

Solución Numérica de la Ecuación de Calor por el Método de las Diferencias Finitas

Miguel Antonio Caro Candezano¹, Boris Lora Castro¹, Valdemir Garcia Ferreira²

¹Departamento de Matemáticas de la Universidad del Atlántico, Colombia

²Departamento de Matemática Aplicada e Estatística, ICMC - USP

Julio de 2008

Abstract

In this report, we present the numerical solution of the heat equation by using the finite difference method. Particular, we employ an explicit and implicit schemes with different values of the stability parameter. The numerical results are compared with analytical solutions.

Keywords: Ecuación de calor, método de diferencias finitas, método explícito, método implícito, soluciones numéricas.

1 Introducción

Existen diferentes métodos para resolver Ecuaciones Diferenciales Parciales (EDP) y Ecuaciones Diferenciales Ordinarias (EDO), entre ellos se encuentra el Método de las Diferencias Finitas (MDF), el cual consiste en introducir una malla ¹ sobre una región Ω y aproximar las derivadas del problema planteado por medio de técnicas de aproximación de las derivadas como lo es, por ejemplo, la descomposición en serie de Taylor de la función. Ahora tomemos la descomposición en serie de Taylor de la función U en los puntos $x + h$ y $x - h$, después sumamos como es habitual en estos casos para hallar una expresión tanto de la primera como de la segunda derivada de la función, entre otras, así tenemos lo siguiente

$$U(x + h) \approx U(x) + hU'(x) + \frac{h^2}{2!} U''(x) + \frac{h^3}{3!} U'''(x) + \dots \quad (1)$$

$$U(x - h) \approx U(x) - hU'(x) + \frac{h^2}{2!} U''(x) - \frac{h^3}{3!} U'''(x) + \dots \quad (2)$$

$$U(x + h) + U(x - h) = 2U(x) + h^2 U''(x) + O(h^4), \quad (3)$$

¹Sea $U(x, t)$ una función de x y de t , subdividiremos el plano $x - t$ en un conjunto de rectángulos iguales de lados δx y δt respectivamente, de tal manera que cada punto de los ejes OX y OY están definidos respectivamente por $x_i = ih, h = \delta x, i = \dots - 1, 0, 1, \dots$, $y_j = jk, k = \delta y, j = 0, 1, \dots$, así cada punto de la grilla tendrá coordenadas $(x_i, y_j) = (ih, jk)$. Ver Fig.1.

En (3) el error es de al menos h^4 .

De lo anterior podemos deducir que al sumar (1) y (2) tenemos

$$U''(x) = \frac{1}{h^2}[U(x+h) - 2U(x) + U(x-h)] \quad (4)$$

con un error de h^2 .

Ahora de (2) restemos (3), obtenemos la denominada **diferencia central**

$$U'(x) = \frac{1}{2h}[U(x+h) - U(x-h)] \quad (5)$$

también con un error de h^2 .

Definimos la “**diferencia para adelante**” por

$$U'(x) = \frac{1}{h}[U(x+h) - U(x)] \quad (6)$$

y la **diferencia “diferencia para atrás”** por

$$U'(x) = \frac{1}{h}[U(x) - U(x-h)] \quad (7)$$

A continuación veamos la aplicación del MDF en una EDO con un problema de Cauchy:

$$\begin{cases} u''(x) + \mu u(x) &= f(x), & a \leq x \leq b, & \mu = cte \\ u(a) = \alpha, & u(b) &= \beta \end{cases} \quad (8)$$

Introducimos una malla continua en el segmento $[a, b]$, $a = x_0, x_1, \dots, x_N = b$, representemos la segunda derivada de $u(x)$, aproximadamente por (3), teniendo en cuenta que $u_n = u(x_n)$:

$$u''(x_n) \approx \frac{u_{n-1} - 2u_n + u_{n+1}}{h^2}, \quad h = x_{n+1} - x_n \quad (9)$$

esta aproximación se puede utilizar en cada nodo $x_n, 1 \leq n \leq N-1$, haciendo la sustitución en (8) y representado $f(x_n) = f_n$, tenemos

$$\frac{u_{n-1} - 2u_n + u_{n+1}}{h^2} + \mu u_n = f_n, \quad 1 \leq n \leq N-1 \quad (10)$$

esto es un sistema de $N-1$ ecuaciones algebraicas cuyas incógnitas son las soluciones aproximadas en cada retículo de la malla, el número de soluciones $u_n, 0 \leq n \leq N$ es igual a $N+1$, es decir, una más que el número de ecuaciones en el sistema (8), faltarían dos ecuaciones que pueden ser suplidas por las condiciones en los extremos

$$u_0 = u(x_0) = a, \quad (11)$$

$$u_N = u(x_N) = b \quad (12)$$

Con el sistema de ecuaciones completo, podemos resolverlo por diferentes métodos numéricos (Gauss-sin Pivoteo, por ejemplo).

2 Formulación del problema

Un ejemplo clásico de una EDP del tipo parabólico es la ecuación de la distribución de calor (Difusión) en una barra de longitud L con las particularidades siguientes:

- La barra esta aislada totalmente en sus extremos.
- La distribución de la temperatura es uniforme en las secciones rectas.
- El material que compone a la barra es homogéneo.

La ecuación de distribución de calor tiene la siguiente forma

$$\frac{\partial U}{\partial t} = \mu \frac{\partial^2 U}{\partial x^2}, \quad 0 < x < L, \quad t > 0, \quad \mu = \text{cte.} \quad (13)$$

donde la solución a esta ecuación es la temperatura U a una distancia X del extremo de la barra delgada después de un tiempo T , la solución es una función que depende de dos variables $U(x, t)$, μ es el coeficiente de distribución de calor.

3 El Método Explicito

El Método Explicito consiste en calcular los valores desconocidos de u en un $t = k$ cualquiera a partir de los valores conocidos de u para valores anteriores de t , los valores de u son conocidos en $t = 0, j = 0$ gracias a las condiciones iniciales. Veamos como queda (13), cuando $\mu = 1$ utilizando MDF, para $j = 0$:

$$\frac{u_{i,j+1} - u_{i,j}}{k} = \frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{h^2} \quad (14)$$

teniendo en cuenta que $x_i = ih, i = 0, 1, 2, \dots, y_j = jk, j = 0, 1, 2, \dots$ al operar (14) tenemos

$$u_{i,j+1} = \frac{k}{h^2}(u_{i+1,j} - 2u_{i,j} + u_{i-1,j}) + u_{i,j} \quad (15)$$

Sea $r = \frac{k}{h^2}$, entonces (15) toma la forma

$$u_{i,j+1} = ru_{i-1,j} + (1 - 2r)u_{i,j} + ru_{i+1,j} \quad (16)$$

La ecuación (16) se denomina **fórmula explícita** o **modelo explícito** para la solución de una EDP usando MDF. El Método explícito es convergente siempre que $0 < r \leq \frac{1}{2}$ o que $k \leq \frac{h^2}{2}$. [3]

En la siguiente página encontramos un modelo gráfico del Método Explícito e Implícito en forma de celulas.

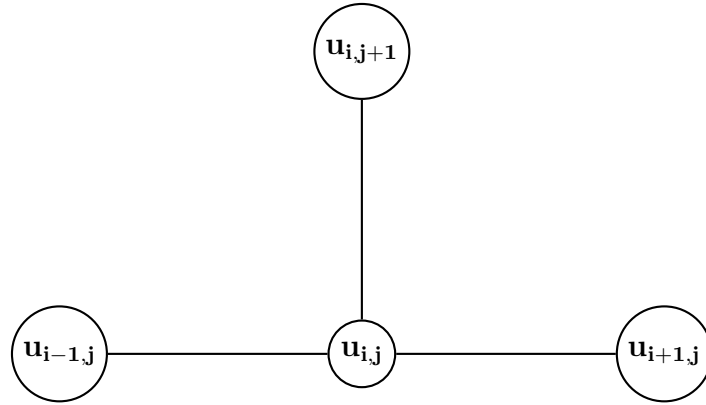


Figure 1: Célula reticular para el Método Explícito

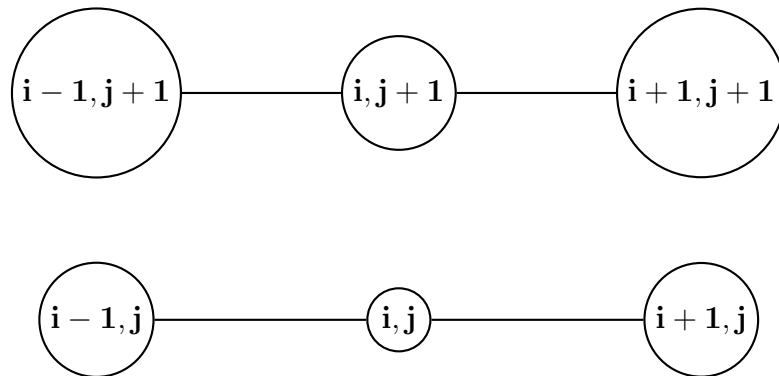


Figure 2: Célula reticular para el Método Implícito

Resolveremos una EDP parabólica utilizando para ello el denominado **método explícito** para diferentes valores de r como se indican en las respectivas tablas de datos. Los resultados numéricos fueron obtenidos a partir de programas hecho en el lenguaje de programación **C++**, en los anexos se encuentran los códigos de los programas.

El problema es el siguiente:

Problema 1.

$$\frac{\partial U}{\partial t} = \frac{\partial^2 U}{\partial x^2}, \quad 0 < x < 1, \quad t > 0, \quad (17)$$

$$U(x, t) = 0, \quad \text{si } x = 0 \text{ y } x = 1, \quad \forall t > 0 \quad (18)$$

$$U(x, 0) = \begin{cases} 2x, & 0 \leq x \leq 1/2 \\ 2(1 - x), & 1/2 \leq x \leq 1 \end{cases} \quad (19)$$

donde (18) son las condiciones en la frontera, (19) son las denominadas condiciones iniciales. La solución analítica para el problema descrito anteriormente es

$$U(x, t) = \frac{8}{\pi^2} \sum_{n=1}^{\infty} \frac{1}{n^2} \left(\sin \frac{n\pi}{2} \right) (\sin n\pi x) \exp(-n^2 \pi^2 t) \quad (20)$$

A continuación, en la siguiente página, mostramos las soluciones numéricas del Problema 1. para diferentes valores de r ². El **error** lo definimos por la siguiente fórmula [4]

$$error = \frac{|U(x_i, y_j) - U(x, y)|}{U(x, y)} \times 100. \quad (21)$$

La **Diferencia** se define por la resta entre el valor numérico de la función $U(x_i, y_j)$ obtenido por el MDF y el valor analítico de la misma $U(x, y)$.

²Los resultados numéricos de los problemas analizados y presentados en este trabajo son para los siguientes casos: $r = 0.48, r = 0.50, r = 0.30, r = 0.70$. En las Tablas **Dif** significa **Diferencia**.

Tabla 1: Resultados Numéricos del Problema 1. aplicando el Método Explícito.

Caso 1. $1.r = 0.30, h = 0.1, k = 0.0030$							
i	j	x	t	$u(x_i, t_j)$	$U(x, t)$	Dif	Error
1	2	0.100	0.006	0.2000	0.2000	0.0000	0.0069
2	2	0.200	0.006	0.4000	0.3996	0.0004	0.1024
3	2	0.300	0.006	0.6000	0.5941	0.0059	0.9865
4	2	0.400	0.006	0.7640	0.7570	0.0070	0.9218
5	2	0.500	0.006	0.8320	0.8252	0.0068	0.8250
.
.
1	10	0.100	0.030	0.1829	0.1812	0.0016	0.9081
2	10	0.200	0.030	0.3512	0.3484	0.0028	0.8046
3	10	0.300	0.030	0.4890	0.4857	0.0033	0.6724
4	10	0.400	0.030	0.5803	0.5770	0.0032	0.5621
5	10	0.500	0.030	0.6123	0.6091	0.0032	0.5193
.
.
1	30	0.100	0.090	0.1033	0.1030	0.0002	0.2409
2	30	0.200	0.090	0.1964	0.1960	0.0005	0.2376
3	30	0.300	0.090	0.2704	0.2698	0.0006	0.2334
4	30	0.400	0.090	0.3179	0.3171	0.0007	0.2301
5	30	0.500	0.090	0.3342	0.3335	0.0008	0.2288
.
.
1	38	0.100	0.114	0.0814	0.0813	0.0001	0.0758
2	38	0.200	0.114	0.1548	0.1547	0.0001	0.0752
3	38	0.300	0.114	0.2130	0.2129	0.0002	0.0744
4	38	0.400	0.114	0.2504	0.2502	0.0002	0.0738
5	38	0.500	0.114	0.2633	0.2631	0.0002	0.0736

Caso 3. $r = 0.50, h = 0.1, k = 0.0050$							
i	j	x	t	$u(x_i, t_j)$	$U(x, t)$	Dif	Error
1	1	0.100	0.005	0.2000	0.2000	0.0000	0.0014
2	1	0.200	0.005	0.4000	0.3998	0.0002	0.0382
3	1	0.300	0.005	0.6000	0.5966	0.0034	0.5693
4	1	0.400	0.005	0.8000	0.7667	0.0333	4.3469
5	1	0.500	0.005	0.8000	0.8404	0.0404	4.8099
.
.
1	6	0.100	0.030	0.1875	0.1812	0.0063	3.4570
2	6	0.200	0.030	0.3438	0.3484	0.0046	1.3281
3	6	0.300	0.030	0.5000	0.4857	0.0143	2.9338
4	6	0.400	0.030	0.5625	0.5770	0.0145	2.5161
5	6	0.500	0.030	0.6250	0.6091	0.0159	2.6058
.
.
1	18	0.100	0.090	0.10490	0.1030	0.0019	1.8338
2	18	0.200	0.090	0.18978	0.1960	0.0062	3.1551
3	18	0.300	0.090	0.27466	0.2698	0.0049	1.8188
4	18	0.400	0.090	0.30708	0.3171	0.0101	3.1715
5	18	0.500	0.090	0.33951	0.3335	0.0060	1.8096
.
.
1	24	0.100	0.120	0.07764	0.0766	0.0010	1.3101
2	24	0.200	0.120	0.14044	0.1458	0.0053	3.6490
3	24	0.300	0.120	0.20325	0.2006	0.0026	1.3085
4	24	0.400	0.120	0.22724	0.2359	0.0086	3.6506
5	24	0.500	0.120	0.25123	0.2480	0.0032	1.3075

Caso 1. $2.r = 0.48, h = 0.1, k = 0.0048$							
i	j	x	t	$u(x_i, t_j)$	$U(x, t)$	Dif	Error
1	1	0.100	0.005	0.2000	0.2000	0.0000	0.0010
2	1	0.200	0.005	0.4000	0.3999	0.0001	0.0300
3	1	0.300	0.005	0.6000	0.5970	0.0030	0.4987
4	1	0.400	0.005	0.8000	0.7686	0.0314	4.0841
5	1	0.500	0.005	0.8080	0.8436	0.0356	4.2254
.
.
1	5	0.100	0.024	0.1898	0.1891	0.0007	0.3766
2	5	0.200	0.024	0.3754	0.3658	0.0096	2.6170
3	5	0.300	0.024	0.5093	0.5141	0.0048	0.9356
4	5	0.400	0.024	0.6261	0.6146	0.0115	1.8742
5	5	0.500	0.024	0.6389	0.6504	0.0115	1.7648
.
.
1	20	0.100	0.096	0.0969	0.0971	0.0002	0.2344
2	20	0.200	0.096	0.1827	0.1847	0.0020	1.1016
3	20	0.300	0.096	0.2536	0.2542	0.0006	0.2441
4	20	0.400	0.096	0.2956	0.2989	0.0033	1.1113
5	20	0.500	0.096	0.3135	0.3143	0.0008	0.2501
.
.
1	25	0.100	0.120	0.0756	0.0766	0.0010	1.3250
2	25	0.200	0.120	0.1446	0.1458	0.0011	0.7712
3	25	0.300	0.120	0.1980	0.2006	0.0027	1.3266
4	25	0.400	0.120	0.2340	0.2359	0.0018	0.7728
5	25	0.500	0.120	0.2447	0.2480	0.0033	1.3276

Caso 4. $r = 0.70, h = 0.1, k = 0.0070$							
i	j	x	t	$u(x_i, t_j)$	$U(x, t)$	Dif	Error
1	1	0.100	0.007	0.2000	0.2000	0.0000	0.022
2	1	0.200	0.007	0.4000	0.3992	0.0008	0.212
3	1	0.300	0.007	0.6000	0.5911	0.0089	1.491
4	1	0.400	0.007	0.8000	0.7475	0.0525	7.023
5	1	0.500	0.007	0.7200	0.8112	0.0912	11.24
.
.
1	5	0.100	0.035	0.1328	0.1741	0.0413	23.73
2	5	0.200	0.035	0.4576	0.3335	0.1242	37.24
3	5	0.300	0.035	0.2268	0.4630	0.2362	51.01
4	5	0.400	0.035	0.8898	0.5481	0.3417	62.34
5	5	0.500	0.035	0.1860	0.5778	0.3918	67.80
.
.
1	8	0.100	0.056	0.5529	0.1436	0.4093	285.00
2	8	0.200	0.056	-0.582	0.2735	0.8551	312.58
3	8	0.300	0.056	1.6757	0.3771	1.2985	344.31
4	8	0.400	0.056	-1.209	0.4439	1.6526	372.26
5	8	0.500	0.056	2.2456	0.4670	1.7786	380.82
.
.
1	11	0.100	0.077	-2.355	0.1171	2.4718	2111.4
2	11	0.200	0.077	5.0544	0.2227	4.8317	2169.3
3	11	0.300	0.077	-6.582	0.3067	6.8883	2246.2
4	11	0.400	0.077	8.6683	0.3606	8.3077	2303.9
5	11	0.500	0.077	-8.454	0.3793	8.8329	2329.4

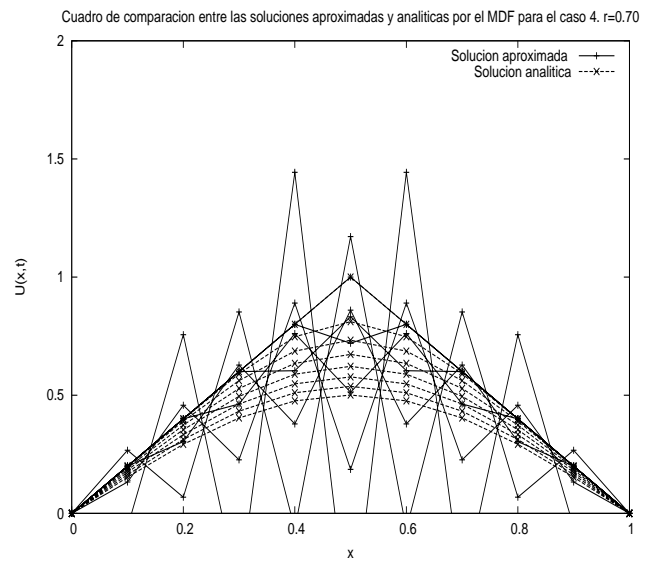
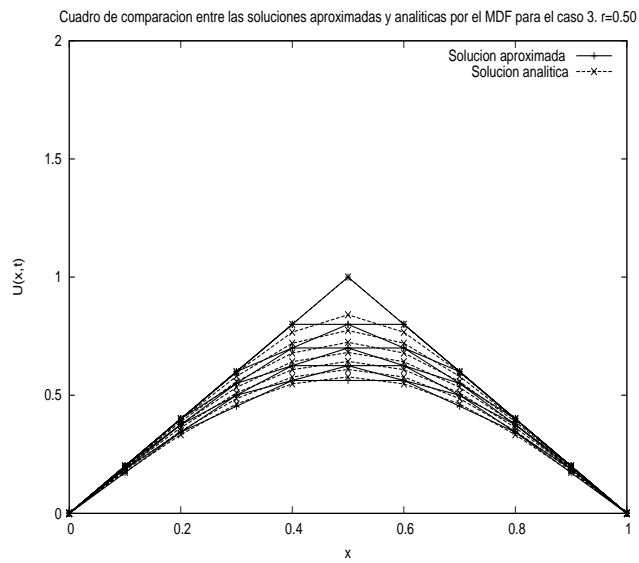
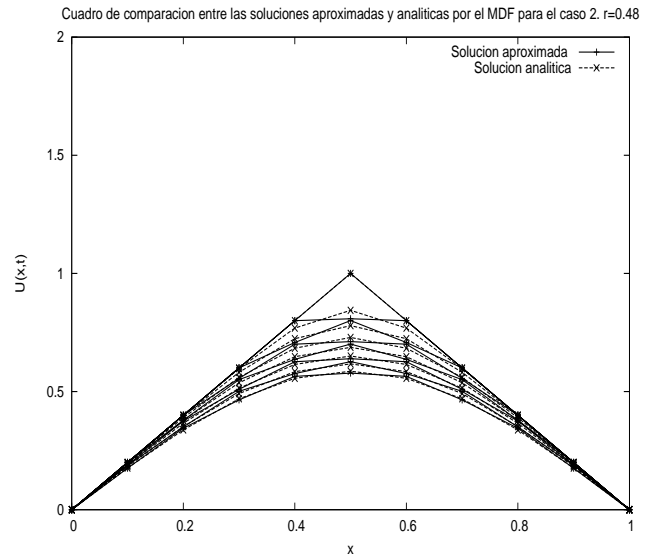
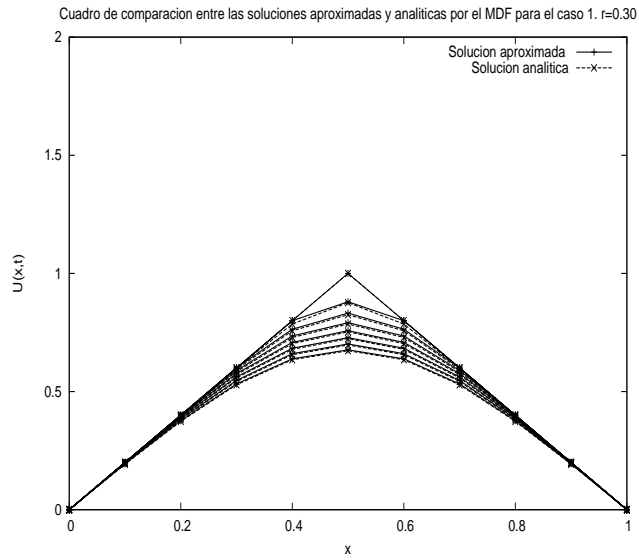


Figure 3: Tablas de comparación entre las soluciones analíticas y aproximadas por el Método Explícito

Ahora analicemos un problema similar al Problema 1. pero empleando derivadas en las condiciones de frontera

Problema 2.

$$\frac{\partial U}{\partial t} = \frac{\partial^2 U}{\partial x^2}, \quad 0 < x < 1, \quad t > 0, \quad (22)$$

$$U(x, 0) = 1, \quad 0 \leq x \leq 1 \quad (23)$$

$$\begin{cases} \frac{\partial U}{\partial x} = U, & x = 0, \forall t \\ \frac{\partial U}{\partial x} = -U, & x = 1, \forall t \end{cases} \quad (24)$$

Usemos el MDF en el problema y expresemos las condiciones de frontera mediante las **diferencias centrales**, de tal manera que (22) queda así

$$u_{i,j+1} = u_{i,j} + r(u_{i-1,j} - 2u_{i,j} + u_{i+1,j}). \quad (25)$$

Aplicamos (25) para $x = 0$, cuando $i = 0$

$$u_{0,j+1} = u_{0,j} + r(u_{-1,j} - 2u_{0,j} + u_{1,j}). \quad (26)$$

Para la condición de frontera en $x = 0$, en términos del MDF es

$$\frac{u_{1,j} - u_{-1,j}}{2h} = u_{0,j}. \quad (27)$$

De (25) y (26) eliminamos $u_{-1,j}$, y la expresión en este caso es

$$u_{0,j+1} = u_{0,j} + 2r(u_{1,j} - (1+h)u_{0,j}). \quad (28)$$

Actuamos de la misma manera con la otra condición del valor de frontera en $x = 1$, cuando $i = 10$

$$u_{10,j+1} = u_{10,j} + r(u_{9,j} - 2u_{10,j} + u_{11,j}), \quad (29)$$

también eliminamos el valor $u_{11,j}$, el cual no existe según el planteamiento de nuestro problema, por eso expresemos (24) usando MDF

$$\frac{u_{11,j} - u_{9,j}}{2h} = -u_{10,j}, \quad (30)$$

así tenemos que al excluir $u_{11,j}$ de (30), tenemos

$$u_{10,j+1} = u_{10,j} + 2r(u_{9,j} - (1+h)u_{10,j}). \quad (31)$$

Podemos resumir la expresión en diferencias del Problema 2. en estas tres expresiones

$$\begin{cases} u_{i,j+1} = u_{i,j} + r(u_{i-1,j} - 2u_{i,j} + u_{i+1,j}) \\ u_{0,j+1} = u_{0,j} + 2r(u_{1,j} - (1+h)u_{0,j}) \\ u_{10,j+1} = u_{10,j} + 2r(u_{9,j} - (1+h)u_{10,j}) \end{cases} \quad (32)$$

La solución analítica del Problema 2. es

$$U(x, t) = 4 \sum_{n=1}^{\infty} \left\{ \frac{\sec \alpha_n}{(3 + 4\alpha_n^2)} \exp(-4\alpha_n^2 t) \cos(2\alpha_n(x - \frac{1}{2})) \right\} \quad (33)$$

α_n son las raíces positivas de

$$\alpha \tan \alpha = \frac{1}{2} \quad (34)$$

Hay que tener presente que tanto en el Problema 1. como en el Problema 2., con respecto a $x = \frac{1}{2}$, existe una simetría, la cual facilita, en este caso en particular, el cálculo numérico.

Se ha demostrado que el esquema utilizado para el Problema 2. es convergente [3] siempre que $r \leq \frac{1}{2+h\delta x}$.

En la siguiente página mostramos las Tablas que muestran los resultados numéricos para el Problema 2.

Tabla 2: Resultados Numéricos del Problema 2. aplicando el Método Explicito.

Caso 1. $r = 0.30, h = 0.1, k = 0.0030$							
i	j	x	t	$u(x_i, t_j)$	$U(x, t)$	Dif	Error
0	1	0.0000	0.0030	0.9400	0.9411	0.0011	0.11
1	1	0.1000	0.0030	1.0000	0.9930	0.0070	0.70
2	1	0.2000	0.0030	1.0000	0.9998	0.0002	0.02
3	1	0.3000	0.0030	1.0000	1.0000	0.0000	0.00
4	1	0.4000	0.0030	1.0000	1.0000	0.0000	0.00
5	1	0.5000	0.0030	1.0000	1.0000	0.0000	0.00
.
.
.
0	10	0.0000	0.0300	0.8308	0.8311	0.0002	0.03
1	10	0.1000	0.0300	0.9021	0.9018	0.0003	0.03
2	10	0.2000	0.0300	0.9496	0.9488	0.0008	0.08
3	10	0.3000	0.0300	0.9772	0.9761	0.0011	0.12
4	10	0.4000	0.0300	0.9906	0.9893	0.0013	0.13
5	10	0.5000	0.0300	0.9944	0.9931	0.0013	0.13
.
.
.
0	34	0.0000	0.1020	0.7148	0.7150	0.0002	0.03
1	34	0.1000	0.1020	0.7800	0.7800	0.0000	0.00
2	34	0.2000	0.1020	0.8316	0.8313	0.0002	0.03
3	34	0.3000	0.1020	0.8688	0.8683	0.0004	0.05
4	34	0.4000	0.1020	0.8912	0.8906	0.0006	0.06
5	34	0.5000	0.1020	0.8987	0.8981	0.0006	0.07
.
.
.
0	68	0.0000	0.2040	0.5995	0.5999	0.0004	0.07
1	68	0.1000	0.2040	0.6543	0.6546	0.0003	0.05
2	68	0.2000	0.2040	0.6979	0.6981	0.0002	0.03
3	68	0.3000	0.2040	0.7297	0.7298	0.0001	0.02
4	68	0.4000	0.2040	0.7489	0.7490	0.0001	0.01
5	68	0.5000	0.2040	0.7554	0.7554	0.0001	0.01

Caso 2. $r = 0.48, h = 0.1, k = 0.0048$							
i	j	x	t	$u(x_i, t_j)$	$U(x, t)$	Dif	Error
0	1	0.0000	0.0048	0.9040	0.9264	0.0224	2.42
1	1	0.1000	0.0048	1.0000	0.9850	0.0150	1.53
2	1	0.2000	0.0048	1.0000	0.9986	0.0014	0.14
3	1	0.3000	0.0048	1.0000	0.9999	0.0001	0.01
4	1	0.4000	0.0048	1.0000	1.0000	0.0000	0.00
5	1	0.5000	0.0048	1.0000	1.0000	0.0000	0.00
.
.
.
0	5	0.0000	0.0240	0.8372	0.8467	0.0095	1.12
1	5	0.1000	0.0240	0.9227	0.9172	0.0055	0.60
2	5	0.2000	0.0240	0.9569	0.9610	0.0041	0.43
3	5	0.3000	0.0240	0.9887	0.9841	0.0046	0.46
4	5	0.4000	0.0240	0.9949	0.9942	0.0007	0.07
5	5	0.5000	0.0240	1.0000	0.9968	0.0032	0.32
.
.
.
0	21	0.0000	0.1008	0.7144	0.7165	0.0021	0.30
1	21	0.1000	0.1008	0.7828	0.7817	0.0011	0.14
2	21	0.2000	0.1008	0.8317	0.8331	0.0014	0.17
3	21	0.3000	0.1008	0.8716	0.8701	0.0015	0.17
4	21	0.4000	0.1008	0.8917	0.8924	0.0007	0.08
5	21	0.5000	0.1008	0.9016	0.8999	0.0017	0.19
.
.
.
0	42	0.0000	0.2016	0.6020	0.6024	0.0004	0.06
1	42	0.1000	0.2016	0.6563	0.6573	0.0010	0.15
2	42	0.2000	0.2016	0.7007	0.7010	0.0003	0.04
3	42	0.3000	0.2016	0.7320	0.7328	0.0008	0.11
4	42	0.4000	0.2016	0.7519	0.7521	0.0002	0.03
5	42	0.5000	0.2016	0.7578	0.7585	0.0007	0.10

Caso 3. $r = 0.50, h = 0.1, k = 0.0050$							
i	j	x	t	$u(x_i, t_j)$	$U(x, t)$	Dif	Error
0	1	0.0000	0.0050	0.9000	0.9250	0.0250	2.70
1	1	0.1000	0.0050	1.0000	0.9841	0.0159	1.62
2	1	0.2000	0.0050	1.0000	0.9984	0.0016	0.16
3	1	0.3000	0.0050	1.0000	0.9999	0.0001	0.01
4	1	0.4000	0.0050	1.0000	1.0000	0.0000	0.00
5	1	0.5000	0.0050	1.0000	1.0000	0.0000	0.00
.
.
.
0	4	0.0000	0.0200	0.8691	0.8585	0.0106	1.24
1	4	0.1000	0.0200	0.9170	0.9286	0.0116	1.24
2	4	0.2000	0.0200	0.9775	0.9695	0.0080	0.83
3	4	0.3000	0.0200	0.9875	0.9891	0.0016	0.16
4	4	0.4000	0.0200	1.0000	0.9967	0.0033	0.33
5	4	0.5000	0.0200	1.0000	0.9985	0.0015	0.15
.
.
.
0	20	0.0000	0.1000	0.7260	0.7176	0.0085	1.18
1	20	0.1000	0.1000	0.7741	0.7828	0.0086	1.10
2	20	0.2000	0.1000	0.8416	0.8342	0.0074	0.89
3	20	0.3000	0.1000	0.8646	0.8713	0.0067	0.77
4	20	0.4000	0.1000	0.9007	0.8936	0.0071	0.79
5	20	0.5000	0.1000	0.8952	0.9011	0.0059	0.65
.
.
.
0	41	0.0000	0.2050	0.5868	0.5989	0.0121	2.02
1	41	0.1000	0.2050	0.6631	0.6535	0.0096	1.47
2	41	0.2000	0.2050	0.6868	0.6970	0.0102	1.46
3	41	0.3000	0.2050	0.7370	0.7285	0.0085	1.16
4	41	0.4000	0.2050	0.7385	0.7477	0.0092	1.24
5	41	0.5000	0.2050	0.7622	0.7541	0.0081	1.07

Caso 4. $r = 0.70, h = 0.1, k = 0.0070$							
i	j	x	t	$u(x_i, t_j)$	$U(x, t)$	Dif	Error
0	1	0.0000	0.0070	0.8600	0.9122	0.0522	5.72
1	1	0.1000	0.0070	1.0000	0.9752	0.0248	2.55
2	1	0.2000	0.0070	1.0000	0.9958	0.0042	0.43
3	1	0.3000	0.0070	1.0000	0.9996	0.0004	0.04
4	1	0.4000	0.0070	1.0000	1.0000	0.0000	0.00
5	1	0.5000	0.0070	1.0000	1.0000	0.0000	0.00
.
.
.
0	3	0.0000	0.0210	0.7576	0.8554	0.0978	11.44
1	3	0.1000	0.0210	0.9941	0.9256	0.0685	7.40
2	3	0.2000	0.0210	0.9314	0.9673	0.0359	3.71
3	3	0.3000	0.0210	1.0000	0.9879	0.0121	1.22
4	3	0.4000	0.0210	1.0000	0.9961	0.0039	0.39
5	3	0.5000	0.0210	1.0000	0.9981	0.0019	0.19
.
.
.
0	15	0.0000	0.1050	-62.0875	0.7113	62.7987	8829.33
1	15	0.1000	0.1050	56.4138	0.7759	55.6379	7170.53
2	15	0.2000	0.1050	-46.6049	0.8270	47.4319	5735.09
3	15	0.3000	0.1050	40.5358	0.8639	39.6718	4592.03
4	15	0.4000	0.1050	-33.2094	0.8862	34.0956	3847.51
5	15	0.5000	0.1050	32.9599	0.8936	32.0663	3588.42
.
.
.
0	29	0.0000	0.2030	-234318	0.6009	234319	38992788
1	29	0.1000	0.2030	212842	0.6557	212841	32458883
2	29	0.2000	0.2030	-195210	0.6993	195211	27913594
3	29	0.3000	0.2030	181872	0.7310	181871	24878652
4	29	0.4000	0.2030	-173483	0.7503	173484	23123068
5	29	0.5000	0.2030	170616	0.7567	170615	22546964

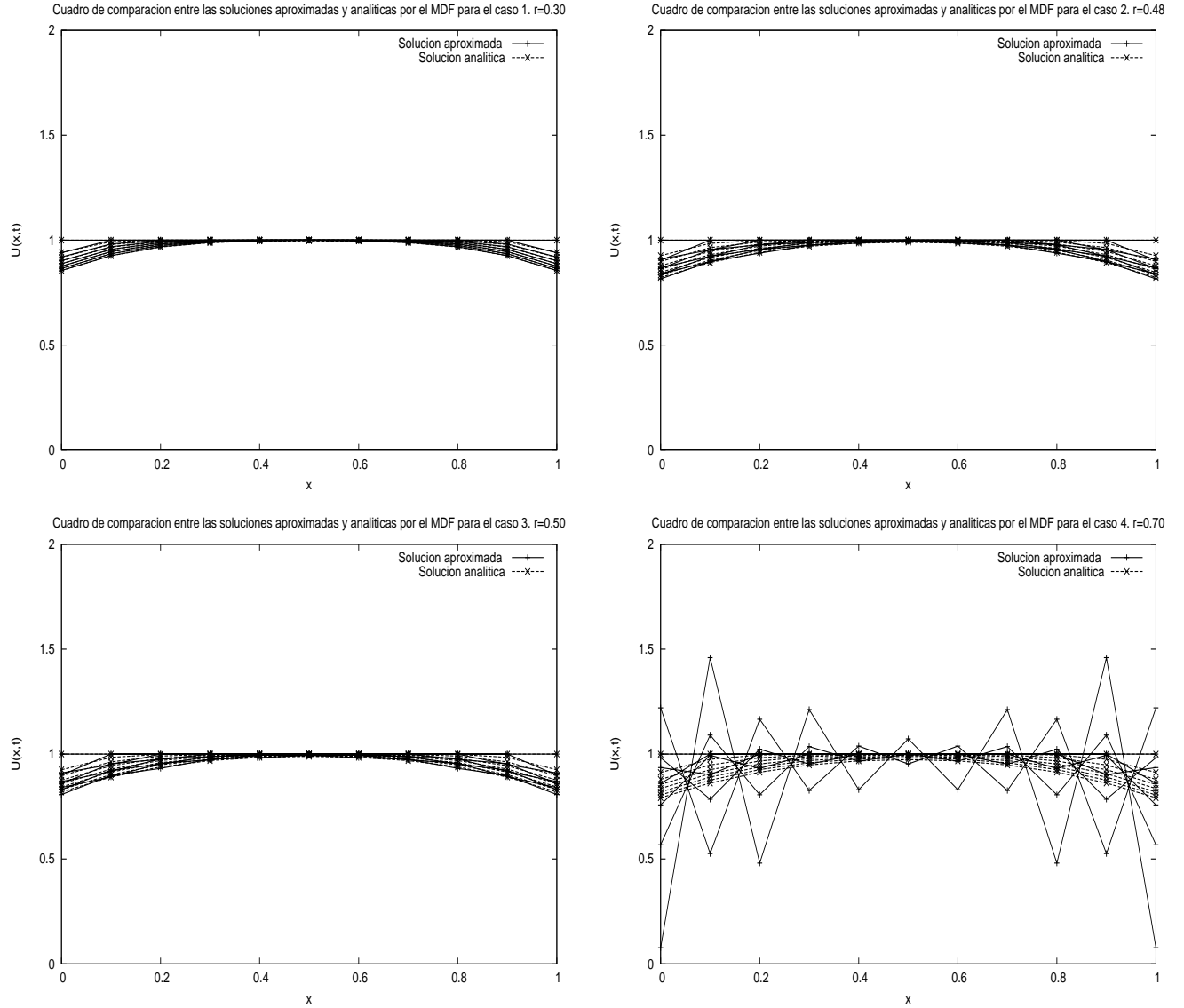


Figure 4: Cuadros de comparación entre las soluciones analíticas y aproximadas por el Método Explícito para el Problema 2.

En la tabla anterior observamos que para el **caso** $r= 0.70$, el proceso es divergente.

4 El Método Implícito Crank-Nicolson

Este método consiste en analizar la EDP en el punto $(ih, (j + \frac{1}{2})k)$ y aproximar la derivada con respecto al tiempo por la **diferencia central** y la derivada con respecto a x por la media de las diferencias centrales en los niveles de tiempo $j, j+1$. Según este método el problema (13) se expresa en MDF así:

$$\frac{u_{i,j+1} - u_{i,j}}{k} = \frac{1}{2} \left(\frac{u_{i+1,j+1} - 2u_{i,j+1} + u_{i-1,j+1}}{h^2} + \frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{h^2} \right). \quad (35)$$

Al operar (35) tenemos

$$-ru_{i-1,j+1} + (2 + 2r)u_{i,j+1} - ru_{i+1,j+1} = ru_{i-1,j} + (2 - 2r)u_{i,j} + ru_{i+1,j}, \quad r = \frac{k}{h^2} \quad (36)$$

Del lado izquierdo de (36) hay tres valores desconocidos y del lado derecho tres conocidos. Para cada valor de $t > 0$, o $j > 1$, se forman N ecuaciones simultaneas con N incógnitas, este sistema puede ser resuelto , por ejemplo, por el Método de Gauss sin pivoteo.(Dibujar la célula)

Ahora resolvemos el Problema 1. por el método **Crank-Nicolson**; en la siguiente página mostramos los resultados numéricos para los distintos casos de r determinados en las anteriores tablas.

Tabla 3: Resultados Numéricos del Problema 1. aplicando el Método Implícito.

Caso 1. $r = 0.30, h = 0.1, k = 0.0030$							
i	j	x	t	$u(x_i, t_j)$	$U(x, t)$	Dif	Error
1	1	0.1000	0.0030	0.1995	0.2000	0.0005	0.2354
2	1	0.2000	0.0030	0.3981	0.4000	0.0019	0.4698
3	1	0.3000	0.0030	0.5929	0.5995	0.0066	1.0978
4	1	0.4000	0.0030	0.7736	0.7856	0.0120	1.5255
5	1	0.5000	0.0030	0.9016	0.8764	0.0252	2.8773
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
1	6	0.1000	0.0180	0.1770	0.1954	0.0183	9.3795
2	6	0.2000	0.0180	0.3435	0.3816	0.0381	9.9848
3	6	0.3000	0.0180	0.4874	0.5430	0.0557	10.2494
4	6	0.4000	0.0180	0.5953	0.6561	0.0608	9.2634
5	6	0.5000	0.0180	0.6549	0.6972	0.0423	6.0700
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
1	11	0.1000	0.0330	0.1410	0.1770	0.0359	20.3029
2	11	0.2000	0.0330	0.2714	0.3394	0.0680	20.0451
3	11	0.3000	0.0330	0.3809	0.4720	0.0911	19.3023
4	11	0.4000	0.0330	0.4606	0.5594	0.0989	17.6717
5	11	0.5000	0.0330	0.5037	0.5901	0.0864	14.6436
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
1	15	0.1000	0.0450	0.1156	0.1593	0.0437	27.4318
2	15	0.2000	0.0450	0.2222	0.3040	0.0818	26.9190
3	15	0.3000	0.0450	0.3113	0.4201	0.1088	25.9009
4	15	0.4000	0.0450	0.3758	0.4954	0.1196	24.1373
5	15	0.5000	0.0450	0.4106	0.5215	0.1109	21.2652

Caso 3. $r = 0.50, h = 0.1, k = 0.0050$							
i	j	x	t	$u(x_i, t_j)$	$U(x, t)$	Dif	Error
1	1	0.1000	0.0050	0.1993	0.2000	0.0007	0.3488
2	1	0.2000	0.0050	0.3972	0.3998	0.0026	0.6626
3	1	0.3000	0.0050	0.5895	0.5966	0.0071	1.1920
4	1	0.4000	0.0050	0.7608	0.7667	0.0059	0.7700
5	1	0.5000	0.0050	0.8536	0.8404	0.0132	1.5667
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
1	4	0.1000	0.0200	0.1836	0.1935	0.0099	5.1057
2	4	0.2000	0.0200	0.3563	0.3766	0.0203	5.3891
3	4	0.3000	0.0200	0.5042	0.5334	0.0291	5.4602
4	4	0.4000	0.0200	0.6105	0.6418	0.0313	4.8731
5	4	0.5000	0.0200	0.6590	0.6808	0.0218	3.2038
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
1	12	0.1000	0.0600	0.1130	0.1382	0.0252	18.2645
2	12	0.2000	0.0600	0.2159	0.2631	0.0472	17.9370
3	12	0.3000	0.0600	0.2997	0.3626	0.0628	17.3302
4	12	0.4000	0.0600	0.3569	0.4267	0.0697	16.3473
5	12	0.5000	0.0600	0.3822	0.4488	0.0665	14.8286
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
1	20	0.1000	0.1000	0.0666	0.0933	0.0267	28.6218
2	20	0.2000	0.1000	0.1273	0.1776	0.0502	28.2939
3	20	0.3000	0.1000	0.1767	0.2444	0.0677	27.7122
4	20	0.4000	0.1000	0.2103	0.2873	0.0770	26.8120
5	20	0.5000	0.1000	0.2252	0.3021	0.0769	25.4701

Caso 2. $r = 0.48, h = 0.1, k = 0.0048$							
i	j	x	t	$u(x_i, t_j)$	$U(x, t)$	Dif	Error
1	1	0.1000	0.0048	0.1993	0.2000	0.0007	0.3389
2	1	0.2000	0.0048	0.3973	0.3999	0.0026	0.6499
3	1	0.3000	0.0048	0.5898	0.5970	0.0072	1.2092
4	1	0.4000	0.0048	0.7619	0.7686	0.0067	0.8688
5	1	0.5000	0.0048	0.8579	0.8436	0.0143	1.6922
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
1	5	0.1000	0.0240	0.1756	0.1891	0.0135	7.1315
2	5	0.2000	0.0240	0.3391	0.3658	0.0267	7.2965
3	5	0.3000	0.0240	0.4770	0.5141	0.0371	7.2208
4	5	0.4000	0.0240	0.5746	0.6146	0.0400	6.5085
5	5	0.5000	0.0240	0.6194	0.6504	0.0310	4.7622
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
1	9	0.1000	0.0432	0.1389	0.1620	0.0231	14.2364
2	9	0.2000	0.0432	0.2660	0.3092	0.0433	13.9879
3	9	0.3000	0.0432	0.3700	0.4275	0.0575	13.4496
4	9	0.4000	0.0432	0.4416	0.5044	0.0628	12.4537
5	9	0.5000	0.0432	0.4739	0.5311	0.0572	10.7737
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
1	17	0.1000	0.0816	0.0832	0.1119	0.0287	25.6877
2	17	0.2000	0.0816	0.1590	0.2129	0.0539	25.3273
3	17	0.3000	0.0816	0.2207	0.2931	0.0723	24.6839
4	17	0.4000	0.0816	0.2630	0.3446	0.0816	23.6810
5	17	0.5000	0.0816	0.2820	0.3623	0.0804	22.1768

Caso 4. $r = 0.70, h = 0.1, k = 0.0070$							
i	j	x	t	$u(x_i, t_j)$	$U(x, t)$	Dif	Error
1	1	0.1000	0.0070	0.1991	0.2000	0.0008	0.4209
2	1	0.2000	0.0070	0.3965	0.3992	0.0027	0.6749
3	1	0.3000	0.0070	0.5867	0.5911	0.0044	0.7486
4	1	0.4000	0.0070	0.7504	0.7475	0.0029	0.3877
5	1	0.5000	0.0070	0.8149	0.8112	0.0037	0.4540
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
1	5	0.1000	0.0350	0.1662	0.1741	0.0079	4.5455
2	5	0.2000	0.0350	0.3186	0.3335	0.0149	4.4577
3	5	0.3000	0.0350	0.4433	0.4630	0.0196	4.2390
4	5	0.4000	0.0350	0.5273	0.5481	0.0208	3.7969
5	5	0.5000	0.0350	0.5604	0.5778	0.0175	3.0220
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
1	13	0.1000	0.0910	0.0895	0.1020	0.0125	12.2943
2	13	0.2000	0.0910	0.1705	0.1940	0.0235	12.1169
3	13	0.3000	0.0910	0.2356	0.2671	0.0315	11.8020
4	13	0.4000	0.0910	0.2785	0.3140	0.0355	11.3149
5	13	0.5000	0.0910	0.2952	0.3302	0.0350	10.5901
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
1	20	0.1000	0.1400	0.0513	0.0629	0.0116	18.3896
2	20	0.2000	0.1400	0.0978	0.1197	0.0218	18.2221
3	20	0.3000	0.1400	0.1352	0.1647	0.0295	17.9261
4	20	0.4000	0.1400	0.1598	0.1936	0.0338	17.4705
5	20	0.5000	0.1400	0.1694	0.2036	0.0342	16.7954

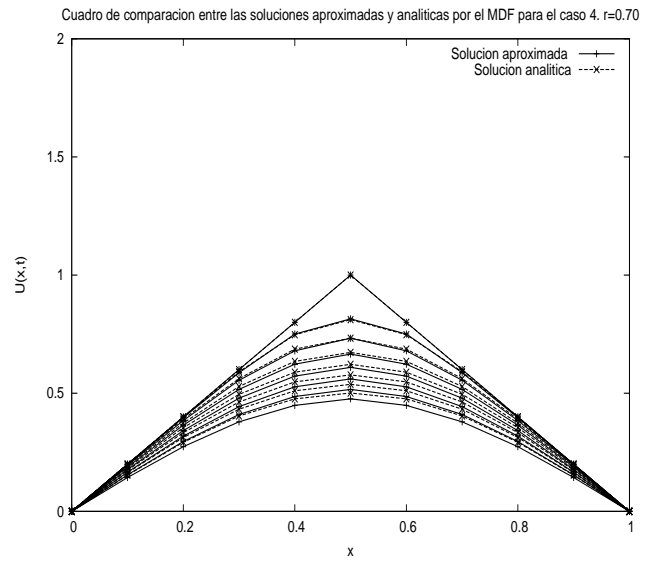
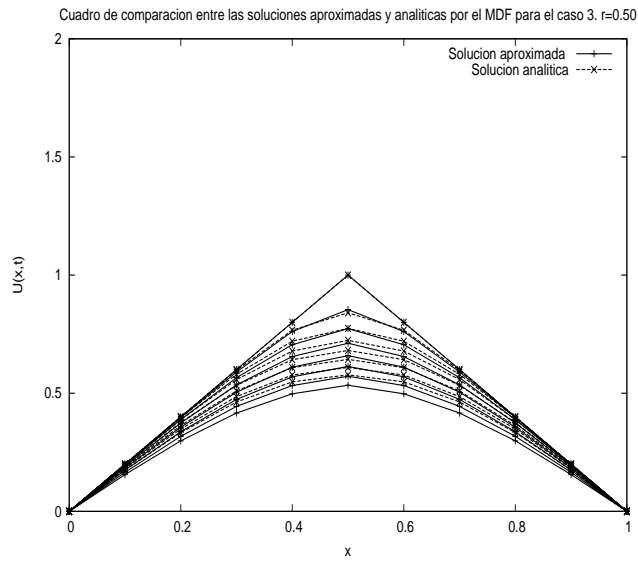
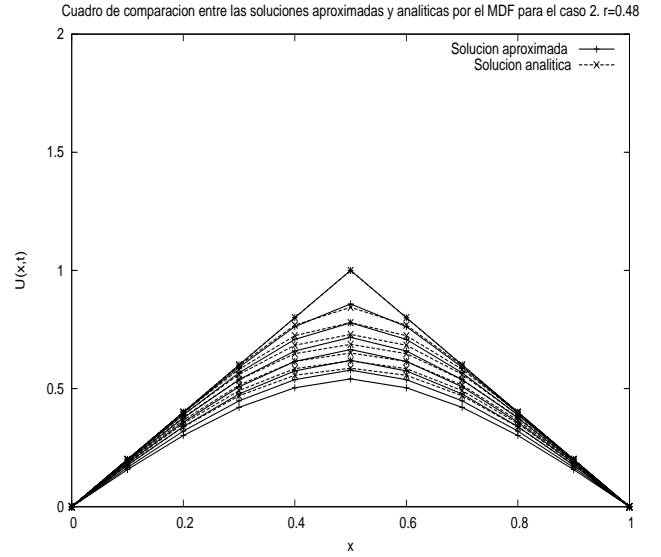
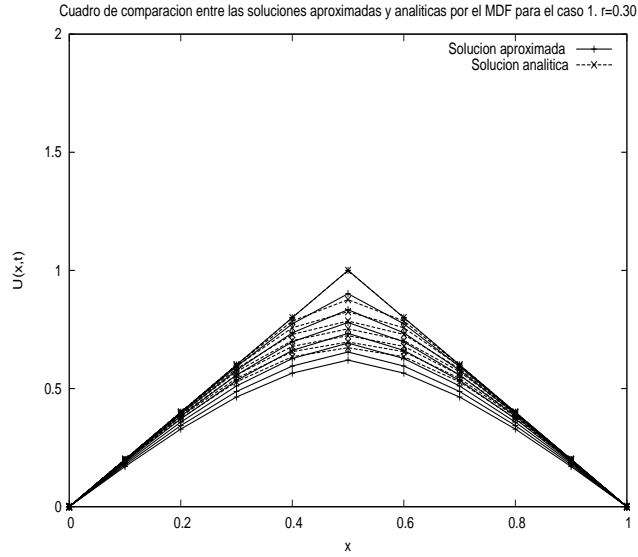


Figure 5: Cuadros de comparación entre las soluciones analíticas y aproximadas por el Método Implícito para el Problema 1.

E Problema 2., puede ser resuelto aplicando el método **Crank-Nicolson**, teniendo en cuenta que la expresión en MDF se ve así:

$$\frac{u_{i,j+1} - u_{i,j}}{k} = \frac{1}{2} \left(\frac{u_{i+1,j+1} - 2u_{i,j+1} + u_{i-1,j+1}}{h^2} + \frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{h^2} \right) \quad (37)$$

Recordemos que expresamos la segunda derivada con respecto a x mediante la media de las derivadas centrales, después de algunas transformaciones tenemos:

$$-ru_{i-1,j+1} + (2 + 2r)u_{i,j+1} - ru_{i+1,j+1} = ru_{i-1,j} + (2 - 2r)u_{i,j} + ru_{i+1,j}. \quad (38)$$

Si colocamos $i = 0$ tenemos de (38)

$$-ru_{-1,j+1} + (2 + 2r)u_{0,j+1} - ru_{1,j+1} = ru_{-1,j} + (2 - 2r)u_{0,j} + ru_{1,j}. \quad (39)$$

De (39) observamos que tenemos valores desconocidos como $u_{-1,j}$ y $u_{-1,j+1}$, para eliminarlos utilizamos las condiciones de frontera en $x = 0$ tanto para j como para $j + 1$ (también usando diferencias centrales),

$$\frac{u_{1,j} - u_{-1,j}}{2h} = u_{0,j}, \quad (40)$$

$$\frac{u_{1,j+1} - u_{-1,j+1}}{2h} = u_{0,j}. \quad (41)$$

De lo anterior sigue

$$u_{-1,j} = u_{1,j} - 2hu_{0,j}, \quad (42)$$

y

$$u_{-1,j+1} = u_{1,j+1} - 2hu_{0,j+1}. \quad (43)$$

Con (39) , (40) y (41) eliminamos los valores ficticios $u_{-1,j}$ y $u_{-1,j+1}$.

Por idea, algo parecido ocurre cuando $x = 1$, realizando un análisis similar podemos eliminar los valores ficticios para $i = N$, en (39) tales como $u_{N+1,j}$ y $u_{N+1,j+1}$.

En la siguiente página mostramos los resultados numéricos para el Problema 2. aplicando **Crank-Nicolson**.

Tabla 4: Resultados Numéricos del Problema 2. aplicando el Método Implícito.

Caso 1. $r = 0.30, h = 0.1, k = 0.0030$							
i	j	x	t	$u(x_i, t_j)$	$U(x, t)$	Dif	Error
0	1	0.0000	0.0030	0.9537	0.9411	0.0126	1.3376
1	1	0.1000	0.0030	0.9946	0.9930	0.0015	0.1557
2	1	0.2000	0.0030	0.9994	0.9998	0.0004	0.0399
3	1	0.3000	0.0030	0.9999	1.0000	0.0001	0.0072
4	1	0.4000	0.0030	1.0000	1.0000	0.0000	0.0009
5	1	0.5000	0.0030	1.0000	1.0000	0.0000	0.0005
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
0	5	0.0000	0.0150	0.8794	0.8755	0.0039	0.4480
1	5	0.1000	0.0150	0.9478	0.9444	0.0033	0.3540
2	5	0.2000	0.0150	0.9814	0.9802	0.0012	0.1220
3	5	0.3000	0.0150	0.9944	0.9945	0.0001	0.0103
4	5	0.4000	0.0150	0.9984	0.9988	0.0004	0.0424
5	5	0.5000	0.0150	0.9989	0.9996	0.0007	0.0720
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
0	34	0.0000	0.1020	0.7130	0.7150	0.0021	0.2887
1	34	0.1000	0.1020	0.7776	0.7800	0.0024	0.3093
2	34	0.2000	0.1020	0.8278	0.8313	0.0035	0.4258
3	34	0.3000	0.1020	0.8628	0.8683	0.0055	0.6347
4	34	0.4000	0.1020	0.8822	0.8906	0.0084	0.9409
5	34	0.5000	0.1020	0.8859	0.8981	0.0122	1.3556
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
0	67	0.0000	0.2010	0.5948	0.6030	0.0082	1.3648
1	67	0.1000	0.2010	0.6488	0.6580	0.0092	1.3914
2	67	0.2000	0.2010	0.6911	0.7017	0.0107	1.5215
3	67	0.3000	0.2010	0.7207	0.7335	0.0128	1.7474
4	67	0.4000	0.2010	0.7373	0.7528	0.0156	2.0694
5	67	0.5000	0.2010	0.7404	0.7593	0.0189	2.4935

Caso 1. $r = 0.50, h = 0.1, k = 0.0050$							
i	j	x	t	$u(x_i, t_j)$	$U(x, t)$	Dif	Error
0	1	0.0000	0.0050	0.9317	0.9250	0.0067	0.7294
1	1	0.1000	0.0050	0.9883	0.9841	0.0042	0.4290
2	1	0.2000	0.0050	0.9980	0.9984	0.0004	0.0369
3	1	0.3000	0.0050	0.9997	0.9999	0.0003	0.0271
4	1	0.4000	0.0050	0.9999	1.0000	0.0001	0.0061
5	1	0.5000	0.0050	1.0000	1.0000	0.0000	0.0031
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
0	3	0.0000	0.0150	0.8791	0.8755	0.0035	0.4030
1	3	0.1000	0.0150	0.9477	0.9444	0.0032	0.3439
2	3	0.2000	0.0150	0.9816	0.9802	0.0014	0.1421
3	3	0.3000	0.0150	0.9945	0.9945	0.0000	0.0005
4	3	0.4000	0.0150	0.9984	0.9988	0.0004	0.0409
5	3	0.5000	0.0150	0.9990	0.9996	0.0006	0.0602
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
0	20	0.0000	0.1000	0.7166	0.7176	0.0010	0.1368
1	20	0.1000	0.1000	0.7817	0.7828	0.0011	0.1405
2	20	0.2000	0.1000	0.8325	0.8342	0.0017	0.2071
3	20	0.3000	0.1000	0.8684	0.8713	0.0029	0.3323
4	20	0.4000	0.1000	0.8890	0.8936	0.0046	0.5171
5	20	0.5000	0.1000	0.8941	0.9011	0.0069	0.7674
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
0	40	0.0000	0.2000	0.5991	0.6040	0.0049	0.8077
1	40	0.1000	0.2000	0.6537	0.6591	0.0054	0.8145
2	40	0.2000	0.2000	0.6967	0.7029	0.0062	0.8870
3	40	0.3000	0.2000	0.7273	0.7348	0.0075	1.0202
4	40	0.4000	0.2000	0.7450	0.7541	0.0092	1.2137
5	40	0.5000	0.2000	0.7494	0.7606	0.0112	1.4711

Caso 1. $r = 0.48, h = 0.1, k = 0.0048$							
i	j	x	t	$u(x_i, t_j)$	$U(x, t)$	Dif	Error
0	1	0.0000	0.0048	0.9337	0.9264	0.0073	0.7899
1	1	0.1000	0.0048	0.9890	0.9850	0.0040	0.4042
2	1	0.2000	0.0048	0.9982	0.9986	0.0004	0.0401
3	1	0.3000	0.0048	0.9997	0.9999	0.0002	0.0249
4	1	0.4000	0.0048	0.9999	1.0000	0.0001	0.0053
5	1	0.5000	0.0048	1.0000	1.0000	0.0000	0.0027
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
0	3	0.0000	0.0144	0.8815	0.8778	0.0037	0.4189
1	3	0.1000	0.0144	0.9498	0.9465	0.0033	0.3504
2	3	0.2000	0.0144	0.9828	0.9815	0.0013	0.1355
3	3	0.3000	0.0144	0.9950	0.9951	0.0000	0.0046
4	3	0.4000	0.0144	0.9986	0.9990	0.0004	0.0396
5	3	0.5000	0.0144	0.9991	0.9997	0.0006	0.0557
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
0	11	0.0000	0.0528	0.7863	0.7853	0.0010	0.1281
1	11	0.1000	0.0528	0.8562	0.8551	0.0011	0.1295
2	11	0.2000	0.0528	0.9078	0.9072	0.0006	0.0677
3	11	0.3000	0.0528	0.9421	0.9425	0.0004	0.0443
4	11	0.4000	0.0528	0.9606	0.9626	0.0019	0.2018
5	11	0.5000	0.0528	0.9650	0.9691	0.0040	0.4148
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
0	42	0.0000	0.2016	0.5972	0.6024	0.0052	0.8661
1	42	0.1000	0.2016	0.6515	0.6573	0.0057	0.8746
2	42	0.2000	0.2016	0.6943	0.7010	0.0067	0.9521
3	42	0.3000	0.2016	0.7248	0.7328	0.0080	1.0933
4	42	0.4000	0.2016	0.7423	0.7521	0.0098	1.2979
5	42	0.5000	0.2016	0.7466	0.7585	0.0119	1.5698

Caso 1. $r = 0.70, h = 0.1, k = 0.0070$							
i	j	x	t	$u(x_i, t_j)$	$U(x, t)$	Dif	Error
0	1	0.0000	0.0070	0.9135	0.9122	0.0014	0.1492
1	1	0.1000	0.0070	0.9814	0.9752	0.0062	0.6379
2	1	0.2000	0.0070	0.9960	0.9958	0.0002	0.0232
3	1	0.3000	0.0070	0.9991	0.9996	0.0005	0.0457
4	1	0.4000	0.0070	0.9998	1.0000	0.0002	0.0178
5	1	0.5000	0.0070	0.9999	1.0000	0.0001	0.0098
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
0	8	0.0000	0.0560	0.7808	0.7799	0.0009	0.1164
1	8	0.1000	0.0560	0.8505	0.8495	0.0011	0.1246
2	8	0.2000	0.0560	0.9026	0.9018	0.0008	0.0851
3	8	0.3000	0.0560	0.9377	0.9376	0.0001	0.0122
4	8	0.4000	0.0560	0.9574	0.9582	0.0008	0.0829
5	8	0.5000	0.0560	0.9630	0.9649	0.0019	0.1972
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
0	15	0.0000	0.1050	0.7108	0.7113	0.0004	0.0626
1	15	0.1000	0.1050	0.7755	0.7759	0.0004	0.0542
2	15	0.2000	0.1050	0.8263	0.8270	0.0007	0.0846
3	15	0.3000	0.1050	0.8626	0.8639	0.0013	0.1492
4	15	0.4000	0.1050	0.8840	0.8862	0.0022	0.2464
5	15	0.5000	0.1050	0.8902	0.8936	0.0034	0.3776
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
0	29	0.0000	0.2030	0.5984	0.6009	0.0025	0.4217
1	29	0.1000	0.2030	0.6530	0.6557	0.0027	0.4137
2	29	0.2000	0.2030	0.6962	0.6993	0.0031	0.4436
3	29	0.3000	0.2030	0.7273	0.7310	0.0037	0.5079
4	29	0.4000	0.2030	0.7457	0.7503	0.0045	0.6059
5	29	0.5000	0.2030	0.7511	0.7567	0.0056	0.7395

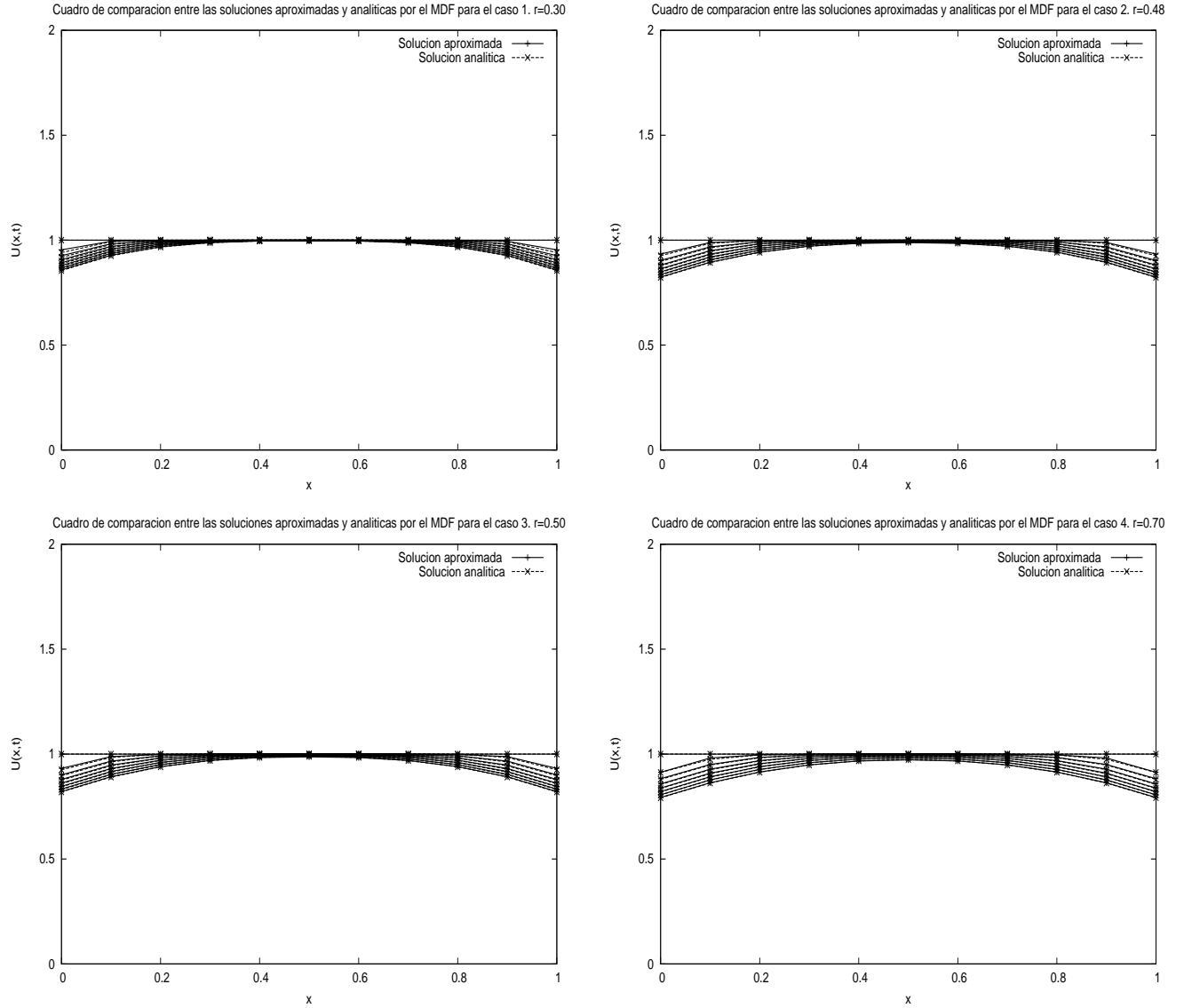


Figure 6: Cuadros de comparación entre las soluciones analíticas y aproximadas por el Método Implícito para el Problema 2.

5 Conclusión

En este artículo, Nosotros mostramos numéricamente que para valores del parámetro de estabilidad $r \leq \frac{1}{2}$, el método es estable(útil), en cambio para $r > \frac{1}{2}$ el proceso es divergente. El método implícito es convergente para cualquier r , teóricamente, sin embargo para r grandes, el método genera soluciones oscilatorias.

6 Agradecimientos

Queremos agradecer al profesor Valdemir Garcia Ferreira por todos los consejos y sugerencias hechas durante la realización de este trabajo. Extendemos nuestros agradecimientos también a la Universidad del Atlántico por el apoyo prestado.

Bibliografía

- [1] N.N. Kalitkin, “Chislennie Metodi”, Nauka,Moskva,1978.
- [2] G.U. Marchuk, “Metodi vuichislitelnoi matematiki”, Nauka, Moskva 1977.
- [3] G.D. Smith, “Numerical Solution of Partial DifferencIal Equations.Finite Difference Methods”.Third Edition.Oxford Apllied Mathematics and Computing Science Series.UK,2005.
- [4] V. Garcia Ferreira, “Introducao À Solucao Numérica de Equacoes Diferenciais Parciais com Aplicacoes Em Dinamica dos Fluidos”.Cursillo Universidad del Atlántico.2007.

7 Anexos

A continuación presentamos los programas hechos y compilados en **C++(Borland C++ V. 4.0)**, los cuadros fueron representados mediante el programa **GNU PLOT**.

```
\\PROGRAMA QUE HALLA NUMERICAMENTE MEDIANTE MDF LA SOLUCION DEL PROBLEMA 1. POR EL METODO
EXPLICITO

#include <stdio>
#include<conio>
#include<iostream>
#include<math.h>
#define Pi 3.141592653589
using namespace std;
//r=0.48 explicito048.txt
//r=0.50 explicito050.txt
//r=0.70 explicito070.txt
//r=0.30 explicito030.txt
//funcion u(x,t) aproximada long double U(double x, double t)
{
long double z; int i; z=0;
for (i=1; i<70;i++)
{
z = z+8/(Pi*Pi)*1/(i*i)*sin(Pi*i/2)*sin(i*Pi*x)*exp(-Pi*Pi*t*i*i);
}
}
```

```

return(z);
}
//fin funcion aproximada
////Main program
main ()
{
FILE *f;
f=fopen ("explicito070.txt","w+");
int i,j,N,T,l,c;
float s1,s2,s3,r,h,k,x,xx,t,error,diferencia;
float zz;
float u[11][101];
r=0.70;
h=0.1;
k=r*h*h;
N=10;
T=50;
fprintf(f,"r=%4.2f, h=%4.2f,k=%6.4f \n ",r,h,k);
fprintf(f,"i   j   x   t   U[xi,yj]   U[x,y]   diferencia   error \n ");
printf("i   j   x   t   U[xi,yj]   U[x,y]   diferencia   error \n ");
for (l=0;l<=N+1;l++) //valores de U con las condiciones iniciales
{
x=l*h;
if ((x>=0 ) & (x<= 0.5))
{
u[l][0]=2*x;
fprintf(f,"%d %d %5.3f %5.3f %6.4f %6.4f %6.4f %6.4f \n ",l,0,x,0.0,u[l][0],u[l][0],0.0,0.0);
printf("%d %d %5.3f %5.3f %6.4f %6.4f %6.4f %6.4f \n ",l,0,x,0.0,u[l][0],u[l][0],0.0,0.0);
} //fin del if
if ((x>0.5 ) & (x<=1))
{
u[l][0]=2*(1-x);
fprintf(f,"%d %d %5.3f %5.3f %6.4f %6.4f %6.4f %6.4f \n ",l,0,x,0.0,u[l][0],u[l][0],0.0,0.0);
printf("%d %d %5.3f %5.3f %6.4f %6.4f %6.4f %6.4f \n ",l,0,x,0.0,u[l][0],u[l][0],0.0,0.0);
} //fin del if
} //fin del for
fprintf(f,"%d %d %5.3f %5.3f %6.4f %6.4f %6.4f %6.4f \n ",l,0,x,0.0,u[l][0],u[l][0],0.0,0.0);
printf("%d %d %5.3f %5.3f %6.4f %6.4f %6.4f %6.4f \n ",l,0,x,0.0,u[l][0],u[l][0],0.0,0.0);
for (j=0;j<=100;j++)
{
u[0][j]=u[10][j]=0;
t=(j+1)*k;
for (i=0;i<N+1;i++)
{
if ((i==0) || (i==10))
{
x=i*h;
fprintf(f,"%d %d %5.3f %5.3f %6.4f %6.4f %6.4f %6.4f \n ",i,j+1,x,t,u[i][j],u[i][j],0.0,0.0);
printf("%d %d %5.3f %5.3f %6.4f %6.4f %6.4f %6.4f \n ",i,j+1,x,t,u[i][j],u[i][j],0.0,0.0);
}
else
{
x=i*h;
t=(j+1)*k;
s1=u[i-1][j];
s2=u[i][j];
s3=u[i+1][j];
u[i][j+1]=r*s1+(1-2*r)*s2+r*s3;
zz=U(x,t);
diferencia=fabs((u[i][j+1]-zz)/zz)*100;
fprintf(f,"%d %d %5.3f %5.3f %6.4f %6.4f %6.4f %6.4f \n ",i,j+1,x,t,u[i][j+1],zz,diferencia,error);
printf("%d %d %5.3f %5.3f %6.4f %6.4f %6.4f %6.4f \n ",i,j+1,x,t,u[i][j+1],zz,diferencia,error);
}
} // fin del for i
} //fin del for j
fclose(f);
printf("fin del programa ");

```

```
while((c = getchar()) != '\n' & & c != EOF);  
}  
///fin del programa
```

\\PROGRAMA QUE HALLA NUMERICAMENTE MEDIANTE MDF LA SOLUCION DEL PROBLEMA 2. POR EL METODO EXPLICITO

```
#include<stdio>
#include<conio>
#include<iostream>
#include<math.h>
#define Pi 3.141592653589
using namespace std;
// funcion original
double foriginal(double xx)
{
double t,zz;
t=cos(xx);
if (cos(xx)!=0)
{
zz= xx*tan(xx)-0.5;
return(zz);
}
else {
printf("Division por cero\n"); exit(0); }
}
/////
/// primera derivada de la solucion analitica
double fderivada(double xx )
{
double t,zz;
t= cos(xx);
if (t==0)
{
printf("hay un error\n"); exit(1); }
else {
zz=tan(xx)+xx/(pow(cos(xx),2));
return(zz);
}
}
/// Metodo de Newton para hallar la solución numérica de  $x \tan x = 1/2$ 
double Newton( int n )
{
int j;
double u1,u2,epsilon,zz,x1,x2;
double c,u3;
n=n-1;
c=(2*n+1)*Pi/2.0-Pi/180.0;
epsilon=0.001;
j=0;
x2=c;
do
{
x1=x2;
u2=foriginal(x1);
u3=fderivada(x1);
x2=x1-u2/u3;
j=j+1;
u1=fabs(x2-x1);
} while(u1>epsilon);
zz=x2;
return(zz);
} //fin de la funcion newton
//
long double U(double xx, double tt)
{
double alpha,z; int i;
z=0;
for (i=1;i<10;i++)
{
alpha=Newton(i);
z = 1/(cos(alpha)*(3+4*pow(alpha,2))) * exp(-4*pow(alpha,2)*tt)*cos(2*alpha*(xx-0.5))+z;
}
```

```

}
z=z*4;
return(z);
}
////////fin funcion analitica
///Main program
main ()
{
FILE *f;
f=fopen("explicitobvp030.txt","w+");
int i,j,N,T,l,m,c;
double s1,s2,s3,s4,s5,s6,s7,s8,r,h,k,x,xx,t,error,diferencia;
double zz,zzz;
float u[11][101];
r=0.30;
h=0.1;
k=r*h*h;
j=0;
N=10;
T=100;
fprintf(f,"PARA EL CASO CUANDO BVP CON DERIVADAS PARCIALES\n");
fprintf(f,"r=%4.2f, h=%6.4f,k=%6.k\n",r,h,k);
fprintf(f,"i j x t U[i,j] Uanalitica diferencia error \n");
printf("i j x t U[i,j] Uanalitica diferencia error \n");
///vienen las condiciones iniciales
for (l=0; l<11;l++)
{
x=l*h;
s5=u[l][0]=1.0;
zzz=1.0;
diferencia=fabs(zzz-s5);
error=(diferencia)/zzz*100;
fprintf(f,"%d %d %6.4f %6.4f %6.4f %6.4f %6.4f %4.2f \n", l,0,x,0.0,s5,zzz,diferencia,error);
printf("%d %d %6.4f %6.4f %6.4f %6.4f %6.4f %4.2f \n", l,0,x,0.0,s5,zzz,diferencia,error);
}
for (j=0;j<=T-1;j++)
{
///calculamos los valores en la frontera
s7=u[0][j];s8=u[1][j];
u[0][j+1]=u[0][j]+2*r*(u[1][j]-(1+h)*u[0][j]);
s5=u[0][j+1];
t=(j+1)*k;//ojo con la t
l=1;//un nuevo contador sirve cuando se vaya a buscar los valores simetricos
u[10][j+1]=u[10][j]+2*r*(u[9][j]-(1+h)*u[10][j]);
s6=u[10][j+1];
for (i=0;i<N+1;i++)
{
x=i*h;
if (i==0)
{
zzz=U(x,t);
diferencia=fabs(zzz-s5);
error=(diferencia)/zzz*100;
}
if (i==10)
{
zzz=U(x,t);
diferencia=fabs(zzz-s6);
error=(diferencia)/zzz*100;
}
if ((i<5) && (i!=0))
{
s1=u[i-1][j];
s2=u[i][j];
s3=u[i+1][j];
u[i][j+1]=s2+r*(s1-2*s2+s3);
s4=u[i][j+1];
}
}
}

```

```

if (i==5)
{
s1=u[i-1][j];
s2=u[i][j];
s3=u[i+1][j];
u[i][j+1]=s2+r*(s1-2*s2+s3);
s4=u[i][j+1];
}
if ((i>5) && (i!=N))
{
m=i-2*1;
u[i][j+1]=u[m][j+1];
l=l+1;
s4=u[i][j+1];
}
zzz=U(x,t);
diferencia=fabs(zzz-u[i][j+1]);
error=diferencia/zzz*100;
fprintf(f, "%d %d %6.4f %6.4f %6.4f %6.4f %6.4f %4.2f \n", i,j+1,x,t,u[i][j+1],zzz,diferencia,error);
printf("%d %d %6.4f %6.4f %6.4f %6.4f %6.4f %4.2f \n", i,j+1,x,t,u[i][j+1],zzz,diferencia,error);
} // FIN DEL FOR PRINCIPAL DEL I
} //FIN DEL FOR PRINCIPAL DE J
fclose(f);
printf("fin del programa");
while((c = getchar()) != '\n' && c != EOF);
} //fin del programa

```

\\PROGRAMA QUE HALLA NUMERICAMENTE MEDIANTE MDF LA SOLUCION DEL PROBLEMA 1. POR EL METODO IMPLICITO

```

#include<iostream>
#include<stdlib>
#include<stdio>
#define Pi 3.141592653589
using namespace std;
///funcion u(x,t) aproximada
long double U(double x, double t)
{
    long double z; int i;
    z=0;
    for (i=1; i<20;i++)

    z = z+8/(Pi*Pi)*1/(i*i)*sin(Pi*i/2)*sin(i*Pi*x)*exp(-Pi*Pi*t*i*i);

    return(z);
} //fin funcion aproximada //Main Program
int main()
{
    int i,j,c,l,s;
    FILE *f1;
    int N=5;
    float r=0.70;
    float h=0.1;
    float k=r*h*h;
    float a[20][20],b[20][20],A[20],B[20],x,xx,diferencia,error,t,zzz;
    f=fopen("implicito070.txt","w+");
    fprintf(f1,"PARA EL CASO IMPLICITO CRANCK NICHOLSON BVP \n");
    fprintf(f1,"r=%3.2f, h=%3.2f,k=%5.3f \n",r,h,k);
    fprintf(f1," i j x t U[i,j] Uanalitica diferencia error \n");
    printf("PARA EL CASO IMPLICITO CRANCK NICHOLSON BVP r=%3.2f, h=%3.2f\n",r,h);
    printf(" i j x t U[i,j] Uanalitica diferencia error \n");
    for (i=0;i<=2*N+1;i++) {
        x=i*h;
        if ((x>=0 ) & (x<=0.5)) {
            a[i][0]=2*x;
            a[10-i][0]=a[i][0];
            zzz=a[i][0];
            diferencia=fabs(zzz-a[i][0]);
            error=0;
            fprintf(f1," %d %d %6.4f %6.4f %6.4f %6.4f %6.4f %6.4f \n", i,0,x,0.0,a[i][0],zzz,diferencia,error);
            printf(" %d %d %6.4f %6.4f %6.4f %6.4f %6.4f %6.4f \n", i,0,x,0.0,a[i][0],zzz,diferencia,error);
        }
        //fin del if
        //por simetria tenemos
        if ((x>0.5 ) & (x<1))
        {
            zzz=a[10-i][0];
            diferencia=fabs(zzz-a[i][0]);
            error=diferencia/zzz*100;
            fprintf(f1," %d %d %6.4f %6.4f %6.4f %6.4f %6.4f %6.4f \n", i,0,x,0.0,a[i][0],zzz,diferencia,error);
            printf(" %d %d %6.4f %6.4f %6.4f %6.4f %6.4f %6.4f \n", i,0,x,0.0,a[i][0],zzz,diferencia,error);
        }
        //fin del if
        if (x==1)
        {
            zzz=a[10-i][0];
            diferencia=fabs(zzz-a[i][0]);
            error=0;
            fprintf(f1," %d %d %6.4f %6.4f %6.4f %6.4f %6.4f %6.4f \n", i,0,x,0.0,a[i][0],zzz,diferencia,error);
            printf(" %d %d %6.4f %6.4f %6.4f %6.4f %6.4f %6.4f \n", i,0,x,0.0,a[i][0],zzz,diferencia,error);
        }
    }
    //fin del for i para llenarlos valores iniciales
    for (j=0; j<20;j++) {
        for (i=1;i<N+1;i++) {

```



```

if (i==1) {
b[i][i]=4.0;
b[i][i+1]=-1;
rellena de ceros el lado derecho
for (l=i+2;l<N+1;l++)
{
b[i][l]=0.0;
}
b[i][N+1]= (2-2*r)*a[i][j]+r*a[i+1][j];
printf("\n");
fprintf(f,"\n");
}
if ((i>1) && (i!=N)) {
if(i>2) {
///rellena de ceros el lado izquierdo
for (l=1;l<i-1;l++)
{
b[i][l]=0.0;
}
}
b[i][i-1]=-1;
b[i][i]=4.0;
b[i][i+1]=-1;
for (l=i+2;l<N+1;l++)
{
b[i][l]=0.0;
}
b[i][N+1]= r*a[i-1][j]+(2-2*r)*a[i][j]+r*a[i+1][j];
} //fin del if i>1
if (i==N)
{
for (l=1;l<i-1;l++)
{
b[i][l]=0.0;
}
b[i][N-1]=-2.0*r;
b[i][N]=(2+2*r);
b[i][N+1]= r*a[i-1][j]+(2-2*r)*a[i][j]+r*a[i+1][j];
//a[i-1][j]+a[i+1][j];
printf("\n");
} } //fin del for I=1,N
printf("\n");
///METODO GAUSS SIN PIVOTEO-METOD PROGONKA//////-----///// //apli-
camos progona directa A[1]=-b[1][2]/b[1][1];
B[1]=b[1][N+1]/b[1][1];
for (i=2;i<N+1;i++)
{
if (i!= N)
{
A[i]=-b[i][i+1]/(b[i][i]+b[i][i-1]*A[i-1]);
}
B[i]=(b[i][N+1]-b[i][i-1]*B[i-1])/(b[i][i]+b[i][i-1]*A[i-1]);
}
//ahora hallamos las soluciones
//printf("\n");
xx=B[N];
//las soluciones son los elementos del vector a[i][j] a[N][j+1]=xx;
printf( "a[%d][%d]= %6.4f \n",N,j+1,a[N][j+1]);
l=N-1;
while (l>0) {
s=10-l;
xx=A[l]*xx+B[l];
a[l][j+1]=xx;
a[s][j+1]=xx;
fprintf(f1, "a[%d][%d]= %6.4f \n",l,j+1,a[s][j+1]);
printf( "a[%d][%d]= %6.4f \n",l,j+1,a[l][j+1]);
l=l-1;
}
}

```

```

printf("\n");
t=k*(j+1);
for (i=0;i<2*N+1;i++) //ahora escribe las soluciones en el archivo de texto
{
x=i*h;
if ((i==0) || (i==2*N))
{
printf(" %d %d %6.4f %6.4f %6.4f %6.4f %6.4f %6.4f \n", i,j+1,x,t,0.0,0.0,0.0,0.0);
fprintf(f1," %d %d %6.4f %6.4f %6.4f %6.4f %6.4f %6.4f \n", i,j+1,x,t,0.0,0.0,0.0,0.0);
}
else {
zzz=U(x,t);
diferencia=fabs(zzz-a[i][j+1]);
error=diferencia/zzz*100;
fprintf(f1," %d %d %6.4f %6.4f %6.4f %6.4f %6.4f %6.4f \n", i,j+1,x,t,a[i][j+1],zzz,diferencia,error);
printf(" %d %d %6.4f %6.4f %6.4f %6.4f %6.4f %6.4f \n", i,j+1,x,t,a[i][j+1],zzz,diferencia,error);
}
}
} //fin del j principal fclose(f);
fclose(f1);
while((c = getchar()) != '\n' && c != EOF);
}

```

\\PROGRAMA QUE HALLA NUMERICAMENTE MEDIANTE MDF LA SOLUCION DEL PROBLEMA 2. POR EL METODO IMPLICITO

```

#include<iostream>
#include<stdlib>
#include<stdio>
#define Pi 3.141592653589
using namespace std;
//programa con el metodo Crank-Nicholson para bvp con derivadas double foriginal(double xx)
{
double t,zz;
t=cos(xx);
if (cos(xx)!=0)
{
zz= xx*tan(xx)-0.5;
return(zz);
}
else
{
printf("Division por cero \n");
exit(0);
}
}
////
/// primera derivada de la solucion analitica
double fderivada(double xx )
{
double t,zz;
t= cos(xx);
if (t==0)
{
printf("hay un error\n");
exit(1);
}
else
{
zz=tan(xx)+xx/(pow(cos(xx),2));
return(zz);
}
}
double Newton( int n )
{
int j;
double u1,u2,epsilon,zz,x1,x2;
double c,u3;

```

```

n=n-1;
c=(2*n+1)*Pi/2.0-Pi/180.0;
epsilon=0.001;
j=0;
x2=c;
do
{
x1=x2;
u2=foriginal(x1);
u3=fderivada(x1);
x2=x1-u2/u3;
j=j+1;
u1=fabs(x2-x1);
}
while(u1>epsilon);
zz=x2;
return(zz);
}
//fin de la funcion newton/// //la siguiente procedura halla el valor de la funcion analitica en un punto long double U(double xx, double
tt) {
double alpha,z; int i;
z=0;
for (i=1;i<10;i++) {
alpha=Newton(i);
z = 1/(cos(alpha)*(3+4*pow(alpha,2))) * exp(-4*pow(alpha,2)*tt)*cos(2*alpha*(xx-0.5))+z;
}
z=z*4;
return(z);
}
////////fin funcion analitica
main ()
{
FILE *f;
f=fopen("imptobvpder070.txt","w+");
int i,j,N,T,l,m,c,s,ls,jj;
double s1,s2,s3,s4,s5,s6,r,h,k,x,xx,t,error,diferencia;
double zz,zzz;
float u[11][101],b[8][8],A[8],B[8];
r=0.70;
h=0.1;
k=r*h*h;
j=0;
N=5;
T=20;
fprintf(f,"PARA EL CASO IMPLICITO CRANK-IMPLICITO CON BVP CON DERIVADAS PARCIALES\n");
fprintf(f,"r=%5.3f, h=%5.3f,k=%6.4f\n",r,h,k);
fprintf(f," i j x t U[i,j] Uanalitica diferencia error \n");
printf(" i j x t U[i,j] Uanalitica diferencia error \n");
///vienen las condiciones iniciales
for (l=0; l<11;l++)
{
x=l*h;
u[l][0]=1.0;
zzz=1.0;
diferencia=fabs(zzz-u[l][0]);
error=(diferencia)/zzz*100;
fprintf(f," %d %d %6.4f %6.4f %6.4f %6.4f %5.3f \n", l,x,0.0,u[l][0],zzz,diferencia,error);
printf(" — %d — %d — %6.4f — %6.4f — %6.4f — %6.4f — %5.3f \n", l,x,0.0,u[l][0],zzz,diferencia,error);
}
//printf("\n");
for (j=0;j<100;j++)//ciclo del j principal
{
for (i=0;i<N+1;i++)
{
if (i==0)
{
b[i][i]=1+r*r*h;
b[i][i+1]=-r;

```

```

//rellena de ceros el lado derecho
for (l=i+2;l<N+1;l++)
{
b[i][l]=0.0;
}
b[i][N+1]= r*u[1][j]+(1-r*r*h)*u[0][j];
}
if ((i>0)&& (i!=N))
{
if(i>1)
{
//rellena de ceros el lado izquierdo
for (l=0;l<i-1;l++)
{
b[i][l]=0.0;
}
}
b[i][i-1]=-r;
b[i][i]=2+2*r;
b[i][i+1]=-r;
for (l=i+2;l<N+1;l++) //llena de ceros el lado derecho {
b[i][l]=0.0;
}
b[i][N+1]= r*u[i-1][j]+(2-2*r)*u[i][j]+r*u[i+1][j];
}
//fin del if i>1 if (i==N)
{
for (l=0;l<N-1;l++)
{
b[i][l]=0.0;
}
b[i][N-1]=-2.0;
b[i][N]=4.0;
b[i][N+1]= r*u[i-1][j]+(2-2*r)*u[i][j]+r*u[i+1][j];
}
}
//fin del for I=1,N
//ahora usemos el metod progonka
A[0]=-b[0][1]/b[0][0];
B[0]=b[0][N+1]/b[0][0];
for (i=1;i<N+1;i++)
{
if (i!= N) {
A[i]=-b[i][i+1]/(b[i][i]+b[i][i-1]*A[i-1]);
}
B[i]=(b[i][N+1]-b[i][i-1]*B[i-1])/(b[i][i]+b[i][i-1]*A[i-1]);
}
//ahora hallamos las soluciones xx=B[N];
//las soluciones son los elementos del vector a[i][j] jj=j+1;
u[N][jj]=xx;
u[2*N][jj]=xx;
l=N-1;
while (l>-1)
{
s=10-l;
xx=A[l]*xx+B[l];
u[l][jj]=xx;
u[s][jj]=xx;
l=l-1;
}
t=k*(jj);
for (i=0;i<2*N+1;i++) {
x=i*h;
zzz=U(x,t);
diferencia=fabs(zzz-u[i][jj]);
error=(diferencia)/zzz*100;
fprintf(f," %d %d %6.4f %6.4f %6.4f %6.4f %6.4f %6.4f \n", i,jj,x,t,u[i][jj],zzz,diferencia,error);
printf(" %d %d %6.4f %6.4f %6.4f %6.4f %6.4f %6.4f \n", i,jj,x,t,u[i][jj],zzz,diferencia,error);
}

```

```
}  
// printf("%6.4f, \n",u[10][50]);  
//////////fin progonka////////// }  
//fin del ciclo de j printf("fin del programa");  
while((c = getchar()) != '\n' && c != EOF);  
}  
///fin del programa
```