



## **Casa abierta al tiempo**

Universidad Autónoma Metropolitana  
Unidad Iztapalapa  
División de Ciencias Básicas e Ingeniería

# **“Inferencia bayesiana en modelos de ecuaciones diferenciales: Un enfoque computacional”**

Tesis que presenta:

**Erick Iván Guerrero Flores**

Matrícula 2192802601

Para obtener el grado de:

**Maestro en Ciencias (Matemáticas  
Aplicadas e Industriales)**

Asesores:

Dr. José Héctor Morales Bárcenas

Dr. Gabriel Núñez Antonio

Jurado Calificador:

Presidente: Dr. Joaquín Delgado Fernández

Secretario: Dr. José Héctor Morales Bárcenas

Vocal: Dr. Miguel Ángel Moreles Vázquez

Vocal: Dr. Gabriel Núñez Antonio

Iztapalapa, Ciudad de México, agosto 2021

# Índice general

<b>Agradecimientos</b>	<b>3</b>
<b>1. Introducción</b>	<b>4</b>
<b>2. Preliminares</b>	<b>6</b>
2.1. Estadística bayesiana . . . . .	6
2.1.1. Distribución inicial. . . . .	9
2.1.2. Aproximaciones numéricas. . . . .	11
2.1.3. Intervalos de probabilidad . . . . .	14
2.1.4. La distribución predictiva. . . . .	15
2.2. Modelos basados en sistemas de ecuaciones diferenciales . . . . .	16
2.2.1. Soluciones numéricas . . . . .	18
2.3. Datos Circulares . . . . .	18
<b>3. Metodología</b>	<b>23</b>
3.1. Métodos de Monte Carlo vía cadenas de Markov . . . . .	23
3.1.1. El algoritmo Metropolis . . . . .	23
3.1.2. El algoritmo de Metropolis-Hastings . . . . .	24
3.1.3. El algoritmo Metropolis Adaptivo (AM) . . . . .	24
3.1.4. El algoritmo Metropolis Adaptivo Robusto (RAM) . . . . .	25
3.1.5. El algoritmo Monte Carlo Hamiltoniano (HMC) . . . . .	26
3.1.6. El algoritmo Markov Chain Monte Carlo ABC (MCMC ABC) . . . . .	28
<b>4. Aplicaciones</b>	<b>31</b>
4.1. El modelo FH-N . . . . .	31
4.1.1. Método RAM . . . . .	33
4.1.2. Método HMC . . . . .	55
4.1.3. Método ABC . . . . .	75
4.1.4. Comparación . . . . .	100
4.2. Modelo Dinámico de Población . . . . .	103
4.2.1. Método RAM . . . . .	105
4.3. Modelos de Datos Circulares . . . . .	111
4.3.1. Primer análisis . . . . .	112
4.3.2. Segundo análisis . . . . .	119
4.3.3. Tercer análisis . . . . .	126
<b>Conclusiones</b>	<b>134</b>

Bibliografía	136
Apéndice A. Códigos en R	139

# Agradecimientos

*Gracias a mi papá y mamá, por siempre creer en mí y apoyarme en mis decisiones de vida, y siempre poder contar con ellos para lo que necesito; y a mi hermano por hacer la vida siempre un poco menos estresante, y siempre estar ahí cuando lo necesito.*

*A mis asesores, el Dr. Núñez y el Dr. Morales, por nunca dejarme solo mientras trabaja en este proyecto, y guiarme cuando llegaba a callejones sin salida, e inspirarme a continuar mis estudios en el doctorado.*

*A mi abuelo, Daniel, por ser una inspiración para continuar mis estudios.*

*A toda mi demás familia que siempre veían con entusiasmo la continuación de mis estudios.*

*A mis compañeros de maestría, por hacer el tiempo más pasadero, y haber hecho amistades para toda la vida.*

*A la UAM, por haber sido mi segunda casa, y poder tener el privilegio de haber realizado mis estudios de licenciatura y maestría en una de las mejores universidades del país.*

# Capítulo 1

## Introducción

La modelación matemática es el arte de traducir problemas de un área de aplicación en formulaciones matemáticas manejables cuyo análisis teórico y numérico proporciona información, respuestas y orientación útiles para la descripción de fenómenos reales. Este enfoque es el que nos lleva a tener interés muy especial en su correcta aplicación y estudio.

“El propósito principal de la modelación estadística es derivar de las observaciones de un fenómeno aleatorio una inferencia sobre la distribución de probabilidad subyacente a este fenómeno. Es decir, proporciona un análisis (descripción) de un fenómeno pasado, o algunas predicciones sobre un fenómeno futuro de una naturaleza similar.” (Christian P. Robert, 1994)

El celebre estadístico George Box dijo “Todos los modelos están equivocados” el cual es un aforismo común en estadística; a menudo se toma como “Todos los modelos son incorrectos, pero algunos son útiles”. Por lo general, se considera aplicable no solo a modelos estadísticos, sino a modelos en general. El aforismo reconoce que los modelos matemáticos siempre están y estarán fuera del alcance de las complejidades de la realidad, pero aún así pueden ser útiles, y pueden ayudarnos a comprender como funciona el mundo a nuestro alrededor. En su sentido más amplio, la modelación matemática es el proceso de tratar de describir matemáticamente una situación no matemática.

El enfoque de esta tesis es similar al enfoque de lo que conocemos actualmente como ciencia de datos. La ciencia de datos es un campo interdisciplinario que involucra métodos científicos, procesos y sistemas para extraer conocimiento o un mejor entendimiento de la realidad guiada por datos en sus diferentes formas, lo cual comúnmente se puede ver como la unión de varios campos de estudio, como lo son la estadística, la modelación matemática y el cómputo científico.

Como varios de los campos de conocimientos están surgiendo continuamente, la importancia de la ciencia de datos también está aumentando rápidamente. Su efecto se puede observar en múltiples sectores como la industria minorista, la salud, la educación y más sectores de importancia en nuestra vida diaria.

Este trabajo combina varios campos de conocimiento como la estadística bayesiana y los métodos MCMC (Monte Carlo vía Markov Chain), la modelación matemática mediante sistemas de ecuaciones diferenciales (en específico, sistemas de ecuaciones diferenciales no lineales). Así como, el estudio en modelos de datos

circulares, y el cómputo científico mediante el lenguaje R, el cual es un entorno y lenguaje de programación con un enfoque al análisis estadístico.

El objetivo general de esta tesis es el análisis y comparación de algunos métodos numéricos MCMC que se emplean en la inferencia bayesiana, particularmente en el caso donde las soluciones a los sistemas de ecuaciones diferenciales no son analíticas o donde no se cuenta con una verosimilitud que describa la relación entre los parámetros del modelo y la solución del sistema. Así ejemplificamos como la inferencia es guiada por datos de soluciones numéricas de sistemas de ecuaciones diferenciales y modelos de datos circulares, para así poder lograr modelar el comportamiento de fenómenos reales, basados en los datos.

# Capítulo 2

## Preliminares

En este capítulo revisaremos los conceptos básicos del enfoque bayesiano de la estadística. También, presentamos los sistemas de ecuaciones diferenciales utilizados en la modelación determinista. Por último, veremos la importancia de los datos circulares y su modelación.

### 2.1. Estadística bayesiana

La estadística es la rama de las matemáticas que estudia la variabilidad, así como el proceso aleatorio que la genera siguiendo las leyes de la probabilidad. En la estadística clásica se tiende a usar informalmente la probabilidad para expresar información de cantidades desconocidas, sin embargo, este proceso se puede formalizar. Matemáticamente se ha demostrado la relación que guarda la teoría de las probabilidades con la teoría de la información (Shannon, 1948). Es la regla de Bayes que nos provee de un método racional para actualizar nuestras conclusiones con base a nueva información. El proceso inductivo de actualización vía la regla de Bayes se enmarca en lo que se conoce como inferencia bayesiana (Bernardo, J.M. & Smith, A.F.M., 2007).

Los métodos bayesianos son herramientas de análisis de datos que derivan de la inferencia bayesiana. Además de su interpretación formal, los métodos bayesianos también permiten:

- Incorporar la información que el investigador tenga del problema, al análisis formal de inferencia estadística.
- Definir distribuciones de probabilidad para describir el grado de creencia sobre los parámetros de un modelo.
- Un análisis natural de la distribución de la variable de respuesta en un modelo a través de la distribución predictiva.

Uno de los principales teoremas que se usa en la estadística bayesiana es el teorema de Bayes, el cual se describe a continuación:

**Teorema 2.1.1. (de Bayes.)** Si  $A$  y  $E$  son eventos tales que  $P(E) \neq 0$  entonces la probabilidad condicional de  $A$  dado  $E$  se puede expresar de la siguiente forma:

$$P(A|E) = \frac{P(E|A)P(A)}{P(E)}.$$

Este se puede generalizar al caso donde  $(A_1, A_2, \dots, A_k)$  es una partición del espacio de probabilidad  $\Omega$ . Así:

$$P(A_j|B) = \frac{P(B|A_j)P(A_j)}{\sum_{i=1}^k P(B|A_i)P(A_i)}.$$

Con el teorema de Bayes, el proceso bayesiano de análisis de datos puede ser idealizado considerando las siguientes componentes. Supongamos que tenemos un muestra aleatoria  $(y_1, \dots, y_n)$ , entonces:

1. Definimos la *distribución inicial* (Rowe, D.B., 2002) del parámetro  $\theta$  por medio de  $p(\theta)$ , donde es nuestra creencia de cómo se modela el parámetro  $\theta$  según nuestros conocimientos previos. Este modelo jamás depende de la muestra. (también conocido como distribución a priori)
2. Definimos  $p(y|\theta)$  el modelo que creemos que explica de donde se obtiene el valor  $y$  si conociéramos el verdadero valor de  $\theta$ .
3. Con los dos elementos anteriores se obtiene la *distribución final* (o posteriori) de  $\theta$ ,  $p(\theta|y)$ .

La *distribución final* se obtiene mediante el teorema de Bayes:

$$p(\theta|y) = \frac{p(y|\theta)p(\theta)}{p(y)}.$$

Conocemos ambos términos que están multiplicando en el numerador, nuestro problema es calcular la función  $p(y)$ , pero ésta se puede escribir de la siguiente forma:

$$p(y) = \int_{\Theta} p(y|\theta)p(\theta)d\theta.$$

Con lo cual obtendríamos:

$$p(\theta|y) = \frac{p(y|\theta)p(\theta)}{\int_{\Theta} p(y|\theta)p(\theta)d\theta}.$$

Es importante resaltar que la regla de Bayes no nos dice cuáles deben ser nuestras hipótesis antes de hacer inferencia bayesiana, eso depende únicamente del investigador, pero si nos dice cómo deben de cambiar al tener nueva información.

Después de obtener la *distribución final*, también debemos de revisar qué tan bien se describe este modelo a nuestros datos, si tenemos conclusiones razonables y que tan sensibles son nuestros resultados a las suposiciones que hicimos desde el principio. Lo anterior ya que el haber calculado la *distribución final* no implica que el modelo esté correcto o que todo lo que hayamos hecho esté bien, ya que bien



podríamos modificar nuestras suposiciones de las componentes y obtener resultados completamente distintos.

Una equivalencia a la ecuación de la *distribución final* se obtiene omitiendo el factor  $p(y)$ , ya que como no depende de  $\theta$  se puede considerar como un valor constante, con la cual obtenemos:

$$p(\theta|y) \propto p(y|\theta)p(\theta).$$

Lo anterior, ya que este denominador escala la densidad posterior para que sea una densidad adecuada, siendo la densidad de muestreo proporcional a la función de probabilidad. En otras palabras, el teorema de Bayes nos dice que la función de probabilidad para la *distribución final* es proporcional al producto de la distribución inicial y la función de verosimilitud. Así:

$$p(\theta|y) \propto p(y|\theta) \times p(\theta).$$

Afortunadamente, el que la densidad posterior sea solo proporcional al producto de la función de verosimilitud y a la distribución inicial, no es generalmente un problema en el análisis bayesiano. Sin embargo, siempre debemos de estar conscientes del hecho que estamos trabajando con la proporcionalidad. Las funciones de densidad a menudo necesitan tener una constante de normalización incluida para hacer las funciones de densidad adecuadas, pero también debemos recordar que esta constante de normalización solo tiene el efecto de escalar la densidad, no cambiar fundamentalmente las frecuencias relativas de diferentes valores de la variable aleatoria.

La función de verosimilitud juega un rol muy importante en la inferencia bayesiana, ya que es la función a través de la cual los datos  $y$  modifican los supuestos de la distribución inicial, razón por la cual se puede pensar como la representación de información de  $\theta$  de los datos.

Estos simples pero poderosos conceptos enmarcan las técnicas básicas de la inferencia bayesiana, donde el punto de partida que es desarrollar un modelo  $p(\theta|y)$  y poder resumir este modelo en formas apropiadas.

Antes de revisar más el enfoque bayesiano, valdría la pena conocer antes la siguiente definición y teorema:

**Definición 2.1.1.** (*Intercambiabilidad*) Sea  $p(y_1, \dots, y_n)$  la función de densidad conjunta de  $Y_1, \dots, Y_n$ . Si  $p(y_1, \dots, y_n) = p(y_{\pi_1}, \dots, y_{\pi_n})$  para todas las permutaciones  $\pi = (\pi_1, \dots, \pi_n)$  de  $\{1, \dots, n\}$ , entonces  $Y_1, \dots, Y_n$  son intercambiables.

La propiedad de intercambiabilidad está estrechamente relacionada con el uso de variables aleatorias independientes e idénticamente distribuidas (i.i.d.) en modelos estadísticos. Una sucesión de variables aleatorias que son (i.i.d.), condicionadas a alguna forma distributiva subyacente es intercambiable

**Teorema 2.1.2.** (*De Finetti*) Sea  $Y_i \in Y$  para todo  $i \in \{1, 2, \dots\}$ . Suponga que para cualquier  $n$ , nuestro modelo para  $Y_1, \dots, Y_n$  es intercambiable:

$$p(y_1, \dots, y_n) = p(y_{\pi_1}, \dots, y_{\pi_n}).$$

para todas las permutaciones  $\pi$  de  $\{1, \dots, n\}$ , entonces nuestro modelo puede ser escrito como:

$$p(y_1, \dots, y_n) = \int \left( \prod_1^n p(y_i|\theta) \right) p(\theta) d\theta.$$

Para algún parámetro  $\theta$ , alguna distribución inicial  $p(\theta)$  y modelo de muestro  $p(y|\theta)$ .

La idea principal de esta definición y teorema es que si  $Y_1, \dots, Y_n$  son i.i.d y  $\theta \sim p(\theta)$  entonces  $Y_1, \dots, Y_n$  son intercambiables. Si  $Y_1, \dots, Y_n$  son intercambiables y muestreados de una población finita sin reemplazo, entonces se pueden modelar de forma aproximada como condicionalmente i.i.d. (Diaconis y Freedman, 1980). Esto da una mayor flexibilidad con las muestras que trabajaremos con los experimentos computacionales.

### 2.1.1. Distribución inicial.

Una propiedad básica de una función de densidad de probabilidad  $f(x)$  es que su integral sobre todo su dominio vale uno. En muchos casos, hay distribuciones iniciales que no necesariamente integran uno, cuando este es el caso, la distribución inicial se llama: *distribución inicial impropia*. Los estadísticos a veces usan distribuciones iniciales impropias como distribuciones iniciales no informativas.

Distribuciones finales obtenidas de distribuciones iniciales impropias deben de interpretarse con cuidado, siempre se debe verificar que la distribución final tiene una integral finita.

Una de las interpretaciones más básicas que pueden darse a las distribuciones iniciales, es la interpretación de la población, la distribución inicial representa una población de posibles valores de los parámetros, de los cuales se ha extraído el  $\theta$  de interés actual. En el estado más subjetivo de interpretación del conocimiento, el principio es tal que debemos expresar nuestro conocimiento (e incertidumbre) sobre  $\theta$  como si su valor pudiera considerarse como una realización aleatoria de la distribución inicial.

Cuando la información previa sobre el modelo es demasiado vaga o poco confiable, la derivación de la distribución inicial es obviamente imposible. Otras razones (demoras, prohibiciones de costos, falta de comunicación entre el estadístico y el tomador de decisiones, etc.) pueden explicar la ausencia de una distribución inicial bien definida. Además, los requisitos de objetividad pueden forzar al estadístico a proporcionar una respuesta con el menor aporte subjetivo posible, para basar la inferencia solo en el modelo propuesto.

Hay varias formas de clasificar las distribuciones iniciales:

1. **Distribución inicial conjugada.** La familia conjugada es matemáticamente conveniente porque la distribución final sigue una forma paramétrica conocida.

La conjugación se define formalmente de la siguiente manera:

**Definición 2.1.2.** Si  $F$  es una clase de distribuciones de muestreo  $p(y|\theta)$  y  $P$  es una clase de distribuciones iniciales para  $\theta$ , entonces la clase  $P$  es conjugada para  $F$  si la distribución final  $p(\theta|y) \in P$  para toda densidad  $p(y|\theta) \in F$  y distribución inicial  $p(\theta) \in P$ .

El aspecto automatizado de las distribuciones iniciales conjugadas es simultáneamente una ventaja y un problema. Se ha argumentado que este es un enfoque objetivo, donde la entrada subjetiva se reduce a la elección de los hiperparámetros. El hecho es que la objetividad es un concepto difícil de definir por lo que se puede optar, en el mejor de los casos y sin preferencia, por cualquier otra distribución inicial, pero con el mismo número de hiperparámetros. Además, las distribuciones iniciales conjugadas no son necesariamente las distribuciones iniciales más robustas y, a partir de este punto de vista, se podrían preferir distribuciones alternativas, si el objetivo es minimizar la influencia de la elección de la distribución inicial.

El Apéndice 2 del libro de Bernardo, J.M. & Smith, A.F.M. (2007) es un resumen donde al escoger distribuciones iniciales conjugadas que proponen, nos proporcionan la distribución final y la actualización de sus parámetros. Definitivamente, la elección de las distribuciones iniciales conjugadas debe facilitar el cómputo, pero no es garantía que sea lo más adecuado para el problema concreto que estemos resolviendo.

2. **Distribuciones iniciales no informativas.** Se supone que una distribución inicial debe de representar el conocimiento sobre los parámetros antes del resultado de un experimento. El diseño de un experimento es conocido y controlado pero, en el análisis estadístico de los resultados, el problema principal es cómo seleccionar una distribución inicial que proporcione poca información relativa a lo que se espera que proporcione el experimento.

Una distribución inicial localmente uniforme puede considerarse no informativa sobre los parámetros. Es importante tener en cuenta que uno nunca puede estar en un estado de completa ignorancia. Si decimos que tenemos poco conocimiento a priori sobre un parámetro relativo a la información, se espera que sean provistos por los datos. Históricamente, Laplace fue el primero en utilizar técnicas no informativas ya que, aunque no tenía información sobre el número de bolas blancas en el urna o la proporción de nacimientos masculinos, eligió una distribución inicial sobre estos parámetros que tomaba en cuenta su ignorancia y le dio la misma probabilidad a cada valor del parámetro, es decir, usó una distribución inicial uniforme.

3. **Distribución inicial de Jeffreys.** Jeffreys propone un enfoque intrínseco que de hecho obvia la necesidad de tener en cuenta una posible estructura de invariancia. La distribución inicial de Jeffreys no informativa se basan en la información de Fisher dada por la siguiente esperanza matemática:

$$I(\theta) = E_{\theta} \left[ \left( \frac{\partial \log f(x|\theta)}{\partial \theta} \right)^2 \right].$$

En el caso unidimensional, también es conocida como la cota inferior de Cramer-Rao. Bajo condiciones de regularidad, esto se puede escribir de la siguiente manera:

$$I(\theta) = -E_{\theta} \left[ \frac{\partial^2 \log f(x|\theta)}{\partial \theta^2} \right].$$

La distribución inicial de Jeffreys se define como:

$$\pi^*(\theta) \propto I^{1/2}(\theta).$$

Definido bajo una constante de normalización cuando  $\pi^*$  es una distribución inicial propia. En el caso multidimensional, el enfoque no informativo de Jeffreys puede conducir a incoherencias o incluso paradojas y debemos enfatizar que Jeffreys enfatizaba principalmente el uso de estas distribuciones en el caso unidimensional. Sin embargo, su método proporciona una de las mejores técnicas automatizadas para derivar distribuciones iniciales no informativas. Además, a menudo es relacionado con algunos estimadores clásicos (Jeffreys, 1961) y (Robert E. Kass, 1996).

En esta tesis se trabajo principalmente con distribuciones iniciales no informativas, ya que reflejan el caso general donde el investigador desconoce el comportamiento de los parámetros.

### 2.1.2. Aproximaciones numéricas.

Nuestro interés es obtener la distribución final, en donde no podemos omitir el cálculo de la constante de normalización. Específicamente, dado un modelo  $p(y|\theta)$  y una distribución inicial  $p(\theta)$ , el punto de partida para todos los resúmenes de inferencia posteriores es la distribución final:

$$p(\theta|y) = \frac{p(y|\theta)p(\theta)}{\int_{\Omega} p(y|\theta)p(\theta)d\theta}.$$

Para su cálculo tenemos dos opciones, cuando  $p(\theta|y)$  es intratable analíticamente:

1. Integración numérica del denominador
2. Aproximaciones vía muestreo estocástico (métodos de simulación, procesos estocásticos, etc.) como el de aceptación y rechazo, o incluso algoritmos como el Metropolis-Hastings. Con una muestra de la distribución final se pueden encontrar medias u otras cantidades de interés, y aproximaciones a la función de densidad de la distribución final.

En el caso de integración numérica, generalmente se reducen al cálculo de una relación de dos integrales. La clave técnica para la implementación de la solución formal dada por el teorema de Bayes, para la probabilidad especificada y distribución inicial, es la capacidad de realizar una serie de integraciones. Primero, necesitamos evaluar el denominador en el teorema de Bayes para obtener la constante de normalización de la distribución final, entonces necesitamos integrar sobre componentes complementarios de  $\theta$ , o transformaciones de  $\theta$ , para obtener densidades marginales

(univariadas o multivariadas). Generalmente la integración numérica es muy complicada de aplicar puesto que como  $\theta$  puede ser un vector de dimensiones altas, los métodos actuales de integración numérica no son suficientes para resolver el problema, lo cual nos deja con una sola opción, la aproximación vía muestreo estocástico.

Algunos métodos para el muestreo estocástico de la distribución final son:

### 1. Muestreo por importancia.

El enfoque de muestreo por importancia para la integración numérica se basa en la observación que si  $f$  es una función y  $g$  es una función de densidad de probabilidad, entonces:

$$\int f(x)dx = \int \left[ \frac{f(x)}{g(x)} \right] g(x)dx.$$

Esto sugiere que el enfoque estadístico de generar una muestra a partir de la distribución de la función  $g$ - referida en este contexto como la distribución de muestreo por importancia - y usando el promedio de los valores de la relación  $f/g$  como un estimador insesgado de  $\int f(x)dx$ . Sin embargo, la varianza de dicho estimador depende claramente de la elección de  $g$ , por lo que es deseable elegir  $g$  para que sea “similar” a la forma de  $f$ .

### 2. Remuestreo por importancia

En lugar de usar solo el muestreo por importancia para estimar integrales y, por lo tanto, calcular constantes, también podemos explotar la idea para producir muestras simuladas de distribuciones finales o predictivas. Rubin (1988) se refiere a esta técnica como Sampling-Importance-Resampling (SIR).

### 3. Algoritmo de aceptación y rechazo

El muestreo de rechazo es una técnica básica utilizada para generar observaciones a partir de una distribución. También se le llama comúnmente el método de aceptación-rechazo o “algoritmo de aceptación-rechazo” y es un tipo de método de simulación exacto. El método funciona para cualquier distribución en  $\mathbb{R}^m$ .

El muestreo de rechazo se basa en la observación de que para muestrear una variable aleatoria en una dimensión, se puede realizar un muestreo aleatorio uniforme del gráfico cartesiano bidimensional y mantener las muestras en la región debajo del gráfico de su función de densidad. Esta propiedad se puede extender a las funciones de dimensión  $m$ .

Supongamos que queremos simular observaciones de  $f(x)$  función de densidad, con dominio en  $(a, b)$  y  $f(x) \in [0, k] \forall x \in (a, b)$ , entonces el algoritmo sigue los siguientes pasos:

- a) Generar  $X \sim U(a, b)$
- b) Generar  $Y \sim U(0, k)$
- c) Aceptar aquellos valores de  $X$  si  $Y \leq f(x)$

d) Regresar al primer paso

Aunque el método original depende de densidades uniformes, se puede modificar para utilizar otras funciones de densidad  $g(x)$  donde  $f(x) \leq kg(x)$ , esto da a lugar que  $k = \max(f(x)/g(x))$ . El algoritmo resultante es:

a) Generar  $u \sim U(0, 1)$

b) Generar  $X \sim g(x)$

c) Aceptar aquellos valores de  $X$  si  $u \leq f(x)/kg(x)$

d) Regresar al primer paso

Aunque es un método básico, la probabilidad de aceptación en el segundo algoritmo es  $1/k$ . Si  $k$  toma valores grandes tendríamos que simular muchos valores para solo aceptar unos cuantos, lo cual lo vuelve en ocasiones un método muy ineficiente.

#### 4. Métodos de Monte Carlo vía cadenas de Markov

Una de las mayores ventajas del enfoque bayesiano es la flexibilidad de modelado a través de de las cadenas de Markov vía Monte Carlo (MCMC). Ulam y Metrópolis superaron el problema de no conocer la constante de normalización de la distribución final al construir una cadena de Markov para la cual la distribución deseada sea la distribución estacionaria de la cadena de Markov. Entonces solo necesitaban simular la cadena de Markov hasta que se lograra la estacionariedad. Con este fin, introdujeron el algoritmo Metropolis (Hastings, 1970) y (Metropolis, 1987). Posteriormente se generalizó con el algoritmo de Metropolis-Hastings algoritmo (1970) y el muestreo de Gibbs de Geman y Geman (1984).

La idea clave es muy simple. Supongamos que deseamos generar una muestra de un distribución posterior  $p(\theta|x)$  para  $\theta \in \Theta \subset \mathbb{R}^k$  pero no se puede hacer directamente. Sin embargo, supongamos que podemos construir una cadena de Markov con espacio de estado  $\Theta$ , simple de simular y cuya distribución de equilibrio es  $p(\theta|x)$ . Si entonces podemos ejecutar la cadena durante mucho tiempo, los valores simulados de la cadena se pueden utilizar como base para resumir las características de la distribución final de interés, esto gracias a la convergencia de la distribución de equilibrio. Para implementar esta estrategia, simplemente necesitamos algoritmos para construir cadenas con distribuciones de equilibrio específico.

En condiciones de regularidad adecuadas, existen resultados asintóticos que aclaran el sentido en el que la muestra generada de una cadena con distribución de equilibrio  $p(\theta|x)$ , y se puede utilizar para imitar una muestra aleatoria de  $p(\theta|x)$  o para estimar el valor esperado, con respecto a  $p(\theta|x)$ , de una función  $g(x)$  de interés.

El método de Monte Carlo, se basa en un muestreo aleatorio y su implementación no requiere un profundo conocimiento de cálculo o análisis numérico. Sea  $\theta$  un parámetro de interés y sean  $y_1, \dots, y_n$  valores numéricos de una muestra

de una distribución  $p(y|\theta)$ . Supongamos que muestreáramos algunos  $S$  valores de  $\theta$  independientes y aleatorios de la distribución final  $p(\theta|y_1, \dots, y_n)$ :

$$\theta^{(1)}, \dots, \theta^{(S)} \sim i.i.d. p(\theta|y_1, \dots, y_n).$$

Luego, la distribución empírica de las muestras  $\{\theta^{(1)}, \dots, \theta^{(S)}\}$  aproximaría a  $p(\theta|y_1, \dots, y_n)$ , con la aproximación mejorando con el aumento de  $S$ . La distribución empírica  $\{\theta^{(1)}, \dots, \theta^{(S)}\}$  es conocida como una aproximación de Monte Carlo a  $p(\theta|y_1, \dots, y_n)$ .

Algunos de estos métodos son:

- a) The Gibbs Sampling Algorithm
- b) The Metropolis-Hastings algorithm
- c) The Robust Adaptive Metropolis (RAM)
- d) The Hamiltonian Monte-Carlo Algorithm (HMC)
- e) Approximate Bayesian Computation (ABC)

En esta tesis se trabajaran con los últimos tres métodos.

### 2.1.3. Intervalos de probabilidad

A veces es necesario dar además regiones de valores de  $\theta$ ; es decir, subconjuntos  $D$  del espacio de parámetros  $\Theta$  donde  $\theta$  debe estar con alta probabilidad.

La formulación bayesiana donde  $\theta$  tiene una probabilidad dada de pertenecer a una región fija  $D$  es más atractivo que la interpretación frecuentista donde una región aleatoria  $D$  tiene una probabilidad de contener el parámetro desconocido  $\theta$ . Esta es una más de las ventajas que ofrece la estadística bayesiana sobre la estadística clásica. El paradigma bayesiano propone una noción de regiones de probabilidad que es más natural que su contraparte frecuentista, la notación  $P(\theta \in D)$  es significativa.

**Definición 2.1.3.** Para una distribución  $p$ , se dice que un región  $D$  es  $\alpha$ -creíble si:

$$P(\theta \in D|x) = 1 - \alpha.$$

Esta región es llamada un HPD  $\alpha$ -creíble (*Highest posterior density, región de máxima probabilidad, denotada por  $D^*$* ) si cumple las siguientes condiciones:

- $P(\theta \in D^*) = 1 - \alpha$ .
- Si  $\theta_1 \in D^*$  y  $\theta_2 \notin D^*$  entonces  $p(\theta_1|y) \geq p(\theta_2|y)$ .

Se prueba que de todas la regiones de tamaño  $1 - \alpha$ ,  $D^*$  es la de menor volumen.

Notemos que cuando usamos  $P(\theta \in D|x) = 1 - \alpha$ , se refiere a la distribución final, es decir, utilizamos la distribución final para encontrar los intervalos de credibilidad, pero la segunda parte de la definición asegura que esta sea óptima, en el sentido de que al considerar solo las regiones HPD se minimiza la longitud de los intervalos de credibilidad  $\alpha$  y, por lo tanto, se pueden obtener soluciones óptimas en una toma de decisiones.

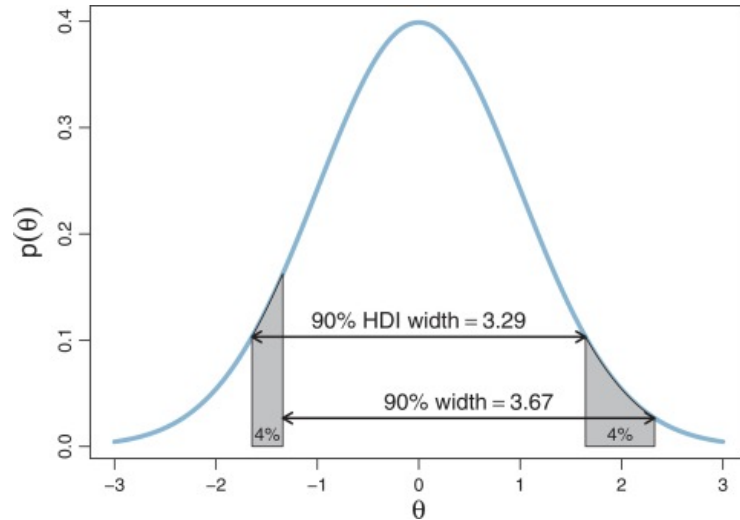


Figura 2.1: En esta figura se muestra la diferencia entre dos intervalos de credibilidad al 90 %. Sin embargo solo el de menor longitud es la región de máxima probabilidad; pues minimiza la longitud del intervalo de credibilidad.

#### 2.1.4. La distribución predictiva.

Una característica importante de la inferencia bayesiana es la existencia de una distribución de predicción para nuevas observaciones. Como el objetivo es hacer *predicciones* entonces se sigue el mismo razonamiento que se empleó para la distribución final, podemos observar que la función de distribución de  $y$  dado una muestra aleatoria  $D_n = (y_1, \dots, y_n)$  sería:

$$p(y|D_n) = \int_{\Omega} f(y|\theta)p(\theta|D_n)d\theta.$$

Donde:

- $f(y|\theta)$  es la función de densidad de nuestra muestra.
- $p(\theta|D_n)$  es la distribución final.

A esta función se le conoce como *distribución predictiva final*. En muchas situaciones de inferencia bayesiana, podremos tomar muestras de  $p(\theta|y_1, \dots, y_n)$  y  $p(y|\theta)$ , pero para  $p(y|D_n)$  será demasiado complicado tomar muestras directamente. En esta situación, podemos generar muestras de la distribución predictiva final igual que se hizo con la distribución final, indirectamente usando procedimientos de Monte Carlo.

El uso de los métodos de Monte Carlo está muy extendido en estadística y modelación matemática en general. Rubinstein y Kroese (2008) explican los métodos de Monte Carlo para una amplia variedad de problemas estadísticos, y Robert y Casella (2004) muestran más aplicaciones bayesianas.

Guttman (1967) y Rubin (1984) sugirieron utilizar la distribución predictiva final para evaluar el ajuste del modelo, y ahora es una práctica común. En algunos problemas es útil evaluar la bondad de ajuste utilizando funciones que dependerán



de los parámetros y de los datos pronosticados. Esto se discute en Gelman et al. (1996) y más recientemente en Johnson (2007)

## 2.2. Modelos basados en sistemas de ecuaciones diferenciales

Desde hace siglos, la modelación matemática ha tenido como objetivo participar de la generación de nuevo conocimiento, por lo que ha sido una parte esencial del método científico. Por otra parte, las leyes de la física están basadas en experimentos u observaciones que generalizan y abstraen rasgos o patrones de los fenómenos naturales. Bajo condiciones particulares dichas leyes adoptan la forma de modelos matemáticos concretos y, comúnmente, si el proceso es continuo, se expresan como ecuaciones diferenciales, lo cual es un legado del pensamiento científico del Siglo XVII con Newton y Leibniz como sus mejores representantes. Actualmente, la modelación es intrínsecamente multidisciplinaria ya que, prácticamente, se desarrolla en todas las ingenierías, la física, la biología, la química, la economía, la misma estadística y las matemáticas y en otras ciencias. Las ecuaciones diferenciales son de los modelos matemáticos más usados y estudiados en las aplicaciones. Dichas ecuaciones son modelos *locales* en el sentido de que expresan la dependencia de la variable de estado y sus derivadas en un punto dado. Otra característica típica de las leyes así expresadas, es que son *causales*: las condiciones posteriores dependen de las previas. La localidad y causalidad son características asociadas a lo que se conoce como *problemas directos*. Por otro lado, los *problemas inversos* frecuentemente son problemas *no locales* y *no causales*, lo cual contribuye a su inestabilidad en la búsqueda de solución (J. Kaipio and E. Somersalo, 2005).

En términos de las ecuaciones diferenciales, un típico problema a resolver es el de *valores iniciales*. El problema directo es relativamente estable y se puede alcanzar suficiente precisión en las predicciones de modelo, basta con tener información suficiente y precisa de las condiciones iniciales y de los valores paramétricos del sistema. En cambio, en el problema inverso, para determinar las condiciones iniciales (u otros parámetros) del sistema dado un muestreo de la solución del mismo sistema o datos, nos puede conducir a situaciones no causales: puede que haya varias condiciones iniciales que nos puedan reproducir las observaciones o datos. Esto es un problema serio que requiere un análisis cuidadoso de la información (datos) que tenemos a nuestra disposición. Precisamente, el esquema de la *teoría de inversión bayesiana* nos resuelve de forma sistemática el problema inverso, de tal forma que toda la información disponible la incorporaremos apropiadamente en los modelos. Justamente esto último es el tema central de la tesis y se desarrolla en los Capítulos 3 y 4.

En la literatura se reportan, de forma artificial pero útil, dos tipos de modelación matemática:

1. La *modelación determinista* implica un modelo donde no hay eventos aleatorios, esto quiere decir, que al ingresar los mismos datos de entrada el resultado va a ser siempre el mismo.

2. La *modelación estocástica* implica un modelo en el que los eventos aleatorios se toman en cuenta, esto quiere decir, que al ingresar los mismos datos de entrada el resultado va a ser diferente.

En nuestro trabajo de tesis vamos a considerar sistemas de ecuaciones diferenciales ordinarias de primer orden y deterministas, cuya única variable independiente será el tiempo  $t$ . Las ecuaciones diferenciales son modelos adecuados para describir tasas de cambio de variables de estado. Las ecuaciones incorporan relaciones mecanicistas entre las variables de estado y están sujetas a condiciones iniciales y dependen de parámetros de interés.

En general, se tiene un sistema de dimensión  $n$  donde:

$$\begin{aligned}x'_1 &= f_1(t, x_1, \dots, x_n) \\x'_2 &= f_2(t, x_1, \dots, x_n) \\&\vdots \\x'_n &= f_n(t, x_1, \dots, x_n)\end{aligned}$$

Así, usando notación vectorial de la siguiente manera:

$$x = (x_1, \dots, x_n)^T$$

y

$$f(t, x(t)) = (f_1(t, x), \dots, f_n(t, x))^T$$

Entonces, la forma general de un sistema de ecuaciones y de primer orden se puede expresar como:

$$x' = f(t, x(t))$$

El *orden* de un sistema de ecuaciones diferenciales es el orden de la mayor derivada que aparece en el sistema.

El problema de valores iniciales para un sistema de dimensión  $n$

$$x' = f(t, x(t))$$

consiste en encontrar  $n$  funciones  $x(t) = (x_1(t), x_2(t), \dots, x_n(t))^T$  que sean solución del sistema y cumplan una determinada condición inicial  $x(t_0) = x_0$ . El problema de condiciones iniciales lo escribiremos de forma similar a como lo hacíamos para el sistema de ecuaciones:

$$\begin{cases}x' = f(t, x(t)) \\x(t_0) = x_0\end{cases}$$

Si cada una de las funciones  $f_j$  son lineales, entonces el sistema se llama *lineal*, y existe mucha literatura sobre como resolver este tipo de sistemas.

En esta tesis en específico trabajamos con sistemas de ecuaciones diferenciales ordinarias no lineales, aunque la metodología puede ser utilizada para sistemas de ecuaciones diferenciales ordinarias lineales.

### 2.2.1. Soluciones numéricas

Resolver sistemas de ecuaciones no lineales aparece frecuentemente en la modelación matemática. El problema de la existencia y la unicidad de soluciones de este tipo de problemas depende fundamentalmente de las propiedades de la función  $f(t, x)$ . En particular, se requiere que  $f$  sea continua respecto a  $t$  y Lipschitz continua respecto a  $x$ .

Los métodos numéricos son procedimientos utilizados para encontrar aproximaciones numéricas a las soluciones de ecuaciones diferenciales ordinarias (EDO). Su uso también se conoce como integración numérica, aunque este término a veces se toma para significar el cálculo de una integración. Muchas ecuaciones diferenciales no pueden resolverse analíticamente, sin embargo, una aproximación numérica a la solución suele ser suficiente para la modelación matemática.

Los siguientes algoritmos pueden usarse para calcular tal aproximación:

1. Método de Euler
2. Método de Euler modificado (o mejorado)
3. Método de Heun (o del trapecio)
4. Método RK2 óptimo
5. Método de Runge-Kutta clásico o RK4

Particularmente, el integrador predeterminado para la solución numérica de sistemas de ecuaciones diferenciales utilizado por el paquete “desolve” de R es: “lsoda”. La función R lsoda proporciona una interfaz para el solucionador FORTRAN ODE del mismo nombre, escrito por Linda R. Petzold y Alan C. Hindmars (2003). El trabajo lo realiza la subrutina lsoda de FORTRAN, cuya documentación se debe consultar para obtener más detalles. lsoda se diferencia de los otros integradores (excepto lsodar) en que cambia automáticamente entre métodos rígidos (stiff en inglés) y no rígidos. Esto significa que el usuario no tiene que determinar si el problema es rígido o no, y el solucionador elegirá automáticamente el método apropiado. Siempre empieza con el método no rígido.

En matemáticas, un sistema rígido es una expresión que está mal condicionada y, por lo tanto, para resolverla los métodos numéricos son inestables para la cual ciertos métodos numéricos para resolver la ecuación son numéricamente inestables, a menos que se considere que el tamaño del paso es extremadamente pequeño. La idea principal de la rigidez en las ecuaciones se centra en algunos términos que pueden conducir a una rápida variación en la solución (Brenan, K. E., 1996).

## 2.3. Datos Circulares

Los datos circulares pueden surgir de varios enfoques científicos, hasta la biología y geología, como el movimiento del viento, dirección del movimiento del océano o animales e incluso direcciones de salida; es por eso que su análisis y estudio es

importante. La naturaleza especial de los datos circulares implica que no se pueden aplicar los métodos convencionales adecuados para el análisis de datos lineales.

En particular para empezar a trabajar con los datos circulares, demostraremos las siguientes dos proposiciones:

**Proposición 2.3.1.** *Sea  $Y$  un vector aleatorio con distribución  $N_2(\mu, I)$ , si se define la transformación:*

$$y = r(\cos \theta, \text{sen } \theta)' = rv'$$

donde  $\theta \in (0, 2\pi]$  y  $r \in \mathbb{R}^+$ . Entonces, la función de densidad conjunta de la transformación  $(r, \theta)$ , esta dada por:

$$f(r, \theta | \mu, I) = r C_6 d^2 \exp(-\frac{1}{2}d^2r^2 + d^2br)$$

donde  $d^2 = v'v = 1$ ,  $b = \frac{v'\mu}{v'v} = v'\mu$  (pues  $v'v = \cos^2 \theta + \text{sen}^2 \theta = 1$ ) y:

$$C_6 = \frac{\exp(-\frac{1}{2}\mu'\mu)}{2\pi v'v}$$

Utilizaremos el teorema de cambio de variable a continuación para la proposición anterior:

*Demostración.* Sea la transformación de coordenadas polares  $g(r, \theta) = (r \cos \theta, r \text{sen } \theta)'$ , su jacobiano es:

$$J(g(r, \theta)) = \begin{vmatrix} \cos \theta & -r \text{sen } \theta \\ \text{sen } \theta & r \cos \theta \end{vmatrix} = r \cos^2 \theta + r \text{sen}^2 \theta = r$$

Por lo tanto:

$$\int_{\mathbb{R}^2} f(x | \mu, I) = \int_{\mathbb{R}^+ \times [0, 2\pi)} r f(r \cos \theta, r \text{sen } \theta | \mu, I) = \int_{\mathbb{R}^+ \times [0, 2\pi)} f(r, \theta | \mu, I)$$

Entonces sea  $rv = r(\text{Cos}\theta, \text{Sen}\theta)'$ :

$$\begin{aligned} f(r, \theta | \mu, I) &= r \frac{1}{(2\pi)} \exp\left(-\frac{1}{2}(rv - \mu)'I(rv - \mu)\right) \\ f(r, \theta | \mu, I) &= r \frac{1}{(2\pi)} \exp\left(-\frac{1}{2}(rv'Irv - 2rv'I\mu + \mu'I\mu)\right) \\ f(r, \theta | \mu, I) &= \frac{r}{2\pi} \exp\left(-\frac{1}{2}r^2v'v + rv'\mu - \frac{1}{2}\mu'\mu\right) = \frac{r}{2\pi} \exp\left(-\frac{1}{2}r^2 + rv'\mu - \frac{1}{2}\mu'\mu\right). \end{aligned}$$

Ahora vamos a sustituir los valores que tenemos en el resultado, para observar que tienen el mismo valor:

$$\begin{aligned}
f(r, \theta | \mu, \Lambda) &= r C_6 \exp\left(-\frac{1}{2}r^2 + br\right) \\
&= r \frac{\exp(-\frac{1}{2}\mu'\mu)}{2\pi} \exp\left(-\frac{1}{2}r^2 + rv'\mu\right) \\
&= \frac{r}{2\pi} \exp\left(-\frac{1}{2}r^2 + rv'\mu - \frac{1}{2}\mu'\mu\right).
\end{aligned}$$

Que es el mismo resultado que obtuvimos en la primera parte de la demostración, con lo cual obtenemos que:

$$f(r, \theta | \mu, I) = r C_6 \exp\left(-\frac{1}{2}r^2 + br\right).$$

Por lo tanto la proposición queda demostrada, pero una forma más sencilla de observar la función de densidad sería:

$$f(r, \theta | \mu, I) = \frac{r}{2\pi} \exp\left(-\frac{1}{2}r^2 + rv'\mu - \frac{1}{2}\mu'\mu\right).$$

□

**Proposición 2.3.2.** *Bajo las mismas condiciones de la Proposición (2.3.1), la función de densidad del ángulo aleatorio  $\Theta$ , esta dada por:*

$$NP(\theta | \mu, I) = \int_0^\infty f(r, \theta | \mu, I) dr = C_6 \left[1 + \frac{db}{\phi(db)} \Phi(db)\right] I_{(0, 2\pi)}(\theta)$$

donde  $d$ ,  $b$  y  $C_6$  son como en la proposición anterior.  $\Phi(\cdot)$  y  $\phi(\cdot)$  denotan las funciones de distribución y de densidad de una normal estándar, respectivamente.

A la densidad  $NP(\theta | \mu, I)$  se le conoce como distribución normal proyectada.

*Demostración.* Como se vio en la proposición anterior, como  $d^2 = 1$  entonces, sea:

$$\phi(b) = \frac{1}{\sqrt{2\pi}} e^{-\frac{b^2}{2}}$$

y también

$$\Phi(b) = \frac{1}{\sqrt{2\pi}} \int_b^\infty e^{-\frac{u^2}{2}} du$$

entonces:

$$NP(\theta | \mu, I) = \int_0^\infty f(r, \theta | \mu, I) dr$$

Por medio de la integral por partes, tenemos  $u = \exp\{br\}$  y  $dv = r \exp\{-\frac{1}{2}r^2\}$ , y dejando el termino  $C_6$  afuera, tenemos:

$$\begin{aligned}
NP(\theta|\mu, I)/C_6 &= uv - \int_0^\infty v du \\
&= -\exp\left\{br - \frac{1}{2}r^2\right\}_0^\infty + \int_0^\infty b \exp\left\{-\frac{1}{2}r^2 + br\right\} \\
&= 1 + \int_0^\infty b \exp\left\{-\frac{1}{2}r^2 + br\right\}.
\end{aligned}$$

Completando el binomio cuadrado en el exponencial de la integral, obtenemos:

$$= 1 + \int_0^\infty b \exp\left\{-\frac{(r-b)^2}{2} + \frac{b^2}{2}\right\}.$$

Y esto se puede escribir de la siguiente forma:

$$= 1 + b\sqrt{2\pi} \exp\left\{\frac{b^2}{2}\right\} \int_b^\infty \frac{1}{\sqrt{2\pi}} \exp\left\{-\frac{u^2}{2}\right\} du = 1 + b\frac{\Phi(b)}{\phi(b)}.$$

Por lo tanto:

$$NP(\theta|\mu, I) = C_6 \left[1 + \frac{b}{\phi(b)}\Phi(b)\right] I_{(0,2\pi)}(\theta).$$

□

Una vez obtenida la distribución de la normal proyectada  $NP(\theta|\mu, I)$  podemos realizar inferencia bayesiana para los datos circulares de la forma  $\theta \in (0, 2\pi]$  usando métodos de simulación estocástica.

**Proposición 2.3.3.** *Bajo las mismas condiciones de las proposiciones anteriores, la función de densidad condicional de  $R$  dado  $\Theta$  esta dada por:*

$$f(r|\theta, \mu, I) = \frac{r \exp(-\frac{1}{2}[r^2 - 2br])}{1 + \frac{b}{\phi(b)}\Phi(b)} I_{(0,\infty)}(r)$$

*Demostración.* Se sigue el cálculo de las probabilidades condicionales:

$$P(A|B) = \frac{P(A,B)}{P(B)}$$

entonces:

$$f(r|\theta, \mu, I) = \frac{f(r, \theta|\mu, I)}{NP(\theta|\mu, I)}$$

por lo que:

$$f(r|\theta, \mu, I) = \frac{r C_6 \exp\left(-\frac{1}{2}r^2 + br\right)}{C_6\left[1 + \frac{b}{\phi(b)}\Phi(b)\right]} = \frac{r \exp\left(-\frac{1}{2}[r^2 - 2br]\right)}{1 + \frac{b}{\phi(b)}\Phi(b)}.$$

Pero recordando que  $d^2 = 1$ , entonces tenemos:

$$f(r|\theta, \mu, I) = \frac{r \exp\left(-\frac{1}{2}[r^2 - 2br]\right)}{1 + \frac{b}{\phi(b)}\Phi(b)}.$$

Donde  $b = v'\mu$ , por lo tanto la proposición queda demostrada. □

# Capítulo 3

## Metodología

En este capítulo revisaremos varios algoritmos MCMC que existen en la literatura, en particular revisaremos los tres métodos MCMC de interés para esta tesis, los cuales son: RAM, HMC y ABC.

### 3.1. Métodos de Monte Carlo vía cadenas de Markov

Como ya se menciona en el Capítulo de preliminares, los métodos MCMC tienen varias ventajas para la inferencia estadística bayesiana. Sin embargo, existe una gran variedad de algoritmos, cada uno con sus propias ventajas y desventajas, por lo cual debemos de tener cuidado en decidir que método es más apropiado para cada problema. Sea nuestra distribución final el producto de la función de verosimilitud por la distribución inicial:

$$p(\theta|y) \propto p(y|\theta) \times p(\theta)$$

#### 3.1.1. El algoritmo Metropolis

Sea  $p(\theta|y)$  una densidad objetivo el algoritmo Metropolis requiere la elección de una densidad propuesta  $q$  también llamada densidad de transición. La transición del valor de la cadena de Markov  $\theta(t)$ , en el momento  $t$  y su valor en el tiempo  $t + 1$ , procede mediante el siguiente paso de transición:

---

**Algoritmo 1:** Metropolis

---

Dado que  $\theta(t) = \theta_t$ ,

Generamos  $\theta^* \sim q(\theta|\theta_t)$ ,

Sea:

$$\alpha = \min \left( 1, \frac{p(\theta^*|y)}{p(\theta_t|y)} \right),$$

Entonces:

$$\theta(t+1) = \begin{cases} \theta^* & \text{con probabilidad } \alpha \\ \theta_t & \text{con probabilidad } 1 - \alpha \end{cases} .$$

---



Desde hace mucho tiempo se reconoce que la elección de la densidad  $q$  es crucial para el éxito (por ejemplo, rápida convergencia) del algoritmo de Metropolis, claro la mejor elección sería la distribución objetivo pero está implícito que no podemos usar esta distribución. El caso más común es el algoritmo de Metropolis simétrico de caminata aleatoria (Random Metropolis Walking, RMW) en el que se da el valor de la propuesta por  $\theta_{n+1} = \theta_n + Z_{n+1}$ , donde los incrementos  $Z_n$  son i.i.d. de alguna distribución simétrica fija (normalmente  $N(0, \sigma^2)$ ), pero esto puede arrojar problemas, puesto que si  $\sigma$  es muy pequeño la convergencia tardaría demasiado e implicaría un mayor costo computacional, y si  $\sigma$  es muy grande, los valores propuestos serán rechazados.

### 3.1.2. El algoritmo de Metropolis-Hastings

La principal diferencia entre el algoritmo Metropolis y el algoritmo Metropolis-Hastings (Hoff, Peter D., 2009) es que el algoritmo Metropolis-Hastings asociado a una densidad objetivo  $p(\theta|y)$ , es que no tiene el requisito de que la distribución propuesta  $q$  sea simétrica, lo cual implica que el cociente adicional en el cálculo de  $\alpha$ , corrige cualquier asimetría de la distribución propuesta  $q$ . La transición del valor de la cadena de Markov  $\theta(t)$ , en el momento  $t$  y su valor en el tiempo  $t + 1$ , procede mediante el siguiente paso de transición:

---

**Algoritmo 2:** Metropolis-Hastings

---

Dado que  $\theta(t) = \theta_t$ ,

Generamos  $\theta^* \sim q(\theta|\theta_t)$ ,

Sea:

$$\alpha = \min \left( 1, \frac{p(\theta^*|y) q(\theta_t|\theta^*)}{p(\theta_t|y) q(\theta^*|\theta_t)} \right),$$

Entonces:

$$\theta(t+1) = \begin{cases} \theta^* & \text{con probabilidad } \alpha \\ \theta_t & \text{con probabilidad } 1 - \alpha \end{cases} .$$


---

Se puede observar que el algoritmo Metropolis es un caso especial del algoritmo Metropolis-Hastings.

### 3.1.3. El algoritmo Metropolis Adaptivo (AM)

Algoritmos MCMC como el algoritmo Metropolis-Hastings (Hoff, Peter D., 2009) se utilizan muy ampliamente en la inferencia estadística, es conocido que la elección de la distribución propuesta  $q$  es crucial para lograr una mezcla eficiente y evitar bajas tasas de aceptación. Los algoritmos adaptativos de MCMC intentan solucionar este problema automáticamente aprendiendo los mejores valores de los parámetros de los algoritmos mientras se ejecutan.

Suponga  $P_\gamma$  es una familia de cadenas de Markov, cada una con distribución estacionaria  $p$ . Un algoritmo MCMC adaptativo actualiza aleatoriamente el valor de  $\gamma$  en cada iteración, en un intento por encontrar el mejor valor. Los métodos MCMC adaptativos se han aplicado en un gran variedad de contextos (por ejemplo, Haario et al., 2001; Roberts y Rosenthal, 2006; Giordani y Kohn, 2006) incluyendo problemas

en genética estadística (Turro et al., 2007). Contrariamente a la intuición, es posible que los algoritmos MCMC adaptativos no siempre preserven la estacionariedad de  $p$  (Andrieu & Moulines, 2006). Sin embargo, si las adaptaciones están diseñadas para satisfacer ciertas condiciones, entonces la estacionariedad está garantizado y son posibles aceleraciones significativas (Gilks et al., 1998). Roberts y Rosenthal (2005) demostraron la ergodicidad de los MCMC adaptativos en condiciones que son simples de aplicar (condición de adaptación decreciente y condición de convergencia acotada).

El primer uso moderno importante de un MCMC adaptativo fue el Metropolis Adaptativo (AM) algoritmo de Haario et al. (2001). Este algoritmo está motivado por la observación de que, para un Metropolis de Caminata Aleatoria (RWM) en  $\mathbb{R}^d$ , al menos con distribuciones normales, es óptimo tener como propuesta una matriz de covarianza de la forma  $(2.38)^2/d$  multiplicado por la matriz de covarianza objetivo  $\Sigma$ . Ya que  $\Sigma$  es desconocida generalmente, se estima por  $\Sigma_n$ , la matriz de covarianza empírica de  $X_0, \dots, X_n$ . Por lo tanto, el algoritmo AM utiliza una distribución de transición para la  $n$ -ésima iteración dada por:

$$Y_{n+1} \sim N(X_n, ((2.38)^2/d)\Sigma_n).$$

Para asegurarse de que las covarianzas de la propuesta no tiendan a 0 Haario et al. (2001) añadió  $I_d$  en cada iteración  $\Sigma_n$ , para cotas relativamente pequeñas  $\epsilon > 0$ . Otra posibilidad (Roberts y Rosenthal, 2006) es dejar que la propuesta sea una distribución mixta de la forma:

$$(1 - \beta)N(X_n, ((2.38)^2/d)\Sigma_n) + \beta N(X_n, ((0.1)^2/d)I_d),$$

para algún  $0 < \beta < 1$  (Se debe de señalar que es necesario utilizar alguna distribución alternativa de propuesta fija para las primeras iteraciones cuando la covarianza empírica aún no está bien definida). Notemos que con estas distribuciones propuestas  $q$ , el algoritmo para la obtención de una muestra de nuestra distribución objetivo es el mismo al algoritmo de Metropolis.

### 3.1.4. El algoritmo Metropolis Adaptivo Robusto (RAM)

El algoritmo, llamado aquí Metropolis Adaptivo Robusto (RAM) de Vihola, M. (2012), es un método que procura evitar el uso de la covarianza empírica. El objetivo es buscar un factor (matriz)  $S$  que capture la forma de la función objetivo  $p(\theta|y)$  y, al mismo tiempo, permitir alcanzar una tasa de aceptación media determinada. A diferencia del método ASWAM (Adaptive Scaling within Adaptive Metropolis), este algoritmo se basa en una sola fórmula para actualizar la matriz que es computacionalmente equivalente a la actualización del factor de covarianza en el algoritmo AM. En el RAM  $q(x)$  es una distribución de propuesta simétrica  $d$ -dimensional, donde  $d$  es la longitud del vector  $\theta$ , es decir, el número de parámetros.

---

**Algoritmo 3:** Robust Adaptive Metropolis

---

Dado que  $\theta(t) = \theta_t$ ,

Generamos  $\theta^* = \theta_t + S_{n-1}u_n$  donde  $u_n \sim q$  es un vector independiente, normalmente  $q = N(0, I)$ ,

Sea:

$$\alpha = \min \left( 1, \frac{p(\theta^*|y)}{p(\theta_t|y)} \right),$$

Entonces:

$$\theta(t+1) = \begin{cases} \theta^* & \text{con probabilidad } \alpha \\ \theta_t & \text{con probabilidad } 1 - \alpha \end{cases}$$

Se actualiza  $S_n$  tal que:

$$S_n S_n^T = S_{n-1} \left( I + \eta_n (\alpha_n - \alpha^*) \frac{u_n u_n^T}{\|u_n\|^2} \right) S_{n-1}^T$$

---

La reciente teoría de escalamiento óptimo ha producido una condición para que la tasa de aceptación asintóticamente óptima de los algoritmos de Metropolis es el conocido 0.234 cuando se aplica a ciertas distribuciones de destino multidimensionales, por lo cual  $\alpha^* = 0.234$ . Y suponga que  $\{\eta_n\}_{n \geq 1} \subset (0, 1]$  es una secuencia de tamaño de paso que decae a cero. La matriz  $S$  para el primer paso suele ser la identidad  $I$ , también la función  $\{\eta_n\}$  suele ser  $\{\eta_n\} = \min(1, dn^{-2/3})$ . El factor extra  $d$  se agregó para compensar el crecimiento esperado o la contracción de los valores propios siendo del orden  $d^{-1}$  (Cody McRae, 2019) y (Matti Vihola, 2011). En R la librería `ramcmc` realiza el paso de actualización de la matriz  $S$  sin necesidad de programarlo, pudiendo modificar el valor de  $\alpha^*$  así como el valor  $2/3$  de la función  $\{\eta_n\}$ .

### 3.1.5. El algoritmo Monte Carlo Hamiltoniano (HMC)

Suponga que  $p(\theta|y)$  es una función de densidad final unidimensional, y  $-\log p(\theta|y)$  supone la forma de una curva en forma de campana inversa, para generar muestras eficientes, es necesario muestrear en la región correspondiente a los valores mínimos de  $-\log p(\theta|y)$ ; y la región se puede alcanzar con la guía del gradiente de  $-\log p(\theta|y)$  (Samuel Thomas and Wanzhu Tu, 2020).

#### Ecuaciones Hamiltonianas

Las funciones hamiltonianas son funciones de la forma  $H(\theta, w)$  las cuales consisten de dos partes tal que:  $H(\theta, w) = U(\theta) + K(w)$  donde  $w, \theta \in \mathbb{R}^k$ . En general, las funciones  $U$  y  $K$  son funciones cuadráticas asociadas con las energías mecánicas potencial y cinética, respectivamente. Este hecho hace atractivo utilizar la estructura hamiltoniana para minimizar un funcional, como mostraremos (Steve Brooks et al., 2011).

Sea  $p(\theta|y)$  la función de distribución final de interés, entonces podemos definir  $U(\theta) := -\ln(p(\theta|y))$ . Ahora sea  $w \sim N_k(0, M)$  con  $M$  matriz de varianza-covarianza entonces:

$$H(\theta, w) = -\ln(p(\theta|y)) + w^T M^{-1}w.$$

Las trayectorias hamiltonianas están gobernadas por el siguiente sistema de ecuaciones diferenciales:

$$\begin{aligned}\frac{dw}{dt} &= \nabla_{\theta} \ln(p(\theta|y)) \\ \frac{d\theta}{dt} &= M^{-1}w.\end{aligned}$$

Donde  $\nabla_{\theta} \ln(p(\theta|y))$  es el gradiente del logaritmo de la distribución final. La solución a estas ecuaciones Hamiltonianas define el camino de dónde se puede muestrear valores de  $\theta$ . La aleatoriedad del algoritmo viene de  $w \sim N_k(0, M)$ . El método de salto es una buena alternativa al método estándar de Euler para aproximar las soluciones numéricas a las ecuaciones hamiltonianas, y se resuelve con un salto de tamaño  $\epsilon$ , de la siguiente manera:

$$\begin{aligned}w\left(t + \frac{\epsilon}{2}\right) &= w(t) + (\epsilon/2)\nabla_{\theta} \ln(p(\theta(t)|y)), \\ \theta(t + \epsilon) &= \theta(t) + \epsilon M^{-1}w\left(t + \frac{\epsilon}{2}\right), \\ w(t + \epsilon) &= w\left(t + \frac{\epsilon}{2}\right) + (\epsilon/2)\nabla_{\theta} \ln(p(\theta(t + \epsilon)|y)).\end{aligned}$$

el algoritmo resultante es:

---

**Algoritmo 4:** Hamiltonian Monte Carlo

---

Considerando que  $\theta(t) = \theta_t$ ,

Generamos  $w_t \sim N(0, M)$ ,

Resolvemos el mediante el método de salto,  $L$  veces con paso  $\epsilon$  el sistema de ecuaciones hamiltonianas, generando así  $\theta_{t+1}$  y  $w_{t+1}$ ,

Sea:

$$\alpha = \min\left(1, \frac{\exp(\ln(p(\theta_{t+1}|y)) - (1/2)w_{t+1}^T M w_{t+1})}{\exp(\ln(p(\theta_t|y)) - (1/2)w_t^T M w_t)}\right),$$

Entonces:

$$\begin{aligned}\theta(t + 1) &= \begin{cases} \theta_{t+1} & \text{con probabilidad } \alpha \\ \theta_t & \text{con probabilidad } 1 - \alpha \end{cases} \text{ y,} \\ w(t + 1) &= \begin{cases} -w_{t+1} & \text{con probabilidad } \alpha \\ -w_t & \text{con probabilidad } 1 - \alpha \end{cases}.\end{aligned}$$

---

Al igual que con otros algoritmos MCMC, la probabilidad de transición de HMC está diseñada para cumplir los requisitos teóricos para el equilibrio y la reversibilidad (Steve Brooks et al., 2011). Estas condiciones garantizan que nuestras muestras de HMC proporcionan una representación válida de la distribución final. Dos parámetros que deben especificarse son el tamaño del paso  $\epsilon$  y el número de pasos de salto  $L$ . Los elementos de la matriz de covarianza  $M$  también se pueden ajustar en lugar de normalmente usar la matriz de identidad predeterminada para mejorar la eficiencia (Samuel Thomas and Wanzhu Tu, 2020). Por lo general, es una buena práctica

establecer un valor  $\epsilon$  menor en relación con la magnitud del parámetro de interés. Uno menor da como resultado aproximaciones más cercanas y, por lo tanto, mayor tasas de aceptación. Pero una pequeña  $\epsilon$  debe ir acompañada de un valor grande  $L$  para asegurar la convergencia de la trayectoria (Samuel Thomas, 2020).

### 3.1.6. El algoritmo Markov Chain Monte Carlo ABC (MCMC ABC)

Todos los métodos MCMC revisados anteriormente, en la inferencia bayesiana, requieren del cómputo de la función objetivo  $p(\theta|y) \propto p(y|\theta) \times p(\theta)$ . Donde  $p(y|\theta)$  y  $p(\theta)$  son la verosimilitud y distribución inicial, respectivamente, pero ¿qué sucede cuando la función de verosimilitud  $p(y|\theta)$  no está disponible analíticamente o computacionalmente sea muy costosa de evaluar? La clase de algoritmos y métodos desarrollados para realizar inferencia bayesiana en este tipo de situaciones es conocido como métodos libres de verosimilitud (likelihood-free computation) o métodos bayesianos aproximados (Approximate Bayesian Computation-ABC). Los métodos libres de verosimilitud están ganando popularidad rápidamente como un enfoque práctico para ajustar modelos bajo el paradigma bayesiano que de otra forma habrían sido computacionalmente imprácticos (Steve Brooks et al., 2011). Hasta la fecha, han encontrado un uso generalizado en una amplia gama de aplicaciones.

La idea general de estos métodos se puede ver en el algoritmo de muestreo de rechazo sin verosimilitud (Tavaré et al., 1997). Dado que tenemos  $y$  datos, este procedimiento se describe a continuación:

---

**Algoritmo 5:** Muestreo de rechazo sin verosimilitud

---

Generamos  $\theta' \sim p(\theta)$ ,  
 Generar  $z$  datos del modelo  $p(z|\theta')$ ,  
 Aceptar  $\theta'$  si  $z \approx y$ .

---

De esta manera, la evaluación de la verosimilitud  $p(y|\theta')$ , esencial para la mayoría de los métodos MCMC, se compensa por una estimación de la proximidad de un conjunto de datos simulados  $z$  al conjunto de datos observados  $y$ . Aunque esta proximidad puede especificarse en varias formas, todos los métodos ABC aplican este principio básico. Un procedimiento común para mejorar la eficiencia del muestreador en un MCMC es incorporar a la función objetivo dentro de un modelo aumentado. En esta configuración, se introducen parámetros auxiliares en el modelo cuyo único propósito es facilitar los cálculos (por ejemplo los modelos jerárquicos, ver, por ejemplo, (Robert, Christian P., 1994)). La inferencia libre de verosimilitud adopta un enfoque similar al aumentar la densidad (Steve Brooks et al., 2011):

$$p(\theta, z|y)_{LF} \propto p(y|z, \theta)p(z|\theta)p(\theta).$$

donde el parámetro auxiliar  $z$  es un conjunto de datos simulado de  $p(z|\theta)$ . Se supone que la función  $p(y|z, \theta)$  es constante con respecto al punto  $z = y$ , de modo que  $p(y|y, \theta) = c$ , para alguna constante  $c > 0$ , con el resultado de que la función objetivo se recupera exactamente en  $z = y$ ; es decir,

$$p(\theta, y|y)_{LF} \propto p(y|\theta)p(\theta).$$

En el fondo la función  $p(\theta|y)_{LF}$  actúa como una aproximación de  $p(\theta|y)$ . Se hacen dos concesiones en la forma de  $p(y|z, \theta)$ , y cada una de ellas puede inducir alguna forma de aproximación a  $p(\theta, y|y)_{LF}$ . El primero considera que la función debe ser una densidad de kernel de suavizado estándar,  $K$ , centrada en el punto  $x = y$  y con escala determinada por un vector de parámetros, generalmente tomado como escalar. Es decir,

$$p_\epsilon(y|z, \theta) = \frac{1}{\epsilon} K\left(\frac{|z-y|}{\epsilon}\right).$$

La segunda concesión en la forma de  $p_\epsilon(y|z, \theta)$  permite la comparación de los conjuntos de datos,  $z$  y  $y$ , que ocurren a través de un vector de baja dimensión de alguna estadística  $T(\cdot)$ . En consecuencia, dada la baja probabilidad de generar un conjunto de datos auxiliar tal que  $z \approx y$ , se emplean funciones como:

$$p_\epsilon(y|z, \theta) = \frac{1}{\epsilon} K\left(\frac{|T(z)-T(y)|}{\epsilon}\right),$$

las cuales proporciona regiones de alto valor cuando  $T(z) \approx T(y)$  y valores bajos en caso contrario. Si el vector de estadísticas es suficiente para los parámetros, entonces el comparar las estadísticas de dos conjuntos de datos serán equivalentes a comparar los conjuntos de datos  $z$  y  $y$ , por lo tanto, no habrá pérdida de información en la selección del modelo. Sin embargo, el evento  $T(z) \approx T(y)$  será sustancialmente más probable que  $z \approx y$ , por lo que los muestreadores sin probabilidad basados en estadísticas  $T(\cdot)$  en general, son considerablemente más eficientes en términos de tasas de aceptación que las basadas solo en conjuntos de datos completos.

Una función de ponderación utilizada con frecuencia es el kernel de densidad uniforme, donde  $T(y)$  se distribuye uniformemente en la esfera centrada en  $T(z)$  con radio  $\epsilon$ . Esto se escribe comúnmente como:

$$p_\epsilon(y|z, \theta) \begin{cases} 1 & \text{si } \rho(T(z), T(y)) \leq \epsilon \\ 0 & \text{en otro caso} \end{cases},$$

donde  $\rho$  denota una medida de distancia (por ejemplo, euclidiana) entre  $T(z)$  y  $T(y)$ . La efectividad de los enfoques ABC depende de elegir una función de discrepancia adecuada  $\rho(z, y)$  y una tolerancia adecuada  $\epsilon$ . El muestreo de rechazo ABC es muy simple de implementar, aunque puede sufrir de una tasa aceptación extremadamente baja cuando la distribución inicial es muy diferente a la distribución final. Para contrarrestar esta deficiencia, se desarrolló una técnica ABC más eficiente basada en métodos MCMC. A continuación presentamos esta variante, sea  $p$  la distribución inicial de los parámetros  $\theta$  y  $q$  una distribución propuesta, la implementación del MCMC-ABC es la que sigue:

---

**Algoritmo 6:** MCMC-ABC

---

Obtener  $\theta_0$  y  $z_0$  del muestreo de rechazo ABC.

Para  $t = 1, \dots, T$

Genera  $\theta^* \sim q(\theta^*|\theta^{t-1})$ ,

Calcula

$$r = \frac{p(\theta^*)q(\theta_{t-1}|\theta^*)}{p(\theta_{t-1})q(\theta^*|\theta_{t-1})},$$

Si  $U(0, 1) < r$  entonces:

Simula  $z^*$  del modelo dado  $\theta^*$

Si  $\rho(z^*, y) < \epsilon$  entonces:

$\theta_t := \theta^*$  si no  $\theta_t := \theta_{t-1}$ ,

Si no entonces  $\theta_t := \theta_{t-1}$ .

---

# Capítulo 4

## Aplicaciones

En este capítulo primero propondremos un modelo empleando ecuaciones diferenciales no lineales, y compararemos los resultados de los tres métodos MCMC de interés mencionados en el capítulo anterior, luego revisaremos un modelo depredador-presa el cual resolveremos solamente con el método RAM, y por último trabajaremos en modelos de datos circulares utilizando los tres métodos MCMC antes mencionados más el método Metropolis-Hastings y compararemos los resultados obtenidos.

### 4.1. El modelo FH-N

El modelo de FitzHugh-Nagumo (FH-N) describe un prototipo de un sistema excitable (por ejemplo los “disparos” en una neurona). Toma su nombre de Richard FitzHugh (1922 - 2007), quien propuso el modelo teórico en 1961, así como de J. Nagumo y otros que construyeron un circuito electrónico equivalente. El modelo FitzHugh–Nagumo es una versión simplificada del modelo de Hodgkin y Huxley (1952), que describe la dinámica de una neurona más en detalle (Cody Mcrae, 2019). El sistema considera las variables  $V(t)$  como el potencial y a  $R(t)$  como la variable de recuperación de un axón. Las ecuaciones son las siguientes:

$$\begin{aligned}\frac{dV}{dt} &= \gamma \left( V - \frac{V^3}{3} + R \right), \\ \frac{dR}{dt} &= -\frac{1}{\gamma} (V - \alpha + \beta R).\end{aligned}$$



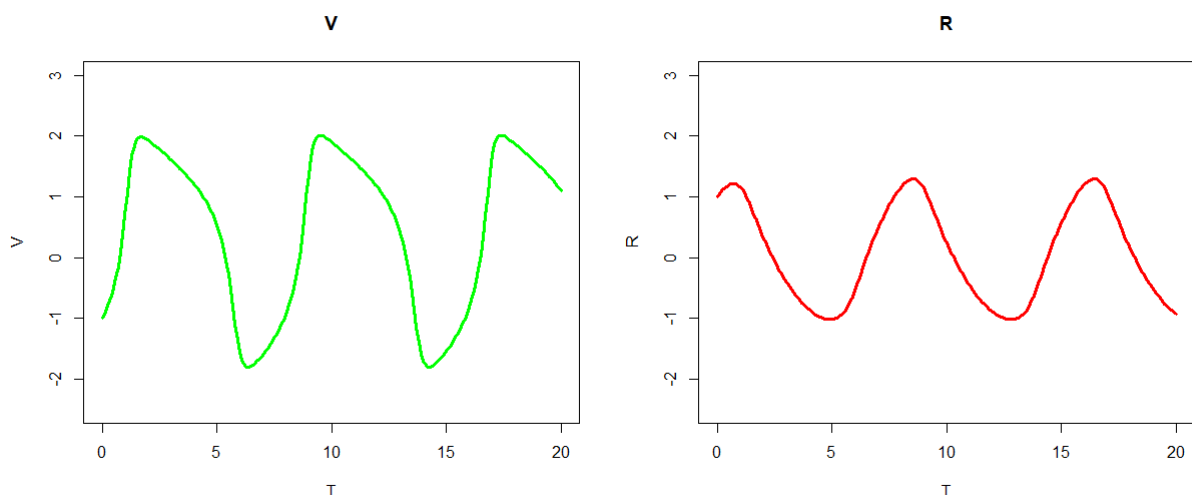


Figura 4.1: Solución del sistema FH-N que ilustra su comportamiento cualitativo

Donde los parámetros  $\alpha, \beta, \gamma \in \mathbb{R}^+$ . Para entender este modelo, simulamos datos con los valores iniciales  $V_0 = -1$ ,  $R_0 = 1$ , y parámetros  $\alpha=0.3$ ,  $\beta=0.3$ ,  $\gamma=2$  en un intervalo de tiempo de 0 a 20. Adicionalmente, incluimos un error gaussiano  $N(0, \sigma_i^2 = 0.25)$  para  $i = 1, 2$  que emula el error de medición en cada uno de los estados del sistema. Como el valor de algunos de los parámetros son positivos se tomó una distribución inicial log normal, por lo tanto, definiremos  $\theta = (\alpha, \beta, \gamma, V_0, R_0, \sigma_1, \sigma_2)$ , donde para los parámetros  $(\alpha, \beta, \gamma, \sigma_1, \sigma_2)$  su distribución inicial propuesta es:

$$\theta_i \sim LN(\mu_{\theta_i}, \sigma_{\theta_i}).$$

Por lo anterior, se cumple:

$$\ln(\theta_i) \sim N(\mu_{\theta_i}, \sigma_{\theta_i}).$$

Mientras que para  $V_0$  y  $R_0$  se les asignó una distribución inicial normal. Es posible que el experimental tenga conocimiento de algunos valores paramétricos con sus respectivas incertidumbres, en tal caso eso sirve de información a priori para incorporarla al esquema de estimación bayesiano. Así, dado el conjunto de datos  $y_{ij}$  y el sistema de ecuaciones diferenciales, se considera el siguiente modelo estadístico:

$$y_{ij} = \hat{y}_j(t_i|\theta) + \epsilon_{ij},$$

donde:

- $\hat{y}_j(t_i|\theta)$  es la solución (numérica) del sistema de ecuaciones del estado  $j$ , en este casos  $j = 1, 2$ , en el tiempo  $t_i$  con valor de los parámetros  $\theta$ .
- $\epsilon_{ij}$  es el error de medición de la observación  $i$  del estado  $j$ ,  $\epsilon_{ij} \sim N(0, \sigma_j^2)$

La distribución está dada por

$$p(\theta|y) \propto p(y|\theta) \times p(\theta).$$

Donde  $l(y|\theta)$  es la función de máxima verosimilitud asociada a una distribución normal. Es decir, si  $n_j$  el número de observaciones del estado  $j$ , luego entonces,

$$p(y|\theta) = \prod_{j=1}^2 \prod_{i=1}^{n_j} N(y_{ij}|\hat{y}_j(t_i|\theta), \sigma_j),$$

por otro lado, la distribución inicial que se propone es

$$p(\theta) = N(\ln(\alpha)|\mu_\alpha, \sigma_\alpha)N(\ln(\beta)|\mu_\beta, \sigma_\beta)N(\ln(\gamma)|\mu_\gamma, \sigma_\gamma) \\ N(V_0|\mu_{V_0}, \sigma_{V_0})N(R_0|\mu_{R_0}, \sigma_{R_0})N(\ln(\sigma_1)|\mu_{\sigma_1}, \sigma_{\sigma_1})N(\ln(\sigma_2)|\mu_{\sigma_2}, \sigma_{\sigma_2}).$$

#### 4.1.1. Método RAM

Como el método MCMC RAM realiza pasos adaptativos, a diferencia del Metropolis, no se puede controlar el soporte sobre el cual realiza los pasos, pues el método RAM mueve a todo el vector  $\theta$  sobre  $\mathbb{R}^7$  por lo que, para evitar el uso de banderas en el cómputo de la verosimilitud (que verifican que los parámetros sean positivos), realizamos el cambio a la distribución log-normal, ya que esta distribución nos facilita realizar inferencias para  $\ln(\theta)$  y pueda muestrear sobre todos los números reales. Posteriormente, regresar al soporte original de los parámetros con la exponencial. Se tomaron como hiper-parámetros los siguientes valores:

- $\ln(\alpha) \sim N(\ln(0.5), 3)$
- $\ln(\beta) \sim N(\ln(0.5), 3)$
- $\ln(\gamma) \sim N(\ln(5), 3)$
- $V_0 \sim N(0, 1)$
- $R_0 \sim N(0, 1)$
- $\ln(\sigma_1^2) \sim N(\ln(5), 5)$
- $\ln(\sigma_2^2) \sim N(\ln(5), 5)$

Con las especificaciones anteriores se obtuvo la siguiente distribución final,

$$p(\theta) = \prod_{j=1}^2 \prod_{i=1}^{n_j} N(y_{ij}|\hat{y}_j(t_i|\theta), \sigma_j) \times N(\ln(\alpha)|\ln(0.5), 3)N(\ln(\beta)|\ln(0.5), 3) \\ N(\ln(\gamma)|\ln(5), 3)N(V_0|0, 1)N(R_0|0, 1)N(\ln(\sigma_1)|\ln(5), 5)N(\ln(\sigma_2)|\ln(5), 5).$$

#### Primer Análisis

Para llevar a cabo inferencias bayesianas sobre el sistema, como primer acercamiento se realizó la estimación vía RAM solo de  $(\alpha, \beta, \gamma)$ , y suponiendo todos los demás valores conocidos. Lo anterior produjo una sobrestimación de los valores de  $\sigma_1$  y  $\sigma_2$  respecto de su verdadero valor. Esto implicó que las estimaciones y los intervalos que comprenden a los parámetros fueran más anchos de lo esperado, reflejando

así el hecho de que suponíamos que las desviaciones estándar eran mayor a la que los datos nos explicaban. Se simuló una muestra de tamaño  $n=20$ , y se corrió un RAM con un periodo de calentamiento e iteraciones intermedias de 20,000 y un lag de 45 para tener finalmente una muestra de tamaño 5,000. Las siguientes gráficas se obtuvieron al realizar las estimaciones con  $\sigma_1^2 = 5$  y  $\sigma_2^2 = 5$ , y valores iniciales conocidos,  $V_0 = -1$  y  $R_0 = 1$ .

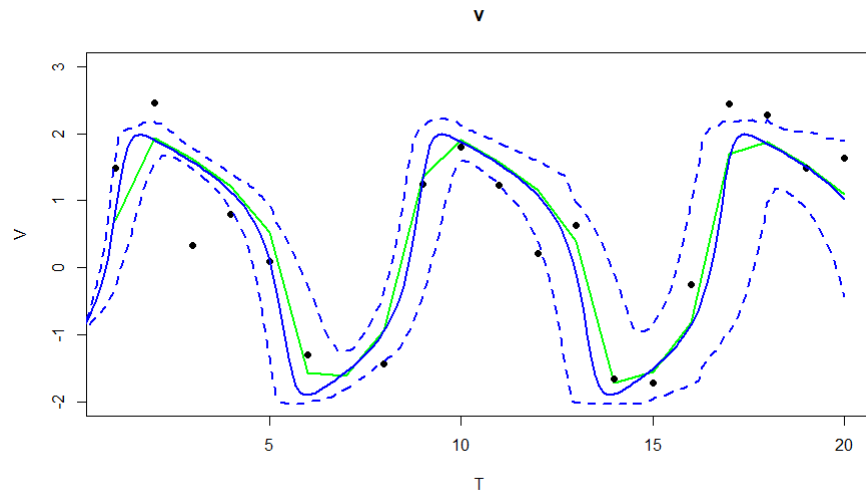


Figura 4.2: En esta figura se muestra los límites de muestra solución al ir variando sobre la muestra para es estado  $V$  con los resultados del primer análisis

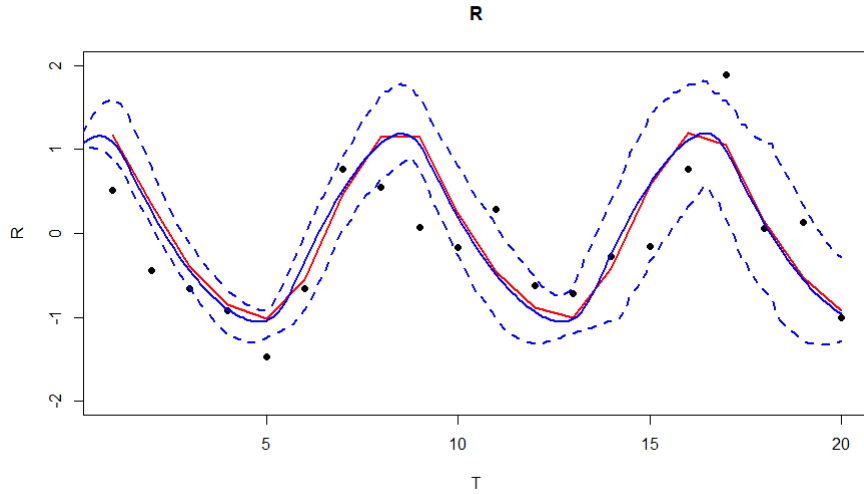


Figura 4.3: En esta figura se muestra los límites de nuestra solución al ir variando sobre la muestra para es estado  $R$  con los resultados del primer análisis

Las líneas verde y roja son la solución del verdadero sistema, mientras que las líneas azules son la solución promedio y las líneas azules punteadas son los límites de las soluciones del sistema al ir variando sobre los valores obtenidos de la distribución final. Se debe de señalar que estas bandas **no** son bandas de probabilidad, pues no se estiman los parámetros  $\sigma_1^2$  y  $\sigma_2^2$  para poder estimarlas. Los resultados obtenidos fueron:

Cuadro 4.1: Resultados 1 RAM

Parámetro	Verdadero valor	Media	Mediana	Intervalo de probabilidad
$\alpha$	0.3	0.164	0.102	$(1.085e^{-05} ; 0.513)$
$\beta$	0.3	0.275	0.114	$(5.972e^{-06} ; 0.612)$
$\gamma$	2	2.187	2.145	$(1.075 ; 2.761)$

En el cuadro 4.1 se pueden ver los valores verdaderos de cada parámetro. También se muestran la media y la mediana de las distribuciones finales de cada parámetro, como estimadores puntuales de cada uno de ellos. Adicionalmente se presentan intervalos de credibilidad al 95 % de cada parámetro, derivados de las correspondientes distribuciones finales.

Se puede ver que aunque puntualmente los parámetros no se estiman muy bien, los intervalos de credibilidad sí capturan al verdadero valor. Esto va en concordancia con los resultados de las gráficas 4.2 y 4.3.

## Segundo Análisis

Como segundo análisis se realizaron inferencias sobre el modelo FH-N, considerando ahora a los valores iniciales como parámetros desconocidos para observar si la metodología propuesta en esta tesis, via el algoritmo RAM mejoraba las inferencias para el modelo. Sin embargo, aun se consideraron fijos los valores  $\sigma_1^2 = 5$  y  $\sigma_2^2 = 5$ .

Para este análisis se simularon ahora un conjunto de 40 datos y se aplicó la metodología propuesta usando el RAM, con un periodo de calentamiento e iteraciones intermedias de 20,000 y con lag de 45 para obtener una muestra final de tamaño 5,000. Los resultados fueron:

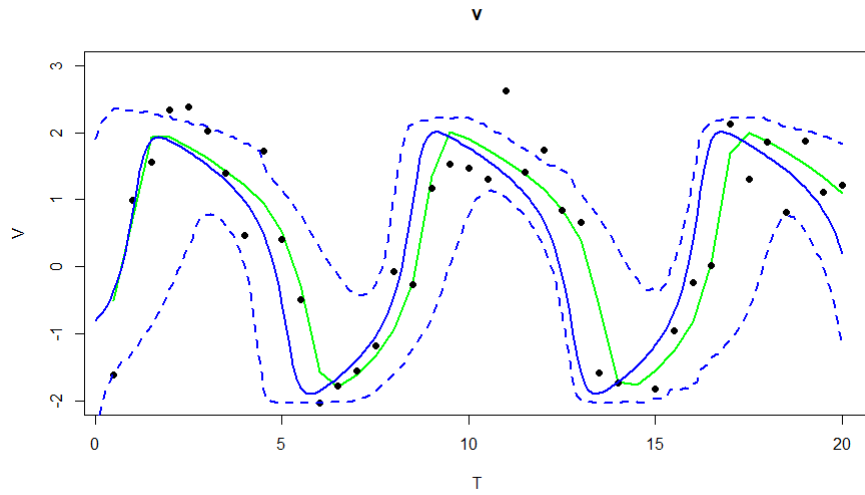


Figura 4.4: En esta figura se muestra los límites de nuestra solución al ir variando sobre la muestra para el estado  $V$  con los resultados del segundo análisis

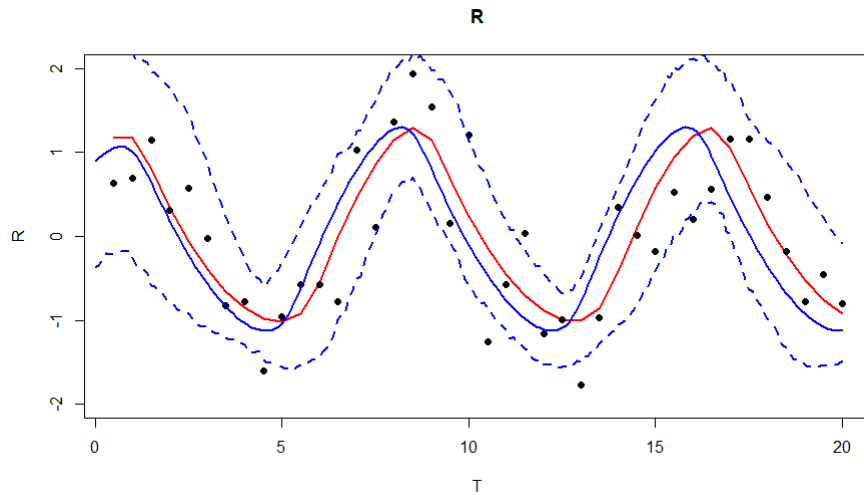


Figura 4.5: En esta figura se muestra los límites de nuestra solución al ir variando sobre la muestra para es estado  $R$  con los resultados del segundo análisis

De las gráficas 4.4 y 4.5, se puede ver que el incluir los dos valores iniciales como parámetros, se tiene una mejor descripción de la variabilidad de los datos (esta aumenta). Además, la respuesta media estimada (línea azul) mejora respecto a la verdadera respuesta (líneas verde y roja respectivamente). Sin embargo, a pesar de las inferencias mejoran, aun no son del todo adecuadas ya que algunos datos no son bien descritos por el proceso estimado.

El cuadro 4.2 muestra los estimadores puntuales y los intervalos de credibilidad al 95 % para cada uno de los parámetros involucrados en el modelo. Se puede notar que aunque los estimadores puntuales (media y mediana) todavía no son muy buenos, los intervalos de credibilidad contienen a todos los verdaderos valores de los parámetros. El análisis anterior nos dice que al parecer faltan más iteraciones para lograr una mejor convergencia de las Cadenas de Markov, pero que se va por buen camino.

Cuadro 4.2: Resultados 2 RAM

Parámetro	Verdadero valor	Media	Mediana	Intervalo de Prob.
$\alpha$	0.3	0.182	0.124	$(5.009e^{-06} ; 0.549)$
$\beta$	0.3	0.197	0.105	$(6.745e^{-06} ; 0.649)$
$\gamma$	2	1.983	2.019	$(0.994 ; 2.814)$
$V_0$	-1	-0.805	-0.809	$(-2.566 ; 0.797)$
$R_0$	1	0.892	0.825	$(-0.123 ; 2.064)$

### Tercer Análisis

Los dos escenarios anteriores se hicieron con el objetivo de ser pasos intermedios para llegar al caso general donde todos los parámetros se consideran desconocidos. Se consideraron los 40 datos, con un periodo de calentamiento e iteraciones intermedias de 50,000 y un lag de 60 para obtener una muestra final de tamaño 2,400. Las

verdaderas soluciones con los puntos simulados se pueden ver en las gráficas 4.6 y 4.7,

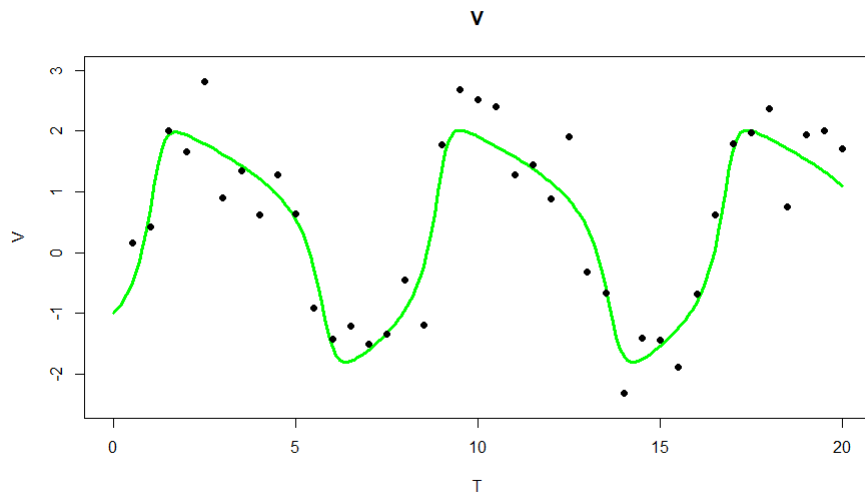


Figura 4.6: Verdadera solución para es estado  $V$  y los 40 datos simulados

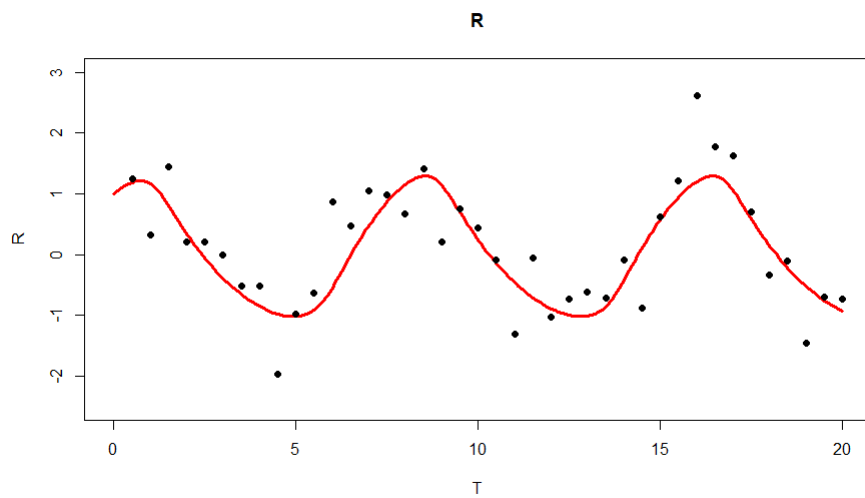


Figura 4.7: Verdadera solución para es estado  $R$  y los 40 datos simulados

Los histogramas de las muestras de la distribución final de cada uno de los parámetros  $\theta$  y  $\ln(\theta)$  para  $(\alpha, \beta, \gamma)$  donde las líneas rojas son los verdaderos valores.

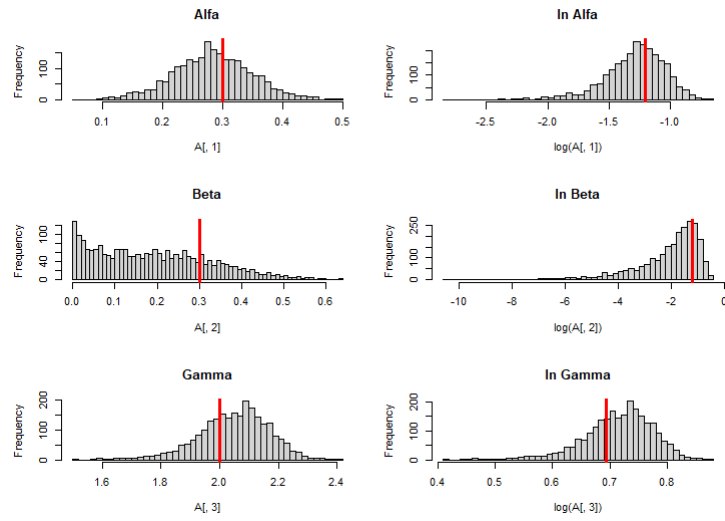


Figura 4.8: Histogramas de  $\alpha$ ,  $\beta$  y  $\gamma$  y sus respectivos logaritmos del tercer análisis

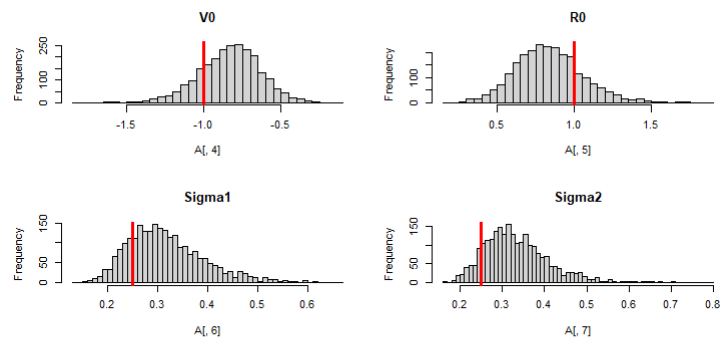


Figura 4.9: Histogramas de  $V_0$ ,  $R_0$ ,  $\sigma_1^2$  y  $\sigma_2^2$  del tercer análisis

Por su parte las siguientes figuras muestran promedios ergódicos, gráficas de auto correlación y trazas correspondientes:



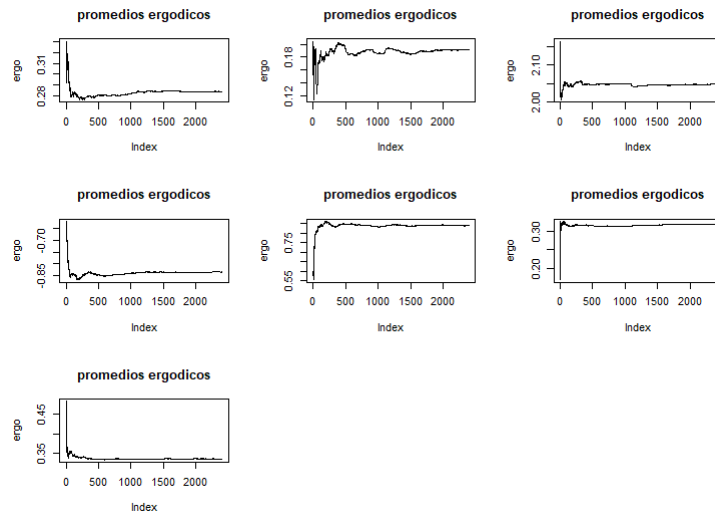


Figura 4.10: Promedios ergodicos de los parámetros  $\alpha$ ,  $\beta$ ,  $\gamma$ ,  $V_0$ ,  $R_0$ ,  $\sigma_1^2$  y  $\sigma_2^2$

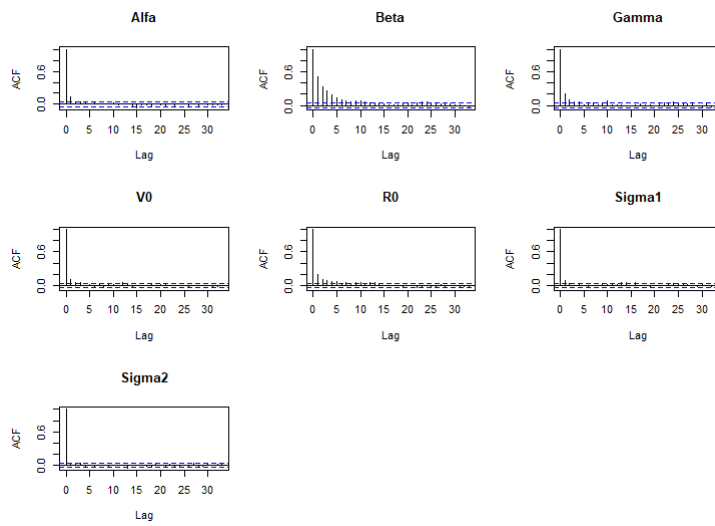


Figura 4.11: Gráficas de auto correlación de los parámetros  $\alpha$ ,  $\beta$ ,  $\gamma$ ,  $V_0$ ,  $R_0$ ,  $\sigma_1^2$  y  $\sigma_2^2$

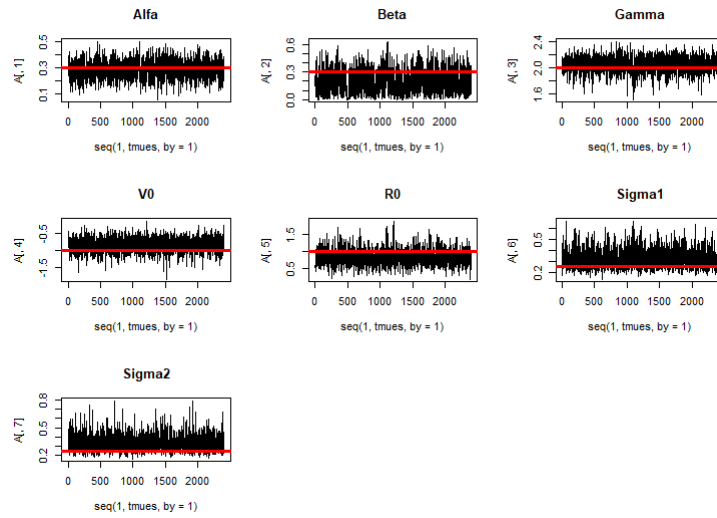


Figura 4.12: Trazas de la muestra final de los parámetros  $\alpha$ ,  $\beta$ ,  $\gamma$ ,  $V_0$ ,  $R_0$ ,  $\sigma_1^2$  y  $\sigma_2^2$

En la figura 4.13 y 4.14 se muestran las verdaderas soluciones del sistema y la solución media obtenida con la metodología desarrollada en esta tesis:

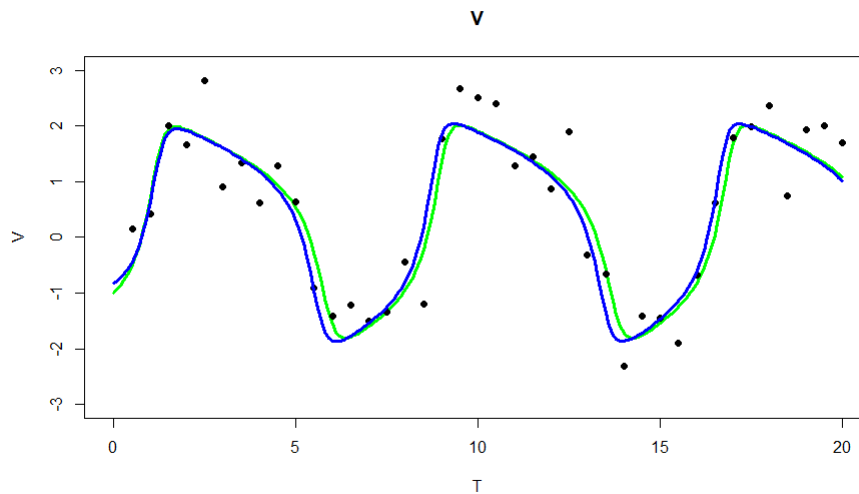


Figura 4.13: Solución media del estado  $V$  y su verdadera solución del tercer análisis

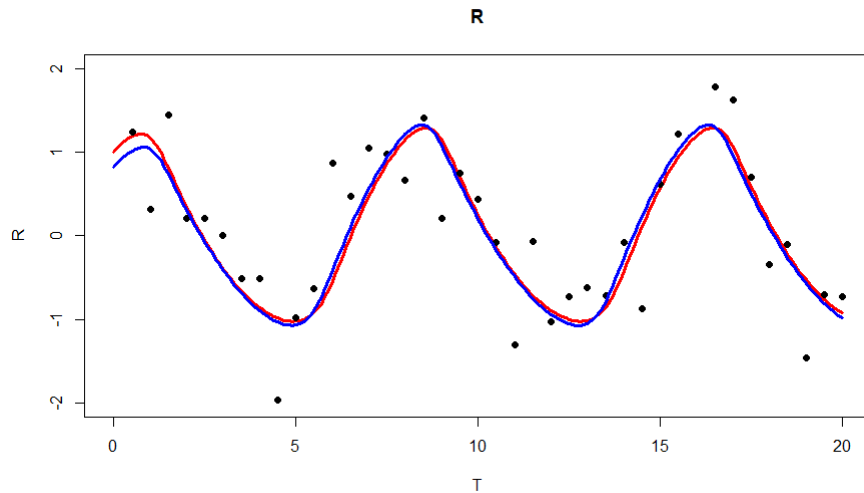


Figura 4.14: Solución media del estado  $R$  y su verdadera solución del tercer análisis

Cuadro 4.3: Resultados 40 datos

Parámetro	Verdadero valor	Media	Mediana	Intervalo de Prob.
$\alpha$	0.3	0.2835	0.2827	(0.1476 ; 0.4092)
$\beta$	0.3	0.1920	0.1770	( $2.543e^{-05}$ ; 0.4365)
$\gamma$	2	2.048	2.058	(1.802 ; 2.270)
$V_0$	-1	-0.8337	-0.8223	(-1.2128 ; -0.4369)
$R_0$	1	0.8380	0.8229	( 0.411 ; 1.268)
$\sigma_1^2$	0.25	0.3177	0.3060	(0.188 ; 0.480)
$\sigma_2^2$	0.25	0.3344	0.3227	(0.1968 ; 0.4882)

De las gráficas 4.13 y 4.14 se puede apreciar que el haber considerado a las varianzas,  $\sigma_1^2$  y  $\sigma_2^2$ , de los estados como parámetros desconocidos y haber hecho inferencia sobre ellos, la estimación puntual fue muy buena. Lo anterior, ya que la estimación se ajusta prácticamente a la verdadera solución del sistema (para ambos estados). Por otro lado, el cuadro 4.3 se puede ver que los estimadores puntuales de todos los parámetros (incluidos los valores iniciales y varianzas) aproximan bastante bien a los verdaderos valores. Adicionalmente, se puede observar que todos los intervalos de credibilidad contienen a los verdaderos valores correspondientes. Aún más, estos intervalos resultan más angostos que en los dos análisis anteriores.

Algo que no considero en los casos anteriores es que estimando las desviaciones estándar  $\sigma_1$  y  $\sigma_2$  se puede obtener bandas de probabilidad, mediante la relación:

$$y_{ij} = \hat{y}_j(t_i|\theta) + \epsilon_{ij}$$

Entonces, si resolvemos el sistema de ecuaciones con todos los vectores  $\theta$  muestreados, y se simulan 2,000 valores con la distribución normal que se considero desde el principio se pueden obtener las bandas de probabilidad de la distribución predictiva. A continuación se muestran las gráficas donde las líneas naranjas son estimaciones,

estas bandas si están justificadas por la metodología de la estadística bayesiana vista en el Capítulo 2.

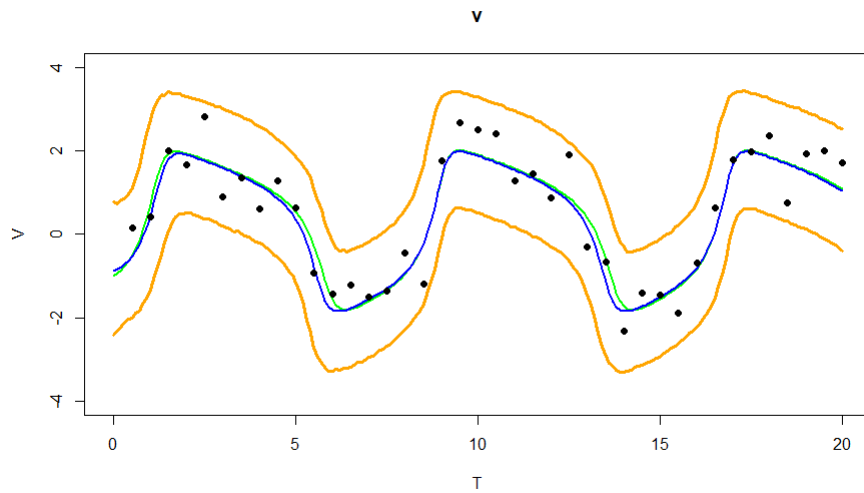


Figura 4.15: Bandas de probabilidad del estado  $V$  del tercer análisis

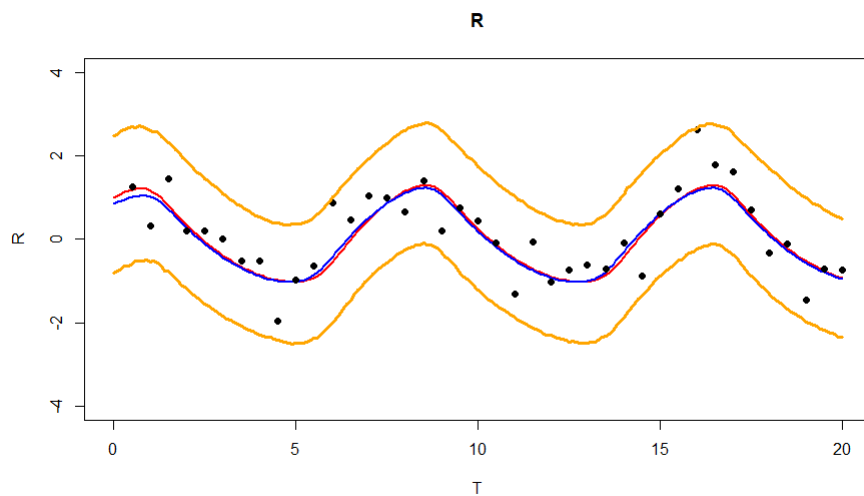


Figura 4.16: Bandas de probabilidad del estado  $R$  del tercer análisis

De la gráfica 4.15 y 4.16 y del todo análisis previo, podemos concluir que la metodología propuesta en este trabajo de tesis, resulta en una buena opción para describir datos a través de modelos como el FH-N. Hay que mencionar que este tercer análisis recurrió de un tiempo de 40 minutos. Además, se llevó a cabo en una programación en paralelo (la computación en paralelo es el uso de múltiples recursos computacionales para resolver un problema y se distingue de la computación secuencial en que varias operaciones pueden ocurrir simultáneamente, disminuyen-

do el tiempo de cómputo en comparación a la programación secuencial) usando el lenguaje R para aprovechar los ocho cores.

### Cuarto Análisis

En este cuarto análisis, se quiso investigar el efecto de tener varianzas,  $\sigma_1^2$  y  $\sigma_2^2$ , distintas para cada uno de los estados correspondientes. Es por eso, que se consideraron los 40 datos, con un periodo de calentamiento e iteraciones intermedias de 50,000 y un lag de 60 para obtener una muestra final de tamaño 2,400. Las verdaderas soluciones, así como los puntos simulados, son:

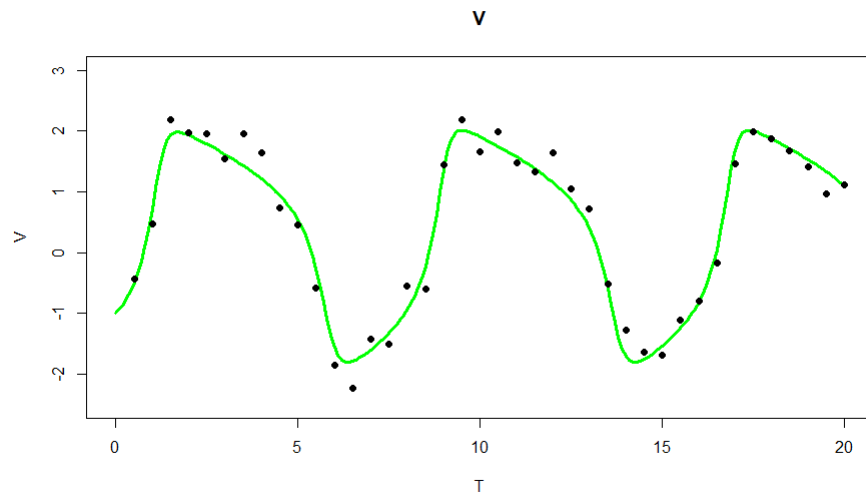


Figura 4.17: Verdadera solución para es estado  $V$  y los 40 datos simulados

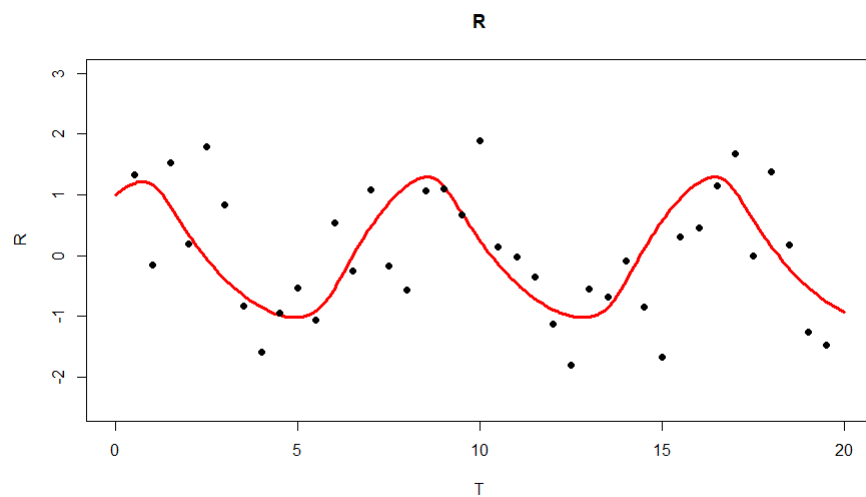


Figura 4.18: Verdadera solución para es estado  $R$  y los 40 datos simulados

Los histogramas de las muestras de la distribución final de cada uno de los parámetros  $\theta$  y  $\ln(\theta)$  para  $(\alpha, \beta, \gamma)$  con las líneas rojas son los verdaderos valores.

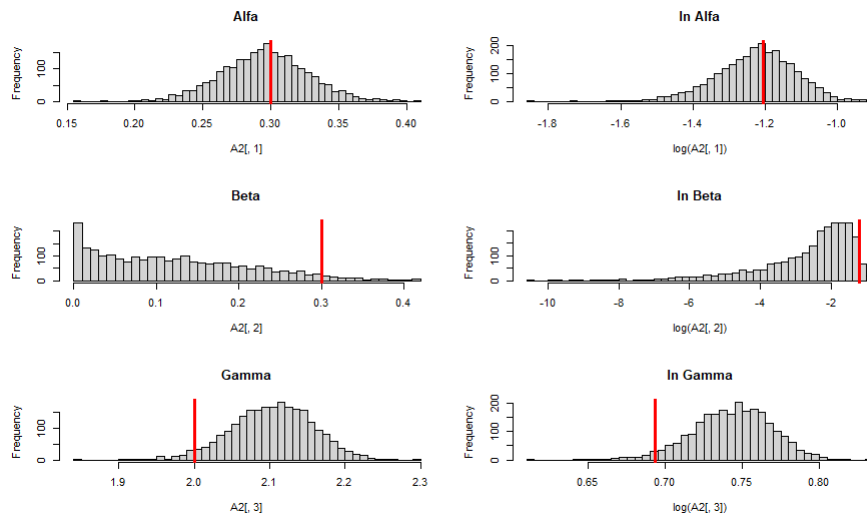


Figura 4.19: Histogramas de  $\alpha$ ,  $\beta$  y  $\gamma$  y sus respectivos logaritmos del cuarto análisis

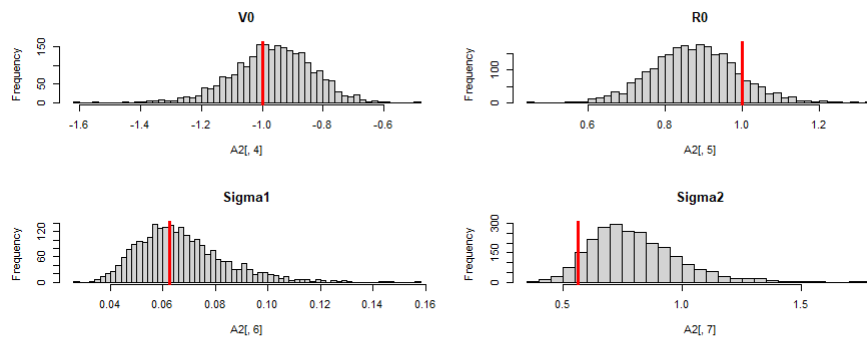


Figura 4.20: Histogramas de  $V_0$ ,  $R_0$ ,  $\sigma_1^2$  y  $\sigma_2^2$  del cuarto análisis

Por su parte las siguientes figuras muestran promedios ergódicos, gráficas de auto correlación y trazas correspondientes:

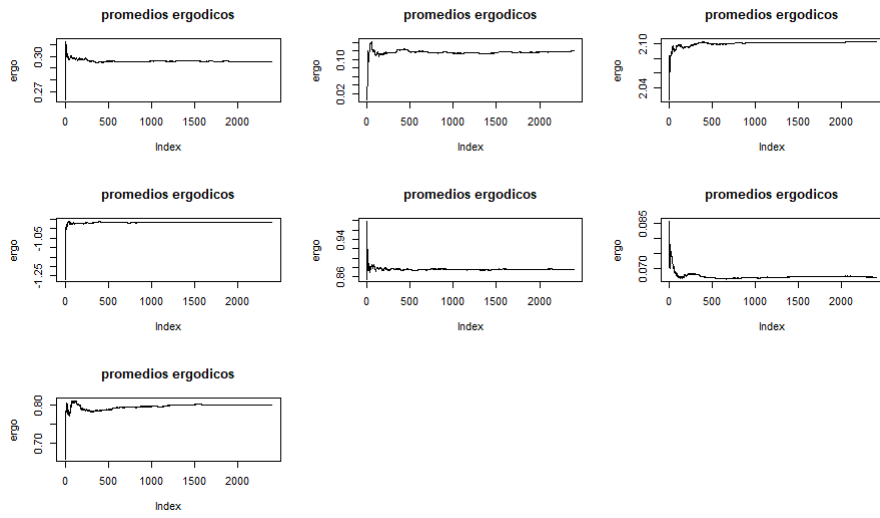


Figura 4.21: Promedios ergodicos de los parámetros  $\alpha$ ,  $\beta$ ,  $\gamma$ ,  $V_0$ ,  $R_0$ ,  $\sigma_1^2$  y  $\sigma_2^2$

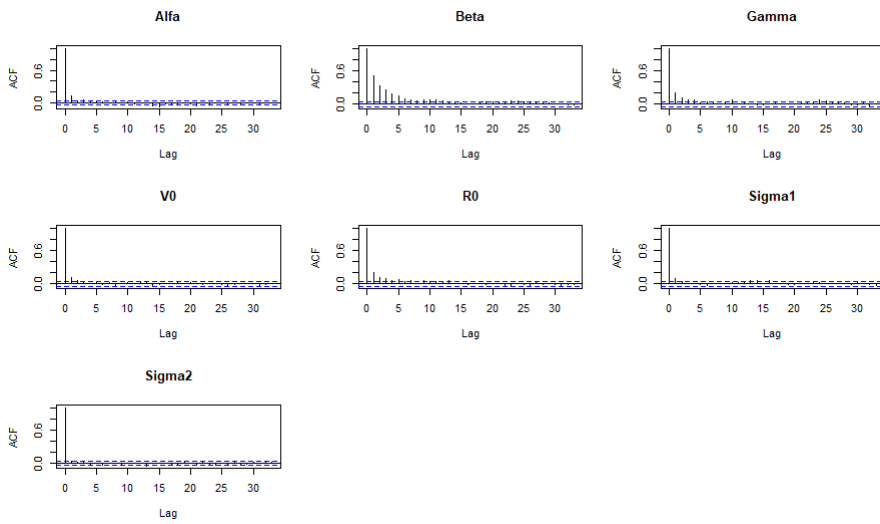


Figura 4.22: Gráficas de auto correlación de los parámetros  $\alpha$ ,  $\beta$ ,  $\gamma$ ,  $V_0$ ,  $R_0$ ,  $\sigma_1^2$  y  $\sigma_2^2$

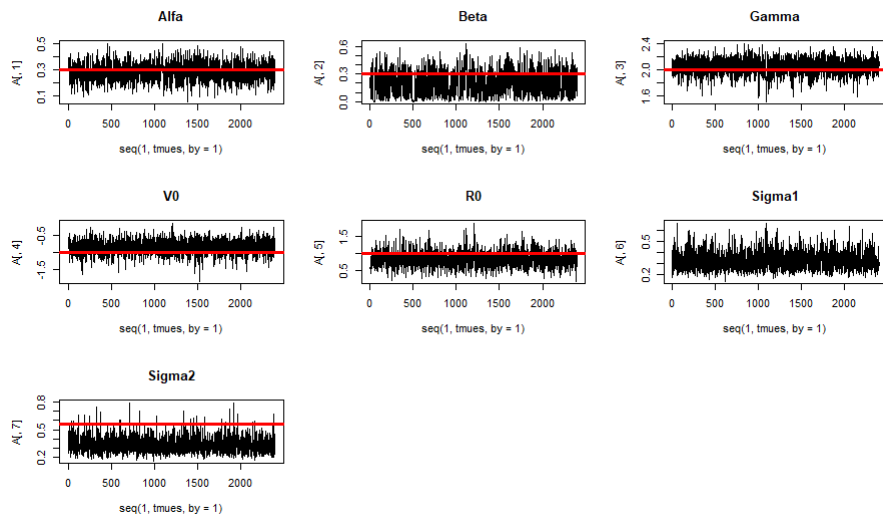


Figura 4.23: Trazas de la muestra final de los parámetros  $\alpha$ ,  $\beta$ ,  $\gamma$ ,  $V_0$ ,  $R_0$ ,  $\sigma_1^2$  y  $\sigma_2^2$

En la figura 4.24 y 4.25 se muestran las verdaderas soluciones del sistema y la solución media obtenida con la metodología desarrollada en esta tesis:

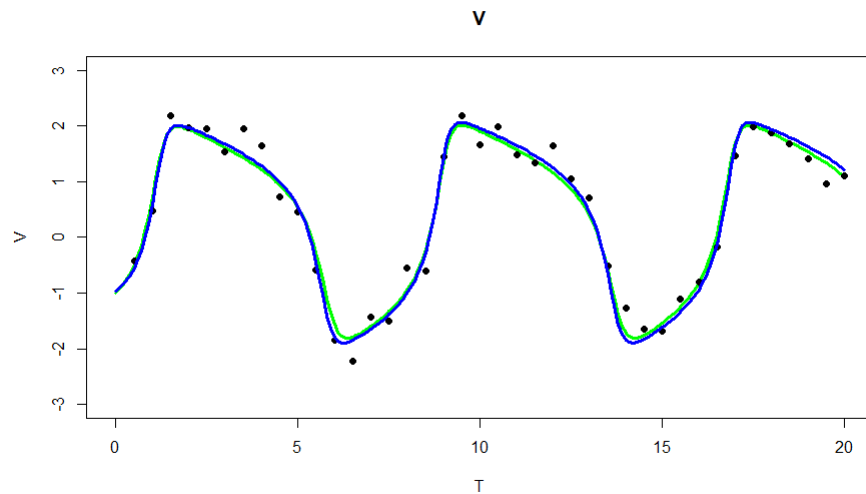


Figura 4.24: Solución media del estado  $V$  y su verdadera solución del cuarto análisis



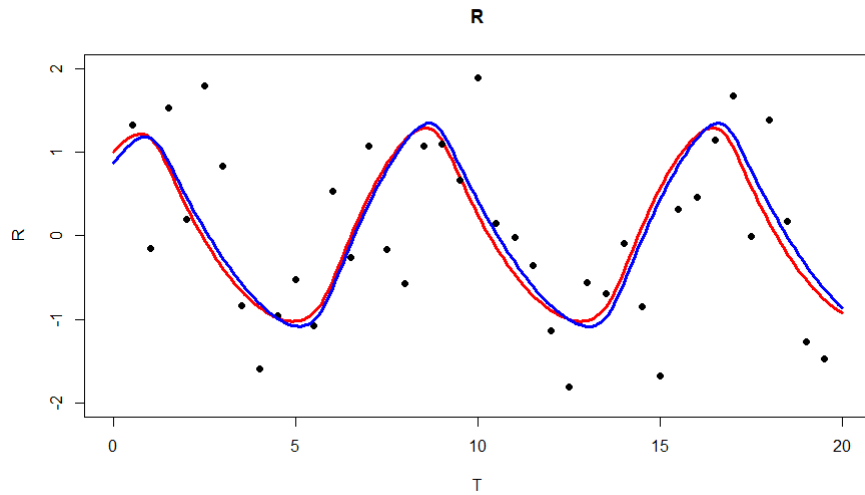


Figura 4.25: Solución media del estado  $R$  y su verdadera solución del cuarto análisis

Cuadro 4.4: Resultados 40 datos y varianzas distintas

Parámetro	Verdadero valor	Media	Mediana	Intervalo de Prob.
$\alpha$	0.3	0.2956	0.2959	(0.2352 ; 0.3573)
$\beta$	0.3	0.1202	0.1075	( $2.6e^{-05}$ ; 0.2874)
$\gamma$	2	2.103	2.106	(1.998 ; 2.2044)
$V_0$	-1	-0.9643	-0.9598	(-1.2149 ; -0.7114)
$R_0$	1	0.8755	0.8737	(0.6615 ; 1.0932)
$\sigma_1^2$	0.0625	0.0669	0.0647	(0.0376 ; 0.0996)
$\sigma_2^2$	0.5625	0.8010	0.7763	(0.4955 ; 1.1829)

En las gráficas 4.26 y 4.27, se puede notar que el algoritmo no tiene dificultades cuando las varianzas de los estados son distintas, y el hecho que las variaciones en uno de los estados del sistema sea distinta al otro estado no perjudica la calidad de las estimaciones. Además el cuadro 4.4 nos deja ver que los estimadores puntuales sigue siendo buenos, y que, los intervalos de probabilidad contienen a todos los verdaderos valores de sus respectivos parámetros.

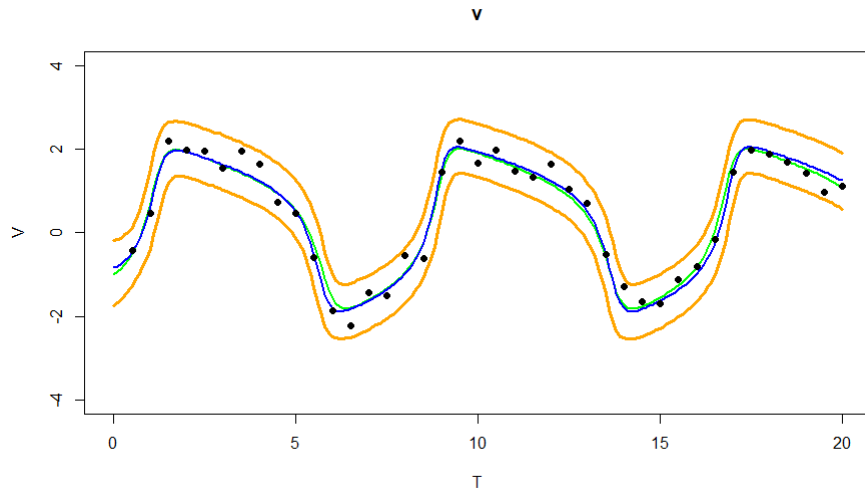


Figura 4.26: Bandas de probabilidad del estado  $V$  del cuarto análisis

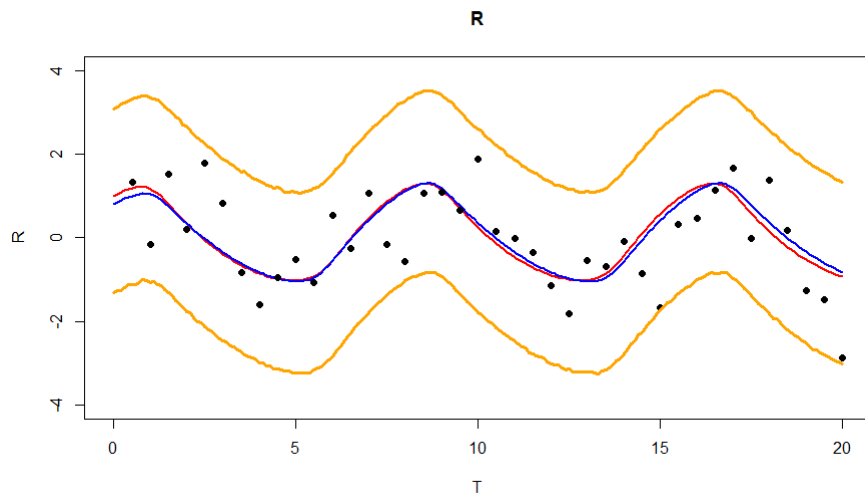


Figura 4.27: Bandas de probabilidad del estado  $R$  del cuarto análisis

Hay que mencionar que este cuarto análisis recurrió de un tiempo de 45 minutos, similar al análisis anterior, se llevo a cabo en una programación en paralelo usando el lenguaje R para aprovechar los ocho cores.

### Quinto Análisis

Se repitió este mismo escenario con todos los parámetros desconocidos, solo que esta vez se tomaron 20 datos. El objetivo, era analizar como impactaba un tamaño de muestra menor a comparación de los análisis anteriores en nuestra modelación. Las verdaderas soluciones, así como los puntos simulados, son:

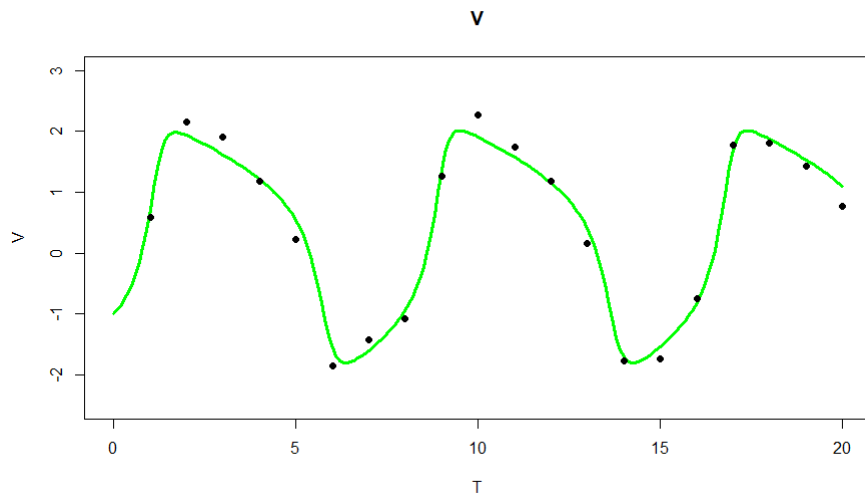


Figura 4.28: Verdadera solución para es estado  $V$  y los 20 datos simulados

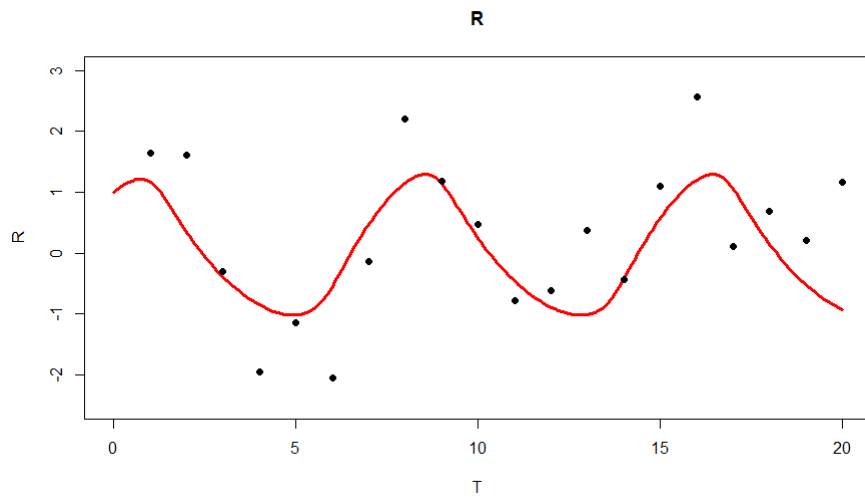


Figura 4.29: Verdadera solución para es estado  $R$  y los 20 datos simulados

Los histogramas resultantes, con las líneas rojas indicando los verdaderos valores, fueron:

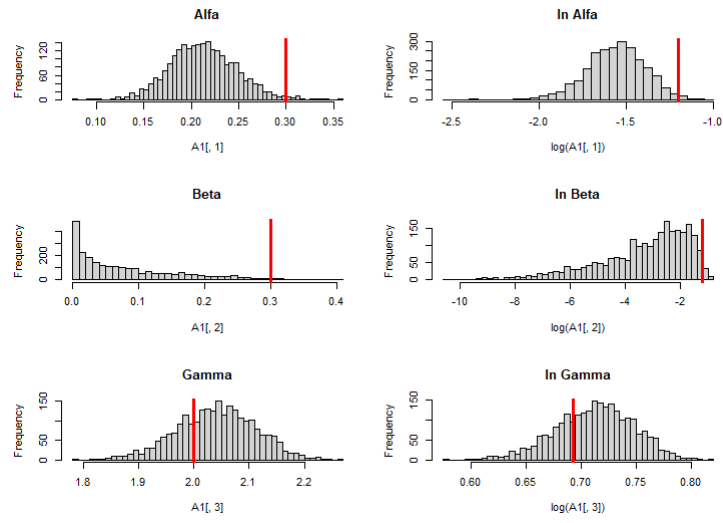


Figura 4.30: Histogramas de  $\alpha$ ,  $\beta$  y  $\gamma$  y sus respectivos logaritmos del quinto análisis

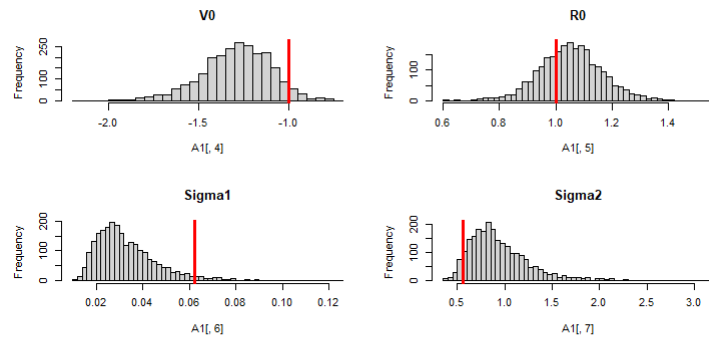


Figura 4.31: Histogramas de  $V_0$ ,  $R_0$ ,  $\sigma_1^2$  y  $\sigma_2^2$  del quinto análisis

Sus promedios ergódicos, así como, las gráficas de auto correlación:

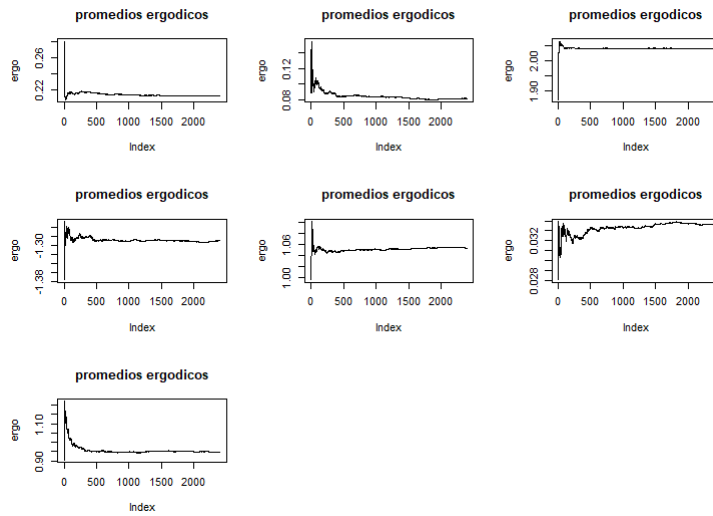


Figura 4.32: Promedios ergódicos de los parámetros  $\alpha$ ,  $\beta$ ,  $\gamma$ ,  $V_0$ ,  $R_0$ ,  $\sigma_1^2$  y  $\sigma_2^2$

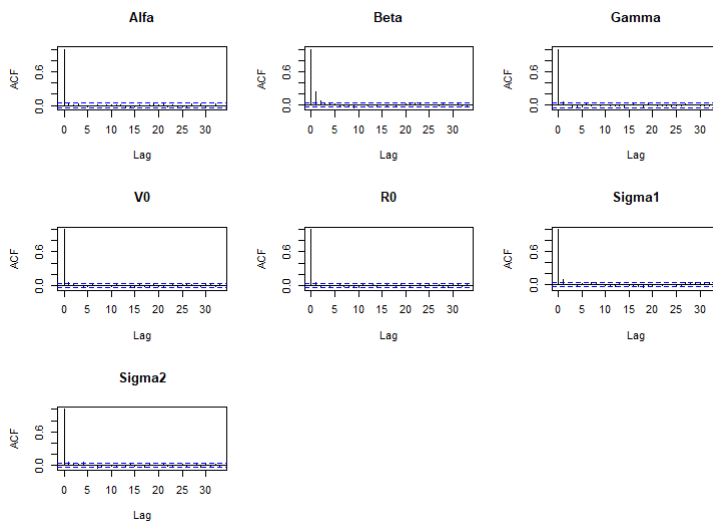


Figura 4.33: Gráficas de auto correlación de los parámetros  $\alpha$ ,  $\beta$ ,  $\gamma$ ,  $V_0$ ,  $R_0$ ,  $\sigma_1^2$  y  $\sigma_2^2$

Se puede notar claramente que aunque realizamos las mismas iteraciones que en el caso anterior (muestra de 40 datos), éstas pudieron no haber sido suficientes, lo cual nos lleva concluir que: A menor número de datos requerimos un mayor número de iteraciones. Por otra parte, las trazas de la muestra final para cada parámetro:

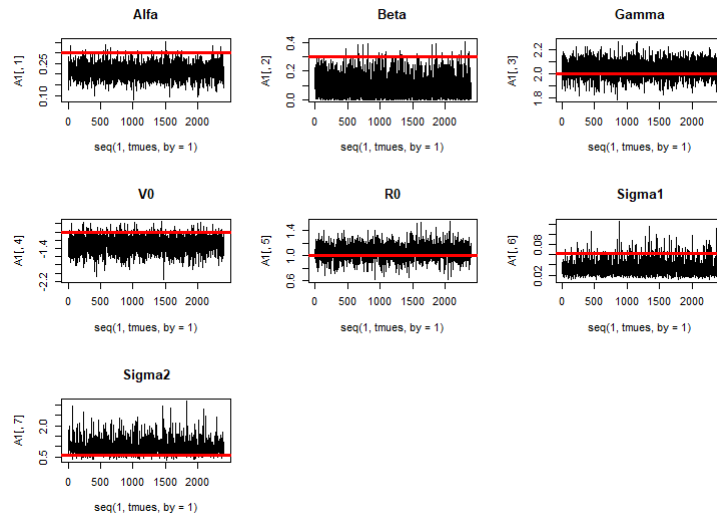


Figura 4.34: Trazas de la muestra final de los parámetros  $\alpha$ ,  $\beta$ ,  $\gamma$ ,  $V_0$ ,  $R_0$ ,  $\sigma_1^2$  y  $\sigma_2^2$

Una comparación de la verdadera solución y la solución promedio, se puede observar en las siguientes gráficas:

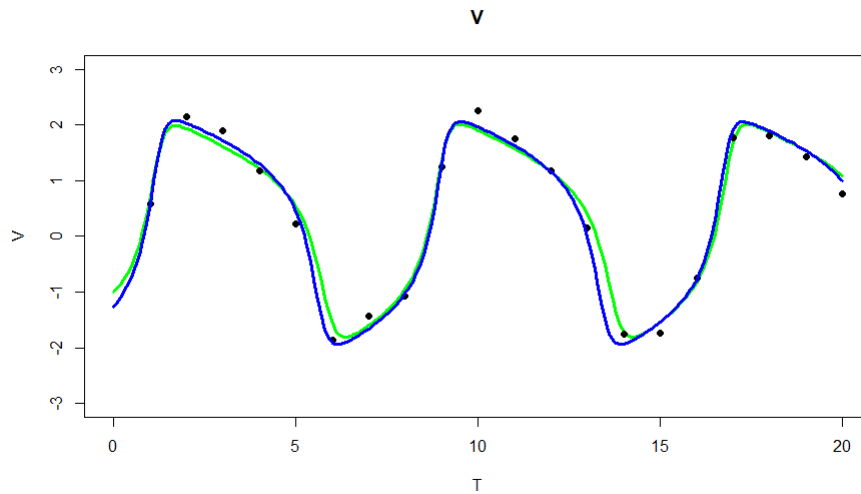


Figura 4.35: Solución media del estado  $V$  y su verdadera solución del quinto análisis

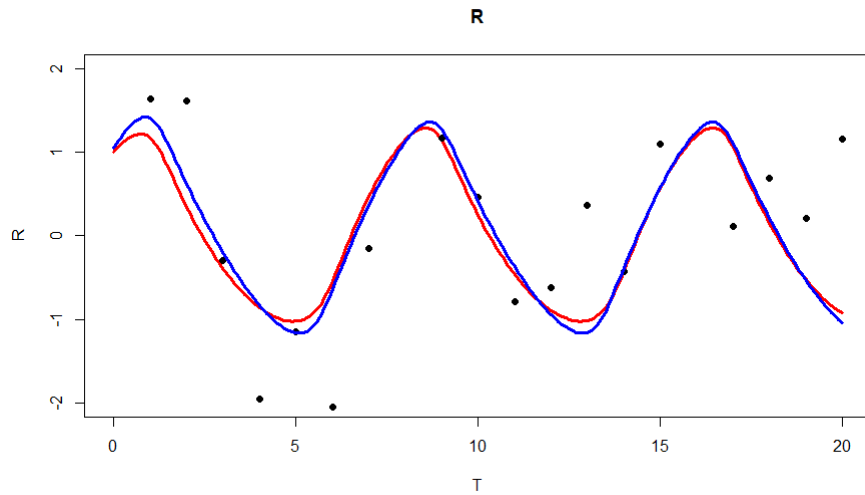


Figura 4.36: Solución media del estado  $R$  y su verdadera solución del quinto análisis

Cuadro 4.5: Resultados 20 datos

Parámetro	Verdadero valor	Media	Mediana	Intervalo de Prob.
$\alpha$	0.3	0.2127	0.2116	(0.1469 ; 0.2848)
$\beta$	0.3	0.0819	0.0559	( $2.968e^{-05}$ ; 0.2491501)
$\gamma$	2	2.040	2.042	(1.907 ; 2.180)
$V_0$	-1	-1.289	-1.277	(-1.6703 ; -0.9356)
$R_0$	1	1.0535	1.0533	(0.8497 ; 1.3039)
$\sigma_1^2$	0.0625	0.0335	0.0303	(0.0135 ; 0.0612)
$\sigma_2^2$	0.5625	0.946	0.873	(0.4482 ; 1.636)

Otro motivo que nos hace creer que el número de iteraciones no fue suficiente son los intervalos de credibilidad para los parámetros  $\alpha$  y  $\beta$  que no contienen a los verdaderos valores, por último, repitiendo la metodología de la simulación anterior, podemos estimar la distribución predictiva con las bandas de probabilidad, obteniendo los siguientes resultados:

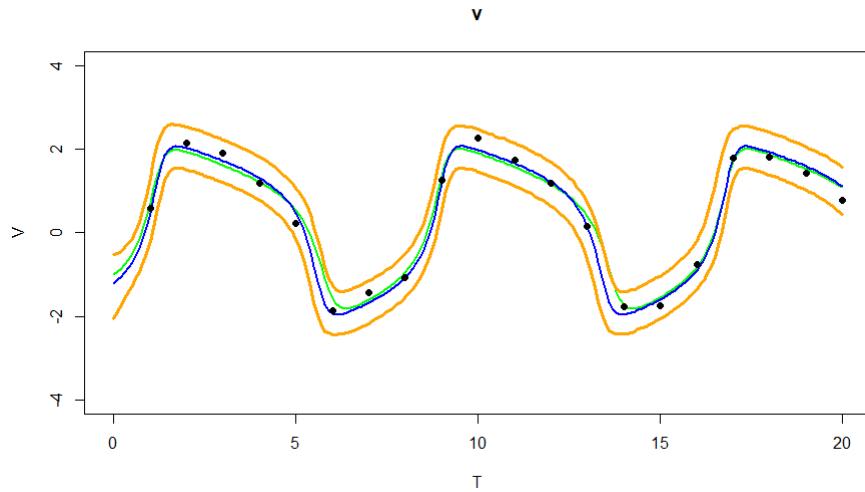


Figura 4.37: Bandas de probabilidad del estado  $V$  del quinto análisis

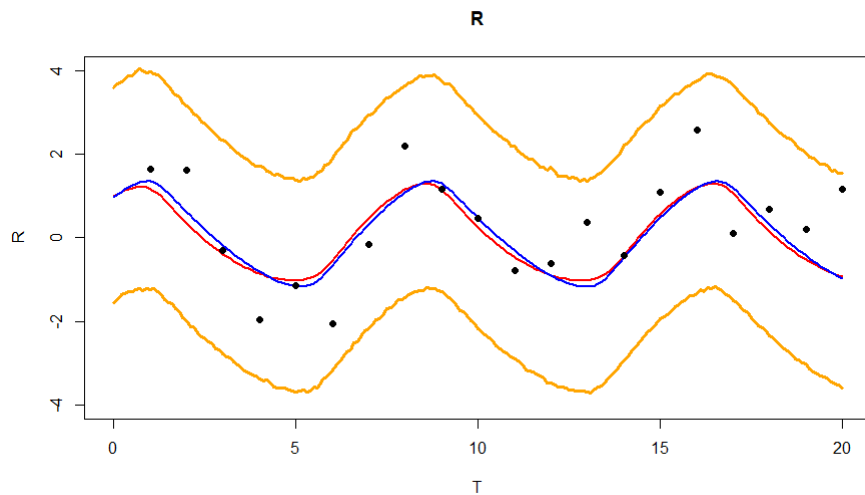


Figura 4.38: Bandas de probabilidad del estado  $R$  del quinto análisis

Hay que mencionar que este quinto análisis recurrió de un tiempo de 20 minutos, similar al análisis anterior, se llevo a cabo en una programación en paralelo usando el lenguaje R para aprovechar los ocho cores. La disminución en el tiempo se debe a la disminución en el número de datos, sin embargo, el tener un menor número de datos también implica que se requieren más iteraciones, algo que siempre se debe de tomar en cuenta para cualquier método MCMC.

#### 4.1.2. Método HMC

Para el método hamiltoniano, con el mismo modelo y los mismos valores en los parámetros  $\theta = (\alpha = 0.3, \beta = 0.3, \gamma = 2, V_0 = -1, R_0 = 1, \sigma_1^2 = 0.25, \sigma_2^2 = 0.25)$ , se



realizaron las simulaciones utilizando la función “hmc” del paquete hmclearn de R. Se tomaron como hiper-parámetros los siguientes valores:

- $\ln(\alpha) \sim N(\ln(0.5), 3)$
- $\ln(\beta) \sim N(\ln(0.5), 3)$
- $\ln(\gamma) \sim N(\ln(5), 3)$
- $V_0 \sim N(0, 1)$
- $R_0 \sim N(0, 1)$
- $\ln(\sigma_1^2) \sim N(\ln(5), 10)$
- $\ln(\sigma_2^2) \sim N(\ln(5), 10)$

Uno de los mayores intereses de este método fue el cálculo del gradiente, el cual se calculó numéricamente mediante diferencias finitas con pasos de tamaños  $\epsilon = 10^{-4}$ . Anteriormente utilizamos el método Richardson pero los tiempos de cómputo se volvían excesivamente grandes (más de dos días). Nuestros datos para ambos estados son los siguientes:

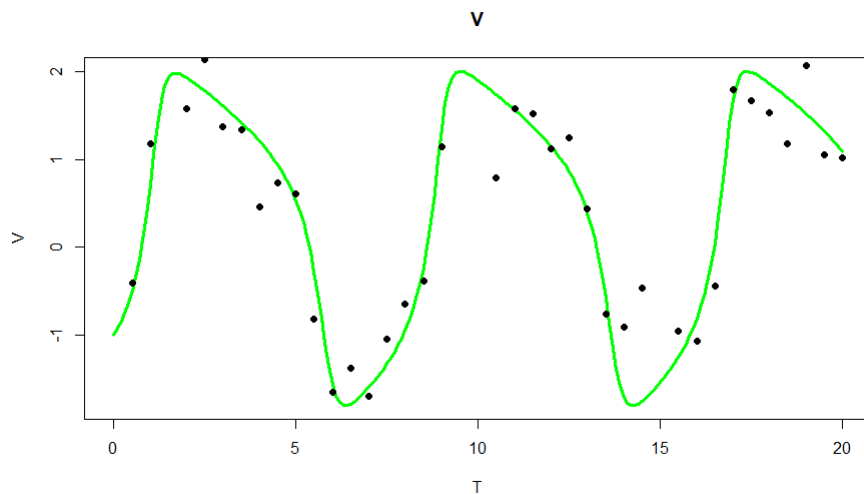


Figura 4.39: Verdadera solución para es estado  $V$  y los 40 datos simulados

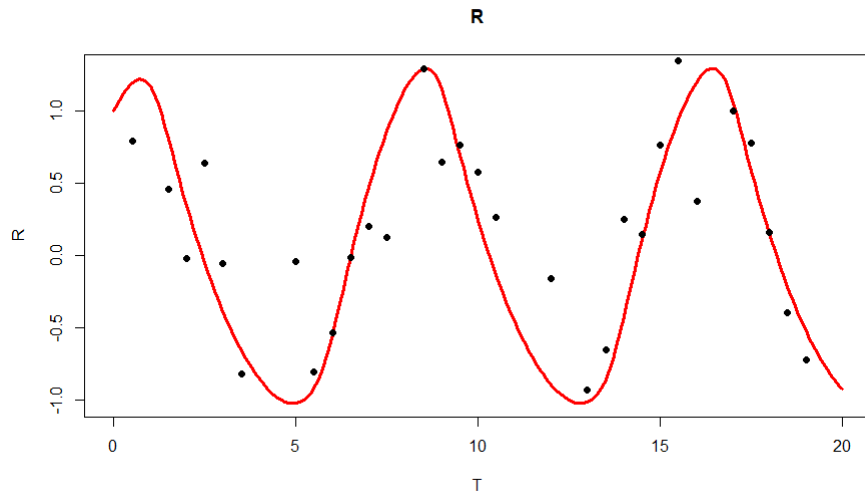


Figura 4.40: Verdadera solución para es estado  $R$  y los 40 datos simulados

### Primer HMC

Para nuestro primer estudio se realizaron 10,000 iteraciones con pasos del método Leapfrog de tamaño  $L$  igual a 10 con  $\epsilon=0.01$  y solo se tomaron las últimas 1,000 como muestra final de nuestros parámetros, obteniendo así los siguientes resultados:

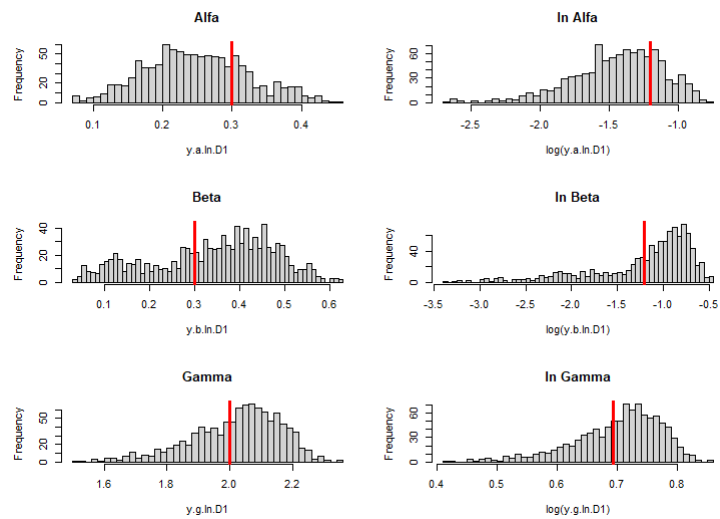


Figura 4.41: Histogramas de  $\alpha$ ,  $\beta$  y  $\gamma$  y sus respectivos logaritmos de la primera simulación de HMC

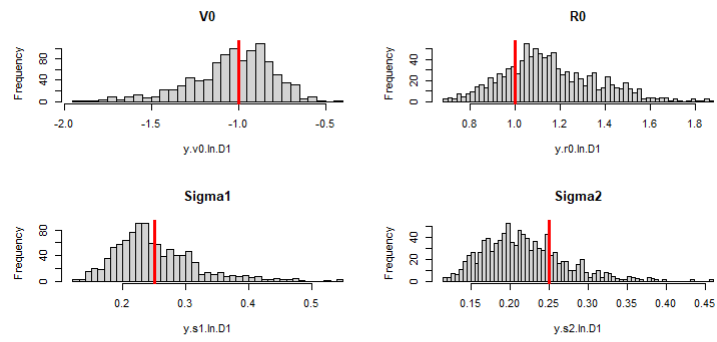


Figura 4.42: Histogramas de  $V_0$ ,  $R_0$ ,  $\sigma_1^2$  y  $\sigma_2^2$  de la primera simulación de HMC

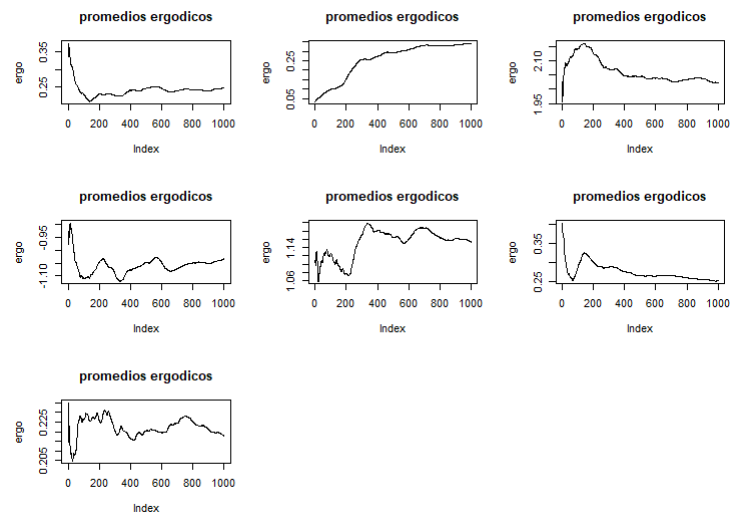


Figura 4.43: Promedios ergódicos de los parámetros  $\alpha$ ,  $\beta$ ,  $\gamma$ ,  $V_0$ ,  $R_0$ ,  $\sigma_1^2$  y  $\sigma_2^2$

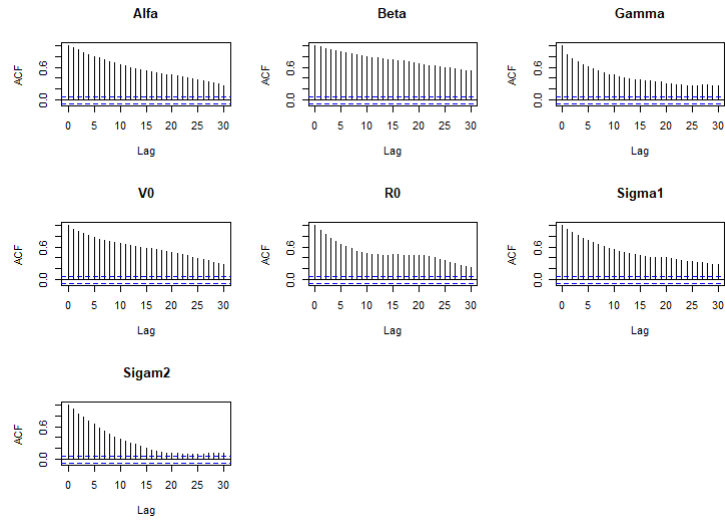


Figura 4.44: Gráficas de auto correlación de los parámetros  $\alpha$ ,  $\beta$ ,  $\gamma$ ,  $V_0$ ,  $R_0$ ,  $\sigma_1^2$  y  $\sigma_2^2$

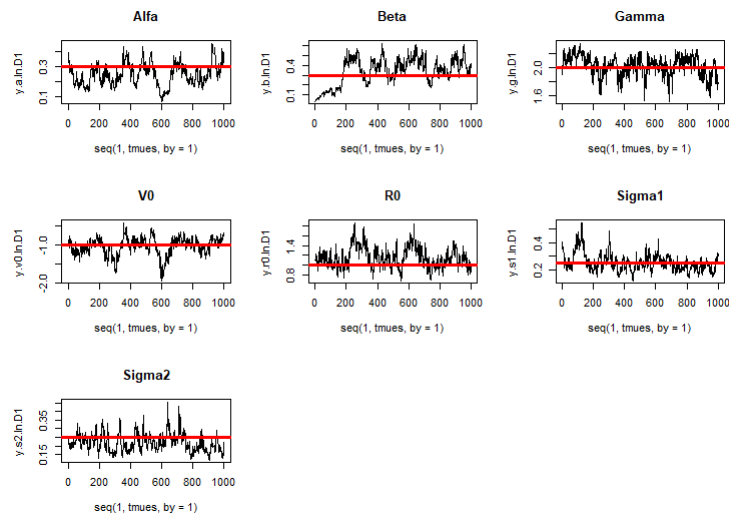


Figura 4.45: Trazas de la muestra final de los parámetros  $\alpha$ ,  $\beta$ ,  $\gamma$ ,  $V_0$ ,  $R_0$ ,  $\sigma_1^2$  y  $\sigma_2^2$

Donde nuestros resultados numéricos fueron los siguientes:

Cuadro 4.6: 1 HMC

Parámetro	Verdadero valor	Media	Mediana	Intervalo de Prob.
$\alpha$	0.3	0.2487	0.2459	(0.1181 ; 0.3993)
$\beta$	0.3	0.3421	0.3647	(0.0798 ; 0.5659)
$\gamma$	2	2.022	2.045	(1.711 ; 2.2564)
$V_0$	-1	-1.033	-1.007	(-1.583 ; -0.6581)
$R_0$	1	1.153	1.127	(0.7948 ; 1.5558)
$\sigma_1^2$	0.25	0.2508	0.2379	(0.1405 ; 0.3899)
$\sigma_2^2$	0.25	0.2178	0.2118	(0.1263 ; 0.3152)

En el cuadro 4.6 todavía no se puede notar una diferencia significativa entre el RAM y el HMC, solo en el tiempo, pues fue con un tiempo total de cómputo de 5 horas. Podemos ver en la siguiente gráfica nuestra verdadera solución del sistema (líneas verde y roja respectivamente, al estado  $V$  y  $R$ ) y nuestra solución media estimada.

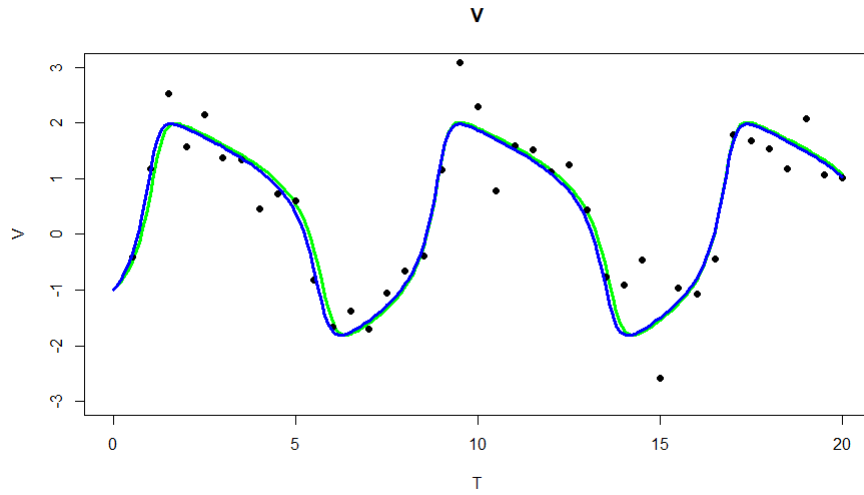


Figura 4.46: Solución media del estado  $V$  y su verdadera solución de la primera simulación de HMC

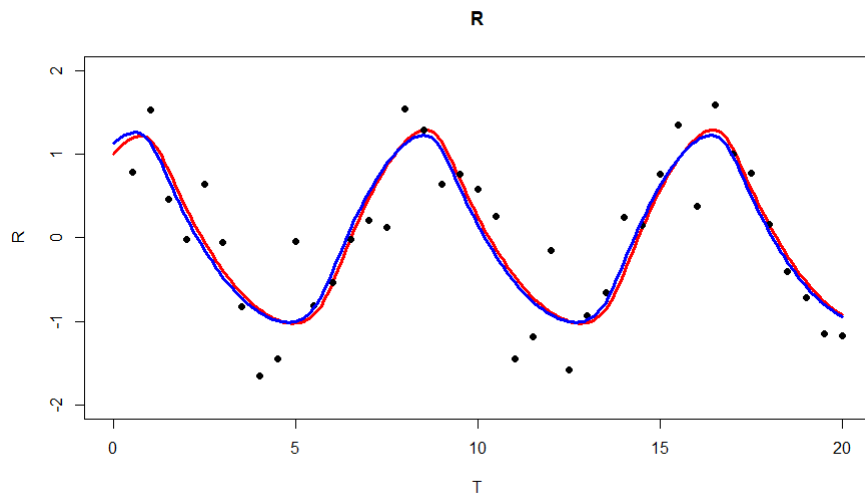


Figura 4.47: Solución media del estado  $R$  y su verdadera solución de la primera simulación de HMC

Las bandas de probabilidad obtenidas para la distribución predictiva para nuestra solución verdadera se presenta en las siguientes gráficas, (líneas naranjas).

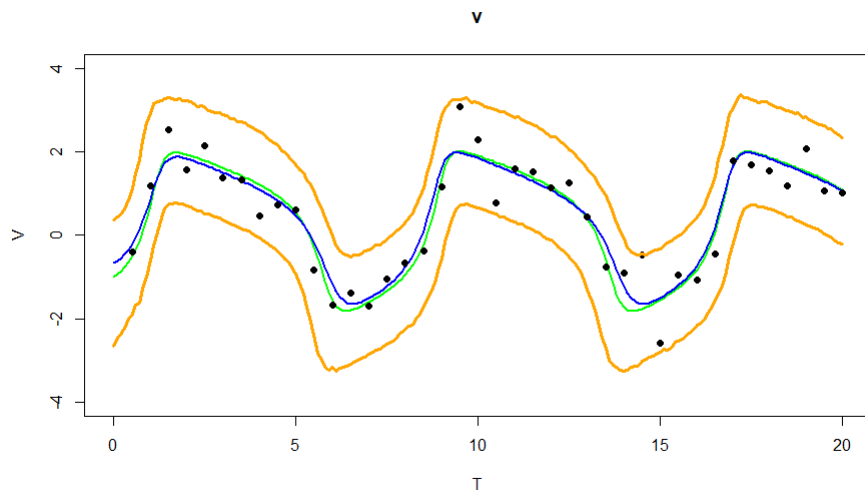


Figura 4.48: Bandas de probabilidad del estado  $V$  de la primera simulación de HMC

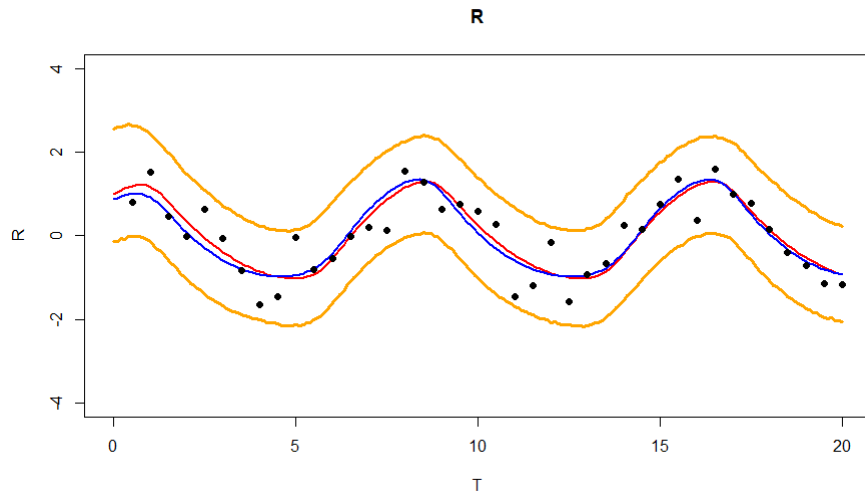


Figura 4.49: Bandas de probabilidad del estado  $R$  de la primera simulación de HMC

### Segundo HMC

Para nuestro segundo estudio se realizaron 10,000 iteraciones con pasos del método Leapfrog de tamaño  $L$  igual a 20 con  $\epsilon=0.01$  y solo se tomaron las últimas 1,000 como muestra de nuestros parámetros, obteniendo así los siguientes resultados:

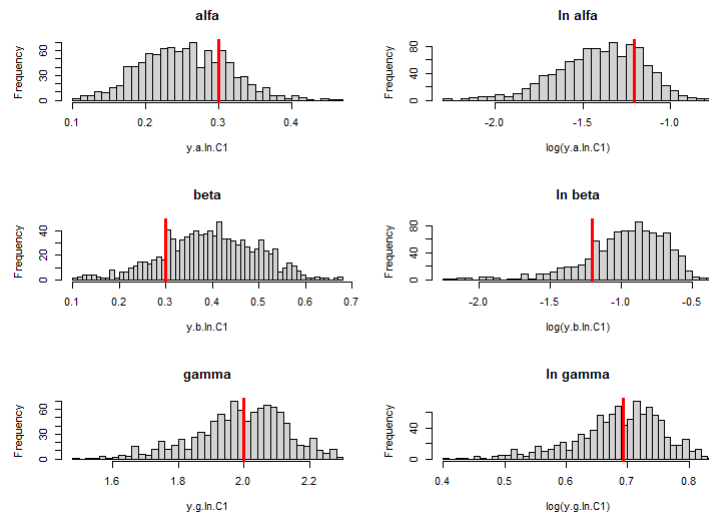


Figura 4.50: Histogramas de  $\alpha$ ,  $\beta$  y  $\gamma$  y sus respectivos logaritmos de la segunda simulación de HMC

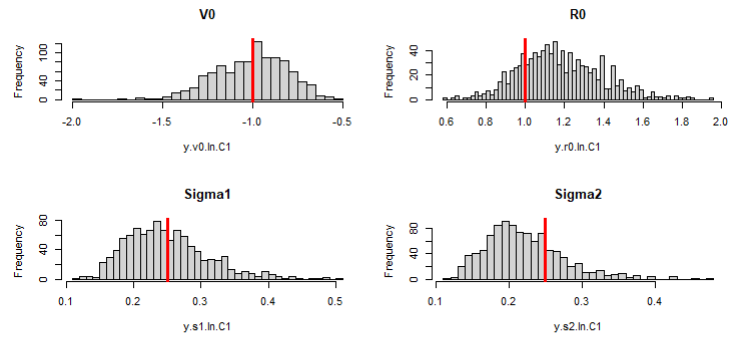


Figura 4.51: Histogramas de  $V_0$ ,  $R_0$ ,  $\sigma_1^2$  y  $\sigma_2^2$  de la segunda simulación de HMC

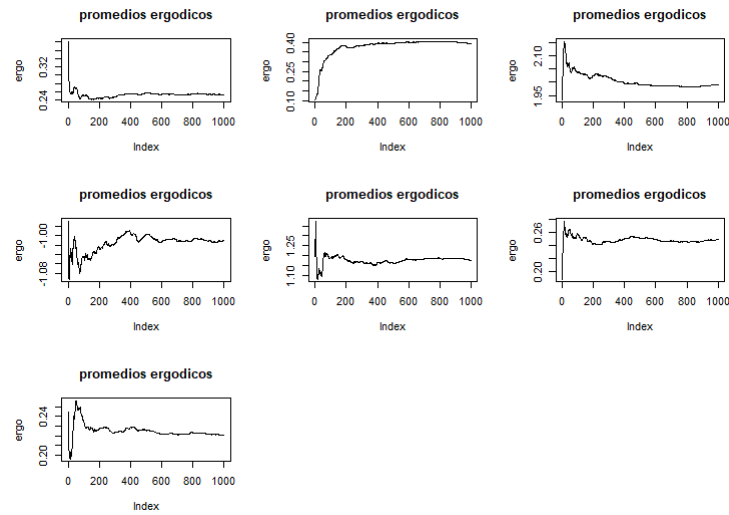


Figura 4.52: Promedios ergódicos de los parámetros  $\alpha$ ,  $\beta$ ,  $\gamma$ ,  $V_0$ ,  $R_0$ ,  $\sigma_1^2$  y  $\sigma_2^2$



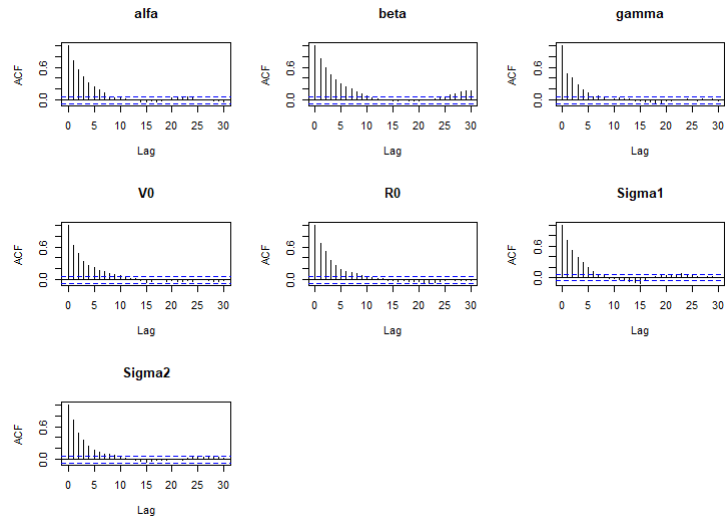


Figura 4.53: Gráficas de auto correlación de los parámetros  $\alpha$ ,  $\beta$ ,  $\gamma$ ,  $V_0$ ,  $R_0$ ,  $\sigma_1^2$  y  $\sigma_2^2$

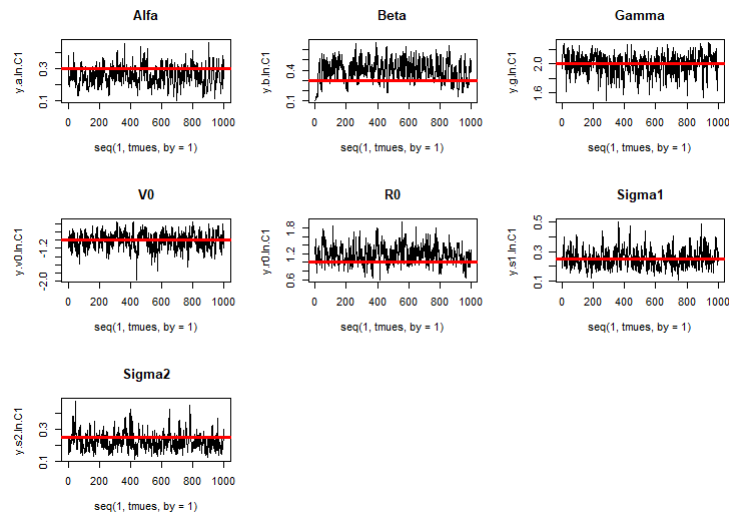


Figura 4.54: Trazas de la muestra final de los parámetros  $\alpha$ ,  $\beta$ ,  $\gamma$ ,  $V_0$ ,  $R_0$ ,  $\sigma_1^2$  y  $\sigma_2^2$

Donde nuestros resultados numéricos fueron los siguientes:

Cuadro 4.7: 2 HMC

Parámetro	Verdadero valor	Media	Mediana	Intervalo de Prob.
$\alpha$	0.3	0.2533	0.2517	(0.1379 ; 0.3641)
$\beta$	0.3	0.3952	0.3968	(0.2107 ; 0.5814)
$\gamma$	2	1.990	2.001	(1.728 ; 2.268)
$V_0$	-1	-1.0107	-0.987	(-1.398 ; -0.656)
$R_0$	1	1.176	1.154	(0.806 ; 1.6064)
$\sigma_1^2$	0.25	0.249	0.241	(0.1507 ; 0.3693)
$\sigma_2^2$	0.25	0.220	0.2125	(0.134 ; 0.3269)

Podemos ver que nuestros intervalos de probabilidad mejoraron bastante a comparación del escenario anterior y ya se puede notar una diferencia entre el método HMC y el RAM. Sin embargo, el tiempo de cómputo se volvió el doble con un total de 10 horas. En la siguiente gráfica podemos comparar nuestra verdadera solución del sistema (líneas verde y roja respectivamente al estado V y R) y nuestra solución estimada con la mediana de nuestra muestra de los parámetros.

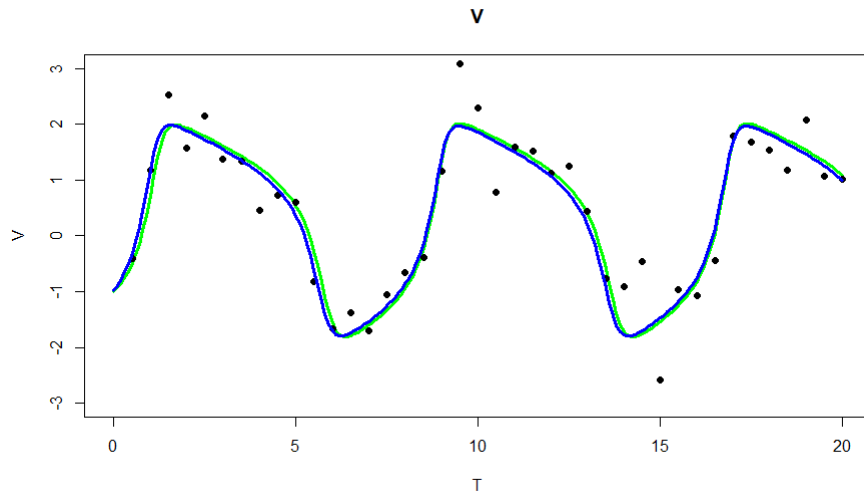


Figura 4.55: Solución media del estado  $V$  y su verdadera solución de la segunda simulación de HMC

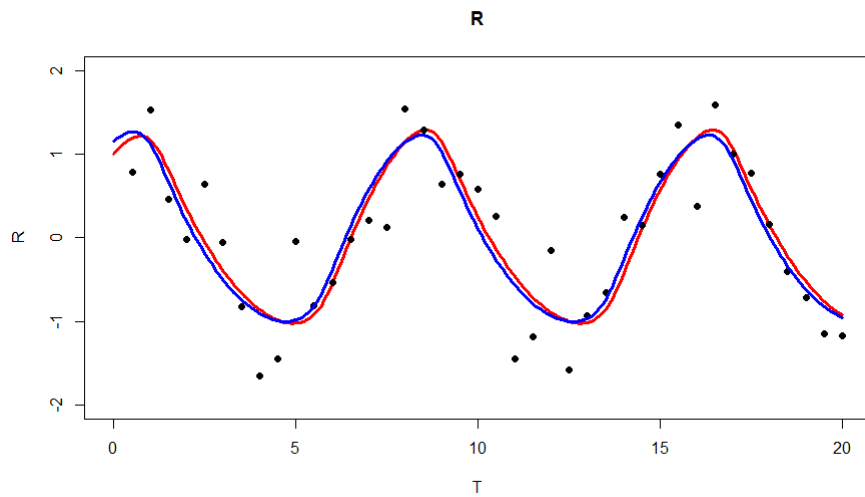


Figura 4.56: Solución media del estado  $R$  y su verdadera solución de la segunda simulación de HMC

Y por último el cálculo de las bandas de probabilidad de la distribución predictiva para nuestra solución estimada (líneas naranjas).

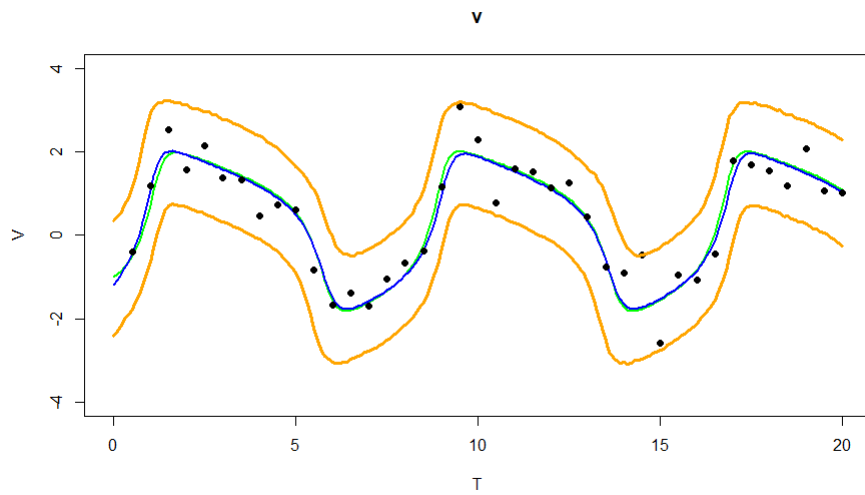


Figura 4.57: Bandas de probabilidad del estado  $V$  de la segunda simulación de HMC

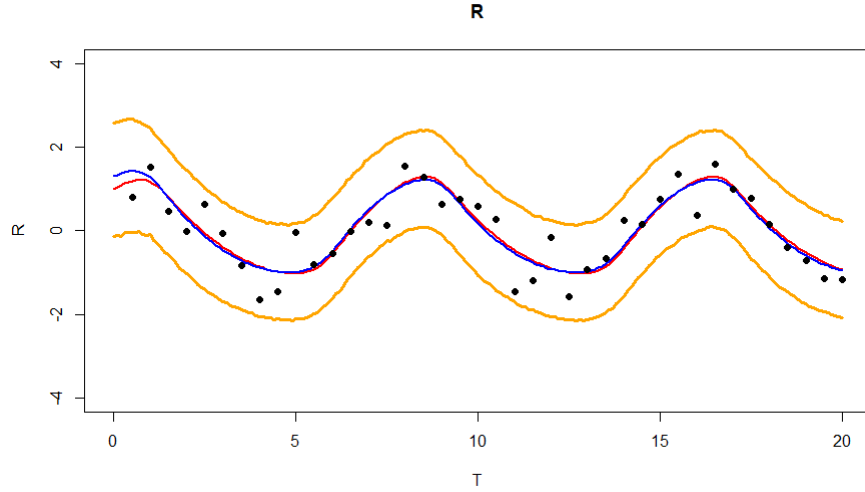


Figura 4.58: Bandas de probabilidad del estado  $R$  de la segunda simulación de HMC

Podemos notar que el paso  $L$  del método Leapfrog mejora la calidad de nuestras estimaciones pero el tiempo de cómputo tiene mayor costo.

### Método adjunto

Para evitar tiempos largos de computo, se propuso utilizar el método adjunto para agilizar el cálculo del gradiente numérico a comparación de seguir utilizando diferencias finitas (Sengupta, 2014). Sea  $lp(\theta)$  mi función de logaritmo de la distribución final, la podemos separar como:

$$lp(\theta) = l(\theta|y) + \ln(p(\theta)).$$

Donde  $l(\theta|y)$  es la función log verosimilitud y  $p(\theta)$  la distribución inicial. Se puede observar que:

$$\nabla_{\theta} lp = \nabla_{\theta} l(\theta|y) + \nabla_{\theta} \ln(p(\theta)).$$

Las derivadas de las iniciales son bastante sencillas de calcular, mientras que para la función log verosimilitud podemos suponer que tenemos datos  $y_1 = (y_{11}, \dots, y_{1n})$  y  $y_2 = (y_{21}, \dots, y_{2n})$ , los cuales se distribuyen  $y_{1i} \sim N(v(\theta, t_i), \sigma_1^2)$  y  $y_{2i} \sim N(r(\theta, t_i), \sigma_2^2)$ , entonces:

$$\begin{aligned} l(\theta|y) = & -\frac{n}{2} \ln(2\pi\sigma_1^2) - \frac{1}{2\sigma_1^2} (y_1 - v(\theta, t))^T (y_1 - v(\theta, t)) \\ & -\frac{n}{2} \ln(2\pi\sigma_2^2) - \frac{1}{2\sigma_2^2} (y_2 - r(\theta, t))^T (y_2 - r(\theta, t)) \end{aligned}$$

Donde el vector  $v(\theta, t)$  y  $r(\theta, t)$  son la solución numérica del sistema de EDO (es decir  $v(\theta, t) = (v(\theta, t_1), \dots, v(\theta, t_n))$ ). Lo que nos interesa es calcular el gradiente de la función log verosimilitud de forma eficiente. Donde si tomamos cualquier parámetro  $a^* = (\alpha, \beta, \gamma, V_0, R_0)$ , entonces:

$$\frac{dl}{da^*} = \frac{\partial l}{\partial a^*} + \frac{\partial l}{\partial v(\theta, t)} \frac{\partial v(\theta, t)}{\partial a^*} + \frac{\partial l}{\partial r(\theta, t)} \frac{\partial r(\theta, t)}{\partial a^*}$$

$$\frac{dl}{da^*} = \frac{1}{\sigma_1^2} \sum_{i=1}^n (y_{1i} - v(\theta, t_i)) \frac{\partial v(\theta, t)}{\partial a^*} + \frac{1}{\sigma_2^2} \sum_{i=1}^n (y_{2i} - r(\theta, t_i)) \frac{\partial r(\theta, t)}{\partial a^*}$$

En esta ecuación se puede notar porque el gradiente de nuestra función log verosimilitud tiene que ser calculado numéricamente, ya que no hay forma analítica de evaluar el termino  $\frac{\partial v(\theta, t)}{\partial a^*}$  y  $\frac{\partial r(\theta, t)}{\partial a^*}$ . Cabe mencionar que para los parámetros  $\sigma_1^2$  y  $\sigma_2^2$  no sucede esto:

$$\frac{dl}{d\sigma_1^2} = -\frac{n}{2} 2\pi \frac{1}{2\pi\sigma_1^2} + \frac{1}{2\sigma_1^4} (y_1 - v(\theta, t))^T (y_1 - v(\theta, t)) = -\frac{n}{2\sigma_1^2} + \frac{1}{2\sigma_1^4} (y_1 - v(\theta, t))^T (y_1 - v(\theta, t)).$$

Análogamente:

$$\frac{dl}{d\sigma_2^2} = -\frac{n}{2\sigma_2^2} + \frac{1}{2\sigma_2^4} (y_2 - r(\theta, t))^T (y_2 - r(\theta, t)).$$

Es por eso, que de aquí en adelante trabajaremos con  $\theta = (\alpha, \beta, \gamma, V_0, R_0)$ . Definamos nuestra función de costo como la función los verosimilitud:

$$J(\theta) = -\frac{n}{2} \ln(2\pi\sigma_1^2) - \frac{1}{2\sigma_1^2} (y_1 - v(\theta, t))^T (y_1 - v(\theta, t))$$

$$- \frac{n}{2} \ln(2\pi\sigma_2^2) - \frac{1}{2\sigma_2^2} (y_2 - r(\theta, t))^T (y_2 - r(\theta, t)).$$

Donde cada entrada de los vectores  $y_1, v(\theta, t), y_2, r(\theta, t)$  dependen del tiempo  $t$ , podemos tomar:

$$J(\theta) = \int_0^T j(x, \theta) dt.$$

A continuación se presenta la demostración de Sengupta (2014) para utilizar el método adjunto para cálculo de derivadas. Tomemos  $x = (v, r)$ :

*Demostración.*

$$j(x, \theta) = -\frac{1}{2} \ln(2\pi\sigma_1^2) - \frac{1}{2\sigma_1^2} (v - v(\theta, t_i))^2 - \frac{1}{2} \ln(2\pi\sigma_2^2) - \frac{1}{2\sigma_2^2} (r - r(\theta, t_i))^2$$

Entonces:

$$L(\theta) = \int_0^T j(x, \theta) dt + \int_0^T \lambda^T [\dot{x} - f(x, \theta)] dt.$$

Si se cumple que  $\dot{x} - f(x, \theta) = 0$  (para esto  $f(x, \theta)$  va a ser nuestro sistemas de ecuaciones diferenciales, lo cual implica que si los puntos calculados son solución del sistema de ecuaciones diferenciales ordinarias) entonces:

$$\frac{dJ}{d\theta} = \frac{dL}{d\theta}.$$

El termino  $\frac{dL}{d\theta}$  se puede calcular de forma eficiente, puesto que:

$$\frac{dL}{d\theta} = \int_0^T \left( \frac{\partial j}{\partial x} \frac{dx}{d\theta} + \frac{\partial j}{\partial \theta} \right) dt + \int_0^T \lambda^T \left( \frac{d\dot{x}}{d\theta} - \frac{\partial f}{\partial x} \frac{dx}{d\theta} - \frac{\partial f}{\partial \theta} \right) dt.$$

Sin embargo, integrando por partes:

$$\int_0^T \lambda^T \frac{d\dot{x}}{d\theta} dt = \lambda^T \frac{dx}{d\theta} \Big|_0^T - \int_0^T \frac{d\lambda^T}{dt} \frac{dx}{d\theta} dt.$$

Sustituyendo en la ecuación anterior y agrupando tenemos:

$$\frac{dL}{d\theta} = \int_0^T \frac{dx}{d\theta} \left( \frac{\partial j}{\partial x} - \lambda^T \frac{\partial f}{\partial x} - \frac{d\lambda^T}{dt} \right) dt + \int_0^T \left( \frac{\partial j}{\partial \theta} - \lambda^T \frac{\partial f}{\partial \theta} \right) dt + \lambda^T \frac{dx}{d\theta} \Big|_0^T.$$

El primer termino se puede eliminar si:

$$\frac{d\lambda^T}{dt} = \frac{\partial j}{\partial x} - \lambda^T \frac{\partial f}{\partial x}.$$

Esta es la ecuación adjunta y se usa para calcular a  $\lambda$ . Por lo tanto el gradiente esta dado por:

$$\frac{dJ}{d\theta} = \int_0^T \left( \frac{\partial j}{\partial \theta} - \lambda^T \frac{\partial f}{\partial \theta} \right) dt + \lambda^T \frac{dx}{d\theta} \Big|_0^T.$$

Y el último termino se puede eliminar integrando la ecuación adjunta hacia atrás en el tiempo, empezando con  $\lambda_T = 0$ . Lo cual final implica que el cálculo del gradiente se reduce a:

$$\frac{dJ}{d\theta} = \int_0^T \left( \frac{\partial j}{\partial \theta} - \lambda^T \frac{\partial f}{\partial \theta} \right) dt.$$

□

Para nuestro modelo FHN, tenemos que la ecuación adjunta vista anteriormente:

$$\dot{\lambda}_i^T = \frac{\partial J}{\partial x_i} - \lambda_i^T \frac{\partial f}{\partial x_i}.$$

Sea  $n$  el número de datos, y la condición inicial  $\lambda(T) = 0$  (por esto es importante integrar hacia atrás). También es importante ver que  $\lambda_i = \lambda(t_i)$ , es decir  $i$  esta relacionado al tiempo  $t_i$ . Esto se puede ver con el siguiente sistema:

$$\begin{pmatrix} \dot{\lambda}_1(t_i) \\ \dot{\lambda}_2(t_i) \end{pmatrix} = \begin{pmatrix} -\gamma + \gamma V(t_i)^2 & \frac{1}{\gamma} \\ -\gamma & \frac{\beta}{\gamma} \end{pmatrix} \begin{pmatrix} \lambda_1(t_i) \\ \lambda_2(t_i) \end{pmatrix} + \begin{pmatrix} \frac{1}{\sigma_1^2} (y_{1i} - V(t_i)) \\ \frac{1}{\sigma_2^2} (y_{2i} - R(t_i)) \end{pmatrix}.$$

Sengupta propone resolver este sistema de ecuaciones de forma numérica, también notemos que:

$$\frac{\partial j}{\partial \theta} = 0$$

$$-\frac{\partial f}{\partial \theta} = - \begin{pmatrix} \frac{df_1}{d\alpha} & \frac{df_1}{d\beta} & \frac{df_1}{d\gamma} & \frac{df_1}{dV_0} & \frac{df_1}{dR_0} \\ \frac{df_2}{d\alpha} & \frac{df_2}{d\beta} & \frac{df_2}{d\gamma} & \frac{df_2}{dV_0} & \frac{df_2}{dR_0} \end{pmatrix} = - \begin{pmatrix} 0 & 0 & (V - \frac{V^3}{3} + R) & 0 & 0 \\ \frac{1}{\gamma} & -\frac{R}{\gamma} & \frac{1}{\gamma^2}(V - \alpha + \beta R) & 0 & 0 \end{pmatrix}.$$

Por lo cual, nuestro cálculo del gradiente se reduce a:

$$\frac{dJ}{d\theta} = \int_0^T \left( \frac{\partial j}{\partial \theta} - \lambda^T \frac{\partial f}{\partial \theta} \right) dt.$$

$$\frac{dJ}{d\theta} = \sum_{i=1}^{n-1} -(t_{i+1} - t_i) \left( (\lambda_{1i} \quad \lambda_{2i}) \begin{pmatrix} 0 & 0 & (V_i - \frac{V_i^3}{3} + R_i) & 0 & 0 \\ \frac{1}{\gamma} & -\frac{R_i}{\gamma} & \frac{1}{\gamma^2}(V_i - \alpha + \beta R_i) & 0 & 0 \end{pmatrix} \right).$$

Recordemos que se puede evitar el último término de  $\lambda_n$  puesto que vale cero por ser nuestra condición inicial del sistema de ecuaciones. El siguiente escenario se realizó con el método adjunto para calcular el gradiente, para 30,000 iteraciones, con  $L=20$  y  $\epsilon=0.0005$  para el método hamiltoniano Monte Carlo solo para los parámetros  $\alpha$ ,  $\beta$ ,  $\gamma$ ,  $\sigma_1^2$  y  $\sigma_2^2$ , obteniendo los siguientes resultados:

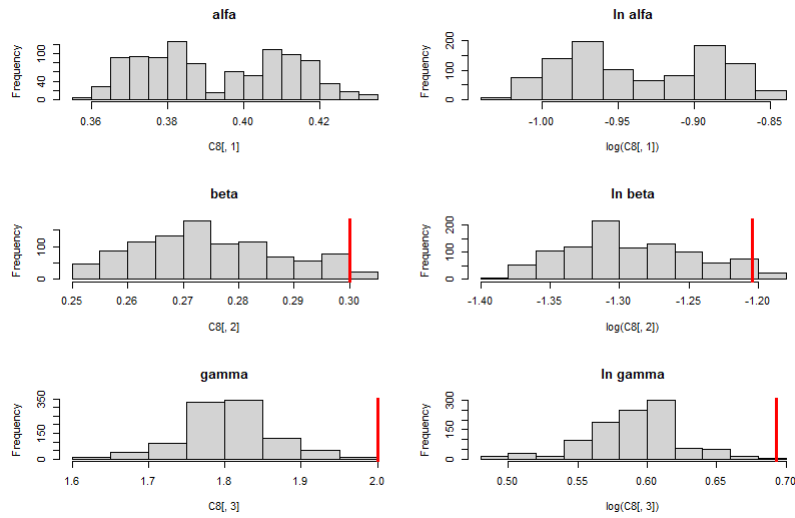


Figura 4.59: Histograma de  $\alpha$ ,  $\beta$  y  $\gamma$  y sus respectivos logaritmos de la simulación utilizando el método adjunto

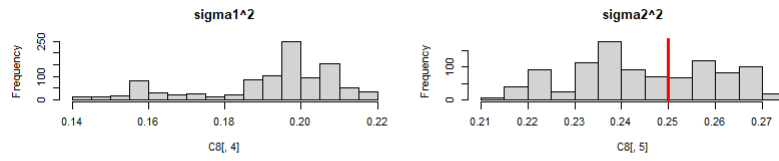


Figura 4.60: Histograma de  $\sigma_1^2$  y  $\sigma_2^2$  de la simulación utilizando el método adjunto

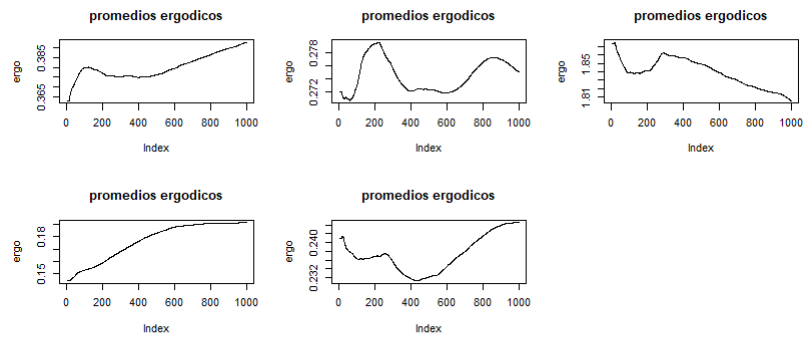


Figura 4.61: Promedios ergódicos de los parámetros  $\alpha$ ,  $\beta$ ,  $\gamma$ ,  $V_0$ ,  $R_0$ ,  $\sigma_1^2$  y  $\sigma_2^2$



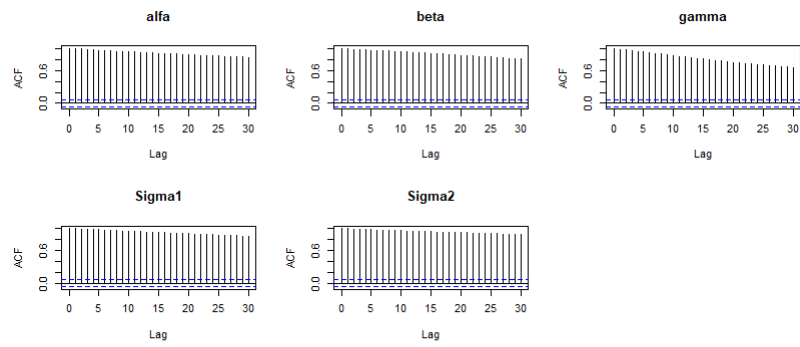


Figura 4.62: Gráficas de auto correlación de los parámetros  $\alpha$ ,  $\beta$ ,  $\gamma$ ,  $V_0$ ,  $R_0$ ,  $\sigma_1^2$  y  $\sigma_2^2$

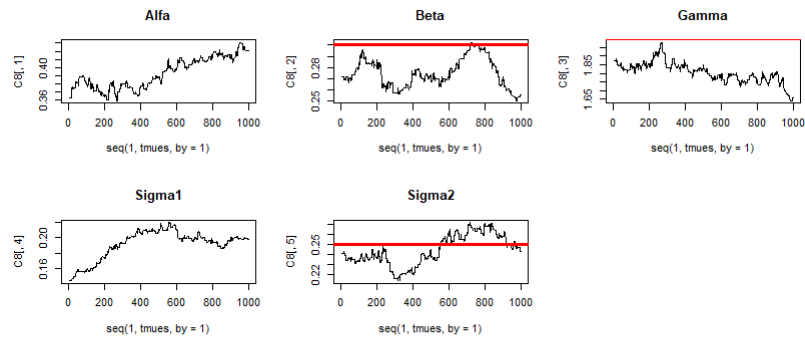


Figura 4.63: Trazas de la muestra final de los parámetros  $\alpha$ ,  $\beta$ ,  $\gamma$ ,  $V_0$ ,  $R_0$ ,  $\sigma_1^2$  y  $\sigma_2^2$

La comparación del verdadero comportamiento del modelos con las soluciones medias, se muestran en las siguientes figuras:

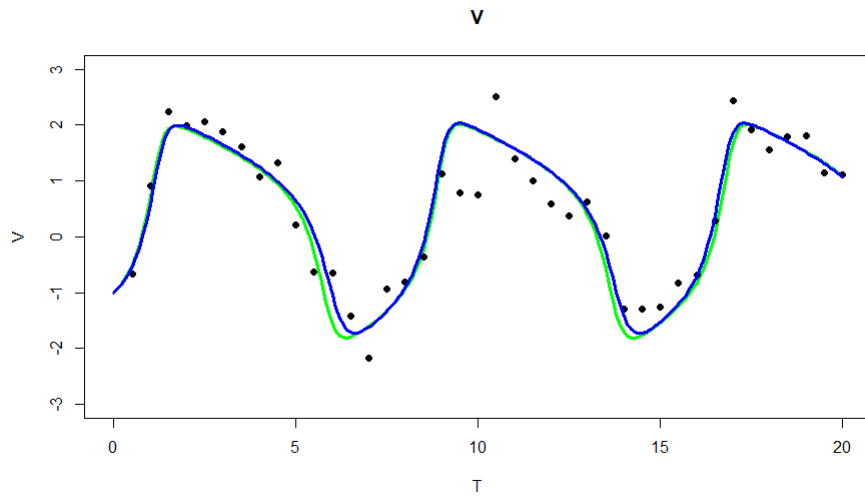


Figura 4.64: Solución media del estado  $V$  y su verdadera solución de la simulación utilizando el método adjunto

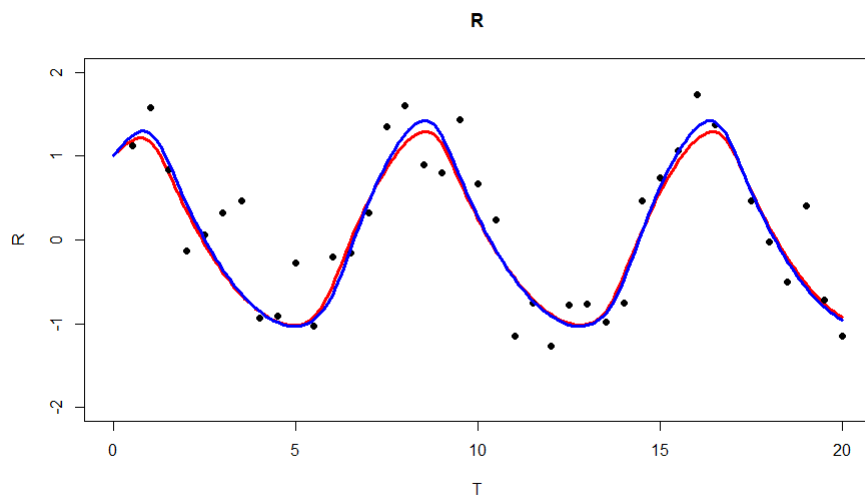


Figura 4.65: Solución media del estado  $R$  y su verdadera solución de la simulación utilizando el método adjunto

Cuadro 4.8: Método adjunto

Parámetro	Verdadero valor	Media	Mediana	Intervalo de prob.
$\alpha$	0.3	0.3928	0.3887	(0.3628 ; 0.4222)
$\beta$	0.3	0.2750	0.2726	(0.2542 ; 0.2992)
$\gamma$	2	1.805	1.806	(1.6758 ; 1.9248)
$\sigma_1^2$	0.25	0.1914	0.1974	(0.1552 ; 0.2176)
$\sigma_2^2$	0.25	0.2446	0.2586	(0.2199 ; 0.2703)

Podemos notar que aunque las estimaciones puntuales del cuadro 4.8 sean buenas, hay un problema en los intervalos de probabilidad que no contienen a los verdaderos valores, lo cual nos lleva a concluir que se requiere mayor estudio para aplicar este método. Similar a todos los escenarios que anteriores, se estiman las bandas de probabilidad,

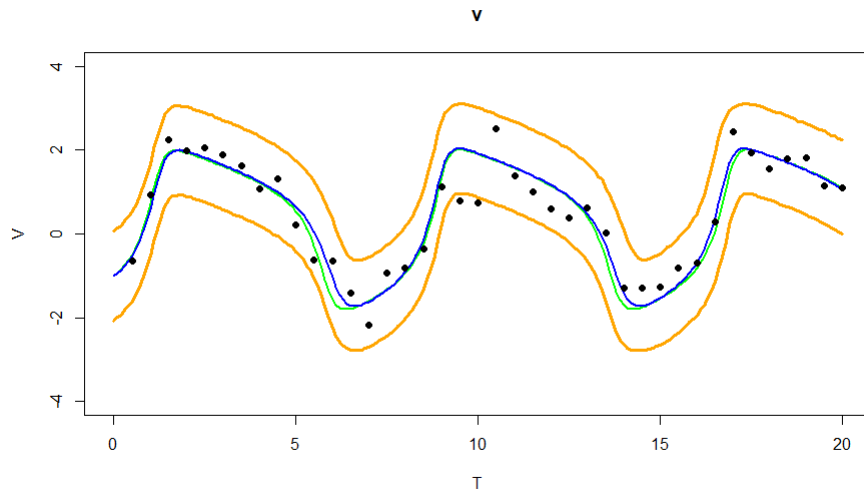


Figura 4.66: Bandas de probabilidad del estado  $V$  de la simulación utilizando el método adjunto

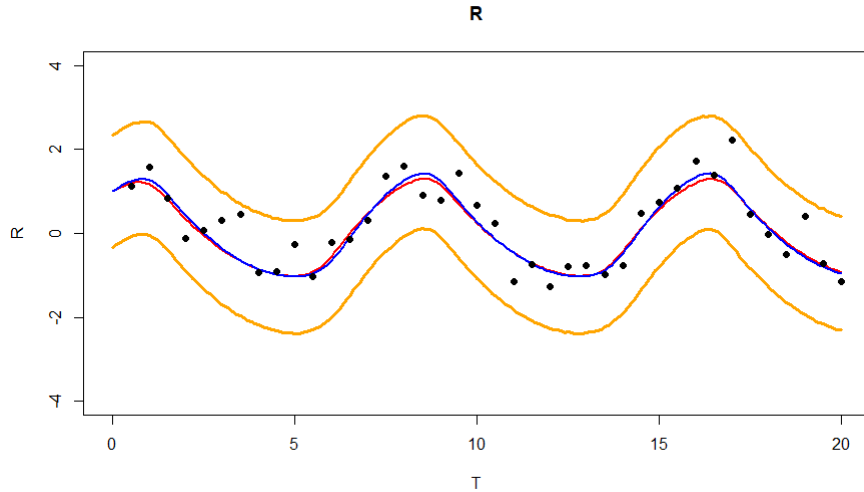


Figura 4.67: Bandas de probabilidad  $R$  de la simulación utilizando el método adjunto

El algoritmo tardó cerca de nueve horas y media, y aunque no se note una mejoría, tenemos que recordar que se cálculo tres veces más el número de iteraciones a los estudios usando diferencias finitas, esto lo podemos ver en la siguiente tabla:

Cuadro 4.9: Comparación HMC

Método cálculo gradiente	Iteraciones	Tiempo de CPU
Diferencias finitas	10,000	10 horas
Método adjunto	30,000	9.6 horas

### 4.1.3. Método ABC

Bajo el método ABC, para la inferencia de parámetros, revisamos dos análisis posibles, primero estimamos los parámetros y los valores iniciales del sistema de ecuaciones diferenciales ordinarias, y como segundo análisis estimamos los parámetros junto el error que existe en los datos. El algoritmo a utilizar es el MCMC-ABC. La distribución propuesta  $q$  es  $N(\theta_{t-1}, I)$ .

#### Estimación de parámetros sin ruido.

Para cada vector de parámetros  $\theta$ , se resolvió el sistema de ecuaciones diferenciales ordinarias para obtener los datos  $z$  y se comparó con los verdaderos datos  $y$  con la siguiente medida:

$$\rho(z, y) = \sum_{j=1}^n (z_j - y_j)^2$$

Esto se realizó para ambos estados  $V$  y  $R$  tal que  $\rho(z, y) < \epsilon$ . Se tomaron las mismas distribuciones iniciales no informativas de los MCMC RAM anteriores y, como primer análisis se tomo el valor de  $\epsilon = 1$ , donde la simulación tardo 10 minutos para

68,000 iteraciones, con 40 datos por estado, obteniendo así los siguientes resultados:

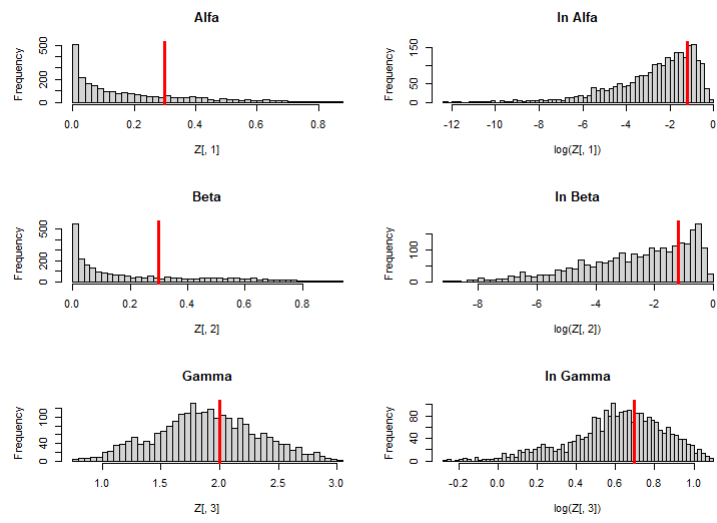


Figura 4.68: Histogramas de  $\alpha$ ,  $\beta$  y  $\gamma$  y sus respectivos logaritmos del escenario  $\epsilon=1$

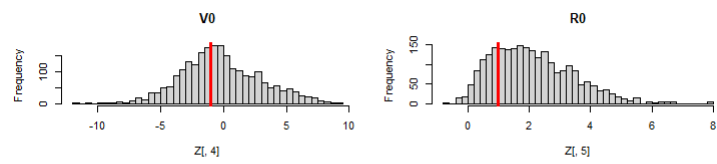


Figura 4.69: Histogramas de  $V_0$  y  $R_0$  del escenario  $\epsilon=1$

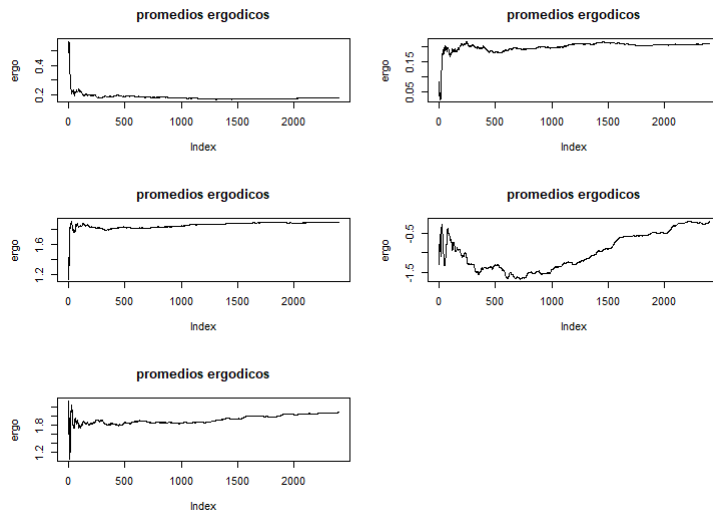


Figura 4.70: Promedios ergódicos de los parámetros  $\alpha$ ,  $\beta$ ,  $\gamma$ ,  $V_0$  y  $R_0$

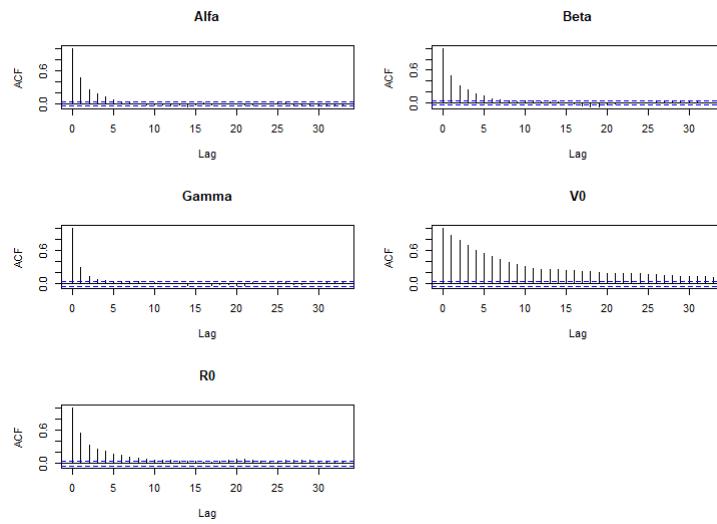


Figura 4.71: Gráficas de auto correlación de los parámetros  $\alpha$ ,  $\beta$ ,  $\gamma$ ,  $V_0$  y  $R_0$

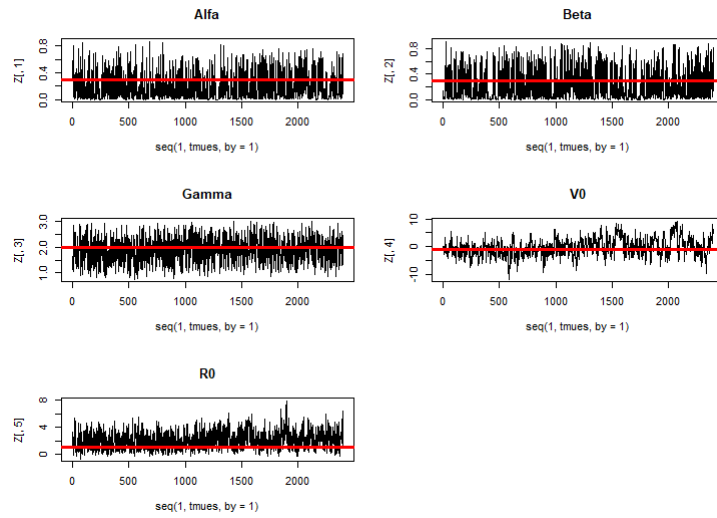


Figura 4.72: Trazas de la muestra final de los parámetros  $\alpha$ ,  $\beta$ ,  $\gamma$ ,  $V_0$  y  $R_0$

La siguiente tabla que muestra nuestros resultados numéricos:

Cuadro 4.10:  $\epsilon=1$

Parámetro	Verdadero valor	Media	Mediana	Intervalo de Prob.
$\alpha$	0.3	0.1803	0.1120	$(4.64e^{-06} ; 0.5759)$
$\beta$	0.3	0.2127	0.1148	$(0.0001 ; 0.6782)$
$\gamma$	2	1.8894	1.8804	$(1.0456 ; 2.6942)$
$V_0$	-1	-0.2155	-0.4904	$(-6.026 ; 6.3164)$
$R_0$	1	2.0858	1.9276	$(0.0165 ; 4.6337)$

Se comparo la solución media (líneas azul) con las verdaderas soluciones (líneas verde y roja respectivamente) y los datos:

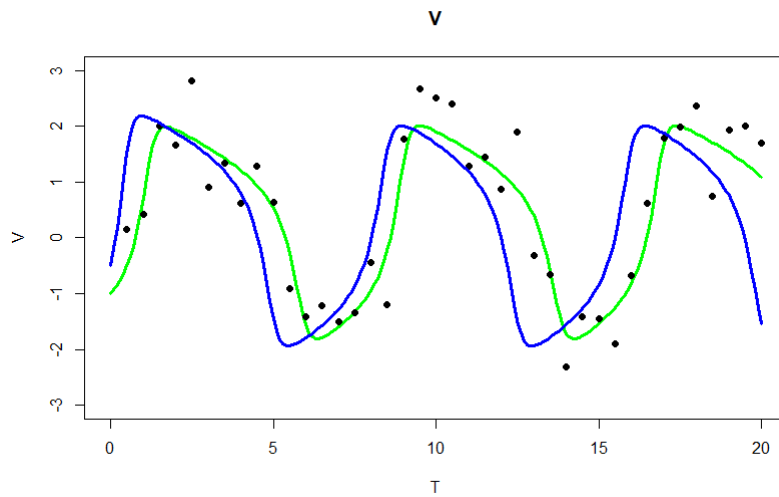


Figura 4.73: Solución media del estado  $V$  y su verdadera solución del escenario  $\epsilon=1$

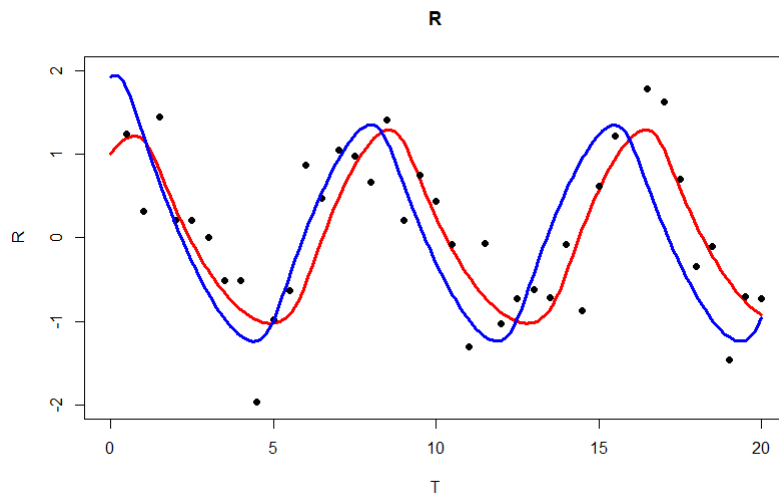


Figura 4.74: Solución media del estado  $R$  y su verdadera solución del escenario  $\epsilon=1$

Para un segundo análisis se tomó el valor de  $\epsilon=0.5$ , con tiempo de simulación de 11 minutos, obteniendo los siguientes resultados:



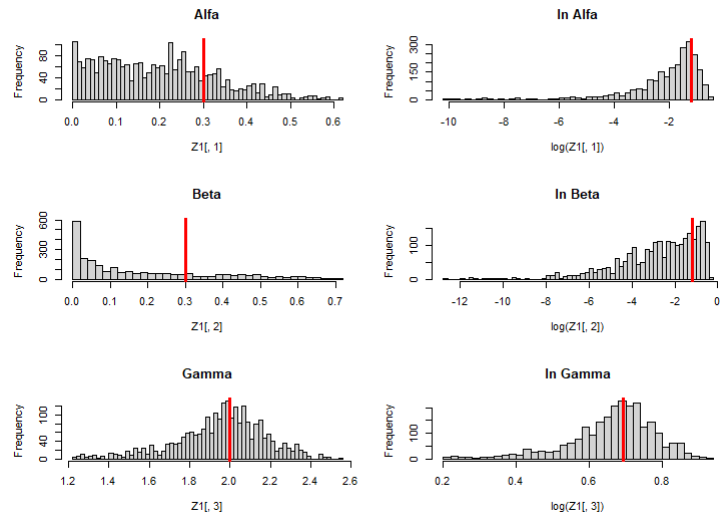


Figura 4.75: Histogramas de  $\alpha$ ,  $\beta$  y  $\gamma$  y sus respectivos logaritmos del escenario  $\epsilon=0.5$

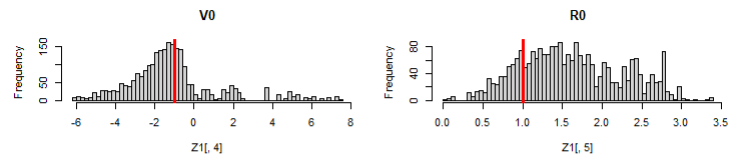


Figura 4.76: Histogramas de  $V_0$  y  $R_0$  del escenario  $\epsilon=0.5$

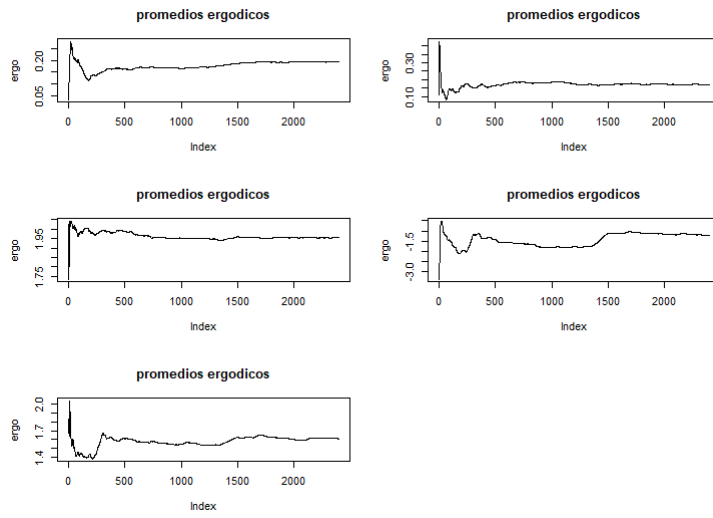


Figura 4.77: Promedios ergódicos de los parámetros  $\alpha$ ,  $\beta$ ,  $\gamma$ ,  $V_0$  y  $R_0$

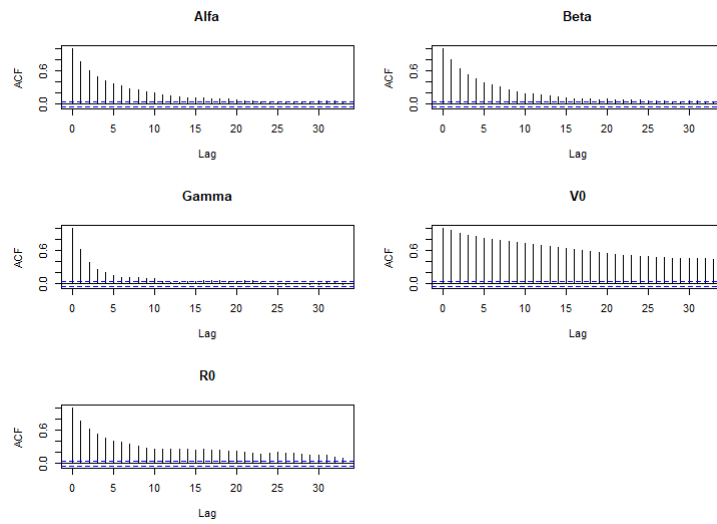


Figura 4.78: Gráficas de auto correlación de los parámetros  $\alpha$ ,  $\beta$ ,  $\gamma$ ,  $V_0$  y  $R_0$

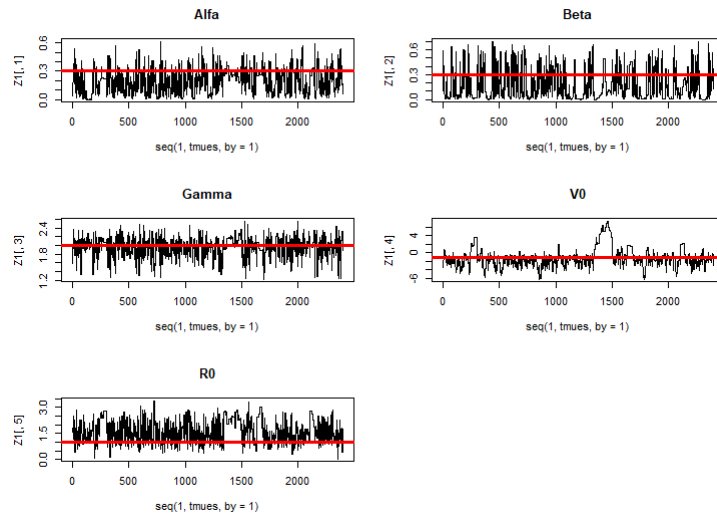


Figura 4.79: Trazas de la muestra final de los parámetros  $\alpha$ ,  $\beta$ ,  $\gamma$ ,  $V_0$  y  $R_0$

Con la siguiente tabla de resultados:

Cuadro 4.11:  $\epsilon=0.5$

Parámetro	Verdadero valor	Media estimada	Mediana	Intervalo de Prob.
$\alpha$	0.3	0.1940	0.1866	(0.00005 ; 0.4313)
$\beta$	0.3	0.1704	0.0964	( $3.291e^{-06}$ ; 0.5435)
$\gamma$	2	1.955	1.978	(1.511 ; 2.38)
$V_0$	-1	-1.2068	-1.4278	(-5.9 ; 3.673)
$R_0$	1	1.608	1.5395	(0.5703 ; 2.798)

Se comparo la solución media (líneas azul) con las verdaderas soluciones (líneas verde y roja respectivamente) y los datos:

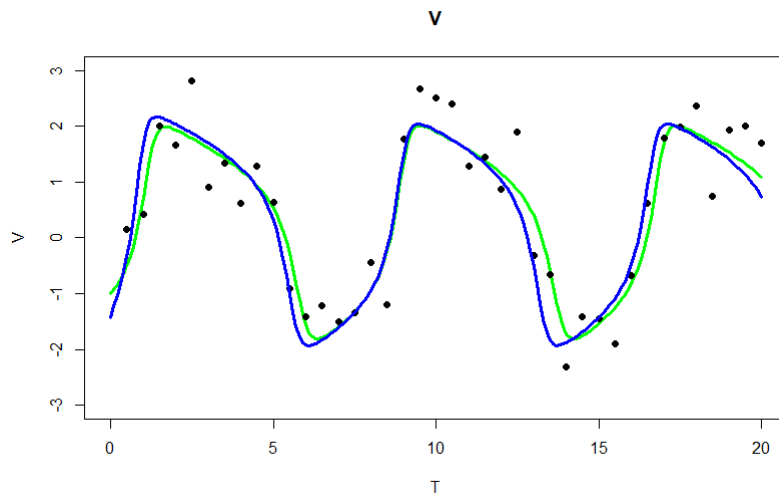


Figura 4.80: Solución media del estado  $V$  y su verdadera solución del escenario  $\epsilon=0.5$

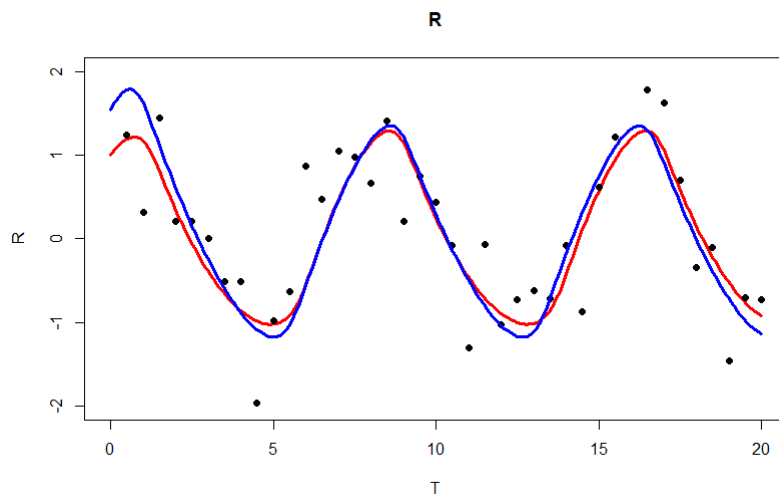


Figura 4.81: Solución media del estado  $R$  y su verdadera solución del escenario  $\epsilon=0.5$

Se puede observar, como en el segundo análisis mejoran los intervalos de probabilidad. Sin embargo, si seguimos tomando valores de  $\epsilon$  más pequeños podemos correr el riesgo de obtener tasas de aceptación muy bajas o incluso no poder muestrear adecuadamente los parámetros. Como ejemplo de esto, si resolvemos el sistema de EDO con los verdaderos valores y lo comparamos a nuestros datos, obtenemos que nuestra función  $\rho(z^*, y)$  toma un valor mayor a 0.3. Es decir, si tomáramos nuestro  $\epsilon=0.3$  el algoritmo no funcionaría. Como tercer análisis, investigamos el impacto que tienen las distribuciones iniciales en las estimaciones, así que se realizó la simulación con  $\epsilon=0.5$  con iniciales más informativas, con tiempo de 7 minutos. Las distribuciones iniciales consideradas fueron:

- $\ln(\alpha) \sim N(\ln(0.5), 1)$

- $\ln(\beta) \sim N(\ln(0.5), 1)$
- $\ln(\gamma) \sim N(\ln(3), 1)$
- $V_0 \sim N(0, 1)$
- $R_0 \sim N(0, 1)$

Los resultados obtenidos se muestran a continuación.

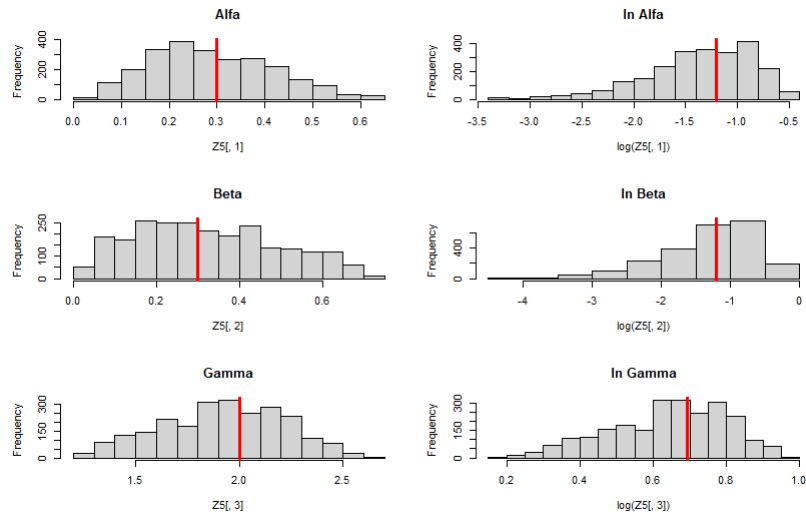


Figura 4.82: Histogramas de  $\alpha$ ,  $\beta$  y  $\gamma$  y sus respectivos logaritmos del escenario  $\epsilon=0.5$  con distribuciones iniciales informativas

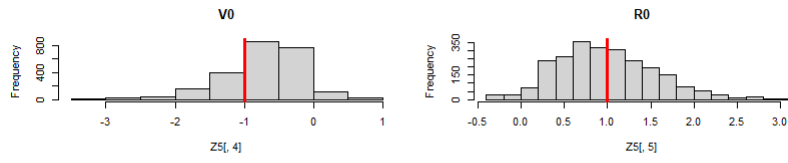


Figura 4.83: Histogramas de  $V_0$  y  $R_0$  del escenario  $\epsilon=0.5$  con distribuciones iniciales informativas

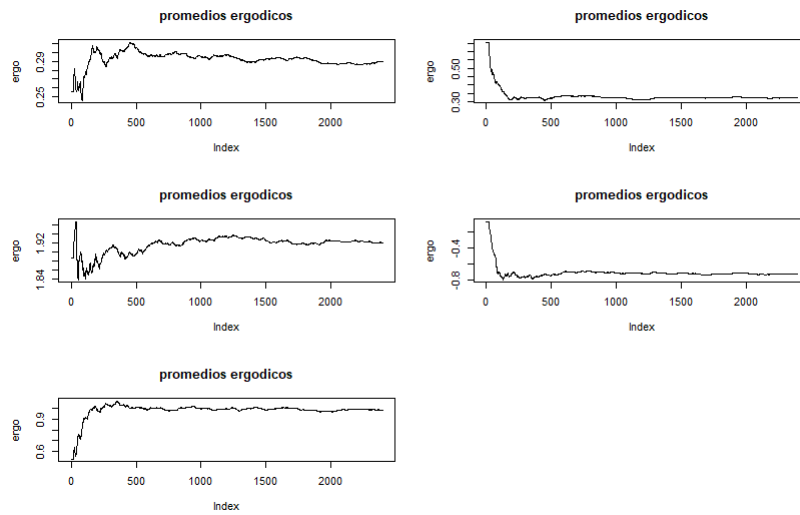


Figura 4.84: Promedios ergódicos de los parámetros  $\alpha$ ,  $\beta$ ,  $\gamma$ ,  $V_0$  y  $R_0$

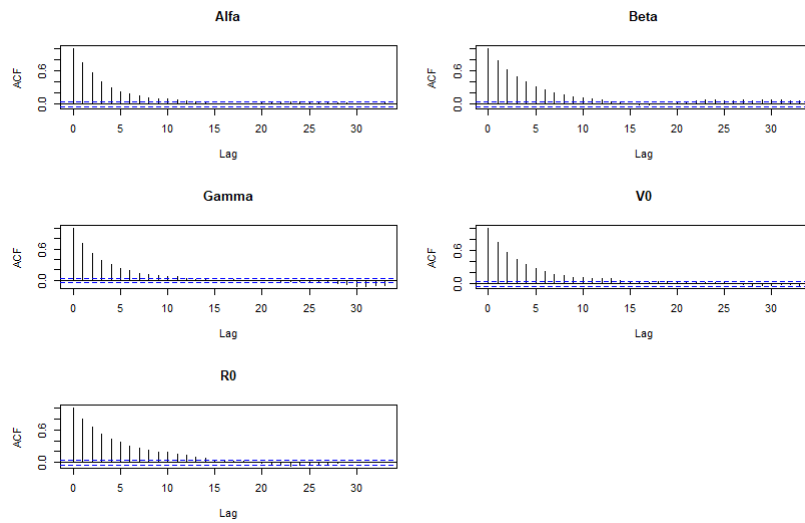


Figura 4.85: Gráficas de auto correlación de los parámetros  $\alpha$ ,  $\beta$ ,  $\gamma$ ,  $V_0$  y  $R_0$

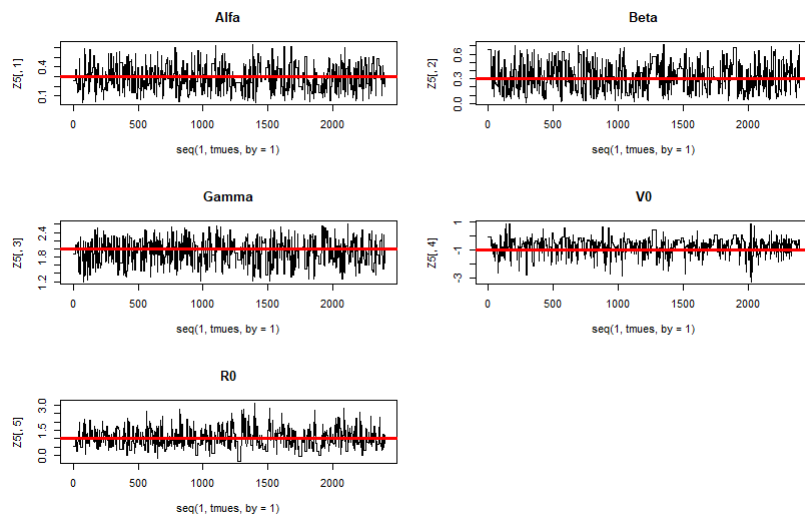


Figura 4.86: Trazas de la muestra final de los parámetros  $\alpha$ ,  $\beta$ ,  $\gamma$ ,  $V_0$  y  $R_0$

Con la siguiente tabla de resultados numéricos:

Cuadro 4.12:  $\epsilon=0.5$  con iniciales informativas

Parámetro	Verdadero valor	Media	Mediana	Intervalo de Prob.
$\alpha$	0.3	0.2897	0.2718	(0.0542 ; 0.5113)
$\beta$	0.3	0.3251	0.304	(0.039 ; 0.63)
$\gamma$	2	1.921	1.932	(1.365 ; 2.457)
$V_0$	-1	-0.7156	-0.6288	(-2.127 ; 0.0777)
$R_0$	1	0.9829	0.9266	(-0.0668 ; 2.0511)

Se comparó la solución media (líneas azul) con las verdaderas soluciones (líneas verde y roja respectivamente) y los datos:

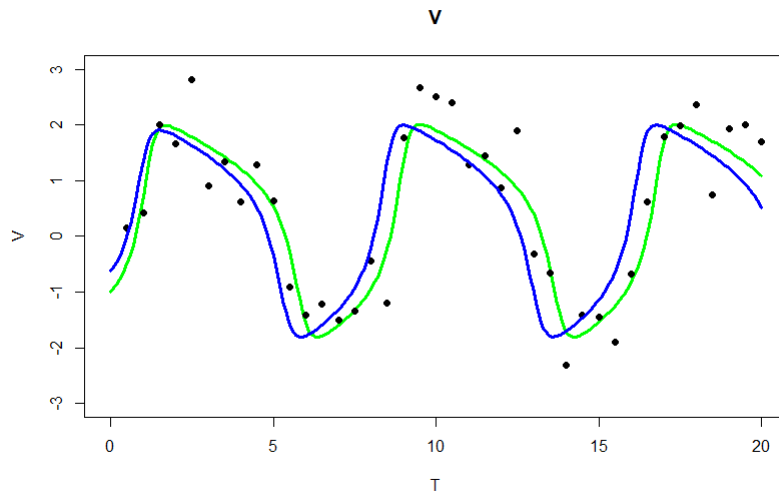


Figura 4.87: Solución media del estado  $V$  y su verdadera solución del escenario  $\epsilon=0.5$  con distribuciones iniciales informativas

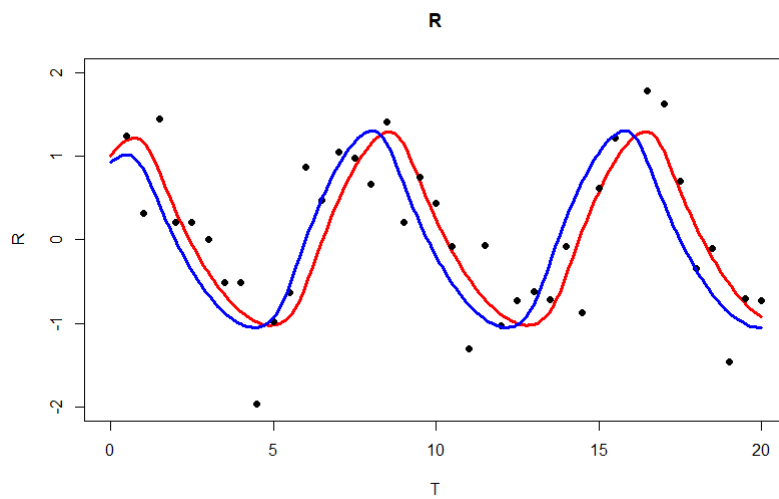


Figura 4.88: Solución media del estado  $R$  y su verdadera solución del escenario  $\epsilon=0.5$  con distribuciones iniciales informativas



Se puede observar que los métodos ABC dependen mucho del valor  $\epsilon$  que pongamos, así como de las distribuciones iniciales, si estas no se eligen con cuidado podemos tener tasas de aceptación muy bajas, lo cual vuelve al método ineficiente.

### Estimación de parámetros con ruido.

Para poder estimar los parámetros  $\sigma_1$  y  $\sigma_2$  con este método, lo que se realizó fue que al simular los datos  $z$  se agregó un error normal con la varianza como parámetro a estimar. Es decir,

$$z_V := z_V + N(0, \theta_6)$$

$$z_R := z_R + N(0, \theta_7)$$

Como primer caso se realizó la simulación con las siguientes distribuciones iniciales informativas:

- $\ln(\alpha) \sim N(\ln(0.5), 1)$
- $\ln(\beta) \sim N(\ln(0.5), 1)$
- $\ln(\gamma) \sim N(\ln(3), 1)$
- $V_0 \sim N(0, 1)$
- $R_0 \sim N(0, 1)$
- $\ln(\sigma_1^2) \sim N(\ln(0.4), 1)$
- $\ln(\sigma_2^2) \sim N(\ln(0.4), 1)$

Con  $\epsilon=0.5$  y con un tiempo de 4 minutos, nuestros resultados fueron los siguientes, con 100,000 + 300 de lag de iteraciones para tamaños de muestra 300 por *core*.

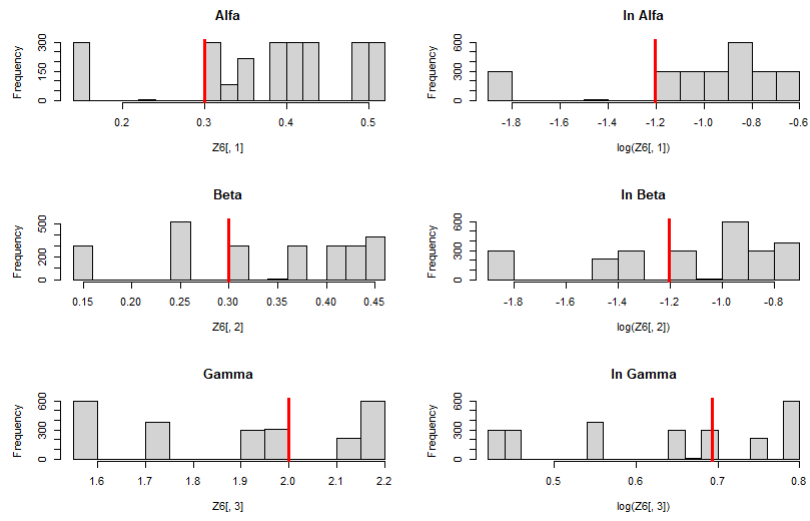


Figura 4.89: Histogramas de  $\alpha$ ,  $\beta$  y  $\gamma$  y sus respectivos logaritmos del escenario  $\epsilon=0.5$  con estimación de la variabilidad

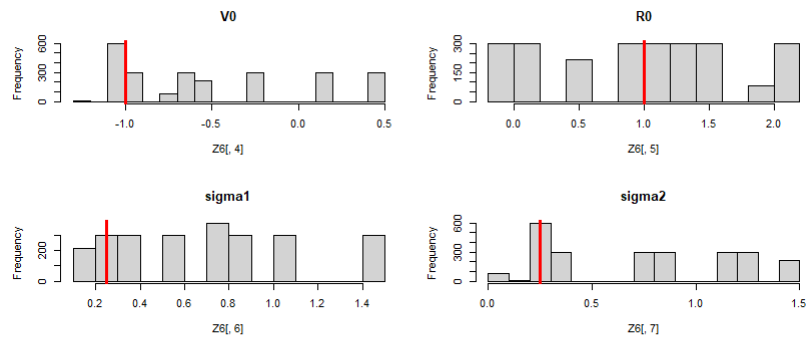


Figura 4.90: Histogramas de  $V_0$ ,  $R_0$ ,  $\sigma_1^2$  y  $\sigma_2^2$  del escenario  $\epsilon=0.5$  con estimación de la variabilidad

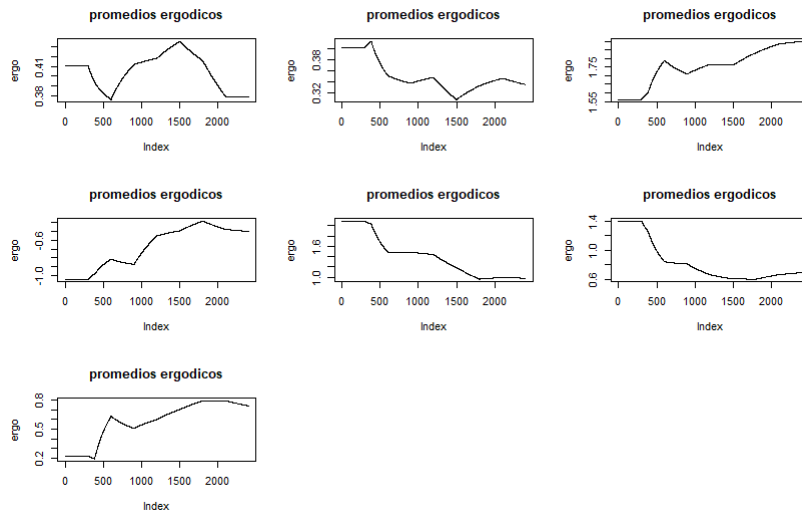


Figura 4.91: Promedios ergódicos de los parámetros  $\alpha$ ,  $\beta$ ,  $\gamma$ ,  $V_0$ ,  $R_0$ ,  $\sigma_1^2$  y  $\sigma_2^2$

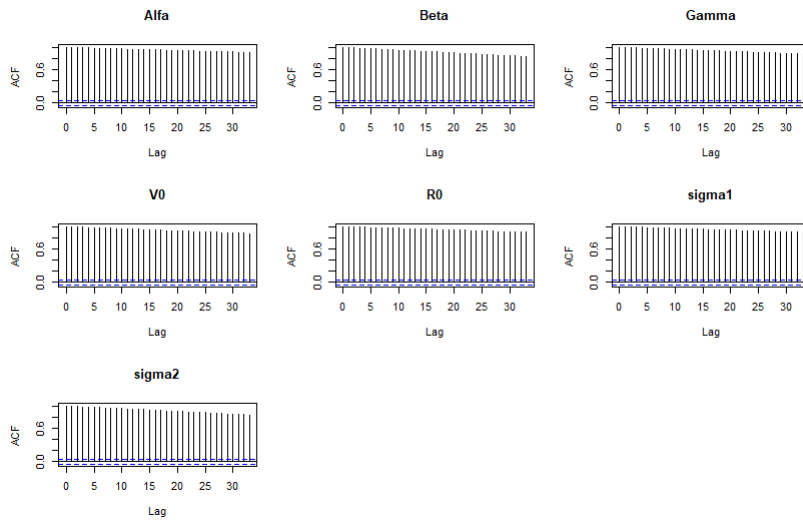


Figura 4.92: Gráficas de auto correlación de los parámetros  $\alpha$ ,  $\beta$ ,  $\gamma$ ,  $V_0$ ,  $R_0$ ,  $\sigma_1^2$  y  $\sigma_2^2$

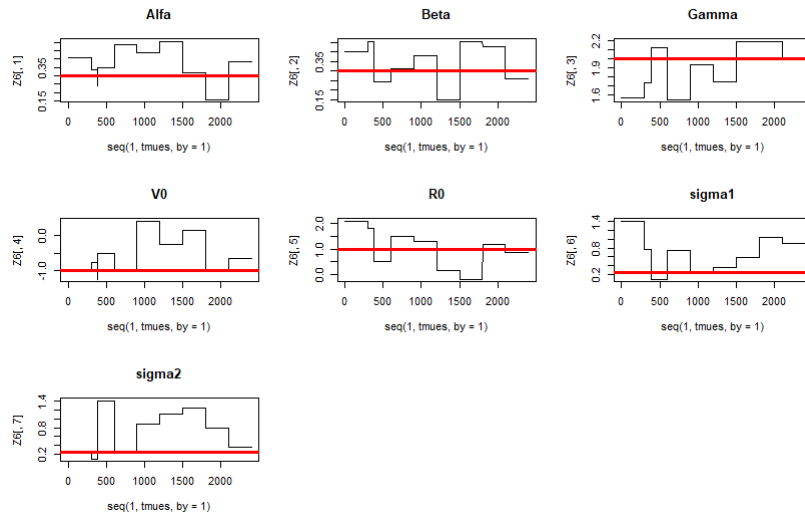


Figura 4.93: Trazas de la muestra final de los parámetros  $\alpha$ ,  $\beta$ ,  $\gamma$ ,  $V_0$ ,  $R_0$ ,  $\sigma_1^2$  y  $\sigma_2^2$

Con la siguiente tabla de resultados:

Cuadro 4.13:  $\epsilon=0.5$  con iniciales informativas y  $\sigma^2$

Parámetro	Verdadero valor	Media estimada	Mediana	Intervalo de Prob.
$\alpha$	0.3	0.3793	0.3961	(0.1583 ; 0.5045)
$\beta$	0.3	0.3355	0.3799	(0.151 ; 0.455)
$\gamma$	2	1.899	1.934	(1.55 ; 2.195)
$V_0$	-1	-0.5025	-0.6659	(-1.0423 ; 0.4169)
$R_0$	1	0.9775	1.1886	(-0.1717 ; 2.0899)
$\sigma_1$	0.5	0.6938	0.749	(0.102 ; 1.4)
$\sigma_2$	0.5	0.7405	0.7974	(0.218 ; 1.415)

Se comparo la solución media (líneas azul) con las verdaderas soluciones (líneas verde y roja respectivamente) y los datos:

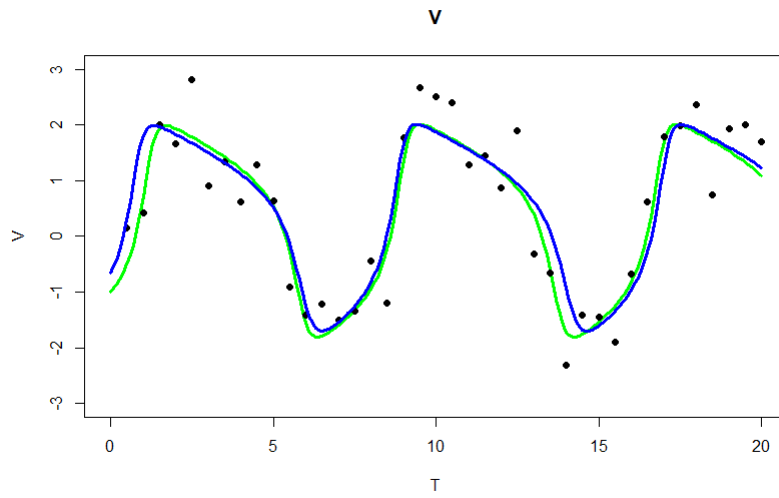


Figura 4.94: Solución media del estado  $V$  y su verdadera solución del escenario  $\epsilon=0.5$  con estimación de la variabilidad

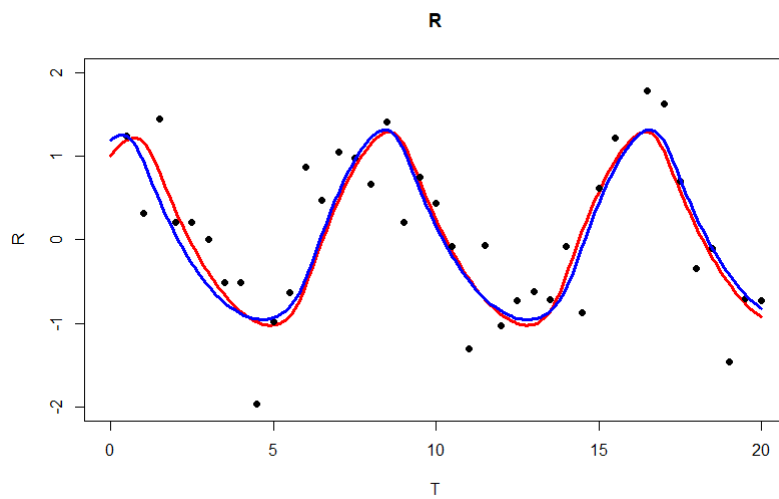


Figura 4.95: Solución media del estado  $R$  y su verdadera solución del escenario  $\epsilon=0.5$  con estimación de la variabilidad

Se puede observar que el agregar un error a los datos simulados no es suficiente para la estimación correcta con el método ABC MCMC, esto requiere de un mayor estudio para mejorar las estimaciones de los errores con métodos ABC (lo cual es complicado en la literatura actualmente (Amani A. Alahmadi, 2020)). Sin embargo, como ya tenemos estimaciones de  $\sigma_1$  y  $\sigma_2$ , podemos calcular las bandas de probabilidad para ambos estados:

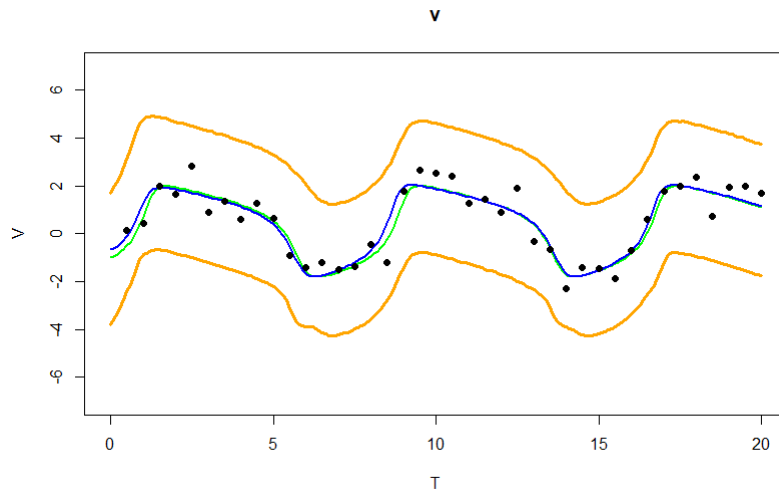


Figura 4.96: Bandas de probabilidad del estado  $V$  del escenario  $\epsilon=0.5$  con estimación de la variabilidad

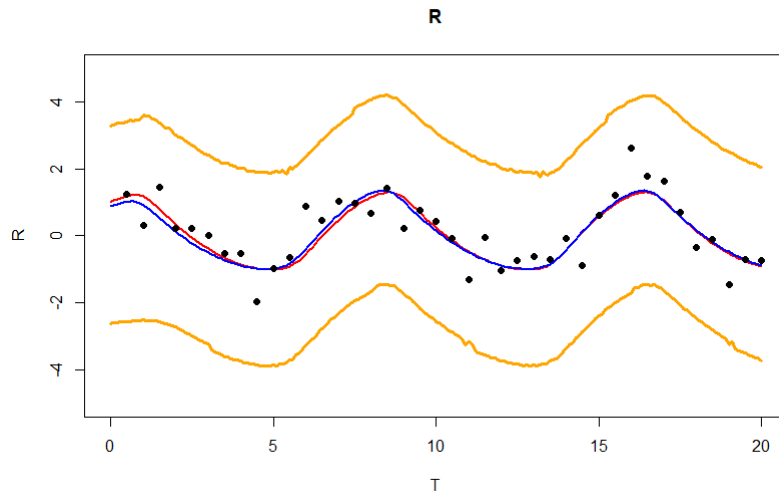


Figura 4.97: Bandas de probabilidad del estado  $R$  del escenario  $\epsilon=0.5$  con estimación de la variabilidad

Como segundo caso, se tomó  $\epsilon=0.5$  y se consideraron iniciales centradas en otro punto con mayor varianza (es decir, no informativas) con el doble de iteraciones (200,000 y 500 de lag) para tamaños de muestra 200 por *core*, y donde nuestra distribución propuesta fue  $N(\theta_{t-1}, 1.5 I)$ . Con un tiempo de 33 minutos, obtuvimos los siguientes resultados:

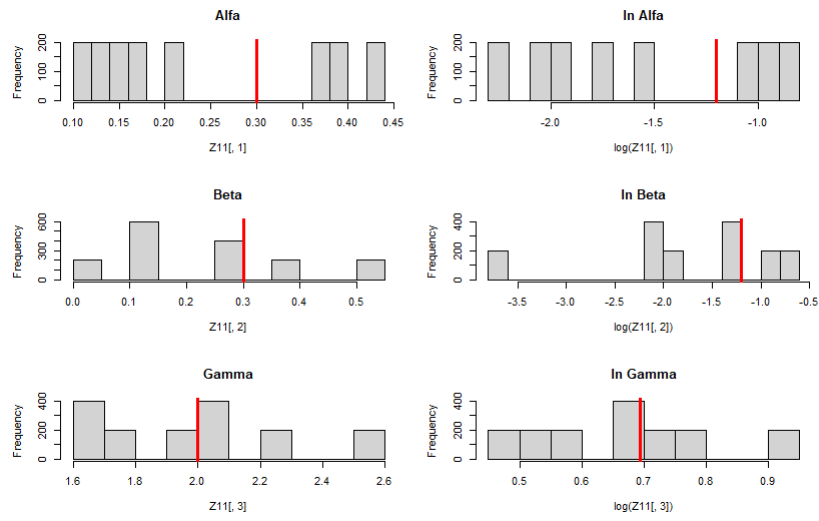


Figura 4.98: Histogramas de  $\alpha$ ,  $\beta$  y  $\gamma$  y sus respectivos logaritmos del escenario  $\epsilon=0.5$  con estimación de la variabilidad y distribuciones iniciales lejanas

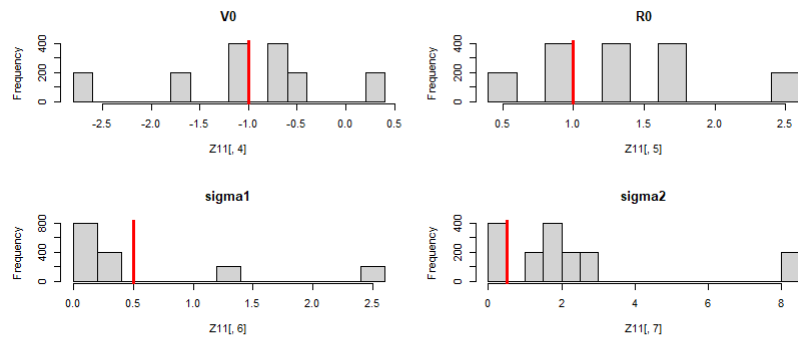


Figura 4.99: Histogramas de  $V_0$ ,  $R_0$ ,  $\sigma_1^2$  y  $\sigma_2^2$  del escenario  $\epsilon=0.5$  con estimación de la variabilidad y distribuciones iniciales lejanas



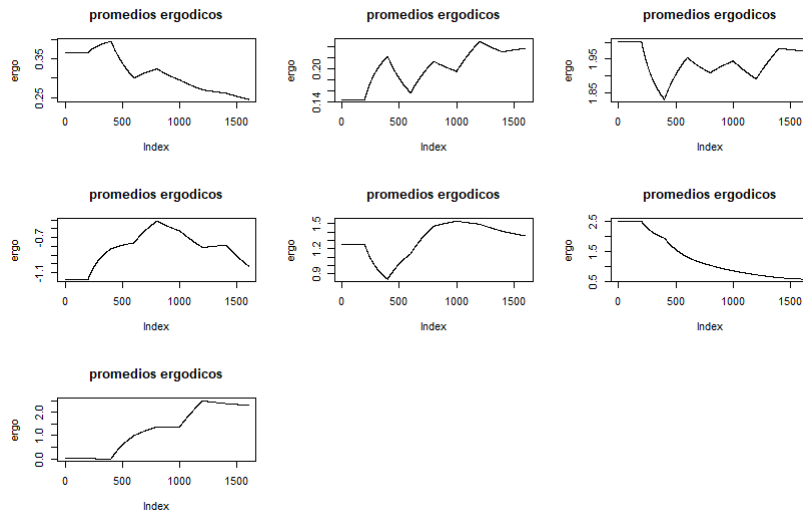


Figura 4.100: Promedios ergódicos de los parámetros  $\alpha$ ,  $\beta$ ,  $\gamma$ ,  $V_0$ ,  $R_0$ ,  $\sigma_1^2$  y  $\sigma_2^2$

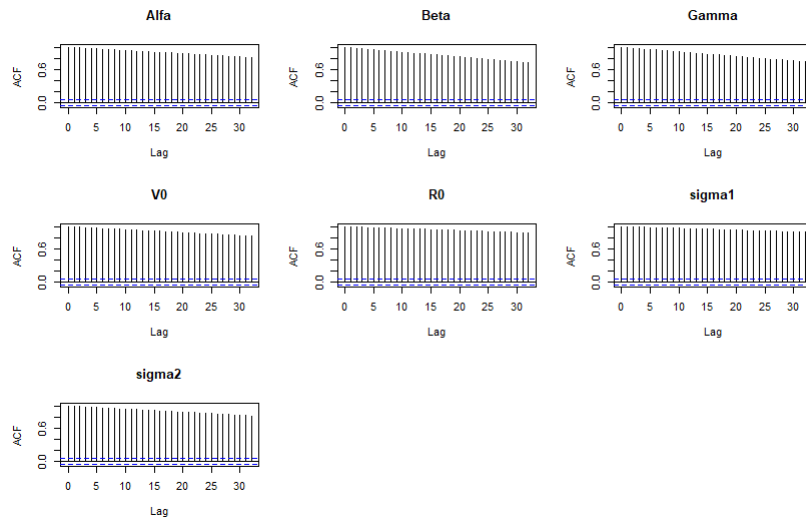


Figura 4.101: Gráficas de auto correlación de los parámetros  $\alpha$ ,  $\beta$ ,  $\gamma$ ,  $V_0$ ,  $R_0$ ,  $\sigma_1^2$  y  $\sigma_2^2$

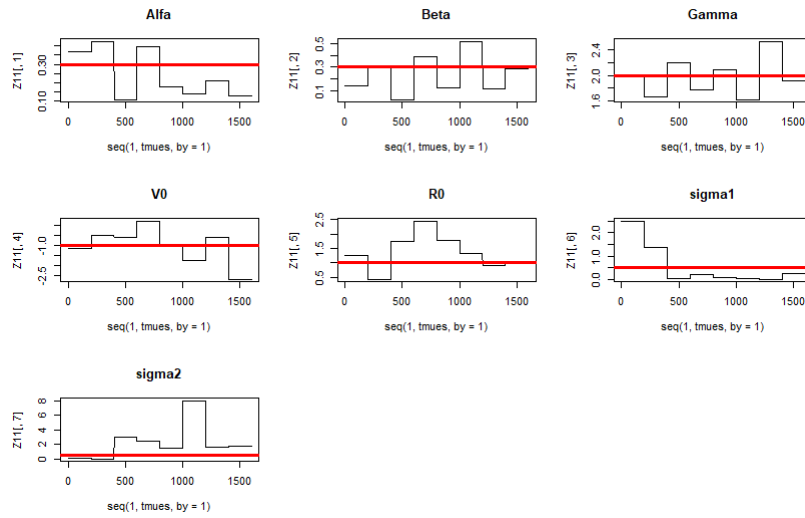


Figura 4.102: Trazas de la muestra final de los parámetros  $\alpha$ ,  $\beta$ ,  $\gamma$ ,  $V_0$ ,  $R_0$ ,  $\sigma_1^2$  y  $\sigma_2^2$

Con la siguiente tabla de resultados:

Cuadro 4.14:  $\epsilon = 0.5$  y  $\sigma^2$  y menos varianza propuesta

Parámetro	Verdadero valor	Media	Mediana	Intervalo de Prob.
$\alpha$	0.3	0.2446	0.1937	(0.1104 ; 0.4244)
$\beta$	0.3	0.2366	0.2131	(0.0233 ; 0.5204)
$\gamma$	2	1.974	1.962	(1.6218 ; 2.5311)
$V_0$	-1	-0.8456	-1.0252	(-2.6953 ; 0.2238)
$R_0$	1	1.3527	1.282	(0.4288 ; 2.45)
$\sigma_1$	0.5	0.5681	0.164	(0.0240 ; 2.5005)
$\sigma_2$	0.5	2.2974	1.7235	(0.0043 ; 8.0366)

Se comparo la solución media (líneas azul) con las verdaderas soluciones (líneas verde y roja respectivamente) y los datos:

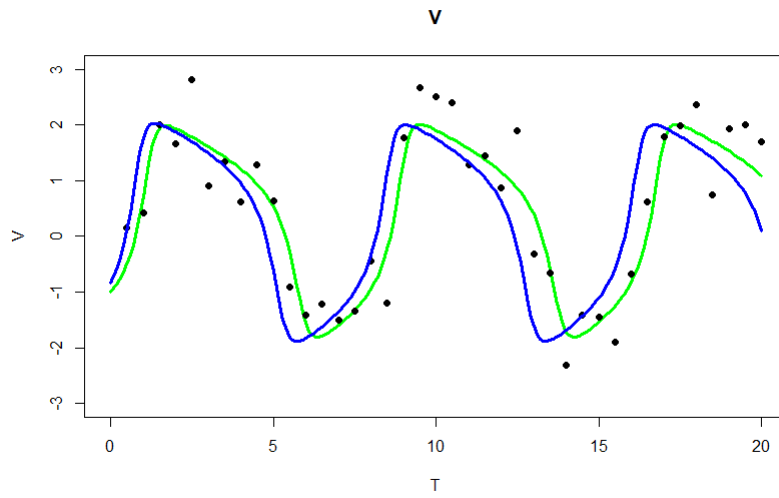


Figura 4.103: Solución media del estado  $V$  y su verdadera solución del escenario  $\epsilon=0.5$  con estimación de la variabilidad y distribuciones iniciales lejanas

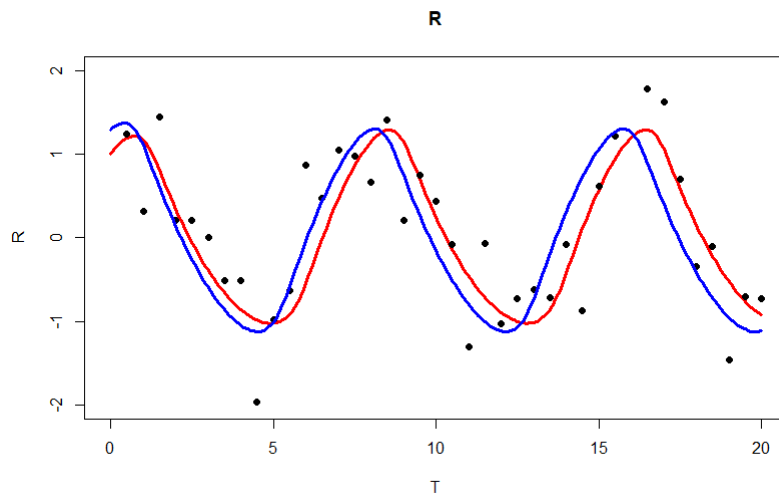


Figura 4.104: Solución media del estado  $R$  y su verdadera solución del escenario  $\epsilon=0.5$  con estimación de la variabilidad y distribuciones iniciales lejanas

Se calcularon las bandas de probabilidad para ambos estados:

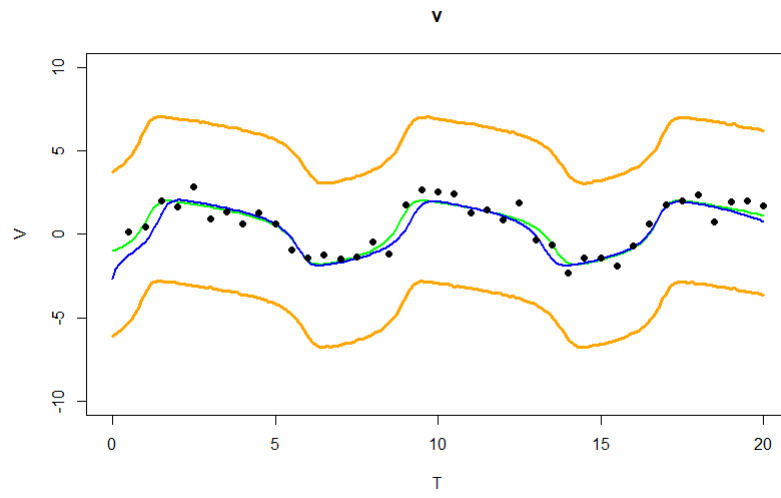


Figura 4.105: Bandas de probabilidad del estado  $V$  del escenario  $\epsilon=0.5$  con estimación de la variabilidad y distribuciones iniciales lejanas

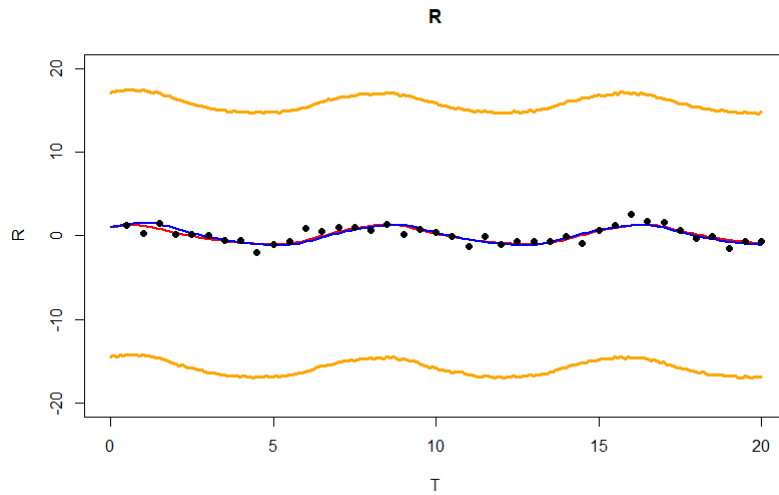


Figura 4.106: Bandas de probabilidad del estado  $R$  del escenario  $\epsilon=0.5$  con estimación de la variabilidad y distribuciones iniciales lejanas

#### 4.1.4. Comparación

Por último, comparamos los diferentes resultados para todos los métodos MCMC utilizados para este modelo con 40 datos. Lo anterior considerando tanto su tiempo de cómputo como el número de iteraciones, para todos los parámetros  $\theta = (\alpha, \beta, \gamma, V_0, R_0, \sigma_1, \sigma_2)$ . Las figuras 4.107 y 4.108 muestra la verdadera solución del sistema, así como, la solución media obtenida bajo los diferentes procedimientos analizados.

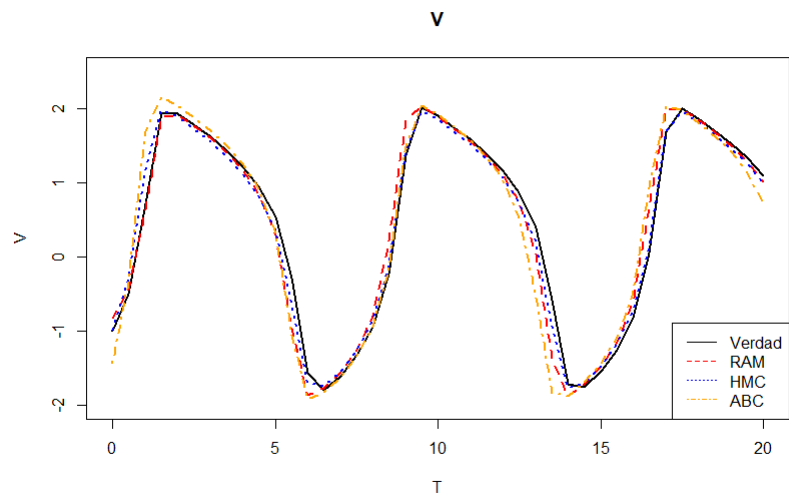


Figura 4.107: Comparación soluciones medias de los distintos métodos MCMC en el estado  $V$

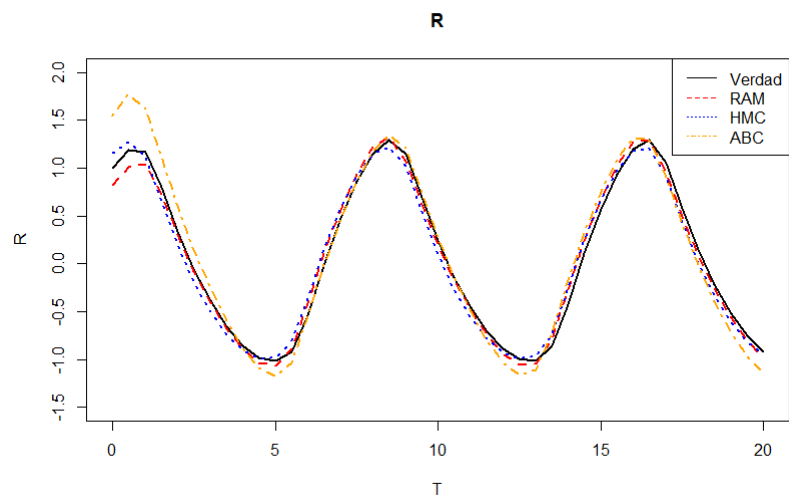


Figura 4.108: Comparación soluciones medias de los distintos métodos MCMC en el estado  $R$

Al igual que con las soluciones medias, las figuras 4.109 y 4.110 muestran una comparación de las bandas de probabilidad obtenidas bajo los tres métodos MCMC:

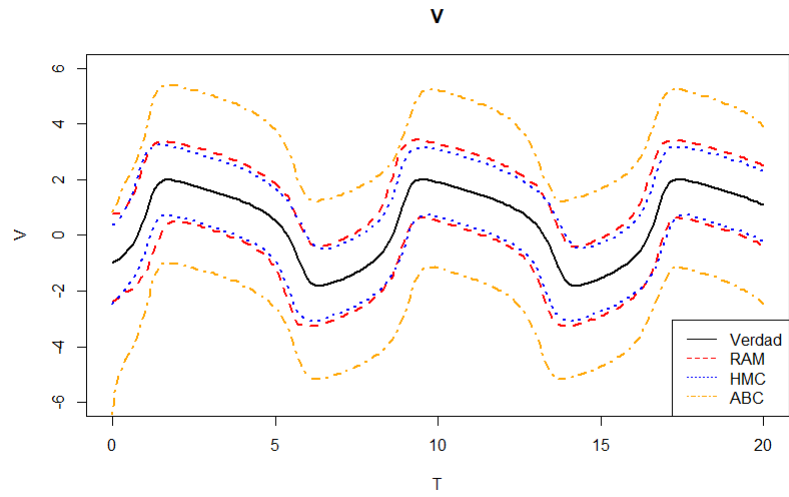


Figura 4.109: Comparación bandas de probabilidad de los distintos métodos MCMC en el estado  $V$

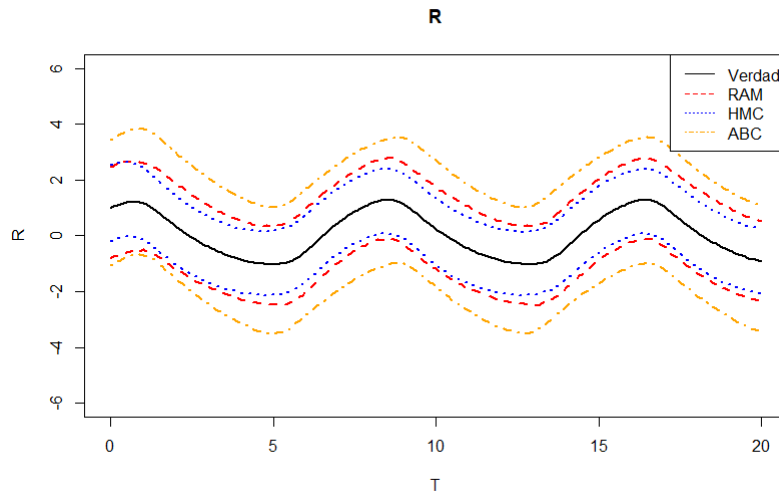


Figura 4.110: Comparación bandas de probabilidad de los distintos métodos MCMC en el estado  $R$

Cuadro 4.15: Comparación de métodos MCMC

Método MCMC	Iteraciones	Tiempo de CPU
HMC	10,000	10 horas
RAM	118,000	40 minutos
ABC	300,000	33 minutos

Se puede notar claramente que los métodos MCMC, entre menos requisitos (como se puede considerar al gradiente y/o verosimilitud) menor es tiempo de cómputo que requieren. Por otro lado, entre más requisitos requieren, más eficientes son las estimaciones y las bandas de probabilidad (cosa que sucede con el HMC).

También se puede ver que si el gradiente y/o verosimilitud fueran calculadas de forma rápida, dependiendo del problema, el tiempo de cómputo del método HMC no sería tan diferente comparado con los métodos que no utilizan estos procedimientos. Se sabe que el método hamiltoniano se desempeña de forma eficiente cuando su gradiente se puede calcular analíticamente sin ser costoso computacionalmente.

Hay que mencionar que todas las simulaciones se hicieron en una laptop Asus tuf gaming fx505dv.

## 4.2. Modelo Dinámico de Población

Las series de tiempo empíricas de especies que interactúan son altamente útiles para estudiar los mecanismos ecológicos. Los modelos matemáticos son herramientas valiosas para aclarar aún más esos mecanismos y procesos subyacentes. El modelo explica la relación entre un depredador metazoario, el rotífero *Brachionus calyciflorus s.s.*, y su presa *Monoraphidium minutum*, una alga verde unicelular, y las



algas crecen en un medio con nitrógeno (Rosenbaum, 2019). El siguiente sistema de ecuaciones modela la dinámica de población en tiempo continuo de un sistema predador-presa en un quimiostato con nitrógeno S, algas A y rotíferos R:

$$\begin{aligned}\frac{dS}{dt} &= (S^* - S)\delta - \frac{1}{c_A} \frac{f_A S}{h_A + S} A \\ \frac{dA}{dt} &= \frac{f_A S}{h_A + S} A - \frac{1}{c_R} \frac{f_R A}{h_R + A} R - \delta A \\ \frac{dR}{dt} &= \frac{f_R A}{h_R + A} R - \delta R\end{aligned}$$

Donde  $S^*$ ,  $\delta$ ,  $h_A$ ,  $h_R$  se pueden suponer como parámetros constantes.

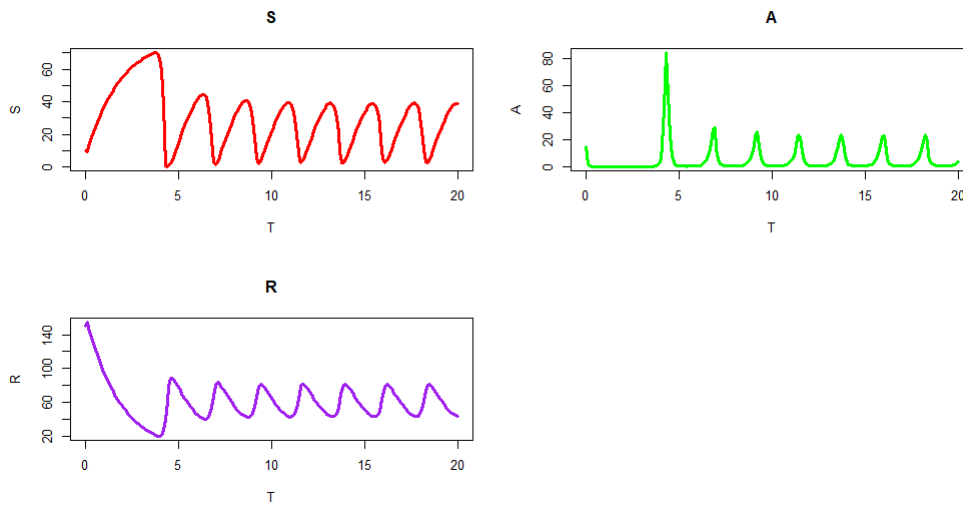


Figura 4.111: Solución del sistema predador-presa que ilustra su comportamiento cualitativo

Análogamente al análisis de la sección 4.1, se simularon datos con ciertos valores de los parámetros y se agregó un error gaussiano a ellos, en este caso solo para los estados  $A$  y  $R$ . En este modelo tenemos que  $\theta_i \in \mathbb{R}^+$  para todos las entradas del vector  $\theta$ . Nuestros datos simulados son los siguientes:

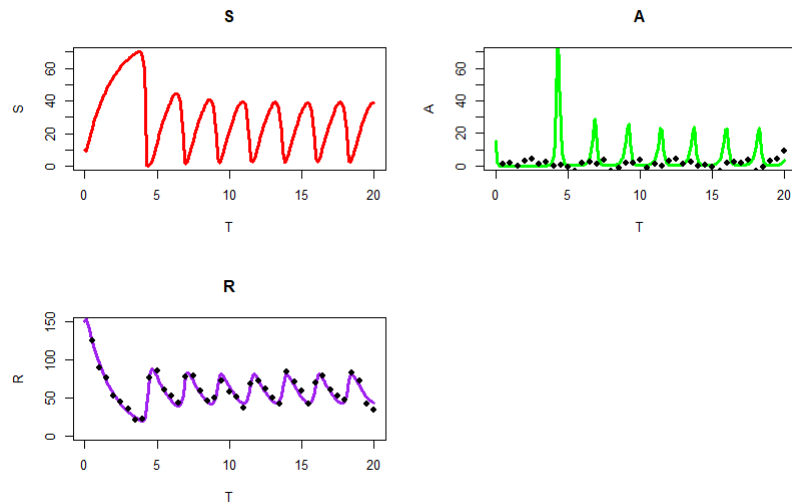


Figura 4.112: Solución del sistema predador-presa con los datos simulados del estado  $A$  y  $R$

#### 4.2.1. Método RAM

Se tomaron como hiper-parámetros los siguientes valores, para tener distribuciones iniciales no informativas:

- $\ln(f_A) \sim N(0, 3)$
- $\ln(f_R) \sim N(0, 3)$
- $\ln(c_A) \sim N(0, 3)$
- $\ln(c_R) \sim N(0, 3)$
- $\ln(S_0) \sim N(0, 3)$
- $\ln(A_0) \sim N(0, 3)$
- $\ln(R_0) \sim N(0, 3)$
- $\ln(\sigma_1^2) \sim N(0, 3)$
- $\ln(\sigma_2^2) \sim N(0, 3)$

Se realizaron simulaciones con 40 datos, con un periodo de calentamiento e iteraciones intermedias de 200,000 y lag 100 para poder obtener una muestra final de tamaño 1,600 utilizando programación en paralelo (220,000 por core). El tiempo de cómputo fue de 6 horas. Se tomaron los valores constantes de  $S^*=80$ ,  $\delta=0.55$ ,  $h_A=4.3$ ,  $h_R=75$ . Cabe notar que, como este sistema puede tener problemas de rigidez, se resolvió con el método “bdf” de la librería de DeSolve de R; se notó que este método es adecuado para este tipo de problemas, ya que el aumentar el número de pasos

para la solución numérica del sistema de EDO no tuvo un impacto significativo en la calidad de las simulaciones.

Los resultados obtenidos fueron los siguientes, los histogramas de las muestras finales en el siguiente orden:  $f_A$ ,  $f_R$ ,  $c_A$ ,  $c_R$ ,  $S_0$ ,  $A_0$ ,  $R_0$ ,  $\sigma_1^2$ ,  $\sigma_2^2$  donde las líneas rojas denotan los verdaderos valores de los parámetros.

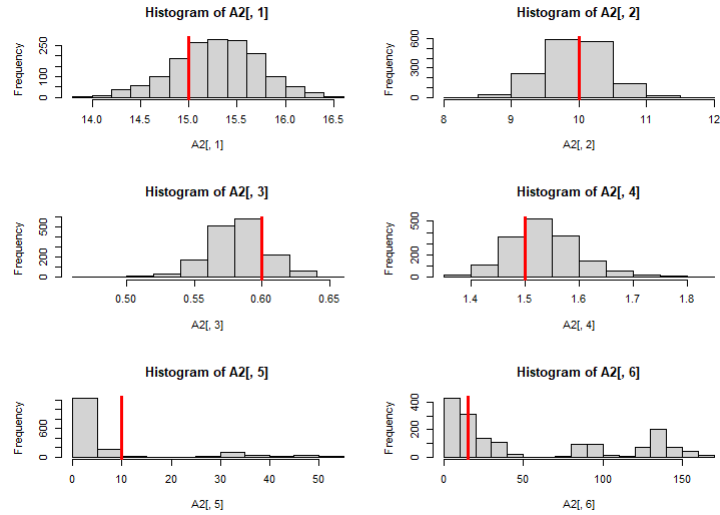


Figura 4.113: Histogramas de los parámetros  $f_A$ ,  $f_R$ ,  $\frac{1}{c_A}$ ,  $\frac{1}{c_R}$ ,  $S_0$ ,  $A_0$  usando el método RAM

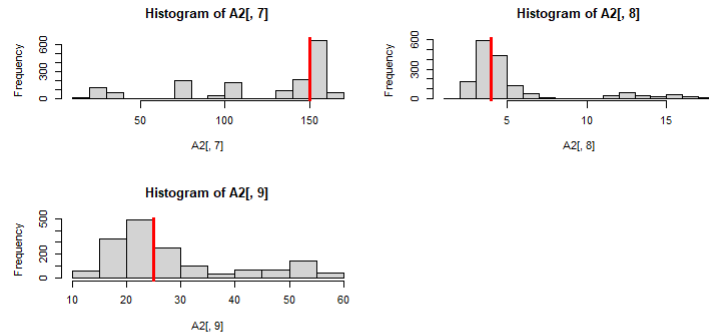


Figura 4.114: Histogramas de los parámetros  $R_0$ ,  $\sigma_1^2$ ,  $\sigma_2^2$  usando el método RAM

Con los siguientes promedios ergódicos y gráficas de autocorrelación:

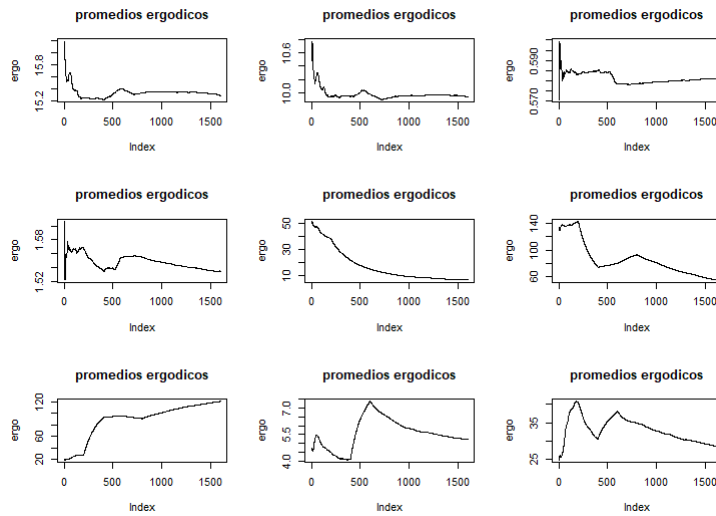


Figura 4.115: Promedios ergódicos de los parámetros  $f_A, f_R, \frac{1}{c_A}, \frac{1}{c_R}, S_0, A_0, R_0, \sigma_1^2, \sigma_2^2$

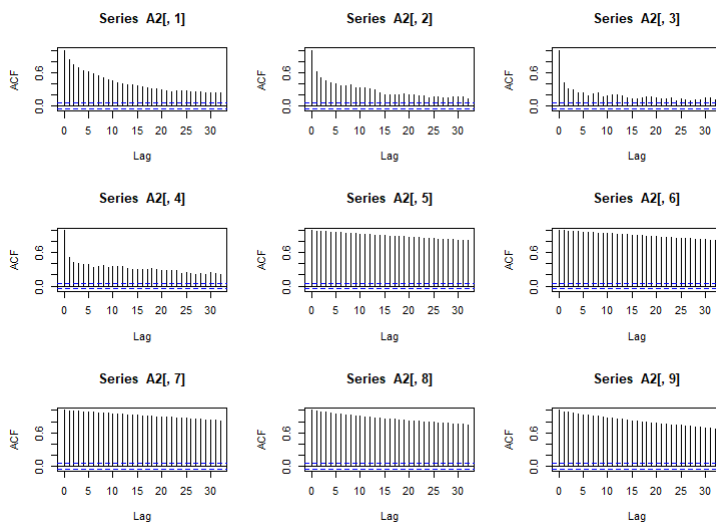


Figura 4.116: Gráficas de auto correlación de los parámetros  $f_A, f_R, \frac{1}{c_A}, \frac{1}{c_R}, S_0, A_0, R_0, \sigma_1^2, \sigma_2^2$

Y los resultados numéricos de nuestras estimaciones fueron los siguientes:

Cuadro 4.16: Modelo predador presa

Parámetro	Verdadero valor	Media	Mediana	Intervalo de Prob.
$f_A$	15	15.29	15.30	(11.626 ; 16.6134)
$f_R$	10	9.937	9.952	(9.0533 ; 10.7972)
$1/c_A$	0.6	0.5811	0.5821	(0.5396 ; 0.6260)
$1/c_R$	1.5	1.535	1.53	(1.4052 ; 1.6701)
$S_0$	10	6.5287	1.3192	( $5.63e^{-04}$ ; 37.10)
$A_0$	15	54.895	24.882	(2.21 ; 145.93)
$R_0$	150	120.87	146.58	(24.893 ; 161.868)
$\sigma_1^2$	4	5.265	4.036	(2.35 ; 14.339)
$\sigma_2^2$	25	28.58	24.12	(14.756 ; 55.545)

Podemos notar claramente que los parámetros que más problema causaron son los valores iniciales para los tres estados del sistema. En las siguientes gráficas mostramos la comparación de nuestras verdaderas soluciones para cada uno de los estados  $A$ ,  $S$  y  $R$  con la solución estimada respectivamente (líneas azules).

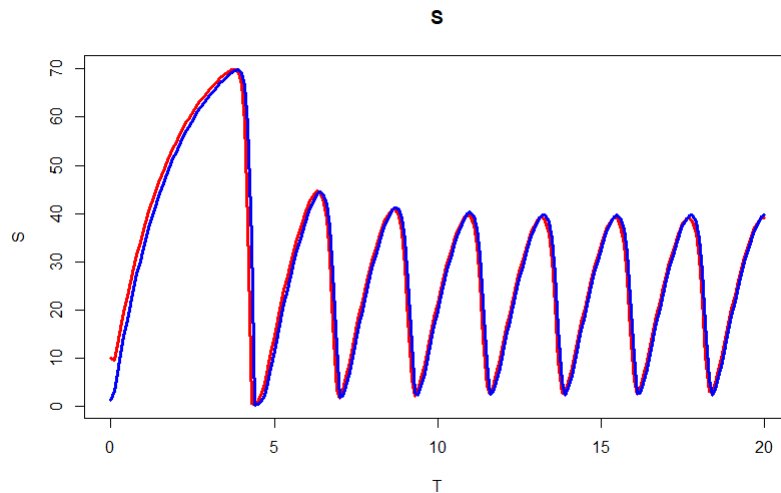


Figura 4.117: Solución media del estado  $S$  y su verdadera solución usando el método RAM

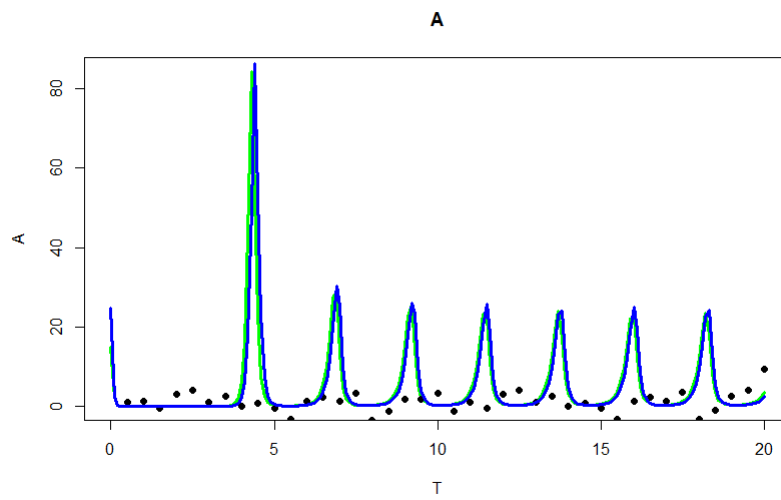


Figura 4.118: Solución media del estado  $A$  y su verdadera solución usando el método RAM

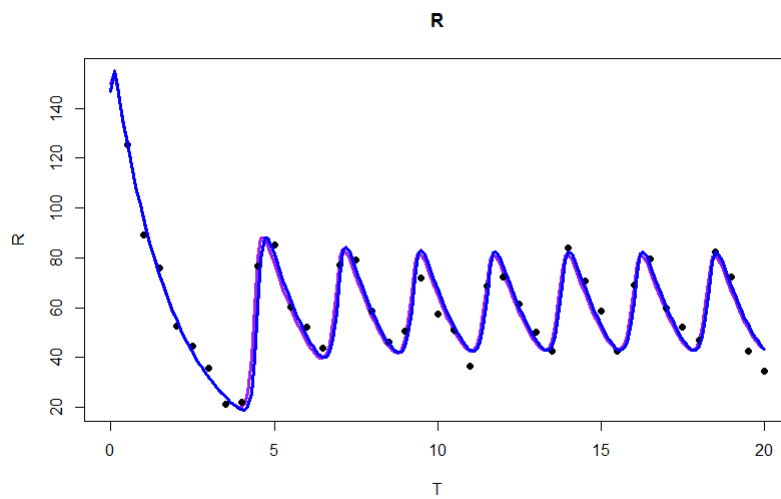


Figura 4.119: Solución media del estado  $R$  y su verdadera solución usando el método RAM

Podemos notar que aunque no tengamos información del estado  $S$ , nuestros datos de los otros dos estados ayudan a explicar la relación que tienen con el estado  $S$ , obteniendo buenas estimaciones de su verdadero comportamiento. También podemos calcular las bandas de probabilidad respectivas de los estados  $A$  y  $R$ . Las gráficas 4.120 y 4.121 muestran estas bandas (líneas naranjas), con las líneas azules muestran el comportamiento medio del sistema. Cabe resaltar que, como no tenemos datos del estado  $S$ , no podemos realizar estimaciones de su variabilidad y, por tanto, no podemos calcular las bandas de probabilidad para este estado en particular.

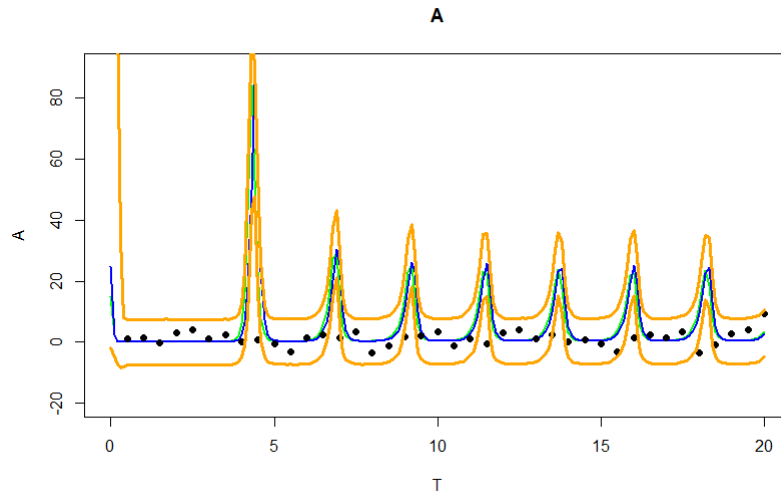


Figura 4.120: Bandas de probabilidad del estado  $A$  usando el método RAM

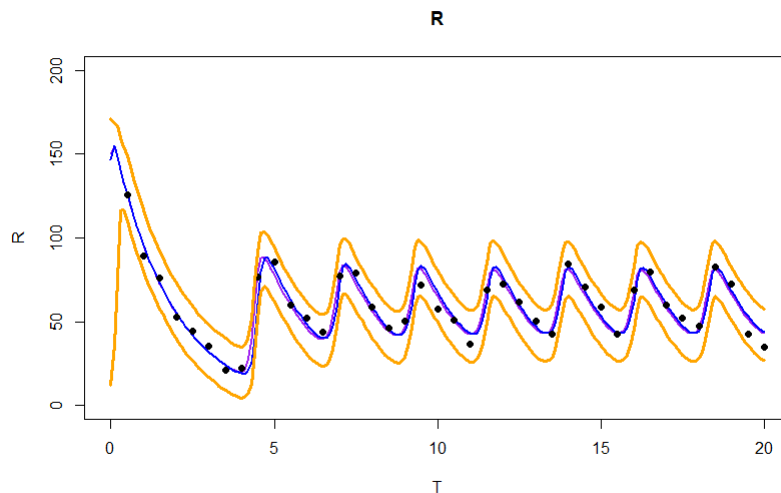


Figura 4.121: Bandas de probabilidad del estado  $R$  usando el método RAM

Las trazas de nuestra muestra se muestran en la siguiente gráfica:

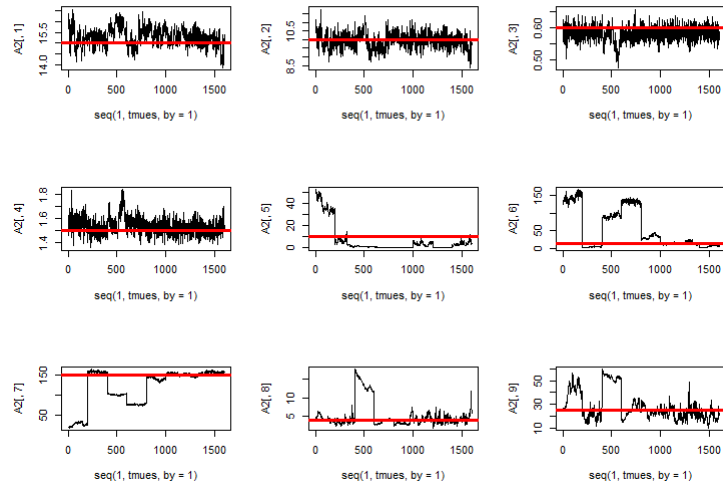


Figura 4.122: Trazas de la muestra final de los parámetros  $f_A, f_R, \frac{1}{c_A}, \frac{1}{c_R}, S_0, A_0, R_0, \sigma_1^2, \sigma_2^2$

Podemos observar que como se programó el algoritmo en paralelo, para cada uno de los 8 núcleos disponibles tenemos una submuestra de tamaño 200, y al juntarles tenemos una muestra final de tamaño 1,600. Podemos observar que algunos núcleos presentan un comportamiento distinto a los otros al todavía no presentar convergencia a los verdaderos valores. Una forma que podría resolver este problema es aumentando el periodo de calentamiento hasta garantizar convergencia en los 8 núcleos, pero también se debe tomar en cuenta que esto va a aumentar el costo computacional. En este modelo los parámetros que presentan este problema en particular son los valores iniciales del sistema de EDO. Sin embargo, la metodología propuesta en este trabajo de tesis sigue ofreciendo inferencia adecuada en términos de la distribución predictiva, y sus correspondientes bandas de probabilidad.

Hay que mencionar que todas las simulaciones se hicieron en una laptop Asus tuf gaming fx505dv.

### 4.3. Modelos de Datos Circulares

Para los modelos de datos circulares ( $\theta$ ) se trabajó con 4 métodos MCMC, primero con el Metropolis-Hastings (MH) al completar el espacio y hacer simulaciones sobre las variables  $(r, \theta) \in [\mathbb{R}^+ \times (0, 2\pi)]$ , y por medio de transformaciones a coordenadas polares calcular el vector  $\mu$  correspondiente, los otros 3 métodos a utilizar son el HMC, el RAM y algoritmo de muestreo de rechazo ABC, donde utilizamos distribuciones normales proyectadas  $NP(\theta|\mu, I)$ . Se realizaron tres análisis con una muestra simulada de tamaño 100 y valores del vector  $\mu$  dados por (5,5); (1,1) y (0.5,0.5). Para las valores de  $\mu_1$  y  $\mu_2$  se consideraron distribuciones iniciales no informativas  $N(0, 5)$ . Utilizamos el algoritmo de muestreo de rechazo ABC que realiza iteraciones hasta obtener el número de muestra de que fijamos, por lo mismo no



tuvimos un control en el número de iteraciones totales del algoritmo.

### 4.3.1. Primer análisis

Para nuestro primer estudio consideramos 100 datos de la variable  $\theta$  a partir del vector  $\mu = (5, 5)$ , bajo el modelo  $NP(\theta, \mu, I)$ . La medida de distancia utilizada para el algoritmo de muestreo de rechazo ABC, que se empleó fue el siguiente (Mardia & Jupp, 2009)

$$\rho(\theta, z) = \frac{1}{n} \sum_{j=1}^n (1 - \text{Cos}(z_j - \bar{\theta}_D)),$$

donde  $\bar{\theta}_D$  es la media muestral de los datos circulares  $\theta$  (notemos que  $\bar{\theta}_D \neq \frac{1}{n} \sum_{i=1}^n \theta_i$ ). Esta es una función que mide la dispersión de los datos  $z$  con respecto a la media de  $\bar{\theta}_D$ . El valor de  $\epsilon$  considerado fue de  $\epsilon = 0.01$ .

Nuestros datos gráficamente se ven de la siguiente manera:

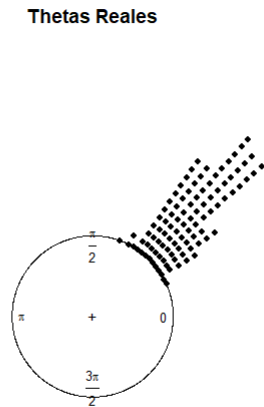


Figura 4.123: Datos del vector  $\mu=(5,5)$

Los resultados obtenidos fueron:

Cuadro 4.17: HMC

Parámetro	Verdadero valor	Media	Mediana	Intervalo de Prob.
$\mu_1$	5	4.816	4.809	(4.13 ; 5.48)
$\mu_2$	5	4.905	4.896	(4.18 ; 5.61)

Cuadro 4.18: RAM

Parámetro	Verdadero valor	Media	Mediana	Intervalo de Prob.
$\mu_1$	5	4.797	4.786	(4.15 ; 5.49)
$\mu_2$	5	4.885	4.868	(4.15 ; 5.53)

Cuadro 4.19: MH

Parámetro	Verdadero valor	Media	Mediana	Intervalo de Prob.
$\mu_1$	5	4.876	4.871	(4.17 ; 5.51)
$\mu_2$	5	4.961	4.944	(4.32 ; 5.66)

Cuadro 4.20: ABC

Parámetro	Verdadero valor	Media	Mediana	Intervalo de Prob.
$\mu_1$	5	6.479	6.274	(4.2482 ; 8.7071)
$\mu_2$	5	6.595	6.365	(4.6391 ; 9.3031)

Donde si comparamos el tiempo de CPU y el número de iteraciones.

Cuadro 4.21: Comparación

Método MCMC	Iteraciones	Tiempo de CPU
HMC	30,000	18 minutos
RAM	160,000	4 minutos
MH	40,000	30 minutos
ABC	—	1 minuto

Los histogramas, promedios ergódicos, gráficas de auto correlación y trazas de cada método MCMC se presentan a continuación:

## HMC

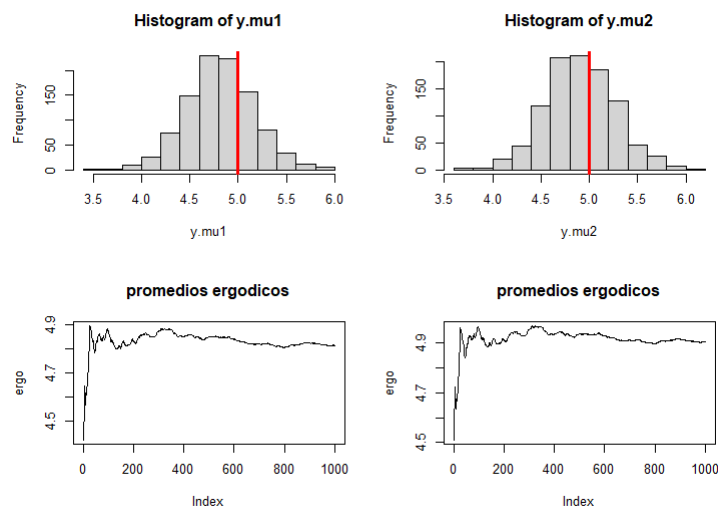


Figura 4.124: Histogramas y promedios ergódicos del escenario  $\mu = (5, 5)$  usando el método HMC

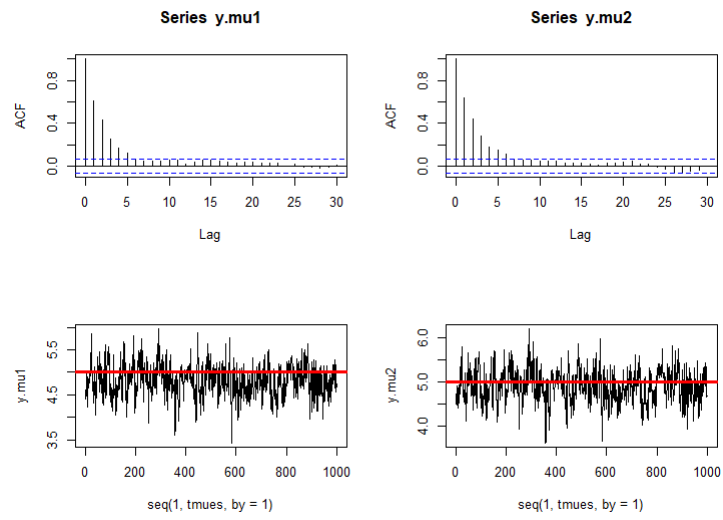


Figura 4.125: Gráficas de auto correlación y trazas del escenario  $\mu = (5, 5)$  usando el método HMC

**RAM**

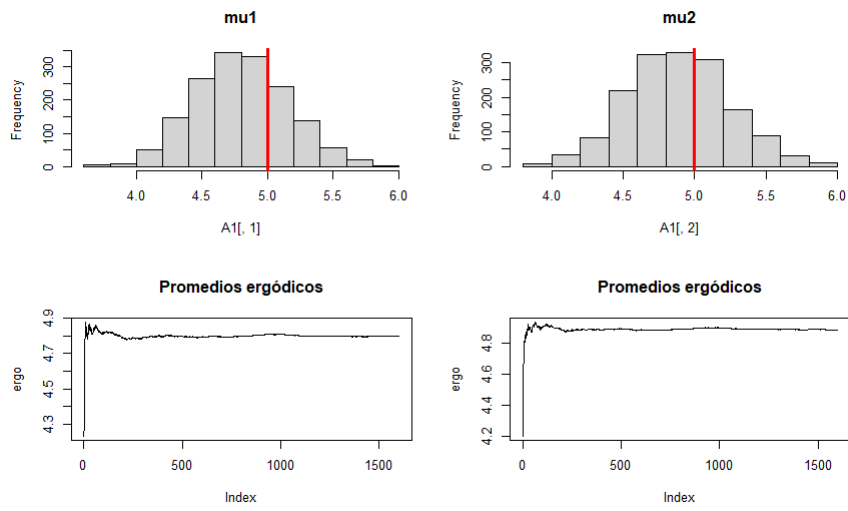


Figura 4.126: Histogramas y promedios ergódicos del escenario  $\mu = (5, 5)$  usando el método RAM

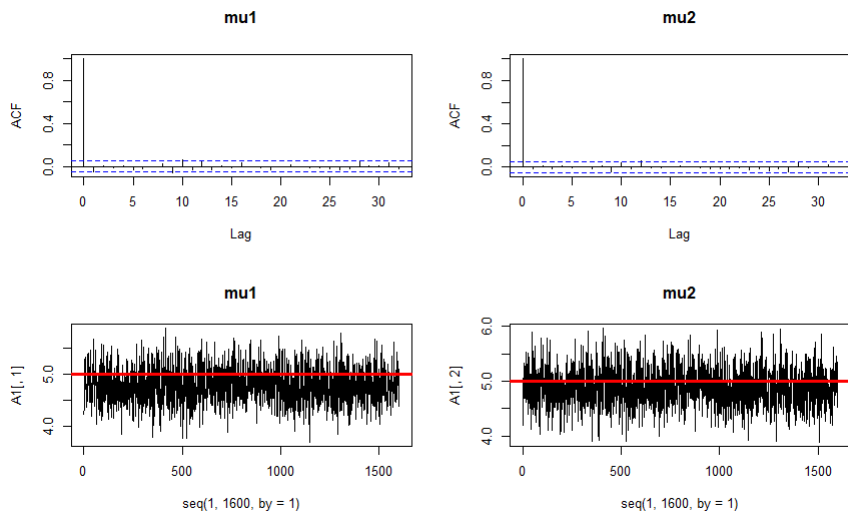


Figura 4.127: Gráficas de auto correlación y trazas del escenario  $\mu = (5, 5)$  usando el método RAM

MH

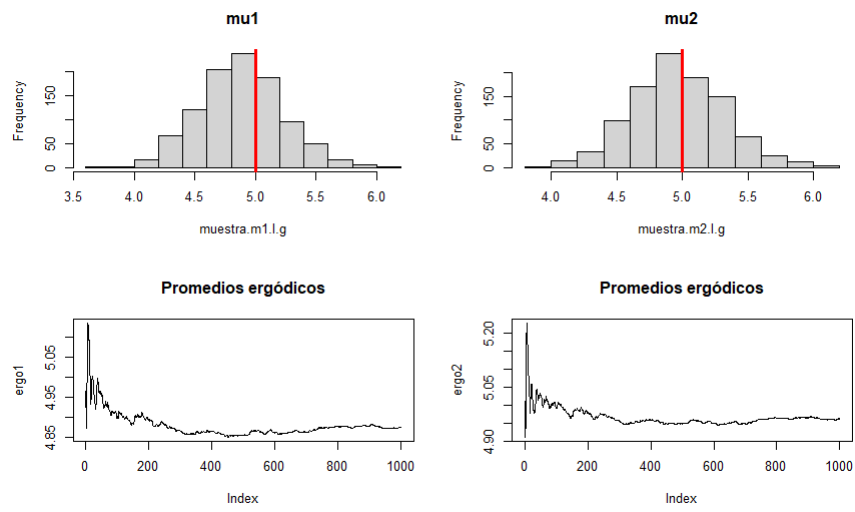


Figura 4.128: Histogramas y promedios ergódicos del escenario  $\mu = (5, 5)$  usando el método MH

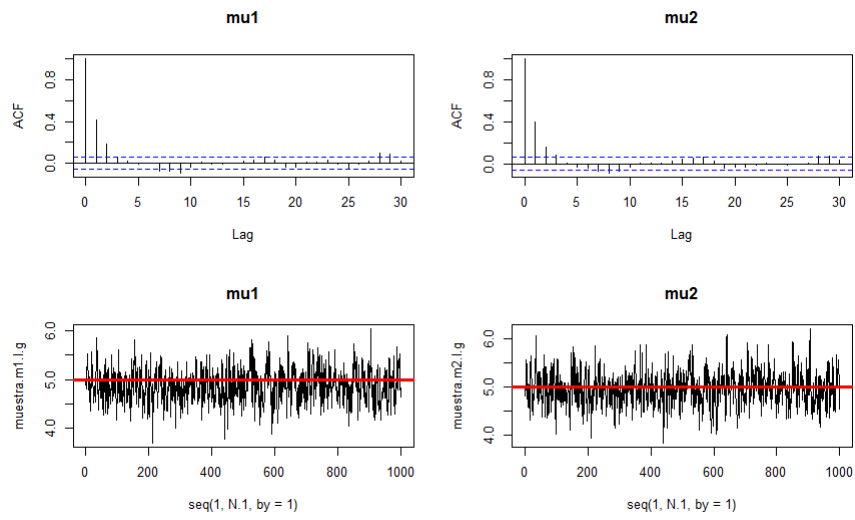


Figura 4.129: Gráficas de auto correlación y trazas del escenario  $\mu = (5, 5)$  usando el método MH

ABC

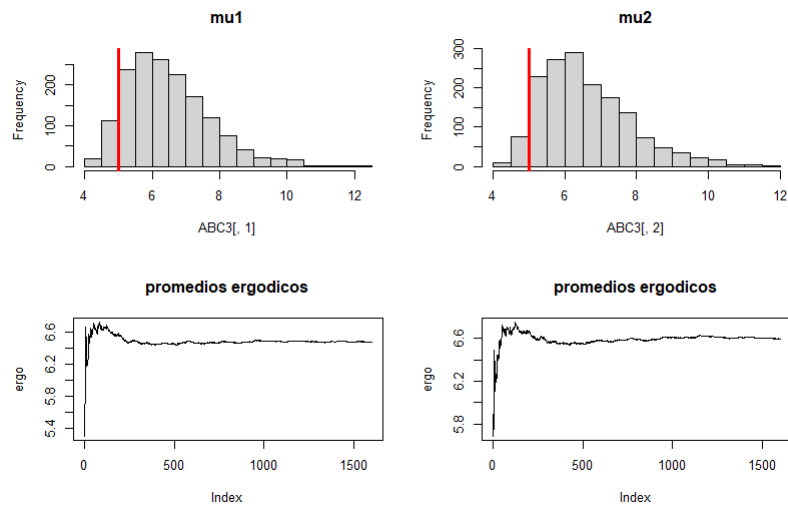


Figura 4.130: Histogramas y promedios ergódicos del escenario  $\mu = (5, 5)$  usando el método ABC

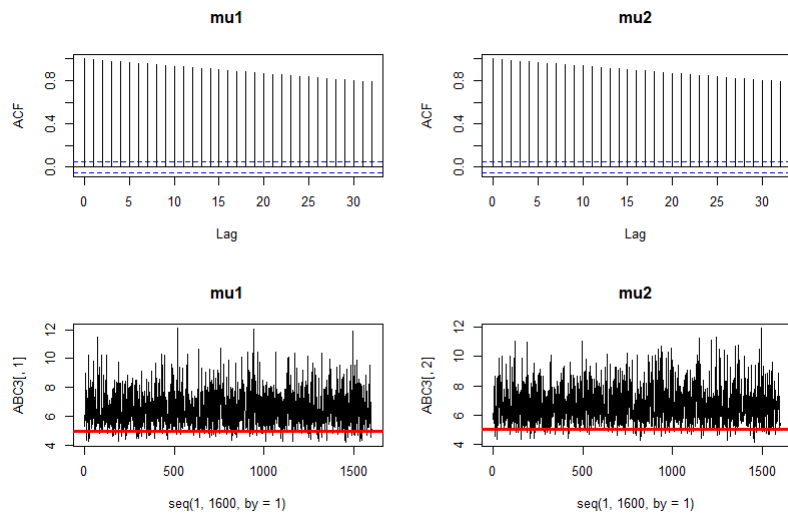


Figura 4.131: Gráficas de auto correlación y trazas del escenario  $\mu = (5, 5)$  usando el método ABC

### 4.3.2. Segundo análisis

Para nuestro segundo estudio también consideramos una muestra aleatoria de 100 datos de la variable  $\theta$  ahora a partir del vector  $\mu = (1, 1)$ , bajo el modelo  $NP(\theta, \mu, I)$ . La medida de distancia, para el algoritmo de muestreo de rechazo ABC, que se empleó fue la diferencia circular (Mardia & Jupp, 2009)

$$\rho(\theta, z) = \frac{1}{n^2} \sum_{i=1}^n \sum_{j=1}^n (1 - \text{Cos}(z_i - \theta_j)),$$

con el valor de  $\epsilon = 0.0005$ , nuestros datos gráficamente se ven de la siguiente manera:

**Thetas Reales**

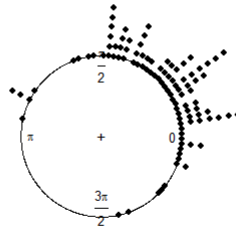


Figura 4.132: Datos del vector  $\mu=(1,1)$

Los resultados obtenidos fueron:

Cuadro 4.22: HMC2

Parámetro	Verdadero valor	Media	Mediana	Intervalo de Prob.
$\mu_1$	1	1.170	1.174	(0.947 ; 1.423)
$\mu_2$	1	0.976	0.976	(0.735 ; 1.188)

Cuadro 4.23: RAM2

Parámetro	Verdadero valor	Media	Mediana	Intervalo de Prob.
$\mu_1$	1	1.182	1.183	(0.919 ; 1.416)
$\mu_2$	1	0.978	0.979	(0.744 ; 1.225)

Cuadro 4.24: MH2

Parámetro	Verdadero valor	Media	Mediana	Intervalo de Prob.
$\mu_1$	1	1.216	1.215	(0.966 ; 1.468)
$\mu_2$	1	1.010	1.006	(0.766 ; 1.251)



Cuadro 4.25: ABC2

Parámetro	Verdadero valor	Media	Mediana	Intervalo de Prob.
$\mu_1$	1	1.1952	1.1610	(0.7054 ; 1.771)
$\mu_2$	1	1.0393	1.0407	(0.6969 ; 1.3404)

Donde si comparamos el tiempo de CPU y el número de iteraciones.

Cuadro 4.26: Comparación

Método MCMC	Iteraciones	Tiempo de CPU
HMC	30,000	13 minutos
RAM	160,000	4 minutos
MH	40,000	30 minutos
ABC	—	1 minuto

Los histogramas, promedios ergódicos, gráficas de auto correlación y trazas de cada método MCMC se presentan a continuación:

### HMC

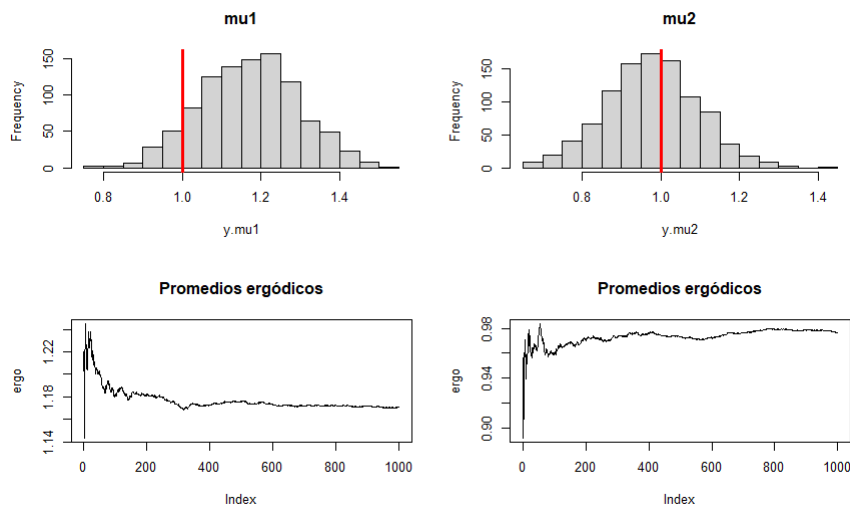


Figura 4.133: Histogramas y promedios ergódicos del escenario  $\mu = (1, 1)$  usando el método HMC

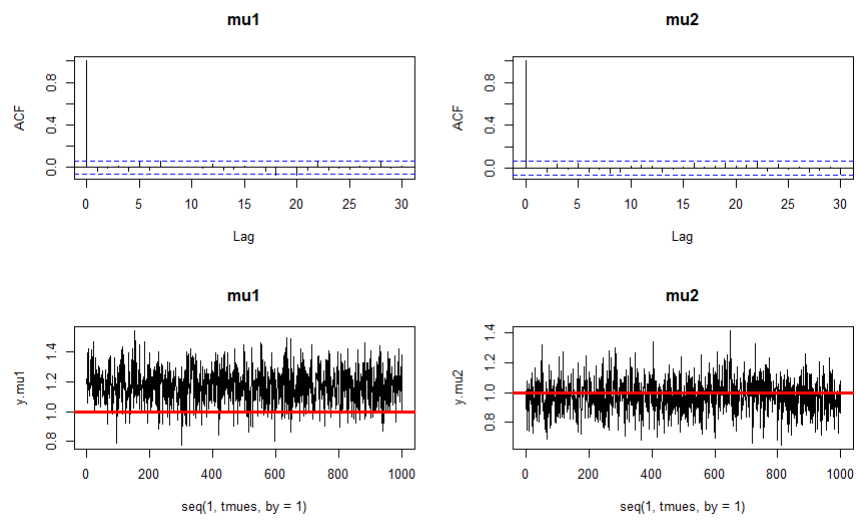


Figura 4.134: Gráficas de auto correlación y trazas del escenario  $\mu = (1, 1)$  usando el método HMC

**RAM**

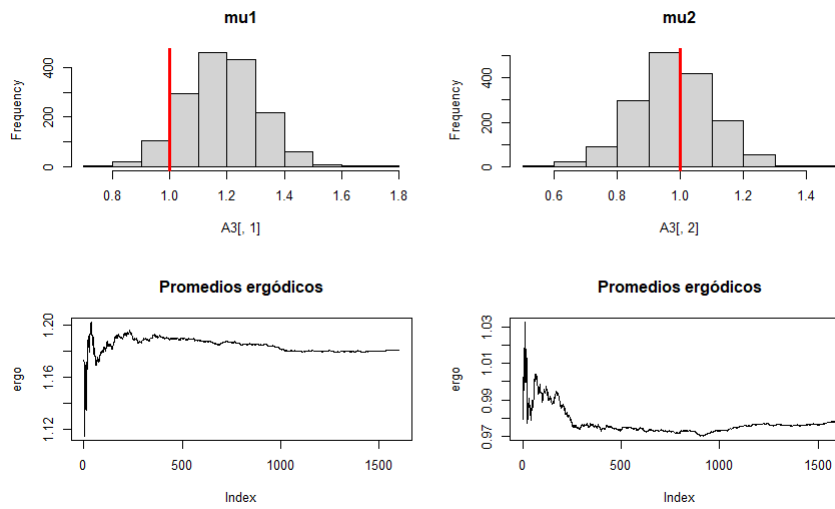


Figura 4.135: Histogramas y promedios ergódicos del escenario  $\mu = (1, 1)$  usando el método RAM

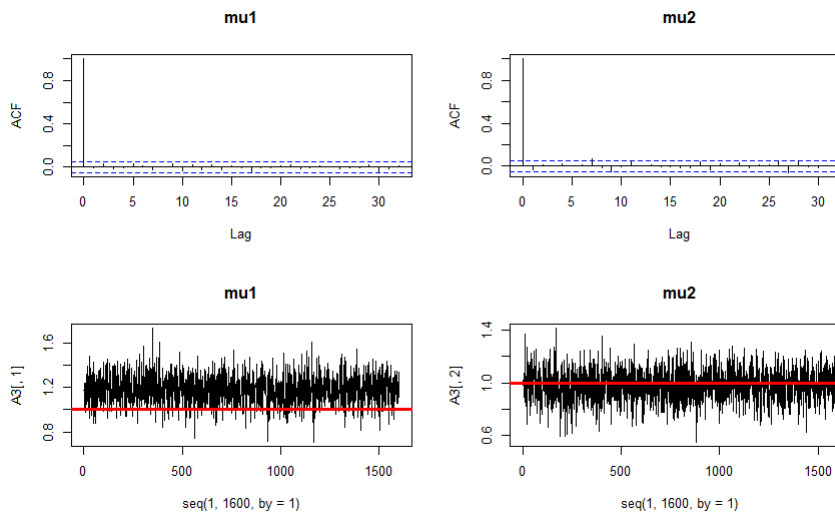


Figura 4.136: Gráficas de auto correlación y trazas del escenario  $\mu = (1, 1)$  usando el método RAM

MH

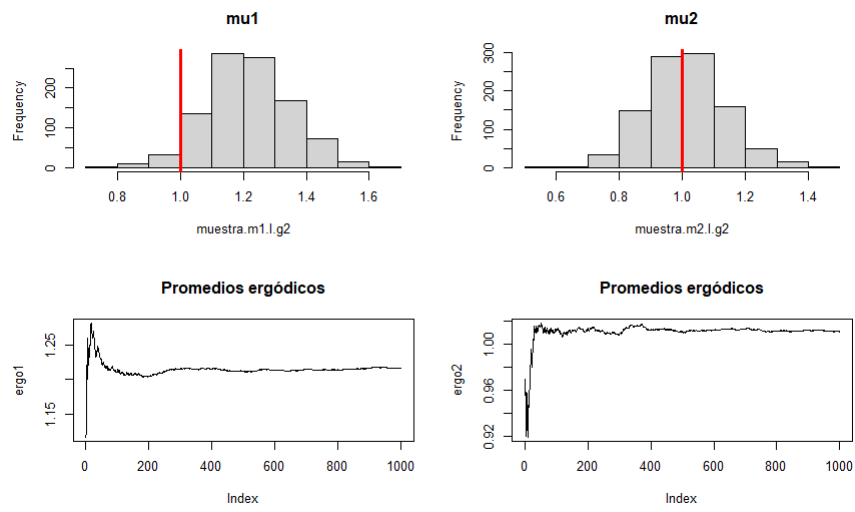


Figura 4.137: Histogramas y promedios ergódicos del escenario  $\mu = (1, 1)$  usando el método MH

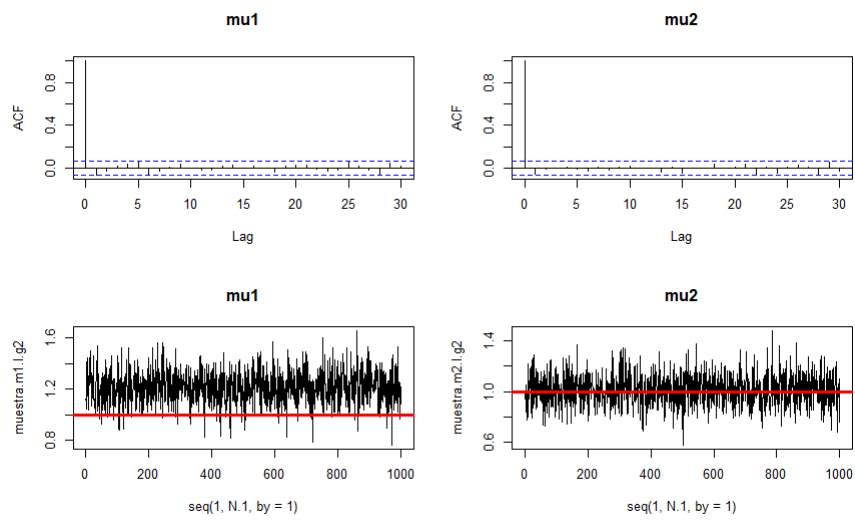


Figura 4.138: Gráficas de auto correlación y trazas del escenario  $\mu = (1, 1)$  usando el método MH

ABC

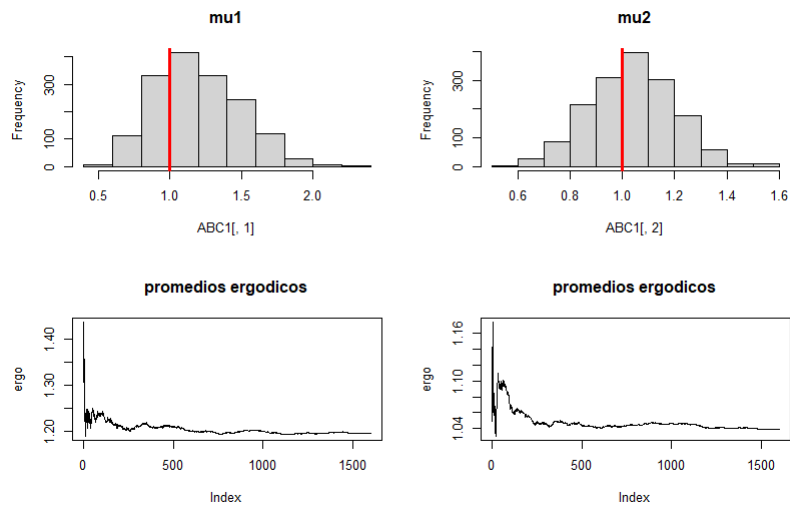


Figura 4.139: Histogramas y promedios ergódicos del escenario  $\mu = (1, 1)$  usando el método ABC

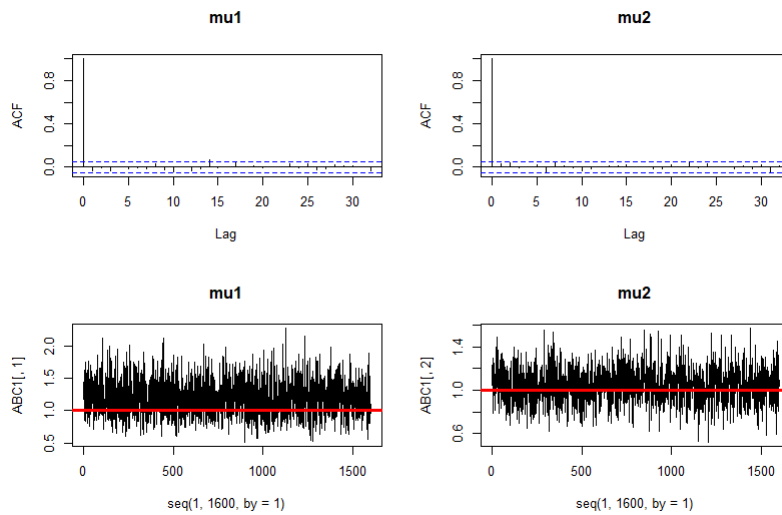


Figura 4.140: Gráficas de auto correlación y trazas del escenario  $\mu = (1, 1)$  usando el método ABC

Una observación importante que debemos de mencionar, es la importancia, como ya se ha mencionado varias veces anteriormente, de escoger la medida de distancia  $\rho$  y el valor de  $\epsilon$  adecuados para cada problema, la diferencia circular que usamos para este estudio se desempeña mejor para datos circulares que presentan una mayor variabilidad. Para los otros tres métodos, podemos ver que se siguen desempeñando de forma adecuada para los dos estudios realizados hasta ahora.

### 4.3.3. Tercer análisis

Para nuestro tercer estudio consideramos 100 datos de la variable  $\theta$  obtenido del vector  $\mu=(0.5,0.5)$ . La medida de distancia, para el algoritmo de muestreo de rechazo ABC, que se empleó fue la diferencia circular,

$$\rho(\theta, z) = \frac{1}{n^2} \sum_{i=1}^n \sum_{j=1}^n (1 - \text{Cos}(z_i - \theta_j)),$$

con el valor de  $\epsilon = 0.0005$ , nuestros datos gráficamente se ven de la siguiente manera:

Thetas Reales

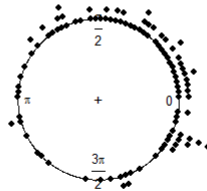


Figura 4.141: Datos del vector  $\mu=(0.5,0.5)$

Los resultados obtenidos fueron:

Cuadro 4.27: HMC3

Parámetro	Verdadero valor	Media	Mediana	Intervalo de Prob.
$\mu_1$	0.5	0.5001	0.5004	(0.290 ; 0.726)
$\mu_2$	0.5	0.283	0.361	(0.137 ; 0.582)

Cuadro 4.28: RAM3

Parámetro	Verdadero valor	Media	Mediana	Intervalo de Prob.
$\mu_1$	0.5	0.497	0.499	(0.263 ; 0.738)
$\mu_2$	0.5	0.362	0.359	(0.138 ; 0.592)

Cuadro 4.29: MH3

Parámetro	Verdadero valor	Media	Mediana	Intervalo de Prob.
$\mu_1$	0.5	0.510	0.509	(0.262 ; 0.712)
$\mu_2$	0.5	0.377	0.378	(0.147 ; 0.584)

Cuadro 4.30: ABC3

Parámetro	Verdadero valor	Media	Mediana	Intervalo de Prob.
$\mu_1$	0.5	0.4943	0.4882	(0.1061 ; 0.9053)
$\mu_2$	0.5	0.3863	0.3877	(0.0979 ; 0.6617)

Donde si comparamos el tiempo de CPU y el número de iteraciones.

Cuadro 4.31: Comparación

Método MCMC	Iteraciones	Tiempo de CPU
HMC	30,000	13 minutos
RAM	160,000	4 minutos
MH	40,000	30 minutos
ABC	—	1 minuto

Los histogramas, promedios ergódicos, gráficas de auto correlación y trazas de cada método MCMC se presentan a continuación:

## HMC

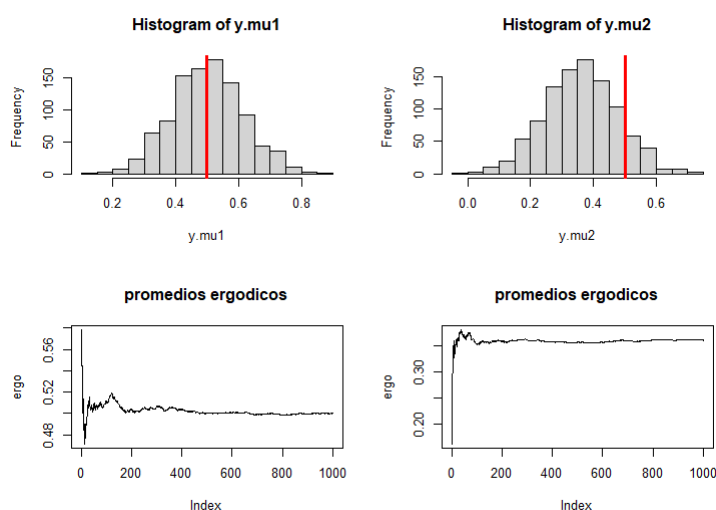


Figura 4.142: Histogramas y promedios ergódicos del escenario  $\mu = (0.5, 0.5)$  usando el método HMC



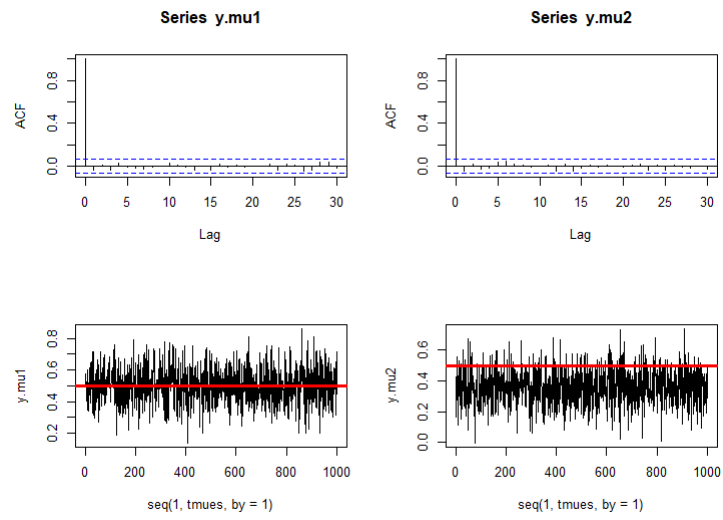


Figura 4.143: Gráficas de auto correlación y trazas del escenario  $\mu = (0.5, 0.5)$  usando el método HMC

**RAM**

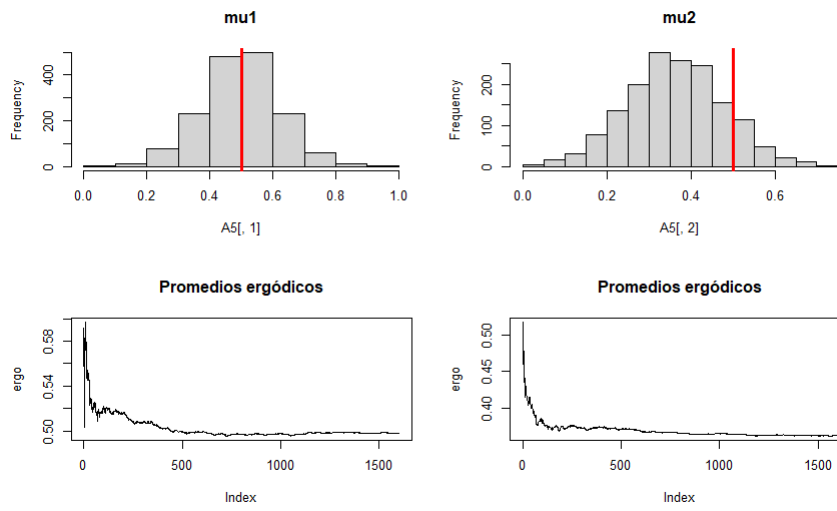


Figura 4.144: Histogramas y promedios ergódicos del escenario  $\mu = (0.5, 0.5)$  usando el método RAM

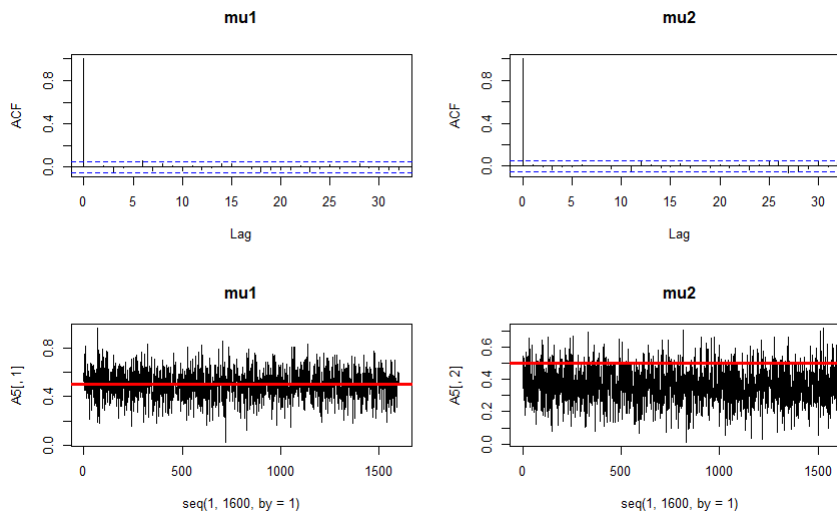


Figura 4.145: Gráficas de auto correlación y trazas del escenario  $\mu = (0.5, 0.5)$  usando el método RAM

MH

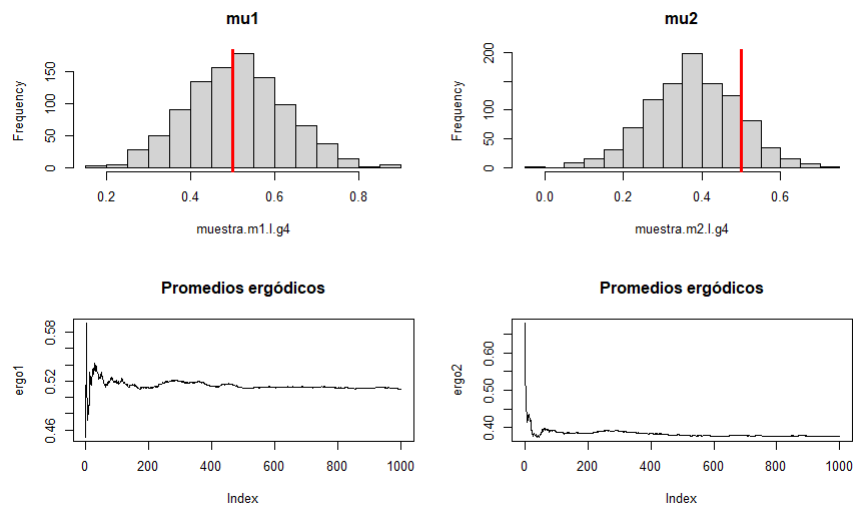


Figura 4.146: Histogramas y promedios ergódicos del escenario  $\mu = (0.5, 0.5)$  usando el método MH

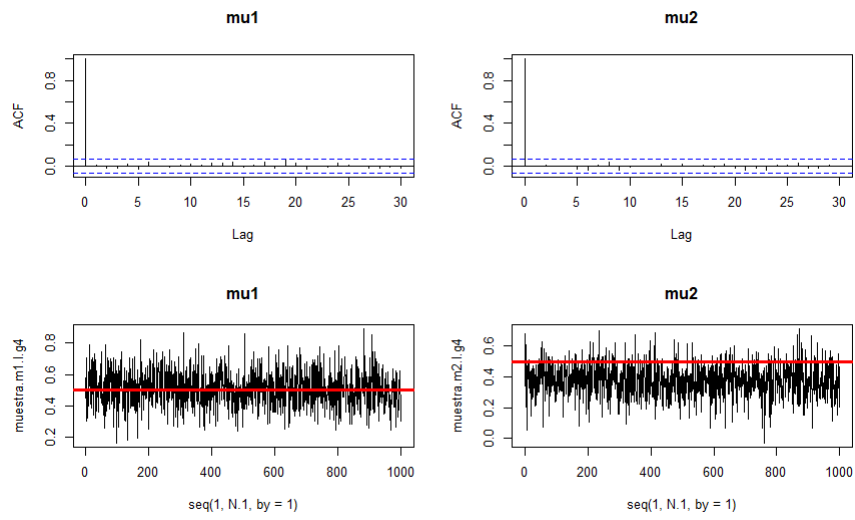


Figura 4.147: Gráficas de auto correlación y trazas del escenario  $\mu = (0.5, 0.5)$  usando el método MH

ABC

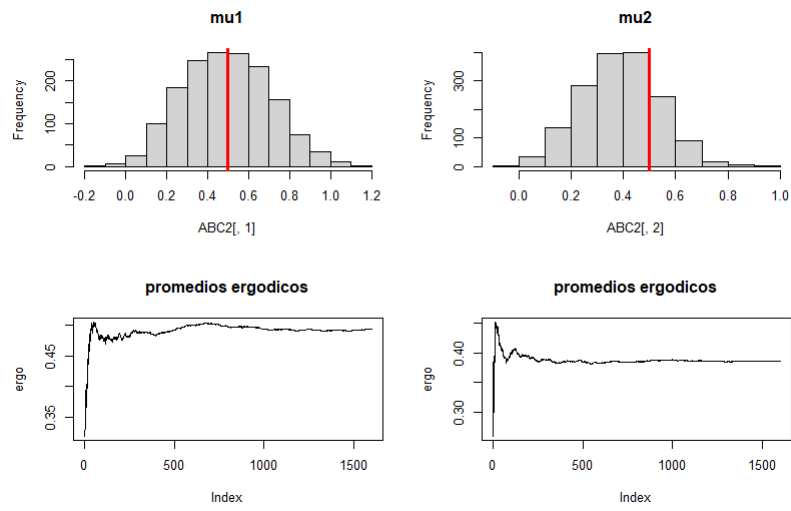


Figura 4.148: Histogramas y promedios ergódicos del escenario  $\mu = (0.5, 0.5)$  usando el método ABC

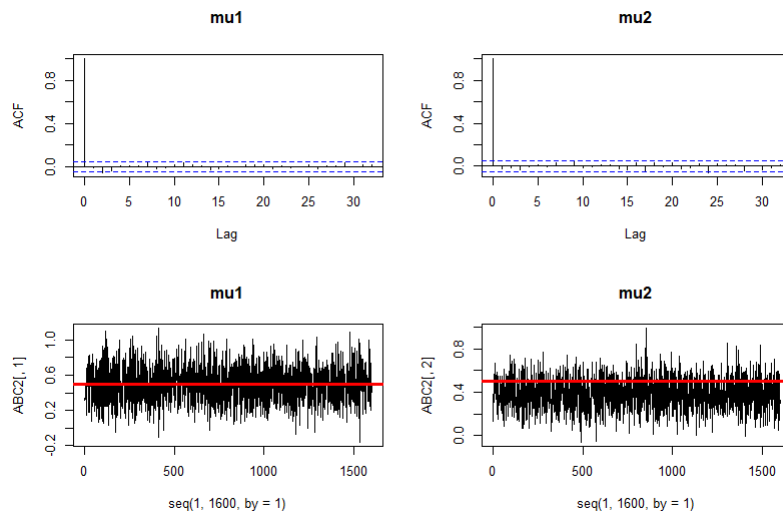


Figura 4.149: Gráficas de auto correlación y trazas del escenario  $\mu = (0.5, 0.5)$  usando el método ABC

Podemos observar que con los tres análisis realizados anteriormente, es que todos los métodos MCMC que se utilizaron para este modelo funcionaron de forma adecuada. Sin embargo, aunque todos los intervalos de probabilidad contienen a los verdaderos valores de los parámetros  $\mu_1$  y  $\mu_2$ , es el método HMC el que ofrece los mejores intervalos, algo que también sucedió con el modelo FH-N, también hay que resaltar que es el método ABC el que requiere un menor tiempo de cómputo, mientras que el RAM es el punto medio entre estos dos métodos, ya es el método que mejor optimiza la relación entre calidad de las inferencias y tiempo de cómputo.

Algo que también se debe de señalar que aunque los datos circulares son datos definidos en un espacio muestral distinto a  $\mathbb{R}^k$ , y las metodologías propuestas en esta tesis ofrecen alternativas a considerar para su análisis.

Hay que mencionar que todas las simulaciones se hicieron en una laptop Asus tuf gaming fx505dv.

# Conclusiones

Una de nuestras primeras conclusiones es que, con los resultados mostrados de los datos circulares del método hamiltoniano de Monte Carlo, podemos concluir que éste método es óptimo cuando el gradiente se obtiene analíticamente o su cómputo numérico no es tan costoso, lo cual no ocurre con los sistemas de EDO. En cambio, para estimar los parámetros del sistema FH-N tuvimos que emplear, alternativamente, una estimación numérica para el cálculo del gradiente que involucraba dicho método. En contraste, el MCMC RAM, que es una MCMC adaptativo, es más eficiente en comparación, pues es menos costoso comparado al HMC. De cualquier modo, siempre hay que resolver internamente en el algoritmo general, al sistema de EDO en cada paso para el cálculo de la distribución final, lo cual se evitó usando los métodos ABC. Después de trabajar con los cuatro métodos MCMC analizados en esta tesis, podemos hacer una tabla de ventajas y desventajas:

Cuadro 4.32: Ventajas y Desventajas

<i>Métodos</i> <i>MCMC</i>	<i>Ventajas</i>	<i>Desventajas</i>
RAM	Es fácil de implementar. Es un método adaptativo. Es un precio justo entre tiempo de cómputo y calidad de estimaciones.	Requiere de varias iteraciones y éstas aumentan según la dificultad del modelo a analizar.
HMC	El gradiente nos ayuda a mover el vector de parámetros de forma más eficiente. No requiere tantas iteraciones Rápido si el gradiente se puede calcular analíticamente.	Requiere el cálculo del gradiente y si éste es numérico puede ser muy costoso computacionalmente.
ABC	No requiere verosimilitud. Es bastante rápido.	Depende de las distribuciones iniciales. Depende del valor de $\epsilon$ . Necesita que la medida de comparación de los datos sea suficiente. Es complicado estimar la varianza.
MH	Es fácil de implementar. Es un método clásico para la inferencia bayesiana.	Depende fuertemente de la elección de la distribución propuesta. Puede tener tasas de aceptación muy baja.

Algo que nos dimos cuenta en los modelos de datos circulares fue que el método HMC, el cálculo numérico del gradiente mediante diferencias finitas, puede requerir más tiempo de cómputo si se aumenta el número de datos. Mientras que con los otros tres algoritmos MH, RAM y ABC con los que se comparó esto no sucedía, ya que el aumentar el número de datos a 500, el tiempo de cómputo permanecía casi igual mientras que en el HMC aumentaba a 1 hora.

Otra conclusión importante es que, para todos los escenarios mostrados, la metodología presentada en el Capítulo 2 nos permite una mejor descripción del verdadero comportamiento de los datos, al tener la estimación de las varianzas  $\sigma^2$  en los estados de los sistema, al igual el tener intervalos de probabilidad de los parámetros, y también poder realizar la estimación de la distribución predictiva por medio de las bandas de probabilidad mostradas. Es en estos resultados donde se ve la importancia de la metodología mostrada y no simplemente realizar estimación de parámetros vía mínimos cuadrados.

Algo que debe de quedar claro con el análisis presentado es que los métodos MCMC **no** son cajas negras que van a arrojar resultados positivos en todos los escenarios o modelos de interés, y no nos debemos quedar con un solo método para todos los problemas que vayamos a enfrentar, sino que uno debe de elegir adecuadamente el método MCMC que más se adapte al modelo matemático que se planea estudiar, esto para poder hacer más eficiente el análisis que se propone hacer.

Aunque se cumplió el objetivo planteado de esta tesis, al comparar las ventajas y desventajas de los métodos estocásticos de interés, aún hay mucho espacio para ampliar lo que se trabajó en esta tesis. Un ejemplo de lo anterior, sería utilizar modelos de ecuaciones en derivadas parciales o sistemas de ecuaciones más complejos. Otra oportunidad de interés podría ser la aplicación de procesos gaussianos para estos modelos, es por eso que se recomienda una ampliación de los estudios hechos en esta tesis en trabajos futuros.



# Bibliografía

- [1] Amani A. Alahmadi, Jennifer A. Flegg, Davis G. Cochrane, Christopher C. Drovandi, and Jonathan M. Keith. A comparison of approximate versus exact techniques for bayesian parameter inference in nonlinear ordinary differential equation models. *Royal Society Open Science*, 7(3):Article number: 191315, March 2020.
- [2] Christophe Andrieu and Eric Moulines. On the ergodic properties of some adaptive mcmc algorithms. *The Annals of Applied Probability*, 16, 11 2006.
- [3] J.M. Bernardo and A.F.M. Smith. *Bayesian Theory*. Wiley Series in Probability and Statistics. Wiley, 2007.
- [4] G.E.P. Box and G.C. Tiao. *Bayesian inference in statistical analysis*. Addison-Wesley series in behavioral science: quantitative methods. Addison-Wesley Pub. Co., 1973.
- [5] Campbell S.L. Brenan, K. E. and L. R. Petzold. *Numerical solution of initial-value problems in differential-algebraic equations*. Philadelphia: Society for Industrial and Applied Mathematics, 1996.
- [6] Steve Brooks, Andrew Gelman, Galin Jones, and Xiao-Li Meng. *Handbook of Markov Chain Monte Carlo*. CRC press, 2011.
- [7] A.M.K. Hein E. Turro, N. Bochkina and S. Richardson. Bgx: a bioconductor package for the bayesian integrated analysis of affymetrix genechips. *BMC Bioinformatics*, 8:439448, url=<http://www.biomedcentral.com/1471-2105/8/439>, 2007.
- [8] N. I. Fisher. *Statistical Analysis of Circular Data*. Cambridge University Press, 1993.
- [9] Andrew Gelman, John B. Carlin, Hal S. Stern, and Donald B. Rubin. *Bayesian Data Analysis*. Chapman and Hall/CRC, 2nd ed. edition, 2004.
- [10] Neil Gershenfeld. *The Nature of Mathematical Modeling*. Cambridge University Press, USA, 1999.
- [11] Mark Girolami. Bayesian inference for differential equations. *Theoretical Computer Science*, 408:4–16, 11 2008.

- [12] Heikki Haario, Eero Saksman, and Johanna Tamminen. An adaptive Metropolis algorithm. *Bernoulli*, 7(2):223 – 242, 2001.
- [13] W. K. Hastings. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57(1):97–109, 04 1970.
- [14] Martin Haugh. Mcmc and bayesian modeling. *Columbia University*, 2017.
- [15] Peter D. Hoff. *A First Course in Bayesian Statistical Methods*. Springer Publishing Company, Incorporated, 1st edition, 2009.
- [16] H. Jeffreys. *Theory of Probability*. Oxford, Oxford, England, third edition, 1961.
- [17] Jari Kaipio and Erkki Somersalo. *Statistical and computational inverse problems*. Springer, 2004.
- [18] Robert E. Kass and Adrian E. Raftery. Bayes factors. *Journal of the American Statistical Association*, 90(430):773–795, 1995.
- [19] Robert E. Kass and Larry Wasserman. The selection of prior distributions by formal rules. *Journal of the American Statistical Association*, 91(435):1343–1370, 1996.
- [20] K.V. Mardia and P.E. Jupp. *Directional Statistics*. Wiley Series in Probability and Statistics. Wiley, 2009.
- [21] Cody McRae. Bayesian inference in nonlinear differential equation models. *Monash University*, 2019.
- [22] N. Metropolis. The beginning of the monte carlo method. *Los Alamos Science Special Issue*, 15:125–130, 1987.
- [23] Linda Petzold. Automatic selection of methods for solving stiff and nonstiff systems of ordinary differential equations. *SIAM journal on scientific and statistical computing*, 4(1):136–148, 1983.
- [24] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2013.
- [25] Christian P. Robert. *The Bayesian choice: a decision-theoretic motivation*. Springer-Verlag, 1994.
- [26] Christian P. Robert. The metropolis-hastings algorithm, 2016.
- [27] G.O. Roberts and J.S. Rosenthal. Coupling and ergodicity of adaptive mcmc. *Journal of Applied Probability*, 2007.
- [28] G.O. Roberts and J.S. Rosenthal. Examples of adaptive mcmc. *Journal of Computational and Graphical Statistics*, (18(2)):349–367, 2009.

- [29] Benjamin Rosenbaum, Michael Raatz, Guntram Weithoff, Gregor F. Fussmann, and Ursula Gaedke. Estimating parameters from multiple time series of population dynamics using bayesian inference. *Frontiers in Ecology and Evolution*, 6:234, 2019.
- [30] D.B. Rowe. *Multivariate Bayesian Statistics: Models for Source Separation and Signal Unmixing*. (1st ed.). CRC Press., 2002.
- [31] Biswa Sengupta, Karl J Friston, and Will D Penny. Gradient-based mcmc samplers for dynamic causal modelling. *NeuroImage (Orlando, Fla.)*, 125:1107–1118, 2016.
- [32] Penny WD. Sengupta B, Friston KJ. Efficient gradient computation for dynamical models. *Neuroimage.*, pages 98(100):521–527, 2014.
- [33] Claude E. Shannon. A mathematical theory of communication. *Bell Syst. Tech. J.*, 27(3):379–423, 1948.
- [34] Karline Soetaert, Thomas Petzoldt, and R. Woodrow Setzer. Solving differential equations in R: Package deSolve. *Journal of Statistical Software*, 33(9):1–25, 2010.
- [35] Balding D. J. Griffiths R. C. Tavaré, S. and P. Donnelly. Inferring coalescence times from dna sequence data. *Genetics*, 145:505–518, 1997.
- [36] Samuel Thomas. *hmclearn: Fit Statistical Models Using Hamiltonian Monte Carlo*, 2020. R package version 0.0.4.
- [37] Samuel Thomas and Wanzhu Tu. Learning Hamiltonian Monte Carlo in R, 2020.
- [38] M. Vihola. Robust adaptive metropolis algorithm with coerced acceptance rate. *Stat Comput* 22, page 997–1008, 2012.
- [39] Matti Vihola. On the stability and ergodicity of adaptive scaling metropolis algorithms. *Stochastic Processes and their Applications*, 121(12):2839–2860, Dec 2011.
- [40] G.O. Roberts W.R. Gilks and S.K. Sahu. Adaptive markov chain monte carlo. *Amer. Stat. Assoc.*, 93:1045–1054, 1998.

# Apéndice A

## Códigos en R

A continuación presentaremos los scripts escritos en R para las simulaciones que se realizaron en el capítulo 4 de esta tesis, así como una breve introducción para la programación en paralelo en R.

### Código RAM para EDO

```
library(deSolve)
library(bbmle)
library(emdbook)
library(LaplacesDemon)
library(ramcmc)
library(TeachingDemos)
library(modeest)

# RAM Script General:

# Sistema EDO

McræeODE = function(t, y, parms) {
  # Variables del sistema de Ecuaciones Diferenciales
  V = y[1]
  R = y[2]
  # Parámetros del vector "parms"
  alpha = parms["alpha"]
  beta = parms["beta"]
  gamma = parms["gamma"]

  # Sistema de EDO
  dV = gamma*(R+V-((V^3)/3))
  dR = -(1/gamma)*(V-alpha+beta*R)

  res = c(dV,dR)
  # Return lista de gradientes
  list(res)
}

# Numero de datos
n<-40
# Tiempo de la ecuación diferencial
t<-20
times = seq(0, t, by = .5)
parms = c(alpha=0.3,beta=0.3,gamma=2)
start = c(V=-1,R=1)

out=ode(y=start, times=times, func=McræeODE, parms=parms)
out=as.data.frame(out)
```

```

# Creación de nuestros datos V y R
V<-out$V[2:41]+rnorm(20,0,.5)
R<-out$R[2:41]+rnorm(20,0,.5)

dsim.V.T<-out$V
dsim.R.T<-out$R

#Verosimilitud
L<-function(n,t,theta,y1,y2){
  times = seq(0, t, by = .5)
  parms = c(alpha=exp(theta[1]),beta=exp(theta[2]),gamma=exp(theta[3]))
  start = c(V=theta[4],R=theta[5])
  out=ode(y=start, times=times, func=McræODE, parms=parms,atol = 10^-6, rtol = 10^-6)
  out=as.data.frame(out)
  dsim.V<-out$V[2:(n+1)]
  dsim.R<-out$R[2:(n+1)]
  a<-dmvn(y1,mu=dsim.V,Sigma=exp(theta[6])*diag(n))*dmvn(y2,mu=dsim.R,Sigma=exp(theta[7])*diag(n))
  return(a)
}

#Dn Final
dist.f<-function(n,t,l.theta,y1,y2){
  #Verosimilitud y Dn inicial Parametros.
  a<-L(n,t,l.theta,y1,y2)*dnorm(l.theta[1],log(0.5),3)*dnorm(l.theta[2],log(0.5),3)*dnorm(l.theta[3],log(5),3)
  #Dn Valores Iniciales
  b<-dnorm(l.theta[4])*dnorm(l.theta[5])
  #Dn Sigmas
  c<-dnorm(l.theta[6],log(5),5)*dnorm(l.theta[7],log(5),5)
  d<-a*b*c
  return(d)
}

### Paralelo:

library(parallel)
detectCores()
numCores <- detectCores()
cl <- makeCluster(numCores)

clusterEvalQ(cl,library(deSolve))
clusterEvalQ(cl,library(bbmlé))
clusterEvalQ(cl,library(emdbook))
clusterEvalQ(cl,library(LaplacesDemon))
clusterEvalQ(cl,library(ramcmc))
clusterEvalQ(cl,library(TeachingDemos))
clusterEvalQ(cl,library(modeest))

clusterEvalQ(cl,
  # Sistema EDO
  McræODE<-function(t, y, parms) {
    # Variables del sistema de Ecuaciones Diferenciales
    V = y[1]
    R = y[2]
    # Parametros del vector "parms"
    alpha = parms["alpha"]
    beta = parms["beta"]
    gamma = parms["gamma"]

    #Sistema de EDO
    dV = gamma*(R+V-((V^3)/3))
    dR = -(1/gamma)*(V-alpha+beta*R)

    res = c(dV,dR)
    # Return lista de gradientes
    list(res)
  }
)

#Numero de datos
clusterEvalQ(cl,n<-40)

```

```

#Tiempo de la ecuación diferencial
clusterEvalQ(c1,t<-20)

#Verosimilitud
clusterEvalQ(c1,
  L<-function(n,t,theta,y1,y2){
    times = seq(0, t, by = .5)
    parms = c(alpha=exp(theta[1]),beta=exp(theta[2]),gamma=exp(theta[3]))
    start = c(V=theta[4],R=theta[5])
    out=ode(y=start, times=times, func=McraeODE, parms=parms,atol = 10^-6, rtol = 10^-6)
    out=as.data.frame(out)
    dsim.V<-out$V[2:(n+1)]
    dsim.R<-out$R[2:(n+1)]
    a<-dmvn(y1,mu=dsim.V,Sigma=exp(theta[6])*diag(n))*dmvn(y2,mu=dsim.R,Sigma=exp(theta[7])*diag(n))
    return(a)
  }
)

#Dn Final
clusterEvalQ(c1,
  dist.f<-function(n,t,l.theta,y1,y2){
    #Verosimilitud y Dn inicial Parametros.
    a<-L(n,t,l.theta,y1,y2)*dnorm(l.theta[1],log(0.5),3)*dnorm(l.theta[2],log(0.5),3)*dnorm(l.theta[3],log
    #Dn Valores Iniciales
    b<-dnorm(l.theta[4])*dnorm(l.theta[5])
    #Dn Sigmas
    c<-dnorm(l.theta[6],log(5),5)*dnorm(l.theta[7],log(5),5)
    d<-a*b*c
    return(d)
  }
)

#Parametros Libres
clusterEvalQ(c1,NP<-7)
## V y R son los datos en general
clusterEvalQ(c1,x<-c(rnorm(1,log(0.5),3),rnorm(1,log(0.5),3),rnorm(1,log(5),3),
  rnorm(1),rnorm(1),rnorm(1,log(5),5),rnorm(1,log(5),5))
)
clusterExport(c1,"V")
clusterExport(c1,"R")

clusterEvalQ(c1,while(dist.f(40,20,x,V,R)==0){
  x<-c(rnorm(1,log(0.5),3),rnorm(1,log(0.5),3),rnorm(1,log(5),3),
    rnorm(1),rnorm(1),rnorm(1,log(5),5),rnorm(1,log(5),5))
}
)

clusterEvalQ(c1,t<-20)
clusterEvalQ(c1,burn<-25000)
clusterEvalQ(c1,ite.inter<-25000)
clusterEvalQ(c1,lag<-60)
clusterEvalQ(c1,tmues<-300)
clusterEvalQ(c1,n<-40)

clusterEvalQ(c1,ite<-burn+ite.inter+lag*tmues)
clusterEvalQ(c1,burn.tot<-burn+ite.inter)

clusterEvalQ(c1,y.a.ln<-rep(NA,tmues))
clusterEvalQ(c1,y.b.ln<-rep(NA,tmues))
clusterEvalQ(c1,y.g.ln<-rep(NA,tmues))
clusterEvalQ(c1,y.v0.ln<-rep(NA,tmues))
clusterEvalQ(c1,y.r0.ln<-rep(NA,tmues))
clusterEvalQ(c1,y.s1.ln<-rep(NA,tmues))
clusterEvalQ(c1,y.s2.ln<-rep(NA,tmues))
clusterEvalQ(c1,n.sample<-0)

clusterEvalQ(c1,S<-diag(NP))

## Algoritmo general:

```

```

# For en cada nucleo:
start_time <- Sys.time()
clusterEvalQ(cl,for(j in 1:ite){
  u<-c(rmvn(1,c(0,0,0,0,0,0,0),diag(NP)))
  y0<-c(x)
  y1<-c(x+S%*%u)

  alpha<-min(1,dist.f(n,t,y1,V,R)/dist.f(n,t,y0,V,R))
  unif<-runif(1)

  if(unif<=alpha){x<-c(y1)}
  if(unif>alpha){x<-c(y0)}

  # Actualización de S
  S <- adapt_S(S, u,current=alpha,n=j)

  if(j>burn.tot & (j%lag)==0){
    n.sample<-n.sample+1
    y.a.ln[n.sample]<-exp(x[1])
    y.b.ln[n.sample]<-exp(x[2])
    y.g.ln[n.sample]<-exp(x[3])
    y.v0.ln[n.sample]<-x[4]
    y.r0.ln[n.sample]<-x[5]
    y.s1.ln[n.sample]<-exp(x[6])
    y.s2.ln[n.sample]<-exp(x[7])
  }
})
end_time <- Sys.time()

end_time - start_time

### Obtuve 8 sub-muestras de 300 valores.

A<-matrix(data=NA,nrow=2400,ncol=7)

## Guardo datos en matriz B por core
clusterEvalQ(cl,B<-matrix(data=NA,nrow=300,ncol=7))
clusterEvalQ(cl,for(i in 1:300){
  B[i,1]<-y.a.ln[i]
  B[i,2]<-y.b.ln[i]
  B[i,3]<-y.g.ln[i]
  B[i,4]<-y.v0.ln[i]
  B[i,5]<-y.r0.ln[i]
  B[i,6]<-y.s1.ln[i]
  B[i,7]<-y.s2.ln[i]
})

# Observar los resultados del primer core
clusterEvalQ(cl,B)[[1]]

# Guardamos los datos en la matriz A del proceso general
for(j in 1:8){
  for(i in 1:7){
    for(k in 1:300){
      A[k+(300*(j-1)),i]<-clusterEvalQ(cl,B)[[j]][k,i]
    }
  }
}

stopCluster(cl)

```

## Código HMC para EDO

```

library(deSolve)
library(bbmle)
library(emdbook)
library(LaplacesDemon)
library(ramcmc)

```

```

library(modeest)
library(numDeriv)
library(hmclearn)

## HMC script general

McræODE = function(t, y, parms) {
  # Variables del sistema de Ecuaciones Diferenciales
  V = y[1]
  R = y[2]
  # Parámetros del vector "parms"
  alpha = parms["alpha"]
  beta = parms["beta"]
  gamma = parms["gamma"]

  #Sistema de EDO
  dV = gamma*(R+V-((V^3)/3))
  dR = -(1/gamma)*(V-alpha+beta*R)

  res = c(dV,dR)
  # Return lista de gradientes
  list(res)
}

t<-20
#Tiempo inicial, hasta cuando y secuencia de pasos
times = seq(0, t, by = .1)
# Parametros
parms = c(alpha=0.3,beta=0.3,gamma=2)

# Valores iniciales
start = c(V=-1,R=1)
out=ode(y=start, times=times, func=McræODE, parms=parms)
out=as.data.frame(out)

# Creación de nuestros datos V y R
V<-out$V[2:41]+rnorm(40,0,.5)
R<-out$R[2:41]+rnorm(40,0,.5)

# Escogemos los datos simulados de y1 y y2:

dsim.V.T<-out$V[2:41]
dsim.R.T<-out$R[2:41]

# Definimos la función ll (log-verosimilitud)

ll<-function(n,t,theta,y1,y2){
  times = seq(0, t, by = .5)
  parms = c(alpha=exp(theta[1]),beta=exp(theta[2]),gamma=exp(theta[3]))
  start = c(V=theta[4],R=theta[5])
  out=ode(y=start, times=times, func=McræODE, parms=parms,atol = 10^-6, rtol = 10^-6)
  out=as.data.frame(out)
  dsim.V<-out$V[2:(n+1)]
  dsim.R<-out$R[2:(n+1)]
  ll = sum(dmvn(y1, dsim.V, Sigma=exp(theta[6])*diag(n),log=TRUE))+sum(dmvn(y2,dsim.R,Sigma=exp(theta[7])*diag(n),log=TRUE))
  return(ll)
}

# Log Dn Final

l.dist.f<-function(n,t,l.theta,y1,y2){
  #Verosimilitud y Dn inicial Parametros.
  a<-ll(n,t,l.theta,y1,y2)+log(dnorm(l.theta[1],log(0.5),3))+log(dnorm(l.theta[2],log(0.5),3))+log(dnorm(l.theta[3],log(0.5),3))
  #Dn Valores Iniciales
  b<-log(dnorm(l.theta[4]))+log(dnorm(l.theta[5]))
  #Dn Sigmas
  c<-log(dnorm(l.theta[6],log(5),10))+log(dnorm(l.theta[7],log(5),10))
  d<-a+b+c
  return(d)
}

```



```

}

tmues<-1000

x<-c(rnorm(1,log(0.5),3),rnorm(1,log(0.5),3),rnorm(1,log(5),3),
     rnorm(1),rnorm(1),rnorm(1,log(5),10),rnorm(1,log(5),10))

# Función de log Dn final dependiendo de los parámetros
f1<-function(x){
  a<-l.dist.f(n=40,t=20,x,y1=V,y2=R)
  return(a)
}
# Función de gradiente numerico de log Dn final
f2<-function(x){
  a<-grad(l.dist.f,x,method="simple",n=40,t=20,y1=V,y2=R)
  return(a)
}

# Matriz que va a guardar los datos
C1.hmc<-matrix(NA,nrow=1000,ncol=7)

# Función hmc de la libreria hmclean, tiene el algoritmo ya optimizado
start_time <- Sys.time()
b.C1<-hmc(N=10000,theta.init=x,epsilon = 0.01,L=20,f1,f2)
end_time <- Sys.time()

end_time - start_time

# Guardamos los datos
j<-1
for(k in 9001:10000){
  C1.hmc[j,1]<-exp(b.C1$theta.all[[1]][[k]][1])
  C1.hmc[j,2]<-exp(b.C1$theta.all[[1]][[k]][2])
  C1.hmc[j,3]<-exp(b.C1$theta.all[[1]][[k]][3])
  C1.hmc[j,4]<-b.C1$theta.all[[1]][[k]][4]
  C1.hmc[j,5]<-b.C1$theta.all[[1]][[k]][5]
  C1.hmc[j,6]<-exp(b.C1$theta.all[[1]][[k]][6])
  C1.hmc[j,7]<-exp(b.C1$theta.all[[1]][[k]][7])
  j<-j+1
}

```

## Código MCMC-ABC para EDO

```

library(deSolve)
library(bbmle)
library(emdbook)
library(LaplacesDemon)
library(ramcmc)
library(TeachingDemos)
library(modeest)

# McMC-ABC Script General:

# Sistema EDO

McræeODE = function(t, y, parms) {
  # Variables del sistema de Ecuaciones Diferenciales
  V = y[1]
  R = y[2]
  # Parametros del vector "parms"
  alpha = parms["alpha"]
  beta = parms["beta"]
  gamma = parms["gamma"]

  #Sistema de EDO
  dV = gamma*(R+V-((V^3)/3))
  dR = -(1/gamma)*(V-alpha+beta*R)

```

```

    res = c(dV,dR)
    # Return lista de gradientes
    list(res)
}

# Numero de datos
n<-40
# Tiempo de la ecuación diferencial
t<-20
times = seq(0, t, by = .5)
parms = c(alpha=0.3,beta=0.3,gamma=2)
start = c(V=-1,R=1)

out=ode(y=start, times=times, func=McraeODE, parms=parms)
out=as.data.frame(out)

# Creación de nuestros datos V y R
V<-out$V[2:41]+rnorm(20,0,.5)
R<-out$R[2:41]+rnorm(20,0,.5)

dsim.V.T<-out$V
dsim.R.T<-out$R

### Paralelo:

library(parallel)
detectCores()
numCores <- detectCores()
cl <- makeCluster(numCores)

clusterEvalQ(cl,library(deSolve))
clusterEvalQ(cl,library(bbmle))
clusterEvalQ(cl,library(emdbook))
clusterEvalQ(cl,library(LaplacesDemon))
clusterEvalQ(cl,library(ramcmc))
clusterEvalQ(cl,library(TeachingDemos))
clusterEvalQ(cl,library(modeest))

clusterEvalQ(cl,
  # Sistema EDO
  McraeODE<-function(t, y, parms) {
    # Variables del sistema de Ecuaciones Diferenciales
    V = y[1]
    R = y[2]
    # Parametros del vector "parms"
    alpha = parms["alpha"]
    beta = parms["beta"]
    gamma = parms["gamma"]

    #Sistema de EDO
    dV = gamma*(R+V-((V^3)/3))
    dR = -(1/gamma)*(V-alpha+beta*R)

    res = c(dV,dR)
    # Return lista de gradientes
    list(res)
  }
)

#Numero de datos
clusterEvalQ(cl,n<-40)

#Dn Inicial NO INFORMATIVAS
clusterEvalQ(cl,
  dist.i<-function(l.theta){
    #Dn inicial Parametros.
    a<-dnorm(l.theta[1],0,2)*dnorm(l.theta[2],0,2)*dnorm(l.theta[3],0,2)
    #Dn Valores Iniciales
    b<-dnorm(l.theta[4],0,2)*dnorm(l.theta[5],0,2)

```

```

        #Dn sigmas
        c<-dnorm(1.theta[6],0,2)*dnorm(1.theta[7],0,2)
        d<-a*b*c
        return(d)
    }
)

### V y R son los datos en general
clusterExport(cl,"V")
clusterExport(cl,"R")

clusterEvalQ(cl, burn<-150000)
clusterEvalQ(cl, ite.inter<-150000)
clusterEvalQ(cl, lag<-750)
clusterEvalQ(cl, tmues<-200)
clusterEvalQ(cl, n<-40)

clusterEvalQ(cl, ite<-burn+ite.inter+lag*tmues)
clusterEvalQ(cl, burn.tot<-burn+ite.inter)

clusterEvalQ(cl, z.a<-rep(NA, tmues))
clusterEvalQ(cl, z.b<-rep(NA, tmues))
clusterEvalQ(cl, z.g<-rep(NA, tmues))
clusterEvalQ(cl, z.v0<-rep(NA, tmues))
clusterEvalQ(cl, z.r0<-rep(NA, tmues))
clusterEvalQ(cl, z.s1<-rep(NA, tmues))
clusterEvalQ(cl, z.s2<-rep(NA, tmues))
clusterEvalQ(cl, n.sample<-0)

clusterEvalQ(cl, eps<-0.5)

clusterEvalQ(cl, theta0<-FALSE)
clusterEvalQ(cl, while(theta0==FALSE){
  x<-c(rnorm(1,0,2), rnorm(1,0,2), rnorm(1,0,2),
       rnorm(1,0,2), rnorm(1,0,2), rnorm(1,0,2), rnorm(1,0,2))

  # Tiempo de la ecuación diferencial
  t<-20
  times = seq(0, t, by = .5)
  parms = c(alpha=exp(x[1]), beta=exp(x[2]), gamma=exp(x[3]))
  start = c(V=x[4], R=x[5])

  out2=ode(y=start, times=times, func=McraeODE, parms=parms)
  out2=as.data.frame(out2)
  dsim.V<-out2$V[2:(n+1)] + rnorm(n,0,exp(x[6]))
  dsim.R<-out2$R[2:(n+1)] + rnorm(n,0,exp(x[7]))

  d1<-(1/n)*sum((dsim.V-V)^2)
  d2<-(1/n)*sum((dsim.R-R)^2)

  if(d1<eps && d2<eps){
    theta0<-TRUE
    x0<-x
  }
}
)

## Algoritmo General:
start_time <- Sys.time()
clusterEvalQ(cl, for(i in 1:ite){

  x1<-rmvn(1,x0,1.5*diag(7))

  r<-(dist.i(x1)*dmvn(x0,x1,1.5*diag(7)))/(dist.i(x0)*dmvn(x1,x0,1.5*diag(7)))
  u<-runif(1)

  if(u<r){
    #Tiempo de la ecuación diferencial

```

```

t<-20
times = seq(0, t, by = .5)
parms = c(alpha=exp(x1[1]),beta=exp(x1[2]),gamma=exp(x1[3]))
start = c(V=x1[4],R=x1[5])

out2=ode(y=start, times=times, func=McraeODE, parms=parms)
out2=as.data.frame(out2)
dsim.V<-out2$V[2:(n+1)] + rnorm(n,0,exp(x[6]))
dsim.R<-out2$R[2:(n+1)] + rnorm(n,0,exp(x[7]))

d1<-(1/n)*sum((dsim.V-V)^2)
d2<-(1/n)*sum((dsim.R-R)^2)

if(d1<eps && d2<eps){x<-x1}
else{x<-x0}
}
else{x<-x0}

if(i>burn.tot & (i%%lag)==0){
n.sample<-n.sample+1
z.a[n.sample]<-exp(x[1])
z.b[n.sample]<-exp(x[2])
z.g[n.sample]<-exp(x[3])
z.v0[n.sample]<-x[4]
z.r0[n.sample]<-x[5]
z.s1[n.sample]<-exp(x[6])
z.s2[n.sample]<-exp(x[7])
}

x0<-x
})
end_time <- Sys.time()

end_time - start_time

### Obtuve 8 muestras de 300 valores.

Z<-matrix(data=NA,nrow=1600,ncol=7)

## Guardo datos en matriz B por core
clusterEvalQ(c1,B<-matrix(data=NA,nrow=200,ncol=7))
clusterEvalQ(c1,for(i in 1:200){
B[i,1]<-z.a[i]
B[i,2]<-z.b[i]
B[i,3]<-z.g[i]
B[i,4]<-z.v0[i]
B[i,5]<-z.r0[i]
B[i,6]<-z.s1[i]
B[i,7]<-z.s2[i]
})

# Guardamos los datos en la matriz Z del proceso general
for(j in 1:8){
for(i in 1:7){
for(k in 1:200){
Z[k+(200*(j-1)),i]<-clusterEvalQ(c1,B)[[j]][k,i]
}}
}

stopCluster(c1)

```

## Código RAM para Datos Circulares

```

library(deSolve)
library(bbmle)
library(emdbook)
library(LaplacesDemon)

```

```

library(ramcmc)
library(TeachingDemos)
library(modeest)

## RAM datos circulares script general:

# Verosimilitud
L<-function(n,mu,y1){
  a<-rep(0,n)
  b<-rep(0,n)
  C6<-exp(-.5*((mu[1]^2)+(mu[2]^2)))/(2*pi)
  for(i in 1:n){
    b[i]<-(mu[1]*cos(y1[i]))+(mu[2]*sin(y1[i]))
    a[i]<-C6*(1+(b[i]*pnorm(b[i])/dnorm(b[i])))
  }
  l<-prod(a)
  return(l)
}

# Dn final
dist.f<-function(n,mu,y1){
  #Verosimilitud y Dn inicial Parametros.
  a<-L(n,mu,y1)*dnorm(mu[1],0,5)*dnorm(mu[2],0,5)
  return(a)
}

## Paralelo
library(parallel)
detectCores()
numCores <- detectCores()
cl <- makeCluster(numCores)

clusterEvalQ(cl,library(deSolve))
clusterEvalQ(cl,library(bbmlE))
clusterEvalQ(cl,library(emdbook))
clusterEvalQ(cl,library(LaplacesDemon))
clusterEvalQ(cl,library(ramcmc))
clusterEvalQ(cl,library(TeachingDemos))
clusterEvalQ(cl,library(modeest))

#Numero de datos
clusterEvalQ(cl,n<-100)

#Verosimilitud
clusterEvalQ(cl,
  L<-function(n,mu,y1){
    a<-rep(0,n)
    b<-rep(0,n)
    C6<-exp(-.5*((mu[1]^2)+(mu[2]^2)))/(2*pi)
    for(i in 1:n){
      b[i]<-(mu[1]*cos(y1[i]))+(mu[2]*sin(y1[i]))
      a[i]<-C6*(1+(b[i]*pnorm(b[i])/dnorm(b[i])))
    }
    l<-prod(a)
    return(l)
  }
)

#Dn Final
clusterEvalQ(cl,
  dist.f<-function(n,mu,y1){
    #Verosimilitud y Dn inicial Parametros.
    a<-L(n,mu,y1)*dnorm(mu[1],0,5)*dnorm(mu[2],0,5)
    return(a)
  }
)

#Parametros Libres
clusterEvalQ(cl,NP<-2)

```

```

# Exportamos nuestros datos "theta1"
clusterExport(cl,"theta1")

# Valores iniciales
clusterEvalQ(cl,x<-c(rnorm(2,0,5)))

clusterEvalQ(cl,burn<-25000)
clusterEvalQ(cl,ite.inter<-25000)
clusterEvalQ(cl,lag<-50)
clusterEvalQ(cl,tmues<-200)
clusterEvalQ(cl,n<-100)

clusterEvalQ(cl,ite<-burn+ite.inter+lag*tmues)
clusterEvalQ(cl,burn.tot<-burn+ite.inter)

clusterEvalQ(cl,y.mu1.r<-rep(NA,tmues))
clusterEvalQ(cl,y.mu2.r<-rep(NA,tmues))
clusterEvalQ(cl,n.sample<-0)

clusterEvalQ(cl,S<-diag(NP))

## Algoritmo General:
# For en cada nucleo:
start_time <- Sys.time()
clusterEvalQ(cl,for(j in 1:ite){
  u<-c(rmvn(1,c(0,0),diag(NP)))
  y0<-c(x)
  y1<-c(x+S%*%u)

  alpha<-min(1,dist.f(n,y1,theta1)/dist.f(n,y0,theta1))
  unif<-runif(1)

  if(unif<=alpha){x<-c(y1)}
  if(unif>alpha){x<-c(y0)}

  # Paso Adaptativo
  S <- adapt_S(S, u,current=alpha,n=j)

  if(j>burn.tot & (j%lag)==0){
    n.sample<-n.sample+1
    y.mu1.r[n.sample]<-x[1]
    y.mu2.r[n.sample]<-x[2]
  }
})
end_time <- Sys.time()

end_time - start_time

### Obtuve 8 muestras de 200 valores.

T1<-matrix(data=NA,nrow=1600,ncol=2)

## Guardo datos en matriz B por core
clusterEvalQ(cl,B<-matrix(data=NA,nrow=200,ncol=2))
clusterEvalQ(cl,for(i in 1:200){
  B[i,1]<-y.mu1.r[i]
  B[i,2]<-y.mu2.r[i]
})

# Guardamos los datos en la matriz T1 del proceso general
for(j in 1:8){
  for(i in 1:2){
    for(k in 1:200){
      T1[k+(200*(j-1)),i]<-clusterEvalQ(cl,B)[[j]][k,i]
    }
  }
}

stopCluster(cl)

```

# Código HMC para Datos Circulares

```
library(deSolve)
library(bbmle)
library(emdbook)
library(LaplacesDemon)
library(ramcmc)
library(modeest)
library(numDeriv)
library(hmclearn)
library(TeachingDemos)
library(REdaS)

## HMC datos circulares script general:

# Log-Verosimilitud
ll<-function(n,mu,y1){
  a<-rep(0,n)
  b1<-rep(0,n)
  C6<-(-.5*((mu[1]^2)+(mu[2]^2))-log(2*pi)
  for(i in 1:n){
    b1[i]<-(mu[1]*cos(y1[i]))+(mu[2]*sin(y1[i]))
    a[i]<-C6+log(1+(b1[i]*pnorm(b1[i])/dnorm(b1[i])))
  }
  ll<-sum(a)
  return(ll)
}

# Log Dn Final
l.dist.f<-function(n,mu,y1){
  #Verosimilitud y Dn inicial Parametros.
  a<-ll(n,mu,y1)+log(dnorm(mu[1],0,5))+log(dnorm(mu[2],0,5))
  return(a)
}

# Valores iniciales
x<-c(rnorm(2,0,5))

# Tamaño de muestra y número de datos
tmues<-1000
n<-100

y.mu1<-rep(NA,tmues)
y.mu2<-rep(NA,tmues)

# Función de log Dn final
f1<-function(x){
  a<-l.dist.f(100,x,theta1)
  return(a)
}

# Función gradiente numerico de log Dn final
f2<-function(x){
  a<-grad(l.dist.f,x,method="simple",n=100,y1=theta1)
  return(a)
}

# Función hmc de la libreria hmclear, tiene el algoritmo ya optimizado
start_time <- Sys.time()
b<-hmc(N=30000,theta.init=x,epsilon=0.01,L=10,f1,f2)
end_time <- Sys.time()

end_time - start_time

# Guardamos los datos con un lag=20
j<-1
for(k in 10001:30000){
  if(k%20==0){
    y.mu1[j]<-b$theta.all[[1]][[k]][1]
    y.mu2[j]<-b$theta.all[[1]][[k]][2]
  }
}
```

```

    j<-j+1
  }}

```

## Código ABC para Datos Circulares

```

library(deSolve)
library(bbmle)
library(emdbook)
library(LaplacesDemon)
library(ramcmc)
library(TeachingDemos)
library(modeest)

## ABC Alogritmo Aceptación-Rechazo datos circulares script general:

## Paralelo
library(parallel)
detectCores()
numCores <- detectCores()
cl <- makeCluster(numCores)

clusterEvalQ(cl,library(deSolve))
clusterEvalQ(cl,library(bbmle))
clusterEvalQ(cl,library(emdbook))
clusterEvalQ(cl,library(LaplacesDemon))
clusterEvalQ(cl,library(ramcmc))
clusterEvalQ(cl,library(TeachingDemos))
clusterEvalQ(cl,library(modeest))
clusterEvalQ(cl,library(CircStats))

# Numero de datos
clusterEvalQ(cl,n<-100)

# Exportamos nuestros datos "theta1"
clusterExport(cl,"theta1")

clusterEvalQ(cl,tmues<-200)

clusterEvalQ(cl,z.m1<-rep(NA,tmues))
clusterEvalQ(cl,z.m2<-rep(NA,tmues))

clusterEvalQ(cl,n.sample<-0)

# Epsilon
clusterEvalQ(cl,eps<-0.01)

## Algoritmo general:
start_time <- Sys.time()
clusterEvalQ(cl,while(n.sample<tmues){
  # Muestreando de las Dn iniciales:
  x<-c(rnorm(1,0,5),rnorm(1,0,5))

  theta.p<-rep(0,n)
  for(j in 1:n){
    a1<-rnorm(1,x[1],1)
    a2<-rnorm(1,x[2],1)

    theta.p[j]<-atan2(a2,a1)%%(2*pi)
  }

  # Primera medida de comparación:
  S<-(1/n)*sum(sin(theta1))
  C<-(1/n)*sum(cos(theta1))

```



```

if(C<0){theta.bar<-atan(S/C)+pi}
else{theta.bar<-atan(S/C)}

d<-(1/n)*sum(1-cos(theta.bar-theta.p))

# Segunda medida de comparación:
#d<-(1/(n^2))*sum(1-cos(sort(theta5)-sort(theta.p)))

if(d<eps){
  n.sample<-n.sample+1
  z.m1[n.sample]<-x[1]
  z.m2[n.sample]<-x[2]
}
}
)
end_time <- Sys.time()

end_time - start_time

ABC<-matrix(data=NA,nrow=1600,ncol=2)

## Guardo datos en matriz B por core
clusterEvalQ(cl,B<-matrix(data=NA,nrow=200,ncol=2))
clusterEvalQ(cl,for(i in 1:200){
  B[i,1]<-z.m1[i]
  B[i,2]<-z.m2[i]
})

# Guardamos los datos en la matriz ABC del proceso general
for(j in 1:8){
  for(i in 1:2){
    for(k in 1:200){
      ABC[k+(200*(j-1)),i]<-clusterEvalQ(cl,B)[[j]][k,i]
    }
  }
}

stopCluster(cl)

```

## Código de introducción a programación en paralelo

```

library(parallel)
detectCores()
numCores <- detectCores()
cl <- makeCluster(numCores)

clusterEvalQ(cl,2+2)

y<-c(1,2,3,4,5)
clusterExport(cl,"y")

clusterEvalQ(cl,y)

clusterEvalQ(cl,x<-rnorm(20,0,1))
clusterEvalQ(cl,x)

clusterEvalQ(cl,x)[[1]]
clusterEvalQ(cl,x)[[1]][1]

A<-matrix(data=NA,nrow=160,ncol=1)

for(j in 1:8){
  for(k in 1:20){

```

```
    A[k+(20*(j-1)),1]<-clusterEvalQ(cl,x)[[j]][k]
  }
summary(A)
plot(density(A))
stopCluster(cl)
```



Inferencia bayesiana en modelos de ecuaciones diferenciales: Un enfoque computacional.



ERICK IVAN GUERRERO FLORES  
ALUMNO

REVISÓ

MTRA. ROSALÍA SERRANO DE LA PAZ  
DIRECTORA DE SISTEMAS ESCOLARES

Con base en la Legislación de la Universidad Autónoma Metropolitana, en la Ciudad de México se presentaron a las 15:00 horas del día 25 del mes de agosto del año 2021 POR VÍA REMOTA ELECTRÓNICA, los suscritos miembros del jurado designado por la Comisión del Posgrado:

DR. JOAQUIN DELGADO FERNANDEZ  
DR. GABRIEL NUÑEZ ANTONIO  
DR. MIGUEL ANGEL MORELES VAZQUEZ  
DR. JOSE HECTOR MORALES BARCENAS

Bajo la Presidencia del primero y con carácter de Secretario el último, se reunieron para proceder al Examen de Grado cuya denominación aparece al margen, para la obtención del grado de:

MAESTRO EN CIENCIAS (MATEMÁTICAS APLICADAS E INDUSTRIALES)

DE: ERICK IVAN GUERRERO FLORES

y de acuerdo con el artículo 78 fracción III del Reglamento de Estudios Superiores de la Universidad Autónoma Metropolitana, los miembros del jurado resolvieron:

**A P R O B A R**

Acto continuo, el presidente del jurado comunicó al interesado el resultado de la evaluación y, en caso aprobatorio, le fue tomada la protesta.

DIRECTOR DE LA DIVISIÓN DE CBI

DR. JESUS ALBERTO OCHOA TAPIA

PRESIDENTE

DR. JOAQUIN DELGADO FERNANDEZ

VOCAL

DR. GABRIEL NUÑEZ ANTONIO

VOCAL

DR. MIGUEL ANGEL MORELES VAZQUEZ

SECRETARIO

DR. JOSE HECTOR MORALES BARCENAS

El presente documento cuenta con la firma -autógrafa, escaneada o digital, según corresponda- del funcionario universitario competente, que certifica que las firmas que aparecen en esta acta - Temporal, digital o dictamen- son auténticas y las mismas que usan los c.c. profesores mencionados en ella