Teaching effects of numeric integration by using MATLAB Programming

Khin Ma Ma Than[1], Marlar Win Khin[2]

[1,2]*Lecturer, University of Computer Studies, Banmaw, Myanmar*

ABSTRACT

Teaching mathematical programs parallel to a traditional course is being introduced all over the world and MATLAB is the most popular program for that. Numerical analysis is a useful course which can solve various practical problems and research with the combination of mathematics and computer. In this paper, the demonstrating systems of numerical integration problems are implemented based on MATLAB. The structure of the system and some specific algorithms are given, including Rectangular Rule, Trapezoidal Rule, and Simpson's Rule. The main idea is to show basic principles and solutions to students that MATLAB could provide. They can extend knowledge gained from the MATLAB course on the other courses. This paper aims at teaching how to program the numerical integration with a step-by-step approach in transforming their algorithms to the most basic lines of code that can run on the computer efficiently and output the solution at the required degree of accuracy. In addition, this paper gives notes on the use of each of these MATLAB commands and programs through several Numerical examples.

Keywords— Mathematical Programs, MATLAB, Numerical Analysis, Numerical Integration

1. INTRODUCTION

Computer Algebra Systems (CASs) are software programs with the ability to carry out mathematical computations both numerically and algebraically. CASs are increasingly used in learning mathematics at all levels of the education system. Among the widely available symbolic programs such as MATHEMATICA, MATLAB, and others, MATLAB is one of the easiest programming to learn in mathematical computation. [1]

MATLAB is a powerful package for numerical computation. It contains a rich pool of mathematical functions and provides flexible plotting functions for illustrating mathematical solutions.

This is the best idea to learn the numerical methods of mathematical problems by using computers programming. It is important to get theory skill and knowledge of viewing practical testing. Traditionally, covering the theory is needed to do practical working. So, numerically calculation is used with the calculation program, typical mathematical problems, such as iteration, interpolation, root finding, solving systems of linear equations (matrices), and solving differential equations.

To teach numerical analysis course using MATLAB, it should have two parts, such as theory and laboratory courses. In theory part, the whole course should be covered in such a way so that the course objectives are full-filed and the learning outcomes are achieved. In the laboratory part, there should be some experiments that are to be done using MATLAB. But to overcome the constraints discussed in section II, in the first few laboratory classes students should be taught the basic MATLAB commands and coding. Then using simple MATLAB programs student should solve various problems using different numerical techniques learned in the theory classes. Then they should compare various techniques in terms of execution time, the number of iterations required, the accuracy of the results, etc. [4]

Therefore, in this paper, the teaching method of numerical integration for the students of some numerical programs using MATLAB has been described.

2. RELATED WORKS

Numerical Analysis course is a very important and useful course for students. Therefore, this course has to be taught effectively so that students can apply the knowledge earned from this course in solving their practical problems. On the other hand, MATLAB is a very powerful computing software for scientists. If the students become skilled enough in developing MATLAB codes for different numerical techniques then they will be able to solve many real-life problems numerically. In this way, teaching and learning will be effective. [4]

The focus of [2] is to describe the use of MATLAB as a technological tool for the teaching and learning of mathematics in schools. Application of this software in the teaching of some difficult topics like vectors and matrices, the distance between two points on a coordinate plane, graphs of sine, 3D bar, simultaneous and quadratic equations were demonstrated using the MATLAB. Some recommendations were made on how MATLAB can be applied to enhance the pedagogical process.

MATLAB has been integrated as a supplement to the traditional classroom teaching and learning. The aims of the study are to assess the effects of the use of the software on

students' attitudes and motivation towards the utilization of technology in the teaching and learning of engineering mathematics and its impact on learning. Qualitative and quantitative data collected from the study revealed that the students have benefitted from the software. The use of the software has enhanced students' conceptual understanding despite their weak mathematical skills. It has been noticed that students' attitudes have been positive and their performance in the course has improved. [3]

3. BACKGROUND THEORY

MATLAB, the product of MathWorks Company, is the general-purpose computing software. It contains a vast range of specialized toolboxes and also works as the computer algebra system through its symbolic math toolbox. This toolbox performs symbolic algebraic/mathematical manipulative operations with a lot of built-in interactivity. MATLAB undoubtedly is popular among computer and multi-disciplinary scientists, engineers and particularly with experts in the area of computational mathematics. It integrates numerical, symbolic and state-of-the-art graphics visualization capabilities with quite intuitive computer programming environment. [5]

A teacher using technology to motivate students is more powerful and productive than one simply using lectures and textbooks. The use of MATLAB enhanced students' motivation for learning mathematics. [3]

A number of problems for evaluating definite integrals were demonstrated to students followed by their practice in the computer lab. Following the practice sessions of pattern recognition, students were given the paper-based classroom tests. They found no difficulty in solving the problems taught through the guided discovery using the software. [3]

4. EXPERIMENTAL RESULTS OF NUMERICAL INTEGRATION USING MATLAB

This section shows some numerical examples that will be performed using MATLAB. For the experiment in numerical analysis, we use some varied functions as Rectangular Rule, Trapezoidal Rule, and Simpson's Rule.

4.1 Numeric Integration

Numeric Integration means the numeric evaluation of integrals
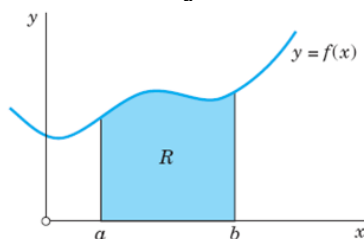
$$J = \int_a^b f(x)dx$$

Fig. 1: Geometric Interpretation of a definite integral

4.1.1 Rectangular Rule: Numeric integration methods are obtained by approximating the integrand f by functions that can easily be integrated. The rectangular rule is the simplest formula to compute an approximation of a definite integral. Divide the interval [a, b] into n pieces; each piece has the same width $h = \frac{b-a}{n}$. Then the n rectangles in Figure 2 have the areas $f(x_1^*)h, \dots f(x_n^*)h$ and the rectangular rule is
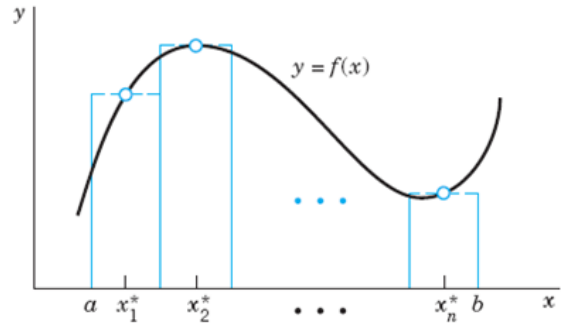
Fig. 2: Rectangular Rule

$$J = \int_a^b f(x)\,dx \approx h[f(x_1^*) + f(x_2^*) + \cdots + f(x_n^*)] \quad \left(h = \frac{b-a}{n}\right).$$

(a) MATLAB Function of Rectangular Rule

Example: Rectangular Rule

Evaluate $J = \int_0^1 e^{-x^2}\,dx$ by Rectangular Rule with n=10.

$$h = \frac{b-a}{n} = \frac{1-0}{10} = 0.1$$

$$J = \int_a^b f(x)\,dx \approx h[f(x_1^*) + f(x_2^*) + \cdots + f(x_n^*)]$$

i	x^*	f(x^*)
1	0.05	0.997503
2	0.15	0.977751
3	0.25	0.939413
4	0.35	0.884706
5	0.45	0.816686
6	0.55	0.738968
7	0.65	0.655406
8	0.75	0.569783
9	0.85	0.485537
10	0.95	0.405555

$$J = \int_0^1 e^{-x^2}\,dx \approx 0.1(0.997503 + 0.977751 + \cdots + 0.405555)$$

$$= 0.747131.$$

When we run the MATLAB Function of Rectangular Rule, the output is

```
>> f=@(x) exp(-x^2)

f =

    @(x) exp(-x^2)

>> a=0;b=1;n=10;
>> rectangular(f, a, b, n)
i       mid-x       f(mid-x)
1       0.05        0.997503

2       0.15        0.977751

3       0.25        0.939413

4       0.35        0.884706

5       0.45        0.816686

6       0.55        0.738968

7       0.65        0.655406

8       0.75        0.569783

9       0.85        0.485537

10      0.95        0.405555

Area= 0.747131
```

4.1.2 Trapezoidal Rule: The Trapezoidal Rule is generally more accurate. The trapezoidal rule works by approximating the region under the graph of a function as a trapezoid and calculating its area. Then the area under the curve f between a and b is approximated by n trapezoids of areas.

$$\frac{1}{2}[f(a) + f(x_1)]h, \frac{1}{2}[f(x_1)f(x_2)]h\,..., \qquad \frac{1}{2}[f(x_{n-1}) + f(b)]h$$

Fig. 3: Trapezoidal Rule

$$J = \int_a^b f(x)\,dx \approx h\left[\frac{1}{2}f(a) + f(x_1) + f(x_2) + \cdots + f(x_{n-1}) + \frac{1}{2}f(b)\right]$$

(a) MATLAB Function of Trapezoidal Rule

```
Editor - C:\Users\CUBMW\Marlarwinkhin\matlabprogram...
trapezoidal.m  +
1    function trapezoidal(f,a, b, n)
2    h = (b-a)/n
3    sum = 0;
4    VarNames = {'i', 'x(i)', 'f(xi)'};
5    fprintf( ' %s\t%s\t\t%s\t\t%s\n', VarNames{:})
6    for i = 1:n-1
7        x = a + i*h;
8        sum = sum + f(x);
9        fprintf('\n%d\t%4.2f\t\t%8.6f\n', i,x,f(x))
10   end
11   fprintf('Sum=%9.6f\n',sum)
12   fprintf('f(a)=%9.6f\n',f(a))
13   fprintf('f(b)=%9.6f\n',f(b))
14   Area = h*(f(a)/2 + sum + f(b)/2);
15   fprintf('Area = h*(f(a)/2 + sum + f(b)/2)=%9.6f\n',Area)
```

Example: Trapezoidal Rule

Evaluate $J = \int_0^1 e^{-x^2}\,dx$ by Trapezoidal Rule with n=10.

$$h = \frac{b - a}{n} = \frac{1 - 0}{10} = 0.1$$

j	x_j	x_j^2	$e^{-x_j^2}$	
0	0	0	1.000000	
1	0.1	0.01		0.990050
2	0.2	0.04		0.960789
3	0.3	0.09		0.913931
4	0.4	0.16		0.852144
5	0.5	0.25		0.7788001
6	0.6	0.36		0.697676
7	0.7	0.49		0.612626
8	0.8	0.64		0.527292
9	0.9	0.81		0.444858
10	1.0	1.00	0.367879	
Sums				6.778167

$$J = \int_0^1 e^{-x^2}\,dx \approx 0.1(0.5 * 1.367879 + 6.778167)$$
$$= 0.746211$$

When we run the MATLAB Function of Trapezoidal Rule, the output is

```
>> trapezoidal(f, a, b, n)

h =

    0.1000

i       x(i)        f(xi)
1       0.10        0.990050

2       0.20        0.960789

3       0.30        0.913931

4       0.40        0.852144

5       0.50        0.778801

6       0.60        0.697676

7       0.70        0.612626

8       0.80        0.527292

9       0.90        0.444858
Sum= 6.778168
f(a)= 1.000000
f(b)= 0.367879
Area = h*(f(a)/2 + sum + f(b)/2)= 0.746211
```

4.3.3 Simpson's Rule: Piecewise quadratic approximation will lead to Simpson's rule, which is of great practical importance because it is sufficiently accurate for most problems, but still sufficiently simple.

To derive Simpson's rule, we divide the interval of integration $a \le x \le b$ into an **even number** of equal subintervals, say, into $n = 2m$ subintervals of length $h = (b - a)/2m$, with endpoints $x_0(= a), x_1, ..., x_{2m-1}, x_{2m}(= b)$. We now take the first two subintervals and approximate f(x) in the interval

$$x_0 \le x \le x_2 = x_0 + 2h$$

by the Lagrange polynomial $p_2(x)$ through $(x_0, f_0), (x_1, f_1), (x_2, f_2)$, where $f_j = f(x_j)$

$$p_2(x) = \frac{(x - x_1)(x - x_2)}{(x_0 - x_1)(x_0 - x_2)} f_0 + \frac{(x - x_0)(x - x_2)}{(x_1 - x_0)(x_1 - x_2)} f_1 + \frac{(x - x_0)(x - x_1)}{(x_2 - x_0)(x_2 - x_1)} f_2.$$

The denominators of the above equations are $2h^2, -h^2$ and $2h^2$, respectively. Setting

$$s = \frac{x - x_1}{h} \text{ , we have}$$

$$x - x_1 = sh, x - x_0 = x - (x_1 - h) = (s + 1)h$$
$$x - x_2 = x - (x_1 \mp h) = (s - 1)h$$

And we obtain

$$p_2(x) = \frac{1}{2}s(s - 1)f_0 - (s + 1)(s - 1)f_1 + \frac{1}{2}(s + 1)sf_2$$

$$\int_{x_0}^{x_2} f(x)dx \approx \int_{x_0}^{x_2} p_2(x)dx = h(\frac{1}{3}f_0 + \frac{4}{3}f_1 + \frac{1}{3}f_2)$$
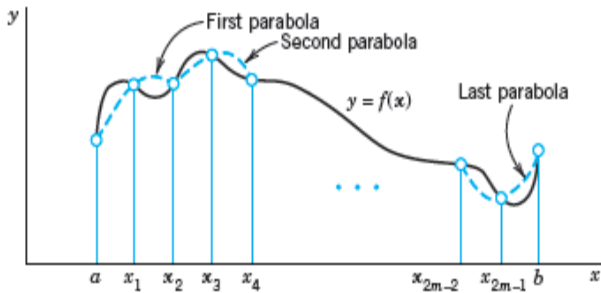
Fig. 4: Simpson's Rule

A similar formula holds for next two subintervals from x_2 to x_4, and so on. By summing all these m formulas we obtain Simpson's rule

$$\int_a^b f(x)\,dx \approx \frac{h}{3}\,(f_0 + 4f_1 + 2f_2 + 4f_3 + \cdots + 2f_{2m-2} + 4f_{2m-1} + f_{2m}),$$

Where $h = \frac{b-a}{2m}$ and $f_j = f(x_j)$.

Table 1: Simpson's Rule of Integration

ALGORITHM SIMPSON $(a, b, f_0, f_1, \dots f_{2m})$
This algorithm computes the integral $J = \int_a^b f(x)dx$ from given values f at equidistant $x_0 = a, x_1 = x_0 + h, \dots, x_{2m} = x_0 + 2mh = b$ by Simpson's rule.

Set $h = \frac{b-a}{2m}$

INPUT : a, b, m, f_0, …, f_{2m}
OUTPUT: Approximate value of $\widetilde{J}$ of J
Compute $s_0 = f_0 + f_{2m}$
$\qquad\quad s_1 = f_1 + f_3 + \cdots + f_{2m-1}$
$\qquad\quad s_2 = f_2 + f_4 + \cdots + f_{2m-2}$
$\qquad\quad \widetilde{J} = \frac{h}{3}(s_0 + 4s_1 + 2s_2)$

OUTPUT $\widetilde{J}$. Stop
End SIMPSON

(a) MATLAB Function of Simpson's Rule of Integration

```matlab
function simpson(f,a, b, n)
    h = (b-a)/n
    s0=0;s1=0;s2=0;
    VarNames = {'i', 'x(i)', 'f(xi)'};
    fprintf('\t%s\t\t%s\t\t%s\n', VarNames{:})
    s0=f(a)+f(b);
    for i=1:n-1
        x = a+i*h;
        fprintf(1,'\n\t%d\t\t%4.2f\t\t%8.6f\n', i,x,f(x));
        if mod(i,2)~=0
            s1=s1+ f(x);
        else
            s2=s2+ f(x);
        end
    end
    Area = h*(s0+ 4*s1 + 2*s2 )/3;
    fprintf('\nArea =h*(s0+ 4*s1 + 2*s2 )/3=%9.6f\n',Area)
```

When we run the MATLAB Function of Simpson's Rule, the output is

```
>> f=@(x)exp(-x^2)

f =

    @(x)exp(-x^2)

>> a=0;b=1;n=10;
>> simpson(f, a, b, n)

h =

    0.1000

    i       x(i)        f(xi)
    1       0.10        0.990050

    2       0.20        0.960789

    3       0.30        0.913931

    4       0.40        0.852144

    5       0.50        0.778801

    6       0.60        0.697676

    7       0.70        0.612626

    8       0.80        0.527292

    9       0.90        0.444858

Area =h*(s0+ 4*s1 + 2*s2 )/3= 0.746825
```

Exact value of $\int_0^1 e^{-x^2} dx$ is 0.746825. Thus our result from Simpson's rule is much better than the trapezoidal rule, whereas the number of operations is nearly the same in both cases.

5. CONCLUSIONS

In this paper, the MATLAB programming language used in teaching the numerical integration problems to the undergraduate students is discussed. Students' understanding will get widened and their ability to numerical integration will be enhanced. The skill of using computer technology will be sharpened. How MATLAB can be used in numerical integration to enhance the teaching of Rectangular Rule, Trapezoidal Rule, and Simpson's Rule are illustrated. Therefore, numerical integration course should be taught in such a way so that the students can be able to develop their problem-solving skills on numerical integration to solve verification problems using MATLAB.

6. REFERENCES

[1] A. Nyamapfene, "Integrating MATLAB into First-Year Engineering Mathematics ". 2016 IEEE International Conference on Teaching and Learning in Education (ICTLE'16), Volume: 2016.

[2] C. Ogan, G. Ibibo (Ph.D.), "Utilization of MATLAB as a Technological Tool for Teaching and Learning of Mathematics in Schools". International Journal of Mathematics and Statistics Studies Vol.3, No.5, pp.10-24, November 2015.

[3] M. Abdul Majid, Z. A. Huneiti, M. A. Al-Naafa, W. Balachandran, "A Study of the Effects of Using MATLAB as a Pedagogical Tool for Engineering Mathematics Students". iJOE- Volume 9, Issue 2, May 2013, pp-27-35.

[4] M. Haque Bhuyan, S. Shermin Azmiri Khan, "Teaching Numerical Analysis Course for Electrical Engineering Students using MATLAB". SEU Journal of Science and Engineering, Vol. 10, No. 2, December 2016, pp-38-46.

[5] M. Abdul Majid, Z. A. Huneiti, W. Balachandran, "Matlab As A Teaching and Learning Tool For Mathematics: A Literature Review". International Journal of Arts & Sciences, CD-ROM. ISSN: 1944-6934: 6(3):23–44 (2013).

[6] S. I. Barry, T. Webb, "Multi-disciplinary approach to teaching numerical methods to engineers using MATLAB". ANZIAM J. 47 (EMAC2005) pp-C216-C230, 2006.

BIOGRAPHY

Khin Ma Ma Than
Lecturer
Faculty of Computing
University of Computer Studies, Banmaw, Myanmar

Marlar Win Khin
Lecturer
Faculty of Computing
University of Computer Studies, Banmaw, Myanmar