Asignatura	Datos del alumno	Fecha
Métodos Numéricos II	**Apellidos:** Avilés Cahill	27/02/2024
	Nombre: Adán	

Laboratorio: Ecuaciones en derivadas parciales parabólicas.

1. Introducción

Con esta actividad se pondrá en práctica los conceptos relacionados con las EDP's parabólicas. Concretamente, la técnica de diferencias finitas aplicada a un problema con condiciones no Dirichlet.

1.1. Descripción:

Consideramos la siguiente ecuación en derivadas parciales parabólica:

$$u_t = u_{xx} - u_x + u, \qquad 0 < x < 1, t > 0 \tag{1}$$

Con las siguientes condiciones de contorno:

$$u_x(0,t) = 2t, \quad u(1,t) = \frac{t^2}{2}, \qquad t > 0$$

y la condición inicial:

$$u(x,0) = sin(x) + cos(x), \qquad 0 < x < 1$$

2. Actividades

2.1. Método explícito

Transformaremos el problema en un esquema de diferencias finitas de orden $\mathcal{O}(k + h^2)$, para ello aproximaremos usando las diferencias progresivas en u_t y centrales en u_{xx} y u_x. Por tanto, nuestro problema se transforma en:

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

Evaluaremos esta expresión en los pares (x_i, t_j), con $i = 0, \dots, nx-1$ y $j = 0, \dots, nt-1$. Por simplificar, usaremos la notación $u(x_i, t_j) = u_{i,j}$. Por tanto, nuestra ecuación anterior se convierte en

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

Si definimos $\lambda = \frac{k}{2h^2}$ y hacemos las transformaciones pertinentes para despejar $u_{i,j+1}$, obtendremos

$$u_{i,j+1} = \lambda(2-h)u_{i+1,j} + \lambda(2+h)u_{i-1,j} + (1 + 2h^2\lambda - 4\lambda)u_{i,j}$$

que es la expresión que usaremos para iterar y encontrar la solución al problema. Hemos de tener en cuenta, que cuando tomemos $i = 0$, tendremos un término $u_{-1,j}$ que no pertenece al dominio, para suplirlo, utilizaremos las condiciones de contorno.

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

por tanto, despejando, obtenemos que $u_{-1,j} = u_{1,j} - 4ht$. Así pues, nuestro sistema será, para $j = 0, \dots, nt$

$$\begin{cases} u_{0,j+1} = 4\lambda u_{1,j} + (1 + 2h^2\lambda - 4\lambda)u_{0,j} \\ u_{i,j+1} = \lambda(2-h)u_{i+1,j} + \lambda(2+h)u_{i-1,j} + (1 + 2h^2\lambda - 4\lambda)u_{i,j}, \quad i = 1, \dots, nx-1. \end{cases}$$

Programando el código en matlab, obtenemos

```
function [u,x,t] = metodo_explicito_practica(ci, h2, L, nx, Tmax, nt)
    % Metodo explicito para la ecuacion del calor con condiciones Dirichlet
    % Inicializacion
    h = L/nx;
    x = 0: h:L;
    k = Tmax/nt;
    t = 0:k:Tmax;
    cix = feval (ci ,x);
```

```
    u = zeros (nx+1, nt+1) ;
    u(:, 1) =cix';
    c2t = feval(h2,t);
    u(nx+1,:) = c2t;
    % Condicion de estabilidad/convergencia
    lambda = k/(2*h^2);
    if lambda>1/2
        disp ('No se cumple el criterio de convergencia')
    else
        disp ('sin problema ')
    end
    L = 1:nx-1; C = 2:nx; R = 3:nx+1;
    for j = 1:nt
        u(1, j+1) = 4*lambda*u(2,j) - lambda*(2+h)*(4*h*t(j))...
                        +(1+2*h^2*lambda-4*lambda)*u(1,j);
        u(C, j+1) = lambda*(2-h)*u(R,j)+lambda*(2+h)*u(L,j)...
                        +(1+2*h^2*lambda-4*lambda)*u(C,j);
    end
end
```

La solución en $t = 0{,}5$, tomando $h = 0{,}1$ $(nx = 10)$ y $k = 0{,}005$ $(nt = 1000)$ es:

x	$u(x, 0{,}5)$
0	0,337102
0,1	0,425285
0,2	0,488653
0,3	0,526422
0,4	0,538272
0,5	0,524413
0,6	0,485668
0,7	0,423562
0,8	0,340420
0,9	0,239476
1	0,125000

2.2. Método implícito

Usaremos ahora las diferencias regresivas en u_t y centrales en u_{xx} y u_x. Además, como en el paso anterior, denotaremos $u(x_i, t_j) = u_{i,j}$, así pues tendremos:

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

Tomando el mismo lambda que en el método explícito, y haciendo las transformaciones pertinentes, llegamos a la expresión implícita:

$$(1 + 4\lambda - 2h^2\lambda)u_{i,j} + \lambda(h - 2)u_{i+1,j} - \lambda(h + 2)u_{i-1,j} = u_{i,j-1}$$

Como en el caso anterior, en $i = 0$, tenemos $u_{-1,j}$, que tendremos que paliar también. En este caso (gracias a haber despejado de las condiciones de contorno), la relación será:

$$(1 + 4\lambda - 2h^2\lambda)u_{0,j} + \lambda(h-2)u_{1,j} - \lambda(h+2)(u_{1,j} - 4ht_j) = u_{0,j-1}$$

reordenando, obtenemos:

$$(1 + 4\lambda - 2h^2\lambda)u_{0,j} - 4\lambda u_{1,j} + \lambda(h+2) * 4ht_j = u_{0,j-1}$$

Si lo expresamos en forma matricial, tendremos:

$$A \begin{pmatrix} u_{0,j} \\ u_{1,j} \\ \vdots \\ u_{nx-2,j} \\ u_{nx-1,j} \end{pmatrix} = \begin{pmatrix} u_{0,j-1} \\ u_{1,j-1} \\ \vdots \\ u_{nx-2,j-1} \\ u_{nx-1,j-1} \end{pmatrix} + \begin{pmatrix} -\lambda(h+2)4ht_j \\ 0 \\ \vdots \\ 0 \\ \lambda\frac{t_j^2}{2} \end{pmatrix}$$

donde

$$A = \begin{pmatrix} 1-4\lambda-2h^2\lambda & -4\lambda & 0 & \dots & 0 & 0 \\ -\lambda(2+h) & 1-4\lambda-2h^2\lambda & \lambda(h-2) & \dots & 0 & 0 \\ 0 & -\lambda(2+h) & 1-4\lambda-2h^2\lambda & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 1-4\lambda-2h^2\lambda & \lambda(h-2) \\ 0 & 0 & 0 & \dots & -\lambda(2+h) & 1-4\lambda-2h^2\lambda \end{pmatrix}$$

si lo implementamos en Matlab, obtendremos:

```
function [U,x,t] = metodo_implicito_practica(ci, c2,x0, xf,nx, Tmax, nt)
    h = (xf-x0)/nx;
    x = 0: h:xf;
    k = Tmax/nt;
    t = 0:k:Tmax;
    U = zeros(nx+1, nt+1) ;
    U(end,:) = feval(c2, t);
    U(:,1) = feval(ci,x);
    lambda = k/(2*h^2);
    %Calculo de diagonales
    dp = (1+4*lambda-2*h^2*lambda)*ones(1 ,nx-1) ;
    ds= [-4*lambda lambda*(h-2)*ones(1, nx-3)] ;
    di = -lambda*(h+2)*ones(1, nx-2);
    for j =2:nt+1
        d=U(2:nx, j-1);
        d(1) =d(1) - lambda*(2+h)*4*h*t(j) ;
        d(end) =d(end) + lambda*U(end, j-1);
        z= Crout(dp ,ds ,di ,d);
        U(2:nx, j)=z;
    end
end
```

Una vez implementado, lo ejecutamos:

```
[U,x,t] = metodo_implicito_practica(@(x) sin(x)+cos(x), @(x) x.^2./2, 0, 1, 10, 0.5, 1000);
```

obteniendo como resultados:

x	$u(x, 0{,}5)$
0	0,336820
0,1	0,424576
0,2	0,487964
0,3	0,525769
0,4	0,537669
0,5	0,523875
0,6	0,485211
0,7	0,423201
0,8	0,340167
0,9	0,239345
1	0,125000

2.3. Método de Crank-Nicholson

Para la realización de este método, hemos de realizar la media aritmética entre el resultado de aproximar en el instante t_j con diferencias progresivas en u_t,

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

y el resultado de aproximar en t_{j+1} con diferencias regresivas para u_t

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

obteniendo la expresión:

$$\begin{aligned}\frac{u_{i,j+1}-u_{i,j}}{k} =& \frac{1}{2}\Big(\frac{u_{i+1,j}-2u_{i,j}+u_{i-1,j}}{h^2}+\frac{u_{i+1,j+1}-2u_{i,j+1}+u_{i-1,j+1}}{h^2} \\ &-\frac{u_{i+1,j}-u_{i-1,j}}{2h}-\frac{u_{i+1,j+1}-u_{i-1,j+1}}{2h} \\ &+u_{i,j}+u_{i,j+1}\Big)\end{aligned}$$

tomando $\lambda = \dfrac{k}{2h^2}$ y reordenando,

$$\begin{aligned}&(1+2\lambda-h^2\lambda)u_{i,j+1}+(\frac{h\lambda}{2}-\lambda)u_{i+1,j+1}-(\frac{h\lambda}{2}+\lambda)u_{i-1,j+1}= \\ &(1-2\lambda+h^2\lambda)u_{i,j}+(-\frac{h\lambda}{2}+\lambda)u_{i+1,j}+(\frac{h\lambda}{2}+\lambda)u_{i-1,j}\end{aligned}$$

Como vemos, los coeficientes son muy similares y hemos de tener cuidado con los signos. Vamos a reescribir esta expresión en forma matricial, pero antes hemos de tener cuidado en el caso de $i=0$, pues tendremos $u_{-1,j}$ y $u_{-1,j+1}$, así que usaremos las condiciones de contorno para evitarlo (como en los métodos anteriores), obteniendo que $u_{-1,j}=u_{1,j}-4ht_j$ y que $u_{-1,j+1}=u_{1,j+1}-4ht_{j+1}$.Así, la expresión para $i=0$ será:

$$\begin{aligned}&(1+2\lambda-h^2\lambda)u_{0,j+1}-2\lambda u_{1,j+1}-(\frac{h\lambda}{2}+\lambda)4ht_{j+1}= \\ &(1-2\lambda+h^2\lambda)u_{0,j}+2\lambda u_{1,j+1}+(\frac{h\lambda}{2}+\lambda)4ht_j\end{aligned}$$

Ahora sí, podemos pasar a reescribir la ecuación en forma $Au^{j+1} = Bu^j + d_j$, con $j = 0, 1, \dots, nt-1$. Entonces, tendremos:

$$A = \begin{pmatrix} 1+2\lambda-h^2\lambda & -2\lambda & 0 & \dots & 0 & 0 \\ -\lambda(1+\frac{h}{2}) & 1+2\lambda-h^2\lambda & \lambda(\frac{h}{2}-1) & \dots & 0 & 0 \\ 0 & -\lambda(1+\frac{h}{2}) & 1+2\lambda-h^2\lambda & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 1+2\lambda-h^2\lambda & \lambda(\frac{h}{2}-1) \\ 0 & 0 & 0 & \dots & -\lambda(1+\frac{h}{2}) & 1+2\lambda-h^2\lambda \end{pmatrix}$$

$$B = \begin{pmatrix} 1-2\lambda+h^2\lambda & 2\lambda & 0 & \dots & 0 & 0 \\ \lambda(1+\frac{h}{2}) & 1-2\lambda+h^2\lambda & \lambda(1-\frac{h}{2}) & \dots & 0 & 0 \\ 0 & \lambda(1+\frac{h}{2}) & 1-2\lambda+h^2\lambda & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 1-2\lambda+h^2\lambda & \lambda(1-\frac{h}{2}) \\ 0 & 0 & 0 & \dots & \lambda(1+\frac{h}{2}) & 1-2\lambda+h^2\lambda \end{pmatrix}$$

$$d_j = \begin{pmatrix} \lambda(1+\frac{h}{2})4h(t_{j+1}-t_j) \\ 0 \\ \vdots \\ 0 \\ \frac{\lambda}{4}(t_{j+1}^2+t_j^2 \end{pmatrix}$$

implementamos el método en matlab:

```
function [U,x,t] = metodo_crank_nicholson(ci, c2, a, b, nx, T, nt)
    h = (b-a)/nx;
    x = a:h:b;
    k = T/nt;
    t = 0:k:T;
    U = zeros(nx+1, nt+1) ;
    U(end,:) = feval(c2, t);
    U(:,1) = feval(ci,x);
    lambda = k/(2*h^2);
    %Calculo de diagonales
    dpA = (1+2*lambda-h^2*lambda)*ones(1 ,nx+1) ;
    dsA= [-2*lambda lambda*(h/2-1)*ones(1, nx-1)] ;
    diA = -lambda*(1+h/2)*ones(1, nx);
    d = zeros(nx+1,1);
    dpB = (1-2*lambda+h^2*lambda)*ones(1 ,nx+1) ;
    dsB= [2*lambda lambda*(-h/2+1)*ones(1, nx-1)] ;
    diB = lambda*(1+h/2)*ones(1, nx);
    B = diag(dpB)+diag(dsB,1)+diag(diB,-1);
    for j =1:nt
        d(1) = d(1) - lambda*(1+h/2)*4*h*(t(j+1)-t(j)) ;
        d(end) = d(end) + (lambda/2)*(U(end, j+1)+U(end, j));
        d = B*U(:,j)+d;
        U(:,j +1) = Crout(dpA ,dsA,diA,d)';
    end
end
```

En este caso, claramente el resultado no es correcto, ya que me salen valores del orden de 10^3. Tras repasar las cuentas, no logro ver donde está el fallo, si en el código o en la obtención de las matrices. Dicho eso, esa es la idea general del método. Probablemente en el código esté machacando algún subíndice que no deba, pues no se mantiene ni el $0{,}125$ final, pero no consigo ver el error.