Foundations of Computational Math 2 Spring 2024

Programming assignment 1

General Task

Your task is to implement, analyze, and empirically test and demonstrate the capabilities of the polynomial interpolation methods discussed in the class:

1. monomial basis
2. Lagrange basis
3. Newton basis

You will need to test for two types of meshes:

- uniform meshes (i.e. equally spaced)
- Chebyshev meshes

Part routines

Horner's Rule for Newton Form

The Newton Form of the polynomial has the increasing degree form required to adapt Horner's Rule. For example, let $n = 4$,

$$\begin{aligned} p_4(x) &= \alpha_0 + \alpha_1(x - x_0) \\ &\quad + \alpha_2(x - x_0)(x - x_1) + \alpha_3(x - x_0)(x - x_1)(x - x_2) \\ &\quad + \alpha_4(x - x_0)(x - x_1)(x - x_2)(x - x_3) \end{aligned}$$

$$p_4(x) = \left[\left([\alpha_4(x - x_3) + \alpha_3](x - x_2) + \alpha_2\right)(x - x_1) + \alpha_1\right](x - x_0) + \alpha_0$$

Algorithm 1: Horner's Rule

Data: x
Result: $p_n(x) = s$
$s = \alpha_n$;
for $i = n - 1, \cdots, 0$ **do**
| $s = s(x - x_i) + \alpha_i$
end
$p_n(x) = s$

Tasks

Functions of Interest

- Function 1

$$\begin{aligned} f_1(x) &= (x - 2)^9 \\ &= x^9 - 18x^8 + 144x^7 - 672^6 + 2016x^5 - 4032^4 + 5376x^3 - 4608x^2 + 2304x - 512 \end{aligned}$$

- Function 2

$$f_2(x) = \frac{1}{1 + x^2}$$

Task 1: Polynomial interpolation

Test for both uniform meshes:

$$x_i = a + \frac{b-a}{n} i, \ 0 \le i \le n$$

and Chebyshev points of the first kind:

$$x_i = \cos \frac{(2i+1)\pi}{2n+2}, \ 0 \le i \le n$$

1. monomial basis: to solve the Vandermonde matrix, you may use any useful codes from FCM1 (e.g. LU factorization code), or you may use environments. **Be sure to cite the libraries/routines or environments/commands used in your solutions**.

2. Lagrange basis: to construct $p_n(x)$, you need to implement Barycentric 1 form and Barycentric 2 form (see attached notes).

3. Newton's basis: using **divided difference** to compute the coefficient α_i's and using Horner's rule to evaluate $p_n(x)$.

   - A routine to evaluate the divided differences required for the Newton form of an interpolating polynomial $p_n(x)$ using $O(n^2)$ operations and $O(n)$ space. The routine should compute the function values $f(x_i)$, $0 \le i \le n$ and the divided differences that are required. The output should be the divided differences and $f_i = f(x_i)$ values.
   - A routine based on the adapted Horner's rule to evaluate a polynomial $p_n(x)$ defined in terms of the Newton basis. (Note this also makes it possible to evaluate the monomial basis by taking all x_i to be the same value or any set of x_i with all or some of the values repeated.) It may be useful to give a vector of x values at which you need $p_n(x)$ rather than calling the routine once for each value.
   - Extra credit (10 points): order the set of distinct mesh points x_i, $0 \le i \le n$ into **decreasing order** $x_0 > x_1 > \cdots > x_{n-1} > x_n$ and **Leja ordering** (see attached notes), comparing the results with the increasing order.

Note: When applying Chebyshev points of the first kind as the mesh, it is assumed the interval of interest is $[-1, 1]$. This is clearly not the interval on which you will assess. So be sure to develop the code to use the appropriate change of variables:

$$x \in [a, b] \leftrightarrow z \in [-1, 1]$$
$$x(z) = \frac{a+b}{2} + z\frac{(b-a)}{2}$$

Requirement of Tests

Describe the design of your codes and discuss the complexity with respect to time and space. Empirically validate your routines. Your arguments for the correctness of your codes may include referencing their behaviors on the later tasks if appropriate, but your write-up for this task should summarize those behaviors leaving the details for the write-up of the later tasks.

1. Test interval: $[a, b] = [-4, 4]$
2. Design a test to show that your code works correctly.
3. Test for three different (not successive) n's. At least a 'small' n and a 'big' n.
4. Plot $f(x)$ and $p_n(x)$ from each method for each case.
5. Discuss the results for all methods. Investigate and discuss the results of convergence to achieve various level approximations as measured by $\|f(x) - p_n(x)\|_\infty$.

Submission of Results

Expected results comprise:

- A document describing your solutions as prescribed in the notes on writing up a programming solution posted on the Canvas.
- The source code, makefiles, and instructions on how to compile and execute your code including the Math Department machine used, if applicable.
- Code documentation should be included in each routine. (You don't need to paste your code in the writing report).
- All text files that do not contain code or makefiles must be PDF files. **Do not send Microsoft word files of any type.**

These results should be submitted by 11:59 PM on the due date. Submission of results is to be done via Canvas.