

Solución numérica de una ecuación de difusión - reacción por el método de diferencias finitas

Numerical solution of a diffusion equation - reaction by the finite difference method

Julio José Augusto Becerra Saucedo ¹

RESUMEN

En el presente artículo se ha resuelto de manera numérica el problema de difusión – reacción uno dimensional:

$$\begin{cases} u_t(x,t) = u_{xx}(x,t) + f(x,t,u(x,t)), & x \in (a,b), t > 0, \\ u_x(a,t) = u_x(b,t) = 0, & t > 0, \\ u(x,0) = x_0(x), & x \in (a,b). \end{cases}$$

donde f y u_0 on elementos de ciertos espacios funcionales. Los resultados de existencia y unicidad de soluciones así como la determinación de soluciones explosivas han sido demostrados utilizando el Método de Semigrupos de Operadores. Para encontrar la solución numérica se ha empleado el Método de Diferencias Finitas (MDF) con el esquema explícito, esto es, se han discretizado la derivada espacial de segundo orden utilizando diferencias centradas y la derivada temporal usando diferencias de primer orden hacia adelante. La programación del MDF se ha desarrollado en dos partes: la escritura del código y ejecución se realizaron en lenguaje C y, las gráficas fueron visualizadas en Scilab. Así mismo se han estudiado y analizado los criterios de estabilidad, consistencia y convergencia, llegando a concluir que el MDF es condicionalmente estable, consistente y convergente.

Palabras clave: Problema de difusión – reacción, solución explosiva, método de semigrupos, método de diferencias finitas, esquema explícito.

ABSTRACT

This article shows by numerical way the one-dimensional diffusion-reaction problema:

$$\begin{cases} u_t(x,t) = u_{xx}(x,t) + f(x,t,u(x,t)), & x \in (a,b), t > 0, \\ u_x(a,t) = u_x(b,t) = 0, & t > 0, \\ u(x,0) = x_0(x), & x \in (a,b). \end{cases}$$

Where f and u_0 are elements of certain functional spaces. The results about existence and unicity as well as determination of blow-up solutions has been demonstrated by semigroup operators methods. To find the numerical solution. To find the numerical solution I used the Finite Difference Method (MDF) with the Explicit Method, that is, we have discretized the second order spatial derivative using centered differences and the temporal derivative using first order derivatives and forward.

The FDM's programming has been developed in two parts: the writing of the code and execution were done in C programming language and, the graphics were visualized in Scilab. Also, the criteria of stability, consistency and convergence have been studied and analyzed, leading to the conclusion that the FDM is conditionally stable, consistent and convergent.

Keywords: Diffusion-reaction problema, blow-up solution, semigroup methods, finite difference method, explicit method.

¹ Universidad César Vallejo - Escuela de Ingeniería Industrial.
E-mail: jbecerras@ucvvirtual.edu.pe

1. INTRODUCCIÓN

Gran parte de los fenómenos estudiados en la física, química, medicina, biología, etc, son modelados mediante ecuaciones en derivadas parciales del tipo difusión - reacción ([4], [6], [9], [13], [14], [17], [22]) de la forma:

$$u_t(x,t) = \alpha u_{xx}(x,t) + f(x,t,u(x,t)), \quad x \in \Omega, t > 0$$

donde Ω es un dominio suave y acotado de \mathbb{R}^n . α es conocido como coeficiente de difusión y f es una función continua sobre $\Omega \times (0,T) \times (0,+\infty)$. La dificultad de esta ecuación yace en encontrar de manera explícita la función u que la satisface, es decir, su solución. Esta dificultad está sujeta a la forma funcional o algebraica que presenta f así como de las condiciones iniciales y de frontera del problema.

En este trabajo se ha estudiado la ecuación de difusión - reacción uno dimensional con condiciones de frontera del tipo Neumann:

$$\begin{cases} u_t(x,t) = u_{xx}(x,t) + f(x,t,u(x,t)), & x \in (a,b), t > 0, \\ u_x(a,t) = u_x(b,t) = 0, & t > 0, \\ u(x,0) = x_0(x), & x \in (a,b). \end{cases} \quad (1)$$

donde $u_0(x) \in L^\infty(a,b)$ y f satisface las siguientes condiciones:

- C1: $f(\cdot, \cdot, u)$ es una función continua sobre $[a,b] \times [0,T]$, para $u > 0$.
- C2: $f(x,t,u) = u(x,t)g(x,t)$

Algunos trabajos recurren al MDF para solucionar ecuaciones de la forma (1) con $f \equiv 0$ ([1], [3], [12], [15], [16], [18], [19], [23], [24]), éstos trabajos demuestran que este método usando el esquema explícito es consistente y condicionalmente estable con constante de estabilidad $r \leq 1/2$. Otros autores ([5], [7], [8], [10], [11], [20], [21]) considerando $f = f(u(x,t))$ han logrado obtener resultados teóricos con respecto a las soluciones explosivas y ciertas estimaciones del tiempo de explosión. Dichos resultados han sido corroborados de forma numérica en [1] y [16].

El objetivo de este trabajo fue la implementación del MDF para resolver numéricamente el problema (1) asumiendo el caso funcional $f(x,t,u(x,t)) = u(x,t)g(x,t)$. Desde luego, se demostró que el método implementado es condicionalmente estable.

2. MATERIAL Y MÉTODOS

2.1. Existencia y unicidad de soluciones del problema (1)

Sea X un espacio de Banach con norma $\|\cdot\|_X$, considerando el problema (1) en su forma abstracta:

$$\begin{cases} \dot{u}(t) + Au(t) = f(t,u(t)), & t > 0 \\ u(0) = u_0 \in X \end{cases} \quad (2)$$

donde $-A$ es el generador de un semigrupo fuertemente continuo $\{T(t)\}_{t \geq 0}$ sobre X con $D(A)$ como su dominio, $f: [0,T] \times X \rightarrow X$ es una función continua en t ($T \leq +\infty$) y de Lipschitz en u . La función $u: [0,T] \rightarrow X$ es una solución clásica de (2) si y sólo si $u \in C^1([0,T]; X) \cap C([0,T]; D(A))$. En cambio, u será una solución débil de (2) si y sólo si $u \in C([0,T]; X)$.

Con el siguiente resultado se garantiza la existencia y unicidad de la solución débil del problema (2):

Teorema 1. Sea $f: [0,T] \times X \rightarrow X$ una función continua en t y globalmente Lipschitz sobre X . Si $-A$ es el generador de un semigrupo fuertemente continuo $\{T(t)\}_{t \geq 0}$, entonces para cada $u_0 \in X$ el problema de valor inicial (2) tiene una única solución débil $u \in C([0,T]; X)$. Además la aplicación $u_0 \rightarrow u$ es Lipschitz de X en $C([0,T]; X)$.

Para la demostración de este teorema consultar [2] ó [10].

El siguiente resultado muestra que si la solución de (2) existe localmente entonces presenta explosión:

Teorema 2. Sea $f: [0,+\infty) \times X \rightarrow X$ una función continua en t , uniformemente acotada en t sobre intervalos acotados y localmente Lipschitz en u sobre X . Si $-A$ es el generador de un semigrupo fuertemente continuo $\{T(t)\}_{t \geq 0}$, entonces para cada $u_0 \in X$ existe $T_{\max} \leq +\infty$ tal que el problema de valor inicial (2) tiene una única solución débil en $[0, T_{\max})$. Si $T_{\max} < +\infty$ entonces:

En [5] se determinó una cota para el tiempo de explosión T_{\max} adicionado que f sea una función impar sobre la tercera coordenada ó $f(x,t,0) = 0$. Por otro lado, el siguiente resultado proporciona las condiciones necesarias para la existencia de una solución clásica de (2): $\lim_{t \rightarrow T_{\max}^-} \|u(t)\|_X = +\infty$

Teorema 3. Si $-A$ es el generador de un semigrupo fuertemente continuo $\{T(t)\}_{t \geq 0}$ sobre X y sea $f: [0,T] \times X \rightarrow X$ una función continuamente diferenciable en t , entonces la solución débil de (2) para $u_0 \in D(A)$ es una solución clásica.

Para las demostraciones de los teoremas 2 y 3 consultar [2].

2.2. Método de Diferencias Finitas (MDF)

Este método numérico consiste en introducir una malla sobre el dominio acotado $[a,b] \times [0,T]$ y aproximar las derivadas de la ecuación del problema (1) por medio de la descomposición de u en su serie de Taylor. Consideramos el caso lineal.

$$\begin{cases} u_t(x,t) = \alpha u_{xx}(x,t), & x \in (a,b), t > 0, \\ u_x(a,t) = u_x(b,t) = 0, & t > 0, \\ u(x,0) = u_0(x), & x \in (a,b). \end{cases} \quad (3)$$

El dominio es discretizado de la siguiente manera: dados M y N se partitionan los conjuntos $[a,b]$ y $[0,T]$ por medio de $x_i=ih$ y $t_j=jk$, donde: $h=\frac{b-a}{M}$ y $k=\frac{T}{N}$ para $i=(1,M)$ y $j=(1,N)$. Las derivadas de u respecto de x y t se expresan en su serie de Taylor:

$$u_{xx}(x,t) = \frac{u(x+h,t) - 2u(x,t) + u(x-h,t)}{h^2} + O(h^4)$$

$$u_t(x,t) = \frac{u(x,t+k) - u(x,t)}{k} + O(k^2)$$

Considerando que los datos tomados son puntuales y además de despreciar los errores ($O(h^4) \approx O(k^2) \approx 0$), se aproxima entonces la solución $u(x,t_j)$ por $u_{i,j}$. Luego, reemplazando en la ecuación de (3) se obtiene la ecuación en su forma discreta:

$$\frac{u_{i,j+1} - u_{i,j}}{k} = \alpha \frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{h^2} \quad (4)$$

Reduciendo, y reemplazando el valor $r = \alpha k/h^2$ se obtiene:

$$u_{i,j+1} = \gamma u_{i,j} + (1-2\gamma)u_{i,j} + \gamma u_{i+1,j}$$

De la misma forma se trabaja para los problemas no lineales de la forma:

$$\begin{cases} u_t(x,t) = \alpha u_{xx}(x,t) + f(x,t), & x \in (a,b), t > 0, \\ u_x(a,t) = u(b,t) = 0, & t > 0, \\ u(x,0) = u_0(x), & x \in (a,b). \end{cases} \quad (5)$$

y

$$\begin{cases} u_t(x,t) = \alpha u_{xx}(x,t) + f(x,t, u(x,t)), & x \in (a,b), t > 0, \\ u_x(a,t) = u(b,t) = 0, & t > 0, \\ u(x,0) = u_0(x), & x \in (a,b). \end{cases} \quad (6)$$

Obteniéndose de éste último la ecuación discretizada como:

$$u_{i,j+1} = \gamma u_{i,j} + (1-2\gamma)u_{i,j} + \gamma u_{i+1,j} + kf(x_p, t_p, u_{i,j})$$

2.3. Estabilidad, consistencia y convergencia del MDF
Diremos que el algoritmo de MDF (ver sección 3.1) es estable si y sólo si los errores en cada iteración no se magnifican conforme el cómputo avanza.

La ecuación en diferencias (4) tiene consistencia si y sólo aproxima a la ecuación en derivadas parciales de (3).

El MDF es convergente si y sólo la solución numérica obtenida se aproxima a la solución real cada vez que k y h tienden a cero.

3. RESULTADOS

Se enuncia el siguiente resultado:

Teorema 5. Dado el problema (1) satisfaciendo las condiciones C1 y C2. Sea $A \geq 0$ tal que: $|g(x,t)| \leq A, \forall (x,t) \in [a,b] \times [0,T]$ entonces el MDF con el esquema explícito es condicionalmente estable si:

$$\frac{k\alpha}{h^2} \leq \frac{2+kA}{4}$$

Además de ser consistente y por el Teorema de consistencia de Lax [19], implica que el MDF es convergente.

Demostración:

Discretizando por diferencias finitas a la ecuación de (1) se tiene:

$$\frac{u_{i,j+1} - u_{i,j}}{k} = \alpha \frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{h^2} + f(x_p, t_p, u_{i,j}) \quad (7)$$

Para usar el criterio de estabilidad de Von Neumann es necesario un cambio de notación: $u_{m,n} = u_{i,j}$ con $m=i, n=j$, entonces la ecuación (7) queda como:

$$\frac{u_{m,n+1} - u_{m,n}}{k} = \alpha \frac{u_{m+1,n} - 2u_{m,n} + u_{m-1,n}}{h^2} + f(x_m, t_n, u_{m,n}) \quad (8)$$

Asumiendo una perturbación de la forma:

$$u_{m,n} = G^m e^{ikh}, \quad i = \sqrt{-1}$$

y reemplazando en (8):

$$G^{m+1} e^{ikh} - G^m e^{ikh} = \frac{k\alpha}{h^2} (G^m e^{ik(n+1)h} - 2G^m e^{ikh} + G^m e^{ik(n-1)h}) + kf(x_m, t_n, G^m e^{ikh})$$

usando C_2 y reduciendo la expresión anterior:

$$G-1 = \frac{k\alpha}{h^2} (e^{-ikh} - 2 + e^{ikh}) + kg(x_m, t_n)$$

$$G-1 = \frac{k\alpha}{h^2} (2\cos(kh) - 2) + kg(x_m, t_n)$$

$$G = 1 + 2r (\cos(kh) - 1) + kg(x_m, t_n)$$

donde $r = \frac{k\alpha}{h^2}$. Luego, la perturbación de la solución no debería crecer, y por tanto, el criterio de estabilidad de Von Neumann expresa que:

$$|G| \leq 1$$

esto es:

$$|1 + 2r (\cos(kh) - 1) + kg(x_m, t_n)| \leq 1$$

$$-1 - kg(x_m, t_n) \leq 1 + 2r (\cos(kh) - 1) \leq 1 - kg(x_m, t_n)$$

usando el hecho que $|g(x,t)| \leq A$:

$$-1 - kA \leq 1 + 2r (\cos(kh) - 1) \leq 1 - kA$$

$$\frac{-2 - kA}{2} \leq r (\cos(kh) - 1) \leq \frac{kA}{2}$$

$$\frac{-2 - kA}{2} \leq -2r \sin^2(kh) \leq \frac{kA}{2}$$

$$\frac{-kA}{4} \leq r \sin^2(kh) \leq 2 + \frac{kA}{4}$$

Como $\sin^2(x) \leq 1$, entonces:

$$r \leq \frac{2 + kA}{4}$$

Para la consistencia del MDF se puede consultar [1], [19] ó [23].

3.1. Codificación del algoritmo en C y SCILAB:

Programa escrito en Lenguaje C:

```
/* Programa para resolver:  $u_t - (a^2)u_{xx} = f(x, t, u(x, t))$ ,
```

```
     $u_x(0, t) = 0$   
     $u_x(L, t) = 0$ ,  
     $u(x, 0) = f(x)$ ,
```

```
*/
```

```
#include<stdio.h>  
#include<math.h>  
#include<conio.h>  
#define pi 3.14159265  
#define e 2.71828182
```

```
float funcion_inicial(float x);  
float funcion_reaccion(float x, float y, float u);  
float L, T, A;
```

```
int main()
```

```
{  
    int i, j, m, n, p;  
    float alfa, h, k, r, t, ejex[50], ejet[2000], ezez[50][2000];  
    char opcion;
```

```
    printf("\n*****  
*****");
```

```
    printf("\n PROGRAMA QUE SOLUCIONA  
    NUMERICAMENTE LA ECUACION DE DIFUSION:
```

```
     $u_t = a^2 u_{xx} + f(x, t, u(x, t))$ ");  
    printf("\n frontera Neumann aplicando metodo explicito  
    ");
```

```
    printf("\n*****  
*****\n");
```

```
    printf("\ningrese [0, L]: L = ");  
    scanf("%f", &L);  
    printf("\nnumero de particiones: M = ");  
    scanf("%d", &m);  
    printf("\ningrese T = ");  
    scanf("%f", &T);  
    printf("\nnumero de particiones: N = ");  
    scanf("%d", &n);  
    printf("\ningrese constante a = ");  
    scanf("%f", &alfa);  
    printf("\ningrese cota A = ");  
    scanf("%f", &A);
```

```
    h = L/m;  
    k = T/n;  
    r = pow(alfa, 2)*k/pow(h, 2);  
    if(r<0.5+A)
```

```
//-----  
    printf("\nEl esquema es estable...\n");
```

```
    for(i=0; i<=m; i++)
```

```
    {  
        ejex[i] = i*h;  
        ezez[i][0] = funcion_inicial(i*h);  
    }
```

```
    for(j=0; j<=n; j++)
```

```
    {  
        ejet[j] = j*k;  
    }
```

```
    for(j=0; j<=n-1; j++)
```

```
    {
```

```
        ezez[0][j+1] = ezez[1][j];
```

```
        for(i=1; i<=m-1; i++)
```

```
        {
```

```
            ezez[i][j+1] = r*ezez[i-1][j] + (1 - 2*r)*ezez[i][j] +  
            r*ezez[i+1][j] + k*funcion_reaccion(ejex[i], ejet[j],  
            ezez[i][j]);
```

```
        }
```

```
        ezez[m][j+1] = ezez[m-1][j+1];
```

```
    }
```

```
    printf("\n APROXIMACION \n");
```

```
    for(j=0; j<=n; j++)
```

```
    {
```

```
        printf("\nx\tt\tu(x, t)");
```

```
        for(i=0; i<=m; i++)
```

```
        {
```

```
            printf("\n%.3f\t%.2f\t%.8f", ejex[i], ejet[j], ezez[i][j]);
```

```
        }
```

```
        printf("\n-----");
```

```
    }
```

```
    printf("\nr = %f", r);
```

```
    //llevar la matriz de datos a Scilab para luego graficarlos
```

```
    FILE *Ejex;
```

```
    Ejex=fopen("C:/Users/USUARIO/Documents/C/Ejex.C
```

```
    SV", "wt");
```

```
    for(i=1; i<=m+1; i++)
```

```
    { fprintf(Ejex, "%f\n", ejex[i-1]);
```

```
    }
```

```
    fclose(Ejex);
```

```
    FILE *Ejet;
```

```
    Ejet=fopen("C:/Users/USUARIO/Documents/C/Ejet.CS
```

```
    V", "wt");
```

```
    for(i=1; i<=n+1; i++)
```

```
    { fprintf(Ejet, "%f\n", ejet[i-1]);
```

```
    }
```

```
    fclose(Ejet);
```

```
    FILE *Ezez;
```

```

Ejez=fopen("C:/Users/USUARIO/Documents/C/Ejez.C
SV","wt");
for(i=1;i<=m+1;i++)
{
for(j=1;j<=n+1;j++)
{
fprintf(Ejez,"%f",ejez[i-1][j-1]);
}
fprintf(Ejez,"\n");
}
fclose(Ejez);
}
else
{
printf("\nEl esquema no es estable ");
}
getch();
return 0;
}
//-----
float funcion_inicial(float x)
{
float valor;
valor = 2*x - pow(x, 2); //esta parte es editable según la
condición inicial
return (valor);
}
//-----
float funcion_reaccion(float x, float y, float u)
{
float valor;
valor = 0; // esta parte es editable según el término
reacción
return (valor);
}
//-----

```

Código en SCILAB para obtener la gráfica de la solución numérica:

```

f=gcf();
f.color_map = hotcolormap(128);
X=read_csv('Ejex.CSV');
x=evstr(X);
x_tam=size(x);
T=read_csv('Ejet.CSV');
t=evstr(T);
t_tam=size(t);
Z=read_csv('Ejez.CSV');
z=evstr(Z);
z_tam=size(z);

```

```

[X, T] = meshgrid(x, t);
surf(t, x, z)

```

3.2. Solución numérica de algunos problemas de difusión reacción:

Ejemplo 1.

$$\begin{cases} u_t(x,t) = u_{xx}(x,t), & x \in (0,1), t > 0, \\ u_x(0,t) = u_x(1,t) = 0, & t > 0, \\ u(x,0) = \text{sen}(\pi x), & x \in (0,1). \end{cases}$$

Considerando $L=1$, $M=10$, $T=1$, $N=200$ y calculando $A=0$, se obtiene la solución numérica:

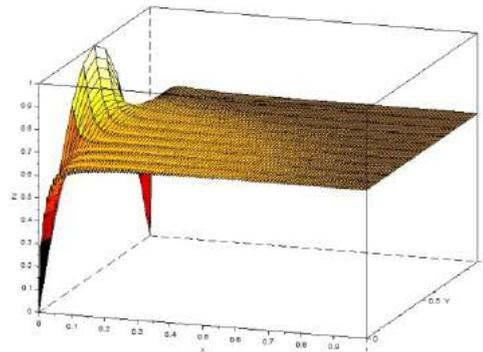


Figura 01.

Ejemplo 2.

$$\begin{cases} u_t(x,t) = u_{xx}(x,t) + ue^{-(x-2)^2}, & x \in (0,2), t > 0, \\ u_x(0,t) = u_x(2,t) = 0, & t > 0, \\ u(x,0) = 2x - x^2, & x \in (0,2). \end{cases}$$

Considerando $L=1$, $M=10$, $T=1$, $N=200$ y calculando $A=1$, se obtiene la solución numérica:

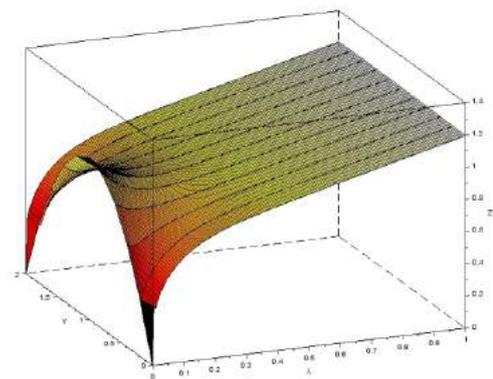


Figura 02.

Ejemplo 3.

$$\begin{cases} u_t(x,t) = u_{xx}(x,t) + \frac{u}{x^2 + t^2}, & x \in (0,4), t > 0, \\ u_x(0,t) = u_x(4,t) = 0, & t > 0, \\ u(x,0) = x^3 - 6x^2 + 9x, & x \in (0,4). \end{cases}$$

Considerando $L=4, M=15, T=1, N=150$ y calculando $A=1$, se obtiene la solución numérica:

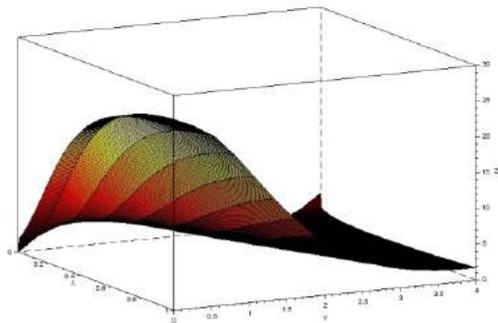


Figura 03.

Ejemplo 4. Solución inestable.

$$\begin{cases} u_t(x,t) = u_{xx}(x,t) + ux^2, & x \in (0,1), t > 0, \\ u_x(0,t) = u_x(1,t) = 0, & t > 0, \\ u(x,0) = e^x, & x \in (0,1). \end{cases}$$

Considerando $L=1, M=10, T=1, N=150$ y calculando $A=1+\infty$, se obtiene la solución numérica inestable:

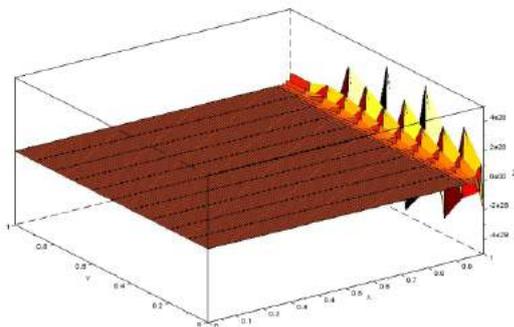


Figura 04.

4. CONCLUSIONES

Se ha llegado a concluir que, en base al Teorema 5, el MDF implementado proporciona una estimación aceptable de la solución numérica del problema (1), siempre que se satisfaga la condición de estabilidad.

5. REFERENCIAS BIBLIOGRÁFICAS

[1] A. Carrillo L., O. Mendoza B. Introducción al método de diferencias finitas y su implementación computacional. Seminario de Internet. 2015.

[2] A. Pazy. Semigroups of linear operators and applications to partial differential equations. Springer, NY. 1983.

[3] C. Bandle, H. Brunner. Numerical analysis of semilinear parabolic problems with blow-up solutions. Revista de la Real Academia de Ciencias Exactas, Físicas y Naturales. 88 – 2/3 (1994). 203 – 222.

[4] J.C. Venegas A., N.S. Landinez P., D.A. Garzón A., Solución computacional de modelos biológicos de formación de patrones espacio-temporales. Revista chilena de ingeniería Inginiare. 17-02 (2009) 182 - 194

[5] J.J.A. Becerra S. Existencia global y explosion de la solución de un problema de diffusion – reacción. Selecciones Matemáticas. 04-01 (2017) 82 – 101.

[6] J Lefevre, J.F. Mangin. A reaction – diffusion model of human brain development. PLOS Computational Biology. 6-4 (2010).

[7] J.M. Arrieta, A. Rodriguez B. Blow-up versus global boundedness of solutions of reaction-diffusion equations with nonlinear boundary conditions, Proceedings of Equations. 11 (2005) 1 – 7.

[8] J.M. Arrieta, A. Rodriguez B. Localization on the boundary of blow – up for reaction – diffusion equations with nonlinear boundary conditions. Communications in Partial Differential Equations. 29-7-8 (2004) 1127 – 1148.

[9] J. Smoller. Shock waves and reaction – diffusion equations. (Springer, NY) 1983.

[10] K. Balázs. Semilinear parabolic problems. Tesis de Maestría. Eotvos Loránd University. (2011).

[11] L. Lorenzi, A. Lanardi, G. Metafuno, D. Palladara. Analytic Semigroups and reaction – diffusion problems. (2004).

[12] L.M. Sánchez, F Ureña, J.J. Benito, L. Gavete. Resolución de la ecuación de diffusion en 2-D y 3-D utilizando diferencias finitas generalizadas. Consistencia y estabilidad. XXI Congreso de Ecuaciones Diferenciales y Aplicaciones - XI Congreso de Matemática Aplicada. 2009.

[13] M. R. Garvie. Finite-difference schemes for reaction-diffusion equations modeling predator-prey interactions in MATLAB. Bulletin of Mathematical Biology 69 (2007) 931- 956.

[14] N.El Khatib, S. Genieys, B. Kazmierczak. Reaction-diffusion model of atherosclerosis development. Journal Mathematical Biology. Publicación online (2011).

- [15] O. J. Mendoza B. Resolución de ecuaciones diferenciales parciales mediante el método de diferencias finitas y su paralelización. Tesis de Licenciatura. UNAM. 2016.
- [16] P. Goodwill Dlamini. Numerical simulation of finite-time blow-up in nonlinear ODEs, reaction-diffusion equations and VIDEs. Disertación. 2012.
- [17] P. Grindrod. The theory and applications of reaction - diffusion equations patterns and waves. (1996).
- [18] R. Bhadauria, A.K. Singh, D.P. Singh. A mathematical model to solve reaction diffusion equation using differential transformation method. International Journal of Mathematics Trends and Technology. 2-2 (2011).
- [19] R. L. Barden, J D. Faires. Análisis Numérico. Thompson Learnig. 9^{ma} edición. 2011.
- [20] T. Cazenave, A. Heraux. An Introduction to Semilinear Evolution Equations. 1996.
- [21] V.A. Galaktionov, J.L. Vázquez. The problem of blow - up in nonlinear parabolic equations. Discrete and Continuous Dynamical Systems. 8-2 (2002) 399 - 433.
- [22] V. Volpert, S. Petrovski. Reaction - diffusion waves in biology. Physics of Life Reviews. 6 (2009) 267 - 310.
- [23] X. Zhang. Numerical Methods for PDEs. Lectures Notes. Springer. 2017.
- [24] Z. Millán, L. De la Torre, L. Oliva, M. Berenguer. Simulación Numérica. Ecuación de difusión. Revista Iberoamericana de Ingeniería Mecánica. 15 - 2 (2011) 29 - 38.

