TRANSPORT MAP ACCELERATED-PAIS, AND APPLICATION TO INVERSE PROBLEMS ARISING FROM MULTISCALE STOCHASTIC REACTION NETWORKS

SIMON COTTER, YANNIS KEVREKIDIS, PAUL RUSSELL

Abstract. In many applications, inverse problems arise where where there are complex correlations between the different parameters which we wish to infer from data. The correlations often manifest themselves as lower dimensional manifolds on which the likelihood function is invariant, or varies very little. This can be due to trying to infer unobservable parameters, or due to sloppiness in the model which is being used to describe the data. In such a situation, standard sampling methods for characterising the posterior distribution which do not incorporate information about this structure will be highly inefficient. Moreover, most methods are inherently serial in nature, and as such are not expoiting the parallelised nature of modern computer infrastructure. In this paper, we seek to develop a method to tackle this problem, using optimal transport maps to simplify posterior distributions which are concentrated on lower dimensional manifolds.

We demonstrate the approach by considering inverse problems arising from partially observed stochastic reaction networks. In particular, we consider systems which exhibit multiscale behaviour, but for which only the slow variables in the system are observable. We demonstrate that certain multiscale approximations lead to more consistent approximations of the posterior than others.

1. Introduction. In Section 2, we will briefly reintroduce Parallel Adaptive Importance Sampling (PAIS), a variant of Population Monte Carlo (PMC) which incorporates state of the art resamplers. In Section 3 we show how an appropriate transport map can be constructed from importance samples which maps the posterior close to a reference Gaussian measure. In Section 4 we show how such a map can be incorporated into a sophisticated parallel MCMC infrastructure in order to accelerate mixing. In Section 5 we seek to show the advantages of this approach through the analysis of test problems. In Section 6 we consider how likelihoods can be approximated using multiscale methodologies in order to carry out inference for multiscale and/or partially observed stochastic reaction networks. In Section 7 we present some numerical examples, which serve to demonstrate the increased efficiency of the described sampling methodologies, as well as investigating the posterior approximations discussed in the previous section. We conclude with a discussion in Section 8.

2. Parallel Adaptive Importance Sampling. PAIS****CITE**** is a variant of PMC****CITE****, which is a family of methods which are based on importance sampling. In importance sampling, we attempt to characterise a target density through sampling from that density. However, the target density π itself is often too complex to sample from directly, so we instead sample from a proposal density χ. Each sample $\theta^{(k)} \sim \chi$ is then weighted by $w_k = \frac{\pi(\theta^{(k)})}{\chi(\theta^{(k)})}$, to take account of the bias of sampling from a different distribution to π. Monte Carlo estimates using a sample of size N of a function f with respect to π can then be made through the formula

$$\mathbb{E}_\pi(f) \approx \frac{1}{\bar{w}} \sum_{k=1}^{N} w_k f(\theta^(k)).$$

This method works well when π and χ are close, but can be excruciatingly slow when they are not. The idea behind PMC methods is to construct a good proposal distribution, either from the entire history of the algorithm up to the current point, or to use the current state of a whole ensemble of M particles in the system.
In PAIS, the proposal distribution χ is chosen to be the equally weighted mixture of any choice of MCMC proposal kernel, evaluated at each of the current particles in

the system. If $\theta^{(k)} = [\theta_1^{(k)}, \theta_2^{(k)}, \ldots, \theta_M^{(k)}]^\top$ is the current state of the ensemble, and we wish to use an MCMC proposal density $q(\cdot;\cdot,\beta)$ then

$$\chi^{(k)} = \frac{1}{M}\sum_{i=1}^{M} q(\cdot;\theta_i^{(k)}.$$

Often the variance of the MCMC proposal kernels can be tuned using their respective algorithmic parameters β. Good values for these algorithmic parameters can be found by optimising for the effective sample size of the importance sample that is produced (see ***CITE*** for more details).
If the ensemble is large enough, and the chain has entered probabilistic stationarity, then the current state of the ensemble is a good rough discrete approximation of the target density, and in turn $\chi^{(k)}$ is close enough to π to produce an efficient importance sample $\{\hat{\theta}^{(k)}, w^{(k)}\}$. It can be advantageous to use stratified sampling of the mixture in order to ensure that the sample made is as representative as possible of the target density, i.e.

$$\hat{\theta}_i^{(k)} \sim q(\cdot;\theta_i^{(k)}$$

for each $i = 1, 2, \ldots, M$. We now have a weighted sample, and it would be inadvisable to use an equally weighted mixture of proposal distributions from each of these points. Therefore, before starting the next iteration, the importance sample $\{\hat{\theta}^{(k)}, w^{(k)}\}$ is resampled to produce an equally weighted sample, ready for the next iteration of the algorithm. In PAIS, a state-of-the-art resampler is used, which uses optimal transport methods to find the equally weighted discrete sample which best represents the statistics of the importance sample***CITE****. For larger ensemble sizes M this can become expensive, in which case a greedy approximation of this algorithm, the Approximate Multinomial Resampler (AMR) can be implemented***CITE****. The output of the resampler is then denoted $\theta^{(k+1)}$ and the algorithm is ready for the next iteration. The importance samples $\{\hat{\theta}^{(k)}, w^{(k)}\}$ The PAIS is summarised in Algorithm 1.

Algorithm 1: The PAIS Algorithm.

Initialise $\theta^{(1)} \sim \mu_0$.
for $k = 1, \ldots, N$ **do**
  Sample $\hat{\theta}_i^{(k)} \sim q(\cdot;\theta_j^{(k)},\beta)$, for $i = 1, \ldots, M$.
  Calculate $w^{(k)} = (w_1^{(k)}, \ldots, w_M^{(k)})^\top$, where

$$w_i^{(k)} = \frac{\mu(\hat{\theta}_i^{(k)})}{\chi^{(k)}(\hat{\theta}_i^{(k)};\theta^{(k)},\beta)}.$$

  Resample $\theta^{(k+1)} \leftarrow \|w^{(k)}\|_1^{-1} \sum_{j=1}^{M} w_j^{(k)} \delta_{\hat{\theta}_j^{(k)}}(\cdot)$.
Output $\{(w^{(n)}, \hat{\theta}^{(n)}\}_{n=1}^{N}$.

One problem with the PAIS and other PMC methods can become apparent if the target density has very strong correlations in its structure, in particular if that correlation is not global but only local. In this case, unless the proposal densities q are informed

by this local structure, the mixture distribution proposal may not well approximate π without a very large large ensemble size M, which can become inhibitively expensive. Some methods have been proposed ****CITE**** which use samples local to each particle to inform local covariance structure.
In this paper, we investigate the use of transport maps to learn local covariances across the whole of the domain, in order to stabilise PMC-type methods, and make these methods more applicable to a wider range of more challenging inference problems.

3. Construction of transport maps in importance sampling. In [1] the transport map was introduced to provide a transformation from the prior distribution to the posterior distribution, the idea being that one could draw a moderately sized sample from the prior distribution and use this sample to approximate a map onto the target space. Once this map was known to the desired accuracy a larger sample from the prior could be used to investigate the posterior distribution. This methodology was adapted in [3] to form a new proposal method for MH algorithms. In this case, rather than transforming a sample from the prior into a sample from the target distribution, the map transforms a sample from the posterior onto a reference space. The reference density is chosen to allow efficient proposals using a simple proposal distribution such as a Gaussian centred at the previous state. Proposed states can then be mapped back into a sample from the posterior by applying the inverse of the transport map. Proposing new states in this way allows us to make large steps around complex probability distributions. It is also feasible in this framework to assume that the reference density is close enough to a standard Gaussian that we can efficiently propose moves using a proposal distribution which is independent of the current state, e.g. choose $q(\theta) = \mathcal{N}(0, I_n)$.
In this Section we outline the methodology in [3] for approximately coupling the target, μ_θ, with the reference distribution, μ_r, and show how the map can be constructed using a weighted sample and hence how we can incorporate the map into importance sampling schemes.
DEFINITION 3.1 ((Exact) Transport Map T). *A transport map T is a function $T\colon \mathcal{X} \to \mathbb{R}^d$ such that the pullback of the reference measure with density $\phi(\cdot)$,*

$$\tilde{\pi}(\theta) = \phi(T(\theta))|J_T(\theta)|, \tag{3.1}$$

is equal to the target density $\pi(\theta)$ for all $\theta \in \mathcal{X}$. The pullback is defined in terms of the determinant of the Jacobian of T,

$$|J_T(\theta)| = det \begin{bmatrix} \partial_{\theta_1} T_1(\theta) & \dots & \partial_{\theta_d} T_1(\theta) \\ \vdots & \ddots & \vdots \\ \partial_{\theta_1} T_d(\theta) & \dots & \partial_{\theta_d} T_d(\theta) \end{bmatrix}.$$

DEFINITION 3.2 (Target and Reference Space). *The transport map pushes a particle from a target space $\mathcal{X}$, that is a subset of $\mathbb{R}^d$ equipped with a target measure μ_θ, onto a reference space, R, again a subset of $\mathbb{R}^d$ equipped with the reference measure μ_r.*
Armed with such a map, independent samples can be made of the target measure, using the pullback of the reference density ϕ through T^{-1}. Clearly the pullback only exists when T is monotonic, i.e. has a positive definite Jacobian, and has continuous first derivatives. Not all maps satisfy these conditions, so we define a smaller space of maps, $\mathcal{T}^\uparrow \subset \mathcal{T}$ which contains all feasible maps. This space does not necessarily contain an exact coupling between target and reference space, and so we are motivated

to formulate an optimisation problem to find the map $\tilde{T} \in \mathcal{T}^{\uparrow}$ which most closely maps the target density to the reference density.
As in previous work in [3], we can ensure invertibility if we restrict the map to be lower triangular, i.e. $\tilde{T} \in \mathcal{T}^{\triangle} \subset \mathcal{T}^{\uparrow}$. This lower triangular map has the form,

$$T(\theta_1, \ldots, \theta_n) = \begin{bmatrix} T_1(\theta_1) \\ T_2(\theta_1, \theta_2) \\ \vdots \\ T_n(\theta_1, \ldots, \theta_n) \end{bmatrix},$$

where $T_i \colon \mathbb{R}^i \to \mathbb{R}$.

3.1. The optimisation problem. Our aim is now to find the lower triangular map $\tilde{T} \in \mathcal{T}^{\triangle}$ such that the difference between target density and the pullback of the reference density is minimised. As in [3], we choose the cost function to be the Kullback-Leibler (KL) divergence between the posterior density and the pullback density,

$$D_{\mathrm{KL}}(\pi \| \tilde{\pi}) = \mathbb{E}_{\pi}\left[\log\left(\frac{\pi(\theta)}{\tilde{\pi}(\theta)}\right)\right].$$

This divergence results in some nice properties which we will explore in the following derivation. The KL divergence is not a true metric since it is not symmetric, however it is commonly used to measure the distance between probability distributions due to it's relatively simple form, and because it provides a bound for the square of the Hellinger distance by Pinsker's inequality [4],

$$D_{KL}(p \| q) \geq D_H^2(p, q),$$

which is a true metric between probability distributions p and q. Given the form of the pullback in Equation (3.1), now taken through an approximate map $\tilde{T}$, the divergence becomes

$$D_{\mathrm{KL}}(\pi \| \tilde{\pi}) = \mathbb{E}_{\pi}\left[\log \pi(\theta) - \log \pi_r(\tilde{T}(\theta)) - \log |J_{\tilde{T}}(\theta)|\right].$$

We note the posterior density is independent of $\tilde{T}$, and so it is not necessary for us to compute it when optimising this cost function. This expression is a complicated integral with respect to the target distribution, for which the normalisation constant is unknown. However this is exactly the scenario for which we would turn to MCMC methods for a solution.
To find the best coupling, $\tilde{T} \in \mathcal{T}^{\triangle}$, we solve the optimisation problem,

$$\tilde{T} = \arg\min_{T \in \mathcal{T}^{\triangle}} \mathbb{E}_{\pi}\left[-\log \pi_r(T(\theta)) - \log |J_T(\theta)|\right]$$

which has a unique solution since the cost function is convex. We also include a regularisation term, which is required for reasons which will become clear later. The optimisation problem now takes the form

$$\tilde{T} = \arg\min_{T \in \mathcal{T}^{\triangle}} \left[\mathbb{E}_{\pi}\left[-\log \pi_r(T(\theta)) - \log |J_T(\theta)|\right] + \beta \mathbb{E}(T(\theta) - \theta)^2\right]. \tag{3.2}$$

The parameter $\beta > 0$ does not need to be tuned, as experimentation has shown that the choice $\beta = 1$ is sufficient for most problems. This expectation can be approximated by using an MCMC approximation. The form of the penalisation term promotes maps which are closer to the identity, and so prevents overfitting when the quality or size of the current sample from the posterior is not sufficient.

3.2. The structure of the map. Before we continue with the derivation of the optimisation problem, we consider the structure of the map in more detail. The lower triangular structure of the map not only guarantees monotonicity, it also allows for efficient calculation of the pullback density, as well as the inverse of the map, $\tilde{T}^{-1}$. The Jacobian of $\tilde{T}$ is a lower triangular matrix,

$$J_T(\theta) = \begin{bmatrix} \partial_{\theta_1}\tilde{T}_1(\theta) & \dots & \partial_{\theta_d}\tilde{T}_1(\theta) \\ \vdots & \ddots & \vdots \\ \partial_{\theta_1}\tilde{T}_d(\theta) & \dots & \partial_{\theta_d}\tilde{T}_d(\theta) \end{bmatrix} = \begin{bmatrix} \partial_{\theta_1}\tilde{T}_1(\theta) & \dots & 0 \\ \vdots & \ddots & \vdots \\ \partial_{\theta_1}\tilde{T}_d(\theta) & \dots & \partial_{\theta_d}\tilde{T}_d(\theta) \end{bmatrix}$$

since $\partial_{\theta_n}\tilde{T}_k(\theta) = 0$ for all $n > k$. This lower triangular structure means that the determinant of the Jacobian is a product of the diagonal elements which, when we take logs, becomes

$$\log |J_{\tilde{T}}(\theta)| = \sum_{i=1}^{d} \log \partial_{\theta_i}\tilde{T}_i(\theta), \tag{3.3}$$

where we note that this term is separable in terms of the dimension i.
Inverting $\tilde{T}$ at a point r is simplified by the lower triangular structure of the map. The map component $\tilde{T}_1(\theta)$ is a univariate polynomial in θ_1, so we can find the inverse of this function by solving the equation $T_1(\theta_1) = r_1$. This inversion tells us the value of θ_1, which means the next component is again a univariate polynomial, $T_2(\theta_2; \theta_1) = r_2$. We can then perform d root finding problems instead of a full d dimensional non-linear solve.
We require that the first derivatives of the map are continuous, which is easy to enforce by the choice of basis functions. Here we assume that the map will be built from a family of orthogonal polynomials, $\mathcal{P}(\theta)$, not necessarily orthogonal with respect to the target distribution. Each component of the map is defined as a multivariate polynomial expansion,

$$\tilde{T}_i(\theta; \gamma_i) = \sum_{\mathbf{j} \in \mathcal{J}_i} \gamma_{i,\mathbf{j}} \psi_{\mathbf{j}}(\theta). \tag{3.4}$$

The parameter $\gamma_i \in \mathbb{R}^{M_i}$ is a vector of coefficients. Each component of γ_i corresponds to a basis function $\psi_{\mathbf{j}}$, indexed by the multi-index $\mathbf{j} \in \mathbb{N}_0^d$. These multi-indices are elements of the multi-index set $\mathcal{J}_i$. A multi-index defines a product of univariate polynomials in θ_k,

$$\psi_{\mathbf{j}}(\theta) = \prod_{k=1}^{i} \varphi_{j_k}(\theta_k), \quad \text{for} \quad \mathbf{j} \in \mathcal{J}_i,$$

and where $\varphi_{j_k}(\theta_k) \in \mathcal{P}(\theta_k)$. Since $\tilde{T}$ is lower triangular, a multi-index $\mathbf{j} \in \mathcal{J}_i$ only contains entries for univariate polynomials in θ_k for $k \leq i$.
The cardinalities of the multi-index sets, $M_i = \text{card}(\mathcal{J}_i)$, give the number of unknowns in our optimisation problem, and so we would like to keep this number as small as possible. One option is to use polynomials of total order p,

$$\mathcal{J}_i^{\text{TO}} = \{\mathbf{j} : \|\mathbf{j}\|_1 \leq p, j_k = 0 \ \forall k > i\},$$

which is optimal in terms of the amount of information captured by the map about the target. The cardinality of $\mathcal{J}_i^{\text{TO}}$ is $M_i = \binom{i+p}{p}$ which increases rapidly in d and p,

where $i = 1, \ldots, d$. Smaller optimisation problems can be produced by constructing subsets of $\mathcal{J}_i^{\text{TO}}$. These index sets are discussed in [3]. Increased information with a slower increase in the number of map parameters can be achieved with the composition of maps discussed in [2]. Here we stick with polynomials of total order p since we work with low dimensional problems with the PAIS algorithm.

3.3. Implementation of the optimisation problem. We now discuss how we can evaluate the cost function in Equation (3.2). In [3], this expectation is approximated using an MCMC estimator, such that

$$\begin{aligned} C(T) &= \mathbb{E}_\pi \left[-\log \pi_r(T(\theta)) - \log |J_T(\theta)| \right] + \beta \mathbb{E}(T(\theta) - \theta)^2 \\ &\approx \frac{1}{K} \sum_{i=1}^{d} \sum_{k=1}^{K} \left[-\log \pi_r(T_i(\theta^{(k)})) - \log \left| \frac{\partial T_i}{\partial \theta_i}(\theta^{(k)}) \right| + \beta (T_i(\theta^{(k)}) - \theta^{(k)})^2 \right]. \end{aligned} \tag{3.5}$$

Here we diverge from previous work, as we aim to build a map from samples from an importance sampling scheme. Such samples no longer carry equal weight, and as such the Monte Carlo estimator becomes

$$C(T) = \frac{1}{\bar{w}} \sum_{i=1}^{d} \sum_{k=1}^{K} w_k \left[-\log \pi_r(T_i(\theta^{(k)})) - \log \left| \frac{\partial T_i}{\partial \theta_i}(\theta^{(k)}) \right| + \beta (T_i(\theta^{(k)}) - \theta^{(k)})^2 \right], \tag{3.6}$$

where w_k are the weights associated with each sample $\theta^{(k)}$, and $\bar{w}$ is the sum of all these weights. Optimisation of this cost function results in a map from π to some reference density π_r. By choosing the reference density to be a Gaussian density, we can simplify this expression greatly. Substitution of the Gaussian density into Equation (3.6) leads to

$$C(T) = \frac{1}{\bar{w}} \sum_{i=1}^{d} \sum_{k=1}^{K} w_k \left[\frac{1}{2} T_i^2(\theta^{(k)}) - \log \frac{\partial T_i}{\partial \theta_i}(\theta^{(k)}) + \beta (T_i(\theta^{(k)}) - \theta^{(k)})^2 \right], \tag{3.7}$$

Note that since we assume that the map is monotonic, the derivatives of each component are positive and so this functional is always finite. In practice it is infeasible to enforce this condition across the whole parameter space. We instead enforce this condition by ensuring that the derivatives are positive at each sample point. This means that when we sample away from these support points while in reference space, it is possible to enter a region of space where the map is not monotonic.

We now return to the structure of the map components given in Equation (3.4). Since the basis functions are fixed, the optimisation problem in (3.2) is really over the map components $\bar{\gamma} = (\gamma_1, \ldots, \gamma_d)$ where $\gamma_i \in \mathbb{R}^{M_i}$. Note that $C(T)$ is the sum of d expectations, and these expectations each only concern one dimension. Therefore we can rewrite (3.2) as d separable optimisation problems.

$$\arg \min_{\gamma_i \in \mathbb{R}^{M_i}} \frac{1}{\bar{w}} \sum_{k=1}^{K} w_k \left[\frac{1}{2} T_i^2(\theta^{(k)}; \gamma_i) - \log \frac{\partial T_i}{\partial \theta_i}(\theta^{(k)}; \gamma_i) + \beta (T_i(\theta^{(k)}; \gamma_i) - \theta^{(k)})^2 \right], \tag{3.8}$$

$$\text{subject to} \quad \frac{\partial T_i}{\partial \theta_i}(\theta^{(k)}; \gamma_i) > 0 \text{ for all } k = 1, \ldots, K, \ i = 1, \ldots, d.$$

The sum in Equation (3.4) is an inner product between the vector of map coefficients, and the evaluations of the basis function at a particular $\theta^{(k)}$. If we organise our basis

evaluations into two matrices,

$$(F_i)_{k,\mathbf{j}} = \psi_{\mathbf{j}}(\theta^{(k)}), \quad \text{and} \quad (G_i)_{k,\mathbf{j}} = \frac{\partial \psi_{\mathbf{j}}}{\partial \theta_i}(\theta^{(k)}),$$

for all $\mathbf{j} \in \mathcal{J}_i^{\mathrm{TO}}$, and $k = 1, \ldots, K$, then we have that

$$T_i(\theta^{(k)}) = (F_i)_{k.}\gamma_i \quad \text{and} \quad \frac{\partial T_i}{\partial \theta_i}(\theta^{(k)}; \gamma_i) = (G_i)_{k.}\gamma_i,$$

so (3.8) becomes

$$\arg\min_{\gamma_i \in \mathbb{R}^{M_i}} \frac{1}{2}(F_i\gamma_i)^\top W(F_i\gamma_i) - \mathbf{w}^\top \log(G_i\gamma_i) + \frac{\beta}{\bar{w}} \sum_{k=1}^{K} w_k (F_i\gamma_i - \theta^{(k)})^\top (F_i\gamma_i - \theta^{(k)}), \tag{3.9}$$
$$\text{subject to} \quad G_i\gamma_i > 0.$$

In this expression, the vector $\mathbf{w} = [w_1, w_2, \ldots, w_K]^\top$ is the vector of the weights, W is the diagonal matrix $W = \mathrm{diag}(w)$ and $\log(G_i\gamma_i)$ is to be evaluated element-wise. As more importance samples are made, new rows can be appended to the F_i and G_i matrices, and $F_i^\top W F_i$ can be efficiently updated via the addition of rank-1 matrices. The regularisation term in Equation (3.9) can be approximated using Parseval's identity,

$$\frac{1}{\bar{w}} \sum_{k=1}^{K} w_k (F_i\gamma_i - \theta^{(k)})^\top (F_i\gamma_i - \theta^{(k)}) \xrightarrow[K\to\infty]{} \int_{\mathbb{R}^n} |T(\theta) - \theta|^2 \mathrm{d}\mu_\theta = \sum_{\mathbf{j} \in \mathcal{J}_i^{\mathrm{TO}}} (\gamma_{i,\mathbf{j}} - \iota_{\mathbf{j}})^2,$$

where ι is the vector of coefficients for the identity map. This is of course only true when the polynomial family $\mathcal{P}(\theta)$ is chosen to be orthonormal with respect to μ_θ; however this approximation prevents the map from collapsing onto a Dirac when the expectation is badly approximated by a small number of samples.
These simplifications result in the efficiently implementable, regularised optimisation problem for computing the map coefficients,

$$\arg\min_{\gamma_i \in \mathbb{R}^{M_i}} \frac{1}{2\bar{w}} \gamma_i^\top F_i^\top W F_i \gamma_i - \frac{w^\top}{\bar{w}} \log(G_i\gamma_i) + \beta \|\gamma_i - \iota\|^2, \tag{3.10}$$
$$\text{subject to} \quad G_i\gamma_i > 0,$$

This optimisation problem can be efficiently solved using Newton iterations. It is suggested in [3] that this method usually converges in around 10-15 iterations, and we have seen no evidence that this is not a reasonable estimate. When calculating the map several times during a Monte Carlo run, using previous guesses of the optimal map to seed the Newton algorithm results in much faster convergence, usually taking only a couple of iterations to satisfy the stopping criteria.
The Hessian takes the form

$$HC_i(\gamma_i) = \frac{1}{\bar{w}} \left[F_i^\top W F_i + G_i^\top W \mathrm{diag}([G_i\gamma_i]^{-2}) G_i \right] + \beta I, \tag{3.11}$$

where $[G_i\gamma_i]^{-2}$ is to be taken element-wise, and I is the $M_i \times M_i$ identity matrix. The first derivative of $C_i(T)$ is

$$\nabla C_i(\gamma_i) = \frac{1}{\bar{w}} \left[F_i^\top W F_i \gamma_i - G_i^\top W [G_i\gamma_i]^{-1} \right] + \beta(\gamma_i - \iota),$$

again $[G_i\gamma_i]^{-1}$ is taken element-wise.

Algorithm 2: PAIS algorithm with adaptive transport map. Option 1.

Initialise state $\theta_i^{(1)} = \theta_0, \quad i = 1, \dots, M$.
Initialise map $\bar{\gamma}^{(1)} = \iota$.
for $k \leftarrow 1, \dots, L-1$ **do**
  Compute $r_i = \tilde{T}(\theta_i^{(k)}; \bar{\gamma}^{(k)}), \quad i = 1, \dots, M$.
  Sample $r_i' \sim q_r(\cdot; r_i)$.
  Invert $\hat{\theta}_i^{(k)} = \tilde{T}^{-1}(r_i'; \bar{\gamma}^{(k)})$.
  Calculate:

$$w_i^{(k)} = \frac{\pi(\hat{\theta}_i^{(k)})}{\left(\sum_{j=1}^{M} q_r(r_i'; r_j)\right) |J_{\tilde{T}}(\hat{\theta}_i^{(k)}; \bar{\gamma}^{(k)})|}.$$

  Resample $\theta^{(k+1)} \leftarrow \|w^{(k)}\|^{-1} \sum_{j=1}^{M} w_j^{(k)} \delta_{\hat{\theta}_j^{(k)}}(\cdot)$.
  if $k \ mod \ K_U = 0 \ and \ k < K_{stop}$ **then**
    for $i \leftarrow 1, \dots, n$ **do**
      Solve (3.10) with $\{(w^{(1)}, \hat{\theta}^{(1)}), \dots, (w^{(k+1)}, \hat{\theta}^{(k+1)})\}$ and update $\gamma_i^{(k+1)}$.
  else
    $\bar{\gamma}^{(k+1)} = \bar{\gamma}^{(k)}$.

4. Transport map usage in PAIS and other PMC algorithms. Given importance samples from the target distribution, we have demonstrated how to construct an approximate transport map from the target measure to a reference measure. We now consider how to implement an importance sampling-based MCMC algorithm which uses these maps to propose new states. In [3] it was shown how approximate transport maps can be used to accelerate Metropolis-Hastings methods, with the map being periodically updated with the samples produced from the target measure. Convergence of this adaptation is shown in [3]. In this Section, we will show how similarly, these maps can be used to construct highly efficient importance sampling schemes.

In particular, we will show how we can use the transport map derived in Equation (3.10) to design a proposal scheme for the PAIS algorithm. In this case we have a choice in how to proceed; we propose new samples on reference space and resample on target space, or we both propose and resample on reference space, mapping onto target space to output the samples. The first option allows us to reuse much of the framework from the standard PAIS algorithm and in the numerics later we see that this performs better than both the Transport MH algorithm, and the standard PAIS algorithm. The second option requires some restructuring but results in improved performance from the resampler.

The first option is given in Algorithm 2. We denote the ensembles of states in target space $\theta^{(k)} = \{\theta_1^{(k)}, \dots, \theta_M^{(k)}\}$, and the states in the reference space, $r = \{r_1, \dots, r_M\}$, where M is the ensemble size. Similarly, the proposal states are denoted $r' = \{r_1', \dots, r_M'\}$ and $(w^{(k)}, \hat{\theta}^{(k)}) = \{(w_1^{(k)}, \hat{\theta}_1^{(k)}), \dots, (w_M^{(k)}, \hat{\theta}_M^{(k)})\}$, where these pairs are the states which together form our sample from the target distribution. As in the

Algorithm 3: PAIS algorithm with adaptive transport map. Option 2.

1 Initialise state $\theta_i^{(1)} = \theta_0, \quad i = 1, \dots, M.$
2 Initialise map $\bar{\gamma}^{(1)} = \iota.$
3 **for** $k \leftarrow 1, \dots, N-1$ **do**
4 Compute $r_i = \tilde{T}(\theta_i^{(k)}; \bar{\gamma}^{(k)}), \quad i = 1, \dots, M.$
5 Sample $r_i' \sim q_r(\cdot; r_i).$
6 Invert $\hat{\theta}_i^{(k)} = \tilde{T}^{-1}(r_i'; \bar{\gamma}^{(k)}).$
7 Calculate:

$$w_i^{(k)} = \frac{\pi(\hat{\theta}_i^{(k)})}{\left(\sum_{j=1}^{M} q_r(r_i'; r_j)\right) |J_{\tilde{T}}(\hat{\theta}_i^{(k)}; \bar{\gamma}^{(k)})|}.$$

8 Resample $r^* \leftarrow \|w^{(k)}\|^{-1} \sum_{j=1}^{M} w_j^{(k)} \delta_{r_j'}(\cdot).$
9 Invert $\theta_i^{(k+1)} = \tilde{T}^{-1}(r_i^*).$
10 **if** $k \ mod \ K_U = 0 \ and \ k < K_{stop}$ **then**
11 **for** $i \leftarrow 1, \dots, n$ **do**
12 Solve (3.10) with $\{(w^{(1)}, \hat{\theta}^{(1)}), \dots, (w^{(k+1)}, \hat{\theta}^{(k+1)})\}$ and update $\gamma_i^{(k+1)}$.
13 **else**
14 $\bar{\gamma}^{(k+1)} = \bar{\gamma}^{(k)}.$

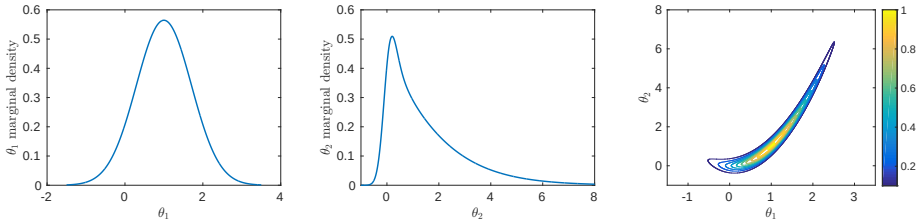
standard version of the PAIS algorithm we use the deterministic mixture weights. The second option, Algorithm 3, is similar to the first except on Line 8 where rather than resampling in target space we resample in reference space. In reference space the dimensions are roughly uncorrelated, and the Gaussian marginals are easy to approximate with fewer ensemble members. This means that the resampling step will be more efficient in moderately higher dimensions, which we discuss in Section 8.1.

5. Convergence of the transport proposal based MCMC algorithms. In this Section we study the convergence of the transport based proposal distributions which we have described in Section 4. We take as a test problem the ubiquitous Rosenbrock banana-shaped density. This target density is given by

$$\pi(\theta) = \frac{\sqrt{10}}{\pi} \exp\left\{-(1-\theta_1)^2 - 10(\theta_2 - \theta_1^2)^2\right\}. \tag{5.1}$$

A contour plot of the target density is given in Figure 5.1. This problem is challenging to sample from since it has a highly peaked and curved ridge, and is often used as a test problem in optimisation and MCMC communities.

5.1. Implementation details. Before looking at the performance of the MCMC algorithms, we demonstrate some properties of the transport maps we will be using in our MCMC algorithms. We draw 1 million samples from the density in (5.1), and use this sample in the framework of Section 3 to build a transport map. We use this map to push forward the original sample onto the reference space, where we will be able to



(a) Marginal density function for θ_1.

(b) Marginal density function for θ_2.

(c) Contour plot for Rosenbrock density.

FIG. 5.1. *Visualisation of the Rosenbrock density as given in Equation* (5.1).

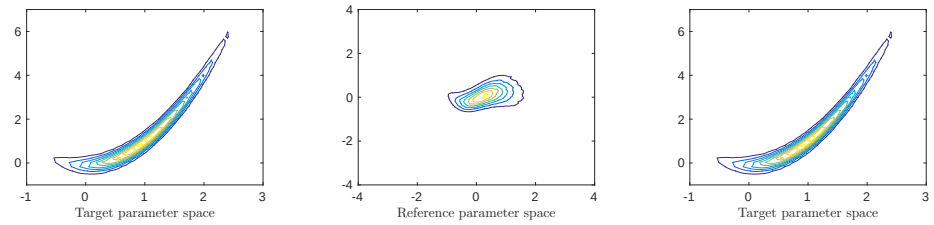
see how well the map has performed at converting the original sample to a standard Gaussian. We then pull the sample back on to target space using the inverse map to check that our map is invertible and well behaved.
For this example, we use an index set of total order 3 with monomial basis functions. It is important that total order is an odd number, since otherwise the map will not be surjective. This results in a map of the form

$$T(\theta_1, \theta_2) = \begin{bmatrix} T_1(\theta_1) \\ T_2(\theta_1, \theta_2) \end{bmatrix},$$

where

$$\begin{aligned} T_1(\theta_1) &= \gamma_{1,1} + \gamma_{1,2}\theta_1 + \gamma_{1,3}\theta_1^2 + \gamma_{1,4}\theta_1^3, \\ T_2(\theta_1, \theta_2) &= \gamma_{2,1} + \gamma_{2,2}\theta_1 + \gamma_{2,3}\theta_1^2 + \gamma_{2,4}\theta_1^3 + \gamma_{2,5}\theta_2 + \gamma_{2,6}\theta_1\theta_2 \\ &\quad + \gamma_{2,7}\theta_1^2\theta_2 + \gamma_{2,8}\theta_2^2 + \gamma_{2,9}\theta_1\theta_2^2 + \gamma_{2,10}\theta_2^3. \end{aligned}$$

Clearly even with only basis functions of total order 3, we have a large number of unknowns in our optimisation problem, $\bar{\gamma} \in \mathbb{R}^{14}$. If we were to increase the dimension of θ further we would need to reduce the number of terms we include in the expansion by, for example, removing all the "cross" terms. This reduces the quality of our map but since we only require an approximate map we can afford to reduce the accuracy.



(a) Original sample θ from RWMH algorithm.

(b) Push forward of θ onto reference space.

(c) Pull back of reference sample onto target space.

FIG. 5.2. *Rosenbrock target density as described in Equation* (5.1).

Figure 5.2 shows the output of the approximate transport map. Even though we have truncated the infinite expansion in the monomial basis down to 4 and 10 terms in respective dimensions, the push forward of the sample is still a unimodal distribution

centred at the origin with standard deviation 1. As you move out into the tails of the reference density more non-Gaussian features are clearly visible. However, overall, the push forward of the target density does not look a challenging one to sample from, with even relatively simple MCMC methods such as RWMH. The pullback from reference space, in Figure 5.2, is an exact match of the original sample since we have not perturbed the sample in reference space. This inversion is well defined in the sampling region, although not necessarily outside [3].

5.2. Numerical results for convergence of transport map based algorithms on the Rosenbrock density. We first find the optimal scaling parameters for the individual algorithms. This is done, as in ****CITE US**** by optimising for the effective sample size in the PAIS algorithm, and by tuning the relative L^2 error in the MH algorithm. There is currently no guidance on the best way of tuning the MH algorithm with transport map proposals although one might expect results similar to the standard MH results, especially if adaptation of the map is stopped after a given point. As in the PAIS algorithm, optimising for the effective sample size might be the best option.

Statistic / Algorithm	Transport M-H	Alg. 2	Alg. 3
δ_{L^2}	1.0e-0	1.1e-1	3.5e-1
δ_{ESS}	-	1.0e-1	5.2e-1
Acc. rate	0.23	-	-
ESS ratio	-	0.62	0.71

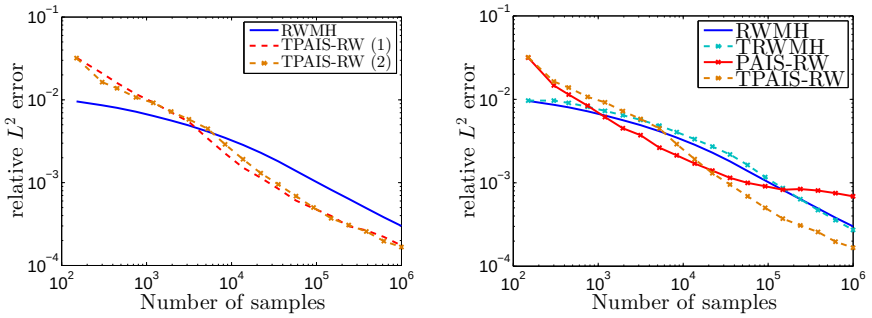
TABLE 5.1
Optimal scaling parameters for the transport map based algorithms applied to R_1.

The optimal scaling parameters are given in Table 5.1. Here we see that the effective sample size is much lower than we see in the one-dimensional examples with the PAIS algorithms. However, in the Rosenbrock density (5.1) we are dealing with a much more complicated correlation structure, as well as a very slowly decaying tail in θ_2. From our experiments, we have observed that the standard PAIS-RW required an ensemble size of $M = 500$ to overcome the problems in this density, however the transport map transforms the tails to be more like those of a Gaussian which can be approximated well by a smaller ensemble size of $M = 150$.
The convergence of the three algorithms is displayed in Figure 5.3. Figure (a) shows that the two variations of the transport based PAIS algorithms converge with similar rates. The second version, which performs the resampling stage in reference space rather than target space, has a slightly higher ESS, and is more stable than option (1). This version also has a property that we can exploit in Section 8.1.

6. Multiscale Methods for Stochastic Chemical Reaction Networks. In this Section, we discuss some recent advances in multiscale methods for stochastic reaction networks. Inverse problems arising in this area often lead to highly correlated and complex posterior distributions, which traditional MCMC methods can struggle to sample from. We will then go on to solve some inverse problems related to this in Section 7.

7. Numerical Examples.

8. Discussion.



(a) Comparison of the two Transport PAIS options.

(b) Comparison of Transport PAIS option (2) with the standard algorithms.

FIG. 5.3. *Convergence of Algorithms Transport M-H, 2, 3 for density* (5.1). *Ensemble size* $M = 150$*, resampling performed using the AMR algorithm.*

8.1. Sampling in (moderately) higher dimensions. One major problem with importance sampling schemes is the curse of dimensionality, which means that methods such as PAIS, and other PMC methods, can only be used for relatively low dimensional problems. Here, we will briefly discuss how transport maps could aid with making moderately higher dimensional problems accessible to this family of methods. Algorithm 3 allows us to decorrelate the dimensions of our random parameter on reference space, where we then can resample and map the resulting ensemble back onto target space. Since, on reference space, the dimensions are uncorrelated, we are able to resample in each dimension separately. Resampling in a single dimension allows for optimisations in resampling code, and also means that the resampler is not affected by the curse of dimensionality.

If we can approximate the posterior well with our mixture and with the transport map, we should not be affected by the increase in dimension to the extent we have been with the standard PAIS-RW algorithm. In one dimension the ETPF algorithm can be implemented very efficiently. As described in [5], the coupling matrix has all non-zero entries in a staircase pattern when the state space is ordered. We can exploit this knowledge to produce Algorithm 4. Which is much faster than using the simplex algorithm to minimise the associated cost function, and faster than the AMR algorithm****CITE OUR OTHER PAPER****.

REFERENCES

[1] T. El Moselhy and Y. Marzouk, *Bayesian inference with optimal maps*, Journal of Computational Physics, 231 (2012), pp. 7815–7850.

[2] M. Parno, *Transport maps for accelerated Bayesian computation*, PhD thesis, Massachusetts Institute of Technology, 2015.

[3] M. Parno and Y. Marzouk, *Transport map accelerated Markov chain Monte Carlo*, arXiv preprint arXiv:1412.5492, (2014).

[4] M. Pinsker, *Information and information stability of random variables and processes*, (1960).

[5] S. Reich, *A nonparametric ensemble transform method for Bayesian inference*, SIAM Journal on Scientific Computing, 35 (2013), pp. A2013–A2024.

Algorithm 4: ETPF algorithm in one dimension.

Sort the states, $\{(w_i, x_i)\}_{i=1}^M$, into ascending order.
Normalise the weights $p_i = w_i / \|w\|_1$.
Set $y_i \leftarrow 0$ for all $i = 1, \ldots, M$.
Set $c \leftarrow 0$
for $i \leftarrow 1, \ldots, M$ **do**
    Set $t \leftarrow p_i$
    while $j \leq M$ *and* $t > 0$ **do**
        Set $s \leftarrow \left(M^{-1} - c\right) \wedge t$
        Increase y_j by $M \times s \times x_i$.
        Decrease t by s.
        Increase c by s.
        if $t > 0$ **then**
            Increase j by 1.
            Set $c \leftarrow 0$.
Return y.