ_ϵ^2 is essentially the gamma distribution, which has a relatively large variance (spans 0 to 0.2 approximately). Following the Monte Carlo simulations, the posterior density of σ_ϵ^2 is plotted in Fig. 7 (right). We see that the variance of the distribution has greatly reduced, and the span of the σ_ϵ^2 is now between 0 to 0.0006. This implies that the data is effective in inference, reducing the uncertainty in the value of σ_ϵ^2 .

The posterior mean and standard deviation of the log permeability as a function of the x coordinate is plotted in Fig. 8. Similar to the previous part, the mean and standard deviation of the model prediction align with the true values of the permeability. However, we see that the variance from the MC simulation is now larger as compared to before. This arises due to the contributions of the uncertainty from σ_ϵ^2 , leading to higher standard deviations in the log permeability to be obtained for a given dataset. Nevertheless, the inferred permeability field still replicates the maximum and minimum trends fairly accurately.

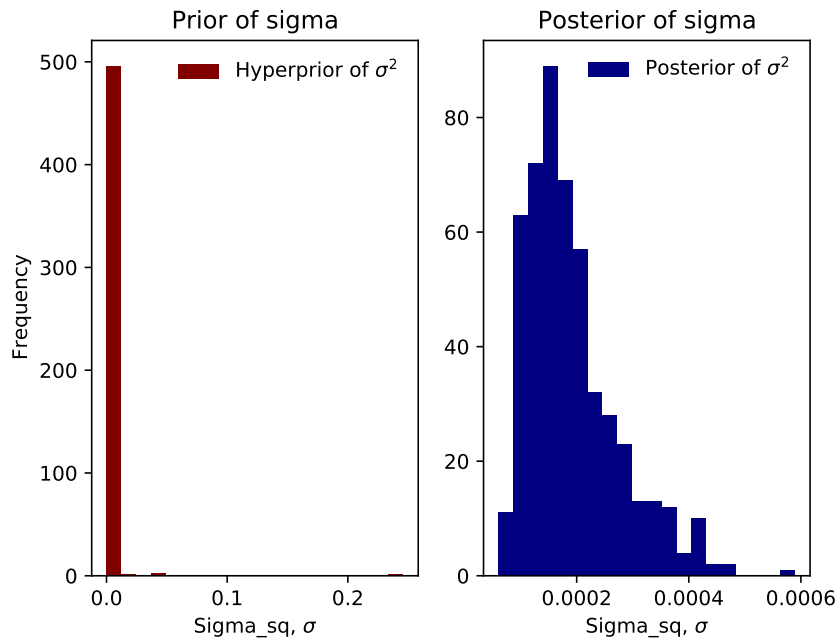


Figure 7: (Left) Plot of the hyperprior density of σ^2 , (Right) plot of the marginal poster density of σ^2 .

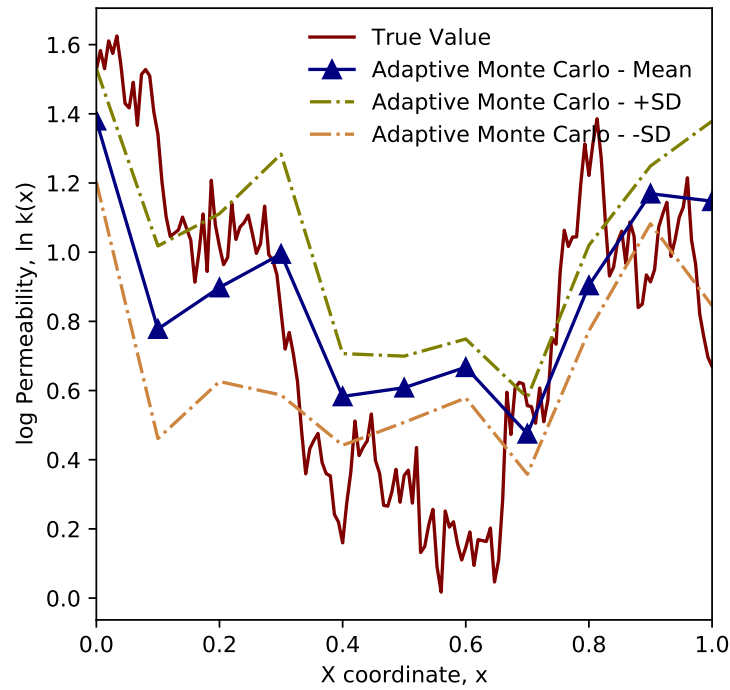


Figure 8: Plot of the marginal posterior mean and standard deviation of the log permeability as a function of the x coordinate, as compared to the true values from the provided dataset.

Subsection (c)

Here, we combine the approached in parts (a) and (b) to investigate the posterior distribution of $u(x = 0)$. The result of the simulations are presented in Fig. 9. From inspection, with a deterministic σ_ϵ , it appears that the posterior distribution of $u(x = 0)$ is flatter and more uniformly distributed about approximately 1.33. With a variable σ_ϵ , the distribution becomes more less uniform and more centered around the mode. Similarly, the mode appears to be close to 1.33.

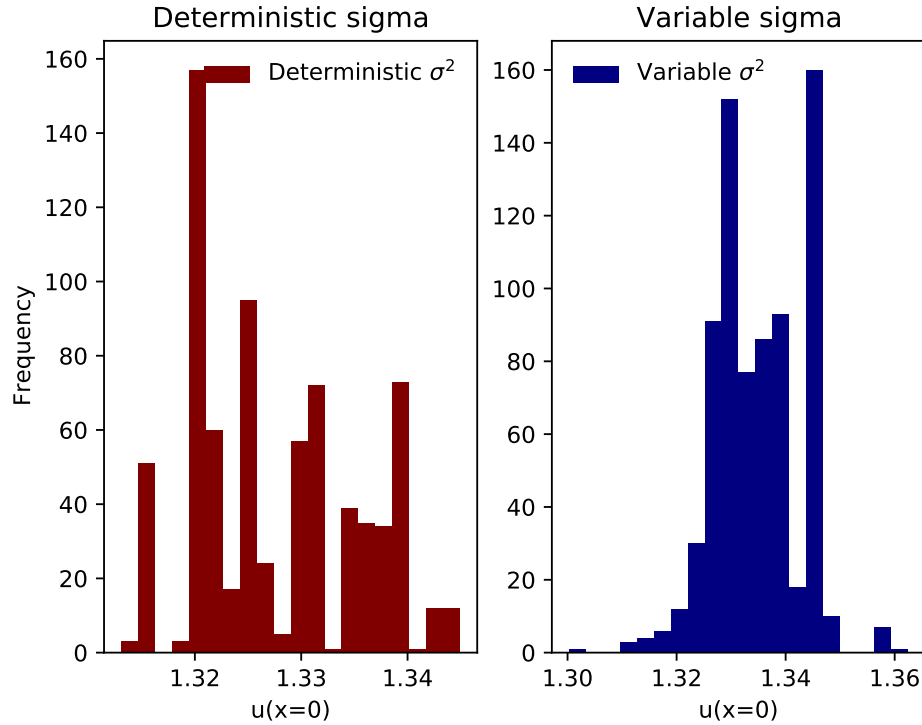


Figure 9: Posterior predictive density of $u(x=0)$ for (left) deterministic σ_ϵ and (right) variable σ_ϵ .

Subsection (d)

Here, we want to find the best x coordinate out of the ten given points such that the spatially averaged posterior variance of the permeability is minimized, implying that the most information is contained within the observation. We run 5000 Monte Carlo samples, with 30 stochastic dimensions in the KL expansion, and calculate the variance of the points at each location. The averaged variance of $k(x)$ is plotted in Fig. 10. Based on the plot, it appears that $x = 0.3$ and $x = 0.6$ has the lowest spatially averaged variance, and one should place their pressure monitor at those locations.

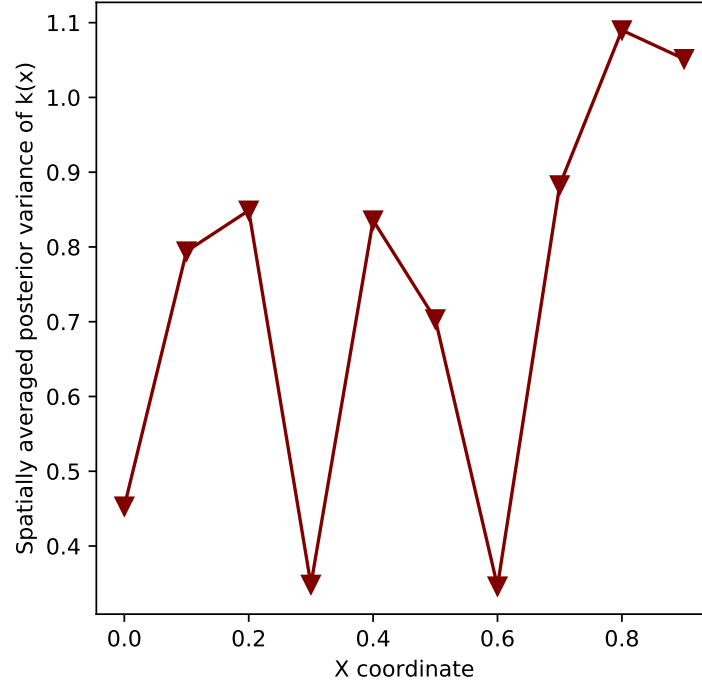


Figure 10: Plot of the spatially averaged posterior variance as a function of the x coordinate.

Question 2

Subsection (a)

In this problem, we use the analysis step of the Kalman filter algorithm to infer the posterior density of the coefficients of the KL expansion, using the supplied dataset as the observation. Using an ensemble size of 500 (M), we define the state space as the values of the coefficients ($\vec{\theta}$) in the KL expansion (size d). We pick $d = 10$ for this problem. Given that the coefficients of the KL expansion are distributed as standard normal distributions for their priors, we can generate $M = 500$ samples of the $d = 10$ coefficients. For each $m \in M$, we evaluate the value of \mathbf{u} at each position.

Here, we are unable to derive the analytical form of the Kalman gain matrix. However, we can estimate from the sample data of $\vec{\theta}$ and the observation data \vec{Y} (essentially ‘uobserved’).

$$\mathbf{G} = \Sigma_{\theta, \mathbf{Y}} \Sigma_{\mathbf{Y}}^{-1} \quad (21)$$

where $\Sigma_{\theta, \mathbf{Y}}$ is the covariance of $\vec{\theta}$ and \vec{Y} .

The $\vec{\theta}$ can then be updated using the analysis step:

$$\hat{\vec{\theta}} = \vec{\theta} + \mathbf{G}(\vec{Y} - \vec{u} + \vec{\eta}) \quad (22)$$

where $\vec{\eta}$ is the vector of noise sampled for the observation model. The derived $\hat{\vec{\theta}}$ is used to calculate the posterior density field of $k(x)$ based on the KL expansion of Y .

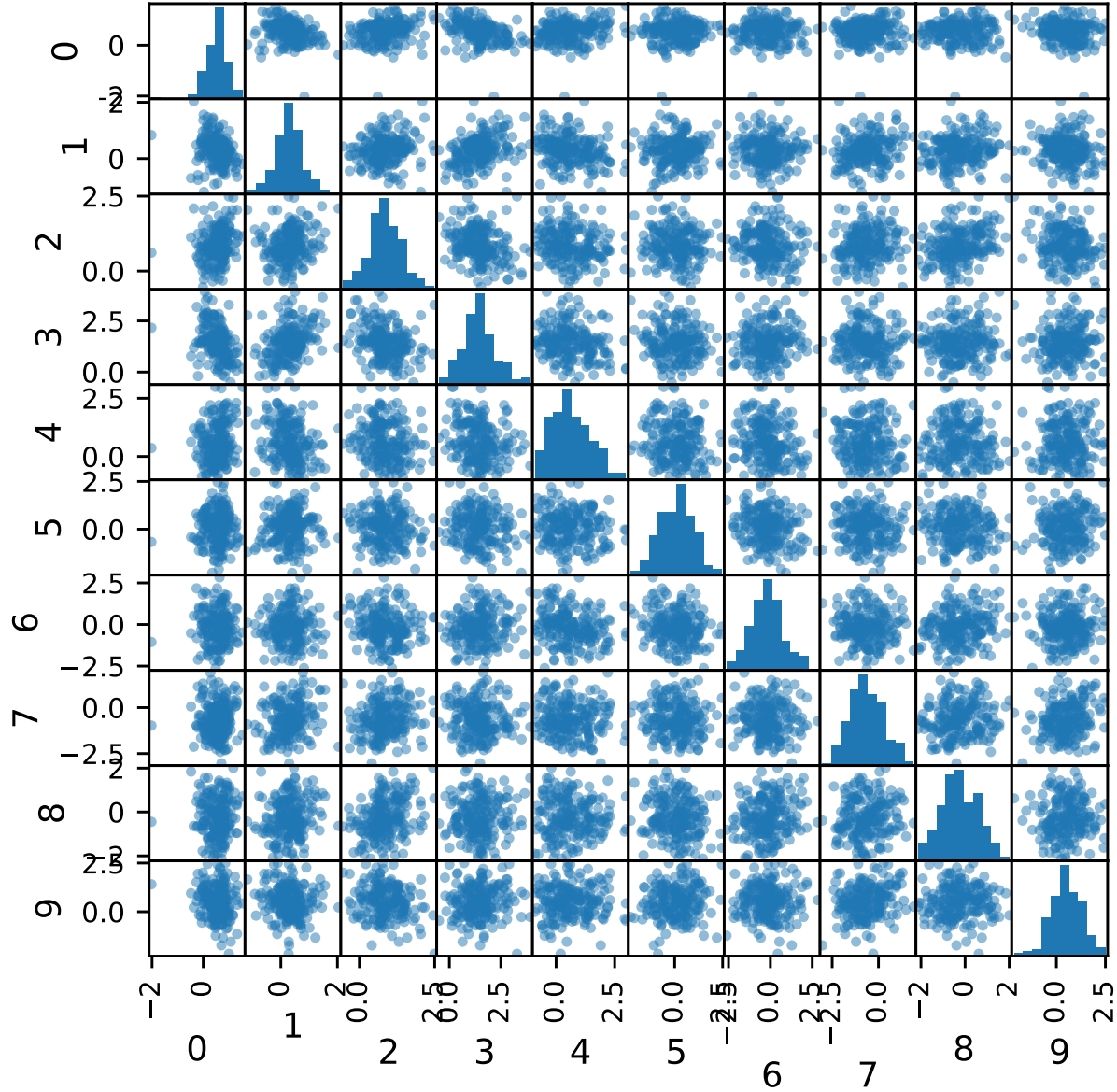


Figure 11: ‘Scattermatrix’ diagram of the 10 stochastic components. The posterior densities of the 10 respective components are depicted along the diagonal, while the off-diagonal plots illustrate the correlation between the respective components.

The posterior ensemble of the d components of $\hat{\theta}$ is presented in Fig. 11. Unlike the results from AM Metropolis Hasting, the posterior densities of the respective components are still single modal and centered. Based on the off-diagonal plots, it appears that each $\hat{\theta}$ remains uncorrelated, as evident by the assorted and messy scatter plots. This implies that the Kalman filter method does not introduce correlation between the respective dimensions, unlike the AM MCMC algorithms.

Subsection (b)

In this section, we compare the results of the posterior from the Kalman filter to the previous section with MCMC.

Fig. 12 illustrate a plot of the marginal posterior mean and standard deviation of the log permeability as a function of the x coordinate. In general, the Kalman update algorithm aided in statistical inference, as the posterior densities are able to capture and reproduce the regions of high densities near $x = 0$ and $x = 1$, and the low density region near $x = 0.6$. However, unlike the results from MCMC, the variance of the posterior with Kalman update is a lot larger, as evident by the wider standard deviation intervals in Fig. 12 as compared to Fig. 4. Nevertheless, the means of the log permeability from both methods are similar and largely coincide.

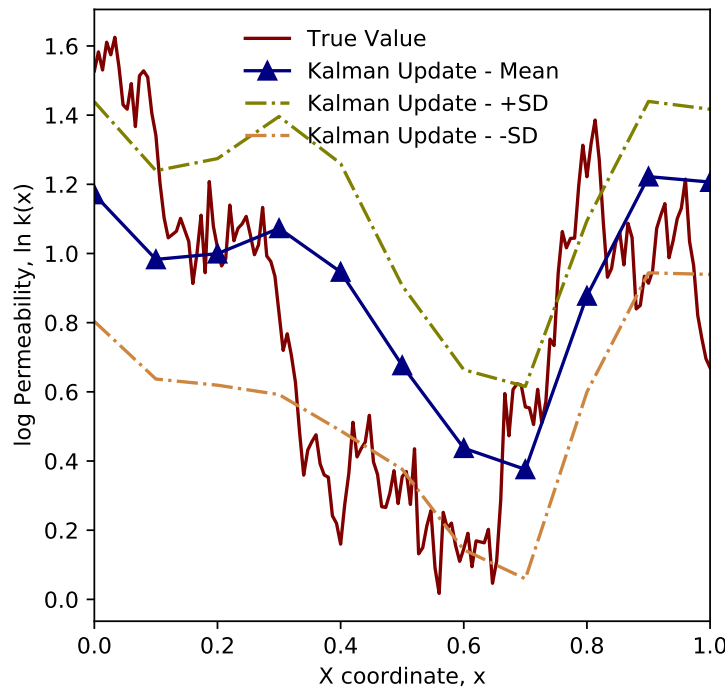


Figure 12: Plot of the marginal posterior mean and standard deviation of the log permeability as a function of the x coordinate, as compared to the true values from the provided dataset.

Fig. 13 and 14 illustrate the posterior and prior covariance fields of the log permeabilities, as a function of the x spatial coordinates. Comparing the two plots, in agreement with the previous paragraph, the Kalman update algorithm reduced the variance of the posterior density relative to the prior. The maximum covariance along the main diagonal fell from approximately 0.36 to 0.14. However, unlike the MCMC algorithm, the covariance between the different spatial coordinates decays quickly with the distance apart. This implies that the permeability at different spatial coordinates appear to be largely uncorrelated with each other, an observation that is distinct from the results from the MCMC algorithm. The precise reason for the observation remains unclear to me presently. However, it is evident that the Metropolis-Hasting acceptance condition introduces a dynamic coupling between the various stochastic parameters, which inherent couples and creates

correlations between the parameters. This practice is not observed when the Kalman gain method is used, leading to much weaker correlative behaviors between the respective positions.

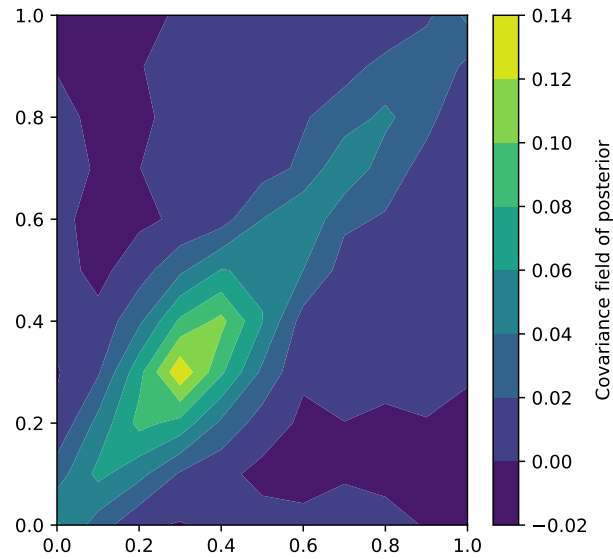


Figure 13: Plot of the posterior covariance field of the log permeability as a function of the spatial coordinate.

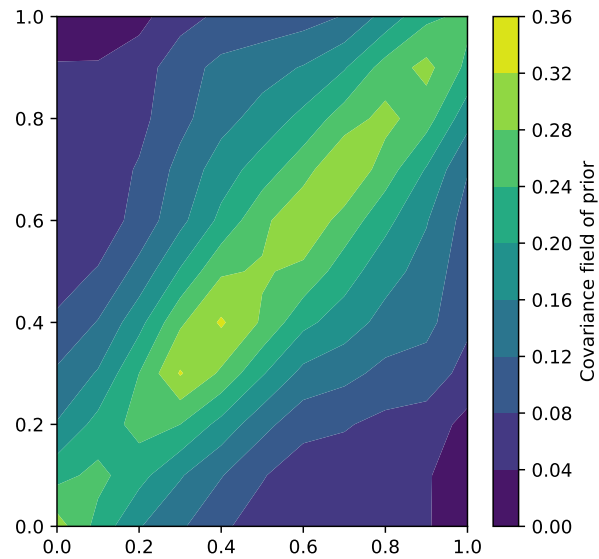


Figure 14: Plot of the prior covariance field of the log permeability as a function of the spatial coordinate.

Question 3

Subsection (a)

In this problem, we model the chaotic Lorenz-63 system, which is described by the ODE system:

$$\frac{dZ_1}{dt} = -\sigma Z_1 + \sigma Z_2 \quad (23)$$

$$\frac{dZ_2}{dt} = -Z_1 Z_3 + \rho Z_1 - Z_2 \quad (24)$$

$$\frac{dZ_3}{dt} = Z_1 Z_2 - \beta Z_3 \quad (25)$$

In this part, we implement an ensemble Kalman filter to estimate the state of the Lorentz-63 system. To do so, we first need an observation dataset to guide its motion. To accomplish this, we implement an 4th order conventional Runge-Kutta integration scheme on the ODE system Eq. 25, from $t = 0$ to $t = 100$, using $\delta_t = 0.05$. The state parameters are used to derive the observation vector, \mathbf{Y} , where $\mathbf{Y} \sim N(\mathbf{Z}_{\mathbf{K}}, \sigma^2 \mathbf{I}_3)$, and $\sigma^2 = 4$.

Here, we describe the ensemble Kalman filtering algorithm that was coded. We vary the ensemble size from 25 to 1000, with t increasing from 0 to 100. For each ensemble simulation, we begin by sampling the initial states from a normal distribution, $N(\mathbf{0}, \mathbf{I}_3)$. Then, the forecast stage is performed as:

$$\hat{\mathbf{Z}}_{K(n+1,i)} = \Psi(\mathbf{Z}_{\mathbf{K}(n,i)}) \quad (26)$$

where Ψ is the ODE solver with a time step of δ_t . The sample covariance is calculated as:

$$\mathbf{C}_{n+1} = \text{Cov}_M(\hat{\mathbf{Z}}_{K(n+1,i)}) \quad (27)$$

Then, the Kalman gain matrix is calculated with:

$$\mathbf{K}_{(n+1)} = \mathbf{C}_{n+1}(\mathbf{\Gamma} + \mathbf{C}_{n+1})^{-1} \quad (28)$$

In this case, $\mathbf{H} = \mathbf{I}$, and $\mathbf{\Gamma}$ is the covariance matrix of the noise. The assimilation of the state is finalized with:

$$\mathbf{Z}_K(n+1, i) = \hat{\mathbf{Z}}_{K(n+1,i)} + \mathbf{K}_{(n+1)}(\mathbf{Y}_{(n+1)}^{ob} + \eta_{n+1,i} - \hat{\mathbf{Z}}_{K(n+1,i)}) \quad (29)$$

where $\eta_{n+1,i} \sim N(0, \mathbf{\Gamma})$.

Using the aforementioned algorithm, the plot of the root mean square error as a function of the ensemble size is plotted in Fig. 15. Overall, we see that the RMSE decreases with increasing ensemble size, and approaches a minimum value of approximating 2. Coincidentally, the standard deviation of the noise (in this case, of η) is also equals to 2, as reflected by the blue line. The noise is introduced in the observation model to improve model fidelity when noisy data is used for statistical inference. Consequently, the model results suggests the inference problem can be solved up to the expected variance induced by the artificial noise in the observation model.

Nevertheless, with a sufficiently large ensemble (here $M > 200$), the Kalman filter is efficient in its inferential application. This is exemplified in the tracer plot in Fig. 16. Here, the green lines

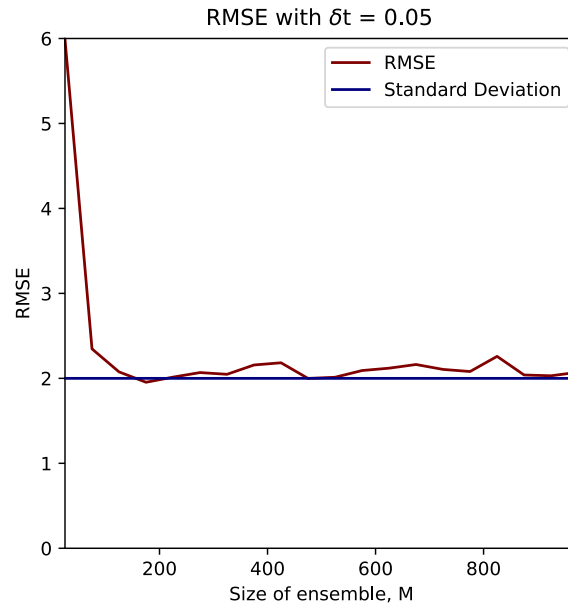


Figure 15: RMSE of the Kalman filtering algorithm as a function of the ensemble size, M .

reflect the trajectory of the state of the observation data, which was derived from the direct RK-4 integration of the ODE system. The red line reflects the trajectory of the mean value of the ensemble. Overall, we see that the red line lags behind the green line, but follows the overall trends effectively. After the transient phase passes, the samples from the ensemble is highly similar in the posterior density as the observation. This can be seen by the "red circling" path the mean takes about the space where the observation states reside (green).

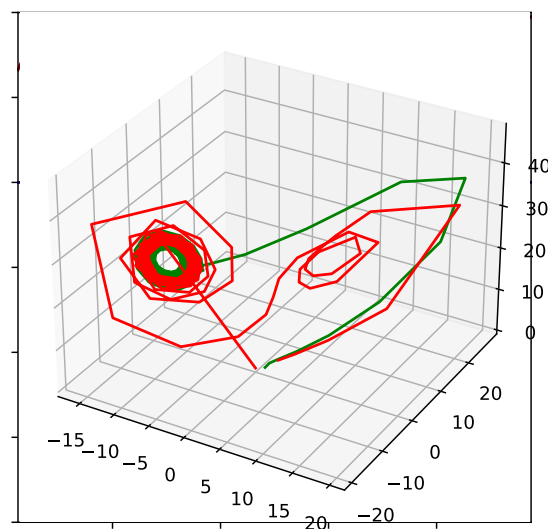


Figure 16: Tracer plot of the mean state (in red) and the observation state (in green).

Subsection (b)

In this section, we perform particle filtering to infer the posterior density of the states $\mathbf{Z}_{\mathbf{K}}$, based on the same observation data that was used for the Kalman filters. We let $\mathbf{Z}_{\mathbf{K}}^{(i)}$ denote the state of the i -th particle at time t . To enable stable inference from noisy data, we introduce some stochastic behavior by adding Gaussian noise into the state space model. The variance of the Gaussian noise is 10^{-4} , and is drawn from $N(\mathbf{0}, 10^{-4}\mathbf{I}_3)$. Then, we let the likelihood of the observation be denoted by:

$$g_t(y_t^* | \mathbf{Z}_{\mathbf{K}}^{(i)}) = N(y_t^*; \mathbf{Z}_{\mathbf{K}}^{(i)}, 10^{-4}\mathbf{I}_3) \quad (30)$$

The procedure of particle filtering is as follows. First, we create an ensemble of size M , for the initial conditions of $\mathbf{Z}_{\mathbf{K}}^{(0)}$. Here, $\mathbf{Z}_{\mathbf{K}}^{(0)}$ is drawn uniformly from $N(\mathbf{0}, \mathbf{I}_3)$. At each time step t , we draw the state of the next particle from a biasing proposal distribution, which is denoted by $q_t^{(i)}$. For classical particle filtering, we can either pick the prior or the locally optimal distribution as the proposal distribution, with the latter being the ideal choice. However, we pick the prior distribution as the proposal in this application, due to computational difficulties in estimating the locally optimal distribution given the non-linearity of the system. Nevertheless, we will show later that the proposal still works in inferring the correct posterior distribution, with the appropriate re-sampling strategies. Thus, the chosen proposal is written as:

$$q_t^{(i)}(\mathbf{Z}_{\mathbf{K}}^{(i)} | \mathbf{Z}_{1:\mathbf{K}-1}^{(i)}) = N(\mathbf{Z}_{\mathbf{K}-1}^{(i)}, 10^{-4}\mathbf{I}_3) \quad (31)$$

where $q_t^{(i)}$ is the prior distribution.

At each step, we calculate the effective sample size to assess for the need of re-sampling. The EES can be computed as:

$$EES = \frac{1}{\sum_{i=1}^M W(\mathbf{Z}_{1:\mathbf{K}-1}^{(i)})^2} \quad (32)$$

When the EES exceeds a threshold (here, we define the threshold as $M/10$), we resample the particles from a multinomial distribution, where each i -th particle has a probability of sampling that is proportional to its weight $W(\mathbf{Z}_{1:\mathbf{K}-1}^{(i)})$. We then draw number of the respective particles. We reject the particles that return a draw of 0, and duplicate the states of the selected particles proportionally to the number of times they are drawn. The re-drawn state vectors are then used in the next iteration of the particle filtering algorithm.

At each step of the algorithm, the expected/mean value of the ensemble is assessed, using the following equation, using the normalized weights:

$$\mathbb{E}[\mathbf{Z} | Y] = \sum_{i=1}^M \mathbf{Z}_{1:\mathbf{K}}^{(i)} W(\mathbf{Z}_{\mathbf{K}}^{(i)}) \quad (33)$$

The plot of the RMSE as a function of the ensemble size, M is depicted in Fig. 17. Unlike the RMSE relationship observed for the Kalman filtering, the RMSE appears to decay very quickly with M , and oscillate under the value of 2. However, there are two sudden spikes observed at $M = 1500$ and $M = 2400$ approximately. It remains unclear why the spikes occurred, and might just be numerical noise from the re-sampling algorithm due to poor initial guesses. Nevertheless, the RMSE values obtained by particle filtering, on average, are lower than the ones obtained by

Kalman filtering previously. From inspection, it seems like an ensemble size of $M = 300$ or larger is required for particle filtering to be more effective than Kalman filtering.

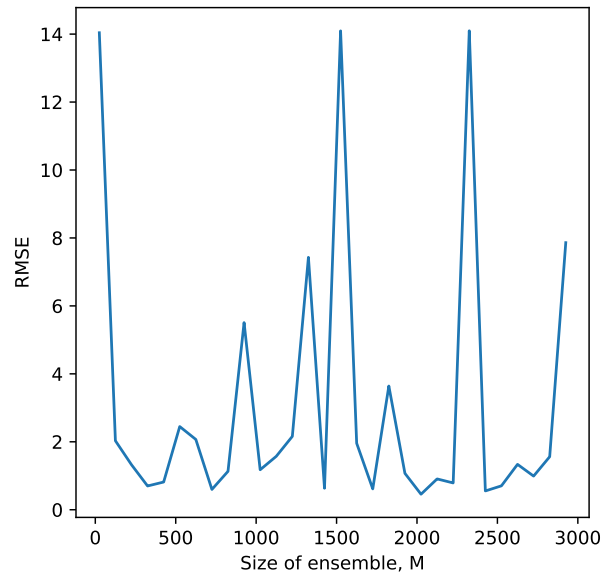


Figure 17: RMSE of the Kalman filtering algorithm as a function of the ensemble size, M .

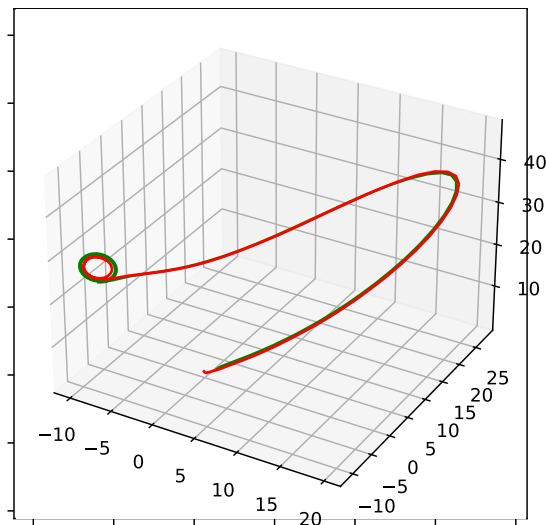


Figure 18: Tracer plot of the mean state (in red) and the observation state (in green).

Similar to the previous section, we plot the trajectory of the states for the observation and the mean field. This is illustrated in Fig. 18. Compared to the same figure obtained for Kalman

filtering, we see that the state trajectory with particle filtering is a lot smoother and closer to the observation states. We believe that this occurred due to the weights used in the computation of the expectation. Using larger weights for points with higher confidence tend to reduce the variance of the sample mean, leading to more accurate trajectories to be observed. The tighter adherence between the two trajectories also lead to much lower RMSE to be obtained, supporting the conclusions from the previous figure.

Lastly, we investigate the compare the covariance of the posterior estimates from Kalman and particle filtering. At each step of the simulation, we use the state vector to compute both the mean and covariance matrices. The covariance matrix at each step is a 3 by 3 matrix, and we average the values of the covariance matrix for an ensemble size of 1000, over 1000 timesteps.

The average covariance matrix from the particle filter is:

$$\begin{bmatrix} 0.00083655 & 0.00022522 & -0.00064789 \\ 0.00022522 & 0.00073335 & -0.00011323 \\ -0.00064789 & -0.00011323 & 0.00167213 \end{bmatrix} \quad (34)$$

Similarly, the average covariance matrix from the ensemble Kalman filter is:

$$\begin{bmatrix} 21.64698269 & 23.4039027 & 15.87880341 \\ 23.4039027 & 37.05896005 & 5.90809206 \\ 15.87880341 & 5.90809206 & 51.35770641 \end{bmatrix} * 10^{-3} \quad (35)$$

Overall, we see that the average covariance matrix from the Kalman filter is much higher than the particle filter.