Computer exercise 3
Gaussian fields
Spatial statistics and image analysis, TMS016

1 Introduction

The purpose of this computer exercise is to give an introduction to how Gaussian random fields can be simulated and visualised in Matlab. Before you begin, download the Matlab files for the course from the course homepage. When in doubt about how to use a specific function in Matlab, use `help` and `doc` to get more information.

2 The multivariate Gaussian distribution

Let $\mathbf{x} \in \mathbb{R}^d$ have a multivariate Gaussian distribution $\sim \mathsf{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, where $\boldsymbol{\mu} \in \mathbb{R}^d$ and $\boldsymbol{\Sigma}$ is a $d \times d$ positive semi-definite matrix. An important property of the multivariate Gaussian distribution is that if $\mathbf{R}$ is a $d \times d$ matrix, and $\mathbf{b} \in \mathbb{R}^d$ is a vector, then

$$\mathbf{y} = \mathbf{R}\mathbf{x} + \mathbf{b} \sim \mathsf{N}(\mathbf{R}\boldsymbol{\mu} + \mathbf{b}, \mathbf{R}\boldsymbol{\Sigma}\mathbf{R}^T).$$

This means that if we take $\mathbf{z} \sim \mathsf{N}(\mathbf{0}, \mathbf{I})$, and $\mathbf{R}$ such that $\mathbf{R}^T\mathbf{R} = \boldsymbol{\Sigma}$, then $\boldsymbol{\mu} + \mathbf{R}^T\mathbf{z} \sim \mathsf{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$. A matrix $\mathbf{R}$ satisfying $\mathbf{R}^T\mathbf{R} = \boldsymbol{\Sigma}$ can be thought of as a square-root of the matrix $\boldsymbol{\Sigma}$. There are several such matrices, but since $\boldsymbol{\Sigma}$ is positive semi-definite, we can compute one such matrix using the Cholesky factor:

```
>> Sigma = [2 1 0; 1 2 1; 0 1 2];
>> R = chol(Sigma);
```

The Cholesky factor of $\boldsymbol{\Sigma}$ is the unique upper-triangular matrix with non-negative diagonal elements satisfying $\mathbf{R}^T\mathbf{R} = \boldsymbol{\Sigma}$. Simulating $\mathbf{z}$ is easy since it is a vector of independent normal random variables. Therefore, $\mathbf{x}$ can be simulated as

```
>> z = randn(3,1);
>> mu = [1 2 3]';
>> x = mu + R'*z;
```

3 Covariance functions

We now turn to the problem of creating the covariance matrix $\boldsymbol{\Sigma}$ when $\mathbf{x} = (X(\mathbf{s}_1), \ldots, X(\mathbf{s}_d))$ is a vector with the values of a Gaussian field X at some locations in $\mathbb{R}^d$. Use the function `matern_covariance` to compute and plot the Matérn covariance function. For example,

```
>> h = linspace(0,10,1e3);
>> r = matern_covariance(h,1,1,1);
>> plot(h,r)
```

- Investigate the behaviour of the Matérn covariance function when changing the parameters. Make sure that you understand the effect each parameter has on the covariance function.
- Two special cases of the Matérn covariance function are given in the functions `exponential_covariance` and `gaussian_covariance`. Try to obtain the same (or similar in the Gaussian case) covariance function by using `matern_covariance`.

- Do the same investigations with some covariance functions with compact support, such as `spherical_covariance` and `euclidshat_covariance`.

4 Simulation of Gaussian fields

We now want to use the covariance functions above to define and simulate Gaussian fields. Start by defining a set of locations $s_1, \dots, s_n$ on a regular lattice in $\mathbb{R}^d$ using the `meshgrid` function. To compute the covariance matrix for these locations, we first need to compute a distance matrix $\mathbf{D}$ with elements $D_{ij} = \|\mathbf{s}_i - \mathbf{s}_j\|$. If `loc` is the $n \times d$ matrix with the measurement locations, we can do this as

```
>> D = squareform(pdist(loc));
```

A covariance matrix can then be computed as for the covariance functions in the section above, where we replace `h` with `D`. If this covariance matrix is used for simulating a Gaussian field (as in Section 2), the realisation can then be visualised using

```
>> imagesc(reshape(x,[m n]));
```

where m and n are the dimensions of your lattice.

- Simulate Gaussian fields with Matérn covariance functions and study the effects the parameters have on the realisations. To easier understand the effect ν has on the simulations, you may want to rescale κ so that the practical correlation range r is constant when changing ν. This can be done using the relation $r \approx \sqrt{8\nu}/\kappa$.
- Simulate some Gaussian fields with compactly supported covariance functions. Can one see that the values are independent for distances larger than the correlation range?