GALERKIN DIFFERENCES FOR HIGH-ORDER PARTIAL DIFFERENTIAL EQUATIONS[*]

J. JACANGELO[†], J. W. BANKS[†], AND T. HAGSTROM[‡]

Abstract. Galerkin differences (GDs) were introduced in [J. W. Banks and T. Hagstrom, *J. Comput. Phys.*, 313 (2016), pp. 310–327; J. W. Banks, T. Hagstrom, and J. Jacangelo, *J. Comput. Phys.*, 372 (2018), pp. 864–892] and employed C^0 basis functions in a Galerkin projection to approximate solutions to the wave equation. In those works, the basis was derived by considering standard piecewise continuous polynomial interpolation. The resulting GD approximations were found to have excellent properties in terms of both their accuracy and computational efficiency. In the present work we further extend GD by considering higher derivative operators, such as those commonly found in beam or plate models of solid mechanics. These higher-order PDEs necessitate higher continuity basis functions. To derive this smoother basis, we introduce the difference spline, which is a locally constructed C^1 polynomial interpolant using only discrete data at $p+1$ consecutive grid points. We show that the difference spline interpolant is a pth-order accurate approximation, and basis functions associated with each grid point are derived. The basis is then used in a standard weak-form finite element approximation of the PDEs, and classical finite element theory shows that the method is pth-order accurate in the L^2-norm. Numerical convergence studies on the Euler–Bernoulli beam and the Kirchhoff–Love plate are performed and verify the theory.

Key words. difference methods, Galerkin methods, initial-boundary value problems, biharmonic equations

AMS subject classifications. 65M60, 65M12, 74K10, 74K20

DOI. 10.1137/19M1259456

1. Introduction. Many problems originating in science and engineering rely on accurate solutions to wave equations. Examples include applications in acoustics, solid mechanics, and gravity. In general, wave problems involve propagation of information over long distances without loss of energy. Lossless propagation is reflected in the governing equations in that they lack any natural dissipation. That is to say that $\omega(\mathbf{k}) \in \mathbb{R}$ for solutions of the form $\exp(i(\mathbf{k} \cdot \mathbf{x} - \omega t))$.

A lack of dissipation can present challenges for numerical methods approximating solutions to the governing PDEs. In particular, small perturbations, e.g., from underintegration or other variational crimes, can cause ω to move into the right half-plane and thereby render the scheme unstable. In this sense, numerical methods for wave equations are fragile, and it is therefore useful to have a theoretical guarantee of numerical stability, usually through an energy estimate. Common energy stable methods are summation-by-parts [19, 18, 21] and finite elements [13, 20], which can roughly be considered as being built around discrete and continuous inner products, respectively. In the present work, the focus is on high-derivative PDE operators, such as those commonly found in models of beams and plates. Summation-by-parts difference methods

[†]Department of Mathematical Sciences, Rensselaer Polytechnic Institute, Troy, NY 12180 (jjack12@gmail.com, banksj3@rpi.edu).

[‡]Department of Mathematics, Southern Methodist University, Dallas, TX 75275-0156 (thagstrom@mail.smu.edu).

for third- and fourth- order derivatives are constructed by Mattsson in [15]. For finite element approximations, existing approaches to addressing high-order operators include conversion to a low-order system, use of special elements such as Hermite or Argyris elements, spline-based FEM, or methods with lower-order continuity bases such as interior penalty and local discontinuous Galerkin formulations [23, 6, 10, 17, 16, 24].

The Galerkin difference (GD) schemes discussed in [2, 3] focused on discretizing the first- and second-order formulations of wave equations, using a newly derived set of C^0 continuous basis functions based on piecewise polynomial interpolation. Standard finite element techniques using a continuous inner product then led to discretizations with excellent properties in terms of accuracy and computational efficiency. However, because the basis functions are only C^0 continuous, direct application to higher-order PDEs is not feasible. In the present work, we describe a modification to the derivation that yields C^1 GD basis functions. The construction uses local interpolants with additional continuity at element boundaries imposed using local difference approximations. This new interpolation approach will be called a D-Spline (for difference spline), and the corresponding basis functions are the D-Spline basis functions. The C^1 D-Spline basis can then be used in a Galerkin approach for solving biharmonic equations typical for modeling motion of beams and plates. The accuracy of the resulting discretizations follows from the accuracy of the underlying D-Spline interpolant. As shown in subsection 3.2, for basis functions of support width $p + 1$, the error in the interpolant is proportional to h^p, where h is the mesh size. The accuracy of the GD D-Spline discretizations for beams and plates is therefore found to be h^p. However, analysis of the dispersion error away from boundaries exposes a superconvergence phenomenon similar to what was found for the C^0 basis. As in [2, 3], the D-Spline basis functions span multiple elements so boundary closures are necessary. However, the general-purpose closures introduced in [2, 3] (ghost basis and extrapolation) were found to be impractical due to poor conditioning of the discrete operators. We therefore introduce a new approach to boundary closures, essentially a hybrid of extrapolation and compatibility closures, that yields discretization matrices with much more favorable spectral properties.

The remainder of this paper is organized as follows. In section 2 we review the governing equations for the Euler–Bernoulli beam and the Kirchhoff–Love plate. In section 3 we discuss the construction of the D-Spline interpolant and its corresponding basis functions. This section also provides a brief contrast with B-Splines as commonly used in isogeometric analysis. Additionally we develop a theoretical interpolation error bound and analyze the weak derivative constructed from the D-Spline basis functions. This is followed by a discussion of the finite element formulation, boundary closures, and convergence of the finite element method in section 4. Also, section 4 contains a description of the high-order accurate A-stable time-stepping scheme we use, which is a spectral deferred correction (SDC) scheme [7, 8], as well as a brief discussion of the computational cost of our approach. Theoretical results are then confirmed with numerical studies in section 5. Finally, we offer some concluding remarks and avenues for future work in section 6

2. Governing equations. This work considers classical linear models for free vibrations of thin objects subject to small deformations, for example, beams or plates. Letting $u = u(\mathbf{x}, t)$ be the displacement in the normal direction, the evolution of the deflection is governed by

$$(2.1) \qquad\qquad \rho u_{tt} = -\nabla^2 (D \nabla^2 u), \quad \mathbf{x} \in \Omega,$$

where subscripts indicate differentiation ($u_{tt} = \partial^2 u/\partial t^2$), ρ is the density, D is the flexural rigidity of the model, and Ω is the domain of interest (to be specified later as appropriate). The form of D is dependent on the model, and here we consider two simple models; the Euler–Bernoulli beam (one-dimensional (1D)) and the Kirchhoff–Love plate (2D).

Euler–Bernoulli beams. The first model considered is the 1D beam with $D = EI$, where E is the Young's modulus, I is the area moment of inertia, and ρ is the linear mass density. For simplicity consider only constant material parameters ρ and EI so the governing equation becomes

$$(2.2) \qquad \rho u_{tt} = -EI u_{xxxx}, \quad x \in \Omega.$$

Kirchhoff–Love plates. The second model considered is that for a uniform plate of thickness $2H$. It is assumed that in-plane deformations are negligible in the model. Here $D = 2H^3 E/(3(1 - \nu^2))$, where E is Young's modulus and ν is Poisson's ratio. Taking E, H, ν, ρ to be constants results in the simplified model

$$(2.3) \qquad \rho u_{tt} = -D(u_{xxxx} + 2u_{xxyy} + u_{yyyy}), \quad (x,y) \in \Omega.$$

3. D-Spline interpolation. The previous construction of GD approximations in [3, 2] used a C^0 piecewise polynomial interpolant $\hat{u}(x)$ from discrete data u_j. For PDEs with higher spatial derivatives, such as those of the beam (2.2) and plate (2.3), additional smoothness is needed to build the discretization of the weak-form operator. In the present work, a C^1 interpolant, $\tilde{u}(x)$, of nodal data u_j is constructed. To that end, assume that a computational grid is denoted $x_j = jh$, where $j = \ldots, -1, 0, 1, \ldots$, and h is a uniform grid spacing. Furthermore, assume that u_j are discrete approximations of a continuous function $u(x)$ at the grid points so that $u_j \approx u(x_j)$. Finally, let p be an odd integer ($p+1$ being the size of the interpolation stencil), and $q = (p+1)/2$.

The interpolant, $\tilde{u}(x)$, is now defined piecewise as

$$(3.1) \qquad \tilde{u}(x) = \tilde{u}_j(x) \quad \text{for} \ \ x_j < x \leq x_{j+1},$$

where $\tilde{u}_j(x)$ is a degree $p + 2$ polynomial which interpolates nodal data at the $p + 1$ consecutive grid points $x_{j-q+1}, \ldots x_{j+q}$. Fitting data at $p+1$ consecutive points would uniquely define a degree p polynomial and a global C^0 interpolant. Here, however, two additional constraints are needed to uniquely define the degree $p + 2$ polynomial $\tilde{u}_j(x)$ and ensure C^1 continuity. So that the interpolation remains local and uses only data at the aforementioned $p+1$ grid points, the derivatives are constrained to satisfy the $\mathcal{O}(h^{p-1})$ accurate centered difference approximation of the first derivative at the endpoints x_j and x_{j+1}, specifically

$$(3.2a) \qquad \frac{d}{dx}\tilde{u}_j(x_j) = \mathcal{D}^{(1,p)}u_j,$$

$$(3.2b) \qquad \frac{d}{dx}\tilde{u}_j(x_{j+1}) = \mathcal{D}^{(1,p)}u_{j+1},$$

where $\mathcal{D}^{(1,p)}(\cdot)$ is the $O(h^{p-1})$ centered finite difference operator using data at p consecutive grid points. For example,

$$(3.3) \qquad \mathcal{D}^{(1,5)}u_j = \frac{1}{12h}\left(u_{j-2} - 8u_{j-1} + 8u_{j+1} - u_{j+2}\right).$$
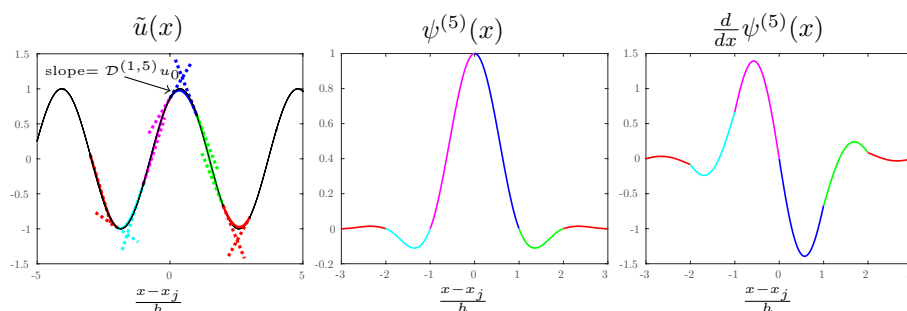
FIG. 3.1. *Left: Illustration of piecewise polynomial reconstruction of $\tilde{u}(x)$ from discrete data u_k. The thin black line represents $u(x)$, while the interpolant is represented by the thicker colored solid lines. The dashed colored lines indicate the value of the approximated derivative at each node point. Center: The GD D-Spline basis function for $p = 5$. Right: First derivative of the GD D-Spline basis function.*

Piecewise interpolation in this manner is called D-Spline interpolation.[1] This D-Spline construction is illustrated graphically in Figure 3.1, which shows data from a sinusoid being fit using piecewise degree 7 polynomials with $p = 5$.

In this work we restrict ourselves to uniform partitions, $x_{j+1} - x_j = h$. Note that curvilinear coordinate systems can be handled in the usual manner for difference methods by solving the equations in a transformed coordinate system. In the reference space a uniform partition can be used. Such an approach is taken in the curvilinear example in section 5.3. It is convenient to introduce local scaled coordinates $\xi = \frac{x - x_j}{h}$, and the restriction of the D-Spline interpolant to the interval (x_j, x_{j+1}) may be represented as

$$(3.4) \qquad \tilde{u}_j(x) = \sum_{\alpha=-q+1}^{q} H_{\alpha,0}^{(p,1)}(\xi) u_{\alpha+j} + \sum_{\alpha=0}^{1} H_{\alpha,1}^{(p,1)}(\xi) h \mathcal{D}^{(1,p)} u_{\alpha+j},$$

where $H_{\alpha,\beta}^{(p,1)}$ are the Hermite basis functions, i.e., the unique polynomials of degree $p + 2 = 2q + 1$ satisfying the following conditions:

$$(3.5a) \qquad H_{\alpha,0}^{(p,1)}(\nu) = \delta_{\alpha\nu} \qquad \text{for } \alpha, \nu \in \{-q+1, -q+2, \ldots, q\},$$

$$(3.5b) \qquad \left(\frac{d}{dx}\right) H_{\alpha,0}^{(p,1)}(\nu) = 0 \qquad \text{for } \alpha \in \{-q+1, -q+2, \ldots, q\}, \ \nu \in \{0, 1\},$$

$$(3.5c) \qquad \left(\frac{d}{dx}\right) H_{\alpha,1}^{(p,1)}(\nu) = \delta_{\alpha\nu} \qquad \text{for } \alpha, \nu \in \{0, 1\},$$

$$(3.5d) \qquad H_{\alpha,1}^{(p,1)}(\nu) = 0 \qquad \text{for } \alpha = 0, 1, \ \nu \in \{-q+1, -q+2, \ldots, q\}.$$

Note that D-Spline interpolants share many similarities to Hermite interpolants [22] but are different in two significant ways. First, the derivatives in D-Splines are taken to be defined from a discrete approximation using data at grid points, rather than given as exact values. Second, only the derivatives at the endpoints of the interval at x_j and x_{j+1} are constrained, as opposed to all grid points as in standard Hermite

---

[1]This name is meant to indicate that the D-Spline interpolant shares some commonality with splines in that derivatives are matched at data points. However, with D-Splines the derivatives match a local difference approximation rather than imposing a global constraint as in traditional splines.

interpolation. By constraining derivatives of the interpolant at particular locations, D-Splines have similarity to Hermite–Birkhoff interpolation [5].

3.1. D-Spline basis functions. The interpolating function $\tilde{u}(x)$ is linear in u_j and each u_j appears exactly in $p + 1 = 2q$ intervals. Therefore, basis functions $\psi_j^{(p,1)}(x)$ can be associated with each x_j so that $\tilde{u}(x) = \sum_j u_j \psi_j^{(p,1)}(x)$. Since $\tilde{u}(x)$ is linear in u_j, the basis functions can be determined by differentiation with respect to u_j, as in

$$(3.6) \qquad \psi_j^{(p,1)}(x) = \frac{\partial \tilde{u}(x)}{\partial u_j}.$$

The resulting basis functions are C^1 piecewise polynomials. Moreover, they are independent of j when written as functions of $\xi = \frac{x - x_j}{h}$:

$$(3.7) \qquad \psi^{(p,1)}(\xi) = \begin{cases} \psi_k^{(p,1)}(\xi), & k < \xi \le k + 1, \ k \in \{-q, \dots, q-1\}, \\ 0, & \xi \le -q \text{ or } \xi \ge q. \end{cases}$$

Define the undivided finite difference coefficients by

$$(3.8) \qquad h\mathcal{D}^{(1,p)} u_j = \sum_{\nu = 1-q}^{q-1} \eta_\nu^{(1,p)} u_{\nu+j}.$$

Then, with the convention $\eta_{-q}^{(1,p)} = \eta_q^{(1,p)} = 0$, we can write $\psi_k^{(p)}$ using the Hermite basis functions defined above:

$$(3.9) \qquad \psi_k^{(p,1)}(\xi) = H_{-k,0}^{(p,1)}(\xi - k) + \eta_{-k}^{(1,p)} H_{0,1}^{(p,1)}(\xi - k) + \eta_{-k-1}^{(1,p)} H_{1,1}^{(p,1)}(\xi - k).$$

The support of these basis functions is the same as the GD basis function from [3, 2], but with additional smoothness since the derivatives have been constrained at element boundaries. For example, Figure 3.1 illustrates the D-Spline basis function and its derivative for $p = 5$. Explicit formulae for the basis corresponding to $p = 5$ are

(3.10)
$$\psi_{-3}^{(5,1)}(\xi) = -\frac{(\xi+3)^7}{180} + \frac{7(\xi+3)^6}{360} + \frac{(\xi+3)^5}{36} - \frac{7(\xi+3)^4}{72} - \frac{(\xi+3)^3}{45} + \frac{7(\xi+3)^2}{90},$$

$$\psi_{-2}^{(5,1)}(\xi) = \frac{(\xi+2)^7}{36} - \frac{7(\xi+2)^6}{72} - \frac{5(\xi+2)^5}{36} + \frac{19(\xi+2)^4}{36} + \frac{7(\xi+2)^3}{36}$$
$$\qquad - \frac{31(\xi+2)^2}{72} - \frac{\xi+2}{12},$$

$$\psi_{-1}^{(5,1)}(\xi) = -\frac{(\xi+1)^7}{18} + \frac{7(\xi+1)^6}{36} + \frac{5(\xi+1)^5}{18} - \frac{41(\xi+1)^4}{36} - \frac{7(\xi+1)^3}{18}$$
$$\qquad + \frac{13(\xi+1)^2}{9} + \frac{2(\xi+1)}{3},$$

$$\psi_0^{(5,1)}(\xi) = \frac{\xi^7}{18} - \frac{7\xi^6}{36} - \frac{5\xi^5}{18} + \frac{11\xi^4}{9} + \frac{2\xi^3}{9} - \frac{73\xi^2}{36} + 1,$$

$$\psi_1^{(5,1)}(\xi) = -\frac{(\xi-1)^7}{36} + \frac{7(\xi-1)^6}{72} + \frac{5(\xi-1)^5}{36} - \frac{47(\xi-1)^4}{72} + \frac{(\xi-1)^3}{18}$$
$$\qquad + \frac{19(\xi-1)^2}{18} - \frac{2(\xi-1)}{3},$$

$$\psi_2^{(5,1)}(\xi) = \frac{(\xi - 2)^7}{180} - \frac{7\,(\xi - 2)^6}{360} - \frac{(\xi - 2)^5}{36} + \frac{5\,(\xi - 2)^4}{36} - \frac{11\,(\xi - 2)^3}{180}$$
$$- \frac{43\,(\xi - 2)^2}{360} + \frac{\xi - 2}{12}.$$

Although not used in this work, it is useful to recognize that smoother D-Spline basis functions, for example, C^d, $d \le p - 1$, can be constructed by applying additional constraints that equate higher derivatives of the interpolant to difference approximations of the corresponding derivatives. The resulting basis functions can be expressed

$$(3.11) \qquad \psi^{(p,d)}(\xi) = \begin{cases} \psi_k^{(p,d)}(\xi), & k < \xi \le k + 1, \\ 0, & \xi \le -q \text{ or } \xi \ge q, \end{cases}$$

where

$$(3.12) \quad \psi_k^{(p,d)}(\xi) = H_{-k,0}^{(p,d)}(\xi - k) + \sum_{\beta=1}^{d} \left(\eta_{-k}^{(\beta,p)} H_{0,\beta}^{(p,d)}(\xi - k) + \eta_{-k-1}^{(\beta,p)} H_{1,\beta}(\xi - k) \right),$$

the undivided central difference approximations of the βth derivative are

$$(3.13) \qquad h^\beta \mathcal{D}^{(\beta,p)} = \sum_{\nu=1-q}^{q-1} \eta_\nu^{(\beta,p)} u_{\nu+j},$$

and the Hermite basis functions are the unique polynomials of degree $p + 2d$ satisfying

$$(3.14\mathrm{a}) \qquad H_{\alpha,0}^{(p,d)}(\nu) = \delta_{\alpha\nu} \qquad\qquad \text{for } \alpha, \nu = -q + 1, \ldots, q,$$

$$(3.14\mathrm{b})$$
$$\left(\frac{d}{dx} \right)^\gamma H_{\alpha,0}^{(p,d)}(\nu) = 0 \qquad\qquad \text{for } \alpha = -q + 1, \ldots, q; \ \nu = 0, 1; \gamma = 1, 2, \ldots, d,$$

$$(3.14\mathrm{c})$$
$$\left(\frac{d}{dx} \right)^\gamma H_{\alpha,\beta}^{(p,d)}(\nu) = \delta_{\alpha\nu} \delta_{\beta\gamma} \qquad\qquad \text{for } \alpha, \nu = 0, 1; \ \ \beta, \gamma = 1, 2, \ldots d,$$

$$(3.14\mathrm{d}) \qquad H_{\alpha,\beta}^{(p,d)}(\nu) = 0 \qquad \text{for } \alpha = 0, 1; \ \nu = -q + 1, -q + 2, \ldots, q; \beta \ge 1.$$

High continuity D-Splines will be the subject of future work.

3.2. D-Spline interpolation error analysis. In order to estimate the error for the GD approximations, as well as for other applications of D-Splines, it is useful to study the error for D-Spline interpolants of smooth functions. Error formulas follow directly from the application of the Peano kernel theorem (e.g., [9, Chapter 5]) and the fact, stated in the following lemma, that the interpolant is exact for polynomials of degree $p - 1$.

LEMMA 3.1. *Let $P(x)$ be a polynomial of degree $p - 1$. Then for $x \in [x_j, x_{j+1}]$ and $\xi = (x - x_j)/h$*

$$(3.15) \qquad P(x) = \sum_{k=-q}^{q-1} \psi_k^{(p,1)}(\xi + k) P(x_{j-k}).$$

Proof of Lemma 3.1. We first note that the D-Spline interpolant on $[x_j, x_{j+1}]$ is the unique polynomial of degree $p + 2$ satisfying the interpolation conditions. Suppose two such polynomials exist and let $\delta(x)$ be their difference. Then $\delta(x)$ is a polynomial of degree $p + 2$ which is zero at x_ν, $\nu = j - q + 1, \ldots, j + q$, and whose derivative vanishes at x_j and x_{j+1}. Counting multiplicities, $\delta(x)$ has $p + 3$ zeros and thus vanishes

identically. Since $P(x)$ is a polynomial of degree $p - 1$, the p-point central difference formulas are exact. That is,

$$(3.16) \qquad hP'(x_j) = \sum_{\nu=1-q}^{q-1} \eta_\nu^{(1,p)} P(x_{\nu+j}), \quad hP'(x_{j+1}) = \sum_{\nu=2-q}^{q} \eta_\nu^{(1,p)} P(x_{\nu+j}).$$

Therefore P satisfies all the interpolation conditions and, by the uniqueness result, equals its D-Spline interpolant.   □

Remark 3.2. The same result holds for the D-Splines with higher continuity.

Now define the error functional for fixed $x \in [x_j, x_{j+1}]$:

$$(3.17) \qquad \mathcal{E}_x^{(p)}(u) = u(x) - \sum_{k=-q}^{q-1} \psi_k^{(p,1)}(\xi + k) u(x_{j-k}).$$

Since $\mathcal{E}_x^{(p)}$ is linear and by Lemma 3.1 $\mathcal{E}_x^{(p)}(P) = 0$ if P is a polynomial of degree $p - 1$ the Peano kernel theorem applies. More particularly its corollary, known as Kowalewski's formula for standard polynomial interpolation, applies.

THEOREM 3.3 (D-Spline interpolation error formula).   *Assume again that the function $u(x) \in C^p$. Let $\tilde{u}(x)$ be its D-Spline interpolant. Then the error, $u - \tilde{u}$, for $x \in [x_j, x_{j+1}]$, is given by*

$$(3.18) \qquad u(x) - \tilde{u}(x) = \frac{1}{(p-1)!} \sum_{k=-q}^{q-1} \psi_k^{(p,1)}(\xi + k) \int_{x_{j-k}}^{x} (x_{j-k} - t)^{p-1} u^{(p)}(t) dt.$$

Since the basis functions are independent of the mesh spacing, h, and

$$(3.19) \qquad \left| \int_{x_{j-k}}^{x} (x_{j-k} - t)^{p-1} u^{(p)}(t) dt \right| \le \frac{\max_{t \in [x_{j-k}, x]} \left| u^{(p)}(t) \right|}{p} |x_{j-k} - x|^p,$$

we have the immediate corollary that there exists a constant C_p independent of h and u such that, for $u(x) \in C^p$,

$$(3.20) \qquad |u(x) - \tilde{u}(x)| \le C_p h^p \max_{t \in [x_{j-q+1}, x_{j+q}]} \left| u^{(p)}(t) \right|.$$

We note that C_p can be easily estimated using the basis functions.

3.3. Comparison with B-Splines. The proposed methods based on the use of D-Splines can be compared to isogeometric methods which use B-Splines or NURBS (e.g., [12, 11]). There are two fundamental differences between D-Splines and B-Splines. For B-Splines the continuity and the local polynomial degree are in lock-step; if the degree is p the splines have $p - 1$ continuous derivatives and the approximation order is $p + 1$. For D-splines, on the other hand, for a fixed approximation order q, continuity up to $q - 1$ derivatives is enforced by increasing the polynomial degree within each subinterval; here we imposed the minimal continuity required by the problem but, as mentioned above, methods based on the maximal continuity are also possible. The support of the basis functions is determined purely by q, independent of the continuity imposed. Thus increasing the local degree increases neither the approximation order nor the bandwidth of the matrices in a Galerkin approximation. However, it does increase the cost of assembly. A second fundamental difference is

that the D-Spline basis functions are Lagrangian while B-Splines are not. Lagrange splines have global support. The fact that the D-Spline degrees of freedom (DOFs) are the function values at the nodes is a potentially useful feature of the approach. For example, this fact was found to be advantageous for solving time-domain integral equations [4], where related B-Spline-based methods are limited to second order. In the context of PDE discretizations, the Lagrangian nature of D-Spline interpolation is also likely to provide benefit, e.g., for nonlinear problems.

The references [12, 11] contain detailed studies of the spectra of B-spline-based approximations to the second derivative and in particular note the absence of so-called optical branches which appear in standard high-order finite element methods. Specifically, it is stated that all but a fixed number of outlier modes converge with increasing order, with the outliers localized at the boundary. We have not as yet carried out a systematic study of the spectra for D-Spline methods. However, a glance at Figure 4.3 shows no evidence of an optical branch with an outlier mode localized at the boundary. Moreover, using the direct enforcement boundary closure discussed below significantly decreased the size of the outlier. It is an interesting question if a similar modification could affect the outliers for B-Spline-based methods, but we have not considered it.

4. Discretizations of governing equations. Having formulated a set of basis functions for the D-Spline interpolant, standard Galerkin projection for the weak form of the operator can be used to discretize the governing equations from section 2. Away from boundaries, the discrete equations are translation invariant, and it is useful to pursue a dispersion analysis for the corresponding weak derivative operators. Such an analysis is presented in subsection 4.2. Because the basis functions span multiple nearby elements, closures must be employed near physical boundaries. A brief review of existing boundary closures previously discussed in [2, 3] is provided in subsection 4.3.1, and an analysis of the spectral properties of the resulting discrete equations motivates the development of a new boundary closure technique that we call *direct enforcement*; see subsection 4.4. This new boundary closure shows significant improvement in the spectral properties of the discrete operators. By leveraging standard finite element theory, we then provide a discussion on the convergence of the Galerkin discretization. Finally, since time-step constraints for explicit methods for biharmonic problems typically require $\Delta t \propto h^2$, an A-stable implicit SDC time-stepping method is discussed in subsection 4.6.

4.1. Finite element formulation of the governing equations. As with prior GD methods in [3, 2], the GD D-Spline method is fundamentally a Galerkin finite element scheme with a specific choice of basis functions. For clarity of exposition, we first focus on the beam equation (2.2) and then later briefly summarize results for the plate (2.3). For the 1D beam, assume that the domain is given as $\Omega = (x_a, x_b)$, where either x_a or x_b may be infinite. The weak-form discretization of (2.2) is constructed in the usual way through multiplying by a test function v and integrating by parts twice to give

$$(4.1) \qquad (v, \rho u_{tt}) = -(v_{xx}, EIu_{xx}) + v_x EIu_{xx}|_{x_a}^{x_b} - vEIu_{xxx}|_{x_a}^{x_b},$$

where $(f, g) = \int_{x_a}^{x_b} fg\,dx$, and the last two terms are boundary jump terms. Typically, the treatment of the boundary terms depends on the nature of the boundary conditions, which in this work will always be homogeneous. We will consider

   i. free (natural) boundary conditions: $u_{xx} = u_{xxx} = 0$;
   ii. clamped boundary conditions: $u = u_x = 0$;
   iii. simply supported boundary conditions: $u_x = u_{xxx} = 0$;
   iv. sliding boundary conditions: $u = u_{xx} = 0$.

In all cases the boundary terms will vanish due to the boundary conditions on u, imposition of the boundary conditions on v, or some combination of the two; for example, for simply supported boundary conditions we must enforce $v_x = 0$. However, we will find it advantageous to enforce additional contraints on the basis beyond those required by the weak formulation.

   We assume, in addition, that the constants EI and ρ are uniform in space and so may be taken out of the inner products. Now let the discrete trial and test functions be given as

$$(4.2) \qquad v_h(x,t) = \sum_\alpha v_\alpha \psi_\alpha^{(p)}(x), \qquad u_h(x,t) = \sum_\alpha u_\alpha \psi_\alpha^{(p)}(x),$$

where the limits of the summation are defined based on the physical domain and boundary closure. Away from boundaries the basis functions will simply be given by the D-Spline basis functions defined above, but there will be modifications near the boundaries. Substitution into (4.1) gives the D-Spline discretization of the weak form as

$$(4.3) \qquad \rho\left(\sum_\alpha v_\alpha \psi_\alpha^{(p)}(x), \sum_\alpha u_{\alpha,tt}\psi_\alpha^{(p)}(x)\right) = -EI\left(\sum_\alpha v_\alpha \psi_{\alpha,xx}^{(p)}, \sum_\alpha u_\alpha \psi_{\alpha,xx}^{(p)}\right).$$

This is further reduced to a system of linear equations

$$(4.4) \qquad \rho M^{(p)}\mathbf{u}_{tt} = -EIK^{(4,p)}\mathbf{u},$$

where $\mathbf{u}$ is the vector of coefficients, $M^{(p)}$ is the mass matrix, and $K^{(4,p)}$ is the stiffness matrix for the fourth derivative whose entries are given by

$$(4.5) \qquad M_{\alpha,\beta}^{(p)} = \left(\psi_\alpha^{(p)}, \psi_\beta^{(p)}\right), \qquad K_{\alpha,\beta}^{(4,p)} = \left(\psi_{\alpha,xx}^{(p)}, \psi_{\beta,xx}^{(p)}\right).$$

Stiffness matrices for other derivative operators with d-spatial derivatives can be similarly defined and will be notated $K^{(d,p)}$. The mass and stiffness matrices are banded with bandwidth $2p+1$. Away from boundaries, the entries for each equation are all identical (i.e., the discretization is translation invariant), and the matrices are symmetric or skew symmetric for even and odd derivative approximations, respectively. Taking advantage of these symmetries, the matrix coefficients can be expressed in a compact form, e.g., $M_{\alpha,\beta}^{(p)} = M_{\beta-\alpha}^{(p)}$, where $M_\gamma^{(p)}$ are the coefficients for the mass matrix relative to the diagonal. As a complete example, Table 4.1 lists the mass and stiffness matrix entries for $p = 5$.

   A similar approach to the plate equation using a tensor product construction gives

$$(4.6) \qquad \rho M^{(x,p)} \otimes M^{(y,p)}\mathbf{u}_{tt}$$
$$= -D\left(K^{(4,x,p)} \otimes M^{(y,p)} + 2K^{(2,x,p)} \otimes K^{(2,y,p)} + M^{(x,p)} \otimes K^{(4,y,p)}\right)\mathbf{u},$$

where $M^{(x,p)}, K^{(2,x,p)}, K^{(4,x,p)}$ are the mass, second derivative stiffness, and fourth derivative stiffness matrices with the inner product taken in the x-direction, with the y-direction similarly defined. For more details about the tensor product construction of multidimensional GDs see [3].

TABLE 4.1
Coefficients of the mass and stiffness matrices for $p = 5$ basis functions.

γ	0	1	2	3	4	5
$h^{-1}M_\gamma^{(5)}$	$\frac{18407531}{21621600}$	$\frac{88995391}{778377600}$	$\frac{-196283}{3891888}$	$\frac{5131}{449280}$	$\frac{-1233509}{1167566400}$	$\frac{40571}{667180800}$
$K_\gamma^{(1,5)}$	0	$\frac{732187}{950400}$	$\frac{-427277}{2494800}$	$\frac{50773}{1900800}$	$\frac{-4663}{2138400}$	$\frac{17329}{119750400}$
$hK_\gamma^{(2,5)}$	$\frac{-18672743}{7722000}$	$\frac{6973763}{5559840}$	$\frac{-158809}{6949800}$	$\frac{-1223419}{43243200}$	$\frac{721639}{116756640}$	$\frac{-239303}{833976000}$
$h^2 K_\gamma^{(3,5)}$	0	$\frac{-23879}{10800}$	$\frac{60911}{37800}$	$\frac{-6959}{16800}$	$\frac{14291}{226800}$	$\frac{-1919}{453600}$
$h^3 K_\gamma^{(4,5)}$	$\frac{442273}{29700}$	$\frac{-157723}{14256}$	$\frac{280111}{62370}$	$\frac{-341927}{332640}$	$\frac{25253}{149688}$	$\frac{-208871}{14968800}$

4.2. Dispersion analysis for weak derivatives on an infinite domain.
Owing to the construction of the method, the GD discretization is translation invariant
on a uniform grid. Thus on an infinite domain, or with periodic boundary conditions,
Fourier analysis can be used to investigate the accuracy of the scheme. In particular,
$(M^{(p)})^{-1}K^{(d,p)}$ gives a weak finite element operator approximating the dth derivative
with basis functions $\psi^{(p)}$. The action of this operator on a grid function $u_j = \exp(ikx)$
then gives the discrete Fourier symbol of the GD D-Spline approximation to the
derivatives as

$$(4.7) \qquad (ik)^d \approx \frac{\sum_{-\gamma=-p}^{p} K_\gamma^{(d,p)} \exp(i\gamma\chi)}{\sum_{-\gamma=-p}^{p} M_\gamma^{(p)} \exp(i\gamma\chi)} \equiv \frac{1}{h^d}\mathcal{S}^{(d,p)}(\chi),$$

where k is a physical wave number and $\chi = kh$ is a grid wave number. Expanding $\mathcal{S}$
in powers of χ then exposes the local error of the discrete operator.

As an example, for $p = 5$ the symbols, $\mathcal{S}^{(d,p)}(\chi)$ are given by (4.8).

$$(4.8a) \qquad \mathcal{S}^{(1,5)}(\chi) = i\chi - \frac{80723}{1459458000}i\chi^{11} + \mathcal{O}(\chi^{13}),$$

$$(4.8b) \qquad \mathcal{S}^{(2,5)}(\chi) = -\chi^2 - \frac{448229}{1667952000}\chi^{10} + \mathcal{O}(\chi^{12}),$$

$$(4.8c) \qquad \mathcal{S}^{(3,5)}(\chi) = -i\chi^3 + \frac{4121}{2851200}i\chi^{11} + \mathcal{O}(\chi^{13}),$$

$$(4.8d) \qquad \mathcal{S}^{(4,5)}(\chi) = \chi^4 + \frac{29141}{2138400}\chi^{10} + \mathcal{O}(\chi^{12}).$$

Therefore for $p = 5$, the error converges at 10th order for the first derivative, 8th
order for the second and third derivatives, and 6th order for the fourth derivative.
As a comparison, the traditional GD basis functions for $p = 5$ in [2] reveals 10th-
order errors in the second derivative. In general the dispersion error order of the
GD operator is higher than that of the GD D-Spline operator for the first and second
derivatives, but it is not possible to construct the third and fourth derivative operators
with the standard GD basis functions.

Fourier analysis can be performed for other p, and the errors in the symbol of
the discrete operator using GD D-Splines for $p = 5, 7, 9, 11$ are shown in Figure 4.1.
One useful way to interpret the error in the discrete symbols is in terms of points per
wavelength (PPW), which is related to the grid wave number χ by $PPW = 2\pi/\chi$.
The error in the discrete symbols is plotted versus PPW in Figure 4.1. The dashed
reference line indicates an error of 10^{-4} and is included in the plots for convenient
reference. For example, to achieve an error of 10^{-4} in the fourth derivative operators
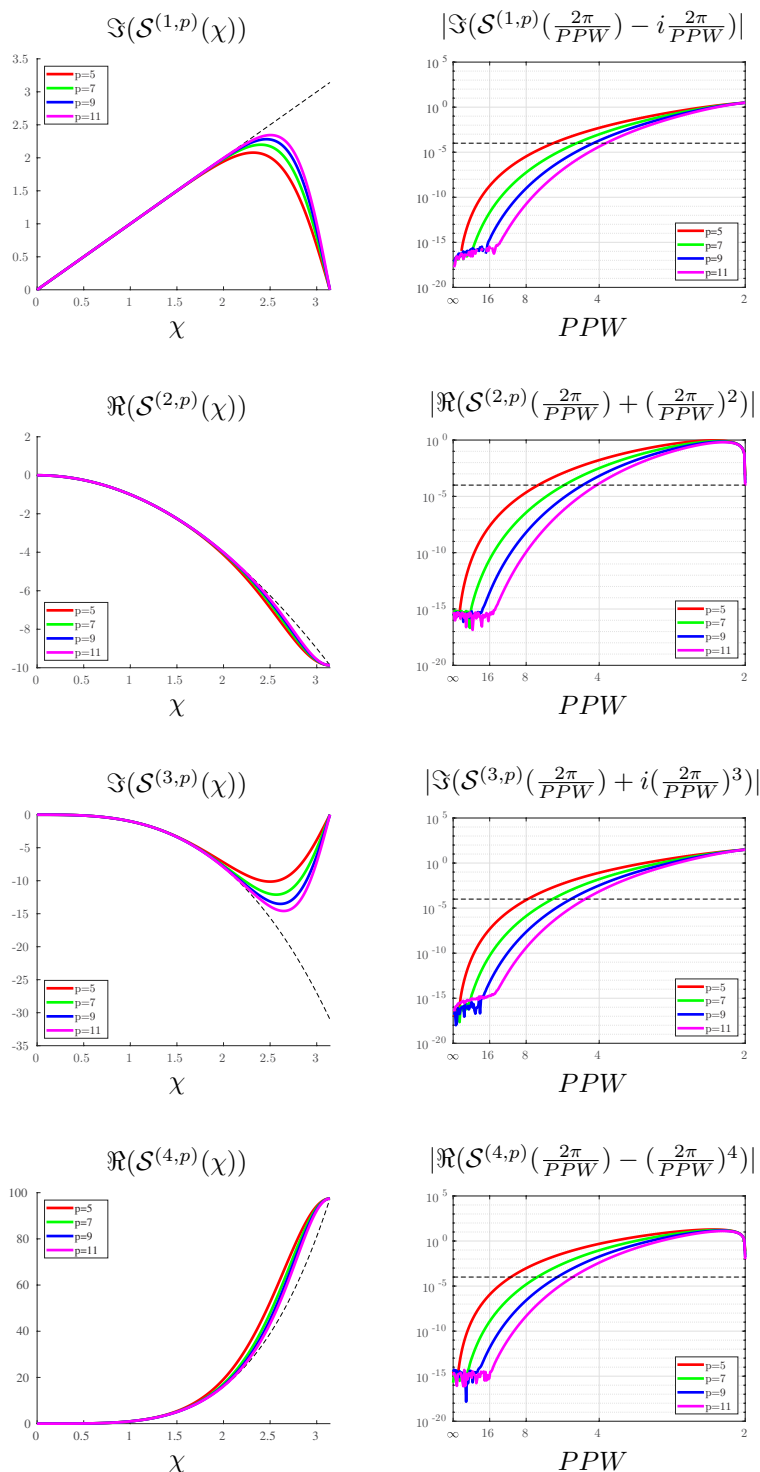
Fig. 4.1. *Plots of the symbol, $\mathcal{S}^{(d,p)}(\chi)$ (left), and the error of the symbol (right) for discrete operator for approximating the dth derivative using the D-Spline method with $p = 5, 7, 9, 11$. In the plots of the error, a reference line is included to indicate an error of 10^{-4}.*

corresponding to $p = 5$ and $p = 11$ one would need roughly 10 and 5 PPW, respectively. Note that the Nyquist limit implies that at least 2 PPW must be used to represent a signal. However, Figure 4.1 makes clear that high-order methods achieve double precision accuracy using modest numbers of PPW, e.g., ≈ 15 PPW for $p = 11$.

4.3. Boundary closures. The construction of GD D-Spline basis functions relies on neighboring data and the support of basis functions span multiple elements. Thus near boundaries, boundary closures must be utilized. Two existing generic methods from [2, 3], ghost basis and extrapolation, are explored in subsection 4.3.1 for application to the beam equation. However, for high-derivative operators the resulting discrete equations were found to have unacceptably large spurious spectral radii. Therefore, in subsection 4.4 a new alternate closure technique, dubbed *direct enforcement*, is developed.

4.3.1. Existing closures.
Ghost basis. The basic idea of the ghost basis closure is to maintain DOFs associated with the ghost nodes and evolve them just as any other DOF. As with other Galerkin approaches, the GD discretization is based on L^2 inner products over the domain interior. Due to the finite support of the GD basis functions, only a finite number of DOFs will contribute to the inner products, including DOFs associated with points outside the domain. For additional details concerning the mechanics of the ghost basis closure, refer to [2, 3].

Extrapolation. In the extrapolation closure, the solution values at the ghost nodes are related to solution values in the domain interior via an extrapolation formula. The ghost node can then be algebraically eliminated from the system. Alternately, one can view the extrapolation closure as a modification of the basis functions near the boundary in accordance with the extrapolation formula. As a concrete example, take $p = 5$ and thus $N_g = 2$ ghost points are needed. The ghost values at a left boundary with $j = 0$ can be obtained via sixth-th order extrapolation as given by

$$(4.9a) \qquad u_{-1} = 6u_0 - 15u_1 + 20u_2 - 15u_3 + 6u_4 - u_5,$$

$$(4.9b) \qquad u_{-2} = 6u_{-1} - 15u_0 + 20u_1 - 15u_2 + 6u_3 - u_4.$$

In general, one can use order $p + 1$ extrapolation to maintain the accuracy of the interpolant. For further details on the extrapolation closure, refer to [2, 3].

The use of ghost basis and extrapolation closures has an effect on the spectral properties of the resulting schemes. For the present discussion, we focus on the spectral radius, since it can play an important role in determining the overall efficacy of the discretization. For example, the spectral radius is important in determining the size of time-steps for explicit schemes, or the efficiency of linear solvers needed for implicit schemes.

Consider the spectral radius of the fourth derivative operator $D^{(4,p)} = (M^{(p)})^{-1} K^{(4,p)}$, which will be denoted $\rho^{(4,p)}$. The eigenvalues of the discrete operator corresponding to $p = 5, 7, 9, 11, 13, 25$ for $\Omega = (0, \pi)$ are computed for the ghost basis and extrapolation closures using $N = 200$ points. Natural boundary conditions are applied on both ends (i.e., free boundary closures for a beam), so that no constraints on the basis functions are required for the weak formulation. Note that the continuous operator applied to the grid function gives the ideal bound of $\pi^4/h^4 \approx 97.41$, and it is therefore sensible to normalize the results by this value. Table 4.2 provides the normalized spectral radius $\rho^{(4,p)} h^4/\pi^4$. From the table, the normalized radius is seen to grow quite rapidly with respect to p for both closures. To further understand this spurious

TABLE 4.2

Normalized spectral radius, $\rho^{(4,p)}h^4/\pi^4$ for various boundary closures and free ends. Values significantly larger than 1 indicate spurious behavior in the discrete equations.

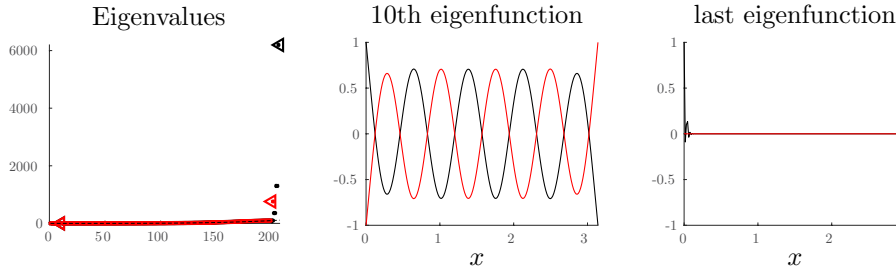| Closure	$p = 5$	$p = 7$	$p = 9$	$p = 11$	$p = 13$	$p = 25$
Ghost basis	21.2	63.6	287	405	445	2529
Extrapolation	2.38	7.80	16.4	33.4	59.5	589

FIG. 4.2. *At left, eigenvalues of the discrete operator for the ghost basis (black) and extrapolation (red) boundary closures with $p = 7$. At center, the eigenfunction of the discrete operator corresponding to the 10th smallest eigenvalue. At right, the eigenfunctions of the discrete operators corresponding to the largest eigenvalues.*

behavior for increasing p, it is useful to probe the spectra in more detail. In particular, Figure 4.2 illustrates the eigenvalues, and selected eigenfunctions, for the discrete operator for $p = 7$ with the two different boundary closures. The eigenfunctions associated with the largest eigenvalues are seen to be essentially boundary modes and do not correspond to any analytic eigenfunctions. In fact, for the extrapolation closure we see two such boundary modes, while for the ghost basis closure we see, in general, $N_g = p - 1$ such modes. On the other hand, eigenfunctions associated with small eigenvalues are seen to be accurate approximations to continuous eigenfunctions. For example, the 10th eigenfunction (plotted in Figure 4.2) approximates the 10th analytic eigenfunction with an error on the order of 10^{-9}. In addition, we observed that the eigenvalues of the discrete operators ordered from small to large tend to approximate $(kh)^4$, where k is the index of the eigenvalue. This curve is illustrated in Figure 4.2 by the dashed black line.

4.4. Direct enforcement boundary closure. As discussed in subsection 4.3.1, the spectral properties of the fourth derivative operator with the ghost basis and extrapolation closures are less than ideal. To address similar spurious behavior for wave equations, the work in [2, 3] introduced compatibility closures. Remarkably, these closures showed no growth in the spectral radius, although it was unclear how to formulate these closures for general equations. Indeed it is unclear how these ideas might be extended to the equations governing beams or plates. Therefore, we now pursue a new boundary closure that is, in some sense, a middle ground between compatibility and extrapolation closures.

For the biharmonic problems considered in this work, two boundary conditions must be supplied on each boundary. For simplicity, let us focus on the beam equation in one dimension, with extension to 2D plates being straightforward. Consider the physically motivated boundary conditions leading to continuous energy estimates as discussed above. Depending on the boundary condition considered, the weak form may require setting $v = 0$, $v_x = 0$, or both, and this can be simply enforced as the basis functions are C^1. However, for all cases other than clamped boundary conditions, a

strong solution satisfies conditions involving u_{xx}, u_{xxx}, or both. The idea of the direct enforcement closure is to enforce these conditions directly in the interpolant (equivalently the basis) and use extrapolation conditions to define any remaining ghost DOFs. Because the D-Spline interpolant uses discrete approximations of derivatives in its construction, it is natural to adopt a similar approach here and use the following approximations for the higher-order boundary derivatives at $x = 0$:

(4.10a) $$u(0) = u_0,$$

(4.10b) $$u_x(0) = \mathcal{D}^{(1,p)}u_0,$$

(4.10c) $$u_{xx}(0) \approx \mathcal{D}^{(2,p)}u_0,$$

(4.10d) $$u_{xxx}(0) \approx \mathcal{D}^{(3,p)}u_0.$$

Here $\mathcal{D}^{(d,p)}u_0$ is the centered finite difference formulas for approximating the dth derivative at x_0 in a stencil of size p. Depending on the boundary conditions, two of these quantities will be set to zero. The remaining $N_g - 2$ equations are given by order $p + 1$ extrapolation equations. We take these to be undivided central difference approximations to the $p + 1$st derivative centered at $x_0, x_1, \ldots, x_{N_g-3}$. These discrete equations are then considered as generic algebraic relationships between ghost and interior DOFs. Therefore, as advocated in [2, 3], the ghost DOFs can be eliminated from the system, or equivalently the basis can be modified according to the conditions.

As a concrete example, for the free end of a beam at $x = 0$ and $p = 7$, the equations imposed at the boundary are

(4.11a)
$$\mathcal{D}^{(2,7)}u_0 = \frac{1}{90}u_{-3} - \frac{3}{20}u_{-2} + \frac{3}{2}u_{-1} - \frac{49}{18}u_0 + \frac{3}{2}u_1 - \frac{3}{20}u_2 + \frac{1}{90}u_3 \qquad = 0,$$

(4.11b)
$$\mathcal{D}^{(3,7)}u_0 = \frac{1}{8}u_{-3} - u_{-2} + \frac{13}{8}u_{-1} - \frac{13}{8}u_1 + u_2 - \frac{1}{8}u_3 \qquad = 0,$$

(4.11c)
$$E^8 u_0 = -u_{-3} + 8u_{-2} - 28u_{-1} + 56u_0 - 70u_1 + 56u_2 - 28u_3 + 8u_4 - u_5 \quad = 0.$$

$E^{(8)}u_0$ is the eighth-order accurate extrapolation formula that effectively sets the eighth derivative to zero at the boundary. This gives three equations for the three unknowns u_{-3}, u_{-2}, u_{-1}.

Once again, the boundary closure has an effect on the spectral properties of the scheme. Following the discussion in subsection 4.3.1, we again measure the normalized spectral radius $\rho^{(4,p)}$ of the discrete operator for various boundary conditions utilizing the direct enforcement boundary closure. As before, the domain is set as $\Omega = (0, \pi)$, and $N = 200$ points are used. The normalized spectral radii for the $p = 5, 7, 9, 11, 13, 25$ and various boundary conditions are reported in Table 4.3. One interesting note is that the spectral radius is not monotonic in p, but as a general rule it does appear to grow slowly with p. Comparing the results in Table 4.2 with those in Table 4.3, the advantage of the direct enforcement closure is immediately clear. As such, the remainder of the results presented in this manuscript will adopt the direct enforcement closure.

As discussed in subsection 4.3.1, it is again useful to probe the spectrum of the discrete operator in slightly more depth. Figure 4.3 illustrates the eigenvalues and a selected eigenfunction for the discrete operator with $p = 7$ and the direct enforcement

TABLE 4.3
Normalized spectral radius, $\rho^{(4,p)}h^4/\pi^4$, with the direct enforcement boundary closure (BC) and various boundary conditions.

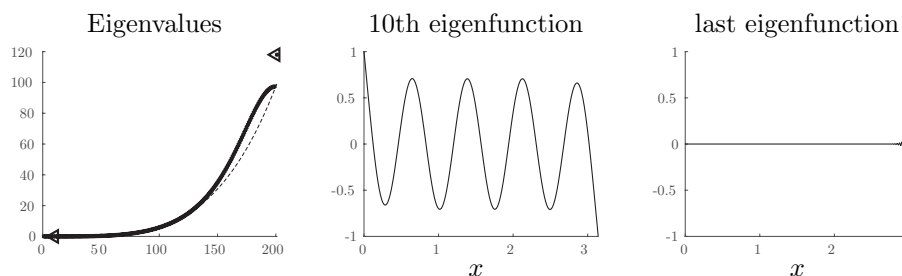| BC	$p=5$	$p=7$	$p=9$	$p=11$	$p=13$	$p=25$
Free	1.0003	1.2115	1.0151	1.0518	1.0955	3.6465
Clamped	1.0307	1.0025	1.1496	1.6319	2.5405	16.889
Simply supported	1.0003	0.9998	0.9996	1.1013	1.7222	11.398
Sliding supports	1.0007	1.9467	1.2978	1.2608	1.5193	5.2813

FIG. 4.3. *At left, eigenvalues of the discrete operator for the direct enforcement boundary closure ($p = 7$). At center, the eigenfunction of the discrete operator corresponding to the 10th smallest eigenvalue. At right, the eigenfunctions of the discrete operator corresponding to the largest eigenvalue.*

boundary closure for free boundary closures. As before, the eigenfunctions associated with the largest eigenvalues are essentially boundary modes; however, the support width of the mode is approximately double that of the boundary mode in the extrapolation closure in Figure 4.2. Also as before, the error between the 10th analytic eigenfunction and the 10th numerical eigenfunction is found to be on the order of 10^{-9}. Finally, the dashed line in Figure 4.3 indicates the continuous spectrum $(kh)^4$, where k is the index of the eigenvalue, and good agreement with the discrete spectrum is observed.

4.5. Convergence theory. Convergence of the GD D-Spline discretization follows from the approximation properties of the D-Spline in section 3 and straightforward application of standard FEM theory (for example, the analysis presented in [1] or [20]). Consider the spatially discrete, time continuous problem. Adapting the methods presented in [1, 20] gives the following theorem.

THEOREM 4.1 (convergence of GD D-Splines). *Assume the solution to* (2.2) *or* (2.3) *is sufficiently smooth, the initial data is approximated by L^2-projection onto the D-Spline basis, and the boundary conditions are one of the following type: periodic, fixed, free, simply supported, or sliding. Then there exists a constant $C(T)$ for the problem given by* (2.2) *such that*

$$(4.12) \qquad ||u_h(\cdot,T) - u(\cdot,T)||_{L^2} \le C(T)h^p$$

and a constant $\tilde{C}(T)$ for the problem given by (2.3) *such that*

$$(4.13) \qquad ||u_h(\cdot,\cdot,T) - u(\cdot,\cdot,T)||_{L^2} \le \tilde{C}(T)h^p,$$

where h is a measure of the grid spacing.

Proof. The proof follows from a modification of the proof of Theorem 3.1 in [1]. Let

$$(4.14) \qquad a(u,v) = \int_0^L EI u_{xx} v_{xx} dx$$

be the bilinear form associated with (2.2). It follows that the Cauchy–Schwarz inequality and the coercivity of the bilinear form $a(\cdot,\cdot)$ hold. Namely

$$(4.15a) \qquad |a(u,v)| \leq C_1 ||u||_{H^2} ||v||_{H^2},$$

$$(4.15b) \qquad a(u,u) \geq C_2 ||u||^2_{H^2} - C_3 ||u||^2_{L^2}$$

for $C_1, C_3 < \infty, \ C_2 > 0$.

Let u_h be the approximate solution and w_h be the solution to the related elliptic problem

$$(4.16) \qquad a(w_h,v) + (w_h,v) = a(u,v) + (u,v)$$

for all appropriate v. (Here $(\cdot,\cdot)$ denotes the L^2 inner product—it is added to handle boundary conditions admitting a null mode.) Then define $\eta = u - w_h, \psi = u_h - w_h, l = u - u_h$ as in [1]. It then follows that

$$(4.17) \qquad \begin{aligned} (\psi_{tt}, v) + a(\psi, v) &= -(w_{h,tt}, v) - a(w_h, v) \\ &= -(w_{h,tt}, v) - a(u, v) \\ &= -(\eta_{tt}, v). \end{aligned}$$

Now through the same arguments in [1], we arrive at

$$(4.18) \qquad ||l||_{L^2} \leq ||\eta(T)||_{L^2} + 2\sqrt{T} ||\eta_t(T)||_{L^2} + \sqrt{2} ||\eta(0)||_{L^2} + \sqrt{2} ||l(0)||_{L^2}.$$

By definition of η, l and Theorem 3.3, it follows that $||l(0)|| \leq Ch^p$ and $||\eta(0)|| \leq Ch^p$. Furthermore, by a generalization of Lemma 2.1 in [1], $||\eta(T)|| \leq Ch^p$ and $||\eta_t(T)|| \leq Ch^p$. The result then follows. Given that the assumptions in [1] about coercivity and definiteness of the operator still hold true, the error is directly related to the error in the interpolation of the solution. This error is given by Theorem 3.3 and the result follows. □

Remark 4.2. In practice, we use the discrete L^2-norm when measuring errors. In addition, we employ straightforward interpolation for the initial data, as opposed to L^2-projection. This modification does not appear to affect the overall performance of the scheme.

4.6. SDC time-stepping scheme. One final item in the description of the discretization is the choice of time-stepper. Here we consider the equations in their first-order temporal formulation

$$(4.19) \qquad \begin{bmatrix} u \\ \dot{u} \end{bmatrix}_t = \begin{bmatrix} \dot{u} \\ \nabla^2(D\nabla^2 u) \end{bmatrix},$$

where $\dot{u} = u_t$ is the time derivative of u. Classical stability analysis of explicit schemes for the biharmonic equations reveals a CFL-like constraint with $\Delta t \propto h^2$. Particularly for fine computational grids, this can be computationally burdensome. Instead, since the eigenvalues of the system lie on the imaginary axis, we wish to consider implicit A-stable schemes such as classical backward Euler or Crank–Nicholson. These schemes,

while stable, are only low-order accurate. To reveal the accuracy of the spatial discretization, we seek an A-stable scheme that is more accurate than the underlying finite element discretization.

In the present work, we choose to pursue an SDC method, similar to those discussed in [7, 8]. These methods make use of a low-order method, backward Euler in our case, and then use an iteration on a defect equation to achieve the desired order of accuracy. To avoid complicated notation, we describe the scheme for the simple test equation $y' = \lambda y$. Application of the time-stepper to discretizations of the 1D beam or 2D plate then follows with simple modifications.

Consider the initial value problem

$$(4.20a) \qquad\qquad y'(t) = \lambda y(t) \quad t \in [a, b],$$
$$(4.20b) \qquad\qquad y(a) = y_a,$$

with $\lambda, y \in \mathbb{C}$ as our problem of interest. Integrating (4.20) with respect to t yields the equivalent Picard equation

$$(4.21) \qquad\qquad y(t) = y_a + \int_a^t \lambda y(\tau) d\tau.$$

Suppose now that $y(t)$ is approximated by $\tilde{y}(t)$, with the residual in the Picard equation given as

$$(4.22) \qquad\qquad \epsilon(t) = y_a + \int_a^t \lambda \tilde{y}(\tau) d\tau - \tilde{y}.$$

The error $\delta(t) = y(t) - \tilde{y}(t)$ is then related to the residual as

$$(4.23) \qquad\qquad \delta(t) - \int_a^t \lambda \delta(s) ds = \epsilon(t),$$

which is also a Picard-type integral equation.

The integrals in the Picard equations can be approximated using numerical quadrature at sub-time-step nodes. The classic SDC scheme described in [7] uses Gauss points. As suggested by [8], we choose to use Gauss–Radau sub-time-step nodes, which has an added benefit that the last node aligns with the final time. Let $t_0 = a < t_1 < t_2 < t_m = b$ be the sub-time-step nodes with $t_1, t_2, \ldots, t_m$ the m Gauss–Radau points on the interval $(a, b]$. Define the initial backward Euler approximation $y^{[0]}$, and the nth approximate solution $y^{[n]}$ as m-vectors with entries $y_i^{[n]}$ where

$$(4.24) \qquad\qquad y_i^{[0]} = y_{i-1}^{[0]} + h_i \lambda y_i^{[0]}, \qquad h_i = t_i - t_{i-1},$$

and $y_i^{[n]}$ is to be defined. Furthermore, define the spectral integration matrix as S^m with entries

$$(4.25) \qquad\qquad S_{i,j}^m = \int_a^{t_i} \prod_{k \neq j} \frac{t - t_k}{t_j - t_k} dt.$$

The residual $\epsilon(t)$ is then simply the vector

$$(4.26) \qquad\qquad \sigma\left(y^{[n]}\right) = S^m \lambda y^{[n]} - y^{[n]} + y_a,$$

Algorithm 4.1. SDC algorithm for the model problem given by (4.20).

Data: $y_0^{[0]} = y_a, a, b, m, J$

Calculate m Radau points $t_i \in (a, b]$ and define $t_0 = a$ and $h_i = t_i - t_{i-1}$

    /* Calculate initial guess   */

for $i \leftarrow 1$ **to** m **do**

    Solve $y_i^{[0]} = y_{i-1}^{[0]} + h_i \lambda y_i^{[0]}$

end

Compute spectral integration matrix S^m

    /* Perform correction steps   */

for $j \leftarrow 0$ **to** $J - 1$ **do**

    Compute $\sigma^{[n]} = S^m \lambda y^{[n]} - y^{[n]} + y_a$

    Define $\delta_0^{[n]} = \sigma_0^{[n]} = 0$

    for $i \leftarrow 1$ **to** m **do**

        Solve $\delta_i^{[n]} = \delta_{i-1}^{[n]} + h_i \delta_i^{[n]} + \left(\sigma_i^{[n]} - \sigma_{i-1}^{[n]} \right)$

    end

    Update $y^{[n+1]} \leftarrow y^{[n]} + \delta^{[n]}$

end

return $y^{[J]}$

---

and the error $\delta_{[j]}$ is defined by

$$(4.27) \qquad \delta_i^{[n]} = \delta_{i-1}^{[n]} + h_i \delta_i^{[n]} + \left(\sigma \left(y^{[n]} \right)_i - \sigma \left(y^{[n]} \right)_{i-1} \right).$$

With these definitions the SDC time-stepper can be summarized as in Algorithm 4.1.

Remaining to be discussed is the number of sub-time-step points m and the number of correction steps J. The accuracy of the scheme is dependent on the accuracy of the quadrature, and so m governs the accuracy of the scheme (provided that enough correction steps have been taken). We therefore take $m = 10$, which corresponds to 19th-order accuracy for Gauss–Radau quadrature, which is sufficiently more accurate than the spatial schemes considered (13th order in space). To determine the number of correction steps, J, we must take into account both the accuracy and the stability of the time-stepper. Since the focus of the present work is the spatial discretization, we simply observe that using $J = 3m$, the accuracy of the scheme is limited only by that of the quadrature rule per Lemmas 3.1 and 3.2 in [8], and the scheme appears to be A-stable. The latter can be seen in Figure 4.4, which shows that the stability region encompasses the entire left half-plane where $\Re(\lambda) < 0$, and so the SDC time-stepper with $m = 10$, $J = 3m$ is A-stable as desired.

4.7. Computational cost. One important property of any algorithm is its computational cost. The present manuscript focuses on algorithm formulation, and there has been no particular effort at addressing computational cost. Nevertheless, there are certain features inherent to the method which bear on the question of expense, and we briefly discuss aspects of matrix assembly and matrix inversion. For both aspects it is useful to note that the mass and stiffness matrices are banded in one dimension and block banded in two dimentions, each with a bandwidth of $2p + 1$.
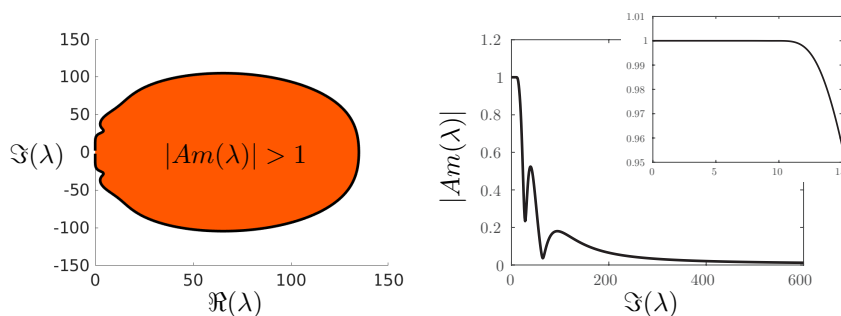
FIG. 4.4. *At left, the region of instability for the SDC scheme with $m = 10$, $J = 3m$ is colored orange. At right, the magnitude of the amplification factor of the scheme is plotted along the imaginary axis (inset is the growth factor on the imaginary axis for small λ). Taken together, the SDC time-stepper is shown to be A-stable.*

This is similar to other finite element approaches of similar asymptotic order. Since the method uses a single basis function per element (similar to a maximal continuity B-Spline FEM), all entries within the band are nonzero. This is in contrast to more traditional FEM approaches where many entries within the band are zero.

Before proceeding to discussion of cost, note that the method in two dimensions uses Kronecker products of 1D matrices to form the global matrix. The discussion of assembly therefore addresses the construction of 1D matrices. To discuss the cost of matrix assembly, one only need realize that each nonzero element in the matrix requires the evaluation of an inner product between two basis elements or their derivatives. As is typical, Gaussian quadrature is used to perform the integrations exactly. The primary differentiator of GD is that because the basis is piecewise continuous, each integral must be broken into $p + 1$ smaller integrals over each smooth piece, thus increasing the cost as compared to spline-based methods that may use a single quadrature. However, in comparison to traditional FEM approaches that have more than one basis function per element, the matrices for GD are much smaller and the cost is commensurately decreased. Note also that since both spline-based methods and GD are based on function spaces which are subspaces of larger, more traditional FEM function spaces, it may be possible to apply projections to arrive at equivalent schemes. Such an approach may have a significant impact on overall cost. Last, in regions where the material properties are constant, the translation-invariance of the discrete operators renders the assembly cost negligible.

To consider the cost of matrix inversion for the present method, there is no differentiation between other approaches. In the current implementation we use a direct sparse banded solver whose behavior is entirely dictated by the band structure. One might also consider a sparse iterative solver, whose performance would likely be significantly determined by the application of suitable preconditioners, which is beyond the scope of this work. Also, since the problem is linear one could conceivably perform a matrix factorization and store it in order to reduce cost. In the present work such an approach is complicated by the fact that each stage in the SDC integrator requires the solution of a different linear system. It may be possible to reformulate the integrator so that each stage uses the same matrix, e.g., [8], but this possibility has not been pursued here. If explicit time integrators are used, it is possible to formulate the method to exploit the tensor-product structure of the mass matrix, even if the material properties are nonconstant or if the grids are mapped; see [3, 14].

5. Numerical tests. In this section we verify the theoretical predictions of the convergence rates for the GD D-Spline discretization of both beams and plates on a variety of simple geometries for which the exact solution is known. Numerical solutions are computed to an $\mathcal{O}(1)$ time, and the errors are shown to behave as predicted for a variety of p. In addition, an example extension to non-Cartesian geometry is presented.

5.1. Time dependent Euler–Bernoulli beam. Consider a beam of length L with $EI = \rho = 1$. The left end of the bar will be located at $x = 0$ and the right end at $x = L$. Given the four boundary conditions associated with a beam (free, clamped, simply supported, and sliding supports), we will present results for two separate scenarios of an Euler–Bernoulli beam. The first situation is that of a cantilever beam where the left end of the bar uses a clamped (essential) boundary closure, while the right end has a free (natural) boundary closure. The full governing system for this case is given as

$$(5.1a) \qquad \rho u_{tt} = - EI u_{xxxx}, \qquad x \in (0, L),$$

$$(5.1b) \qquad u(0) = 0, \qquad u_{xx}(L) = 0,$$

$$(5.1c) \qquad u_x(0) = 0, \qquad u_{xxx}(L) = 0.$$

Setting $L = 10$ and seeking separable solutions of the form $u = f(x)g(t)$ leads to a BVP for the spatial eigenfunction f

$$(5.2a) \qquad f_{xxxx} - \lambda^4 f = 0, \qquad x \in (0, L),$$

$$(5.2b) \qquad f(0) = 0, \qquad f_{xx}(L) = 0,$$

$$(5.2c) \qquad f_x(0) = 0, \qquad f_{xxx}(L) = 0,$$

where λ is a separation constant. We pick $\lambda = .65\pi$, which ultimately yields the exact solution

$$(5.3)$$
$$u_e(x,t) = [\cos(.65\pi x) - S \sin(.65\pi x) - \cosh(.65\pi x) + S \sinh(.65\pi x)] \cos((.65\pi)^2 t),$$

where $S \approx 0.99999999729240505925690522824599$ is computed to high precision in order to avoid cancellation errors. The initial condition $u_e(x, t = 0)$, i.e., the spatial eigenfunction, is shown in Figure 5.1.

For a second test, consider the two other common types of boundary condition for the beam. Specifically, consider a simple support on the left and a sliding support on the right. The full problem is given as
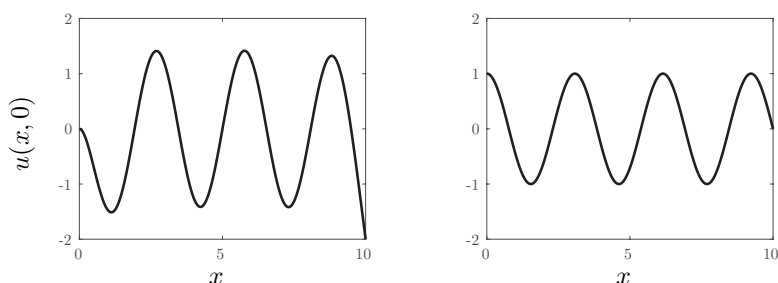

FIG. 5.1. *Left: Initial condition for the cantilever beam whose solution is given by* (5.3)*; Right: Initial condition for a beam with simple support at left and sliding support at right whose solution is as given by* (5.5)*.*

(5.4a) $$\rho u_{tt} = -EI u_{xxxx}, \qquad x \in (0, L),$$

(5.4b) $$u_x(0) = 0, \qquad u(L) = 0,$$

(5.4c) $$u_{xxx}(0) = 0, \qquad u_{xx}(L) = 0.$$

Again taking $L = 10$ and seeking separable solutions leads to the exact solution

(5.5) $$u_e(x, t) = \cos(.65\pi x)\cos((.65\pi)^2 t).$$

This initial condition $u_e(x, t = 0)$, i.e., the spatial eigenfunction, is also shown in Figure 5.1.

For both test cases, solutions are computed to a final time $t_f = 3$, which is chosen so as not to correspond to a node or antinode of the standing-wave oscillation. Results will be presented for the GD D-Spline scheme with $p = 5 : 2 : 13$ using the direct enforcement boundary closure. Note that the notation $a : s : b$ is the MATLAB style notation for a list starting at a and incrementing by s to b. To illustrate the properties of the schemes, convergence studies using $N = 10 : 5 : 100$ and the discrete L^2-norm are performed for each p. The results are presented in Figure 5.2. As predicted by Theorem 4.1, the errors for both test cases are seen to converge at a rate of p. Also shown in the figure are the errors in the approximate solution at $t_f = 3$ for $p = 9$ and $N = 55$, which show no severe error localization (e.g., near the boundaries).
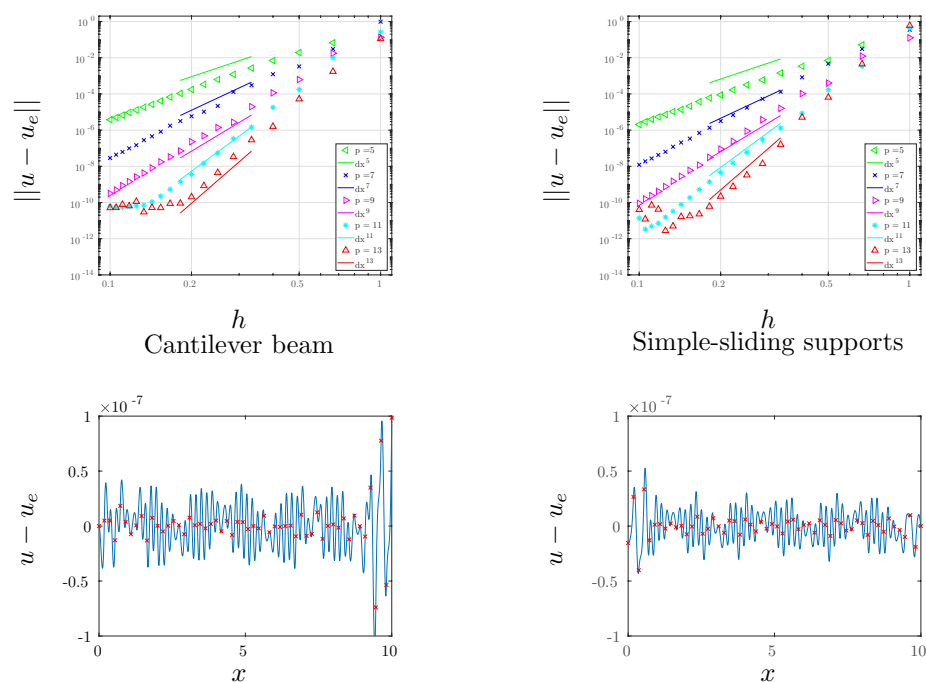

FIG. 5.2. *Top left: Convergence results using the discrete L^2-norm for the cantilever beam at $t_f = 3$. Bottom Left: Error as a function of x for the cantilever beam using $p = 9, N = 55, t_f = 3$. The red x's indicate errors at the points x_j. Top right: Convergence results using the discrete L^2-norm for the cantilever beam with simple support at one end and sliding support on the opposite end at $t_f = 3$. Bottom Right: Error as a function of x for the simple-sliding beam using $p = 9, N = 55, t_f = 3$.*

5.2. Time dependent Kirchhoff–Love plate. Consider now a Kirchhoff–Love plate in two spatial dimensions. Unfortunately, exact closed-form solutions for the plate can be challenging to derive. However, the eigenvalue problem for a plate with simply supported and sliding boundary conditions does yield a simple closed-form solution, and so we adopt those boundary closures here for testing purposes. In particular, consider the domain $\Omega = (0, \frac{29\pi}{2}) \times (0, \frac{29\pi}{2})$, with sliding support at left and right and simple supports at right and bottom. A depiction of the domain of interest with boundary conditions and governing equation is illustrated in Figure 5.3, where the blue boundaries indicate a simple support and the red boundaries indicate a sliding support. Solutions to this problem are of the form

$$(5.6) \qquad u(x, y, t) = \cos(k_x x) \sin(k_y y) \cos\left(\left(k_x^2 + k_y^2\right) t\right).$$

We take $k_x = k_y = 1$ and integrate to a final time $t_f = 3$ using the GD D-Spline method. Spatial schemes corresponding to $p = 5 : 2 : 13$ and spatial grids corresponding to $N = N_x = N_y = 20 : 15 : 200$ are tested. This range of grids corresponds to approximately 2.7 PPW to 27 PPW. Results of discrete 2-norm convergence studies are presented in Figure 5.4. As in the case of a 1D beam, the errors are seen to converge at a rate of p with respect to the size of the grid, as predicted by the theory. The error in the solution with $p = 9$ and using $N = 95$ points per dimension at the final time $t_f = 3$ is also plotted in Figure 5.4. Consistent with observations from the beam, no error localization is apparent.
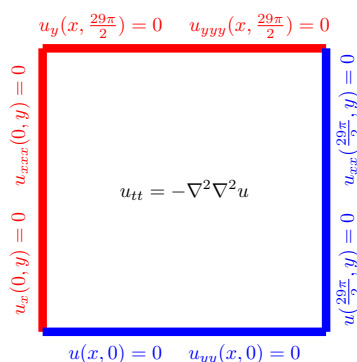

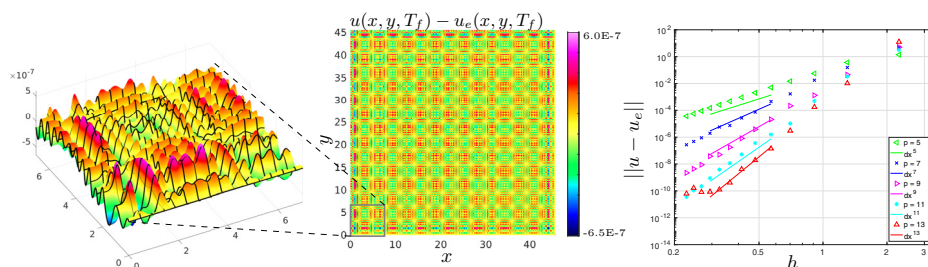FIG. 5.3. *Schematic of plate with simple supports on the bottom and right and sliding supports on the left and top.*

FIG. 5.4. *Left: Error in the solution at the final time $t_f = 3$ with $N = 95$, $p = 9$. A zoomed region is also shown with solid black lines indicating the cell boundaries. Right: Discrete 2-norm error versus grid size for the problem whose solution is given by* (5.6). *The the solid reference lines indicate pth-order convergence.*

5.3. Kirchhoff–Love plate in an annular geometry. Having discussed the theoretical and practical properties of the GD D-Spline approach for 2D plates in simple geometry, we move now to a brief illustration of the approach for simple mapped geometries. Consider a Kirchhoff–Love plate in a quarter annular geometry with the radius $r \in [.3, 1]$ and the polar angle $\theta \in [0, \pi/2]$, and for simplicity take the density $\rho = 1$ and flexural rigidity $D = 1$ to be constants. The strong-form equations are thus

$$(5.7) \qquad u_{tt} = -\Delta^2 u, \quad \mathbf{x} \in \mathcal{A},$$

where $\mathcal{A}$ indicates the quarter annular domain and Δ^2 is the biharmonic operator in Cartesian coordinates $\mathbf{x} = (x, y)$. The weak-form operator is derived in the usual way using multiplication by a test function v and taking the inner product over the domain. Given the annular geometry, it is convenient to represent both the differential operator and inner products in polar coordinates. We therefore introduce the notation

$$(u, v) \equiv \int \int uv \, dx \, dy,$$

$$< u, v > \equiv \int \int uv \, dr \, d\theta,$$

so that $(\cdot, \cdot)$ is the true inner product, and $(v, u) = < v, ru > = < rv, u >$. Given this notation, the weak-form representation of (5.7) is simply

$$< \phi, ru_{tt} > = - \left\langle (r\phi_r)_r, \frac{1}{r}(ru_r)_r \right\rangle - \left\langle (r\phi_r)_r, \frac{1}{r^2} u_{\theta\theta} \right\rangle$$

$$(5.8) \qquad - \left\langle \frac{1}{r}\phi_{\theta\theta}, \frac{1}{r}(ru_r)_r \right\rangle - \left\langle \frac{1}{r}\phi_{\theta\theta}, \frac{1}{r^2} u_{\theta\theta} \right\rangle + B.T., \quad (r, \theta) \in \mathcal{A},$$

where u is considered a function of r and θ, and $B.T.$ is a placeholder for boundary terms. Note that in this simple case the coordinate system is orthogonal and so (5.8) can be computed as the summation of 1D tensor product matrices, which is how it has been implemented.

To specify a numerical test, consider the quarter annulus with $r \in [0.3, 1]$, and $\theta \in [0, \pi/2]$, with initial conditions

$$(5.9a) \qquad u(r, \theta, 0) = \exp\left(-100\left((x - x_0)^2 + (y - y_0)^2\right)\right),$$

$$(5.9b) \qquad u_t(r, \theta, 0) = 0,$$

where $x = r\cos(\theta)$, $y = r\sin(\theta)$, and $x_0 = y_0 = 0.5$. For simplicity we use natural (free) boundary conditions with extrapolation boundary closures on all sides and run to a final time $t = 0.007$. Color contours of the solution and the time derivative at the final time, computed using $p = 5$ with 50 elements in each of the radial and angular dimensions, are shown in Figure 5.5.

Because the exact solution is not known in this case, we instead use the calculation with $p = 13$ as an approximate exact solution. This approximation is valid only if the error in the approximate exact solution is much smaller than the error being measured, and so we make the comparison only for $p = 5, 7, 9$. Color contours of the approximate errors are shown in Figure 5.6, which shows the error converging with respect to p. Of note, the error becomes dominated by boundary effects, as can be expected for the extrapolation closure.
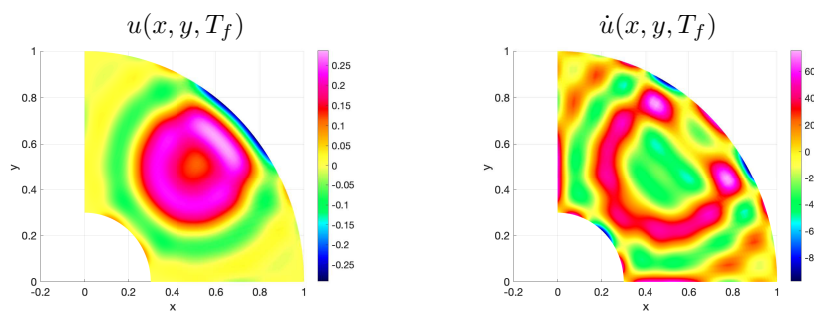
FIG. 5.5. *Computed solution at* $t = 0.007$, $p = 5$, *and* 50 *elements in each of the radial and angular dimensions.*
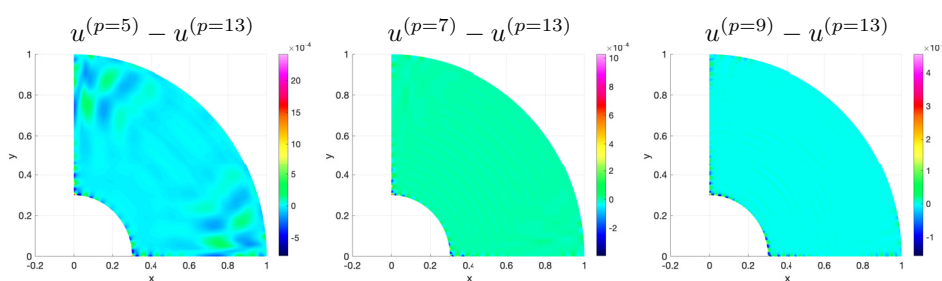

FIG. 5.6. *Color contours of approximate errors in u for* $p = 5$ *(left),* $p = 7$ *(center), and* $p = 9$ *(right). In all cases extrapolation closures are used. Note that the error becomes dominated by boundary effects for higher p.*

6. Conclusions. In this paper we have provided details of the construction of the Galerkin difference, or GD, D-Spline method with intended application to the fourth-order spatial operators found in beams and plates. The core of the method is a new approach to smooth data interpolation, the D-Spline interpolant. Theoretical convergence rates of $\mathcal{O}(h^p)$, where p is the width of support of the underlying basis functions and h is a grid spacing, are established for the D-Spline interpolant. Basis functions for the D-Spline interpolant are established and are subsequently applied in a weak-form Galerkin discretization of the beam and plate equations. We have also provided a new boundary closure strategy that yields very favorable spectral properties in comparison to existing approaches. The theoretically predicted convergence rates for the PDE approximation are then demonstrated for simple 1D and 2D test cases. An example extension to a quarter annular geometry is also given.

There are many avenues for extension of these ideas and future work. For example, D-Spline interpolation is a generic approach to produce a locally generated smooth interpolant from data. Therefore, D-Splines may find useful application in other fields where smooth data fitting is needed. An example of this is [4], where D-Splines with maximal smoothness are used to develop a convolution quadrature time-stepping scheme for time-domain integral equations. In the context of PDE discretization, the direct enforcement boundary closure could yield benefits for other GD discretizations, and its utility should be investigated. Similar ideas may also be fruitful in mitigating so-called outlier eigenvalues in isogeometric analysis. A detailed comparison of the GD D-Spline approach to other methods in terms of accuracy and efficiency would

also be of interest. Finally, the proposed GD D-Spline discretization for beams and plates should be investigated for physically motivated problems from science and engineering. This might include extension to more general geometric domains or application to multiphysics problems such as fluid-structure interaction.

REFERENCES

[1] G. A. BAKER, *Error estimates for finite element methods for second order hyperbolic equations*, SIAM J. Numer. Anal., 13 (1976), pp. 564–576.

[2] J. W. BANKS AND T. HAGSTROM, *On Galerkin difference methods*, J. Comput. Phys., 313 (2016), pp. 310–327.

[3] J. W. BANKS, T. HAGSTROM, AND J. JACANGELO, *Galerkin differences for acoustic and elastic wave equations in two space dimensions*, J. Comput. Phys., 372 (2018), pp. 864–892.

[4] A. BARNETT, L. GREENGARD, AND T. HAGSTROM, *High-order discretization of a stable time-domain integral equation for acoustic wave propagation*, J. Comput. Phys., to appear.

[5] G. D. BIRKHOFF, *General mean value and remainder theorems with applications to mechanical differentiation and quadrature*, Trans. Amer. Math. Soc., 7 (1906), pp. 107–136.

[6] S. C. BRENNER AND L. R. SCOTT, *The Mathematical Theory of Finite Element Methods*, Texts in Appl. Math. 15, Springer, New York, 2008.

[7] A. DUTT, L. GREENGARD, AND V. ROKHLIN, *Spectral deferred correction methods for ordinary differential equations*, BIT, 40 (2000), pp. 241–266.

[8] T. HAGSTROM AND R. ZHOU, *On the spectral deferred correction of splitting methods for initial value problems*, Commun. Appl. Math. Comput. Sci., 1 (2007), pp. 169–205.

[9] G. HÄMMERLIN AND K.-H. HOFFMANN, *Numerical Mathematics*, Springer, New York, 1991.

[10] T. HUGHES, J. A. COTTRELL, AND Y. BAZILEVS, *Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement*, Comput. Methods Appl. Mech. Engrg., 194 (2005), pp. 4135–4195.

[11] T. J. R. HUGHES, J. A. EVANS, AND A. REALI, *Finite element and NURBS approximations of eigenvalue, boundary-value, and initial-value problems*, Comput. Methods Appl. Mech. Engrg., 272 (2014), pp. 290–320.

[12] T. J. R. HUGHES, A. REALI, AND G. SANGALLI, *Duality and unified analysis of discrete approximations in structural dynamics and wave propagation: Comparison of p-method finite elements with k-method NURBS*, Comput. Methods Appl. Mech. Engrg., 197 (2008), pp. 4104–4124.

[13] T. R. HUGHES, *The Finite Element Method: Linear Static and Dynamic Finite Element Analysis*, Dover Publications, New York, 2000.

[14] J. KOZDON, L. WILCOX, T. HAGSTROM, AND J. BANKS, *Robust approaches to handling complex geometries with Galerkin difference methods*, J. Comput. Phys., 392 (2019), pp. 483–510.

[15] K. MATTSSON, *Diagonal-norm summation by parts operators for finite difference approximations of third and fourth derivatives*, J. Comput. Phys., 274 (2014), pp. 432–454.

[16] B. RIVIERE, *Discontinuous Galerkin Methods for Solving Elliptic and Parabolic Equations: Theory and Implementation*, Frontiers in Appl. Math. 35, SIAM, Philadelphia, PA, 2008.

[17] B. RIVIERE AND M. WHEELER, *Discontinuous finite element methods for acoustic and elastic wave problems*, Contemp. Math., 329 (2003), pp. 271–282.

[18] B. SJÖGREEN AND N. A. PETERSSON, *A fourth order finite difference scheme for the elastic wave equation in second order formulation*, J. Sci. Comput., 52 (2012), pp. 17–48.

[19] B. STRAND, *Summation by parts for finite difference approximations for d/dx*, J. Comput. Phys., 110 (1994), pp. 47–67.

[20] G. STRANG AND G. FIX, *An Analysis of the Finite Element Method*, 2nd ed., Wellesley-Cambridge, Wellesley, MA, 2008.

[21] M. SVÄRD AND J. NORDSTRÖM, *Review of summation-by-parts schemes for initial-boundary-value problems*, J. Comput. Phys., 268 (2014), pp. 17–38.

[22] G. SZEGÖ, *Orthogonal Polynomials*, AMS, Providence, RI, 1975.

[23] R. L. TAYLOR, F. C. FILIPPOU, A. SARITAS, AND F. AURICCHIO, *A mixed finite element method for beam and frame problems*, Comput. Mech., 31 (2003), pp. 192–203.

[24] Y. XU AND C.-W. SHU, *Local discontinuous Galerkin methods for high-order time-dependent partial differential equations*, Commun. Comput. Phys., 7 (2010), pp. 1–46.