

METODOS AUXILIARES DE MONTE CARLO

Eva Besada-Portas, Sergey M. Plis

A7-1.1 INTRODUCCIÓN

A continuación se describe el ejemplo que sirve para ilustrar el método de integración de Monte Carlo y el funcionamiento de los métodos auxiliares de muestreo que se presentan en los capítulos 7 y 8.

El problema elegido consiste en el cálculo de la concentración C media de un componente que se obtiene durante un proceso industrial cuyo comportamiento se describe con la densidad de probabilidad recogida la ecuación (A7-1.1). La concentración media se obtiene calculando el valor de $I = \iint Cp(C,T)dCdT$. Para realizar dicha operación se pueden seguir distintas estrategias.

La estrategia más sencilla consiste en calcular la concentración media en cada instante de tiempo $\bar{c}(T) = \int Cp(C,T)dC$ y a partir de la misma el valor de la integral $I = \int \left(\int Cp(C,T)dC \right) dT = \int \bar{c}(T)dT$, ya que debido a las características de la ecuación (A7-1.1) tanto la $\bar{c}(T)$ como la I admiten las expresiones analíticas recogidas en las ecuaciones (A7-1.2) y (A7-1.3). Por lo tanto, es posible calcular el

valor exacto de la concentración media para un conjunto de valores de $\{m_k, c_l, s_l | k=1:3, l=1:5\}$.

$$p(C, T) = \begin{cases} \mathcal{N}_C(m_1, c_1)/s_5 & 0 \leq T < s_1 \\ \mathcal{N}_C\left(m_1 + \frac{m_2 - m_1}{s_2 - s_1}(T - s_1), c_2\right)/s_5 & s_1 \leq T < s_2 \\ \mathcal{N}_C(m_2, c_3)/s_5 & s_2 \leq T < s_3 \\ \mathcal{N}_C\left(m_2 + \frac{m_3 - m_2}{s_4 - s_3}(T - s_3), c_4\right)/s_5 & s_3 \leq T < s_4 \\ \mathcal{N}_C(m_3, c_5)/s_5 & s_4 \leq T < s_5 \\ 0 & c.c. \end{cases} \quad (\text{A7-1.1})$$

$$\text{con } \mathcal{N}_C(a, b) = \frac{1}{\sqrt{2\pi b}} e^{-\frac{(C-a)^2}{2b}}$$

$$\bar{c}(T) = \begin{cases} m_1/s_5 & 0 \leq T < s_1 \\ \left(m_1 + \frac{m_2 - m_1}{s_2 - s_1}(T - s_1)\right)/s_5 & s_1 \leq T < s_2 \\ m_2/s_5 & s_2 \leq T < s_3 \\ \left(m_2 + \frac{m_3 - m_2}{s_4 - s_3}(T - s_3)\right)/s_5 & s_3 \leq T < s_4 \\ m_3/s_5 & s_4 \leq T < s_5 \\ 0 & c.c. \end{cases} \quad (\text{A7-1.2})$$

$$I = \frac{m_1 s_1 + a(s_2 - s_1) + m_2(s_3 - s_2) + b(s_4 - s_3) + m_3(s_5 - s_4)}{s_5} \quad (\text{A7-1.3})$$

$$\text{con } a = \left(\frac{m_2 + m_1}{2}\right) \quad \text{y} \quad b = \left(\frac{m_3 + m_2}{2}\right)$$

La segunda estrategia, basada parcialmente en la anterior y útil en aquellos casos en los que se disponga de una función $\bar{c}(T)$ cuya expresión integral no pueda ser calculada de forma exacta, consiste en calcular $I = \int \bar{c}(T) dT$ utilizando el método de integración de Monte Carlo. La forma mas sencilla de obtener las

muestras de tiempo $t^{(i)}$ es usar la distribución uniforme $\mathcal{U}_{(0,1500)}(T)$ en el intervalo en el que las concentraciones medias $\bar{c}(T)$ toman valores distintos de cero, ya que $I = \int \bar{c}(T) \mathcal{U}_{(0,1500)}(T) dT = \int \bar{c}(T) dT$.

Finalmente, la tercera estrategia propuesta en este apéndice se basa en la obtención de la función $p(C) = \int p(C, T) dT$ y el cálculo de la integral $I = \int Cp(C) dC$. Este caso es mucho mas complicado, ya que no es posible obtener muestras directamente a partir de la $p(C)$, cuya expresión analítica se presenta en la ecuación A7-1.4. Por lo tanto, calcularemos el valor $I = \int Cp(C) dC$ por el método de integración de Monte Carlo obteniendo las muestras a partir de los diferentes métodos auxiliares propuestos en los capítulos 7 y 8. Nótese que para poder aplicarlos, únicamente es necesario disponer de algún mecanismo que nos permita evaluar $p(C)$, hecho posible en este caso ya que los valores de $f_2(C)$ y $f_4(C)$ se encuentran tabulados.

$$p(C) = \frac{f_1(C) + f_2(C) + f_3(C) + f_4(C) + f_5(C)}{s_5} \quad (\text{A7-1.4})$$

$$\text{con } f_1(C) = s_1 \mathcal{N}_C(m_1, c_1)$$

$$f_2(C) = (s_2 - s_1) \int_{s_1}^{s_2} \mathcal{N}_C \left(m_1 + \frac{m_2 - m_1}{s_2 - s_1} (T - s_1), c_2 \right) dT$$

$$f_3(C) = (s_3 - s_2) \mathcal{N}_C(m_2, c_3)$$

$$f_4(C) = (s_4 - s_3) \int_{s_3}^{s_4} \mathcal{N}_C \left(m_2 + \frac{m_3 - m_2}{s_4 - s_3} (T - s_3), c_2 \right) dT$$

$$f_5(C) = (s_5 - s_4) \mathcal{N}_C(m_3, c_5)$$

En las siguientes secciones se presentan los datos y programas¹ de Matlab utilizados para implementar las diferentes estrategias y realizar un análisis de los resultados obtenidos por cada una de ellas. Las gráficas incluidas en el apéndice

¹ Al final de este documento se presenta una tabla explicativa de todos los ficheros de Matlab proporcionados. Aquellos cuyo nombre empieza por “Main” se corresponden a los diferentes programas principales explicados a lo largo del documento.

tienen como objeto mostrar al lector el programa que las genera. Su interpretación se incluye en las secciones correspondientes de los capítulos 7 y 8.

A7-1.2 DATOS DEL PROBLEMA

Los valores $\{m_k, c_l, s_l | k=1:3, l=1:5\}$ utilizados en los ejemplos de este apéndice y de los capítulos 7 y 8 se presentan en el conjunto de expresiones recogidas en la ecuación (A7-1.5).

$$\begin{aligned} m_1 &= 30, \quad m_2 = 70, \quad m_3 = 40 \\ c_1 &= 4, \quad c_2 = 2, \quad c_3 = 7, \quad c_4 = 2, \quad c_5 = 1 \\ s_1 &= 300, \quad s_2 = 600, \quad s_3 = 700, \quad s_4 = 1200, \quad s_5 = 1500 \end{aligned} \quad (\text{A7-1.5})$$

La definición de dichos valores se realiza en el programa “*Data.m*”. A partir de los mismos, y utilizando las expresiones recogidas en las ecuaciones (A7-1.1), (A7-1.2) y (A7-1.4) y el programa “*MainProblem.m*” se obtienen las representaciones gráficas de las figuras A7-1.1, A7-1.2, A7-1.3 y A7-1.4. La primera representa la densidad de probabilidad $p(C, T)$, la segunda el valor de la concentración $c(T)$ durante un experimento, la tercera el valor de la concentración media $\bar{c}(T)$ en cada instante de tiempo, y la cuarta la densidad marginal $p(C)$.

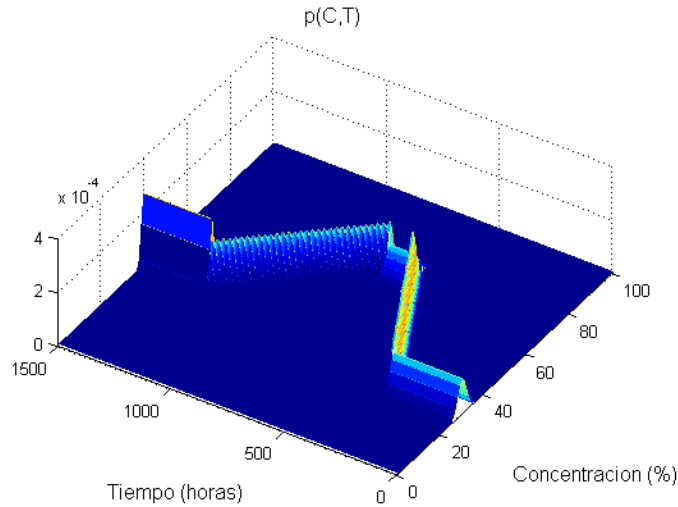


Figura A7-1.1 Función de probabilidad $p(C, T)$

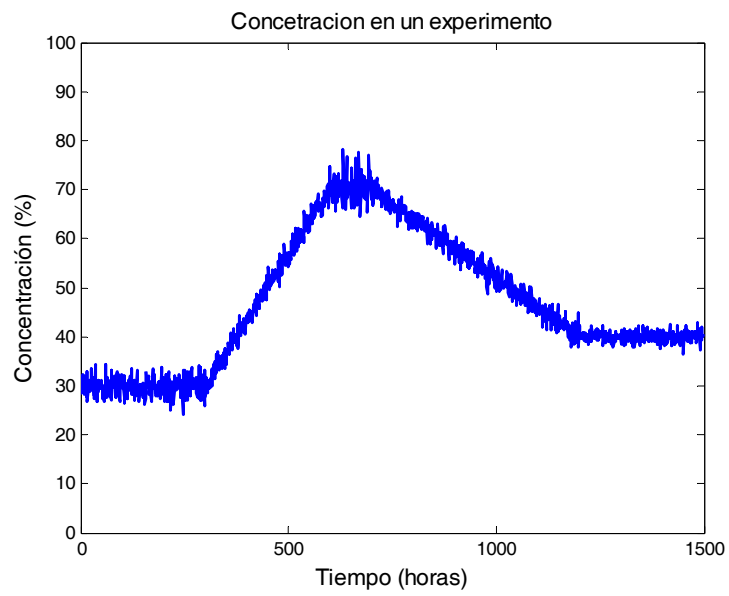


Figura A7-1.2 Concentración $c(T)$ durante un experimento

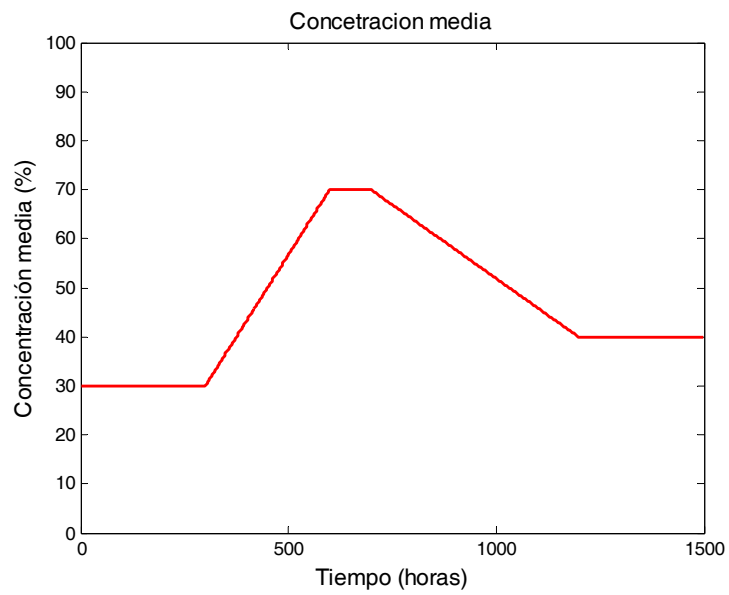


Figura A7-1.3 Concentración media $\bar{c}(T)$.

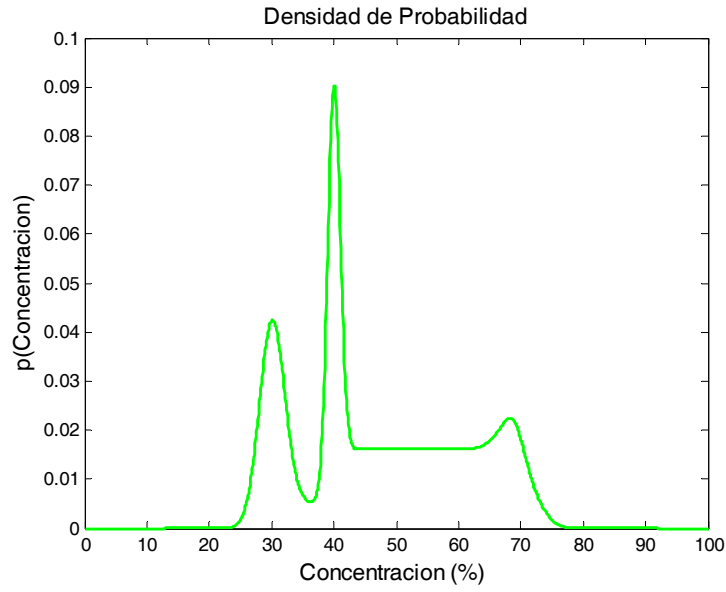


Figura A7-1.4 Densidad marginal $p(C)$

A7-1.3 CÁLCULO EXACTO DE LA INTEGRAL

En este problema particular y debido a la función $p(C, T)$ elegida es posible realizar el cálculo exacto del valor de la integral. Este hecho nos permite comparar el valor real con los valores obtenidos con el resto de las estrategias, las técnicas de integración de Monte Carlo y diferentes funciones de muestreo.

El programa “*MainExact.m*” calcula dicho valor en función de los valores $\{m_k, c_l, s_l | k=1:3, l=1:5\}$ recogidos en las expresiones de (A7-1.5) y en el programa “*Data.m*”. El valor teórico de la integral para los $\{m_k, c_l, s_l | k=1:3, l=1:5\}$ elegidos es $I = 47$.

A7-1.4 CÁLCULO APROXIMADO A PARTIR DE $\bar{c}(T)$

En este caso se genera un conjunto de N valores de $t^{(i)} \sim \mathcal{U}_{(0,1500)}(T)$ y se calcula el valor aproximado de la integral $I_N = \sum_{i=1}^N \bar{c}(t^{(i)})$.

El programa principal “*MainCMean.m*” realiza dicha operación para múltiples conjuntos (50 por defecto) de múltiples números de muestras ($N = \{1e2, 2e2, 5e2, 1e3, 2e3, 5e3, 1e4, 2e4, 5e4, 1e5, 2e5, 5e5, 1e6\}$ por defecto), salva en “*MainCMeanData.mat*” los valores de I_N generados en los diferentes casos, y representa el valor promedio de I_N obtenido sobre todos los conjuntos para los diferentes valores de N . El resultado se presenta en la figura A7-1.5. Los valores por defecto pueden ser modificados en el programa “*MainCMean.m*”. La generación de las muestras, el cálculo de la integral y su almacenamiento se realizan con la función “*LoopCMean.m*”. La representación gráfica se hace con la función “*DrawAnalysis.m*”.

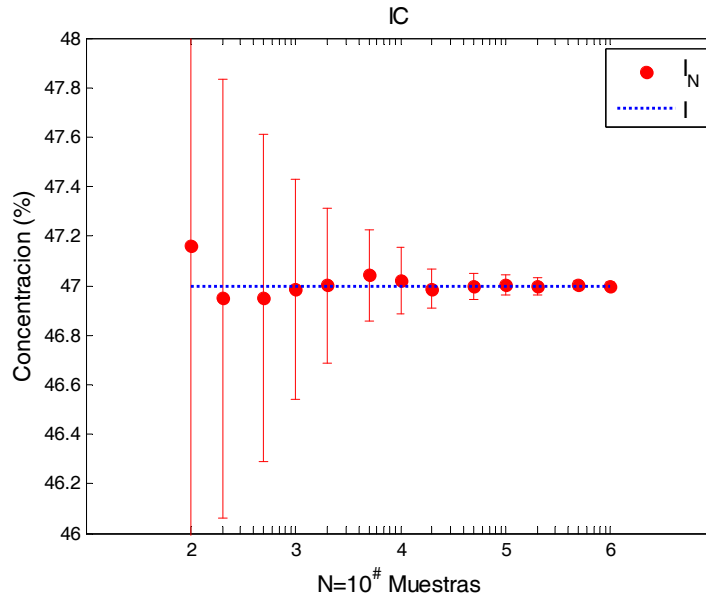


Figura A7-1.5 Gráfica de análisis del calculo integral a partir de $\bar{c}(T)$.

A7-1.5 CÁLCULO APROXIMADO A PARTIR DE $p(C)$

En este caso se generan los conjuntos de N valores $c^{(i)}$ y sus N pesos² $w(c^{(i)})$ a través de los procedimientos auxiliares propuestos en los capítulos 7 y 8 y se calcula el valor aproximado de la integral como $I_N = \sum_{i=1}^N w(c^{(i)})c^{(i)}$. Los métodos auxiliares utilizados son el muestreo por rechazo, el muestreo enfatizado, el remuestreo por pesos, y el muestreo mediante cadenas de Markov.

Cada uno de los cuatro métodos se implementa de forma genérica para problemas unidimensionales por medio de las funciones de Matlab “*RejectionSampling.m*” (muestreo por rechazo), “*ImportanceSampling.m*” (muestreo enfatizado), “*WeightedResampling.m*” (remuestreo por pesos) y “*MarkovChain.m*” (muestreo mediante cadenas de Markov). A las dos primeras funciones se les proporciona la función de muestreo original $p(C)$ “*pfun*” y la función de muestreo auxiliar $q(C)$ “*qfun*”. A la tercera se le proporciona la función muestreada original y sus pesos “*samplesinit*” y “*weightsinit*”, y el valor “*resample*” de la función $s(C)$ en las muestras originales. Finalmente, a la cuarta se le proporciona la función “*chainfun*” que implementa la cadena de Markov $K(C|c^{(i-1)})$. A todas ellas también se les proporciona un parámetro adicional “*draw*” que le indica al método de muestreo si además de obtener las muestras (y pesos en los métodos de muestreo enfatizado y remuestreo por pesos) es necesario realizar las representaciones gráficas asociadas a cada uno de los métodos.

La definición de las funciones (“*pfun*”, “*qfun*” y “*chainfun*”) se realiza por medio a una llamada a la función “*DefineFunction.m*” en la que se le proporciona el nombre de la función y los parámetros adicionales utilizados durante su definición. Este proceso de definición se detalla en la siguiente sección de este apéndice. La función “*pfun*” que implementa la $p(C)$ definida en la ecuación A7-1.4 con los parámetros de las expresiones A7-1.5 es “*pC.m*”. Las funciones “*qfun*” son la distribución uniforme en el intervalo $(10,90)$ y la gaussiana $\mathcal{N}_C(43.5,15)$. Son implementadas respectivamente en “*quniform.m*” y “*qgaussian.m*”. Las cadenas de Markov implementadas son la del método de Metropolis “*ChainMetropolis.m*” y la del muestreador independiente

²El cálculo de los pesos de los métodos de muestreo por rechazo y mediante cadenas de Markov se obvian habitualmente al ser todos ellos iguales a $1/N$.

“*ChainIndependentSampler.m*”, configurables a su vez respectivamente a partir de la definición de $p(C)$ y una función $q(C|c^{(i-1)})$ simétrica (en los experimentos una gaussiana centrada en la muestra anterior con varianza igual a 1, 5 o 10), y de $p(C)$ y $q(C)$ (en los experimentos la distribución uniforme en el intervalo (10,90) o la gaussiana $\mathcal{N}_c(43.5,15)$).

En el programa principal “*MainSingleAuxilarSamplers.m*” se definen la función “*pfun*”, la función “*qfun*”, y el número de muestras generadas N ; y se llaman a las diferentes funciones que implementan los métodos de muestro auxiliar de forma que además de generar los conjuntos de muestras y sus pesos, se realicen las representaciones gráficas correspondientes a las figuras A7-1.6, A7-1.7, A7-1.8, y A7-1.9. Los valores por defecto del número de muestras y de las funciones auxiliares $q(C)$ se pueden modificar en el programa principal “*MainSingleAuxilarSamplers.m*”.

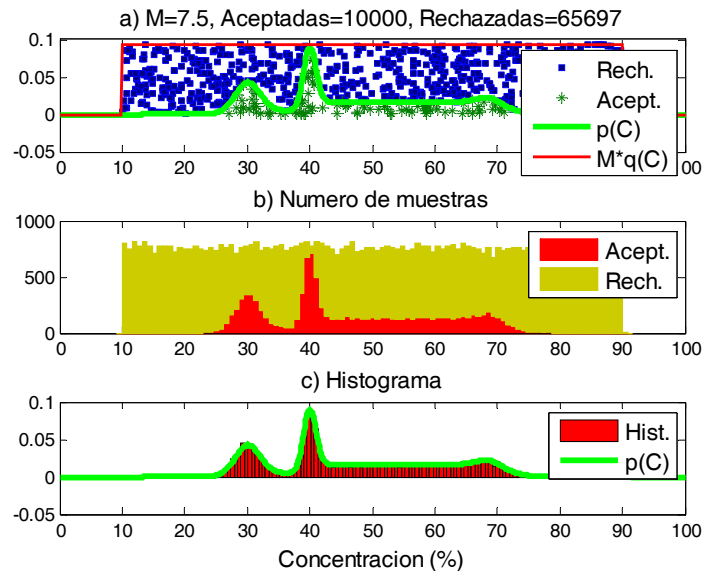


Figura A7-1.6 Gráfica del método de muestreo por rechazo.

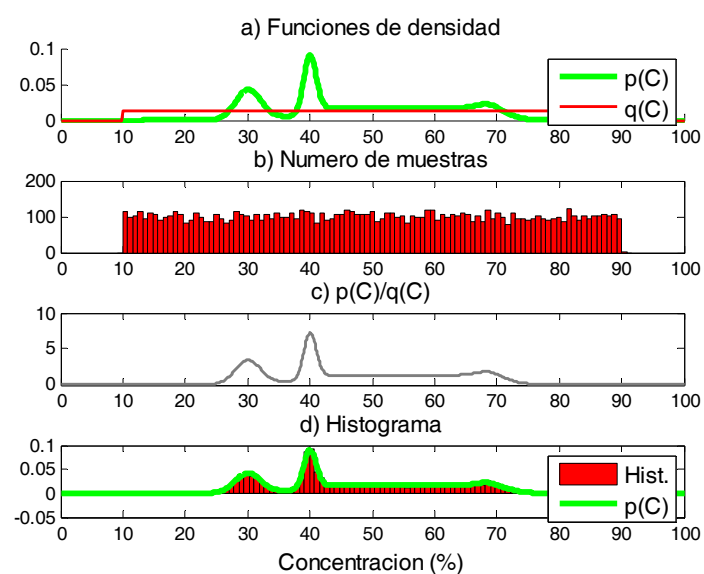


Figura A7-1.7 Gráfica del método de muestreo enfatizado

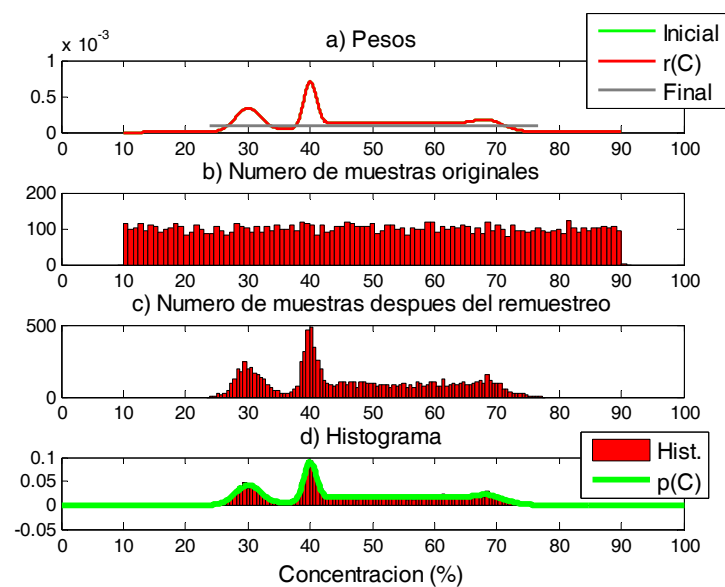


Figura A7-1.8 Gráfica del método de remuestreo por pesos.

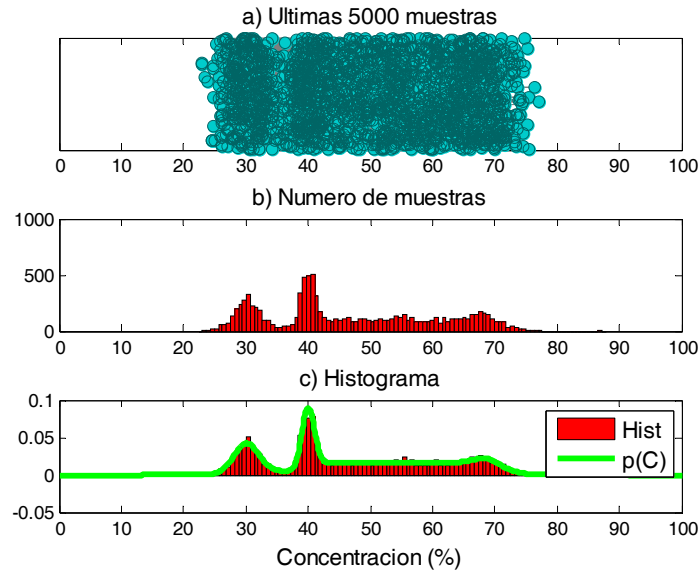


Figura A7-1.9 Gráfica del método de muestreo mediante una cadena de Markov.

El programa principal “*MainLoopAuxiliarSamplers.m*” genera conjuntos de muestras con sus correspondientes pesos, con los diferentes métodos de muestreo, y calcula el valor aproximado de la integral $I_N = \sum_{i=1}^N w(c^{(i)})c^{(i)}$. Realiza dicha operación sobre los diferentes métodos de muestreo auxiliar para múltiples conjuntos (50 por defecto) de múltiples números de muestras ($N = \{1e2, 2e2, 5e2, 1e3, 2e3, 5e3, 1e4, 2e4, 5e4, 1e5, 2e5, 5e5, 1e6\}$ por defecto), salva en múltiples ficheros los valores de I_N generados en los diferentes casos, y representa el valor promedio de I_N obtenido para cada uno de los métodos sobre todos los conjuntos para los diferentes valores de N . Los resultados para diferentes casos se representan en las figuras A7-1.10, A7-1.11, A7-1.12, y A7-1.13. Los valores por defecto pueden ser modificados en el programa principal “*MainLoopAuxiliarSamplers.m*”. La generación de las muestras, el cálculo integral y el almacenamiento se realizan con la funciones “*LoopRS.m*” (para el muestreo por rechazo), “*LoopIS.m*” (para el muestreo enfatizado), “*LoopWR.m*” (para el remuestreo por pesos), “*LoopMCMCMetropolis.m*” (para el muestreo mediante la cadena de Markov propuesta por Metropolis), y “*LoopMCMCIndependentSampling.m*” (para el muestreo mediante la cadena de Markov del muestreador independiente). La representación gráfica se realiza con la

función “*DrawAnalysis.m*”. Los ficheros generados son “*LoopRS0Data.mat*” (para el muestreo con rechazo con $q(C)$ uniforme), “*LoopRS1Data.mat*” (para el muestreo por rechazo con $q(C)$ gaussiana), “*LoopIS0Data.mat*” (para el muestreo enfatizado con $q(C)$ uniforme), “*LoopIS1Data.mat*” (para el muestreo enfatizado con $q(C)$ gaussiana), “*LoopWR1Data.mat*”, (para el remuestreo por pesos), “*LoopMCMCMetropolisData.mat*” (para el muestreo mediante la cadena de Markov de Metropolis con $q(C|c^{(i-1)})$ gaussiana y tres varianzas diferentes) y “*LoopMCMCIndependentSamplingData.mat*” (para el muestreo mediante la cadena de Markov del muestreador independiente con la $q(C)$ uniforme y gaussiana).

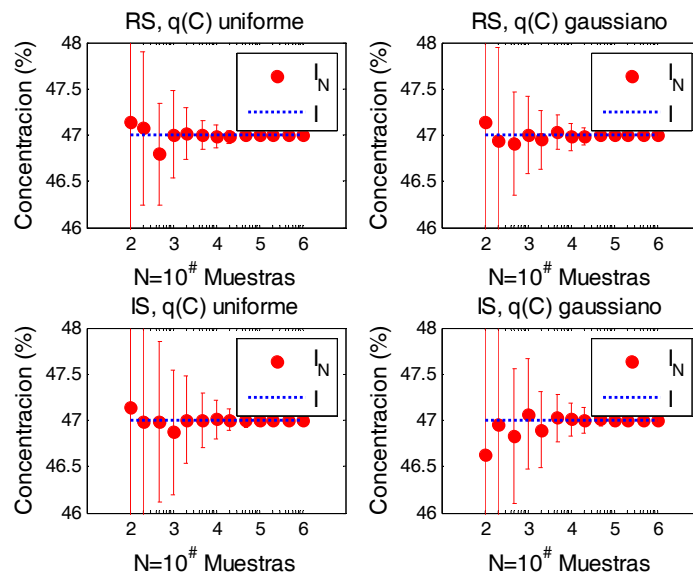


Figura A7-1.10 Gráfica del análisis del muestreo por rechazo y enfatizado.

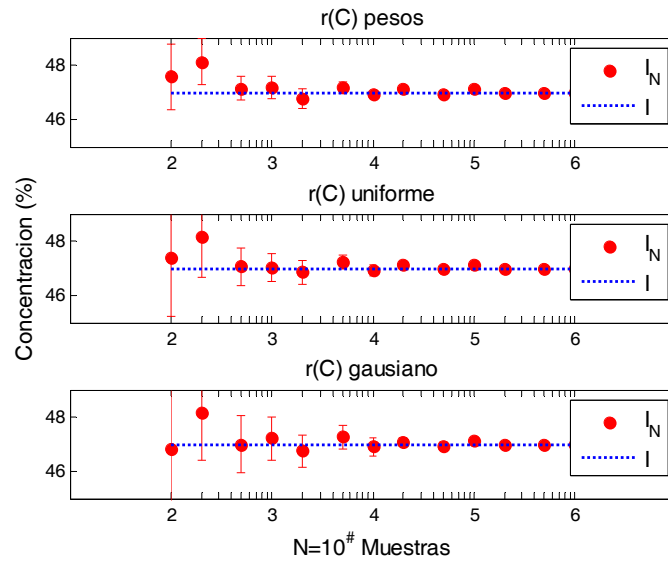


Figura A7-1.11 Gráfica del análisis del remuestreo por pesos.

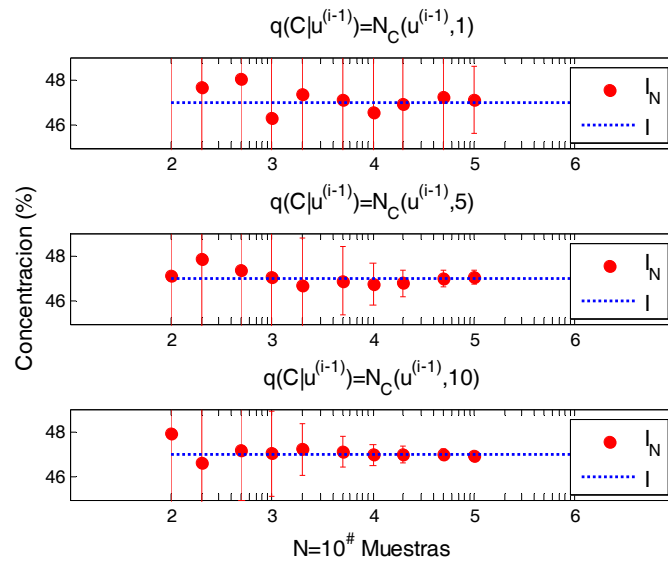


Figura A7-1.12 Gráfica del análisis del muestreo mediante cadenas de Markov con el método de Metropolis y tres gaussianas centradas en la muestra anterior con diferentes varianzas.

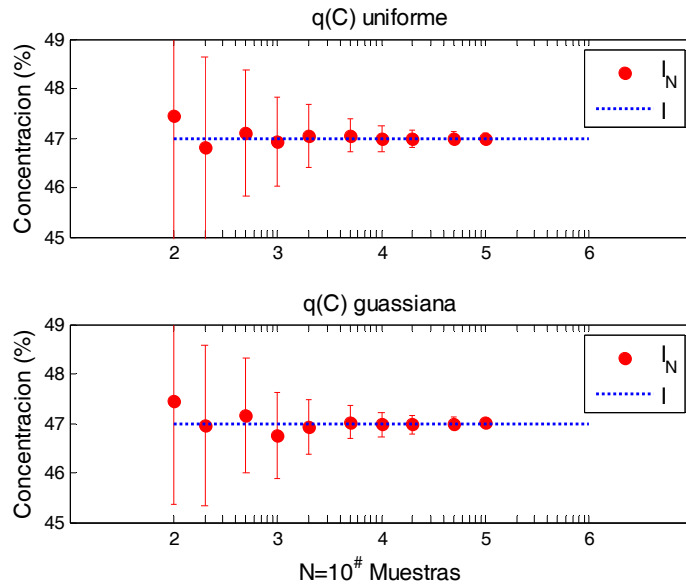


Figura A7-1.13 Gráfica del análisis del muestreo mediante cadenas de Markov con el muestreador independiente y la $q(C)$ uniforme y gaussiana.

A7-1.6 DEFINICION GENERICA DE LAS FUNCIONES

La función densidad original $p(U)$, la función densidad de muestreo auxiliar $q(U)$ de los métodos de muestreo por rechazo y muestreo enfatizado, la función que define la cadena de Markov $K(U|u^{(i-1)})$, y las funciones $q(U|u^{(i-1)})$ y $q(U)$ que respectivamente configuran el comportamiento de las cadenas de Metropolis y del muestreo independiente tienen que ser definidas antes de utilizar los métodos que las usan.

Con el objeto de lograr una implementación general³ de las técnicas, independiente del problema y fácilmente configurable, la definición de dichas funciones se realiza utilizando la función “*DefineFunction.m*” que genera una

³ Los métodos se han implementado únicamente para problemas unidimensionales ya que en ese caso la representación gráfica es mucho más sencilla. Su generalización es sencilla, utilizando matrices en vez de vectores para almacenar los valores de las muestras.

estructura en la que se almacena el nombre de la función utilizada y los parámetros adicionales necesarios para configurar su comportamiento. Esta información es utilizada por todos métodos cada vez que es necesario llamar a cada una de las funciones así definidas. Esta forma de proceder requiere que las diferentes funciones mantengan las cabeceras estándar que se presentan a continuación.

A7-1.6.1 Definición estándar de $p(U)$ y $q(U)$

La cabecera y el comportamiento de las funciones $p(U)$ (utilizada en el método de muestreo por rechazo, de muestreo enfatizado y de muestreo mediante una cadena de Markov) y $q(U)$ (utilizada en el método de muestreo por rechazo y muestreo enfatizado) tienen que ser los siguientes:

```
function val = fun(dosample,var[,parametrosadicionales])
% p(U) cabecera habitual
%
% u = fun(1, N[,parametrosadicionales])
%   Cuando el primer parametro de entrada toma el valor 1,
%   el segundo parámetro de entrada es el número de
%   muestras que se desea generar, y el parámetro de salida
%   es el valor de las muestras.
% p = fun(0, u[,parametrosadicionales])
%   Cuando el primer parametro de entrada toma el valor 0,
%   el segundo es el valor de las muestras u, y la función
%   evalúa el valor de la función sobre las muestras y
%   devuelve el resultado.
%
%   Los parametros adicionales son cualquier conjunto de
%   parametros que el usuario desee definir para configurar
%   el comportamiento de la funcion definida.
```

La definición de dichas funciones se realiza en los programas utilizando la siguiente orden. Los parámetros adicionales son aquellos parámetros definidos por el usuario que aparecen en la cabecera de la función después de los dos parámetros obligatorios anteriormente expuestos.

```
pfun = DefineFunction('fun'[,parametrosadicionales])
```

Funciones de este tipo son “*pC.m*”, “*quniform.m*” y “*qgaussian.m*”. La primera es la que implementa⁴ la $p(C)$, las dos restantes son las que implementan las dos $q(C)$ elegidas.

A7-1.6.2 Definición estándar de $K(U|u^{(i-1)})$

La cabecera y el comportamiento de la función $K(U|u^{(i-1)})$ (utilizada en el método de muestreo mediante una cadena de Markov) tienen que ser los siguientes:

```
function u = fun(uold[,parametrosadicionales])
% K(U|u^(i-1)) cabecera habitual
%
% u = fun(uold[,parametrosadicionales])
%   Genera el valor de la nueva muestra u en función de la
%   muestra anterior uold y los parámetros adicionales
%   utilizados para configurar el comportamiento de la
%   función
```

La definición de dichas funciones se realiza en los programas utilizando la siguiente orden. Los parámetros adicionales son aquellos parámetros definidos por el usuario que aparecen en la cabecera de la función después del parámetro obligatorio anteriormente expuesto.

```
chainfun = DefineFunction('fun',[,parametrosadicionales])
```

Funciones de este tipo son las funciones “*ChainMetropolis.m*” y “*ChainIndependentSampler.m*”, que implementan respectivamente las cadenas de Metropolis y del muestreador independiente. Como el comportamiento de dichos métodos se configura a partir de las funciones $p(U)$ y $q(U|u^{(i-1)})$ o $q(U)$, ambas funciones requieren como parámetros adicionales las dos estructuras, obtenidas a partir de la orden “*DefineFunction.m*”, que permiten definir ambas funciones. La definición de $p(U)$ y $q(U)$ es la misma que la que se presenta en la sección anterior. La definición de la $q(U|u^{(i-1)})$ se presenta a continuación. En todos los

⁴ Como no queremos generar muestras con esta función, sino únicamente evaluarlas, el comportamiento asociado al caso de generación ha sido sustituido por un mensaje de error que pone de manifiesto que la generación de las muestras no ha sido implementada.

casos se pueden añadir tantos parámetros adicionales como se consideren oportunos.

La cabecera y el comportamiento de la función $q(U|u^{(i-1)})$ (utilizada en la cadena del método de Metropolis) tienen que ser los siguientes:

```
function val = fun(dosample,var,uold[,parametrosadicionales])
% q(U|u^(i-1)) cabecera habitual
%
% u = fun(1, N,uold[,parmetrosadicionales])
% Cuando el primer parametro de entrada toma el valor 1,
% el segundo parámetro de entrada es el número de
% muestras que se desea generar, el tercero el valor de
% la muestra en el instante anterior, y el parámetro de
% salida es el valor de las nuevas muestras.
% p = fun(0, u,uold[,parmetrosadicionales])
% Cuando el primer parametro de entrada toma el valor 0,
% el segundo es el valor de las muestras u, el tercero el
% valor de las muestras en el instante anterior, y la
% función evalúa el valor de la función sobre las
% muestras y devuelve el resultado.
%
% Los parametros adicionales son cualquier conjunto de
% parametros que el usuario desee definir para configurar
% el comportamiento de la funcion definida.
```

La definición de dichas funciones se realiza en los programas utilizando la siguiente orden. Los parámetros adicionales son aquellos parámetros definidos por el usuario que aparecen en la cabecera de la función después de los tres parámetros obligatorios anteriormente expuestos.

```
qMCMCMetrofun = DefineFunction('fun'[,parametrosadicionales])
```

Es importante destacar la semejanza de la declaración de la función $q(U|u^{(i-1)})$ con la declaración de $q(U)$, donde la única diferencia se debe a la existencia del campo adicional asociado a $u^{(i-1)}$. De esta forma es posible utilizar como $q(U|u^{(i-1)})$ las funciones $q(U)$ cuyo primer parametro adicional pueda ser utilizado para pasar $u^{(i-1)}$. Este hecho se da en la función “*qgaussian.m*”, motivo por el que dicha función es utilizada tanto como $q(U)$ como como $q(U|u^{(i-1)})$.

A7-1.7 LISTADO DE FUNCIONES Y PROGRAMAS

Finalmente se incluye el listado de funciones y programas suministrados, con una breve descripción de su utilidad. Los programas principales han sido resaltados en negrita. En el directorio “*Data*” se almacena una copia de seguridad con los ficheros de datos utilizados en los capítulos 7 y 8. En el directorio “*private*” se encuentra un fichero de una versión de Matlab reciente, que no aparece en versiones más antiguas.

<i>ChainIndependentSampler.m</i>	<i>Cadena de Markov del muestreador independiente, configurable mediante la definición de $p(C)$ y $q(C)$</i>
<i>ChainMetropolis.m</i>	<i>Cadena de Markov del método de Metropolis configurable mediante la definición de $p(C)$ y $q(C c^{(i-1)})$</i>
<i>Data</i>	<i>Directorio donde se almacenan los ficheros de datos con los que se han realizado los análisis de los métodos recogidos en los capítulos 7 y 8</i>
<i>Data.m</i>	<i>Fichero donde se fijan los valores de los parámetros del problema (expresión A7-1.5)</i>
<i>DefineFunction.m</i>	<i>Función para definir el nombre y parámetros adicionales de las funciones elegidas</i>
<i>DrawAnalysis.m</i>	<i>Función que realiza las representaciones gráficas asociadas a los valores del cálculo integral</i>
<i>HistogramCount.m</i>	<i>Función que representa gráficamente, mediante un histograma, el número de muestras en diferentes regiones del espacio</i>
<i>HistogramWeighted.m</i>	<i>Función que representa gráficamente, mediante un histograma, la distribución asociada a un conjunto de muestras y sus correspondientes pesos</i>
<i>ImportanceSampling.m</i>	<i>Método de muestreo enfatizado</i>
<i>LoopCMean.m</i>	<i>Función que realiza el cálculo integral en función de $\bar{c}(T)$</i>
<i>LoopIS.m</i>	<i>Función que realiza el cálculo integral mediante el método de muestreo enfatizado</i>
<i>LoopMCMCIndependentSampler.m</i>	<i>Función que realiza el cálculo integral mediante el muestreador independiente</i>

<i>LoopMCMCMetropolis.m</i>	<i>Función que realiza el cálculo integral mediante el método de Metropolis</i>
<i>LoopRS.m</i>	<i>Función que realiza el cálculo integral mediante el método de muestreo por rechazo</i>
<i>LoopWR.m</i>	<i>Función que realiza el cálculo integral usando el método de remuestreo por pesos</i>
MainCMean.m	<i>Programa principal que analiza los resultados del calculo aproximado de la integral a partir de $\bar{c}(T)$</i>
MainExact.m	<i>Programa principal que calcula el valor exacto de la integral</i>
MainLoopAuxiliarSamplers.m	<i>Programa principal que realiza el análisis de los resultados del cálculo integral con los diferentes métodos de muestreo auxiliar</i>
MainProblem.m	<i>Programa principal que representa las funciones $p(C,T)$, $c(T)$, $\bar{c}(T)$ y $p(C)$.</i>
MainSingleAuxiliarSamplers.m	<i>Programa principal que ejecuta cada uno de los métodos de muestreo auxiliar y representa las gráficas asociadas a cada uno de los mismos</i>
<i>MarkovChain.m</i>	<i>Función que implementa el método de muestreo mediante cadenas de Markov</i>
<i>private</i>	<i>Directorio donde se almacena la función de Matlab randsample</i>
<i>RejectionSampling.m</i>	<i>Función que implementa el método de muestreo por rechazo</i>
<i>WeightedResampling.m</i>	<i>Función que implementa el método de remuestreo por pesos</i>
<i>gsamp.m</i>	<i>Función que genera muestras de acuerdo con una distribución gaussiana</i>
<i>qgaussian.m</i>	<i>Función $q(C)$ y $q(C c^{(i-1)})$ gaussiana</i>
<i>quniform.m</i>	<i>Función $q(C)$ uniforme</i>
<i>pC.m</i>	<i>Función $p(C)$ del problema</i>