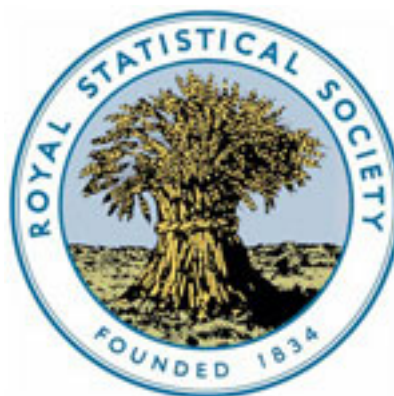


# WILEY



---

Algorithm AS 137: Simulating Spatial Patterns: Dependent Samples from a Multivariate Density

Author(s): B. D. Ripley

Source: *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, Vol. 28, No. 1 (1979), pp. 109-112

Published by: [Wiley](#) for the [Royal Statistical Society](#)

Stable URL: <http://www.jstor.org/stable/2346831>

Accessed: 05/04/2014 15:12

---

Your use of the JSTOR archive indicates your acceptance of the Terms & Conditions of Use, available at <http://www.jstor.org/page/info/about/policies/terms.jsp>

JSTOR is a not-for-profit service that helps scholars, researchers, and students discover, use, and build upon a wide range of content in a trusted digital archive. We use information technology and tools to increase productivity and facilitate new forms of scholarship. For more information about JSTOR, please contact [support@jstor.org](mailto:support@jstor.org).



Wiley and Royal Statistical Society are collaborating with JSTOR to digitize, preserve and extend access to *Journal of the Royal Statistical Society. Series C (Applied Statistics)*.

<http://www.jstor.org>

**Algorithm AS 137****Simulating Spatial Patterns : Dependent Samples from a Multivariate Density**

By B. D. RIPLEY

*Department of Mathematics, Imperial College, London**Keywords:* SIMULATION; SPATIAL PATTERN; STRAUSS'S DENSITY

LANGUAGE

ISO Fortran

DESCRIPTION AND PURPOSE

*Purpose*

Suppose we wish to generate samples of  $m$  points in  $d$ -dimensional space from a joint distribution with density  $p$ . Examples occur frequently when studying spatial patterns for which the normalizing constant is not known in closed form. We will assume that the density is symmetric in its arguments, i.e. independent of the ordering of the  $m$  random points.

If all the marginal densities were known, the obvious way to generate samples from the density would be to add points in turn drawn from the appropriate conditional densities. However, it is often impossible to find even the marginal density of a subset of  $(m-1)$  points. Rejection sampling is theoretically feasible whenever the density is bounded, by  $M$  say. For a process on  $[0, 1]^d$  generate  $m$  independent uniform random vectors  $x_1, \dots, x_m$  and compute  $X = p(x_1, \dots, x_m)/M$ . Generate an independent uniform variable  $Y$ . If  $X \geq Y$  accept  $(x_1, \dots, x_m)$  otherwise repeat the process. Although this process will terminate (with probability one) it is almost always prohibitively long. The  $m$  independent uniform random vectors may be replaced by a sample from any easily simulated density  $q$ , provided  $X$  is redefined as  $cp(x)/q(x)$  with  $c$  chosen so that  $P(X \leq 1) = 1$ . Useful choices of  $q$  for this envelope rejection technique are rare.

The method used here exploits the observation that the conditional density  $p(x_1 | x_2, \dots, x_m)$  is known up to a constant depending only on  $(x_2, \dots, x_m)$  and so may be simulated by the rejection technique. The algorithm is mainly to illustrate the technique; the example used is Strauss's density as given by Kelly and Ripley (1976). Define two points to be *neighbours* if their distance apart is less than  $R$ . Then  $p(x_1, \dots, x_m)$  is proportional to  $c^s$  where  $s$  is the number of pairs of neighbours in  $(x_1, \dots, x_m)$ , and  $0 \leq c < \infty$ . The case  $c = 0$  is one definition of the random distribution of non-overlapping spheres of radius  $\frac{1}{2}R$ , cf. Ripley (1977, Section 3.2). Values of  $c$  between 0 and 1 provide a gradation between this and a "random" pattern.

## THEORY

The algorithm works by repeating a single step in which one of the points is chosen at random and deleted, then replaced by a point drawn from the conditional density given the remaining  $(m-1)$  points. For Strauss's density this conditional density is proportional to  $c^t$ , where  $t$  is the number of neighbour pairs which would be created if that point were added. If the process is in equilibrium then samples a long distance apart in time will be almost independent. It is not clear how one should test statistically independence of two spatial patterns; the algorithm take samples every  $4m$  steps which have always proved adequately independent. For some applications samples could be taken much more frequently. It is necessary to supply a starting pattern. Ideally this should be a sample from the density  $p$  or a very similar density but the theory below shows that any starting pattern will do and  $m$  independent uniformly distributed points are usually adequate. For Strauss's density a

suitable choice may be the distinct but very similar sequential version in which when  $r$  points are present the next is generated with density proportional to  $c^t$ , where  $t$  is again the number of extra pairs of neighbours which would be created.

It is clear from the description that the basic step describes the transition of a Markov process. The method is related to the Markov processes used in statistical mechanics and surveyed by Hastings (1970). Formally the statespace is  $R^{md}$  but we can avoid technical complications by noting that in computing we replace this by a finite set of numbers. Thus we take the statespace to be those representable patterns with positive density; a finite set. The required distribution is an equilibrium distribution of this Markov chain; if we start with this distribution after dropping a point we have the marginal distribution on  $(m-1)$  points to which we add a point with the necessary conditional density. The Markov chain is aperiodic, so if it is irreducible the distribution after  $m$  steps converges to the required distribution *whatever* the initial state (Cox and Miller, 1965, Section 3.8). Irreducibility is almost always satisfied but rules out densities positive only if all the points lie in the same quadrant, for example.

#### EXTENSIONS

The algorithm will still work if points are deleted in turn rather than chosen for deletion at random. If the state of the stochastic process is the *ordered* set of points  $(x_1, \dots, x_m)$  and the basic step is to replace  $x_1$  then relabel as  $(x_2, \dots, x_m, x_1)$  the process is again a Markov chain, now of period  $m$  but with equilibrium distribution unique up to the ordering of the points. This modification has the advantage of ensuring that samples taken every  $m$  steps have no points in common.

The density can easily be altered. A useful alternative is the family of pairwise interaction processes (Ripley, 1977, Section 3.4) for which

$$p(x_1, \dots, x_m) \propto \prod_{i < j} h\{d(x_i, x_j)\}, \quad p(x_1 | x_2, \dots, x_m) \propto \prod_{j > 1} h\{d(x_j, x_1)\}.$$

#### STRUCTURE

##### SUBROUTINE SIMPAT ( $NPT, X, Y, C, R$ )

###### Formal parameters

$NPT$	Integer	input: number of points
$X$	Real array ( $NPT$ )	input: $x$ -coordinates of initial pattern
$Y$	Real array ( $NPT$ )	input: $y$ -coordinates of initial pattern
$C$	Real	input: parameter $c$
$R$	Real	input: parameter $R$
$X$	Real array ( $NPT$ )	output: $x$ -coordinates of sample
$Y$	Real array ( $NPT$ )	output: $y$ -coordinates of sample

###### Failure

No failures are indicated. The only cause should be an inadmissible set of parameters which should be trapped before the first call.

###### Auxiliary algorithms

*SIMPAT* calls a function

##### FUNCTION DEN ( $X, Y, NPT, C, R$ )

which calculates  $d(x_1, \dots, x_m)$  where  $x_r = \{X(r), Y(r)\}$ . There is also a call to a function *RANF* which should return a pseudo-random number uniformly distributed on the unit interval. (With a suspect generator it would be desirable to use separate generators for  $X(1)$  and  $Y(1)$ .) The statement function *FA* allows periodic boundary conditions.

RESTRICTIONS

The parameter  $C$  may be any non-negative real number. The interaction range  $R$  is formally restricted only when  $C = 0$ . Then the packing density  $\rho = NPT\pi R^2/4$  must not exceed 0.9609 (Rogers, 1964). However, for small  $C$  and large values of  $R$  or large  $C$  and small values of  $R$  there will be many rejections.

TIMING

The main subroutine takes  $O(m)$  steps. For most densities (including Strauss's) the function  $DEN$  takes  $O(m)$  operations on each call, so  $O(m^2)$  total operations are needed. Table 1 gives some timings for one sample in seconds for Strauss's density. The machine was the CDC 6600 of the University of London Computer Centre.

TABLE 1

*Time in seconds for one sample on a CDC 6600*

$NPT$	$C$	$R$	$\rho$	Time
25	0	0.05	0.049	0.08
25	0	0.10	0.196	0.17
25	0	0.15	0.441	0.99
50	0	0.02	0.016	0.31
50	0	0.04	0.063	0.38
50	0	0.06	0.141	0.56
50	0	0.08	0.251	1.01
50	0	0.10	0.393	3.11
100	0	0.01	0.008	1.25
100	0	0.03	0.071	1.54
100	0	0.06	0.283	5.82
50	0.2	0.04	0.063	0.37
50	0.5	0.04	0.063	0.32
50	0.8	0.04	0.063	0.32
50	0.2	0.08	0.251	0.71
50	0.5	0.08	0.251	0.49
50	0.8	0.08	0.251	0.34

ADDITIONAL COMMENTS

The efficiency of the algorithm can be improved in special cases. For the case  $c = 0$   $DEN$  can be altered to return the value 0 immediately a distance less than  $R$  is found. For large samples the  $TILE$  algorithm of Green and Sibson (1978) may provide a more efficient way to count the neighbours of  $x_1$ . The rejection technique for  $p(x_1 | x_2, \dots, x_m)$  may possibly be made more efficient by envelope rejection sampling.

The hypercube  $[0, 1]^d$  may be replaced by any compact region (of positive measure) by generating  $x_1$  uniformly in  $A$ ; with a suitable density for  $x_1$   $A$  could even be unbounded.

For  $c > 1$  a better bound for  $q$  can be found by finding the maximum number of neighbours an added point could have in  $(x_2, \dots, x_m)$ , to replace  $m - 1$ . Unfortunately I cannot see how to do this in less than  $O(m^2)$  operations. For  $c$  much larger than 1 the algorithm as given is rather slow.

REFERENCES

COX, D. R. and MILLER, H. D. (1965). *The Theory of Stochastic Processes*. London: Methuen.  
 GREEN, P. J. and SIBSON, R. (1978). Computing Dirichlet tessellations in the plane. *Computer J.*, **21**, 168-173.  
 HASTINGS, W. K. (1970). Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, **57**, 97-109.  
 KELLY, F. P. and RIPLEY, B. D. (1976). A note on Strauss's model for clustering. *Biometrika*, **63**, 357-360.  
 RIPLEY, B. D. (1977). Modelling spatial patterns (with Discussion). *J. R. Statist. Soc. B*, **39**, 172-212.  
 ROGERS, C. A. (1964). *Packing and Covering*. London: Cambridge University Press.

```

SUBROUTINE SIMPAT(NPT, X, Y, C, R)
C
C   ALGORITHM AS 137 APPL. STATIST. (1979) VOL.28, NO.1
C
C   GENERATE SAMPLES OF SIZE NPT FROM A BIVARIATE JOINT DENSITY
C
C   DIMENSION X(NPT), Y(NPT)
C
C   RN = NPT
C   MM = 4 * NPT
C
C   MM IS THE NUMBER OF STEPS
C
C   DO 100 I = 1, MM
C   ID = RN * RANF(DUM) + 1.0
C   X(ID) = X(1)
C   Y(ID) = Y(1)
C 20 X(1) = RANF(DUM)
C   Y(1) = RANF(DUM)
C   F = DEN(X, Y, N, C, R)
C   IF (F .LE. RANF(DUM)) GOTO 20
C 100 CONTINUE
C   RETURN
C   END
C
C   FUNCTION DEN(X, Y, N, C, R)
C
C   ALGORITHM AS 137.1 APPL. STATIST. (1979) VOL.28, NO.1
C
C   CALCULATES STRAUSS CONDITIONAL DENSITY. DEN IS
C   PROPORTIONAL TO C ** (NUMBER OF NEIGHBOURS OF
C   THE FIRST POINT)
C
C   DIMENSION X(N), Y(N)
C   FA(A) = A
C
C   FOR TORUS FA(A) = AMIN1 (ABS(A), 1.0 - ABS(A))
C
C   DEN = 1.0
C   RR = R * R
C   DO 20 I = 2, N
C   X1 = FA(X(I) - X(1))
C   Y1 = FA(Y(I) - Y(1))
C   IF (X1 * X1 + Y1 * Y1 .LT. RR) DEN = C * DEN
C 20 CONTINUE
C
C   NORMALIZE BY THE MAX. OF THE DENSITY
C
C   IF (C .GT. 1.0) DEN = DEN / C ** (N - 1)
C   RETURN
C   END

```

## Remark AS R29

### Remarks on AS 110: $L_p$ Norm Fit of a Straight Line

By M. A. PORTER and D. J. WINSTANLEY

*Thames Polytechnic, London*

Sposito *et al.* (1977) give an algorithm for fitting a straight line to data by minimizing the  $L_p$  norm  $\sum |y_i - \alpha - \beta x_i|^p$ , summing over  $i$  from 1 to  $n$ , for a set of  $n$  observations  $(x_i, y_i)$  and for values of  $p$  in the interval  $1 \leq p < 2$ . We have found that the deletion of observations near the line can lead to their routine terminating before the minimum norm solution has been obtained.