Interfaces and Free Boundaries **22** (2020), 443–464
DOI 10.4171/IFB/446

A convergent algorithm for forced mean curvature flow driven by diffusion on the surface

BALÁZS KOVÁCS
Mathematisches Institut, Universität Tübingen,
Auf der Morgenstelle 10, 72076 Tübingen, Germany
E-mail: kovacs@na.uni-tuebingen.de

BUYANG LI
Department of Applied Mathematics, The Hong Kong Polytechnic University,
Hong Kong
E-mail: buyang.li@polyu.edu.hk

CHRISTIAN LUBICH
Mathematisches Institut, Universität Tübingen,
Auf der Morgenstelle 10, 72076 Tübingen, Germany
E-mail: lubich@na.uni-tuebingen.de

[Received 9 September 2019 and in revised form 14 August 2020]

The evolution of a closed two-dimensional surface driven by both mean curvature flow and a reaction–diffusion process on the surface is formulated as a system that couples the velocity law not only to the surface partial differential equation but also to the evolution equations for the normal vector and the mean curvature on the surface. Two algorithms are considered for the obtained system. Both methods combine surface finite elements for space discretization and linearly implicit backward difference formulae for time integration. Based on our recent results for mean curvature flow, one of the algorithms directly admits a convergence proof for its full discretization in the case of finite elements of polynomial degree at least two and backward difference formulae of orders two to five, with optimal-order error bounds. Numerical examples are provided to support and complement the theoretical convergence results (illustrating the convergence behaviour of both algorithms) and demonstrate the effectiveness of the methods in simulating a three-dimensional tumour growth model.

2010 Mathematics Subject Classification: Primary 35R01, 65M60, 65M15, 65M12.

Keywords: Forced mean curvature flow, reaction–diffusion on surfaces, evolving finite element method, linearly implicit, backward difference formula, convergence, tumour growth.

1. Introduction

We consider the numerical approximation of an unknown evolving two-dimensional closed surface $\Gamma(t)$ that is driven by both mean curvature flow and a reaction–diffusion process on the surface, starting from a given smooth initial surface Γ^0. The outer normal velocity V of the surface is determined by the velocity law

$$V = -H + u, \tag{1.1}$$

where H is the mean curvature of the evolving surface, and where $u(x,t)$ ($x \in \Gamma(t)$, $t \in [0,T]$) is the solution of a reaction–diffusion equation on the evolving surface,

$$\partial^\bullet u + u\,\nabla_\Gamma \cdot v - \Delta_\Gamma u = F(u, \nabla_\Gamma u), \tag{1.2}$$

with given initial data u^0. Here, $F : \mathbb{R} \times \mathbb{R}^3 \to \mathbb{R}$ is a given smooth function, and v is the surface velocity: $v = Vn$ with V of (1.1) and the outer normal n. Problem (1.1)–(1.2) can be viewed as *forced* mean curvature flow driven by the solution of the parabolic equation (1.2) on the evolving surface.

While we study the numerical approximation of Problem (1.1)–(1.2) with a scalar parabolic equation for notational simplicity, we remark that the numerical method and its convergence properties extend readily to the case of a *system* of reaction-diffusion equations (1.2) with solution $u = (u_1, \dots, u_m)$ and the velocity law $V = -H + \alpha_1 u_1 + \dots + \alpha_m u_m$ with constant real coefficients α_i. We will encounter such a more general problem in our numerical experiments with a tumour growth model.

Many practical applications concern mean curvature flow coupled with surface partial differential equations (PDEs), for example tumour growth [2, 7–9, 22];surface dissolution [18, 21] (also see [16, Section 10.4]); diffusion induced grain boundary motion [11, 23, 35]. These models all use a velocity law that is linear in u, as in (1.1) or as in the previous paragraph, except for diffusion induced grain boundary motion where $V = -H + u^2$.

Numerical approximations to forced mean curvature flow coupled with surface partial differential equations have been considered in some of these papers. For *curves*, convergence of numerical methods for such coupled problems of forced curve shortening flow was proved in [3, 34].

Numerical approximation to pure mean curvature flow of *surfaces* – i.e., the case $u \equiv 0$ in (1.1) – was first addressed by Dziuk [14], based on a formulation of mean curvature flow as a formally heat-like equation on a surface. He proposed an evolving surface finite element method in which the moving nodes of the finite element mesh determine the approximate evolving surface. Different surface finite element based methods were proposed by Barrett, Garcke & Nürnberg [5] based on different variational formulations, and by Elliott & Fritz [19] based on DeTurck's trick of reparametrizing the surface. However, proving convergence of any of these methods has remained an open problem for the mean curvature flow of closed two-dimensional surfaces.

In [27] we proved the first convergence result for semi- and full discretizations of mean curvature flow of closed surfaces with evolving surface finite elements. Discretizing the coupled system for the velocity law together with evolution equations for the normal vector field and mean curvature, we obtained a method with provable error bounds of optimal order.

To our knowledge, no convergence results have yet been proved for forced mean curvature flow of closed surfaces (1.1)–(1.2). For a *regularized* version of forced mean curvature flow of closed surfaces, optimal-order convergence results for semi- and full discretizations were obtained in [28] and [29], respectively.

In this paper, we extend the approach and techniques of our previous paper [27] to the forced mean curvature flow problem (1.1)–(1.2) as a coupled problem together with evolution equations for the normal vector and mean curvature. These evolution equations, as compared with those for pure mean curvature flow given in [25], contain additional forcing terms depending on u. We present two fully discrete evolving finite element algorithms for the obtained coupled system. The first algorithm discretizes the two terms $\partial^\bullet u + u\,\nabla_{\Gamma[X]} \cdot v$ separately in the spatial discretization by using the velocity law for v and the approach in [27]. The second algorithm combines the two terms

in the spatial discretization by an idea of [15] for treating conservation laws on an evolving surface. Both algorithms use evolving surface finite elements for spatial discretization and linearly implicit backward difference formulae for time integration, and for both algorithms the moving nodes of a finite element mesh determine the approximate evolving surface.

The convergence proof for the forced mean curvature algorithm considered here is a very minor modification compared to the convergence proof for the pure mean curvature algorithm of [27], since that algorithm is already built on coupling evolution equations on the surface to the evolution of the surface. The first algorithm can be written in the same matrix–vector form as the method proposed in [27] for the mean curvature flow. The convergence analysis in [27] applies directly to the present algorithm for forced mean curvature flow as well, except for one term which corresponds to the term $\Delta_\Gamma u$ in the evolution equation for H. The necessary changes to the stability analysis brought about by this term are carried out in detail. Under the assumption that the problem admits a sufficiently regular solution, this yields uniform in time, optimal-order H^1-norm convergence results for the semi- and full discretizations of forced mean curvature flow when using at least quadratic evolving surface FEM and linearly implicit backward difference formulae of order two to five.

For the second algorithm, we indicate how such an optimal-order convergence estimate of the evolving surface finite element semi-discretization can be obtained by combining results of [28] and [27], but we do not carry out the details.

For the velocity law $V = -H + g(u)$ with a nonlinear smooth function g, we expect that convergence of a direct generalization of the algorithms presented in this paper can be shown with a combination of the techniques of [27, 28, 31]. As this would become a nontrivial lengthy extension, it is not worked out here.

Finally, we present numerical experiments to support and complement the theoretical results. We present convergence tests for both algorithms, and also present an experiment with the numerical simulation for a tumour growth model, using the parameters in [2] for the sake of easy comparison.

2. Evolution equations for mean curvature flow driven by diffusion on the surface

2.1 *Basic notions and notation*

We consider the evolving two-dimensional closed surface $\Gamma(t) \subset \mathbb{R}^3$ for times $t \in [0,T]$ as the image

$$\Gamma(t) = \Gamma[X(\cdot,t)] := \big\{X(p,t) : p \in \Gamma^0\big\},$$

of a smooth flow map $X : \Gamma^0 \times [0,T] \to \mathbb{R}^3$ such that $X(\cdot,t)$ is an embedding for every t. Here, Γ^0 is a smooth closed initial surface, and $X(p,0) = p$. When the time t is clear from the context, we drop t in the notation and write for short

$$\Gamma[X] = \Gamma[X(\cdot,t)].$$

In view of the subsequent numerical discretization, it is convenient to think of $X(p,t)$ as the position at time t of a moving particle with label p, and of $\Gamma[X]$ as a collection of such particles.

The *velocity* $v(x,t) \in \mathbb{R}^3$ at a point $x = X(p,t) \in \Gamma(t)$ equals

$$\partial_t X(p,t) = v\big(X(p,t),t\big). \tag{2.1}$$

For a known velocity field v, the position $X(p,t)$ at time t of the particle with label p is obtained by solving the ordinary differential equation (2.1) from 0 to t for a fixed p.

For a function $w(x,t)$ ($x \in \Gamma(t)$, $0 \leqslant t \leqslant T$) we denote the *material derivative* as

$$\partial^\bullet w(x,t) = \frac{\mathrm{d}}{\mathrm{d}t}\, w\big(X(p,t),t\big) \quad \text{for } x = X(p,t).$$

On any regular surface $\Gamma \subset \mathbb{R}^3$, we denote by $\nabla_\Gamma w : \Gamma \to \mathbb{R}^3$ the *tangential gradient* of a function $w : \Gamma \to \mathbb{R}$, and in the case of a vector field $f = (f_1, f_2, f_3)^T : \Gamma \to \mathbb{R}^3$, we let $\nabla_\Gamma f = (\nabla_\Gamma f_1, \nabla_\Gamma f_2, \nabla_\Gamma f_3)$. We thus use the convention that the gradient of f has the gradient of the components as column vectors. We denote by $\nabla_\Gamma \cdot f$ the *surface divergence* of a vector field f on Γ, and by $\Delta_\Gamma w = \nabla_\Gamma \cdot \nabla_\Gamma w$ the *Laplace–Beltrami operator* applied to w; see the review [10] or [17, Appendix A] or any textbook on differential geometry for these notions.

We denote the unit outer normal vector field to Γ by $\nu : \Gamma \to \mathbb{R}^3$. Its surface gradient contains the (extrinsic) curvature data of the surface Γ. At every $x \in \Gamma$, the matrix of the extended Weingarten map,

$$A(x) = \nabla_\Gamma \nu(x),$$

is a symmetric 3×3 matrix (see, e.g., [36, Proposition 20]). Apart from the eigenvalue 0 with eigenvector ν, its other two eigenvalues are the principal curvatures κ_1 and κ_2. They determine the fundamental quantities

$$H := \mathrm{tr}(A) = \kappa_1 + \kappa_2, \qquad |A|^2 = \kappa_1^2 + \kappa_2^2, \tag{2.2}$$

where $|A|$ denotes the Frobenius norm of the matrix A. Here, H is called the *mean curvature* (as in most of the literature, we do not put a factor 1/2).

2.2 *Evolution equations for normal vector and mean curvature*

Forced mean curvature flow driven by diffusion on the surface sets the velocity (2.1) of the surface $\Gamma[X]$ to

$$v = V\nu \qquad \text{with the normal velocity} \qquad V = -H + u, \tag{2.3}$$

where u is the solution of the non-linear reaction–diffusion equation on the surface $\Gamma[X]$ with given initial value u^0,

$$\partial^\bullet u + u\nabla_{\Gamma[X]} \cdot v - \Delta_{\Gamma[X]} u = F(u, \nabla_{\Gamma[X]} u) \quad \text{on} \quad \Gamma[X], \tag{2.4}$$

with a given smooth function $F : \mathbb{R} \times \mathbb{R}^3 \to \mathbb{R}$.

The geometric quantities H and ν on the right-hand side of (2.3) satisfy the following evolution equations, which are modifications of the evolution equations for pure mean curvature flow (i.e., $V = -H$) as derived by Huisken [25].

Lemma 2.1 *For a regular surface $\Gamma[X]$ moving under forced mean curvature flow (2.3), the normal vector and the mean curvature satisfy*

$$\partial^\bullet \nu = \Delta_{\Gamma[X]} \nu + |A|^2 \nu - \nabla_{\Gamma[X]} u, \tag{2.5}$$

$$\partial^\bullet H = \Delta_{\Gamma[X]} H + |A|^2 H - \Delta_{\Gamma[X]} u - |A|^2 u. \tag{2.6}$$

Proof. Using the normal velocity V in the proof of [25, Lemma 3.3], or see also [6, Lemma 2.37], the following evolution equation for the normal vector holds:

$$\partial^\bullet \nu = -\nabla_{\Gamma[X]} V.$$

On any surface Γ, it holds true that (see [17, (A.9)] or [36, Proposition 24])

$$\nabla_{\Gamma[X]} H = \Delta_{\Gamma[X]} \nu + |A|^2 \nu,$$

which, in combination with $V = -H + u$ from (2.3), gives the stated evolution equation for ν.

By revising the proof of [25, Theorem 3.4 and Corollary 3.5], or see [6, Lemma 2.39], with the normal velocity V we obtain

$$\partial^\bullet H = -\Delta_{\Gamma[X]} V - |A|^2 V,$$

which, again with $V = -H + u$ from (2.3), yields the evolution equation for H. □

2.3 *The system of equations used for discretization*

Similarly to [27], collecting the above equations, we have reformulated forced mean curvature flow as the system of semi-linear parabolic equations (2.5)–(2.6) on the surface coupled to the velocity law (2.3) and the surface PDE (2.4). The numerical discretization is based on a weak formulation of (2.3)–(2.6), together with the velocity equation (2.1). For the velocity law (2.3) we use a weak formulation that turns into the standard Ritz projection when restricted to a subspace. The weak formulation reads, with $V = -H + u$ and $A = \nabla_{\Gamma[X]} \nu$,

$$\int_{\Gamma[X]} \nabla_{\Gamma[X]} v \cdot \nabla_{\Gamma[X]} \varphi^v + \int_{\Gamma[X]} v \cdot \varphi^v = \int_{\Gamma[X]} \nabla_{\Gamma[X]} (V\nu) \cdot \nabla_{\Gamma[X]} \varphi^v + \int_{\Gamma[X]} V\nu \cdot \varphi^v \tag{2.7a}$$

$$\int_{\Gamma[X]} \partial^\bullet \nu \cdot \varphi^\nu + \int_{\Gamma[X]} \nabla_{\Gamma[X]} \nu \cdot \nabla_{\Gamma[X]} \varphi^\nu = \int_{\Gamma[X]} |A|^2 \, \nu \cdot \varphi^\nu - \int_{\Gamma[X]} \nabla_{\Gamma[X]} u \cdot \varphi^\nu \tag{2.7b}$$

$$\int_{\Gamma[X]} \partial^\bullet H \, \varphi^H + \int_{\Gamma[X]} \nabla_{\Gamma[X]} H \cdot \nabla_{\Gamma[X]} \varphi^H = - \int_{\Gamma[X]} |A|^2 \, V \, \varphi^H + \int_{\Gamma[X]} \nabla_{\Gamma[X]} u \cdot \nabla_{\Gamma[X]} \varphi^H, \tag{2.7c}$$

$$\frac{\mathrm{d}}{\mathrm{d}t} \int_{\Gamma[X]} u \varphi^u + \int_{\Gamma[X]} \nabla_{\Gamma[X]} u \cdot \nabla_{\Gamma[X]} \varphi^u = \int_{\Gamma[X]} F(u, \nabla_{\Gamma[X]} u) \varphi^u, \tag{2.8}$$

for all test functions $\varphi^v \in H^1(\Gamma[X])^3$ and $\varphi^\nu \in H^1(\Gamma[X])^3$, $\varphi^H \in H^1(\Gamma[X])$, and $\varphi^u \in H^1(\Gamma[X])$ with $\partial^\bullet \varphi^u = 0$. Here, we use the Sobolev space $H^1(\Gamma) = \{u \in L^2(\Gamma) : \nabla_\Gamma u \in L^2(\Gamma)\}$. Throughout the paper both the usual Euclidean scalar product for vectors and the Frobenius inner product for matrices (which equals to the Euclidean product using an arbitrary vectorization) are denoted by a dot. This system is complemented with the initial data X^0, ν^0, H^0 and u^0.

An alternative weak formulation of (2.8), which is similar to (2.7b)–(2.7c), is based on

$$\int_{\Gamma[X]} \partial^\bullet u \, \varphi^u + \int_{\Gamma[X]} \nabla_{\Gamma[X]} u \cdot \nabla_{\Gamma[X]} \varphi^u = \int_{\Gamma[X]} F(u, \nabla_{\Gamma[X]} u) \varphi^u - \int_{\Gamma[X]} \big(\nabla_{\Gamma[X]} \cdot v\big) u \varphi^u,$$

for $\varphi^u \in H^1(\Gamma[X])$. Using that $\nabla_\Gamma V \cdot \nu = 0$ and $H = \nabla_{\Gamma[X]} \cdot \nu$ and inserting the velocity law (2.3), we obtain

$$\begin{aligned} \nabla_{\Gamma[X]} \cdot v &= \nabla_{\Gamma[X]} \cdot (V\nu) = (\nabla_{\Gamma[X]} V) \cdot \nu + V(\nabla_{\Gamma[X]} \cdot \nu) = V(\nabla_{\Gamma[X]} \cdot \nu) \\ &= (-H + u) H. \end{aligned}$$

This yields a weak formulation of a similar form as (2.7b) and (2.7c),

$$\int_{\Gamma[X]} \partial^\bullet u\, \varphi^u + \int_{\Gamma[X]} \nabla_{\Gamma[X]} u \cdot \nabla_{\Gamma[X]} \varphi^u = \int_{\Gamma[X]} f(H, u, \nabla_{\Gamma[X]} u) \varphi^u \tag{2.9}$$

for all $\varphi^u \in H^1(\Gamma[X])$, where we set

$$f(H, u, \nabla_{\Gamma[X]} u) = F(u, \nabla_{\Gamma[X]} u) - (-H + u) H u.$$

3. Evolving finite element semi-discretization

3.1 *Evolving surface finite elements*

We formulate the evolving surface finite element (ESFEM) discretization for the velocity law coupled with evolution equations on the evolving surface, following the description in [27, 28], which is based on [13] and [12]. We use a surface approximation consisting of curved elements of polynomial degree k over a flat triangular reference element, which are therefore simply called triangles (even if they are curved), and use continuous piecewise polynomial basis functions of degree k, as defined in [12, Section 2.5].

We triangulate the given smooth initial surface Γ^0 by an admissible family of triangulations $\mathcal{T}_h$ of decreasing maximal element diameter h; see [15] for the notion of an admissible triangulation, which includes quasi-uniformity and shape regularity. For a momentarily fixed h, we denote by $\mathbf{x}^0$ the vector in $\mathbb{R}^{3N}$ that collects all nodes p_j $(j = 1, \dots, N)$ of the initial triangulation. By piecewise polynomial interpolation of degree k, the nodal vector defines an approximate surface Γ_h^0 that interpolates Γ^0 in the nodes p_j of the (curved) triangles of $\mathcal{T}_h$. We will evolve the jth node in time according to an approximation of the ODE (2.1), denoted $x_j(t)$ with $x_j(0) = p_j$, and collect the nodes at time t in a column vector

$$\mathbf{x}(t) \in \mathbb{R}^{3N}.$$

We just write $\mathbf{x}$ for $\mathbf{x}(t)$ when the dependence on t is not important.

By piecewise polynomial interpolation on the plane reference triangle that corresponds to every curved triangle of the triangulation, the nodal vector $\mathbf{x}$ defines a closed surface denoted by $\Gamma_h[\mathbf{x}]$. We can then define globally continuous finite element *basis functions*

$$\phi_i[\mathbf{x}] : \Gamma_h[\mathbf{x}] \to \mathbb{R}, \qquad i = 1, \dots, N,$$

which have the property that on every triangle their pullback to the reference triangle is polynomial of degree k, and which satisfy at the nodes $\phi_i[\mathbf{x}](x_j) = \delta_{ij}$ for all $i, j = 1, \dots, N$. These functions span the finite element space on $\Gamma_h[\mathbf{x}]$,

$$S_h[\mathbf{x}] = S_h(\Gamma_h[\mathbf{x}]) = \operatorname{span}\{\phi_1[\mathbf{x}], \phi_2[\mathbf{x}], \dots, \phi_N[\mathbf{x}]\}.$$

For a finite element function $u_h \in S_h[\mathbf{x}]$, the tangential gradient $\nabla_{\Gamma_h[\mathbf{x}]} u_h$ is defined piecewise on each element.

The discrete surface at time t is parametrized by the initial discrete surface via the map $X_h(\cdot, t) : \Gamma_h^0 \to \Gamma_h[\mathbf{x}(t)]$ defined by

$$X_h(p_h, t) = \sum_{j=1}^{N} x_j(t)\, \phi_j[\mathbf{x}(0)](p_h), \qquad p_h \in \Gamma_h^0,$$

which has the properties that $X_h(p_j,t) = x_j(t)$ for $j = 1,\dots,N$, that $X_h(p_h,0) = p_h$ for all $p_h \in \Gamma_h^0$, and

$$\Gamma_h[\mathbf{x}(t)] = \Gamma[X_h(\cdot,t)] = \big\{X_h(p_h,t) \,:\, p_h \in \Gamma_h^0\big\}.$$

The *discrete velocity* $v_h(x,t) \in \mathbb{R}^3$ at a point $x = X_h(p_h,t) \in \Gamma[X_h(\cdot,t)]$ is given by

$$\partial_t X_h(p_h,t) = v_h\big(X_h(p_h,t),t\big).$$

In view of the transport property of the basis functions [15], $\frac{\mathrm{d}}{\mathrm{d}t}\big(\phi_j[\mathbf{x}(t)](X_h(p_h,t))\big) = 0$, the discrete velocity equals, for $x \in \Gamma_h[\mathbf{x}(t)]$,

$$v_h(x,t) = \sum_{j=1}^{N} v_j(t)\,\phi_j[\mathbf{x}(t)](x) \qquad \text{with } v_j(t) = \dot{x}_j(t),$$

where the dot denotes the time derivative $\mathrm{d}/\mathrm{d}t$. Hence, the discrete velocity $v_h(\cdot,t)$ is in the finite element space $S_h[\mathbf{x}(t)]$, with nodal vector $\mathbf{v}(t) = \dot{\mathbf{x}}(t)$.

The *discrete material derivative* of a finite element function $u_h(x,t)$ with nodal values $u_j(t)$ is

$$\partial_h^\bullet u_h(x,t) = \frac{\mathrm{d}}{\mathrm{d}t} u_h\big(X_h(p_h,t)\big) = \sum_{j=1}^{N} \dot{u}_j(t)\phi_j[\mathbf{x}(t)](x) \quad \text{at} \quad x = X_h(p_h,t).$$

3.2 *ESFEM spatial semi-discretizations*

Now we will describe the semi-discretization of the coupled system using both formulations of the surface PDE.

The finite element spatial semi-discretization of the weak coupled parabolic system (2.7) and (2.9) reads as follows: Find the unknown nodal vector $\mathbf{x}(t) \in \mathbb{R}^{3N}$ and the unknown finite element functions $v_h(\cdot,t) \in S_h[\mathbf{x}(t)]^3$ and $\nu_h(\cdot,t) \in S_h[\mathbf{x}(t)]^3$, $H_h(\cdot,t) \in S_h[\mathbf{x}(t)]$, and $u_h(\cdot,t) \in S_h[\mathbf{x}(t)]$ such that, by denoting $\alpha_h^2 = |\nabla_{\Gamma_h[\mathbf{x}]}\nu_h|^2$ and $V_h = -H_h + u_h$,

$$\int_{\Gamma_h[\mathbf{x}]} \nabla_{\Gamma_h[\mathbf{x}]} v_h \cdot \nabla_{\Gamma_h[\mathbf{x}]}\varphi_h^v + \int_{\Gamma_h[\mathbf{x}]} v_h\cdot\varphi_h^v = \int_{\Gamma_h[\mathbf{x}]} \nabla_{\Gamma_h[\mathbf{x}]}(V_h\nu_h)\cdot\nabla_{\Gamma_h[\mathbf{x}]}\varphi_h^v + \int_{\Gamma_h[\mathbf{x}]} V_h\nu_h\cdot\varphi_h^v \tag{3.1a}$$

$$\int_{\Gamma_h[\mathbf{x}]} \partial_h^\bullet \nu_h\cdot\varphi_h^\nu + \int_{\Gamma_h[\mathbf{x}]} \nabla_{\Gamma_h[\mathbf{x}]}\nu_h\cdot\nabla_{\Gamma_h[\mathbf{x}]}\varphi_h^\nu = \int_{\Gamma_h[\mathbf{x}]} \alpha_h^2\,\nu_h\cdot\varphi_h^\nu - \int_{\Gamma_h[\mathbf{x}]} \nabla_{\Gamma_h[\mathbf{x}]}u_h\cdot\varphi_h^\nu \tag{3.1b}$$

$$\int_{\Gamma_h[\mathbf{x}]} \partial_h^\bullet H_h\,\varphi_h^H + \int_{\Gamma_h[\mathbf{x}]} \nabla_{\Gamma_h[\mathbf{x}]}H_h\cdot\nabla_{\Gamma_h[\mathbf{x}]}\varphi_h^H = -\int_{\Gamma_h[\mathbf{x}]} \alpha_h^2\,V_h\,\varphi_h^H + \int_{\Gamma_h[\mathbf{x}]} \nabla_{\Gamma_h[\mathbf{x}]}u_h\cdot\nabla_{\Gamma_h[\mathbf{x}]}\varphi_h^H \tag{3.1c}$$

and

$$\int_{\Gamma_h[\mathbf{x}]} \partial_h^\bullet u_h\varphi_h^u + \int_{\Gamma_h[\mathbf{x}]} \nabla_{\Gamma[X]}u_h\cdot\nabla_{\Gamma_h[\mathbf{x}]}\varphi_h^u = \int_{\Gamma_h[\mathbf{x}]} f(H_h,u_h,\nabla_{\Gamma_h[\mathbf{x}]}u_h)\varphi_h^u, \tag{3.2}$$

for all $\varphi_h^v \in S_h[\mathbf{x}(t)]^3$, $\varphi_h^\nu \in S_h[\mathbf{x}(t)]^3$, $\varphi_h^H \in S_h[\mathbf{x}(t)]$, and $\varphi_h^u \in S_h[\mathbf{x}(t)]$ with the surface $\Gamma_h[\mathbf{x}(t)] = \Gamma[X_h(\cdot,t)]$ given by the differential equation

$$\partial_t X_h(p_h,t) = v_h\big(X_h(p_h,t),t\big), \qquad p_h \in \Gamma_h^0. \tag{3.3}$$

The initial values for the nodal vector $\mathbf{x}$ are taken as the positions of the nodes of the triangulation of the given initial surface Γ^0. The initial data for v_h, H_h and u_h are determined by Lagrange interpolation of v^0, H^0 and u^0, respectively.

Alternatively, the finite element spatial semi-discretization of the weak coupled parabolic system (2.7) and (2.8) determines the same unknown functions, but, instead of (3.2), the equations (3.1) and the ODE (3.3) are coupled to

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_{\Gamma_h[\mathbf{x}]} u_h\varphi_h^u + \int_{\Gamma_h[\mathbf{x}]} \nabla_{\Gamma_h[\mathbf{x}]}u_h\cdot\nabla_{\Gamma_h[\mathbf{x}]}\varphi_h^u = \int_{\Gamma_h[\mathbf{x}]} F(u_h,\nabla_{\Gamma_h[\mathbf{x}]}u_h)\varphi_h^u \tag{3.4}$$

for all $\varphi_h^u \in S_h[\mathbf{x}(t)]$ with $\partial_h^\bullet \varphi_h^u = 0$.

In the above approaches, the discretization of the evolution equations for v, H and u is done in the usual way of evolving surface finite elements. The velocity law (2.3) is enforced by a Ritz projection to the finite element space on $\Gamma_h[\mathbf{x}]$. Note that the finite element functions v_h and H_h are *not* the normal vector and the mean curvature of the discrete surface $\Gamma_h[\mathbf{x}(t)]$.

3.3 *Matrix–vector formulation*

We collect the nodal values in column vectors $\mathbf{v} = (v_j) \in \mathbb{R}^{3N}$, $\mathbf{n} = (\nu_j) \in \mathbb{R}^{3N}$, $\mathbf{H} = (H_j) \in \mathbb{R}^N$ and $\mathbf{w} = (u_j) \in \mathbb{R}^N$. We define the surface-dependent mass matrix $\mathbf{M}(\mathbf{x})$ and stiffness matrix $\mathbf{A}(\mathbf{x})$ on the surface determined by the nodal vector $\mathbf{x}$:

$$\begin{aligned} \mathbf{M}(\mathbf{x})|_{ij} &= \int_{\Gamma_h[\mathbf{x}]} \phi_i[\mathbf{x}]\phi_j[\mathbf{x}], \\ \mathbf{A}(\mathbf{x})|_{ij} &= \int_{\Gamma_h[\mathbf{x}]} \nabla_{\Gamma_h[\mathbf{x}]}\phi_i[\mathbf{x}]\cdot\nabla_{\Gamma_h[\mathbf{x}]}\phi_j[\mathbf{x}], \end{aligned} \qquad i,j = 1,\dots,N,$$

with the finite element nodal basis functions $\phi_j[\mathbf{x}] \in S_h[\mathbf{x}]$. We further let, for an arbitrary dimension d (with the identity matrices $I_d \in \mathbb{R}^{d\times d}$),

$$\mathbf{M}^{[d]}(\mathbf{x}) = I_d \otimes \mathbf{M}(\mathbf{x}), \qquad \mathbf{A}^{[d]}(\mathbf{x}) = I_d \otimes \mathbf{A}(\mathbf{x}), \qquad \mathbf{K}^{[d]}(\mathbf{x}) = I_d \otimes \big(\mathbf{M}(\mathbf{x}) + \mathbf{A}(\mathbf{x})\big).$$

When no confusion can arise, we write $\mathbf{M}(\mathbf{x})$ for $\mathbf{M}^{[d]}(\mathbf{x})$, $\mathbf{A}(\mathbf{x})$ for $\mathbf{A}^{[d]}(\mathbf{x})$, and $\mathbf{K}(\mathbf{x})$ for $\mathbf{K}^{[d]}(\mathbf{x})$.

We define nonlinear functions $\mathbf{f}(\mathbf{x},\mathbf{n},\mathbf{H},\mathbf{u}) \in \mathbb{R}^{5N}$ and $\mathbf{g}(\mathbf{x},\mathbf{n},\mathbf{H},\mathbf{u}) \in \mathbb{R}^{3N}$, where

$$\mathbf{f}(\mathbf{x},\mathbf{n},\mathbf{H},\mathbf{u}) = \begin{pmatrix} \mathbf{f}_v(\mathbf{x},\mathbf{n},\mathbf{H},\mathbf{u}) \\ \mathbf{f}_H(\mathbf{x},\mathbf{n},\mathbf{H},\mathbf{u}) \\ \mathbf{f}_u(\mathbf{x},\mathbf{H},\mathbf{u}) \end{pmatrix}$$

with $\mathbf{f}_v(\mathbf{x},\mathbf{n},\mathbf{H},\mathbf{u}) \in \mathbb{R}^{3N}$, $\mathbf{f}_H(\mathbf{x},\mathbf{n},\mathbf{H},\mathbf{u}) \in \mathbb{R}^N$ and $\mathbf{f}_u(\mathbf{x},\mathbf{n},\mathbf{H},\mathbf{u}) \in \mathbb{R}^N$. These functions are given as follows, with the notations $\alpha_h^2 = |\nabla_{\Gamma_h[\mathbf{x}]}\nu_h|^2$ and $V_h = -H_h + u_h$,

$$\begin{aligned} \mathbf{f}_v(\mathbf{x},\mathbf{n},\mathbf{H},\mathbf{u})|_{j+(\ell-1)N} &= \int_{\Gamma_h[\mathbf{x}]} \alpha_h^2\,(\nu_h)_\ell\ \phi_j[\mathbf{x}] - \int_{\Gamma_h[\mathbf{x}]} \Big(\nabla_{\Gamma_h[\mathbf{x}]}u_h\Big)_\ell \cdot \phi_j[\mathbf{x}], \\ \mathbf{f}_H(\mathbf{x},\mathbf{n},\mathbf{H},\mathbf{u})|_j &= -\int_{\Gamma_h[\mathbf{x}]} \alpha_h^2\, V_h\,\phi_j[\mathbf{x}] + \int_{\Gamma_h[\mathbf{x}]} \nabla_{\Gamma_h[\mathbf{x}]}u_h\cdot\nabla_{\Gamma_h[\mathbf{x}]}\phi_j[\mathbf{x}], \\ \mathbf{f}_u(\mathbf{x},\mathbf{H},\mathbf{u})|_j &= \int_{\Gamma_h[\mathbf{x}]} f(H_h,u_h,\nabla_{\Gamma_h[\mathbf{x}]}u_h)\,\phi_j[\mathbf{x}]; \\ \mathbf{g}(\mathbf{x},\mathbf{n},\mathbf{H},\mathbf{u})|_{j+(\ell-1)N} &= \int_{\Gamma_h[\mathbf{x}]} V_h(\nu_h)_\ell\,\phi_j[\mathbf{x}] + \int_{\Gamma_h[\mathbf{x}]} \nabla_{\Gamma_h[\mathbf{x}]}(V_h(\nu_h)_\ell)\cdot\nabla_{\Gamma_h[\mathbf{x}]}\phi_j[\mathbf{x}], \end{aligned}$$

for $j = 1, \dots, N$ and $\ell = 1, 2, 3$. We abbreviate

$$\mathbf{w} = \begin{pmatrix} \mathbf{n} \\ \mathbf{H} \\ \mathbf{u} \end{pmatrix} \in \mathbb{R}^{5N}.$$

Equations (3.1) and (3.2) with (3.3) can then be written in the matrix–vector formulation

$$\begin{aligned} \mathbf{K}^{[3]}(\mathbf{x})\mathbf{v} &= \mathbf{g}(\mathbf{x}, \mathbf{w}), \\ \mathbf{M}^{[5]}(\mathbf{x})\dot{\mathbf{w}} + \mathbf{A}^{[5]}(\mathbf{x})\mathbf{w} &= \mathbf{f}(\mathbf{x}, \mathbf{w}), \\ \dot{\mathbf{x}} &= \mathbf{v}. \end{aligned} \tag{3.5}$$

The system (3.5) for *forced* mean curvature flow is formally the same as the matrix–vector form of the coupled system for *non-forced* mean curvature flow derived in [27], cf. (3.4)–(3.5) therein, with $\mathbf{w} = (\mathbf{n}; \mathbf{H}; \mathbf{u}) \in \mathbb{R}^{5N}$ in the role of $\mathbf{u} = (\mathbf{n}; \mathbf{H}) \in \mathbb{R}^{4N}$ of [27]. The nonlinearity $\mathbf{f}(\mathbf{x}, \mathbf{w})$ is built up from integrals of the same type as $\mathbf{f}(\mathbf{x}, \mathbf{u})$ in [27], with the only exception of the second term in $\mathbf{f}_H$, whose entries contain the tangential gradient of the basis functions and which in total can be written as $\mathbf{A}(\mathbf{x})\mathbf{u}$. This term stems from the term $-\Delta_{\Gamma[X]} u$ in the evolution equation for H in Lemma 2.1. The function $\mathbf{g}$ is defined in the same way as $\mathbf{g}$ in [27], just with $V_h = -H_h + u_h$ in place of $-H_h$.

REMARK 3.1 For the alternative system of equations (3.1) and (3.4) with (3.3) we denote

$$\mathbf{z} = \begin{pmatrix} \mathbf{n} \\ \mathbf{H} \end{pmatrix} \in \mathbb{R}^{4N}, \qquad \mathbf{f}(\mathbf{x}, \mathbf{z}, \mathbf{u}) = \begin{pmatrix} \mathbf{f}_v(\mathbf{x}, \mathbf{n}, \mathbf{H}, \mathbf{u}) \\ \mathbf{f}_H(\mathbf{x}, \mathbf{n}, \mathbf{H}, \mathbf{u}) \end{pmatrix} \in \mathbb{R}^{4N}$$

and introduce

$$\mathbf{F}(\mathbf{x}, \mathbf{u})|_j = \int_{\Gamma_h[\mathbf{x}]} F(u_h, \nabla_{\Gamma_h[\mathbf{x}]} u_h)\, \phi_j[\mathbf{x}].$$

Equations (3.1) and (3.4) with (3.3) can then be written in the following matrix–vector form:

$$\begin{aligned} \mathbf{K}^{[3]}(\mathbf{x})\mathbf{v} &= \mathbf{g}(\mathbf{x}, \mathbf{z}, \mathbf{u}), \\ \mathbf{M}^{[4]}(\mathbf{x})\dot{\mathbf{z}} + \mathbf{A}^{[4]}(\mathbf{x})\mathbf{z} &= \mathbf{f}(\mathbf{x}, \mathbf{z}, \mathbf{u}), \\ \frac{\mathrm{d}}{\mathrm{d}t}\Big(\mathbf{M}(\mathbf{x})\mathbf{u}\Big) + \mathbf{A}(\mathbf{x})\mathbf{u} &= \mathbf{F}(\mathbf{x}, \mathbf{u}), \\ \dot{\mathbf{x}} &= \mathbf{v}. \end{aligned} \tag{3.6}$$

3.4 *Lifts*

As in [28] and [27, Section 3.4], we compare functions on the *exact surface* $\Gamma[X(\cdot, t)]$ with functions on the *discrete surface* $\Gamma_h[\mathbf{x}(t)]$, via functions on the *interpolated surface* $\Gamma_h[\mathbf{x}^*(t)]$, where $\mathbf{x}^*(t)$ denotes the nodal vector collecting the grid points $x_j^*(t) = X(p_j, t)$ on the exact surface, where p_j are the nodes of the discrete initial triangulation Γ_h^0.

Any finite element function w_h on the discrete surface, with nodal values w_j, is associated with a finite element function $\widehat{w}_h$ on the interpolated surface Γ_h^* with the exact same nodal values. This can be further lifted to a function on the exact surface by using the *lift operator* l, mapping a

function on the interpolated surface Γ_h^* to a function on the exact surface Γ, provided that they are sufficiently close, see [12, 13].

Then the composed lift L maps finite element functions on the discrete surface $\Gamma_h[\mathbf{x}(t)]$ to functions on the exact surface $\Gamma[X(\cdot,t)]$ via the interpolated surface $\Gamma_h[\mathbf{x}^*(t)]$. This is denoted by

$$w_h^L = (\widehat{w}_h)^l.$$

4. Convergence of the semi-discretization

We are now in the position to formulate the first main result of this paper, which yields optimal-order error bounds for the finite element semi-discretization (using finite elements of polynomial degree $k \geqslant 2$) (3.1), and (3.4) or (3.2), with (3.3) of the system for forced mean curvature equations (2.7), and one of the weak formulations (2.8) or (2.9) for the surface PDE, with the ODE (2.1) for the positions. We introduce the notation

$$x_h^L(x,t) = X_h^L(p,t) \in \Gamma_h[\mathbf{x}(t)] \qquad \text{for} \quad x = X(p,t) \in \Gamma[X(\cdot,t)].$$

Theorem 4.1 *For the coupled forced mean curvature flow problem (2.7) and (2.9) with a smooth function F, taken together with the velocity equation (2.1), we consider the space discretization (3.1)–(3.3) (or equivalently (3.5) in matrix–vector form) with evolving surface finite elements of polynomial degree $k \geqslant 2$. Suppose that the problem admits an exact solution (X, v, ν, H, u) that is sufficiently regular on the time interval $t \in [0,T]$, and that the flow map $X(\cdot,t)$ is non-degenerate so that $\Gamma(t) = \Gamma[X(\cdot,t)]$ is a regular surface on the time interval $t \in [0,T]$.*

Then, there exists a constant $h_0 > 0$ such that for all mesh sizes $h \leqslant h_0$ the following error bounds for the lifts of the discrete position, velocity, normal vector and mean curvature hold over the exact surface $\Gamma(t)$ for $0 \leqslant t \leqslant T$:

$$\begin{aligned}
\|x_h^L(\cdot,t) - \mathrm{id}_{\Gamma(t)}\|_{H^1(\Gamma(t))^3} &\leqslant Ch^k,\\
\|v_h^L(\cdot,t) - v(\cdot,t)\|_{H^1(\Gamma(t))^3} &\leqslant Ch^k,\\
\|\nu_h^L(\cdot,t) - \nu(\cdot,t)\|_{H^1(\Gamma(t))^3} &\leqslant Ch^k,\\
\|H_h^L(\cdot,t) - H(\cdot,t)\|_{H^1(\Gamma(t))} &\leqslant Ch^k,\\
\|u_h^L(\cdot,t) - u(\cdot,t)\|_{H^1(\Gamma(t))} &\leqslant Ch^k,
\end{aligned}$$

and also

$$\|X_h^l(\cdot,t) - X(\cdot,t)\|_{H^1(\Gamma_0)^3} \leqslant Ch^k,$$

where the constant C is independent of h and t, but depends on bounds of higher derivatives of the solution (X, v, ν, H, u) of the forced mean curvature flow and on the length T of the time interval.

Sufficient regularity assumptions are the following: with bounds that are uniform in $t \in [0,T]$, we assume $X(\cdot,t) \in H^{k+1}(\Gamma^0)^3$ and for $w = (\nu, H, u)$ we assume $w(\cdot,t), \partial^\bullet w(\cdot,t) \in W^{k+1,\infty}(\Gamma(t))^5$.

Under these strong regularity conditions on the solution, we only require local Lipschitz continuity of the function F in (1.2). This condition is, of course, not sufficient to ensure the existence of even just a weak solution. The point here is that we restrict our attention to cases where

a sufficiently regular solution exists, which we can then approximate with optimal order under weak conditions on F. The regularity theory of Problem (1.1)–(1.2) is, however, outside the scope of this paper.

The remarks made after the convergence result in [27] apply also here. In particular, it is explained that the admissibility of the triangulation over the whole time interval $[0, T]$ is preserved for sufficiently fine grids, provided the exact surface is sufficiently regular.

Proof. The proof reduces in essence to the proof of Theorem 4.1 in [27], since the matrix–vector formulation (3.5) is of precisely the same form as the matrix–vector formulation of [27], formulas (3.4)–(3.5) therein, with the same mass and stiffness matrices and with nonlinear functions given as integrals over products of smooth pointwise nonlinearities and finite element basis functions (and with $\mathbf{w}$ in the role of $\mathbf{u}$ of [27]). The proof of the stability bounds of [27, Proposition 7.1] uses energy estimates (testing with the time derivative of the error) on the equations of the matrix–vector formulation to bound errors in terms of defects in (3.5) in the appropriate norms. These stability bounds apply immediately to (3.5) with the same proof, except for one subtle point: Because of the term $-\Delta_{\Gamma[X]}u$ in the evolution equation for H in Lemma 2.1, which translates into the second term $\mathbf{A}(\mathbf{x})\mathbf{u}$ in $\mathbf{f}_H(\mathbf{x}, \mathbf{w})$ in the matrix–vector formulation, the bound for the nonlinearity in part (v) of the proof of Proposition 7.1 in [27] needs to be changed. This is a very local modification to the proof. No other part of the stability proof is affected.

To explain and resolve this local difficulty, we must assume that the reader has acquired some familiarity with Section 7 of [27]. We use the same notation $\mathbf{e}_\mathbf{w} = \mathbf{w} - \mathbf{w}^*$ etc. for the error vectors and note that $\mathbf{e}_\mathbf{w} = (\mathbf{e}_\mathbf{n}; \mathbf{e}_\mathbf{H}; \mathbf{e}_\mathbf{u})$ now is in the role of $\mathbf{e}_\mathbf{u} = (\mathbf{e}_\mathbf{n}; \mathbf{e}_\mathbf{H})$ of [27]. Because of the extra term $\mathbf{A}(\mathbf{x})\mathbf{u}$ in $\mathbf{f}_H(\mathbf{x}, \mathbf{w})$, the same argument as in part (v) of the proof of Proposition 7.1 in [27] yields only a modified bound

$$\dot{\mathbf{e}}_\mathbf{w}^T \big(\mathbf{f}(\mathbf{x}, \mathbf{w}) - \mathbf{f}(\mathbf{x}^*, \mathbf{w}^*)\big) \leqslant c \|\dot{\mathbf{e}}_\mathbf{w}\|_{\mathbf{K}(\mathbf{x}^*)} \big(\|\mathbf{e}_\mathbf{w}\|_{\mathbf{K}(\mathbf{x}^*)} + \|\mathbf{e}_\mathbf{x}\|_{\mathbf{A}(\mathbf{x}^*)}\big),$$

whereas in [27] only the weaker norm $\|\dot{\mathbf{e}}_\mathbf{w}\|_{\mathbf{M}(\mathbf{x}^*)}$ appears on the right-hand side. This modified estimate is not sufficient for the further course of the proof.

It can be circumvented as follows. We write the error vector as $\mathbf{e}_\mathbf{w} = (\mathbf{e}_\mathbf{n}; \mathbf{e}_\mathbf{H}; \mathbf{e}_\mathbf{u})$ and take the inner product of $\dot{\mathbf{e}}_\mathbf{H}$ with $\big(\mathbf{f}_H(\mathbf{x}, \mathbf{w}) - \mathbf{f}_H(\mathbf{x}^*, \mathbf{w}^*)\big)$. We note that

$$\mathbf{f}_H(\mathbf{x}, \mathbf{w}) = \widetilde{\mathbf{f}}_H(\mathbf{x}, \mathbf{w}) + \mathbf{A}(\mathbf{x})\mathbf{u},$$

where $\widetilde{\mathbf{f}}_H$ is a nonlinearity of the same type as those studied in [27], and so we have the following bound as in part (v) of the proof of Proposition 7.1 in [27],

$$\dot{\mathbf{e}}_\mathbf{H}^T \big(\widetilde{\mathbf{f}}_H(\mathbf{x}, \mathbf{w}) - \widetilde{\mathbf{f}}_H(\mathbf{x}^*, \mathbf{w}^*)\big) \leqslant c \|\dot{\mathbf{e}}_\mathbf{w}\|_{\mathbf{M}(\mathbf{x}^*)} \big(\|\mathbf{e}_\mathbf{w}\|_{\mathbf{K}(\mathbf{x}^*)} + \|\mathbf{e}_\mathbf{x}\|_{\mathbf{A}(\mathbf{x}^*)}\big).$$

For the solution $\mathbf{x}(t)$ of (3.5) we have

$$\mathbf{A}(\mathbf{x})\mathbf{u} = -\mathbf{M}(\mathbf{x})\dot{\mathbf{u}} + \mathbf{f}_u(\mathbf{x}, \mathbf{w})$$

and for the nodal vector $\mathbf{u}^*(t)$ of the Ritz projection of the exact solution $u(\cdot, t)$ and the nodal vector $\mathbf{x}^*(t)$ of the exact positions we have, with a defect $\mathbf{d}_\mathbf{u}(t)$,

$$\mathbf{A}(\mathbf{x}^*)\mathbf{u}^* = -\mathbf{M}(\mathbf{x}^*)\dot{\mathbf{u}}^* + \mathbf{f}_u(\mathbf{x}^*, \mathbf{w}^*) + \mathbf{M}(\mathbf{x}^*)\mathbf{d}_\mathbf{u}.$$

So we can write

$$\dot{\mathbf{e}}_{\mathbf{H}}^T\big(\mathbf{A}(\mathbf{x})\mathbf{u}-\mathbf{A}(\mathbf{x}^*)\mathbf{u}^*\big) = -\dot{\mathbf{e}}_{\mathbf{H}}^T\mathbf{M}(\mathbf{x})\dot{\mathbf{e}}_{\mathbf{u}} - \dot{\mathbf{e}}_{\mathbf{H}}^T\big(\mathbf{M}(\mathbf{x})-\mathbf{M}(\mathbf{x}^*)\big)\mathbf{u}^* + \dot{\mathbf{e}}_{\mathbf{H}}^T\big(\mathbf{f}_u(\mathbf{x},\mathbf{w})-\mathbf{f}_u(\mathbf{x}^*,\mathbf{w}^*)\big) - \dot{\mathbf{e}}_{\mathbf{H}}^T\mathbf{M}(\mathbf{x}^*)\mathbf{d}_{\mathbf{u}}.$$

By the same estimates as used repeatedly in the proof of Proposition 7.1 in [27], this yields

$$\dot{\mathbf{e}}_{\mathbf{H}}^T\big(\mathbf{A}(\mathbf{x})\mathbf{u}-\mathbf{A}(\mathbf{x}^*)\mathbf{u}^*\big) \leqslant \|\dot{\mathbf{e}}_{\mathbf{H}}\|_{\mathbf{M}(\mathbf{x})}\|\dot{\mathbf{e}}_{\mathbf{u}}\|_{\mathbf{M}(\mathbf{x})} + c\|\dot{\mathbf{e}}_{\mathbf{H}}\|_{\mathbf{M}(\mathbf{x}^*)}\|\mathbf{e}_{\mathbf{x}}\|_{\mathbf{A}(\mathbf{x}^*)} + c\|\dot{\mathbf{e}}_{\mathbf{H}}\|_{\mathbf{M}(\mathbf{x}^*)}\big(\|\mathbf{e}_{\mathbf{w}}\|_{\mathbf{K}(\mathbf{x}^*)} + \|\mathbf{e}_{\mathbf{x}}\|_{\mathbf{A}(\mathbf{x}^*)}\big) + \|\dot{\mathbf{e}}_{\mathbf{H}}\|_{\mathbf{M}(\mathbf{x}^*)}\|\mathbf{d}_{\mathbf{u}}\|_{\mathbf{M}(\mathbf{x}^*)}.$$

We now fix a small $\rho > 0$ and use the scaled norm, for $\dot{\mathbf{e}}_{\mathbf{w}} = (\dot{\mathbf{e}}_{\mathbf{n}}; \dot{\mathbf{e}}_{\mathbf{H}}; \dot{\mathbf{e}}_{\mathbf{u}})$,

$$\|\dot{\mathbf{e}}_{\mathbf{w}}\|_{\mathbf{M}(\mathbf{x})}^2 = \|\dot{\mathbf{e}}_{\mathbf{n}}\|_{\mathbf{M}(\mathbf{x})}^2 + \|\dot{\mathbf{e}}_{\mathbf{H}}\|_{\mathbf{M}(\mathbf{x})}^2 + \omega^2\|\dot{\mathbf{e}}_{\mathbf{u}}\|_{\mathbf{M}(\mathbf{x})}^2$$

with a large weight ω. If $\omega \geqslant 1/(2\rho)$, then we have

$$\|\dot{\mathbf{e}}_{\mathbf{H}}\|_{\mathbf{M}(\mathbf{x})}\|\dot{\mathbf{e}}_{\mathbf{u}}\|_{\mathbf{M}(\mathbf{x})} \leqslant \rho\|\dot{\mathbf{e}}_{\mathbf{w}}\|_{\mathbf{M}(\mathbf{x})}^2.$$

Altogether, this yields the bound

$$\dot{\mathbf{e}}_{\mathbf{w}}^T\big(\mathbf{f}(\mathbf{x},\mathbf{w})-\mathbf{f}(\mathbf{x}^*,\mathbf{w}^*)\big) \leqslant \rho\|\dot{\mathbf{e}}_{\mathbf{w}}\|_{\mathbf{M}(\mathbf{x})}^2 + c\|\dot{\mathbf{e}}_{\mathbf{w}}\|_{\mathbf{M}(\mathbf{x}^*)}\big(\|\mathbf{e}_{\mathbf{w}}\|_{\mathbf{K}(\mathbf{x}^*)} + \|\mathbf{e}_{\mathbf{x}}\|_{\mathbf{A}(\mathbf{x}^*)} + \|\mathbf{d}_{\mathbf{u}}\|_{\mathbf{M}(\mathbf{x}^*)}\big).$$

With this bound, the further parts of the stability proof remain unchanged.

Since the additional terms in (2.7) and (2.9) to those in the evolution equations of pure mean curvature flow in [27] do not present additional difficulties in the consistency error analysis, the same bounds for the consistency errors in (X, v, H, ν, u) are obtained as for (X, v, H, ν) in [27, Proposition 8.1]. Furthermore, the combination of the stability bounds and the consistency error bounds to yield optimal-order H^1 error bounds is verbatim the same as in [27, Section 9]. □

REMARK 4.2 For the semi-discretization (3.6) a convergence proof can be obtained by combining the convergence proofs of our previous works [28] and [27]. The stability of the scheme is obtained by combining the results of [28, Proposition 6.1] (in particular part (A)) for the surface PDE, and of [27, Proposition 7.1] for the velocity law and for the geometric quantities, and further using the same modification for the extra term $\mathbf{A}(\mathbf{x})\mathbf{u}$ as in the proof above. As this extension does not require any new ideas beyond [28] and [27], we do not present the lengthy but straightforward details. Since there are no additional difficulties in bounding the consistency errors, together with the stability bounds we then obtain the same error bounds as in Theorem 4.1. This is in agreement with the results of numerical experiments presented in Section 7.

5. Linearly implicit full discretization

For the time discretization of the system of ordinary differential equations of Section 3.3 we use a q-step linearly implicit backward difference formula (BDF) with $q \leqslant 5$. For a step size $\tau > 0$, and with $t_n = n\tau \leqslant T$, let us introduce, for $n \geqslant q$,

$$\text{the discrete time derivative} \qquad \dot{\mathbf{u}}^n = \frac{1}{\tau}\sum_{j=0}^{q}\delta_j\,\mathbf{u}^{n-j}, \qquad \text{and} \tag{5.1}$$

$$\text{the extrapolated value} \qquad \widetilde{\mathbf{u}}^n = \sum_{j=0}^{q-1}\gamma_j\,\mathbf{u}^{n-1-j}, \tag{5.2}$$

where the coefficients are given by $\delta(\zeta) = \sum_{j=0}^{q} \delta_j \zeta^j = \sum_{\ell=1}^{q} \frac{1}{\ell}(1-\zeta)^\ell$ and $\gamma(\zeta) = \sum_{j=0}^{q-1} \gamma_j \zeta^j = (1-(1-\zeta)^q)/\zeta$, respectively.

We determine the approximations $\mathbf{x}^n$ to $\mathbf{x}(t_n)$, $\mathbf{v}^n$ to $\mathbf{v}(t_n)$, and $\mathbf{w}^n$ to $\mathbf{w}(t_n)$ or $\mathbf{z}^n$ to $\mathbf{z}(t_n)$ and $\mathbf{u}^n$ to $\mathbf{u}(t_n)$ (only if not already collected into $\mathbf{w}^n$) by the linearly implicit BDF discretization of both systems (3.5) and (3.6).

For (3.5) we obtain

$$\begin{aligned} \mathbf{K}(\widetilde{\mathbf{x}}^n)\mathbf{v}^n &= \mathbf{g}(\widetilde{\mathbf{x}}^n, \widetilde{\mathbf{w}}^n), \\ \mathbf{M}(\widetilde{\mathbf{x}}^n)\dot{\mathbf{w}}^n + \mathbf{A}(\widetilde{\mathbf{x}}^n)\mathbf{w}^n &= \mathbf{f}(\widetilde{\mathbf{x}}^n, \widetilde{\mathbf{w}}^n), \\ \dot{\mathbf{x}}^n &= \mathbf{v}^n. \end{aligned} \tag{5.3}$$

For (3.6) we obtain

$$\begin{aligned} \mathbf{K}(\widetilde{\mathbf{x}}^n)\mathbf{v}^n &= \mathbf{g}(\widetilde{\mathbf{x}}^n, \widetilde{\mathbf{z}}^n, \widetilde{\mathbf{u}}^n), \\ \mathbf{M}(\widetilde{\mathbf{x}}^n)\dot{\mathbf{z}}^n + \mathbf{A}(\widetilde{\mathbf{x}}^n)\mathbf{z}^n &= \mathbf{f}(\widetilde{\mathbf{x}}^n, \widetilde{\mathbf{z}}^n, \widetilde{\mathbf{u}}^n), \\ \frac{1}{\tau}\sum_{j=0}^{q} \delta_j \mathbf{M}(\widetilde{\mathbf{x}}^{n-j})\mathbf{u}^{n-j} + \mathbf{A}(\widetilde{\mathbf{x}}^n)\mathbf{u}^n &= \mathbf{F}(\widetilde{\mathbf{x}}^n, \widetilde{\mathbf{u}}^n), \\ \dot{\mathbf{x}}^n &= \mathbf{v}^n. \end{aligned} \tag{5.4}$$

The starting values $\mathbf{x}^i$ and $\mathbf{w}^i$, or, in case of (5.4), $\mathbf{z}^i$ and $\mathbf{u}^i$, for $i = 0, \ldots, q-1$, are assumed to be given. They can be precomputed using either a lower order method with smaller step sizes or an implicit Runge–Kutta method.

The classical BDF method is known to be $A(\theta)$-stable for some $\theta > 0$ for $q \leqslant 6$ and to have order q; see [24, Chapter V]. This order is retained by the linearly implicit variant using the above coefficients γ_j; cf. [1].

From the vectors $\mathbf{x}^n = (x_j^n)$, $\mathbf{v}^n = (v_j^n)$, and $\mathbf{w}^n = (w_j^n)$ with $w_j^n = (v_j^n, H_j^n, u_j^n) \in \mathbb{R}^3 \times \mathbb{R} \times \mathbb{R}$ for the first method and $\mathbf{z}^n = (z_j^n)$ with $z_j^n = (v_j^n, H_j^n) \in \mathbb{R}^3 \times \mathbb{R}$ and $\mathbf{u}^n = (u_j^n)$ for the second method, we obtain approximations to their respective variables as finite element functions whose nodal values are collected in these vectors.

6. Convergence of the full discretization

We are now in the position to formulate the second main result of this paper, which yields optimal-order error bounds for the combined ESFEM–BDF full discretizations (5.3) of the forced mean curvature flow problem (2.7) coupled to the weak form (2.9) of the surface PDE, with (2.1), for finite elements of polynomial degree $k \geqslant 2$ and BDF methods of order $2 \leqslant q \leqslant 5$.

Theorem 6.1 *Consider the ESFEM–BDF full discretizations (5.3) of the coupled forced mean curvature flow problem (2.7) and (2.9), with (2.1), using evolving surface finite elements of polynomial degree $k \geqslant 2$ and linearly implicit BDF time discretization of order q with $2 \leqslant q \leqslant 5$. Suppose that the forced mean curvature flow problem admits an exact solution (X, v, ν, H, u) that is sufficiently smooth on the time interval $t \in [0, T]$, and that the flow map $X(\cdot, t) : \Gamma^0 \to \Gamma(t) \subset \mathbb{R}^3$ is non-degenerate so that $\Gamma(t)$ is a regular surface on the time interval $t \in [0, T]$. Assume that the starting values are sufficiently accurate in the H^1 norm at time $t_i = i\tau$ for $i = 0, \ldots, q-1$.*

Then there exist $h_0 > 0$ and $\tau_0 > 0$ such that for all mesh sizes $h \leqslant h_0$ and time step sizes $\tau \leqslant \tau_0$ satisfying the step size restriction

$$\tau \leqslant C_0 h \tag{6.1}$$

(where $C_0 > 0$ can be chosen arbitrarily), the following error bounds for the lifts of the discrete position, velocity, normal vector and mean curvature hold over the exact surface $\Gamma(t_n) = \Gamma[X(\cdot,t_n)]$ at time $t_n = n\tau \leqslant T$:

$$\begin{aligned}
\|(x_h^n)^L - \mathrm{id}_{\Gamma(t_n)}\|_{H^1(\Gamma(t_n))^3} &\leqslant C(h^k + \tau^q),\\
\|(v_h^n)^L - v(\cdot,t_n)\|_{H^1(\Gamma(t_n))^3} &\leqslant C(h^k + \tau^q),\\
\|(\nu_h^n)^L - \nu(\cdot,t_n)\|_{H^1(\Gamma(t_n))^3} &\leqslant C(h^k + \tau^q),\\
\|(H_h^n)^L - H(\cdot,t_n)\|_{H^1(\Gamma(t_n))} &\leqslant C(h^k + \tau^q),\\
\|(u_h^n)^L - u(\cdot,t_n)\|_{H^1(\Gamma(t_n))} &\leqslant C(h^k + \tau^q),
\end{aligned}$$

and also

$$\|(X_h^n)^l - X(\cdot,t_n)\|_{H^1(\Gamma_0)^3} \leqslant C(h^k + \tau^q),$$

where the constant C is independent of h, τ and n with $n\tau \leqslant T$, but depends on bounds of higher derivatives of the solution (X, v, ν, H, u) of the forced mean curvature flow problem, on the length T of the time interval, and on C_0.

Sufficient regularity assumptions are the following: uniformly in $t \in [0,T]$ and for $j = 1,\dots,q+1$,

$$\begin{aligned}
&X(\cdot,t) \in H^{k+1}(\Gamma^0)^3,\ \partial_t^j X(\cdot,t) \in H^1(\Gamma^0)^3,\\
&v(\cdot,t) \in H^{k+1}\big(\Gamma(t)\big)^3,\ \partial^{\bullet\, j} v(\cdot,t) \in H^2\big(\Gamma(t)\big)^3,\\
\text{for } w = (\nu,H,u),\quad &w(\cdot,t), \partial^\bullet w(\cdot,t) \in W^{k+1,\infty}\big(\Gamma(t)\big)^5,\ \partial^{\bullet\, j} w(\cdot,t) \in H^2\big(\Gamma(t)\big)^5.
\end{aligned}$$

For the starting values, sufficient approximation conditions are as follows: for $i = 0,\dots,q-1$,

$$\begin{aligned}
&\|(x_h^i)^L - \mathrm{id}_{\Gamma(t_i)}\|_{H^1(\Gamma(t_i))^3} \leqslant C(h^k + \tau^q),\\
\text{for } w = (\nu,H,u),\qquad &\|(w_h^i)^L - w(\cdot,t_i)\|_{H^1(\Gamma(t_i))^5} \leqslant C(h^k + \tau^q),
\end{aligned}$$

and in addition, for $i = 1,\dots,q-1$,

$$\tau^{1/2}\Big\|\frac{1}{\tau}\big(X_h^i - X_h^{i-1}\big)^l - \frac{1}{\tau}\big(X(\cdot,t_i) - X(\cdot,t_{i-1})\big)\Big\|_{H^1(\Gamma_0)^3} \leqslant C(h^k + \tau^q).$$

Since (5.3) is the same as the matrix–vector form of mean curvature flow in [27, equation (5.1)] (recalling that here $\mathbf{w} = (\mathbf{n};\mathbf{H};\mathbf{u})$ takes the role of $\mathbf{u} = (\mathbf{n};\mathbf{H})$ of [27]) and the only problematic additional term in (5.3) is the term $\mathbf{A}(\widetilde{\mathbf{x}}^n)\widetilde{\mathbf{u}}^n$ that appears in $\mathbf{f}(\widetilde{\mathbf{x}}^n,\widetilde{\mathbf{w}}^n)$, the proof of Theorem 6.1 directly follows from the error analysis presented in [27] together with the modification concerning $\mathbf{A}(\mathbf{x})\mathbf{u}$ given in the proof of Theorem 4.1.

REMARK 6.2 For the second algorithm (5.4), we expect that a fully discrete error estimate can be obtained by combining the stability results for the coupled mean curvature flow, [27, Proposition 10.1], with the extension of the stability analysis for the surface PDE [29, Proposition 6.1] (via energy estimates obtained by testing with $\dot{\mathbf{e}}^n$). We note here that this extension, in particular the analogous steps to part (iv) in [27, Proposition 10.1], is lengthy and possibly nontrivial. Numerical experiments presented in Section 7 illustrate that optimal-order error estimates are also observed for the scheme (5.4).

7. Numerical experiments

We present numerical experiments for the forced mean curvature flow, using both (5.3) and (5.4). For our numerical experiments we consider the problem coupling forced mean curvature flow (with a new parameter $\varepsilon > 0$) of the surface $\Gamma(X(\cdot,t))$, together with evolution equations for its normal vector ν and mean curvature H, where the forcing is given through the solution u of a reaction–diffusion problem on the surface:

$$\begin{aligned}
\partial^\bullet u &= -u(\nabla_{\Gamma[X]}\cdot v) + \Delta_{\Gamma[X]}u + f(u,\nabla_{\Gamma[X]}u) + \varrho_1,\\
v &= -\varepsilon H\nu + g(u)\nu + \varrho_2,\\
\partial^\bullet \nu &= \varepsilon\Delta_{\Gamma[X]}\nu + \varepsilon|A|^2\nu - \nabla_{\Gamma[X]}(g(u)) + \varrho_3,\\
\partial^\bullet H &= \varepsilon\Delta_{\Gamma[X]}H + \varepsilon|A|^2H - \Delta_{\Gamma[X]}(g(u)) - |A|^2 g(u) + \varrho_4,\\
\partial_t X &= v,
\end{aligned} \tag{7.1}$$

where the inhomogeneities ϱ_i are scalar or vector valued functions on $\mathbb{R}^3\times[0,T]$, to be specified later on.

We used this problem to perform:

- A convergence order experiment for the algorithm (5.3), in order to illustrate our theoretical results of Theorem 4.1 and 6.1.
- A convergence order experiment for algorithm (5.4), illustrating Remark 4.2 and 6.2.
- An experiment, using algorithm (5.3), for a tumour growth model from [2, Section 5], where one component of a reaction–diffusion surface PDE system forces the mean curvature flow motion of the surface. This experiment allows a direct comparison on the same problem with other methods published in the literature.

All our numerical experiments use quadratic evolving surface finite elements, and linearly implicit BDF methods. The numerical computations were carried out in Matlab. The initial meshes for all surfaces were generated using DistMesh [33], without taking advantage of the symmetries of the surfaces.

7.1 *Convergence experiments*

In order to illustrate the convergence results of Theorem 4.1 and 6.1, we have computed the errors between the numerical and exact solutions of the system (7.1), where the forcing is set to be $g(u) = u$, and $\epsilon = 1$. The reaction term in the PDE is $F(u,\nabla_{\Gamma[X]}u) = u^2$. The inhomogeneities ϱ_i are chosen such that the exact solution is $X(q,t) = R(t)q$, with q on the initial surface Γ_0, the sphere with radius R_0, and $u(x,t) = e^{-t}x_1x_2$, for all $x\in\Gamma[X]$ and $0\leqslant t\leqslant T$. The function R satisfies the logistic differential equation:

$$\begin{aligned}
\frac{\mathrm{d}R(t)}{\mathrm{d}t} &= \left(1-\frac{R(t)}{R_1}\right)R(t), \qquad t\in[0,T],\\
R(0) &= R_0,
\end{aligned}$$

with $R_1 \geqslant R_0$, i.e., the exact evolving surface $\Gamma[X(\cdot,t)]$ is a sphere with radius $R(t) = R_0R_1\big(R_0(1-e^{-t}) + R_1e^{-t}\big)^{-1}$.

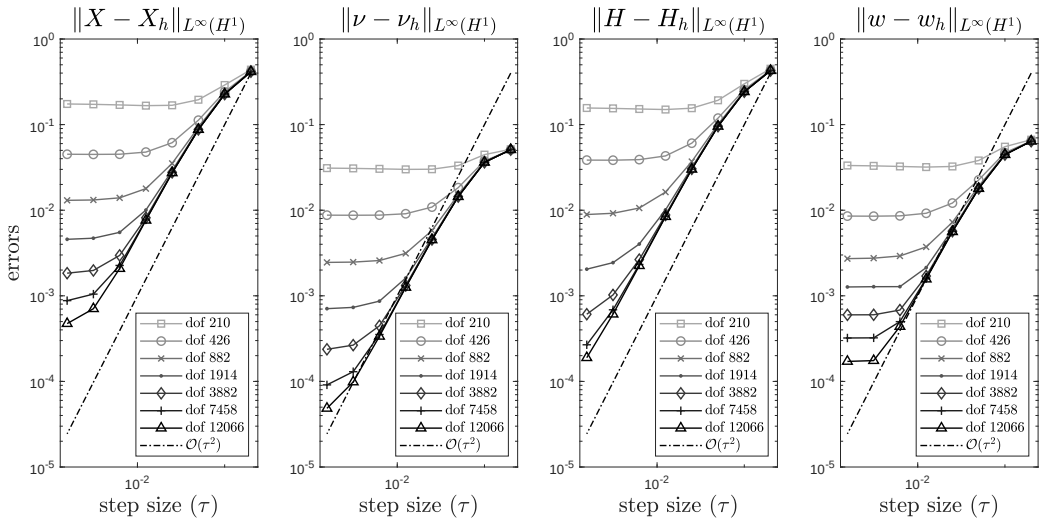

FIG. 1. Temporal convergence of the algorithm (5.3) for forced MCF with $g(u) = u$, using BDF2/quadratic ESFEM

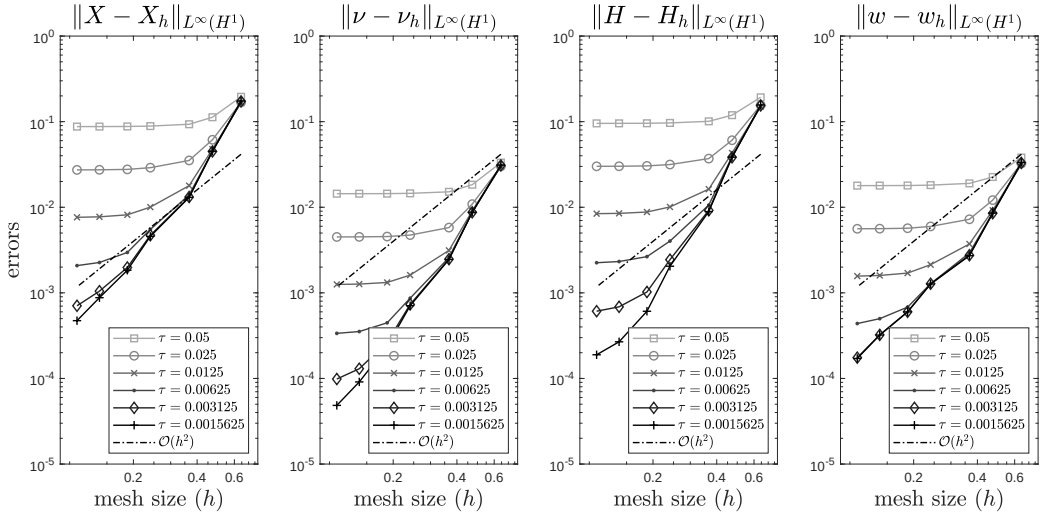

FIG. 2. Spatial convergence of the algorithm (5.3) for forced MCF with $g(u) = u$, using BDF2/quadratic ESFEM

Using the algorithm in (5.3) with 2-step BDF method and quadratic evolving surface FEM, we computed approximations to forced mean curvature flow, using $R_0 = 1$ and $R_1 = 2$, until time $T = 1$. For our computations we used a sequence of time step sizes $\tau_k = \tau_{k-1}/2$ with $\tau_0 = 0.2$, and a sequence of initial meshes of mesh widths $h_k \approx 2^{-1/2} h_{k-1}$ with $h_0 \approx 0.5$. The numerical experiments suggest that the step size restriction (6.1) is not required in practice.

In Figure 1 and 2 we report the errors between the exact and both numerical solutions for all four variables, i.e., the surface error, the errors in the dynamic variables ν and H, and the error

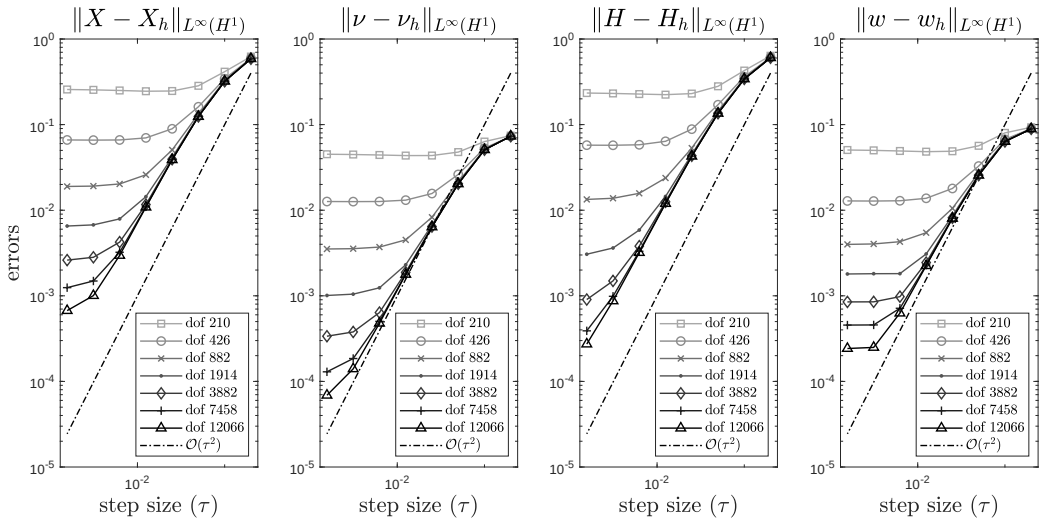

FIG. 3. Temporal convergence of the algorithm (5.4) for forced MCF with $g(u) = u$, using BDF2/quadratic ESFEM

in the PDE variable u. The logarithmic plots show the $L^\infty(H^1)$ norm errors against the time step size τ in Figure 1, and against the mesh width h in Figure 2. The lines marked with different symbols correspond to different mesh refinements and to different time step sizes in Figure 1 and 2, respectively.

In Figure 1 we can observe two regions: a region where the temporal discretization error dominates, matching to the $O(\tau^2)$ order of convergence of our theoretical results, and a region, with small time step sizes, where the spatial discretization error dominates (the error curves flatten out). For Figure 2, the same description applies, but with reversed roles.

Both the temporal and spatial convergence, as shown by Figures 1 and 2, respectively, are in agreement with the theoretical convergence results of Theorem 4.1 and 6.1 (note the reference lines).

We have performed the same convergence experiments using algorithm (5.4), which, in view of Remarks 4.2 and 6.2, and the stability and convergence results of previous works [27–29, 32], should also have the same convergence properties as the algorithm (5.4). As Figures 3 and 4 (created analogously as Figure 1 and 2) illustrate, this expectation appears to be fulfilled.

We have obtained similar convergence plots for the non-linear forcing term $g(u) = \frac{1}{2}u^2$ for both algorithms.

7.2 *Tumour growth*

We performed numerical experiments, using (5.3), on a well-known model for forced mean curvature flow from [2, Section 5]: The problem (7.1), with vector valued unknown $u = (u_1, u_2)$ and with a small parameter $\varepsilon = 0.01$, models solid tumour growth, for further details we refer to [7–9] and [2]. Our results can be compared to those in these references, in particularly with those in [2].

The surface PDE system for $u = (u_1, u_2)$ describes the activator–depleted kinetics, and has diffusivity constants 1 and $d = 10$ for u_1 and u_2, respectively. The reaction term is given by, with

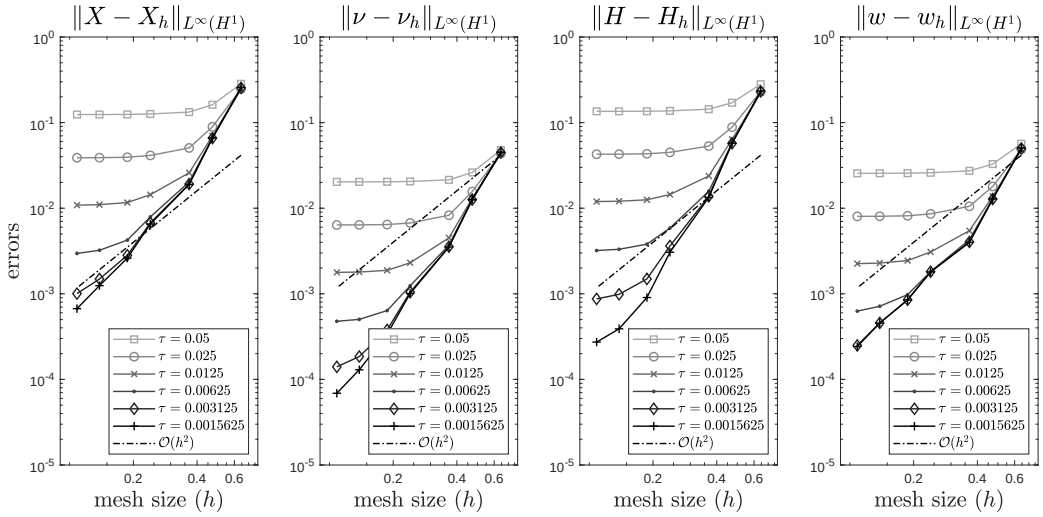

FIG. 4. Spatial convergence of the algorithm (5.4) for forced MCF with $g(u) = u$, using BDF2/quadratic ESFEM

$\gamma > 0$,

$$F(u) = F(u_1, u_2) = \begin{pmatrix} \gamma(a - u_1 + u_1^2 u_2) \\ \gamma(b - u_1^2 u_2) \end{pmatrix},$$

while in the velocity law the non-linearity is given by

$$g(u) = g(u_1, u_2) = \delta\, u_1.$$

The parameters are chosen exactly as in [2, Table 5]: $d = 10$, $a = 0.1$, $b = 0.9$, $\delta = 0.1$, and $\epsilon = 0.01$. The parameter γ will be varied for different experiments.

The initial data for all of the presented experiments are obtained (exactly as in [2, Section 4.1.1 and Figure 8]) by integrating the reaction–diffusion system on the fixed unit sphere over the time interval $[0, 5]$, with small random perturbations of the steady state $u_1 = a + b$ and $u_2 = b/(a + b)^2$ as initial data. Further initial values (for $i = 1, \ldots, q-1$) for high-order BDF methods are computed using a cascade of steps performed by the corresponding lower order methods.

To mitigate the stiffness of the non-linear term, the linear part of $F(u)$ is handled fully implicitly, while the non-linear parts of F, and the velocity law as well, are treated linearly implicitly using the extrapolation (5.2).

In Figure 5 and 6 we report on the evolution of the surface (and the approximated mean curvature and normal vector) and the component u_1 for parameters $\gamma = 30$ and $\gamma = 300$, respectively, at different times over the time interval $[5, 8]$. In these plots the linear interpolation of the computed quadratic surface is plotted (since Matlab can only visualise polygonal objects). Figure 5 and 6 we present the surface evolution and the component u_1 of the surface PDE system (left-hand side columns) and the computed mean curvature H_h and normal vector ν_h (right-hand side columns) at times $t = 5, 6, 7, 8$ (the rows from top to bottom), on a mesh with 3882 nodes and time step size $\tau = 0.0015625$. In particular the top rows show the initial data where the surface evolution is started. The obtained results for the surface evolution and the reaction–diffusion PDE system (left

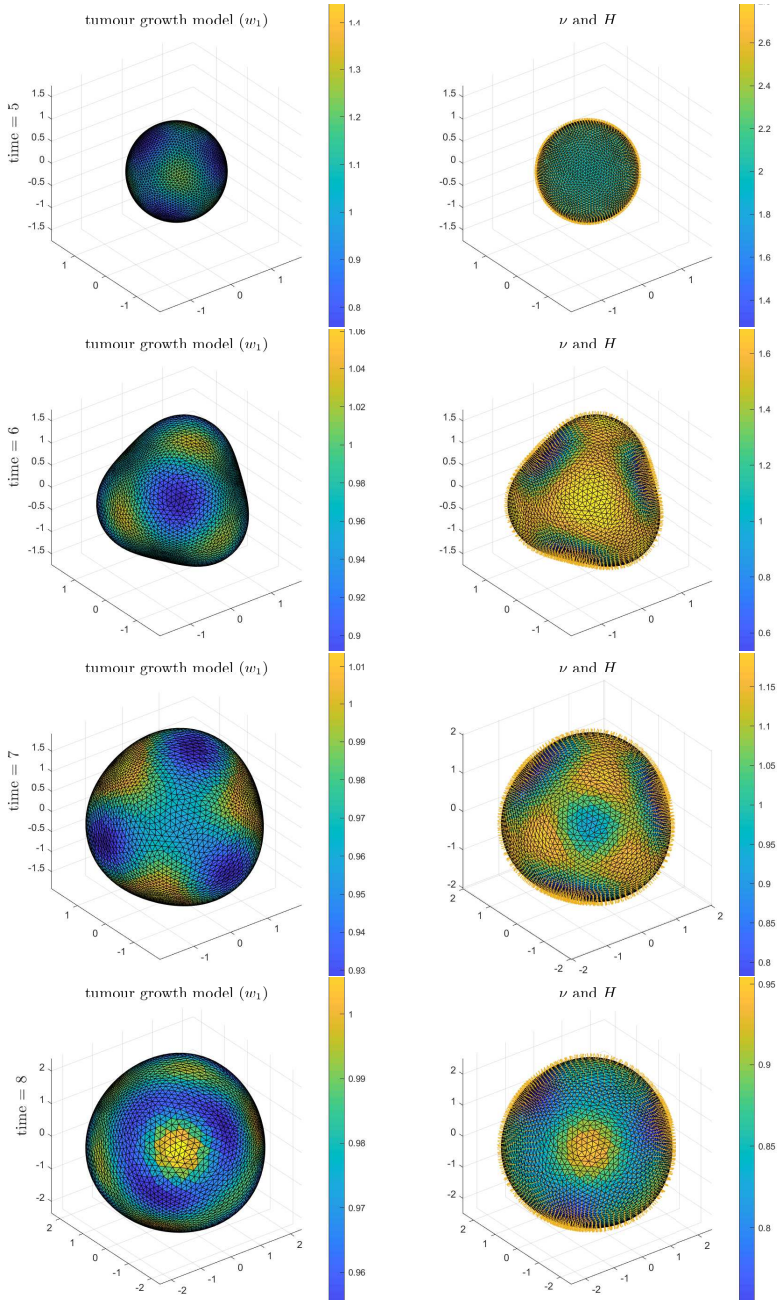

FIG. 5. Evolution of the solution (u_1), normal vector and mean curvature for tumour growth model with $\gamma = 30$ at time $t = 5, 6, 7, 8$; dof 3882

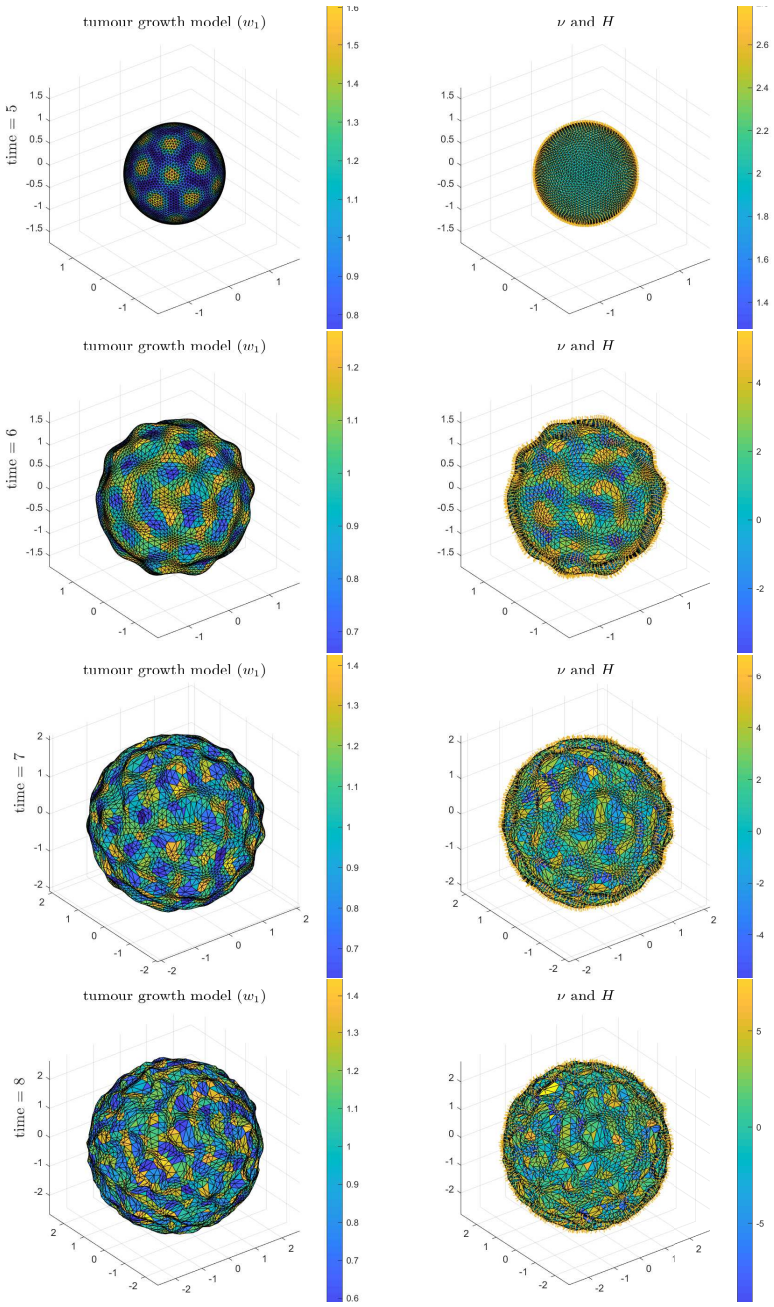

FIG. 6. Evolution of the solution (u_1), normal vector and mean curvature for tumour growth model with $\gamma = 300$ at time $t = 5, 6, 7, 8$; dof 3882

columns) match nicely (note the random effects in generating initial data) to previously reported results.

In spite of the smoothing effect of the mean curvature flow, for some more complicated examples it would be beneficial to use an algorithm which allows the tangential motion of the surface nodes, for example based on the DeTurck trick [19], or on the velocity law $v \cdot \nu = V$, e.g., [4, 5], or on ALE techniques [20, 26, 30]. However, in our experiments – both here and in [27] – this was not found necessary.

Acknowledgement The work of Balázs Kovács and Christian Lubich is supported by Deutsche Forschungsgemeinschaft, SFB 1173. The work of Buyang Li is partially supported by an internal grant (Project ZZKQ) of The Hong Kong Polytechnic University.

References

1. Akrivis, G. & and Lubich, C., Fully implicit, linearly implicit and implicit–explicit backward difference formulae for quasi-linear parabolic equations. *Numer. Math.* **131** (2015), 713–735. Zbl1334.65124 MR3422451
2. Barreira, R., Elliott, C. M., & Madzvamuse, A., The surface finite element method for pattern formation on evolving biological surfaces. *J. Math. Biol.* **63** (2011), 1095–1119. Zbl1234.92007 MR2855805
3. Barrett, J., Deckelnick, K. & Styles, V., Numerical analysis for a system coupling curve evolution to reaction diffusion on the curve. *SIAM J. Numer. Anal.* **55** (2017), 1080–1100. Zbl1365.65218
4. Barrett, J., Garcke, H., & Nürnberg, R., On the variational approximation of combined second and fourth order geometric evolution equations. *SIAM J. Sci. Comput.* **29** (2007), 1006–1041. Zbl1148.65074 MR2318696
5. Barrett, J., Garcke, H., & Nürnberg, R., On the parametric finite element approximation of evolving hypersurfaces in $\mathbb{R}^3$. *J. Comput. Phys.* **227** (2008), 4281–4307. Zbl1145.65068 MR3639582
6. Barrett, J., Garcke, H., & Nürnberg, R., Parametric finite element approximations of curvature driven interface evolutions. *Handbook of Numerical Analysis* **21** (2020), 275–423. Zbl07224912
7. Chaplain, M., Ganesh, M., & Graham, I., Spatio-temporal pattern formation on spherical surfaces: Numerical simulation and application to solid tumour growth. *J. Math. Biol.* **42** (2001), 387–423. Zbl0988.92003 MR1842835
8. Crampin, E. J., Gaffney, E. A. & Maini, P. K., Reaction and diffusion on growing domains: Scenarios for robust pattern formation. *Bull. Math. Biol.* **61** (1999), 1093–1120. Zbl1323.92028
9. Crampin, E. J., Gaffney, E. A. & Maini, P. K., Mode-doubling and tripling in reaction-diffusion patterns on growing domains: A piecewise linear model. *J. Math. Biol.* **44** (2002), 107–128. Zbl1016.35035 MR1889906
10. Deckelnick, K., Dziuk, G., & Elliott, C. M., Computation of geometric partial differential equations and mean curvature flow. *Acta Numerica* **14** (2005), 139–232. Zbl1113.65097 MR2168343
11. Deckelnick, K., Elliott, C., & Styles, V., Numerical diffusion-induced grain boundary motion. *Interfaces Free Bound.* **3** (2001), 393–414. Zbl0991.35095 MR1869586
12. Demlow, A., Higher–order finite element methods and pointwise error estimates for elliptic problems on surfaces. *SIAM J. Numer. Anal.* **47** (2009), 805–807. Zbl1195.65168 MR2485433
13. Dziuk, G., Finite elements for the Beltrami operator on arbitrary surfaces. *Partial Differential Equations and Calculus of Variations*, Lecture Notes in Math., 1357. Springer, Berlin, pages 142–155, 1988. Zbl0663.65114 MR0976234
14. Dziuk, G. An algorithm for evolutionary surfaces. *Numer. Math.* **58** (1990), 603–611, 1990. Zbl0714.65092 MR1083523

15. Dziuk, G. & Elliott, C., Finite elements on evolving surfaces. *IMA J. Numer. Anal.* **27** (2007), 262–292. Zbl1120.65102 MR2317005
16. Dziuk, G. & Elliott, C., Finite element methods for surface PDEs. *Acta Numerica* **22** (2013), 289–396. Zbl1296.65156 MR3038698
17. Ecker, K., *Regularity Theory for Mean Curvature Flow*. Springer, Berlin, 2012.
18. Eilks, C. & Elliott, C., Numerical simulation of dealloying by surface dissolution via the evolving surface finite element method. *J. Comput. Phys.* **227** (2008), 9727–9741. Zbl1149.76027
19. Elliott, C. & Fritz, H., On approximations of the curve shortening flow and of the mean curvature flow based on the DeTurck trick. *IMA J. Numer. Anal.* **37** (2017), 543–603. Zbl1433.65219 MR3649420
20. Elliott, C. M. & Venkataraman, C., Error analysis for an ALE evolving surface finite element method. *Numerical Methods for Partial Differential Equations* **31** (2015), 459–499. Zbl1318.65063 MR3312128
21. Erlebacher, J., Aziz, M., Karma, A., Dimitrov, N., & Sieradzki, K., Evolution of nanoporosity in dealloying. *Nature* **410** (2001), 450.
22. Eyles, J., King, J. F., & Styles, V., A tractable mathematical model for tissue growth. *Interfaces Free Bound.* **21** 92019), 463–493. Zbl 07167800 MR4046018
23. Fife, P., Cahn, J., & Elliott, C., A free-boundary model for diffusion-induced grain boundary motion. *Interfaces Free Bound.* **3** (2001), 291–336. Zbl0986.35136 MR1843589
24. Hairer, E. & Wanner, G., *Solving Ordinary Differential Equations II. Stiff and Differential-Algebraic Problems*. Springer, Berlin, Second edition, 1996. Zbl0859.65067 MR1439506
25. Huisken, G., Flow by mean curvature of convex surfaces into spheres. *J. Differential Geometry* **20** (1984), 237–266. Zbl0556.53001 MR0772132
26. Kovács, B., Computing arbitrary Lagrangian Eulerian maps for evolving surfaces. *Numer. Methods Partial Differ. Equations* **35** (2019), 1093–1112. Zbl1418.65176 MR3940700
27. Kovács, B., Li, B., & Lubich, C., A convergent evolving finite element algorithm for mean curvature flow of closed surfaces. *Numer. Math.* **143** (2019), 797–853. Zbl1427.65250 MR4026373
28. Kovács, B., Li, B., Lubich, C., & Power Guerra, C., Convergence of finite elements on an evolving surface driven by diffusion on the surface. *Numer. Math.* **137** (2017), 643–689. Zbl1377.65131 MR3712288
29. Kovács, B. & Lubich, C., Linearly implicit full discretization of surface evolution. *Numer. Math.* **140** (2018), 121–152. Zbl1401.65105 MR3832984
30. Kovács, B. & Power Guerra, C., Higher-order time discretizations with ALE finite elements for parabolic problems on evolving surfaces. *IMA J. Numer. Anal.* **38** (2018), 460–494. Zbl1406.65087 MR3800029
31. Lubich, C. & Mansour, D., Variational discretization of wave equations on evolving surfaces. *Math. Comp.* **84** (2015), 513–542.
32. Lubich, C., Mansour, D., & Venkataraman, C., Backward difference time discretization of parabolic differential equations on evolving surfaces. *IMA J. Numer. Anal.* **33** (2013), 1365–1385. Zbl1401.65108 MR3119720
33. Persson, P.-O. & Strang, G., A simple mesh generator in MATLAB. *SIAM Review* **46** (2004), 329–345. Zbl1061.65134 MR2114458
34. Pozzi, P. & Stinner, B., Curve shortening flow coupled to lateral diffusion. *Numer. Math.* **135** (2017), 1171–1205. Zbl1369.65111 MR3621828
35. Styles, V., An evolving surface finite element method for the numerical solution of diffusion induced grain boundary motion. In *Numerical Mathematics And Advanced Applications 2011*, pages 469–477. Springer, Heidelberg, 2013. Zbl1267.65139 MR3000495
36. Walker, S. W., *The Shape of Things: A Practical Guide to Differential Geometry and the Shape Derivative*. SIAM, Philadelphia, 2015. Zbl1336.53001 MR3486164