Differentiation and Numerical Integral of the Cubic Spline Interpolation

Shang Gao
School of Computer Science and Technology, Jiangsu University of Science and Technology, Zhenjiang 212003, China
Email: gao_shang@hotmail.com

Zaiyue Zhang and Cungen Cao
Key Laboratory of Intelligent Information Processing, Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100080, China
Email: yzzjzzy@sina.com, cgcao@ict.ac.cn

***Abstract*—Based on analysis of cubic spline interpolation, the differentiation formulas of the cubic spline interpolation on the three boundary conditions are put up forward in this paper. At last, this calculation method is illustrated through an example. The numerical results show that the spline numerical differentiations are quite effective for estimating first and higher derivatives of equally and unequally spaced data. The formulas based on cubic spline interpolation solving numerical integral of discrete function are deduced. The degree of integral formula is n=3.The formulas has high accuracy. At last, these calculation methods are illustrated through examples.**

***Index Terms*—cubic spline function, numerical differentiation, numerical integral, first derivative, second derivative**

I. Introduction

Splines and particularly cubic splines are very popular models for interpolation. Historically, a ``spline" was a common drafting tool, a flexible rod, that was used to help draw smooth curves connecting widely spaced points. The cubic spline curve accomplishes the same result for an interpolation problem. The spline technology has applications in CAD, CAM, and computer graphics systems. We describe cubic splines in this note and discuss their use in interpolation and curve fitting.. The cubic spline interpolation is a piecewise continuous curve, passing through each of the values in the table. There is a separate cubic polynomial for each interval, each with its own coefficients. The first derivative and the second derivative of a cubic spline are continuous. For the approximation of gradients from data values at vertices of a uniform grid, P. Sablonnière[1] compare two methods based on cubic spline interpolation with a classical method based on finite differences. For univariate cubic splines, p. Sablonnière use the so-called de Boor's Not a Knot property and a new method giving pretty good slopes. J. S. Behar, S. J. Estrada and M. V. Hernández [2] have developed a G 2-continuous cubic A-spline scheme smoothing the polygon defined by the line segments joining consecutive data points, such that the spline curve lies completely on the same side of the boundary polygon as the data. The proposed A-spline scheme provides an efficient method for generating a smooth robot's path that avoids corners or polygonal objects for a given planned path, for designing a smooth curve on a polygonal piece of material, etc. Petrinovic, Davor [3] presents two formulations of causal cubic splines with equidistant knots. Both are based on a causal direct B-spline filter with parallel or cascade implementation. In either implementation, the causal part of the impulse response is realized with an efficient infinite-impulse-response (IIR) structure, while only the anticausal part is approximated with a finite-order finite-impulse-response (FIR) filter.

Formulas for numerical derivatives are important in developing algorithms for solving boundary value problems for ordinary differential equations and partial differential equations. Numerical differentiation is a technique of numerical analysis to produce an estimate of the derivative of a mathematical function or function subroutine using values from the function and perhaps other knowledge about the function. Numerical differentiation is the process of finding the numerical value of a derivative of a given function at a given point. In general, numerical differentiation is more difficult than numerical integration. This is because while numerical integration requires only good continuity properties of the function being integrated, numerical differentiation requires more complicated properties such as Lipschitz classes. The differentiation formulas of the cubic spline interpolation on the three boundary conditions are put up forward in this paper.

Numerical integration is concerned with developing algorithms to approximate the integral of a function f(x). The most commonly used algorithms are Newton-Cotes formulas, Romberg's method, Gaussian quadrature, and to lesser extents Hermite's formulas and certain adaptive techniques. We got interpolatory quadrature formulas with equidistance knots using three types of cubic spline for oscillating integral, estimated the error, and a numerical example was given to illustrate the high accuracy of our method as wel1.

doi:10.4304/jcp.6.10.2037-2044

II. CUBIC SPLINE INTERPOLATION

Suppose that $\{(x_i, y_i)\}_{i=0}^{N}$ are $N+1$ points, where $a = x_0, x_1, \cdots, x_n = b$. The function $S(x)$ is called a cubic spline if there exit N cubic polynomials and satisfy the properties:

I. $S(x)$ is a cubic polynomial on $[x_{i-1}, x_i]$ ($i = 1,2,\cdots,n$)

II. $S(x_i^-) = S(x_i^+)$ $(i = 1,2,\cdots,n-1)$

III. $S'(x_i^-) = S'(x_i^+)$ $(i = 1,2,\cdots,n-1)$

IV. $S''(x_i^-) = S''(x_i^+)$ $(i = 1,2,\cdots,n-1)$

V. $S(x_i) = y_i (i = 0,1,\cdots,n)$.

Each cubic polynomial has four unknown constants, hence there are $4N$ coefficients to be determined. The data points supply $N+1$ conditions, and properties II, III and IV each supply $N-1$ conditions. Hence, $N+1+3(N-1) = 4N-2$ conditions are specified. This leaves us two additional degrees of freedom. The choice of these two extra conditions determines the type of the cubic spline obtained.

One of the following sets of boundary conditions is satisfied [4][5][6][7]:

(i) Clamped spline: $S'(x_0) = y_0^{'}$, $S'(x_n) = y_n^{'}$

(ii) Curvature-adjusted cubic spline: $S^{''}(x_0) = y_0^{''}$, $S^{''}(x_n) = y_n^{''}$

(iii) Periodic spline: $S(x_0) = S(x_n)$, $S'(x_0) = S'(x_n)$, $S''(x_0) = S''(x_n)$.

For the clamped boundary conditions, we use $m_i = S'(x_i)(i = 0,1,\cdots,n)$. Specify $m_0 = y_0^{'}$, $m_n = y_n^{'}$, we obtain $N-1$ linear equations involving the coefficients $m_1, m_2, \cdots, m_{n-1}$.

$$\begin{bmatrix} 2 & \alpha_1 & & & & \\ 1-\alpha_2 & 2 & \alpha_2 & & & \\ & 1-\alpha_3 & 2 & \alpha_3 & & \\ & & \ddots & \ddots & \ddots & \\ & & & 1-\alpha_{n-2} & 2 & \alpha_{n-2} \\ & & & & 1-\alpha_{n-1} & 2 \end{bmatrix} \begin{bmatrix} m_1 \\ m_2 \\ m_3 \\ \vdots \\ \\ m_{n-1} \end{bmatrix} = \begin{bmatrix} \beta_1 - (1-\alpha_1)m_0 \\ \beta_2 \\ \beta_3 \\ \vdots \\ \beta_{n-2} \\ \beta_{n-1} - \alpha_{n-1}m_n \end{bmatrix}$$

where $h_{i-1} = x_i - x_{i-1}$, $\alpha_i = \dfrac{h_{i-1}}{h_{i-1} + h_i}$,

$$\beta_i = 3[(1-\alpha_i)\frac{y_i - y_{i-1}}{h_{i-1}} + \alpha_i \frac{y_{i+1} - y_i}{h_i}](i = 1,2,\cdots,n-1)$$

This is a tridiagonal linear system. The $m_1, m_2, \cdots, m_{n-1}$ are obtain by Crout Factorization algorithm. The result is the following expression for the cubic function $S_i(x)$ on $[x_{i-1}, x_i](i = 1,2,\cdots,n)$.

$$S_i(x) = \varphi_0(\frac{x - x_{i-1}}{h_{i-1}})y_{i-1} + \varphi_1(\frac{x - x_{i-1}}{h_{i-1}})y_i + h_{i-1}\psi_0(\frac{x - x_{i-1}}{h_{i-1}})m_{i-1} + h_{i-1}\psi_1(\frac{x - x_{i-1}}{h_{i-1}})m_i \quad (1)$$

where $\varphi_0(x) = (x-1)^2(2x-1)$, $\varphi_1(x) = x^2(-2x+3)$, $\psi_0(x) = x(x-1)^2$ and $\psi_1(x) = x^2(x-1)$.

III. DIFFERENTIATION OF THE CUBIC SPLINE INTERPOLATION

A. Finite difference formulae

The simplest method is to use finite difference approximations. A simple two-point estimation is to compute the slope of a nearby secant line through the points (*x*,*f*(*x*)) and (*x*+*h*,*f*(*x*+*h*)). Choosing a small number *h*, *h* represents a small change in *x*, and it can be either positive or negative. The slope of this line is

$$\frac{f(x+h) - f(x)}{h}$$

This expression is Newton's difference quotient.

The slope of this secant line differs from the slope of the tangent line by an amount that is approximately proportional to h. As h approaches zero, the slope of the secant line approaches the slope of the tangent line. Therefore, the true derivative of f at *x* is the limit of the value of the difference quotient as the secant lines get closer and closer to being a tangent line:

$$f'(x) = \lim_{h \to 0} \frac{f(x+h) - f(x)}{h}$$

Since immediately substituting 0 for h results in division by zero, calculating the derivative directly can be unintuitive.

A simple three-point estimation is to compute the slope of a nearby secant line through the points (*x*-*h*,*f*(*x*-*h*)) and (*x*+*h*,*f*(*x*+*h*)). The slope of this line is

$$\frac{f(x+h) - f(x-h)}{2h}$$

B. Differentlation of cubic spline

If the nodes are equally spaced, that is, when $x_1 = x_0 + h$, $x_2 = x_0 + 2h$, the formulas (1) can be expressed as

$$S(x) = \varphi_0(\frac{x - x_{i-1}}{h})y_{i-1} + \varphi_1(\frac{x - x_{i-1}}{h})y_i + h\psi_0(\frac{x - x_{i-1}}{h})m_{i-1} + h\psi_1(\frac{x - x_{i-1}}{h})m_i \quad (2)$$

From Eq.(2),

$$S''(x) = \frac{6}{h^2}[2(\frac{x-x_{i-1}}{h})-1]y_{i-1} - \frac{6}{h^2}[2(\frac{x-x_{i-1}}{h})-1]y_{i+1} \quad (3)$$
$$+\frac{1}{h}[6(\frac{x-x_{i-1}}{h})-4]m_{i-1} + \frac{1}{h}[6(\frac{x-x_{i-1}}{h})-2]m_{i+1}$$

Use $S''(x_1^-) = S''(x_1^+)$ in (3) to get

$$m_0 + 4m_1 + m_2 = \frac{3}{h}(y_2 - y_0) \quad (4)$$

Substitution of x_0 , $x_1 = x_0 + h$, $x_2 = x_0 + 2h$ in equation (3) produces the solution

$$S''(x_0) = \frac{6(y_1-y_0)}{h^2} - \frac{4m_0}{h} - \frac{2m_1}{h} \quad (5)$$

$$S''(x_1) = \frac{6(y_2-y_1)}{h^2} - \frac{4m_1}{h} - \frac{2m_2}{h} \quad (6)$$

$$S''(x_2) = \frac{6(y_1-y_2)}{h^2} + \frac{4m_2}{h} + \frac{2m_1}{h} \quad (7)$$

C. Clamped spline

The $m_0 = y_0'$ and $m_2 = y_2'$ are known.

From Eq.(4),

$$y_1' = \frac{3}{4h}(y_2 - y_0) - \frac{1}{4}y_0' - \frac{1}{4}y_2' \quad (8)$$

Substituting y_1' into equation (5),(6) and (7) respectively, the solutions are

$$y_0'' = \frac{-9y_0 + 12y_1 - 3y_2}{2h^2} - \frac{7y_0'}{2h} + \frac{y_2'}{2h} \quad (9)$$

$$y_1'' = \frac{3y_0 - 6y_1 + 3y_2}{2h^2} + \frac{y_0'}{h} - \frac{y_2'}{h} \quad (10)$$

$$y_2'' = \frac{-3y_0 + 12y_1 - 9y_2}{2h^2} - \frac{y_0'}{2h} + \frac{7y_2'}{2h} \quad (11)$$

The solution of differentiation of the Clamped Spline is shown in table 1.

TABLE I.
THE SOLUTION OF DIFFERENTIATION OF THE CLAMPED

x_i	x_0	x_1	x_2
y_i	y_0	y_1	y_2
y_i'	y_0'	(8)	y_2'
y_i''	(9)	(10)	(11)

D. Curvature-adjusted cubic spline

The $S''(x_0) = y_0''$ and $S''(x_n) = y_n''$ are known.

Substituting $S''(x_0) = y_0'', S''(x_n) = y_n''$ into equation (5), (7) and (4), we obtain the linear system

$$\begin{cases} 4m_0 + 2m_1 = \dfrac{6(y_1-y_0)}{h} - hy_0'' \\ m_0 + 4m_1 + m_2 = \dfrac{3(y_2-y_0)}{h} \\ 2m_1 + 4m_2 = \dfrac{6(y_2-y_1)}{h} + hy_2'' \end{cases}$$

The solution are

$$\begin{cases} y_0' = \dfrac{-5y_0 + 6y_1 - y_2}{4h} - \dfrac{7}{24}hy_0'' + \dfrac{1}{24}hy_2'' \\ y_1' = \dfrac{y_2 - y_0}{2h} + \dfrac{1}{12}hy_0'' - \dfrac{1}{12}hy_2'' \\ y_2' = \dfrac{y_0 - 6y_1 + 5y_2}{4h} - \dfrac{1}{24}hy_0'' + \dfrac{7}{24}hy_2'' \end{cases} \quad (12)$$

The values y_0', y_1', y_2' are substituted in (6), the solution is

$$y_1'' = \frac{3}{2h^2}(y_0 - 2y_1 + y_2) - \frac{1}{4}y_0'' - \frac{1}{4}y_2'' \quad (13)$$

The solution of differentiation of the Curvature-adjusted cubic spline is shown in table 2.

TABLE II.
T THE SOLUTION OF DIFFERENTIATION OF THE CURVATURE-ADJUSTED CUBIC SPLINE

x_i	x_0	x_1	x_2
y_i	y_0	y_1	y_2
y_i'	(12)	(12)	(12)
y_i''	y_0''	(13)	y_2''

E. Periodic spline

When $S(x_0^+) = S(x_2^-)$, $S'(x_0^+) = S'(x_2^-)$, and $S''(x_0^+) = S''(x_2^-)$, using $S(x_0^+) = S(x_2^-)$, the solution is $y_0 = y_2$.Using $S'(x_0^+) = S'(x_2^-)$, we find $m_0 = m_2$.

Substituting $S''(x_0^+) = S''(x_2^-)$ into Eq.(5) and Eq.(7) we get

$$m_0 + m_1 + m_2 = \frac{3}{2h}(y_2 - y_0) = 0$$

Using Eq.(4), we obtain the linear system

$$\begin{cases} m_0 = m_2 \\ m_0 + 4m_1 + m_2 = 0 \\ m_0 + m_1 + m_2 = 0 \end{cases}$$

The solution are

$$\begin{cases} y_0' = m_0 = 0 \\ y_1' = m_1 = 0 \\ y_2' = m_2 = 0 \end{cases} \quad (14)$$

The values y_0', y_1', y_2' are substituted in (5), (6) and (7) respectively, the solution is

$$y_0'' = y_2'' = \frac{6(y_1 - y_0)}{h^2} \quad (15)$$

$$y_1'' = \frac{6(y_0 - y_1)}{h^2} \quad (16)$$

The solution of differentiation of the periodic spline is shown in table 3.

TABLE III.
THE SOLUTION OF DIFFERENTIATION OF THE PERIODIC SPLINE

x_i	x_0	x_1	x_2
y_i	y_0	y_1	y_2
$y_i^{'}$	0	0	0
$y_i^{''}$	$\frac{6(y_1 - y_0)}{h^2}$	$\frac{6(y_0 - y_1)}{h^2}$	$\frac{6(y_1 - y_0)}{h^2}$

IV. DIFFERENTIATION EXAMPLES

Example 1: Clamped spline: Find differentlation for the points (0,1), (1,2), (2,3), where $y'(0) = 0$ and $y'(2) = -6$.

It is a straightforward task to compute the solution $y_1^{'} = 3$, $y_0^{''} = 0$, $y_1^{''} = 6$ and $y_2^{''} = -24$.

The cubic spline is

$$S(x) = \begin{cases} x^3 + 1 & 0 \le x \le 1 \\ -5x^3 + 18x^2 - 18x + 7 & 1 < x \le 2 \end{cases}$$

The cubic spline is shown in figure 1.

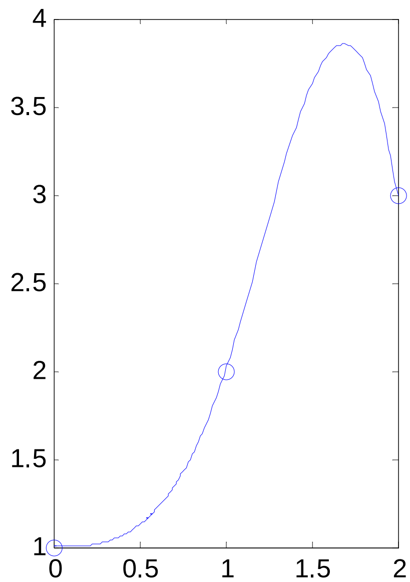

Fig. 1: The cubic spline of Example 1

Example 2: Curvature-adjusted cubic spline: Find differentlation for the points (0,1), (1,0), (2,-5), where $y''(0) = 0$ and $y''(2) = 0$.

The solution is $y_0^{'} = 0$, $y_1^{'} = -3$, $y_2^{'} = -6$ and $y_1^{''} = -6$.

The cubic spline is

$$S(x) = \begin{cases} -x^3 + 1 & 0 \le x \le 1 \\ x^3 - 6x^2 + 6x - 1 & 1 < x \le 2 \end{cases}$$

The cubic spline is shown in figure 2.

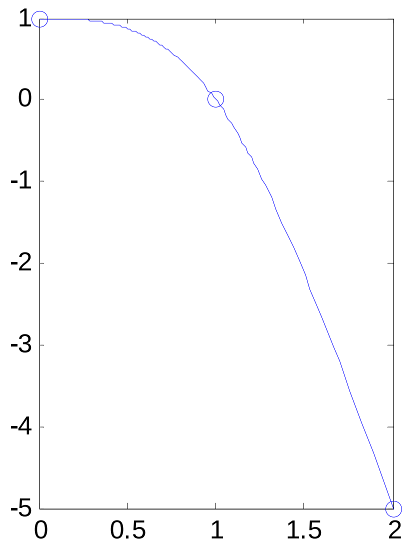

Fig. 2: The cubic spline of Example 2

Example 3: Periodic spline: Find differentlation for the points (0,2), (1,3), (2,2).

The solution is $y_0^{'} = y_1^{'} = y_2^{'} = 0$, $y_0^{''} = y_2^{''} = 6$ and $y_1^{''} = -6$.

The cubic spline is

$$S(x) = \begin{cases} -2x^3 + 3x^2 + 2 & 0 \le x \le 1 \\ 2x^3 - 9x^2 + 12x - 2 & 1 < x \le 2 \end{cases}$$

The cubic spline is shown in figure 3.

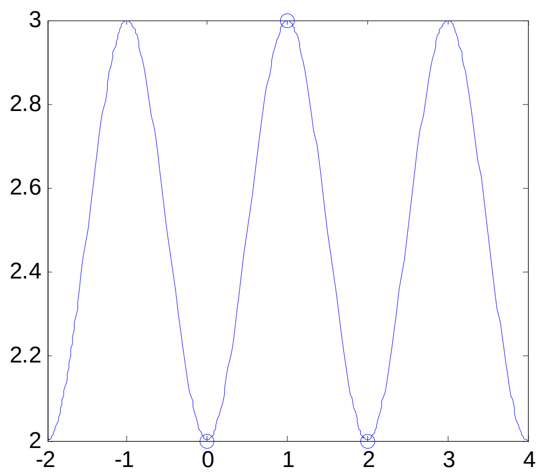

Fig. 3: The cubic spline of Example3

V. NUMERICAL INTEGRAL OF THE CUBIC SPLINE INTERPOLATION

A. Quadrature rules based on interpolating functions

There are several reasons for carrying out numerical integration. The integrand *f*(x) may be known only at certain points, such as obtained by sampling. Some embedded systems and other computer applications may need numerical integration for this reason.

A large class of quadrature rules can be derived by constructing interpolating functions which are easy to integrate. Typically these interpolating functions are polynomials.

The simplest method of this type is to let the interpolating function be a constant function (a polynomial of degree zero) which passes through the point ((*a*+*b*)/2, *f*((*a*+*b*)/2)). This is called the midpoint rule or rectangle rule.

$$\int_a^b f(x)dx \approx (b-a)f(\frac{a+b}{2})$$

The interpolating function may be an affine function (a polynomial of degree 1) which passes through the points (a, f(a)) and (b, f(b)). This is called the trapezoidal rule.

$$\int_a^b f(x)dx \approx (b-a)\frac{f(a)+f(b)}{2}$$

In numerical analysis, Simpson's rule is a method for numerical integration, the numerical approximation of definite integrals. Specifically, it is the following approximation:

$$\int_a^b f(x)dx \approx \frac{b-a}{6}[f(a)+4f(\frac{a+b}{2})+f(b)]$$

B. Clamped spline

For the clamped boundary conditions, we use $m_i = S'(x_i)(i=0,1,\cdots,n)$. Specify $m_0 = y_0^{'}$, $m_n = y_n^{'}$, we obtain $N-1$ linear equations involving the coefficients $m_1, m_2, \cdots, m_{n-1}$.

$$\begin{bmatrix} 2 & \alpha_1 & & & & \\ 1-\alpha_2 & 2 & \alpha_2 & & & \\ & 1-\alpha_3 & 2 & \alpha_3 & & \\ & & \ddots & \ddots & \ddots & \\ & & & 1-\alpha_{n-2} & 2 & \alpha_{n-2} \\ & & & & 1-\alpha_{n-1} & 2 \end{bmatrix} \quad (17)$$

$$\begin{bmatrix} m_1 \\ m_2 \\ m_3 \\ \vdots \\ \\ m_{n-1} \end{bmatrix} = \begin{bmatrix} \beta_1 - (1-\alpha_1)m_0 \\ \beta_2 \\ \beta_3 \\ \vdots \\ \beta_{n-2} \\ \beta_{n-1} - \alpha_{n-1}m_n \end{bmatrix}$$

where $h_{i-1} = x_i - x_{i-1}$, $\alpha_i = \frac{h_{i-1}}{h_{i-1}+h_i}$,

$$\beta_i = 3[(1-\alpha_i)\frac{y_i - y_{i-1}}{h_{i-1}} + \alpha_i \frac{y_{i+1}-y_i}{h_i}]$$

$(i=1,2,\cdots,n-1)$.

This is a tridiagonal linear system. The $m_1, m_2, \cdots, m_{n-1}$ are obtain by Crout Factorization algorithm. The result is the following expression for the cubic function $S_i(x)$ on $[x_{i-1}, x_i](i=1,2,\cdots,n)$.

$$S_i(x) = \varphi_0(\frac{x-x_{i-1}}{h_{i-1}})y_{i-1} + \varphi_1(\frac{x-x_{i-1}}{h_{i-1}})y_i + h_{i-1}\psi_0(\frac{x-x_{i-1}}{h_{i-1}})m_{i-1} + h_{i-1}\psi_1(\frac{x-x_{i-1}}{h_{i-1}})m_i \quad (18)$$

where $\varphi_0(x) = (x-1)^2(2x+1)$, $\varphi_1(x) = x^2(-2x+3)$, $\psi_0(x) = x(x-1)^2$ and $\psi_1(x) = x^2(x-1)$.

$$\int_a^b f(x)dx \approx \int_a^b S(x)dx = \sum_{i=0}^{n-1}\int_{x_i}^{x_{i+1}} S_i(x)dx$$

$$= \sum_{i=0}^{n-1}[\frac{h_{i-1}}{2}(y_{i-1}+y_i) + \frac{h_{i-1}^2}{12}(m_{i-1}-m_i)]$$

If $h_0 = h_1 = h_2 = \cdots = h_{n-1} = h$, then

$$\int_a^b f(x)dx \approx \frac{h}{2}(y_0 + 2y_1 + \cdots + 2y_{n-1} + y_n) + \frac{h^2}{12}(y_0^{'} - y_n^{'}) \quad (19)$$

where $h = \frac{b-a}{n}$ and $x_k = a + kh$.

It will suffice to apply formula (19) over interval [a,b] with the five test functions $f(x) = 1, x, x^2, x^3$ and x^4. For the first four functions, formula (19) is exact.

For $f(x) = 1$, $\int_a^b 1dx = b-a$

$$\frac{h}{2}(y_0 + 2y_1 + \cdots + 2y_{n-1} + y_n) + \frac{h^2}{12}(y_0^{'} - y_n^{'})$$

$$= \frac{h}{2}(1+2+\cdots+2+1) = nh = b-a$$

For $f(x) = x$,

$$\int_a^b xdx = \frac{b^2-a^2}{2} = \frac{1}{2}(n^2h^2 + 2nah)$$

$$\frac{h}{2}(y_0 + 2y_1 + \cdots + 2y_{n-1} + y_n) + \frac{h^2}{12}(y_0^{'} - y_n^{'})$$

$$= \frac{h}{2}(a + 2x_1 + \cdots + 2x_{n-1} + b)$$

$$= \frac{h}{2}\{a + 2(a+h) + 2(a+2h)\cdots$$

$$+ 2[a+(n-1)]h + a + nh\}$$

$$= \frac{1}{2}(n^2h^2 + 2nah)$$

For $f(x) = x^2$,

$$\int_a^b x^2dx = \frac{b^3-a^3}{3} = \frac{1}{3}n^3h^3 + na^2h + an^2h^2$$

$$\frac{h}{2}(y_0+2y_1+\cdots+2y_{n-1}+y_n)+\frac{h^2}{12}(y_0'-y_n')$$
$$=\frac{h}{2}(a^2+2x_1^2+\cdots+2x_{n-1}^2+b^2)+\frac{h^2}{12}(2a-2b)$$
$$=\frac{1}{3}n^3h^3+na^2h+an^2h^2$$

For $f(x)=x^3$,

$$\int_a^b x^3dx=\frac{b^4-a^4}{4}=\frac{3}{2}a^2n^2h^2+na^3h+an^3h^3+\frac{1}{4}n^4h^4$$

$$\frac{h}{2}(y_0+2y_1+\cdots+2y_{n-1}+y_n)+\frac{h^2}{12}(y_0'-y_n')$$
$$=\frac{h}{2}(a^3+2x_1^3+\cdots+2x_{n-1}^3+b^3)+\frac{h^2}{12}(3a^2-3b^2)$$
$$=\frac{3}{2}a^2n^2h^2+na^3h+an^3h^3+\frac{1}{4}n^4h^4$$

For $f(x)=x^4$, $\int_a^b x4dx=\frac{b^5-a^5}{5}$

$$\frac{h}{2}(y_0+2y_1+\cdots+2y_{n-1}+y_n)+\frac{h^2}{12}(y_0'-y_n')$$
$$=\frac{h}{2}(a^4+2x_1^4+\cdots+2x_{n-1}^4+b^4)+\frac{h^2}{12}(4a^3-4b^3)$$
$$\neq\frac{b^5-a^5}{5}$$

Therefore, the degree of formula (19) is $n=3$.

C. Curvature-adjusted cubic spline

When $S''(x_0)=y_0''$, $S''(x_n)=y_n''$ are known, we obtain $N-1$ linear equations involving the coefficients $m_0,m_1,\cdots,m_n$.

$$\begin{bmatrix} 2 & 1 & & & & \\ 1-\alpha_2 & 2 & \alpha_2 & & & \\ & 1-\alpha_3 & 2 & \alpha_3 & & \\ & & \ddots & \ddots & \ddots & \\ & & & 1-\alpha_{n-1} & 2 & \alpha_{n-1} \\ & & & & 1 & 2 \end{bmatrix} \begin{bmatrix} m_0 \\ m_1 \\ m_2 \\ \vdots \\ \\ m_n \end{bmatrix} = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \vdots \\ \beta_{n-1} \\ \beta_n \end{bmatrix} \quad (20)$$

Where $h_{i-1}=x_i-x_{i-1}(i=1,2,\cdots,n)$,

$\alpha_i=\frac{h_{i-1}}{h_{i-1}+h_i}(i=1,2,\cdots,n-1)$,

$\beta_i=3[(1-\alpha_i)\frac{y_i-y_{i-1}}{h_{i-1}}+\alpha_i\frac{y_{i+1}-y_i}{h_i}]$, $(i=1,2,\cdots,n-1)$

$$\beta_0=3\frac{y_1-y_0}{h_1}-\frac{h_1}{2}y_0'',$$
$$\beta_n=3\frac{y_n-y_{n-1}}{h_n}+\frac{h_n}{2}y_n''.$$

From equations (20), we can get m_0,m_n, the value m_0 and m_n are substituted into equations (19), then obtain the integral.

D. Periodic spline

The condition of periodic spline are $S(x_0)=S(x_n)$, $S'(x_0)=S'(x_n)$ and $S''(x_0)=S''(x_n)$. Then numerical integral formula is

$$\int_a^b f(x)dx\approx\frac{h}{2}(y_0+2y_1+\cdots+2y_{n-1}+y_n) \quad (21)$$

It is compound trapezoid formula.

VI NUMERICAL INTEGRAL EXAMPLES

Example 4: Find integral for the points (-1,14), (0,-3), (1,8), (2,24), (3,34), (4,32), and (5,16), where $f'(-1)=-31$ and $f'(5)=-19$.

Using equation (19), the numerical integral

$$\int_{-1}^5 f(x)dx\approx\frac{1}{2}(14-6+16+48+68+64+16)+\frac{1}{12}(-31+19)=75$$

The cubic spline is

$$S(x)=\begin{cases} 2x^3+18x^2-x-3 & -1\le x\le 0 \\ -6x^3+18x^2-x-3 & 0<x\le 1 \\ -x^3+3x^2+14x-8 & 1<x\le 4 \\ 3x^3-45x^2+206x-264 & 4<x\le 5 \end{cases}$$

The cubic spline is shown in figure 4.

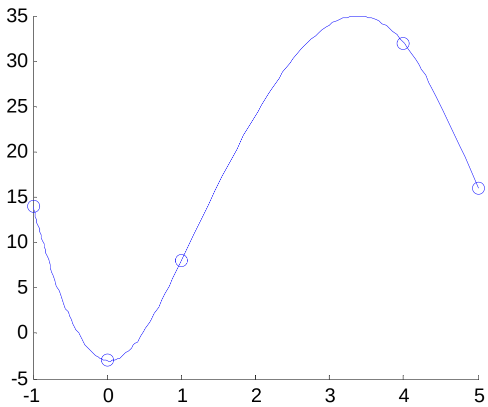

Fig. 4: The cubic spline of Example 4

Example 5: Find the integral for the points (0,0), (1,1), (2,9), (3,34), and (4,84), where $f''(0)=0$ and $f''(4)=36$.

Using equation (20), we obtain

$$\begin{bmatrix} 2 & 1 & & & & \\ \frac{1}{2} & 2 & \frac{1}{2} & & & \\ & \frac{1}{2} & 2 & \frac{1}{2} & & \\ & & \frac{1}{2} & 2 & \frac{1}{2} & \\ & & & \frac{1}{2} & 2 & \frac{1}{2} \\ & & & & 1 & 2 \end{bmatrix} \begin{bmatrix} m_0 \\ m_1 \\ m_2 \\ m_3 \\ m_n \end{bmatrix} = \begin{bmatrix} 3 \\ \frac{27}{2} \\ \frac{99}{2} \\ \frac{225}{2} \\ 168 \end{bmatrix}$$

The solution is $y_0' = 0$, $y_4' = 66$.

$$\int_0^4 f(x)dx \approx \frac{1}{2}(0+2+18+68+84)$$
$$+\frac{1}{12}(0-66) = \frac{161}{2}$$

The cubic spline is

$$S(x) = \begin{cases} x^3 & 0 \le x \le 1 \\ 2x^3 - 3x^2 + 3x - 1 & 1 < x \le 2 \\ x^3 + 3x^2 - 9x + 7 & 2 < x \le 3 \\ 2x^3 - 6x^2 + 18x - 20 & 3 < x \le 4 \end{cases}$$

The cubic spline is shown in figure 5.

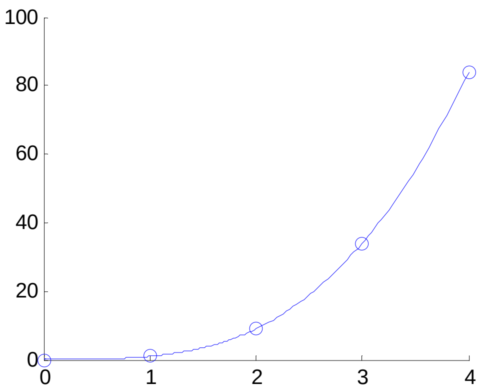

Fig. 5: The cubic spline of Example 5

Example 6: Find the integral for the points (0,4), (1,5), (2,6), (3,5), and (4,4).

Using equation (21), the numerical integral

$$\int_0^4 f(x)dx \approx \frac{1}{2}(4+10+12+10+4) = 20$$

The cubic spline is

$$S(x) = \begin{cases} -\frac{1}{2}x^3 + \frac{3}{2}x^2 + 4 & 0 \le x \le 2 \\ \frac{1}{2}x^3 - \frac{9}{2}x^2 + 12x - 4 & 2 < x \le 4 \end{cases}$$

The cubic spline is shown in figure 6.

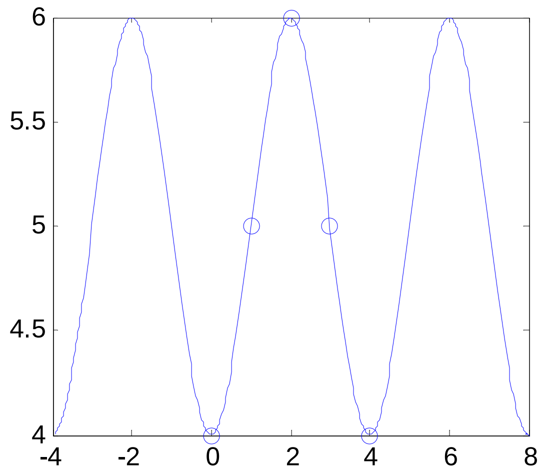

Fig. 6: The cubic spline of Example 5

VII. Conclusions

Cubic splines are popular because they are easy to implement and produce a curve that appears to be seamless. As we have seen, a straight polynomial interpolation of evenly spaced data tends to build in distortions near the edges of the table. Cubic splines avoid this problem, but they are only piecewise continuous, meaning that a sufficiently high derivative (third) is discontinuous. So if the application is sensitive to the smoothness of derivatives higher than second, cubic splines may not be the best choice. The formulas solving numerical differential and integral of discrete function are deduced in this paper. By numerical simulation the practicability and effectiveness are verified. The three-point spline numerical differentiation formulas are given in this paper. Similarly, the other points spline numerical differentiation formulas can also computed.

Acknowledgment

This work was partially supported by National Basic Research Program of Jiangsu Province University (08KJB520003) and the National Natural Science Foundation of China under Grant No.60773059.

References

[1] P. Sablonnière. Gradient approximation on uniform meshes by finite differences and cubic spline interpolation. Lecture Notes in Computer Science (including subseries Lecture

Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), v 5654 LNCS, p 322-334, 2009.

[2] J. S. Behar, S. J. Estrada, M. V. Hernández. Constrained interpolation with implicit plane cubic a-splines. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), v 5197 LNCS, p 724-732, 2008.

[3] Petrinovic, Davor. Causal cubic splines: Formulations, interpolation properties and implementations. IEEE Transactions on Signal Processing, v 56, n 11, p 5442-5453, 2008.

[4] R. L. Burden, J.D. Faires. Numerical Analysis. Higher Education Press & Thomson Learning, Inc., 2001,pp.141-150, 408-409.

[5] J. H. Mathews, K.D. Fink. Numerical Methods Using MATLAB. Publishing House of Electronics Industry, 2002, pp.280-290.

[6] M. I. Syam. Cubic spline interpolation predictors over implicitly defined curves. Journal of Computational and Applied Mathematics, 2003, 157(2), pp.283-295.

[7] T. L. Tsai, I. Y. Chen. Investigation of effect of endpoint constraint on time-line cubic spline interpolation. Journal of Mechanics, v 25, n 2, p 151-160, June 2009.

Shang Gao was born in 1972, and received his M.S. degree in 1996 and Ph.D degree in 2006. He now works in school of computer science and technology, Jiangsu University of Science and Technology. He is a professor and He is engage mainly in systems engineering and soft computing.

Zaiyue Zhang was born in 1961, and received his M.S. degree in mathematics in 1991 from the Department of Mathematics, Yangzhou Teaching College, and the Ph.D. degree in mathematics in 1995 from the Institute of Software, the Chinese Academy of Sciences. His current research areas are recursion theory, knowledge representation and knowledge reasoning.

Cungen Cao was born in 1964, and received his M.S. degree in 1989 and Ph.D. degree in 1993 both in mathematics from the Institute of Mathematics, the Chinese Academy of Sciences. Now he is a professor of the Institute of Computing Technology, the Chinese Academy of Sciences. His research area is large scale knowledge processing.