

## ALGORITMO PARA SIMULAR LA EVOLUCIÓN TEMPORAL DE LA CONDUCCIÓN DEL CALOR EN DOS DIMENSIONES

Lyon, Euclides<sup>1</sup>

(Recibido Marzo 2013, Aceptado Junio 2013)

<sup>1</sup> UNEXPO, Puerto Ordaz, Venezuela

[ealyon.doctorando@unexpo.edu.ve](mailto:ealyon.doctorando@unexpo.edu.ve), [ealyon.unexpo@gmail.com](mailto:ealyon.unexpo@gmail.com)

**Resumen:** Se propone un algoritmo que simula la evolución temporal y en dos dimensiones de la conducción del calor en un material metálico, homogéneo e isótropo. El algoritmo se obtiene mediante la aplicación del principio de superposición a las ecuaciones en diferencias finitas y evoluciona desde las condiciones iniciales hasta el estado estable, utilizando el promedio de la vecindad en cada dirección para el punto de la grilla que se calcula. Se aplica el análisis fundamental de convergencia y estabilidad, se toma como datos patrones para el contraste, el estado estable proveniente de la solución del laplaciano. El algoritmo propuesto presenta un comportamiento aceptable y predecible dentro de las restricciones propias del dominio y el esquema de discretización.

**Palabras claves:** Algoritmo/ Diferencias finitas/ Superposición/ Promedio de la Vecindad.

## ALGORITHM TO SIMULATE THE TEMPORAL EVOLUTION OF THE HEAT CONDUCTION IN TWO DIMENSIONS

**Abstract:** It is presented an algorithm that simulates the temporary evolution, in two dimensions, of the conduction of the heat in a metallic, isotropous homogeneous material. The algorithm is obtained by means of the application of the superposition principle to the equations in finite Differences and evolves from the initial conditions up to the stable condition, using the vicinity average in every direction for the point of the grid that is calculated. There is applied the fundamental analysis of convergence and stability. It takes the stable state from the Laplace's Solution as pattern data for the contrast. The proposed algorithm presents an acceptable and predictable behavior inside the own restrictions of the domain and the discretization scheme.

**Keywords:** Algorithm/ Finite Differences/ Superposition/Vicinity Average

### I. INTRODUCCIÓN

Se planteó la necesidad de estudiar los conceptos fundamentales de la simulación mediante el desarrollo de un algoritmo, basado en discretización espacial y temporal mediante aproximación en diferencias finitas y método explícito, para simular la evolución temporal en 2D, del fenómeno físico de la conducción del calor, modelado en tiempo continuo con ecuación diferencial parcial parabólica.

Se desarrolló un algoritmo para la solución del problema, se estableció una metodología para el análisis de la convergencia y estabilidad intrínsecas al algoritmo y su relación con la dinámica del fenómeno estudiado, en esta metodología se fundamenta en el contraste de los resultados del algoritmo de evolución temporal, mediante comparación con los obtenidos del laplaciano. Se toma como material de experimentación el reportado en el antecedente [1].

Se toma como antecedentes los artículos [1-2], en el primero se estudia el problema de la conducción del calor en 1D con fuente exógena, contrastan los resultados con

una solución analítica. En [2] se establece la formulación del problema basados en el la ecuación de difusión de calor en 3D, con una fuente de calor endógena acoplada con un suministro de energía eléctrica. El trabajo de investigación presente se relaciona con el antecedente [1] en que se toma de el, los datos del metal utilizado para la simulación en 1D, se diferencia en que se desarrolló un algoritmo para simular el fenómeno de conducción expandido a 2D y se contrastó con los resultados de estado estable obtenidos del laplaciano. Se relaciona con el antecedente [2] en que se toma la ecuación de difusión y la particularizamos a dos dimensiones, sin fuente permanente de calor.

El objetivo es estudiar los conceptos fundamentales de la simulación numérica, mediante la propuesta de un algoritmo basado en método explícito de diferencias finitas.

El objeto de estudio es el fundamento de la simulación numérica, específicamente el refinamiento de mallas, condiciones iniciales y de bordes, convergencia y estabilidad del algoritmo de solución de la ecuación

diferencial parcial parabólica aproximada mediante diferencias finitas y método explícito. Se utiliza como fenómeno físico a estudiar la conducción de calor en un metal homogéneo e isótropo con una fuente de calor exógena al metal que proporciona el patrón de calentamiento inicial.

Se ejecutaron nueve pasos orientados por los siguientes objetivos específicos:

1. Estudiar el fenómeno físico a simular en 2D para un metal homogéneo e isótropo.
2. Establecer el conjunto de ecuaciones en diferencias finitas DF, para dar solución aproximada a la ecuación (2) mediante una ecuación en diferencias finitas y método explícito; el patrón inicial de temperatura para el dominio 2D de una placa de 20 cm. x 20 cm.
3. Establecer el núcleo del algoritmo iterativo.
4. Establecer el límite teórico de estabilidad según Von Neuman y las condiciones de borde para garantizar la consistencia en la iteración.
5. Codificar el algoritmo en Matlab™.
6. Estudio de la convergencia y estabilidad del algoritmo.
7. Afinar la ventana temporal de simulación (Tv).
8. Afinar el paso temporal Δt.
9. Afinar el paso espacial Δx=Δy.

En lo que resta del artículo se presenta el desarrollo, que contiene métodos, materiales, software, los resultados y su análisis; las conclusiones y finalmente las referencias.

## II. DESARROLLO

### 1. Métodos y materiales.

El trabajo se desarrolló en nueve pasos, cada uno orientado por un objetivo específico:

1. Estudiar el fenómeno físico a simular en 2D para un metal homogéneo e isótropo.

La conducción del calor en 3D es modelada en [2], como muestra la ecuación (1)

$$\frac{\partial}{\partial x} \left( k \frac{\partial T}{\partial x} \right) + \frac{\partial}{\partial y} \left( k \frac{\partial T}{\partial y} \right) + \frac{\partial}{\partial z} \left( k \frac{\partial T}{\partial z} \right) + \dot{q} = c_p \rho \frac{\partial T}{\partial t} \quad (1)$$

La ecuación (1) se particulariza para las siguientes condiciones: una placa rectangular de un material metálico homogéneo e isótropo, sin fuente de calor permanente, dando como resultado la ecuación (2)

$$\alpha \left( \frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} \right) = \frac{\partial T}{\partial t} \quad \alpha = k / (c_p \rho) \quad (2)$$

Siendo α definido en [3] como coeficiente de difusión del calor y depende del material utilizado, k es el coeficiente de conductividad térmica, ρ es la densidad y c<sub>p</sub> el calor

específico (p. 98). Se tomó como material para el estudio el acero reportado en [1]

Tabla IV. Propiedades del material objeto de estudio

Densidad	Conductividad térmica	Calor específico	Difusión térmica
$\rho$	$k$	$c$	$\alpha$
7.8 g/cm <sup>3</sup>	0.13 cal/s.cm.°C	0.11 cal/g.°C	0.15 cm <sup>2</sup> /s

2. Establecer el conjunto de ecuaciones en diferencias finitas DF, para dar solución aproximada a la ecuación (2) mediante una ecuación en diferencias finitas y método explícito, el patrón inicial de temperatura para el dominio 2D de una placa de 20 cm. x 20 cm.

Las derivadas temporales y espaciales se aproximaron utilizando lo propuesto en [4], según tabla reportada en (p. 23). Se aproximaron las derivadas temporales con diferencias progresivas dada por las ecuaciones (3 y 4).

$$\frac{\partial T(t, x)}{\partial t} = \frac{T(x, t + \Delta t) - T(x, t)}{\Delta t} + O(\Delta t) \quad (3)$$

$$\frac{\partial T(t, y)}{\partial t} = \frac{T(y, t + \Delta t) - T(y, t)}{\Delta t} + O(\Delta t) \quad (4)$$

Se aproximaron las segundas derivadas espaciales en x e y mediante diferencia simétrica, como lo muestra la ecuación (5 y 6).

$$\frac{\partial^2 T}{\partial x^2} = \frac{T(x + \Delta x, t) - 2T(x, t) + T(x - \Delta x, t)}{(\Delta x)^2} + O(\Delta x^2) \quad (5)$$

$$\frac{\partial^2 T}{\partial y^2} = \frac{T(y + \Delta y, t) - 2T(y, t) + T(y - \Delta y, t)}{(\Delta y)^2} + O(\Delta y^2) \quad (6)$$

Siendo Δt, Δx y Δy los incrementos finitos de la variable temporal y espaciales respectivamente y O(Δt), O(Δx<sup>2</sup>) y O(Δy<sup>2</sup>) los errores de precisión cometidos en la aproximación de las derivadas.

El patrón inicial de temperatura incluyendo los bordes, está dado por una distribución de Gauss para 2D, ecuación (7).

$$T(0, x, y) = 100 \exp(-0.01d^2) \dots d = \sqrt{(x - xc)^2 + (y - yc)^2} \quad (7)$$

Siendo d el radio de la isoterma, xc e yc las coordenada x e y del centro de la placa.

El patrón inicial de temperaturas visto en la grilla X-Y para t = 0 queda como se muestra en la GRÁFICA 1, en la que se representan las líneas isoterma de la superficie del material especificado en la TABLA I, para una placa de 20 cm. x 20 cm. un Δx=2 cm., un Δy=2 cm., luego de haber sido puesta a una fuente de calor en el centro.

3. Establecer el núcleo del algoritmo iterativo.

Dado que la ecuación (2) no contiene productos entre derivadas, ni potencias de la variable dependiente. El núcleo del algoritmo iterativo se obtiene combinando, mediante un procedimiento de superposición, las ecuaciones en diferencias (3-6), dándole notación matricial,  $i$  para la coordenada  $x$ ,  $j$  para la coordenada  $y$  y  $k$  para la variable  $t$ . Para simplicidad en las formulaciones vamos a garantizar  $\Delta x = \Delta y$ , dando como resultado la ecuación (8)

$$T(k+1, j, i) = (1-2r)T(k, j, i) + rV$$

$$r = \alpha \frac{\Delta t}{(\Delta x)^2} \quad (8)$$

Siendo  $V$  el promedio de valores de la vecindad superficial de  $T(k, j, i)$ , en cada eje de iteración como se muestra en (9)

$$V = (1/2)(T(k, j+1, i) + T(k, j-1, i) + T(k, j, i+1) + T(k, j, i-1)) \quad (9)$$

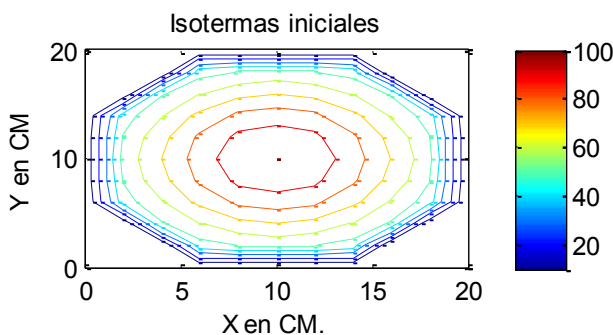


Figura 1. Condiciones Iniciales

4. Establecer el límite teórico de estabilidad según Von Neuman y las condiciones de borde para garantizar la consistencia en la iteración.

En la ecuación (8), el coeficiente  $r = \alpha(\Delta t/\Delta x^2)$  es determinante en el proceso iterativo en el avance del algoritmo hacia el estado estable. En [4] se establece que para garantizar la estabilidad el coeficiente debe cumplir con  $0 \leq r \leq 0.5$ , lo cual implica que el límite de estabilidad se da cuando  $\Delta t = \Delta x^2 / (2\alpha)$  (p. 79). Lo anterior garantiza que el factor  $(1-2r)$  no se torne retrógrado para el cálculo iterativo.

Para el cálculo iterativo de la vecindad  $V$  de las ecuaciones (8 y 9), se requiere de nodos sombreados alrededor de la grilla para garantizar la consistencia en la iteración en los nodos de los bordes. La simulación se iniciará con un paso  $\Delta x = 2$  y  $\Delta t = 0.5$  lo cual implica un  $r = 0.019$  esto predice un comportamiento sin que aparezca el fenómeno de la inestabilidad.

5. Codificar el algoritmo en Matlab™.

El núcleo del algoritmo iterativo es la ecuación (8), y será implementado en el lenguaje de programación de

Matlab™. MATLAB Versión 7.4.0.287 (R2007a). Operating System: Microsoft Windows XP Version 5.1 (Build 2600: Service Pack 3).

6. Estudio de la convergencia y estabilidad del algoritmo.

Utilizando la solución del laplaciano como patrón y la norma L1 como parámetro de optimización, se diseñó una batería de experimentos para comparar el estado estable de la evolución temporal del algoritmo, con el estado estable obtenido mediante el laplaciano.

La solución numérica del laplaciano la tomamos de [4] (p. 35)

$$T(i, j) = \frac{T(i+1, j) + T(i-1, j) + b^2 T(i, j+1) + b^2 T(i, j-1)}{2(1+b^2)} \quad (10)$$

$$b = (\Delta x / \Delta y)$$

7. Afinar la ventana temporal de simulación  $T_v$ .

Estudiar el comportamiento del algoritmo al afinar la ventana temporal de simulación  $T_v$  dejando fijos  $\Delta t$  y  $\Delta x$ , midiendo el acumulado L1.

8. Afinar  $\Delta t$ .

Estudiar el comportamiento de la evolución temporal al afinar  $\Delta t$  dejando fijos  $\Delta x$  y  $T_v$ , midiendo el acumulado L1.

9. Afinar  $\Delta x$ .

Estudiar el comportamiento del algoritmo al afinar  $\Delta x$  dejando fijos  $\Delta t$  y  $T_v$ , midiendo el acumulado L1.

Resultados y su análisis

1. Algoritmo implementado en Matlab™.

```
% Define dominio y grilla.
p=0; % Inicio dominio en x
q=20; % Fin dominio en x
r=0; % Inicio dominio en y
s=20; % Fin dominio en y
vx=0.15; % Difusión en x
vy=0.15; % Difusión en y
NX=51; % Cantidad puntos grilla en x
NY=51; % Cantidad puntos grilla en y
dx=((q-p)/(NX-1)); % Incrementos en x
dy=((s-r)/(NY-1)); % Incrementos en y
x=p:dx:q; % Vector puntos grilla en x
y=r:dy:s; % Vector puntos grilla en y
Tv=24; % Tiempo simulación
dt=0.5; % Incrementos tiempo
NT=round(Tv/dt); % Cantidad iteraciones en el tiempo
E=((dt*vx)/(dx*dx)); % Factor de estabilidad
T=zeros(NX,NY); % Inicializa grilla
```

% Define condiciones Iniciales.

```
To=T;
for j=1:NY
    for i=1:NX
        To(j,i)=100*Gauss2D(x(i),y(j),p, q, r, s);
    end
end
```

```
% Algoritmo Laplaciano
ToL=To; % Condiciones iniciales Laplaciano
Ttemp=zeros(NY+2,NX+2); % Grilla mas nodos de sombra
Ttemp(2:NY+1,2:NX+1)=ToL; % Condiciones iniciales
ToL=Ttemp; % Condiciones iniciales mas nodos de sombra
clear Ttemp % Borra Ttemp del espacio de trabajo
b=(dx/dy); % Factor de avance
Den=2*(1+(b*b)); % Denominador
T=ToL; % Carga condiciones iniciales
for j=2:NY+1
    for i=2:NX+1
        T(j,i)=(T(j+1,i)+T(j-1,i)+((b*b)*(T(j,i+1)+T(j,i-1))))/Den;
    end % Lazo dirección x
end % Lazo dirección y
```

```
% Algoritmo propuesto
ToP=To; % Condiciones iniciales solución parabólica
V=0.0; % Promedio vecindad punto en iteración
for k=1:NT
    Ttemp=zeros(NY+2,NX+2); % Grilla mas nodos de sombra
    Ttemp(2:NY+1,2:NX+1)=ToP; % Condiciones iniciales
    ToP=Ttemp; % Condiciones iniciales mas nodos de sombra
    clear Ttemp % Borra Ttemp del espacio de trabajo
    for j=2:NY+1
        for i=2:NX+1
            V=(ToP(j,i+1)+ToP(j,i-1)+ToP(j+1,i)+ToP(j-1,i))/2;
            T(j,i)=((1-2*E)*ToP(j,i))+E*V;
        end % Lazo en la dirección x
    end % Lazo en la dirección y
    ToP=T(2:NY+1,2:NX+1); % Memoriza solución estimada
end % Lazo de tiempo
```

```
%L1 comparar resultados solución parabólica Vs. Laplaciano
L1=0;
for j=2:NY
    for i=2:NX
        L1=L1+abs(ToL(j,i)-ToP(j,i));
    end
end
```

**2. Afinación de la ventana temporal (Tv).**

Se estudió el comportamiento del algoritmo al afinar la ventana temporal de simulación (Tv) dejando fijos Δt = 0.5 y Δx = 2, midiendo el acumulado L1.

La Tabla II recoge los resultados del experimento de afinación del tiempo de simulación Tv.

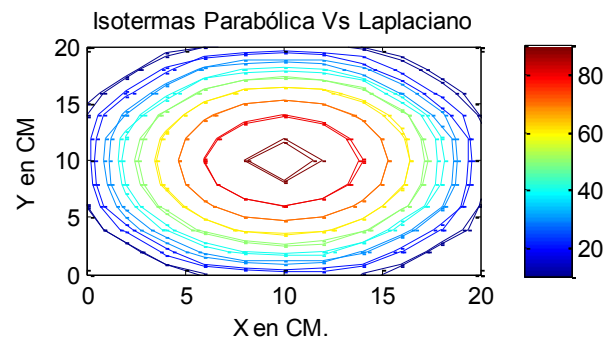
En la Tabla II, se observa que durante todo el experimento, la estabilidad del algoritmo se mantuvo insensible, ante variaciones del tiempo de simulación Tv, y la exactitud alcanza su máximo (L1 mínimo) para Tv=24.

La Figura 2 muestra el resultado del algoritmo para Tv = 24 seg., se puede observar que las isothermas casi se superponen, lo cual se refleja en el valor mínimo de L1=180 de la Tabla II.

**Tabla V. Resultados de la afinación del tiempo de simulación**

Δx=2 cm., Δt=0.5 seg., r = 0.018, Tv = Variable			
Tv	L1	COMENTARIOS	
		L	P
30	223	Estable	Estable
28	201	Estable	Estable
27	192	Estable	Estable
26	185	Estable	Estable
25	181	Estable	Estable
24	180	Estable	Estable
23	184	Estable	Estable
22	190	Estable	Estable
20	211	Estable	Estable

L: Laplaciano, P: Parabólica



**Figura 2. Máxima exactitud (L1=180) al afinar Tv**

**3. Afinación del paso temporal Δt.**

Se estudió el comportamiento del algoritmo al afinar el paso temporal Δt dejando fijos Δx = 2 y Tv = 24 seg., midiendo el acumulado L1.

La Tabla III recoge los resultados del experimento de afinación del paso temporal Δt y muestra que durante el experimento, la estabilidad del algoritmo se vió afectada solo cuando el paso temporal Δt se iguala a la ventana de tiempo Tv = 24 seg., esto corrobora el límite teórico de estabilidad r = 0.5 en este caso r alcanzó el valor 0.9.

La exactitud disminuye para Δt > 0.5, se mantiene constante L1=180 mientras Δt disminuye y la cantidad de iteraciones NT crece, lo cual implica que el esfuerzo computacional es inútil.

La Figura 3 muestra el resultado del algoritmo para Δt = 24 seg., se puede observar el comportamiento inestable propio del algoritmo que no tiene nada que ver con el proceso físico de conducción del calor

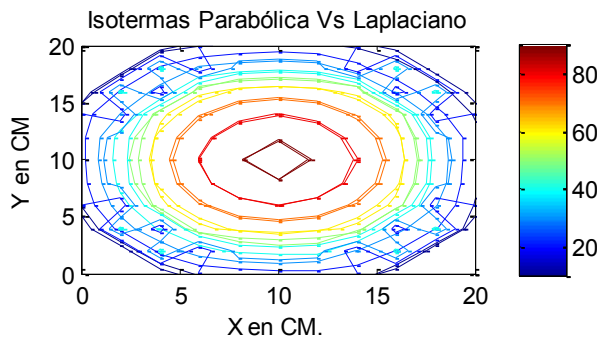
**Tabla VI. Resultados de la afinación del paso temporal**

$\Delta x=2$ cm., $\Delta t$ =Variable, $r$ = Variable, $T_v = 24$ seg.					
$\Delta t$	L1	NT	$r$	COMENTARIOS	
				L	P
24	688	1	0.9	Estable	Inestable
12	256	2	0.45	Estable	Estable
10	228	2	0.37	Estable	Estable
8	191	3	0.30	Estable	Estable
6	187	4	0.22	Estable	Estable
4	184	6	0.15	Estable	Estable
2	182	12	0.08	Estable	Estable
1	181	24	0.04	Estable	Estable
0.5	180	48	0.02	Estable	Estable
0.25	180	96	0.01	Estable	Estable
0.125	180	192	0.005	Estable	Estable
0.0625	180	384	0.002	Estable	Estable

L: Laplaciano, P: Parabólica

La exactitud disminuye para  $\Delta t > 0.5$ , se mantiene constante  $L1=180$  mientras  $\Delta t$  disminuye y la cantidad de iteraciones NT crece, lo cual implica que el esfuerzo computacional es inútil.

La Figura 3 muestra el resultado del algoritmo para  $\Delta t = 24$  seg., se puede observar el comportamiento inestable propio del algoritmo que no tiene nada que ver con el proceso físico de conducción del calor.



**Figura 3. Comportamiento inestable para  $\Delta t = 24$  seg.**

**4. Afinación del paso espacial  $\Delta x$ .**

Se estudió el comportamiento del algoritmo al afinar el paso  $\Delta x$ , dejando fijos  $\Delta t = 0.5$  seg. y  $T_v = 24$  seg., midiendo el acumulado L1.

La Tabla IV recoge los resultados del experimento de afinación del paso  $\Delta x$ .

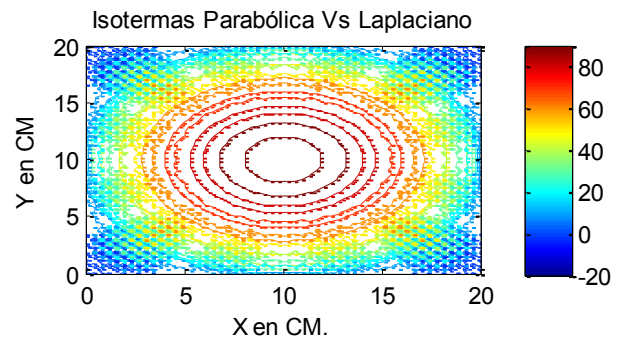
**Tabla VII. Resultados de la afinación del paso espacial**

$\Delta x=\Delta y$ =Variable., $\Delta t=0.5$ seg., $r$ = Variable, $T_v = 24$ seg.					
$\Delta x$	L1	NX	$r$	COMENTARIOS	
				L	P
2	180	11	0.0187	Estable	Estable
1	1703	21	0.075	Estable	Estable
0.5	10180	41	0.3	Estable	Estable
0.4	17116	51	0.47	Estable	Estable
0.38	27960	54	0.53	Estable	Inestable

L: Laplaciano, P: Parabólica

La Tabla IV, muestra que durante el experimento, la estabilidad del algoritmo se vió afectada solo cuando el paso temporal  $\Delta x = 0.38$  cm., lo que ocasiona la corroboración del límite teórico de estabilidad  $r = 0.5$  en este caso  $r$  alcanzó el valor 0.53. La exactitud disminuye para  $\Delta x < 2$  cm..

La Figura 4 muestra el resultado del algoritmo para  $\Delta x = 0.38$  cm., se puede observar el comportamiento inestable propio del algoritmo que no tiene nada que ver con el proceso físico de conducción del calor.



**Figura 4. Comportamiento inestable para  $\Delta x = 0.38$  cm.**

**III. CONCLUSIONES**

1. El algoritmo propuesto es fácil de implementar, pero requiere de un conocimiento suficiente de la dinámica del sistema a simular, para evitar llevar el algoritmo a la inestabilidad por un muestreo en el tiempo o paso espacial inadecuado.
2. El modelado mediante ecuaciones diferenciales parciales parabólicas permite estudiar el fenómeno en su evolución temporal y espacial, el algoritmo propuesto así lo permite.
3. La dinámica del sistema determinada por las características de difusión térmica del material, conjuntamente con la razón de paso temporal / espacial configuran la base para el análisis de convergencia y estabilidad del algoritmo.

4. El algoritmo propuesto en este artículo se comporta aceptablemente al comparar su estado estable con los resultados del laplaciano.
5. Para el caso de estudio la convergencia o exactitud del algoritmo varía con la ventana temporal de simulación.
6. Para el caso de estudio la estabilidad el algoritmo no es sensible a la variación de la ventana temporal.
7. Durante el refinamiento de los pasos temporal y espacial la estabilidad del algoritmo mostró mayor sensibilidad cuando variamos el paso espacial.
8. Durante la variación del paso temporal también se presenta la inestabilidad pero con menor sensibilidad.

#### IV. REFERENCIAS

1. Millán, Zulma, et al. *Simulación numérica. Ecuación de difusión*. Revista Iberoamericana de Ingeniería Mecánica 15.2 (2011): 29-38.
2. Gutiérrez, E., & Troyani, N. (1999). Formulación Variacional del Problema Termoeléctrico e una Celda de Reducción de Aluminio Hall-Héroult en Tres Dimensiones. Universidad, Ciencia y Tecnología, Marzo, 3(9), 25-29.
3. Romero, S., et al. *Introducción a las Ecuaciones en Derivadas Parciales (EDP's)*, Huelva, Servicio de Publicaciones. Universidad de Huelva, 2001, 195. [http://www.uhu.es/sixto.romero/EDP\\_libro.pdf](http://www.uhu.es/sixto.romero/EDP_libro.pdf) 19-07-2013
4. Causon, D. M., & Mingham, C. G. (2010). *Introductory Finite Difference Methods for PDEs*. Bookboon. Tomado de <http://80.191.221.11/files/research/site/ebooks/mechanics/introductory-finite-difference-methods-for-pdes.pdf> , 15-07-2013.