

Gaussian quadrature for splines via homotopy continuation: rules for C^2 cubic splines

Michael Barton^{a,*}, Victor Manuel Calo^{a,b}

^a*Numerical Porous Media Center,*

King Abdullah University of Science and Technology, Thuwal 23955-6900, KSA

^b*Applied Mathematics & Computational Science and Earth Science & Engineering,*
King Abdullah University of Science and Technology, Thuwal 23955-6900, KSA

Abstract

We introduce a new concept for generating optimal quadrature rules for splines. Given a target spline space where we aim to generate an optimal quadrature rule, we build an associated source space with known optimal quadrature and transfer the rule from the source space to the target one, preserving the number of quadrature points and therefore optimality. The quadrature nodes and weights are, considered as a higher-dimensional point, a zero of a particular system of polynomial equations. As the space is continuously deformed by modifying the source knot vector, the quadrature rule gets updated using polynomial homotopy continuation. For example, starting with C^1 cubic splines with uniform knot sequences, we demonstrate the methodology by deriving the optimal rules for uniform C^2 cubic spline spaces where the rule was only conjectured heretofore. We validate our algorithm by showing that the resulting quadrature rule is independent of the path chosen between the target and the source knot vectors as well as the source rule chosen.

Keywords: Gaussian quadrature, B-splines, well-constrained polynomial system, polynomial homotopy continuation

1. Introduction

Numerical integration of univariate functions is a fundamental mathematical task which is a subroutine of many complex algorithms and is typically frequently invoked. Naturally, such an integration (or quadrature) rule must be as efficient as possible. We derive a new class of quadrature rules that are optimal in the sense that they require the minimal number of function's evaluations.

*Corresponding author

Email addresses: `Michael.Barton@kaust.edu.sa` (Michael Barton),
`Victor.Calo@kaust.edu.sa` (Victor Manuel Calo)

A quadrature rule, or shortly a *quadrature*, is an *m-point rule*, if m evaluations of a function f are needed to approximate its weighted integral over a closed interval $[a, b]$

$$\int_a^b w(x)f(x) dx = \sum_{i=1}^m \omega_i f(\tau_i) + R_m(f), \quad (1)$$

where w is a fixed non-negative *weight function* defined over $[a, b]$. The rule is required to be *exact*, that is, $R_m(f) \equiv 0$ for each element of a predefined linear function space \mathcal{L} . The rule is said to be *optimal* if m is the minimal number of *weights* ω_i and *nodes* τ_i , points at which f has to be evaluated.

For the space of polynomials, the optimal rule is known to be the classical Gaussian quadrature [7] with the order of exactness $2m - 1$, that is, only m evaluations are needed to exactly integrate any polynomial of degree at most $2m - 1$. Consider a sequence of polynomials $(p_0, p_1, \dots, p_m, \dots)$ that form an orthogonal basis with respect to the scalar product

$$\langle f, g \rangle = \int_a^b f(x)g(x)w(x)dx. \quad (2)$$

The quadrature points are the roots of the m -th orthogonal polynomial p_m which in the case when $w(x) \equiv 1$ is the degree- m Legendre polynomial [13].

In this paper, we focus on piece-wise cubic polynomials, cubic splines, but the methodology is general and could be used for higher degrees as well. A univariate space of cubic splines is uniquely determined by its *knot vector*. This knot vector is a non-decreasing sequence of real numbers called *knots* and the multiplicity of each knot determines the smoothness between the cubic pieces. To simplify the argument, we first study uniform knot vectors with all interior knots with *uniform multiplicity*. In particular, we investigate knot vectors with single interior knots which yield C^2 cubic splines and knot vectors with double interior knots which give C^1 cubic splines. For a more detailed introduction on splines, we refer the reader, e.g., to [4–6].

The quadrature rules for splines have been studied since late 50's [9, 10, 12]. Firstly conjectured by Schoenberg [12], later proved by Micchelli and Pinkus [9], the conditions of the existence and uniqueness of the optimal (Gaussian) rule have been derived. For spline spaces with maximum continuity (e.g., C^2 cubic splines), Micchelli and Pinkus [9] proved that there always exists a *unique* Gaussian quadrature rule. Their result, however, also reveals a nice phenomenon: the number of optimal nodes stays fixed as long as the number of interior knots stays constant. Therefore if one desires to derive a class of quadrature rules with the same number of nodes, spline spaces with higher continuity must have adequately more sub-intervals (elements) than spaces of lower continuity. This is natural since splines of lower continuity are limits of the higher continuous ones, when merging continuously two (or several) knots together, and their result is in agreement with this fact.

For spaces with lower continuity, or when boundary constraints are involved, the rule is not guaranteed to be unique. Only spaces with fixed continuities

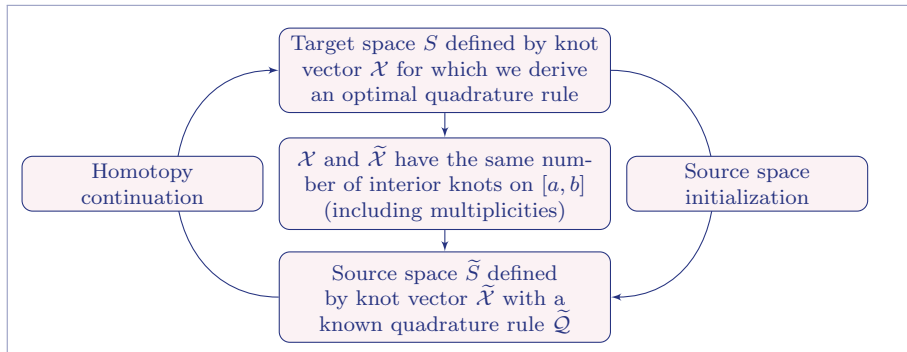


Figure 1: Spline space initialization flowchart

were studied in [9]. To the best of our knowledge, the results on existence and uniqueness are not known for spaces with mixed continuities, we refer to these as knot vectors with mixed multiplicities. For example, for cubic splines, this includes to knot vectors with both single and double knots. In this work, we derive quadratures for this kind of mixed continuity spaces and show numerically that optimal quadratures exist.

We use the observation that a spline space with multiple knots is a limit case of a spline space with the same number of single knots (when counting the multiplicities) when two, or several, knots merge. Abstracting this merging as a continuous transition between the two knot vectors of the same cardinality, the *source* and the *target* ones, the corresponding spline spaces continuously evolve from one into the other. The quadrature rule also depends *continuously* on the spline space since the quadrature rule can be seen as a zero (root) of a certain polynomial system and an infinitesimal change of the system does not significantly change the root. Based on these facts, we propose a new methodology that for a given (target) spline space S generates an associated source space \tilde{S} where the Gaussian quadrature rule is known. These spaces are defined above knot vectors \mathcal{X} and $\tilde{\mathcal{X}}$, respectively, and the quadrature rule \tilde{Q} of \tilde{S} is numerically traced as $\tilde{\mathcal{X}}$ evolves into \mathcal{X} , see Fig. 1.

The rest of the paper is organized as follows. Section 2 briefly overviews the homotopy continuation of polynomial systems and Section 3 summarizes a few basic properties of cubic splines. In Section 4, we introduce a homotopy-continuation-based algorithm and discuss the results and the validity of the new quadratures obtained in Section 5. Finally, Section 6 summarizes our conclusions and describes future research directions.

2. Homotopy continuation for polynomial systems

Polynomial homotopy continuation (PHC) is a numerical scheme that solves polynomial systems of equations. This approach was introduced to solve the problem of movability of kinematic mechanisms [15] where the variables are the

free parameters of a certain mechanism tied together by a set of polynomial constraints. However, the unknowns and the constraints may relate to an arbitrary problem. As our Gaussian quadrature rules are derived using a variation of this method, for the sake of completeness, we now briefly review the ideas used in homotopy continuation. For a detailed explanation, we refer the reader to the book [14].

Consider a well-constrained $2m \times 2m$ polynomial system

$$\mathcal{F}(\mathbf{x}) = \mathbf{0}, \quad \mathbf{x} \in \Omega \subset \mathbb{R}^{2m}, \quad (3)$$

where the domain Ω is a hypercube in \mathbb{R}^{2m} , i.e., $\Omega = [\underline{x}_1, \bar{x}_1] \times \cdots \times [\underline{x}_{2m}, \bar{x}_{2m}]$.

Let K be an upper bound of the number of *real* roots of \mathcal{F} in Ω and let us consider a simpler system $\tilde{\mathcal{F}}(\mathbf{x}) = \mathbf{0}$ defined over some $\tilde{\Omega}$ with *exactly* K real roots $\{\mathbf{r}_1, \dots, \mathbf{r}_K\}$, that is,

$$\tilde{\mathcal{F}}(\mathbf{r}_i) = \mathbf{0}, \quad i = 1, \dots, K \quad (4)$$

For now, let us assume $\Omega = \tilde{\Omega}$. The roots $\{\mathbf{r}_1, \dots, \mathbf{r}_K\}$ are known, in fact they are chosen as an input since the system $\tilde{\mathcal{F}}$ is as simple as possible, see [14]. Consider now a continuous deformation of the known system into the desired one

$$\tilde{\mathcal{F}}(\mathbf{x}) \rightarrow \mathcal{F}(\mathbf{x}). \quad (5)$$

Let us denote by \mathcal{H} a one parameter family of systems created, as $\tilde{\mathcal{F}}$ is transformed into \mathcal{F} . The parameter characterizing the transformation can be thought of as time or pseudo-time t . Thus

$$\mathcal{H}(\mathbf{x}, t) = \mathbf{0}, \quad \mathbf{x} \in \Omega, \quad t \in [0, 1], \quad (6)$$

where $\mathcal{H}(\mathbf{x}, 0)$ is the artificially constructed system $\tilde{\mathcal{F}}(\mathbf{x}) = \mathbf{0}$, for which the roots are known, and $\mathcal{H}(\mathbf{x}, 1)$ is the given system (3) which we seek to solve. Consider now the roots (4) as a function of time, $\mathbf{r}_1(t), \dots, \mathbf{r}_K(t)$, and think of them as trajectories (curves) in \mathbb{R}^{2m} . As t runs over $[0, 1]$, some of these roots may vanish, which corresponds to the fact that (3) may have less real roots than K , but no new trajectory may rise. The transformation is guaranteed not to introduce new roots.

Typically, the source and the target systems are blended in a linear fashion, corresponding to the shortest path when deforming one polynomial system into another. However, one can consider different paths between the systems [14].

3. Transition between C^1 and C^2 cubic splines

We recall several properties of spline basis functions. We consider a uniform knot vector

$$\tilde{\mathcal{X}}_n = (a = \tilde{x}_0, \tilde{x}_1, \tilde{x}_1, \dots, \tilde{x}_{n-1}, \tilde{x}_{n-1}, \tilde{x}_n = b) \quad (7)$$

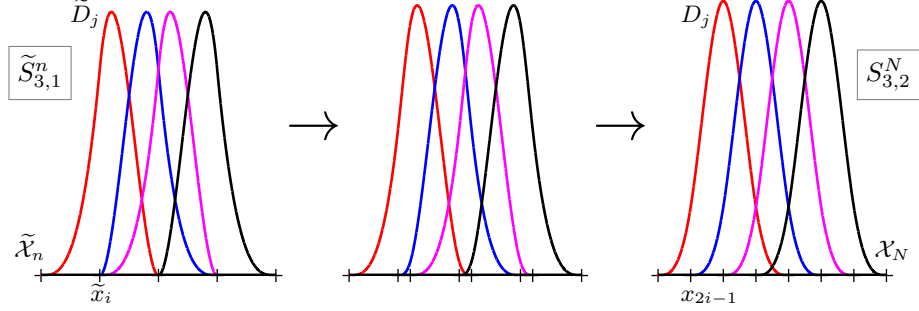


Figure 2: Continuous transformation of cubic spline spaces. The uniform knot sequence with knots of multiplicity two is transformed to a uniform knot sequence consisting of single knots. Four corresponding basis functions of the source space $\tilde{S}_{3,1}^n$, an intermediate space, and the target space $S_{3,2}^N$ are shown.

on the interval $[a, b]$, that is, each of the $n-1$ interior knots has multiplicity two. We define $h := \frac{b-a}{n} = \tilde{x}_k - \tilde{x}_{k-1}$ for all $k = 1, \dots, n$. Let us denote $N := 2n - 1$ and consider a uniform knot vector

$$\mathcal{X}_N = (a = x_0, x_1, \dots, x_{N-1}, x_N = b), \quad (8)$$

each knot of multiplicity exactly one and define $H := \frac{b-a}{N} = x_k - x_{k-1}$ for all $k = 1, \dots, N$. We denote by π_3 the space of polynomials of degree at most 3 and define $\tilde{S}_{3,1}^n$, the linear space of cubic splines over a uniform knot sequence $\tilde{\mathcal{X}}_n$ as

$$\tilde{S}_{3,1}^n = \{f \in C^1[a, b] : f|_{(\tilde{x}_{k-1}, \tilde{x}_k)} \in \pi_3, k = 1, \dots, n\}. \quad (9)$$

Similarly we denote by $S_{3,2}^N$ the linear space of cubic splines over a uniform knot sequence \mathcal{X}_N

$$S_{3,2}^N = \{f \in C^2[a, b] : f|_{(x_{k-1}, x_k)} \in \pi_3, k = 1, \dots, N\}. \quad (10)$$

The dimension of both spaces is $2n + 2 = N + 3$. That is, the total number of interior knots is the same for both spaces, while the number of non-zero knot spans is different.

Remark 1. The uniform knot sequences $\tilde{\mathcal{X}}_n$ and \mathcal{X}_N are, in this work, taken as the source and the target knot sequences, respectively. Additionally, we consider the intermediate knot sequences generated as $\tilde{\mathcal{X}}_n$ continuously evolves to \mathcal{X}_N , see Fig. 2. Given a particular knot vector, which may contain both single and double knots, we also consider all the spaces of mixed continuities, not just (9) and (10).

Consider now the $\tilde{S}_{3,1}^n$ space. Similarly to [1, 3, 11], we work with the non-normalized B-spline basis. To define the basis, we consider \tilde{x}_0 and \tilde{x}_n as double knots and extend our knot sequence $\tilde{\mathcal{X}}_n$ with two extra double knots outside the interval $[a, b]$ that we set to be

$$\tilde{x}_{-1} = \tilde{x}_0 - h \quad \text{and} \quad \tilde{x}_{n+1} = \tilde{x}_n + h, \quad (11)$$

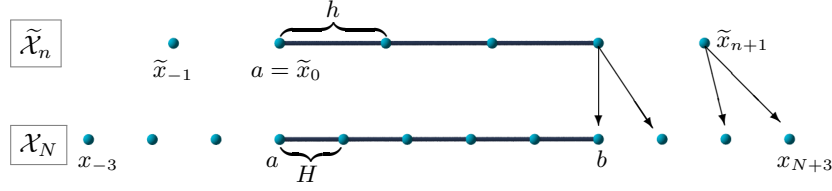


Figure 3: A target uniform knot vector with single knots \mathcal{X}_N for $N = 5$ elements and an associated source knot vector $\tilde{\mathcal{X}}_n$ with all knots of multiplicity two are shown. Both knot vector are of the same cardinality and have the same number of interior knots $i = 4$.

see Fig. 3. The choice of \tilde{x}_{-1} and \tilde{x}_{n+1} yields particular integral values for the first and last functions as discussed in (13). Denote by $\tilde{\mathbf{D}} = \{\tilde{D}_k\}_{k=1}^{2n+2}$ the basis of $\tilde{S}_{3,1}^n$ where

$$\begin{aligned}\tilde{D}_{2k-1}(t) &= [\tilde{x}_{k-2}, \tilde{x}_{k-2}, \tilde{x}_{k-1}, \tilde{x}_{k-1}, \tilde{x}_k](\cdot - t)_+^3 \\ \tilde{D}_{2k}(t) &= [\tilde{x}_{k-2}, \tilde{x}_{k-1}, \tilde{x}_{k-1}, \tilde{x}_k, \tilde{x}_k](\cdot - t)_+^3,\end{aligned}$$

where $[.]f$ stands for the divided difference and $u_+ = \max(u, 0)$ is the truncated power function. Direct computation gives

$$I[\tilde{D}_k] = \frac{1}{4} \quad \text{for } k = 3, 4, \dots, 2n, \quad (12)$$

where $I[f]$ stands for the integral of f over the interval $[a, b]$. We work with non-normalized basis functions, and therefore the integrals above are independent on the knot sequence. With the choice made in (11), direct integration gives

$$I[\tilde{D}_1] = I[\tilde{D}_{2n+2}] = \frac{1}{16} \quad \text{and} \quad I[\tilde{D}_2] = I[\tilde{D}_{2n+1}] = \frac{3}{16}. \quad (13)$$

Similarly to (11), we extend the knot sequence of \mathcal{X}_N by two triplets of single knots as

$$x_{-k} = x_0 - kH \quad \text{and} \quad x_{N+k} = x_N + kH, \quad k = 1, 2, 3. \quad (14)$$

and define $\mathbf{D} = \{D_k\}_{k=1}^{2n+2}$ the basis of $S_{3,2}^N$ where

$$D_k(t) = [\tilde{x}_{k-4}, \tilde{x}_{k-3}, \tilde{x}_{k-2}, \tilde{x}_{k-1}, \tilde{x}_k](\cdot - t)_+^3 \quad k = 1, \dots, 2n + 2.$$

We obtain

$$I[\tilde{D}_k] = I[D_k], \quad k = 4, \dots, 2n - 1 \quad (15)$$

and the six boundary integrals (three only due to symmetry) are computed directly by integrating D_1 , D_2 , and D_3 on $[a, b]$. These integrals change during continuation and therefore have to be recomputed for various t .

Now we consider a continuous transition between $\tilde{S}_{3,1}^n$ and $S_{3,2}^N$, see Fig. 2. Since the transition of the spline spaces is governed by the transformation of the corresponding knot vectors, consider the following mapping

$$\tilde{\mathcal{X}}_n \rightarrow \mathcal{X}_N, \quad (16)$$

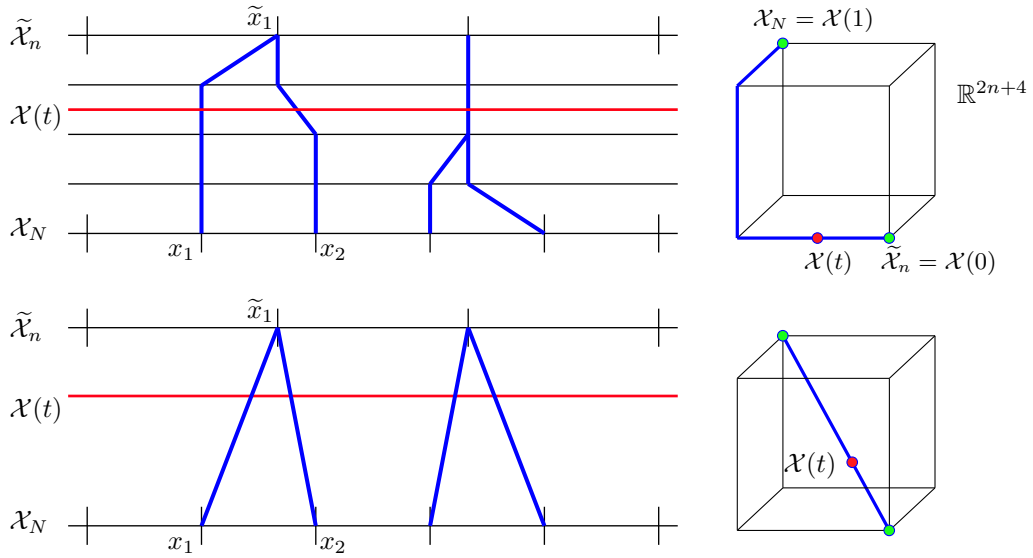


Figure 4: Continuous transformation of knots sequences. The source knot sequence $\tilde{\mathcal{X}}_n$ with double knots is changed over time and is transformed into the target uniform knot sequence with single knots, \mathcal{X}_N . The transformation is metaphorized as a path in $2n + 4$ dimensional space of free knots. Two particular paths are shown: whereas one transformation changes always only a single knot (top), the geodesic transformation (bottom) generates the shortest path in the space of knots and consequently the shortest transformation from $\tilde{\mathcal{X}}_n$ into \mathcal{X}_N .

including the six outer knots defined in (11) and (14), see Fig. 3. Due to the fact that we work with non-normalized basis functions, Eqns. (15) remain unchanged. The total number of knots is $2n + 6$, but since the two boundary knots are constrained to stay fixed, there is $2n + 4$ free knots. The transformation can be conceptualized as a curve between $\tilde{\mathcal{X}}_n$ and \mathcal{X}_N , two points in \mathbb{R}^{2n+4} , see Fig. 4. There exist infinitely many paths connecting the source and target knot vectors. In particular, we analyze two specific knot transformations: the first one sequentially changes only one knot whilst all the others remain unchanged; the second one simultaneously spreads all free knots from the source position to the target one in a linear fashion. Such a transition can be seen as a diagonal straight line connecting $\tilde{\mathcal{X}}_n$ and \mathcal{X}_N , i.e., a geodesic path when under the Euclidean metric on the vector of free knots, while the first one corresponds to a path along the edges of a $(2n + 4)$ -dimensional hypercube.

Remark 2. Alternatively, one can extend $\tilde{\mathcal{X}}_n$ and \mathcal{X}_N by adding different six knots than those in (11) and (14). For example, one can extend them in the open knot fashion by setting the multiplicity four to both boundary knots. In that case, the six (three due to symmetry) initial boundary integrals have to be computed accordingly.

4. Gaussian quadrature rules via homotopy continuation

In this section, we use polynomial homotopy continuation (PHC) to generate Gaussian quadrature rules for families of cubic spline spaces above arbitrary knot sequences.

The key observation is the following. Assume you have an exact and optimal quadrature rule for a spline space \tilde{S} defined above a knot vector $\tilde{\mathcal{X}}$ and consider a continuous transformation of $\tilde{\mathcal{X}}$ as a function of time, i.e., $\tilde{\mathcal{X}}(t)$. As a quadrature rule is a solution of a particular polynomial system defined above $\tilde{\mathcal{X}}$, the quadrature nodes and weights also change continuously. We use the PHC framework, to take an exact and optimal rule as the initial root of a certain polynomial system and trace this high-dimensional point (“follow the path of the root” in the PHC terminology, see [14]) while changing continuously $\tilde{\mathcal{X}}$. This idea is in accordance with the result of Micchelli and Pinkus [9] which states that there exists an optimal quadrature rule with m nodes such that

$$d + i + 1 = 2m \tag{17}$$

where d is the spline degree and i is the number of interior knots, counted including their multiplicities. Therefore m stays fixed as long as the number of interior knots remains constant. Homotopy continuation transfers the optimal quadrature rule from one space to another because the number of interior knots remains unchanged.

In this paper, we want to derive optimal quadrature rules for spaces of C^2 cubic splines, so for uniform knot vectors we have $S_{3,2}^N$ as our target spaces, see (10). The multiplicity of the interior knots is one and therefore Eq. (17) requires N to be odd. Further, we have $n = \lfloor \frac{N}{2} \rfloor + 1$ and set $\tilde{S}_{3,1}^n$ (9) as our source spaces. For these spaces, unique quadrature rules that are explicit were derived in [11]. The framework we present is general and one can derive new rules by applying PHC to any other appropriate source space where an optimal quadrature rule is known, e.g., C^1 quintic splines with uniform knot sequences [3].

4.1. Gaussian quadrature formula

We set our source space as $\tilde{S}_{3,1}^n$, see (9) and know, according to (17), that $m = n + 1$. The optimal source quadrature rule is of the form

$$\tilde{Q}_a^b[f] = \int_a^b f(t)dt = \sum_{i=1}^{n+1} \tilde{\omega}_i f(\tilde{\tau}_i) \tag{18}$$

and the nodes and weights can be computed by a recursion derived by Nikolov [11], see Fig. 5. Due to the equal dimensions of $\tilde{S}_{3,1}^n$ and $S_{3,2}^N$, the target rule requires the same number of nodes and therefore

$$Q_a^b[f] = \int_a^b f(t)dt = \sum_{i=1}^{n+1} \omega_i f(\tau_i). \tag{19}$$

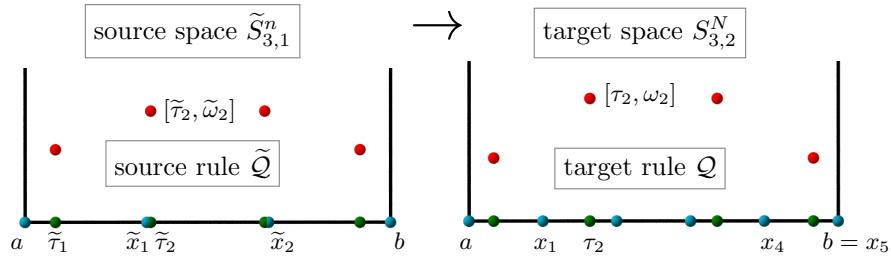


Figure 5: For a desired (target) space $S_{3,2}^N$ with $N = 5$ elements on $[a, b]$, an associated source space $\tilde{S}_{3,1}^n$, $n = 3$, is built (left). The source knot sequence (blue) is uniform with two interior knots \tilde{x}_1 and \tilde{x}_2 , each of multiplicity two. The target knot sequence has four single knots, so the total number of interior knots is unchanged; $i = 4$ in (17). Therefore both optimal rules require $m = n + 1$ nodes (green) and weights (red). The target rule (right) is derived via homotopy continuation, see Algorithm 1.

During the continuation, we transform the spline space $\tilde{S}_{3,1}^n$ to $S_{3,2}^N$ and accordingly the optimal rule $\tilde{Q} \rightarrow Q$. Therefore Q , represented by its nodes and weights, is a function of t . To simplify notation, if no ambiguity is imminent, we omit the time parameter and write τ_i instead of $\tau_i(t)$. The source rule is $\tilde{Q} = Q(0)$ and the target rule is $Q = Q(1)$. We denote by Q the rule (linear operator) and later in Section 4.3 use the symbol \mathbf{r} for a $2m$ -dimensional point, but they both refer to the same set of optimal nodes and weights. Before we proceed to our homotopy continuation setting, we need to establish notation that unifies the source and the target quadrature rules.

Let $\mathcal{X}_N = (a = x_0, x_1, \dots, x_{N-1}, x_N = b)$ be a knot vector consisting of N subintervals, some of the subintervals may degenerate to zero length in the case of a double knot and let $\{\tau_1, \dots, \tau_m\}$ be the quadrature points. We define a *nodal pattern* \mathbf{p} of the quadrature rule Q on \mathcal{X}_N as an N -dimensional vector where the i -th coordinate specifies the number of quadrature nodes inside the i -th sub-interval. We say \mathbf{p} is *regular* if no node coincides with a knot.

Example 4.1. Consider the Gaussian quadrature rule of Nikolov [11] for (9) in the case when n is odd, $n > 1$. The rule says every sub-interval contains exactly one node except the middle one which contains two, see Fig 5 for $n = 3$. Considering the multiplicities, the $\tilde{S}_{3,1}^n$ contains n sub-intervals of non-zero length and $(n - 1)$ degenerated subintervals (double knots). The nodal pattern is then

$$\mathbf{p} = \underbrace{(1, 0, \dots, 1, 0)}_{n-1}, \underbrace{2}_1, \underbrace{(0, 1, \dots, 0, 1)}_{n-1} \quad (20)$$

and the number of nodes is exactly $m = n + 1$. In the case when n is even, the middle node coincides with the middle knot. In such a case, we say the nodal pattern is *irregular*. We will discuss these patterns later in Section 4.3.

The nodal patterns describe the layout of the quadrature nodes with respect to the knots and, as long as the pattern does not change, the pattern of the polynomial system to solve remains unchanged. The crucial part of any optimal quadrature rule is to derive a correct nodal pattern. The result of Micchelli and

Pinkus [9] states the number of optimal nodes and also the knot span in which each particular node lies. However, one cannot conclude the nodal pattern for $S_{3,2}^N$ from their result. In our approach once we know the nodal pattern of the source rule, then we know the initial polynomial system and its root.

4.2. Homotopic setting

We are now ready to apply the PHC framework to the quadrature problem. The vector of unknowns consists of the quadrature nodes and weights

$$\mathbf{x} = (\tau_1, \dots, \tau_m, \omega_1, \dots, \omega_m), \quad \mathbf{x} \in \mathbb{R}^{2m},$$

our source polynomial system $\tilde{\mathcal{F}}$ expresses that the source rule $\tilde{\mathcal{Q}}$ exactly integrates the source basis $\tilde{\mathbf{D}}$, that is,

$$\tilde{\mathcal{Q}}_a^b(\tilde{D}_i) = I[\tilde{D}_i], \quad i = 1, \dots, 2m \quad (21)$$

and the source root \mathbf{r} that solves (21) is the quadrature rule of Nikolov [11]. The domain $\tilde{\Omega} \subset \mathbb{R}^{2m}$ is generated as follows. For this quadrature rule we know the nodal pattern, e.g., for n odd, $n = 2k + 1$, we have (20), therefore

$$(\tau_1, \dots, \tau_m) \in [\tilde{x}_0, \tilde{x}_1] \times \dots \times [\tilde{x}_k, \tilde{x}_{k+1}] \times [\tilde{x}_k, \tilde{x}_{k+1}] \times \dots \times [\tilde{x}_{n-1}, \tilde{x}_n] \quad (22)$$

and for the weights we use (a very rough) range $[0, b - a]$. Combined together, the source domain is

$$\tilde{\Omega} = \underbrace{[\tilde{x}_0, \tilde{x}_1] \times \dots \times [\tilde{x}_{n-1}, \tilde{x}_n]}_m \times \underbrace{[0, b - a] \times \dots \times [0, b - a]}_m. \quad (23)$$

Remark 3. Eq. (23) gives a very loose bound on ω_i , $i = 1, \dots, m$. However, this part of $\tilde{\Omega}$ is not affected by the nodal pattern and thus less important since it does not influence the change of the pattern of the polynomial system. A tighter bound could eventually serve as a better stopping criterion in cases, where the optimization is required to keep the root inside the domain (see Section 4.5) but we expect this would bring only marginal computational gains.

There are several differences between the homotopy framework we propose and the classical setting [14]:

- There is only one root that we follow: the quadrature rule $\tilde{\mathcal{Q}}$ of $\tilde{S}_{3,1}^n$. Moreover, we know the quadrature rules (the source rule, the intermediate ones and the target rule) are *unique* for geodesic knot transformations. This fact is a great advantage as there is no danger of numerical instabilities like jumping from one traced path to another as is the case in classical homotopy continuation methods.
- The systems do not require linear blending as in [14]. In our case, the continuous transition between the systems is governed by the change of the knot vector.

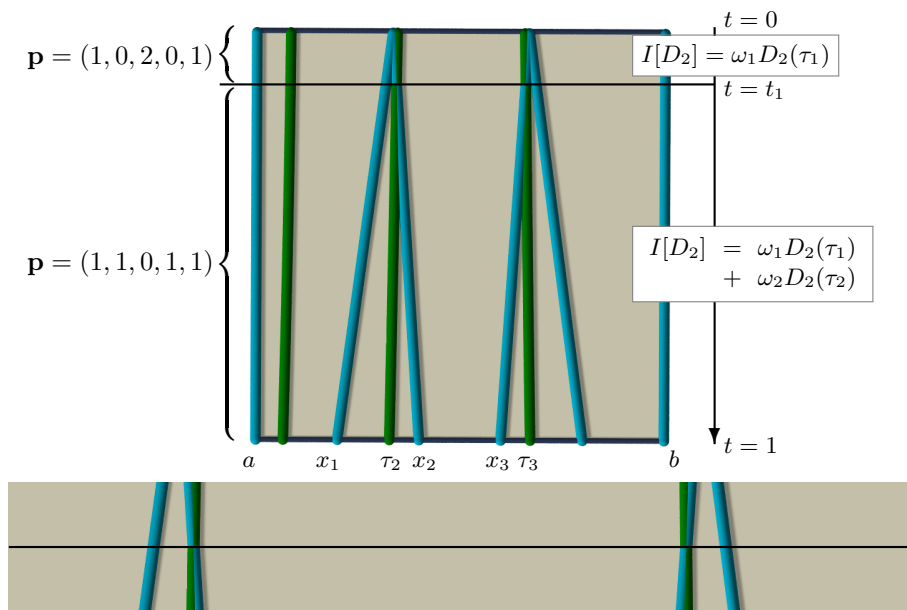


Figure 6: Nodal pattern change. Top: evolution of the nodes (green) and knots (blue) is shown. At time instant t_1 , the source nodal pattern changes which corresponds to the fact that τ_2 leaves (x_2, x_3) by crossing the knot x_2 . Due to symmetry, τ_3 leaves (x_2, x_3) and the new nodal pattern is $\mathbf{p} = (1, 1, 0, 1, 1)$. The new nodal pattern requires a change of the system (21), e.g., τ_2 starts affecting the basis function D_2 and the second equation of the system must be updated (right). Bottom: A zoom-in on the critical instant $t = t_1$.

- The domain $\tilde{\Omega}$ also changes in time because the range of every node is determined by the knot positions and these vary.
- The system is only *locally* polynomial. When a node leaves its subinterval, the nodal pattern of the quadrature rule changes, a new polynomial system has to be built and solved.

The last issue is important. The most difficult part of any optimal spline quadrature rule is to derive the correct layout of the nodes (the nodal pattern). Since the splines are piece-wise polynomials, the systems that we build and solve are also only “piece-wise polynomial”, i.e. locally polynomial and the main difficulty is to select the “right pieces”. However, homotopy continuation leads us to the correct nodal pattern automatically, by tracing down the trajectory of optimal quadrature rules as the knot pattern evolves. That is, the generated quadrature rules are exact (up to machine precision, see Section 5) and optimal for all the intermediate steps of the transformation.

4.3. Irregular nodal patterns

We now discuss the changing the nodal patterns, see Fig. 6. This is the situation when some (one or more) node(s) crosses the boundary of its subinterval.

That is, the node overlaps with a knot and the nodal pattern of the quadrature becomes irregular. We recall that $\Omega(t)$ is a hypercube in \mathbb{R}^{2m} and the system (21) can be seen as an intersection problem of $2m$ hypersurfaces inside $\Omega(t)$, which has at time $t = 0$ only one root $\mathbf{r}(0)$. During the continuation, as time evolves, $\Omega(t)$ is changing and so does the root $\mathbf{r}(t)$ that is being traced.

The situation when a node coincides with a knot corresponds to $\mathbf{r}(t)$ leaving $\Omega(t)$, i.e., there exist some $t_1 \in [0, 1]$ such that $\mathbf{r}(t_1)$ is on the boundary of the hypercube $\Omega(t_1)$, $\mathbf{r}(t_1) \in \bar{\Omega}(t_1)$. For simplicity, we assume that $\mathbf{r}(t_1)$ lies on a hyperface of $\Omega(t_1)$ which corresponds to the fact that only *one* node becomes a knot, $\tau_i(t_1) = x_j(t_1)$. Let the current sub-interval where $\tau_i(t)$ lies in for $t < t_1$ be $(x_{j-1}(t), x_j(t))$. We know the hyper-face that is being crossed, $\tau_i(t_1) = x_j(t_1)$, so the algorithm changes the nodal pattern by switching the sub-interval of τ_i from (x_{j-1}, x_j) to (x_j, x_{j+1}) . Then the system (21) is updated accordingly.

If $\mathbf{r}(t_1)$ lies on a hyper-edge, i.e., two (or more) nodes become knots at the same time, the situation is similar because for every node we know the current interval that the node leaves and the new one that it enters.

4.4. Quadrature rule tracing

The goal is to trace down a curve $\mathbf{r}(t) \in \mathbb{R}^{2m}$, $t \in [0, 1]$ knowing the initial point $\mathbf{r}(0)$, for example the quadrature rule of Nikolov [11]. We know the source and the target knot sequences, $\tilde{\mathcal{X}}_n$ (7) and \mathcal{X}_N (8), respectively. As the number of non-degenerate intervals varies from n to N , we simplify the notation by omitting the subscripts and write $\mathcal{X}(0) := \tilde{\mathcal{X}}_n$ and $\mathcal{X}(1) = \mathcal{X}_N$. We recall that the number of interior knots and total knots remains constant. We further discretize time and build $M - 1$ intermediate knot vectors

$$\{\mathcal{X}(t_i)\}_{i=0}^M \quad (0 = t_0, t_1, \dots, t_M = 1) \quad (24)$$

and consequently discretize $\mathbf{r}(t)$ as a polyline $\{\mathbf{r}^i\}_{i=0}^M$. We follow two different types of generation of the intermediate knot sequences, see Section 3. The geodesic path, where each knot moves linearly in t to its target position, and along-the-edge paths, where only a single knot moves at each time instant, see Fig. 4. The first path is unique; the number of the paths of the second kind equals the number of “bottom-to-top” paths on a hypercube in \mathbb{R}^{2n+4} , $(2n+4)!$. The latter paths generate spline spaces with mixed continuities (various knot multiplicities).

Remark 4. The source quadrature rule is unique and so is the target rule. Therefore the target rule should get derived independently on the path between $\mathcal{X}(0)$ and $\mathcal{X}(1)$. However, to the best of our knowledge, there are no theoretical results on the quadrature rules for splines with *mixed* continuities. The results of Micchelli and Pinkus [9] apply only for families of splines with uniform continuity. Our target quadrature rule gets derived independently of the path chosen, by parsing these spaces with mixed continuities as seen later in Section 5. This result provides numerical evidence for the existence of optimal rules for these mixed continuity spaces.

Algorithm 1GaussianQuadrature($[a, b], N$)

```
1: INPUT: compact interval  $[a, b]$  and odd number of uniform segments  $N$ 
2:  $n = \lfloor \frac{N}{2} \rfloor + 1$ ;
3:  $\mathbf{r}^0 :=$  the source rule  $\tilde{Q}$  on  $[a, b]$ ;
4: build  $\mathcal{F}^0(\mathbf{x}) = \mathbf{0}$  from (21) over  $\Omega^0$  (23);
5: build  $\{\mathcal{X}(t_i)\}_{i=0}^M$ , Section 3;
6: for  $i = 1$  to  $M$  do
7:   build  $\mathcal{F}^i(\mathbf{x}) = \mathbf{0}$  over  $\Omega^i$ ;
8:    $\mathbf{r}_{opt}^i :=$  solution of  $\mathcal{F}^i(\mathbf{x}) = \mathbf{0}$  by multivariate Newton-Raphson
   with initial guess  $\mathbf{r}^{i-1}$ ;
9:   if  $\mathbf{r}_{opt}^i \in \Omega^i$  then
10:    if  $\|\mathbf{r}_{opt}^i\| < \varepsilon$  then
11:       $\mathbf{r}^i := \mathbf{r}_{opt}^i$ ;          /* the quadrature rule for  $\mathcal{X}(t_i)$  */
12:    else
13:      subdivide  $\mathcal{X}(t_{i-1})$  and  $\mathcal{X}(t_i)$ ;          /* finer stepsize */
14:       $M := M + 1$ ; and go to line 7;
15:    end if
16:  else
17:    update  $\mathcal{F}^i$  and  $\Omega^i$  and go to line 7;  /* nodal pattern changes */
18:  end if
19: end for
20: OUTPUT:  $\mathbf{r}^M$ , the Gaussian quadrature rule for  $S_{3,2}^N$  on  $[a, b]$ ;
```

4.5. Implementation

In our implementation, the time-discretization of the intermediate knot vectors is uniform. In the case of the geodesic knot transformation, we sample uniformly the diagonal (source-target) of the hypercube, see Fig. 4. In the case of the along-the-edge transformation, we sample uniformly every hyperedge.

However, one can apply a non-uniform stepsize, e.g. this stepsize could get increased in cases where the nodes differ from the knots by more than a certain threshold. Reversely, a finer modification of the knot vector would be beneficial in cases when a node approaches a knot, i.e., in cases that precede a change of the nodal pattern. Such an analysis goes beyond the scope of this paper. We set $M = 200$ for the geodesic path and $M = 20(N - 1)$ for along-the-edge paths.

The tracing algorithm sequentially transforms the knot sequence $\mathcal{X}(t_i)$ and updates the system (21), $\mathcal{F}^i(\mathbf{x}) = \mathbf{0}$ considered over the domain Ω^i . The exact quadrature rule from the previous iteration $\mathbf{r}^{i-1} \in \mathbb{R}^{2m}$ is taken as an initial guess and a multivariate Newton-Raphson iterative scheme is applied to get the root \mathbf{r}_{opt}^i . If \mathbf{r}_{opt}^i lies in the current domain Ω^i , the next knot sequence $\mathcal{X}(t_{i+1})$ is taken with $\mathbf{r}^i := \mathbf{r}_{opt}^i$ and the algorithm proceeds iteratively. When $\mathbf{r}_{opt}^i \notin \Omega^i$, this implies that the nodal pattern has changed as explained in Section 4.3. In these instances, the algorithm swaps to a different polynomial system defined

above Ω^{new} . Then $\mathbf{r}^i := \mathbf{r}_{opt}^i$ and the Newton-Raphson optimization is applied again with the new system $\mathcal{F}^{new}(\mathbf{x}) = \mathbf{0}$. If the optimization fails, i.e., if $\|\mathbf{r}_{opt}^i\| > \varepsilon$ an additional intermediate knot sequence is inserted between $\mathcal{X}(t_{i-1})$ and $\mathcal{X}(t_i)$ and the original point \mathbf{r}^{i-1} and the original system $\mathcal{F}^i(\mathbf{x}) = \mathbf{0}$ are considered again. Otherwise the nodal pattern is updated and $\Omega^i := \Omega^{new}$. The numerical threshold was set $\varepsilon = 10^{-16}$. The whole procedure is summarized in Algorithm 1.

Remark 5. Algorithm 1 requires the number of elements N to be odd. This is because the associated source space has all interior knots of multiplicity two and therefore $N = n + n - 1$. This is in accordance with the result of Micchelli (17) that states the optimal rule for $S_{3,2}^N$ is guaranteed only for odd N .

5. Numerical examples

In this section, we show the results of our algorithm and derive Gaussian quadrature rules for various target spaces $S_{3,2}^N$. The evolution of the quadrature rules for $N = 11$ is shown in Fig. 7. Starting with the Gaussian quadrature rule for a spline space with uniformly distributed double knots as a source rule, the target rule for $S_{3,2}^N$ is traced by the homotopy continuation-based Algorithm 1, deriving the optimal rules also for all the intermediate spaces.

The evolution of the nodes is visualized as a set of planar curves, the evolution of weights is represented by the corresponding space trajectories. The error of the rule \mathcal{Q} is measured in terms of the Euclidean norm of the vector of the residues of the system (21), normalized by the dimension of the system

$$\|\mathbf{r}\| = \frac{1}{N+3} \left(\sum_{i=1}^{N+3} (\mathcal{Q}_a^b[D_i] - I[D_i])^2 \right)^{\frac{1}{2}}. \quad (25)$$

Our results coincide with those of [2] (up to twelve digits precision shown therein), where the correct layout of nodes was conjectured as

$$\mathbf{p} = (1, 1, 0, 1, 0, 1, \dots, 1, 0, 1, 0, 1, 1), \quad (26)$$

that is, the first two boundary elements contain a single node each, whilst the middle elements follow a node-gap pattern. Our algorithm yielded the same nodal pattern, see Fig. 8. Their algorithm is a trial-error scheme that chooses the first node and, under the assumption of the nodal pattern (26), computes iteratively the remaining nodes and weights. Based on the error of the rule, the first node is modified and the next iteration is invoked. Such an approach relies heavily on the initial node and, since the problem is highly non-linear. In our experience, there is no guarantee that the scheme of [2] will converge to a correct rule. We emphasize that our approach derives the layout of nodes (26) automatically and is general because it is not limited to only uniform target knot sequences.

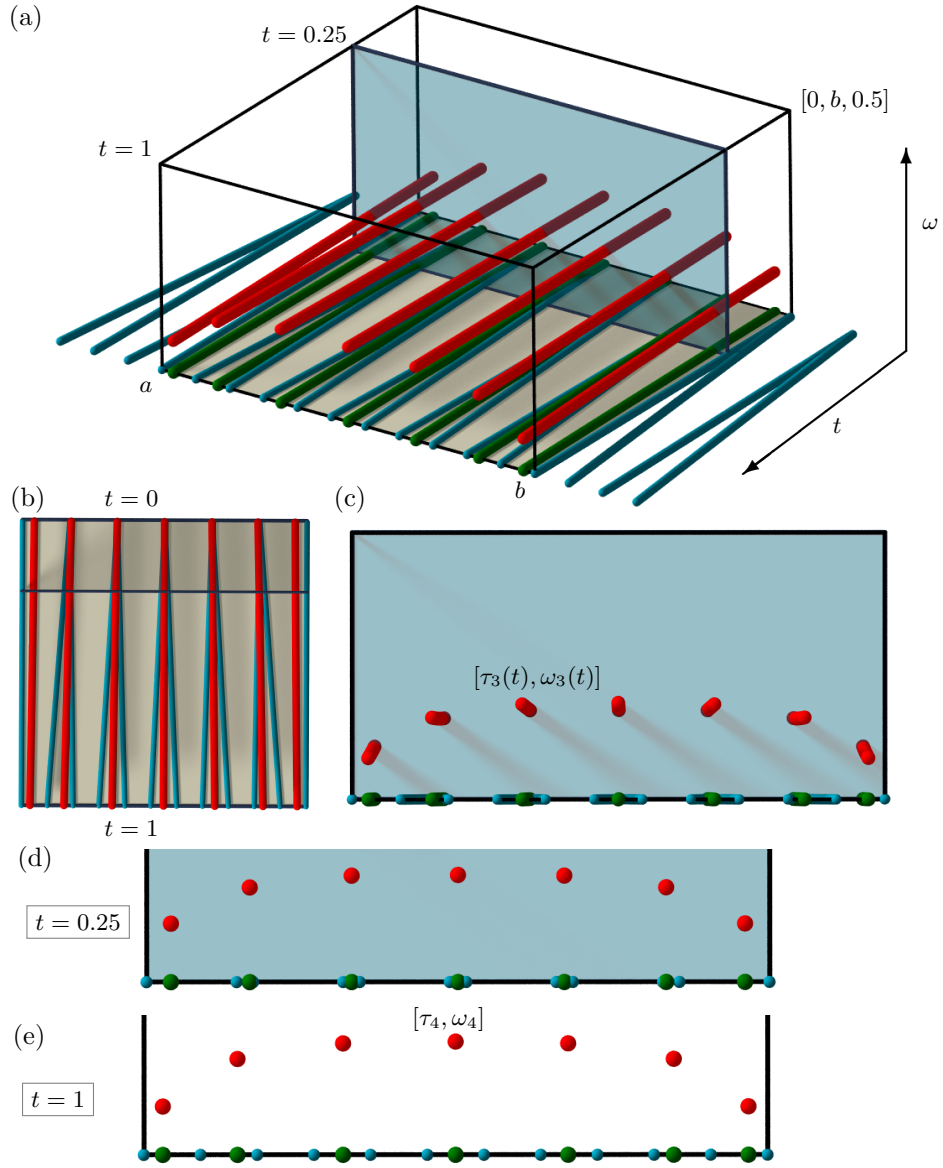


Figure 7: Geodesic continuation of the Gaussian quadrature rules for $N = 11$. (a) The evolution of a Gaussian quadrature rule from the source space $\tilde{S}_{3,1}^n$ with $n = 6$ is shown. As the knot vector changes (blue paths), the nodes (green) and the weights (red) move continuously from their source position ($t = 0$) to the target configuration ($t = 1$). The evolution of $m = n + 1 = 7$ Gaussian nodes and weights on $[a, b]$ is shown from the top view (b) and the side view (c). (d) A zoom-in on the Gaussian quadrature rule for a specific time instant $t = 0.25$. The current nodes (green), weights (red) and knots (blue) are shown. (e) The target uniform knot configuration and the target rule at $t = 1$.

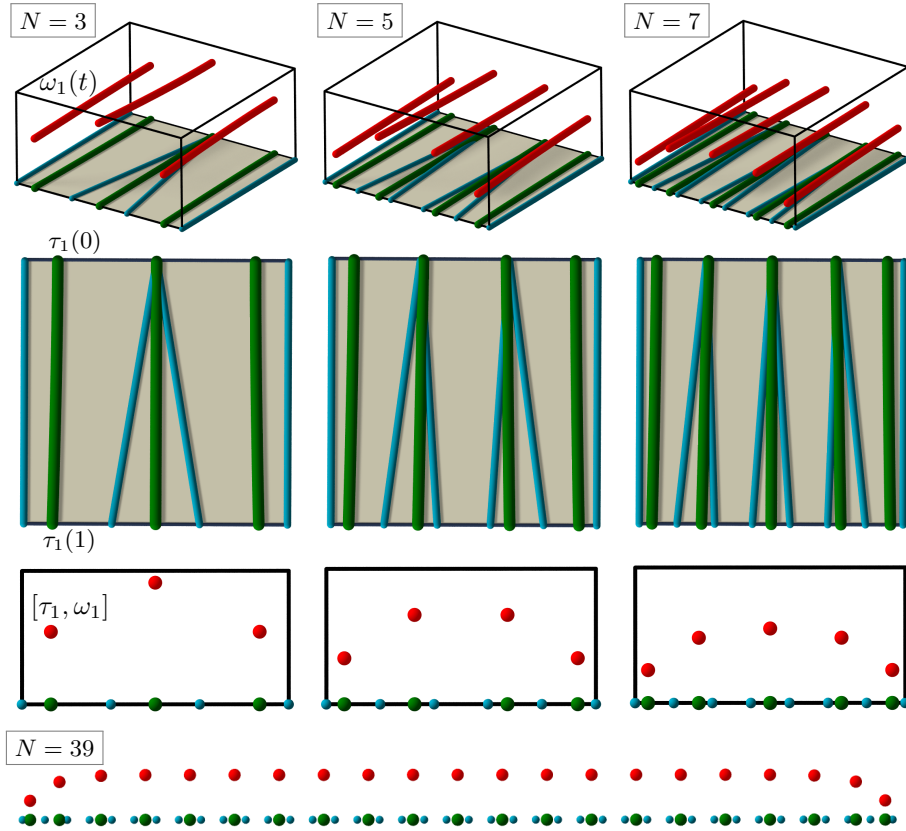


Figure 8: The evolution of Gaussian quadrature rules (19) for various source N are shown. Top: The evolution of the nodes (green), weights (red) for geodesic knot transformation (blue). Middle: The evolution of nodes in time and the target rules for the time instant $t = 1$. Bottom: The rule for $N = 39$; observe the convergence to the midpoint rule of Hughes et al. [8], cf. Table 1.

Also observe the convergence to the midpoint rule of Hughes et al. [8], see Table 1, where the limit weight for $N = 39$ ($n = 20$) is

$$\frac{2}{39} = 0.051282 \quad (27)$$

The data in Table 1 are shown with double-precision (16 decimal digits). Observe the slow convergence, e.g., for $N = 39$, the results meet the half-point-rule limit with only five digits of accuracy. This fact is not a limitation of our work, on the contrary, it shows that the midpoint rule is an approximation of our exact rule and can be used only when N is large enough.

The evolution of the Gaussian nodes and weights for various N is shown in Fig. 8. The modification of the knots is geodesic, see Section 3, i.e., every knot transforms to its target destination along the shortest possible path. As previously discussed in Section 4, the source and the target rules are both unique.

Table 1: Nodes and weights for Gaussian quadrature rules (19) with double-precision for uniform knot distribution for various target N are shown. To compare the results with algorithm of Nikolov [2], the interval was set as $[a, b] = [0, 1]$. Due to the symmetry, only the first $\lfloor \frac{N+1}{4} \rfloor + 1$ nodes and weights are displayed. The error $\|\mathbf{r}\|$ of the rule is measured as the Euclidean norm of the vector of the residues, normalized by the system's dimension $N + 3$, see (25).

i	τ_i	ω_i	$\ \mathbf{r}\ $
$N = 3$			
1	0.1086264370680297	0.2720231005023455	7.90^{-20}
2	0.5	0.4559537989953090	
$N = 5$			
1	0.0669578918742195	0.1698605936669416	1.04^{-19}
2	0.3275898516368645	0.3301394063330584	
$N = 7$			
1	0.0479188107803577	0.1216810800700958	1.95^{-18}
2	0.2358921494969001	0.2408185184939348	
3	0.5	0.2750008028719389	
$N = 9$			
1	0.0372757529111283	0.0946622477445919	2.08^{-18}
2	0.1835904624135774	0.1876252194189693	
3	0.3904233866079767	0.2177125328364388	
$N = 11$			
1	0.0304987043023585	0.0774523185174377	6.68^{-18}
2	0.1502181009517147	0.1535325192913209	
3	0.3195393932155687	0.1783894870783702	
4	0.5	0.1812513502257421	
\vdots			
$N = 39$			
1	0.0086022074347388	0.0218455595269063	1.02^{-17}
2	0.0423693959303822	0.0433045545577068	
3	0.0901289847662636	0.0503213631747089	
4	0.1410569521267253	0.0512021143533085	
5	0.1923101843694322	0.0512756766459810	
6	0.2435899416018961	0.0512815446928528	
7	0.2948718106031808	0.0512820110347811	
8	0.3461538474036372	0.0512820480845737	
9	0.3974358975351839	0.0512820510280155	
10	0.4487179487257872	0.0512820512617426	
11	0.5	0.0512820512788446	

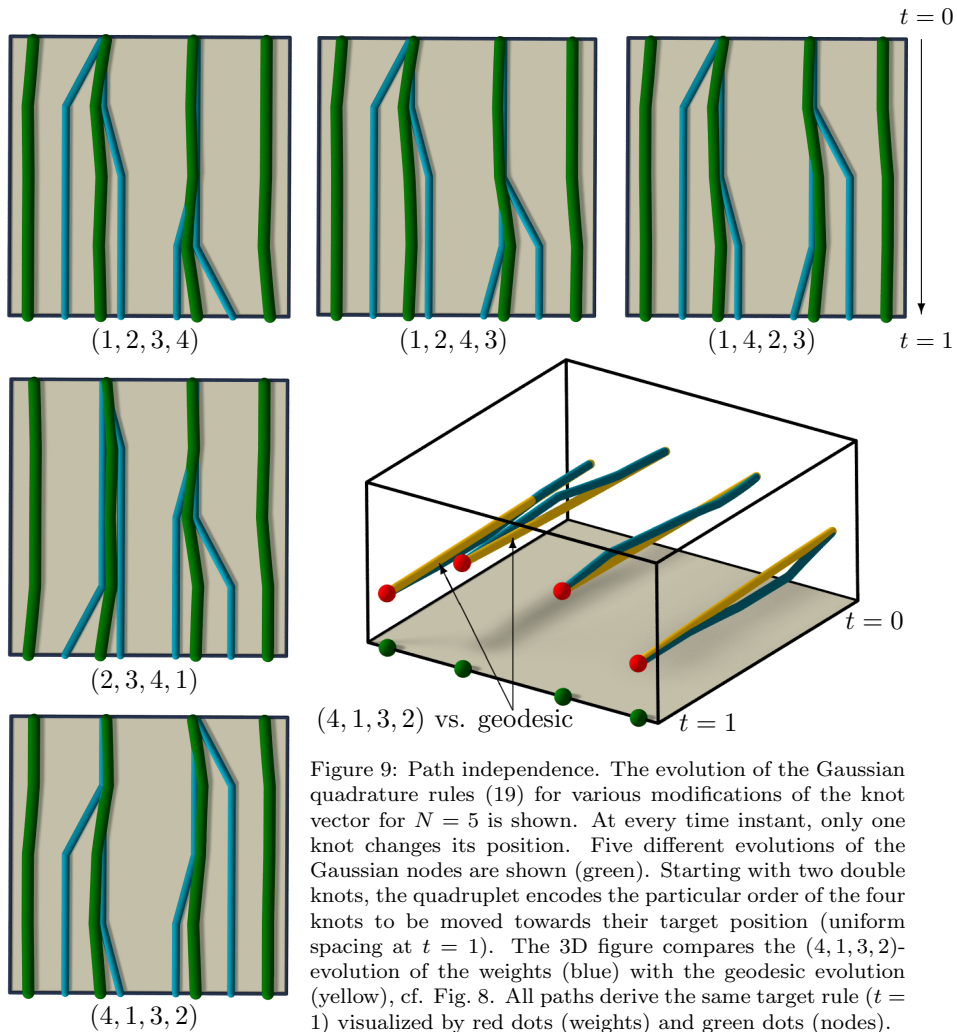


Figure 9: Path independence. The evolution of the Gaussian quadrature rules (19) for various modifications of the knot vector for $N = 5$ is shown. At every time instant, only one knot changes its position. Five different evolutions of the Gaussian nodes are shown (green). Starting with two double knots, the quadruplet encodes the particular order of the four knots to be moved towards their target position (uniform spacing at $t = 1$). The 3D figure compares the $(4, 1, 3, 2)$ -evolution of the weights (blue) with the geodesic evolution (yellow), cf. Fig. 8. All paths derive the same target rule ($t = 1$) visualized by red dots (weights) and green dots (nodes).

Therefore, any path deforming the uniform double knot sequence $\tilde{\mathcal{X}}_n$ (7) into the uniform knot sequence \mathcal{X}_N (8) must produce the same rule. The results of five random along-the-edge knot modifications are shown in Fig. 9. Contrary to the geodesic knot deformation, only one knot moves while the other stay fixed during the evolution. Depending on the order of how the knots are moved, there exist $(N + 5)!$ possible paths. The results of the target rule for various along-the-edge knot vector deformations are shown in Table 2. The results are independent on the path which validates the correctness of our algorithm.

An example of Gaussian rules for non-uniform target knot sequences is shown in Fig. 10. The space is kept C^1 , i.e., all the knots keep their multiplicity two. The target knot sequence was required to have larger elements close to the boundary and smaller ones in the interval's center. Unlike our previous result

Table 2: Path independence. The target quadrature rule obtained by different along-the-edge paths of the knot vector. The results for five paths shown in Fig. 9 are displayed with twenty digits accuracy. The target nodes and weights match up to eighteen digits.

	τ_1, τ_2	ω_1, ω_2
(1, 2, 3, 4)	0.06695789187421950918 0.32758985163686446374	0.16986059366694164265 0.33013940633305835725
(1, 2, 4, 3)	0.06695789187421950918 0.32758985163686446374	0.16986059366694164265 0.33013940633305835725
(1, 4, 2, 3)	0.06695789187421950918 0.32758985163686446374	0.16986059366694164265 0.33013940633305835725
(2, 3, 4, 1)	0.06695789187421950915 0.32758985163686446341	0.16986059366694164265 0.33013940633305835787
(4, 1, 3, 2)	0.06695789187421950915 0.32758985163686446342	0.16986059366694164265 0.33013940633305835721

[1] where finer spacing close to the boundary guaranteed the same nodal pattern for the whole family of symmetrically stretched knot sequences, here we can observe a change of the nodal pattern of the rule.

Throughout the paper, we used the rule of Nikolov [11] as our source rule. However, one may start, e.g., with the classical Gaussian rule for polynomials, see Fig. 11. In such a case, the knots have multiplicity four and one needs to count the proper number of initial elements to get the desired number of target elements N . This counting obeys the rule of Micchelli [9], see (17). The results show the stability of our Algorithm 1. The results from Fig. 11 both correspond to values shown in Table 1 for $N = 9$. Both target rules derived from different source rules match up to sixteen decimal digits.

6. Conclusion

We have introduced a new methodology to compute Gaussian quadrature rules. Starting with a known Gaussian quadrature rule for a specific spline space, the rule is interpreted as a point in a higher dimensional space, a zero of a specific polynomial system. A homotopy continuation-based algorithm has been presented that numerically traces the Gaussian quadrature rule as the knot vector, and consequently the whole spline space, is continuously modified. We have recovered the Gaussian quadrature rule of Nikolov [2] for the uniform C^2 cubic spline spaces, a rule that was derived under an assumption of the conjectured nodal pattern (26).

The continuation approach shows the connection between different Gaussian rules. This concept eliminates the need for the construction of particular rules

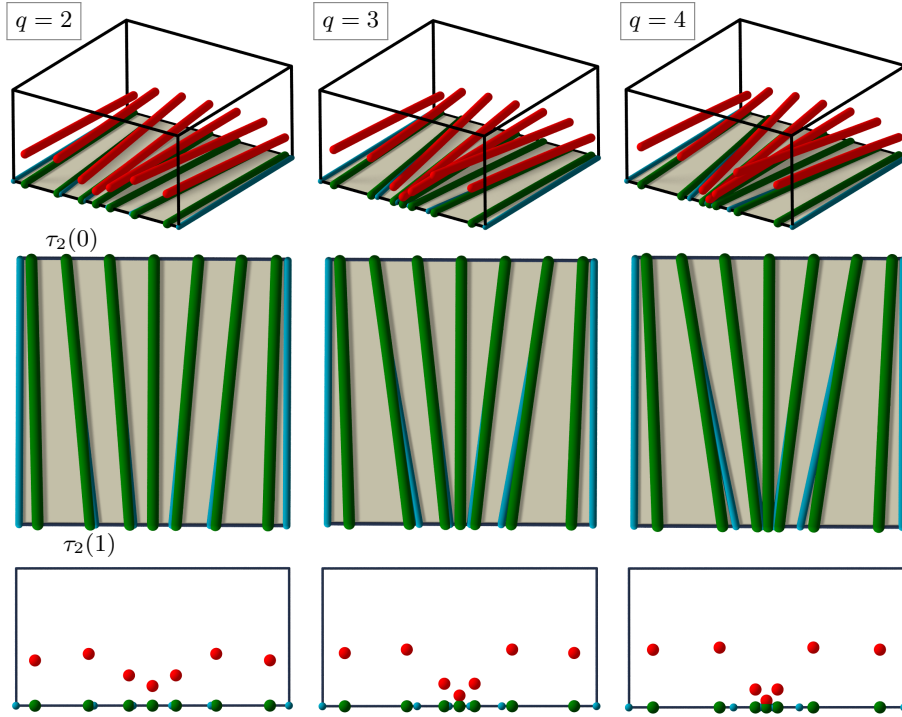


Figure 10: The evolution of the Gaussian rules for C^1 spaces with non-uniform target knot sequences. Five interior source double knots (blue) are moved towards the interval's center. The parameter q is the shrinking ratio; it is the ratio of lengths of two neighboring elements when moving towards the center. Top: The evolution of the rule in time visualized in 3D: the trajectories of the weights (red), nodes (green) and knots (blue) are shown. Middle: The evolution of the nodes. For all q shown, τ_2 starts ($t = 0$) in the second elements but ends ($t = 1$) in the first element. Bottom: The target Gaussian rules for particular non-uniform knot sequences.

for different kinds of non-uniform knot vectors. Thus, work such as [1, 3] is hereafter needed only to validate numerical results obtained with this method. We plan to further exploit the proposed technique to derive the Gaussian rules for the particular spline spaces of a high interest such as $S_{4,0}$ with non-uniform knot sequences, or $S_{6,1}$, spaces frequently appearing in the Galerkin method when building the mass and stiffness matrices.

References

- [1] R. Ait-Haddou, M. Bartoň, and V.M. Calo. Explicit Gaussian quadrature rules for cubic splines with non-uniform knot sequences. *submitted 2015*.
- [2] A. Avdzhieva and G. Nikolov. Numerical computation of Gaussian quadrature formulae for spaces of cubic splines with equidistant knots. *BGSIAM*

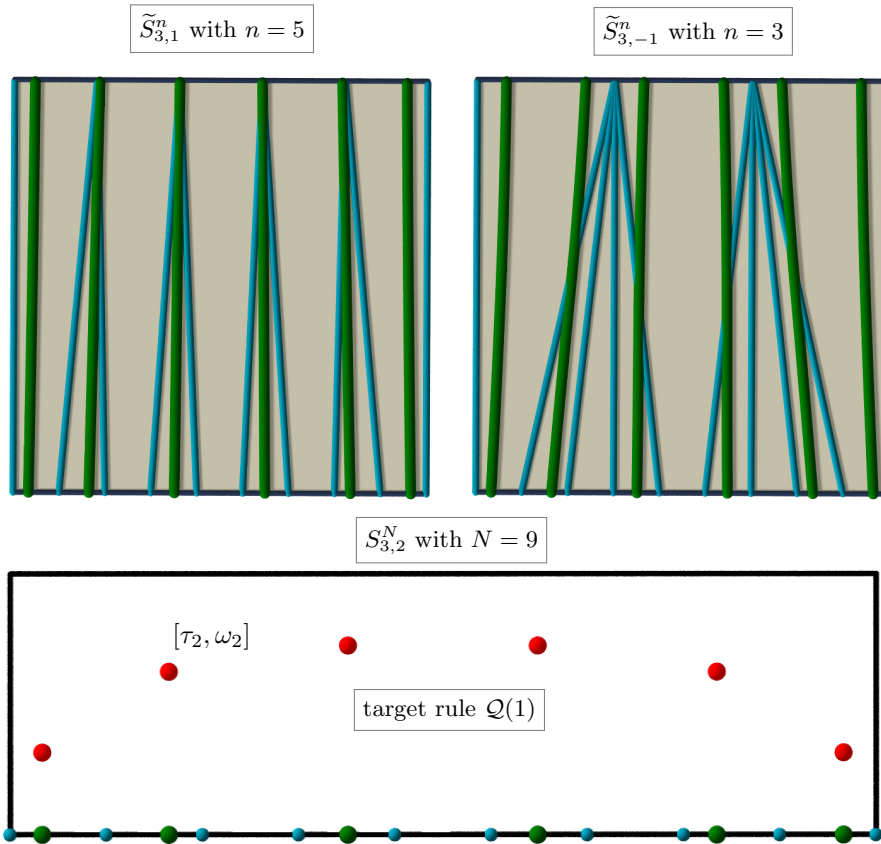


Figure 11: Evolution of the Gaussian rule from two different source rules is shown: the Gaussian rule for $\tilde{S}_{3,1}^n$ with $n = 5$ uniform elements (left) and the classical Gaussian rule for polynomials with $n = 3$ elements (right). The target space is $S_{3,2}^N$, the space of C^2 cubics with $N = 9$ uniform elements. The same target rule (bottom) was derived by Algorithm 1 with the accuracy of $\|\mathbf{r}\| < 10^{-16}$, see (25).

'12 *Proceedings of the seventh annual meeting of Bulgarian section of SIAM*, pages 28 – 38, 2012.

- [3] M. Bartoň, R. Ait-Haddou, and V.M. Calo. Gaussian quadrature rules for quintic splines. *submitted 2015*.
- [4] E. Cohen, R. F. Riesenfeld, and G. Elber. *Geometric Modeling with Splines: An Introduction*. A. K. Peters, 2001.
- [5] C. de Boor. On calculating with B-splines. *Journal of Approximation Theory*, 6(1):50–62, 1972.
- [6] G. Farin, J. Hoschek, and M. S. Kim. *Handbook of Computer Aided Geometric Design*. Elsevier, Amsterdam, 2002.
- [7] W. Gautschi. *Numerical Analysis*. Springer, 1997.

- [8] T.J.R. Hughes, A. Reali, and G. Sangalli. Efficient quadrature for NURBS-based isogeometric analysis. *Computer Methods in Applied Mechanics and Engineering*, 199(58):301–313, 2010.
- [9] C.A. Micchelli and A. Pinkus. Moment theory for weak Chebyshev systems with applications to monosplines, quadrature formulae and best one-sided l^1 approximation by spline functions with fixed knots. *SIAM J. Math. Anal.*, 8:206–230, 1977.
- [10] Ch. Micchelli. The fundamental theorem of algebra for monosplines with multiplicities. *Lineare Operatoren und Approximation*, pages 419–430, 1972.
- [11] G. Nikolov. On certain definite quadrature formulae. *Journal of Computational and Applied Mathematics*, 75(2):329–343, 1996.
- [12] I. J. Schoenberg. Spline functions, convex curves and mechanical quadrature. *Bulletin of the American Mathematical Society*, 64(6):352–357, 1958.
- [13] G. Szegő. Inequalities for the zeros of Legendre polynomials and related functions. *Transactions of the American Mathematical Society*, 39(1):1 – 17, 1936.
- [14] C.W. Wampler and A.J. Sommese. *The Numerical Solution of Systems of Polynomials Arising in Engineering and Science*. World Scientific, 2005.
- [15] C.W. Wampler, J. Verschelde, and A.J. Sommese. Numerical continuation methods for solving polynomial systems arising in kinematics. *ASME J. Mech. Design*, 112(1):59 – 68, 1990.