

Direct Sequential Simulation for spherical linear inverse problems*

Mikkel Otzen*, Chris Finlay

DTU Space, Centrifugevej, bygning 356, 2800 Kgs. Lyngby, Denmark

ARTICLE INFO

Keywords:

sequential simulation
linear inverse problems
spherical geometry
kriging

Abstract

Here is the abstract.

It will be paragraphs one day.

There will be some lines.

One more to fit the line.

Maybe a couple more to represent the size better.

There we go.

Yes.

1. Introduction

Introduce the article. Lets get the references going with Tarantola (2005) and Hansen and Mosegaard (2008).

2. Theory

2.1. The linear forward problem in spherical geometry

In spherical geometry, given observations, $d(\mathbf{r})$, and model parameters on a spherical surface, $m(\mathbf{s})$, related through an integrating kernel $\mathcal{G}(\mathbf{r}, \mathbf{s})$ we consider a linear inverse problem of the form shown in equation (1).

$$d(\mathbf{r}) = \int_S \mathcal{G}(\mathbf{r}, \mathbf{s})m(\mathbf{s}) dS \quad (1)$$

Where $dS = \sin \theta_s d\theta_s d\phi_s$, with $\mathbf{r} = (r, \theta, \phi)$ indicating locations of the observations, and $\mathbf{s} = (R, \theta_s, \phi_s)$ indicating locations of the model parameters on a spherical surface of radius R . This integral equation may be approximated numerically via quadrature rules. For illustration we use a Gauss-Legendre quadrature scheme (e.g. Atkinson, 1982), in which the integration is carried out on a $2N_q \times N_q$ grid. $\cos \theta_s$ are then the Gauss-Legendre nodes on the interval $[-1, 1]$, with corresponding integration weights, w_s . ϕ_s is chosen such that the points are equally spaced with separation π/N_q on the interval $[0, 2\pi]$. For model parameters on a sphere distributed according to Gauss-Legendre quadrature rules, this numerical integration is exact for polynomials of degrees less than $2N_q$ (Atkinson, 1982). The integral may then be discretized according to equation (2).

$$d(\mathbf{r}) = \frac{\pi}{N_q} \sum_{i=1}^{N_m} w_i \mathcal{G}(\mathbf{r}, \mathbf{s}_i) m(\mathbf{s}_i) \quad (2)$$

Where $N_m = 2N_q \times N_q$ is the number of model parameters. For a series of N_d observations, $\mathbf{d} = [d_1(\mathbf{r}_1), \dots, d_i(\mathbf{r}_i), \dots, d_{N_d}(\mathbf{r}_{N_d})]^T$, with a vector of model parameters, $\mathbf{m} = [m_1(\mathbf{s}_1), \dots, m_i(\mathbf{s}_i), \dots, m_{N_m}(\mathbf{s}_{N_m})]^T$, and

* Github repository containing implementation is available at github.com/mikkelotzen/spherical_direct_sequential_simulation

*Corresponding author

 miketz@space.dtu.dk (M. Otzen)

ORCID(s): 0000-0001-6227-6916 (M. Otzen)

where the constant, $\frac{\pi}{N_q}$, and the integration weights, w_i , have been absorbed into the elements of the matrix \mathbf{G} (size $N_d \times N_m$), such that

$$G_{ij} = \frac{\pi}{N_q} w_i \mathcal{G}(r_j, s_i) \quad (3)$$

The linear forward problem in spherical geometry dealt with here may be written in the familiar form

$$\mathbf{d} = \mathbf{G}\mathbf{m} \quad (4)$$

The inverse problem is then to estimate \mathbf{m} , which is a vector of parameter values on a spherical surface grid, given observed data and any prior information on the model parameters.

2.2. Equivalent least-squares solution to the linear inverse problem

A simple solution to the above exists if we are able to assume the spherical surface model parameters can be represented by a Gaussian probability density function (pdf) with a priori mean μ_0 and covariance \mathbf{C}_m , while the observations, \mathbf{d} , represent the mean of a Gaussian pdf with data error covariance \mathbf{C}_e (Hansen, Journel, Tarantola and Mosegaard, 2006). The least-squares solution is then also a Gaussian pdf with mean

$$\hat{\mathbf{m}}_{LSQ} = \mu_0 + \mathbf{C}_m \mathbf{G}^T \mathbf{S}^{-1} (\mathbf{d} - \mathbf{G}\mu_0) \quad (5)$$

and covariance

$$\hat{\mathbf{C}}_{LSQ} = \mathbf{C}_m - \mathbf{C}_m \mathbf{G}^T \mathbf{S}^{-1} \mathbf{G} \mathbf{C}_m \quad (6)$$

where

$$\mathbf{S} = \mathbf{C}_e + \mathbf{G} \mathbf{C}_m \mathbf{G}^T \quad (7)$$

The Gaussian assumption is often restrictive, here we present in section 2.4 a more flexible scheme that enables the reproduction of more general histograms and semi-variograms describing the model on the spherical surface, as obtained from a priori training images. As a preliminary to this, we first describe the method of sequential Gaussian simulation on the sphere.

2.3. Sequential Gaussian Simulation on the sphere

The observations \mathbf{d} can be considered to result from a set of true random variables, \mathbf{m} . From the observations it is possible to infer Gaussian pdf estimates of the random variables through the method of sequential simulation, here represented by the estimated realizations of the model parameters $\hat{\mathbf{m}}$ on the spherical surface. This leads to Gaussian posterior realizations of the random variable field (Deutsch and Journel, 1998; Hansen and Mosegaard, 2008). Consider inverse transform sampling our chosen method of drawing from a random variable distribution. If a cumulative distribution function (cdf), $F_{\mathbf{m}}$, describing the random variable distribution, \mathbf{m} , exists, variable estimates belonging to the distribution may be drawn through the inverse cdf (the so-called quantile function) by sampling randomly from a standard uniform distribution in the range zero to one, $U(0, 1)$, such that

$$\hat{m}_i = F_{\mathbf{m}}^{-1}(U(0, 1)_i) \quad (8)$$

In our context we consider the joint distribution of N_m random variables, m_i , conditioned on a set of known observations, \mathbf{d} . This conditioning can be expressed through the N_m variate cdf (9) where P denotes probability.

$$\begin{aligned} F_m(m_1, \dots, m_{N_m} | \mathbf{d}) &= P\{m_i \geq \hat{m}_i, i = 1, \dots, N_m | \mathbf{d}\} \\ &= P\{m_1 \geq \hat{m}_1 | \mathbf{d}\} P\{m_2 \geq \hat{m}_2 | \mathbf{d}, \hat{m}_1\} \dots P\{m_N \geq \hat{m}_N | \mathbf{d}, \hat{m}_1, \hat{m}_2, \dots, \hat{m}_{N-1}\} \end{aligned} \quad (9)$$

Sequential simulation is carried out by drawing a valid N_m variate sample from (9) by using the product rule of probability, such that each probability term on the right-hand side is sampled in succession (Deutsch and Journel, 1998). Estimates of the set of random variables are then found in N_m steps continuously increasing the conditioning as each step of the sequential simulation is taken, beginning with the observations, \mathbf{d} , available at the start of the simulation. This requires N_m univariate cdf's

$$\begin{aligned} F_{m_1} &= P\{m_1 \geq \hat{m}_1 | \mathbf{d}\} \\ F_{m_k} &= P\{m_k \geq \hat{m}_k | \mathbf{d}, \hat{m}_1, \dots, \hat{m}_{k-1}\} \\ F_{m_N} &= P\{m_{N_m} \geq \hat{m}_{N_m} | \mathbf{d}, \hat{m}_1, \dots, \hat{m}_k, \dots, \hat{m}_{N_m-1}\} \end{aligned} \quad (10)$$

on which inverse transform sampling can be applied to draw estimates of the random variables. One a posteriori realization of the Gaussian random field consists of randomly visiting and estimating each model parameter. For realizations of a Gaussian random field, drawing samples satisfying the sequential probability relations in (9) and (10) equates to drawing from the Gaussian pdf, $\mathcal{N}(\mu_k, \sigma_k^2)$, at each model parameter location, where μ_k and σ_k^2 define the kriging mean and variance. These two parameters are found by solving the kriging system (e.g. Journel and Huijbregts, 1978; Deutsch and Journel, 1998). Following Hansen and Mosegaard (2008), but adopting our notation this can be written

$$\sum_{i=1}^{N_v} C_v(v_i, v_j) \lambda_i = c_{vm}(v_i, \hat{m}_k) \quad \forall j = 1, \dots, N_v \quad (11)$$

Where v_i is a member of the N_v available conditional variables in each step (observations and previously estimated model parameters), \hat{m}_k is the model parameter currently being estimated, and λ_i are known as the kriging weights which determine the desired Gaussian pdf, $\mathcal{N}(\mu_k, \sigma_k^2)$. $C_v(v_i, v_j)$ are the a priori covariances between conditional variables including any measurement error covariance, and $c_{vm}(v_i, m_t)$ are the covariances between conditional variables and the target model parameter. This may be written compactly in matrix notation as

$$\mathbf{C}_v \boldsymbol{\lambda} = \mathbf{c}_{vm} \quad (12)$$

After solving equation (12) for the kriging weights $\boldsymbol{\lambda}$, the kriging mean and variance describing the desired Gaussian pdf are

$$\mu_k = \boldsymbol{\lambda} \cdot (\mathbf{v} - \mu_0 \bar{\mathbf{e}}) + \mu_0 \quad \text{where } \bar{\mathbf{e}} = [1, \dots, 1]^T \text{ of length } N_v \quad (13)$$

$$\sigma_k^2 = \sigma_0^2 - \boldsymbol{\lambda} \cdot \mathbf{c}_{vm} \quad (14)$$

where μ_0 and σ_0^2 are a priori estimates of the mean and variance of the model parameters. In order to estimate the necessary a priori mean, variance, and covariance, second order stationarity is assumed and it is then possible to define a semi-variogram (e.g. Deutsch and Journel, 1998).

$$\gamma_{ij} = \frac{1}{2} E \left\{ [m_0(s_i) - m_0(s_j)]^2 \right\} \quad (15)$$

$$C\{m_0(s_i), m_0(s_j)\} = \sigma_0^2 - \gamma_{ij}, \quad \forall s \quad (16)$$

Here γ_{ij} is the semi-variance for some location on the spherical surface, s_i , with respect to another location, s_j , and m_0 is the training image (an a priori realization of m on the model parameter surface), from which estimates

of the a priori mean, variance, and covariances are found. $E\{\}$ and $C\{\}$ are expectation and covariance measures respectively. Given an estimate of the semi-variogram, covariance estimates for each model parameter location pair can be computed directly from (16). Additionally, covariance estimates between observation pairs and observation/model parameter pairs are needed. Those may be obtained by considering the forward relation between the observation and model parameters defined in (2) and (4) in conjunction with (16) and the training image. Taking the covariance of an observation pair as defined by (2), and as before absorbing the constant and integration weights into G , we find

$$\begin{aligned} C\{d(\mathbf{r}_p), d(\mathbf{r}_q)\} &= C \left\{ \sum_{i=1}^{N_m} G(\mathbf{r}_p, \mathbf{s}_i) m_0(\mathbf{s}_i), \sum_{j=1}^{N_m} G(\mathbf{r}_q, \mathbf{s}_j) m_0(\mathbf{s}_j) \right\} \\ &= \sum_{i=1}^{N_m} \sum_{j=1}^{N_m} G(\mathbf{r}_p, \mathbf{s}_i) G(\mathbf{r}_q, \mathbf{s}_j) C\{m_0(\mathbf{s}_i), m_0(\mathbf{s}_j)\} \end{aligned} \quad (17)$$

While the same method applied to observation/model parameter pairs lead to

$$C\{d(\mathbf{r}_p), m_0(\mathbf{s}_q)\} = \sum_{i=1}^{N_m} G(\mathbf{r}_p, \mathbf{s}_i) C\{m_0(\mathbf{s}_i), m_0(\mathbf{s}_q)\} \quad (18)$$

For any observation pair, $(d(\mathbf{r}_p), d(\mathbf{r}_q))$, the covariance is given by (17) and for any observation/model parameter pair, $(d(\mathbf{r}_p), m_0(\mathbf{s}_q))$, by (18). All required covariances are available given the semi-variogram through 16, which should be obtained from prior information regarding the random variable on the spherical surface. Expanding the kriging system from (12) to explicitly show the various parts of the covariance matrix

$$\begin{bmatrix} \mathbf{C}_d + \mathbf{C}_e & \mathbf{C}_{dm} \\ \mathbf{C}_{dm}^T & \mathbf{C}_m \end{bmatrix} \boldsymbol{\lambda} = \begin{bmatrix} \mathbf{c}_{dm} \\ \mathbf{c}_{mm} \end{bmatrix} \quad (19)$$

The covariance terms are as follows, \mathbf{C}_d is a matrix containing observation to observation covariances based on the prior information regarding the field on the spherical surface and is computed using (17), \mathbf{C}_e holds observation to observation data error covariances, \mathbf{C}_{dm} is covariance between observations and any previously simulated values on the spherical surface as estimated through (18), \mathbf{C}_m contains covariance between pairs of previously simulated values and is found directly through the semi-variogram, finally \mathbf{c}_{dm} and \mathbf{c}_{mm} are vectors containing the covariance between observations and previously simulated values to the target model parameter, found respectively through (18) and (16). Solving the kriging system sequentially using the covariances as defined above, the sequential Gaussian simulation parameter estimates are each defined by their resulting Gaussian pdf $\mathcal{N}(\mu_k, \sigma_k^2)$. The model parameter estimates of one posterior realization are then

$$\hat{\mathbf{m}}_{SGS} = [\mu_1, \dots, \mu_k, \dots, \mu_{N_m}]^T \quad (20)$$

Several realizations of the posterior can be generated by visiting the model parameters in a random order for each realization. Collecting N_r realization model parameter estimates as columns in an $N_m \times N_r$ matrix, $\hat{\mathbf{M}}_{SGS}$, a sample covariance is computed by

$$\hat{\mathbf{C}}_{SGS} = \frac{1}{N_r - 1} (\hat{\mathbf{M}}_{SGS} - \boldsymbol{\mu}_{SGS} \bar{\mathbf{e}})(\hat{\mathbf{M}}_{SGS} - \boldsymbol{\mu}_{SGS} \bar{\mathbf{e}})^T \quad \text{where } \bar{\mathbf{e}} = [1, \dots, 1] \text{ of length } N_r \quad (21)$$

Where $\boldsymbol{\mu}_{SGS}$ is an N_m length column vector of the model parameter means. This sample covariance is a contrast to the equivalent least squares solution in section 2.2 in which the covariance is directly estimated, as the probabilistic information is here derived from generating a group of realizations. Note that in sequential Gaussian simulation, the model parameters and the data errors are required to follow a normal distribution. In the following section we extend the method above to allow for non-Gaussian data and model parameter distributions without any transformation of the data, relying on information from normal-score transformations of an a priori training image.

2.4. Direct Sequential Simulation for non-Gaussian fields on a sphere

The system described in section 2.3 allows one to sequentially estimate model parameters on a spherical surface given observations, leading to a realization of a Gaussian random field which fit the observations to within measurement error, and as far as this fit allow, reproduce the semi-variogram, mean, and variance of the a priori information. We now go further and simulate non-Gaussian random fields using direct sequential simulation with histogram reproduction for a given training image, following the method proposed by Tran, Deutsch and Xie (2001) and Oz, Deutsch, Tran and Xie (2003). A normal-score transform of a training image to variables, \mathbf{y} , that follow a standard Gaussian distribution (i.e. zero mean, variance one), and the associated back-transformation to original values may be performed through (22) and (23).

$$\mathbf{y} = H^{-1}(F(\mathbf{m}_0)) \quad (22)$$

$$\mathbf{m}_0 = F^{-1}(H(\mathbf{y})) \quad (23)$$

Where H^{-1} is a standard Gaussian quantile function with cdf, H , F is the target histogram cdf with quantile function F^{-1} , and \mathbf{m}_0 are the training image values. This transformation describes the connection between random variables with a Gaussian distribution, and non-Gaussian distributions defined by the training image. It allows one to generate a collection of non-Gaussian cdf's through which the sampling in the sequential simulation steps described in equation (10) can occur. Generating a non-Gaussian cdf is achieved by substituting the standard Gaussian representation of the training image, \mathbf{y} , for a Gaussian distribution, \mathbf{y}_n , with mean, μ_n , and variance σ_n^2 . This can be achieved through inverse transform sampling using a vector, \mathbf{u} , of N_u uniformly spaced quantiles between zero and one, which divides the Gaussian distribution into intervals of equal probability, as follows

$$\mathbf{y}_n = H^{-1}(\mathbf{u})\sigma_n + \mu_n \quad (24)$$

The ranges of the mean and variance should cover approximately $[-3.5, 3.5]$ and $[0, 2]$ respectively, to fully utilize the target histogram in characterizing conditional distributions (Oz et al., 2003). If the distribution is generated from these uniformly spaced quantiles, performing transformation through (23) results in a discrete vectorized quantile function, \mathbf{q} , conditional on the training image and with length N_u

$$\mathbf{q} = F^{-1}(H(\mathbf{y}_n)) \quad (25)$$

from which a sample, z_i , can be drawn with inverse transform sampling using a uniform distribution, U , discretized in N_u intervals

$$z_i = \mathbf{q}(U(0, N_u)_i) \quad (26)$$

Solving the kriging system yields an estimated mean and variance of the local Gaussian distributions and a local distribution is then assigned using (25) choosing the mean, μ_i , and variance, σ_i^2 , closest to the kriging mean and variance. In this step a distance measure must be used, our implementation is described in section 3.2.1. Note the difference between the mean and variance of the Gaussian distribution used to generate the local distribution, μ_n and σ_n^2 , and the mean and variance of the generated local distribution, μ_i and σ_i^2 .

Knowledge of the target histogram thus enables sampling from a priori local distributions with shapes conditional on the target histogram. We refer to these collectively as the local conditional distributions. However, reproduction of the target histogram is only ensured if the applied local conditional distribution has mean and variance equal to the kriging mean and variance (Journel, 1994). This further requires that the local conditional distributions are scaled to be exactly the kriging mean and variance. For a value sampled from one of the local conditional distributions, this is achieved by

$$\hat{m}_k = (z_i - \mu_i) \cdot \frac{\sigma_k}{\sigma_i} + \mu_k \quad (27)$$

$$\hat{\mathbf{m}}_{DSS} = [\hat{m}_1, \dots, \hat{m}_k, \dots, \hat{m}_{N_m}]^T \quad (28)$$

\hat{m}_k is the final simulated model parameter value that makes up the vector of estimated model parameters $\hat{\mathbf{m}}_{DSS}$ in a given realization. z_i is a sample from the local conditional distribution with mean, μ_i , and variance, σ_i^2 , that is nearest to the kriging mean, μ_k , and variance, σ_k^2 . μ_k and σ_k are as before found by solving the system equations (12)-(14). As for sequential Gaussian simulation in 21 a probabilistic solution is achieved by collecting N_r model parameter realizations in the matrix, $\hat{\mathbf{M}}_{DSS}$, and computing the sample covariance as follows

$$\hat{\mathbf{C}}_{DSS} = \frac{1}{N_r - 1} (\hat{\mathbf{M}}_{DSS} - \boldsymbol{\mu}_{DSS} \bar{\mathbf{e}}) (\hat{\mathbf{M}}_{DSS} - \boldsymbol{\mu}_{DSS} \bar{\mathbf{e}})^T \quad \text{where } \bar{\mathbf{e}} = [1, \dots, 1] \text{ of length } N_r \quad (29)$$

Where $\boldsymbol{\mu}_{DSS}$ is an N_m length column vector of the model parameter means. This procedure is known as direct sequential simulation (Tran et al., 2001; Oz et al., 2003). It ensures that simulations represent samples from the posterior probability density function of the model parameters based on the mean, variance, covariance structure, and histogram provided by the training image, while honoring the data.

3. Implementation

We have implemented the methods described in section 2 as a Python repository called Spherical Direct Sequential Simulation, which is freely available on Github at github.com/mikkelotzen/spherical_direct_sequential_simulation. We consider the implementation to be six distinct groups. The geometry defining the observation and model parameter spaces, the required prior information, the measured random variable observations, the system equations, the simulation itself, and the output results from said simulation. Figure 1 gives an overview of these implementation groups and their content. How each part propagates in the implementation is shown with arrows. Table 1 gives an overview of the static implementation variables leading into the Spherical Direct Sequential Simulation algorithm itself. In the following we describe each implementation group in detail, starting with observations and geometry.

3.1. Observations and Geometry

Equation (1) defines the spherical linear inverse problem by connecting random variables on a spherical surface to observations. Such a spatial problem is illustrated in figure 2, where a Gauss-Legendre quadrature (GLQ) grid is displayed among randomly distributed observations. In Spherical Direct Sequential Simulation the spherical coordinates (r, θ, ϕ) for the surface and observations are each kept in arrays of length N_d and N_m respectively, except for the surface radius which is a constant, R . In addition each observation location has an associated measurement kept in array \mathbf{d} .

Finally, in order to later estimate the semi-variogram, by computing an $N_m \times N_m$ grid of the co-latitude and longitude pairs in radians, $\boldsymbol{\Theta}$ and $\boldsymbol{\Phi}$, the great circle distances between each surface location pair are determined through the haversine formula as follows

$$L_{GCD} = 2R \arcsin \left(\sqrt{\sin^2 \frac{\boldsymbol{\Phi}^T - \boldsymbol{\Phi}}{2} + \cos \boldsymbol{\Phi} \odot \cos \boldsymbol{\Phi}^T \odot \sin^2 \frac{\boldsymbol{\Theta}^T - \boldsymbol{\Theta}}{2}} \right) \quad (30)$$

Where R is the radius of the sphere, \odot denotes the element-wise product (Hadamard product), and the final array size is $N_m \times N_m$.

3.2. Prior information

For direct sequential simulation each location on the spherical surface needs an a priori realization of the random variable being modelled. This is the training image, \mathbf{m}_0 , which is contained in an array of length N_m , in order according to the coordinates of the sphere. The training image will have a target histogram (i.e. distribution), and based on this or other reasonable assumptions an a priori mean, μ_0 , and variance, σ_0^2 , must be set. Based on the training image values and their associated geometry, further prior information in the form of a semi-variogram model and a range of conditional distributions are generated. Below we describe the implementation details, starting with the conditional distributions.

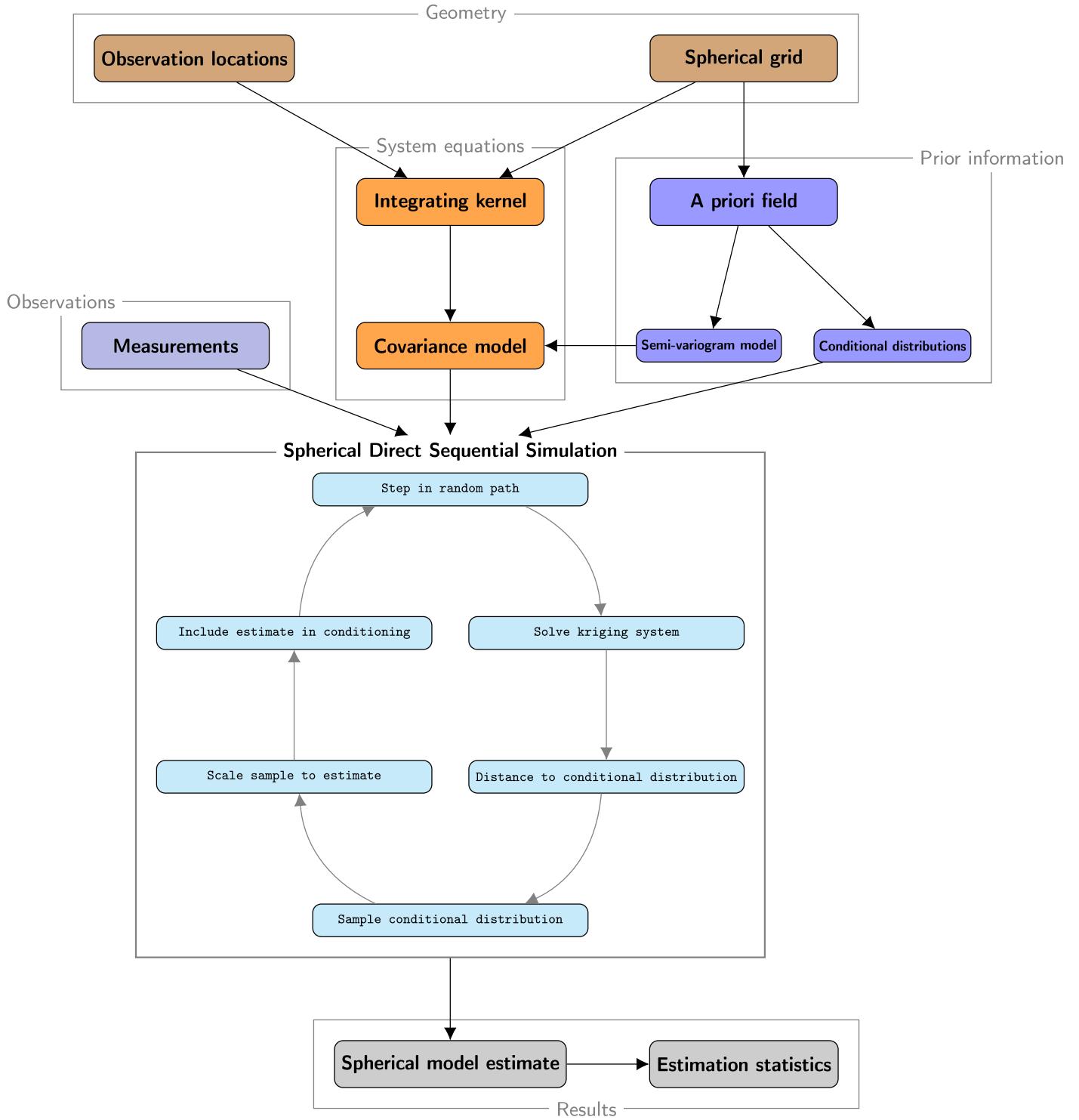


Figure 1: The implementation of Spherical Direct Sequential Simulation shown as a flowchart.

3.2.1. Conditional distributions

The conditional distributions are implemented in a lookup-table (LUT) according to the explanation in section 2.4, specifically equations (22)-(25). First the discrete quantile function, F^{-1} , of the target histogram is found by sorting \mathbf{m}_0 , then the number of quantiles, N_u , is set, which define the resolution of the distributions in the LUT. A uniformly spaced array, \mathbf{u} , in the range zero to one is generated containing the N_u quantiles and, along with a range

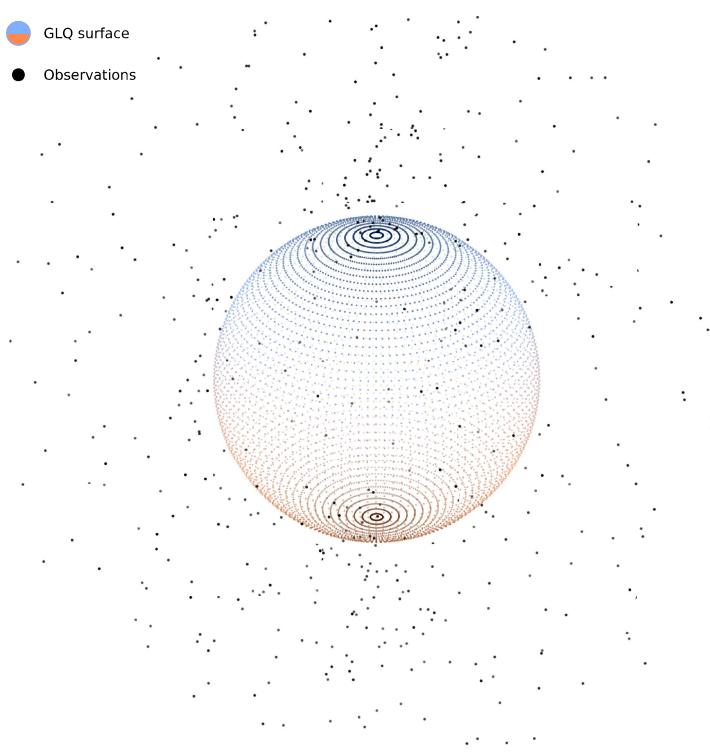


Figure 2: Illustration of Gauss-Legendre quadrature grid and observations located randomly in spherical space.

of mean and standard deviation values of length N_μ and N_σ respectively, is then used iteratively in equation (24) and (25) to generate the local conditional distribution LUT, \mathbf{Q} , of size (N_u, N_μ, N_σ) . In our implementation the Python package scikit-learn (Pedregosa, Varoquaux, Gramfort, Michel, Thirion, Grisel, Blondel, Prettenhofer, Weiss, Dubourg, Vanderplas, Passos, Cournapeau, Brucher, Perrot and Duchesnay, 2011) is used to handle the normal-score and inverse transformations. Through pseudo-code in algorithm 1 we show the above procedure.

Algorithm 1: Generating a local conditional distribution lookup-table

```

input :  $\mathbf{m}_0$ ,  $N_u$ 
output:  $\mathbf{Q}$ 
From  $\mathbf{m}_0$  compute  $F^{-1}$ 
Generate  $\mathbf{u}$  based on  $N_u$ 
 $i = 0, j = 0$ 
for  $\mu$  in the range  $-3.5 \dots 3.5$  do
    for  $\sigma$  in the range  $0.0 \dots 2.0$  do
         $\mathbf{y} = H^{-1}(\mathbf{u})\sigma + \mu$                                 // Equation (24)
         $\mathbf{q} = F^{-1}(H(\mathbf{y}))$                                 // Equation (25)
         $\mathbf{Q}_{ij} = \mathbf{q}.copy()$ 
         $j += 1$ 
    end
     $i += 1$ 
end

```

3.2.2. Semi-variogram models

In accordance with equation (15) and (16), semi-variogram values are calculated by ordering the squared location-pair difference values, $[m_0(s_i) - m_0(s_j)]^2$, for all grid locations in accordance to the great circle distances in \mathbf{L}_{GCD} . These distances are typically denoted the lag, h . Once the values are ordered according to lag, half the mean is taken over equal ranges of lags resulting in a semi-variogram which may be modelled using a positive definite function. (see

Gneiting, 2013, for a discussion of valid positive definite functions on spheres). Two such semi-variogram models are the classical exponential and spherical semi-variogram models in (31).

$$\begin{aligned} \text{Exponential} \quad \gamma(h) &= c_0 + c_1 \left(1 - \exp \left(-\frac{3h}{a} \right) \right) \\ \text{Spherical} \quad \gamma(h) &= \begin{cases} c_0 + c_1 \left(3/2 \frac{h}{a} - 1/2 \left(\frac{h}{a} \right)^3 \right) & \text{if } h \leq a \\ c_0 + c_1 & \text{if } h \geq a \end{cases} \end{aligned} \quad (31)$$

Where c_0 is a small variance contribution coined the nugget, which may be due to short scale noise in the training image, c_1 is a positive variance contribution defining the sill (max semi-variance), and a is the range at which the semi-variance is constant. Suitable semi-variogram model parameters are estimated by fitting to the training image semi-variogram, which in conjunction with the great circle distances based on the geometry of the sphere allow computation of a semi-variogram lookup-table, Γ , of size (N_m, N_m) through 31.

3.3. System equations

The final variables to be defined before spherical the direct simulation algorithm can be adequately described, are the variables through which the system equations are defined. First is the integrating kernel, \mathbf{G} , which defines the connection between the observations and spherical surface in equations (1) - (4), making it an array of size (N_d, N_m) . For now we leave out any details on computing the content of the integrating kernel as that will be problem dependent. However, in section 4 and 5 examples of integrating kernels will be shown through two cases.

3.3.1. Covariance model

As defined in (15) a covariance measure between two random variables on the spherical surface at any two locations s_i and s_j , may be estimated through a semi-variogram model. E.g. by fitting the training image semi-variogram to a model such as the ones shown in (31). Given the semi-variogram LUT, Γ , described in 3.2.2, a model parameter to model parameter covariance LUT is constructed as follows

$$\mathbf{C}_m = \sigma_0^2 \bar{\mathbf{E}} - \Gamma \quad (32)$$

Where $\bar{\mathbf{E}}$ is an (N_m, N_m) array of ones. With the model parameter covariance above, a LUT for the data to data and data to model parameter covariance is computed through (17) and (18) respectively, as

$$\mathbf{C}_d = \mathbf{G} \mathbf{C}_m \mathbf{G}^T \quad (33)$$

$$\mathbf{C}_{dm} = \mathbf{G} \mathbf{C}_m \quad (34)$$

With these covariance LUTs and the integrating kernel, all possible kriging systems as required in spherical direct sequential simulation and described by equation (12), can be constructed. In the following we explain explicitly how this is accomplished in the implemented algorithm.

3.4. Spherical Direct Sequential Simulation

1. Determine a random path through the target source locations.
2. At each location in the random path, the Kriging mean, μ_k , and variance, σ_k^2 , are calculated using all available observations and previously simulated values, as well as through the prior covariance information.
3. Find the correct local conditional distribution. This corresponds to the Gaussian distribution closest to the Kriging mean and variance pre-back-transformation.
4. A simulated value is drawn from this correctly shaped local conditional distribution.
5. The simulated value is scaled such that it originates from a correctly shaped local conditional distribution, with mean and variance equal to the Kriging mean and variance.

Variable	Content	Type	Size
Observations and Geometry			
N_d	Number of observations	float	1
N_m	Number of model parameters	float	1
R	Radius of the model parameter surface	float	1
θ_s, ϕ_s	Model parameter coordinates	arrays	$(N_m,)$
r_d, θ_d, ϕ_d	Observation coordinates	arrays	$(N_d,)$
d	Observed random variables	array	$(N_d,)$
L_{GCD}	Pairwise great circle distances	array	(N_m, N_m)
Prior information			
m_0	A priori model parameters	array	$(N_m,)$
μ_0	A priori mean	float	1
σ_o^2	A priori variance	float	1
N_u	Number of local conditional dist. quantiles	float	1
u	Uniformly spaced quantiles	array	$(N_u,)$
Q	Local conditional distribution LUT	array	(N_μ, N_σ, N_u)
Γ	Semi-variogram LUT	array	(N_m, N_m)
System equations			
G	Integrating kernel	array	(N_d, N_m)
C_m	Model covariance LUT	array	(N_m, N_m)
C_d	Data covariance LUT	array	(N_d, N_d)
C_{dm}	Data to model covariance LUT	array	(N_d, N_m)

Table 1
Implementation variables.

6. The simulated value is added to the conditional data for use in the rest of the simulation.
7. 3.-7. is repeated until all target source locations have been visited.

Algorithm 2: Spherical Direct Sequential Simulation

```

input :  $m_0, \sigma_0^2, Q, G, C_m, C_d, C_{dm}$ 
output:  $\hat{m}_{DSS}, \hat{C}_{DSS}$ 
From  $m_0$  compute  $F^{-1}$ 
Generate  $u$  based on  $N_u$ 
 $i = 0, j = 0$ 
for  $\mu$  in the range  $-3.5 \dots 3.5$  do
  for  $\sigma$  in the range  $0.0 \dots 2.0$  do
     $y = H^{-1}(u)\sigma + \mu$                                 // Equation (24)
     $q = F^{-1}(H(y))$                                      // Equation (25)
     $Q_{ij} = q.\text{copy}()$ 
     $j+ = 1$ 
  end
   $i+ = 1$ 
end

```

4. Synthetic example

5. Case study: Geomagnetic Observations

1. Synthetic observations
2. A priori grid map
3. A priori histogram
4. A priori semi-variogram
5. Power spectrum with reference and other models

5.1. Stochastic

1. Grid map realizations
2. Semi-variogram with model, reference, and realizations
3. Power spectrum with reference, realizations, and other models

5.2. Core synthetic

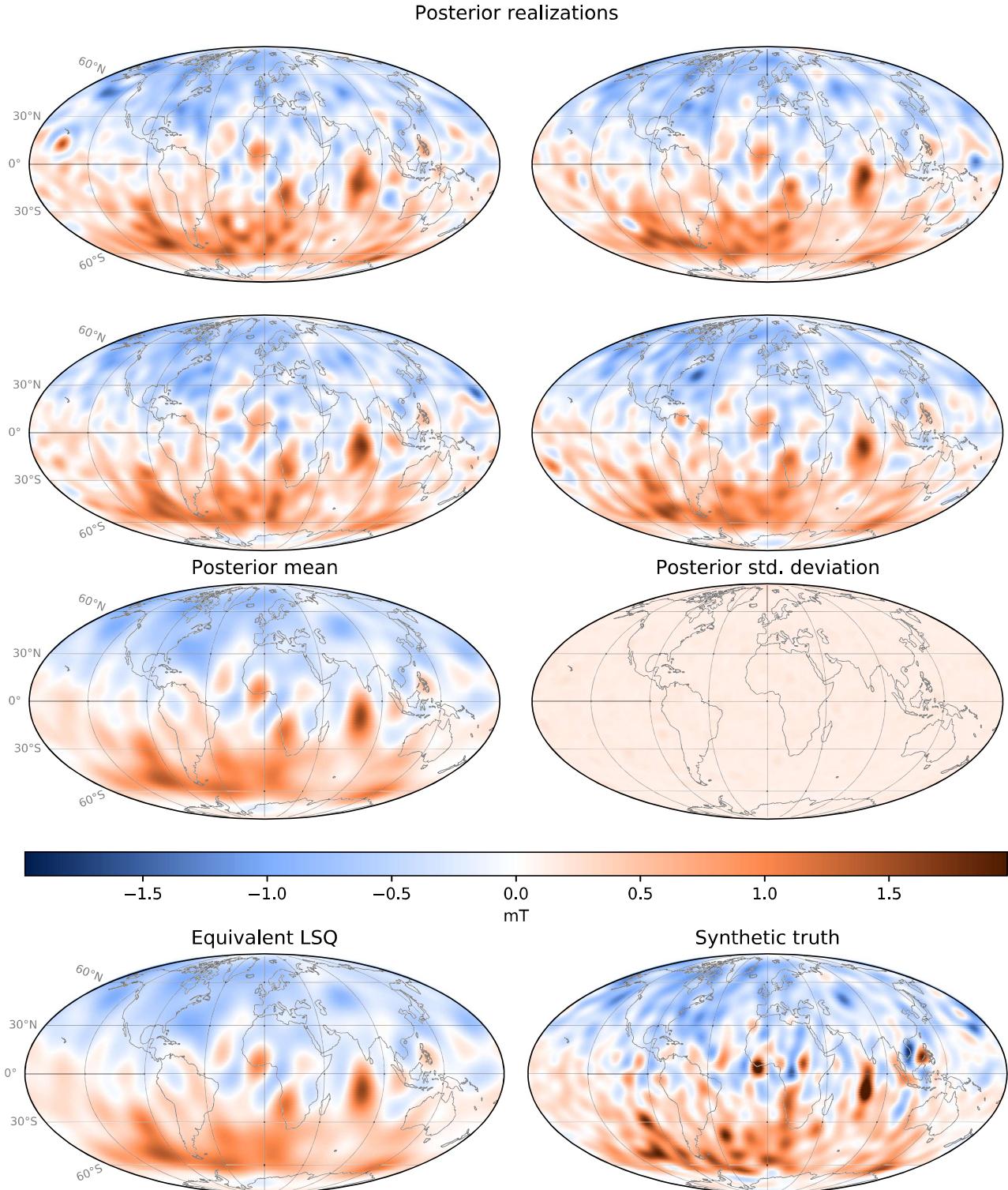


Figure 3: Posterior realizations based on an Aubert a priori field and synthetic observations at locations taken from Swarm A. Estimated to spherical harmonic degree 30. In solving the system, measurement error covariance has been added as a matrix with $9nT^2$ along the diagonal, corresponding to Gaussian white noise with a std. dev of $3nT$.

Direct sequential simulation for spherical linear inverse problems

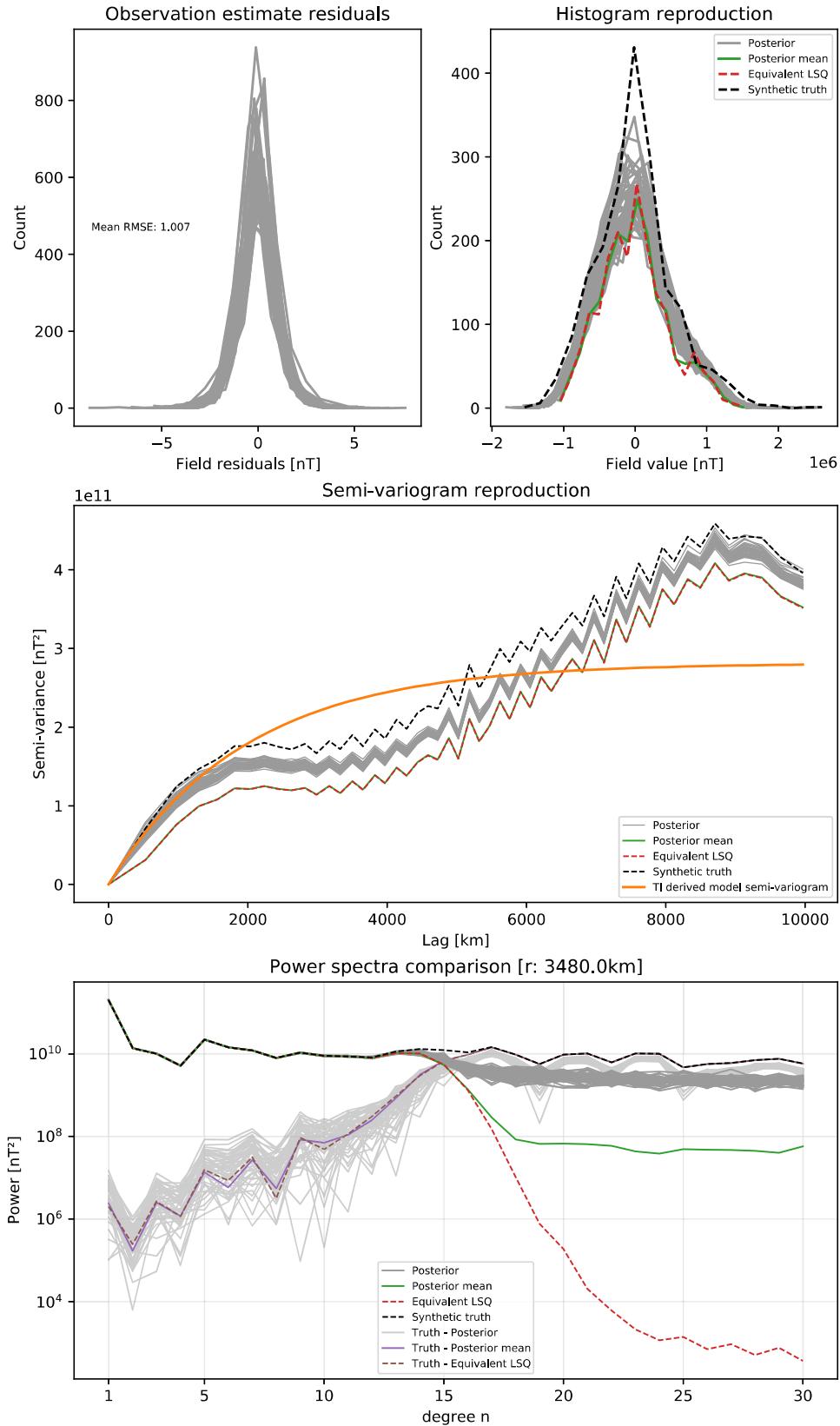


Figure 4: Analysis of posterior realizations based on an Aubert a priori field and synthetic observations at locations taken from Swarm A. Estimated to spherical harmonic degree 30. Field realizations are shown in figure 3.

5.3. Core observations

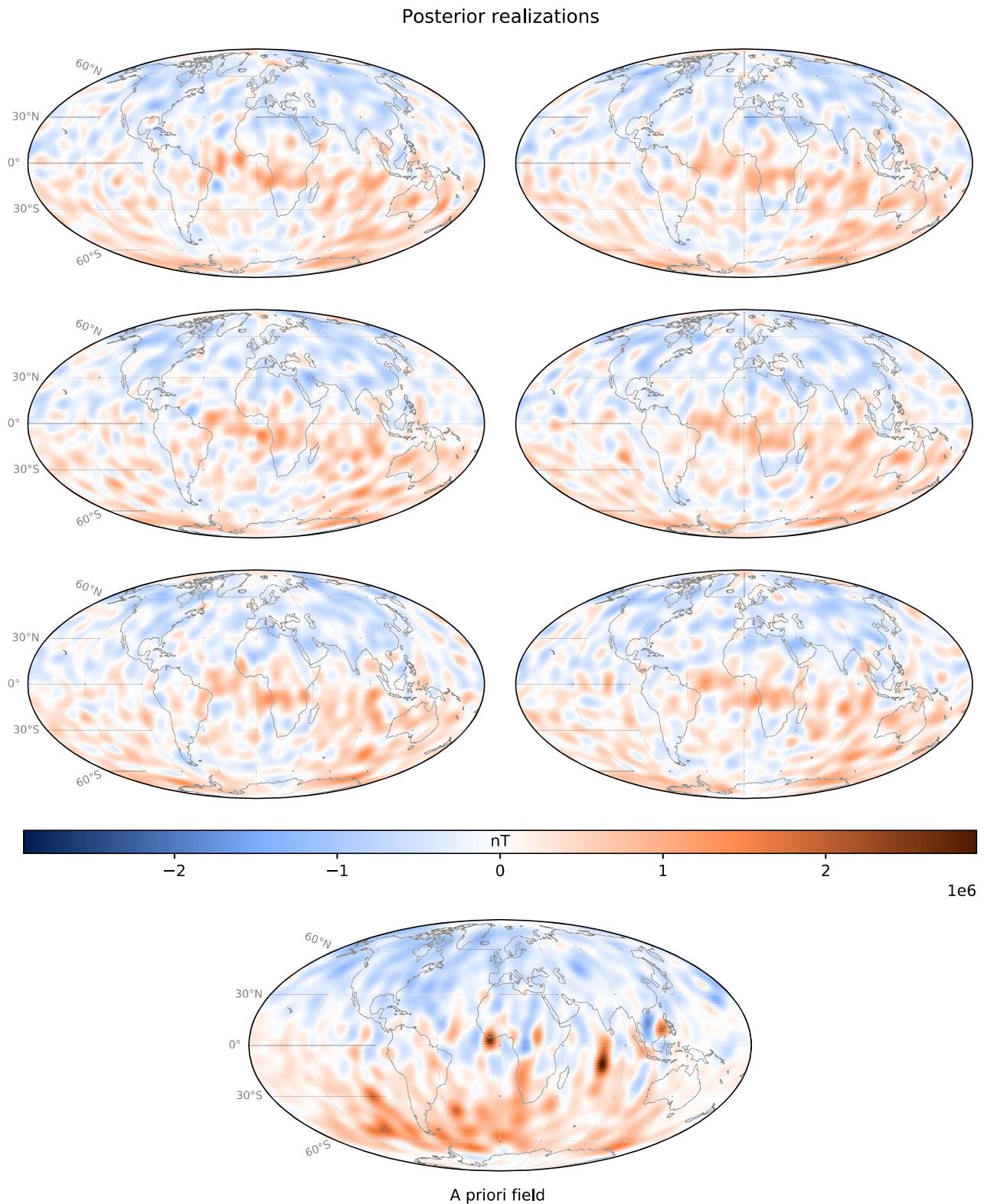


Figure 5: Posterior realizations based on an Aubert a priori field and real satellite observations from Swarm A. Estimated to spherical harmonic degree 30. In solving the system, measurement error covariance has been added as a matrix with $25nT^2$ along the diagonal, corresponding to Gaussian white noise with a std. dev of $5nT$.

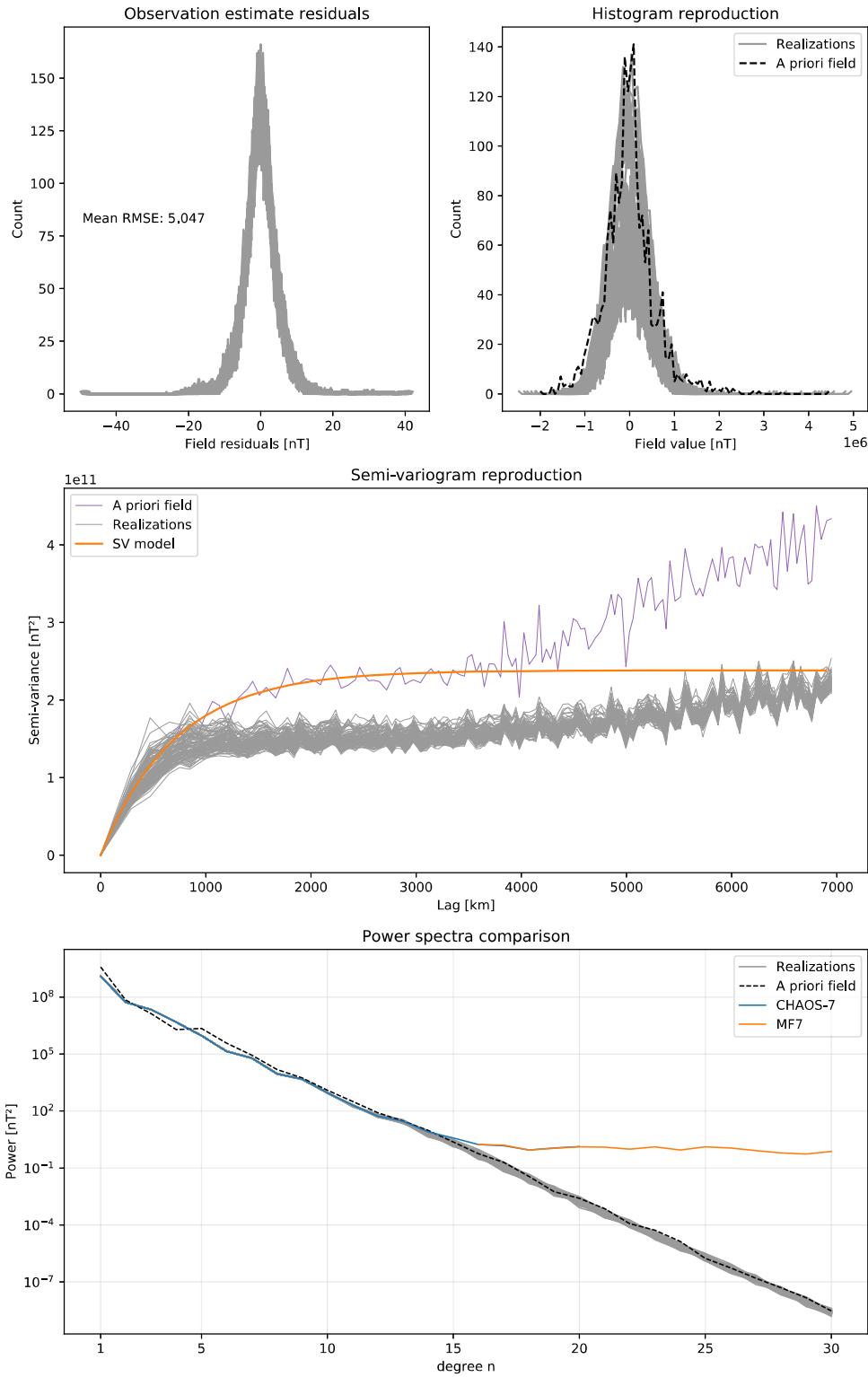


Figure 6: Analysis of posterior realizations based on an Aubert a priori field and real satellite observations from Swarm A. Estimated to spherical harmonic degree 30. Field realizations are shown in figure 5.