```
######################################## Superthinning .
## First simulate a Hawkes process.
## Then calculate lambda at all the points.
## Then superthin.
## lambda(t,x,y) = mu rho(x,y) + K SUM gt(t-t_i)gxy(x-xi,y-yi),
    ##### with rho(x,y) = 1/(X1Y1),
    ##### gt(t) = beta e^(-beta t),
    ##### g(x,y) = alpha/pi exp(-alpha r^2), with x^2+y^2=r^2,
theta0 = list(mu=.08,K=.75,alpha=2.5,beta=3.5,b=1)
T = 10^3
X1 = 1
Y1 = 1
M0 = 3.5
z = simhawk(T=10^3,gmi=pointprod, gt=expgt, gxy = expxy, theta=theta0)
mu = theta0$mu; K = theta0$K; alpha = theta0$alpha; beta=theta0$beta
lambda = rep(mu/X1/Y1,z$n)
const = K*alpha/pi*beta
for(j in 2:(z$n)){
   gij = 0
   for(i in 1:(j-1)){
    r2 = (z$lon[j]-z$lon[i])^2+(z$lat[j]-z$lat[i])^2
    gij = gij + exp(-beta*(z$t[j]-z$t[i])-alpha*r2)
    }
   lambda[j] = mu / X1 / Y1 + const*gij
}
f = function(t,x,y,z){
## compute lambda(t,x,y) given data, z.
const = K*alpha/pi*beta
gij = 0
j = 0
if(t > z$t[1]) j = max(c(1:z$n)[z$t<t])
if(j>0) for(i in 1:j){
    r2 = (x-z$lon[i])^2+(y-z$lat[i])^2
    gij = gij + exp(-beta*(t-z$t[i])-alpha*r2)
}
mu / X1 / Y1 + const*gij
}

supthin = function(z,lambda,f,b=mean(lambda)){
## z = data, lambda = conditional intensity at pts, f = function to
compute lambda,
## and b = resulting rate.
## First thin, then superpose
keepz = list()
for(i in 1:z$n){
if(runif(1) < b/lambda[i]){
keepz$t = c(keepz$t,z$t[i])
keepz$lon = c(keepz$lon,z$lon[i])
keepz$lat = c(keepz$lat,z$lat[i])
}
```

```
}
candn = rpois(1,b*X1*Y1*T)
candt = sort(runif(candn)*T)
candx = runif(candn)*X1
candy = runif(candn)*Y1
for(i in 1:candn){
v = f(candt[i],candx[i],candy[i],z)
if(v < b){
if(runif(1) < (b-v)/b){
keepz$t = c(keepz$t,candt[i])
keepz$lon = c(keepz$lon,candx[i])
keepz$lat = c(keepz$lat,candy[i])
}}
}
keepz$lon = keepz$lon[order(keepz$t)]
keepz$lat = keepz$lat[order(keepz$t)]
keepz$t = sort(keepz$t)
keepz$n = length(keepz$t)
keepz
}

s = supthin(z,lambda,f)
par(mfrow=c(1,2))
plot(z$lon,z$lat,pch=3,cex=.5,xlab="lon",ylab="lat",main="original
pts.")
plot(s$lon,s$lat,pch=3,cex=.5,xlab="lon",ylab="lat",main="superthinned
points")
par(mfrow=c(1,1))
plot(z$lon,z$lat,pch=3,cex=.5,xlab="lon",ylab="lat",col="green")
points(s$lon,s$lat,pch=1,cex=.5)

##################################### ppm
library(spatstat)
ppm(X ~ 1, Strauss(r=0.1), ....)
```

fits the stationary Strauss process with interaction radius r = 0.1, and

```
ppm(X ~ x, Strauss(r=0.07), ....)
```

fits the non-stationary Strauss process with a loglinear spatial trend of the form $b(x, y) = \exp(a + bx)$.

the Strauss process on W with parameters $\beta > 0$ and $0 \leq \gamma \leq 1$ and interaction radius $r > 0$, has conditional intensity

$\lambda(u, x) = \beta \cdot \gamma$^t(u,x)

where t(u, x) is the number of points of x that lie within a distance r of the location u. If $\gamma < 1$, the term γ^t(u,x) makes it unlikely that the pattern will contain many points that are close together.

The pairwise interaction process on W with trend or activity function

$b_\theta : W \to R+$ and interaction function $h_\theta : W \times W \to R+$ has conditional intensity

$$\lambda(u, x) = b_\theta(u) \prod h_\theta(u, x_i).$$

Our technique only estimates parameters θ for which the model is in "canonical exponential family" form,

$$f(x; \theta) = \alpha(\theta) \exp(\theta^T V(x))$$

$$\lambda_\theta(u, x) = \exp(\theta^T S(u, x)),$$

where $V(x)$ and $S(u, x)$ are statistics, and $\alpha(\theta)$ is the normalising constant.

```
X <- rStrauss(100,0.7,0.05) ## beta, gamma, r.
fit <- ppm(X, ~1, Strauss(r=0.05))
exp(coef(fit))
fit <- ppm(X, ~1, Strauss(r=0.1))
exp(coef(fit))
plot(fit)

X = ppp(z$lon, z$lat, c(0,1), c(0,1),marks=z$t)
plot(X)

X = ppp(z$lon, z$lat, c(0,1), c(0,1))
fit <- ppm(X, ~1, Strauss(r=0.1))
exp(coef(fit))
plot(fit)

fit = ppm(X, ~ polynom(x, y, 2), Poisson())
summary(fit)
plot(fit)

fit <- ppm(X, ~x+y, Strauss(r=0.1))
plot(fit)
plot(fit) ## and hit stop
points(z$lon,z$lat,pch=3)

fit <- ppm(X, ~x+y, Strauss(r=0.05))
fit <- ppm(X, ~x+y, Strauss(r=.1))
exp(coef(fit))
plot(fit)

fit = ppm(X, ~ sqrt(x^2 + y^2), Poisson())
plot(fit)

fit = ppm(X, ~ polynom(x, y, 2), Poisson())
summary(fit)
plot(fit)
pf <- predict(fit) ## for just the trend.
plot(pf)
coef(fit)

data(cells)
```

```
m <- ppm(cells,~polynom(x,y,2),Strauss(0.05),rbord=0.05)
trend <- predict(m,type="trend",ngrid=100)
cif <- predict(m,type="cif",ngrid=100)
persp(trend,theta=-30,phi=40,d=4,ticktype="detailed",zlab="z")
persp(cif,theta=-30,phi=40,d=4,ticktype="detailed",zlab="z")

## You can also fit models with covariates in ppm.
## See SpatStatIntro.pdf.

########################## Nonparametric Hawkes process modelling.
if(!require(devtools)) install.packages('devtools')
devtools::install_github('mrjoshuagordon/nphawkes')
library(nphawkes)
# install.packages("ggplot2")
library(ggplot2)
help(nphawkesMSTNH)

mydata = nphData(data = z, time_var = "t")
output = nphawkesTNS(data = mydata, nbins_t = 15, nbins_mu = 50,
bw=100)
plot(output,type="time")
plot(output, type = 'background', smooth = TRUE)

mydata = nphData(data = z, time_var = "t", x_var = "lon", y_var =
"lat", mag = "m")
output = nphawkesMSTNH(data = mydata,xrange = c(0,1), yrange = c(0,1))
plot(output, type = 'time')
plot(output, type = 'space')
plot(output, type = 'magnitude')
plot(output, type = 'background')

## compare the fit.
r1 = output$est_time
r = seq(0,20,length=100)
s = theta0$beta * exp(-theta0$beta * r)
plot(r,s,type="l",col="green",xlab="t",ylab="g(t)")
for(i in 1:length(r1$t_lb)) if(r1$est[i] > 0.01)
lines(c(r1$t_lb[i],r1$t_ub[i]),rep(r1$est[i],2))

g2 = expxy(100000,1,theta0)
library(MASS)
g3 = kde2d(g2[,1],g2[,2],lims=c(-1,1,-1,1))
image(g3,main="spatial triggering")
contour(g3,add=T)
```