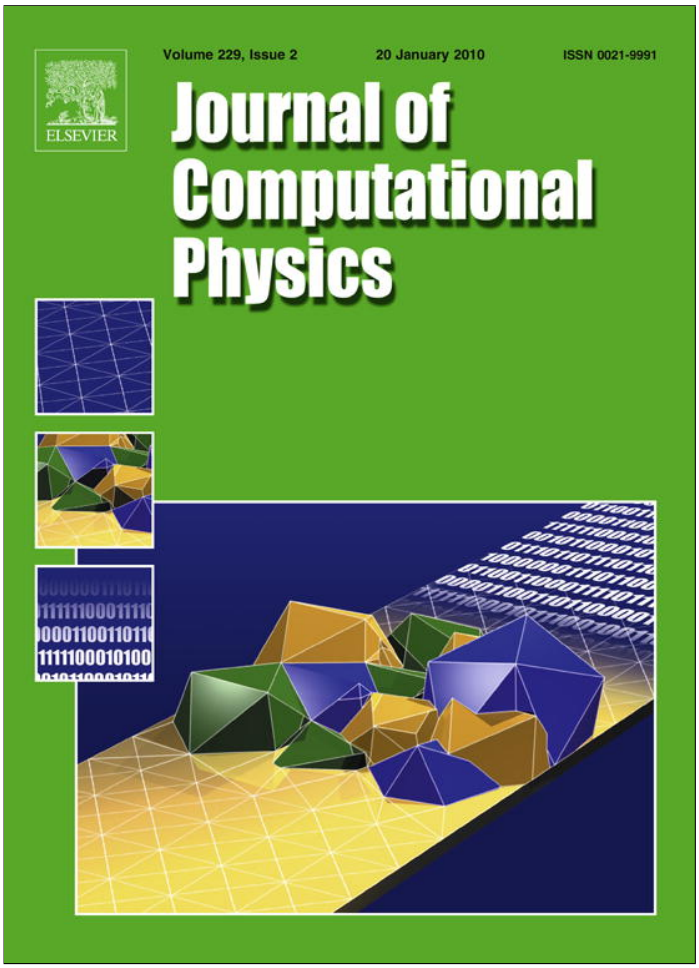

Journal of Computational Physics 229 (2010) 379–398

Contents lists available at ScienceDirect

Journal of Computational Physics

journal homepage: www.elsevier.com/locate/jcp

An algorithm for simulation of electrochemical systems with surface–bulk coupling strategies

Matthew Buoni [a,*], Linda Petzold [a,b]

[a] Mechanical Engineering Department, University of California, Santa Barbara, CA 93106, United States
[b] Computer Science Department, University of California, Santa Barbara, CA 93106, United States

ARTICLE INFO

Article history:
Received 8 January 2009
Received in revised form 30 August 2009
Accepted 22 September 2009
Available online 12 October 2009

PACS:
82.47.a

Keywords:
Electrochemical systems
Numerical simulation
Electrodeposition
Coupling method

ABSTRACT

In Buoni and Petzold (2007) [13] we described a new algorithm for simulation of electrochemical systems on two-dimensional irregular, time-dependent domains. Here we show how to extend the algorithm to three dimensions. We demonstrate our three-dimensional algorithm by simulating copper electrodeposition into a via structure. This problem poses challenges for the coupling of the dilute electrolyte (bulk) model to the surface dynamics model, which involves a complex network of reactions. To handle this coupling, we introduce a new and highly effective semi-implicit method.

Published by Elsevier Inc.

1. Introduction

Electrochemical processes are widely used throughout industry. Applications include batteries, fuel cells, photovoltaics, application of coatings onto metals and fabrication of interconnects in computer processors. Generally such processes involve electrolytic solutions containing various ions and additives interacting with conducting surfaces on which chemical reactions and various physical processes occur. These surface processes either produce a voltage difference or are induced by an applied voltage difference between the surface and electrolyte solution. Although the fundamental equations describing the bulk electrolyte solution are well known [5], the surface chemistry and dynamics are often less clear and subject to modeling trial and error. We have developed a highly efficient algorithm that solves the governing equations for the bulk while allowing the user to include his or her particular surface model with relative ease.

In [13] we described a new algorithm for simulation of electrochemical systems on irregular, time-dependent domains in two dimensions. Our method uses finite volume discretization of an arbitrarily shaped spatial domain. The resulting differential algebraic equation system is solved with a time splitting method that involves a projection step used to satisfy the algebraic constraint (electroneutrality). Here, we extend the method to three dimensions, and apply it to the challenging problem of simulating electrodeposition of a copper via structure. This requires treatment of the moving boundary and coupling of the dilute electrolyte (bulk) model to the surface model. The boundary is moved using a highly accurate version of

* Corresponding author.
E-mail address: mjbuoni@gmail.com (M. Buoni).

0021-9991/$ - see front matter Published by Elsevier Inc.
doi:10.1016/j.jcp.2009.09.032

the level set method, which uses points seeded along the interface to track the surface properties and to reconstruct the interface as required. Both the moving boundary method and the surface–bulk coupling strategy will be addressed here.

It is important to emphasize that to date very few three-dimensional simulations of realistic models for copper electrodeposition have been performed. The model we use here was developed by researchers at IBM in collaboration with our collaborator R. Alkire and his research group at the University of Illinois, Urbana-Champaign in order to explain their observations on trench infill experiments. Solution additives *PEG*, Cl^-, *SPS* and *HIT* are included and allowed to participate in fourteen different surface reactions as well as three bulk reactions. We will summarize this model in Section 2.3. For further details of this model and the determination of the physical parameters, we refer the interested reader to [20–23]. Our work is distinguished from that of other researchers who have worked on copper electrodeposition [24–28] by the complexity of our model and the generality of our algorithm. Specifically, our algorithm is designed to handle arbitrarily complex reaction networks in principle by simply modifying an input file. We include the effects of suppressor, accelerator and leveler additives through the reaction pathway proposed in [20]. From our experience, more complex reaction networks generally have wider ranges of timescales and thus are susceptible to surface–bulk coupling stiffness. This results in severe time step restrictions for the standard coupling methods used by other researchers. We address this issue with our semi-implicit coupling algorithm. Additionally, we include electroneutrality of the solution in our formulation, which is usually neglected by others due to the computational difficulty it presents. As a result, our simulations are able to capture the often subtle but sometimes significant role that electric migration can play in the local rates of copper deposition.

Three-dimensional simulations are essential for studying problems such as copper nucleation and growth dynamics on resistive metal substrates, and for studying the process of copper infill of via interconnect structures in computer processors. Fig. 1 illustrates such a system. In this figure we see that the via consists of a trench shaped region 20–200 nm in width and

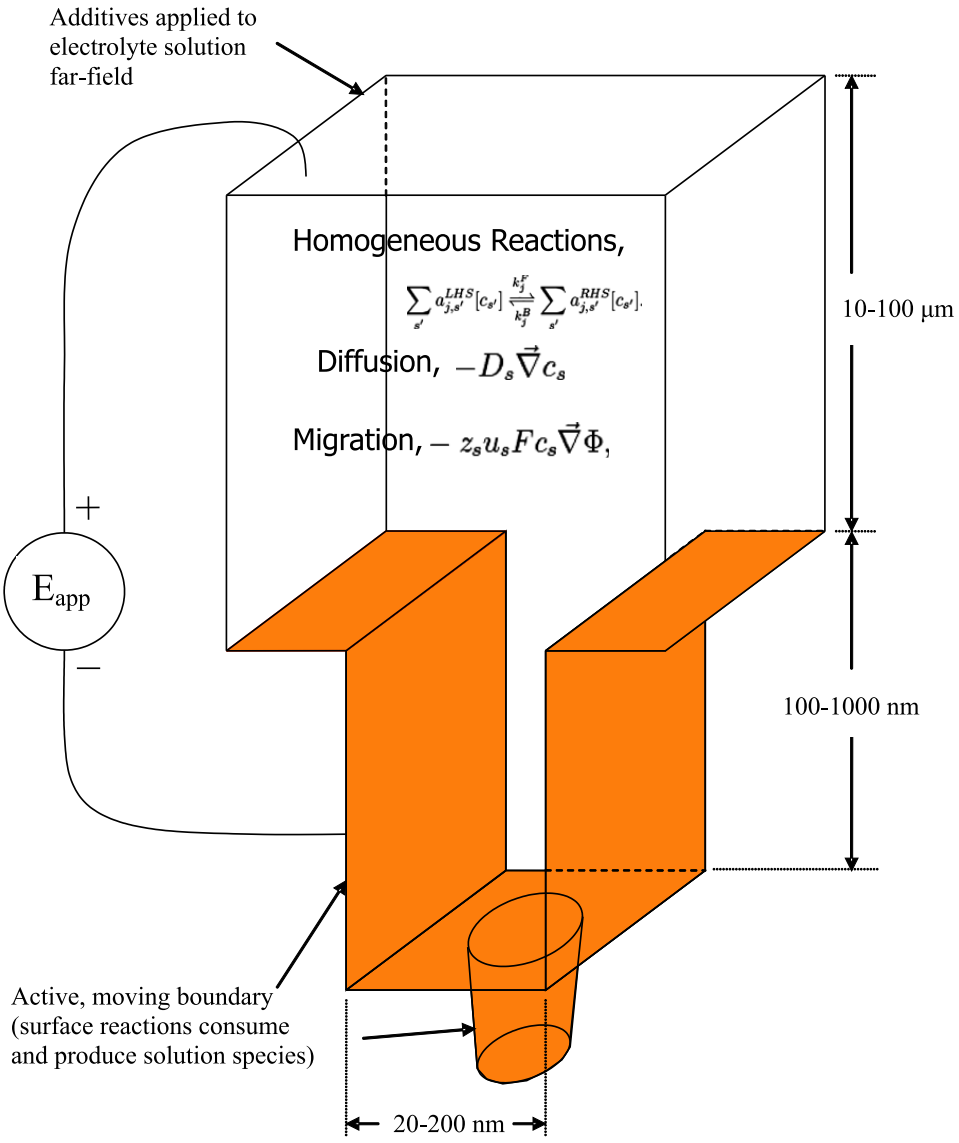

Fig. 1. Illustration shows the shape and dimensions of the via physical domain. The governing physics are also indicated for the different regions.

around 5:1 to 15:1 aspect ratio. At the bottom of the trench a tapered cylinder is bored out (used to connect to a lower layer of interconnects on a computer processor). The region above the trench consists of dilute electrolyte solution (bulk) and forms a mass flux boundary layer. At the top of the layer (10–100 μm), the solution is well mixed. It has a fixed far-field composition represented by a Dirichlet boundary condition.

This paper is organized as follows. In Section 2 we describe the governing equations, including the complex surface–bulk reaction network. In Section 3 we discuss the numerical method, with particular attention given to the moving boundary and surface–bulk model coupling. In Section 4 we perform a series of numerical experiments that test the algorithm's accuracy and efficiency. Finally, in Section 5 we illustrate the method by simulating the fully three-dimensional copper infill of a via.

2. Governing equations

2.1. Bulk electrolyte

The governing equations are stiff nonlinear partial differential equations with algebraic constraints [5]. These equations describe the time evolution of the concentrations of each chemical species, c_s. They are derived by conservation of mass with chemical reactions, diffusion and migration due to electric fields,

$$\frac{\partial c_s}{\partial t} = R_s(\{c_{s'}\}) - \vec{\nabla}\cdot\vec{N}_s, \tag{2.1a}$$

$$\sum_s z_s c_s = 0, \tag{2.1b}$$

where R_s is the net rate of production of chemical species s due to chemical reactions, and is a function of all the other chemical species concentrations, $\{c_{s'}\}$. The flux, $\vec{N}_s$, of chemical species s due to diffusion and migration is given by

$$\vec{N}_s = -D_s\vec{\nabla}c_s - z_s u_s F c_s \vec{\nabla}\Phi, \tag{2.2}$$

where Φ is the electric potential, D_s is the diffusion coefficient for species s, z_s is the charge of species s, $u_s = \frac{D_s}{RT}$ is the mobility constant for species s, and F is Faraday's constant. The algebraic constraint (Eq. (2.1b)) enforces zero net charge density for the electrolyte solution.

Convective transport may also be included by adding $c_s\vec{v}$ (where $\vec{v}$ is the fluid velocity) to the flux and coupling these equations to the Navier–Stokes equations [4], but this is often neglected for systems with dimensions below 1 μm. For these systems, which will be our focus here, diffusion dominates because the Peclet number is small.

At the upper boundary there is a far-field set of fixed values for each of the chemical species concentrations and the electric potential, represented by Dirichlet boundary conditions,

$$c_s = c_s^{\infty}, \tag{2.3}$$

$$\Phi = \Phi^{\infty}. \tag{2.4}$$

On the sides of the physical domain one can assume either symmetry or periodic boundary conditions, depending on the shape of the active copper boundary. Here we assume symmetry boundary conditions, which implies that the gradient of both the concentration fields and electric potential are zero in the normal direction,

$$\vec{\nabla}c_s\cdot\hat{n} = 0, \tag{2.5}$$

$$\vec{\nabla}\Phi\cdot\hat{n} = 0. \tag{2.6}$$

Although the specific form of the boundary conditions at the conducting (i.e. "active") surface are application dependent, in many systems they can be represented as a flux, J_s, for each bulk species, s. These fluxes usually depend on many variables, including the bulk species concentrations at the surface. For example, in copper electrodeposition the surface is modeled as copper metal covered in a monolayer of adsorbed molecules where reactions occur. Using the reaction rates, we can compute a net production rate of each chemical species, which results in a flux into the electrolyte solution. This boundary condition can be expressed as

$$-\vec{N}_s\cdot\hat{n} = (D_s\vec{\nabla}c_s + z_s u_s F c_s\vec{\nabla}\Phi)\cdot\hat{n} = J_s, \tag{2.7}$$

where $\hat{n}$ is the outward normal direction along the active boundary. The flux, J_s, is calculated from a separate surface reaction model using either deterministic or stochastic dynamics. Generally speaking, J_s is the production rate of species s due to surface reactions and is a function of the bulk chemical species concentrations at the surface, $\{c_{s'}\}|_{surface}$, as well as coverage densities of the adsorbed surface species, $\{\theta_{ss'}\}$, which are tracked by the surface reaction model.

Because we solve for both the concentration fields and electric potential, we need one additional boundary condition for our system. This condition is contained in the electroneutrality constraint (Eq. (2.1b)). By first computing the gradient and then taking the dot product in the normal direction, we find

$$\sum_s z_s\vec{\nabla}c_s\cdot\hat{n} = 0. \tag{2.8}$$

We can explicitly solve Eqs. (2.7) and (2.8) for $\frac{\partial c_s}{\partial n} = \vec{\nabla} c_s \cdot \hat{n}$ and $\frac{\partial \Phi}{\partial n} = \vec{\nabla} \Phi \cdot \hat{n}$, which yields

$$D_s \frac{\partial c_s}{\partial n} = J_s - z_s u_s c_s \left(\sum_{s'} \frac{z_{s'} J_{s'}}{D_{s'}} \right) \Bigg/ \left(\sum_{s'} \frac{z_{s'}^2 u_{s'} c_{s'}}{D_{s'}} \right), \tag{2.9}$$

$$\frac{\partial \Phi}{\partial n} = \frac{1}{F} \left(\sum_{s'} \frac{z_{s'} J_{s'}}{D_{s'}} \right) \Bigg/ \left(\sum_{s'} \frac{z_{s'}^2 u_{s'} c_{s'}}{D_{s'}} \right). \tag{2.10}$$

The active boundary moves due to the deposition of copper resulting from surface reactions, with the local speed given by

$$v_n = \frac{J_{Cu}}{\rho_{Cu}}, \tag{2.11}$$

where ρ_{Cu} is the density of solid copper.

2.2. Active surface

The active surface forms the lower boundary for the bulk electrolyte region and is the surface on which copper is deposited by a complex sequence of surface reactions. These surface reactions drive the entire system by consuming and producing chemical species in the surrounding electrolyte solution. In addition, the surface reactions create new adsorbed surface species which in turn react to form additional solution species. The rate for any surface reaction is calculated as the product of concentrations of the reactants involved (both in solution and absorbed). Additionally, a multiplicative Boltzmann factor is included (like in the ButlerVolmer equation) for those reactions involving consumption or production of electrons. These reaction rates are used to calculate a flux for each chemical species in solution as well as the time rate of change of the surface coverages for each adsorbed surface species. Also included is the rate of solid copper production which determines the speed at which the active boundary moves by Eq. (2.11).

Our computational framework is quite general and can accommodate a variety of different types of surface models, including stochastic Kinetic Monte-Carlo models, deterministic mean field models, and hybrid approaches that treat important events (such as copper nucleation) stochastically but everything else deterministically.In the example that follows, we use an ordinary differential equation model, for which we have a set of adsorbed surface species, *ss*, each having a constant molar surface density Π_{ss} and fractional surface coverage θ_{ss}, satisfying

$$\sum_{ss} \theta_{ss} = 1. \tag{2.12}$$

Mass conservation on the moving, reacting surface is represented by the reaction–advection equation,

$$\frac{\partial(\theta_{ss} \Pi_{ss})}{\partial t} + \vec{\nabla} \cdot (\vec{v} \theta_{ss} \Pi_{ss}) = R_{ss}^{surf}, \tag{2.13}$$

where R_{ss}^{surf} is the net production rate of surface species *ss* due to surface reactions, and $\vec{v} = v_n \hat{n}$ is the interface velocity.

Eq. (2.13) may be expressed in terms of interface curvature κ as,

$$\frac{\partial \theta_{ss}}{\partial t} = \frac{R_{ss}^{surf}}{\Pi_{ss}} - v_n \kappa \theta_{ss}, \tag{2.14}$$

since spatial gradients in the surface coverage, $\vec{\nabla} \theta_{ss}$, occur along the interface perpendicular to $\vec{v}$.

2.3. Model specification

Copper electrodeposition is carried out experimentally by our collaborator R. Alkire and his research group at the University of Illinois, Urbana-Champaign using an aqueous solution consisting of the following ingredients: sulfuric acid (H_2SO_4), copper sulfate salt ($CuSO_4$), polyethylene glycol (*PEG*), Chloride (Cl^- in the form of *HCl*), bis(3-sulfopropyl)disulfide (*SPS*) and (1-(2-hydroxyethyl)-2-imidazolidinethione) (*HIT*).

Through the proposed surface/bulk reaction network, the bulk electrolyte solution (*aq*) consists of twelve ions and molecules, with their properties given in Table 1. On the active surface, there are exposed copper atoms in addition to four adsorbed (*ads*) molecules, given with their properties in Table 2. The reactions in the bulk electrolyte solution are given in Table 3, while the surface reactions and rates are given in Tables 4 and 5, respectively. The rate constants used are those determined in [22]. Expressions for bulk species fluxes and surface species production rates are calculated from the surface reaction rates and are provided in Tables 6 and 7, respectively. In the reaction tables, the associated mechanism is indicated by Mech., with the additive-free mechanism denoted by (A-F), suppressor (S), accelerator (A) and leveler (L).

During trench/via infill experiments, it is desirable to control the current rather than the applied potential directly. This is done by manipulating E_{app} to give the target current. In our simulations, we do this by solving the following equation for E_{app} at every time step,

Table 1
Bulk species properties.

s	Species	Charge, z_s	Diff. coef., D_s ($\times 10^9$ m^2/s)	Far field conc. (mol/m^3)
1	Cu_{aq}^{2+}	2	0.714	300.0
2	Cu_{aq}^{+}	1	1.0	0.0
3	Cl_{aq}^{-}	−1	2.032	1.4
4	PEG_{aq}	0	0.05	0.0882
5	SPS_{aq}	0	1.0	0.0282
6	MPS_{aq}	0	1.0	0.0
7	$thiolate_{aq}^{-}$	−1	1.0	0.0
8	HIT_{aq}	0	1.0	0.0
9	H_{aq}^{+}	1	9.311	0.0
10	$(HSO_4^-)_{aq}$	−1	1.385	0.0
11	$(SO_4^{2-})_{aq}$	−2	1.065	0.0
12	$(H_2SO_4)_{aq}$	0	1.0	1250.0

Table 2
Surface species properties.

ss	Species	Surface density, Γ_{ss} ($\times 10^5$ mol/m^2)	Initial coverage
1	Cu_s	2.53	1.0
2	$Cu(I)thiolate_{ads}$	2.53	0.0
3	$CuClPEG_{ads}$	0.0633	0.0
4	$CuCl_{ads}$	2.53	0.0
5	$Cu(I)HIT_{ads}$	2.53	0.0

Table 3
Homogeneous bulk reactions.

j	Bulk reaction	For. rate, k^F (s^{-1})	Back. rate, k^B (m^3(mol s)$^{-1}$)	Mech.
1	$MPS_{aq} \rightleftharpoons H_{aq}^{+} + thiolate_{aq}^{-}$	10	10^5	A
2	$(H_2SO_4)_{aq} \rightleftharpoons (HSO_4^-)_{aq} + H_{aq}^{+}$	10^6	10	A-F
3	$(HSO_4^-)_{aq} \rightleftharpoons (SO_4^{2-})_{aq} + H_{aq}^{+}$	10^5	10^4	A-F

Table 4
Surface reactions.

j	Surface reaction	Mech.
1	$Cu_{aq}^{2+} + e^- \rightarrow Cu^{+}$	A-F
2	$Cu_{aq}^{+} + e^- \rightarrow Cu_s$	A-F
3	$Cu_s \rightarrow Cu_{aq}^{+} + e^-$	A-F
4	$Cu_{aq}^{+} + Cl_{aq}^{-} \rightarrow CuCl_{ads}$	S
5	$CuCl_{ads} \rightarrow Cu_{aq}^{+} + Cl_{aq}^{-}$	S
6	$CuCl_{ads} + e^- \rightarrow Cu_s + Cl_{aq}^{-}$	S
7	$CuCl_{ads} + PEG_{aq} \rightarrow CuClPEG_{ads}$	S
8	$CuClPEG_{ads} \rightarrow CuCl_{ads} + PEG_{aq}$	S
9	$SPS_{aq} + 2e^- \rightarrow 2thiolate_{aq}^{-}$	A
10	$MPS_{aq} + Cu_{aq}^{+} \rightarrow Cu(I)thiolate_{ads} + H_{aq}^{+}$	A
11	$Cu(I)thiolate_{ads} + H_{aq}^{+} \rightarrow MPS_{aq} + Cu_{aq}^{+}$	A
12	$Cu_{aq}^{+} + Cu(I)thiolate_{ads} + e^- \rightarrow Cu(I)thiolate_{ads} + Cu_s$	A
13	$Cu(I)thiolate_{ads} + HIT_{aq} \rightarrow Cu(I)HIT_{ads} + MPS_{aq}$	L
14	$Cu(I)HIT_{ads} + H_{aq}^{+} + e^- \rightarrow HIT_{aq} + Cu_s$	L

Table 5
Surface reaction rates.

j	Reaction rate, r_j^{surf}
1	$k_1^{surf} C_1(\theta_1 \Pi_1) \exp(-\alpha_1 F(E_{app} - E_1^0)/RT)$
2	$k_2^{surf} C_2 \theta_1 \Pi_1 \exp(-\alpha_2 F(E_{app} - E_2^0)/RT)$
3	$k_3^{surf} \theta_1 \Pi_1 \exp((1-\alpha_3) F(E_{app} - E_3^0)/RT)$
4	$k_4^{surf} C_2 C_3 \theta_1 \Pi_1$
5	$k_5^{surf} \theta_4 \Pi_4$
6	$k_6^{surf} \theta_4 \Pi_4 \exp(-\alpha_6 F(E_{app} - E_6^0)/RT)$
7	$k_7^{surf} C_4 \theta_4 \Pi_4$
8	$k_8^{surf} \theta_3 \Pi_3$
9	$k_9^{surf} C_5(\theta_1 \Pi_1) \exp(-2\alpha_9 F(E_{app} - E_9^0)/RT)$
10	$k_{10}^{surf} C_2 C_6 \theta_1 \Pi_1$
11	$k_{11}^{surf} C_9 \theta_2 \Pi_2$
12	$k_{12}^{surf} C_2 \theta_2 \Pi_2 \exp(-\alpha_{12} F(E_{app} - E_{12}^0)/RT)$
13	$k_{13}^{surf} C_8 \theta_2 \Pi_2$
14	$k_{14}^{surf} C_9 \theta_5 \Pi_5 \exp(-\alpha_{14} F(E_{app} - E_{14}^0)/RT)$

Table 6
Bulk species fluxes, J_s.

s	Species	Flux, J_s
1	Cu_{aq}^{2+}	$-r_1^{surf}$
2	Cu_{aq}^{+}	$r_1^{surf} - r_2^{surf} + r_3^{surf} - r_4^{surf} + r_5^{surf} - r_{10}^{surf} + r_{11}^{surf} - r_{12}^{surf}$
3	Cl_{aq}^{-}	$-r_4^{surf} + r_5^{surf} + r_6^{surf}$
4	PEG_{aq}	$-r_7^{surf} + r_8^{surf}$
5	SPS_{aq}	$-r_9^{surf}$
6	MPS_{aq}	$-r_{10}^{surf} + r_{11}^{surf} + r_{13}^{surf}$
7	$thiolate_{aq}^{-}$	$2r_9^{surf}$
8	HIT_{aq}	$-r_{13}^{surf} + r_{14}^{surf}$
9	H_{aq}^{+}	$r_{10}^{surf} - r_{11}^{surf} - r_{14}^{surf}$
10	$(HSO_4^{-})_{aq}$	0
11	$(SO_4^{2-})_{aq}$	0
12	$(H_2SO_4)_{aq}$	0

Table 7
Surface species rates of production, R_{ss}^{surf}.

ss	Species	Rate of production, R_{ss}^{surf}
1	Cu_s	Determined by Eq. (2.12)
2	$Cu(I)thiolate_{ads}$	$r_{10}^{surf} - r_{11}^{surf} - r_{13}^{surf}$
3	$CuClPEG_{ads}$	$r_7^{surf} - r_8^{surf}$
4	$CuCl_{ads}$	$r_4^{surf} - r_5^{surf} - r_6^{surf} - r_7^{surf} + r_8^{surf}$
5	$Cu(I)HIT_{ads}$	$r_{13}^{surf} - r_{14}^{surf}$
6	Solid copper	$J_{Cu} = r_2^{surf} - r_3^{surf} + r_6^{surf} + r_{12}^{surf} + r_{14}^{surf}$

$$I = F \int_S J_{e^-} dA,$$
$$J_{e^-} = \sum_j (n_j^{RHS} - n_j^{LHS}) r_j^{surf} = r_1^{surf} + r_2^{surf} - r_3^{surf} + r_6^{surf} + 2r_9^{surf} + r_{12}^{surf} + r_{14}^{surf}, \tag{2.15}$$

where I is the total current, J_{e^-} is the flux of electrons, F is Faraday's constant (the charge per mole of electrons), and the integration surface, S, is the entire active boundary. In order to solve Eq. (2.15) for E_{app}, the concentrations and coverages are taken at the previous time step.

3. Numerical solution

3.1. Bulk electrolyte

Temporal discretization is accomplished via a splitting technique that uses the Backward (implicit) Euler method combined with a projection step [13]. First, we perform a finite volume discretization over the spatial domain. For this, we consider the most complex case: a cell that is cut in an arbitrary way by the active (moving) boundary. After applying the divergence theorem and approximating boundary integrals by products of average values (at boundary midpoints) multiplied by boundary length and area integrals by average values (at cell centroids) multiplied by area, we obtain an equation of the form:

$$V^{rel}_{i,j,k}\frac{\partial\overline{(c_s)}_{i,j,k}}{\partial t} = (RHS)^{(rxns)} + (RHS)^{(diff)} + (RHS)^{(mig)} + (RHS)^{(flux,c)} + (RHS)^{(flux,\Phi)}, \tag{3.1}$$

where $\overline{(c_s)}_{i,j,k}$ is the concentration of chemical species s at the centroid of cell (i,j,k).

$(RHS)^{(rxns)}$, $(RHS)^{(diff)}$, $(RHS)^{(mig)}$, $(RHS)^{(flux,c)}$ and $(RHS)^{(flux,\Phi)}$ are the cell-integrated reaction, diffusion, migration and boundary flux terms and are given by:

$$(RHS)^{(rxns)} = V^{rel}_{i,j,k}R_s(\{\overline{(c_{s'})}_{i,j,k}\}) \tag{3.2}$$

$$(RHS)^{(diff)} = \frac{D_s}{\Delta x}\left(\left.\frac{\partial c_s}{\partial x}\right|_{x_+}\theta_{x_+} - \left.\frac{\partial c_s}{\partial x}\right|_{x_-}\theta_{x_-}\right) + \frac{D_s}{\Delta y}\left(\left.\frac{\partial c_s}{\partial y}\right|_{y_+}\theta_{y_+} - \left.\frac{\partial c_s}{\partial y}\right|_{y_-}\theta_{y_-}\right) + \frac{D_s}{\Delta z}\left(\left.\frac{\partial c_s}{\partial z}\right|_{z_+}\theta_{z_+} - \left.\frac{\partial c_s}{\partial z}\right|_{z_-}\theta_{z_-}\right) \tag{3.3}$$

$$\begin{aligned}(RHS)^{(mig)} = {} & \frac{(z_s u_s F)}{\Delta x}\left(\left(c_s\frac{\partial\Phi}{\partial x}\right)\Big|_{x_+}\theta_{x_+} - \left(c_s\frac{\partial\Phi}{\partial x}\right)\Big|_{x_-}\theta_{x_-}\right) + \frac{(z_s u_s F)}{\Delta y}\left(\left(c_s\frac{\partial\Phi}{\partial y}\right)\Big|_{y_+}\theta_{y_+} - \left(c_s\frac{\partial\Phi}{\partial y}\right)\Big|_{y_-}\theta_{y_-}\right)\\ & + \frac{(z_s u_s F)}{\Delta z}\left(\left(c_s\frac{\partial\Phi}{\partial z}\right)\Big|_{z_+}\theta_{z_+} - \left(c_s\frac{\partial\Phi}{\partial z}\right)\Big|_{z_-}\theta_{z_-}\right)\end{aligned} \tag{3.4}$$

$$(RHS)^{(flux,c)} = \frac{\Delta s_{i,j,k}}{\Delta x\Delta y\Delta z}D_s\left.\frac{\partial c_s}{\partial n}\right|_{active} = \frac{\Delta s_{i,j,k}}{\Delta x\Delta y\Delta z}\left(J_s - z_s u_s c_s\left(\sum_{s'}\frac{z_{s'}J_{s'}}{D_{s'}}\right)\Big/\left(\sum_{s'}\frac{z_{s'}^2 u_{s'}c_{s'}}{D_{s'}}\right)\right) \tag{3.5}$$

$$(RHS)^{(flux,\Phi)} = \frac{\Delta s_{i,j,k}}{\Delta x\Delta y\Delta z}z_s u_s F c_s\left.\frac{\partial\Phi}{\partial n}\right|_{active} = \frac{\Delta s_{i,j,k}}{\Delta x\Delta y\Delta z}\left(z_s u_s c_s\left(\sum_{s'}\frac{z_{s'}J_{s'}}{D_{s'}}\right)\Big/\left(\sum_{s'}\frac{z_{s'}^2 u_{s'}c_{s'}}{D_{s'}}\right)\right) \tag{3.6}$$

In these equations, $\Delta x\times\Delta y\times\Delta z$ are the cell dimensions and $V^{rel}_{i,j,k}$ is volume fraction of the cell in solution. In addition, $\theta_{x_+},\theta_{x_-},\theta_{y_+},\theta_{y_-},\theta_{z_+},\theta_{z_-}$ are the fractions of the cell faces that are in electrolyte solution (e.g. $\theta=1$ for interior cells and $0\leqslant\theta\leqslant 1$ for boundary cells) and Δs is the area of the active boundary element ($\Delta s=0$ for non-boundary cells). See Fig. 2

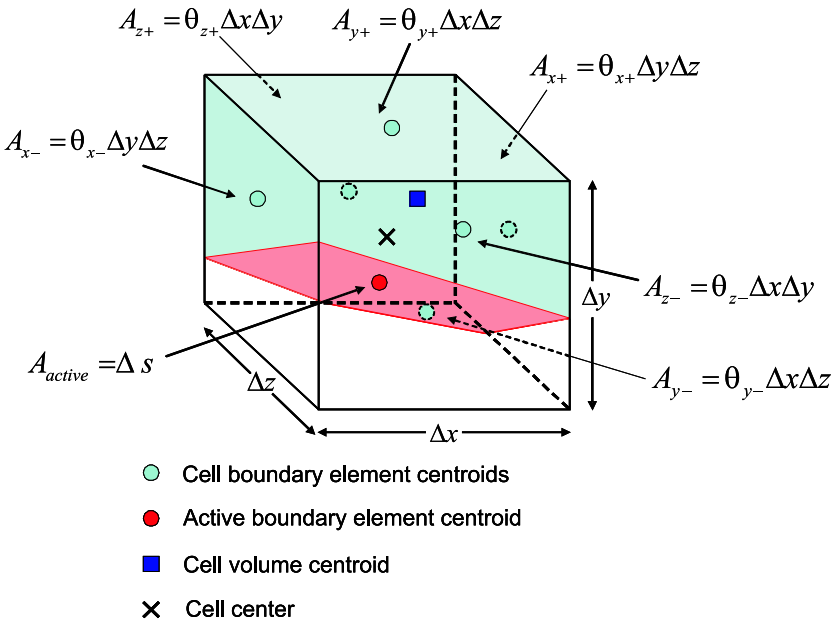

Fig. 2. A boundary cell is illustrated above showing the centroid location and area of cell boundary and active boundary elements, as well as cell volume centroid and cell center locations. These variables are used in the FV equations for cells cut by the active boundary.

for an illustration. Details with regard to calculating these geometrical quantities for boundary cells and complete details of the algorithm including cell-by-cell discretization formulas are included in [14].

The right hand side of Eq. (3.1) is then split into three sets of terms: (1) reaction terms, (2) diffusion terms (plus diffusive flux terms), and (3) migration terms (plus migration flux terms), as indicated by the superscript. To advance the concentration fields, $(c_s)_{i,j,k}$, from time t_n to $t_{n+1} = t_n + \Delta t$, two intermediate values, $(c_s)_{i,j,k}^{(*,rxns)}$ and $(c_s)_{i,j,k}^{(*,diff)}$, are calculated. Schematically, we do the following:

$$(c_s)_{i,j,k}^{(n)} \overset{reactions}{\rightarrow} (c_s)_{i,j,k}^{(*,rxns)} \overset{diffusion}{\rightarrow} (c_s)_{i,j,k}^{(*,diff)} \overset{projection}{\rightarrow} \Phi_{i,j} \overset{migration}{\rightarrow} (c_s)_{i,j,k}^{(n+1)}. \tag{3.7}$$

By *projection*, what is meant is that $\Phi_{i,j,k}$ is computed such that after migration, the charge neutrality constraint is satisfied at every solution-containing cell center.

Starting from Eq. (3.1), with the left hand side discretized in time, the algorithm proceeds as follows:

(1) *Reaction terms*

$$V_{i,j,k}^{rel} \frac{\left(\overline{(c_s)}_{i,j,k}^{(*,rxns)} - \overline{(c_s)}_{i,j,k}^{(n)}\right)}{\Delta t} = (RHS)^{(*,rxns)}. \tag{3.8}$$

(2) *Diffusion terms* (plus boundary flux)

$$V_{i,j,k}^{rel} \frac{\left(\overline{(c_s)}_{i,j,k}^{(*,diff)} - \overline{(c_s)}_{i,j,k}^{(*,rxns)}\right)}{\Delta t} = (RHS)^{(*,diff)} + (RHS)^{(flux,c)}. \tag{3.9}$$

(3) *Projection step*

$$\sum_s z_s V_{i,j,k}^{rel} \frac{\left(\overline{(c_s)}_{i,j,k}^{(n+1)} - \overline{(c_s)}_{i,j,k}^{(*,diff)}\right)}{\Delta t} = \sum_s z_s (RHS)^{(mig)} + \sum_s z_s (RHS)^{(flux,\Phi)}. \tag{3.10}$$

Eq. (3.10), together with the charge neutrality condition, $\sum_s z_s \overline{(c_s)}_{i,j,k}^{(n+1)} = 0$, leads to an implicit Poisson-like equation for the electric potential, $\Phi_{i,j,k}$ (contained in $(RHS)^{(mig)}$):

$$\Delta t \sum_s z_s (RHS)^{(mig)} = -V_{i,j,k}^{rel} \sum_s z_s \overline{(c_s)}_{i,j,k}^{(*,diff)} - \Delta t \sum_s z_s (RHS)^{(flux,\Phi)}. \tag{3.11}$$

(4) *Migration terms* (using $\Phi_{i,j,k}$ obtained in step 3)

$$V_{i,j,k}^{rel} \frac{\left(\overline{(c_s)}_{i,j,k}^{(n+1)} - \overline{(c_s)}_{i,j,k}^{(*,diff)}\right)}{\Delta t} = (RHS)^{(mig)} + (RHS)^{(flux,\Phi)}. \tag{3.12}$$

The resulting concentrations, $(c_s)_{i,j,k}^{(n+1)}$, are $O(\Delta t)$ accurate and satisfy the charge neutrality condition, $\sum_s z_s (c_s)_{i,j,k}^{(n+1)} = 0$, to machine precision. Eq. (3.8) is solved independently at each finite volume cell by Newton's method. Eqs. (3.9)–(3.12) lead to large sparse nearly symmetrical linear systems (one for each concentration field), which are solved with the Incomplete LU (ILU) preconditioned Biconjugate Gradient Stabilized (BCGSTAB) method as implemented in SPARSEKIT2 [11].

In [13], we showed that this time discretization is convergent with $O(\Delta t)$ accuracy, by writing Eq. (3.1) as a DAE system and verifying that the discretization is $O(\Delta t)$-consistent and 0-stable [2,3].

3.2. Moving boundary

The active, moving boundary between the bulk electrolyte solution and solid copper is tracked by the level set method. In the level set method a signed distance function, $\phi_{i,j,k}$, is defined at the corners of the finite volume cells (staggered grid). We adopt the convention that points in the electrolyte solution have $\phi > 0$ while points in the solid copper have $\phi < 0$. Level set methods have been shown to be robust and efficient numerical methods to track moving boundaries and to compute relevant interface properties, such as curvature and velocity [6–9].

For electrodeposition we compute the follow interfacial quantities: curvature, normal velocity, and adsorbed surface species coverages. In order to track these variables as the interface moves, we define extended variable fields, f^{ext}, on the staggered grid in a neighborhood (usually two FV cell diagonals) of the interface with the following properties:

$$f^{ext}(\vec{r}_s) = f_s, \tag{3.13}$$

$$\vec{\nabla} f^{ext} \cdot \vec{\nabla} \phi = 0, \tag{3.14}$$

where f is the surface-defined variable, $\vec{r}_s$ is a point on the surface and $f_s = f(\vec{r}_s)$. Eq. (3.13) says that f^{ext} should match f_s at surface point $\vec{r}_s$, while Eq. (3.14) says that f^{ext} should be constant along lines normal to the interface, i.e. $\vec{\nabla}\phi$.

We compute f^{ext} as follows. We begin with f defined at the boundary cell centroids. Denote these coordinates and the corresponding f values by $\{(\vec{r}_s, f_s)\}$.

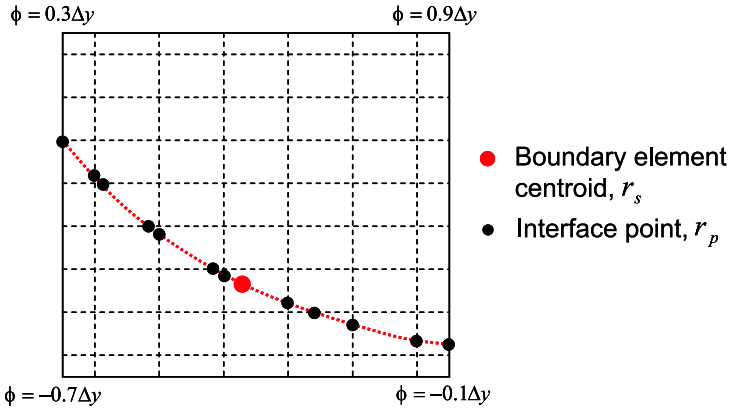

Fig. 3. Interface points and boundary element centroid are shown for a boundary cell. Illustration is two-dimensional for clarity, with $M_x = 6$ and $M_y = 8$.

Next we define a set of interface points $\{\vec{r}_p\}$ using a sub-grid within each boundary cell (see Fig. 3). The locations of the interface points are computed from ϕ using trilinear interpolation (see [14] for details). Then we use the function points, $\{\vec{r}_s, f_s\}$, to compute f at the finer resolution interface points, $\{\vec{r}_p, f_p\}$, which we do in the following way. Analytically, we write

$$f(\vec{r}_p) = \int_{\partial\Omega} f(\vec{r})\delta(\vec{r} - \vec{r}_p)dA, \tag{3.15}$$

where $\delta(\vec{r})$ is the two-dimensional delta function on the interface, i.e. $\int_{\partial\Omega} \delta(\vec{r})dA = 1$.

Eq. (3.15) may be approximated with a finite width delta function, $\delta_\epsilon(\vec{r})$ where $\epsilon \sim \Delta x$. The particular functional form used here is a Gaussian, $\delta_\epsilon(\vec{r}) \propto \exp(-r^2/2\epsilon^2)$. This yields

$$f(\vec{r}_p) \cong \int_{\partial\Omega} f(\vec{r})\delta_\epsilon(\vec{r} - \vec{r}_p)dA. \tag{3.16}$$

Finally, we apply Eq. (3.16) numerically as

$$f_p = \frac{\sum_s f_s \delta_\epsilon(\vec{r}_s - \vec{r}_p)\Delta A_s}{\sum_s \delta_\epsilon(\vec{r}_s - \vec{r}_p)\Delta A_s}, \tag{3.17}$$

where ΔA_s is the area of the interface element in boundary cell s. In practice, we truncate this summation outside a window of two cell widths surrounding point p. Note that for interface points near symmetry boundaries, we use ghost surface points s_g to exactly mimic the effect of the surface outside of the domain.

The last step is to map $f^{ext}_{i,j,k}$ (located at $\vec{r}_{i,j,k}$) to its closest interface point, $\vec{r}_p$. This mapping is computed by looping through all the interface points. For each interface point, the squared distance to all the staggered grid points in a neighborhood of $\vec{r}_p$ is computed. The minimum squared distance, d_p^2, is monitored and updated for each staggered grid point, along with the corresponding interface point, p. Finally, f^{ext} is given by

$$f^{ext}(\vec{r}_{i,j,k}) = f_p. \tag{3.18}$$

Correspondingly, the level set function is periodically redistanced (after moving by more than a FV cell diameter) as

$$\phi_{i,j,k} = d_p. \tag{3.19}$$

It is straightforward to verify that Eqs. (3.13) and (3.14) are satisfied to second order spatial accuracy provided that the interface point sub-grid is sufficiently well refined. First, each interface point location is second order accurate. Next, Eq. (3.17) computes f_p as a weighted average of nearby surface points, which is a form of second order interpolation. Finally, to see Eq. (3.14) is satisfied notice that the line connecting $\vec{r}_{i,j,k}$ and $\vec{r}_p$ (for which f^{ext} is constant) is parallel to $\vec{\nabla}\phi$ since point p is grid point (i,j,k)'s closest interface point. We note that a conceptually similar method has been reported in [10].

We advance the level set function using the advection equation $\left(\frac{\partial\phi}{\partial t} = -v_n\right)$,

$$\frac{\phi^{(n+1)}_{i,j,k} - \phi^{(n)}_{i,j,k}}{\Delta t} = -(v^{ext}_n)^{(n)}_{i,j,k}, \tag{3.20}$$

which preserves the signed distance property.

Curvature can be calculated directly from the level set function,

$$\kappa = \vec{\nabla}\cdot\hat{n} = \vec{\nabla}\cdot\left(\frac{\vec{\nabla}\phi}{|\vec{\nabla}\phi|}\right), \tag{3.21}$$

where all terms are evaluated by second order centered difference formulas. However, accuracy will be degraded wherever $\kappa \sim \Delta x^{-1}$. For electrodeposition, we found that no matter how refined we made our grid, regions of high curvature would develop near the bottom corners of the trench and result in inaccuracies when computing Eq. (2.14). These inaccuracies led to excessive numerical mass loss of key adsorbed surface species.

We note that this mass loss issue was not observed for the standard expanding and contracting circle test cases. For these cases, we observed excellent mass conservation with $O(\Delta x^2)$ accuracy [14]. Although there are fully conservative approaches to the mass conservation issue [28,29], we chose to use an empirical approach that we believe is useful due its simplicity and effectiveness. After calculating the curvature according to Eq. (3.21), we multiply by a scaling factor, C_κ. By performing numerical experiments, we found a good choice for C_κ to be

$$C_\kappa = 1.0 + 0.5(1 - \exp(-1.5|\kappa|\Delta x)). \tag{3.22}$$

From this form, we can see that $C_\kappa \approx 1$ for $|\kappa| \ll \Delta x^{-1}$, which says we do not alter the curvature calculation in well-resolved regions of the interface. However, $C_\kappa \approx 1.5$ for $|\kappa| \sim \Delta x^{-1}$. This says we scale the curvature by a factor of up to 1.5 in unresolved regions of very high curvature. Numerical tests of a conformally filling trench of aspect ratio 2 having an initially uniform accelerator coverage reveal that the total accelerator mass changes by less than 3% for three different resolution grids up to the point that a seam is formed. This is in contrast to a mass loss of around 25% when using the unmodified curvature calculation on grids of the same resolution. Notice that the mass loss becomes worse as the grid is refined, which is due to steeper gradients in the surface coverage. We imagine that conformal infill represents the "worst case scenario," since the sharp corner at the trench bottom remains unresolved over the entire infill process. During electrodeposition, the accumulation of accelerator at these corners would cause a local increase in deposition rate causing rounding of the corners (see Figs. 4 and 5).

3.3. Coupling algorithm

The bulk electrolyte and surface models are coupled by the fluxes, $\{J_{s'}\}$, and bulk electrolyte concentrations and potential at the surface, $\{c_{s'}, \Phi\}$. The fluxes are output from the surface reaction code and provide boundary conditions to the bulk

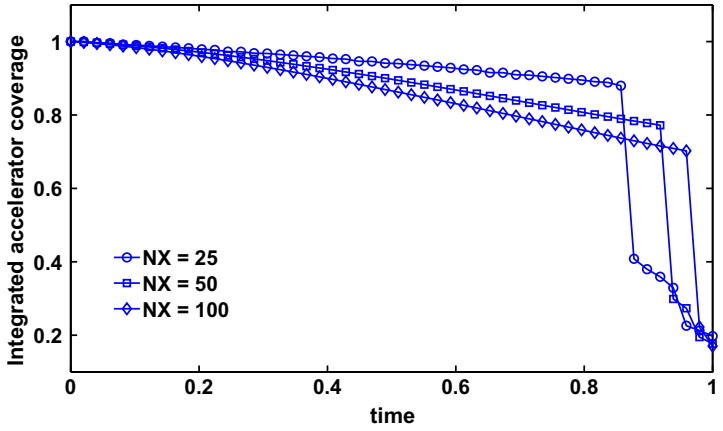

Fig. 4. Total accelerator coverage integrated over trench during conformal infill starting with uniform coverage of accelerator using unscaled curvature.

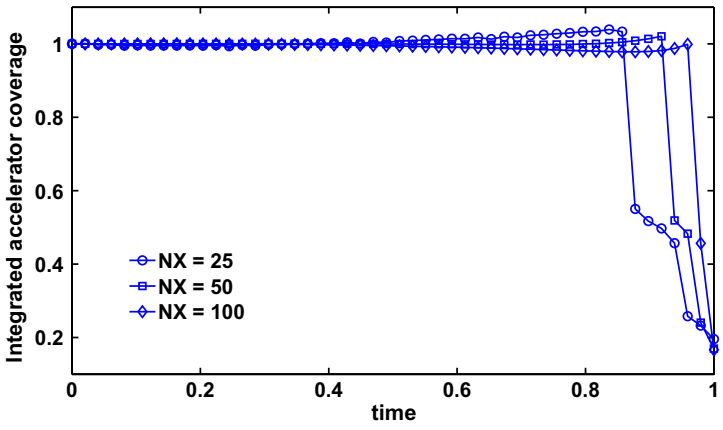

Fig. 5. Total accelerator coverage integrated over trench during conformal infill starting with uniform coverage of accelerator using curvature with scale factor, C_κ, given by Eq. (3.22).

electrolyte code, while the concentrations and potential are output from the bulk electrolyte code and act as parameters in the surface reaction code. In addition, the flux of copper J_{Cu} is used to advect the level set function (Eq. (2.11)). There are many ways that one can try to couple these computations. In our work, we explored three different approaches: *explicit coupling*, *fully implicit coupling*, and *semi-implicit coupling*. Explicit coupling is efficient per time step, but becomes unstable and inaccurate for most realistic surface reaction rates using reasonable time step sizes. Fully implicit coupling is stable and accurate for a much larger range of surface reaction rates than explicit coupling, but becomes inefficient and occasionally nonconvergent for very stiff surface reactions. Finally, semi-implicit coupling is stable and accurate for all the surface reaction rates we have tested, while attaining the same high efficiency per time step as explicit coupling.

In the subsections that follow, we will briefly describe these methods. To aid in the discussion, we introduce the following notation. Define one iteration of the surface code (reactions only) to be **F**,

$$\theta^{(n+1)} = \mathbf{F_1}(\theta^{(n)}; c), \tag{3.23a}$$

$$J^{(n+1)} = \mathbf{F_2}(\theta^{(n)}; c). \tag{3.23b}$$

Here, the bulk concentration (and potential) c may be regarded as a constant parameter during the integration window $[t^{(n)}, t^{(n)} + \Delta t]$. Similarly, we define one iteration of the bulk code to be **G**,

$$c^{(n+1)} = \mathbf{G}(c^{(n)}; J). \tag{3.24}$$

This time, the active boundary flux J is regarded as the constant parameter. We will show that different strategies for choosing these input parameters, c and J, lead to very different outcomes in accuracy and stability of the coupling.

3.3.1. Explicit coupling

Explicit coupling proceeds as follows for each time step. First the surface code is run and the flux output is used as the input to the bulk code. Then the bulk code is run and the concentrations and potential are used as inputs to the surface code. Using the notation introduced above, explicit coupling may be written as

$$\theta^{(n+1)} = \mathbf{F_1}(\theta^{(n)}; c^{(n)}), \tag{3.25a}$$

$$J^{(n+1)} = \mathbf{F_2}(\theta^{(n)}; c^{(n)}), \tag{3.25b}$$

$$c^{(n+1)} = \mathbf{G}(c^{(n)}; J^{(n+1)}). \tag{3.25c}$$

To understand the instabilities that can arise in this coupling, substitute (3.25b) into (3.25c), yielding

$$c^{(n+1)} = \mathbf{G}(c^{(n)}; \mathbf{F_2}(\theta^{(n)}; c^{(n)})). \tag{3.26}$$

Next compute the Jacobian (for simplicity, we consider the case of fixed θ, which more clearly illustrates the dominant mechanism causing instability),

$$\frac{\partial c^{(n+1)}}{\partial c^{(n)}} = \frac{\partial \mathbf{G}}{\partial c} + \frac{\partial \mathbf{G}}{\partial J}\frac{\partial \mathbf{F_2}}{\partial c}. \tag{3.27}$$

The first term, $\frac{\partial \mathbf{G}}{\partial c}$, is stable since the bulk code always converges for constant flux input, as reported in [13]. Therefore, any instabilities that arise are due to the second term, which describes the bulk–surface coupling. The first part of the second term is well-behaved, $\left\|\frac{\partial \mathbf{G}}{\partial J}\right\| \sim 1$, since it a measure of the bulk sensitively to the flux boundary condition, which is controlled by diffusion. However, the second part of this term is a measure of the surface reaction sensitivity to bulk concentrations. Since many of the surface reactions are very fast, this results in $\left\|\frac{\partial \mathbf{F_2}}{\partial c}\right\| \gg 1$ for reasonably small time steps. This causes explicit linkage to be highly unstable. For example, using typical surface reaction rate constants time step size must be reduced by more than four orders of magnitude (from 10^{-3} s to less than 10^{-7} s) compared to the time step required to accurately track system changes.

3.3.2. Fully implicit coupling

In fully implicit coupling, both the concentration input to the surface code and the flux input to the bulk code are taken at the next time step, $(n+1)$. Using the notation defined above, we have

$$\theta^{(n+1)} = \mathbf{F_1}(\theta^{(n)}; c^{(n+1)}), \tag{3.28a}$$

$$J^{(n+1)} = \mathbf{F_2}(\theta^{(n)}; c^{(n+1)}), \tag{3.28b}$$

$$c^{(n+1)} = \mathbf{G}(c^{(n)}; J^{(n+1)}). \tag{3.28c}$$

Computing the Jacobian, we find

$$\frac{\partial c^{(n+1)}}{\partial c^{(n)}} = \left(\mathbf{I} - \frac{\partial \mathbf{G}}{\partial J}\frac{\partial \mathbf{F_2}}{\partial c}\right)^{-1}\frac{\partial \mathbf{G}}{\partial c}. \tag{3.29}$$

We see that the terms responsible for instability in the *explicit coupling* method now appear in a matrix inverse, which should stabilize this coupling.

However, solving (3.28a)–(3.28c) consistently is not a simple task. They comprise a very large nonlinear equation system connected at the boundary, which may be written compactly in terms of the unknown boundary fluxes as

$$J^{(n+1)} - \mathbf{F_2}(\mathbf{G}(J^{(n+1)})) = 0, \tag{3.30}$$

The challenge for this method lies in approximating the Jacobian (required for Newton iteration of (3.30)) in an accurate and efficient manner. Since we found semi-implicit coupling to be a better choice for our problems, we will not describe fully implicit coupling in detail. The interested reader is referred to [14].

3.3.3. Semi-implicit coupling

Semi-implicit coupling attempts to capture the best of both worlds: explicit coupling and fully implicit coupling. The fact is it does even better than that. It is more stable and robust than fully implicit coupling, but every bit as efficient as explicit coupling. Semi-implicit coupling solves the following equations at each time step:

$$\theta^{(n+1)} = \mathbf{F_1}(\theta^{(n)}; c^{(n)}), \tag{3.31a}$$

$$J^{(n+1)} = \mathbf{F_2}(\theta^{(n)}; c^{(n)}), \tag{3.31b}$$

$$c^{(n+1)} = \mathbf{G}(c^{(n)}; J^{(*)}). \tag{3.31c}$$

Here, $J^{(*)}$ is the semi-implicit flux given by the mixed time step expression:

$$J_s^{(*)} = \alpha_s^{(n)} + \beta_s^{(n)} c_s^{(n+1)}, \tag{3.32}$$

where α and β are functions of θ and c. Eq. (3.32) uses the fact that J_s is linear in c_s. In other words, both α_s and β_s are independent of c_s. This linearity assumption holds in our model since surface reaction rates are proportional to bulk concentrations with constant rest potentials for Faradaic reactions. Furthermore, they are trivial to compute in terms of the flux function $\mathbf{F_2}$,

$$\alpha_s^{(n)} = \mathbf{F_2}\left(\theta^{(n)}; c^{(n)} - c_s^{(n)}\right)\Big|_s, \tag{3.33a}$$

$$\beta_s^{(n)} = \mathbf{F_2}\left(\theta^{(n)}; c^{(n)} - c_s^{(n)} + 1_s\right)\Big|_s - \alpha_s^{(n)}. \tag{3.33b}$$

Thus, $\alpha_s^{(n)}$ is calculated by computing the flux function $\mathbf{F_2}$ with $c_s = 0$, and $\beta_s^{(n)}$ is calculated by computing the flux function with $c_s = 1$ and subtracting $\alpha_s^{(n)}$. Although linearity does not hold for more refined surface models, we can modify this approach by allowing $\beta_s^{(n)}$ to have c_s dependence. We have done this successfully for rest potentials with c_s dependence.

The linear decomposition of J_s with respect to c_s is precisely what gives this method its high computational efficiency. Recall from Section 3.1 that the active boundary flux terms are included in the *diffusion step* of the bulk splitting algorithm. In that step, we solve a separate linear system for each bulk species, s. Since our expression for $J_s^{(*)}$ is linear in c_s, the implicit part $(\beta_s^{(n)} c_s^{(n+1)})$ may be readily incorporated into the linear system and solved at little or no extra computational expense.

As for the stability of semi-implicit linkage, let us first look more closely at the most problematic term, $\frac{\partial J}{\partial c}$. Writing $J_s = \alpha_s + \beta_s c_s$, we examine the Jacobian

$$\frac{\partial J}{\partial c} = \frac{\partial \alpha}{\partial c} + \Lambda_c \frac{\partial \beta}{\partial c} + \Lambda_\beta, \tag{3.34}$$

where Λ_c and Λ_β are the diagonal matrices with elements c and β, respectively.

In our surface reaction model, the largest magnitude terms (by far) are contained in Λ_β. This is because the fastest reactions are those that rapidly consume bulk species s resulting in a flux, $J_s \propto -c_s$, with a large proportionality constant, i.e. $|\beta_s| \gg 1$. These reactions quickly become diffusion rate limited with c_s attaining very small values along the active boundary, resulting in a bounded and well-behaved product, $\beta_s c_s$. By this same reasoning, one can see that the product $\Lambda_c \frac{\partial \beta}{\partial c}$ is also bounded and well-behaved.

To see how the preceding discussion helps us understand the stability of this coupling method, we examine the Jacobian, $\frac{\partial c^{(n+1)}}{\partial c^{(n)}}$. Taking the directional derivative of Eq. (3.31c), we obtain

$$\frac{\partial c^{(n+1)}}{\partial c^{(n)}} = \frac{\partial \mathbf{G}}{\partial c} + \frac{\partial \mathbf{G}}{\partial J}\left(\frac{\partial \alpha}{\partial c} + \Lambda_c \frac{\partial \beta}{\partial c} + \Lambda_\beta \frac{\partial c^{(n+1)}}{\partial c^{(n)}}\right). \tag{3.35}$$

Solving for $\frac{\partial c^{(n+1)}}{\partial c^{(n)}}$ yields

$$\frac{\partial c^{(n+1)}}{\partial c^{(n)}} = \left(\mathbf{I} - \frac{\partial \mathbf{G}}{\partial J}\Lambda_\beta\right)^{-1}\left(\frac{\partial \mathbf{G}}{\partial c} + \frac{\partial \mathbf{G}}{\partial J}\left(\frac{\partial \alpha}{\partial c} + \Lambda_c \frac{\partial \beta}{\partial c}\right)\right). \tag{3.36}$$

Now we can see more clearly why this method works. The implicit part of the flux containing the stiffest terms appears in the inverse matrix, which provides stability. This strategy may also be regarded as a highly efficient, diagonal approximation Λ_β to the Jacobian term $\frac{\partial \mathbf{F_2}}{\partial c}$ described in the previous section, which is made even more efficient by the observation that linearity of the flux expression enables a semi-implicit method.

We conclude this section by discussing similar methods that have been implemented in the literature. It seems that our method is most directly comparable to domain decomposition methods. One situation where domain decompositions are often performed is when the physical domain can be split into two or more subdomains, each governed by different physical equations. The subdomains are usually coupled by the continuity or equality of certain quantities along the boundary. In fact, the explicit coupling algorithm is nothing but the Schwarz alternating method applied to the surface and bulk subdomains. There is a considerable body of theory describing the conditions for convergence [19]; obviously, our problem does not meet these criterion. In our search, we found the most applicable literature to be for acoustic systems with solid and fluid subdomains. For this problem, effective algorithms have been proposed using variants of the Schwarz alternating method [16,17]. To improve on the accuracy, others have applied schemes to iterate between the subdomains (often on a series of coarser grids) to better satisfy the boundary condition between the two subdomains [18]. Such techniques have been widely implemented in FETI methods (used in structural mechanics) and have a well-developed theory with regard to convergence properties [19]. Our fully implicit algorithm may be regarded as a variant of such methods. We are not aware of an analog of our semi-implicit coupling method.

4. Tests and validations

4.1. Accuracy tests

4.1.1. COMSOL comparisons

Since analytical solutions to our entire model are not feasible, we decided to compare simulation results using our algorithm to COMSOL simulation results using the Nernst–Planck module. For simplicity we consider a fixed flat one-dimensional surface rather than a filling trench. However, no other simplifications are made; the full model of Section 2.3 is solved. We set a boundary layer thickness of 50 μm and a fixed applied potential of −0.5 V. For our algorithm, we use

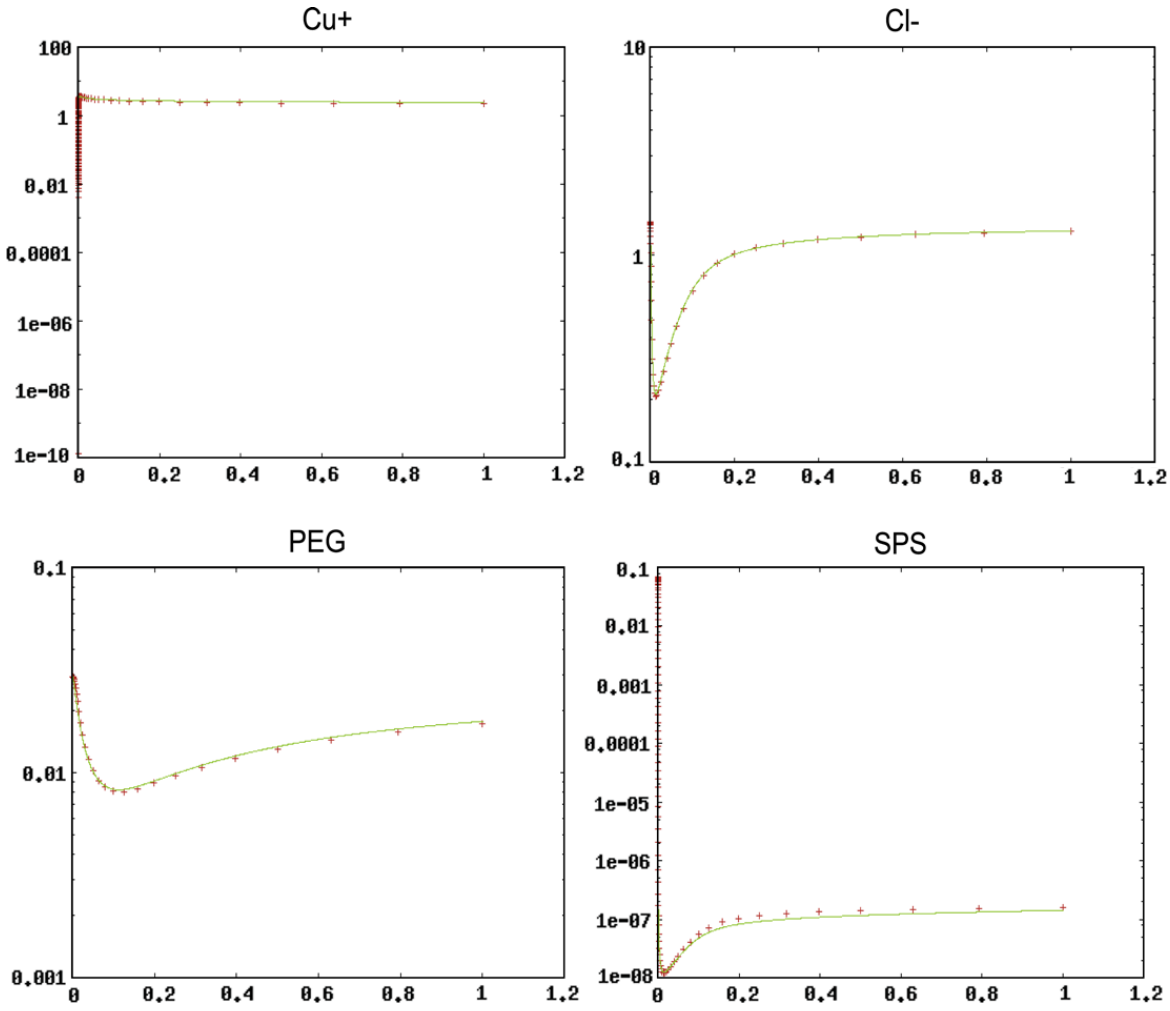

Fig. 6. Above we show COMSOL simulation results (indicated by crosses) of our model along with simulation results using our algorithm (solid line). We plot the transient concentrations of a few key solution species (Cu^+, Cl^-, *PEG*, and *SPS*) at the active boundary for a one-dimensional flat surface.

360 FV cells in the solution ranging in thickness from 1 nm at the active surface to just under 1 μm at the far field. We specify an initial time step size, $\Delta t = 5 \times 10^{-7}$ s, and gradually increase (over the duration of the initial transient) to 2.5×10^{-4} s, which corresponds to a typical resolution and time step size used in our trench simulations. We plot the transient concentrations of four key solution species evaluated at the interface. We see excellent agreement to within about 5 % over the entire time interval (see Fig. 6).

4.1.2. Convergence and self-consistency

Here we describe a set of tests done to validate the accuracy and confirm the convergence properties of our coupled bulk electrolyte plus surface model. First, we test the accuracy for a fixed boundary system after a relatively short time has elapsed. This test allows us to isolate the coupling without considering the interaction with the moving boundary. Additionally, the short time interval allows us to refine the grid and time step sufficiently to verify the expected convergence. Next, we test the accuracy of our method for more realistic numerical simulation conditions. This test includes the moving boundary as well as long time dynamics. It was not computationally feasible to run this case at a fine enough resolution to see the expected convergence. Instead, we regard this test as an assessment of "real world" qualitative performance of our method.

For these tests, we used the reaction mechanism described in Section 2.3. The domain shape was a fixed trench shape (180 nm width, 5:1 aspect ratio and 50 μm boundary layer) and was run for $t = 0.5$ s (during the initial system transient). A total of four computational grids were used, with $\frac{\Delta t}{\Delta x^2}$ fixed. The resolutions and fixed time step sizes are given in Table 8.

The coarser grid solutions (grids 1, 2 and 3) were compared to grid 4 by averaging the grid 4 numerical solutions over 8×8, 4×4 and 2×2 sub-grid squares, respectively. Errors were computed using the following definition for the error in species s,

$$(E_{L_\infty})_s = \frac{\max_{i,j}\left|(c_s)_{i,j} - (c_s)_{i,j}^{(exact)}\right|}{\max_{i,j}\left|(c_s)_{i,j}^{(exact)}\right|}. \tag{4.1}$$

The typical value of the fluxes, J_s, ranged from 10^{-1} mol/m^2 s for Cl^- and PEG, to greater than 100 mol/m^2 s for Cu^+ and for Cu^{2+}. Other species are have either zero or order 1 mol/m^2 s flux.

Numerical errors are plotted in Fig. 7 for each of the species concentration fields. The first observation is that there is a large difference in the accuracy of the most (HSO_4^-) to least (Cu^+) accurate species. This is because some species (such as HSO_4^-) change very little relative to their far-field concentration and reach a steady-state quickly. In addition, species with small flux to diffusivity ratios tend to be computed more accurately, which is the result of small, well-resolved gradients near the active boundary. On the other hand, species that are not present as additives but are generated by fast surface reactions

Table 8
Grid resolutions and time step sizes for coupled models, short timescale tests.

Grid no.	Uniform region res.	Nonuniform region res.	Time step, Δt (s)
1	20×50	20×40	4×10^{-3}
2	40×100	40×80	1×10^{-3}
3	80×200	80×160	2.5×10^{-4}
4	160×400	160×320	6.25×10^{-5}

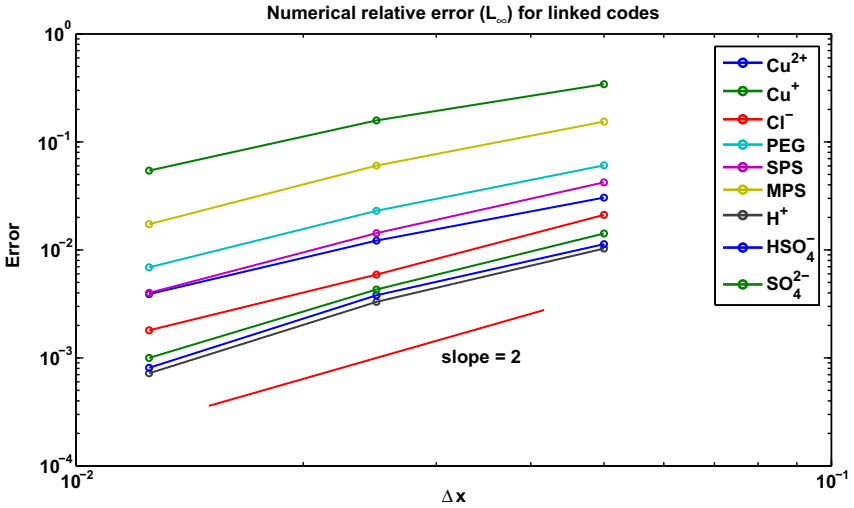

Fig. 7. Here we plot the numerical error in the coupled surface–bulk code for each chemical species.

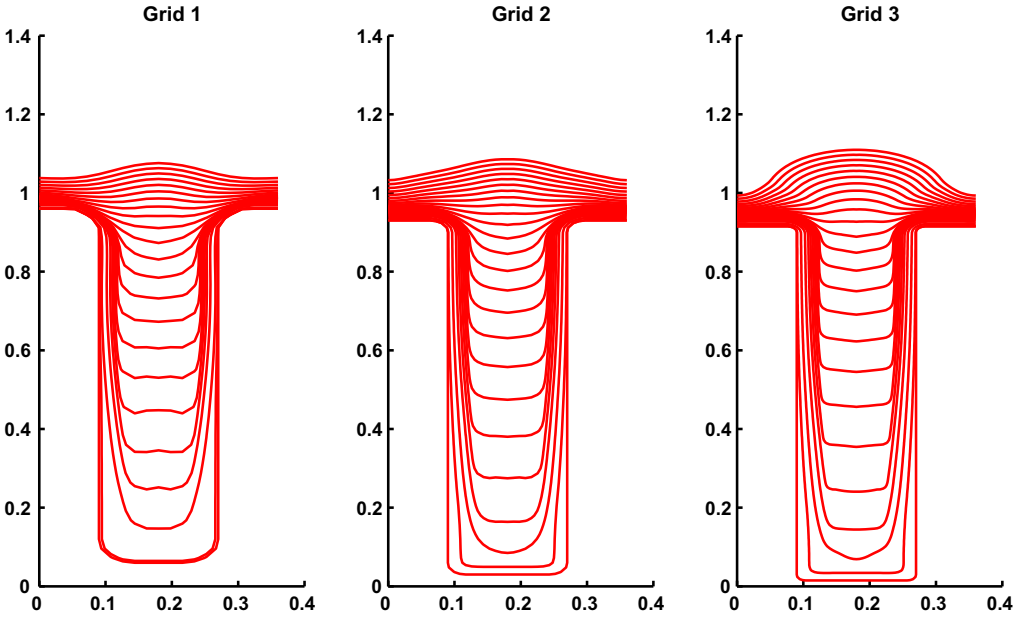

Fig. 8. Infill contours shown at 4 s intervals for trench infill simulations *with additives* on grids 1, 2 and 3.

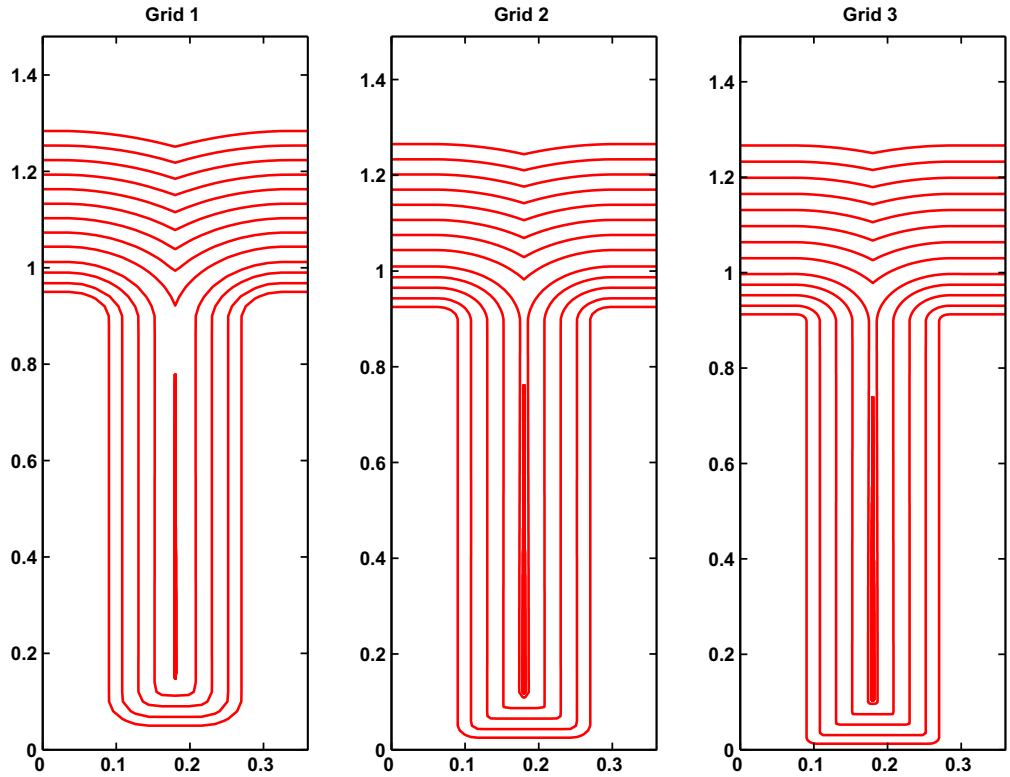

Fig. 9. Infill contours shown at 8 s intervals for trench infill simulations *without additives* on grids 1, 2 and 3.

and consumed equally quickly (e.g. Cu^+, MPS) tend to have more complex spatial concentration distributions and are more difficult to resolve, making them subject to larger numerical errors. We see that the numerical errors in $[Cu^{2+}]$, $[Cu^+]$ and $[MPS]$, in particular, converge slightly slower than the expected $O(\Delta x^2)$, which is similar to the convergence results reported in Fig. 7 of [13]. The large flux creates a sharp boundary layer in $[Cu^+]$ near the active boundary during the transient that is difficult to resolve with our uniform grid spacing.

Next we tested the accuracy of our coupled code over long time scales, namely the entire process of trench infill (~100 s for 180 nm wide trench). We already learned that the standard error measures can be quite large for species with steep gradients in their concentration fields even over short timescales. Now we investigate how these errors contribute to the overall

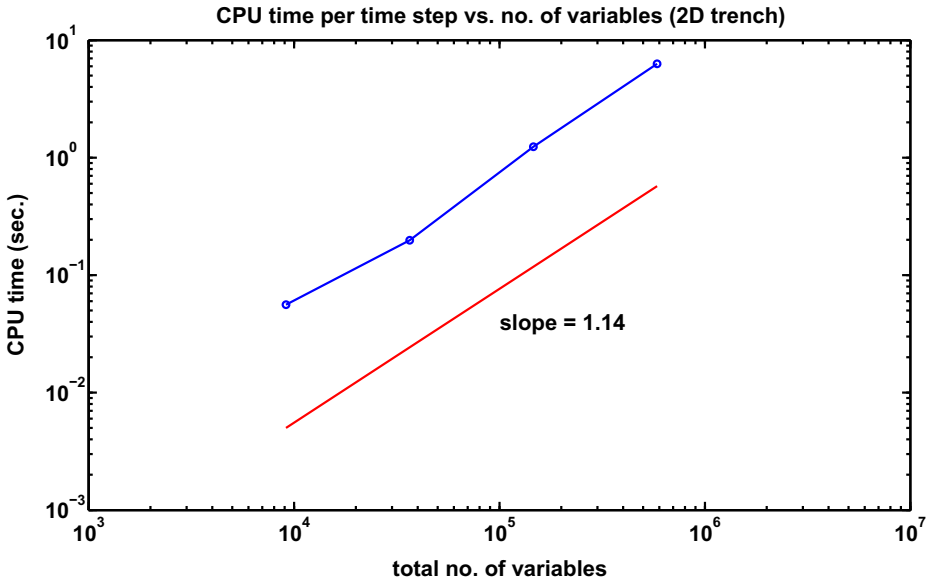

Fig. 10. Efficiency of the coupled simulation for 2D trench problem.

Table 9
CPU time (P4 – 3.2 GHz) data for 2D trench problem.

Grid dimensions	CPU time/time step (s)	No. of time steps	Total CPU time
20×90	0.056	2.5×10^4	37 min
40×180	0.198	1.0×10^5	5.5 h
80×360	1.24	4.0×10^5	5.7 days
160×720	6.30	1.6×10^6	117 days

qualitative features of the simulation. To this end, we perform two simulations (additive-free and additive infill) each on three different grids (grids 1–3 of Table 8). All simulations start out with small initial time step size, $\Delta t_{min} = 5 \times 10^{-7}$ s to better resolve the initial transient, and is increased gradually.

In Figs. 8 and 9, we compare the infill contours obtained on our three grids. Here, we see a remarkable qualitative similarity. All three simulations for different grid resolutions predict a similar size void for additive-free infill, while they predict bottom-up superfilling (trench filling without voids or seams) for infill with additives.

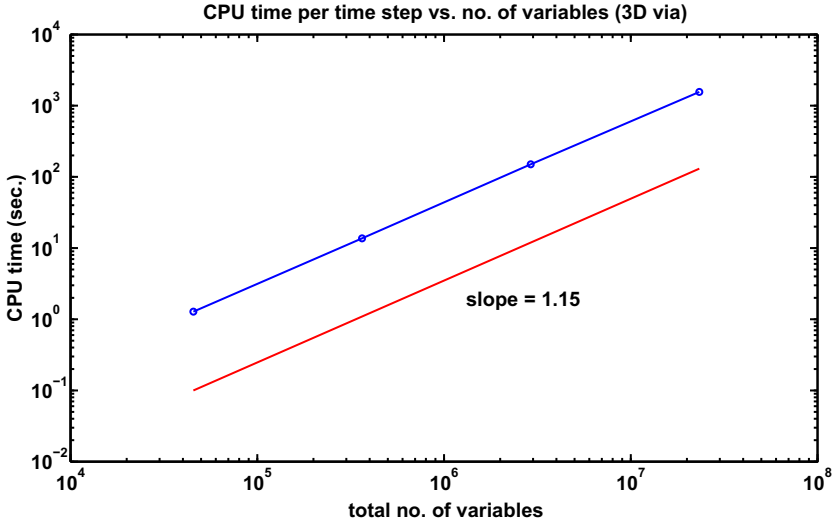

Fig. 11. Efficiency of the coupled simulation for 3D via problem.

Table 10
CPU time (P4 – 3.2 GHz) data for 3D via problem.

Grid dimensions	CPU time/time step (s)	No. of time steps	Total CPU time
$20 \times 90 \times 10$	1.28	2.5×10^4	8.9 h
$40 \times 180 \times 20$	13.7	1.0×10^5	15.9 days
$80 \times 360 \times 40$	150	4.0×10^5	1.9 years
$160 \times 720 \times 80$	1558	1.6×10^6	79 years

4.2. Efficiency tests

Here, we measure the efficiency for both a 2D and 3D problem (using semi-implicit coupling).

First, we consider the trench shaped domain in 2D. CPU time versus problem size is shown in Fig. 10. Actual CPU times for entire simulations are shown in Table 9. The scaling complexity of our method is excellent. With optimal complexity scaling being $O((N_{eqns})^{1.0})$, our coupled method scales as $O((N_{eqns})^{1.14})$. We note that this is a huge improvement over previous methods, that scaled approximately as $O((N_{eqns})^2)$ [1]. Most of our simulations are run on grid 2, requiring a total CPU time of about 5.5 h running on a single core of a 3.2 GHz Pentium 4 processor. We note that we can perform entire numerical simulations on grid 1 in about 37 min. This is remarkable since we verified that this grid yields good qualitative accuracy. As a point of reference, previous methods using even coarser grids to solve this same set of equations required 7–8 h [12]. This is a 15-fold increase in computational efficiency that only gets better as the grid is further refined!

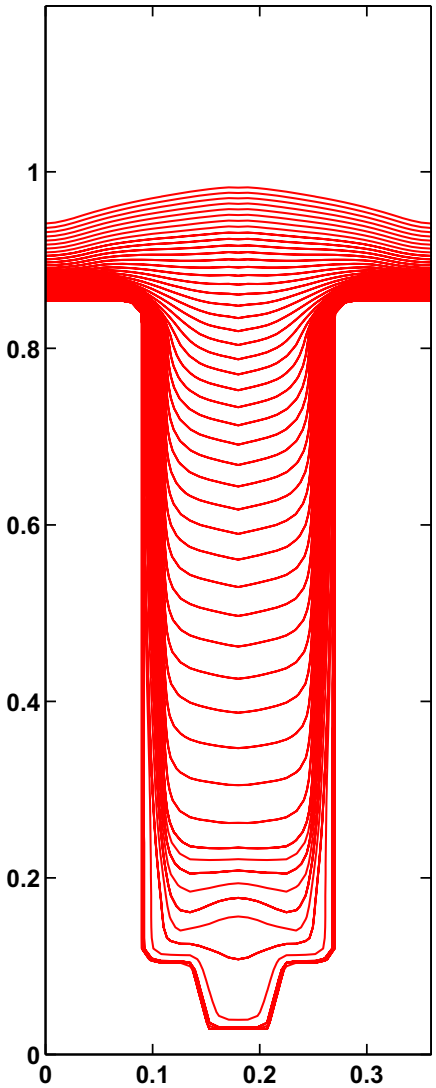

Fig. 12. Simulation of copper electrodeposition with additives for 180 nm wide and 180 nm long, 5:1 aspect ratio trench with 90 nm 1:1 aspect ratio tapered cylinder at the bottom: infill contours perpendicular to the trench axis are plotted at 1 s intervals for first 10 s followed by every 2 s.

Next, we measured the efficiency for the 3D problem. The specific geometry that we consider is a 2D trench revolved about its central axis, thus a right cylindrical via. We note that even though this problem is axially symmetric, it represent a fully 3D domain since we use cartesian coordinates in our simulation code. The computational complexity is plotted in Fig. 11 and the CPU times for entire simulations are given in Table 10. Note that the last two simulations were not performed in their entirety but instead were solved for a few time steps and the timing results were extrapolated. The complexity scales about the same as for the 2D trench problem, but the overall constant of proportionality is about four times larger for the 3D problem. This can be attributed to the extra couplings of the unknown variables, making the linear solvers less efficient, as is observed by the increased number of iterations required for convergence. It is clear from the total simulation time that highly resolved 3D simulations are out of the question at this point. However, from the results of Section 4.1.2, we do not actually have to use highly resolved grids to make qualitatively accurate predictions. One way we can envision to improve the efficiency is to parallelize the linear solvers. We conclude this section by noting that our semi-implicit algorithm has been stable and robust for all the surface model parameters that we have tested.

5. Application to 3D via infill

In this section, we consider the infill of a three-dimensional via consisting of a 2D trench with a tapered cylinder extending down from the trench bottom. The trench width is 180 nm with 5:1 aspect ratio, and the cylinder diameter is 90 nm with 1:1 aspect ratio. These geometries are often used in computer processor chips to connect one level of interconnects to the next lower level, and are of interest to researchers at IBM. The surface chemistry mechanism is that described in Section 2.3 with a constant current of 75 A/m^2, and may be understood as follows. Two suppressor additives (Cl^- and *PEG*) and one accelerator additive (*SPS*) are included in solution in addition to the basic $CuSO_4$ salt plus acid H_2SO_4 cocktail required for electrodeposition. Note that we set the leveler additive (*HIT*) concentration to zero for simplicity.

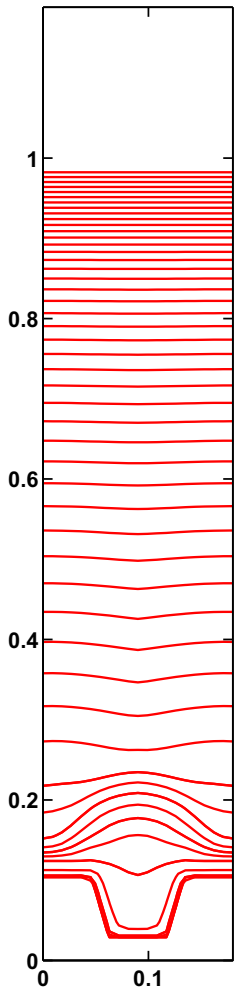

Fig. 13. Simulation of copper electrodeposition with additives for 180 nm wide and 180 nm long, 5:1 aspect ratio trench with 90 nm 1:1 aspect ratio tapered cylinder at the bottom: infill contours parallel to the trench axis are plotted at 1 s intervals for first 10 s followed by every 2 s.

The infill profiles are plotted along the two planes of symmetry. Fig. 12 shows infill contours along the plane perpendicular to the trench and through the cylinder's diameter. Fig. 13 displays infill contours along the plane parallel to the trench axis and through the cylinder's diameter.

The simulations reveal that the small cylinder initially begins to fill conformally. As it fills, the deposition rate near the cylinder axis rapidly increases, leading to the formation of a large hump at the trench bottom. The hump then spreads out (more so along the trench axis) as the trench section starts to fill. The trench then continues to fill from the bottom up, i.e. superfilling.

The mechanism for the observed infill may be understood as follows. To begin, Cu^{2+} is readily reduced to Cu^{+} at the surface. Next, Cu^{+} and Cl^{-} react to produce adsorbed *CuCl* on the surface. Solution additive *PEG* then reacts rapidly with adsorbed *CuCl*, being converted to *CuClPEG* on the surface, which occupies a much greater surface area (estimated at 40 times) than *CuCl* alone. This rapidly covers the top part of the trench and continues to spread down into the trench as depleted *PEG* is gradually replenished by diffusion. Copper deposition is effectively suppressed on those parts of the trench surface covered by *CuClPEG* (suppressor). During the window of time that the trench bottom is free of *CuClPEG*, additive *SPS* is converted to intermediate *MPS*, which builds in solution. *MPS* and Cu^{+} then react to create adsorbed *Cu(I)thiolate* (accelerator), which catalyzes copper deposition. As deposition occurs more rapidly near the trench and cylinder bottom and they begin to fill, the adsorbed *Cu(I)thiolate* becomes more concentrated due to the surface's concave curvature, thus covering an increasing fraction of the surface. This so-called "curvature enhanced accelerator coverage" [15] is greatest near the trench corners initially but quickly builds at the cylinder bottom, resulting in the high rate of deposition near the center of the cylinder and the subsequent hump formation. The hump dissipates as new *Cu* sites open up and *Cu(I)thiolate* is rapidly adsorbed. After a short time (about 10 s), the entire trench bottom is covered with accelerator while the sidewalls and top are covered with suppressor. In this way, deposition continues at the trench bottom while being suppressed elsewhere. In order to achieve this balance, it was important to choose additives with very specific reaction chemistries and rate constants in addition to the choosing the right amounts of each. This is the main goal of such a simulation: to identify the properties of and quantities of additives that will result in superfilling for these and ever-smaller geometries.

6. Conclusions

In this work we have developed a general and fully three-dimensional numerical strategy for simulating electrochemical systems on irregular domains with moving boundaries. This involves solving the governing partial differential equations with algebraic constraints in the bulk electrolyte which are stiffly coupled to an active surface where chemical reactions and other physics occur. Our method makes only a few assumptions about the active surface, namely that it is driven by the bulk chemical species concentrations and applied potential and that it produces a flux of each species back into solution. Otherwise, the details of the surface model are of little consequence. The particular application we study here is copper electrodeposition as applied to filling trench/via interconnects in computer processors.

To summarize, our computational method for the bulk electrolyte region splits the three distinct physical phenomena that occur into fractional steps. The homogeneous reactions and diffusion are handled with backward Euler discretization to treat stiffness, while the electrical migration is treated with a projection step which satisfies the charge neutrality constraint exactly. Spatial discretization is performed using the finite volume method, which conserves species mass exactly and retains second order spatial accuracy, even near the irregular boundary. A nonuniform grid region above the active surface is used to resolve the diffusion boundary layer that is about two orders of magnitude thicker than the trench dimensions. The level set method is adopted to move the interface, but modified to prevent the degradation of accuracy that can result from the first order accurate fast marching method. The closest point algorithm is used to reconstruct the interface with second order accuracy before redistancing is performed. Curvature is calculated using an empirically determined scaling factor multiplied by the finite difference formula. This was necessary in order to better conserve surface species in regions of very high curvature. To couple the bulk electrolyte and active surface regions, we developed a semi-implicit coupling method that handles the stiff coupling problem robustly and efficiently.

With numerical experiments, we found that the CPU times scale as a small power of the problem size, N, namely $T_{\text{CPU}} \propto N^{p}$ where $p \approx 1.15$. Direct comparison of our method's efficiency with existing numerical strategies for solving this same complex model revealed that our method can compute in about 30 min what previously took 8 h. In addition, previous methods scaled very poorly with grid refinement ($p \approx 2$) and were not second order accurate near the moving boundary.

Finally, we applied our coupled bulk–surface algorithm to study the infill of a three-dimensional via. We found that our method is able to predict the superfilling phenomenon seen experimentally with a careful balance of solution additives, as well as some subtle details of the infill characteristics.

Acknowledgments

We would like to give a special thanks to Dr. Richard Alkire and Dr. Richard Braatz at the University of Illinois, Urbana-Champaign. Our ongoing collaboration with them has motivated our study of this problem and has provided guidance and many insights throughout our research. Also, we thank Andri Bezzola for performing the COMSOL simulations that we used in Section 4.1.1. Finally, we thank NSF for the providing the grant which has funded this work.

References

[1] Xiaohai Li, Simulation of Electrochemical Surface Roughness Evolution in Moving Boundary Systems, M.S. Thesis, University of Illinois, Urbana-Champaign, 2004.
[2] U.M. Ascher, L.R. Petzold, Computer Methods for Ordinary Differential Equations and Differential Algebraic Equations, SIAM Press, Philadelphia, Pennsylvania, USA, 1998.
[3] K.E. Brenan, S.L. Campbell, L.R. Petzold, Numerical Solution of Initial-Value Problems in Differential Algebraic Equations, Elsevier Science Publishing Co., Inc., New York, New York, USA, 1989.
[4] M. Georgiadou, D. Veyret, R.L. Sani, R.C. Alkire, Simulation of shape evolution during electrodeposition of copper in the presence of additive, J. Electrochem. Soc. 148 (2001) C54–C58.
[5] John Newman, Karen E. Thomas-Alyea, Electrochemical Systems, John Wiley and Sons, Inc., Hoboken, New Jersey, 2004.
[6] J.A. Sethian, Level Set Methods, Cambridge University Press, New York, 1996.
[7] J.A. Sethian, Level Set Methods and Fast Marching Methods, Cambridge University Press, New York, 1999.
[8] J.A. Sethian, D. Adalsteinsson, An overview of level set methods for etching deposition and lithography development, IEEE Trans. Semiconduct. Manuf. 10 (1) (1997) 167–184.
[9] S. Osher, R.P. Fedkiw, Level set methods: an overview and some recent results, J. Comput. Phys. 169 (2) (2001) 463–502.
[10] S. Mauch, A Fast Algorithm for Computing the Closest Point and Distance Transform, Tech. Rept. Caltech, 2000.
[11] Y. Saad, SPARSKIT: A Basic Tool-kit for Sparse Matrix Computations, 2005, software documentation available at <http://www-users.cs.umn.edu/saad/software/SPARSKIT/sparskit.html>.
[12] Xiaohai Li, Multiscale Simulation and Optimization of Copper Electrodeposition, Ph.D. Thesis, University of Illinois, Urbana-Champaign, 2007.
[13] M. Buoni, L. Petzold, An efficient scalable numerical algorithm for the simulation of electrochemical systems on irregular domains, J. Comput. Phys. 225 (2) (2007) 2320–2332.
[14] M. Buoni, Multiscale Modeling and Simulation of Copper Electrodeposition, Ph.D. Thesis, University of California, Santa Barbara, 2008.
[15] T.P. Moffat, D. Wheeler, W.H. Huber, D. Josell, Superconformal electrodeposition of copper, Electrochem. Solid-State Lett. 4 (2001) C26.
[16] P. Cummings, X. Feng, Domain decomposition methods for a system of coupled acoustic and elastic Helmholtz equations, in: Eleventh International Conference on Domain Decomposition Methods, Domain Decomposition Press, Bergen, Norway, 1999, pp. 203–210.
[17] X. Feng, Analysis of finite element methods and domain decomposition algorithms for a fluid–solid interaction problem, SIAM J. Numer. Anal. 38 (4) (2000) 1312–1336.
[18] Jan Mandel, An iterative substructuring method for coupled fluid solid acoustic problems, J. Comput. Phys. 177 (2002) 95–116.
[19] Andrea Toselli, Olof Widlund, Domain Decomposition Methods – Algorithms and Theory, Springer Publishing, Verlag, Berlin, Heidelberg, Germany, 2005.
[20] P.M. Vereecken, R.A. Binstead, H. Deligianni, P.C. Andricacos, IBM J. Res. Dev. 49 (3) (2005).
[21] Xiaohai Li, Timothy O. Drews, Effendi Rusli, Feng Xue, Yuan He, Richard Braatz, Richard Alkire, Effect of additives on shape evolution during electrodeposition I. Multiscale simulation with dynamically coupled kinetic Monte Carlo and moving-boundry finite-volume codes, J. Electrochem. Soc. 154 (4) (2007) D230–D240.
[22] Effendi Rusli, Feng Xue, Timothy O. Drews, Philippe M. Vereecken, Panos Andricacos, Hariklia Deligianni, Richard D. Braatz, Richard C. Alkire, Effect of additives on shape evolution during electrodeposition II. Parameter estimation from roughness evolution experiments, J. Electrochem. Soc. 154 (11) (2007) D584–D597.
[23] Yan Qin, Xiaohai Li, Feng Xue, Philippe M. Vereecken, Panayotis Andricacos, Hariklia Deligianni, Richard D. Braatz, Richard C. Alkire, Effect of additives on shape evolution during electrodeposition III. Trench infill for on-chip interconnects, J. Electrochem. Soc. 155 (3) (2007) D223–D233.
[24] D. Josell, D. Wheeler, W.H. Huber, T.P. Moffat, Superconformal electrodeposition in submicron features, Phys. Rev. Lett. 87 (1) (2001).
[25] T.P. Moffat, D. Wheeler, W.H. Huber, D. Josell, Superconformal electrodeposition of copper, Electrochem. Solid-State Lett. 4 (4) (2001) C26–C29.
[26] D. Wheeler, D. Josell, T.P. Moffat, Modeling superconformal electrodeposition using the level set method, J. Electrochem. Soc. 150 (5) (2003) C302–C310.
[27] T.P. Moffat, D. Wheeler, D. Josell, Electrodeposition of Cu in the PEG–Cl–SPS additive system I. Kinetic measurements: influence of SPS, J. Electrochem. Soc. 151 (4) (2004) C262–C271.
[28] J.A. Sethian, Solving partial differential equations on irregular domains with moving interfaces, with applications to superconformal electrodeposition in semiconductor manufacturing, J. Comput. Phys. 227 (2008) 6411–6447.
[29] Ashley J. James, John Lowengrub, A surfactant-conserving volume-of-fluid method for interfacial flows with insoluble surfactant, J. Comput. Phys. 201 (2004) 685–722.