MA5233 Computational Mathematics

Assignment 2

Deadline: 1 October 2020, 7pm
Total marks: 20

1 Convergence of Chebyshev interpolation [6 marks]

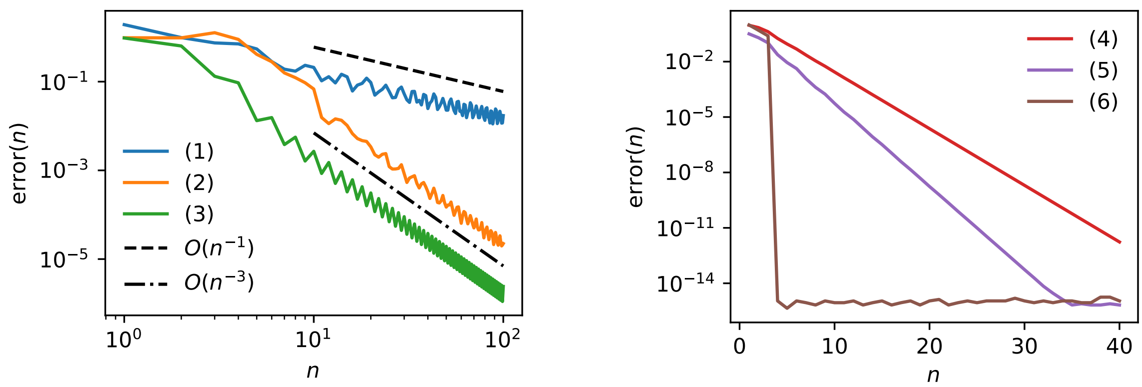
Match the below functions to the below Chebyshev interpolation convergence plots. Motivate your answers by listing the relevant properties of $f(x)$. [1 mark each]

Functions:

(a) $f(x) = |x|^3$ (c) $f(x) = \left|1 - x^3/4\right|^{-1}$ (e) $f(x) = \left|1 - x^3\right|$

(b) $f(x) = \left|1 - 2\,|x|^3\right|^3$ (d) $f(x) = \left|1 - x^3/2\right|^{-1}$ (f) $f(x) = \left|1 - 2x^3\right|$

Convergence plots:



The y-axis in these plots shows error$(n) = \|f - p_n\|_{[-1,1]}$ where $p_n \in \mathcal{P}_{n-1}$ denotes the n-point Chebyshev interpolant to $f(x)$.

2 Hermite interpolation [5 marks]

Piecewise Hermite interpolation is popular in computer graphics because it is fairly intuitive to work with and leads to functions $p \in P_m\mathcal{P}_n$ which look smooth to the human eye.[1] This assignment illustrates this point by considering the following interpolation problem.

Problem: Given values $f \in \mathbb{R}^4$, determine $p \in \mathcal{P}_3$ such that

$$p(0) = f_1, \qquad p'(0) = f_2, \qquad p(1) = f_3, \qquad p'(1) = f_4.$$

[1] More precisely, it is the Bézier curves which are popular in computer graphics. Bézier curves are closely related to but different from Hermite interpolation, see https://en.wikipedia.org/wiki/Bezier_curve.

Tasks:

1. [2 marks] Determine polynomials $\ell_1, \ell_3 \in \mathcal{P}_3$ such that

$$\ell_1(0) = 1, \quad \ell_1'(0) = \ell_1(1) = \ell_1'(1) = 0 \qquad \text{and} \qquad \ell_3(1) = 1, \quad \ell_3(0) = \ell_3'(0) = \ell_3'(1) = 0.$$

2. [2 marks] Show that the above interpolation problem has a unique solution for any $f \in \mathbb{R}^4$.

   Hint. The polynomials $\ell_2, \ell_4 \in \mathcal{P}_3$ such that

$$\ell_2'(0) = 1, \quad \ell_2(0) = \ell_2(1) = \ell_2'(1) = 0 \qquad \text{and} \qquad \ell_4(1) = 1, \quad \ell_4'(0) = \ell_4'(0) = \ell_4(1) = 0$$

   are given by

$$\ell_2(x) = x\,(1-x)^2 \qquad \text{and} \qquad \ell_4(x) = (1-x)\,x^2.$$

3. [1 mark] Complete the function `hermite_interpolate(f,x)`.

4. [unmarked] Run `draw_heart()`. If your implementation of `hermite_interpolate(f,x)` is correct, then this function will draw a familiar shape. Study the code of `draw_heart()` and see if you can figure out the meaning of the parameters in the matrix `f`.

3 Composite Gauss quadrature [3 marks]

1. [3 marks] Complete the function `composite_gauss(f,a,b,m,n)` such that it approximates $\int_a^b f(x)\,dx$ using composite Gauss quadrature with m intervals and n quadrature points in each interval.

   Hint. You may compute the Gauss quadrature rule $(x_k, w_k)_{k=1}^n$ for the interval $[-1, 1]$ using the function `x,w = gausslegendre(n)` provided by the `FastGaussQuadrature.jl` package. You will then have to map this quadrature rule to $[y_k, y_{k+1}]$ using the integration by substitution formula

$$\int_{\phi(-1)}^{\phi(1)} f(x)\,dx = \int_{-1}^{1} f(\phi(\hat{x}))\,\phi'(\hat{x})\,d\hat{x}.$$

2. [unmarked] Check your answer to Task 1 using `composite_gauss_convergence()`.

4 Equioscillation theorem and Newton's method for computing $1/d$ [6 marks]

This assignment illustrates how we can use polynomial approximation and root-finding to compute a floating-point representation of $\frac{1}{d}$ using only addition and multiplication.

To this end, recall that the floating-point representation of a real numbers $d \in \mathbb{R}$ is given by $d = s \times m \times 2^e$ where $s \in \{\pm 1\}$, $m \in [1, 2)$ and $e \in \mathbb{N}$. We therefore have

$$\tfrac{1}{d} = \big(s \times m \times 2^e\big)^{-1} = s \times m^{-1} \times 2^{-e},$$

and conclude that the only challenge is to compute m^{-1} for $m \in [1, 2)$.

1. [1 mark] Show that Newton's iteration applied to the function $f(x) = mx - 1$ is given by

$$x_{k+1} = \tfrac{1}{m}.$$

While convergent in a single step, this iteration cannot be evaluated without computing $\frac{1}{m}$ by other means and is therefore not useful for our purposes.

2. [1 mark] Show that Newton's iteration applied to the function $f(x) = \frac{1}{x} - m$ is given by

$$x_{k+1} = x_k + x_k\,(1 - m\,x_k).$$

   This iteration requires only addition and multiplication and is therefore a good algorithm for computing $\frac{1}{m}$.

3. [1 mark] Show that the Newton iteration introduced in Task 2 satisfies the recurrence relation

$$x_{k+1} - \tfrac{1}{m} = -m\left(x_k - \tfrac{1}{m}\right)^2. \tag{1}$$

Multiplying both sides of (1) by m, we obtain

$$mx_{k+1} - 1 = -\big(mx_k - 1\big)^2.$$

This shows that the Newton iteration from Task 2 leads to the same reduction in the relative error

$$\frac{x_k - \frac{1}{m}}{\frac{1}{m}} = mx_k - 1$$

for all values of m. It therefore remains to determine a starting value x_0 which makes the initial error as small as possible throught $[1, 2]$. You will do so in the next task using best linear approximation.

4. [2 marks] Determine $p \in \mathcal{P}_1$ such that $e(m) = m\,p(m) - 1$ equioscillates in three points in $[1, 2]$.

   Using a result analogous to the equioscillation theorems presented in Lecture 4, one can show that this p minimises $\|e(m)\|_{[1,2]}$ (you do not have to show this part).

5. [1 mark] Determine the smallest integer K such that for all $m \in [1, 2)$ we have

$$|mx_K - 1| \le 10^{-16} \qquad \text{where} \qquad x_0 = p(m) \quad \text{and} \quad x_{k+1} = x_k + x_k\,(1 - m\,x_k).$$