Transactions on Comparative Education (2025)
Clausius Scientific Press, Canada

DOI: 10.23977/trance.2025.070210
ISSN 2523-5818 Vol. 7 Num. 2

Research on Calculation and Comparison of Multiple Integrals Based on MATLAB

Zhao Ye*, Zhang Bowei, Zhang Jiaqi, Li Huilong

Zhiyuan College, Beijing Institute of Petrochemical Technology, Beijing, China
Corresponding author: zye@bipt.edu.cn

Keywords: MATLAB, Simulation, Double Integrals, Multiple Integrals

Abstract: Based on MATLAB software, this paper discusses double integrals over general regions, and presents corresponding computational methods. Finally, numerical simulations are conducted. The effectiveness and feasibility of the proposed methods are demonstrated through comparative examples. Through practical implementation, students are cultivated to develop an awareness of utilizing MATLAB software for calculating double integrals, thereby enhancing their efficiency in computing double integrals and fostering their interest in this area of study.

1. Introduction

Integral calculus of multivariate functions constitutes a pivotal part of advanced mathematics, finding extensive and impactful applications in practical computations. A multitude of real-world challenges can be reformulated as problems involving the integration of multivariate functions. Double integrals act as a crucial intermediary between definite and triple integrals, and they also lay the groundwork for computing surface integrals. In practical scenarios, double integrals can be harnessed to determine the area of geometric figures, the centroid of planar laminas, the mass of planar laminas, among other quantities. Given the intricacy and foundational importance of double integrals, many students frequently find themselves perplexed during their study, grappling with the effective analysis of integration regions and the efficient transformation of double integrals into iterated integrals. In the process of solving double integrals, harnessing the powerful functionalities offered by large-scale computational software like MATLAB as auxiliary tools carries significant practical value[1-3].

MATLAB, as a highly adaptable mathematical software, excels in both robust numerical computation capabilities and symbolic operation proficiencies[4,5]. It is distinguished by its low barrier to entry, concise input syntax, high computational efficiency, strong scalability, and rich feature set, playing an indispensable role in university mathematics instruction and students' learning endeavors. When compared to similar software such as Mathematica and Maple, MATLAB emerges as more user-friendly. Although MATLAB offers methods for calculating double integrals, encompassing both symbolic and numerical approaches, the symbolic method often involves the sequential computation of two single integrals using MATLAB's built-in 'int' command, frequently yielding symbolic outcomes. To derive numerical values, one must resort to the 'vpa' command for computation, or in more recent versions, 'quad2d' can be employed to compute double integral values. However, for slightly more convoluted double integrals, these two commands may prove inadequate in computing the integral values. The numerical method utilizes the 'dblquad' function, but it necessitates that both the inner and outer integration limits be constant functions, implying that it can only compute double integrals over rectangular domains. Despite the effectiveness of the aforementioned methods in addressing fundamental computational problems such as definite and indefinite integrals, they fall short of delivering complete satisfaction when it comes to computing multiple integrals. Other challenges associated with multiple integral calculations, such as altering the order of integration, computing integrals with parameters, and selecting suitable integration regions, cannot be resolved solely through direct function commands. Furthermore, MATLAB appears to be ill-equipped when confronted with more intricate integrals, such as those that are "non-integrable" in terms of elementary functions.

This paper initially introduces MATLAB's built-in functions and subsequently modifies pertinent commands to propose several computational methodologies and corresponding MATLAB commands for calculating double integrals over general regions. The efficacy of the proposed methods is substantiated through comparative examples.

2. Symbolic solution method for double integrals

Calculating double integrals using MATLAB can be understood as computing two single integrals sequentially. Therefore, the calculation of double integrals can be achieved by iterating through the process of single integrals.

2.1. Computing two single integrals sequentially

Example 1: Compute $\int_{10}^{20}\int_{5x}^{x^2} e^{\sin x} \ln y dx dy$

Using MATLAB with the following code:

```
syms x y
f=exp(sin(x))*log(y);
ff=int(f,y,5*x,x^2);
fff=int(ff,x,10,20);
vpa(fff)
```

or

```
syms x y
vpa(int(int(exp(sin(x))*log(y),y,5*x,x^2),x,10,20))
```

The result is

```
ans =
   9368.6713426796924423355001787843
```

Remark:

1) The int command is used to compute a single integral, but its result is often in symbolic form;

2) To calculate the numerical value of the integral, one must use the vpa function to obtain a numerical approximation. The command format for invocation is as follows:

```
vpa(fun,n)
```

Which indicates that the result is rounded to n significant digits;

3) The following command can be utilized to compute the value of a double integral:

```
quad2d(@(x,y) exp(sin(x)).*log(y),10,20,@(x)5*x,@(x)x.^2
```

4) For integrals that cannot be evaluated in closed form, numerical algorithms are necessary.

2.2. Using 'dblquad' command

Example 2: Compute $\int_{-1}^{1}\int_{-2}^{2} e^{\frac{x^2}{2}} \sin(x^2+y)dxdy$

Using MATLAB with the following code:

```
f1=@(x,y)exp(-x.^2/2).*sin(x.^2+y)
ff1=dblquad(f1,-2,2,-1,1)
```

The result is

```
ans =
    1.5745
```

Remark:

1) The dblquad function is invoked with the syntax dblquad(fun, xmin, xmax, ymin, ymax);
2) `dblquad` command is restricted to computing double integrals over rectangular domains;
3) The integrand must be defined as a function handle.

3. Numerical techniques for evaluating double integrals

Symbolic integration methods may sometimes fail to yield a solution. In such cases, numerical methods can be employed to compute the value of the integral. When calculating definite integrals numerically, common approaches include composite quadrature rules, Romberg's method, and Gaussian quadrature. However, MATLAB does not provide built-in functions specifically tailored for numerical integration over general regions. This section adopts the approach used for numerical integration of univariate functions (such as the trapezoidal rule) to solve the problem.

For a double integral $I=\iint_D f(x,y)dxdy$, we first transform it into an iterated integral $I=\int_a^b\int_{c(x)}^{d(x)} f(x,y)dxdy$. Using the trapezoidal rule, we partition the integration interval $[a,b]$ into m equal subintervals. Let $h_x=\frac{(b-a)}{m}$ and $x_i=a+ih_x$, and $I\approx h_x\left(\frac{G(a)-G(b)}{2}+\sum_{i=1}^{m-1}G(x_i)\right)$, where $G(x_i)=\int_{c(x_i)}^{d(x_i)} f(x,y)dy$. For each fixed *x*, we further divide the integration interval for the inner variable $[c(x_i),d(x_i)]$ into *n* equal subintervals, where $h_y(i)=\frac{c(x_i)-d(x_i)}{n}$, $y_{ij}=c(x_i)+jh_y(i)$. We represent the integration interval for *y* corresponding to *x*. Thus, the double integral can then be approximated using the following approach

$$G(x_i)\approx h_y(i)\left\{\frac{1}{2}\left[f(x_i,c(x_i))+f(x_i,d(x_i))\right]+\sum_{j=1}^{n-1} f(x_i,y_{ij})\right\}.$$

MATLAB code as follows:

```
function I=Idblquad(fun,a,b,cfun,dfun,m,n)
if nargin<7
    n=100;
end
if nargin<6
    m=100;
end
```

```
if m<2|n<3
    error('Number if intervals invalid \n');
end
mpt=m+1;
hx=(b-a)/m;
x=a+(0:m).*hx;
for i=1:mpt
    ylo=feval(cfun,x(i));
    yhi=feval(dfun,x(i));
    hy=(yhi-ylo)/n;
    y(i,:)=ylo+(0:n).*hy;
    f(i,:)=feval(fun,x(i),y(i,:));
    G(i)=trapz(y(i,:),f(i,:));
end
I=trapz(x,G);
```

Example 3: Compute $\int_1^2 \int_{\sin x}^{\cos x} xydxdy$

Using MATLAB with the following code:

```
function Intdemo
function f1=myfun1(x)
f1=zeros(size(x));
for k=1:length(x)
    f1(k)=quadl(@(y) x(k)*y,sin(x(k)),cos(x(k)));
end
quadl(@myfun1,1,2)
```

The result is

```
ans =
   -0.6354
```

Remark:

1) The myfun1 function is designed to construct the antiderivative (indefinite integral) of a given function. In this context, it is a function with respect to a certain variable;

2) The template command for calculating a double integral can be written as

quadl(@(x) arrayfun(@(xx) quadl(@(y)

3) The integrand must be defined as a function handle.

4. Comparison of three methods

The following example compares three methods for calculating double integrals in terms of computation time to illustrate the effectiveness of each method.

Example 4: Compute $\int_{-1}^{1} \int_{-2}^{2} e^{\frac{x^2}{2}} \sin(x^2 + y)dxdy$

The results are demonstrated in Table 1.

Table 1 Comparison of Computational Results

Method	Result	Running time (s)
int	1.5263	0.359000
dblquad	1.5745	0.203000
Intdemo	1.5263	0.015000

Each of the three methods has its own scope of application and is not a universal MATLAB method for calculating double integrals. Therefore, it is crucial to pay attention to their practical conditions when utilizing them.

As can be seen from Table 1, the computational results of various methods are quite close to each other, but there are significant differences in their running times. The method developed in this project for calculating double integrals over general regions has the shortest computation time.

5. Conclusion

Double integrals constitute a fundamental yet challenging topic in advanced mathematics, posing significant difficulties for many students during the learning process. The complexity of solving double integrals frequently diminishes students' enthusiasm for advanced mathematics. This paper presents a comprehensive analysis of MATLAB's application in double integral computation, a prominent large-scale commercial computing software. We first introduce the fundamental concept of double integrals, followed by detailed examples demonstrating their solution using MATLAB. Furthermore, we systematically summarize a general procedural framework for computing double integrals with MATLAB assistance. As a powerful computational tool, MATLAB plays a crucial role in facilitating double integral calculations. It is strongly recommended to actively incorporate MATLAB training in mathematics teaching, particularly in double integral instruction, to significantly improve students' computational efficiency and, more importantly, to revitalize their interest and engagement in advanced mathematics learning. The integration of computational software like MATLAB not only enhances problem-solving capabilities but also bridges the gap between theoretical concepts and practical applications in mathematical teaching.

Acknowledgements

This work is supported by Undergraduate Research and Training Program of Beijing Institute of Petrochemical Technology (NO. 2025J00200) and Beijing Association of Higher Education (NO. MS2024210).

References

[1] Xu Weiwei. (2017) Application of MATLAB in Double Integral Calculation. Journal of Tianjin Vocational Institutes. 19(2):115-120.

[2] Yang Huaijun, Zhou Yongwei, Liu Lin. (2023) Analysis of Triple Integral Volume Calculus in Spherical Coordinate System. College Mathematics, 39(5):108-112.

[3] He Pu. (2025) Application of Mathematical Software in the Teaching of Advanced Mathematics Courses. Journal of Liaoning Open University, 3(1):60-65.

[4] Shi Jiarong, Yong Longquan. (2025) Visualization of Fourier Series Based on MATLAB. Studies in College Mathematics, 28(1):75-78.

[5] Yang Mengyun. (2024) Studies of Space Analytical Geometry Teaching Based on MATLAB. Science & Technology Information, 15: 211-214.