A space-time adaptive finite element method with exponential time integrator for the phase field model of pitting corrosion

Huadong Gao[a], Lili Ju[b,*], Xiao Li[b,c], Ravindra Duddu[d]

[a]*School of Mathematics and Statistics and Hubei Key Laboratory of Engineering Modeling and Scientific Computing, Huazhong University of Science and Technology, Wuhan, Hubei 430074, China*
[b]*Department of Mathematics, University of South Carolina, Columbia, SC 29208, USA*
[c]*Department of Applied Mathematics, The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong*
[d]*Department of Civil and Environmental Engineering, Vanderbilt University, Nashville, TN 37235, USA*

Abstract

In this paper we propose a space-time adaptive finite element method for the phase field model of pitting corrosion, which is a parabolic partial differential equation system consisting of a phase variable and a concentration variable. A major challenge in solving this phase field model is that the problem is very stiff, which makes the time step size extremely small for standard temporal discretizations. Another difficulty is that a high spatial resolution is required to capture the steep gradients within the diffused interface, which results in very large number of degrees of freedom for uniform meshes. To overcome the stiffness of this model, we combine the Rosenbrock–Euler exponential integrator with Crank–Nicolson scheme for the temporal discretization. Moreover, by exploiting the fact that the speed of the corroding interface decreases with time, we derive an adaptive time stepping formula. For the spatial approximation, we propose a simple and efficient strategy to generate adaptive meshes that reduce the computational cost significantly. Thus, the proposed method utilizes local adaptivity and mesh refinement for efficient simulation of the corrosive dissolution over long times in heterogeneous media with complex microstructures. We also present an extensive set of numerical experiments in both two and three dimensional spaces to demonstrate efficiency and robustness of the proposed method.

Keywords: pitting corrosion, phase field model, finite element method, adaptivity, exponential integrator, semi-implicit scheme

1. Introduction

Pitting corrosion is a form of localized corrosion that causes perforations or pits on the surface of stainless steel and aluminum alloys, which are otherwise resistant to uniform corrosion [51]. Corrosion pits can act as initiation sites for fatigue cracks leading to accelerated mechanical failure of structural components [17], and hence the occurrence of unscheduled downtime in industrial and infrastructure systems (e.g., refineries, power plants, pipelines) [37]. The development of

*Corresponding author
Email addresses: huadong@hust.edu.cn (Huadong Gao), ju@math.sc.edu (Lili Ju), xiao1li@polyu.edu.hk (Xiao Li), ravindra.duddu@vanderbilt.edu (Ravindra Duddu)

efficient and accurate approaches for predicting pitting corrosion damage is essential for avoiding unscheduled downtime and for implementing life extension strategies. While the recently proposed phase field model for pitting corrosion [38] is a powerful approach for predicting the evolution of pit morphologies, it may not be viable for simulating evolution over long times using standard time stepping schemes, such as the backward difference formula (BDF). To overcome this limitation, we present a space-time adaptive finite element method for the phase field model of pitting corrosion based on exponential time-integrators, which can enable efficient simulation over long times.

Existing physics-based (or deterministic) models for simulating pitting corrosion can be broadly classified into three types: sharp-interface moving boundary models [13, 48, 57], diffuse-interface cellular-automaton models [45, 49, 50] and diffuse-interface phase field models [33, 38–40]. Due to the vast number of studies in the literature on sharp and diffuse interface modeling approaches, we mostly restrict the discussion here to only those relevant to corrosion modeling. It is worth mentioning that recently a peridynamic model for simulating pitting corrosion was proposed in [9, 10], which could be classified as a type of diffuse interface model. The sharp-interface models either used moving mesh finite element method [48] or the extended finite element method coupled with the level set method [13, 14, 57] to track the evolution of the corroding interface. However, the implementation of the sharp-interface models can be cumbersome due to algorithmic complexity, especially, in three dimensional spaces. The diffuse-interface cellular-automaton models have produced interesting results; however, they employ *ad hoc* schemes to determine mass transport and interface location [45, 49]. The diffuse-interface phase field method has been widely used to solve problems with evolving phase interface geometries [6, 8], such as the solidification of materials [5, 32], superconductivity [12, 22], dendrite growth [60, 61]. While the phase field method provides a thermodynamically-consistent approach, simulating these models over long times can be computationally expensive, particularly, when the phase field equations are highly stiff.

For illustration, we consider the phase field model proposed by Mai et al. [38]. This pitting corrosion model can be viewed as a gradient flow problem of a free energy functional consisting of a phase field variable ϕ and a normalized concentration variable c. The model is described by a system of two coupled parabolic partial differential equations, see Section 2 for details. One main difficulty of its numerical solution lies in the strong stiffness of this parabolic system. For an illustration, taking all the parameters used in practice (see Table 2) into the pitting corrosion system (2.6)–(2.7) gives the following equations:

$$\begin{cases} \dfrac{\partial \phi}{\partial t} = 6.01 \times 10^{-6} \Delta \phi - 4.16 \times 10^{6} g'(\phi) + 2.06 \times 10^{8} (c - 0.04 - 0.96 h(\phi)) h'(\phi), & (1.1) \\ \dfrac{\partial c}{\partial t} = 8.50 \times 10^{-10} \Delta c - 8.50 \times 10^{-10} \Delta \big(0.96 h(\phi) + 0.04\big), & (1.2) \end{cases}$$

where $h(\phi) = -2\phi^3 + 3\phi^2$ and $g(\phi) = \phi^2(1-\phi)^2$. In the above equations, we only show three effective digits for each coefficient. By noting that the range for values of ϕ and c is $[0, 1]$, the coefficients of the reaction and diffusion terms in (1.2) are in the same magnitude of $O(10^{-10})$. On the other hand, the coefficient of the reaction term in (1.1) for ϕ is $O(10^8)$ while the diffusion coefficient is only $O(10^{-6})$, and consequently, equation (1.1) is extremely stiff with respect to ϕ. Thus, if conventional backward differentiation formula (BDF) schemes are used for the temporal discretization, one must take a very small time step size which is actually much smaller than that required to maintain sufficient accuracy.

To overcome this severe time step restriction due to the stiffness in (1.1), we shall turn to exponential time integrators, which were first used successfully in chemical physics [43, 44]. Their

applications were initially limited due to the difficulty of calculations of the product of a matrix exponential with a vector, which is particularly challenging if the dimension of the matrix is large. Since then tremendous studies have been devoted to efficient algorithms of this type of computations and achieved significant progresses, see [1–3, 20, 25–27, 46, 58]. Exponential integrator methods have been widely used for reaction-advection-diffusion systems [19, 56], Navier-Stokes equations [31, 35], shallow water equations [23], porous media flow [54], and visual computing [36]. One major advantage of the exponential integrators is their robustness and effectiveness in dealing with stiff evolution problems [28, 29]. Inspired from the above observations and discussions, we apply the exponential integrators to solve the equation (1.1) in the pitting corrosion model. Moreover, by noting that the diffusion and the reaction coefficients are of the same scale in the equation (1.2), we use the low-cost Crank–Nicolson schemes to solve it. Therefore, we are able to decouple ϕ with c in the system and successfully develop an efficient, linear, second-order accurate temporal discretization scheme. Furthermore, to accelerate the computation, we develop an adaptive strategy in time, wherein the time step size gradually increases as the speed of the corroding interface decreases with time. Thus, the proposed method is suitable for long time simulation.

The diffusion-reaction parabolic equation system defining phase field models may be solved using mesh/grid based numerical methods, namely, finite difference [21, 59], finite volume [55], Fourier spectral [16], finite element [11, 47], and isogeometric analysis [24]. However, in any mesh-based method, it is important that the mesh-size is taken small enough at the solid-liquid (i.e., metal-solution) interface to resolve the transition of the phase field ϕ and concentration c variables and to accurately capture the morphology evolution of the pit during the corrosive dissolution. Using uniform structured meshes would result in a large number of degrees of freedom (DOFs) and make long time simulation prohibitively expensive, despite the implementation of efficient time-stepping schemes. Therefore, local adaptivity and mesh refinement are still important to the efficient simulation of phase transition to capture the evolution of complex geometries in heterogeneous media (e.g., alloy microstructures with non-corroding intermetallic particles). Herein, we use an adaptive finite element mesh that is extremely fine at the interface and coarse within the bulk domains to reduce the total number of DOFs. There is a vast amount of work done on *a posterior* error estimates for phase field equations, for example, see [18, 30, 62]. A crucial step in adaptive finite element method (FEM) is to compute the local error indicators. Based on the property of the pitting corrosion model, we design a very simple, but efficient and accurate error indicator. Thus, we develop a space-time adaptive FEM for the pitting corrosion model and demonstrate its efficiency through an extensive set of numerical examples.

The rest of the paper is organized as follows: In Section 2, we provide a brief description of the phase field model for pitting corrosion. In Section 3, we propose a space-time second-order fully discrete scheme for solving the phase field model, wherein the FEM is adopted for the spatial discretization and the Rosenbrock–Euler exponential time integrator in combination with a Crank–Nicolson scheme is used for the time integration. In Section 4, a space-time adaptive strategy is developed to accelerate simulation so that is practicable for investigating corrosion evolution in representative metal/alloy microstructures with non-corroding particles. In Section 5, we present various numerical examples in both two and three dimensional spaces. The results show the efficiency and robustness of the proposed method in capturing morphological evolution during the pitting corrosion process. Finally, some concluding remarks are provided in Section 6.

2. The phase field model of pitting corrosion

In this section, we give a brief description of the phase field pitting corrosion model and refer to [33, 38] for a more detailed discussion. There are two variables in this model. A phase field variable ϕ represents the state of the material with $\phi = 1$ for the metal and $\phi = 0$ for the electrolyte, respectively, and varies continuously across the narrow diffusive interface with the thickness l. To incorporate the effect of the ions accumulation, the ion concentration c is introduced as another variable. Note here c is often normalized by c_{solid} such that $0 \leq c \leq 1$ holds. We list the nomenclature in Table 1.

Table 1: Nomenclature.

Notation	Physical interpretation
ϕ	Phase field variable
c	Normalized molar concentration
c_{solid}	Molar concentration of the solid metal
c_{sat}	Saturation concentration in the solution
c_{Se}	Normalized solid equilibrium concentration
c_{Le}	Normalized solution equilibrium concentration
α_c	Concentration gradient energy coefficient
α_ϕ	Phase field variable gradient energy coefficient
f	Local free energy density
f_{S}	Free energy density of the solid phase
f_{L}	Free energy density of the solution phase
f_{int}	Interface free energy density
$\mathcal{F}$	Free energy functional of the system
l	Interface thickness
D	Diffusion coefficient
L	Interface kinetics coefficient
A	Free energy density curvature
M	Diffusion mobility

The driving force for any phase transformation is the reduction of the system's free energy $\mathcal{F}$. In the pitting corrosion model, $\mathcal{F}$ consists of the homogeneous bulk energy $\mathcal{F}_{\text{bulk}}$ and the interface energy $\mathcal{F}_{\text{int}}$:

$$\mathcal{F}(\phi, c) = \mathcal{F}_{\text{bulk}} + \mathcal{F}_{\text{int}} = \int_\Omega \left[f(\phi(\boldsymbol{x}, t), c(\boldsymbol{x}, t)) + f_{\text{int}}(\phi(\boldsymbol{x}, t), c(\boldsymbol{x}, t)) \right] \mathrm{d}\boldsymbol{x},$$

where Ω denotes the computational domain, $\boldsymbol{x}$ denotes the spatial coordinates, t denotes the time, $f(\phi, c)$ is the local free energy density, and $f_{\text{int}}(\phi, c)$ is the excessive energy associated with the diffuse interface. The excessive interface energy arises from the inhomogeneity within the interface region, which can be written as a function of the field variable gradients

$$f_{\text{int}}(\phi, c) = \frac{\alpha_\phi}{2}(\nabla\phi)^2 + \frac{\alpha_c}{2}(\nabla c)^2,$$

where α_ϕ and α_c are the gradient energy coefficients associated with the phase field and concentration variables, respectively.

The definition of the local free energy density $f(\phi, c)$ plays a key role in identifying different models. Here, we adopt the function proposed in [33, 38]. Because the concentration at any point is evaluated as the weighted sum of the solid and liquid concentrations, the local free energy density can be represented by

$$f(\phi, c) = h(\phi) f_S(c_S) + [1 - h(\phi)] f_L(c_L) + w g(\phi), \tag{2.1}$$

where $h(\phi)$ is a C^∞-continuous interpolation function such that $h(0) = 0$ and $h(1) = 1$. As in [38], w is the height of the double-well potential $g(\phi) = \phi^2(1-\phi)^2$ and we take $h(\phi) = -2\phi^3 + 3\phi^2$. The function $h(\phi)$ is a smooth interpolating function, such that $h(\phi) = 0$ for $\phi = 0$ and $h(\phi) = 1$ for $\phi = 1$. In multiphase field models [53], this interpolation function is taken as $h(\phi) = \phi$; however, this can shift the local minima of the total free energy and may thus introduce inaccuracies that are difficult to control [41]. In this work, we simply choose the interpolation function $h(\phi) = -2\phi^3+3\phi^2$ used in the KKS model [33], which satisfies the additional condition that it has zero-slope at the equilibrium values of ϕ, that is, $h'(\phi) = 0$ for $\phi = 0$ and $\phi = 1$. Alternatively, the function can also be chosen as $h(\phi) = \phi^3(10 - 15\phi + 6\phi^2)$, which also satisfies the zero-slope condition.

In (2.1), $f_S(c_S)$ and $f_L(c_L)$ are the free energy density functions with c_S and c_L being the normalized concentrations of the coexisting solid and liquid phases, respectively. In the pitting corrosion model, the free energy densities can reasonably be approximated as

$$f_S(c_S) = A(c_S - c_{Se})^2, \quad f_L(c_L) = B(c_L - c_{Le})^2 \tag{2.2}$$

with the following assumptions

$$c = h(\phi) c_S + (1 - h(\phi)) c_L, \quad \frac{\partial f_S(c_S)}{\partial c_S} = \frac{\partial f_L(c_L)}{\partial c_L}, \tag{2.3}$$

where $c_{Se} = c_{\text{solid}}/c_{\text{solid}} = 1$ and $c_{Le} = c_{\text{sat}}/c_{\text{solid}}$ are the normalized equilibrium concentrations for the solid and liquid phases, respectively. In (2.2), A and B are the free energy density curvatures of the solid and liquid phases, respectively. The curvatures are assumed to be the same for both the solid and liquid phases in this model, i.e., $A = B$ (see [38]). Using (2.2) and (2.3) in (2.1), we get the following expression for the local energy density

$$f(\phi, c) = A\big(c - h(\phi)(c_{Se} - c_{Le}) - c_{Le}\big)^2 + w g(\phi)\,.$$

To ensure the decrease of the free energy during the corrosion process, the phase field governing equations are derived by minimizing $\mathcal{F}$ via variational differentiation [38]:

$$\begin{cases} \dfrac{\partial \phi}{\partial t}(\boldsymbol{x}, t) = -L \dfrac{\delta \mathcal{F}}{\delta \phi} = -L \left(\dfrac{\partial f}{\partial \phi} - \alpha_\phi \nabla^2 \phi \right), & (2.4) \\ \dfrac{\partial c}{\partial t}(\boldsymbol{x}, t) = \nabla \cdot M \nabla \dfrac{\delta \mathcal{F}}{\delta c} = \nabla \cdot M \nabla \left(\dfrac{\partial f}{\partial c} - \alpha_c \nabla^2 c \right), & (2.5) \end{cases}$$

where L is the interface kinetics parameter and $M = D/2A$ is the diffusion mobility for mass transport. In practice, only one of the gradient terms (∇c and $\nabla \phi$) would be sufficient to approximate the energy contribution from the diffuse interface, thus the concentration gradient energy coefficient $\alpha_c = 0$ is often assumed (see [38]). The gradient energy coefficient α_ϕ and the height of double-well potential w are correlated to the interface energy σ and interface thickness l as

$$\sigma \approx \sqrt{16 w \alpha_\phi}, \quad l = \alpha^* \sqrt{\frac{2\alpha_\phi}{w}},$$

where $\alpha^* = 2.94$ is a constant parameter. The original pitting corrosion system (2.4)–(2.5) can be represented by the following parabolic equation system:

$$\begin{cases} \dfrac{\partial \phi}{\partial t} = -L\Big(2A(c_{Le}-c_{Se})\big[c-h(\phi)(c_{Se}-c_{Le})-c_{Le}\big]h'(\phi)+wg'(\phi)-\alpha_\phi\Delta\phi\Big), & (2.6)\\ \dfrac{\partial c}{\partial t} - 2AM\Delta c = -2AM\Delta\Big(h(\phi)(c_{Se}-c_{Le})+c_{Le}\Big), & (2.7)\end{cases}$$

where $\boldsymbol{x} \in \Omega$, $t \in [0,T]$. Appropriate Dirichlet/Neumann boundary and initial conditions are supplemented to the above system depending on the problem setup.

Remark 2.1. *In the phase field governing equations (2.6)–(2.7), there are two field variables. The normalized molar concentration c is a conserved field variable constrained by mass balance because the changes in local concentration can only occur through flux of atoms between neighboring elements. In contrast, the phase-field ϕ is a phenomenological variable used to indicate which phase (metal or solution) is present at a particular position in the physical system, so it is a non-conserved variable. The evolution of the non-conserved phase-field variable is governed by a time-dependent Ginzburg–Landau equation, also referred to as the Allen–Cahn equation; whereas, that of the conserved concentration variable is governed by the Cahn–Hilliard equation. For a more detailed discussion, we refer the reader to the comprehensive review paper by Moelans, Blanpain and Wollants [42].*

3. Space-time second-order fully discrete numerical scheme

3.1. Spatial discretization

We assume Ω is a bounded Lipschitz domain in $\mathbb{R}^3$ (or polygonal domain in $\mathbb{R}^2$), which might be non-convex or multi-connected. Let $\mathcal{T}$ be a tetrahedral partition (or triangular partition in 2D) of Ω with $\Omega = \cup_{K\in\mathcal{T}} K$. The size of element K is denoted by $h_K = \operatorname{diam}(K)$, where $\operatorname{diam}(K)$ can be simply taken to be the length of the longest edge of K. Let $\{P_j\}_{j=1}^{N_h}$ be the N_h vertices of $\mathcal{T}$ and V_h be the linear Lagrange finite element space. Furthermore, let $\{\psi_j\}_{j=1}^{N_h}$ be the standard hat basis functions for the finite element space V_h. Thus, the function ψ_j is linear on each element K and $\psi_j(P_k) = \delta_{j,k}$, where $\delta_{j,k}$ is the Kronecker delta.

With the above notations and the standard Galerkin finite element method (FEM), the semi-discrete weak form of the pitting corrosion problem (2.6)–(2.7) can be stated as follows: Find $\phi_h, c_h : [0,T] \to V_h$ such that for any test function $v_h \in V_h$,

$$\begin{cases} \Big(\dfrac{\partial \phi_h}{\partial t}, v_h\Big) = -L\alpha_\phi(\nabla\phi_h,\, \nabla v_h) \\ \qquad\qquad - L\big(2A(c_{Le}-c_{Se})\big[c_h - h(\phi_h)(c_{Se}-c_{Le})-c_{Le}\big]h'(\phi_h)+wg'(\phi_h),\, v_h\big) \\ \qquad\quad := -L\alpha_\phi(\nabla\phi_h,\, \nabla v_h) + \big(F_1(\phi_h, c_h), v_h\big), & (3.1)\\ \Big(\dfrac{\partial c_h}{\partial t}, v_h\Big) = -2AM\Big(\nabla\big(c_h - h(\phi_h)(c_{Se}-c_{Le})-c_{Le}\big), \nabla v_h\Big). & (3.2)\end{cases}$$

At the initial time step, we take the Lagrange interpolation $\phi_h(0) = \Pi_h\phi_0$ and $c_h(0) = \Pi_h c_0$, where Π_h is the conventional Lagrange interpolation operator.

By using method of lines, one can directly obtain fully discrete schemes based on the above semi-discrete FEM. However, certain numerical quadrature must be used to compute the right hand side of the system (3.1)–(3.2). To simplify the implementation of the FEM and temporal discretization, we shall develop a quadrature-free approach by taking advantages of mass-lumping and interpolation techniques. To this end, let K be an element of the mesh $\mathcal{T}$ and $P_{K,j}$, $j = 1, \ldots, d+1$ be its vertices. We define the following quadrature formula

$$Q_{K,h}(u) = \text{measure}(K)\,\frac{1}{d+1}\sum_{j=1}^{d+1} u(P_{K,j}) \approx \int_K u\,\mathrm{d}\boldsymbol{x}\,, \quad \text{for a function } u \text{ defined on } \Omega,$$

and inner product

$$(u, v)_h = \sum_{K\in\mathcal{T}_h} Q_{K,h}(u\,v), \quad \text{for } u \text{ and } v \text{ defined on } \Omega. \tag{3.3}$$

Then, we can simplify the semi-discrete FEM (3.1)–(3.2) as follows: Find $\phi_h, c_h : [0, T] \to V_h$ such that for any test function $v_h \in V_h$,

$$\begin{cases} (\dfrac{\partial \phi_h}{\partial t}, v_h)_h = -L\alpha_\phi(\nabla\phi_h,\, \nabla v_h) + (F_1(\phi_h, c_h), v_h)_h\,, \\ (\dfrac{\partial c_h}{\partial t}, v_h) = -2AM\Big(\nabla\big(\Pi_h\big[c_h - h(\phi_h)(c_{Se} - c_{Le}) - c_{Le}\big]\big), \nabla v_h\Big) \end{cases}$$

with initial condition $\phi_h(0) = \Pi_h\phi_0$ and $c_h(0) = \Pi_h c_0$. Here the two unknowns can be represented as the linear combinations in terms of basis functions $\{\psi_j\}_{j=1}^{N_h}$:

$$\phi_h(t) = \sum_{j=1}^{N_h} \alpha_j(t)\psi_j, \quad c_h(t) = \sum_{j=1}^{N_h} \beta_j(t)\psi_j.$$

Let $\boldsymbol{\alpha}(t) = [\alpha_1(t), \alpha_2(t), \ldots, \alpha_{N_h}(t)]^T$ and $\boldsymbol{\beta}(t) = [\beta_1(t), \beta_2(t), \ldots, \beta_{N_h}(t)]^T$. To obtain ϕ_h and c_h, we only need to compute the coefficients $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$. The mass and stiffness matrices for the mesh $\mathcal{T}$ are defined by

$$\begin{aligned} \mathbf{M}_{i,j}:&\quad (\psi_j, \psi_i), \quad \psi_i, \psi_j \in V_h, \quad \text{for } i, j = 1, \ldots, N_h, \\ \mathbf{K}_{i,j}:&\quad (\nabla\psi_j, \nabla\psi_i), \quad \psi_i, \psi_j \in V_h, \quad \text{for } i, j = 1, \ldots, N_h. \end{aligned}$$

Furthermore, we can obtain a diagonal matrix $\widetilde{\mathbf{M}}$ by mass-lumping

$$\widetilde{\mathbf{M}}_{i,j} = \delta_{i,j}\sum_{k=1}^{N_h} \mathbf{M}_{i,k} \quad \text{for } i, j = 1, \ldots, N_h,$$

which in fact, with the help of (3.3), can be also obtained by

$$\widetilde{\mathbf{M}}_{i,j}:\quad (\psi_j, \psi_i)_h, \quad \psi_i, \psi_j \in V_h, \quad \text{for } i, j = 1, \ldots, N_h.$$

Thus, we can rewrite the semi-discrete FEM (3.1)–(3.2) into the following matrix form

$$\begin{cases} \widetilde{\mathbf{M}}\,\boldsymbol{\alpha}_t = -L\alpha_\phi\,\mathbf{K}\,\boldsymbol{\alpha} + \widetilde{\mathbf{M}}\,\boldsymbol{F}_1(\boldsymbol{\alpha}, \boldsymbol{\beta})\,, \\ \mathbf{M}\,\boldsymbol{\beta}_t = -2AM\mathbf{K}\,\boldsymbol{\beta} + 2AM(c_{Se} - c_{Le})\,\mathbf{K}\,h(\boldsymbol{\alpha})\,, \end{cases}$$

where $\boldsymbol{\alpha}_t, \boldsymbol{\beta}_t$ denote the nodal vectors of the time-derivatives, the right hand side vectors are defined by $\{\boldsymbol{F}_1(\boldsymbol{\alpha},\boldsymbol{\beta})\}_j = F_1(\alpha_j,\beta_j)$ and $\{h(\boldsymbol{\alpha})\}_j = h(\alpha_j)$, for $j = 1, 2, \ldots, N_h$. By using the fact that the mass-lumped mass matrix $\widetilde{\mathbf{M}}$ is diagonal, we can obtain $\widetilde{\mathbf{M}}^{-1}$ trivially. Hence, the above system can be further reduced to

$$\begin{cases} \boldsymbol{\alpha}_t = -L\alpha_\phi \widetilde{\mathbf{M}}^{-1}\mathbf{K}\,\boldsymbol{\alpha} + \boldsymbol{F}_1(\boldsymbol{\alpha},\boldsymbol{\beta}), & (3.4) \\ \mathbf{M}\,\boldsymbol{\beta}_t = -2AM\mathbf{K}\,\boldsymbol{\beta} + 2AM(c_{Se} - c_{Le})\,\mathbf{K}\,h(\boldsymbol{\alpha}). & (3.5) \end{cases}$$

At the initial time step, $\boldsymbol{\alpha}(0)$ and $\boldsymbol{\beta}(0)$ are computed by interpolation of ϕ_0 and c_0.

The above mass-lumped FEM (3.4)–(3.5) has many advantages for the time discretization discussed in the next section. Since the stiffness of this problem exists only in the equation for ϕ, we deal with ϕ and c differently in our time integrator. In particular, the ODE system (3.4) is of standard type $u_t = F(u)$ for which exponential integrators can be applied directly. By noting that the diffusion coefficient is of the same order as the reaction coefficient, standard Crank–Nicolson and BDF type scheme can be used to solve (3.5). We use the Matlab package `iFEM` by Long Chen [7] to do the spatial approximations. In `iFEM`, the mass and stiffness matrices can be efficiently assembled by vectoring the calculations. More importantly, the semi-discrete system (3.4)–(3.5) is quadrature-free and the time-consuming `for` loop in Matlab is avoided in the assembling procedure.

3.2. Temporal integration

In this subsection, we will derive a decoupled, second-order accurate, linear scheme in time for the above ODE system (3.4)–(3.5). A time partition $0 = t_0 < t_1 < \cdots < t_N = T$ is assumed, where $\{\tau_n = t_n - t_{n-1}\}_{n=1}^N$ is the time step size which might be non-uniform. Due to the strong stiffness of ϕ, using an efficient time integrator is crucial for a robust numerical method. The Rosenbrock–Euler integrator has been widely used to solve stiff problems, because it is simple, efficient and second order accurate. For the sake of clarity, we shall introduce this integrator using a simple ODE system defined by

$$u'(t) = F(u(t)), \quad t \in [0,T], \quad u(0) = u_0, \tag{3.6}$$

where $u \in \mathbb{R}^n$ represents the state vector and $F : \mathbb{R}^n \to \mathbb{R}^n$ is the vector field. By linearizing $F(u(t))$ at $u(t_n)$, we obtain

$$u'(t) = J_n u(t) + [F(u(t)) - J_n u(t)] := J_n u(t) + G_n(u(t)), \qquad t \in [t_n, t_{n+1}]$$

with the Jacobian $J_n := \frac{\partial F(u)}{\partial u}\big|_{u(t_n)}$. Then, $u(t_{n+1})$ can be expressed by the following integral form

$$u(t_{n+1}) = \exp(\tau_{n+1}J_n)u(t_n) + \int_0^{\tau_{n+1}} \exp((\tau_{n+1}-s)J_n)\,G_n(u(t_n+s))\,\mathrm{d}s.$$

By taking $g_n(u(t_n+s)) \approx g_n(u(t_n))$ for $s \in [0,\tau]$, we obtain the following Rosenbrock–Euler scheme

$$\begin{aligned} u_{n+1} &= \exp(\tau_{n+1}J_n)u_n + \int_0^{\tau_{n+1}} \exp((\tau_{n+1}-s)J_n)\,G_n(u_n)\,\mathrm{d}s \\ &= \exp(\tau_{n+1}J_n)u_n + \tau_{n+1}\varphi_1(\tau_{n+1}J_n)G_n(u_n)\,, \end{aligned} \tag{3.7}$$

where $\varphi_1(z) = \frac{\exp(z)-1}{z}$. The above Rosenbrock–Euler scheme (3.7) can be further rewritten as

$$u_{n+1} = u_n + \tau_{n+1}\varphi_1(\tau_{n+1}J_n)\, F(u_n)\,.$$

A local truncation error analysis shows that it is a second-order time integration method for the model problem (3.6), see [28, 36].

We now introduce the temporal discretization for (3.4)–(3.5). Our strategy is to use Rosenbrock–Euler integrator for $\boldsymbol{\alpha}$ (or ϕ) and Crank–Nicolson scheme for $\boldsymbol{\beta}$ (or c), respectively. To apply the Rosenbrock–Euler integrator to $\boldsymbol{\alpha}$, we define the Jacobian matrix of the right hand side of (3.4) with respect to $\boldsymbol{\alpha}$ by

$$\boldsymbol{J}_{\boldsymbol{\alpha}}(\boldsymbol{\alpha},\boldsymbol{\beta}) = -L\,\alpha_\phi\,\widetilde{\mathbf{M}}^{-1}\,\mathbf{K} + \frac{\partial \boldsymbol{F}_1(\boldsymbol{\alpha},\boldsymbol{\beta})}{\partial \boldsymbol{\alpha}},$$

and the function

$$\begin{aligned}\boldsymbol{G}(\boldsymbol{\alpha},\boldsymbol{\beta}) &= \left\{-L\alpha_\phi\,\widetilde{\mathbf{M}}^{-1}\,\mathbf{K}\,\boldsymbol{\alpha} + \boldsymbol{F}_1(\boldsymbol{\alpha},\boldsymbol{\beta})\right\} - \boldsymbol{J}_{\boldsymbol{\alpha}}(\boldsymbol{\alpha},\boldsymbol{\beta})\,\boldsymbol{\alpha}\\ &= \boldsymbol{F}_1(\boldsymbol{\alpha},\boldsymbol{\beta}) - \frac{\partial \boldsymbol{F}_1(\boldsymbol{\alpha},\boldsymbol{\beta})}{\partial \boldsymbol{\alpha}}\boldsymbol{\alpha},\end{aligned}$$

where $\frac{\partial \boldsymbol{F}_1(\boldsymbol{\alpha},\boldsymbol{\beta})}{\partial \boldsymbol{\alpha}}$ is a diagonal matrix. We now define the extrapolation $\widehat{\boldsymbol{\beta}}^{n+1/2} = (1+\frac{\tau_{n+1}}{2\tau_n})\boldsymbol{\beta}^n - \frac{\tau_{n+1}}{2\tau_n}\boldsymbol{\beta}^{n-1}$. It is easy to verify that the standard extrapolation $\widehat{\boldsymbol{\beta}}^{n+1/2} = \frac{3}{2}\boldsymbol{\beta}^n - \frac{1}{2}\boldsymbol{\beta}^{n-1}$ is recovered if the time step τ_n is uniform. By combining the Rosenbrock–Euler with the Crank–Nicolson schemes, a fully discrete scheme for solving $(\boldsymbol{\alpha}_h^{n+1}, \boldsymbol{\beta}_h^{n+1})$ is then obtained below for $n = 1, 2, \cdots,$

$$\begin{cases}\boldsymbol{\alpha}_h^{n+1} = \exp\left(\tau_{n+1}\boldsymbol{J}_{\boldsymbol{\alpha}}(\boldsymbol{\alpha}_h^n, \widehat{\boldsymbol{\beta}}_h^{n+1/2})\right)\boldsymbol{\alpha}_h^n + \tau\varphi_1(\tau_{n+1}\boldsymbol{J}_{\boldsymbol{\alpha}}(\boldsymbol{\alpha}_h^n, \widehat{\boldsymbol{\beta}}_h^{n+1/2}))\,\boldsymbol{G}(\boldsymbol{\alpha}_h^n, \widehat{\boldsymbol{\beta}}_h^{n+1/2}), & (3.8)\\ \mathbf{M}\left(\dfrac{\boldsymbol{\beta}_h^{n+1}-\boldsymbol{\beta}_h^n}{\tau_{n+1}}\right) + AM\mathbf{K}\,\left(\boldsymbol{\beta}_h^{n+1}+\boldsymbol{\beta}_h^n\right) = 2AM(c_{Se}-c_{Le})\,\mathbf{K}\; h\left(\dfrac{\boldsymbol{\alpha}_h^{n+1}+\boldsymbol{\alpha}_h^n}{2}\right). & (3.9)\end{cases}$$

The above scheme is decoupled as we can first solve for $\boldsymbol{\alpha}_h^{n+1}$ from (3.8), and next solve for $\boldsymbol{\beta}_h^{n+1}$ by plugging the computed $\boldsymbol{\alpha}_h^{n+1}$ into (3.9). It is also linear so that no nonlinear iteration is needed. Alternatively, one can use the exponential integrator directly to the whole system of (ϕ, c) by also lumping the mass matrix for c. However, this would yield a very large Jacobian matrix. For the proposed scheme (3.8)–(3.9), one only need to compute the product of $\exp(\tau\boldsymbol{J}_{\boldsymbol{\alpha}})$ with a vector and a smaller Jacobian matrix $\boldsymbol{J}_{\boldsymbol{\alpha}}$. Moreover, as the computation of $\boldsymbol{\alpha}_h^{n+1}$ dominates the computational effort, the cost for solving $\boldsymbol{\beta}_h^{n+1}$ is almost negligible.

To initialize the scheme (3.8)–(3.9), one can simply take a first-order temporal discretization scheme at the first time step t_1:

$$\begin{cases}\boldsymbol{\alpha}_h^1 = \exp\left(\tau_1\boldsymbol{J}_{\boldsymbol{\alpha}}(\boldsymbol{\alpha}_h^0, \boldsymbol{\beta}_h^0)\right)\boldsymbol{\alpha}_h^0 + \tau\varphi_1(\tau_1\boldsymbol{J}_{\boldsymbol{\alpha}}(\boldsymbol{\alpha}_h^0, \boldsymbol{\beta}_h^0))\,\boldsymbol{G}(\boldsymbol{\alpha}_h^0, \boldsymbol{\beta}_h^0), & (3.10)\\ \mathbf{M}\frac{\boldsymbol{\beta}_h^1-\boldsymbol{\beta}_h^0}{\tau_1} + 2AM\mathbf{K}\,\boldsymbol{\beta}_h^1 = 2AM(c_{Se}-c_{Le})\,\mathbf{K}\,h(\boldsymbol{\alpha}_h^1), & (3.11)\end{cases}$$

where $\boldsymbol{\alpha}_h^0 = \boldsymbol{\alpha}(0)$ and $\boldsymbol{\beta}_h^0 = \boldsymbol{\beta}(0)$.

4. Space-time adaptive strategy

The proposed fully discrete scheme (3.8)–(3.9) can be used to simulate the pitting corrosion with uniform spatial mesh and time step sizes. However, the efficiency of the method can be enhanced by accounting for the behavior of the field variables and its evolution over longer times. First, by observing the previous results presented in [38], we notice that steep gradients in field variables ϕ and c exist within a thin region at the corrosion (solid-liquid) interface. Second, we notice that the corrosion process becomes slower with time [13] as the pit grows due to the increase in the diffusion length. Therefore, we implement a simple space-time adaptive algorithm to accelerate simulation by exploiting the above two observations of the pitting corrosion process. In the adaptive algorithm, a finer mesh with more nodes near the corrosion interface is used to capture the phase field evolution accurately and coarse mesh is adopted elsewhere, where the phase field variable is constant, to reduce the total DOFs. Adaptive non-uniform grids/meshes are often used in the numerical simulation of phase field models in the literature [11, 16]. Also, we increase the time step size adaptively to accelerate the simulation at later stages of corrosion evolution, based on the speed of the evolving corrosion interface. In this section, we first show the numerical results generated by the scheme from solving equations (3.8)–(3.9) with the uniform spatial mesh and time step sizes in Example 4.1. These results establish the viability of the proposed method and provide a reference for the comparison with the adaptive algorithm presented later. In Table 2, we list the parameters used in all the numerical examples, which are taken from [38].

Table 2: The physical parameters of phase field model.

Parameter	Physical interpretation	Value
σ	Interface energy	10 J/m^2
l	Interface thickness	5 μm
D	Diffusion coefficient	8.5×10^{-10} m^2/s
L	Interface kinetics coefficient	2 m^3/(Js)
A	Free energy density curvature	5.35×10^7 J/mol
c_{solid}	Average concentration of metal	143 mol/L
c_{sat}	Average saturation concentration	5.1 mol/L

Example 4.1. *We use the proposed FEM to study the "pencil electrode test", where the sample is a 150 μm long metal wire with diameter $d = 25$ μm in an epoxy coat. The top end of the wire is exposed to the solution and then the metal is gradually corroded from the top side. Dirichlet boundary condition $c = 0$ and $\phi = 0$ are used at the top, while homogeneous Neumann boundary condition is assigned on the other parts. At the initial time $t = 0$, the whole domain Ω consists of the metal, with only a thin layer of the electrolyte solution on the top (see Figure 1).*

We use a uniform mesh with 26×151 nodes (i.e., with 3926 DOFs) to solve this problem, with spatial resolution of 1 μm in each direction. In the temporal direction, a uniform time step $\tau = 1.0 \times 10^{-3}$ s is used. We show the numerical results of ϕ and c at $t = 1, 38, 152$ and 225 s in Figure 1, where both ϕ and c agree with previous results in [38]. Note that in all figures of

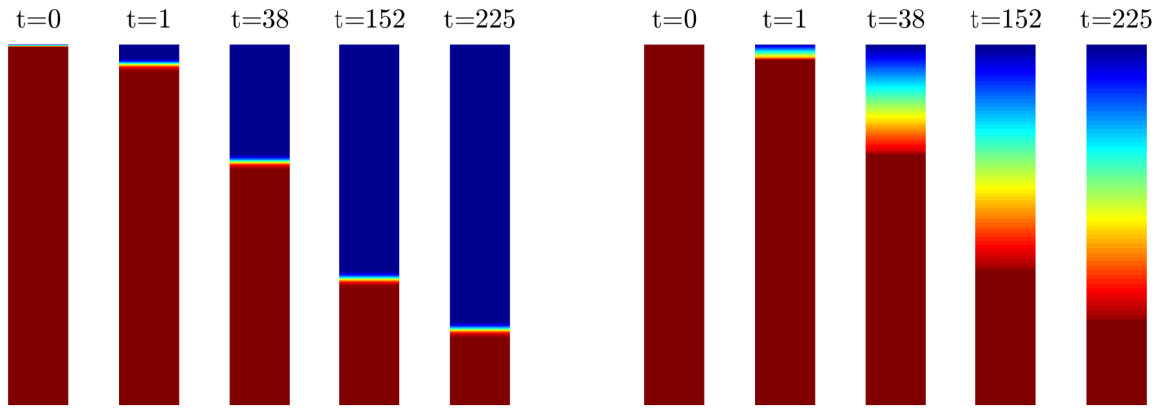

Figure 1: Numerical results of ϕ (left) and c (right) at times $t = 0, 1, 38, 152$ and 225 s with the uniform time step $\tau = 1 \times 10^{-3}$ s on a uniform mesh with resolution 1 μm. Both the concentration and phase variable are normalized, so the red color represents the value 1, the blue color corresponds to the value 0, and other colors represent the intermediate values. (Example 4.1)

the numerical results throughout this paper, the red color represents where the normalized variable takes the value 1 and the blue color corresponds to the value 0. For the terminal time $T = 225$ s, we find that the CPU time of running our code with this fixed mesh and constant time step τ is about 20148.16 seconds. Using these simulation results, we will devise the space-time adaptive algorithm. We first study the considerations for spatial resolution of the mesh, and then the appropriate choice of the time step sizes. Finally, we summarize the adaptive space-time strategy.

4.1. Spatial resolution for the sharp interface

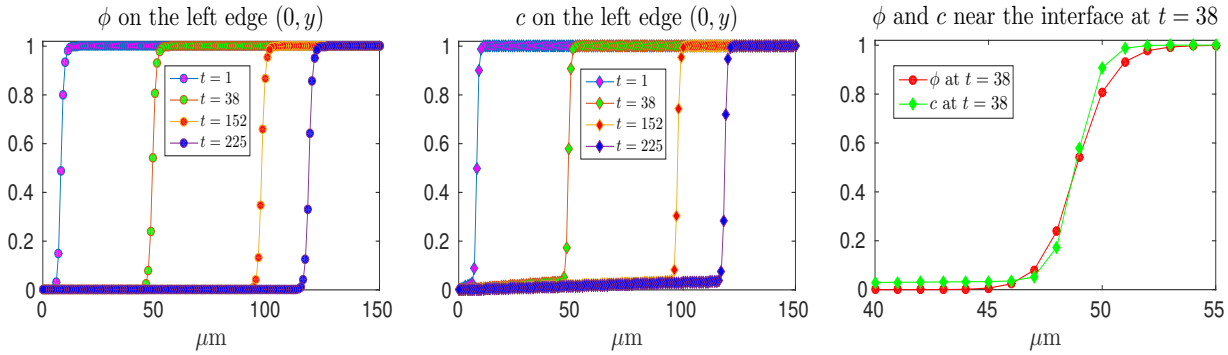

Figure 2: Numerical results of ϕ and c on the left edge with the uniform mesh resolution 1 μm and the time step $\tau = 1.0 \times 10^{-3}$ s. (Example 4.1)

In Example 4.1 the spatial resolution 1 μm is enough for approximation of (ϕ, c), as the interface thickness parameter l is 5 μm [38] (see Table 2). Here, we expect to determine the adaptive size of the spatial mesh with nodes as few as possible but without loss of accuracy to capture the interfaces. As the setting is homogeneous in the y direction, Example 4.1 indeed can be reduced to a one dimensional problem. Therefore, we can only check the morphology of the pit along the left edge $(0, y)$ with $0 \le y \le 150\mu$m. We plot ϕ and c on the left edge at $t = 1, 38, 152$ and 225 s in Figure 2. Clearly, the variables ϕ and c vary steeply near the metal–solution interface region.

As the unknowns ϕ and c are almost flat away from the narrow interface region, using a uniform structured mesh with 1 μm is clearly unnecessary. A spatially adaptive mesh would reduce the computational cost significantly. A crucial step in designing the mesh adaptive strategy is to find a good monitor to control the spatial approximation. We also show the interface region at $t = 38$ s in Figure 2. For this uniform resolution 1 μm, we find that the largest variances in one element are $\max_{K\in\mathcal{T}}\{\max \phi|_K - \min \phi|_K\} \approx 0.3$ and $\max_{K\in\mathcal{T}}\{\max c|_K - \min c|_K\} \approx 0.4$, respectively. Also, corrosion is usually an irreversible process, which implies that for any fixed $\boldsymbol{x}$, $\phi(\boldsymbol{x}, t)$ and $c(\boldsymbol{x}, t)$ is decreasing monotonically as time evolves until the whole metal is corroded. Therefore, our idea is to use element variance of the concentration variable

$$\eta_K := \max_K c - \min_K c \tag{4.1}$$

as indicator to control the spatial approximation. We will show by extensive numerical examples that this simple element indicator (4.1) is able to control the mesh size efficiently and ensures adequate accuracy.

4.2. Choice of the time step sizes

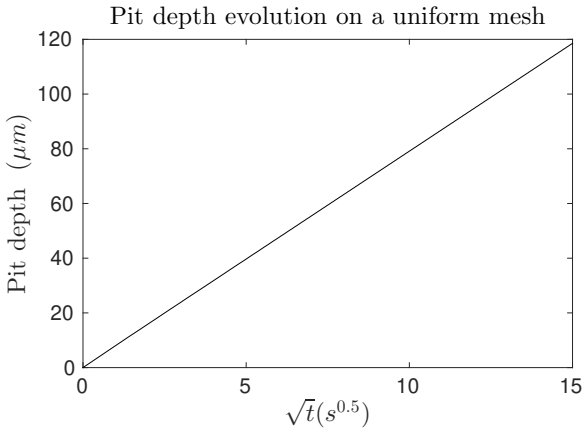

Figure 3: Pit depth evolution computed with $\tau = 1.0 \times 10^{-3}$ s on a uniform mesh with resolution 1 μm. (Example 4.1)

We examine the choice of the time step sizes by considering the speed of the pit interface motion. The plot of the pit depth ($\phi = 0.5$) is shown in Figure 3. Evidently, the pit depth varies linearly with the square root of time, which agrees with previous results in [13, 38]. This observation indicates that the speed of the diffusion-controlled corrosion process decreases in time due to the increased length of the corrosion cavity. Consequently, the evolution of the pit morphology becomes slower, although there is no stationary state for this problem. Therefore, it is inefficient to use a uniform time step size, especially, for long time simulations. The overall simulation can be accelerated significantly by using variable time step sizes $\{\tau_n\}_{n=1}^N$.

Herein, we propose a strategy for the adaptive time stepping such that during each time interval $[t_n, t_{n+1}]$, the pit interface moves by the same distance, that is,

$$\tau_{n+1} v_n = \text{constant},$$

where v_n is the interface speed and $\tau_{n+1} = (t_{n+1} - t_n)$ is the time step size at time t_n. In practical simulations, we use the uniform small time step size $\hat{\tau} = 1.0 \times 10^{-3}$ s for $t_n \leq 1$ to ensure accuracy and avoid any numerical instability, and adopt the adaptive time stepping scheme for $t_n > 1$ to accelerate the simulation. Denoting the interface speed at $t = 1$ by $\hat{v}$, we can express the formula for the adaptive time step size as follows:

$$\tau_{n+1} = \begin{cases} \hat{\tau}, & \text{if } t_n \leq 1, \\ \hat{\tau}\hat{v}/v_n, & \text{if } t_n > 1. \end{cases} \tag{4.2}$$

From the sharp interface formulation [13], it is known that the velocity of the interface is related to the gradient of the concentration field c. Because we adopt the piecewise linear finite element for spatial approximation, the gradient ∇c_h is constant in each element. To make the adaptive time stepping strategy applicable for a generalized interface evolution scenario, we determine the speed of interface Γ at time t_n as

$$v_n = \max_{\substack{K \in \mathcal{T} \\ K \cap \Gamma \neq \emptyset}} |\nabla c_h(K, t_n)|.$$

Thus, we automate the determination of the time step size based on the maximum interface speed at any particular time t_n.

4.3. Space-time adaptive strategy

We now present the space-time adaptive strategy for the system of equations (3.8)–(3.9). Let $h_{\min}$ and $h_{\max}$ denote the minimum and maximum element sizes used in the computation, respectively, which are prescribed beforehand. The space-time adaptive algorithm is given below:

Step 1. Determine an initial mesh $\mathcal{T}^0$ and initial approximations ϕ_h^0 and c_h^0, such that

$$\max_{K \in \mathcal{T}^0} h_K \leq h_{\max}, \quad \min_{K \in \mathcal{T}^0} h_K \geq h_{\min}, \quad \max_{K \in \mathcal{T}^0} \eta_K < 0.4.$$

Step 2. Take a time step $\tau_1 = 1.0 \times 10^{-3}$ s and compute ϕ_h^1 and c_h^1 by (3.10)–(3.11) and then adaptively refine/coarsen the mesh and perform interpolation as follows:

(2.a) Calculate the element indicator η_K for c_h^1 at t_1 by formula (4.1) and refine the element K if

$$h_K > 2h_{\min} \quad \text{and} \quad \eta_K > 0.4. \tag{4.3}$$

After this refinement process, we obtain a finer mesh $\widehat{\mathcal{T}}^1$. Interpolate ϕ_h^0, ϕ_h^1, c_h^0 and c_h^1 from $\mathcal{T}^0$ to $\widehat{\mathcal{T}}^1$.

(2.b) Calculate the element indicator η_K again for each $K \in \widehat{\mathcal{T}}^1$ by formula (4.1) and coarsen the element K if

$$h_K < h_{\min}/2 \quad \text{or} \quad (h_K < h_{\max}/2 \quad \text{and} \quad \eta_K < 0.02)\,. \tag{4.4}$$

After this coarsening process, we obtain a new mesh $\mathcal{T}^1$. Interpolate ϕ_h^0, ϕ_h^1, c_h^0 and c_h^1 from $\widehat{\mathcal{T}}^1$ to $\mathcal{T}^1$.

Step 3. For $n \geq 2$, execute the following iteration:

(3.a) Determine the time step size τ_n by the formula (4.2), then compute ϕ_h^n and c_h^n by (3.8)-(3.9) based on previous results ϕ_h^{n-2}, c_h^{n-2}, ϕ_h^{n-1} and c_h^{n-1} defined on the mesh $\mathcal{T}^{n-1}$.

(3.b) Calculate the element indicator η_K for c_h^n at t_n by formula (4.1) and then refine $\mathcal{T}^{n-1}$ by the refinement rule (4.3) to obtain $\widehat{\mathcal{T}}^{n-1}$. Interpolate ϕ_h^{n-1}, c_h^{n-1}, ϕ_h^n and c_h^n from $\mathcal{T}^{n-1}$ to the refined mesh $\widehat{\mathcal{T}}^{n-1}$.

(3.c) Calculate the element indicator η_K again for c_h^n on the refined mesh $\widehat{\mathcal{T}}^{n-1}$ by formula (4.1) and coarsen the element K by the coarsening rule (4.4) to obtain $\mathcal{T}^n$. Interpolate ϕ_h^{n-1}, c_h^{n-1}, ϕ_h^n and c_h^n from $\widehat{\mathcal{T}}^{n-1}$ to $\mathcal{T}^n$. If $t_n < T$, set $n = n + 1$ and go to (3.a); otherwise, stop iteration.

Remark 4.1. *It should be pointed out that Mai et al. [38] solved the two nonlinear PDEs (2.6)–(2.7) of phase field model in a fully coupled (implicit) manner, whereas in our method we decouple the two equations and through mass lumping and explicit time differencing schemes (extrapolation) we only have to solve a linear system. Moreover, in Mai et al. [38] adaptive time stepping is used, so that if the nonlinear solver takes too many iterations to converge they reduce the time step and restart the nonlinear solve. Because we solve a fully linear discrete system given by (3.8)-(3.9), we do not need nonlinear iterations. Instead, through the adaptive time stepping we improve computational efficiency based on the observation that the speed of the corroding interface decreases as pits/crevices grow deeper. Thus, the proposed space-time finite element method is quite efficient not only due to space-time adaptivity but also because it avoids nonlinear iterations.*

5. Numerical experiments

In this section, we provide a broad spectrum of numerical experiments to test the proposed space-time adaptive FEM for both two and three dimensional problems, including model verification and comparison studies [13, 38]. The mesh adaptivity and matrix assembling is done in the Matlab package `iFEM` by Long Chen [7], which uses an efficient vectorial assembling procedure. The inherent refinement and coarsening functions in `iFEM` are used to adapt the mesh at each time step. The package `Expokit` (Matlab version) by Sidje [52] is used for computing the action of the matrix functions $\exp(\tau A)$ and $\varphi_1(\tau A)$ on a vector. All computations in this paper were done on a Linux laptop with a four-core Intel 2.5 GHz Processor and 7.9 GB Memory.

5.1. Numerical results in two-dimensional space

Example 5.1. *We use the proposed space-time adaptive FEM to solve the benchmark problem in Example 4.1. Here, we set* $h_{\min} = 1$ μm *and* $h_{\max} = 15$ μm. *We show in Figure 4 the numerical results of* ϕ *at* $t = 0$, 38, 152, *and* 225 *s. The pit depth (*$\phi = 0.5$*) evolution in time is plotted in Figure 5. As expected, the pit depth varies linearly with respect to* $\sqrt{t}$. *The results agree well with those in Example 4.1, the previous ones in [38, Section 3.1], and the analytic solution in [51]. For comparison, we also plot the experiment data in Figure 5 which is taken from [15]. Thus, this example both verifies and validates our FEM implementation.*

It is no surprise that the computational cost of the proposed space-time adaptive FEM is significantly less than that of the method in Example 4.1 with uniform mesh and time step size. As shown in Figure 4, the adaptive mesh has less than 200 DOFs, whereas the uniform mesh has 3926 DOFs. For a terminal time $T = 225$, we find that the CPU time of running our code is 936.91

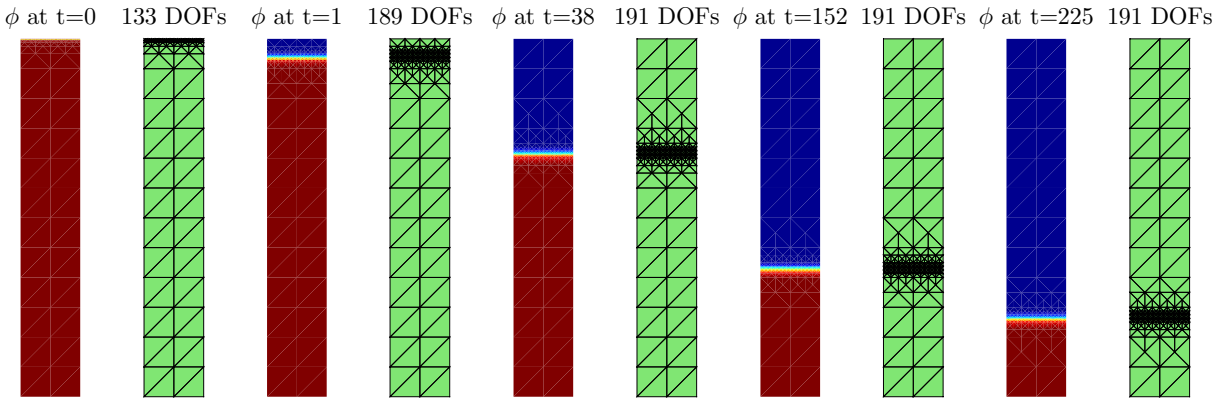

Figure 4: Numerical results of ϕ and meshes obtained by the proposed space-time adaptive FEM. (Example 5.1)

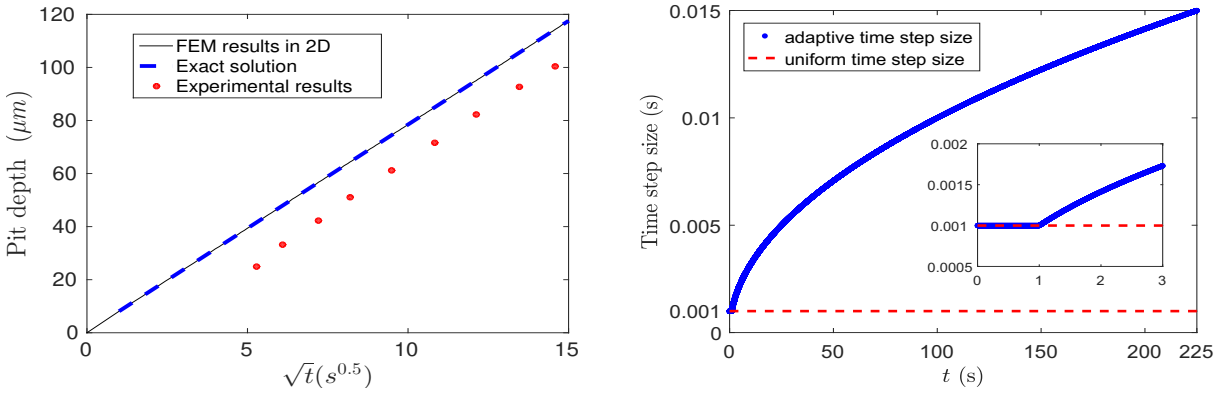

Figure 5: Pit depth evolution (left) computed by the proposed space-time adaptive FEM along with the evolution of the time step sizes (right). (Example 5.1)

seconds which is only 4.65% of that in Example 4.1 (20148.16 seconds) and about 12.87% of that for the simulation with adaptive space meshing and uniform time stepping (7279.38 seconds). The evolutions of adaptive and uniform time step sizes are shown by the right picture in Figure 5. The adaptive time step sizes determined from (4.2) are ten more times larger than the uniform one when $t \geq 100$ s and numerical stability is still maintained during the computations, owing to the use of the exponential time integration scheme. Although there is some overhead cost from mesh adapting and assembling, the overall computational cost of the space-time adaptive FEM is less. It should be noted that for rectangular domains, a uniform-grid-based finite difference scheme with Fast Fourier Transform (FFT) may be quite efficient, although we did not compare the computational cost of FEM and FFT based methods. Here, we use the FEM because it is more suitable for complicated geometries and general boundary conditions encountered in corrosion simulations.

Example 5.2. *We use the proposed space-time adaptive FEM to investigate the evolution of the semi-circular pit growth. The sample size is* 400 $\mu m \times$ 200 μm. *A semi-circular opening with diameter* 16 μm *at the center of the top edge exposes the metal to bulk solution environment. Homogeneous Dirichlet boundary conditions* $\phi = c = 0$ *are applied on this opening, whereas, homogeneous Neumann boundary condition is prescribed on the rest of the external boundary. This*

benchmark example has been previously studied in [13, 38] and qualitatively validates our implementation against the experiment conducted in [15].

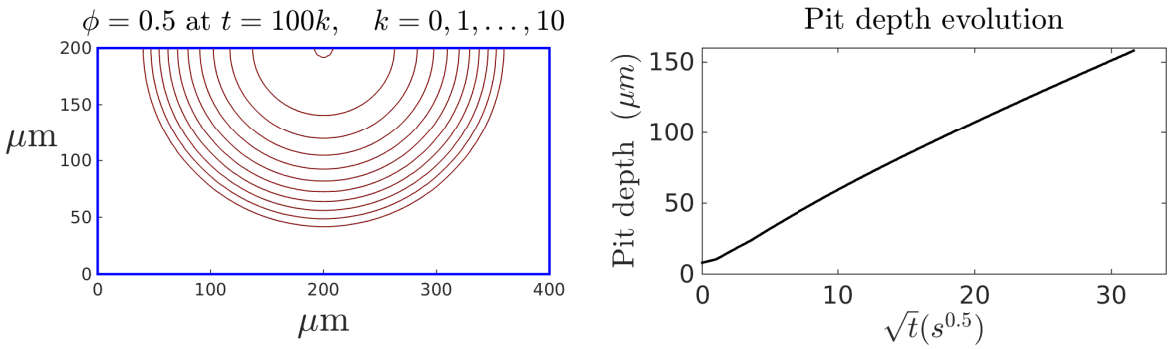

Figure 6: Iso-isoface ($\phi = 0.5$) of the pit evolution (left) and pit depth evolution (right). (Example 5.2)

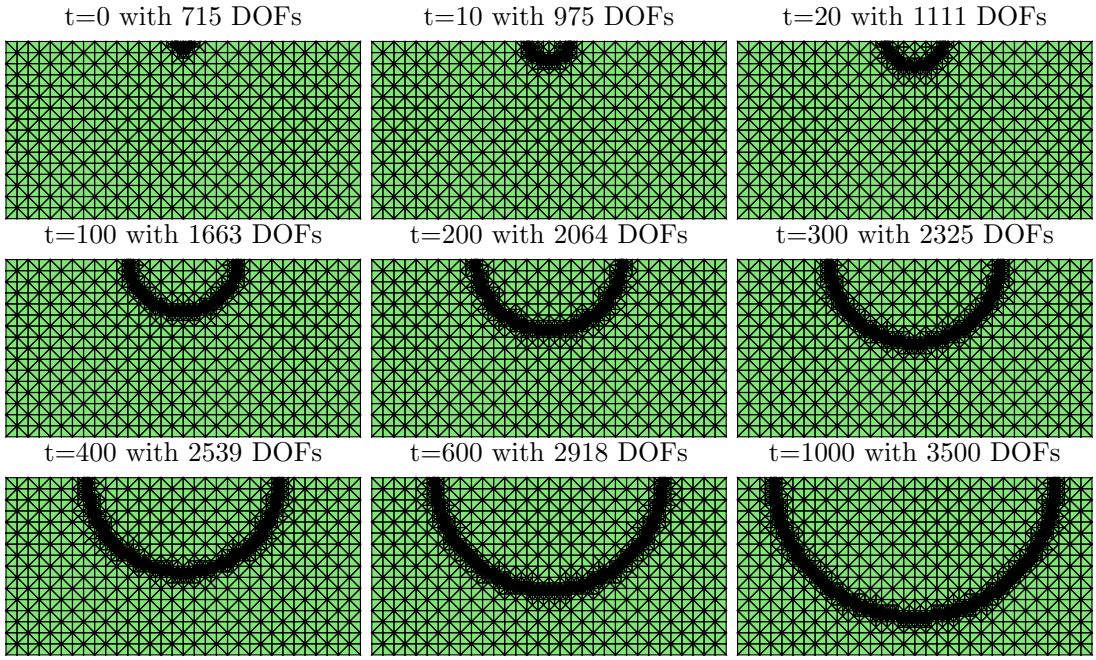

Figure 7: The adaptive meshes generated by the proposed method at times $t = 0$, 10, 20, 100, 200, 300, 400, 600 and 1000 s. (Example 5.2)

We set $h_{\min} = 1$ μm and $h_{\max} = 25$ μm in this example. The CPU time of running our code till the terminal time $T = 1000$ s is 2 hours and 9 minutes. The plots of the iso-surface $\phi = 0.5$ at time $t = 100\,k$ with $k = 0, 1, \ldots, 10$, and the pit depth (at $\phi = 0.5$) evolution in time is shown in Figure 6. Our numerical results in Figure 6 show that the shape of the corrosion pit remains semi-circular. This is because of three reasons: (1) corrosion process is purely diffusion-controlled; (2) the metal domain is homogeneous without any large heterogeneities; and (3) the metal is only exposed to the corrosive solution only through a small semi-circular opening on the top boundary

of the domain. It is important to note that even though this 2D example is an idealization of the experiment conducted by Ernst and Newman [15], the pit shapes are in good agreement with those observed in experiments. We also observe that the pit depth increases linearly with the square root of time $\sqrt{t}$. However, the speed of pit moving is slower than that in Example 5.1, because gradient of concentration near the interface is lesser during semi-circular pit growth.

In Figure 7, we show the adaptive meshes at time $t = 0$, 10, 20, 100, 200, 300, 400, 600 and 1000 s. The number of DOFs gradually increases as time evolves, because the length of the corrosion interface increases. In the adaptive mesh at the terminal time $T = 1000$ s there are only 3500 DOFs, which is much smaller than that in a uniform mesh with 1 μm resolution (80601 DOFs). Thus, the computational efficiency of the proposed space-time adaptive FEM stems from the reduction of number of DOFs.

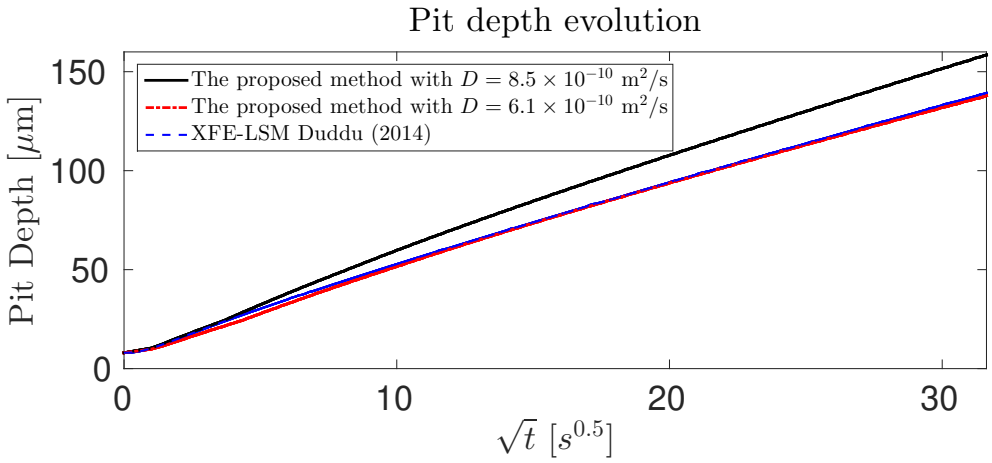

Figure 8: The pit depth evolution obtained by the proposed method for $D = 8.5 \times 10^{-10}$ and $D = 6.1 \times 10^{-10}$ compared with Duddu's previous results in [13]. (Example 5.2)

For comparison, we plot the evolution of pit depth in Figure 8 with $D = 8.5 \times 10^{-10}$ m^2/s and $D = 6.1 \times 10^{-10}$ m^2/s, and also the sharp interface result of Duddu [13]. Because the metal ions only diffuse in the solution phase and the diffusion in the solid phase is negligible, we need to define $D = 8.5 \times 10^{-10}$ m^2/s in the liquid phase and $D = 0$ in the solid phase. In the phase field model of Mai et al. [38], the diffusion coefficient was interpolated as $D = 8.5 \times 10^{-10}(1-\phi)$ m^2/s. Obviously, taking a constant value of $D = 8.5 \times 10^{-10}$ m^2/s will lead to a slight over prediction of the interface velocity and pit depth. Instead, taking a smaller constant value of $D = 6.1 \times 10^{-10}$ m^2/s gives a better match with the sharp interface result of Duddu [13], as shown in Figure 8. Nevertheless, we note that proposed space-time FEM can accommodate nonlinear diffusion equation with some minor changes.

Example 5.3. *We use the proposed space-time adaptive FEM to simulate the electropolishing process, which is widely used to eliminate the surface roughness of metallic materials in industry [34, 38]. Here we consider a* 350 μm $\times$ 150 μm *rectangular domain where the rough metal surface is exposed to the solution, see the first sub-figure in Figure 9. We set homogeneous Dirichlet boundary condition for* ϕ *and* c *on the top boundary and homogeneous Neumann boundary condition on the rest of the external boundary. This example demonstrates the ability of our numerical implementation to handle the evolution of arbitrarily shaped features on the corrosion interface.*

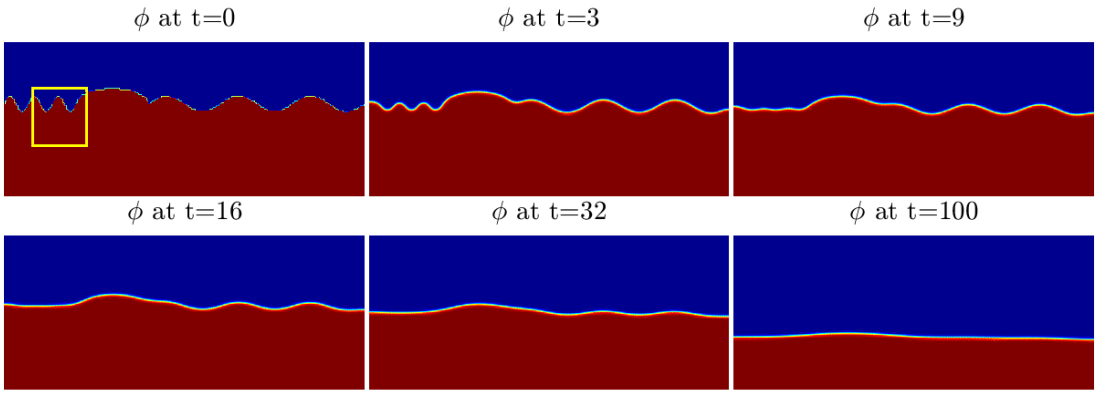

Figure 9: Numerical results of ϕ obtained by the proposed method at $t = 0$, 3, 9, 16, 32 and 100 s. (Example 5.3)

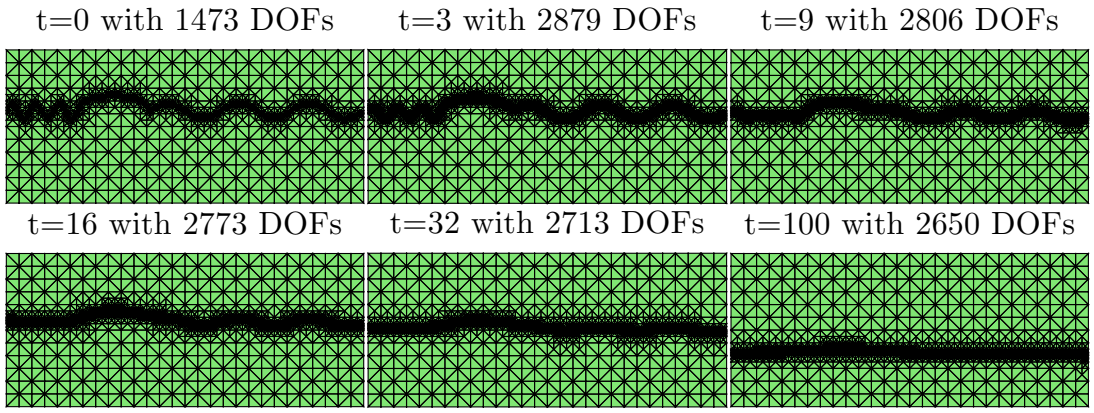

Figure 10: Meshes generated by the proposed method at $t = 0$, 3, 9, 16, 32 and 100 s. (Example 5.3)

In the computation, we set $h_{\min} = 1$ μm and $h_{\max} = 15$ μm. The CPU time of running our code till the terminal time $T = 100$ s is 55 minutes. Numerical results of ϕ at $t = 0$, 3, 9, 16, 32, 100 s are shown in Figure 9. We can see clearly that as time evolves, the rough surface is electrochemically polished into a smooth flat surface without applying mechanical abrasion. We show the adaptive meshes generated by the proposed method in Figure 10. The number of DOFs used in the adaptive mesh is much less than that of the uniform meshes (53001 DOFs) with spatial resolution 1μm. In Figure 11, we show a zoomed-in picture of local mesh and the iso-surface corresponding to $\phi = 0.5$. We can see that the adaptive meshing algorithm is able to provide sufficient resolution needed to accurately simulate the evolution of the phase field variable.

Example 5.4. *We use the proposed space-time adaptive FEM to investigate pitting corrosion evolution in a composite material or alloy with non-corroding particles in the microstructure. Here we consider a* 200 $\mu m \times$ 140 μm *square specimen with embedded non-corroding (ceramic) particles, yellow regions in Figure 12. Zero flux boundary condition for* ϕ *and* c *is applied on the particle-metal interfaces, as the particles are non-corroding. The geometries of the particles and pit morphology can have complex shapes, and the adaptive mesh is able to capture them accurately. To initialize, we set a* 10 μm *semi-circular opening on the top boundary, where* $\phi = c = 0$ *and* $\phi = c = 1$ *in the*

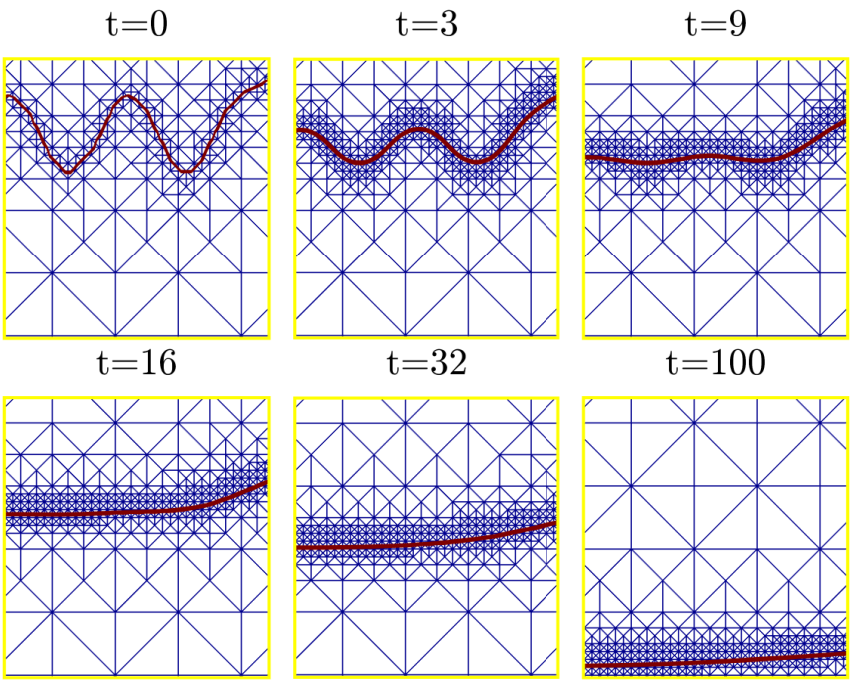

Figure 11: Local meshes and iso-surface of $\phi = 0.5$ at $t = 0$, 3, 9, 16, 32 and 100 s. (Example 5.3)

rest of the domain. Homogeneous Dirichlet boundary condition is applied on the initial pit opening on the top boundary, whereas homogeneous Neumann boundary condition is applied on the rest of external boundary and the particle-metal interface. This example demonstrates the robustness and the stability of our implementation in simulating the evolution of complex pit shapes in alloy microstructures with large time steps.

We first generate an adaptively refined mesh by choosing $h_{\min} = 1$ μm and define the arbitrarily shaped regions of non-corroding particles. Starting with this adaptive mesh we perform computations till terminal time $T = 100$ s and the CPU time of running our code is around 3 hours and 18 minutes. We plot the snapshots of numerical results at $t = 1$, 10, 50, 100, 250, 350, 500, 750 and 1000 s in Figure 12. Due to the complex shapes of the particles and the nonlinear nature of the corrosion process, the evolution of the pit morphology is quite intricate and interesting. Figure 13, plots the mesh generated by our algorithm, where the number of DOFs is less than 3000 at any time during the simulation.

5.2. Numerical results in three-dimensional space

In this subsection, we provide four numerical examples in three-dimensional space. Examples 5.5 and 5.6 are the 3D versions of Examples 5.1 and 5.2, respectively. The phase field model presented in Section 2 naturally extends to three dimensions, so we still use the same model parameters listed in Table 2. In 5.7 and 5.8, our intention is to only demonstrate the suitability of the space-time FEM for simulating the evolution of corrosion pits in 3D. However, detailed studies involving model calibration/validation against experiments is beyond the scope of this article.

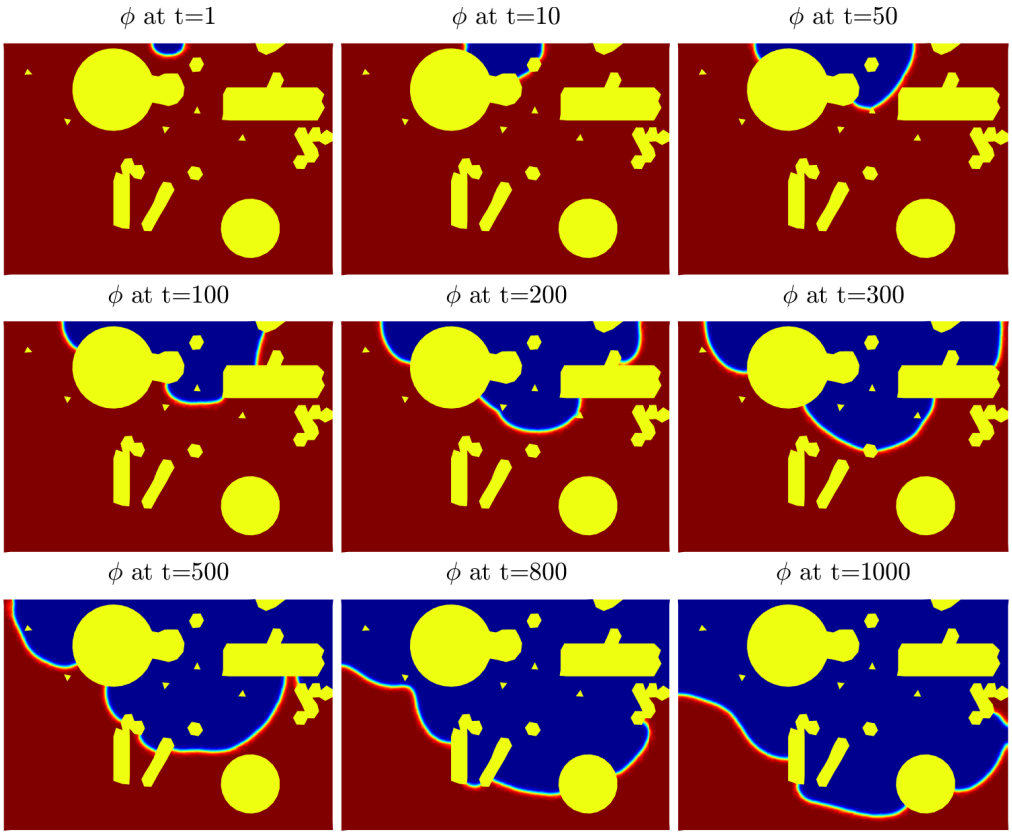

Figure 12: Numerical results for the composite material model. (Example 5.4)

Example 5.5. *We use the proposed scheme to study "pencil electrode test" in three-dimensions. We consider a* 150 $\mu m \times 25$ $\mu m \times 25$ μm *metal wire coated with corrosion-resistant epoxy material on the lateral surface, so that only the top end of the wire is exposed to the solution. Homogeneous Dirichlet boundary condition* $c = \phi = 0$ *are used at the top boundary, and* $c = \phi = 1$ *are used at the bottom boundary, respectively; whereas, homogeneous Neumann boundary condition is prescribed along the lateral surface of the metal wire. At the initial time* $t = 0$ *s, the whole domain is in the pure metal state. This study illustrates the performance of the space-time FEM on 3D meshes, verifies the results against 1D analytical solution and validates the phase field model with experimental data.*

We set $h_{\min} = 1$ μm and $h_{\max} = 15$ μm. The CPU time of running our code is around 8 hours and 20 minutes till the terminal time $T = 225$ s. In Figure 14, we show the field plots of ϕ at $T = 0$, 38, 152, and 225 s. We show the evolution of the pit depth ($\phi = 0.5$) in Figure 15. As expected, the pit depth varies linearly with $\sqrt{t}$. Our numerical results agree with the 2D results in Examples 4.1 and 5.1, the previous works [38, Section 3.1], and the analytic solution given in [49]. It should be noted that, if a uniform mesh with 1 μm in each direction is used, there will be $102,076$ DOFs; whereas, the maximum DOFs used in our method is less than $6,000$, which demonstrates its computational efficiency.

Example 5.6. *We use the proposed space-time adaptive FEM to investigate the evolution of the semi-cylinder pit growth in three dimensions. This example is a three-dimensional analogue of*

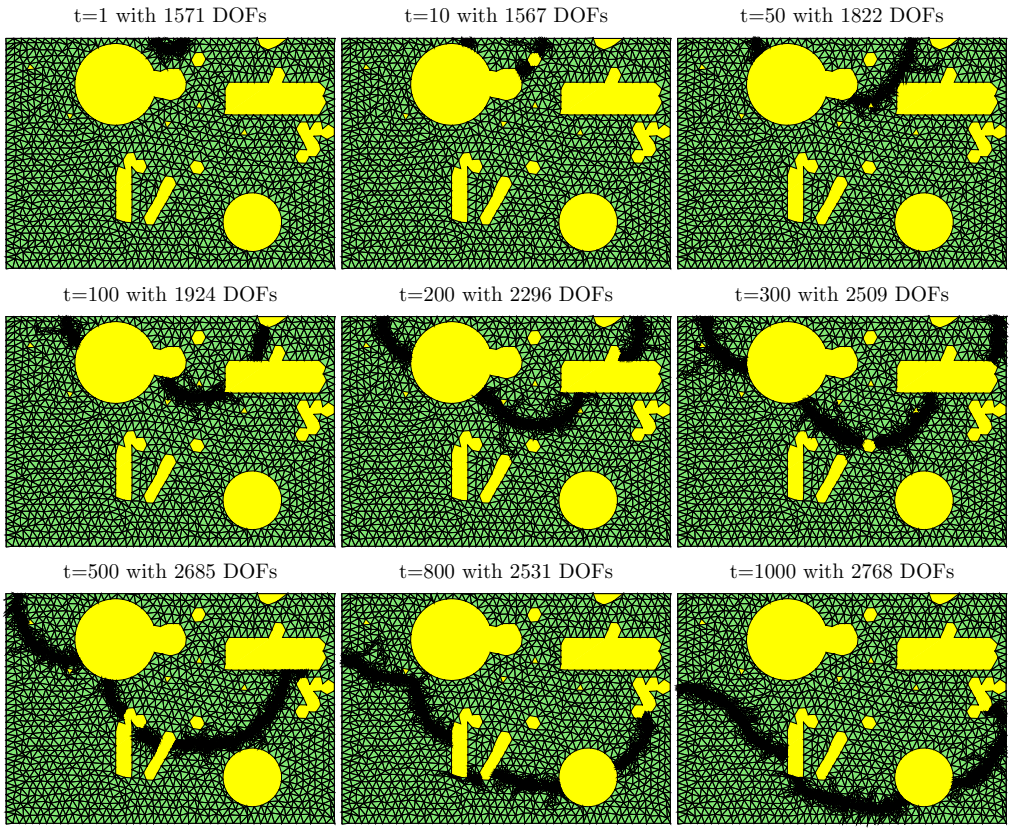

Figure 13: Adaptive meshes at $t = 1$, 10, 50, 100, 250, 350, 500, 750 and 1000 s. (Example 5.4)

Example 5.2. We consider a 400 $\mu m \times$ 50 $\mu m \times$ 200 μm *sample. In the initial state, there is a semi-cylinder opening with diameter* 16 μm *and length* 50 μm *at the center line of the top surface. Homogeneous Dirichlet boundary conditions for both* ϕ *and* c *are applied on this opening, whereas homogeneous Neumann boundary condition is used on the rest of the external boundary.*

It should be noted that, if a uniform mesh with 1 μm in each direction is used, there will be $4,110,651$ DOFs; whereas, the maximum DOFs used in our method is less than $68,400$. To reduce the cost, we use the proposed adaptive meshing algorithm with $h_{\min} = \sqrt{2}$ μm and $h_{\max} = 30$ μm. The CPU time of running our code till the terminal time $T = 225$ is nearly 84 hours. The plots of ϕ and the meshes at times $t = 1$, 20, 100, 200 and 300 s are provided in Figure 16 and the iso-surface plot of $\phi = 0.5$ is presented in the left picture of Figure 17. The pit depth ($\phi = 0.5$) evolutions of 2D and 3D cases are plotted in the right picture in Figure 17, in which the two curves overlap. Because corrosion is uniform in the (50 μm) thickness direction, this 3D problem is equivalent to the 2D problem, and this demonstrates that our 3D numerical implementation is consistent.

Example 5.7. *We use the proposed space-time adaptive FEM to investigate the evolution of the semi-circular pit growth in three dimensions. We consider a* $200\mu m \times 200\mu m \times 100\mu m$ *sample. In the initial state, there is a hemispherical opening with diameter* $16\mu m$ *at the center of the top surface. Homogeneous Dirichlet boundary conditions for both* ϕ *and* c *are applied on this opening, whereas homogeneous Neumann boundary condition is used on the rest of the external boundary. This*

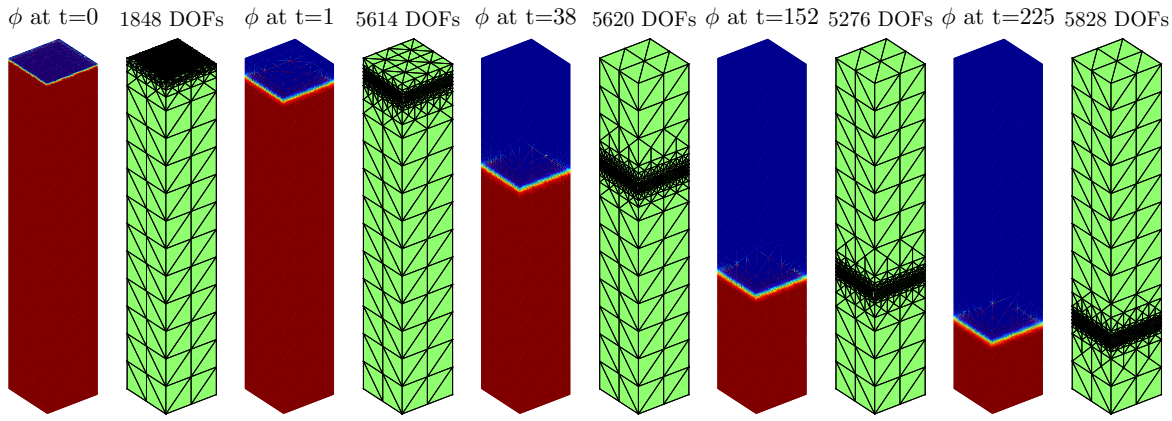

Figure 14: Numerical results of ϕ and meshes obtained by the proposed method. (Example 5.5)

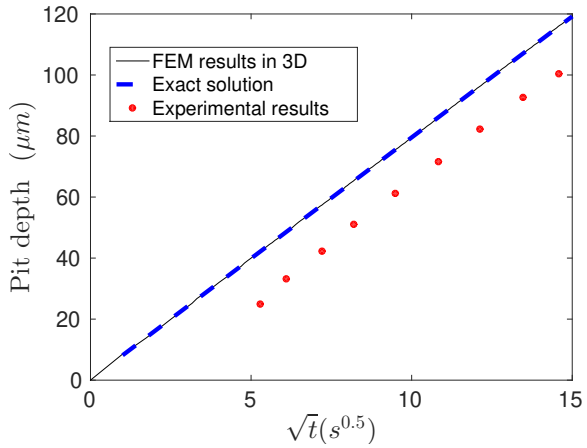

Figure 15: Pit depth evolution obtained by the proposed method. (Example 5.5)

example simulates the experiment conducted in [15] that studies the evolution of an approximately 250 μm^2 pit opening on a metal surface that is elsewhere protected from corrosion using a epoxy coating. Due to complex nature of corrosion and the simplistic assumptions of phase field model, we can only qualitatively validate the results based on experimental results.

It should be noted that, if a uniform mesh with 1 μm in each direction is used, there will be $4,080,501$ DOFs; whereas, the maximum DOFs used in our method is less than $90,300$. To reduce the cost, we use the proposed adaptive meshing algorithm with $h_{\min} = \sqrt{2}\mu$m and $h_{\max} = 30\mu$m. The CPU time for running the code till the terminal time $T = 300$ s is nearly 52 hours. The plots of ϕ and the meshes at times $t = 1$, 20, 100, 200 and 300 s are provided in Figure 18. The iso-surface plot of $\phi = 0.5$ is presented in the left subfigure of Figure 19. It is evident that the pit remains semi-circular as time evolves, which is qualitatively consistent with the experiments of Ernst and Newman [15]. However, due to the difference in specimen geometry and the lack of time-series data of pit depth from experiments in [15], we cannot quantitatively validate the phase field model. We compare the 2D and 3D results in Figure 19 for the pit depth ($\phi = 0.5$) evolution. The pit depth varies (somewhat) linearly with $\sqrt{t}$ in both 2D and 3D experiments, but the rate of pit growth is

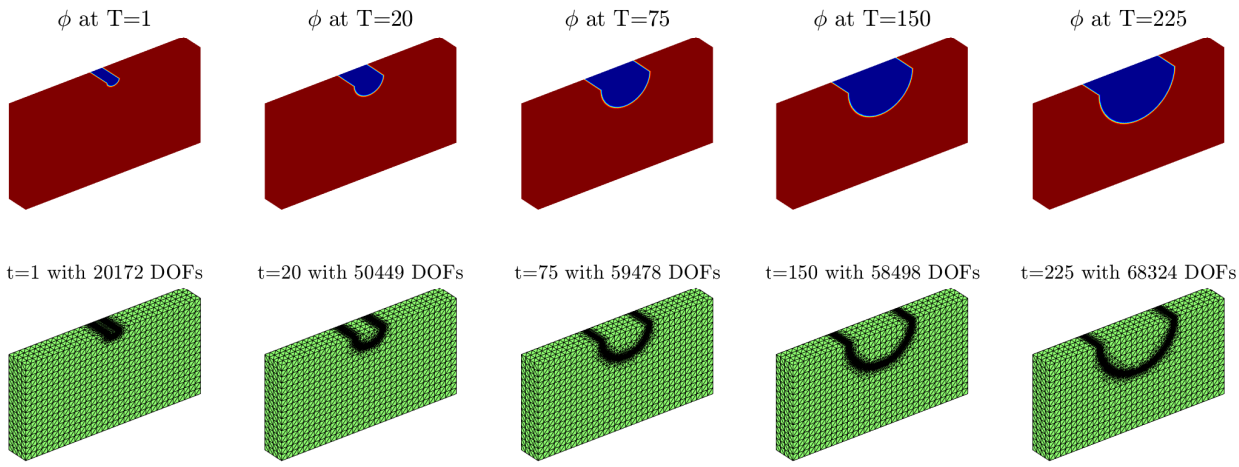

Figure 16: Numerical results of ϕ and the meshes at times $t = 1, 20, 75, 150$ and 225 s. (Example 5.6)

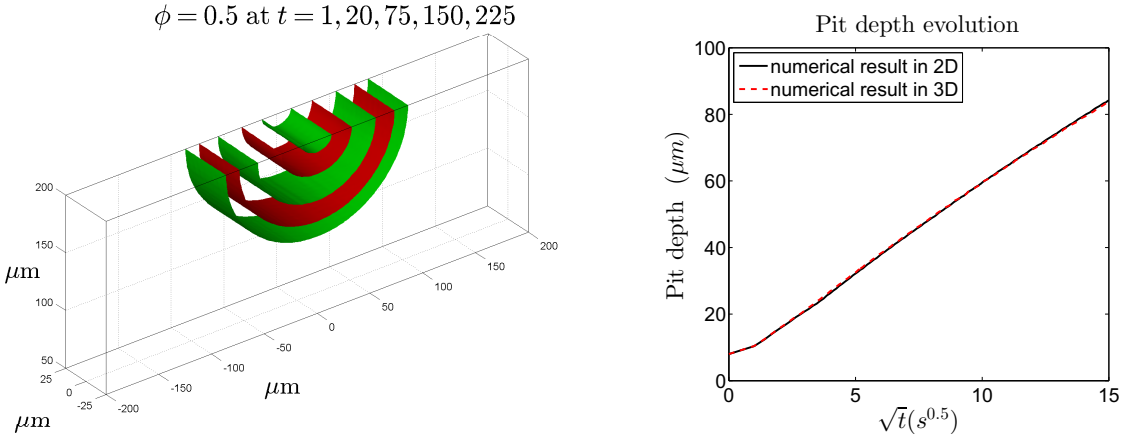

Figure 17: Iso-isoface ($\phi = 0.5$) of the pit evolution (left) and pit depth evolution (right). (Example 5.6)

different. At any given time t, the pit depth in the 3D case is smaller than that in the 2D case. This is expected because the 3D case is not an extrusion of the 2D domain, and the pit opening that exposes the metal to the solution does not span the entire width of the specimen. Because the pit opening is smaller in the 3D case, the solution becomes more saturated in the pit solution domain; consequently the gradients in concentration are smaller leading to slower pit growth.

Example 5.8. *In the final example, we simulate the evolution and coalescence of four small square pits, as shown in Figure 20. The metal sample is a cube with dimensions* 400 μm × 400 μm × 200 μm. *The pit morphology and its evolution in this example is much more complex, so the use of an adaptive mesh is crucial to ensure computational feasibility. It should be noted that a uniform mesh with* 1 μm *in each direction has* $32,321,001$ *DOFs, which is almost impossible to run on a laptop, due to memory restrictions. The adaptive mesh uses less than* $163,000$ *DOFs, which is* 200 *times less than the DOFs the structured uniform mesh. We set* $h_{\min} = \sqrt{2}\mu m$ *and* $h_{\max} = 30\mu m$. *The CPU time for running this code till the terminal time* $T = 150$ *s is nearly* 86 *hours. The phase*

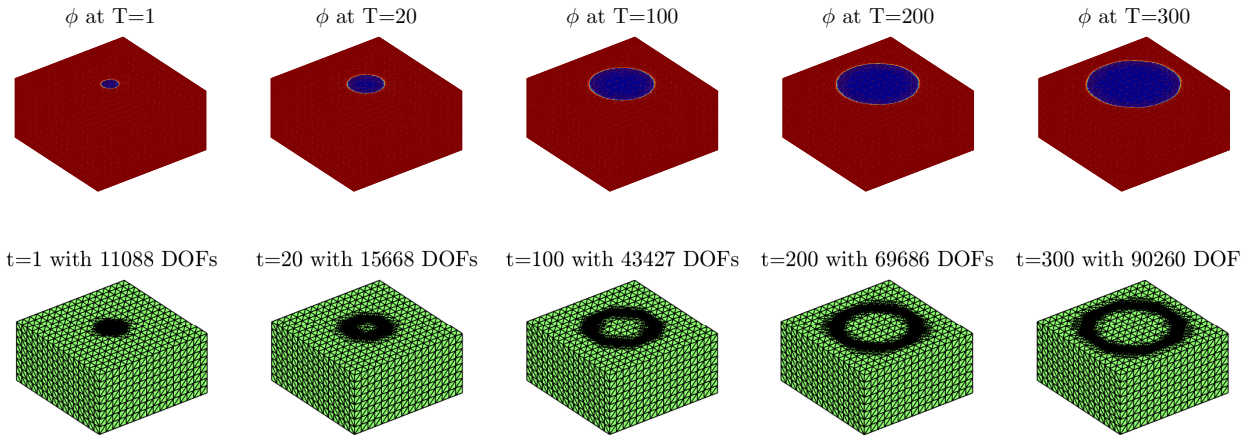

Figure 18: Numerical results of ϕ and the mesh obtained by the proposed method at time $t = 0, 20, 100, 200$ and 300 s. (Example 5.7)

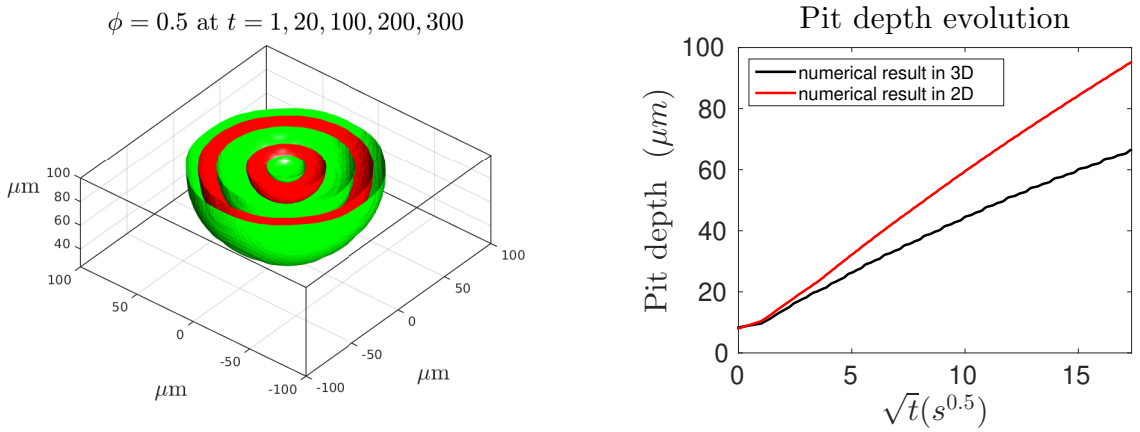

Figure 19: Iso-isoface ($\phi = 0.5$) of the pit evolution (left) and pit depth evolution (right). (Example 5.7)

field results at times $t = 0, 1, 2, 5, 8, 20, 50, 100$ *and* 150 *s are shown in Figure 20. In Figure 21, we show the iso-surface of* $\phi = 0.5$ *at* $t = 1, 5, 8, 20, 150$ *s. We observe that the four pits gradually merge into a big ellipsoid looking pit, as time evolves. This observation is consistent with the experiments in [15] and with the 2D numerical results from a sharp-interface model in [13].*

6. Conclusion

The phase field model of pitting corrosion is computationally tedious and expensive because the stiffness of the reaction-diffusion equation system requires very restricted time step size and the resolution of the steep gradients in phase field and concentration variables near the interface requires very fine mesh resolution. In this paper, we introduce a mass-lumped FEM to solve the pitting corrosion model, where both space and time adaptivity are applied. For the time discretization, we combine the exponential integrator with backward Euler/Crank–Nicolson scheme and design an adaptive time stepping formula to gradually increase the time step size as the diffusion-controlled

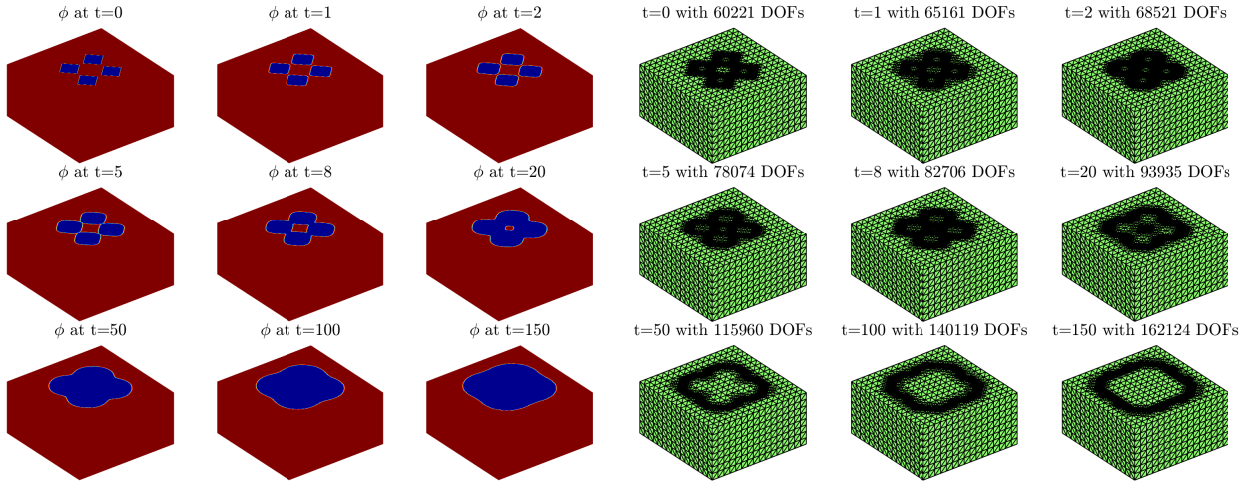

Figure 20: Numerical results of ϕ and meshes obtained by the proposed space-time adaptive method at times $t = 0, 1, 2, 5, 8, 20, 50, 100$ and 150 s. (Example 5.8)

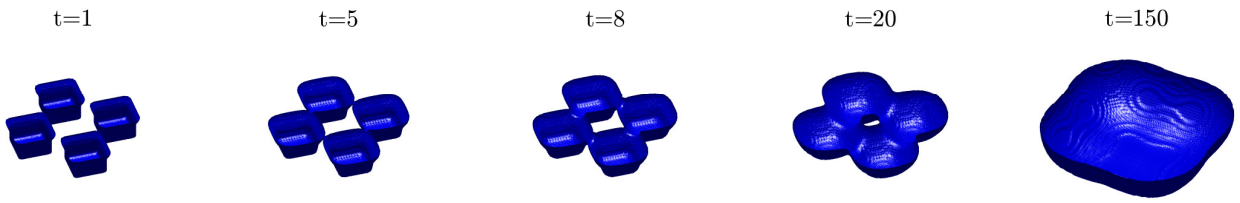

Figure 21: Iso-surface of $\phi = 0.5$ obtained by the proposed method at times $t = 1, 5, 8, 20$ and 150 s. (Example 5.8)

corrosion process slows down. We find that initially a small time step size of 10^{-3} seconds is sufficient to ensure numerical stability for time $t < 1$ second and later on we are able to take larger time steps to accelerate the simulation over longer times. For the spatial approximation, we propose a simple adaptive mesh strategy that automatically refines the mesh in the vicinity of the interface and coarsens it away from the interface based on element variance indicators, thus reducing the number of DOFs significantly. Another major contribution of this paper is that we present an extensive set of benchmark numerical experiments to demonstrate that our method is suitable for practical simulations of the pitting corrosion model in two and three dimensional spaces. The 1D pencil electrode test allows us to verify and calibrate the model parameters of the phase field model; however, due to the lack of time-series data of semi-circular and hemispherical pit growth in the literature we cannot validate the 2D and 3D model results. In fact, the phase field model based on the simple assumption of diffusion-controlled corrosion may not be adequate to estimate the corrosion rate very accurately in more complex 2D and 3D geometries. In our future work, we will apply the proposed space-time adaptive algorithm to more advanced corrosion models, including stress corrosion cracking and galvanic corrosion [4, 39, 40], and calibrate and validate the simulation results against detailed laboratory experiments.

Acknowledgements

H. Gao's research is partially supported by National Natural Science Foundation of China under grant number 11871234 and Hubei Key Laboratory of Engineering Modeling and Scientific Computing. L. Ju's research is partially supported by US National Science Foundation under grant number DMS-1818438 and US Department of Energy under grant numbers DE-SC0016540 and DE-SC0020270. X. Li's research is partially supported by National Natural Science Foundation of China under grant number 11801024.

References

[1] M. Afanasjew, M. Eiermann, O. Ernst and S. Güttel, Implementation of a restarted Krylov subspace method for the evaluation of matrix functions, *Linear Algebra Appl.*, 429(2008), pp. 2293–2314.

[2] E. Allopoulos and Y. Saad, Efficient solution of parabolic equations by Krylov approximation methods, *SIAM J. Sci. Stat. Comput.*, 13(1992), pp. 1236–1264.

[3] A. Al-Mohy and N. Higham, Computing the action of the matrix exponential with an application to exponential integrators, *SIAM J. Sci. Comput.*, 33(2011), pp. 488–511.

[4] T. Ansari, Z. Xiao, S. Hu, Y. Li, J. Luo and S. Shi, Phase-field model of pitting corrosion kinetics in metallic materials, *npj Comput. Mat.*, 4(2018), pp. 38.

[5] C. Beckermann, H. Diepers, I. Steinbach, A. Karma and X. Tong, Modeling melt convection in phase-field simulations of solidification, *J. Comput. Phys.* 154(1999), pp. 468–496.

[6] W. Boettinger, J. Warren, C. Beckermann and A. Karma, Phase-field simulation of solidification, *Annu. Rev. Mater. Res.*, 32(2002), pp. 163–194.

[7] L. Chen, iFEM: An Innovative Finite Element Method Package in MATLAB, *Technical report*, University of California at Irvine, 2008

[8] L. Chen, Phase-field models for microstructure evolution, *Annu. Rev. Mater. Res.* 32(2002), pp. 113–140.

[9] Z. Chen and F. Bobaru, Peridynamic modeling of pitting corrosion damage, *J. Mech. Phys. Solids*, 78(2015), pp. 352–381.

[10] Z. Chen, G. Zhang and F. Bobaru, The influence of passive film damage on pitting corrosion, *J. Electrochem. Soc.*, 163(2016), pp. C19–C24.

[11] D. Danilov and B. Nestler, Phase-field simulations of solidification in binary and ternary systems using a finite element method, *J. Cryst. Growth.* 275(2005), pp. e177–e182.

[12] Q. Du, M. Gunzburger and J. Peterson, Analysis and approximation of the Ginzburg–Landau model of superconductivity, *SIAM Rev.*, 34(1992), pp. 54–81.

[13] R. Duddu, Numerical modeling of corrosion pit propagation using the combined extended finite element and level set method, *Comput. Mech.*, 54(2014), pp. 613–627.

[14] R. Duddu, N. Kota and S. Qidwai, An extended finite element method based approach for modeling crevice and pitting corrosion, *J. Appl. Mech.*, 83(2016), 081003.

[15] P. Ernst and R. Newman, Pit growth studies in stainless steel foils. I. Introduction and pit growth kinetics, *Corros. Sci.*, 44(2002), pp. 927–941.

[16] W. Feng, P. Yu, S. Hu, Z. Liu, Q. Du and L. Chen, Spectral implementation of an adaptive moving mesh method for phase-field equations, *J. Comp. Phys.*, 220(2006), pp. 498–510.

[17] G. Frankel, Pitting corrosion of metals: A review of the critical factors, *J. Electrochem. Soc.*, 145(1998), pp. 2186–2198.

[18] X. Feng and H. Wu, A posteriori error estimates and an adaptive finite element method for the Allen–Cahn equation and the mean curvature flow, *J. Sci. Comput.*, 24(2005), pp. 121–146.

[19] R. Friesner, L. Tuckerman, B. Dornblaser and T. Russo, A method for exponential propagation of large systems of stiff nonlinear differential equations, *J. Sci. Comput.*, 4(1989), pp. 327–354.

[20] A. Frommer and V. Simoncini, Stopping criteria for rational matrix functions of Hermitian and symmetric matrices, *SIAM J. Sci. Comput.*, 30(2008), pp. 1387–1412.

[21] H. Gao, L. Ju, R. Duddu and H. Li, An efficient second-order linear scheme for the phase field model of corrosive dissolution, *J. Comput. Appl. Math.*, 367(2020), pp. 112472.

[22] H. Gao and W. Sun, An efficient fully linearized semi-implicit Galerkin-mixed FEM for the dynamical Ginzburg–Landau equations of superconductivity, *J. Comput. Phys.*, 294(2015), pp. 329–345.

[23] S. Gaudreault and J. Pudykiewicz, An efficient exponential time integration method for the numerical solution of the shallow water equations on the sphere, *J. Comput. Phys.*, 322(2016), pp. 827–848.

[24] H. Gomez, V. Calo, Y. Bazilevs and T. Hughes, Isogeometric analysis of the Cahn–Hilliard phase-field model, *Comput. Methods Appl. Mech. Engg.*, 197(2008), pp. 4333-4352.

[25] N. Higham and A. Al-Mohy, Computing matrix functions, *Acta Numerica*, 19(2010), pp. 159–208.

[26] M. Hochbruck and C. Lubich, On Krylov subspace approximations to the matrix exponential operator, *SIAM J. Numer. Anal.*, 34(1997), pp. 1911–1925.

[27] M. Hochbruck, C. Lubich and H. Selhofer, Exponential integrators for large systems of differential equations, *SIAM J. Sci. Comput.*, 19(1998), pp. 1552–1574.

[28] M. Hochbruck and A. Ostermann, Exponential integrators, *Acta Numerica*, 19(2010), pp. 209–286.

[29] M. Hochbruck, A. Ostermann, and J. Schweitzer, Exponential Rosenbrock-type methods, *SIAM J. Numer. Anal.*, 47(2009), pp. 786–803.

[30] V. Joshi and R. Jaiman, An adaptive variational procedure for the conservative and positivity preserving Allen–Cahn phase-field model, *J. Comput. Phys.*, 366(2018), pp 478–504.

[31] L. Ju and Z. Wang, Exponential time differencing gauge method for incompressible viscous flows, *Commun. Comput. Phys.*, 22(2017), pp. 517–541.

[32] A. Karma and W. Rappel, Phase-field method for computationally efficient modeling of solidification with arbitrary interface kinetics, *Phys. Rev. E*, 53(1996), pp. R3017.

[33] S. Kim, W. Kim and T. Suzuki, Phase-field model for binary alloys, *Physical Review E*, 60(1999), pp. 7186.

[34] D. Landolt, P. Chauvy and O Zinger, Electrochemical micromachining, polishing and surface structuring of metals: fundamental aspects and new developments, *Electrochim. Acta*, 48(2003), pp. 3185–3201.

[35] S. Li, L. Luo, Z. Wang and L. Ju, An exponential time-integrator scheme for steady and unsteady inviscid flows, *J. Comput. Phys.*, 365(2018), pp. 206–225.

[36] V. Luan and D. Michels, Explicit exponential Rosenbrock methods and their application in visual computing, *arXiv:1805.08337*.

[37] D.D. Macdonald and G.R. Engelhardt, 2.39 Predictive Modeling of Corrosion, in: *T.J.A. Richardson, B.R.A. Cottis, R. Lindsay, S. Lyon, D.J.D. Scantlebury, H. Stott, et al. (Eds.), Shreirs Corrosion, 4th ed.*, Elsevier B.V., 2010, pp. 1630–1679.

[38] W. Mai, S. Soghrati and R. Buchheit, A phase field model for simulating the pitting corrosion, *Corros. Sci.*, 110(2016), pp. 157–166.

[39] W. Mai and S. Soghrati, A phase field model for simulating the stress corrosion cracking initiated from pits, *Corros. Sci.*, 125(2017), pp. 87–98.

[40] W. Mai and S. Soghrati, New phase field model for simulating galvanic and pitting corrosion processes, *Electrochim. Acta*, 260(2018), pp. 290–304.

[41] N. Moelans, A quantitative and thermodynamically consistent phase-field interpolation function for multi-phase systems, *Acta Mater.*, 59(2011), pp. 1077–1086.

[42] N. Moelans, B. Blanpain and P. Wollants, An introduction to phase-field modeling of microstructure evolution, *Comput. Coupling Phase Diagrams Thermochem.*, 32(2008). pp. 268–294.

[43] A. Nauts and R. Wyatt, New approach to many-state quantum dynamics: The recursive-residue-generation method, *Phys. Rev. Lett.*, 51(1983), pp. 2238–2241.

[44] T. Park and J. Light, Unitary quantum time evolution by iterative Lanczos reduction, *J. Chem. Phys.*, 85(1986), pp. 5870–5876.

[45] R.M. Pidaparti, L. Fang and M.J. Palakal, Computational simulation of multi-pit corrosion process in materials, *Comp. Mat. Sci.*, 41(2008), pp. 255–265.

[46] M. Popolizio and V. Simoncini, Acceleration techniques for approximating the matrix exponential operator, *SIAM J. Matrix Anal. Appl.*, 30(2008), pp. 657–683.

[47] N. Provatas, N. Goldenfeld and J. Dantzig, Adaptive Mesh Refinement Computation of Solidification Microstructures Using Dynamic Data Structures, *J. Comput. Phys.*, 148(1999), pp. 265–290.

[48] S. Sarkar, J. Warner and W. Aquino, A numerical framework for the modeling of corrosive dissolution, *Corros. Sci.*, 65(2012), pp. 502–511.

[49] S. Scheiner and C. Hellmich, Stable pitting corrosion of stainless steel as diffusion-controlled dissolution process with a sharp moving electrode boundary, *Corros. Sci.*, 49(2007) pp. 319–346.

[50] S. Scheiner and C. Hellmich, Finite volume model for diffusion- and activation-controlled pitting corrosion of stainless steel, *Comput. Methods Appl. Mech. Engg.*, 198(2009), pp. 2898–2910.

[51] A. Sedriks, *Corrosion of Stainless Steel, 2nd ed.*, John Wiley and Sons, Inc, New York, NY, United States, 1996.

[52] R. Sidje, Expokit: A software package for computing matrix exponentials, *ACM Trans. Math. Software*, 24(1998), pp. 130–156.

[53] I. Steinbach, F. Pezzolla and B. Nestler, M. Seesselberg, R. Prieler, et al., A phase field concept for multiphase systems, *Physica D*, (94)1996, pp.135–147.

[54] A. Tambue, I. Berre and J. Nordbotten, Efficient simulation of geothermal processes in heterogeneous porous media based on the exponential Rosenbrock–Euler and Rosenbrock-type methods, *Adv. Water Resour.*, 53(2013), pp. 250–262.

[55] Z. Tan, T. Tang and Z. Zhang, A simple moving mesh method for one- and two-dimensional phase-field equations, *J. Comp. Appl. Math.*, 190(2006), pp. 252–269.

[56] M. Tokman, Efficient integration of large stiff systems of ODEs with exponential propagation iterative (EPI) methods, *J. Comput. Phys.*, 213(2006), pp. 748–776.

[57] A. Vagbharathi and S. Gopalakrishnan, An extended finite-element model coupled with level set method for analysis of growth of corrosion pits in metallic structures, *Proc. R. Soc. A*, 470(2014), 20140001.

[58] H. Vo and R. Sidje, Approximating the large sparse matrix exponential using incomplete orthogonalization and Krylov subspaces of variable dimension, *Numer. Linear Algebra Appl.*, 24(2017), e2090.

[59] S. Wang and R. Sekerka, Algorithms for Phase Field Computation of the Dendritic Operating State at Large Supercoolings, *J. Comput. Phys.*, 127(1996), pp. 110–117.

[60] J. Warren and W. Boettinger, Prediction of dendritic growth and microsegregation patterns in a binary alloy using the phase-field method, *Acta Metall. Mater.*, 43(1995), pp. 689–703.

[61] A. Wheeler, B. Murray and R. Schaefer, Computation of dendrites using a phase field model, *Physica D*, 66(1993), pp. 243–262.

[62] J. Zhang and Q. Du, Numerical Studies of Discrete Approximations to the Allen–Cahn Equation in the Sharp Interface Limit, *SIAM J. Sci. Comput.*, 31(2009), pp. 3042–3063.