International Journal of Engineering Technology Science and Research
IJETSR
www.ijetsr.com
ISSN 2394 – 3386
Volume 4, Issue 5
May 2017

# Solution of Multidimensional Integration using Adaptive Monte Carlo Method with General Division Approach and its Validation

**Falguni Kudu, Tapan Santra*, AnupamMaity, Anik Roy**
Department of Electrical Engineering,
Kalyani Government Engineering College, Kalyani, Nadia, West Bengal, India

***Abstract-****This paper represents the technique to find out the solution of multidimensional integral using Adaptive Monte Carlo (AMC) method with the general division approach using an efficient way storage. The general quadrature method in this regards is laborious and monotonous too. The proposedAMC algorithm estimates the multidimensional integrals over a hyper-rectangular region using very simple technique, It uses iteratively the idea of separating the domain of integration into $2^s$ sub regions, where s is the number of coordinates of the hyper domain which will be divided in successive iterations. The algorithm divides the region with highest standard error in successive iterations. Thus the sample density is maximum in important (peak) region and minimum in the flat region of the overall integration domain, thus it reduces the computational burden. The proposed method is validated by two numerical examples and other existing techniques.*

***Keywords: Multidimensional integration, AMC, Storage mechanism, standard error***

## 1. Introduction

The value of the integrand can vary significantly in the domain of integration, and in most of the cases, there may be only small parts of the domain in which the integrand value varies prominently (non zero or non-constant). In this paper, the integrand which is complex, non-smooth with large number of dimensions is considered. The general quadrature method and other existing numerical techniques work very well for integration with one dimension. If the dimension increased these one dimensional numerical techniques are repetitively use, which is called cubature rules of integration. But this cubature technique becomes less efficient, monotonous and laborious for complex problem with higher dimensions [1]. So the different stochastic methods are introduced to get rid of these laborious quadrature techniques. Monte Carlo (MC) method is one of the greatest stochastic integration techniques. Though it is verly old technique, but recently it has become very popular and finds applications in physics, engineering, mathematics, In basic Monte Carlo method (MC)evaluation points are distributedrandomly over the integration domain and takes the sample average. As MC randomly distributes the points, it will waste calculations in the flat regions that are not important [2] for calculation. So to make the point distribution more uniform low discrepancy points (QMC) are introduced. Pskov [3] proved that QMC converges faster and smoother than MC.Xiaoqun Wang and Kai-Tai Fang [4] represented Quasi-Monte Carlo (QMC) methods, successfully used for high-dimensional integrals by reducing the effective dimensions of the problem. But after reducing the effective dimension integration error will be more. Schürer [5] compared of MC and QMC algorithms to integration routines based on the interpolatory cubature rules. AMC algorithm is advantageous over MC and QMC regarding convergence rate and computational burden. It distributes the sample points adaptively such that in important region sample density is more and less in the flat regions. So certainly the computational burden will be less over MC and QMC. TeemuPennanen and MattiKoivu [6] represented a new adaptive importance sampling (AIS) technique for the approximate evaluation of multidimensional integrals. Whereas known AIS algorithms try to find a sampling density that is approximately proportional to the integrand, the new

International Journal of Engineering Technology Science and Research
IJETSR
www.ijetsr.com
ISSN 2394 – 3386
Volume 4, Issue 5
May 2017

algorithm aims directly at the minimization of the variance of the sample average estimate. The algorithm was implemented in the C - programming language and compared with VEGAS and MISER. The information aboutthe important and non important regions are not always given. M.H. Alrefaei and H.M. Abdul-Rahman [7] addressed a AMC techniquewhich learns about the important and non important parts of the domain of integration as the integration .Santra et al. [8] represented an application of AMC algorithm to calculate the magnetic force between two ring magnets which also validated by the experimental results.

The proposedAMC algorithm estimates the multidimensional integrals over a hyper-rectangular region using very simple technique, It uses iteratively the idea of separating the domain of integration into $2^s$ sub regions, where s is the dimensions of the integration. The algorithm divides the region with highest integration error in successive iterations. Thus the sample density is maximum in important (peak) region and minimum in the flat region of the overall integration domain, thus it reduces the computational burden. A novel storage mechanism is introduced in the algorithm. The proposed method is validated by numerical examples and other existing techniques.

## 2. Solution of the Multidimensional Integration

### 2.1. Theory of AMC with general division (s-division) approach

Let a $d$ -dimensional integral is given by (1)

$I(f) = \int_H f(x)dx$ (1) Where $x \in \Re^d, d \geq 1, H$ denotes the $d$ -dimensional region of intregration and the integrand $f(x): H \to \Re$ is integrable on $H$. And $x$ is the random vector distributed over the region $H$. The basic principle of Monte Carlo (MC) technique is to generate N independent random samples $x^1, x^2, \cdots x^d$ according to the uniform density over H and to compute the sample mean using (2) [7].

$$\hat{I}_N = (V/N)\sum_{i=1}^{N} f(x^i) \quad (2)$$

The main concept of the AMC integration is to apply the basic MC method to smaller sub regions of the original integration domain (see Dahl [9]) and, this needs a subdivision strategy. There are two common types of subdivision strategy (see Schürer [6]); local subdivision strategy and global subdivision strategy. In this paper the global subdivision strategy is used in the AMC algorithm.

**The global subdivision strategy**

In this paper the iterative AMC algorithm uses a global subdivision strategy. In the first iteration, the basic MC is applied to the whole integration domain to estimate the integral and the standard error. The region with its integral value and standard error is stored in a region collection. In the next iteration, the whole region is divided into $2^s$ sub regions and the basic MC method is applied to each of these sub regions. The sub regions are stored in a region collection. Next, the region that has the largest estimated error is taken and divided into new sub regions. This process continues until a stopping criterion is satisfied. The estimated integral is the sum of the estimated integrals at each sub region from the region collection, see Dahl [9] and Schürer [6]. The following algorithm describes the general form of being an AMC integration algorithm, seeAlrefaei, Rahman [2]. To get a clear idea about AMC, some notations are frequently required. Those are given as follows:

$\Omega_i$: Region collection in iteration $i$.

$H_i(j)$: Subregin that belongs to $\Omega_i$ with index j;

$\hat{I}_{ij}$: Basic MC integration estimate over the region $H_i(j)$ initeration $i$.

$\hat{I}_i$: Integral estimate over all the integral domain in iteration $i$. So, $\hat{I}_i = \sum_j \hat{I}_{ij}$ $E_{ij}$: Standard estimated error of integration over the sub region of index $j$ in iteration $i$;. $E_{ij} = V_{ij}\frac{\sigma_f|_{H_i(j)}}{\sqrt{N_{ij}}}$

International Journal of Engineering Technology Science and Research
IJETSR
www.ijetsr.com
ISSN 2394 – 3386
Volume 4, Issue 5
May 2017

Where $V_{ij}$ is the volume of the region $H_i(j)$, $N_{ij}$ is the number of random points taken in $H_i(j)$ and $\hat{\sigma}_f|_{H_i(j)}$ is the estimated standard deviation of $f$ over the sub region $H_i(j)$.

$\hat{E}_i$: Standard estimated error of integration over the sub region of index $j$ in iteration $i$; $(H_i(j))$, so, $\hat{E}_i = \sqrt{\sum_j \hat{E}_{ij}^{\,2}}$.

$\alpha_i$: Promising index; the index of the region that has the maximum estimated error in iteration $i$;i.e., $\alpha_i = \arg\max_j \hat{E}_{ij}$

$H_i^*$: Promising region; the region that has the maximum estimated error in iteration $i$, so $H_i^*$ is the region whose index is the promising index $H_i^* = H_i(\alpha_i)$.

***S-division of hyper rectangle***

The s-division of a hyper-rectangle $H = [a_1, b_1] \times [a_2, b_2] \times \ldots \times [a_d, b_d] \subset \Re^d$ can be done by taking s coordinates (where s is a positive integer with $1 \le s \le d$) and divide each coordinate into two equal (or, non-equal) intervals. To make a one-division of the region $H_i(j)$, the fundamental property of division of the Riemann integral in one dimension is generalized to d-dimensions as given by (3)

$$\int_{a_d}^{b_d} \mathrm{L} \int_{a_k}^{b_k} \mathrm{L} \int_{a_1}^{b_1} f(\mathrm{x}) = \int_{a_d}^{b_d} \mathrm{L} \int_{a_k}^{c} \mathrm{L} \int_{a_1}^{b_1} f(\mathrm{x}) + \int_{a_d}^{b_d} \mathrm{L} \int_{c}^{b_k} \mathrm{L} \int_{a_1}^{b_1} f(\mathrm{x}) \qquad (3)$$

where $c \in (a_k, b_k)$, $1 \le k \le d$ and $\mathrm{x} \in \Re^d$. To make a two division of the same region, the idea of the division of the square into four sub regions is used and generalized to get the following formula (4)

$$\begin{aligned} \int_{a_d}^{b_d} \mathrm{L} \int_{a_{k1}}^{b_{k1}} \mathrm{L} \int_{a_{k2}}^{b_{k2}} \mathrm{L} \int_{a_1}^{b_1} f(\mathrm{x})d\mathrm{x} &= \int_{a_d}^{b_d} \mathrm{L} \int_{a_{k1}}^{c_1} \mathrm{L} \int_{a_{k2}}^{c_2} \mathrm{L} \int_{a_1}^{b_1} f(\mathrm{x})d\mathrm{x} + \int_{a_d}^{b_d} \mathrm{L} \int_{c_1}^{b_{k1}} \mathrm{L} \int_{a_{k2}}^{c_2} \mathrm{L} \int_{a_1}^{b_1} f(\mathrm{x})d\mathrm{x} \\ &+ \int_{a_d}^{b_d} \mathrm{L} \int_{a_{k1}}^{c_1} \mathrm{L} \int_{c_2}^{b_{k2}} \mathrm{L} \int_{a_1}^{b_1} f(\mathrm{x})d\mathrm{x} + \int_{a_d}^{b_d} \mathrm{L} \int_{c_1}^{b_{k1}} \mathrm{L} \int_{c_2}^{b_{k2}} \mathrm{L} \int_{a_1}^{b_1} f(\mathrm{x})d\mathrm{x} \end{aligned} \qquad (4)$$

with $c_j \in (a_{kj}, b_{kj})$, where $1 \le kj \le d$ for $j = 1, 2, \ldots\ldots$, and $\mathrm{x} \in \Re^d$. Note that, when one-division is used, then the region is divided into two sub regions; whereas when two-division approach is used, then the region is divided into four sub regions. Thus, in general, s-division divides the whole region into $2^s$ sub regions. In iteration $0, \Omega_0$ contains only one region; that means the whole integration domain. In iteration 1, the whole integration domain is divided into $2^s$ sub regions, $|\Omega_1| = 2^s$. In iteration 2, the region which has the maximum estimated error is selected and then this sub region is divided into $2^s$ sub regions, so $|\Omega_2| = 2^s - 1 + 2^s = 2 \times 2^s - 1$. Thus in general,

$$|\Omega_i| = 2^s i - (i-1) = i(2^s - 1) + 1. \qquad (4)$$

$$\Omega_i = \{H_i(1), H_i(2), \mathrm{K}, H_i(v_i)\} \qquad (5)$$

where $v_i = i(2^s - 1) + 1$. The total estimated integral in iteration $i$ is given by (6)and the total estimated error in iteration $i$, $\hat{E}_i$ is given by (7)

$$\hat{I}_i = \sum_{j=1}^{v_i} \hat{I}_{ij}, \qquad (6)\ \hat{E}_i = \sqrt{\sum_{j=1}^{v_i} \hat{E}_{ij}}$$

(7)

International Journal of Engineering Technology Science and Research
IJETSR
www.ijetsr.com
ISSN 2394 – 3386
Volume 4, Issue 5
May 2017

In each iteration $i$, s distinct coordinates is chose to be divided, namely; $r_{i1},\ldots,r_{is}$ ,and a reasonable values for $c_1,c_2,\mathrm{K},c_s$ ( where $c_j$ divides the axis $r_{ij}, j=1,2,\mathrm{K},s$ ). So, after selecting the divided coordinates $r_{i1},r_{i2},\mathrm{K},r_{is}$, choose $c_j=(a_{kj}-b_{kj})/2, j=1,2,\mathrm{K},s$

## 2.2 Storage mechanism

Each d-dimensional subregion is represented by $d$ lower bounds and $d$ upper bounds, namely; $(a_1,K,a_d)$ and $(b_1,\mathrm{L},\mathrm{b}_d)$, respectively. To store all these elements two $v_i\times d$ arrays ; $A^i$ and $B^i$ where $i$ denotes the $i^{th}$ iteration. The $k^{th}$ row of $A^i$ and $B^i$ stores the lower bounds and upper bounds, respectively, of the region $H_i(k)$. The $l^{th}$ column of $A^i$ and $B^i$ contains the lower bounds upper bounds respectively, of the $l^{th}$ coordinate of all sub regions $H_i(j), j=1,2,K,v_i$ .For an example,mlet the sub regions in iteration $i$ are defined as follows:

$$H_i(j)\equiv\{\mathrm{x}|\mathrm{x}\in[a_{j1},b_{j1}]\times[a_{j2},b_{j2}]\times L\times[a_{jd},b_{jd}]\}, j=1,2,.....,v_i, \quad (8)$$

Then all these sub regions are represented by the following two arrays.

$$A^i=\begin{pmatrix} a_{11} & a_{12} & \mathrm{L} & \mathrm{L} & a_{1d}\\ a_{21} & a_{22} & \mathrm{L} & \mathrm{L} & a_{2d}\\ M & & & & \\ a_{v_i1} & a_{v_i2} & \mathrm{L} & \mathrm{L} & a_{v_id}\end{pmatrix}, \quad B^i=\begin{pmatrix} b_{11} & b_{12} & \mathrm{L} & \mathrm{L} & b_{1d}\\ b_{21} & b_{22} & \mathrm{L} & \mathrm{L} & b_{2d}\\ \mathrm{M} & & & & \\ b_{v_i1} & b_{v_i2} & \mathrm{L} & \mathrm{L} & b_{v_id}\end{pmatrix} \quad (9)$$

Where L represents coloums and M represents rows. Also, the estimated MC integrals $(\hat{I}_{ij})$ and the standard estimated errors $(\bar{E}_{ij})$ over each region $H_i(j)$, $j=1,\ldots,v_i$, from $\Omega_i$. are stored by a vector array $TM^i$ as given by (10)

$$TM^i=[\hat{I}_{i1},\mathrm{K}\,\mathrm{K},\hat{I}_{iv_i}] \quad (10)$$

$$\text{Where } \hat{I}_{ij}=\frac{V_{ij}}{N_{ij}}\sum_{k=1}^{N_{ij}}f(\mathbf{x}_k), j=1,2,\ldots,v_i \quad (11)$$

With $x_k=(x_{k1},x_{k2},\ldots,x_{kd})$, where $x_{kl}$ is a random variable in the interval $(A_i(j,l),B^i(j,l))$ for $l=1,2,\ldots,d$ ; $N_{ij}$ is the number of random points generated in $H_i(j)$ and $V_{ij}$ is calculated as follows by (12).

$$V_{ij}=\prod_{k=1}^{d}(B^i(j,k)-A^i(j,k)) \quad (12)$$

To store the standard estimated errors $\bar{E}_{ij}$ over all the regions of $\Omega_i$ ,the vector array $\bar{E}_{ij}$ is used as shown in (13).

$$\mathrm{TE}^i=[\bar{E}_{i1},\mathrm{K},\bar{E}_{iv_i}] \quad (13)$$

where $\bar{E}_{ij}$ is calculated using formula (14).

$$\bar{E}_{ij}=V_{ij}\frac{\bar{\sigma}_f|_{H_i(j)}}{\sqrt{N_{ij}}} \quad (14)$$

Where $V_{ij}$ is the volume of the region $H_i(j)$, $N_{ij}$ is the number of random points taken in $H_i(j)$ and $\bar{\sigma}_f|_{H_i(j)}$ is the estimated standard deviation of $f$ over the sub region $H_i(j)$.

**The transition from iteration $i$ to iteration $i+1$**

In iteration $i$, let $\Omega_i$ be represented by the arrays $A^i$ and $B^i$, the promising region $H_i^*=H_i(\alpha_i)$ willbe divided

International Journal of Engineering Technology Science and Research
IJETSR
www.ijetsr.com
ISSN 2394 – 3386
Volume 4, Issue 5
May 2017

into $2^s$ sub regions $H_{i+1}(\alpha_i), H_{i+1}(\alpha_i+1), \ldots, H_{i+1}(\alpha_i+2^s-1)$. $\Omega_i$ is updated as per (15).

$$\Omega_{i+1} = \{H_i(1), \mathrm{K}, H_i(\alpha_i-1), H_{i+1}(\alpha_i+1), \mathrm{K}\ \ \mathrm{K}, H_{i+1}(\alpha_i+2^s-1), H_i(\alpha_i+1), \mathrm{K}, H_i(v_{i+1})\}. \quad (15)$$

It is noted that $\Omega_{i+1}$ is completely determined by the matrices $A^{i+1}$ and $B^{i+1}$;so only one row $\alpha_i$ will be replaced by $2^s$ new rows in both $A^{i+1}$ and $B^{i+1}$. The two arrays $A^i$ and $B^i$ are updated into $A^{i+1}$ and $B^{i+1}$ respectively, as follows:

$$MA = \begin{pmatrix} A(\alpha_i,1) & A(\alpha_i,2) & \mathrm{L} & A(\alpha_i,s-1) & A(\alpha_i,s) \\ C_1 & A(\alpha_i,2) & \mathrm{L} & A(\alpha_i,s-1) & A(\alpha_i,s) \\ A(\alpha_i,1) & C_2 & \mathrm{L} & & \\ C_1 & C_2 & \mathrm{L} & & \mathrm{M} \\ A(\alpha_i,1) & A(\alpha_i,2) & \mathrm{M} & \mathrm{M} & \mathrm{M} \\ C_1 & A(\alpha_i,2) & \mathrm{L} & C_{s-1} & \mathrm{M} \\ A(\alpha_i,1) & & \mathrm{L} & C_{s-1} & \mathrm{M} \\ C_1 & & \mathrm{L} & \mathrm{M} & \mathrm{M} \\ & & \mathrm{L} & A(\alpha_i,s-1) & C_s \\ & & \mathrm{M} & \mathrm{M} & \mathrm{M} \\ * & * & * & * & * \end{pmatrix}_{2^s \times s} \quad (16)$$

$$MB = \begin{pmatrix} C_1 & C_2 & \mathrm{L} & C_{s-1} & C_s \\ B(\alpha_i,1) & C_2 & \mathrm{L} & C_{s-1} & C_s \\ C_1 & B(\alpha_i,2) & \mathrm{L} & & \\ B(\alpha_i,1) & B(\alpha_i,2) & \mathrm{L} & & \mathrm{M} \\ C_1 & C_2 & \mathrm{M} & \mathrm{M} & \mathrm{M} \\ B(\alpha_i,1) & C_2 & \mathrm{L} & B(\alpha_i,s-1) & \mathrm{M} \\ C_1 & & \mathrm{L} & B(\alpha_i,s-1) & \mathrm{M} \\ B(\alpha_i,1) & & \mathrm{L} & \mathrm{M} & \mathrm{M} \\ & & \mathrm{L} & C_{s-1} & B(\alpha_i,s) \\ & & \mathrm{M} & \mathrm{M} & \mathrm{M} \\ * & * & * & * & * \end{pmatrix}_{2^s \times s} \quad (17)$$

The $(\alpha_i-1)$ rows after $2^s$ rows of $A^{i+1}$ and $B^{i+1}$ remain the same as the first $(\alpha_i-1)$ rows of $A^i$ and $B^i$, respectively. The last $(v_i-\alpha_i)$ rows of $A^{i+1}$ and $B^{i+1}$ are the same as the last $(v_i-\alpha_i)$ rows of $A^i$ and $B^i$ respectively. The remaining $2^s$ rows of $A^{i+1}$ and $B^{i+1}$ (which are the rows of $1, 2, \ldots, 2^s$, that represent the new $2^s$ regions in $\Omega_{i+1}$ ) are division of the $\alpha_i$ row of $A^i$ and $B^i$ ,according to the new $2^s$ devided regions. After upgradationthe $A^{i+1}$ and $B^{i+1}$ matrixes look like (16) and (17).

Where * (star) denotes that the value in the corresponding location is unaltered (the original value in the $\alpha_i$ row).It is noted that MA nd MB are not portion arrays.The corresponding modified array in $B'^{+1}$ (the $\mathrm{MB}_{2^s \times s}$ array) by replacing the unchanged values (the stars) in MA by $c_j$'s in the $j^{th}$ column of MB, $j = 1, \mathrm{K}, s,$ and replacing the $c_j$'s in the $j^{th}$ column of MA by stars (become unchanged values); or simply, by reversing the rows of the MA array. For example, if one division approach is taken into consideration, then

International Journal of Engineering Technology Science and Research
IJETSR
www.ijetsr.com
ISSN 2394 – 3386
Volume 4, Issue 5
May 2017

$$MA=\begin{pmatrix} * \\ c_1 \end{pmatrix}, \quad MB=\begin{pmatrix} c_1 \\ * \end{pmatrix}.$$

In this case, $A^i$ and $B^i$ are updated as follows:

$$\left.\begin{array}{l} for\ i=1 \text{ to } d \text{ (where } d \text{ is the dimension) take a variable } p=0 \\ for\ x=1 \text{ to } 2^d/2^i \\ for\ \mathrm{y}=1:2^{(i-1)} \text{ then } p=p+1; \\ MA(p,i)=A(m,i); \\ MB(p,i)=(A(m,i)+B(m,i))/2; \end{array}\right\} \quad (18)$$

In the above process $\mathrm{A}^i$ is updated.

$$\left.\begin{array}{l} for\ i=1 \text{ to } d \text{ (where } d \text{ is the dimension) take a variable } p=0 \\ for\ x=1 \text{ to } 2^d/2^i \\ for\ z=1:2^{(i-1)} \text{ then } p=p+1; \\ MA(p,i)=(A(m,i)+B(m,i))/2; \\ MB(p,i)=B(m,i); \end{array}\right\} \quad (19)$$

In the above process $B^i$ is updated. It is noted that $v_{i+1}=v_i+1=i+2.$ If two division approach is taken into consideration ,then

$$MA=\begin{pmatrix} * & * \\ * & C_2 \\ C_1 & * \\ C_1 & C_2 \end{pmatrix}, \quad MB=\begin{pmatrix} C_1 & C_2 \\ C_1 & * \\ * & C_2 \\ * & * \end{pmatrix} \quad (20)$$

There are many ways for selecting the values of $c(1,i), i=1,2,\mathrm{K},s.\ c(1,i)$ can be selected randomly in the interval $[A^i(\alpha_i,i), B^i(\alpha_i,i)]$, or the midpoint of this interval .In this paper the choice of $c(1,i)$ as the middle point of the boundaries is considered, so, $c(1,i)=\dfrac{A^i(\alpha_i,i)+B^i(\alpha_i,i)}{2}$ (21)The vector $\mathrm{TM}^i(l)$ that stores the estimates of the integration over all sub regions $j=1,2,\mathrm{K},v_i$ is updated as follows:

$$\left.\begin{array}{l} for\ k=(2^d+1):v_i+1 \\ if\ k<(m+2^d) \\ MA(k,l)=A((k-2^d),l); \\ MB(k,l)=B((k-2^d),l); \\ TE(1,k)=E(1,(k-2^d)); \\ TM(1,k)=I(1,(k-2^d)); \\ else \\ if\ k>(m+2^d) \\ MA((k-1),l)=A((k-2^d),l); \\ MB((k-1),l)=B((k-2^d),l); \\ TE(1,(k-1))=E(1,(k-2^d)) \\ TM(1,(k-1))=I(1,(k-2^d)); \end{array}\right\} \quad (22)$$

International Journal of Engineering Technology Science and Research
IJETSR
www.ijetsr.com
ISSN 2394 – 3386
Volume 4, Issue 5
May 2017

In the above process $TM^i$ is updated.Therefore, the mean values of the intregrand function of a region. The total estimated error in iteration $i$ ; $(TE)^i$ is updated in the same manner. So, in iteration $(i+1)$, the estimates are estimated using formula (14).

### 2.3. Algorithm of AMC

**Step 1:**Initialize all the variables, i.e. the limits of integration, upgrading matrix MA and MB, final matrix TM and TE. Initialize the number of iterations. Initialize the number of dimensions.

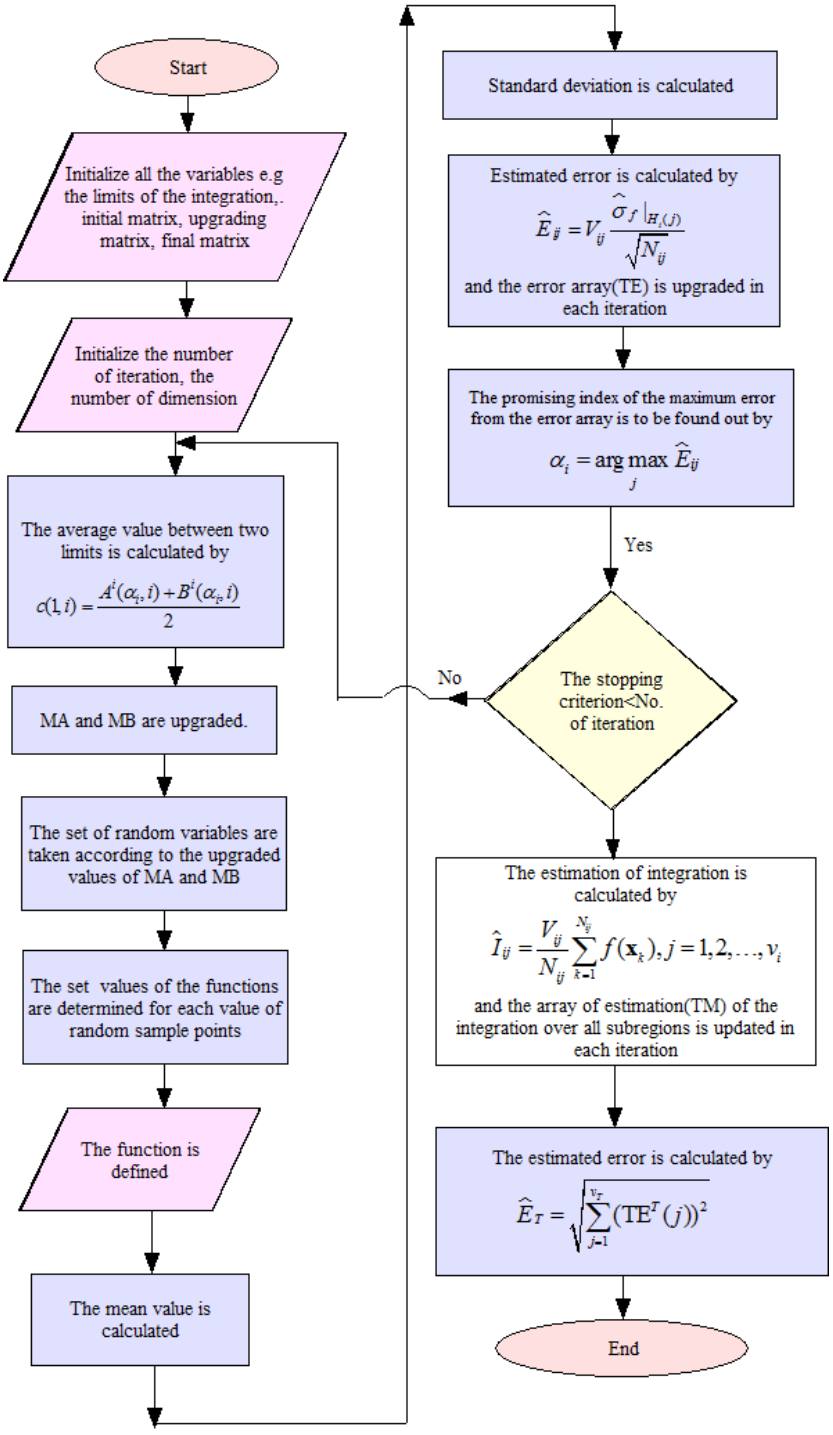



Fig.1 Flowchart for AMC

International Journal of Engineering Technology Science and Research
IJETSR
www.ijetsr.com
ISSN 2394 – 3386
Volume 4, Issue 5
May 2017

**Step 2:** Then the average value between two limits is calculated using formula $c(1,i)=\frac{A^i(\alpha_i,i)+B^i(\alpha_i,i)}{2}$ .

**Step 3:**The upgrading matrix, i.e. MA and MB are upgraded.

**Step 4:** The set of random variables are taken according to the upgraded MA and MB.

**Step 5:** Then the integrand function $f(x)$ is defined.The values of the function are determined for each value of random sample points.

**Step 6:** Then the mean value of the function from each division is calculated and the array of the estimation of the mean (TM) over all sub regions is upgraded.

**Step 7:** Standard deviationsof each divisions $\sigma_f$ is calculated and the estimation of error is calculated using formula $E_{ij}=V_{ij}\frac{\sigma_f|_{H_i(j)}}{\sqrt{N_{ij}}}$ and the error array(TE) is upgraded in each iteration.

**Step 8:** Then the promising index of the maximum error from the error array is found out using the formula $\alpha_1=\arg\max_j E_{ij}$

**Step 9:** Then goto step 2 until the stopping criterion is achieved.

**Step 10:**Finally after all iterations the estimation of integration $\hat{I}_{ij}$ and estimated error $E_T$ is calculated using formula

$$\hat{I}=\frac{V_{ij}}{N_{ij}}\sum_{k=1}^{N_{ij}}f(\mathbf{x}_k), i=1,2,.....n \text{ and } j=1,2,\ldots,v_i$$

$$E_T=\sqrt{\sum_{j=1}^{v_T}(\mathrm{TE}^T(j))^2}$$

Table 2 Output of AMC with 1-division approach up to third iteration for single Gaussian integration

| Iteration number $i$ | Region collection $H_i$ | Sub region $W_i(j)$ | Estimated Integral $\hat{I}_{i,j}$ | Estimated total Integral $\hat{I}_i$ | Estimated Error $\hat{E}_{i,j}$ | Estimated total error $\hat{E}_i$ |
|---|---|---|---|---|---|---|
| 0 | $H_0$ | $W_0(1)$ | 6.1454 | 6.1454 | **0.0852** | 0.0852 |
| 1 | $H_1$ | $W_1(1)$ | 3.0997 | 6.1734 | 0.0425 | 0.0603 |
| | | $W_1(2)$ | 3.0737 | | **0.0429** | |
| 2 | $H_2$ | $W_2(1)$ | 3.0997 | 6.2601 | **0.0425** | 0.0496 |
| | | $W_2(2)$ | 2.7479 | | 0.0250 | |
| | | $W_2(3)$ | 0.4125 | | 0.0055 | |

International Journal of Engineering Technology Science and Research
IJETSR
www.ijetsr.com
ISSN 2394 – 3386
Volume 4, Issue 5
May 2017

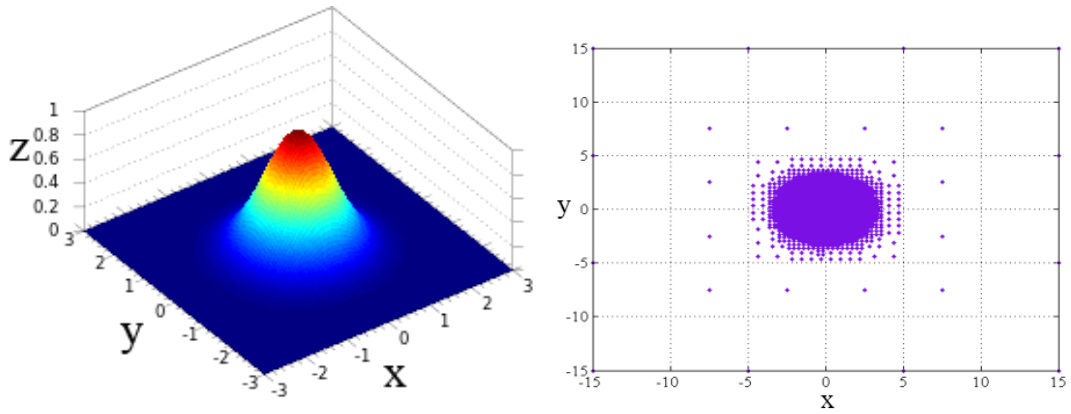
Table 3 Output of AMC with 2-division approach up to third iteration for single Gaussian integration

| Iteration number $i$ | Region collection $H_i$ | Sub region $W_i(j)$ | Estimated Integral $\hat{I}_{i,j}$ | Estimated total Integral | Estimated Error $\hat{E}_{i,j}$ | Estimated total error $\hat{E}_i$ |
|---|---|---|---|---|---|---|
| 0 | $H_0$ | $W_0(1)$ | 6.1454 | 6.1454 | **0.0852** | 0.0852 |
| 1 | $H_1$ | $W_1(1)$ | 1.5666 | 6.2228 | 0.0216 | 0.0429 |
| | | $W_1(2)$ | 1.5491 | | 0.0213 | |
| | | $W_1(3)$ | 1.5781 | | **0.0217** | |
| | | $W_1(4)$ | 1.5290 | | 0.0213 | |
| 2 | $H_2$ | $W_2(1)$ | 1.5666 | **6.2077** | **0.0216** | **0.0375** |
| | | $W_2(2)$ | 1.5491 | | 0.0213 | |
| | | $W_2(3)$ | 0.1772 | | 0.0016 | |
| | | $W_2(4)$ | 1.1818 | | 0.0051 | |
| | | $W_2(5)$ | 0.0272 | | 0.0004 | |
| | | $W_2(6)$ | 0.1767 | | 0.0016 | |
| | | $W_2(7)$ | 1.5290 | | 0.0213 | |

## 3. Numerical examples

In this paper different types of test functions are taken into consideration and the results are discussed. The integration of Gaussian function, Double Gaussian function is analyzed by the algorithms of AMC method with general division approach. Table 1 represents the two test functions. First, the integration of single peak Gaussian function is calculated using 1-division and 2-division approach.

### 3.1 Result and discussion



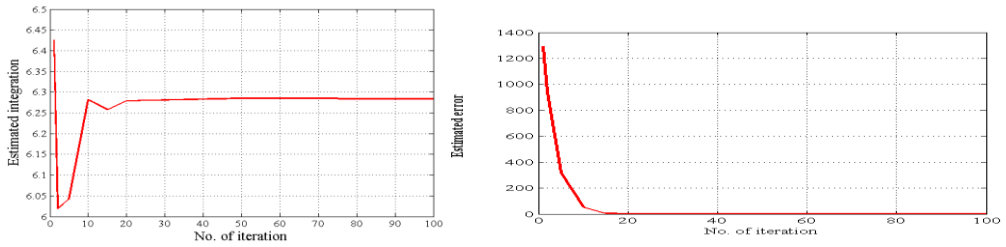

(a) (b)

Fig. 2 (a) Gaussian function with integration limits (b) Sample density in different subregion for 2-AMC

Fig. 2 (a) shows the single peak Gaussian function with interrogation limits x= (-3 to +3) and y=(-3 to +3). Fig. 2 (b) shows the sample points in different sub-regions in the domain. As the maximum variation of error occurs at the peak regions the square in this region succecively divided in each iteration, so the sample points are concentrated at the peak region og the Gaussian function as shown in Fig. 2 (b). The result of the proposed AMC algorithm is given in Table 1 and Table 2 for 1-division and 2-division AMC respectively up to third iterations. Dimensions of the problem, $d = 2$, number of iteration $M = 100$ and number of random samples in each region $N = 10000$. A MATLAB code is developed to calculate the integration. Two detailed outputs are given, first 1-AMC and then 2-AMC. In Table 2 outputs from 1- AMC is given, up to third iteration. In iteration 0 the estimated integral is truly the basic MC over the entire domain W. In iteration 1, W

International Journal of Engineering Technology Science and Research
IJETSR
www.ijetsr.com
ISSN 2394 – 3386
Volume 4, Issue 5
May 2017

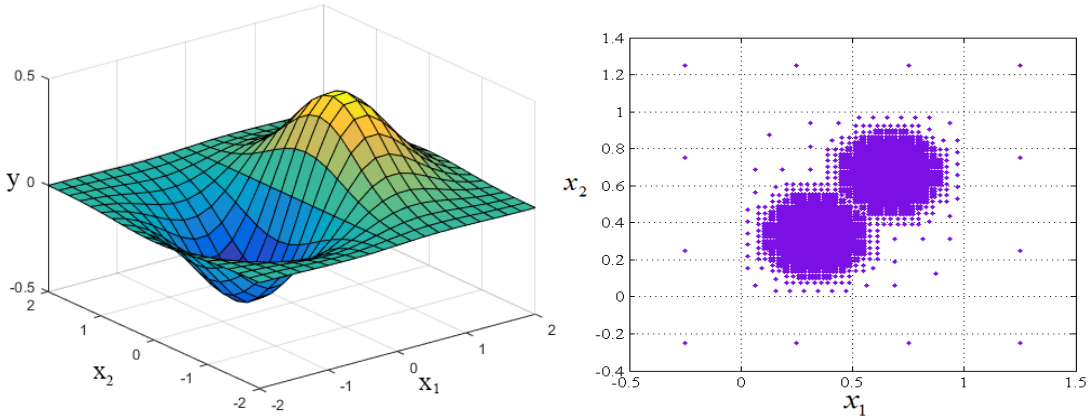
is divided into two $(2^1)$ sub regions: $W_1(1)$ and $W_1(2)$. The basic MC is applied to each region and $W_1(2)$ becomes the promising region due to high estimated error. So in iteration 2, $W_1(2)$ is divided into another two sub-regions $W_2(2)$ and $W_2(3)$. In this way the iteration progresses and final estimated integral is given by the summation of basic MC integral of all




(a) (b)

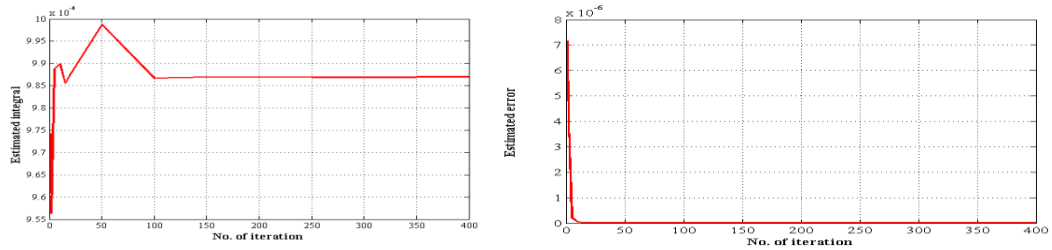
Fig. 3 (a) Estimated integration with iteration (b) Estimated error with iteration for single peak Gaussian function using the 2-AMC method.

Fig. 3 shows the estimated integral and estimated error with iterations. It is observed that estimated integral is achieved after 30th iteration and it is 6.2835. Initially the error fluctuates with large amplitude because the sub-region was large as the sub-region comes close to the peak the error becomes smaller and gradually it saturated to 0.0032 .




(a) (b)

Fig. 4 (a) Double Gaussian function with integration limits (b) Sample density in different sub-region for 2-AMC.




(a) (b)

Fig. 5 (a) Estimated integration with iteration (b) Estimated error with iteration for double peak Gaussian function using the 2-AMC method.

Fig. 4 (a) shows the double peak Gaussian function with integration limits x= (-3 to +3) and y=(-3 to +3). Fig. 4 (b) shows the sample points in different sub-regions in the domain. In the same manner using 2-AMC the

International Journal of Engineering Technology Science and Research
IJETSR
www.ijetsr.com
ISSN 2394 – 3386
Volume 4, Issue 5
May 2017

integral and the error is estimated for the double peak Gaussian function. The final value of the integration ( $9.87\times10^{-4}$ ) and estimated error ($1.52\times10^{-10}$) is achieved after 120$^{th}$ iterations.

Table 4 Comparison between Cubature rule MC an 2-AMC

| | | Cubature rule | MC with 400000 samples | 2-AMC with 10000 samples per region |
|---|---|---|---|---|
| Single Gaussian | Error | 0.0051 | 0.0061 | 0.0032 |
| | Time | 172 ms | 35 ms | 45 ms |
| Double Gaussian | Error | 0.0058 | 0.0067 | $1.52\times10^{-10}$ |
| | Time | 312 ms | 71 ms | 98 ms |

From Table 4 it is observed that Cubature rule takes more computational time than MC method with almost same integration error. Whereas, 2-AMC method incurred, less error with less computational time. If the dimension of the integration is increased this result will be more favorable for 2-AMC method.

## 4. Conclusion

In this paper, an algorithm is developed to obtain the solution of a multidimensional integral over a hyper rectangular region using an adaptive Monte Carlo (AMC) technique with general division ($2^s$) approach and also modify the storage mechanism of existing AMC, discussed in the review of literatures. This adaptive algorithm uses iteratively the basic idea of dividing the integration domain into sub-regions depending on the estimated standard error. We have discussed the storage mechanism and how to update this storage in an easy and efficient way. The algorithm is tested using Gaussian and double Gaussian function and compared with other existing method of integration. It is observed that the proposed 2-AMC method provides a better solution than many other methods regarding integration error and computational burden.

## References

[1] Mishra. M and Gupta. N, (2008) "Monte Carlo integration technique for the analysis of electromagnetic Scattering from conducting surfaces" *PIER, 79, 91–106.*

[2]Qiang Zhao, Guo Liu, Guiding Gu,2013,"Variance Reduction Techniques of Importance Sampling Monte Carlo Methods for Pricing Options",Journal of Mathematical Finance, 2013, 3, 431-436

[3] Paskov, S. H. and Traub, J. F. (1995), Faster evaluation of financial derivatives, J. Portfolio Management, 22(1), 113-120.

[4]Xiaoqun Wang and Kai-Tai Fang,2002," The effective dimension and quasi-Monte Carlo integration", Journal of Complexity ,19 , 101–124

[5] Rudolf Schürer ,2003," A Comparison between (Quasi-)Monte Carloand Cubature Rule Based Methods for Solving High-dimensional Integration Problems",Elsevier Science, MSC: 65C05, 65D30, 65D32.

[6]TeemuPennanenandMattiKoivu,"An Adaptive Importance Sampling Technique", Department of Management Science, Helsinki School of Economics, PL 1210,00101 Helsinki, Finland

[7] M.H. Alrefaei and H.M. Abdul-Rahman,(2007) "An adaptive Monte Carlo integration algorithm with general division approach", *Math. Comput.Simul*., doi:10.1016/j.matcom.2007.09.009.

[8] Santra, T., Roy, D., and Yamada, S., 2016, "Calculation of Force between Two Ring Magnets Using Adaptive Monte Carlo Technique with Experimental Verification," Progress in Electromagnetics Research M., 49, pp.181-193.