International Journal of Latest Engineering and Management Research (IJLEMR)
ISSN: 2455-4847
www.ijlemr.com || Volume 07 - Issue 05 || May 2022 || PP. 38-42

Implementing the Multiple Integral on General Domain with an Adjusted Step-size Approach

Dinh Van Tiep[1], Hoang Thanh Nga[2]
[1](Thai Nguyen University of Technology, TNU, Vietnam)
[2](Thai Nguyen University of Technology, TNU, Vietnam)

Abstract: The paper aims to present the construction of an implementation to an adjusted step-size quadrature method to approximate the multiple integral on a quite general domain. The implementation is written in the Matlab program so that it could be easy for one who work in the engineering field to refer to it. In this approach, a technique of discretizing the boundary of the domain of integral is used to finitely code the domain. So that more data on the domain we need for making use of more accuracy we need on the resulting approximation. This way will reduce the computational cost and is preferable to apply in practical application.
Keywords: Ajusted step-size, implementation, Matlab code, multiple integral, quadrature method.

I. INTRODUCTION

For the quadrature technique in multiple integral, the number of functions evaluations increase largely in order to reach some require level of accuracy. Hence, the adjusted step-size approach is often preferred choice because of a high efficiency and a low cost of calculation ([1-2]). Sincethe continuity of the data for the integrated regions and of values for the integrand, the discretization of data for both of the region and the function values are used by constructing the net of the potential mesh points. It makes the storing data on computer processor becomes solvable. This technique is highly approved since the result we want is to extract only an approximation within a requisite accuracy.

II. MULTIPLE INTEGRALS ON NON-RECTANGULAR DOMAINS

Consider the problem of approximating multiple integral I of a function f, defining and continuously differentiable up to order 4 on a region $B \subset \mathbb{R}^{n+1}$, where B is a non-rectangular hyper-box and

$$B = \{(x_1, x_2, \dots, x_{n+1}) = (\mathbf{x}, x_{n+1}) \in \mathbb{R}^n \times \mathbb{R} \mid F(\mathbf{x}) \le x_{n+1} \le G(\mathbf{x}), \mathbf{x} \in \mathrm{D} \subset \mathbb{R}^n\},$$
$$D = [a_1, b_1] \times [a_2, b_2] \times \dots \times [a_n, b_n].$$

Here, F, G are both continuous on the rectangular hyper-box $D \subset \mathbb{R}^n$. We have,

$$I = \int_B f(\mathbf{x}, x_{n+1})\, d\mathbf{y} = \int_D d\mathbf{x} \int_{F(\mathbf{x})}^{G(\mathbf{x})} f(\mathbf{x}, x_{n+1}) dx_{n+1}\,, \text{with } \mathbf{y} = (\mathbf{x}, x_{n+1}). \quad \text{(Error! Bookmark not defined.)}$$

Our aim now is to formulate an approximation for *I* by using Simpson's rule and Composite Simpson's rule with $n = 4$. By Simpson's rule for one variable integral, the inner integral of (1) is treated as

$$\int_{F(\mathbf{x})}^{G(\mathbf{x})} f(\mathbf{x}, x_{n+1}) dx_{n+1} = \frac{k(\mathbf{x})}{3}\left[n_1 f\left(\mathbf{x}, x_{n+1}^{\mathbf{x},1}\right) + n_2 f\left(\mathbf{x}, x_{n+1}^{\mathbf{x},2}\right) + n_3 f\left(\mathbf{x}, x_{n+1}^{\mathbf{x},3}\right)\right] - E_0, \quad \text{(Error! Bookmark not defined.)}$$

$E_0 = \frac{k^5(\hat{\mathbf{x}})}{90}\frac{\partial^4 f}{\partial x_{n+1}^4}\left(\hat{\mathbf{x}}, x_{n+1}^{\hat{\mathbf{x}}}\right)$, and $n_1 = n_3 = 1, n_2 = 4, k(\mathbf{x}) = \frac{G(\mathbf{x}) - F(\mathbf{x})}{2}$, with $F(\mathbf{x}) = x_{n+1}^{\mathbf{x},1} < x_{n+1}^{\mathbf{x},2} = x_{n+1}^{\mathbf{x},1} + k(\mathbf{x}) < x_{n+1}^{\mathbf{x},3} = G(\mathbf{x}), \hat{\mathbf{x}} \in D$, where $x_{n+1}^{\mathbf{x},}$ depends on the fixed point $\mathbf{x} \in D$.

We now develop from the single variable case two results which are used in our approach ([3-4]).

Theorem 1 (Simpson's rule for Multiple Integral)

$$I = \frac{h_1 h_2 \dots h_n}{3^{n+1}} \sum_{\substack{1 \le i_1, i_2, \dots, i_{n+1} \le 3 \\ 1 \le j_1, j_2, \dots, j_{n+1} \le 3}} \left(\prod_{q=1}^{n+1} n_{j_q}\right) k\left(x_1^{i_1}, \dots, x_n^{i_n}\right) f\left(x_1^{i_1}, x_2^{i_2}, \dots, x_{n+1}^{\mathbf{x}, i_{n+1}}\right) + E_1,$$

(Error! Bookmark not defined.)

where $h_i = \frac{b_i - a_i}{2}, x_i^{i_q} = a_i + (i_q - 1)h_i (\forall i, q = 1,2, \dots, n), \mathbf{x} = (x_1^{i_1}, \dots, x_n^{i_n})$,

$$E_1 = \frac{Vol(D)}{90}\left[k^5(\bar{\mathbf{x}})\frac{\partial^4 f}{\partial x_{n+1}^4}(\bar{\mathbf{x}}, x_{n+1}^{\bar{\mathbf{x}}}) + \sum_{i=1}^{n} h_i^4 \frac{\partial^4}{\partial x_i^4}\left(k(\hat{\mathbf{x}}^{(i)}) f\left(\hat{\mathbf{x}}^{(i)}, x_{n+1}^{\hat{\mathbf{x}}^{(i)}}\right)\right)\right],$$

$Vol(D) = (b_1 - a_1)(b_2 - a_2) \dots (b_n - a_n)$, for some $\bar{\mathbf{x}}, \hat{\mathbf{x}}^{(i)} \in D$, and some $x_{n+1}^{\hat{\mathbf{x}}^{(i)}} \in \left(F(\hat{\mathbf{x}}^{(i)}), G(\hat{\mathbf{x}}^{(i)})\right)$.

PROOF: Integrate both sides of (2) over D with respect to **x**with the use of Mean Value Theorem for the multiple integral of the term E_0 over D to get

$$I = \frac{1}{3}\int_D \sum_{j=1}^{3} n_j k(\mathbf{x}) f\left(\mathbf{x}, x_{n+1}^{\mathbf{x},j}\right) d\mathbf{x} + \frac{Vol(D)}{90} k^5(\bar{\mathbf{x}}) \frac{\partial^4 f}{\partial x_{n+1}^4}(\bar{\mathbf{x}}, x_{n+1}^{\bar{\mathbf{x}}}). \quad (4)$$

Reapply Simpson's rule for one variable on $[a_n, b_n]$ for terms in the sum of the first term of (4), denoting that $D = D_{n-1} \times [a_n, b_n]$, to get the first term of (4)

$$\frac{h_n}{3^2}\int_{D_{n-1}} \sum_{1 \le j_n, j_{n+1}, i_n \le 3} n_{j_n} n_{j_{n+1}} k(\mathbf{x}', x_n^{i_n}) f(\mathbf{x}', x_n^{i_n})\, d\mathbf{x}' + \int_{D_{n-1}} Er_n(\mathbf{x}') d\mathbf{x}', \quad (5)$$

where $\mathbf{x}' = (x_{i_1}, \dots, x_{i_{n-1}})$, and for some $\tilde{x}_n \in (a_n, b_n)$,

$$Er_n(\mathbf{x}') = \frac{1}{3}\frac{h_n^5}{90}\sum_{j_{n+1}=1}^{3} n_{j_{n+1}} \frac{\partial^4}{\partial x_n^4}\left(k(\mathbf{x}', \tilde{x}_n) f\left(\mathbf{x}', \tilde{x}_n, x_{n+1}^{(\mathbf{x}', \tilde{x}_n), j_{n+1}}\right)\right). \quad (6)$$

Applying Intermediate Value Theorem for (5), and then Mean Value Theorem for the second term of (4), we obtain

$$\int_{D_{n-1}} Er_n(\mathbf{x}') d\mathbf{x}' = \frac{Vol(D) h_n^4}{90}\frac{\partial^4}{\partial x_n^4}\left(k(\hat{\mathbf{x}}^{(n)}) f\left(\hat{\mathbf{x}}^{(n)}, x_{n+1}^{\hat{\mathbf{x}}^{(n)}}\right)\right), \text{for some} \hat{\mathbf{x}}^{(n)} \in D.$$

Similarly, reapplying Simpson's rule consecutively for other $(n-1)$ dimensions of D, we complete the proof. □
Similar to Simpson's rule, we could derive the following rule.

Theorem 2 (Composite Simpson's rule for Multiple Integral) Let $m_1 = m_5 = 1, m_2 = m_4 = 4, m_3 = 2$ to be coefficients of Composite Simpson's rule with $n = 4$. We have,

$$I = \frac{p_1 p_2 \dots p_n}{3^{n+1}} \sum_{\substack{1 \le i_1, i_2, \dots, i_{n+1} \le 5 \\ 1 \le j_1, j_2, \dots, j_{n+1} \le 5}} \left(\prod_{s=1}^{n+1} m_{j_s}\right) q(z_1^{i_1}, \dots, z_n^{i_n}) f(z_1^{i_1}, z_2^{i_2}, \dots, z_{n+1}^{\mathbf{z}, i_{n+1}}) + E_2, (7)$$

where $z_i^{i_s} = a_i + (i_s - 1)p_i (\forall i, s = 1,2, \dots, n), \mathbf{z} = (z_1^{i_1}, z_2^{i_2}, \dots, z_n^{i_n})$,

$$E_2 = \frac{1}{16}\frac{Vol(D)}{90}\left[k^5(\bar{\mathbf{z}})\frac{\partial^4 f}{\partial x_{n+1}^4}(\bar{\mathbf{z}}, z_{n+1}^{\bar{\mathbf{z}}}) + \sum_{i=1}^{n} h_i^4 \frac{\partial^4}{\partial x_i^4}\left(k(\hat{\mathbf{z}}^{(i)}) f\left(\hat{\mathbf{z}}^{(i)}, z_{n+1}^{\hat{\mathbf{z}}^{(i)}}\right)\right)\right],$$

for some $\bar{\mathbf{z}}, \hat{\mathbf{z}}^{(i)} \in D$, and some $z_{n+1}^{\hat{\mathbf{z}}^{(i)}} \in \left(F(\hat{\mathbf{z}}^{(i)}), G(\hat{\mathbf{z}}^{(i)})\right)$.

III. ADJUSTED STEP-SIZE QUADRATURE IN MULTIPLE INTEGRAL

Set S_1, S_2 be the first term on the right-hand side of (3), (7), respectively. Assuming $\bar{\mathbf{x}} \approx \bar{\mathbf{z}}, \hat{\mathbf{x}}^{(i)} \approx \hat{\mathbf{z}}^{(i)}, \forall i = 1,2, \dots, n$. So, $E_1 \approx 16E_2$. Since $I = S_1 + E_1 = S_2 + E_2, E_2 \approx (S_2 - S_1)/15$. Hence,

$$|S_2 - I| \approx \frac{1}{15}|S_2 - S_1| \quad (8)$$

If $|S_2 - S_1| < 15\varepsilon$, then we accept that S_2 approximates I to within ε. Otherwise, for$|S_2 - S_1| \ge 15\varepsilon$, we reapply the above-mentioned procedure on smaller regions with a size of nearly $1/2^{n+1}$ of the original region B, and the expected tolerance for approximation of the integral over that sub-region is only $\varepsilon/2^{n+1}$. Now in such smaller region, we search for $\bar{\mathbf{x}} \approx \bar{\mathbf{z}}, \hat{\mathbf{x}}^{(i)} \approx \hat{\mathbf{z}}^{(i)}, \forall i = 1,2, \dots, n$. The size of such sub-regions becomes smaller and smaller, that is $|\bar{\mathbf{x}} - \bar{\mathbf{z}}| \to 0, \text{then} |\hat{\mathbf{x}}^{(i)} - \hat{\mathbf{z}}^{(i)}| \to 0, \forall i = 1,2, \dots, n$. The procedure always succeeds in finding an approximation of I lying to within the given tolerance. We set up a limitation for the iteration by requiring that the level of subdivision does not exceed a prior number N.This reduces the memory storage but still keeps the error within a designed bound.

IV. IMPLEMENTING THE ADJUSTED STEP-SIZE MULTIVARIATE QUADRATURE

Implementing the algorithm mentioned here can be performed for a particular moderate dimension. We present in this paper the one in two-variables case. The implementation is coded in Matlab.

INPUT integration region B,described by two functions $c = F(\mathbf{x}), d = G(\mathbf{x})$, integrand$f$, tolerance ε, limited level N.

OUTPUT approximation AP of I, or a message announces that the level N is exceeded.

```
function APP=DoubleINT(f,a,b,c,d,tol,N)
%-------------STEP 1------------
AP=0;
i=1;
L(i)=1;
eps(i)=15*tol;
n(1)=1;n(2)=4;n(3)=1;
m(1)=1;m(2)=4;m(3)=2;m(4)=4;m(5)=1;
R=[0];level=[];status=[];% THIS indicates which Subrectangle is using to do
%approximation and at which level of subdivision we are,
% and also to which circumstance we are facing

r1(i)=1;r2(i)=1+2^N;
F=[];G=[];
for j=1:(1+2^N)
    X(j)=a+(j-1)*(b-a)/2^N;
    F=[F,c(X(j))];
    G=[G,d(X(j))];
end
FT(i,:)=F;GT(i,:)=G;
%-------------STEP 2------------Do STEPS 3-5
while i>0
%----------STEP 3------------
   h(i)=(X(r2(i))-X(r1(i)))/2;
   ht(i)=(r2(i)-r1(i))/2;
   FS=FT(i,r1(i):r2(i));
   GS=GT(i,r1(i):r2(i));
if (L(i)>=N)|(mod(ht(i),2)~=0)
       AP='LEVEL EXCEEDED';
break;
end
for j=1:3
       x(j)=X(r1(i))+(j-1)*h(i);
       k(i,j)=0.5*(GS(1+(j-1)*ht(i))-FS(1+(j-1)*ht(i)));
for l=1:3
           y(l,j)=FS(1+(j-1)*ht(i))+(l-1)*k(i,j);
end
end
%==========SIMPSON=======
   S=0;
for l=1:3
for j=1:3
           S=S+h(i)*k(i,j)*n(l)*n(j)*f(x(j),y(l,j))/9;
end
end
%=======================
for j=1:5
       z(j)=X(r1(i))+(j-1)*h(i)*0.5;
       p(i,j)=0.25*(GS(1+(j-1)*ht(i)*0.5)-FS(1+(j-1)*ht(i)*0.5));
for l=1:5
           q(l,j)=FS(1+(j-1)*ht(i)*0.5)+(l-1)*p(i,j);
end
end
%===============COMPOSITE SIMPSON============
   T=0;
```

```
for l=1:5
for j=1:5
           T=T+h(i)*p(i,j)*m(l)*m(j)*f(z(j),q(l,j))/18;
end
end
%==================SAVE DATA==================
   u1=r1(i);
   u2=r2(i);
   u3=ht(i);
   u4=eps(i);
   u5=L(i);
   u6=FT(i,:);
   u7=GT(i,:);
%====================DATA for Information of Subdivision================
   level=[level,u5];
if size(level,2)>1
       count=0;
       j=1;
while j<= size(level,2)-1
if level(j)==level(size(level,2))
              count=count+1;
end
           j=j+1;
end
       R=[R,4-mod(count,4)];
end
%----------STEP 4------------
   i=i-1;
%----------STEP 5------------
if abs(S-T)<u4
       AP=AP+T;
       status=[status,"Pass"];
else
       status=[status,"Fail"];
%----------DATA for R1------
       i=i+1;
       r1(i)=u1;
       r2(i)=u1+u3;
       FT(i,:)=u6;
       GT(i,:)=0.5*(u6+u7);
       eps(i)=u4/4;
       L(i)=u5+1;
%----------DATA for R2------
       i=i+1;
       r1(i)=u1;
       r2(i)=u1+u3;
       FT(i,:)=GT(i-1,:);
       GT(i,:)=u7;
       eps(i)=eps(i-1);
       L(i)=L(i-1);
%----------DATA for R3------
       i=i+1;
       r1(i)=u1+u3;
       r2(i)=u2;
       FT(i,:)=u6;
       GT(i,:)=0.5*(u6+u7);
       eps(i)=eps(i-1);
       L(i)=L(i-1);
%----------DATA for R4------
       i=i+1;
       r1(i)=u1+u3;
       r2(i)=u2;
       FT(i,:)=GT(i-1,:);
       GT(i,:)=u7;
       eps(i)=eps(i-1);
       L(i)=L(i-1);
```

```
end
end
%------------END STEP 2---------
%------------STEP 6------------
fprintf('Table Of Process\n');
Tb=table;
Tb.Subrectangles=R';
Tb.Level_Of_Subdivion=level';
Tb.Result=status';
Tb
fprintf('Approximation : ');
APP=AP;
end
```

V. NUMERICAL EXPERIMENT

We approximate the double integral

$$I = \iint_B (x + y)xdxdy,$$

where $B = \{(x, y)|0 \leq x \leq 1, x^2 \leq y \leq x\}$.

The exact result is $I = \frac{11}{120} = 0.0919(6)$.Theobtained approximation to I is$AP = 0.0916621$ within the given tolerance $\varepsilon = 10^{-5}$. The limit level is $N = 4$. The result is shown in Table 1.

$Level_i$	Sub-region R_i	$Status_i$		$Level_i$	Sub-region R_i	$Status_i$		$Level_i$	Sub-region R_i	$Status_i$
1	0	Fail		3	4	Pass		3	2	Pass
2	4	Fail		3	3	Pass		3	1	Pass
3	4	Pass		3	2	Pass		2	1	Fail
3	3	Pass		3	1	Pass		3	4	Pass
3	2	Pass		2	2	Fail		3	3	Pass
3	1	Pass		3	4	Pass		3	2	Pass
2	3	Fail		3	3	Pass		3	1	Pass

Table 1. The procedure yielding the result produced by the implementation.

VI. CONCLUSION

The implementation constructed shows the effective on approximating the multiple integral in the moderate dimension. The flexibility in changing the step-size with respective to the variation of the integrand and the (boundary of) region of integration. Thank to this dynamical adaptation, the computational reduces comparing to the usual approach which use uniform partitions of the integration region. Matlab code written performs quite nicely.

The limitations of the approach still remains following the curse of dimensionality. This can somewhat be solved with the use of Monte-Carlo method. The paper could be developed in this approach in the future study.

REFERENCES

[1] John R. Rice. A Metalgorithm for Adaptive Quadrature, Journal of the ACMM Ozaki, Y. Adachi, Y. Iwahori, and N. Ishii, Application of fuzzy theory to writer recognition of Chinese characters, International Journal of Modelling and Simulation, 22(1), 1975, 61-82.
[2] M.K. William, Algorithm 145: Adaptive numerical integration by Simpson's rule". Communications of the ACM (Periodical), New York: ACM. **5** (12), 604–605
[3] R.L. Burden, J.D. Faires, Numerical Analysis (Brook/Cole, Cengage Learning, Boston 2000).
[4] S. C. Chapra, Applied Numerical Methods with MATLAB (McGraw-Hill, New York, 2012).