

Algoritmo de Metropolis-Hastings por componentes

Um vetor aleatório $(X_1, X_2)^\top$ tem função densidade

$$f(x_1, x_2) \propto \frac{(x_1 x_2)^5}{2^{x_1 - 1}} \left(1 - \frac{1}{2^{x_1}}\right)^{x_2 - 1} \exp(- (5x_1 + 7x_2)), \quad (1)$$

em que $x_1 > 0$ e $x_2 > 0$.

Será apresentado um gerador de amostras do vetor $(X_1, X_2)^\top$.

A partir da expressão (1), as distribuições condicionais completas têm funções densidade tais que

$$f(x_1|x_2) \propto \frac{x_1^5}{2^{x_1}} \left(1 - \frac{1}{2^{x_1}}\right)^{x_2 - 1} \exp(-5x_1) \quad (2)$$

e

$$f(x_2|x_1) \propto x_2^5 \left(1 - \frac{1}{2^{x_1}}\right)^{x_2} \exp(-7x_2). \quad (3)$$

Os núcleos das funções em (2) e (3) não correspondem a distribuições conhecidas (pelo menos de imediato). Utilizamos o algoritmo de Metropolis-Hastings por componentes. Tanto em (2) quanto em (3) adotamos uma distribuição proposta gama(ax, a) com esperança $ax/a = x$ e variância $ax/a^2 = x/a$, sendo que a seleção do valor de a está relacionada com a taxa de aceitação dos candidatos gerados.

```
# Núcleos das distribuições condicionais completas
f1 <- function(x1, x2) {
  x1^5 * (1 - 1 / 2^x1)^(x2 - 1) * exp(-5 * x1) / 2^x1
}

f2 <- function(x2, x1) {
  x2^5 * (1 - 1 / 2^x1)^x2 * exp(-7 * x2)
}

# Constantes para o amostrador
descarte <- 5000
espac <- 5
M <- 3000
nsim <- descarte + espac * M

# Distribuição proposta: gama(forma = a * x, taxa = a)
a10 <- 2
a20 <- 2
```

```

# Valores iniciais
x1 <- x2 <- c()
x1[1] <- 0.5
x2[1] <- 1

# Contadores de aceitações
nac1 <- nac2 <- 0

# Simulação
for (j in 2:nsim) {
  # candidato x1
  y1 <- rgamma(1, shape = a10 * x1[j - 1], rate = a10)

  # probabilidade x1
  alfa <- min(1, f1(y1, x2[j - 1]) * dgamma(x1[j - 1], a1 * y1,
    ate = a1) / (f1(x1[j - 1], x2[j - 1]) * dgamma(y1, a1 * x1[j - 1],
    rate = a1)))

  # geração x1
  if (runif(1) <= alfa) {
    x1[j] <- y1
    nac1 <- nac1 + 1
  } else {
    x1[j] <- x1[j - 1]
  }

  # candidato x2
  y2 <- rgamma(1, shape = a20 * x2[j - 1], rate = a20)

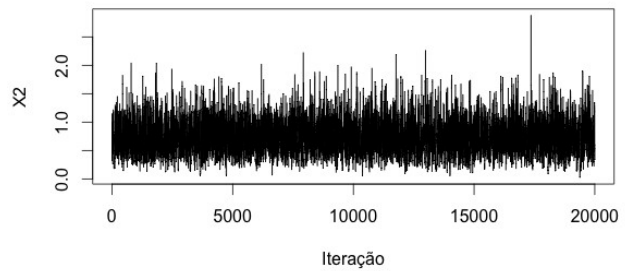
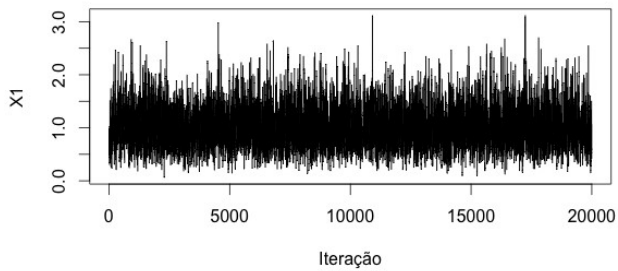
  # probabilidade x2
  alfa <- min(1, f2(y2, x1[j]) * dgamma(x2[j - 1], a2 * y2,
    rate = a2) / (f2(x2[j - 1], x1[j]) * dgamma(y2, a2 * x2[j - 1],
    rate = a2)))

  # geração x2
  if (runif(1) <= alfa) {
    x2[j] <- y2
    nac2 <- nac2 + 1
  } else {
    x2[j] <- x2[j - 1]
  }
}

cat("\n Taxa de aceitação de X1 (%):", round(nac1 / (nsim - 1) * 100, 1))
cat("\n Taxa de aceitação de X2 (%):", round(nac2 / (nsim - 1) * 100, 1))
  Taxa de aceitação de X1 (%): 50.7
  Taxa de aceitação de X2 (%): 44.5

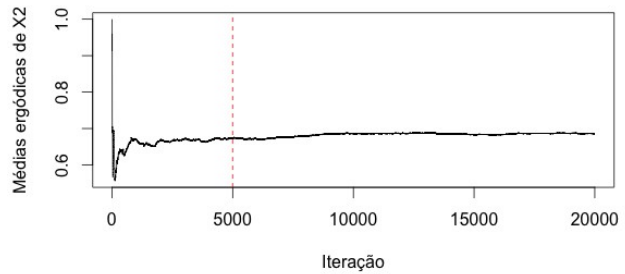
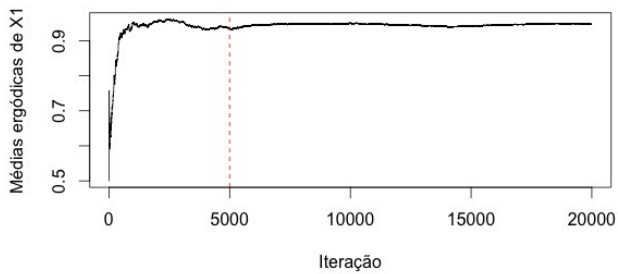
## Gráficos
# Sequência
plot(x1, type = "l", xlab = "Iteração", ylab = "X1")
plot(x2, type = "l", xlab = "Iteração", ylab = "X2")

```



```
# Médias ergódicas
plot(cumsum(x1) / (1:nsim), type = "l", xlab = "Iteração",
     ylab = "Médias ergódicas de X1")
abline(v = descarte, lty = 2, col = "red")

plot(cumsum(x2) / (1:nsim), type = "l", xlab = "Iteração",
     ylab = "Médias ergódicas de X2")
abline(v = descarte, lty = 2, col = "red")
```



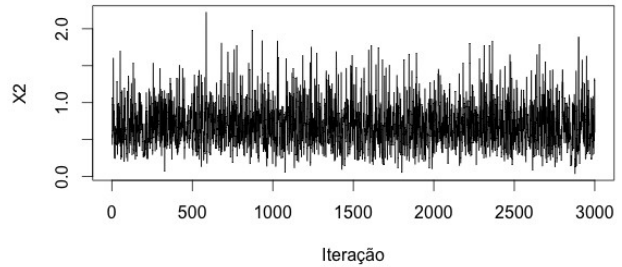
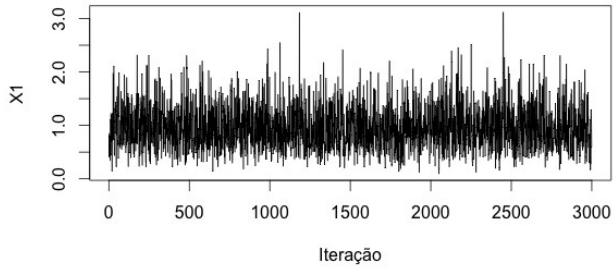
```
# Amostras (descarte e espaçamento levados em conta)
indices <- seq(descarte + 1, nsim, by = espac)
x1 <- x1[indices]
x2 <- x2[indices]

for (nome in c("x1", "x2")) {
  cat("\n Estatísticas descritivas de", nome, "\n")
  print(summary(get(nome)))
}
```

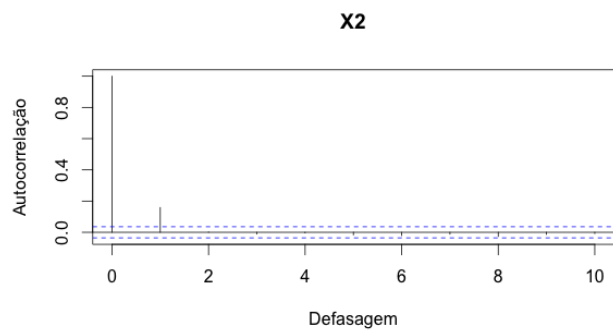
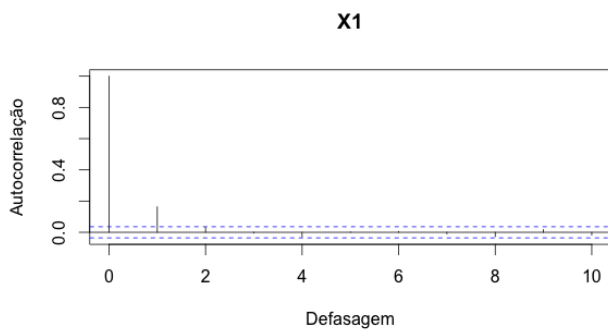
```
Estatísticas descritivas de x1
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
0.09214 0.65310 0.90160 0.95060 1.20200 3.11800
```

```
Estatísticas descritivas de x2
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
0.03582 0.47200 0.65400 0.69050 0.86730 2.22400
```

```
plot(x1, type = "l", xlab = "Iteração", ylab = "X1")
plot(x2, type = "l", xlab = "Iteração", ylab = "X2")
```

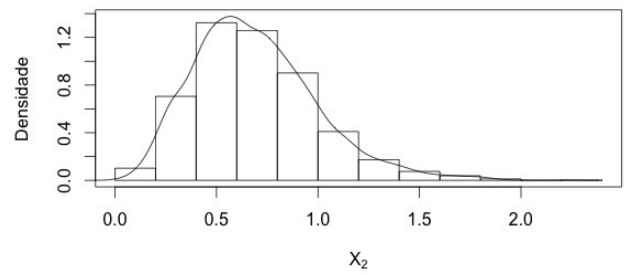
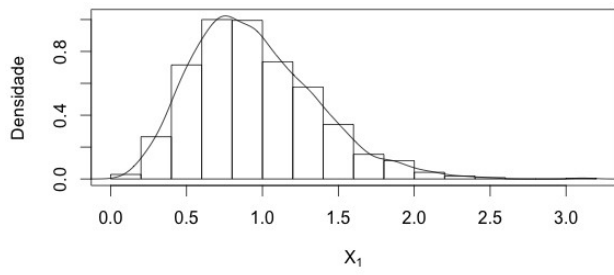


```
acf(x1, lag.max = 10, main = "X1", xlab = "Defasagem",
    ylab = "Autocorrelação")
acf(x2, lag.max = 10, main = "X2", xlab = "Defasagem",
    ylab = "Autocorrelação")
```



```
# X1
fhist <- hist(x1, plot = FALSE)
fdens <- density(x1)
inty <- c(0, max(fhist$density, fdens$y))
plot(fhist, freq = FALSE, main = "", xlab = expression(X[1]),
     ylab = "Densidade", ylim = inty)
lines(fdens)
box()

# X2
fhist <- hist(x2, plot = FALSE)
fdens <- density(x2)
inty <- c(0, max(fhist$density, fdens$y))
plot(fhist, freq = FALSE, main = "", xlab = expression(X[2]),
     ylab = "Densidade", ylim = inty)
lines(fdens)
box()
```



Nota 1. Obtenha amostras com outras escolhas para as distribuições propostas (a10 e a20, pag. 1).