



Ecuaciones Diferenciales y Métodos Numéricos  
4º curso de Ingeniería Industrial

*Ecuaciones Diferenciales en Derivadas Parciales*

---

**Juan Pedro Ramírez Sánchez**  
**UHU 2008**



## Índice

---

Objeto e introducción .....	1
1. La Ecuación del Calor .....	2
1.1. Consideraciones previas.....	2
1.2. Condiciones de contorno.....	4
1.3. Métodos de resolución.....	4
1.4. Método Progresivo.....	6
1.4.1. Condiciones tipo Dirichlett-Dirichlett .....	7
1.4.2. Condiciones tipo Dirichlett-Neumann .....	8
1.4.3. Condiciones tipo Neumann-Dirichlett.....	9
1.4.4. Condiciones tipo Neumann-Neumann .....	10
1.5. Método Regresivo.....	11
1.5.1. Condiciones tipo Dirichlett-Dirichlett .....	13
1.5.2. Condiciones tipo Dirichlett-Neumann .....	14
1.5.3. Condiciones tipo Neumann-Dirichlett.....	15
1.5.4. Condiciones tipo Neumann-Neumann .....	16
1.6. Método de Crank-Nicholson.....	17
1.6.1. Condiciones tipo Dirichlett-Dirichlett .....	19
1.6.2. Condiciones tipo Dirichlett-Neumann .....	20
1.6.3. Condiciones tipo Neumann-Dirichlett.....	21
1.6.4. Condiciones tipo Neumann-Neumann .....	22
2. La Ecuación de Ondas .....	24
2.1. Consideraciones previas y condiciones de contorno .....	24
2.2. Resolución .....	25
2.2.1. Condiciones tipo Dirichlett-Dirichlett .....	26
2.2.2. Condiciones tipo Dirichlett-Neumann .....	27
2.2.3. Condiciones tipo Neumann-Dirichlett.....	28
2.2.4. Condiciones tipo Neumann-Neumann .....	29
3. Algunos ejercicios resueltos con MATLAB 7.0 .....	31
3.1. Ecuación del Calor, método Progresivo y condiciones DD .....	31
3.2. Ecuación del Calor, método Regresivo y condiciones NN.....	37
3.3. Ecuación del Calor, método Crack-Nicholson y condiciones NN .....	44
3.4. Ecuación de Ondas, condiciones NN.....	52



## Objeto e introducción

---

Este trabajo se presenta como un resumen del desarrollo teórico-práctico expuesto en la asignatura Ecuaciones Diferenciales y Métodos Numéricos para las Ecuaciones Diferenciales en Derivadas Parciales.

La estructura propuesta está formada por dos bloques, ambos referidos a las Ecuaciones del Calor y de Ondas. Los apartados 1 y 2 contienen el desarrollo teórico de la diferente casuística asociada a estas ecuaciones en función del método de resolución y las condiciones de contorno consideradas. El segundo bloque (apartado 3) presenta algunos ejercicios en los que se emplea el programa MATLAB 7.0 como herramienta de resolución.



## 1. La Ecuación del Calor

---

La ecuación del calor es una importante ecuación diferencial en derivadas parciales que describe la distribución del calor (o variaciones de la temperatura) en una región a lo largo del transcurso del tiempo. Para el caso de una función de tres variables en el espacio  $(x,y,z)$  y la variable temporal  $t$ , la ecuación del calor se define como:

$$\frac{\partial T}{\partial t} = k \left( \frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} + \frac{\partial^2 T}{\partial z^2} \right)$$

donde  $K$  es una constante dependiente del material y  $T$  es la temperatura en una posición e instante determinados.

La ecuación del calor es de una importancia fundamental en numerosos y diversos campos de la ciencia. En matemáticas, es la ecuación parabólica diferencial en derivadas parciales por antonomasia. En estadística, la ecuación del calor está vinculada con el estudio del movimiento browniano a través de la ecuación de Fokker–Planck. La ecuación de difusión es una versión más general de la ecuación del calor y se relaciona principalmente con el estudio de procesos de difusión química.

### *1.1. Consideraciones previas*

En la asignatura hemos empleado la Ecuación del Calor para resolver problemas en una sola dimensión conocida la distribución inicial de temperaturas (el caso base consiste en determinar la temperatura en cualquier punto de la barra para un instante dado  $T$ )

Se ha tomado la siguiente notación:

$$\frac{\partial u}{\partial t}(\mathbf{x}, t) = c^2 \frac{\partial^2 u}{\partial \mathbf{x}^2}(\mathbf{x}, t) + F(\mathbf{x}, t)$$



Simplificando:

$$u_t = c^2 u_{xx} + F$$

Para las condiciones de contorno:

$$u(0, t) \quad \text{ó} \quad \frac{\partial u}{\partial x}(0, t) = a(t), \quad u(L, t) \quad \text{ó} \quad \frac{\partial u}{\partial x}(L, t) = b(t), \quad t > 0$$

Y dadas unas condiciones iniciales tales que:

$$u(x, 0) = f(x), \quad 0 < x < L$$

En forma simplificada:

$$u_t = c^2 u_{xx} + F$$

Donde:

- $u(x, t)$  es la temperatura de la barra en un punto  $x$  y un instante  $t$
- $c$  es la difusividad térmica del material
- $F(x, t)$  es la perturbación externa para un punto  $x$  y un instante  $t$
- $u(x, 0)$  es la distribución inicial de temperaturas en la barra
- $u(0, t)$  es la distribución de temperaturas en el extremo inicial de la barra
- $u(L, t)$  es la distribución de temperaturas en el extremo final de la barra



## 1.2. Condiciones de contorno

En el estudio de la Ecuación del Calor se consideran dos tipos de condiciones de contorno.

- Las condiciones de contorno tipo Dirichlett, con las que las funciones  $u(x,0)$  y  $u(L,t)$  son conocidas y por tanto se dispone de la evolución de la temperatura en los extremos de la barra a lo largo del tiempo.
- Las condiciones tipo Neumann, caso en el que no se conoce directamente la temperatura en los extremos de la barra, sino su flujo ( $u_x(x,0)$  y  $u_x(L,t)$ )

Las condiciones pueden darse en sólo uno o en los dos extremos, de forma que los problemas se clasifican en tipo DD, DN, ND ó NN.

## 1.3. Métodos de resolución

En función del proceso empleado para la discretización del problema consideramos tres métodos de resolución de la Ecuación del Calor. En todos ellos se supone un número finito de nodos equiespaciados formando una malla rectangular. El paso horizontal de la malla se denomina  $h=L/m$  (correspondiente al eje espacial, donde  $L$  es la longitud de la barra y  $m$  el número de intervalos tomados sobre dicha magnitud) y el vertical  $k=T/N$  (correspondiente al eje temporal, donde  $T$  es el instante final en el que se quiere conocer la distribución de temperaturas y  $N$  el número de intervalos hasta alcanzarlo), de forma que:

$$x_{i+1} = x_i + h, \quad t_{j+1} = t_j + k$$

En el método *Progresivo*,  $u_x(x,t)$  y  $u_t(x,t)$  se aproximan en el nodo  $(x_i, t_j)$  mediante las expresiones:

$$u_x(x_i, t_j) \approx \frac{u_{i+1,j} - u_{i,j}}{h} \quad u_t(x_i, t_j) \approx \frac{u_{i,j+1} - u_{i,j}}{k}$$



Con errores de orden 1  $O(h)$  y  $O(k)$  respectivamente. Estas aproximaciones se conocen como diferencias finitas progresivas, dando lugar al nombre del método.

En caso de emplearse las diferencias finitas hacia atrás el método se denomina *Regresivo*. El error en este método sigue siendo de orden 1 ( $O(h)$  y  $O(k)$ ) y las expresiones son:

$$u_x(x_i, t_j) \approx \frac{u_{i,j} - u_{i-1,j}}{h} \qquad u_t(x_i, t_j) \approx \frac{u_{i,j} - u_{i,j-1}}{k}$$

Un tercer procedimiento, esta vez más exacto, se basa en las diferencias centrales o simétricas. Este método se denomina de *Crank-Nicholson*, presenta un error de orden 2 ( $O(h^2)$  y  $O(k^2)$ ) y emplea las siguientes expresiones de aproximación:

$$u_x(x_i, t_j) \approx \frac{u_{i+1,j} - u_{i-1,j}}{2h} \qquad u_t(x_i, t_j) \approx \frac{u_{i,j+1} - u_{i,j-1}}{2k}$$

Para aproximar las segundas derivadas se utilizarán diferencias centrales (errores de truncamiento de orden 2):

$$u_{xx}(x_i, t_j) \approx \frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{h^2} \qquad u_{tt}(x_i, t_j) \approx \frac{u_{i,j+1} - 2u_{i,j} + u_{i,j-1}}{k^2}$$

Una vez definidos los criterios de discretización y las condiciones de contorno, pasamos a exponer los diferentes escenarios de resolución de la Ecuación del Calor.



### 1.4. Método Progresivo

Partimos de la ecuación del calor:

$$\frac{\partial u}{\partial t}(x, t) = c^2 \frac{\partial^2 u}{\partial x^2}(x, t) + F(x, t)$$

y con las expresiones vistas en el apartado anterior:

$$\frac{\partial u}{\partial t}(x_i, t_j) = \frac{u(x_i, t_j + k) - u(x_i, t_j)}{k} + O(k)$$

$$\frac{\partial^2 u}{\partial x^2}(x_i, t_j) = \frac{u(x_i + h, t_j) - 2u(x_i, t_j) + u(x_i - h, t_j)}{h^2} + O(h^2)$$

Aproximamos eliminando los errores de truncamiento:

$$u(x_i, t_j) \approx W_{i,j}$$

$$F(x_i, t_j) \approx F_{i,j}$$

Y sustituyendo en la ecuación inicial:

$$\frac{W_{i,j+1} - W_{i,j}}{k} = c^2 \frac{W_{i+1,j} - 2W_{i,j} + W_{i-1,j}}{h^2} + F_{i,j}$$



Reuniendo las constantes en un mismo término:

$$\lambda = \frac{c^2 K}{h^2}$$

Finalmente obtenemos:

$$W_{i,j+1} = (1 - 2\lambda)W_{i,j} + \lambda W_{i+1,j} + W_{i-1,j} + kF_{i,j}$$

Esta expresión junto a las obtenidas a partir de las condiciones de contorno específicas nos permitirán definir un sistema matricial de ecuaciones que resuelva el problema.

#### 1.4.1. Condiciones tipo Dirichlett-Dirichlett

Las condiciones de contorno tipo Dirichlett implican valores conocidos de temperatura en los extremos, de forma que:

$$W_{0,j} = a(t_j); \quad W_{m,j} = b(t_j)$$

La condición inicial:

$$W_{i,0} = f(x_i)$$

Con las expresiones obtenidas hasta el momento y dando valores a  $i$  desde 1 hasta  $m-1$  es posible construir el sistema matricial de ecuaciones que resuelva la ecuación:



$$\begin{pmatrix} W_{1,j+1} \\ W_{2,j+1} \\ W_{3,j+1} \\ W_{4,j+1} \\ \vdots \\ W_{m-1,j+1} \end{pmatrix} = \begin{pmatrix} 1-2\lambda & \lambda & 0 & 0 & 0 & 0 & \dots & 0 & 0 \\ \lambda & 1-2\lambda & \lambda & 0 & 0 & 0 & \dots & & \\ 0 & \lambda & 1-2\lambda & \lambda & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & 0 & \lambda & 1-2\lambda & \lambda & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & 0 & \dots & \lambda & 1-2\lambda \end{pmatrix} \begin{pmatrix} W_{1,j} \\ W_{2,j} \\ W_{3,j} \\ W_{4,j} \\ \vdots \\ W_{m-1,j} \end{pmatrix} + \begin{pmatrix} kF_{1,j} + \lambda a(t_j) \\ kf_{2,j} \\ kf_{3,j} \\ kf_{4,j} \\ \vdots \\ kF_{m-1,j} + \lambda b(t_j) \end{pmatrix}$$

Simplificando, el sistema puede expresarse como:

$$\begin{cases} W_{j+1} = AW_j + bj \\ W_0 = \{f(x_i)\}_{i=1 \dots m-1} \end{cases}$$

#### 1.4.2. Condiciones tipo Dirichlett-Neumann

Se conocen directamente los valores de la distribución temporal de temperaturas en el extremo inicial de la barra pero no se dispone de información explícita del parámetro en el extremo final:

$$b(t_j) = \frac{W_{m+1,j} - W_{m-1,j}}{2h} \qquad W_{m+1} = W_{m-1,j} + 2hb(t_j)$$

$$W_{0,j} = a(t_j)$$

La condición inicial:

$$W_{i,0} = f(x_i)$$



En este caso el subíndice  $i$  toma valores desde  $1$  hasta  $m$  y el sistema matricial resulta:

$$\begin{pmatrix} W_{1,j+1} \\ W_{2,j+1} \\ W_{3,j+1} \\ W_{4,j+1} \\ \vdots \\ W_{m,j+1} \end{pmatrix} = \begin{pmatrix} 1-2\lambda & \lambda & 0 & 0 & 0 & 0 & \cdots & 0 & 0 \\ \lambda & 1-2\lambda & \lambda & 0 & 0 & 0 & \cdots & & \\ 0 & \lambda & 1-2\lambda & \lambda & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 0 & \lambda & 1-2\lambda & \lambda & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 2\lambda & 1-2\lambda \end{pmatrix} \begin{pmatrix} W_{1,j} \\ W_{2,j} \\ W_{3,j} \\ W_{4,j} \\ \vdots \\ W_{m,j} \end{pmatrix} + \begin{pmatrix} kF_{1,j} + \lambda a(t_j) \\ kF_{2,j} \\ kF_{3,j} \\ kF_{4,j} \\ \vdots \\ kF_{m,j} + 2h\lambda b(t_j) \end{pmatrix}$$

En forma simplificada:

$$\begin{cases} W_{j+1} = AW_j + bj \\ W_0 = \{f(x_i)\}_{i=1\dots m} \end{cases}$$

### 1.4.3. Condiciones tipo Neumann-Dirichlett

En este caso sólo se dispone de información explícita para el extremo final de la barra.

$$a(t_j) = \frac{W_{1,j} - W_{-1,j}}{2h} \quad W_{-1,j} = W_{1,j} - 2ha(t_j)$$

$$W_{m,j} = b(t_j)$$

La condición inicial:

$$W_{i,0} = f(x_i)$$



En este caso el subíndice  $i$  toma valores desde  $0$  hasta  $m-1$  y el sistema queda como sigue:

$$\begin{pmatrix} W_{0,j+1} \\ W_{1,j+1} \\ W_{2,j+1} \\ W_{3,j+1} \\ \vdots \\ W_{m-1,j+1} \end{pmatrix} = \begin{pmatrix} 1-2\lambda & 2\lambda & 0 & 0 & 0 & 0 & \dots & 0 & 0 \\ \lambda & 1-2\lambda & \lambda & 0 & 0 & 0 & \dots & & \\ 0 & \lambda & 1-2\lambda & \lambda & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & 0 & \lambda & 1-2\lambda & \lambda & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & 0 & \dots & \lambda & 1-2\lambda \end{pmatrix} \begin{pmatrix} W_{0,j} \\ W_{1,j} \\ W_{2,j} \\ W_{3,j} \\ \vdots \\ W_{m-1,j} \end{pmatrix} + \begin{pmatrix} kF_{0,j} - 2h\lambda a(t_j) \\ kF_{1,j} \\ kF_{2,j} \\ kF_{3,j} \\ \vdots \\ kF_{m-1,j} + \lambda b(t_j) \end{pmatrix}$$

Simplificando:

$$\begin{cases} W_{j+1} = AW_j + bj \\ W_0 = \{f(x_i)\}_{i=0\dots m-1} \end{cases}$$

#### 1.4.4. Condiciones tipo Neumann-Neumann

En este caso no se conocen directamente ninguna de las distribuciones temporales de temperatura de los extremos de barra:

$$a(t_j) = \frac{W_{1,j} - W_{-1,j}}{2h}; \quad b(t_j) = \frac{W_{m+1,j} - W_{m-1,j}}{2h}$$

$$W_{-1,j} = W_{1,j} - 2ha(t_j)$$

$$W_{m+1,j} = W_{m-1,j} + 2hb(t_j)$$



La condición inicial es la misma y en este caso el subíndice  $i$  toma valores desde  $0$  hasta  $m$ . El sistema queda como sigue:

$$\begin{pmatrix} W_{0,j+1} \\ W_{1,j+1} \\ W_{2,j+1} \\ W_{3,j+1} \\ \vdots \\ W_{m,j+1} \end{pmatrix} = \begin{pmatrix} 1-2\lambda & 2\lambda & 0 & 0 & 0 & 0 & \dots & 0 & 0 \\ \lambda & 1-2\lambda & \lambda & 0 & 0 & 0 & \dots & & \\ 0 & \lambda & 1-2\lambda & \lambda & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & 0 & \lambda & 1-2\lambda & \lambda & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & 0 & \dots & 2\lambda & 1-2\lambda \end{pmatrix} \begin{pmatrix} W_{0,j} \\ W_{1,j} \\ W_{2,j} \\ W_{3,j} \\ \vdots \\ W_{m,j} \end{pmatrix} + \begin{pmatrix} kF_{0,j} - 2h\lambda a(t_j) \\ kF_{1,j} \\ kF_{2,j} \\ kF_{3,j} \\ \vdots \\ kF_{m,j} + 2h\lambda b(t_j) \end{pmatrix}$$

Simplificando:

$$\begin{cases} W_{j+1} = AW_j + b_j \\ W_0 = \{f(x_i)\}_{i=0\dots m} \end{cases}$$

### 1.5. Método Regresivo

Partimos de la ecuación del calor:

$$\frac{\partial u}{\partial t}(x, t) = c^2 \frac{\partial^2 u}{\partial x^2}(x, t) + F(x, t)$$

y con las expresiones vistas anteriormente para las diferencias finitas regresivas:

$$\frac{\partial u}{\partial t}(x_i, t_j + k) = \frac{u(x_i, t_j + k) - u(x_i, t_j)}{k} + O(k)$$



$$\frac{\partial^2 u}{\partial x^2}(xi, tj + k) = \frac{u(xi + h, tj + k) - 2u(xi, tj + k) + u(xi - h, tj + k)}{h^2} + O(h^2)$$

Eliminando los errores de truncamiento, aproximamos:

$$u(xi, tj) \approx W_{i,j}$$

$$F(xi, tj) \approx F_{i,j}$$

Y sustituyendo en la ecuación inicial:

$$\frac{W_{i,j+1} - W_{i,j}}{k} = c^2 \frac{W_{i+1,j+1} - 2W_{i,j+1} + W_{i-1,j+1}}{h^2} + F_{i,j+1}$$

Reuniendo las constantes en un mismo término:

$$\lambda = \frac{c^2 K}{h^2}$$

Finalmente obtenemos:

$$(1 + 2\lambda)W_{i,j+1} - \lambda W_{i+1,j+1} - \lambda W_{i-1,j+1} = W_{i,j} + kF_{i,j+1}$$



Esta expresión junto a las obtenidas a partir de las condiciones de contorno específicas nos permitirán definir un sistema matricial de ecuaciones que resuelva el problema.

### 1.5.1. Condiciones tipo Dirichlett-Dirichlett

Las condiciones de contorno tipo Dirichlett implican valores conocidos de temperatura en los extremos, de forma que:

$$W_{0,j} = a(t_j + 1); \quad W_{m,j} = b(t_j + 1)$$

La condición inicial:

$$W_{i,0} = f(x_i)$$

Con las expresiones obtenidas hasta el momento y dando valores a  $i$  desde 1 hasta  $m-1$  es posible construir el sistema matricial de ecuaciones que resuelva la ecuación:

$$\begin{pmatrix} 1+2\lambda & -\lambda & 0 & 0 & 0 & 0 & \dots & 0 & 0 \\ -\lambda & 1+2\lambda & -\lambda & 0 & 0 & 0 & \dots & & \\ 0 & -\lambda & 1+2\lambda & -\lambda & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & 0 & -\lambda & 1+2\lambda & -\lambda & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & 0 & \dots & -\lambda & 1+2\lambda \end{pmatrix} \begin{pmatrix} W_{1,j+1} \\ W_{2,j+1} \\ W_{3,j+1} \\ W_{4,j+1} \\ \vdots \\ W_{m-1,j+1} \end{pmatrix} = \begin{pmatrix} W_{1,j} \\ W_{2,j} \\ W_{3,j} \\ W_{4,j} \\ \vdots \\ W_{m-1,j} \end{pmatrix} + \begin{pmatrix} kF_{1,j} + \lambda a(t_{j+1}) \\ kF_{2,j} \\ kF_{3,j} \\ kF_{4,j} \\ \vdots \\ kF_{m-1,j} + \lambda b(t_{j+1}) \end{pmatrix}$$

Simplificando, el sistema puede expresarse como:

$$\begin{cases} AW_{j+1} = W_j + bj \\ W_0 = \{f(x_i)\}_{i=1 \dots m-1} \end{cases}$$



### 1.5.2. Condiciones tipo Dirichlett-Neumann

Se conocen directamente los valores de la distribución temporal de temperaturas en el extremo inicial de la barra pero no se dispone de información explícita del parámetro en el extremo final:

$$W_{m+1,j+1} = W_{m-1,j+1} + 2hb(t_j + 1)$$

La condición inicial:

$$W_{0,j+1} = a(t_j + 1); \quad W_{i,0} = f(x_i)$$

En este caso el subíndice  $i$  toma valores desde 1 hasta  $m$ .

$$(1 + 2\lambda)W_{i,j+1} - \lambda W_{i+1,j+1} - \lambda W_{i-1,j+1} = W_{i,j} + kF_{i,j+1}$$

El sistema matricial resulta:

$$\begin{pmatrix} 1+2\lambda & -\lambda & 0 & 0 & 0 & 0 & \dots & 0 & 0 \\ -\lambda & 1+2\lambda & -\lambda & 0 & 0 & 0 & \dots & & \\ 0 & -\lambda & 1+2\lambda & -\lambda & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & 0 & -\lambda & 1+2\lambda & -\lambda & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & 0 & \dots & -2\lambda & 1+2\lambda \end{pmatrix} \begin{pmatrix} W_{1,j+1} \\ W_{2,j+1} \\ W_{3,j+1} \\ W_{4,j+1} \\ \vdots \\ W_{m,j+1} \end{pmatrix} = \begin{pmatrix} W_{1,j} \\ W_{2,j} \\ W_{3,j} \\ W_{4,j} \\ \vdots \\ W_{m,j} \end{pmatrix} + \begin{pmatrix} kF_{1,j+1} + \lambda a(t_{j+1}) \\ kF_{2,j+1} \\ kF_{3,j+1} \\ kF_{4,j+1} \\ \vdots \\ kF_{m,j+1} + 2h\lambda b(t_{j+1}) \end{pmatrix}$$



En forma simplificada:

$$\begin{cases} \mathbf{A}W_{j+1} = W_j + \mathbf{b}_j \\ W_0 = \{f(x_i)\}_{i=1\dots m} \end{cases}$$

### 1.5.3. Condiciones tipo Neumann-Dirichlett

En este caso sólo se dispone de información explícita para el extremo final de la barra.

$$W_{-1,j+1} = W_{1,j+1} - 2h\alpha(t_{j+1})$$

$$W_{m,j+1} = \mathbf{b}(t_{j+1}); \quad W_{i,0} = f(x_i)$$

En este caso el subíndice  $i$  toma valores desde  $0$  hasta  $m-1$  y el sistema queda como sigue:

$$\begin{pmatrix} 1+2\lambda & -2\lambda & 0 & 0 & 0 & 0 & \dots & 0 & 0 \\ -\lambda & 1+2\lambda & -\lambda & 0 & 0 & 0 & \dots & & \\ 0 & -\lambda & 1+2\lambda & -\lambda & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & 0 & -\lambda & 1+2\lambda & -\lambda & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & 0 & \dots & -\lambda & 1+2\lambda \end{pmatrix} \begin{pmatrix} W_{0,j+1} \\ W_{1,j+1} \\ W_{2,j+1} \\ W_{3,j+1} \\ \vdots \\ W_{m-1,j+1} \end{pmatrix} = \begin{pmatrix} W_{0,j} \\ W_{1,j} \\ W_{2,j} \\ W_{3,j} \\ \vdots \\ W_{m-1,j} \end{pmatrix} + \begin{pmatrix} kF_{0,j+1} - 2h\lambda\alpha(t_{j+1}) \\ kF_{1,j+1} \\ kF_{2,j+1} \\ kF_{3,j+1} \\ \vdots \\ kF_{m-1,j+1} + \lambda\mathbf{b}(t_{j+1}) \end{pmatrix}$$

Simplificando:

$$\begin{cases} \mathbf{A}W_{j+1} = W_j + \mathbf{b}_j \\ W_0 = \{f(x_i)\}_{i=0\dots m-1} \end{cases}$$



#### 1.5.4. Condiciones tipo Neumann-Neumann

En este caso no se conocen directamente ninguna de las distribuciones temporales de temperatura de los extremos de barra:

$$a(t_j + 1) = \frac{W_{1,j+1} - W_{-1,j+1}}{2h}; \quad b(t_j) = \frac{W_{m+1,j+1} - W_{m-1,j+1}}{2h}$$

$$W_{-1,j+1} = W_{1,j+1} - 2ha(t_j + 1)$$

$$W_{m+1,j+1} = W_{m-1,j+1} + 2hb(t_j + 1)$$

En este caso el subíndice  $i$  toma valores desde  $0$  hasta  $m$  y el sistema queda como sigue:

$$\begin{pmatrix} 1+2\lambda & -2\lambda & 0 & 0 & 0 & 0 & \dots & 0 & 0 \\ -\lambda & 1+2\lambda & -\lambda & 0 & 0 & 0 & \dots & & \\ 0 & -\lambda & 1+2\lambda & -\lambda & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & 0 & -\lambda & 1+2\lambda & -\lambda & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & 0 & \dots & -2\lambda & 1+2\lambda \end{pmatrix} \begin{pmatrix} W_{0,j+1} \\ W_{1,j+1} \\ W_{2,j+1} \\ W_{3,j+1} \\ \vdots \\ W_{m,j+1} \end{pmatrix} = \begin{pmatrix} W_{0,j} \\ W_{1,j} \\ W_{2,j} \\ W_{3,j} \\ \vdots \\ W_{m,j} \end{pmatrix} + \begin{pmatrix} kF_{0,j} - 2h\lambda a(t_{j+1}) \\ kF_{1,j} \\ kF_{2,j} \\ kF_{3,j} \\ \vdots \\ kF_{m,j} + 2h\lambda b(t_{j+1}) \end{pmatrix}$$

Simplificando:

$$\begin{cases} AW_{j+1} = W_j + bj \\ W_0 = \{f(x_i)\}_{i=0\dots m} \end{cases}$$



### 1.6. Método de Crank-Nicholson

Partimos de la ecuación del calor:

$$\frac{\partial u}{\partial t}(x, t) = c^2 \frac{\partial^2 u}{\partial x^2}(x, t) + F(x, t)$$

y con las siguientes expresiones:

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

$$\frac{\partial^2 u}{\partial x^2}(x, t + k/2) = \frac{u_{xx}(x, t + k) + u_{xx}(x, t)}{2}$$

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

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

En este método se toman los nodos:

$$(x, t) = (x_i, t_j + k/2)$$



Sustituyendo en la ecuación inicial:

$$\frac{\partial u}{\partial t}(x_i, t_j + k/2) = c^2 \frac{\partial^2 u}{\partial x^2}(x_i, t_j + k/2) + F(x_i, t_j + k/2)$$

Simplificando y eliminando errores de redondeo, las expresiones aproximadas son:

$$F(x_i, t_j + 1/2) \approx F(i, j + 1/2)$$

$$u(x_i, t_j + 1/2) \approx W(i, j + 1/2)$$

Sustituyendo de nuevo obtenemos:

$$\frac{W_{i,j+1} - W_{i,j}}{k} = \frac{c^2}{2h^2} [W_{i+1,j+1} - 2W_{i,j+1} + W_{i-1,j+1} - 2W_{i+1,j} - 2W_{i,j} + W_{i-1,j}] + F_{i,j+1/2}$$

Tomando el parámetro lambda como:

$$\lambda = \frac{c^2 K}{h^2}$$

Finalmente obtenemos:

$$(1 + \lambda)W_{i,j+1} - \frac{\lambda}{2}W_{i+1,j+1} - \frac{\lambda}{2}W_{i-1,j+1} = (1 - \lambda)W_{i,j+1} + \frac{\lambda}{2}W_{i+1,j} + \frac{\lambda}{2}W_{i-1,j} + kF_{i,j+1/2}$$



Esta expresión junto a las obtenidas a partir de las condiciones de contorno específicas nos permitirán definir un sistema matricial de ecuaciones que resuelva el problema.

### 1.6.1. Condiciones tipo Dirichlett-Dirichlett

Las condiciones de contorno tipo Dirichlett implican valores conocidos de temperatura en los extremos, de forma que:

$$W_{0,j+1} = a(t_j); \quad W_{m,j+1} = b(t_j)$$

La condición inicial:

$$W_{i,0} = f(x_i)$$

Con las expresiones obtenidas hasta el momento y dando valores a  $i$  desde 1 hasta  $m-1$  es posible construir el sistema matricial de ecuaciones que resuelva la ecuación:

$$\begin{pmatrix} 1+\lambda & -\lambda/2 & 0 & 0 & 0 & 0 & \dots & 0 & 0 \\ -\lambda/2 & 1+\lambda & -\lambda/2 & 0 & 0 & 0 & \dots & 0 & 0 \\ 0 & -\lambda/2 & 1+\lambda & -\lambda/2 & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & 0 & -\lambda/2 & 1+\lambda & -\lambda/2 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & 0 & \dots & -\lambda/2 & 1+\lambda \end{pmatrix} \begin{pmatrix} W_{1,j+1} \\ W_{2,j+1} \\ W_{3,j+1} \\ W_{4,j+1} \\ \vdots \\ W_{m-1,j+1} \end{pmatrix} =$$

$$\begin{pmatrix} 1-\lambda & \lambda/2 & 0 & 0 & 0 & 0 & \dots & 0 & 0 \\ \lambda/2 & 1-\lambda & \lambda/2 & 0 & 0 & 0 & \dots & 0 & 0 \\ 0 & \lambda/2 & 1-\lambda & \lambda/2 & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & 0 & \lambda/2 & 1-\lambda & \lambda/2 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & 0 & \dots & \lambda/2 & 1-\lambda \end{pmatrix} \begin{pmatrix} W_{1,j} \\ W_{2,j} \\ W_{3,j} \\ W_{4,j} \\ \vdots \\ W_{m-1,j} \end{pmatrix} + \begin{pmatrix} kF_{1,j+1/2} + \frac{\lambda}{2} [a(t_j) + a(t_{j+1})] \\ kF_{2,j+1/2} \\ kF_{3,j+1/2} \\ \vdots \\ kF_{m-2,j+1/2} \\ kF_{m-1,j/2} + \frac{\lambda}{2} [b(t_j) + b(t_{j+1})] \end{pmatrix}$$



Simplificando, el sistema puede expresarse como:

$$\begin{cases} \mathbf{A}W_{j+1} = \mathbf{B}W_j + \mathbf{b}_j \\ W_0 = \{f(x_i)\}_{i=1\dots m-1} \end{cases}$$

### 1.6.2. Condiciones tipo Dirichlett-Neumann

Se conocen directamente los valores de la distribución temporal de temperaturas en el extremo inicial de la barra pero no se dispone de información explícita del parámetro en el extremo final:

$$W_{m+1,j+1} = W_{m-1,j+1} + 2hb(t_j + 1)$$

Las condiciones inicial y de contorno directa:

$$W_{0,j+1} = a(t_j + 1); \quad W_{i,0} = f(x_i)$$

En este caso el subíndice  $i$  toma valores desde 1 hasta  $m$ .

$$(1 + \lambda)W_{i,j+1} - \frac{\lambda}{2}W_{i+1,j+1} - \frac{\lambda}{2}W_{i-1,j+1} = (1 - \lambda)W_{i,j+1} + \frac{\lambda}{2}W_{i+1,j+1} + \frac{\lambda}{2}W_{i-1,j+1} + kF_{i,j} + 1/2$$

El sistema matricial resulta:

$$\begin{pmatrix} 1+\lambda & -\lambda/2 & 0 & 0 & 0 & 0 & \dots & 0 & 0 \\ -\lambda/2 & 1+\lambda & -\lambda/2 & 0 & 0 & 0 & \dots & 0 & 0 \\ 0 & -\lambda/2 & 1+\lambda & -\lambda/2 & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & 0 & -\lambda/2 & 1+\lambda & -\lambda/2 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & 0 & \dots & -\lambda & 1+\lambda \end{pmatrix} \begin{pmatrix} W_{1,j+1} \\ W_{2,j+1} \\ W_{3,j+1} \\ W_{4,j+1} \\ \vdots \\ W_{m,j+1} \end{pmatrix} =$$



$$\begin{pmatrix} 1-\lambda & \lambda/2 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 \\ \lambda/2 & 1-\lambda & \lambda/2 & 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & \lambda/2 & 1-\lambda & \lambda/2 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 0 & \lambda/2 & 1-\lambda & \lambda/2 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & 0 & \cdots & \lambda & 1-\lambda \end{pmatrix} \begin{pmatrix} W_{1,j} \\ W_{2,j} \\ W_{3,j} \\ W_{4,j} \\ \vdots \\ W_{m,j} \end{pmatrix} + \begin{pmatrix} kF_{1,j+1/2} + \frac{\lambda}{2}(a[t_j] + a[t_{j+1}]) \\ kF_{2,j+1/2} \\ kF_{3,j+1/2} \\ \vdots \\ kF_{m-2,j+1/2} \\ kF_{m,j+1/2} + \lambda h(b[t_j] + b[t_{j+1}]) \end{pmatrix}$$

En forma simplificada:

$$\begin{cases} \mathbf{A}W_{j+1} = \mathbf{B}W_j + \mathbf{b}_j \\ W_0 = \{f(x_i)\}_{i=1\dots m} \end{cases}$$

### 1.6.3. Condiciones tipo Neumann-Dirichlett

En este caso sólo se dispone de información explícita para el extremo final de la barra.

$$W_{-1,j+1} = W_{1,j+1} - 2ha(t_j + 1)$$

$$W_{m,j+1} = b(t_j + 1); \quad W_{i,0} = f(x_i)$$

En este caso el subíndice  $i$  toma valores desde  $0$  hasta  $m-1$  y el sistema queda como sigue:

$$\begin{pmatrix} 1+\lambda & -\lambda & 0 & 0 & 0 & 0 & \cdots & 0 & 0 \\ -\lambda/2 & 1+\lambda & -\lambda/2 & 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & -\lambda/2 & 1+\lambda & -\lambda/2 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 0 & -\lambda/2 & 1+\lambda & -\lambda/2 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & 0 & \cdots & -\lambda/2 & 1+\lambda \end{pmatrix} \begin{pmatrix} W_{0,j+1} \\ W_{1,j+1} \\ W_{2,j+1} \\ W_{3,j+1} \\ \vdots \\ W_{m-1,j+1} \end{pmatrix} =$$



$$\begin{pmatrix} 1-\lambda & \lambda & 0 & 0 & 0 & 0 & \dots & 0 & 0 \\ \lambda/2 & 1-\lambda & \lambda/2 & 0 & 0 & 0 & \dots & 0 & 0 \\ 0 & \lambda/2 & 1-\lambda & \lambda/2 & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & 0 & \lambda/2 & 1-\lambda & \lambda/2 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & 0 & \dots & \lambda/2 & 1-\lambda \end{pmatrix} \begin{pmatrix} W_{0,j} \\ W_{1,j} \\ W_{2,j} \\ W_{3,j} \\ \vdots \\ W_{m-1,j} \end{pmatrix} + \begin{pmatrix} kF_{0,j+1/2} - \lambda h(a(t_j) + a(t_{j+1})) \\ kF_{2,j+1/2} \\ kF_{3,j+1/2} \\ \vdots \\ kF_{m-2,j+1/2} \\ kF_{m-1,j+1/2} + \lambda/2(b(t_j) + b(t_{j+1})) \end{pmatrix}$$

Simplificando:

$$\begin{cases} AW_{j+1} = BW_j + bj \\ W_0 = \{f(x_i)\}_{i=0\dots m-1} \end{cases}$$

#### 1.6.4. Condiciones tipo Neumann-Neumann

En este caso no se conocen directamente ninguna de las distribuciones temporales de temperatura de los extremos de barra:

$$a(t_j + 1) = \frac{W_{1,j+1} - W_{-1,j+1}}{2h}; \quad b(t_j) = \frac{W_{m+1,j+1} - W_{m-1,j+1}}{2h}$$

$$W_{-1,j+1} = W_{1,j+1} - 2ha(t_j + 1)$$

$$W_{m+1,j+1} = W_{m-1,j+1} + 2hb(t_j + 1)$$

En este caso el subíndice  $i$  toma valores desde  $0$  hasta  $m$  y el sistema queda como sigue:



$$\begin{pmatrix} 1+\lambda & -\lambda & 0 & 0 & 0 & 0 & \dots & 0 & 0 \\ -\lambda/2 & 1+\lambda & -\lambda/2 & 0 & 0 & 0 & \dots & 0 & 0 \\ 0 & -\lambda/2 & 1+\lambda & -\lambda/2 & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & 0 & -\lambda/2 & 1+\lambda & -\lambda/2 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & -\lambda/2 \\ 0 & 0 & 0 & 0 & 0 & 0 & \dots & -\lambda & 1+\lambda \end{pmatrix} \begin{pmatrix} W_{0,j+1} \\ W_{1,j+1} \\ W_{2,j+1} \\ W_{3,j+1} \\ \vdots \\ W_{m,j+1} \end{pmatrix} =$$

$$\begin{pmatrix} 1-\lambda & \lambda & 0 & 0 & 0 & 0 & \dots & 0 & 0 \\ \lambda/2 & 1-\lambda & \lambda/2 & 0 & 0 & 0 & \dots & 0 & 0 \\ 0 & \lambda/2 & 1-\lambda & \lambda/2 & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & 0 & \lambda/2 & 1-\lambda & \lambda/2 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & 0 & \dots & \lambda & 1-\lambda \end{pmatrix} \begin{pmatrix} W_{0,j} \\ W_{1,j} \\ W_{2,j} \\ W_{3,j} \\ \vdots \\ W_{m,j} \end{pmatrix} + \begin{pmatrix} kF_{0,j+1/2} - \lambda h(a(t_j) + a(t_{j+1})) \\ kF_{1,j+1/2} \\ kF_{2,j+1/2} \\ \vdots \\ kF_{m-1,j+1/2} \\ kF_{m,j+1/2} + \lambda h(b(t_j) + b(t_{j+1})) \end{pmatrix}$$

Simplificando:

$$\begin{cases} \mathbf{A}W_{j+1} = \mathbf{B}W_j + \mathbf{b}_j \\ \mathbf{W}_0 = \{f(x_i)\}_{i=0\dots m} \end{cases}$$



## 2. La Ecuación de Ondas

---

La Ecuación de Ondas es una importante E.D. en derivadas parciales que describe una variedad de ondas, como las ondas sonoras, las ondas de luz y las ondas de agua. Es interesante en varios campos como la acústica, el electromagnetismo y la dinámica de fluidos. Históricamente, el problema de una cuerda vibrante como lo es en algunos instrumentos musicales fue estudiado por Jean le Rond d'Alembert, Leonhard Euler, Daniel Bernoulli y Joseph-Louis Lagrange.

La ecuación de onda es el ejemplo prototipo de una ecuación diferencial parcial hiperbólica. En su forma más elemental, la ecuación de onda hace referencia a un desplazamiento escalar  $u$  que satisface:

$$\frac{\partial^2 u}{\partial t^2} = c^2 \Delta u = c^2 \left( \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2} \right)$$

donde  $c$  es una constante equivalente a la velocidad de propagación de la onda y  $\Delta u$  el Laplaciano de  $u$ .

Para una onda sonora en el aire a 20°C, la velocidad de propagación es aproximadamente de 343 m/s. Para una cuerda vibrante, la velocidad puede variar mucho dependiendo de la densidad lineal de la cuerda y su tensión. Para un resorte de espiral puede ser tan lento como un metro por segundo.

### 2.1. Consideraciones previas y condiciones de contorno

Al igual que en la Ecuación del Calor, en la asignatura hemos tratado la Ecuación de Ondas en una sola dimensión. La terminología empleada ha sido:

$$\frac{\partial^2 u}{\partial t^2}(x, t) = c^2 \cdot \frac{\partial^2 u}{\partial x^2}(x, t) + F(x, t)$$

Simplificando:



$$u_{tt} = c^2 u_{xx} + F$$

Las condiciones de contorno:

$$u(0, t) = a(t) \quad \text{ó} \quad \frac{\partial u}{\partial x}(0, t) = a(t), \quad u(L, t) = b(t) \quad \text{ó} \quad \frac{\partial u}{\partial x}(L, t) = b(t), \quad t > 0$$

En este caso las condiciones iniciales son posición y velocidad:

$$u(x, 0) = f(x), \quad \frac{\partial u}{\partial t}(x, 0) = g(x), \quad 0 < x < L$$

Donde:

- $u(x, t)$  es la posición de la onda en un punto  $x$  y un instante  $t$
- $c$  es la velocidad de propagación de la onda
- $F(x, t)$  es la perturbación externa para un punto  $x$  y un instante  $t$
- $u(x, 0)$  es la distribución inicial de posiciones en la onda
- $u_t(x, 0)$  es la distribución inicial de velocidades en la onda
- $u(0, t)$  es la distribución temporal de posiciones en el extremo inicial de la onda
- $u(L, t)$  es la distribución temporal de posiciones en el extremo final de la onda

Las condiciones de contorno presentan la misma casuística que en la Ecuación del Calor: tipo DD, DN, ND o NN.

## *2.2. Resolución*

Partimos de las expresiones:



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

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

Eliminando los errores de truncamiento y sustituyendo en la ecuación original:

$$u(x_i, t_j) \approx W(i, j) \quad F(x_i, t_j) \approx F(i, j)$$

$$\frac{W_{i,j+1} - 2W_{i,j} + W_{i,j-1}}{k^2} = c^2 \frac{W_{i+1,j} - 2W_{i,j} + W_{i-1,j}}{h^2} + F(i, j)$$

Reuniendo en un solo parámetro  $\lambda$  las constantes  $c$ ,  $k$  y  $h$ , la ecuación resulta:

$$W_{i,j+1} = 2(1 - \lambda^2)W_{i,j} + \lambda^2 W_{i+1,j} + \lambda^2 W_{i-1,j} - W_{i,j-1} + k^2 F_{i,j}$$

En este momento será necesario definir las condiciones de contorno para formar el sistema matricial de ecuaciones.

### 2.2.1. Condiciones tipo Dirichlett-Dirichlett

Se suponen:

$$W_{0,j} = a(t_j); \quad W_{m,j} = b(t_j)$$



El subíndice  $i$  toma valores desde 1 hasta  $m-1$  y la ecuación resulta:

$$W_{i,j+1} = 2(1-\lambda^2)W_{i,j} + \lambda^2 W_{i+1,j} + \lambda^2 W_{i-1,j} - W_{i,j-1} + k^2 F_{i,j}$$

El sistema matricial:

$$\begin{pmatrix} W_{1,j+1} \\ W_{2,j+1} \\ W_{3,j+1} \\ W_{4,j+1} \\ \vdots \\ W_{m-1,j+1} \end{pmatrix} = \begin{pmatrix} 2(1-\lambda^2) & \lambda^2 & 0 & 0 & 0 & 0 & \dots & 0 & 0 \\ \lambda^2 & 2(1-\lambda^2) & \lambda^2 & 0 & 0 & 0 & \dots & & \\ 0 & \lambda^2 & 2(1-\lambda^2) & \lambda^2 & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & 0 & \lambda^2 & 2(1-\lambda^2) & \lambda^2 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & 0 & \dots & \lambda^2 & 2(1-\lambda^2) \end{pmatrix} \begin{pmatrix} W_{1,j} \\ W_{2,j} \\ W_{3,j} \\ W_{4,j} \\ \vdots \\ W_{m-1,j} \end{pmatrix} - \begin{pmatrix} W_{1,j-1} \\ W_{2,j-1} \\ W_{3,j-1} \\ W_{4,j-1} \\ \vdots \\ W_{m-1,j-1} \end{pmatrix} + \begin{pmatrix} k^2 F_{1,j} + \lambda^2 a(t_j) \\ k f_{2,j} \\ k f_{3,j} \\ k f_{4,j} \\ \vdots \\ k^2 F_{m-1,j} + \lambda^2 b(t_j) \end{pmatrix}$$

Simplificando:

$$\begin{cases} W_{j+1} = AW_j - W_{j-1} + bj \\ W_0 = \{f(x_i)\}_{i=1\dots m-1} \\ W_1 = (1-\lambda^2)f(x_i) + \frac{\lambda^2}{2} [(f(x_i+1) + f(x_i-1))] + kg(x_i) \frac{k^2}{2} F(x_i,0)_{i=1\dots m-1} \end{cases}$$

### 2.2.2. Condiciones tipo Dirichlett-Neumann

Se tiene:

$$b(t_j) = \frac{W_{m+1,j} - W_{m-1,j}}{2h}$$



$$W_{m+1,j} = W_{m-1,j} + 2hb(t_j)$$

El subíndice  $i$  va desde 1 hasta  $m$  y el sistema resulta:

$$\begin{pmatrix} W_{1,j+1} \\ W_{2,j+1} \\ W_{3,j+1} \\ W_{4,j+1} \\ \vdots \\ W_{m,j+1} \end{pmatrix} = \begin{pmatrix} 2(1-\lambda^2) & \lambda^2 & 0 & 0 & 0 & 0 & \dots & 0 & 0 \\ \lambda^2 & 2(1-\lambda^2) & \lambda^2 & 0 & 0 & 0 & \dots & & \\ 0 & \lambda^2 & 2(1-\lambda^2) & \lambda^2 & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & 0 & \lambda^2 & 2(1-\lambda^2) & \lambda^2 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & 0 & \dots & 2\lambda^2 & 2(1-\lambda^2) \end{pmatrix} \begin{pmatrix} W_{1,j} \\ W_{2,j} \\ W_{3,j} \\ W_{4,j} \\ \vdots \\ W_{m,j} \end{pmatrix} - \begin{pmatrix} W_{1,j-1} \\ W_{2,j-1} \\ W_{3,j-1} \\ W_{4,j-1} \\ \vdots \\ W_{m,j-1} \end{pmatrix} + \begin{pmatrix} k^2 F_{1,j} + \lambda^2 a(t_j) \\ k f_{2,j} \\ k f_{3,j} \\ k f_{4,j} \\ \vdots \\ k^2 F_{m,j} + 2\lambda^2 b(t_j) \end{pmatrix}$$

Simplificando:

$$\begin{cases} W_{j+1} = AW_j - W_{j-1} + bj \\ W_0 = \{f(x_i)\}_{i=1\dots m} \\ W_1 = (1 - \lambda^2)f(x_i) + \frac{\lambda^2}{2} [(f(x_i + 1) + f(x_i - 1)) + kg(x_i)] - \frac{k^2}{2} F(x_i, 0)_{i=1\dots m} \end{cases}$$

### 2.2.3. Condiciones tipo Neumann-Dirichlett

En este caso se conocen directamente valores en el extremo final, luego:

$$a(t_j) = \frac{W_{1,j} - W_{-1,j}}{2h}$$

$$W_{-1,j} = W_{1,j} - 2ha(t_j)$$



Dando valores a  $i$  desde  $0$  hasta  $m-1$ , la ecuación resulta:

$$\begin{pmatrix} 1+\lambda & -\lambda & 0 & 0 & 0 & 0 & \dots & 0 & 0 \\ -\lambda/2 & 1+\lambda & -\lambda/2 & 0 & 0 & 0 & \dots & 0 & 0 \\ 0 & -\lambda/2 & 1+\lambda & -\lambda/2 & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & 0 & -\lambda/2 & 1+\lambda & -\lambda/2 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & 0 & \dots & -\lambda/2 & 1+\lambda \end{pmatrix} \begin{pmatrix} W_{0,j+1} \\ W_{1,j+1} \\ W_{2,j+1} \\ W_{3,j+1} \\ \vdots \\ W_{m-1,j+1} \end{pmatrix} =$$

$$\begin{pmatrix} 1-\lambda & \lambda & 0 & 0 & 0 & 0 & \dots & 0 & 0 \\ \lambda/2 & 1-\lambda & \lambda/2 & 0 & 0 & 0 & \dots & 0 & 0 \\ 0 & \lambda/2 & 1-\lambda & \lambda/2 & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & 0 & \lambda/2 & 1-\lambda & \lambda/2 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & 0 & \dots & \lambda/2 & 1-\lambda \end{pmatrix} \begin{pmatrix} W_{0,j} \\ W_{1,j} \\ W_{2,j} \\ W_{3,j} \\ \vdots \\ W_{m-1,j} \end{pmatrix} + \begin{pmatrix} kF_{0,j+1/2} - \lambda h(a(t_j) + a(t_{j+1})) \\ kF_{2,j+1/2} \\ kF_{3,j+1/2} \\ \vdots \\ kF_{m-2,j+1/2} \\ kF_{m-1,j+1/2} + \lambda/2(b(t_j) + b(t_{j+1})) \end{pmatrix}$$

Simplificando:

$$\begin{cases} W_{j+1} = AW_j - W_{j-1} + bj \\ W_0 = \{f(x_i)\}_{i=1\dots m-1} \\ W_1 = (1 - \lambda^2)f(x_i) + \frac{\lambda^2}{2} [(f(x_i + 1) + f(x_i - 1)) + kg(x_i)] \frac{k^2}{2} F(x_i, 0)_{i=1\dots m-1} \end{cases}$$

#### 2.2.4. Condiciones tipo Neumann-Neumann

Se tiene:

$$a(t_j) = \frac{W_{1,j} - W_{-1,j}}{2h}; \quad b(t_j) = \frac{W_{m+1,j} - W_{m-1,j}}{2h}$$

$$W_{-1,j} = W_{1,j} - 2ha(t_j)$$



$$W_{m+1,j} = W_{m-1,j} + 2hb(t_j)$$

El subíndice  $i$  va desde  $0$  hasta  $m$  y la ecuación resulta:

$$\begin{pmatrix} W_{0,j+1} \\ W_{1,j+1} \\ W_{2,j+1} \\ W_{3,j+1} \\ \vdots \\ W_{m,j+1} \end{pmatrix} = \begin{pmatrix} 2(1-\lambda^2) & \lambda^2 & 0 & 0 & 0 & 0 & \dots & 0 & 0 \\ \lambda^2 & 2(1-\lambda^2) & \lambda^2 & 0 & 0 & 0 & \dots & & \\ 0 & \lambda^2 & 2(1-\lambda^2) & \lambda^2 & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & 0 & \lambda^2 & 2(1-\lambda^2) & \lambda^2 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & 0 & \dots & \lambda^2 & 2(1-\lambda^2) \end{pmatrix} \begin{pmatrix} W_{0j} \\ W_{1j} \\ W_{2j} \\ W_{3j} \\ \vdots \\ W_{mj} \end{pmatrix} - \begin{pmatrix} W_{0j-1} \\ W_{1j-1} \\ W_{2j-1} \\ W_{3j-1} \\ \vdots \\ W_{mj-1} \end{pmatrix} + \begin{pmatrix} k^2 F_{0j} - 2h\lambda^2 a(t_j) \\ kf_{2j} \\ kf_{3j} \\ kf_{4j} \\ \vdots \\ k^2 F_{mj} + 2h\lambda^2 b(t_j) \end{pmatrix}$$

Simplificando:

$$\begin{cases} W_{j+1} = AW_j - W_{j-1} + bj \\ W_0 = \{f(x_i)\}_{i=0\dots m} \\ W_1 = (1-\lambda^2)f(x_i) + \frac{\lambda^2}{2}[(f(x_i+1) + f(x_i-1))] + kg(x_i) \frac{k^2}{2} F(x_i,0)_{i=0\dots m} \end{cases}$$



### 3. Algunos ejercicios resueltos con MATLAB 7.0

---

#### 3.1. Ecuación del Calor, método Progresivo y condiciones DD

Dada la siguiente ecuación para  $x$  en  $[0,1]$ :

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

Con las condiciones de contorno:

$$u(0, t) = u(1, t) = 0 \quad u(x, 0) = \sin(\pi x)$$

Se pide:

- Calcular y representar  $u(x, 0.5)$  tomando  $N=1000$  y  $N=50$ .
- Presentar una matriz que contenga los diferentes nodos, la distribución inicial de temperaturas, la distribución final de temperaturas, la distribución real de temperaturas y el error cometido si la solución exacta de la ecuación es:

$$u(x, t) = e^{-(\pi^2 t)} \sin(\pi x)$$

- Dibujar una malla con la distribución de temperaturas de la barra en el intervalo  $t=(0, 0.5)$

#### SOLUCIÓN a)

Para resolver el apartado se emplea el script del método Progresivo con condiciones tipo DD:



### function ProgresivoDD(c,L,m,N,T)

#### % Entradas:

% c=coeficiente de difusión  
% L=longitud de la barra  
% m=nº de divisiones espaciales  
% N=nº de divisiones temporales  
% T=tiempo final

#### % Salidas:

% W=vector distribución de temperaturas en el instante T  
% WW=matriz de temperaturas en (0,T)

h=L/m;  
K=T/N;  
la=c^2\*(K/h^2)

#### % Cálculo de A y B

A=tridiagonal(1,0,m-1);  
B=tridiagonal(1-2\*la,la,m-1);

#### % Cálculo de Wo

for i=1:m-1  
    W(i)=f(i\*h);  
end  
W=W';  
Wo=W;

#### % Constitución de la matriz WW

WW(1,:)= [a(0);W;bb(0)]';

#### % Cálculo del vector solución real

for i=1:m+1  
    x(i)=(i-1)\*h;  
    U(i)=u(x(i),T);  
end

#### % Calculo de W1.....Wn

for j=0:N-1  
    % Cálculo de bj  
    for i=1:m-1  
        b(i)=K\*FF(i\*h,(j+1)\*K);



```
end
b(1)=b(1)+la*(a(j+1)*K);
b(m-1)=b(m-1)+la*(bb(j+1)*K);
W=A\ (B*W+b');
% Constitución de la matriz WW
WW(j+2,:)= [a((j+1)*K);W;bb((j+1)*K)]';
end

% Creación del vector W en el instante de tiempo requerido
W=[a(T);W;bb(T)];
```

Y se definen las funciones condición de contorno, inicial y perturbación externa:

```
function y=a(t)
y=0;
```

```
function y=bb(t)
y=0;
```

```
function y=f(x)
y=sin(pi*x);
```

```
function y=FF(x,t)
y=0;
```

```
function y=u(x,t)
y=exp(-pi^2*t)*sin(pi*x);
```

Escribiendo la siguiente orden:

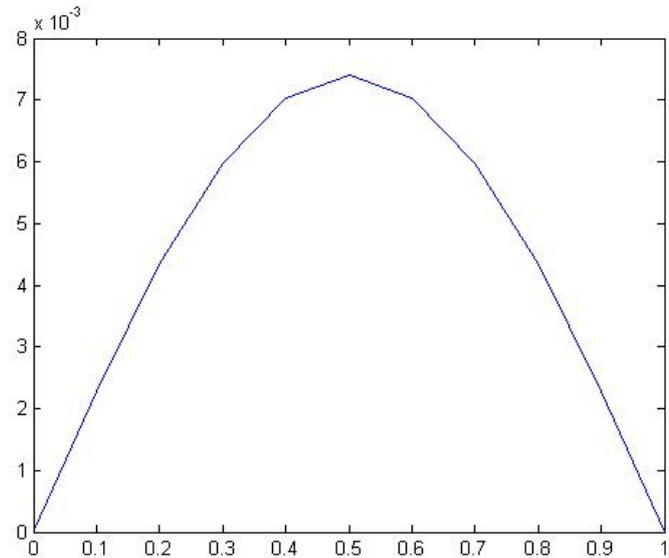
*[W]=ProgresivoDD(1,1,10,1000,1/2)*

Se obtiene el vector  $\mathbf{u}(\mathbf{x},0.5)$  para  $N=1000$ :



W =

```
0
0.00228652078658
0.00434922098744
0.00598618913525
0.00703718738226
0.00739933669733
0.00703718738226
0.00598618913525
0.00434922098744
0.00228652078658
0
```



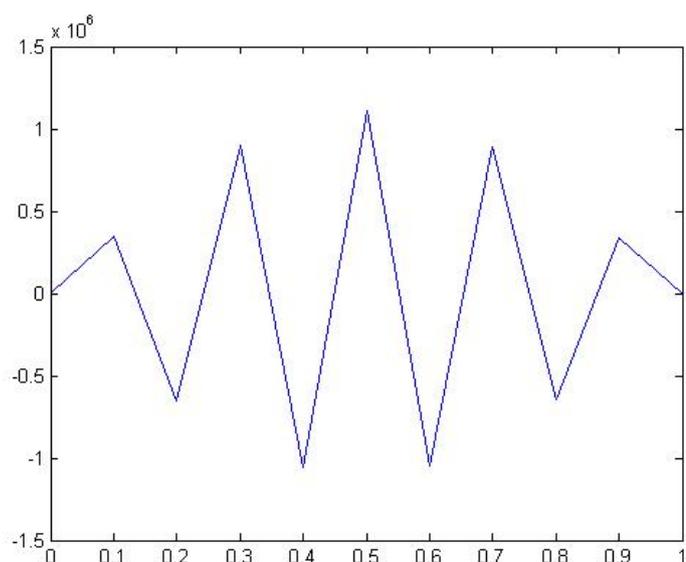
Tomando ahora N=50:

*[W]=ProgresivoDD(1,1,10,50,1/2)*

Se obtiene:

W =

```
1.0e+006 *
0
0.34540324073161
-0.65593076045493
0.90052692618896
-1.05524767113756
1.10560546497161
-1.04773690954975
0.88837425865892
-0.64377809292543
0.33789247914466
0
```





La diferencia entre los resultados obtenidos es debida al valor del parámetro lambda. Cuando lambda es inferior a 0,5 (para  $N=1000$  vale 0.05) el método es estable. En el caso  $N=50$  su valor es de 1 y la aproximación no es válida.

### SOLUCIÓN b)

La siguiente implementación en el script permite obtener la matriz buscada, que denominaremos [AA]:

```
% x=longitud de la barra
AA(:,1)=x';
% Wo=Temperatura en el instante inicial
AA(:,2)=WW(1,:);
% W=Temperatura en el instante definido T
AA(:,3)=W;
% U=Temperatura real
AA(:,4)=U';
% Error
AA(:,5)=abs(W-U');
E=norm(W-U');
```

Escribiendo ahora:

```
[AA]=ProgresivoDD(1,1,10,1000,1/2)
```

Se obtiene la matriz buscada:

```
AA =
      0      0      0      0      0
0.100000000000000 0.30901699437495 0.00228652078658 0.00222241417851 0.00006410660807
0.200000000000000 0.58778525229247 0.00434922098744 0.00422728297276 0.00012193801468
0.300000000000000 0.80901699437495 0.00598618913525 0.00581835585643 0.00016783327882
0.400000000000000 0.95105651629515 0.00703718738226 0.00683988752999 0.00019729985227
0.500000000000000 1.00000000000000 0.00739933669733 0.00719188335583 0.00020745334151
0.600000000000000 0.95105651629515 0.00703718738226 0.00683988752999 0.00019729985227
```



0.700000000000000	0.80901699437495	0.00598618913525	0.00581835585643	0.00016783327882
0.800000000000000	0.58778525229247	0.00434922098744	0.00422728297276	0.00012193801468
0.900000000000000	0.30901699437495	0.00228652078658	0.00222241417851	0.00006410660807
1.000000000000000	0	0	0.000000000000000	0.000000000000000

Nodos

W0

W1

$u(x,1)$

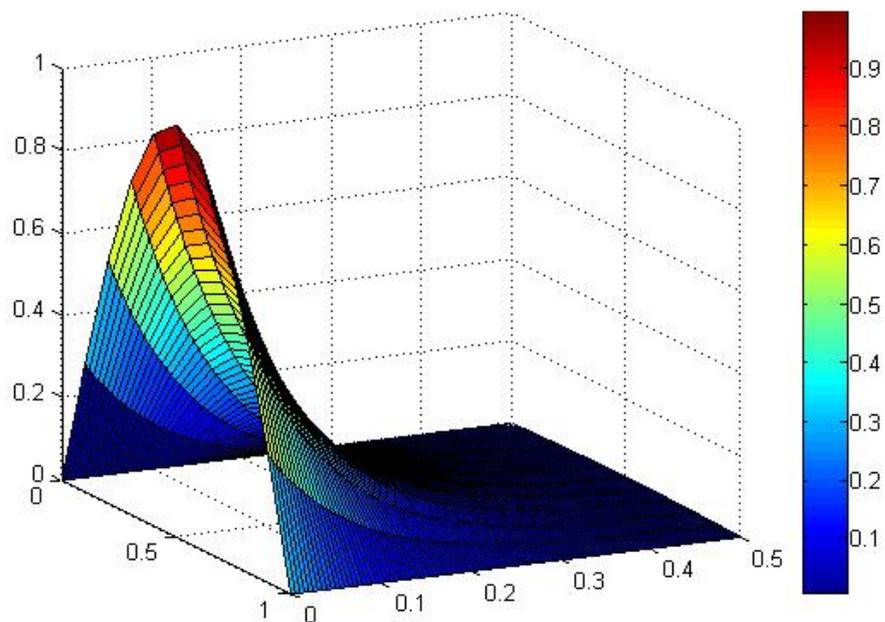
Error

### SOLUCIÓN c)

La malla con la evolución temporal de la temperatura se obtiene escribiendo:

```
[WW,L,m,N,T]=ProgresivoDD(1,1,10,100,1/2)
figure
r=linspace(0,L,m+1);
s=linspace(0,T,N+1);
[rr,ss]=meshgrid(r,s);
surf(rr,ss,WW);
```

Y resulta:





### 3.2. Ecuación del Calor, método Regresivo y condiciones NN

Dada la siguiente ecuación para  $x$  en  $(0, \pi)$ :

$$\frac{\partial u}{\partial t}(x, t) = \frac{\partial^2 u}{\partial x^2}(x, t) + 2e^t \sin(x)$$

Con las condiciones de contorno:

$$u_x(0, t) = e^t \quad u_x(\pi, t) = -e^t \quad u(x, 0) = \sin(x)$$

Se pide:

- Calcular y representar  $u(x, 1)$ . Graficar el vector junto a  $u(x, 0)$  y la solución real:

$$u(x, t) = e^t \sin x$$

- Calcular y representar  $u(3\pi/10, t)$ ,  $u(\pi/2, t)$  y  $u(9\pi/10, t)$  para  $t$  en  $(0, 1)$
- Dibujar una malla con la distribución de temperaturas de la barra en el intervalo de tiempo  $(0, 1)$

#### SOLUCIÓN a)

El script del método Regresivo con condiciones NN:

```
function RegresivoNN(c,L,m,N,T)
% Entradas:
% c=coeficiente de difusión
% L=longitud de la barra
```



```
% m=nº de divisiones espaciales
% N=nº de divisiones temporales
% T=tiempo final
% Salidas:
% W=vector distribución de temperaturas en el instante T
% WW=matriz de temperaturas en (0,T)

h=L/m;
k=T/N;
la=c^2*k/h^2;

% Cálculo de la matriz A
A=tridiagonal(1+2*la,-la,m+1);
A(m+1,m)=-(2*la);
A(1,2)=-(2*la);

% Cálculo de la matriz B (identidad)
B=tridiagonal(1,0,m+1);

% Cálculo de Wo y la solución real
for i=0:m
    x(i+1)=i*h;
    % Creación del vector longitud de la barra
    ejex=[x'];
    W(i+1)=ff(i*h);
    U(i+1)=u(x(i+1),T);
end
W=W';
Wo=W;
U=U;

% Constitución de la matriz WW
WW(1,:)=W';

%Cálculo de W1,...,WN
for j=0:N-1
    % Creación del vector tiempo
    J(1,j+1)=j*k;
    tiempo=[J';T];
    % Cálculo de bj
    for i=0:m
```



```
b(i+1)=k*gg(i*h,(j+1)*k);  
end  
b(1)=b(1)-2*la*h*aa((j+1)*k);  
b(m+1)=b(m+1)+2*la*h*bb((j+1)*k);  
W=A\ (B*W+b');  
% Constitución de la matriz WW  
WW(j+2,:)= [W]';  
end
```

Y se definen las funciones condición de contorno, inicial y perturbación externa:

```
function y=aa(t)  
y=exp(t);
```

```
function y=bb(t)  
y=-exp(t);
```

```
function y=ff(x)  
y=sin(x);
```

```
function y=gg(x,t)  
y=2*exp(t)*sin(x);
```

```
function y=u(x,t)  
y=exp(t)*sin(x);
```

Escribiendo la llamada al script:

*[U,W,Wo,ejex]=RegresivoNN(1,pi,10,1000,1)*

Se obtienen los resultados pedidos y se crean las variables necesarias para dibujar un gráfico comparativo. Para representar la información obtenida:



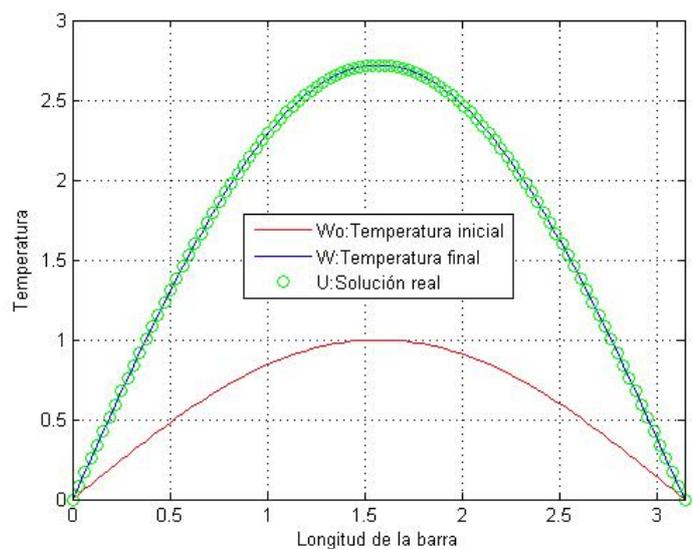
```
hold on  
figure  
plot(ejex, Wo, 'r', ejex, W, 'b', ejex, U, 'g o')  
axis([0 pi 0 3])  
grid
```

```
legend('Wo: Temperatura inicial', 'W: Temperatura final', 'U: Solución  
real', 'Location', 'Best')
```

```
xlabel('Longitud de la barra')  
ylabel('Temperatura')
```

Finalmente:

$U =$	$W =$
0	-0.0304
0.8400	0.8218
1.5978	1.5888
2.1991	2.1966
2.5852	2.5865
2.7183	2.7208
2.5852	2.5865
2.1991	2.1966
1.5978	1.5888
0.8400	0.8218
0.0000	-0.0304



### SOLUCIÓN b)

Para obtener las evoluciones previstas es necesario llamar al programa base e identificar en la matriz solución WW las columnas correspondientes:

```
[WW, tiempo]=RegresivoNN(1, pi, 10, 1000, 1)
```



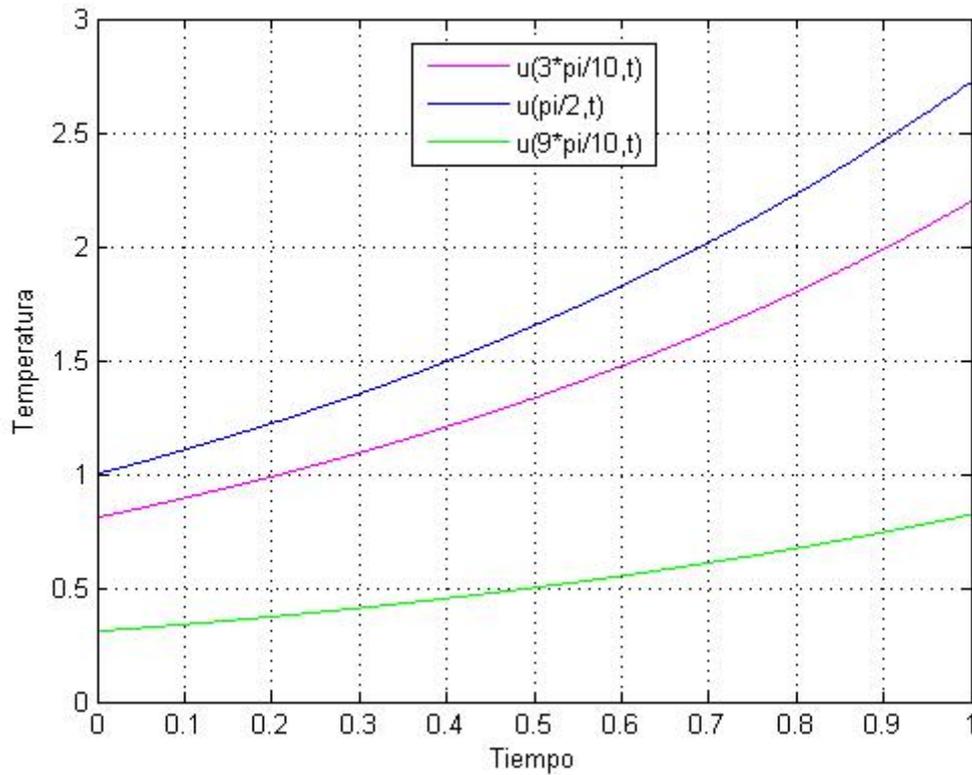
Recortando, el resultado:

*WW* =

0	0.3090	0.5878	0.8090	0.9511	1.0000	0.9511	0.8090	0.5878	0.3090	0.0000
-0.0001	0.3093	0.5884	0.8098	0.9520	1.0010	0.9520	0.8098	0.5884	0.3093	-0.0001
-0.0002	0.3096	0.5890	0.8107	0.9530	1.0020	0.9530	0.8107	0.5890	0.3096	-0.0002
-0.0003	0.3099	0.5896	0.8115	0.9539	1.0030	0.9539	0.8115	0.5896	0.3099	-0.0003
-0.0004	0.3103	0.5902	0.8123	0.9549	1.0040	0.9549	0.8123	0.5902	0.3103	-0.0004
-0.0005	0.3106	0.5908	0.8131	0.9559	1.0051	0.9559	0.8131	0.5908	0.3106	-0.0005
-0.0006	0.3109	0.5914	0.8139	0.9568	1.0061	0.9568	0.8139	0.5914	0.3109	-0.0006
-0.0007	0.3112	0.5919	0.8147	0.9578	1.0071	0.9578	0.8147	0.5919	0.3112	-0.0007
-0.0008	0.3115	0.5925	0.8156	0.9588	1.0081	0.9588	0.8156	0.5925	0.3115	-0.0008
-0.0009	0.3118	0.5931	0.8164	0.9597	1.0091	0.9597	0.8164	0.5931	0.3118	-0.0009
-0.0009	0.3121	0.5937	0.8172	0.9607	1.0101	0.9607	0.8172	0.5937	0.3121	-0.0009
-0.0299	0.8121	1.5699	2.1704	2.5557	2.6884	2.5557	2.1704	1.5699	0.8121	-0.0299
-0.0300	0.8129	1.5715	2.1726	2.5582	2.6911	2.5582	2.1726	1.5715	0.8129	-0.0300
-0.0300	0.8137	1.5730	2.1748	2.5608	2.6938	2.5608	2.1748	1.5730	0.8137	-0.0300
-0.0300	0.8145	1.5746	2.1769	2.5633	2.6964	2.5633	2.1769	1.5746	0.8145	-0.0300
-0.0301	0.8153	1.5762	2.1791	2.5659	2.6991	2.5659	2.1791	1.5762	0.8153	-0.0301
-0.0301	0.8162	1.5777	2.1813	2.5685	2.7018	2.5685	2.1813	1.5777	0.8162	-0.0301
-0.0301	0.8170	1.5793	2.1835	2.5710	2.7045	2.5710	2.1835	1.5793	0.8170	-0.0301
-0.0302	0.8178	1.5809	2.1856	2.5736	2.7072	2.5736	2.1856	1.5809	0.8178	-0.0302
-0.0302	0.8186	1.5825	2.1878	2.5762	2.7099	2.5762	2.1878	1.5825	0.8186	-0.0302
-0.0303	0.8194	1.5840	2.1900	2.5787	2.7126	2.5787	2.1900	1.5840	0.8194	-0.0303
-0.0303	0.8202	1.5856	2.1922	2.5813	2.7154	2.5813	2.1922	1.5856	0.8202	-0.0303
-0.0303	0.8210	1.5872	2.1944	2.5839	2.7181	2.5839	2.1944	1.5872	0.8210	-0.0303
-0.0304	0.8218	1.5888	2.1966	2.5865	2.7208	2.5865	2.1966	1.5888	0.8218	-0.0304

Las columnas 4, 6 y 10 de la matriz se corresponden con las evoluciones  $u(3\pi/10,t)$ ,  $u(\pi/2,t)$  y  $u(9\pi/10,t)$ . La siguiente secuencia de órdenes permite graficar los tres vectores:

```
hold on
figure
plot(tiempo,WW(:,4),'m',tiempo,WW(:,6),'b',tiempo,WW(:,10),'g')
axis([0 1 0 3])
grid
legend('u(3*pi/10,t)', 'u(pi/2,t)', 'u(9*pi/10,t)', 'Location', 'Best')
xlabel('Tiempo')
ylabel('Temperatura')
```

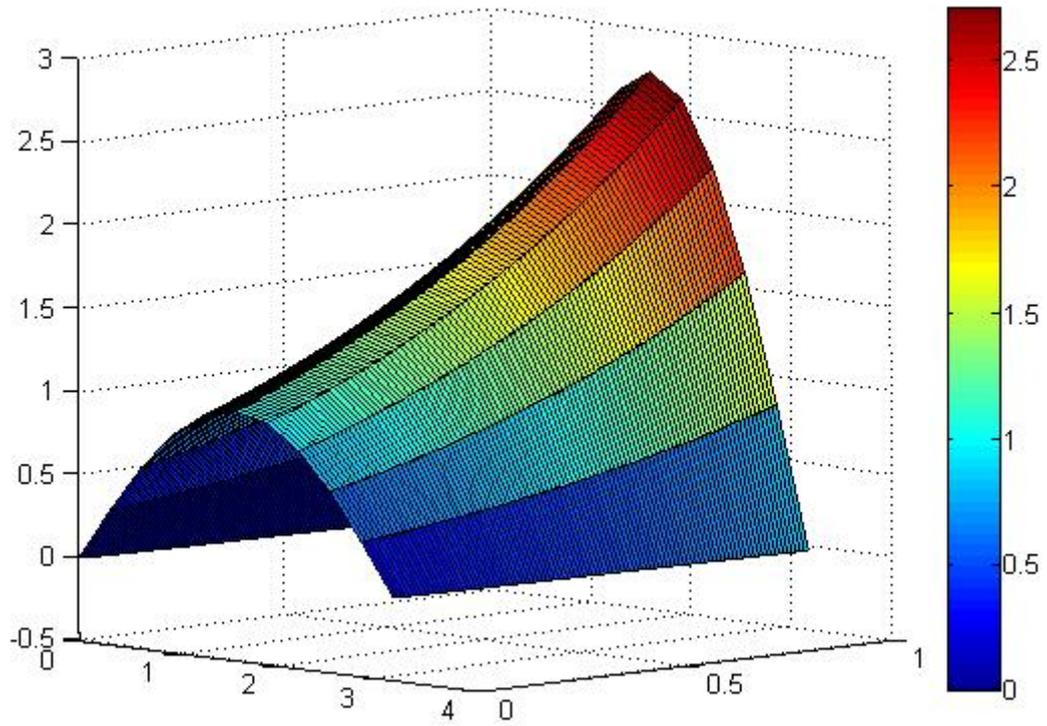


### SOLUCIÓN c)

Escribiendo la secuencia de órdenes:

```
[WW,L,m,N,T]=RegresivoNN(1,pi,10,100,1)
figure
r=linspace(0,L,m+1);
s=linspace(0,T,N+1);
[rr,ss]=meshgrid(r,s);
surf(rr,ss,WW);
```

Se obtiene una malla con la evolución de la distribución de temperaturas en la barra para el intervalo pedido:





### 3.3. Ecuación del Calor, método Crack-Nicholson y condiciones NN

Dada la Ecuación del Calor para  $x$  en  $(0,1)$ :

$$\frac{\partial u}{\partial t}(x,t) = \frac{\partial^2 u}{\partial x^2}(x,t) + e^x \cdot (-\cos(t) - \sin(t))$$

Y las siguientes condiciones de contorno e iniciales:

$$\frac{\partial u}{\partial x}(0,t) = \cos(t); \quad \frac{\partial u}{\partial x}(1,t) = e \cdot \cos(t)$$

$$u(x,0) = e^x$$

Se pide:

- Calcular  $u(x,5)$  con un error de truncamiento inferior a  $10^{-2}$  y graficar el vector junto a la condición inicial y la solución real, dada por la ecuación:

$$u(x,t) = e^x \cos t$$

- Calcular y representar  $u(0,t)$ ,  $u(1/2,t)$  y  $u(1,t)$  para  $t$  en  $(0,5)$
- Dibujar una malla con la distribución de temperaturas de la barra en el intervalo de tiempo  $(0,5)$

#### SOLUCIÓN a)

Se utiliza el script del método de Crack-Nicholson con condiciones de contorno tipo NN:



function CrankNicholsonNN(c,L,m,N,T)

% Entradas:

% c=coeficiente de difusión  
% L=longitud de la barra  
% m=nº de divisiones espaciales  
% N=nº de divisiones temporales  
% T=tiempo final

% Salidas:

% W=vector distribución de temperaturas en el instante T  
% WW=matriz de temperaturas en (0,T)

h=L/m;  
k=T/N;  
la=c^2\*k/h^2;

% Cálculo de A

A=tridiagonal(1+la,-la/2,m+1);  
A(m+1,m)=-la;  
A(1,2)=-la;

% Cálculo de B

B=tridiagonal(1-la,la/2,m+1);  
B(m+1,m)=la;  
B(1,2)=la;

% Cálculo de Wo y la solución real

```
for i=0:m
    x(i+1)=i*h;
    % Creación del vector longitud de de la barra
    ejex= [x'];
    W(i+1)=ff(i*h);
    U(i+1)=u(x(i+1),T);
end
W=W';
Wo=W;
U=U';
```

% Constitución de la matriz WW

WW(1,:)=W';



```
% Cálculo de W1,...,WN
for j=0:N-1
    J(1,j+1)=j*k;
    % Creación del vector tiempo
    tiempo=[J';T];
    % Cálculo de bj
    for i=0:m
        b(i+1)=k*gg(i*h,(j+1/2)*k);
    end
    b(1)=b(1)-la*h*(aa((j+1)*k)+aa(j*k));
    b(m+1)=b(m+1)+la*h*(bb((j+1)*k)+bb(j*k));
    W=A\(B*W+b');
    % Constitución de la matriz WW
    WW(j+2,:)= [W]';
end
```

Y se definen las funciones condición de contorno, inicial y perturbación externa:

```
function y=aa(t)
y=cos(t);
```

```
function y=bb(t)
y=exp(1)*cos(t);
```

```
function y=ff(x)
y=exp(x);
```

```
function y=gg(x,t)
y=exp(x)*(-cos(t)-sin(t));
```

```
function y=u(x,t)
y=exp(x)*sin(t);
```

Para trabajar con un error determinado se emplea el siguiente script :



```
function [norma,m,N]=Truncamiento(c,L,m,N,T,Tol)
```

```
% El algoritmo base deberá tener como salida W
```

```
% Entradas:
```

```
% c=coeficiente de difusión
```

```
% L=longitud de la barra
```

```
% m=nº de divisiones espaciales
```

```
% N=nº de divisiones temporales
```

```
% T=tiempo final
```

```
% Tol=tolerancia deseada
```

```
% Salidas:
```

```
% m=nº de divisiones espaciales necesarias para trabajar con un Error < Tol
```

```
% N=nº de divisiones temporales necesarias para trabajar con un Error < Tol
```

```
% norma=norma del error
```

```
cont=0;
```

```
[W]=CrankNicholsonNN(c,L,m,N,T);
```

```
m=2*m; N=2*N;
```

```
cont=cont+1;
```

```
[W2]=CrankNicholsonNN(c,L,m,N,T);
```

```
Wcomparada=W2(1:2^cont:m+1);
```

```
while norm(W-Wcomparada) > Tol
```

```
    W=Wcomparada;
```

```
    m=2*m; N=N*2;
```

```
    cont=cont+1;
```

```
    [W2]=CrankNicholsonNN(c,L,m,N,T);
```

```
    Wcomparada=W2(1:2^cont:m+1);
```

```
end
```

```
norma=norm(W-Wcomparada);
```

Escribiendo la orden:

```
[norma,m,N]=Truncamiento(1,1,2,2,5,10^-2)
```

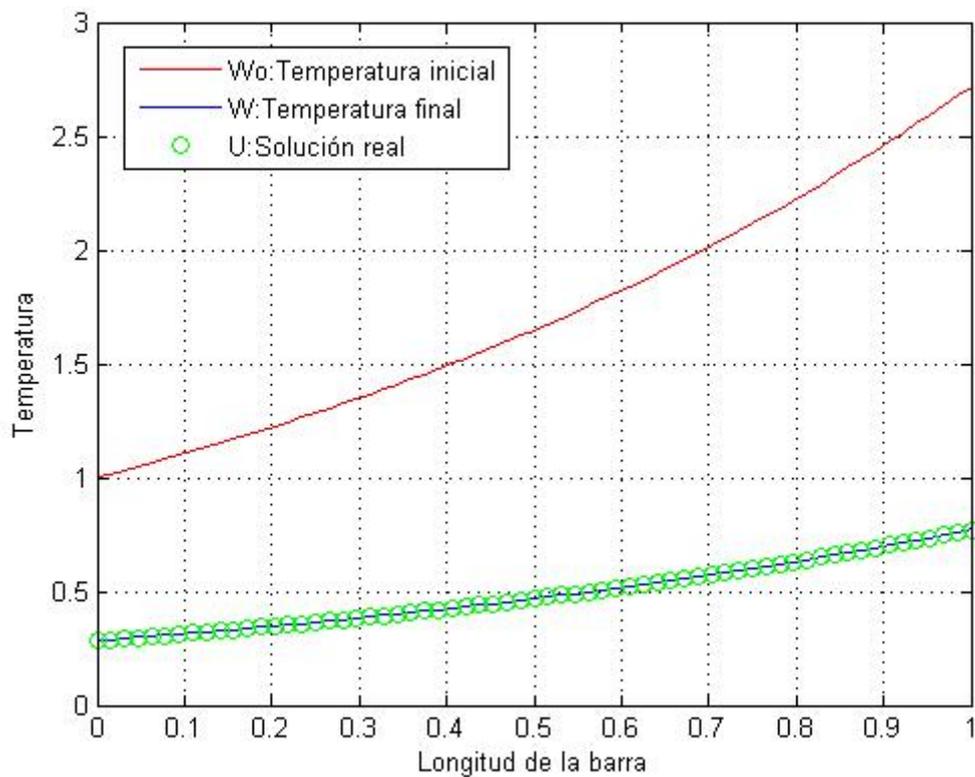
Se obtienen los valores de **m** y **N** para que el error en el cálculo sea inferior al pedido; en este caso **m=N=64**. Ahora se recurre al script del método de Crank-Nicholson:



```
[WW,tiempo]=CrankNicholsonNN(1,1,64,64,5)
```

Y se grafican las soluciones:

```
hold on  
figure  
plot(tiempo,WW(:,4),'m',tiempo,WW(:,6),'b',tiempo,WW(:,10),'g')  
axis([0 1 0 3])  
grid  
legend('u(3*pi/10,t)', 'u(pi/2,t)', 'u(9*pi/10,t)', 'Location', 'Best')  
xlabel('Tiempo')  
ylabel('Temperatura')
```





**SOLUCIÓN b)**

Para calcular las evoluciones previstas es necesario llamar al programa base e identificar en la matriz solución **WW** las columnas correspondientes. En este caso supondremos un número menor de nodos espaciales:

$$[WW, tiempo]=CrankNicholsonNN(1, 1, 10, 1000, 5)$$

Se obtiene:

**WW=**

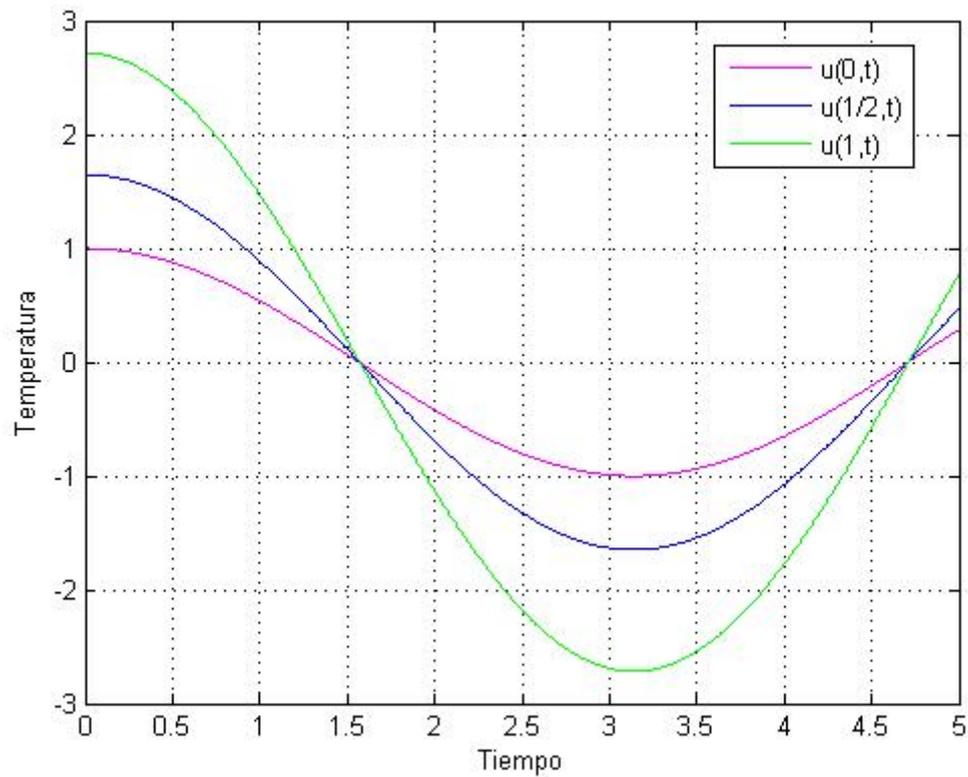
1.0000	1.1052	1.2214	1.3499	1.4918	1.6487	1.8221	2.0138	2.2255	2.4596	2.7183
1.0001	1.1052	1.2214	1.3498	1.4918	1.6487	1.8221	2.0137	2.2255	2.4595	2.7179
1.0001	1.1052	1.2214	1.3498	1.4918	1.6487	1.8220	2.0137	2.2254	2.4593	2.7177
1.0001	1.1052	1.2213	1.3497	1.4917	1.6486	1.8219	2.0135	2.2252	2.4591	2.7174
1.0001	1.1051	1.2212	1.3496	1.4916	1.6484	1.8218	2.0133	2.2250	2.4588	2.7171
1.0000	1.1050	1.2211	1.3495	1.4914	1.6482	1.8216	2.0131	2.2247	2.4585	2.7167
0.9999	1.1049	1.2210	1.3493	1.4912	1.6480	1.8213	2.0128	2.2243	2.4581	2.7163
0.9998	1.1047	1.2208	1.3491	1.4910	1.6477	1.8210	2.0124	2.2239	2.4576	2.7157
0.9996	1.1046	1.2206	1.3489	1.4907	1.6474	1.8206	2.0120	2.2235	2.4571	2.7152
0.9994	1.1043	1.2204	1.3486	1.4904	1.6471	1.8202	2.0116	2.2230	2.4565	2.7145
0.2321	0.2563	0.2831	0.3126	0.3453	0.3815	0.4214	0.4656	0.5143	0.5682	0.6277
0.2370	0.2617	0.2890	0.3192	0.3526	0.3895	0.4303	0.4753	0.5251	0.5802	0.6410
0.2418	0.2670	0.2949	0.3258	0.3598	0.3975	0.4391	0.4851	0.5359	0.5921	0.6541
0.2467	0.2724	0.3009	0.3323	0.3671	0.4055	0.4480	0.4949	0.5467	0.6040	0.6673
0.2515	0.2778	0.3068	0.3389	0.3743	0.4135	0.4568	0.5046	0.5575	0.6159	0.6805
0.2564	0.2831	0.3127	0.3454	0.3815	0.4215	0.4656	0.5144	0.5683	0.6278	0.6936
0.2612	0.2885	0.3186	0.3519	0.3887	0.4294	0.4744	0.5241	0.5790	0.6397	0.7067
0.2660	0.2938	0.3245	0.3584	0.3959	0.4374	0.4832	0.5338	0.5897	0.6516	0.7198
0.2708	0.2991	0.3304	0.3649	0.4031	0.4453	0.4920	0.5435	0.6005	0.6634	0.7329
0.2757	0.3044	0.3363	0.3714	0.4103	0.4533	0.5007	0.5532	0.6112	0.6752	0.7460
0.2805	0.3098	0.3421	0.3779	0.4175	0.4612	0.5095	0.5629	0.6219	0.6870	0.7591
0.2853	0.3151	0.3480	0.3844	0.4246	0.4691	0.5182	0.5725	0.6325	0.6988	0.7721

Las columnas **1, 6 y 11** de la matriz se corresponden con las evoluciones **u(0,t), u(1/2,t) y u(1,t)**. La siguiente secuencia de órdenes permite graficar los tres vectores:



```
hold on  
figure  
plot(tiempo, WW(:, 1), 'm', tiempo, WW(:, 6), 'b', tiempo, WW(:, 11), 'g')  
axis([0 5 -3 3])  
grid  
legend('u(0,t)', 'u(1/2,t)', 'u(1,t)', 'Location', 'Best')  
xlabel('Tiempo')  
ylabel('Temperatura')
```

El resultado:



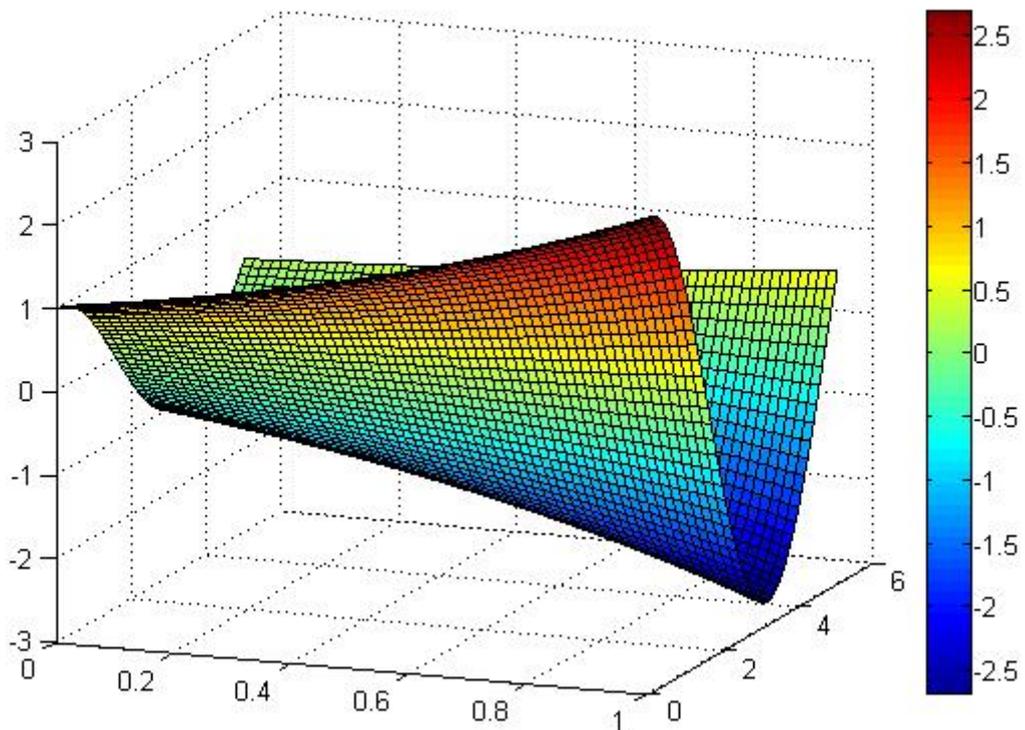
### SOLUCIÓN c)

Para dibujar la malla:



```
[WW,L,m,N,T]=CrankNicholsonNN(1,1,64,64,5)  
figure  
r=linspace(0,L,m+1);  
s=linspace(0,T,N+1);  
[rr,ss]=meshgrid(r,s);  
surf(rr,ss,WW);
```

El resultado:





### 3.4. Ecuación de Ondas, condiciones NN

Dada la Ecuación de Ondas para  $x$  en  $(0, 2\pi)$ :

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

Y las siguientes condiciones de contorno e iniciales:

$$\frac{\partial u}{\partial x}(0, t) = 1; \quad \frac{\partial u}{\partial x}(2\pi, t) = 1$$

$$u(x, 0) = 1 + \sin(x), \quad \frac{\partial u}{\partial t}(x, 0) = 0$$

Se pide:

- Calcular  $u(x, 1)$  con un error de truncamiento inferior a  $10^{-1}$  y graficar el vector junto a la condición de posición inicial.
- Dibujar una malla con la distribución de temperaturas de la barra en el intervalo de tiempo  $(0, 1)$
- Calcular los valores mínimo y máximo de  $u(x, t)$  en los intervalos de tiempo  $(0, 1)$  y espacio  $(0, 2\pi)$

#### SOLUCIÓN a)

El script de resolución de la Ecuación de Ondas con condiciones tipo Neumann en los extremos:



function OndasNN(c,L,m,N,T)

% Entradas:

% c=velocidad de propagación de la onda

% L=longitud de la cuerda

% m=nº de divisiones espaciales

% N=nº de divisiones temporales

% T=tiempo final

% Salidas:

% W=vector distribución de posiciones en el instante T

% MW=matriz de posiciones en (0,T)

h=L/m;

k=T/N;

la=(c\*k)/h;

% Cálculo de la matriz A

A=tridiagonal(2\*(1-la^2),la^2,m+1);

A(1,2)=2\*la^2;

A(m+1,m)=2\*la^2;

% Cálculo de la solución real

for i=1:m+1

% Calculo el vector longitud de la cuerda

    x(i)=(i-1)\*h;

    ejex=x';

    U(i)=u(x(i),T);

end

U=U';

% Cálculo de W0 y W1

for i=0:m

    W0(i+1)=ff(i\*h);

    W1(i+1)=((1-la^2)\*ff(i\*h)+(la^2/2)\*(ff((i+1)\*h)+ff((i-1)\*h))+k\*v(i\*h)+(k^2/2)\*gg(i\*h,0));

end

W0=W0';

MW(1,:)=W0';

W1=W1';

MW(2,:)=W1';



```
% Cálculo de W2,W3..., WN
for j=1:N-1
    % Creación del vector tiempo
    J(1,j+1)=j*k;
    tiempo=[J';T];
    %Cálculo de bj
    for i=0:m
        b(i+1)=k^2*gg(i*h,j*k);
    end
    b(1)=b(1)-2*(la^2)*h*aa(j*k);
    b(m+1)=b(m+1)+2*(la^2)*h*bb(j*k);
    W=A*W1-W0+b';
    MW(j+2,:)= [W]';
    W0=W1;
    W1=W;
end
```

Y las funciones condición de contorno, iniciales y perturbación externa:

```
function y=aa(t)
y=1;
```

```
function y=bb(t)
y=1;
```

```
function y=ff(x)
y=1+sin(x);
```

```
function y=gg(x,t)
y=sin(x)-cos(t);
```

```
function y=u(x,t)
y=0;
```

```
function y=v(x)
y=0;
```



Para trabajar con una tolerancia determinada se emplea el siguiente script, que determina los valores necesarios de  $m$  y  $N$ :

```
function [norma,m,N]=Truncamiento(c,L,m,N,T,Tol)

% El algoritmo base deberá tener como salida W

% Entradas:
% c=coeficiente de difusión
% L=longitud de la barra
% m=nº de divisiones espaciales
% N=nº de divisiones temporales
% T=tiempo final
% Tol=tolerancia deseada
% Salidas:
% m=nº de divisiones espaciales necesarias para trabajar con un Error < Tol
% N=nº de divisiones temporales necesarias para trabajar con un Error < Tol
% norma=norma del error

cont=0;
[W]=OndasNN(c,L,m,N,T);
m=2*m; N=2*N;
cont=cont+1;
[W2]= OndasNN(c,L,m,N,T);
Wcomparada=W2(1:2^cont:m+1);
while norm(W-Wcomparada)> Tol
    W=Wcomparada;
    m=2*m; N=N*2;
    cont=cont+1;
    [W2]= OndasNN(c,L,m,N,T);
    Wcomparada=W2(1:2^cont:m+1);
end
norma=norm(W-Wcomparada);
```

Escribiendo:

```
[norma,m,N]=Truncamiento(1,2*pi,5,5,1,10^-1)
```



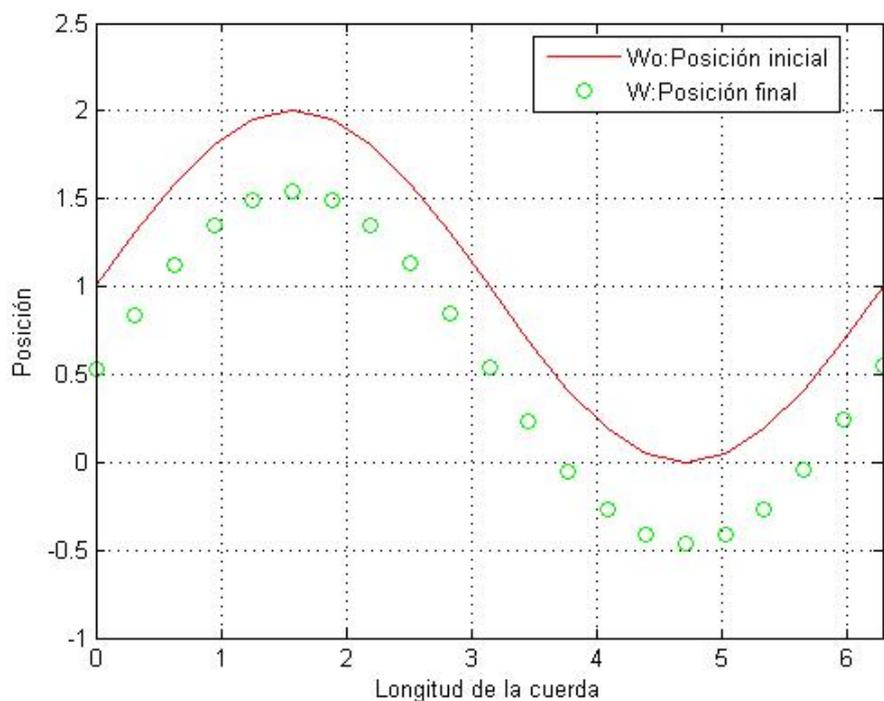
Se obtienen los valores necesarios de equidistancia nodal espacial y temporal para trabajar con un tolerancia de **0.1**, en este caso **m=N=20**. Ahora se llama al script base:

```
[W,ejex,Wo]=ondasNN(1,2*pi,20,20,1)
```

Para plotear el vector **u(x,1)** junto a la condición inicial se escribe:

```
hold on  
figure  
plot(ejex,Wo,'r',ejex,W,'g o')  
axis([0 2*pi -1 2.5])  
grid  
legend('Wo:Posición inicial','W:Posición final','Location','Best')  
xlabel('Longitud de la cuerda')  
ylabel('Posición')
```

Y resulta:



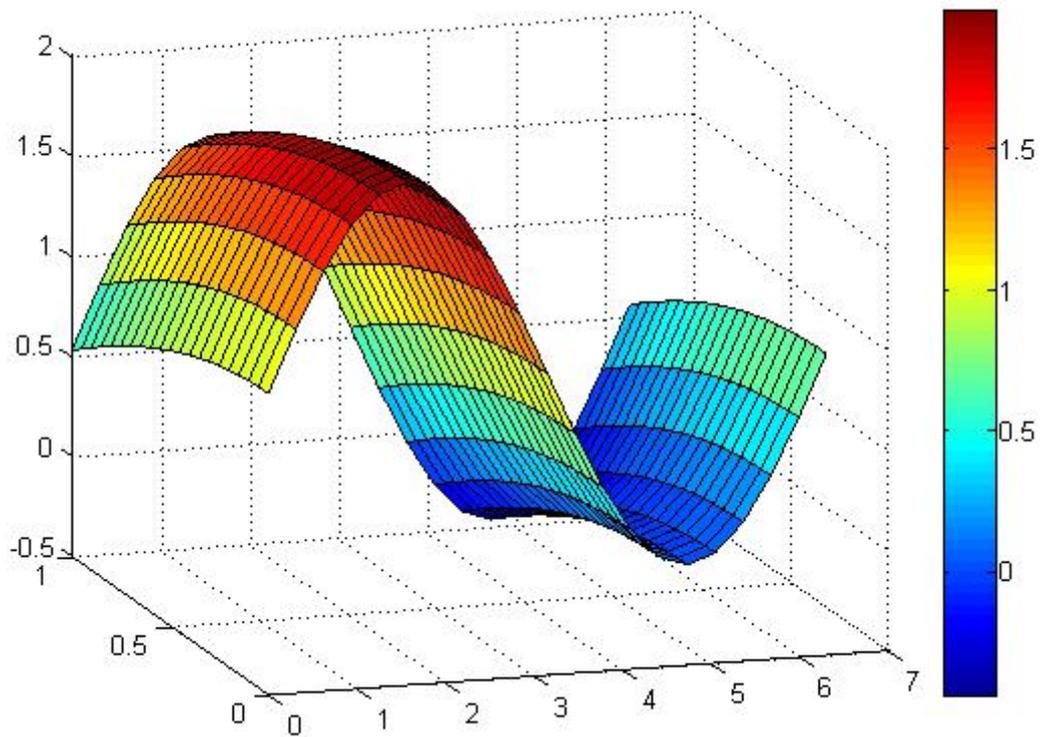


### SOLUCIÓN b)

La malla se obtiene escribiendo:

```
[MW,L,m,N,T]=ondasNN(1,2*pi,20,20,1)
figure
r=linspace(0,L,m+1);
s=linspace(0,T,N+1);
[rr,ss]=meshgrid(r,s);
surf(rr,ss,MW);
```

El resultado:





### SOLUCIÓN c)

Los valores máximo y mínimo de  $u(x,t)$  en los intervalos tratados se obtienen incluyendo en el script base las siguientes líneas:

#### % Cálculo del valor mínimo

```
MIN=MW(1,1);  
x=0;  
t=0;  
for i=1:N+1  
    for j=1:m+1  
        if MIN > MW(i,j)  
            x=(j-1)*h;  
            t=(i-1)*k;  
            MIN=MW(i,j);  
        end  
    end  
end  
MIN;  
xiMIN=x;  
tiMIN=t;
```

#### % Cálculo del valor máximo

```
MAX=MW(1,1);  
x=0;  
t=0;  
for i=1:N+1  
    for j=1:m+1  
        if MAX < MW(i,j)  
            x=(j-1)*h;  
            t=(i-1)*k;  
            MAX=MW(i,j);  
        end  
    end  
end  
MAX;  
xiMAX=x;  
tiMAX=t;
```



Al escribir la llamada al script:

*[MW,MAX,xiMAX,tiMAX,MIN,xiMIN,tiMIN]=ondasNN(1,2\*pi,20,20,1)*

Se obtienen los valores buscados:

<i>MAX =</i>	<i>xiMAX =</i>	<i>tiMAX =</i>	<i>MIN =</i>	<i>xiMIN =</i>	<i>tiMIN =</i>
<i>2</i>	<i>1.5708</i>	<i>0</i>	<i>-0.4636</i>	<i>4.7124</i>	<i>1</i>