Journal of Computational Physics **159,** 274–289 (2000)
doi:10.1006/jcph.2000.6436, available online at http://www.idealibrary.com on IDEAL®

A Crank–Nicolson Type Space–Time Finite Element Method for Computing on Moving Meshes

Peter Hansbo

Department of Solid Mechanics, Chalmers University of Technology, Göteborg, Sweden
E-mail: hansbo@solid.chalmers.se

Received June 2, 1999; revised December 27, 1999

In this paper, a space–time finite element method for evolution problems that is second-order accurate in both space and time is proposed. For convection dominated problems, the elements may be aligned along the characteristics in space–time, which results in a Crank–Nicolson method along the characteristics. The method is also suitable as an alternative to other moving mesh methods for problems in deforming domains. Numerical examples dealing with diffusion and convection are given.

Key Words: Crank–Nicolson; space–time finite element; method of characteristics.

1. INTRODUCTION

In previous work [6–9], I have suggested the use of a space–time finite element method for convection–diffusion problems based on a time-Discontinuous Galerkin (DG) method of Jamet [12]. This version of the DG method, known as the Characteristic Streamline Diffusion (CSD) method, consists of aligning the element sides along the characteristic directions in space–time. It has been shown to have some good points from both a theoretical [14] and a practical [6] point of view. For an introduction to the CSD method, see the monograph [5, Chapter 19]. Essentially the same method has been independently proposed by Tezduyar and coworkers, who have used this approach in large-scale computations (see, e.g., [17, 18]).

However, the CSD method has a drawback: the use of the lowest possible order of temporal approximation leads to a variant of the backward Euler method along the characteristic direction, meaning that it is only a first-order method. While a first-order method along the characteristics may outperform a standard high-order method in many cases, it is in a general situation insufficient. Higher order polynomial approximation, leading to variants of the first subdiagonal Padé approximants, is prohibitively expensive to use. For this reason I instead suggest the use of continuous Galerkin methods, which are related to the diagonal

0021-9991/00 $35.00

Padé approximants. I shall focus on the lowest-order continuous Galerkin method, which is a Crank–Nicolson-type method, i.e., a one-step method of second order.

It must be noted that, while the exposition of the method relies heavily on the notion of characteristics, the method is not limited to convection–diffusion-type problems. It is also suited to handling problems requiring deforming regions (e.g., large deformation problems, arbitrary Lagrangian–Eulerian problems, Stefan problems, flow problems with moving parts, and interface tracking in general). The movement of the mesh does not have to be connected to a physical velocity field.

Second-order time-accurate space–time finite element methods on moving meshes have been proposed earlier by Bonnerot and Jamet [2, 3]. Their approach is close to the one presented here in special cases; however, as far as I am aware, this is the first attempt at a general approach which is independent of quadrature and allows for higher order approximations of both the mesh movement velocity and the polynomial degree of approximation in both space and time.

An outline of this paper is as follows: in Section 2 a linear model problem is introduced, with the purpose of defining the method and its qualities. In Section 3, the space–time finite element method introduced by Aziz and Monk [1] for the solution of the heat equation is recalled. In Section 4 the method is defined and the effects of space–time orientation of the elements are discussed. In Section 6 the practical implementation is discussed. In Section 7 some numerical examples are presented, and in Section 8 some concluding remarks are given.

2. PROBLEM FORMULATION

While the present method is applicable to any type of evolution problem, I will, for the purpose of introduction, focus on the linear convection–diffusion problem. The reason for this is simply that the question of how to move the nodes is not an issue in this case; the natural choice is to let them move with the flow. For other problems, nodal motion may be an issue in itself, cf. [9]. Convection–diffusion also displays some of the computational difficulties encountered in more complex flow problems.

Consider thus first the numerical solution of the following model problem:

$$\left.\begin{aligned} \frac{\partial u}{\partial t} + \mathbf{c}\cdot\nabla u - \nabla\cdot(\varepsilon\nabla u) &= f && \text{in } Q \equiv \Omega \times I, \\ u &= 0 && \text{on } \partial\Omega \times I, \\ u(\mathbf{x}, 0) &= u_0 && \text{in } \Omega, \end{aligned}\right\} \tag{1}$$

where Ω is a polygonal domain in $\mathcal{R}^2$ with boundary $\partial\Omega$, $I = (0, T)$ is a given time interval, and $\mathbf{c}$ and $\varepsilon > 0$ are functions of $(\mathbf{x}, t)$ representing a given convection velocity and diffusion coefficient, respectively. To avoid the complications associated with time-variable space domain, it is also assumed that

$$\mathbf{c} = 0 \qquad \text{on } \partial\Omega. \tag{2}$$

(This is not a limitation of the method, however; for computations with moving boundaries using CSD, see, e.g., [7, 8].) Further, $f(\mathbf{x}, t)$ is a given production term and u_0 is given initial data.

Equation (1) is written in Eulerian coordinates $(\mathbf{x}, t)$, where $\mathbf{x}$ refers to a certain fixed location in $\mathcal{R}^2$ and t is time. In Eulerian coordinates, $u(\mathbf{x}, t)$ is the concentration at the space–time location $(\mathbf{x}, t)$ of a certain quantity convected by the velocity field $\mathbf{c}(\mathbf{x}, t)$. Using Eulerian coordinates, observers are fixed at points $\mathbf{x}$ in $\mathcal{R}^2$ and measure the concentration or velocity at $\mathbf{x}$ for different times t. Most of the work done in the field of computational fluid dynamics is based on the use of Eulerian coordinates, but to understand the benefits of the space–time orientation along the characteristics it is helpful to consider instead the Lagrangian coordinates associated with the physics of problem (1).

The characteristics corresponding to the *material derivative,*

$$\frac{\partial}{\partial t} + \mathbf{c} \cdot \nabla,$$

in (1) are space–time curves $(\mathbf{x}, t) = (\mathbf{x}(\mathbf{X}, t), t)$, where $\mathbf{x}(\mathbf{X}, t)$ satisfies

$$\left.\begin{aligned} \frac{\partial \mathbf{x}}{\partial t} &= \mathbf{c}(\mathbf{x}(\mathbf{X}, t), t) \quad t > 0, \\ \mathbf{x}(\mathbf{X}, 0) &= \mathbf{X}. \end{aligned}\right\} \tag{3}$$

The coordinates $\mathbf{x}$ are the Eulerian coordinates and $\mathbf{X}$ are the Lagrangian coordinates, where $\mathbf{X}$ acts as a label identifying a particle which at time $t = 0$ is at location $\mathbf{X} \in \mathcal{R}^2$ and moves according to the velocity field $\mathbf{c}(\mathbf{x}(\mathbf{X}, t), t)$ so that its position at time t is given by $\mathbf{x}(\mathbf{X}, t)$. Alternatively, one may simply write $\mathbf{x} = F(\mathbf{X}, t)$, where it is understood that the mapping F is given implicitly by (3).

Notice that, defining $\bar{u}(\mathbf{X}, t) \equiv u(\mathbf{x}(\mathbf{X}, t), t)$, by (3) and the chain rule,

$$\frac{\partial \bar{u}}{\partial t} = \frac{\partial u}{\partial t} + \mathbf{c} \cdot \nabla u. \tag{4}$$

Thus, the convection equation,

$$\frac{\partial u}{\partial t} + \mathbf{c} \cdot \nabla u = f, \tag{5}$$

in the Eulerian coordinates $\mathbf{x}$, corresponding to taking $\varepsilon = 0$ in (1), takes the simple form,

$$\frac{\partial \bar{u}}{\partial t} = \bar{f}, \tag{6}$$

in the Lagrangian coordinates $(\mathbf{X}, t)$, where $\bar{f}(\mathbf{X}, t) = f(\mathbf{x}(\mathbf{X}, t), t)$. Consequently, in global Lagrangian coordinates the convection term disappears and the original partial differential equation (5) reduces to a set of first-order ordinary differential equations with respect to t indexed by $\mathbf{X}$. This fact is the driving force behind many numerical schemes for fluid mechanics problems. Most methods use the characteristics in an indirect manner, while others, like the finite element method to be presented, are more directly linked to (6). Other examples of a direct incorporation of characteristics include the methods proposed by Pironneau [16] and by Hasbani *et al.* [10] and also the characteristic Galerkin method of Morton [15] and the semi-Lagrangian method of Côté and Staniforth [4]. A difference between these methods and the continuous Galerkin method is that the latter is based on a finite element discretization of both space and time, while the former methods are based on finite difference stencils along the characteristics that do not a priori take into account the fact that the characteristics may only be approximate.

3. THE TIME-CONTINUOUS GALERKIN METHOD

First, the method will be described using a traditional nonoriented space–time mesh. For discretization in space and time, a continuous Galerkin method in space is employed in combination with a continuous method in time using an arbitrary polynomial order. The temporal variation of the test functions will be one order lower than that of the trial functions so that if a trial function is denoted v, the corresponding test function can be written $w = \partial v/\partial t$. Aziz and Monk [1] were the first to analyze this method for parabolic partial differential equations. However, the concept has been used earlier by a number of authors (see, e.g., Hulme [11], Winther [19], and Zienkiewicz [20, Chapter 21.2]).

Attention will be focused on the simplest possible variant using a linear approximation in both space and time in two dimensions, but the framework allows for an arbitrary order of approximation. The method is based on a partition $0 = t_0 < t_1 < \cdots < t_N = T$ of the total time interval $(0, T)$ into smaller time intervals, $I_n = (t_{n-1}, t_n)$, of length $k_n = t_n - t_{n-1}$. It is assumed that there is given a fixed subdivision T_h of Ω into triangles, and the space V_h is then defined by

$$V_h(\Omega) = \{v \in C(\bar{\Omega})\colon v \text{ is linear in } \mathbf{x} \text{ on each triangle } K \text{ in } T_h,\ v = 0 \text{ on } \partial\Omega\}.$$

The corresponding space–time mesh on $\Omega \times (0, T)$ is built from space–time "slabs" $S_n = \Omega \times I_n$. Approximate solutions U which are continuous, piecewise linear in space and time are sought. This means that $U|_{\Omega \times I_n}$ belongs to the following space of functions defined on the slab S_n:

$$W_{hn}(S_n) = \{v(\mathbf{x}, t)\colon v \text{ is linear in time and } v \in V_h \text{ for } t \text{ fixed}\}.$$

The space of functions v defined on the whole of the space–time domain is denoted by W_h; i.e.,

$$W_h(Q) = \{v \in C(\bar{Q})\colon v|_{S_n} \in W_{hn}\}.$$

The method proposed in [1] may be formulated as follows: for $n = 1, 2, \ldots$, find $U \in W_{hn}$ such that

$$\int_{S_n} \left(\frac{\partial U}{\partial t} + \mathbf{c} \cdot \nabla U\right) \frac{\partial v}{\partial t}\, \mathrm{d}\Omega\, \mathrm{d}t + \int_{S_n} \varepsilon \nabla U \cdot \nabla \left(\frac{\partial v}{\partial t}\right) \mathrm{d}\Omega\, \mathrm{d}t = \int_{S_n} f \frac{\partial v}{\partial t}\, \mathrm{d}\Omega\, \mathrm{d}t \qquad (7)$$

for all $v \in W_{hn}$.

Now, in spite of the fact that U is continuous in time, it is computed stepwise in the same way as in a one-step finite difference method. To ensure the temporal continuity of U, the value $U(t_{n-1})$ is taken from the preceding time step (with U_0 a suitable interpolation of the initial data onto the mesh). This means that there is only one unknown in time, the value at the end of the time step. Since there is in fact only one test function in time ($\partial v/\partial t$ is constant), the right number of equations is obtained. The resulting scheme is closely related to the classic Crank–Nicolson method (using a particular temporal mean value for the right-hand side; see [1] for details).

4. THE SPACE–TIME-ORIENTED CHARACTERISTIC METHOD

Next, the space–time mesh will be defined; a piecewise linear approximation will be used along space–time characteristics (and in space). The purpose of the space–time orientation

is to mimic the use of Lagrangian coordinates. Traditionally, one simply replaces the actual $\mathbf{c}$ by an approximation $\mathbf{c}_h$, followed by computing the convective derivative by differencing along $\mathbf{c}_h$. This means that the numerical solution is by definition convected by $\mathbf{c}_h$, so that, in effect, the original equation has been changed, and a consistency error has been introduced. In the present method, on the other hand, the space–time elements are deformed according to $\mathbf{c}_h$, which has a similar effect, but it is the original equation that is discretized. Thus, there will be no consistency error due to the approximation of the convective velocity.

4.1. *Definition of the Method*

Recall the definition of the global Lagrangian coordinates in (3) and the corresponding map $\mathbf{x} = F(\mathbf{X}, t)$. Let us, for ease of presentation, assume that this map is a bijection for all $t \in I$. One can then proceed as follows: denote by $\hat{Q} = \hat{\Omega} \times \hat{I}$ the image of $\Omega \times I$ under F^{-1}, divide $\hat{Q}$ into slabs $\hat{S}_n$ and let the spaces $V_h(\hat{\Omega})$, $W_{hn}(\hat{S}_n)$, and $W_h(\hat{Q})$ be defined as in the previous section.

For simplicity, I consider only the case when $\mathbf{c}_h \in [V_{\mathbf{h}}(\hat{\Omega})]^2$ is piecewise constant in time. The numerical examples of Section 7 show that the choice of $\mathbf{c}_h$ affects the accuracy of the resulting scheme. *A priori* the nodal positions are known only at the beginning of the time step, but the choice of $\mathbf{c}_h$ as the spatial interpolant of $\mathbf{c}$ at the beginning of the time step is not sufficient to ensure second-order temporal accuracy. As the mesh deforms, the nodes will occupy new positions $\mathbf{x}_0, \mathbf{x}_1, \ldots$, so it is natural to define, on each space–time slab, a local Lagrangian coordinate $\mathbf{X}_n \equiv \mathbf{x}_{n-1}$. A second-order approximation of the deformation $\mathbf{x}(t)$ is given by Heun's method; define

$$\tilde{\mathbf{x}}_n = \mathbf{X}_n + k_n \mathbf{c}(\mathbf{X}_n, t_{n-1})$$

and compute $\mathbf{x}_n$ by

$$\mathbf{x}_n = \mathbf{X}_n + \frac{k_n}{2}(\mathbf{c}(\mathbf{X}_n, t_{n-1}) + \mathbf{c}(\tilde{\mathbf{x}}_n, t_n)).$$

Consequently, a natural choice for $\mathbf{c}_h(\mathbf{X}_n)$ is the spatial interpolant of the velocity field

$$\tilde{\mathbf{c}}(\mathbf{X}_n) = (\mathbf{c}(\mathbf{X}_n, t_{n-1}) + \mathbf{c}(\tilde{\mathbf{x}}_n, t_n))/2, \tag{8}$$

constant over each time step.

To proceed, define on each slab the corresponding local Eulerian coordinate $\mathbf{x}_n$ as the solution of

$$\left.\begin{aligned} \frac{\partial \mathbf{x}_n}{\partial t} &= \mathbf{c}_h(\mathbf{X}_n) \qquad \text{on } \hat{I}_n, \\ \mathbf{x}_n(\mathbf{X}_n, t_{n-1}) &= \mathbf{X}_n, \end{aligned}\right\} \tag{9}$$

so that

$$\mathbf{x}_n = \mathbf{X}_n + \mathbf{c}_h(\mathbf{X}_n)(t - t_{n-1}),$$

and denote the corresponding mapping by $F_n : \hat{S}_n \to S_n \equiv F_n(\hat{S}_n)$. The space of functions in which the approximate solution of (1) is sought is then defined by composition with F_n^{-1},

so that

$$W_{hn}^{\mathbf{c}_h}(S_n) = \left\{v(\mathbf{x}, t) = \hat{v}\left(F_n^{-1}(\mathbf{x}, t)\right) : \hat{v} \in W_{hn}(\hat{S}_n)\right\}$$

and

$$W_h^{\mathbf{c}_h}(Q) = \left\{v \in C(\bar{Q}) : v|_{S_n} \in W_{hn}^{\mathbf{c}_h}\right\}.$$

The method based on the space $W_h^{\mathbf{c}_h}$ can now be formulated as follows: Find $U \in W_h^{\mathbf{c}_h}$ such that for $n = 1, 2, \ldots,$

$$\int_{S_n} \left(\frac{\partial U}{\partial t} + \mathbf{c} \cdot \nabla U\right) D_h v \, \mathrm{d}\Omega \, \mathrm{d}t + \int_{S_n} \varepsilon \nabla U \cdot \nabla (D_h v) \, \mathrm{d}\Omega \, \mathrm{d}t = \int_{S_n} f \, D_h v \, \mathrm{d}\Omega \, \mathrm{d}t \quad (10)$$

for all $v \in W_h^{\mathbf{c}_h}$, where

$$D_h v \equiv \frac{\partial v}{\partial t} + \mathbf{c}_h \cdot \nabla v$$

denotes the approximate material derivative of v. Again, note that this effectively yields only one test function, since $D_h v$ is constant along $\mathbf{c}_h$, and consequently (10) is the natural counterpart to (7). It is important to note that the same idea can be used for *any* choice of $\mathbf{c}_h$, and the method is thus possible to use also on moving grids for other applications (interface tracking, ALE methods, etc.).

The space–time mesh corresponding to the approximative convective velocity consists of elements that are inclined in space–time with slope given by the velocity $\mathbf{c}_h$. Depending on the regularity of the velocity field $\mathbf{c}$, it is possible to maintain matching meshes over a certain length of time, until the mesh is so distorted that this is no longer feasible. When the mesh becomes too distorted, a new triangulation of Ω must be introduced, and an L_2-projection from the old to the new mesh must be performed. Since a second-order temporal accuracy is expected, this may be bad news: too many projections will reduce the order of approximation. However, given some regularity of $\mathbf{c}$, the same number of projections *per unit time* is expected as the time-step length is reduced, which still yields the right asymptotic behavior. This is unlike the traditional characteristics-based finite element methods, where one projection is performed each time step (e.g., [16]).

4.2. *Effects of Space–Time Orientation*

Next, (10) is rewritten in the local Lagrangian coordinates on S_n to see the effect of the orientation. Extending $\mathbf{c}_h$ to S_n by setting $\mathbf{c}_h(\mathbf{x}, t) = \mathbf{c}_h(\mathbf{X})$ if $\mathbf{x} = F_n(\mathbf{X}, t)$, the chain rule yields

$$\begin{aligned}\frac{\partial v}{\partial t} + \mathbf{c} \cdot \nabla v &= \frac{\partial v}{\partial t} + \mathbf{c}_h \cdot \nabla v + (\mathbf{c} - \mathbf{c}_h) \cdot \nabla v \\ &= \frac{\partial \hat{v}}{\partial t} + (\hat{\mathbf{c}} - \hat{\mathbf{c}}_h) \cdot J_n^{-1} \nabla_{\mathbf{x}} \hat{v} \equiv \frac{\partial \bar{v}}{\partial t} + \hat{\alpha} \cdot \nabla_{\mathbf{x}} \hat{v}, \end{aligned} \quad (11)$$

where

$$J_n(\mathbf{X}, t) = \frac{\partial \mathbf{x}}{\partial \mathbf{X}}(\mathbf{X}, t)$$

and

$$\hat{\alpha} = J_n^{-T}(\hat{\mathbf{c}} - \hat{\mathbf{c}}_h).$$

With $\mathbf{c}_h$ selected according to (8), one may expect to have $\hat{\alpha} \leq C(k_n^2 + h_n^2)$ if $\mathbf{c}$ is smooth. Reformulating (10) in $(\mathbf{X}, t)$ coordinates the method takes the following form: For $n = 1, 2, \ldots,$ find $\hat{U} \equiv \hat{U}|_{S_n} \in \hat{W}_h$ such that $\forall \hat{v} \in \hat{W}_h$,

$$\int_{S_n} \left(\frac{\partial \hat{U}}{\partial t} + \hat{\alpha} \cdot \nabla_{\mathbf{x}} \hat{U} \right) \frac{\partial \hat{v}}{\partial t} |J_n| \, \mathrm{d}\hat{\Omega} \, \mathrm{d}t + \int_{S_n} \hat{\varepsilon} \hat{\nabla} U \cdot \hat{\nabla} \left(\frac{\partial \hat{v}}{\partial t} \right) |J_n| \, \mathrm{d}\hat{\Omega} \, \mathrm{d}t = \int_{S_n} \hat{f} \frac{\partial \hat{v}}{\partial t} |J_n| \, \mathrm{d}\hat{\Omega} \, \mathrm{d}t, \tag{12}$$

where $\hat{\nabla} = J_n^{-1} \nabla_{\mathbf{x}}$.

Comparing (10) and (12), it is clear that the effect of using the oriented space–time elements is to transform, on each slab S_n, the original problem with velocity $\mathbf{c}$ to a problem resembling the original, but with small velocity $\hat{\alpha}$. To this problem the method is applied on a tensor–product mesh in $(\mathbf{X}, t)$ coordinates without orientation, corresponding directly to Eq. (7). Through the space–time orientation, the convective term is thus effectively eliminated, which both improves the precision and facilitates the solution of the resulting discrete system. The drawback is that L_2 projections have to be performed at mesh changes.

Remark. The presented method directly extends to higher order approximations for the velocity, by using on S_n an approximate velocity $\mathbf{c}^h$ defined by

$$\mathbf{c}_h(\mathbf{X}, t) = \sum_{m=0}^{r} \left(\frac{t - t_{n-1}}{k_n} \right)^m \mathbf{c}_{hm}(\mathbf{X}),$$

where $\mathbf{c}_{hm}(\mathbf{X}) \in [V_h]^2$, followed by solving (9) using an appropriate method. In this case the characteristics $\mathbf{x}(\mathbf{X}, t)$ are given by

$$\mathbf{x}(\mathbf{X}, t) = \mathbf{X} + \sum_{m=1}^{r+1} \frac{k_n}{m} \left(\frac{t - t_{n-1}}{k_n} \right)^m \mathbf{c}_{hm}(\mathbf{X}).$$

5. COMPARISON WITH THE BONNEROT–JAMET SCHEME

The scheme used by Bonnerot and Jamet [2] was analyzed by Jamet [13] for the heat equation in a particular one-dimensional case. Consider the problem of finding u such that

$$\frac{\partial u}{\partial t} - \frac{\partial^2 u}{\partial x^2} = 0, \tag{13}$$

for $s_1(t) < x < s_2(t)$ with $u(s_1(t), t) = u(s_2(t), t) = 0$ and $u(x, 0) = u^0(x)$. Dividing the spatial domain into finite elements with nodes x_j^n such that $s_1(t^n) = x_0^n < x_1^n < \cdots < x_I^n = s_2(t^n)$,

the Bonnerot–Jamet scheme can be written: Find the sequence $\{u_i^n\}$ such that

$$\begin{aligned}
&\frac{1}{2}\left\{\left(x_{i+1}^{n+1}-x_{i-1}^{n+1}\right)u^{n+1}-\left(x_{i+1}^{n}-x_{i-1}^{n}\right)u_i^n\right\}\\
&\quad-\frac{1}{4}\left\{\left(x_{i+1}^{n+1}-x_{i+1}^{n}\right)\left(u_{i+1}^{n+1}+u_{i+1}^{n}\right)-\left(x_{i-1}^{n+1}-x_{i-1}^{n}\right)\left(u_{i-1}^{n+1}+u_{i-1}^{n}\right)\right\}\\
&\quad-\frac{k_n}{2}\left(\frac{u_{i+1}^n-u_i^n}{x_{i+1}^n-x_i^n}+\frac{u_{i+1}^{n+1}-u_i^{n+1}}{x_{i+1}^{n}-x_i^{n+1}}-\frac{u_i^n-u_{i-1}^n}{x_i^n-x_{i-1}^n}-\frac{u_i^{n+1}-u_{i-1}^{n+1}}{x_i^{n+1}-x_{i-1}^{n+1}}\right)=0, \qquad (14)
\end{aligned}$$

with $u_0^n=u_I^n=0$ and $u_i^0=u^0(x_i^0)$ (see [2]). Comparing with the present approach, the mesh velocity implicit in (14) is given pointwise by

$$c_h(x_i(t))=\frac{x_i^{n+1}-x_i^n}{t^{n+1}-t^n},$$

constant over the time step. Now, $\hat{U}\equiv\hat{U}|_{S_n}\in\hat{W}_h$ is sought, such that

$$\int_{S_n}\left(\frac{\partial\hat{U}}{\partial t}+\hat{\alpha}\frac{\partial\hat{U}}{\partial X}\right)\frac{\partial\hat{v}}{\partial t}|J_n|\,\mathrm{d}X\,\mathrm{d}t+\int_{S_n}\left|J_n^{-1}\right|\frac{\partial\hat{U}}{\partial X}\frac{\partial^2\hat{v}}{\partial X\partial t}\,\mathrm{d}X\,\mathrm{d}t=0, \qquad (15)$$

$\forall\hat{v}\in\hat{W}_h$. Consider now the case of a piecewise linear approximation in space and time. Identifying X with $x(t^n)$ and denoting by $\hat{\phi}_i(X)$ the hat function associated with node i,

$$\hat{\phi}_i(X)=\begin{cases}\frac{X-x_{i-1}^n}{x_i^n-x_{i-1}^n} & x_{i-1}^n\le X\le x_i^n,\\ \frac{x_{i+1}^n-X}{x_{i+1}^n-x_i^n} & x_i^n\le X\le x_{i+1}^n,\end{cases}$$

one can set

$$\hat{U}(X,t)=\sum_{j=i-1}^{i+1}\hat{\phi}_j(X)\left(\frac{t-t^n}{k_n}u_j^{n+1}+\left(1-\frac{t-t^n}{k_n}\right)u_j^n\right)\qquad x_{i-1}^n\le X\le x_{i+1}^n,$$

and to get an equation for node i, one may test with $\partial\hat{v}/\partial t=\hat{\phi}_i(X)$ on the same interval. Next, the integrals in (15) must be evaluated on the elements L left of node i and R right of node i. At node i,

$$\frac{\partial\hat{U}}{\partial t}\left(X_i,t^{n+1}\right)=\frac{\partial\hat{U}}{\partial t}\left(X_i,t^n\right)=\frac{u_i^{n+1}-u_i^n}{t^{n+1}-t^n},$$

and, on the element level,

$$\begin{aligned}
\frac{\partial\hat{U}^R}{\partial X}\left(X_i,t^{n+1}\right)&=\frac{u_{i+1}^{n+1}-u_i^{n+1}}{x_{i+1}^n-x_i^n}, & \frac{\partial\hat{U}^R}{\partial X}\left(X_i,t^n\right)&=\frac{u_{i+1}^n-u_i^n}{x_{i+1}^n-x_i^n}.\\
\frac{\partial\hat{U}^L}{\partial X}\left(X_i,t^{n+1}\right)&=\frac{u_i^{n+1}-u_{i-1}^{n+1}}{x_i^n-x_{i-1}^n}, & \frac{\partial\hat{U}^L}{\partial X}\left(X_i,t^n\right)&=\frac{u_i^n-u_{i-1}^n}{x_i^n-x_{i-1}^n}.
\end{aligned}$$

The Jacobian of transformation can be written on the element level as

$$J_n^R=1+(t-t^n)\frac{c_h(x_{i+1}(t))-c_h(x_i(t))}{x_{i+1}^n-x_i^n},$$

so that $J_n^R(X_i, t^n) = 1$ and

$$J_n^R\big(X_i, t^{n+1}\big) = 1 + (t^{n+1} - t^n)\frac{\frac{x_{i+1}^{n+1} - x_{i+1}^n}{t^{n+1}-t^n} - \frac{x_i^{n+1} - x_i^n}{t^{n+1}-t^n}}{x_{i+1}^n - x_i^n} = \frac{x_{i+1}^{n+1} - x_i^{n+1}}{x_{i+1}^n - x_i^n}.$$

Similarly,

$$J_n^L\big(X_i, t^{n+1}\big) = \frac{x_i^{n+1} - x_{i-1}^{n+1}}{x_i^n - x_{i-1}^n}.$$

Further,

$$\hat{\alpha}^L\big(X_i, t^n\big) = \hat{\alpha}^R\big(X_i, t^n\big) = -\frac{x_i^{n+1} - x_i^n}{t^{n+1} - t^n},$$

and

$$\hat{\alpha}^R\big(X_i, t^{n+1}\big) = -\frac{x_i^{n+1} - x_i^n}{t^{n+1} - t^n}\frac{x_{i+1}^n - x_i^n}{x_{i+1}^{n+1} - x_i^{n+1}},$$
$$\hat{\alpha}^L\big(X_i, t^{n+1}\big) = -\frac{x_i^{n+1} - x_i^n}{t^{n+1} - t^n}\frac{x_i^n - x_{i-1}^n}{x_i^{n+1} - x_{i-1}^{n+1}}.$$

To obtain a scheme comparable to the Bonnerot–Jamet scheme, the integrals in (15) will be evaluated using nodal quadrature on the space–time quadrilaterals. One then finds that

$$\int_{S_n} \frac{\partial \hat{U}}{\partial t}\hat{\phi}_i(X) J_n \,\mathrm{d}X\,\mathrm{d}t \approx \frac{u_i^{n+1} - u_i^n}{4}\big(\big(x_{i+1}^n - x_{i-1}^n + x_{i+1}^{n+1} - x_{i-1}^{n+1}\big)\big),$$
$$\int_{S_n} \hat{\alpha}\frac{\partial \hat{U}}{\partial X}\hat{\phi}_i(X) J_n \,\mathrm{d}X\,\mathrm{d}t \approx -\frac{1}{4}\big(x_i^{n+1} - x_i^n\big)\big(u_{i+1}^n - u_{i-1}^n + u_{i+1}^{n+1} - u_{i-1}^{n+1}\big),$$

and

$$\int_{S_n} \frac{\partial \hat{U}}{\partial X}\frac{\partial \hat{\phi}_i(X)}{\partial X}\frac{1}{J_n}\mathrm{d}X\mathrm{d}t$$
$$\approx -\frac{k_n}{2}\left(\frac{u_{i+1}^n - u_i^n}{x_{i+1}^n - x_i^n} - \frac{u_i^n - u_{i-1}^n}{x_i^n - x_{i-1}^n} + \frac{u_{i+1}^{n+1} - u_i^{n+1}}{x_{i+1}^{n+1} - x_i^{n+1}} - \frac{u_i^{n+1} - u_{i-1}^{n+1}}{x_i^{n+1} - x_{i-1}^{n+1}}\right).$$

In conclusion, the new scheme can be written: Find the sequence $\{u_i^n\}$ such that

$$\frac{1}{4}\big(\big(x_{i+1}^n - x_{i-1}^n + x_{i+1}^{n+1} - x_{i-1}^{n+1}\big)\big)\big(u_i^{n+1} - u_i^n\big)$$
$$-\frac{1}{4}\big(x_i^{n+1} - x_i^n\big)\big\{\big(u_{i+1}^{n+1} + u_{i+1}^n\big) - \big(u_{i-1}^{n+1} + u_{i-1}^n\big)\big\}$$
$$-\frac{k_n}{2}\left(\frac{u_{i+1}^n - u_i^n}{x_{i+1}^n - x_i^n} + \frac{u_{i+1}^{n+1} - u_i^{n+1}}{x_{i+1}^{n+1} - x_i^{n+1}} - \frac{u_i^n - u_{i-1}^n}{x_i^n - x_{i-1}^n} - \frac{u_i^{n+1} - u_{i-1}^{n+1}}{x_i^{n+1} - x_{i-1}^{n+1}}\right) = 0, \quad (16)$$

which is close, but not identical, to the Bonnerot–Jamet scheme (14). Note, however, that with $x_i^{n+1} = x_i^n$, $x_{i+1}^{n+1} = x_{i+1}^n = x_i^n + h$ and $x_{i-1}^{n+1} = x_{i-1}^n = x_i^n - h$, one obtains

$$\frac{u^{n+1} - u^n}{k_n} - \frac{1}{2}\left(\frac{u_{i+1}^n - 2u_i^n + u_{i-1}^n}{h^2} + \frac{u_{i+1}^{n+1} - 2u_i^{n+1} + u_{i-1}^{n+1}}{h^2}\right) = 0,$$

which is the Crank–Nicolson scheme.

6. PRACTICAL IMPLEMENTATION

In this section, the practical implementations of the method are discussed in the case of a linear approximation in space and time. Clearly, the method can be put in a standard finite element format where the trial functions are isoparametrically mapped in space–time, and the test functions are superparametrically mapped. To simplify matters, consider instead approximations written in the form

$$U = \sum_i \left(U_i^{n-1}\varphi_i(\mathbf{x}, t) + U_i^*\psi_i(\mathbf{x}, t)\right),$$

where the sum is taken over all of the nodes. Here, $\varphi_i(\mathbf{x}, t)$ is chosen piecewise constant in time along the deforming mesh, and $\psi_i(\mathbf{x}, t)$ piecewise linear in time. Thus, U^* signifies the increase in the solution along the deformed mesh, and consequently $U_i^n = U_i^{n-1} + U_i^*$. In the standard way, at time $t = t_{n-1}$ an elementwise approximation is defined on the element with local node coordinates (X_1, Y_1), (X_2, Y_2), and (X_3, Y_3). With

$$\Delta = X_1Y_2 - X_2Y_1 + X_2Y_3 - X_3Y_2 + X_3Y_1 - X_1Y_3,$$

the shape functions can be written

$$\begin{aligned}
\hat{\varphi}_1 &= (X_2Y_3 - X_3Y_2 + (Y_2 - Y_3)X + (X_3 - X_2)Y)/\Delta \\
\hat{\varphi}_2 &= (X_3Y_1 - X_1Y_3 + (Y_3 - Y_1)X + (X_1 - X_3)Y)/\Delta \\
\hat{\varphi}_3 &= (X_1Y_2 - X_2Y_1 + (Y_1 - Y_2)X + (X_2 - X_1)Y)/\Delta,
\end{aligned}$$

and $\hat{\psi}_i = (t - t_{n-1})\hat{\varphi}_i/k_n$. By definition of the mapping F, $D_h\varphi_i = 0$ and $D_h\psi_i = \varphi_i/k_n$. Furthermore, since U is taken as time-continuous, U^{n-1} is the known solution at the beginning of the time-step, and, on each space–time slab, the discrete problem can be written

$$(\mathbf{M}_* + \mathbf{A}_*)\mathbf{u}^* = \mathbf{f} - \mathbf{A}_{n-1}\mathbf{u}^{n-1},$$

where $(\mathbf{u}^*)_i = U_i^*$, $(\mathbf{u}^{n-1})_i = U_i^{n-1}$,

$$\begin{aligned}
(\mathbf{M}_*)_{ij} &= \frac{1}{k_n^2}\int_{S_n} \varphi_j\varphi_i \,\mathrm{d}\Omega\,\mathrm{d}t, \\
(\mathbf{A}_*)_{ij} &= \frac{1}{k_n}\int_{S_n} \frac{t - t_{n-1}}{k_n}((\mathbf{c} - \mathbf{c}_h)\cdot\nabla\varphi_j\varphi_i + \varepsilon\nabla\varphi_j\cdot\nabla\varphi_i)\,\mathrm{d}\Omega\,\mathrm{d}t, \\
(\mathbf{A}_{n-1})_{ij} &= \frac{1}{k_n}\int_{S_n} ((\mathbf{c} - \mathbf{c}_h)\cdot\nabla\varphi_j\varphi_i + \varepsilon\nabla\varphi_j\cdot\nabla\varphi_i)\,\mathrm{d}\Omega\,\mathrm{d}t,
\end{aligned}$$

and

$$(\mathbf{f})_i = \frac{1}{k_n} \int_{S_n} f \varphi_i \, \mathrm{d}\Omega \, \mathrm{d}t.$$

Note that, due to the definition of F, φ_i is spatially continuous for all t, and if $\mathbf{c} = \mathbf{c}_h$ both $\mathbf{A}_*$ and $\mathbf{A}_n$ are symmetric matrices. Furthermore, the formulation of the discrete problem is independent of the temporal approximation used for $\mathbf{c}_h$.

There now remains to compute the gradients of the shape functions according to (11),

$$\begin{bmatrix} \frac{\partial \varphi}{\partial x} \\ \frac{\partial \varphi}{\partial y} \end{bmatrix} = \begin{bmatrix} 1 + (t - t_{n-1}) \frac{\partial c_h^X}{\partial X} & (t - t_{n-1}) \frac{\partial c_h^Y}{\partial X} \\ (t - t_{n-1}) \frac{\partial c_h^X}{\partial Y} & 1 + (t - t_{n-1}) \frac{\partial c_h^Y}{\partial Y} \end{bmatrix}^{-1} \begin{bmatrix} \frac{\partial \hat{\varphi}}{\partial X} \\ \frac{\partial \hat{\varphi}}{\partial Y} \end{bmatrix},$$

which can be done analytically since $\mathbf{c}_h = (c_h^X, c_h^Y)$ is a mesh function.

7. NUMERICAL EXAMPLES

7.1. *Diffusion on a Deforming Grid*

Consider the diffusion equation,

$$\frac{\partial u}{\partial t} - \frac{1}{5} \nabla^2 u = 0,$$

in $\Omega = (0, 1) \times (0, 1)$, $u = 0$ on $\partial\Omega$. The diffusion problem is solved on a deforming grid, with mesh velocity

$$\mathbf{c}(\mathbf{X}) = 100(-(Y - 1/2)(1 - X)X(1 - Y)Y, (X - 1/2)(1 - X)X(1 - Y)Y,$$

so that the velocity is associated with nodes rather than with the nodal positions on the deformed mesh. The computational mesh is shown in Fig. 1. The initial solution was $u^0 = 1 + \cos(4\pi r)$ if $r \leq 1/4$, $u^0 = 0$ elsewhere, where $r = ((x - 1/2)^2 + (y - 1/2)^2)^{1/2}$.

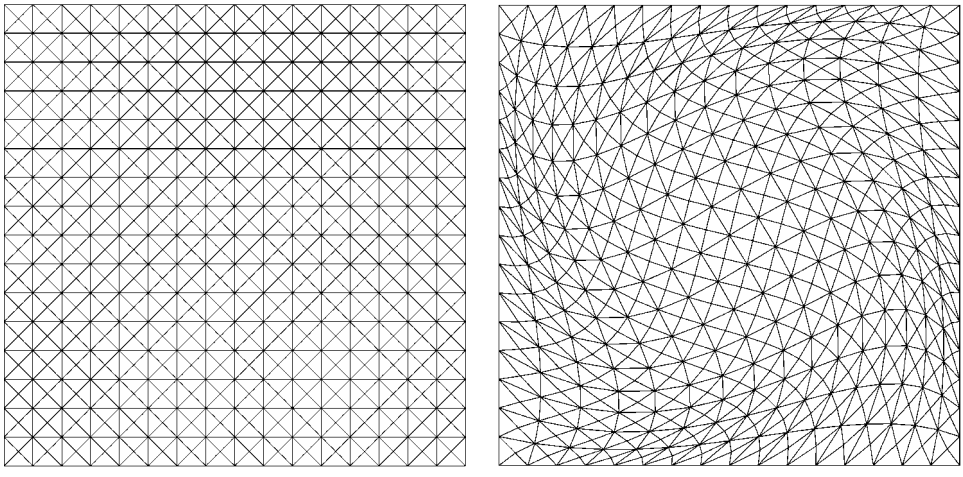

FIG. 1. Initial mesh and maximally deformed mesh.

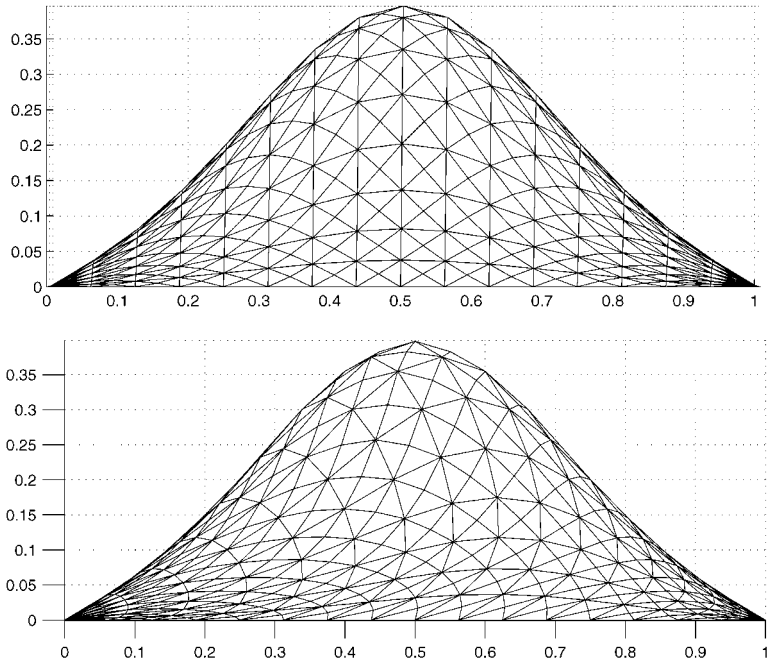

FIG. 2. Solution on a fixed and on a deforming mesh.

This problem was solved over the time span $T = (0, 0.1)$ using 1, 2, 4, 8, and 16 time steps, and the "exact" solution was obtained using 60 time steps. Elevations of the solution are given for a fixed mesh computation and a computation on the deforming mesh in Fig 2, and the convergence behaviors on the fixed and deforming meshes are shown in Fig. 3.

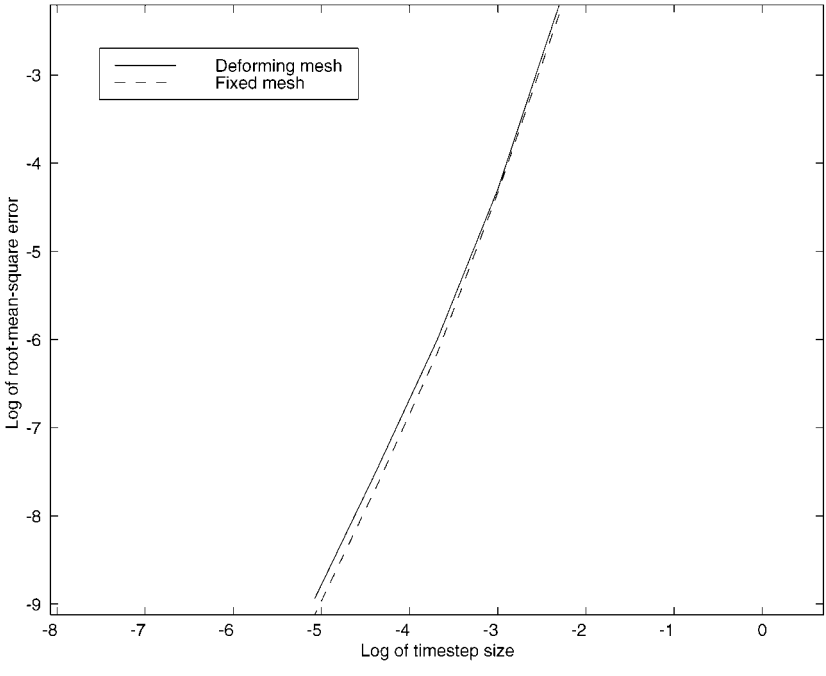

FIG. 3. Convergence rates on the fixed and deforming meshes.

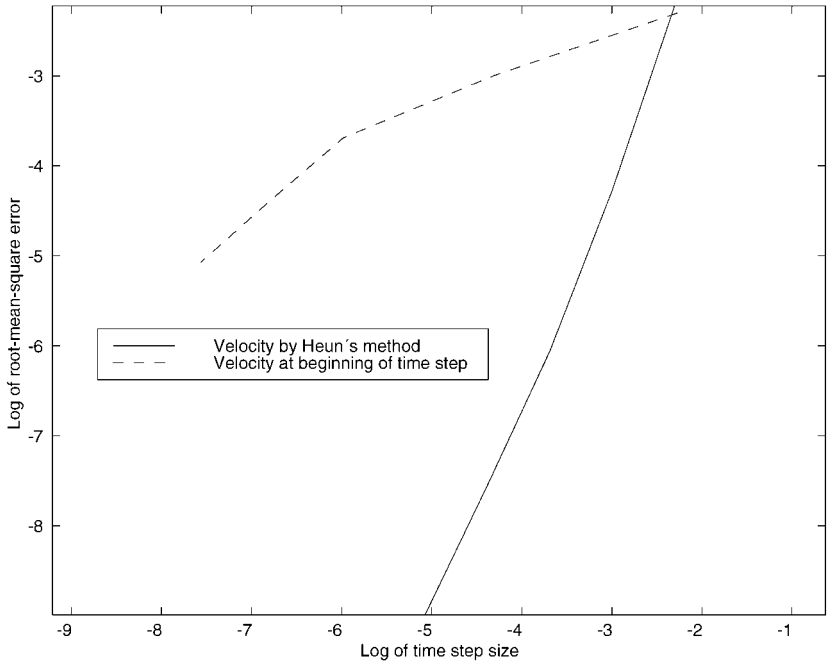

FIG. 4. Convergence for convection–diffusion on a deforming mesh.

7.2. *Convection–Diffusion on a Deforming Grid*

This problem is similar to the previous example, the difference being that there is now also a convective term:

$$\frac{\partial u}{\partial t} + \mathbf{c} \cdot \nabla u - \frac{1}{5}\nabla^2 u = 0.$$

The domain, boundary conditions, and initial solution are the same as in Example 7.1. The same number of time steps was taken, and the exact solution was obtained in the same way. Now, however, the velocity field was given by

$$\mathbf{c}(\mathbf{x}) = 100(-(y - 1/2)(1 - x)x(1 - y)y, (x - 1/2)(1 - x)x(1 - x)x,$$

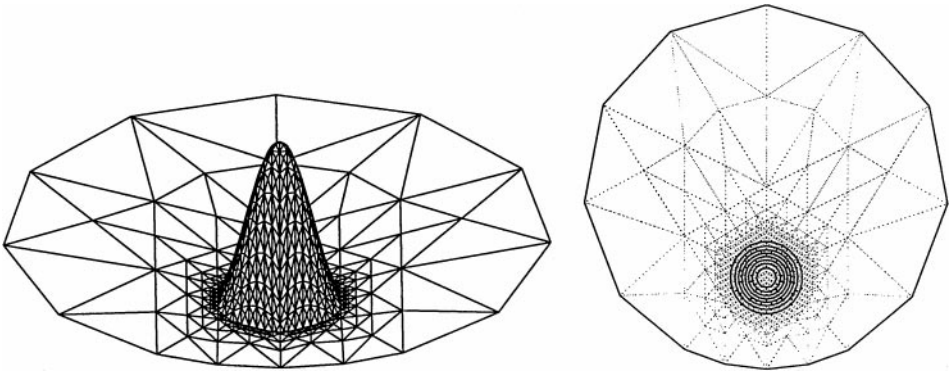

FIG. 5. Initial cosine hill for convection problem.

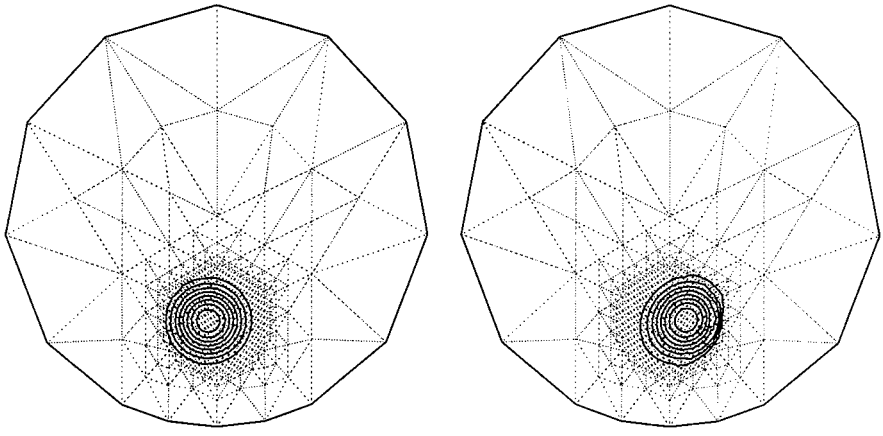

FIG. 6. Solutions after one rotation using the midpoint rule (left) and trapezoidal rule (right) for the temporal integration of the finite element matrices.

defined by the position in space, rather than associated with the nodes. This problem was solved using two different definitions of $\mathbf{c}^h$, one obtained taking the spatial interpolant of the velocity at the beginning of each time step and the other using Heun's method (8). In order to preserve the second-order convergence behavior, the second choice is crucial, as seen in Fig. 4.

7.3. *Convection on a Deforming Grid*

In this example, the influence of the temporal integration procedure upon accuracy is shown. The problem is in a sense dual to the previous example: the mesh rotates with the exact velocity (corresponding to solving for the characteristics exactly), whereas the convective velocity *in the equation* is being evaluated numerically.

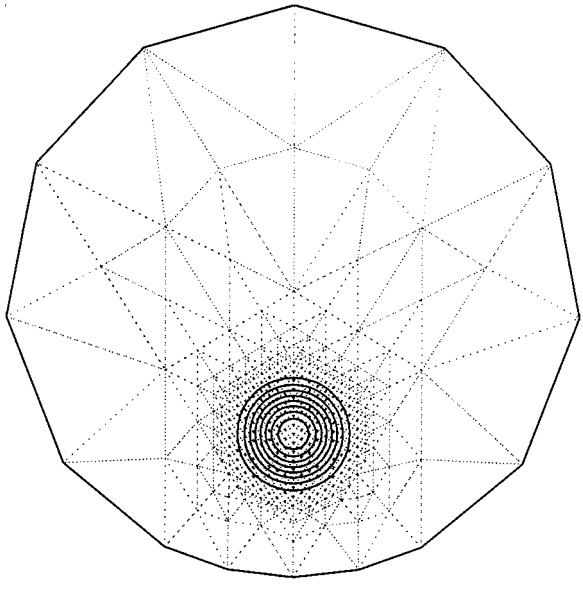

FIG. 7. Solution after one rotation using two-point Gaussian quadraure for the temporal integration of the finite element matrices.

With the origin in the center of a unit disc, and $r = (x^2 + (y - 1/2)^2)^{1/2}$, the initial solution is $u = 1 + \cos 4\pi r$ for $r \leq 0.22$, $u = 0$ elsewhere. This hill is being convected in a flowfield $\mathbf{c} = (-y, x)$, and the number of time-steps used to complete one full rotation is $N = 16$. In Fig. 5, the initial solution and the computational mesh is shown. In Fig. 6, the solution after one rotation using the midpoint rule (left) and the trapeziodal rule (right) for the temporal integration of the finite element matrices are shown. With such a small number of time steps, these second-order accurate time integration schemes clearly do not suffice to give an accurate answer (a first-order accurate integration method would just yield noise). Finally, in Fig. 7, two Gauss points have been used, and the solution is very close to the initial. This serves to show that the mesh velocity and the velocity in the equation are separate things. Both have to be evaluated to second-order accuracy in order to achieve the right asymptotics, which explains the convergence behavior in Example 7.2.

8. CONCLUDING REMARKS

In this paper, a variant of the continuous finite element method of Aziz and Monk [1] suitable for computation on deforming meshes has been introduced. The method is an extension of the Crank–Nicolson method, and computational experience confirms that the method retains the second-order temporal accuracy of the Crank–Nicolson method. The exact conditions necessary for second order accuracy with regards to the deformation of the mesh remains to be investigated.

The interest in space–time finite element methods for flows in deforming domains is motivated by the fact that it allows for mimicking the use of Lagrangian coordinates without introducing actual differencing along the characteristics. This means that if the velocity field used to convect the mesh is different from the actual velocity field (which is the usual situation), there will remain a small residual convective term. The resulting nonsymmetry of the corresponding matrix problem is the price paid for consistency; our method will work also if it is difficult to align the mesh along the characteristics. Furthermore, by coupling the approximation of the solution to the approximation of the flow field, there is an immediate gain when the method is applied to nonlinear flow problems where the flow field is identical to the solution, as, e.g., in the Navier–Stokes equations. Clearly, in such a case the residual convective term becomes identical to zero by definition, which *automatically* reduces (the discretized) Navier–Stokes to Stokes without further approximations, cf. [7, 8].

REFERENCES

1. A. K. Aziz and P. Monk, Continuous finite elements in space and time for the heat equation, *Math. Comp.* **52**, 255 (1989).
2. R. Bonnerot and P. Jamet, A second order finite element method for the one-dimensional Stefan problem, *Int. J. Numer. Methods Eng.* **8**, 811 (1974).
3. R. Bonnerot and P. Jamet, Numerical computation of the free boundary for the two-dimensional Stefan problem by space-time elements, *J. Comput. Phys.* **25**, 163 (1977).
4. J. Côté and A. Staniforth, A two time-level semi-Lagrangian semi-implicit scheme for spectral models, *Mon. Weather Rev.* **116**, 2003–2012 (1988).
5. K. Eriksson, D. Estep, P. Hansbo, and C. Johnson, *Computational Differential Equations* (Cambridge Univ. Press, Cambridge, UK, 1996).
6. P. Hansbo, The characteristic streamline diffusion method for convection-diffusion problems, *Comput. Methods Appl. Mech. Eng.* **96**, 239 (1992).

7. P. Hansbo, The characteristic streamline diffusion method for the time-dependent incompressible Navier–Stokes equations, *Comput. Methods Appl. Mech. Eng.* **99**, 171 (1992).
8. P. Hansbo, Lagrangian incompressible flow computations in three dimensions by use of space–time finite elements, *Int. J. Numer. Methods Fluids* **20**, 989 (1995).
9. P. Hansbo, Moving finite element methods by use of space–time elements. I. Scalar problems, *Numer. Methods Partial Differ. Eq.* **14**, 251 (1998).
10. Y. Hasbani, E. Livne, and M. Bercovier, Finite elements and characteristics applied to advection–diffusion equations, *Comput. Fluids* **11**, 71 (1983).
11. B. L. Hulme, One-step piecewise polynomial Galerkin methods for initial value problems, *Math. Comput.* **26**, 415 (1972).
12. P. Jamet, Galerkin-type approximations which are discontinuous in time for parabolic problems on a variable domain, *SIAM J. Numer. Anal.* **15**, 912 (1978).
13. P. Jamet, Stability and convergence of a generalized Crank–Nicolson scheme on a variable mesh for the heat equation, *SIAM J. Numer. Anal.* **17**, 530 (1980).
14. C. Johnson, A new approach to algorithms for convection problems which are based on exact transport + projection, *Comput. Methods Appl. Mech. Eng.* **100**, 45 (1992).
15. K. W. Morton, Generalised Galerkin methods for hyperbolic problems, *Comput. Methods Appl. Mech. Eng.* **52**, 847 (1985).
16. O. Pironneau, On the transport–diffusion algorithm and its application to the Navier–Stokes equations, *Numer. Math.* **38**, 309 (1982).
17. O. Pironneau, J. Liou, and T. E. Tezduyar, Characteristic Galerkin and Galerkin/least-aquares space-time formulations for the advection–diffusion equation with time-dependent domains, *Comput. Methods Appl. Mech. Eng.* **100**, 117 (1992).
18. T. E. Tezduyar, M. Behr, and J. Liou, A new strategy for finite element computations involving moving boundaries and interfaces—The deforming-spatial-domain/space–time procedure. I. The concept and the preliminary numerical tests, *Comput. Methods Appl. Mech. Eng.* **94**, 339 (1992).
19. R. Winther, A stable finite element method for initial-boundary value problems for first order hyperbolic systems, *Math. Comp.* **36**, 45 (1981).
20. O. C. Zienkiewicz, *The Finite Element Method* (McGraw-Hill, London, 1977), 3rd ed.