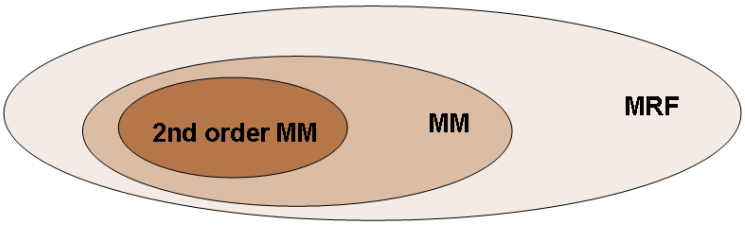
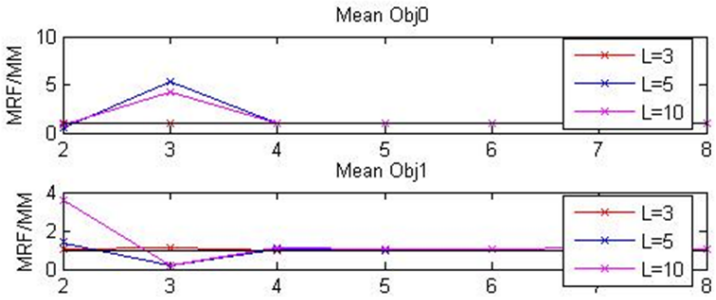
Norsk Regnesentral
NORWEGIAN COMPUTING CENTER

Note

Second order Markov mesh models described as Markov Random Fields





Note no SAND/07/06

Author Heidi Kjønsberg and Ingeborg Ligaarden

Date 18th December 2006

Norsk Regnesentral
NORWEGIAN COMPUTING CENTER

Norwegian Computing Center

Norsk Regnesentral (Norwegian Computing Center, NR) is a private, independent, non-profit foundation established in 1952. NR carries out contract research and development projects in the areas of information and communication technology and applied statistical modeling. The clients are a broad range of industrial, commercial and public service organizations in the national as well as the international market. Our scientific and technical capabilities are further developed in co-operation with The Research Council of Norway and key customers. The results of our projects may take the form of reports, software, prototypes, and short courses. A proof of the confidence and appreciation our clients have for us is given by the fact that most of our new contracts are signed with previous customers.

Norsk Regnesentral
Norwegian Computing Center
Postboks 114, Blindern
NO-0314 Oslo, Norway

Besøksadresse
Office address
Gaustadalléen 23
NO-0373 Oslo, Norway

Telefon · telephone
(+47) 22 85 25 00
Telefaks · telefax
(+47) 22 69 76 60

Internett · internet
www.nr.no
E-post · e-mail
nr@nr.no

Title	**Second order Markov mesh models described as Markov Random Fields**
Author	**Heidi Kjønsberg and Ingeborg Ligaarden**
Date	18th December 2006
Publication number	SAND/07/06

Abstract

We consider the problem of formulating Markov Mesh models as Markov Random field. Since Markov Mesh models are a subclass of Markov Random Fields, this can in principle always be done. In these notes we explore the details of the parameter translation for the case of a 1D stationary Markov Mesh model consisting of a homogeneous external field and general two-particle interactions. We show theoretically that the corresponding Markov Random Field includes also higher order interactions, the complexity of the interactions being limited by the neighbourhood structure. Explicite formulas and recursive algorithms expressing the MRF parameters in terms of the independent Markov Mesh parameters are provided. A matlab implementation of the parameter translation is also described. Using simulations we explore to which extent the higher order interactions in the MRF formulation are necessary in order to reproduce the statistics of the considered 2nd order Markov Mesh models. We find that in general the higher order interactions have significant impact.

Keywords	Markov Mesh models, Markov Random Fields
Target group	All employees, MPS project partners
Availability	Open
Project	Multipoint
Project number	808002
Research field	
Number of pages	50

Contents

1 Introduction

Markov Mesh models have a great advantage over general Markov Random Fields in their being very time efficient for doing simulations. For Markov Mesh models it is only necessary to scan through the grid once, updating each grid cell conditioned on those neighbouring cells that have already being visited during the simulation. MRFs, on the other hand, are commonly simulated using repeated scans through the grid, each update being conditioned on all cells in the Markov neighbourhood. Typically, for MRFs it is necessary to scan the grid hundreds or even thousands of times.

Markov Mesh models have also proved to be easier to fit to a training image than have Markov Random Field models. This has been documented in internal presentations in the Multipoint projec.

However, to combine Markov Mesh models with hard data, for instance well data, is not a trivial task. Since Markov Mesh conditioning only includes cells that have already been visited during the simulation, it is not possible to take into account future data events. When the simulation hits upon hard data it may therefore be that these data points represent a misfit when seen in relation to the simulated cells. Markov Random Fields do not have this problem since the conditioning procedure is symmetric and the simulation is iterative.

The purpose of these notes is to show how to translate a Markov Mesh model into a Markov Random Field formulation. This being done, it will be possible to use the MRF formulation of a (well functioning) Markov Mesh model to perform simulations conditioned on hard data.

The organization of the paper is as follows: Chapter 2 develops the theoretical formulas for the parameter translation, starting from general expressions that are accounted for in Appendices A and B. Chapter 3 presents our Matlab implementation of the parameter translation and MRF simulation, with further details given in Appendix C. Chapter 4 presents results based on three different training images and Markov Mesh and Markov Random Field models generated from these images. Chapter 5 gives a short summary and points out some main ideas for further extensions of this work.

2 Theoretical translation of parameters

We consider an infinite sequence of cells in one spatial dimension and fix one of these cells to be the focus of our study. Label this cell as cell i. We assume a sequential neighbourhood for cell i consisting of the previous L cells:

$$\eta_i = \{i - L, i - L + 1, ..., i - 1\}.$$

The corresponding markov neighbourhood is $\partial_i = \{i - L, ..., i - 1, i + 1, ..., i + L\}$. Fig.2.1 illustrates the relationship between the neighbourhoods.

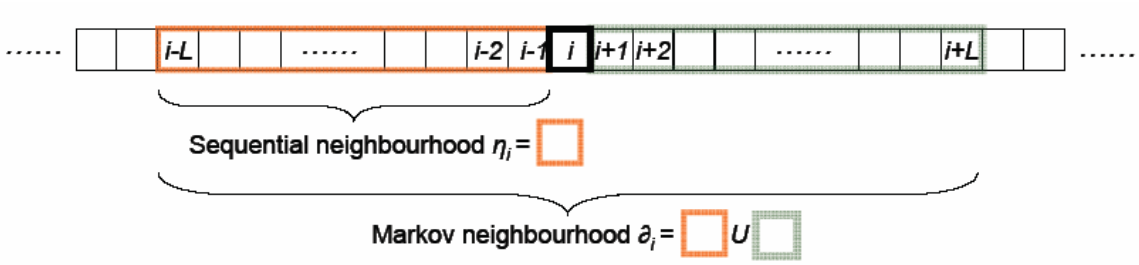


Figure 2.1. Neighbourhoods, sequential and Markov

Basic expressions for the conditional probability in the two models are found in Eqs.2.1 and 2.2. Derivation of the MM expression is shown in detail in Appendix A. The expression for the MRF conditional probability is identical to the general form provided by the Hammersley Clifford theorem, given that the model should be stationary.

MM: (2.1)

$$p(z_i|z_{-i}) = C_i(z_{\partial_i}) \cdot \frac{\exp(\alpha_0 z_i + \sum_{l=1}^{L} \theta_l z_i (z_{i-l} + z_{i+l}))}{\prod_{k=i+1}^{i+L} (1 + \exp(\alpha_0 + \sum_{l=1}^{L} \theta_l z_{k-l}))},$$

MRF: (2.2)

$$p(z_i|z_{-i}) = D_i(z_{\partial_i}) \cdot \exp(F_0 z_i + \sum_{l=1}^{L} F_l z_i (z_{i-l} + z_{i+l}) + \text{h.o.}).$$

$C_i(z_{\partial_i})$ and $D_i(z_{\partial_i})$ are normalization constants. Each of them is specific for cell i, but does not depend on the value of z_i. The external field in the MM model is denoted α_0, while the (2-particle) interactions are called θ_l. For $l \notin \{1, 2, ..., L\}$ the interaction $\theta_l = 0$. Higher order interactions in MRF are in Eq.2.2 not expressed explicitely, this will come in later sections.

Notation becomes easier if we define, for $k \in \{i+1, i+2, ..., i+L\}$

$$\phi_k = \phi_k(\eta_k \backslash i) = \exp(\alpha_0 + \sum_{l=1, l\neq k-i}^{L} \theta_l z_{k-l}), \tag{2.3}$$

$$\psi_k = \phi_k e^{\theta_{k-i}}. \tag{2.4}$$

Notice that neither ϕ_k nor ψ_k depend on z_i. We can now write the MM probability as:

$$\text{MM:} \qquad p(z_i|z_{-i}) = C_i(z_{\partial_i}) \cdot \frac{\exp(\alpha_0 z_i + \sum_{l=1}^{L} \theta_l z_i (z_{i-l} + z_{i+l}))}{\prod_{k=i+1}^{i+L}(1 + \phi_k e^{\theta_{k-i} z_i})}. \tag{2.5}$$

2.1 Normalization constants

If we can find a relation between the normalization constants $C_i(z_{\partial_i})$ and $D_i(z_{\partial_i})$ by assuming some specific value of z_i, but no assumptions on any other facies values, then this relation will hold for all values of z_i. This statement is true because the normalization constants themselves are independent of z_i. So assume $z_i = 0$. Then

$$\text{MM:} \qquad p(z_i = 0|z_{-i}) = C_i(z_{\partial_i}) \prod_{k=i+1}^{i+L} (1 + \phi_k)^{-1}, \tag{2.6}$$

$$\text{MRF:} \qquad p(z_i = 0|z_{-i}) = D_i(z_{\partial_i}). \tag{2.7}$$

The result for the MRF is obtained by assuming that each term in the higher order interactions include one and only one instance of z_i, all other interaction terms are part of $D_i(z_{\partial_i})$. Hence we have

$$D_i(z_{\partial_i}) = C_i(z_{\partial_i}) \prod_{k=i+1}^{i+L} (1 + \phi_k)^{-1}, \tag{2.8}$$

which according to the statement above is valid for all configurations of the grid. We insert this into Eq.2.2. The MM and MRF expressions then read

$$\text{MM:} \qquad p(z_i|z_{-i}) = C_i(z_{\partial_i}) \cdot \frac{\exp(\alpha_0 z_i + \sum_{l=1}^{L} \theta_l z_i (z_{i-l} + z_{i+l}))}{\prod_{k=i+1}^{i+L}(1 + \phi_k e^{\theta_{k-i} z_i})}, \tag{2.9}$$

$$\text{MRF:} \qquad p(z_i|z_{-i}) = C_i(z_{\partial_i}) \cdot \frac{\exp(F_0 z_i + \sum_{l=1}^{L} F_l z_i (z_{i-l} + z_{i+l}) + \text{h.o.})}{\prod_{k=i+1}^{i+L}(1 + \phi_k)}. \tag{2.10}$$

Expressions 2.9 and 2.10 form the foundation of the following discussion.

2.2 External field

External field and interactions with other particles is experienced by cell i when $z_i = 1$. Hence we will now study $p(z_i = 1|z_{-i})$. We have the general expressions

$$\text{MM:} \qquad p(z_i = 1|z_{-i}) = C_i(z_{\partial_i}) \cdot \frac{\exp(\alpha_0 + \sum_{l=1}^{L} \theta_l(z_{i-l} + z_{i+l}))}{\prod_{k=i+1}^{i+L}(1 + \psi_k)}, \tag{2.11}$$

$$\text{MRF:} \qquad p(z_i = 1|z_{-i}) = C_i(z_{\partial_i}) \cdot \frac{\exp(F_0 + \sum_{l=1}^{L} F_l(z_{i-l} + z_{i+l}) + \text{h.o.})}{\prod_{k=i+1}^{i+L}(1 + \phi_k)}. \tag{2.12}$$

The external field is for cell i the only felt potential when all other cells have $z_{-i} = 0$. Let us call this configuration Case 0:

· Case 0: $z_{-i} = 0$

This configuration gives

$$\phi_k^0 = \phi_k(\text{Case 0}) = e^{\alpha_0}.$$

The superscript in ϕ_k^0 is an indicator for Case 0. Similarly we denote $\psi_k^0 = \phi_k^0 e^{\theta_{k-i}} = e^{\alpha_0 + \theta_{k-i}}$. For the probabilities we find

$$\text{MM:} \qquad p(z_i = 1|z_{-i} = 0) = C_i(z_{\partial_i}) \frac{e^{\alpha_0}}{\prod_{k=i+1}^{i+L}(1 + \psi_k^0)}.$$

$$\text{MRF:} \qquad p(z_i = 1|z_{-i} = 0) = C_i(z_{\partial_i}) \frac{e^{F_0}}{\prod_{k=i+1}^{i+L}(1 + \phi_k^0)}.$$

Hence

$$e^{F_0} = e^{\alpha_0} \prod_{k=i+1}^{i+L} \frac{(1 + \phi_k^0)}{(1 + \psi_k^0)}$$

Denote

$$c_0 = \prod_{k=i+1}^{i+L} \frac{(1 + \phi_k^0)}{(1 + \psi_k^0)} = \prod_{l=1}^{L} \frac{(1 + e^{\alpha_0})}{(1 + e^{\alpha_0 + \theta_l})} \tag{2.13}$$

The external field in the MRF formulation finally reads

$$F_0 = \alpha_0 + \ln(c_0). \tag{2.14}$$

Notice that all MM interaction parameters, in combination with α_0, contribute to the correction term $\ln(c_0)$. Generally, the correction term may be large compared to α_0. Hence the external field F_0 may be quite different from α_0, and any intuitive understanding of the Markov Mesh external field is not necessarily applicable to the MRF model (and vice versa). However, in one specific case intuition holds: If all 2-particle interactions θ_l in the MM model are weak compared to the external field α_0, then $c_0 \approx 1$ and $F_0 \approx \alpha_0$. Of course, if for all l the interaction parameters satisfy $||\theta_l|| \to 0$, the particles are independent of one another. A slight modification is provided by the case when $\alpha_0 >> 0$ and the interaction parameters all

satisfy $\alpha_0 + \theta_l >> 0$; in this case $F_0 \approx \alpha_0 - \sum_{l=1}^{L} \theta_l$. If on the other hand we have a situation where $\alpha_0 << 0$ and one of the interaction parameters is positive and strong enough the outweigh α_0, satisfying $\alpha_0 + \theta_{l_0} >> 0$, while the other interactions are weak, then $F_0 \approx -\theta_{l_0}$. We will encounter this situation several times in the examples shown in Chapter 4.

2.3 Two-particle interactions

We now turn to the problem of finding the MRF 2-particle interaction parameters. It is convenient to start by inserting the expression for F_0 (Eq.2.14) into the MRF-expression 2.10, and set $z_i = 1$ to find

$$\text{MRF:} \qquad p(z_i = 1|z_{-i}) = C_i(z_{\partial_i}) e^{\alpha_0} c_0 \frac{\exp(\sum_{l=1}^{L} F_l(z_{i-l} + z_{i+l}) + \text{h.o.})}{\prod_{k=i+1}^{i+L}(1 + \phi_k)}.$$

Define what we will call Case 1 as follows:

- Case 1, l_1-specified: $z_j = 0 \; \forall j \neq i$, except that $z_{i+l_1} = 1$. We assume $l_1 \in \{1, 2, ..., L\}$

That is, in Case 1 all cells in the Markov neighbourhood of i have the value $z_j = 0$, except for one cell. The distance between the nonzero cell and cell i determines the lag l_1. Let

$$\phi_k^{1;l_1} = \phi_k(\text{Case 1, } l_1) = e^{\alpha_0 + \theta_{k-i-l_1}}.$$

and $\psi_k^{1;l_1} = \phi_k^{1;l_1} e^{\theta_{k-i}}$. For Case 1 we then find

$$\text{MM:} \qquad p(z_i = 1|\text{Case 1, } l_1)) = C_i(z_{\partial_i}) \frac{e^{\alpha_0 + \theta_{l_1}}}{\prod_{k=i+1}^{i+L}(1 + \psi_k^{1;l_1})},$$

$$\text{MRF:} \qquad p(z_i = 1|\text{Case 1, } l_1)) = C_i(z_{\partial_i}) \frac{c_0 e^{\alpha_0 + F_{l_1}}}{\prod_{k=i+1}^{i+L}(1 + \phi_k^{1;l_1})}.$$

This gives the requirement

$$e^{F_{l_1}} = e^{\theta_{l_1}} \frac{c_{1;l_1}}{c_0},$$

where

$$c_{1;l_1} = \prod_{k=i+1}^{i+L} \frac{(1 + \phi_k^{1;l_1})}{(1 + \psi_k^{1;l_1})}. \tag{2.15}$$

That is, the 2-particle interaction in the MRF formulation can be expressed by the MM parameters as

$$F_{l_1} = \theta_{l_1} + \ln(\frac{c_{1;l_1}}{c_0}). \tag{2.16}$$

For $l_1 = L$ the result is $F_L = \theta_L$. For all other lags the correction term is non-zero, and its contribution can in general be large relative to θ_{l_1}.

2.4 Higher order interactions

In this section we develop expressions and relations for the higher order interactions in the MRF model.

2.4.1 Combined effect of higher order interactions

By substituting the expressions for F_0 and F_l (Eqs.2.14 and 2.16) into the MRF expression in Eq.2.2, setting $z_i = 1$, and finally require the resulting expression to equal the Markov Mesh conditional probability (Eq.2.11), we find that for the higher order interactions in the MRF model the following relation must hold:

$$e^{\text{h.o. interactions, when } z_i = 1} = f(z_{\partial_i}), \tag{2.17}$$

where

$$f(z_{\partial_i}) = c_0^{-1} \prod_{l=1}^{L} \left(\frac{c_0}{c_{1;l}} \right)^{(z_{i-l}+z_{i+l})} \prod_{k=i+1}^{i+L} \frac{(1+\phi_k)}{(1+\psi_k)}. \tag{2.18}$$

The above relation gives an expression for how the combination of all higher order interactions in the MRF model depends on the MM parameters and the configuration of the grid. The relation can be quite useful as it is. In a simulation it is normally not necessary to know each of the interaction parameters individually, only their combined effect. Eq.2.17 can be used as it is for this purpose. That is, calculate the value of $f(z_{\partial_i})$ and use the result directly in the MRF simulations.

It is, however, a main purpose of these notes to study to which extent the higher order MRF interactions are important in order to reproduce the statistics of its mother Markov Mesh model. For this reason we want to break the relation in Eq.2.17 into components enabling us to find the value of each interaction parameter individually.

2.4.2 Recursive algorithm

The higher order interactions can schematically be written

$$\begin{aligned} \text{h.o. interactions} \;=\; & \sum_{k,l} z_i \left(z_{i+k} z_{i+l} + \sum_{\text{other 3-cliques};k,l} \right) F_{k,l} \\ & + \sum_{k,l,m} z_i \left(z_{i+k} z_{i+l} z_{i+m} + \sum_{\text{other 4-cliques};k,l} \right) F_{k,l,m} \\ & + \ldots + \\ & + z_i \left(z_{i+1} z_{i+2} \ldots z_{i+L} + \sum_{\text{other max-cliques}} \right) F_{1,1,\ldots,1}. \end{aligned} \tag{2.19}$$

Every clique in this expression contains one and only one instance of z_i, since all cliques not containing z_i are part of the normalization factor $D_i(z_{\partial_i})$. Notice also

that for each interaction level (each line in Eq.2.19) we have singled out the cliques belonging to the future neighbourhood $\partial_i \backslash \eta_i$ of cell i. The reason will become clear in a moment.

Consider now a generalization of the cases we considered above; Case M. In Case M there are M particles from the future neighbourhood of cell i that have a nonzero facies indicator. All other cells have the value zero. That is:

· Case M, $l_1, l_2, ..., l_M$-specified: $z_j = 0 \ \forall j \neq i$, except that $z_{i+l_m} = 1, m = 1, 2, ..., M$.

We assume $l_m \in \{1, 2, ..., L\}$ and $l_1 < l_2 < ... < l_{M-1} < l_M$.

Now concentrate on which contributions the left hand side of Eq.2.17 receives for Case M. Since z_i is part of any clique that enters Eq.2.19, the contributing 3-particle interactions are

$$\begin{aligned} & F_{l_1,l_2} + F_{l_1,l_3} + ... + F_{l_1,l_M} + F_{l_2,l_3} + F_{l_2,l_4} + ... + F_{l_2,l_M} + ... + F_{l_{M-1},l_M} \\ = & \sum_{m_1=1}^{M-1} \sum_{m_2=m_1+1}^{M} F_{l_{m_1},l_{m_2}}. \end{aligned}$$

Similarly, the 4-particle interactions contribute with

$$\sum_{m_1=1}^{M-2} \sum_{m_2=m_1+1}^{M-1} \sum_{m_3=m_2+1}^{M} F_{l_{m_1},l_{m_2},l_{m_3}},$$

and so on. Every specific parameter appears only once. No interactions of higher order that $M + 1$ contribute. We then separate the left hand side of Eq.2.17 into two factors, one that contains the interaction of order $M + 1$, the other factor containing all lower order interactions. The factor of the lower order interactions is divided on both sides of the equation so that it in the following appears as a denominator of the right hand side. We then find that Eq.2.17 can be expressed as

$$\begin{aligned} e^{F_{l_1,l_2,...,l_M}} = & \ f(\text{Case M};l_1, l_2, ..., l_M) \\ & \cdot \exp(- \sum_{m_1=1}^{M-1} \sum_{m_2=m_1+1}^{M} F_{l_{m_1},l_{m_2}} \\ & - \sum_{m_1=1}^{M-2} \sum_{m_2=m_1+1}^{M-1} \sum_{m_3=m_2+1}^{M} F_{l_{m_1},l_{m_2},l_{m_3}} \\ & - ... \\ & - \sum_{m1<m2<...<m_{M-1}} F_{l_{m_1}} F_{l_{m_2}} ... F_{l_{m_{M-1}}}) \end{aligned} \tag{2.20}$$

This expression is the main result for the higher order interactions. It gives a recursive algorithm for finding the individual interaction parameters. First find all parameters for 3-particle interactions by using the already found results for the

2-particle interactions. Then find all 4-particle interactions by using the found 3-particle and 2-particle interactions, and so on. The algorithm can schematically be written as:

- for $M = 2 : L$
  - for all combinations $l_1, l_2, \ldots, l_M$ such that $l_1 < l_2 < \cdots < l_M$ and $l_m \in \{1, 2, \ldots, L\}$
    - calculate $f(\text{Case M;}l_1, l_2, ..., l_M)$
    - find the M-particle interaction parameter $F_{l_1,l_2,...,l_M}$ from

$$\begin{aligned} e^{F_{l_1,l_2,...,l_M}} &= f(\text{Case M;}l_1, l_2, ..., l_M) \\ &\cdot \exp(-\sum_{m_1=1}^{M-1}\sum_{m_2=m_1+1}^{M} F_{l_{m_1},l_{m_2}} \\ &-\sum_{m_1=1}^{M-2}\sum_{m_2=m_1+1}^{M-1}\sum_{m_3=m_2+1}^{M} F_{l_{m_1},l_{m_2},l_{m_3}} \\ &-... \\ &-\sum_{m1<m2<...<m_{M-1}} F_{l_{m_1}} F_{l_{m_2}} ... F_{l_{m_{M-1}}}) \end{aligned}$$

  - end
- end

2.4.3 Helpful relations

The functional value of f must in each case be found separately. To easen the computational cost of this, one can benefit from the following recursive relationsships. Generally, for Case M we have that the value of f can be expressed as

$$f(\text{Case M;}l_1, l_2, ..., l_M) = c_0^{-1} \left(\frac{c_0}{c_{1;l_1}}\right)\left(\frac{c_0}{c_{1;l_2}}\right) ... \left(\frac{c_0}{c_{1;l_M}}\right) \prod_{k=i+1}^{i+L} \frac{1+\phi_k(\text{Case M;}l_1, l_2, ..., l_M)}{1+\psi_k(\text{Case M;}l_1, l_2, ..., l_M)}.$$

We denote

$$c_{M;l_1,l_2,...,l_M} = \prod_{k=i+1}^{i+L} \frac{1+\phi_k(\text{Case M;}l_1, l_2, ..., l_M)}{1+\psi_k(\text{Case M;}l_1, l_2, ..., l_M)}. \tag{2.21}$$

For Case M-1, when specified by the same lags $l_1, l_2, ..., l_{M-1}$ as Case M (but not lag l_M), we similarly have

$$f(\text{Case M-1;}l_1, l_2, ..., l_{M-1}) = c_0^{-1} \left(\frac{c_0}{c_{1;l_1}}\right)\left(\frac{c_0}{c_{1;l_2}}\right) ... \left(\frac{c_0}{c_{1;l_{M-1}}}\right) c_{M-1;l_1,l_2,...,l_{M-1}},$$

and hence

$$\begin{aligned} & f(\text{Case M;}l_1, l_2, ..., l_M) \\ = \; & f(\text{Case M-1;}l_1, l_2, ..., l_{M-1}) \cdot \left(\frac{c_0}{c_{1;l_M}} \right) \cdot \left(\frac{c_{M;l_1,l_2,...,l_M}}{c_{M-1;l_1,l_2,...,l_{M-1}}} \right). \end{aligned} \tag{2.22}$$

This relation can be used to save computational resources. Finally, to easen the calculation of $c_{M;l_1,l_2,...,l_M}$ we mention the relation

$$\phi_k(\text{Case M;}l_1, l_2, ..., l_M) = \phi_k(\text{Case M-1;}l_1, l_2, ..., l_{M-1}) e^{\theta_{k-i-l_M}}, \tag{2.23}$$

which is valid for $k \in \{i + l_M + 1, ..., i + L\}$.

3 Implementation

This chapter is devoted to the practical implementation of the theoretical results of the previous chapter. The implemetation can be divided into the following steps

1. estimate MM parameters
2. translate MM parameters to MRF parameters
3. simulate MM and MRF (Gibb's sampler) to compare results

The estimation of MM parameters and simulation of MM-models in 1-D was already implemented elsewhere in the Multipoint project, and will not be described here. While this chapter describes the overall methods used, the detailed documentation of all functions described in this chapter is found in appendix C. The implementation is done in Matlab.

3.1 Implementation of parameter translation

For external field and 2-particle interactions we have the expressions in Eqs.2.14 and 2.16, which are straightforward to implement. For the higher order interactions we use the relation in Eq.2.20, and the resulting recursive algorithm:

- for $M = 2 : L$
  - for all combinations $l_1, l_2, \ldots, l_M$ such that $l_1 < l_2 < \cdots < l_M$ and $l_m \in \{1, 2, \ldots, L\}$
    - calculate $f(\text{Case M};l_1, l_2, ..., l_M)$
    - find the M-particle interaction parameter $F_{l_1,l_2,...,l_M}$ from

$$
\begin{aligned}
e^{F_{l_1,l_2,...,l_M}} \quad = \quad & f(\text{Case M};l_1, l_2, ..., l_M) \\
& \cdot \exp(-\sum_{m_1=1}^{M-1} \sum_{m_2=m_1+1}^{M} F_{l_{m_1},l_{m_2}} \\
& - \sum_{m_1=1}^{M-2} \sum_{m_2=m_1+1}^{M-1} \sum_{m_3=m_2+1}^{M} F_{l_{m_1},l_{m_2},l_{m_3}} \\
& - \ldots \\
& - \sum_{m1<m2<\ldots<m_{M-1}} F_{l_{m_1}} F_{l_{m_2}} \ldots F_{l_{m_{M-1}}})
\end{aligned}
$$

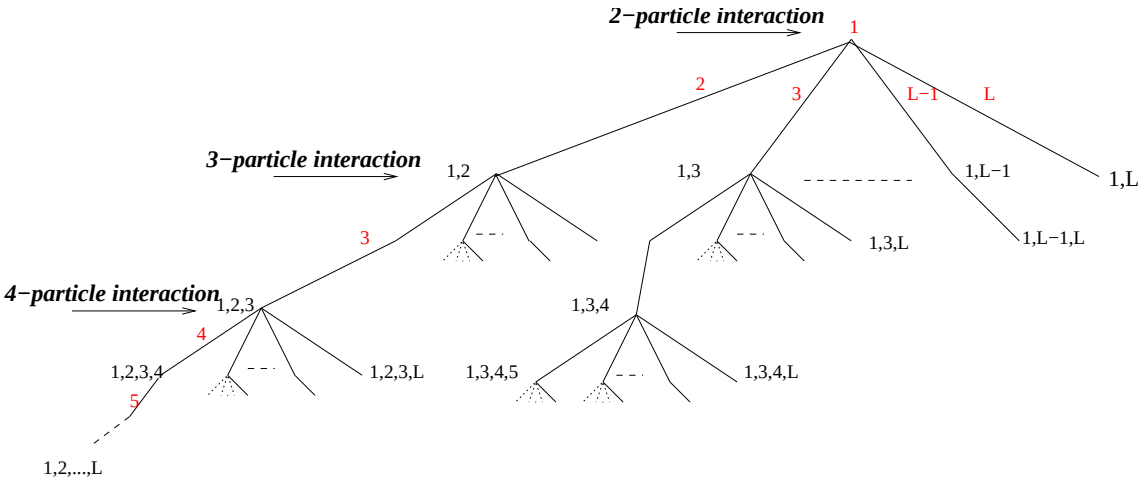


Figure 3.1. Combinations with $l_1 = 1$

- end

· end

Main challenges in the algorithm

1. For each $M \in [2, L]$ find all combinations $l_1, l_2 \ldots, l_M$ such that $l_1 < l_2 < \cdots < l_M$ and $l_m \in 1, 2, \ldots, L \quad \forall\, m$.
2. For each configuration with M nonzero particles, find all subconfigurations with $M - 1, \ldots, 2$ nonzero particles.
3. Save the interaction parameters $F_{l_1,l_2,\ldots,l_M}$ in such a way that they are easy to access during the algorithm and afterwards.
4. Try to benefit from the relations on f and ϕ_k, see Eqs.2.22 and 2.23.

Solutions

1. **Finding combinations:** Figure 3.1 displays all combinations with $l_1 = 1$.

   For each M we start with the configuration $l_1 = 1, l_2 = 2, \ldots, l_M = M$. After computing $F_{1,2,\ldots,M}$ we use the function **permutation.m**[1] to generate the next configuration. This is repeated till the last configuration with M nonzero particles is reached. There are $\binom{L}{M}$ different configurations for each M.
2. **Finding subinteractions:** In Eq.2.20 we see that in order to compute the interaction parameter for a given configuration K of M particles, we need the interaction parameter of all subconfigurations, meaning that we need to find all combinations of $M - 1, M - 2, \ldots, 2$ particles $\in K$. This is done in the function **sumSubInteract.m**.

1. We acknowledge Ragnar Hauge at NR for letting us use this function.

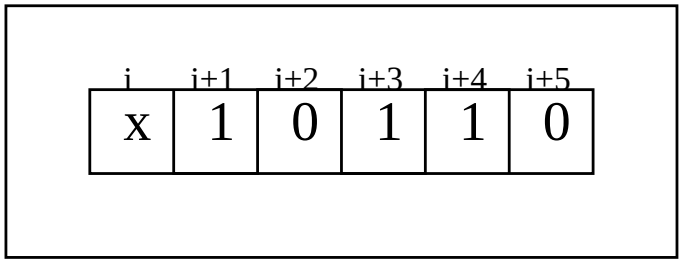


Figure 3.2. Example of configuration of future neigbourhood. The index of this configuration is $2^0 + 2^2 + 2^3 = 13$, since counting from cell i, cell $1, 3$ and 4 are nonzero.

There are $\sum_{m=2}^{M} \binom{M}{m} \sim 2^M$ different subconfigurations per configuration of M particles. For each of these subconfigurations we add in the corresponding parameter value that has previously been calculated. Hence a total of $\sum_{M=2}^{L} \binom{L}{M} 2^M \sim 3^L$ additions are needed in order to take into account all subinteractions for all configurations for all M.

3. **Saving the parameters:** Each interaction parameter $F_{l_1,l_2,\ldots,l_M}$ corresponds to a unique configuration of the future neigboorhood. We can look at each configuration as a unique binary number which we call the index of the configuration. The different configurations corresponds to $2^L - 1$ different nonzero indexes. We use this to make a table of all $F_{l_1,l_2,\ldots,l_M}$ ordered by the indexes. See Fig.3.2 for an example of indexing. The external field parameter F_0 is given index 2^L.

4. **Using the relations on f and ϕ_k**

   Using the binary index of a configuration, we can also make a table of values of the function f, Eq.2.18, evaluated on each configuration. Then we can use the relation between $f(\text{Case M}; l_1, l_2, \ldots, l_M)$ and $f(\text{Case M-1}; l_1, l_2, \ldots, l_{M-1})$ given in Eq.2.22 to compute the higher order interaction parameters. This enables us to find the corresponding interaction parameters for a configuration of the future neigboorhood in a straightforward way.

   For each configuration of M particles the complexity of finding the function $f_{M;l_1,l_2,\ldots,l_M}$ is almost identical to the complexity of finding the factor $c_{M;l_1,l_2,\ldots,l_M}$ in Eq.2.22, which is again of the order LM. The total complexity of finding all f-values is then approximately $\sum_{M=2}^{L} \binom{L}{M} LM \sim L^2 2^{L-1}$.

   We can also benefit from the relations on ϕ_k from Eq.2.23. In order to use this relation we must save the ϕ_ks for each configuration of the grid. This is done in a $((2^L - 1) \times L)$ matrix **phi**, where $phi(\text{binary index of config}, :)$ gives ϕ_k for $k \in i+1, i+2, \ldots, i+L$ for the configuration, given its binary index. When L is small, using the relation (2.23) will not save much running time, but for larger Ls it is useful.

Computational complexity of parameter translation

The overall complexity of the parameter translation is of the order $3^L + L^2 2^{L-1}$. The first term is due to the need to sum up all parameters corresponding to sub-configurations, see point 2 above. The second term represents the complexity of evaluating the values of the function f, see point 4 above.

Useful reformulations used in the implementation

To find both 2-order interactions and higher order interactions we need $c_{1;l_1}$, defined in (2.15), for $l_1 \in 1, 2, \ldots, L$. Recall that:

$$\phi_k^{1;l_1} = \phi_k(\text{Case 1}, l_1) = e^{\alpha_0 + \theta_{k-i-l_1}}$$

and

$$\psi_k^{1;l_1} = \phi_k^{1;l_1} e^{\theta_{k-i}}. \tag{3.1}$$

Equation (2.15) can be written as

$$\begin{aligned}
c_{1;l_1} &= \prod_{k=i+1}^{i+L} \frac{1+\phi_k^{1;l_1}}{1+\psi_k^{1;l_1}} \\
&= \prod_{k=i+1}^{i+L} \frac{1+e^{\alpha_0+\theta_{k-i-l_1}}}{1+e^{\alpha_0+\theta_{k-i-l_1}+\theta_{k-i}}} \\
&= \prod_{l=1}^{L} \frac{1+e^{\alpha_0+\theta_{l-l_1}}}{1+e^{\alpha_0+\theta_{l-l_1}+\theta_l}} \\
&= \prod_{l=1}^{l_1} \frac{1+e^{\alpha_0}}{1+e^{\alpha_0+\theta_l}} \cdot \prod_{l=l_1+1}^{L} \frac{1+e^{\alpha_0+\theta_{l-l_1}}}{1+e^{\alpha_0+\theta_{l-l_1}+\theta_l}} \\
&= \prod_{l=1}^{l_1} \frac{1+e^{\alpha_0}}{1+1+e^{\alpha_0+\theta_l}} \cdot \prod_{n=1}^{L-l_1} \frac{1+e^{\alpha_0+\theta_n}}{1+e^{\alpha_0+\theta_n+\theta_{n+l_1}}}
\end{aligned} \tag{3.2}$$

since $\theta_l = 0$ for $\theta \notin 1, 2, \ldots, L$.

We can also reformulate ϕ_k to clarify which parameters to use in the computation. Let $J = l_1, l_2, \ldots, l_M$. Looking closer at $\phi_k(\text{Case M} : l_1, l_2, \ldots, l_M)$ we find

$$\phi_k(\text{Case M} : l_1, l_2, \ldots, l_M) = \exp(\alpha_0 + \sum_{l \in J : k-i > l} \theta_{k-i-l}). \tag{3.3}$$

3.2 Gibb's sampler for MRF

Let

$$g(1|z_{\delta_i}) = \exp(F_0 + \sum_{l=1}^{L} F_l(z_{i-l} + z_{i+l}) + h.o(z_{\delta_i})) \tag{3.4}$$

be the potential of cell z_i. Gibb's sampling algorithm can schematically be written:

```
nx=N+200;
grid=[zeros(1,100) randGrid zeros(1,100)];

for t=1:T
  for  all cells in  (L+1:nx-L) in random order %update cell i

    g0=1; %non-normalized prob for facies=0
    find g1=g(1|z_delta_i)

    normConst=1/(g0+g1);
    p0=unif(0,1)/normConst;

    if p0 < g0 %update grid
      grid(i)=0;
    else
      grid(i)=1;
    end
  end
end %end gibbs simulation
```

The grid on which we simulate is padded in each end to avoid boundary effects. For the complete algorithm, see **gibbsMRF.m**. If the size N of the grid is large, computing $g(1|z_{\delta_i})$ for each cell i, T times will be a slow process. Instead we first find the value of

$$h(1|z_{\delta_i}) = \sum_{l=1}^{L} F_l(z_{i-l} + z_{i+l}) + h.o(z_{\delta_i})) \tag{3.5}$$

for all possible neighbourhood configurations and store these in a table.

Calculating the potential of a configuration

As suggested above, calculating the potential of all configurations before we run the Gibb's sampler will save us a lot of time. We have $(2^{2L} - 1)$ different possible configurations of the neigboorhood if we remove cell i, and the binary value of the configuration (without cell i) is therefore used as the index of the table. See figure 3.3.

Pseudo-code for making the table:

```
for all possible configurations c of a neighbourhood z_delta_i ,:
  calculate h(1,c)
  save potential for later use
```

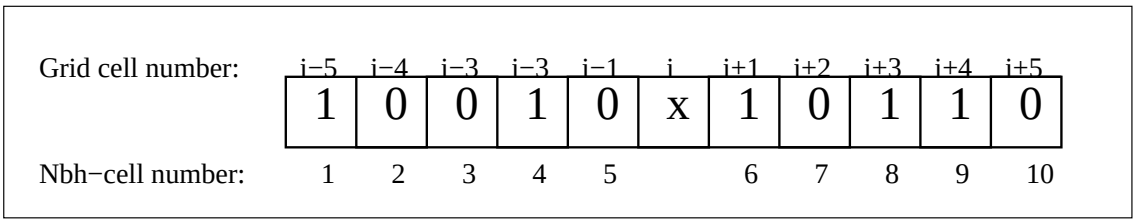


Figure 3.3. Example of grid configuration. The index of this configuration is $2^0 + 2^3 + 2^5 + 2^7 + 2^8 = 425$, since counting from l.h.s, not including cell i itself, the cells $1, 4, 6, 8$ and 9 are nonzero.

```
end
```

However, since our main goal is to test if the higher order interactions give a significant contribution, we make it possible to ignore some of the higher order interactions and calculate an approximated potential. Let

$$\rho(z_{\delta_i}, n) \tag{3.6}$$

be an approximation to $h(1|z_{\delta_i})$ using all particle interactions up to and including n-particle interactions. Our new algorith is:

```
for all possible configurations, c, of a neighbourhood z_delta_i ,:
  calculate rho(c, n)
  save potential for later use
end
```

When L is large, calculating the potential from scratch each time will be a time consuming process. For one given configuration we have the following relation:

$$\rho(c, n+1) = \rho(c, n) + \text{“}(n+1) - \text{particle interactions of c”}.$$

This means that if we have $\rho(c, n)$ for some n and want to find $\rho(c, m)$ for $m > n$ we only need to add the $\{n+1, n+2, \ldots, m\}$-particle interactions to get $\rho(c, m)$. The final algorithm then is:

```
for all possible configurations of the neighbourhood:
   sum n+1,n+2,...,m-particle and add rho(c,n)
   rho(c,m)=rho(c,n)+ {n+1,n+2,...,m}-particle interactions
   save rho(c,m)
end
```

This algorithm is found in **findSumInteractOfConf.m**.

3.3 Statistics

In order to compare simulations (see the introduction to chapter 4 and also section 4.1 for more details on what we actually want to compare), our main concern is the size of the objects in the simulated grids. We would like to know; size of 0-objects and 1-objects, mean of sizes and variance of size. We also find number of 0-objects and 1-objects. All this is done in the function **statisticsOnObj.m**.

4 Results from simulations

Having translated the estimated MM parameters to MRF parameters we want to compare simulations of the MM model and the MRF model. According to theory, when all parameters for the MRF model are taken into account, the statistics produced by the MRF should be identical to that of the MM model. Hence a comparison between the full MRF model and the MM model can be viewed merely as a check that the implementation is correct. It is on the other hand of great interest to explore to which extent it is necessary to include higher order interactions in the MRF model. If it could be shown that only interactions up to a certain order are necessary in order to reproduce the MM statistics, then further exploitation of the MRF, for instance for doing simulations conditioned on well data, could be performed with a simplified MRF model. This chapter gives an initial analysis of this topic. We also include a brief comparison between the statistics of the MM model and that of the original training image, although this is not the main topic of these notes.

4.1 Training images and model specifications

We have used three different training images, each TI consisting of 10^5 cells. Mean object lengths in the three TIs are listed in Table 4.1. The objects' lengths represent very simple combinations of long and short object sizes. We have chosen mean object lengths of 50 and 10. The small variability is chosen in order to have TIs with clearly identifiable characteristics.

For each TI we have estimated MM parameters for the neighbourhood sizes $L = 3$ and $L = 5$. This means we have created two MM models per TI. For TI 2 we have in addition made an MM model characterized by $L = 10$. Sequential simulations were performed for each of these Markov mesh models, and statistics collected with the purpose of comparing it to statistics from MRF simulations.

TI #	0-object size	1-object size
1	49.99 ± 1.23	49.97 ± 1.22
2	10.01 ± 1.25	49.97 ± 1.23
3	9.99 ± 1.24	9.99 ± 1.23

Table 4.1. Training Images' mean object lengths, with standard deviation

MM model	MRF model, highest interaction order									
	2	3	4	5	6	7	8	9	10	11
TI1, $L = 3$	x	x	x							
TI1, $L = 5$	x	x	x	x	x					
TI2, $L = 3$	x	x	x							
TI2, $L = 5$	x	x	x	x	x					
TI2, $L = 10$	x	x	x	x	x	x	x	-	-	-
TI3, $L = 3$	x	x	x							
TI3, $L = 5$	x	x	x	x	x					

Table 4.2. Models considered

For each of the two/three MM models per TI the MM parameters have been translated into MRF parameters, using the theory and implementation described in earlier chapters. For each MM model the full MRF model consists of all interaction parameters, i.e. up to and including order $L+1$. In addition we have defined approximate MRF models by including interactions up to and including order $2, 3, ..., L$ (for the $L = 10$ we only considered interactions up to order 8). Table 4.2 lists the different models considered. For each MRF model, full or approximative, we have then carried out simulations. Each simulation was done for a grid consisting of 2000 cells, with a total number of 50,000 Gibb's iterations per cell and sampling every 50th iteration. The statistics studied are mean object length for 0-objects and for 1-objects, and standard deviation of object lengths.

4.2 Brief comparison of MM to TI statistics

Figure 4.3 displays mean object lengths and standard deviation of object lengths as found by the sequential simulation of the Markov Mesh model for TI 1 and $L = 3$. Comparing these histograms to the characteristics of TI 1, where mean object lengths were given by

$$\begin{aligned} \text{length(0-objects)} &= 49.99 \pm 1.23, \\ \text{length(1-objects)} &= 49.97 \pm 1.22, \end{aligned}$$

it is clear that mean object lengths are well reproduced by the Markov Mesh model. Standard deviations of TI object lengths are however not well reproduced by the MM model, the MM standard deviations being much higher than the variability in the training image.

Similar results are found for all TIs and MM models ($L = 3, 5, 10$) considered. The reason for the dicrepancy between MM statistics and TI statistics is likely to

be the choice of model, i.e. the restriction to 2-particle interactions in the Markov Mesh models. It is of course of interest to find Markov Mesh models that better represent the characteristics of the TIs. This is however not to be followed up here, as the main purpose of this paper is to compare the different MRF models to their mother MM model. For this purpose the defined Markov Mesh models are fine.

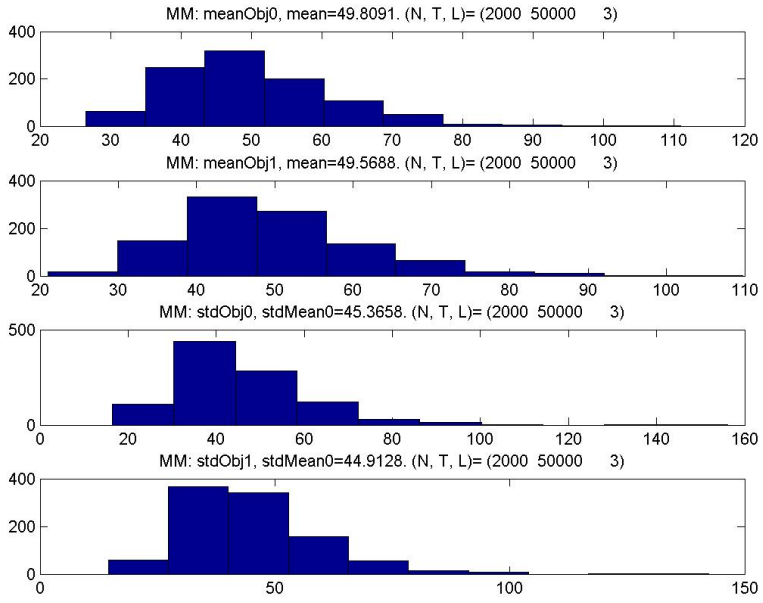


Figure 4.1. Mean object lengths and object length standard deviations, Markov Mesh model estimated from TI 1, L=3

4.3 Comparing MRF to Markov Mesh

We now break away from comparing the simulation results to the TI statistics. Instead we focus on the comparison of different MRF models to their mother Markov Mesh model.

4.3.1 Detailed comparison for the model defined by TI 1, $L = 5$

For a neighbourhood size determined by $L = 5$ we have studied five different MRF models. These models consist of interactions up to and including those of order 2, 3, 4, 5, and 6, respectively, see Table 4.2. The model with interactions of order 6 is the most general MRF model that exists for this neighbourhood, it is referred to as the full MRF model.

Consider first the results obtained for the full MRF. Fig.4.2 compares the statistics of the MRF (right) to the MM (left) for this case. We observe that the his-

tograms are quite similar. This is as expected, since the full MRF model according to theory is identical to the MM model.

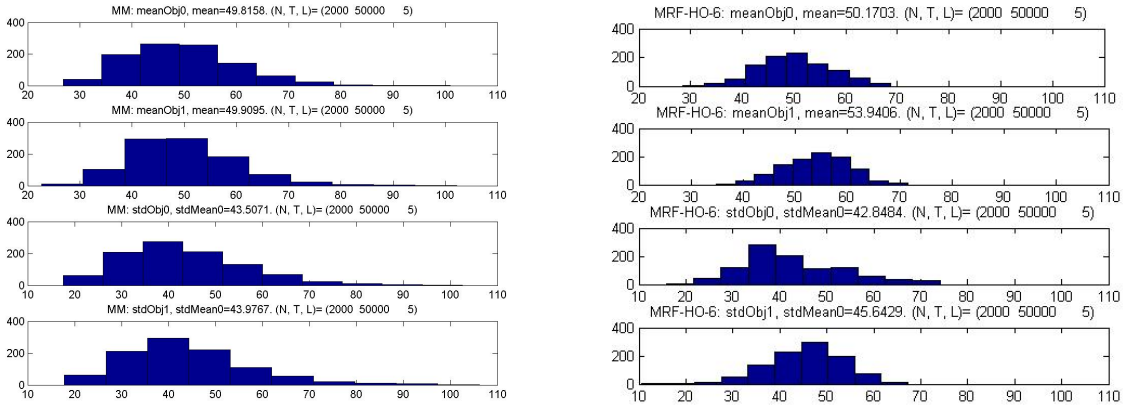


Figure 4.2. Mean object lengths and object length standard deviations. Markov Mesh model left, Markov Random Field for h.o.i.=6 to the right. TI 1, L=5

We next look into whether the MRF model defined only in terms of the 2-particle interactions can reproduce the statistics of the MM model. Fig.4.3 displays histograms analogous to those in Fig.4.2, but obtained by using the 2nd order MRF model. The histograms to the left in the figure are those for the MM model. The figure shows very clearly that the statistics of this MRF does not match the MM statistics at all. The MRF model produces objects whose mean length is eather large (1-objects, mean length is around 100) or quite small (0-objects, mean length is around 12). For the MM model on the other hand, both object types have similar mean lengths, around 50. The mean standard deviation is, for each object type in the MRF model, of the same magnitude as the mean length. This behaviour of variability is analogous to the MM results.

Fig.4.4 illustrates the development of the grid throughout simulations for the $L = 5$, 2nd order MRF (left) and full MRF (right). Each row shows a snapshot of the grid configuration. The initial grid configurations for the two simulations were identical. Starting from the first measurement, shown in the uppermost row of the figure, the two models clearly develop different mixes of object lengths as the simulation goes on (moving downward in the figure). The 2nd order model has a strong dependency between subsequent measurements and is quite unwilling to develop anything but the small 0-objects and the large 1-objects. The full MRF model is different. In this case there is more variability in the object lengths, and mixing of different configurations throughout the simulation is better.

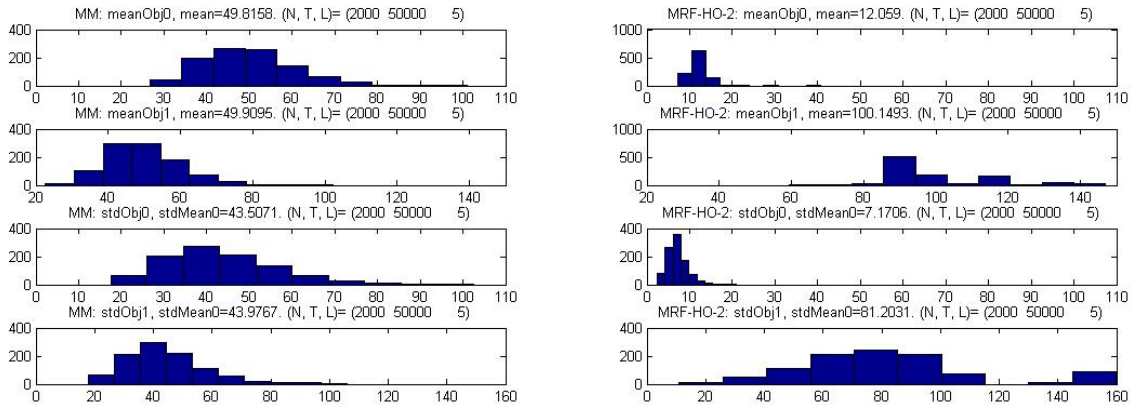


Figure 4.3. Mean object lengths and object length standard deviations. Markov Mesh model left, Markov Random Field for h.o.i.=2 to the right. TI 1, L=5

The need for higher order interactions in the MRF model can be understood in terms of the interaction parameters. Fig.4.5 lists the estimated MM parameters and the MRF parameters that arise as a result of the parameter translation.

Comparing the external field and 2nd order interaction parameters for the MM and MRF models it becomes clear that the correction terms in Eqs.2.14 and 2.16 have an effect. For instance is the MRF external field F_0 approximately equal to $-\theta_1$, in agreement with the comment made at the end of chapter 2.2. Also the 2nd order interactions of lag 3 and 4 differ quite a lot between MM to MRF, whereas interactions at the shorter lags 1 and 2 are rather similar. For the highest lag, 5, the interactions are identical, in accordance with theory (see Eq.2.16).

Now compare the magnitude of MRF higher order interactions (order 3, 4, 5, 6) to the MRF 2nd order interactions. If all higher order interactions were small compared to the 2nd order interactions, the statistics would not change very much as more and more interactions were included in the model. But of the values listed in Fig.4.5, the interaction parameter $F_{1,3,4}$ has a magnitude indicating that it may have a significant impact on the results. This is the reason why the results for the MRF models in Figs.4.2 and 4.3 are so different. From the values listed in Fig.4.5 it is to be expected that the three MRF models characterized by interactions of order 4, 5, and 6, respectively, will give quite similar statistics. Some of the 3-particle interactions are large enough that they might produce some changed behaviour when compared to the 2nd order model. But the big impact, making the MRF behave in the same way as the MM model, is expected to set in when 4-particle interactions are included. We will see in the next section that this is indeed the case.

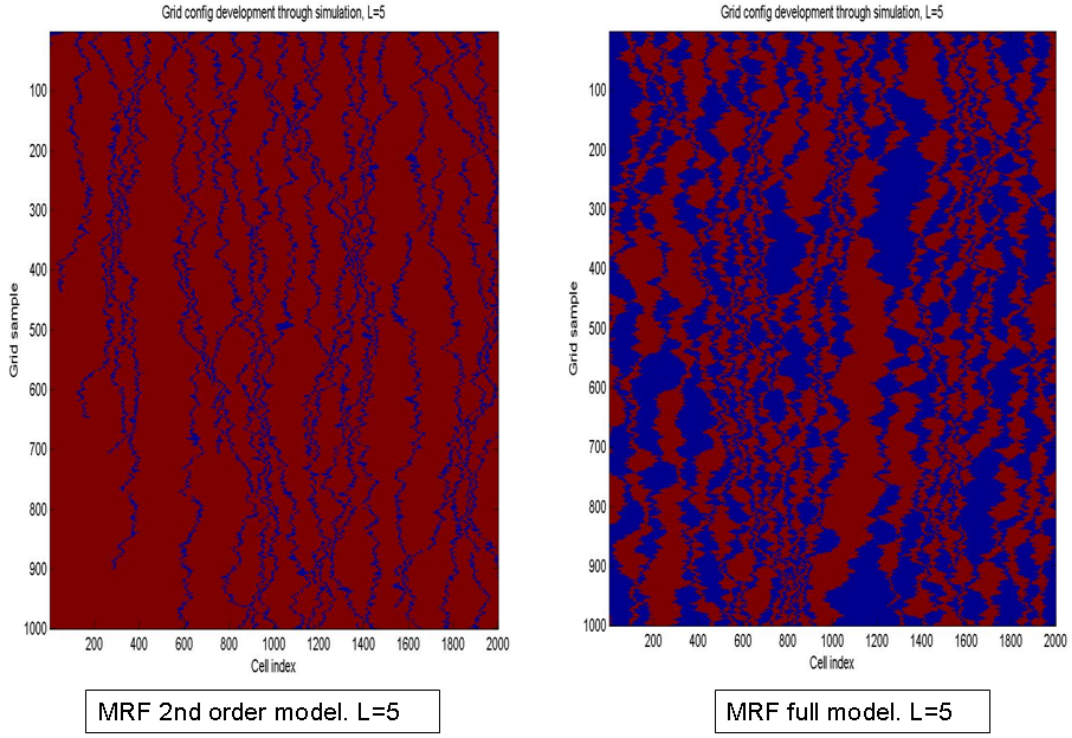


Figure 4.4. MRF grid snaphots during simulation. The vertical axis counts snapshots through the simulation, in units of 50 Gibb's updates per cell. The horisontal axis show the cell number in the grid. MRF 2nd order model to the left, MRF full model to the right. TI 1, L=5

4.3.2 Comparison for all models

Fig.4.6 summarises the comparison between all the different MRF models and each of their mother MM model. The figure consists of three subfigures, each subfigure being labelled by the training image that was used to define the Markov Mesh models. Each of these subfigures in turn consists of four panes, representing the four quantities mean object lengths for 0-objects and 1-objects, and mean standard deviation for 0-objects and 1-objects, respectively. Now focus on one of these quantities and a single training image, thus identifying one single figure pane. For each training image there are several MM models, each defined by the neighbourhood size L. Recall the models listed in Table 4.2. In the given figure pane there is one graph per MM model. Each point on the graph is found by comparing the MRF statistics to the statistics of its mother MM model. More specifically, we have computed the given quantity's mean value for the MRF model and divided by the mean value obtained for its mother MM model. The horisontal axis represents in all cases the highest interaction order included in the considered MRF model. Each figure pane also includes a horisontal line at the value 1;

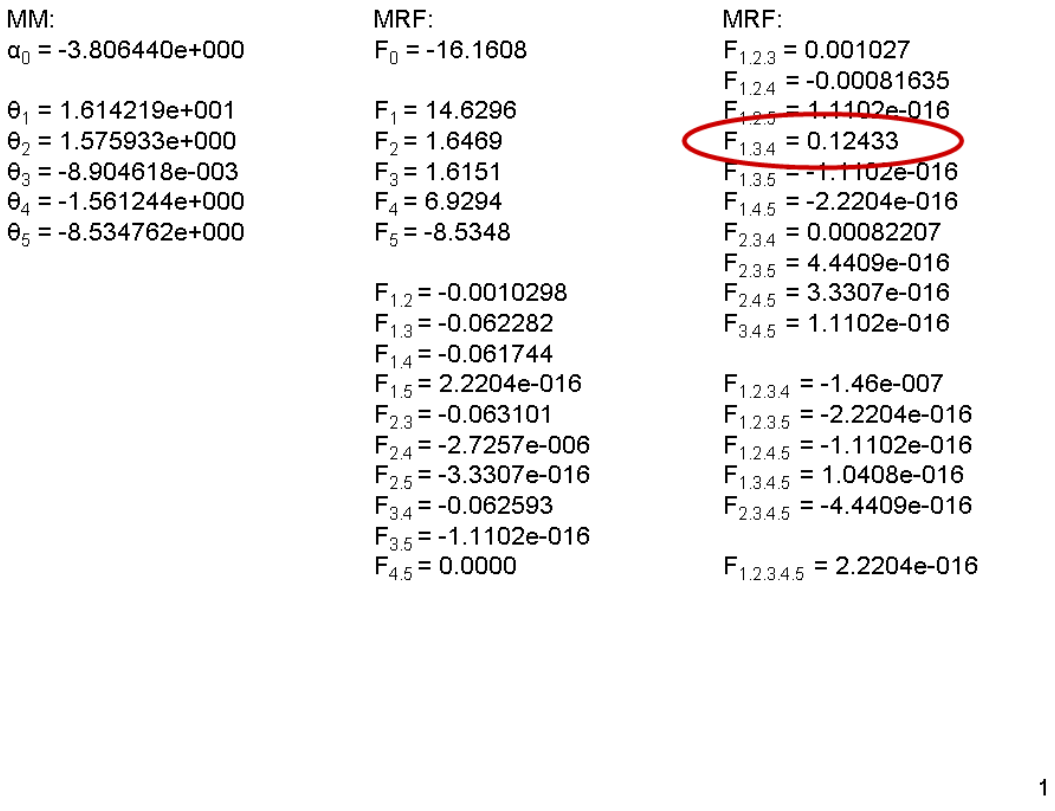


Figure 4.5. MM parameters (left column) and MRF parameters. TI 1, $L = 5$

if and only if the fraction "MRF/MM" equals 1 do the two models have identical mean values. The graph thus shows how the inclusion of higher and higher order interactions in the MRF model affects the statistics of the MRF as compared to the MM model. For each graph in the figure the rightmost point, representing the behaviour of the full MRF models, is expected to be very close to 1.

Some observations are:

- The graph patterns are almost independent of the TI. Be aware that the MM models produce different results for each TI, the mean object lengths observed in the MM statistics being similar to the mean object lengths of the TI from which it was derived, see Table 4.1. Thus the fact that the graph patterns in Fig.4.6 are almost independent of the TI is merely a statement that the higher order interaction have analogous effects on the MRF's approximation to the full model, independent of the object patterns seen.

- The full MRF models produce statistics that is very close to the MM statistics

- When $L > 3$, as soon as the 4-particle interactions are included the MRF statistics is very close to the MM statistics. This observation corresponds to the expectations we developed based on the interaction parameters in

Fig.4.5. Fig.4.6 shows that this holds for all TIs.

- For $L = 3$ it is enough to include 2nd order interactions in the MRF model, neither 3- nor 4-particle interactions make much of a difference to the statistical results.

- For $L = 5$ and $L = 10$ the 2nd order MRF model tends to make shorter 0-objects and longer 1-objects than the MM model, while the model that includes 3-particle interactions tends to make the 0-objects longer and the 1-objects shorter. This is true for all three training images.

The fact that the graph patterns are almost independent of the TIs can be understood by looking at the MM parameter values. Fig.4.7 shows the estimated parameters for all MM models considered. It is a striking fact that when compared across TIs, the relative strengths of the parameters are very similar. For instance, for $L = 3$ the interactions θ_1 and θ_3 are quite strong compared to the lag 2 interaction θ_2, while the ratio of for instance θ_3/α_0 is also almost constant across the TIs. Similar regularities are seen for the $L = 5$ MM models. It is a general trend for the $L = 3$ and $L = 5$ models that interactions at the shortest and longest lags are significantly stronger (i.e. large absolute value) than the interactions at the other lags. For $L = 10$ there is a similar pattern, although not so striking. With very similar relative strengths among MM parameters for a given model, the ratios being compared across TIs, it is likely that also the MRF parameters develop similar patterns. This we have observed in the MRF parameters. Fig.4.8 illustrates this. We have plotted the values of the MRF parameters against their one-dimensional index (see for instance Fig.3.2 for an explanation of this index), the top figure pane being for $L = 3$ models and the bottom pane for $L = 5$ models. The figure clearly shows that there is very little variation across the different TIs.

It is appropriate to at this point remind that even though the parameter values are quite similar across the TIs, the patterns developed are not the same: Both the full MRF and the MM models make grid patterns where the mean object lengths are comparable to those in their respective TI (long-long for TI 1, long-short for TI 2, short-short for TI 3).

To summarize, we can understand why the graph patterns in Fig.4.6 are almost independent of the TIs by looking at the MM parameters. It is at this point an open question why the different TIs give so similar parameter estimates for the MM models. To explore this problem further it is necessary to study the estimation procedure more closely. This is left for future work and will not be discussed here.

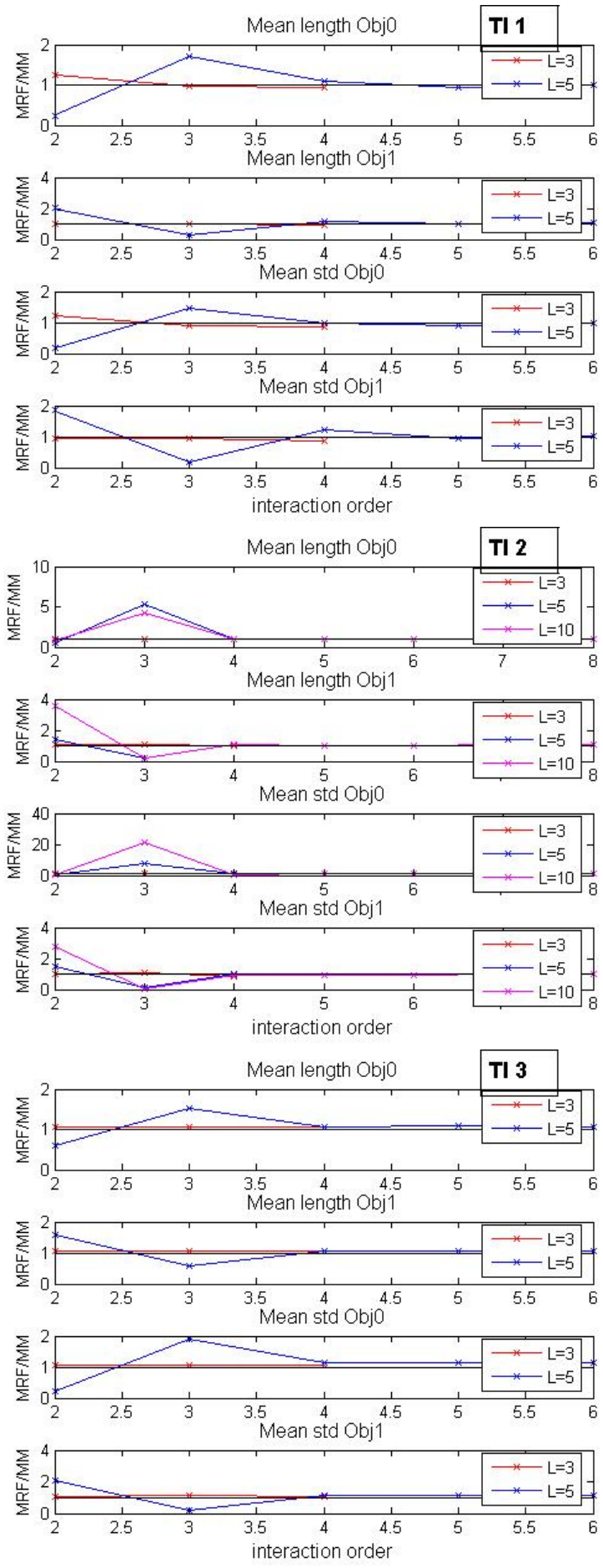


Figure 4.6. MRF mean values relative to MM mean values. TI 1, TI 2, and TI 3

L=3:

MM parameter for TI 1:
α_0 = -3.849935e+000

θ_1 = 1.596265e+001
θ_2 = 9.032075e-003
θ_3 = -8.271404e+000

MM parameter for TI 2:
α_0 = -1.947884e+000

θ_1 = 1.591641e+001
θ_2 = 1.571504e-003
θ_3 = -1.011988e+001

MM parameter for TI 3:
α_0 = -1.944313e+000

θ_1 = 1.565413e+001
θ_2 = 3.220941e-004
θ_3 = -1.176628e+001

L=5:

MM parameter for TI 1:
α_0 = -3.806440e+000

θ_1 = 1.614219e+001
θ_2 = 1.575933e+000
θ_3 = -8.904618e-003
θ_4 = -1.561244e+000
θ_5 = -8.534762e+000

MM parameter for TI 2:
α_0 = -1.612198e+000

θ_1 = 1.606101e+001
θ_2 = 1.568359e+000
θ_3 = -7.240950e-004
θ_4 = -1.567206e+000
θ_5 = -1.064255e+001

MM parameter for TI 3:
α_0 = -1.607041e+000

θ_1 = 1.553917e+001
θ_2 = 1.564881e+000
θ_3 = -2.521150e-004
θ_4 = -1.564475e+000
θ_5 = -1.232572e+001

L=10:

MM parameter for TI 2:
α_0 = 2.050102e-001

θ_1 = 1.592460e+001
θ_2 = 1.906712e+000
θ_3 = 7.529665e-001
θ_4 = 2.151924e-001
θ_5 = -2.145778e-001

θ_6 = -7.522308e-001
θ_7 = -1.914858e+000
θ_8 = -1.031275e+001
θ_9 = -9.374010e-001
θ_{10} = -1.173001e+000

Figure 4.7. Markov Mesh parameters

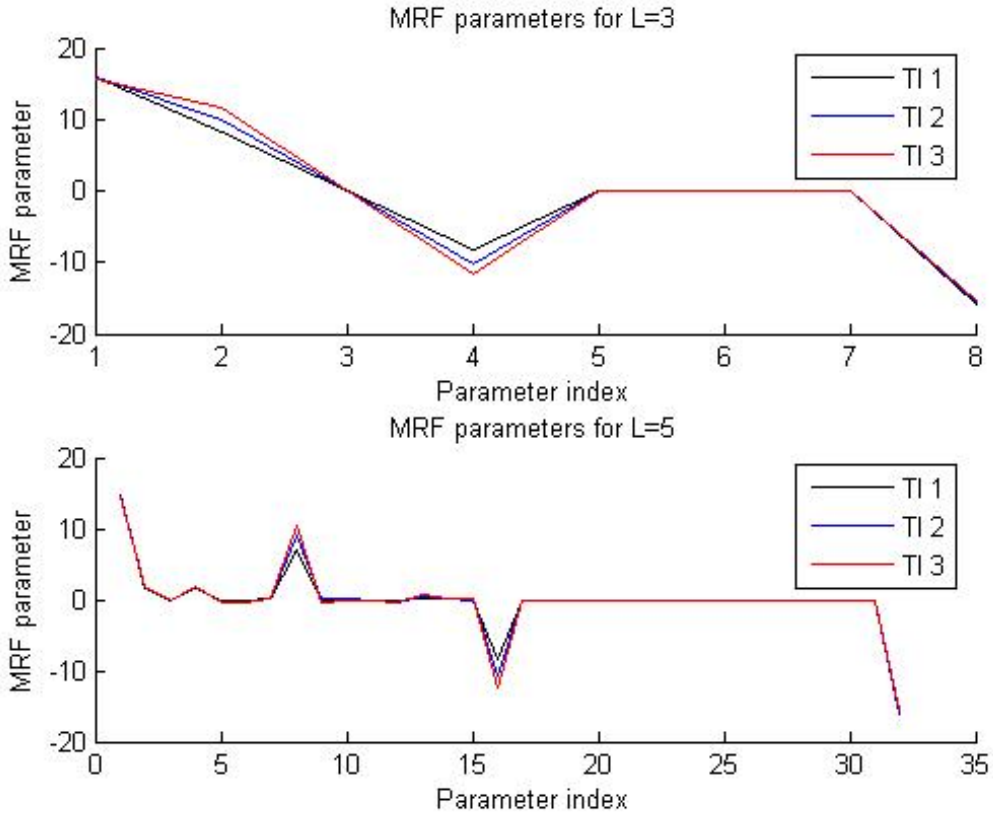


Figure 4.8. MRF parameters for $L = 3$ and $L = 5$

5 Summary and concluding remarks

We have in this report shown how a stationary Markov Mesh model consisting of external field and 2-particle interactions in one spatial dimension can be formulated as a Markov Random Field. Explicite formulas and recursive algorithms expressing the individual MRF parameters in terms of the MM parameters have been developed. It has also been shown how the combined effect of all higher order interactions in the MRF model can be computed directly from the MM parameters, without a need to first find all individual MRF parameters.

We have also described a Matlab implementation for the parameter translation and a Gibb's sampler simulation procedure for the Markov Random Field.

The Matlab implementation has been used to study the relation between a Markov Random Field and the Markov Mesh model from which it was derived. In particular we have explored to which extent higher order interactions in the MRF model are of any significance to the observed statistics. The results showed that in general the higher order interactions are indeed important; the 2nd order MRF is not sufficient to reproduce the statistics of the MM model.

We would like to point out that our study does not discuss to which extent a 2nd order MRF, with parameters estimated in the best possible way, is able to produce statistics similar to a 2nd order MM model. Our work has focused on the importance of the higher order interaction parameters in the derived MRF. An important motivation for this is that it has been observed elsewhere in the Multipoint project that successful model fitting is often easier for MM models than for MRF models. And hence the parameter translation developed in this report can be viewed as a method for doing successful MRF model identification.

In future deployment of MRF models derived from MM models, for instance with the purpose of performing simulations conditioned on well data, it is likely to be time saving to exploit (generalizations of) the expression where the combined effect of all higher order interactions in MRF are taken into account simultaneously. Since we have shown in this report that the higher order interactions in general cannot be neglected, to find each one of them invidually may not be the best method.

Future extensions of the work described here include the following topics:

- **More general interactions:** Realistic grid patterns generally require more general interactions than the 2nd order interactions in the considered MM models; even the simple patterns of the TIs used in this report are not well reproduced by this simple set of MM models. The extension of the translation

procedure to more general MM models is thus of uttermost importance. It is likely that some of the ideas shown in this report can be generalized, in particular the method of identifying parameters in terms of the neighbourhood configuration it represents. This problem will be followed up in the future.

- **Higher spatial dimensions:** For higher spatial dimensions we expect that many of same procedures and formulas can be used. Simply let the one-dimensional indices represent an ordering of the cells in the neighbourhood.

A Markov Mesh conditional probability

This appendix shows the derivation of the expression in Eq.2.1, the starting point for the Markov Mesh model in these notes' main discussion.

A.1 General relation between Markov Mesh models and Markov Random Fields

The joint probability of an N-particle system can always be written as

$$p(\mathbf{z}) = p(z_1)p(z_2|z_1)p(z_3|z_1,z_2)\cdots p(z_i|\{z_j : 1 \leq j < i\})\cdots p(z_N|z_{-N}).,\qquad (A.1)$$

where the subscripts i, j indicate cells and we have used the notation

$$z_{-i} = \{z_j : j \neq i\}. \qquad (A.2)$$

The definition of the Markov Mesh model is that the conditional probability $p(z_i|\{z_j : 1 \leq j < i\})$ depends, not on all cells with $j < i$, but only on the cells that belong to the sequential neighbourhood η_i. That is

$$p(z_i|\{z_j : 1 <= j < i\}) = p(z_i|\{z_j : j \in \eta_i\}). \qquad (A.3)$$

Let's now concentrate on one specific cell i and calculate the conditional probability for z_i, given all other values, $p(z_i|z_{-i})$. The general expression is

$$p(z_i|z_{-i}) = \frac{p(\mathbf{z})}{p(z_{-i})}. \qquad (A.4)$$

We use the general expression from A.1 for the joint probability and insert the expression that is specific for Markov Mesh models, A.3. This gives

$$\begin{aligned} p(z_i|z_{-i}) &= \frac{p(\mathbf{z})}{p(z_{-i})} \\ &= \frac{1}{p(z_{-i})}p(z_1)p(z_2|z_1)p(z_3|z_1,z_2)\cdots p(z_{i-1}|\{z_l : l \in \eta_{i-1}\}) && (A.5) \\ &\quad \cdot p(z_i|\{z_j : j \in \eta_i\}) \prod_{k:i\in\eta_k} p(z_k|\{z_p : p \in \eta_k\}) && (A.6) \\ &\quad \cdot \prod_{\hat{k}:i\notin\eta_{\hat{k}}\wedge\hat{k}\neq i} p(z_{\hat{k}}|\{z_{\hat{p}} : \hat{p} \in \eta_{\hat{k}}\}). && (A.7) \end{aligned}$$

The factors in lines A.5 and A.7 are independent of z_i, whereas all factors in line A.6 depend on the value of z_i. We denote the product of the z_i-independent factors as $\hat{C}_i$, indicating with the subscript that although there is no dependence on z_i the factor $\hat{C}_i$ is nevertheless specific for cell i. The conditional probability for a Markov Mesh model can then be written as

$$p(z_i|z_{-i}) = \hat{C}_i \cdot p(z_i|\{z_j : j \in \eta_i\}) \prod_{k:i\in\eta_k} p(z_k|\{z_p : p \in \eta_k\}). \tag{A.8}$$

It follows from this expression that the Markov Mesh model is a Markov Random Field model with a Markov neighbourhood ∂_i that satisfies

$$\partial_i \cup i = \eta_i \cup \{\eta_k : i \in \eta_k\} \cup \{k : i \in \eta_k\}. \tag{A.9}$$

The set $\{k : i \in \eta_k\}$ needs to be included since the MM neighbourhood η_k does not include cell k itself. The analog for the Markov neighbourhood is the property $i \notin \partial_i$

A.2 Sequential conditional probability for 2-particle interactions

We assume that the Markov Mesh model is characterized by 2-particle interactions and an external field. This assumption is expressed in terms of the sequential conditional probability as

$$p(z_i|\{z_j : j \in \eta_i\}) = R_i(z_{\eta_i}) e^{G_0 z_i + \sum_{j\in\eta_i} G_{i,j}(z_i,z_j)}. \tag{A.10}$$

Here $R_i(z_{\eta_i})$ is a normalization constant specific for cell i, G_0 is the constant external field, and $G_{i,j}(z_i, z_j)$ denotes the interaction between cells i and j. For binary facies fields the interaction can be written

$$\begin{aligned} G_{i,j}(z_i, z_j) &= \beta_0(1-z_i)(1-z_j) + \beta_1 z_i(1-z_j) + \beta_2(1-z_i)z_j + \beta_3 z_i z_j \\ &= \beta_0 + (-\beta_0+\beta_1)z_i + (-\beta_0+\beta_2)z_j + (\beta_0-\beta_1-\beta_2+\beta_3)z_i z_j. \end{aligned}$$

The four β-parameters denote the interaction strengths, and since they occur only in the specific combinations shown in the last expression the interaction can be expressed as

$$G_{i,j}(z_i, z_j) = \theta_0(z_i + z_j) + \theta_{i,j} z_i z_j. \tag{A.11}$$

The constant e^{β_0} has been absorbed by the normalization constant $R_i(z_{\eta_i})$, and we have assumed translational invariance so that $\beta_1 = \beta_2$. Now redefine the normalization constant according to

$$R_i(z_{\eta_i}) \cdot e^{\sum_{j\in\eta_i} \theta_0 z_j} \rightarrow R_i(z_{\eta_i}).$$

The sequential conditional probability can then be written

$$p(z_i|\{z_j : j \in \eta_i\}) = R_i(z_{\eta_i}) e^{(G_0+n\theta_0)z_i + \sum_{j\in\eta_i}\theta_{i,j}z_i z_j},$$

where $n = \sum_{j\in\eta_i}$. We also redefine the external field in the sense

$$\alpha_0 = G_0 + n\theta_0,$$

and so

$$p(z_i|\{z_j : j \in \eta_i\}) = R_i(z_{\eta_i}) e^{\alpha_0 z_i + \sum_{j\in\eta_i}\theta_{i,j}z_i z_j}. \tag{A.12}$$

Now require normalization, i.e. require that

$$p(z_i = 0|\{z_j : j \in \eta_i\}) + p(z_i = 1|\{z_j : j \in \eta_i\}) = 1.$$

This gives that the normalization constant must be

$$R_i(z_{\eta_i}) = \left(1 + e^{\alpha_0 + \sum_{j\in\eta_i}\theta_{i,j}z_j}\right)^{-1}. \tag{A.13}$$

And hence

$$p(z_i|\{z_j : j \in \eta_i\}) = \frac{e^{\alpha_0 z_i + \sum_{j\in\eta_i}\theta_{i,j}z_i z_j}}{\left(1 + e^{\alpha_0 + \sum_{j\in\eta_i}\theta_{i,j}z_j}\right)}. \tag{A.14}$$

A.3 Full conditional probability for 2-particle interactions

From Eq.A.8 we have that the full conditional probability for a Markov Mesh model can be written

$$p(z_i|z_{-i}) = \hat{C}_i \cdot p(z_i|\{z_j : j \in \eta_i\}) \prod_{k:i\in\eta_k} p(z_k|\{z_p : p \in \eta_k\}). \tag{A.15}$$

Insert into this equation the sequential conditional probabilities from Eq.A.12. This gives

$$\begin{aligned} p(z_i|z_{-i}) &= \hat{C}_i \cdot R_i(z_{\eta_i}) \cdot e^{\alpha_0 z_i + \sum_{j\in\eta_i}\theta_{i,j}z_i z_j} \cdot \prod_{k:i\in\eta_k} R_k(z_{\eta_k}) \cdot e^{\alpha_0 z_k + \sum_{l\in\eta_k}\theta_{l,k}z_k z_l} \\ &= \hat{C}_i \cdot R_i(z_{\eta_i}) \cdot \left\{\prod_{k:i\in\eta_k} \exp(\alpha_0 z_k + \sum_{l:l\neq i\wedge l\in\eta_k} \theta_{k,l}z_k z_l)\right\} \cdot \qquad \text{(A.16)} \\ &\quad \left\{\prod_{k:i\in\eta_k} R_k(z_{\eta_k})\right\} \cdot \exp\left(\alpha_0 z_i + \sum_{j\in\eta_i}\theta_{i,j}z_i z_j + \sum_{k':i\in\eta_{k'}}\theta_{k',i}z_{k'}z_i\right) \qquad \text{(A.17)} \end{aligned}$$

The factors in line A.16 are independent of z_i, but depend on the facies of the MRF neighbourhood ∂_i. We denote the product as

$$C_i(z_{\partial_i}) = \hat{C}_i \cdot R_i(z_{\eta_i}) \cdot \left\{\prod_{k:i\in\eta_k} \exp(\alpha_0 z_k + \sum_{l:l\neq i\wedge l\in\eta_k} \theta_{k,l}z_k z_l)\right\}. \tag{A.18}$$

The product $\prod_{k:i\in\eta_k} R_k(z_{\eta_k})$ depends on z_i through the expression in Eq.A.13. The expression for the full conditional probability is therefore

$$p(z_i|z_{-i}) = C_i(z_{\partial_i}) \cdot \frac{\exp\left(\alpha_0 z_i + \sum_{j\in\eta_i} \theta_{i,j} z_i z_j + \sum_{k':i\in\eta_{k'}} \theta_{k',i} z_{k'} z_i\right)}{\prod_{k:i\in\eta_k}\left(1 + e^{\alpha_0 + \sum_{j\in\eta_k} \theta_{k,j} z_j}\right)}. \tag{A.19}$$

Notice that this expression is valid for any spatial dimension, any choice of sequential neighbourhood, and any choice of how the cells are labelled. It can in general be further refined by taking into account translational symmetry, since this gives relations between the interaction parameters $\theta_{i,j}$ and $\theta_{k',i}$.

For a 1-dimensional system where the cells are labeled according to their physical ordering along a line, translational symmetry gives the relation $\theta_{i,j} = \theta_{j,i}$. This implies, with the notation

$$\theta_l = \theta_{i,i\pm l}$$

and a sequential neighbourhood

$$\eta_i = \{i - L, i - L + 1, ..., i - 1\},$$

that the final expression for the conditional probability is

$$p(z_i|z_{-i}) = C_i(z_{\partial_i}) \cdot \frac{\exp(\alpha_0 z_i + \sum_{l=1}^{L} \theta_l z_i (z_{i-l} + z_{i+l}))}{\prod_{k=i+1}^{i+L} (1 + \exp(\alpha_0 + \sum_{l=1}^{L} \theta_l z_{k-l}))}. \tag{A.20}$$

This is identical to Eq.2.1.

B Normalization constant in Markov Mesh model

The conditional probability of the Markov Mesh model can be written

$$p(z_i|z_{-i}) = C_i(z_{\partial_i})\frac{\exp(\alpha_0 z_i + \sum_{l=1}^{L}\theta_l z_i(z_{i-l}+z_{i+l}))}{\prod_{k=i+1}^{i+L}(1+e^{\alpha_0+\sum_{l=1}^{L}\theta_l z_{k-l}})}, \tag{B.1}$$

$$= C_i(z_{\partial_i})\frac{\exp(\alpha_0 z_i + \sum_{l=1}^{L}\theta_l z_i(z_{i-l}+z_{i+l}))}{\prod_{k=i+1}^{i+L}(1+\phi_k e^{\theta_{k-i}z_i})}, \tag{B.2}$$

where we in the last expression have used the shorthand notation

$$\begin{aligned}\phi_k &= \phi_k(z_{k-L},...,z_{i-1},z_{i+1},...,z_{k-1})\\ &= \exp(\alpha_0 + \sum_{l=1,l\neq k-i}^{L}\theta_l z_{k-l}),\\ &\quad \text{for } k\in\{i+1,i+2,...,i+L\}.\end{aligned}$$

The function ϕ_k depends on the facies of each cell in the sequential neighbourhood η_k, except for the facies in cell i.

The normalization constant is determined by the requirement $p(z_i=1|z_{-i}) + p(z_i=0|z_{-i}) = 1$. I.e. we require

$$C_i(z_{\partial_i})\left(\frac{e^{\alpha_0+\sum_{l=1}^{L}\theta_l(z_{i-l}+z_{i+l})}}{\prod_{k=i+1}^{i+L}(1+\phi_k e^{\theta_{k-i}})} + \frac{1}{\prod_{k=i+1}^{i+L}(1+\phi_k)}\right) = 1.$$

Using the notation $\psi_k = \phi_k e^{\theta_{k-i}}$, the requirement can be written

$$C_i(z_{\partial_i})\prod_{k=i+1}^{i+L}(1+\psi_k)^{-1}\cdot\left(e^{\alpha_0+\sum_{l=1}^{L}\theta_l(z_{i-l}+z_{i+l})} + \prod_{k=i+1}^{i+L}\frac{1+\psi_k}{1+\phi_k}\right) = 1$$

and we have that the normalization constant can be expressed as

$$C_i(z_{\partial_i}) = \prod_{k=i+1}^{i+L}(1+\psi_k)\cdot\left(e^{\alpha_0+\sum_{l=1}^{L}\theta_l(z_{i-l}+z_{i+l})} + \prod_{k=i+1}^{i+L}\frac{1+\psi_k}{1+\phi_k}\right)^{-1}. \tag{B.3}$$

This expression is valid for any grid configuration.

The conditional probabilities can then be written, for $z_i = 0$ and $z_i = 1$ respectively,

$$
\begin{aligned}
p(z_i = 0|z_{-i}) &= \prod_{k=i+1}^{i+L} \left(\frac{1+\psi_k}{1+\phi_k}\right) \cdot \left(e^{\alpha_0 + \sum_{l=1}^{L} \theta_l (z_{i-l}+z_{i+l})} + \prod_{k=i+1}^{i+L} \frac{1+\psi_k}{1+\phi_k}\right)^{-1}, \\
p(z_i = 1|z_{-i}) &= e^{\alpha_0 + \sum_{l=1}^{L} \theta_l (z_{i-l}+z_{i+l})} \cdot \left(e^{\alpha_0 + \sum_{l=1}^{L} \theta_l (z_{i-l}+z_{i+l})} + \prod_{k=i+1}^{i+L} \frac{1+\psi_k}{1+\phi_k}\right)^{-1}.
\end{aligned}
$$

C Documentation of scripts/functions

This appendix is a documentation of the scripts and functions used in the simulations. The script **compareMM_MRF** runs all simulations given a training image and saves simulation data for each case in a unique folder with name according to the case. The following shows which functions are used in a full simulation. The functions in bold, except **makeTI**, are called from **compareMM_MRF**. A function in a lower level is called by the previous function in a higher level, ex: **paramTrans** calls **permutation**.

C.1 Stepwise /Dependence

Step 0: make TI.

- **makeTI**

Run **compareMM_MRF**:

Step 1: Estimate MM-parameters from TI.

- **estSequential1D_2order** (estimate coeff) (Odd)
  - sequential1D_2order (Odd)
  - logisticRegression (Odd)

Step 2: Translate MM-parameters to MRF-parameters.

- **paramTrans**
  - sumSubInteract
  - permutation

Step 3: find the potential (MRF) for each possible configuration of the neighbourhood.

- **findSumInteractOfConf**
  - interactOfConf (subfunction)
    - permutation

Step 4: run MRF-gibbs sampler using the potential found in step 3.

- **gibbsMRF**

– statisticsObj

Step 5: simulate MM-model using parameters found in step 1.

· **simMM**

– simSequential1D_2order (Odd)

– statisticsObj

Final: compare MRF- and MM-simulations.
Visual inspection of figures and comparison of calculated means (calculated in statisticsObj).

C.2 Main parameters

Syntax: **name (type, size):**

coeff (int array, $L+1 \times 1$):
$=\{\theta_L, \theta_{L-1}, \ldots, \theta_1, \alpha_o\}$. Array of estimated MM-coefficients, indexed as shown.

conf (int array, M):
Indexes of nonzero cells in future nbh. Cell number in grid $\rightarrow$ index in table: $i+1 \rightarrow 1, i+2 \rightarrow 2, \ldots, i+L \rightarrow L$.

grid (int array, $N \times 1$):
simulated MM- or MRF-model.

highestOrder (int):
highest order interaction to be considered.

K (int):
number of particles with facies=1 in markov neighbourhood, δ_i.

L (int):
size of sequential neighbourhood.

logProb (double array, $(T/sample) \times 1$): log(P(grid))). Gives the log-probability of the configuration of the whole grid for each sampled grid.

lowestOrder : lowest order interaction to be considered.

M (int):
number of particles with facies=1 in future neighbourhood.

N (int):
size of simulated grid

nx (int):
size of grid to simulate on, nx=N+200, padding in each end.

nonZeroNbh (int array, K):
Current configuration of markov nbh, δ_i. Array of index of nonzero cells, not including cell i. Index: $i - L = 1, \ldots, i + 1 = L + 1, \ldots, i + L = 2 \times L$

param (double array, $2^L \times 1$):
translated MRF-interaction parameters. Index of array is binary configuration of the interaction seen as particles in the future nbh.

potOfPast (double array, $2^2 L \times 1$):
potential of seqential nbh. index is binary value of seq.-nbh.

sample (int):
how often to sample in Gibb's-simulation of MRF. Assumed "distance" between independent grids.

sample (int array, $(T/sample) \times N$:
array of all sampled grids in one MRF-simulation.

statObjSizeArray (double array, $6\times$ number of sampled grids):
array of statistics on the size of the objects in the simulated grids.
statObjSizeArray(1,:): mean of size of 0-objects in each simulated grid.
statObjSizeArray(2,:): mean of size of 1-objects –"–.
statObjSizeArray(3,:): standard deviation of size of 0-objects –"–.
statObjSizeArray(4,:): standard deviation of size of 1-objects –"–.
statObjSizeArray(5,:): number of 0-objects –"–.
statObjSizeArray(6,:): number of 1-objects –"–.

statMM_array (double array, $6 \times (T/M)$):
statObjSizeArray for the current case of MM simulations

statMRF_array (double array, $6 \times (T/sample)$):
same as above but for MRF simulations

statTI (double array, 6×1):
statistics on current training image.

sumInteractUp2HO (double array, $2^L \times 1$):
log-potential (sum of particle interactions up to highestOrder) of all possible configurations of neighbourhood. Index is binary value of configuration. Given a nbh, index is: $\sum 2^{(nonZeroNbh-1)}$. See fig 3.3.

T (int): number of simulations (Gibb's)

TI (int array,1×100000): training image

C.3 Description of each function

The parameters described in the last section are not repeated here.

Syntax: **function [output]=nameOfFunction(input)**.

compareMM_MRF (script)

the main script that runs all simulations. The functions called is described in C.1. All results are saved in a directory called <prefixDirName>_TI_<intTI>_T_<T>_N_<N>, and subdirectories. For each L we run simulations with all numbers in *highestOrder* as the highest order particle interaction. Results for each case is saved in its own dir.
PARAMETERS: *intTI*: number on current training image, see table 4.1.
SAVE: In main dir, for each L: L-info.mat={param, coeff, L, T ,N} and MM-variabels.mat={*statMM_array*}.
In dir *h.o*<..>, for each *higherOrder*: MRF-variabels.mat={*sumInteractUp2HO*, *statMRF_array*,*logProb*,*sampleGridArray*}, and plots of burn in, histogram and *logProb*.
RETURN: -

function [sumInteract]= findSumInteractOfConf(L, param,highestOrder)

find potential for all possible configurations of neighbourhoods.
PARAMETERS: *highestOrder*: the highestOrder particle interaction to be considered (included)
RETURN: array of potential (particle interactions) for all possible configuration of the nbh-grid.

function [statObjSize]= gibbsMRF(param, sumInteract,randGrid,T,N,L,sample)

simulate MRF and return statistics of simulations. Updates the grid-cells in random order.
PARAMETERS: *startGrid*: initial state of grid to run Gibb's sampler on
nx: size of grid to simulate on, (we cut of 100 in each end t o do stat)
RETURN: statistics of $T/sample$ grids.

function [sumInteractParam]=interactOfConf(nonZeroNbh,L,param, highestOrder)

Internal function in the script higherOrderCombinations. Sums the higher order interactions $\leq$ highestOrder for a given configuration of the Markov neighbourhood (nonZeroNbh). Starts from left in the array nonZeroNbh and adds all forward particle interactions up to highestOrder, then moves one step right to next nonzero cell and repeats.
RETURN: sum of interactions parameters for one markov neighbourhood with the cells in nonZeroNbh nonzero.

function [TI]=makeTI(N,mean0, mean1, spread0, spread1)
Make a training image (TI) with mean of size of 0-objects is mean0, and mean of size of 1-objects is mean1. Size of 0-objects vary between
$[mean0 - spread, mean0 + spread]$ and same with 1-objects.
RETURN: *TI*.

function [param]=paramTrans(L,coeff,pathDir):
Recursive algorithm for translating MM-coefficients to MRF-parameters. Makes a text-file, *paramTransOut*, which lists the parameters. The file is saved to *pathDir*.
PARAMETERS: *pathDir*: current directory to save results in
RETURN: translated MRF-parameters.

function [pos, findParam]=permutation(pos,M,L)
find next configuation of the nbh.
PARAMETERS: pos: array, current position of the M nonzero particles
(configuration of future nbh or whole nbh)
findParam: boolean, if false, last configuration is reached.
RETURN: new position of nonzero particles, cont or not.

function [statObjSize] = simMM (coeff, J, N, L)
simulate J times the MM-grid of size N .
RETURN: statistics of all simulations.

function [statObjSize]=statisticsObj(grid)
find statistics on sizes of 0-objects and 1-objects.
PARAMETERS: grid: array of simulated grid (MM or MRF)

function [subInteract]= sumSubInteract(conf,param)
find sum of subinteractions for a given conf.
PARAMETERS:-
RETURN:- sum of subinteractions.

C.4 Organizing result files and folders

We have been looking at 3 different training images, see Table 4.1. The training images are saved in a directory called *data*. All results from one training image is saved in a directory called <prefixDirName>_TI_<intTI>_T_<T>_N_<N>. A subdirectory with name L_<L> is made for each L. This directory contains the following:

- L-info

- MM-simulation plots
- MM-variabels.mat
- one directory for each h.o

For each highestOrder, a directory called h.o_<highestOrder> is made. It contains the following:

- MRF-simulation plots
- MRF-variabels.mat=(sumInteractUp2HO, statMRF_array,logProb,sampleGridArray)