

A New Unified Scheme Computing the Quadrature Weights, Integration and Differentiation Matrix for the Spectral Method

Chang-Joo Kim*, Joon-Goo Park* and Sangkyung Sung[†]

Abstract – A unified numerical method for computing the quadrature weights, integration matrix, and differentiation matrix is newly developed in this study. For this purpose, a spline-like interpolation using piecewise continuous polynomials is converted into a global spline interpolation formula, with which the quadrature formulas can be derived from integration and differentiation of the transformed function in an exact manner. To prove the usefulness of the suggested approach, both the Lagrange and tension spline interpolations are represented in exactly the same form as global spline interpolation. The applicability of the proposed method on arbitrary nodes is illustrated using two different sets of nodes. A series of validations using three test functions is conducted to show the flexibility in selecting computational nodes with the present method.

Keywords: Computational nodes, Quadrature formula, Spline interpolation, Spectral method

1. Introduction

Numerical methods for system analysis and design optimization are widely studied in many engineering problem, where much effort is paid on selecting node or boundary conditions in each applications [1-3]. In consideration of both efficiency and accuracy, this paper proposes a new unified numerical scheme in the computation of the quadrature weights, integration matrix, and differentiation matrix for spectral methods, collectively referred to in this paper as the coefficients for quadrature formulas (CQFs). The Gauss-type formulas are frequently used in solving the differential and integro-differential equations arising in many engineering and scientific areas, since they can provide extreme accuracy and high convergence at an exponential rate to iterative analyses for problems with well-behaved solutions [4-7]. CQFs for spectral methods are generally derived using the orthogonal properties among interpolating polynomials. Many of them, expressed in an analytic form, are easily accessible in the literature [8-10]. Derivation of CQFs generally requires rigorous application of complex function theories and is strictly dependent on the types of interpolating functions and quadrature nodes, which may cause inconvenience when they are required for other interpolating functions and quadrature nodes different from the standard ones.

This work proposes a new numerical approach in the computation of CQFs that requires only information on the computational nodes and can retain the same level of

accuracy as traditional methods. For this purpose, the spline interpolation approach utilizing the piecewise continuous polynomials is used, rather than the global Lagrange interpolation. The spline-like interpolation function can be converted into a global spline interpolating (G-SPIN) formula that has exactly the same form as in the global Lagrange interpolation. CQF computations are performed using integration and differentiation of the G-SPIN formula in an exact manner. As a result, computed CQFs can attain high precision equivalent to those obtained using the traditional spectral method. This paper shows that Lagrange interpolation can be represented by a G-SPIN formula. Therefore, Gauss-type formulas for any kind of computational node can be computed using the proposed methods. The present method is validated through a series of applications using Legendre-Gauss, Legendre-Gauss-Lobatto, and Chevyshev-Gauss-Lobatto points and by showing that a high accuracy in predicting CQFs is obtained and equivalent to those reached using the traditional quadrature formulas.

The extended applicability of the proposed method to solutions on arbitrary nodes is shown next. For this purpose, two different sets of nodes are considered. One is generated using the coordinate transformation devised by the authors [11] for more uniformly distributed nodes than those obtained using the traditional spectral method. The other is a set of uniform nodes known as the worst possible choice for polynomial interpolation, with which the derived formulas are extremely inaccurate as mentioned in [4] and [12]. The Lagrange and tension spline interpolations over these sets of nodes are investigated by applying the present method to three carefully selected test functions and by comparing the resultant absolute errors in predicting integration and differentiation for those functions. The results of comparative studies demonstrate

[†] Corresponding Author: Dept. of Aerospace Information Engineering, Konkuk University, Korea. (sksung@konkuk.ac.kr)

* Dept. of Aerospace Information Engineering, Konkuk University, Korea. (cjkim@konkuk.ac.kr)

** School of Electrical Engineering, Kyungpook National University Korea. (jgpark@ee.knu.ac.kr)

Received: June 3, 2014; Accepted: February 9, 2015

that the quadrature weights can be retained positivity with the tension spline interpolation regardless on the number of nodes. Whereas, the Lagrange interpolation show highly oscillatory behaviors in the quadrature weights, some of which become negative, as the number of nodes are increased. Therefore, the tension spline is more robust for uniform nodes than the Lagrange interpolation. In addition, it is indicated that the applications of advanced grid topologies, such as the local adaptive grid and multi-grid techniques, is possible to enhance both accuracy and convergence for the iterative solutions based on the quadrature formula with the flexibility allowed by the preset method in selecting computational nodes.

2. Unified Quadrature Formulas using Spline Interpolation

The spline interpolation for a set of function data $\{f_m = f(t_m)\}_{m=0}^M$ given over the computational nodes $\{t_m\}_{m=0}^M$ can be represented using the $(M+1) \times (N+1)$ unknown coefficients $\{a_{m,n}\}_{m=0, n=0}^{m=M, n=N}$ as

$$f(t) \approx \tilde{f}_m(\tau_m) = \sum_{n=0}^N a_{m,n} \tau_m^n, \quad t \in [t_m, t_{m+1}] \quad (1)$$

where N is the leading order of the piecewise continuous interpolating polynomials and $\tau_m \in [0,1]$ is a non-dimensional independent variable defined by

$$\tau_m = \frac{(t - t_m)}{\Delta t_m} \quad (2)$$

where $\Delta t_m = t_{m+1} - t_m$.

For deriving CQFs corresponding to the set of nodes $\{t_m\}_{m=0}^M$, this paper makes the following assumption, the rationale of which will become clear in the later part of this section.

Assumption: The unknown coefficient $a_{m,n}$ in (1) can be represented by a linear combination of function values $\{f_m\}_{m=0}^M$ as

$$a_{m,n} = \sum_{k=0}^M \alpha_{m,n,k} f_k \quad (3)$$

Using the assumption, the function can be approximated over $t \in [t_m, t_{m+1}]$ with the following G-SPIN formula.

$$f(t) \approx \tilde{f}_m(\tau_m) = \sum_{k=0}^M \tilde{\phi}_{m,k}(\tau_m) f_k \quad (4)$$

where

$$\tilde{\phi}_{m,k}(\tau_m(t)) = \sum_{n=0}^N \alpha_{m,n,k} \tau_m^n \quad (5)$$

The partial integration and the p^{th} derivative of the function at any points over $t \in [t_m, t_{m+1}]$ can be approximated using (6) - (8), which come from their exact derivation using (4) and (5).

$$\int_{t_m}^t f(t) dt \approx \sum_{k=0}^M I_{m,k}(\tau) f_k \quad (6)$$

$$\frac{d^p f(t)}{dt^p} \approx \sum_{k=0}^M D_{m,k}^{(p)}(\tau) f_k \quad (7)$$

where

$$\begin{aligned} \tau &= \frac{t - t_m}{\Delta t_m} \\ I_{m,k}(\tau) &= \Delta t_m \sum_{n=0}^N \frac{\alpha_{m,n,k}}{n+1} \tau^{n+1} \\ D_{m,k}^{(p)}(\tau) &= \frac{1}{(\Delta t_m)^p} \sum_{n=p}^N \alpha_{m,n,k} \tau^{n-p} \prod_{l=0}^{p-1} (n-l) \quad (p \leq N) \end{aligned} \quad (8)$$

The partial integration over $t \in [t_0, t_j]$ and the differentiation at t_j can be represented by

$$\begin{aligned} \int_{t_0}^{t_j} f(t) dt &\approx \sum_{m=0}^{j-1} \sum_{k=0}^M I_{m,k}(1) f_k \\ &= \sum_{k=0}^M \left(\sum_{m=0}^{j-1} \Delta t_m \sum_{n=0}^N \frac{\alpha_{m,n,k}}{n+1} \right) f_k \quad (j=1, \dots, M) \end{aligned} \quad (9)$$

$$\begin{aligned} \frac{d^p f(t_j)}{dt^p} &\approx \sum_{k=0}^M D_{j,k}^{(p)}(0) f_k = \frac{1}{(\Delta t_m)^p} \sum_{k=0}^M \left\{ \alpha_{m,p,k} \prod_{l=0}^{p-1} (p-l) \right\} f_k \\ &\quad (j=1, \dots, M, p \leq N) \end{aligned} \quad (10)$$

$$\begin{aligned} \frac{d^p f(t_M)}{dt^p} &\approx \sum_{k=0}^M D_{M-1,k}^{(p)}(1) f_k \\ &= \frac{1}{(\Delta t_{M-1})^p} \sum_{k=0}^M \left(\sum_{n=p}^N \alpha_{M-1,n,k} \prod_{l=0}^{p-1} (n-l) \right) f_k \quad (p \leq N) \end{aligned} \quad (11)$$

Therefore, the CQFs can be represented for the traditional spectral method, where the domain $[t_0, t_M]$ of interests is transformed into $[-1, 1]$ using the affine transformation [4-7].

$$\int_{t_0}^{t_j} f(t) dt \approx \frac{t_M - t_0}{2} \sum_{k=0}^M \tilde{I}_{j,k} f_k \quad (12)$$

$$\int_{t_0}^{t_M} f(t) dt \approx \frac{t_M - t_0}{2} \sum_{k=0}^M \tilde{w}_k f_k \quad (13)$$

$$\frac{df(t_j)}{dt} \approx \frac{2}{t_M - t_0} \sum_{k=0}^M \tilde{D}_{j,k} f_k \quad (14)$$

where

$$\tilde{I}_{j,k} = \frac{2}{t_M - t_0} \sum_{m=0}^{j-1} \Delta t_m \sum_{n=0}^N \frac{\alpha_{m,n,k}}{n+1} \quad (j=1, \dots, M, k=0, \dots, M) \quad (15)$$

$$\tilde{w}_k = \tilde{I}_{M,k} \quad (16)$$

$$\tilde{D}_{j,k} = \begin{cases} \frac{t_M - t_0}{2} \frac{\alpha_{j-1,k}}{(\Delta t_j)} & (j=0, \dots, M-1, k=0, \dots, M) \\ \frac{t_M - t_0}{2} \frac{\sum_{n=1}^N n \alpha_{M-1,n,k}}{(\Delta t_{M-1})} & (j=M, k=0, \dots, M) \end{cases} \quad (17)$$

Therefore, once the coefficients $\{\alpha_{m,n,k}\}_{m=0, n=0, k=0}^{m=M, n=N, k=M}$ are known for the set of arbitrary nodes $\{t_m\}_{m=0}^{m=M}$, the corresponding CQFs can be computed using (12) ~ (17).

To access the usefulness of the assumption and the resultant formulas in obtaining CQFs, we first formulate the coefficients $\{\alpha_{m,n,k}\}$ corresponding to the Lagrange interpolation as follows. The Lagrange interpolation uses continuous polynomials with global support, but it can be reformulated using piecewise continuous polynomials of the same form as the G-SPIN formula. For this, the following reformulation is applied for $t \in [t_m, t_{m+1}]$ using the definition given in (2).

$$\phi_k(t(\tau_m)) = \prod_{\substack{j=0 \\ j \neq k}}^N \frac{(t - t_j)}{(t_k - t_j)} = \prod_{\substack{j=0 \\ j \neq k}}^N \frac{(t_m - t_j + \Delta t_m \tau_m)}{(t_k - t_j)} = \sum_{n=0}^N \alpha_{m,n,k} \tau_m^n \quad (18)$$

The required coefficients can be obtained using the following recursive procedures.

(i) set $a_{m,0}^{(0)} = 1.0$; $l = 0$

(ii) for $n = 0, N$ when $n \neq k$
 $l = l + 1$

$$\alpha_n = \frac{(t_m - t_n)}{(t_k - t_n)}$$

$$\beta_n = \frac{\Delta t_m}{(t_k - t_n)}$$

$$a_{m,0}^{(l)} = \alpha_n a_{m,0}^{(l-1)}$$

$$a_{m,j}^{(l)} = \alpha_n a_{m,j}^{(l-1)} + \beta_n a_{m,j-1}^{(l-1)} \quad (j=1, \dots, l-1)$$

$$a_{m,l}^{(l)} = \beta_n a_{m,l-1}^{(l-1)}$$

(iii) set $\alpha_{m,j,k} = a_{m,j}^{(N)}$ ($j=0, \dots, N$)

Therefore, the Lagrange interpolation for a data set $\{(t_m, f_m)\}_{m=0}^{m=N}$ defined over arbitrarily distributed nodes can be reduced to G-SPIN form as shown in (4) and (5). The corresponding CQFs can be computed using (12) ~ (17).

This procedure is represented for the Lobatto-type quadrature points containing two endpoints in the domain of interest. However, minor modifications are sufficient for its extended application to other types of nodes. As an

example with the Radau-type quadrature points, the given data set can be defined by $\{f_m, t_m\}_{m=1}^{m=N}$ with the order of interpolating polynomials $M = N - 1$. Even in this case, the coefficients $\{\alpha_{0,n,k}\}_{n=0, k=1}^{n=N-1, k=M}$ corresponding to the first interval $t \in [t_0, t_1]$ can be obtained using the procedures shown in (i)-(iii) for each of the Lagrange interpolation polynomials defined with N Radau-type points $\{t_m\}_{m=1}^{m=N}$. Therefore, the proposed method can be used without limits for other types of nodes, such as Gauss-type and flipped Radau-type points, and regardless of the adopted orthogonal polynomials, such as Legendre and Chebyshev polynomials.

The multiple derivatives of the function can easily be formulated using the G-SPIN formula shown in (4) and (5). The differentiation matrices for obtaining higher function derivatives are typically obtained by recursively applying the differentiation formula given in (14). Using the following definition for vectors and matrices,

$$\mathbf{f}^{(p)} = \left(\frac{d^p f(t_0)}{dt^p}, \frac{d^p f(t_1)}{dt^p}, \dots, \frac{d^p f(t_M)}{dt^p} \right)^T \quad (19)$$

$$\tilde{\mathbf{D}} = \begin{pmatrix} \tilde{D}_{0,0} & \tilde{D}_{0,1} & \dots & \tilde{D}_{0,M} \\ \tilde{D}_{1,0} & \tilde{D}_{1,1} & \dots & \tilde{D}_{1,M} \\ \vdots & \vdots & \ddots & \vdots \\ \tilde{D}_{M,0} & \tilde{D}_{M,1} & \dots & \tilde{D}_{M,M} \end{pmatrix} \quad (20)$$

the first derivative can be expressed by

$$\mathbf{f}^{(1)} = \frac{2}{t_M - t_0} \tilde{\mathbf{D}} \mathbf{f}^{(0)} \quad \text{where } \mathbf{f}^{(0)} = (f_0, \dots, f_M)^T \quad (21)$$

Recursive application of (21) results in the conventional formula for the approximation of the p^{th} function-derivative vector corresponding to the collocation points.

$$\mathbf{f}^{(p)} = \left(\frac{2}{t_M - t_0} \right)^p \tilde{\mathbf{D}}^p \mathbf{f} \quad (22)$$

The G-SPIN formula allows multiple differentiation to get the p^{th} derivatives of the function at each node, as shown in (10) and (11). If the components of the matrix $\tilde{\mathbf{D}}^p$ are denoted by $\{\tilde{d}_{j,k}^{(p)}\}_{j=0, k=0}^{j=M, k=M}$, then each component $\tilde{d}_{j,k}^{(p)}$ can be obtained directly using these equations as

$$\tilde{d}_{j,k}^{(p)} = \begin{cases} \left(\frac{t_M - t_0}{2\Delta t_j} \right)^p \alpha_{j,p,k} \prod_{l=0}^{p-1} (p-l) & (j=1, \dots, M, k=0, \dots, M, p \leq N) \\ \left(\frac{t_M - t_0}{2\Delta t_{M-1}} \right)^p \sum_{n=p}^N \alpha_{M-1,n,k} \prod_{l=0}^{p-1} (n-l) & (j=M, k=0, \dots, M, p \leq N) \end{cases} \quad (23)$$

respectively.

3. Validation of the Proposed Method

The proposed formulas for computing CQFs are validated using existing formulas available for the pseudospectral methods. For this purpose, we consider three sets of quadrature points defined by (a)-(c) with the corresponding formulas for the computation of CQFs.

(a) Legendre-Gauss (LG) points $\{t_k^{LG}\}_{k=0}^{k=N}$: the roots of $g(t) = L_{N+1}(t) = 0$.

$$D_{j,k}^{LG} = \begin{cases} \frac{\dot{L}_{N+1}(t_j)}{(\tau_j - \tau_k)\dot{L}_{N+1}(t_k)} & (j \neq k) \\ \frac{t_j}{1-t_j^2} & (j = k) \end{cases} \quad (24)$$

$$I_{j,k}^{LG} = \frac{w_k}{2} \left(1 + t_j + \sum_{p=1}^{N-1} L_p(t_k) \{L_{p+1}(t_j) - L_{p-1}(t_j)\} \right) \quad (25)$$

$$w_k^{LGL} = \frac{2}{(1-t_k^2)\{\dot{L}_{N+1}(t_k)\}^2} \quad (26)$$

(b) Legendre-Gauss-Lobatto (LGL) points $\{t_k^{LGL}\}_{k=0}^{k=N}$: the roots of $g(t) = L_{N+1}(t) - L_{N-1}(t) = 0$.

$$D_{j,k}^{LGL} = \begin{cases} \frac{L_N(t_j)}{(t_j - t_k)L_N(t_k)} & (j \neq k) \\ -\frac{N(N+1)}{4} & (j = k = 0) \\ +\frac{N(N+1)}{4} & (j = k = N) \\ 0 & (1 \leq k = j \leq N-1) \end{cases} \quad (27)$$

$$I_{j,k}^{LGL} = \frac{w_k}{2} \left(1 + t_{j+1} + \sum_{p=1}^N L_p(t_k) \{L_{p+1}(t_{j+1}) - L_{p-1}(t_{j+1})\} \right) \quad (28)$$

$$w_k^{LGL} = \frac{2}{N(N+1)} \frac{1}{\{L_N(t_k)\}^2} \quad (29)$$

(c) Chebyshev-Gauss-Lobatto (CGL) points $\{t_k^{CGL}\}_{k=0}^{k=N}$:

$$t_k^{CGL} = -\cos\left(\frac{k\pi}{N}\right) \quad (k = 0, \dots, N)$$

$$D_{j,k}^{CGL} = \begin{cases} \left(\frac{c_j}{c_k}\right) \left(\frac{(-1)^{j+k}}{t_j - t_k}\right) & (k \neq j) \\ -\frac{t_j}{2(1-t_j^2)} & (1 \leq k = j \leq N-1) \\ -\frac{(2N^2+1)}{6} & (0 = k = j) \\ +\frac{(2N^2+1)}{6} & (N = k = j) \end{cases} \quad (30)$$

$$I_{j,k}^{CGL} = \frac{\theta_k}{N} \left[(1+t_j) + t_k(t_j^2-1) + \sum_{n=2}^N \alpha_n T_n(t_k) \cdot \left\{ \frac{T_{n+1}(t_j)}{(n+1)} - \frac{T_{n-1}(t_j)}{(n-1)} + 2 \frac{(-1)^{n+1}}{n^2-1} \right\} \right] \quad (31)$$

$$w_k^{CGL} = I_{Nk} \quad (32)$$

where

$$c_0 = c_N = 2$$

$$c_j = 1, \quad (j = 1, 2, \dots, N-1)$$

$$\alpha_j = 1 \quad (j = 0, 1, 2, \dots, N-1)$$

$$\alpha_N = \frac{1}{2}$$

$$\theta_0 = \theta_N = \frac{1}{2}$$

$$\theta_j = 1 \quad (j = 1, 2, \dots, N-1)$$

In these equations the orthogonal polynomials $L_j(t)$ and $T_k(t)$ represent the j^{th} Legendre and k^{th} Chebyshev polynomials, respectively. The integration matrices for (a) and (b) come from Axelson's algorithm derived using the Christoffel-Darboux identity theorem for the kernel function defined with orthogonal polynomials [9]. The integration matrix for (c) represents the Clenshaw-Curtis algorithm for Chebyshev polynomials [10, 13]. The accuracy of (12)-(17) is estimated from the root-mean-squares (RMS) value of the differences in the components of the CQFs, defined by

$$\Delta D_{j,k} = \tilde{D}_{j,k} - D_{j,k}^{LGL} \quad (33)$$

Table 1. Normalized differences in the CQFs

M	LG points			LGL points			CGL points		
	$\ \Delta w\ _n$	$\ \Delta l\ _n$	$\ \Delta D\ _n (\times 10^{13})$	$\ \Delta w\ _n$	$\ \Delta l\ _n$	$\ \Delta D\ _n (\times 10^{13})$	$\ \Delta w\ _n$	$\ \Delta l\ _n$	$\ \Delta D\ _n (\times 10^{13})$
21	0.0187	0.0171	0.0915	0.0074	0.0074	0.0132	0.0114	0.0101	0.0333
41	0.0197	0.0182	0.0922	0.0175	0.0165	0.0459	0.0188	0.0166	0.0411
61	0.0276	0.0255	0.1529	0.0229	0.0225	0.1283	0.0389	0.0328	0.2364
81	0.0316	0.0293	0.4563	0.0318	0.0299	0.1046	0.0503	0.0452	0.3324
101	0.0324	0.0301	0.2112	0.0329	0.0301	0.2381	0.0738	0.0655	0.6785
121	0.0496	0.0481	3.4201	0.0397	0.0382	0.7164	0.0962	0.0802	0.4380
141	0.0575	0.0533	2.2772	0.0445	0.0420	1.7351	0.0642	0.0583	1.3944
161	0.0522	0.0488	1.1952	0.0442	0.0413	0.3086	0.1313	0.1076	1.4977
181	0.0501	0.0467	1.7362	0.0548	0.0521	1.0757	0.1214	0.1126	1.5792
201	0.0580	0.0560	0.1236	0.0569	0.0558	3.0018	0.1203	0.1099	1.0221

where $\tilde{D}_{j,k}$ is computed using (10). The normalized norm defined by $\|\Delta D\|_n = \|\Delta D\|_{RMS} / \|D\|_{RMS}$ is used for convenience in comparison.

Table 1 shows that the computed results and the proposed method can predict CQFs regardless of the number of nodes at a level of accuracy equivalent to that of spectral methods.

The computational domain typically includes both end points to impose the initial and final conditions. As an example of the differentiation method with LG points, the modified Lagrange interpolation functions are used openly to include the effect of the initial end point. The Lagrange interpolation functions for the LG points $\{t_j\}_{j=1}^{j=M-1}$ and the approximation of the function and its first derivatives at each node can be represented by (35) and (36).

$$\phi_k^{LG}(t) = \prod_{\substack{l=1 \\ l \neq k}}^{M-1} \frac{(t-t_l)}{(t_k-t_l)} \quad (k=1,2,\dots,M-1) \quad (34)$$

$$f(t) \approx \sum_{k=0}^{M-1} \phi_k^{LG}(t) f_k \quad (35)$$

$$\frac{df(t_j)}{dt} \approx \sum_{k=0}^{M-1} D_{j,k}^{LG} f_k \quad \text{where } D_{j,k}^{LG} = \phi_k^{LG}(t_j) \quad (36)$$

The corresponding modified Lagrange interpolation functions, the first of which corresponds to the initial end point $t_0 = -1$, are typically defined as

$$\tilde{\phi}_k^{LG}(t) = \begin{cases} \prod_{l=1}^{M-1} \frac{(t-t_l)}{(t_0-t_l)}, & k=0 \\ \frac{(t-t_0)}{(t_k-t_0)} \phi_k^{LG}(t), & k=1,\dots,M-1 \end{cases} \quad (37)$$

The function and its first derivative at each node can be approximated with $\{f_j\}_{j=0}^{j=M-1}$ as

$$f(\tau) \approx \sum_{k=0}^{M-1} \tilde{\phi}_k^{LG}(\tau) f_k \quad (38)$$

$$\frac{df(t_j)}{dt} \approx \sum_{k=0}^{M-1} \tilde{D}_{j,k}^{LG} f_k \quad (39)$$

where $\tilde{D}_{j,k}^{LG}$ can be defined using (36) and (37) for $j=0,\dots,N$ and $k=1,\dots,M-1$ as

$$\tilde{D}_{j,k}^{LG} = \frac{1}{(t_k-t_0)} \delta_{jk} + \frac{(t_j-t_0)}{(t_k-t_0)} D_{j,k}^{LG} \quad (40)$$

The elements corresponding to $k=0$ can be computed using the fact that the derivative of a constant function is uniformly zero at all nodes.

$$\tilde{D}_{j,0}^{LG} \approx -\sum_{p=1}^{M-1} \tilde{D}_{j,p}^{LG}, \quad j=0,1,\dots,M-1 \quad (41)$$

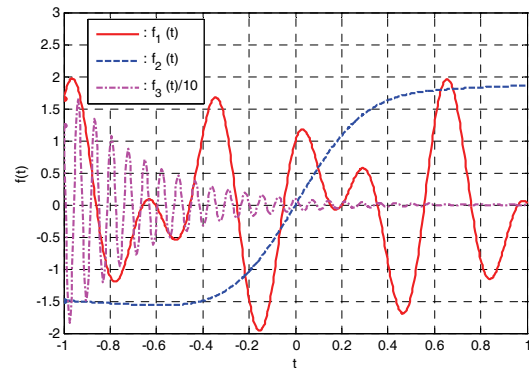


Fig. 1. Test functions

These derivations are typically used in the spectral method. The modified differentiation matrix is obtainable directly from the G-SPIN formula by including the data (t_0, f_0) in the data set defined over the LG points. Likewise, the elements related to the final endpoint are easily computed in the same manner, if required. Consequently, the proposed method allows straightforward computing of all elements of the differentiation matrix related to all nodes $\{t_j\}_{j=0}^{j=M}$ including two end points, regardless of the types of nodes and without computing the modified differentiation matrix using formulas such as (40) and (41).

The accuracy of the proposed method is validated using the following three test functions selected from [14]. Fig. 1 shows these functions over $t \in [-1,1]$.

Function 1: $f_1(t) = \cos(\sqrt{377}t) + \sin(\sqrt{135}t)$

Function 2: $f_2(t) = \ln(t+2e)\text{erf}(\pi t)$

Function 3: $f_3(t) = e^{-3t} \cos(16\sqrt{3}\pi t)$

The error function $\text{erf}(y)$ in $f_2(t)$ is an entire function defined by (42) and can be approximated using the series expansion of the exponential function as shown in (43). Therefore, the first derivative of $f_2(t)$ can be approximated by (45).

$$\text{erf}(y) = \frac{2}{\sqrt{\pi}} \int_0^y e^{-t^2} dt \quad (42)$$

$$\text{erf}(y) = \frac{2}{\sqrt{\pi}} \sum_{k=0}^{\infty} \frac{(-1)^k y^{2k+1}}{k!(2k+1)} \quad (43)$$

$$\frac{d\{\text{erf}(y)\}}{dy} = \frac{2}{\sqrt{\pi}} \sum_{k=0}^{\infty} \frac{(-1)^k y^{2k}}{k!} \quad (44)$$

$$\frac{df_2(t)}{dt} = \frac{\text{erf}(\pi t)}{t+2e} + \pi \ln(t+2e) \frac{d\{\text{erf}(y)\}}{dy} \Big|_{y=\pi t} \quad (45)$$

The accuracy of (12)~(17) is estimated using the RMS norms of the absolute error vectors for partial integration and differentiation and by the absolute value of the integration error over $t \in [-1, 1]$, presented with $\|\Delta I_j\|$,

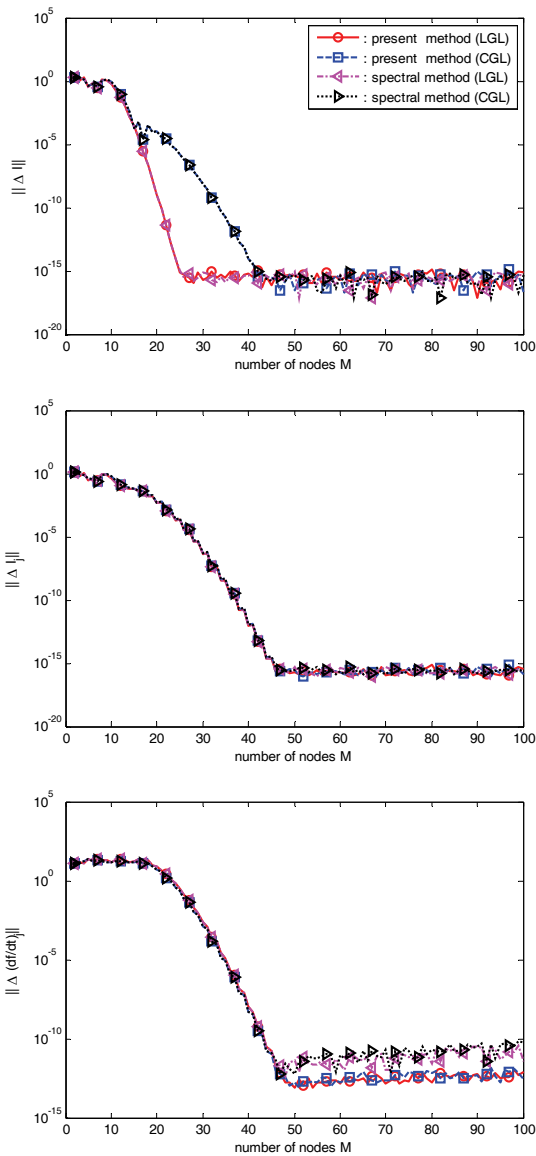


Fig. 2. Computational errors for $f_1(t)$.

$\|\Delta D_j\|$, and $\|\Delta I\|$, respectively. The integrations over the entire domain for $f_1(t)$ and $f_2(t)$ are given, respectively, by $I_2(-1, 1) \approx 0.175664900305971$ and $I_3(-1, 1) = -0.176358246030565$ [14]. These values are used to measure the absolute integration errors. Fig. 2 shows the computational errors using different formulas and with varying numbers of nodes for $f_1(t)$.

Fig. 3 shows the results for $f_2(t)$ and $f_3(t)$. The present method shows nearly the same level of accuracy as the conventional methods in both the integration and the differentiation varying the number of nodes. The present method is more advantageous in obtaining the highly accurate first derivative for $f_1(t)$ and $f_3(t)$ in the case when the number of nodes is sufficiently large to provide machine precision. The LGL points are the better selection for accurate integration than the CGL points for all test functions.

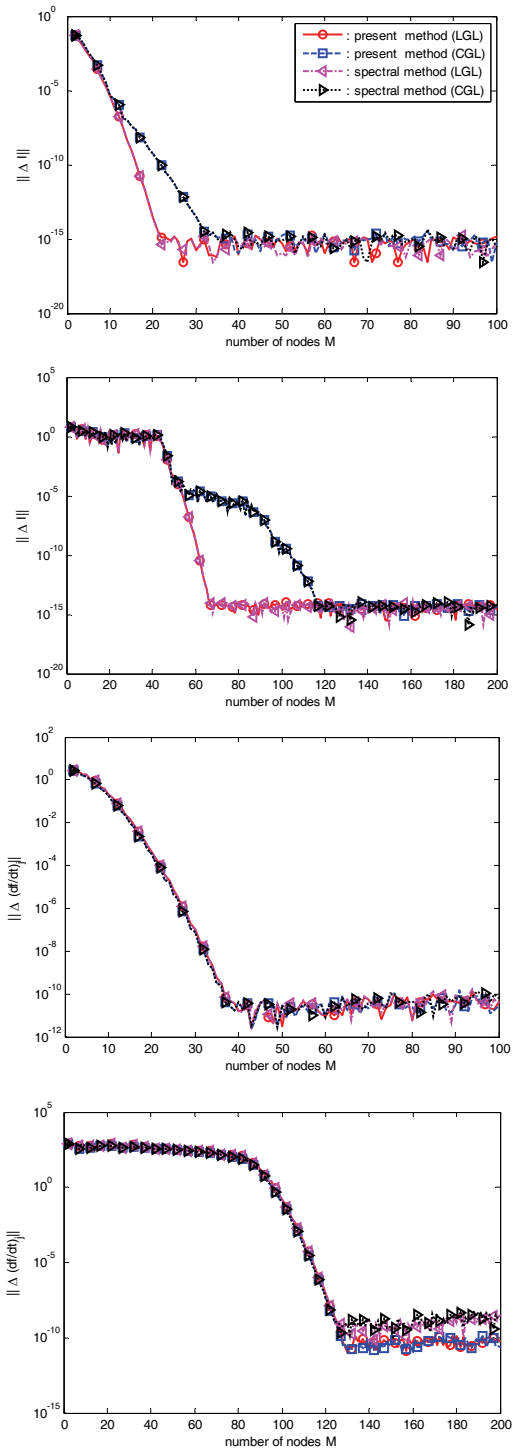


Fig. 3. Computational errors for $f_2(t)$ and $f_3(t)$.

4. Extension to Spline Interpolation

Since the formulas given in (12)-(17) are derived on the basis of the assumption shown in (3), other interpolations rather than the Lagrange interpolation can be used, once they are represented in the G-SPIN form. As shown in the

authors' paper [12] and from the present study, the spline interpolation is a natural choice for the G-SPIN formula. For completeness, we repeat the main results of the earlier paper in this Section with detailed procedures for the derivation of the G-SPIN formula for spline interpolation. From the definition given by (1), the function values and its derivatives up to the k^{th} order can be approximated at each node by

$$f^{(k)}(t_m) \approx \frac{\tilde{f}_m^{(k)}(0)}{(\Delta t_m)^k} = \frac{k!}{(\Delta t_m)^k} a_{m,k}, \quad (k=0,1,\dots,N) \quad (46)$$

$$f^{(k)}(t_{m+1}) \approx \frac{\tilde{f}_m^{(k)}(1)}{(\Delta t_m)^k} = \sum_{n=k}^N \frac{n!}{(n-k)! (\Delta t_m)^k} a_{m,n} \quad (47)$$

The unknown coefficients $\{a_{m,n}\}_{m=0,n=0}^{m=M,n=N}$ for spline interpolation are typically determined by imposing the following conditions over the computational nodes.

i) function values at each node

$$f_m = \tilde{f}_m(0) = a_{m,0} \quad (m=0, M-1) \quad (48)$$

$$f_M = \tilde{f}_{M-1}(1) = \sum_{n=0}^N a_{M-1,n} \quad (49)$$

ii) continuity of the function and its higher derivatives at each inter-connecting node

$$f^{(k)}(t_m) \approx \tilde{f}_m^{(k)}(1) = \tilde{f}_{m+1}^{(k)}(0) \quad \text{or} \quad \sum_{n=k}^N \frac{n!}{(n-k)! (\Delta t_m)^k} a_{m,n} = \frac{k!}{(\Delta t_{m+1})^k} a_{m+1,k} \quad \begin{cases} m=0, \dots, M-1 \\ k=0, \dots, N-1 \end{cases} \quad (50)$$

iii) $(N-1)$ additional conditions to close the solution

There exist many alternatives for defining the additional conditions (iii). The most widely used among them are the natural spline and the not-a-knot spline. The natural spline can be obtained by setting higher order derivatives at two end points to zeroes as

$$f^{(k)}(t_0) \approx \frac{\tilde{f}_0^{(k)}(0)}{(\Delta t_0)^k} = 0 \quad (k=N-N_0+1, \dots, N-1, N) \quad (51)$$

or $a_{0,k} = 0$

$$f^{(k)}(t_f) \approx \frac{\tilde{f}_{N-1}^{(k)}(1)}{(\Delta t_{N-1})^k} = 0 \quad (k=N-N_M+1, \dots, N-1, N) \quad (52)$$

or $\sum_{n=k}^N \frac{n!}{(n-k)!} a_{N-1,n} = 0$

where N_0 and N_M are positive integers and can be selected under the following condition.

$$N_0 + N_M = N - 1$$

In contrast, the not-a-knot spline imposes continuity conditions for the N^{th} order derivatives near two end points with the same condition in selecting N_0 and N_M , such as

$$f^{(N)}(t_j) \approx \frac{\tilde{f}_j^{(N)}(0)}{(\Delta t_j)^N} = 0 \quad (j=1, \dots, N_0, N-N_M, \dots, N-1) \quad (53)$$

or $a_{j,N} = 0$.

From the conditions given in (48)-(53), the unknown coefficients $\{a_{m,n}\}_{m=0,n=0}^{m=M,n=N}$ can be obtained by solving the following linear algebraic equation.

$$\mathbf{M}\bar{\mathbf{a}} = \mathbf{C}\mathbf{f} \quad \text{or} \quad \bar{\mathbf{a}} = \mathbf{M}^{-1}\mathbf{C}\mathbf{f} \quad (54)$$

where \mathbf{a} and \mathbf{f} are defined by

$$\bar{\mathbf{a}} = \begin{pmatrix} \mathbf{a}_0 \\ \vdots \\ \mathbf{a}_{M-1} \end{pmatrix}, \quad \mathbf{a}_m = \begin{pmatrix} a_{m,0} \\ \vdots \\ a_{m,N} \end{pmatrix} \quad (m=0,1,\dots,M-1), \quad \mathbf{f} = \begin{pmatrix} f_0 \\ \vdots \\ f_M \end{pmatrix}$$

The coefficient matrix \mathbf{C} has the dimension $M(N+1)$ -by- $(M+1)$ and most of its components are zeroes, except at the rows related to condition (i) when the natural or not-a-knot spline is used. The matrix \mathbf{M} is easily defined by the coefficients $\{a_{m,n}\}_{m=0,n=0}^{m=M,n=N}$ in (48) ~ (53).

Therefore, the coefficients $\{a_{m,n,k}\}_{n=0,m=0}^{n=N,m=M}$ corresponding to the function value f_k can be computed from the solution of (54) by substituting the elements of the function vector with $f_j = \delta_{jk}$ ($j=0, \dots, M$). Therefore, the CQFs corresponding to the spline interpolation can be derived using (12) ~ (17) regardless of what kinds of $(N-1)$ additional conditions (iii) are imposed.

Next, we consider the application of the proposed method using the tension spline interpolation. The tension spline is not a kind of the piecewise continuous polynomial interpolation. However, it will be shown that the tension spline interpolation can be represented by the G-SPIN formula. The tension spline interpolation for a data set $\{(t_m, f_m)\}_{m=0}^{m=M}$ related to a function $f(t) \in C^2[t_0, t_f]$ can be defined using the solution of the fourth order differential equation [16] for each time interval $t \in [t_m, t_{m+1}]$ with an adjustable tension parameter $\tilde{\rho}_m$ as given by

$$\frac{d^4 y}{dt^4} - \tilde{\rho}_m^2 \frac{d^2 y}{dt^2} = 0, \quad t \in [t_m, t_{m+1}] \quad (55)$$

with the following boundary conditions

$$y(t_m) = f_m, y(t_{m+1}) = f_{m+1}, y''(t_m) = f_m'', y''(t_{m+1}) = f_{m+1}'' \quad (56)$$

and using the definition given in (3), the solution of (55)

can be represented by

$$f(t) \approx f_m(\tau_m) = \frac{1}{\tilde{\rho}_m^2 \sinh \tilde{\rho}_m \Delta t_m} \{ f_m'' \sinh \tilde{\rho}_m \Delta t_m (1 - \tau_m) + f_{m+1}'' \sinh \tilde{\rho}_m \Delta t_m \tau_m \} + \left\{ \left(f_m - \frac{1}{\tilde{\rho}_m^2} f_m'' \right) (1 - \tau_m) + \left(f_{m+1} - \frac{1}{\tilde{\rho}_m^2} f_{m+1}'' \right) \tau_m \right\} \quad (57)$$

The unknown second derivatives $\{f_m''\}_{m=0}^M$ are typically determined by imposing the continuity conditions for the first derivatives at each interconnecting node and by adding two end conditions. The possible selection of two end conditions is exemplified in [13] with the following three types.

Type I: $y'(t_0) = f_0', y'(t_m) = f_m'$

Type II: $y''(t_0) = y''(t_m) = 0$

Type III: $y''(t_0) = f_0'', y''(t_m) = f_m''$

Types I and III need the direct specification of the first and second derivatives at two ends, which is inconvenient for general cases without prior information on these values. In addition, Type III is not the usual one for the tension spline as mentioned in [16], and the following modified end condition using the third derivatives is proposed for the application of the present G-SPIN formula.

$$y'''(t_0) = y'''(t_M) = 0 \quad (58)$$

Using the following definition,

$$\rho_m = \tilde{\rho}_m \Delta t_m$$

$$S_m = \sinh \rho_m$$

(58) can be rewritten as

$$f_m(\tau_m) = \frac{\Delta t_m^2}{\rho_m^2 S_m} \{ f_m'' \sinh \rho_m (1 - \tau_m) + f_{m+1}'' \sinh \rho_m \tau_m \} + \left\{ \left(f_m - \frac{\Delta t_m^2}{\rho_m^2} f_m'' \right) (1 - \tau_m) + \left(f_{m+1} - \frac{\Delta t_m^2}{\rho_m^2} f_{m+1}'' \right) \tau_m \right\} \quad (59)$$

Therefore, the first and third derivatives of the function over $t \in [t_m, t_{m+1}]$ can be approximated by

$$\frac{df(t)}{dt} \approx \frac{1}{\Delta t_m} \frac{df_m(\tau_m)}{d\tau_m} = \frac{\Delta t_m}{\rho_m S_m} \{ -f_m'' \cosh \rho_m (1 - \tau_m) + f_{m+1}'' \cosh \rho_m \tau_m \} - \frac{1}{\Delta t_m} \left(f_m - \frac{\Delta t_m^2}{\rho_m^2} f_m'' \right) + \frac{1}{\Delta t_m} \left(f_{m+1} - \frac{\Delta t_m^2}{\rho_m^2} f_{m+1}'' \right)$$

$$\frac{d^3 f(t)}{dt^3} \approx \frac{1}{\Delta t_m^3} \frac{d^3 f_m(\tau_m)}{d\tau_m^3} = \frac{\rho_m}{\Delta t_m S_m} \{ -f_m'' \cosh \rho_m (1 - \tau_m) + f_{m+1}'' \cosh \rho_m \tau_m \}$$

The continuity conditions for the first derivatives can be expressed as

$$\frac{df(t_m)}{dt} \approx \frac{1}{\Delta t_m} \frac{df_m(0)}{d\tau_m} = \frac{1}{\Delta t_{m-1}} \frac{df_{m-1}(1)}{d\tau_{m-1}} \quad (m=1,2,\dots,N-1)$$

$$\frac{\Delta t_m}{\rho_m S_m} (-f_m'' \cosh \rho_m + f_{m+1}'') - \frac{1}{\Delta t_m} \left(f_m - \frac{\Delta t_m^2}{\rho_m^2} f_m'' \right) + \frac{1}{\Delta t_m} \left(f_{m+1} - \frac{\Delta t_m^2}{\rho_m^2} f_{m+1}'' \right)$$

or

$$= \frac{\Delta t_{m-1}}{\rho_{m-1} S_{m-1}} (-f_{m-1}'' + f_m'' \cosh \rho_{m-1}) - \frac{1}{\Delta t_{m-1}} \left(f_{m-1} - \frac{\Delta t_{m-1}^2}{\rho_{m-1}^2} f_{m-1}'' \right) + \frac{1}{\Delta t_{m-1}} \left(f_m - \frac{\Delta t_{m-1}^2}{\rho_{m-1}^2} f_m'' \right) \quad (60)$$

Two end conditions shown in (58) can be represented by

$$0 = -f_0'' \cosh \rho_0 + f_1''$$

$$0 = -f_{M-1}'' + f_M'' \cosh \rho_{M-1} \quad (61)$$

As the results of (60) and (61), the unknown second derivatives $\{f_m''\}_{m=0}^M$ can be obtained by solving the following system of the linear algebraic equations in the same form as given by (54).

$$\tilde{\mathbf{M}} \mathbf{f}'' = \tilde{\mathbf{C}} \mathbf{f} \quad \text{or} \quad \mathbf{f}'' = \tilde{\mathbf{M}}^{-1} \tilde{\mathbf{C}} \mathbf{f} \quad \text{with} \quad \mathbf{f}'' = (f_0'', \dots, f_M'')^T \quad (62)$$

The trigonometric hyperbolic functions in (59) can be expressed by the linear combination of $e^{\rho_m \tau_m}$ and $e^{-\rho_m \tau_m}$ using the following relations.

$$\sinh \rho_m (1 - \tau_m) = \frac{e^{\rho_m} e^{-\rho_m \tau_m} - e^{-\rho_m} e^{\rho_m \tau_m}}{2}$$

$$\sinh \rho_m \tau_m = \frac{e^{\rho_m \tau_m} - e^{-\rho_m \tau_m}}{2}$$

In addition, the hyperbolic functions $e^{\pm \rho_m \tau_m}$ can be approximated using the Taylor series expansion as

$$e^{\pm \rho_m \tau_m} \approx \sum_{n=0}^N \frac{(\pm \rho_m)^n}{n!} \tau_m^n + O\left(\frac{\rho_m^{N+1}}{(N+1)!} \tau_m^{N+1} \right) \quad (63)$$

where $O(\cdot)$ is the magnitude of the truncated terms. An accurate approximation for each of the exponential functions can be obtained by increasing N until $O(\cdot)$ reaches machine precision. Since $|\tau_m| \leq 1$, the magnitude of $\rho_m^{N+1}/(N+1)!$ determines the required N , depending on the non-dimensional tension parameter ρ_m . Table 2 shows the estimated magnitude of the truncated terms varying the non-dimensional tension parameter. With the tension parameter less than $\rho = 0.01$, the tension spline approaches the cubic spline (with $N=3$). For the extreme case with $\rho = 10.0$, more than 50 terms are required to get machine precision accuracy. However, this does not cause a large increase in computing time, because the arithmetic operations for (12)-(17) are simple enough,

Table 2. Truncation errors with varying tension parameter

N	$\rho^{N+1}/(N+1)!$			
	$\rho=0.01$	$\rho=0.1$	$\rho=1.0$	$\rho=10.0$
5	1.3889E-15	1.3889E-09	1.3889E-03	1.3889E+03
10	2.5052E-30	2.5052E-19	2.5052E-08	2.5052E+03
15		4.7795E-30	4.7795E-14	4.7794E+02
20			1.9573E-20	1.9572E+01
25			2.4796E-27	2.4796E-01
30				1.2161E-03
35				2.6882E-06
40				2.9893E-09
45				1.8173E-12
50				6.4470E-16

once $\{\alpha_{m,n,k}\}_{k=0,n=N,m=M}^{k=M,n=N,m=M}$ is known, and the CQFs can be computed prior to their applications. Therefore, (59) can be reduced to G-SPIN form with high precision, and (12) ~ (17) can be used in an efficient manner even with tension spline interpolation.

The traditional quadrature nodes are highly clustered around two end points to preserve extreme accuracy, and there is no flexibility in adjusting their spacing. This kind of distribution can be inconvenient in some problems with rapid variations in function values around the midpoint of the computational domain, as shown in [11]. In addition, conventional quadrature weights and matrices for the integration and differentiation might be inadequate for good convergence in the iterative solution processes as the number of nodes is increased [11, 15]. Spline interpolation combined with the present approaches to the CQFs can be a good alternative allowing an adaptive selection of nodes considering the solution properties. This possibility will be demonstrated using two different kinds of node. First, more evenly distributed nodes generated using the coordinate transformation method shown in [8] are used. Second, uniform nodes are selected to show that arbitrary nodes can be utilized in real applications without limits on the number of nodes and without numerical failures using the spline interpolation. With the given set of conventional quadrature nodes $\{\tau_m\}_{m=0}^{m=M}$, the coordinate transformation proposed in [11] is devised to preserve symmetry about $\tau=0$ and monotonicity in the corresponding computational nodes $\{\xi_m\}_{m=0}^{m=M}$, using the following equation.

$$\xi(\tau) = \tau + \tau(\tau^2 - 1) \left(\sum_{k=0}^K a_k \tau^{2k} \right) \quad (64)$$

The coefficients a_j ($j=0,1,2, \dots$) have been determined by the least squares method to match best with target uniform nodes $\{\xi_m^{\text{target}} = -1 + 2m/M\}_{m=0}^{m=M}$. [11] showed the effect of the leading order, K , with the selection of the third-order transformation being recommended, since it provides the best integration accuracy for the series of polynomial test functions. The formulas for integration and

differentiation can be defined easily in the transformed domain using (64) as

$$I(-1, \xi_j) = \int_{-1}^{\xi_j} f(\xi(\tau)) d\xi = \int_{-1}^{\tau_j} f(\xi(\tau)) \frac{d\xi}{d\tau} d\tau \approx \sum_{k=0}^M I_{j,k} f(\xi(\tau_k)) \frac{d\xi(\tau_k)}{d\tau} \quad (65)$$

$$\frac{df(\xi(\tau_j))}{d\xi} = \frac{df(\xi(\tau_j))}{d\tau} \left\{ \frac{d\xi(\tau_j)}{d\tau} \right\}^{-1} \quad (66)$$

If accurate interpolations over an entire domain of interest are possible, the derivative $df(\xi(\tau_j))/d\tau$ can be estimated with high precision in principle. In such a case, the interpolating functions $\{\phi_k^\xi(\xi)\}_{k=0}^{k=M}$ and $\{\phi_m^\tau(\tau)\}_{m=0}^{m=M}$ can be defined to approximate the function and its derivatives using the nodes $\{\xi_k = \xi(\tau_k)\}_{k=0}^{k=M}$ and $\{\tau_m\}_{m=0}^{m=M}$, respectively. The following formula can be derived from a simple manipulation to estimate $df(\xi(\tau_j))/d\tau$ with reference to Fig. 4.

$$\frac{df(\xi(\tau_j))}{d\xi} \approx \sum_{k=0}^M \tilde{D}_{j,k} f(\xi_k) \quad (67)$$

where

$$\tilde{D}_{j,k} = \sum_{m=0}^M \sum_{l=0}^M \phi_m^\tau(\xi(\tau_j)) D_{m,l} \phi_k^\xi(\tau_l) \quad (68)$$

However, it is not a simple task to find such interpolating functions with the required accuracy in real applications. Therefore, we use the local Lagrange interpolating functions using four-point data in this study to demonstrate the performance of (66), because the global one generally results in extremely poor accuracy, whereas the CQFs corresponding to the transformed nodes are directly obtainable using (12)~(17), which require only the information on node distribution. The Lagrange interpolation and the tension spline with the non-dimensional tension parameter of $\rho=0.01$ are used for the applications of the present method. Fig. 5 compares the results of different approaches for $f_1(t)$ and $f_2(t)$. The integration

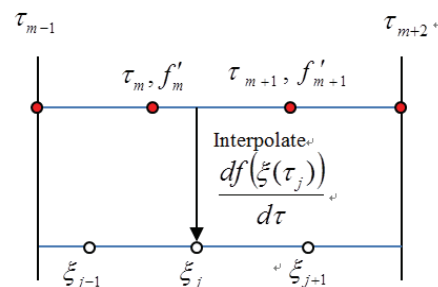


Fig. 4. Schematic diagram to estimate $df(\xi(\tau_j))/d\tau$ using interpolation

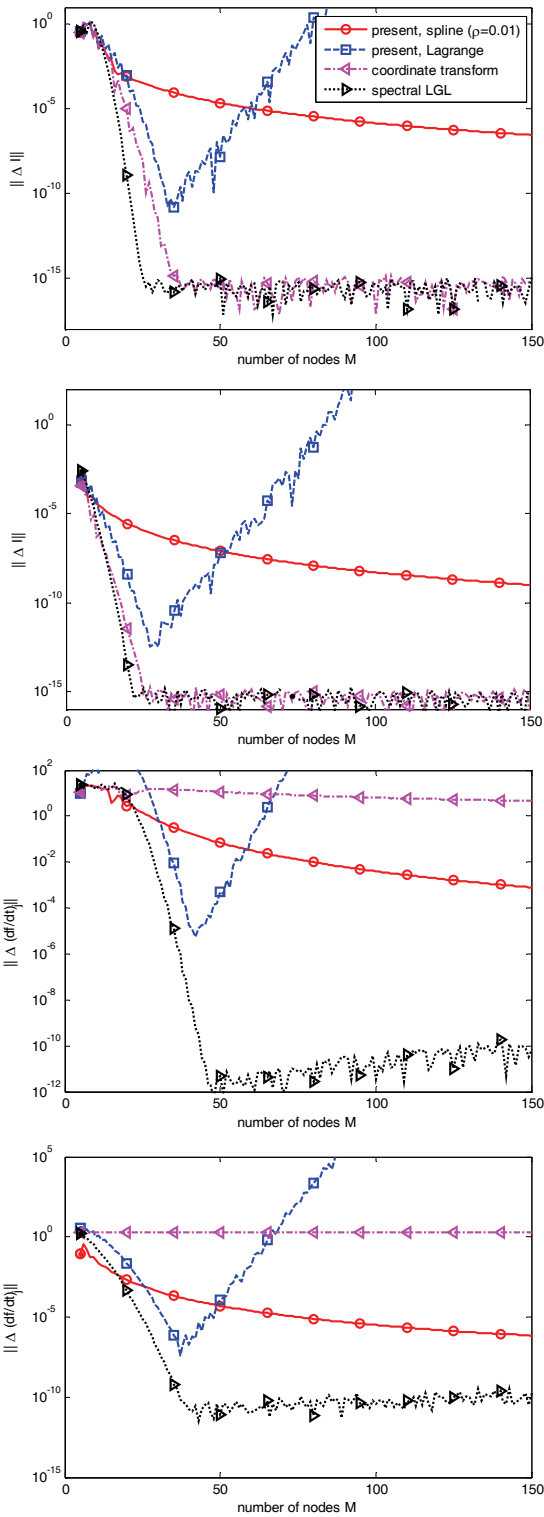


Fig. 5. Computational errors for $f_1(t)$ and $f_2(t)$.

and differentiation using the coordinate transformation are estimated using (65) and (66). The coordinate transformation method shows nearly the same performance as the spectral method in predicting the integration as the number of nodes is increased. However, the differentiation computed with (66) contains a larger error than that with

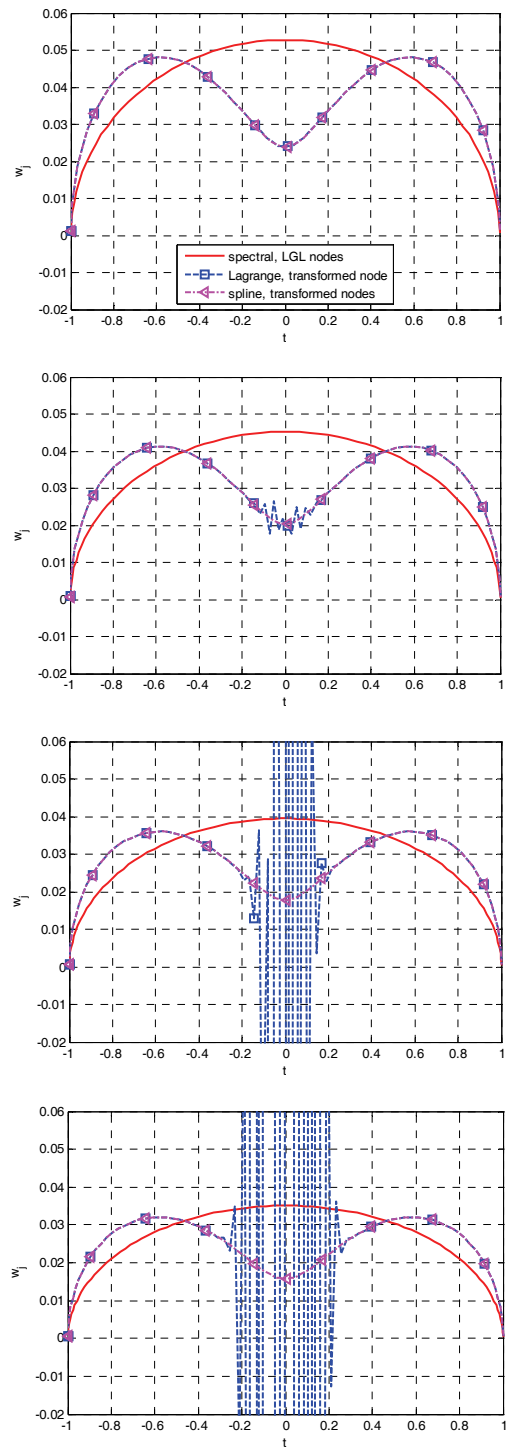


Fig. 6. Quadrature weight distribution with varying node number.

the present method when the tension spline interpolation is used. The present results with Lagrange interpolation show rapid increase in computational errors for both the differentiation and the integration with more than 40 nodes. Fig. 6 presents the effect of the number of nodes on the variation of the quadrature weights.

The oscillatory behaviors are made more apparent by

increasing the number of nodes, and some weights even become negative owing to poor interpolation accuracy, whereas the present method using the tension spline shows improvement in accuracy both for the integration and for the differentiation as the number of nodes is increased. Even though the same spectral accuracy cannot be obtained as that achievable with the spectral method using the LGL nodes, this method can be used for the problems, where reasonable accuracy is enough for its applications.

Finally, the applicability of the proposed method on

arbitrary nodes is investigated. Efforts to utilize arbitrary nodes include recent studies by Ross et al. [4] and Gong et al. [12] showing that convergence for the solution of nonlinear optimal control problems with pseudospectral methods can be guaranteed only when all quadrature weights are positive. They pointed out that analyses with more than ten uniform nodes might fail, because at least one of the weights becomes negative when Lagrange polynomials are used on uniform nodes. Regarding the positivity requirement for the weights, the spline interpolation is more robust than Lagrange interpolation, as shown in Figs. 5 and 6, because the corresponding weights retain positivity even up to 150 nodes. For these reasons, we use the tension splines in the present investigation. Fig. 7 shows the prediction accuracy with different tension parameters for $f_1(t)$ and $f_2(t)$. As expected, uniform nodes present much worse accuracy than the LGL points. However, absolute prediction errors using uniform nodes gradually decrease as the number of nodes is increased. In the cases when these errors are allowable with a reasonable number of nodes, uniform nodes can also be used, meaning that arbitrary nodes can be selected for real applications in the mentioned conditions.

Two aspects are considered for a further work. First, various grid topologies can be used to enhance both accuracy and convergence in the iterative solutions using the flexible selection of nodes allowed by the proposed method. The local adaptive grid and multi-grid techniques developed for computational fluid dynamics can be promising topologies for those purposes.

Second, the proposed enhanced computation method can provide theoretic background in analyzing and resolving a group of optimal control and state estimation problems with the help of NOCP formulation. Application to optimal control problem with practical inequality constraints is already under investigation through pseudo-spectral method [15] and can be further extended to the optimal estimation problems with non-stochastic system models.

5. Conclusion

This paper proposed a unified approach to the computation of the coefficients for quadrature formulas, such as the quadrature weights, integration matrix, and differentiation matrix. These formulas were derived from the exact integration and differentiation of the piecewise continuous polynomials in the global spline interpolation formula. To demonstrate the usefulness of the present approach, it was shown that Lagrange and tension spline interpolation can be transformed into the global spline interpolation formula, regardless of the type of nodes. The present method was adopted to predict the coefficients of the quadrature formulas for the Legendre-Gauss, Legendre-Gauss-Lobatto, and Chevyshev-Gauss-Lobatto points with levels of precision equivalent to those attainable with

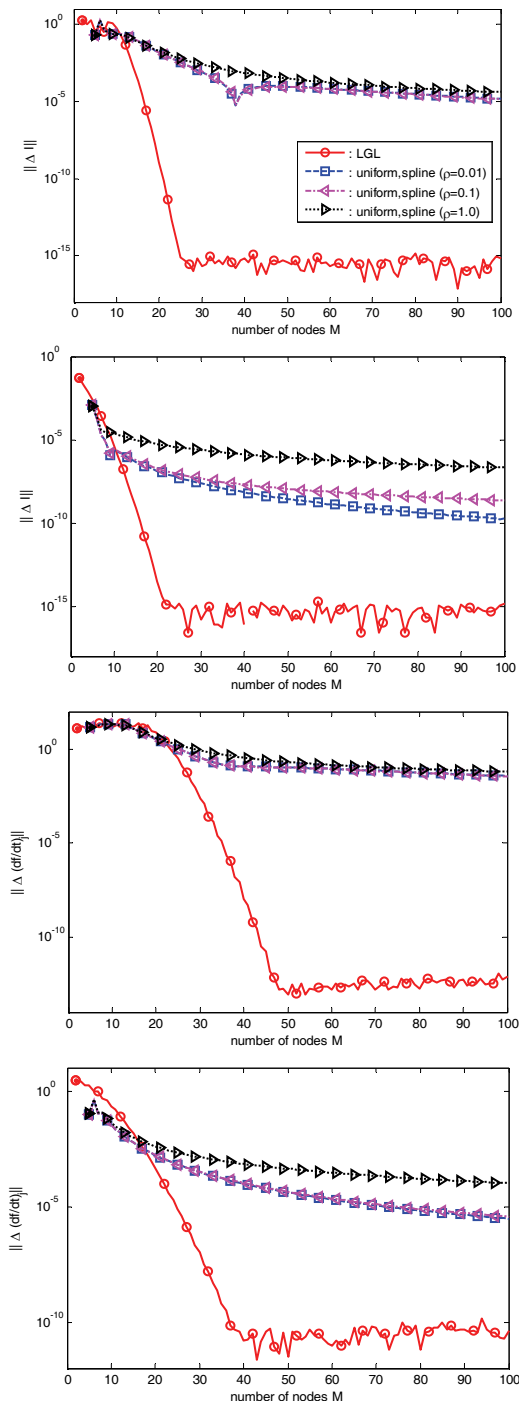


Fig. 7. Computational errors for $f_1(t)$ and $f_2(t)$.

traditional spectral methods. The proposed method was validated through the application of the present method to carefully selected test functions. The results are compared with those using the traditional quadrature formula. The integrations for all functions can be approximated in nearly the same level of accuracy with both methods. However, the differentiations can be more accurately predicted with the present method than with the traditional method.

As an effort to extend the applicability of the proposed method to solutions on arbitrary nodes, nodes generated using an analytic coordinate transformation method and uniform nodes were selected to show the accuracy achievable with the proposed method. The coordinate transformation method can be used to compute the quadrature integration with high precision equivalent to that obtained using the spectral method at the Legendre-Gauss-Lobatto nodes. However, it revealed poor accuracy in predicting the differentiation owing to high interpolation error when Local Lagrange interpolation was adopted.

For the proposed method, spline interpolation outperformed Lagrange interpolation in that its accuracy in predicting the integration and differentiation improved uniformly as the number of nodes increased for both sets of nodes. Therefore, the proposed method using spline interpolation can be used on arbitrary nodes when the incurred errors are allowable for real applications. The flexibility in selecting computational nodes allows the use of advanced grid topologies, such as the local adaptive grid and multi-grid techniques, to enhance both accuracy and convergence in iterative solution processes. As a result of this study, a unified method of computing quadrature weights, the integration matrix, and the differentiation matrix corresponding to the quadrature nodes for the spectral methods has been established. Furthermore, the proposed methods are applicable to arbitrarily distributed nodes.

Acknowledgements

This work was supported by the Faculty Research Fund of Konkuk University in 2013.

References

- [1] H.-H. Kim, C.-J. Lee, J.-B. Park, J.-R. Shin, S.-Y. Jeong, 'Two-Terminal Numerical Algorithm for Single-Phase Arcing Fault Detection and Fault Location Estimation Based on the Spectral Information,' *Journal of Electrical Engineering & Technology*, Vol. 3, No. 4, Dec. 2008, pp. 460-467.
- [2] Le Chi Kien, 'Numerical Calculations and Analyses in Diagonal Type Magneto-hydro-dynamic Generator,' *Journal of Electrical Engineering & Technology*, Vol. 8, No. 6, Nov. 2013, pp. 1365-1370.
- [3] M.-H. Kim, H.-B. Lee, H.-S. Kim, J.-K. Byun, 'A New Unified Design Environment for Optimization of Electric Machines Based on Continuum Sensitivity and B-Spline Parametrization,' *Journal of Electrical Engineering & Technology*, Vol. 6 No. 4, Jul. 2011, pp. 513-518.
- [4] I. M. Ross and M. Karpenko, "A Review of Pseudo-spectral Optimal Control: From theory to flight," *Annual Reviews in Control*, Vol. 36, Issue 2, pp. 182-197, December 2012.
- [5] D. Benson, "A Gauss Pseudo-spectral Transcription for Optimal Control," *Ph.D. Thesis, MIT*, Department of Aeronautics and Astronautics, Nov. 2004.
- [6] G. T. Huntington, "Advancement and Analysis of a Gauss Pseudo-spectral Transcription for Optimal Control Problems," *Ph. D. Thesis, MIT*, Jun. 2007.
- [7] P. Grandclément, "Introduction to Spectral Methods," *European Astronomical Society Publications Series*, Vol. 21, pp. 153-180, 2006. DOI: <http://dx.doi.org/10.1051/eas:2006112>.
- [8] M. Maleki, I. Hashim, and S. Abbasbandy, "Analysis of IVPs and BVPs on Semi-Infinite Domains via Collocation Methods," *Journal of Applied Mathematics*, Vol. 2012, Article ID 696574, 21 pages, 2012. doi:10.1155/2012/696574
- [9] P. Williams, "A Gauss-Lobatto Quadrature Method for Solving Optimal Control Problems," *Australian and New Zealand Industrial and Applied Mathematics Journal*, Vol. 47, 2005, pp. 101-115.
- [10] S. E. El-gendi, "Chebyshev Solution of Differential, Integral and Integro-Differential Equations," *The Computer Journal*, Vol. 12, No. 3, pp. 282-287, 1969. doi: 10.1093/comjnl/12.3.282.
- [11] C.-J. Kim, S. K. Sung, S. H. Park, S. N. Jung, and T. S. Park, "Numerical Time Scale Separation for Non-linear Optimal Control Analyses with Applications to Rotorcrafts," *AIAA Journal of Guidance, Control and Dynamics*, Vol. 37, No. 2, March 2014, pp. 658-673.
- [12] Q. Gong, I. M. Ross, and F. Fahroo, "Pseudospectral Optimal Control on Arbitrary Grids," in *Proc. of AAS / AIAA Astrodynamics Specialist Conference*, Pittsburgh, PA, August 9-13, 2009.
- [13] J. Waldvogel, "Fast Construction of the Fejér and Clenshaw-Curtis Quadrature Rules," *BIT Numerical Mathematics*, Vol. 46, Issue 1, pp. 195-202, Mar. 2006. DOI:10.1007/s10543-006-0045-4.
- [14] J. M. Barden, "A Modified Clenshaw-Curtis Quadrature Algorithm," *Worcester Polytechnic Institute, Master Thesis*, Department of Science In Applied Mathematics, Apr. 2013.
- [15] C. -J. Kim and S. Sung, "Efficient Spline Transcription Techniques for Nonlinear Optimal Control Analyses Using a Pseudospectral Framework," *IEEE Trans. Control Systems Technology*, in press, 2015.
- [16] S. Pruess, "Properties of Splines in Tension," *Journal of Approximation Theory*, Vol. 17, pp. 86-96, 1976.



Chang-Joo Kim He is an associate Professor in the Department of Aerospace Information Engineering at Konkuk University, Korea. His research interests include nonlinear optimal control, helicopter flight mechanics and helicopter system design.



Joon-Goo Park He received his Ph.D. degree in Electronic Engineering from Seoul National University in 2001. He is presently a professor at the School of Electrical Engineering at Kyungpook National University. His research interests include GPS, mobile software navigation and algorithm analysis.



Sangkyung Sung He is an associate Professor in the Department of Aerospace Information Engineering at Konkuk University, Korea. His research interests include inertial sensors, aerospace system control, navigation and estimation.