HELSINKI UNIVERSITY OF TECHNOLOGY
Department of Automation and Systems Technology
Control Engineering

**Antti Pohjoranta**

# Modelling surfactant mass balance with the ALE method on deforming 2D surfaces

Thesis for the degree of Licentiate of Science in Technology submitted for inspection, Espoo, June 23rd, 2008.

| | |
|---|---|
| Supervisor: | Professor Robert Tenno, D.Sc. |
| Instructor: | Professor Robert Tenno, D.Sc. |
| Inspector: | Jarmo Ritola, Lic.Sc. (Tech.) |

# Abstract


| | |
|---|---|
| **Author:** | Antti Pohjoranta |
| **Title:** | Modelling surfactant mass balance with the ALE method on deforming 2D surfaces |
| **Date:** | June 23, 2008 |
| **Department:** | Department of Automation and Systems Technology |
| **Supervisor:** | Professor Robert Tenno, D.Sc. |
| **Instructor:** | Professor Robert Tenno, D.Sc. |
| **Inspector:** | Jarmo Ritola, Lic.Sc. (Tech.) |



This thesis describes simple and computationally light, yet considerably accurate, means to model the mass balance of soluble and insoluble, mobile and non-mobile surfactants, acting on a shape-changing surface. The presented mechanisms are applied using the finite element method and the arbitrary Lagrange-Eulerian (ALE) method. The discussion is restricted to two dimensional formulations of plane-symmetric and axially symmetric geometries.

A brief introduction to physical systems where surfactant mass balance computation is needed is given first. The introduction is followed by a presentation of physical equations required to model surfactant mass transfer dynamically on a shape-changing surface. Based on the physical equations, two different approaches to numerically model surfactant mass balance are formulated as well as implemented computationally and evaluated based on simulation. A practical application part puts the developed means for surfactant mass balance computation into use by modelling an electrochemical deposition process in various practical cases, one of which is also compared against empirical data. A discussion on modelling considerations related to the temporal and spatial dimensions of the system at hand is given in the end of the thesis.



Keywords: surfactant modelling, microvia filling, ALE method

# Tiivistelmä


| | |
|---|---|
| **Tekijä:** | Antti Pohjoranta |
| **Otsikko:** | Pinta-aktiivisten aineiden massatasemallintaminen ALE-menetelmään perustuen, muuttuvilla kaksiulotteisilla pinnoilla |
| **Pvm:** | 23.06.2008 |
| **Laitos:** | Automaatio- ja systeemitekniikan laitos |
| **Valvoja:** | Professori Robert Tenno, D.Sc. |
| **Ohjaaja:** | Professori Robert Tenno, D.Sc. |
| **Tarkastaja:** | Jarmo Ritola, TkL |



Tässä työssä kuvataan kaksi yksinkertaista ja laskennallisesti kevyttä, mutta tarkkaa mekanismia liukenevien, liukenemattomien, liikkuvien ja ei-liikkuvien pinta-aktiivisten aineiden massataseen laskemiseen muotoaan muuttavalla kaksiulotteisella pinnalla. Mekanismit perustuvat elementtimenetelmän sekä nk. arbitrary Lagrange-Eulerian (ALE) -menetelmän soveltamiseen. Mekanismien esitys on rajattu formulaatioihin kaksiulotteisilla tasosymmetrisillä ja pyörähdyssymmetrisillä pinnoilla.

Työssä esitellään aluksi lyhyesti fysikaalisia tilanteita, joissa pinta-aktiivisten aineiden massataseen laskeminen on tarpeen. Seuraavaksi kuvataan oleellisimmat em. massataseen laskennassa tarvittavat fysikaaliset yhtälöt ja fysikaalisten yhtälöiden pohjalta muotoillaan kaksi erilaista formulaatiota yhtälöiden numeerista ratkaisemista varten. Saadut formulaatiot toteutetaan laskennallisesti ja toteutuksia arvioidaan simulaatioon perustuen. Työssä tutkittuja laskentamekanismeja sovelletaan käytännön ongelmien ratkaisemiseen laskemalla pinta-aktiivisten aineiden massatase sähkökemiallisen pinnnoitusprosessin prosessimallissa, lukuisissa käytännön tilanteissa. Yhtä malleista verrataan kokeellista dataa vasten. Soveltavan osion päättää mallinnettavan järjestelmän ajallisiin ja avaruudellisiin mittasuhteisiin liittyviä näkökulmia esiin tuova osa.



Avainsanat: surfactant modelling, microvia filling, ALE method

# Preface

This thesis is based on work done as a part of the VIATON research project, a cooperation of the Helsinki University of Technology (TKK) Control Engineering group, the Aspocomp-Meadville industry partner and the Finnish Funding Agency for Technology and Innovation (TEKES). The project aims to create sophisticated means for output quality control for the microvia filling process applied in production of modern multilayered printed circuit boards. The goal is, modestly put, ambitious.

Within the mentioned organizations, professor **Heikki Koivo**, head of TKK Control Engineering and Mr. **Timo Närhi** as well as Ms. **Tarja Rapala-Virtanen** at Aspocomp-Meadville are the excellent people enabling this co-operative research work between the industry partner and the university. The ever-generous **Finnish tax payer** is the power behind TEKES.

A special acknowledgement goes to Mr. **Jarmo Ritola** at COMSOL Oy for spending his time and energy to inspect and to discuss the thesis text and content, after merely being asked to do so. A wonderful effort I am indeed grateful for.

However, most of all I would like to point gratitude to my instructor, Professor **Robert Tenno**. His effort on guiding the writing work as well as on giving inspiring ideas for the study itself was, and is, invaluable, totally. Thanks Robert.

On a field trip, the journey is way more important than ever returning to the class room. This one was a rough ride, but well worth it. Though the thesis surely still has its mistakes and shortcomings, most of what there was to find is now stowed on board and the rest can rot on the bookshelf. A writing work never finishes itself unless one at some point puts a period to it.

Over the course of this work I got to rediscover something I evidently had lost since my previous degree work, namely, the power and energy concealed in something mysterious called *inspiration*. This was not least thanks to, but my friends and family, the acknowledged people above and the whole crew at TKK Control Engineering. No authority at a university, or at any organization relying on its members' mental effort for that matter, should ever belittle, let alone forget, the enormous positive force embodied in an enthusiastic mind.

# Contents

# Abbreviations and symbols

| | |
|---|---|
| ALE | Arbitrary Lagrange-Eulerian |
| CEAC | Curvature enhanced accelerator coverage |
| FEM | Finite element method |
| MLB | Multilayered printed circuit board |
| ODE | Ordinary differential equation |
| PCB | Printed circuit board |
| PDE | Partial differential equation |

| | |
|---|---|
| $A$ | Area of local area element on cathode surface, $\mathrm{m}^2$ |
| $\alpha_i$ | Apparent transfer coefficient of process $i$ |
| $\beta$ | Harmonic mean apparent transfer coefficient |
| $c_i$ | Concentration of species $i$, $\mathrm{mol/m}^3$ |
| $c_{i,b}$ | Concentration of species $i$ in the solution bulk, $\mathrm{mol/m}^3$ |
| $D_i$ | Diffusion coefficient of species $i$, $\mathrm{m}^2/\mathrm{s}$ |
| $D_i^s$ | Surface diffusion coefficient of species $i$, $\mathrm{m}^2/\mathrm{s}$ |
| $dA$, $ds$, $dp$ | Infinitesimal segment on model element of area, line and point respectively |
| $F$ | Faraday's constant, 96485 As/mol |
| $f$ | A variable defined everywhere in the modelling domain |
| $\varphi$ | Electric potential, V |
| $\Gamma_i$ | Surface concentration of species $i$, $\mathrm{mol/m}^2$ |
| $\Gamma_i^{sat}$ | Surface saturation concentration of species $i$, $\mathrm{mol/m}^2$ |
| $i_e$ | Current density on electrode $e$, $\mathrm{A/m}^2$ |
| $i_0$ | Exchange current density, $\mathrm{A/m}^2$ |
| $k$, $k_i$ | Constant scaling coefficient |

| | |
|---|---|
| $M_i$ | Molecular weight of species $i$, g/mol |
| $\mu_i$ | Chemical term related to species $i$ |
| $N_i$ | Material flux of species $i$, mol/m$^2$/s |
| $\mathbf{n}$ | The mesh boundary outward normal vector (surface normal vector) |
| $n_X, n_Y, n_x, n_y$ | Normal vector components |
| $\Omega$ | Modelling domain |
| $\partial\Omega$ | Modelling domain boundary |
| $R$ | Ideal gas coefficient, 8.314 J/mol/K |
| $r$ | Radius, m |
| $\rho_i$ | Density of $i$, kg/m$^3$ |
| $\sigma$ | Conductivity (of electrolyte), S/m |
| $T$ | Temperature, K or C$^\circ$ |
| $t$ | Time, s |
| $\mathbf{t}$ | Mesh boundary tangent vector (surface tangent vector) |
| $t_X, t_Y, t_x, t_y$ | Tangent vector components |
| $\theta_i$ | Proportional surface coverage of species $i$ |
| $u_i$ | Mobility of $i$, mol·s/kg |
| $V$ | Volume, m$^3$ |
| $\mathbf{v}$ | Mesh movement velocity (vector), m/s |
| $v_i$ | Mesh movement velocity in direction of coordinate $i$, m/s |
| $X$ | The horizontal (Cartesian) or radial (cylindrical) coordinate in the fixed coordinate system |
| $x$ | The horizontal (Cartesian) or radial (cylindrical) in the moving coordinate system |
| $Y$ | The vertical (Cartesian and cylindrical) coordinate in the fixed coordinate system |
| $y$ | The vertical (Cartesian and cylindrical) coordinate in the moving coordinate system |
| $z_i$ | Electron number of species $i$ |

Selected subscripts

| | |
|---|---|
| $ALE$ | Refers to term in moving coordinate system |
| $FIX$ | Refers to term in fixed coordinate system |
| $T$ | Refers to a tangential term |
| $i$ | Refers to species (see below) |
| $Acc$ | Accelerator species |
| $Add$ | Generic additive chemical(s) |
| $Supp$ | Suppressor species |
| Cu | $Cu^{2+}$ ion |
| H | $H^{+}$ ion |
| $HSO_4$ | $HSO_4^{-}$ ion |
| $SO_4$ | $SO_4^{2-}$ ion |
| $CuSO_4$ | $CuSO_4 \cdot 5H_2O$ (copper sulfate pentahydrate) |
| $H_2SO_4$ | $H_2SO_4$ (sulfuric acid) |

# Chapter 1

# Introduction

## 1.1 Work background and motivation

A variety of formation and destruction processes occurring in the spatially microscopic domain have become and constantly are a subject of interest due to the explosive increase of microtechnologies. Examples are found in research of dendritic growth or tumor growth, bubble formation or other interfacial flows as well as in applications of etching and electrochemical deposition, the last of which is also the case-example in this paper. With the continuing increase in popularity of nanometer-scale technologies, where similar processes are applied in e.g. microchip manufacturing, the significance of these processes can only be expected to increase.

The movement or shape change in the described processes is generally driven by a known driving force, such as buoyancy in naturally convective flows, the thermodynamic potential in dendrite growth or the electric potential in electrochemical deposition. Simultaneously, however, the shape-change process is essentially governed by surfactants - i.e. substances acting on the surface that is evolving. This calls for means to accurately simulate the mass transfer and mass balance of the surfactant substances both on the surface strictly as well as in the near-space – e.g. in a film or diffusive layer – that covers the surface. Further, the simulation must also take into regard that the surface shape is constantly changing.

The described kind of processes are clearly both temporally and spatially varying and thus lead to a system of partial differential equations (PDEs) that need to be solved during the simulation. There are several approaches to solving such

equation systems numerically, but the finite element method (FEM) was chosen as the basic tool in this work. The selection is natural thanks to the suitability of the method itself and the availability of both good method documentation as well as readily made FEM-based solver applications.

Specialized computational methods for capturing or tracking moving boundaries and surfaces are numerous. Some methods applied to modelling surface movement with surfactants attached to the moving surface include at least the level-set methods [1, 2, 3], the boundary element method (BEM) [4], boundary markers [5], the volume-of-fluid method [6] and the arbitrary Lagrange-Eulerian (ALE) method [7, 8]. The ALE method, in combination with FEM is applied also in this study and one subject is to present a simple formulation for the local area element computations necessary to follow surfactant behavior on a shape-changing surface. Though a variety of methods for modelling surface movement already exists and many of them are also well documented, it is fair to say that computing mass transfer phenomena on shape-changing surfaces is a field of modelling still far from established.

Recent studies giving derivations for the mass balance of insoluble surfactants on a shape-changing surface are given in e.g. the studies [9, 10]. Based on these, calculating surface curvature is required when modelling mass balance of surfactants on a moving surface. However, other studies [6, 7] indicate that the curvature-containing mass balance formulations can be modified to obtain formulations where computing surface curvature is not necessary. Modifying the original equations for computational purposes is motivated by the desire to apply the ALE method in the computational implementation. Though the FE and ALE methods generally work well together, computing the moving boundary curvature in an FE-ALE model can be a problem because the moving surface is modelled as an actual finite element model boundary, and the curvature as well as the velocity divergence of this is often numerically very noisy. Such a problem does not occur when using e.g. the level-set method, because then the moving boundary is modelled as a contour line of a function, which is generally much smoother than a discretized boundary curve. On the other hand, very important features of the ALE method are simplicity of implementation and good accuracy, obtained with a relatively light computational load (compared e.g. to an equally accurate level-set model). Both are features, which make the ALE method interesting even when developing models for online process control purposes.

In addition to the noisiness of surface curvature and velocity divergence values, especially predominant on sharply deforming surfaces, it turns out that when applying the analytical equations for surfactant mass balance with the ALE method in practice, accurate results are hard to obtain with the curvature-based modelling system, when the surface movement velocity direction deviates significantly from the surface normal direction. Such problems are completely avoided when applying the surface area-based approach, also studied and applied in this research. The described problem does not exist either when applying e.g. boundary capturing methods, like the level-set method, that are constructed in such a way that the surface movement velocity is always normal to surface. The ALE method, however, is essentially a boundary marker-based method and the boundary elements' locations are thus explicitly tracked, which means there often also is a surface velocity component not normal to the surface. When applying the surface area-based approach for surfactant mass balance modelling, such problems related to direction of surface velocity can be overcome.

The problems of the area-based approach are related to how boundary movement velocity is computed (in this implementation) when utilizing the ALE method. With the transformations and smoothing equations required with the ALE method, the movement of a boundary may become partially non-physical, which leads to non-physical evolution of *local* surface mass balance.

The discussion and formulations in this thesis are restricted to two dimensional models, which describe either plane-symmetric or axially symmetric physical geometries, where the above described two approaches for surfactant mass balance computation are studied in depth. They are compared and their differences are discussed both in respect to physical interpretation and implementation-wise. Several other formulations and methods are introduced as reference material but the chosen two methods were found superior to others in terms of actual simulated results and therefore picked for further inspection. The studies mentioned here and later in the text, along with several others, are used as reference material in this study but the computational evaluations and extension of the methods to an axially symmetric system as well as the discussions related to coordinate systems, temporal and spatial considerations on model scaling are created by the author. The COMSOL Multiphysics FEM-computation software was utilized as solution platform when referring to computational implementations of the models and mechanisms developed. Though no application specific utilities were applied and all equations were configured to the solver software

manually, no actual computer code was written by the author.

Basic knowledge of the finite element method as well as knowledge over mass transfer phenomena such as diffusion and migration are assumed from the reader. Knowing fundamentals of fluid dynamics as well as surface chemistry helps following the text because the surfactant mass balance treatment essentially stems from and combines these fields of science. Elementary vector calculus is applied in several derivations and basic tensor notations are included in some equations without explanation. Otherwise the text aims to uncover the discussed subjects as thoroughly as necessary.

The thesis is self-contained on behalf of its computational work, which is the main yield of the thesis research. However, two journal articles and a conference publication are mentioned as directly related material, in order to further document the case process analyzed in the practical part of this thesis.

This introductory section is completed with a brief description on basic techniques related to modelling moving geometries with the ALE method as well as some aspects related to the general treatment of model geometries. In the beginning of Section 2, the necessary physical equations for modelling surfactant mass balance are introduced and formulated based on two distinctively different approaches. The equality of these are shown in the appendices. The formulations are compared numerically based on simulations in the end of Section 2. The third part of this thesis illustrates practical application of the developed computational means by applying the methods to model surfactant mass balance in the electrolytic microvia filling process and comparing the model results with empirical data. The section is concluded with a discussion on temporal and spatial dimensions of the shape-change process and how these need to be considered when constructing a model with surfactant mass balances included.

## 1.2 ALE basics

The arbitrary Lagrange-Eulerian (ALE) method is well documented in e.g. [11]. The basic principle is to create two overlapping coordinate systems, of which the other is spanned by dependent variables and the other by independent variables. In this study, the former of these coordinate systems is used to model the deforming model geometry and is thus called the current coordinate system of the model. The latter again is used to configure the original model

geometry and is thus called the original coordinate system. Already the naming convention indicates that the change from the original model geometry into the deformed geometry occurs over a time and that the model is inherently time-dependent. Of course, this need not be so but in our case this is exactly the case – exactly because the electrochemical deformation process is time-dependent.

For practical reasons, the models in this study are constructed in two dimensions by utilizing symmetry conditions. Therefore also the equations and models in this paper are formulated in two dimensions though their extension to three dimensions is straightforward. Adding the third spatial dimension, however, increases computational load of the models significantly, in such an extent in fact that solving the models on a desktop PC becomes impractical. The current, deforming coordinate system consists of equally as many dependent variables as there are independent variables in the original coordinate system. In the two-dimensional Cartesian coordinate system this number is two, the current coordinates being denoted $(x, y)$ and the corresponding original coordinates $(X, Y)$. In a plane-symmetric case, the $x$ and $X$ coordinates correspond to the horizontal direction and $y$ and $Y$ to the vertical direction. When considering the axially symmetric system (cylindrical coordinate system), reduced to two dimensions thanks to symmetry, $x$ and $X$ represent the current and original radial coordinate, respectively, and $y$ and $Y$ again the vertical coordinates respectively.

In order to account for the model geometry deformations, the physical and chemical equations are first formulated as functions of variables and differentials in the deforming geometry coordinate system. Then, these equations are solved in respect to the independent variables, i.e. in the original coordinate system, by using the ALE coordinate transformation. The ALE transformation for a scalar variable $f$ solved in the deforming 2D coordinate system is given by (1.1).

$$\begin{pmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{pmatrix} = \frac{1}{\det(\mathbf{J})} \begin{pmatrix} \frac{\partial y}{\partial Y} & -\frac{\partial y}{\partial X} \\ -\frac{\partial x}{\partial Y} & \frac{\partial x}{\partial X} \end{pmatrix} \begin{pmatrix} \frac{\partial f}{\partial X} \\ \frac{\partial f}{\partial Y} \end{pmatrix}, \text{ where} \tag{1.1}$$

$$\det(\mathbf{J}) = \frac{\partial x}{\partial X}\frac{\partial y}{\partial Y} - \frac{\partial x}{\partial Y}\frac{\partial y}{\partial X}.$$

If $f$ is a vector, the components of $f$ should be transformed individually.

The velocity $\mathbf{v} = (\partial x/\partial t, \partial y/\partial t)$ is that of the deforming model geometry and

in practice, is the movement rate of the computational mesh created in the time-dependently deforming geometry. $\mathbf{v}$ can be specified explicitly over the whole system or only on a part of the system, from where the movement of the rest of the system is then derived. The latter case is applied in this study and Laplacian smoothing (1.2) is utilized to solve the deforming model points' movement where it is not explicitly specified. This is illustrated in Figure 1.1 - a boundary condition is used to specify the mesh points' velocity ($\mathbf{v}_{BC}$) on the boundary indicated with blue color and (1.2) is then used to solve the mesh movement elsewhere.

$$\nabla^2 \mathbf{v} = 0 \tag{1.2}$$

It should be noted that (1.2) is completely non-physical and thus the movement solved upon this equation may have nothing in common with actual physical material movement.

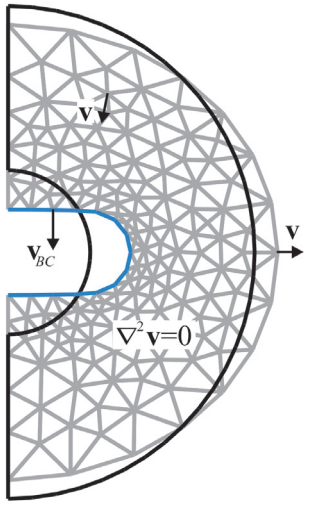


**Figure 1.1:** The movement of a deforming modelling domain due to a boundary condition velocity $\mathbf{v}_{BC}$, solved by using Laplacian smoothing.

The boundary condition $\mathbf{v}_{BC}$ can be specified arbitrarily or derived from physical conditions as is done in the practical part of this work, where the electrodeposition process specifies boundary movement. Naturally, the boundary conditions for the structure movement need not be only boundary conditions. For instance, the system movement can be specified over the whole model when modelling e.g. fluid flow.

After solving (1.2) with the given boundary conditions, the positions of the

current, deformed coordinate system is obtained by integrating in time. Finally, the transformation (1.1) is, as said, used to return the variables whose values are actually sought for in the deformed coordinate system back to the original coordinates for solution.

The transformation (1.1) must also be accounted for in integral equations because it alters the infinitesimal volume element, denoted $dV$. When formulating integral equations over the moving system, the infinitesimal volume element, $dv$ becomes $dv = \det(\mathbf{J})dV$.

When applying the ALE method, the initial geometry of the shape changing objects is configured as in any FEM model, as domains and boundaries. This means all moving boundaries or interfaces are tracked automatically as the model shape changes in time, which is indeed a benefit, manifesting as easy model configuration.

The biggest handicap of the ALE method, following its Lagrangian, boundary marker-based approach, is its incapability to tolerate dramatic deformations in the geometry and especially its total intolerance to topological changes in the model geometry. In practice, topological changes mean collision or breaking apart of domain boundary lines or spatial domains. Computationally this would mean addition or removal of computation grid nodes and when applying the ALE method, would always require complete restart of the model if encountered. The mentioned level-set method, as well as other so-called boundary capturing methods do not suffer of this handicap since they, as said, essentially model the shape changing surfaces as contour lines of some level function. This contour line is of course independent of the computation grid, whereby the boundary capturing methods also are fully capable of handling topological changes in the modelling geometry, on the expense of vastly increased computational load when accuracy is maintained.

Since the problem at hand essentially includes physical phenomena that take place on an actual surface, i.e. in a layer of practically infinitely small thickness, some special aspects of the model domain geometries should be noticed. Generally model domains represent physical spaces with non-zero ranges in all three spatial dimensions and boundaries are interfaces between such spaces, thus being e.g. planes with non-zero ranges in two spatial dimensions. (The "thickness" of a boundary is zero, so to say.) Since in this model the third spatial dimension is reduced away by applying symmetry conditions, the domains are planar areas with two spatial dimensions of non-zero size, and the

boundaries are generally lines with only a length dimension.

Because some of the model boundaries also represent the surfaces of a solid physical structure where surface-chemical phenomena take place, though the actual boundaries have no thickness, in the model they should be considered to have a range in two or three dimensions, depending on model symmetry. In a plane-symmetric model, the surface phenomena are considered only in two dimensions but in a cylindrically symmetric case also the angular dimension is included in the surface deformation, and in this way also in the surface chemical phenomena. Further it should be noted, that due to reducing one spatial dimension away from the original model, the boundaries of the surface domains are modelled as points. In a plane symmetric case, these boundary-points are of zero thickness in all dimensions but in the axially symmetric case, they represent a circle circulating the symmetry axis and thus have a range in one dimension. The coordinate system details will be explained more further in the text. Figure 1.2 illustrates the conversion from three to two dimensions in a plane-symmetric case. The lower boundary is intentionally pictured as a surface layer with a thickness, though in the model the boundary of course has zero thickness.

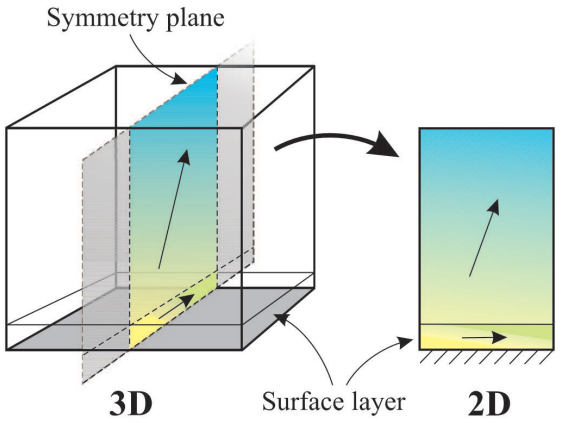


**Figure 1.2:** An illustration of reducing three dimensions to two dimensions for modelling purposes in a plane-symmetric case. The straight arrows indicate the direction of e.g a concentration gradient of species in the open space and the direction of the surface concentration gradient of a species attached on the surface.

In the domain space, mass transfer, electric field formation and other phenomena taking place there are configured according to the equations commonly applied to describe these phenomena. However, on the boundaries that represent surfaces of solid structures, mass balance and mass transfer of surfactants are configured as boundary conditions or as equation sub-systems, only valid on these boundaries and by applying *tangential differential equations* (Appendix

C). These equation systems will be described next and more discussion related to the practical application of the ALE method will be given in Sections 2.2 and 3.

# Chapter 2

# Surfactant mass balance

The amount of species attached on a certain area of a surface is measured by the species' surface concentration $\Gamma$ (mol/m$^2$) and is straightforwardly calculated as $m/A$, where $m$ is the amount of species attached (mol) and $A$ is the area in question (m$^2$). The dynamic surface concentration of a surfactant is affected by the following factors:

- How fast species are brought onto the surface by either adsorption from the surrounding environment or by e.g. a chemical reaction occurring between reactants already on the surface
- How fast the species is removed from the surface by either desorption to the surrounding environment or by a surfactant consuming chemical reaction or by incorporation into an evolving surface, which is a considerable factor in many growth processes
- How fast species on the surface move along the surface by diffusion or convection
- How the surface area locally changes due to surface deformation

In this study only adsorption, desorption, consumption on surface, surface diffusion and effects of local area change are considered to affect surfactant concentration. Adding further factors is straightforward.

Figure 2.1 illustrates what factors affect the surface concentration of a soluble and diffuse surfactant, attached on a moving surface. Surface concentration is denoted by $\Gamma$ and nothing can be said about $\Gamma(t)$ compared to $\Gamma(t + \Delta t)$ without

knowing the factors $N^{ads}$ (adsorption flux), $N^{des}$ (desorption flux), $N^{cons}$ (flux of surfactant consumption on surface), $-D^s\nabla_T\Gamma$ (flux of surfactant diffusion along surface). In the illustrated case, however, it is clear that $A(t+\Delta t) > A(t)$ since the surface is convex in respect to the direction of surface movement velocity. (No convective mass transfer of surfactant along the surface is illustrated.)

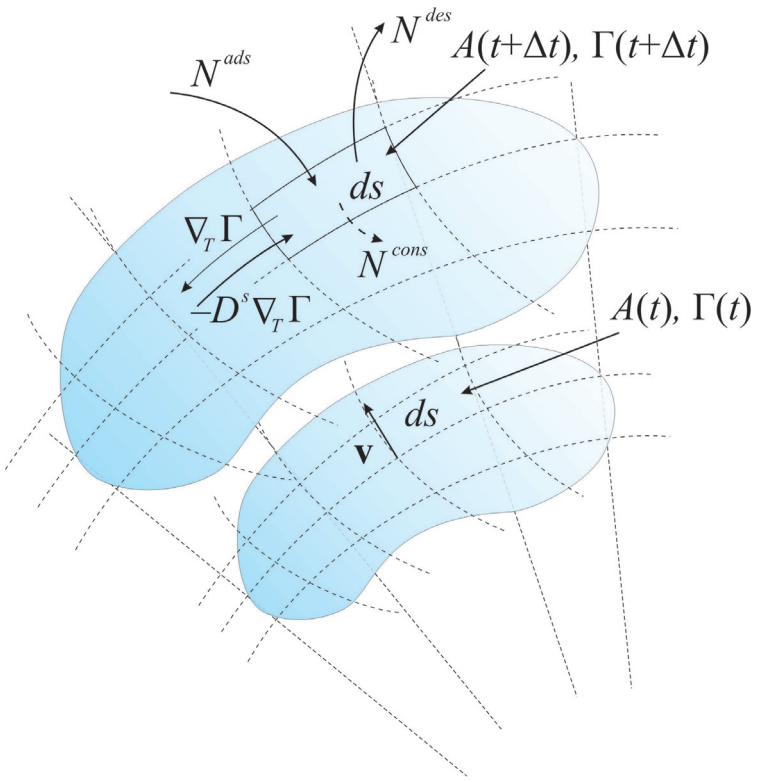


**Figure 2.1:** An illustration of factors affecting surfactant surface concentration.

## 2.1 Surfactant mass balance equations

Formulations for mass balance equations of surfactants on an evolving surface have been formulated in e.g. [9, 6]. The equations have been applied into practice in e.g. [6] where the volume-of-fluid method in combination with the level-set method are used to capture the moving interface. In a recent study [7], the aforementioned methods are completed with the ALE method to model the mass balance of an insoluble surfactant on a two-solution interface.

The equations for the surface concentration of an insoluble surfactant species are based on conservation of mass of species attached on a specific area $dA$ of

a surface and in the general, integral form can be given as (2.1).

$$\frac{d}{dt}\int_{A(t)} \Gamma dA = \int_{A(t)} \left( \frac{d\Gamma}{dt} dA + \Gamma \frac{d}{dt} dA \right) = 0 \tag{2.1}$$

For implementation purposes (2.1) has to be formulated in a manner that it can be solved numerically and upon physical constraints that are deduced based on the problem nature. How this is done and how (2.1) is eventually applied requires understanding what factors of those listed in the beginning of this chapter are effective in the system and being able to model them.

To classify and to correctly configure the problem one should consider both the surfactant in question as well as the type surface included. Physically appropriate terms, introduced below, should be included in the equations based on whether the surfactant being modelled is soluble or insoluble in its environment and if it can diffuse along the surface it is attached to. Then on the other hand, the surface movement and deformation depends significantly on whether it is e.g. phase boundary between two mutually insoluble liquids or the face of a solid immersed in a liquid or vapor.

### 2.1.1 Components of surfactant mass balance

To configure (2.1) according to the physical conditions prevailing in a certain system, some basic components of equations for surface concentration are given next. These can be compiled and substituted into the integral form equation (2.1) to obtain a realization for surface concentration, as is eventually done in the text.

**Insoluble species strictly attached on the surface** Consider that an amount of surfactant species are attached on the surface strictly, unable to further accumulate onto the surface from the surrounding environment or escape the surface, and unable to anyhow move along the surface. In this case, once the initial conditions are set the local surfactant species' concentration is essentially determined by local surface area change. Such a situation is simply modelled by (2.2).

$$\frac{\partial \Gamma}{\partial t} = -\Gamma \left( \frac{1}{A} \frac{dA}{dt} \right) \tag{2.2}$$

An illustrating and familiar, though rather unscientific, example of a case described by (2.2) would be the intensity of a colored paint-mark on an inflating/deflating balloon. Equation (2.2) of course means that $\Gamma = \Gamma_0 A_0/A$ and that there is no upper bound to surface concentration and the lower bound is zero, assuming infinitely large local surface expansion. Subscript 0 indicates initial value.

**Surface diffusion** If the surfactant diffuses along the surface, a diffusive flux term $D^s \nabla_T \Gamma$ is added to the equation.

$$\frac{\partial \Gamma}{\partial t} = -\Gamma \left( \frac{1}{A} \frac{dA}{dt} \right) + \nabla_T \bullet (D^s \nabla_T \Gamma) \tag{2.3}$$

In (2.3) $D^s$ is the surface diffusivity of surfactant (spatially constant) and the surface differential operator (or tangential differential operator) $\nabla_T$ is introduced. The $\nabla_T$ operator functions as the regular differential operator $\nabla$, but only on a surface, always yielding a result which has values appropriate to the surface. In practice $\nabla_T$ is implemented as $\nabla_T = (\mathbf{I} - \mathbf{nn}^{\mathrm{T}}) : \nabla$, where $\mathbf{I}$ is the identity tensor, $\mathbf{n}$ the surface outward normal vector of the deformed geometry surface and the colon denotes tensor product. Hence, the diffusive flux term $D^s \nabla_T \Gamma$ is identical to a regular diffusive flux term (e.g. $-D\nabla c$) but only along the surface in question. The sign of $D^s \nabla_T \Gamma$ is not intuitive, but this is because the gradient is along a surface and the direction of the gradient is also affected by the surface normal vector direction, depending on situation.

From here on the Laplace operator is applied in mass balance equations, e.g. $\nabla_T \bullet (D^s \nabla_T \Gamma) = D^s \nabla_T^2 \Gamma$.

**A soluble or consumable species** Usually the surfactant arrives to the surface from the environment surrounding the surface and attaches onto the surface by physical or chemical adsorption. Analogously, the surfactant can escape the surface by desorbing away from it back into the surrounding environment. Thus the adsorption and desorption fluxes of the surfactant affect its local surface concentration and the corresponding flux terms should be added to the mass balance equation as in (2.4).

$$\frac{\partial \Gamma}{\partial t} = -\Gamma \left( \frac{1}{A} \frac{dA}{dt} \right) + D^s \nabla_T^2 \Gamma + N^{ads} - N^{des} \tag{2.4}$$

In (2.4) $N^{asd}$ and $N^{des}$ are the adsorption and desorption fluxes (considered positive). The formulation of these is chemistry and generally not straightforward. Determining and even measuring maximum surface concentration values in steady state, that depend on e.g. the surfactant concentration in the surrounding environment as well as the surface type (i.e. the maximum concentration of possible adsorption sites) is tricky, to say the least. However, since the system is a chemical reaction system that tends towards an equilibrium formulations for steady state surface concentration - known as adsorption isotherms - have been established. These formulations rely on various assumptions on the adsorption and desorption reaction orders as well as on the surfactants' interactions once attached on the surface [12].

When modelling dynamic adsorption and desorption, no steady state assumptions can be made and all the interactions as well as factors affecting the rates of these reactions must be included in the $N^{asd}$ and $N^{des}$ flux terms. Such an example case will be given in the practical part of this text.

From the surface mass transport point of view, the phenomena of a surfactant being chemically or otherwise consumed on the surface it is attached to or being incorporated into the surface, correspond exactly to that of the desorption phenomenon. On behalf of the surface, such phenomena simply remove species from the surface, exactly like desorption does and can be modelled by simply adding a consumption or incorporation flux term to the surface mass balance equation. The difference with desorption is in how the removed species are treated, namely, whether they are returned to the surrounding environment (as happens after desorption) or completely removed from the system or added to the evolving surface. Also processes of this type will be illustrated in the practical part of the text.

**Surface convection of a mobile surfactant** If the surface is of such nature that the surfactants are mobile on the surface, there can exist a difference in movement velocities of the surface and the surfactant on-top of the surface. Such a situation can be imagined e.g. on the surface of an oil drop, where tensides are spread evenly due to van der Waals forces pushing tenside molecules apart or pulling them together. The surfactant distribution looks for the minimum of surface tension by flowing on the surface as a fluid.

A simpler situation is that where an actual fluid front is flowing on-top of a solid surface due to either gravity or the capillary force. Now the surfactant is

a film-like layer of fluid, beneath which the examined surface lies or moves and the surface concentration would essentially describe the thickness of this layer.

In both cases the movement of the surfactant relative to the examined surface is exactly tangential and thus a surface convection term should be added to the mass balance system. Formulation of such a term is straightforward if the surface velocity $\mathbf{v}_s$ of the surfactant is known (2.5).

$$\frac{\partial \Gamma}{\partial t} = -\mathbf{v}_s \bullet \nabla_T \Gamma \tag{2.5}$$

Note that in 2.5 $\mathbf{v}_s$ is the relative tangential movement velocity of the surfactant in respect to the surface.

If, in a special case, the surfactant layer is assumed to remain on the surface and fixed in respect to a reference point of deformation, the relative tangential movement of the surfactant layer is then exactly the opposite of the surface tangential velocity $\mathbf{v}_T$ of a geometry deforming "beneath" the surfactant layer. $\mathbf{v}_T$ is computed simply as the surface velocity projection on the surface itself, $\mathbf{v}_T = (\mathbf{I} - \mathbf{nn}^\mathsf{T}) : \mathbf{v}$.

The (2.5) term can be simply added to (2.2)–(2.4) but is left out here because from here on, we assume that no surface convective mass transport of surfactant species occurs.

### 2.1.2 Area terms, curvature and surface concentration

To accompany the surface concentration equation components given Section 2.1 and to clarify the coupling between the *area term* – and – *surface curvature* and *surface velocity divergence*, these two terms as well as various related aspects are discussed next.

**Surface area terms** The area term $A$ and its time derivative, $dA/dt$ occur in the mass balance equations given above. However, often the area term is omitted but another term containing the surface curvature is included in the equation system.

In [9, 6, 13] the formulae (2.6a) and (2.6b) have been given for incompressible fluid flows and are used to model local surface area on a deforming and moving

surface. [1]

$$\frac{dA}{dt} = -A\left(\mathbf{n} \bullet \nabla\mathbf{v} \bullet \mathbf{n}\right) \tag{2.6a}$$

$$\frac{dA}{dt} = A\left(\mathbf{I} - \mathbf{n}\mathbf{n}^{\mathsf{T}}\right) : \nabla\mathbf{v} \tag{2.6b}$$

The equations (2.6a) and (2.6b) are essentially the same, only having a different form and the latter being implementation-wise more intuitive. Already these two equations point out the coupling between surface movement velocity and local area. In practice the equations are applied in models solved by finite difference or finite element methods and $A$ denotes the area of one computational element. $\mathbf{n}$ is the (outward) unit normal vector of the surface in question.

In an ALE-based model the ALE transformation (1.1) can be applied to obtain the local area element directly as (2.7).

$$A = A_0 \parallel \mathbf{n} \parallel \tag{2.7}$$

In (1.1) $A_0$ is the initial surface area element size. The initial model geometry typically equals the the model geometry in the original coordinate system geometry (which remains unchanged during model solution) and thus all initial terms are known and constant. (See Appendices A and B for expansion of (2.7)). Whereas (2.6a), (2.6b) are ordinary differential equations (ODEs) in respect to time, (2.7) is and ODE in respect to local deformation and static.

**Surface curvature and surface velocity divergence** The surface curvature and the surface velocity divergence are main topics of this study. Throughout the discussion here, surface curvature refers to *the surface outward normal vector divergence*, $\nabla \bullet \mathbf{n}$ and surface velocity divergence is $\nabla_T \bullet \mathbf{v}_T$ correspondingly. Both, but especially surface curvature, are often applied in surfactant mass balance computations and this certainly is handy in cases, when they are easily found, especially in plane-symmetric geometries.

In an axially symmetric system, be it either due to numerical instability or the difficulty to formulate local surface curvature or the surface velocity divergence

[1] Equation notations are as given in the referred sources, but the transpose notation is used because some terms in the equations are matrices.

of an axially symmetric arbitrary surface, applying these with the ALE method has proven difficult. It can however be shown that the surface curvature, surface velocity divergence and local surface area are coupled as given in (2.8).

$$\frac{1}{A}\frac{dA}{dt} = (\nabla \bullet \mathbf{n})\,(\mathbf{n} \bullet \mathbf{v}) + \nabla_T \bullet \mathbf{v}_T \tag{2.8}$$

Equation (2.8) says that proportional rate of local area change is due to a combination of movement in the direction of the normal of a curved surface as well as a difference in movement velocity in the direction tangent to the surface. However, (2.6b) clearly says that the proportional rate of local area change can also be formulated without using the curvature $(\nabla \bullet \mathbf{n})$ or divergence $(\nabla_T \bullet \mathbf{v}_T)$ terms, simply by dividing the equation side-wise with $A$, as given in (2.9).

$$\frac{1}{A}\frac{dA}{dt} = \left(\mathbf{I} - \mathbf{n}\mathbf{n}^{\mathsf{T}}\right) : \nabla\mathbf{v} \tag{2.9}$$

The equality of (2.8) and (2.9) in the plane-symmetric system is shown in the final part of Appendix D by performing an element calculation in two dimensions. In the axially symmetric case, the formulation for $A$ must be completed as shown in Appendix A.

Recognizing the equality of (2.8) and (2.9) enables actually interpreting what the the movement of a curved surface does to the surface physically, and how local area change of a surface can be divided into compression or expansion due to a curved surface moving in the direction of its normal or due to a (planar) surface having differential surface tangential velocity. I.e. the surface area change phenomena in the two different cases – curved surface and planar surface – can now be handled and examined separately.

Though the observation that the right-hand sides of (2.8) and (2.9) are equal has also been made in [6], in the case of interfacial flows of incompressible fluids, the equality is left unspecified. Further, the area-based approach for surfactant mass balance modelling has also been taken in [6, 7], as said, in the case of fluid dynamics and interfacial flows but the computational mechanisms include utilization of several different means for boundary capturing as well as a complex ad-hoc grid adaptation technique. In this work, simplicity and a computationally modest mechanism is sought for.

**Surface concentration on a moving surface** The integral form equation (2.1) is the starting point for formulating an equation for surfactant mass balance, i.e. a PDE including the surface concentration $\Gamma$ as dependent variable and time and location as independent variables. In [9] the surface concentration equation is derived into the form (2.10), which explicitly yields the dynamics of surface concentration $\Gamma$ in terms of surface deformation. This is the common form generally applied to computing mass balances on surfaces.

$$\frac{\partial \Gamma}{\partial t} = -\nabla_T \bullet (\Gamma \mathbf{v}_T) - \Gamma \, (\nabla_T \bullet \mathbf{n}) \, (\mathbf{v} \bullet \mathbf{n}) \tag{2.10}$$

Though equation (2.10) is based on the integral form and is thus general, it clearly contains no area elements. However, as said in the previous paragraph, the effects of local area change are incorporated into the two product terms that include the surface tangential velocity, $\mathbf{v}_T$, and the surface normal velocity, $\mathbf{v} \bullet \mathbf{n}$. These terms also include the surface curvature, $\nabla \bullet \mathbf{n}$, as well as the tangential divergence of the surface tangential velocity, $\nabla_T \bullet \mathbf{v}_T$, which both can be numerically very noisy on a deforming ALE model boundary.

The problems caused by numerical noise can be come about in two ways; (i) by solving the boundary curvature through a weak formulation of the boundary normal components, which is a computational arrangement illustrated in Appendix B or (ii) by using a surface area-based formulation of (2.10), given in (2.11).

$$\frac{\partial \Gamma}{\partial t} = -\mathbf{v}_T \bullet \nabla_T \Gamma - \Gamma \left( \frac{1}{A} \frac{dA}{dt} \right) \tag{2.11}$$

Though (2.10–2.11) model exactly the same phenomenon, they are based on different approaches, which manifests when the equations are implemented computationally. Equation (2.11) is very different from (2.10), since it aims to track surface concentration by following local area changes and not surface curvature, partially actually neglecting how the surface moves or whether it moves at all.

In Section 2.2, a simulation-based evaluation is carried out to examine the presented equations when they are put into practice. The equations (2.8) and (2.9) are combined with (2.7) to formulate the surfactant mass balance equations without computing surface curvature nor surface velocity divergence and further, without implementing any additional computational elements to the model by utilizing readily existing ALE transformation components. The pro-

cedures given in Appendix B are used to compute surface curvature and thus to implement a curvature-based approach to surfactant mass balance computation. The means and differences of implementation and applicability of both of the two approaches described above, namely that of applying curvature and surface velocity divergence, and that of applying local surface area for surfactant mass balance modelling, are illustrated by using the simple computational mechanisms developed in this work. The exact equations related to the approaches referred to as *curvature-based* and the *area-based* approach are specified in the text. (Note that references to the tangential divergence of the surface tangential velocity is omitted from the name of the former approach.) Necessary surface diffusion, adsorption, desorption etc. terms may of course be added, but these are the same regardless of the approach.

**Remarks** The most recent formulation of (2.1) found for computational purposes was given in [10] as (2.12).

$$\frac{\partial \Gamma}{\partial t} = -\nabla_T \bullet (\Gamma \mathbf{v}_T) - \Gamma (\nabla_T \bullet \mathbf{n}) (\mathbf{v} \bullet \mathbf{n}) + D^s \nabla_T^2 \Gamma + \mathbf{v} \bullet \nabla_T \Gamma \qquad (2.12)$$

Equation (2.12) is also meant for modelling surface concentration of an insoluble species on a moving surface, in a case where the surface can also move in the surface tangential direction - i.e. not only in the direction of its normal. This is generally the situation when using boundary marker methods, such as the ALE method, to follow boundary movement. When using boundary capturing methods, such as the level-set method, one can construct the system so that the boundary movement velocity is always normal to the boundary that is moving.

The formulation (2.12) includes an extra term (at extreme right) that is there to account for the difference between the direction of the surface normal and the surface movement velocity, and thus the formulation is meant to better model surfactant mass balance when surface deformations are modelled with boundary marker methods (i.e. mass is to be better conserved with this formula). It turns out after a short rearrangement that – if the surface diffusion term is neglected for simplicity – (2.12) is in fact exactly (2.2), which is the same formula as is applied in Section 2.2, in the *area-based* approach for surfactant modelling.

The difference with (2.10) and (2.12) is that the term $-\mathbf{v}_T \bullet \nabla_T \Gamma$ is not included in (2.12). In the cited study [10], it is explained that (2.10) accounts only for situations where the surface moves normally to itself. However, it is not noticed

that in this case, also $-\mathbf{v}_T$ and thus $-\mathbf{v}_T \bullet \nabla_T \Gamma$ is zero, essentially equating the two cases compared.

### 2.1.3 The weak formulation

In order to solve the surfactant mass balance equations in coupling with other related equations, it is often useful to formulate the surface mass balance equations into the weak form. In practical applications, the weak form simplifies, or even enables, making couplings between dependent variables and especially their differentials, as well as enables using the Lagrangian multipliers applied in solving the PDE system. Further, the weak formulation enables reducing second order spatial differentials from the equation system by applying the divergence theorem, as is illustrated in Appendix D.

For instance, in the practical cases documented in this text, the equation system to be solved together with the mass balance equations includes (i) mass transport of surfactant and other species in the environment surrounding the examined surface, (ii) an electric field system and most essentially (iii) the equations describing the moving coordinate system required to model surface deformations.

The weak form of (2.4), which describes a multi-functional surfactant mass balance system is obtained as (2.13).

$$\begin{aligned} &\int_{\partial\Omega} \hat{\Gamma} \frac{\partial \Gamma}{\partial t} ds \\ &= \int_{\partial\Omega} \left[ \hat{\Gamma} \left( N^{ads} - N^{des} \right) - \hat{\Gamma}\Gamma \left( \frac{1}{A} \frac{dA}{dt} \right) - \left( D^s \nabla_T \Gamma \right) \bullet \nabla_T \hat{\Gamma} - \hat{\Gamma} \left( \mathbf{v}_T \bullet \nabla_T \Gamma \right) \right] ds \end{aligned} \tag{2.13}$$

In (2.13) $\hat{\Gamma}$ is the surface concentration test function. Note that the notion $\nabla_T \hat{\Gamma}$ is somewhat misleading because it seems to refer to the gradient of the test function $\hat{\Gamma}$ - not to the test function of the gradient of $\Gamma$, which actually is used, i.e. $\nabla_T \hat{\Gamma} = \widehat{\nabla_T \Gamma}$ and the former notion is used only for aesthetics.

In (2.13) $\Omega$ denotes the modelling domain and $\partial\Omega$ thus its boundary, which $ds$ is an infinitesimal segment of. In the physical case, $\Omega$ is a space and $\partial\Omega$ a surface.

The weak form is also known as the variational formulation of PDEs. Here, the weak formulation in (3.15) is obtained through the common formulation procedure of weak form equations. The procedure to formulate (2.13) from (2.4) is given in Appendix D, but basically includes the three four steps: (i) multiplying the equation side-wise by the test function, (ii) removing the divergence terms by reformulation, (iii) integrating the equation by parts and (iv) applying the divergence theorem (Gaussian theorem) on closed boundaries to again remove divergence terms. The divergence terms in step (ii) include a gradient vector, i.e. the divergence would include second order spatial derivatives, which can be computationally difficult.

## 2.2 Simulation and evaluation

Some examples of deforming geometries are given here to illustrate the applicability of the equations. The simulations presented here were carried out using the COMSOL Multiphysics software, version 3.2b (3.2.0.304) with the weak form PDE modes for subdomains and boundaries as application mode for model configuration. On average, 8-12 variables were solved for in each model and second order and first order Lagrangian shape functions were utilized.

For analytical purposes, all surfactants are considered insoluble and fully immobile. Except for selected test-cases, surface diffusivity of the species is taken as zero, meaning $N^{ads} = N^{des} = D^s = 0$. In most of the cases, the initial surface concentration distribution over the analyzed surface is even, i.e. $\Gamma$ is the same everywhere on the modelled surface. Test-cases are provided, however, to illustrate the adequacy of the method, regardless of what the initial surface concentration distribution. The mentioned simplifications help following and comparing mass conservation between the various implementations in various cases. Mass conservation is here considered as a measure of adequacy.

Computations are carried out in both plane symmetric and axially symmetric geometries and the deformations describe (i) symmetric expansion-contraction (Figure 2.2) of a cylindrical/spherical object, (ii) even linear stretching during propagation of a planar/conical surface (Figure 2.7) and (iii) asymmetric "squeeze" (Figure 2.11) of a cylindrical/spherical object. The first case emphasizes movement in the surface normal direction, with surface tangential velocities being essentially zero. In the second case, surface tangential velocities have a significant role but also translation and normal movement of the surface exist.

The third case finally represents a hybrid system with various forms of surface deformation present.

When referring to *area-based* or *curvature-based* implementation, the models are otherwise identical except for the equation that is implemented for surfactant mass balance calculation. Specifically, in the area-based approach, the implemented equation is (2.14) and in the curvature-based implementation (2.15). In the area-based approach, the proportional rate of change of the area element is computed upon (2.9), modified accordingly when applied in an axially symmetric system. Due to immobility of surfactants, the surface convection term $\mathbf{v}_T \bullet \nabla\Gamma$ is zero and thus left out from the equations.

$$\int_{\partial\Omega} \hat{\Gamma}\frac{\partial\Gamma}{\partial t}ds = -\int_{\partial\Omega}\left[\hat{\Gamma}\Gamma\left(\frac{1}{A}\frac{dA}{dt}\right)\right]ds \tag{2.14}$$

$$\int_{\partial\Omega} \hat{\Gamma}\frac{\partial\Gamma}{\partial t}ds = -\int_{\partial\Omega}\left[\hat{\Gamma}\Gamma\left(\nabla_T \bullet \mathbf{n}\right)\left(\mathbf{v} \bullet \mathbf{n}\right) + \hat{\Gamma}\Gamma\left(\nabla_T \bullet \mathbf{v}_T\right)\right]ds \tag{2.15}$$

When referring to the area-based method, the area element proportional rate of change is computed upon (2.9). Similarly, when applying the curvature-based method, the surface curvature is computed by applying the arrangement given in Appendix B.

In all the examined cases, the computational load of the model is kept modest. In cases (i) and (iii), the semi-circle shown in figures 2.2 and 2.11, which represents the geometry initial shape was discretized into 14 equally spaced intervals and in case (ii) the line (figure 2.7) is divided into 12 intervals. In total the models contain from 166 to 286 elements and approximately 1600 to 2600 degrees of freedom to be solved for, in cases (ii) and (i),(iii) respectively. The discretization is selected such that the model accuracy is no longer significantly improved despite adding grid points. The area-based approach reduces the computational load somewhat compared to the curvature-based method because fewer (additional) degrees of freedom need to be solved for, thanks to utilizing the readily existing ALE components.

The ALE system as well as the surfactant concentration variables are solved using second order Lagrangian shape functions but the area computation as well as surface curvature and surface tangential velocity differentials are solved by using first order Lagrangian elements. This selection is to reduce noise in the

lastly mentioned components as well as to reduce computational requirements. Preserving accuracy despite reduction of shape function order was confirmed empirically.

### 2.2.1 Surface velocity normal to surface

Figure 2.2 illustrates how the examined geometry changes during the first half of simulation. During the second half of simulation the geometry retracts back to its initial shape.

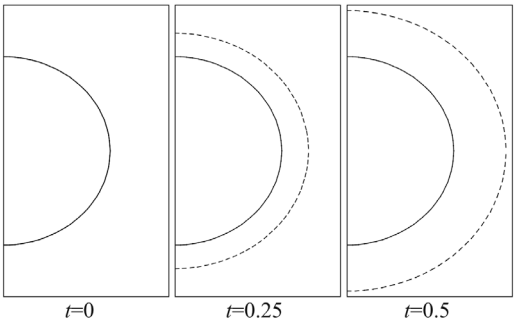


**Figure 2.2:** Surface boundary (dashed line) during first half of deformation. The sphere retracts back to initial shape (solid line) during the second half of deformation.

The velocity enforced on the deforming boundary in the symmetric expansion-contraction of a cylinder/sphere is given as a boundary condition in (2.16) and plotted in Figure 2.3. The smooth transition in boundary velocity direction is obtained by applying a twice differentiable smoothed Heaviside step function.

$$\mathbf{n} \bullet \mathbf{v} = \begin{cases} -1, & 0 \leq t < 0.48 \\ \text{smoothed}, & 0.48 \leq t \leq 0.52 \\ 1, & 0.52 < t \leq 1 \end{cases} \tag{2.16}$$

Figure 2.4 shows the computed surface area compared to the analytically calculated during deformation. The area computation is accurate during the whole simulation in both the axially symmetric as well as plane-symmetric case, however the inserts show that computation is significantly more accurate in the latter (up to $4^{\text{th}}$ digit) than the former (up to $3^{\text{rd}}$ digit) case.

Figure 2.5 illustrates how both the local area-based approach as well as the surface curvature-based approach yield consistent, and rather accurate results

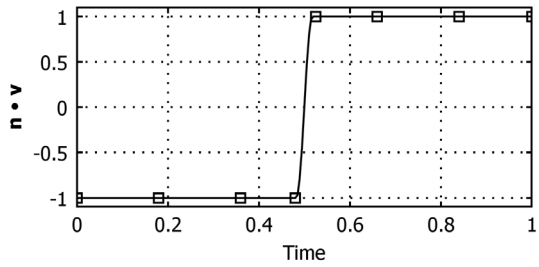


**Figure 2.3:** Surface (outward) normal velocity during symmetric expansion-contraction of circular objects.

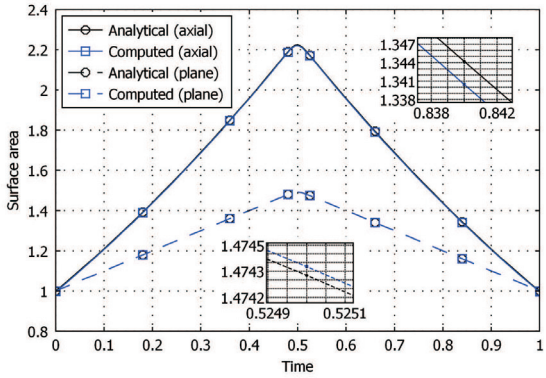


**Figure 2.4:** Surface area evolution during symmetric expansion-contraction of circular objects.

when the surface shape change is simple. Surfactant mass is conserved within a variation of $\pm 0.4\%$ in both cases. It is important to notice that in this case of symmetric radial expansion/contraction of a circular/spherical object, the surface always moves normally to itself and thus the net surface tangential velocity per element is always zero.

To confirm curvature computation accuracy, the computed curvature (at an arbitrary point on the surface) is compared with the analytical curvature of a sphere ($\nabla \bullet \mathbf{n} = 2/r$) and that of a cylinder ($\nabla \bullet \mathbf{n} = 1/r$), where the $r$ are radii (m) of the sphere and cylinder, respectively. The comparison is shown in figure 2.6 and a only small difference between the values can be observed.

### 2.2.2 Surface velocity tangential to surface

A straight front is moved according to the boundary condition in (2.17). This simulates propagation and linear stretching/compression of a planar or conical surface and surface deformation during the first half of simulation is illustrated

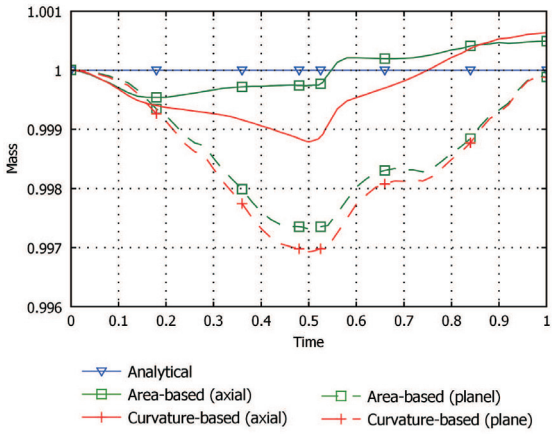


**Figure 2.5:** Evolution of surfactant total mass during symmetric expansion-contraction of circular objects.

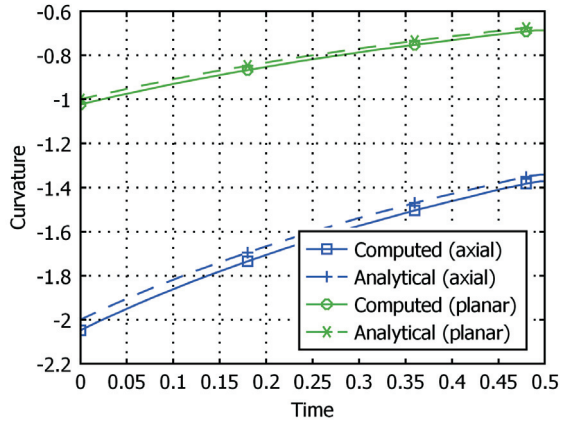


**Figure 2.6:** Analytical and computed surface curvature compared during first half of symmetric deformation of circular objects.

in Figure 2.7.

$$v_y = (1/100 + Y) \cdot \begin{cases} -1, & 0 \leq t < 0.48 \\ \text{smoothed}, & 0.48 \leq t \leq 0.52 \\ 1, & 0.52 < t \leq 1 \end{cases} \tag{2.17}$$

In (2.17) $v_y$, is the vertical velocity of the front and $Y$ is the vertical coordinate of the front in the fixed coordinate system, initially running from 0 to 1.

The curvature of a planar surface is by definition zero but for a conical surface

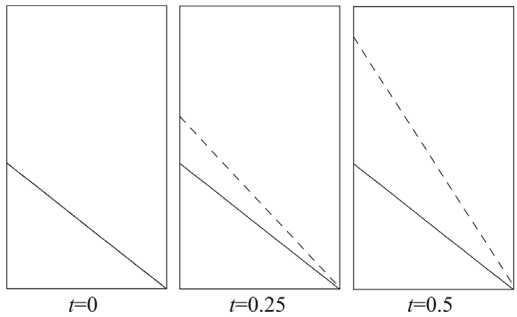


**Figure 2.7:** Surface boundary (dashed line) during first half of propagation and stretching of a moving front. The front retracts back to the initial shape (solid line) during the second half of deformation.

- specifically, a right cone - the curvature can be presented with the parametric representation in (2.18) and is thus compared here to the computed curvature of the deformed conical surface in Figure 2.8.

$$\nabla \bullet \mathbf{n} = \frac{h^2}{2r(h-z)\sqrt{h^2+r^2}} \tag{2.18}$$

In (2.18) $h$ is the total height of the cone, $r$ the radius of the cone base circle and $z$ is the parameter between $[0, h]$ (namely the vertical position on the cone surface). Figure 2.8 shows good correspondence between the computed and analytical curvature during deformation of the conical surface being simulated.

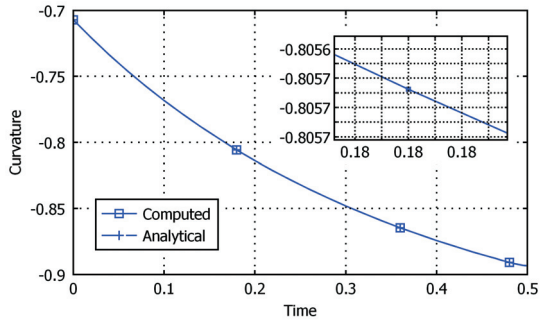


**Figure 2.8:** Curvature during first half of deformation of a conical surface, measured at a point on the arc circulating the cone at half way up its height.

It is noteworthy, that whereas the sphere and circle are symmetric, the conical surface is not, and thus its surface curvature also depends on where on the surface one examines it. Further, note that when reducing the angular coordinate from an axially symmetric geometry as is the case here, (2.18) reduces to $\nabla \bullet \mathbf{n} = h/(2x\sqrt{h^2+r^2})$.

It should also be noted that the cone tip is essentially non-smooth due to its geometry, even when considering axial symmetry. A capability to handle such conditions is vital considering practical application of a computational mechanism.

The surface area in this tangential deformation case is also analytically computable and therefore also the computed surface area values are compared with the analytical value for the simulated surface area.

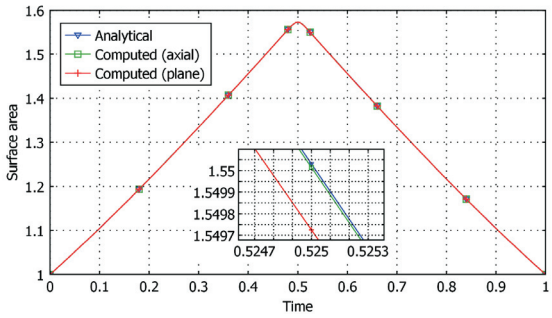


**Figure 2.9:** Surface area evolution of propagating front being stretched.

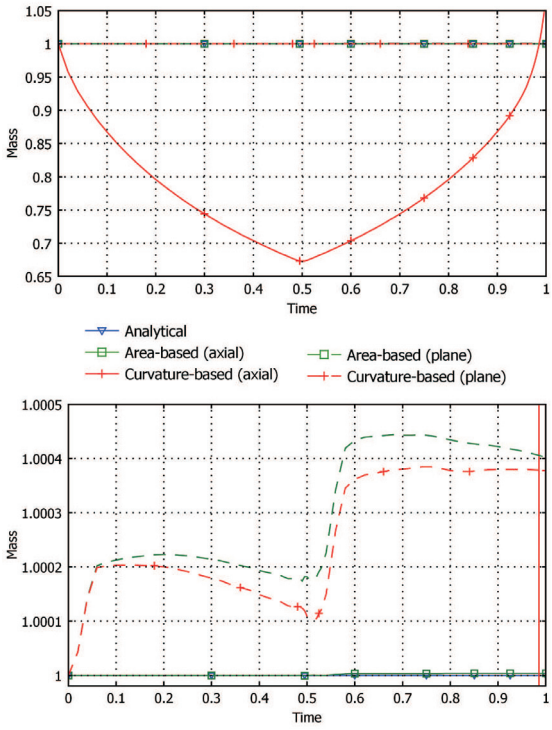


**Figure 2.10:** Evolution of total mass of surfactant adsorbed on the surface of a propagating and stretching front. Figure (b) is zoomed in on figure (a).

Evolution of total surfactant mass during deformation is given in figure 2.10.

It is evident that when a significant tangential velocity prevails during surface shape change in an axially symmetric system, the area-based implementation yields a significantly more accurate solution compared to the curvature-based method. Combining this with the recognition that surface curvature was computed accurately, one can deduce that problems on the surfactant mass balance computation are caused by the surface tangential velocity formulation in the axial symmetric case. (In the previous examined case, no surface tangential velocity existed due to symmetric radial deformation.)

### 2.2.3 Arbitrary surface deformation

Deformation of the cylindrical or spherical object during the first half of the simulation is illustrated in Figure 2.11. The boundary is moved according to the boundary condition in (2.19).

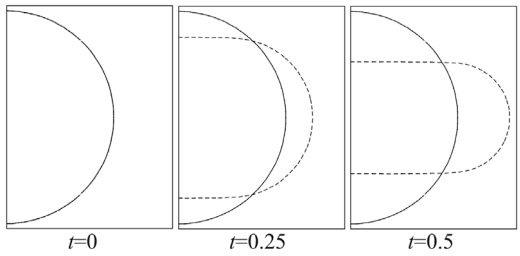


**Figure 2.11:** Surface boundary (dashed line) during first half of asymmetric deformation (squeeze). The object retracts back to the initial, spherical/cylindrical shape (solid line) during the second half of deformation.

$$\mathbf{n} \bullet \mathbf{v} = \left(2\sin\left(\frac{\pi}{2} + \arctan\left(\frac{n_y}{n_x}\right)\right) - 1\right) \cdot \begin{cases} -1, & 0 \leq t < 0.48 \\ \text{smoothed}, & 0.48 \leq t \leq 0.52 \\ 1, & 0.52 < t \leq 1 \end{cases} \tag{2.19}$$

The boundary velocity condition (2.19) is obviously non-physical and the deformation does not follow e.g. the shape of droplet oscillation as it would occur according to fluid dynamics. This is clear already upon the deformation shape illustrated in Figure 2.11. The boundary velocity condition was chosen so in order to fully control the deformation as well as the evaluation procedure also in the the case of asymmetric deformation.

The surface area evolution of the deforming object in different situations is plotted in Figure 2.12 and the corresponding plots for total surfactant mass are shown in Figure 2.13.

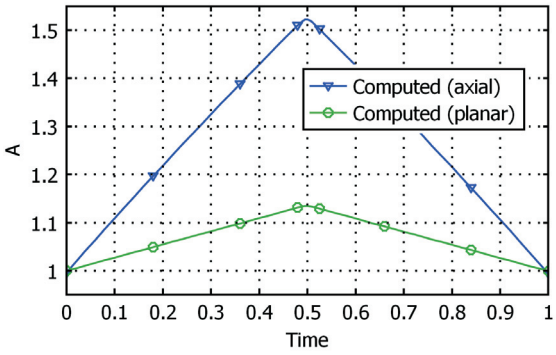


**Figure 2.12:** Surface area evolution of spherical (axial symmetry) and cylindrical (planar symmetry) objects being asymmetrically deformed.

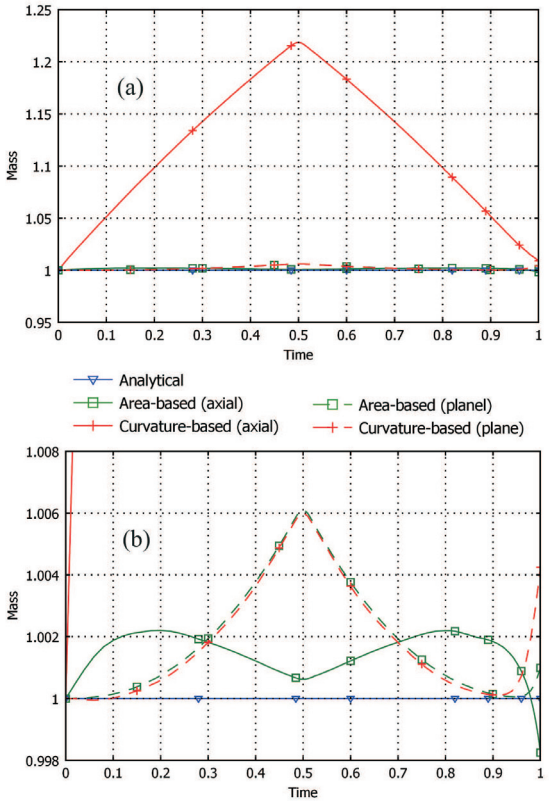


**Figure 2.13:** Evolution of total mass of surfactant on the surface of a spherical/cylindrical object being squeezed. Object type is indicated by the type of symmetry (axial/plane). Figure (b) is zoomed in from figure (a).

As seen in Figure 2.13, the problems with of curvature-based approach manifest

themselves again in the axially symmetric case, but the area-based approach yields results accurate up to 3rd digit.

The conclusion to be drawn based on the tested cases (i) versus (ii) and (iii) is that the surfactant total mass is, in general, better conserved during a symmetric deformation that causes no surface translations than during asymmetric deformation with both propagating and translating surfaces. Further, when surface deformations of an axially symmetric system become complex, the area-based approach yields at least as accurate, if not better, results as the curvature-based approach.

### 2.2.4 Uneven initial surfactant distribution

The distortions brought to the mass conservation are emphasized when the initial surfactant distribution (i.e. $\Gamma$ at $t = 0$) is uneven and there exists an initial surfactant concentration gradient ($\nabla\Gamma$). A test with the initial surfactant concentration indicated by color in Figure 2.14 was done to examine adequacy of the computational means in this case. The initial surfactant surface concentration is varied from 0.08 to 0.8 over the surface, the minimum being on the "vertical equator" of the sphere and the maximum on the "horizontal equator".

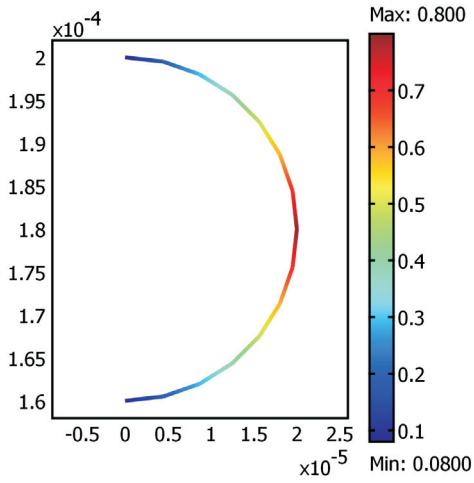


**Figure 2.14:** Tested initial surface concentration on the cross-section of a spherical surface.

Figure 2.15 illustrates the evolution of total surfactant mass computed with the different approaches in an axially symmetric system. The result is similar to the previous tests and the conclusion again clear, there appear harsh problems in the solution obtained with the curvature-based approach.

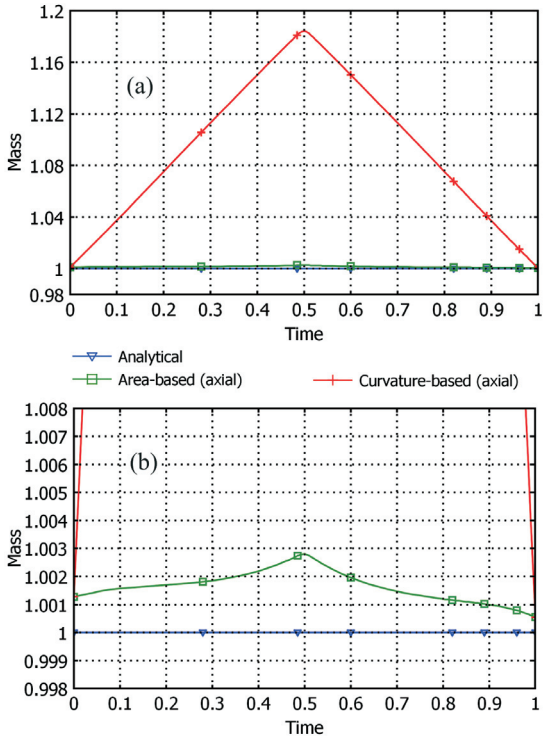


**Figure 2.15:** Evolution of total mass of surfactant on the surface of a spherical object being squeezed. Figure (b) is zoomed in from figure (a).

## 2.3 Evaluation summary

As seen in test case (i), symmetric deformation of a spherical or cylindrical surface, both the curvature-based and area-based approaches yield good results. However, in axially symmetric systems, when local surface curvature changes over time and there exists a notable surface velocity divergence, the curvature-based approach becomes inaccurate, though in a systematic way, which indicates that the error is in the implementation, not in the principles. Figures 2.8 and 2.9 illustrate further that the problem does not lie in either how local surface area or how surface curvature are computed, since both computational properties coincide well with respective analytical values. Analyzing figures 2.8–2.10 further enables deducing that the problem is somewhere in the curvature-based implementation of surfactant concentration computation or surface velocity divergence computation.

On the other hand, the curvature-independent, area-based approach seems to simultaneously yield reasonably accurate results in all of the tested cases. This approach is but lighter to compute also easier to implement, which of course

makes it less prone to human error and also therefore preferable at least in the examined cases.

The presented examination shows that when the surface curvature changes and the surface velocity direction deviates from the surface normal direction, the curvature-based surfactant mass balance computation is hard to implement accurately - for a reason or another. Such a result has constitutive significance when applying the surfactant mass balance equations in practice to model real systems with arbitrary geometries and surface movement. The observation is even more relevant since the computation clearly does function and produces results, yet the results being wrong. By only running an arbitrary real-world model without a possibility to compare computational results with analytical ones, the problem might go unnoticed eventually leading to an incorrect model. Unfortunately, though a problem was clearly indicated and located, a fix to the problem could not but fortunately, a means to avoid the problem could be given.

From the computational aspect, the area-based approach is beneficial since it requires no extra dependent variables to be defined and thus adds no extra degrees of freedom to be solved during model solution. In the curvature-based implementation two such variables are brought to the system, namely the *auxiliary normal components*. These need to be solved as dependent variables because the direct differentiation of the ALE-model boundary normal vector components would yield too noisy results, and bring significant instability to the model.

This all been said, it should be noted that in all the tests above, the *total surfactant mass* was considered as a measure of adequacy for the applied methods. When creating the practical application part of this text it was noted that also the area-based surfactant mass balance computation mechanism has its problems, related to the non-physical movement of the ALE boundary. Since the boundary velocity, when applying the ALE method, is computed partially upon the Laplacian smoothing mechanism, local surface area change, and thus local area-based surfactant concentration may also change non-physically. These aspects will be discussed further in Section 3.

# Chapter 3

# Practical application of mass balance modelling

This section describes briefly a typical modelling problem where both curvature computation and local surface area computation can - and either one must - be applied to computing local surfactant mass balance on a shape-changing surface. The case process is that of electrolytic filling of microscale features[1] on multilayered printed circuit boards, and the area-based approach for surfactant mass balance computation is demonstrated in modelling this process. The model results are also compared with empirical measurements.

To accompany the practical section, aspects related to surfactant mass balance computation in the microvia fill process, as well as in the process known as damascene electroplating for creating on-chip interconnections in microchip manufacturing are compared. The microscale process, i.e. via filling applied in manufacturing of multilayered circuit boards (MLBs) is closely related to the process of filling trenches and vias in microchip manufacturing. There are differences between the two, however, the most obvious ones being the ca. hundred-fold difference in scales of dimension and plating time. The motivation for discussing both processes is that during the past decade, the latter process has been studied and modelled significantly more extensively than the former, in e.g. [14, 1, 15, 16, 17, 18] to mention a few.

The inspected problem domains contain both an axially symmetric problem, namely that of filling a cylindrical via hole, and a plane symmetric problem,

[1] The word *feature* is used to denote any deliberately made shape, e.g. an excavation or a protrusion, on the electroplated surface.

which is that of filling a trench. The problem scales vary significantly both in spatial and temporal dimensions. For instance, where as a microvia is approximately 100–150 µm in width and perhaps 50 µm in depth, a nanoscale feature is usually something from ten to ten thousand nanometers in width and depth. Figure 3.1 shows the cross-sectional image of both a typical partially filled (a) microvia and (b) nanovia [19]. Temporally, the nanoscale process is of course also much faster than the micrometer scale process, the former typically lasting from one to tens of minutes and the latter from a half-an-hour to over an hour. This means proportional shape deformations are much faster and larger in the former process than in the latter, but on the other hand, much more mass is transferred in the latter type process.

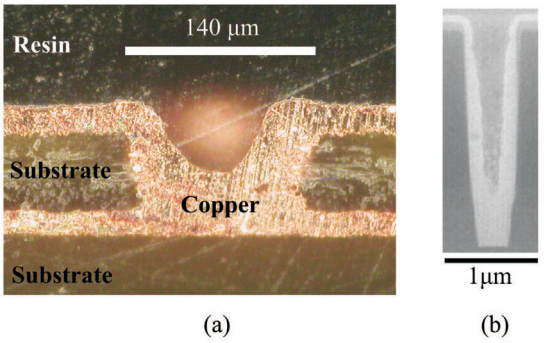


**Figure 3.1:** Cross-sections of of a typical partially filled microvia (a) and nanovia (b). Figure (a) is taken with a digital camera through an optical microscope, whereas figure (b) is a scanning electron microscope photograph.

## 3.1 The microvia filling process model

A thorough description of mass transfer, thermodynamics and electrochemistry related to, as well as a general process description of a typical microvia filling process as encountered in MLB manufacturing can be read for example in [20].

The model presented here is based on work reported in detail in [8, 21, 22] and only a compact overview of the model is given here. In this document, however, some improvements that are detailed later have been made to the previously reported models and in this text special emphasis is given to discussing the modelling of surfactant behavior on the shape changing surface.

**The physical system** The system under inspection here consists of two copper metal electrodes and a plating bath with aqueous $CuSO_4$ (760 mol/m$^3$) and

$H_2SO_4$ (1000 mol/m$^3$). Direct current is used for plating and the control current density on the cathode surface is approximately 215 A/m$^2$. Chloride ions ($Cl^-$) are added to the bath by using hydro-chloric acid, and two organic additive chemicals are added to the bath to obtain the desired fill result.

The organic additive chemicals are surface-active species which generally govern the deposit growth process and function in two distinct ways; one of them inhibits copper deposition and the other cancels the effect of the inhibiting one. Thus the chemicals are denoted with the generic names, suppressor and accelerator, respectively.

In general, the additive chemicals can be various, their composition may vary and their affecting mechanisms very diverse. Several suppressor compositions have been reported, polyetylene glycol (PEG) or polypropylene glycol (PPG) in co-action with the $Cl^-$-ion, Janus Green B (JGB) and thiourea, benzotriazole (BTA) [15, 23] are common. The diversity of reported accelerator chemicals is smaller, mainly bis(3-sulfopropyl) disulfide (SPS) and mercaptopropane sulfonic acid (MPSA), a derivative of the former, have been reported [15]. For a more thorough overview of reported additive compositions and their affecting mechanisms the reader is may refer to [20]. In this case-study, the exact additive system is not and can not be in a key role, since the exact composition of the additive chemicals is not known, which is a typical situation for the applicant of proprietary electroplating baths.

**Model overview** Only the half with $x \geq 0$ of the two-dimensional cross-section of the diffusive liquid layer above the plated microvia is actually modelled. This geometrical reduction can be made thanks to axial symmetry of the microvia system. The modelling domain geometry can be seen in Figure 3.2 (b). The horizontal top boundary (i) is the interphase between the agitated bulk solution, where convective mass transfer dominates, and the stagnant solution layer next to the cathode surface, where diffusive mass transfer is dominant. The vertical right boundary (ii) is considered a symmetry-and-insulation boundary, representing a plane over which all variables are symmetric or do not change. The bottom boundary (iii) is the moving cathode surface that is being plated and at $x = 0$ is the symmetry axis (iv). Figure 3.2 (a) shows a SEM image of a typical barrel-type microvia in the very early stage of the filling process.

When modelling a trench-like feature the model geometry can also be kept as illustrated in Figure 3.2. This is because the trench feature, in addition to hav-

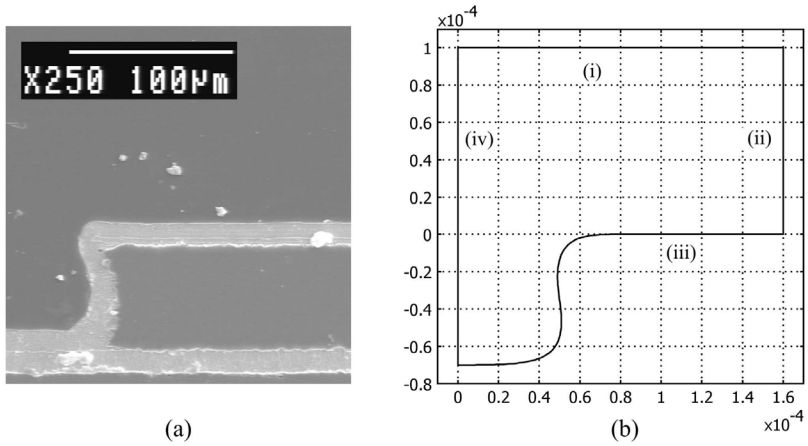


**Figure 3.2:** A cross-section image of a typical barrel-type microvia in the beginning of the fill process (a) and the initial microvia model geometry (b). The model geometry is a half of the cross-section of the cylindrical microvia and the diffusive solution layer covering the via.

ing a symmetry plane perpendicular to its length dimension, also is symmetric over the plane spanned by its length and depth dimensions. Hence, by merely modifying the model equations, the model geometry presented here can be applied to model plane-symmetric trench-like features as well as axially symmetric via hole features. The necessary equation modifications are explained on behalf of the area element computation in Appendix A.

The current density on the plated surface follows a modified bi-directional Butler-Volmer system, which is assumed symmetric in respect to the electrodes and thus obtains the form of (3.1).

$$i_c = -2i_0\mu_{\mathrm{Cu}}^{\beta}\mu_{Add}^{\frac{\beta}{\alpha_c}}\sinh\left(k\beta H\right) \tag{3.1}$$

In (3.1) $\alpha_c$ is the cathodic apparent transfer coefficient of the electrochemical reaction $\mathrm{Cu}^{2+} + 2\mathrm{e}^- \rightleftharpoons \mathrm{Cu}$ (s) and $\alpha_a$ that for the anodic reaction. $\beta$ is a shorthand, $\beta = \alpha_c\alpha_a/(\alpha_c+\alpha_a)$, $i_0$ is the system exchange current density and $k$ is $z_{\mathrm{Cu}}F/RT$ for the system where $z_{\mathrm{Cu}}$ is 2, $F$ is Faraday's constant, $T$ is 298K, $R$ the ideal gas constant and $k$ thus 78 1/V.

The $\mu$-terms are chemical terms which depict the effect of concentration of the associated species on the cathode surface and they are given in (3.2) and (3.3).

$$\mu_{\mathrm{Cu}} = \frac{c_{\mathrm{Cu}}}{c_{\mathrm{Cu}}^b}\left(\frac{c_{\mathrm{Cu}}+c^{lim}}{c_{\mathrm{Cu}}^b+c^{lim}}\right)^{-0.5554} \tag{3.2}$$

$$\mu_{Add} = (1 - \theta_{Supp})\theta_{Acc} \tag{3.3}$$

In (3.2) $c_{\mathrm{Cu}}$ is the $Cu^{2+}$ ion concentration and $c^b_{\mathrm{Cu}}$ that in the bulk solution. $c^{lim}$ is a limit concentration below which all ionic solutions are considered infinitely dilute, 3.5 $\mathrm{mol/m^3}$. This factor is added to solve computability problems that arise when the Cu(II) ion concentration in the cathodic boundary vicinity is very close to zero and the solver may pick negative concentration values for initial guess of solution iteration. The situation is referred to as copper depletion.

In (3.3) $\theta_i$ are used instead of $\Gamma_i$. $\theta_i$ is the proportional surface coverage of additive $i$ on the cathode, calculated as $\theta_i = \Gamma_i/\Gamma_i^{sat}$, where $\Gamma_i^{sat}$ is the surface concentration at its maximum (at saturation). Using $\theta$ instead of $\Gamma$ is only for convenience, as will soon become clear. Note that whereas $c_{\mathrm{Cu}}$ is a variable defined everywhere in the modelling domain, $\theta_i$ are defined only on the cathodic boundary and thus so called boundary variables.

$H$ in (3.1) is a voltage between the anode and cathode determined by the plating system galvanostat, based on the desired cathode current density, $i_{target}$. Because the vast majority of the cathode (∼99.8% to 100.0%) is flat copper metal surface, whose area is known, $i_{target}$ corresponds well to the current density on the flat copper surface where the modelling geometry doesn't essentially change. Hence, $H$ can be calculated by setting $i_c = i_{target}$ in (3.1) on the flat copper surface and solving $H$ as in (3.4).

$$H = \frac{1}{k\beta}\sinh^{-1}\left(\frac{i_{target}}{-2i_0\mu_{Add}^{\frac{\beta}{\alpha c}}\mu_{\mathrm{Cu}}^{\beta}}\right)\Bigg|_{flat} \tag{3.4}$$

Equation (3.4) enables solving the model without explicitly computing reaction overpotentials of copper reduction and oxidation. Further, the galvanostatic control essentially determines that it is not as much the absolute values of chemical terms inside and outside the via, as it is their proportional relations, that determine how the fill process proceeds. This will become apparent when comparing surfactant concentrations inside and outside the via during the filling process.

Mass transfer of the bulk solution ions, namely $Cu^{2+}$, $SO_4^{2-}$, $H^+$ and $HSO_4^+$ ions follows the Nernst-Planck system of diffusion and migration (3.5) with the electroneutrality condition (3.6).

**Table 3.1:** Diffusivity estimates ($m^2/s$) for bulk solution species.

| | | |
|---|---|---|
| $D_{Cu}$ | = | $8.65 \cdot 10^{-10} - 9.12 \cdot 10^{-11} \ln(c_{Cu} + c^{lim})$ |
| $D_{SO_4}$ | = | $12.9 \cdot 10^{-10} - 13.6 \cdot 10^{-11} \ln(c_{SO_4} + c^{lim})$ |
| $D_{H}$ | = | $70.8 \cdot 10^{-10} - 35.2 \cdot 10^{-11} \ln(c_{H} + c^{lim})$ |
| $D_{HSO_4}$ | = | $10.5 \cdot 10^{-10} - 5.24 \cdot 10^{-11} \ln(c_{HSO_4} + c^{lim})$ |

$$\frac{\partial c_i}{\partial t} = \nabla \bullet (-D_i \nabla c_i - z_i F u_i c_i \nabla \varphi) \tag{3.5}$$

$$\sum z_i c_i = 0 \tag{3.6}$$

Mass transfer of the additives in the solution is modelled based on Fick's law of diffusion (3.7).

$$\frac{\partial c_i}{\partial t} = \nabla \bullet (-D_i \nabla c_i) \tag{3.7}$$

In (3.5)–(3.7) $c$ is concentration ($mol/m^3$), $D$ is diffusivity ($m^2/s$) and $u$ is mobility ($mol \cdot s/kg$), which is computed upon the Einstein relation $u_i = D_i/RT$. $\varphi$ is electric potential (V), explained below. Estimates for species' diffusivities are given in table 3.1.

Boundary conditions for mass transfer in the solution domain are set as (i) constant concentration on the bulk solution boundary, (ii) zero mass net flux over the symmetry boundaries and (iii) removal of species on the cathode boundary according to rate of reduction of consumption of $Cu^{2+}$ ions (3.1) or the additives (3.8), resepectively.

The electric potential $\varphi$ in the solution layer is computed based on the electric field equation $-\sigma \nabla^2 \varphi = 0$, where $\sigma$ is electrolyte conductivity (S/m). The electric field equation is restricted on the model domain boundaries by (i) a known potential on the bulk solution boundary, (ii) zero current over the symmetry boundaries and (iii) a defined current density on the cathode (3.1).

**The additives** In an electroplating system, the surfactant additives can not be assumed to be completely insoluble - quite the contrary, they are very soluble to the electrolyte. Therefore the adsorption and desorption terms described in

(2.4) must be included in the mass balance equations. Whether the surface diffusivity term, introduced in (2.3) has to be incorporated to the model or not depends totally on how large is the surface diffusivity coefficient $D^s$ in respect to the other mass flux terms. This will be discussed in the end of this section, along with the nanoscale process.

The additives' mass balance *between the electrolyte solution and the cathode surface* is given as a mass flux (3.8), where $i$ is either the suppressor ($Supp$) or the accelerator ($Acc$).

$$N_i = k_i^{cons} c_i + \Gamma_i^{sat} N_i^{ads} \tag{3.8}$$

Here $k_i^{cons}$ is the rate of the consumption reaction of additive $i$ in the vicinity of the cathode surface. $\Gamma_i^{sat}$ is the maximum surface concentration of species $i$, which multiplies the adsorption flux of $i$ in order to scale the flux formulated in terms of proportional surface coverage to an actual mass flux. The adsorption fluxes $N_i^{ads}$ are given formulated as (3.10).

The fluxes (3.8) effectively remove additives from the solution in the very vicinity of the cathode, thus creating a continuous additive flux from the bulk solution to the cathode surface and eventually away from the solution. The flux is assumed to be dominated by a first order consumptive reaction as seen in the first right-hand side term in (3.8). The surface-adsorbed additives may be consumed by either a chemical decomposition reaction or due to being incorporated into the deposit.

Consecutively, the mass balance of additives *on the cathode surface* is based on the mass conservation law and is given by (3.9).

$$\frac{\partial \theta_{Supp}}{\partial t} = N_{Supp}^{ads} - N_{Supp}^{des} + D_{Supp}^s \nabla_T^2 \theta_{Supp} \tag{3.9a}$$

$$\frac{\partial \theta_{Acc}}{\partial t} = N_{Acc}^{ads} - N_{Acc}^{des} + D_{Acc}^s \nabla_T^2 \theta_{Acc} - \theta_{Acc}(\mathbf{v} \bullet \mathbf{n})(\nabla \bullet \mathbf{n}) \tag{3.9b}$$

In (3.9) $t$ is time (s) and $\nabla_T$ is the tangential differential operator. The $N_i^{ads}$ and $N_i^{des}$ terms describe the adsorption and desorption fluxes of species $i$ to and from the surface. $D_i^s$ is the diffusivity of $i$ along the surface. In (3.9b) there is the curvature term $(\nabla \bullet \mathbf{n})$ multiplied with the surface normal velocity $(\mathbf{v} \bullet \mathbf{n})$, included to account for the CEAC effects.

Equations (3.9) and (3.10) together realize the following principles according to which the mass balance of the additives is assumed to form. The additives adsorb on the cathode surface and a once adsorbed additive particle is assumed to desorb from the surface in such a form that it can never be effective again (destructive desorption). The accelerator additive may surpass the suppressor additive on the cathode as shown in (3.10a) but the contrary is not possible (3.10b). The adsorbed accelerator additives move with the moving surface and due to surface curvature, the amount of adsorbed additives per unit area (the additive surface concentration) tends to decrease on convex surfaces and increase on concave surfaces. Additionally, the additives move diffusively along the surface, against their surface concentration gradient $\nabla_T \theta_i$, represented by the third term in (3.9).

Equations (3.10a) and (3.10b) show how the additives interact as described above.

$$N_{Supp}^{ads} - N_{Supp}^{des} = k_{Supp}^{ads} c_{Supp} + (1 - \theta_{Supp} - \theta_{Acc}) - k_{Supp}^{des}(\theta_{Supp} + \theta_{Acc}) \quad (3.10a)$$

$$N_{Acc}^{ads} - N_{Acc}^{des} = k_{Acc}^{ads} c_{Acc}(1 - \theta_{Acc}) - k_{Acc}^{des}\theta_{Acc} \quad (3.10b)$$

Boundary conditions for the mass balance equations on the cathode surface (3.9) are given by symmetry. Zero net mass flux of additives is assumed over the surface layer boundaries.

The formulation for accelerator mass balance (3.9b), the adsorption-desorption flux system (3.10) as well as their eventual impact on the electrode equation (3.3) are similar to those reported in e.g. [24]. Yet, there are notable differences, such as the formulation of (3.9a) and (3.8). Whereas e.g. [24] equates the removal flux of additives with the adsorption on the surface, the model in this thesis also considers consumptive fluxes of additives on the cathode surface, included in $N_i$. Though adding the fluxes $N_i$ into the system as such may be argued, consumption of additives as a phenomenon may not, and adding $N_i$ essentially simplifies and stabilizes the model. The consumption of additives through a process of adsorption and destructive desorption on the cathode surface can also be modelled based on these two separate fluxes. However, in this case, much larger, well balanced, fluxes are needed to obtain the desired total flux of additives, which results in a significantly less stable model. Further, this model assumes the asorption-desorption dynamics of the suppressor additive is

fast and thus no surface deformation-dependent terms are included in the suppressor surface mass balance equation. This again simplifies and clarifies the equation system as well as the roles of the additives.

**The moving geometry (ALE)** The computational model is implemented as the models described earlier in this text, as a two-dimensional ($x$, $y$) finite element model, applying the arbitrary Lagrange-Eulerian method for solving the movement of the cathode surface and the whole geometry. Laplacian smoothing is applied for computing the moving mesh points. The initial model mesh can be seen in Figure 3.3 on the left.

The mesh velocity is $\mathbf{v} = (v_x,\ v_y)^\mathsf{T}$ and the tangential velocity of the cathode surface (i.e. the surface velocity in the direction of the surface itself) is thus $\mathbf{v}_T = (\mathbf{I} - \mathbf{n}\mathbf{n}^\mathsf{T}) : \mathbf{v}$, where $\mathbf{n}$ is the surface outward normal vector ($n_x$, $n_y$).

Accordingly to the ALE method, the moving mesh is formulated as a set of dependent geometry variables, denoted ($x$, $y$), dependent on the independent coordinates system ($X$, $Y$) and time $t$. The initial mesh configuration is seen in Figure 3.3 on the left. The mesh movement is evoked by the cathode boundary movement, which is constrained as in (3.11), corresponding to the copper deposit growth.

$$\mathbf{n} \bullet \mathbf{v} = i_c \frac{M_{\mathrm{Cu}}}{z_{\mathrm{Cu}} F \rho_{\mathrm{Cu}}} \tag{3.11}$$

In (3.11) $M_{\mathrm{Cu}}$ is molar mass of copper (63.55g/mol) and $\rho_{\mathrm{Cu}}$ density of copper, 8960kg/m$^3$.

Mesh velocity on all other but the cathode boundary is constrained to zero in the x-axis' direction. Further, in order to keep the diffusive layer thickness constant, mesh velocity in the y-axis' direction on the bulk solution boundary (i) is also constrained to equal that of the flat cathode surface.

The moved grid point coordinates ($x$, $y$) are computed based on Laplacian smoothing, by solving (3.12) with boundary condition (3.11) on the cathode boundary, and applying the ALE transformation (3.13) for all variables $f$ to be solved in the moving coordinate system, also for $\mathbf{v}$ in (3.12).

$$\nabla^2 \mathbf{v} = 0 \tag{3.12}$$

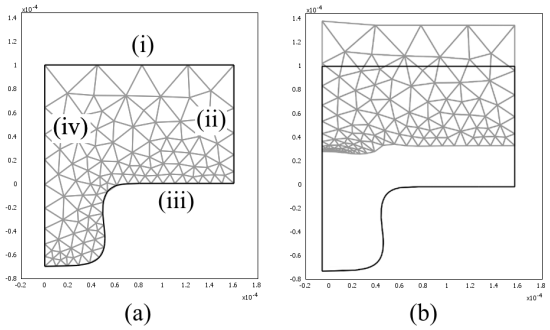


**Figure 3.3:** The model computing grid (mesh) and boundaries initially (a) and the moved grid as well as the deformed boundaries in the end of computation (b).

$$\nabla f = \frac{1}{\det(\mathbf{J})} \begin{pmatrix} \frac{\partial y}{\partial Y} & -\frac{\partial y}{\partial X} \\ -\frac{\partial x}{\partial Y} & \frac{\partial x}{\partial X} \end{pmatrix} \begin{pmatrix} \frac{\partial f}{\partial X} \\ \frac{\partial f}{\partial Y} \end{pmatrix} \tag{3.13}$$

$$\det(\mathbf{J}) = \frac{\partial x}{\partial X}\frac{\partial y}{\partial Y} - \frac{\partial x}{\partial Y}\frac{\partial y}{\partial X} \tag{3.14}$$

It is evident that according to (3.11) and (3.12), only the cathode surface normal velocity is defined and the rest of the mesh movement on the cathode boundary is computed upon the Laplacian smoothing equation. Further, since when solving (3.12) numerically, the components of $\mathbf{v}$ on cathode boundary are not solved individually but only the scalar $\mathbf{v} \bullet \mathbf{n}$ is equated with the corresponding scalar growth velocity of a planar copper deposit given in (3.11), the movement of mesh points on the cathode boundary is most likely non-physical to some extent.

The above explained background of non-physical movement of the cathode boundary is considered as the explanation for why the area-based mass balance computation yields non-functioning results when put into practice in the viafill electroplating model. However, now the only curvature-containing (thus also non-physical) equation (given in [24]) yields well-functioning results. The reason is considered to be that curvature of the surface boundary is not mesh-point-dependent in the same manner as the element area term is and thus a formulation with only the surface curvature best describes the strongly local surface-deformation dependent mass fluxes, appearing in the microvia filling process. In general, it is considered that during the copper deposition process, in macro scale the actual metal layer grows only directly normal to its surface and that the tangential surface velocity observed in the ALE model is only

a feature of the computational process. Though the metal lattice may grow tangential on atomic-scale, the bulk growth is always normal to surface. Thus omitting all terms dependent on surface tangential velocity may be justifiable when modelling a practical copper deposition process

**Weak form of surface coverage** When solving the microvia fill model a weak formulation is again utilized for all the PDEs being solved. This brings along several advantages, e.g. it enables solving (3.12) without actually computing the second order spatial derivatives of $\mathbf{v}$. Further, adding special-purpose terms, such as the area-dependent and curvature-dependent source terms does not only become possible, but is rather straight-forward, when using the weak formulation. This is naturally of primary importance, when computing surfactant mass balance.

The weak form of the surfactant mass balance equations (3.9) is given in (3.15) and is obtained similarly as the weak form for (2.13), by a procedure shown in Appendix D.

$$\begin{aligned} &\int_{\partial\Omega} \hat{\theta}_i \frac{\partial \theta_i}{\partial t} ds \\ &= \int_{\partial\Omega} \left[ \hat{\theta}_i \left( N^{ads} - N^{des} - \theta_i (\mathbf{v} \bullet \mathbf{n})(\nabla_T \bullet \mathbf{n}) \right) - \nabla_T \hat{\theta}_i \bullet (D^s \nabla_T \theta_i) \right] ds \end{aligned} \tag{3.15}$$

The $\hat{\theta}_i$ in (3.15) is a test function and also the test function derivatives have to be transformed according to the ALE transformation (3.13) when the weak form equations are solved.

## 3.2 Results and discussion

As a result of the model, the surfactant coverage evolution is depicted in figures 3.4 and 3.5 on behalf of the suppressor and accelerator additives, respectively.

Based on the surfactant coverage plots, it is clear that the surfactant coverage behaves dynamically and varies greatly over the cathode surface. The suppressor surfactant coverage is especially mobile, its coverage varying practically between zero coverage and saturation, depending on model location and process

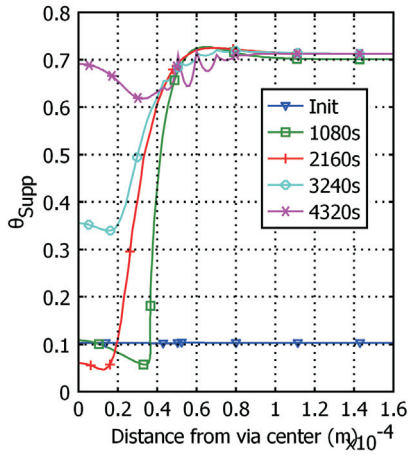


**Figure 3.4:** Proportional surface coverage of the suppressor additive during simulation.

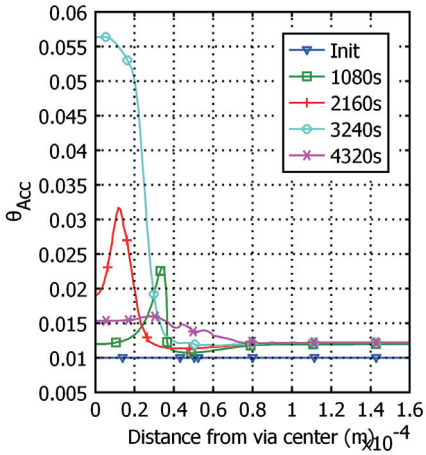


**Figure 3.5:** Proportional surface coverage of the accelerator additive during simulation.

time. The temporal behavior of the chemical term $\mu_{Add}$ in (3.1) is plotted over the simulation time span in Figure 3.6. The plot shows how $\mu_{Add}$ in the center of the microvia bottom behaves in proportion to $\mu_{Add}$ outside the microvia, on the level board surface.

The surfactants do govern the deposit evolution process but their behavior in the system *per se* is not as interesting as the final result of the model, which is the estimate for how the actual microvia filling process proceeds. To inspect this, the deposition evolution estimated by the model is given in Figure 3.7. Note that in order to improve visualization and ease comparison, these images have been mirrored over the model symmetry axis – the left-hand side of the picture is identical but mirrored to that on the right-hand side. The estimated images

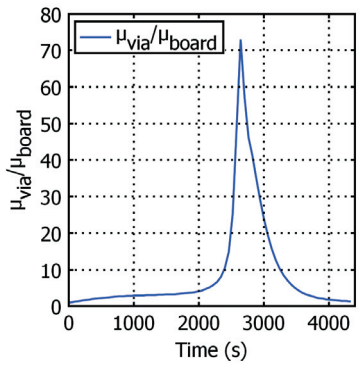


**Figure 3.6:** Evolution of chemical term $\mu_{Add}$ inside the via in respect to the same term outside the via, on the flat board surface.

may be compared to the cross-sectional images of filled microvias in Figure 3.8, obtained from via fill tests with increasing plating time. Similarly, Figure 3.9 shows how the copper deposit thickness grows during plating as estimated by the model and measured in the experiments.

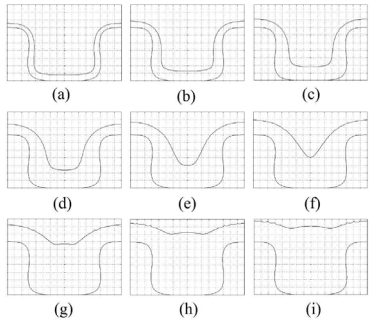


**Figure 3.7:** Estimated cross-sectional profile as output by model during via filling. The images are created by mirroring the model domain over its symmetry axis. Simulation times are 12 (a), 18 (b), 24 (c), 30 (d), 36 (e), 42 (f), 48 (g), 54 (h) and 60 (i) minutes.

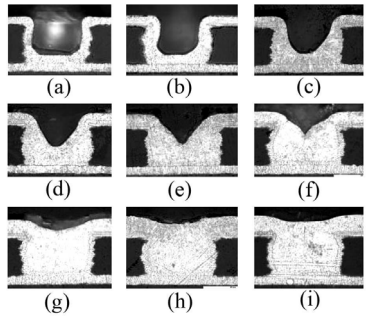


**Figure 3.8:** Images of cross-sectioned micro vias filled in via fill experiments. Fill times are 12 (a), 18 (b), 24 (c), 30 (d), 36 (e), 42 (f), 48 (g), 54 (h) and 60 (i) minutes. Pictures courtesy of Aspocomp.

The figures 3.7, 3.8 and 3.9 show a good correspondence between the model and

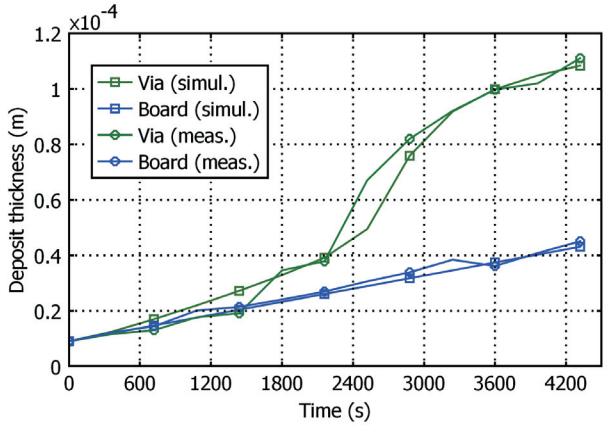


**Figure 3.9:** Copper deposit thickness during via filling as estimated by model and as observed in via filling tests.

the measurements. However, the model can clearly be seen to slightly lag behind on deposit growth rate in the via center during the period of fastest growth, between 1500 to 3000 seconds. The surfactant coverage behavior can not of course be compared with measurements from experiments and therefore only the secondary result of the fill process estimate must be considered a measure for model adequacy.

The model responds to parameter changes and is stable in the required operating area. This enables proper tuning and parametrization of the model in case of process changes. The parameter values applied in the documented model are listed in Table 3.2. The additives' reaction rate coefficients for adsorption and desorption have been chosen by fitting them with the assumed the adsorption-desorption steady state surface coverage levels as well as with adsorption reaction times found in the literature. Consumption rate coefficients have been chosen so the reaction rate corresponds qualitatively with the background assumptions and with the time scale of the practical process. Surface diffusivity values have also been chosen to logically match with the process time scale. No reference data for these coefficients could be found. Other parameters are based on either literature data, practical data or values estimated upon experiments.

The model consists of ca. 200 elements and 4900 degrees of freedom to be solved for and a time-dependent FE solver with a direct UMFPACK linear system solver is used to solve the model. With a simulation time of 4320 seconds, the model requires ca. 200-300 seconds to solve on a regular desktop PC (2.66GHz, 1Gb). The computing time is adequately short compared to

**Table 3.2:** The microvia fill model parameters and their units.

| Symbol | Value | Unit | Note |
|---|---|---|---|
| $T$ | 295.15 | K | |
| $c^b_{\mathrm{Cu}}$ | 760 | $\mathrm{mol/m^3}$ | |
| $c^b_{\mathrm{H_2SO_4}}$ | 1000 | $\mathrm{mol/m^3}$ | |
| $c^b_{Supp}$ | 0.05 | $\mathrm{mol/m^3}$ | |
| $c^b_{Acc}$ | 0.005 | $\mathrm{mol/m^3}$ | |
| $c^{lim}$ | 3.5 | $\mathrm{mol/m^3}$ | Chosen |
| $i_{target}$ | 215 | $\mathrm{A/m^2}$ | |
| $i_0$ | 21 | $\mathrm{A/m^2}$ | Identified |
| $\alpha_a$ | 0.232 | no dim. | Identified |
| $\alpha_c$ | 0.170 | no dim. | Identified |
| $k^{ads}_{Supp}$ | 45 | $\mathrm{m^3/mol/s}$ | |
| $k^{des}_{Supp}$ | 0.001 | 1/s | |
| $k^{cons}_{Supp}$ | 0.01 | m/s | |
| $\Gamma^{sat}_{Supp}$ | $4.15 \cdot 10^{-7}$ | $\mathrm{mol/m^2}$ | Data [25] |
| $D_{Supp}$ | $5.95 \cdot 10^{-10}$ | $\mathrm{m^2/s}$ | Estimated [20] |
| $D^s_{Supp}$ | $2.97 \cdot 10^{-14}$ | $\mathrm{m^2/s}$ | Fit, see Sect. 3.3 |
| $k^{ads}_{Acc}$ | 0.005 | $\mathrm{m^3/mol/s}$ | |
| $k^{des}_{Acc}$ | 0.002 | 1/s | |
| $k^{cons}_{Acc}$ | $6.0 \cdot 10^{-9}$ | m/s | |
| $\Gamma^{sat}_{Acc}$ | $1.6 \cdot 10^{-5}$ | $\mathrm{mol/m^2}$ | Data [25] |
| $D_{Acc}$ | $4.05 \cdot 10^{-10}$ | $\mathrm{m^2/s}$ | Estimated [20] |
| $D^s_{Acc}$ | $2.02 \cdot 10^{-14}$ | $\mathrm{m^2/s}$ | Fit, see Sect. 3.3 |

the process flow time, around an hour, which makes the model applicable for control applications on the production line. The model provides a good basis for monitoring system development and by improving parametrization as well as model geometry, computing time of the model can be reduced.

## 3.3 Comparison between microscale and nanoscale processes

This section describes briefly based on physical inspection and simulation studies, how the most essential differences between the processes of electrolytic feature filling in microchip manufacturing and in MLB production affect their computational treatment.

### 3.3.1 Mass transfer-related spatial differences

There are two matters related to general mass transfer of soluble surfactant species that should be discussed in respect to whether the process in question is a microscopic or nanometer-scale deposition process. These are (i) dimensions of the surface features in proportion to the diffusive electrolyte layer and (ii) surfactant surface diffusivity.

Also the effect of convective mass transfer inside the features may have to be considered, especially if the microvia width increases significantly above 100 µm and strong agitation of electrolyte is applied in the plating bath. In this study however, even the microscopic features are assumed small enough so to enable neglecting convective mass transfer of species inside the features. In the nanoscale electroplating systems this certainly is the case without exception.

**Size of feature vs. diffusion layer thickness** The diffusion layer thickness, which is the distance that the surface-active species has to traverse by diffusion in the electrolyte surrounding the surface before reaching the surface, is effectively determined by flow velocity of the electrolyte and the surfactant diffusivity.[2] This means that the diffusion layer thickness is the same no matter what the dimensions of the feature to be filled are. Depending on bath agitation and species in question, the diffusion layer thickness is approximately 20–100 µm.

[2] For a more thorough description of affecting factors see e.g. [12]

When the species are consumed or produced on the surface, a concentration gradient between the diffusion layer boundary and the surface prevails. If the consumption of species is roughly equal per unit surface area, then, the further away from the diffusion layer outer boundary the surface is, the lower is the species concentration there. Vice versa if the process at the surface produces species.

In microvia filling the feature depth is generally comparable in magnitude to the diffusion layer thickness, whereas in the damascene process the feature size is only a fraction of the diffusion layer thickness. Hence, it can be said that whereas in microvia filling the species concentration in the solution on the level board and inside the via may vary significantly due to differences in distance traversed by the species by diffusive mass transfer, in the nanoscopic processes these levels of concentration are roughly equal.

**Surface diffusion effects** Since in the nanoscopic process the absolute dimensions are hundred-fold smaller than in the micrometer-scale process, surface diffusion effects should be stronger and should thus be considered more carefully. Empirically (computationally) it can be verified that a mass flux due to surface diffusion, that might go unnoticed in the microscopic domain, might in the nanoscopic process practically even up surface concentrations everywhere. For illustration, the effect of the surface diffusivity coefficient $D^s$ is compared in 3.10, in both the microscopic (a) and nanoscopic (b) domain. The time scale of both illustrations is the same. The model is the plane-symmetric case of the third example case introduced in Section 2.2, a cylindrical object being deformed asymmetrically.

Due to asymmetric deformation, the surfactant surface concentration becomes uneven over the surface but as is seen in Figure 3.10, surface diffusion evens up surface concentration differences. The larger the surface diffusivity $D^s$, the smaller the differences in surface concentration.

Whereas a surface diffusivity of $1 \cdot 10^{-14}$ m$^2$/s in the nanoscopic model significantly decreases proportional surface concentration differences over the surface in question, such a diffusivity has practically no effect in the microscopic model (the lines collide). Then again, given a surface diffusivity whose effects are somewhat notable in the microscopic model, the nanoscopic model becomes completely insensitive to local area change. (The plot lines are horizontal at all times.)

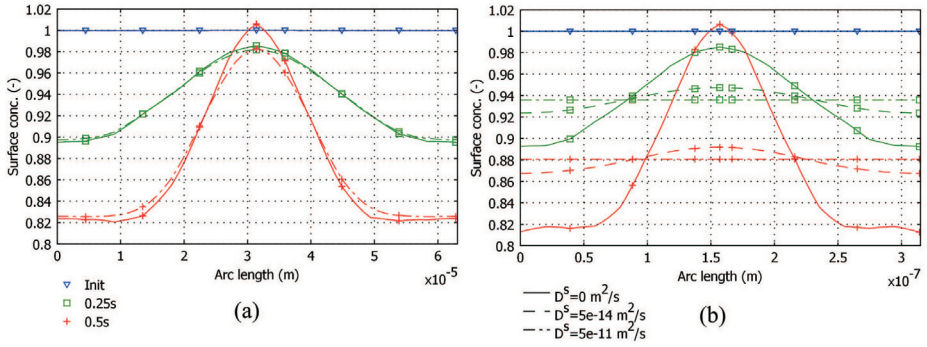


**Figure 3.10:** The effect of surface diffusivity in both (a) microscopic and (b) nanoscopic domains. The plotted simulation time instants are denoted with line markers, in ascending order, with $\triangledown$ ($t$=0s), $\square$ ($t$=0.25s) and $+$ ($t$=0.5s). Analyzed surface diffusivity coefficients are denoted in ascending order with line types "solid" ($D^s$=0 m$^2$/s),"dashed" ($D^s$=5 $\cdot$ 10$^{-14}$ m$^2$/s) and "dash-dot" ($D^s$=5 $\cdot$ 10$^{-11}$ m$^2$/s).

### 3.3.2 Reaction rate-related temporal differences

The variety of adsorption and desorption reactions as well as reactions where surfactants interact on the active surface is numerous. Each of these have their own reaction rate-related dynamics. The time within which e.g. a surfactant reaches equilibrium coverage at given conditions, compared to the actual plating process duration determines whether a steady-state assumption of surfactant dynamics is appropriate or not. Several studies [24, 26, 17] show that time constants of the adsorption-desorption reactions may vary widely from nearly zero to thousands of seconds. In the case of e.g. microvia filling, a process that lasts for approximately an hour, the fastest surfactant dynamics may be reduced by assuming steady state coverage at all times. Similarly, in the fast nanoscale processes, slow reactions may be assumed to essentially not proceed either forward nor backward during the process. On the other hand again, reactions whose dynamics may be neglected when inspecting the other size process, of course have to be accounted for in the other.

### 3.3.3 Differences in curvature effects

A very common and intuitive explanation for the superfill phenomenon in nanoscale domains is the mechanism generally known as the *curvature enhanced accelerator coverage* mechanism (CEAC). The main principle that this mechanism relies on is such that since the accelerator additive is attached firmly on the deforming surface, it gets accumulated where ever the surface is concave (in

respect to its movement velocity) and its movement velocity has a component normal to the surface. Correspondingly, the attached surfactant gets diluted where the surface is convex and moves normally relative to itself. These cases are illustrated in Figure 3.11 (a) and (b).

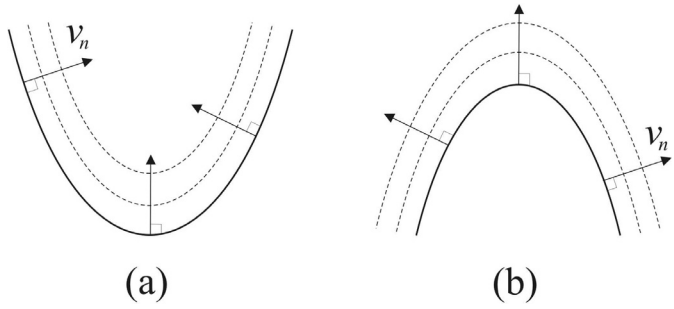


**Figure 3.11:** Movement of a (a) concave and (b) convex surface.

This CEAC mechanism is supported by the formulations for surface mass balance given in Section 2. Based on equation (2.10) the local surface concentration dynamics is clearly dependent on the surface normal velocity as well as surface curvature, and exactly as described by the CEAC mechanism.

In order to understand the mechanism fully, however, one should also pay attention to the system dimensions. Curvature, defined also as $\nabla \bullet \mathbf{n}$, has the unit $\mathrm{m}^{-1}$ and its magnitude is clearly dependent on absolute feature size. For example, when considering a cylinder surface, it is easy to understand that the surface normal divergence becomes smaller as the cylinder radius grows. The same applies also for a sphere.

Figures 3.12 (a) and (b) show the initial cross-sectional geometries of a typical trench-like feature on a microchip as well as a that of a typical microvia inside a multilayered circuit board. It is important to notice the hundred-fold difference in dimensional scales.

Figures 3.13 show the corresponding values of computed surface curvature. (Note that the nanotrench is a plane-symmetric feature, whereas the microvia is an axially-symmetric feature. If they would otherwise be identical in size, the curvature of the former would be smaller than that of the latter due to the third curvature axis of the axially symmetric system.)

It is clearly seen that curvature of the features' surfaces decrease as their dimensions increase. Yet, in respect to surfactant mass balance modelling, the surfactant surface concentration is nevertheless within the same scale in both

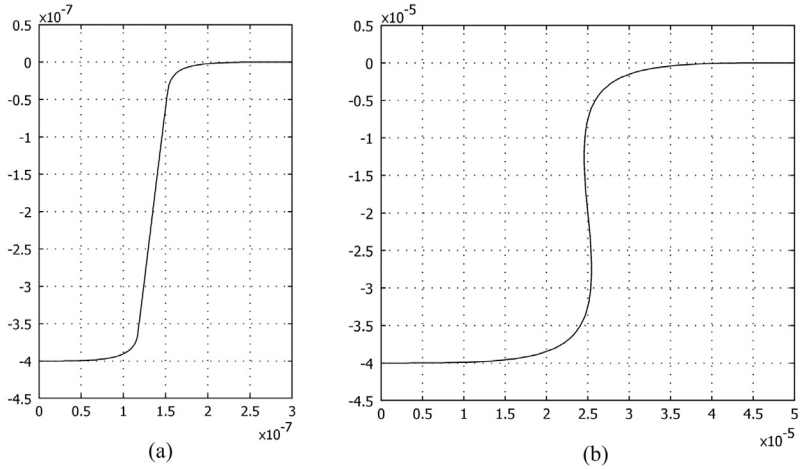


**Figure 3.12:** Initial (half of cross-section) shapes of (a) a typical nanoscale trench feature and (b) a typical barrel-like microvia.

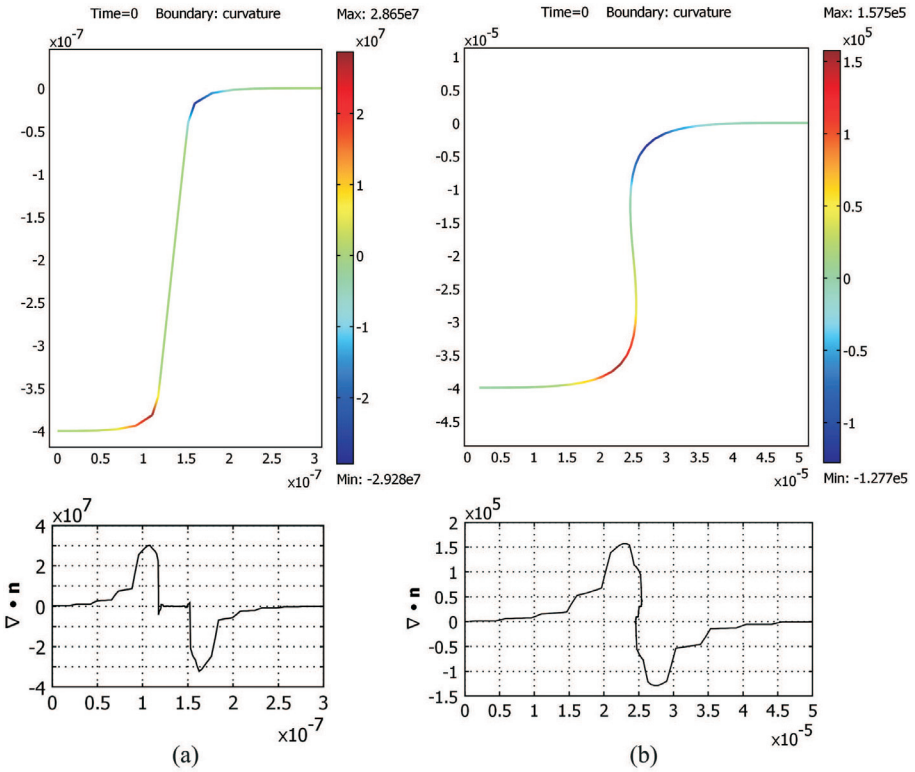


**Figure 3.13:** Surface curvature on typical (a) nanoscale and (b) microscale features.

cases. This also applies to the surface movement velocities; the deposit growth speed is determined electrochemically and usually the deposition current density is similar regardless of whether the process is that of damascene electroplating

or microvia filling. Therefore, models developed specifically for the microchip manufacturing processes do not generally apply as such to modelling processes utilized in circuit board production, and a separate inspection is required. This conclusion has also been drawn in [27] and a brief simulation study to support the deduction is presented next.

**Scale-up of nanoscale model** To confirm the conclusion drawn above, a model based on literature [24], originally developed for modelling filling of trenches in a nanoscopic domain has been implemented first in nanoscale and then modified to microscale by direct scale-up of geometry. Naturally, differences unavoidably exist between the original model described in the cited report and the one implemented here. For example, no initial values of time-dependent variables were mentioned in the report. Further, the system in the original report is based on a potentiostatic approach, whereas here the system is implemented with galvanostatic control. Also the computational implementations surely have differences but nevertheless, the results are clear and the conclusion drawn above is given fair support.

The fill results as well as the deposit front evolution are essentially different between the nanoscale and microscale simulations after scale-up of the nanoscale model. Figure 3.14 (a) illustrates the evolution of the nanoscale trench feature surface during a plating process of four minutes. The scaled-up system surface evolution is shown in 3.14 (b) for the first 120 minutes. It can be seen that the scaled-up process is fully conformal during the simulated time period. If linear scale-up were to be possible, the 100-fold magnified model should exhibit superconformal growth at the trench corners latest at $t = 6000$ seconds. Moreover, due to fully conformal growth the simulation runs into numerical problems and eventually does not converge after $t > 2$ hours.

Figure 3.15 (a) and (b) illustrate the deposit growth inside the trench feature. In figure 3.15 (b) it can be seen that the deposit growth in the microscale process is actually subconformal, which is due to increased effects of diffusive mass transfer of particularly the Cu(II) ion being most substantially consumed at the deposited surface (a phenomenon a.k.a. copper ion depletion on cathode).

This concludes the discussion on effects that differences in temporal and spatial dimensions have on model considerations when modelling a deposition including shape-changing surfaces.

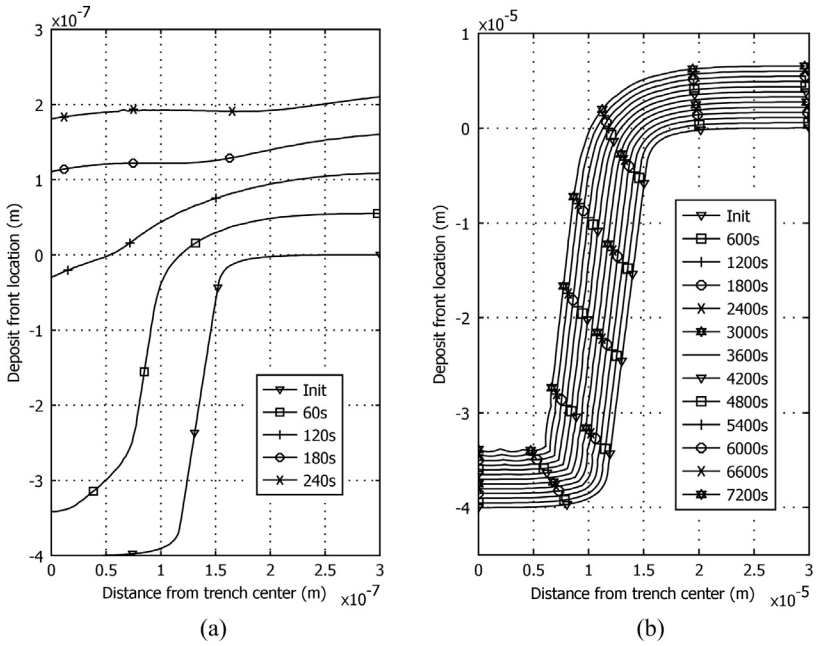


**Figure 3.14:** The deposit evolution during simulation of (a) a nanoscale and (b) a microscale process.

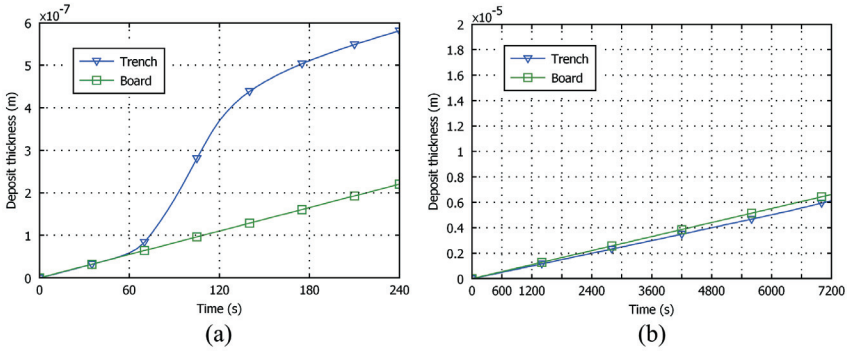


**Figure 3.15:** Deposit thickness growth on both inside the trench as well as ob the level wafer/board surface plotted during (a) a nanoscale process simulation and (b) a microscale process simulation.

## 3.4 Practical application conclusions

Several conclusions that improve a microscopic via filling process model can be drawn based on both the simulation analysis of surfactant mass balance modelling techniques as well as the comparison of microscale and nanoscale electrolytic filling process models. These conclusions are summarized below.

- A direct scale up of nanometer scale models to micrometer scale is not

feasible due to different surface chemical mechanisms and mass transport phenomena governing the system when the spatial dimensions change significantly.

  - Surface diffusivity may have a role in surfactant mass transport when the system dimensions are very small but surface diffusivity can be neglected as the spatial dimensions grow.
  - In nanoscale models the diffusion layer thickness exceeds the surface feature dimensions manifold thus enabling neglecting convection fully as well as diminishing the influence of diffusion limited mass transfer.
  - The time constants of relevant surface chemical processes must be matched with the process duration. Surfactant dynamics which are significantly faster or slower than the process under inspection can be reduced from the model by steady-state assumption.

- An area-based approach to implementing surfactant mass balance computation in a shape-changing model based on the ALE method is significantly easier to implement than a curvature-based implementation. However, since the ALE transformation enables a non-physical movement of the surface boundary, the area-based approach for surfactant mass balance may also yield non-physical modelling results. Therefore consideration and special-purpose computing mechanisms need to be used – in the microvia filling process model, this means utilizing a surfactant mass balance computation based only on surface curvature. Accuracy and consistency is lost but functionality, crucial to practical model applications, is obtained via this approach.

# Chapter 4

# Conclusions

A problem domain consisting of computing the mass balance of various kind surface-active species affecting on a shape-changing surface was gone through in terms of physical examples, the theoretical equations, numerical implementation and practical application. The common formulations of physical equations necessary to model surfactant mass transfer *from and to* as well as *on* a shape changing surface were presented first, and two conceptually different approaches to computationally implement these formulations were given to follow. A collection of necessary calculations, given in the appendices in this work, was produced to support the discussion.

The two aforementioned approaches, entitled the area-based and the curvature-based approach, were implemented computationally to simulate surfactant surface concentration behavior in models of three distinctively different two- dimensional, plane-symmetric as well as axially symmetric, shape-changing surfaces. Based on these simulations, the computational implementations were evaluated upon the criteria of total surfactant mass conservation. Several differences between the methods were found and based on the mentioned criterion, the area-based approach was found equally or more accurate in all the test cases compared with the curvature-based approach.

A practical process model of electrochemical deposition of copper in microscopic as well as nanoscopic surface features, in both plane-symmetric as well as axially symmetric geometries was implemented to examine practical applicability of the tested surfactant mass balance computation methods. A simplified curvature-based method was now found most applicable when comparing against empirical data. Though accuracy and physical consistency is compromised when apply-

ing this method, good practical functionality is obtained, which is essential in practical applications.

A discussion on how temporal and spatial dimensions should be taken into regard when modelling diffusive or convective mass transfer as well as adsorption and desorption reactions in microscopic as well as nanometer-scale physical domains was given to conclude the thesis. The discussion was illustrated and supported with several observations based on simulations of the considered cases.

The following work will focus on practical application of the developed surfactant mass balance computation method, aiming to create a model based monitoring system for quality control of the electrolytic microvia filling process output. Reliability and physical consistency of the underlying computational methods utilized in a model which is to be applied to model based estimation of any kind of a process is elementary. Therefore the effort to assure the outcomes of this thesis will also surely continue.

# Related publications

The publications listed below are directly related to the work. This work represents the most up-to-date status quo of the research but relies largely on the material given in the related publications. Though a list of publications is given here, this work is meant to be self-contained as such.

### Publications related to the thesis (in chronological order)

P1 Pohjoranta A., Tenno R., *A Method for Microvia-Fill Process Modeling in a Cu Plating System with Additives*, 2007, Journal of The Electrochemical Society, Vol. 154, N:o 10, pp. D502–D509

P2 Pohjoranta A., Tenno R., *Modelling of Surfactant Mass Balance for Microvia Fill Monitoring*, 2007, International Symposium on High Density Packaging and microsystem integration (HDP'07), IEEE, Shanghai, China

P3 Tenno R., Pohjoranta A., *An ALE Model for Prediction and Control of the Microvia Fill Process with Two Additives*, 2008, Journal of the Electrochemical Society, Vol. 155 N:o 5, pp. D383–D388

### Contributions of the author

P1 The author designed and completed the experimental work presented in the article, formulated the estimates for electrolyte electric conductivity, species diffusivity and Cu(II) ion activity given in the article, formulated the area element computation mechanism and wrote the article text.

P2 The author formulated weak-form equations for surfactant mass balance, implemented the computational model and wrote the proceedings text.

P3 The author did the related chemical experiments, designed the scripts necessary for sequential simulations applied in parameter sensitivity analysis and wrote the article text.

# References


[1] Wheeler D., Josell D., and Moffat T.P. Modeling superconformal electrodeposition using the level set method. *J. Electrochem. Soc.*, 150(5):C302–C310, 2003.

[2] J.-J. Xu, Z. Li, J. Lowengrub, and Z. Hongkai. A level-set method for interfacial flows with surfactant. *Journal of Computational Physics*, 212:590–616, 2006.

[3] S. Osher and R. P. Fedkiw. Level set methods: An overview and some recent results. *Journal of Computational Physics*, 169:463–502, 2000.

[4] S. N. Ghadiali, D. Halpern, and D. P. Gaver. A dual-reciprocity boundary element method for evaluating bulk convective transport of surfactant in free-surface flows. *Journal of Computational Physics*, 171:534–559, 2001.

[5] Moffat T.P., Wheeler D., Huber W.H., and Josell D. Superconformal electrodeposition of copper. *Electrochem. Ss. Lett.*, 4(4):C26–C29, 2001.

[6] J. James and J. Lowengrub. A surfactant-conserving volume-of-fluid method for interfacial flows with insoluble surfactant. *Journal of Computational Physics*, 201:685–722, 2004.

[7] Yang X. and James A.J. An arbitrary lagrange-eulerian (ale) method for interfacial flows with insoluble surfactants. *Fluid Dynamics and Material Processing*, 3(1):65–95, 2007.

[8] Pohjoranta A. J. and Tenno R. A method for microvia fill process modelling in a cu-cu-electrode system with additives. *J. Electrochem. Soc.*, 154(10):D502–D509, 2007.

[9] Stone H.A. A simple derivation of the time-dependent convective-diffusion equation for surfactant transport along a deforming interface. *Physics of Fluids A*, 2(1):111–112, 1990.

[10] Wong H., Rumschitzki D., and Maldarelli C. On the surfactant mass balance at a deforming fluid interface. *Physics of Fluids A*, 8(11):3203–3204, 1996.

[11] Donea J. and Huerta A. *Finite Element Methods for Flow Problems*. Wiley, 2003. ISBN 0-417-49666-9.

[12] Bard A.J. and Faulkner L.R. *Electrochemical Methods; Fundamentals and Applications*. John Wiley & Sons, 2001. ISBN 0-471-043272-9.

[13] Batchelor G.K. *An Introduction to Fluid Dynamics*. Cambridge University Press, 1967. ISBN 789-0-521-66396-0.

[14] Gabrielli C., Mocoteguy P., Perrot H., Nieto-Sanz D., and Zdunek A. A model for copper deposition in the damascene process. *Electrochim. Acta*, 51(11):1462–1472, 2005.

[15] Vereecken P. M., Binstead R. A., Deligiani H., and Andricacos P. C. The chemistry of additives in damascene copper plating. *IBM J. Res. Dev.*, 49(1):3–18, 2005.

[16] Moffat T. P., Wheeler D., Edelstein M. D., and Josell D. Superconformal film growth: Mechanism and quantification. *IBM J. Res. Dev.*, 49(1):19–36, 2005.

[17] Akolkar R. and Landau U. A time-dependent transport-kinetics model for additive interactions in copper interconnect metallization. *J. Electrochem. Soc.*, 151(11):C702–C711, 2004.

[18] Dow W-P., Huang H-S., and Linb Z. Interactions between brightener and chloride ions on copper electroplating for laser-drilled via-hole filling. *Electrochem. Ss. Lett.*, 6(9):C134–C136, 2003.

[19] Wang Z., Yaegashi O., Sakaue H., Takahagi T., and Shingubara S. Bottom-up fill for submicrometer via holes of ulsis by electroplating. *J. Electrochem. Soc.*, 151(12):C781–C785, 2004.

[20] Pohjoranta A.J. Microvia fill electroplating: a model for process monitoring development. Master's thesis, 2006. http://users.tkk.fi/u/apohjora/pohjoranta_MSc_thesis_2006.pdf.

[21] Tenno R. and Pohjoranta A. J. An ale model for prediction and control of the microvia fill process with two additives. *J. Electrochem. Soc.*, 155(5):123–123, 2008.

[22] Pohjoranta A.J. and Tenno R. Modelling of surfactant mass balance for microvia fill monitoring. pages 198–201, 2007. 2007 Intl. Symp. on High-density Packaging and Microsystem Integration (IEEE).

[23] Dow W-P., Yen M-Y., Lin W-B., and Ho S.W. Influence of molecular weight of polyethylene glycol on microvia filling by copper electroplating. *Journal of The Electrochemical Society*, 152(11):C769–C775, 2005.

[24] Moffat T.P., Wheeler D., Kim S.-K., and Josell D. Curvature enhanced adsorbate coverage mechanism for bottom-up superfilling and bump control in damascene processing. *Electrochim. Acta*, 53(1):145–154, 2007.

[25] Stangl M., Acker J. andOswald S., Uhlemann M., Gemming, T., Baunack S., and Wetzig K. Incorporation of sulfur, chlorine, and carbon into electroplated cu thin films. 2007.

[26] Dow W.-P. and Liu C.-W. Evaluating the filling performance of a copper plating formula using a simple galvanostat method. *J. Electrochem. Soc.*, 153(3):C190–C194, 2006.

[27] Barkley D.P., Callahan J., Keigler A., Liu Z., Ruff A., Trezza J., and Wu B. Studies on through-chip via filling for wafer-level 3d packaging. 2006. 210th ECS Meeting, 2006, Abstract #1642.

[28] Bonet J. and Wood R.D. *Nonlinear Continuum Mechanics for Finite Element Analysis*. Cambridge University Press, 1997. ISBN 0-521-57272-X.

[29] Crisfield M. A. *Non-linear Finite Element Analysis of Solids and Structures, Volume 1: Essentials*. John Wiley & Sons Ltd., 1991. ISBN 047195649X.

[30] COMSOL AB. *COMSOL Multiphysics User Guide and Model Library, version 3.2*. COMSOL AB, 2005. Sweden.

# Appendix A

# Equations for local area element

**Plane-symmetric Cartesian coordinates case** In the plane-symmetric case, in two dimensions the area element equation (2.7) becomes

$$A = A_0 \parallel \mathbf{n} \parallel \text{, where} \tag{A.1}$$

$A_0$ is the initial surface area element size and $(n_X, n_Y)^\mathsf{T}$ is the initial surface normal vector, which is constant. The norm $\parallel \mathbf{n} \parallel$ is computed as

$$\parallel \mathbf{n} \parallel = \left( \left( \frac{\partial x}{\partial X} n_Y - \frac{\partial x}{\partial Y} n_X \right)^2 + \left( \frac{\partial y}{\partial Y} n_X - \frac{\partial y}{\partial X} n_Y \right)^2 \right)^{1/2}. \tag{A.2}$$

The relation (A.1) is essentially only a transformation of arc length since symmetry is utilized to represent area. The transformation can also be considered as a form of the relation known as the Nanson's formula. [28, 29]

In a plane-symmetric case, the two dimensional area element is a line segment on a boundary and $A_0$ is the length of the line segment (multiplied by the structure "depth", being constant). Hence $A_0 = ((dX)^2 + (dY)^2)^{1/2}$, where $d$ denotes the computational difference operation.

**Axially symmetric, cylindrical coordinates case** When applying (2.7) in an axially symmetric system, a cylindrical coordinate system is applied and the area element $A$ as well as the initial area element $A_0$ model a belt-like segment of the surface circulating the symmetry axis. By choosing the Cartesian $X$-axis as the radial coordinate axis and the Cartesian $Y$-axis as the vertical coordinate in the cylindrical coordinate system (from which the angular coordinate is reduced to obtain a 2D system), equations for the area elements and the tangential vector can easily be formulated in the axially symmetric case.

The initial area element area is thus $A_0 = 2\pi X((dX)^2 + (dY)^2)^{1/2}$ and correspondingly, the area element equation A.1 has to be completed with the change in the modelled belt radius, namely $x/X$ (A.3).

$$A = A_0 \frac{x}{X} \parallel \mathbf{n} \parallel \tag{A.3}$$

The above equations also illustrate the physical interpretation to the norm $\parallel \mathbf{n} \parallel$ – it is the size ratio between the deformed and the original, non-deformed boundary elements.

# Appendix B

# Curvature and the ALE method

Computing a 2D boundary curvature by applying the terms given in the ALE transformation (1.1) is explained here briefly.

First the moving boundary normal vector elements are formulated as in (B.1).

$$\mathbf{n} = \begin{pmatrix} n_x \\ n_y \end{pmatrix} = \frac{1}{\| \mathbf{n} \|} \begin{pmatrix} \frac{\partial y}{\partial Y} n_X - \frac{\partial y}{\partial X} n_Y \\ \frac{\partial x}{\partial X} n_Y - \frac{\partial x}{\partial Y} n_X \end{pmatrix} \tag{B.1}$$

In (B.1) the $n_X$, $n_Y$ are the boundary outward normal vector components in the $X$ and $Y$ coordinate directions (the original coordinate system), respectively, and $\| \mathbf{n} \|$ is the norm (A.2).

Also the boundary tangential vector components can be formulated now as e.g. $t_x = -n_y$ and $t_y = n_x$, but these are not unique and the signs depend on the orientation of the boundary element.

To compute the curvature, $\nabla \bullet \mathbf{n}$, i.e. divergence of $\mathbf{n}$, the above terms can be straightforwardly differentiated computationally. However, to assure computability and to reduce numerical noise, this is done by formulating the weak form equations for a *auxiliary normal vector*, $\overline{\mathbf{n}}$ and then differentiating the auxiliary components solved from these equations, given in (B.2).

$$\begin{pmatrix} \int_{\partial\Omega} \hat{\overline{n}}_x \, (\overline{n}_x - n_x) ds \\ \int_{\partial\Omega} \hat{\overline{n}}_y \, (\overline{n}_y - n_y) ds \end{pmatrix} = \mathbf{0} \tag{B.2}$$

# Appendix C

# Implementation of tangential derivatives with COMSOL

The COMSOL Multiphysics software was applied to create and solve the microvia fill model described in this article. The computational implementation of some boundary variables may vary between computer softwares and is generally not straight forward. Thus, one set of formulations for the dependent variables defined on the cathode boundary, and their gradients, with the ALE transformation, is given here. These formulations were developed and applied to compute all terms including tangential derivatives and the discussion here is strongly software-related. The advice obtained from various COMSOL Multiphysics manuals [30] as well as the ideas given by some discussions on the Yahoo news groups forum `COMSOL_Users` must be acknowledged here.

As mentioned, the ALE method is implemented by introducing a set of dependent variables that model the deforming geometry in a reference basis. The former of this is referred to as the current and the latter as the original coordinates of the modelled domain. The original coordinates, of a two-dimensional (Cartesian or cylindrical) geometry are denoted $X$ and $Y$, and the current coordinates $x$ and $y$, for the horizontal and vertical dimension, respectively. The system is time-dependent and thus the initial condition of $(x, y) = (X, Y)$ at $t = 0$ is chosen for solution of (1.2) with (1.1).

The coordinate variables $x$ and $y$ are naturally defined everywhere in the modelling space but the proportional surface coverage variables, $\theta_i$ and the area element variables $A$ are defined only on the boundary representing the cathode surface. All variables that are defined (and continuously differentiable) on a boundary have tangential derivatives. For variables defined in the whole spatial domain, also outside the boundary like $x$ and $y$, the tangential derivatives equal the orthogonal projection (on the boundary in question) of their gradient. In the two-dimensional case, the tangential derivatives are vector variables including two terms, which are the two Cartesian components of the mentioned projection.

The tangential derivatives are given by the tangential derivative operator, here denoted $\nabla_T$ and thus for a variable $f$ defined everywhere in space $\nabla_T f = (\mathbf{I} - \mathbf{n}\mathbf{n}^\mathsf{T}) : \nabla f$ and if the tangential vector is defined e.g. $\mathbf{t} = (t_x, t_y)^\mathsf{T} = (-n_y, n_x)^\mathsf{T}$ then $\nabla_T f = \mathbf{t}\mathbf{t}^\mathsf{T}\nabla f = (\mathbf{t} \bullet \nabla f)\mathbf{t}$. Here $\mathbf{I}$ is the identity tensor, in this case the two-dimensional identity matrix, and the term $(\mathbf{I} - \mathbf{n}\mathbf{n}^\mathsf{T})$ may be called a *projection operator*.

The notations follow the convention given in Appendix B.

However, variables that are defined only on a boundary do not, in the same manner as above, have a conventional gradient which could be projected on the boundary in question. This is reasonable also physically, since the boundary variables naturally cannot have values outside the boundary and thus have no gradient pointing away from the boundary. For such boundary variables, e.g. $\theta_i$ and $A$ in this study, only tangential derivatives in the original coordinate system and on the model boundary are available. A transformation of these to the current coordinate system needs to be formulated.

For distinction, the outward boundary normal vector of the original (non-deformed) model geometry is here denoted $\mathbf{n}_0 = (n_X, n_Y)$ and $\mathbf{n}$ refers to the boundary outward normal vector of the current (deformed) model geometry. (Similarly, the subscript $|_0$ is used to refer to variables and terms in the original geometry.)

Knowing that $\nabla_T f|_0 = (\mathbf{I} - \mathbf{n}_0\mathbf{n}_0^\mathsf{T}) : \nabla f|_0 = \left(\frac{\partial f}{\partial X_T}, \frac{\partial f}{\partial Y_T}\right)^\mathsf{T}$ we can formulate $\nabla_T f$ as function of $\nabla_T f|_0$.

$$\begin{aligned} \nabla_T f &= (\mathbf{I} - \mathbf{n}\mathbf{n}^\mathsf{T}) : \nabla f \\ &= \frac{1}{\| \mathbf{n} \|^2} \begin{pmatrix} \frac{\partial x}{\partial X}\frac{\partial f}{\partial X_T} + \frac{\partial x}{\partial Y}\frac{\partial f}{\partial Y_T} \\ \frac{\partial y}{\partial X}\frac{\partial f}{\partial X_T} + \frac{\partial y}{\partial Y}\frac{\partial f}{\partial Y_T} \end{pmatrix} \end{aligned} \tag{C.1}$$

In (C.1), the differentials $\partial f/\partial X_T$ and $\partial f/\partial Y_T$ refer to the aforementioned tangential derivatives in the original coordinate system $\left(\frac{\partial f}{\partial X_T}, \frac{\partial f}{\partial Y_T}\right)^\mathsf{T} = \nabla_T f|_0$, which are computationally available as such. The equation (C.1) is obtained by using the formulations in Appendix B for $\mathbf{n}$ and computing $\nabla f$ with the ALE transformation 1.1. The rest is rearrangement.

# Appendix D

# Derivations related to the mass balance equations

This section is a collection of derivations and calculations related to formulating the surfactant mass balance equation in weak form (2.13), given below also as (D.1). The syntax here follow that of the main material and the other appendices.

$$\int_{\partial\Omega} \hat{\Gamma}\frac{\partial\Gamma}{\partial t}ds = \int_{\partial\Omega}\left[\hat{\Gamma}\left(N^{ads}-N^{des}\right)-\hat{\Gamma}\frac{\Gamma}{A}\frac{\partial A}{\partial t}-\hat{\Gamma}\left(\mathbf{v}_T\bullet\nabla_T\Gamma\right)-\nabla_T\hat{\Gamma}\bullet\left(D^s\nabla_T\Gamma\right)\right]ds \tag{D.1}$$

The weak form (D.1) is formulated upon the original form (D.2) as detailed below.

$$\frac{\partial\Gamma}{\partial t}=N^{ads}-N^{des}-\nabla_T\bullet\left(\Gamma\mathbf{v}_T\right)-\Gamma\left(\nabla_T\bullet\mathbf{n}\right)\left(\mathbf{v}\bullet\mathbf{n}\right)+D^s\nabla_T^2\Gamma \tag{D.2}$$

The surface derivative, or tangential differential operator ($\nabla_T$) is essential.

$$\nabla_T=\left(\mathbf{I}-\mathbf{n}\mathbf{n}^{\mathsf{T}}\right):\nabla, \text{ where} \tag{D.3}$$

$\mathbf{I}$ is the identity tensor and $\mathbf{n}$ the surface (outward) normal vector.

It is recognized that the notations for mean curvature are equal regardless of differential operator

$$\nabla_T\bullet\mathbf{n}=\nabla\bullet\mathbf{n} \tag{D.4}$$

and that the product of the surface curvature and surface normal velocity can be written as a sum of local area change and the tangential velocity divergence (D.5).

$$\left(\nabla\bullet\mathbf{n}\right)\left(\mathbf{n}\bullet\mathbf{v}\right)=\left(\mathbf{I}-\mathbf{n}\mathbf{n}^{\mathsf{T}}\right):\nabla\mathbf{v}-\nabla_T\bullet\mathbf{v}_T \tag{D.5}$$

Further, also the rule for expanding the divergence of a vector multiplied with a scalar is needed for the first right-hand side term in (D.2).

$$\nabla_T \bullet (\Gamma \mathbf{v}_T) = \nabla_T \Gamma \bullet \mathbf{v}_T + \Gamma \left(\nabla_T \bullet \mathbf{v}_T\right) \tag{D.6}$$

Also it is known that

$$\frac{1}{A}\frac{dA}{dt} = \left(\mathbf{I} - \mathbf{n}\mathbf{n}^{\mathsf{T}}\right) : \nabla \mathbf{v}. \tag{D.7}$$

Hence,

$$\begin{aligned}
\frac{\partial \Gamma}{\partial t} &= -\nabla_T \bullet (\Gamma \mathbf{v}_T) - \Gamma \left(\nabla_T \bullet \mathbf{n}\right)(\mathbf{v} \bullet \mathbf{n}) + D^s \nabla_T^2 \Gamma \\
&= -\nabla_T \bullet (\Gamma \mathbf{v}_T) - \Gamma \left(\mathbf{I} - \mathbf{n}\mathbf{n}^{\mathsf{T}}\right) : \nabla \mathbf{v} + \Gamma \left(\nabla_T \bullet \mathbf{v}_T\right) + D^s \nabla_T^2 \Gamma \\
&= -\mathbf{v}_T \bullet \nabla_T \Gamma - \Gamma \left(\nabla_T \bullet \mathbf{v}_T\right) - \Gamma \left(\mathbf{I} - \mathbf{n}\mathbf{n}^{\mathsf{T}}\right) : \nabla \mathbf{v} + \Gamma \left(\nabla_T \bullet \mathbf{v}_T\right) + D^s \nabla_T^2 \Gamma \\
&= -\mathbf{v}_T \bullet \nabla_T \Gamma - \Gamma \left(\mathbf{I} - \mathbf{n}\mathbf{n}^{\mathsf{T}}\right) : \nabla \mathbf{v} + D^s \nabla_T^2 \Gamma \\
&= -\mathbf{v}_T \bullet \nabla_T \Gamma - \Gamma \left(\frac{1}{A}\frac{dA}{dt}\right) + D^s \nabla_T^2 \Gamma.
\end{aligned} \tag{D.8}$$

Formulating (D.8) into the weak form is done as usual; multiply the equation with a test function, $\hat{\Gamma}$, and integrate over the domain in question (D.9). In fact, the only term that can be treated in any manner is the product containing the surface diffusion term.

$$\begin{aligned}
&\int_{\partial\Omega} \hat{\Gamma} \frac{\partial \Gamma}{\partial t} ds \\
&= \int_{\partial\Omega} \hat{\Gamma} \left(-\nabla_T \Gamma \bullet \mathbf{v}_T - \Gamma \frac{1}{A}\frac{dA}{dt} + D^s \nabla_T^2 \Gamma\right) ds \\
&= \int_{\partial\Omega} \hat{\Gamma} \left(-\nabla_T \Gamma \bullet \mathbf{v}_T - \Gamma \frac{1}{A}\frac{dA}{dt}\right) ds + D^s \int_{\partial\Omega} \hat{\Gamma} \nabla_T \bullet (\nabla_T \Gamma) ds \\
&= \int_{\partial\Omega} \hat{\Gamma} \left(-\nabla_T \Gamma \bullet \mathbf{v}_T - \Gamma \frac{1}{A}\frac{dA}{dt}\right) ds + D^s \int_{\partial\Omega} \hat{\Gamma} \nabla_T \bullet (\nabla_T \Gamma) ds \\
&= \int_{\partial\Omega} \hat{\Gamma} \left(-\nabla_T \Gamma \bullet \mathbf{v}_T - \Gamma \frac{1}{A}\frac{dA}{dt}\right) ds + D^s \int_{\partial\Omega} \nabla \bullet (\hat{\Gamma} \nabla_T \Gamma) - \nabla_T \hat{\Gamma} \bullet \nabla_T \Gamma ds \\
&= \int_{\partial\Omega} \hat{\Gamma} \left(-\nabla_T \Gamma \bullet \mathbf{v}_T - \Gamma \frac{1}{A}\frac{dA}{dt} - D^s \nabla_T \hat{\Gamma} \bullet \nabla_T \Gamma\right) ds + D^s \int_{\partial^2\Omega} \mathbf{n} \bullet (\hat{\Gamma} \nabla_T \Gamma) dp
\end{aligned} \tag{D.9}$$

Equation (D.9) yields the weak form for the mass balance on the surface, as well as a contribution to the boundary conditions *on this surface domain* $\partial\Omega$. In a two-dimensional model the surface to be modelled is one-dimensional (the dimension is along the surface) its boundaries, $\partial^2\Omega$ thus being point-like, $dp$ is a point element.

**Tangential divergence of scalar-vector product** Here $f$ is a scalar variable and $\mathbf{G}$ is a vector variable, both defined on a surface where $\nabla_T$ operates.

$$\begin{aligned}
\nabla_T \bullet (f\mathbf{G}) &= (\mathbf{I} - \mathbf{n}\mathbf{n}^\mathsf{T}) : \nabla \bullet f\mathbf{G} \\
&= (\mathbf{I} - \mathbf{n}\mathbf{n}^\mathsf{T}) : (\nabla f \bullet \mathbf{G} + f(\nabla \bullet \mathbf{G})) \\
&= (\mathbf{I} - \mathbf{n}\mathbf{n}^\mathsf{T}) : \nabla f \bullet \mathbf{G} + (\mathbf{I} - \mathbf{n}\mathbf{n}^\mathsf{T}) : (\nabla \bullet \mathbf{G}) f \\
&= \nabla_T f \bullet \mathbf{G} + (\nabla_T \bullet \mathbf{G}) f
\end{aligned}$$

**Curvature formulations in plane symmetric geometries** In a two-dimensional plane symmetric model the boundaries are lines in the Cartesian space and taking a surface differential on a line segment includes dividing the differential operation into a sum of differentials in the directions of the basis vectors. Assuming $x$ and $y$ as system coordinates, $\mathbf{n} = (n_x, n_y)$ and $n_x^2 + n_y^2 = 1$,

$$\nabla_T = (\mathbf{I} - \mathbf{n}\mathbf{n}^\mathsf{T}) : \nabla = \begin{pmatrix} 1 - n_x^2 & -n_x n_y \\ -n_x n_y & 1 - n_y^2 \end{pmatrix} : \nabla = \begin{pmatrix} n_y^2 \frac{\partial}{\partial x} - n_x n_y \frac{\partial}{\partial y} \\ n_x^2 \frac{\partial}{\partial y} - n_x n_y \frac{\partial}{\partial x} \end{pmatrix}.$$

Operating (dot-product) on the surface normal with $\nabla_T$ yields

$$\nabla_T \bullet \mathbf{n} = n_y^2 \frac{\partial n_x}{\partial x} - n_x n_y \left( \frac{\partial n_x}{\partial y} + \frac{\partial n_y}{\partial x} \right) + n_x^2 \frac{\partial n_y}{\partial y}.$$

Based on the normal vector norm, the normal vector component cross-differentials can be transformed as below.

$$\begin{aligned}
\frac{\partial n_x}{\partial y} &= -\frac{n_y}{n_x} \frac{\partial n_y}{\partial y}, \text{and} \\
\frac{\partial n_y}{\partial x} &= -\frac{n_x}{n_y} \frac{\partial n_x}{\partial x}
\end{aligned}$$

Hence,

$$\begin{aligned}
\nabla_T \bullet \mathbf{n} &= n_y^2 \frac{\partial n_x}{\partial x} + n_x n_y \left( \frac{n_y}{n_x} \frac{\partial n_y}{\partial y} + \frac{n_x}{n_y} \frac{\partial n_x}{\partial x} \right) + n_x^2 \frac{\partial n_y}{\partial y} \\
&= (n_y^2 + n_x^2) \frac{\partial n_x}{\partial x} + (n_y^2 + n_x^2) \frac{\partial n_y}{\partial y} \\
&= \frac{\partial n_x}{\partial x} + \frac{\partial n_y}{\partial y} \\
&= \nabla \bullet \mathbf{n}.
\end{aligned}$$

**Decomposition of $\mathbf{v}$ into $\mathbf{v}_T$ and $\mathbf{v}_n$** The movement velocity $\mathbf{v}$ of an interface or a surface can be decomposed into its tangential-to-surface ($\mathbf{v}_T$) and normal-to-surface ($\mathbf{v}_n$) components, just like any vector on the surface: $\mathbf{v} = \mathbf{v}_T + \mathbf{v}_n$. The

$\mathbf{v}_n$ component is obtained as $(\mathbf{n} \bullet \mathbf{v})\mathbf{n}$. This decomposition is applied in deriving the general surfactant mass balance equation (D.2). A clarifying computation is given here.

$$\begin{aligned}
\nabla_T \bullet (\Gamma \mathbf{v}) &= \nabla_T \bullet (\Gamma(\mathbf{v}_T + \mathbf{v}_n)) \\
&= \nabla_T \bullet (\Gamma \mathbf{v}_T + \Gamma(\mathbf{n} \bullet \mathbf{v})\mathbf{n})) \\
&= \nabla_T \bullet (\Gamma \mathbf{v}_T) + \nabla_T \bullet (\Gamma(\mathbf{n} \bullet \mathbf{v})\mathbf{n})) \\
&= \nabla_T \bullet (\Gamma \mathbf{v}_T) + \nabla_T (\Gamma(\mathbf{n} \bullet \mathbf{v})) \bullet \mathbf{n}) + \Gamma(\mathbf{n} \bullet \mathbf{v})(\nabla_T \bullet \mathbf{n}) \\
&= \nabla_T \bullet (\Gamma \mathbf{v}_T) + \Gamma(\mathbf{n} \bullet \mathbf{v})(\nabla_T \bullet \mathbf{n}) \\
&= \nabla_T \bullet (\Gamma \mathbf{v}_T) + \Gamma(\mathbf{n} \bullet \mathbf{v})(\nabla \bullet \mathbf{n}) \qquad \text{(D.10)}
\end{aligned}$$

In (D.10) the term $\nabla_T (\Gamma(\mathbf{n} \bullet \mathbf{v})) \bullet \mathbf{n}$ is zero by definition since the surface gradient is always along the surface and thus perpendicular to the surface normal, whereby their dot product equals zero.

**Simplification of $\mathbf{v}_T \bullet \nabla_T f$** By calculation, it can be shown that in a 2D model $\mathbf{v}_T \bullet \nabla_T f = \mathbf{v} \bullet \nabla_T f$ - that is, the velocity component at the boundary need not be tangential-projected, when included in a dot-product-operation with a tangential gradient variable. $f$ can be any scalar variable, for example $\Gamma$.

$$
\begin{aligned}
\mathbf{v}_T \bullet \nabla_T f &= \left((\mathbf{I} - \mathbf{n}\mathbf{n}^\mathsf{T}) : \mathbf{v}\right) \bullet \left(((\mathbf{I} - \mathbf{n}\mathbf{n}^\mathsf{T}) : \nabla) f\right) \\
&= \begin{pmatrix} 1-n_x^2 & -n_x n_y \\ -n_y n_x & 1-n_y^2 \end{pmatrix} \begin{pmatrix} v_x \\ v_y \end{pmatrix} \bullet \begin{pmatrix} 1-n_x^2 & -n_x n_y \\ -n_y n_x & 1-n_y^2 \end{pmatrix} \begin{pmatrix} \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \end{pmatrix} f \\
&= \begin{pmatrix} n_y^2 v_x - n_x n_y v_y \\ n_x^2 v_y - n_x n_y v_x \end{pmatrix} \bullet \begin{pmatrix} n_y^2 \frac{\partial f}{\partial x} - n_x n_y \frac{\partial f}{\partial y} \\ n_x^2 \frac{\partial f}{\partial y} - n_x n_y \frac{\partial f}{\partial x} \end{pmatrix} \\
&= \left(n_y^2 v_x - n_x n_y v_y\right) \left(n_y^2 \frac{\partial f}{\partial x} - n_x n_y \frac{\partial f}{\partial y}\right) \\
&+ \left(n_x^2 v_y - n_x n_y v_x\right) \left(n_x^2 \frac{\partial f}{\partial y} - n_x n_y \frac{\partial f}{\partial x}\right) \\
&= n_y^2 n_y^2 \frac{\partial f}{\partial x} v_x - n_x n_y n_y^2 \frac{\partial f}{\partial x} v_y - n_x n_y n_y^2 \frac{\partial f}{\partial y} v_x + n_x^2 n_y^2 \frac{\partial f}{\partial y} v_y \\
&+ n_x^2 n_x^2 \frac{\partial f}{\partial y} v_y - n_x n_y n_x^2 \frac{\partial f}{\partial y} v_x - n_x n_y n_x^2 \frac{\partial f}{\partial x} v_y + n_x^2 n_y^2 \frac{\partial f}{\partial x} v_x \\
&= \left(n_y^2 n_y^2 \frac{\partial f}{\partial x} - n_x n_y n_y^2 \frac{\partial f}{\partial y} - n_x n_y n_x^2 \frac{\partial f}{\partial y} + n_x^2 n_y^2 \frac{\partial f}{\partial x}\right) v_x \\
&+ \left(n_x^2 n_y^2 \frac{\partial f}{\partial y} - n_x n_y n_y^2 \frac{\partial f}{\partial x} - n_x n_y n_x^2 \frac{\partial f}{\partial x} + n_x^2 n_x^2 \frac{\partial f}{\partial y}\right) v_y \\
&= \left((n_x^2 + n_y^2) n_y^2 \frac{\partial f}{\partial x} - (n_x^2 + n_y^2) n_x n_y \frac{\partial f}{\partial y}\right) v_x \\
&+ \left((n_x^2 + n_y^2) n_x^2 \frac{\partial f}{\partial y} - (n_x^2 + n_y^2) n_x n_y \frac{\partial f}{\partial x}\right) v_y \\
&= \left(n_y^2 \frac{\partial f}{\partial x} - n_x n_y \frac{\partial f}{\partial y}\right) v_x + \left(n_x^2 \frac{\partial f}{\partial y} - n_x n_y \frac{\partial f}{\partial x}\right) v_y \\
&= \begin{pmatrix} v_x \\ v_y \end{pmatrix} \begin{pmatrix} n_y^2 \frac{\partial f}{\partial x} - n_x n_y \frac{\partial f}{\partial y} \\ n_x^2 \frac{\partial f}{\partial y} - n_x n_y \frac{\partial f}{\partial x} \end{pmatrix} \\
&= \mathbf{v} \bullet \nabla_T f
\end{aligned}
\tag{D.11}
$$

**Equation (D.5) in 2D** The computation below explains the content of surface curvature ($\nabla \bullet \mathbf{n}$) multiplied by the surface normal velocity ($\mathbf{v} \bullet \mathbf{n}$), as in (D.5).

The main procedure is carried out in a two-dimensional case, not considering the physical content of the vectors or whether the underlying physical phenomenon is plane-symmetric or axially symmetric. The treatment is more or less merely a series vector computations, independent of the vectors' physical meaning. This is exactly why the procedure is sufficient for both axially and plane-symmetric cases, as long as the vector components included are defined correctly in the first place. In practice the $x$ and $y$-dimensions are used to denote the horizontal and vertical coordinates, respectively, in the plane-symmetric case with Cartesian coordinates, and, in the axially symmetric case with cylindrical coordinates, the radial and the vertical coordinates, respectively. With these selections the following treatment covers both the plane-symmetric as well as the axially symmetric modelling cases.

Handling the $\mathbf{n}$ and $\mathbf{v}$ vectors term-wise as $\mathbf{n} = (n_x, n_y)$ and $\mathbf{v} = (v_x, v_y)$ yields the following computations.

$$
\begin{aligned}
&(\mathbf{I}-\mathbf{nn}):\nabla\mathbf{v}-\nabla_T\bullet\mathbf{v}_T\\
&= \begin{pmatrix} 1-n_x^2 & -n_xn_y \\ -n_yn_x & 1-n_y^2 \end{pmatrix} : \begin{pmatrix} \frac{\partial v_x}{\partial x} & \frac{\partial v_y}{\partial x} \\ \frac{\partial v_x}{\partial y} & \frac{\partial v_y}{\partial y} \end{pmatrix} - (\mathbf{I}-\mathbf{nn}):\nabla:(\mathbf{I}-\mathbf{nn}):\mathbf{v}\\
&= ((1-n_x^2)\frac{\partial v_x}{\partial x} - n_xn_y\frac{\partial v_y}{\partial x} - n_yn_x\frac{\partial v_x}{\partial y} + (1-n_y^2)\frac{\partial v_y}{\partial y})\\
&- \begin{pmatrix} (1-n_x^2)\frac{\partial}{\partial x} - n_xn_y\frac{\partial}{\partial y} \\ -n_yn_x\frac{\partial}{\partial x} + (1-n_y^2)\frac{\partial}{\partial y} \end{pmatrix} \bullet \begin{pmatrix} (1-n_x^2)v_x - n_xn_yv_y \\ -n_yn_xv_x + (1-n_y^2)v_y \end{pmatrix}\\
&= n_y^2\frac{\partial v_x}{\partial x} - n_xn_y\frac{\partial v_y}{\partial x} - n_yn_x\frac{\partial v_x}{\partial y} + n_x^2\frac{\partial v_y}{\partial y}\\
&- (n_y^2\frac{\partial}{\partial x} - n_xn_y\frac{\partial}{\partial y})(n_y^2v_x - n_xn_yv_y) - (n_x^2\frac{\partial}{\partial y} - n_yn_x\frac{\partial}{\partial x})(n_x^2v_y - n_yn_xv_x)\\
&= n_y^2\frac{\partial v_x}{\partial x} - n_xn_y\frac{\partial v_y}{\partial x} - n_yn_x\frac{\partial v_x}{\partial y} + n_x^2\frac{\partial v_y}{\partial y}\\
&- n_y^2\left(\frac{\partial(n_y^2v_x)}{\partial x} - \frac{\partial(n_xn_yv_y)}{\partial x}\right) + n_xn_y\left(\frac{\partial(n_y^2v_x)}{\partial y} - \frac{\partial(n_xn_yv_y)}{\partial y}\right)\\
&- n_x^2\left(\frac{\partial(n_x^2v_y)}{\partial y} - \frac{\partial(n_xn_yv_x)}{\partial y}\right) + n_xn_y\left(\frac{\partial(n_x^2v_y)}{\partial x} - \frac{\partial(n_xn_yv_x)}{\partial x}\right)\\
&= n_y^2\frac{\partial v_x}{\partial x} - n_xn_y\frac{\partial v_y}{\partial x} - n_yn_x\frac{\partial v_x}{\partial y} + n_x^2\frac{\partial v_y}{\partial y}\\
&- n_y^2\left(n_y^2\frac{\partial v_x}{\partial x} - 2v_xn_x\frac{\partial n_x}{\partial x}\right) + n_y^2\left(n_xn_y\frac{\partial v_y}{\partial x} + v_y\left(n_y - \frac{n_x^2}{n_y}\right)\frac{\partial n_x}{\partial x}\right)\\
&+ n_xn_y\left(n_y^2\frac{\partial v_x}{\partial y} + 2v_xn_y\frac{\partial n_y}{\partial y}\right) - n_xn_y\left(n_xn_y\frac{\partial v_y}{\partial y} + v_y\left(n_x - \frac{n_y^2}{n_x}\right)\frac{\partial n_y}{\partial y}\right)\\
&- n_x^2\left(n_x^2\frac{\partial v_y}{\partial y} - 2v_yn_y\frac{\partial n_y}{\partial y}\right) + n_x^2\left(n_xn_y\frac{\partial v_x}{\partial y} + v_x\left(n_x - \frac{n_y^2}{n_x}\right)\frac{\partial n_y}{\partial y}\right)\\
&+ n_xn_y\left(n_x^2\frac{\partial v_y}{\partial x} + 2v_yn_x\frac{\partial n_x}{\partial x}\right) - n_xn_y\left(n_xn_y\frac{\partial v_x}{\partial x} + v_x\left(n_y - \frac{n_x^2}{n_y}\right)\frac{\partial n_x}{\partial x}\right)\\
&= (n_y^2 - n_y^4 - n_x^2n_y^2)\frac{\partial v_x}{\partial x} + (n_x^3n_y + n_xn_y^3 - n_xn_y)\left(\frac{\partial v_y}{\partial x} + \frac{\partial v_x}{\partial y}\right) + (n_x^2 - n_x^4 - n_x^2n_y^2)\frac{\partial v_y}{\partial y}\\
&+ \left(2v_xn_xn_y^2 + n_y^2v_y\left(n_y - \frac{n_x^2}{n_y}\right) + 2v_yn_yn_x^2 - n_xn_yv_x\left(n_y - \frac{n_x^2}{n_y}\right)\right)\frac{\partial n_x}{\partial x}\\
&+ \left(2v_xn_xn_y^2 - n_xn_yv_y\left(n_x - \frac{n_y^2}{n_x}\right) + 2v_yn_yn_x^2 + n_x^2v_x\left(n_x - \frac{n_y^2}{n_x}\right)\right)\frac{\partial n_y}{\partial y}
\end{aligned}
\tag{D.12}
$$

Next note the following:

$$\begin{aligned}
(n_y^2 - n_y^4 - n_x^2 n_y^2) = (n_x^3 n_y + n_x n_y^3 - n_x n_y) = (n_x^2 - n_x^4 - n_x^2 n_y^2) &= 0 \\
\left(2v_x n_x n_y^2 + n_y^2 v_y \left(n_y - \frac{n_x^2}{n_y}\right) + 2v_y n_y n_x^2 + n_x n_y v_x \left(n_y - \frac{n_x^2}{n_y}\right)\right) &= (n_x v_x + n_y v_y) \\
\left(2v_x n_x n_y^2 - n_x n_y v_y \left(n_x - \frac{n_y^2}{n_x}\right) + 2v_y n_y n_x^2 + n_x^2 v_x \left(n_x - \frac{n_y^2}{n_x}\right)\right) &= (n_x v_x + n_y v_y)
\end{aligned}$$

Now that,

$$(n_x v_x + n_y v_y)\frac{\partial n_x}{\partial x} + (n_x v_x + n_y v_y)\frac{\partial n_y}{\partial y} = (\mathbf{n} \bullet \mathbf{v})(\nabla \bullet \mathbf{n}) = (\nabla \bullet \mathbf{n})(\mathbf{n} \bullet \mathbf{v})$$

is

$$(\mathbf{I} - \mathbf{n}\mathbf{n}^{\mathsf{T}}) \bullet \nabla \mathbf{v} - \nabla_T \bullet \mathbf{v}_T = (\nabla \bullet \mathbf{n})(\mathbf{n} \bullet \mathbf{v}).$$

**Equality of 2.6a and 2.6b** The equality is given only for the case of incompressible fluids and the idea for concluding that $-\left(\mathbf{n}^{\mathsf{T}} \bullet \nabla \mathbf{v} \bullet \mathbf{n}\right)$ equals $\left(\mathbf{I} - \mathbf{n}\mathbf{n}^{\mathsf{T}}\right) : \nabla \mathbf{v}$ is in noticing that for incompressible fluids $\nabla \bullet \mathbf{v} = 0$.

$$\begin{aligned}
-(\mathbf{n} \bullet \nabla \mathbf{v} \bullet \mathbf{n}) &= -\begin{pmatrix} n_x & n_y \end{pmatrix} \bullet \begin{pmatrix} \frac{\partial v_x}{\partial x} & \frac{\partial v_y}{\partial x} \\ \frac{\partial v_x}{\partial y} & \frac{\partial v_y}{\partial y} \end{pmatrix} \bullet \begin{pmatrix} n_x & n_y \end{pmatrix} \\
&= -(n_x^2 \frac{\partial v_x}{\partial x} + n_x n_y \frac{\partial v_y}{\partial x} + n_y n_x \frac{\partial v_x}{\partial y} + n_y^2 \frac{\partial v_y}{\partial y}) \\
&= (n_y^2 - 1)\frac{\partial v_x}{\partial x} - n_x n_y (\frac{\partial v_y}{\partial x} + \frac{\partial v_x}{\partial y}) + (n_x^2 - 1)\frac{\partial v_y}{\partial y} \\
&= n_y^2 \frac{\partial v_x}{\partial x} - n_x n_y (\frac{\partial v_y}{\partial x} + \frac{\partial v_x}{\partial y}) + n_x^2 \frac{\partial v_y}{\partial y} - (\frac{\partial v_x}{\partial x} + \frac{\partial v_y}{\partial y}) \\
&= n_y^2 \frac{\partial v_x}{\partial x} - n_x n_y (\frac{\partial v_y}{\partial x} + \frac{\partial v_x}{\partial y}) + n_x^2 \frac{\partial v_y}{\partial y} \\
&= \left(\mathbf{I} - \mathbf{n}\mathbf{n}^{\mathsf{T}}\right) : \nabla \mathbf{v} \qquad \text{(D.13)}
\end{aligned}$$