Solución de una clase simple de problemas de frontera usando un esquema más preciso con diferencias finitas

Requisitos. Es necesario resolver los ejercicios del tema “Solución de una clase simple de problemas de frontera usando diferencias finitas” y programar las funciones correspondientes.

1. Una clase simple de problemas de frontera. Consideremos problemas de la forma

$$X''(t) = F(t), \qquad X(0) = X(1) = 0,$$

donde F es una función dada y X es la función incógnita.

2. Fórmula más precisa para la segunda derivada. Supongamos que X es una función bastante suave y escribamos su expansión de Taylor alrededor de un punto general t, denotando por h el incremento del argumento:

$$X(t+h) = X(t) + \square\, h + \frac{\square\, h^2}{2} + \frac{\square}{6} + \frac{X^{(4)}(t)\,\square}{24} + \frac{\square}{\square} + O(h^6).$$

Escriba expansiones similares para $X(t+2h)$, $X(t-h)$, $X(t-2h)$, y encuentre coeficientes α, β, γ tales que en la expresión

$$\alpha X(t-2h) + \beta X(t-h) + \gamma X(t) + \beta X(t+h) + \alpha X(t+2h)$$

se cancelen todos los términos con h^0, h^1, h^3, h^4 y h^5. Haga los cálculos correspondientes aparte (se recomienda hacer estos cálculos a mano en papel). Despejando $X''(t)$ de la expresión obtenida deduzca una fórmula de la siguiente forma:

$$X''(t) = -\frac{\alpha X(t-2h) + \beta X(t-h) + \gamma X(t) + \beta X(t+h) + \alpha X(t+2h)}{h^2} + O(h^4).$$

La respuesta final (después de calcular α, β, γ y factorizar el denominador común):

$$X''(t) = -\frac{X(t-2h) - \square\, X(t-h) + \square}{\square\, h^2} + O(h^4). \qquad (1)$$

3. Pasamos de las funciones a los vectores que constan de sus valores en una malla uniforme (repaso). Sea n un número entero positivo. Dividimos el intervalo cerrado $[0,1]$ en n partes iguales y denotemos por $t_0,\ldots,t_n$ a los *nodos* correspondientes:

$$t_0=\frac{0}{n}=0,\quad t_1=\frac{\square}{n},\quad t_2=\frac{2}{n},\quad \ldots,\quad t_k=\frac{\square}{n},\quad \ldots,\quad t_n=\frac{n}{n}=1.$$

Denotemos por x_k al valor de la función incógnita X en el punto t_k, y por f_k al valor de la función dada F en el punto t_k:

$$x_k=X(t_k)=X\left(\frac{\square}{\square}\right),\qquad f_k=F(\square)=F\left(\frac{\square}{\square}\right).$$

En vez de buscar la función $\square$, vamos a buscar el vector $[\square]_{k=0}^{n}$ de sus valores en los puntos t_k. Las condiciones de frontera $X(0)=X(1)=0$ significan que las siguientes componentes del vector x son cero:

$$\square=0,\qquad \square=0. \tag{2}$$

4. Ecuaciones en los nodos lejanos de la frontera. Vamos a reemplazar la ecuación diferencial $X''(t)=F(t)$ por su análogo discreto.
Primero notamos que si $t=t_k=k/n$ y $h=1/n$, entonces

$$t+h=\frac{\square}{\square}=\underbrace{\square}_{t_{???}},\quad t+2h=\underbrace{\square}_{t_{???}},\quad t-h=\underbrace{\square}_{t_{???}},\quad t-2h=\underbrace{\square}_{t_{???}}.$$

Consideremos la ecuación $X''(t)=F(t)$ solamente en los puntos t_k, $k\in\{2,\ldots,n-2\}$. En el lado derecho tenemos $F(t_k)$, lo cual es $\underbrace{\square}_{f_{???}}$.

Aproximamos $X''(t_k)$ por la fórmula (1):

$$X''(t_k)=-\frac{x_{k-2}-\square\, x_{k-1}+\square}{12/n^2}+O\left(\frac{1}{n^4}\right), \tag{3}$$

Entonces la ecuación $X''(t)=F(t)$ se transforma en

$$-\frac{x_{k-2}-\square\, x_{k-1}+\square}{12/n^2}+O\left(\frac{1}{n^4}\right)=\square.$$

Multipliquemos ambos lados por $-12/n^2$ y omitamos el término residuo (por supuesto, la ecuación nueva no será exactamente equivalente a la ecuación original). Obtenemos

$$x_{k-2}-\square=-\frac{12}{n^2}\square\qquad (k\in\{\square,\ldots,\square\}). \tag{4}$$

5. Ecuaciones en los nodos cercanos a la frontera. Se puede demostrar que

$$X''(t) \approx -\frac{-11X(t-h) + 20X(t) - 6X(t+h) - 4X(t+2h) + X(t+3h)}{12h^2}, \tag{5}$$

y

$$X''(t) \approx -\frac{-11X(t+h) + 20X(t) - 6X(t-h) - 4X(t-2h) + X(t-3h)}{12h^2}. \tag{6}$$

Utilice (5) para aproximar $X''(t_1)$, tomando en cuenta que $X(t_1 - h) = x_0 = 0$:

$$X''(t_1) \approx -\frac{20x_1 - 6x_2 + \boxed{}}{12/n^2}.$$

Entonces la ecuación que corresponde al nodo 1 es

$$20x_1 - 6x_2 + \boxed{} = \boxed{}. \tag{7}$$

De manera similar, utilice la fórmula (6) para aproximar $X''(t_{n-1})$, tomando en cuenta que $X(t_{n-1} + h) = X(\boxed{}) = x_n = 0$:

$$X''(t_{n-1}) \approx -\frac{20x_{n-1} - \boxed{}}{12/n^2}.$$

Entonces la ecuación que corresponde al nodo $n-1$ es

$$20x_{n-1} - \boxed{} = \boxed{}. \tag{8}$$

6. Sistema de ecuaciones. Del Problema 1 hemos pasado a un sistema de $n-1$ ecuaciones (7), (4), (8), para las incógnitas $x_1, \ldots, x_{n-1}$:

$$\begin{aligned} k = 1: &\quad 20x_1 - 6x_2 + \qquad\qquad\qquad\qquad\qquad\qquad = \\ 2 \le k \le n-1: &\quad \\ k = n-1: &\quad \end{aligned}$$

Se puede demostrar (no lo vamos a hacer) que el nuevo sistema de ecuaciones tiene una única solución. Notamos que el vector $x = \left[x_0, \ldots, x_n\right]^\top$ **no coincide exactamente** con los valores de la solución X del Problema 1 en los puntos t_k.

7. El sistema de ecuaciones en la forma matricial. Consideremos el caso $n = 9$. Escribimos la ecuación (4) para $k = 3$:

$$x_1 + \boxed{} = -\frac{12}{81} \boxed{}.$$

De manera similar se escriben las ecuaciones para $k = 3, 4, 5, 6, 7$. Los casos extremos $k = 1$ y $k = 8$ son especiales. Pasamos el sistema de ecuaciones del Problema 6 a la forma matricial

$$Mx = b,$$

donde M es una matriz cuadrada de tamaño $\boxed{} \times \boxed{}$ y B es un vector de longitud $\boxed{}$:

$$M = \left[\begin{array}{c|c|c|c|c|c|c|c}
 & & & & 0 & 0 & 0 & 0 \\ \hline
 & & & & 0 & 0 & 0 & 0 \\ \hline
 & & & & & 0 & 0 & 0 \\ \hline
0 & & & & & & 0 & 0 \\ \hline
0 & 0 & & & & & & 0 \\ \hline
0 & 0 & 0 & & & & & \\ \hline
0 & 0 & 0 & 0 & & & & \\ \hline
0 & 0 & 0 & 0 & & & &
\end{array}\right], \qquad b = -\frac{12}{81}\left[\begin{array}{c}
 \\ \hline
\\ \hline
\\ \hline
\\ \hline
\\ \hline
\\ \hline
\\ \hline

\end{array}\right].$$

La matriz M es casi una matriz de Toeplitz pentadiagonal, pero el primer renglón y el último tienen otra estructura.

8. Construcción de matrices de banda en Matlab con el comando spdiags (repaso). La siguiente matriz A tiene tres diagonales, las cuales se pueden guardar juntas formando una matriz D (hemos puesto 99 en los lugares que no se toman en cuenta):

$$A = \begin{bmatrix} 11 & 12 & 0 & 0 \\ 21 & 22 & 23 & 0 \\ 0 & 32 & 33 & 34 \\ 0 & 0 & 43 & 44 \end{bmatrix}, \qquad D = \begin{bmatrix} 21 & 11 & 99 \\ 32 & 22 & 12 \\ 43 & 33 & 23 \\ 99 & 44 & 34 \end{bmatrix}.$$

Estas tres diagonales en Matlab se numeran como $-1, 0, 1$. Construimos la matriz D y la matriz A en el formato de matriz disperza, luego observamos A en la forma completa:

```
D = [21, 11, 99; 32, 22, 12; 43, 33, 23; 99, 44, 34]
A = spdiags(D, -1 : 1, 4, 4)
full(A)
```

9. Construcción de la matriz M en el formato de matriz dispersa. Complete los siguientes comandos para construir la matriz M de tamaño 8×8 escrita en el Ejercicio 7:

```
D = repmat([1, -16, ???, ???, ???], ???, ???)
M = spdiags(D, -2 : 2, ???, ???)
disp(full(M))
M(1, 1 : 4) = [20, ???, ???, ???];
M(8, 5 : 8) = [???, ???, ???, 20];
disp(full(M))
```

10. Solución de una clase especial de problemas de frontera con el método de diferencias finitas. En la siguiente función se supone que `f` es un apuntador a una función.

```
function [t, x] = solvesimplebvpviafindif5(f, n),
   D = repmat([???, ???, ???, ???, ???], ???, ???);
   M = spdiags(???, ???, ???, ???);
   M(1, ??? : ???) = [???, ???, ???, ???];
   M(n - 1, ??? : ???) = [???, ???, ???, ???];
   t = (1 : ???)' / ???;
   b = - 12 * f(t) / ???;
   x = T \ b;
   t = [0; t; 1];
   x = [0; x; 0];
end
```

Los últimos dos operadores sirven para resolver el sistema (con algún método que elige Matlab o GNU Octave) y completar la solución con los valores nulos en la frontera.

11. Pruebas. La prueba se puede agregar a la función que hace pruebas de otros métodos.

```
function [] = testsolvesimplebvp(n),
   f1 = @(t) exp(t); # se puede usar otro ejemplo
   xsol = @(t) ???; # la solucion exacta
   xexact = xsol(t);
   # con tres diagonales, de la clase anterior:
   [t, x3] = solvesimplebvpviafindif(f1, n);
   display(norm(x3 - xexact, inf));
   # con cinco diagonales:
   [t, x5] = solvesimplebvpviafindif5(f1, n);
   display(norm(x5 - xexact, inf));
end
```