Contents

List of Figures

columnwidth 421.10046pt

1 Introduction

1.1 What is going on?, 3 facts, What is new in this thesis?

- hierachical Bayesian model, sampling to TT approx
- RTE as an example
- nonLinear to Linear Affine funciton (affine RTO)

1.2 Thesis Outline

2

Theoretical and Technical Background

2.1 Forward Model

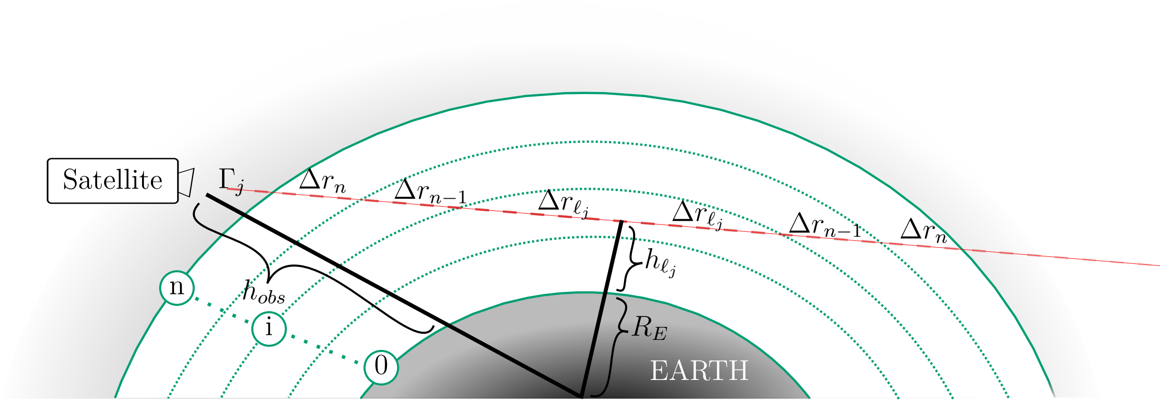

Figure 2.1: text

In this section we describe the forward model which we use to simulate data, as in section **??** and base the Bayesian inference, see section **??** and **??**.

As shown in Figure ??, one measurement of a stationary satellite can be describes as the path integral along the line of sight Γ_j for $j = 1, 2, \dots, m$. For each measurement we can define a tangent height h_{ℓ_j} as the shortest distance along the line of sight to the earth.

The j^{th} measurement, taken on line of sight Γ_j is modelled by the radiative transfer equation (RTE) [1]

$$y_j = \int_{\Gamma_j} B(\nu,T)k(\nu,T)\frac{p(T)}{k_{\mathrm{B}}T(r)}x(r)\tau(r)\mathrm{d}r + \eta_j \tag{2.1}$$

$$\tau(r) = \exp\left\{-\int_{r_{\mathrm{obs}}}^{r} k(\nu,T)\frac{p(T)}{k_B T(r')}x(r')\mathrm{d}r'\right\} \tag{2.2}$$

where the path from the satellite along the line-of-sight of the j^{th} pointing direction is Γ_j and the ozone concentration at distance r from the radiometer is $x(r)$ plus some noise η_j. Within the stratosphere the number density $p(T)/(k_{\mathrm{B}}T(r))$ of molecules is dependent on the pressure $p(T)$, the temperature $T(r)$, and the Boltzmann constant k_{B}. The factor $\tau(r) \leq 1$ accounts for re-absorption of the radiation along the line-of-sight, which makes the RTE non linear. The absorption constant $k(\nu,T)$ for a single gas molecule at a specific wavenumber ν is given by the HITRAN database [2] and acts as a source function when multiplied with the black body radiation $B(\nu,T)$, given by Planck‘s law [3]. For fundamentals on the Radiative transfer equation we recommend Chapter one in [3].

To enable Matrix-Vector multiplication, we parametrise the ozone profile as a function of height, discretised into the n values in each of n layers of the discretised stratosphere where the i^{th} layer is defined by two spheres of radii $h_{i-1} < h_i$, $i = 1,\dots,n$, with h_0 and h_n. In between the heights h_{i-1} and h_i, each of the ozone concentration x_i, the pressure p_i, the temperature T_i, and thermal radiation is assumed to be constant. Above h_n and below h_0, the ozone concentration is set to zero, so no signal can be obtained. Then depending on the parameter of interest, which is either the ozone volume mixing ratio $\boldsymbol{x} = \{x_1, x_2, \dots, x_n\} \in \mathbb{R}^n$ or the fraction of pressure and temperature $\boldsymbol{p/T} = \{p_1/T_1, p_2/T_2, \dots, p_n/T_n\} \in \mathbb{R}^n$, we can rewrite the integral in Eq. (2.1) as e.g. as a vector multiplication $\boldsymbol{A_j}(\boldsymbol{x},\boldsymbol{p},\boldsymbol{T})\,\boldsymbol{x}$, where the non-linear absorption $\tau(r)$ is included in $\boldsymbol{A_j}(\boldsymbol{x},\boldsymbol{p},\boldsymbol{T})$. Here, the row vector $\boldsymbol{A_j}(\boldsymbol{x},\boldsymbol{p},\boldsymbol{T}) \in \mathbb{R}^n$ defines a Kernel for each measurement so that the data vector

$$\boldsymbol{y} = \boldsymbol{A}(\boldsymbol{x},\boldsymbol{p},\boldsymbol{T})\,\boldsymbol{x} + \boldsymbol{\eta} = \boldsymbol{A}(\boldsymbol{x},\boldsymbol{p},\boldsymbol{T})\,\frac{\boldsymbol{p}}{\boldsymbol{T}} + \boldsymbol{\eta}\,. \tag{2.3}$$

can be written as a matrix vector multiplication, where the matrix $\boldsymbol{A}(\boldsymbol{x},\boldsymbol{p},\boldsymbol{T}) \in \mathbb{R}^{m\times n}$ and the noise vector $\boldsymbol{\eta} \in \mathbb{R}^m$.

Since the measurement process includes absorption $\tau(r)$ reducing measurements slighty and making the inverse problem only weakly non-linear. We use that to approximate the non-linear forward model $\boldsymbol{A}(\boldsymbol{x},\boldsymbol{p},\boldsymbol{T})$ with a map $\boldsymbol{M}$ so that $\boldsymbol{A}(\boldsymbol{x},\boldsymbol{p},\boldsymbol{T}) \approx \boldsymbol{M}\boldsymbol{A}_L$ Where each row $\boldsymbol{A}_{L,j}$ of matrix as $\boldsymbol{A}_L \in \mathbb{R}^{m\times n}$ is defined by the linear forward model, where absorption is neglected, e.g. $\tau = 1$. Then each entry in the row vector $\boldsymbol{A}_{L,j}$ is either defined by $B(\nu,T)S(\nu,T)\frac{\boldsymbol{p}(T)}{k_{\mathrm{B}}\boldsymbol{T}(r)}\mathrm{d}r$ or $B(\nu,T)S(\nu,T)\frac{\boldsymbol{x}}{k_{\mathrm{B}}}\mathrm{d}r$, as in Eq.. (2.1), depending on the parameter of interest. This poses a linear inverse problem with the forward map defined by the matrix $\boldsymbol{A} = \boldsymbol{M}\boldsymbol{A}_L$, where $\boldsymbol{M}$ is, more specifically, an affine map.

2.2 Affine Map

To approximate the non-linear forward model we use an affine map $M : \boldsymbol{A}_L\boldsymbol{x} \rightarrow \boldsymbol{A}(\boldsymbol{x},\boldsymbol{p},\boldsymbol{T})\boldsymbol{x}$, which maps the linear forward model $\boldsymbol{A}_L\boldsymbol{x}$ onto the non-linear forward model $\boldsymbol{A}(\boldsymbol{x},\boldsymbol{p},\boldsymbol{T})\boldsymbol{x}$.

An affine map is any linear map in between two vector spaces or affine spaces, where an affine space does not need to preserve a zero origin, see Def. 2.3.1. in [4]. In other words an affine map does not need to map to the origin of the associated vector space, or is a linear map on vector spaces including translation, or in the words of my supervisor, C. F., an affine map is a Taylor series of first order. For more information on affine spaces and maps we refer to the books [4, 5]

To find the affine map, we generate two affine subspaces spaces $V = \left\{\boldsymbol{A}(\boldsymbol{x}^{(1)},\boldsymbol{p},\boldsymbol{T}),\dots,\boldsymbol{A}(\boldsymbol{x}^{(m)},\boldsymbol{p},\boldsymbol{T})\right\}$ and $W = \left\{\boldsymbol{A}_L\boldsymbol{x}^{(1)},\dots,\boldsymbol{A}_L\boldsymbol{x}^{(m)}\right\}$ over the same field, with fixed $\boldsymbol{p},\boldsymbol{T}$. Then we can use a linear solver to find the affine map $\boldsymbol{M}$ so that we can approximate the non linear forward model $\boldsymbol{A}(\boldsymbol{x},\boldsymbol{p},\boldsymbol{T}) \approx \boldsymbol{M}\boldsymbol{A}_L$ with a linear forward model, we go into further detail in section **??**. The parameter $\boldsymbol{x}$ is distributed as the so-called posterior distribution $\left\{\boldsymbol{x}^{(1)},\dots,\boldsymbol{x}^{(m)}\right\} \sim \pi(\boldsymbol{x}|\boldsymbol{\theta},\boldsymbol{y})$ conditioned on the hyper-parameters $\boldsymbol{\theta}$, according to a Bayesian hierarchical model.

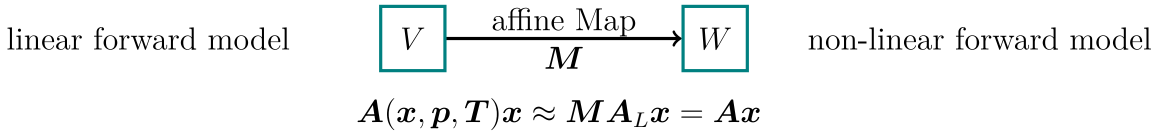
linear forward model
V
affine Map
$\boldsymbol{M}$
W
non-linear forward model
$\boldsymbol{A}(\boldsymbol{x},\boldsymbol{p},\boldsymbol{T})\boldsymbol{x} \approx \boldsymbol{M}\boldsymbol{A}_L\boldsymbol{x} = \boldsymbol{A}\boldsymbol{x}$

Figure 2.2: Schematics of Affine Map, which approximates the linear forward model to the non-linear forward model.

2.3 Bayesian Inference

In this this section we give a short introduction to Bayesian inference for a general parameter $\boldsymbol{x}$ given some data

$$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{\eta} \tag{2.4}$$

based on a linear forward model $\boldsymbol{A}$ and some noise $\boldsymbol{\eta}$. Later in section **??** we set up a more sophisticated Bayesian framework applied to the forward model in section **??**.

We can visualise the correlation structure of a measurement process through a hierarchiallly ordered directed acyclic graph (DAG), see Figure 2.3. As an observatory process naturally includes some random noise we include that in our DAG and classify the noise as a hyper-parameter in $\boldsymbol{\theta}$ [6]. Other hyper-parameters influence the parmeters $\boldsymbol{x}$ detemernistaclly, which are then mapped through the forward model onto the space of all measurables $\boldsymbol{u}$, from which we observe some data $\boldsymbol{y}$ including noise as previously mentioned. Drawing a DAG can help us to dependences within the measurement and modelling process. Given some data we inferer the distribution of the underling parameters and hyper-parameters by following the arrows in Figure 2.3 backwards and set up a Bayesian model, hierarchially ordered.

Within a linear Bayesian hierarchial model we need to define a likelihood function as well as a distribution over the unknown parameters $\boldsymbol{x}$ and hyper-parameters $\boldsymbol{\theta}$.

$$\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta} \sim \mathcal{N}(\boldsymbol{A}\boldsymbol{x}, \boldsymbol{\Sigma}(\boldsymbol{\theta})) \tag{2.5a}$$

$$\boldsymbol{x}|\boldsymbol{\theta} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{Q}^{-1}(\boldsymbol{\theta})) \tag{2.5b}$$

$$\boldsymbol{\theta} \sim \pi(\boldsymbol{\theta})\,. \tag{2.5c}$$

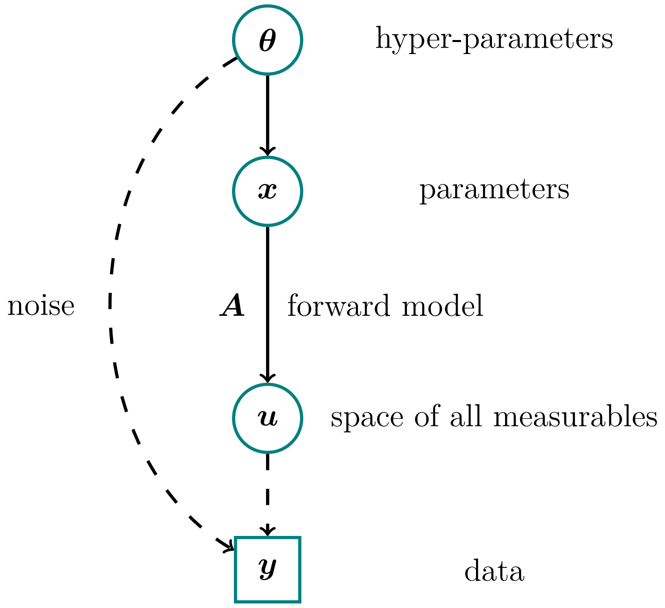

Figure 2.3: The directed acyclic graph (DAG) for a typical linear inverse problem visualises forward dependencies as solid line arrows for deterministic dependencies and dotted arrows for statistical dependencies. Naturally the data $\boldsymbol{y}$ has some noise described through included in some hyper-parameters $\boldsymbol{\theta}$. The parameters $\boldsymbol{x}$ have some dependency of those hyper-parameters $\boldsymbol{\theta}$. The parameter $\boldsymbol{x}$ is mapped onto the space of all measurables $\boldsymbol{u}$ through the linear forward model $\boldsymbol{A}$, so that $\boldsymbol{Ax}$ is a linear operation. From the space of all measurables we can observe some data $\boldsymbol{y}$, statistically, where as prevoiusly mentioned some random noise is added. We set up a more sophisticated Bayesian model in chapter **??** explicitly including all hyper-parameters and parameters of interest according to the forward model in section **??**.

Due to the assumption of Gaussian noise $\boldsymbol{\eta} \sim \mathcal{N}(0, \boldsymbol{\Sigma}(\boldsymbol{\theta}))$ the likelhood function $\pi(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{\theta})$, which includes information about the measurement process and gives a measure of how well parametersand hyperparemters fit given some data, is normally distributed with the noise covariance matrix $\boldsymbol{\Sigma}(\boldsymbol{\theta})$. More specifically because we choose a normally distributed prior $\pi(\boldsymbol{x}|\boldsymbol{\theta})$ with the prior precision matrix $\boldsymbol{Q}(\boldsymbol{\theta})$ and prior mean $\boldsymbol{\mu}$ this is a linear-Gaussian Bayesian hierarchical model as in [6], including some distribution over the hyper-parameters $\pi(\boldsymbol{\theta})$. One of the main strengths of a Bayesian framework is that through those prior and hyper-prior distributions we can incorporate functional dependencies as well model physical properties of the parameters $\boldsymbol{x}$.

Then Bayes Theorem gives the posterior distribution, the function of interest,

$$\pi(\boldsymbol{x}, \boldsymbol{\theta}|\boldsymbol{y}) = \frac{\pi(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{\theta})\pi(\boldsymbol{x}, \boldsymbol{\theta})}{\pi(\boldsymbol{y})}\,, \tag{2.6}$$

with the prior distribution $\pi(\boldsymbol{x}, \boldsymbol{\theta}) = \pi(\boldsymbol{x}|\boldsymbol{\theta})\pi(\boldsymbol{\theta})$ and the normalising constant $\pi(\boldsymbol{y})$. If the normalising constant is finite and non-zero we can approximate the posterior distribution

$$\pi(\boldsymbol{x}, \boldsymbol{\theta}|\boldsymbol{y}) \propto \pi(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{\theta})\pi(\boldsymbol{x}, \boldsymbol{\theta}) \,. \tag{2.7}$$

and the expectation of any a function $h(\boldsymbol{x}(\boldsymbol{\theta}))$, where $\boldsymbol{x}$ may be depending on $\boldsymbol{\theta}$, can be described as

$$\mathrm{E}_{\boldsymbol{x},\boldsymbol{\theta}|\boldsymbol{y}}[h(\boldsymbol{x}(\boldsymbol{\theta}))] = \int\int h(\boldsymbol{x}(\boldsymbol{\theta}))\,\pi(\boldsymbol{x}, \boldsymbol{\theta}|\boldsymbol{y})\,\mathrm{d}\boldsymbol{x}\,\mathrm{d}\boldsymbol{\theta} \,, \tag{2.8}$$

which is usually a high dimensional integral and computationally not feasible to solve.

One way to work around the high dimensionality is to parameterise $\boldsymbol{x}$ using hyper-parameters $\boldsymbol{\theta}$ so that $\boldsymbol{x}(\boldsymbol{\theta})$. Another way is to seperate the posterior distribution over latent field $\boldsymbol{x}$ and the hyper-parameters $\boldsymbol{\theta}$. This is particular benefitial, when $\boldsymbol{x}$ is high dimensional, e.g. $\boldsymbol{x} \in \mathbb{R}^n$ with $n = 45$ and can not be parametrised, and $\boldsymbol{\theta}$ is low dimensional, e.g. two dimensional. Then by the law of total expectation eq. 2.8 becomes

$$\mathrm{E}_{\boldsymbol{x}|\boldsymbol{y}}[h(\boldsymbol{x})] = \mathrm{E}_{\boldsymbol{\theta}|\boldsymbol{y}}\Big[\mathrm{E}_{\boldsymbol{x}|\boldsymbol{\theta},\boldsymbol{y}}[h(\boldsymbol{x}(\boldsymbol{\theta}))]\Big] = \int \mathrm{E}_{\boldsymbol{x}|\boldsymbol{\theta},\boldsymbol{y}}[h(\boldsymbol{x}(\boldsymbol{\theta}))]\,\pi(\boldsymbol{\theta}|\boldsymbol{y})\,\mathrm{d}\boldsymbol{\theta} \,, \tag{2.9}$$

where in the case of a linear-Gaussian Bayesian hierarchical model $\mathrm{E}_{\boldsymbol{x}|\boldsymbol{\theta},\boldsymbol{y}}[h(\boldsymbol{x}(\boldsymbol{\theta}))]$ is well defined. The marginal and then condtional method, does exactly that.

2.3.1 Marginal and then Conditional

The marginal and then conditional (MTC) method factorises the full posterior distribution

$$\pi(\boldsymbol{x}, \boldsymbol{\theta}|\boldsymbol{y}) = \pi(\boldsymbol{x}|\boldsymbol{\theta}, \boldsymbol{y})\pi(\boldsymbol{\theta}|\boldsymbol{y}) \tag{2.10}$$

into the marginal posterior distribution $\pi(\boldsymbol{\theta}|\boldsymbol{y})$ and conditional posterior distribution $\pi(\boldsymbol{x}|\boldsymbol{\theta}, \boldsymbol{y})$.

For the in Eq. 2.5 specified linear-Gaussian Bayesian hierarchical model the marginal posterior distribution is given as

$$\pi(\boldsymbol{\theta}|\boldsymbol{y}) = \int \pi(\boldsymbol{x},\boldsymbol{\theta}|\boldsymbol{y})\,\mathrm{d}\boldsymbol{x} \tag{2.11}$$

$$\propto \sqrt{\frac{\det(\boldsymbol{\Sigma}^{-1})\,\det(\boldsymbol{Q})}{\det(\boldsymbol{Q}+\boldsymbol{A}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{A})}} \times \exp\Big[-\frac{1}{2}(\boldsymbol{y}-\boldsymbol{A}\boldsymbol{\mu})^T\boldsymbol{Q}_{\boldsymbol{\theta}|\boldsymbol{y}}(\boldsymbol{y}-\boldsymbol{A}\boldsymbol{\mu})\Big]\pi(\boldsymbol{\theta})\,, \tag{2.12}$$

with

$$\boldsymbol{Q}_{\boldsymbol{\theta}|\boldsymbol{y}} = \boldsymbol{\Sigma}^{-1} - \boldsymbol{\Sigma}^{-1}\boldsymbol{A}(\boldsymbol{A}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{A}+\boldsymbol{Q})^{-1}\boldsymbol{A}^T\boldsymbol{\Sigma}^{-1}\,, \tag{2.13}$$

see Lemma 2 in [6]. Conditioned on the hyper-parameters $\boldsymbol{\theta}$ we can draw samples from the conditional posterior distribution

$$\boldsymbol{x}|\boldsymbol{\theta},\boldsymbol{y} \sim \mathcal{N}\Big(\underbrace{\boldsymbol{\mu}+(\boldsymbol{A}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{A}+\boldsymbol{Q})^{-1}\boldsymbol{A}^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{y}-\boldsymbol{A}\boldsymbol{\mu})}_{\boldsymbol{\mu}_{\boldsymbol{x}|\boldsymbol{\theta},\boldsymbol{y}}},\underbrace{(\boldsymbol{A}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{A}+\boldsymbol{Q})^{-1}}_{\boldsymbol{\Sigma}_{\boldsymbol{x}|\boldsymbol{y},\boldsymbol{\theta}}}\Big) \tag{2.14}$$

using the Randomis then optimize (RTO) method, see section 2.5.2, or calculate weighted expectations e.g. $\mathrm{E}_{\boldsymbol{x}|\boldsymbol{\theta},\boldsymbol{y}}[\boldsymbol{x}|\boldsymbol{\theta},\boldsymbol{y}] = \boldsymbol{\mu}_{\boldsymbol{x}|\boldsymbol{\theta},\boldsymbol{y}}$ or use Eq. 2.9 to calculate weighted expectations of $\mathrm{E}_{\boldsymbol{x}|\boldsymbol{\theta},\boldsymbol{y}}[\mathrm{Var}(\boldsymbol{x}|\boldsymbol{\theta},\boldsymbol{y})] = \boldsymbol{\Sigma}_{\boldsymbol{x}|\boldsymbol{y},\boldsymbol{\theta}}$ with weights given by $\pi(\boldsymbol{\theta}|\boldsymbol{y})$. Note that the noise covariance $\boldsymbol{\Sigma} = \boldsymbol{\Sigma}(\boldsymbol{\theta})$ and the prior precision $\boldsymbol{Q} = \boldsymbol{Q}(\boldsymbol{\theta})$ are depending on hyper-parameters $\boldsymbol{\theta}$.

2.4 Regularisation

Another method to find a solution to a linear inverse problem as in Eq. 2.4 is to find a solution $\boldsymbol{x}_\lambda$ according to a data misfit norm and a regularisation semi-norm as in [6]. We will discuss the case of Tikhonov regularisation [7, 8] as this is the most similar to a linear-Gaussian hierarchial Bayesian model.

For a parameter $\boldsymbol{x}$ a linear forward model matrix $\boldsymbol{A}$ and some data $\boldsymbol{y}$ the data misfit norm

$$\|\boldsymbol{y}-\boldsymbol{A}\boldsymbol{x}\| \tag{2.15}$$

gives a measure of how good data fits to a mapped parameter $\boldsymbol{Ax}$ and the regularisation semi norm

$$\lambda \left\|\boldsymbol{Tx}\right\| \tag{2.16}$$

penalises $\boldsymbol{x}$ according to $\boldsymbol{T}$ and the regularisation parameter $\lambda > 0$. Given λ a regularised solution

$$\boldsymbol{x}_\lambda = \arg\min_{\boldsymbol{x}} \left\|\boldsymbol{y} - \boldsymbol{Ax}\right\|^2 + \lambda \left\|\boldsymbol{Tx}\right\|^2 \tag{2.17}$$

can be found by the derivativ

$$\begin{aligned}
& \nabla_{\boldsymbol{x}} \Big\{ (\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}_\lambda)^T (\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}_\lambda) + \lambda \boldsymbol{x}_\lambda^T \boldsymbol{T}^T \boldsymbol{T} \boldsymbol{x}_\lambda \Big\} = 0 && (2.18)\\
\iff \quad & \nabla_{\boldsymbol{x}} \Big\{ \boldsymbol{y}^T \boldsymbol{y} + \boldsymbol{x}_\lambda^T \boldsymbol{A}^T \boldsymbol{A} \boldsymbol{x}_\lambda - \boldsymbol{y}^T \boldsymbol{A} \boldsymbol{x}_\lambda - \boldsymbol{x}_\lambda^T \boldsymbol{A}^T \boldsymbol{y} + \lambda \boldsymbol{x}_\lambda^T \boldsymbol{T}^T \boldsymbol{T} \boldsymbol{x}_\lambda \Big\} = 0 && (2.19)\\
\iff \quad & 2\boldsymbol{A}^T \boldsymbol{A} \boldsymbol{x}_\lambda - 2\boldsymbol{A}^T \boldsymbol{y} + 2\lambda \boldsymbol{T}^T \boldsymbol{T} \boldsymbol{x}_\lambda = 0. && (2.20)
\end{aligned}$$

Then a regularised solution is given as:

$$(\boldsymbol{A}^T \boldsymbol{A} + \lambda \boldsymbol{L})^{-1} \boldsymbol{A}^T \boldsymbol{y} = \boldsymbol{x}_\lambda, \tag{2.21}$$

where we can set $\boldsymbol{T}^T \boldsymbol{T} = \boldsymbol{L}$, which is typically matrix approximation of the nth derivative [8]. In practise $\boldsymbol{x}_\lambda$ is calculated for a range of λ, and is evaluated by the data-misfit norm with respect to the regularised semi-norm so that the best $\boldsymbol{x}_\lambda$ lays on the point of maximum curvature of a so-called L-Curve [9], which we will show in section **??**.

2.5 Sampling Methods

In this chapter we present the sampling methods used in this thesis and also argue why we can use sampling methods to calculate the expectation

$$\mathrm{E}_{\boldsymbol{x},\boldsymbol{\theta}|\boldsymbol{y}}[h(\boldsymbol{x}(\boldsymbol{\theta}))] = \underbrace{\int\int h(\boldsymbol{x}(\boldsymbol{\theta}))\pi(\boldsymbol{x},\boldsymbol{\theta}|\boldsymbol{y})\,\mathrm{d}\boldsymbol{\theta}\,\mathrm{d}\boldsymbol{x}}_{\boldsymbol{\mu}_{\text{int}}} \tag{2.22}$$

of a function $h(\boldsymbol{x}(\boldsymbol{\theta}))$ with respect to a probability density $\pi(\boldsymbol{x}, \boldsymbol{\theta}|\boldsymbol{y})$. In the case where the calculation of the integral is not feasible we can approximate Eq. 2.22 with a sample based estimate

$$\mathrm{E}_{\boldsymbol{x},\boldsymbol{\theta}|\boldsymbol{y}}[h(\boldsymbol{x}(\boldsymbol{\theta}))] \approx \underbrace{\frac{1}{N}\sum_{k=1}^{N} h(\boldsymbol{x}^{(k)}(\boldsymbol{\theta}^{(k)}))}_{\boldsymbol{\mu}_{\text{samp}}}\,, \tag{2.23}$$

for large enough samples size N of a sample set $\mathcal{M} = \{(\boldsymbol{x}, \boldsymbol{\theta})^{(1)}, \ldots, (\boldsymbol{x}, \boldsymbol{\theta})^{(k)}, \ldots, (\boldsymbol{x}, \boldsymbol{\theta})^{(N)}\} \sim \pi(\boldsymbol{x}, \boldsymbol{\theta}|\boldsymbol{y})$. The variance of this unbiased estimator is $\mathcal{O}(1/N)$ so large samples sizes can reduce the uncertainty of $\boldsymbol{\mu}_{samp}$ [10]. We can do this as the central limit theorem states that the samples means $\boldsymbol{\mu}^{(i)}_{samp}$, of samples sets $\mathcal{M}_i$ for $i = 1, \ldots, n$ of any distribution , converge in distribution to a normal distribution so that

$$\sqrt{n}(\boldsymbol{\mu}^{(i)}_{\text{samp}} - \boldsymbol{\mu}_{\text{int}}) \xrightarrow{\mathcal{D}} \mathcal{N}(0, \sigma^2)[11], \tag{2.24}$$

and if $\sigma^2 < \infty$ the error $\boldsymbol{\mu}^{(i)}_{\text{samp}} - \boldsymbol{\mu}_{\text{int}}$ is bound.

Now, we have to show that the methods we use actually draw samples $\mathcal{M} = \{(\boldsymbol{x}, \boldsymbol{\theta})^{(1)}, \ldots, (\boldsymbol{x}, \boldsymbol{\theta})^{(k)}, \ldots, (\boldsymbol{x}, \boldsymbol{\theta})^{(N)}\} \sim \pi(\boldsymbol{x}, \boldsymbol{\theta}|\boldsymbol{y})$ from the desired target distribution so that we can use sample based estimates as in Eq. 2.23. In doing so we will use Markov-Chain Monte Carlo (MCMC) methods, where ergodicity of a Markov-Chain $\mathcal{M}$ is a sufficient criterium to use samples based estimates [8, 10]. The ergodicy theorem in [8] states that, if an aperiodic and irreducible Markov chain $\mathcal{M}$ is reversible then it converges towards a stationary unique equilibirum distribution $\pi(\boldsymbol{x}, \boldsymbol{\theta}|\boldsymbol{y})$. In other words if from any state in the chain we can reach any other state in the sampling space and the previous state, and we do not get stuck in periodic loop, then the chain converges towards a stationary distribution. In practise one can look at the trace $\pi(\boldsymbol{x}^{(k)}, \boldsymbol{\theta}^{(k)}|\boldsymbol{y})$ for $k = 1, \ldots, N$ of the samples and eyeball ergodicity.

The sampling methods used in this thesis have proven ergodic properties, so we will cite and refer the reader to the respective documents.

2.5.1 Metropolis- within Gibbs sampling

As introduced in section 2.3.1 when using the MTC method will sample seperatly from $\pi(\boldsymbol{\theta}|\boldsymbol{y})$ and $\pi(\boldsymbol{x}|\boldsymbol{\theta}, \boldsymbol{y})$. To sample from $\pi(\boldsymbol{\theta}|\boldsymbol{y})$ we use a Metropolis-within-Gibbs sampler as in [6] and discuss the 2 dimesnional case, as used in this thesis, with $\boldsymbol{\theta} = (\theta_1, \theta_2)$, where we do a metroplis step in θ_1 deirection and a gibbs stepp in θ_2 direction. Ergodicity is proven here [12].

The Metropolis-within-Gibbs algorithm starts with a initial guess $\boldsymbol{\theta}^{(t)}$ at $t = 0$. Then, we propose a new sample $\theta_1 \sim q(\theta_1|\theta_1^{(t-1)})$ conditioned on the previous state according to a symmetric proposal distribution $q(\theta_1|\theta_1^{(t-1)}) = q(\theta_1^{(t-1)}|\theta_1)$, which is a special case of the Metropolis-Hastings algorithm [12] and cancels when computing the acceptance probability α. We accept and set $\theta_1^{(t)} = \theta_1$ with

$$\alpha(\theta_1|\theta_1^{(t-1)}) = \min \left\{ 1, \frac{\pi(\theta_1|\theta_2^{(t-1)}, \boldsymbol{y}) q(\theta_1^{(t-1)}|\theta_1)}{\pi(\theta_1^{(t-1)}|\theta_2^{(t-1)}, \boldsymbol{y}) q(\theta_1|\theta_1^{(t-1)})} \right\} \tag{2.25}$$

or reject and keep $\theta_1^{(t)} = \theta_1^{(t-1)}$, which we do by comparing α to a uniform random number $u \sim \mathcal{U}(0,1)$. Next, we do a Gibbs step in θ_2 direction, where Gibbs sampling is a special case of the Metropolis-Hastings algorithm with acceptance probaility of 1, and draw the next sample $\theta_2^{(t)} \sim \pi(\cdot|\theta_1^{(t)}, \boldsymbol{y})$ conditioned on $\theta_1^{(t)}$ at step t. We reapet this N times and assure convergence independent of the intial sample (irreducibility) as we discard samples after the so-called burn-in period so that we produce a Markov-Chain of length $N - N_{\text{burn-in}}$.

2.5.2 Draw a sample from a multivariate normal distribution

after sampling from $\pi(\boldsymbol{\theta}|\boldsymbol{y})$ we draw samples from $\pi(\boldsymbol{x}|\boldsymbol{\theta}, \boldsymbol{y})$ within the MTC scheme. For Linear Gaussian Bayesian hierarchical model we can draw a sample $\boldsymbol{x}$ from the multivariate normal distribution$\pi(\boldsymbol{x}|\boldsymbol{\theta}, \boldsymbol{y})$ using the radomize then optimise (RTO) method [13].

In doing so we can rewrite the full conditional normal distribution $\pi(\boldsymbol{x}|\boldsymbol{y}, \boldsymbol{\theta})$ to:

$$\pi(\boldsymbol{x}|\boldsymbol{y}, \boldsymbol{\theta}) \propto \pi(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{\theta}) \pi(\boldsymbol{x}|\boldsymbol{\theta}) \tag{2.26}$$

$$= \exp \| \hat{\boldsymbol{A}} \boldsymbol{x} - \hat{\boldsymbol{y}} \|^2 \, , \tag{2.27}$$

Algorithm 1: Metropolis within Gibbs

```
Initialize and suppose two dimensional vector $\boldsymbol{\theta}^{(0)} = (\theta_1^{(0)}, \theta_2^{(0)})$
for $k = 1, \dots, N$ do
    Propose $\theta_1 \sim q(\cdot|\theta_1^{(t-1)}) = q(\theta_1^{(t-1)}|\cdot)$
    Compute
        $$\alpha(\theta_1|\theta_1^{(t-1)}) = \min\left\{1, \frac{\pi(\theta_1|\theta_2^{(t-1)}, \boldsymbol{y})\cancel{q(\theta_1^{(t-1)}|\theta_1)}}{\pi(\theta_1^{(t-1)}|\theta_2^{(t-1)}, \boldsymbol{y})\cancel{q(\theta_1|\theta_1^{(t-1)})}}\right\}$$
    Draw $u \sim \mathcal{U}(0, 1)$
    if $\alpha \geq u$ then
        Accept and set $\theta_1^{(t)} = \theta_1$
    else
        Reject and keep $\theta_1^{(t)} = \theta_1^{(t-1)}$
    end if
    Draw $\theta_2^{(t)} \sim \pi(\cdot|\theta_1^{(t)}, \boldsymbol{y})$
end for
Output: $\boldsymbol{\theta}^{(0)}, \dots, \boldsymbol{\theta}^{(k)}, \dots, \boldsymbol{\theta}^{(N)} \sim \pi(\boldsymbol{\theta}|\boldsymbol{y})$
```

where

$$\hat{\boldsymbol{A}} = \begin{bmatrix} \boldsymbol{\Sigma}^{-1/2}(\boldsymbol{\theta})\boldsymbol{A} \\ \boldsymbol{Q}^{1/2}(\boldsymbol{\theta}) \end{bmatrix}, \quad \hat{\boldsymbol{y}} = \begin{bmatrix} \boldsymbol{\Sigma}^{-1/2}(\boldsymbol{\theta})\boldsymbol{y} \\ \boldsymbol{Q}^{1/2}(\boldsymbol{\theta})\boldsymbol{\mu} \end{bmatrix} \text{[14].} \tag{2.28}$$

Then one sample can be computed by minimising the following equation with respect to $\hat{\boldsymbol{x}}$:

$$\boldsymbol{x}_i = \arg\min_{\hat{\boldsymbol{x}}} \|\hat{\boldsymbol{A}}\hat{\boldsymbol{x}} - (\hat{\boldsymbol{y}} + \boldsymbol{b})\|^2, \quad \boldsymbol{b} \sim \mathcal{N}(\boldsymbol{0}, \mathbf{I}), \tag{2.29}$$

where we add a randomised perturbation $\boldsymbol{b}$. Similarly as in section **??** we can rewrite the argument of Eq. 2.28 to

$$(\boldsymbol{A}^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{\theta})\boldsymbol{A} + \boldsymbol{Q}(\boldsymbol{\theta}))\boldsymbol{x}_i = \boldsymbol{A}^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{\theta})\boldsymbol{y} + \boldsymbol{Q}(\boldsymbol{\theta})\boldsymbol{\mu} + \boldsymbol{v}_1 + \boldsymbol{v}_2, \tag{2.30}$$

where we substitute $-\hat{\boldsymbol{A}}^T\boldsymbol{b} = \boldsymbol{v}_1 + \boldsymbol{v}_2$ so that $\boldsymbol{v}_1 \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{A}^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{\theta})\boldsymbol{A})$ and $\boldsymbol{v}_2 \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{Q}(\boldsymbol{\theta}))$ are independent random variables [6, 13].

If within the MTC scheme the Markov chain from the marginal posterior is ergodic then with independent samples $\boldsymbol{x}^{(k)}$ from the full conditional $\pi(\boldsymbol{x}|\boldsymbol{\theta}, \boldsymbol{y})$ the combined chain $\{(\boldsymbol{x}, \boldsymbol{\theta})^{(1)}, \dots, (\boldsymbol{x}, \boldsymbol{\theta})^{(k)}, \dots, (\boldsymbol{x}, \boldsymbol{\theta})^{(N)}\} \sim \pi(\boldsymbol{x}, \boldsymbol{\theta}|\boldsymbol{y})$ is ergodic as well [15].

2.5.3 t-walk

If there is a functional dependency of the parameters $\boldsymbol{x}$ and the hyper-parameters $\boldsymbol{\theta}$ so that $\boldsymbol{x}(\boldsymbol{\theta})$ we can use the t-walk algorithm by Christens and Fox on $\pi(\boldsymbol{\theta}|\boldsymbol{y})$. We use the t-walk as a black box sampler, where convergence is guaranteed by construction [16].

2.6 Numerical Approxiamtion Methods - Tensor Train

Using the tensor train format to approximate a d-dimensional function $\pi(x)$ enables us to compute marginal posterior probability distribution cheaply. As the name suggest the tensor train format is a train of tensors, more specifically two and three dimensional tensors which we call cores $\pi_k \in \mathbb{R}^{r_{k-1} \times n \times r_k}$ and which are connected through a ranks r_k and r_{k-1} for the kth dimension and defined by the number of gridpoints n, as in Figure **??** displayed. For the first and last dimensional core the outer ranks are $r_0 = r_d = 1$.

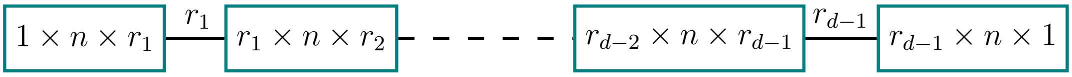
$1 \times n \times r_1$
r_1
$r_1 \times n \times r_2$
$r_{d-2} \times n \times r_{d-1}$
r_{d-1}
$r_{d-1} \times n \times 1$

Figure 2.4: text

Figure 2.5: nice matries picture

The approximated marginal target function

$$
\begin{aligned}
f_{X_k}(x_k) = \frac{1}{z} \Bigg| & \left(\int_{\mathbb{R}} \pi_1(x_1) \boldsymbol{\lambda}_1(x_1) \mathrm{d}x_1 \right) \cdots \left(\int_{\mathbb{R}} \pi_{k-1}(x_{k-1}) \boldsymbol{\lambda}_{k-1}(x_{k-1}) \mathrm{d}x_{k-1} \right) \\
& \quad \pi_k(x_k) \boldsymbol{\lambda}_k(x_k) \\
& \quad \left(\int_{\mathbb{R}} \pi_{k+1}(x_{k+1}) \boldsymbol{\lambda}_{k+1}(x_{k+1}) \mathrm{d}x_{k+1} \right) \cdots \left(\int_{\mathbb{R}} \pi_d(x_d) \boldsymbol{\lambda}_d(x_d) \mathrm{d}x_d \right) \Bigg| ,
\end{aligned}
\tag{2.31}
$$

is given by integration over each core, where kth is in $\mathbb{R}^{r_{k-1} \times r_k}$ and z is some normalising constant [17]. Here we introduce some Lebesgue measurable weight function $\boldsymbol{\lambda}(x) = \prod_{i=1}^{d} \boldsymbol{\lambda}_i(x_i)$. Why? [].

From here the notation and procedure is taken mostly from [18]. For numerical stability we can approximate the sqaure root of

$$\sqrt{\pi(x)} \approx \tilde{g}(x) = \boldsymbol{G}_1(x_1), \dots, \boldsymbol{G}_k(x_k), \dots, \boldsymbol{G}_d(x_d) \tag{2.32}$$

where the TT-core

$$G_k^{(\alpha_{k-1},\alpha_k)}(x_k) = \sum_{i=1}^{n_k} \phi_k^{(i)}(x_k)\boldsymbol{A}_k[\alpha_{k-1}, i, \alpha_k], \quad k = 1, ..., d, \tag{2.33}$$

with the associated kth coefficient tensor $\boldsymbol{A}_k \in \mathbb{R}^{r_{k-1}\times n_k \times r_k}$ and the k-th basis functions $\phi_k^{(i)}(x_k)$.

We assume the function

$$\pi(x) \approx \gamma' + g^2(x)\,, \tag{2.34}$$

where $g(x)$ is defined through the tensor train decomposition plus an error γ' according to the l2 norm. Then the normalised target function is

$$f_X(x) = \frac{1}{z}\pi(x)\boldsymbol{\lambda}(x) = \frac{1}{z}(\gamma'\boldsymbol{\lambda}(x) + g^2(x)\boldsymbol{\lambda}(x)) \tag{2.35}$$

with a normalisation constant z. Consquently the approximated marginal functions can be expressed as

$$\begin{aligned} f_{X_k}(x_k) = \frac{1}{z}\Bigg(&\gamma' \prod_{i=1}^{k-1} \boldsymbol{\lambda}_i(\mathcal{X}_i) \prod_{i=k+1}^{d} \boldsymbol{\lambda}_i(\mathcal{X}_i) \\ &+ \left(\int_{\mathbb{R}} \boldsymbol{G}_1^2(x_1)\boldsymbol{\lambda}_1(x_1)\mathrm{d}x_1\right) \cdots \left(\int_{\mathbb{R}} \boldsymbol{G}_{k-1}^2(x_{k-1})\boldsymbol{\lambda}_{k-1}(x_{k-1})\mathrm{d}x_{k-1}\right) \\ &\boldsymbol{G}_k^2(x_k)\boldsymbol{\lambda}_k(x_k) \\ &\left(\int_{\mathbb{R}} \boldsymbol{G}_{k+1}^2(x_{k+1})\boldsymbol{\lambda}_{k+1}(x_{k+1})\mathrm{d}x_{k+1}\right) \cdots \left(\int_{\mathbb{R}} \boldsymbol{G}_d^2(x_d)\boldsymbol{\lambda}_d(x_d)\mathrm{d}x_d\right)\Bigg)\,, \end{aligned} \tag{2.36}$$

where $\boldsymbol{\lambda}_k(\mathcal{X}_k) = \int_{\mathcal{X}_k} \boldsymbol{\lambda}_k(x_k)\mathrm{d}x_k$.

To effeciently calculate these marginals on can use a procedure similiar to something that is called left and right orthogonalization of cores [19]. To do so we define the mass matrix $\boldsymbol{M}_k \in \mathbb{R}^{n_k \times n_k}$ by

$$\boldsymbol{M}_k[i,j] = \int_{X_k} \phi_k^{(i)}(x_k)\phi_k^{(j)}(x_k)\boldsymbol{\lambda}(x_k)\,dx_k, \quad i = 1, ..., n_k, \quad j = 1, ..., n_k, \tag{2.37}$$

where $\{\phi_k^{(i)}(x_k)\}_{i=1}^{n_k}$ is the set of basis functions for the k-th coordinate.

2.6.1 Marginal Functions

We calculate the marginal functions through procedures, which we call backward marginalisation [] and forward marginalisation. We gain the coefficient matrices $\boldsymbol{B}_k$ through backward marginalisation and the coefficient matrices $\boldsymbol{B}_{pre,n}$ through forward marginalisation, which enables us to calculate marginal fuunction similar to [].

The proposition 1 to caculte $\boldsymbol{B}_k$ is taken from [].

Proposition 1 (Backward Marginalisation)**:** Starting with the last coordinate $k = d$, we set $\boldsymbol{B}_d = \boldsymbol{A}_d$. The following procedure can be used to obtain the coefficient tensor $\boldsymbol{B}_{k-1} \in \mathbb{R}^{r_{k-2} \times n_{k-1} \times r_{k-1}}$, which we need for defining the marginal function $f_{X_k}(x_k)$:

1. Use the Cholesky decomposition of the mass matrix, $\boldsymbol{L}_k \boldsymbol{L}_k^\top = \boldsymbol{M}_k \in \mathbb{R}^{n_k \times n_k}$, to construct a tensor $\boldsymbol{C}_k \in \mathbb{R}^{r_{k-1} \times n_k \times r_k}$:

$$\boldsymbol{C}_k[\alpha_{k-1}, \tau, l_k] = \sum_{i=1}^{n_k} \boldsymbol{B}_k[\alpha_{k-1}, i, l_k] \boldsymbol{L}_k[i, \tau]. \tag{2.38}$$

2. Unfold $\boldsymbol{C}_k$ along the first coordinate and compute the thin QR decomposition, so that $\boldsymbol{C}_k^{(R)} \in \mathbb{R}^{r_{k-1} \times (n_k r_k)}$:

$$\boldsymbol{Q}_k \boldsymbol{R}_k = \left(\boldsymbol{C}_k^{(R)}\right)^\top. \tag{2.39}$$

3. Compute the new coefficient tensor:

$$\boldsymbol{B}_{k-1}[\alpha_{k-2}, i, l_{k-1}] = \sum_{\alpha_{k-1}=1}^{r_{k-1}} \boldsymbol{A}_{k-1}[\alpha_{k-2}, i, \alpha_{k-1}] \boldsymbol{R}_k[l_{k-1}, \alpha_{k-1}]. \tag{2.40}$$

Then we need to do this the other way as well.

In addiotn we also have to do forward marginalistion starig with the first dimension

Proposition 2 (Forward Marginalistaion)**:** Starting with the first coordinate $k = 1$, we set $\boldsymbol{B}_{pre,1} = \boldsymbol{A}_1$. The following procedure can be used to obtain the coefficient tensor $\boldsymbol{B}_{pre,k+1} \in \mathbb{R}^{r_k \times n_{k+1} \times r_{k+1}}$ for defining the marginal function $f_{X_k}(x_k)$:

1. Use the Cholesky decomposition of the mass matrix, $\boldsymbol{L}_k \boldsymbol{L}_k^\top = \boldsymbol{M}_k \in \mathbb{R}^{n_k \times n_k}$, to construct a tensor $\boldsymbol{C}_k \in \mathbb{R}^{r_{k-1} \times n_k \times r_k}$:

$$\boldsymbol{C}_{pre,k}[\alpha_{k-1}, \tau, l_k] = \sum_{i=1}^{n_k} \boldsymbol{L}_k[i, \tau] \boldsymbol{B}_{pre,k}[\alpha_{k-1}, i, l_k]. \tag{2.41}$$

2. Unfold $\boldsymbol{C}_{pre,k}$ along the first coordinate and compute the thin QR decomposition, so that $\boldsymbol{C}_{pre,k}^{(R)} \in \mathbb{R}^{(r_{k-1}n_k) \times r_k}$:

$$\boldsymbol{Q}_{pre,k} \boldsymbol{R}_{pre,k} = (\boldsymbol{C}_{pre,k}^{(R)}). \tag{2.42}$$

3. Compute the new coefficient tensor $\boldsymbol{B}_{pre,k+1} \in \mathbb{R}^{r_{k-1} \times n_k \times r_k}$:

$$\boldsymbol{B}_{pre,k+1}[l_{k+1}, i, \alpha_{k+1}] = \sum_{\alpha_k=1}^{r_k} \boldsymbol{R}_{pre,k}[l_{k+1}, \alpha_k] \boldsymbol{A}_{k+1}[\alpha_k, i, \alpha_{k+1}]. \tag{2.43}$$

The marginal PDF of X_k can be expressed as

$$f_{X_k}(x_k) = \frac{1}{z}\left(\gamma' \prod_{i=1}^{k-1} \lambda_i(X_i) \prod_{i=k+1}^{d} \lambda_i(X_i) + \sum_{l_{k-1}=1}^{r_{k-1}} \sum_{l_k=1}^{r_k} \left(\sum_{i=1}^{n_k} \phi_k^{(i)}(x_k) \boldsymbol{D}_k[l_{k-1}, i, l_k]\right)^2\right) \lambda_k(x_k), \tag{2.44}$$

where $\boldsymbol{D}_k \in \mathbb{R}^{r_{k-1} \times n_k \times r_k}$ and $\boldsymbol{R}_{pre,k-1} \in \mathbb{R}^{r_{k-1} \times r_{k-1}}$ and $\boldsymbol{B}_k \in \mathbb{R}^{r_{k-1} \times n_k \times r_k}$

$$\boldsymbol{D}_k[l_{k-1}, i, l_k] = \sum_{\alpha_{k-1}=1}^{r_{k-1}} \boldsymbol{R}_{pre,k-1}[l_{k-1}, \alpha_{k-1}] \boldsymbol{B}_k[\alpha_{k-1}, i, l_k]. \tag{2.45}$$

Special Cases The marginal PDF of X_1 can be expressed as

$$f_{X_1}(x_1) = \frac{1}{z}\left(\gamma' \prod_{i=2}^{d} \boldsymbol{\lambda}_i(\boldsymbol{\mathcal{X}}_i) + \sum_{l_1=1}^{r_1} \left(\sum_{i=1}^{n_1} \phi_1^{(i)}(x_1) \boldsymbol{D}_1[i, l_1]\right)^2\right) \boldsymbol{\lambda}_1(x_1), \tag{2.46}$$

where $\boldsymbol{D}_1[i, l_1] = \boldsymbol{B}_1[\alpha_0, i, l_1]$ and $\alpha_0 = 1$.

The marginal PDF of X_n can be expressed as

$$f_{X_n}(x_n) = \frac{1}{z}\left(\gamma' \prod_{i=1}^{d-1} \boldsymbol{\lambda}_i(\boldsymbol{\mathcal{X}}_i) + \sum_{l_{n-1}=1}^{r_{n-1}} \left(\sum_{i=1}^{n_1} \phi_1^{(i)}(x_1) \boldsymbol{D}_n[l_{n-1}, i]\right)^2\right) \boldsymbol{\lambda}_n(x_n), \tag{2.47}$$

where $\boldsymbol{D}_n[l_{n-1}, i] = \boldsymbol{B}_{pre,n}[l_{n-1}, i, \alpha_n]$ and $\alpha_n = 1$.

Appendices

References

[1] *Envisat MIPAS An Instrument for Atmospheric Chemistry and Climate Research.* Noordwijk: ESA Publications Division, 2000.

[2] Iouli E Gordon et al. “The HITRAN2020 molecular spectroscopic database”. In: *Journal of Quantitative Spectroscopy and Radiative Transfer* 277 (2022), p. 107949.

[3] George B. Rybicki and Alan P. Lightman. *Radiative processes in Astrophysics.* Weinheim: Wiley-VCH, 2004.

[4] Marcel Berger. *Geometry I. 4th Edition.* Berlin Heidelberg: Springer-Verlag, 2009.

[5] Katsumi Nomizu and Takeshi Sasaki. *Affine differential geometry.* Cambridge: Cambridge University Press, 1994.

[6] Colin Fox and Richard A Norton. “Fast sampling in a linear-Gaussian inverse problem”. In: *SIAM/ASA Journal on Uncertainty Quantification* 4.1 (2016), pp. 1191–1218.

[7] Jari P. Kaipio and Erkki Somersalo. *Statistical and Computational Inverse Problems.* New York: Springer-Verlag New York, 2005.

[8] Sze M Tan, Colin Fox, and Geoff K. Nicholls. *Course notes for ELEC 445 – Inverse Problems and Imaging.* `https://coursesupport.physics.otago.ac.nz/wiki/pmwiki.php/ELEC445/HomePage`. [Online; accessed 10/12/23]. 2016.

[9] Per Christian Hansen and Dianne Prost O’Leary. “The use of the L-curve in the regularization of discrete ill-posed problems”. In: *SIAM Journal on Scientific Computing* 14.6 (1993), pp. 1487–1503.

[10] Gareth O. Roberts and Jeffrey S Rosenthal. “General state space Markov chains and MCMC algorithms”. In: *Probability Surveys* 1 (2004), pp. 20–71.

[11] Charles J Geyer. “Practical markov chain monte carlo”. In: *Statistical science* (1992), pp. 473–483.

[12] Gareth O. Roberts and Jeffrey S Rosenthal. “Harris recurrence of Metropolis-within-Gibbs and trans-dimensional Markov chains”. In: *The Annals of Applied Probability* 16.4 (2006), 2123–2139.

[13] Johnathan M Bardsley. “MCMC-based image reconstruction with uncertainty quantification”. In: *SIAM Journal on Scientific Computing* 34.3 (2012), A1316–A1332.

[14] Johnathan M Bardsley et al. “Randomize-then-optimize: A method for sampling from posterior distributions in nonlinear inverse problems”. In: *SIAM Journal on Scientific Computing* 36.4 (2014), A1895–A1910.

[15] Felipe Acosta, Mark L Huber, and Galin L Jones. “Markov chain Monte Carlo with linchpin variables”. In: *preprint* (2014).

[16] J. Andrés Christen and Colin Fox. “A general purpose sampling algorithm for continuous distributions (the t-walk)”. In: *Bayesian Analysis* 5.2 (2010), pp. 263 –281. URL: `https://doi.org/10.1214/10-BA603`.

[17] Sergey Dolgov et al. “Approximation and sampling of multivariate probability distributions in the tensor train decomposition”. In: *Statistics and Computing* 30 (2020), pp. 603–625.

[18] Tiangang Cui and Sergey Dolgov. “Deep composition of tensor-trains using squared inverse rosenblatt transports”. In: *Foundations of Computational Mathematics* 22.6 (2022), pp. 1863–1922.

[19] Ivan V Oseledets. “Tensor-train decomposition”. In: *SIAM Journal on Scientific Computing* 33.5 (2011), pp. 2295–2317.