Programación: un estudio experimental del error de truncamiento local en varios esquemas de diferencias finitas para la ecuación de calor sobre un intervalo

Objetivos. Calcular el error de truncamiento local para una solución de la ecuación de calor y para varios esquemas de diferencias finitas.

1. Ejemplo trigonométrico (repaso). Consideremos la ecuación de calor

$$(D_2 u)(x, t) = (D_1^2 u)(x, t) \tag{1}$$

con las condiciones de frontera $u(0, t) = u(1, t) = 0$ y con la condición inicial

$$u(x, 0) = \operatorname{sen}(\pi x).$$

En las clases pasadas vimos que la solución de este problema es

$$u(x, t) = \operatorname{sen}(\qquad\quad)\,\underbrace{\qquad\qquad\quad}_{e^{???}}. \tag{2}$$

La función (2) se puede programar con una linea en Matlab/Octave:

```
utrig = @(x, t) ???;
```

2. Aproximación de la derivada parcial respecto al tiempo. La derivada $D_2 u$ respecto al tiempo se puede aproximar usando la diferencia finita progresiva (hacia adelante):

$$(D_2 u)(x, t) \approx \frac{u(x, t + \tau) - u(x, t)}{\qquad}. \tag{3}$$

También se puede aproximar por la diferencia finita regresiva (hacia atrás). Escribimos esta aproximación para el punto $(x, t + \tau)$:

$$(D_2 u)(x, t + \tau) \approx \frac{u(x, t + \tau) - u(x, t)}{\tau}. \tag{4}$$

Notamos que las fórmulas (3) y (4) tienen el mismo lado derecho.

3. Aproximación de la segunda derivada parcial respecto al espacio. Aproximamos la segunda derivada respecto a la variable espacial usando la diferencia finita del segundo orden:

$$(D_1^2 u)(x,t) \approx \frac{u(x-h,t) - \boxed{\qquad\qquad} + \boxed{\qquad\qquad}}{\boxed{\quad}}. \tag{5}$$

Escribimos la misma aproximación para el punto $(x, t+\tau)$ en vez del punto (x,t):

$$(D_1^2 u)(x,t+\tau) \approx \frac{u(x-h,t+\tau) - \boxed{\qquad\qquad} + \boxed{\qquad\qquad}}{\boxed{\quad}}. \tag{6}$$

4. Error de truncamiento local en el método explícito. En la ecuación (1) cambiamos ambos lados por las aproximaciones (3) y (5):

$$\frac{u(x,t+\tau) - u(x,t)}{\boxed{\quad}} \approx \frac{u(x-h,t) - \boxed{\qquad\qquad} + \boxed{\qquad\qquad}}{\boxed{\quad}},$$

luego multiplicamos ambos lados por τ y denotamos τ/h^2 por ρ:

$$u(x,t+\tau) - u(x,t) \approx \rho\Big(\boxed{\qquad\qquad\qquad\qquad\qquad\qquad}\Big).$$

Del lado izquierdo restamos el lado derecho y denotamos su diferencia por $R_{1,u,x,t}(h,\tau)$:

$$R_{1,u,x,t}(h,\tau) \coloneqq u(x,t+\tau) - \rho u(x-h,t) + \big(\boxed{\qquad\qquad}\big)u(x,t) - \boxed{\qquad\qquad}. \tag{7}$$

5. Error de truncamiento local en el método implícito. En la ecuación (1) cambiamos ambos lados por las aproximaciones (4) y (6):

$$\frac{u(x,t+\tau) - u(x,\boxed{\quad})}{\boxed{\quad}} \approx \frac{u(x-h,t+\tau) - \boxed{\qquad\qquad} + \boxed{\qquad\qquad}}{\boxed{\quad}},$$

luego multiplicamos ambos lados por τ y denotamos τ/h^2 por ρ:

$$u(\boxed{\qquad\quad}) - u(\boxed{\qquad\qquad}) \approx \rho\Big(\boxed{\qquad\qquad\qquad\qquad\qquad\qquad\qquad}\Big).$$

Del lado izquierdo restamos el lado derecho y denotamos su diferencia por $R_{2,u,x,t}(h,\tau)$:

$$R_{2,u,x,t}(h,\tau) \coloneqq -\rho u(x-h,t+\tau) + \big(\boxed{\qquad\qquad}\big)u(x,t+\tau) - \boxed{\qquad\qquad} - u(x,t). \tag{8}$$

6. Error de truncamiento local en el método de Crank y Nicolson. Recordamos las fórmula del método de Crank y Nicolson:

$$
\begin{aligned}
&-\rho u(x-h,t+\tau)+(2+2\rho)u(x,t+\tau)-\qquad\qquad\qquad \\
&\quad=\rho u(x-h,t)+(2-2\rho)u(x,t)+\qquad\qquad .
\end{aligned}
$$

Del lado izquierdo restamos el lado derecho y denotamos su diferencia por $R_{3,u,x,t}(h,\tau)$:

$$R_{3,u,x,t}(h,\tau) \coloneqq \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad . \tag{9}$$

7. Función que calcula y devuelve los errores de truncamiento local. Programemos una función que calcule las cantidades $R_{1,u,x,t}(h,\tau)$, $R_{2,u,x,t}(h,\tau)$ y $R_{3,u,x,t}(h,\tau)$. Suponemos que `u` es un apuntador a una función de dos variables, y los parámetros `x`, `t`, `h`, `tau` son algunos números reales positivos. Primero calculamos los valores de la función u en 6 puntos:

$u(x-h,t+\tau)$	$u(x,t+\tau)$	$u(x+h,t+\tau)$
`vl`	`vc`	`vr`
$u(x-h,t)$	$u(x,t)$	$u(x+h,t)$
`ul`	`uc`	`ur`

Luego programamos las fórmulas (7), (8) y (9):

```
function [R123] = loctruncerrors(u, x, t, h, tau),
   rho = ???;
   uc = u(x, t); ul = u(x - h, t); urW = u(x + h, t);
   vc = u(x, t + tau); vl = ???; vr = ???;
   R1 = vc - rho * ul + ???;
   R2 = ???;
   R3new = - rho * vl + ???  - ???;
   R3old = rho * ul + ??? + ???;
   R3 = Rnew - Rold;
   R123 = [R1, R2, R3];
end
```

8. Pruebas. Usamos la función (2) del ejemplo trigonométrico. Primero calculamos los errores de truncamiento local con un h muy pequeño y observamos la dependencia de τ. Luego fijamos un τ muy pequeño y observamos la dependencia de h. Al final, dividimos los errores de truncamiento local entre sus cotas superiores teóricas y calculamos estos cocientes para varios valores aleatorios de h y τ.

```
function [] = testloctruncerrors(),
   utrig = @(x, t) ???;
   x = 0.3; t = 0.05;
   h = 1.0E-5;
   for tau = [0.01, 0.005, 0.0025, 0.00125],
      disp(tau);
      display(loctruncerrors(utrig, x, t, h, tau));
   end
   tau = 1.0E-5;
   for h = [0.1, 0.05, 0.025, 0.0125],
      disp(h);
      display(loctruncerrors(utrig, x, t, h, tau));
   end
   for testiter = 1 : 10,
      h = exp(- 3 * rand() - 1);
      tau = exp(- 3 * rand() - 2);
      est1 = (h ^ 2) * tau + tau ^ 2;
      est2 = ???;
      est3 = ???;
      R123 = loctruncerrors(utrig, x, t, h, tau);
      disp([h, tau]);
      display(R123 ./ [est1, est2, est3]);
   end
end
```