

# Contents

<b>List of Figures</b>	<b>iii</b>
<b>1 Introduction</b>	<b>3</b>
1.1 What is going on?, 3 facts, What is new in this thesis? . . . . .	3
1.2 Thesis Outline . . . . .	3
<b>2 Theoretical and Technical Background</b>	<b>5</b>
2.1 Forward Model . . . . .	5
2.2 Affine Map . . . . .	7
2.3 Bayesian Inference . . . . .	8
2.3.1 Marginal and then Conditional . . . . .	10
2.4 Regularisation . . . . .	11
2.5 Sampling Methods . . . . .	12
2.5.1 Metropolis- within Gibbs sampling . . . . .	13
2.5.2 Draw a sample from a multivariate normal distribution . . . . .	14
2.5.3 t-walk . . . . .	15
2.6 Numerical Approxiamtion Methods - Tensor Train . . . . .	16
2.6.1 Marginal Functions . . . . .	18
<b>Appendices</b>	
<b>References</b>	<b>23</b>



# List of Figures

2.1	text . . . . .	5
2.2	Schematics of Affine Map . . . . .	8
2.3	Bayesian Inference DAG . . . . .	9
2.4	text . . . . .	16
2.5	nice matrices picture [18] . . . . .	16



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**

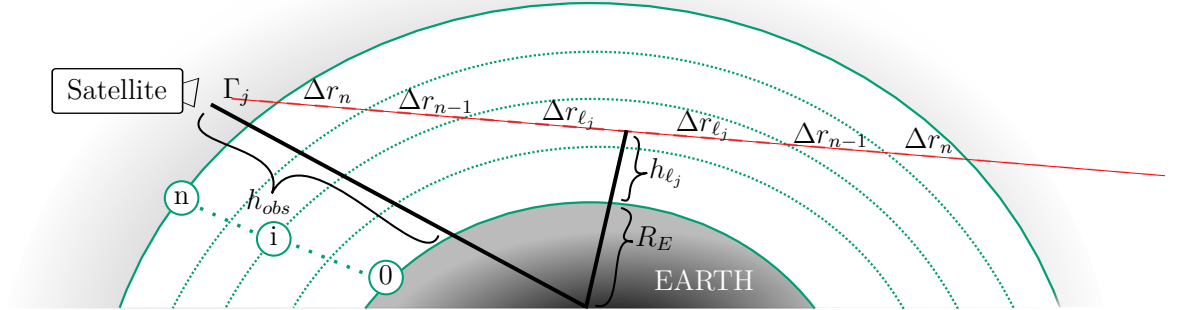




# 2

## 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  $\Gamma_j$  for  $j = 1, 2, \dots, m$ . For each measurement we can define a tangent height  $h_{\ell_j}$  as the shortest distance along the line of sight to the earth.

The  $j^{\text{th}}$  measurement, taken on line of sight  $\Gamma_j$  is modelled by the radiative

transfer equation (RTE) [1]

$$y_j = \int_{\Gamma_j} B(\nu, T) k(\nu, T) \frac{p(T)}{k_B T(r)} x(r) \tau(r) dr + \eta_j \quad (2.1)$$

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

where the path from the satellite along the line-of-sight of the  $j^{\text{th}}$  pointing direction is  $\Gamma_j$  and the ozone concentration at distance  $r$  from the radiometer is  $x(r)$  plus some noise  $\eta_j$ . Within the stratosphere the number density  $p(T)/(k_B T(r))$  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  $\nu$  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  $i^{\text{th}}$  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  $\mathbf{x} = \{x_1, x_2, \dots, x_n\} \in \mathbb{R}^n$  or the fraction of pressure and temperature  $\mathbf{p}/\mathbf{T} = \{p_1/T_1, p_2/T_2, \dots, p_n/T_n\} \in \mathbb{R}^n$ , we can rewrite the integral in Eq. (2.1) as e.g. as a vector multiplication  $\mathbf{A}_j(\mathbf{x}, \mathbf{p}, \mathbf{T}) \mathbf{x}$ , where the non-linear absorption  $\tau(r)$  is included in  $\mathbf{A}_j(\mathbf{x}, \mathbf{p}, \mathbf{T})$ . Here, the row vector  $\mathbf{A}_j(\mathbf{x}, \mathbf{p}, \mathbf{T}) \in \mathbb{R}^n$  defines a Kernel for each measurement so that the data vector

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

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

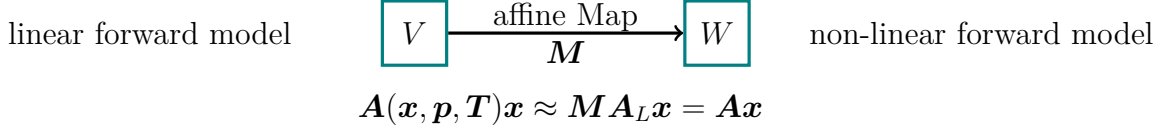
Since the measurement process includes absorption  $\tau(r)$  reducing measurements slightly and making the inverse problem only weakly non-linear. We use that to approximate the non-linear forward model  $\mathbf{A}(\mathbf{x}, \mathbf{p}, \mathbf{T})$  with a map  $\mathbf{M}$  so that  $\mathbf{A}(\mathbf{x},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

## 2.2 Affine Map

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

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



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

$$\mathbf{y} = \mathbf{A}\mathbf{x} + \boldsymbol{\eta} \quad (2.4)$$

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

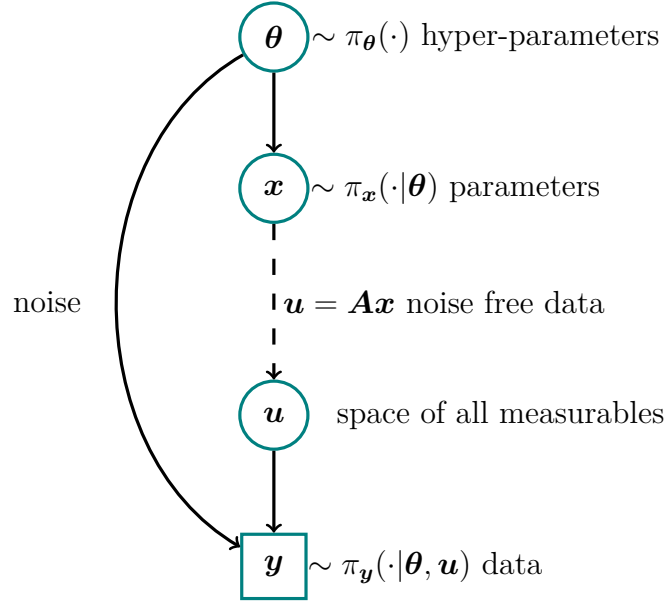
We can visualise the correlation structure of a measurement process through a hierarchially ordered directed acyclic graph (DAG), see Figure 2.3. As an observatory process naturally includes some random noise we include that in our DAG and classify the noise as a hyper-parameter in  $\boldsymbol{\theta}$  [6]. Other hyper-parameters influence the parameters  $\mathbf{x}$  deterministically, which are then mapped through the forward model onto the space of all measurables  $\mathbf{u}$ , from which we observe some data  $\mathbf{y}$  including noise as previously mentioned. Drawing a DAG can help us to dependences within the measurement and modelling process. Given some data we infer the distribution of the underlying 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 hierarchical model we need to define a likelihood function as well as a distribution over the unknown parameters  $\mathbf{x}$  and hyper-parameters  $\boldsymbol{\theta}$ .

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

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

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



**Figure 2.3:** The directed acyclic graph (DAG) for a linear inverse problem visualises statistical dependencies as solid line arrows and deterministic dependencies as dotted arrows. The parameters  $\mathbf{x}$  have some statistical dependency of those hyper-parameters  $\boldsymbol{\theta}$ , which are distributed as  $\pi(\boldsymbol{\theta})$ . Then a parameter  $\mathbf{x} \sim \pi_{\mathbf{x}}(\cdot|\boldsymbol{\theta})$  is mapped onto the space of all measurables  $\mathbf{u} = \mathbf{A}\mathbf{x}$  deterministically through the linear forward model  $\mathbf{A}$ . From the space of all measurables we observe some data  $\mathbf{y} = \mathbf{A}\mathbf{x} + \boldsymbol{\eta}$ , statistically, so that  $\mathbf{y} \sim \pi_{\mathbf{y}}(\cdot|\boldsymbol{\theta}, \mathbf{u})$ , with naturally some random noise  $\boldsymbol{\eta} \sim \pi_{\boldsymbol{\eta}}(\cdot|\boldsymbol{\theta})$ .

Due to the assumption of Gaussian noise  $\boldsymbol{\eta} \sim \mathcal{N}(0, \boldsymbol{\Sigma}(\boldsymbol{\theta}))$  the likelihood function  $\pi(\mathbf{y}|\mathbf{x}, \boldsymbol{\theta})$ , which includes information about the measurement process and gives a measure of how well parameters and hyperparameters 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  $\pi(\mathbf{x}|\boldsymbol{\theta})$  with the prior precision matrix  $\mathbf{Q}(\boldsymbol{\theta})$  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  $\mathbf{x}$ .

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

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

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

the posterior distribution

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

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

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

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

One way to work around the high dimensionality is to parameterise  $\mathbf{x}$  using hyper-parameters  $\boldsymbol{\theta}$  so that  $\mathbf{x}(\boldsymbol{\theta})$ . Another way is to separate the posterior distribution over latent field  $\mathbf{x}$  and the hyper-parameters  $\boldsymbol{\theta}$ . This is particular beneficial, when  $\mathbf{x}$  is high dimensional, e.g.  $\mathbf{x} \in \mathbb{R}^n$  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 [7] eq. 2.8 becomes

$$\mathbb{E}_{\mathbf{x} | \mathbf{y}}[h(\mathbf{x})] = \mathbb{E}_{\boldsymbol{\theta} | \mathbf{y}}[\mathbb{E}_{\mathbf{x} | \boldsymbol{\theta}, \mathbf{y}}[h(\mathbf{x}(\boldsymbol{\theta}))]] = \int \mathbb{E}_{\mathbf{x} | \boldsymbol{\theta}, \mathbf{y}}[h(\mathbf{x}(\boldsymbol{\theta}))] \pi(\boldsymbol{\theta} | \mathbf{y}) d\boldsymbol{\theta}, \quad (2.9)$$

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

### 2.3.1 Marginal and then Conditional

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

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

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

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

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

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

with

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

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

$$\mathbf{x}|\theta, \mathbf{y} \sim \mathcal{N}\left(\underbrace{\mu + (\mathbf{A}^T\Sigma^{-1}\mathbf{A} + \mathbf{Q})^{-1}\mathbf{A}^T\Sigma^{-1}(\mathbf{y} - \mathbf{A}\mu)}_{\mu_{\mathbf{x}|\theta, \mathbf{y}}}, \underbrace{(\mathbf{A}^T\Sigma^{-1}\mathbf{A} + \mathbf{Q})^{-1}}_{\Sigma_{\mathbf{x}|\mathbf{y}, \theta}}\right) \quad (2.14)$$

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

## 2.4 Regularisation

Another method to find a solution to a linear inverse problem as in Eq. 2.4 is to find a solution  $\mathbf{x}_\lambda$  according to a data misfit norm and a regularisation semi-norm as in [6]. We will discuss the case of Tikhonov regularisation [8, 9] as this is the most similar to a linear-Gaussian hierarchical Bayesian model.

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

$$\|\mathbf{y} - \mathbf{Ax}\| \quad (2.15)$$

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

$$\lambda \|\mathbf{T}\mathbf{x}\| \quad (2.16)$$

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

$$\mathbf{x}_\lambda = \arg \min_{\mathbf{x}} \|\mathbf{y} - \mathbf{Ax}\|^2 + \lambda \|\mathbf{T}\mathbf{x}\|^2 \quad (2.17)$$

can be found by the derivativ

$$\nabla_{\mathbf{x}} \left\{ (\mathbf{y} - \mathbf{A}\mathbf{x}_\lambda)^T (\mathbf{y} - \mathbf{A}\mathbf{x}_\lambda) + \lambda \mathbf{x}_\lambda^T \mathbf{T}^T \mathbf{T} \mathbf{x}_\lambda \right\} = 0 \quad (2.18)$$

$$\iff \nabla_{\mathbf{x}} \left\{ \mathbf{y}^T \mathbf{y} + \mathbf{x}_\lambda^T \mathbf{A}^T \mathbf{A} \mathbf{x}_\lambda - \mathbf{y}^T \mathbf{A} \mathbf{x}_\lambda - \mathbf{x}_\lambda^T \mathbf{A}^T \mathbf{y} + \lambda \mathbf{x}_\lambda^T \mathbf{T}^T \mathbf{T} \mathbf{x}_\lambda \right\} = 0 \quad (2.19)$$

$$\iff 2\mathbf{A}^T \mathbf{A} \mathbf{x}_\lambda - 2\mathbf{A}^T \mathbf{y} + 2\lambda \mathbf{T}^T \mathbf{T} \mathbf{x}_\lambda = 0. \quad (2.20)$$

Then a regularised solution is given as:

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

where we can set  $\mathbf{T}^T \mathbf{T} = \mathbf{L}$ , which is typically matrix approximation of the nth derivative [9]. In practise  $\mathbf{x}_\lambda$  is calculated for a range of  $\lambda$ , and is evaluated by the data-misfit norm with respect to the regularised semi-norm so that the best  $\mathbf{x}_\lambda$  lays on the point of maximum curvature of a so-called L-Curve [10], 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

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

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

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

for large enough samples size  $N$  of a sample set  $\mathcal{M} = \{(\mathbf{x}, \boldsymbol{\theta})^{(1)}, \dots, (\mathbf{x}, \boldsymbol{\theta})^{(k)}, \dots, (\mathbf{x}, \boldsymbol{\theta})^{(N)}\} \sim \pi(\mathbf{x}, \boldsymbol{\theta} | \mathbf{y})$ . The variance of this unbiased estimator is  $\mathcal{O}(1/N)$  so large samples sizes can reduce the uncertainty of  $\boldsymbol{\mu}_{\text{samp}}$  [11]. We can do this as the central limit



theorem states that the samples means  $\boldsymbol{\mu}_{\text{samp}}^{(i)}$ , of samples sets  $\mathcal{M}_i$  for  $i = 1, \dots, n$  of any distribution, converge in distribution to a normal distribution so that

$$\sqrt{n}(\boldsymbol{\mu}_{\text{samp}}^{(i)} - \boldsymbol{\mu}_{\text{int}}) \xrightarrow{\mathcal{D}} \mathcal{N}(0, \sigma^2) [12], \quad (2.24)$$

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

Now, we have to show that the methods we use actually draw samples  $\mathcal{M} = \{(\mathbf{x}, \boldsymbol{\theta})^{(1)}, \dots, (\mathbf{x}, \boldsymbol{\theta})^{(k)}, \dots, (\mathbf{x}, \boldsymbol{\theta})^{(N)}\} \sim \pi(\mathbf{x}, \boldsymbol{\theta} | \mathbf{y})$  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 [9, 11]. The ergodicity theorem in [9] states that, if an aperiodic and irreducible Markov chain  $\mathcal{M}$  is reversible then it converges towards a stationary unique equilibrium distribution  $\pi(\mathbf{x}, \boldsymbol{\theta} | \mathbf{y})$ . 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  $\pi(\mathbf{x}^{(k)}, \boldsymbol{\theta}^{(k)} | \mathbf{y})$  for  $k = 1, \dots, N$  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 separately from  $\pi(\boldsymbol{\theta} | \mathbf{y})$  and  $\pi(\mathbf{x} | \boldsymbol{\theta}, \mathbf{y})$ . To sample from  $\pi(\boldsymbol{\theta} | \mathbf{y})$  we use a Metropolis-within-Gibbs sampler as in [6] and discuss the 2 dimensional case, as used in this thesis, with  $\boldsymbol{\theta} = (\theta_1, \theta_2)$ , where we do a metropolis step in  $\theta_1$  direction and a gibbs step in  $\theta_2$  direction. Ergodicity is proven here [13].

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 [13] and cancels when computing

the acceptance probability  $\alpha$ . We accept and set  $\theta_1^{(t)} = \theta_1$  with

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

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

**Algorithm 1:** Metropolis within Gibbs

- 1: Initialize and suppose two dimensional vector  $\boldsymbol{\theta}^{(0)} = (\theta_1^{(0)}, \theta_2^{(0)})$
- 2: **for**  $k = 1, \dots, N$  **do**
- 3:   Propose  $\theta_1 \sim q(\cdot|\theta_1^{(t-1)}) = q(\theta_1^{(t-1)}|\cdot)$
- 4:   Compute

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

- 5:   Draw  $u \sim \mathcal{U}(0, 1)$
- 6:   **if**  $\alpha \geq u$  **then**
- 7:     Accept and set  $\theta_1^{(t)} = \theta_1$
- 8:   **else**
- 9:     Reject and keep  $\theta_1^{(t)} = \theta_1^{(t-1)}$
- 10:   **end if**
- 11:   Draw  $\theta_2^{(t)} \sim \pi(\cdot|\theta_1^{(t)}, \mathbf{y})$
- 12: **end for**
- 13: Output:  $\boldsymbol{\theta}^{(0)}, \dots, \boldsymbol{\theta}^{(k)}, \dots, \boldsymbol{\theta}^{(N)} \sim \pi(\boldsymbol{\theta}|\mathbf{y})$

### 2.5.2 Draw a sample from a multivariate normal distribution

after sampling from  $\pi(\boldsymbol{\theta}|\mathbf{y})$  we draw samples from  $\pi(\mathbf{x}|\boldsymbol{\theta}, \mathbf{y})$  within the MTC scheme. For Linear Gaussian Bayesian hierarchical model we can draw a sample

$\mathbf{x}$  from the multivariate normal distribution  $\pi(\mathbf{x}|\boldsymbol{\theta}, \mathbf{y})$  using the randomize then optimise (RTO) method [14].

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

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

$$= \exp\|\hat{\mathbf{A}}\mathbf{x} - \hat{\mathbf{y}}\|^2, \quad (2.27)$$

where

$$\hat{\mathbf{A}} = \begin{bmatrix} \boldsymbol{\Sigma}^{-1/2}(\boldsymbol{\theta})\mathbf{A} \\ \mathbf{Q}^{1/2}(\boldsymbol{\theta}) \end{bmatrix}, \quad \hat{\mathbf{y}} = \begin{bmatrix} \boldsymbol{\Sigma}^{-1/2}(\boldsymbol{\theta})\mathbf{y} \\ \mathbf{Q}^{1/2}(\boldsymbol{\theta})\boldsymbol{\mu} \end{bmatrix} \quad [15]. \quad (2.28)$$

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

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

where we add a randomised perturbation  $\mathbf{b}$ . Similarly as in section 2.4 we can rewrite the argument of Eq. 2.28 to

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

where we substitute  $-\hat{\mathbf{A}}^T \mathbf{b} = \mathbf{v}_1 + \mathbf{v}_2$  so that  $\mathbf{v}_1 \sim \mathcal{N}(\mathbf{0}, \mathbf{A}^T \boldsymbol{\Sigma}^{-1}(\boldsymbol{\theta})\mathbf{A})$  and  $\mathbf{v}_2 \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}(\boldsymbol{\theta}))$  are independent random variables [6, 14].

If within the MTC scheme the Markov chain from the marginal posterior is ergodic then with independent samples  $\mathbf{x}^{(k)}$  from the full conditional  $\pi(\mathbf{x}|\boldsymbol{\theta}, \mathbf{y})$  the combined chain  $\{(\mathbf{x}, \boldsymbol{\theta})^{(1)}, \dots, (\mathbf{x}, \boldsymbol{\theta})^{(k)}, \dots, (\mathbf{x}, \boldsymbol{\theta})^{(N)}\} \sim \pi(\mathbf{x}, \boldsymbol{\theta}|\mathbf{y})$  is ergodic as well [16].

### 2.5.3 t-walk

If there is a functional dependency of the parameters  $\mathbf{x}$  and the hyper-parameters  $\boldsymbol{\theta}$  so that  $\mathbf{x}(\boldsymbol{\theta})$  we can use the t-walk algorithm by Christens and Fox on  $\pi(\boldsymbol{\theta}|\mathbf{y})$ . We use the t-walk as a black box sampler, where convergence is guaranteed by construction [17].

## 2.6 Numerical Approxiamtion Methods - Tensor Train

Using the tensor train (TT) format to approximate a  $d$ -dimensional function  $\pi(\mathbf{x}, \boldsymbol{\theta} | \mathbf{y})$  which we set to  $\pi(x) \equiv \pi(\mathbf{x}, \boldsymbol{\theta} | \mathbf{y})$ , with  $x \in \mathbb{R}^n$  to improve readability, enables us to compute marginal posterior probability distributions 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 are connected through ranks  $r_k$  and  $r_{k-1}$  for each  $k = 1, \dots, d$ , with the number of grid points  $n$ , as in Figures 2.4 and ?? displayed. For the first and last dimensional the core outer ranks are  $r_0 = r_d = 1$ , so that for  $x = x_1, \dots, x_d$  the function value  $\pi(x) = (\pi_1(x_1) \dots \pi_d(x_d)) \in \mathbb{R}$  is a vector-matrix-vector multiplication and each core  $\pi_k(x_k)$  at  $x_k$  has dimensions  $r_{k-1} \times 1 \times r_k = r_{k-1} \times r_k$ .

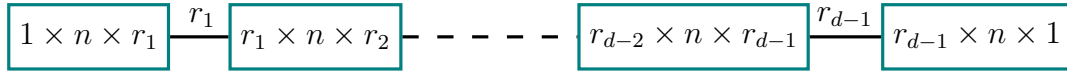


Figure 2.4: text

Figure 2.5: nice matrices picture [18]

Consequently we can approximate the marginal target function

$$f_{X_k}(x_k) = \frac{1}{z} \left| \left( \int_{\mathbb{R}} \pi_1(x_1) \lambda_1(x_1) dx_1 \right) \cdots \left( \int_{\mathbb{R}} \pi_{k-1}(x_{k-1}) \lambda_{k-1}(x_{k-1}) dx_{k-1} \right) \right. \\ \left. \pi_k(x_k) \lambda_k(x_k) \right. \\ \left. \left( \int_{\mathbb{R}} \pi_{k+1}(x_{k+1}) \lambda_{k+1}(x_{k+1}) dx_{k+1} \right) \cdots \left( \int_{\mathbb{R}} \pi_d(x_d) \lambda_d(x_d) dx_d \right) \right|, \quad (2.31)$$

by integration over each core [19] with some normalising constant  $z$  and some Lebesgue measurable weight  $\lambda_i(x_i)$  for  $i = 1, \dots, d$  [20]. Why? [].

From here we follow the notation and procedure mostly from [20] as for numerical stability we approximate the square root

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

where each TT-core

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

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

$$\pi(x) \approx \gamma' + g^2(x), \quad (2.34)$$

is approximated through the TT decomposition  $g(x)$  plus an error  $\gamma'$  to assure positivity. The error  $\gamma'$  is chosen according to the L2 norm

$$\gamma' \leq \frac{1}{\lambda(\mathcal{X})} \|g - \sqrt{\pi}\|_2^2, \quad (2.35)$$

with  $\lambda(\mathcal{X}) = \prod_{i=1}^d \lambda_i(\mathcal{X}_i)$  and  $\lambda_i(\mathcal{X}_i) = \int_{\mathcal{X}_i} \lambda_i(x_i) dx_i$ , and . Then the normalised target function with respect to the weight  $\lambda(x) = \prod_{i=1}^d \lambda_i(x_i)$  is:

$$f_X(x) = \frac{1}{z} \pi(x) \lambda(x) = \frac{1}{z} (\gamma' \lambda(x) + g^2(x) \lambda(x)), \quad (2.36)$$

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

$$\begin{aligned} f_{X_k}(x_k) = \frac{1}{z} & \left( \gamma' \prod_{i=1}^{k-1} \lambda_i(\mathcal{X}_i) \prod_{i=k+1}^d \lambda_i(\mathcal{X}_i) \right. \\ & + \left( \int_{\mathbb{R}} \mathbf{G}_1^2(x_1) \lambda_1(x_1) dx_1 \right) \cdots \left( \int_{\mathbb{R}} \mathbf{G}_{k-1}^2(x_{k-1}) \lambda_{k-1}(x_{k-1}) dx_{k-1} \right) \\ & \mathbf{G}_k^2(x_k) \lambda_k(x_k) \\ & \left. \left( \int_{\mathbb{R}} \mathbf{G}_{k+1}^2(x_{k+1}) \lambda_{k+1}(x_{k+1}) dx_{k+1} \right) \cdots \left( \int_{\mathbb{R}} \mathbf{G}_d^2(x_d) \lambda_d(x_d) dx_d \right) \right). \end{aligned} \quad (2.37)$$

To efficiently calculate these marginals one can use a procedure similar to something that is called left and right orthogonalisation of cores [21]. To do so we define the mass matrix  $\mathbf{M}_k \in \mathbb{R}^{n_k \times n_k}$  by

$$\mathbf{M}_k[i, j] = \int_{X_k} \phi_k^{(i)}(x_k) \phi_k^{(j)}(x_k) \lambda(x_k) dx_k, \quad i = 1, \dots, n_k, \quad j = 1, \dots, n_k, \quad (2.38)$$

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 [20] and forward marginalisation. We gain the coefficient matrices  $\mathbf{B}_k$  through backward marginalisation and the coefficient matrices  $\mathbf{B}_{pre,n}$  through forward marginalisation, which enables us to calculate marginal function similar to [20]. The proposition 1 to calculate  $\mathbf{B}_k$  is taken from [20].

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

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

$$\mathbf{C}_k[\alpha_{k-1}, \tau, l_k] = \sum_{i=1}^{n_k} \mathbf{B}_k[\alpha_{k-1}, i, l_k] \mathbf{L}_k[i, \tau]. \quad (2.39)$$

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

$$\mathbf{Q}_k \mathbf{R}_k = (\mathbf{C}_k^{(R)})^\top. \quad (2.40)$$

3. Compute the new coefficient tensor:

$$\mathbf{B}_{k-1}[\alpha_{k-2}, i, l_{k-1}] = \sum_{\alpha_{k-1}=1}^{r_{k-1}} \mathbf{A}_{k-1}[\alpha_{k-2}, i, \alpha_{k-1}] \mathbf{R}_k[l_{k-1}, \alpha_{k-1}]. \quad (2.41)$$

We start the forward marginalisation with the first dimension as in Proposition 2.

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

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

$$\mathbf{C}_{pre,k}[\alpha_{k-1}, \tau, l_k] = \sum_{i=1}^{n_k} \mathbf{L}_k[i, \tau] \mathbf{B}_{pre,k}[\alpha_{k-1}, i, l_k]. \quad (2.42)$$

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

$$\mathbf{Q}_{pre,k} \mathbf{R}_{pre,k} = (\mathbf{C}_{pre,k}^{(R)}). \quad (2.43)$$

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

$$\mathbf{B}_{pre,k+1}[l_{k+1}, i, \alpha_{k+1}] = \sum_{\alpha_k=1}^{r_k} \mathbf{R}_{pre,k}[l_{k+1}, \alpha_k] \mathbf{A}_{k+1}[\alpha_k, i, \alpha_{k+1}]. \quad (2.44)$$

The marginal PDF of  $X_k$  can be expressed as

$$f_{X_k}(x_k) = \frac{1}{z} \left( \gamma' \prod_{i=1}^{k-1} \lambda_i(X_i) \prod_{i=k+1}^d \lambda_i(X_i) + \sum_{l_{k-1}=1}^{r_{k-1}} \sum_{l_k=1}^{r_k} \left( \sum_{i=1}^{n_k} \phi_k^{(i)}(x_k) \mathbf{D}_k[l_{k-1}, i, l_k] \right)^2 \right) \lambda_k(x_k), \quad (2.45)$$

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

$$\mathbf{D}_k[l_{k-1}, i, l_k] = \sum_{\alpha_{k-1}=1}^{r_{k-1}} \mathbf{R}_{pre,k-1}[l_{k-1}, \alpha_{k-1}] \mathbf{B}_k[\alpha_{k-1}, i, l_k]. \quad (2.46)$$

In the special case for the first dimension,  $f_{X_1}(x_1)$  can be expressed as

$$f_{X_1}(x_1) = \frac{1}{z} \left( \gamma' \prod_{i=2}^d \lambda_i(\mathcal{X}_i) + \sum_{l_1=1}^{r_1} \left( \sum_{i=1}^{n_1} \phi_1^{(i)}(x_1) \mathbf{D}_1[i, l_1] \right)^2 \right) \lambda_1(x_1), \quad (2.47)$$

where  $\mathbf{D}_1[i, l_1] = \mathbf{B}_1[\alpha_0, i, l_1]$  and  $\alpha_0 = 1$ , and similarly in the last dimesnion

$$f_{X_n}(x_n) = \frac{1}{z} \left( \gamma' \prod_{i=1}^{d-1} \lambda_i(\mathcal{X}_i) + \sum_{l_{n-1}=1}^{r_{n-1}} \left( \sum_{i=1}^{n_1} \phi_1^{(i)}(x_1) \mathbf{D}_n[l_{n-1}, i] \right)^2 \right) \lambda_n(x_n), \quad (2.48)$$

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





# Appendices



# References

- [1] C. Readings. *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] Charles W. Champ and Andrew V. Sills. “The Generalized Law of Total Covariance”. In: *preprint* (2022). URL: <https://arxiv.org/abs/2205.14525>.
- [8] Jari P. Kaipio and Erkki Somersalo. *Statistical and Computational Inverse Problems*. New York: Springer-Verlag New York, 2005.
- [9] 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.
- [10] 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.
- [11] Gareth O. Roberts and Jeffrey S Rosenthal. “General state space Markov chains and MCMC algorithms”. In: *Probability Surveys* 1 (2004), pp. 20–71.
- [12] Charles J Geyer. “Practical markov chain monte carlo”. In: *Statistical science* (1992), pp. 473–483.
- [13] 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.
- [14] Johnathan M Bardsley. “MCMC-based image reconstruction with uncertainty quantification”. In: *SIAM Journal on Scientific Computing* 34.3 (2012), A1316–A1332.
- [15] 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.

- [16] Felipe Acosta, Mark L Huber, and Galin L Jones. “Markov chain Monte Carlo with linchpin variables”. In: *preprint* (2014). URL: <https://arxiv.org/abs/2205.14525>.
- [17] 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>.
- [18] Colin Fox et al. “Grid methods for Bayes-optimal continuous-discrete filtering and utilizing a functional tensor train representation”. In: *Inverse Problems in Science and Engineering* 29.8 (2021), pp. 1199–1217.
- [19] 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.
- [20] 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.
- [21] Ivan V Oseledets. “Tensor-train decomposition”. In: *SIAM Journal on Scientific Computing* 33.5 (2011), pp. 2295–2317.