INDIAN INSTITUTE OF TECHNOLOGY GUWAHATI

DEPARTMENT OF MATHEMATICS

MA 322: SCIENTIFIC COMPUTING

Semester–II, Academic Year 2022-23

Lab – 5

L5_1. Write a subroutine to compute the integral $I(f) = \int_a^b f(x)\mathrm{d}x$ using a numerical quadrature formula having degree of precision three using $3m+1$ nodes ($m \in \mathbb{N}$). Modify the above algorithm such that the number of nodes will be an user input and appropriate output will be displayed.

L5_2. Implement your best closed Newton-Cotes formula to approximate $I(f) = \int_{-5}^{5} f(x)\mathrm{d}x$ using the following data: with an absolute error $< 10^{-5}$. Compare your result with

x	-5	-4	-3	-2	-1	0	1	2	3	4	5
$f(x)$	0.0385	0.0588	0.1	0.2	0.5	1	0.5	0.2	0.1	0.0588	0.0385

the exact value of the integration 2.7468. Modify your integration formula to integrate $I(f) = \int_{-5}^{4} f(x)\mathrm{d}x$ as well as to integrate $I(f) = \int_{-5}^{6} f(x)\mathrm{d}x$ with $f(6) = 0.0270$.

L5_3. Write a subroutine to compute the integral $I(f) = \int_a^b f(x)\mathrm{d}x$ using a numerical quadrature formula that works for (a) both even and odd number of nodes, (b) only odd number of nodes.

L5_4. Consider a data set $\{(x_i, y_i)\}_{i=0}^{n}$. We want to fit a curve to the given data

$$y = f(x; p_1, p_2, \cdots, p_m)$$

having m parameters. The principle of the least-squares approximation requires the square of the error in the fit

$$R^2 = \sum_{i=0}^{n} [y_i - f(x_i; p_1, p_2, \cdots, p_m)]^2$$

to be minimum. Therefore, we require

$$\frac{\partial R^2}{\partial p_j} = 0, \qquad j = 1, 2, \cdots m.$$

Solving the above system of m equations in m unknowns, we obtained the desired value of $p_j,\ j = 1, 2, \cdots m$ such that the least-squares approximation is obtained.

For a linear least-squares fit

$$f(x; a, b) = a + bx$$

$$R^2 = \sum_i [y_i - (a + bx_i)]^2$$

$$\frac{\partial R^2}{\partial a} = 0 \Rightarrow \sum_i [y_i - (a + bx_i)] = 0 \tag{1}$$

$$\frac{\partial R^2}{\partial b} = 0 \Rightarrow \sum_i [y_i - (a + bx_i)]x_i = 0 \tag{2}$$

Solving the above system (1)–(2), we obtain a and b.

Write a subroutine that will give an output of linear least-squares approximation for a given data set as input.

L5_5. We assume that a table of values has been given: and that a cubic spline S is to be

x	x_0	x_1	$\cdots$	x_n
$f(x)$	y_0	y_1	$\cdots$	y_n

constructed to interpolate the table. On each interval $[x_0, x_1]$, $[x_1, x_2]$, $\ldots$, $[x_{n-1}, x_n]$, S is given by a different cubic polynomial. Let S_i be the cubic polynomial that represent S on $[x_i, x_{i+1}]$. Thus,

$$S(x) = \begin{cases} S_0(x) & x \in [x_0, x_1] \\ S_1(x) & x \in [x_1, x_2] \\ \vdots & \vdots \\ S_{n-1}(x) & x \in [x_{n-1}, x_n] \end{cases}$$

The polynomials S_{i-1} and S_i interpolate the same value at the point x_i and therefore

$$S_{i-1}(x_i) = y_i = S_i(x_i), \qquad 1 \le i \le n-1.$$

Hence, S is automatically continuous. Simple algebra yields that

$$S_i(x) = y_i + (x - x_i)\left[C_i + (x - x_i)[B_i + (x - x_i)A_i]\right],$$

where

$$\begin{aligned} A_i &= \frac{1}{6h_i}(z_{i+1} - z_i) \\ B_i &= \frac{z_i}{2} \\ C_i &= -\frac{h_i}{6}z_{i+1} - \frac{h_i}{3} + \frac{1}{h_i}(y_{i+1} - y_i) \end{aligned}$$

Here, $h_i = x_{i+1} - x_i$ and $z_i = S''(x_i)$. The latter satisfies

$$h_{i-1}z_{i-1} + 2(h_i + h_{i-1})z_i + h_i z_{i+1} = \frac{6}{h_i}(y_{i+1} - y_i) - \frac{6}{h_{i-1}}(y_i - y_{i-1}), \qquad 1 \le i \le n-1.$$

Write a computer program to compute the natural cubic spline that interpolates the table

x_i	0	1	2	3
$f(x_i)$	1	1	0	10

$$f(x) = \begin{cases} 1 + x - x^3, & x \in [0, 1] \\ 1 - 2(x-1) - 3(x-1)^2 + 4(x-1)^3, & x \in [1, 2] \\ 4(x-2) + 9(x-2)^2 - 3(x-2)^3, & x \in [2, 3] \end{cases}$$

Verify whether your result is the same as the function $f(x)$ given above.