A diagonally-implicit time integration scheme for space-time moving finite elements

Randolph E. Bank[1)]
Department of Mathematics, University of California, San Diego, La Jolla, California 92093-0112.
Email: rbank@ucsd.edu
Maximilian S. Metti
Adtile Technologies, 11750 Sorrento Valley Road, San Diego, California, 92121.
Email: maxx@adtile.me.

Abstract

In this paper, we analyze and provide numerical experiments for a moving finite element method applied to convection-dominated, time-dependent partial differential equations. We follow a method of lines approach and utilize an underlying tensor-product finite element space that permits the mesh to evolve continuously in time and undergo discontinuous reconfigurations at discrete time steps. We employ the TR-BDF2 method as the time integrator for piecewise quadratic tensor-product spaces, and provide an almost symmetric error estimate for the procedure. Our numerical results validate the efficacy of these moving finite elements.

Mathematics subject classification: 65M55, 65F10
Key words: TR-BDF2, Moving Finite Elements, Method of Characteristics, Convection-Dominated, Moving Mesh Methods, Error Analysis

1. Introduction

The method of lines is an efficient approach for computing numerical solutions to parabolic partial differential equations by converting these problems into systems of ordinary differential equations. This provides a great deal of flexibility in how the solution may be computed, as the time discretization then becomes independent of the spatial discretization. For finite element methods, the spatial dimensions are discretized in the usual way, leading to a semi-discrete problem that is subsequently propagated in time by numerical integration.

When dealing with convection-dominated problems, the spatial discretization can be chosen to evolve continuously in time, which allows the finite element mesh to continuously track moving structures in the solution such as steep sweeping fronts [1, 2, 3]. These moving finite elements can lead to remarkably improved stability in computing a solution, with respect to the length of permissible time steps [4, 5]. While the literature of moving mesh finite element methods is expanding rapidly, rigorous error analysis of these methods is still relatively unknown.

In [6, 7], tensor-product finite element spaces compatible with a method of lines discretization were introduced that allowed these moving finite element solutions to be studied in a space-time finite element framework. As a result, these papers established symmetric error estimates for these finite element solutions of arbitrary order when the numerical time integrator

[1)] The work of this author was supported by the National Science Foundation under contract DMS-1318480.

belongs to a particular class of fully implicit collocation methods. The first symmetric error estimate is proven in [8] for semi-discrete moving finite elements,

$$|||u - u_h||| \leq C \inf_{\chi \in \mathcal{V}_h} |||u - \chi|||, \tag{1.1}$$

using a mesh-dependent energy semi-norm, $|||\cdot|||$, where u is the true solution, u_h is the finite element solution, and $\mathcal{V}_h$ is a (tensor-product) moving finite element space [6, 7]. Other error analyses include [9] that bound the error in alternative energy norms, [10] for mixed finite element methods, and [11] for moving finite difference schemes.

In this paper, we consider the effects of employing a time integrator that does not belong to previous classes of collocation methods. This is a valuable modification because the collocation methods in past works are fully implicit and couple all intermediate stages of each time step, significantly increasing the computational complexity when using higher order quadrature. We consider the second-order and diagonally-implicit time integrator, TR-BDF2, introduced in [12, 13], and using piecewise quadratic tensor-product finite element spaces to discretize the problem. This time integration scheme is known for its favorable stability properties [14, 15], which motivates its study in the present context as moving finite element discretizations often lead to stiff systems of ODEs [4, 5]. In Section 3, we prove an error estimate like (1.1) with an additional term corresponding to the truncation error of TR-BDF2.

This work largely builds on the analyses in [6, 16], where parts of the preliminary analysis are given in more detail. This paper is organized as follows: in Section 2, we describe the model equation, the piecewise quadratic tensor-product finite element space, and some preliminary results. In Section 3, a space-time moving finite element method using TR-BDF2 time integration is proposed and an error estimate for the finite element solution is proven. We note that the proposed scheme is a simple discretization resulting directly from a finite element discretization in space and the method of lines to discretize the time variable; many moving mesh methods fit into the framework of our analysis without modification, as discussed in section 3. Section 4 describes and reports some numerical experiments that validate the efficacy of these moving finite element methods.

2. Preliminary results

The model problem used in this error analysis is the linear convection-diffusion-reaction equation. The spatial domain, Ω, is assumed to be a simply connected set in $\mathbb{R}^d$, where $d = 1, 2$, or 3, with boundary $\partial\Omega$. The time domain is a finite interval, $(0, T]$, and the space-time domain is given by $\mathcal{F} \equiv \Omega \times (0, T]$.

Let a, b, c, and f be smooth and bounded functions defined on $\mathcal{F}$ such that there exist constants $\bar{a} > 0$ and $\bar{c} \geq 0$ with $a \geq \bar{a}$ and $c \geq \bar{c}$ on Ω, and let g be integrable on $\partial\Omega$. Let u_0 be a given initial condition for the solution on $\mathcal{F}$ and let n denote the outward unit normal vector to the boundary $\partial\Omega$. The solution to the differential equation, denoted by u, is the function that satisfies

$$u_t - \nabla \cdot (a\nabla u) + b \cdot \nabla u + cu = f, \qquad \text{in } \mathcal{F}, \tag{2.1}$$

$$a\nabla u \cdot n = g, \qquad \text{on } \partial\Omega \times (0, T], \tag{2.2}$$

$$u(x, 0) = u_0(x), \qquad \text{for } x \text{ in } \Omega.$$

When the convection term, b in (2.1), is large relative to the other coefficients, steep shock layers may develop in the solution that propagate through the spatial domain. Basic finite

element discretizations consequently require short time steps to maintain accuracy of the computed solution at this moving front, or a moving mesh can be employed, allowing for more flexibility in the length of the time step. Moving finite elements offset the convection velocity in a similar manner to the method of characteristics.

We assume a time-dependent parametrization of the spatial domain, $x(t)$, where $x(\cdot,t) : \Omega \to \Omega$ is a continuous bijective map for fixed t in $(0,T]$ and $x_t(\cdot,t)$ is continuous on Ω. We also require $x_t(y,\cdot)$ to be piecewise continuous on $(0,T]$ for each $y \in \Omega$. We define the space-time derivative as

$$u_\tau(x(t),t) \equiv \frac{d}{dt}u(x(t),t) = u_t(x(t),t) + x_t \cdot \nabla_x u(x(t),t).$$

We refer to this as the *characteristic derivative* of u.

The weak form of the problem is: find u with $u(\cdot,t) \in \mathcal{H}^1(\Omega)$ and $u_t(\cdot,t) \in \mathcal{L}_2(\Omega)$ such that for all χ in $\mathcal{H}^1(\Omega)$ and $0 < t \le T$,

$$\big(u_\tau(\cdot,t),\chi\big) + \mathcal{A}_\tau(t;u,\chi) = \big(f(\cdot,t),\chi\big) + \big\langle g(\cdot,t),\chi\big\rangle, \tag{2.3}$$

and when $t = 0$

$$\big(u(\cdot,0),\chi\big) = \big(u_0,\chi\big).$$

The inner-products are given by

$$(f,\chi) = \int_\Omega f(x)\chi(x)\ dx \qquad \text{and} \qquad \langle g,\chi\rangle = \int_{\partial\Omega} g(s)\chi(s)\ ds,$$

and define the time-dependent bilinear form

$$\mathcal{A}_\tau(t;u,\chi) \equiv \int_\Omega a(x,t)\nabla u(x,t)\cdot\nabla\chi(x) + (b(x,t) - x_t(t))\cdot\nabla u(x,t)\ \chi(x) + c(x,t)u(x,t)\chi(x)\ dx.$$

Notice that parameterizing the spatial variable so that $x_t \approx b$ leads to a formulation where the convection velocity is much less prominent, as it is "absorbed" into the characteristic derivative.

The finite element space we use to discretize the differential equation is a tensor-product of a discontinuous piecewise quadratic finite elements in time with continuous piecewise quadratic finite elements in space.

Let $0 < t_1 < \ldots < t_m = T$ form a strict ordered partition of the time domain and define $\Delta t_i \equiv t_i - t_{i-1}$. For $1 \le i \le m$, let $\{x_k(t)\}_k$ represent the vertices of a triangulation of the domain at time t, where $0 \le k \le N_i$, and we assume that $|x_k(t) - x_j(t)| > \Delta x$ throughout each time partition for some minimum mesh size $\Delta x > 0$ and $j \ne k$. The vertices of the mesh are permitted to move along quadratic trajectories throughout each time partition—that is, each node $x_k(t)$ is a quadratic polynomial for $t_{i-1} < t \le t_i$, though discontinuous reconfigurations of the mesh are permitted at the beginning of each partition. These discontinuous changes in the mesh provide flexibility for discretely adding and removing degrees of freedom, as well as keeping the nodes in the mesh from colliding and tangling [17].

The reference element for this finite element space is the Cartesian product of the unit interval ($d = 1$), triangle ($d = 2$), or tetrahedron ($d = 3$) for space with the unit interval reference element for the time domain. Let e be an element in the mesh with vertices given by

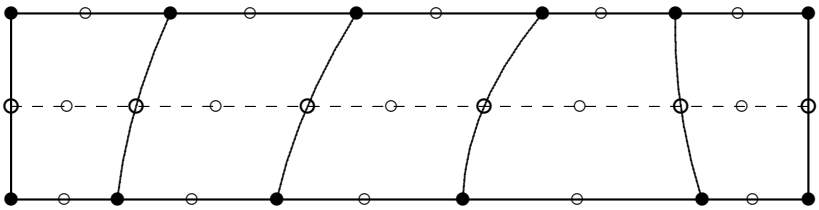

Fig. 2.1. A space-time mesh partition of a single dimension in space. The filled circles represent the space-time "hat" basis nodes; hollow circles correspond to "bump" function nodes.

$x_k(t)$ at time t, where $k = 0, 1, \ldots, d$ and $t_{i-1} \le t \le t_i$. Then, the isoparametric mapping from the reference element to e is given by

$$t = t_{i-1} + \hat{t}\Delta t_i,$$
$$x(t) = \mathcal{J}_e(t)\hat{x} + x_0(t),$$

for $0 \le \hat{t} \le 1$ and $\hat{x}$ in the spatial reference element, where $x(t)$ is affine in space and quadratic in time. Notice that the time variable is space invariant, but the spatial variable does in fact depend on t. The $d \times d$ spatial Jacobian matrix, $\mathcal{J}_e(t)$, determines the shape and size of the element e at time slice t. Since the determinant of the spatial Jacobian is proportional to the size of the element, we require $|\det \mathcal{J}_e(t)| > 0$ for all t to ensure a non-degenerating mesh.

Since we are using a tensor-product space-time discretization, the degrees of freedom of the finite element space are distributed in time slices. Namely, for fixed time t, the degrees of freedom $\{x_k(t)\}_k$ define a standard finite element space of continuous piecewise-quadratic polynomials on Ω, which we denote by $\mathcal{V}_h(t)$. This is an important property that permits us to discretize the differential equation using the method of lines. We emphasize that the finite element functions are piecewise quadratic polynomials along the node trajectories, $x_k(t)$, rather than in the time direction as with standard tensor-product discretizations. This ensures that the *characteristic* derivative of a finite element function, ϕ, is continuous within each time partition and satisfies $\phi_\tau(t) \in \mathcal{V}_h(t)$. We denote the tensor-product finite element space on the space-time domain by $\mathcal{V}_h$. More detailed descriptions of these tensor product finite element spaces can be found in [6, 16].

Much of the literature regarding moving mesh methods focus on finding a suitable mesh motion for a given problem. This paper intends to provide an error analysis and does not seek to determine the "best" scheme for moving the mesh, though we discuss the relevance of our error analysis to some well-studied techniques for finding the mesh motion, x_t, in the next section.

For our moving finite element method, it is convenient to define the finite element functions at the mesh discontinuity by an $\mathcal{L}_2$-projection. For ϕ in $\mathcal{V}_h$, we represent the limiting values near the discontinuities as $\phi(t_{i^+}) \equiv \lim_{\delta \to 0^+} \phi(t + \delta)$ and $\phi(t_{i^-}) \equiv \lim_{\delta \to 0^+} \phi(t - \delta)$. We require

$$(\phi(t_{i^+}), \chi) = (\phi(t_{i^-}), \chi)$$

for all χ in $\mathcal{V}_h(t_{i^+})$, $i = 1, \ldots, m$. To uniquely determine the finite element functions, we take $\phi(t_i) \equiv \phi(t_{i^-})$ at the discontinuities.

Multi-index notation is used to represent spatial derivatives, but time and characteristic derivatives does not follow this convention. The $\mathcal{H}^k(\Omega)$ semi-norm and norm follow conventional

notation and we write

$$|v|_k = \left(\sum_{|\alpha|=k} (D_\alpha v, D_\alpha v) \right)^{1/2} \quad \text{and} \quad \|v\|_k = \left(\sum_{|\alpha|\leq k} (D_\alpha v, D_\alpha v) \right)^{1/2}.$$

Following Dupont [8], a mesh-dependent semi-norm is defined that allows us to prove our error estimate,

$$\|v\|_{(-1,\mathcal{V}_h(t))} = \sup_{\substack{\chi\in\mathcal{V}_h(t) \\ \chi\neq 0}} \frac{|(v,\chi)|}{\|\chi\|_1}.$$

We also use the infinity norm, $\|v\|_\infty = \max_{x\in\Omega} |v(x)|$.

We now introduce a space-time shape regularity constraint for the moving finite element mesh that controls the time evolution of the spatial elements and prevents degenerate elements. Fix e to be an element in the time partition with $t_{i-1} \leq t \leq t_i$. Then, the Jacobian matrix at time t can be represented as

$$\mathcal{J}_e(t) = \big(\mathcal{R}_e(t) + \Delta t_i \mathcal{H}_e(t)\big)\mathcal{J}_e(t_{i-1^+}), \tag{2.4}$$

for some orthogonal *rotation* matrix, $\mathcal{R}_e(t)$, and *evolution* matrix, $\mathcal{H}_e(t)$. The matrix $\mathcal{R}_e + \Delta t_i \mathcal{H}_e$ is constrained to have quadratic polynomial entries throughout the time partition. (We remark that this constraint on the node trajectories does not enter our error analysis directly, though the use of discrete time integration implicitly approximates these trajectories as quadratic polynomials; making this constraint explicit disallows any unexpected errors arising from poor shape regularity of the moving mesh.) The matrix $\mathcal{R}_e(t)$ describes the element rotation in time, and the evolution matrix describes the deformation of the shape of the element. Since the trajectories of the spatial nodes are restricted to quadratic polynomial paths, elements cannot rotate perfectly in time and more of a twisting action is observed; the evolution matrix necessarily reflects these deformations. If an element is merely translated in time, without rotation or changing shape, then the Jacobian matrix, $\mathcal{J}_e(t)$, remains unchanged.

Let $\rho(\cdot)$ represent the spectral norm for $d \times d$ matrices. It is assumed that the evolution matrix, $\mathcal{H}_e$, has a uniformly bounded spectral radius throughout the time step; namely, there exists some positive constant μ that does not depend on e or t such that

$$\rho\big(\mathcal{H}_e(t)\big) \leq \mu. \tag{2.5}$$

This bounds the relative change in shape and size of the element over time. Many schemes for moving the mesh implicitly satisfy (2.5), for example the algorithms in [18, 1, 19, 4, 5]; it is noted that such a bound tends to permits larger time steps in [5]. Assuming a non-degenerate finite element space and the space-time shape regularity bound (2.5), if we have $\Delta t_i \leq 1/2\tilde{c}_{\mu,d}$, with $\tilde{c}_{\mu,d} = [(1+\mu\Delta t_i)^d - 1]/\Delta t_i = \mu d + \mathcal{O}(\Delta t_i)$, it follows that

$$\rho\big(\mathcal{J}_e(t)\mathcal{J}_e^{-1}(t_{i-1^+})\big) = \rho\big(\mathcal{R}_e(t) + \Delta t_i \mathcal{H}_e(t)\big) \leq 1 + \mu\Delta t_i \tag{2.6}$$

and

$$1 - \tilde{c}_{\mu,d}\Delta t_i \leq \frac{|\det \mathcal{J}_e(t)|}{|\det \mathcal{J}_e(t_{i-1^+})|} \leq 1 + \tilde{c}_{\mu,d}\Delta t_i. \tag{2.7}$$

Let ϕ be a function in the finite element space, $\mathcal{V}_h(t)$, for some fixed t in the time partition $(t_{i-1}, t_i]$. We *shift* ϕ onto the mesh of $\mathcal{V}_h(t_{i-1^+})$, at the beginning of the time partition by

replacing the basis functions of $\mathcal{V}_h(t)$ with their corresponding basis functions in $\mathcal{V}_h(t_{i-1^+})$, while preserving the basis coefficients. Formally, this operation can be defined by an element-wise composition of the inverse of the affine spatial isoparametric maps for the elements in the mesh at time t, with the affine spatial isoparametric maps for the elements at the beginning of the time step. The following lemma, proven in [16], establishes the relationship between the space-time shape regularity constraint (2.5) and the continuity of this shift operation.

Lemma 2.1 (Shift lemma) *Let $\phi, \chi \in \mathcal{V}_h(t)$ and $\tilde{\phi}, \tilde{\chi} \in \mathcal{V}_h(t_{i-1^+})$ represent a pair of finite element functions and their shifts, respectively, on a non-degenerate time partition of the mesh that satisfies (2.5) on each element. If $\Delta t_i \leq 1/2\tilde{c}_{\mu,d}$, as defined in (2.7), then there exists a positive constant, $C_{\mu,d}$, such that*

$$\left| \|\phi\|_0^2 - \|\tilde{\phi}\|_0^2 \right| \leq C_{\mu,d} \Delta t_i \|\tilde{\phi}\|_0^2, \tag{2.8}$$

$$\left| \|\phi\|_1^2 - \|\tilde{\phi}\|_1^2 \right| \leq C_{\mu,d} \Delta t_i \|\tilde{\phi}\|_1^2. \tag{2.9}$$

We now present a local Grönwall lemma that is used to bound the maximum error of the finite element solution over each time partition. The proof for this lemma is given in [16].

Lemma 2.2 (Local Grönwall inequality) *Suppose there are two distinct times on each time partition, $t_{i-1} < t_{i,1} < t_{i,2} = t_i$, when the mesh satisfies the regularity constraint (2.5) and that there exists a positive constant κ such that*

$$\|b - x_t\|_\infty \leq \kappa. \tag{2.10}$$

If $\Delta t_i \leq 1/2\tilde{c}_{\mu,d}$ as defined in (2.7), and functions $\phi \in \mathcal{V}_h$ and $\eta \in \mathcal{H}^1(\Omega)$, with $\eta_\tau \in \mathcal{L}_2(\Omega)$, satisfy

$$\big(\phi_\tau(t_{i,j}), \chi\big) + \mathcal{A}_\tau\big(\phi(t_{i,j}), \chi\big) = \big(\eta_\tau(t_{i,j}), \chi\big) + \mathcal{A}_\tau\big(\eta(t_{i,j}), \chi\big) \tag{2.11}$$

for all χ in $\mathcal{V}_h(t_{i,j})$ at time each collocation node $j = 1, 2$, then, there exists a constant such that

$$\max_{1\leq j\leq 2} \|\phi(t_{i,j})\|_0^2 \leq C \left\{ \|\phi(t_{i-1^+})\|_0^2 + \sum_{j=1}^{2} \Delta t_i \left(\|\eta_\tau(t_{i,j})\|^2_{(-1,\mathcal{V}_h(t_{i,j}))} + \|\eta(t_{i,j})\|_1^2 + \|\phi(t_{i,j})\|_1^2 \right) \right\},$$

where C depends on κ, μ, d, p, and the differential equation.

Another discrete Grönwall lemma is used to aggregate the spatial error bounds from each time partition over the entire time domain. This result is proved in [20].

Lemma 2.3 (Discrete Grönwall inequality) *Let $\Delta t_i > 0$ and $\alpha_i, \gamma_i, \theta_i, q_i \geq 0$, for $1 \leq i \leq m$, with $\theta_i \Delta t_i \leq \frac{1}{2}$ and $\theta = \max_i \theta_i$. Then, if*

$$\frac{q_i - q_{i-1}}{\Delta t_i} + \gamma_i \leq \alpha_i + \theta_i(q_i + q_{i-1}),$$

there exists a positive constant, $C_\theta \approx e^{\theta T}$, such that

$$\max_{1\leq i\leq m} q_i + \sum_{i=1}^{m} \gamma_i \Delta t_i \leq C_\theta \left\{ q_0 + \sum_{i=1}^{m} \alpha_i \Delta t_i \right\}.$$

3. A space-time moving finite element method with TR-BDF2

To achieve second-order accuracy for the finite element solution, we employ the two-stage, diagonally-implicit time integration scheme, TR-BDF2. Since this scheme is diagonally implicit, significant savings are realized when applied to parabolic differential equations. TR-BDF2 was proposed by Bank, et al. in [12, 13] and has been analyzed in several other papers for its efficiency and stability [14, 15]. TR-BDF2 refers to a family of time stepping methods that is parametrized by the location of an intermediate collocation node. For this method, we define the collocation nodes on the reference interval to be

$$\hat{t}_0 = 0, \quad \hat{t}_1 = \varepsilon/2, \quad \text{and} \quad \hat{t}_2 = 1,$$

where $0 < \varepsilon < 1$ is a free parameter and this fixes the basis nodes on the reference interval to be

$$\hat{\zeta}_0 = 0, \quad \hat{\zeta}_1 = \varepsilon, \quad \text{and} \quad \hat{\zeta}_2 = 1,$$

so that $\hat{t}_1$ is the midpoint of the first two basis nodes. The Runge-Kutta coefficients correspond to integrating the computed solution a step of length $\varepsilon\Delta t_i$ by the trapezoid rule, then completing the time step by a second-order backward difference. Let

$$\zeta_{i,j} \equiv t_{i-1} + \hat{\zeta}_j \Delta t_i, \quad x_{i,j} \equiv x(\zeta_{i,j}), \quad \text{and} \quad t_{i,k} \equiv t_{i-1} + \hat{t}_k \Delta t_i$$

for $j = 0, 1, 2$ and $k = 1, 2$. For any function u defined on the domain $\Omega \times (0, T]$, define

$$\begin{aligned} u_{TR}\big(x(t_{i,1}), t_{i,1}\big) &\equiv \frac{1}{2} u\big(x_{i,1}, \zeta_{i,1}\big) + \frac{1}{2} u\big(x_{i,0}, \zeta_{i,0}\big), \\ u\big(x(t_{i,2}), t_{i,2}\big) &\equiv u\big(x_{i,2}, \zeta_{i,2}\big), \end{aligned}$$

where the values of u are sampled along the characteristic trajectories of the finite element mesh. The coefficients for the time derivative are determined by the interpolating quadratic Lagrange polynomials along the characteristic trajectories and computing the derivative:

$$\begin{aligned} \partial_\tau u\big(x(t_{i,1}), t_{i,1}\big) &= \frac{u\big(x_{i,1}, \zeta_{i,1}\big) - u\big(x_{i,0}, \zeta_{i,0}\big)}{\varepsilon \Delta t_i}, \\ \partial_\tau u\big(x(t_{i,2}), t_{i,2}\big) &= \frac{\varepsilon(2-\varepsilon) u\big(x_{i,2}, \zeta_{i,2}\big) - u\big(x_{i,1}, \zeta_{i,1}\big) + (1-\varepsilon)^2 u\big(x_{i,0}, \zeta_{i,0}\big)}{\varepsilon(1-\varepsilon)\Delta t_i}. \end{aligned}$$

The fact that $u_{TR}(t_{i,1})$ and $\partial_\tau u(t_{i,1})$ do not depend on $u(t_{i,2})$ allows the solver to propagate the solution one basis value at a time, which is why the TR-BDF2 scheme is so efficient, especially for higher dimensional problems.

The optimal choice for the free parameter is known to be $\varepsilon = 2 - \sqrt{2}$, as it minimizes the local truncation error [21]. The TR-BDF2 scheme is A-stable and L-stable [12, 13, 15], which is critical property for this application as moving mesh methods typically lead to stiff systems of equations [4, 5].

In this section, we present the fully discretized PDE corresponding to (2.3), where moving finite elements are used to discretize the problem and TR-BDF2 is used to integrate in time. The solution to the fully discrete problem is characterized as the finite element function $u_h \in \mathcal{V}_h$ that satisfies

$$\big(\partial_\tau u_h(t_{i,1}), \chi_1\big) + \mathcal{A}_\tau\big(u_{h,TR}(t_{i,1}), \chi_1\big) = \big(f(t_{i,1}), \chi_1\big) + \big\langle g(t_{i,1}), \chi_1 \big\rangle \tag{3.1}$$

for all χ_1 in $\mathcal{V}_h(t_{i,1})$ and

$$\big(\partial_\tau u_h(t_{i,2}), \chi_2\big) + \mathcal{A}_\tau\big(u_h(t_{i,2}), \chi_2\big) = \big(f(t_{i,2}), \chi_2\big) + \big\langle g(t_{i,2}), \chi_2\big\rangle, \tag{3.2}$$

for all χ_2 in $\mathcal{V}_h(t_{i,2})$ with $i = 1, \ldots, m$.

The constraint at the mesh discontinuity,

$$\big(u_h(t_{i^+}), \chi\big) = \big(u_h(t_{i^-}), \chi\big), \tag{3.3}$$

must also hold for all χ in $\mathcal{V}_h(t_{i^+})$, $i = 1, \ldots, m-1$, to ensure that $u_h \in \mathcal{V}_h$. As a matter of practical interest, we note that this requirement is stricter than necessary and that we only require

$$\|u_h(t_{i^+})\|_0 \leq \|u_h(t_{i^-})\|_0 \tag{3.4}$$

for our error analysis. This can also be accomplished by *conservative* interpolation schemes given in [22, 23], which are more efficient and straightforward to implement compared to traditional $\mathcal{L}_2$-projection—especially in higher dimensions. Another alternative is to use standard, higher order interpolation for improved stability, though the bound (3.4) cannot be guaranteed in general.

In contrast to the weak formulation (2.3), the finite element formulation only requires the finite element solution to satisfy the constraints (3.1)–(3.2) at the collocation nodes. It is important to recognize that (3.1)–(3.2) are equivalent to imposing a Galerkin orthogonality of the finite element solution at the collocation nodes; this requirement is satisfied (with appropriate changes to reflect various choices of time integration) by many popular moving mesh finite element methods applied to linear problems including their original formulation by Miller and Miller [4, 5], moving mesh methods using harmonic maps or equidistribution principles [18, 24, 19, 25], and some Arbitrary Lagrangian-Eulerian methods [26, 27, 28]. Unfortunately, gradient-weighted schemes [2, 3] do not fall into this framework since they employ a weight function that has a nonlinear dependence on the computed solution, nor do some methods based on conservation principles [29, 30].

Furthermore, the moving mesh finite element method considered here is general enough to include adaptive finite elements [26, 27, 28]. By permitting discontinuities at the discrete time steps, this analysis covers more specific mesh motion schemes, such as Euler-Langrange schemes where $x_t = b$ is a critical component of the method. One small difference in this analysis is that any interpolation error in the mesh motion, $x_t - b$, is not set to zero when imposing the Galerkin orthogonalities (3.1)–(3.2); in most cases, this error is quite small and leads only to minor differences in the computed solution. Moreover, this analysis covers methods where a posteriori error estimates are prescribed to define how the mesh should change at each time step. If only refinement and coarsening occurs at the discrete time-steps and interpolation is used, then (3.4) should hold as well, unless coarsening the mesh where the solution may undergo large changes in value, which is not typically the case in refinement methods based on a posteriori error estimates.

We now prove an error estimate for u_h satisfying (3.1)–(3.3). The proof follows that of Theorem 4.3 in [6] with some additional arguments that bound the error introduced by the trapezoid approximation at the mid-step of each time partition. Due to the departure of this method from a strict finite element framework, the symmetry of the error bound is broken and an additional term proportional to the error of the trapezoid approximation is introduced. Furthermore, for simplicity of the presented result, it is assumed that the diffusion coefficient,

a, is a positive constant; an analogous theorem holds for general diffusivity and is presented in full detail in [16].

One final aspect in which the error bound changes for TR-BDF2 integration is an increased sensitivity to discontinuous changes in the mesh. Bank and Yserentant [31] proved the $\mathcal{H}^1$-stability of $\mathcal{L}_2$-projections onto finite element spaces with potentially nonuniform meshes. Using this result, we assume $\|\chi(t_{i^+})\|_1 \leq C_{\mathcal{H}}\|\chi(t_{i^-})\|_1$, for χ in $\mathcal{V}_h$. As can be seen from [31], the bounding constant, $C_{\mathcal{H}}$, is smaller when the mesh reconfiguration is more subtle at the mesh discontinuities. This intuitively makes sense, since $\chi(t_{i^+}) \approx \chi(t_{i^-})$ in such cases.

The norm in which the error is bounded employs the trapezoid approximation at the mid-step collocation node of each time partition. The semi-norm in which we bound the error of the finite element solution with TR-BDF2 time integration is given by

$$|\!|\!| u |\!|\!|^2 \equiv \max_{\substack{1\leq i\leq m\\ 1\leq j\leq 2}} \|u(t_{i,j})\|_0^2 + \sum_{i=1}^{m} \Delta t_i \Big[\|\partial_\tau u(t_{i,1})\|^2_{(-1,\mathcal{V}_h(t_{i,1}))} + \|u_{TR}(t_{i,1})\|_1^2 + \|u(\zeta_{i,1})\|_1^2 + \|\partial_\tau u(t_{i^-})\|^2_{(-1,\mathcal{V}_h(t_{i^-}))} + \|u(t_{i^-})\|_1^2 \Big].$$

Theorem 3.1. *Suppose that $\mathcal{V}_h$ is a finite element space with a non-degenerate mesh and let $\varepsilon = 2/3$. Furthermore, assume that the diffusion term, a, is a positive constant, that there exist positive constants μ and κ such that at each collocation node*

$$\rho(\mathcal{H}_e) \leq \mu, \tag{3.5}$$

$$\|b - x_t\|_\infty \leq \kappa, \tag{3.6}$$

and that the mesh discontinuities are controlled for the spatial meshes and the length of the time steps are graded so that $\|\chi(t_{i^+})\|_1 \leq C_{\mathcal{H}}\|\chi(t_{i^-})\|_1$ for all $\chi \in \mathcal{V}_h$ with

$$C_{\mathcal{H}}\Delta t_i \leq 8\Delta t_{i-1}. \tag{3.7}$$

Then, if $\Delta t = \max_{1\leq i\leq m} \Delta t_i$ is sufficiently small, there exists a positive constant C such that the finite element solution satisfies

$$|\!|\!| u - u_h |\!|\!|^2 \leq C\Big\{ \inf_{\chi\in\mathcal{V}_h} |\!|\!| u - \chi |\!|\!|^2 + \int_0^T \|\Delta t^2 u_{\tau\tau}(t)\|_1^2 \, dt \Big\}, \tag{3.8}$$

where C depends on μ, κ, d, p, and the differential equation.

We emphasize that the diffusivity is assumed constant to simplify the proof and that the theorem extends to more general linear parabolic equations in which $\|a_\tau\|_\infty$ is bounded (see [16]). Furthermore, the proof is restricted to the case where we use the collocation nodes determined by Gauss-Radau quadrature, characterized by $\varepsilon = 2/3$, as Gauss-Radau quadrature has a positive truncation error, which helps bound the aggregation of the local truncation errors. Furthermore, the intermediate time basis node for Gauss-Radau is 2/3, which is close to the optimal value $\varepsilon = 2 - \sqrt{2} \approx 0.5858$ for TR-BDF2.

Proof. For this proof, we use the discrete Galerkin orthogonalities

$$(u_\tau - u_{h,\tau}, \chi) + \mathcal{A}_\tau(u - u_{h,RK}, \chi) = 0,$$

at $t = t_{i,1}, t_{i,2}$ for any finite element function, $\chi \in \mathcal{V}_h$, where $u_{h,RK}(t_{i,1}) = u_{h,TR}(t_{i,1})$ and $u_{h,RK}(t_{i,2}) = u_h(t_{i^-})$.

Following Dupont [8], let ψ in $\mathcal{V}_h$ be an arbitrary function and define $\phi \equiv u_h - \psi$ in $\mathcal{V}_h$ and $\eta \equiv u - \psi$. Then, at the time collocation node

$$(\phi_\tau, \chi) + \mathcal{A}_\tau(\phi_{RK}, \chi) = (\eta_\tau, \chi) + \mathcal{A}_\tau(\eta_{RK}, \chi) + \mathcal{A}_\tau(u - u_{RK}, \chi), \tag{3.9}$$

where ϕ_{RK}, η_{RK} are defined similarly to $u_{h,RK}$.

At $t = t_{i,1}$, $t_{i,2}$, choose $\chi = \phi_{RK}$ to get

$$(\phi_\tau, \phi_{RK}) = \frac{1}{2}\partial_\tau \|\phi\|_0^2 + (\phi_\tau, \phi_{RK} - \phi), \tag{3.10}$$

$$\mathcal{A}_\tau(\phi, \phi_{RK}) \geq a(1-\epsilon)\|\phi_{RK}\|_1^2 - C\|\phi_{RK}\|_0^2, \tag{3.11}$$

$$(\eta_\tau, \phi_{RK}) \leq C\|\eta_\tau\|^2_{(-1,\mathcal{V}_h(t))} + a\epsilon\|\phi_{RK}\|_1^2, \tag{3.12}$$

$$\mathcal{A}_\tau(\eta_{RK}, \phi_{RK}) \leq C\|\eta_{RK}\|_1^2 + a\epsilon\|\phi_{RK}\|_1^2, \tag{3.13}$$

$$\mathcal{A}_\tau(u - u_{RK}, \phi_{RK}) \leq C\|u - u_{RK}\|_1^2 + a\epsilon\|\phi_{RK}\|_1^2, \tag{3.14}$$

where ϵ is used to represent arbitrarily small positive constants and its value may differ from bound to bound for notational convenience; similarly, the value of C may also change throughout the proof. Combining (3.9)–(3.14) gives the bound

$$\begin{aligned}\frac{1}{2}\partial_\tau \|\phi\|_0^2 + a(1-\epsilon)\|\phi_{RK}\|_1^2 + (\phi_\tau, \phi_{RK} - \phi) \\ \leq C\Big\{\|\eta_\tau\|^2_{(-1,\mathcal{V}_h(t))} + \|\eta_{RK}\|_1^2 + \|u - u_{RK}\|_1^2 + \|\phi_{RK}\|_0^2\Big\}.\end{aligned} \tag{3.15}$$

This provides a bound at each time collocation node, which now needs to be extended to a telescoping sum on each time-step so that the Discrete Grönwall lemma can be applied.

Using the error estimate for Gauss-Radau collocation nodes, it holds that

$$\frac{3}{4}f(1/3) + \frac{1}{4}f(1) = \int_0^1 f(t)\,dt + \frac{1}{6^3}f'''(\zeta),$$

for some ζ in $[0,1]$ and any bounded function with $\left|\frac{d^k}{dt^k}f(t)\right| < \infty$ on $[0,1]$, for $k \leq 3$. For the characteristic derivative terms in (3.15), applying the quadrature rule over the time partition gives

$$\begin{aligned}\frac{3\Delta t_i}{4}\partial_\tau\|\phi(t_{i,1})\|_0^2 + \frac{\Delta t_i}{4}\partial_\tau\|\phi(t_{i,2})\|_0^2 &= \|\phi(t_{i^-})\|_0^2 - \|\phi(t_{i-1^+})\|_0^2 + \frac{\Delta t_i^4}{6^3}\partial_\tau^4\|\phi(t_{i-1} + \zeta\Delta t_i)\|^2 \\ &= \|\phi(t_{i^-})\|_0^2 - \|\phi(t_{i-1^+})\|_0^2 + \frac{1}{36}\|\Delta t_i^2\phi_{\tau\tau}\|_0^2,\end{aligned}$$

where the last equality comes from $\partial_\tau^3\phi \equiv 0$, so that $\Delta t_i^4\partial_\tau^4\|\phi\|_0^2 = 6\|\Delta t_i^2\phi_{i,\tau\tau}\|_0^2$. Apply the quadrature rule to the characteristic derivative and use (3.4) to show

$$\frac{3\Delta t_i}{4}\partial_\tau\|\phi(t_{i,1})\|_0^2 + \frac{\Delta t_i}{4}\partial_\tau\|\phi(t_{i,2})\|_0^2 \geq \|\phi(t_{i^-})\|_0^2 - \|\phi(t_{i-1^-})\|_0^2 + \frac{1}{36}\|\Delta t_i^2\phi_{i,\tau\tau}\|_0^2. \tag{3.16}$$

Combining the bound from the inequality and (3.15) yields

$$\begin{aligned}\|\phi(t_{i^-})\|_0^2 - \|\phi(t_{i-1^-})\|_0^2 + a(1-\epsilon)\Delta t_i\Big(\frac{3}{2}\|\phi_{TR}(t_{i,1})\|_1^2 + \frac{1}{2}\|\phi(t_{i^-})\|_1^2\Big)\\ + \frac{1}{36}\|\Delta t_i^2\phi_{\tau\tau}\|_0^2 + \frac{3}{2}\Delta t_i\big(\phi_\tau(t_{i,1}), \phi_{TR}(t_{i,1}) - \phi(t_{i,1})\big)\\ \le C\Delta t_i\Big\{\|\eta_\tau(t_{i,1})\|^2_{(-1,\mathcal{V}_h(t_{i,1}))} + \|\eta_\tau(t_{i^-})\|^2_{(-1,\mathcal{V}_h(t_{i^-}))}\\ + \|\eta_{TR}(t_{i,1})\|_1^2 + \|\eta(t_{i^-})\|_1^2 + \|u(t_{i,1}) - u_{TR}(t_{i,1})\|_1^2\\ + \|\phi_{TR}(t_{i,1})\|_0^2 + \|\phi(t_{i^-})\|_0^2\Big\}. \quad (3.17)\end{aligned}$$

From the auxiliary Lemma 6.1 (in the appendix), bound

$$\begin{aligned}\frac{1}{36}\|\Delta t_i^2\phi_{\tau\tau}\|_0^2 + \frac{3}{2}\Delta t_i\big(\phi_\tau(t_{i,1}), \phi_{TR}(t_{i,1}) - \phi(t_{i,1})\big)\\ \ge \frac{3}{32}a\Delta t_i\Big((1-\epsilon - C\Delta t_i)\|\phi(\zeta_{i,1})\|_1^2 - (1+\epsilon+C\Delta t_i)\|\phi(t_{i-1^+})\|_1^2\Big)\\ - C\Delta t_i\Big\{\|\eta_\tau(t_{i,1})\|^2_{(-1,\mathcal{V}_h(t_{i,1}))} + \|\eta_{TR}(t_{i,1})\|_1^2\\ + \|u(t_{i,1}) - u_{TR}(t_{i,1})\|_1^2 + \|\phi_{TR}(t_{i,1})\|_0^2 + \|\phi(t_{i-1^+})\|_0^2\Big\}.\end{aligned}$$

Combining this result with (3.17) gives

$$\begin{aligned}\|\phi(t_{i^-})\|_0^2 - \|\phi(t_{i-1^-})\|_0^2 + a(1-\epsilon)\Delta t_i\Big(\frac{3}{2}\|\phi_{TR}(t_{i,1})\|_1^2 + \frac{1}{2}\|\phi(t_{i^-})\|_1^2\Big)\\ + \frac{3}{32}a\Delta t_i\Big((1-\epsilon - C\Delta t_i)\|\phi(\zeta_{i,1})\|_1^2 - (1+\epsilon+C\Delta t_i)\|\phi(t_{i-1^+})\|_1^2\Big)\\ \le C\Delta t_i\Big\{\|\eta_\tau(t_{i,1})\|^2_{(-1,\mathcal{V}_h(t_{i,1}))} + \|\eta_\tau(t_{i^-})\|^2_{(-1,\mathcal{V}_h(t_{i^-}))}\\ + \|\eta_{TR}(t_{i,1})\|_1^2 + \|\eta(t_{i^-})\|_1^2 + \|u(t_{i,1}) - u_{TR}(t_{i,1})\|_1^2\\ + \|\phi(t_{i-1^+})\|_0^2 + \|\phi_{TR}(t_{i,1})\|_0^2 + \|\phi(t_{i^-})\|_0^2\Big\}. \quad (3.18)\end{aligned}$$

To remove the ϕ terms from the upper bound, Lemma 2.2 with (3.4) shows

$$\begin{aligned}\|\phi(t_{i-1^+})\|_0^2 + \|\phi_{TR}(t_{i,1})\|_0^2 + \|\phi(t_{i^-})\|_0^2\\ \le \|\phi(t_{i-1^-})\|_0^2 + C\Big\{\|\eta_\tau(t_{i,1})\|^2_{(-1,\mathcal{V}_h(t_{i,1}))} + \|\eta_\tau(t_{i^-})\|^2_{(-1,\mathcal{V}_h(t_{i^-}))}\\ + \|\eta_{TR}(t_{i,1})\|_1^2 + \|\eta(t_{i^-})\|_1^2 + \|u(t_{i,1}) - u_{TR}(t_{i,1})\|_1^2\Big\}. \quad (3.19)\end{aligned}$$

The $\mathcal{H}^1$-stability of $\mathcal{L}_2$-projection [31] with (3.7) yields a telescoping sum for

$$\begin{aligned}\frac{1}{2}\Delta t_i(1-\epsilon)\|\phi(t_{i^-})\|_1^2 - \frac{3}{32}\Delta t_i(1+\epsilon)\|\phi(t_{i-1^+})\|_1^2\\ \ge \frac{1}{2}\Delta t_i(1-\epsilon)\|\phi(t_{i^-})\|_1^2 - \frac{3}{8}\Delta t_{i-1}(1+\epsilon)\|\phi(t_{i-1^-})\|_1^2\\ \ge \frac{1}{8}\Delta t_i\big(1-\epsilon\big)\|\phi(t_{i^-})\|_1^2 + \frac{3}{8}(1+\epsilon)\Big(\Delta t_i\|\phi(t_{i^-})\|_1^2 - \Delta t_{i-1}\|\phi(t_{i-1^-})\|_1^2\Big).\end{aligned}$$

This telescoping sum, together with (3.18)–(3.19), gives the bound

$$\begin{aligned}\Big[\|\phi(t_{i^-})\|_0^2 + \beta\Delta t_i\|\phi(t_{i^-})\|_1^2\Big] - (1+C''\Delta t_i)\Big[\|\phi(t_{i-1^-})\|_0^2 + \beta\Delta t_{i-1}\|\phi(t_{i-1^-})\|_1^2\Big] \\ + \theta\Delta t_i\Big\{\|\phi_{TR}(t_{i,1})\|_1^2 + \|\phi(t_{i^-})\|_1^2 + \|\phi(\zeta_{i,1})\|_1^2\Big\} \\ \leq C\Delta t_i\sum_{j=1}^{2}\left(\|\eta_\tau(t_{i,j})\|_{(-1,\mathcal{V}_h(t_{i,j}))}^2 + \|\eta_{RK}(t_{i,j})\|_1^2\right) + C\Delta t_i\|u(t_{i,1}) - u_{TR}(t_{i,1})\|_1^2, \quad (3.20)\end{aligned}$$

for $\beta = \frac{3}{8}a(1+\epsilon)$ and some $\theta > 0$.

For sufficiently small Δt_i, apply the discrete Grönwall lemma to bound

$$\begin{aligned}\max_{1\leq i\leq m}\|\phi(t_{i^-})\|_0^2 + \sum_{i=1}^{m}\Delta t_i\Big\{\|\phi_{i,TR}\|_1^2 + \|\phi(t_{i^-})\|_1^2 + \|\phi(\zeta_{i,1})\|_1^2\Big\} \\ \leq C\Big\{|\!|\!|\eta|\!|\!|^2 + \sum_{i=1}^{m}\Delta t_i\|u(t_{i,1}) - u_{TR}(t_{i,1})\|_1^2 + \|\phi(0)\|_0^2 + \Delta t\|\phi(0)\|_1^2\Big\}. \quad (3.21)\end{aligned}$$

For the additional terms in the upper bound, we have

$$\|\phi(0)\|_0 \leq \|\eta(0)\|_0 \leq |\!|\!|\eta|\!|\!|, \quad (3.22)$$

and by the $\mathcal{H}^1$-stability of $\mathcal{L}_2$-projection [31], we have

$$\Delta t\|\phi(0)\|_1^2 \leq C\Delta t\|\eta(0)\|_1^2 \leq C\Delta t\Big(\|\eta_{1,TR}\|_1^2 + \|\eta(\zeta_{1,1})\|_1^2\Big) \leq C|\!|\!|\eta|\!|\!|^2. \quad (3.23)$$

Since the trapezoid approximation is second order, we have

$$\Delta t_i\|u(t_{i,1}) - u_{TR}(t_{i,1})\|_1^2 \leq C_{TR}\int_{t_{i-1}}^{t_i}\|\Delta t_i^2 u_{\tau\tau}(t)\|_1^2\,dt, \quad (3.24)$$

which gives

$$\begin{aligned}\max_{1\leq i\leq m}\|\phi(t_{i^-})\|_0^2 + \sum_{i=1}^{m}\Delta t_i\Big\{\|\phi_{i,TR}\|_1^2 + \|\phi(t_{i^-})\|_1^2 + \|\phi(\zeta_{i,1})\|_1^2\Big\} \\ \leq C\Big\{|\!|\!|\eta|\!|\!|^2 + \int_0^T\|\Delta t^2 u_{\tau\tau}(t)\|_1^2\,dt\Big\}. \quad (3.25)\end{aligned}$$

For $t = t_{i,1}, t_{i,2}$, bound the characteristic derivative of ϕ:

$$\|\phi_\tau\|_{(-1,\mathcal{V}_h(t))} \leq C\Big\{\|\eta_\tau\|_{(-1,\mathcal{V}_h(t))} + \|\eta_{RK}\|_1 + \|u - u_{RK}\|_1 + \|\phi_{RK}\|_1\Big\}, \quad (3.26)$$

and use the local Grönwall lemma again to bound the maximum $\|\phi(t_{i,1})\|_0$ at the intermediate collocation nodes. Thus, combining (3.25)–(3.26) and the local Gröwall inequality, we have

$$|\!|\!|\phi|\!|\!|^2 \leq C\Big\{|\!|\!|\eta|\!|\!|^2 + \int_0^T\|\Delta t^2 u_{\tau\tau}(t)\|_1^2\,dt\Big\}.$$

The result follows from an application of the reverse triangle inequality.

For problems with Dirichlet boundary conditions, let $u_D(\cdot,t)$ be some $\mathcal{H}^1(\Omega)$ function satisfying the Dirichlet boundary condition at time t. Then, the solution can be decomposed, $u = u_D + u_0$, with $u_0(\cdot,t) \in \mathcal{H}^1_0(\Omega) \subset \mathcal{H}^1(\Omega)$, where $\mathcal{H}^1_0(\Omega)$ is the subspace satisfying a homogeneous Dirichlet boundary condition. Solving for the weak solution, u, is then equivalent to solving for $u_0(\cdot,t) \in \mathcal{H}^1_0(\Omega)$ satisfying

$$\big(u_{0,\tau}(\cdot,t),\chi\big) + \mathcal{A}_\tau(t;u_0,\chi) = \big(f(\cdot,t),\chi\big) + \big\langle g(\cdot,t),\chi\big\rangle - \Big[\big(u_{D,\tau}(\cdot,t),\chi\big) + \mathcal{A}_\tau(t;u_D,\chi)\Big],$$

for all $\chi \in \mathcal{H}^1_0(\Omega)$, and $0 < t \leq T$. To compute the finite element solution, we seek $u_{h,0} \in \mathcal{V}_{h,0} \subset \mathcal{V}_h$, which is the subspace of $\mathcal{V}_h$ without the degrees of freedom on the Dirichlet boundary. Then, the Galerkin orthogonality holds at the collocation points for u_0 and $u_{h,0}$, and an analogous argument to the proof of Theorem 3.1 can be used to bound $|\!|\!| u_0 - u_{h,0} |\!|\!|$ over all functions in $\mathcal{V}_{h,0}$.

4. Numerical experiments

In this section, we present numerical results that demonstrate some applications of TR-BDF2 quadratic moving finite elements. Although Theorem 3.1 assumed the basis node $\varepsilon = 2/3$, corresponding to Gauss-Radau quadrature, the numerical experiments here verify that second order convergence in time holds for nearby values of this collocation node, especially when the solution varies slowly along the characteristics of the mesh.

We implemented a moving mesh solver in MATLAB to test the performance of our method on a Dirichlet boundary value problem that develops a boundary layer. The test problem is

$$u_t(x,t) - 10^{-4}u_{xx}(x,t) + u_x(x,t) = 1, \qquad \text{for } (x,t) \in (0,1)\times(0,1],$$

subject to the boundary conditions $u(0,t) = u(1,t) = 1$ and initial condition $u(x,0) = 1$.

Dirichlet boundary conditions readily lead to boundary layers in the presence of strong convection; our mesh motion scheme uses a uniform mesh at the beginning of each time-step and $\mathcal{L}_2$-projection is used at the mesh discontinuities so that $\|u_h(t_{i^+})\|_1 \leq C_{\mathcal{H}}\|u_h(t_{i^-})\|_1$ holds across these mesh discontinuities [31]. The mesh evolves in time using $x_t = 1$ as an initial guess; this causes elements to flow out the right-side of the domain and so a relaxation is used $\frac{d}{dt}x_k(t) = \theta_k < 1$ to ensure that these outflowing elements crowd inside of the boundary and do not shrink by more than a factor of ten: $\frac{1}{10}\Delta x_k(t_{i-1^+}) \leq \Delta x_k(t_{i^-})$ for all i and k.

Figure 4.1 depicts the standard finite element solution, the moving finite element solution, and a moving space-time mesh (coarsened for clearer presentation). For this problem, $\varepsilon = 2-\sqrt{2}$, which is the value that optimizes the truncation error of TR-BDF2 integration scheme.

The moving mesh yields a steady state solution $u_h \approx 1+x$ with a boundary layer, as expected so that $u_x \approx 1$, as suggested by the differential equation. We see that the adaptivity provided by the mesh motion has appropriately aggregated nodes in the boundary layer, stabilizing the computed solution.

Another experiment implements the moving finite element scheme for a nonlinear problem. Burgers' equation is an interesting test problem for our scheme as it is a simple nonlinear equation that is a one-dimensional analogue of the Navier-Stokes equation that develops steep moving fronts that sweep through the domain. As both Burgers' equations and the Navier-Stokes equations have a convection term that is proportional to the solution of the equation,

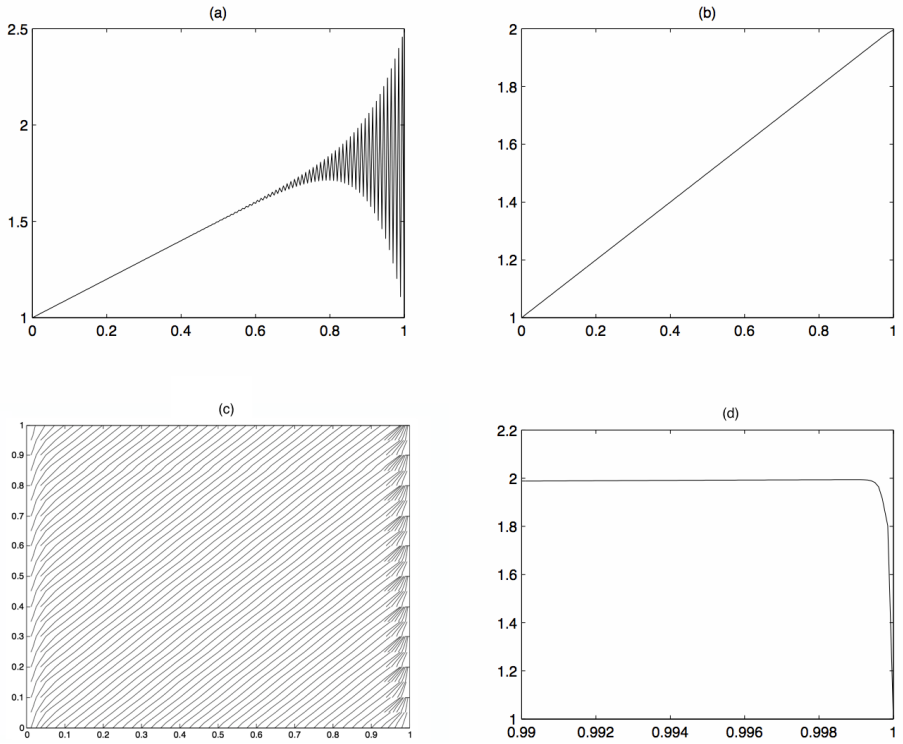

Fig. 4.1. Computed steady state solutions to the Dirichlet test problem with the same initial mesh configuration. The standard finite element solution is shown in (a); the moving mesh solution is shown in (b). A moving space-time mesh is depicted in (c) that has been coarsened for a clearer presentation; the mesh is reset to be uniform at each time step and outflowing nodes are crowded inside the boundary to ensure a quasi-uniform mesh. Plot (d) shows a close-up of the boundary layer for the moving mesh solution.

the exact same mesh-moving scheme can be used for both problems. (Note that the divergence-free property of the solution to Navier-Stokes helps avoid mesh degeneration as the mesh evolves along characteristics of the computed solution.)

Artificial oscillations are commonly found near the shock layer of the computed solution of Burgers' equations when the time discretization is not sufficiently refined [4, 5]. Several techniques have been developed to stabilize the computed solutions for such problems using moving meshes, adaptive meshing [32, 33], space-time formulations [34], Galerkin methods using a method-of-lines approach [35], and upwind post-processing techniques [36]. For this experiment, the moving space-time method solves a one-dimensional Burgers' equation with a large Reynolds number. The differential equation is given by

$$u_t - \frac{1}{R}u_{xx} + uu_x = 0 \qquad \text{for } (x,t) \text{ in } (-3,3)\times(0,2], \tag{4.1}$$

where we choose a large $R > 0$ and assume Neumann boundary conditions, $u_x(\pm 3, t) = 0$. The initial condition is chosen so that a moving front forms in the middle of the domain and propagates toward the right boundary.

We implemented a solver for this problem in C++ that uses an approximate method of characteristics to move the mesh, $x_t \approx u$. The solution is computed at the end of each time step so that we can use linear mesh motion, $x_t(t) = u_h(x, t_{i-1^+})$ for $t_{i-1} \le t \le t_i$. Furthermore,

we use a single step of Newton's method to linearize the equation at each collocation node, using $u_h(t_{i-1^+})$ as the initial guess. Unlike the previous experiment, we do not reset the mesh to be uniform at the beginning of each time partition. This allows the nodes to appropriately accumulate near the shock layer, where elements are coarsened once they shrink by more than a factor of 10^{-4} of their original size. To ensure the $\mathcal{H}^1$-stability of the solution across these discontinuities, $\mathcal{L}_2$-projection is used to define the solution on the new mesh.

In addition to the standard Galerkin discretizations for the non-moving and moving meshes, we also compare our results to a solution computed using a Streamline Upwind/Petrov Galerkin (SUPG) discretization [37] to the *linearized* equation on a fixed mesh. Let $\bar{\bar{u}}_h$ denote the SUPG computed solution and $w_h = \bar{\bar{u}}_h(t_{i-1^+})$ denote the initial guess for the Newton iteration; the linearized equation describing the SUPG finite element solution is given by modifying the space of test functions to take the form $\chi + \delta_h w_h \chi_x$ with $\chi \in \mathcal{V}_h$, where $\delta_h = \sigma \Delta x_i$ on each element for some SUPG coefficient $\sigma > 0$:

$$\begin{aligned}\left(\bar{\bar{u}}_{h,t}(t_{i,j}), \chi\right) + \frac{1}{R}\left(\bar{\bar{u}}_{h,x}(t_{i,j}), \chi_x\right) + \left(w_h \bar{\bar{u}}_{h,x}(t_{i,j}), \chi\right) + \left(w_{h,x} \bar{\bar{u}}_h(t_{i,j}), \chi\right) \\ + \left(\bar{\bar{u}}_{h,t}(t_{i,j}) - \tfrac{1}{R}\bar{\bar{u}}_{h,xx}(t_{i,j}) + w_h \bar{\bar{u}}_{h,x}(t_{i,j}) + w_{h,x}\bar{\bar{u}}_h(t_{i,j}), \delta_h w_h \chi_x\right)_h = 0, \quad (4.2)\end{aligned}$$

where we use the broken $\mathcal{L}_2$-inner-product summing over the elements in the mesh, $(v, w)_h = \sum_e (v, w)_e$. We see that this formulation gives a consistent diffusive term of order $\delta_h w_h^2$, which provides stability near the shock layer.

Figure 4.2 displays solutions for equation (4.1) with $R = 100$, computed on static and moving meshes, as well as the solution to (4.2) on a fixed mesh with SUPG coefficient $\sigma = 0.1$. Numerically-induced oscillations are expectedly present in solution on the non-moving mesh [38], whereas they have been suppressed in the SUPG and the moving mesh solutions. From Figure 4.2, the moving finite element solution maintains a steep drop into the moving front, where artificial diffusion is clearly present at the top of the shock layer in the SUPG solution. This demonstrates the increased flexibility in the time discretization when moving the mesh, without the artificial diffusion of a SUPG discretization. Experiments combining SUPG and moving meshes give similar results to the moving finite element solution in Figure 4.2 since the "streamline direction," now given by $u - x_t$, is small. Figure 4.3 depicts an example of a moving mesh used for solving the PDE, coarsened for improved presentation. We see the spatial nodes accumulating at the shock layer, as desired.

We also ran simulations where the Reynolds number is set to $R = 1000$. This reduces the diffusive forces in the equation and leads to a thinner shock layer. The SUPG parameter is set to $\sigma = 1$ and solutions computed with 100 time steps are displayed in Figure 4.4, where we again see the moving finite element solution is remarkably smoother on either side of the shock layer without artificial diffusion at the shock.

We compare the runtimes of computing a moving mesh solution using TR-BDF2 and the fully-implicit second-order backward difference time integration scheme with the mid-step basis node set to be $\varepsilon = 2 - \sqrt{2}$. Thus, the only difference between the TR-BDF2 scheme and the implicit backward difference scheme is that $u_{h,TR}(t_{i,1})$ is replaced in (3.1) by its quadratic interpolant:

$$u_h(t_{i,1}) = \frac{2-\varepsilon}{4} u_h(t_{i-1^+}) + \frac{2-\varepsilon}{4(1-\varepsilon)} u_h(\zeta_{i,1}) - \frac{\varepsilon^2}{2(1-\varepsilon)} u_h(t_{i^-})$$

so that the solution at the mid-step and end-step must be computed simultaneously. We see that the semi-implicit property of TR-BDF2 always leads to faster runtimes, as evidenced in

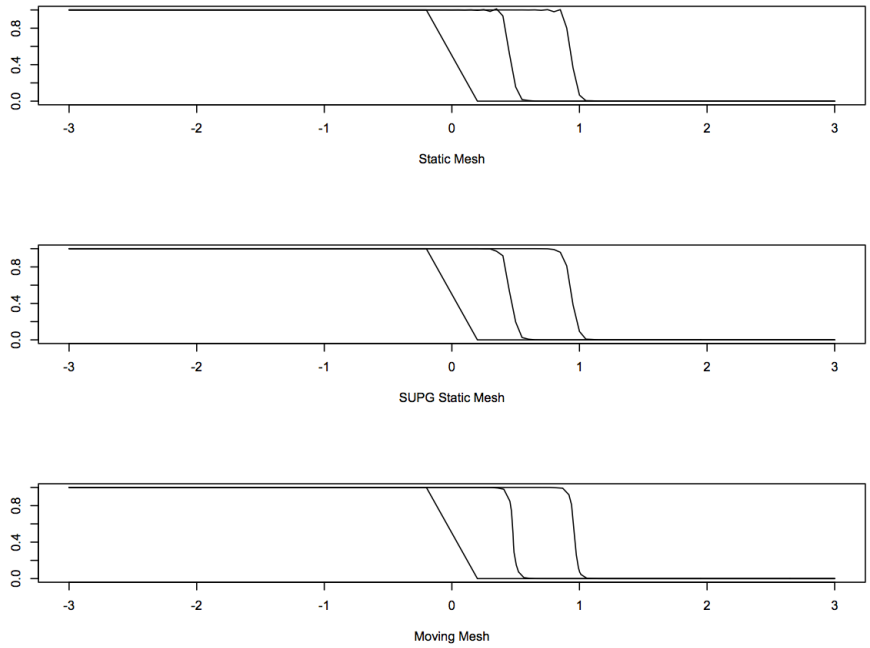

Fig. 4.2. Solutions computed for Burgers' equation with $R = 100$, using $n = 61$ spatial nodes and $m = 25$ time steps. For the SUPG discretization, we take the SUPG coefficient $\sigma = 0.1$. Each graph displays the initial condition and the solution at the intermediate $t = 1$, and the final time $t = 2$.

Table 4.1. For a discretization with N nodes, the fully-implicit scheme requires solving a linear system of $2N$ unknowns for each time step, whereas two linear systems of N unknowns are sequentially solved for the TR-BDF2 scheme. As a result, the relative speedup becomes more pronounced as the spatial mesh is refined. We note that more iterations of Newton's method or a higher-dimensional problem would indeed lead to an even greater relative speedup when using TR-BDF2.

The last experiment of this section presents the results of a two-dimensional problem where a Gaussian bump is transported along a circular trajectory. The exact solution to the problem is given by

$$u(x,t) = e^{-2\left\{(x_1-\cos(2\pi t))^2+(x_2-\sin(2\pi t))^2\right\}}$$

on a disc with $x_1^2 + x_2^2 \leq 9$ and $0 \leq t \leq T = 1$. The solution is propagated in time using the differential equation described by (2.3) with diffusion coefficient $a = 10^{-4}$, convection term $b = (-2\pi x_2,\, 2\pi x_1)^T$, reaction term $c = 0$, and the source function f is chosen so that the solution is given by u. The choice of convection term, b, closely tracks the motion of the Gaussian bump so that the source term, f, is small. Dirichlet boundary conditions are imposed, where the values are sampled from the known solution.

Two strategies for mesh motion are compared. The first approach employs a static mesh; the second approach requires analytically solving a simple ordinary differential equation so that $x_t = b$, which is possible for this convection term, and shows how de-stabilizing the convection term truly is. Since the mesh motion corresponds to rotating a mesh on a circular domain, the mesh does not undergo any discontinuous changes at the discrete time-steps.

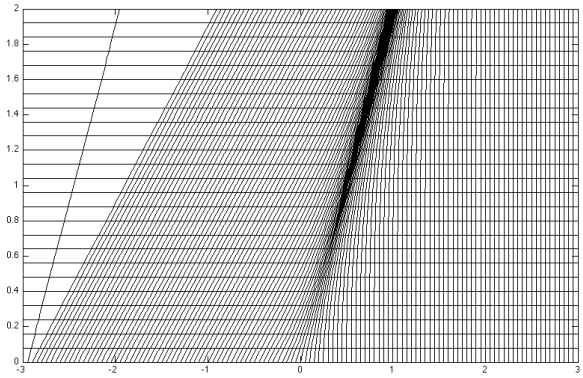

Fig. 4.3. An example of a moving mesh with $m = 25$ time steps and initialized with $n = 61$ spatial nodes at the beginning of the simulation. The method of characteristics sets $x_t(t) = u_h(x, t_{i-1^+})$ at the vertices in the mesh, bump node trajectories are defined to bisect their neighbors as seen on the left side of the mesh.

Relative Speedup					
Δx	$\Delta t = \frac{1}{100}$	$\Delta t = \frac{1}{200}$	$\Delta t = \frac{1}{400}$	$\Delta t = \frac{1}{800}$	$\Delta t = \frac{1}{1600}$
1/50	1.3250	1.3781	1.3827	1.3840	1.3839
1/100	1.5820	1.5701	1.5769	1.5787	1.5814
1/200	1.7696	1.7613	1.7664	1.7728	1.7815
1/400	2.0921	2.1061	2.0799	2.0880	2.1038
1/800	2.2674	1.8399	2.3516	2.3585	2.3683

Table 4.1: This table displays the relative speedup of the total CPU runtime when using TR-BDF2 time integration compared to a fully implicit second-order integration scheme (BDF2). The total CPU runtime includes computing the mesh evolution, assembling the stiffness matrix, solving the linear systems, and projecting the solution across mesh discontinuities. The values in the table are the (BDF2 runtime) / (TR-BDF2 runtime) computed for the entire simulation, for each discretization.

The software package PLTMG by Bank [39] provides the foundation for the presented work, in which the proposed finite element method of this paper uses TR-BDF2 integration to evolve a computed solution to the initial value problem to time $T = 1$. For the moving mesh, the mid-point basis in time is set to $\varepsilon = 2 - \sqrt{2}$ and $2/3$ to test the optimal truncation collocation scheme and Gauss-Radau quadrature scheme. Below are three sets of tables that detail the $\mathcal{H}^1$-error of the computed solution for $t = 0.25$, 0.5, 1 using the above meshing strategies.

Figure 4.5 displays the relative $\mathcal{H}^1$-error values and convergence plots for the first approach using a static mesh: $\|u - u_h\|_1/\|u\|_1$. Figures 4.2 and 4.3 present error values and convergence plots for the moving mesh with $\varepsilon = 2 - \sqrt{2}$ and $\varepsilon = 2/3$, respectively. Reviewing the order of convergence for these computed solutions shows that the solution computed on a static mesh suffers from too much truncation error in time. As a result, the static mesh strategy does not appear to have quadratic convergence as the bounding constant from the error analysis is too large. On the other hand, the moving meshes clearly show quadratic convergence as the spatial

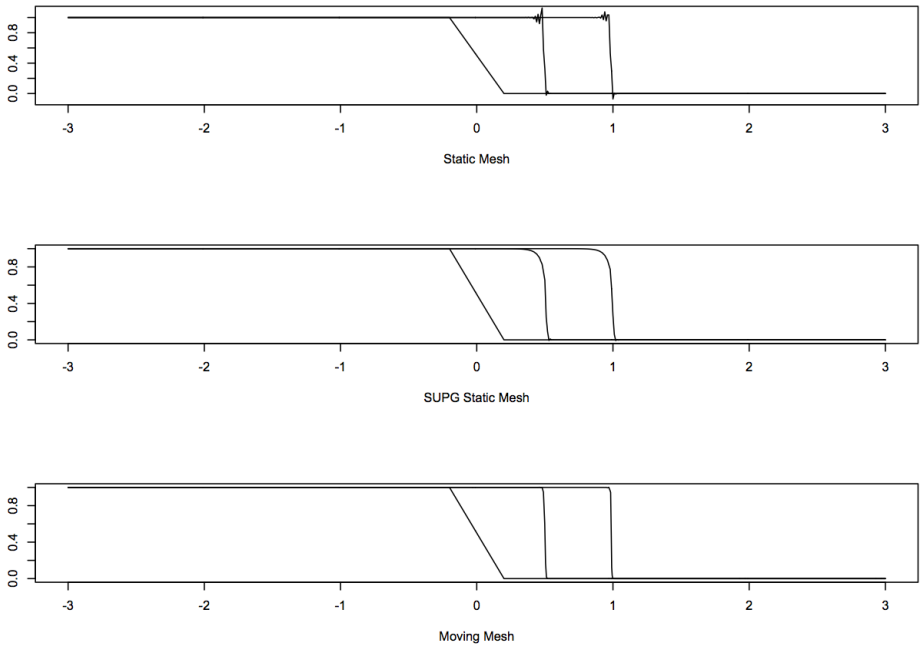

Fig. 4.4. Solutions computed for Burgers' equation with $R = 1000$, using $n = 301$ spatial nodes and $m = 100$ time steps. For the SUPG discretization, we take the SUPG coefficient $\sigma = 1$. Each graph displays the initial condition and the solution at the intermediate $t = 1$, and the final time $t = 2$.

and time discretization are refined. Figure 4.6 depicts solutions computed at various times on a static mesh and a moving mesh, where the moving mesh clearly shows an improved solution.

We close this section by commenting that mesh motion schemes have been developed using error estimates [18] and predictor-corrector techniques [24, 1] that fall into the theoretical framework presented here and could be used for extending our experiments. A multi-dimensional mesh motion scheme of primary interest could use a posteriori error estimates and adaptive meshing (h-refinement at the mesh discontinuities and mesh smoothing to evolve the mesh continuously) to find a suitable moving mesh [40, 41].

5. Conclusion

Theorem 4.3 of [6] provides a symmetric error bound for some space-time moving finite element methods that use a one parameter family of fully implicit collocation methods. Employing TR-BDF2 for time integration, however, does not fit into this framework and complicates the analysis of this finite element method, ultimately breaking the symmetry of the error bound. Nevertheless, it is straightforward to show that Theorem 3.1 implies second order accuracy with respect to mesh refinement using Theorem 2 of [6]. Furthermore, improved accuracy for the moving finite element solution is maintained in application to a simple nonlinear problem, although the theoretical analysis does not cover such a case.

In the numerical experiments, the method of characteristics was used to determine the mesh motion, though other mesh moving schemes may yield superior results. Generally, however, an

	$T = 1/4$			
Δx	$\Delta t = \frac{1}{4}$	$\Delta t = \frac{1}{8}$	$\Delta t = \frac{1}{12}$	$\Delta t = \frac{1}{20}$
3/10	0.986	0.582	0.372	0.193
3/20	0.986	0.579	0.366	0.176
3/30	0.985	0.579	0.365	0.174
3/50	0.985	0.579	0.364	0.172
	$T = 1/2$			
3/10	1.24	0.876	0.619	0.343
3/20	1.24	0.872	0.612	0.323
3/30	1.24	0.872	0.610	0.320
3/50	1.24	0.872	0.610	0.319
	$T = 1$			
3/10	1.37	1.12	0.898	0.569
3/20	1.37	1.12	0.888	0.546
3/30	1.37	1.12	0.887	0.543
3/50	1.37	1.12	0.886	0.542

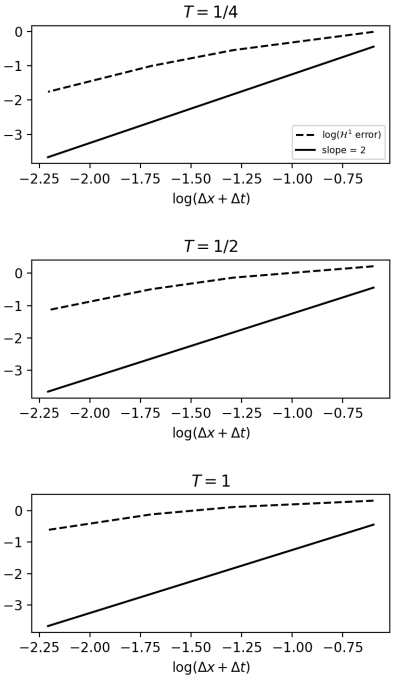

Fig. 4.5. The table on the left displays the relative $\mathcal{H}^1$-norm of the error of the computed solution on a static mesh at times $T = 0.25$, 0.5, and 1. The values for Δx represent the diameter of the spatial elements at each time-slice. TR-BDF2 time integration is used with $\varepsilon = 2 - \sqrt{2}$. The figures on the right plot the $\log(\mathcal{H}^1\text{-error})$ as sampled along the diagonal of the tables to show the convergence as the spatial and temporal discretizations are refined. The convergence rate is shown to be less than second-order when compared to a line of slope two, and larger values of T show that accumulated truncation errors lead to worse performance.

optimally robust and well-defined mechanism for evolving the mesh for general PDEs is still a matter of active research. It is suspected that predictor-corrector methods, coupled with adaptive meshing for the spatial discretization, can be powerful tools in moving the mesh without requiring additional user-supplied information about a PDE or its solution.

6. Appendix

Here is a technical lemma that is used to prove Theorem 3.1.

Lemma 6.1. *For ϕ in $\mathcal{V}_h$ and η in $\mathcal{V}$ satisfying (3.9) at time $t = t_{i,1}$, it holds that*

$$\begin{aligned}\frac{1}{36}\|\Delta t_i^2 \phi_{\tau\tau}(t_{i,1})\|_0^2 + \frac{3}{2}\Delta t_i\big(\phi_\tau(t_{i,1}), \phi_{TR}(t_{i,1}) - \phi(t_{i,1})\big)\\ \geq \frac{3}{32}a\Delta t_i\Big((1-\epsilon-C\Delta t_i)\|\phi(\zeta_{i,1})\|_1^2 - (1+\epsilon+C\Delta t_i)\|\phi(t_{i-1^+})\|_1^2\Big)\\ - C\Delta t_i\Big\{\|\eta_\tau\|_{(-1,\mathcal{V}_h)}^2 + \|\eta_{TR}\|_1^2 + \|u-u_{TR}\|_1^2 + \|\phi_{TR}\|_0^2 + \|\phi(t_{i-1^+})\|_0^2\Big\},\end{aligned}$$

for positive constant C and arbitrarily small positive constant ϵ.

	$T = 1/4$			
Δx	$\Delta t = \frac{1}{4}$	$\Delta t = \frac{1}{8}$	$\Delta t = \frac{1}{12}$	$\Delta t = \frac{1}{20}$
3/10	2.95E-1	5.58E-2	3.29E-2	2.87E-2
3/20	2.92E-1	4.81E-2	1.81E-2	8.52E-3
3/30	2.92E-1	4.75E-2	1.69E-2	5.81E-3
3/50	2.92E-1	4.74E-2	1.67E-2	5.04E-3
	$T = 1/2$			
3/10	5.58E-1	9.99E-2	4.41E-2	3.00E-2
3/20	5.56E-1	9.50E-2	3.41E-2	1.21E-2
3/30	5.56E-1	9.46E-2	3.34E-2	1.03E-2
3/50	5.56E-1	9.44E-2	3.32E-2	9.90E-3
	$T = 1$			
3/10	9.90E-1	1.92E-1	7.31E-2	3.46E-2
3/20	9.89E-1	1.88E-1	6.69E-2	2.09E-2
3/30	9.89E-1	1.88E-1	6.65E-2	1.99E-2
3/50	9.88E-1	1.88E-1	6.63E-2	1.97E-2

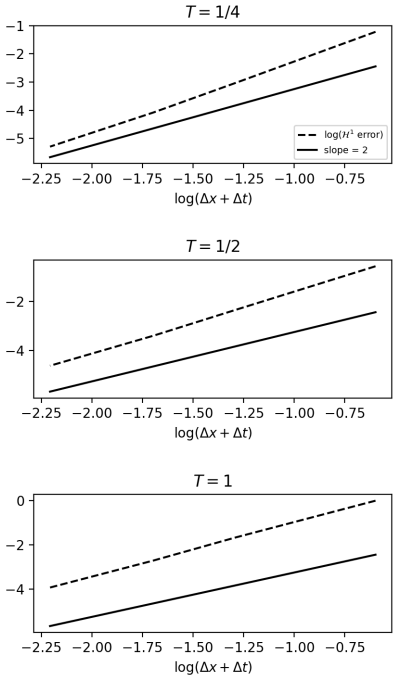

Table 4.2: The table on the left displays the relative $\mathcal{H}^1$-norm of the error of the computed solution on a moving mesh at times $T = 0.25$, 0.5, and 1. TR-BDF2 time integration is used with $\varepsilon = 2 - \sqrt{2}$. The figures on the right plot the $\log(\mathcal{H}^1\text{-error})$ as sampled along the diagonal of the tables to show the convergence as the spatial and temporal discretizations are refined. The convergence rate is shown to be second-order when compared to a line of slope two; note that the convergence rate is the same for larger T, though the bounding constant is larger.

Proof. Fix $1 \leq i \leq M$ and, to avoid clutter, suppress the time variable, though it is understood that the following argument holds at time $t = t_{i,1}$. The error of the trapezoid approximation is given exactly by

$$\phi_{TR} - \phi = \frac{1}{18}\Delta t_i \phi_{\tau\tau},$$

which is used with the Cauchy-Schwartz inequality to show

$$\begin{aligned}
\Delta t_i(\phi_\tau, \phi_{TR} - \phi) &= \frac{1}{18}(\Delta t_i \phi_\tau, \Delta t_i^2 \phi_{\tau\tau}) \\
&\geq -\frac{1}{18}\|\Delta t_i \phi_\tau\|_0 \|\Delta t_i^2 \phi_{\tau\tau}\|_0 \\
&\geq -\frac{1}{18}\left\{\frac{\delta}{4}\|\Delta t_i \phi_\tau\|_0^2 - \frac{1}{\delta}\|\Delta t_i^2 \phi_{\tau\tau}\|_0^2\right\} \quad (\delta = 3) \\
&\geq -\frac{1}{24}\|\Delta t_i \phi_\tau\|_0^2 - \frac{1}{54}\|\Delta t_i^2 \phi_{\tau\tau}\|_0^2.
\end{aligned}$$

This leads to the initial lower bound:

$$\frac{1}{36}\|\Delta t_i^2 \phi_{\tau\tau}\|_0^2 + \frac{3}{2}\Delta t_i(\phi_\tau, \phi_{TR} - \phi) \geq -\frac{1}{16}\|\Delta t_i \phi_\tau\|_0^2. \tag{6.1}$$

	$T = 1/4$			
Δx	$\Delta t = \frac{1}{4}$	$\Delta t = \frac{1}{8}$	$\Delta t = \frac{1}{12}$	$\Delta t = \frac{1}{20}$
3/10	2.54E-1	4.51E-2	3.02E-2	4.51E-2
3/20	2.51E-1	3.54E-2	1.25E-2	7.30E-3
3/30	2.51E-1	3.47E-2	1.08E-2	3.80E-3
3/50	2.51E-1	3.45E-2	1.04E-2	2.50E-3
	$T = 1/2$			
3/10	4.89E-1	7.55E-2	3.52E-2	7.55E-2
3/20	4.86E-1	6.96E-2	2.19E-2	8.28E-3
3/30	4.86E-1	6.91E-2	2.09E-2	5.45E-3
3/50	4.86E-1	6.89E-2	2.07E-2	4.63E-3
	$T = 1$			
3/10	8.98E-1	1.42E-1	5.06E-2	1.42E-1
3/20	8.97E-1	1.38E-1	4.21E-2	1.14E-2
3/30	8.96E-1	1.37E-1	4.15E-2	9.50E-3
3/50	8.96E-1	1.37E-1	4.13E-2	9.05E-3

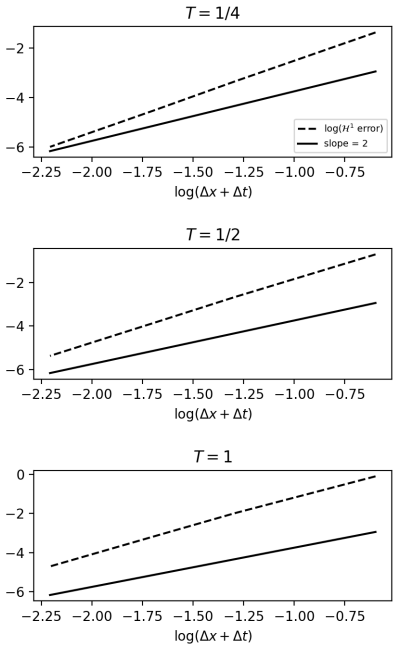

Table 4.3: The table on the left displays the relative $\mathcal{H}^1$-norm of the error of the computed solution on a moving mesh at times $T = 0.25$, 0.5, and 1. TR-BDF2 time integration is used with the Gauss-Radau collocation node, $\varepsilon = 2/3$, as required by Theorem 3.1. The figures on the right plot the $\log(\mathcal{H}^1\text{-error})$ as sampled along the diagonal of the tables to show the convergence as the spatial and temporal discretizations are refined. The convergence rate is shown to be slightly faster than second-order when compared to a line of slope two, as Gauss-Radau integration converges at a third-order rate. The convergence rate is at least second-order for all values of T.

To bound the norm of the time derivative, set $\chi = \Delta t_i^2 \phi_\tau$ in (3.9). This gives

$$\|\Delta t_i \phi_\tau\|_0^2 = (\eta_\tau, \Delta t_i^2 \phi_\tau) + \mathcal{A}_\tau(\eta_{TR}, \Delta t_i^2 \phi_\tau) + \mathcal{A}_\tau(u - u_{TR}, \Delta t_i^2 \phi_\tau) - \mathcal{A}_\tau(\phi_{TR}, \Delta t_i^2 \phi_\tau), \quad (6.2)$$

where each term on the right can be bounded separately.

Let $\tilde{\phi}_{i,j} = \phi\big(x(t_{i,1}), \zeta_{i,j}\big)$ denote $\phi(\zeta_{i,j})$ shifted onto the mesh of $\mathcal{V}_h(t_{i,1})$:

$$(\eta_\tau, \Delta t_i^2 \phi_\tau) \le C \Delta t_i \|\eta_\tau\|^2_{(-1, \mathcal{V}_h(t_{i,1}))} + a\epsilon \Delta t_i \Big(\big\|\tilde{\phi}_{i,1}\big\|_1^2 + \big\|\tilde{\phi}_{i,0}\big\|_1^2 \Big), \quad (6.3)$$

$$\mathcal{A}_\tau(\eta_{TR}, \Delta t_i^2 \phi_\tau) \le C \Delta t_i \|\eta_{TR}\|_1^2 + a\epsilon \Delta t_i \Big(\big\|\tilde{\phi}_{i,1}\big\|_1^2 + \big\|\tilde{\phi}_{i,0}\big\|_1^2 \Big), \quad (6.4)$$

$$\mathcal{A}_\tau(u - u_{TR}, \Delta t_i^2 \phi_\tau) \le C \Delta t_i \|u - u_{TR}\|_1^2 + a\epsilon \Delta t_i \Big(\big\|\tilde{\phi}_{i,1}\big\|_1^2 + \big\|\tilde{\phi}_{i,0}\big\|_1^2 \Big), \quad (6.5)$$

and, assuming the Gauss-Radau collocation node $\varepsilon = 2/3$,

$$\begin{aligned} -\mathcal{A}_\tau(\phi_{TR}, \Delta t_i^2 \phi_\tau) &= -\frac{3}{2} \Delta t_i \mathcal{A}_\tau(\tilde{\phi}_{i,1} - \tilde{\phi}_{i,0}, \tilde{\phi}_{i,1} + \tilde{\phi}_{i,0}) \\ &\le -\frac{3}{2} a \Delta t_i \Big((1-\epsilon) \big\|\tilde{\phi}_{i,1}\big\|_1^2 - (1+\epsilon) \big\|\tilde{\phi}_{i,1}\big\|_1^2 \Big) \\ &\quad + C \Delta t_i \Big(\|\phi_{TR}\|_0^2 + \|\tilde{\phi}_{i,0}\|_0^2 \Big). \end{aligned} \quad (6.6)$$

Combining bounds (6.1)–(6.6) closely resembles the desired result, where the finite element function ϕ is shifted onto the mesh at time $t = t_{i,1}$. To get the desired result, apply (2.8) and (2.9) from the Shift Lemma 2.1. Note that the ϵ in the result is understood to be different from the ϵ in the above bounds.

References

[1] M.J. Baines, Moving finite elements, Clarendon Press Oxford, 1994.

[2] N.N. Carlson and K. Miller, Design and application of a gradient-weighted moving finite element code. I. In one dimension, *SIAM J. Sci. Comput.*, **19**:3 (1998), 728–765.

[3] N.N. Carlson and K. Miller, Design and application of a gradient-weighted moving finite element code. II. In two dimensions, *SIAM J. Sci. Comput.*, **19**:3 (1998), 766–798.

[4] K. Miller, Moving finite elements. II, *SIAM J. Numer. Anal.*, **18** (1981), 1033–1057.

[5] K. Miller and R.N. Miller, Moving finite elements. I, *SIAM J. Numer. Anal.*, **18** (1981), 1019–1032.

[6] R.E. Bank and M.S. Metti, An error analysis of some higher order space-time moving finite elements, *Computing and Visualization in Science*, **16**:5 (2013), 219–229.

[7] R.E. Bank and R.F. Santos, Analysis of some moving space-time finite element methods, *SIAM J. Numer. Anal.*, **30**:1 (1993), 1–18.

[8] T. Dupont, Mesh modification for evolution equations, *Math. Comp.*, **39**:159 (1982), 85–107.

[9] T.F. Dupont and Y. Liu, Symmetric error estimates for moving mesh Galerkin methods for advection-diffusion equations, *SIAM Journal on Numerical Analysis*, **40**:3 (2002), 914–927.

[10] R.E. Bank, T.F. Dupont, S. Garcia, Y. Liu and R.F. Santos, Symmetric error estimates for moving mesh mixed methods for advection-diffusion equations, *SIAM J. Numer. Anal.*, **40**:6 (2002), 2270–2291 (electronic) (2003).

[11] N. Kopteva and M. Stynes, A robust adaptive method for a quasi-linear one-dimensional convection-diffusion problem, *SIAM Journal on Numerical Analysis*, **39**:4 (2001), 1446–1467.

[12] R.E. Bank, W.M. Coughran, W. Fichtner, E.H. Grosse and R.K. Smith, Transient simulation of silicon devices and circuits, *IEEE Trans. on Electron Devices*, **ED-32** (1985), 1992–2005.

[13] R.E. Bank, W.M. Coughran, W. Fichtner, E.H. Grosse and R.K. Smith, Transient simulation of silicon devices and circuits, *IEEE Trans. CAD*, **CAD-4** (1985), 436–451.

[14] S. Dharmaraja, G. Strang and Y. Wang, Optimal stability for trapezoidal–backward difference split-steps, *IMA J. Numer. Anal.*, **30**:1 (2010), 141–148.

[15] M. Hosea and L. Shampine, Analysis and implementation of TR-BDF2, *Applied Numerical Mathematics*, **20**:1 (1996), 21–37.

[16] M.S. Metti, Analysis of Some Higher Order Space-Time Moving Finite Element Methods, PhD thesis, Department of Mathematics, University of California, San Diego, CA, 2013.

[17] A.P. Kuprat, Creation and Annihilation of Nodes for the Moving Finite Element Method, PhD thesis, Department of Mathematics, University of California, Berkeley, CA, 1992.

[18] S. Adjerid and J.E. Flaherty, A moving finite element method with error estimation and refinement for one-dimensional time dependent partial differential equations, *SIAM Journal on Numerical Analysis*, **23**:4 (1986), 778–796.

[19] R. Li, T. Tang and P. Zhang, Moving mesh methods in multiple dimensions based on harmonic maps, *Journal of Computational Physics*, **170**:2 (2001), 562–588.

[20] R.F. Santos, Moving Space-Time Finite Element Methods for Convection-Diffusion Problems, PhD thesis, Department of Mathematics, University of California, San Diego, CA, 1991.

[21] S.R. Choudhury, Waveform relaxation techniques for linear and nonlinear diffusion equations, *Journal of computational and applied mathematics*, **42**:2 (1992), 253–267.

[22] P. Farrell, G. Gorman, C. Pain, M. Piggott and C. Wilson, Conservative interpolation between unstructured meshes via supermesh construction, *Computer Methods in Applied Mechanics and*

Engineering, **198**:33 (2009), 2632–2642.

[23] M.T. Heath and X. Jiao, Common-refinement-based data transfer between non-matching meshes in multiphysics simulations, *International Journal for Numerical Methods in Engineering*, **61**:14 (2004), 2402–2427.

[24] S. Adjerid and J.E. Flaherty, A moving-mesh finite element method with local refinement for parabolic partial differential equations, *Computer methods in applied mechanics and engineering*, **55**:1 (1986), 3–26.

[25] T. Tang, Moving mesh methods for computational fluid dynamics, *Contemporary mathematics*, **383** (2005), 141–174.

[26] P. Houston and E. Süli, Adaptive lagrange–galerkin methods for unsteady convection-diffusion problems, *Mathematics of computation*, **70**:233 (2001), 77–106.

[27] Z. Chen and G. Ji, Adaptive computation for convection dominated diffusion problems, *Science in China Series A: Mathematics*, **47** (2004), 22–31.

[28] X. Hu, Y.J. Lee, J. Xu and C.S. Zhang, On adaptive eulerian–lagrangian method for linear convection–diffusion problems, *Journal of Scientific Computing*, **58**:1 (2014), 90–114.

[29] M. Baines, M. Hubbard and P. Jimack, A moving mesh finite element algorithm for the adaptive solution of time-dependent partial differential equations with moving boundaries, *Applied Numerical Mathematics*, **54**:3 (2005), 450–469.

[30] M. Baines, M. Hubbard and P. Jimack, Velocity-based moving mesh methods for nonlinear partial differential equations, *Commun. Comput. Phys.*, **10**:3 (2011), 509–576.

[31] R.E. Bank and H. Yserentant, On the H^1 stability of L_2 projection onto finite element spaces, *Numerische Mathematik*, **126**:2 (2014), 361–381.

[32] J.de Frutos, B. García-Archilla, V. John and J. Novo, An adaptive SUPG method for evolutionary convection–diffusion equations, *Computer Methods in Applied Mechanics and Engineering*, **273** (2014), 219–237.

[33] J.de Frutos, B. García-Archilla and J. Novo, An adaptive finite element method for evolutionary convection dominated problems, *Computer Methods in Applied Mechanics and Engineering*, **200**:49 (2011), 3601–3612.

[34] E. Varoğlu and W.D. Liam Finn, Space-time finite elements incorporating characteristics for the Burgers' equation, *International Journal for Numerical Methods in Engineering*, **16**:1 (1980), 171–184.

[35] A. Dogan, A Galerkin finite element approach to Burgers' equation, *Applied Mathematics and Computation*, **157**:2 (2004), 331 – 346.

[36] J.de Frutos, B. García-Archilla and J. Novo, Stabilization of Galerkin finite element approximations to transient convection-diffusion problems, *SIAM Journal on Numerical Analysis*, **48**:3 (2010), 953–979.

[37] A.N. Brooks and T.J. Hughes, Streamline upwind/Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equations, *Computer Methods in Applied Mechanics and Engineering*, **32**:1–3 (1982), 199 – 259.

[38] X. Hu, J. Jia, J. Xu and C.S. Zhang, Effects of integrations and adaptivity for the Eulerian–Lagrangian method, *J. Comp. Math.*, **29** (2011), 367–395.

[39] R.E. Bank, PLTMG, a software package for solving elliptic partial differential equations: users' guide 11.0, volume 5, Department of Mathematics, University of California, San Diego, 2012.

[40] R.E. Bank, A.H. Sherman and A. Weiser, Refinement algorithms and data structures for regular local mesh refinement, Scientific Computing (Applications of Mathematics and Computing to the Physical Sciences) (R. S. Stepleman, ed.), pages 3–17, North Holland, 1983.

[41] K. Lipnikov and M. Shashkov, The error-minimization-based strategy for moving mesh methods, *Commun. Comput. Phys*, **1**:1 (2006), 53–80.

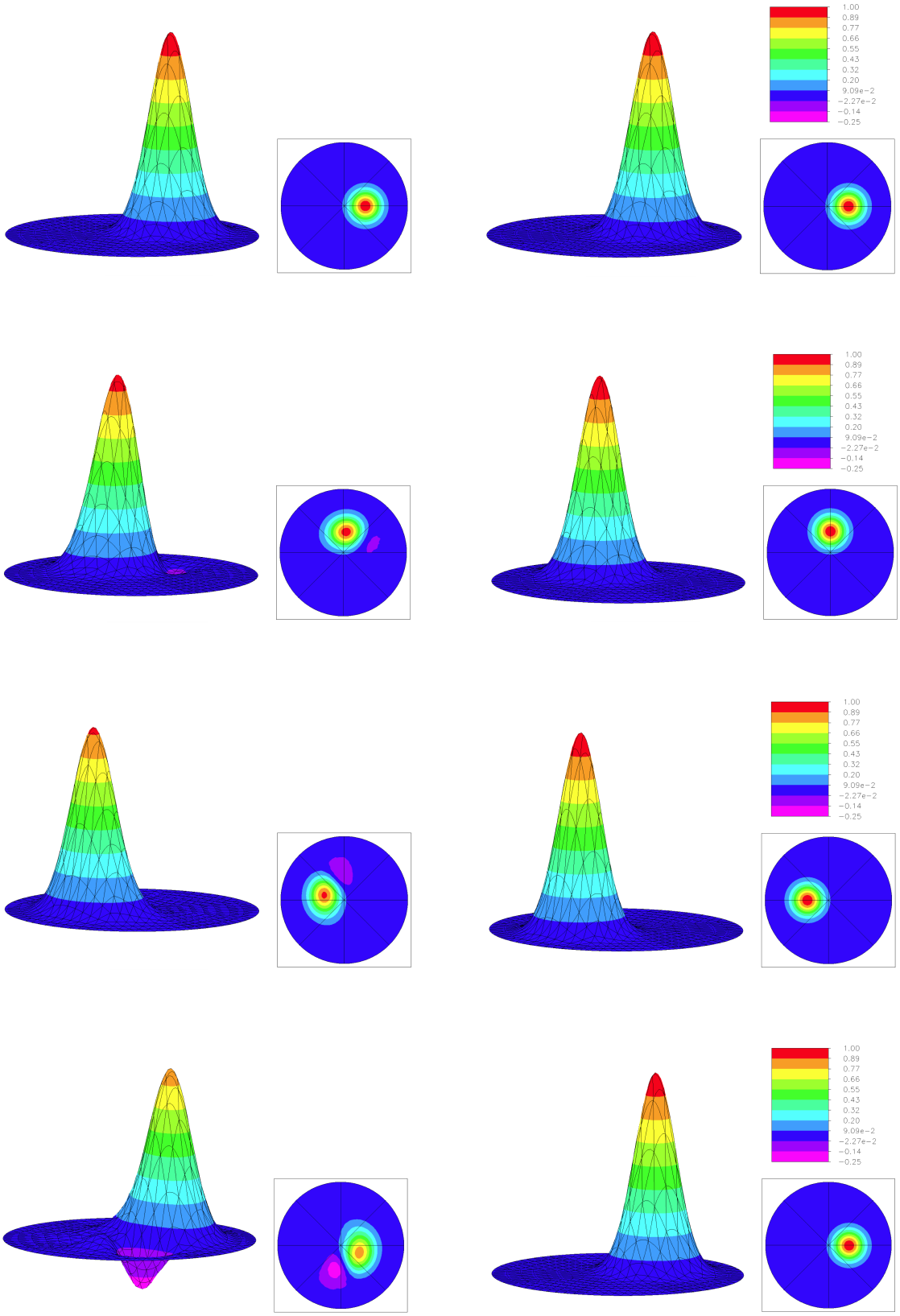

Fig. 4.6. The computed solutions at various times are plotted on a mesh with $\Delta x = 3/10$, $\Delta t = 1/12$, and $\varepsilon = 2 - \sqrt{2}$. The solution computed on a static mesh is on the left with its level-sets and the moving mesh solution is on the right with its level sets. From top to bottom, the plots show the initial conditions at $t = 0$, the solutions at $t = 1/4$, the solutions at $t = 1/2$, and the solutions at $t = 1$. The computed solution with $\varepsilon = 2/3$ as required by Theorem 3.1 are similar. The illustrated solutions show a more accurate solution computed on the mesh that offsets the convection term.