Journal of Computational Physics 507 (2024) 112960

Contents lists available at ScienceDirect

Journal of Computational Physics

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

A hybrid finite difference level set–implicit mesh discontinuous Galerkin method for multi-layer coating flows

Luke P. Corcos [a,b], Robert I. Saye [b,*], James A. Sethian [a,b]

[a] Department of Mathematics, University of California, Berkeley, Berkeley, CA 94720, United States of America
[b] Mathematics Group, Lawrence Berkeley National Laboratory, Berkeley, CA 94720, United States of America

ARTICLE INFO

Keywords:
Coating flows
Level set methods
Discontinuous Galerkin methods
Multi-physics
Evaporation
Marangoni dynamics

ABSTRACT

A mathematical model and numerical framework are presented for computing multi-physics multi-layer coating flow dynamics, with applications to the leveling of multi-layer paint films. The algorithm combines finite difference level set methods and high-order accurate sharp-interface implicit mesh discontinuous Galerkin methods to capture a complex set of multi-physics, incorporating Marangoni-driven multi-phase interfacial flow and the transport, mixing, and evaporation of multiple dissolved species. In particular, we develop several numerical methods for this multi-physics problem, including: high-order local discontinuous Galerkin methods for Poisson problems with Robin boundary conditions on implicitly-defined domains, to capture solvent evaporation; finite difference surface gradient methods, to robustly and accurately incorporate Marangoni stresses; and a coupled multi-physics time stepping approach, to incorporate all the different solvers at play including quasi-Newtonian fluid flow. The framework is applicable to an arbitrary number of layers and dissolved species; here, we apply it in a variety of settings, including multi-solvent evaporative paint dynamics, the flow and leveling of multi-layer automobile paint coatings in both 2D and 3D, and an examination of interfacial turbulence within a multi-layer matter cascade. Our results reproduce several phenomena observed in experiment, such as the formation of Marangoni plumes and Bénard cells. We also use the model to study the impact of long-wave deformational surface modes on immersed interfaces as well as the emergence of the final multi-layer film profile.

1. Introduction

Thin liquid films covering a solid surface are ubiquitous in nature and industrial settings, ranging from nanofluidics and the macroscale flow of lava to lacquer spin coatings for compact discs and industrial paint lines. Understanding the coupled fluid flow and settling of multiple liquid film-layers is of particular interest for the control and design of modern coating systems. For example, automotive coating operations use 10,000s of gigawatt-hours each year in the U.S. alone [1]; manufacturing lines that co-cure (i.e., simultaneously bake) multiple layers of paint have the potential to reduce this energy consumption by 30% [2].

In this paper, we present a high-fidelity mathematical and numerical framework to model the complex physics driving fluid flow and leveling in a multi-layer coating flow system. This system includes a range of multi-physics, including multi-phase quasi-Newtonian fluid dynamics, mass transfer and interface recession from solvent evaporation, strong destabilizing surface forces such as

* Corresponding author.
E-mail address: rsaye@lbl.gov (R.I. Saye).

https://doi.org/10.1016/j.jcp.2024.112960
Received 31 August 2023; Received in revised form 20 March 2024; Accepted 21 March 2024

Available online 27 March 2024

Marangoni stresses, and intricate couplings between paint films. The numerical methods developed for this problem include hybrid finite difference level set methods and implicit mesh discontinuous Galerkin (DG) methods, newly developed local discontinuous Galerkin (LDG) solvers for Poisson problems with Robin boundary conditions on implicitly-defined domains, and finite difference surface gradient schemes for Marangoni stress calculations. Many components of this framework are high-order accurate, while other components, whose dynamics do not require high-order methods, benefit from a simpler lower-order implementation. Our particular implementation choices lead to a fully coupled numerical algorithm for the multi-layer coating flow problem that is 2^{nd} order accurate in space and 1^{st} order in time. Results are presented in both 2D and 3D at industrially-relevant conditions, motivated in part by automotive paint coating applications. This model and the developed numerical framework provide opportunities to develop new coating formulas and identify specific features critical to achieving a smooth paint surface. Full details on the mathematical model and developed numerical methods, as well as the results of a parametric study on paint film surface evolution, can be found in the dissertation of Corcos [3].

1.1. Physical background and Marangoni instabilities

A multi-layer coating consists of several layers of liquid paint covering a solid surface. The motion of the paint films is intricately coupled and strong interfacial forces along the paint-paint and paint-gas surfaces drive fluid flow, during which solvents mix between paint layers and evaporate along the paint-gas interface. Two of the key driving forces within the multi-layer coating flow problem are mass transfer from evaporation and Marangoni forces, the latter being tangential surface forces brought on by surface tension gradients, most commonly caused by variations in surface temperature or species concentration. First reported by James Thomson while studying tears of wine [4], and later attributed to Carlo Marangoni [5], Marangoni forces are powerful interfacial phenomena prominent in many heat and mass transfer processes, including crystal growth [6,7], inkjet printing [8], the motion of bubbles [9], and thin film and paint coatings [10–13].

The Marangoni effect was further studied by Bénard in the context of thermally driven flows heated from below [14], notably capturing the emergence of hexagonal-shaped circulation cells within the fluid. Later, the experiments of Block [15] and the mathematical analysis of Pearson [16] demonstrated that the flow patterns arise due to gradients in surface tension caused by variations in surface temperature. Pearson's analysis shows a fundamental stationary hydrodynamic instability in thermal Marangoni flows that produces cellular convection when the system's dimensionless Marangoni number exceeds a critical threshold. This became known as Bénard-Marangoni (or thermocapillary) convection. These are short-wave Marangoni instability modes that, for thin films, have a wavelength on the order of the film thickness [11]. Scriven and Sternling provided analysis for the case of these instabilities arising from a species concentration-dependent surface tension [17] and attributed the Marangoni effect as a mechanism for the spontaneous agitation of an interface between two fluids undergoing mass transfer, known as "interfacial turbulence" [18]. Scriven and Sternling showed the various regimes of stability for different rheological parameters, such as viscosity and mass diffusivity, and for the direction of mass transfer, with some systems being unstable in one direction of mass transfer and stable in the reverse direction. These short-wave Marangoni instabilities are highlighted in the experimental results of Sherwood and Wei [19], where the transfer of hydrochloric acid from a solution of isobutyl alcohol across an interface into water containing ammonia produces spontaneous emulsification of the two solutions and droplets develop without the presence of any chemical reactions. Here the Marangoni effect also accelerates the rate of mass transfer when compared to pure diffusion, an effect first observed by Langmuir and Langmuir [20].

A second type of Marangoni hydrodynamic instability is the long-wave instability described by Scriven and Sternling in [21]. While Pearson's analysis in [16] assumes a flat free surface, Scriven and Sternling found that the presence of surface deformations permits oscillatory long-wave instabilities for which there is no critical Marangoni number, the strength of which is dependent on the mean value of surface tension. These modes were examined experimentally and numerically by van Hook in the context of thermal Marangoni forces [22] and were found to induce significant deformation of the free surface, resulting in areas of local depression and elevation. These long-wave oscillatory deformational instabilities also exist in evaporating flows; the oscillatory nature of paint drying is described by Overdiep in [23] and is studied numerically in the context of lubrication theory in [24,25].

The presence of both instability types is well established in the drying of thin films of paint [10,11,26–29], affecting the final paint smoothness and potentially leading to film defects, holes, tears, and surface corrugations. For reviews on the thermal Marangoni effect, see, e.g., [7,30,31], and for reviews of the Marangoni effect specific to thin films, see, e.g., [12,13,32].

1.2. Previous work

A large body of work has been performed to numerically model evaporating flows, as well as solve the associated Robin boundary problem along a moving interface. These include arbitrary Euler-Lagrange (ALE) finite element methods [33–35], where the computational mesh is deformed to align with the interface and the Robin boundary condition is naturally handled by the finite element method's weak formulation. Finite volume methods combined with level set methods [36,37] and ghost fluid methods [38] have been applied to vaporizing two-phase flows [39,40]. In this setting, boundary conditions are applied along an implicitly-defined interface via extrapolation onto fictitious cells or nodes, with the cut-cell Robin solver of Papac [41] being one of the first methods for applying Robin boundary conditions within the level set framework. These methods were later extended to solve for Robin boundary conditions on piecewise-smooth interfaces [42]. Ghost-cell finite difference methods [43–45] have recently been developed and applied to similar flow problems. Here linear PDE extrapolation onto ghost nodes is combined with a cell-wise calculation of the normal derivative to impose Robin boundary conditions along interfaces implicitly defined by the level set method.

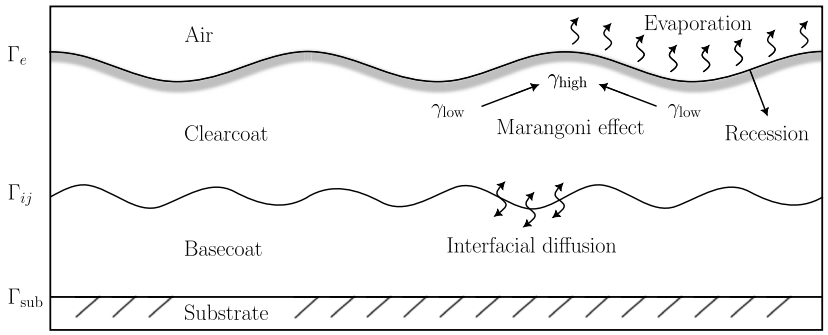


Fig. 1. The two-layer coating flow problem.

In recent years, a number of numerical studies on the Marangoni effect have been performed, particularly in the context of thermocapillary convection. Interface tracking finite element methods have widely been used to model the Marangoni effect within evaporating micro-droplets [46–48], with these and other boundary-fitting methods [49] allowing for accurate implementation of the Marangoni effect along surfaces with limited deformation. An alternative approach is to capture the interface implicitly in a fully-Eulerian framework. For example, a common technique used in the volume of fluid (VOF) [50,51] and level set frameworks [52–54] is to apply a diffuse-interface approach, whereby interfacial jump conditions are replaced by locally-smoothed Dirac delta forcing terms added to the right-hand-side of the Navier-Stokes momentum equations. This grid-dependent smoothing often impacts numerical resolution power, particularly in regard to accurately handling thin solutal boundary layer dynamics present within evaporating Marangoni flows. Several additional numerical methods for Marangoni-driven flows have been recently presented, including ghost fluid level set methods applied to soluble surfactant-driven flows [55], semi-Lagrangian methods for soap film dynamics [56], and material point methods for liquid metal droplet dynamics [57]. Another notable work is that of Köllner et al. [58], which examines the behavior of short-wave solutal Marangoni instabilities and the onset of interfacial turbulence using pseudo-spectral methods along a fixed flat interface. Their work, in collaboration with experiments [59], classifies the various complex patterns arising from short-wave Marangoni instabilities into different hierarchical spatial structures, including associated merging/coarsening processes. These results were replicated in the work of Yiantsios [35], in which an ALE finite element method captures the formation and merger of solutal Marangoni plumes/roll cells during the drying of polymer solutions.

1.3. Outline

The outline of this paper is as follows: in section 2, the equations of motion, boundary, and jump conditions for the multi-layer coating flow problem are introduced. In section 3, the developed numerical methods are presented, including: hybrid finite difference level set–implicit mesh discontinuous Galerkin methods; algorithms for the evaporative mass transfer system and associated LDG methods for Poisson problems with Robin boundary conditions; algorithms for quasi-Newtonian fluid dynamics and associated multi-phase Stokes solvers; finite difference surface gradient algorithms for Marangoni stress calculations; and, lastly, the fully coupled numerical algorithm for the multi-layer coating flow problem. The results of several convergence studies are shown in section 4 and the results of our numerical studies on multi-layer coatings are presented in section 5. We conclude our discussion in section 6.

2. Equations of motion

2.1. Domain geometry

In the setting of multi-layer coating flow, one or more film-layers of basecoat paint are applied onto a substrate, followed by a final layer of protective clearcoat. Our focus in this work is the case when all film-layers are cured simultaneously. The multi-layer coating flow problem therefore has a domain of interest consisting of four subdomains—the substrate, the basecoats, the clearcoat, and the air—separated by the following interfaces:

Γ_{sub}	The top of the substrate.
Γ_{ij}	The middle embedded paint-paint surfaces separating the coats of paint.
Γ_e	The evaporative surface between the clearcoat and the air.

In this model, the dynamics of the air are assumed to have a negligible effect and the computational domain is restricted to the layers of liquid paint. A free-surface model is used for the evaporative surface Γ_e and we define the d-dimensional multi-phase liquid domain $\Omega \subset \mathbb{R}^d$ as $\Omega = \bigcup_i \Omega_i$, where Ω_i represents the region of paint layer i, with $\Gamma_{ij} = \Omega_i \bigcap \Omega_j$. The domain is taken to be periodic along all horizontal dimensions (x in 2D; x, y in 3D) and we make the simplifying assumption that the interfaces never cross. Together these two assumptions alleviate the numerical difficulties of contact line and triple point dynamics; the extension of this framework to more general situations is discussed in the concluding remarks. An example domain for the case of two paint layers is shown in Fig. 1.

The coating flow problem is in the micro-flow regime, with the initial film thicknesses of the basecoats ranging from $30-50$ μm and that of the clearcoat ranging from $50-100$ μm, while typical "roughness" wavelengths along the surface of a dried paint film range from $1-10$ mm.

2.2. Governing equations

The model for the multi-layer coating flow problem must incorporate numerous physical effects, including quasi-Newtonian fluid dynamics; transport, diffusion, and mixing of multiple dissolved species; mass transfer and interface recession from solvent evaporation; a constantly-evolving coating rheology; intricate interfacial forces of surface tension and Marangoni stresses on paint-gas and paint-paint interfaces and their coupling; and substrate roughness and the pull of gravity. The model developed in this paper is purely isothermal, with the Marangoni effect driven by variations in species concentration.

The paints are composed of multiple solvents dissolving an underlying resin (also known as the solute or solid). The effect of buoyancy is small in the microflow regime and, therefore, a constant paint density is assumed in this model. The fluids are considered quasi-Newtonian, governed by the incompressible Navier-Stokes equations with the viscosity varying with respect to resin concentration:

$$\begin{aligned}(\rho \boldsymbol{u})_t + \nabla \cdot (\rho \boldsymbol{u}\boldsymbol{u}) &= -\nabla p + \nabla \cdot \left(\mu(c_R)(\nabla \boldsymbol{u} + \nabla \boldsymbol{u}^T)\right) + \rho \boldsymbol{g}, \\ \nabla \cdot \boldsymbol{u} &= 0,\end{aligned} \tag{1}$$

where $\boldsymbol{u}$ is the velocity field, p is the pressure, ρ is the density of the fluid, μ is the dynamic viscosity (a scalar function of the resin mass concentration c_R), and $\boldsymbol{g}$ is the gravity vector. As the paints dry, the solvents evaporate into the air, thereby causing the liquid volume to shrink, the liquid-gas interface to recess, and eventually the paint to solidify. When the paint is fully dry, only the resin remains and any imprints or irregularities caused by, or occurring during, the drying process will become permanent. The solidification process is captured by the exponential nature of the viscosity profile, described later.

The motion of the dissolving solvents within the system is governed by the convection-diffusion equations

$$\begin{aligned}(c_k)_t + \nabla \cdot (c_k \boldsymbol{u}) &= \nabla \cdot (D_k \nabla c_k), \\ c_R &= 1 - \sum_k c_k,\end{aligned} \tag{2}$$

where c_k is the mass concentration and D_k is the mass diffusion coefficient for solvent k. Our framework is capable of handling multiple solvents, which may differ in both their mass diffusion coefficients as well as their evaporation rates. For simplicity, we assume that each solvent's diffusion coefficient is constant within each paint layer, but we allow for different values in different phases. Note that the equations of motion demonstrate a direct coupling between the motion of the solvents and the fluid dynamics within the bulk of the paint.

2.2.1. Boundary and jump conditions

To ensure conservation of mass, momentum, species, and volume within the multi-layer coating flow problem, we must examine the boundary and jump conditions arising at a moving interface. Briefly, a conservation law of the form (3) will have a jump condition of Rankine-Hugoniot type (4) across a moving surface:

$$a_t + \nabla \cdot (a\boldsymbol{u}) + \nabla \cdot \boldsymbol{J} = f, \tag{3}$$

$$[a(\boldsymbol{u} \cdot \boldsymbol{n} - V) + \boldsymbol{J} \cdot \boldsymbol{n}] = f_S, \tag{4}$$

where a is a general quantity with flux $\boldsymbol{J}(a, \nabla a)$, V is the interface velocity in the normal direction $\boldsymbol{n}$, f and f_S are the volumetric and surface sources respectively, and $[\cdot]$ represents the jump across the surface in the direction of the normal vector. Specifically, $[a] = a^- - a^+$, where a^- and a^+ are the values of a restricted to the surface's "bottom" and "top" phases, respectively, with the normal vector pointing from the bottom layer into the top.

Applying (4) to the conservation of mass and the continuity equation of the Navier-Stokes equations (i.e., $\rho_t + \nabla \cdot (\rho \boldsymbol{u}) = 0$, which reduces to the incompressibility constraint $\nabla \cdot \boldsymbol{u} = 0$ in this model) dictates that $[\rho(\boldsymbol{u} \cdot \boldsymbol{n} - V)] = 0$ along a moving surface and implies that the mass flux across the interface is given by

$$m = \rho(\boldsymbol{u} \cdot \boldsymbol{n} - V), \tag{5}$$

where m is the total mass flux; here, the values of ρ and $\boldsymbol{u}$ may be determined by their restriction to either phase. Rearranging (5) provides an expression for the interface velocity that includes the fluid velocity and the motion induced by mass transfer:

$$V = \boldsymbol{u} \cdot \boldsymbol{n} - m/\rho. \tag{6}$$

As solvents leave the domain at the free evaporative surface Γ_e, the effect of (6) is to cause a recession of the interface, thereby reducing the bulk liquid volume while also ensuring conservation of mass. Equation (6) is also applicable to the motion of the embedded paint-paint surfaces Γ_{ij}, where mass transfer between two paints causes one film-layer to shrink while the other swells.

Applying (4) to the solvent convection-diffusion equations (2) and assuming that the system is free of chemical reactions[1] yields the following expression for conservation of species across a general moving interface:

$$m[c_k] - [\rho D_k \nabla c_k \cdot \boldsymbol{n}] = 0. \tag{7}$$

Expanding for each phase gives an equivalent Robin boundary condition for the solvent mass concentration c_k:

$$m_k = m c_k - \rho D_k \nabla c_k \cdot \boldsymbol{n}, \tag{8}$$

where m_k is the mass flux for solvent k and $m = \sum_k m_k$. In the multi-layer coating flow problem, the solvent mass loss from evaporation is captured by (8) along the free evaporative surface Γ_e. The specific value for evaporative mass flux m_k—which is dependent on the solvent mass concentration at the interface—is explained in more detail in section 3.3. Jump condition (7) also holds along the embedded paint-paint surfaces Γ_{ij}, however here we simplify[2]by enforcing continuity of the solvent mass concentration profile, setting $[c_k] = 0$ and reducing (7) to the standard jump condition $[D_k \nabla c_k \cdot \boldsymbol{n}] = 0$. Summing (8) for each solvent gives the following expression for the total mass flux between paints at the embedded surfaces Γ_{ij}:

$$m = -\frac{\rho}{c_R} \sum_k D_k \nabla c_k \cdot \boldsymbol{n}. \tag{9}$$

Using (9) in the interface velocity equation (6) ensures that the resins of the basecoat and clearcoat paints do not mix and that conservation of mass, volume, and species is respected between paint layers. The solvent convection-diffusion system (2) is closed by applying the no-penetration condition $\nabla c_k \cdot \boldsymbol{n} = 0$ along the substrate Γ_{sub}.

In general, when mass transfers across a moving interface and the densities of the two phases differ, a discontinuity in the fluid's normal velocity occurs at the interface as a consequence of Stefan flow. However, when the densities of the two phases are equal, the fluids stick together and there is no jump in the velocity field. Our assumption of a constant paint density sets the no slip-condition $[\boldsymbol{u}] = 0$ on embedded paint-paint surfaces Γ_{ij} while the conservation condition (4) for the Navier-Stokes momentum equation (1) produces the following jump in stress

$$[\boldsymbol{\sigma} \cdot \boldsymbol{n}] = -\gamma \kappa \boldsymbol{n} + \nabla_S \gamma, \tag{10}$$

where $\boldsymbol{\sigma} = -p\mathbb{I} + \mu(\nabla \boldsymbol{u} + \nabla \boldsymbol{u}^T)$ is the stress-tensor, γ is the coefficient of surface tension, κ is the mean curvature of the interface, and $\nabla_S = (\mathbb{I} - \boldsymbol{n} \otimes \boldsymbol{n})\nabla$ is the surface gradient operator. The two terms on the right-hand side of (10) represent the forces of surface tension and Marangoni stresses respectively. As motivated in the introduction, surface tension gradients, or Marangoni forces, are powerful tangential forces that may produce short-wave and long-wave hydrodynamic instabilities within the system. In the multi-layer coating flow problem, Marangoni forces are caused by species concentration variations along the surface, wherein the surface tension coefficient is a function of resin mass concentration, i.e., $\gamma = \gamma(c_R)$. Therefore, by the chain rule, Marangoni forces are of the form $\nabla_S \gamma = \frac{d\gamma}{dc_R} \nabla_S c_R$. Along the free evaporative surface Γ_e, the stress from the gas phase is negligible and (10) reduces to the following stress boundary condition

$$\boldsymbol{\sigma} \cdot \boldsymbol{n} = -p_{\text{ext}} \boldsymbol{n} + \left[\frac{1}{\rho}\right] m^2 \boldsymbol{n} - \gamma \kappa \boldsymbol{n} + \nabla_S \gamma, \tag{11}$$

where p_{ext} is an external pressure, here set to atmospheric. The second term on the right-hand side of (11) represents the stress caused by evaporation, which is negligible in the micro-flow regime and only included for presentation. Lastly, we apply the no-slip condition $\boldsymbol{u} = 0$ on the substrate.

3. Numerical methods

To solve this set of coupled multi-physics equations, we have developed a hybrid numerical framework consisting of finite difference level set methods [36,37,61,62] and high-order accurate multi-phase implicit mesh discontinuous Galerkin methods [63–65,1]. These methods use an implicit level set representation of the paint surfaces combined with a structured background quadtree or octree to create a collection of interface-conforming elements on which high-order accurate DG methods are applied. The methods sharply capture evolving interface dynamics to high-order accuracy in a dimension-independent fashion. In the present setting, it is especially important to capture spatial characteristics with sufficiently high-order accuracy, such as thin boundary layers; on the other hand, the temporal accuracy is less important. Consequently, and for ease of implementation and presentation, we describe a simple 1^{st} order mixed explicit-implicit time stepping method, where advective terms are treated explicitly and the diffusive and viscous terms are solved implicitly via backward Euler.

This section begins by first outlining our hybrid numerical framework, including the notion of implicitly-defined meshes and their evolution, the definition of the DG polynomial spaces, and a discussion of numerical quadrature schemes for implicitly-defined

[1] In the automotive paint applications motivating this work, inter-species chemical reactions are assumed to have negligible effect. In general, drying paint may involve volumetric or surface chemical reaction processes [60]; the latter could be incorporated via more complex Robin boundary conditions.

[2] A closure condition specifying the jump $[c_k]$ is required for the Robin-style jump condition (7). The specific value of this jump is, in general, unknown for multi-layer coatings and is set to zero in this work.

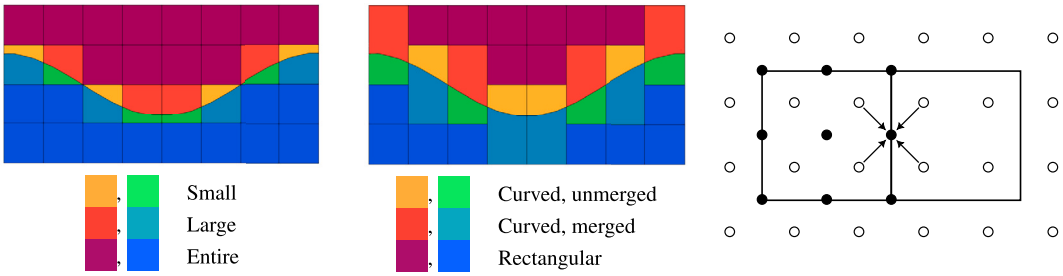


Fig. 2. The construction of an implicit mesh for a two-phase (red and blue) domain in 2D with an embedded surface. **Left:** The zero isocontour of the level set function cuts through the cells of the background quadtree, resulting in phase-cells that are classified as empty, small, large, or entire. **Center:** The elements of the implicitly-defined mesh after cell merging. The mesh consists mostly of standard rectangular elements, along with a collection of curved interface-conforming elements, which may or may not be merged and extend outside of their parent cell. **Right:** The finite difference nodes ○ located within the background DG cells. The level set function ϕ is interpolated from the finite difference nodes onto the cells' Gauss-Lobatto nodes • and used in the generation of numerical quadrature schemes for implicitly-defined elements and surfaces [68]. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

elements and surfaces; then a new LDG formulation for Poisson problems with Robin boundary conditions on implicitly-defined domains is presented in the context of evaporating flows; next the methods for solving quasi-Newtonian fluid flow are discussed, including the application of fast multigrid Stokes solvers [66] and a finite difference Marangoni calculation algorithm; lastly, the coupled algorithm for the multi-layer coating flow problem is presented.

3.1. Hybrid finite difference level set and implicit mesh discontinuous Galerkin methods

In typical flow leveling problems, there is no self-folding of paint layers nor situations in which the paints develop significant profile steepness. For this reason, the surfaces within the multi-layer coating flow problem $(\Gamma_{\text{sub}}, \Gamma_{ij}, \Gamma_e)$ are each represented by a height function $h : \mathbb{R}^{d-1} \times [0,T) \to \mathbb{R}$. The height function describing the substrate is held fixed and those capturing paint-paint and paint-gas surfaces evolve under the advection equation

$$h_t + \boldsymbol{v} \cdot \nabla h - v\hat{e}_d = 0, \tag{12}$$

where $\boldsymbol{v}$ and v are the horizontal and vertical components of the interface velocity vector, incorporating the motion of the fluid and the motion induced by mass transfer (6). Here $\hat{e}_d$ represents the vertical dimension, while along the horizontal dimensions, ∇h is computed by second-order ENO schemes [67]. The temporal evolution of the height functions is captured via forward Euler. We note that if the surfaces $(\Gamma_{\text{sub}}, \Gamma_{ij}, \Gamma_e)$ should intersect, the simulation is halted; further comments on this aspect are discussed in the concluding remarks.

Within the hybrid framework, a level set function $\phi : \mathbb{R}^d \times [0,T) \to \mathbb{R}$ is constructed from the height functions and used to generate numerical quadrature schemes for the implicit mesh DG methods [68,64]. The zero isosurfaces of ϕ align with the location of the height functions and interfaces, and the level set function together with a phase indicator defines the multi-phase liquid domain Ω. The level set function ϕ is defined at finite difference nodes located within the DG cells of the background quad/octree, as illustrated in Fig. 2 (right); in particular, we employ uniformly spaced DG cells and finite difference nodes, with each cell containing two finite difference nodes per dimension.[3] To couple the finite difference methods to the implicit mesh DG formulation, a piecewise polynomial representation of ϕ is constructed; specifically, the values of the level set function are bi-linearly or tri-linearly interpolated onto the Gauss-Lobatto nodes of the background DG cells, from which a bi-quadratic or tri-quadratic polynomial on each cell is naturally defined (see 3.1.3). This process is highlighted in Fig. 2 (right). The level set polynomials are then used in the algorithms of [68] to generate the quadrature schemes. We note that this specification of the level set polynomials ensures interface continuity between the DG cells.

Additionally, standard 2^{nd} order finite difference methods are used to compute values of interfacial curvature $\kappa = \nabla \cdot (\nabla\phi/|\nabla\phi|)$ for surface tension calculations [37] and the finite difference methods discussed in section 3.4.2 are employed to compute surface gradients for Marangoni stresses. In summary, the finite difference methods within the hybrid numerical framework for the multi-layer coating flow problem provide a mechanism to robustly and accurately calculate the interface evolution as well as interfacial forces, whereas the implicit-mesh DG methods, discussed next, provide for a high-order accurate, sharp-interface method for computing fluid velocity fields, pressure, and solvent mass concentration profiles.

3.1.1. Implicitly-defined meshes

At each time step, the domain geometry is discretized into a collection of elements on which the DG methods are applied. An implicitly-defined mesh [64] is constructed from two objects: (i) a background quadtree or octree depending on whether the problem is in 2D or 3D respectively, and (ii) an implicit representation of the interfaces—the zero isocontour of a level set function in this

[3] We found this to strike a good balance between the resolution of the finite difference grid and the piecewise polynomial bi-quadratic/tri-quadratic discontinuous Galerkin methods.

problem—which either defines the domain's boundary, the embedded surfaces within the domain, or both. The zero level set cuts through the cells of the background quad/octree, resulting in a collection of rectangular and cut cells for each phase of the domain, which are denoted as *phase-cells*. Each phase-cell is then classified as empty, small, large, or entire by the volume fraction of that phase within the cell, with small and large classifications demarcated by a user-defined volume fraction threshold, taken to be 40% in this work (see [64] for further discussion on the impact of this threshold). Empty phase-cells are discarded and small phase-cells are merged with neighboring large or entire phase-cells of the same phase (known as the *parent* phase-cell) to avoid the numerical conditioning issues and time step restrictions caused by arbitrarily-small cut cells. In [64], a general cell-merging algorithm is described, but in the present setting we note the geometry allows for a simpler approach: small phase-cells are merged with the large or entire phase-cells directly above or below.

This process defines a collection of mesh elements that are mostly rectangular with a small number of interfacial curved elements that may or may not extend outside of their parent cell. The curved interfacial elements are interface-conforming and the resulting mesh sharply captures the interfaces, allowing for the high-order accurate imposition of boundary and jump conditions and the capturing of thin solutal boundary layers present in evaporating Marangoni flows. Note that the interface is never explicitly constructed as a discretized mesh; instead, the geometry of interfacial elements and corresponding surfaces are determined solely by means of numerical quadrature in the weak formulation of the DG methods. An example of an implicit mesh and its cell merging procedure for a two-phase domain with an embedded surface is demonstrated in Fig. 2.

The faces of the implicitly-defined meshes are classified as follows: (i) *intraphase* faces are shared between elements of the same phase, the collection of which is denoted by Γ_0. These faces are always flat and the normal vector $\boldsymbol{n}$ is defined to point from "left" to "right". (ii) *Interphase* faces lie between elements of differing phases and are situated on Γ_{ij}, whose normal vector points in the direction defined by the level set function. These are the faces on which interfacial jump conditions are applied. (iii) Lastly, *boundary* faces are situated on the free evaporative surface Γ_e and the substrate Γ_{sub}, with the normal vectors taken to be outwards pointing.

3.1.2. Numerical quadrature rules

The computation of volume and surface integrals arising from the variational forms of the DG methods requires appropriate numerical quadrature rules for the mesh elements and faces. Integrals along rectangular elements and faces use tensor-product Gauss-Legendre quadrature rules; on the other hand, the curved interfacial elements, implicit interfaces, and cell faces cut by the zero level set use the high-order quadrature algorithms of [68]; an open-source C++ implementation of these algorithms is available at [69].

3.1.3. Discontinuous Galerkin methods

Owing to the use of a background quadtree/octree grid, it is natural to adopt a tensor-product piecewise polynomial space for the DG methods, including for the fluid velocity $\boldsymbol{u}$, pressure p, and solvent mass concentration c_k. Let $\mathcal{E}$ represent the collection of mesh elements and define $\mathcal{Q}_p(E)$ to be the space of d-dimensional tensor-product polynomials of degree $p \geq 1$ on element $E \in \mathcal{E}$. Now define the discontinuous piecewise polynomial space V_h such that

$$V_h = \{u : \Omega \to \mathbb{R} \mid u|_E \in \mathcal{Q}_p(E) \text{ for every } E \in \mathcal{E}\},$$

with analogous definitions for the space of piecewise polynomial vectors (V_h^d) and rank-2 tensors ($V_h^{d\times d}$). In our particular implementation of these methods, we are free to choose the polynomial degree; for the kinds of dynamics studied in our results, we found degree $p = 2$ (i.e., bi-quadratic and tri-quadratic polynomial spaces in 2D and 3D, resp.) provided a suitable balance between speed and high-order accuracy. This work employs a nodal basis and the values of polynomials in V_h are defined on tensor-product Gauss-Lobatto nodes located within the parent cells of the background quad/octree.

3.1.4. Temporal evolution and state transfer

We now briefly discuss the notion of temporal evolution within the implicit mesh DG framework—with the end goal of modeling time-dependent interface problems. At every time step $n = 0, 1, \ldots$, the level set function ϕ^n (defined in this work via the height functions) combined with the background quad/octree generates an implicitly-defined mesh as well as its elemental and surface quadrature rules. The DG polynomial space V_h^n (as well as the space of vectors and rank-2 tensors) and its corresponding nodal basis are defined specifically for the corresponding time step's implicit mesh. The polynomial space then generates the LDG operators specific to each mesh and interface configuration. As the level set function—the interfaces—evolves to the next time step ϕ^{n+1}, the previously defined implicit mesh no longer aligns with the interfaces, meaning that numerical quadrature rules, DG polynomial spaces, and LDG operators must be recreated for the new interface locations. State variables that were defined in the polynomial space of the previous mesh are then transferred onto the polynomial space of a new implicit mesh that captures the updated interface locations, i.e., a state variable in V_h^n is transferred to one appropriate in the new space V_h^{n+1}. In the general setting, this can be done via a general kind of L^2 projection. In fact, because the interface usually moves only a small fraction of the mesh spacing, it is possible to create time-evolving implicit meshes which, for the most part, use the same cell-merging decisions as prior time steps. Using this approach, it is often the case that a one-to-one correspondence exists between the elements of one mesh and the next, which simplifies the state transfer operation to a very simple injection procedure. Subsequently, the transference of state variables has a negligible impact on the performance of the overall multi-layer coating flow algorithm. For more details on this approach, and time-stepping implicit mesh DG methods in general, see [64].

3.2. Algorithm outline

With the preliminaries of the hybrid numerical framework introduced, we now outline the basic structure of our numerical algorithm for the multi-layer coating flow problem. The developed numerical methods for approximating the multi-phase incompressible Navier-Stokes equations (1) and the convection-diffusion equations (2) treat advective terms explicitly-in-time, with numerical fluxes given by standard upwinding. The viscous and diffusive terms are solved implicitly-in-time via backward Euler, requiring solutions to Stokes and heat operator[4] problems, respectively. The algorithm for the multi-layer coating flow problem proceeds as follows:

1. At time step 0, initialize the height functions Γ_{sub}, Γ_{ij}, and Γ_e to create the initial interface configuration, construct the associated implicitly-defined mesh, DG polynomial spaces, and LDG operators; and set the initial DG state variables for the solvent mass concentrations c_k, paint velocity field $\boldsymbol{u}$, and pressure p.
2. Begin time stepping: For $n = 0, 1, 2, \ldots$

   (i) Compute the concentration and velocity field advection terms via an upwinding scheme.
   (ii) Evolve the height functions under (12) to find ϕ^{n+1}. Create a new implicit mesh, transfer state variables onto the new mesh's DG polynomial spaces, and create LDG operators.
   (iii) Update the solvent mass concentrations c_k^{n+1} by solving the mass transfer convection-diffusion equations (2), use this to determine the new concentration-dependent rheological parameters.
   (iv) Update the velocity field $\boldsymbol{u}^{n+1}$ and pressure p^{n+1} by solving the incompressible Navier-Stokes equations (1) for quasi-Newtonian fluid dynamics.
   (v) Repeat until the final time is reached.

The numerical methods for the evaporative mass transfer system (step (iii)) and the Marangoni-driven multi-phase quasi-Newtonian fluid dynamics (step (iv)) are presented in the upcoming sections. These methods are coupled together in the full numerical algorithm for the multi-layer coating flow problem, described in section 3.5.

3.3. Mass transfer system

As motivated in section 2.2, the motion of solvent within the multi-layer coating flow problem is described by the convection-diffusion equations (2) coupled to evaporation and mixing at the paint-gas and paint-paint surfaces. Applying the mixed explicit-implicit time stepping scheme to (2) gives the following heat operator system for the mass transfer problem,

$$\frac{c_k^{n+1} - c_k^n}{\Delta t} + \nabla \cdot (c_k \boldsymbol{u})^n = \nabla \cdot \left(D_k \nabla c_k^{n+1} \right), \tag{13}$$

together with the Robin boundary condition $m_k = m c_k^{n+1} - \rho D_k \nabla c_k^{n+1} \cdot \boldsymbol{n}$ along Γ_e, as well as continuity and zero diffusive flux jump conditions along Γ_{ij}. The advective term $\nabla \cdot (c_k \boldsymbol{u})^n$ is treated explicitly via an upwinding scheme and the Robin boundary problem is solved by the LDG schemes discussed next in section 3.3.1 and presented in more detail in Appendix A.

The solvent evaporative mass flux m_k accounts for two aspects: first, the evaporation rate should be proportional to the amount of the solvent at the interface and tend towards zero as the solvent mass concentration goes to zero; second, in multi-solvent cases, the solvent with the largest species concentration is preferential to evaporation. The following solvent evaporation rate m_k takes into account both considerations:

$$m_k = \frac{\varepsilon}{C} \cdot \frac{\left(c_k^n\right)^2}{\sum_j c_j^n}, \tag{14}$$

where ε is the coefficient of evaporation, a tunable, application-defined parameter that can incorporate additional physics if necessary. In all case studies in this work, this coefficient is taken to be a constant whose value is chosen to match experimental data. Note that (14) is normalized by the number of solvents C to ensure equivalent dynamics between the single solvent case and the case where multiple solvents all have equal mass diffusion and evaporation coefficients. Also note that the evaporative process introduces a fully non-linear constraint to the mass transfer system. This nonlinearity is treated in our model by using the solvent mass concentration values from the previous time step (i.e., from the traces of c_k^n) in the definition of the solvent mass flux m_k (14) and total evaporation rate m. The remaining description of the evaporation process is then the linear Robin boundary condition (8).

Additionally, for the inter-paint mixing rate (9), normal derivatives of the solvent mass concentration are computed directly from the DG polynomials c_k^n. The computed evaporation and mixing rates are embedded within the level set speed law (6) and used in the advection equation (12). After the solvent mass transfer system is advanced to the next time step, an updated value of resin mass concentration $c_R^{n+1} = 1 - \sum_k c_k^{n+1}$ is computed to define the concentration-dependent rheological parameters (viscosity and surface tension) for the quasi-Newtonian fluid dynamics.

[4] The heat operator equation involves finding $u : \Omega \to \mathbb{R}$ such that $(\mathbb{I}/\delta - \nabla \cdot (D\nabla))u = f$, where $\mathbb{I}$ is the identity operator, δ is a constant (typically describing a discrete time step), D is the diffusion coefficient, and $f : \Omega \to \mathbb{R}$ is a source function, subject to suitable boundary conditions.

Table 1
The general steps to define the LDG methods for solving the Poisson and Stokes equations. See [64,66] for more details.

Poisson $-\nabla \cdot (\mu \nabla u) = f$	Stokes $-\nabla \cdot (\mu(\nabla \boldsymbol{u} + \nabla \boldsymbol{u}^T)) + \nabla p = \boldsymbol{f}$ $-\nabla \cdot \boldsymbol{u} = f_{\text{div}}$
(i) Introduce the gradient $\boldsymbol{\eta} \in V_h^d$, such that $\boldsymbol{\eta} = \nabla u$ weakly via the strong-weak form.	(i) Introduce the gradient $\boldsymbol{\eta} \in V_h^{d\times d}$, such that $\boldsymbol{\eta} = \nabla \boldsymbol{u}$ weakly via the strong-weak form.
(ii) Define $\boldsymbol{q} \in V_h^d$ as the L^2 projection of $\mu\boldsymbol{\eta}$.	(ii) Define the stress-tensor $\boldsymbol{\sigma} \in V_h^{d\times d}$ as the L^2 projection of $\mu(\boldsymbol{\eta} + \boldsymbol{\eta}^T) - p\mathbb{I}$.
(iii) Compute the divergence $w \in V_h$, such that $w = \nabla \cdot \boldsymbol{q}$ weakly via the weak-weak form.	(iii) Compute the divergence of the stress-tensor $\boldsymbol{w} \in V_h^d$, such that $\boldsymbol{w} = \nabla \cdot \boldsymbol{\sigma}$ weakly via the weak-weak form.
(iv) Require that $-w$ equals the L^2 projection of f, while also incorporating penalty stabilization to enforce continuity, boundary, and jump conditions.	(iv) Enforce the divergence constraint $w \in V_h$, such that $w = \nabla \cdot \boldsymbol{u}$ via the strong-weak form.
	(v) Require that $-(\boldsymbol{w}, w)$ equals the L^2 projection of $(\boldsymbol{f}, f_{\text{div}})$, while also incorporating penalty stabilization to enforce continuity, boundary, and jump conditions.

3.3.1. Poisson problems with Robin boundary conditions

A key driving force in the multi-layer coating flow problem is mass transfer at the free evaporative surface Γ_e, with the evaporation process described by the Robin boundary condition (8). In particular, the mixed explicit-implicit time stepping method leads to a heat operator problem (13) with boundary conditions of Robin type. In turn, this requires the development of tailored local discontinuous Galerkin methods [70,71] specifically targeting Robin boundary conditions. In Appendix A, we present the derivation of such an approach for the Poisson problem.[5] The resulting discretization has several favorable properties: for example, it is optimally high-order accurate and the final linear system which solves for the updated solvent mass concentration profiles c_k^{n+1} is symmetric positive definite; moreover, the LDG discretization is amenable to fast, multigrid-preconditioned conjugate gradient solvers [72–74], even for the long-and-thin domains considered here.

3.4. Quasi-Newtonian fluid solvers

To capture the flow and leveling of the quasi-Newtonian liquid paint films, we wish to solve the multi-phase incompressible Navier-Stokes equations (1) wherein the viscosity varies with respect to resin mass concentration, while also incorporating the various boundary and jump conditions that capture the effects of surface tension, Marangoni forces, and the couplings between film-layers. To do so, we use the LDG Stokes solver of [66] in conjunction with finite difference methods for computing interfacial curvature and surface gradients for the calculations of surface tension and Marangoni stresses respectively.

In more detail, our mixed first-order explicit-implicit time stepping scheme applied to (1) results in the following system of equations for updating the fluid velocity $\boldsymbol{u}^{n+1}$ and pressure p^{n+1}:

$$\rho\left(\frac{\boldsymbol{u}^{n+1}-\boldsymbol{u}^n}{\Delta t}\right) + \nabla\cdot(\rho\boldsymbol{u}\boldsymbol{u})^n = -\nabla p^{n+1} + \nabla\cdot\left(\mu(c_R^{n+1})\left(\nabla\boldsymbol{u}^{n+1} + (\nabla\boldsymbol{u}^{n+1})^T\right)\right) + \rho\boldsymbol{g}, \tag{15}$$
$$\nabla\cdot\boldsymbol{u}^{n+1} = 0,$$

including the stress conditions (10), (11) along the embedded paint-paint surfaces Γ_{ij} and free evaporative surface Γ_e, as well as the no-slip conditions on Γ_{ij} and substrate Γ_{sub}. The advection term $\nabla\cdot(\rho\boldsymbol{u}\boldsymbol{u})^n$ is discretized by a standard DG upwinding scheme and the viscous components via backward Euler. The resulting time-dependent Stokes system is solved by LDG schemes that provide high-order accurate solutions to (15) on multi-phase implicitly-defined domains while seamlessly incorporating the paint layer couplings, varying viscosity profile, and the boundary and jump conditions. This formulation enforces the divergence constraint without an intermediate projection step and, in [66], rapid multigrid performance is achieved when the proper pressure penalty parameter is chosen, with performance matching that of classical geometric multigrid methods applied to Poisson problems. In the remainder of this section, we summarize the LDG methods for solving the time-dependent Stokes system and then present our finite difference surface gradient formulation for Marangoni stress calculations. For full details on the LDG Stokes solver, the reader is referred to [66].

3.4.1. LDG for the Stokes equations

The LDG methods of [66] for the Stokes equations are constructed similarly as that for the Poisson equation [64]. To highlight some of the similarities, the steps outlining the LDG methods for the Poisson and Stokes equations are shown in Table 1. After performing the five steps described in the table, the LDG discretization for the time-dependent Stokes equations (15) results in the following symmetric block-form linear system

$$\begin{pmatrix} A & M\mathcal{G} \\ \mathcal{G}^T M & -E_p \end{pmatrix}\begin{pmatrix} \boldsymbol{u}^{n+1} \\ p^{n+1} \end{pmatrix} = \begin{pmatrix} \boldsymbol{b} \\ b_{\text{div}} \end{pmatrix}, \tag{16}$$

[5] The derivation of these methods is deferred to the appendix so as to not distract from the main objectives of the present section in laying out the general numerical framework.

where M is the mass matrix, $\mathscr{G} = -M^{-1}G^T M$ is a discrete gradient operator,[6] $\mathscr{G}^T$ is a discrete divergence operator that enforces the divergence constraint, E_p is the pressure penalty operator, and A is the $d \times d$ block operator containing the temporal and viscous components of the momentum equation, with

$$A_{ij} = \delta_{ij}\left(\frac{M_\rho}{\Delta t} + \sum_{k=1}^{d} G_k^T M_\mu G_k + E_{\boldsymbol{u}}\right) + G_j^T M_\mu G_i, \tag{17}$$

where δ_{ij} is the Kronecker delta function, M_ρ and M_μ are the ρ and μ-weighted mass matrices,[7] resp., such that $v^T M_\rho u = \int_\Omega v\rho u$ and $v^T M_\mu u = \int_\Omega v\mu u$ for all $u, v \in V_h$, and $E_{\boldsymbol{u}}$ is the velocity penalty operator. The right-hand side $(\boldsymbol{b}, b_{\text{div}})$ of (16) combines the source, jump, and boundary data present within the Stokes system. The degrees of freedom for $\boldsymbol{u}$ and p are blocked together on an element-wise basis and the block-sparse linear system may be written as $\mathscr{A}x = b$, where $x = (\boldsymbol{u}^{n+1}, p^{n+1})$. The resulting Stokes operator $\mathscr{A}$ is symmetric indefinite and the system (16) is solved via operator coarsening multigrid-preconditioned GMRES methods [66]. A few notes are in order:

- Key to rapid multigrid performance is the choice of pressure penalty stabilization parameter τ_p within the discrete pressure penalty operator E_p, which is defined over all intraphase faces such that

$$v^T E_p p = \int_{\Gamma_0} \tau_p [v][p], \tag{18}$$

for all $v \in V_h^{n+1}$, where

$$\tau_p = \left(\frac{\mu}{\tau h} + \frac{h\rho}{\tau_0 \Delta t}\right)^{-1} \tag{19}$$

harmonically averages the penalty weightings between the viscous and temporal components of the Stokes operator. Here τ and τ_0 are user-defined penalty parameters; optimal values for τ which lead to superior multigrid performance are given in [66].
- Jumps in viscosity across phases are accurately handled by the application of the viscosity-weighted upwinding strategy of [75], which biases the direction of the interphase numerical fluxes based on the liquids' local viscosity coefficients.

3.4.2. Finite difference Marangoni formulation

To properly capture the Marangoni forces present within the multi-layer coating flow problem, which occur as a consequence of surface tension variations caused by solutal concentration gradients along the paint-gas and paint-paint surfaces, we employ a finite difference algorithm for calculating surface gradients along an implicit interface defined via a height function. The algorithm can be stated in two equivalent manners, either as an extension (from interfacial values into the volumetric region) or as a projection (from interfacial values down onto a fictitious $d-1$ dimensional flat plane). We present both formulations, examining the surface gradient $\nabla_S c$ of a general species mass concentration in 2D, with 3D needing only a simple modification of the following presentation. The extension formulation extends the concentration values vertically from the height function, via the function $c_{\text{ext}}(x, y) := c(x, h(x))$. It is clear that

$$\nabla_S c = \nabla c_{\text{ext}} - (\nabla c_{\text{ext}} \cdot \boldsymbol{n})\boldsymbol{n}, \tag{20}$$

where $\boldsymbol{n}$ is the normal vector of the interface. The goal of the projection formulation is to approximate (20) via a finite difference method in the $d-1$ dimensional horizontal plane, specifically at the finite difference nodes on which the height function is defined. The projection formulation proceeds following the steps outlined in Fig. 3:

(a) Project the values of c downwards from the height function onto the $d-1$ dimensional plane, setting $\breve{c}_i := c(x_i, h(x_i))$. Here $\breve{c}_i$ represents the projected value at finite difference node $x_i \in \mathbb{R}^{d-1}$ and the height function values $c(x_i, h(x_i))$ are computed from the traces of the DG polynomials. Additionally, the normal vector $\boldsymbol{n}$ is projected onto the $d-1$ dimensional plane, where

$$\text{2D:}\quad \boldsymbol{n} = \frac{1}{\sqrt{h_x^2+1}}(-h_x, 1), \qquad \text{3D:}\quad \boldsymbol{n} = \frac{1}{\sqrt{h_x^2+h_y^2+1}}(-h_x, -h_y, 1),$$

setting $\breve{\boldsymbol{n}}_i = \boldsymbol{n}(x_i, h(x_i))$, with the derivatives of the height function calculated via central finite differences.

(b) Next the projected gradient $\widetilde{\nabla c}_i$ is computed via central finite differences. For example, a 2^{nd} order formulation in 2D is given by

$$\widetilde{\nabla c}_i = \left(\frac{\breve{c}_{i+1} - \breve{c}_{i-1}}{2\Delta x}, 0\right). \tag{21}$$

[6] The lifting operator is modified from its form in Appendix A to impose Dirichlet boundary and interfacial jump conditions.

[7] It is advantageous to abuse notation and view u as either (a) a vector of coefficients relative to the nodal basis or (b) a polynomial in V_h. The choice of which should be clear by the context.

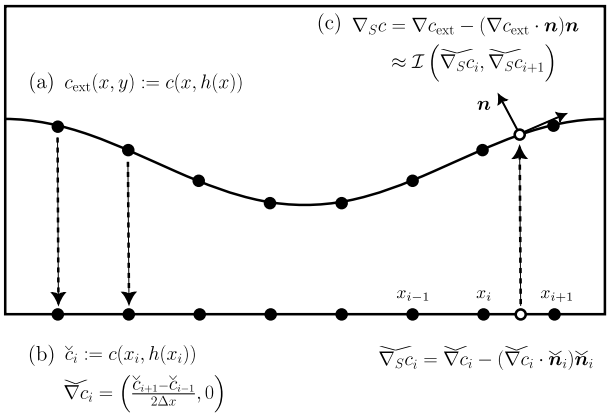


Fig. 3. A 2nd-order example of the finite difference projection algorithm for surface gradient calculations in 2D. (a) Concentration values are projected downwards onto the finite difference nodes • of the height function along the $d-1$ dimensional plane. (b) On this plane, the projected surface gradients are approximated through standard central finite differences. (c) The surface gradient approximation at point o along the height function is then the interpolated projected surface gradient lifted upwards from the horizontal plane.

(c) The projected surface gradient is then calculated at each finite-difference node, with

$$\widetilde{\nabla_S c_i} = \widetilde{\nabla c_i} - (\widetilde{\nabla c_i} \cdot \check{\boldsymbol{n}}_i)\check{\boldsymbol{n}}_i. \tag{22}$$

The values of the projected surface gradient are now available along the $d-1$ dimensional plane through interpolation. The surface gradient approximation $\nabla_S c \approx \mathcal{I}(\widetilde{\nabla_S c})$ is therefore the interpolated value projected vertically back onto the height function.

Fig. 3 illustrates a 2^{nd} order accurate formulation of the surface gradient projection algorithm in 2D. The extension to 3D is straightforward, requiring only an additional dimension to the definitions of the normal vector and projected gradient. Section 4.2 demonstrates convergence of the algorithm in both 2D and 3D, specifically for 2^{nd} and 4^{th} order accurate formulations.

3.4.3. A note on time step restrictions

In the micro-flow regime of coating flow dynamics, strong surface tension forces impose a severe constraint on the numerical time step size. Fully resolving small-scale capillary wave dynamics requires limiting the numerical time step within our hybrid framework to $\Delta t < Ch\mu/\gamma$, for some constant C, with h being the mesh spacing.[8] Typical simulations require a time step size around 10^{-4} seconds, which in conjunction with the long time scales relevant to the coating flow problem—on the order of hundreds of seconds—necessitates millions of time steps. Fortunately, the dynamics of paint drying itself provides a source of time step speedup. Typical paints used in industrial applications have a viscosity that, roughly speaking, increases exponentially as a function of resin concentration c_R (see, e.g., the graph of μ in Table 2). Since the time step constraint is directly proportional to μ/γ, and μ tends to increase as solvent evaporates, it follows that we can increase the time step size as a simulation progresses. In our implementation, we update the time step size every 100 time steps according to the formula

$$\Delta t = \min_{c_R}\left(\frac{\mu(c_R)}{\gamma(c_R)} \Big/ \frac{\mu_0}{\gamma_0}\right)\Delta t_0, \tag{23}$$

where Δt_0 is the initial stable time step size required for the initial ratio of viscosity to surface tension μ_0/γ_0. Here, the minimum value of resin concentration (i.e., c_R) is taken from the values of the DG polynomials at the Gauss-Lobotto nodes. This adaptive time stepping approach is a straightforward yet effective means to take advantage of the fact the capillary number is rapidly monotonically increasing: by the end of a typical simulation, the final time step can be more than 100 times larger than the initial time step. Even with this large speedup, further mitigation of the capillary wave time step constraint was necessary in the present work: one of our objectives here is to explore several aspects of multi-layer coating flow, requiring a multitude of parametric studies. To help facilitate this, in addition to the above adaptive time stepping, we also damp the force of surface tension whereby the terms in (10), (11) involving κ are multiplied by 0.1. This ad-hoc alteration is simply so that we can take time steps ten times larger, representing a modest but beneficial speedup. The net effect of the procedures outlined in this section is to reduce the total number of time steps from millions to around 30,000–100,000 for the presented results.

[8] We note our time step restriction's linear $\mathcal{O}(h)$ dependence, as opposed to the $\mathcal{O}\left(h^{3/2}\right)$ restriction commonly seen in capillary wave dynamics.

3.5. Numerical algorithm for the multi-layer coating flow problem

Combining the numerical methods developed in the previous sections, the fully coupled numerical algorithm for the multi-layer coating flow problem is:

1. Set up the background quad/octree.
2. Define the initial height functions for surfaces $\Gamma_{\text{sub}}, \Gamma_{ij}, \Gamma_e$ and construct the initial level set functions ϕ^0.
3. Use ϕ^0 to define the initial implicit mesh, DG polynomial spaces, and LDG operators.
4. Initialize the DG state variables $\boldsymbol{u}^0, p^0$, and c_k^0 at time $t = 0$.
5. For time step $n = 0, 1, 2, \ldots$

(i) Every 100 time steps, update the time step size Δt.
(ii) Compute advection terms: $\nabla \cdot (c_k \boldsymbol{u})^n, \nabla \cdot (\rho \boldsymbol{u}\boldsymbol{u})^n$.
(iii) Calculate inter-paint mixing rates $m = -\frac{\rho}{c_R^n} \sum_k D_k \nabla c_k^n \cdot \boldsymbol{n}$ on Γ_{ij}, solvent evaporation rates $m_k = \frac{\varepsilon}{C} \cdot \frac{(c_i^n)^2}{\sum_j c_j^n}$ on Γ_e, and total evaporation rate $m = \sum_k m_k$.
(iv) Determine the interfacial speed functions (6): $V = \boldsymbol{u} \cdot \boldsymbol{n} - \frac{1}{\rho} m$.
Advect the height functions (12) and update the level set function ϕ^{n+1}.
(v) Use ϕ^{n+1} to create a new implicit mesh for time step $n+1$.
(vi) Transfer all necessary quantities onto the new mesh's DG polynomial spaces and define LDG operators.
(vii) Solve the solvent mass transfer heat operator problem (13) for each c_k^{n+1} such that

$$\begin{cases} \left(\frac{\mathbb{I}}{\Delta t} - \nabla \cdot (D_k \nabla) \right) c_k^{n+1} = \frac{c_k^n}{\Delta t} - \nabla \cdot (c_k \boldsymbol{u})^n & \text{in } \Omega \\ \nabla c_k^{n+1} \cdot \boldsymbol{n} = 0 & \text{on } \Gamma_{\text{sub}} \\ [c_k^{n+1}] = 0 & \text{on } \Gamma_{ij} \\ [D_k \nabla c_k^{n+1} \cdot \boldsymbol{n}] = 0 & \text{on } \Gamma_{ij} \\ m_k = m c_k^{n+1} - \rho D_k \nabla c_k^{n+1} \cdot \boldsymbol{n} & \text{on } \Gamma_e. \end{cases}$$

(viii) Calculate the updated resin mass concentration $c_R^{n+1} = 1 - \sum_k c_k^{n+1}$.
(ix) Determine concentration-dependent rheological parameters: viscosity $\mu(c_R^{n+1})$, surface tension $\gamma(c_R^{n+1})$, and Marangoni forces $\nabla_S \gamma(c_R^{n+1}) = \frac{\partial \gamma}{\partial c_R^{n+1}} \nabla_S c_R^{n+1}$.
(x) Solve the time-dependent Stokes system (16) for $\boldsymbol{u}^{n+1}, p^{n+1}$ such that

$$\begin{cases} \frac{\rho}{\Delta t} \boldsymbol{u}^{n+1} - \nabla \cdot (\mu(\nabla \boldsymbol{u}^{n+1} + (\nabla \boldsymbol{u}^{n+1})^T)) + \nabla p^{n+1} = \frac{\rho}{\Delta t} \boldsymbol{u}^n - \nabla \cdot (\rho \boldsymbol{u}\boldsymbol{u})^n + \rho \boldsymbol{g} & \text{in } \Omega \\ \nabla \cdot \boldsymbol{u}^{n+1} = 0 & \text{in } \Omega \\ \boldsymbol{u}^{n+1} = 0 & \text{on } \Gamma_{\text{sub}} \\ [\boldsymbol{u}^{n+1}] = 0 & \text{on } \Gamma_{ij} \\ [\mu(\nabla \boldsymbol{u}^{n+1} + (\nabla \boldsymbol{u}^{n+1})^T) \cdot \boldsymbol{n} - p^{n+1} \boldsymbol{n}] = -\gamma \kappa \boldsymbol{n} + \nabla_S \gamma & \text{on } \Gamma_{ij} \\ \mu(\nabla \boldsymbol{u}^{n+1} + (\nabla \boldsymbol{u}^{n+1})^T) \cdot \boldsymbol{n} - p^{n+1} \boldsymbol{n} = -p_{\text{ext}} \boldsymbol{n} - \gamma \kappa \boldsymbol{n} + \nabla_S \gamma & \text{on } \Gamma_e. \end{cases}$$

4. Convergence studies

In this section, we test several components of the multi-layer coating flow numerical framework, including convergence of the LDG method for Poisson problems with Robin boundary conditions, the finite difference surface gradient projection algorithm for Marangoni stress calculations, and the fully coupled multi-physics coating flow framework.

4.1. LDG for Poisson problems with Robin boundary conditions

We now test the order of accuracy of the LDG method of Appendix A for Poisson problems with Robin boundary conditions, as well as examine the performance of the associated multigrid solver algorithms. In particular, we demonstrate both high-order accuracy as well as rapid bounded multigrid performance for the challenging case of variable diffusion and Robin coefficients spanning several orders of magnitude on a curved implicitly-defined domain, in both 2D and 3D. Recall the general Poisson problem with Robin boundary conditions (A.1) is given by

$$\begin{cases} -\nabla \cdot (\mu \nabla u) = f & \text{in } \Omega \\ au + (b\mu \nabla u) \cdot \boldsymbol{n} = r & \text{on } \Gamma_R. \end{cases}$$

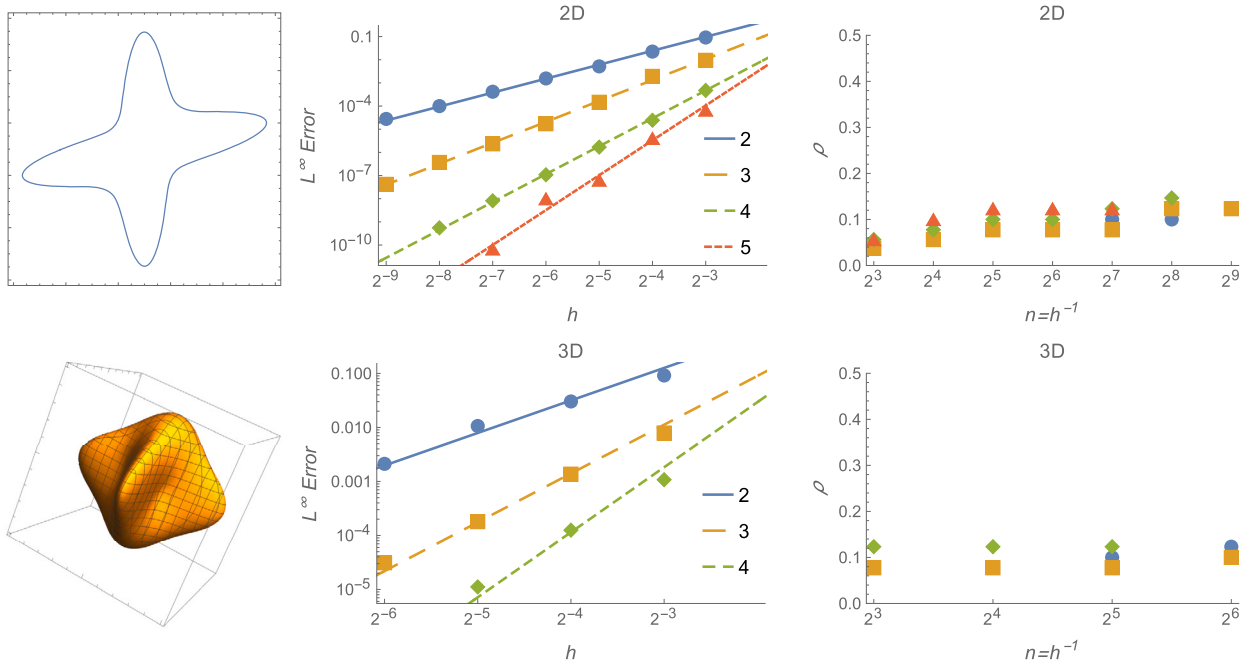


Fig. 4. Convergence rates and multigrid performance for the Poisson problem with Robin boundary conditions on an implicitly-defined curved domain (pictured left) with $\mu, a, b = 1$. h denotes the background mesh spacing and polynomial degrees are represented by ●,■,◆,▲ for $p = 1, 2, 3, 4$ respectively, with the slopes of the lines indicating asymptotic convergence rates.

We test convergence by comparing against an exact offset sinusoid solution, setting $u : \Omega \to \mathbb{R}$ to

$$u(x) = \prod_{i=1}^{d} \sin 2\pi(x_i - 0.05), \tag{24}$$

and using this exact solution to generate the source data f and Robin boundary data r. Multigrid efficiency is assessed by the average residual reduction factor per iteration of the multigrid-preconditioned conjugate gradient algorithm, which is defined as

$$\rho = \exp\left(\frac{1}{N}\log\left(\frac{||VAx_N - Vb||_2}{||VAx_0 - Vb||_2}\right)\right), \tag{25}$$

where N is the number of iterations needed to reduce the residual by a factor of 10^{10} from its initial value. Here V represents the multigrid V-cycle preconditioner while A represents the discrete Laplacian. All multigrid tests use a homogeneous $b = 0$ and a random initial guess x_0, thereby assessing performance across the full spectrum of eigenmodes. We test our formulation within a curved "amoeba" domain embedded in a $[0, 1]^d$ box, implicitly defined by the zero level set of

$$\phi(x, y, (z)) = \begin{cases} r^2 - 0.1 - \left(y_c^4 + 10x_c^3 y_c - 20x_c^2 y_c^2\right)/\left(2r^2\right) & \text{in 2D} \\ r^2 - 0.1 - \left(y_c^4 + 10x_c^3 y_c - 20x_c^2 y_c^2 + z_c^4\right)/\left(2r^2\right) & \text{in 3D} \end{cases},$$

where $(x_c, y_c, z_c) = (x, y, z) - 0.5$ and $r^2 = x_c^2 + y_c^2 (+ z_c^2)$. Both the 2D and 3D domains are shown in Fig. 4.

Test 1: Constant diffusion and Robin coefficients: We begin with perhaps the simplest nontrivial Robin boundary condition, setting unit diffusion coefficient $\mu = 1$, with equal unit weighting between the Robin coefficients, $a = b = 1$. Fig. 4 presents the computed L^∞ errors against the exact solution along with the multigrid convergence rate ρ in both 2D and 3D. For each polynomial order p, optimal $p + 1$ order accuracy is achieved, as indicated by the fitted lines. Good multigrid performance is demonstrated in which the iteration count remains bounded as the background mesh spacing h tends towards zero. The multigrid algorithm has a convergence rate of $\rho \leq 0.15$ in all cases, indicating 6-12 multigrid-preconditioned conjugate gradient iterations are needed to reduce the residual by a factor of 10^{10}.

Test 2: Variable diffusion and Robin coefficients spanning several orders of magnitude: Next, we examine a challenging case with variable coefficients, where the diffusion coefficient varies by four orders of magnitude and the Robin coefficients by eight orders of magnitude throughout the domain, setting

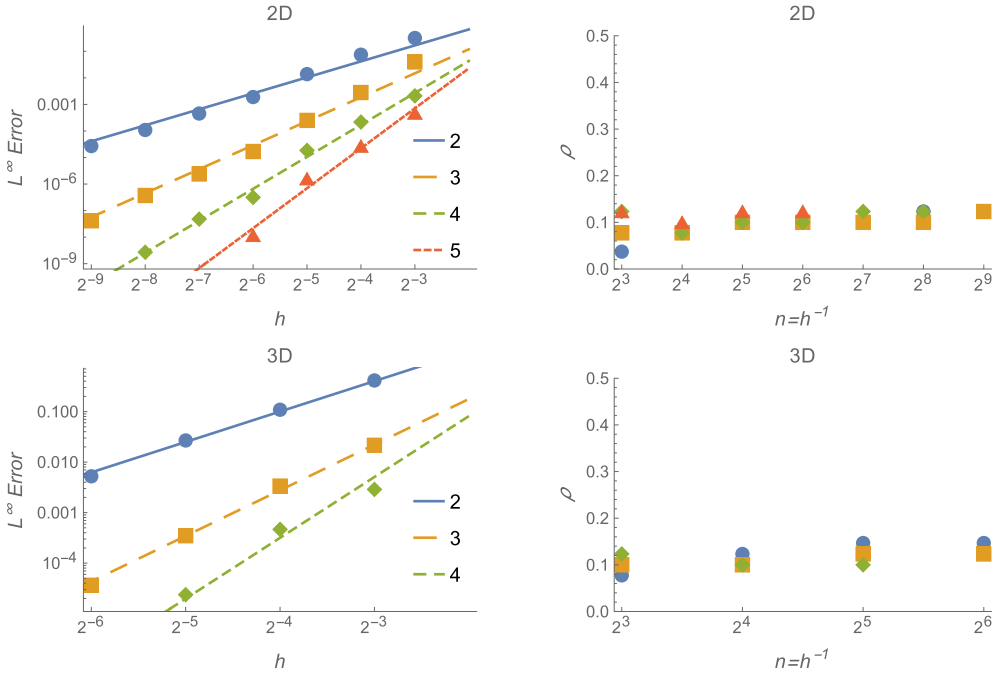


Fig. 5. Convergence rates and multigrid performance for the Poisson problem with Robin boundary conditions with variable μ, a, b spanning several orders of magnitude. h denotes the background mesh spacing and polynomial degrees are represented by $\bullet, \blacksquare, \blacklozenge, \blacktriangle$ for $p = 1, 2, 3, 4$ respectively, with the slopes of the lines indicating asymptotic convergence rates.

in 2D:	in 3D:
$\mu = 10^{2\sin(2\pi(x-0.1))\sin(2\pi(y+0.1))}$	$\mu = 10^{2\sin(2\pi(x-0.1))\sin(2\pi(y+0.1))\sin(2\pi(z-0.1))}$
$a = 10^{-4+8\sin(\pi x/2)\sin(\pi y/2)}$	$a = 10^{-4+8\sin(\pi x/2)\sin(\pi y/2)\sin(\pi(z+0.5)/2)}$
$b = 10^{4-8\sin(\pi x/2)\sin(\pi y/2)}$	$b = 10^{4-8\sin(\pi x/2)\sin(\pi y/2)\sin(\pi(z+0.5)/2)}$,

which gives a diffusion coefficient ranging from 10^{-2} to 10^{2} and a spectrum of Robin coefficient ratios a/b ranging from 10^{-8} to 2×10^{5} for both 2D and 3D. Fig. 5 illustrates that the LDG method produces high-order accurate solutions and that the multigrid algorithm performs well even in this challenging setting, achieving optimal $p + 1$ order accuracy and good bounded multigrid performance with $\rho \leq 0.15$ for each polynomial order.

In the limits: The Robin boundary condition, $au + (b\mu\nabla u) \cdot \boldsymbol{n} = r$ on Γ_R, can be viewed as a weighted combination of Dirichlet and Neumann boundary conditions. As $a \to 0$ with b nonzero, the Robin boundary condition approaches a pure Neumann condition, which for Poisson problems requires appropriate treatment of the kernel, being in that case the span of globally-constant functions. In the limit $a \to \infty$ or $b \to 0$, the Robin boundary condition approaches a pure Dirichlet condition, whose LDG implementation requires additional penalty stabilization to ensure a well-conditioned discrete Laplacian. It is possible to incorporate this limit and its requirements into our LDG formulation—for example, one could adapt the Nitsche's finite element method of [76] for general Robin boundary conditions to our problem—but this limit is not relevant to the multi-layer coating flow problem and therefore not explored here.

4.2. Finite difference Marangoni formulation

Fig. 6 shows the results of a convergence study examining the finite difference projection algorithm for computing surface gradients within the Marangoni stress calculations. 3D tests are performed along the surface of Fig. 6 (left) while 2D tests are performed along a central slice of this surface. The surface is embedded within a $[0, L]^d$ box, with $L = 100$ μm to match the micro-fluidic domains of interest. The 2^{nd} order formulations use 2^{nd} order central differences with $p = 2$ DG polynomials and the 4^{th} order formulations use 4^{th} order central differences with $p = 3$ DG polynomials. We have opted to use the 2nd order method in the remainder of the results presented in this work; this is simply for reasons of consistency, e.g., our finite difference curvature calculation is also second-order. In the concluding remarks, we provide further commentary on the usefulness of higher-order discretizations in coating flow problems.

4.3. Convergence for the multi-layer coating flow problem

We next test convergence of the fully coupled multi-layer coating flow framework. As discussed in the introduction, some surface tension profiles lead to hydrodynamic instabilities, which naturally complicate a grid convergence study. For example, Fig. 9 in the

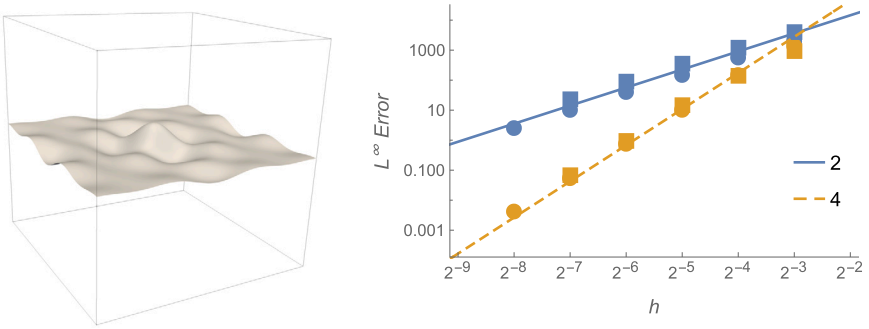


Fig. 6. Convergence tests of the finite difference surface gradient calculation. **Left:** The 3D surface on which the tests are performed; 2D tests are performed along a central slice of this surface. **Right:** L^∞ errors and convergence rates of the finite difference surface gradient calculation for 2^{nd} and 4^{th} order formulations, with ●, ■ representing 2D and 3D calculations respectively, with the slopes of the lines indicating asymptotic convergence rates.

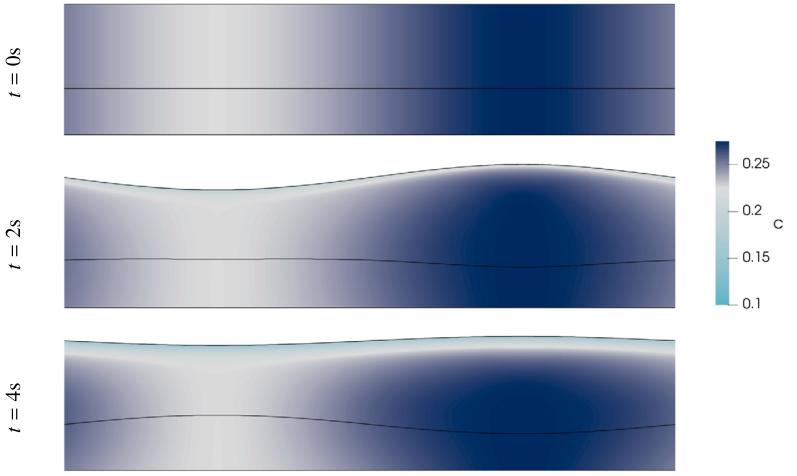


Fig. 7. The evolution of the solvent mass concentration profile of a 128×128 mesh for the two-layer coating flow convergence test considered in section 4.3. The Marangoni effect drives flow along the top surface from regions of low surface tension (white) to regions of high tension (dark blue) while evaporation creates a low-solvent boundary layer. Marangoni plumes do not form as in Fig. 9 since this test is in the short-wave-stable regime. The solvent mass concentration values are indicated by the color bar, while the curved black lines indicate the paint-gas surface Γ_e and embedded paint-paint interface Γ_{ij}.

results section illustrates the evolution of a single-layer coating in 2D with a surface tension value that increases with respect to resin concentration. The corresponding Marangoni-driven evaporative flow produces the short-wave hydrodynamic instabilities discussed in the introduction; in particular, perturbations in solutal concentration along the free evaporative surface grow to form Marangoni plumes/roll-cells. Despite the physical significance of this Marangoni-driven flow regime, it is not suitable for a grid-convergence study as the instability naturally gives rise to grid-dependent dynamics, very similar in character to, e.g., the well-known Kelvin-Helmholtz instability.

It is instead appropriate to examine a regime in which the Marangoni forces play not only a pivotal role, but also one in which stable dynamics ensue. Such flows arise when the surface tension value decreases with respect to resin concentration, placing the system into a short-wave-stable regime. One could view this as taking the unstable regime, but "flipping the sign" on the Marangoni forcing term within the simulation code. Doing so therefore stress tests each part of the implementation, in a physically-motivated set of parameters, but yields stable dynamics for the purposes of a grid convergence study.

Specifically, we examine convergence for both the single and two-layer coating flow problems in 2D within a 400 μm × 100 μm domain. The specified mesh sizes (e.g., 32×32) represent the number of DG cells of the background mesh within an $L \times L$ block, where $L = 100$ μm is the characteristic length scale in this and all remaining studies. The initial height function values and rheological parameters are as specified in Table 2 and the initial fluid velocity is set to zero. A single solvent is considered with its initial mass concentration profile set to $c = 0.25 + 0.025\cos(\pi x/2L)$ in order to produce smooth Marangoni forces along the free evaporative surface that are resolvable by the coarsest (16×16) mesh. Boundary and jump conditions that are not compatible with the initial conditions (i.e., specifically the Robin and stress conditions) are "slowly turned on" via a ramping method.[9] In this study, we set the time step size small enough so that the spatial errors dominate. Fig. 7 shows the evolution of the solvent mass concentration profile

[9] For example, an interfacial jump condition of the form $[u] = f$ is replaced by $[u] = R(t)f$, where $R(t)$ is a piecewise linear ramping function defined by $R(t) = t/T_R$ when $t \leq T_R$ and $R(t) = 1$ when $t > T_R$. Here T_R is the time at which ramping is complete, equal to 50% of the final time in our convergence tests.

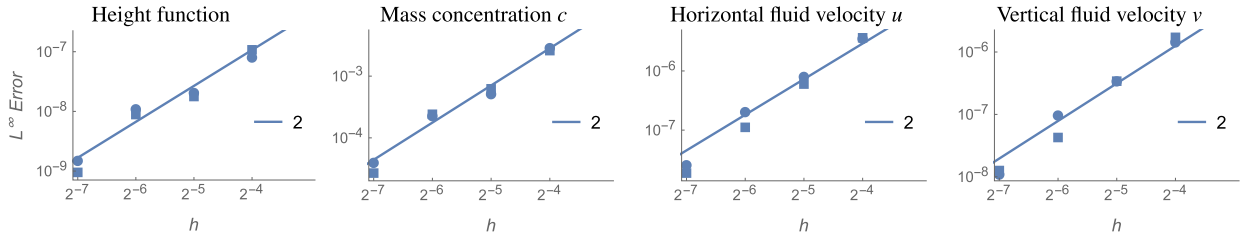


Fig. 8. L^∞ errors and convergence rates of the multi-physics coating flow problem with short-wave-stable parameters. Here ●, ■ represent single-layer and two-layer calculations, respectively, while the slopes of the lines indicate 2^{nd} order asymptotic convergence rates.

of the two-layer problem for $T = 5$ seconds, during which fluid is pulled along the free evaporative surface by the Marangoni forces while the evaporation quickly forms a solutal boundary layer, noting a heightened evaporation coefficient ε.

The computed solutions are tested against a reference solution u_{ref} computed on a fine 256×256 ($h = L/2^8$) mesh, with comparison performed using a maximum norm metric. Since the interface locations will differ slightly for different background grid sizes, the error metric for the computed DG polynomial values is calculated according to

$$\max_{t\in(0,T]} \sup_{x\in\bigcup_i(\Omega_i \bigcap \Omega_{\text{ref},i})} ||u - u_{\text{ref}}||_{L^\infty}, \tag{26}$$

where $\Omega_{\text{ref},i}$ represents phase i of the reference mesh. This metric calculates the L^∞ errors between points that have the same phase identifier for both meshes and establishes convergence of the DG polynomial solutions when the interface location is convergent. Using this metric, Fig. 8 demonstrates the results of this convergence test. We observe second-order spatial accuracy in the L^∞ norm for the height function locations, solvent mass concentration profile, and fluid velocity field, for all time $t \in (0, T]$.

5. Results

In the previous sections, a mathematical model and several numerical methods are developed for the multi-layer coating flow problem. The framework solves the evolution equations, boundary, and jump conditions of this multi-physics problem and captures the coupling between the evaporative mass transfer system and the multi-phase interfacial quasi-Newtonian fluid dynamics. In this section, several results from the hybrid numerical framework are presented. These include numerical tests of well-known experimentally observed phenomena, specifically:

- Short-wave stationary Marangoni instabilities and the development of Marangoni plumes, which are described mathematically by Pearson [16] and examined experimentally in [19,59].
- 3D two-layer coating flows illustrating the formation of hexagonal-shaped Bénard cells [14,15].
- The long-wave oscillatory deformational modes of Scriven and Sternling [21] and their impact on immersed interfaces; these modes are highlighted in the experimental results of [11,22].

Additionally, we apply our numerical framework to study several flow regimes not easily assessed through laboratory experiment, exploring:

- Multi-solvent evaporative paint dynamics, specifically single-layer coatings that are composed of multiple solvents each with different evaporation rates or diffusion coefficients.
- A Marangoni-driven drilling phenomena within a multi-layer coating, with links to experimentally observed "cratering" that sometimes occurs in automotive paint films [77].
- "Interfacial turbulence" within a multi-layer matter cascade.

The physical parameters of the liquid paints are chosen to match industrially-relevant conditions and are motivated by experiments on automobile paint coatings. The specific rheological parameters for each study are given in Table 2. All results assume a uniform initial coating and that the initial solvent mass concentrations are constant in each phase. All embedded paint-paint surfaces Γ_{ij} are initially flat while the initial free evaporative surface Γ_e includes a small perturbation to induce Marangoni flow. The substrate is completely flat and placed horizontally, with gravity pulling in the vertical direction, however, we note that the effect of gravity is small in this setting. Tests involving vertically-oriented substrates with industrially-relevant roughness profiles can be found in [3].

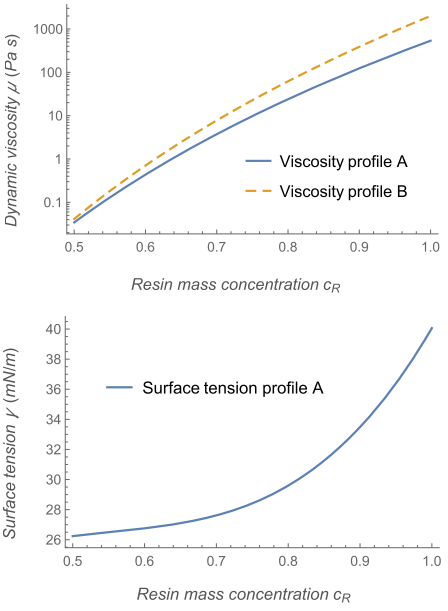
5.1. Short-wave Marangoni instabilities

Fig. 9 illustrates the evolution of a single-layer coating in 2D. The Marangoni-driven evaporative flow produces the short-wave hydrodynamic instabilities discussed in the introduction and perturbations in solutal concentration along the free evaporative surface grow to form Marangoni plumes/roll-cells. The initial small-scale plumes quickly merge together and coalesce into larger structures, with new plumes continuing to form and merge throughout the evaporative process. This result is consistent with that of [58,59],

Table 2
The rheological parameters used in the numerical simulations. Note the mesh resolution specifies the number of cells of the DG background mesh per 100 μm block. Included here are the viscosity and surface tension profiles for simulations of automobile coatings.

Parameter (units)	Symbol	Fig. 7	Fig. 9	Fig. 11	Fig. 14	Fig. 15	Fig. 16	Fig. 17
Number of solvents	C	1	1	1	1	1	3	3
Number of layers		2	1	2	2	2	1	1
Mesh resolution		128	32	32	32	32	64	64
Domain width (μm)		400	400	100	25600	25600	400	400
Init. height Γ_e (μm)		90	90	90	90	90	50	50
Init. height Γ_{ij} (μm)		35	—	35	35	35	—	—
Fluid density (kg.m^{-3})	ρ	1000	1000	1000	1000	1000	1000	1000
Basecoat viscosity (Pa.s)	μ_{BC}	4	—	Visc. B	Visc. B	Visc. B	—	—
Clearcoat viscosity (Pa.s)	μ_{CC}	2	2	Visc. A	Visc. A	Visc. B	Visc. A	Visc. A
Basecoat diffusion (m^2.s^{-1})	D_{BC}	2×10^{-12}	—	1×10^{-12}	1×10^{-12}	1×10^{-12}	—	—
Clearcoat diffusion (m^2.s^{-1})	D_{CC}	1×10^{-11}	1×10^{-11}	4×10^{-12}	5×10^{-12}	5×10^{-12}	5×10^{-12}	$1.25, 2.5, 5.0\times10^{-12}$
Surface tension Γ_e (mN.m^{-1})	γ_e	$50-25c_R$	$10+25c_R$	$27+6.67c_R$	St. A	St. A	$27+6.67c_R$	$27+6.67c_R$
Surface tension Γ_{ij} (mN.m^{-1})	γ_{ij}	30	—	3	30	30	—	—
Evaporation coeff (kg.m^{-2}.s^{-1})	ϵ	1×10^{-3}	3.33×10^{-4}	3.33×10^{-4}	3.33×10^{-4}	3.33×10^{-4}	$3.33, 5.0, 7.67\times10^{-4}$	3.33×10^{-4}
Init. basecoat resin cons	c_R^{BC}	As specified	—	0.75	0.75	0.75	—	—
Init. clearcoat resin cons	c_R^{CC}	As specified	0.65	0.65	0.65	0.65	0.65	0.65

Parameter (units)	Symbol	Fig. 18	Fig. 19	Fig. 20
Number of solvents	C	1	1	1
Number of layers		2	3	3
Mesh resolution		32	64	64
Domain width (μm)		25600	400	400
Init. height Γ_e (μm)		90	180	180
Init. height Γ_{ij} (μm)		35	35,90	35, 90
Fluid density (kg.m^{-3})	ρ	1000	1000	1000
Basecoat viscosity (Pa.s)	μ_{BC}	Visc. B	8,4	4,2
Clearcoat viscosity (Pa.s)	μ_{CC}	Visc. A	2	1
Basecoat diffusion (m^2.s^{-1})	D_{BC}	1×10^{-12}	$1.25, 2.5\times10^{-12}$	$1.25, 2.5\times10^{-12}$
Clearcoat diffusion (m^2.s^{-1})	D_{CC}	5×10^{-12}	5×10^{-12}	5×10^{-12}
Surface tension Γ_e (mN.m^{-1})	γ_e	St. A	$10+25c_R$	$10+25c_R$
Surface tension Γ_{ij} (mN.m^{-1})	γ_{ij}	6	$10+25c_R$	$10+25c_R$
Evaporation coeff (kg.m^{-2}.s^{-1})	ϵ	3.33×10^{-4}	0	0
Init. basecoat resin cons	c_R^{BC}	0.75	0.8,1	0.8,1
Init. clearcoat resin cons	c_R^{CC}	0.65	1	1



which classifies this phenomenon as the coarsening of low-order Marangoni roll cells into larger high-order cells. Our numerical results are also in good qualitative agreement with the experimental results of Zhong et al. [78], which capture a side profile of the formation and merger of short-wave Marangoni plumes in an evaporating *n*-heptane/ether mixture through the use of shadowgraph imagery. These images are reproduced in Fig. 10 (left).

5.1.1. 3D results

Fig. 11 shows the solvent mass concentration profile from a 3D simulation of a two-layer coating within a 100 μm cube. A notable feature in this result is that the short-wave Marangoni plumes of the 2D simulations become sheets in 3D that are rooted to filaments on the free evaporative surface Γ_e. These filaments merge in a variety of patterns, including a hexagonal diamond pattern within the first few seconds of the simulation and later hexagonal reticulated patterns. Hexagons are a common shape found in Marangoni-driven flows and are often seen in thermally driven Rayleigh-Bénard convection cells [14]. Eventually, the system

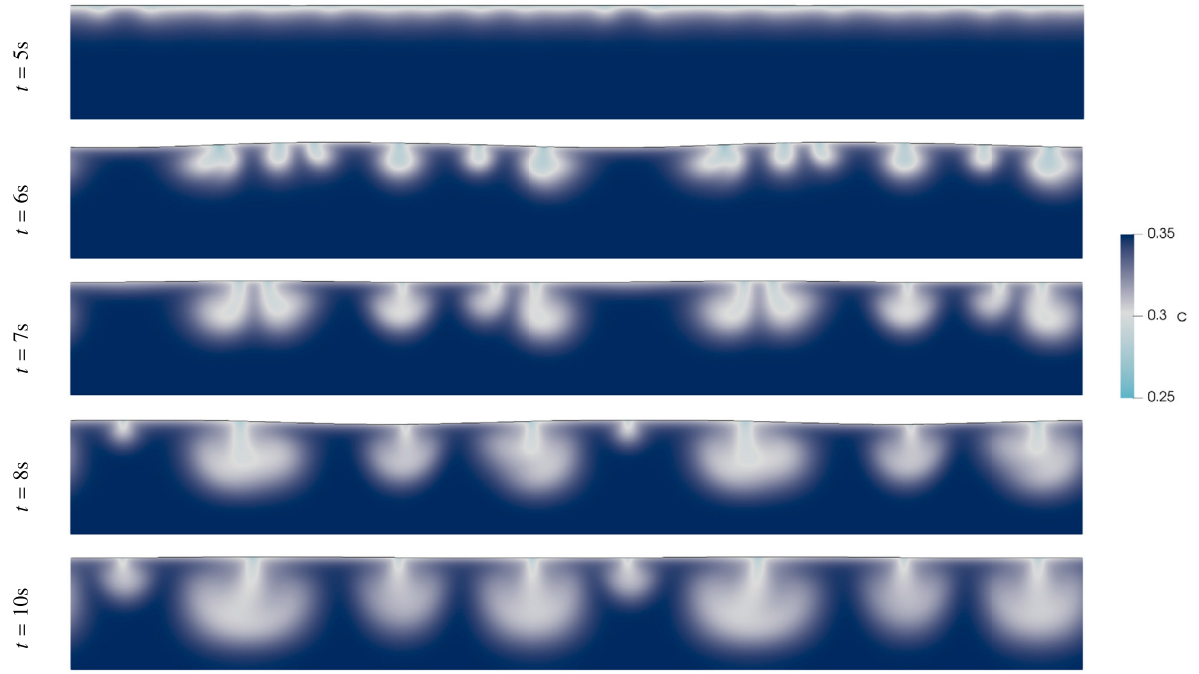


Fig. 9. The evolution of the solvent mass concentration profile illustrating the formation and merger of Marangoni plumes/roll cells within a single paint film with short-wave-unstable parameters, at the indicated times. The DG background mesh has 32×32 cells per 100 μm × 100 μm block and the 400 μm × 100 μm computational domain is repeated along its periodic axis for presentation. The solvent mass concentration values are indicated by the color bar.

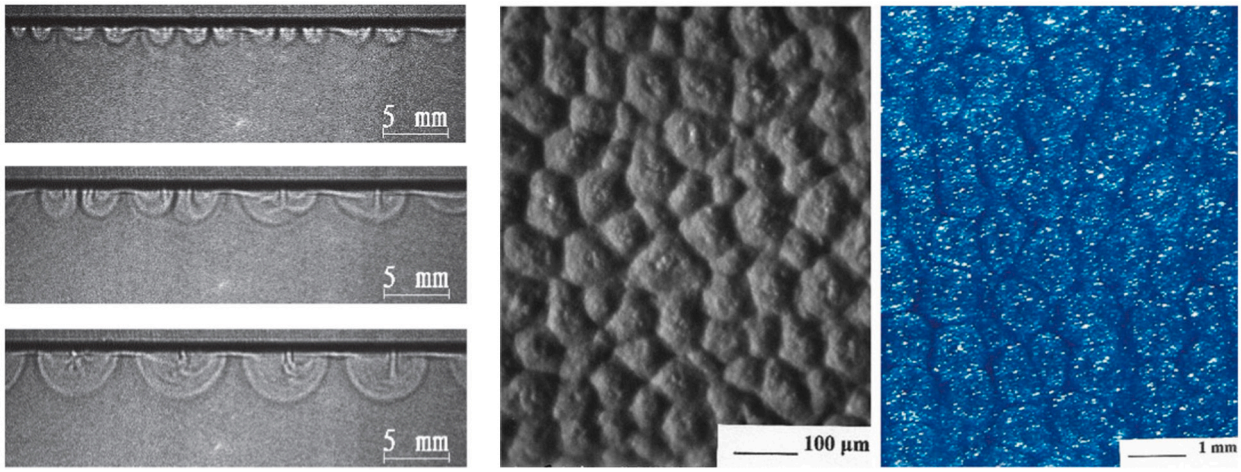


Fig. 10. Left: Shadowgraph imagery illustrating the formation and merger of short-wave Marangoni plumes in an evaporating *n*-heptane/ether mixture, over 10 s. **Center/Right:** Marangoni-induced hexagonal-shaped Bénard cells from an aluminum can coating and an automobile top coat. Images reproduced with permission from [78] and [77] respectively.

reaches a "steady-state" arrangement of circulating Bénard cells that impact and deform the basecoat. Here the cells form a square shape; this is, however, likely due to the enforcement of periodic boundary conditions on a small domain. Examples from [77] of hexagonal Bénard cells within paint coatings can be seen in Fig. 10 (center/right). The progression of the 3D Marangoni sheet structures is illustrated in Figs. 12 and 13, which capture the evolution of solvent mass concentration isocontours (for $c = 0.31$ and $c = 0.29$, resp.) and highlight the Marangoni-induced flow patterns within the clearcoat over the first 20 seconds of the simulation.

5.2. Long-wave deformational modes in multi-layer automobile coatings

The mathematical model and numerical methods developed in this paper are designed, in part, to predict the ultimate surface roughness of multi-layer automobile paint coatings. To this end, we present a few of our results (see [3]) from a 2D parametric study on two-layer coatings that identifies some key features impacting the final surface profile. To properly capture the long-wave

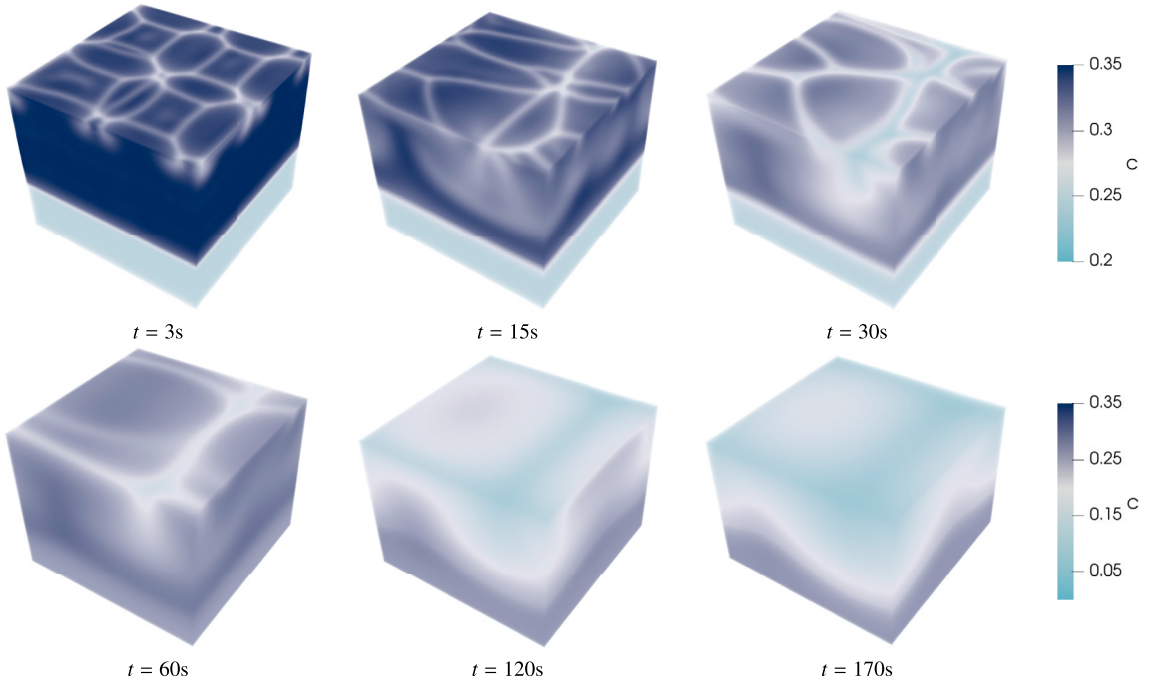


Fig. 11. Snapshots of the solvent mass concentration profile in 3D, at the indicated times. Hexagonal-shaped cells form on the top evaporative surface and Marangoni plumes form in a sheet-like structure. To better illustrate the solvent concentration, each row applies its own color bar scale.

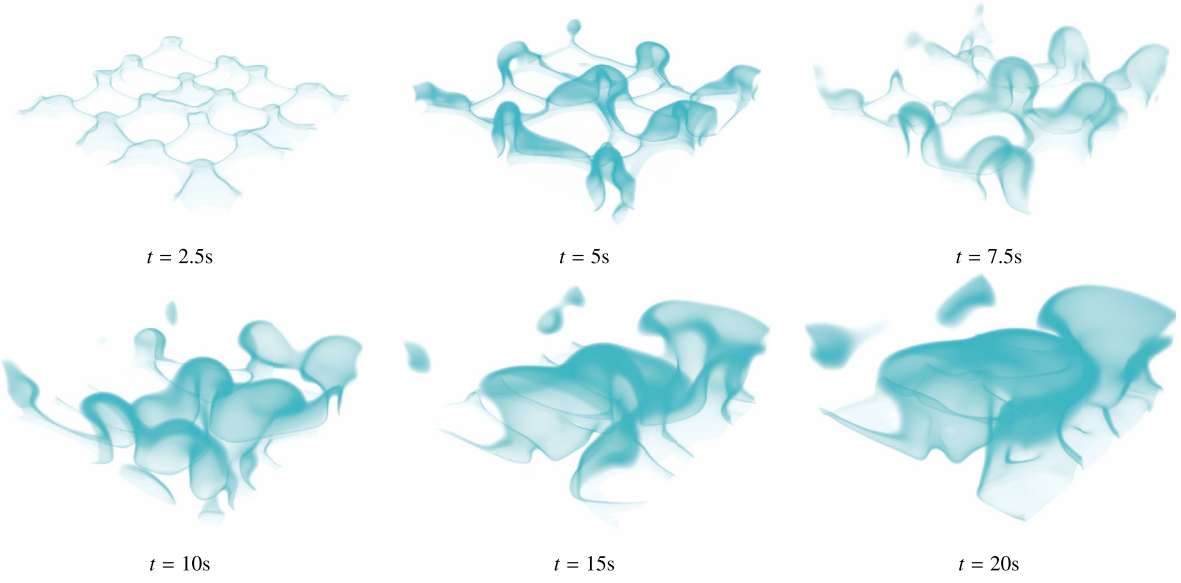


Fig. 12. The evolution of solvent mass concentration isosurfaces for $c = 0.31$, illustrating the Marangoni-driven fluid flow within the bulk of the clearcoat, at the indicated times. For illustration, the z-axis is inverted so that the base of the figures is the free evaporative surface.

surface modes found along dried paint films, the wavelengths of which range from $1 - 10$ mm, the 2D simulations in this section are performed on long-skinny domains with a horizontal length of 25.6 mm and an aspect ratio of 256×1. Additionally, we note that since little is known about the values of surface tension of the embedded paint-paint surface, constant surface tension is considered along Γ_{ij} in this section.

Fig. 14 demonstrates the effect of the long-wave Marangoni modes on the paint films. This figure shows the evolution of the solvent mass concentration profile along the entire 25.6 mm $\times$ 0.1 mm domain, with the horizontal x-axis scaled by 1/20. Around 60 seconds into the simulation, long-wave surface modes become noticeable along the free evaporative surface. As described mathematically in [21,23] and captured experimentally in [11,22], these modes deform the evaporative surface and are oscillatory, e.g., the troughs on the left side of the figure at $t = 60$ seconds become peaks at around 120 seconds. Fourier frequency data (see [3])

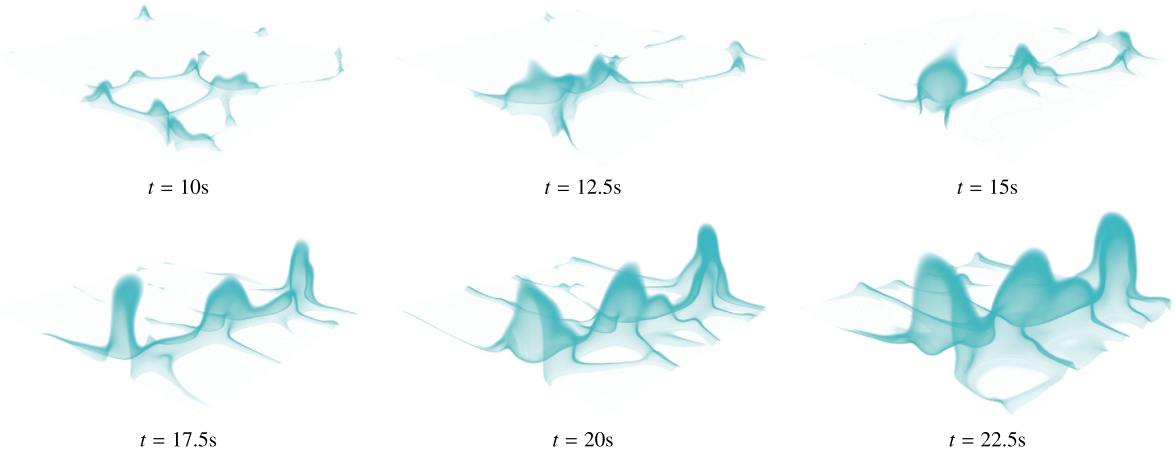


Fig. 13. The evolution of solvent mass concentration isosurfaces for $c = 0.29$, illustrating the Marangoni-driven fluid flow within the bulk of the clearcoat, at the indicated times. For illustration, the z-axis is inverted so that the base of the figures is the free evaporative surface.

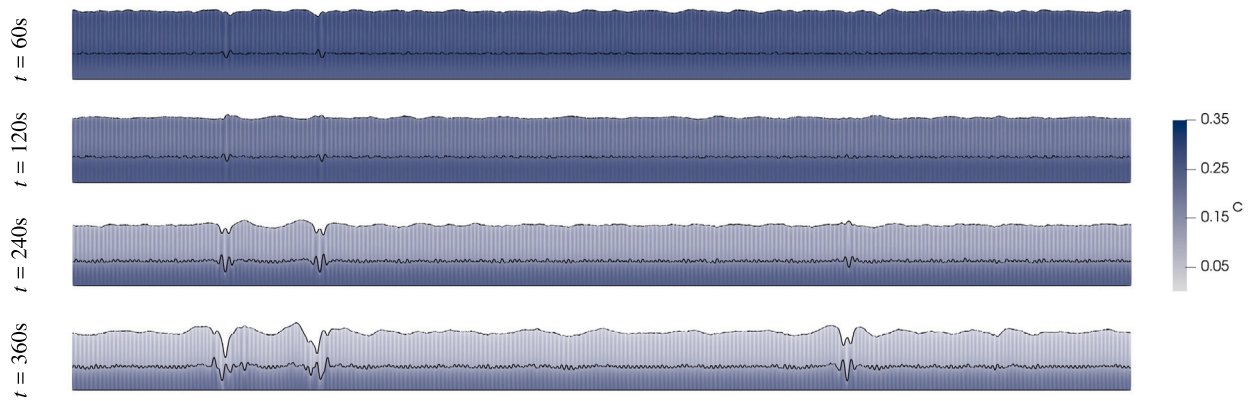


Fig. 14. Long-wave instability causing deformation. Illustrated is the solvent mass concentration profile at the indicated times for a 25.6 mm × 0.1 mm domain with the horizontal x-direction scaled by 1/20. The left side of the images illustrates the oscillatory nature of paint drying, with peaks becoming troughs, and vice versa. Here, the long-wave Marangoni instabilities grow and compound, and eventually the paint layers tear.

shows that, in this regime of coating flow dynamics, the peak amplitude wavelength is roughly 1 mm and that the oscillations grow in amplitude each cycle.

Fig. 15 shows the drying of a two-layer coating in its entirety, with the final simulation time being 45 minutes. The long-wave oscillatory Marangoni modes deform the paint-gas and paint-paint surfaces during the first 10 minutes of drying and these early deformations leave a lasting imprint on the final paint profile—with moderate smoothing and settling over time. This test demonstrates the utility of the adaptive time stepping technique of section 3.4.3, where the time step size is changed based on the ratio of viscosity to surface tension. In this particular problem, the time step size is nearly 400 times larger at the end of the simulation than at the beginning.

5.3. Multi-solvent evaporation

To demonstrate the motion and evaporation of coatings with multiple dissolved species, Fig. 16 shows the mass concentration profile of a single-layer paint composed of three solvents that each evaporate at a different rate. The initial mass concentration is constant and equal for each solvent, the mass diffusion coefficients are set to 5×10^{-12} m^2.s^{-1}, and the ratio of the solvents' evaporation coefficients are 1, 1.5, and 2.3 times a base value of $\varepsilon = 3.33 \times 10^{-4}$ kg.m^{-2}.s^{-1}. The effect of the different evaporation rates is clear, e.g., with more material ejected in the bottom case, while each solvent has the same dynamical plume structure driven by the overall fluid flow and the short-wave Marangoni instabilities.

Fig. 17 illustrates the evaporation of a single-layer paint composed of three solvents each with different mass diffusion coefficients. The mass diffusion coefficients are taken to be 1.25×10^{-12} m^2.s^{-1}, 2.5×10^{-12} m^2.s^{-1}, and 5×10^{-12} m^2.s^{-1}, and the evaporation coefficient is set to 3.33×10^{-4} kg.m^{-2}.s^{-1} for each solvent. This test illustrates the dramatic impact of the diffusion coefficient on the Marangoni plume structures within the coating flow problem. The low diffusion solvent exhibits tighter formations and thinner boundary layers, whereas the higher diffusion solvents have smoothed-out features and thicker boundary layers. In general, larger

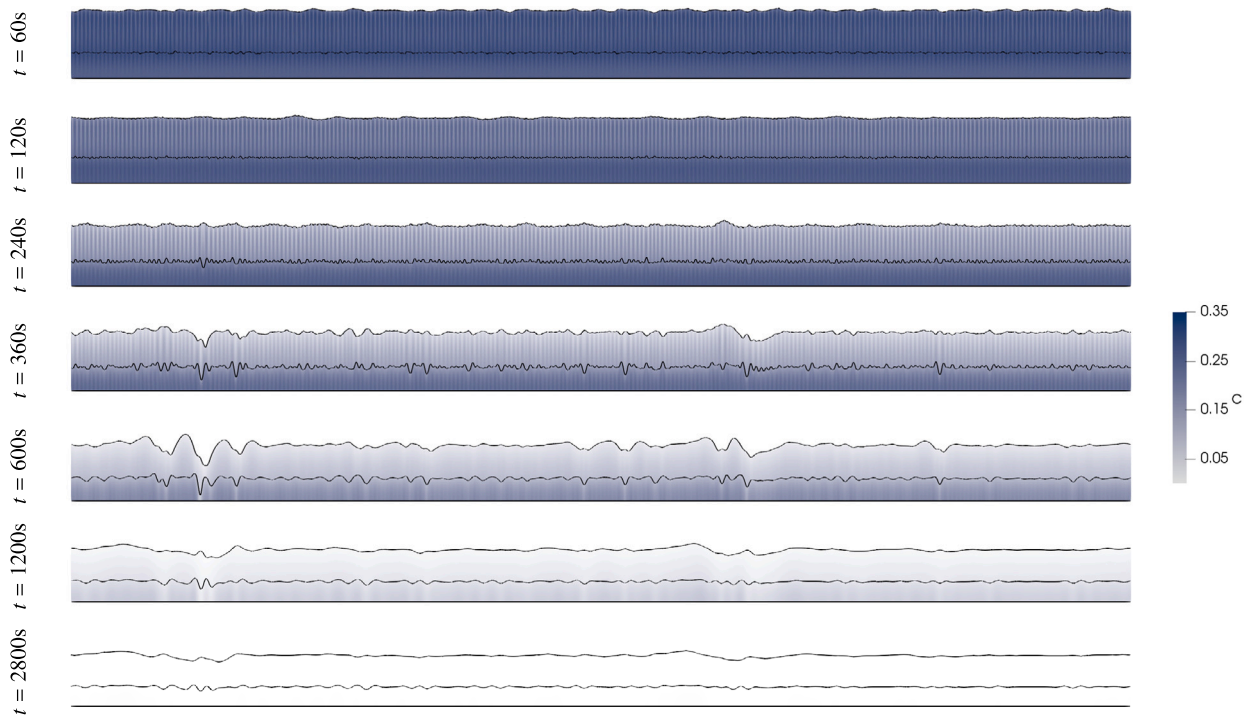


Fig. 15. A two-layer paint drying in entirety. Illustrated is the solvent mass concentration profile at the indicated times for a 25.6 mm × 0.1 mm domain with the horizontal x-direction scaled by 1/20. The deformations from the Marangoni effect leave an imprint on the final paint surface.

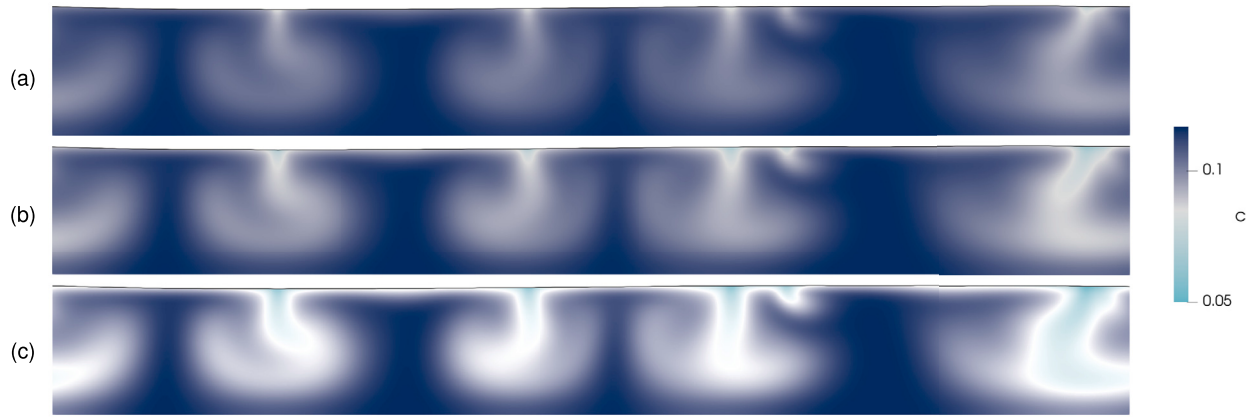


Fig. 16. The solvent mass concentration profile of three solvents evaporating at different rates. Relative to a reference coefficient of $\epsilon = 3.3 \times 10^{-4}$ kg.m^{-2}.s^{-1}, the solvent evaporation coefficient corresponding to each example is (a) ϵ, (b) 1.5ϵ, and (c) 2.3ϵ. These results are taken 20 seconds into the simulation.

Marangoni forces are present in low diffusion liquids, owing to the sharper/thinner solutal boundary layers forming at the free evaporative surface.

5.4. Marangoni drilling

To highlight the coupling of paint films within the multi-layer system, Fig. 18 (top) shows the evolution of the solvent mass concentration within a 1.6 mm window of a two-layer coating. This figure illustrates the uniformity of the short-wave Marangoni plumes within the clearcoat and highlights an irregularity not seen in the single-layer case. Around 45 seconds into the simulation, two large Marangoni plumes merge together and drill into the basecoat, causing a wide deformation that raises the basecoat and drags down the clearcoat along the edges of the plume. About 90 seconds in, the embedded paint-paint surface and the free evaporative surface are close to intersecting such that the clearcoat recedes and exposes the basecoat to air. In industrial operations and laboratory experiments, multi-layer paints sometimes develop holes while drying. A conjecture is that these holes form when the interfaces Γ_e and Γ_{ij} intersect, exposing the basecoats to air, and the recession progresses further such that substrate is exposed and the system

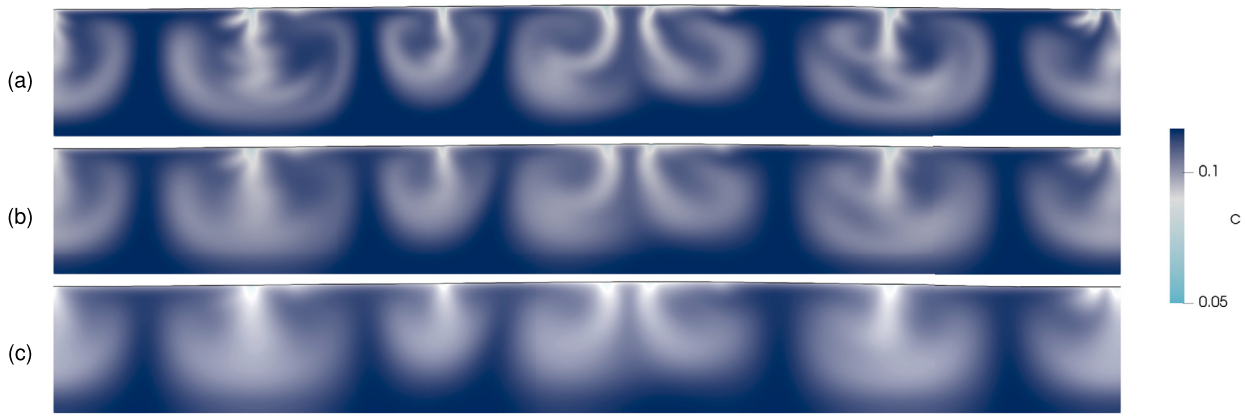


Fig. 17. The solvent mass concentration profile of three solvents with different mass diffusion coefficients. Relative to a reference coefficient of $D = 1.25 \times 10^{-12}$ $m^2.s^{-1}$, the diffusion coefficient corresponding to each example is (a) D, (b) $2D$, and (c) $4D$. These results are taken 20 seconds into the simulation.

undergoes dewetting. The drilling phenomena may be the first phase of this process. Additionally, this result is reminiscent of the "cratering" irregularity that sometimes occurs in automotive paint coatings [77], i.e., the rapid collapse of a coating caused by the Marangoni effect in the presence of a low surface tension impurity.

Also observed within Fig. 18 (along the sides) is the progression of the short-wave Marangoni plumes outside of the drill towards steady-state Bénard cells. During the initial stages of drying, the Marangoni effect drives the formation, merger, and coarsening of plumes of various sizes, as seen in Fig. 9. Over time, the plumes reach the bottom of the clearcoat and larger individual circulation cells form. The geometry and position of these cells are approximately steady and the cells have a width on the order of the film height, in agreement with the experiments of [11]. Fluid flow persists within these cells and the circulation helps replenish the evaporative process. The vertical and horizontal fluid velocities of the Marangoni drill and the Bénard cells can be seen in Fig. 18 (middle, bottom).

5.5. Interfacial turbulence

In [18], Sternling and Scriven propose the Marangoni effect as a mechanism for producing interfacial turbulence, i.e., the spontaneous agitation of the interface between two unequilibrated liquids. Their work classifies the various regimes of stability and shows that mass flowing from a fluid of (i) higher viscosity and (ii) lower mass diffusivity across an interface whose surface tension (iii) decreases with respect to the mass concentration produces unstable fluid flow and may produce localized stirring and even droplets without any chemical reactions [19]. These parameters describe the short-wave unstable evaporating Marangoni flows of the previous results and, in this section, the hybrid numerical framework is used to model interfacial turbulence within a three-layer matter cascade. Here, a species originating in the bottom layer flows upwards, passing through multiple interfaces in a Marangoni short-wave unstable fashion.

The first example is shown in Fig. 19. At $t = 0$, the bottom layer contains a species with a mass concentration of $c = 0.2$, while the other two layers do not contain the species. The physical parameters are as in Table 2 and each subsequent layer has half the viscosity and double the mass diffusion coefficient and thickness of the previous layer. For these tests, evaporation is disabled at the top free surface. By 25 seconds into the simulation, the species has transferred from the bottom layer into the middle layer and the Marangoni effect produces the familiar short-wave plume structures in both phases. The species then reaches the next interface between the middle and top layers at around 50 seconds. Here, the Marangoni effect quickly captures and transports the matter tangentially along this interface, creating a "T" shape and a "matter conduit"—i.e., a concentrated region of mass flow—between the two surfaces. The species enters the top layer on the opposite side of the conduit; after which the flow reaches the top interface and Marangoni circulation cells form. After some time, the mass flow through the conduit dies down, and then at time 175 seconds, the process repeats.

In the last example, Fig. 20 shows the results of a three-layer interfacially turbulent matter cascade with viscosity halved from the previous result. The color scheme is chosen to highlight the species mass concentration isocontours. In this example, similar matter junctions form between interfaces, with a higher degree of vorticity and more pronounced "matter turbulence" than in the previous result. This example highlights a cellular structure within the interfacially turbulent Marangoni matter cascade.

6. Concluding remarks

In this work, we developed a multi-physics mathematical model and accompanying high-order numerical framework to study multi-layer coating flow dynamics. These methods were used to study the fluid flow, leveling, and ultimate surface profile of multi-layer automobile paint coatings and to examine interfacial turbulence within a multi-layer matter cascade. Several numerical methods were developed, including: hybrid finite difference level set methods and implicit mesh discontinuous Galerkin methods for capturing sharp-interface multi-phase quasi-Newtonian fluid dynamics, making use of state-of-the-art fast multigrid Stokes solvers; local

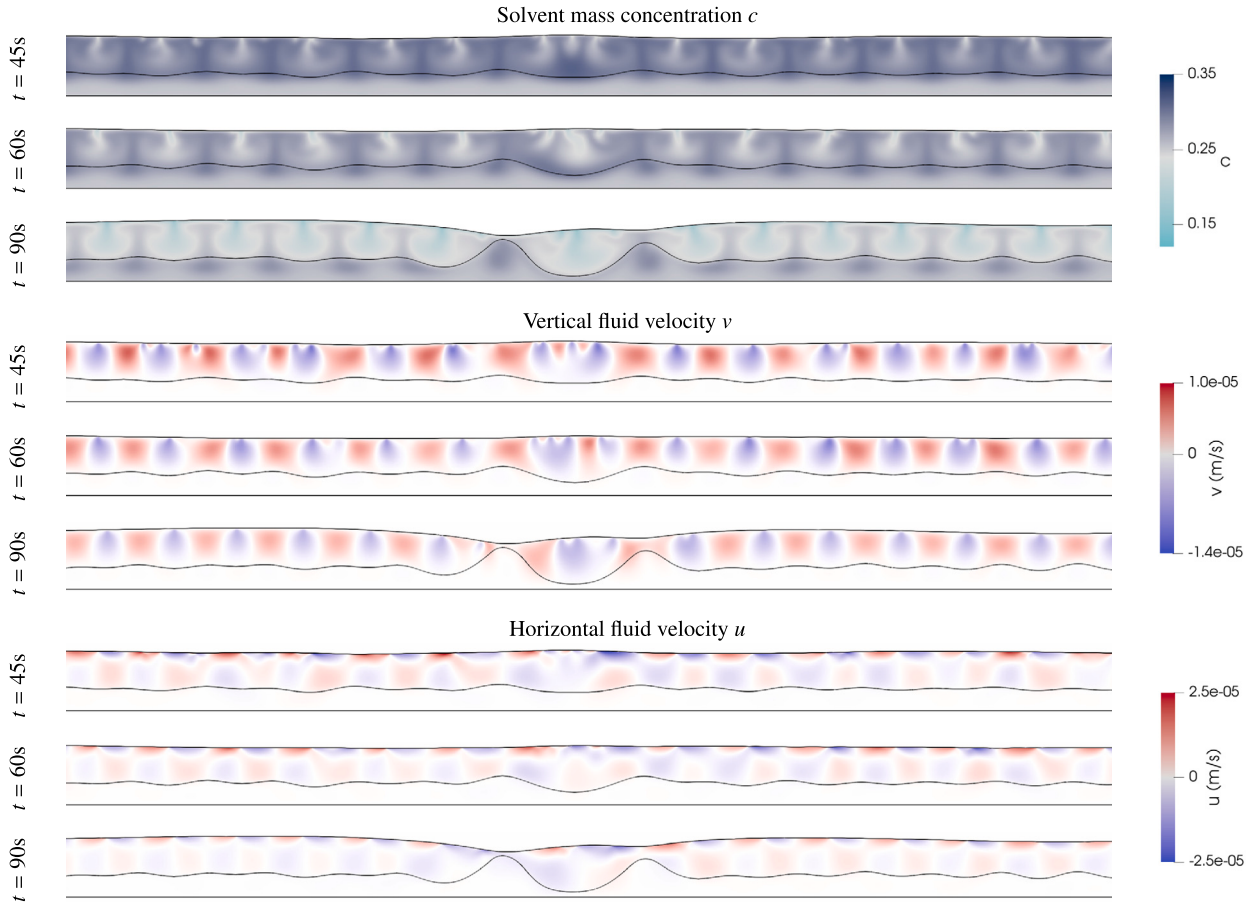


Fig. 18. Short-wave instability causing deformation of the basecoat. **Top:** A 1.6 mm window of the solvent mass concentration profile at the indicated times. Here two large Marangoni plumes (near center of images) merge together and drill into the basecoat—setting the two surfaces on a collision course. **Middle:** The vertical component of the fluid velocity field within the same window at the same times. Blue represents a downward negative velocity and red an upward positive velocity. **Bottom:** The horizontal component of the velocity field, which has an almost lattice structure within the clearcoat. The blue indicates a leftwards velocity and the red rightwards. The values of the solvent mass concentration and fluid velocity are indicated by the color bars, while the curved black lines indicate the paint-gas and embedded paint-paint interfaces.

discontinuous Galerkin methods for Poisson problems with Robin boundary conditions on implicitly-defined curved domains, to capture solvent evaporation; and a tailored finite difference projection algorithm for computing surface gradients within Marangoni stress calculations. Individually, the components range in orders of accuracy, from first-order mixed explicit-implicit time stepping methods, chosen for simplicity, to arbitrarily-high order accurate fluid dynamics via high-order DG methods; our particular choices led to an overall framework which is 2^{nd} order accurate in space and 1^{st} order in time. In a typical simulation, 5–10% of the computational time is spent constructing the implicitly-defined meshes (e.g., computing quadrature schemes and building element connectivity), 25% in building LDG operators (e.g., discrete gradient and variable-viscosity Laplacian operators), 60–70% in multigrid Stokes solvers, with the remaining portion spent in cheaper computations such as advection and state transfer. We parallelized the entire framework using standard domain decomposition together with MPI. Small-scale 2D simulations can be run on a laptop, while medium-scale 2D simulations over long-and-thin domains can be run on a modest multicore system; as example, our largest-sized 2D simulations required a few days of compute time using 8 nodes of NERSC's *Cori* (256 cores). Three-dimensional simulations naturally require larger resources; e.g., the simulation of section 5.1.1 took four days with similar computing resources. As discussed in section 3.4.3, the main impediment is the need for hundreds of thousands of time steps required to simulate over the long time scales associated with drying paint.

Many avenues are available for future work on multi-layer coatings. Producing a smooth defect-free film of paint is of utmost importance to the automobile industry, therefore, a study on the formation of holes within multi-layer coatings may be considered. The drilling phenomenon observed in this work is one possible explanation of hole creation, wherein the paint-paint and paint-gas surfaces intersect and expose the basecoats to air. Toward this end, one could generalize the numerical framework to allow the tracking of multiple intersecting paint-paint films, including contact lines and triple point motion. One possibility is to replace the use of height functions with a more general approach, such as the Voronoi implicit interface method (VIIM) [79]. Along the same lines, the finite difference algorithm for surface gradient and Marangoni stress calculations would need to be extended to more

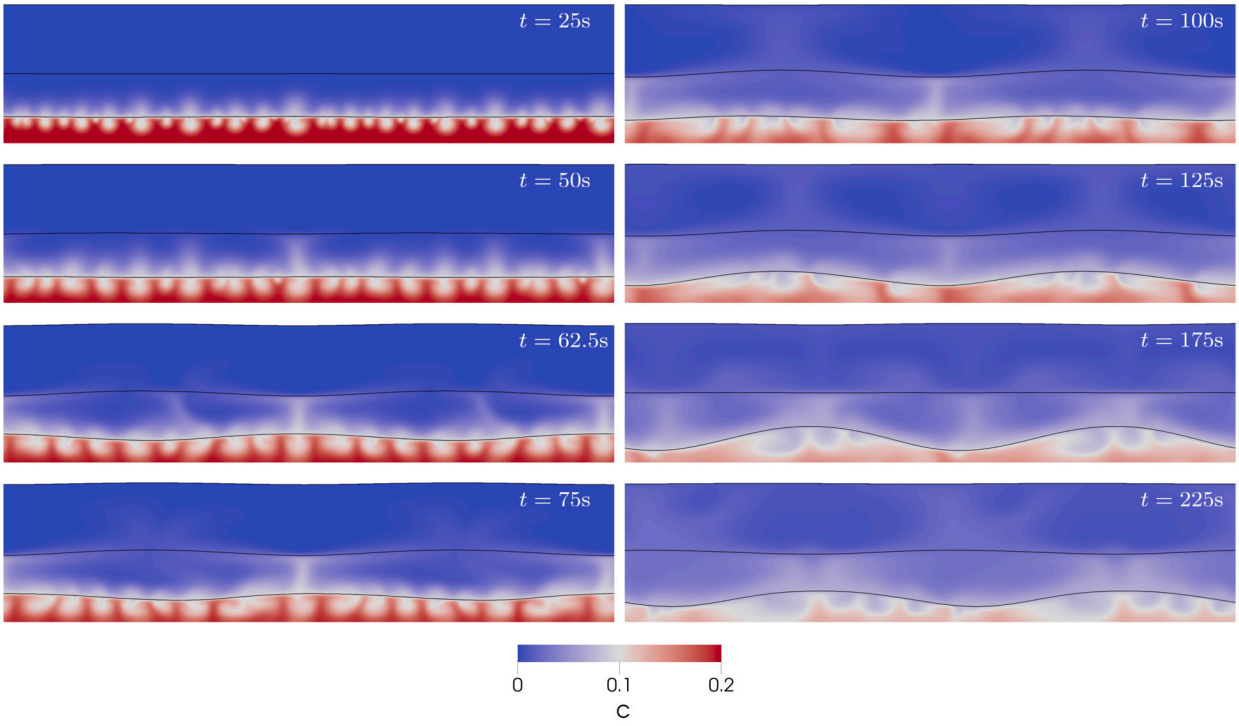


Fig. 19. An interfacially turbulent matter cascade. Illustrated is the evolution and transfer of species between three fluid layers, starting from the bottom layer, at the indicated times. The Marangoni effect first generates the familiar plume structures in both the bottom and middle layers. Then, mass in the middle layer is captured and transported by the Marangoni stresses of the second interface, after which it enters the top layer. The process then repeats until the matter is exhausted. The 400 μm × 200 μm computational domain is repeated across its periodic axis for presentation. The color bar indicates the species mass concentration and the curved black lines (three in each panel) indicate the free surface and embedded interfaces.

intricate geometries in this setting. Additionally, the motion of contact lines between a paint surface and a solid wall may necessitate the use of slip models [80,81]. Towards this objective, a high-order accurate LDG method for the Stokes equations with Navier-slip boundary conditions on implicitly-defined curved domains is presented in the appendix of [3].

Numerous additional physical effects could be included in future work and some model assumptions reconsidered. For example, the model in this paper is purely isothermal; also, the interfaces do not carry solvent nor mass. At a first approximation, the effects of temperature could be incorporated by means of a quasi-thermodynamic model that slowly varies the rheological properties as a function of time. Meanwhile, dynamics constrained to the paint-paint and paint-gas interfaces, such as the motion and transfer of soluble or insoluble surfactants, could be incorporated via surface PDEs coupled to the volumetric flow dynamics. Doing so would allow for further fine-scaled studies of the intricate nature of the thin solutal boundary layer dynamics in evaporating Marangoni flows. Additionally, further expansion of the domain size in 3D studies, which may require anisotropic meshes, will allow for capturing long-wave oscillatory Marangoni modes and provide greater insight into the nature of multi-layer coatings.

CRediT authorship contribution statement

Luke P. Corcos: Conceptualization, Formal analysis, Methodology, Validation, Visualization, Writing – original draft, Writing – review & editing. **Robert I. Saye:** Conceptualization, Formal analysis, Methodology, Validation, Visualization, Writing – original draft, Writing – review & editing. **James A. Sethian:** Conceptualization, Formal analysis, Methodology, Validation, Visualization, Writing – original draft, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

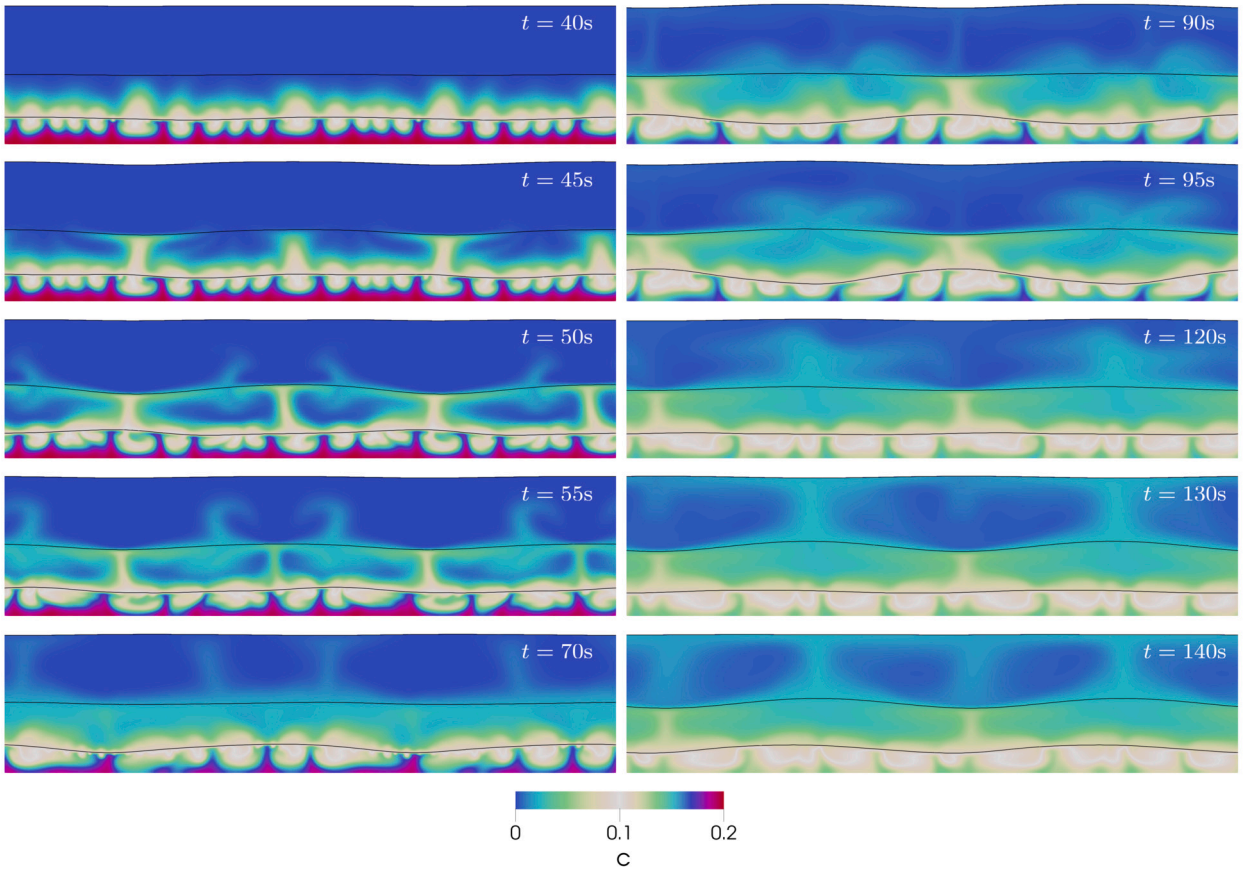


Fig. 20. A three-layer interfacially turbulent matter cascade. The 400 μm × 200 μm computational domain is repeated across its periodic axis for presentation. The color bar indicates the species mass concentration and the curved black lines (three in each panel) indicate the free surface and embedded interfaces.

Acknowledgements

We would like to thank our collaborators at PPG Industries, Inc., including Reza Rock, Brandon Petrouskie, and Xinyu Lu, for introducing us to the coating flow problem and for many intriguing discussions. Through our collaboration we uncovered many facets of "watching paint dry" that are far more complex and nuanced than watching paint dry. This research was supported in part by the Applied Mathematics Program of the US Department of Energy (DOE) Office of Advanced Scientific Computing Research under contract number DE-AC02-05CH11231, by a DOE Office of Science Early Career Research Program award, and by the High Performance Computing for Manufacturing program sponsored by the US DOE Advanced Manufacturing Office of the Energy Efficiency and Renewable Energy Office under Contract DE-AC02-05CH11231 and a Cooperative Research and Development Agreement (CRADA) number FP00011585. Some computations used resources of the National Energy Research Scientific Computing Center, a US Department of Energy Office of Science User Facility at Lawrence Berkeley National Laboratory, operated under Contract No. DE-AC02-05CH11231.

Appendix A. LDG methods for Poisson problems with Robin boundary conditions

A key driving force in the multi-layer coating flow problem is mass transfer at the free evaporative surface Γ_e, with the evaporation process described by the Robin boundary condition (8). In particular, the mixed explicit-implicit time stepping method leads to a heat operator problem (13) with boundary conditions of Robin type. In this section, we present the derivation of a local discontinuous Galerkin method (LDG) [70,71] for Poisson problems with Robin boundary conditions, of which (13) is a simple extension. The resulting discretization has several favorable properties: for example, it is optimally high-order accurate and the final linear system is symmetric positive definite; moreover, the LDG discretization is amenable to straightforward and fast multigrid solvers.

Consider a d-dimensional domain $\Omega \subset \mathbb{R}^d$ with Robin boundary conditions applied on the domain boundary, denoted as Γ_R. We wish to find $u : \Omega \to \mathbb{R}$ such that

$$\begin{cases} -\nabla \cdot (\mu \nabla u) = f & \text{in } \Omega \\ au + (b\mu \nabla u) \cdot \boldsymbol{n} = r & \text{on } \Gamma_R, \end{cases} \tag{A.1}$$

where $\mu, a, b : \mathbb{R}^d \to \mathbb{R}^+$ are functions mapping onto the space of positive real numbers. Here μ takes the role of the diffusion coefficient, whereas a, b are the Robin coefficients. We now present the LDG method for (A.1) in four steps, in a manner similar to [64] and outlined in Table 1, ensuring that the choice of numerical fluxes is consistent with the PDE and Robin boundary conditions. Along mesh faces, define the jump $[u] = u^- - u^+$ to align with the normal vector, where u^- and u^+ represent the trace of the DG polynomials restricted from the left and right elements respectively. Along boundary faces, define u^- to represent the trace of the DG polynomials from the interior element.

(i) Define auxiliary variable $\boldsymbol{\eta} \in V_h^d$ such that $\boldsymbol{\eta} = \nabla u$ weakly on element $E \in \mathcal{E}$ via the strong-weak form[10]:

$$\int_E \boldsymbol{\eta} \cdot \boldsymbol{\omega} = \int_E \nabla u \cdot \boldsymbol{\omega} + \int_{\partial E} (u^* - u)\boldsymbol{\omega} \cdot \boldsymbol{n},$$

for all test functions $\boldsymbol{\omega}$ on element E. Here u^* is the *numerical flux* of u, which depends on the traces of the DG polynomials and on boundary or jump data. Summing over all elements gives

$$\int_\Omega \boldsymbol{\eta} \cdot \boldsymbol{\omega} = \sum_E \int_E \nabla u \cdot \boldsymbol{\omega} + \int_{\Gamma_0} (u^* - u^-)\boldsymbol{\omega}^- \cdot \boldsymbol{n} - \int_{\Gamma_0} (u^* - u^+)\boldsymbol{\omega}^+ \cdot \boldsymbol{n} + \int_{\Gamma_R} (u^* - u^-)\boldsymbol{\omega}^- \cdot \boldsymbol{n}. \tag{A.2}$$

In LDG methods, the numerical fluxes for the collection of intraphase faces Γ_0 are often one-sided [70,71]. In this work, the intraphase numerical flux for u is taken from the trace of the left elements' polynomials, and the numerical flux u^* for the Robin boundary is taken from the trace of the boundary elements' polynomials, setting $u^* = u^-$ for both types of faces. Plugging this numerical flux into (A.2) eliminates the Robin boundary's contribution to the gradient and gives

$$\int_\Omega \boldsymbol{\eta} \cdot \boldsymbol{\omega} = \sum_E \int_E \nabla u \cdot \boldsymbol{\omega} + \int_{\Gamma_0} (u^+ - u^-)\boldsymbol{\omega}^+ \cdot \boldsymbol{n}.$$

Define the *broken gradient* operator $\nabla_h : V_h \to V_h^d$ and the *lifting* operator $L : V_h \to V_h^d$ such that, respectively,

$$\int_\Omega \nabla_h u \cdot \boldsymbol{\omega} = \sum_E \int_E \nabla u \cdot \boldsymbol{\omega}, \qquad \int_\Omega Lu \cdot \boldsymbol{\omega} = \int_{\Gamma_0} (u^+ - u^-)\boldsymbol{\omega}^+ \cdot \boldsymbol{n},$$

for all $\boldsymbol{\omega} \in V_h^d$. The broken gradient operator defines the gradient of polynomial $u \in V_h$ on an element-wise basis while the lifting operator incorporates the jumps in u across faces in each dimension, effectively taking the surface integrals and "lifting" them into the domain. Now, define the *gradient* operator $G : V_h \to V_h^d$ as $G = \nabla_h + L$. Thus, $\boldsymbol{\eta} = Gu$.

(ii) Now define $\boldsymbol{q} \in V_h^d$ as the L^2 projection of $\mu\boldsymbol{\eta}$. In terms of DG operators, this is equivalent to $\boldsymbol{q} = M^{-1} M_\mu \boldsymbol{\eta}$, where M is the mass matrix and M_μ is the μ-weighted mass matrix such that $v^T M_\mu u = \int_\Omega v\mu u$, for all $u, v \in V_h$. So the k-th component of $\boldsymbol{q}$ is equal to

$$q_k = M^{-1} M_\mu G_k u. \tag{A.3}$$

(iii) Next we seek to define the divergence of $\boldsymbol{q}$. For element $E \in \mathcal{E}$, let $w \in V_h$ be such that $w = \nabla \cdot \boldsymbol{q}$ weakly via the weak-weak form

$$\int_E wv = -\int_E \boldsymbol{q} \cdot \nabla v + \int_{\partial E} v\boldsymbol{q}^* \cdot \boldsymbol{n},$$

for all test functions v on element E, with $\boldsymbol{q}^*$ being the numerical flux of $\boldsymbol{q}$. Summing over all elements gives

$$\int_\Omega wv = -\sum_E \int_E \boldsymbol{q} \cdot \nabla v + \int_{\Gamma_0} (v^- - v^+)\boldsymbol{q}^* \cdot \boldsymbol{n} + \int_{\Gamma_R} v^- \boldsymbol{q}^* \cdot \boldsymbol{n}. \tag{A.4}$$

For the collection of intraphase elements Γ_0, define the numerical flux for $\boldsymbol{q}$ in the opposite direction as that for u, while for the boundary faces we incorporate the Robin boundary condition (A.1), as follows:

$$\boldsymbol{q}^* = \begin{cases} \boldsymbol{q}^+ & \text{on } \Gamma_0 \\ \frac{1}{b}(r - au^-)\boldsymbol{n} & \text{on } \Gamma_R. \end{cases}$$

Plugging the numerical fluxes into (A.4) gives

$$\int_\Omega wv = -\sum_E \int_E \boldsymbol{q} \cdot \nabla v - \int_{\Gamma_0} (v^+ - v^-)\boldsymbol{q}^+ \cdot \boldsymbol{n} + \int_{\Gamma_R} v^- \frac{1}{b} r - \int_{\Gamma_R} v^- \frac{1}{b} au^-. \tag{A.5}$$

[10] For brevity, we drop the measure of integration, which should be clear from the specified domain of integration, i.e., volume or surface.

We now break (A.5) into three components. Notice that the first two terms are equivalent to $-(\nabla_h v, \boldsymbol{q}) - (Lv, \boldsymbol{q}) = -(Gv, \boldsymbol{q})$, where $(\cdot,\cdot)$ is the standard inner product. Secondly, define $J_R \in V_h$ such that

$$\int_\Omega J_R(r) v = \int_{\Gamma_R} v^- \frac{1}{b} r,$$

for all $v \in V_h$. Now define block-sparse matrix $A_R(u)$ such that

$$v^T A_R u = \int_{\Gamma_R} v^- \frac{1}{b} a u^-,$$

for all $v \in V_h$. Note the Robin term A_R is akin to penalization methods for weakly imposing Dirichlet boundary conditions in finite element methods [82,83]. Combining these terms gives the following weak definition of $\nabla \cdot \boldsymbol{q}$ incorporating the Robin boundary data:

$$w = -\sum_k M^{-1} G_k^T M q_k - M^{-1} A_R(u) + J_R.$$

Plugging in the definition of $\boldsymbol{q}$ (A.3) gives:

$$w = -\sum_k M^{-1} G_k^T M_\mu G_k u - M^{-1} A_R(u) + J_R.$$

(iv) Additionally, to ensure wellposedness of the discrete problem, penalty stabilization is employed within the LDG framework to weakly impose continuity between intraphase elements, adding a block-sparse penalty matrix E_u of the form $v^T E_u u = \int_{\Gamma_0} \tau_0 [u][v]$ to the discrete Laplacian, where τ_0 is a positive penalty parameter. In general for Poisson problems, penalty parameters should vary in proportion with the diffusion coefficient μ and in inverse proportion to the mesh spacing h. Additionally, the penalty parameter is scaled with respect to polynomial degree p. For the convergence tests of section 4.1, the penalty parameter is set to $\tau_0 = 0.5\mu p/h$ and in simulations of multi-layer coating flows, a higher degree of penalty stabilization is employed, with $\tau_0 = \alpha\mu p/h$ and α ranging from 15-30.

Summary: the Poisson problem with Robin boundary conditions (A.1) is solved via LDG by finding $u \in V_h$ such that the following block-sparse matrix equation holds:

$$\left(\sum_k G_k^T M_\mu G_k + A_R + E_u \right) u = M \mathbb{P}_{V_h}(f) + M J_R, \tag{A.6}$$

where $\mathbb{P}_{V_h}(f)$ is the L^2 projection of f onto V_h. Note that both sides of the equation have been multiplied by the mass matrix M. This results in a symmetric positive-definite linear system when the Robin coefficients are positive. The linear system (A.6) is solved by the fast operator coarsening multigrid methods of [72]. We note that incorporating the Robin boundary condition into the Poisson problem within the operator coarsening multigrid paradigm requires only the added coarsening of the Robin operator A_R, which is coarsened in a manner similar to the penalty operators. The LDG methods developed in this section are demonstrated to be optimally high-order accurate in section 4.1. Rapid multigrid performance is also demonstrated in which the iteration count remains bounded as the background mesh is refined.

References

[1] R.I. Saye, J.A. Sethian, B. Petrouskie, A. Zatorsky, X. Lu, R. Rock, Insights from high-fidelity modeling of industrial rotary bell atomization, Proc. Natl. Acad. Sci. 120 (2023).

[2] T. Pennington, Waterborne painting process is a first at South Carolina BMW plant, https://www.pfonline.com/articles/waterborne-painting-process-is-a-first-at-south-carolina-bmw-plant, 2013.

[3] L.P. Corcos, A Hybrid Finite Difference Level Set–Implicit Mesh Discontinuous Galerkin Method for Multi-Layer Coating Flow Problems, Ph.D. thesis, University of California Berkeley, 2023.

[4] J. Thomson, On certain curious motions observable at the surfaces of wine and other alcoholic liquors, Lond. Edinb. Dublin Philos. Mag. J. Sci. 10 (1855) 330–333.

[5] C. Marangoni, Ueber die Ausbreitung der Tropfen einer Flüssigkeit auf der Oberfläche einer anderen, Ann. Phys. 219 (1871) 337–354.

[6] D. Schwabe, A. Scharmann, Some evidence for the existence and magnitude of a critical Marangoni number for the onset of oscillatory flow in crystal growth melts, J. Cryst. Growth 46 (1979) 125–131.

[7] M.F. Schatz, G.P. Neitzel, Experiments on thermocapillary instabilities, Annu. Rev. Fluid Mech. 33 (2001) 93–127.

[8] D. Lohse, Fundamental fluid dynamics challenges in inkjet printing, Annu. Rev. Fluid Mech. 54 (2022) 349–382.

[9] J. Ratulowski, H.-C. Chang, Marangoni effects of trace impurities on the motion of long gas bubbles in capillaries, J. Fluid Mech. 210 (1990) 303–328.

[10] A. Abbasian, S.R. Ghaffarian, N. Mohammadi, M.R. Khosroshahi, M. Fathollahi, Study on different planforms of paint's solvents and the effect of surfactants (on them), Prog. Org. Coat. 49 (2004) 229–235.

[11] N. Bassou, Y. Rharbi, Role of Bénard-Marangoni instabilities during solvent evaporation in polymer surface corrugations, Langmuir 25 (2009) 624–632.

[12] R.V. Craster, O.K. Matar, Dynamics and stability of thin liquid films, Rev. Mod. Phys. 81 (2009) 1131–1198.

[13] A. Oron, S.H. Davis, S.G. Bankoff, Long-scale evolution of thin liquid films, Rev. Mod. Phys. 69 (1997) 931–980.

[14] H. Bénard, Les tourbillons cellulaires dans une nappe liquide. - Méthodes optiques d'observation et d'enregistrement, J. Phys. Theor. Appl. 10 (1901) 254–266, https://doi.org/10.1051/jphystap:0190100100025400.

[15] M.J. Block, Surface tension as the cause of Bénard cells and surface deformation in a liquid film, Nature 178 (1956) 650–651.

[16] J.R.A. Pearson, On convection cells induced by surface tension, J. Fluid Mech. 4 (1958) 489–500.
[17] L.E. Scriven, C.V. Sternling, The Marangoni effects, Nature 187 (1960) 186–188.
[18] C.V. Sternling, L.E. Scriven, Interfacial turbulence: hydrodynamic instability and the Marangoni effect, AIChE J. 5 (1959) 514–523.
[19] T. Sherwood, J. Wei, Interfacial phenomena in liquid extraction, Ind. Eng. Chem. 49 (1957) 1030–1034.
[20] I. Langmuir, D.B. Langmuir, The effect of monomolecular films on the evaporation of ether solutions, J. Phys. Chem. 31 (1927) 1719–1731.
[21] L.E. Scriven, C.V. Sternling, On cellular convection driven by surface-tension gradients: effects of mean surface tension and surface viscosity, J. Fluid Mech. 19 (1964) 321–340.
[22] S.J. Vanhook, M.F. Schatz, J.B. Swift, W.D. Mccormick, H.L. Swinney, Long-wavelength surface-tension-driven Bénard convection: experiment and theory, J. Fluid Mech. 345 (1997) 45–78.
[23] W.S. Overdiep, The levelling of paints, Prog. Org. Coat. 14 (1986) 159–175.
[24] S.D. Howison, J.A. Moriarty, J.R. Ockendon, E.L. Terrill, S.K. Wilson, A mathematical model for drying paint layers, J. Eng. Math. 32 (1997) 377–394.
[25] M.H. Eres, D.E. Weidner, L.W. Schwartz, Three-dimensional direct numerical simulation of surface-tension-gradient effects on the leveling of an evaporating multicomponent fluid, Langmuir 15 (1999) 1859–1871.
[26] P. de Gennes, Instabilities during the evaporation of a film: non-glassy polymer + volatile solvent, Eur. Phys. J. E 6 (2001) 421–424.
[27] S. Kojima, T. Moriga, K. Takenouchi, The leveling of thermosetting waterborne coatings. Part I: observation of leveling process, Polym. Eng. Sci. 33 (1993) 1320–1328.
[28] K.E. Strawhecker, S.K. Kumar, J.F. Douglas, A. Karim, The critical role of solvent evaporation on the roughness of spin-cast polymer films, Macromolecules 34 (2001) 4669–4672.
[29] D.J. Taylor, D.P. Birnie, A case study in striation prevention by targeted formulation adjustment: aluminum titanate sol-gel coatings, Chem. Mater. 14 (2002) 1488–1492.
[30] E. Bodenschatz, W. Pesch, G. Ahlers, Recent developments in Rayleigh-Bénard convection, Annu. Rev. Fluid Mech. 32 (2000) 709–778.
[31] S.H. Davis, Thermocapillary instabilities, Annu. Rev. Fluid Mech. 19 (1987) 403–435.
[32] A.F. Routh, Drying of thin colloidal films, Rep. Prog. Phys. 76 (2013) 046603.
[33] E. Bänsch, S. Basting, R. Krahl, Numerical simulation of two-phase flows with heat and mass transfer, Discrete Contin. Dyn. Syst. 35 (2015) 2325.
[34] J. Donea, S. Giuliani, J.P. Halleux, An arbitrary Lagrangian-Eulerian finite element method for transient dynamic fluid-structure interactions, Comput. Methods Appl. Mech. Eng. 33 (1982) 689–723.
[35] S.G. Yiantsios, S.K. Serpetsi, F. Doumenc, B. Guerrier, Surface deformation and film corrugation during drying of polymer solutions induced by Marangoni phenomena, Int. J. Heat Mass Transf. 89 (2015) 1083–1094.
[36] S. Osher, J.A. Sethian, Fronts propagating with curvature-dependent speed: algorithms based on Hamilton-Jacobi formulations, J. Comput. Phys. 79 (1988) 12–49.
[37] J.A. Sethian, Level Set Methods and Fast Marching Methods: Evolving Interfaces in Geometry, Fluid Mechanics, Computer Vision, and Materials Sciences, Cambridge University Press, 1999.
[38] R.P. Fedkiw, T. Aslam, B. Merriman, S. Osher, A non-oscillatory Eulerian approach to interfaces in multimaterial flows (the ghost fluid method), J. Comput. Phys. 152 (1999) 457–492.
[39] S. Tanguy, T. Ménard, A. Berlemont, A level set method for vaporizing two-phase flows, J. Comput. Phys. 221 (2007) 837–853.
[40] L. Rueda Villegas, R. Alis, M. Lepilliez, S. Tanguy, A ghost fluid/level set method for boiling flows and liquid evaporation: application to the Leidenfrost effect, J. Comput. Phys. 316 (2016) 789–813.
[41] J. Papac, F. Gibou, C. Ratsch, Efficient symmetric discretization for the Poisson, heat and Stefan-type problems with Robin boundary conditions, J. Comput. Phys. 229 (2010) 875–889.
[42] D. Bochkov, F. Gibou, Solving Poisson-type equations with Robin boundary conditions on piecewise smooth interfaces, J. Comput. Phys. 376 (2019) 1156–1198.
[43] K. Luo, C. Shao, M. Chai, J. Fan, Level set method for atomization and evaporation simulations, Prog. Energy Combust. Sci. 73 (2019) 65–94.
[44] M. Chai, K. Luo, C. Shao, H. Wang, J. Fan, A finite difference discretization method for heat and mass transfer with Robin boundary conditions on irregular domains, J. Comput. Phys. 400 (2020) 108890.
[45] M. Chai, K. Luo, C. Shao, H. Wang, J. Fan, Interface-resolved detailed numerical simulation of evaporating two-phase flows with Robin boundary conditions on irregular domains, Int. J. Heat Mass Transf. 145 (2019) 118774.
[46] R. Bhardwaj, X. Fang, D. Attinger, Pattern formation during the evaporation of a colloidal nanoliter drop: a numerical and experimental study, New J. Phys. 11 (2009) 075020.
[47] F. Girard, M. Antoni, S. Faure, A. Steinchen, Evaporation and Marangoni driven convection in small heated water droplets, Langmuir 22 (2006) 11085–11091.
[48] H. Hu, R.G. Larson, Analysis of the effects of Marangoni stresses on the microflow in an evaporating sessile droplet, Langmuir 21 (2005) 3972–3980.
[49] O.E. Ruiz, W.Z. Black, Evaporation of water droplets placed on a heated horizontal surface, J. Heat Transf. 124 (2002) 854–863.
[50] G.P. Sasmal, J.I. Hochstein, Marangoni convection with a curved and deforming free surface in a cavity, J. Fluids Eng. 116 (1994) 577–582.
[51] M. Francois, J. Sicilian, D. Kothe, Modeling of thermocapillary forces within a volume tracking algorithm, in: Modeling of Casting, Welding and Advanced Solidification Processes–XI, Opio, France, May, 2006, pp. 935–942.
[52] G. Son, A sharp-interface level-set method for analysis of Marangoni effect on microdroplet evaporation, Int. Commun. Heat Mass Transf. 58 (2014) 156–165.
[53] J.-J. Xu, Y. Yang, J. Lowengrub, A level-set continuum method for two-phase flows with insoluble surfactant, J. Comput. Phys. 231 (2012) 5897–5909.
[54] J.-J. Xu, W. Shi, M.-C. Lai, A level-set method for two-phase flows with soluble surfactant, J. Comput. Phys. 353 (2018) 336–355.
[55] C.C. de Langavant, A. Guittet, M. Theillard, F. Temprano-Coleto, F. Gibou, Level-set simulations of soluble surfactant driven flows, J. Comput. Phys. 348 (2017) 271–297.
[56] W. Huang, J. Iseringhausen, T. Kneiphof, Z. Qu, C. Jiang, M.B. Hullin, Chemomechanical simulation of soap film flow on spherical bubbles, ACM Trans. Graph. 39 (2020) 1–41.
[57] J. Chen, V. Kala, A. Marquez-Razon, E. Gueidon, D.A. Hyde, J. Teran, A momentum-conserving implicit material point method for surface tension with contact angles and spatial gradients, ACM Trans. Graph. 40 (2021) 1–16.
[58] T. Köllner, K. Schwarzenberger, K. Eckert, T. Boeck, Multiscale structures in solutal Marangoni convection: three-dimensional simulations and supporting experiments, Phys. Fluids 25 (2013) 092109.
[59] K. Schwarzenberger, T. Köllner, H. Linde, T. Boeck, S. Odenbach, K. Eckert, Pattern formation and mass transfer under stationary solutal Marangoni instability, Adv. Colloid Interface Sci. 206 (2014) 344–371.
[60] J. Bentley, G.P.A. Turner, Introduction to Paint Chemistry and Principles of Paint Technology, CRC Press, 1997.
[61] D. Adalsteinsson, J.A. Sethian, Transport and diffusion of material quantities on propagating interfaces via level set methods, J. Comput. Phys. 185 (1) (2003) 271–288.
[62] J.A. Sethian, P. Smereka, Level set methods for fluid interfaces, Annu. Rev. Fluid Mech. 35 (2003) 341–372.
[63] R. Saye, Interfacial gauge methods for incompressible fluid dynamics, Sci. Adv. 2 (2016).
[64] R. Saye, Implicit mesh discontinuous Galerkin methods and interfacial gauge methods for high-order accurate interface dynamics, with applications to surface tension dynamics, rigid body fluid–structure interaction, and free surface flow: part I, J. Comput. Phys. 344 (2017) 647–682.
[65] R. Saye, Implicit mesh discontinuous Galerkin methods and interfacial gauge methods for high-order accurate interface dynamics, with applications to surface tension dynamics, rigid body fluid–structure interaction, and free surface flow: part II, J. Comput. Phys. 344 (2017) 683–723.

[66] R. Saye, Fast multigrid solution of high-order accurate multiphase Stokes problems, Commun. Appl. Math. Comput. Sci. 15 (2020) 147–196.
[67] C.-W. Shu, Essentially Non-oscillatory and Weighted Essentially Non-oscillatory Schemes for Hyperbolic Conservation Laws, Springer Berlin Heidelberg, Berlin, Heidelberg, 1998, pp. 325–432.
[68] R. Saye, High-order quadrature methods for implicitly defined surfaces and volumes in hyperrectangles, SIAM J. Sci. Comput. 37 (2015) A993–A1019.
[69] R. Saye, Algoim – algorithms for implicitly defined geometry, level set methods, and Voronoi implicit interface methods, https://algoim.github.io/, 2022.
[70] D.N. Arnold, F. Brezzi, B. Cockburn, L.D. Marini, Unified analysis of discontinuous Galerkin methods for elliptic problems, SIAM J. Numer. Anal. 39 (2002) 1749–1779.
[71] B. Cockburn, C.-W. Shu, The local discontinuous Galerkin method for time-dependent convection-diffusion systems, SIAM J. Numer. Anal. 35 (1998) 2440–2463.
[72] D. Fortunato, C.H. Rycroft, R. Saye, Efficient operator-coarsening multigrid schemes for local discontinuous Galerkin methods, SIAM J. Sci. Comput. 41 (2019) A3913–A3937.
[73] W. Briggs, V. Henson, S. McCormick, A Multigrid Tutorial, 2nd edition, 2000.
[74] Y. Saad, Iterative Methods for Sparse Linear Systems, 2nd edition, 2003.
[75] R. Saye, Efficient multigrid solution of elliptic interface problems using viscosity-upwinded local discontinuous Galerkin methods, Commun. Appl. Math. Comput. Sci. 14 (2019) 247–283.
[76] M. Juntunen, R. Stenberg, Nitsche's method for general boundary conditions, Math. Comput. 78 (2009) 1353–1374.
[77] C.K. Schoff, Craters and other coatings defects: mechanisms and analysis, in: M. Wen, K. Dušek (Eds.), Protective Coatings: Film Formation and Properties, Springer International Publishing, Cham, 2017, pp. 403–425.
[78] Y. Zhong, Y. Zhuo, Z. Wang, Y. Sha, Marangoni convection induced by simultaneous mass and heat transfer during evaporation of n-heptane/ether binary liquid mixture, Int. J. Heat Mass Transf. 108 (2017) 812–821.
[79] R.I. Saye, J.A. Sethian, The Voronoi implicit interface method for computing multiphase physics, Proc. Natl. Acad. Sci. 108 (2011) 19498–19503.
[80] C. Huh, L.E. Scriven, Hydrodynamic model of steady movement of a solid/liquid/fluid contact line, J. Colloid Interface Sci. 35 (1971) 85–101.
[81] V. Dussan, The moving contact line: the slip boundary condition, J. Fluid Mech. 77 (1976) 665–684.
[82] J. Lions, Problemes aux limites non homogenesa donées irrégulieres: Une méthode d'approximation, in: Numerical Analysis of Partial Differential Equations, CIME 2 Ciclo, Ispra, 1967, Edizioni Cremonese, Rome, 1968, pp. 283–292.
[83] J. Nitsche, Über ein Variationsprinzip zur Lösung von Dirichlet-Problemen bei Verwendung von Teilräumen, die keinen Randbedingungen unterworfen sind, in: Abhandlungen aus dem mathematischen Seminar der Universität Hamburg, vol. 36, Springer, 1971, pp. 9–15.