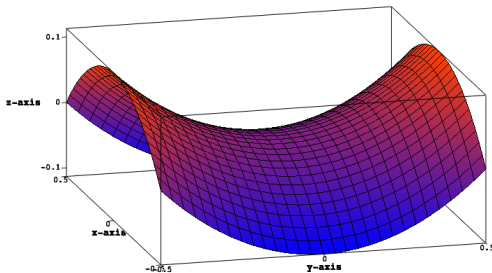


Minimalflächenprojekt

Chun-Kan Chow, Daniel Partida, Stefan Jeske, Tom Witter

November 4, 2015



Einführung

- Letter of Requirements
- Einführung in Minimalflächen

Überlegung und Vorentwurf

- Letter of Requirements
- Vorabtreffen mit Jens Deussen
- Unser Ui-Sketch

Analyse und Entwurf

- Funktionale und Nichtfunktionale Anforderungen
- Top Level Use Cases
- Aktivitätsdiagramm: Run
- Begriffsmodell
- Klassendiagramm
- Sequenzdiagramm: Run
- Arbeitsaufteilung

Implementierung

- Anmerkungen
- Live-Demo

Letter of Requirements

Aachen, 22. April 2015

Auftrag zur Entwicklung einer Simulationssoftware

Sehr geehrte Damen und Herren,

aufgrund der Notwendigkeit zur Reduzierung von Materialverbrauch, aber auch aus dem Wunsch heraus, ästhetisch ansprechende Produkte zu produzieren, benötigt unser Unternehmen eine Simulationssoftware zur (instationären) Berechnung von Minimalflächen.

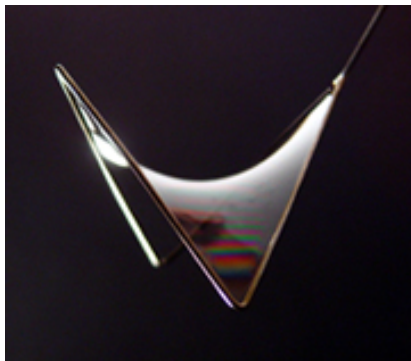
Ausschlaggebend für eine gute Integration der Software in unseren Arbeitsprozess ist, dass mithilfe einer graphischen Benutzerschnittstelle Randbedingungen und numerische Parameter eingestellt, sowie erstellte Konfigurationen als Datei gespeichert und auch wieder eingelesen werden können. Außerdem soll eine Visualisierung der resultierenden Minimalfläche während der Simulation oder im Nachhinein möglich sein.

Wir freuen uns auf eine Zusammenarbeit.

Mit freundlichen Grüßen

(Jens Deussen und Uwe Naumann)

Einführung in Minimalflächen



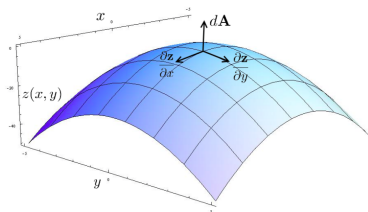
- ▶ Definition [Wikipedia.de]: Eine Minimalfläche ist eine Fläche im Raum, die lokal minimalen Flächeninhalt hat. Derartige Formen nehmen beispielsweise Seifenhäute an, wenn sie über einen entsprechenden Rahmen (wie etwa einem Blasring) gespannt sind.
- ▶ Beispiel: Münchener Olympiastadion

Mathematischer Hintergrund

Vektorisierte Darstellung der Fläche:

$$\vec{z}(x, y) = \begin{pmatrix} x \\ y \\ z(x, y) \end{pmatrix}$$

Darstellung des differentiellen Flächenelements:



$$dA = \left\| \frac{\partial \vec{z}}{\partial x} \times \frac{\partial \vec{z}}{\partial y} \right\| = \sqrt{1 + z_x^2 + z_y^2}$$

Mathematischer Hintergrund

- ▶ Lösen des Funktionals

$$\delta A = 0$$

- ▶ Lösung:

$$(1 + z_x^2)z_{yy} - 2z_x z_y z_{xy} + (1 + z_y^2)z_{xx} = 0$$

- ▶ → Diskretisierung

Diskretisierung

Schrittweite in x Richtung

$$h_x = \frac{1}{n}$$

Stützstellen in x Richtung

$$x_i = ih_x \quad i = 0, \dots, n$$

Schrittweite in y Richtung

$$h_y = \frac{1}{m}$$

Stützstellen in y Richtung

$$y_j = jh_y \quad j = 0, \dots, m$$

→ finite Differenzen in Residuumsmatrix $R \in \mathbb{R}^{(m+1) \times (n+1)}$

→ Lösung ist Nullstelle von R (Newton Verfahren)

Newton Verfahren

- ▶ Lösung nichtlinearer Gleichungssysteme
- ▶ Benötigt Jacobimatrix $J(\hat{R}(z_n)) \in \mathbb{R}^{(m+1)(n+1) \times (m+1)(n+1)}$
- ▶ Berechnet Schrittweite Δz_n durch Gleichungssystem:

$$J(\hat{R}(z_n))\Delta z_n = -\hat{R}(z_n)$$

$$z_{n+1} = z_n + \Delta z_n$$

- ▶ Iterativ

Letter of Requirements

Aachen, 22. April 2015

Auftrag zur Entwicklung einer Simulationssoftware

Sehr geehrte Damen und Herren,

aufgrund der Notwendigkeit zur Reduzierung von Materialverbrauch, aber auch aus dem Wunsch heraus, ästhetisch ansprechende Produkte zu produzieren, benötigt unser Unternehmen eine Simulationssoftware zur (instationären) Berechnung von Minimalflächen.

Ausschlaggebend für eine gute Integration der Software in unseren Arbeitsprozess ist, dass mithilfe einer graphischen Benutzerschnittstelle Randbedingungen und numerische Parameter eingestellt, sowie erstellte Konfigurationen als Datei gespeichert und auch wieder eingelesen werden können. Außerdem soll eine Visualisierung der resultierenden Minimalfläche während der Simulation oder im Nachhinein möglich sein.

Wir freuen uns auf eine Zusammenarbeit.

Mit freundlichen Grüßen

(Jens Deussen und Uwe Naumann)

Vorabtreffen mit Jens Deussen


Boundaries

a $\sin(x)$

b $\exp(y)$

c

d



Numerische Parameter

- Diskretisierung h_x, h_y
- Fehlergrenze ϵ
- max. Iterationen

- Makefile
- Laden der Konfig.
- Speichern der Konfiguration
- Speichern des Ergebnisses (Daten + Plot) \rightarrow pdf
- Daten einlesen zum Plotten
- Berechnung
- Plot (newpoint variabel)
- Reset/Beenden
- optional:
 - instationär
 - Fortschrittsbalken (Fehler, Anz. Iteration)
 - parallel

Unser Ui-Sketch

Hauptfenster

Speichern

Laden

Numerische Parameter

Randfunktionen

a

c

b

d

Gebietsgrösse

a

b

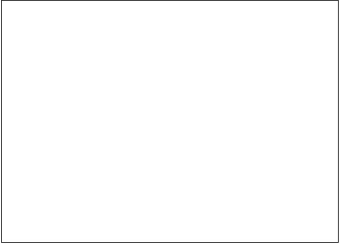
c

d

y

x

Plot



Run

Funktionale Anforderungen

- ▶ Speichern der Einstellungen
- ▶ Laden der Einstellungen
- ▶ Speichern der Ergebnisse
- ▶ Laden der Ergebnisse
- ▶ Möglicher Wechsel zwischen drei Tabs.
Funktionen des ersten Tabs (Einstellungen) :
 1. Eingabe der Anzahl der Stützstellen in X- und Y- Richtung
 2. Eingabe der Randfunktionen
 3. Eingabe des Abbruchfehlers
 4. Eingabe der Anzahl der maximalen Iterationen
 5. Änderung des Gebietes
 6. Run Button
 7. Quit Button

Funktionale Anforderungen

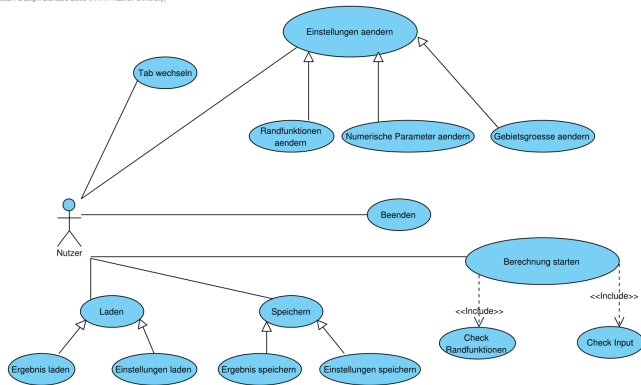
- ▶ Funktionen des zweiten Tabs (Konsolenausgabe)
 1. Anzeigen des Fortschrittes (Berechnung)
 2. Benachrichtigung, falls das Newtonverfahren nicht konvergiert
 3. Anzeige des aktuellen Iterationschrittes
 4. Anzeige des aktuellen Fehlers
- ▶ Funktionen des dritten Tabs (Anzeige)
 1. Graphische Darstellung der berechneten Minimalflächen
 2. Plot: variabler viewpoint

Nicht funktionale Anforderungen

- ▶ Anzeigen einer Benutzeroberfläche
- ▶ Eingabe der Zahlen durch Tastatur
- ▶ Skalierung des Hauptfensters bei unterschiedlichen Auflösungen
- ▶ Erzeugung von korrekten Ergebnissen
- ▶ Intuitive Bedienbarkeit
- ▶ Plattformunabhängig
- ▶ Effiziente Berechnung

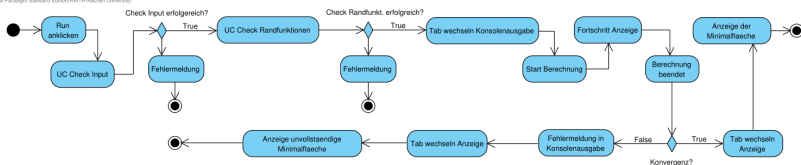
Top Level Use Cases

Visual Paradigm Standard Edition (RWTH Aachen University)



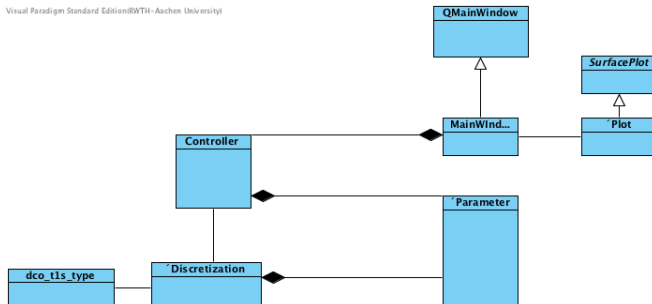
Aktivitätsdiagramm: Run

Visual Paradigm Standard Edition (RWTH Aachen University)

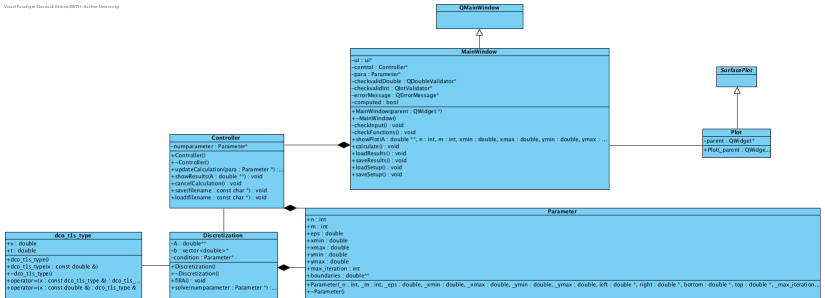


Begriffsmodell

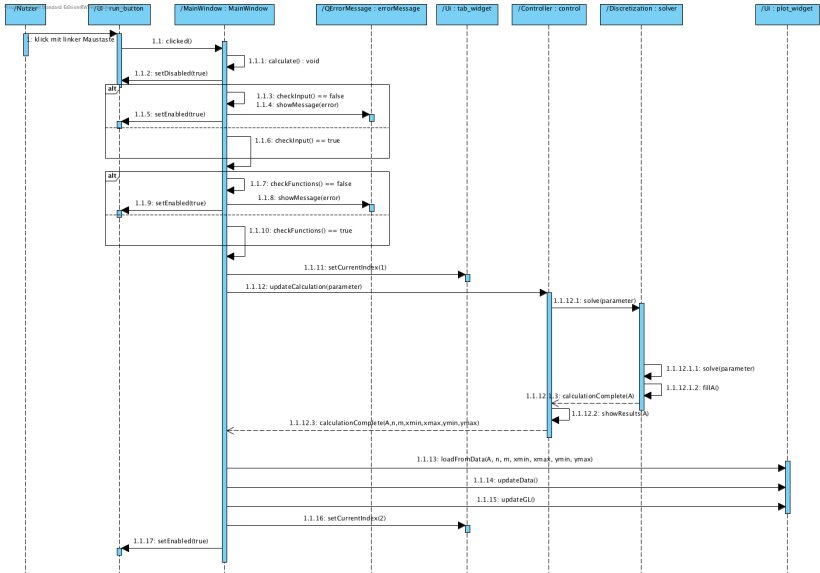
Visual Paradigm Standard Edition(RWTH-Aachen University)



Visual Perception Standard Edition (RHT)-Rachon, University of



Sequenzdiagramm: Run



Arbeitsaufteilung

- ▶ Brainstorming → Top-Level Use Cases, Begriffsmodell
- ▶ Implementierung:
 - ▶ Tom: Diskretisierung, User Interface
 - ▶ Daniel: Fehlermeldungen
 - ▶ Chun: Controller/Mainwindow (Laden/Speichern), User Interface
 - ▶ Stefan: Controller, User Interface, Bibliotheken, Debugging, Diskretisierung
- ▶ Analyse&Entwurf:
 - ▶ Tom: Benutzeranforderungen, Sequenzdiagramme
 - ▶ Daniel: Aktivitätsdiagramme, Use Case Beschreibungen
 - ▶ Chun: Benutzerdokumentation, Sequenzdiagramme
 - ▶ Stefan: Entwicklerdokumentation, Benutzerdokumentation

Anmerkungen zur Implementierung

- ▶ DCO
- ▶ Template Funktionen
- ▶ EIGEN
- ▶ 2 Threads
- ▶ Fortschrittsbalken
- ▶ Ausgabe Berechnungsfortschritt
- ▶ Cancel Button
- ▶ Weitere Optimierung
 - ▶ Startvektor
 - ▶ Sparse-Matrix
 - ▶ Zwischenspeicher
 - ▶ Verzweigungen

Live-Demo

Ende

Vielen Dank für Ihre Aufmerksamkeit
Fragen?