Monte Carlo Methods Appl. **19** (2013), 73–75

Erratum
Fast simulation of Gaussian random fields

[Monte Carlo Methods Appl. 17 (2011), 195–214]

Annika Lang and Jürgen Potthoff

Abstract. In the paper "Fast simulation of Gaussian random fields", a typo occurred. Instead of $(p_{k_1 \cdots k_d})_i = k_i / l_i$ it should read $(p_{k_1 \cdots k_d})_i = (k_i - N_i/2)/l_i$ in Algorithm 3.1, Remark d). For convenience of the reader we reproduce below the complete corrected algorithms.

Keywords. Gaussian random fields, fast Fourier transform, simulation.

2010 Mathematics Subject Classification. 65C05, 65C50, 60G60, 60H40.

Algorithm 3.1.

Remarks

a) The functions FFT and FFT^{-1} include all necessary rescaling depending on the used FFT algorithm and the integers N_i.

b) A is a d-dimensional complex-valued array, B is real-valued.

c) $x_{k_1 \cdots k_d}$ denotes the grid point corresponding to the integers $(k_1, \ldots, k_d)$. The grid points are distributed equidistantly in each direction, i.e., the distance of two arbitrary neighbor grid points in direction e_i is given by a constant Δx_i.

d) The points $p_{k_1 \cdots k_d}$ in the Fourier domain are given by

$$(p_{k_1 \cdots k_d})_i = (k_i - N_i/2)/l_i \quad \text{for } i = 1, \ldots, d.$$

Input

a) D a d-dimensional rectangular region with $l_1, \ldots, l_d$ lengths of the edges,

b) $N_1, \ldots, N_d$ the numbers of discretization points in each direction, all even,

c) $\gamma^{1/2}$ a symmetric, positive function on $\mathbb{R}^d$,

d) $R()$ a function that generates independent $\mathcal{N}(0, |\Delta^N|^{-1})$-distributed random numbers.

Output

GRF B on D, where the covariance is given by the Fourier transform of γ.

for $k_i = 0, \dots, N_i - 1,\ i = 1, \dots, d$ **do**
  $B(k_1, \dots, k_d) \leftarrow R()$;
end for
$A \leftarrow \text{FFT}\, B$;
for $k_i = 0, \dots, N_i - 1,\ i = 1, \dots, d$ **do**
  $A(k_1, \dots, k_d) \leftarrow A(k_1, \dots, k_d) \cdot \gamma(p_{k_1 \cdots k_d})^{1/2} \cdot |D|^{-1}$;
end for
$B \leftarrow \text{FFT}^{-1} A$;

Algorithm 3.2.

Remarks Same as in Algorithm 3.1.
All calculations have to be done modulo N_i in the i-th direction.

Input Same as in Algorithm 3.1.

Output Same as in Algorithm 3.1.

for $k_i = 0, \dots, N_i - 1, i = 1, \dots, d-1, k_d = 0, \dots, N_d/2$ **do**
  if $k_i \in \{0, N_i/2\}$, for all $i = 1, \dots, d$ **then**
    $\text{Re}\, A(k_1, \dots, k_d) \leftarrow R() \cdot \gamma(p_{k_1 \cdots k_d})^{1/2} \cdot |D|^{-1}$;
    $\text{Im}\, A(k_1, \dots, k_d) \leftarrow 0$;
  else
    $\text{Re}\, A(k_1, \dots, k_d) \leftarrow 2^{-1/2} R() \cdot \gamma(p_{k_1 \cdots k_d})^{1/2} \cdot |D|^{-1}$;
    $\text{Im}\, A(k_1, \dots, k_d) \leftarrow 2^{-1/2} R() \cdot \gamma(p_{k_1 \cdots k_d})^{1/2} \cdot |D|^{-1}$;
    $\text{Re}\, A(N_1 - k_1, \dots, N_d - k_d) \leftarrow \text{Re}\, A(k_1, \dots, k_d)$;
    $\text{Im}\, A(N_1 - k_1, \dots, N_d - k_d) \leftarrow -\text{Im}\, A(k_1, \dots, k_d)$;
  end if
end for
$B \leftarrow \text{FFT}^{-1} A$;

Algorithm 3.3.

Remarks Same as in Algorithm 3.2.

Input Same as in Algorithm 3.2.

Output Same as in Algorithm 3.2.

function *create_random_field*$(D, N_1, \dots, N_d)$
$A \leftarrow \text{complex\_array}(N_1, \dots, N_d)$;

set_array($\{N_1, \dots, N_d\}$, { }, A);
$B \leftarrow \mathrm{FFT}^{-1} A$;
end function

function *set_array*($\{N_1, \dots, N_j\}$, $\{k_{j+1}, \dots, k_d\}$, A)
for $k_i = 0, \dots, N_i,\ i = 1, \dots, j-1, k_j = 1, \dots, N_j/2 - 1$ **do**
Re $A(k_1, \dots, k_d) \leftarrow 2^{-1/2} R() \cdot \gamma(p_{k_1 \cdots k_d})^{1/2} \cdot |D|^{-1}$;
Im $A(k_1, \dots, k_d) \leftarrow 2^{-1/2} R() \cdot \gamma(p_{k_1 \cdots k_d})^{1/2} \cdot |D|^{-1}$;
Re $A(N_1 - k_1, \dots, N_d - k_d) \leftarrow$ Re $A(k_1, \dots, k_d)$;
Im $A(N_1 - k_1, \dots, N_d - k_d) \leftarrow -$Im $A(k_1, \dots, k_d)$;
if ($|\{N_1, \dots, N_j\}| = 1$) **then**
Re $A(0, \dots, k_d) \leftarrow R() \cdot \gamma(p_{0\, k_2 \cdots k_d})^{1/2} \cdot |D|^{-1}$;
Im $A(0, \dots, k_d) \leftarrow 0$;
Re $A(N_1/2, \dots, k_d) \leftarrow R() \cdot \gamma(p_{N_1/2\, k_2 \cdots k_d})^{1/2} \cdot |D|^{-1}$;
Im $A(N_1/2, \dots, k_d) \leftarrow 0$;
else
set_array($\{N_1, \dots, N_{j-1}\}$, $\{0, k_{j+1}, \dots, k_d\}$, A);
set_array($\{N_1, \dots, N_{j-1}\}$, $\{N_j/2, k_{j+1}, \dots, k_d\}$, A);
end if
end for
end function

Received February 2, 2013; accepted February 18, 2013.

Author information

Annika Lang, Seminar für Angewandte Mathematik, ETH Zürich,
Rämistrasse 101, 8092 Zürich, Switzerland.
E-mail: annika.lang@sam.math.ethz.ch

Jürgen Potthoff, Institut für Mathematik, Universität Mannheim,
68131 Mannheim, Germany.
E-mail: potthoff@math.uni-mannheim.de