# AM205: Assignment 0 solutions

For this assignment, solution codes in Python, MATLAB, and C++ are provided. The Python and C++ codes[1] each have identical filenames, although with `.py` and `.cc` suffixes, respectively. The C++ codes output the results to files that can be read by the freeware plotting program Gnuplot. The Python programs make use of Matplotlib. The MATLAB programs[2] are ordered by question number, and make use of MATLAB's built-in graphics.

1. The program `cheby_2d` calculates the Chebyshev polynomials $T_k(x)$ for $k = 0, 1, \ldots, 5$. Figure 1 shows a two-dimensional plot of the function $T_3(x)T_5(y)$ in the region $(x, y) \in [-1, 1]^2$.

2. The program `heron` implements Heron's formula for calculating the square root of five. The first few iterations are

| $k$ | $x_k$ |
|---|---|
| 0 | 5 |
| 1 | 3 |
| 2 | 2.33333333333333 |
| 3 | 2.23809523809524 |
| 4 | 2.23606889564336 |
| 5 | 2.23606797749998 |

   where the digits shown in purple match the exact decimal expansion for $\sqrt{5}$. As is typical for Newton–Raphson iterations, the number of correct digits approximately doubles for each iteration. Using the program, one can determine that four iterations are needed to reduce the absolute error to less than $10^{-3}$ and five iterations are needed to reduce the absolute error to less than $10^{-9}$.

3. (a) The program `finite_diff` calculates the finite-difference approximation of $f(x) = \tan x$ using the formula

$$f_{\text{diff},2}(x; h) = \frac{f(x + h) - f(x - h)}{2h}. \tag{1}$$

   Figure 2 shows the relative error $y$ of $f_{\text{diff},2}(x; h)$ as a function of $h$. The plot is U-shaped, with discretization error dominating for $h > 10^{-8}$ and rounding error dominating for $h < 10^{-8}$. Fitting the data over the range $[10^{-6}, 0.1]$ to

$$\log y = \alpha \log(h) + \beta \tag{2}$$

   gives $\alpha = 2.02$ and $\beta = 1.15$, and hence

$$y \approx 3.17h^{2.02}. \tag{3}$$

[1] Written by Chris H. Rycroft.

[2] Written by Kevin Chen (Teaching Fellow, Fall 2014).

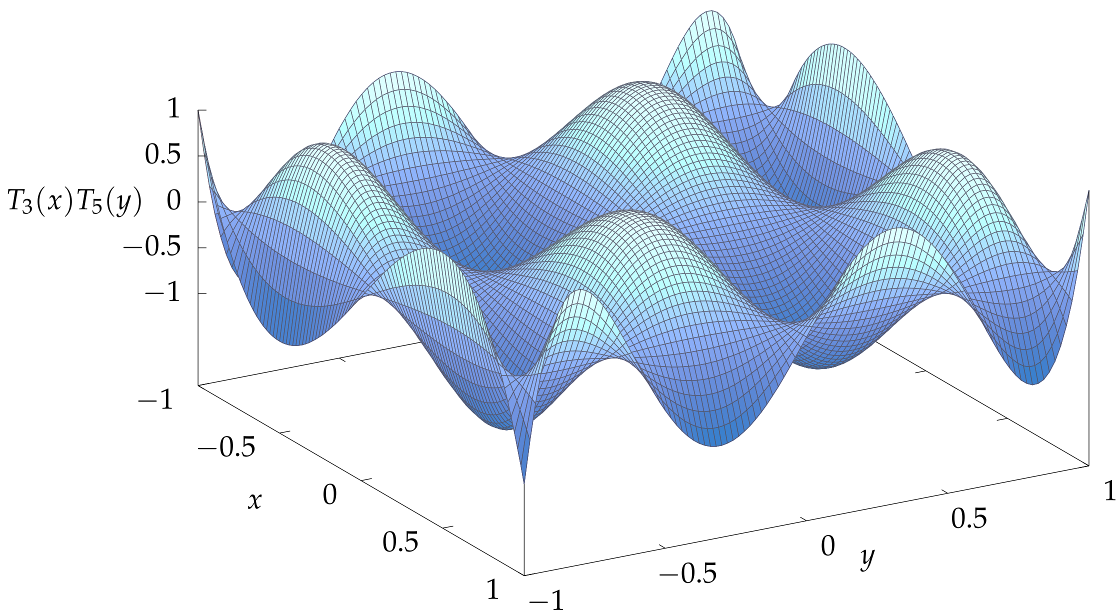


Figure 1: Plot of the product of Chebyshev polynomials $T_3(x)T_5(y)$ considered in question 1.

The relative error therefore scales quadratically with the grid spacing and hence the finite-difference formula in Eq. 1 is second-order accurate.

(b) The program `finite_diff` also calculates the finite-difference approximation of $f(x) = \tan x$ using the formula

$$f_{\text{diff}}(x;h) = \frac{-11f(x) + 18f(x+h) - 9f(x+2h) + 2f(x+3h)}{6h} \tag{4}$$

and the relative errors are plotted on Fig. 2. A similar U-shape is seen, and fitting Eq. 2 in the regime $[10^{-4}, 0.1]$ where discretization error dominates gives $\alpha = 3.15$ and $\beta = 4.50$. Hence

$$y \approx 89.63h^3 \tag{5}$$

and the finite-difference formula is third-order accurate.

4. (a) The inscribed triangle has vertices at $(x, y)$ positions

$$\mathbf{x}_0 = (1, 0), \qquad \mathbf{x}_1 = (-\tfrac{1}{2}, \tfrac{\sqrt{3}}{2}), \qquad \mathbf{x}_2 = (-\tfrac{1}{2}, -\tfrac{\sqrt{3}}{2}).$$

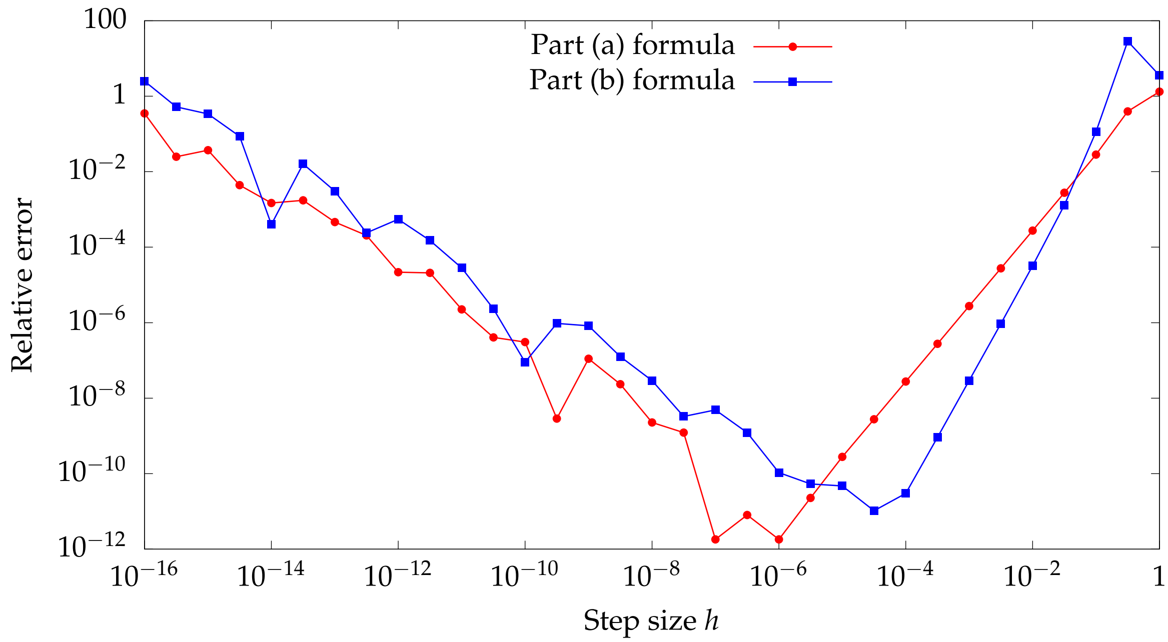


Figure 2: Plot of the relative errors of the two finite-difference approximations considered in question 2.

The triangle's area can be calculated using the three-dimensional vector product,

$$\begin{aligned} a_0 &= \frac{|(\mathbf{x}_1 - \mathbf{x}_0) \times (\mathbf{x}_2 - \mathbf{x}_0)|}{2} = \frac{|(-\frac{3}{2}, \frac{\sqrt{3}}{2}, 0) \times (-\frac{3}{2}, -\frac{\sqrt{3}}{2}, 0)|}{2} \\ &= \frac{|(0, 0, \frac{3\sqrt{3}}{2})|}{2} = \frac{3\sqrt{3}}{4}. \end{aligned} \tag{6}$$

By symmetry, the superscribed triangle's vertices must be an overall scaling of the inscribed triangle's vertices. The correct scaling factor is 2, giving vertices at

$$\mathbf{x}_3 = (2, 0), \qquad \mathbf{x}_4 = (-1, \sqrt{3}), \qquad \mathbf{x}_5 = (-1, -\sqrt{3}),$$

since the edge from $\mathbf{x}_4$ to $\mathbf{x}_5$ will exactly touch the circle at $(-1, 0)$. The area is given by

$$\begin{aligned} b_0 &= \frac{|(\mathbf{x}_4 - \mathbf{x}_3) \times (\mathbf{x}_5 - \mathbf{x}_3)|}{2} = \frac{|(-3, \sqrt{3}, 0) \times (-3, -\sqrt{3}, 0)|}{2} \\ &= \frac{|(0, 0, 6\sqrt{3})|}{2} = 3\sqrt{3}. \end{aligned} \tag{7}$$

(b) Consider a regular inscribed polygon with $k = 3 \times 2^n$ sides. Then it can be broken down into $2k$ right-angled triangles each with area $\frac{1}{2} \cos \frac{\pi}{k} \sin \frac{\pi}{k}$ and hence

$$a_n = k \cos \tfrac{\pi}{k} \sin \tfrac{\pi}{k}. \tag{8}$$

The corresponding superscribed polygon can be broken down into $2k$ right-angled triangles each with area $\frac{1}{2} \tan \frac{\pi}{k}$, and hence

$$b_n = k \tan \tfrac{\pi}{k}. \tag{9}$$

As a check, note that

$$a_0 = 3 \cos \frac{\pi}{3} \sin \frac{\pi}{3} = \frac{3\sqrt{3}}{4}, \tag{10}$$

and

$$b_0 = 3 \tan \tfrac{\pi}{3} = 3\sqrt{3} \tag{11}$$

which agree with Eqs. 6 and 7 from part (a). Then

$$\begin{aligned} \frac{2}{b_{n+1}} &= \frac{1}{k \tan \frac{\pi}{2k}} = \frac{\left(1 + \cos \frac{\pi}{k}\right)}{k \sin \frac{\pi}{k}} = \frac{1}{k \sin \frac{\pi}{k}} + \frac{\cos \frac{\pi}{k}}{k \sin \frac{\pi}{k}} \\ &= \frac{1}{2k \sin \frac{\pi}{2k} \cos \frac{\pi}{2k}} + \frac{1}{b_n} = \frac{1}{a_{n+1}} + \frac{1}{b_n}, \end{aligned} \tag{12}$$

where the half-angle trigonometric identity $\tan \frac{\theta}{2} = (1 + \cos\theta)/ \sin\theta$ and the double angle identity $\sin 2\theta = 2 \sin\theta \cos\theta$ are used. In a similar manner,

$$a_{n+1}^2 = 4k^2 \cos^2 \tfrac{\pi}{2k} \sin^2 \tfrac{\pi}{2k} = k^2 \sin^2 \tfrac{\pi}{k} = k^2 \tan \tfrac{\pi}{k} \sin \tfrac{\pi}{k} \cos \tfrac{\pi}{k} = a_n b_n. \tag{13}$$

The program `archimedes` calculates the values of $a_n$ and $b_n$ up to $n = 40$.

(c) Figure 3 shows a semi-log plot of the absolute errors $E_n^a = |a_n - \pi|$ and $E_n^c = |c_n - \pi|$ as a function of $n$. Fitting the data to

$$\log E_n^a = \alpha^a n + \beta^a, \qquad \log E_n^c = \alpha^c n + \beta^c \tag{14}$$

over the range $n = 1, 2, \ldots, 24$ gives $\alpha^a = -1.386$, $\beta^a = 0.822$, $\alpha^c = -1.385$, $\beta^c = -0.596$ and hence

$$E_n^a \approx 2.275 \times 0.250^n, \qquad E^c \approx 0.551 \times 0.250^n. \tag{15}$$

While $E^c$ is consistently more accurate, the rates of convergence of $E^c$ and $E^a$ are approximately equal.

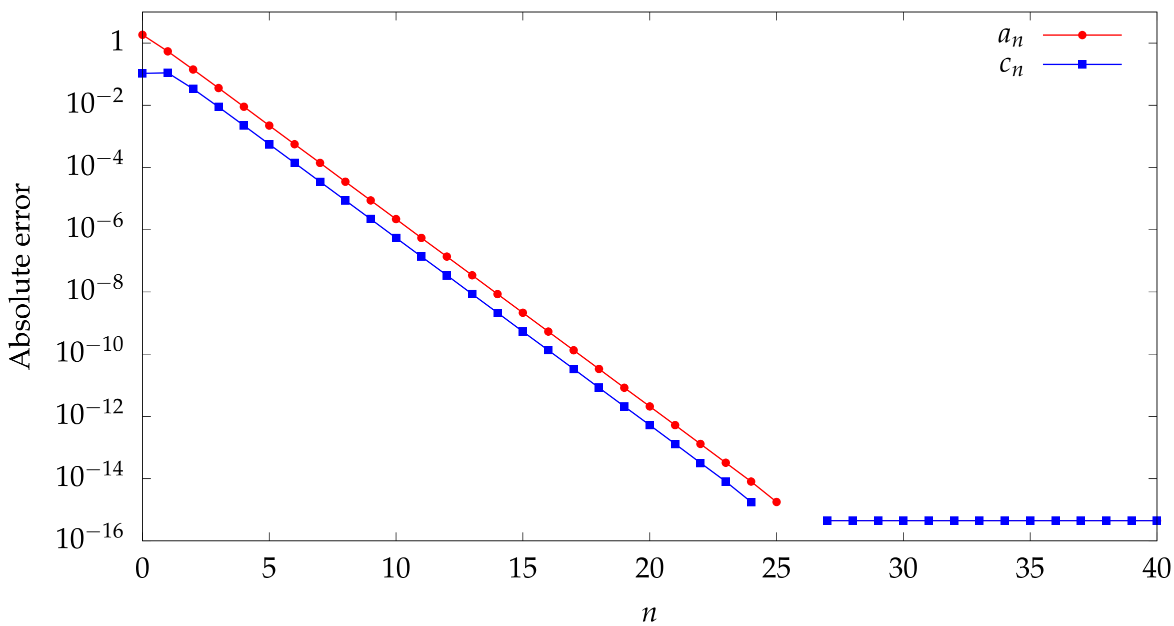


Figure 3: Plot of the absolute errors of the sequences $a_n$ and $c_n$ that converge to $\pi$. Note that there are small breaks in the plots around $n = 25$ due to the absolute errors being identically zero.