UNIVERSIDAD DE MURCIA

Departamento de Matemáticas
Cálculo Numérico en una Variable
Curso 2021/22

Prof. S. Sánchez-Pedreño

Práctica 4

28 y 31 de octubre de 2022

Para realizar esta práctica necesitas descargar algunos script de Octave que he subido al Aula Virtual. Si no lo has hecho ya, descarga el archivo `Practica4Anadir_Biblioteca.zip`, situado en la carpeta `CodigoPracticas` de los Recursos del AV. Descomprímelo y guarda su contenido en la carpeta `Biblioteca` que debe estar contenida en tu carpeta de trabajo con Octave de tu ordenador personal.

El objetivo de esta práctica es experimentar con la interpolación por splines. El tercer ejercicio vuelve sobre los polinomios interpoladores en la forma de Lagrange. Este tercer ejercicio, que iniciaremos si es posible en la sesión de prácticas, formará parte de la primera enterega de prácticas.

Para comprender bien las funciones de cálculo de los splines, en particular sus variables de entrada y salida, debes tener en cuenta que para estas funciones construidas en Octave, un spline se identifica con los nodos en los que interpola $(x_0, \ldots, x_n)$ y los coeficientes de los distintos polinomios cúbicos que lo componen:

$$p_i(x) = a_i + b_i(x - x_i) + c_i(x - x_i)^2 + d_i(x - x_i)^3, \quad i = 0, \ldots, n-1.$$

donde $p_i(x)$ es la pieza polinomial que forma el spline definida en $[x_i, x_{i+1}]$. Si el spline es sujeto necesitas proporcionar además los valores de la derivada en los puntos extremos x_0 y x_n. Ten en cuenta, finalmente, que para evaluar un spline en una abscisa se deben conocer los nodos, pues la evaluación necesita conocer en qué intervalo $[x_i, x_{i+1}]$ se encuentra dicha abscisa.

- `splineNatural.m`, devuelve los coeficientes b_i, c_i, d_i (en la forma `[b,c,d]`) de los polinomios cúbicos que componen el spline natural. Los argumentos que espera son: el vector de nodos(`x`) y el vector de ordenadas (`a`).
- `splineSujeto.m`, devuelve los coeficientes b_i, c_i, d_i (en la forma `[b,c,d]`) de los polinomios cúbicos que componen el spline sujeto. Se le suministra como variables, naturalmente, además de las coordenadas de los puntos que se interpolan (`x` y `a`), la derivada en los puntos inicial y final (`tan_ini` y `tan_fin`).
- `splineEval.m`, devuelve la evaluación de un spline en una abscisa (que puede ser un vector de abscisas). necesita como argumentos: los nodos (`x`) y ordenadas (`a`), así como los vectores de coeficientes (`b`, `c`, `d`) y el vector de abscisas donde se evalúa (`t`).
- `splineTabla.m`, es una función auxiliar, en general innecesaria, que únicamente produce una salida impresa de los polinomios que componen el spline: necesita por tanto todos los datos que lo definen (`x`, `a`, `b`, `c` y `d`).

Antes de utilizar cualquiera de estas funciones, asegúrate de entender qué argumentos espera, su tipo y significado, así como qué salidas produce, su significado, orden y formato. Importante: ¡los nodos deben pasarse ordenados!

1. *Ejemplos sencillos*

   a) Considera la lista de puntos $\{(0,1),(3,0),(2,2),(1,4)\}$. Representa en un panel de dibujo el spline cúbico natural que interpola los puntos de la lista. ¿Serías capaz de dibujar de distinto color cada una de las tres piezas que componen el spline?

b) Representa en otra ventana gráfica los splines cúbicos sujetos $S(x)$ que interpolan los mismos puntos y que tienen derivadas en los extremos $S'(0) = a, S'(3) = b$, para distintos valores de $(a, b) = (0, -1), (1, 5), (-2, -5)$ y $(-5, -1)$.

c) Considera los puntos $\{(0, 0), (1, 2), (2, 4), (5, 10)\}$. ¿Qué función spline natural debes encontrar que interpole dichos puntos? Responde a la cuestión antes de verificarlo con un pequeño programa... Es una forma de verificar que el programa de cálculo de splines funciona correctamente.

d) Responde a la misma pregunta anterior para un spline sujeto que interpola los puntos $(0, 0)$, $(1, 1)$, $(2, 8)$, $(3, 27)$ y tiende derivada 0 en $x = 0$ y 27 en $x = 3$.

2. *Curvas "suaves" en el plano obtenidas mediante el cálculo de splines: aproximar una espiral de Arquímedes mediante splines cúbicos*

a) Consideramos una partición $\{\theta_i\}_{i=0}^{9}$ de 10 puntos equidistribuidos en el intervalo $[0, 4\pi]$. Calculamos los 10 puntos correspondientes en la espiral arquimediana, que tiene por ecuación en coordenadas polares $\rho = 2\theta$, con $\theta \in [0, 4\pi]$. Así, las ecuaciones cartesianas de la espiral son:

$$x(\theta) = 2\theta \cos\theta, \quad y(\theta) = 2\theta \operatorname{sen}\theta.$$

b) Con las listas de puntos $\{(\theta_i, x(\theta_i))\}$ y $\{(\theta_i, y(\theta_i)\}$ calculamos sendos splines cúbicos naturales, que llamaremos $x(\theta)$, $y(\theta)$.

c) Ahora representaremos en un panel gráfico la propia espiral arquimediana anterior y su aproximación por splines $(x(\theta), y(\theta))$.

d) Realiza ahora la misma tarea con splines sujetos, añadiendo derivadas en los extremos: $x'(0) = 2$, $x'(4\pi) = 2$, $y'(0) = 0$, $y'(4\pi) = 8\pi$.

3. **Ejercicio que formará parte de la primera entrega de prácticas**

A estas alturas tenemos a nuestra disposición (en teoría) varias formas de calcular y evaluar un polinomio interpolador. Esta formas son (en términos generales):

- Calcular los coeficientes del polinomio en su forma de Newton y evaluarlo mediante el método de Horner.
- Calcular los coeficientes de la forma de Lagrange (que es un polinomio escrito en la forma habitual $a_n x^n + \cdots + a_1 x + a_0$) y evaluarlo mediante el método de Horner.
- Evaluar directamente el polinomio mediante la fórma de Lagrange.

La primera de las formas anteriores ya es conocida y la tenemos programad; ha sido la forma habitual de cálculo, hasta ahora. El objetivo de este ejercicio es implementar en Octave el segundo procedimiento.

a) Escribe una función denominada `factLagrange.m`, que guardarás en `/Biblioteca`, que realice la tarea siguiente: a partir de su único argumento `x`, que es el vector de nodos de interpolación $(x = \{x_0, \ldots, x_n\})$, devuelve una matriz cuya fila k-ésima contiene los coeficientes del factor de Lagrange $(k = 0, \ldots, n)$:

$$L_k(x) = \prod_{j=0\, j\neq k}^{n} \frac{x - x_j}{x_k - x_j}$$

Para facilitar la tarea usaremos la función de Octave:

`conv(p,q)`

que devuelve el polinomio producto de los polinomios `p` y `q` (¡recuerda cómo se representan los polinomios en Octave!).

Departamento de Matemáticas. Campus de Espinardo. 30100 Espinardo Murcia
e-mail: pedreno@um.es

b) Crea en `/Biblioteca` la función `interpolLagrange.m` que espera como argumentos la matriz de coeficientes de los factores de Lagrange y el vector de ordenadas a interpolar, y devuelve los coeficientes del polinomio interpolador en la forma de Lagrange:

$$p(x) = \sum_{k=0}^{n} y_k L_k(x)$$

c) Para verificar que la implementación es correcta: escribe en el script `Ejercicio3_1.m` el cálculo de los coeficientes del polinomio interpolador en los conjuntos de puntos

$$\{(0,0),(1,2),(2,4),(5,10)\} \quad \text{y} \quad \{(0,0),(1,1),(2,8),(3,27)\}.$$

¿Coinciden exactamente los polinomios que produce el script con los esperados? Si no es así, ¿hay alguna explicación o hay errores en el script?

d) Una vez creadas las funciones anteriores disponemos ya del polinomio interpolador en la forma de Lagrange: para evaluarlo en un vector de abscisas, `z`, utilizaremos la siguiente función de Octave

`polyval(p,z)`

que devuelve el vector de las evaluaciones del polinomio `p` en `z`: este cálculo está implementado en Octave mediante el método de Horner (así que no vale la pena que lo implementemos nosotros...)

Para comparar los resultados de ambas formas de cálculo utilizaremos la interpolación de la función $\text{sen}(x)$ en el intervalo $[0, 2\pi]$, mediante nodos equidistribuidos: $x_0 = a$, $x_k = a + k\frac{2\pi}{n}$. Para esta función sabemos que los polinomios interpoladores $p_n(x)$ convergen uniformemente a $f(x)$, por lo que si aumentamos considerablemente el número de puntos a interpolar deberíamos aproximar más y más a la función...

e) Crea el script `Ejercicio3_2.m` que para un valor de n realice las tareas siguientes:

- en una primera ventana gráfica incluya las gráficas de f y del polinomio interpolador en la forma de Newton;
- en una segunda ventana gráfica incluya las gráficas de f y del polinomio interpolador en la forma de Lagrange;
- en otra ventana incluya la gráfica del valor absoluto de la diferencia de los dos polinomios interpoladores;
- nos dé como salida en la ventana de comandos el máximo del valor absoluto de la diferencia anterior (calculado sobre una malla de puntos en el intervalo $[0, 2\pi]$).

Ejecuta el script anterior para distintos valores de n, por ejemplo: 5, 10, 15 y otros mayores, por ejemplo 20, 30, 40. ¿Qué observas? Observa que las diferencias grandes aparecen cerca de los extremos del intervalo, ¿tiene esto que ver con el fenómeno de Runge? Comenta lo que consideres adecuado sobre la estabilidad de los cálculos en una u otra forma.