Communications on Applied Nonlinear Analysis
ISSN: 1074-133X
Vol 31 No. 8s (2024)

Function Quadrature Using Quartic Spline Interpolation

Kallur V Vijayakumar [1*], Hariprasad A S [2], Shilpa R[3]

[1] Kallur V Vijayakumar, BMS Institute of Technology and Management, Bengaluru-560064. kallurvijayakumar@bmsit.in

[2]Hariprasad A S, Rajarajeswari College of Engineering, Bengaluru-560074. ashariprasad@yahoo.co.in

[3]Shilpa R, Research Scholar, Rajarajeswari College of Engineering, Bengaluru-560074. Shilpa8906@gmail.com* corresponding author

Article History:

Received: *20-04-2024*

Revised: *10-06-2024*

Accepted: *24-06-2024*

Abstract:

This research paper explores the application of quartic splines for numerical integration, particularly in situations with curved boundaries in two-dimensional space. The paper dives into the concept of quartic splines, which are piecewise polynomial functions where each piece is a polynomial of the fourth degree. These splines are designed to ensure continuity of the function itself, its first, second, and third derivatives across the boundaries between the pieces. The core technique involves using quartic splines to approximate the curved boundary of a two-dimensional domain.

This allows for the calculation of integrals defined over that domain by transforming it into a simpler shape (often a rectangle) where traditional numerical integration methods like Gauss-Legendre quadrature can be applied effectively.

The paper validates the method by applying it to various curved domains and comparing the obtained integral values with known analytical solutions (if available). The absolute errors are calculated and tabulated, demonstrating the accuracy of the approach. The paper showcases the efficiency of the method by solving numerical integration problems from existing literature. These examples serve as evidence that the method is effective and can handle real-world scenarios. A significant advantage of this method is its ability to handle situations where only coordinate data for the boundary is available. Quartic splines can accurately reconstruct the curved boundary from these coordinates, enabling the integration process.

Keywords: Numerical Integration, Quadrature, Spline, curved boundary, spline.

Introduction

Numerical integration, the art of approximating definite integrals through computational methods, finds widespread application in various scientific and engineering disciplines. Among different techniques, quadrature using quartic splines emerges as a robust and accurate approach, offering numerous benefits over earlier traditional approaches. In this comprehensive discussion, we delve into the theoretical foundations, practical implementation, and performance analysis of quartic spline quadrature, equipping you with the knowledge to confidently apply it to your own integration problems. Definite integrals represent the area under a curve within a specified interval. However, analytically solving integrals becomes cumbersome or impossible for many functions. Numerical integration bridges this gap by approximating the area through weighted evaluations of the function at specific points, known as nodes. The objective is to find a formula that expresses the integral as a linear combination of these function values, minimizing the error between the approximation and the true integral. Spline functions offer flexibility in approximating complex curves. Quartic splines, polynomials of degree four, can closely fit diverse function behaviors through piecewise construction. By partitioning the interval of integration into subintervals, we can represent the integral as the sum of integrals over each subinterval, which we further approximate using quartic splines. Splines are piecewise polynomial functions, meaning they consist of multiple polynomial segments joined smoothly at specific points called knots. This construction allows them to capture intricate curve shapes that single polynomials might struggle to represent accurately. Quartic splines, in particular, use polynomials of degree four, providing a balance between flexibility and computational efficiency. They can model a wide variety of function behaviors with relatively few knots. A quartic spline is a type of piecewise polynomial function that is used for interpolation or approximation of data points over an interval. Specifically, it is defined on an interval [a, b] and is characterized by the following properties: Continuity: A quartic spline is continuous at each knot, which are the distinct points in the interval [a, b]. This means that the function and its derivatives up to the fourth order are continuous across the entire interval. Piecewise Quartic Polynomials: Within each subinterval determined by the knots, the quartic spline is represented by a quartic (degree-4) polynomial. The use of piecewise quartic polynomials allows the spline to flexibly capture and approximate the behavior of the underlying data within each interval. Interpolatory Conditions: The quartic spline satisfies interpolatory conditions, meaning that it accurately passes through specified data points. In other words, the function coincides with given data values at the knots, ensuring that the spline interpolates the data over the entire interval. Uniqueness: In certain conditions, there exists a unique quartic spline that fulfills the continuity, piecewise quartic polynomial, and interpolatory conditions. This uniqueness ensures that the spline provides a distinct representation of the data within the specified interval. In summary, a quartic spline is a smooth and continuous piecewise polynomial function of degree 4, designed to interpolate or approximate data points over an interval [a, b].

Quartic splines use polynomials of degree four, providing a balance between flexibility and computational efficiency. They can model a wide variety of function behaviors with relatively few knots. Knot Placement: We strategically place knots, points demarcating the subintervals, to influence the spline's shape and accuracy. Different strategies, such as equispaced, Chebyshev, or adaptive knot placement, exist, each with its own strengths and weaknesses. Within each subinterval, we construct a quartic spline that interpolates or passes through the function values at specific points within the interval. These points, typically chosen as function extrema or points where the integral is accurately known, further enhance the approximation. We integrate the constructed quartic splines over their respective subintervals and express the overall integral as a weighted sum of these sub-integrals. The weights depend on the chosen knots and spline construction method.

Partitioning the Interval: We divide the interval of integration [a, b] into n subintervals using knots xi: $a = x_0 < x_1 <\ldots\ldots< x_{n-1} < x_n = b$. Constructing Quartic Splines: Within each subinterval $[x_i, x_{i+1}]$, we construct a quartic spline $S_i(x)$ that interpolates or approximates the function $f(x)$ at specific points within that subinterval. We approximate the integral of $f(x)$ over each subinterval using the integral of the corresponding quartic spline: $\int_{x_i}^{x_{i+1}} f(x)dx \approx \int_{x_i}^{x_{i+1}} S_i(x)dx$. The overall integral is then approximated as the sum of these sub-integrals: $\int_a^b f(x)dx \approx \sum_{i=0}^{n-1} \int_{x_i}^{x_{i+1}} S_i(x)dx$.

Quartic Splines can closely fit diverse function behaviors, leading to more accurate approximations than lower-degree polynomials. They ensure continuity and smoothness of the approximation, even at knot points, making them suitable for applications requiring continuous derivatives. The number of knots and their placement can be adjusted to refine the approximation in critical regions, offering flexibility for different problem complexities.

The integro cubic spline methods over a uniform partition were studied in [5,23], but their error orders are lower. Later, an integro quintic spline approach over a uniform partition was discussed in [6].

In The present study focus is on creating spline functions that interpolate a given function's first derivatives at specific points (knots) and match the second derivatives between those knots. The derivation of these quartic splines in a standard form. An interesting approach is taken, Instead of solving for the spline function values at the knots, the second derivatives are chosen to coincide with the initial knots' values. References [12] and [22] likely provide more information on this technique. A method for numerically integrating over curved two-dimensional domains. The theoretical foundation relies on several concepts: Green's theorem for boundary integration [2] (likely referring to a specific reference on vector calculus), Gauss-Legendre quadrature rule, a popular technique for numerical integration [1], Parametric relations describing the boundary curve (either exact form or obtained through spline interpolation polynomials) - references [3], [4], [11], [14], [15], [17], [19], might be relevant for these methods. The application of the method on a curved domain shaped like a lunar model. The integration of challenging functions over this curved domain is performed using both explicit parametric relations and parametric quartic spline interpolating polynomials. This highlights the advantage of splines when only coordinate data is available along the boundary. Another application of the numerical integration scheme. Here, it's used to compute the geometric moments of a general ellipse. Overall, this chapter presents a valuable approach for numerical integration over complex, curved boundaries. By leveraging quartic splines, the method can handle scenarios where only coordinate data is available.

Quartic spline interpolant

Definition:

Let f be a function defined on the interval [a, b] and let the knots $\{x_i\}_{i=0}^{N+1}$

$$a = x_0 < x_1 < x_2 < .... < x_N < x_{N+1} = b$$

be the $(N+2)$ distinct points. Note that x_i's divide [*a*, *b*] into (N+1) subintervals. A function S is said to be a quartic spline on the interval [*a*, *b*], is S, S', S'' and S''' are continuous in [*a*, *b*] and S is a polynomial of degree ≤ 4 in each knot interval $[x_{i-1}, x_i]$. We shall now present the quasi – Hermite lacunary quartic spline which interpolates the first derivatives of a given function at the knots and the second derivatives between them as proposed in [12, 22].

Theorem:

Given the real numbers $f'(x_i) = f_i'(i = 0,1,2,..,N+1)$, $(x_i + \lambda h) = f''_{i+\lambda}(i = 0,1,2,..,N)$, $f(x_0) = f_0$, and $f_{N+1} = f(x_{N+1})$, then there exists a unique quartic spline *S*(*x*) such that

(1) On each subinterval $[x_i, x_{i+1}]$, $i = 0,1,2,..,N$, *S*(*x*) coincides with the quartic polynomial

$$S(x) = S_i(u) = a_i + b_i u + c_i u^2 + d_i u^3 + e_i u^4 \ , \ u = (x - x_i)/h, \ h = (x_{i+1} - x_i) \ 0 \le u \le 1 \quad (1)$$

(2) S, S' and S'' satisfy the interpolatory conditions.

$$\begin{aligned} &S_i' = f_i', (i = 0,1,2,..,N+1) \\ &S''_{i+\lambda} = f''_{i+\lambda}, (i = 0,1,2,..,N) \\ &S_0 = f_0, S_{N+1} = f_{N+1} \end{aligned} \qquad (2)$$

Now, we shall prove the above theorem. The proof is constructive: it gives on a algorithm to compute the spline.

Proof: We consider the quartic spline $S_i(x)$ over the interval $[x_i, x_{i+1}]$ defined in eqn (1). On satisfying the interpolatory conditions of eqn (2), we obtained

$$\begin{aligned} &S(x_i) = S_i(0) = a_i = f_i \\ &S(x_{i+1}) = S_i(1) = a_i + b_i + c_i + d_i + e_i = f_{i+1} \\ &S'(x_i) = S_i'(0) = b_i / h = f_i' \\ &S'(x_{i+1}) = S_i'(1) = b_i / h + 2c_i / h + 3d_i / h + 4e_i / h = f'_{i+1} \\ &S''(x_i + \lambda h) = (2c_i + 6\lambda d_i + 12\lambda^2 e_i) / h^2 = f''_{i+\lambda} \end{aligned} \qquad (3)$$

Solving the above linear equations in eqn (3), we obtain

$$\begin{aligned} &a_i = f_i, \quad b_i = hf_i', \\ &|A| c_i = f_i(-36\lambda^2 + 24\lambda) + f_{i+1}(36\lambda^2 - 24\lambda) \\ &\qquad + hf_i'(-24\lambda^2 + 24\lambda - 6\lambda) + hf'_{i+1}(-12\lambda^2 + 6\lambda) + h^2 f''_{i+\lambda} \\ &|A| d_i = f_i(24\lambda^2 - 8) + f_{i+1}(-24\lambda^2 + 8) + hf_i'(12\lambda^2 - 6) + hf_{i+1}'(12\lambda^2 - 2) - 2h^2 f''_{i+\lambda} \\ &|A| e_i = f_i(-12\lambda + 6) + f_{i+1}(12\lambda - 6) + hf_i'(-6\lambda + 4) + hf'_{i+1}(-6\lambda + 2) + h^2 f''_{i+\lambda}, \\ &|A| = 2(6\lambda^2 - 6\lambda + 1) \end{aligned} \qquad (4)$$

On substituting for a_i, b_i, c_i, d_i and e_i in eqn (1), we obtain the following alternative expression

$$S_i(u) = f_i\left[(1-u)^2(1+2u) - \frac{3(2\lambda-1)}{\Delta}u^2(1-u^2)\right] + f_{i+1}\left[u^2(3-2u) + \frac{3(2\lambda-1)}{\Delta}u^2(1-u)^2\right]$$
$$+hf_i'\left[u(1-u)^2 - \frac{(3\lambda-2)}{\Delta}u^2(1-u)^2\right] - hf_{i+1}'\left[(1-u)u^2 + \frac{(3\lambda-1)}{\Delta}u^2(1-u)^2\right] + h^2 f_{i+\lambda}''\left[\frac{u^2(1-u)^2}{2\Delta}\right]$$

(5)

where,
$$\begin{aligned} &u = (x-x_i)/h,\ h = (x_{i+1}-x_i),\ \forall_i \\ &\Delta = (6\lambda^2 - 6\lambda + 1) \end{aligned} \tag{6}$$

It is clear that $S_i(u)$ is not defined for $\Delta = 0$, i.e., when $\lambda = (3 \pm \sqrt{3})/6$. Now using the continuity of the second derivatives over the intervals [x_{i-1}, x_i] and [x_i, x_{i+1}], we obtain the following relation as in [22].

$$\begin{aligned} &-(\lambda-1)(3\lambda-1)f_{i-1} + (1-2\lambda)f_i + \lambda(3\lambda-2)f_{i+1} \\ &= \frac{h}{2}\left[(\lambda-1)(2\lambda-1)f_{i-1}' + (8\lambda^2 - 8\lambda + 1)f_i' + \lambda(2\lambda-1)f_{i+1}'\right] \\ &+h^2\left[f_{i-1+\lambda}'' - f_{i+\lambda}''\right],\quad i = 1,2,..,N \end{aligned} \tag{7}$$

The above system of N linear equations will determine f_1, f_2,..,f_N, since f_0 and f_{N+1}, $\left\{f_i'\right\}_{i=0}^{N+1}$ and $\left\{f_{i+\lambda}''\right\}_{i=0}^{N}$ are already known. We see that $\lambda \neq (3 \pm \sqrt{3})/\sqrt{6}$ and for some simple values of $\lambda = 0, 1/3.2/3, 1$, the coefficient matrix in the linear system of eqn (7) is nonsingular. We shall now consider the simple case λ=0 only.

Case λ=0

The following explicit expression over the interval $[x_i, x_{i+1}]$ represent the quartic spline interpolant,

$$S_i(u) = a_i + b_i u + c_i u^2 + d_i u^3 + e_i u^4, \ldots. \tag{8}$$

where
$$\begin{aligned} &u = (x-x_i)/h,\ h = (x_{i+1}-x_i),\ \forall_i \\ &a = f_i,\ b_i = hf_i',\ c_i = h^2 f_i'', \\ &d_i = -4f_i + 4f_{i+1} - 3hf_i' - hf_{i+1}' - h^2 f_i'', \\ &e_i = 3f_i - 3f_{i+1} + 2hf_i' + hf_{i+1}' + h^2 f_i''/2 \end{aligned} \tag{9}$$

and, $\left\{f_i\right\}_{i=1}^{N}$ can be evaluated by the below said recurrence relation for the quartic spline from eqn(5.2.7),

$$f_i = f_{i-1} + (h/2)(f_{i-1}' + f_{i+1}') + (h^2/12)(f_{i-1}'' + f_i'')\ (i = 1,2,\ldots,N) \tag{10}$$

by forward recurrence computing.

Numerical integration over curved domains in 2-space

Consider the integral [44, 3] $II \overset{def}{=} \iint\limits_{\pi_{xy}} f(x,y)dx\,dy$ (11)

where, π_{xy} is the curved domain.

Let's denote the closed boundary of the domain π_{xy}, by $\partial\pi_{xy}$. We discretise this boundary $\partial\pi_{xy}$ into N curved arcs by points $((x_k, y_k), k = 1, 2, .., N+1)$. Let the *i-th* arc be joined by the vertices $V_i = (x_i, y_i)$ and $V_{i+1} = (x_{i+1}, y_{i+1})$ and denoted as $\partial\pi_{i,i+1}$. Thus, we can now represent the closed boundary $\partial\pi_{xy}$ as a sum of N curved arcs:

$$\partial\pi_{xy} = \sum_{i=1}^{N} \partial\pi_{i,i+1} \tag{12}$$

Here, each $\partial\pi_{i,i+1}$ signifies a specific curved arc bounded by the vertices $V_i = (x_i, y_i)$ and $V_{i+1} = (x_{i+1}, y_{i+1})$. Hence, by the use of the boundary integration formula of Green's theorem for the closed boundary $\partial\pi_{xy}$ and the relation of eqn (12), we obtain:

$$\text{II} = \iint\limits_{\pi_{xy}} f(x,y)dx\,dy = \sum_{i=1}^{N} \oint\limits_{\partial\pi_{i,i+1}} \Phi(x,y)dy \tag{13}$$

where $$\Phi(x,y) = \int_{\alpha}^{x} f(u,y)du \tag{14}$$

in which α is fixed, $\alpha \neq x_i$ and arbitrary.

Let us now choose the replacement $u = \left(\frac{x-\alpha}{2}\right)s + \left(\frac{x+\alpha}{2}\right)$ (15)

using for *u* from eqn (15) into eqn (14), we have

$$\Phi(x,y) = \left(\frac{x-\alpha}{2}\right)\int_{-1}^{1} f\left(\left(\frac{x-\alpha}{2}\right)s + \left(\frac{x+\alpha}{2}\right), y\right)ds \tag{16}$$

Further by using $\Phi(x,y)$ from eqn (16) into eqn (13), we obtain

$$\begin{aligned} \text{II} &= \sum_{i=1}^{N} \oint\limits_{\partial\pi_{i,i+1}} \Phi(x,y)dy \\ &= \sum_{i=1}^{N} \oint\limits_{\partial\pi_{i,i+1}} \left(\frac{\hat{x}_i(t)-\alpha}{2}\right)\left[\int_{-1}^{1} f\left(\left(\frac{\hat{x}_i(t)-\alpha}{2}\right)s + \left(\frac{\hat{x}_i(t)+\alpha}{2}\right), \hat{y}_i(t)\right)ds\right]dy \qquad (17) \\ &= \sum_{i=1}^{N} \int_{y_i}^{y_{i+1}} \left(\frac{\hat{x}_i(t)-\alpha}{2}\right)\left[\int_{-1}^{1} f\left(\left(\frac{\hat{x}_i(t)-\alpha}{2}\right)s + \left(\frac{\hat{x}_i(t)+\alpha}{2}\right), \hat{y}_i(t)\right)ds\right]y_i'(t)dt \end{aligned}$$

where, $\hat{x}_i(t)$ and $\hat{y}_i(t)$ refer to the appropriate parametric form over the curved boundary arc joining the points (x_i, y_i) and (x_{i+1}, y_{i+1})

Now letting $$t=\left(\frac{y_{i+1}-y_i}{2}\right)T+\left(\frac{y_{i+1}+y_i}{2}\right) \tag{18}$$

and substituting in equation (17), we finally obtain :

$$I=\sum_{i=1}^{N}\int_{-1}^{1}\left(\frac{X_i(T)-\alpha}{2}\right)\left[\int_{-1}^{1} f\left(\left(\frac{X_i(T)-\alpha}{2}\right)s+\left(\frac{X_i(T)+\alpha}{2}\right),Y_i(T)ds\right)\right]\left(\frac{y_{i+1}-y_i}{2}\right)Y_i'(T)dT \tag{19}$$

where
$$X_i(T)=\hat{x}_i\left(\left(\frac{y_{i+1}-y_i}{2}\right)T+\left(\frac{y_{i+1}+y_i}{2}\right)\right),$$
$$Y_i(T)=\hat{y}_i\left(\left(\frac{y_{i+1}-y_i}{2}\right)T+\left(\frac{y_{i+1}+y_i}{2}\right)\right) \tag{20}$$

We shall now apply the Gauss Legendre Quadrature rules of order n and m along the directions s and T respectively. This gives

$$I=\sum_{i=1}^{N}\sum_{j=1}^{n}\sum_{k=1}^{m}W_j^n W_k^m\left(\frac{X_i(T_k^m)-\alpha}{2}\right)\left(\frac{y_{i+1}-y_i}{2}\right)Y_i'(T_k^m)f\left(\left(\frac{X_i(T_k^m)-\alpha}{2}\right)s_j^n+\left(\frac{X_i(T_k^m)+\alpha}{2}\right),Y_i(T_k^m)\right) \tag{21}$$

From the above formula, it is quite obvious that $m>n$, for a bivariate polynomial function of degree less than or equal to (2n-1). The parametric form of $\left(X_i(T),Y_i(T)\right)$ must be of degree greater than or equal to two for a curved boundary, Hence in equation (21), we must have $m\cong(n+p)$ for the curved boundary, as p–point Gauss Legendre quadrature rule is necessary to integrate the following product in equation (21).

$$\left(\left(X_i(T_k^m)-\alpha\right)/2\right)Y_i'(T_k^m)$$

a polynomial of degree, say, (2p-1). On the other hand, we have chosen n–th order Gauss Legendre Quadrature rule to integrate the bivariate function $f\left(\left(X_i(T_k^m)-\alpha\right)/2\right)s_1^{'}+\left(X_i(T_k^{'})+\alpha\right)/2, Y_i(T_k^m)\right)$ as polynomial degree (2n-1) in s in equation (21).

Application example: a lunar model

We shall now consider the curved domain in the shape of a lunar model whose boundary is composed of two circular arcs. The outer circular arc satisfies the equation: $\left(x-1/2\right)^2+\left(y-1/2\right)^2=1/4$ and the inner circular arc satisfies the equation $x^2+y^2=1/4$. This is shown in figure: Fig 1.

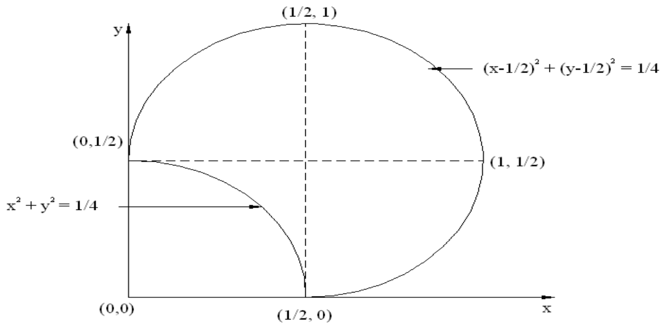

Figure 1: Lunar Model

The numerical scheme developed in the previous section relies on having suitable parametric equations that describe the curved boundary of the domain π_{xy}, as specified by equations (17) to (21). In some scenarios, the boundary of the curved domain can be precisely defined by mathematical relations. In such cases, it's feasible to transform these mathematical relations into explicit parametric equations. However, there are situations where the boundary curve is described by a set of discrete coordinate points, making it necessary to generate parametric equations through alternative methods. One such method involves utilizing spline interpolating polynomials of varying orders (e.g., 2nd, 3rd, 4th order, etc.) to generate parametric equations for the curved boundary. This approach proves to be advantageous even when dealing with a boundary curve represented by discrete coordinate points. Now, let's delve into the application of these two methods in the context of a lunar model, as discussed previously.

Exact Mathematical Relations Method: If the boundary of the lunar model can be precisely defined by mathematical relations, we can derive explicit parametric equations directly from these relations. This method is straightforward and offers accurate representations of the boundary. However, it relies on the availability of exact mathematical descriptions of the boundary.

Spline Interpolating Polynomials Method: In cases where the boundary of the lunar model is not described by exact mathematical relations but rather by a set of discrete coordinate points, spline interpolating polynomials can be employed to generate parametric equations. Spline interpolation allows us to construct a smooth curve that passes through these discrete points, providing a continuous representation of the boundary. The choice of spline order (e.g., 2nd, 3rd, 4th order) depends on the desired level of smoothness and accuracy required for the boundary representation.

By applying these two methods, we can effectively obtain parametric equations for the curved boundary of the lunar model, enabling the implementation of the numerical scheme discussed earlier. This ensures accurate simulation and analysis of the lunar environment for the intended application.

Explicit form of Parametric Equations

The following parametric relations can be immediately obtained for the lunar model describe above.

(1) Outer circular arc

The outer circular arc is the boundary curve over the fourth first and second quadrant and its boundary is described by the equation $(x-1/2)^2+(y-1/2)^2=1/4$. Hence the parametric equations.

$$x=\frac{1}{2}+\left(\frac{1}{2}\right)\cos\theta,\ y=\frac{1}{2}+\left(\frac{1}{2}\right)\sin\theta$$
$$\theta\in\left(\left[0,\pi/2\right],\left[\pi/2,\pi\right],\left[3\pi/2,2\pi\right]\right) \tag{22}$$

(2) Inner circular arc

The inner circular arc is the boundary curve over the third quadrant and its boundary is described by the equation $x^2+y^2=1/4$ and hence the parametric equations are

$$x=\frac{1}{2}\cos\theta,\ y=\frac{1}{2}\sin\theta,(\pi/2\le\theta\le 0) \tag{23}$$

Quartic spline interpolants as parametric equations

The parametric equations described above can be represented by two integral functions. $F(\theta)$ and $G(\theta)$ and the associated boundary conditions on $F(\theta)$ and $G(\theta)$ over each curved boundaries in the four quadrants (see fig 1). We have used $F(\theta)$ and $G(\theta)$ to generate the parametric equations for ($x(\theta)$, $y(\theta)$). We also impose the conditions $x^{(p)}\left(\theta\right)=F^{(p)}\left(\theta\right),\ y^{(p)}\left(\theta\right)=G^{(p)}\left(\theta\right),\ p=1,2$. We shall now explain the method of finding $F(\theta)$, $G(\theta)$ and their relations with $x(\theta)$, $y(\theta)$ the parametric equations. Once $x(\theta)$ and $y(\theta)$ are found in terms of the quartic spline integral representations as in [22] for $F(\theta)$ and $G(\theta)$, we can implement the numerical scheme described in eqns (11) – (21). We now present the integral function representations for the curved arcs in each of the four quadrants.

(i) Over the first quadrant $\theta\in[0,\pi/2]$, we define :

$$F\left(\theta\right)=\int_0^\theta d\left(\frac{\cos t}{2}\right),\ G\left(\theta\right)=\int_0^\theta d\left(\frac{\sin t}{2}\right)$$
$$F(0)=0,\ F(\pi/2)=-1/2,\ G\left(\theta\right)=0,\ G(\pi/2)$$

so that we have $x(\theta)=F(\theta)+1$ and $y(\theta)=G(\theta)+1/2$

The quartic splines for ($x(\theta)$, $y(\theta)$) over the first quadrant, $\theta\in\left[0,\pi/2\right]$ can be then computed by the MATLAB function : quartic_spline (*M, a, b, N*) in which *M* refers to function type, *a, b* refer to end points $a = 0$, $b = \pi/2$ for the present case and (N+1) refers to the number of subdivisions made in [*a*, *b*]. Similar reasoning is applicable to the curved arcs of the remaining quadrants viz second third and fourth.

(ii) Over the second quadrant, $\theta\in[\pi/2,\pi]$,we define :

$$F(\theta)=\int_{\pi/2}^{\theta} d\left(\frac{\cos t}{2}\right), G(\theta)=\int_{\pi/2}^{\theta} d\left(\frac{\sin t}{2}\right)$$

$$F(\pi/2)=0, F(\pi)=-1/2, G(\pi/2)=0, G(\pi)=-1/2$$

so that $x(\theta)=F(\theta),\ y(\theta)=G(\theta)+1/2$.

(iii) Over the third quadrant, $\theta \in [\pi/2,0]$ and we define :

$$F(\theta)=\int_{\pi/2}^{\theta} d\left(\frac{\cos t}{2}\right), G(\theta)=\int_{\pi/2}^{\theta} d\left(\frac{\sin t}{2}\right)$$

$$F(\pi/2)=0, F(0)=1/2, G(\pi/2)=0, G(0)=-1/2$$

so that we have $x(\theta)=F(\theta),\ y(\theta)=G(\theta)+1/2$

Here, we may note that it is required to generate a circular arc in first quadrant of the circle $x^2+y^2=1/4$.

(iv) Over the fourth quadrant, $\theta \in [3\pi/2, 2\pi]$ and we define :

$$F(\theta)=\int_{3\pi/2}^{\theta} d\left(\frac{\cos t}{2}\right), G(\theta)=\int_{3\pi/2}^{\theta} d\left(\frac{1+\sin t}{2}\right)$$

$$F(3\pi/2)=0, F(2\pi)=0 \text{ and } G(3\pi/2)=0, G(2\pi)=1/2$$

so that $x(\theta)=F(\theta)+1/2,\ y(\theta)=G(\theta)$

It is important to note here that as shown in Fig1, the outer circular arcs covers the first, second and fourth quadrants of the circle. $(x-1/2)^2+(y-1/2)^2=1/4$

Examples

We Consider the Following Integrals:

$$II_i=\iint_{\pi_{xy}} f_i(x,y)\ dxdy, \quad (i=1,2,3,4,5,6,7) \tag{24}$$

The values if the integrals II_i $(i=1,2,3,4,5,6,7)$ with a relative error around 10^{-14}, are

$II_1 = I_{\Omega_0}(f_1) = 0.20307626985342$

$II_2 = I_{\Omega_0}(f_2) = 0.20646770293563$

$II_3 = I_{\Omega_0}(f_3) = 638.55743274702$

$II_4 = I_{\Omega_0}(f_4) = 0.57263720432530$

$II_5 = I_{\Omega_0}(f_5) = 0.03137185199242$

$II_6 = I_{\Omega_0}(f_6) = 0.0062895812195655$

$II_7 = I_{\Omega_0}(f_7) = 0.6426990816987241$

where π_{xy} is the curved domain described in fig 1. The integrands $f_i(x, y)$ are the following [18].

$$\begin{aligned} f_1(x,y) = {} & \frac{3}{4}\exp\left\{-\frac{1}{4}(9x-2)^2+(9y-2)^2\right\}+\frac{3}{4}\exp\left\{-\frac{1}{49}(9x+1)^2-\frac{1}{10}(9y+1)\right\} \\ & +\frac{1}{2}\exp\left\{-\frac{1}{4}(9x-7)^2+(9y-3)^2\right\}-\frac{1}{5}\exp\left\{-((9y-4)^2+(9y-7)^2\right\} \end{aligned} \tag{25}$$

$$f_2(x,y) = \sqrt{(x-0.5)^2+(y-0.5)^2} \tag{26}$$

$$f_3(x,y) = (x+y)^{19} \tag{27}$$

$$f_4(x,y) = \exp\left\{-((x-0.5)^2+(y-0.5)^2)\right\} \tag{28}$$

$$f_5(x,y) = \exp\left\{-100((x-0.5)^2+(y-0.5)^2)\right\} \tag{29}$$

$$f_6(x,y) = \cos(20(x+y)) \tag{30}$$

$$f_7(x,y) = 1 \tag{31}$$

We have implemented the numerical scheme outlined in the above by leveraging the explicit parametric equations to compute the integrals specified by Equation (24). To facilitate this computation, we've developed two MATLAB programs. These programs utilize the precise representation of the boundary curve of the lunar model provided by the parametric equations elucidated in Equations (24). Notably, even with as few as four points strategically placed along the curved boundary of the lunar model, employing Gauss-Legendre quadrature rules of order (32, 36) or (32, 40) yields near-exact results.

A glimpse of the output generated is showcased in Table-1. Moving forward, we address scenarios where the boundary curve of the lunar model is characterized by a set of discrete coordinate points. In such instances, we employ this discrete information to derive parametric equations in the form of spline interpolating polynomials. Specifically, we explore the application of quartic splines, a topic thoroughly discussed in this chapter.

By transitioning to spline interpolating polynomials, we ensure a continuous and smooth representation of the boundary curve, even when described by discrete points. This strategy enables us to maintain the accuracy and reliability of our numerical computations, crucial for robust analysis and simulation of the lunar environment.

Number of Points(n_p)	I1	I2	I3	I4	I5	I6	I7
1	637.8629706	0.206422298	0.572584929	0.031372008	0.210500552	0.642625025	0.006318813
2	638.39271	0.2064559	0.5726234	0.0313718	0.2105027	0.6426797	0.0062962

	55	95	33	82	43	04	11
3	638.5016688	0.206463556	0.572632265	0.031371862	0.210503386	0.64269216	0.006291569
4	638.5338184	0.206465895	0.572635033	0.031371856	0.210503614	0.642696048	0.006290388
5	638.5458203	0.206466796	0.572636109	0.031371854	0.210503709	0.642697554	0.006289965
6	638.5510842	0.2064672	0.572636594	0.031371853	0.210503754	0.642698232	0.006289786
7	638.5536778	0.206467402	0.572636838	0.031371853	0.210503778	0.642698572	0.006289701
8	638.5550729	0.206467512	0.572636972	0.031371852	0.210503791	0.642698758	0.006289655
9	638.5558765	0.206467576	0.572637049	0.031371852	0.210503798	0.642698867	0.00628963
10	638.5563655	0.206467616	0.572637097	0.031371852	0.210503803	0.642698933	0.006289614
100	638.5574326	0.206467703	0.572637204	0.031371852	0.210503815	0.642699082	0.006289581
200	638.5574327	0.206467703	0.572637204	0.031371852	0.210503815	0.642699082	0.006289581
300	638.5574327	0.206467703	0.572637204	0.031371852	0.210503815	0.642699082	0.006289581
400	638.5574327	0.206467703	0.572637204	0.031371852	0.210503815	0.642699082	0.006289581
500	638.5574327	0.206467703	0.572637204	0.031371852	0.210503815	0.642699082	0.006289581
600	638.5574327	0.206467703	0.572637204	0.031371852	0.210503815	0.642699082	0.006289581
700	638.5574327	0.206467703	0.572637204	0.031371852	0.210503815	0.642699082	0.006289581
800	638.5574327	0.206467703	0.572637204	0.031371852	0.210503815	0.642699082	0.006289581
900	638.5574327	0.206467703	0.572637204	0.031371852	0.210503815	0.642699082	0.006289581
1000	638.5574327	0.206467703	0.572637204	0.031371852	0.210503815	0.642699082	0.006289581

Table - 1 Spline Integration for $II(i) = \iint_D f_i(x,y)\ dxdy, i = 1(1)7$, D: Domain of Lunar model

I1 integral seems to converge to approximately 638.5574327 as the number of points Np increases. This suggests that I1 has reached a stable value, indicating convergence of the numerical method used to compute it. I2: Similarly, converges to a stable value of approximately 0.206467703 as Np increases. The convergence of I2 indicates that the numerical method accurately computes this integral. I3, I4, I5, I6, I7: These integrals also converge to stable values as Np increases, with each approaching approximately 0.572637204, 0.031371852, 0.210503815, 0.642699082, 0.572637204, 0.031371852, 0.210503815, 0.642699082, and 0.006289581 respectively. The convergence of all these integrals suggests the effectiveness and accuracy of the numerical method across different aspects of the problem. Overall, the trend observed in the table indicates that increasing the number of points Np in the numerical method leads to improved accuracy and convergence of the computed integrals. This convergence behavior is consistent with expectations from numerical methods, where increasing the resolution typically leads to more accurate results.

Application example 2: general ellipse:

Consider an ellipse in (x, y) plane with semi major axis of length a and semi minor axis of length b. The ellipse is then rotated anticlockwise by an angle of **θ** degrees and translated to coincide the centre of the ellipse to (x_0, y_0).

The equation of the ellipse in the (x, y) plane is $\frac{x^2}{a^2}+\frac{y^2}{b^2}=1$.

Let the new coordinate system with origin at (x_0, y_0) be denoted by (x', y') plane. The ellipse in (x', y') plane is then rotated by an angle **θ** in anticlockwise direction. Let us denote the points on the rotated ellipse with respect to (x', y') coordinate system as (x^*, y^*).

(i). The points on the ellipse $\frac{x'^2}{a^2}+\frac{y'^2}{b^2}=1$ are $x' = a\cos t, y' = b\sin t, 0 \le t \le 2\pi$

(ii). The points (x^*, y^*) corresponding to the point (x', y') on the rotated ellipse are

$$x^* = x'\cos\theta - y'\sin\theta, y^* = x'\sin\theta + y'\cos\theta$$

(iii). Points (x, y) corresponding to (x^*, y^*) must satisfy

$$\begin{aligned} &x' = x^*, y' = y^* \\ i.e,\ &x' = x - x_0 = x^*, y' = y - y_0 = y^* \\ \therefore\ &x = x_0 + x^*, y = y_0 + y^* \end{aligned}$$

We shall now determine the points on the rotated ellipse. Let the points on the ellipse with respect to (x', y') should be {(a, 0), (0, b), (- a, 0), (0, - b)}. These points with respect to (x, y) system are { $(x_0 + a\cos\theta, y_0 + b\sin\theta), (x_0 - b\sin\theta, y_0 + b\cos\theta)$, $(x_0 - a\cos\theta, y_0 - a\sin\theta)$, $(x_0 + b\sin\theta, y_0 - b\cos\theta)$ }. The remaining points on the ellipse can be determined in a similar manner. The geometric moments over the general ellipse described above is given by,

$$m_{pq} = \iint_A x^p y^q dxdy \qquad (32)$$

where A is the area of the ellipse.

The compound moment values and the exact moment values are tabulated. We may note that the δ -method, the rectangular grid method, trapezoidal integration method and contour integration method proposed also tabulated for the purpose of integration.

Number of points on general ellipse n_p	**Moments**	**Computed Values of Moments**	**Theoretical Values** **1.0e+006 ***
498	m00 = 1.884955592153876e+003 m10 = 1.884955592153876e+004 m01 = 5.654866776461627e+004 m20 = 4.948008429403924e+005 m02 = 2.002765316663493e+006 m11 = 6.832964021557800e+005	**fn=1** 1.884955592022257e+003 fn=x 1.884955592161867e+004 fn=y 5.654866776206377e+004 fn=x^2 4.948008429123767e+005 fn=y^2 2.002765316586024e+006 fn=x*y 6.832964021556849e+005	0.00188495559202 0.01884955592162 0.05654866776206 0.49480084291238 2.00276531658602 0.68329640215568
998	m00 = 1.884955592153876e+003 m10 = 1.884955592153876e+004 m01 = 5.654866776461627e+004 m20 = 4.948008429403924e+005 m02 = 2.002765316663493e+006 m11 = 6.832964021557800e+005	fn=1 1.884955592145685e+003 fn=x 1.884955592154394e+004 fn=y 5.654866776445747e+004 fn=x^2 4.948008429386528e+005 fn=y^2 2.002765316658672e+006 fn=x*y 6.832964021557791e+005	1.884955592145685e+003 1.884955592154394e+004 5.654866776445747e+004 4.948008429386528e+005 2.002765316658672e+006 6.832964021557791e+005
1498	m00 = 1.884955592153876e+003 m10 =	fn=1 1.884955592152255e+003 fn=x	1.884955592152255e+003 1.884955592153955e+004 5.654866776458480e+004

	1.884955592153876e+004 m01 = 5.654866776461627e+004 m20 = 4.948008429403924e+005 m02 = 2.002765316663493e+006 m11 = 6.832964021557800e+005	1.884955592153955e+004 fn=y 5.654866776458480e+004 fn=x^2 4.948008429400452e+005 fn=y^2 2.002765316662538e+006 fn=x*y 6.832964021557742e+005	4.948008429400452e+005 2.002765316662538e+006 6.832964021557742e+005
1998	m00 = 1.884955592153876e+003 m10 = 1.884955592153876e+004 m01 = 5.654866776461627e+004 m20 = 4.948008429403924e+005 m02 = 2.002765316663493e+006 m11 = 6.832964021557800e+005	fn=1 1.884955592153362e+003 fn=x 1.884955592153892e+004 fn=y 5.654866776460630e+004 fn=x^2 4.948008429402803e+005 fn=y^2 2.002765316663193e+006 fn=x*y 6.832964021557753e+005	1.884955592153362e+003 1.884955592153892e+004 5.654866776460630e+004 4.948008429402803e+005 2.002765316663193e+006 6.832964021557753e+005

Table - 2 Computed Values of Moments

Conclusions

we have presented quartic spline, which interpolates the first derivatives of a given function at the knots and the second derivatives between them. A simple approximation to function values is determined by choosing the second derivate to coincide with the initial knots. A numerical integration scheme Gauss Legendre quadrature rules and the explicit parametric relations along the curved boundary of the two dimensional Cartesian space is considered. The above numerical integration scheme is illustrated by computing several difficult integrals over the curved domain. The integral values with its absolute error of integrals are tabulated. Integration for $II(i) = \iint_D f_i(x,y)\ dxdy, i = 1(1)4$, D: Domain of as shown in figure 1

Acknowledgment: Both the authors acknowledge Late Dr. H T Rathod for his valuable suggestions, the support and guidance.

References:

[1] Anwar, M.N and El-Tarazi, M.N: Direct cubic spline with application to quadrature, communications in applied numerical methods, 5, pp 237 – 246 (1989).

[2] Apostol T. M., Calculus, Vol. II, Second Edition Blaisdell (1969).

[3] Elden, L., LeWittmeyer-Koch, H. B. Nielson Introduction to Numerical computation, Analysis and MATLABR illustrations, Overseas Press India Pvt, Ltd (2006).

[4] Gustafasson, F. and Bergman, N., MATLAB for engineers explained, Springer International Edition (2003).

[5] H. Behforooz, Approximation by integro cubic splines, Appl. Math. Comput. 175 (2006) 8–15.

[6] H. Behforooz, Interpolation by integro quintic splines, Appl. Math. Comput. 216 (2010) 364–367

[7] H. T. Rathod and H. S. Govinda Rao, Integration of polynomials over n-dimensional linear polyhedra, Computers and Structures, Vol.65, No.6, pp.829-847 (1997).

[8] H. T. Rathod and S. V. Hiremath, Boundary Integration of polynomials over an arbitrary linear hexahedron in Euclidean three dimensional space, Computer Methods in Applied Mechanics and Engineering, Vol.161, pp.155-193 (1998).

[9] H. T. Rathod, Md.Shafiqul.Islam, Bharath Rathod , K. Sugantha Devi , Finite element solution of Poisson Equation over Polygonal Domains using a novel auto mesh generation technique and an explicit integration scheme for linear convex quadrilaterals of cubic order Serendipity and Lagrange families, International Journal Of Engineering And Computer Science ISSN:2319-7242,Volume 7 Issue 1 January 2018, Page No. 23329-23482

[10] H.T. Rathod, A. S. Hariprasad, K.V.Vijayakumar, Numerical Integration over Curved Domains Using Convex Quadrangulations And Gauss Legendre Quadrature Rules International Journal Of Engineering And Computer Science, 2(11), 2013, 3290-3332

[11] Hanselman.D and B. Littlefield, Mastering MATLABR, Pearson Education Inc (2007).

[12] John Mathews, Kurtis Fink, Numerical Methods Using Matlab, International Edition, 4th Edition, Pearson, Jan 2004.

[13] Lo. S. H, Generating quadrilateral elements on plane and over curved surfaces, Comput. Stuct. 31(3) 421-426(1989)

[14] MATLAB Programming Version 7, The Maths Works INC, 2004.

[15] MIcula C., Sanda Micula, Hand Book of Splines, Kluwer Academic Publishers (1999).

[16] Moler, C. B., Numerical Computing with MATLAB, Society for Industrial and Applied Mathematics, Phildelphia (2004).

[17] Otto S. R. and J. P. Denier., An Introduction to programming and numerical methods in MATLAB for beginners and experienced users, Cambridge University Press (2000).

[18] Sallam S., W. Ameen Numerical solution of general nth-order differential equations via splines Applied Numerical Mathematics, 6(3), 1990, 225-238.

[19] Shikin E. V., Alexander I. Plis, Handbook on splines for the user, CRC Press, 1995.

[20] Sommariva A. and M. Vianello, Gauss-Green Cubature over spline curvilinear polygons, Applied Mathematics and Computation 183(2), 1098 - 1107 (2006)

[21] Sommariva A. and M. Vianello, Gauss-Green Cubature over spline curvilinear polygons, Applied Mathematics and Computation Vol. 183 (2) pp 1098 - 1107 (2006)

[22] Stroud A. H., Approximate calculation of multiple integrals, prentice Hall series in automated computation, Prentice Hall, Inc Englewood Cliffs N. J., (1971).

[23] T. Zhanlav, R. Mijiddorj, The local integro cubic splines and their approximation properties, Appl. Math. Comput. 216 (2010) 2215–2219.

[24] Zienkiewicz. O. C, Taylor. R. L and Zhu. J. Z, The finite element method, its basis and fundamentals, 6th Edn, Elsevier (2007).