Solución de la ecuación de calor sobre un intervalo usando el método explícito de diferencias finitas

Agradezco a Yahir Uriel Guzmán Pérez por el estudiar juntos este tema.

Requisitos. Solución de problemas de frontera para ODE de segundo orden con el método de diferencias finitas.

1. La ecuación de calor sobre un intervalo finito. Consideremos la ecuación

$$(D_2 u)(x,t) = (D_1^2 u)(x,t) \qquad (0 < x < 1,\ t > 0), \tag{1}$$

o, en la notación clásica, $\frac{\partial}{\partial t}u(x,t) = \frac{\partial^2}{\partial x^2}u(x,t)$. Suponemos que la función u se anula en la frontera del intervalo:

$$u(0,t) = \quad, \qquad u(1,t) = \quad \qquad (t \geq 0), \tag{2}$$

y satisface una condición inicial:

$$u(x,t) = f(x) \qquad (0 \leq x \leq 1). \tag{3}$$

Suponemos que $f \in C([0,1])$, $f(0) = f(1) = 0$.

2. Discretización del espacio. Dividimos el intervalo espacial $[0,1]$ en n partes iguales, entonces cada parte es de longitud $h := \quad/n$. En otras palabras, trabajamos con la malla uniforme de $\underbrace{\qquad\qquad}_{\text{¿cuántos?}}$ nodos equidistantes:

$$x_0 = \frac{\quad}{n} = \quad h = \quad, \qquad x_1 = \frac{\quad}{n} = \quad h, \qquad x_2 = \frac{\quad}{n} = \quad h,$$

$$x_j = \frac{\quad}{h} = \quad h, \qquad x_n = \frac{\quad}{n} = \quad h = \quad .$$

Notamos que

$$x_j + h = \frac{\qquad}{\quad} = \underbrace{\qquad\qquad}_{x_{???}}, \qquad x_j - h = \underbrace{\qquad\qquad}_{x_{???}}. \tag{4}$$

El vector $[x_0, x_1, \ldots, x_n]^\top$ se puede construir con el siguiente comando del lenguaje Matlab.

```
x = linspace(???, ???, ???)';
```

3. Discretización del tiempo. Nuestra respuesta final será la solución aproximada en un momento dado T, es decir, el vector de los valores aproximados de $u(x_j, T)$. Dividimos el intervalo $[0, T]$ en m partes iguales, cada una de longitud

$$\tau := \frac{\quad}{\quad}.$$

Los momentos del tiempo forman una malla uniforme $[t_0, \ldots, t_m]^\top$, donde

$$t_k = \frac{\quad}{\quad} = \quad \tau.$$

Notamos que

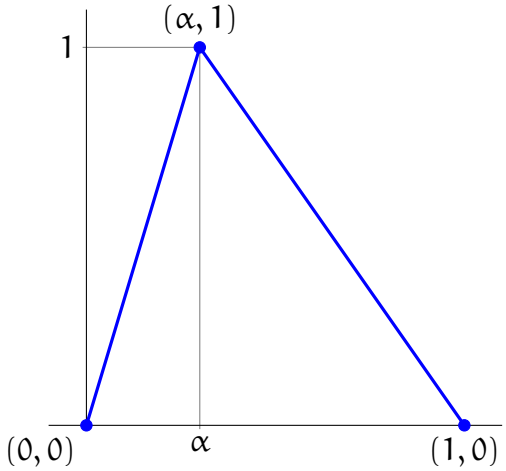
$$t_k + \tau = \underbrace{\quad}_{t_{???}}. \tag{5}$$

4. Ecuación de la recta que pasa por dos puntos dados en el plano (repaso). Para el siguiente ejemplo necesitamos recordar la ecuación de la recta que pasa por dos puntos dados (x_0, y_0), (x_1, y_1). Escrimos la ecuación en la siguiente forma:

$$y = k(x - x_0) + b.$$

Poniendo $x = x_0$ obtenemos $b =$ __. Al sustituir $x = x_1$, obtenemos $k = \frac{\quad}{\quad}$.

5. Ejemplo de condición inicial. Como un ejemplo de condición inicial, podemos trabajar con la siguiente función lineal a trozos:



Una de las rectas pasa por $(0, 0)$ y $(\alpha, 1)$, por eso su ecuación es

$$y = \frac{\quad}{\quad} x. \tag{6}$$

La otra recta pasa por los puntos (__ , __) y (__ , __), luego su ecuación es

$$y = \frac{\quad}{\quad}(x - \quad). \tag{7}$$

La función que nos interesa se puede obtener como el $\underbrace{\qquad\qquad}_{\text{mínimo/máximo}}$ de las dos funciones lineales dadas (6) y (7):

$$f(x) = \underbrace{\qquad}_{\min\,/\,\max} \left\{ \qquad , \qquad \right\}.$$

Definamos `f` en el intérprete de Matlab o GNU Octave (denotándola por `f1`) y dibujemos su gráfica para comprobar que la fórmula es correcta:

```
alpha = 0.3
f1 = @ (x) m?? (x / alpha, ???)
xg = linspace(0, 1, 201)';
plot(xg, f1(???));
```

6. Código para comprobación. Antes de escribir la función `solveheatfindifexplicit` que resuelva el problema, escribamos la función que hará la comprobación. Denotemos por n al número de las partes en las cuales se divide el intervalo espacial $[0, 1]$, por T al tiempo en el cual queremos calcular la solución, y por nT al número de partes en las cuales se divide el intervalo del tiempo $[0, T]$.

```
function [] = testsolveheat(n, nT),
   alpha = 0.3;
   f1 = @ (x) ???;
   x = linspace(0, 1, n);
   # despues de escribir solveheatfindifexplicit,
   # podremos descomentar la siguiente linea:
   # u = solveheatfindifexpl(f1, n, nT);
   # mientras definimos u de manera trivial:
   u = zeros(n + 1, 1);
   plot(x, f1(x), x, u, '*');
end
```

La función `testsolveheat` ya está lista para ejecutar, pero muestra solamente la gráfica de la función `f` y la gráfica de la función nula.

7. Aproximación de las derivadas parciales. En la definición de la derivada parcial D_2 fijamos la primera variable (la coordenada espacial), por eso tenemos la siguiente aproximación:

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

En la definición de D_1^2 fijamos la segunda variable (el momento del tiempo), por eso tenemos la siguiente aproximación:

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

Usando (5) y (4), obtenemos

$$(D_2 u)(x_j,t_k) \approx \frac{u(x_j,\quad)-u(\quad,\quad)}{\quad}, \tag{8}$$

$$(D_1^2 u)(x_j,t_k) \approx \frac{u(\quad,\quad)-2\quad+\quad}{\quad}. \tag{9}$$

8. Un análogo discretizado de la ecuación original. Aproximamos ambos lados de la ecuación (1) por los valores de la función u en los puntos de la malla, usando (8) y (9):

$$\frac{u(x_j,\quad)-u(\quad,\quad)}{\quad} = \frac{u(\quad,\quad)-2\quad+\quad}{\quad}.$$

Despejamos $u(x_j,t_{k+1})$:

$$u(x_j,t_{k+1}) = u(\quad,\quad) - \frac{\quad}{h^2}\big(-u(\quad,\quad)+2\quad-\quad\big). \tag{10}$$

Las condiciones de frontera significan que

$$u(x_0,t_k) = \quad, \qquad u(\quad,\quad) = \quad,$$

y la condición inicial se convierte en

$$u(\quad,\quad) = f_j.$$

Denotemos por U al vector $[u(x_j,t_k)]_{j=1}^{n-1}$, y por V al vector $[u(x_j,t_{k+1}]_{j=1}^{n-1}$. Entonces V se puede expresar a través de U usando la matriz de Toeplitz tridiagonal T_{n-1} con entradas diagonales $-1,2,-1$:

$$V = U + ??? * U.$$

Falta programar esta receta en una función `solveheatfindifexplicit`.