MAT2605, Cálculo Científico I, 2021-2

Tarea 3

Entrega para el 13 de septiembre de 2021

Pueden discutir los problemas con sus compañeros. Sin embargo, sus entregas, tanto la parte escrita como los códigos, deben ser individuales. **No** está permitido copiar las respuestas de alguien más ni dejar que otros copien sus respuestas. Su entrega debe estar compuesta por un único archivo en formato `.pdf` que incluya todas las respuestas y todos los gráficos que se están pidiendo, junto con un archivo `.zip` que incluya todos los archivos `.m` que produjeron y utilizaron durante esta tarea. Estos dos archivos deben ser subidos en Canvas. Si hay más de una línea en un gráfico, utilice distintos estilos de línea o distintos colores para diferenciarlas e incluya leyendas. No se olvide de nombrar todos sus ejes y líneas en las leyendas.

1. Se sabe que para $f \in C^{n+1}([a,b])$ (Teorema 3.3 del libro),

$$\begin{aligned}\|f-P_n\|_\infty := \max_{a\le x\le b}|f(x)-P_n(x)| &\le \max_{a\le x\le b}\left|\frac{f^{(n+1)}\big(\xi(x)\big)}{(n+1)!}\prod_{k=0}^{n}(x-x_k)\right| \\ &\le \max_{a\le x\le b}\left|\frac{f^{(n+1)}(x)}{(n+1)!}\right|\max_{a\le x\le b}\left|\prod_{k=0}^{n}(x-x_k)\right| = \frac{\|f^{(n+1)}\|_\infty}{(n+1)!}\|W_n\|_\infty\,,\end{aligned} \tag{$*$}$$

donde $P_n(x)$ es el polinomio interpolante de grado a lo sumo n pasando exactamente por los valores tabulados $(x_0,f(x_0)),(x_1,f(x_1)),\ldots,(x_n,f(x_n))$, con $x_k\in[a,b]$ para todo $k=0,1,\ldots,n$, $\xi(x)\in[a,b]$, y donde $W_n(x):=\prod_{k=0}^{n}(x-x_k)=(x-x_0)(x-x_1)\cdots(x-x_n)$ es un polinomio de grado $n+1$.

9 pts. (a) (i) Suponga que una función $f\in C^{n+1}[a,b]$ es interpolada con un polinomio cuadrático usando los nodos de interpolación igualmente espaciados $x_0=a$, $x_1=\frac{a+b}{2}$, y $x_2=b$. Deduzca que

$$\|W_n\|_\infty = \max_{a\le x\le b}\left|\prod_{k=0}^{n}(x-x_k)\right| = \max_{a\le x\le b}\left|(x-a)\big(x-\tfrac{a+b}{2}\big)(x-b)\right| = \tfrac{1}{12\sqrt{3}}(b-a)^3\,.$$

3 pts. (ii) Sea $f(x)=\arctan(x)$ y considere su interpolante cuadrático en los puntos $x_0=-1$, $x_1=0$ y $x_2=1$. Usando a cota en $(*)$ y la computación de la parte (i), deduzca que el error supremo en $[-1,1]$ está acotado por

$$\|f-P_n\|_\infty \le \frac{2}{9\sqrt{3}}\,.$$

(b) Escoger nodos equidistantes puede ser contraproducente cuando se interpolan ciertas funciones con un polinomio, especialmente cuando n es grande. Para estas funciones, el polinomio interpolante puede tener oscilaciones grandes que inclusive pueden subir en magnitud a medida que más nodos de interpolación son agregados (n crece). Esto va en contra de la intuición, pues uno esperaría que agregar más nodos interpolantes subiría la calidad de la aproximación, pero este no es el caso para funciones como la función de Runge, $f(x)=1/(1+25x^2)$. Este comportamiento usualmente es referido como el fenómeno de Runge, y su existencia es sugerida (pero no asegurada) por $(*)$, como se verá continuación.

7 pts. (i) Demuestre que $\|W_n\|_\infty=\max_{x\in[a,b]}|W_n(x)|\le n!h^{n+1}$ cuando los nodos equidistan por h. *Pista*: Asuma primero que $x\in[x_0,x_1]$ y note que $|x-x_0|<h$, $|x-x_1|<h$ y $|x-x_k|<kh$ para todo $k>1$. ¿Cuál sería la cota para $|W_n(x)|$ en este caso? Repita este razonamiento para $x\in[x_1,x_2]$ y así sucesivamente para todos los $[x_j,x_{j+1}]$. ¿Cuál es la cota máxima para $|W_n(x)|$ entre todos los x?

6 pts. (ii) Para la función de Runge, $f(x)=1/(1+25x^2)$, se da que $\|f^{(n)}\|_\infty\le 5^n n!$ para $x\in[-1,1]$. Por lo tanto, con nodos equidistantes, (i) muestra que $\|W_n\|_\infty\le\left(\frac{2}{n}\right)^{(n+1)}n!$ (pues $h=\frac{2}{n}$) y $(*)$ implica que

$$\|f-P_n\|_\infty \le \left(\frac{10}{n}\right)^{n+1} n! =: A_n\,.$$

Usando que $\left(\frac{n+1}{n}\right)^n\le 3$ para todo $n\in\mathbb{N}$, muestre que $\frac{A_{n+1}}{A_n}>2$ para $n\ge 2$, y usando esto muestre que $\lim_{n\to\infty}A_n=\infty$. Como consecuencia, a medida que más nodos interpolantes equidistantes son agregados, la cota superior sigue creciendo. Desde un punto de vista teórico, esto no garantiza que la aproximación misma vaya a diverger, pues es solo una cota superior sobre el error, pero sí es sugestivo de que esto podría suceder, cosa que efectivamente ocurre con la función de Runge.

2. Sea $f : [-1,1] \to \mathbb{R}$ una función continua, $x_0, \ldots, x_n \in [-1,1]$ nodos distintos, y $f(x_0), \ldots, f(x_n)$ muestras de la función. La idea es construir el polinomio interpolante de f usando diferencias divididas. Cuando muestras adicionales $f'(x_0), \ldots, f'(x_n)$ son dadas, la idea es construir el poliniomio interpolante de Hermite de f usando una versión modificada de diferencias divididas.

4 pts. (a) En general, el polinomio de Newton toma la forma

$$p(x) = f[x_0] + f[x_0, x_1](x - x_0) + \cdots + f[x_0, \ldots, x_n](x - x_0) \cdots (x - x_{n-1}). \qquad (\dagger)$$

Asumiendo que las diferencias divididas ya han sido calculadas, muestre que toma $\mathcal{O}(n^2)$ flops evaluar $p(x)$ en esta forma.

4 pts. (b) Identifique las computaciones repetidas innecesarias en (†) y reescriba (†) para evitarlas. ¿Cuántos flops toma evaluar $p(x)$ ahora?

4 pts. (c) Un nuevo dato es obtenido: $f'(x_n)$ en x_n. Escriba el nuevo polinimio $p(x)$ actualizando (†) tal que $p(x_j) = f(x_j)$ para $0 \leq j \leq n$ y $p'(x_n) = f'(x_n)$. ¿Cuántos flops toma actualizar la fórmula (acuérdese de contar el costo de actualizar las diferencias divididas)?

4 pts. (d) Para practicar encontrar un polinomio interpolante a mano, complete la tabla de diferencias divididas de Newton para los puntos $(2, 2067)$, $(4, 3601)$ y $(5, 4581)$. Escriba el polinomio cuadrático resultante y expanda la expresión para obtenerlo en su forma estándar $ax^2 + bx + c$.

7 pts. (e) Escriba un programa llamado `DividedDifferences(x,fx)`, donde `x` es un vector de nodos y `fx` es el vector de evaluaciones de f en los nodos, que calcule la tabla de diferencias divididas de Newton. Úsela para calcular y luego graficar (en `xplot=-1:0.01:1`) el polinimio interpolante de grado 5 de $\cos(2x)$ en los nodos $x_j = -1 + \frac{2j}{n}$ para $j = 0, 1, \ldots, n = 5$.

7 pts. (f) Escriba un programa llamado `HermiteDividedDifferences(x,fx,dfx)`, donde `x` es un vector de nodos, `fx` es el vector de evaluaciones de f en los nodos, y `dfx` es el vector de derivadas de f en los nodos (del mismo tamaño que `fx` y `x`), que calcula la tabla modificada de diferencias divididas. Úsela para calcular y luego graficar (en `xplot=-1:0.01:1`) el polinimio interpolante de Hermite de grado $2n + 1 = 11$ de la función $f(x) = 1/(1 + 25x^2)$ en los nodos equidistantes $x_j = -1 + \frac{2j}{n}$ para $j = 0, 1, \ldots, n = 5$. Compárelo con los interpolantes polinomiales de grado $n = 5$ y $n = 11$ calculados usando `DividedDifferences` y grafíquelos (en `xplot=-1:0.01:1`) junto con el interpolante de Hermite.

3 pts. 3. (a) Una spline cúbica natural S de una función f es definida en $[1, 4]$ por

$$S(x) = \begin{cases} S_0(x) = b_0(x-1) + d_0(x-1)^3, & 1 \leq x < 3, \\ S_1(x) = 1 + b_1(x-3) - \frac{3}{4}(x-3)^2 + d_1(x-3)^3, & 3 \leq x \leq 4. \end{cases}$$

Si $S(x)$ interpola los datos $(1, 0), (3, 1), (4, 0)$, encuentre b_0, d_0, b_1, y d_1.

8 pts. (b) Escriba una rutina `Scf=SplineConstruction(x,fx)` que dada los datos de interpolación en $x_0, \ldots, x_n$ produce los coeficientes asociados a la spline cúbica natural S tal que $S(x_j) = f(x_j)$, donde $f(x_0), \ldots, f(x_n)$ es un conjunto de muestras (ya conociendo el conjunto de $f(x_j) = a_j$'s, construya un sistema lineal y soluciónelo con el operador \ de MATLAB para obtener un vector de c_j's y luego calcule los coeficientes $d_j = \frac{1}{3h_j}(c_{j+1} - c_j)$ y $b_j = \frac{1}{h_j}(a_{j+1} - a_j) - \frac{h_j}{3}(2c_j + c_{j+1})$ para $j = 0, \ldots, n-1$ donde $h_j = x_{j+1} - x_j$, para finalmente producir una matriz `Scf` de tamaño $n \times 4$ donde las columnas corresponden a los a_j's, b_j's, c_j's y d_j's respectivamente). Construya una rutina `S=EvaluateSpline(x,Scf,xplot)` que evalúe la spline en `xplot` y úsela para graficar (en `xplot=-1:0.01:1`) la spline natural de $f(x) = 1/(1 + 25x^2)$ pasando por los nodos $x_j = -1 + \frac{2j}{n}$ para $j = 0, 1, \ldots, n = 5$.

4 pts. (c) Se dice que una matriz A es de *diagonal estrictamente dominante* si $|A_{ii}| > \sum_{i \neq j} |A_{ij}|$ para todo i (i.e. que la suma de valores absolutos de una fila con excepción al elemento diagonal es

estrictamente menor al valor absoluto del componente diagonal de esa misma fila). Demuestre que una matriz de diagonal estrictamente dominante es invertible.
Pista: Proceda por contradicción asumiendo que es singular, por lo que debe existir un vector $c \neq 0$ tal que $Ac = 0$, y luego mire el i-ésimo componente del vector Ac, donde i es el índice de la fila para el que $|c_i|$ es el máximo entre los $|c_j|$'s.

2 pts. (d) Use (c) para demostrar que la spline cúbica natural, $S(x)$, interpolando $f(x_0), \ldots, f(x_n)$ en los nodos $x_0 < x_1 < \ldots < x_{n-1} < x_n$ es única (indique donde usa que los x_j's deberían ser distintos y ordenados).

4. Usted probablemente ha visto millones de curvas de Bézier en su vida. La razón es que las fuentes (tipo de letras) que ve todos los días están compuestas de muchas de estas curvas, y es el uso de estas curvas que permite mantener la calidad de la fuente a medida que hace zoom en su pantalla. Las fuentes PostScript (desarrolladas por Adobe) usan curvas cúbicas de Bézier, mientras que la fuentes TrueType (desarrolladas por Apple) usan curvas cuadráticas de Bézier. Las fuentes OpenType (desarrolladas por Microsoft y Adobe) trataron de unificar ambos formatos. Fuentes basadas en curvas cúbicas de Bézier son preferidas por profesionales. En este ejercicio, usted va a construir "glifos" de PostScript.

10 pts. (a) Escriba una rutina `v=BezierCurve(v0,c0,c1,v1,t)` que encuentre la curva de Bézier empezando en el vector 2×1 `v0` $= \binom{v_{0x}}{v_{0y}}$ en $t = 0$, acabando en `v1` $= \binom{v_{1x}}{v_{1y}}$ en $t = 1$, y determinada por los puntos de control `c0` $= \binom{c_{0x}}{c_{0y}}$ y `c1` $= \binom{c_{1x}}{c_{1y}}$. La salida debería ser la matriz $2 \times (n+1)$ `v` $= \begin{pmatrix} v_x(t_0) & \cdots & v_x(t_n) \\ v_y(t_0) & \cdots & v_y(t_n) \end{pmatrix}$ que resulta de evaluar la curva de Bézier en el vector paramétrico $1 \times (n+1)$ `t` $= (t_0, \ldots, t_n)$. La curva cúbica de Bézier es dada por

$$\vec{v}(t) = (1-t)^3 \vec{v}_0 + 3(1-t)^2 t \vec{c}_0 + 3(1-t) t^2 \vec{c}_1 + t^3 \vec{v}_1 \,. \tag{‡}$$

Adicionalmente, use la rutina adjunta `Hermite` (o `HermiteDividedDifferences` del problema 2) para construir los polinmios cúbicos de Hermite $v_x(t)$ y $v_y(t)$ que satisfacen

$$\begin{aligned} v_x(0) &= v_{0x}, \quad v_x(1) = v_{1x}, \quad v_x'(0) = 3(c_{0x} - v_{0x}), \quad v_x'(1) = 3(v_{1x} - c_{1x}), \\ v_y(0) &= v_{0y}, \quad v_y(1) = v_{1y}, \quad v_y'(0) = 3(c_{0y} - v_{0y}), \quad v_y'(1) = 3(v_{1y} - c_{1y}). \end{aligned}$$

Use `subplot` para graficar (con `t=0:0.01:1`) la curva cúbica de Bézier que resulta de `BezierCurve` y el polinomio parámetrico de Hermite $(v_x(t), v_y(t))$ encontrado (una en cada subgráfico) para los datos `v0=[0;0]`, `v1=[1;0]`, `c0=[2;1]` y `c1=[-1;1]`. Se deberían ver iguales.

5 pts. (b) Curvas de Bézier de grado arbitrario $n \in \mathbb{N}$ pueden ser definidas por $n+1$ puntos: los extremos $\vec{p}_0 = \vec{v}_0$ y $\vec{p}_n = \vec{v}_1$, y los $n-1$ puntos de control $\vec{p}_j$ para $j = 1, \ldots, n-1$. Basándose en la expresión (‡) (donde $\vec{p}_0 = \vec{v}_0$, $\vec{p}_1 = \vec{c}_0$, $\vec{p}_2 = \vec{c}_1$ y $\vec{p}_3 = \vec{v}_1$), ¿puede adivinar la expresión para una curva de Bézier general de grado $n \in \mathbb{N}$?

3 pts. (c) El archivo adjunto `GlyphData.mat` contiene datos asociados a los glifos (caracteres) de la fuente gratis Nimbus Roman No. 9 L, frecuentemente usada como una alternativa para la fuente icónica Times New Roman. Los glifos forman una palabra y pueden ser graficados usando el archivo adjunto `PlotBezierGlyphs.m`, que llama a su rutina `BezierCurve` (grafica las curvas de Bézier asociadas a los glifos con `t=0:0.01:1` para cada curva). ¿Cuál es la palabra y qué significa? ¿Cuántas curvas de Bézier componen el tercer glifo de la palabra?