UNIVERSIDAD DE MURCIA

Departamento de Matemáticas
Cálculo Numérico en una Variable
Curso 2023/24

Prof. S. Sánchez-Pedreño

Práctica 3

23 y 27 de octubre de 2023

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 `Practica3_AnadirBiblioteca.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. Este archivo contiene el código de todas las funciones descritas a continuación.

Nota. Quizás haya modificado algunas funciones desde la última vez que las subí al Aula Virtual, si ya descargaste las funciones siguientes mejor reemplázalas por las versiones en el archivo comprimido `Practica3_AnadirBiblioteca.zip`.

El objetivo general de esta práctica es experimentar con la interpolación polinomial, tanto de Lagrange como de Hermite, con diversas funciones. Para ello, en `Biblioteca` dispones de las funciones siguientes

- `dif_divNewton.m`, devuelve la tabla de diferencias divididas y los coeficientes del polinomio interpolador en la forma de Newton. Espera como variables los vectores `x` e `y` de nodos y ordenadas, respectivamente.
- `dif_divHermite.m`, devuelve la tabla de diferencias divididas y los coeficientes del polinomio interpolador de Hermite en la forma de Newton. Espera como variables el vector `x` de los nodos y una celda `y` que contiene los datos para la ordenada y las derivadas en cada nodo. Para comprender cómo funcionan las celdas consulta los comentarios sobre Octave al final del ejercicio 1.

  Las dos funciones anteriores no son realmente necesarias: las he incluido para que tengáis una herramienta que muestre la tabla de diferencias divididas, que puede ser práctico para chequear algunos cálculos.
- `interpolNewton.m`, devuelve los coeficientes del polinomio interpolador en la forma de Newton. Espera las mismas variables que `dif_divNewton`.
- `interpolHermite.m`, devuelve los coeficientes del polinomio interpolador de Hermite en la forma de Newton y una lista de los nodos repetidos, que será necesaria para evaluar el polinomio. Espera las mismas variables que `dif_divHermite`.
- `polyinterpolador_eval.m`, espera como variables el vector de coeficientes del polinomio en la forma de Newton, el vector de nodos (repetidos en el caso de que el polinomio a evaluar sea de Hermite) y el vector de abscisas donde se desea evaluar el polinomio. Devuelve el vector de evaluaciones.
- `polyinterpolador.m`, devuelve los coeficientes de un polinomio interpolador escrito en la forma estándar $a_n x^n + \cdots + a_1 x + a_0$. Espera como variables los vectores de coeficientes de la forma de Newton y el vector de nodos (repetidos si se trata de un polinomio de Hermite).
- `nodosCheby.m`, espera como variables los extremos del intervalo $[a, b]$ (en el orden natural) y un natural n. Devuelve los $n+1$ nodos de Chebyshev en $[a, b]$, es decir los ceros del polinomio de grado $n + 1$.

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.

1. *Dos ejemplos sencillos*

a) Crea el script `Ejercicio1.m` para construir los polinomios interpoladores correspondiente a los datos

x	1.5	2.7	3.1	-2.1	-6.6	11.0
y	0.0	1.0	-0.5	1.0	0.5	0.0

y a los datos

x	y	y'	y''
0	1	2	
1	0	1	1
2	3		
3	1	1	

b) En cada caso, representa en una gráfica el polinomio interpolador y los puntos de la tabla interpolada.

Comentarios sobre Octave.

Para la interpolación de Hermite debemos utilizar variables de tipo celda (`cell`). Las celdas se pueden definir directamente como los vectores o matrices, pero utilizando llaves (`{ }`) en lugar de corchetes (`[ ]`). Así, por ejemplo, si escribimos

```
y={[1,2],[0,1,1],[3],[1,1]}
```

estaríamos definiendo una celda con una fila y cuatro «columnas» (denominadas `y{1,1}`, `y{1,2}`, etc.), cada una de ellas contiene los vectores indicados entre las llaves (que son vectores fila en realidad, por ejemplo `y{1,2}` es de tamaño 1×3). Como ves podemos de esta forma definir objetos estructurados de tipo matriz, pero donde la longitud de las columnas (o filas) es distinto para cada fila (o columna); incluso podemos definir una componente de una celda como una matriz o una celda.

En Octave no es posible indexar sobre las componentes de una celda. Por ejemplo, con la definición anterior de la celda `y`, no podemos hacer `y{1:4}(1)`, que sería lo natural para recuperar la primera componente de cada una de las cuatro filas de `y`. Sí podemos indexar como `y{2:4}`, que devuelve las componentes 2 a 4 (que sería más lógico y válido denominar `y{:,2:4}`). También podemos recuperar una fila, por ejemplo, con `z=y{2}` obtenemos un vector `z` cuyo valor es `[0,1,1]`. Para recuperar por ejemplo la primera componente de cada componente haremos un bucle:

```
for i=1:length(y)
z(i)=y{i}(1)
endfor
```

2. *Fenómeno de Runge*

Considera la función de Runge, $f(x) = \frac{1}{1+x^2}$, sobre el intervalo $[-5, 5]$. Vamos a representar gráficamente distintos polinomios interpoladores para esta función, así como los errores en la aproximación de los polinomios a la propia función. Los tres primeros apartados los realizamos en un script llamado `Ejercicio2_1.m`, el apartado *d*) en `Ejercicio2_2.m` y el apartado *f*) en `Ejercicio2_3.m`.

a) Escribe un programa que, dado un vector de enteros positivos N, dibuje en una ventana la gráfica de la función de Runge y la gráfica de los polinomios interpoladores p_{N_i} de f en los $N_i + 1$ puntos equiespaciados que dividen al intervalo $[-5, 5]$ en N_i partes iguales, para cada componente N_i de N. Ejecútalo con el vector `N=[5,10,15]`. Puedes intentar hacerlo con valores mayores de N, por ejemplo `N=[20,30,40]`. Juega un poco con la orden `ylim`, para poder apreciar bien la gráfica de la función de Runge y los polinomios a su alrededor.

b) En el mismo programa anterior, introduce el código necesario para que dibuje en una segunda ventana las gráficas de las distintas funciones de error $|f(x) - p_N(x)|$. Utilizando la orden de Octave `max(x)`, que devuelve la mayor de las componentes del vector `x`, aproxima el error máximo, aplicando `max` a una muestra de 1000 puntos en el intervalo.

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

c) Realiza las mismas tareas anteriores pero considerando interpolación con nodos de Chebyshev, para los mismos números de puntos.

d) Ahora considera la interpolación de Hermite para valores `N=[5,10,15]`, añadiendo las condiciones sobre las derivadas siguientes:

1) Valor de las derivadas primera y segunda en -5 y 5
2) Valor de la derivada primera en -5, $-1/\sqrt{3}$, $1/\sqrt{3}$ y 5.

Añade la segunda ventana con los respectivos errores.

e) Analiza la aproximación que proporcionan los distintos polinomios interpoladores con los que hemos experimentado: para el análisis puedes variar los valores elegidos, por ejemplo, puedes hacer también `N=[20,30,40]` y también puedes cambiar el valor $1/\sqrt{3}$ por 2.5 (o cualquier otro).

f) El propio Runge demostró que, para nodos equidistribuidos, los polinomios $p_n(x)$ que interpolan a la función $f(x)$ anterior convergen uniformemente a $f(x)$ en el intervalo $[-c, c]$, con $c \approx 3.63$ y divergen fuera de $[-c, c]$. Considera ahora funciones $f_a(x) = \frac{1}{1+ax^2}$, para $a > 0$. Estudia gráficamente si el intervalo donde los polinomios convergen a $f_a(x)$ crece o decrece con el valor de a.

3. *Interpolación inversa: solución de una ecuación*

El propósito del ejercicio es aproximar la solución de la ecuación $\operatorname{sen}(x) = 0.75$, es decir, el valor de $x = \operatorname{arc\,sen}(0.75)$. Esto es un avance de un método de aproximación de soluciones que veremos con detalle en el tema 4. Si queremos calcular una solución de la ecuación $f(x_0) = 0$, lo que buscamos es la función inversa f^{-1} pues, de esa forma, $x_0 = f^{-1}(0)$. Puesto que no conocemos $f^{-1}(y)$ lo que hacemos es aproximarla por un polinomio interpolador.

a) Utiliza la lista de puntos de la gráfica de la función seno,

x	0	$\frac{\pi}{6}$	$\frac{\pi}{4}$	$\frac{\pi}{3}$	$\frac{\pi}{2}$
$\operatorname{sen}(x)$	0	0.5	$\frac{\sqrt{2}}{2}$	$\frac{\sqrt{3}}{2}$	1

para construir el polinomio interpolador $p(y)$ de la lista de puntos (y, x) que se obtiene al intercambiar las coordenadas de los puntos de la lista (x, y).

b) Considera la aproximación $x = \operatorname{arc\,sen}(0.75) \approx p(0.75)$.

c) Evalúa la aproximación conseguida (puedes calcular el valor de $\operatorname{arc\,sen}(75)$ que proporciona Octave, función `asin(x)`, donde `x` se expresa en radianes; también existe `asind` que devuelve el ángulo en grados sexagesimales).

d) Elimina de la lista de datos el último punto, ¿mejora la aproximación?