

# *Técnicas Monte Carlo aplicadas a la resolución de Ecuaciones en Derivadas Parciales*

**Armando Serrano Lombillo**  
UNIVERSIDAD POLITÉCNICA DE VALENCIA

---

## **Abstract**

*En este trabajo se aplican las técnicas Monte Carlo a la resolución de ecuaciones en derivadas parciales, en concreto a la ecuación del calor. Si bien estas técnicas no mejoran las técnicas clásicas proporcionan una manera alternativa y más intuitiva de entender los métodos tradicionales. La última parte del trabajo se dedica al estudio estadístico de la importancia de cuatro factores en la precisión del método Monte Carlo. Para realizar el estudio se utilizó STATGRAPHICS y MATHEMATICA.*

*In this article Monte Carlo's method is applied in order to solve the Heat Equation. This method provides an alternative and more intuitive possibility to understand the classical techniques. In the last part of the work we study the relevance in the precision of Monte Carlo's method of four statistic variables. The study has been done bu using STATGRAPHICS and MATHEMATICA.*

**Keywords:** MATHEMATICA, STATGRAPHICS, *Monte Carlo*

---

## 1 Introducción

Este trabajo trata de aplicar las técnicas Monte Carlo a la resolución de ecuaciones en derivadas parciales, en concreto a la ecuación del calor.

Las técnicas Monte Carlo (MC) están generalmente asociadas a la simulación de procesos caóticos o estocásticos, sin embargo, también se pueden utilizar para hallar aproximaciones numéricas de problemas determinísticos. Por ejemplo, se pueden utilizar técnicas MC para hallar aproximaciones a integrales que no se pueden resolver analíticamente. En el caso de las ecuaciones en derivadas parciales estudiadas en este trabajo, las técnicas MC no han aportado ventajas en el sentido de que hayan resuelto problemas que otros métodos no podían, ni tampoco han sido más eficientes. Sin embargo, sí que han aportado algo positivo ya que dan una manera alternativa y más intuitiva de entender los métodos tradicionales. Además, no hay que descartar que con ecuaciones en derivadas parciales más complejas el método MC pueda ser ventajoso.

Dice R. Y. Rubinstein en su libro “Simulation and the Monte Carlo Method” [2] que *“un problema determinístico puede ser resuelto por el método Monte Carlo si tiene la misma expresión formal que algún proceso estocástico”*. Es este el fundamento de los algoritmos que se desarrollan aquí. Como se verá más adelante, en el método explícito para resolver la ecuación del calor se obtiene el valor de la función en un punto como la media ponderada de los valores de la función en puntos vecinos. En el método MC, esos coeficientes de ponderación se convierten en probabilidades, de tal manera que adopta la misma expresión formal que un proceso estocástico.

Para acabar el trabajo, se dedica un apartado al estudio estadístico de la importancia de cuatro factores en la precisión del método MC. Para realizar el estudio se utilizó el STATGRAPHICS. En el resto del trabajo, los algoritmos están implementados en MATHEMATICA. El texto está editado con el sistema de código abierto L<sup>A</sup>T<sub>E</sub>X.

## 2 Estudio de la ecuación del calor

### 2.1 Significado físico

La ecuación del calor se llama así porque, entre otras cosas, se puede utilizar para modelar el flujo de calor (variación de la temperatura) en un cuerpo. Para el caso de un sólido unidimensional la ecuación es la siguiente:

$$cd \frac{\partial u}{\partial t} - \frac{\partial}{\partial x} \left( k \frac{\partial u}{\partial x} \right) = F(x, t)$$

donde  $x$  es la coordenada  $x$  de la barra,  $t$  es el tiempo,  $u$  es la temperatura,  $d$  es la densidad del material,  $c$  es el calor específico,  $k$  es el coeficiente de conductividad y  $F(x, t)$  es una función que representa la acción exterior. Si  $k$  no varía con  $x$ , se puede sacar fuera de la derivada. Si además se dividen todos los miembros de la ecuación por  $cd$ , queda:

$$\frac{\partial u}{\partial t} - \frac{k}{cd} \cdot \frac{\partial^2 u}{\partial x^2} = \frac{F(x, t)}{cd}$$

Si ahora se llama  $a$  a  $\frac{k}{cd}$  y  $f(x, t)$  a  $\frac{F(x, t)}{cd}$  queda:

$$\frac{\partial u}{\partial t} - a \frac{\partial^2 u}{\partial x^2} = f(x, t)$$

Cuando  $a$  valga uno, la ecuación será todavía más simple:

$$\frac{\partial u}{\partial t} - \frac{\partial^2 u}{\partial x^2} = f(x, t)$$

que es la ecuación que se utilizará de ahora en adelante, ya que siempre se podrá llegar a esta forma mediante un cambio de variables adecuado (ver [3]).

## 2.2 Resolución por diferencias finitas: el método explícito

Si se discretiza el espacio  $(x, t)$  y se sustituye  $\frac{\partial u}{\partial t}$  y  $\frac{\partial^2 u}{\partial x^2}$  por las aproximaciones obtenidas en el Apéndice 4 (se utilizará la fórmula de la diferencia adelantada para la primera derivada), se tiene (con la notación habitual de los métodos de diferencias finitas):

$$\frac{u_{i,j+1} - u_{i,j}}{\Delta t} - \frac{u_{i+1,j} + u_{i-1,j} - 2 \cdot u_{i,j}}{\Delta x^2} = f_{i,j}$$

donde el primer subíndice indica espacio y el segundo tiempo. Despejando  $u_{i,j+1}$  se llega a la fórmula:

$$u_{i,j+1} = R \cdot u_{i-1,j} + (1 - 2R) \cdot u_{i,j} + R \cdot u_{i+1,j} + \Delta t \cdot f_{i,j} \quad (7.1)$$

$$\text{donde } R = \frac{\Delta t}{\Delta x^2}.$$

Por lo tanto, cada  $u_{i,j+1}$  (lease temperatura en un punto  $i$  en el instante  $j+1$ ) se calcula a partir de la temperatura en el instante anterior en el mismo punto ( $u_{i,j}$ ) y en los puntos contiguos ( $u_{i-1,j}$  y  $u_{i+1,j}$ ) y de la influencia externa en el punto en el instante anterior ( $f_{i,j}$ ).

Como este método calcula las temperaturas de un instante a partir de las del anterior, se necesita conocer la temperatura inicial en toda la barra ( $u(x, 0)$ ). Además, no se pueden calcular las temperaturas en la frontera (los extremos de la barra) porque se necesitarían saber temperaturas fuera de la barra (que no están definidas). Por lo tanto, también es necesario conocer la evolución de la temperatura en los dos extremos ( $u(0, t)$  y  $u(L, t)$ ). A  $u(x, 0)$  se le llama *condición inicial* y a  $u(0, t)$  y  $u(L, t)$  *condiciones de contorno*. Las condiciones de contorno también pueden venir dadas por las derivadas de  $u$  ( $u_x(0, t)$  y  $u_x(L, t)$ ) si se hacen unos pequeños cambios en el algoritmo.

Con respecto al parámetro  $R$ , se puede demostrar que debe ser menor o igual que 0'5 para que el método sea estable y por lo tanto convergente (ver [3]).

A continuación, un ejemplo muy simple que ilustra el funcionamiento del método.

## 2.3 Ejemplo del método explícito

Se trata de encontrar  $u(x, 0'1)$  en el siguiente problema:

$$\left\{ \begin{array}{l} \frac{\partial u}{\partial t} - \frac{\partial^2 u}{\partial x^2} = 0 \\ u(0, t) = 0 \\ u(1, t) = 0 \\ u(x, 0) = \begin{cases} x & \text{si } 0 \leq x \leq 0'5 \\ 1 - x & \text{si } 0'5 \leq x \leq 1 \end{cases} \\ 0 \leq x \leq 1 \end{array} \right.$$

Se tomarán 10 intervalos en  $x$  y 30 en  $t$ . De esta manera  $R < 0'5$  y el problema será estable. Se incluye el algoritmo comentado y en dos partes. 'ECUACIÓN DEL CALOR - PRELIMINARES' es la introducción de datos y en un mismo problema es igual para cualquier algoritmo que se utilice (versión tradicional o MC). 'MÉTODO EXPLÍCITO - VERSIÓN TRADICIONAL' es la resolución del problema por el método explícito tradicional y utiliza los datos introducidos anteriormente.

## ECUACIÓN DEL CALOR - PRELIMINARES

Discretización del problema continuo:

*In*[1] :=  $L = 1$ ;

$N_{\text{IntervalosL}} = 10$ ;

$\text{deltaL} = \frac{L}{N_{\text{IntervalosL}}}$ ;

$T = 0.1$ ;

$\text{deltaT} = \frac{1}{300}$ ;

*In*[2] :=  $N_{\text{IntervalosT}} = \text{Ceiling}\left[\frac{T}{\text{deltaT}}\right]$

*Out*[2] = 30

*In*[3] :=  $R = \frac{\text{deltaT}}{\text{deltaL}^2}$

*Out*[3] =  $\frac{1}{3}$

Introducción de las condiciones iniciales y de contorno:  $T_{\text{Inicial}}$ ,  $T_0$  y  $T_L$

*In*[4] :=  $T_{\text{Inicial}} =$

$\text{Table}[N[\text{If}[i < N_{\text{IntervalosL}}/2, i/N_{\text{IntervalosL}},$   
 $1 - i/N_{\text{IntervalosL}}]], \{i, 0, N_{\text{IntervalosL}}\}]$

*Out*[4] = {0., 0.1, 0.2, 0.3, 0.4, 0.5, 0.4, 0.3, 0.2, 0.1, 0.}

*In*[5] :=  $T_0 = \text{Table}[0, \{i, 0, N_{\text{IntervalosT}}\}]$

*Out*[5] = {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,  
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0}

*In*[6] :=  $T_L = T_0$

*Out*[6] = {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,  
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0}

## MÉTODO EXPLÍCITO - VERSIÓN TRADICIONAL

Este método sólo es estable para  $0 < R \leq 0.5$ .

Creación de la matriz de datos:

```
In[7]:= TemperaturasEx = Table[Table[0, {i, 0, NIntervalosL}],
    {i, 0, NIntervalosT};
```

Introducción de TInicial, T0 y TL en la matriz de datos:

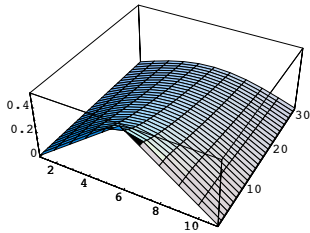
```
In[8]:= TemperaturasEx = ReplacePart[TemperaturasEx, TInicial, 1];
```

```
In[9]:= For[i = 1, i ≤ NIntervalosT + 1, i ++, TemperaturasEx[[i, 1]] = T0[[i]];
    TemperaturasEx[[i, NIntervalosL + 1]] = TL[[i]]]
```

Cálculo:

```
In[10]:= For[i = 2, i ≤ NIntervalosT + 1, i ++,
    For[j = 2, j < NIntervalosL + 1, j ++,
        TemperaturasEx[[i, j]] =
            R TemperaturasEx[[i - 1, j - 1]] +
            (1 - 2 R) TemperaturasEx[[i - 1, j]] +
            R TemperaturasEx[[i - 1, j + 1]]
    ]]
```

```
In[11]:= ListPlot3D[TemperaturasEx];
```



## 2.4 Versión Monte Carlo del método explícito

Para entender mejor como funciona el método MC, se puede diseñar un juego que sirva para aproximar la solución de la ecuación del calor en un punto basándose en la ecuación (7.1). Para mayor claridad, se considerará primero que  $f(x, t) = 0$  y más tarde se explicarán las modificaciones necesarias para tener en cuenta la influencia externa. Las instrucciones del juego son las siguientes:

- Se juega en un tablero de  $(NIntervalosL) \times (NIntervalosT)$ .
- La ficha empieza en la casilla (espacio, tiempo) correspondiente al punto donde queremos conocer la temperatura (parte superior del tablero).
- En cada turno, la ficha se mueve aleatoriamente a uno de los siguientes puntos y con una de las siguientes probabilidades:

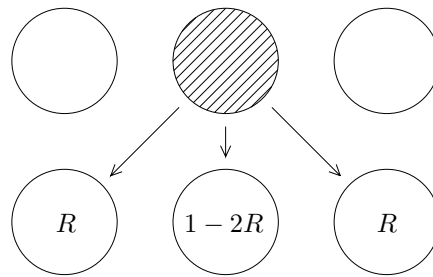


Figure 7.1: Movimientos y probabilidades.

- La jugada acaba cuando la ficha llega a uno de los lados del tablero (izquierdo, derecho o inferior) y entonces recibe una puntuación igual a la temperatura de ese punto. Las temperaturas en estos puntos son conocidas por ser condiciones de contorno o condiciones iniciales del problema.
- Se hacen tantas jugadas como se quieran (cuantas más jugadas, mayor precisión) y se lleva la cuenta de todas las puntuaciones obtenidas.
- Para saber la temperatura buscada, se hace la media de todas las puntuaciones obtenidas.

Es importante observar que como  $R$  tiene que ser menor o igual que 0,5 para que el método explícito sea convergente, no habrá ninguna probabilidad negativa. De no ser así, el problema no tendría la misma expresión formal que un proceso estocástico y no se podría aplicar el método MC. Esta forma de resolver el problema es una técnica conocida como “random walk” (ver por ejemplo [1]).

A continuación, el código de MATHEMATICA para el ejemplo resuelto en 2.3 pero utilizando ahora la versión MC (hay que haber ejecutado ‘ECUACIÓN DEL CALOR - PRELIMINARES’ antes):

## MÉTODO EXPLÍCITO - VERSIÓN MONTE CARLO

Aunque en el método anterior se ha mostrado la evolución de las temperaturas a lo largo del tiempo, normalmente sólo se quieren saber las temperaturas en un instante. En el método anterior había que calcular las temperaturas en los instantes anteriores inevitablemente, sin embargo, con la versión Monte Carlo no se está obligado a calcularlas, así que se obtendrán únicamente las temperaturas (para toda la barra) en el instante deseados.

Creación de la matriz de datos:

```
In[12]:= TemperaturasExMC = Table[Table[0, {i, 0, NIntervalosL}],
      {i, 0, NIntervalosT}];
```

Introducción de TInicial, T0 y TL en la matriz de datos:

```
In[13]:= TemperaturasExMC = ReplacePart[TemperaturasExMC, TInicial, 1];
```

```
In[14]:= For[i = 1, i ≤ NIntervalosT + 1, i ++,
      TemperaturasExMC[[i, 1]] = T0[[i]];
      TemperaturasExMC[[i, NIntervalosL + 1]] = TL[[i]]]
```

Cálculo:

```
In[15]:= NIteraciones = 500;
```

(\* Número de iteraciones por punto \*)

```
In[16]:= If[R > 0.5,
```

```
Print["No se puede ejecutar este método para para
      R > 0'5"],
```

```
For[j = 2, j ≤ NIntervalosL, j ++, (* Una vez por punto *)
```

```
ProbPunto = Table[0, {i, 0, NIntervalosT},
      {j, 0, NIntervalosL}];
```

```
x = j; (* Por claridad *)
```

```
For[i = 0, i < NIteraciones, i ++,
```

(\* Una vez por punto e iteración \*)

```
HeLlegado = False;
```

```
PuntoActual = {x, NIntervalosT + 1};
```

```
While[HeLlegado == False,
```

(\* Este es el bucle que mueve la ficha hasta que se llega a un borde \*)

```
Direccion = Random[];
```

```
If[Direccion < R, PuntoActual = PuntoActual + {-1, -1},
```

```
If[Direccion < 1 - R, PuntoActual = PuntoActual + {0, -1},
```

```
PuntoActual = PuntoActual + {1, -1}];
```

(\* Con esto se acaba de desplazar la ficha una casilla \*)

(\* A continuación comprueba si se ha llegado a un borde \*)

```
If[PuntoActual[[1]] == 1 || PuntoActual[[2]] == 1 ||
```

```
PuntoActual[[1]] == NIntervalosL + 1, HeLlegado = True, ]];
```

```
ProbPunto[[PuntoActual[[2]], PuntoActual[[1]]]] =
```

```
ProbPunto[[PuntoActual[[2]], PuntoActual[[1]]] + 1];
```

```
ProbPunto = ProbPunto/NIteraciones;
```

```
A = ProbPunto * TemperaturasExMC;
```

```
TemperaturasExMC[[NIntervalosT + 1, x]] =
```

```
Sum[A[[i + 1, j + 1]], {i, 0, NIntervalosT},
      {j, 0, NIntervalosL}]]];
```



```
In[17]:= TemperaturasExMC[[NIntervalosT + 1]]
Out[17]= {0, 0.0462, 0.0938, 0.1368, 0.149,
          0.1452, 0.15, 0.1194, 0.0832, 0.0494, 0}
```

En cuanto a la introducción de la influencia externa  $f(x, t)$ , habría que modificar ligeramente las reglas del juego. En cada turno, además de los tres movimientos ya descritos habría una cuarta posibilidad. Para que las probabilidades sigan siendo sobre 1 habría que dividir por  $(R) + (1 - 2R) + (R) + (\Delta T) = 1 + \Delta T$ , quedando  $R/(1 + \Delta T)$ ,  $(1 - 2R)/(1 + \Delta T)$ ,  $R/(1 + \Delta T)$  y  $1/(1 + \Delta T)$ . Cuando toque la influencia externa, habría que sumar la  $f$  del punto en el que está la ficha en ese momento.

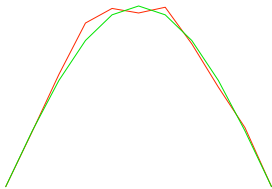
## 2.5 Comparativa

Para acabar con el método explícito, una pequeña comparativa entre la versión tradicional y el método MC:

### MÉTODO EXPLÍCITO - COMPARATIVA

Comparación gráfica de los dos métodos en el instante final:

```
In[18]:= Show[ListPlot[TemperaturasExMC[[NIntervalosT + 1]],
               PlotJoined -> True, Axes -> False,
               PlotStyle -> {RGBColor[1, 0, 0]}],
             ListPlot[TemperaturasEx[[NIntervalosT + 1]],
               PlotJoined -> True, Axes -> False,
               PlotStyle -> {RGBColor[0, 0.9, 0]}];
```



## 2.6 El método de Crank-Nicolson

A continuación se expone el segundo método para resolver la ecuación del calor: el método de Crank-Nicolson (CN).

En CN, se utiliza la misma aproximación para  $\frac{\partial u}{\partial t}$  pero se hace una media para  $\frac{\partial^2 u}{\partial x^2}$  de la siguiente manera:

$$\frac{\partial^2 u}{\partial x^2} = \frac{1}{2} \left( \frac{u_{i+1,j+1} + u_{i-1,j+1} - 2 \cdot u_{i,j+1}}{\Delta x^2} + \frac{u_{i+1,j} + u_{i-1,j} - 2 \cdot u_{i,j}}{\Delta x^2} \right)$$

Sustituyendo las aproximaciones en la ecuación del calor, se llega a:

$$(2 + 2R) \cdot u_{i,j+1} - R \cdot u_{i+1,j+1} - R \cdot u_{i-1,j+1} = R \cdot u_{i+1,j} + (2 - 2R) \cdot u_{i,j} + R \cdot u_{i-1,j} + 2 \cdot \Delta T \cdot f_{i,j}$$

Como se puede observar, este es un método implícito, ya que las temperaturas del tiempo  $j + 1$  dependen de otras temperaturas en  $j + 1$ . Cuando se resuelve la ecuación del calor por este método, esto lleva a que todas las temperaturas de un mismo tiempo se tengan que resolver a la vez mediante un SEL. Afortunadamente esto no es un problema para MC, ya que no es necesario conocer las temperaturas de los puntos vecinos.

El método de CN tiene una clara ventaja con respecto al método explícito explicado anteriormente: es estable para cualquier  $R$ . Sin embargo, a la hora de implementar una versión MC hay que tener precaución. Si se despeja  $u_{i,j+1}$  de la fórmula anterior (sin tener en cuenta la influencia externa):

$$u_{i,j+1} = \frac{R}{(2 + 2R)} \cdot (u_{i+1,j+1} + u_{i-1,j+1} + u_{i+1,j} + u_{i-1,j}) + \frac{(2 - 2R)}{(2 + 2R)} \cdot u_{i,j}$$

se puede observar que el coeficiente  $\frac{(2-2R)}{(2+2R)}$  (que en MC se convierte en una probabilidad) puede tomar valores negativos. Esto no tiene sentido, por lo tanto, para que la versión MC funcione hay que exigir que  $\frac{(2-2R)}{(2+2R)} \geq 0$ , o lo que es lo mismo, que  $R \leq 1$ . Esto es una pequeña desventaja, ya que se pierde parte de la flexibilidad característica de CN.

Se incluye a continuación el código de MATHEMATICA para los métodos CN (tradicional y MC) con el mismo ejemplo que para el método explícito. El código de la versión tradicional se podría haber optimizado notablemente teniendo en cuenta que la matriz de coeficientes de cada SEL a resolver es tridiagonal y además es siempre la misma. No se he hecho por claridad.

## MÉTODO DE CRANK-NICOLSON - VERSIÓN TRADICIONAL

Creación de la matriz de datos:

```
In[19]:= TemperaturasCN =
  Table[Table[0, {i, 0, NIntervalosL}],
    {i, 0, NIntervalosT};
```

Introducción de TInicial, T0 y TL en la matriz de datos:

```
In[20]:= TemperaturasCN =
  ReplacePart[TemperaturasCN, TInicial, 1];
In[21]:= For[i = 1, i ≤ NIntervalosT + 1, i ++,
  TemperaturasCN[[i, 1]] = T0[[i]];
  TemperaturasCN[[i, NIntervalosL + 1]] =
  TL[[i]];]
```

Cálculo:

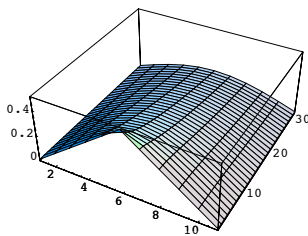
Hay que resolver un SEL en cada instante. De forma matricial:  $A \cdot x = B \cdot x_{ant} + C$ . Aunque  $x_{ant}$  cambia cada vez, A y B son constantes una vez se ha fijado R.

```
In[22]:= A = Table[Table[0, {i, 0, NIntervalosL}],
  {i, 0, NIntervalosL};

  B = Table[Table[0, {i, 0, NIntervalosL}],
  {i, 0, NIntervalosL};
In[23]:= A[[1, 1]] = 1;

  A[[NIntervalosL + 1, NIntervalosL + 1]] = 1;

  For[i = 2, i < NIntervalosL + 1, i ++,
  A[[i, i]] = 2 + 2R;
  B[[i, i]] = 2 - 2R;
  A[[i, i - 1]] = -R;
  A[[i, i + 1]] = -R;
  B[[i, i - 1]] = R;
  B[[i, i + 1]] = R;]
In[24]:= For[i = 2, i ≤ NIntervalosT + 1, i ++,
  TemperaturasCN[[i]] =
  Transpose[LinearSolve[A,
  B.Transpose[
  {TemperaturasCN[[i - 1]]}]]][[1]]]
In[25]:= ListPlot3D[TemperaturasCN];
```



## MÉTODO DE CRANK-NICOLSON - VERSIÓN MONTE CARLO

Creación del vector de datos:

```
In[26]:= TemperaturasCNMC = Table[Table[0, {i, 0, NIntervalosL}],
    {i, 0, NIntervalosT};
```

Introducción de TInicial, T0 y TL en la matriz de datos:

```
In[27]:= TemperaturasCNMC = ReplacePart[TemperaturasCNMC, TInicial, 1];

In[28]:= For[i = 1, i ≤ NIntervalosT + 1, i ++,
    TemperaturasCNMC[[i, 1]] = T0[[i]];
    TemperaturasCNMC[[i, NIntervalosL + 1]] = TL[[i]]]
```

Cálculo:

```
In[29]:= NIteraciones = 500 ;
    (* Número de iteraciones por punto *)

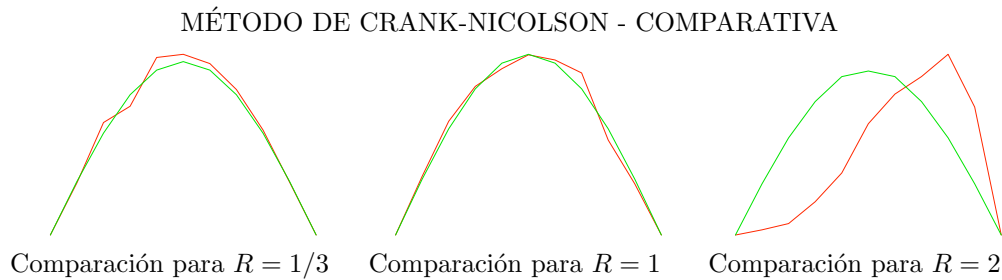
In[30]:= For[j = 2, j ≤ NIntervalosL, j ++,
    ProbPunto = Table[0, {i, 0, NIntervalosT},
        {j, 0, NIntervalosL}];
    x = j;
    For[i = 0, i < NIteraciones, i ++,
        HeLlegado = False;
        PuntoActual = {x, NIntervalosT + 1};
        While[HeLlegado == False,
            Direccion = Random[];
            If[Direccion < R/(2 + 2R),
                PuntoActual = PuntoActual + {-1, 0},
                If[Direccion < 2R/(2 + 2R),
                    PuntoActual = PuntoActual + {1, 0},
                    If[Direccion < 3R/(2 + 2R),
                        PuntoActual = PuntoActual + {-1, -1},
                        If[Direccion < 4R/(2 + 2R),
                            PuntoActual = PuntoActual + {1, -1},
                            PuntoActual = PuntoActual + {0, -1}]]];
                If[PuntoActual[[1]] == 1 || PuntoActual[[2]] == 1 ||
                    PuntoActual[[1]] == NIntervalosL + 1, HeLlegado = True,]];
            ProbPunto[[PuntoActual[[2]], PuntoActual[[1]]]] =
                ProbPunto[[PuntoActual[[2]], PuntoActual[[1]]]] + 1];
        ProbPunto = ProbPunto/NIteraciones;
        A = ProbPunto * TemperaturasCNMC;
        TemperaturasCNMC[[NIntervalosT + 1, x]] =
            Sum[A[[i + 1, j + 1]], {i, 0, NIntervalosT},
                {j, 0, NIntervalosL}]]

In[31]:= TemperaturasCNMC[[NIntervalosT + 1]]

Out[31]= {0, 0.052, 0.0776, 0.1242, 0.1418,
    0.1588, 0.1322, 0.1186, 0.0926, 0.046, 0}
```

## 2.7 Comparativa

A continuación una comparativa de los resultados obtenidos con el método tradicional y el método MC. Se incluyen tres gráficas. La primera es la correspondiente a  $R = 1/3$ , la segunda es para un valor de  $R$  de 1 (el método explícito no hubiese funcionado con esta  $R$ ) y la tercera



es para  $R = 2$ . Con este último valor de  $R$  se observa que la versión tradicional sigue funcionando pero no la MC.

### 3 Estudio del error cometido por el método Monte Carlo

#### 3.1 Introducción al anova

Antes de exponer el estudio realizado, es necesaria una breve introducción al diseño de experimentos y las técnicas del ANOVA (más información en [6]).

El Análisis de la Varianza (ANalysis Of VAriance) es una herramienta de la Inferencia Estadística que sirve para estudiar observaciones que dependen de varios factores. Fue desarrollado en 1930 por R. A. Fisher.

Para introducir el ANOVA lo mejor es valerse de un ejemplo. El primer paso en un ANOVA es decidir cuál es la variable respuesta (*por ejemplo, la resistencia a compresión de una probeta de hormigón*). A continuación se decide cuáles son los factores que pueden influir en la variable respuesta (*cantidad de agua, cantidad de cemento y calidad de los áridos*). Se decide también a qué niveles trabajarán dichos factores (*5, 6 o 7 litros de agua, 12 o 13 kg de cemento y áridos rodados o áridos machacados*). En el ejemplo del hormigón se dice que *agua* y *cemento* son factores cuantitativos y que *tipo de áridos* es un factor cualitativo. Las técnicas para estudiar unos y otros son diferentes. Además, se dice que *agua* se estudiará a 3 niveles y que *cemento* y *tipo de áridos* se estudiarán a dos niveles. Lo siguiente es realizar los experimentos. Para ello se fabricarían varias probetas de hormigón con los factores a distintos niveles. Si se ensayan todas las combinaciones posibles una sola vez, se dice que se ha hecho un diseño factorial sin replicaciones; si se ensayan todas las combinaciones y además varias veces, se dice que se ha hecho un diseño factorial con replicaciones.

Una vez se han recopilado todos los datos, se procede a su análisis. Sin entrar en mucho detalle, el fundamento del ANOVA es el siguiente. El total de datos observados tendrá una variabilidad. Si se descompone esa variabilidad en la debida a los distintos factores y en la debida al propio proceso (varianza residual) se podrán comparar unas con otras y ver qué factores son significativos.

#### 3.2 Planteamiento del estudio

A lo largo de este trabajo, se ha experimentado con una versión MC del método explícito para resolver la ecuación del calor. La versión MC es una aproximación del método explícito y converge a él al aumentar el número de iteraciones. La pregunta que se plantea es ¿qué factores

pueden influir en esa convergencia? La variable respuesta a estudiar es por lo tanto el error que introduce el método Monte Carlo. Este error se ha medido como la media de las diferencias de temperaturas en el instante final de todos los puntos de la barra con respecto al método explícito tradicional.

Con respecto a los factores que pueden afectar a la convergencia, se han estudiado cuatro (ver cuadro 7.1). En primer lugar, el más claro, es el número de *iteraciones*. Se ha estudiado a tres niveles para poder descomponerlo en su parte lineal y su parte cuadrática (esto quedará más claro después del análisis). También se consideró que podía ser significativa la cantidad de *subdivisiones* de la barra. Otro factor estudiado es si la temperatura en los *extremos* se mantiene constante o no. Por último, en el ejemplo desarrollado en el trabajo, la temperatura inicial era siempre *positiva*, así que se pensó en reducir tanto esta como las temperaturas en los extremos para que hubiese valores positivos y negativos de temperatura. No olvidar que no se está estudiando la precisión del método explícito, sino la convergencia de la versión MC hacia la versión tradicional.

Se ensayaron todas las combinaciones posibles, cada una tres veces, para un total de 72 pruebas.

	+	0	-
Iteraciones	50	125	200
Subdivisiones	10	-	20
Extremos	cte	-	seno
Positivo	pos	-	pos/neg

Table 7.1: Significado de los distintos niveles a los que trabajan los factores

### 3.3 Análisis de los resultados

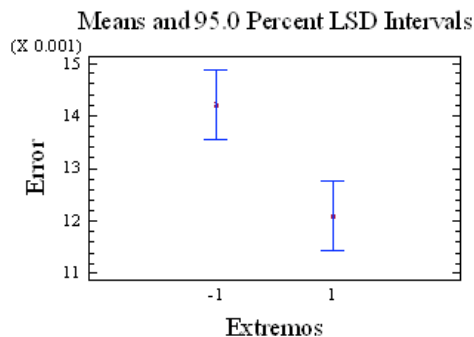
Una vez introducidos todos los datos en el **Statgraphics** se procedió a hacer el ANOVA. Después de quitar los efectos no significativos se obtuvo el siguiente resultado (ninguna interacción salió significativa):

Analysis of Variance for Error - Type III Sums of Squares

Source	Sum of Squares	Df	Mean Square	F-Ratio	P-Value
MAIN EFFECTS					
A:Extremos	0.0000810689	1	0.0000810689	8.54	0.0047
B:Iteraciones	0.00114571	2	0.000572857	60.37	0.0000
RESIDUAL	0.000645289	68	0.0000948955		
TOTAL (CORRECTED)	0.00187207	71			

All F-ratios are based on the residual mean square error.

Extremos es un factor cualitativo, así que se estudian sus intervalos LSD:



Esto indica que se produce menos error cuando se mantienen los extremos constantes.

Como iteraciones es un factor cuantitativo a tres niveles, se puede descomponer en su componente lineal y su componente cuadrática. Para ello se obtiene la tabla de medias

Level	Count	Mean
-1	24	0.00966083
0	24	0.0110525
1	24	0.0187325

y se realizan los siguientes cálculos:

$$Z_{lineal} = \sum \lambda_i \bar{x}_i = -1 \cdot 0.00966083 + 0 \cdot 0.0110525 + 1 \cdot 0.0187325 = 0.00907167$$

$$Z_{cuadrático} = \sum \lambda_i \bar{x}_i = 1 \cdot 0.00966083 - 2 \cdot 0.0110525 + 1 \cdot 0.0187325 = 0.00628833$$

$$SC_{lineal} = CM = \frac{n \cdot Z^2}{\sum \lambda_i^2} = \frac{24 \cdot 0.00907167^2}{(-1)^2 + 0^2 + 1^2} = 9.87542 \cdot 10^{-4}$$

$$SC_{cuadrática} = CM = \frac{n \cdot Z^2}{\sum \lambda_i^2} = \frac{24 \cdot 0.00628833^2}{(-1)^2 + 0^2 + 1^2} = 4.74517 \cdot 10^{-4}$$

Dividiendo por el cuadrado medio residual se obtienen los F-ratios, que son 104.06 y 50.00. Comparando con las  $F$  de tablas se tiene  $F_{1,68}(\alpha = 0.05) \approx 4$ . Por lo tanto, las dos componentes son significativas y su interpretación es que a medida que se aumenta el número de iteraciones disminuye el error (componente lineal) pero que el efecto es cada vez menor (componente cuadrática).

A continuación se deben estudiar los residuos al cuadrado para ver si se puede sacar alguna conclusión sobre la varianza:

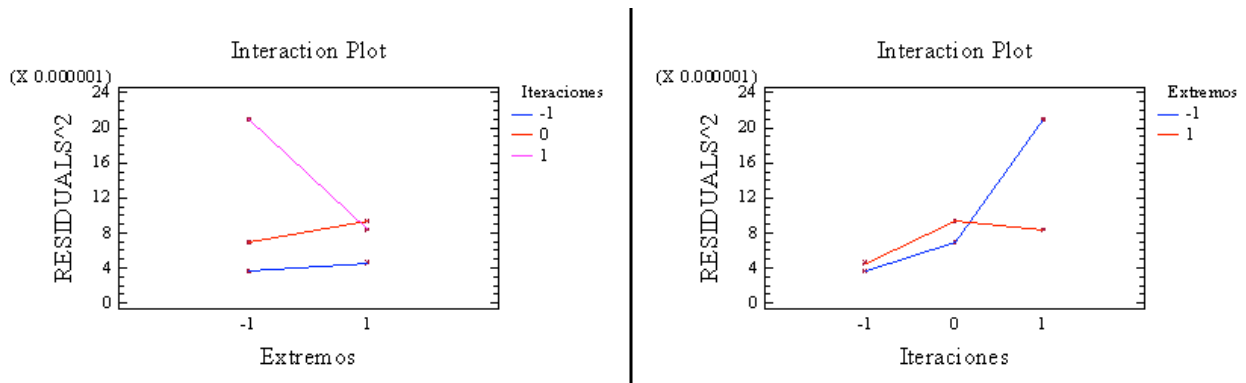


Analysis of Variance for RESIDUALS<sup>2</sup> - Type III Sums of Squares

Source	Sum of Squares	Df	Mean Square	F-Ratio	P-Value
MAIN EFFECTS					
A:Extremos	1.71515E-10	1	1.71515E-10	1.32	0.2545
B:Iteraciones	1.37628E-9	2	6.88138E-10	5.30	0.0073
INTERACTIONS					
AB	8.29886E-10	2	4.14943E-10	3.20	0.0473
RESIDUAL	8.56529E-9	66	1.29777E-10		
TOTAL (CORRECTED)	1.0943E-8	71			

All F-ratios are based on the residual mean square error.

Como era de esperar, Iteraciones salió significativo, pero además, salió significativa su interacción con Extremos. Para interpretar estos efectos se utilizan las siguientes gráficas:



Del estudio de estas gráficas se deduce que la varianza de los datos es mayor cuantas menos iteraciones se hagan (valor 1), pero que se ve afectado por Extremos (cuando los extremos no se mantienen constantes el efecto es más acusado).

### 3.4 Conclusiones del análisis

Teniendo en cuenta que el objetivo es minimizar la media y la varianza se llega a la conclusión de que las condiciones operativas óptimas son Iteraciones a -1 y Extremos a 1. Esto no quiere decir que a partir de ahora siempre se deba utilizar el método Monte Carlo manteniendo los extremos constantes, ya que esto vendrá impuesto por el problema a resolver. Quiere decir que el método MC será más adecuado cuando el problema tenga los extremos constantes. Además, se sabe que se tendrá que utilizar una cantidad elevada de iteraciones para obtener buenos resultados, pero que aumentar esa cantidad demasiado tampoco tendrá sentido (debido a la componente cuadrática).

Utilizando los datos obtenidos mediante el análisis de residuos se obtiene el intervalo de confianza para la media del error en las condiciones operativas óptimas, que es  $0'008600 \pm 0'004153$ . Se hicieron tres pruebas más con los factores significativos en sus niveles óptimos obteniendo 0.01102, 0.006414 y 0.006670. Se puede observar que las tres están dentro del intervalo de confianza, lo cual es un indicativo de que el análisis ha sido correcto.

## 4 APENDICE. Aproximación de la derivada

Para resolver las ecuaciones en derivadas parciales por el método de diferencias finitas se necesita un método *aproximado* para obtener las derivadas. Este método debe dar el valor de la derivada de la función  $u$  en un punto a partir de valores (discretos) de  $u$ . Es importante puntualizar qué se entiende por *aproximado*. Un método aproximado es aquel que se pueda acercar a la solución exacta tanto como se quiera, o lo que es lo mismo, que su error esté acotado y se pueda reducir arbitrariamente. Con esta condición es natural obtener las aproximaciones a partir de la fórmula de Taylor [5].

Para obtener  $f'(x)$  se procede de la siguiente manera:

$$f(x) = f(a) + \frac{(x-a)}{1!} \cdot f'(a) + \frac{(x-a)^2}{2!} \cdot f''(a + \theta \cdot (x-a)) \quad \text{con } 0 < \theta < 1$$

Si ahora se hace que la función  $f(x)$  sea  $f(x + \Delta x)$  y la función  $f(a)$  sea  $f(x)$  queda:

$$\begin{aligned} f(x + \Delta x) &= f(x) + (x + \Delta x - x) \cdot f'(x) + \\ &\quad + \frac{(x + \Delta x - x)^2}{2} \cdot f''(x + \theta \cdot (x + \Delta x - x)) \\ &= f(x) + \Delta x \cdot f'(x) + \frac{\Delta x^2}{2} \cdot f''(x + \theta \cdot \Delta x) \end{aligned}$$

Reordenando:

$$f'(x) = \frac{f(x + \Delta x) - f(x)}{\Delta x} + \frac{\Delta x \cdot f''(x + \theta \cdot \Delta x)}{2}$$

Si  $f$  es continua y teniendo en cuenta que  $0 < \theta < 1$ , haciendo  $\Delta x$  tan pequeño como se quiera se hace  $\frac{\Delta x \cdot f''(x + \theta \Delta x)}{2}$  tan pequeño como se quiera y  $\frac{f(x + \Delta x) - f(x)}{\Delta x}$  será una aproximación de  $f'(x)$  tan buena como se quiera.

A esta aproximación de la derivada se le conoce como fórmula de la *diferencia adelantada*. Si en lugar de desarrollar la fórmula de Taylor para  $f(x + \Delta x)$  se hubiese hecho para  $f(x - \Delta x)$  se hubiera llegado a

$$f'(x) \approx \frac{f(x) - f(x - \Delta x)}{\Delta x}$$

que se conoce como fórmula de la *diferencia atrasada*. Si se suman las fórmulas de Taylor para los dos casos se llega a:

$$f'(x) \approx \frac{f(x + \Delta x) - f(x - \Delta x)}{2\Delta x}$$

que se conoce como fórmula de la *diferencia central*.

La figura 7.2 contiene la interpretación gráfica de las fórmulas. Se observa que la tangente de  $f$  en  $(x, f(x))$  se aproxima por la secante de dos puntos cercanos.

Para la segunda derivada se puede proceder como para la fórmula de la diferencia central pero desarrollando un término más en la fórmula de Taylor. Las fórmulas de las que se parte son:

$$f(x + \Delta x) = f(x) + \Delta x \cdot f'(x) + \frac{\Delta x^2}{2} \cdot f''(x) + \frac{\Delta x^3}{6} \cdot f'''(x + \theta \cdot \Delta x)$$

$$f(x - \Delta x) = f(x) - \Delta x \cdot f'(x) + \frac{\Delta x^2}{2} \cdot f''(x) - \frac{\Delta x^3}{6} \cdot f'''(x - \theta \cdot \Delta x)$$

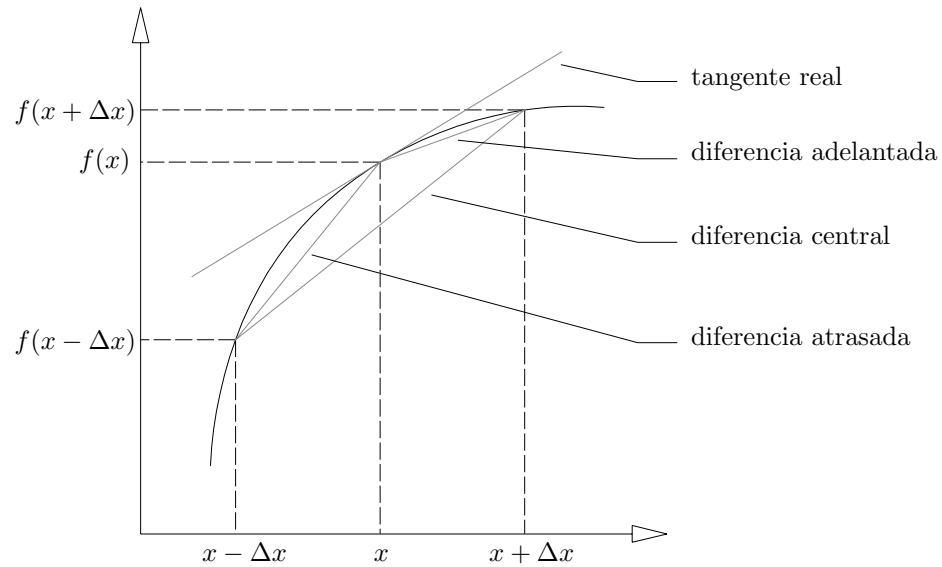


Figure 7.2: Interpretación gráfica de las distintas aproximaciones de la derivada

Que sumadas miembro a miembro dan:

$$f(x + \Delta x) + f(x - \Delta x) = 2 \cdot f(x) + \Delta x^2 \cdot f''(x) + \frac{\Delta x^3}{6} \cdot (f'''(x + \theta \cdot \Delta x) - f'''(x - \theta \cdot \Delta x))$$

Reordenando:

$$f''(x) = \frac{f(x + \Delta x) + f(x - \Delta x) - 2 \cdot f(x)}{\Delta x^2} + \frac{\Delta x \cdot (f'''(x + \theta \cdot \Delta x) - f'''(x - \theta \cdot \Delta x))}{6}$$

Finalmente, si se generaliza para funciones de dos variables y se utiliza la notación habitual para los métodos de diferencias finitas se tiene:

$$\begin{aligned} \frac{\partial u}{\partial x} &\approx \frac{u_{i+1,j} - u_{i,j}}{\Delta x} && \text{(diferencia adelantada)} \\ \frac{\partial^2 u}{\partial x^2} &\approx \frac{u_{i+1,j} + u_{i-1,j} - 2 \cdot u_{i,j}}{\Delta x^2} \end{aligned}$$

# Referencias

- [1] Dr. Everett F. Carter Jr, Random Walks, Markov Chains and the Monte Carlo Method, <http://www.taygeta.com/rwalks/rwalks.html>, 1996
- [2] Rubinstein, R. Y., Simulation and the Monte Carlo Method, Wiley, New York, 1981
- [3] A.N. Tijonov y A.A. Samarsky, Ecuaciones de la física matemática, Mir, 1980
- [4] Smith, G.D., Numerical solution of partial differential equations: Finite difference methods, Clarendon Press, 1993
- [5] Rafael Ballesteros Murillo ... [et al.], Curso de cálculo, SPUPV, 1998
- [6] Rafael Romero Villafranca y Luisa Rosa Zúnica Ramajo, Estadística (Proyecto de innovación educativa): conceptos básicos, análisis de la varianza, diseño de experimentos, modelos de regresión, procesos estocásticos, SPUPV, 1993