

**Report**  
**Simulation Sciences Laboratory**

# **Minimal Surfaces**

Chenfei Fan  
Praveen Mishra  
Sankarasubramanian Ragunathan  
Philipp Schleich  
January 1, 2020

Supervisor: Prof. Dr. Uwe Naumann  
Klaus Leppkes

## **Abstract**

Abstract might be unnecessary

# Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
<b>2</b>	<b>Background</b>	<b>3</b>
2.1	General background (don't like this title) . . . . .	3
2.2	Numerical solution . . . . .	4
<b>3</b>	<b>Implementation</b>	<b>5</b>
<b>4</b>	<b>Results, benchmark</b>	<b>5</b>
<b>5</b>	<b>Conclusions and outlook</b>	<b>5</b>

# 1 Introduction

...

## 2 Background

### 2.1 General background (don't like this title)

We look at surfaces in  $\mathbb{R}^3$ , defined over an open set  $\Omega \subset \mathbb{R}^2$ . The surface of desire should contain the least possible area among all possible surfaces, that assume given values on the boundary of  $\Omega$ , denoted by  $\partial\Omega$ . ?

Lagrange showed in 1760, that such a surface is characterized by the graphic of a function  $z(x, y)$ ,  $z : \mathbb{R}^2 \rightarrow \mathbb{R}$ , which is twice continuously differentiable on a twodimensional domain, particularly in a subset of  $\mathbb{R}^2$ . This function  $z$  has to fulfill the so called *Minimal-Surface Equation* (MSE) stated below.

$$\begin{aligned} (1 + z_y^2)z_{xx} - 2z_x z_y z_{xy} + (1 + z_x^2)z_{yy} &= \mathcal{F}[z] = 0 && \text{in } \Omega \\ z(x, y) &= g(x, y) && \text{on } \partial\Omega \end{aligned} \quad (1)$$

Clearly, this formulation satisfies the prescribed boundary values given by  $g(x, y)$  due to the Dirichlet boundary condition on  $\partial\Omega$ . As to why the graphic of functions solving this equation describes a minimal surface, we refer to the literature. For example ? gives a very straightforward proof. [if there's time, we might write this up here]

In the following, we will call the differential operator  $\mathcal{F}[\cdot]$  the Minimal-Surface Operator (MSO). The resulting partial differential equation (PDE) in eq. (1) turns out to be an elliptic PDE of second order, which is in particular *non-linear*. The solution of such a PDE is not trivial, and typically requires numerical treatment. For certain cases, analytical descriptions are available, such as for Scherk's surface<sup>1</sup>.

[Here, introduce Scherks surface]

Later on, we will use Scherk's surface as a test-case to verify our numerical results.

---

<sup>1</sup>Here, we only look at Scherk's first surface

## 2.2 Numerical solution

In this project, we are supposed to solve the MSE numerically on  $\Omega \equiv (0, 1) \times (0, 1)$ . In the following, discretized quantities are indicated by a superscript  $h$ . The spatial domain is to be discretized using a structured mesh with equidistant grid spacing both in  $x, y$ , i.e. we have the same number of grid points in both directions,  $N = N_x = N_y$ . Thus, we define  $\Omega^h := \{(x, y) \in \mathbb{R}^2 : (x, y) = (ih, jh), 0 \leq i, j < N, hN = 1\}$ .

[Picture of Omega-h]

We choose to discretize the MSO on  $\Omega^h$  by Finite Differences, since this is usually the easiest way to go, and on a structured grid, would anyways yield similar discrete equation as in Finite Volume or Finite Element methods.

To obtain a second order consistent discrete MSO ( $F^h[\cdot]$ ), we use central difference stencils on the first, second and mixed derivative. Since all these stencils have make only use of immediate neighbours, there is no need to treat nodes close to the boundary differently, since the boundary is given by  $g(\cdot)$ .

This way, we obtain a discrete version of (1):

$$(1 + d_x[z^h]^2)d_{yy}[z^h] - d_x[z^h]d_y[z^h]d_{xy}[z^h] + (1 + d_y[z^h]^2)d_{xx}[z^h] = F^h[z^h] = 0 \quad \text{in } \Omega^h$$

$$z^h = g \quad \text{on } \partial\Omega^h$$
(2)

– introduce stencils and maybe provide pictures, consistency

If totally bored, check consistency of the method

Note, that the discrete equation is not anymore consistent with the right-hand-side of (1), but yields  $F^h[z^h] = r^h \neq 0$ . This is due to the non-linear nature of the PDE, and the solution to (2) thus boils down to an iterative root-finding problem, i.e. in iteration  $k$ , we get  $F^h[z_k^h] = r_k^h$ , and we want  $r_k^h \rightarrow 0$  for increasing, yet still reasonably small  $k$ . This can be achieved by the standard Newton-Raphson procedure as follows:

---

**Algorithm 1** Newton's method applied on the discrete MSE

---

```

k ← 0
z_k^h ← 0
r_k^h ← F^h[z_k^h]
while ||r_k^h|| > TOL do
    z_{k+1}^h ← z_k^h - (F^h[z_k^h])^{-1} r_k^h
    k ← k + 1
    r_k^h ← F^h[z_k^h]
end while

```

▷ Or a more sophisticated initial guess

▷ We choose  $\|\cdot\| = \|\cdot\|_2$

---

### **3 Implementation**

How we implemented things, ...

- a nice flowchart (can be maybe done in powerpoint, easier here) on how the code works, plus explanations

- test cases -¿ this might aswell be mentioned in results (the larger testcases)

### **4 Results, benchmark**

A few nice pictures for stuff

- timing benchmarks for different approaches (openMP and not), compare to matlab maybe

### **5 Conclusions and outlook**

Some wise words, on what how can we extend stuff, ...