Int. Journal of Math. Analysis, Vol. 4, 2010, no. 28, 1387 - 1415

On a New Cubic Spline Interpolation with Application to Quadrature

H. T. Rathod [a*], H. Y. Shrivalli [b], K. V. Nagaraja[c]

and Kesavulu Naidu[c]

[a] Department of Mathematics, Central College Campus,
Bangalore University, Bangalore - 560001, India
htrathod@yahoo.com

[b] Department of Mathematics, B.M.S college of Engineering
Basavanagudi, Bangalore-560019, India
hys.maths@bmsce.ac.in

[c] Department of Mathematics, Amritha school of Engineering,
#26&27 carmelram post, Kasavanahalli
Bangalore-560035, India
nagarajaitec123@yahoo.com

Abstract

This paper presents a formulation and a study of an interpolatory cubic spline which is new and akin to the Subbotin quadratic spline. This new cubic spline interpolates at the first and last knots and at the two points located at trisections between the knots. Application of the proposed spline to integral function approximations and quadrature over curved domains are investigated. Numerical illustrations, sample outputs and MATLAB programs are appended.

Mathematics Subject Classifications: 41A15, 65D05, 65D32

Keywords: Cubic splines, natural, clamped, not a knot, Subbotin, tridiagonal two band matrices, integral function approximations, moments for general ellipse

1. INTRODUCTION

The fitting of a polynomial curve to a set of data points has applications in CAD (Computer assisted design), CAM (Computer assisted manufacturing) and Computer graphic systems and robot path/trajectory planning. The interpolating methods lead to higher order polynomials, in general of order equal to one less than the number of data points. In addition to being laborious computationally, these high order polynomials does indeed pass through the data points as required, it is quite well known that it does so by oscillating wildly between the given data points, particularly near the ends or middle of the data range. This is a characteristic feature of all high order polynomials, a feature which makes them unattractive for interpolation applications. Spline methods involve interpolation between the given data points by attaching together several low order polynomials. Various continuity requirements can be specified at the data points to impose various degrees of smoothness of the resulting curve. The order of continuity becomes important when a complex curve is modeled by several curve segments pieced together end to end. A cubic polynomial is the minimum order polynomial that guarantees the generation of C^0, C^1 *or* C^2 curves. The best known methods resort to piecewise cubic curve constructed by the individual third degree polynomials assigned to each subinterval, which is called cubic spline interpolation.

In this paper, we present a new cubic spline, where the nodes for interpolation are chosen to be the first and the last knots and the two points at the trisections between the knots. We shall note here that the knots are defined as the points where the spline function is permitted to change in form from one polynomial to another and the nodes are the points where the values of the splines are specified. The new cubic spline presented in this paper is akin to the Subbotin quadratic splin [5-17]. We mention here that in Subbotin quadratic spline the nodes for interpolation are chosen to be the first and last knots and the midpoint between the knots, this gives (n+2) interpolation condition over the n-subinterval (n-midpoints and 2-first and last points), and (n-1) conditions from the continuity of spline functions. However, we note that since the first derivatives are defined at the nodes, the continuity of the first derivatives add no additional conditions. This determines the 3n parameters for the n-quadratic splines over the n-subintervals. In a similar manner, for the proposed new cubic spline there are (2n+2) interpolation conditions over n-subintervals at the interpolation nodes as mentioned earlier and 2(n-1) conditions from the continuity of the spline functions and their first derivatives over n-subintervals. However, as in case of Subbotin quadratic splines, for the new cubic splines, since the values of the second derivatives at the nodes are defined and hence in this instance also continuity adds no additional conditions. Thus we can determine the 4n parameters of the n-spline functions over n-subintervals for the proposed new cubic spline function. In section-2, of this paper, we present the derivation of the proposed new cubic spline. In section-3 of this paper, we present the applications of proposed cubic spline to integral function approximations and to the numerical integration over curved domains in 2-space. Applications to integral function approximations are illustrated for the indefine integral of Runge function, logarithmic function and normal distribution. This application is useful in the construction of table of integrals. As an application of numerical integration over curved domains the computation of moments for a general ellipse is considered [12]. We also compare the function approximations of the proposed new spline with the standard cubic splines (natural, clamped and not a knot) [1–18].

2. THE INTERPOLATION PROBLEM

We shall apply a useful approximation process, first proposed by Subbotin [1967] and latter by Kammerer, Reddien and Varga [10] see the texts [5, 6]. We propose a cubic spline, where the nodes for the interpolation are choosen to be the first and the last and two points between the knots. It is assumed that the knots are defined as the points where the spline function is permitted to change in form from one polynomial to another and the nodes are the points where the values of splines are specified.

2.1 THEOREM :

Let $\{t_i\}_{i=1}^{n+1}$ be a nonuniform partition of the interval $\left[t_{1,}\, t_{n+1}\right]$. Denote by $Q_i\left(x\right)$, the cubic polynomial over the interval $t_i \le x \le t_{i+1}$ *and* $h_i = t_{i+1} - t_i$ the length of the interval. Let us further assume that we are given the real numbers $\{y_i\}_{i=1}^{2n+2}$ such that the cubic spline interpolant $Q\left(x\right)$ over the interval $\left[t_{1,}\ t_{n+1}\right]$ satisfies

$$T_1 = t_{1,}\ T_{2i} = t_i + \frac{h_i}{3},\ T_{2i+1} = t_i + \frac{2h_i}{3} \qquad \text{........................(1)}$$

$$Q\left(T_i\right) = y_i \qquad \left(i = 1, 2, 3, ..., 2n+2\right) \qquad \text{..........................(2)}$$

then there exists a unique cubic polynomial

$$Q_i\left(x\right) = a_i + b_i u + c_i u^2 + d_i u^3 \qquad \text{.............................(3a)}$$

Where,
$$d_i = \frac{\left(Z_{i+1} - Z_i\right)}{6},$$

$$c_i = \frac{Z_i h_i^2}{2},$$

$$b_i = 3\left(y_{k+1} - y_k\right) - \frac{7}{54} h_i^2 Z_{i+1} - \frac{10}{27} Z_i h_i^2$$

$$a_i = \frac{\left(Z_{i+1} + 2Z_i\right)h_i^2}{27} + \left(2y_k - y_{k+1}\right)$$

$$k = 2i, \quad t_i \le x \le t_{i+1,} \qquad u = \frac{\left(x - t_i\right)}{h_i} \qquad \left(\,0 \le u \le 1\right)$$

and $Z_i = Q^{//}(t_i), \quad i = 1, 2, ...n+1$

Proof: The constructive proof of the above statement can be easily obtained.

2.2 Determination of Z_i values

The knots create n subintervals $[t_{i,}\ t_{i+1}], \quad (i = 1, 2, ..., n)$ and in each of them Q can be a different cubic polynomial, that is on $[t_{i,}\ t_{i+1}]$. Q is equal to a cubic polynomial Q_i. Since Q is a cubic polynomial, Q, $Q^/$ and $Q^{//}$ should be continues. Thus $Z_i = Q^{//}(t_i)$ is well defined. However, we do not yet know the values of $Z_{1,} Z_{2,}.........Z_{n+1}$. We shall now show how to determine the unknowns Z_i's. The cubic spline over $[t_{i,}\ \ t_{i+1}]$, $Q_i(x)$ as given in eqns must also satisfy the following continuity relations.

$$Q_{i-1}(t_i) = Q_i(t_i) \qquad \text{.......................(4)}$$

$$Q^/_{i-1}(t_i) = Q^/_i(t_i) \qquad \text{.......................(5)}$$

From eqns (3) and (4), eqns (3) and (5), the continuity of the spline function and its first derivatives imply the following for $(i = 2, 3, ..., n)$:

$$-h_{i-1}^{\ 2} Z_{i-1} + 2\left(h_i^2 - h_{i-1}^{\ 2}\right) Z_i + h_i^2 Z_{i+1} = 27\left[(2y_{2i-1} - y_{2i-2}) - (2y_{2i} - y_{2i+1})\right] \qquad \text{......................(6)}$$

$$7h_{i-1} Z_{i-1} + 20(h_i + h_{i-1}) Z_i + 7h_i Z_{i+1} = 162\left[\frac{y_{2i+1} - y_{2i}}{h_i - (y_{2i-1} - y_{2i-2})}\right] \qquad \text{.........................(7)}$$

Now eliminating Z_{i+1} between eqns (6) and (7), we obtain

$$7h_{i-1}^{\ 2}(h_i + h_{i-1}) Z_{i-1} + 2h_{i-1}(h_i + h_{i-1})(3h_i + 7h_{i-1}) Z_i = 162h_i(y_{2i-2} - y_{2i-1}) + 27h_{i-1}\left(-14y_{2i-1} + 7y_{2i-2} + 8y_{2i} - y_{2i+1}\right) \qquad \text{..................(8)}$$

Finally, imposing the first and the last interpolation conditions:

$$Q(T_1) = Q(t_1) = y_1 \qquad \text{.....................(9)}$$

$$Q(T_{2n+2}) = Q(t_{n+1}) = y_{2n+2} \qquad \text{.....................(10)}$$

We obtain

$$h_1^2(2Z_1 + Z_2) = 27(y_3 - 2y_2 + y_1) \qquad \text{..........................(11)}$$

$$h_n^2(Z_n + 2Z_{n+1}) = 27(y_{2n+2} - 2y_{2n+1} + y_{2n}) \qquad \text{.........................(12)}$$

Now eqns (8), (11) ands (12) can be expressed as a two band matrix:

$$\begin{bmatrix} 2h_1 & h_1 & 0 & 0 & 0 & & & \\ \alpha_2 & \beta_2 & 0 & 0 & 0 & & & \\ 0 & \alpha_3 & \beta_3 & 0 & 0 & & & \\ 0 & 0 & \alpha_4 & \beta_4 & 0 & & & \\ - & - & - & - & - & - & - & - \\ - & - & - & - & - & - & - & - \\ & & & \alpha_{n-2} & \beta_{n-2} & 0 & 0 & 0 \\ & & & 0 & \alpha_{n-1} & \beta_{n-1} & 0 & 0 \\ & & & 0 & 0 & \alpha_n & \beta_n & 0 \\ & & & 0 & 0 & 0 & h_n & 2h_n \end{bmatrix} \begin{bmatrix} Z_1 \\ Z_2 \\ Z_3 \\ Z_4 \\ - \\ Z_{n-3} \\ Z_{n-2} \\ Z_{n-1} \\ Z_n \\ Z_{n+1} \end{bmatrix} = \begin{bmatrix} r_1 \\ r_2 \\ r_3 \\ r_4 \\ - \\ r_{n-3} \\ r_{n-2} \\ r_{n-1} \\ r_n \\ r_{n+1} \end{bmatrix} \quad \text{.........................}(13)$$

Where,

$$r_1 = 27\left(y_3 - 2y_2 + y_1\right)/h_1$$
$$r_{n+1} = 27\left(y_{2n+2} - 2y_{2n+1} + y_{2n}\right)$$
$$\alpha_i = 7h_{i-1}^{\ 2}\left(h_i + h_{i-1}\right)$$
$$\beta_i = 2h_{i-1}\left(h_i + h_{i-1}\right)\left(3h_i + 7h_{i-1}\right)$$
$$r_i = 162h_i\left(y_{2i-2} - y_{2i-1}\right) + 27h_{i-1}\left(-14y_{2i-1} + 7y_{2i-2} + 8y_{2i} - y_{2i+1}\right)$$
$$\left(i = 2, 3, ...n\right) \quad \text{........................}(14)$$

The coefficient matrix of eqn (13) is a two band matrix and diagonally dominant. It can be shown that such a matrix is nonsingular, which means that the system has a unique solution.

The two band matrix eqns (13) - (14) can be solved by the following recurrence relations.

$$Z_i = \frac{\left(r_i - \alpha_{i-1}\right)}{\beta_i}, \qquad \left(i = 3, 4, ..., n\right) \qquad \text{...............................}(15)$$

With $Z_{1,}\ Z_2\ and\ Z_{n+1}$ given by :

$$Z_1 = \frac{\left(r_1\beta_2 - r_2h_1\right)}{\left(2h_1\beta_2 - h_1\alpha_2\right)}$$
$$Z_2 = \frac{\left(2h_1r_2 - r_1\alpha_2\right)}{\left(2h_1\beta_2 - h_1\alpha_2\right)}$$
$$Z_{n+1} = \frac{\left(r_{n+1}/h_n - Z_n\right)}{2} \qquad \text{..............................}(16)$$

3 APPLICATIONS TO QUADRATURE

3.1 Integral function approximations

Consider the indefinite integral

$$S(x)=\int_a^x f(t)dt, \qquad x\in[a,b] \qquad \text{.......................(17)}$$

We can choose an appropriate step size h and approximate the right side of eqn (17) for $x=a+h,\ a+2h,..................$ inturn. Each integral can be estimated by the spline integration.

Letting $f(t)=S(t)$, we can write

$$S(x)=\int_a^x S(t)dt, \qquad x\in[a,b] \qquad \text{....................(18)}$$

Where a is the first break point. Since $S(x)$ can be viewed as composed of connected Cubic polynomial with the k^{th} cubic polynomial being

$$S_k(u)=a_k+b_k u+c_k u^2+d_k u^3,$$

$$u=\frac{(x-x_k)}{h}, \qquad h=x_{k+1}-x_k \quad \forall k$$

$$x_k\le x\le x_{k+1}, \qquad 0\le u\le 1$$

$$a=x_1<x_2<........<x_{k+1}<x_k=x \qquad \text{............................(19)}$$

Letting, $x=x_k,$ we can write

$$S(x_k)=\int_{x_1}^{x_2}S_1(u)\,dx+\int_{x_2}^{x_3}S_2(u)dx+...........+\int_{x_{k-1}}^{x_k}S_{k-1}(u)du+\int_{x_k}^{x_{k+1}}S_k(u)du$$

$$=\sum_{x_k}^{x_{k+1}}S_i(1), \qquad (0\le u\le 1) \qquad \text{.............................}(20a)$$

Where

$$S_i(1)=\int_{x_i}^{x_{i+1}}S_i(u)dx=h\int_0^1 S_i(u)du$$

$$=h\left(a_i+\frac{b_i}{2}+\frac{c_i}{3}+\frac{d_i}{4}\right) \qquad \text{.........................(20b)}$$

3.2 COMPUTATION OF TABLE OF INTEGRALS

Example 1: Indefinite integral of Runge Function

$$I(x)=\int_{-1}^{x}\frac{dt}{(1+25t^2)}\ , \qquad x\in(-1,1] \qquad \text{..............................(21)}$$

Example 2: Logarithmic Function

$$I(x)=\int_1^x\frac{dt}{(1+t)}\ , \qquad x\in(1,5] \qquad \text{.............................(22)}$$

Example 3: Normal Distribution

$$I\left(x\right)=\int_{0}^{x}\frac{dt}{\left(\sqrt{2\pi}\right)}\exp\left(\frac{-t^{2}}{2}\right)dt\ ,\qquad x\in(0,4]\qquad ....(23)$$

The approximations to $I\left(x\right)$ will produce table of integrals in each case. These are implemented in the MATLAB program:

new_spline_int_cubic_subbotin.m

This program also computes the above integral functions approximations using the cubic splines available in standard texts, viz the clamped spline, natural spline and the not a knot spline (using matlab function spline). The sample outputs of these are expressed in the form of tables.

3.3. NUMERICAL INTEGRATION OVER CURVED DOMAINS IN 2-SPACE

Consider the integral [2, 16]

$$\iint_{\pi_{xy}} f\left(x,y\right)dx\ dy \qquad ....(24)$$

Where π_{xy} is the curved boundary.

Let the curved boundary be discritised into N curved arcs by points $\left(x_k,y_k\right),\left(k=1,2,3,..........N+1\right)$ and the i^{th} arc is joined by the vertices $V_i=\left(x_i,y_i\right)$ *and* $V_{i+1}=\left(x_{i+1},y_{i+1}\right)$, let us denote this curved arc as $\partial\pi_{i,\,i+1}$.

Thus we represent the closed curved boundary as

$$\partial\pi_{xy}=\partial\pi_{1,2}+\partial\pi_{2,3}+.........\partial\pi_{N,N+1}$$

We have by application of Green's theorem

$$\iint_{\pi_{xy}} f\left(x,y\right)dx\ dy=\oint\phi(x,y)\,dy$$

$$\partial\pi_{xy}=\sum_{i=1}^{N}\partial\pi_{i,i+1,} \qquad ....(25a)$$

$$=\sum_{1}^{N}\int_{\partial\pi_{i,i+1}}\varphi\left(x,y\right)\,dy \qquad ....(25b)$$

Where

$$\phi\left(x,y\right)=\int_{\alpha}^{x}f\left(u,y\right)du \qquad ....(26)$$

In which α is fixed and $x_i\neq\alpha$ $\left(i=1,2,........,N+1\right)$

Let us now choose the substitution

$$u=\left(\frac{x-\alpha}{2}\right)s+\left(\frac{x+\alpha}{2}\right) \qquad ....(27)$$

Substituting for u from eqn (30) into eqn (29)

We obtain

$$\phi(x,y)=\left(\frac{x-\alpha}{2}\right)\int_{-1}^{1} f\left(\left(\frac{x-\alpha}{2}\right)s+\left(\frac{x+\alpha}{2}\right),y\right)ds \quad \text{.............................(28)}$$

Substituting the above expression for $\phi(x,y)$ from eqn (28) into eqn (25b), we obtain

$$\iint_{\pi_{xy}} f(x,y)\,dx\,dy=\sum_{i=1}^{N}\int_{\partial\pi_{i,i+1}}\phi(x.y)\,dy$$

$$=\sum_{i=1}^{N}\int_{\partial\pi_{i,i+1}}\left(\frac{x-\alpha}{2}\right)\left[\int_{-1}^{1} f\left(\left(\frac{x-\alpha}{2}\right)s+\left(\frac{x+\alpha}{2}\right),y\right)ds\right]dy \quad \text{.........(29)}$$

By using suitable parametric form of splines for x and y with parameter t, we can write

$$\iint_{\pi_{xy}} f(x,y)dx\,dy=$$

$$\sum_{i=1}^{N}\int_{-1}^{1}\left(\frac{x^i(t)-\alpha}{2}\right)\left[\int_{-1}^{1} f\left(\frac{x^i(t)-\alpha}{2}\right)s+\frac{x^i(t)+\alpha}{2},y^i(t),\right]ds\left(y^i(t)\right)^{/}dt \quad \text{.........(30)}$$

Where $x^i(t)$ *and* $y^i(t)$ refer to parametric splines of order higher than two over the curved arc joining the points (x_i,y_i) *and* (x_{i+1},y_{i+1}).

3.4 COMPUTATION OF GEOMETRIC MOMENTS

We 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 the location (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 angle θ in anticlockwise direction. Let us denote the points on the rotated ellipse with respect to $(x^/,y^/)$ coordinate system as (x^*,y^*). This is illustrated in Fig-5 and Fig-6. Thus we have from [14],

(i) The points on the ellipse $\frac{x^{/2}}{a^2}+\frac{y^{/2}}{b^2}=1$ are

$$x^/=a\cos t,\quad y^/=b\sin t,\quad 0\le t\le 2\pi$$

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

$$x^*=x^/\cos\theta-y^/=b\sin\theta,\qquad y^*=x^/\sin\theta+y^/\cos\theta$$

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

$$x^/=x^*,\quad y^/=y^*$$

ie,
$$x' = x - x_0 = x^*, \quad y' = y - y_0 = y^*$$
$$\therefore\ x = x_0 + x^*,\ \ y = y_0 + y^*$$

We shall now determine the points on the rotated ellipse. Let the points on the ellipse with respect to (x', y') system 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. We have implement this procedure in the MATLAB program points_on_general_ellipse.m. The geometric moments over the general ellipse described above are given by

$$m_{pq} = \iint_A x^p y^q \, dx\, dy \qquad \text{..............................(31)}$$

Where A is the general ellipse.

The computed 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 in [12] are also tabulated for the purpose of comparison.

In tables 1 a-b,we have depicted the results from [12].We note that in these tables the following nomenclature is used:

Method-1 : The δ -method

Method-2 : The rectangular integration method

Method-3 : The trapezoidal integration method

Method-4: The contour integration method.

The exact values are computed by the following formulas [12].

$$m_{00} = \pi ab$$
$$m_{10} = \pi ab x_0$$
$$m_{01} = \pi ab y_0$$
$$m_{20} = \pi ab\left(\frac{a^2\cos^2\lambda + b^2\sin^2\lambda}{4} + x_0^{\,2}\right)$$
$$m_{02} = \pi ab\left(\frac{a^2\sin^2\lambda + b^2 cos^2\lambda}{4} + y_0^{\,2}\right)$$
$$m_{11} = \pi ab\left(\frac{(a^2 - b^2)\cos\lambda\sin\lambda}{4} + x_0 y_0\right)$$

Where a = length of semi major axis

b = length of semi minor axis

x_0, y_0 : coordinates of the centre of ellipse

$\lambda =$ Angle which the major axis makes with the x-axis.

The matlab program :

new_greens_cubic_spline_integration_general_ellipse$\left(div, n, xc, yc, a, b, z\right)$

where,

div = number of spline subintervals * 3

n = order Gauss Legendre quadrature

$\left(xc, yc\right)$: centre of ellipse

a, b : length of semi major and minor axes

z : angle of rotation of the ellipse

Sample output for n=32, xc=10, yc=30, a=30, b= 20, $z = \frac{pi}{4}$ and div given as numbers 3,6,9,………..1500 were computed. We have appended some sample output results for div=3,6,9,96,1500 are given in table-5.

EXACT VALUES

m_{00} = 1.884955592153876 e + 003

m_{01} = 5.654866776461627 e + 004

m_{20} = 4.948008429403924 e + 005

m_{02} = 2.002765316663493 e + 006

m_{11} = 6.832964021557800 e + 005

We have also computed the percentage error by the formula

Percentage error = $abs\left(m_{pq}^{computed} - m_{pq}^{exact}\right)100 / m_{pq}^{exact}$

4. CONCLUSIONS

In this paper, we have presented a formulation and study of an interpolatory cubic spline which is new and akin to the Subbotin quadratic spline [5-17]. This spline interpolates at the first and the last knots and at two points located at trisections between the knots. This paper also presents numerical schemes to compute the integral approximations (indefinite integral) in one dimension and the integration over curved domains in the Cartesian two space. Computation of the integral approximation is illustrated with examples. It is shown that they are useful in the construction of the table of integrals. Integration over curved domain is explained by computing geometric moments for a general ellipse. The results are obtained with excellent accuracy which have a maximum percentage error equal to 10^{-11}, where as the computations in the classical text [12] were obtained with a percentage error of 2.62, 2.70, 2.21, 1.01 by the four different methods viz; the δ-method, the rectangular integration method, the trapezoidal integration method and the contour integration method. The performance of the proposed new cubic spline is also compared with the standard cubic splines (natural, clamped and a not a knot).

REFERENCES

[1] Antia, H.M., (1991) Numerical Methods for scientists and engineers, Tata Mc Graw Hill Company Ltd, New Delhi.

[2] Apostol. T.M. Calculus, Vol.II, 2^{nd} edition, Blaisdell, 1969.

[3] B.Bradie (2008) A friendly introduction to numerical analysis, Pearson Education Inc.

[4] Boar. C.de, (1978) A practical guide to splines, Springer-Verlag, New York.

[5] Burden,R.L and J.D.Faires,(2001) Numerical Analysis 7th Edn.Pacific Grove, California: Brooks/Cole

[6] Cheney.w and D.Kinicaid,(2004), Numerical Mathematics and Computing, 5th Edn, Thomson Brooks/cole.

[7] Elden,L, LeWittmeyer-Koch, H.B.Nielsen, (2006) Introduction to Numerical Computation, analysis and MATLAB® illustrations, Overseas Press India Pvt Ltd.

[8] Gerald, C.F. and P.O. Wheatly. (1999) Applied Numerical Analysis, 6th Edn, Pearson Education Inc.

[9] Hanselman, D and B.Littlefield (2007), Mastering MATLAB7® , Pearson Education, Inc.

[10] Kammerer, W.J, G.W. Reddien, and R.S. Varga (1974), Quadratic splines, Numerisohe Mathematik, 22, 241-251.

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

[12] Mukundan R. and K.R.Ramakrishanan Moment Functions in Image Analysis, World Scientific Publishing Co.Pvt Ltd (1998)

[13] Phillips. G.M. and P.J.Taylor (1996) Theory and applications of Numerical Analysis, second Edition, Academic Press, London.

[14] Rogers, D.F and J.A.Adams (2002) Mathematical elements of Computer graphics, Tata Mc Graw-Hill Publishing Co.Ltd.

[15] Schilling. J. and S.L.Harris (2007) Applied Numerical Methods for engineering, using MATLABR and C, Books/Cole, (a part of cengage Learning).

[16] Sommariva.A and M.Vianello, Gauss-Green cubature over spline curvilinear polygons, Applied Mathematics and Computation, vol 183, No.2, pp 1098-1107 (2006).

[17] Subbotin, Y.N. (1967), On piecewise-polynomial approximation, Mat.Zametcki 1, 63-70 (Translation:1967. Math Notes 1, 41-46)

[18] Yang,W.Y, W.Cao, T.S.Chung, and J.Morris, (2005) Applied Numerical Methods using MATLAB®, John Wiley and Sons.Inc.

TABLE 1- a

Computed values of moment m_{00}, m_{10} and m_{01}

Methods	m_{00}	m_{10}	m_{01}
Method 1	1878.00	17841.00	56340.00
Method 2	1929.00	19290.00	57870.00
Method 3	1857.00	18657.50	55056.00
Method 4	1869.00	1869.00	56070.00
Exact as in [12]	1884.96	18849.60	56548.66

TABLE 1- b

Computed values of moment m_{20}, m_{02} and m_{11}

Methods	m_{20},	m_{02}	m_{11}
Method 1	472419.00	1994596.00	651180.00
Method 2	516768.75.00	2051706.88	698866.00
Method 3	482495.50	1933136.75	668597.00
Method 4	489113.00	1980871.00	674512.50
Exact as in [12]	494800.84	2002765.31	683296.40

TABLE-2-a

RUNGE FUNCTION INTEGRAL $= \int_{-1}^{x} \frac{1}{\left(1+25t^2\right)} dt$

x	Cubic Subbotin Spline	Natural Spline	Exact
-0.96	0.00159996602042	0.00159996602042	0.00159996586798
-0.92	0.00333302534289	0.00333291432551	0.00333302474279
-0.88	0.00521620933478	0.00521612638606	0.00521620825815
-0.84	0.00726952592022	0.00726943331129	0.00726952420380
-0.80	0.00951662318369	0.00951653042666	0.00951662065540
-0.76	0.01198563460978	0.01198553835825	0.01198563102424
-0.72	0.01471026306531	0.01471016314832	0.01471025809771
-0.68	0.01773118316554	0.01773107818659	0.01773117637349
-0.64	0.02109787092452	0.02109775929644	0.02109786170502
-0.60	0.02487101138281	0.02487089084876	0.02487099890935
-0.56	0.02912569297475	0.02912556046133	0.02912567611416

-0.52	0.03395567758607	0.03395552888187	0.03395565479367
-0.48	0.03947914276296	0.03947897217396	0.03947911196998
-0.44	0.04584642810741	0.04584622815271	0.04584638665540
-0.40	0.05325046504333	0.05325022652693	0.05325040983018
-0.36	0.06194066062908	0.06194037399654	0.06194058890849
-0.32	0.07224083917045	0.07224049950146	0.07224075109874
-0.28	0.08457087983609	0.08457050102144	0.08457078522659
-0.24	0.09946860939441	0.09946825922504	0.09946854326936
-0.20	0.11760046918276	0.11760034302998	0.11760052070951
-0.16	0.13973162302703	0.13973215617372	0.13973196494429
-0.12	0.16659542193349	0.16659729235774	0.16659625333489
-0.08	0.19857766701988	0.19858114780595	0.19857887796653
-0.04	0.23520031780477	0.23520347456817	0.23520104141903
0.00	0.27468081092706	0.27468007022368	0.27468015338900
0.04	0.31416062591739	0.31415666587920	0.31415926535898
0.08	0.35078230627314	0.35077899264142	0.35078142881148
0.12	0.38276433473857	0.38276284808962	0.38276405344312
0.159	0.40962833358333	0.40962798427365	0.40962834183371
0.20	0.43175968738734	0.43175979741739	0.43175978606849
0.24	0.44989165576688	0.44989188122233	0.44989176350864
0.28	*0.46478942923129*	0.46478963942593	0.46478952155142
0.32	0.47711948282318	0.47711964094591	0.47711955567927
0.36	0.48741966211928	0.48741976645083	0.48741971786952
0.399	0.49610985468976	0.49610991392044	0.49610989694782
0.44	0.50351388804248	0.50351391229466	0.50351392012261
0.48	0.50988117027174	0.50988116827340	0.50988119480803
0.52	0.51540463300912	0.51540461156549	0.51540465198434
0.56	0.52023461579352	0.52023457998603	0.52023463066384
0.60	0.52448929604318	0.52448924959860	0.52448930786865
0.639	0.52826243552178	0.52826238115093	0.52826244507299
0.68	0.53162912256560	0.53162906226078	0.53162913040452
0.72	0.53465004214157	0.53464997729905	0.53465004868030
0.76	0.53737467021086	0.53737460208911	0.53737467575376
0.80	0.53984368134898	0.53984361002071	0.53984368612261
0.84	0.54209077839986	0.54209070713607	0.54209078257420
0.879	0.54414409481617	0.54414401406130	0.54414409851986
0.92	0.54602727870406	0.54602722612185	0.54602728203522
0.96	0.54776033787596	0.54776017442694	0.54776034091003
1.00	0.54936030383282	0.54936054973792	0.54936030677801

TABLE-2-b

RUNGE FUNCTION INTEGRAL $= \int_{-1}^{x} \frac{1}{\left(1+25t^2\right)} dt$

TABLE-3a

x	Clamped Spline	Not a knot Spline	Exact
-0.96	0.00159996602042	0.00159996602042	0.00159996586798
-0.92	0.00333302417451	0.00333302182898	0.00333302474279
-0.88	0.00521620680111	0.00521620508407	0.00521620825815
-0.84	0.00726952161314	0.00726951972770	0.00726952420380
-0.80	0.00951661661525	0.00951661477492	0.00951662065540
-0.76	0.01198562511309	0.01198562326068	0.01198563102424
-0.72	0.01471024975143	0.01471024790226	0.01471025809771
-0.68	0.01773116483035	0.01773116298031	0.01773117637349
-0.64	0.02109784592931	0.02109784407950	0.02109786170502
-0.60	0.02487097748456	0.02487097563468	0.02487099890935
-0.56	0.02912564709634	0.02912564524649	0.02912567611416
-0.52	0.03395561551709	0.03395561366723	0.03395565479367
-0.48	0.03947905880913	0.03947905695927	0.03947911196998
-0.44	0.04584631478789	0.04584631293803	0.04584638665540
-0.40	0.05325031316210	0.05325031131224	0.05325040983018
-0.36	0.06194046063171	0.06194045878185	0.06194058890849
-0.32	0.07224058613663	0.07224058428678	0.07224075109874
-0.28	0.08457058765661	0.08457058580675	0.08457078522659
-0.24	0.09946834586021	0.09946834401035	0.09946854326936
-0.20	0.11760042966515	0.11760042781530	0.11760052070951
-0.16	0.13973224280889	0.13973224095903	0.13973196494429
-0.12	0.16659737899292	0.16659737714306	0.16659625333489
-0.08	0.19858123444112	0.19858123259127	0.19857887796653
-0.04	0.23520356120334	0.23520355935349	0.23520104141903
0.00	0.27468015685886	0.27468015500900	0.27468015338900
0.04	0.31415675251437	0.31415675066451	0.31415926535898
0.08	0.35077907927659	0.35077907742673	0.35078142881148
0.12	0.38276293472480	0.38276293287494	0.38276405344312

0.159	0.40962807090882	0.40962806905897	0.40962834183371
0.20	0.43175988405256	0.43175988220270	0.43175978606849
0.24	0.44989196785750	0.44989196600764	0.44989176350864
0.28	0.46478972606110	0.46478972421124	0.46478952155142
0.32	0.47711972758108	0.47711972573122	0.47711955567927
0.36	0.48741985308600	0.48741985123615	0.48741971786952
0.399	0.49611000055562	0.49610999870576	0.49610989694782
0.44	0.50351399892983	0.50351399707997	0.50351392012261
0.48	0.50988125490859	0.50988125305873	0.50988119480803
0.52	0.51540469820062	0.51540469635077	0.51540465198434
0.56	0.52023466662137	0.52023466477151	0.52023463066384
0.60	0.52448933623316	0.52448933438331	0.52448930786865
0.639	0.52826246778841	0.52826246593850	0.52826244507299
0.68	0.53162914888736	0.53162914703769	0.53162913040452
0.72	0.53465006396628	0.53465006211574	0.53465004868030
0.76	0.53737468860462	0.53737468675732	0.53737467575376
0.80	0.53984369710247	0.53984369524307	0.53984368612261
0.84	0.54209079210457	0.54209079029030	0.54209078257420
0.879	0.54414410691660	0.54414410493393	0.54414409851986
0.92	0.54602728954320	0.54602728818901	0.54602728203522
0.96	0.54776034769729	0.54776034399758	0.54776034091003
1.00	0.54936031304622	0.54936031810013	0.54936030677801

NORMAL DISTRIBUTION INTEGRAL = $\int_0^x \frac{e^{\frac{-t^2}{2}}}{\sqrt{2\pi}} dt$

x	Cubic Subbotin Spline	Natural Spline	Exact
0.08	0.03188137343786	0.03188137343786	0.03188137201399
0.16	0.06355946719297	0.06356113109220	0.06355946289143
0.24	0.09483487811231	0.09483608690316	0.09483487169780
0.32	0.12551584316831	0.12551716551086	0.12551583472332
0.40	0.15542175168553	0.15542303638763	0.15542174161032
0.48	0.18438631483954	0.18438760376580	0.18438630348378
0.56	0.21226029337979	0.21226157681008	0.21226028115097
0.64	0.23891371300311	0.23891499511253	0.23891370030714
0.72	0.26423751498017	0.26423879623512	0.26423750222075
0.80	0.28814461385800	0.28814589567200	0.28814460141660
0.88	0.31057035699849	0.31057164043043	0.31057034522329
0.96	0.33147240333869	0.33147368939436	0.33147239253316
1.04	0.35083005925386	0.35083134876756	0.35083004966902
1.12	0.36864312712815	0.36864442076743	0.36864311895727

1.20	0.38493033640194	0.38493163465055	0.38493032977829
1.28	0.39972743704822	0.39972874020598	0.39972743204556
1.36	0.41308504141739	0.41308634960587	0.41308503805292
1.44	0.42506630222622	0.42506761540202	0.42506630046567
1.52	0.43574451241669	0.43574583039020	0.43574451218106
1.60	0.44520070712697	0.44520202958538	0.44520070830044
1.68	0.45352133969787	0.45352266623069	0.45352134213628
1.76	0.46079609317294	0.46079742329864	0.46079609671252
1.84	0.46711587687507	0.46711721006742	0.46711588134084
1.92	0.47257104508264	0.47257238079541	0.47257105029616
2.00	0.47724986226558	0.47725119995480	0.47724986805182
2.08	0.48123722737203	0.48123856651497	0.48123723356506
2.16	0.48461365876857	0.48461499887909	0.48461366521607
2.24	0.48745453199754	0.48745587263763	0.48745453856405
2.319	0.48982955476235	0.48983089554978	0.48982956133128
2.40	0.49180245760085	0.49180379821340	0.49180246407540
2.48	0.49343087456130	0.49343221473777	0.49343088086445
2.56	0.49476638576280	0.49476772530135	0.494766391836440
2.64	0.49585469283547	0.49585603158990	0.49585469863896
2.72	0.49673589867582	0.49673723655025	0.49673590418411
2.799	0.49744486446812	0.49744620141064	0.49744486966957
2.88	0.49801161925099	0.49801295524678	0.49801162414511
2.96	0.49846180019308	0.49846313525744	0.49846180478826
3.04	0.49881710494551	0.49881843911708	0.49881710925690
3.12	0.49909574075267	0.49909707408715	0.49909574480018
3.20	0.49931285825549	0.49931419082013	0.49931286206208
3.279	0.49948096097656	0.49948229284490	0.49948096456679
3.36	0.49961028423860	0.49961161548812	0.49961028763742
3.44	0.49970913967489	0.49971047037758	0.49970914290671
3.52	0.49978422351271	0.49978555375612	0.49978422660071
3.60	0.49984088844420	0.49984221824195	0.49984089140984
3.68	0.49988338016042	0.49988470977864	0.49988338302318
3.759	0.49991504054429	0.49991636927864	0.49991504332150
3.84	0.49993848013798	0.49993981087527	0.49993848284482
3.92	0.49995572286615	0.49995704505960	0.49995572551569
4.00	0.49996832612598	0.49996967936185	0.49996832875817

TABLE-3b

NORMAL DISTRIBUTION INTEGRAL = $\int_0^x \frac{e^{\frac{-t^2}{2}}}{\sqrt{2\pi}}dt$

x	Clamped Spline	Not a knot Spline	Exact
0.08	0.03188137343786	0.03188137343786	0.03188137201399
0.16	0.06355945901043	0.06355943939126	0.06355946289143
0.24	0.09483486285434	0.09483484849212	0.09483487169780
0.32	0.12551582141198	0.12551580564116	0.12551583472332
0.40	0.15542172445606	0.15542170906268	0.15542174161032
0.48	0.18438628321502	0.18438626772050	0.18438630348378
0.56	0.21226025856881	0.21226024310139	0.21226028115097
0.64	0.23891367625244	0.23891366077775	0.23891370030714
0.72	0.26423747754084	0.26423746206811	0.26423750222075
0.80	0.28814457693329	0.28814456146004	0.28814460141660
0.88	0.31057032170363	0.31057030623051	0.31057034522329
0.96	0.33147237066436	0.33147235519120	0.33147239253316
1.04	0.35083003003842	0.35083001456527	0.35083004966902
1.12	0.36864310203806	0.36864308656492	0.36864311895727
1.20	0.38493031592125	0.38493030044810	0.38493032977829
1.28	0.39972742147666	0.39972740600351	0.39972743204556
1.36	0.41308503087655	0.41308501540340	0.41308503805292
1.44	0.42506629667270	0.42506628119955	0.42506630046567
1.52	0.43574451166088	0.43574449618773	0.43574451218106
1.60	0.44520071085606	0.44520069538291	0.44520070830044
1.68	0.45352134750136	0.45352133202822	0.45352134213628
1.76	0.46079610456932	0.46079608909617	0.46079609671252
1.84	0.46711589133810	0.46711587586495	0.46711588134084
1.92	0.47257106206609	0.47257104659294	0.47257105029616
2.00	0.47724988122548	0.47724986575233	0.47724986805182
2.08	0.48123724778565	0.48123723231250	0.48123723356506
2.16	0.48461368014977	0.48461366467663	0.48461366521607
2.24	0.48745455390831	0.48745453843516	0.48745453856405
2.319	0.48982957682046	0.48982956134731	0.48982956133128
2.40	0.49180247948408	0.49180246401093	0.49180246407540
2.48	0.49343089600844	0.49343088053530	0.49343088086445
2.56	0.49476640657203	0.49476639109888	0.494766391836440
2.64	0.49585471286058	0.49585469738743	0.49585469863896
2.72	0.49673591782093	0.49673590234778	0.49673590418411
2.799	0.49744488268132	0.49744486720817	0.49744486966957
2.88	0.49801163651746	0.49801162104431	0.49801162414511
2.96	0.49846181652812	0.49846180105497	0.49846180478826

3.04	0.49881712038776	0.49881710491461	0.49881710925690
3.12	0.49909575535784	0.49909573988469	0.49909574480018
3.20	0.49931287209076	0.49931285661762	0.49931286206208
3.279	0.49948097411575	0.49948095864259	0.49948096456679
3.36	0.49961029675815	0.49961028128504	0.49961028763742
3.44	0.49970915165070	0.49970913617742	0.49970914290671
3.52	0.49978423501770	0.49978421954504	0.49978422660071
3.60	0.49984089954657	0.49984088407160	0.49984089140984
3.68	0.49988339092263	0.49988337545627	0.49988338302318
3.759	0.49991505102209	0.49991503552362	0.49991504332150
3.84	0.49993849038154	0.49993847500289	0.49993848284482
3.92	0.49995573291516	0.49995571708931	0.49995572551569
4.00	0.49996833605743	0.49996832190057	0.49996832875817

TABLE-4a

LOGARITHMIC FUNCTION INTEGRAL = $\int_1^x \frac{1}{(1+t)}dt$

x	cubic_subbotin_spline	natural_ spline	Exact
1.08	0.03922071396027	0.03922071396027	0.03922071315328
1.16	0.07696104319944	0.07695999495151	0.07696104113613
1.24	0.11332868821323	0.11332791689560	0.11332868530700
1.32	0.14842000880542	0.14841915996469	0.14842000511827
1.40	0.18232156111573	0.18232073025801	0.18232155679395
1.48	0.21511138448704	0.21511054644682	0.21511137961695
1.56	0.24686008326688	0.24685924513945	0.24686007793153
1.64	0.27763174233409	0.27763090250854	0.27763173659828
1.72	0.30748470582867	0.30748386497839	0.30748469974796
1.80	0.33647224300070	0.33647140114729	0.33647223662121
1.88	0.36464312022710	0.36464227753464	0.36464311358791
1.96	0.39204209464187	0.39204125120987	0.39204208777602
2.04	0.41871034192254	0.41870949784609	0.41871033485819
2.12	0.44468582850026	0.44468498385764	0.44468582126145
2.20	0.47000363663837	0.47000279149742	0.47000362924574
2.28	0.49469624936477	0.49469540378362	0.49469624183611
2.36	0.51879380106450	0.51879295509338	0.51879379341517
2.439	0.54232429858202	0.54232345226443	0.5423242908256
2.52	0.56531381690245	0.56531297027618	0.5653138090506
2.60	0.58778667284010	0.58778582593813	0.58778666490212
2.679	0.60976557963560	0.60976473248676	0.60976557162089
2.760	0.63127178492549	0.63127093755506	0.63127177684186

2.840	0.65232519418539	0.65232434661562	0.65232518603969
2.920	0.67294448144414	0.67294363369464	0.67294447324243
3.00	0.69314718881230	0.69314634090043	0.69314718055995
3.080	0.71294981615437	0.71294896809550	0.71294980785613
3.160	0.73236790205313	0.73236705386091	0.73236789371323
3.240	0.75141609706171	0.75141524874832	0.75141608868392
3.319	0.77010823010838	0.77010738168467	0.77010822169607
3.40	0.78845736880808	0.78845652028376	0.78845736036427
3.480	0.80647587433955	0.80647502572333	0.80647586586695
3.560	0.82417545146531	0.82417460276501	0.82417544296635
3.640	0.84156719420135	0.84156634542400	0.84156718567822
3.720	0.85866162758285	0.85866077873478	0.85866161903752
3.79	0.87546874591965	0.87546789700658	0.87546873735390
3.880	0.89199804788966	0.89199719891678	0.89199803930511
3.960	0.90825856877877	0.90825771975076	0.90825856017689
4.040	0.92425891014122	0.92425806106235	0.92425890152333
4.120	0.94000726712415	0.94000641799821	0.94000725849147
4.20	0.95551145367381	0.95551060450466	0.95551144502744
4.279	0.97077892581729	0.97077807660683	0.97077891715822
4.360	0.98581680319361	0.98581595394963	0.98581679452277
4.440	1.00063188898968	1.00063103969670	1.00063188030791
4.520	1.01523068842100	1.01522983914930	1.01523067972906
4.60	1.02961942588257	1.02961857638119	1.02961941718116
4.680	1.04380406088334	1.04380321209931	1.04380405217311
4.760	1.05779030286630	1.05778945127489	1.05779029414785
4.840	1.07158362500631	1.07158278377077	1.07158361628019
4.920	1.08518927706926	1.08518839707176	1.08518926833597
5.00	1.09861229740415	1.09861156196633	1.09861228866811

TABLE-4b

LOGARITHMIC FUNCTION INTEGRAL = $\int_1^x \frac{1}{(1+t)}dt$

x	clamped_spline	Not a knot_spline	Exact
1.08	0.03922071396027	0.03922071396027	0.03922071315328
1.16	0.07696103950333	0 . 07696102915780	0.07696104113613
1.24	0.11332868156060	0.11332867398715	0.11332868530700
1.32	0.14841999962514	0 .14841999130891	0.14842000511827
1.40	0.18232154982349	0.18232154170628	0.18232155679395
1.48	0.21511137139673	0.21511136322620	0.21511137961695
1.56	0.24686006864661	0.24686006049037	0.24686007793153
1.64	0.27763172640228	0.27763171824221	0.27763173659828

1.72	0.30748468876854	0.30748468060950	0.30748469974796
1.80	0.33647222496520	0.33647221680588	0.33647223662121
1.88	0.36464310134512	0.36464309318587	0.36464311358791
1.96	0.39204207502234	0.39204206686307	0.39204208777602
2.04	0.41871032165802	0.41871031349876	0.41871033485819
2.12	0.44468580766972	0.44468579951046	0.44468582126145
2.20	0.47000361530946	0.47000360715019	0.47000362924574
2.28	0.49469622759567	0.49469621943641	0.49469624183611
2.36	0.51879377890542	0.51879377074616	0.51879379341517
2.439	0.54232427607647	0.54232426791721	0.54232429082536
2.52	0.56531379408823	0.56531378592897	0.56531380905006
2.60	0.58778664975018	0.58778664159092	0.58778666490212
2.679	0.60976555629880	0.60976554813954	0.60976557162089
2.760	0.63127176136711	0.63127175320785	0.63127177684186
2.840	0.65232517042767	0.65232516226841	0.65232518603969
2.920	0.67294445750669	0.67294444934743	0.67294447324243
3.00	0.69314716471248	0.69314715655322	0.69314718055995
3.080	0.71294979190754	0.71294978374828	0.71294980785613
3.160	0.73236787767296	0.73236786951370	0.73236789371323
3.240	0.75141607256037	0.75141606440110	0.75141608868392
3.319	0.77010820549672	0.77010819733746	0.77010822169607
3.40	0.78845734409581	0.78845733593655	0.78845736036427
3.480	0.80647584953538	0.80647584137612	0.80647586586695
3.560	0.82417542657706	0.82417541841780	0.82417544296635
3.640	0.84156716923605	0.84156716107678	0.84156718567822
3.720	0.85866160254683	0.85866159438757	0.85866161903752
3.79	0.87546872081863	0.87546871265937	0.87546873735390
3.880	0.89199802272882	0.89199801456956	0.89199803930511
3.960	0.90825854356281	0.90825853540355	0.90825856017689
4.040	0.92425888487438	0.92425887671512	0.92425890152333
4.120	0.94000724181031	0.94000723365105	0.94000725849147
4.20	0.95551142831649	0.95551142015722	0.95551144502744
4.279	0.97077890041969	0.97077889226042	0.97077891715822
4.360	0.98581677775865	0.98581676959939	0.98581679452277
4.440	1.00063186352005	1.00063185536077	1.00063188030791
4.520	1.01523066291915	1.01523065475995	1.01523067972906
4.60	1.02961940035073	1.02961939219125	1.02961941718116
4.680	1.04380403532359	1.04380402716516	1.04380405217311
4.760	1.05779027728051	1.05779026911814	1.05779029414785
4.840	1.07158359939627	1.07158359124860	1.07158361628019
4.920	1.08518925143643	1.08518924323388	1.08518926833597
5.00	1.09861227175446	1.09861226375673	1.09861228866811

TABLE-5
computation of geometric moments for a general ellipse

div	computed values	---divisions exact values	percentage error
	1.860102337133282e+003	1.884955592153876e+003	1.318506129483686e+000
	1.859463903101015e+004	1.884955592153876e+004	1.352376106841460e+000
	5.597130033331437e+004	5.654866776461627e+004	1.021009785244086e+000
3	4.924105139843419e+005	4.948008429403924e+005	4.830891034553890e-001
	1.984629784867467e+006	2.002765316663493e+006	9.055245587255791e-001
	6.790337462588199e+005	6.832964021557800e+005	6.238370176561014e-001
	1.883355429654148e+003	1.884955592153876e+003	8.489125719397192e-002
	1.883634458436845e+004	1.884955592153876e+004	7.008832051696488e-002
	5.650263677728872e+004	5.654866776461627e+004	8.140065742867456e-002
6	4.945741579882905e+005	4.948008429403924e+005	4.581337225595861e-002
	2.001398816247745e+006	2.002765316663493e+006	6.823068106772859e-002
	6.829473824294771e+005	6.832964021557800e+005	5.107881809443068e-002
	1.884622648459191e+003	1.884955592153876e+003	1.766321159344815e-002
	1.884664916900807e+004	1.884955592153876e+004	1.542080112015428e-002
	5.653895679223974e+004	5.654866776461627e+004	1.717276951059834e-002
9	4.947521906250537e+005	4.948008429403924e+005	9.832706640018219e-003
	2.002461613468638e+006	2.002765316663493e+006	1.516419284518073e-002
	6.832114618469439e+005	6.832964021557800e+005	1.243096093702114e-002
	1.884955562797002e+003	1.884955592153876e+003	1.557430509528396e-006
	1.884955563693564e+004	1.884955592153876e+004	1.509866427527215e-006
96	5.654866689802185e+004	5.654866776461627e+004	1.532475406228956e-006
	4.948008377920404e+005	4.948008429403924e+005	1.040489728873435e-006
	2.002765288877702e+006	2.002765316663493e+006	1.387371316980932e-006
	6.832963929957649e+005	6.832964021557800e+005	1.340562473484970e-006
	1.884955592153383e+003	1.884955592153876e+003	2.615160433555116e-011
	1.884955592153412e+004	1.884955592153876e+004	2.458829809851822e-011
1500	5.654866776460162e+004	5.654866776461627e+004	2.591357005254615e-011
	4.948008429403082e+005	4.948008429403924e+005	1.702230029384447e-011
	2.002765316663024e+006	2.002765316663493e+006	2.342529816444665e-011
	6.832964021556297e+005	6.832964021557800e+005	2.199516637353943e-011

FIGURE-1

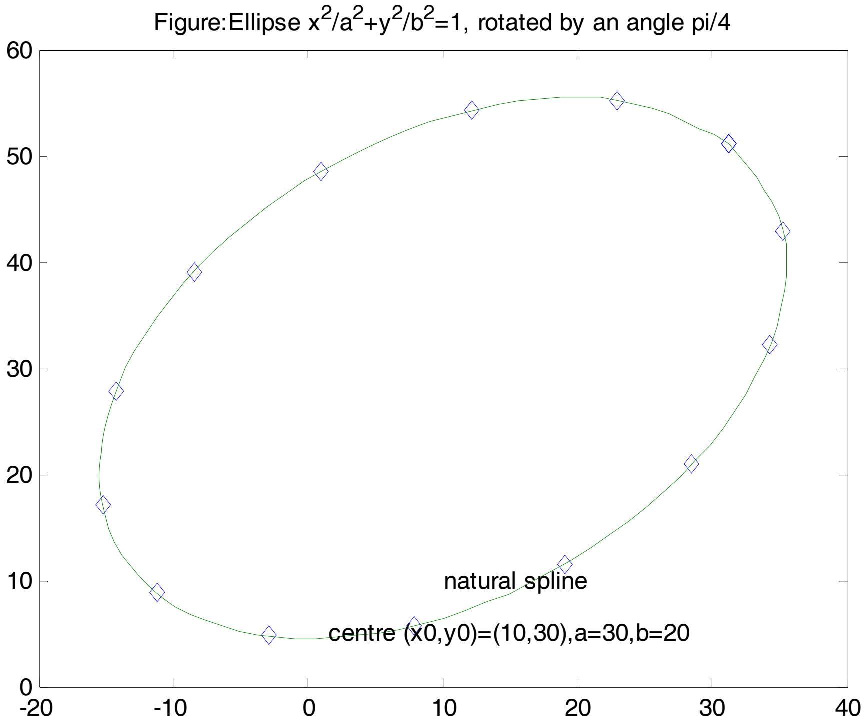


FIGURE-2

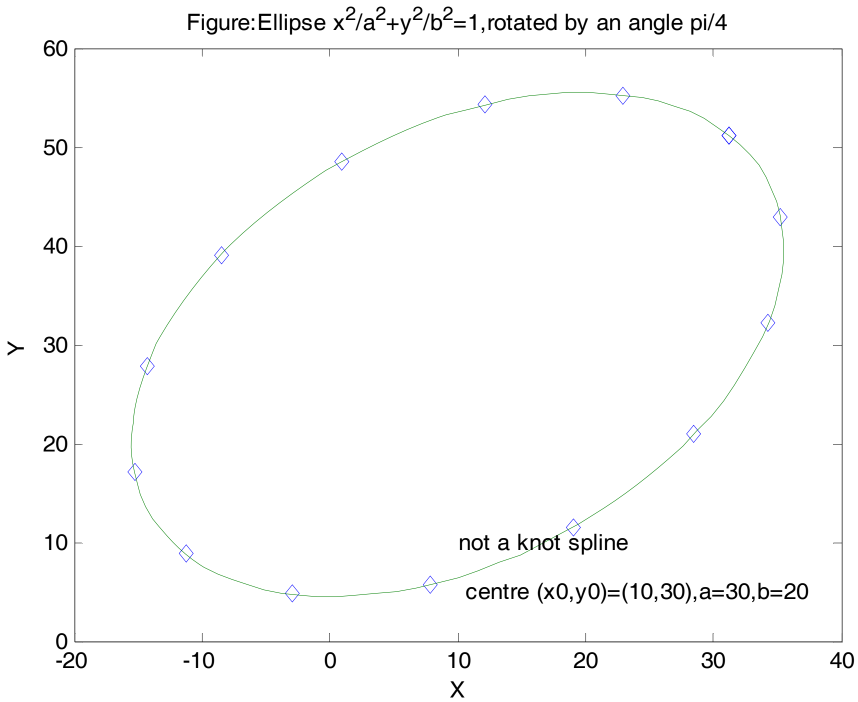
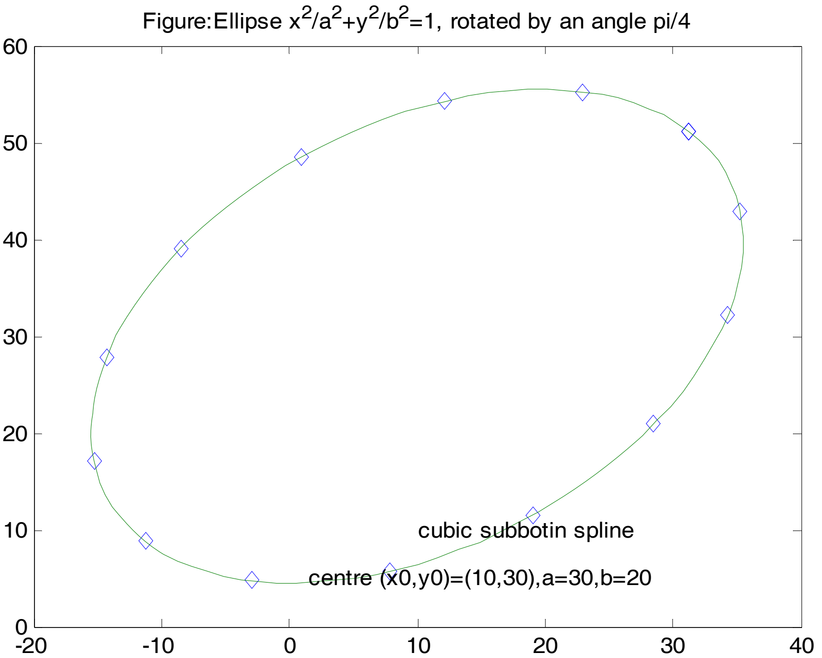


FIGURE-3



APPENDIX-1

```
function []=new_spline_int_cubic_subbotin(x,a,b,N)
%N=FUNCTION CODE,N=1:F(X)=1/(1+25*X^2),N=2:F(X)=(COS(X))^10
ETC
syms  s
n=length(x)-1;
T(1)=x(1);T(2*(n+1))=x(n+1);
for i=1:n
hi=x(i+1)-x(i);ii=2*i;
T(ii)=x(i)+hi/3;
T(ii+1)=T(ii)+hi/3;
h(i)=hi;
end

switch N
case 1
   disp('RUNGE FUNCTION')
fx=1./(1+25*x.^2);
dfe=[-50*a/(1+25*a^2)^2 -50*b/(1+25*b^2)^2];
   for i=1:n
      c(i)=a+i*h(i);
      II(i)=double(int(1/(1+25*s^2),a,a+i*h(i)));
```

```
    end

case 2
   fx=(cos(x)).^(10);
   dfe=[-10*sin(a)*(cos(a))^9 -10*sin(b)*(cos(b))^9];
case 3
    fx=log(x);
    dfe=[1/a 1/b]

case 4
    %fx=x.^(1/3);
    %dfe=[(1/3)*a^(-2/3) (1/3)*b^(-2/3)]
    fx=x.*log(x);%[1 3]
    dfe=[log(a)+1 log(b)+1];
case 5
%[0,4]
    disp('NORMAL DISTRIBUTION')
    fx=exp(-x.^2/2)/(sqrt(2*pi));
    dfe=[-a*exp(-a^2/2)/sqrt(2*pi) -b*exp(-b^2/2)/sqrt(2*pi)];
        for i=1:n
        c(i)=a+i*h(i);
        II(i)=double(int(1/sqrt(2*pi)*exp(-s^2/2),a,a+i*h(i)));
    end
case 6
    fx=1./((x-0.3).^2+.01)+1./((x-.9).^2+.04)-6;
    dfe=[-2*(a-.3)/(((a-.3)^2+.01)^2)-2*(a-.9)/(((a-.9)^2+.04)^2) -2*(b-
.3)/(((b-.3)^2+.01)^2)-2*(b-.9)/(((b-.9)^2+.04)^2)];
case 7
    fx=x.*sin(pi*x);%[0 1]
    dfe=[sin(pi*a)+pi*a*cos(pi*a) sin(pi*b)+pi*b*cos(pi*b)];
case 8
    fx=sin(pi*x)./(1+x.*x);
    dfe=[-2*a*sin(pi*a)/((1+a^2)^2)+pi*cos(pi*a)/(1+a^2) -
2*b*sin(pi*b)/((1+b^2)^2)+pi*cos(pi*b)/(1+b^2)];
case 9
    fx=exp(x);
    dfe=[exp(a) exp(b)];
case 10
    fx=3*x.*exp(x)-exp(2*x);
    dfe=[3*exp(a)*(a+1)-2*exp(2*a) 3*exp(b)*(b+1)-2*exp(2*b)];
case 11
    fx=1./(1+1./(1+exp(20*x)));
    dfe=[20*exp(20*a)/(2+exp(20*a))^2 20*exp(20*b)/(2+exp(20*b))^2];
case 12
    fx=(1+x).^(1/2);
    dfe=[.5/(1+a)^(1/2) .5/(1+b)^(1/2)];
```

```
case 13%[1,5]
    disp('logarithmic function')
    fx=1./(1+x);
    dfe=[-1/((1+a)^2) -1/((1+b)^2)];
        for i=1:n
        c(i)=a+i*h(i);
        II(i)=double(int(1/(1+s),a,a+i*h(i)));
    end

end
 pn=splint1(x,fx);
 %f'(x)=-50*x/(1+25x^2)^2,[f'(-1),f'(1)]=[50 -50]/26^2
 %dfe=[50 -50]/26^2;
 pc=splint1(x,fx,dfe);
 %error curves

% FT=1./(1+25*T.^2);
mm=length(T);
for i=1:mm
TT=T(i);
FT(i)=fspline(N,TT);
end
[p]=cubic_subbotin_new(T,h,FT);
format long
for j=1:n
    I(j)=0;
    I(j)=(p(j,1)+p(j,2)/2+p(j,3)/3+p(j,4)/4)*h(j);
end
IN(1)=I(1);
for k=2:n
    IN(k)=IN(k-1)+I(k);
end
table3(:,1)=c';
table3(:,2)=IN';
table3(:,3)=II';
disp('--------------------cubic_subbotin_spline----------------------------')

disp('---------x----------------IN--------------II------------------')
disp(table3)
disp('------------------------------------------------------------')
pause
for j=1:n
    J(j)=0;
    J(j)=(pn(j,1)+pn(j,2)/2+pn(j,3)/3+pn(j,4)/4)*h(j);
end
JN(1)=I(1);
```

```
for k=2:n
   JN(k)=JN(k-1)+J(k);
end

table4(:,1)=c';
table4(:,2)=JN';
table4(:,3)=II';
pause
disp('--------------------natural_spline-----------------------------')

disp('---------x----------------JN--------------II------------------')
disp(table4)
disp('--------------------------------------------------------------')

for j=1:n
   K(j)=0;
   K(j)=(pc(j,1)+pc(j,2)/2+pc(j,3)/3+pc(j,4)/4)*h(j);
end
KN(1)=I(1);
for k=2:n
   KN(k)=KN(k-1)+K(k);
end

table5(:,1)=c';
table5(:,2)=KN';
table5(:,3)=II';
pause
disp('--------------------clamped_spline-----------------------------')

disp('---------x----------------KN--------------II------------------')
disp(table5)
disp('--------------------------------------------------------------')
 pp=spline(x,fx);
 [breaks,coefs,npolys,ncoefs,dim]=unmkpp(pp);
 coefx=coefs;
 for m=1:n
L(m)=0;
  L(m)=(coefx(m,4)+coefx(m,3)*h(m)/2+coefx(m,2)*h(m)^2/3+coefx(m,1)*
h(m)^3/4)*h(m);
end
LN(1)=I(1);
for k=2:n
   LN(k)=LN(k-1)+L(k);
end
pause
table6(:,1)=c';
```

```
table6(:,2)=LN';
table6(:,3)=II';
disp('--------------------NOT A KNOT_spline----------------------------')

disp('---------x----------------LN--------------II-----------------')
disp(table6)
disp('----------------------------------------------------------')
```

APPENDIX-2

```
28function[]=new_greens_cubic_spline_integration_general_ellipse(div,n,xc,
yc,a,b,z)
ccc=0;
[s3,w3]=glsampleptsweights(n+3);
[s0,w0]=glsampleptsweights(n);
[A,B]=smooth_points_on_general_ellipse(div,xc,yc,a,b,z);
%disp('A=')
%disp(A)
%disp('B=')
%disp(B)

%A=[0.5;1/2+sqrt(1/8);1;1/2+sqrt(1/8);1/2;1/2-sqrt(1/8);0;sqrt(1/8);0.5]
%B=[0;1/2-sqrt(1/8);0.5;1/2+sqrt(1/8);1;1/2+sqrt(1/8);1/2;sqrt(1/8);0]
for m=25:30
switch m
    case 25
        disp('fn=1')
    case 26
         disp('fn=x')
     case 27
          disp('fn=y')
      case
           disp('fn=x^2')
       case 29
            disp('fn=y^2')
        case 30
             disp('fn=x*y')
otherwise
    disp('something wrong')
end
format long e
ccc=ccc+1;
    N=0;M=0;
%    t=0:4*div;
   % x(1:(4*div+1))=A(1:(4*div+1),1);y(1:(4*div+1))=B(1:(4*div+1),1);
```

```
 %[coefx]=cubicspline(t,x,t,3);
 %[coefy]=cubicspline(t,y,t,3);
 nn=(length(A)-1)/3;
 t=0:nn;
 i=1;j=1;
 x(1)=A(1,1);x(2*(nn+1))=A(4*div+1,1);
 y(1)=B(1,1);y(2*(nn+1))=B(4*div+1,1);
 for i=1:nn
     x(2*i)=A(j+1,1);
     x(2*i+1)=A(j+2,1);
     y(2*i)=B(j+1,1);
     y(2*i+1)=B(j+2,1);
     j=i+2*i+1;
 end
for i=1:nn
hi=(t(i+1)-t(i));
h(i)=hi;
end
% FT=1./(1+25*T.^2);
%mm=length(T)
%for i=1:mm
%TT=TX(i);
%FX(i)=fnxy(m,TT);
%end
[px]=new_cubic_subbotin(t,h,x);
[py]=new_cubic_subbotin(t,h,y);

for kk=1:nn
            N=N+1; M=M+1;II(M,1)=0;
  for p=1:(n+3)
 for q=1:n
  tp=s3(p);wp=w3(p);
  ttp=(tp+1)/2;
  tq=s0(q);wq=w0(q);
  ttq=(1+tq)/2;
  xt=px(N,1)+px(N,2)*ttp+px(N,3)*ttp^2+px(N,4)*ttp^3;
  yt=py(N,1)+py(N,2)*ttp+py(N,3)*ttp^2+py(N,4)*ttp^3;
  xx=(xt-xc)*tq/2+(xt+xc)/2;
  wt=(xt-xc)*(3*py(N,4)*ttp^2+2*py(N,3)*ttp+py(N,2))*wp*wq/4;
  II(M,1)=II(M,1)+wt*fnxy(m,xx,yt);
end%end for q
end%end for p
end%end for kk
iii(ccc,1)=0;
for mm=1:M
    iii(ccc,1)=iii(ccc,1)+II(mm,1);
```

```
end
%disp(II)
%disp('--------------------------------')
%disp('X=')
%disp(X)
%disp('--------------------------------')
%disp('Y=')
%disp(Y)
disp(iii(ccc,1))
end%for m
disp(iii)
m00=pi*a*b
m10=m00*xc
m01=m00*yc
m20=m00*((a^2*cos(z)^2+b^2*sin(z)^2)/4+xc^2)
m02=m00*((b^2*cos(z)^2+a^2*sin(z)^2)/4+yc^2)
m11=m00*((a^2-b^2)*cos(z)*sin(z)/4+xc*yc)
III(1,1)=abs(iii(1,1)-m00)*100/m00;
III(2,1)=abs(iii(2,1)-m10)*100/m10;
III(3,1)=abs(iii(3,1)-m01)*100/m01;
III(4,1)=abs(iii(4,1)-m20)*100/m20;
III(5,1)=abs(iii(5,1)-m02)*100/m02;
III(6,1)=abs(iii(6,1)-m11)*100/m11;
disp(III)
table(:,1)=iii;
table(:,2)=[m00;m10;m01;m20;m02;m11];
table(:,3)=III;
disp('                              TABLE')
disp('      computation of geometric moments for a general ellipse')
disp('--------------------------------------------------------------------------------------')
disp('divisions computed values        exact values          percentage error')
disp('--------------------------------------------------------------------------------------')
disp(div)
disp(table)
disp('--------------------------------------------------------------------------------------')
```

Received: January, 2010