SIAM J. SCI. STAT. COMPUT.
Vol. 13, No. 4, pp. 1025–1038, July 1992

011

INTEGRATING PRODUCTS OF B-SPLINES*

A. H. VERMEULEN†, R. H. BARTELS†, AND G. R. HEPPLER†

Abstract. This paper outlines several ways to evaluate the integral of the product of two B-spline functions, followed by a detailed description of an algorithm that is based on integration by parts. The algorithm reduces the integral to a sum of evaluations of a higher-order spline. This reduction involves differentiating one spline by differencing its coefficients, and integrating the other by summing its coefficients.

Key words. B-splines, integration, inner product, finite element method

AMS(MOS) subject classifications. 65D07, 65D30, 65N30

1. Introduction.

In this paper we consider the evaluation of integrals of the forms:

$$\int_a^b \left(\sum_i E_i B_{i,k,x}(t)\right)\left(\sum_j F_j B_{j,l,y}(t)\right) dt, \tag{1}$$

$$\int_a^b f(t)\left(\sum_i E_i B_{i,k,x}(t)\right) dt, \tag{2}$$

where $B_{i,k,x}$ is the ith B-spline of order k defined over the knots $x_i, x_{i+1}, \cdots, x_{i+k}$. We will consider B-splines normalized so that their integral is one. The splines may be of different orders and defined on different knot sequences x and y. Often the limits of integration will be the entire real line, $-\infty$ to $+\infty$. Note that (1) is a special case of (2) where $f(t)$ is a spline.

Integrals of these forms arise in applications such as the finite element method [1] and least squares function fitting [2], [3], [4] when B-splines are used as basis functions. In some problems, the function $f(t)$ in (2) may be nonpolynomial; for example, it may be a sinusoid [5]. It will be seen that the method we propose can be used to integrate such functions, provided that a sufficient number of antiderivatives of $f(t)$ are known.

There are five different methods for calculating (1) that will be considered:

1. Use Gauss quadrature on each interval.
2. Convert the integral to a linear combination of integrals of products of B-splines and provide a recurrence for integrating the product of a pair of B-splines.
3. Convert the sums of B-splines to piecewise Bézier format and integrate segment by segment using the properties of the Bernstein polynomials.
4. Express the product of a pair of B-splines as a linear combination of B-splines. Use this to reformulate the integrand as a linear combination of B-splines, and integrate term by term.
5. Integrate by parts.

Of these five, only methods 1 and 5 are suitable for calculating (2). The first four methods will be touched on and the last will be discussed at length.

* Received by the editors May 16, 1991; accepted for publication (in revised form) August 14, 1991.

† Department of Systems Design Engineering, University of Waterloo, Waterloo, Ontario N2L3G1, Canada (rhbartel@uwaterloo.ca, ahvermen@uwaterloo.ca, and heppler@dial.uwaterloo.ca).

2. Gauss quadrature. The integral (1) can be broken into a sum of integrals, where each integrand is a polynomial:

$$\int_a^b \left(\sum_i E_i B_{i,k,x}(t)\right)\left(\sum_j F_j B_{j,l,y}(t)\right) dt$$
$$= \sum_r \int_{t_r}^{t_{r+1}} \left(\sum_i E_i B_{i,k,x}(t)\right)\left(\sum_j F_j B_{j,l,y}(t)\right) dt,$$

where $t_1, \cdots, t_n$ is the union of the breakpoints of the two splines. Each integrand is polynomial, therefore the integrals can be computed exactly using Gauss quadrature:

$$\int_{t_r}^{t_{r+1}} e(t)f(t)dt = \sum_s H_s e(\xi_s)f(\xi_s). \tag{3}$$

The numbers ξ_s and H_s are the Gauss points and weights. Gauss quadrature can be used to approximate integrals of form (2), for general $f(t)$.

A different application of Gauss quadrature is to evaluate integrals of the form (2) by moving the summation out of the integral and integrating each term using Gauss quadrature, treating the B-spline as a weight function. If $f(t)$ is a polynomial,

$$\int_{-\infty}^{\infty} B_{i,k,x}(t)f(t)dt = \sum_s H_s f(\xi_s). \tag{4}$$

The numbers ξ_s and H_s are Gauss points and weights for the particular weighting function $B_{i,k,x}(t)$. Values for uniformly spaced B-splines are given in [6]. The Gauss points and weights must be recalculated for each different B-spline; if the basis is uniform this is not difficult, but if the basis is arbitrary then it may present a problem. This method is only capable of calculating integrals of form (1) approximately.

3. Integrating products of B-splines. The summations in (1) can be moved outside the integral to yield a linear combination of terms of the form:

$$\int_a^b B_{i,k,x}(t)B_{j,l,y}(t)dt. \tag{5}$$

In the case where integration is over the entire real line this quantity can be calculated using a recurrence [7]. If the limits are not $-\infty$ and $+\infty$ the procedure can still be used as follows. Insert enough knots [8], [9], [10] at a and b into the splines in the integrand so that no B-spline's support crosses the integration limits. Now consider only the B-splines in the integration region and integrate these over the entire real line.

The recurrence described in [7] to evaluate (5) proceeds as follows. Define the number $T_{i,j}^{k,l}$ as

$$T_{i,j}^{k,l} := (-1)^k [x_i, x_{i+1}, \cdots, x_{i+k} : t][y_j, y_{j+1}, \cdots, y_{j+l} : s](s-t)_+^{k+l+1}, \tag{6}$$

where $[x_i, x_{i+1}, \cdots, x_{i+k} : t]$ is the divided difference operator with respect to the sequence $x_i, \cdots, x_{i+k}$ acting on the variable t, and similarly for $[y_j, y_{j+1}, \cdots, y_{j+l} : s]$. It can be shown that

$$\int_{-\infty}^{\infty} B_{i,k,x}(t)B_{j,l,y}(t)dt = \frac{k!l!}{(k+l-1)!} T_{i,j}^{k,l}. \tag{7}$$

The quantities $T_{i,j}^{k,l}$ can be calculated using the definition of the divided difference and equation (6); however, this can lead to loss of significance if the knot spacing is uneven. A stable method of doing the calculation is to use the following recurrence. The recurrence begins by noting, from (6) and the divided difference definition of a B-spline, that for terms where one of $y_j = y_{j+l}$ or $x_i = x_{i+k}$ holds, $T_{i,j}^{k,l}$ is a scalar multiple of the value of a B-spline:

$$(8) \qquad T_{i,j}^{k,l} = \begin{cases} \dfrac{(k+l-1)!}{k!l!} B_{i,k,x}(y_j), & y_j = y_{j+l}, \quad l \geq 0, \\ \dfrac{(k+l-1)!}{k!l!} B_{j,l,y}(x_i), & x_i = x_{i+k}, \quad k \geq 0. \end{cases}$$

Terms with higher numbers in the top indices can be calculated from terms of lower degree. The recurrence is

(9)

$$T_{i,j}^{k,l} = \begin{cases} \dfrac{(x_{i+k} - y_j)T_{i,j}^{k,l-1} + (y_{j+l} - x_{i+k})T_{i,j+1}^{k,l-1}}{y_{j+l} - y_j} + T_{i,j}^{k-1,l}, & x_{i+k} \leq y_{j+l}, \; y_j \leq y_{j+l} \\ \dfrac{(x_i - y_j)T_{i,j}^{k,l-1} + (y_{j+l} - x_i)T_{i,j+1}^{k,l-1}}{y_{j+l} - y_j} + T_{i+1,j}^{k-1,l}, & y_j \leq x_i, \qquad y_j \leq y_{j+l} \\ \dfrac{(y_j - x_i)T_{i,j}^{k-1,l} + (x_{i+k} - y_j)T_{i+1,j}^{k-1,l}}{x_{i+k} - x_i} + T_{i,j+1}^{k,l-1}, & x_i \leq y_j, \qquad x_i \leq x_{i+k} \\ \dfrac{(y_{j+l} - x_i)T_{i,j}^{k-1,l} + (x_{i+k} - y_{j+l})T_{i+1,j}^{k-1,l}}{x_{i+k} - x_i} + T_{i,j}^{k,l-1}, & y_{j+l} \leq x_{i+k}, \; x_i \leq x_{i+k}. \end{cases}$$

Thus the steps in the evaluation of (1) by this method are:

1. If necessary, insert enough knots into the two splines at the integration limits a and b, so that the support of no basis function extends past a or b.
2. Evaluate each of the splines $B_{i,k,x}$ at the points y_j and evaluate each of the splines $B_{j,l,y}$ at the points x_i.
3. Calculate the values $T_{i,j}^{k,l}$ recursively. The values obtained in step 2 provide an end to the recursion.
4. Evaluate the integral (1) by moving the summations and coefficients outside of the integral, replacing the integrals with scaled versions of $T_{i,j}^{k,l}$ according to (7), and summing over all pairs of B-splines.

4. Converting to Bézier form and integrating. The B-splines in the integrand of (1) can be converted to piecewise Bézier format and the result can be integrated segment by segment. On each segment, the B-spline combinations can be represented in Bézier form:

$$(10) \qquad \sum_i E_i B_{i,k,x} = \sum_i G_i P_{i,k},$$

$$(11) \qquad \sum_j F_j B_{j,l,y} = \sum_j H_j P_{j,l} \,.$$

$P_{i,k}$ is the ith Bernstein polynomial of degree $k-1$ on the segment. The Bézier coefficients G_i and H_j can be calculated via knot insertion [8], [9], blossoming [11], or the tetrahedral algorithm of Sablonnière [12]. On each segment, the integral (1) can be expressed as a linear combination of inner products of Bernstein polynomials:

$$\int_a^b \left(\sum_i E_i B_{i,k,x}(t)\right)\left(\sum_j F_j B_{j,l,y}(t)\right) dt = \sum_i \sum_j G_i H_j \int_a^b P_{i,k}(t) P_{j,l}(t) dt. \tag{12}$$

The integrals can be evaluated using the formulae for products and integrals of Bernstein polynomials given by Farin [13]. For a segment of length L that lies in the interval (a, b), the integral evaluates to

$$\int_a^b P_{i+k}(t) P_{j,l}(t) dt = \left[\frac{(k+l-2)!i!j!(k-i-1)!(l-j-1)!}{(i+j)!(k+l-i-j-2)!(k-1)!(l-1)!(k+l)}\right] L. \tag{13}$$

5. Explicit multiplication of the integrand. The product of B-splines can be expressed as a linear combination of B-splines [14]:

$$\left(\sum_i E_i B_{i,k,x}(t)\right)\left(\sum_j F_j B_{j,l,y}(t)\right) = \sum_h G_h B_{h,p,z}(t). \tag{14}$$

The order of the product spline is $p = k + l - 1$; the knot vector z must contain sufficient knots to represent the product spline. Since the order of the product is higher than the order of the factors, yet the continuity is the same, the multiplicity of knots in the product spline is generally much higher than in the original splines. This means that there will, in general, be many more splines in the product than in either of the factors.

The procedure for constructing the knot vector z with the minimum possible number of knots will now be described. Begin by setting a single knot z_i at each point for which $z_i = x_\alpha$, a knot in x, or $z_i = y_\beta$, a knot in y. Assign multiplicity m_i to z_i as follows:

$$m_i = \begin{cases} \max(k + m_\beta - 1, l + m_\alpha - 1), & m_\alpha > 0 \quad \text{and} \quad m_\beta > 0, \\ k + m_\beta - 1, & m_\alpha = 0 \quad \text{and} \quad m_\beta > 0, \\ l + m_\alpha - 1, & m_\alpha > 0 \quad \text{and} \quad m_\beta = 0, \end{cases} \tag{15}$$

where m_α is the multiplicity of x_α and m_β is the multiplicity of y_β. Note that, if there is no knot in x at z_i or no knot in y at z_i, then either m_α or m_β will be zero.

The coefficients of the product spline are related to the coefficients of the factors via the following linear relationship [14]:

$$G_h = \sum_{i,j} \Gamma_{i,j,k,l}(h) E_i F_j. \tag{16}$$

The coefficients Γ can be calculated using the following recurrence:

$$\begin{aligned} \Gamma_{i,j,k,l}(h) = \frac{z_{h+p} - z_h}{p(z_{h+p-1} - z_h)} \Bigg(& \frac{k}{x_{i+k} - x_i} [(z_{h+p-1} - x_i)\Gamma_{i,j,k-1,l}(h) \\ & + (x_{i+k} - z_{h+p-1})\Gamma_{i+1,j,k-1,l}] \\ & + \frac{l}{y_{j+l} - y_j} [(z_{h+p-1} - y_j)\Gamma_{i,j,k,l-1}(h) \\ & + (y_{j+l} - z_{h+p-1})\Gamma_{i,j+1,k,l-1}] \Bigg). \end{aligned}$$

The recursion is initiated by specifying the values of Γ for $k = l = 1$:

(17)
$$\Gamma_{i,j,1,1}(h) = \begin{cases} \dfrac{(x_{i+1} - x_i)(y_{j+1} - y_j)}{z_{h+1} - z_h}, & x_i \leq z_h < x_{i+1} \quad \text{and} \quad y_j \leq z_h < y_{j+1}, \\ 0, & \text{otherwise.} \end{cases}$$

We can rewrite the integral (1) as:

$$\sum_{i,j,h} \Gamma_{i,j,k,l}(h) E_i F_j \int_a^b B_{h,p,z}(t)dt. \tag{18}$$

The integral of a B-spline over its entire support is one; we use this for cases where the entire B-spline support is within the limits of integration. This can be arranged by inserting knots at a and b as described previously. If the entire B-spline is not within the region of integration, the recurrence given to calculate definite integrals of B-splines described in [7] can be used, or the integrand can be converted to Bézier form and integrated as described previously.

6. Integration by parts. The final method of evaluating (1) is integration by parts.

$$\int_a^b e(t)f(t)dt = f^{(-1)}(b)e(b) - f^{(-1)}(a)e(a) - \int_a^b e^{(1)}(t)f^{(-1)}(t)dt, \tag{19}$$

where $f^{(-1)}(t)$ is the antiderivative of $f(t)$ and $e^{(1)}(t)$ is the derivative of $e(t)$. Integration by parts can be applied repeatedly to the integrand. After n applications:

$$\int_a^b e(t)f(t)dt = (-1)^n \int_a^b e^{(n)} f^{(-n)}(t)dt + \text{constant terms.} \tag{20}$$

If $e(t)$ and $f(t)$ are splines, then the degree of $e(t)$ will be lowered while the degree of $f(t)$ is raised. If we apply this enough times, $e(t)$ will be reduced to a simple enough form that its product with $f(t)$ can be integrated directly.

To apply this principle we need to recall some results relating the integration and differentiation of B-splines to Dirac delta functions.

6.1. Differentiating B-splines. The following two-term derivative formula for B-splines is well known:

$$\frac{d}{dt} B_{i,k}(t) = \frac{k}{u_{i+k} - u_i} \left[B_{i,k-1}(t) - B_{i+1,k-1}(t) \right]. \tag{21}$$

We will only concern ourselves with the case where $u_{i+k} > u_i$; thus the right side always has a nonzero denominator. When $u_{i+k-1} > u_i$ and $u_{i+k} > u_{i+1}$ the B-splines on the right side are defined in the normal way. Under the assumption that $u_{i+k} > u_i$, the only special cases that can arise are: (a) $u_{i+k-1} = u_i$ but $u_{i+k} > u_{i+1}$, for which $B_{i,k-1}(t)$ has zero support, and (b) $u_{i+k-1} > u_i$ but $u_{i+k} = u_{i+1}$, for which $B_{i+1,k-1}(t)$ has zero support.

Consider the case where

$$u_i = u_{i+1} = \cdots = u_{i+k-1} < u_{i+k}. \tag{22}$$

FIG. 1. *A cubic* B-*spline with a discontinuity.*

This situation is depicted for a cubic B-spline in Fig. 1. The spline $B_{i,k}$ is discontinuous at the point u_i. Specifically, with the convention that segment intervals are closed on the left,

$$B_{i,k}(u)|_{u<u_i} = 0, \qquad B_{i,k}(u)|_{u=u_i} = \frac{k}{u_{i+k} - u_i}. \tag{23}$$

Since the function is discontinuous at u_i, the derivative does not exist in the normal sense. One option is to consider only right-sided derivatives [15], [16]. With this approach splines with zero support are taken as zero. As pointed out in [16, p. 88], the problem with this option is that the Fundamental Theorem of Calculus does not hold. To see this, note that if the first Fundamental Theorem held, we would expect that

$$B_{i,k}(u) = \int_{-\infty}^{u} \frac{d}{dt} B_{i,k}(t)dt. \tag{24}$$

Substituting from (21) into the above integral, and using the definition that zero-support B-splines are to be interpreted as zero, yields

$$B_{i,k}(u) = \int_{-\infty}^{u} \frac{k}{u_{i+k} - u_i} \left[0 - B_{i+1,k-1}(t)\right] dt. \tag{25}$$

Yet at the point $u = u_{i+k}$ the left side of the equation evaluates to

$$\mathrm{LS} = B_{i,k}(u_{i+k}) = 0, \tag{26}$$

while the right side yields (since the integral of a B-spline over its support is one)

$$\mathrm{RS} = \int_{-\infty}^{u_{i+k}} \frac{k}{u_{i+k} - u_i} \left[0 - B_{i+1,k-1}(t)\right] dt = -\frac{k}{u_{i+k} - u_i} \neq 0. \tag{27}$$

This means that, if B-splines with zero support are taken as zero, we cannot use integration by parts to solve (1), because integration by parts is based on the Fundamental Theorem. Therefore, we will look at one of the alternative ways that B-splines with zero support have been defined.

We begin by recalling a result attributed to Curry and Schoenberg [7] that a kth order B-spline on the knots $u_i, \cdots, u_{i+k}$ can be defined as the function $B_{i,k}$, which satisfies

$$\int_{-\infty}^{\infty} B_{i,k}(t) g^{(k)}(t) dt = k! \left[u_i, u_{i+1}, \cdots, u_{i+k} : t\right] g(t) \tag{28}$$

for any function $g(t)$ with k continuous derivatives. Consider $B_{i,k}$ where $u_i = \cdots = u_{i+k} = \hat{u}$. In this case the right side of (28) becomes the kth derivative of $g(t)$ at $\hat{u}$. To be consistent, the left side must yield

$$\int_{-\infty}^{\infty} B_{i,k}(t) g^{(k)}(t) dt = g^{(k)}(\hat{u}). \tag{29}$$

Distribution theory [17], [18] provides an entity that behaves precisely as $B_{i,k}(t)$ must in this circumstance: the Dirac delta function $\delta(t-\hat{u})$. This function has the property that

$$\int_{a}^{b} \delta(t-\hat{u}) f(t) dt = \begin{cases} f(\hat{u}), & a < \hat{u} < b, \\ 0, & \hat{u} < a \ \text{ or } \ \hat{u} > b, \\ \text{undefined}, & \hat{u} = a \ \text{ or } \ \hat{u} = b \end{cases} \tag{30}$$

for any function $f(t)$ integrable on (a, b). Accordingly,

$$B_{i,k}(t) = \delta(t - \hat{u}), \quad u_i = u_{i+1} = \cdots = u_{i+k} = \hat{u}, \tag{31}$$

which is consistent with the Fundamental Theorem. Consider again the spline $B_{i,k}$ with knots as in (22). Using this definition, the two-term differentiation formula now yields

$$\frac{d}{dt} B_{i,k}(t) = \frac{k}{u_{i+k} - u_i} \left[\delta(t - \hat{u}) - B_{i+1,k-1}(t)\right]. \tag{32}$$

The revised version of (25) is

$$B_{i,k}(u) = \int_{-\infty}^{u} \frac{k}{u_{i+k} - u_i} \left[\delta(t - u_i) - B_{i+1,k-1}(t)\right]. \tag{33}$$

For $u < u_i$ or $u > u_{i+k}$ both left and right sides are zero. For $u_i < u < u_{i+k}$ we first note that the left side of (33) has polynomial form

$$\text{LS} = B_{i,k}(u) = \frac{k}{(u_{i+k} - u_i)} (u_{i+k} - u)^{k-1}. \tag{34}$$

The right side yields:

$$\text{RS} = \frac{k}{u_{i+k} - u_i} \left[\int_{\infty}^{u} \delta(t - u_i) dt - \int_{u_i}^{u} \frac{k-1}{(u_{i+k} - u_i)^{k-1}} (u_{i+k} - t)^{k-2} dt\right]. \tag{35}$$

Since $u > u_i$ the first integral has the value one. Expanding the second integral and simplifying yields

$$\text{RS} = \frac{k}{u_{i+k} - u_i} (u_{i+k} - u)^{k-1}. \tag{36}$$

Hence, the left side of (33) is equal to the right side.

The definition (31) is reasonable from another point of view as well. B-splines have been normalized to integrate to one:

$$\int_{-\infty}^{\infty} B_{i,k}(t)dt = 1. \tag{37}$$

This, too, is consistent with (30) for $f(t) = 1$.

6.1.1. Derivatives of B-spline combinations. The derivative of a linear combination of B-splines is a linear combination of B-splines of next lower order:

$$\frac{d}{dt}\sum_{i=0}^{m} V_i B_{i,k,u}(t) = \sum_{i=0}^{m+1} V_i^{(1)} B_{i,k-1,u}(t), \tag{38}$$

where the coefficients can be obtained by substituting the two-term differentiation formula into the left-hand side and shifting the summation:

$$V_i^{(1)} = \frac{k}{u_{i+k} - u_i} V_i - \frac{k}{u_{i+k-1} - u_{i-1}} V_{i-1}. \tag{39}$$

V_{-1} and V_{m+1} are defined to be zero.

6.2. Integrating B-splines. To obtain a formula for the indefinite integral of a B-spline combination, we integrate (38). This leads to this description of the indefinite integral of a B-spline combination

$$\int \sum_{i=0}^{m} V_i B_{i,k}(t) = \sum_{i=0}^{m} V_i^{(-1)} B_{i,k+1}(t), \qquad -\infty \le t < u_{m+1}, \tag{40}$$

where the coefficients are obtained by inverting (39):

$$V_i^{(-1)} = \frac{u_{i+k+1} - u_i}{k+1} \left[\frac{k+1}{u_{i+k} - u_{i-1}} V_{i-1}^{(-1)} + V_i \right]. \tag{41}$$

$V_{-1}^{(-1)}$ is defined to be zero.

The integral spline requires the existence of a new knot, u_{m+k+1}. The value of this new knot is arbitrary subject to $u_{m+k+1} \ge u_{m+k}$. Adopting the convention that knots with indices past the end of the given knot vector are equal to the last given knot can simplify implementation.

Also note that the integral spline in (40) is only valid on the interval $[-\infty, u_{m+1})$. This condition is necessary because the integral of a B-spline combination will, in general, have unbounded support. Such a function is not representable as a linear combination of a finite number of B-splines, but the portion to the left of u_{m+1} is representable in this way; hence the condition. An alternative method is to define B-spline–like basis functions that have unbounded support on one side. Suitable basis functions for this alternative are described in Barry and Goldman [19] and in de Boor, Lyche, and Schumaker [7].

6.3. The integration by parts algorithm. In this section the integration by parts algorithm will be described.

Begin by defining two splines

(42) $$e(t) = \sum_{i=0}^{m} E_i B_{i,k,x}(t),$$

(43) $$f(t) = \sum_{j=0}^{n} F_j B_{j,l,y}(t).$$

The integral we wish to compute is

(44) $$\int_a^b e(t)f(t)dt.$$

Informally, the approach will be to use integration by parts to reduce the order of $e(t)$ while increasing the order of $f(t)$. This will reduce the support length of e's B-splines. When the support length of one of e's B-splines reaches zero, the B-spline becomes a Dirac delta function and thus the part of the integral on this basis function reduces to an evaluation of f. Eventually, all of e's B-splines will have zero support and the integral will be reduced to a sum of evaluations.

A detailed description of the algorithm is presented below. Step 0 serves to bring the integrand into a canonical form; steps 1–4 constitute the substance of the algorithm.

0. If the lower limit of integration, a, lies exactly on a knot, shift it an infinitesimal amount to the right. Similarly, if b lies exactly on a knot, shift it an infinitesimal amount to the left. This will not affect the value of the integral since the integral of a product of splines of order 1 or more varies continuously as the limits of integration are changed. The shifting of the limits is necessary to avoid the undefined condition in definition (30). In practical terms, however, such infinitesimal shifts correspond to consistently using one index ordering in making comparisons.
   Due to the formula chosen for spline integration, it is necessary that $b \leq y_{n+1}$. If this is not the case, let β be the index such that

(45) $$y_\beta < b \leq y_{\beta+1}.$$

   Now add $\beta - n + 1$ knots into y, such that each of the new knots is larger than or equal to y_{n+l}. Correspondingly, increase the value of n by $\beta - n + 1$. This increases the number of basis splines used to represent $f(t)$. Since the new knots are outside the nonzero part of the spline, set the coefficients for each of the new basis splines to zero. Note that the function $f(t)$ is unchanged, but the new representation satisfies the condition that $b \leq y_{n+1}$.
   If $b = \infty$, then first replace b by $\min(y_{n+l}, x_{m+k})$. Since either $e(t)$ or $f(t)$ is zero past this point, this will not affect the value of the integral. Now adjust the representation of $f(t)$ as described above, if necessary.
   It is also required that y contain sufficient knots so that $B_{m,k+l,y}$ is defined. As previously mentioned, an easy way to implement this is to adopt the convention that added knots past the end of the given knot vector are equal to the last given knot.
1. Calculate the coefficients of $f^{(-1)}(t)$ using (41). We can now discard the coefficients of f itself and can calculate the two constant terms in the integration

by parts formula

$$\int_a^b e(t)f(t)dt = e(b)f^{(-1)}(b) - e(a)f^{(-1)}(a) - \int_a^b e^{(1)}(t)f^{(-1)}(t)dt, \tag{46}$$

leaving the integral term to be dealt with.

2. Calculate the coefficients of $e^{(1)}(t)$ using (39), and discard the coefficients of e.
3. Separate the basis splines of $e^{(1)}(t)$ into two categories: those with finite length support and those with zero length support. Let A and B be sets of integers such that:

$$i \in A \quad \text{if } 0 \le i \le m \quad \text{and} \quad x_i = x_{i+k-1}, \tag{47}$$

$$i \in B \quad \text{if } 0 \le i \le m \quad \text{and} \quad x_i < x_{i+k-1}. \tag{48}$$

The B-splines whose indices are in A are those whose support length is zero; they correspond to Dirac delta functions. We can write the spline $e^{(1)}(t)$ as

$$e^{(1)}(t) = \sum_{i\in A} E_i^{(1)}\delta(t - x_i) + \sum_{i\in B} E_i^{(1)}B_{i,k-1,x}(t). \tag{49}$$

4. Substitute (49) into the integral term in (46), noting that the integrals with delta terms reduce to function evaluations

$$\int_a^b e^{(1)}f^{(-1)}(t)dt = \sum_{i\in A} E_i^{(1)}f^{(-1)}(x_i) + \int_a^b \sum_{i\in B} E_i^{(1)}B_{i,k-1,x}(t)f^{(-1)}(t)dt. \tag{50}$$

The first set of terms require evaluations of the spline $f^{(-1)}(t)$. These terms can be calculated and added to the sum of terms calculated so far. If the set B is empty we are finished. If B is not empty then the integral term in (50) is the integral of a product of two splines. The first spline is the function $e^{(1)}(t)$ with the basis splines of zero support removed. The second is the spline $f^{(-1)}(t)$. Apply steps 1–4 recursively to this term, until all the splines in $e(t)$ are accounted for and B is reduced to the empty set.

Note that the same approach can be used to evaluate integrals of the form (2), provided that the first k antiderivatives of the function $f(t)$ can be calculated.

6.4. Computational cost. We will consider for simplicity problems of the form (1) where both splines are of order k. There are three computational components in the integration by parts algorithm:

1. Repeated differentiation of the spline $e(t)$. This step must be done k times. Each invocation takes one subtraction and one division per interval so the time spent on differentiation is on the order of k operations per segment.

2. Repeated integration of the spline $f(t)$. This step must be done k times. Each invocation takes one addition and one multiplication per interval so the time spent on antidifferentiation is also on the order of k operations per segment.

3. Evaluation of antiderivatives of the spline $f(t)$. One evaluation of an antiderivative of $f(t)$ must be done per segment; each evaluation takes on the order of k^2 operations.

The cost of the spline evaluations dominates the total computational cost. Thus the total cost is $O(k^2)$ operations per segment.

TABLE 1
Stability for order-4 B-splines.

r	Exact	Gauss Quadrature	Divided Differencing	Parts Method
0	4.19444444444444	4.19444444444444	4.19444444444444	4.19444444444444
1	4.06649773598049	4.06649773598049	4.06649773598050	4.06649773598050
2	4.04010964362323	4.04010964362322	4.04010964362323	4.04010964362322
3	4.03734554112486	4.03734554112486	4.03734554112671	4.03734554112486
4	4.03706789985594	4.03706789985594	4.03706789987676	4.03706789985593
5	4.03704012344300	4.03704012344300	4.03704012326651	4.03704012344251
6	4.03703734567887	4.03703734567887	4.03703734571633	4.03703734568062
7	4.03703706790123	4.03703706790123	4.03703707070117	4.03703706789959
8	4.03703704012346	4.03703704012346	4.03703703882036	4.03703703977985
9	4.03703703734568	4.03703703734568	4.03703618402446	4.03703703693127
10	4.03703703706790	4.03703703706790	4.03705093396563	4.03703705312282
11	4.03703703704012	4.03703703704012	4.03707198835796	4.03703672714926
12	4.03703703703735	4.03703703703735	4.03832729958782	4.03703333713402
13	4.03703703703707	4.03703703703707	4.02124815634075	4.03709174968579
14	4.03703703703704	4.03703703703704	4.17366372053858	4.03745891429760
15	4.03703703703704	4.03703703703704	3.95746527777776	4.03398753978588

TABLE 2
Stability for order-6 B-splines.

r	Exact	Gauss Quadrature	Divided Differencing	Parts Method
0	30.3322685185185	30.3322685185185	30.3322685185185	30.3322685185185
1	28.8504734229846	28.8504734229846	28.8504734229846	28.8504734229845
2	28.6816125192285	28.6816125192285	28.6816125192211	28.6816125192286
3	28.6645841566786	28.6645841566786	28.6645841567912	28.6645841566787
4	28.6628799571565	28.6628799571565	28.6628799578620	28.6628799571566
5	28.6627095236305	28.6627095236305	28.6627095075280	28.6627095236304
6	28.6626924801422	28.6626924801422	28.6626923729591	28.6626924801421
7	28.6626907757920	28.6626907757920	28.6626892195090	28.6626907757921
8	28.6626906053570	28.6626906053570	28.6626979434700	28.6626906053572
9	28.6626905883135	28.6626905883135	28.6627689710028	28.6626905883125
10	28.6626905866091	28.6626905866091	28.6635029574743	28.6626905865784
11	28.6626905864387	28.6626905864387	28.6575636537495	28.6626905852876
12	28.6626905864216	28.6626905864216	28.7488702056990	28.6626905950749
13	28.6626905864199	28.6626905864199	28.6558351597237	28.6626906802310
14	28.6626905864198	28.6626905864198	20.9915637713911	28.6626918634761
15	28.6626905864198	28.6626905864197	94.3609932303722	28.6626955924918

6.5. Stability. Of the three steps in the algorithm, only the third, evaluation, is unconditionally stable for all knot sequences. The other two steps, differentiation and antidifferentiation, may lead to numerical problems. It is also possible that forming the weighted sum of the evaluations may lead to loss of significance. To address these issues, a numerical comparison was made between integration by parts, Gauss quadrature, and direct evaluation of the divided difference formula (6). A similar comparison between Gauss quadrature and direct evaluation of (6) was made in [7].

The sample problem chosen was to evaluate the integral of the square of the order-k B-spline defined on the knot sequence $[5, 6, 6 + 10^{-r}, 8, \cdots, 5 + k]$. In Tables 1–3, values of $T_{0,0}^{k,k}$ are shown for r ranging from 0 to 15 for several different orders. The exact values were obtained symbolically using Maple [20]; the algorithmic results were calculated using double precision arithmetic on a DEC 5400 running Ultrix. In each result, the first significant digit in error is indicated using an underline. Note that the integration by parts algorithm is accurate to machine precision in the test case for all orders when the ratio of largest to smallest segment length is less than 10000:1; in

TABLE 3
Stability for order-10 B-splines.

r	Exact	Gauss Quadrature	Divided Differencing	Parts Method
0	2833.16953523513	2833.16953523514	2833.16953523513	2833.16953523517
1	2752.86392636369	2752.86392636369	2752.86392636290	2752.86392636372
2	2744.44592708222	2744.44592708222	2744.44592709179	2744.44592708226
3	2743.60112105862	2743.60112105862	2743.60111969566	2743.60112105872
4	2743.51661119805	2743.51661119805	2743.51660988489	2743.51661119820
5	2743.50815992021	2743.50815992021	2743.50813576840	2743.50815992031
6	2743.50731478951	2743.50731478950	2743.50587880024	2743.50731478967
7	2743.50723027641	2743.50723027641	2743.49088954013	2743.50723027654
8	2743.50722182510	2743.50722182509	2743.34218803890	2743.50722182512
9	2743.50722097996	2743.50722097996	2743.66538821758	2743.50722098001
10	2743.50722089545	2743.50722089545	2745.21802630369	2743.50722089536
11	2743.50722088700	2743.50722088700	2561.13229702871	2743.50722088701
12	2743.50722088616	2743.50722088616	2860.62453767450	2743.50722088613
13	2743.50722088607	2743.50722088607	17477.0284871979	2743.50722088603
14	2743.50722088606	2743.50722088606	−159288.60966869	2743.50722088613
15	2743.50722088606	2743.50722088606	1969830.68552676	2743.50722088608

practical circumstances (e.g., finite elements) it is rare to see ratios this extreme. As r increases, the knots become less uniformly spaced and loss of significance becomes more pronounced in the integration by parts and divided difference algorithms. The accuracy of the divided difference scheme decreases as the order increases; it is interesting that the accuracy of the integration by parts algorithm *increases* as the order increases.

7. Comparison of the algorithms. In this section we present a brief comparison of the five methods considered for evaluation of the integral (1). We will consider for simplicity problems of the form (1) where both splines are of order k. Only rough estimates of the number of operations needed for each method are given.

The first method considered is the use of Gauss quadrature. To do the integration exactly requires evaluating each spline k times per interval. The cost of a B-spline evaluation is $O(k^2)$ operations, thus the total cost will be $O(k^3)$ operations per segment. The method exhibits no loss of significance for any order or knot sequence. Gauss quadrature extends to problems of the type given in (2).

The second method formulated the integrand as a linear combination of products of B-splines and used the recurrence given by de Boor, Lyche, and Schumaker [7] to calculate the integral of each B-spline pair. Each invocation of this recurrence requires $O(k^3)$ operations [7], and the recurrence must be carried out k times per segment, so the total cost is on the order of $O(k^4)$ operations per segment. The chief advantage of this approach is that it is very stable numerically. The disadvantages are that the recurrence is complicated, the approach does not extend to problems of the type given in (2), and this is the most expensive approach considered. It is worth noting that if a large number of integrations needs to be done using splines with the same bases but different coefficients, the inner products of the B-splines could be precalculated, thus reducing the computational expense.

The third method converted the splines to piecewise Bézier format and integrated segment by segment. To convert to Bézier format, we must compute the k Bézier coefficients on each segment. The calculation of these coefficients carries approximately the same price as the evaluation of a B-spline value, which is $k(k-1)/2$ linear combination operations. Thus the cost of converting both splines is about $k(k-1)$ linear combinations per segment. After the coefficients are obtained, we must calculate a weighted sum of all possible inner products of Bernstein polynomials on each interval. There are k^2 such inner products for two splines of order k. Thus the total cost of this method is of order $O(k^2)$. This algorithm cannot be extended to problems of the form (2).

The fourth method represented the integrand as a linear combination of B-splines. This involved the computation of numbers relating the coefficients of the factor splines to the coefficients of the product spline. It can be shown that these coefficients, Γ, are a generalization of the discrete B-splines. In fact, the computation of the Γ implicitly computes the discrete B-splines necessary to convert the splines to Bézier form. Thus we conclude that this method is at least as expensive as method 2. This algorithm also cannot be extended to problems of the form (2).

The final method is the integration by parts algorithm. Although it is not as stable as the other methods, good accuracy is obtained for reasonable knot vectors. The integration by parts method, in our implementation, is less expensive than the other methods; approximately three times as fast as Gauss quadrature for cubics, and about eight times as fast for degree 10 splines. In terms of order of operations, the method requires $O(k^2)$ operations, the same as the conversion to Bézier method and less than the other methods. In addition, this algorithm is extensible to problems of the form (2), provided that a sufficient number of antiderivatives of the function $f(t)$ can be calculated.

In conclusion, the integration by parts method is less expensive than the other methods considered, provides accurate results for reasonably uniform knot vectors, and generalizes to problems of the form (2).

Acknowledgment. The authors gratefully acknowledge the helpful comments of one of the referees.

REFERENCES

[1] O. C. Zienkiewicz, *The Finite Element Method in Engineering Science*, Fourth Edition, Prentice-Hall, Englewood Cliffs, NJ, 1987.

[2] S. D. Conte and C. de Boor, *Elementary Numerical Analysis*, Second Edition, McGraw Hill, New York, 1972.

[3] G. Dahlquist and Å. Björk, *Numerical Methods*, Prentice-Hall, Englewood Cliffs, NJ, 1974.

[4] P. Davis, *Interpolation and Approximation*, Blaidsell, New York, 1963.

[5] G. R. Heppler and J. S. Hansen, *A mindlin element for thick and deep shells*, Comput. Methods Appl. Mech. Engrg., 54 (1986), pp. 21–47.

[6] J. L. Phillips and R. J. Hanson, *Gauss quadrature rules with* B-*spline weight functions*, Math. Comp., 28 (1978), p. 666.

[7] C. de Boor, L. Lyche, and L. L. Schumaker, *On calculating with* B-*splines* ii: *Integration,* in Numerische Methoden der Approximationstheorie, L. Collatz, H. Werner, and G. Meinardus, eds., Birkhäuser, Basel, 1976, pp. 123–146.

[8] W. Boehm, *Inserting new knots into* B-*spline curves*, Computer Aided Design, 12 (1980), pp. 199–201.

[9] E. Cohen, T. Lyche, and R. Riesenfeld, *Discrete* B-*splines and subdivision techniques in computer-aided geometric design and computer graphics*, Computer Graphics and Image Processing, 14 (1980), pp. 87–111.

[10] R. Goldman, *Blossoming and knot insertion algorithms for* B-*spline curves*, Comput. Aided Geom. Design, 7 (1990), pp. 69–82.

[11] L. Ramshaw, *Blossoms are polar forms*, Comput. Aided Geom. Design, 6 (1989), pp. 323–358.

[12] P. Sablonnière, *Spline and Bézier polygons associated with a polynomial spline curve*, Computer Aided Design, 10 (1978), pp. 257–261.

[13] G. Farin, *Curves and Surfaces for Computer Aided Geometric Design*, Academic Press, New York, 1988.

[14] K. MORKEN, *Products of splines as linear combinations of* B-*splines*, manuscript.
[15] R. H. BARTELS, J. C. BEATTY, AND B. A. BARSKY, *An Introduction to Splines for Use in Computer Graphics and Geometric Modeling*, Morgan Kaufmann, San Mateo, CA, 1987.
[16] C. DE BOOR, *A Practical Guide to Splines*, Springer-Verlag, Berlin, New York, 1978.
[17] L. SCHWARTZ, *Mathematics for the Physical Sciences*, Addison-Wesley, New York, 1966.
[18] A. H. ZEMANIAN, *Distribution Theory and Transform Analysis*, McGraw Hill, New York, 1965.
[19] P. J. BARRY AND R. N. GOLDMAN, *Algorithms for progressive curves: Extending* B-*spline and blossoming techniques to the monomial, power and Newton dual bases*, in Knot Insertion and Deletion Algorithms for B-Spline Modeling, R. Goldman and T. Lyche, eds., Society for Industrial and Applied Mathematics, Philadelphia, PA, to appear.
[20] B. W. CHAR, K. O. GEDDES, G. H. GONNET, M. B. MONOGAN, AND S. M. WATT, *Maple Reference Manual*, Watcom Publications Limited, Waterloo, Ontario, Canada, 1988.