TIP ejercicio 1
curso 2012/2013

Carlos Gil Soriano
UAM-HPCN
gilsoriano@gmail.com

20 de febrero de 2013

Resumen

En esta primera entrega de ejercicios se tratan tres problemas que se centran en:

- Comparación del cálculo de momentos de orden n mediante *cuadraturas* con su correspondiente por *Montecarlo*.
- Propiedades del *Browniano aritmético*.
- Propiedades del *Browniano geométrico*.

Historial		
Versión	**Estado**	**Fecha**
0.1	Problema 1 resuelto	18 de febrero de 2013
0.1	Problema 2 resuelto	19 de febrero de 2013
0.1	Problema 3 resuelto	20 de febrero de 2013

Índice

Índice de cuadros

Índice de figuras

Listings

1. Cálculo de momentos centrales

Se define un momento central de orden n como:

$$\mu_n = E[(X - E[X])^n] = \int_{-\infty}^{\infty} (x-\mu)^n pdf(x)\mathrm{d}x$$

Para realizar la cuadratura debemos identificar la función $g(x)$ a integrar:

$$I = \int_{-\infty}^{\infty} g(x)\mathrm{d}x$$

En nuestro caso se trata de:

$$g(x) = (x-\mu)^n pdf(x)$$

Esta función g(x) se ha definido en Matlab en la función *quadfunction.m*. La integración por cuadratura de Lobatto (cuadratura gaussiana con peso constante de 1) se realiza en Matlab por medio del comando *quadl*. El código del cálculo de $N(\mu, \sigma)$ se encuentra en *centralMomentGaussian.m*.

1.1. Salidad del programa y evaluación

```
>> mu = 0; sigma = 1;
   N = 8;
   for n = 1:N;
   moment(n) = centralMomentGaussian(mu,sigma,n);
   end
   moment

moment =
4.0305e-22

moment =
1.0000

moment =
4.0305e-20

moment =
3.0000

moment =
4.0305e-18

moment =
15.0000

moment =
-4.1039e-17

moment =
105.0000

moment =
0.000    1.000    0.000    3.000    0.000    15.000    -0.000    105.000
```

Se observa que el primer y segundo momento ofrecen los valores esperados de cero y varianza, respectivamente. Además, dada la simetría de la función a integrar, g(x), y la simetría del dominio de integración, se comprueba que los momentos de orden impar se anulan.

1.2. Comparación con estimación por Montecarlo

La simulación por el método de Montecarlo ofrece unos resultados similares:

- El histograma de los momentos de orden impar se concentra sobre cero.
- El histograma de momentos de orden 2, se concentra sobre la varianza.

2. Propiedades browniano aritmético

El **browniano aritmético**, también llamado proceso de Wiener con deriva μ y varianza σ^2, es de la forma:

$$dS_t = \mu dt + \sigma dW_t \tag{\textbf{ABM}}$$

Se muestra también el **browniano geométrico**, para ver las diferencias existentes en la formación de la ecuación diferencial estocástica:

$$\frac{dS_t}{S_t} = \mu dt + \sigma dW_t \tag{\textbf{GBM}}$$

2.1. Distribución del browniano aritmético

El browniano aritmético se caracteriza por:

$$E[S(t)] = S_0 + \mu t \tag{Media}$$

$$E[S(t)] = \sigma^2 t \tag{Varianza}$$

Por lo tanto, en la grfica debemos observar una tendencia ascendente junto con una desviación parabólica de las diferentes trayectorias del browniano:

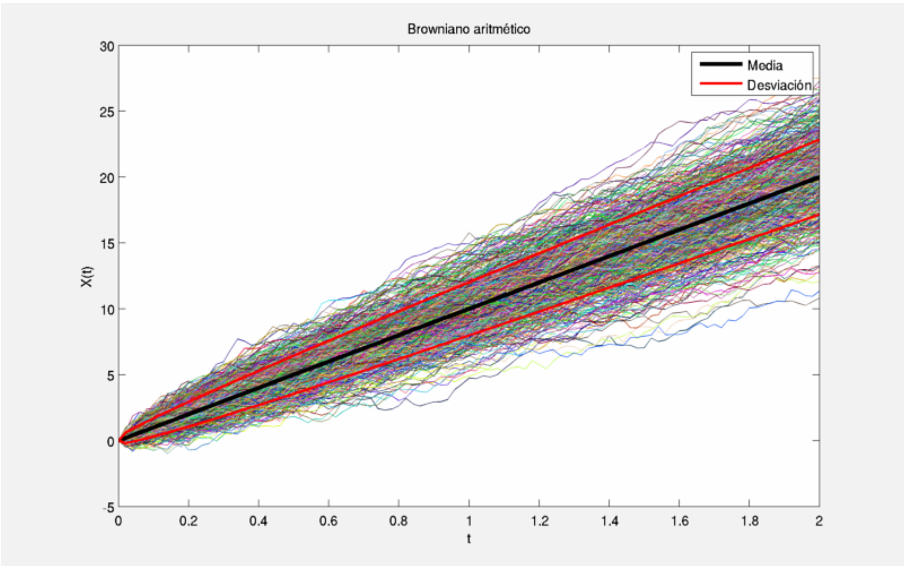

Figura 1: Browniano aritmético

2.2. Diagrama de autocovarianzas

Por la definición de autocovarianza:

$$\mathrm{C}_{SS}(t,s) = \mathrm{E}[(S(t) - \mathrm{E}[S(t)])(S(s) - \mathrm{E}[S(s)])]$$

donde si $t = s$ la autocovarianza la varianza, que ya conocemos a priori por los resultados de la sección anterior y servirá para comprobar si los resultados tienen (algo) de sentido.

3. Valoración de productos derivados

A. centralMomentGaussian.m

```
function moment = centralMomentGaussian(mu,sigma,n)
%centralMomentGaussian: central moment for Gaussian distribution
%
%SYNTAX: moment = centralMomentGaussian(mu,sigma,n)
%
%INPUT:   mu    : Average of the distribution
%         sigma : Standard deviation
%         n     : Order of the moment
%
%STEPS
%   1.- We create the quadrature function to be integrated
%   2.- We define bounds for integration
%   3.- Lastly, we quadrate the function via quadl
%CHECKS
%   A.- The first moment is always zero
%   B.- The second moment equals the variance
%
%EXAMPLE:
%   mu = 0; sigma = 1;
%   N = 8;
%   for n = 1:N;
%       moment(n) = centralMomentGaussian(mu,sigma,n);
%   end
%   moment

    TOL        = 1e-6; %Automatically set precision target

    R          = 10;
    lowerBound = mu - R*sigma;
    upperBound = mu + R*sigma;
    x          = -100:0.1:100;

    moment     = quadl(@(x)quadfunction(x, mu, sigma, n), lowerBound,
        upperBound, TOL)
    plot(x, moment);
```

Listing 1: centralMomentGaussian.m

A.1. quadfunction.m

```
function y = quadfunction(x, mu, sigma, n)
    y = (x - mu).^n.*normpdf(x, mu, sigma);
```

Listing 2: quadfunction.m

B. other