

Code-Generierung für einen neuen 3D-Druckprozess auf Tropfenbasis

Johannes Schwaiger

Vollständiger Abdruck der von der Fakultät für Maschinenwesen der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktor-Ingenieurs (Dr.-Ing.)

genehmigten Dissertation.

Vorsitzender: Univ.-Prof. Dr.-Ing. Michael Friedrich Zäh

Prüfer der Dissertation:

1. Univ.-Prof. Dr. rer. nat. Tim Christian Lüth
2. Univ.-Prof. Dr. rer. nat. habil. Hans-Joachim Bungartz

Die Dissertation wurde am 17.12.2013 bei der Technischen Universität München eingereicht und durch die Fakultät für Maschinenwesen am 08.09.2014 angenommen.

Danksagung

Diese Arbeit entstand während meiner mehrjährigen Tätigkeit als wissenschaftlicher Mitarbeiter am Lehrstuhl für Mikrotechnik und Medizingerätetechnik (MiMed) der TU München.

Hiermit möchte ich mich in erster Linie bei meinem Doktorvater Prof. Dr. rer. nat. Tim Christian Lüth herzlich für die Möglichkeit bedanken diese Arbeit unter seiner Betreuung am MiMed anfertigen zu dürfen. Mein ausdrücklicher Dank gilt dabei vor allem auch für die hervorragende Ausbildung, die ich im Vorfeld dieser Arbeit durch die Vorlesung Automatisierungstechnik in der Medizin genießen durfte und innerhalb einer mehr als zweijährigen Projektphase im Bereich der Medizintechnik praktisch vertiefen konnte. Ohne diese außerordentlichen Grundlagen wäre eine schnelle und erfolgreiche Bearbeitung des Themas dieser Arbeit nicht möglich gewesen.

Bedanken möchte ich mich auch vielmals bei Prof. Dr. rer. nat. Hans-Joachim Bungartz für die schnelle Übernahme der Rolle des Zweitgutachters, genauso wie bei Prof. Dr.-Ing. Michael Friedrich Zäh für die ebenso unkomplizierte Übernahme des Prüfungsvorsitzes. Prof. Bungartz kenne ich noch aus meiner Zeit als Diplomand am Lehrstuhl für Scientific Computing und als Student aus verschiedenen Lehrveranstaltungen in der Informatik.

Mein ganz besonderer Dank gilt auch Dr. Franz Irlinger als Leiter der Gruppe Mikrotechnik für die zahlreichen fachlichen Diskussionen und wertvollen Ratschläge als auch die moralische Betreuung in schwierigeren Phasen. Dankend hervorheben möchte ich auch meinen Kollegen Dr. Nikita Shevchenko, der mich als einziger wissenschaftlicher Mitarbeiter die gesamte Zeit am Lehrstuhl begleitet hat. Fast im selben Atemzug sind auch Markus Kagerer und Jelena Prša zu erwähnen. Beiden möchte ich unter Anderem auch dafür danken, dass sie mich auf wissenschaftlichen Konferenzen so professionell vertreten haben und mehr als Kollegen für mich waren. Zuletzt möchte ich namentlich noch Eva Graf erwähnen, die mich bei meinen letzten Experimenten im Rahmen dieser Arbeit so sehr unterstützt hat. Danke auch an alle weiteren Mitarbeiter für die fachlichen und vor allem nicht fachlichen Diskussionen.

Meinen Studenten Da Che, Marc Bosch Faulí, David Müller und Johannes Krabusch möchte ich ein besonderes Lob für die hervorragende geleistete Arbeit aussprechen.

Vielen lieben Dank an mein gesamtes privates Umfeld und meine Freunde. Speziell hervorheben möchte ich all diejenigen, mit denen ich immer über alles reden konnte und die mir teilweise im Laufe der letzten Jahre schwer ans Herz gewachsen sind. Für euch wahrscheinlich unbewusst, habt ihr so einen immensen Beitrag zum Gelingen meiner Arbeit geleistet. Nicht nur die Unterstützung in schwierigeren Zeiten, sondern auch die Ablenkung vom beruflichen Alltag ist euch sehr gut gelungen. In diesem Sinne: vielen Dank für die zahlreichen schönen Tage und Abende. Ihr hättet mich ruhig noch viel öfter ablenken dürfen.

Abschließend möchte ich mich bei meinen Eltern herzlich bedanken. Sie waren zu jeder Zeit ein starker Rückhalt, ermöglichten mir ein sorgenfreies Studium und somit die Grundlage für diese Arbeit. Auch in arbeitsreicheren Phasen waren sie es, die mir stets bewusst machten, dass es bei aller Anstrengung noch wichtigere Sachen im Leben gibt und somit bei mir für die nötige Lockerheit sorgten, die für eine solche Arbeit genauso wichtig ist.

Johannes Schwaiger, September 2014

Zusammenfassung

Es wird die automatische Code-Generierung für einen neuen 3D-Druckprozess beschrieben, der es ermöglicht durch sequentielles Austragen von Tropfen aus Kunststoff ein 3-dimensionales Bauteil Schicht für Schicht aufzubauen. Ziel der Arbeit ist der Einsatz der gedruckten Teile als Funktionsteile. Wesentlicher Bestandteil dieser Arbeit war dabei die Entwicklung einer prozessspezifischen Software, welche in der Lage ist aus CAD-Daten im STL-Format, automatisch Maschinencode (G-Code) zu generieren, der von der Maschinensteuerung ausgeführt werden kann um ein Bauteil zu drucken.

Abstract

The automatic code generation is described for a new 3D printing process based on droplets that facilitates building a 3-dimensional part layer by layer by sequential discharge of plastic polymer droplets. The aim of this work is the usage of the printed parts as functional parts. An essential component of this work was the development of a process-specific software that is able to automatically generate machine code (G-Code) out of CAD-data in the STL-format and that can be executed by the machine control in order to print a part.

Inhaltsverzeichnis

1	Anwendung: Herstellung von Kunststoffteilen	1
1.1	Beispiele für Anwendungen	1
1.2	Problemstellung	4
2	Stand der Technik	5
2.1	Spritzgießen	5
2.1.1	Phasen des Spritzgießens	5
2.1.2	Schneckenformen	7
2.1.3	Spritzgießverfahren	8
2.1.4	Spritzgießwerkzeuge	8
2.2	Extrusion	8
2.3	Weitere bewährte Verfahren	9
2.4	Fused Deposition Modeling	9
2.5	Vergleich zwischen Spritzgießen und FDM	11
2.6	Systeme auf dem Markt	12
2.6.1	Spritzgießmaschinen	12
2.6.2	FDM-Maschinen	13
2.7	Stand der Forschung	13
2.8	Kritik am Stand der Technik	13
2.8.1	Echte und lösbare Defizite	14
2.8.2	Gegenwärtig nicht mit vertretbarem Aufwand lösbare Defizite	15
3	Datenaufbereitung für einen Tropfenerzeuger	16
3.1	Eigener Ansatz	16
3.2	Abgrenzung	18
3.3	Liste (möglicherweise) lösbarer Defizite	19
3.4	Experimente zur Evaluierung von gedruckten Bauteilen	20
3.4.1	Vergleich mit Spritzgießverfahren	20
3.4.2	Vergleich mit FDM-Verfahren	20
4	Konzept und Systementwurf	22
4.1	Konzept der Hardware	22
4.2	Konzept des Druckprozesses	23
4.3	Nutzung von STL-Dateien	24
4.3.1	Das STL-Format	24
4.3.2	Notation von Vertex List und Facet List	25
4.4	Prozessbeschreibung und Struktur der Software	26
4.5	Prozessspezifischer Maschinencode	28
5	Realisierung der Software-Module	31
5.1	Mathematische Definitionen	31

5.2	Mathematische Grundlagen	39
5.2.1	Berechnung der Transformationsmatrizen	46
5.2.2	Abstandsberechnungen	47
5.2.3	Schnittpunktberechnungen	49
5.2.4	Berechnung von Zusammenhangskomponenten	59
5.2.5	Flächeninhalt von geschlossenen Kurven ohne Überschneidungen	59
5.2.6	Windungszahlberechnung	65
5.3	Algorithmik der Software-Module	71
5.3.1	Importieren der STL-Daten	71
5.3.2	Slicen der STL-Daten	73
5.3.3	Berechnung von Offset-Konturen	78
5.3.4	Füllung der Schichten	89
5.3.5	Optimierung der Daten für den Tropfenerzeuger	94
5.3.6	Generierung von prozessspezifischem Code	95
5.4	Implementierung der Software-Module	96
5.4.1	Eingliederung in den Gesamtprozess	96
5.4.2	Programmierung der Module	98
6	Experimente	105
6.1	Drucken individueller Funktionsbauteile	105
6.2	Validierung der Oberflächenrauheit	107
6.3	Validierung der Zugfestigkeit	116
7	Zusammenfassung und Ausblick	128
	Literaturverzeichnis	130

1 Anwendung: Herstellung von Kunststoffteilen

Die Eigenschaften von Werkstoffen sind ein zentraler Aspekt bei der Herstellung von Gebrauchsgegenständen. So empfehlen sich polymere Werkstoffe beispielsweise insbesondere als Materialien, welche sich durch eine leichte Verformbarkeit auszeichnen. Erhard (2008) beschreibt den Fortschritt bei der Entwicklung polymerer Werkstoffe. So entdeckte Hermann Staudinger 1926 den strukturellen Aufbau der Polymere. Seit den 1950er Jahren konnte somit ein rasanter Anstieg in der Produktion von Polymerwerkstoffen verzeichnet werden, welcher lediglich ab 1973 kurzzeitig aufgrund des 1. Golfkrieges stagnierte. Während die Stahlproduktion seither nicht mehr signifikant gesteigert wurde, überschritt die Menge an produzierten „Kunststoffen“ 1988 erstmalig die Menge an produziertem Stahl (gemessen in m^3) und ist seither der Stahlproduktion überlegen (siehe Abbildung 1.1).

1.1 Beispiele für Anwendungen

Die Kunststoffherzeuger tragen mit ca. 30 % zum industriellen Produktionswert in der Bundesrepublik Deutschland bei. Verarbeitungsbetriebe leisten einen Beitrag von ca. 63 %, die letzten 7 % entfallen auf die Hersteller von Verarbeitungsmaschinen. Bereiche wie Luft- und Raumfahrt, Feinwerktechnik, Fahrzeugbau, Apparatebau, aber auch der allgemeine Maschinenbau und das Bauwesen gehören zu den Branchen, in denen die Verwendung und der Einsatz von Kunststoffteilen immer mehr an Bedeutung gewinnen.

Ein bekanntes Beispiel aus dem Leichtbau bzw. der Luft- und Raumfahrt ist die Anwendung von Kohlefaserverbundwerkstoffen bei Flugzeugen vom Typ Airbus. Hier wurde das Seitenleitwerk des Airbus A310 bereits 1985 aus kohlenstofffaserverstärktem Kunststoff (CFK) hergestellt, was nach Erhard (2008) zu einer erheblichen Gewichtsreduzierung des gesamten Leitwerks von ca. 25 % im Vergleich zur Aluminiumvariante führte. Mittlerweile werden bei Flugzeugen der Airbus Familie neben Seiten- und Höhenleitwerken auch die Landeklappen an den Flügeln aus CFK hergestellt. Beim A340 macht dies sieben Tonnen an CFK aus.

In der Feinwerktechnik konnten im Laufe der Jahre die Kosten durch Einsparung von Fertigungsschritten und Montageschritten reduziert werden. Hierbei werden aus Thermoplasten multifunktionale Bauteile hergestellt, welche in einem Arbeitsgang produziert werden. Ein Beispiel für solche Baugruppen ist die gestanzte Platine eines Laufwerks in Abbildung 1.2. Dabei sind die Bauelemente selbst, wie Schnapp- und Verbindungshaken, Lagerstellen, Federelemente, Distanzbolzen usw. aus dem Polymerwerkstoff Polyoxymethylen (POM).

Im Fahrzeugbau kommt es ebenfalls sehr häufig zu Anwendungen für Polymerwerkstoffe. Etwas mehr als die Hälfte des Verbrauchs an Kunststoff fällt dabei auf die Innenausstattung an, erst danach folgen mit einem Viertel die Außenanwendungen. Abbildung 1.3 zeigt nach Erhard (2008) Nockenwellenantriebsräder für schrägverzahnten Stirnradantrieb bzw. Zahnriemenantrieb. Sie stammen aus den 60er Jahren und kommen so nicht mehr zum Einsatz, sind aber ein gutes Beispiel für die anfängliche Verarbeitung von Polymerwerkstoffen. Durch die

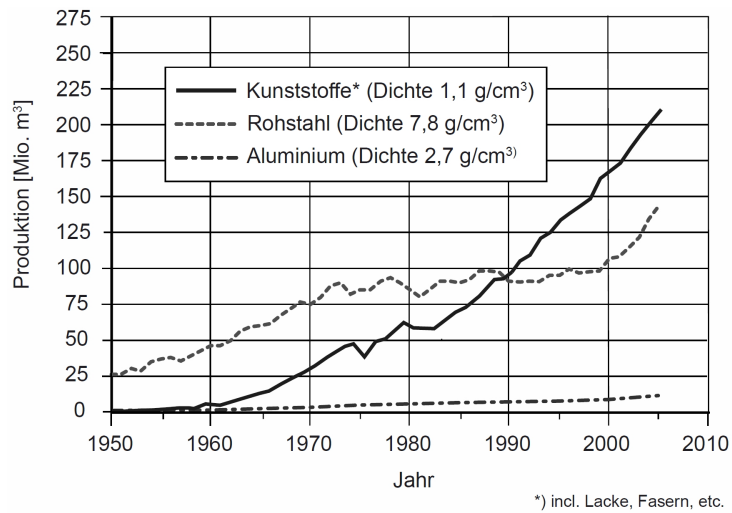


Abbildung 1.1: Produktion von Kunststoff, Rohstahl und Aluminium seit 1950. Entnommen aus Erhard (2008).

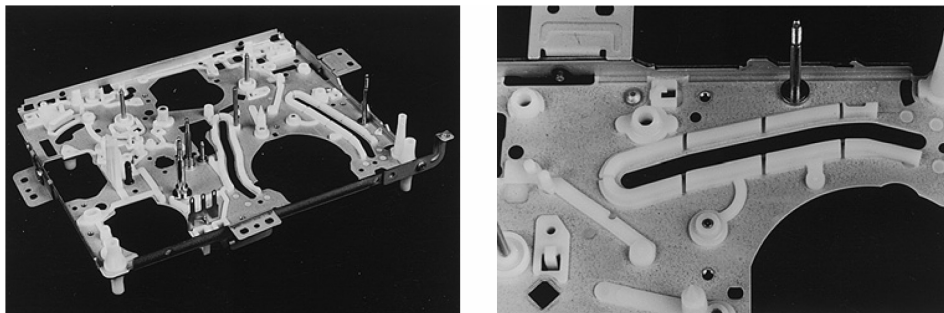


Abbildung 1.2: Gestanzte Platine eines Laufwerks mit Bauteilen aus Polyoxymethylen (POM). Entnommen aus Erhard (2008).

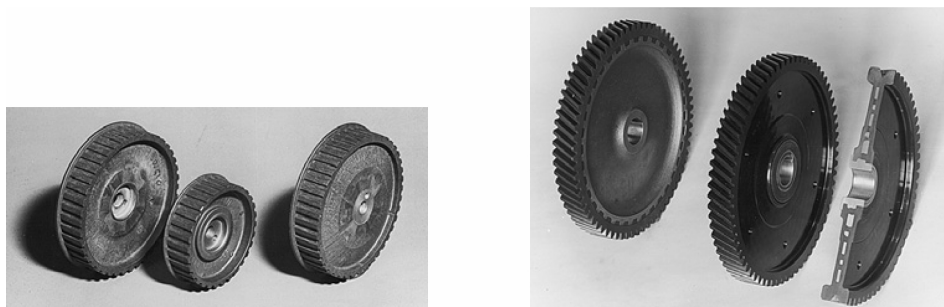


Abbildung 1.3: Verschiedene Ausführungen von Nockenwellenantriebsrädern aus Polymerwerkstoff. Entnommen aus Erhard (2008).

Verwendung bzw. Verarbeitung von Kunststoffen konnten im Bereich des Fahrzeugbaus auf diese Weise im Laufe der Jahre sowohl die Kosten als auch das Gewicht deutlich reduziert werden. Weitere Beispiele aus dem Fahrzeugbau wären Ansauganlagen (siehe Abbildung 1.4) oder auch Kupplungspedale (siehe Abbildung 1.5). Durch Bindenähte kann es zu einer optischen und mechanischen Beeinträchtigung von Bauteilen kommen. Abbildung 1.6 zeigt ein Beispiel für eine günstige Lage von Bindenähten. Die Zahnräder bestehen aus einem Zahnkranz, einem Lager in der Mitte und Federelementen dazwischen. Durch die Verdickungen (1) an den Federelementen (2) werden Bindenähte abgefangen. Zusätzlich befinden sich Verdickungen (3) am

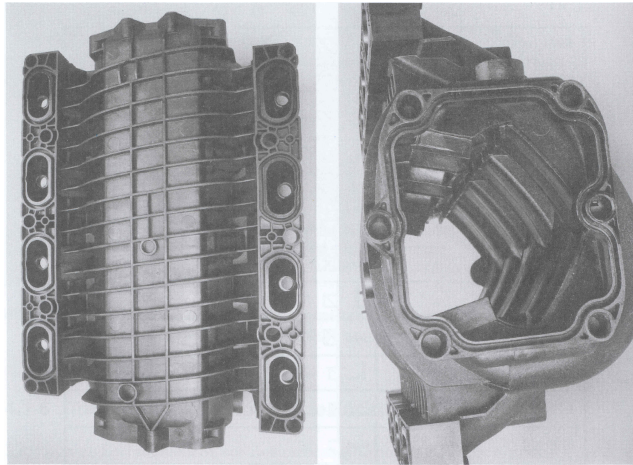


Abbildung 1.4: Ansauganlage mit runden Kanälen. Entnommen aus Ehrenstein (2007).

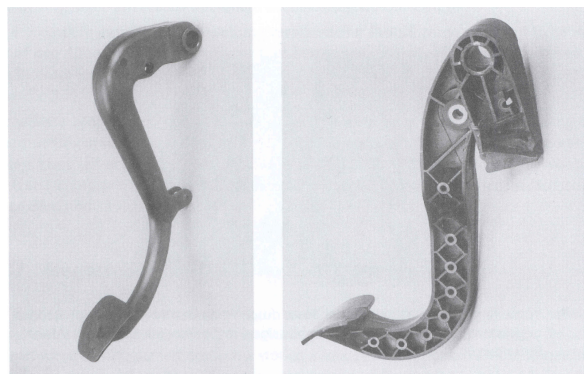


Abbildung 1.5: Kupplungspedal mit geschlossenem Hohlprofil (links) (BMW) und verripptem Doppel-T-Profilquerschnitt aus Polyamid 66 (rechts) (Mercedes-Benz / BASF). Entnommen aus Ehrenstein (2007).

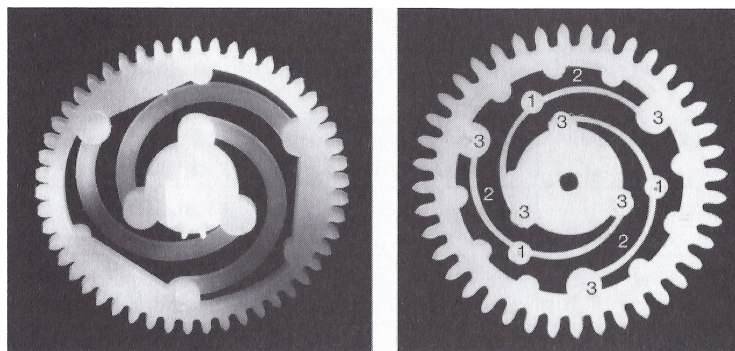


Abbildung 1.6: Weich gelagerte Zahnräder aus Polyoxymethylen (POM), Punktanguss (3) am Lager und am Zahnkranz (Beiter, Dauchingen). Entnommen aus Ehrenstein (2007).

Zahnkranz bzw. am Lager. Insgesamt wird es dadurch möglich die Lagerpartie translatorisch zu verschieben, aber auch direkt zu verdrehen.

1.2 Problemstellung

Insbesondere die Herstellung von Kunststoffteilen bietet ein breites Anwendungsspektrum in der Industrie. Dabei sind die Anforderungen an das verarbeitende Verfahren sehr hoch und steigen immer mehr. Vor allem durch den steigenden Lieferdruck von Firmen, welche Gesamtsysteme verkaufen, die Kunststoffteile beinhalten, ergeben sich folgende Forderungen:

- Automatisierung der Prozessabläufe zur Herstellung von Kunststoffteilen um körperlich schwere Arbeit zu vermeiden und Personal einzusparen,
- Verlagerung des Schwerpunkts von der Fertigung / Produktion zur Planung,
- Lieferung Just-In-Time bei gleichzeitiger Nullfehlertoleranz,
- Einsparungen von Werkstoffmenge, -gewicht und Raum.

Entscheidende Faktoren für den Erfolg eines Verfahrens sind dabei:

- die Produktivität,
- Entwicklungs- und Durchlaufzeiten,
- die Ausschussquote,
- die Kapitalkosten.

Immer mehr sind die verarbeitenden Verfahren dabei werkseigenen Normen unterworfen, welche den Einsatz umweltverträglicher Produkte fordern. Dies betrifft vor allem:

- die Minimierung des Materialeinsatzes,
- die Langlebigkeit und Wiederverwertbarkeit,
- die Lebenszyklus-Kostenbetrachtung.

Neben einer fristgerechten Herstellung von Kunststoffteilen auf Basis einer weitestgehend automatischen Produktion entscheidet aber vor allem auch die Qualität der hergestellten Bauteile über Erfolg und Misserfolg. So hängt die Qualität von Formteilen aus Kunststoff von verschiedenen Fertigungseinflüssen ab, welche bei der Planung berücksichtigt werden müssen:

- Maß- und Formgenauigkeit aufgrund von, unter anderem, Schwindungen bedingt durch beispielsweise thermische oder druckabhängige Einflüsse,
- fertigungsbedingte Einflüsse, z.B. durch Bindenähte,
- Wärmespannungen, z.B. durch nicht isothermes Abkühlen aus der Schmelze oder Einwirken unterschiedlicher Temperaturen aus der Schmelze.

Allgemein müssen die Endprodukte maßgenaue, optisch ansprechende Kunststoffteile mit einer geringen Oberflächenrauheit sein, welche sich durch eine hohe Festigkeit und (Form-)Stabilität auszeichnen. Neben der Fertigung in hohen Losgrößen besteht aber auch zunehmend immer mehr der Bedarf an Bauteilen in kleinen Losgrößen. Dabei handelt es sich in der Regel um nahezu uneingeschränkt individualisierbare Prototypen mit dem Wunsch diese auch als Funktionsbauteile einzusetzen.

2 Stand der Technik

Der Stand der Technik zur Herstellung von Kunststoffteilen umfasst mehrere Verfahren zur Verarbeitung von Kunststoff. Die Wichtigsten werden im Folgenden erläutert.

2.1 Spritzgießen

Spritzgießen ist heutzutage nach Johannaber und Michaeli (2001) das bedeutendste Verfahren zur Verarbeitung von Kunststoffen und somit zur Herstellung von Kunststoffteilen (gemessen an der Höhe der jährlichen Investitionen). Das Grundprinzip des Verfahrens ist es, plastifizierten Kunststoff unter hohem Druck über einen Angusskanal in eine Form einzuspritzen. Diese Form wird gewöhnlich auch als Spritzgießwerkzeug bezeichnet. Der Hohlraum (die Kavität) des Spritzgießwerkzeugs entspricht dabei der Form des fertigen Teils. Nach Abkühlung und Verfestigung des Kunststoffs in der Form, öffnet sich diese und das fertige Bauteil wird ausgeworfen. Eine Spritzgießmaschine besteht dabei aus den folgenden Komponenten: Spritzeinheit, Schließeinheit, Hydraulik / andere Antriebskonzepte und Steuerung. Abbildung 2.1 zeigt schematisch den Aufbau einer Spritzgießmaschine. Anhand einer solchen Spritzgießmaschine mit hydraulischem Antrieb werden im Folgenden die Abläufe beim Spritzgießen von Kunststoffteilen erläutert.

2.1.1 Phasen des Spritzgießens

Die drei wesentlichen Phasen des Spritzgießens sind das Einspritzen, das Plastifizieren (Dosieren) und das Auswerfen (Entformen) (siehe Johannaber und Michaeli (2001)). Dies ist in Abbildung 2.2 skizziert. Zunächst gelangt das Material über einen Trichter in die Spritzeinheit. Meist handelt es sich bei dem Material um Kunststoff im granulatformigen Zustand. Zum Füllen des Trichters mit Material werden in der Regel Fördereinrichtungen genutzt, welche automatisch das Kunststoffgranulat nachfüllen. Durch sogenannte Trichtertrockner kann dem Material unnötige Feuchtigkeit entzogen werden. Sobald sich das Granulat in der Aufbereitungseinheit befindet, wird es durch die Rotation der Schnecke immer in Richtung des Zylinderkopfs vorangetrieben. Die Schnecke wird dabei vom hydraulischen Motor angetrieben.

Plastifizieren (Dosieren)

Eine typische Dreizonen-Schnecke besteht aus drei verschiedenen Zonen. Zunächst wird das Granulat durch die Einzugszone befördert. Die Schneckengangtiefe ist hier konstant gleich groß. Dadurch wird das Granulat zusammengepresst, sich somit unnötiger Luft entledigt und das Material weiterbefördert. In der zweiten Zone, der Kompressionszone, nimmt die Schneckengangtiefe in Richtung des Zylinderkopfes ab. Wie der Begriff Kompression bereits verrät, wird hier das Volumen des Kunststoffgranulats reduziert und im Umkehrschluss die Dichte vergrößert. Dadurch reiben die einzelnen Kunststoffmoleküle immer stärker aneinander, was schließlich zur Plastifizierung (Aufschmelzung) des Materials führt. Die Kompressionszone

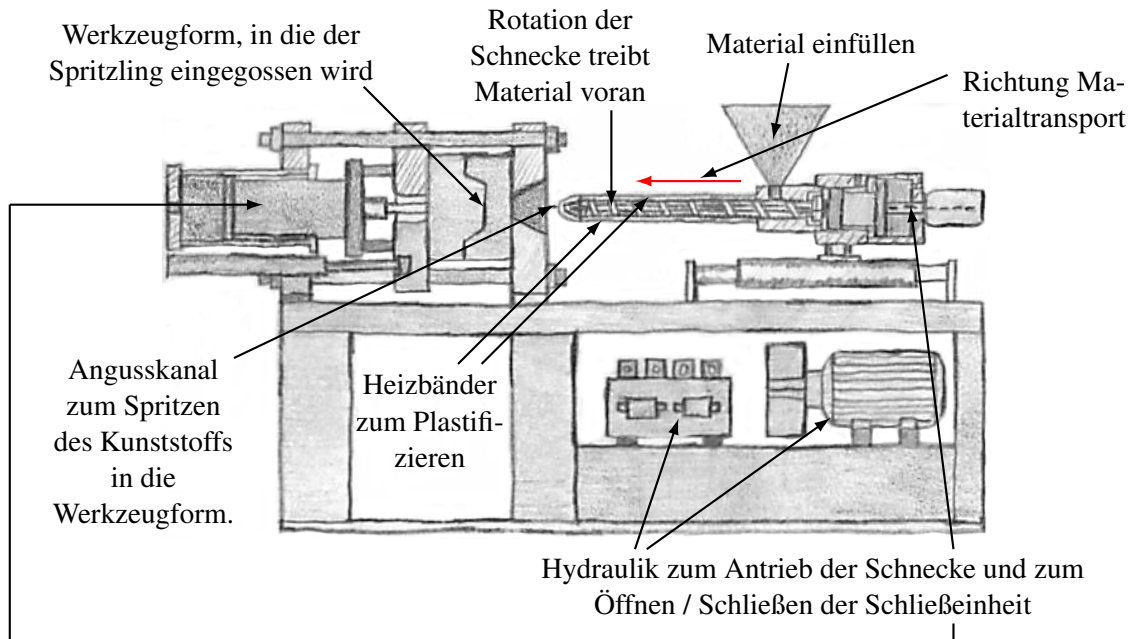


Abbildung 2.1: Spritzgießmaschine mit hydromechanischer Schließeinheit. Entnommen aus Johannaber und Michaeli (2001).

führt schließlich zur dritten Zone, der Meteringzone oder auch Austragszone genannt. Die Schneckengangtiefe ist hier wieder konstant gleich groß, allerdings viel kleiner als in der Einzugszone. Durch drei externe Heizbänder am Zylinder wird das Material zusätzlich erhitzt. Dadurch wird die Schmelze in der Meteringzone auf die gewünschte Verarbeitungstemperatur gebracht und verhindert, dass Wärme nach außen hin verloren geht. An der Spitze der Schnecke befindet sich eine Rückstromsperre, sozusagen also ein Verschlussventil, welches den Materialfluss nur in Richtung der Düse zulässt. Der Druck des angesammelten plastifizierten Materials drückt die Schnecke somit immer weiter zurück. Dadurch bildet sich vorderhalb der Schnecke ein Materialspeicher.

Einspritzen

Sobald genug Material zum Einspritzen im Materialspeicher vorhanden ist, hört die Schnecke auf zu rotieren und zieht sich zurück um das Material wieder zu dekomprimieren. Um das Material nun in die Form einzuspritzen, wird beim Vorfahren der Schnecke durch den Motor das Ventil geschlossen. Meist geschieht dies durch einen stoßartigen Druckaufbau im Hydraulikzylinder. Das geschmolzene Material wird so über die Düse der Einspritzeinheit in den Angusskanal und schließlich in das geschlossene Spritzgießwerkzeug gespritzt. Sobald der Materialspeicher leer ist, herrscht kein Druck mehr auf der Schnecke. Die Schnecke kann somit wieder beginnen zu rotieren. Es startet somit ein neuer Spritzgieß- / Dosierzyklus.

Beim Einspritzen werden nach Johannaber und Michaeli (2001) Drücke bis zu 3000 bar erzeugt. Abhängig von den Eigenschaften (Schmelztemperatur, Viskosität, ...) des verwendeten Kunststoffes erreicht das geschmolzene Material Temperaturen zwischen 160° und 320° Celcius.

Auswerfen (Entformen)

Im Spritzgießwerkzeug selbst erkaltet das eingespritzte Material unter abfallendem Druck. Auf diese Art entsteht ein sogenannter Spritzling. Sobald der Prozess des Einspritzens abgeschlos-

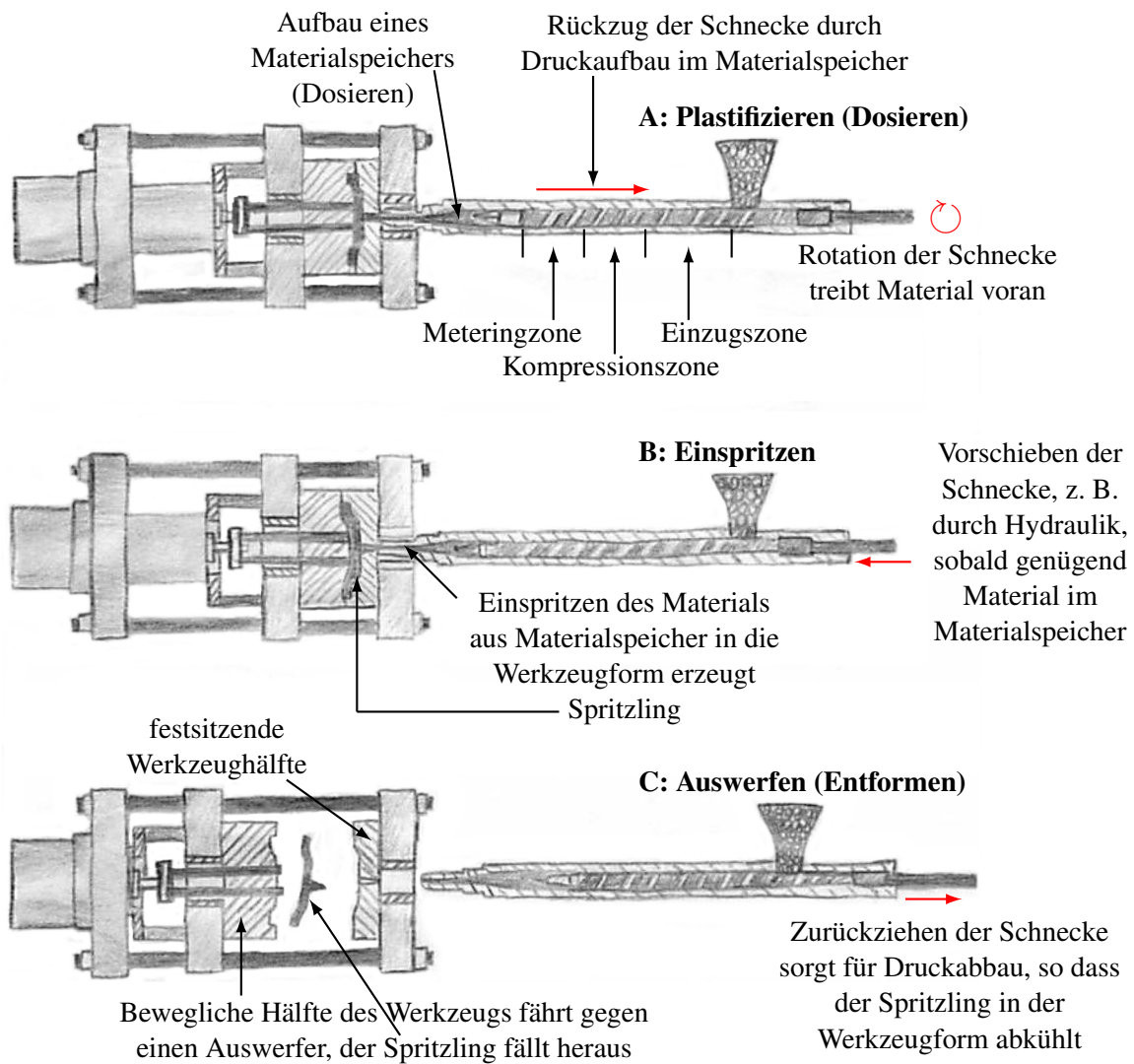


Abbildung 2.2: A: Plastifizieren (Dosieren), B: Einspritzen, C: Auswerfen (Entformen)

sen ist und die Formasse ausreichend abgekühlt ist, wird der Spritzling, welcher aus einem oder mehreren Formteilen und dem Anguss bestehen kann, aus der Form entfernt. Hierbei öffnet das Spritzgießwerkzeug, wobei das Formteil aufgrund der Auslegung des Werkzeugs meist an der beweglichen Seite der Werkzeughälfte haften bleibt. Beim Zurückfahren dieser beweglichen Hälfte, stößt das Werkzeug dann gegen einen feststehenden Auswerfer, so dass das fertige Teil ausgestoßen wird. Alternativ kann der Spritzling auch durch einen hydraulischen Auswerfer aus der Form entfernt werden. Eine detailliertere Skizze eines Spritzgießwerkzeugs ist in Abbildung 2.3 zu sehen.

2.1.2 Schneckenformen

Die genaue Form der verwendeten Schnecke ist abhängig vom Kunststoff. Nach Johannaber und Michaeli (2001) gibt es bei Förderschnecken unter anderem normale Dreizonen-Schnecken, Standardschnecken für die Thermoplastverarbeitung, Sonderschnecken für Thermoplaste, ... Bei den Schneckenantrieben selbst wird unterschieden zwischen rotatorischen und translatorischen Schneckenantrieben. Die Schnecke wird dabei von einem Hydromotor oder einem Elek-

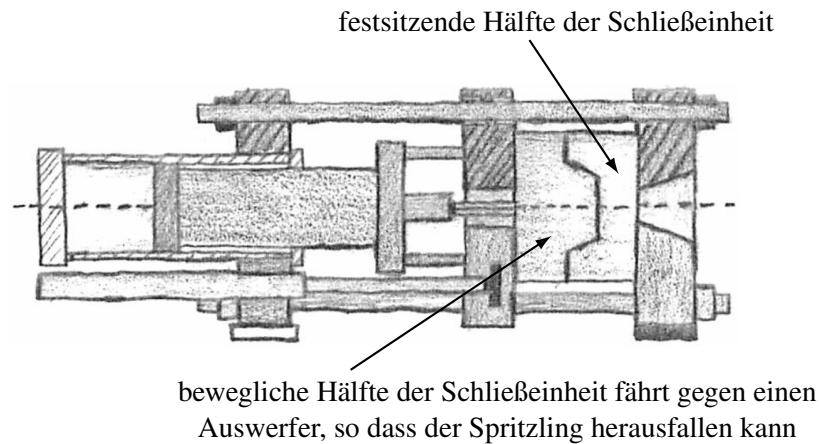


Abbildung 2.3: Beispiel einer Dreiplatten-Schließeinheit mit direkthydraulischer Verriegelung.

tromotor angetrieben.

2.1.3 Spritzgießverfahren

Es gibt eine Vielzahl an verschiedenen Spritzgießverfahren, welche u. A. auch abhängig sind vom eingesetzten Kunststoff. Für eine umfassende Übersicht sei verwiesen auf Johannaber und Michaeli (2001).

2.1.4 Spritzgießwerkzeuge

Bei Spritzgießwerkzeugen kommen verschiedene Varianten von Dreiplatten-Schließeinheiten, Zweiplatten-Schließeinheiten und säulenlosen Schließeinheiten zum Einsatz. Einen umfassenden Überblick geben Johannaber und Michaeli (2001). Bei der Werkzeugauslegung muss vorab auf rheologische und gebrauchstaugliche Aspekte Acht gegeben werden. Darum werden zunächst Prototypen der Werkzeuge angefertigt um damit entsprechende Analysen und Simulationen durchzuführen. Erst danach erfolgt die Fertigung eines maßgenauen Serienprodukts. Dabei kommt es nach Ehrenstein (2007) neben den Maßen auch auf Faktoren wie Fertigungszeiten, Wirtschaftlichkeit, Stückzahlen und Weiter- bzw. Wiederverwertung an. Eine schematische Darstellung der Abläufe beim Herstellen eines Spritzgießwerkzeugs gibt Abbildung 2.4.

2.2 Extrusion

Ähnlich wie beim Spritzgießen herrschen auch beim Extrudieren hohe Temperaturen und hoher Druck. Das Material wird ebenfalls durch eine Schnecke vorangetrieben und im Zylinder plastifiziert. Im Gegensatz zum Spritzgießen wird der plastifizierte Kunststoff jedoch nicht in eine Werkzeugform gespritzt, sondern durch eine formgebende Öffnung (auch Matrix genannt) gepresst, wo das Material geformt und abgekühlt wird beim Austritt aus der Matrix. Mögliche Formen sind dabei z.B. Stangen, Rohre, andere Profile wie beispielsweise bestimmte Winkel oder Querschnitte. Allgemein spricht man aufgrund des konstant gleichbleibenden Querschnitts auch von Rohr- und Profilextrusion. Das Profil wird dabei in einen länglichen mit Wasser gefüllten Kühltank oder -trogl extrudiert. Oft spricht man dabei auch von einem Kalibriertisch,

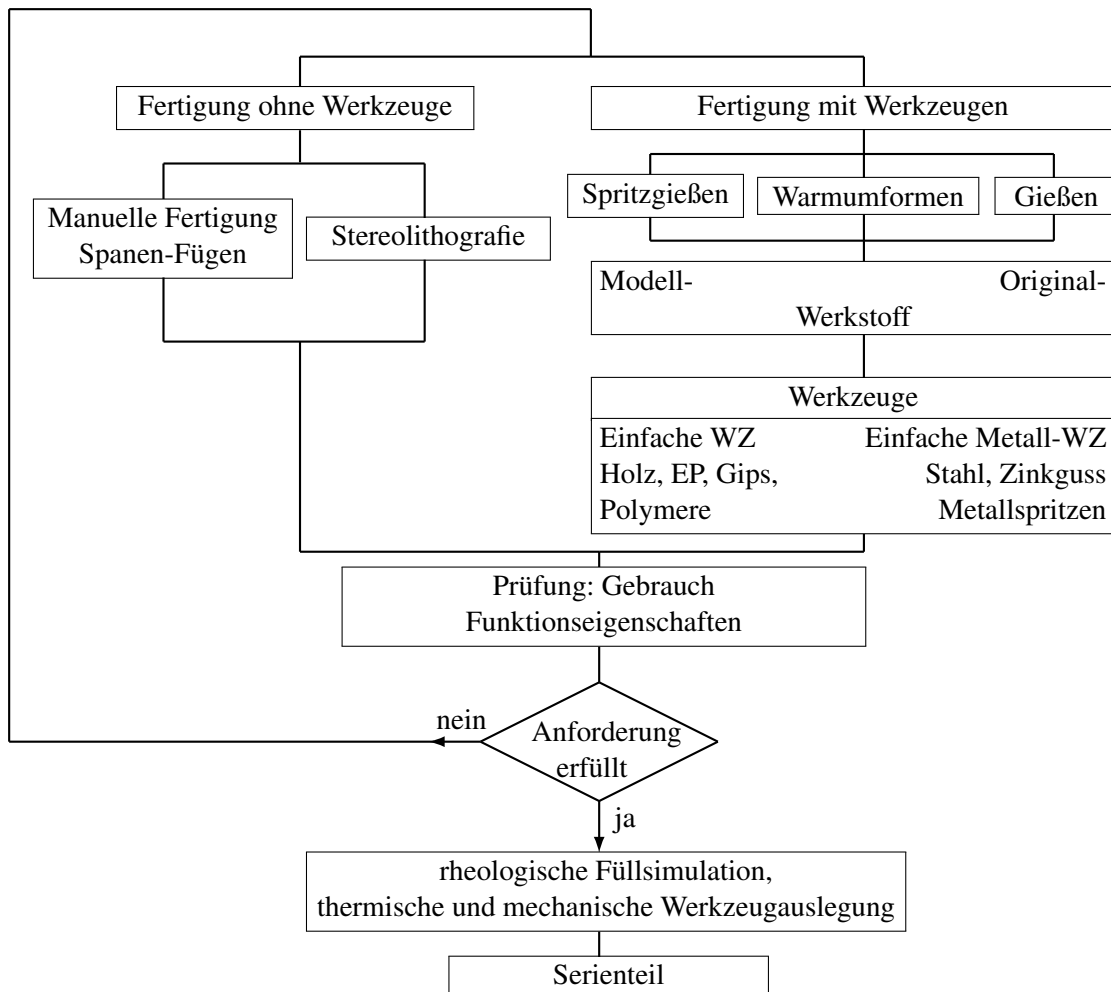


Abbildung 2.4: Abläufe beim Herstellen eines Spritzgießwerkzeugs, angelehnt an Ehrenstein (2007).

dessen Form stark vom extrudierten Kunststoff abhängig ist. Das Wasser kühlt den extrudierten Kunststoff beim Austritt aus der Matrix. Nach dem Kalibriertisch befindet sich ein Abzug, der für den gleichmäßigen Weitertransport des Rohrs bzw. Profils sorgt. Danach werden die Rohre bzw. Profile automatisch nach einer definierten Länge abgeschnitten.

2.3 Weitere bewährte Verfahren

Nach Johannaber und Michaeli (2001) wurden im Jahre 1996 über 80% der gesamten verarbeiteten Kunststoffmenge durch Spritzgießmaschinen und Extruder verarbeitet. Auf eine detaillierte Erklärung weiterer Verfahren, wie beispielsweise Blasformen, Pressen oder Schäumen wird deshalb verzichtet.

2.4 Fused Deposition Modeling

Das Fused Deposition Modeling (FDM) gehört zu den Rapid Prototyping Verfahren. Nach Crump (1992) besteht die Vorrichtung für das Verfahren aus einem beweglichen Dosierkopf,

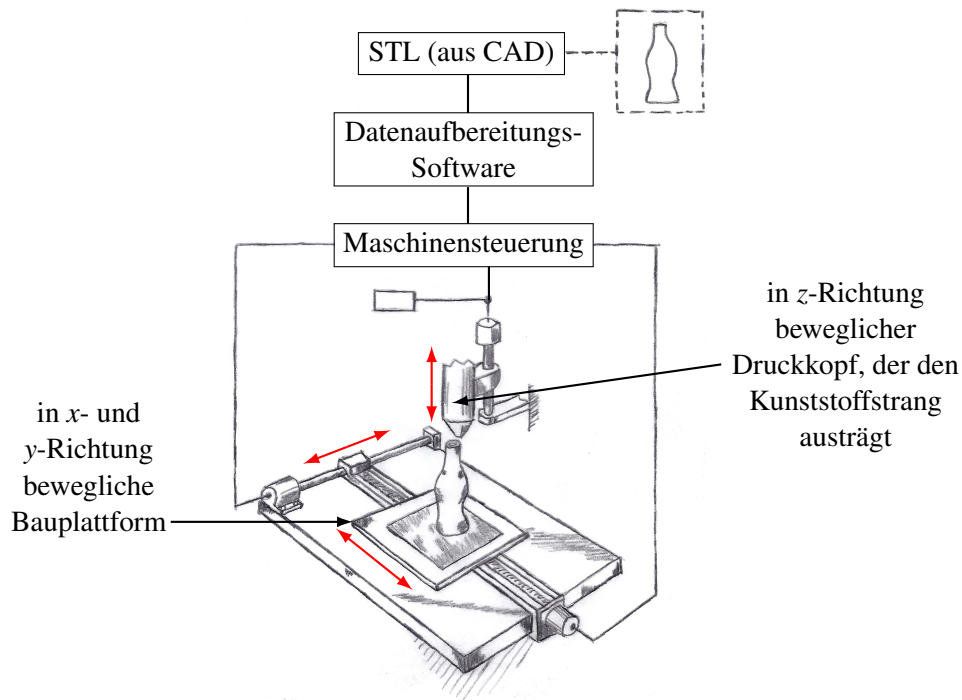


Abbildung 2.5: Grundprinzip von Fused Deposition Modeling. Angelehnt an erstes Patent zu Fused Deposition Modeling von Crump (1992).

der mit Material versorgt wird, welches bei einer bestimmten Temperatur aushärtet, und einer Bauplattform, wobei Dosierkopf und Bauplattform nach einem vorab bestimmten Schema relativ zueinander in x -, y - und z -Richtung verfahren werden, so dass durch kontrollierten Austrag von Material aus dem Dosierkopf auf die Bauplattform Schicht für Schicht dreidimensionale Gegenstände aufgebaut werden können. Der Abstand von der Düse zur vorhergehenden Schicht entspricht dabei der Schichthöhe. Unmittelbar nach dem Austrag des geschmolzenen Materials aus dem Dosierkopf verfestigt sich das Material beim Ablegen auf die Bauplattform bzw. die vorhergehende Schicht. Als Materialien eignen sich dabei beispielsweise selbstaushärtende Wachse, geschmolzene Metalle, aber insbesondere auch bestimmte Kunststoffe. Über die Recheneinheit der Vorrichtung kann so genau die Temperatur des verwendeten Materials reguliert werden.

Abbildung 2.5 zeigt schematisch das Prinzip von FDM. Dies entspricht nur einem Teil der kompletten von Crump (1992) beschriebenen Vorrichtung. Es demonstriert jedoch die Prozessabläufe beim FDM. Als Grundlage für den Bauprozess dienen ähnlich wie bei der Stereolithografie CAD-Daten (Hull, 1986). Hierzu gibt es von 3D Systems (1989) eine Schnittstellen-Spezifikation in Form von STL-Dateien. STL-Dateien beinhalten Oberflächentriangulierungen von dreidimensionalen Objekten. Die STL-Daten werden mit einer Software in Schichtmodelle konvertiert, auf deren Basis die Ansteuerung von Dosierkopf und Bauplattform erfolgen.

Die FDM Technologie nach Crump (1992) kommt in den Rapid Prototyping Systemen der Firma Stratasys zum Einsatz. Comb et al. (1999), Batchelder et al. (1999), Dahlin et al. (2000) und Lombardi et al. (2000) zeigen in ihren Patenten den Aufbau eines Rapid Prototyping Systems basierend auf dieser FDM Technologie. Abbildung 2.6 zeigt den Aufbau dieses Rapid Prototyping Systems. Dieses System beinhaltet zwei jeweils in x - und y -Richtung bewegbare Dosierköpfe, einen zum Austrag des Baumaterials und einen zum Austrag des Stützmaterials. Nach Dahlin et al. (1998) ist dabei stets nur einer der beiden Dosierköpfe im Einsatz. Ein Dosierkopf wird jeweils mit Material versorgt, welches in Form eines Fadens um eine Spule

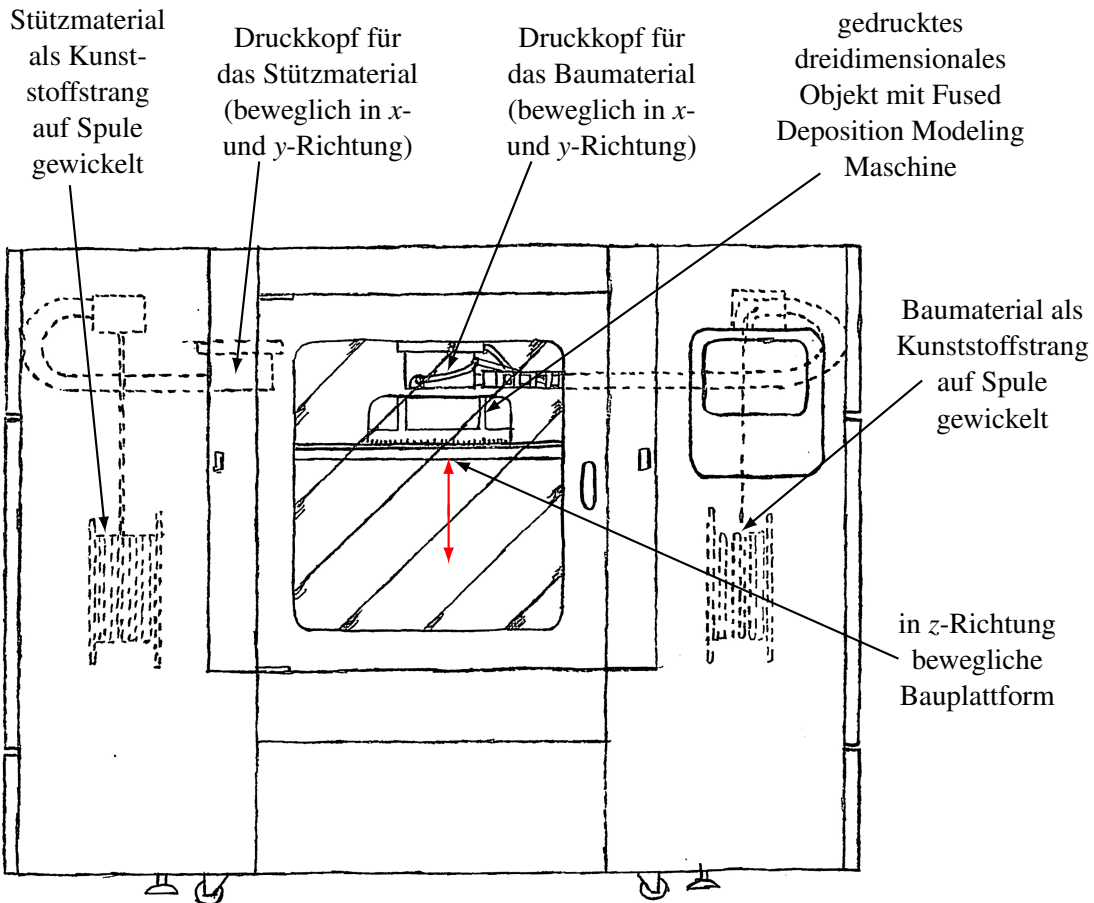


Abbildung 2.6: Aufbau einer Fused Deposition Modeling Vorrichtung von Stratasys. Angelehnt an Dahlin et al. (2000). Im Gegensatz zum ursprünglichen Patent von Fused Deposition Modeling von Crump (1992) ist die Bauplattform hier nur in e_z -Richtung und der Druckkopf / die Druckköpfe in e_x -, e_y -Richtung bewegbar.

gewickelt ist. Im Falle eines Kunststoffstrangs als Baumaterial handelt es sich meist um ein Acrylnitril-Butadien-Styrol (ABS) oder um Polycarbonate (PC). Die Bauplattform selbst ist in diesem System nur in z -Richtung verfahrbar. Das in Abbildung 2.6 gezeigte Rapid Prototyping System bildet die Grundlage der von Stratasys vertriebenen Fortus Serie, deren Systeme speziell für den industriellen Einsatz gemacht sind.

2.5 Vergleich zwischen Spritzgießen und FDM

Tabelle 2.7 zeigt einen Vergleich zwischen Spritzgießmaschinen und FDM Maschinen von Stratasys. Beurteilt wird dabei nach verschiedenen Kriterien. Das sind die Anschaffungskosten für die Maschinen, die Kosten pro Stückzahl, Maß- und Formhaltigkeit, Qualität (Stabilität und optische Anmutung) und der Ausschuss seit der Maschinenanschaffung, d.h. wie viele Teile konnten seit dem Kauf der Maschine produziert werden.

Zäh (2006) gibt einen umfassenderen Einblick in die wirtschaftliche Fertigung mit Rapid-Technologien. Hier ist auch zu sehen, dass sich die Fertigungszeiten und -kosten bei Rapid Prototyping Verfahren auch bei Erhöhung der Bauteilkomplexität nur geringfügig ändern. Im Vergleich dazu steigen die Fertigungszeiten und -kosten bei konventionellen Fertigungsverfahren

	Anschaffungs- kosten	Kosten / Stückzahl	Maß- & Form- haltigkeit	Qualität	Ausschuss seit Ma- schinenan- schaffung
Spritzgießen	200.000 € - 500.000 € alleine für Maschine ohne Werkzeug	siehe Abbildung 2.7a	+	+	siehe Abbildung 2.7b
FDM	ab 10.000 € für Einsteigermo- delle, ansonsten weit darüber	siehe Abbildung 2.7c	o	o	siehe Abbildung 2.7d

Tabelle 2.1: Beurteilung von Spritzgießen und FDM nach verschiedenen Kriterien

mit steigender Bauteilkomplexität immens an. Betrachtet man den gesamten Verlauf einer Produktentwicklung, so liegt ein weiterer Vorteil von Rapid Prototyping Verfahren auf der Hand: so lange beispielsweise die endgültige Geometrie eines Bauteils noch nicht sicher ist, sind die Änderungskosten beim Rapid Prototyping um ein Vielfaches geringer als bei konventionellen Fertigungsverfahren.

2.6 Systeme auf dem Markt

An dieser Stelle soll auf Produkte in der herkömmlichen Spritzgießtechnik und im Rapid Prototyping Bereich, hier also FDM, eingegangen werden.

2.6.1 Spritzgießmaschinen

Das Produktspektrum der meisten Hersteller von Spritzgießmaschinen für Kunststoff umfasst hydraulische Maschinen, hybride Maschinen, elektrische Maschinen und vertikale Maschinen. In Kapitel 2.1 wurden speziell hydraulische Maschinen zur Erklärung der Funktionsweise von Spritzgießmaschinen herangezogen, die prinzipielle Funktionsweise ist allerdings auch bei anderen Antriebskonzepten gleich. Prinzipiell können dem Kunden dabei verschiedene Baureihen, vom Einsteigergerät bis zu individuelleren Expertenmodellen, angeboten werden. Einer der bekanntesten und weltweit führenden Hersteller von Spritzgießmaschinen ist die ARBURG GmbH + Co KG. Hier ist die Einsteigerserie die GOLDEN EDITION, während die individuell anpassbaren Profigeräte unter dem Begriff ALLROUNDER laufen. Weitere bekannte und führende Hersteller von Spritzgießmaschinen sind ENGEL Spritzguss International, Krauss-Maffei, die Ferromatik Milacron Maschinenbau GmbH, die Dr. Boy GmbH + Co KG, Sumitomo (SHI) Demag und Netstal.

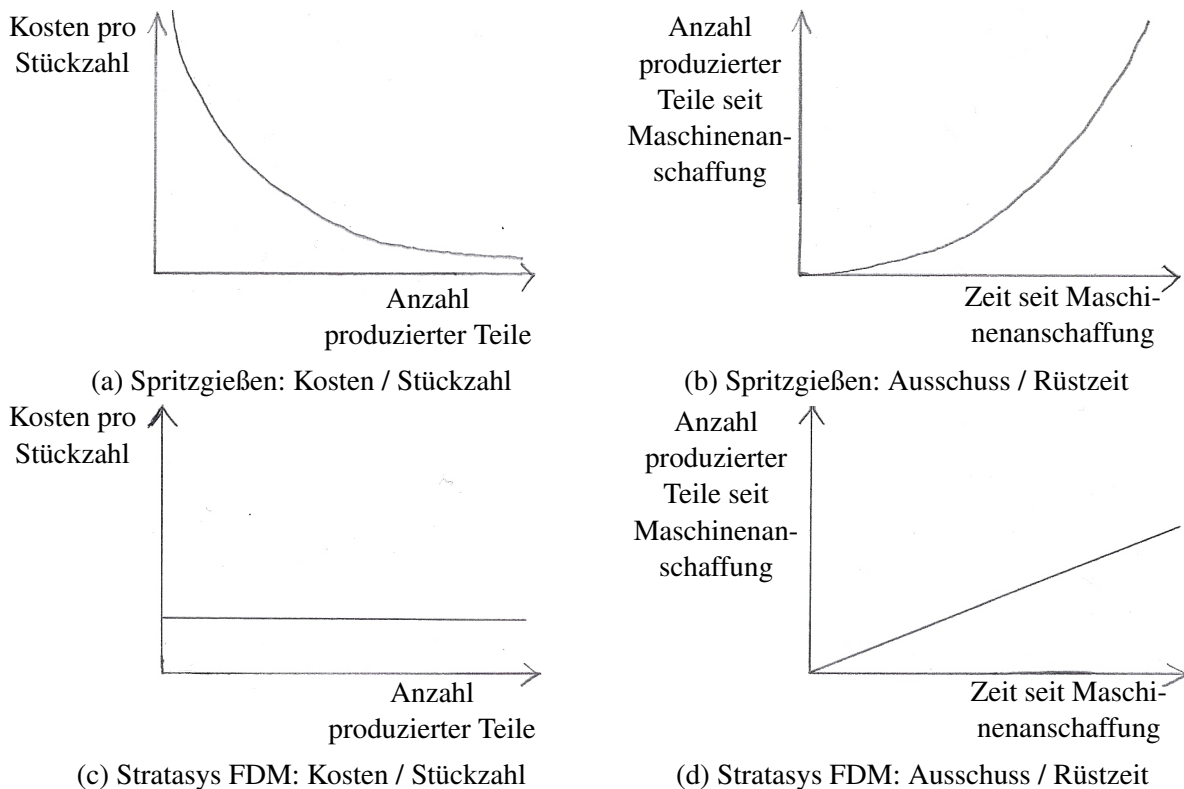


Abbildung 2.7: Vergleich zwischen Spritzgießen und Stratasys FDM

2.6.2 FDM-Maschinen

Marktführer im Rapid Prototyping Bereich bei den FDM-Druckern ist das amerikanische Unternehmen Stratasys, welches auch rechtlicher Inhaber sämtlicher Patente zum Thema FDM ist. Die Einstiegsreihe heißt hier Idea Series. Die Design Series ist die fortgeschrittene Variante, wohingegen es sich bei der Production Series um die Profireihe handelt. Maschinen der Fortus Serie gehören zu den Profigeräten. Dabei sind die Fortus Geräte stark angelehnt an die Patentskizze in Abbildung 2.6, wie bereits in Kapitel 2.4 erwähnt.

2.7 Stand der Forschung

Neben dem Marktführer Stratasys gibt es mittlerweile eine Reihe von FDM-Druckern im Low-Budget Bereich, die für den Hausgebrauch erschwinglich sein sollen. Dies ist dem Trend geschuldet, dass sich auch Privatanwender immer mehr die Möglichkeit wünschen individuelle Bauteile schnell auszudrucken. Was früher Tintenstrahldrucker oder Laserdrucker waren, sind heutzutage 3D FDM-Drucker. FDM-Drucker aus dem Low-Budget Bereich (Pei et al. (2011)) für den Hausgebrauch sind der RapMan, der Ultimaker und MakerBot. Letzteres Unternehmen wurde im Juni 2013 sogar von Stratasys aufgekauft.

2.8 Kritik am Stand der Technik

Im Stand der Technik werden sowohl Verfahren zur Herstellung von Kunststoffteilen beschrieben, welche sich seit mehreren Jahrzehnten in der Kunststoffindustrie bewährt haben, wie speziell Spritzgießen und Extrusion, als auch neuartigere Rapid Prototyping Verfahren wie FDM.

Für eine Analyse der Defizite am Stand der Technik sollen diese alt bewährten Verfahren und das FDM getrennt betrachtet werden.

2.8.1 Echte und lösbare Defizite

Spritzgießen / Extrudieren

Betrachtet man Verfahren wie Spritzgießen und Extrusion, so bezieht sich die Kritik nicht auf die Verfahren selbst. So sind Spritzgießen und Extrusion unangefochten, was die Produktion in großen Stückzahlen, also sozusagen die Anfertigung von Serienteilen, betrifft. Dies betrifft nicht nur die Wirtschaftlichkeit (geringe Kosten pro Teil; sehr geringer Zeitaufwand, da Verfahren automatisiert) bei der Herstellung von Kunststoffteilen, sondern auch die minimale Fehlerquote bei der Produktion. Da sich immer mehr die Forderung nach individuellen Funktionsteilen / Prototypen in kleinen Stückzahlen ergibt, stoßen Verfahren wie Spritzgießen und Extrusion schnell an ihre Grenzen. So ist beim Spritzgießen, Extrudieren und auch den weiteren bewährten Verfahren (siehe Kap. 2.3) die Formenvielfalt von vorne herein eingeschränkt. Bei Extrudern werden in der Regel lediglich Profile mit konstant gleichbleibenden Querschnitten extrudiert. Beim Spritzgießen können beispielsweise keine Teile mit Hinterschneidungen als komplettes Teil produziert werden, da diese nicht entformt werden können, d.h. es gibt keine Möglichkeit, sie aus der Werkzeugform zu bringen. So könnten beispielsweise nie hohle Eier spritzgegossen werden. Insbesondere sind Spritzgießmaschinen aber auch für die Herstellung in kleinen Stückzahlen wirtschaftlich nicht rentabel. So müsste hier für ein individuell anzufertigendes Bauteil extra ein Spritzgießwerkzeug hergestellt werden. Wie umfangreich die Fertigung eines Spritzgießwerkzeugs ist, verdeutlicht Abbildung 2.4 in Kap. 2.1.4. So müsste für ein entsprechendes Bauteil beispielsweise erst eine passende Form gefräst werden. Wie von Johannaber und Michaeli (2001) beschrieben, ist der komplette Prozess nicht nur sehr zeitintensiv, sondern kann außerdem mit einem Kostenaufwand im Millionenbereich (in Euro) verbunden sein. Kosten- und Zeitaufwand wären somit bei kleinen Losgrößen immens hoch pro gefertigtem Teil. Dies ist ein wichtiger Faktor für den kontinuierlichen Aufschwung bei Rapid Prototyping Verfahren.

FDM

Bei einem Rapid Prototyping Verfahren wie FDM sind Kosten- und Zeitaufwand pro zu fertigendem Teil vergleichsweise gering, so lange die Stückzahl nicht groß ist. So fallen in der Regel neben der erforderlichen Arbeitszeit lediglich Kosten für Material und Betreibung einer solchen FDM-Maschine an. Jedoch können die mit dem FDM-Verfahren produzierten Teile in der Regel nicht mit ähnlichen Teilen einer Spritzgießmaschine mithalten, vor allem was die Bauteileigenschaften des fertigen Bauteils betrifft. So gibt es kontinuierlich den Bedarf die Bauteileigenschaften von mit Rapid Prototyping Verfahren gedruckten Teilen zu verbessern. Dies betrifft beispielsweise die Reduzierung der Oberflächenrauheit aber auch die Bauteilstabilität, beispielsweise bei Zugversuchen oder Druckversuchen. So zeigten Ahn et al. (2002), dass FDM-gedruckte Teile aus ABS im Vergleich zu spritzgegossenen Teilen lediglich eine Zugfestigkeit von 65 % – 72 % aufweisen. Des weiteren zeigten sie, dass die Druckfestigkeit nur bei 80 % – 90 % gegenüber spritzgegossenen Teilen liegt. Anitha et al. (2001) und Ahn et al. (2009) zeigten hingegen den kritischen Einfluss von prozessabhängigen Parametern, wie der Schichtdicke, auf die Oberflächenrauheit / -qualität von FDM-gedruckten Bauteilen. Hinzu kommt, dass die gedruckten Teile in ihren Eigenschaften oft anisotrop, d.h. richtungsabhängig sind. So unterscheiden sich Eigenschaften wie Stabilität oder Festigkeit oftmals abhängig von x -, y - und

z-Richtung. FDM-Drucker aus dem Low-Budget Bereich, wie im Stand der Forschung in Kapitel 2.7 aufgelistet, haben zwar mittlerweile ein ordentliches Preis-Leistungs-Verhältnis erreicht, können sich allerdings nicht an der industriell erwarteten Genauigkeit und Qualität messen lassen (Pei et al. (2011)).

2.8.2 Gegenwärtig nicht mit vertretbarem Aufwand lösbare Defizite

Spritzgießen / Extrudieren

Eine generelle Aufhebung der Formeneinschränkung wie beim Spritzgießen und Extrudieren ist auch durch das FDM-Verfahren nicht komplett möglich und auch im Rahmen dieser Arbeit nicht mit vertretbarem Aufwand realisierbar. Es ist nicht möglich Teile, wie beispielsweise Viergelenke, als Ganzes mit einem Verfahren, welches plastifizierten Kunststoff druckt, zu produzieren.

FDM

Es ist mit dem FDM-Verfahren zwar möglich Stützstrukturen für Bauteile mit Überhang zu drucken, jedoch müssen diese nachher erst aufwendig weggebrochen werden, was teils sichtbare Schäden am Bauteil hinterlässt oder das Stützmaterial muss chemisch entfernt werden. Dies hängt vor allem damit zusammen, dass als Stützmaterial auch Kunststoff zum Einsatz kommt und sich dieses Stützmaterial zwangsläufig mit dem Baumaterial verbindet und sich nicht ohne Weiteres lösen lässt. Ein Verfahren, welches Material als Stützmaterial verwendet, das sich während des Bauprozesses nicht mit dem Baumaterial verbindet, ist soweit nicht mit vertretbarem Aufwand realisierbar. Beim FDM-Verfahren wird ein Bauteil aus einem konstant gleich dickem Kunststoffstrang aufgebaut. Ein Verfahren, welches das Volumen des Kunststoffs während des Druckprozesses bei Bedarf variiert, ist ebenfalls nicht mit vertretbarem Aufwand realisierbar.

3 Datenaufbereitung für einen Tropfenerzeuger

Im Folgenden wird der eigene Ansatz erläutert: dies ist die Generierung von prozessspezifischem Maschinencode (G-Code) für einen Tropfenerzeuger, welcher vor Beginn dieser Arbeit nicht möglich war. Darüber hinaus wird auf die Grenzen der im Rahmen dieser Arbeit entwickelten Datenaufbereitung eingegangen und ein Überblick über die (möglicherweise) lösbaren Defizite beim 3D-Druck von Kunststoffteilen gegeben. Zusätzlich gibt es einen kleinen Überblick über die durchzuführenden Experimente zur Evaluierung von gedruckten Bauteilen im Vergleich zum Stand der Technik.

3.1 Eigener Ansatz

Hehl (2010) beschreibt eine Vorrichtung zur Herstellung eines dreidimensionalen Gegenstandes. Im Gegensatz zum Fused Deposition Modeling ermöglicht es diese Vorrichtung durch sequentielles Austragen von Tropfen aus einer Austragseinheit ein Teil im Bauraum aufzubauen. Dies ist in Abbildung 3.1 illustriert. Die Tropfen werden dabei auf die Bauplatzform bzw. das Bauteil selbst aufgetragen und erstarren dort zu „Kugeln“. Als Materialien kommen dabei plastifizierbare Materialien wie Kunststoffe, insbesondere auch in Granulatform, in Frage. Die Vorrichtung besitzt im Vergleich zu einer gewöhnlichen Spritzgießmaschine ebenfalls eine Aufbereitungseinheit zur Plastifizierung des Kunststoffs, unterscheidet sich von dieser aber im Wesentlichen dadurch, dass es kein Spritzgießwerkzeug gibt, in das der plastifizierte Kunststoff hineingespritzt wird. Im Gegensatz dazu besitzt die Vorrichtung eine Austragseinheit bzw. einen Dosierkopf, der es ermöglicht diskrete Volumina aus Kunststoff in Form von Tropfen zu erzeugen. Abbildung 3.2 zeigt eine Skizze dieser Austragseinheit. Das plastifizierte Material wird dabei vom Materialspeicher über einen Kanal in die Austragseinheit befördert. Dadurch steigt in der Austragseinheit der Druck auf das Verschlussmittel (siehe Abbildung 3.3), welches aus einem elastisch verformbaren Festkörpergelenk und der Düsennadel besteht (siehe Abbildung 3.4). Unter diesem Druck wird das Verschlussmittel nach oben gedrückt, wodurch sich die Düse selbst kurzzeitig öffnet. Durch einen Piezo wird ein Gegendruck auf das Verschlussmittel ausgeübt. Dadurch kann aus der Düse ein Tropfen gelöst und die Düse durch die Düsennadel wieder verschlossen werden. Allgemein müssen zur Überwindung der Adhäsionskräfte des Kunststoffs große Drücke und Temperaturen angewandt werden. Die Öffnungszeit der Austragsblende liegt im Bereich von wenigen Millisekunden. Die Größe (Durchmesser) der Austragsöffnung liegt im Bereich weniger Zehntelmmillimeter.

Ein wichtiger Hauptunterschied zu konventionellen FDM-Anwendungen besteht in der Möglichkeit, gezielt verschiedene Mengen an Material zu platzieren. Ein weiterer Unterschied ist die Bewegung der Bauplatzform während des Druckprozesses. Im Gegensatz dazu befindet sich die Düse an der Austragseinheit an einer fest definierten Position / in einer fest definierten Lage. Für FDM-Anwendungen existieren bereits eine Vielzahl von Software-Lösungen zur Erzeugung von ausführbarem Maschinencode zum gezielten Ablegen eines Kunststoffstrangs um ein

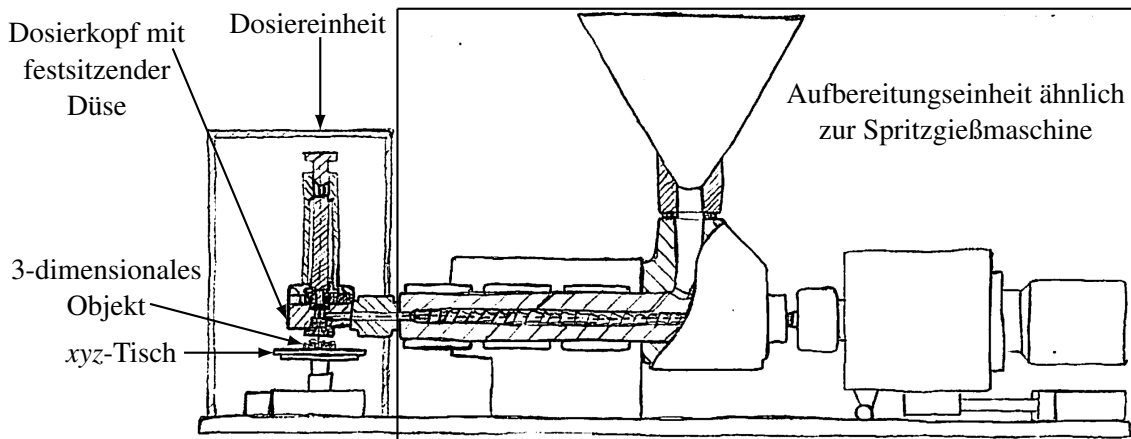


Abbildung 3.1: Tropfenerzeuger. Angelehnt an Hehl (2010).

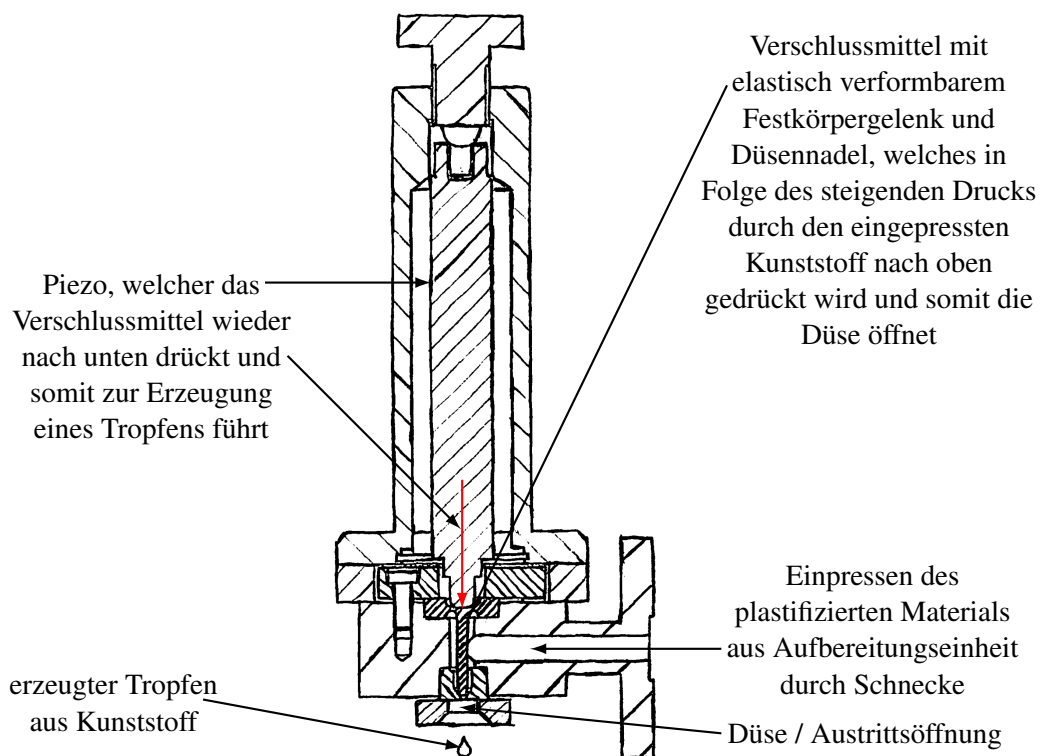


Abbildung 3.2: Austragseinheit des Tropfenerzeugers. Angelehnt an Hehl (2010).

Bauteil aufzubauen. Für die hier beschriebene Vorrichtung des Tropfenerzeugers existierte eine solche Software nicht und musste erst entwickelt werden. Jeder neue Druckprozess erfordert eine eigene prozessspezifische Software zur Generierung von ausführbarem Maschinencode. Im Vergleich zu anderen tropfenbasierten Verfahren, kommt es in diesem Ansatz vor allem darauf an, durch das gezielte Setzen von Tropfen Bauteile zu produzieren, welche gewisse Bauteileigenschaften aufweisen. So sollen die mit dieser Vorrichtung produzierten Bauteile im Vergleich zum strangbasierten Konkurrenzverfahren FDM und sonstigen tropfenbasierten Druckverfahren festigkeitsoptimiert sein und sich durch eine geringere Oberflächenrauheit auszeichnen, um die oben beschriebenen Defizite am Stand der Technik erfolgreich aufzulösen. Die Lösungsstruktur umfasst die für den Lösungsprozess erforderlichen Systemkomponenten, welche auf der Hardware-Seite der xyz-Tisch und die Düse und auf der Software-Seite die STL-Dateien sind, welche als Ausgangsbasis für die entwickelte Software dienen. Die Lösungsstruktur wird

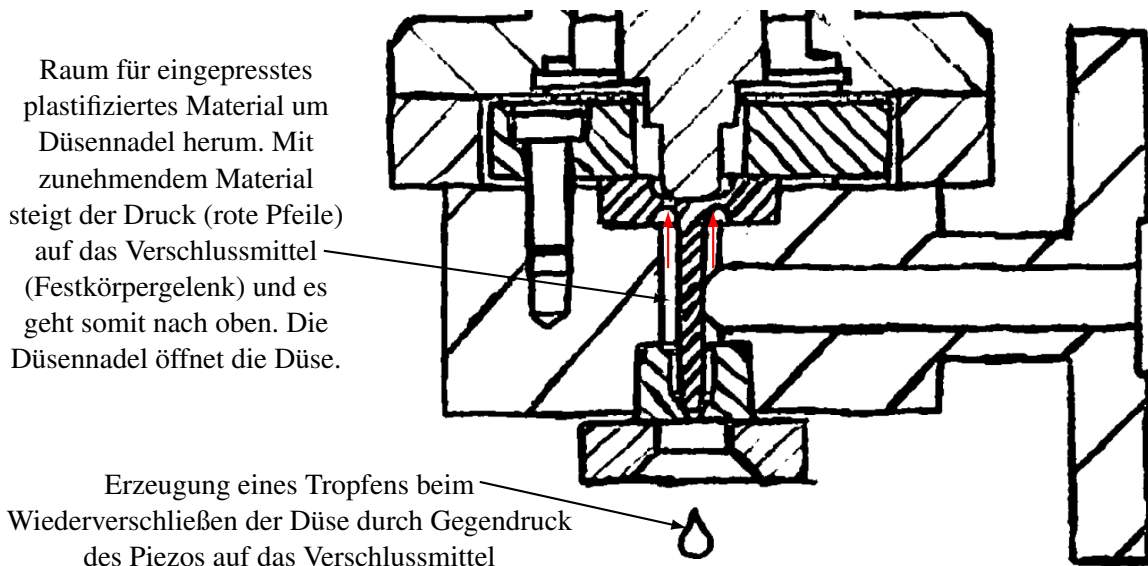


Abbildung 3.3: Detailliertere Ansicht der Austrageinheit des Tropfenerzeugers im Bereich des Verschlussmittels und der Düse. Angelehnt an Hehl (2010).

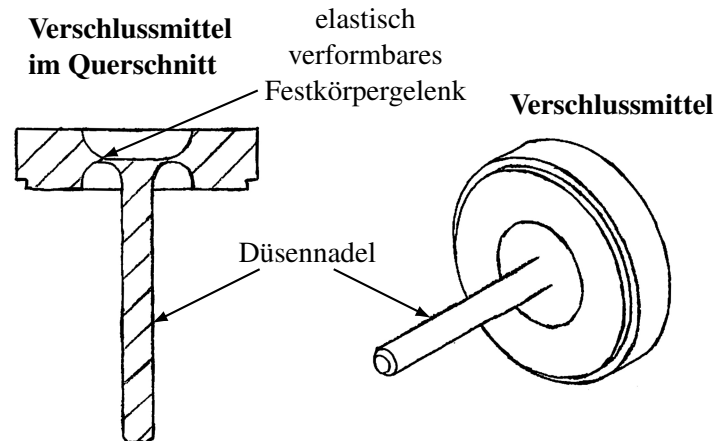


Abbildung 3.4: Verschlussmittel der Austrageinheit des Tropfenerzeugers. Angelehnt an Hehl (2010).

in Kapitel 4 genauso näher erläutert, wie auch die Lösungsprozesse selbst, das heißt, wie aus den STL-Dateien (Eingabedaten) mit Hilfe der entwickelten Software ausführbarer prozessspezifischer Maschinencode erzeugt wird. Schwaiger et al. (2013) zeigt hierzu bereits einen kurzen Überblick.

3.2 Abgrenzung

Das wichtigste Ziel dieser Arbeit ist die:

- **Optimierte schichtweise Datenaufbereitung auf Tropfenbasis:** Im Rahmen der implementierten Software-Module zur Code-Generierung für den Tropfenerzeuger werden die Tropfen beim Berandungsdruck optimiert gesetzt. Die Füllung wird innerhalb einer Schicht so gesetzt, dass sie möglichst dicht ist. Dies soll sich positiv auf die Eigenschaften der gedruckten Bauteile auswirken.

Die Hardware des Tropfenerzeugers wurde im Rahmen dieser Arbeit nur vom Entwickler selbst angepasst. Diese Arbeit beschränkt sich hingegen rein auf die Software-Seite. Folgende Funktionalitäten wurden bei der Softwareentwicklung nicht realisiert:

- **STL-Korrektur:** CAD-Daten in Form von STL-Dateien werden nicht auf mögliche Fehler oder Probleme überprüft. STL-Dateien beinhalten eine Triangulierung der Oberfläche von 3D-Objekten. Die Normalen eines Dreiecks zeigen dabei nach außen. Sollte dies nicht für jedes Dreieck der Fall sein, dann kann dies zu Problemen führen, genauso wie mögliche Lücken oder Überschneidungen im Dreiecksgitter. Die Korrektur von problematischen STL-Dateien ist nicht Bestandteil dieser Arbeit. Einige Probleme können zwar mit Sicherheit in der implementierten Software gelöst werden, allerdings sind für den Einsatz der Software möglichst fehlerfreie STL-Dateien zu empfehlen, um Folgefehler aufgrund der STL-Daten zu vermeiden.
- **Gedächtnis zwischen Schichten:** Wie bereits oben erwähnt, werden die Daten für den Druckprozess schichtweise optimiert. Das heißt also, dass es kein Gedächtnis zwischen benachbarten Schichten gibt. So werden Tropfen beispielsweise nicht auf Lücke gesetzt zwischen benachbarten Schichten.
- **Stützstrukturen:** Es wurde zwar bereits ein Ansatz für Stützstrukturen für einfache Bauteile im Rahmen dieser Arbeit entwickelt. Jedoch ist die Umsetzung aus materialtechnischer Sicht noch nicht ausgereift genug. So lassen sich Stützmaterial und Baumaterial in der Regel nicht ohne Weiteres trennen, da sich die verwendeten Kunststoffe sehr stark binden. Darum müsste erst ein alternatives Konzept für Stützstrukturen entwickelt werden. So könnten Stützen beispielsweise aus einem Material generiert werden, welches sich nicht mit dem plastifizierten Baumaterial bindet. Daher sind Stützstrukturen im Rahmen dieser Arbeit nicht relevant. Des Weiteren ist geplant, durch die Entwicklung von Maschinen mit mehr als drei Freiheitsgraden (z.B. 5-Achs-Systeme) komplett auf den Einsatz von Stützmaterial verzichten zu können.

3.3 Liste (möglicherweise) lösbarer Defizite

Das von mir zu erforschende System zeichnet sich durch folgende quantitativen Eigenschaften aus und hebt sich damit von allen bisher weltweit bekannten Systemen ab:

- Es ist im Vergleich zu anderen Verfahren aus dem Rapid Prototyping Bereich, das System, das Bauteile aus Kunststoff druckt mit der weltweit
 - niedrigsten Oberflächenrauheit,
 - höchsten Stabilität (Zugproben),
 - höchsten Vielfalt an druckbaren Kunststoffen.
- Es ist im Vergleich zu anderen Verfahren aus dem Rapid Prototyping Bereich das einzige System, das Bauteile aus Kunststoff tropfenbasiert druckt.

Das von mir erforschte System ist das erste System, das die folgenden Defizite des bisherigen Stands der Technik hinreichend auflöst und den folgenden messbaren Nutzen hat:

- Das System gehört wie das FDM zu den Rapid Prototyping Verfahren und ermöglicht im Vergleich zum Spritzgießen oder Extrudieren das kostengünstige und zeitsparende Produzieren von individuellen Kunststoffteilen / -prototypen in kleinen Stückzahlen.

- Im Vergleich zum FDM können Defizite beim Berandungsdruck gelöst werden. Da im Vergleich zum FDM-Verfahren kein Kunststoffstrang abgelegt wird, sondern die Berandung aus einzelnen Tropfen besteht, können diese auch optimal bei Bauteilen mit spitzen Ecken positioniert werden.
- Durch das automatische exakte Positionieren diskreter Kunststoffvolumina können somit auch gezielt Bauteileigenschaften positiv beeinflusst werden. Dies ist in diesem Umfang beim FDM nicht möglich und schafft somit ein neues Spektrum mit Rapid Prototyping druckbaren Funktionsbauteilen aus Kunststoff.
- Das automatische Positionieren von Tropfen im Abstand der Tropfenbreite auf der Bauteiloberfläche ermöglicht Oberflächen mit einer im Vergleich zu anderen Verfahren im Rapid Prototyping Bereich nicht erreichbaren niedrigen Oberflächenrauheit der gedruckten Kunststoffteile.
- Durch das Zusammenspiel zwischen Bauteilumrandung und Füllung können im Vergleich zum FDM-Verfahren Bauteile mit einer vergleichsweise hohen Stabilität gedruckt werden, indem sich prozessbasiert Tropfen zwischen und innerhalb der Schichten verkeilen.

3.4 Experimente zur Evaluierung von gedruckten Bauteilen

Neben einer kontinuierlichen Entwicklung von neuen Software-Features, welche genau an die Anforderungen des Druckprozesses angepasst werden müssen, müssen auch Experimente bezüglich der Qualität der gedruckten Bauteile durchgeführt werden, um die Ergebnisse auch quantitativ zu rechtfertigen und zu zeigen, dass die Defizite am Stand der Technik durch das gezielte Setzen von Tropfen mit dem hier entwickelten Ansatz kompensiert werden.

3.4.1 Vergleich mit Spritzgießverfahren

Für den Vergleich mit dem Spritzgießverfahren, werden ausgewählte Funktionsbauteile mit individuellen Geometrien (z.B. Schnapphaken) mit dem Tropfenerzeuger gedruckt. Hierbei werden die Kosten c und die Zeit t für das Drucken an der Maschine festgehalten und ein Vergleich zum Spritzgießen gezogen. Erwartet wird hierbei, dass die Kosten und Zeiten für den Druckprozess viel geringer sind als beim Spritzgießen, da der Bau eines individuellen Spritzgießwerkzeugs nicht erforderlich ist. Der in diesem Ansatz verwendete Tropfenerzeuger, soll dabei nicht mit Spritzgießmaschinen konkurrieren, da Spritzgießmaschinen nach wie vor unangefochten sind bei der Produktion in hohen Stückzahlen bei gleichzeitiger Nullfehlertoleranz. Die Vorteile des Tropfenerzeugers befinden sich in der Anwendung selbst: es sollen möglichst schnell mit geringem Aufwand und niedrigen Kosten individuelle Funktionsteile in kleinen Stückzahlen produziert werden können.

3.4.2 Vergleich mit FDM-Verfahren

Da es sich beim FDM ebenfalls um ein Verfahren aus dem Rapid Prototyping Bereich handelt, muss sich der in diesem Ansatz verwendete Tropfenerzeuger vor allem an der Bauteilqualität von FDM-Maschinen messen lassen. Da beim FDM-Verfahren Kunststoffstränge abgelegt werden und nicht gezielt Tropfen gesetzt werden können, wird erwartet, dass sich dies auch positiv auf die Bauteileigenschaften der mit dem Tropfenerzeuger gedruckten Bauteile auswirkt. So

sollen in einem ersten Experiment $n = 11$ genormte Referenzbauteile einmal mit einer professionellen FDM-Maschine von Stratasys (FORTUS Serie) und einmal mit dem Tropfenerzeuger gedruckt werden. Danach soll bei jedem der Teile die Oberflächenrauheit als erstes Kriterium der Bauteilqualität gemessen und somit ein Fazit zwischen FDM und Tropfenerzeuger gezogen werden. Ein zweites Experiment sieht wie folgt aus: durch das Setzen von Tropfen wird erwartet, dass sich Tropfen auch stärker von Schicht zu Schicht verkeilen als dies beim FDM mit den Kunststoffsträngen der Fall ist. Deshalb sollen jeweils mit beiden Verfahren $n = 10$ Norm-Zugproben gedruckt werden und damit die Belastbarkeit der gedruckten Teile getestet werden. Dies entspricht im Gegensatz zur Oberflächenrauheit einem mechanischem Kriterium. Auf diese Weise soll wiederum ein Fazit zwischen FDM und Tropfenerzeuger gezogen werden.

4 Konzept und Systementwurf

In diesem Kapitel werden die Lösungsstruktur und die Lösungsprozesse zur Generierung von prozessspezifischem Code für den Tropfenerzeuger erläutert.

4.1 Konzept der Hardware

Abbildung 4.1 zeigt schematisch die Zusammenhänge zwischen Düse und Bauplattform. Ein Modell des Tropfens zeigt Abbildung 4.2. Durch die homogene Transformationsmatrix ${}^{noz}\mathbf{T}_{tab}$ kann die Lage der Bauplattform des Tisches (table) bezüglich der Düse (nozzle) beschrieben werden (eine genaue Erklärung zu homogenen Transformationsmatrizen folgt in Kapitel 5). Die \mathbf{e}_z -Achse der Düse zeigt dabei in die entgegengesetzte Richtung der \mathbf{e}_z -Achse des Tisches, d.h. das Material, das aus der Düse kommt, wird lotrecht auf die Bauplattform bzw. das Bauteil selbst gedruckt. Die \mathbf{e}_x - \mathbf{e}_y -Ebenen der beiden Koordinatensysteme sind parallel. Nach dem Montieren einer neuen Bauplatte auf den Tisch und vor dem Start des Druckprozesses selbst, muss also darauf geachtet werden, dass der Abstand zwischen Düse und Bauplattform optimal ist. Hierfür existiert eine Kalibrierplatte aus Metall, welche nach dem Montieren der Bauplatte auf die Bauplatte selbst gelegt wird. Ausgehend von einer festen Nullstellung

$$\mathbf{t}_0 = \begin{pmatrix} x_{t_0} \\ y_{t_0} \\ d \end{pmatrix} \quad (4.1)$$

der Achsen des Tisches im tab -Koordinatensystem, wird der Tisch solange nach oben (in Richtung der \mathbf{e}_z -Achse) in Richtung Düse gefahren, bis die Düse auf die Kalibrierplatte trifft. d entspricht dabei der Dicke der Kalibrierplatte in mm . Durch die Kalibrierplatte kann die eigentliche Bauplattform nicht durch die Düse beschädigt werden. Mit der Matrix ${}^{noz}\mathbf{T}_{tab}$ gilt in dieser Position dann:

$$\begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} \stackrel{!}{=} \begin{pmatrix} 1 & 0 & 0 & x_t \\ 0 & -1 & 0 & y_t \\ 0 & 0 & -1 & z_t \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_{t_0} \\ y_{t_0} \\ d \\ 1 \end{pmatrix} = \begin{pmatrix} x_t + x_{t_0} \\ y_t - y_{t_0} \\ z_t - d \\ 1 \end{pmatrix}, \quad (4.2)$$

was nichts anderes bedeutet, dass der Ursprung des noz -Koordinatensystems mit dem Berührungspunkt auf der Kalibrierplatte übereinstimmt. Es ergibt sich somit:

$$\begin{pmatrix} x_t \\ y_t \\ z_t \end{pmatrix} = \begin{pmatrix} -x_{t_0} \\ y_{t_0} \\ d \end{pmatrix}. \quad (4.3)$$

Folglich kann der Tisch über die inverse Kinematik so angesteuert werden, dass an einer beliebigen Position des Bauraums gedruckt werden kann.

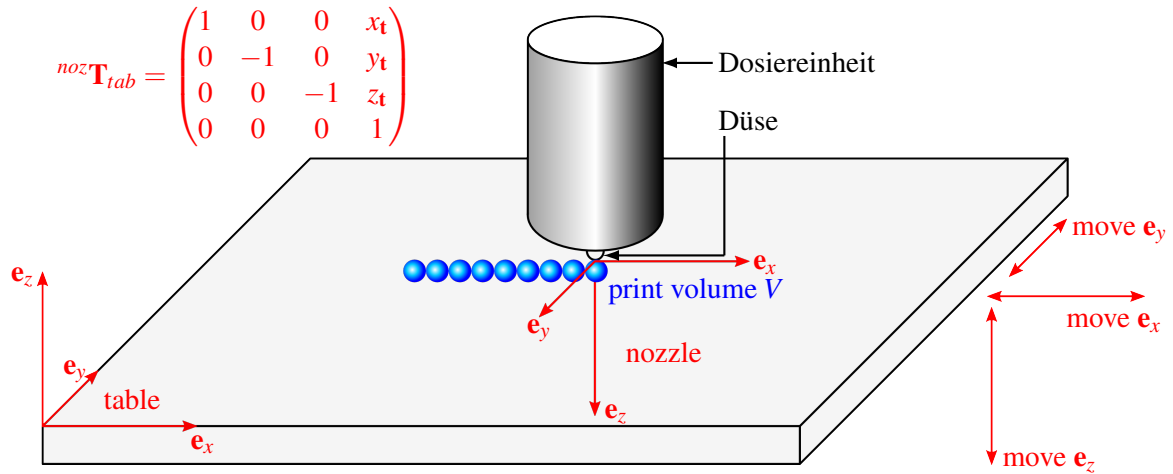


Abbildung 4.1: Modell von Düse und bewegbarem xyz -Tisch. Die Düse ändert die Lage bezüglich des Weltkoordinatensystems nicht. Darum reicht es von einem festen Koordinatensystem an der Düse auszugehen.

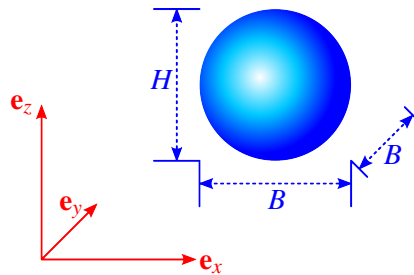


Abbildung 4.2: Tropfen mit Tropfenbreite B und Tropfenhöhe H .

4.2 Konzept des Druckprozesses

Wird nun über die Softwareoberfläche der Maschine ein Bauteil (in diesem Fall der G-Code selbst) in den Bauraum geladen, so kann dieses im Bauraum platziert werden. Der G-Code zeichnet sich dadurch aus, dass das zugrunde liegende Koordinatensystem gco mit dem des Tisches tab übereinstimmt, d.h. es gilt:

$${}^{tab}\mathbf{T}_{gco} = \mathbf{I}, \quad (4.4)$$

wobei \mathbf{I} im Folgenden die Einheitsmatrix bezeichne. Der G-Code kann innerhalb der e_x - e_y -Ebene des Bauraums verschoben und um den Winkel α um die e_z -Achse gedreht werden, d.h. für das Bauteil ($part$) gilt:

$${}^{tab}\mathbf{T}_{part} = \underbrace{{}^{tab}\mathbf{T}_{gco}}_{=\mathbf{I}} {}^{gco}\mathbf{T}_{part} = {}^{gco}\mathbf{T}_{part} := \begin{pmatrix} \cos \alpha & -\sin \alpha & 0 & x_t \\ \sin \alpha & \cos \alpha & 0 & y_t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}. \quad (4.5)$$

Insgesamt ist die Lage eines Bauteils relativ zur Düse also gegeben durch:

$${}^{noz}\mathbf{T}_{part} = {}^{noz}\mathbf{T}_{tab} {}^{tab}\mathbf{T}_{part}. \quad (4.6)$$

Auf diese Art und Weise kann aus dem Maschinencode und der Platzierung des Bauteils im Bauraum jederzeit die Stellung des xyz -Tisches errechnet werden. Genauso funktioniert dies,

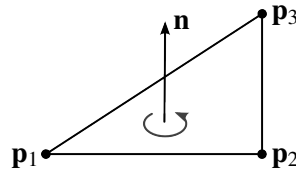


Abbildung 4.3: Darstellung eines Dreiecks mit äußerer Dreiecksnormale

falls mehrere Bauteile gleichzeitig im Bauraum platziert werden. In diesem Fall existiert für jedes Bauteil eine eigene Transformationsmatrix ${}^{tabr}\mathbf{T}_{part}$.

Sind einmal sämtliche Vorbereitungen an der Maschine durchgeführt und alle Bauteile passend im Bauraum platziert, so kann der Druckprozess gestartet werden. Folglich wird der G-Code an die Maschinensteuerung geschickt und entsprechend der vereinbarten Konventionen interpretiert. So kann beispielsweise an einer bestimmten Position im Bauraum ein Tropfen gesetzt werden. Diese Information ist dem G-Code selbst zu entnehmen. Das zugrunde liegende Modell basiert dabei auf der Annahme, dass die Maschine Tropfen der Breite B in \mathbf{e}_x - und \mathbf{e}_y -Richtung und der Höhe H in \mathbf{e}_z -Richtung drucken kann. Die Tropfenhöhe H ist dabei etwas kleiner als die Tropfenbreite B , da der Tropfen beim Auftrag auf eine Schicht etwas zusammensackt. Auf die genauen prozessspezifischen Werte wird später in der Realisierung dieser Arbeit eingegangen.

4.3 Nutzung von STL-Dateien

Grundlage für die Erzeugung von G-Code sind sogenannte STL-Dateien. Sie stellen die Eingabedaten der im Rahmen dieser Arbeit entwickelten Software-Module dar, wohingegen die Ausgabedaten in Form von prozessspezifischen G-Code-Dateien erzeugt werden. Das STL-Format ist ein Daten-Format aus dem Jahre 1989 zur Programmierung von Stereolithografie-Maschinen. Es eignet sich zur Beschreibung von Oberflächen in der Computergrafik. Gewöhnlich können auch CAD-Daten als STL exportiert werden.

4.3.1 Das STL-Format

An dieser Stelle wird das STL-Format im ASCII-Format beschrieben. Ein STL kann somit auch in einem Texteditor erstellt werden. Auf das STL-Format im Binär-Format soll an dieser Stelle nicht explizit eingegangen werden. Die mathematische Repräsentierung bleibt hierbei im Vergleich zum ASCII-Format gleich. In einer STL-Datei wird die Oberfläche eines dreidimensionalen Körpers beschrieben. Die Oberfläche setzt sich aus einem Netz von Dreiecken zusammen, welche eine geschlossene Oberfläche bilden (Triangulierung). Ein Dreieck wird immer durch drei Eckpunkte \mathbf{p}_1 , \mathbf{p}_2 , \mathbf{p}_3 (auch „vertices“ genannt) und einen Normalenvektor \mathbf{n} beschrieben. Die Rechte-Hand-Regel definiert dabei was außen ist (siehe Abbildung 4.3).

Allgemein gilt der Zusammenhang:

$$\mathbf{n} = \frac{(\mathbf{p}_2 - \mathbf{p}_1) \times (\mathbf{p}_3 - \mathbf{p}_1)}{\|(\mathbf{p}_2 - \mathbf{p}_1) \times (\mathbf{p}_3 - \mathbf{p}_1)\|} \quad \text{für } (\mathbf{p}_2 - \mathbf{p}_1) \nparallel (\mathbf{p}_3 - \mathbf{p}_1) \quad (4.7)$$

Bei der Erstellung eines STLs ist darauf zu achten, dass die Angabe des Normalenvektors konsistent ist mit der Gleichung 4.7. Der Normalenvektor zeigt dabei immer aus der Oberfläche des dreidimensionalen Körpers heraus. STL-Dateien, welche nicht dieser Bedingung genügen

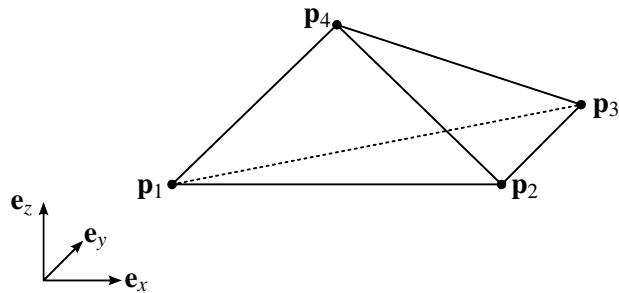


Abbildung 4.4: Tetraeder, der einfachste STL-Körper

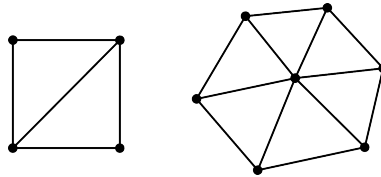


Abbildung 4.5: Beispielhafte Zerlegung von Flächen, wie Quadrate oder Vielecke, in Dreiecke (Triangulierung)

(Stichwort: „gedrehte Dreiecke“), sind fehlerhaft und werden hier nicht betrachtet. Die Syntax des STL-Formats ist folgende:

```

solid name
  {
    facet normal  $x_n$   $y_n$   $z_n$ 
      outer loop
        vertex  $x_{p_1}$   $y_{p_1}$   $z_{p_1}$ 
        vertex  $x_{p_2}$   $y_{p_2}$   $z_{p_2}$ 
        vertex  $x_{p_3}$   $y_{p_3}$   $z_{p_3}$ 
      endloop
    endfacet
  }+
endsolid
  
```

{...}⁺ bedeutet, dass dieser Teil ein oder mehrmals wiederholt wird. Der Inhalt beschreibt jeweils ein Dreieck, wie es in Abbildung 4.3 zu sehen ist. Ein Tetraeder, wie er in Abbildung 4.4 zu sehen ist, stellt den einfachsten Körper dar, der im STL-Format beschrieben werden kann. Er besteht aus vier Dreiecksflächen. Flächen, wie Quadrate oder Vielecke, können wie in Abbildung 4.5 trianguliert werden. Ein einfacher Würfel besteht aus sechs quadratischen Grundflächen und kann somit durch zwölf Dreiecksflächen dargestellt werden. Die Normalen der Dreiecke zeigen dabei nach außen (siehe Abbildung 4.6).

4.3.2 Notation von Vertex List und Facet List

Für eine effiziente datentechnische Verarbeitung von STL-Dateien empfiehlt sich eine Repräsentation der Objekte getrennt nach Ecken (Vertices) und Flächen (Facets). Die Vertex List **VL** besteht aus allen Eckpunkten der Triangulierung:

$$\mathbf{VL} := [\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3, \dots, \mathbf{p}_n] . \quad (4.8)$$

Die Zahl n ist dabei davon abhängig, ob jeder Eckpunkt der Triangulierung nur einmal gespeichert wird oder ob jeder Eckpunkt für jedes Dreieck, analog zur STL-Syntax, genau einmal

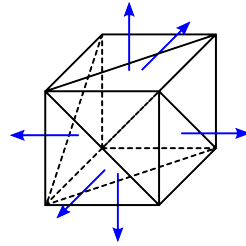


Abbildung 4.6: Triangulierung eines Würfels inklusive äußerer Normalenvektoren

abgespeichert wird. In ersterem Fall entspricht n der Anzahl der Gitterpunkte der Triangulierung. Die Facet List \mathbf{FL} gibt dann an welche Punkte aus der Vertex List ein Dreieck definieren. Angegeben wird dabei nur die Position im Array \mathbf{VL} an der sich der jeweilige Punkt befindet. Jede Dreiecksfläche besteht aus drei Punkten und bei der Anordnung der Positionsindizes ist darauf zu achten, dass die Reihenfolge konform ist mit der Berechnung der äußeren Dreiecksnormale nach der Rechte-Hand-Regel. Für den Tetraeder aus Abbildung 4.4 beispielsweise wären die Vertex List und die Facet List folgendermaßen definiert:

$$\mathbf{VL} := [\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3, \mathbf{p}_4], \quad (4.9)$$

$$\mathbf{FL} := \begin{bmatrix} 1 & 3 & 2 \\ 1 & 4 & 3 \\ 1 & 2 & 4 \\ 2 & 3 & 4 \end{bmatrix}. \quad (4.10)$$

Eine Zeile in \mathbf{FL} definiert dabei eine Dreiecksfläche. Allgemein gilt:

$$\mathbf{FL} \in \mathbb{R}^{m \times 3}, \quad m \leq n, \quad (4.11)$$

wobei m der Anzahl der Dreiecke entspricht, aus denen die Oberfläche des Körpers aufgebaut ist.

4.4 Prozessbeschreibung und Struktur der Software

Abbildung 4.7 zeigt eine grobe Übersicht über die Abläufe bis zum Starten eines Druckjobs am Tropfenerzeuger. So wird zunächst von einem Konstrukteur die gewünschte zu druckende Geometrie mit einer CAD-Software, z.B. CATIA, erzeugt. Die Geometrie wird am Ende als STL-Datei exportiert. Diese dient dann als Eingabe für die im Rahmen dieser Arbeit entwickelte Software zum Zerlegen der STL-Daten in ein prozessoptimiertes schichtbasiertes Format und zur automatischen Erzeugung von ausführbarem druckbarem Maschinencode (G-Code). Wurde der G-Code erzeugt, so können die Daten jederzeit an die Maschinensteuerung gesendet und somit ein Druckjob gestartet werden.

Die Software selbst wird mit der Programmiersprache C++ unter Visual Studio 2010 auf dem Betriebssystem Windows 7 (64 Bit) entwickelt und als DLL (Dynamic Link Library) kompiliert. Auf diese Weise können die entwickelten Funktionalitäten unter Kenntnis einer Schnittstellenfunktion zur DLL über eine grafische Benutzeroberfläche (GUI) aufgerufen und G-Code erzeugt werden. Diese GUI wird vom Hersteller des Tropfenerzeugers entwickelt und dient in erster Linie für eine benutzerkomfortable Bedienung und zur Visualisierung des erzeugten spezifischen G-Codes.

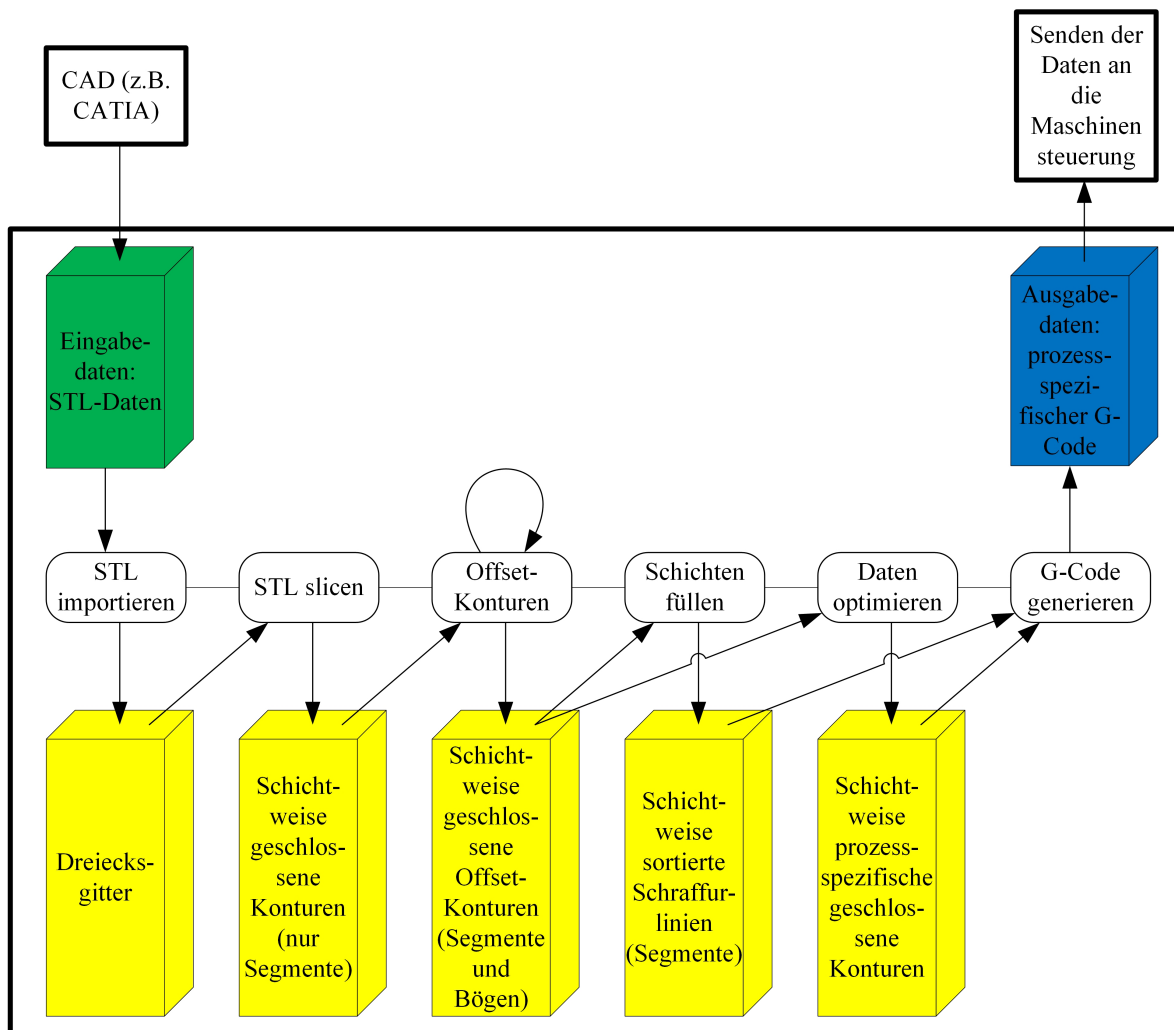


Abbildung 4.7: Ablaufdiagramm und Struktur der Software-Module bestehend aus sechs wichtigen Prozessen. Eingabedatenblock (grün), Ausgabedatenblock (blau), temporäre Datenblöcke (gelb).

Die entwickelte Datenaufbereitungs-Software umfasst dabei im Wesentlichen sechs wichtige Software-Module. Dazu gehören das Importieren der STL-Daten, das Slicen der STL-Daten, die Berechnung von Offset-Konturen, das Füllen der Schichten, die Optimierung der Daten auf Basis des oben erläuterten Druckprozesses und die Generierung von Maschinencode (G-Code). Abbildung 4.7 zeigt die Ablaufprozesse innerhalb der Software. Grundlage bilden die bereits in Kapitel 4.3 vorgestellten Eingabedaten in Form von STL-Dateien. Sie werden so verarbeitet, dass als Ausgabedaten prozessspezifischer G-Code entsteht, welcher am Drucker ausgeführt werden kann.

Im beschränkten Maße finden dabei auch Bestandteile bereits existierender Algorithmen zum Slicen (Tata et al., 1998, Pandey et al., 2003a, 2003b, Vatani, 2009), zum Generieren von Offset-Konturen (Lai et al., 2000, Jibin et al., 2004, Chen und McCains, 2005, Seo et al., 2005, Dhanik und Xirouchakis, 2010, Zhao et al., 2011), zum Füllen (Ma und He, 1999) und zur Generierung von Maschinencode (Chang, 2004), Einfluss in diese Arbeit.

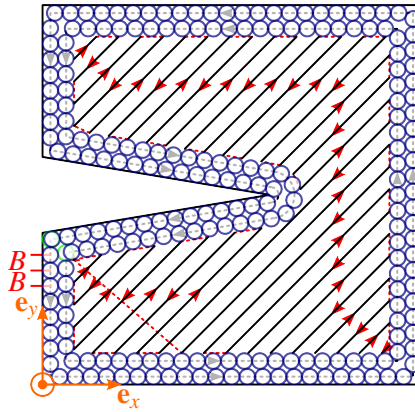


Abbildung 4.8: Beispiel für prozessspezifischen Maschinencode (G-Code) für einen Würfel mit Keil (eine Schicht): Punktweises Setzen von Tropfen beim Berandungsdruck im Abstand der Tropfenbreite B , die Druckrichtung der Konturen ist mit den grauen Pfeilen dargestellt. Füllen mit Schraffursegmenten (automatisches Setzen von Tropfen) in Abstimmung mit der Berandung. Leerfahrten sind rot gestrichelt, die Verfahrswege der Düse auf den schwarz eingezeichneten Schraffurlinien sind mit roten Pfeilen dargestellt. Die Reihenfolge beim Drucken ist variierbar: es können sowohl erst die Konturen als auch die Füllung gedruckt werden. Des Weiteren ist auch die Druckreihenfolge der Konturen variierbar.

4.5 Prozessspezifischer Maschinencode

Generierter Maschinencode (G-Code) für den hier verwendeten Tropfenerzeuger ist eine ASCII-Datei mit prozessspezifischen Positionier- bzw. Druckbefehlen, welche im Folgenden genauer erläutert werden. Die Dateiergung ist „.gco“. Vor den Befehlen befindet sich ein Header, welcher eine Übersicht über die teils material- und teils prozessspezifischen Parameter bietet.

Abbildung 4.8 zeigt ein Beispiel für prozessspezifischen G-Code für einen Würfel mit Keil. Dabei existieren im Wesentlichen drei wichtige Verfahrensbefehle, welche folgendermaßen definiert sind:

$$G01 \ Xx \ Yy \ Zz \quad (4.12)$$

$$G25 \ Xx \ Yy \ Zz \quad (4.13)$$

$$G26 \ Xx \ Yy \ Zz \quad (4.14)$$

wobei x , y bzw. z jeweils für die entsprechenden Koordinatenwerte stehen. Der G01-Befehl in 4.12 ist ein reiner Positionierbefehl ohne Tropfenaustrag, es wird einfach nur die gewünschte Position mit dem xyz -Tisch angefahren ohne zu drucken.

Der G25-Befehl in 4.13 wird für den kontinuierlichen Tropfenaustrag verwendet. Dabei wird zunächst durch einen G01-Befehl eine bestimmte Position angefahren, danach können ein oder mehrere G25-Befehle für den kontinuierlichen Austrag von Tropfen folgen:

$$\begin{aligned} G01 \ Xx_0 \ Yy_0 \ Zz_0 \\ G25 \ Xx_1 \ Yy_1 \ Zz_1 \\ \vdots \\ G25 \ Xx_n \ Yy_n \ Zz_n \end{aligned} \quad (4.15)$$

Dabei gilt: $n \in \mathbb{N}$. Diese Befehlskette ist folgendermaßen zu interpretieren: $\forall i \in \{1, \dots, n\}$ werden auf der geraden Verbindungsstrecke zwischen $(x_{i-1}, y_{i-1}, z_{i-1})^T$ und $(x_i, y_i, z_i)^T$ automatisch Tropfen jeweils im Abstand der Tropfenbreite B gedruckt. Dabei wird der erste Tropfen auf dem geradlinigen Weg zum Punkt $(x_i, y_i, z_i)^T$ immer an der Position $(x_{i-1}, y_{i-1}, z_{i-1})^T$ gesetzt.

Der G26-Befehl in 4.14 ist eine Mischung von Positionier- und Druckbefehl. Man spricht dabei auch von diskontinuierlichem Austrag. Es wird zunächst die Position $(x, y, z)^T$ angefahren ohne auf dem Weg dorthin Material oder Tropfen auszutragen. Erst sobald die Position wirklich erreicht wurde, wird die Düse geöffnet und ein Tropfen gedruckt. Dies ist beim FDM-Verfahren so nicht möglich.

Diesem Prinzip wird sich beim tropfenweisen Drucken der Bauteilberandung innerhalb einer z -Schicht zunutze gemacht, wie in Abbildung 4.8 zu sehen ist. So werden auf den gestrichelten Berandungskonturen in der angegebenen Richtung automatisch Tropfen im Abstand der Tropfenbreite B positioniert, bis die Konturen geschlossen sind. Dabei ist die Reihenfolge der zu druckenden Konturen variabel, in der Regel wird jedoch die innere der äußeren Konturen vor der äußersten Kontur gedruckt, damit sich die äußerste Kontur, welche die eigentliche Bauteilumrandung darstellt, an die bereits vorhandene innere Kontur anschmiegen kann. Bei einer Schichthöhe von $H = 0.20\text{ mm}$ und einer Tropfenbreite von $B = 0.208\text{ mm}$ ist folgender Ausschnitt ein Beispiel für G-Code für den Druck der äußersten Kontur:

$$\begin{aligned}
 & \text{G26 X0.1040 Y19.9119 Z0.2000} \\
 & \text{G26 X0.1040 Y19.7039 Z0.2000} \\
 & \text{G26 X0.1040 Y19.4959 Z0.2000} \\
 & \quad \vdots \\
 & \text{G26 X0.2468 Y19.9357 Z0.2000} \\
 & \text{G26 X0.1040 Y19.9119 Z0.2000}
 \end{aligned} \tag{4.16}$$

Der erste und letzte Tropfen in (4.16) sind dabei identisch, um eventuelle Lücken in der Bauteilumrandung zu vermeiden und für einen stabilen Zusammenhalt der Konturen zu sorgen.

Nach dem Druck der Konturen wird in der Regel die Füllung gedruckt, wobei auch hier die Reihenfolge variierbar ist. Prozessbedingt ist es allerdings vorteilhafter, wenn sich die Füllung mit der bereits vorhandenen Bauteilkontur binden kann. Abbildung 4.8 zeigt auch ein Beispiel für den Druck einer Füllung in Form von Schraffurlinien. Dabei werden die Tropfen auf den Schraffurlinien automatisch kontinuierlich durch die Verwendung von G25-Befehlen gedruckt. Ein Beispielauszug für G-Code für den Druck der Füllung sieht folgendermaßen aus:

$$\begin{aligned}
 & \text{G01 X0.4160 Y49.2898 Z0.2000} \\
 & \text{G25 X0.7102 Y49.5840 Z0.2000} \\
 & \text{G01 X1.0043 Y49.5840 Z0.2000} \\
 & \text{G25 X0.4160 Y48.9957 Z0.2000} \\
 & \quad \vdots
 \end{aligned} \tag{4.17}$$

Dies ist der Beginn der Füllung links oben und stellt die ersten beiden Schraffurlinien dar. Es wird jeweils der Startpunkt einer Schraffurlinie durch einen G01-Befehl angefahren, danach werden bis zum Endpunkt der jeweiligen Schraffurlinie automatisch kontinuierlich Tropfen gesetzt durch Verwendung des G25-Befehls. Die Druckrichtung der Schraffurlinien ist in

der Grafik angegeben, ebenso wie die Leerfahrten ohne Tropfenaustrag zwischen den einzelnen Schraffurlinien. Leerfahrten gibt es dabei immer zwischen einem G25-Befehl gefolgt von einem G01-Befehl zum Anfahren des Startpunkts der nächsten Schraffurlinie. Die Schraffurlinien wurden dabei im Vorfeld der Maschinencode-Generierung bereits so geordnet, dass der Weg für Leerfahrten möglichst klein ist und somit möglichst schnell gedruckt werden kann.

Die hier verwendeten G-Code Befehle sind in dieser Weise prozessspezifisch und können nur vom verwendeten Tropfenerzeuger interpretiert werden. An dieser Stelle sei explizit darauf hingewiesen, dass sich der generierte Code in seinen Möglichkeiten deutlich vom FDM-Verfahren unterscheidet. So werden beim FDM-Verfahren Bauteile zwar auch in Form von Linienzügen gefüllt, jedoch können zwischen einzelnen Schraffurlinien nicht bewusst Leerfahrten eingefügt werden. Dies führt jedoch oftmals zu einem Aufstauen von Material beim Richtungswechsel und wirkt sich auf die Qualität der Teile aus. Wenngleich in diesem Ansatz noch keine kleineren Tropfen genutzt werden können, so gibt es doch bereits deutlich mehr Möglichkeiten zum gezielten Positionieren von Material als beim FDM.

5 Realisierung der Software-Module

In diesem Kapitel wird der mathematische Ansatz genau erläutert, welcher den implementierten Software-Modulen zugrunde liegt. Hierzu werden zunächst einige mathematische Definitionen und Grundlagen beschrieben, bevor es zur konkreten mathematischen Umsetzung der Problemstellung geht. Zu den Grundlagen gehören grundlegende mathematische Problemstellungen wie beispielsweise Abstandsberechnungen oder Schnittpunktberechnungen. Zum Abschluss wird auf die Feinheiten bei der Umsetzung in der Implementierung eingegangen.

5.1 Mathematische Definitionen

Definition 1 (Orthonormales Koordinatensystem):

Ein dreidimensionales orthonormales Koordinatensystem besteht aus drei zueinander senkrechten Vektoren der Länge 1. Analog besteht ein zweidimensionales orthonormales Koordinatensystem aus zwei zueinander senkrechten Vektoren der Länge 1. Die Position eines jeden Punktes im \mathbb{R}^3 bzw. \mathbb{R}^2 kann durch drei bzw. zwei Koordinatenwerte genau angegeben werden (siehe Abbildung 5.1).

Anmerkung: Alle in dieser Arbeit verwendeten Koordinatensysteme sind orthonormale Koordinatensysteme.

Definition 2 (Länge eines Vektors):

Die Länge eines Vektors $\mathbf{p} = (x_{\mathbf{p}}, y_{\mathbf{p}}, z_{\mathbf{p}})^T \in \mathbb{R}^3$ ist definiert durch:

$$|\mathbf{p}| := \sqrt{x_{\mathbf{p}}^2 + y_{\mathbf{p}}^2 + z_{\mathbf{p}}^2}. \quad (5.1)$$

Definition 3 (Skalarprodukt zweier Vektoren):

Das Skalarprodukt zweier Vektoren $\mathbf{a} \in \mathbb{R}^3$ und $\mathbf{b} \in \mathbb{R}^3$ ist definiert durch:

$$\mathbf{a} \circ \mathbf{b} := |\mathbf{a}| \cdot |\mathbf{b}| \cdot \cos \angle(\mathbf{a}, \mathbf{b}), \quad (5.2)$$

wobei $\angle(\mathbf{a}, \mathbf{b})$ den von \mathbf{a} und \mathbf{b} eingeschlossenen Winkel $\alpha \in \mathbb{R}$ ($0 \leq \alpha \leq \pi$) bezeichnet (siehe Abbildung 5.2).

Anmerkung: Aus der Definition folgt unmittelbar:

$$\alpha := \angle(\mathbf{a}, \mathbf{b}) = \frac{\pi}{2} \Rightarrow \mathbf{a} \circ \mathbf{b} = 0. \quad (5.3)$$

Definition 4 (Kreuzprodukt zweier Vektoren):

Das Kreuzprodukt zweier Vektoren $\mathbf{a} \in \mathbb{R}^3$ und $\mathbf{b} \in \mathbb{R}^3$ ist definiert durch:

$$\mathbf{a} \times \mathbf{b} := (|\mathbf{a}| \cdot |\mathbf{b}| \cdot \sin \alpha) \cdot \mathbf{n}, \quad (5.4)$$

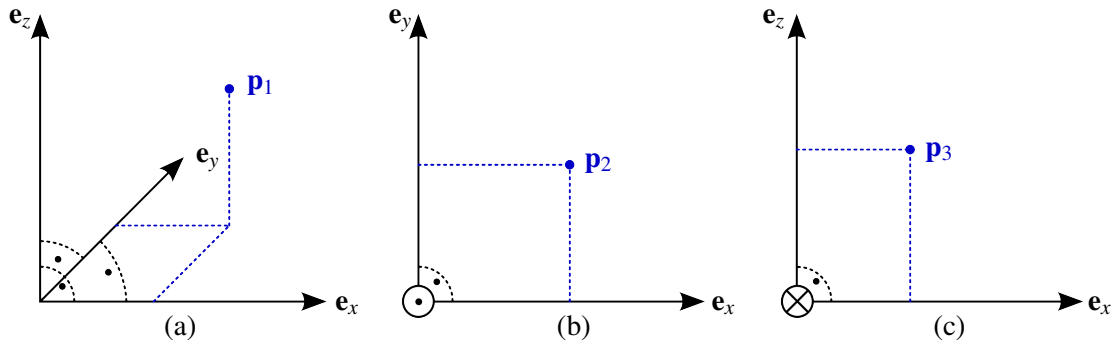


Abbildung 5.1: Orthonormale Koordinatensysteme: (a) Dreidimensionales Koordinatensystem. Alle Koordinatenachsen stehen senkrecht aufeinander. Punkt im Raum: $\mathbf{p}_1 \in \mathbb{R}^3$. (b) Punkt $\mathbf{p}_2 \in \mathbb{R}^3$ in der \mathbf{e}_x - \mathbf{e}_y -Ebene. \mathbf{e}_z -Achse schaut aus der Blattebene heraus (gekennzeichnet durch Kreis mit Punkt im Koordinatenursprung). (c) Punkt $\mathbf{p}_3 \in \mathbb{R}^3$ in der \mathbf{e}_x - \mathbf{e}_z -Ebene. \mathbf{e}_y -Achse schaut in die Blattebene hinein (gekennzeichnet durch Kreis mit Kreuz im Koordinatenursprung).

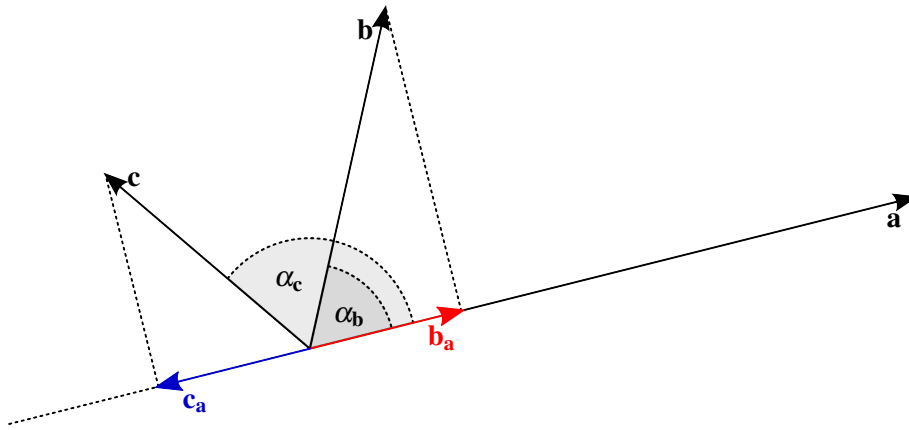


Abbildung 5.2: Veranschaulichung zum Skalarprodukt. \mathbf{b}_a ist die orthogonale Projektion von \mathbf{b} auf \mathbf{a} . \mathbf{c}_a ist die orthogonale Projektion von \mathbf{c} auf \mathbf{a} . Es gilt: $|\mathbf{b}_a| = |\mathbf{b}| \cdot \cos \alpha_b$ bzw. $|\mathbf{c}_a| = |\mathbf{c}| \cdot \cos(\pi - \alpha_c) = -|\mathbf{c}| \cdot \cos \alpha_c$, da \mathbf{c}_a und \mathbf{a} entgegengesetzt sind. Das heißt: $\mathbf{a} \circ \mathbf{b} = |\mathbf{a}| \cdot |\mathbf{b}_a|$ bzw. $\mathbf{a} \circ \mathbf{c} = -|\mathbf{a}| \cdot |\mathbf{c}_a|$.

wobei \mathbf{n} der auf der von \mathbf{a} und \mathbf{b} gebildeten Fläche senkrecht steht (siehe Abbildung 5.3). Die Vektoren \mathbf{a} , \mathbf{b} und $\mathbf{a} \times \mathbf{b}$ ($0 < \alpha < \pi$) spannen in dieser Reihenfolge ein Koordinatensystem nach der Rechte-Hand-Regel auf.

Anmerkung: Der Betrag $|\mathbf{a} \times \mathbf{b}|$ des Kreuzprodukts entspricht dem Flächeninhalt des von \mathbf{a} und \mathbf{b} aufgespannten Parallelogramms (siehe Abbildung 5.4): $|\mathbf{a} \times \mathbf{b}| = |\mathbf{a}| \cdot |\mathbf{b}| \cdot |\sin \alpha|$.

Definition 5 (Segment):

Seien $\mathbf{p}_0 \in \mathbb{R}^3$, $\mathbf{p}_1 \in \mathbb{R}^3$ ($\mathbf{p}_0 \neq \mathbf{p}_1$) und ferner $\mathbf{d} := \frac{\mathbf{p}_1 - \mathbf{p}_0}{|\mathbf{p}_1 - \mathbf{p}_0|} \in \mathbb{R}^3$ ($|\mathbf{d}| = 1$) der normierte Richtungsvektor von \mathbf{p}_0 nach \mathbf{p}_1 . Ein Segment ist definiert durch die Menge aller Punkte auf der geraden Verbindungsstrecke von \mathbf{p}_0 nach \mathbf{p}_1 :

$$\mathbf{S}(\mathbf{p}_0, \mathbf{p}_1) := \{ \mathbf{p}_0 + \mu \cdot \mathbf{d} \mid 0 \leq \mu \leq |\mathbf{p}_1 - \mathbf{p}_0|, \mu \in \mathbb{R} \} \quad (5.5)$$

(siehe Abbildung 5.5).

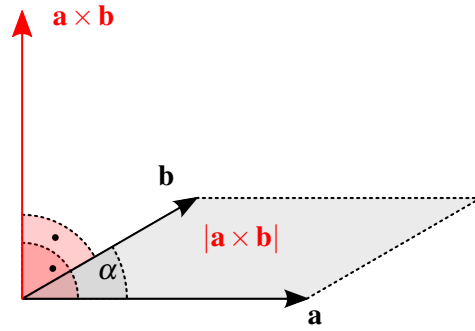


Abbildung 5.3: Das Kreuzprodukt der Vektoren \mathbf{a} und \mathbf{b} steht auf der von \mathbf{a} und \mathbf{b} gebildeten Fläche senkrecht. Der Betrag $|\mathbf{a} \times \mathbf{b}|$ des Kreuzprodukts entspricht, wie auch in Abbildung 5.4 zu sehen ist, dem Flächeninhalt des von \mathbf{a} und \mathbf{b} aufgespannten Parallelogramms.

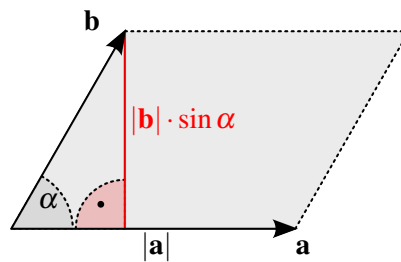


Abbildung 5.4: Der Betrag $|\mathbf{a} \times \mathbf{b}|$ des Kreuzprodukts entspricht dem Flächeninhalt des von \mathbf{a} und \mathbf{b} aufgespannten Parallelogramms: $|\mathbf{a} \times \mathbf{b}| = |\mathbf{a}| \cdot |\mathbf{b}| \cdot |\sin \alpha|$.

Definition 6 (Gerade):

Es gelten dieselben Voraussetzungen wie in Definition 5. Dann wird durch

$$\mathbf{G}(\mathbf{p}_0, \mathbf{p}_1) := \{ \mathbf{p}_0 + \mu \cdot \mathbf{d} \mid \mu \in \mathbb{R} \} \quad (5.6)$$

eine Gerade durch die Punkte \mathbf{p}_0 und \mathbf{p}_1 bestimmt.

Definition 7 (Ebene):

Seien $\mathbf{d}_{i_1} \in \mathbb{R}^3$ und $\mathbf{d}_{i_2} \in \mathbb{R}^3$ zwei linear unabhängige Vektoren, das heißt: $\mathbf{d}_{i_1} \neq \xi \cdot \mathbf{d}_{i_2} \forall \xi \in \mathbb{R}$. Sei weiter $\mathbf{p}_i \in \mathbb{R}^3$. Durch

$$\mathbf{E}(\mathbf{n}, \mathbf{p}_i) := \{ \mathbf{p}_i + \lambda \cdot \mathbf{d}_{i_1} + \mu \cdot \mathbf{d}_{i_2} \mid \lambda \in \mathbb{R}, \mu \in \mathbb{R} \} \quad (5.7)$$

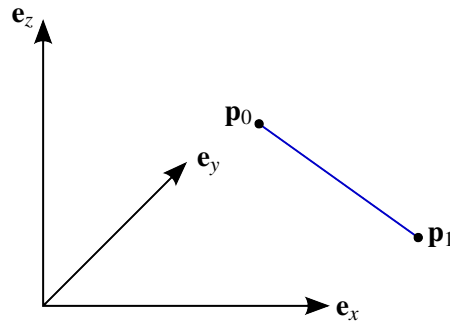
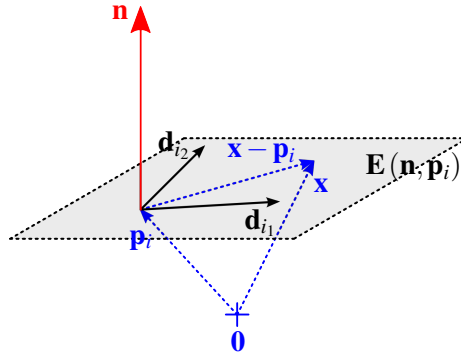
wird eine Ebene im \mathbb{R}^3 definiert (siehe Abbildung 5.6). Dabei ist durch $\mathbf{n} := \frac{\mathbf{d}_{i_1} \times \mathbf{d}_{i_2}}{|\mathbf{d}_{i_1} \times \mathbf{d}_{i_2}|}$ ein Normalenvektor definiert, der auf \mathbf{E} senkrecht steht. Die Hesse-Form von \mathbf{E} ist definiert durch:

$$\mathbf{E}_H(\mathbf{n}, \mathbf{p}_i) := \{ \mathbf{x} \mid \mathbf{x} \in \mathbb{R}^3, \mathbf{n} \circ (\mathbf{x} - \mathbf{p}_i) = 0 \}. \quad (5.8)$$

Definition 8 (Kreis / Kreisrand im \mathbb{R}^3):

Gegeben sei ein Punkt $\mathbf{p}_M = (x_{\mathbf{p}_M}, y_{\mathbf{p}_M})^T \in \mathbb{R}^3$ und $r \in \mathbb{R}$ ($r > 0$). Der Kreis mit dem Kreismittelpunkt \mathbf{p}_M und dem Radius r , auf dem der Normalenvektor $\mathbf{n} \in \mathbb{R}^3$ senkrecht steht, ist definiert als:

$$\mathbf{K}_{r, \mathbf{n}}(\mathbf{p}_M) := \{ \mathbf{p} \mid \mathbf{p} \in \mathbb{R}^3, (\mathbf{p} - \mathbf{p}_M) \circ \mathbf{n} = 0, |\mathbf{p} - \mathbf{p}_M| \leq r \} \quad (5.9)$$


 Abbildung 5.5: Gerades Segment von \mathbf{p}_0 bis \mathbf{p}_1 .

 Abbildung 5.6: Ebene \mathbf{E} nach Definition 7.

(siehe Abbildung 5.7). Der Kreisrand ist definiert als:

$$\partial \mathbf{K}_{r,\mathbf{n}}(\mathbf{p}_M) := \{ \mathbf{p} \mid \mathbf{p} \in \mathbb{R}^3, (\mathbf{p} - \mathbf{p}_M) \circ \mathbf{n} = 0, |\mathbf{p} - \mathbf{p}_M| = r \}. \quad (5.10)$$

Definition 9 (Kreissektor / Kreisbogen):

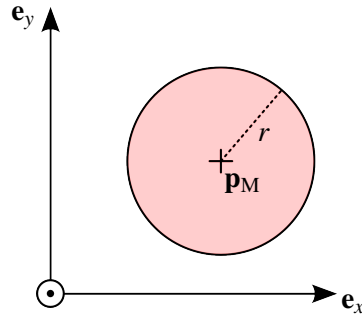
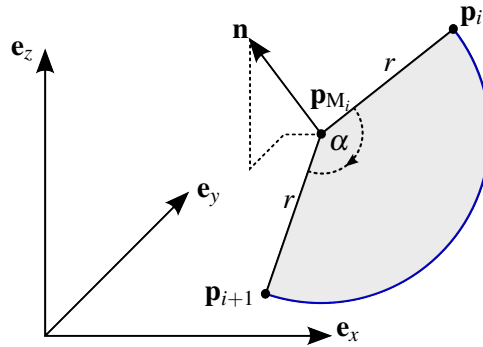
Seien $\mathbf{p}_{M_i} \in \mathbb{R}^3$, $\mathbf{p}_i \in \mathbb{R}^3$, $\mathbf{p}_{i+1} \in \mathbb{R}^3$ ($\mathbf{p}_i \neq \mathbf{p}_{i+1}$), $r \in \mathbb{R}$ ($r > 0$) und $\alpha \in \mathbb{R}$ ($0 < \alpha < \pi$). Weiter seien $\mathbf{K}_{r,\mathbf{n}}(\mathbf{p}_{M_i})$ ein Kreis und $\partial \mathbf{K}_{r,\mathbf{n}}(\mathbf{p}_{M_i})$ der dazugehörige Kreisrand nach Definition 8. Ferner seien $\mathbf{p}_i \in \partial \mathbf{K}_{r,\mathbf{n}}(\mathbf{p}_{M_i})$ und $\mathbf{p}_{i+1} \in \partial \mathbf{K}_{r,\mathbf{n}}(\mathbf{p}_{M_i})$. Durch die Punkte \mathbf{p}_{M_i} , \mathbf{p}_i und \mathbf{p}_{i+1} werden nach Abbildung 5.8 ein Kreissektor $\mathbf{K}_{\text{Sek}}(\mathbf{p}_i, \mathbf{p}_{i+1}) \subset \mathbf{K}_{r,\mathbf{n}}(\mathbf{p}_{M_i})$ und ein Kreisbogen $\mathbf{K}_{\text{Bog}}(\mathbf{p}_i, \mathbf{p}_{i+1}) \subset \partial \mathbf{K}_{r,\mathbf{n}}(\mathbf{p}_{M_i})$ definiert („ \subset “ bedeutet „echte Teilmenge von“). α ist der Mittelpunktswinkel des Kreissektors. Der Normalenvektor $\mathbf{n} \in \mathbb{R}^3$ bezüglich des Kreissektors ist definiert durch:

$$\mathbf{n} := \frac{(\mathbf{p}_{i+1} - \mathbf{p}_{M_i}) \times (\mathbf{p}_i - \mathbf{p}_{M_i})}{|(\mathbf{p}_{i+1} - \mathbf{p}_{M_i}) \times (\mathbf{p}_i - \mathbf{p}_{M_i})|}. \quad (5.11)$$

Anmerkung: Die Kreissektoren, zu denen die Kreisbögen gehören, haben in dieser Realisierung alle einen Mittelpunktswinkel $\alpha < 180^\circ$. Die in Kapitel 4 im eigenen Ansatz beschriebenen verwendeten Konturen bestehen alle aus Segmenten und Kreisbögen. Eine genauere Beschreibung der Konstruktion der Konturen folgt später.

Definition 10 (Element):

Der Begriff „Element“ oder „geometrisches Element“ wird im Folgenden als Überbegriff für Segmente nach Definition 5 und Kreisbögen nach Definition 9 verwendet, falls kein expliziter Unterschied zwischen Segmenten und Kreisbögen gemacht werden soll.


 Abbildung 5.7: Kreis mit Kreismittelpunkt \mathbf{p}_M und Radius r .

 Abbildung 5.8: Kreismittelpunkt: \mathbf{p}_{M_i} . \mathbf{n} ist der Normalenvektor bezüglich des von \mathbf{p}_{M_i} , \mathbf{p}_i und \mathbf{p}_{i+1} aufgespannten Kreissektors (graue Fläche) mit Radius r und dem Winkel α . Kreisbogen (blau) von \mathbf{p}_i nach \mathbf{p}_{i+1} . Für alle Punkte \mathbf{p} auf dem Kreisbogen gilt: $|\mathbf{p} - \mathbf{p}_{M_i}| = r$.

Definition 11 (Tangente):

Eine Tangente $\mathbf{t}(\mathbf{p})$ ist eine Gerade, die eine gegebene Kurve im Punkt \mathbf{p} berührt. In diesem Punkt \mathbf{p} haben sowohl die Tangente, als auch die Kurve, dieselbe Steigung (siehe Abbildung 5.9).

Anmerkung: Auf eine formale Definition einer Tangente wird an dieser Stelle bewusst verzichtet. Wichtig ist nur die bildliche Vorstellung, insbesondere muss eine Kurve im Punkt \mathbf{p} differenzierbar sein, damit man eindeutig eine Tangente durch den Punkt an die Kurve legen kann. Dies bedeutet bildlich, dass es im Punkt \mathbf{p} keinen „Knick“ geben darf.

Definition 12 (Konstruktion einer Kurve aus Segmenten und Kreisbögen):

Seien $\mathbf{p}'_{i-1} \in \mathbb{R}^3$, $\mathbf{p}'_i \in \mathbb{R}^3$, $\mathbf{p}'_{i+1} \in \mathbb{R}^3$, $\mathbf{p}'_{i+2} \in \mathbb{R}^3$, $\mathbf{p}_i \in \mathbb{R}^3$ und $\mathbf{p}_{i+1} \in \mathbb{R}^3$ Punkte ($i \in \mathbb{N}$), die in einer Ebene (Normalenvektor: $\mathbf{n} \in \mathbb{R}^3$) liegen. Seien ferner $\mathbf{S}(\mathbf{p}'_{i-1}, \mathbf{p}'_i)$ und $\mathbf{S}(\mathbf{p}'_{i+1}, \mathbf{p}'_{i+2})$ zwei Segmente nach Definition 5. Weiter sei $\mathbf{K}_{\text{Bog}}(\mathbf{p}_i, \mathbf{p}_{i+1}) \subset \partial \mathbf{K}_{r, \mathbf{n}}(\mathbf{p}_{M_i})$ ein Kreisbogen nach Definition 9, der Startpunkt des Kreisbogens ist also \mathbf{p}_i , der Endpunkt des Kreisbogens \mathbf{p}_{i+1} . Bei der Konstruktion einer Kurve müssen Segmente und Bögen richtig aneinandergereiht, d.h. verbunden, werden. Zwei Segmente $\mathbf{S}(\mathbf{p}'_{i-1}, \mathbf{p}'_i)$ und $\mathbf{S}(\mathbf{p}'_{i+1}, \mathbf{p}'_{i+2})$ sind verbunden, wenn $\mathbf{p}'_i = \mathbf{p}'_{i+1}$ gilt, das heißt, der Endpunkt des einen Segments muss mit dem Startpunkt des nächsten Segments übereinstimmen. Entsprechend werden ein Segment $\mathbf{S}(\mathbf{p}'_{i-1}, \mathbf{p}'_i)$ und ein Kreisbogen $\mathbf{K}_{\text{Bog}}(\mathbf{p}_i, \mathbf{p}_{i+1})$ verbunden, wenn $\mathbf{p}'_i = \mathbf{p}_i$ gilt und zusätzlich im Punkt \mathbf{p}_i eine eindeutige Tangente $\mathbf{t}(\mathbf{p}_i)$ nach Definition 11 existiert. Bei der Konstruktion einer Kurve können nie zwei Bögen miteinander verknüpft werden. Das heißt, der Endpunkt \mathbf{p}_{i+1} des Kreisbogens $\mathbf{K}_{\text{Bog}}(\mathbf{p}_i, \mathbf{p}_{i+1})$

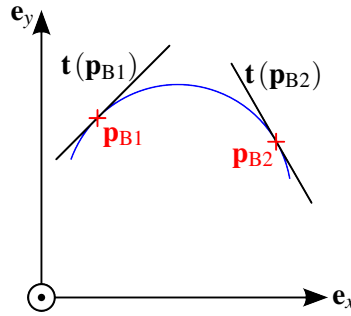


Abbildung 5.9: Tangenten $\mathbf{t}(\mathbf{p}_{B1})$ bzw. $\mathbf{t}(\mathbf{p}_{B2})$ an eine Kurve (blau) in den Berührungspunkten \mathbf{p}_{B1} bzw. \mathbf{p}_{B2} . Tangenten und die Kurve haben in den Berührungspunkten dieselbe Steigung.

kann auch nur mit einem Segment $\mathbf{S}(\mathbf{p}'_{i+1}, \mathbf{p}'_{i+2})$ verknüpft werden, falls $\mathbf{p}_{i+1} = \mathbf{p}'_{i+1}$ gilt und im Punkt \mathbf{p}_{i+1} wiederum eindeutig eine Tangente $\mathbf{t}(\mathbf{p}_{i+1})$ nach Definition 11 angelegt werden kann (siehe auch Abbildung 5.10).

Definition 13 (Kurve):

Seien $\mathbf{p}_0, \mathbf{p}_1, \dots, \mathbf{p}_n$ $n + 1$ Punkte im \mathbb{R}^3 ($n \in \mathbb{N}$) und $\mathbf{n} \in \mathbf{R}^3$ der Normalenvektor einer Ebene im \mathbb{R}^3 . Durch

$$\gamma_{\mathbf{n}} := \left\{ \bigcup_{i=0}^{n-1} \mathbf{E}(\mathbf{p}_i, \mathbf{p}_{i+1}) \mid \mathbf{E}(\mathbf{p}_i, \mathbf{p}_{i+1}) = \mathbf{S}(\mathbf{p}_i, \mathbf{p}_{i+1}) \vee \mathbf{E}(\mathbf{p}_i, \mathbf{p}_{i+1}) = \mathbf{K}_{\text{Bog}}(\mathbf{p}_i, \mathbf{p}_{i+1}) \right\} \quad (5.12)$$

wird eine Kurve zusammenhängender geometrischer Elemente definiert. Es gilt:

$$\mathbf{K}_{\text{Bog}}(\mathbf{p}_i, \mathbf{p}_{i+1}) \subset \partial \mathbf{K}_{r_i, \mathbf{n}}(\mathbf{p}_{M_i}), \quad (5.13)$$

das heißt, Kreisbögen können auch unterschiedliche Radien haben. Gilt für alle Kreisbögen der Kurve $r_i = r$ für ein festes $r > 0$ und genügt die Kurve allen Voraussetzungen in Definition 12, so spricht man bei $\gamma_{\mathbf{n}}$ von einer konstruierten Kurve.

Definition 14 (Geschlossene Kurve):

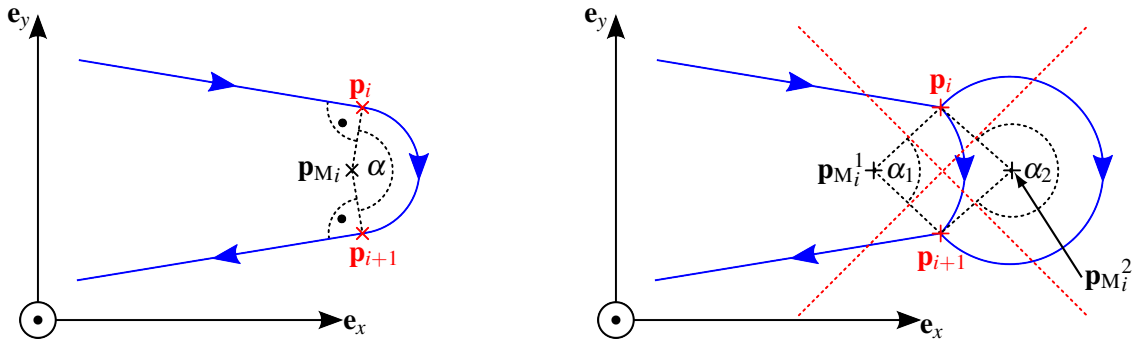
Eine geschlossene Kurve ist eine Kurve $\gamma_{\mathbf{n}}$ nach Definition 13, bei der zusätzlich gilt: $\mathbf{p}_0 = \mathbf{p}_n$.

Definition 15 (Richtung einer geschlossenen Kurve ohne Überschneidungen):

Gegeben sei der Normalenvektor \mathbf{n} einer beliebigen Ebene im \mathbb{R}^3 , in der sich eine Kurve befindet. Entscheidend für die Umlaufrichtung (im Uhrzeigersinn oder gegen den Uhrzeigersinn) einer geschlossenen Kurve ohne Überschneidungen ist, in welcher Richtung die einzelnen geometrischen Elemente beim Blick von der Spitze des Normalenvektors \mathbf{n} auf die Ebene eine geschlossene Kurve nach Definition 14 ohne Überschneidungen umlaufen. Abbildung 5.11 zeigt zwei Beispiele hierfür.

Definition 16 (Windungszahl):

Sei $\gamma_{\mathbf{n}}$ eine geschlossene Kurve nach Definition 14 und \mathbf{p} ein Punkt, der in der durch $\gamma_{\mathbf{n}}$ und \mathbf{n}



- (a) Der Übergang zwischen Segmenten und dem Kreisbogen (entspricht den Punkten \mathbf{p}_i und \mathbf{p}_{i+1}) muss immer „glatt“ sein, d.h. in den Übergangspunkten kann man eine eindeutige Tangente anlegen.
- (b) In den Punkten \mathbf{p}_i und \mathbf{p}_{i+1} (Übergang zwischen Segmenten und dem Kreisbogen) kann keine eindeutige Tangente an die Kurve angelegt werden. Dies ist so nicht erlaubt.

Abbildung 5.10: Beispiel und Gegenbeispiel für das Zusammenfügen von Elementen. Der Endpunkt eines Elements stimmt immer mit dem Startpunkt des nächsten Elements überein, unabhängig davon, ob es sich um ein Segment oder um einen Kreisbogen handelt. Kreisbögen haben als Vorgänger und Nachfolger immer ein Segment.

definierten Ebene liegt mit $\mathbf{p} \notin \gamma_n$. Die Windungszahl $\eta(\gamma_n, \mathbf{p})$ im Punkt \mathbf{p} ist definiert durch:

$$\eta(\gamma_n, \mathbf{p}) := \text{Anzahl der Umläufe von } \gamma_n \text{ um } \mathbf{p} \text{ gegen den Uhrzeigersinn} - \text{Anzahl der Umläufe von } \gamma_n \text{ um } \mathbf{p} \text{ im Uhrzeigersinn} \quad (5.14)$$

Anmerkung: Abbildung 5.12 zeigt einige Beispiele für den Wert $\eta(\gamma_n, \mathbf{p})$ der Windungszahl. Die Windungszahl kann nur ganzzahlige Werte annehmen. Die Definition der Windungszahl in dieser Form reicht für diese Arbeit völlig aus. Ursprünglich stammt der Begriff der Windungszahl allerdings aus der Funktionentheorie (siehe auch Bronstein et al. (2008)). Hier ist γ eine geschlossene Kurve in den komplexen Zahlen \mathbb{C} und $p_0 \in \mathbb{C}$ mit $p_0 \notin \gamma$. Die Windungszahl im Punkt p_0 ist dann definiert als:

$$\eta(\gamma, p_0) := \frac{1}{2\pi i} \int_{\gamma} \frac{1}{\xi - p_0} d\xi \in \mathbb{Z}. \quad (5.15)$$

Definition 17 (Charakteristische Funktion):

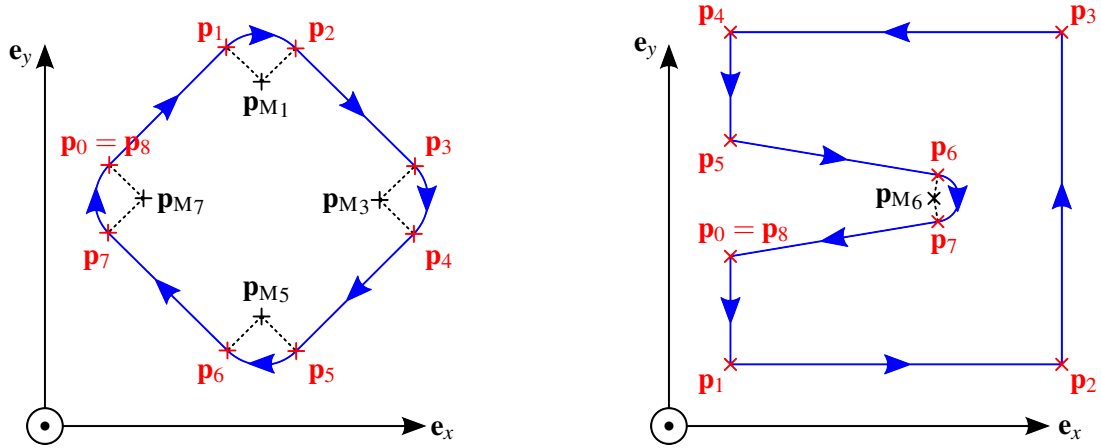
Die charakteristische Funktion ist im Folgenden für $i \in \mathbb{N}_0$ und $j \in \mathbb{N}_0$ wie folgt definiert:

$$\mathbb{1}_i(j) = \begin{cases} 1 & i = j \\ 0 & \text{sonst} \end{cases}. \quad (5.16)$$

Definition 18 (Rotationsmatrix):

Eine Rotationsmatrix $\mathbf{R} \in \mathbb{R}^{3 \times 3}$ ist eine Matrix der Form

$$\mathbf{R} := (\mathbf{e}_x \quad \mathbf{e}_y \quad \mathbf{e}_z) := \begin{pmatrix} x_{\mathbf{e}_x} & x_{\mathbf{e}_y} & x_{\mathbf{e}_z} \\ y_{\mathbf{e}_x} & y_{\mathbf{e}_y} & y_{\mathbf{e}_z} \\ z_{\mathbf{e}_x} & z_{\mathbf{e}_y} & z_{\mathbf{e}_z} \end{pmatrix}, \quad (5.17)$$



(a) Beispiel für eine geschlossene Kurve im Uhrzeigersinn. (b) Beispiel für eine geschlossene Kurve gegen den Uhrzeigersinn.

Abbildung 5.11: Zwei Beispiele für geschlossene Kurven aus Segmenten und Kreisbögen. Dabei ist das Prinzip aus Abbildung 5.10 angewandt.

wobei die Spaltenvektoren der Rotationsmatrix orthonormal sind, das heißt, sie haben die Länge 1 und stehen senkrecht aufeinander:

$$|\mathbf{e}_x| = 1, |\mathbf{e}_y| = 1, |\mathbf{e}_z| = 1, \quad (5.18)$$

$$\mathbf{e}_x = \mathbf{e}_y \times \mathbf{e}_z, \mathbf{e}_y = \mathbf{e}_z \times \mathbf{e}_x, \mathbf{e}_z = \mathbf{e}_x \times \mathbf{e}_y. \quad (5.19)$$

Man sagt auch, die Vektoren \mathbf{e}_x , \mathbf{e}_y und \mathbf{e}_z bilden eine orthonormale Basis des orthonormalen Koordinatensystems.

Definition 19 (Orthonormale Basis des \mathbb{R}^3):

Die Vektoren \mathbf{e}_x , \mathbf{e}_y und \mathbf{e}_z der Rotationsmatrix \mathbf{R} aus Definition 18 bilden eine orthonormale Basis des \mathbb{R}^3 . Jeder Punkt $\mathbf{v} \in \mathbb{R}^3$ lässt sich darstellen als eine Linearkombination dieser drei Basisvektoren:

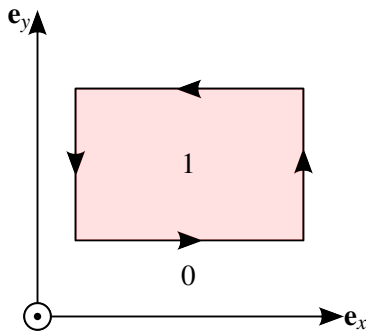
$$\mathbf{v} = x_v \cdot \mathbf{e}_x + y_v \cdot \mathbf{e}_y + z_v \cdot \mathbf{e}_z. \quad (5.20)$$

Man sagt auch, x_v , y_v und z_v sind die Koordinaten des Punkts \mathbf{v} bezüglich der Basis $\{\mathbf{e}_x, \mathbf{e}_y, \mathbf{e}_z\}$.

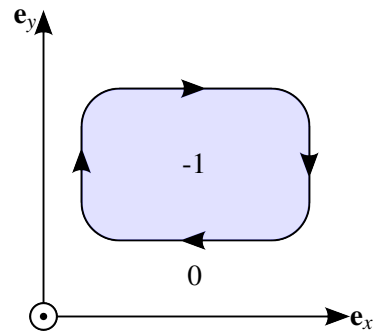
Definition 20 (Rotation eines Koordinatensystems um eine Drehachse / Drehrichtung):

Bei der Rotation eines Ursprungs koordinatensystems U um den Winkel φ um eine Drehachse $\mathbf{v} \in \mathbb{R}^3$ in ein Zielkoordinatensystem Z spricht man von einer Drehung im mathematisch positiven Sinn ($\varphi > 0$) genau dann, wenn Z aus einer Drehung um \mathbf{v} gegen den Uhrzeigersinn aus U hervorgeht. Von einer Drehung im mathematisch negativen Sinn ($\varphi < 0$) spricht man, wenn Z aus einer Drehung um \mathbf{v} im Uhrzeigersinn aus U hervorgeht.

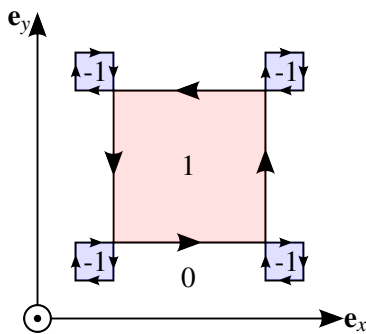
Anmerkung: Abbildung 5.13 zeigt ein Einheitskoordinatensystem. Hier ist auch die Drehrichtung im mathematisch positiven Sinn gegen den Uhrzeigersinn für die drei Koordinatenachsen (entspricht den Drehachsen) \mathbf{e}_x , \mathbf{e}_y und \mathbf{e}_z angegeben.



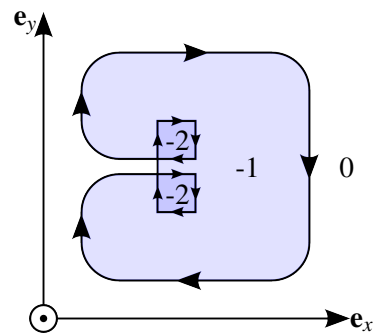
(a) Windungszahl für geschlossene Kurve ohne Überschneidungen gegen den Uhrzeigersinn.



(b) Windungszahl für geschlossene Kurve ohne Überschneidungen im Uhrzeigersinn.



(c) Windungszahl für geschlossene Kurve mit Überschneidungen: Beispiel 1.



(d) Windungszahl für geschlossene Kurve mit Überschneidungen: Beispiel 2.

Abbildung 5.12: Beispiele für Windungszahlen (siehe Definition 16) geschlossener Kurven γ_{e_z} nach Definition 14 in einem Punkt $\mathbf{p} \notin \gamma_{e_z}$.

Definition 21 (Homogene Transformationsmatrix):

Eine homogene Transformationsmatrix $\mathbf{T} \in \mathbb{R}^{4 \times 4}$ ist eine Matrix der Form

$$\mathbf{T} := \begin{pmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0} & 1 \end{pmatrix}, \tag{5.21}$$

wobei $\mathbf{R} \in \mathbb{R}^{3 \times 3}$ eine Rotationsmatrix nach Definition 18 und $\mathbf{t} \in \mathbb{R}^3$ ein Translationsvektor ist. Seien weiter U das Ursprungskordinatensystem und Z das Zielkoordinatensystem, dann beschreibt die Matrix

$${}^U\mathbf{T}_Z := \begin{pmatrix} {}^U\mathbf{R}_Z & {}^U\mathbf{t}_Z \\ \mathbf{0} & 1 \end{pmatrix} \tag{5.22}$$

die Lage (Orientierung und Position) des Zielkoordinatensystems Z im Ursprungskordinatensystem U .

5.2 Mathematische Grundlagen

Satz 1 (Skalarprodukt für orthonormale Koordinatensysteme):

Das Skalarprodukt zweier Vektoren $\mathbf{a} \in \mathbb{R}^3$ und $\mathbf{b} \in \mathbb{R}^3$ lässt sich in orthonormalen Koordinatensystemen berechnen durch

$$\mathbf{a} \circ \mathbf{b} = x_{\mathbf{a}} \cdot x_{\mathbf{b}} + y_{\mathbf{a}} \cdot y_{\mathbf{b}} + z_{\mathbf{a}} \cdot z_{\mathbf{b}}. \tag{5.23}$$

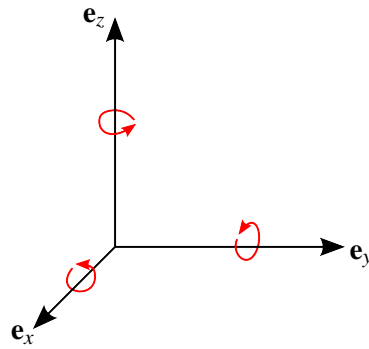


Abbildung 5.13: Ursprungs Koordinatensystem bestehend aus den Einheitsvektoren $\mathbf{e}_x = (1, 0, 0)^T$, $\mathbf{e}_y = (0, 1, 0)^T$ und $\mathbf{e}_z = (0, 0, 1)^T$. Es gilt: $\mathbf{R} = \mathbf{I} = (\mathbf{e}_x, \mathbf{e}_y, \mathbf{e}_z)$. Die Drehrichtung im mathematisch positiven Sinn gegen den Uhrzeigersinn ist für die drei Koordinatenachsen in roter Farbe angegeben.

Des Weiteren kann der Winkel α zwischen \mathbf{a} und \mathbf{b} berechnet werden durch:

$$\alpha = \cos^{-1} \left(\frac{\mathbf{a} \circ \mathbf{b}}{|\mathbf{a}| \cdot |\mathbf{b}|} \right), \quad (5.24)$$

wobei $0 \leq \alpha \leq \pi$ gilt.

Beweis: Nach Definition 18 bilden die Vektoren \mathbf{e}_x , \mathbf{e}_y und \mathbf{e}_z eine orthonormale Basis des Koordinatensystems. Mit

$$\mathbf{a} = x_{\mathbf{a}} \cdot \mathbf{e}_x + y_{\mathbf{a}} \cdot \mathbf{e}_y + z_{\mathbf{a}} \cdot \mathbf{e}_z \quad \text{und} \quad \mathbf{b} = x_{\mathbf{b}} \cdot \mathbf{e}_x + y_{\mathbf{b}} \cdot \mathbf{e}_y + z_{\mathbf{b}} \cdot \mathbf{e}_z \quad (5.25)$$

gilt somit nach Definition 3 des Skalarprodukts:

$$\begin{aligned} \mathbf{a} \circ \mathbf{b} &= x_{\mathbf{a}} \cdot x_{\mathbf{b}} \cdot \underbrace{(\mathbf{e}_x \circ \mathbf{e}_x)}_{=1} + x_{\mathbf{a}} \cdot y_{\mathbf{b}} \cdot \underbrace{(\mathbf{e}_x \circ \mathbf{e}_y)}_{=0} + x_{\mathbf{a}} \cdot z_{\mathbf{b}} \cdot \underbrace{(\mathbf{e}_x \circ \mathbf{e}_z)}_{=0} + \\ & y_{\mathbf{a}} \cdot x_{\mathbf{b}} \cdot \underbrace{(\mathbf{e}_y \circ \mathbf{e}_x)}_{=0} + y_{\mathbf{a}} \cdot y_{\mathbf{b}} \cdot \underbrace{(\mathbf{e}_y \circ \mathbf{e}_y)}_{=1} + y_{\mathbf{a}} \cdot z_{\mathbf{b}} \cdot \underbrace{(\mathbf{e}_y \circ \mathbf{e}_z)}_{=0} + \\ & z_{\mathbf{a}} \cdot x_{\mathbf{b}} \cdot \underbrace{(\mathbf{e}_z \circ \mathbf{e}_x)}_{=0} + z_{\mathbf{a}} \cdot y_{\mathbf{b}} \cdot \underbrace{(\mathbf{e}_z \circ \mathbf{e}_y)}_{=0} + z_{\mathbf{a}} \cdot z_{\mathbf{b}} \cdot \underbrace{(\mathbf{e}_z \circ \mathbf{e}_z)}_{=1} \\ &= x_{\mathbf{a}} \cdot x_{\mathbf{b}} + y_{\mathbf{a}} \cdot y_{\mathbf{b}} + z_{\mathbf{a}} \cdot z_{\mathbf{b}} \end{aligned} \quad (5.26)$$

Die Formel

$$\alpha = \cos^{-1} \left(\frac{\mathbf{a} \circ \mathbf{b}}{|\mathbf{a}| \cdot |\mathbf{b}|} \right) \quad (5.27)$$

für $0 \leq \alpha \leq \pi$ folgt dann unmittelbar aus der Definition 3 des Skalarprodukts. q.e.d.

Satz 2 (Kreuzprodukt zweier Vektoren):

Für das Kreuzprodukt zweier Vektoren \mathbf{a} und \mathbf{b} gilt:

$$\mathbf{a} \times \mathbf{b} = \begin{pmatrix} x_{\mathbf{a}} \\ y_{\mathbf{a}} \\ z_{\mathbf{a}} \end{pmatrix} \times \begin{pmatrix} x_{\mathbf{b}} \\ y_{\mathbf{b}} \\ z_{\mathbf{b}} \end{pmatrix} = \begin{pmatrix} y_{\mathbf{a}} \cdot z_{\mathbf{b}} - y_{\mathbf{b}} \cdot z_{\mathbf{a}} \\ -x_{\mathbf{a}} \cdot z_{\mathbf{b}} + x_{\mathbf{b}} \cdot z_{\mathbf{a}} \\ x_{\mathbf{a}} \cdot y_{\mathbf{b}} - x_{\mathbf{b}} \cdot y_{\mathbf{a}} \end{pmatrix}. \quad (5.28)$$

Beweis: Ähnlich wie Beweis zu Satz 1.

q.e.d.

Satz 3 (Determinante einer Matrix):

Gegeben sei eine Matrix $\mathbf{A} \in \mathbb{R}^3$:

$$\mathbf{A} = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix}. \quad (5.29)$$

Für die Determinante D der Matrix \mathbf{A} gilt:

$$D := \begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix} = a_{11} \cdot a_{22} \cdot a_{33} + a_{12} \cdot a_{23} \cdot a_{31} + a_{13} \cdot a_{21} \cdot a_{32} - a_{31} \cdot a_{22} \cdot a_{13} - a_{32} \cdot a_{23} \cdot a_{11} - a_{33} \cdot a_{21} \cdot a_{12}. \quad (5.30)$$

Beweis: Siehe Bronstein et al. (2008).

q.e.d.

Satz 4 (Lösung linearer Gleichungssysteme):

Gegeben sei wie in Satz 3 eine Matrix $\mathbf{A} \in \mathbb{R}^{3 \times 3}$. Des Weiteren sei $\mathbf{b} = (b_1 \ b_2 \ b_3)^T \in \mathbb{R}^3$ ein gegebener Vektor und $\mathbf{x} = (x_1 \ x_2 \ x_3)^T \in \mathbb{R}^3$ der Vektor der Unbekannten. Das lineare Gleichungssystem:

$$\mathbf{A}\mathbf{x} = \mathbf{b} \quad (5.31)$$

hat genau dann eine einzige Lösung \mathbf{x} wenn für die Determinante D von \mathbf{A} aus Satz 3

$$D \neq 0 \quad (5.32)$$

gilt. Mit den Determinanten

$$D_{x_1} := \begin{pmatrix} b_1 & a_{12} & a_{13} \\ b_2 & a_{22} & a_{23} \\ b_3 & a_{32} & a_{33} \end{pmatrix}, \quad D_{x_2} := \begin{pmatrix} a_{11} & b_1 & a_{13} \\ a_{21} & b_2 & a_{23} \\ a_{31} & b_3 & a_{33} \end{pmatrix}, \quad D_{x_3} := \begin{pmatrix} a_{11} & a_{12} & b_1 \\ a_{21} & a_{22} & b_2 \\ a_{31} & a_{32} & b_3 \end{pmatrix} \quad (5.33)$$

gilt für die Lösung \mathbf{x} :

$$x_1 = \frac{D_{x_1}}{D}, \quad x_2 = \frac{D_{x_2}}{D}, \quad x_3 = \frac{D_{x_3}}{D}. \quad (5.34)$$

Beweis: Siehe Bronstein et al. (2008).

q.e.d.

Satz 5 (Bestimmung einer Rotationsmatrix):

Seien U ein Ursprungskoordinatensystem und Z ein Zielkoordinatensystem mit demselben Koordinatenursprung. Seien des Weiteren $\{\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3\}$ eine orthonormale Basis von U und $\{\mathbf{z}_1, \mathbf{z}_2, \mathbf{z}_3\}$ eine orthonormale Basis von Z . Durch

$$\mathbf{z}_j = \sum_{i=1}^3 r_{ij} \mathbf{u}_i \quad \forall j \in \{1, 2, 3\} \quad (5.35)$$

wird eine Rotationsmatrix

$${}^U\mathbf{R}_Z := (\mathbf{r}_1 \quad \mathbf{r}_2 \quad \mathbf{r}_3) = \begin{pmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{pmatrix} \in \mathbb{R}^{3 \times 3} \quad (5.36)$$

nach Definition 18 bestimmt.

Beweis: Nach Voraussetzung gilt $\forall j_1 \in \{1, 2, 3\}$ und $\forall j_2 \in \{1, 2, 3\}$:

$$\begin{aligned} \mathbb{1}_{j_1}(j_2) &= \mathbf{z}_{j_1} \circ \mathbf{z}_{j_2} = \left(\sum_{i=1}^3 r_{ij_1} \mathbf{u}_i \right) \circ \left(\sum_{i=1}^3 r_{ij_2} \mathbf{u}_i \right) \\ &= (r_{1j_1} \mathbf{u}_1 + r_{2j_1} \mathbf{u}_2 + r_{3j_1} \mathbf{u}_3) \circ (r_{1j_2} \mathbf{u}_1 + r_{2j_2} \mathbf{u}_2 + r_{3j_2} \mathbf{u}_3) \\ &= \left(\begin{array}{l} r_{1j_1} r_{1j_2} \underbrace{\mathbf{u}_1 \circ \mathbf{u}_1}_{=1} + r_{1j_1} r_{2j_2} \underbrace{\mathbf{u}_1 \circ \mathbf{u}_2}_{=0} + r_{1j_1} r_{3j_2} \underbrace{\mathbf{u}_1 \circ \mathbf{u}_3}_{=0} + \\ r_{2j_1} r_{1j_2} \underbrace{\mathbf{u}_2 \circ \mathbf{u}_1}_{=0} + r_{2j_1} r_{2j_2} \underbrace{\mathbf{u}_2 \circ \mathbf{u}_2}_{=1} + r_{2j_1} r_{3j_2} \underbrace{\mathbf{u}_2 \circ \mathbf{u}_3}_{=0} + \\ r_{3j_1} r_{1j_2} \underbrace{\mathbf{u}_3 \circ \mathbf{u}_1}_{=0} + r_{3j_1} r_{2j_2} \underbrace{\mathbf{u}_3 \circ \mathbf{u}_2}_{=0} + r_{3j_1} r_{3j_2} \underbrace{\mathbf{u}_3 \circ \mathbf{u}_3}_{=1} \end{array} \right) \\ &= \mathbf{r}_{j_1} \circ \mathbf{r}_{j_2} \end{aligned} \quad (5.37)$$

q.e.d.

Satz 6 (Multiplikation von Rotationsmatrix und Punkt):

Seien U ein Ursprungskoordinatensystem und Z ein Zielkoordinatensystem mit demselben Koordinatenursprung, das heißt, die Rotation von U nach Z kann beschrieben werden durch eine Rotationsmatrix ${}^U\mathbf{R}_Z$. Die Koordinaten eines Punktes ${}^Z\mathbf{p} \in \mathbb{R}^3$ im Zielkoordinatensystem können in die Koordinaten eines Punktes ${}^U\mathbf{p} \in \mathbb{R}^3$ im Ursprungskoordinatensystem umgerechnet werden durch:

$${}^U\mathbf{p} = {}^U\mathbf{R}_Z {}^Z\mathbf{p}. \quad (5.38)$$

Es handelt sich dabei um denselben Punkt, allerdings sind die Koordinaten des Punkts in unterschiedlichen Koordinatensystemen angegeben.

Beweis: Nach Satz 5 gilt zunächst:

$$\mathbf{z}_j = \sum_{i=1}^3 r_{ij} \mathbf{u}_i \quad \forall j \in \{1, 2, 3\}. \quad (5.39)$$

Seien v_1, v_2 und v_3 die Koordinaten eines Vektors $\mathbf{v} \in \mathbb{R}^3$ bezüglich der Basis von Z . Mit (5.39) gilt also:

$$\begin{aligned} \mathbf{v} &= \sum_{j=1}^3 v_j \mathbf{z}_j = \sum_{j=1}^3 \left(\sum_{i=1}^3 r_{ij} \mathbf{u}_i \right) = \sum_{i=1}^3 v_j \left(\sum_{j=1}^3 r_{ij} \mathbf{u}_i \right) \\ &= \sum_{i=1}^3 \underbrace{\left(\sum_{j=1}^3 r_{ij} v_j \right)}_{w_i :=} \mathbf{u}_i \end{aligned} \quad (5.40)$$

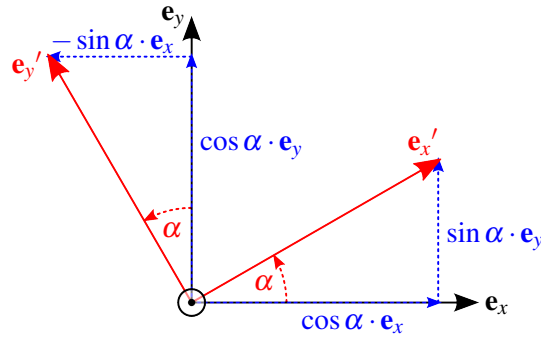


Abbildung 5.14: Beispiel für die Drehung eines Ursprungs Koordinatensystem U (schwarz) um den Winkel α in ein Zielkoordinatensystem Z (rot).

Das heißt also, sind v_1, v_2 und v_3 nach Definition 19 die Koordinaten des Punktes ${}^Z\mathbf{p}$ bezüglich der Basis von Z , dann sind

$$\begin{pmatrix} w_1 \\ w_2 \\ w_3 \end{pmatrix} = \mathbf{R} \begin{pmatrix} v_1 \\ v_2 \\ v_3 \end{pmatrix}, \quad (5.41)$$

die Koordinaten des Punktes ${}^U\mathbf{p}$ bezüglich der Basis von U . q.e.d.

Satz 7 (Rotation um die Koordinatenachsen):

Die Rotationsmatrizen für die Drehung um die ursprünglichen Koordinatenachsen $\mathbf{e}_x, \mathbf{e}_y$ und \mathbf{e}_z in ein Zielkoordinatensystem sind gegeben durch:

$$\mathbf{R}_{\varphi_x} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \varphi_x & -\sin \varphi_x \\ 0 & \sin \varphi_x & \cos \varphi_x \end{pmatrix}, \quad (5.42)$$

$$\mathbf{R}_{\varphi_y} = \begin{pmatrix} \cos \varphi_y & 0 & \sin \varphi_y \\ 0 & 1 & 0 \\ -\sin \varphi_y & 0 & \cos \varphi_y \end{pmatrix}, \quad (5.43)$$

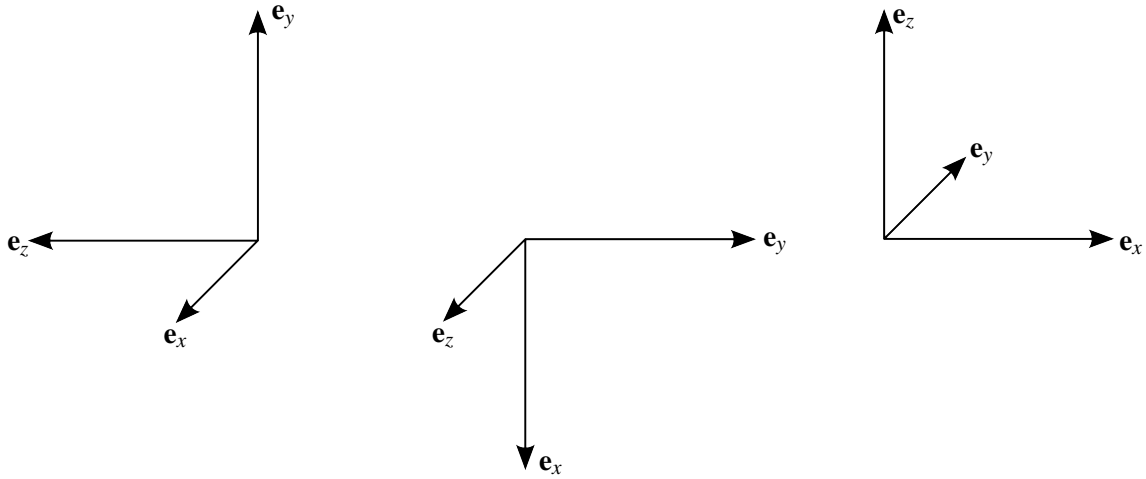
$$\mathbf{R}_{\varphi_z} = \begin{pmatrix} \cos \varphi_z & -\sin \varphi_z & 0 \\ \sin \varphi_z & \cos \varphi_z & 0 \\ 0 & 0 & 1 \end{pmatrix}. \quad (5.44)$$

Beweis: Mit Hilfe von Satz 5. Eine grafische Herleitung für eine Drehung um φ_z ist in Abbildung 5.14 zu sehen. Die weiteren Herleitungen sind analog dazu. q.e.d.

Beispiel: Die Rotationsmatrizen für die Drehungen um $\varphi_x = 90^\circ, \varphi_y = 90^\circ$ und $\varphi_z = 90^\circ$ sind gegeben durch:

$$\mathbf{R}_{\varphi_x=90^\circ} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{pmatrix}, \quad \mathbf{R}_{\varphi_y=90^\circ} = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ -1 & 0 & 0 \end{pmatrix}, \quad \mathbf{R}_{\varphi_z=90^\circ} = \begin{pmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (5.45)$$

Abbildung 5.15 zeigt die entsprechenden Rotation des Ursprungs Koordinatensystems in Abbildung 5.13 um $\varphi_x = 90^\circ, \varphi_y = 90^\circ$ und $\varphi_z = 90^\circ$.



(a) Drehung um die e_x -Achse um den Winkel $\phi_x (= 90^\circ)$. (b) Drehung um die e_y -Achse um den Winkel $\phi_y (= 90^\circ)$. (c) Drehung um die e_z -Achse um den Winkel $\phi_z (= 90^\circ)$.

Abbildung 5.15: Ursprungskoordinatensystem aus Abbildung 5.13: Durch Drehung um die Koordinatenachsen des Ursprungs koordinatensystems entstehen neue orthonormale (Ziel-)Koordinatensysteme.

Satz 8 (Verkettung von Rotation und Translation):

Es gelten dieselben Voraussetzungen wie in Satz 6, das heißt, es gilt:

$${}^U \mathbf{p} = {}^U \mathbf{R}_Z {}^Z \mathbf{p}. \quad (5.46)$$

Sei weiter ${}^U \mathbf{t}_Z$ ein Translationsvektor um den der Ursprung des Zielkoordinatensystems Z gegenüber dem der Ursprung des Ursprungs koordinatensystems U verschoben ist (siehe Abbildung 5.16). Dann gilt:

$${}^U \mathbf{p} = {}^U \mathbf{R}_Z {}^Z \mathbf{p} + {}^U \mathbf{t}_Z. \quad (5.47)$$

Des Weiteren gilt folgende Äquivalenz:

$${}^U \mathbf{p} = {}^U \mathbf{R}_Z {}^Z \mathbf{p} + {}^U \mathbf{t}_Z \Leftrightarrow \begin{pmatrix} {}^U \mathbf{p} \\ 1 \end{pmatrix} = \begin{pmatrix} {}^U \mathbf{R}_Z & {}^U \mathbf{t}_Z \\ \mathbf{0} & 1 \end{pmatrix} \begin{pmatrix} {}^Z \mathbf{p} \\ 1 \end{pmatrix} = {}^U \mathbf{T}_Z \begin{pmatrix} {}^Z \mathbf{p} \\ 1 \end{pmatrix}. \quad (5.48)$$

Beweis: Durch einfaches Nachrechnen. q.e.d.

Satz 9 (Inverse einer homogenen Transformationsmatrix):

Die Inverse einer Rotationsmatrix \mathbf{R} nach Definition 18 ist gegeben durch die Transponierte von \mathbf{R} :

$$\mathbf{R}^{-1} = \mathbf{R}^T. \quad (5.49)$$

Die Inverse einer homogenen Transformationsmatrix ${}^U \mathbf{T}_Z$ nach Definition 21 ist gegeben durch:

$${}^Z \mathbf{T}_U := {}^U \mathbf{T}_Z^{-1} = \begin{pmatrix} {}^U \mathbf{R}_Z^T & -{}^U \mathbf{R}_Z^T \cdot {}^U \mathbf{t}_Z \\ \mathbf{0} & 1 \end{pmatrix} \quad (5.50)$$

Beweis: Durch einfaches Nachrechnen. q.e.d.

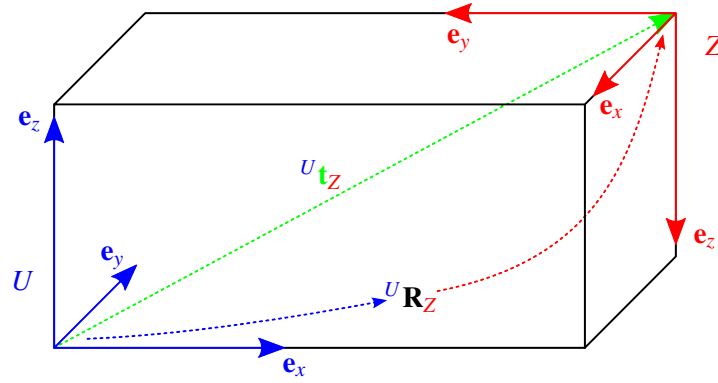


Abbildung 5.16: Beispiel für eine homogene Transformation bestehend aus einer Rotation ${}^U\mathbf{R}_Z$ vom Ursprungskordinatensystem U ins Zielkoordinatensystem Z und einer Translation ${}^U\mathbf{t}_Z$ des Ursprungs von Z gegenüber dem Ursprung von U .

Anmerkung: Homogene Transformationsmatrizen eignen sich insbesondere als kompakte Notation für Berechnungen von Transformationen.

Satz 10 (Verkettung mehrerer Transformation):

Seien A_1, A_2, \dots, A_n n verschiedene Koordinatensysteme. Des Weiteren seien ${}^{A_i}\mathbf{T}_{A_{i+1}}$ ($1 \leq i < n$) die entsprechenden Transformationsmatrizen, dann gilt:

$${}^{A_1}\mathbf{T}_{A_n} = {}^{A_1}\mathbf{T}_{A_2} \cdot \dots \cdot {}^{A_{n-1}}\mathbf{T}_{A_n}. \quad (5.51)$$

Entsprechend gilt für die Inverse von ${}^{A_1}\mathbf{T}_{A_n}$:

$${}^{A_1}\mathbf{T}_{A_n}^{-1} = \left({}^{A_1}\mathbf{T}_{A_2} \cdot \dots \cdot {}^{A_{n-1}}\mathbf{T}_{A_n} \right)^{-1} = {}^{A_n}\mathbf{T}_{A_{n-1}} \cdot \dots \cdot {}^{A_2}\mathbf{T}_{A_1} = {}^{A_n}\mathbf{T}_{A_1} \quad (5.52)$$

Beweis: Klar durch Satz 9.

q.e.d.

Satz 11 (Drehung eines Koordinatensystems um einen Vektor \mathbf{v}):

Die Drehung eines Koordinatensystems vom Ursprungskordinatensystem U ins Zielkoordinatensystem Z um einen Vektor \mathbf{v} kann beschrieben werden durch die Rotationsmatrix

$${}^U\mathbf{R}_Z(\mathbf{v}, \alpha) = \begin{pmatrix} x_{\mathbf{v}}^2(1 - \cos \alpha) + \cos \alpha & x_{\mathbf{v}}y_{\mathbf{v}}(1 - \cos \alpha) - z_{\mathbf{v}} \sin \alpha & x_{\mathbf{v}}z_{\mathbf{v}}(1 - \cos \alpha) + y_{\mathbf{v}} \sin \alpha \\ y_{\mathbf{v}}x_{\mathbf{v}}(1 - \cos \alpha) + z_{\mathbf{v}} \sin \alpha & y_{\mathbf{v}}^2(1 - \cos \alpha) + \cos \alpha & y_{\mathbf{v}}z_{\mathbf{v}}(1 - \cos \alpha) - x_{\mathbf{v}} \sin \alpha \\ z_{\mathbf{v}}x_{\mathbf{v}}(1 - \cos \alpha) - y_{\mathbf{v}} \sin \alpha & z_{\mathbf{v}}y_{\mathbf{v}}(1 - \cos \alpha) + x_{\mathbf{v}} \sin \alpha & z_{\mathbf{v}}^2(1 - \cos \alpha) + \cos \alpha \end{pmatrix}, \quad (5.53)$$

wobei α der Drehwinkel um \mathbf{v} im mathematisch positiven Sinn (gegen den Uhrzeigersinn) ist. Entsprechend gilt für die homogene Transformationsmatrix

$${}^U\mathbf{T}_Z(\mathbf{v}, \alpha) = \begin{pmatrix} {}^U\mathbf{R}_Z(\mathbf{v}, \alpha) & {}^U\mathbf{t}_Z \\ \mathbf{0} & 1 \end{pmatrix}, \quad (5.54)$$

wobei ${}^U\mathbf{t}_Z = (0, 0, 0)^T$, falls es sich nur um eine Drehung ohne Translation handelt.

Beweis: Für einen Beweis der vollständigen Richtigkeit dieser Rotationsmatrix sei auf Bronstein et al. (2008) verwiesen. q.e.d.

Möchte man einen Körper im \mathbb{R}^3 drehen, so kommt es bei der Drehung um mehrere Drehachsen immer auf die Drehkonvention an. So macht es beispielsweise einen Unterschied, ob ein Körper erst um die \mathbf{e}_x - und dann um die \mathbf{e}_y -Achse gedreht wird oder erst um die \mathbf{e}_y - und dann um die \mathbf{e}_x -Achse. Genauso macht es einen Unterschied ob ein Körper um die \mathbf{e}_x - und dann um die neue oder um die ursprüngliche \mathbf{e}_y -Achse gedreht wird. Sinnvoll ist es dabei immer eine Drehkonvention zu wählen, die identisch ist mit der Drehkonvention anderer im Rahmen derselben Arbeit benutzten Software-Tools. Das Programm netfabb liefert beispielsweise eine grafische Benutzeroberfläche (GUI) zur Bearbeitung und Manipulation von STL-Dateien. So lassen sich damit auch STL-Dateien einfach drehen durch Angabe von Drehwinkeln um die \mathbf{e}_x - (α_x), \mathbf{e}_y - (α_y) und \mathbf{e}_z -Achse (α_z). Dieselbe Möglichkeit bietet auch die für den Tropfenerzeuger erhaltene GUI, welche die im Rahmen dieser Arbeit implementierte DLL zur automatischen Erzeugung von Maschinencode aus STL-Dateien nutzt. Für das Drehen eines Körpers in der GUI werden die Funktionalitäten des Microsoft .NET Frameworks genutzt. Diese drehen den Körper nach derselben Konvention wie die Software netfabb. Die STL-Dateien müssen also in der in dieser Arbeit entwickelten Software im Sinne der Benutzerfreundlichkeit auch auf dieselbe Methode gedreht werden, um daraus entsprechenden Maschinencode erzeugen zu können. Im Folgenden wird die zugrunde liegende Drehkonvention mathematisch in Form von homogenen Transformationsmatrizen genauestens erläutert.

5.2.1 Berechnung der Transformationsmatrizen

Drehkonvention

Bei der zugrunde liegenden Drehkonvention wird zunächst um den Winkel α_x um die (ursprüngliche) \mathbf{e}_x -Achse gedreht, danach um den Winkel α_y um die ursprüngliche \mathbf{e}_y -Achse und zuletzt um den Winkel α_z um die ursprüngliche \mathbf{e}_z -Achse. Im ursprünglichen Koordinatensystem gibt es keine Drehungen und keine Translationen. Deswegen ist die Transformationsmatrix im Koordinatensystem A (Ursprungskoordinatensystem) die Einheitsmatrix \mathbf{I} :

$$\mathbf{I} = \begin{pmatrix} \mathbf{e}_x & \mathbf{e}_y & \mathbf{e}_z & \mathbf{0} \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}. \quad (5.55)$$

Drehung vom Koordinatensystem A ins Koordinatensystem B um α_x

Die ursprüngliche \mathbf{e}_x -Achse im Ursprungskoordinatensystem A ist gegeben durch:

$${}^A\mathbf{v} = {}^A\mathbf{e}_x = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \quad (5.56)$$

Somit ergibt sich nach Satz 12 die homogene Transformationsmatrix:

$${}^A\mathbf{T}_B \left({}^A\mathbf{v}, \alpha_x \right) = \begin{pmatrix} {}^A\mathbf{R}_B \left({}^A\mathbf{v}, \alpha_x \right) & \mathbf{0} \\ \mathbf{0} & 1 \end{pmatrix}. \quad (5.57)$$

Drehung vom Koordinatensystem B ins Koordinatensystem C um α_y

Diese Drehung erfolgt um die ursprüngliche \mathbf{e}_y -Achse des Koordinatensystems A . Dafür muss der Vektor

$${}^A\mathbf{v} = {}^A\mathbf{e}_y = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \quad (5.58)$$

ins Koordinatensystem B transformiert werden. Es gilt nach Satz 9:

$$\begin{pmatrix} {}^B\mathbf{v} \\ 1 \end{pmatrix} = ({}^A\mathbf{T}_B)^{-1} \cdot \begin{pmatrix} {}^A\mathbf{v} \\ 1 \end{pmatrix}. \quad (5.59)$$

Somit ergibt sich nach Satz 12 die homogene Transformationsmatrix:

$${}^B\mathbf{T}_C({}^B\mathbf{v}, \alpha_y) = \begin{pmatrix} {}^B\mathbf{R}_C({}^B\mathbf{v}, \alpha_y) & \mathbf{0} \\ \mathbf{0} & 1 \end{pmatrix}. \quad (5.60)$$

Drehung vom Koordinatensystem C ins Koordinatensystem D um α_z

Diese Drehung erfolgt um die ursprüngliche \mathbf{e}_z -Achse des Koordinatensystems A . Dafür muss der Vektor

$${}^A\mathbf{v} = {}^A\mathbf{e}_z = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \quad (5.61)$$

ins Koordinatensystem C transformiert werden. Es gilt nach Satz 9:

$$\begin{pmatrix} {}^C\mathbf{v} \\ 1 \end{pmatrix} = ({}^A\mathbf{T}_B \cdot {}^B\mathbf{T}_C)^{-1} \cdot \begin{pmatrix} {}^A\mathbf{v} \\ 1 \end{pmatrix}. \quad (5.62)$$

Somit ergibt sich nach Satz 12 die homogene Transformationsmatrix:

$${}^C\mathbf{T}_D({}^C\mathbf{v}, \alpha_z) = \begin{pmatrix} {}^C\mathbf{R}_D({}^C\mathbf{v}, \alpha_z) & \mathbf{0} \\ \mathbf{0} & 1 \end{pmatrix}. \quad (5.63)$$

Gesamte Transformation

Satz 12 (Transformationskonvention zur Drehung von Körpern im \mathbb{R}^3):

Die komplette Transformationsmatrix zur Drehung von Körpern im \mathbb{R}^3 nach der Drehkonvention aus Kapitel 5.2.1 ist gegeben durch Multiplikation der einzelnen Transformationsmatrizen aus (5.57), (5.60) und (5.63) auf folgende Weise:

$$\mathbf{T}(\alpha_x, \alpha_y, \alpha_z) := {}^A\mathbf{T}_D = {}^A\mathbf{T}_B \cdot {}^B\mathbf{T}_C \cdot {}^C\mathbf{T}_D. \quad (5.64)$$

5.2.2 Abstandsberechnungen

Gegeben sei ein Segment $\mathbf{S}(\mathbf{p}_i, \mathbf{p}_{i+1})$ nach Definition 5. Weiter sei:

$$\mathbf{d}_i := \frac{\mathbf{p}_{i+1} - \mathbf{p}_i}{|\mathbf{p}_{i+1} - \mathbf{p}_i|}. \quad (5.65)$$

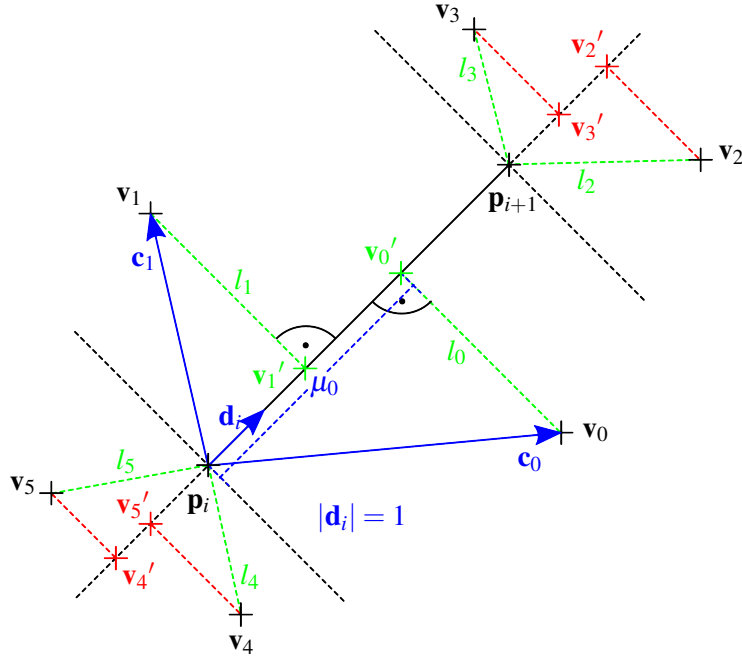


Abbildung 5.17: Minimale Abstände l_j zwischen einem beliebigen Punkt $\mathbf{v}_j \in \mathbb{R}^3$, $j \in \mathbb{N}_0$ zu einem Segment $\mathbf{S}(\mathbf{p}_i, \mathbf{p}_{i+1})$. \mathbf{v}_j' ist die orthogonale Projektion des Punktes \mathbf{v}_j auf die durch \mathbf{p}_i und \mathbf{p}_{i+1} bestimmte Gerade $\mathbf{G}(\mathbf{p}_i, \mathbf{p}_{i+1})$. Nur \mathbf{v}_0' und \mathbf{v}_1' liegen auf dem Segment $\mathbf{S}(\mathbf{p}_i, \mathbf{p}_{i+1})$. Die minimalen Abstände berechnen sich demzufolge nach Formel (5.69).

Gesucht ist nun der minimale Abstand von einem beliebigen Punkt $\mathbf{v}_j \in \mathbb{R}^3$ zum Segment $\mathbf{S}(\mathbf{p}_i, \mathbf{p}_{i+1})$, wie in Abbildung 5.17 zu sehen ist:

$$l_{\min}(\mathbf{S}(\mathbf{p}_i, \mathbf{p}_{i+1}), \mathbf{v}_j) := l_j := \min_{\mu \in \mathbb{R}} \{ |\mathbf{v}_j - \mathbf{v}'_j(\mu)| \mid \mathbf{v}'_j(\mu) = \mathbf{p}_i + \mu \cdot \mathbf{d}_i, \mathbf{v}'_j(\mu) \in \mathbf{S}(\mathbf{p}_i, \mathbf{p}_{i+1}) \}. \quad (5.66)$$

Mit

$$\mathbf{c}_j := \mathbf{v}_j - \mathbf{p}_i \quad (5.67)$$

gilt:

$$\mu_j = |\mathbf{c}_j| \cdot \cos \alpha = |\mathbf{c}_j| \cdot \underbrace{|\mathbf{d}_i|}_{=1} \cdot \cos \alpha = \mathbf{c}_j \circ \mathbf{d}_i. \quad (5.68)$$

Satz 13 (Minimaler Abstand zwischen einem Punkt und einem Segment):

Für den minimalen Abstand eines Punktes \mathbf{v}_j zu einem Segment $\mathbf{S}(\mathbf{p}_i, \mathbf{p}_{i+1})$ nach der Definition in (5.66) gilt mit (5.67):

$$l_{\min}(\mathbf{S}(\mathbf{p}_i, \mathbf{p}_{i+1}), \mathbf{v}_j) = \begin{cases} |\mathbf{c}_j - (\mathbf{c}_j \circ \mathbf{d}_i) \cdot \mathbf{d}_i| & \text{falls } 0 \leq \mathbf{c}_j \circ \mathbf{d}_i \leq |\mathbf{p}_{i+1} - \mathbf{p}_i| \\ \min(|\mathbf{v}_j - \mathbf{p}_i|, |\mathbf{v}_j - \mathbf{p}_{i+1}|) & \text{sonst} \end{cases}. \quad (5.69)$$

Anmerkung: Der erste Fall tritt ein, falls die Projektion des Punktes \mathbf{v}_j auf dem Segment $\mathbf{S}(\mathbf{p}_i, \mathbf{p}_{i+1})$ liegt. Für eine Gerade $\mathbf{G}(\mathbf{p}_i, \mathbf{p}_{i+1})$ ist also immer dieser Fall erfüllt, das heißt:

$$l_{\min}(\mathbf{G}(\mathbf{p}_i, \mathbf{p}_{i+1}), \mathbf{v}_j) = |\mathbf{c}_j - (\mathbf{c}_j \circ \mathbf{d}_i) \cdot \mathbf{d}_i|. \quad (5.70)$$

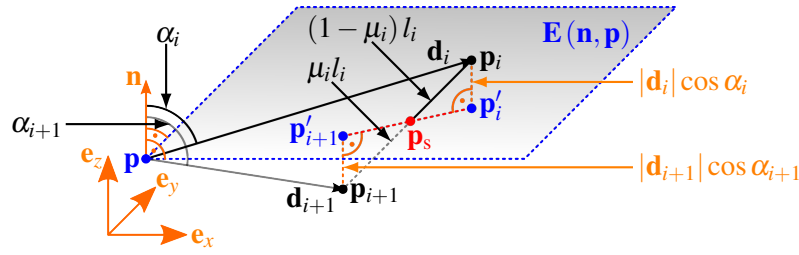


Abbildung 5.18: Schnittpunkt \mathbf{p}_s zwischen einer Ebene $\mathbf{E}(\mathbf{n}, \mathbf{p})$ und einem Segment $\mathbf{S}(\mathbf{p}_i, \mathbf{p}_{i+1})$. \mathbf{p}'_i ist die Projektion von \mathbf{p}_i auf $\mathbf{E}(\mathbf{n}, \mathbf{p})$, \mathbf{p}'_{i+1} ist die Projektion von \mathbf{p}_{i+1} auf $\mathbf{E}(\mathbf{n}, \mathbf{p})$.

Gegeben sei nun ein Punkt $\mathbf{p} \in \mathbb{R}^3$ und eine Ebene $\mathbf{E}(\mathbf{n}, \mathbf{p}_i)$ nach Definition 7. Gesucht ist nun der minimale Abstand von \mathbf{p} zu \mathbf{E} :

$$l_{\min}(\mathbf{E}, \mathbf{p}) := \min_{\mathbf{x} \in \mathbf{E}} \{|\mathbf{p} - \mathbf{x}|\} . \quad (5.71)$$

Satz 14 (Minimaler Abstand zwischen einem Punkt und einer Ebene):

Für den minimalen Abstand von \mathbf{p} zu \mathbf{E} gilt:

$$l_{\min}(\mathbf{E}, \mathbf{p}) = |\mathbf{n} \circ (\mathbf{p} - \mathbf{p}_i)| . \quad (5.72)$$

Falls

$$\mathbf{n} \circ (\mathbf{p} - \mathbf{p}_i) > 0 \quad (5.73)$$

gilt, dann liegen \mathbf{p} und der Normalenvektor \mathbf{n} auf derselben Seite von \mathbf{E} . Falls

$$\mathbf{n} \circ (\mathbf{p} - \mathbf{p}_i) = 0 \quad (5.74)$$

gilt, dann liegt \mathbf{p} direkt in der Ebene \mathbf{E} . Falls

$$\mathbf{n} \circ (\mathbf{p} - \mathbf{p}_i) < 0 \quad (5.75)$$

gilt, dann liegen \mathbf{p} und der Normalenvektor \mathbf{n} auf verschiedenen Seiten von \mathbf{E} .

Beweis: Nach der Definition 3 des Skalarprodukts ist $|\mathbf{n} \circ (\mathbf{p} - \mathbf{p}_i)|$ die Länge der senkrechten Projektion von $\mathbf{p} - \mathbf{p}_i$ auf \mathbf{n} . Diese Projektion hat zur Ebene \mathbf{E} denselben Abstand wie der Punkt \mathbf{p} , da \mathbf{n} senkrecht auf \mathbf{E} steht. Insbesondere ist $\mathbf{n} \circ (\mathbf{p} - \mathbf{p}_i)$ nach Definition des Skalarprodukts größer 0, falls $0 \leq \angle(\mathbf{n}, \mathbf{p} - \mathbf{p}_i) < \frac{\pi}{2}$ und kleiner 0, falls $\frac{\pi}{2} < \angle(\mathbf{n}, \mathbf{p} - \mathbf{p}_i) \leq \pi$. $\mathbf{n} \circ (\mathbf{p} - \mathbf{p}_i) = 0$ gilt, falls $\angle(\mathbf{n}, \mathbf{p} - \mathbf{p}_i) = \frac{\pi}{2}$, das heißt also, falls \mathbf{p} in der Ebene \mathbf{E} liegt. q.e.d.

5.2.3 Schnittpunktberechnungen

Schnittpunkt zwischen einer Ebene und einem Segment

Gesucht ist, wie in Abbildung 5.18 zu sehen ist, der Schnittpunkt \mathbf{p}_s zwischen einer Ebene $\mathbf{E}(\mathbf{n}, \mathbf{p})$ und einem Segment $\mathbf{S}(\mathbf{p}_i, \mathbf{p}_{i+1})$. Gesucht ist also ein

$$\mu_i \in [0, 1] \quad (5.76)$$

mit

$$\mathbf{p}_s = \mathbf{p}_{i+1} + \mu_i (\mathbf{p}_i - \mathbf{p}_{i+1}) = \mu_i \mathbf{p}_i + (1 - \mu_i) \mathbf{p}_{i+1}, \quad (5.77)$$

so dass gilt:

$$\mathbf{p}_s \in \mathbf{E}(\mathbf{n}, \mathbf{p}). \quad (5.78)$$

Nach Satz 14 gilt für den minimalen Abstand von \mathbf{p}_i zur Ebene $\mathbf{E}(\mathbf{n}, \mathbf{p})$:

$$l_{\min}(\mathbf{E}, \mathbf{p}_i) = \underbrace{|\mathbf{n} \circ (\mathbf{p}_i - \mathbf{p})|}_{=: f_i}. \quad (5.79)$$

Genauso gilt für den minimalen Abstand von \mathbf{p}_{i+1} zur Ebene $\mathbf{E}(\mathbf{n}, \mathbf{p})$:

$$l_{\min}(\mathbf{E}, \mathbf{p}_{i+1}) = \underbrace{|\mathbf{n} \circ (\mathbf{p}_{i+1} - \mathbf{p})|}_{=: f_{i+1}}. \quad (5.80)$$

Mit $\mathbf{d}_i := \mathbf{p}_i - \mathbf{p}$ und $\mathbf{d}_{i+1} := \mathbf{p}_{i+1} - \mathbf{p}$ gilt:

$$f_i = \mathbf{n} \circ \mathbf{d}_i = \underbrace{|\mathbf{n}|}_{=1} |\mathbf{d}_i| \cos \alpha_i = |\mathbf{d}_i| \cos \alpha_i \quad (5.81)$$

und

$$f_{i+1} = \mathbf{n} \circ \mathbf{d}_{i+1} = \underbrace{|\mathbf{n}|}_{=1} |\mathbf{d}_{i+1}| \cos \alpha_{i+1} = |\mathbf{d}_{i+1}| \cos \alpha_{i+1}. \quad (5.82)$$

Als Folge von Satz 14 gibt es genau dann einen Schnittpunkt \mathbf{p}_s , wenn

$$(f_i \geq 0 \wedge f_{i+1} < 0) \vee (f_i < 0 \wedge f_{i+1} \geq 0) \quad (5.83)$$

gilt. Mit

$$l_i := |\mathbf{p}_i - \mathbf{p}_{i+1}| \neq 0 \quad (5.84)$$

gilt:

$$\frac{\mu_i l_i}{(1 - \mu_i) l_i} = -\frac{f_{i+1}}{f_i} \Leftrightarrow \mu_i f_i = -f_{i+1} + \mu_i f_{i+1} \Leftrightarrow \mu_i = \frac{f_{i+1}}{\underbrace{f_{i+1} - f_i}_{\neq 0}}, \quad (5.85)$$

falls $f_i \geq 0 \wedge f_{i+1} < 0$ bzw.

$$\frac{(1 - \mu_i) l_i}{\mu_i l_i} = -\frac{f_i}{f_{i+1}} \Leftrightarrow \mu_i f_i = -f_{i+1} + \mu_i f_{i+1} \Leftrightarrow \mu_i = \frac{f_{i+1}}{\underbrace{f_{i+1} - f_i}_{\neq 0}}, \quad (5.86)$$

falls $f_i < 0 \wedge f_{i+1} \geq 0$. Mit

$$\mu_i = \frac{f_{i+1}}{f_{i+1} - f_i} \quad (5.87)$$

gilt also:

$$\mathbf{p}_s = \mu_i \mathbf{p}_i + (1 - \mu_i) \mathbf{p}_{i+1}. \quad (5.88)$$

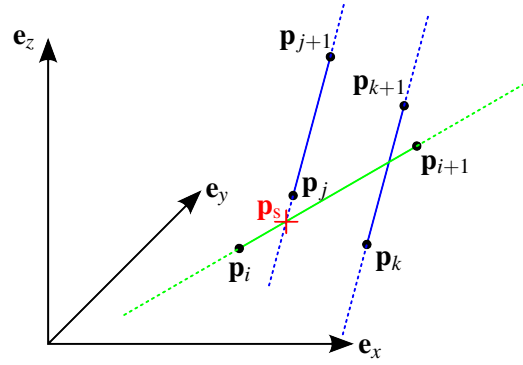


Abbildung 5.19: Die beiden Geraden $G(\mathbf{p}_j, \mathbf{p}_{j+1})$ und $G(\mathbf{p}_k, \mathbf{p}_{k+1})$ bzw. die beiden Segmente $S(\mathbf{p}_j, \mathbf{p}_{j+1})$ und $S(\mathbf{p}_k, \mathbf{p}_{k+1})$ sind parallel, aber nicht identisch. Die Geraden $G(\mathbf{p}_i, \mathbf{p}_{i+1})$ und $G(\mathbf{p}_j, \mathbf{p}_{j+1})$ schneiden sich im Punkt \mathbf{p}_s . Es gilt $\mathbf{p}_s \in S(\mathbf{p}_i, \mathbf{p}_{i+1})$, aber $\mathbf{p}_s \notin S(\mathbf{p}_j, \mathbf{p}_{j+1})$. Die beiden Geraden $G(\mathbf{p}_i, \mathbf{p}_{i+1})$ und $G(\mathbf{p}_k, \mathbf{p}_{k+1})$ bzw. die beiden Segmente $S(\mathbf{p}_i, \mathbf{p}_{i+1})$ und $S(\mathbf{p}_k, \mathbf{p}_{k+1})$ sind windschief.

Schnittpunkt zweier Geraden im \mathbb{R}^3

Gegeben seien im Folgenden zwei Geraden

$$G(\mathbf{p}_i, \mathbf{p}_{i+1}) := \left\{ \mathbf{p}_i + \lambda \cdot \mathbf{d}_i \mid \lambda \in \mathbb{R}, \mathbf{d}_i = \frac{\mathbf{p}_{i+1} - \mathbf{p}_i}{|\mathbf{p}_{i+1} - \mathbf{p}_i|} \right\} \quad (5.89)$$

und

$$G(\mathbf{p}_j, \mathbf{p}_{j+1}) := \left\{ \mathbf{p}_j + \xi \cdot \mathbf{d}_j \mid \xi \in \mathbb{R}, \mathbf{d}_j = \frac{\mathbf{p}_{j+1} - \mathbf{p}_j}{|\mathbf{p}_{j+1} - \mathbf{p}_j|} \right\} \quad (5.90)$$

nach Definition 6. Es gibt nun folgende vier Möglichkeiten:

- $G(\mathbf{p}_i, \mathbf{p}_{i+1})$ und $G(\mathbf{p}_j, \mathbf{p}_{j+1})$ sind identisch.
- $G(\mathbf{p}_i, \mathbf{p}_{i+1})$ und $G(\mathbf{p}_j, \mathbf{p}_{j+1})$ sind parallel, aber nicht identisch.
- $G(\mathbf{p}_i, \mathbf{p}_{i+1})$ und $G(\mathbf{p}_j, \mathbf{p}_{j+1})$ haben genau einen Schnittpunkt.
- $G(\mathbf{p}_i, \mathbf{p}_{i+1})$ und $G(\mathbf{p}_j, \mathbf{p}_{j+1})$ sind windschief.

Abbildung 5.19 zeigt einige Beispiele für die gegenseitige Lage von Geraden. Falls für den Winkel α zwischen \mathbf{d}_i und \mathbf{d}_j gilt

$$\alpha = \sin^{-1}(\mathbf{d}_i \times \mathbf{d}_j) = 0, \quad (5.91)$$

dann sind die beiden Geraden entweder identisch oder parallel (nicht identisch). Nach der Anmerkung zu Satz 13 gilt für den minimalen Abstand von \mathbf{p}_j zu $G(\mathbf{p}_i, \mathbf{p}_{i+1})$:

$$l_{\min}(G(\mathbf{p}_i, \mathbf{p}_{i+1}), \mathbf{p}_j) = |\mathbf{p}_j - \mathbf{p}_i - ((\mathbf{p}_j - \mathbf{p}_i) \circ \mathbf{d}_i) \cdot \mathbf{d}_i|. \quad (5.92)$$

Falls also (5.91) erfüllt ist und $l_{\min}(G(\mathbf{p}_i, \mathbf{p}_{i+1}), \mathbf{p}_j) = 0$, dann sind die beiden Geraden identisch. Falls (5.91) erfüllt ist und $l_{\min}(G(\mathbf{p}_i, \mathbf{p}_{i+1}), \mathbf{p}_j) \neq 0$, dann sind die beiden Geraden parallel, aber nicht identisch. Sei also im Folgenden (5.91) nicht erfüllt, das heißt \mathbf{d}_i und \mathbf{d}_j sind linear unabhängig bzw. $\mathbf{d}_i \times \mathbf{d}_j \neq (0, 0, 0)^T$. Das heißt, durch

$$\mathbf{n} := \mathbf{d}_i \times \mathbf{d}_j \quad (5.93)$$

wird ein Vektor \mathbf{n} definiert, der auf \mathbf{d}_i und \mathbf{d}_j senkrecht steht. Folglich wird durch

$$\mathbf{E} := \left\{ \mathbf{p}_i + \lambda \cdot \mathbf{d}_i + \mu \cdot \mathbf{n} \mid \lambda \in \mathbb{R}, \mu \in \mathbb{R}, \mathbf{d}_i = \frac{\mathbf{p}_{i+1} - \mathbf{p}_i}{|\mathbf{p}_{i+1} - \mathbf{p}_i|}, \mathbf{n} = \mathbf{d}_i \times \mathbf{d}_j \right\} \quad (5.94)$$

eine Ebene definiert, die einen Schnittpunkt mit $\mathbf{G}(\mathbf{p}_j, \mathbf{p}_{j+1})$ hat. Dieser Schnittpunkt kann berechnet werden, indem man die Variable ξ aus der Gleichsetzung von \mathbf{E} und $\mathbf{G}(\mathbf{p}_j, \mathbf{p}_{j+1})$ berechnet:

$$\mathbf{p}_i + \lambda \mathbf{d}_i + \mu \mathbf{n} = \mathbf{p}_j + \xi \mathbf{d}_j. \quad (5.95)$$

Dies ist äquivalent zu:

$$\lambda \mathbf{d}_i + \mu \mathbf{n} + \xi (-\mathbf{d}_j) = \mathbf{p}_j - \mathbf{p}_i. \quad (5.96)$$

Mit

$$\mathbf{A} := (\mathbf{d}_i \quad \mathbf{n} \quad -\mathbf{d}_j) \in \mathbb{R}^{3 \times 3}, \quad \mathbf{x} := (\lambda \quad \mu \quad \xi)^T \in \mathbb{R}^3, \quad \mathbf{b} := \mathbf{p}_j - \mathbf{p}_i \in \mathbb{R}^3 \quad (5.97)$$

gilt:

$$\mathbf{A}\mathbf{x} = \mathbf{b}. \quad (5.98)$$

Dieses lässt sich nach Satz 4 eindeutig lösen, da die Determinante D von A ungleich 0 ist, weil die Spaltenvektoren von \mathbf{A} nach Voraussetzung linear unabhängig sind (siehe Bronstein et al. (2008)). Sei

$$\mathbf{x}_s := (\lambda_s \quad \mu_s \quad \xi_s)^T \quad (5.99)$$

die Lösung von (5.98), dann gilt für den Schnittpunkt \mathbf{p}_s von \mathbf{E} und $\mathbf{G}(\mathbf{p}_j, \mathbf{p}_{j+1})$:

$$\mathbf{p}_s = \mathbf{p}_j + \xi_s \mathbf{d}_j. \quad (5.100)$$

Falls also

$$l_{\min}(\mathbf{G}(\mathbf{p}_i, \mathbf{p}_{i+1}), \mathbf{p}_s) = 0 \quad (5.101)$$

gilt, das heißt, der Schnittpunkt von \mathbf{E} und $\mathbf{G}(\mathbf{p}_j, \mathbf{p}_{j+1})$ liegt auch auf der Geraden $\mathbf{G}(\mathbf{p}_i, \mathbf{p}_{i+1})$, dann haben die beiden Geraden genau diesen einen Schnittpunkt \mathbf{p}_s . Ansonsten sind die Geraden windschief.

Dieser Lösungsweg erscheint vielleicht nicht der eleganteste zu sein, um die Schnittpunkte zweier Geraden zu bestimmen, jedoch eignet er sich für eine Implementierung sehr gut.

Schnittpunkte zweier Segmente im \mathbb{R}^3

Gegeben seien im Folgenden zwei Segmente

$$\mathbf{S}(\mathbf{p}_i, \mathbf{p}_{i+1}) := \left\{ \mathbf{p}_i + \lambda \cdot \mathbf{d}_i \mid \lambda \in \mathbb{R}, 0 \leq \lambda \leq 1, \mathbf{d}_i = \frac{\mathbf{p}_{i+1} - \mathbf{p}_i}{|\mathbf{p}_{i+1} - \mathbf{p}_i|} \right\} \quad (5.102)$$

und

$$\mathbf{S}(\mathbf{p}_j, \mathbf{p}_{j+1}) := \left\{ \mathbf{p}_j + \xi \cdot \mathbf{d}_j \mid \xi \in \mathbb{R}, 0 \leq \xi \leq 1, \mathbf{d}_j = \frac{\mathbf{p}_{j+1} - \mathbf{p}_j}{|\mathbf{p}_{j+1} - \mathbf{p}_j|} \right\} \quad (5.103)$$

Für die Prüfung der gegenseitigen Lage zweier Segmente wird zunächst genauso vorgegangen, wie bei den zwei entsprechenden Geraden $\mathbf{G}(\mathbf{p}_i, \mathbf{p}_{i+1})$ und $\mathbf{G}(\mathbf{p}_j, \mathbf{p}_{j+1})$. Falls die beiden Geraden einen Schnittpunkt \mathbf{p}_s haben, dann muss für diesen zusätzlich geprüft werden, ob

$$\mathbf{p}_s \in \mathbf{S}(\mathbf{p}_i, \mathbf{p}_{i+1}) \quad (5.104)$$

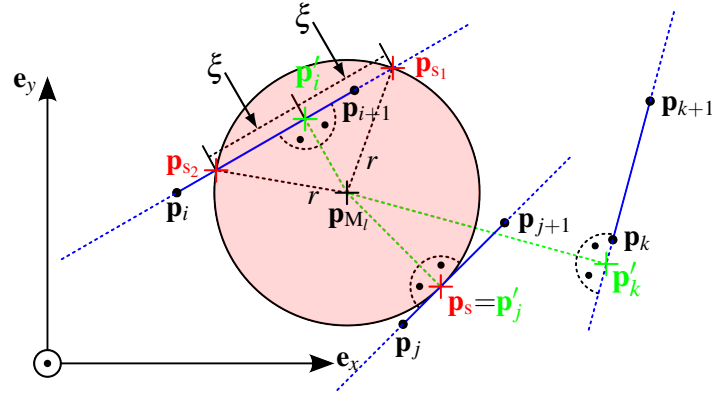


Abbildung 5.20: Der Kreisrand $\partial K_{r, e_z}(\mathbf{p}_{M_l})$ hat mit der Geraden $G(\mathbf{p}_i, \mathbf{p}_{i+1})$ genau zwei Schnittpunkte \mathbf{p}_{s_1} und \mathbf{p}_{s_2} . Es gilt $\mathbf{p}_{s_1} \in S(\mathbf{p}_i, \mathbf{p}_{i+1})$, aber $\mathbf{p}_{s_2} \notin S(\mathbf{p}_i, \mathbf{p}_{i+1})$. Mit der Geraden $G(\mathbf{p}_j, \mathbf{p}_{j+1})$ hat der Kreisrand $\partial K_{r, e_z}(\mathbf{p}_{M_l})$ genau einen Schnittpunkt \mathbf{p}_s . Es gilt $\mathbf{p}_s \in S(\mathbf{p}_j, \mathbf{p}_{j+1})$. Mit der Geraden $G(\mathbf{p}_k, \mathbf{p}_{k+1})$ hat der Kreisrand $\partial K_{r, e_z}(\mathbf{p}_{M_l})$ keinen Schnittpunkt. Mit \mathbf{p}'_i , \mathbf{p}'_j und \mathbf{p}'_k werden die Projektionen von \mathbf{p}_{M_l} auf die jeweiligen Geraden bezeichnet.

und

$$\mathbf{p}_s \in S(\mathbf{p}_j, \mathbf{p}_{j+1}) \quad (5.105)$$

gilt. Es müssen beide Bedingungen erfüllt sein. Dies ist genau dann der Fall, wenn nach Satz 13

$$l_{\min}(S(\mathbf{p}_i, \mathbf{p}_{i+1}), \mathbf{p}_s) = 0 \quad (5.106)$$

und

$$l_{\min}(S(\mathbf{p}_j, \mathbf{p}_{j+1}), \mathbf{p}_s) = 0 \quad (5.107)$$

gilt.

Schnittpunkte einer Geraden und eines Kreisrandes im \mathbb{R}^3

Gegeben sei im Folgenden ein Kreis $K_{r, \mathbf{n}}(\mathbf{p}_{M_j})$ und der entsprechende Kreisrand $\partial K_{r, \mathbf{n}}(\mathbf{p}_{M_j})$ nach Definition 8. Des Weiteren definieren \mathbf{p}_{M_j} und der Normalenvektor \mathbf{n} eine Ebene \mathbf{E} , in der auch die Gerade

$$G(\mathbf{p}_i, \mathbf{p}_{i+1}) := \left\{ \mathbf{p}_i + \lambda \cdot \mathbf{d}_i \mid \lambda \in \mathbb{R}, \mathbf{d}_i = \frac{\mathbf{p}_{i+1} - \mathbf{p}_i}{|\mathbf{p}_{i+1} - \mathbf{p}_i|} \right\} \quad (5.108)$$

liegt. Es gibt nun folgende drei Möglichkeiten:

- $G(\mathbf{p}_i, \mathbf{p}_{i+1})$ und $\partial K_{r, \mathbf{n}}(\mathbf{p}_{M_j})$ haben keinen Schnittpunkt.
- $G(\mathbf{p}_i, \mathbf{p}_{i+1})$ und $\partial K_{r, \mathbf{n}}(\mathbf{p}_{M_j})$ berühren sich in genau einem Punkt. In diesem Fall ist die Gerade $G(\mathbf{p}_i, \mathbf{p}_{i+1})$ auch die Tangente an $\partial K_{r, \mathbf{n}}(\mathbf{p}_{M_j})$ im gemeinsamen Berührungspunkt.
- $G(\mathbf{p}_i, \mathbf{p}_{i+1})$ und $\partial K_{r, \mathbf{n}}(\mathbf{p}_{M_j})$ haben genau zwei Schnittpunkte.

Abbildung 5.20 zeigt einige Beispiele für die gegenseitige Lage von Geraden. Nach der Anmerkung zu Satz 13 gilt für den minimalen Abstand von \mathbf{p}_{M_j} zu $G(\mathbf{p}_i, \mathbf{p}_{i+1})$:

$$l_{\min}(G(\mathbf{p}_i, \mathbf{p}_{i+1}), \mathbf{p}_{M_j}) = |\mathbf{p}_{M_j} - \mathbf{p}_i - ((\mathbf{p}_{M_j} - \mathbf{p}_i) \circ \mathbf{d}_i) \cdot \mathbf{d}_i| \quad (5.109)$$

Falls also

$$l_{\min}(\mathbf{G}(\mathbf{p}_i, \mathbf{p}_{i+1}), \mathbf{p}_{M_j}) > r \quad (5.110)$$

gilt, dann haben $\mathbf{G}(\mathbf{p}_i, \mathbf{p}_{i+1})$ und $\partial\mathbf{K}_{r,\mathbf{n}}(\mathbf{p}_{M_j})$ keinen Schnittpunkt. Falls

$$l_{\min}(\mathbf{G}(\mathbf{p}_i, \mathbf{p}_{i+1}), \mathbf{p}_{M_j}) = r \quad (5.111)$$

gilt, dann haben $\mathbf{G}(\mathbf{p}_i, \mathbf{p}_{i+1})$ und $\partial\mathbf{K}_{r,\mathbf{n}}(\mathbf{p}_{M_j})$ einen gemeinsamen Berührungspunkt. In diesem Fall ist die senkrechte Projektion von \mathbf{p}_{M_j} auf die Gerade $\mathbf{G}(\mathbf{p}_i, \mathbf{p}_{i+1})$ identisch mit dem Berührungspunkt / Schnittpunkt \mathbf{p}_s und es gilt wie in Kapitel 5.2.2 hergeleitet:

$$\mathbf{p}_s := \mathbf{p}'_i(\mu_{M_j}) = \mathbf{p}_i + \mu_{M_j} \mathbf{d}_i \quad (5.112)$$

mit

$$\mu_{M_j} = (\mathbf{p}_{M_j} - \mathbf{p}_i) \circ \mathbf{d}_i. \quad (5.113)$$

Falls

$$l_{\min}(\mathbf{G}(\mathbf{p}_i, \mathbf{p}_{i+1}), \mathbf{p}_{M_j}) < r \quad (5.114)$$

gilt, dann haben $\mathbf{G}(\mathbf{p}_i, \mathbf{p}_{i+1})$ und $\partial\mathbf{K}_{r,\mathbf{n}}(\mathbf{p}_{M_j})$ genau zwei Schnittpunkte. In diesem Fall gilt für die Projektion von \mathbf{p}_{M_j} auf die Gerade $\mathbf{G}(\mathbf{p}_i, \mathbf{p}_{i+1})$:

$$\mathbf{p}'_i(\mu_{M_j}) = \mathbf{p}_i + \mu_{M_j} \mathbf{d}_i \quad (5.115)$$

mit

$$\mu_{M_j} = (\mathbf{p}_{M_j} - \mathbf{p}_i) \circ \mathbf{d}_i. \quad (5.116)$$

Mit

$$\xi := \sqrt{r^2 - l_{\min}^2(\mathbf{G}(\mathbf{p}_i, \mathbf{p}_{i+1}), \mathbf{p}_{M_j})} \quad (5.117)$$

gilt für die beiden Schnittpunkte:

$$\mathbf{p}_{s_1} = \mathbf{p}'_i(\mu_{M_j}) + \xi \mathbf{d}_i \quad (5.118)$$

bzw.

$$\mathbf{p}_{s_2} = \mathbf{p}'_i(\mu_{M_j}) - \xi \mathbf{d}_i. \quad (5.119)$$

Schnittpunkte eines Segments und eines Kreisrandes im \mathbb{R}^3

Gegeben sei im Folgenden ein Kreis $\mathbf{K}_{r,\mathbf{n}}(\mathbf{p}_{M_j})$ und der entsprechende Kreisrand $\partial\mathbf{K}_{r,\mathbf{n}}(\mathbf{p}_{M_j})$ nach Definition 8. Des Weiteren definieren \mathbf{p}_{M_j} und der Normalenvektor \mathbf{n} eine Ebene \mathbf{E} , in der auch das Segment

$$\mathbf{S}(\mathbf{p}_i, \mathbf{p}_{i+1}) := \left\{ \mathbf{p}_i + \lambda \cdot \mathbf{d}_i \mid \lambda \in \mathbb{R}, 0 \leq \lambda \leq 1, \mathbf{d}_i = \frac{\mathbf{p}_{i+1} - \mathbf{p}_i}{|\mathbf{p}_{i+1} - \mathbf{p}_i|} \right\} \quad (5.120)$$

liegt. Für die Schnittpunktberechnung wird zunächst wie im vorigen Abschnitt von einer Geraden $\mathbf{G}(\mathbf{p}_i, \mathbf{p}_{i+1})$ und dem Kreisrand $\partial\mathbf{K}_{r,\mathbf{n}}(\mathbf{p}_{M_j})$ ausgegangen. Sei \mathbf{p}_s ein Schnittpunkt, dann muss für diesen geprüft werden, ob er auch auf dem Segment liegt, das heißt, ob

$$\mathbf{p}_s \in \mathbf{S}(\mathbf{p}_i, \mathbf{p}_{i+1}) \quad (5.121)$$

gilt. Dies ist genau dann der Fall, wenn nach Satz 13

$$l_{\min}(\mathbf{S}(\mathbf{p}_i, \mathbf{p}_{i+1}), \mathbf{p}_s) = 0 \quad (5.122)$$

gilt.

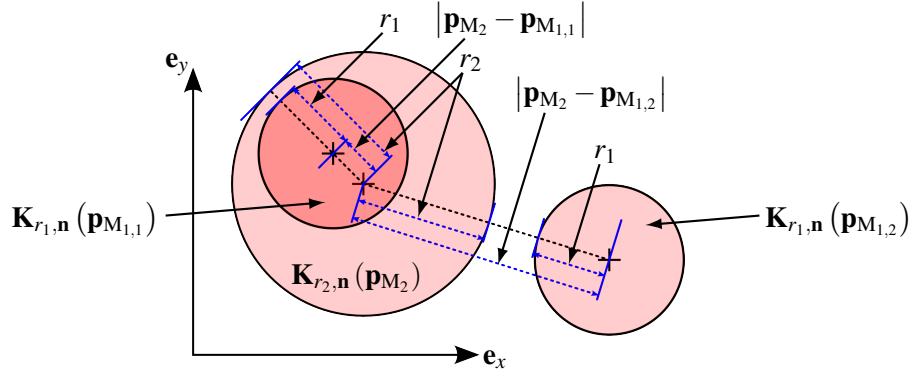


Abbildung 5.21: Drei Kreise $\mathbf{K}_{r_2, n}(\mathbf{p}_{M_2})$, $\mathbf{K}_{r_1, n}(\mathbf{p}_{M_{1,1}})$ und $\mathbf{K}_{r_1, n}(\mathbf{p}_{M_{1,2}})$ mit $r_2 \geq r_1$. $\mathbf{K}_{r_2, n}(\mathbf{p}_{M_2})$ und $\mathbf{K}_{r_1, n}(\mathbf{p}_{M_{1,1}})$ erfüllen die linke Bedingung in (5.127). $\mathbf{K}_{r_2, n}(\mathbf{p}_{M_2})$ und $\mathbf{K}_{r_1, n}(\mathbf{p}_{M_{1,2}})$ erfüllen die rechte Bedingung in (5.127). Hier haben die Kreisränder $\partial\mathbf{K}_{r_1, n}(\mathbf{p}_{M_{1,1}})$ und $\partial\mathbf{K}_{r_1, n}(\mathbf{p}_{M_{1,2}})$ keine Schnittpunkte mit dem Kreisrand $\partial\mathbf{K}_{r_2, n}(\mathbf{p}_{M_2})$.

Schnittpunkte zweier Kreisränder im \mathbb{R}^3

Gegeben seien im Folgenden zwei Kreise $\mathbf{K}_{r_1, n}(\mathbf{p}_{M_1})$ und $\mathbf{K}_{r_2, n}(\mathbf{p}_{M_2})$ nach Definition 8. Im Folgenden werden Formeln für die Berechnung von Schnittpunkten der Kreisränder $\partial\mathbf{K}_{r_1, n}(\mathbf{p}_{M_1})$ und $\partial\mathbf{K}_{r_2, n}(\mathbf{p}_{M_2})$ für $\mathbf{n} = \mathbf{e}_z = (0, 0, 1)^T$ hergeleitet. Die beiden Kreise und somit auch die beiden Kreisränder sind identisch genau dann, wenn

$$0 = |\mathbf{p}_{M_2} - \mathbf{p}_{M_1}| = r_2 - r_1 \quad (5.123)$$

gilt. Falls

$$0 = |\mathbf{p}_{M_2} - \mathbf{p}_{M_1}| \neq r_2 - r_1 \quad (5.124)$$

gilt, dann haben die beiden Kreisränder keinen Schnittpunkt. Im Folgenden gelte also immer:

$$0 \neq |\mathbf{p}_{M_2} - \mathbf{p}_{M_1}| \quad (5.125)$$

und ohne Beschränkung der Allgemeinheit

$$r_2 \geq r_1, \quad (5.126)$$

ansonsten werden einfach die Rollen der zwei Kreise vertauscht. Dann gilt: Keinen Schnittpunkt gibt es genau dann, wenn

$$0 \neq |\mathbf{p}_{M_2} - \mathbf{p}_{M_1}| < r_2 - r_1 \vee r_2 + r_1 < |\mathbf{p}_{M_2} - \mathbf{p}_{M_1}| \neq 0 \quad (5.127)$$

gilt (siehe Abbildung 5.21). Einen Schnittpunkt gibt es genau dann, wenn

$$0 \neq |\mathbf{p}_{M_2} - \mathbf{p}_{M_1}| = r_2 - r_1 \vee r_2 + r_1 = |\mathbf{p}_{M_2} - \mathbf{p}_{M_1}| \neq 0 \quad (5.128)$$

gilt (siehe Abbildung 5.22). Zwei Schnittpunkte gibt es genau dann, wenn

$$r_2 - r_1 < \underbrace{|\mathbf{p}_{M_2} - \mathbf{p}_{M_1}|}_{\neq 0} < r_2 + r_1 \quad (5.129)$$

gilt (siehe Abbildung 5.23). Im Folgenden gebe es mindestens einen Schnittpunkt. In jedem Fall

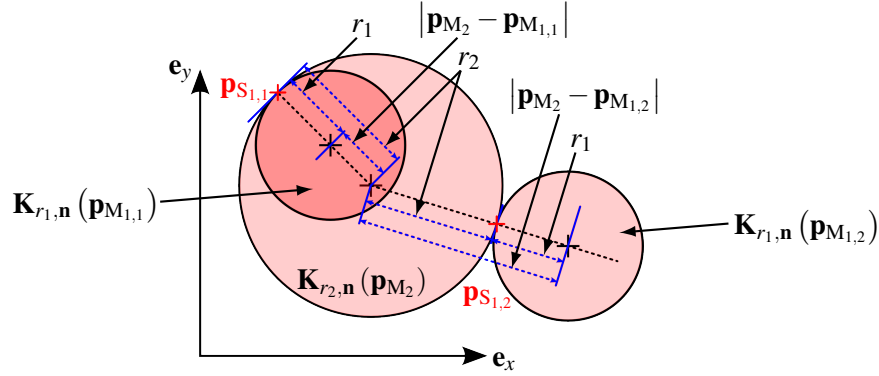


Abbildung 5.22: Drei Kreise $K_{r_2, n}(p_{M_2})$, $K_{r_1, n}(p_{M_{1,1}})$ und $K_{r_1, n}(p_{M_{1,2}})$ mit $r_2 \geq r_1$. $K_{r_2, n}(p_{M_2})$ und $K_{r_1, n}(p_{M_{1,1}})$ erfüllen die linke Bedingung in (5.128). $K_{r_2, n}(p_{M_2})$ und $K_{r_1, n}(p_{M_{1,2}})$ erfüllen die rechte Bedingung in (5.128). Hier haben die Kreisränder $\partial K_{r_1, n}(p_{M_{1,1}})$ und $\partial K_{r_1, n}(p_{M_{1,2}})$ jeweils einen Schnittpunkt $p_{s_{1,1}}$ bzw. $p_{s_{1,2}}$ (rot markiert) mit dem Kreisrand $\partial K_{r_2, n}(p_{M_2})$.

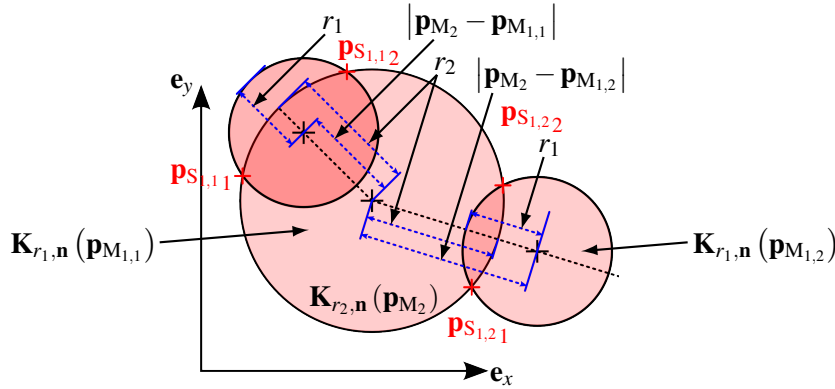


Abbildung 5.23: Drei Kreise $K_{r_2, n}(p_{M_2})$, $K_{r_1, n}(p_{M_{1,1}})$ und $K_{r_1, n}(p_{M_{1,2}})$ mit $r_2 \geq r_1$. Die Kreise erfüllen die Bedingung in (5.129). Der Kreisrand $\partial K_{r_1, n}(p_{M_{1,1}})$ hat die Schnittpunkte $p_{s_{1,1,1}}$ und $p_{s_{1,1,2}}$ mit dem Kreisrand $\partial K_{r_2, n}(p_{M_2})$. Der Kreisrand $\partial K_{r_1, n}(p_{M_{1,2}})$ hat die Schnittpunkte $p_{s_{1,2,1}}$ und $p_{s_{1,2,2}}$ mit dem Kreisrand $\partial K_{r_2, n}(p_{M_2})$. Die Schnittpunkte sind jeweils rot markiert.

müssen für einen Schnittpunkt $p_s = (x_{p_s}, y_{p_s})^T$ die zwei Bedingungen

$$(x_{p_s} - x_{p_{M_1}})^2 + (y_{p_s} - y_{p_{M_1}})^2 = r_1^2 \quad (5.130)$$

und

$$(x_{p_s} - x_{p_{M_2}})^2 + (y_{p_s} - y_{p_{M_2}})^2 = r_2^2 \quad (5.131)$$

erfüllt sein. Gleichung (5.130) ist äquivalent mit:

$$x_{p_s}^2 - 2x_{p_s}x_{p_{M_1}} + x_{p_{M_1}}^2 + y_{p_s}^2 - 2y_{p_s}y_{p_{M_1}} + y_{p_{M_1}}^2 - r_1^2 = 0. \quad (5.132)$$

Gleichung (5.131) ist äquivalent mit:

$$x_{p_s}^2 - 2x_{p_s}x_{p_{M_2}} + x_{p_{M_2}}^2 + y_{p_s}^2 - 2y_{p_s}y_{p_{M_2}} + y_{p_{M_2}}^2 - r_2^2 = 0. \quad (5.133)$$

Subtraktion der Gleichung (5.132) von der Gleichung (5.133) ergibt:

$$-2x_{\mathbf{p}_s} (x_{\mathbf{p}_{M_2}} - x_{\mathbf{p}_{M_1}}) - 2y_{\mathbf{p}_s} (y_{\mathbf{p}_{M_2}} - y_{\mathbf{p}_{M_1}}) + x_{\mathbf{p}_{M_2}}^2 + y_{\mathbf{p}_{M_2}}^2 - r_2^2 - x_{\mathbf{p}_{M_1}}^2 - y_{\mathbf{p}_{M_1}}^2 + r_1^2 = 0 \quad (5.134)$$

Auflösen der Gleichung (5.134) nach x_s liefert:

$$x_{\mathbf{p}_s} = \underbrace{-\frac{y_{\mathbf{p}_{M_2}} - y_{\mathbf{p}_{M_1}}}{x_{\mathbf{p}_{M_2}} - x_{\mathbf{p}_{M_1}}}}_{k_{y1}:=} y_{\mathbf{p}_s} + \underbrace{\frac{x_{\mathbf{p}_{M_2}}^2 + y_{\mathbf{p}_{M_2}}^2 - r_2^2 - x_{\mathbf{p}_{M_1}}^2 - y_{\mathbf{p}_{M_1}}^2 + r_1^2}{2(x_{\mathbf{p}_{M_2}} - x_{\mathbf{p}_{M_1}})}}_{k_{y2}:=}, \quad (5.135)$$

falls

$$|x_{\mathbf{p}_{M_2}} - x_{\mathbf{p}_{M_1}}| > 0. \quad (5.136)$$

Auflösen der Gleichung (5.134) nach y_s liefert:

$$y_{\mathbf{p}_s} = \underbrace{-\frac{x_{\mathbf{p}_{M_2}} - x_{\mathbf{p}_{M_1}}}{y_{\mathbf{p}_{M_2}} - y_{\mathbf{p}_{M_1}}}}_{k_{x1}:=} x_{\mathbf{p}_s} + \underbrace{\frac{x_{\mathbf{p}_{M_2}}^2 + y_{\mathbf{p}_{M_2}}^2 - r_2^2 - x_{\mathbf{p}_{M_1}}^2 - y_{\mathbf{p}_{M_1}}^2 + r_1^2}{2(y_{\mathbf{p}_{M_2}} - y_{\mathbf{p}_{M_1}})}}_{k_{x2}:=}, \quad (5.137)$$

falls

$$|y_{\mathbf{p}_{M_2}} - y_{\mathbf{p}_{M_1}}| > 0. \quad (5.138)$$

Wegen Gleichung (5.125) gilt mindestens eine der Gleichungen (5.136) und (5.138). Deshalb wird immer zunächst überprüft, ob die Bedingung

$$|y_{\mathbf{p}_{M_2}} - y_{\mathbf{p}_{M_1}}| \geq |x_{\mathbf{p}_{M_2}} - x_{\mathbf{p}_{M_1}}| \quad (5.139)$$

erfüllt ist. Ist (5.139) erfüllt, so liefert Einsetzen von (5.137) in (5.133):

$$\begin{aligned} & x_{\mathbf{p}_s}^2 - 2x_{\mathbf{p}_s}x_{\mathbf{p}_{M_2}} + x_{\mathbf{p}_{M_2}}^2 + (k_{x1}x_{\mathbf{p}_s} + k_{x2})^2 - 2(k_{x1}x_{\mathbf{p}_s} + k_{x2})y_{\mathbf{p}_{M_2}} + y_{\mathbf{p}_{M_2}}^2 - r_2^2 = 0 \\ \Leftrightarrow & \underbrace{(k_{x1}^2 + 1)}_{a_x:=} x_{\mathbf{p}_s}^2 + \underbrace{2(k_{x1}(k_{x2} - y_{\mathbf{p}_{M_2}}) - x_{\mathbf{p}_{M_2}})}_{b_x:=} x_{\mathbf{p}_s} + \underbrace{(k_{x2} - y_{\mathbf{p}_{M_2}})^2 + x_{\mathbf{p}_{M_2}}^2 - r_2^2}_{c_x:=} = 0. \end{aligned} \quad (5.140)$$

Ist (5.139) hingegen nicht erfüllt, so liefert Einsetzen von (5.135) in (5.133):

$$\begin{aligned} & (k_{y1}y_{\mathbf{p}_s} + k_{y2})^2 - 2(k_{y1}y_{\mathbf{p}_s} + k_{y2})x_{\mathbf{p}_{M_2}} + x_{\mathbf{p}_{M_2}}^2 + y_{\mathbf{p}_s}^2 - 2y_{\mathbf{p}_s}y_{\mathbf{p}_{M_2}} + y_{\mathbf{p}_{M_2}}^2 - r_2^2 = 0 \\ \Leftrightarrow & \underbrace{(k_{y1}^2 + 1)}_{a_y:=} y_{\mathbf{p}_s}^2 + \underbrace{2(k_{y1}(k_{y2} - x_{\mathbf{p}_{M_2}}) - y_{\mathbf{p}_{M_2}})}_{b_y:=} y_{\mathbf{p}_s} + \underbrace{(k_{y2} - x_{\mathbf{p}_{M_2}})^2 + y_{\mathbf{p}_{M_2}}^2 - r_2^2}_{c_y:=} = 0. \end{aligned} \quad (5.141)$$

Es ergibt sich also entweder eine quadratische Gleichung in $x_{\mathbf{p}_s}$ von der Form:

$$a_x x_{\mathbf{p}_s}^2 + b_x x_{\mathbf{p}_s} + c_x = 0, \quad (5.142)$$

oder eine quadratische Gleichung in $y_{\mathbf{p}_s}$ von der Form:

$$a_y y_{\mathbf{p}_s}^2 + b_y y_{\mathbf{p}_s} + c_y = 0. \quad (5.143)$$

Die Lösungsformel für quadratische Gleichungen ergibt nach Bronstein et al. (2008):

$$x_{\mathbf{p}_{s1/2}} = \frac{-b_x \pm \sqrt{b_x^2 - 4a_x c_x}}{2a_x}, \quad (5.144)$$

bzw.

$$y_{\mathbf{p}_{s1/2}} = \frac{-b_y \pm \sqrt{b_y^2 - 4a_y c_y}}{2a_y}, \quad (5.145)$$

wobei nach Definition

$$a_x \geq 1 \text{ bzw. } a_y \geq 1 \quad (5.146)$$

gilt. Sei

$$D_x := b_x^2 - 4a_x c_x \text{ bzw. } D_y := b_y^2 - 4a_y c_y, \quad (5.147)$$

dann gibt es zwei Lösungen $x_{\mathbf{p}_{s1}}$ und $x_{\mathbf{p}_{s2}}$ bzw. $y_{\mathbf{p}_{s1}}$ und $y_{\mathbf{p}_{s2}}$, falls

$$D_x > 0 \text{ bzw. } D_y > 0 \quad (5.148)$$

gilt. Einsetzen von $x_{\mathbf{p}_{s1}}$ und $x_{\mathbf{p}_{s2}}$ in Gleichung (5.137) bzw. von $y_{\mathbf{p}_{s1}}$ und $y_{\mathbf{p}_{s2}}$ in Gleichung (5.135) liefert dann $y_{\mathbf{p}_{s1}}$ und $y_{\mathbf{p}_{s2}}$ bzw. $x_{\mathbf{p}_{s1}}$ und $x_{\mathbf{p}_{s2}}$. Eine Lösung $x_{\mathbf{p}_s}$ bzw. $y_{\mathbf{p}_s}$ gibt es, falls

$$D_x = 0 \text{ bzw. } D_y = 0 \quad (5.149)$$

gilt. Einsetzen von $x_{\mathbf{p}_s}$ in Gleichung (5.137) bzw. von $y_{\mathbf{p}_s}$ in Gleichung (5.135) liefert dann $y_{\mathbf{p}_s}$ bzw. $x_{\mathbf{p}_s}$. Falls

$$D_x < 0 \text{ bzw. } D_y < 0 \quad (5.150)$$

gilt, dann gibt es keine Lösung.

Einschränkung auf Kreisbögen

Seien $\mathbf{K}_{\text{Sek}}(\mathbf{p}_i, \mathbf{p}_{i+1}) \subset \mathbf{K}_{r,n}(\mathbf{p}_{M_i})$ und $\mathbf{K}_{\text{Bog}}(\mathbf{p}_i, \mathbf{p}_{i+1}) \subset \partial \mathbf{K}_{r,n}(\mathbf{p}_{M_i})$ ein Kreissektor und ein Kreisbogen nach Definition 9. Falls für einen Punkt \mathbf{p}

$$\mathbf{p} \in \mathbf{K}_{\text{Bog}}(\mathbf{p}_i, \mathbf{p}_{i+1}) \quad (5.151)$$

gilt, dann liegt er sowohl auf dem Kreisrand, als auch auf dem Kreisbogen. Gilt hingegen

$$\mathbf{p} \in \partial \mathbf{K}_{r,n}(\mathbf{p}_{M_i}) \setminus \mathbf{K}_{\text{Bog}}(\mathbf{p}_i, \mathbf{p}_{i+1}) \quad (5.152)$$

dann liegt der Punkt zwar auf dem Kreisrand, aber nicht auf dem Kreisbogen (siehe Abbildung 5.24). Für ein Segment und einen Kreisbogen gilt, dass ein Schnittpunkt des Segments und des entsprechenden Kreisrandes nur ein Schnittpunkt des Segments und des Kreisbogens ist, falls der entsprechende Schnittpunkt auch auf dem Kreisbogen liegt. Auch für zwei Kreisbögen gilt, dass die Schnittpunkte der entsprechenden Kreisränder nur Schnittpunkte der Kreisbögen sind, falls der jeweilige Schnittpunkt auf beiden Kreisbögen liegt.

Seien

$$\mathbf{E}_H(\mathbf{n}_i, \mathbf{p}_{M_i}) \quad (5.153)$$

und

$$\mathbf{E}_H(\mathbf{n}_{i+1}, \mathbf{p}_{M_i}) \quad (5.154)$$

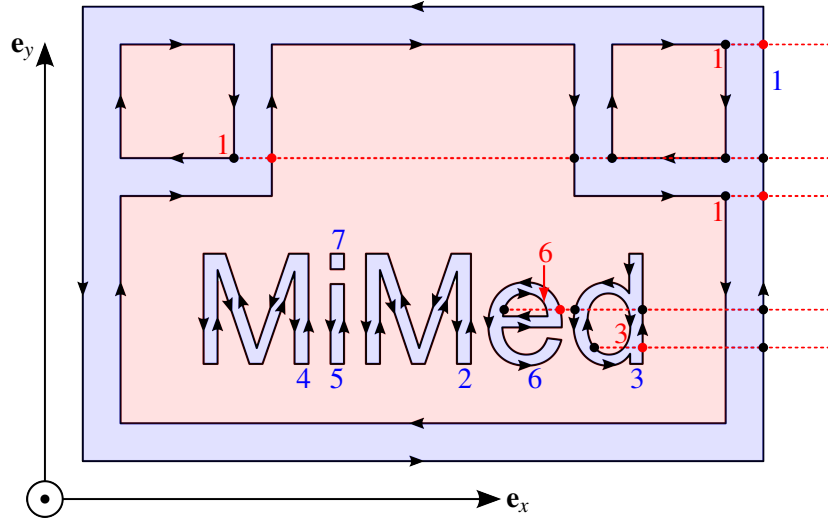


Abbildung 5.25: Zuweisung von Identifikationsnummern (IDs) an die einzelnen geschlossenen Kurven: zunächst werden allen Konturen gegen den Uhrzeigersinn (äußere Konturen) IDs zugewiesen. Jede Kontur im Uhrzeigersinn (innere Konturen) bekommt dann die ID derjenigen äußeren Kontur zugewiesen, in der sie liegt. Dazu werden ausgehend von einem Punkt \mathbf{p}_0 auf der inneren Kontur die Schnittpunkte des Strahls $\mathbf{R} := \{\mathbf{p}_0 + \lambda \mathbf{e}_x \mid 0 \leq \lambda \in \mathbb{R}\}$ mit allen äußeren Konturen betrachtet. Die innere Kontur bekommt dann die ID derjenigen äußeren Kontur zugewiesen, auf der der Schnittpunkt mit dem kleinsten λ liegt. Im Beispiel bekommt die innere Kontur des Buchstabens „d“ die ID 3 zugewiesen und nicht die ID 1.

Beweis: Seien $\alpha, r, h, s \in \mathbb{R}$ wie in Abbildung 5.26. Für die Fläche eines ganzen Kreises mit Radius r gilt nach Barth et al. (1998):

$$A_K = r^2 \pi. \quad (5.160)$$

Für die Fläche eines Kreissektors mit demselben Radius r und dem Mittelpunktswinkel α gilt entsprechend:

$$A_{\text{KSek}} = \frac{\alpha}{2\pi} r^2 \pi = 0.5 \alpha r^2. \quad (5.161)$$

Für die Fläche des Dreiecks mit der Höhe h gilt:

$$A_{\Delta} = 0.5 s h. \quad (5.162)$$

h lässt sich berechnen durch:

$$\frac{h}{r} = \cos \frac{\alpha}{2} \Leftrightarrow h = r \cos \frac{\alpha}{2}. \quad (5.163)$$

s lässt sich berechnen durch:

$$\frac{0.5s}{r} = \sin \frac{\alpha}{2} \Leftrightarrow 0.5s = r \sin \frac{\alpha}{2}. \quad (5.164)$$

Mit (5.163) und (5.164) folgt:

$$A_{\Delta} = r^2 \sin \frac{\alpha}{2} \cos \frac{\alpha}{2} \quad (5.165)$$

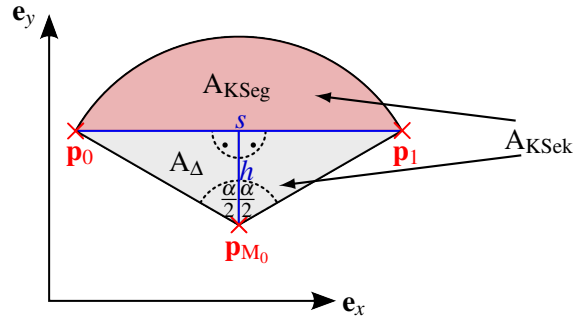


Abbildung 5.26: Kreissektor nach Definition 9 mit Mittelpunktswinkel α und Radius r . Kreismittelpunkt: $\mathbf{p}_{M_0} \in \mathbb{R}^2$. Endpunkte des Kreisbogens: $\mathbf{p}_0 \in \mathbb{R}^2$ und $\mathbf{p}_1 \in \mathbb{R}^2$. Gesucht ist die rötliche Fläche des Kreissegments: A_{KSeg} . Gesamtfläche des Kreissektors: A_{KSek} . Gesamtfläche der zwei grünen Dreiecke: A_{Δ} . Es gilt: $A_{KSeg} = A_{KSek} - A_{\Delta}$.

Nach Barth et al. (1998) gilt:

$$\sin 2\alpha = 2 \sin \alpha \cos \alpha. \quad (5.166)$$

Somit folgt aus (5.165) mit (5.166):

$$A_{\Delta} = 0.5r^2 \sin \alpha. \quad (5.167)$$

Letztlich folgt für die Fläche A_{KSeg} des Kreissegments:

$$A_{KSeg} = A_{KSek} - A_{\Delta} \stackrel{(5.167), (5.161)}{=} 0.5r^2 (\alpha - \sin \alpha). \quad (5.168)$$

q.e.d.

Satz 16 (Flächenberechnungen für Segmente):

Sei $\mathbf{S}(\mathbf{p}_i, \mathbf{p}_{i+1})$ ein Segment mit $z_{\mathbf{p}_i} = z_{\mathbf{p}_{i+1}}$. Für den Flächeninhalt zwischen dem Segment und der \mathbf{e}_x -Achse gilt dann:

$$A_{\mathbf{S}(\mathbf{p}_i, \mathbf{p}_{i+1})} = (x_{\mathbf{p}_{i+1}} - x_{\mathbf{p}_i}) (y_{\mathbf{p}_{i+1}} + y_{\mathbf{p}_i}) 0.5. \quad (5.169)$$

Insbesondere ist der Flächeninhalt also abhängig vom Vorzeichen von $x_{\mathbf{p}_{i+1}} - x_{\mathbf{p}_i}$, das heißt, ob das Segment in positive oder negative \mathbf{e}_x -Richtung läuft.

Beweis: Siehe Abbildung 5.27.

q.e.d.

Satz 17 (Flächenberechnungen für geschlossene Kurve aus Segmenten):

Seien $\mathbf{S}(\mathbf{p}_i, \mathbf{p}_{i+1})$, $0 \leq i < n$, n Segmente mit $z_{\mathbf{p}_i} = z_{\mathbf{p}_{i+1}} \forall i$, die eine geschlossene Kurve $\gamma_{\mathbf{n}}$ ($\mathbf{n} = (0, 0, 1)^T = \mathbf{e}_z$) nach Definition 14 ohne Überschneidungen ergeben. Für den Flächeninhalt von $\gamma_{\mathbf{e}_z}$ gilt dann:

$$A_{\gamma_{\mathbf{e}_z}} = \sum_{i=0}^{n-1} [(x_{\mathbf{p}_{i+1}} - x_{\mathbf{p}_i}) (y_{\mathbf{p}_{i+1}} + y_{\mathbf{p}_i}) 0.5]. \quad (5.170)$$

Beweis: Flächen für jedes Segment nach Satz 16 aufsummiert.

q.e.d.

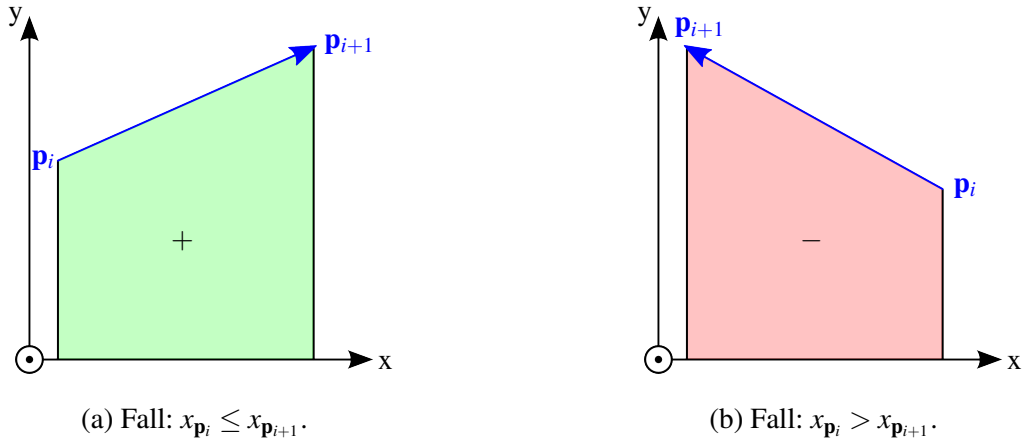


Abbildung 5.27: Flächenberechnungen für Segmente: es wird die Fläche zwischen einem Segment und der e_x -Achse berechnet. Die grüne Fläche zeigt, dass das Segment in positive e_x -Richtung läuft. Die rote Fläche zeigt, dass das Segment in negative e_x -Richtung läuft. Siehe Satz 16.

Satz 18 (Prüfung der Umlaufrichtung einer geschlossenen Kurve aus Segmenten):

Seien sämtliche Voraussetzungen von Satz 17 erfüllt, dann gilt:

$$\gamma_{e_z} \text{ läuft im Uhrzeigersinn} \Leftrightarrow A_{\gamma_{e_z}} > 0, \quad (5.171)$$

$$\gamma_{e_z} \text{ läuft gegen den Uhrzeigersinn} \Leftrightarrow A_{\gamma_{e_z}} < 0. \quad (5.172)$$

Beweis: Siehe Satz 17 und Abbildung 5.28.

q.e.d.

Satz 19 (Flächenberechnungen für Kreisbögen):

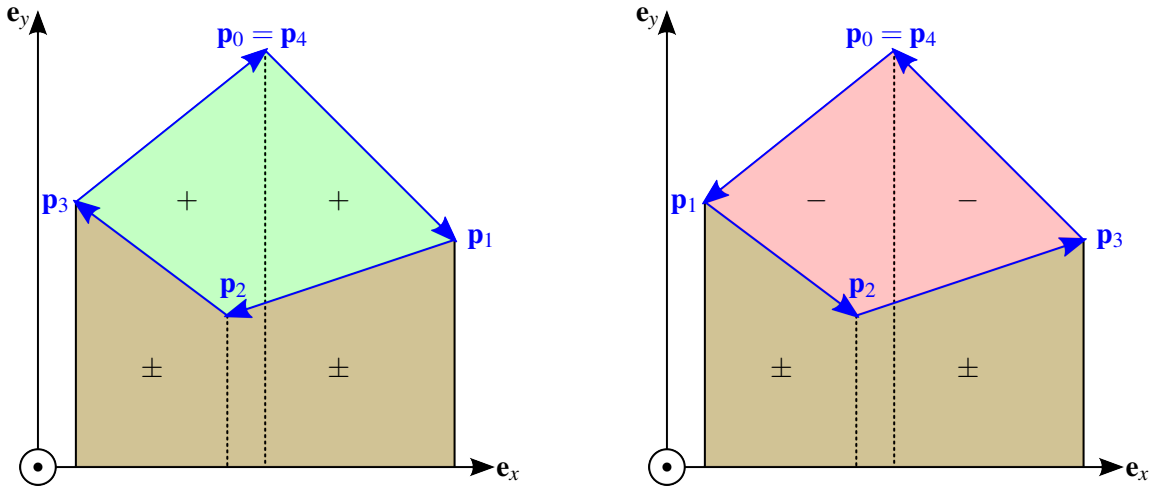
Sei $\mathbf{K}_{\text{Bog}}(\mathbf{p}_i, \mathbf{p}_{i+1}) \subset \partial \mathbf{K}_{r,n}(\mathbf{p}_{M_i})$ mit $z_{\mathbf{p}_i} = z_{\mathbf{p}_{i+1}}$ ein Kreisbogen nach Definition 9 mit einem Mittelpunktswinkel $0 < \alpha < \pi$. Seien weiter

$$\mathbf{p}_{i,i+1}^{\text{li}} \in \{ \mathbf{p} \in \partial \mathbf{K}_{r,n}(\mathbf{p}_{M_i}) \mid x_{\mathbf{p}} \leq x_{\mathbf{p}'} \forall \mathbf{p}' \in \partial \mathbf{K}_{r,n}(\mathbf{p}_{M_i}) \} \quad (5.173)$$

der „linkeste“ und

$$\mathbf{p}_{i,i+1}^{\text{re}} \in \{ \mathbf{p} \in \partial \mathbf{K}_{r,n}(\mathbf{p}_{M_i}) \mid x_{\mathbf{p}} \geq x_{\mathbf{p}'} \forall \mathbf{p}' \in \partial \mathbf{K}_{r,n}(\mathbf{p}_{M_i}) \} \quad (5.174)$$

der „rechteste“ Punkt auf dem Kreisrand $\partial \mathbf{K}_{r,n}(\mathbf{p}_{M_i})$. Für den Flächeninhalt zwischen dem



(a) Geschlossene Kurve im Uhrzeigersinn: der Flächeninhalt (grün) der Kurve ist positiv. (b) Geschlossene Kurve gegen den Uhrzeigersinn: der Flächeninhalt (rot) der Kurve ist negativ.

Abbildung 5.28: Flächenberechnungen für geschlossene Kurven (blau), welche nur aus Segmenten bestehen: es wird jeweils die Fläche zwischen einem Segment und der e_x -Achse berechnet. Die grüne Fläche gehört nur zu Segmenten, die in positive e_x -Richtung laufen. Die rote Fläche gehört nur zu Segmenten, die in negative e_x -Richtung laufen. Die braune Fläche gehört sowohl zu Segmenten, die in positive e_x -Richtung laufen als auch zu solchen, die in negative e_x -Richtung laufen. Die braune Fläche wird sozusagen doppelt, jedoch mit unterschiedlichen Vorzeichen, gewertet.

Kreisbogen und der e_x -Achse gilt dann (siehe Abbildung 5.29):

$$A_{\mathbf{K}_{\text{Bog}}(\mathbf{p}_i, \mathbf{p}_{i+1})} = \begin{cases} \left(\begin{array}{l} (x_{\mathbf{p}_{i+1}} - x_{\mathbf{p}_i}) (y_{\mathbf{p}_{i+1}} + y_{\mathbf{p}_i}) 0.5 \\ + 0.5r^2 (\alpha - \sin \alpha) \end{array} \right) & \begin{array}{l} (y_{\mathbf{p}_i} \geq y_{\mathbf{p}_{M_i}} \wedge y_{\mathbf{p}_{i+1}} \geq y_{\mathbf{p}_{M_i}}) \\ \vee (y_{\mathbf{p}_i} < y_{\mathbf{p}_{M_i}} \wedge y_{\mathbf{p}_{i+1}} < y_{\mathbf{p}_{M_i}}) \end{array} \\ \left(\begin{array}{l} (x_{\mathbf{p}_{i,i+1}^{\text{re}}} - x_{\mathbf{p}_i}) (y_{\mathbf{p}_i} + y_{\mathbf{p}_{i,i+1}^{\text{re}}}) 0.5 \\ + 0.5r^2 (\alpha_1 - \sin \alpha_1) \\ + (x_{\mathbf{p}_{i+1}} - x_{\mathbf{p}_{i,i+1}^{\text{re}}}) (y_{\mathbf{p}_{i+1}} + y_{\mathbf{p}_{i,i+1}^{\text{re}}}) 0.5 \\ + 0.5r^2 (\alpha_2 - \sin \alpha_2) \end{array} \right) & y_{\mathbf{p}_i} \geq y_{\mathbf{p}_{M_i}} \wedge y_{\mathbf{p}_{i+1}} < y_{\mathbf{p}_{M_i}} \\ \left(\begin{array}{l} (x_{\mathbf{p}_{i,i+1}^{\text{li}}} - x_{\mathbf{p}_i}) (y_{\mathbf{p}_i} + y_{\mathbf{p}_{i,i+1}^{\text{li}}}) 0.5 \\ + 0.5r^2 (\alpha_1 - \sin \alpha_1) \\ + (x_{\mathbf{p}_{i+1}} - x_{\mathbf{p}_{i,i+1}^{\text{li}}}) (y_{\mathbf{p}_{i+1}} + y_{\mathbf{p}_{i,i+1}^{\text{li}}}) 0.5 \\ + 0.5r^2 (\alpha_2 - \sin \alpha_2) \end{array} \right) & y_{\mathbf{p}_i} < y_{\mathbf{p}_{M_i}} \wedge y_{\mathbf{p}_{i+1}} \geq y_{\mathbf{p}_{M_i}} \end{cases} \quad (5.175)$$

Insbesondere ist der Flächeninhalt also abhängig vom Vorzeichen der Differenzen von $x_{\mathbf{p}_{i+1}} - x_{\mathbf{p}_i}$, $x_{\mathbf{p}_{i,i+1}^{\text{re}}} - x_{\mathbf{p}_i}$ (≥ 0), $x_{\mathbf{p}_{i+1}} - x_{\mathbf{p}_{i,i+1}^{\text{re}}}$ (< 0), $x_{\mathbf{p}_{i,i+1}^{\text{li}}} - x_{\mathbf{p}_i}$ (< 0) und $x_{\mathbf{p}_{i+1}} - x_{\mathbf{p}_{i,i+1}^{\text{li}}}$ (≥ 0), das heißt, ob der Kreisbogen oder Teil des Kreisbogens in positive oder negative e_x -Richtung läuft.

Beweis: Siehe Abbildung 5.29.

q.e.d.

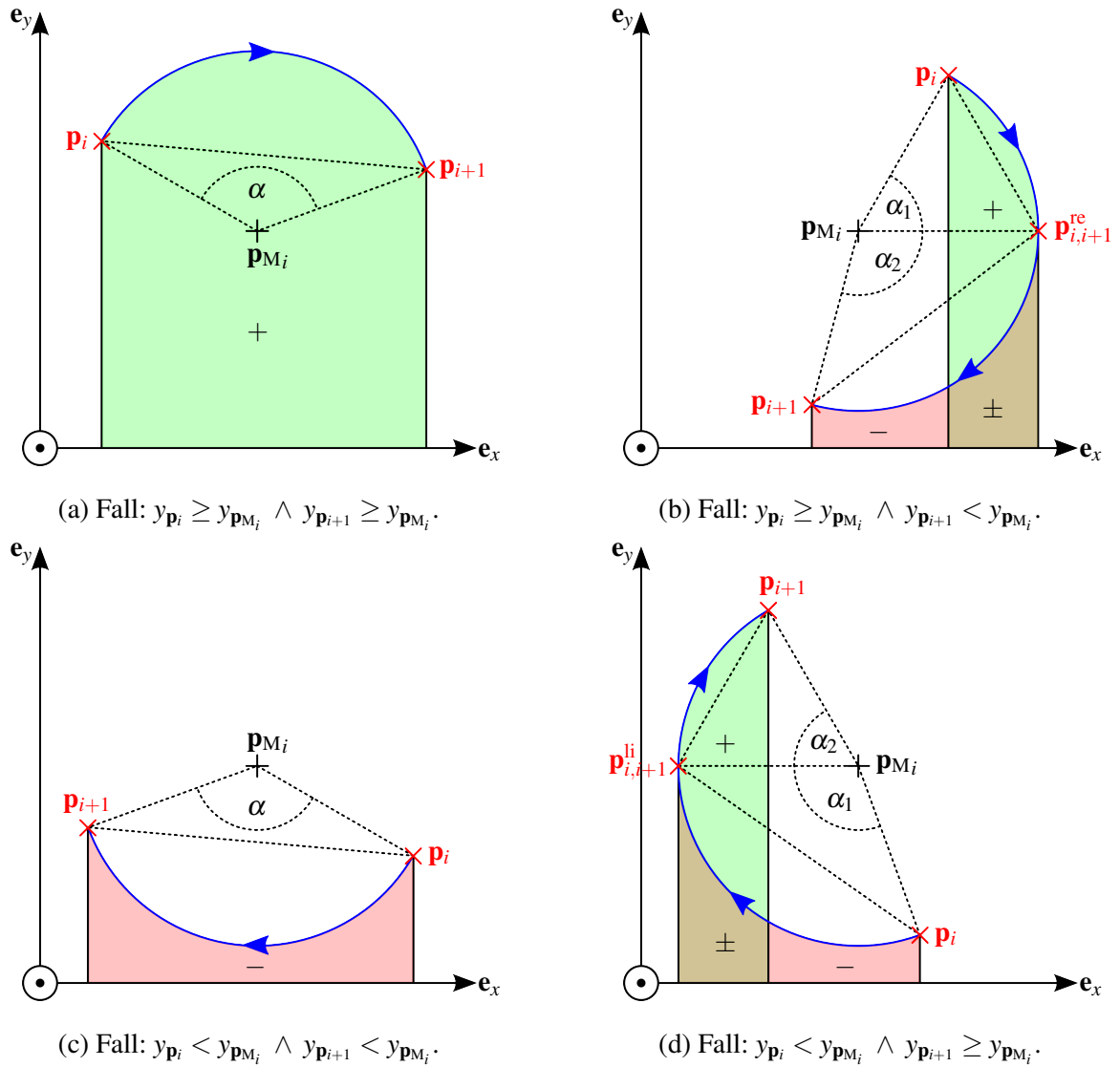
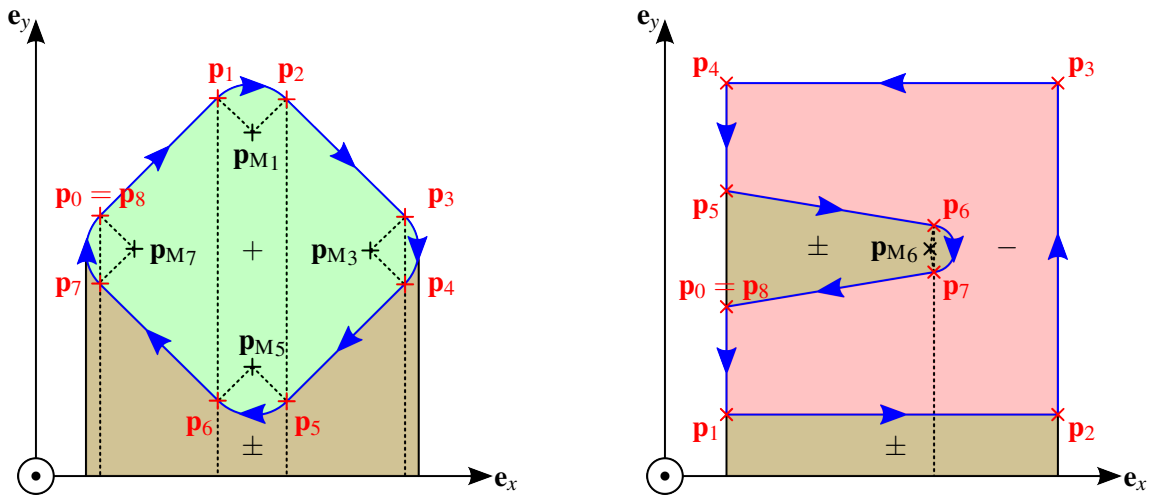


Abbildung 5.29: Flächenberechnungen für Kreisbögen: es wird die Fläche zwischen einem Bogen und der e_x -Achse berechnet. Berücksichtigt werden muss dabei, ob ein Bogen in positive oder in negative e_x -Richtung läuft. Da $0 < \alpha < \pi$ gilt, müssen vier Fälle unterschieden werden. Die grüne Fläche gehört nur zu Bögen, die in positive e_x -Richtung laufen. Die rote Fläche gehört nur zu Bögen, die in negative e_x -Richtung laufen. Die braune Fläche gehört sowohl zu Bögen, die in positive e_x -Richtung laufen als auch zu solchen, die in negative e_x -Richtung laufen. Die braune Fläche wird sozusagen doppelt, jedoch mit unterschiedlichen Vorzeichen, gewertet.

Satz 20 (Flächenberechnungen für geschlossene Kurve aus Segmenten und Kreisbögen): Seien $\mathbf{E}(p_i, p_{i+1})$, $0 \leq i < n$, n geometrische Elemente mit $z_{p_i} = z_{p_{i+1}} \forall i$, die eine geschlossene Kurve $\gamma_{\mathbf{n}}$ ($\mathbf{n} = (0, 0, 1)^T = \mathbf{e}_z$) nach Definition 14 ohne Überschneidungen ergeben, das heißt, es gilt $\forall i$ entweder $\mathbf{E}(p_i, p_{i+1}) = \mathbf{S}(p_i, p_{i+1})$ oder $\mathbf{E}(p_i, p_{i+1}) = \mathbf{K}_{\text{Bog}}(p_i, p_{i+1})$. Für den Flächeninhalt von $\gamma_{\mathbf{e}_z}$ gilt dann:

$$A_{\gamma_{\mathbf{e}_z}} = \sum_{i=0}^{n-1} A_{\mathbf{E}(p_i, p_{i+1})}, \quad (5.176)$$



(a) Geschlossene Kurve aus Segmenten und Bögen im Uhrzeigersinn: der Flächeninhalt (grün) der Kurve ist positiv.

(b) Geschlossene Kurve aus Segmenten und Bögen gegen den Uhrzeigersinn: der Flächeninhalt (rot) der Kurve ist negativ.

Abbildung 5.30: Flächenberechnungen für geschlossene Kurven (blau), welche aus Segmenten und Bögen bestehen. Die einzelnen Flächen werden bei Segmenten wie in Abbildung 5.27 summiert, bei Bögen wie in Abbildung 5.29. Braune Flächen gleichen sich dabei zu 0 aus, wie bereits in den Abbildungen 5.28 und 5.29 zu sehen ist. Grüne Fläche ist der positive Flächeninhalt von im Uhrzeigersinn verlaufenden Kurven. Rote Fläche ist der negative Flächeninhalt von gegen den Uhrzeigersinn verlaufenden Kurven.

wobei

$$A_{\mathbf{E}(\mathbf{p}_i, \mathbf{p}_{i+1})} := \begin{cases} A_{\mathbf{S}(\mathbf{p}_i, \mathbf{p}_{i+1})} & \mathbf{E}(\mathbf{p}_i, \mathbf{p}_{i+1}) = \mathbf{S}(\mathbf{p}_i, \mathbf{p}_{i+1}) \\ A_{\mathbf{K}_{\text{Bog}}(\mathbf{p}_i, \mathbf{p}_{i+1})} & \mathbf{E}(\mathbf{p}_i, \mathbf{p}_{i+1}) = \mathbf{K}_{\text{Bog}}(\mathbf{p}_i, \mathbf{p}_{i+1}) \end{cases} \quad (5.177)$$

Beweis: Flächen für jedes geometrische Element nach Satz 16 und Satz 19 aufsummiert. q.e.d.

Satz 21 (Prüfung der Umlaufrichtung einer geschlossenen Kurve aus Segmenten und Kreisbögen):

Seien sämtliche Voraussetzungen von Satz 20 erfüllt, dann gilt:

$$\gamma_{\mathbf{e}_z} \text{ läuft im Uhrzeigersinn} \Leftrightarrow A_{\gamma_{\mathbf{e}_z}} > 0, \quad (5.178)$$

$$\gamma_{\mathbf{e}_z} \text{ läuft gegen den Uhrzeigersinn} \Leftrightarrow A_{\gamma_{\mathbf{e}_z}} < 0. \quad (5.179)$$

Beweis: Siehe Satz 20 und Abbildung 5.30. q.e.d.

5.2.6 Windungszahlberechnung

In diesem Abschnitt wird erläutert wie die Windungszahl $\eta(\gamma_{\mathbf{n}}, \mathbf{p})$ (siehe Definition 16) einer geschlossenen Kurve $\gamma_{\mathbf{n}}$ (siehe Definition 14) im Punkt $\mathbf{p} = (x_{\mathbf{p}}, y_{\mathbf{p}}, z_{\mathbf{p}})^T \notin \gamma_{\mathbf{n}}$ berechnet wird. Hierzu wird zunächst der Fall betrachtet, dass $\gamma_{\mathbf{n}}$ nur aus Segmenten $\mathbf{S}(\mathbf{p}_i, \mathbf{p}_{i+1})$, $i \in$

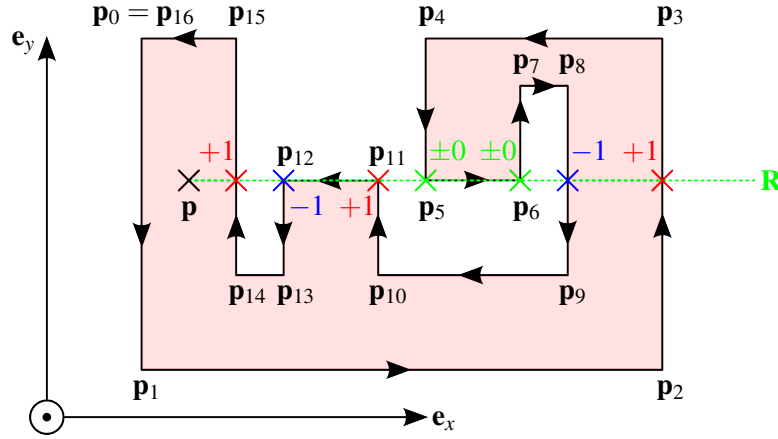


Abbildung 5.31: Beispiel für Punkt-in-Polygon-Test nach Jordan für eine geschlossene Kurve γ_{e_z} nach Definition 14, die allerdings nur aus Segmenten besteht. Idee dahinter ist die Berechnung der Windungszahl $\eta(\gamma_{e_z}, \mathbf{p})$ im Punkt $\mathbf{p} = (x_{\mathbf{p}}, y_{\mathbf{p}}, z_{\mathbf{p}})^T \notin \gamma_{e_z}$ durch folgende Vorgehensweise: Es werden alle Segmente betrachtet, die ausgehend vom Punkt \mathbf{p} einen Schnittpunkt mit dem Strahl $\mathbf{R} := \{\mathbf{p} + \lambda \mathbf{e}_x \mid 0 \leq \lambda \in \mathbb{R}\}$ haben. Die Schnittpunkte der einzelnen Segmente $\mathbf{S}(\mathbf{p}_i, \mathbf{p}_{i+1})$ mit \mathbf{R} müssen nicht explizit berechnet werden: es werden nur Schnittpunkte berücksichtigt, wenn entweder $y_{\mathbf{p}_i} < y_{\mathbf{p}} \leq y_{\mathbf{p}_{i+1}}$ (Fall 1) oder $y_{\mathbf{p}_{i+1}} < y_{\mathbf{p}} \leq y_{\mathbf{p}_i}$ (Fall 2) gilt. Ausgehend von einer Windungszahl $\eta(\gamma_{e_z}, \mathbf{p}) = 0$ wird zu dieser bei Fall 1 eine 1 addiert, bei Fall 2 hingegen eine 1 subtrahiert. Fall 1 gilt im Beispiel für $\mathbf{S}(\mathbf{p}_2, \mathbf{p}_3)$, $\mathbf{S}(\mathbf{p}_{10}, \mathbf{p}_{11})$ und $\mathbf{S}(\mathbf{p}_{14}, \mathbf{p}_{15})$, Fall 2 gilt für $\mathbf{S}(\mathbf{p}_8, \mathbf{p}_9)$ und $\mathbf{S}(\mathbf{p}_{12}, \mathbf{p}_{13})$. Es ergibt sich also insgesamt: $\eta(\gamma_{e_z}, \mathbf{p}) = 1 - 1 + 1 - 1 + 1 = 1$. Anmerkung: Die Segmente $\mathbf{S}(\mathbf{p}_4, \mathbf{p}_5)$, $\mathbf{S}(\mathbf{p}_5, \mathbf{p}_6)$, $\mathbf{S}(\mathbf{p}_6, \mathbf{p}_7)$ und $\mathbf{S}(\mathbf{p}_{11}, \mathbf{p}_{12})$ genügen weder Fall 1 noch Fall 2 und werden daher nicht explizit betrachtet. Wie man sieht, funktioniert die Berechnung also auch korrekt, falls ein Segment $\mathbf{S}(\mathbf{p}_i, \mathbf{p}_{i+1})$ auf dem Strahl \mathbf{R} liegt.

$\{0, 1, \dots, n-1\}$, besteht. In diesem Fall handelt es sich um den sogenannten Punkt-in-Polygon-Test nach Jordan, der in Abbildung 5.31 illustriert ist. Hierbei wird ausgehend vom Punkt \mathbf{p} ein Strahl $\mathbf{R} := \{\mathbf{p} + \lambda \mathbf{e}_x \mid 0 \leq \lambda \in \mathbb{R}\}$ in Richtung der positiven \mathbf{e}_x -Achse betrachtet. Wandert man nun auf dem Strahl \mathbf{R} mit zunehmendem λ , dann werden Segmente $\mathbf{S}(\mathbf{p}_i, \mathbf{p}_{i+1})$, die den Strahl von rechts nach links schneiden (Fall 1: $y_{\mathbf{p}_i} < y_{\mathbf{p}} \leq y_{\mathbf{p}_{i+1}}$) mit +1 gewichtet, Segmente $\mathbf{S}(\mathbf{p}_i, \mathbf{p}_{i+1})$, die den Strahl von links nach rechts schneiden (Fall 2: $y_{\mathbf{p}_{i+1}} < y_{\mathbf{p}} \leq y_{\mathbf{p}_i}$), werden hingegen mit einer -1 gewichtet. Die Summe ergibt dann die Windungszahl $\eta(\gamma_{\mathbf{n}}, \mathbf{p})$. Wie in Abbildung 5.31 zu sehen ist, sind damit auch die sogenannten Grenzfälle abgedeckt. Es gilt folgender Satz:

Satz 22 (Windungszahl für geschlossene Kurven aus Segmenten):

Sei $\gamma_{\mathbf{n}}$ eine geschlossene Kurve nach Definition 14, welche nur aus Segmenten $\mathbf{S}(\mathbf{p}_i, \mathbf{p}_{i+1})$, $i \in \{0, 1, \dots, n-1\}$, besteht. Seien weiter der Punkt $\mathbf{p} \notin \gamma_{\mathbf{n}}$ und der Vektor

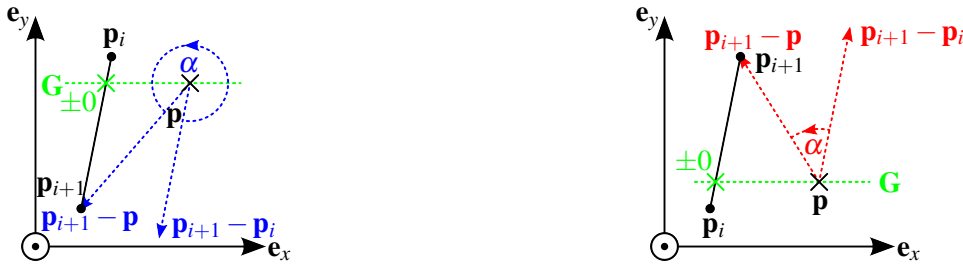
$$\mathbf{n}_i := (\mathbf{p}_{i+1} - \mathbf{p}_i) \times (\mathbf{p}_{i+1} - \mathbf{p}) \quad (5.180)$$

gegeben, wobei \mathbf{n}_i , genau so wie \mathbf{n} , senkrecht auf der von $\gamma_{\mathbf{n}}$ gebildeten Ebene steht, d.h. es gilt:

$$\mathbf{n}_i = \mu \cdot \mathbf{n}, \mu \in \mathbb{R}. \quad (5.181)$$



- (a) Fall $(y_{p_i} < y_p \leq y_{p_{i+1}}) \wedge (\mu \leq 0)$: $\eta(\gamma_n, \mathbf{p})$ muss nach Formel (5.182) um +1 erhöht werden. Dies stimmt mit der Zeichnung überein, da $\mathbf{p}_s \in \mathbf{R}$ gilt.
- (b) Fall $(y_{p_{i+1}} < y_p \leq y_{p_i}) \wedge (\mu > 0)$: $\eta(\gamma_n, \mathbf{p})$ muss nach Formel (5.182) um -1 verringert werden. Dies stimmt mit der Zeichnung überein, da $\mathbf{p}_s \in \mathbf{R}$ gilt.



- (c) Fall $(y_{p_{i+1}} < y_p \leq y_{p_i}) \wedge (\mu \leq 0)$: nach Formel (5.182) muss nichts getan werden. Dies stimmt mit der Zeichnung überein, da $\mathbf{p}_s \in \mathbf{G} \setminus \mathbf{R}$ gilt.
- (d) Fall $(y_{p_i} < y_p \leq y_{p_{i+1}}) \wedge (\mu > 0)$: nach Formel (5.182) muss nichts getan werden. Dies stimmt mit der Zeichnung überein, da $\mathbf{p}_s \in \mathbf{G} \setminus \mathbf{R}$ gilt.

Abbildung 5.32: Auswirkungen auf die Windungszahl $\eta(\gamma_n, \mathbf{p})$ im Punkt $\mathbf{p} = (x_p, y_p, z_p)^T \notin \gamma_n$ in Abhängigkeit der Lage eines Segments $\mathbf{S}(\mathbf{p}_i, \mathbf{p}_{i+1})$. $\mathbf{G} := \{\mathbf{p} + \lambda \mathbf{e}_x \mid \lambda \in \mathbb{R}\}$ bezeichnet die durch \mathbf{p} verlaufende Gerade parallel zur \mathbf{e}_x -Achse. Mit \mathbf{p}_s wird der Schnittpunkt der Gerade \mathbf{G} mit dem Segment $\mathbf{S}(\mathbf{p}_i, \mathbf{p}_{i+1})$ bezeichnet. Weiter sei $\mathbf{R} := \{\mathbf{p} + \lambda \mathbf{e}_x \mid 0 \leq \lambda \in \mathbb{R}\}$ ein Strahl in Richtung der positiven \mathbf{e}_x -Achse. μ ist wie in Gleichung (5.181) zu sehen ist, bestimmt.

Die Windungszahl $\eta(\gamma_n, \mathbf{p})$ im Punkt \mathbf{p} lässt sich dann berechnen durch:

$$\eta(\gamma_n, \mathbf{p}) = \sum_{i=0}^{n-1} \begin{cases} 1 & (y_{p_i} < y_p \leq y_{p_{i+1}}) \wedge (\mu \leq 0) \\ -1 & (y_{p_{i+1}} < y_p \leq y_{p_i}) \wedge (\mu > 0) \end{cases}. \quad (5.182)$$

Beweis: Ohne Beweis. Abbildung 5.32 zeigt eine anschauliche Darstellung der Formel in Übereinstimmung mit dem in Abbildung 5.31 gezeigten Prinzip. q.e.d.

Anmerkung: Im Fall $\mathbf{n} = (0, 0, 1)^T$ sind die Bedingungen $\mu \leq 0$ bzw. $\mu > 0$ aus Formel (5.182) auch ersetzbar durch $z_{n_i} \leq 0$ bzw. $z_{n_i} > 0$, das heißt in ausgeschriebener Form:

$$(y_{p_{i+1}} - y_p) \cdot (x_{p_{i+1}} - x_{p_i}) \leq (y_{p_{i+1}} - y_{p_i}) \cdot (x_{p_{i+1}} - x_p) \quad (5.183)$$

bzw.

$$(y_{p_{i+1}} - y_p) \cdot (x_{p_{i+1}} - x_{p_i}) > (y_{p_{i+1}} - y_{p_i}) \cdot (x_{p_{i+1}} - x_p). \quad (5.184)$$

Im Folgenden wird, ausgehend von dem oben beschriebenen Punkt-in-Polygon-Test nach Jordan gezeigt, wie die Windungszahl $\eta(\gamma_n, \mathbf{p})$ (siehe Definition 16) einer geschlossenen Kurve

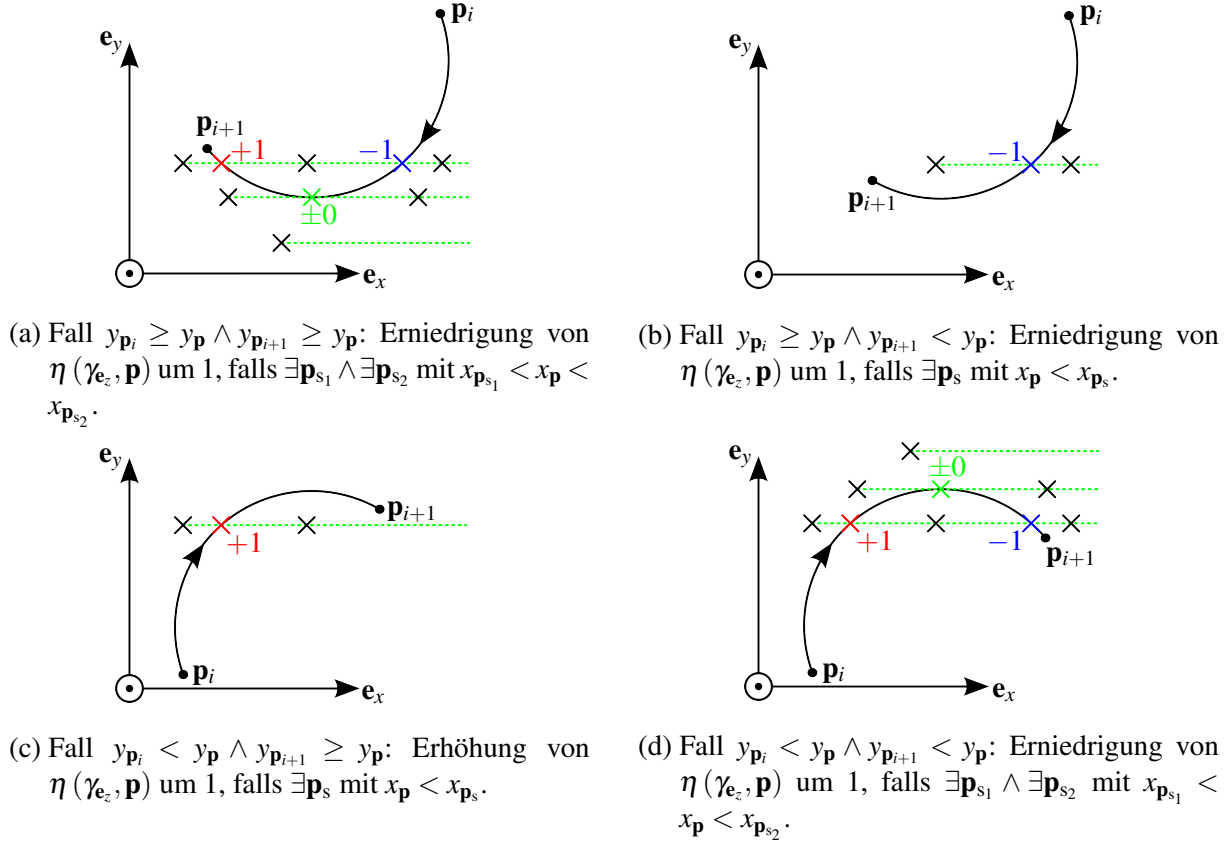


Abbildung 5.33: Auswirkungen auf die Windungszahl $\eta(\gamma_{e_z}, \mathbf{p})$ im Punkt $\mathbf{p} = (x_{\mathbf{p}}, y_{\mathbf{p}}, z_{\mathbf{p}})^T \notin \gamma_{e_z}$ durch Schneiden eines Strahls $\mathbf{R} := \{\mathbf{p} + \lambda \mathbf{e}_x \mid 0 \leq \lambda \in \mathbb{R}\}$ mit einem Kreisbogen $\mathbf{K}_{\text{Bog}}(\mathbf{p}_i, \mathbf{p}_{i+1}) \subset \partial \mathbf{K}_{r,n}(\mathbf{p}_{M_i})$. In jedem Beispiel sind verschiedene Lagen des Punktes \mathbf{p} eingezeichnet (siehe schwarze Kreuze). $\mathbf{G} := \{\mathbf{p} + \lambda \mathbf{e}_x \mid \lambda \in \mathbb{R}\}$ bezeichnet die durch \mathbf{p} verlaufende Gerade parallel zur e_x -Achse. Mit \mathbf{p}_s , \mathbf{p}_{s_1} bzw. \mathbf{p}_{s_2} werden die Schnittpunkte des Strahls \mathbf{G} mit dem Kreisbogen \mathbf{K}_{Bog} bezeichnet.

$\gamma_{\mathbf{n}}$ (siehe Definition 14), welche aus Segmenten $\mathbf{S}(\mathbf{p}_i, \mathbf{p}_{i+1})$ und Kreisbögen $\mathbf{K}_{\text{Bog}}(\mathbf{p}_i, \mathbf{p}_{i+1})$ besteht, im Punkt $\mathbf{p} = (x_{\mathbf{p}}, y_{\mathbf{p}}, z_{\mathbf{p}})^T \notin \gamma_{\mathbf{n}}$ berechnet wird. Anstatt speziell von Segmenten und Kreisbögen zu sprechen, wird auch allgemein der Begriff des geometrischen Element $\mathbf{E}(\mathbf{p}_i, \mathbf{p}_{i+1})$ verwendet. Die Vorgehensweise ist bei Kreisbögen ähnlich wie bei den Segmenten zuvor. Eine detailliertere Illustration ist in Abbildung 5.33 zu sehen. Auch hier wird ausgehend vom Punkt \mathbf{p} ein Strahl $\mathbf{R} := \{\mathbf{p} + \lambda \mathbf{e}_x \mid 0 \leq \lambda \in \mathbb{R}\}$ in Richtung der positiven e_x -Achse betrachtet. Wandert man nun auf dem Strahl \mathbf{R} mit zunehmendem λ , dann wird dort, wo der Kreisbogen $\mathbf{K}_{\text{Bog}}(\mathbf{p}_i, \mathbf{p}_{i+1})$, den Strahl von rechts nach links schneidet, $\eta(\gamma_{\mathbf{n}}, \mathbf{p})$ um 1 erhöht. Dort wo der Kreisbogen $\mathbf{K}_{\text{Bog}}(\mathbf{p}_i, \mathbf{p}_{i+1})$, den Strahl von links nach rechts schneidet, wird $\eta(\gamma_{\mathbf{n}}, \mathbf{p})$ hingegen um 1 erniedrigt. Die Summe ergibt dann die Veränderung der Windungszahl $\eta(\gamma_{\mathbf{n}}, \mathbf{p})$ eingeschränkt auf einen einzelnen Kreisbogen. Wie in Abbildung 5.33 zu sehen ist, ergibt sich daraus ein Gesamteffekt auf die Windungszahl $\eta(\gamma_{\mathbf{n}}, \mathbf{p})$ je nach relativer Lage von \mathbf{p} und $\mathbf{K}_{\text{Bog}}(\mathbf{p}_i, \mathbf{p}_{i+1})$ zueinander. Es gilt nun folgender Satz:

Satz 23 (Windungszahl für geschlossene Kurven aus Segmenten und Kreisbögen):

Sei $\gamma_{\mathbf{n}}$ eine geschlossene Kurve (siehe Definition 14), welche aus geometrischen Elementen $\mathbf{E}(\mathbf{p}_i, \mathbf{p}_{i+1})$, $i \in \{0, 1, \dots, n-1\}$, besteht. Dabei gilt entweder $\mathbf{E}(\mathbf{p}_i, \mathbf{p}_{i+1}) = \mathbf{S}(\mathbf{p}_i, \mathbf{p}_{i+1})$ im Fall

eines Segments oder $\mathbf{E}(\mathbf{p}_i, \mathbf{p}_{i+1}) = \mathbf{K}_{\text{Bog}}(\mathbf{p}_i, \mathbf{p}_{i+1})$ im Fall eines Kreisbogens. Seien weiter der Punkt $\mathbf{p} \notin \gamma_n$ und im Fall $\mathbf{E}(\mathbf{p}_i, \mathbf{p}_{i+1}) = \mathbf{S}(\mathbf{p}_i, \mathbf{p}_{i+1})$ der Vektor

$$\mathbf{n}_i := (\mathbf{p}_{i+1} - \mathbf{p}_i) \times (\mathbf{p}_{i+1} - \mathbf{p}) \quad (5.185)$$

gegeben, wobei \mathbf{n}_i , genau so wie \mathbf{n} , senkrecht auf der von γ_n gebildeten Ebene steht, d.h. es gilt:

$$\mathbf{n}_i = \mu \cdot \mathbf{n}, \mu \in \mathbb{R}. \quad (5.186)$$

Falls ein oder mehrere Schnittpunkte zwischen der Geraden $\mathbf{G} := \{\mathbf{p} + \lambda \mathbf{e}_x \mid \lambda \in \mathbb{R}\}$ und einem Kreisbogen $\mathbf{E}(\mathbf{p}_i, \mathbf{p}_{i+1}) = \mathbf{K}_{\text{Bog}}(\mathbf{p}_i, \mathbf{p}_{i+1})$ existieren, dann werden diese mit \mathbf{p}_s , \mathbf{p}_{s_1} bzw. \mathbf{p}_{s_2} bezeichnet. Die Windungszahl $\eta(\gamma_n, \mathbf{p})$ im Punkt \mathbf{p} lässt sich dann berechnen durch:

$$\eta(\gamma_n, \mathbf{p}) = \eta_S(\gamma_n, \mathbf{p}) + \eta_{\text{K}_{\text{Bog}}}(\gamma_n, \mathbf{p}) \quad (5.187)$$

mit

$$\eta_S(\gamma_n, \mathbf{p}) = \sum_{i=0}^{n-1} \begin{cases} 1 & \mathbf{E}(\mathbf{p}_i, \mathbf{p}_{i+1}) = \mathbf{S}(\mathbf{p}_i, \mathbf{p}_{i+1}) : (y_{\mathbf{p}_i} < y_{\mathbf{p}} \leq y_{\mathbf{p}_{i+1}}) \wedge (\mu < 0) \\ -1 & \mathbf{E}(\mathbf{p}_i, \mathbf{p}_{i+1}) = \mathbf{S}(\mathbf{p}_i, \mathbf{p}_{i+1}) : (y_{\mathbf{p}_{i+1}} < y_{\mathbf{p}} \leq y_{\mathbf{p}_i}) \wedge (\mu > 0) \\ 0 & \mathbf{E}(\mathbf{p}_i, \mathbf{p}_{i+1}) = \mathbf{K}_{\text{Bog}}(\mathbf{p}_i, \mathbf{p}_{i+1}) \end{cases} \quad (5.188)$$

und

$$\eta_{\text{K}_{\text{Bog}}}(\gamma_n, \mathbf{p}) = \sum_{i=0}^{n-1} \begin{cases} 1 & \mathbf{E}(\mathbf{p}_i, \mathbf{p}_{i+1}) = \mathbf{K}_{\text{Bog}}(\mathbf{p}_i, \mathbf{p}_{i+1}) : \\ & (y_{\mathbf{p}_i} < y_{\mathbf{p}} \wedge y_{\mathbf{p}_{i+1}} \geq y_{\mathbf{p}}) \wedge (\exists \mathbf{p}_s : x_{\mathbf{p}} < x_{\mathbf{p}_s}) \\ -1 & \mathbf{E}(\mathbf{p}_i, \mathbf{p}_{i+1}) = \mathbf{K}_{\text{Bog}}(\mathbf{p}_i, \mathbf{p}_{i+1}) : \\ & \left((y_{\mathbf{p}_i} \geq y_{\mathbf{p}} \wedge y_{\mathbf{p}_{i+1}} \geq y_{\mathbf{p}}) \wedge (\exists \mathbf{p}_{s_1} \wedge \exists \mathbf{p}_{s_2} : x_{\mathbf{p}_{s_1}} < x_{\mathbf{p}} < x_{\mathbf{p}_{s_2}}) \right) \\ & \vee \left((y_{\mathbf{p}_i} \geq y_{\mathbf{p}} \wedge y_{\mathbf{p}_{i+1}} < y_{\mathbf{p}}) \wedge (\exists \mathbf{p}_s : x_{\mathbf{p}} < x_{\mathbf{p}_s}) \right) \\ & \vee \left((y_{\mathbf{p}_i} < y_{\mathbf{p}} \wedge y_{\mathbf{p}_{i+1}} < y_{\mathbf{p}}) \wedge (\exists \mathbf{p}_{s_1} \wedge \exists \mathbf{p}_{s_2} : x_{\mathbf{p}_{s_1}} < x_{\mathbf{p}} < x_{\mathbf{p}_{s_2}}) \right) \\ 0 & \mathbf{E}(\mathbf{p}_i, \mathbf{p}_{i+1}) = \mathbf{S}(\mathbf{p}_i, \mathbf{p}_{i+1}) \end{cases} \quad (5.189)$$

Beweis: Ohne Beweis. Abbildung 5.32 zeigt eine anschauliche Darstellung der Formel (5.188) in Übereinstimmung mit Satz 22. In Abbildung 5.33 ist hingegen Formel (5.189) illustriert. q.e.d.

Anmerkung: Satz 22 ist somit ein Spezialfall von Satz 23. Abbildung 5.34 zeigt ein einfaches Beispiel für Satz 23.

Satz 24 (Eigenschaften von Windungszahlen):

Es seien alle Voraussetzungen und Eigenschaften aus Satz 23 erfüllt. Es gilt

$$\eta(\gamma_n, \mathbf{p}) = 0, \quad (5.190)$$

genau dann, wenn \mathbf{p} außerhalb der geschlossenen Kurve γ_n liegt, das heißt, es gibt einen Weg von \mathbf{p} ins Unendliche ohne die geschlossene Kurve γ_n zu schneiden. Außerdem gilt

$$\eta(\gamma_n, \mathbf{p}) \neq 0, \quad (5.191)$$

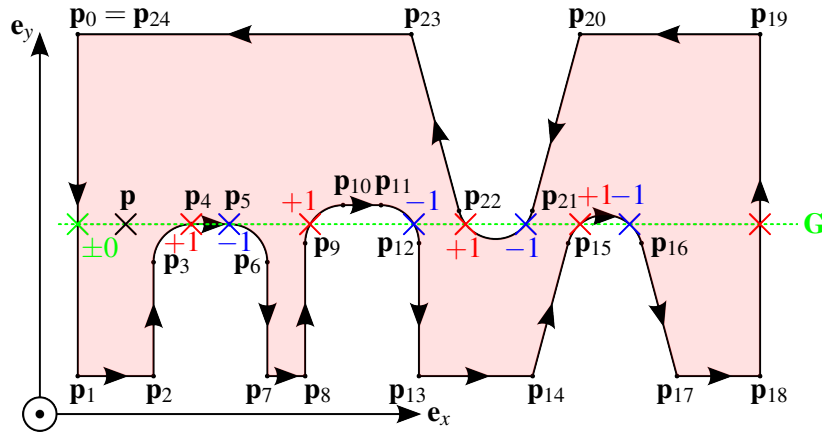


Abbildung 5.34: Beispiel für die Berechnung der Windungszahl $\eta(\gamma_{e_z}, \mathbf{p})$ im Punkt $\mathbf{p} = (x_{\mathbf{p}}, y_{\mathbf{p}}, z_{\mathbf{p}})^T \notin \gamma_{e_z}$ für eine geschlossene Kurve γ_{e_z} nach Definition 14, die aus Segmenten $\mathbf{E}(p_i, p_{i+1}) = \mathbf{S}(p_i, p_{i+1})$ und Kreisbögen $\mathbf{E}(p_i, p_{i+1}) = \mathbf{K}_{Bog}(p_i, p_{i+1})$ besteht. Für Segmente wird bei der Berechnung vorgegangen wie in Satz 22. Die Idee bei den Kreisbögen, ist wie bereits in Abbildung 5.33 illustriert. So ist beispielsweise für $\mathbf{K}_{Bog}(p_3, p_4)$ der Fall in Abbildung 5.33c erfüllt und $\eta(\gamma_{e_z}, \mathbf{p})$ wird um 1 erhöht. Für $\mathbf{S}(p_4, p_5)$ ist hingegen keiner der Fälle aus Abbildung 5.32 erfüllt (Grenzfall), womit $\eta(\gamma_{e_z}, \mathbf{p})$ unverändert bleibt. Für $\mathbf{K}_{Bog}(p_5, p_6)$ ist der Fall 5.33b erfüllt und $\eta(\gamma_{e_z}, \mathbf{p})$ wird um 1 erniedrigt. Auch für $\mathbf{K}_{Bog}(p_9, p_{10})$ bzw. $\mathbf{K}_{Bog}(p_{11}, p_{12})$ wird $\eta(\gamma_{e_z}, \mathbf{p})$ um 1 erhöht bzw. erniedrigt. Für die Kreisbögen $\mathbf{K}_{Bog}(p_{15}, p_{16})$ bzw. $\mathbf{K}_{Bog}(p_{21}, p_{22})$ sind hingegen die Fälle in Abbildung 5.33d bzw. 5.33a nicht erfüllt, da es keine entsprechende Schnittpunkte \mathbf{p}_{s_1} und \mathbf{p}_{s_2} gibt. Die Windungszahl $\eta(\gamma_{e_z}, \mathbf{p})$ bleibt in diesen Fällen also insgesamt unverändert. Für das Segment $\mathbf{S}(p_{18}, p_{19})$ gilt hingegen der Fall in Abbildung 5.32a, womit $\eta(\gamma_{e_z}, \mathbf{p})$ um 1 erhöht wird. Insgesamt ergibt sich also $\eta(\gamma_{e_z}, \mathbf{p}) = 1$.

genau dann, wenn \mathbf{p} innerhalb der geschlossenen Kurve $\gamma_{\mathbf{n}}$ liegt, das heißt, es gibt keinen Weg von \mathbf{p} ins Unendliche ohne die geschlossene Kurve $\gamma_{\mathbf{n}}$ zu schneiden. Sei $\gamma_{\mathbf{n}}$ nun eine geschlossene Kurve ohne Überschneidungen. In diesem Fall gilt speziell

$$\eta(\gamma_{\mathbf{n}}, \mathbf{p}) = 1, \quad (5.192)$$

genau dann, wenn \mathbf{p} innerhalb der gegen den Uhrzeigersinn laufenden geschlossenen Kurve $\gamma_{\mathbf{n}}$ liegt und

$$\eta(\gamma_{\mathbf{n}}, \mathbf{p}) = -1, \quad (5.193)$$

genau dann, wenn \mathbf{p} innerhalb der im Uhrzeigersinn laufenden geschlossenen Kurve $\gamma_{\mathbf{n}}$ liegt.

Beweis: Ohne Beweis.

q.e.d.

Anmerkung: Nach Satz 21 ist es also äquivalent, den Wert der Windungszahl $\eta(\gamma_{\mathbf{n}}, \mathbf{p})$ oder das Vorzeichen der von $\gamma_{\mathbf{n}}$ umschlossenen Fläche zu berechnen um die Umlaufrichtung einer geschlossenen Kurve $\gamma_{\mathbf{n}}$ ohne Überschneidungen zu bestimmen. Für die Berechnung der Windungszahl $\eta(\gamma_{\mathbf{n}}, \mathbf{p})$ ist allerdings die Kenntnis eines Punktes \mathbf{p} erforderlich, der innerhalb von $\gamma_{\mathbf{n}}$ liegt. Das Bestimmen eines solchen Punktes kann abhängig von der Geometrie der Kurve $\gamma_{\mathbf{n}}$ sehr schwierig werden.

5.3 Algorithmik der Software-Module

In diesem Abschnitt werden die algorithmischen Umsetzungen der einzelnen Software-Module genau erläutert. Hierzu wird auf die mathematischen Definitionen aus Kapitel 5.1 und die mathematischen Grundlagen aus Kapitel 5.2 zurückgegriffen.

5.3.1 Importieren der STL-Daten

Die STL-Daten im ASCII- bzw. Binär-Format müssen zunächst eingelesen werden. Dies bedeutet, dass eine entsprechende STL-Datei von der Festplatte geöffnet und die relevanten Daten in den Speicher des Rechners geladen werden. Im Falle von STL-Dateien sind dies die Dreiecke, welche die Oberfläche des Volumenkörpers / Bauteils beschreiben. Hierfür wird die in Abschnitt 4.3.2 dargestellte Notation in Form einer Vertex List **VL** und einer Facet List **FL** genutzt. Somit kann über die Positionsindizes in der Facet List für jedes Dreieck (entspricht einer Zeile in der Notation von Darstellung (4.10)) auf den entsprechenden Eckpunkt (Vertex) zugegriffen werden. D.h. für jedes Dreieck werden drei Positionsindizes gespeichert, welche dann wiederum auf den jeweiligen Eckpunkt verweisen. Aus algorithmischer Sicht gibt es beim Einlesen der Daten keine Besonderheiten.

Obwohl die Transformation von STL-Daten in ein anderes Koordinatensystem nicht direkt zum Importieren der STL-Daten gehört, hat eine solche Transformation natürlich maßgebende Einflüsse auf die im Anschluss berechneten Daten. Dies beginnt gleich beim Slicen der STL-Daten. Darum soll vorab noch auf eventuelle Transformationen der importierten STL-Daten eingegangen werden.

Rotationen

Sollen Bauteile um die Winkel α_x , α_y und α_z um die \mathbf{e}_x -, \mathbf{e}_y - und \mathbf{e}_z -Achse gedreht werden, so geschieht dies nach der in Kapitel 5.2.1 in Satz 12 festgehaltenen Transformationskonvention mit der homogenen Transformationsmatrix $\mathbf{T}(\alpha_x, \alpha_y, \alpha_z)$. So kann unter Umständen eine günstigere Ausrichtung des Bauteils für den Bauprozess erreicht werden. Alle Punkte $\mathbf{p}_i \in \mathbf{VL}$, $i \in \{1, \dots, n\}$ werden dabei durch die Vorschrift

$$\mathbf{p}'_i = \mathbf{T}(\alpha_x, \alpha_y, \alpha_z) \mathbf{p}_i \quad (5.194)$$

transformiert und es ergibt sich eine neue Vertex List \mathbf{VL}' . Die Facet List **FL** bleibt unverändert.

Skalierungen

Bauteile können, soweit gewünscht, auch skaliert werden. Dies geschieht über sogenannte Skalierungsfaktoren $f_x > 0$, $f_y > 0$ und $f_z > 0$ in \mathbf{e}_x -, \mathbf{e}_y - und \mathbf{e}_z -Richtung. Das bedeutet, dass alle Punkte $\mathbf{p}_i \in \mathbf{VL}$, $i \in \{1, \dots, n\}$, transformiert werden durch:

$$\mathbf{p}'_i = \begin{pmatrix} f_x \cdot x_{\mathbf{p}_i} \\ f_y \cdot y_{\mathbf{p}_i} \\ f_z \cdot z_{\mathbf{p}_i} \end{pmatrix}. \quad (5.195)$$

Daraus ergibt sich wiederum eine Vertex List \mathbf{VL}' bei unveränderter Facet List **FL**.

Translationen

Seien $\mathbf{p}_i \in \mathbf{VL}$ und weiter $x_{\min} := \min_{i \in \{1, \dots, n\}} \{x_{\mathbf{p}_i}\}$, $y_{\min} := \min_{i \in \{1, \dots, n\}} \{y_{\mathbf{p}_i}\}$, $z_{\min} := \min_{i \in \{1, \dots, n\}} \{z_{\mathbf{p}_i}\}$,
 $x_{\max} := \max_{i \in \{1, \dots, n\}} \{x_{\mathbf{p}_i}\}$, $y_{\max} := \max_{i \in \{1, \dots, n\}} \{y_{\mathbf{p}_i}\}$ und $z_{\max} := \max_{i \in \{1, \dots, n\}} \{z_{\mathbf{p}_i}\}$. Durch die Vektoren

$$\mathbf{p}_{\min} := \begin{pmatrix} x_{\min} \\ y_{\min} \\ z_{\min} \end{pmatrix} \quad (5.196)$$

und

$$\mathbf{p}_{\max} := \begin{pmatrix} x_{\max} \\ y_{\max} \\ z_{\max} \end{pmatrix} \quad (5.197)$$

wird ein Quader (Bounding Box) definiert, innerhalb dem alle Punkte $\mathbf{p}_i \in \mathbf{VL}$ liegen. Sei

$$\mathbf{I} := \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (5.198)$$

die Einheitsmatrix, dann wird durch

$$\mathbf{T} := \begin{pmatrix} \mathbf{I} & -\mathbf{p}_{\min} \\ \mathbf{0} & 1 \end{pmatrix} \quad (5.199)$$

einer homogene Transformationsmatrix definiert, mit der man alle Punkte $\mathbf{p}_i \in \mathbf{VL}$, $i \in \{1, \dots, n\}$, über die Vorschrift

$$\mathbf{p}'_i = \mathbf{T}\mathbf{p}_i \quad (5.200)$$

so verschieben kann, dass der umhüllende Quader in den Ursprung verschoben wurde:

$$\mathbf{p}'_{\min} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}. \quad (5.201)$$

Natürlich wird über die Vorschrift

$$\mathbf{p}'_i = \mathbf{p}_i - \mathbf{p}_{\min} \quad (5.202)$$

derselbe Effekt erreicht. Außerdem gilt:

$$\mathbf{p}'_{\max} = \mathbf{p}_{\max} - \mathbf{p}_{\min}. \quad (5.203)$$

Es ergibt sich also wieder eine neue Vertex List \mathbf{VL}' bei unveränderter Facet List \mathbf{FL} . Diese Verschiebung wird automatisch so durchgeführt und dient als Grundlage für die anschließende Datenaufbereitung.

Abbildung 5.35 zeigt nochmals ein einfaches Beispiel für die Notation in Form einer Vertex List \mathbf{VL} und einer Facet List \mathbf{FL} . Anhand dieses Beispiels wird im Folgenden auch das Slicen der STL-Daten in Kapitel 5.3.2 erläutert.

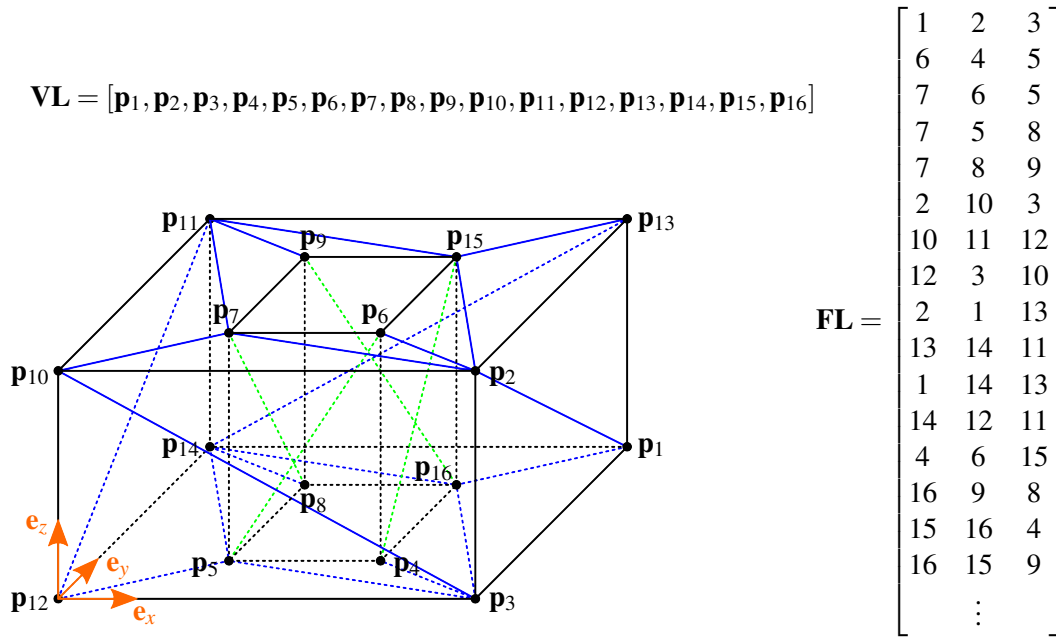


Abbildung 5.35: Quader mit Loch. Die Triangulierung besteht in diesem Fall aus einer Vertex List $\mathbf{VL} \in \mathbb{R}^{3 \times 16}$ und einer Facet List $\mathbf{FL} \in \mathbb{R}^{32 \times 3}$. \mathbf{FL} besteht also aus 32 Dreiecken (Zeilen). \mathbf{FL} ist nicht vollständig angegeben, es fehlen alle Dreiecke mit Normalenvektor $\mathbf{n} = (0, 0, \pm 1)^T$, das heißt, alle Dreiecke, die parallel sind zur \mathbf{e}_x - \mathbf{e}_y -Fläche.

5.3.2 Slicen der STL-Daten

Schneiden der STL-Daten mit Ebenen

Nach Importieren der STL-Daten in den Speicher des Rechners, müssen aus dem Volumenmodell 2.5D Schichtinformationen berechnet werden. Hierfür wird das im STL-Format definierte Dreiecksgitter im Idealfall mit in \mathbf{e}_z -Richtung äquidistanten zweidimensionalen Ebenen der Form $\mathbf{E}_H(\mathbf{e}_z, \mathbf{p}_i)$ geschnitten, wobei $\mathbf{p}_i = (0, 0, z_{\mathbf{p}_i})^T$ mit $z_{\mathbf{p}'_{\min}} < z_{\mathbf{p}_i} \leq z_{\mathbf{p}'_{\max}}$ (siehe Gleichungen (5.201) und (5.203)). Dabei gilt für die erste Schicht in der Regel:

$$z_{\mathbf{p}_i} = z_{\mathbf{p}'_{\min}} + H, \quad (5.204)$$

wobei H der Schichtabstand ist und somit der Tropfenhöhe entspricht. Alle weiteren Schichten ergeben sich durch:

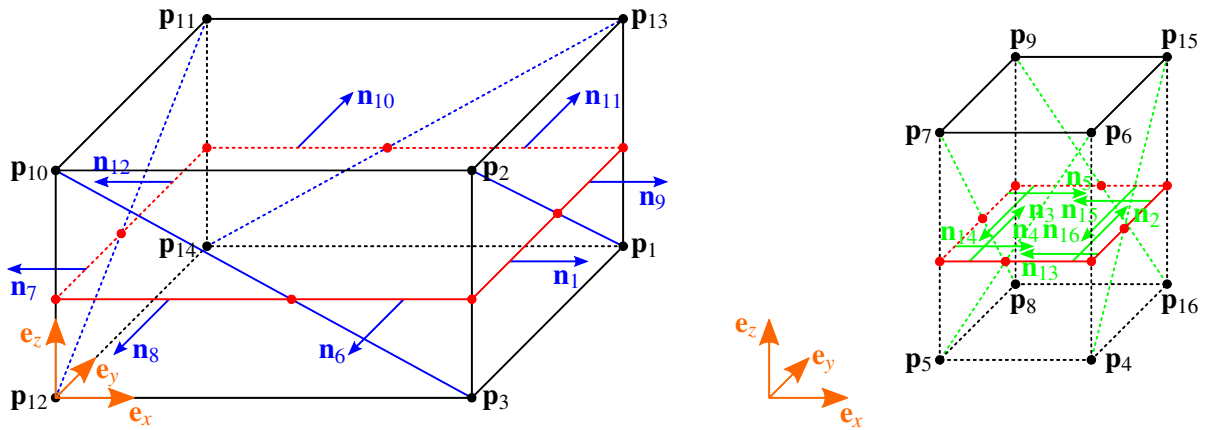
$$z_{\mathbf{p}_{i+1}} = z_{\mathbf{p}_i} + H. \quad (5.205)$$

Da es sich bei dem Dreiecksgitter um eine Darstellung der Oberfläche des Volumenmodells handelt, errechnet sich durch den Schnittvorgang eine Begrenzung des Volumenmodells in der jeweiligen Schnittebene. Diese Begrenzung setzt sich aus den jeweiligen Schnittsegmenten / Schnittpunkten der Schnittebene mit den im STL definierten Dreiecken zusammen. Keine Schnitte gibt es dabei mit Dreiecken, für die gilt:

$$z_{\mathbf{pFL}_{j,1}} < z_{\mathbf{p}_i} \wedge z_{\mathbf{pFL}_{j,2}} < z_{\mathbf{p}_i} \wedge z_{\mathbf{pFL}_{j,3}} < z_{\mathbf{p}_i} \quad (5.206)$$

oder

$$z_{\mathbf{pFL}_{j,1}} > z_{\mathbf{p}_i} \wedge z_{\mathbf{pFL}_{j,2}} > z_{\mathbf{p}_i} \wedge z_{\mathbf{pFL}_{j,3}} < z_{\mathbf{p}_i}, \quad (5.207)$$



(a) Äußerer Quader. Nur Dreiecke betrachtet mit einem Normalenvektor $\mathbf{n} \neq (0, 0, \pm 1)^T$.

(b) Innerer Quader (Loch) bestehend aus acht Dreiecken.

Abbildung 5.36: Quader mit Loch: \mathbf{VL} und \mathbf{FL} sind wie Abbildung 5.35. Die äußeren Normalen werden mit \mathbf{n}_j bezeichnet, wobei damit die Normale des Dreiecks aus der j -ten Zeile von \mathbf{FL} gemeint ist. Das Dreiecksgitter wird mit einer Ebene $\mathbf{E}_H(\mathbf{e}_z, \mathbf{p})$ geschnitten, wobei $z_{\min} < z_{\mathbf{p}} < z_{\max}$ gilt. Rot eingezeichnet sind die Schnittsegmente der Ebene mit den jeweiligen Dreiecken der Triangulierung.

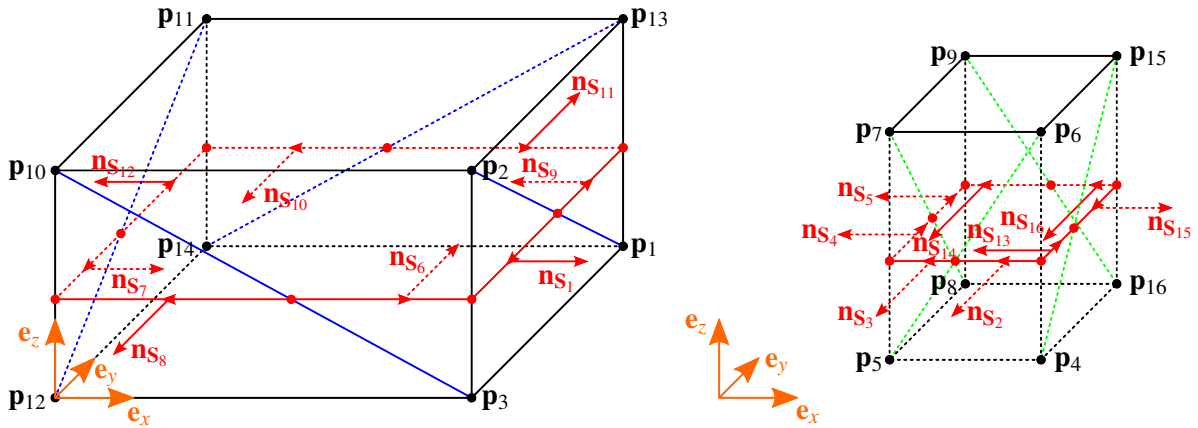
wobei $j \in \{1, \dots, m\}$ gilt und $\mathbf{FL}_{j,1}$, $\mathbf{FL}_{j,2}$, $\mathbf{FL}_{j,3}$ die erste, zweite, dritte Spalte der j -ten Zeile von \mathbf{FL} sind. Diese Dreiecke werden von vornherein für eine Berechnung der Schnittsegmente ausgeschlossen. Falls

$$\left| z_{\mathbf{p}_{\mathbf{VL}_{j,1}}} - z_{\mathbf{p}_i} \right| < 0.001 \wedge \left| z_{\mathbf{p}_{\mathbf{VL}_{j,2}}} - z_{\mathbf{p}_i} \right| < 0.001 \wedge \left| z_{\mathbf{p}_{\mathbf{VL}_{j,3}}} - z_{\mathbf{p}_i} \right| < 0.001 \quad (5.208)$$

gilt, so wird angenommen dass das Dreieck in der Schnittebene liegt, das heißt also insbesondere, dass es keinen eindeutigen Schnittpunkt gibt. Solche Dreiecke werden ebenfalls von vorne herein ausgeschlossen, indem die Schnittebene $\mathbf{E}_H(\mathbf{e}_z, \mathbf{p}_i)$ solange um $\Delta z = 0.001$ gesenkt wird, bis es keine Dreiecke mehr gibt, für die Gleichung (5.208) gilt. Die Schnittebene wird also für die stabile Berechnung etwas gesenkt, wobei für die anschließenden Berechnungen wieder angenommen wird, dass die Schnittsegmente in der ursprünglichen Schnittebene $\mathbf{E}_H(\mathbf{e}_z, \mathbf{p}_i)$ liegen.

Reihenfolge der Schnittsegmente

Es lässt sich von vornherein keine sichere Aussage darüber treffen, in welcher Reihenfolge die Dreiecke in der STL-Datei stehen. Nach und nach werden die Positionsindizes eines Dreiecks Schritt für Schritt bzw. Dreieck für Dreieck zeilenweise in der Facet List \mathbf{FL} hinterlegt. Dies ist auch nochmals in Abbildung 5.36 illustriert. Hier ist am Beispiel des Quaders mit einem Loch an den äußeren Normalen \mathbf{n}_j , $j \in \{1, \dots, m\}$ des j -ten Dreiecks zu sehen, dass bei den Dreiecken keine strikte Reihenfolge einzuhalten ist. Lediglich die Anordnung der Punkte muss so sein, dass die Normale wirklich nach außen vom Bauteil zeigt. In Abbildung 5.36 sind bereits sämtliche Dreiecke vernachlässigt, die für eine Schnittberechnung nicht relevant sind (siehe oben). Abbildung 5.37 zeigt nun, dass die Reihenfolge der Berechnung der Schnittsegmente ebenfalls von der Reihenfolge der Dreiecke im Speicher, also insbesondere in der Facet List \mathbf{FL} , abhängt. Hierbei ist allerdings auch zu erkennen, dass die Anordnung der Positionsindizes in einer Zeile \mathbf{FL} nicht nur dafür sorgt, dass die Normale wirklich nach außen zeigen, sondern dass die



(a) Äußerer Quader. Nur Dreiecke betrachtet mit einem Normalenvektor $\mathbf{n} \neq (0, 0, \pm 1)^T$. (b) Innerer Quader (Loch) bestehend aus acht Dreiecken.

Abbildung 5.37: Quader mit Loch: \mathbf{VL} und \mathbf{FL} sind wie Abbildung 5.35. Das Dreiecksgitter wird mit einer Ebene $E_H(\mathbf{e}_z, \mathbf{p})$ geschnitten, wobei $z_{\min} < z_p < z_{\max}$ gilt. Rot eingezeichnet sind die Schnittsegmente der Ebene mit den jeweiligen Dreiecken der Triangulierung und deren Orientierung (Segment läuft vom ersten berechneten Schnittpunkt zum zweiten berechneten Schnittpunkt). Mit \mathbf{n}_{S_j} werden die Normalen an das Schnittsegment S_j bezeichnet, welches sich aus Dreieck j ergibt (siehe auch die äußeren Normalen des j -ten Dreiecks in Abbildung 5.36). Die Normale an das jeweilige Schnittsegment liegt innerhalb der Schnittebene $E_H(\mathbf{e}_z, \mathbf{p})$.

Anordnung auch Einfluss auf die Richtung der berechneten Schnittsegmente hat: Für das j -te Dreieck wird hier der Reihe nach geprüft, ob die Segmente $S(\mathbf{PFL}_{j,1}, \mathbf{PFL}_{j,2})$, $S(\mathbf{PFL}_{j,2}, \mathbf{PFL}_{j,3})$ und $S(\mathbf{PFL}_{j,1}, \mathbf{PFL}_{j,3})$ einen Schnittpunkt mit der Ebene $E_H(\mathbf{e}_z, \mathbf{p}_i)$ haben. Hierbei wird bei der Schnittpunktberechnung zwischen der Ebene und einem einzelnen Segment vorgegangen wie in Kapitel 5.2.3 gezeigt. Falls es genau zwei Schnittpunkte $\mathbf{p}_{s_{j,1}}$ und $\mathbf{p}_{s_{j,2}}$ mit einem Dreieck gibt, dann ergibt sich daraus ein Schnittsegment $S_j(\mathbf{p}_{s_{j,1}}, \mathbf{p}_{s_{j,2}})$. Für die in Abbildung 5.37 eingezeichneten Normalen an das j -te Schnittsegment gilt dann:

$$\mathbf{n}_{S_j} = \mathbf{e}_z \times (\mathbf{p}_{s_{j,2}} - \mathbf{p}_{s_{j,1}}), \quad (5.209)$$

das heißt, dass \mathbf{n}_{S_j} wieder innerhalb der Ebene $E_H(\mathbf{e}_z, \mathbf{p}_i)$ liegt, da auch $\mathbf{p}_{s_{j,2}} - \mathbf{p}_{s_{j,1}}$ innerhalb der Ebene liegt, \mathbf{e}_z aber senkrecht auf der Ebene steht. Falls $|\mathbf{p}_{s_{j,2}} - \mathbf{p}_{s_{j,1}}| = 0$ gilt, dann schneiden sich die Ebene und das j -te Dreieck in einem Eckpunkt des Dreiecks. Für diesen Fall existiert kein Schnittsegment.

Orientierung der Schnittsegmente

Nun müssen die einzelnen Schnittsegmente noch zu geschlossenen Konturen nach Definition 14 zusammengefügt werden. Hierzu müssen nicht nur die Schnittsegmente richtig geordnet, sondern auch im Vorfeld richtig orientiert werden. Idee dahinter ist, dass Bauteile später immer links von der Umlaufrichtung einer geschlossenen Kontur gefüllt werden sollen. Dies entspricht dem Inneren eines Bauteils und ist sozusagen das Gegenstück zur äußeren Dreiecksnormale \mathbf{n}_j . Um nun zu entscheiden, ob ein Schnittsegment richtig orientiert ist oder falsch, wird die

Normale \mathbf{n}_{S_j} näher betrachtet. Gilt nun

$$\alpha := \angle(\mathbf{n}_{S_j}, \mathbf{n}_j) > \frac{\pi}{2}, \quad (5.210)$$

so zeigt \mathbf{n}_{S_j} ins Innere des Bauteils. Es kann also mit dem Schnittsegment $\mathbf{S}'_j(\mathbf{p}'_{s_{j,1}}, \mathbf{p}'_{s_{j,2}}) := \mathbf{S}_j(\mathbf{p}_{s_{j,1}}, \mathbf{p}_{s_{j,2}})$ und dem Normalenvektor $\mathbf{n}'_{S'_j} := \mathbf{n}_{S_j}$ fortgefahren werden. Gilt hingegen:

$$\alpha := \angle(\mathbf{n}_{S_j}, \mathbf{n}_j) < \frac{\pi}{2}, \quad (5.211)$$

so zeigt \mathbf{n}_{S_j} aus dem Bauteil heraus. In diesem Fall muss die Orientierung geändert und mit dem Schnittsegment $\mathbf{S}'_j(\mathbf{p}'_{s_{j,1}}, \mathbf{p}'_{s_{j,2}}) := \mathbf{S}_j(\mathbf{p}_{s_{j,2}}, \mathbf{p}_{s_{j,1}})$ und dem Normalenvektor $\mathbf{n}'_{S'_j} := -\mathbf{n}_{S_j}$ fortgefahren werden. Der Fall

$$\alpha := \angle(\mathbf{n}_{S_j}, \mathbf{n}_j) = \frac{\pi}{2} \quad (5.212)$$

wurde durch die Behandlung des Falls in Gleichung (5.208) bereits ausgeschlossen. Für die Berechnung des Winkels α wird Satz 1 verwendet.

Zusammenfügen der Schnittsegmente zu geschlossenen Konturen

Haben schließlich alle Schnittsegmente die korrekte Orientierung, so müssen diese nur richtig sortiert werden um eine geschlossene Kontur zu erhalten (die Reihenfolge der Schnittsegmente entspricht zu Beginn der Reihenfolge der Dreiecke in der Facet List **FL**): Zunächst wird das erste Segment zur Kontur hinzugefügt. Dieses wird im Folgenden ohne Beschränkung der Allgemeinheit mit $\mathbf{S}'_1(\mathbf{p}'_{s_{1,1}}, \mathbf{p}'_{s_{1,2}})$ bezeichnet. Als nächstes werden dann Schritt für Schritt weitere Segmente hinzugefügt, indem überprüft wird, ob der Startpunkt des neu einzufügenden Segments mit dem Endpunkt des zuvor eingefügten Segments übereinstimmt, immer unter Berücksichtigung einer gewissen Rundungsungenauigkeit: das heißt, falls $\mathbf{S}'_j(\mathbf{p}'_{s_{j,1}}, \mathbf{p}'_{s_{j,2}})$ das letzte zur Kontur hinzugefügte Segment ist, dann suche nach einem Schnittsegment $\mathbf{S}'_k(\mathbf{p}'_{s_{k,1}}, \mathbf{p}'_{s_{k,2}})$, das noch nicht zur Kontur hinzugefügt wurde, mit

$$\left| \mathbf{p}'_{s_{k,1}} - \mathbf{p}'_{s_{j,2}} \right| < 0.001 \quad (5.213)$$

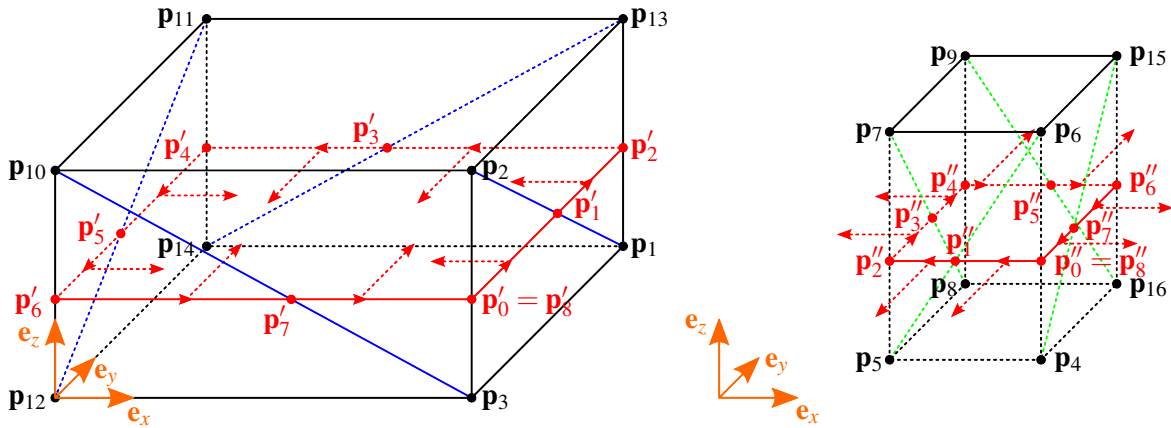
und füge es zur Kontur hinzu, solange bis

$$\left| \mathbf{p}'_{s_{k,2}} - \mathbf{p}'_{s_{1,1}} \right| < 0.001 \quad (5.214)$$

gilt. In diesem Fall ist die Kontur geschlossen. Sollte es noch weitere Schnittsegmente geben, die noch zu keiner Kontur gehören, so werden aus diesen ebenfalls geschlossene Konturen gebildet, so lange dies möglich ist.

Äußere und innere Konturen

In Abbildung 5.38 ist am Beispiel des Quaders mit einem Loch zu sehen, dass sich zwei geschlossene Konturen ergeben. Dies ist zum einen die Kontur des äußeren Quaders selbst (im Folgenden als Außenkontur bezeichnet) und zum anderen die Kontur des inneren Quaders bzw. Lochs (im Folgenden als Innenkontur bezeichnet). Die Orientierung der einzelnen Segmente



(a) Äußerer Quader. Geschlossener Polygonzug gegen den Uhrzeigersinn. (b) Innerer Quader (Loch). Geschlossener Polygonzug im Uhrzeigersinn.

Abbildung 5.38: Quader mit Loch: die Schnittsegmente wurden korrekt orientiert und geordnet. Es ergeben sich zwei geschlossene Kurven nach Definition 14, die nur aus Segmenten bestehen.

auf Basis der äußeren Dreiecksnormalen bildet also die Grundlage dafür, dass schließlich alle Außenkonturen gegen den Uhrzeigersinn und alle Innenkonturen im Uhrzeigersinn orientiert sind, wenn das Auge des Betrachters von der positiven e_z -Achse in Richtung des Ursprungs schaut.

Nachträgliche Bearbeitung der Polygonzüge

Sei $\{\mathbf{p}'_0, \mathbf{p}'_1, \dots, \mathbf{p}'_n\}$ ein geschlossener Polygonzug aus Segmenten. Von zwei benachbarten Punkten \mathbf{p}'_i und \mathbf{p}'_{i+1} kann ein Punkt gelöscht werden, falls

$$|\mathbf{p}'_{i+1} - \mathbf{p}'_i| < 0.001 \quad (5.215)$$

gilt. Dies kann solange wiederholt werden, bis für alle benachbarten Punkte

$$|\mathbf{p}'_{i+1} - \mathbf{p}'_i| \geq 0.001 \quad (5.216)$$

gilt. Auf die Bauteilqualität hat dies keinen Einfluss, jedoch auf die gesamte Stabilität der Implementierung. So kann im Folgenden der Differenzvektor zweier aufeinanderfolgender Punkte $\mathbf{p}'_{i+1} - \mathbf{p}'_i$ bedenkenlos normiert werden:

$$\mathbf{d}_i := \frac{\mathbf{p}'_{i+1} - \mathbf{p}'_i}{|\mathbf{p}'_{i+1} - \mathbf{p}'_i|}. \quad (5.217)$$

In Folge dessen ist auch die stabile Berechnung des Winkels α zwischen zwei aufeinanderfolgenden Segmenten $\mathbf{S}(\mathbf{p}'_{i-1}, \mathbf{p}'_i)$ und $\mathbf{S}(\mathbf{p}'_i, \mathbf{p}'_{i+1})$ problemlos nach Satz 1 durchführbar.

Wie in Abbildung 5.38 zu sehen ist, sind die Punkte $\mathbf{p}'_1, \mathbf{p}'_3, \mathbf{p}'_5$ und \mathbf{p}'_7 bzw. $\mathbf{p}''_1, \mathbf{p}''_3, \mathbf{p}''_5$ und \mathbf{p}''_7 überflüssig, da sich die Konturen auch aus den Segmenten $\mathbf{S}(\mathbf{p}'_0, \mathbf{p}'_2), \mathbf{S}(\mathbf{p}'_2, \mathbf{p}'_4), \mathbf{S}(\mathbf{p}'_4, \mathbf{p}'_6)$ und $\mathbf{S}(\mathbf{p}'_6, \mathbf{p}'_8)$ bzw. $\mathbf{S}(\mathbf{p}''_0, \mathbf{p}''_2), \mathbf{S}(\mathbf{p}''_2, \mathbf{p}''_4), \mathbf{S}(\mathbf{p}''_4, \mathbf{p}''_6)$ und $\mathbf{S}(\mathbf{p}''_6, \mathbf{p}''_8)$ darstellen lassen. Insbesondere werden in dieser Implementierung allgemein zwei aufeinanderfolgende Segmente $\mathbf{S}(\mathbf{p}'_{i-1}, \mathbf{p}'_i)$

und $\mathbf{S}(\mathbf{p}'_i, \mathbf{p}'_{i+1})$ einer geschlossenen Kontur mit einem Winkel $\alpha := \angle(\mathbf{p}'_{i-1} - \mathbf{p}'_i, \mathbf{p}'_i - \mathbf{p}'_{i+1})$ zu einem Segment $\mathbf{S}(\mathbf{p}'_{i-1}, \mathbf{p}'_{i+1})$ zusammengefasst, falls

$$\alpha < 0.01^\circ \quad (5.218)$$

gilt. Dies wirkt sich später auch positiv auf die Stabilität der Implementierung, insbesondere auf die Stabilität bei der Berechnung von Schnittpunkten, wie beispielsweise zwischen zwei Segmenten.

Die Menge aller auf diese Weise nachbearbeiteten Außen- und Innenkonturen für alle Schnittebenen, welche den Volumenkörper schneiden, stellt das Ergebnis des Slice-Vorgangs einer STL-Datei dar.

Bestimmung der Umlaufrichtung der geschlossenen Konturen

Nach dem oben erläuterten Prinzip können zwar die einzelnen Schnittsegmente richtig orientiert zu einer geschlossenen Kontur mit der gewünschten Umlaufrichtung zusammengefügt werden. Jedoch ist die Umlaufrichtung der geschlossenen Kurve alleine auf Basis einer korrekten Orientierung der einzelnen Segmente noch nicht bestimmbar. Da es sich aber um eine geschlossene Kurve nach Definition 14 ohne Überschneidungen handelt, kann für die Bestimmung der Umlaufrichtung Satz 18 angewandt werden. Aus dem Vorzeichen der von der Kurve eingeschlossenen Fläche ergibt sich also die Umlaufrichtung.

5.3.3 Berechnung von Offset-Konturen

Für eine saubere Oberfläche der Bauteile werden in jeder Schicht Grenzkonturen des Bauteils gedruckt. Geht man nun davon aus, dass sich die einzelnen Tropfen lotrecht von der Düse ablösen und auf die Bauplattform aufgetragen werden, dann müssen für einen maßstabgetreuen Bauprozess zunächst die originalen nach dem Slicen erhaltenen Konturen verschoben werden. Dieser Vorgang wird auch als „Contour Offsetting“ bezeichnet. Die berechneten verschobenen Konturen werden im Folgenden als Offset-Konturen bezeichnet.

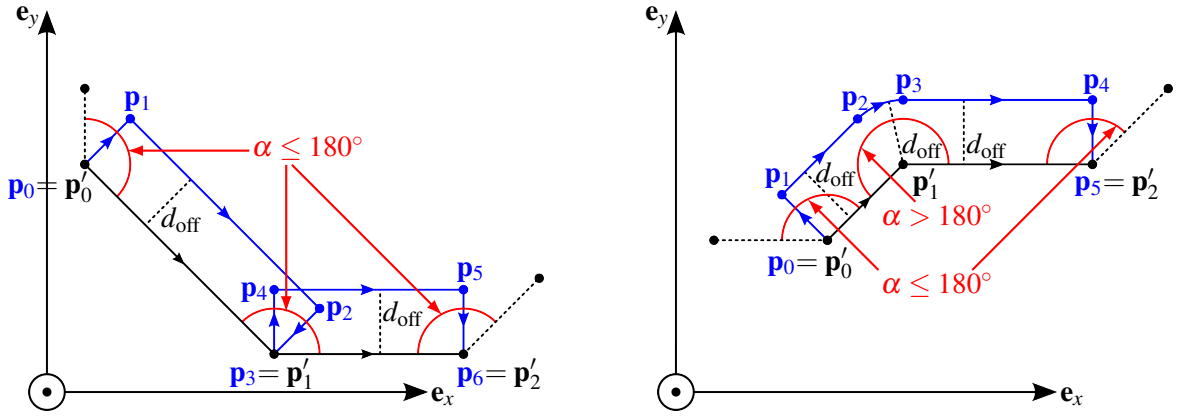
Konstruktion von Offset-Konturen

Zunächst werden auf Grundlage der beim Slicen berechneten Konturen, Offset-Konturen nach Definition 12 konstruiert. Dabei handelt es sich noch nicht um Konturen, welche letztlich gedruckt werden, sondern um eine Zwischenstufe davon, welche als Grundlage für die nachfolgenden Berechnungen der zu druckenden Konturen dient. Abbildung 5.39 zeigt das Konstruktionsprinzip, wie aus aneinandergereiht Segmenten Offset-Konturen konstruiert werden. Sei nun $\mathbf{S}(\mathbf{p}_{i-1}, \mathbf{p}_i)$ ein Segment, dann wird dies nach folgender Regel um den Offset d_{off} versetzt:

$$\mathbf{p}_{j-1} = \mathbf{p}'_{i-1} + d_{\text{off}} \left(\mathbf{e}_z \times \underbrace{\frac{\mathbf{p}'_i - \mathbf{p}'_{i-1}}{|\mathbf{p}'_i - \mathbf{p}'_{i-1}|}}_{=: \mathbf{n}_{i-1}} \right) \quad (5.219)$$

und

$$\mathbf{p}_j = \mathbf{p}'_i + d_{\text{off}} \left(\mathbf{e}_z \times \underbrace{\frac{\mathbf{p}'_i - \mathbf{p}'_{i-1}}{|\mathbf{p}'_i - \mathbf{p}'_{i-1}|}}_{=: \mathbf{n}_{i-1}} \right). \quad (5.220)$$



(a) Verschieben von Segmenten um den Offset d_{off} . Blau eingezeichnet ist die Offset-Kurve.

(b) Verschieben von Segmenten um den Offset d_{off} . Einfügen von Kreisbögen bei Winkeln $\alpha > 180^\circ$ zwischen aufeinanderfolgenden Segmenten. Hier: $\mathbf{K}_{\text{Bog}}(\mathbf{p}_2, \mathbf{p}_3) \subset \partial \mathbf{K}_{d_{\text{off}}, e_z}(\mathbf{p}'_1)$. Blau eingezeichnet ist die Offset-Kurve.

Abbildung 5.39: Konstruktion einer Kurve aus Segmenten und Kreisbögen nach Definition 12.

Es ergibt sich also ein Segment $\mathbf{S}(\mathbf{p}_{j-1}, \mathbf{p}_j)$ mit derselben Länge wie $\mathbf{S}(\mathbf{p}'_{i-1}, \mathbf{p}'_i)$. Die Definition von \mathbf{n}_{i-1} ist auf diese Weise erlaubt, da $|\mathbf{p}'_i - \mathbf{p}'_{i-1}| \neq 0$ gilt, die Konturen beim Slicen in Kapitel 5.3.2 bereits entsprechend nachbearbeitet wurden.

Sei $\mathbf{S}(\mathbf{p}_i, \mathbf{p}_{i+1})$ das auf $\mathbf{S}(\mathbf{p}_{i-1}, \mathbf{p}_i)$ folgende Segment. Durch

$$\mathbf{n} := \mathbf{n}_i \times (-\mathbf{n}_{i-1}) \quad (5.221)$$

ist ein Vektor definiert, der auf der von \mathbf{e}_x und \mathbf{e}_y definierten Fläche senkrecht steht. Nach der Definition 4 des Kreuzprodukts, zeigt der Vektor \mathbf{n} in Richtung \mathbf{e}_z , genau dann, wenn $0 < \alpha < \pi$ gilt. Das Kreuzprodukt der beiden Vektoren zeigt hingegen in entgegengesetzte Richtung von \mathbf{e}_z , genau dann, wenn $\pi < \alpha < 2\pi$ gilt. Falls speziell $\mathbf{e}_z = (0, 0, 1)^T$ gilt, dann gilt:

$$0 \leq \alpha \leq \pi \Leftrightarrow z_{\mathbf{n}} \geq 0 \quad (5.222)$$

und

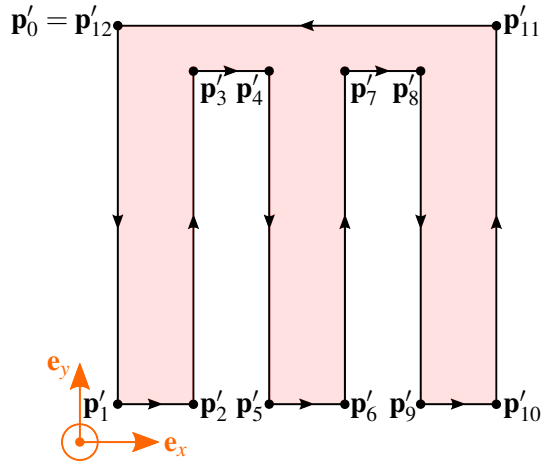
$$\pi < \alpha < 2\pi \Leftrightarrow z_{\mathbf{n}} < 0. \quad (5.223)$$

Falls also $0 \leq \alpha \leq \pi$ gilt, dann werden, wie in Abbildung 5.39a zu sehen ist, auch die Segmente $\mathbf{S}(\mathbf{p}'_{i-1}, \mathbf{p}_{j-1})$ und $\mathbf{S}(\mathbf{p}_j, \mathbf{p}'_i)$ zur Offset-Kontur hinzugefügt. Gilt hingegen $\pi < \alpha < 2\pi$, dann werden, wie in Abbildung 5.39b zu sehen ist, Kreisbögen $\mathbf{K}_{\text{Bog}}(\mathbf{p}_j, \mathbf{p}_{j+1}) \subset \partial \mathbf{K}_{d_{\text{off}}, e_z}(\mathbf{p}'_i)$ zwischen den Segmenten eingefügt.

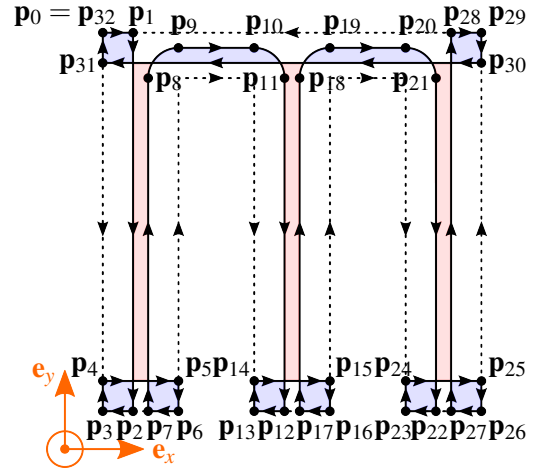
Wie in Abbildung 5.40 bzw. 5.40b zu sehen ist, ergibt sich bei der Konstruktion einer Offset-Kontur nach oben beschriebenen Schema wieder eine geschlossene Kontur γ_{e_z} nach Definition 14. Sei nun $\mathbf{E}(\mathbf{p}_j, \mathbf{p}_{j+1}) \in \gamma_{e_z}$ ein beliebiges geometrisches Element der konstruierten Kontur, dann ist auf Grundlage der oben beschriebenen Konstruktion klar, dass für jeden Punkt $\mathbf{p} \in \mathbf{E}(\mathbf{p}_j, \mathbf{p}_{j+1})$ gilt:

$$l_{\min}(\mathbf{S}(\mathbf{p}_i, \mathbf{p}_{i+1}), \mathbf{p}) \leq d_{\text{off}} \quad (5.224)$$

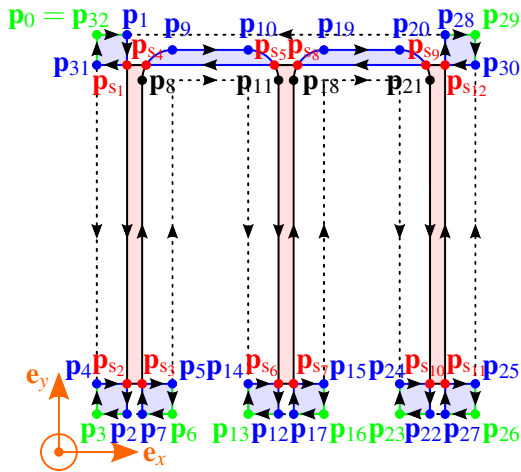
nach Satz 13, das heißt, jeder Punkt auf dem Element hat einen minimalen Abstand zum originalen Segment $\mathbf{S}(\mathbf{p}_i, \mathbf{p}_{i+1})$, der kleiner oder gleich d_{off} ist.



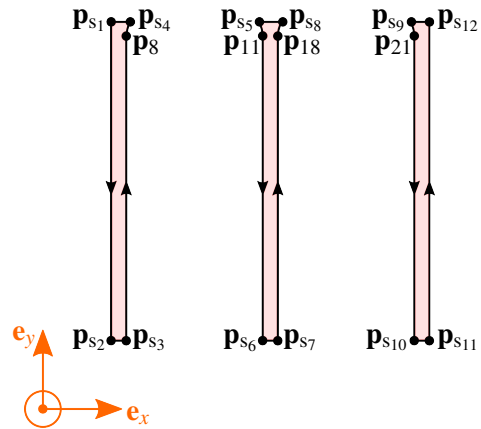
(a) Geschlossene Kontur gegen den Uhrzeigersinn als Ergebnis des Slice-Vorgangs.



(b) Konstruktion einer Offset-Kontur aus Segmenten und Kreisbögen. Kurve aus Abbildung 5.40a als Ausgangsbasis.



(c) Schnittpunktberechnungen zwischen den einzelnen geometrischen Elementen der konstruierten Offset-Kontur (rote Punkte): durch die Schnittpunkte werden die geometrischen Elemente der konstruierten Offset-Kontur (Abbildung 5.40b) nochmals zerlegt. Auf Basis von Abstandsberechnungen können geometrische Elemente detektiert werden (blaue Segmente und Kreisbögen), deren Abstand d zur Original-Kontur (Abbildung 5.40a) kleiner ist als der Offset d_{off} . Grüne Punkte waren bereits Bestandteil der originalen Kontur, die grünen Segmente verbinden lediglich die Offset-Segmente mit den Punkten der originalen Kontur.



(d) Eliminierung aller blauen und grünen geometrischen Elemente aus Abbildung 5.40c. Die verbleibenden geometrischen Elemente werden wieder zu neuen geschlossenen Offset-Konturen zusammengefügt.

Abbildung 5.40: Beispiel: Hier wird gezeigt, wie Schritt für Schritt aus einer gegebenen geschlossenen Kurve, Offset-Konturen berechnet werden.

Eliminierung von Selbstüberschneidungen

Wie in Abbildung 5.40b zu sehen ist, können sich die Elemente $\mathbf{E}(\mathbf{p}_i, \mathbf{p}_{i+1})$ der geschlossenen konstruierten Kontur je nach STL-Geometrie und Offset d_{off} gegenseitig überschneiden. Dies betrifft sowohl Überschneidungen von Segmenten mit Segmenten und Segmenten mit Kreisbögen, als auch Überschneidungen zweier Kreisbögen (dies kommt im Beispiel nicht explizit vor). Grundlage für die Implementierung sind die in Kapitel 5.2.3 aufgeführten Schnittpunktberechnungen. Abbildung 5.40c zeigt die berechneten Schnittpunkte eingezeichnet in roter Farbe. Durch diese Schnittpunkte \mathbf{p}_{s_k} werden Segmente und Kreisbögen in mehrere Segmente und Kreisbögen aufgeteilt. Beispielsweise werden aus dem Kreisbogen $\mathbf{K}_{\text{Bog}}(\mathbf{p}_8, \mathbf{p}_9)$ die Kreisbögen $\mathbf{K}_{\text{Bog}}(\mathbf{p}_8, \mathbf{p}_{s_4})$ und $\mathbf{K}_{\text{Bog}}(\mathbf{p}_{s_4}, \mathbf{p}_9)$. Aus dem Segment $\mathbf{S}(\mathbf{p}_1, \mathbf{p}_2)$ werden hingegen die Segmente $\mathbf{S}(\mathbf{p}_1, \mathbf{p}_{s_1})$, $\mathbf{S}(\mathbf{p}_{s_1}, \mathbf{p}_{s_2})$ und $\mathbf{S}(\mathbf{p}_{s_2}, \mathbf{p}_2)$. Diese neuen geometrischen Elemente werden im Folgenden separat betrachtet. Nicht betrachtet werden müssen Segmente, die als Start- oder Endpunkt einen Punkt (grün eingezeichnet) der originalen Kontur haben, beispielsweise $\mathbf{S}(\mathbf{p}_0, \mathbf{p}_1)$ oder $\mathbf{S}(\mathbf{p}_{28}, \mathbf{p}_{29})$.

Seien nun allgemein $\mathbf{S}(\mathbf{p}_i, \mathbf{p}_{i+1})$ ein beliebiges Segment und $\mathbf{K}_{\text{Bog}}(\mathbf{p}_j, \mathbf{p}_{j+1}) \subset \partial \mathbf{K}_{d_{\text{off}}, \mathbf{e}_z}(\mathbf{p}'_l)$ ein beliebiger Kreisbogen. Dann wird durch

$$\mathbf{p}_s := 0.5(\mathbf{p}_i + \mathbf{p}_{i+1}) \quad (5.225)$$

ein Punkt auf dem Segment und durch

$$\mathbf{p}_{\mathbf{K}_{\text{Bog}}} := \mathbf{p}'_l + d_{\text{off}} \frac{\frac{\mathbf{p}_j - \mathbf{p}'_l}{d_{\text{off}}} + \frac{\mathbf{p}_{j+1} - \mathbf{p}'_l}{d_{\text{off}}}}{\left| \frac{\mathbf{p}_j - \mathbf{p}'_l}{d_{\text{off}}} + \frac{\mathbf{p}_{j+1} - \mathbf{p}'_l}{d_{\text{off}}} \right|} \quad (5.226)$$

ein Punkt auf dem Kreisbogen definiert. Gilt nun für den minimalen Abstand

$$l_{\min}(\mathbf{S}(\mathbf{p}'_k, \mathbf{p}'_{k+1}), \mathbf{p}_s) \quad (5.227)$$

bzw.

$$l_{\min}(\mathbf{S}(\mathbf{p}'_k, \mathbf{p}'_{k+1}), \mathbf{p}_{\mathbf{K}_{\text{Bog}}}) \quad (5.228)$$

von Punkt \mathbf{p}_s bzw. Punkt $\mathbf{p}_{\mathbf{K}_{\text{Bog}}}$ zu den Segmenten $\mathbf{S}(\mathbf{p}'_k, \mathbf{p}'_{k+1})$, $k \in \{0, \dots, n-1\}$, der Original-Kontur

$$l_{\min}(\mathbf{S}(\mathbf{p}'_k, \mathbf{p}'_{k+1}), \mathbf{p}_s) = d_{\text{off}} \quad (5.229)$$

bzw.

$$l_{\min}(\mathbf{S}(\mathbf{p}'_k, \mathbf{p}'_{k+1}), \mathbf{p}_{\mathbf{K}_{\text{Bog}}}) = d_{\text{off}} \quad (5.230)$$

dann gehört das entsprechende geometrische Offset-Element auch zu einer fertigen Offset-Kontur. Gilt hingegen

$$l_{\min}(\mathbf{S}(\mathbf{p}'_k, \mathbf{p}'_{k+1}), \mathbf{p}_s) < d_{\text{off}} \quad (5.231)$$

bzw.

$$l_{\min}(\mathbf{S}(\mathbf{p}'_k, \mathbf{p}'_{k+1}), \mathbf{p}_{\mathbf{K}_{\text{Bog}}}) < d_{\text{off}}, \quad (5.232)$$

dann gehören $\mathbf{S}(\mathbf{p}_i, \mathbf{p}_{i+1})$ bzw. $\mathbf{K}_{\text{Bog}}(\mathbf{p}_j, \mathbf{p}_{j+1})$ nicht zu einer fertigen Offset-Kontur, sondern müssen eliminiert werden, da der Abstand zur Original-Kontur zu klein ist. Innerhalb der Implementierung ist nicht d_{off} diese strikte Grenze, sondern es wird eine Toleranz von 0.001^2 gewährt, das heißt:

$$d_{\text{off}} - 0.001^2. \quad (5.233)$$

Geometrische Elemente $\mathbf{E}(\mathbf{p}_j, \mathbf{p}_{j+1})$, die eliminiert werden müssen, sind in Abbildung 5.40 blau eingezeichnet. Letzten Endes müssen alle verbleibenden geometrischen Elemente, ähnlich wie in Kapitel 5.3.2 wieder zu geschlossenen Konturen zusammengefügt werden. Das Ergebnis dieses Prozesses ist in Abbildung 5.40d zu sehen.

Eliminierung von Überschneidungen innerer Konturen

Bei der Eliminierung globaler Überschneidungen geht es um die Auflösung von Überlappungen verschiedener Konturen. Berücksichtigt werden müssen dabei jeweils immer alle Konturen mit derselben ID (siehe Kapitel 5.2.4), das heißt, alle Konturen, die zur selben Zusammenhangskomponente gehören. Abbildung 5.41 zeigt ein Beispiel für die Eliminierung verschiedener Konturen. Zunächst werden dabei, wie in Kapitel 5.2.3 gezeigt, die Schnittpunkte zwischen inneren Konturen, das heißt, zwischen allen geometrischen Elementen der inneren Konturen, berechnet. In Abbildung 5.41b besteht eine innere Kontur aus den geometrischen Elementen

$$\left\{ \mathbf{K}_{\text{Bog}}(\mathbf{p}_{s_2}, \mathbf{p}_9), \mathbf{S}(\mathbf{p}_9, \mathbf{p}_{10}), \mathbf{K}_{\text{Bog}}(\mathbf{p}_{10}, \mathbf{p}_{11}), \mathbf{S}(\mathbf{p}_{11}, \mathbf{p}_{12}), \mathbf{K}_{\text{Bog}}(\mathbf{p}_{12}, \mathbf{p}_5), \right. \\ \left. \mathbf{S}(\mathbf{p}_5, \mathbf{p}_6), \mathbf{K}_{\text{Bog}}(\mathbf{p}_6, \mathbf{p}_7), \mathbf{S}(\mathbf{p}_7, \mathbf{p}_8), \mathbf{K}_{\text{Bog}}(\mathbf{p}_8, \mathbf{p}_{s_1}), \mathbf{K}_{\text{Bog}}(\mathbf{p}_{s_1}, \mathbf{p}_{s_2}) \right\}, \quad (5.234)$$

die andere innere Kontur besteht aus den geometrischen Elementen

$$\left\{ \mathbf{K}_{\text{Bog}}(\mathbf{p}_{s_1}, \mathbf{p}_{20}), \mathbf{S}(\mathbf{p}_{20}, \mathbf{p}_{21}), \mathbf{K}_{\text{Bog}}(\mathbf{p}_{21}, \mathbf{p}_{14}), \mathbf{S}(\mathbf{p}_{14}, \mathbf{p}_{15}), \mathbf{K}_{\text{Bog}}(\mathbf{p}_{15}, \mathbf{p}_{16}), \right. \\ \left. \mathbf{S}(\mathbf{p}_{16}, \mathbf{p}_{17}), \mathbf{K}_{\text{Bog}}(\mathbf{p}_{17}, \mathbf{p}_{18}), \mathbf{S}(\mathbf{p}_{18}, \mathbf{p}_{19}), \mathbf{K}_{\text{Bog}}(\mathbf{p}_{19}, \mathbf{p}_{s_2}), \mathbf{K}_{\text{Bog}}(\mathbf{p}_{s_2}, \mathbf{p}_{s_1}) \right\}. \quad (5.235)$$

Durch diese Schnittpunkte werden die geometrischen Elemente

$$\mathbf{S}(\mathbf{p}_i, \mathbf{p}_{i+1}) \text{ bzw. } \mathbf{K}_{\text{Bog}}(\mathbf{p}_j, \mathbf{p}_{j+1}) \subset \partial \mathbf{K}_{d_{\text{off}}, \mathbf{e}_z}(\mathbf{p}'_l), \quad (5.236)$$

wie bereits bei den Selbstüberschneidungen, in mehrere Elemente aufgetrennt. Hierbei muss aber im Folgenden nicht auf jedes einzelne geometrische Element Acht gegeben werden. Entscheidend ist die Aufspaltung an den Schnittpunkten: So entstehen in Abbildung 5.41b, die nicht geschlossenen Kurven

$$\left\{ \mathbf{K}_{\text{Bog}}(\mathbf{p}_{s_2}, \mathbf{p}_9), \mathbf{S}(\mathbf{p}_9, \mathbf{p}_{10}), \mathbf{K}_{\text{Bog}}(\mathbf{p}_{10}, \mathbf{p}_{11}), \mathbf{S}(\mathbf{p}_{11}, \mathbf{p}_{12}), \mathbf{K}_{\text{Bog}}(\mathbf{p}_{12}, \mathbf{p}_5), \right. \\ \left. \mathbf{S}(\mathbf{p}_5, \mathbf{p}_6), \mathbf{K}_{\text{Bog}}(\mathbf{p}_6, \mathbf{p}_7), \mathbf{S}(\mathbf{p}_7, \mathbf{p}_8), \mathbf{K}_{\text{Bog}}(\mathbf{p}_8, \mathbf{p}_{s_1}) \right\} \quad (5.237)$$

und

$$\left\{ \mathbf{K}_{\text{Bog}}(\mathbf{p}_{s_1}, \mathbf{p}_{s_2}) \right\} \quad (5.238)$$

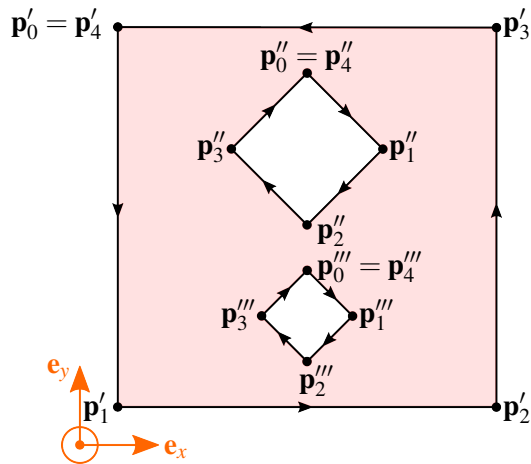
bzw.

$$\left\{ \mathbf{K}_{\text{Bog}}(\mathbf{p}_{s_1}, \mathbf{p}_{20}), \mathbf{S}(\mathbf{p}_{20}, \mathbf{p}_{21}), \mathbf{K}_{\text{Bog}}(\mathbf{p}_{21}, \mathbf{p}_{14}), \mathbf{S}(\mathbf{p}_{14}, \mathbf{p}_{15}), \right. \\ \left. \mathbf{K}_{\text{Bog}}(\mathbf{p}_{15}, \mathbf{p}_{16}), \mathbf{S}(\mathbf{p}_{16}, \mathbf{p}_{17}), \mathbf{K}_{\text{Bog}}(\mathbf{p}_{17}, \mathbf{p}_{18}), \mathbf{S}(\mathbf{p}_{18}, \mathbf{p}_{19}), \mathbf{K}_{\text{Bog}}(\mathbf{p}_{19}, \mathbf{p}_{s_2}) \right\} \quad (5.239)$$

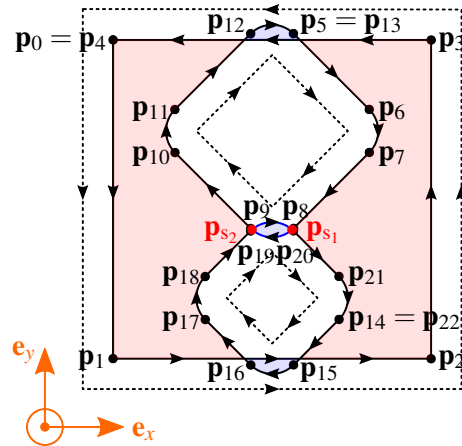
und

$$\left\{ \mathbf{K}_{\text{Bog}}(\mathbf{p}_{s_2}, \mathbf{p}_{s_1}) \right\}. \quad (5.240)$$

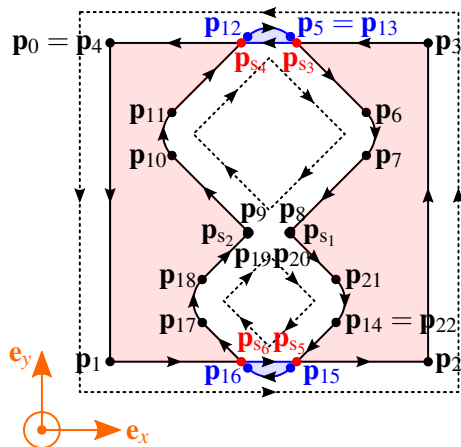
Die Start- und Endpunkte der nicht geschlossenen Kurven sind also jeweils die Schnittpunkte. Die Idee bei der Eliminierung von Überschneidungen zwischen inneren Konturen ist nun, alle geometrischen Elemente zu eliminieren, welche sich innerhalb mindestens einer anderen geschlossenen inneren Kontur befinden. Im Beispiel sind dies die beiden Kreisbögen $\left\{ \mathbf{K}_{\text{Bog}}(\mathbf{p}_{s_1}, \mathbf{p}_{s_2}) \right\}$ und $\left\{ \mathbf{K}_{\text{Bog}}(\mathbf{p}_{s_2}, \mathbf{p}_{s_1}) \right\}$ aus (5.238) und (5.240), die jeweils blau eingezeichnet sind. Alle geometrischen Elemente aus (5.237) und (5.239) befinden sich hingegen nicht innerhalb einer anderen geschlossenen Kontur (Loch) und müssen daher nicht eliminiert werden.



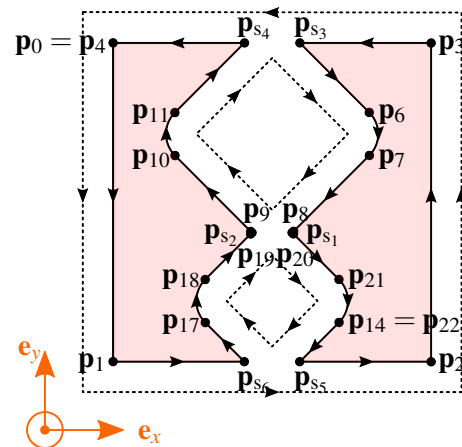
(a) Mehrere geschlossene Kontur gegen den Uhrzeigersinn als Ergebnis des Slice-Vorgangs.



(b) Offset-Konturen für Konturen aus Abbildung 5.40a. Zunächst müssen die Überschneidungen (rote Punkte) innerer Konturen aufgelöst werden. Geometrische Elemente (blau eingezeichnet), welche sich innerhalb einer anderen inneren Kontur befinden, müssen eliminiert werden.



(c) Ergebnis von Abbildung 5.41b. Geometrische Elemente, die eliminiert werden müssen, sind blau eingezeichnet. Dies sind geometrische Elemente der äußeren Kontur, welche sich innerhalb einer inneren Kontur befinden und geometrische Elemente einer inneren Kontur, welche sich außerhalb der äußeren Kontur befinden.



(d) Ergebnis, nachdem auch Überschneidungen der inneren Konturen mit der äußeren Kontur aufgelöst wurden.

Abbildung 5.41: Beispiel: Hier wird gezeigt, wie Schritt für Schritt aus mehreren gegebenen geschlossenen Kurven, Offset-Konturen berechnet werden. Dabei ist es wichtig, Überschneidungen verschiedener Konturen richtig aufzulösen. Mit Hilfe von Windungszahlen kann entschieden werden, was sich innen und was sich außen befindet.

Aus den nicht geschlossenen Kurven $\gamma'_{e_z,k}$, welche nicht eliminiert wurden, können jetzt wieder geschlossene Konturen zusammengefügt werden, wie bereits auch bei Kapitel 5.3.2 näher erläutert. Für das Beispiel ist das Ergebnis in Abbildung 5.41d zu sehen.

Seien nun allgemein $\gamma'_{e_z,k}$, $k \in \{1, \dots, m\}$, nicht geschlossene Kurven mit Startpunkten $\mathbf{p}_{s_1,k}$ bzw. Endpunkten $\mathbf{p}_{s_2,k}$, welche nach dem obigen Prinzip durch Aufspaltung der geschlossenen inneren Kurven $\gamma_{e_z,l}$, $l \in \{1, \dots, n\}$ an den Schnittpunkten entstanden sind. Um zu berechnen welche Kurven $\gamma'_{e_z,k}$ sich im inneren einer anderen geschlossenen inneren Kontur befinden, wird folgendermaßen vorgegangen. Für einen Punkt $\mathbf{p} \in \gamma'_{e_z,k} \setminus \{\mathbf{p}_{s_1,k}, \mathbf{p}_{s_2,k}\}$, $\mathbf{p} \in \gamma_{e_z,t}$ werden nach Satz 23 die Windungszahlen

$$\eta(\gamma_{e_z,l}, \mathbf{p}), \quad l \in \{1, \dots, n\} \setminus \{t\} \quad (5.241)$$

berechnet und aufsummiert:

$$\eta_k^{\text{inner}}(\mathbf{p}) := \sum_{l \neq t} \eta(\gamma_{e_z,l}, \mathbf{p}). \quad (5.242)$$

Nach Satz 24 gilt entweder $\eta(\gamma_{e_z,l}, \mathbf{p}) = 0$, genau dann, falls \mathbf{p} außerhalb von $\gamma_{e_z,l}$ liegt oder $\eta(\gamma_{e_z,l}, \mathbf{p}) = -1$, genau dann, falls \mathbf{p} innerhalb von $\gamma_{e_z,l}$ liegt. Das heißt also, es gilt

$$\eta_k^{\text{inner}}(\mathbf{p}) = 0 \Leftrightarrow \mathbf{p} \text{ liegt innerhalb keiner Kurve } \gamma_{e_z,l} \quad (l \neq t). \quad (5.243)$$

Andernfalls gilt

$$\eta_k^{\text{inner}}(\mathbf{p}) < 0 \Leftrightarrow \mathbf{p} \text{ liegt innerhalb mindestens einer Kurve } \gamma_{e_z,l} \quad (l \neq t). \quad (5.244)$$

Genau genommen, gibt $\eta_k^{\text{inner}}(\mathbf{p})$ also an innerhalb wie vieler geschlossener innerer Konturen der Punkt \mathbf{p} liegt. Ist also $\eta_k^{\text{inner}}(\mathbf{p}) < 0$, so wird die nicht geschlossene Kurve $\gamma'_{e_z,k}$ eliminiert, ansonsten bleibt sie bestehen. Dies gilt für alle $k \in \{1, \dots, m\}$. Für den Punkt \mathbf{p} kann dabei ein Start- bzw. Endpunkt eines geometrischen Elements von $\gamma'_{e_z,k}$ gewählt werden, der ungleich dem Startpunkt $\mathbf{p}_{s_1,k}$ bzw. Endpunkt $\mathbf{p}_{s_2,k}$ der ganzen Kurve ist. Ist dies nicht möglich, dann kann wie bereits oben bei der Eliminierung der Selbstüberschneidungen gezeigt, ein Punkt zwischen dem Start- und Endpunkt berechnet werden.

Eliminierung von Überschneidungen innerer Konturen mit der äußeren Kontur

Nach der Eliminierung von Überschneidungen zwischen inneren Konturen, kann es nur noch Überschneidungen zwischen den neuen inneren Konturen und der äußeren Kontur geben. Dies ist ebenfalls in Abbildung 5.41c zu sehen. Für die Schnittpunktberechnungen der einzelnen geometrischen Elemente sei wiederum auf Kapitel 5.2.3 verwiesen. In Abbildung 5.41 gibt es die äußere Kontur bestehend aus den Elementen

$$\{\mathbf{S}(\mathbf{p}_0, \mathbf{p}_1), \mathbf{S}(\mathbf{p}_1, \mathbf{p}_2), \mathbf{S}(\mathbf{p}_2, \mathbf{p}_3), \mathbf{S}(\mathbf{p}_3, \mathbf{p}_0)\} \quad (5.245)$$

und die innere Kontur bestehend aus den Elementen

$$\begin{aligned} & \{\mathbf{S}(\mathbf{p}_5, \mathbf{p}_6), \mathbf{K}_{\text{Bog}}(\mathbf{p}_6, \mathbf{p}_7), \mathbf{S}(\mathbf{p}_7, \mathbf{p}_8), \mathbf{K}_{\text{Bog}}(\mathbf{p}_8, \mathbf{p}_{s_1}) \\ & \mathbf{K}_{\text{Bog}}(\mathbf{p}_{s_1}, \mathbf{p}_{20}), \mathbf{S}(\mathbf{p}_{20}, \mathbf{p}_{21}), \mathbf{K}_{\text{Bog}}(\mathbf{p}_{21}, \mathbf{p}_{14}), \mathbf{S}(\mathbf{p}_{14}, \mathbf{p}_{15}), \\ & \mathbf{K}_{\text{Bog}}(\mathbf{p}_{15}, \mathbf{p}_{16}), \mathbf{S}(\mathbf{p}_{16}, \mathbf{p}_{17}), \mathbf{K}_{\text{Bog}}(\mathbf{p}_{17}, \mathbf{p}_{18}), \mathbf{S}(\mathbf{p}_{18}, \mathbf{p}_{19}), \mathbf{K}_{\text{Bog}}(\mathbf{p}_{19}, \mathbf{p}_{s_2}) \\ & \mathbf{K}_{\text{Bog}}(\mathbf{p}_{s_2}, \mathbf{p}_9), \mathbf{S}(\mathbf{p}_9, \mathbf{p}_{10}), \mathbf{K}_{\text{Bog}}(\mathbf{p}_{10}, \mathbf{p}_{11}), \mathbf{S}(\mathbf{p}_{11}, \mathbf{p}_{12}), \mathbf{K}_{\text{Bog}}(\mathbf{p}_{12}, \mathbf{p}_5)\} \end{aligned} \quad (5.246)$$

Durch die Schnittpunkte werden die geometrischen Elemente

$$\mathbf{S}(\mathbf{p}_i, \mathbf{p}_{i+1}) \text{ bzw. } \mathbf{K}_{\text{Bog}}(\mathbf{p}_j, \mathbf{p}_{j+1}) \subset \partial \mathbf{K}_{\text{diff}, e_z}(\mathbf{p}'_l), \quad (5.247)$$

wie bereits bei den Überschneidungen der inneren Konturen, in mehrere nicht geschlossene Kurven aufgetrennt. So entstehen in Abbildung 5.41c, die nicht geschlossenen Kurven

$$\{\mathbf{S}(\mathbf{p}_{s_4}, \mathbf{p}_0), \mathbf{S}(\mathbf{p}_0, \mathbf{p}_1), \mathbf{S}(\mathbf{p}_1, \mathbf{p}_{s_6})\}, \quad (5.248)$$

$$\{\mathbf{S}(\mathbf{p}_{s_6}, \mathbf{p}_{s_5})\}, \quad (5.249)$$

$$\{\mathbf{S}(\mathbf{p}_{s_5}, \mathbf{p}_2), \mathbf{S}(\mathbf{p}_2, \mathbf{p}_3), \mathbf{S}(\mathbf{p}_3, \mathbf{p}_{s_3})\} \quad (5.250)$$

und

$$\{\mathbf{S}(\mathbf{p}_{s_3}, \mathbf{p}_{s_4})\} \quad (5.251)$$

bzw.

$$\{\mathbf{S}(\mathbf{p}_{s_3}, \mathbf{p}_6), \mathbf{K}_{\text{Bog}}(\mathbf{p}_6, \mathbf{p}_7), \mathbf{S}(\mathbf{p}_7, \mathbf{p}_8), \mathbf{K}_{\text{Bog}}(\mathbf{p}_8, \mathbf{p}_{s_1}), \mathbf{K}_{\text{Bog}}(\mathbf{p}_{s_1}, \mathbf{p}_{20}), \mathbf{S}(\mathbf{p}_{20}, \mathbf{p}_{21}), \mathbf{K}_{\text{Bog}}(\mathbf{p}_{21}, \mathbf{p}_{14}), \mathbf{S}(\mathbf{p}_{14}, \mathbf{p}_{s_5})\}, \quad (5.252)$$

$$\{\mathbf{S}(\mathbf{p}_{s_5}, \mathbf{p}_{15}), \mathbf{K}_{\text{Bog}}(\mathbf{p}_{15}, \mathbf{p}_{16}), \mathbf{S}(\mathbf{p}_{16}, \mathbf{p}_{s_6})\}, \quad (5.253)$$

$$\{\mathbf{S}(\mathbf{p}_{s_6}, \mathbf{p}_{17}), \mathbf{K}_{\text{Bog}}(\mathbf{p}_{17}, \mathbf{p}_{18}), \mathbf{S}(\mathbf{p}_{18}, \mathbf{p}_{19}), \mathbf{K}_{\text{Bog}}(\mathbf{p}_{19}, \mathbf{p}_{s_2}), \mathbf{K}_{\text{Bog}}(\mathbf{p}_{s_2}, \mathbf{p}_9), \mathbf{S}(\mathbf{p}_9, \mathbf{p}_{10}), \mathbf{K}_{\text{Bog}}(\mathbf{p}_{10}, \mathbf{p}_{11}), \mathbf{S}(\mathbf{p}_{11}, \mathbf{p}_{s_4})\} \quad (5.254)$$

und

$$\{\mathbf{S}(\mathbf{p}_{s_4}, \mathbf{p}_{12}), \mathbf{K}_{\text{Bog}}(\mathbf{p}_{12}, \mathbf{p}_5), \mathbf{S}(\mathbf{p}_5, \mathbf{p}_{s_3})\}. \quad (5.255)$$

Die Elemente in (5.249) und (5.251) gehören zur äußeren Kontur, befinden sich aber innerhalb der inneren Kontur und müssen deshalb eliminiert werden. Die Elemente in (5.253) und (5.255) gehören hingegen zur inneren Kontur, befinden sich jedoch außerhalb der äußeren Kontur und müssen deshalb ebenfalls eliminiert werden. Aus den restlichen Elementen in (5.248), (5.250), (5.252) und (5.254) werden hingegen analog zum Prinzip beim Slicen in Kapitel 5.3.2 zwei neue geschlossene (äußere) Konturen zusammengefügt. Dies ist in Abbildung 5.41d zu sehen.

Seien nun allgemein γ'_{e_z, k_1} , $k_1 \in \{1, \dots, m_1\}$, nicht geschlossene Kurven bestehend aus Elementen der geschlossenen äußeren Kontur $\gamma_{e_z, 1}$ und γ''_{e_z, k_2} , $k_2 \in \{m_1 + 1, \dots, m_2\}$, nicht geschlossene Kurven bestehend aus Elementen der geschlossenen inneren Konturen $\gamma_{e_z, l}$, $l \in \{2, \dots, n\}$, mit Startpunkten \mathbf{p}_{s_1, k_1} bzw. \mathbf{p}_{s_1, k_2} und Endpunkten \mathbf{p}_{s_2, k_1} bzw. \mathbf{p}_{s_2, k_2} , welche nach dem obigen Prinzip durch Aufspaltung der geschlossenen Kurven an den Schnittpunkten entstanden sind. Um zu berechnen welche Teilkurven γ'_{e_z, k_1} der geschlossenen äußeren Kontur sich innerhalb der geschlossenen inneren Konturen $\gamma_{e_z, l}$, $l \in \{2, \dots, n\}$, befinden, wird folgendermaßen vorgegangen. Für einen Punkt $\mathbf{p} \in \gamma'_{e_z, k_1} \setminus \{\mathbf{p}_{s_1, k_1}, \mathbf{p}_{s_2, k_1}\}$, werden nach Satz 23 die Windungszahlen

$$\eta(\gamma_{e_z, l}, \mathbf{p}), \quad l \in \{2, \dots, n\} \quad (5.256)$$

berechnet und aufsummiert:

$$\eta_{k_1}^{\text{inner}}(\mathbf{p}) := \sum_{l \neq 1} \eta(\gamma_{e_z, l}, \mathbf{p}). \quad (5.257)$$

Nach Satz 24 gilt entweder $\eta(\gamma_{e_z, l}, \mathbf{p}) = 0$, genau dann, falls \mathbf{p} außerhalb von $\gamma_{e_z, l}$ liegt oder $\eta(\gamma_{e_z, l}, \mathbf{p}) = -1$, genau dann, falls \mathbf{p} innerhalb von $\gamma_{e_z, l}$ liegt. Das heißt also, es gilt

$$\eta_{k_1}^{\text{inner}}(\mathbf{p}) = 0 \Leftrightarrow \mathbf{p} \text{ liegt innerhalb keiner Kurve } \gamma_{e_z, l}, \mathbf{p} \ (l \neq 1). \quad (5.258)$$

Andernfalls gilt

$$\eta_{k_1}^{\text{inner}}(\mathbf{p}) < 0 \Leftrightarrow \mathbf{p} \text{ liegt innerhalb mindestens einer Kurve } \gamma_{e_z, l}, \mathbf{p} \ (l \neq 1). \quad (5.259)$$

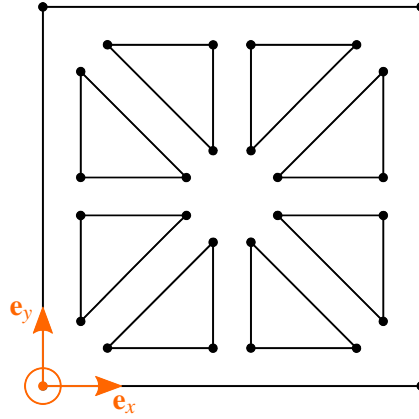


Abbildung 5.42: Beispiel eines Quaders mit 8 kleinen Löchern in Form von Dreiecken.

Genau genommen, gibt $\eta_{k_1}^{\text{inner}}(\mathbf{p})$ also an, innerhalb wie vieler geschlossener innerer Konturen der Punkt \mathbf{p} liegt. Ist also $\eta_{k_1}^{\text{inner}}(\mathbf{p}) < 0$, so wird die nicht geschlossene Kurve γ'_{e_z, k_1} eliminiert, ansonsten bleibt sie bestehen. Dies gilt für alle $k_1 \in \{1, \dots, m_1\}$.

Um hingegen zu berechnen welche Teilkurven γ''_{e_z, k_2} der geschlossenen inneren Konturen sich außerhalb der geschlossenen äußeren Kontur $\gamma_{e_z, 1}$ befinden, wird folgendermaßen vorgegangen. Für einen Punkt $\mathbf{p} \in \gamma''_{e_z, k_2} \setminus \{\mathbf{p}_{s_1, k_2}, \mathbf{p}_{s_2, k_2}\}$, wird nach Satz 23 die Windungszahl

$$\eta(\gamma_{e_z, 1}, \mathbf{p}) \quad (5.260)$$

berechnet. Nach Satz 24 gilt entweder

$$\eta(\gamma_{e_z, 1}, \mathbf{p}) = 0 \Leftrightarrow \mathbf{p} \text{ liegt außerhalb von } \gamma_{e_z, 1} \quad (5.261)$$

oder

$$\eta(\gamma_{e_z, 1}, \mathbf{p}) = 1 \Leftrightarrow \mathbf{p} \text{ liegt innerhalb von } \gamma_{e_z, 1}. \quad (5.262)$$

Ist also $\eta(\gamma_{e_z, 1}, \mathbf{p}) = 0$, so wird die nicht geschlossene Kurve γ''_{e_z, k_2} eliminiert, ansonsten bleibt sie bestehen. Dies gilt für alle $k_2 \in \{m_1 + 1, \dots, m_2\}$.

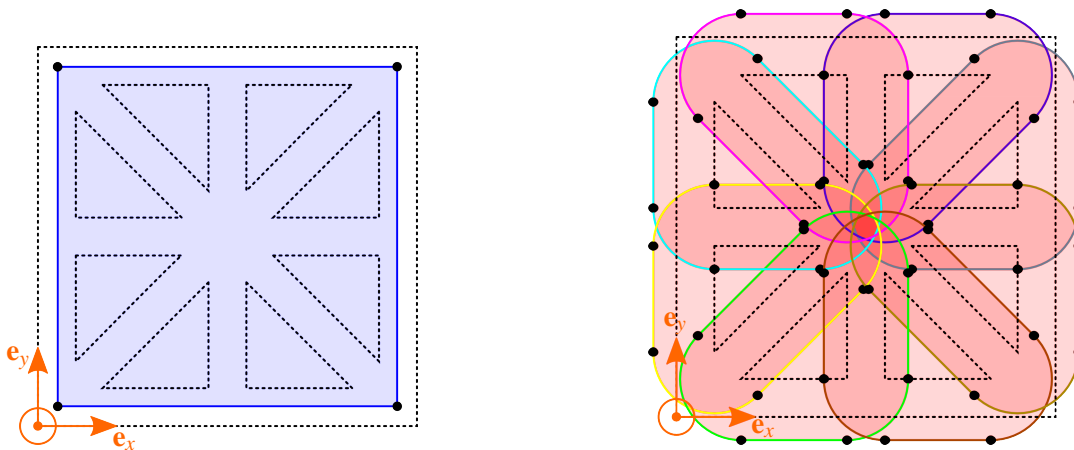
Für den Punkt \mathbf{p} kann dabei ein Start- bzw. Endpunkt eines geometrischen Elements von γ'_{e_z, k_1} bzw. γ''_{e_z, k_2} gewählt werden, der ungleich dem Start- bzw. Endpunkt der ganzen Kurve ist. Ist dies nicht möglich, dann kann wie bereits oben bei der Eliminierung der Selbstüberschneidungen gezeigt, ein Punkt zwischen dem Start- und Endpunkt berechnet werden.

Abbruch-Kriterium für Offset-Konturen

Anhand eines Beispiels soll nun erläutert werden, wann Konturen nicht weiter verschoben werden dürfen. Dies soll am Beispiel in Abbildung 5.42 demonstriert werden. Hierbei wird die äußere Kontur des Quadrats um $d_{\text{off}, 1}$ versetzt (siehe Abbildung 5.43a). Die inneren Konturen der Dreiecke werden hingegen um $d_{\text{off}, 2}$ verschoben (siehe Abbildung 5.43b). Es gilt:

$$d_{\text{off}, 1} < d_{\text{off}, 2}. \quad (5.263)$$

Werden nun die inneren Konturen nach dem selben Prinzip, wie auch in Abbildung 5.41b und 5.41c zu sehen, neu zusammengefügt, dann ergibt sich eine neue innere Kontur (siehe Abbildung 5.44a). In Abbildung 5.44b ist hingegen zu sehen, dass die neue innere Kontur komplett



- (a) Offset der äußeren Kontur um den Offset $d_{\text{off},1}$. Blau eingezeichnet ist die Offset-Kontur und die innere Fläche der Kurve.
- (b) Offset der inneren Konturen um den Offset $d_{\text{off},2}$. Rot eingezeichnet sind die Offset-Konturen und die innere Fläche der Kurven.

Abbildung 5.43: Versatz der Konturen mit verschiedenem Offset. Siehe Beispiel in Abbildung 5.42.

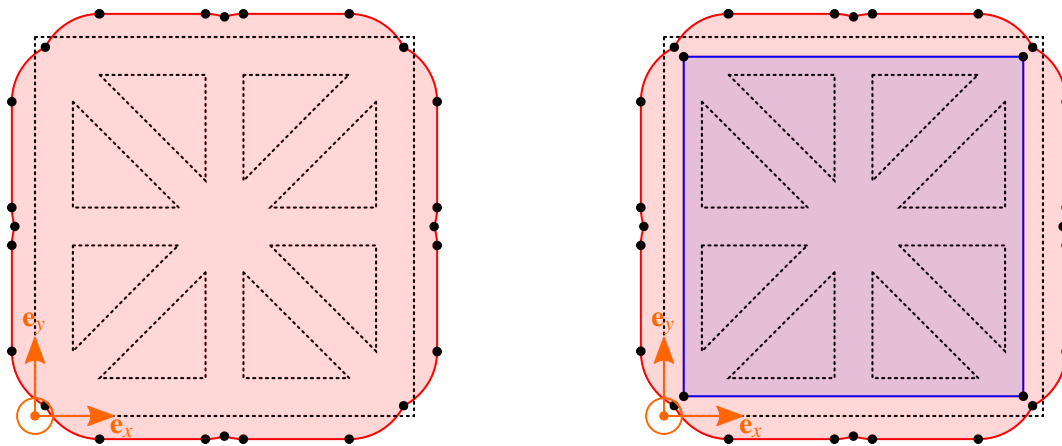
außerhalb der neuen äußeren Kontur liegt. In diesem Fall haben die beiden Konturen keine Schnittpunkte und werden dadurch nicht aufgeteilt. Betrachtet man allerdings allgemein anstatt nicht geschlossener Teilkurven die kompletten geschlossenen Kurven, dann gelten weiterhin die Aussagen in den Gleichungen (5.258), (5.259), (5.261) und (5.262). Auf diese Weise kann festgestellt werden, dass beide Konturen eliminiert werden müssen, da die Wandstärke des Bauteils zu klein, bzw. der Offset zu groß ist.

Bestimmung weiterer Eigenschaften der berechneten Konturen

Bei den fertig berechneten Konturen handelt es sich wiederum um geschlossene Konturen nach Definition 14 ohne Überschneidungen. Somit kann Satz 21 angewandt werden. Aus dem Vorzeichen der von der Kurve eingeschlossenen Fläche ergibt sich also die Umlaufrichtung. Den neu berechneten Konturen kann außerdem wieder eine ID zugeordnet werden.

Konzept Konturdruck

Nachdem die Grundlagen zur Berechnung von Offset-Konturen erläutert wurden, stellt sich nun die Frage, welcher Offset d_{off} am Besten zur Berechnung verwendet wird. Ein Beispiel ist hierfür in Abbildung 5.45 zu sehen. Dieses Modell beruht auf Tropfen der Tropfenbreite B . Zunächst werden die originalen Konturen des Slice-Prozesses aus Kapitel 5.3.2 um $d_{\text{off}} = \frac{B}{2}$ verschoben. Hierbei werden, wie bereits oben erläutert, äußere Konturen nach innen und innere Konturen nach außen verschoben. Jeder Punkt auf der zu druckenden Offset-Kontur hat also einen Abstand von $d_{\text{off}} = \frac{B}{2}$ zu Original-Kontur. Für einen maßstabgetreuen Bauprozess haben die Konturen des zweiten Niveaus alle einen Offset von $d_{\text{off}} = \frac{3B}{2}$. In der Regel werden diese zwei Offset-Kontur-Niveaus beim Druck eines Bauteils verwendet. Es lassen sich jedoch auch weitere Konturen drucken, falls gewünscht. Die Reihenfolge beim Druck der Konturen kann geändert werden, das heißt, es können sowohl erst alle Konturen mit einem Offset von $d_{\text{off}} = \frac{B}{2}$ und dann alle Konturen mit einem Offset von $d_{\text{off}} = \frac{3B}{2}$ gedruckt, als auch genauso umgekehrt. Für die folgende Berechnung der Füllung wird ein weiteres Offset-Kontur-Niveau berechnet. Im



- (a) Nach erläuterten Prinzip neu berechnete innere Kontur. (b) Innere Kontur liegt komplett außerhalb der äußeren Kontur.

Abbildung 5.44: Beispiel für die Eliminierung von Konturen: dient auch als Abbruchkriterium. Siehe Beispiele in Abbildung 5.42 und 5.43.

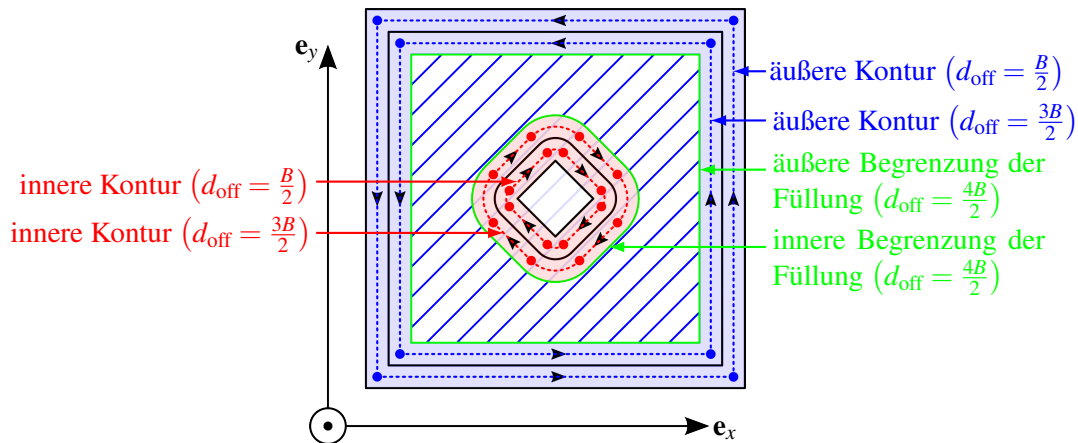


Abbildung 5.45: Das Beispiel zeigt die berechneten Offset-Konturen. Blau gestrichelt sind die Verfahrensweg beim Druck der äußeren Konturen. In diesem Fall sind das zwei Konturen. Die Konturen haben einen Offset von $d_{\text{off}} = \frac{B}{2}$ bzw. $d_{\text{off}} = \frac{3B}{2}$. Der blaue Bereich zeigt jeweils das maximale Ausmaß, das durch die Tropfenbreite B beim Druck der äußeren Konturen erreicht wird. Rot gestrichelt sind hingegen die Verfahrensweg beim Druck der inneren Konturen. Auch in diesem Fall sind das zwei Konturen. Die Konturen haben ebenso einen Offset von $d_{\text{off}} = \frac{B}{2}$ bzw. $d_{\text{off}} = \frac{3B}{2}$. Der rote Bereich zeigt jeweils das maximale Ausmaß, das durch die Tropfenbreite B beim Druck der inneren Konturen erreicht wird. Grün eingezeichnet sind die Konturen zur Begrenzung der Füllung. In diesem Fall gilt $d_{\text{off}} = \frac{4B}{2}$, das heißt, die Füllung überlappt sich um eine halbe Tropfenbreite $\frac{B}{2}$ mit der innersten der äußeren bzw. der äußersten der inneren Konturen. Dieser Überlappfaktor lässt sich auch entsprechend ändern: $f_{\text{ol}}B, 0 \leq f_{\text{ol}} \leq 1$.

Beispiel ist es das dritte Offset-Kontur-Niveau mit einem Offset von $d_{\text{off}} = \frac{4B}{2}$. Dies bedeutet, dass es einen Überlapp f_{ol} von 50% zwischen Füllung und zweitem Offset-Kontur-Niveau gibt, da der Abstand zwischen dem zweiten und dritten Niveau $\Delta d_{\text{off}} = \frac{B}{2}$ beträgt. Der Überlapp lässt

sich jedoch auch hier nahezu beliebig variieren. Es ist zu beachten, dass die Grenzkonturen für die Füllung selbst nicht gedruckt werden, sondern nur als Grundlage für die Berechnung der Füllung dienen.

5.3.4 Füllung der Schichten

Als Grundlage für die Berechnung der Füllung einer Schicht dienen die in Kapitel 5.3.3 berechneten Grenzkonturen, die einen definierten Überlapp f_{ol} zu den gedruckten Konturen haben. Diese geschlossenen Grenzkonturen $\gamma_{\mathbf{e}_z, k}$, $k \in \{1, \dots, n\}$, können, wie alle Offset-Konturen, aus Segmenten $\mathbf{S}(\mathbf{p}_i, \mathbf{p}_{i+1})$ und Kreisbögen $\mathbf{K}_{\text{Bog}}(\mathbf{p}_j, \mathbf{p}_{j+1}) \subset \partial \mathbf{K}_{d_{\text{off}}, \mathbf{e}_z}(\mathbf{p}'_j)$ bestehen.

Definition von Schraffurlinien

Die Füllung wird innerhalb dieser Grenzkonturen in Form von Schraffursegmenten gedruckt. Dabei kann von Schicht zu Schicht der Versatzwinkel variiert werden. Geht man also von einem Schraffurwinkel α_s und einem Versatzwinkel α_v von Schicht zu Schicht aus, so ergibt sich für die m -te Schicht ein Schraffurwinkel von

$$\alpha = \alpha_s + (m - 1) \alpha_v. \quad (5.264)$$

Für den Richtungsvektor der Schraffurlinie ergibt sich somit nach Satz 7:

$$\mathbf{d} = \mathbf{R}_{\phi_z = \alpha} \mathbf{e}_x \quad (5.265)$$

Die Abbildungen 5.46 und 5.47 zeigen, wie sich potentielle Schraffurlinien einfach und sicher mit dem Abstand der Tropfenbreite B zueinander berechnen lassen. Es handelt sich dabei um diejenigen Linien, die verwendet werden, um eine passende Füllung zu berechnen. Die Definition der Schraffurlinien ist allgemein gültig. Die jeweiligen Fälle sind so gewählt, dass immer gilt:

$$\sin \alpha \geq \frac{\sqrt{2}}{2} \quad (5.266)$$

oder

$$\cos \alpha \geq \frac{\sqrt{2}}{2}. \quad (5.267)$$

Auf diese Weise lässt sich der Versatz der Schraffurlinien zueinander stabil berechnen. Auch ein Abbruchkriterium für die Definition von Schraffurlinien ist in den Abbildungen 5.46 und 5.47 gegeben: es muss ja schließlich nicht im ganzen \mathbb{R}^2 schraffiert werden, sondern es reicht, wenn die definierten Schraffurlinien die Bounding Box um die Grenzkonturen innerhalb der Schicht abdecken.

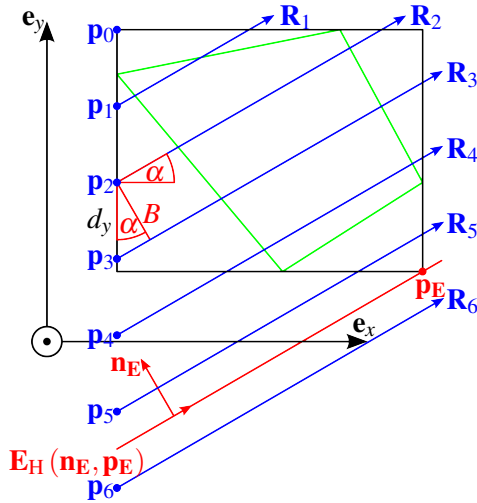
Berechnung von Schraffursegmenten

Für jede nach obigen Vorgehen definierte Schraffurlinie der Form

$$\mathbf{R}_k = \{ \mathbf{p}'_k + \lambda \mathbf{v} \mid 0 \leq \lambda \in \mathbb{R} \} \quad (5.268)$$

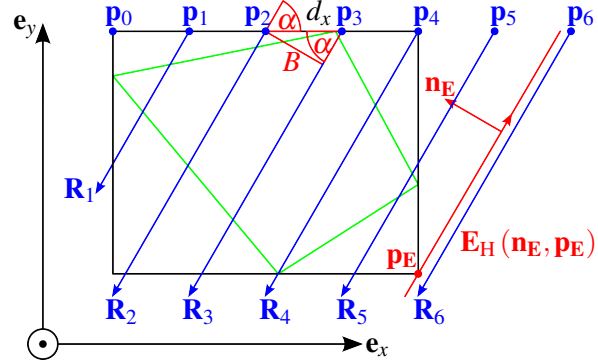
werden nun die Schnittpunkte \mathbf{p}_{s_l} mit allen Segmenten

$$\mathbf{S}(\mathbf{p}_i, \mathbf{p}_{i+1}) \quad (5.269)$$



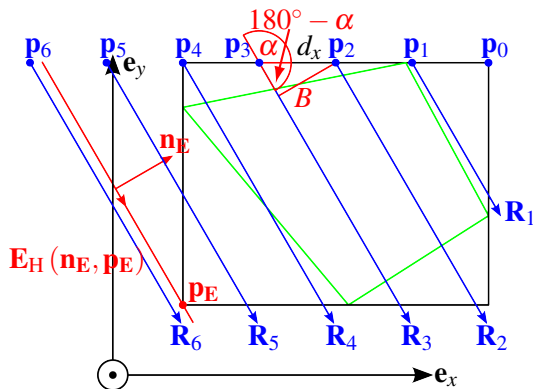
(a) Fall: $0^\circ \leq \alpha < 45^\circ$.

$$\begin{aligned} \mathbf{p}_0 &= (x_{\min}, y_{\max}, z). \\ d_y &= -B / \cos(\alpha\pi/180). \\ \mathbf{p}_{i+1} &= (x_{p_i}, y_{p_i} + d_y, z_{p_i})^T. \\ \mathbf{R}_i &= \{ \mathbf{p}_i + \lambda \mathbf{d} \mid 0 \leq \lambda \in \mathbb{R} \}. \\ \mathbf{n}_E &= \mathbf{e}_z \times \mathbf{d}. \\ \mathbf{p}_E &= (x_{\max}, y_{\min}, z). \end{aligned}$$



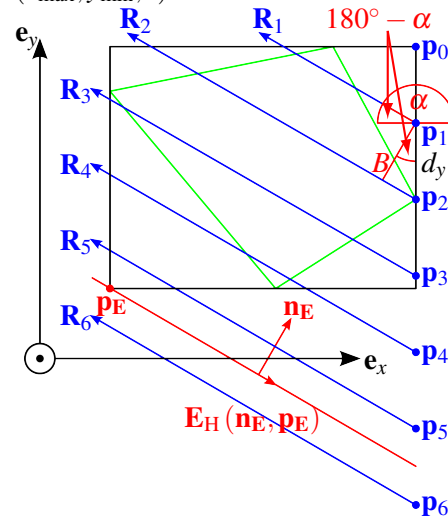
(b) Fall: $45^\circ \leq \alpha < 90^\circ$.

$$\begin{aligned} \mathbf{p}_0 &= (x_{\min}, y_{\max}, z). \\ d_x &= B / \sin(\alpha\pi/180). \\ \mathbf{p}_{i+1} &= (x_{p_i} + d_x, y_{p_i}, z_{p_i})^T. \\ \mathbf{R}_i &= \{ \mathbf{p}_i - \lambda \mathbf{d} \mid 0 \leq \lambda \in \mathbb{R} \}. \\ \mathbf{n}_E &= \mathbf{e}_z \times \mathbf{d}. \\ \mathbf{p}_E &= (x_{\max}, y_{\min}, z). \end{aligned}$$



(c) Fall: $90^\circ \leq \alpha < 135^\circ$.

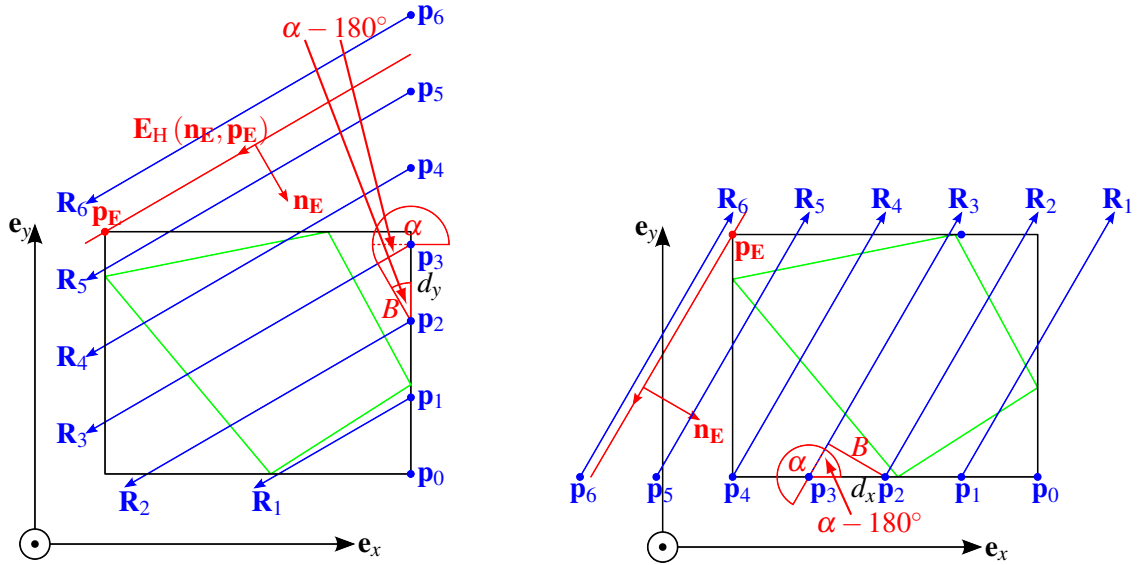
$$\begin{aligned} \mathbf{p}_0 &= (x_{\max}, y_{\max}, z). \\ d_x &= -B / \sin((180 - \alpha)\pi/180). \\ \mathbf{p}_{i+1} &= (x_{p_i} + d_x, y_{p_i}, z_{p_i})^T. \\ \mathbf{R}_i &= \{ \mathbf{p}_i - \lambda \mathbf{d} \mid 0 \leq \lambda \in \mathbb{R} \}. \\ \mathbf{n}_E &= \mathbf{e}_z \times (-\mathbf{d}). \\ \mathbf{p}_E &= (x_{\min}, y_{\min}, z). \end{aligned}$$



(d) Fall: $135^\circ \leq \alpha < 180^\circ$.

$$\begin{aligned} \mathbf{p}_0 &= (x_{\max}, y_{\max}, z). \\ d_y &= -B / \cos((180 - \alpha)\pi/180). \\ \mathbf{p}_{i+1} &= (x_{p_i}, y_{p_i} + d_y, z_{p_i})^T. \\ \mathbf{R}_i &= \{ \mathbf{p}_i + \lambda \mathbf{d} \mid 0 \leq \lambda \in \mathbb{R} \}. \\ \mathbf{n}_E &= \mathbf{e}_z \times (-\mathbf{d}). \\ \mathbf{p}_E &= (x_{\min}, y_{\min}, z). \end{aligned}$$

Abbildung 5.46: Grün eingezeichnet ist die Begrenzungskontur der Füllung, das schwarze Rechteck entspricht der Bounding Box um diese Kontur. Schraffurwinkel: α . Nach Satz 7 gilt: $\mathbf{d} = \mathbf{R}_{\phi_i = \alpha} \mathbf{e}_x$. Falls $\mathbf{n}_E \circ (\mathbf{p}_i - \mathbf{p}_E) < 0$: Abbruch (\mathbf{R}_i schneidet nach Satz 14 die Bounding Box nicht mehr: \mathbf{p}_i liegt unterhalb $\mathbf{E}_H(\mathbf{n}_E, \mathbf{p}_E)$).

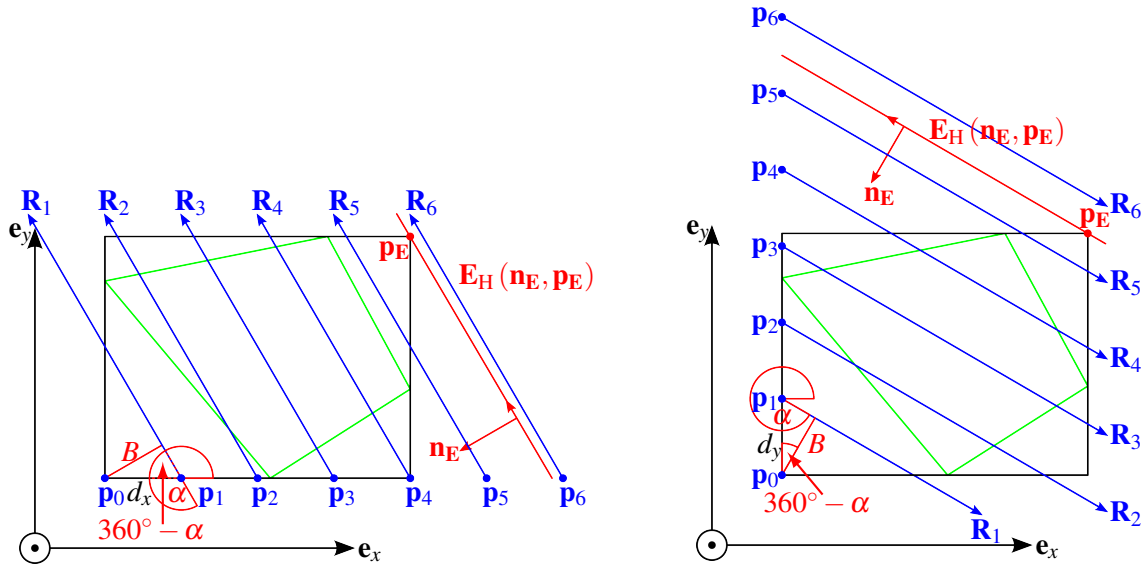


(a) Fall: $180^\circ \leq \alpha < 225^\circ$.

$$\begin{aligned} \mathbf{p}_0 &= (x_{\max}, y_{\min}, z). \\ d_y &= B / \cos((\alpha - 180) \pi / 180). \\ \mathbf{p}_{i+1} &= (x_{\mathbf{p}_i}, y_{\mathbf{p}_i} + d_y, z_{\mathbf{p}_i})^T. \\ \mathbf{R}_i &= \{ \mathbf{p}_i + \lambda \mathbf{d} \mid 0 \leq \lambda \in \mathbb{R} \}. \\ \mathbf{n}_E &= \mathbf{e}_z \times \mathbf{d}. \\ \mathbf{p}_E &= (x_{\min}, y_{\max}, z). \end{aligned}$$

(b) Fall: $225^\circ \leq \alpha < 270^\circ$.

$$\begin{aligned} \mathbf{p}_0 &= (x_{\max}, y_{\min}, z). \\ d_x &= -B / \sin((\alpha - 180) \pi / 180). \\ \mathbf{p}_{i+1} &= (x_{\mathbf{p}_i} + d_x, y_{\mathbf{p}_i}, z_{\mathbf{p}_i})^T. \\ \mathbf{R}_i &= \{ \mathbf{p}_i - \lambda \mathbf{d} \mid 0 \leq \lambda \in \mathbb{R} \}. \\ \mathbf{n}_E &= \mathbf{e}_z \times \mathbf{d}. \\ \mathbf{p}_E &= (x_{\min}, y_{\max}, z). \end{aligned}$$



(c) Fall: $270^\circ \leq \alpha < 315^\circ$.

$$\begin{aligned} \mathbf{p}_0 &= (x_{\min}, y_{\min}, z). \\ d_x &= B / \sin((360 - \alpha) \pi / 180). \\ \mathbf{p}_{i+1} &= (x_{\mathbf{p}_i} + d_x, y_{\mathbf{p}_i}, z_{\mathbf{p}_i})^T. \\ \mathbf{R}_i &= \{ \mathbf{p}_i - \lambda \mathbf{d} \mid 0 \leq \lambda \in \mathbb{R} \}. \\ \mathbf{n}_E &= \mathbf{e}_z \times (-\mathbf{d}). \\ \mathbf{p}_E &= (x_{\max}, y_{\max}, z). \end{aligned}$$

(d) Fall: $315^\circ \leq \alpha < 360^\circ$.

$$\begin{aligned} \mathbf{p}_0 &= (x_{\min}, y_{\min}, z). \\ d_y &= B / \cos((360 - \alpha) \pi / 180). \\ \mathbf{p}_{i+1} &= (x_{\mathbf{p}_i}, y_{\mathbf{p}_i} + d_y, z_{\mathbf{p}_i})^T. \\ \mathbf{R}_i &= \{ \mathbf{p}_i + \lambda \mathbf{d} \mid 0 \leq \lambda \in \mathbb{R} \}. \\ \mathbf{n}_E &= \mathbf{e}_z \times (-\mathbf{d}). \\ \mathbf{p}_E &= (x_{\max}, y_{\max}, z). \end{aligned}$$

Abbildung 5.47: Selbe Voraussetzungen wie in Abbildung 5.46.

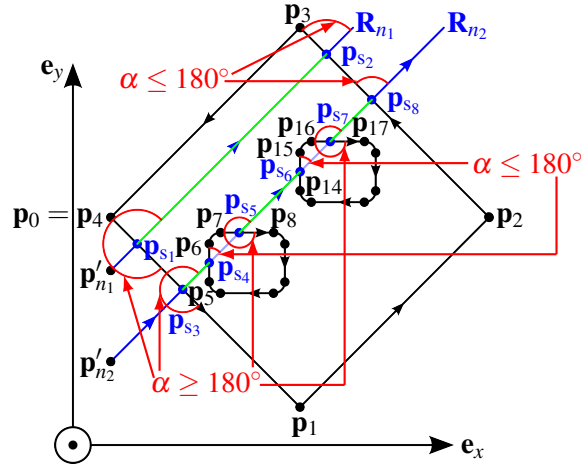


Abbildung 5.48: Eine äußere und zwei innere Konturen, die als Begrenzung für die Füllung dienen. Die Konturen haben dieselbe ID, weil sie zur selben Zusammenhangskomponente gehören. Die Strahlen $\mathbf{R}_k = \{\mathbf{p}'_k + \lambda \mathbf{v} \mid 0 \leq \lambda \in \mathbb{R}\}$ schneiden die Konturen in den Punkten \mathbf{p}_{s_i} . Dadurch wird ein Strahl in einzelne Segmente nach Definition 5 aufgetrennt. Im Beispiel sind dies $\mathbf{S}(\mathbf{p}_{s_1}, \mathbf{p}_{s_2})$, $\mathbf{S}(\mathbf{p}_{s_3}, \mathbf{p}_{s_4})$, $\mathbf{S}(\mathbf{p}_{s_4}, \mathbf{p}_{s_5})$, $\mathbf{S}(\mathbf{p}_{s_5}, \mathbf{p}_{s_6})$, $\mathbf{S}(\mathbf{p}_{s_6}, \mathbf{p}_{s_7})$ und $\mathbf{S}(\mathbf{p}_{s_7}, \mathbf{p}_{s_8})$. Nur die grün markierten Segmente gehören zur Füllung.

und Kreisbögen

$$\mathbf{K}_{\text{Bog}}(\mathbf{p}_j, \mathbf{p}_{j+1}) \subset \partial \mathbf{K}_{d_{\text{off}}, \mathbf{e}_z}(\mathbf{p}'_m) \quad (5.270)$$

aller Grenzkonturen berechnet. Durch die Definition von \mathbf{R}_k ergibt sich für alle Schnittpunkte ein Wert $\lambda_s \geq 0$. Durch die Schnittpunkte werden die Strahlen in einzelne Segmente aufgetrennt. Im Beispiel in Abbildung 5.48 sind dies die Segmente $\mathbf{S}(\mathbf{p}_{s_1}, \mathbf{p}_{s_2})$, $\mathbf{S}(\mathbf{p}_{s_3}, \mathbf{p}_{s_4})$, $\mathbf{S}(\mathbf{p}_{s_4}, \mathbf{p}_{s_5})$, $\mathbf{S}(\mathbf{p}_{s_5}, \mathbf{p}_{s_6})$, $\mathbf{S}(\mathbf{p}_{s_6}, \mathbf{p}_{s_7})$ und $\mathbf{S}(\mathbf{p}_{s_7}, \mathbf{p}_{s_8})$. An dieser Stelle muss nun entschieden werden, welche Schraffursegmente wirklich zur Füllung gehören und welche nicht. Die Grenzkonturen sind, wie alle Offset-Konturen, so orientiert, dass sich links von einer geschlossenen Kurve die Füllung befindet. Seien nun allgemein $\mathbf{S}(\mathbf{p}_{s_i}, \mathbf{p}_{s_{i+1}})$ ein Schraffursegment und $\mathbf{E}(\mathbf{p}_{j_1}, \mathbf{p}_{j_1+1})$ und $\mathbf{E}(\mathbf{p}_{j_2}, \mathbf{p}_{j_2+1})$ diejenigen geometrischen Elemente mit $\mathbf{p}_{s_i} \in \mathbf{E}(\mathbf{p}_{j_1}, \mathbf{p}_{j_1+1})$ und $\mathbf{p}_{s_{i+1}} \in \mathbf{E}(\mathbf{p}_{j_2}, \mathbf{p}_{j_2+1})$. Gilt $\mathbf{E}(\mathbf{p}_{j_1}, \mathbf{p}_{j_1+1}) = \mathbf{S}(\mathbf{p}_{j_1}, \mathbf{p}_{j_1+1})$, dann sei

$$\mathbf{t}_i(\mathbf{p}_{s_i}) = \left\{ \mathbf{p}_{j_1} + \mu_i \underbrace{\frac{\mathbf{p}_{j_1+1} - \mathbf{p}_{j_1}}{|\mathbf{p}_{j_1+1} - \mathbf{p}_{j_1}|}}_{=: \mathbf{d}_i} \mid \mu_i \in \mathbb{R} \right\} \quad (5.271)$$

bzw. gilt $\mathbf{E}(\mathbf{p}_{j_1}, \mathbf{p}_{j_1+1}) = \mathbf{K}_{\text{Bog}}(\mathbf{p}_{j_1}, \mathbf{p}_{j_1+1}) \subset \partial \mathbf{K}_{d_{\text{off}}, \mathbf{e}_z}(\mathbf{p}'_{m_1})$, dann sei

$$\mathbf{t}_i(\mathbf{p}_{s_i}) = \left\{ \mathbf{p}_{s_i} + \mu_i \underbrace{\frac{(\mathbf{p}_{s_i} - \mathbf{p}'_{m_1}) \times \mathbf{e}_z}{|(\mathbf{p}_{s_i} - \mathbf{p}'_{m_1}) \times \mathbf{e}_z|}}_{=: \mathbf{d}_i} \mid \mu_i \in \mathbb{R} \right\}. \quad (5.272)$$

Im Fall von Gleichung (5.272) ist $\mathbf{t}_i(\mathbf{p}_{s_i})$ die Tangente an $\mathbf{K}_{\text{Bog}}(\mathbf{p}_{j_1}, \mathbf{p}_{j_1+1})$ im Punkt \mathbf{p}_{s_i} . Gilt $\mathbf{E}(\mathbf{p}_{j_2}, \mathbf{p}_{j_2+1}) = \mathbf{S}(\mathbf{p}_{j_2}, \mathbf{p}_{j_2+1})$, dann sei

$$\mathbf{t}_{i+1}(\mathbf{p}_{s_{i+1}}) = \left\{ \mathbf{p}_{j_2} + \mu_{i+1} \underbrace{\frac{\mathbf{p}_{j_2+1} - \mathbf{p}_{j_2}}{|\mathbf{p}_{j_2+1} - \mathbf{p}_{j_2}|}}_{=: \mathbf{d}_{i+1}} \mid \mu_{i+1} \in \mathbb{R} \right\} \quad (5.273)$$

bzw. gilt $\mathbf{E}(\mathbf{p}_{j_2}, \mathbf{p}_{j_2+1}) = \mathbf{K}_{\text{Bog}}(\mathbf{p}_{j_2}, \mathbf{p}_{j_2+1}) \subset \partial \mathbf{K}_{d_{\text{off}}, \mathbf{e}_z}(\mathbf{p}'_{m_2})$, dann sei

$$\mathbf{t}_{i+1}(\mathbf{p}_{s_{i+1}}) = \left\{ \mathbf{p}_{s_{i+1}} + \mu_{i+1} \underbrace{\frac{(\mathbf{p}_{s_{i+1}} - \mathbf{p}'_{m_2}) \times \mathbf{e}_z}{|(\mathbf{p}_{s_{i+1}} - \mathbf{p}'_{m_2}) \times \mathbf{e}_z|}}_{=: \mathbf{d}_{i+1}} \mid \mu_{i+1} \in \mathbb{R} \right\}. \quad (5.274)$$

Im Fall von Gleichung (5.274) ist $\mathbf{t}_{i+1}(\mathbf{p}_{s_{i+1}})$ die Tangente an $\mathbf{K}_{\text{Bog}}(\mathbf{p}_{j_2}, \mathbf{p}_{j_2+1})$ im Punkt $\mathbf{p}_{s_{i+1}}$. Seien nun

$$\mathbf{n}_i := (\mathbf{p}_{s_{i+1}} - \mathbf{p}_{s_i}) \times \mathbf{d}_i \quad (5.275)$$

bzw.

$$\mathbf{n}_{i+1} := (\mathbf{p}_{s_{i+1}} - \mathbf{p}_{s_i}) \times \mathbf{d}_{i+1} \quad (5.276)$$

zwei auf der \mathbf{e}_x - \mathbf{e}_y -Fläche senkrechte Vektoren, dann gilt nach Definition 4 des Kreuzprodukts und Abbildung 5.48:

$$\mathbf{S}(\mathbf{p}_{s_i}, \mathbf{p}_{s_{i+1}}) \text{ wird gedruckt} \Leftrightarrow \begin{array}{l} \mathbf{n}_i \text{ zeigt in entgegengesetzte Richtung von } \mathbf{e}_z \text{ und} \\ \mathbf{n}_{i+1} \text{ zeigt in dieselbe Richtung wie } \mathbf{e}_z \end{array} \quad (5.277)$$

Falls $\mathbf{e}_z = (0, 0, 1)^T$ gilt, dann heißt dies:

$$\mathbf{S}(\mathbf{p}_{s_i}, \mathbf{p}_{s_{i+1}}) \text{ wird gedruckt} \Leftrightarrow z_{\mathbf{n}_i} \leq 0 \wedge z_{\mathbf{n}_{i+1}} \geq 0. \quad (5.278)$$

Dies ist also das ausschlaggebende Kriterium, ob ein Schraffursegment gedruckt wird oder nicht.

Reduzierung von Leerfahrten

Um Leerfahrten weitestgehend zu reduzieren, müssen Reihenfolge und Druckrichtung der berechneten Schraffursegmente optimiert werden. Hierzu wird nach dem Druckende eines Schraffursegments immer nach dem nächstgelegenen Start- bzw. Endpunkt eines noch nicht gedruckten Schraffursegments gesucht. Falls es sich dabei um den Endpunkt eines berechneten Schraffursegments handelt, dann muss die Richtung des Schraffursegments für den Druck geändert werden. Mit Hilfe dieser Nearest-Neighbour-Heuristik wird versucht, Leerfahrten weitestgehend zu reduzieren. Die Abbildungen 5.49 und 5.50 zeigen für zwei verschiedene Schraffurwinkel den Bauprozess bestehend aus möglichst kleinen Leerfahrtwegen und den Schraffursegmenten mit Materialaustrag.

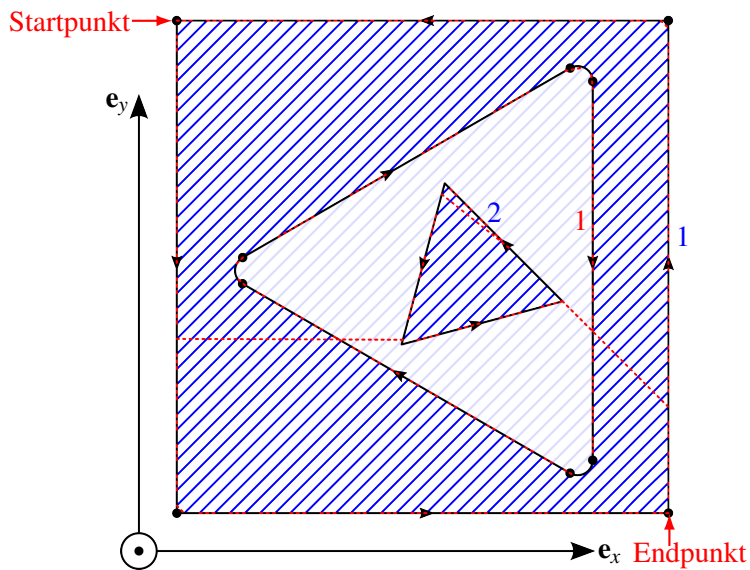


Abbildung 5.49: Drei Konturen die als Begrenzung für die Füllung dienen. Zunächst wird jeder Kontur nach dem Prinzip aus Abbildung 5.25 in Kapitel 5.2.4 eine ID zugewiesen. Gefüllt wird nur innerhalb äußerer und außerhalb innerer Konturen mit derselben ID. Das Beispiel zeigt die Verfahrensweise beim Füllen eines Bauteils: rot gestrichelte Linien sind Leerfahrten, bei blauen Linien gibt es Materialaustage. Nach der Nearest-Neighbour-Heuristik wird versucht, Verfahrenwege möglichst gering zu halten: der Abstand zwischen Endpunkt einer gedruckten Fülllinie und Startpunkt einer neuen noch nicht gedruckten Fülllinie wird minimal gehalten. Das Beispiel zeigt eine 45°-Schraffur.

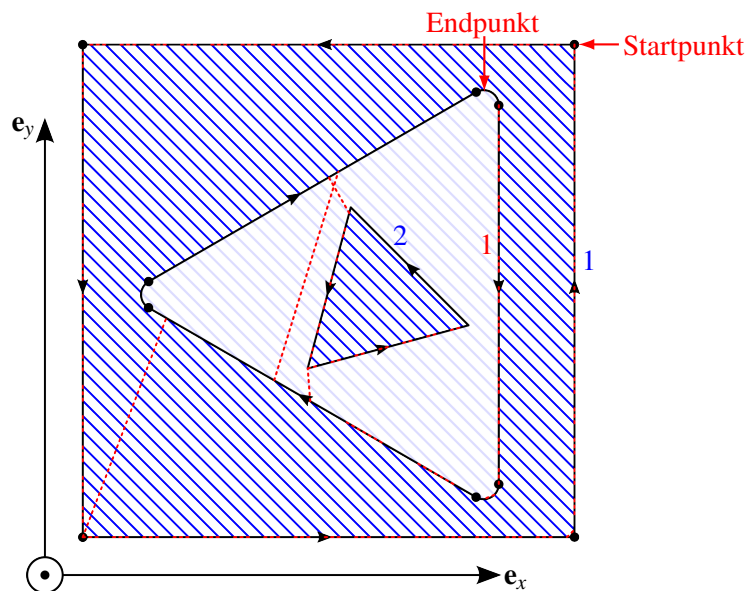


Abbildung 5.50: Selbes Prinzip wie in Abbildung 5.50, jedoch hier mit einer 135°-Schraffur.

5.3.5 Optimierung der Daten für den Tropfenerzeuger

Die Datenoptimierungen betreffen die in Kapitel 5.3.3 berechneten Offset-Konturen. Berücksichtigt werden dabei lediglich Konturen, welche auch gedruckt werden. Das heißt insbesondere

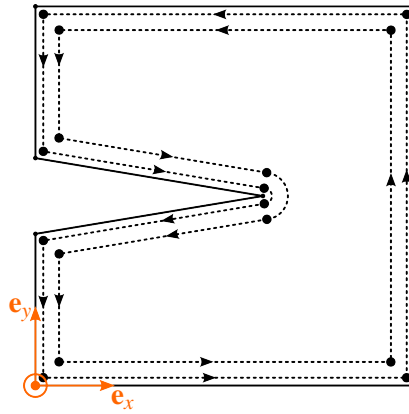


Abbildung 5.51: Beispiel: Würfel mit Keil. Die gestrichelten Linien zeigen die beiden Offset-Kontur-Niveaus mit $d_{\text{off}} = \frac{B}{2}$ bzw. $d_{\text{off}} = \frac{3B}{2}$. Auf Basis dieser Offset-Konturen, bestehend aus Segmenten und Kreisbögen, sollen die Daten für den Druck mit dem Tropfenerzeuger optimiert werden.

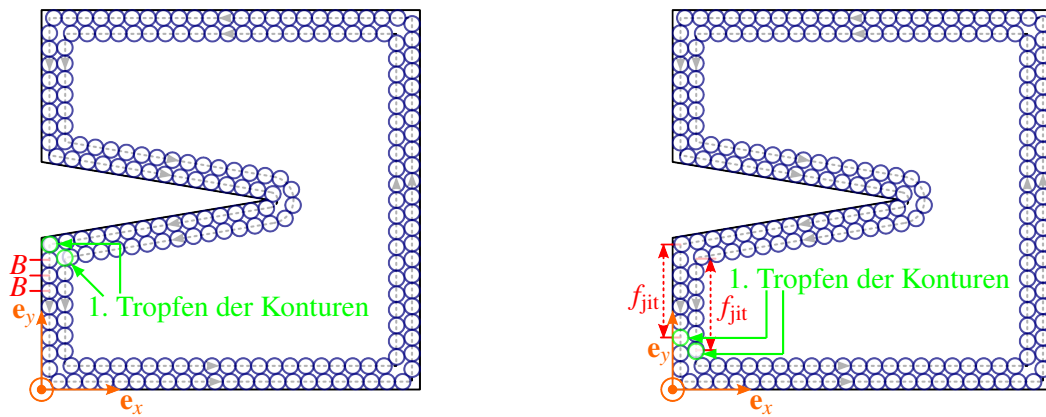
re, dass die Begrenzungskonturen für die Füllung nicht auf diese Weise optimiert werden, da diese nicht gedruckt werden. Auf diesen unnötigen Rechenaufwand wird verzichtet. Abbildung 5.51 zeigt nun ein Beispiel für berechnete Offset-Konturen. Speziell hieran soll die Optimierung der Daten erklärt werden.

Da es sich um einen tropfenbasierten Ansatz handelt, ist die Idee, auf den berechneten Offset-Konturen nacheinander Tropfen im Abstand der Tropfenbreite B zu setzen. Dies ist anschaulich in Abbildung 5.52 illustriert. Ausgehend vom Schwerpunkt \mathbf{p}_0 des ersten Tropfens einer Kontur, werden nacheinander die Schwerpunkte \mathbf{p}_i der folgenden Tropfen berechnet. \mathbf{p}_i ergibt sich dabei durch den nächsten Schnittpunkt des Kreisrandes $\partial\mathbf{K}_{B,e_z}(\mathbf{p}_{i-1})$ mit den geometrischen Elementen der Offset-Kontur, die gerade gedruckt werden soll. Dabei wird der Schnittpunkt genommen, der nicht dem Schwerpunkt \mathbf{p}_{i-2} des vorangegangenen Tropfens entspricht, denn sonst würde die Druckrichtung geändert werden und der Tropfen doppelt gedruckt werden. Sämtliche Schnittpunktberechnungen erfolgen dabei wie in Kapitel 5.2.3 dargestellt.

Wie sich einfach vorstellen lässt, kann es auf diese Weise zu einer Überlappung des ersten und des letzten gedruckten Tropfens einer Kontur geben, wenn man von einem Tropfenmodell mit gleich großen Tropfen ausgeht. Dieser Materialüberfluss kann sich auf die Oberflächenqualität der Bauteile auswirken, wenn Schicht für Schicht der erste Tropfen einer Kontur an derselben \mathbf{e}_x - \mathbf{e}_y -Koordinate gesetzt wird. Um diesen vermeintlichen Fehler über das ganze Bauteil zu verteilen, versetzt man den Startpunkt einer zu druckenden Kontur im Vergleich zur vorangegangenen Schicht immer um einen Wert f_{jit} (siehe Abbildung 5.52b).

5.3.6 Generierung von prozessspezifischem Code

Während die in Kapitel 5.3.5 optimierten Konturdaten als Grundlage für den prozessspezifischen Berandungsdruck dienen, dienen die in Kapitel 5.3.4 berechneten Schraffursegmente die Grundlage für den Druck der Füllung. Die berechneten Daten werden nun nur noch in eine Datei mit der Dateierweiterung „.gco“ geschrieben. Für die genaue Vorgehensweise und Syntax sei an dieser Stelle nur noch einmal auf Kapitel 4.5 und Abbildung 4.8 verwiesen. Es sei nur noch einmal darauf hingewiesen, dass die Tropfen beim Druck eines Schraffursegments automatisch im richtigen Abstand der Tropfenbreite B positioniert werden, ohne die einzelnen Tropfen vorgeben zu müssen.



- (a) Geschlossene Konturen werden tropfenweise erzeugt. Der Abstand zwischen benachbarten Tropfen einer Kontur ist die Tropfenbreite B . Nur zwischen dem ersten und letzten gedruckten Tropfen einer Kontur kann es eine Überlappung geben.
- (b) Geschlossene Konturen aus Tropfen für die auf Abbildung 5.52a folgende Schicht. Im Vergleich zur vorherigen Schicht wird der erste Tropfen einer Kontur um einen Wert f_{jit} versetzt. Somit wird eine mögliche Überlappung des ersten und letzten Tropfens einer Schicht allgemein auf das ganze Bauteil übertragen. Es entsteht kein unschönes Muster in e_z -Richtung.

Abbildung 5.52: Optimierung der Daten aus Abbildung 5.51 für den Tropfenerzeuger.

5.4 Implementierung der Software-Module

In diesem Kapitel soll auf die Implementierung der Software-Module und deren Integration in den Gesamtprozess eingegangen werden. Die Implementierung der in Kapitel 5.3 beschriebenen Software-Module erfolgte mit der Programmiersprache C++ unter Windows 7 Professional 64 Bit inklusive Service Pack 1 mit 8 GB RAM und einer Intel® Core™ i7-2760QM CPU bestehend aus vier Prozessorkernen mit einer Taktfrequenz von je 2.4 GHz. Als Entwicklungsumgebung kam dabei Microsoft Visual Studio 2010 Ultimate zum Einsatz. Des Weiteren wurden zur Implementierung auch die Boost C++ Libraries verwendet. Diese beinhalten unter anderem auch eine Implementierung von sogenannten Smart Pointern. Pointer zeigen im Allgemeinen auf den Beginn eines allokierten Speicherbereichs. Smart Pointer besitzen in der Regel hingegen auch noch einen Referenzzähler, der angibt, wie viele Pointer es auf einen bestimmten Speicherbereich gibt. Erst wenn der Referenzzähler auf 0 steht, dann wird der allokierte Speicherbereich erst wieder freigegeben. Dies geschieht allerdings im Vergleich zu normalen Pointern automatisch und ist einer der Vorteile bei der Nutzung von Smart Pointern. Des Weiteren wurden die Boost C++ Libraries verwendet um die gesamte Anwendung im Hinblick auf Mehrprozessorsysteme zu optimieren. Hierzu wird in der Regel die Abarbeitung mehrerer kleinerer Prozesse (Threads) parallel auf die verschiedenen Prozessorkerne verlagert und somit ein deutlicher Geschwindigkeitsvorteil erzielt im Vergleich zur Abarbeitung in einem einzigen Hauptprozess.

5.4.1 Eingliederung in den Gesamtprozess

Ziel dieser Arbeit war eine für den gesamten Druckprozess spezifische Code-Generierung. Kein Teil der Arbeit war eine Visualisierung der Datenaufbereitung. Für diesen Zweck wurde vom

Industriepartner eine grafische Benutzeroberfläche (Slicer-GUI) zur Verfügung gestellt. Bevor die Code-Generierung gestartet wird, kann der Benutzer hier entscheiden, für welche STL-Datei Code generiert werden soll. Die STL-Datei wird dann von der Festplatte geladen und in der GUI entsprechend angezeigt. Gewöhnlich lädt der Nutzer dann ein Baumaterial aus. Die hierfür erforderlichen Parameter werden aus einer xml-Datei geladen, so dass sich der Benutzer keine Gedanken mehr über das Druck-Setup machen muss. Nun kann der Benutzer entscheiden, wie er das Bauteil gerne im Bauraum platzieren würde und es entsprechend rotieren, verschieben, aber auch skalieren. Danach kann über einen Button die Code-Generierung gestartet werden. Der Benutzer wählt noch schnell einen Speicherort aus, danach erledigt die im Rahmen dieser Arbeit entwickelte Software die Code-Generierung.

Einbindung über eine DLL

Die entwickelte Software wird in Form einer DLL (**D**ynamic **L**ink **L**ibrary) zur Laufzeit in die Slicer-GUI eingebunden. Hierzu existieren in der DLL Schnittstellen-Funktionen, die aus der GUI heraus aufgerufen werden können. Neben einer Hauptfunktion **processAll**, die für die Code-Generierung verantwortlich ist, ist dies im Wesentlichen noch eine Funktion **getProgress-
State**, die den Fortschritt bei der Code-Generierung angibt, um dies entsprechend auch in der GUI dem Nutzer als Rückmeldung anzuzeigen. Beim Aufruf der Funktion **processAll** werden sämtliche druck- bzw. materialspezifischen Parameter, die in der xml-Datei stehen, übergeben, aber auch alle Parameter, die Aufschluss darüber geben, wie der Nutzer das Bauteil im Bauraum platziert hat. Im Folgenden wird ein Überblick über die wichtigsten Übergabeparameter an die DLL dargestellt. Dabei gibt es einen Parameter:

- der Speicherort und -name der STL-Datei spezifiziert,
- der Speicherort und -name des zu generierenden G-Codes definiert,
- für die Rotation um die e_x -Achse,
- für die Rotation um die e_y -Achse,
- für die Rotation um die e_z -Achse,
- für die Skalierung in e_x -Richtung,
- für die Skalierung in e_y -Richtung,
- für die Skalierung in e_z -Richtung,
- für die Schichthöhe,
- für das Verhältnis Breite / Höhe eines Tropfens,
- für den Füllgrad der Füllung,
- für die Anzahl der zu druckenden Konturen pro Rand,
- der angibt, um wie viele *mm* der Startpunkt einer zu druckenden Kontur im Vergleich zur vorangegangenen Schicht verschoben werden muss (Jitter),
- für den Überlappungsgrad der Füllung mit der Berandung,

- der angibt, in welcher Reihenfolge die Konturen gedruckt werden (von innen nach außen oder umgekehrt),
- der angibt, ob erst die Konturen und dann die Füllung gedruckt wird, oder umgekehrt,
- für den Schraffurwinkel für die Füllung der ersten zu druckenden Schicht,
- für den Versatzwinkel der Schraffurlinien der Füllung im Vergleich zur vorangegangenen Schicht,
- für einen Faktor, um ein wie vielfaches der Tropfenbreite äußere gegen den Uhrzeigersinn laufende Konturen verschoben werden sollen,
- für einen Faktor, um ein wie vielfaches der Tropfenbreite innere im Uhrzeigersinn laufende Konturen verschoben werden sollen.

Die ersten acht angegebenen Parameter lassen sich durch die Nutzerinteraktion mit der GUI beeinflussen, alle anderen Parameter sind druck- bzw. materialspezifische Parameter aus einer xml-Datei für ein bestimmtes Material. Während der Entwicklungsphase ist es aber natürlich sinnvoll auch mit diesen Parametern experimentieren zu können um ein optimales Druck-Setup zu finden. Wurde der G-Code generiert, so wird er in der GUI entsprechend visualisiert. Der Nutzer ist dabei in der Lage durch die einzelnen Schichten zu navigieren und das Ergebnis entsprechend zu betrachten bzw. zu prüfen. Der generierte Code kann danach auf die Maschine übertragen werden und entsprechend geladen werden. Der Code enthält alle nötigen Fahrkommandos für die Maschinensteuerung und ermöglicht somit den Druck des Bauteils.

5.4.2 Programmierung der Module

Für jedes Modul aus Kapitel 5.3 existiert eine eigene implementierte Klasse: ein **STL Importer** zum Importieren der STL-Daten, ein **STL Slicer** für das Slicen, ein **Offset Contour** zur Berechnung von Konturen mit einem bestimmten Offset f_o für äußere Konturen (**outer contours**) und f_i für innere Konturen (**inner contours**), ein **Hatching Generator** zur Generierung der Füllung in Form von Schraffursegmenten, ein **Post Processor** zur prozessspezifischen Datenoptimierung und ein **G-Code Generator** zum Schreiben der berechneten Daten in eine Datei. Der G-Code Generator nutzt dabei eine Klasse **G-Code Exporter**, in der sämtliche möglichen G-Code Befehle definiert sind, die in die G-Code Datei geschrieben werden können. Der G-Code Generator wählt also anhand der vorliegenden Daten immer den richtigen Befehl im G-Code Exporter aus, der exportiert werden muss. Eine Übersicht über die wichtigsten implementierten Klassen und deren Interaktion findet sich in Abbildung 5.53 wieder. Eine ausführlichere Beschreibung der bis hierhin noch nicht erwähnten Klassen / Module befindet sich im Anschluss.

Dabei dienen die in Abbildung 5.54 gezeigten Datenstrukturen als Eingabe- bzw. Ausgabedatenstrukturen. Prinzipiell existieren dabei drei verschiedene Arten: eine Datenstruktur **Indexed Triangle Mesh** für das Dreiecksgitter aus der STL-Datei, eine Datenstruktur **CLI Data Structure** nur für Polygonzüge und eine Datenstruktur **Indexed Geometric Elements** für geschlossene Konturen bestehend aus Segmenten (**Contour Segment**) und Kreisbögen (**Arch**). Die Datenstruktur für Polygonzüge ist, wie der Name bereits vermuten lässt, stark angelehnt an das CLI-Format (Common Layer Interface), welches sich im Rapid Prototyping Bereich für die Speicherung von Schichtdaten in Form von Segmenten, bewährt hat. Die Datenstruktur für Segmente und Kreisbögen ist eine Erweiterung davon und wird nur also Zwischendatenstruktur verwendet, da die fertigen Schichtdaten wiederum nur aus Segmenten bzw. einer Aneinanderreihung von Tropfen im Abstand der Tropfenbreite zueinander bestehen.

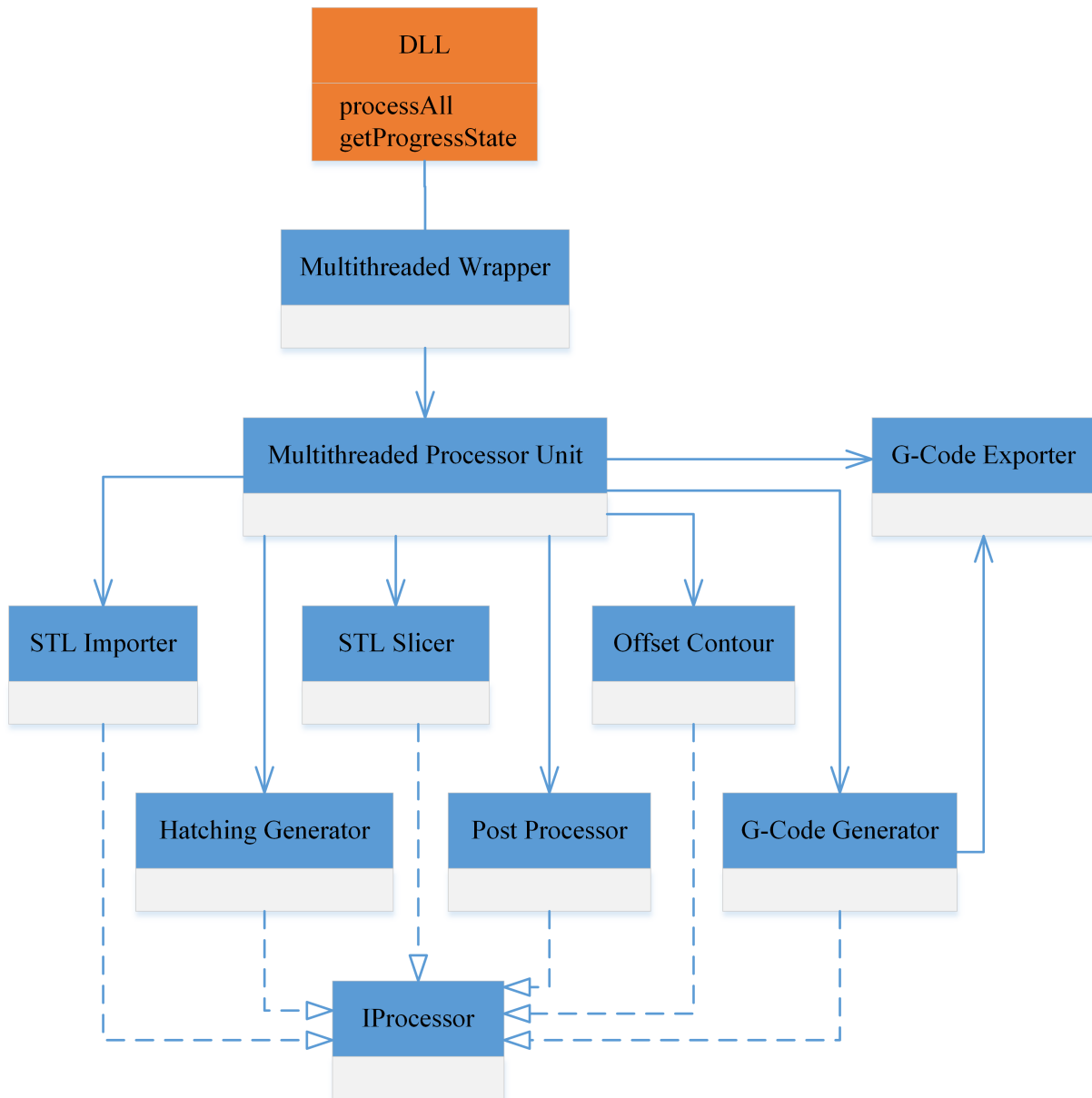


Abbildung 5.53: Überblick über die wichtigsten Software-Module. Die Multithreaded Processor Unit als Herzstück der Implementierung ruft die Berechnungs-Module STL Importer, STL Slicer, Offset Contour, Hatching Generator, Post Processor und G-Code Generator auf. Die Berechnungsmodule haben alle eine gemeinsame Schnittstelle IProcessor. Über den G-Code Exporter wird hier auch eine Datei zum Schreiben des G-Codes angelegt. Der G-Code Generator verarbeitet die berechneten Daten und schreibt sie über den G-Code Exporter in die Datei. Im Modul DLL (keine Klasse) sind Schnittstellenfunktionen der zu generierenden DLL definiert. Über den Multithreaded Wrapper funktioniert die Anbindung an die C++-Welt.

Indexed Triangle Mesh

Die Datenstruktur besteht, wie in Kapitel 4.3.1 beschrieben, aus einer Vertex List **VL** und einer Facet List **FL**. Die Werte in einer Zeile von **FL** sind dabei die Indizes der Punkte in **VL**, die ein Dreieck bilden. In dieser Form werden sowohl STL-Dateien im ASCII- als auch im Binär-

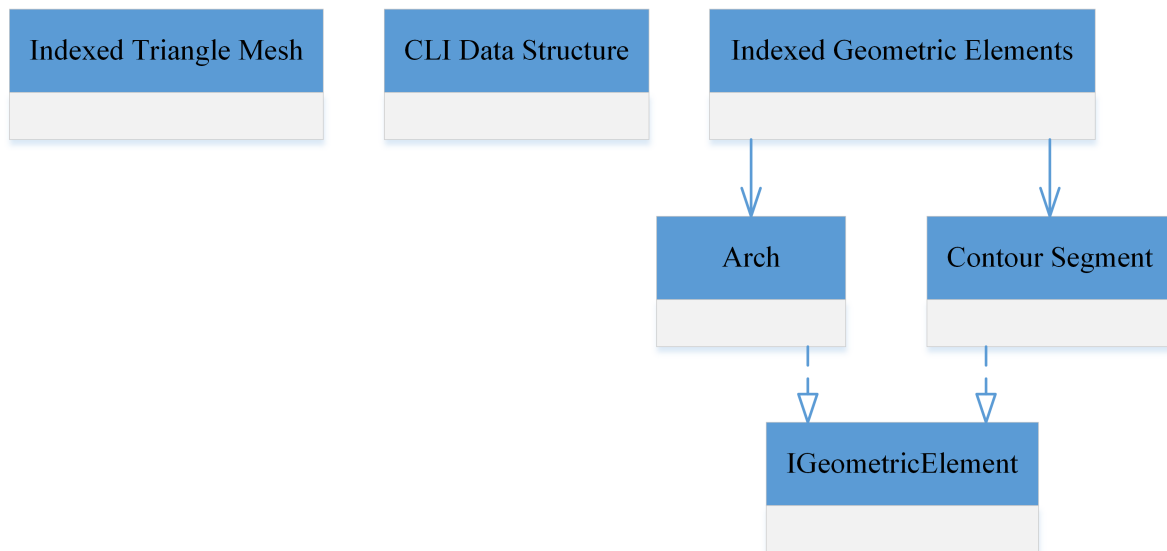


Abbildung 5.54: Verwendete Datenstrukturen. Das Indexed Triangle Mesh ist zum Speichern der Daten des eingelesenen STLs. Die CLI Data Structure ist eine Datenstruktur zur Speicherung von Polygonzügen bzw. Punkten. Die Indexed Geometric Elements sind geschlossene Konturen, bestehend aus Segmenten (Contour Segment) und Kreisbögen (Arch). Contour Segment und Arch haben als geometrische Elemente die gemeinsame Schnittstelle IGeometricElement.

Format eingelesen. Der Aufbau des ASCII-Formats wurde bereits näher erläutert, der Aufbau im Binär-Format ist wie folgt:

1. Header (80 Byte)
2. Anzahl der Dreiecke (4 Byte)
3. Für jedes Dreieck:
 - Normalenvektor (3 · 4 Byte)
 - Eckpunkt 1 (3 · 4 Byte)
 - Eckpunkt 2 (3 · 4 Byte)
 - Eckpunkt 3 (3 · 4 Byte)
 - vorzeichenloser Ganzzahlwert ohne Bedeutung (2 Byte)

Die 4 Byte beim Normalenvektor und bei den Eckpunkten stehen jeweils für einen Fließkommawert. Jeder Vektor bzw. Punkt besteht aus 3 solchen Fließkommawerten (\mathbb{R}^3). Insgesamt hat also jedes Dreieck eine Größe von 50 Byte.

CLI Data Structure

Dies ist die Datenstruktur für die Speicherung von Polygonzügen. Handelt es sich um einen geschlossenen Polygonzug, so spricht man auch von einer geschlossenen Kontur. Bei einem nicht geschlossenen Polygonzug handelt es sich in der Regel um die Füllung. Allgemein besteht ein Polygonzug aus einer Aneinanderreihung von Punkten im \mathbb{R}^3 : $\mathbf{p}_0, \mathbf{p}_1, \dots, \mathbf{p}_n$. Durch diese Punkte sind Segmente $\mathbf{S}(\mathbf{p}_i, \mathbf{p}_{i+1})$, $i = 0, \dots, n - 1$, definiert. Alternativ kann es sich aber

auch um eine Reihe von zu druckenden Tropfen handeln, falls die einzelnen Punkte im Abstand der Tropfenbreite B gesetzt sind, wie dies genau bei den in dieser Arbeit berechneten prozessspezifischen Konturen der Fall ist. Unabhängig von der jeweiligen Interpretation, eignet sich die CLI Data Structure also als Speicherstruktur für die Planung von Bahnen. Sie beinhaltet also, welche Segmente oder Punkte letztendlich in welcher Reihenfolge gedruckt werden. Durch das Abarbeiten einer solchen CLI Data Structure wird durch den G-Code Generator auch der entsprechende G-Code generiert.

Die finale CLI Data Structure beinhaltet also letztlich die Rohdaten sämtlicher Druckkommandos. Allerdings wird eine CLI Data Structure nicht nur als Datenstruktur für die Ausgabedaten verwendet, sondern auch als Ausgangsdatenstruktur, worin die einzelnen geschlossenen Begrenzungskonturen als Ergebnis des Moduls STL Slicer gespeichert sind. Um diese Daten weiterverarbeiten zu können, ist für jede Kontur die Speicherung weiterer Informationen erforderlich, wie bereits aus Kapitel 5.3 hervorgeht. Diese Informationen umfassen:

- die Identifikationsnummer **id** der Zusammenhangskomponente zu der die jeweilige Kontur gehört,
- die Fläche **area** der von der jeweiligen Kontur eingeschlossenen Fläche,
- die Richtung **dir** einer Kontur als Folge des Vorzeichens des entsprechenden Flächeninhalts,
- die Anzahl **n** der Punkte, aus denen die Kontur besteht.

Dabei gilt $dir = 0$ genau dann, wenn eine geschlossene Kontur im Uhrzeigersinn läuft und $dir = 1$, genau dann, wenn eine geschlossene Kontur gegen den Uhrzeigersinn läuft.

Da Polygonzüge aber nicht zwangsweise geschlossen sein müssen, lässt sich dies durch einen anderen Wert ungleich 0 oder 1 kennzeichnen. Den in der Regel nicht geschlossenen Fülllinien wird in dieser Arbeit speziell ein Wert $dir = 3$ zugewiesen, um diese schnell und einfach von den geschlossenen Konturen unterscheiden zu können.

Indexed Geometric Elements

Dies ist die Datenstruktur für die Speicherung geschlossener Konturen, bestehend aus Segmenten (Contour Segment) und Kreisbögen (Arch), wie bereits in Abbildung 5.54 zu sehen war. Sowohl ein Contour Segment, als auch ein Arch, haben die gemeinsame Schnittstelle **IGeometricElement**, da es sich jeweils um ein spezielles geometrisches Element nach Definition 10 handelt. Bei allen in dieser Arbeit konstruierten oder berechneten geschlossenen Konturen handelt es sich nach Definition 14 um Kurven, die aus einer Aneinanderreihung von Segmenten und Kreisbögen bestehen. Die Datenstruktur ist dabei sehr ähnlich zur CLI Data Structure aufgebaut. Für eine einfache Handhabung werden auch für solche Konturen zusätzliche Informationen gespeichert bzw. berechnet. Diese Informationen sind einerseits:

- dieselben Informationen wie bereits bei der CLI Data Structure (siehe oben),

andererseits werden sie aber auch noch durch folgende Informationen ergänzt:

- die Identifikationsnummer **idOrig** der ursprünglichen Zusammenhangskomponente, aus der die Kontur entstanden ist,
- einer Reihe weiterer Informationen, die speziell die geometrischen Elemente der Kontur betrifft. Dabei handelt es sich um:

- die Verkettung durch einen Smart Pointer **pNextElement** auf den Nachfolger und einen Smart Pointer **pLastElement** auf den Vorgänger eines geometrischen Elements,
- eine Liste **Intersection List**, bestehend aus allen Schnittelementen **Intersection Elements**, für jedes geometrische Element. Ein Intersection Element besteht dabei aus folgenden wichtigen Informationen:
 - * dem Schnittpunkt selbst,
 - * einem Smart Pointer **pIntersectionElement** auf das geometrische Element mit demselben Schnittpunkt,
 - * einem Parameter zwischen 0.0 und 1.0, der die relative Lage des Schnittpunkts auf dem geometrischen Element angibt, auf das pIntersectionElement zeigt (dadurch kann speziell dieses Schnittelement wieder schnell in der entsprechenden Intersection List gefunden werden).

Dies sind die wichtigsten Informationen der Indexed Geometric Elements. Man kann sehen, dass die Daten sehr eng miteinander verkoppelt sind.

Einsatz der Datenstrukturen

Im Hinblick auf das Flussdiagramm in Abbildung 4.7 in Kapitel 4.4 zeigt Tabelle 5.1, wo genau die einzelnen erläuterten Datenstrukturen bei den Software-Modulen aus Kapitel 5.3 zum Einsatz kommen.

	Eingabedaten	Ausgabedaten
STL Importer	STL-Datei von Festplatte	Indexed Triangle Mesh
STL Slicer	Indexed Triangle Mesh	CLI Data Structure
Offset Contour	CLI Data Structure	(Indexed Geometric Elements)* & Indexed Geometric Elements
Hatching Generator	Indexed Geometric Elements	CLI Data Structure
Post Processor	(Indexed Geometric Elements)* & CLI Data Structure	CLI Data Structure
G-Code Generator	CLI Data Structure	G-Code auf Festplatte

Tabelle 5.1: Einsatz der Datenstrukturen bei den jeweiligen Software-Modulen. Selbe Farben kennzeichnen dabei, dass es sich um dieselben Datenstrukturen handelt. Die mit einem * gekennzeichneten Datenstrukturen existieren für jedes Offset-Kontur-Niveau, das heißt eine Struktur Indexed Geometric Elements umfasst alle Konturen mit demselben Offset. Es können auch mehrere Offset-Konturen gedruckt werden. Die CLI Data Structure als Eingabestruktur umfasst alle Fülllinien, die CLI Data Structure als Ausgabestruktur beinhaltet zusätzlich auch noch alle tropfenweise zerlegten Konturen. Diese CLI Data Structure umfasst alle Rohdaten zur Generierung von G-Code.

Im folgenden wird nun die Funktionalität der drei Module **Multithreaded Processor Unit**, **DLL** und **Multithreaded Wrapper** aus Abbildung 5.54 beschrieben, welche noch nicht in Kapitel 5.3 erläutert wurden.

Multithreaded Processor Unit

Die Multithreaded Processor Unit (MPU) ist sozusagen das Herzstück der Implementierung und regelt den kompletten Programmablauf. In der MPU sind alle Software-Module aus Kapitel 5.3 integriert. Die MPU regelt sowohl die Interaktion der einzelnen Module als auch den allgemeinen Programmablauf, insbesondere die für eine bestimmte CPU maßgeschneiderte Synchronisation der Threads.

Zunächst werden aber erst sämtliche Übergabeparameter an die DLL für die MPU initialisiert. Nach der Initialisierung werden diese Parameter sofort als Kommentar an den Anfang einer leeren G-Code Datei geschrieben. Dieser sogenannte Header wird also bereits vor Ablauf der eigentlichen Volumendatenaufbereitung generiert. Sollte also der G-Code nicht die gewünschte Form haben oder nicht generiert werden können, so ist anhand des Headers jederzeit problemlos das komplette Setup der Code-Generierung rekonstruierbar.

Nach der erfolgreichen Initialisierung der Parameter wird zunächst die STL-Datei importiert. Ergebnis ist, wie bereits näher erläutert, ein Dreiecksgitter der Oberfläche bestehend aus einer Vertex List **VL** und einer Facet List **FL**. Erst zu diesem Zeitpunkt befinden alle Daten zur Aufbereitung im Speicher. Das Verfahren arbeitet zwar auf Basis des Tropfenmodells, geht dabei aber auch schichtweise vor. Es gibt global gesehen, zwischen benachbarten Schichten kein Gedächtnis. Somit können die Daten sehr einfach schichtweise und parallel verarbeitet werden. Beginnend mit dem Importieren der STL-Daten empfiehlt sich also folgendes Vorgehen:

1. Importieren der STL-Daten.
2. Bestimmung der Anzahl m der virtuellen Prozessorkerne (beim i7-Prozessor sind das: $m = 8$). Anzahl Threads sind dann: $n = \max(1, m - 1)$. Somit soll eine komplette Auslastung des Systems verhindert werden.
3. Anlegen von n Threads, welche die Daten parallel verarbeiten. Ein Thread verarbeitet dabei genau eine Schicht, d.h. es können n Schichten parallel verarbeitet werden. Die Verarbeitung eines Threads umfasst dabei folgende Module:
 - a) Slicen einer Schicht.
 - b) Berechnung aller für den Prozess relevanten Konturen einer Schicht.
 - c) Füllen einer Schicht.
 - d) Datenoptimierung einer Schicht (punktweise Zerlegung, Jitter, ...).
4. Schreiben von G-Code durch Hauptprozess nach Abarbeitung der n Threads. Die n Schichten werden mit aufsteigendem z -Wert der Schichten in die Datei geschrieben.
5. Freigabe des Speichers.
6. Wiederholung der Schritte 3. – 5. bis es nichts mehr zu tun gibt.

Es ist genau zu erkennen, wie für jeden Thread die Module aus Kapitel 5.3 aufgerufen werden. Vorteil dieser Parallelisierung ist ein nicht zu verachtender Performance-Gewinn. Darüber hinaus ist die Freigabe des Speichers ein enorm wichtiger Punkt: bei großen STL-Dateien, werden für den G-Code vor allem auch aufgrund der tropfenweisen Zerlegung der Konturen, immense Datenmengen generiert, wenn alle Daten bis zum Ende im Speicher behalten werden würden. Bei einer Freigabe des Speicher nach der Abarbeitung der n Threads bzw. Schichten bleibt der belegte Speicher hingegen auch bei großen Bauteilen in der Regel unter 100MB.

DLL

Bei dem Modul DLL handelt es sich nicht um eine Klasse. Hier werden lediglich alle Schnittstellen inklusive der Übergabeparameter definiert, welche letztlich in die DLL exportiert werden und in der Slicer-GUI so wieder ansprechbar sind. Dies ist entscheidend um die entwickelte Software zur Code-Generierung auch in der GUI nutzen zu können. Die wichtigsten Schnittstellenfunktionen in der DLL sind, wie bereits vorhin erwähnt, **processAll** und **getProgressState**.

Multithreaded Wrapper

Von der Klasse Multithreaded Wrapper wird beim Importieren der DLL in der Slicer-GUI eine Instanz (Objekt) erzeugt. Dieses ist so lange gültig, so lange die Slicer-GUI auch läuft und ermöglicht die Ausführung von objektorientiertem C++ Code. Für jedes Bauteil, für das während der Laufzeit der GUI G-Code generiert werden soll, wird innerhalb der Klasse Multithreaded Wrapper genau ein Objekt der Klasse Multithreaded Processor Unit (MPU) angelegt. Durch dieses Objekt wird schließlich das eigentliche Programm zur Code-Generierung ausgeführt. Nach der Code-Generierung wird das MPU Objekt wieder gelöscht. Da ja nur genau ein MPU Objekt existiert hatte, werden auch sämtliche Daten wieder gelöscht, die an dieses MPU Objekt oder ein in der MPU allokiertes Objekt gekoppelt waren. Dies gilt insbesondere auch für Speicherbereiche, auf die Smart Pointer verweisen. Der komplette Speicher wird wieder freigegeben. Letztlich kann also ein neues Bauteil in die GUI geladen und für dieses auf dieselbe Weise G-Code erzeugt werden.

6 Experimente

6.1 Drucken individueller Funktionsbauteile

In diesem Experiment soll validiert werden, ob es mit der im Rahmen dieser Arbeit entwickelten Software möglich ist, verschiedene individuelle Funktionsbauteile mit dem Tropfenerzeuger zu drucken.

Material und Methoden

Gedruckt werden sollen in dieser Versuchsreihe $n = 5$ individuelle Schnapphaken, wie sie in Tabelle 6.1 abgebildet sind. Grundlage bilden die STL-Daten der mit einer CAD-Software konstruierten Schnapphaken. Aus den STL-Dateien wird dann, wie ausführlich in den Kapiteln 4 und 5 beschrieben, automatisch prozessspezifischer Maschinencode (G-Code) generiert. Dabei wird eine Schichthöhe von $H = 0.20\text{ mm}$ gewählt. Anschließend erfolgt die Vorbereitung des Druckjobs am Tropfenerzeuger, als Material kommt Terluran GP-35-ABS in Granulatform zum Einsatz. Ist der Drucker funktionsbereit, so wird der generierte Maschinencode an der Benutzeroberfläche des Tropfenerzeugers importiert, im Bauraum platziert und der Druckjob gestartet. Dieser Prozess wird für alle $n = 5$ Varianten durchgeführt. Dabei wird die Druckzeit t in $\text{min}:\text{s}$ gemessen und das entsprechende Bauteil anschließend entnommen. Prinzipiell wäre es aber auch möglich mehrere Schnapphaken innerhalb eines einzelnen Druckjobs zu drucken. Um aber nicht auch die Zeit für Leerfahrten zwischen den einzelnen Ebenen der verschiedenen Schnapphaken mitzustoppen, wird jeder Schnapphaken einzeln gedruckt. Zusätzlich werden die fertigen Schnapphaken einzeln gewogen.

Ergebnisse

Die Ergebnisse der einzelnen Messungen sind in Tabelle 6.1 zu sehen.

Auswertung und Diskussion

Es wurde gezeigt, dass mit der in dieser Arbeit implementierten Software individueller prozessspezifischer Maschinencode generiert werden kann, welcher vom Tropfenerzeuger gedruckt wird. Das Ergebnis sind in diesem Fall Funktionsbauteile in Form von Schnapphaken. Die Produktionszeit ist genauso, wie auch bei anderen Rapid Prototyping Verfahren, insbesondere auch dem FDM-Verfahren, vergleichsweise gering. Die Herstellungskosten beschränken sich im Wesentlichen auf den Betrieb des Tropfenerzeugers und die Materialkosten. So fallen bei einem Gesamtgewicht der Teile von $m = 11.3\text{ mg}$ Materialkosten von insgesamt ca. $0.03\text{ €} - 0.04\text{ €}$ für die Teile selbst an. Die Produktionszeit und die Herstellungskosten sind daher im Vergleich zum Spritzgießen sehr gering. Hier müsste zunächst, wie in Kapitel 2 erläutert, eine Form / ein Spritzgießwerkzeug erzeugt werden, z.B. durch Fräsen. Dies würde bis zur Herstellung des Spritzlings einen verhältnismäßig großen Zeitaufwand erfordern und zusätzliche Herstellungskosten für die Formen verursachen. Möchte man also schnell und kostengünstig (individuelle)

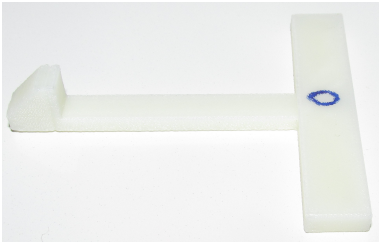
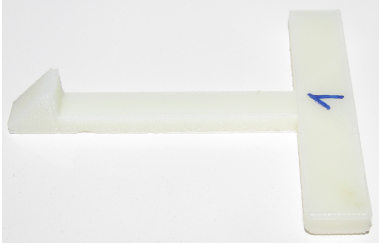
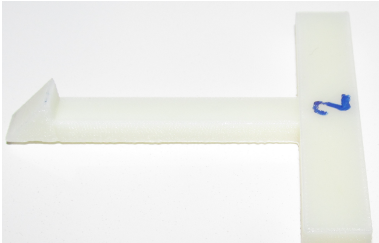

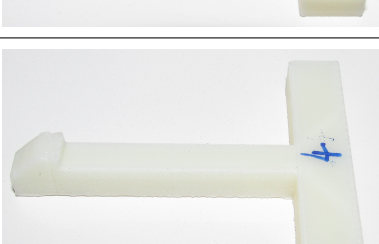
Variante	Bauteil	Druckzeit t [min:s]	Gewicht m [mg]
H0		28:35	1.6
H1		31:15	1.7
H2		38:42	2.2
H3		49:30	2.6
H4		58:29	3.2
		Σ : 206:31	Σ : 11.3

Tabelle 6.1: Übersicht über verschiedene mit dem Tropfenerzeuger gedruckte Varianten von Schnapphaken. Gemessen wurden die Druckzeit in $min:s$ und das Gewicht in mg als Größenordnung für die angefallenen Materialkosten (1 kg kostet ca. 3 €).

Bauteile in kleinen Losgrößen herstellen, so empfiehlt sich der Einsatz eines Rapid Prototyping Systems, wie beispielsweise der hier verwendete Tropfenerzeuger. Da die Materialkosten im Vergleich zu anderen nicht kunststoffbasierten 3D-Druckverfahren verhältnismäßig gering sind, muss sich der Tropfenerzeuger in weiteren Experimenten noch dem Vergleich mit FDM-

Druckern stellen. Dies betrifft insbesondere die optische Anmutung, wie beispielsweise die Oberflächenrauheit, als auch die Stabilität selbst, wie beispielsweise die Zugfestigkeit.

6.2 Validierung der Oberflächenrauheit

Im Rahmen dieser Arbeit wurden für einen neuartigen Druckprozess Druck- bzw. Füllstrategien entwickelt. Diese beinhalten das gezielte Setzen von Tropfen beim Berandungsdruck, sowie das Füllen eines Bauteils durch automatisches Setzen von Tropfen auf vorgegebenen Schraffursegmenten, welche den Abstand der Tropfenbreite B zueinander haben. Es ist daher zu erwarten, dass dies auch Auswirkungen auf die Oberflächenqualität eines Bauteils hat. Es soll daher validiert werden, wo die Unterschiede im Vergleich zu mit dem FDM-Verfahren gedruckten Bauteilen liegen, um einen Vergleich zum bisher marktführenden Verfahren im Bereich des 3D-Kunststoffdrucks zu schaffen.

Material und Methoden

Als Bauteile für die Untersuchungen wurden verschiedene Prüfkörper gedruckt: Tabelle 6.2 zeigt hierfür einen umfassenden Überblick. Für jeden in dieser Tabelle aufgeführten Prüfkörper

Maschine (Schichthöhe)	Anzahl Prüfkörper nach DIN 50125	Anzahl Prüfkörper nach DIN EN ISO 527-2	Material
Stratasys Fortus 360mc ($H = 0.254\text{ mm}$)	$n = 11$	$n = 11$ (Typ 1A)	ABS-M30
Stratasys Fortus 360mc ($H = 0.178\text{ mm}$)	$n = 11$	$n = 11$ (Typ 1A)	ABS-M30
Tropfenerzeuger ($H = 0.20\text{ mm}$)	$n = 11$		Terluran GP-35-ABS
Tropfenerzeuger ($H = 0.265\text{ mm}$)		$n = 11$ (Typ 1B)	Terluran GP-35-ABS

Tabelle 6.2: Übersicht über die für die Versuchsreihe gedruckten Prüfkörper. Die Norm DIN 50125 (2009) ist zur Prüfung metallischer Werkstoffe. Der entsprechende Prüfkörper (runder Zugstab) wurde jeweils stehend gedruckt und eignet sich daher besonders zur Untersuchung der Oberfläche in \mathbf{e}_z -Richtung, das heißt in Druckrichtung. Bei der Norm DIN EN ISO 527-2 (2012) handelt es sich hingegen um flache Zugproben zur Bestimmung der Zugeigenschaften von Kunststoffen. Diese Prüfkörper eignen sich besonders zur Untersuchung der Oberfläche innerhalb einer gedruckten \mathbf{e}_x - \mathbf{e}_y -Ebene.

per wurden mit einem Perthometer verschiedene Kennwerte für die Oberflächenrauheit gemessen. Abbildung 6.1 zeigt das für die Versuche verwendete Perthometer der Firma Mahr. Das Perthometer arbeitet auf Grundlage des Tastschnittverfahrens nach Norm DIN EN ISO 3274 (1998). Die Oberfläche wird hierbei zweidimensional erfasst, indem eine Vorschubeinrichtung das Tastsystem mit einer konstanten Geschwindigkeit horizontal über die Oberfläche bewegt. Aus der Lageverschiebung der Tastspitze (aus Diamant) in vertikaler Richtung, ergibt sich dann



Abbildung 6.1: Perthometer der Firma Mahr für die Messung verschiedener Kenngrößen der Oberflächenrauheit.

das Rauheitsprofil. Die Abbildungen 6.2, 6.3, 6.4 und 6.5 zeigen Ausschnitte aus der Versuchsdurchführung.

Die Taststrecke l_t setzt sich dabei aus einer Vorlaufstrecke, der Gesamtmessstrecke l_n und einer Nachlaufstrecke zusammen. Es gilt:

$$l_t = 5.6 \text{ mm} \quad (6.1)$$

und

$$l_n = 4.0 \text{ mm} . \quad (6.2)$$

Die Gesamtmessstrecke l_n setzt sich zusammen aus $n = 5$ gleich großen aneinandergereihten Einzelmessstrecken der Länge l_r . Es gilt:

$$l_r = 0.8 \text{ mm} . \quad (6.3)$$

Bei einer Rillenbreite RS_m mit

$$0.13 \text{ mm} < RS_m \leq 0.40 \text{ mm} \quad (6.4)$$

gilt nach den Normen DIN EN ISO 4288 (1998) bzw. ASME B46.1 (2009) für die Wahl der Grenzwellenlänge λ_c :

$$\lambda_c = 0.8 \text{ mm} . \quad (6.5)$$

Die Grenzwellenlänge λ_c eines Profilfilters entscheidet dabei, welche Wellenlängen der Rauheit und welche der Welligkeit zugeordnet werden. Hierbei gilt:

$$l_r = \lambda_c . \quad (6.6)$$

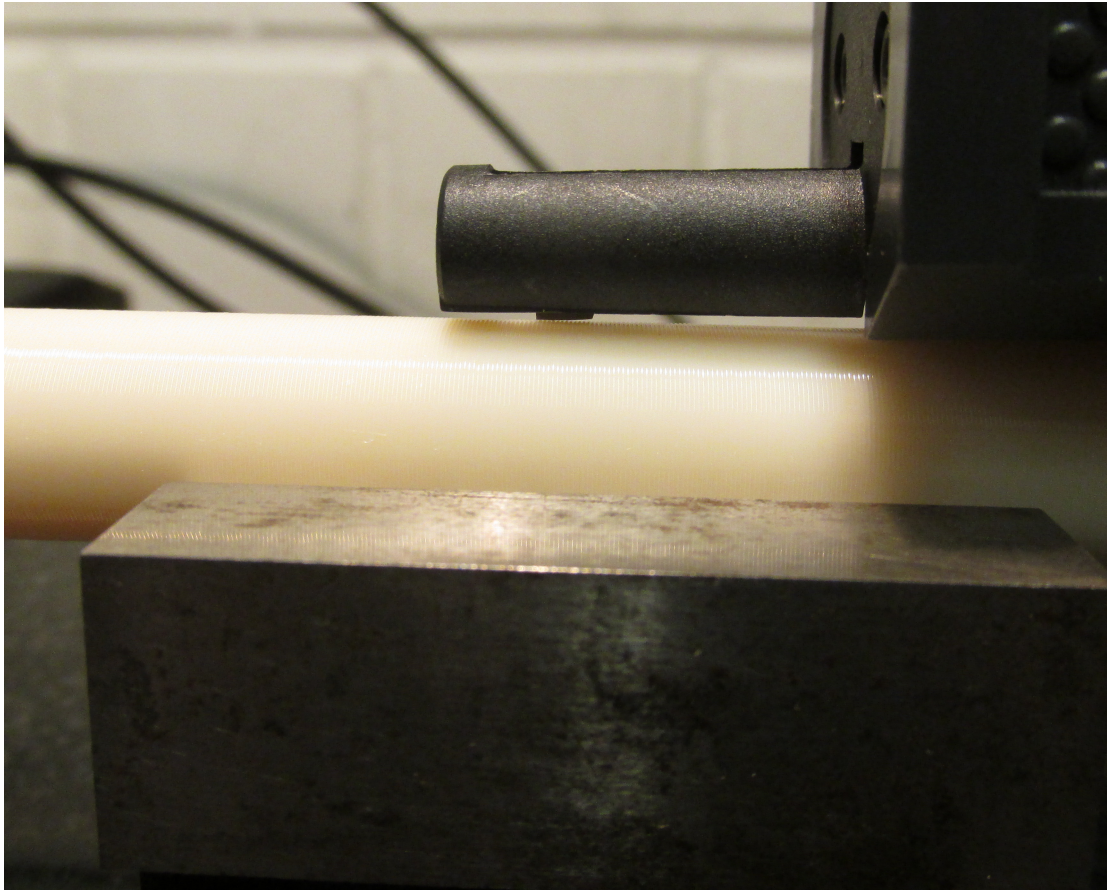


Abbildung 6.2: Messung der Oberflächenrauheit für Prüfkörper nach DIN 50125 (2009) (Stratasys-Maschine).

Die Vorlaufstrecke dient zum Einschwingen der Filter, die Nachlaufstrecke zum Ausschwingen der Filter. Bei jedem der Prüfkörper aus Tabelle 6.2 kann von einer Rillenbreite RS_m nach Gleichung (6.4) ausgegangen werden: bei jedem Prüfkörper nach Norm DIN 50125 (2009) liegt die jeweilige Schichthöhe H klar innerhalb der Grenzen. Bei den Prüfkörpern nach Norm DIN EN ISO 527-2 (2012) wird hingegen über eine gedruckte \mathbf{e}_x - \mathbf{e}_y -Ebene mit einem Schraffurwinkel von $\alpha = 45^\circ$ getastet. Die Tastrichtung geht dabei in \mathbf{e}_x -Richtung. Nach Pythagoras beträgt der Abstand zwischen zwei „Rillen“ daher $\sqrt{2}H$ für die Stratasys-Teile und $\sqrt{2}B$ für die mit dem Tropfenerzeuger gedruckten Teile, wobei hier speziell $B = 1.04H$ gilt. Die Abstände befinden sich also auch hier innerhalb der Grenzen von (6.4). Eine Auswahl der Parameter, wie oben beschrieben, ist daher für alle Prüfkörper und somit Messungen, gerechtfertigt.

Gemessen werden nach den Normen DIN EN ISO 4287 (2010) bzw. ASME B46.1 (2009) der Mittenrauhwert R_a , die Rauhtiefe R_z und die maximale Rauhtiefe R_{\max} . In den Abbildungen 6.6, 6.7, 6.8, 6.9, 6.10 und 6.11 sind Beispiele für Rauheitsprofile $R(x)$ in Abhängigkeit der Taststrecke x zu sehen. Eingezeichnet sind dabei auch die maximale Rauhtiefe R_{\max} und die Einzelrauhtiefen $R_{z,i}$, $i \in \{1, \dots, n\}$, der $n = 5$ Einzelmessstrecken l_r . Sei x_i der Beginn einer Einzelmessstrecke l_r . Dann gilt für die Einzelrauhtiefen:

$$R_{z,i} = \max_{x \in [x_i, x_i + l_r]} R(x) - \min_{x \in [x_i, x_i + l_r]} R(x). \quad (6.7)$$

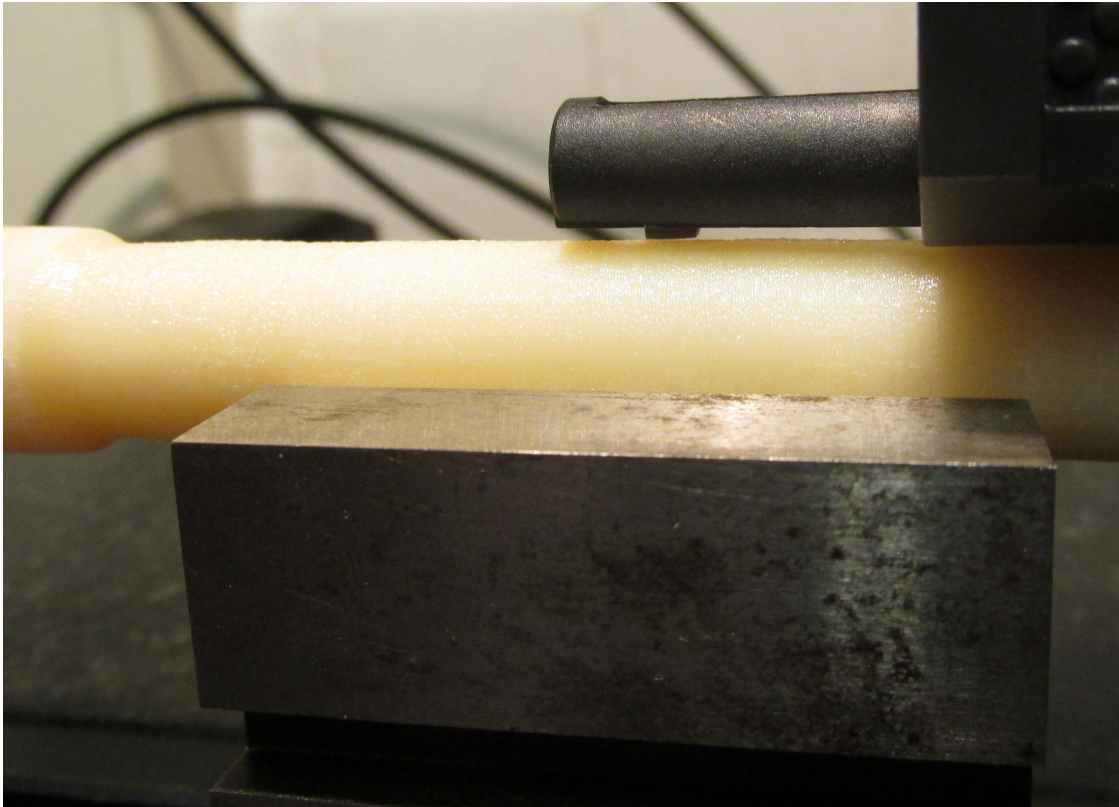


Abbildung 6.3: Messung der Oberflächenrauheit für Prüfkörper nach DIN 50125 (2009) (Tropfenerzeuger).

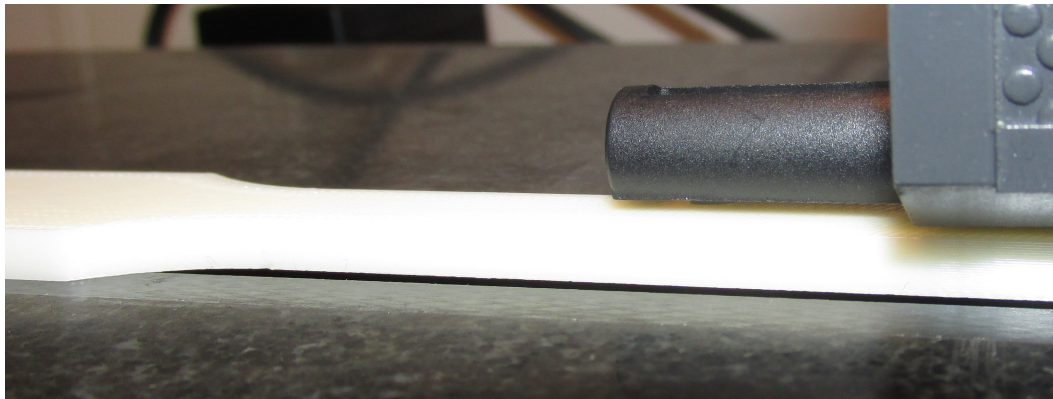


Abbildung 6.4: Messung der Oberflächenrauheit für Prüfkörper nach DIN EN ISO 527-2 (2012) (Stratasy-Maschine).

Für die maximale Rauhtiefe gilt:

$$R_{\max} = \max_{i \in \{1, \dots, 5\}} R_{z,i}. \quad (6.8)$$

Die Rauhtiefe selbst ist gegeben durch:

$$R_z = \frac{1}{5} \sum_{i=1}^5 R_{z,i}. \quad (6.9)$$

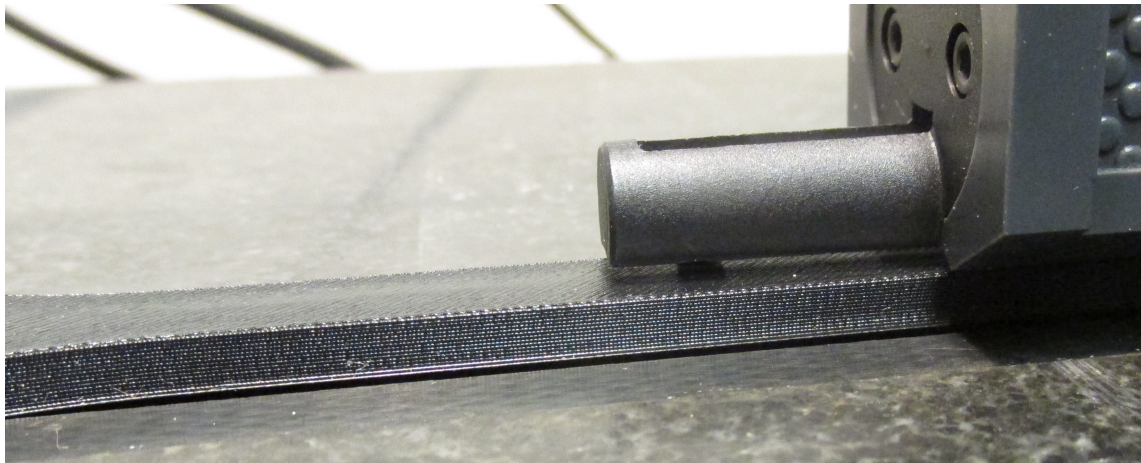


Abbildung 6.5: Messung der Oberflächenrauheit für Prüfkörper nach DIN EN ISO 527-2 (2012) (Tropfenerzeuger).

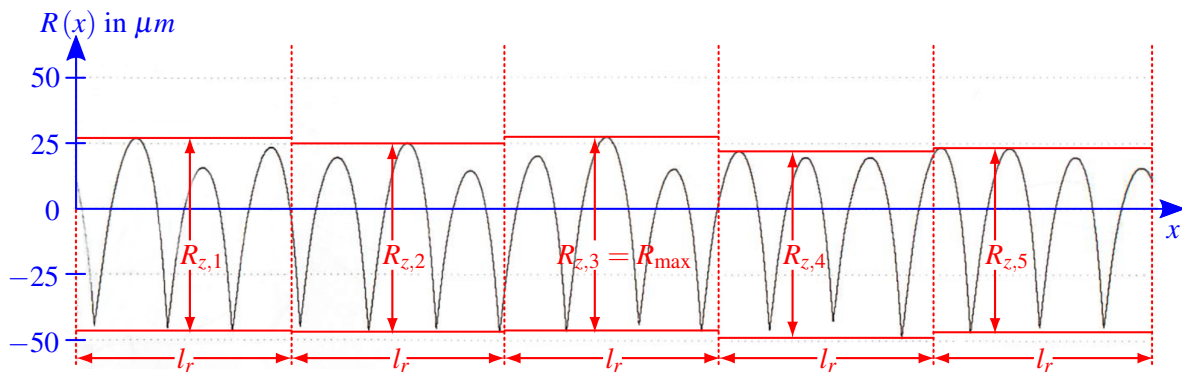


Abbildung 6.6: Rauheitsprofil in e_z -Richtung (entspricht Druckrichtung) für FDM-Bauteil nach Norm DIN 50125 (2009), gedruckt mit Stratasys Fortus 360mc (Schichthöhe: $H = 0.254\text{mm}$).

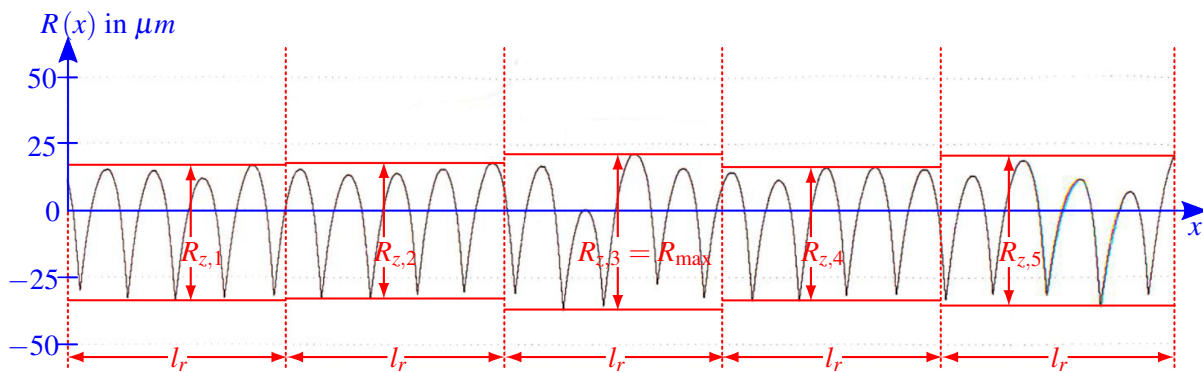


Abbildung 6.7: Rauheitsprofil in e_z -Richtung (entspricht Druckrichtung) für FDM-Bauteil nach Norm DIN 50125 (2009), gedruckt mit Stratasys Fortus 360mc (Schichthöhe: $H = 0.178\text{mm}$).

Der Mittenrauhwert R_a ist als arithmetischer Mittelwert aller Profilwertbeträge des Rauheits-

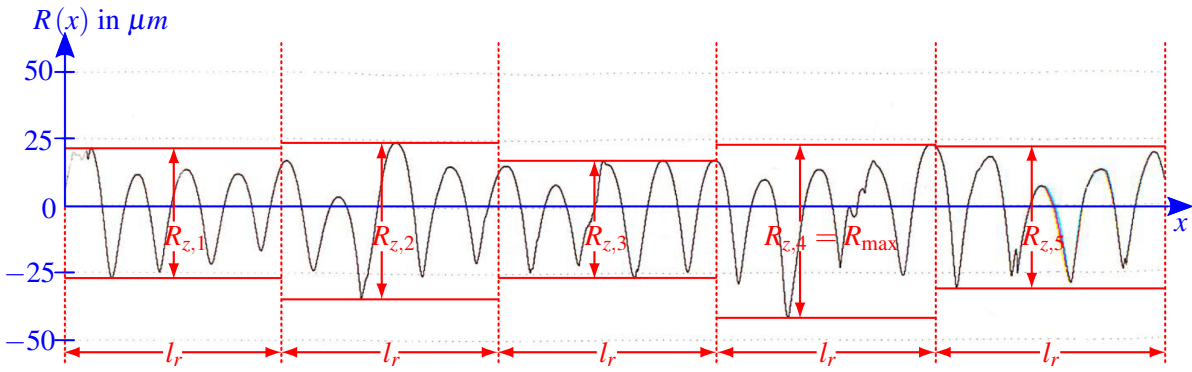


Abbildung 6.8: Rauheitsprofil in e_z -Richtung (entspricht Druckrichtung) für Bauteil nach Norm DIN 50125 (2009), gedruckt mit dem Tropfenerzeuger (Schichthöhe: $H = 0.20\text{mm}$).

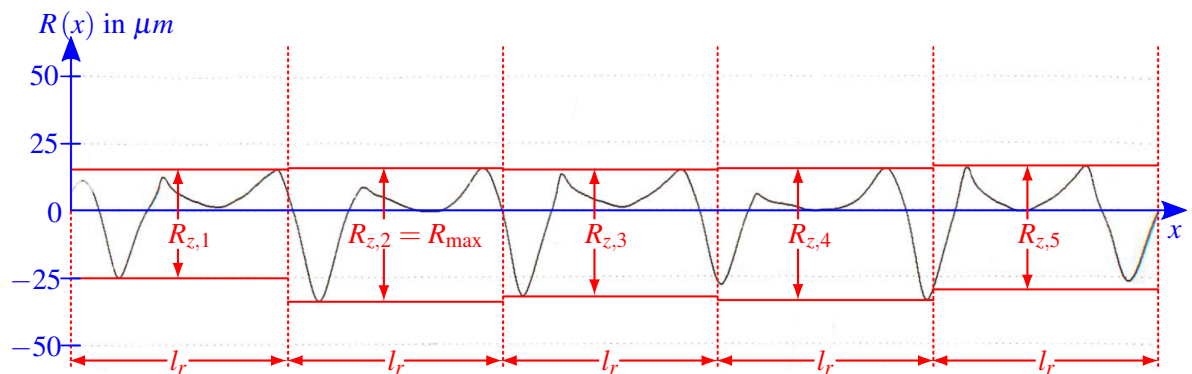


Abbildung 6.9: Rauheitsprofil in der e_x - e_y -Oberfläche für FDM-Bauteil nach Norm DIN EN ISO 527-2 (2009), gedruckt mit Stratasys Fortus 360mc (Schichthöhe: $H = 0.254\text{mm}$).

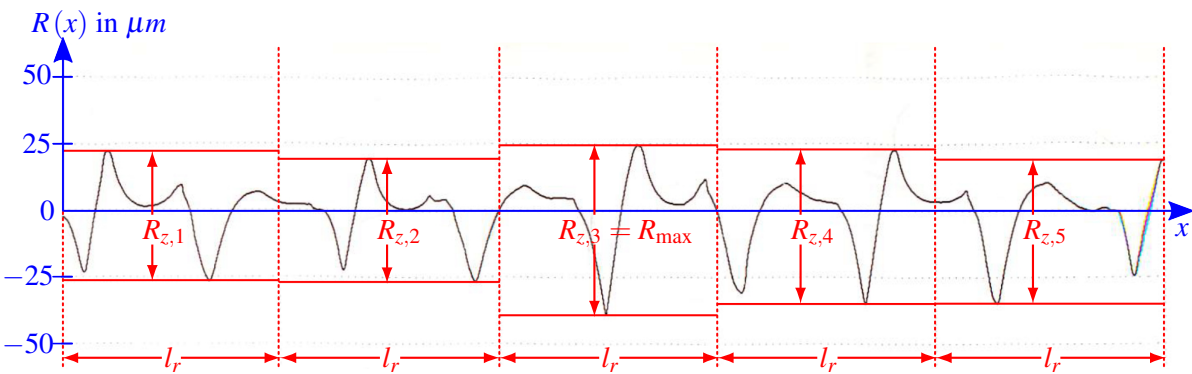


Abbildung 6.10: Rauheitsprofil in der e_x - e_y -Oberfläche für FDM-Bauteil nach Norm DIN EN ISO 527-2 (2012), gedruckt mit Stratasys Fortus 360mc (Schichthöhe: $H = 0.178\text{mm}$).

profils definiert. Es gilt:

$$R_a = \frac{1}{l_n} \int_0^{l_n} R(x) dx. \quad (6.10)$$

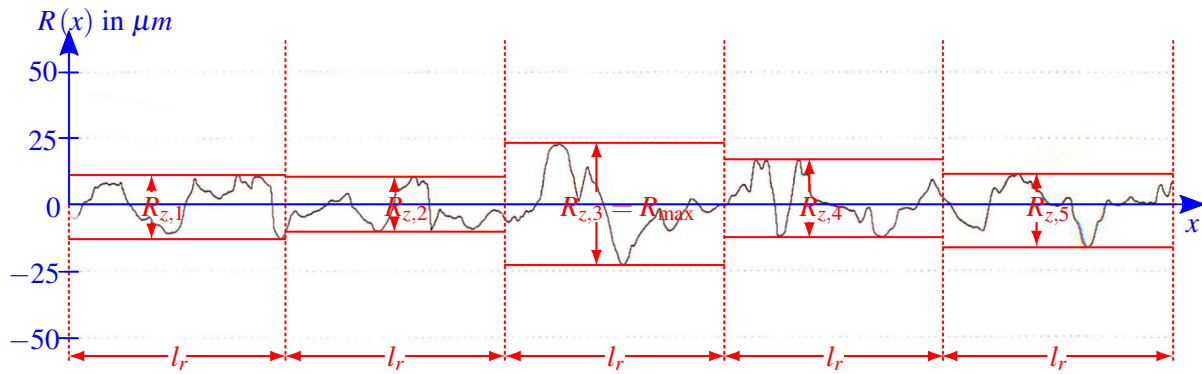


Abbildung 6.11: Rauheitsprofil in der e_x - e_y -Oberfläche für Bauteil nach Norm DIN EN ISO 527-2 (2012), gedruckt mit dem Tropfenerzeuger (Schichthöhe: $H = 0.265\text{ mm}$).

Ergebnisse

Die Tabellen 6.3, 6.4, 6.5, 6.6, 6.7 und 6.8 zeigen die Ergebnisse für alle sechs Versuchsreihen aus Tabelle 6.2.

Bauteil	R_a [μm]	R_z [μm]	R_{max} [μm]
S1	16.78	69.3	72.0
S2	17.38	75.2	76.4
S3	16.63	72.6	79.0
S4	17.03	74.2	77.5
S5	17.47	73.0	74.9
S6	17.46	76.2	81.9
S7	17.05	71.7	76.8
S8	17.38	74.1	78.7
S9	17.14	72.4	74.2
S10	17.34	73.8	81.5
S11	16.30	69.3	73.3
Mittelwerte:	$m_{R_a} = 17.09$	$m_{R_z} = 72.9$	$m_{R_{\text{max}}} = 76.9$
Standardabweichungen:	$\sigma_{R_a} = 0.38$	$\sigma_{R_z} = 2.2$	$\sigma_{R_{\text{max}}} = 3.2$

Tabelle 6.3: Oberflächenkennwerte für FDM-Bauteil nach Norm DIN 50125 (2009), gedruckt mit Stratasys Fortus 360mc (Schichthöhe: $H = 0.254\text{ mm}$). Werte sind in e_z -Richtung gemessen, das heißt in Druckrichtung des Bauteils (siehe Abbildung 6.2). Als Beispiel ist das Rauheitsprofil von Bauteil S9 in Abbildung 6.6 zu sehen.

Auswertung und Diskussion

Betrachtet man die Ergebnisse in den Tabellen 6.3, 6.4 und 6.5, so ist für die Prüfkörper nach der Norm DIN 50125 (2009) Folgendes festzuhalten. Die Mittelwerte der Mittenrauheitswerte R_a und die Mittelwerte der Rauhtiefen R_z stimmen bei den mit der Stratasys-Maschine mit der Schichthöhe $H = 0.178\text{ mm}$ und den mit dem Tropfenerzeuger mit der Schichthöhe $H = 0.20\text{ mm}$ gedruckten Teile nahezu überein: es gilt $m_{R_a} = 12.17\ \mu\text{m}$ bzw. $m_{R_a} = 12.26\ \mu\text{m}$ und $m_{R_z} = 54.4\ \mu\text{m}$ bzw. $m_{R_z} = 56.9\ \mu\text{m}$. Die Streuung ist jedoch bei den mit dem Tropfenerzeuger

Bauteil	R_a [μm]	R_z [μm]	R_{max} [μm]
S12	12.34	53.4	58.8
S13	12.29	52.6	54.4
S14	12.39	56.0	62.8
S15	12.00	69.3	72.0
S16	11.92	52.3	54.2
S17	11.84	51.0	53.4
S18	12.04	50.9	52.1
S19	11.97	51.6	53.0
S20	12.19	55.2	75.8
S21	12.68	53.5	53.1
S22	12.23	53.1	54.8
Mittelwerte:	$m_{R_a} = 12.17$	$m_{R_z} = 54.4$	$m_{R_{max}} = 58.6$
Standardabweichungen:	$\sigma_{R_a} = 0.25$	$\sigma_{R_z} = 5.2$	$\sigma_{R_{max}} = 8.2$

Tabelle 6.4: Oberflächenkennwerte für FDM-Bauteil nach Norm DIN 50125 (2009), gedruckt mit Stratasys Fortus 360mc (Schichthöhe: $H = 0.178\text{mm}$). Werte sind in e_z -Richtung gemessen, das heißt in Druckrichtung des Bauteils (siehe Abbildung 6.2). Als Beispiel ist das Rauheitsprofil von Bauteil S12 in Abbildung 6.7 zu sehen.

Bauteil	R_a [μm]	R_z [μm]	R_{max} [μm]
A1	8.40	51.7	69.5
A2	12.98	54.0	65.1
A3	13.64	67.1	88.8
A4	13.53	62.6	68.3
A5	13.42	62.9	76.1
A6	7.95	34.4	50.9
A7	12.32	54.2	65.2
A8	10.82	50.4	56.7
A9	15.67	75.9	107.0
A10	12.40	53.9	69.7
A11	13.72	58.6	61.8
Mittelwerte:	$m_{R_a} = 12.26$	$m_{R_z} = 56.9$	$m_{R_{max}} = 70.8$
Standardabweichungen:	$\sigma_{R_a} = 2.34$	$\sigma_{R_z} = 10.7$	$\sigma_{R_{max}} = 15.5$

Tabelle 6.5: Oberflächenkennwerte für Bauteil nach Norm DIN 50125 (2009), gedruckt mit Tropfenerzeuger (Schichthöhe: $H = 0.20\text{mm}$). Werte sind in e_z -Richtung gemessen, das heißt in Druckrichtung des Bauteils (siehe Abbildung 6.3). Als Beispiel ist das Rauheitsprofil von Bauteil A7 in Abbildung 6.8 zu sehen.

gedruckten Teilen etwas größer: es gilt $\sigma_{R_a} = 0.25\ \mu m$ bzw. $\sigma_{R_a} = 2.34\ \mu m$ und $\sigma_{R_z} = 5.2\ \mu m$ bzw. $\sigma_{R_z} = 10.7\ \mu m$. Dies spiegelt sich letztlich auch in den Werten R_{max} für die maximale Rauhtiefe wieder: hier ist der Mittelwert der maximalen Rauhtiefen beim Tropfenerzeuger mit $m_{R_{max}} = 70.8\ \mu m$ etwas höher als beim FDM-Verfahren mit $m_{R_{max}} = 58.6\ \mu m$. Zieht man nun die Prüfkörper der Stratasys-Maschine mit der Schichthöhe $H = 0.254\text{mm}$ zum Vergleich hinzu, so ergeben sich deutlich höhere Mittelwerte: es gilt $m_{R_a} = 17.09\ \mu m$ ($\sigma_{R_a} = 0.38\ \mu m$), $m_{R_z} = 72.9\ \mu m$ ($\sigma_{R_z} = 2.2\ \mu m$) und $m_{R_{max}} = 76.9\ \mu m$.

Bauteil	R_a [μm]	R_z [μm]	R_{max} [μm]
S23	8.99	46.4	49.7
S24	11.24	68.4	85.1
S25	5.69	35.2	47.3
S26	7.08	41.7	48.1
S27	9.16	53.8	57.6
S28	8.12	44.9	65.4
S29	9.13	54.6	74.1
S30	8.25	51.6	59.4
S31	9.26	55.3	60.0
S32	6.11	38.6	48.5
S33	13.54	72.9	74.5
Mittelwerte:	$m_{R_a} = 8.78$	$m_{R_z} = 51.2$	$m_{R_{max}} = 60.9$
Standardabweichungen:	$\sigma_{R_a} = 2.23$	$\sigma_{R_z} = 11.7$	$\sigma_{R_{max}} = 12.7$

Tabelle 6.6: Oberflächenkennwerte für FDM-Bauteil nach Norm DIN EN ISO 527-2 (2012), gedruckt mit Stratasys Fortus 360mc (Schichthöhe: $H = 0.254\text{mm}$). Werte sind auf der Oberfläche innerhalb einer gedruckten \mathbf{e}_x - \mathbf{e}_y -Ebene gemessen (siehe Abbildung 6.4). Als Beispiel ist das Rauheitsprofil von Bauteil S23 in Abbildung 6.9 zu sehen.

Bauteil	R_a [μm]	R_z [μm]	R_{max} [μm]
S34	5.24	39.9	53.7
S35	10.08	71.7	94.6
S36	11.94	67.1	76.9
S37	9.00	54.3	63.7
S38	15.58	105.0	116.0
S39	5.98	39.3	47.2
S40	11.19	79.7	116.0
S41	11.20	65.2	79.0
S42	7.95	50.4	61.3
S43	5.13	45.9	58.7
S44	5.16	33.5	40.5
Mittelwerte:	$m_{R_a} = 8.95$	$m_{R_z} = 59.3$	$m_{R_{max}} = 73.4$
Standardabweichungen:	$\sigma_{R_a} = 3.42$	$\sigma_{R_z} = 21.2$	$\sigma_{R_{max}} = 26.0$

Tabelle 6.7: Oberflächenkennwerte für FDM-Bauteil nach Norm DIN EN ISO 527-2 (2012), gedruckt mit Stratasys Fortus 360mc (Schichthöhe: $H = 0.178\text{mm}$). Werte sind auf der Oberfläche innerhalb einer gedruckten \mathbf{e}_x - \mathbf{e}_y -Ebene gemessen (siehe Abbildung 6.4). Als Beispiel ist das Rauheitsprofil von Bauteil S37 in Abbildung 6.10 zu sehen.

Betrachtet man hingegen die Ergebnisse in den Tabellen 6.6, 6.7 und 6.8, so ist für die Prüfkörper nach der Norm DIN EN ISO 527-2 (2012) ein anderes Ergebnis zu beobachten. Hier sind die Mittelwerte der Mittenrauheitswerte R_a bei den mit der Stratasys-Maschine mit der Schichthöhe $H = 0.254\text{mm}$ und mit der Schichthöhe $H = 0.178\text{mm}$ nahezu vergleichbar: es gilt $m_{R_a} = 8.78\mu m$ bzw. $m_{R_a} = 8.95\mu m$. Für die Streuungen gilt $\sigma_{R_a} = 2.23\mu m$ bzw. $\sigma_{R_a} = 3.42\mu m$. Die Mittelwerte der Rauhtiefen R_z und der maximalen Rauhtiefen R_{max} sprechen sogar für die

Bauteil	R_a [μm]	R_z [μm]	R_{max} [μm]
A12	8.56	40.6	54.3
A13	4.82	21.0	22.1
A14	8.43	43.9	61.9
A15	5.47	29.3	32.6
A16	5.40	24.2	33.8
A17	9.58	56.6	71.4
A18	3.99	21.9	30.5
A19	6.28	29.9	46.2
A20	4.85	26.7	33.6
A21	4.60	20.0	21.7
A22	7.00	33.9	49.9
Mittelwerte:	$m_{R_a} = 6.27$	$m_{R_z} = 31.6$	$m_{R_{max}} = 41.6$
Standardabweichungen:	$\sigma_{R_a} = 1.87$	$\sigma_{R_z} = 11.4$	$\sigma_{R_{max}} = 16.3$

Tabelle 6.8: Oberflächenkennwerte für Bauteil nach Norm DIN EN ISO 527-2 (2012), gedruckt mit Tropfenerzeuger (Schichthöhe: $H = 0.265\text{ mm}$). Werte sind auf der Oberfläche innerhalb einer gedruckten e_x - e_y -Ebene gemessen (siehe Abbildung 6.5). Als Beispiel ist das Rauheitsprofil von Bauteil A19 in Abbildung 6.11 zu sehen.

mit der größeren Schichthöhe gedruckten Teile: es gilt $m_{R_z} = 51.2\ \mu m$ ($\sigma_{R_z} = 11.7\ \mu m$) bzw. $m_{R_z} = 59.3\ \mu m$ ($\sigma_{R_z} = 21.2\ \mu m$) und $m_{R_{max}} = 60.9\ \mu m$ bzw. $m_{R_{max}} = 73.4\ \mu m$. Zieht man nun die mit dem Tropfenerzeuger gedruckten Bauteile zum Vergleich hinzu, so sind diese denen der Stratasys-Maschine bei jeder gemessenen Kenngröße deutlich überlegen: es gilt $m_{R_a} = 6.27\ \mu m$ ($\sigma_{R_a} = 1.87\ \mu m$), $m_{R_z} = 31.6\ \mu m$ ($\sigma_{R_z} = 11.4\ \mu m$) und $m_{R_{max}} = 41.6\ \mu m$.

Bei den Prüfkörpern nach der Norm DIN 50125 (2009) sind die Ergebnisse vergleichbar, wenn man den Tropfenerzeuger mit der Schichthöhe $H = 0.20\text{ mm}$ und die Stratasys-Maschine mit der Schichthöhe $H = 0.178\text{ mm}$ miteinander vergleicht. Es ist also durchaus zu erwarten, dass der Tropfenerzeuger bei selben Schichthöhen besser abschneiden würde. Nichtsdestotrotz deckt sich das erzielte Ergebnisse mit der Tatsache, dass im Rahmen dieser Arbeit kein Gedächtnis zwischen benachbarten Schichten entwickelt wurde. So wurden Tropfen beispielsweise auch nicht gezielt auf Lücke gesetzt. Da das Rauheitsprofil bei diesen Teilen aber gerade in Druckrichtung (e_z -Richtung) gemessen wurde, sind auch die im Vergleich relativ hohen Streuungen zu erklären. Im Vergleich dazu wurden in dieser Arbeit gezielt die Daten innerhalb einer gedruckten Schicht optimiert. So konnte bei den nach Norm DIN EN ISO 527-2 (2012) gedruckten Prüfkörpern der Mittelwert der Mittenrauheitswerte um 29.6% von $m_{R_a} = 8.78\ \mu m$ auf $m_{R_a} = 6.27\ \mu m$ verbessert werden. Der Mittelwert der Rauhtiefen konnte um 38.3% von $m_{R_z} = 51.2\ \mu m$ auf $m_{R_z} = 31.6\ \mu m$ und Mittelwert der maximalen Rauhtiefen um 31.7% von $m_{R_{max}} = 60.9\ \mu m$ auf $m_{R_{max}} = 41.6\ \mu m$ verbessert werden.

6.3 Validierung der Zugfestigkeit

Da es sich bei diesem Druckprozess um ein tropfenbasiertes Verfahren handelt, ist zu erwarten, dass dies nicht nur Auswirkungen auf die Oberflächenqualität hat, sondern auch auf die Stabilität. Ziel des Projekts war es dabei unter Anderem, eine Zugfestigkeit der gedruckten Teile von mindestens 25% der Zugfestigkeit des verwendeten Kunststoffes zu erreichen. Im Rahmen

eines Experiments soll daher die Zugfestigkeit von mit dem Tropfenerzeuger gedruckten Prüfkörpern validiert und ein Vergleich zur Zugfestigkeit der mit dem FDM-Verfahren gedruckten Prüfkörper gezogen werden.

Material und Methoden

Als Bauteile für die Untersuchungen dienen die Prüfkörper nach Norm DIN EN ISO 527-2 (2012), welche bereits bei den Experimenten in Kapitel 6.2 zur Beurteilung der Oberflächenqualität verwendet wurden: Tabelle 6.9 zeigt hierfür nochmals einen Überblick. Zur Messung der

Maschine (Schichthöhe)	Anzahl Prüfkörper nach DIN EN ISO 527-2	Material
Stratasys Fortus 360mc ($H = 0.254\text{ mm}$)	$n = 10$ (Typ 1A)	ABS-M30
Stratasys Fortus 360mc ($H = 0.178\text{ mm}$)	$n = 10$ (Typ 1A)	ABS-M30
Tropfenerzeuger ($H = 0.265\text{ mm}$)	$n = 10$ (Typ 1B)	Terluran GP-35-ABS

Tabelle 6.9: Übersicht über die für die Versuchsreihe verwendeten Prüfkörper. Bei der Norm DIN EN ISO 527-2 (2012) handelt es sich um flache Zugproben zur Bestimmung der Zugeigenschaften von Kunststoffen. Im Vergleich zu Tabelle 6.2 wurden nur $n = 10$ Prüfkörper für die Zugsuche verwendet, um jeweils ein Original aufzubewahren.

Zugeigenschaften wurde eine Zugprüfmaschine von Zwick / Roell verwendet. Abbildung 6.12 zeigt den für die Zugversuche typischen Geräteaufbau. Der Anfangsabstand der Einspannklemmen (Einspannlänge) ist dabei nach Norm DIN EN ISO 527-2 (2012)

$$L = 115\text{ mm} \quad (6.11)$$

für alle Prüfkörper vom Typ 1A und 1B. Ermittelt werden sollen für alle Prüfkörper aus Tabelle 6.9 die Zugeigenschaften nach Norm DIN EN ISO 527-1 (2012). Die Prüfgeschwindigkeit der Zugprüfmaschine ist normgerecht

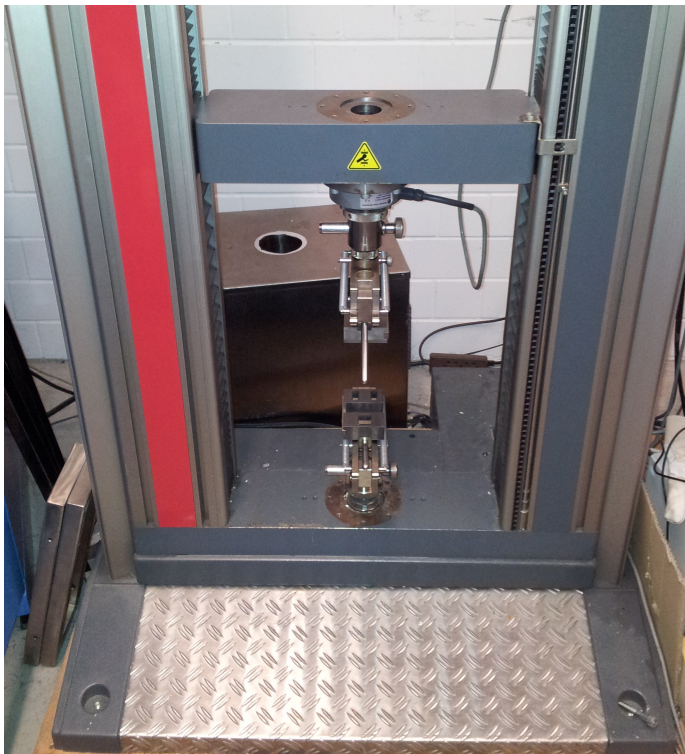
$$v = 50 \frac{\text{mm}}{\text{min}}. \quad (6.12)$$

Mit dieser Geschwindigkeit wird der bewegliche Teil der Einspannvorrichtung nach oben gefahren, bis der jeweilige Prüfkörper zerbricht (siehe hierzu Abbildung 6.13). Aufgezeichnet wird dabei mit einem Extensometer (Dehnungssensor) die Dehnung ΔL in mm des Prüfkörpers und mit einem Kraftsensor die dafür benötigte Kraft F in N (Newton). Der entsprechende Dehnungswert kann durch

$$\varepsilon = \frac{\Delta L}{L} \quad (6.13)$$

berechnet werden, was der prozentualen Verlängerung eines Zugstabs während der Prüfung entspricht. Die während der Prüfung auftretenden Spannungen können über die Vorschrift

$$\sigma = \frac{F}{A} \quad (6.14)$$



(a) Zugprüfmaschine von Zwick / Roell.



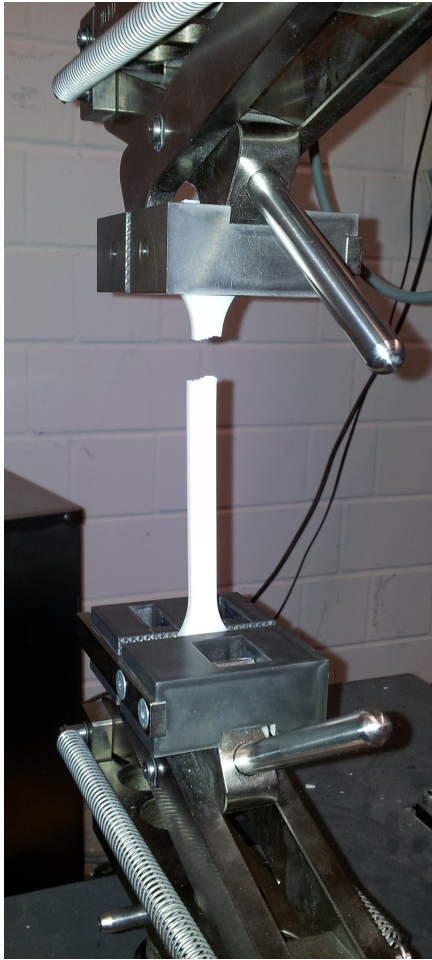
(b) Eingespannter Zugstab.

Abbildung 6.12: Geräteaufbau für die Zugversuche nach Norm DIN EN ISO 527-1 (2012). Die Maschine besteht aus einem festsitzenden (unterer Teil der Einspannvorrichtung) und einem beweglichen (oberer Teil der Einspannvorrichtung) mechanischen Träger.

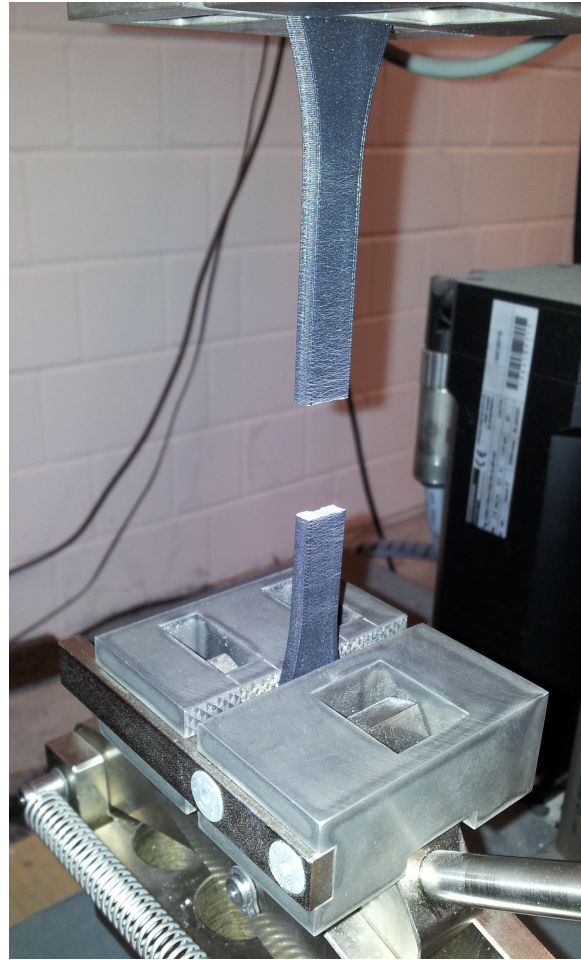
berechnet werden, wobei A die Querschnittsfläche in mm^2 des engen Teils eines Probekörpers ist, das heißt die Fläche, die senkrecht zur Zugrichtung steht. Die Dicke d der Zugproben (Typ 1A und Typ 1B) ist dabei nach Norm DIN EN ISO 527-2 (2012) $d = 4.0\text{mm} \pm 0.2\text{mm}$, die Breite b_1 des engen Teils entsprechend $b_1 = 10.0\text{mm} \pm 0.2\text{mm}$. Für die Querschnittsfläche gilt entsprechend: $A = b_1 d$. Alle für die Zugversuche verwendeten Prüfkörper befanden sich innerhalb dieser Toleranzen, so dass die Zugversuche normgerecht durchgeführt werden konnten. Die Einheit von σ ist MPa (Megapascal). Ein Megapascal ist dabei dasselbe wie ein N/mm^2 . Aus den gemessenen Werten für ε und σ können für jeden Prüfkörper Spannungs- / Dehnungskurven ermittelt werden. Dies ist dann im Ergebnisteil zu sehen.

Des Weiteren werden für alle Prüfkörper auch bestimmte charakteristische Zugspannungs- bzw. Zugdehnungswerte festgehalten. Dies sind die Zugfestigkeit σ_m , die Streckspannung σ_y und die Bruchspannung σ_b bzw. die Dehnung der Zugfestigkeit ε_m , die Streckdehnung ε_y und die Bruchdehnung ε_b . Die Zugfestigkeit σ_m ist dabei die Spannung beim ersten Spannungsmaximum während des Zugversuchs. Die Streckspannung σ_y ist die Spannung bei der Streckdehnung (Erklärung folgt unterhalb). Die Bruchspannung σ_b ist die Spannung, bei der ein Zugstab bricht. Die Dehnung der Zugfestigkeit ε_m ist die Dehnung, die auftritt, wenn die Zugfestigkeit erreicht ist. Die Streckdehnung ε_y ist der erste Dehnungswert bei einem Zugversuch, bei dem die Dehnung steigt, obwohl die Spannung nicht steigt. Die Bruchdehnung ε_b ist der letzte Dehnungswert, bevor ein Spannungsabfall auf $\leq 10\%$ des Festigkeitswerts erfolgt.

Alle Zugversuche wurden am 31. Oktober 2013 durchgeführt. Die gedruckten Proben waren zu diesem Zeitpunkt alle älter als 14 Tage.



(a) FDM-Zugstab (Stratasys).



(b) Zugstab (Tropfenerzeuger).

Abbildung 6.13: Bruch der Zugstäbe nach Durchführung eines Zugversuchs nach Norm DIN EN ISO 527-1 (2012).

Ergebnisse

Die Tabellen 6.10, 6.11 und 6.12 zeigen alle charakteristischen Zugspannungs- und Zugdehnungswerte. Die Abbildungen 6.14, 6.15 und 6.16 zeigen die entsprechenden Spannungs- / Dehnungskurven.

Auswertung und Diskussion

Aus den Materialdatenblättern des Kunststoffes Terluran GP-35-ABS ist eine Zugfestigkeit des Materials von $\sigma_{m_0} = 44 \text{ MPa}$ zu entnehmen. Dem gegenüber steht ein durchschnittlicher Wert von $m_{\sigma_m} = 34.91 \text{ MPa}$ bei den mit dem Tropfenerzeuger gedruckten Zugstäbe. Dies entspricht also 79.34% des Zugfestigkeitswert und übertrifft den zu Beginn der Arbeit festgelegten Wert von 25% um ein Vielfaches. Die Streckdehnung von Terluran GP-35-ABS liegt bei $\epsilon_{y_0} = 2.4\%$. Demgegenüber steht eine durchschnittliche Streckdehnung der mit dem Tropfenerzeuger gedruckten Teile von $\epsilon_y = 5.26\%$. Das Material wird also durch den Druckprozess wesentlich dehnbarer. Die durchschnittliche Bruchspannung liegt bei $m_{\sigma_b} = 29.67 \text{ MPa}$, die durchschnittliche Bruchdehnung bei $\epsilon_{\sigma_b} = 6.53\%$.

Aus den Materialdatenblättern des FDM-Materials ABS-M30 ist eine Zugfestigkeit des Ma-

Bauteil	$\sigma_m = \sigma_y$ [MPa]	$\varepsilon_m = \varepsilon_y$ [%]	σ_b [MPa]	ε_b [%]
S23	31.64	4.63	30.28	5.16
S24	31.69	4.72	30.50	5.27
S25	32.51	4.63	31.05	5.47
S27	31.27	4.63	29.60	5.60
S28	30.65	4.66	29.65	5.05
S29	32.36	4.34	30.09	4.69
S30	31.45	4.63	30.25	5.30
S31	30.94	4.32	29.03	4.97
S32	31.89	4.30	29.40	6.00
S33	31.35	4.33	27.63	7.45
Mittelwerte:	$m_{\sigma_{m/y}} = 31.57$	$m_{\varepsilon_{m/y}} = 4.52$	$m_{\sigma_b} = 29.75$	$m_{\varepsilon_b} = 5.50$
Standardabweichungen:	$\sigma_{\sigma_{m/y}} = 0.58$	$\sigma_{\varepsilon_{m/y}} = 0.17$	$\sigma_{\sigma_b} = 0.95$	$\sigma_{\varepsilon_b} = 0.77$

Tabelle 6.10: Zugeigenschaften für FDM-Bauteile nach Norm DIN EN ISO 527-2 (2012), gedruckt mit Stratasys Fortus 360mc (Schichthöhe: $H = 0.254\text{mm}$). In Abbildung 6.14 sind die entsprechenden Spannungs- / Dehnungskurven zu sehen.

Bauteil	$\sigma_m = \sigma_y$ [MPa]	$\varepsilon_m = \varepsilon_y$ [%]	σ_b [MPa]	ε_b [%]
S34	29.40	4.62	27.14	6.36
S35	28.64	4.63	28.09	5.43
S36	28.01	4.63	27.64	5.53
S37	29.03	4.58	28.07	6.42
S38	28.79	4.53	27.82	6.24
S39	27.92	5.26	27.57	6.14
S40	29.48	4.45	29.13	4.91
S42	29.25	4.68	28.86	5.46
S43	28.91	4.34	28.35	4.95
S44	29.36	4.56	28.61	5.62
Mittelwerte:	$m_{\sigma_{m/y}} = 28.88$	$m_{\varepsilon_{m/y}} = 4.63$	$m_{\sigma_b} = 28.13$	$m_{\varepsilon_b} = 5.71$
Standardabweichungen:	$\sigma_{\sigma_{m/y}} = 0.55$	$\sigma_{\varepsilon_{m/y}} = 0.24$	$\sigma_{\sigma_b} = 0.62$	$\sigma_{\varepsilon_b} = 0.56$

Tabelle 6.11: Zugeigenschaften für FDM-Bauteile nach Norm DIN EN ISO 527-2 (2012), gedruckt mit Stratasys Fortus 360mc (Schichthöhe: $H = 0.178\text{mm}$). In Abbildung 6.15 sind die entsprechenden Spannungs- / Dehnungskurven zu sehen.

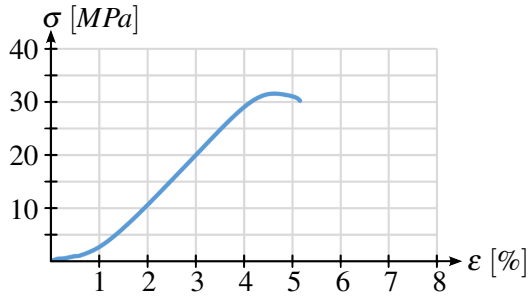
terials von $\sigma_{m_0} = 36\text{MPa}$ zu entnehmen. Dem gegenüber steht ein durchschnittlicher Wert von $m_{\sigma_m} = 31.57\text{MPa}$ (Schichthöhe: $H = 0.254\text{mm}$) bzw. $m_{\sigma_m} = 28.88\text{MPa}$ (Schichthöhe: $H = 0.178\text{mm}$) bei den mit der Stratasys Fortus 360mc gedruckten Zugstäben. Dies entspricht also 87.70% bzw. 80.22% des Zugfestigkeitswert. Die Streckdehnung von ABS-M30 liegt bei $\varepsilon_{y_0} = 4.0\%$. Demgegenüber steht eine durchschnittliche Streckdehnung der mit der Stratasys Fortus 360mc gedruckten Teile von $\varepsilon_y = 4.52\%$ bzw. $\varepsilon_y = 4.63\%$. Interessant ist dabei die deutlich höhere Zugfestigkeit bei den mit der größeren Schichthöhe gedruckten Teile. Die durchschnittliche Bruchspannung liegt bei $m_{\sigma_b} = 29.75\text{MPa}$ (Schichthöhe: $H = 0.254\text{mm}$) bzw. $m_{\sigma_b} = 28.13\text{MPa}$ (Schichthöhe: $H = 0.178\text{mm}$), die durchschnittliche Bruchdehnung bei $\varepsilon_{\sigma_b} = 5.50\%$ bzw. $\varepsilon_{\sigma_b} = 5.71\%$.

Bauteil	$\sigma_m = \sigma_y$ [MPa]	$\varepsilon_m = \varepsilon_y$ [%]	σ_b [MPa]	ε_b [%]
A12	36.04	5.33	28.81	6.55
A13	34.85	5.30	29.48	6.18
A14	32.97	5.04	29.79	6.47
A15	36.17	5.36	30.33	6.21
A16	35.59	5.34	29.08	7.27
A17	33.31	4.78	29.25	6.32
A18	35.70	5.21	28.56	6.98
A19	34.60	5.39	30.30	6.68
A21	35.53	5.55	31.37	6.34
A22	34.32	5.27	29.72	6.32
Mittelwerte:	$m_{\sigma_{m/y}} = 34.91$	$m_{\varepsilon_{m/y}} = 5.26$	$m_{\sigma_b} = 29.67$	$m_{\varepsilon_b} = 6.53$
Standardabweichungen:	$\sigma_{\sigma_{m/y}} = 1.11$	$\sigma_{\varepsilon_{m/y}} = 0.21$	$\sigma_{\sigma_b} = 0.84$	$\sigma_{\varepsilon_b} = 0.35$

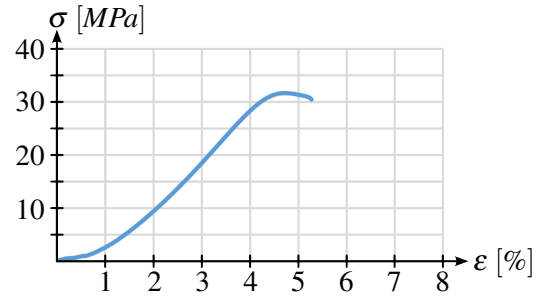
Tabelle 6.12: Zugeigenschaften für Bauteile nach Norm DIN EN ISO 527-2 (2012), gedruckt mit Tropfenerzeuger (Schichthöhe: $H = 0.265\text{ mm}$). In Abbildung 6.16 sind die entsprechenden Spannungs- / Dehnungskurven zu sehen.

Zusammenfassend ist zu sagen, dass ein unmittelbarer Vergleich zwischen beiden Verfahren nur sehr eingeschränkt möglich ist, da die Baumaterialien unterschiedlich waren. Ein besserer Vergleich ist erst möglich, sobald Prüfkörper mit demselben Material gedruckt werden können. Dann gilt es aber auch andere Parameter, wie beispielsweise die Schichthöhe H in den Vergleich mit einzubeziehen. Ein unmittelbarer Zusammenhang zwischen Zugfestigkeit und Schichthöhe ist in diesem Vergleich noch nicht erkennbar. Da der hier verwendete tropfenbasierte Ansatz jedoch Schritt für Schritt weiterentwickelt wird, ist davon auszugehen, dass dadurch auch die Zugfestigkeit positiv beeinflusst werden kann, spätestens wenn es möglich ist, bei der Generierung von Maschinencode ein Gedächtnis zwischen benachbarten Schichten aufzubauen und somit Tropfen auf Lücke setzen zu können. Betrachtet man die Ergebnisse aus Ahn et al. (2002), wo bei Stratasys FDM-Bauteilen eine Zugfestigkeit von 65% – 72% gegenüber der Zugfestigkeit des spritzgegossenen Kunststoffes ABS P400 festgestellt wurde, so ist dieser Wert im Rahmen dieser Arbeit übertroffen. Kritisch bleibt aber nach wie vorher der Vergleich mit unterschiedlichen Materialien.

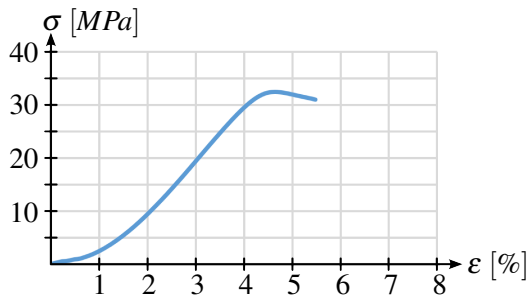
Abschließend zeigen die Abbildungen 6.17, 6.18 und 6.19 Aufnahmen, die von den zerbrochenen Zugproben mit einem Rasterelektronenmikroskop gemacht wurden. Hierbei wurden jeweils Aufnahmen der Bruchstellen, sowie an unbeschädigten Stellen, Aufnahmen der obersten gedruckten Schicht und von der Seite in Druckrichtung (e_z -Richtung) gemacht. Die Aufnahmen offenbaren deutlich die Unterschiede zwischen dem FDM-Verfahren und der tropfenbasierten Realisierung im Rahmen dieser Arbeit.



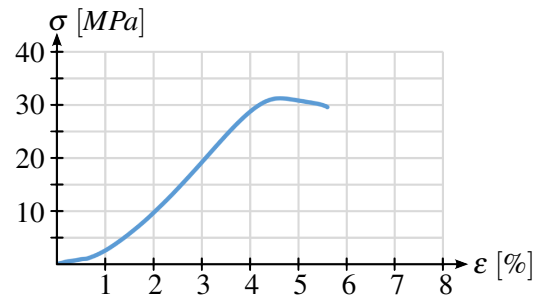
(a) Bauteil S23



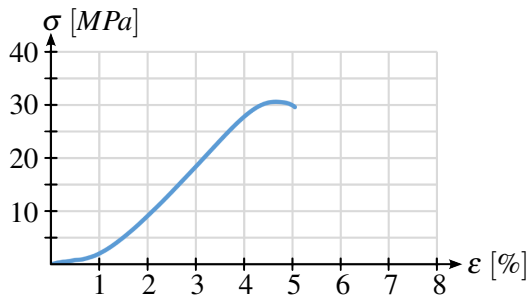
(b) Bauteil S24



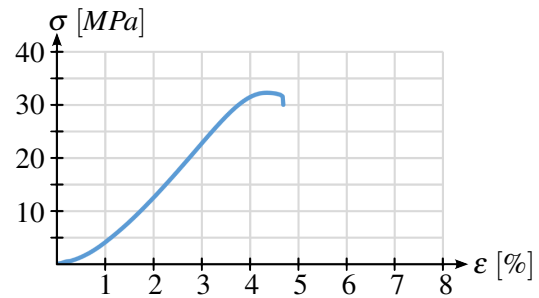
(c) Bauteil S25



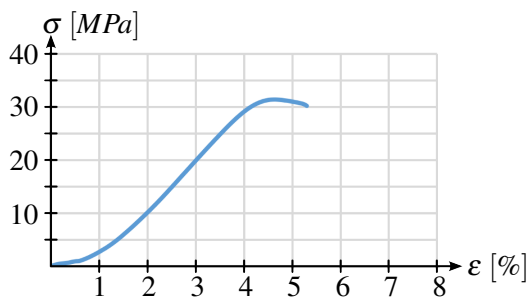
(d) Bauteil S27



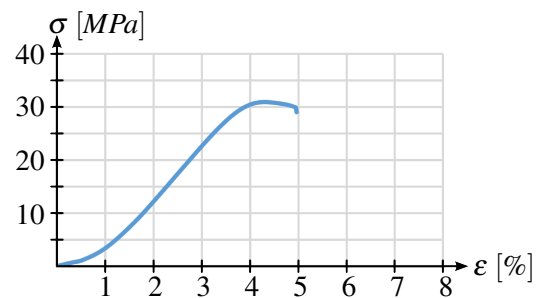
(e) Bauteil S28



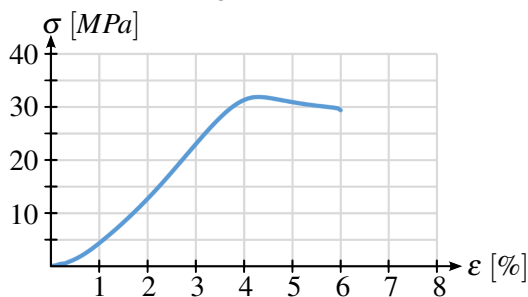
(f) Bauteil S29



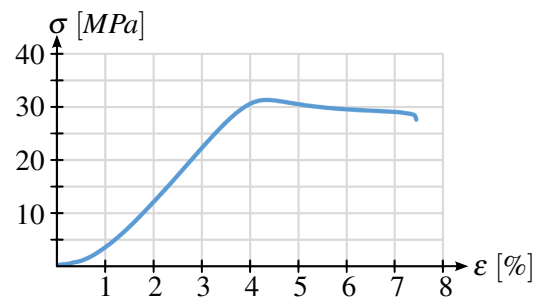
(g) Bauteil S30



(h) Bauteil S31

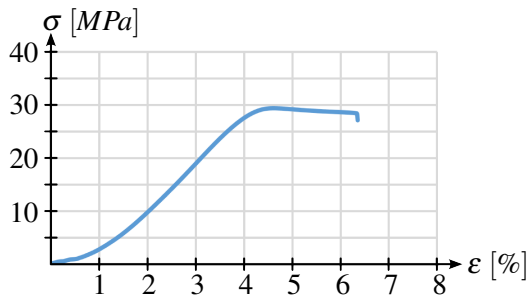


(i) Bauteil S32

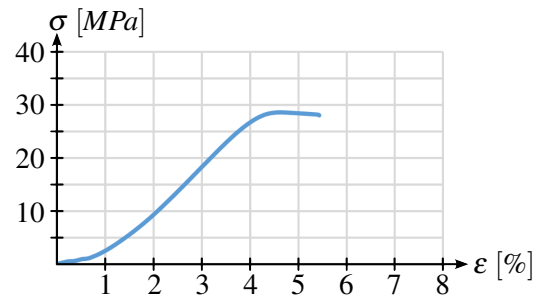


(j) Bauteil S33

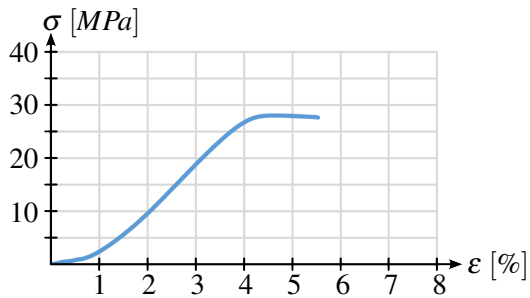
Abbildung 6.14: Spannungs- / Dehnungskurven für FDM-Zugstäbe nach der Norm DIN EN ISO 527-2 (2012) (Typ 1A), hergestellt mit der Stratasys Fortus 360mc (Schichthöhe: $H = 0.254\text{mm}$). Baumaterial: ABS-M30.



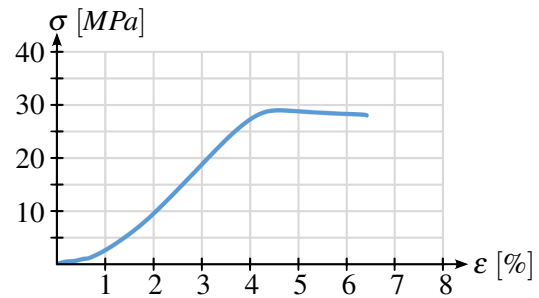
(a) Bauteil S34



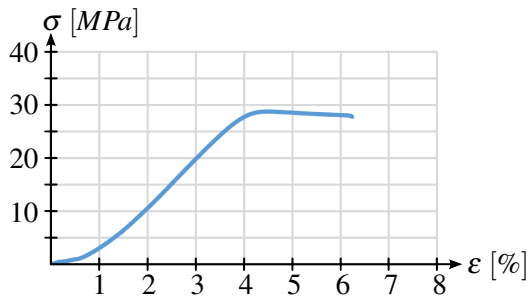
(b) Bauteil S35



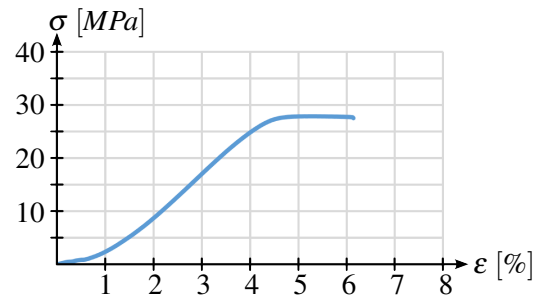
(c) Bauteil S36



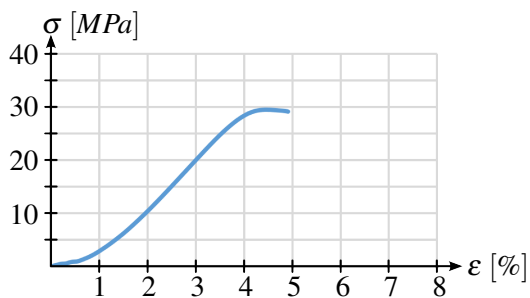
(d) Bauteil S37



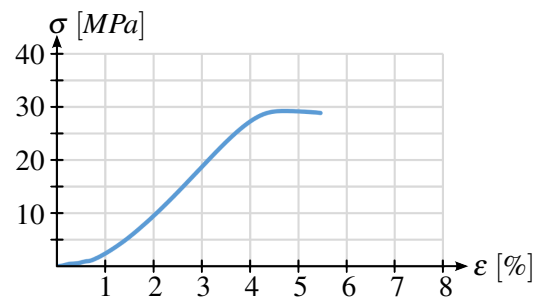
(e) Bauteil S38



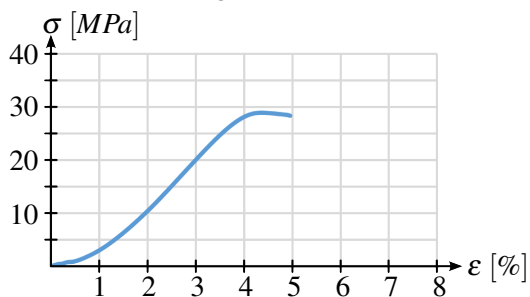
(f) Bauteil S39



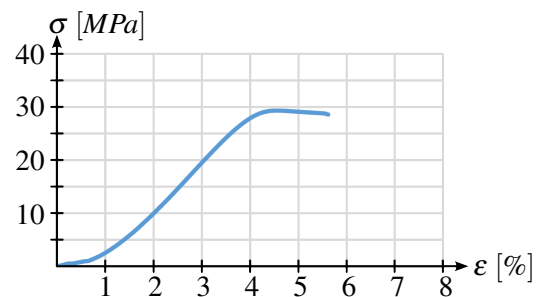
(g) Bauteil S40



(h) Bauteil S42

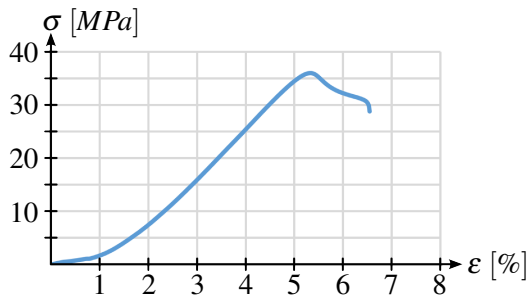


(i) Bauteil S43

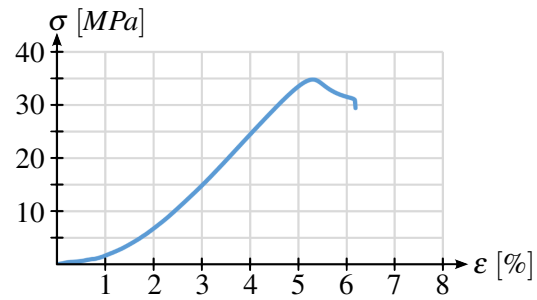


(j) Bauteil S44

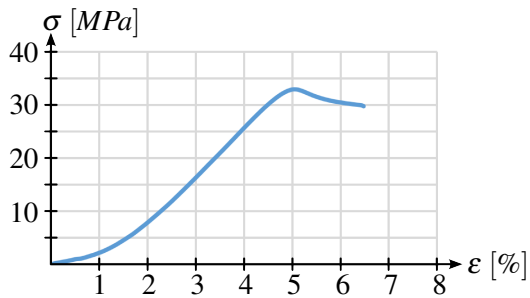
Abbildung 6.15: Spannungs- / Dehnungskurven für FDM-Zugstäbe nach der Norm DIN EN ISO 527-2 (2012) (Typ 1A), hergestellt mit der Stratasys Fortus 360mc (Schichthöhe: $H = 0.178\text{mm}$). Baumaterial: ABS-M30.



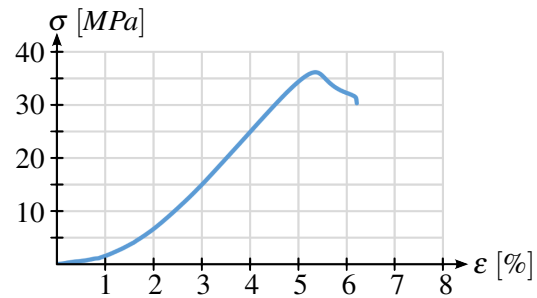
(a) Bauteil A12



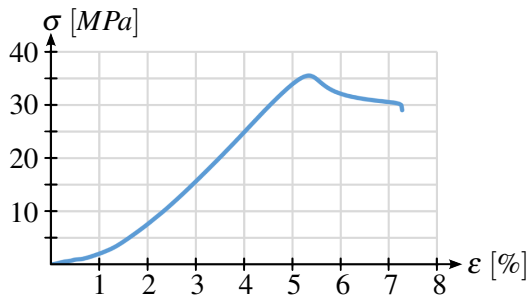
(b) Bauteil A13



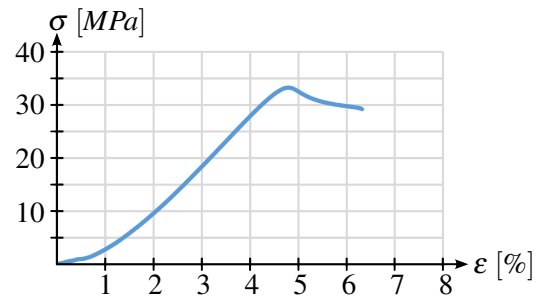
(c) Bauteil A14



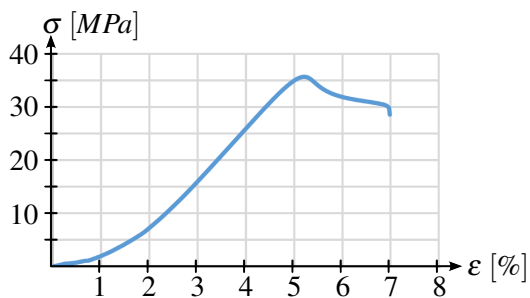
(d) Bauteil A15



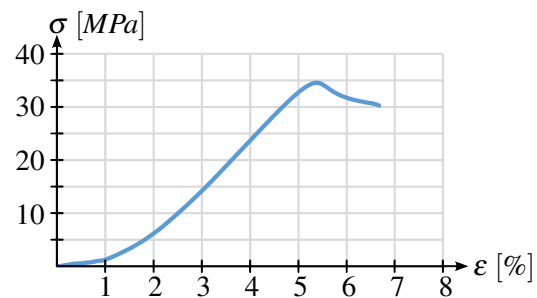
(e) Bauteil A16



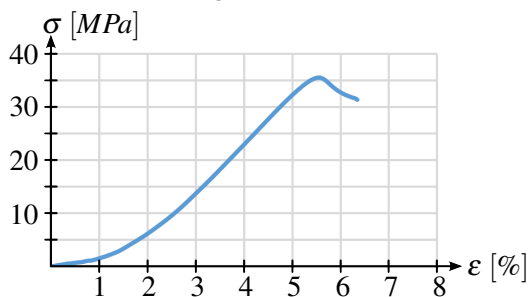
(f) Bauteil A17



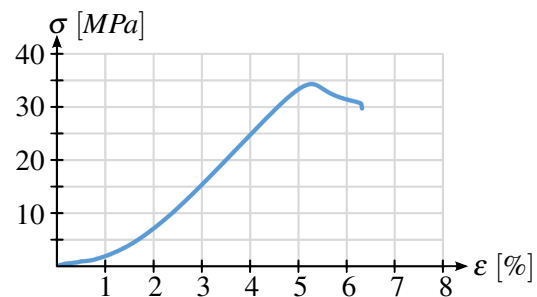
(g) Bauteil A18



(h) Bauteil A19

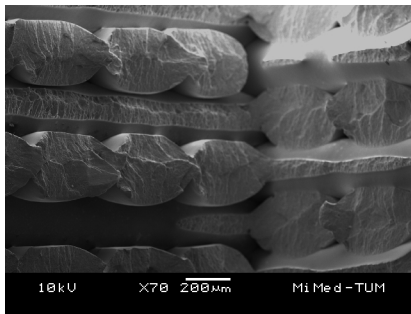


(i) Bauteil A21

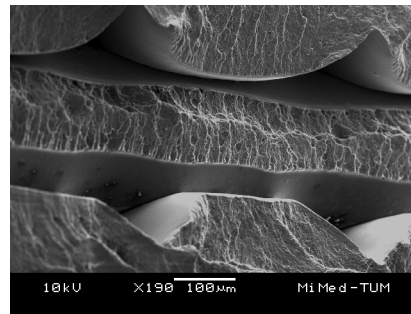


(j) Bauteil A22

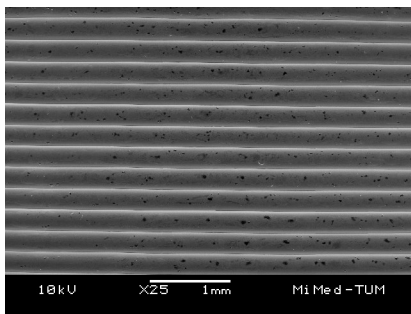
Abbildung 6.16: Spannungs- / Dehnungskurven für Zugstäbe nach der Norm DIN EN ISO 527-2 (2012) (Typ 1B), welche mit dem Tropfenerzeuger hergestellt wurden (Schichthöhe: $H = 0.265\text{ mm}$). Baumaterial: Terluran GP-35-ABS.



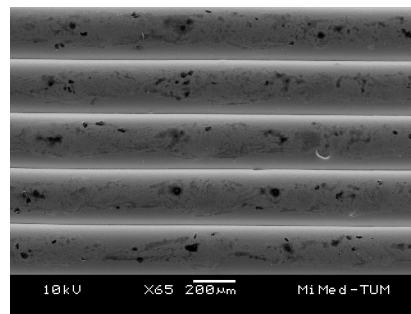
(a) Bruchstelle



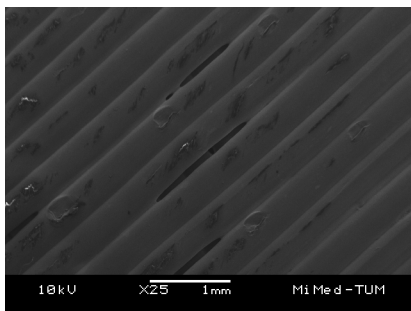
(b) Bruchstelle



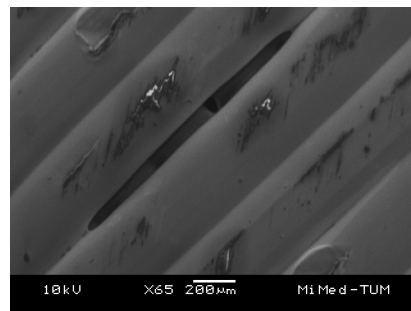
(c) von der Seite in Druckrichtung (e_z -Richtung)



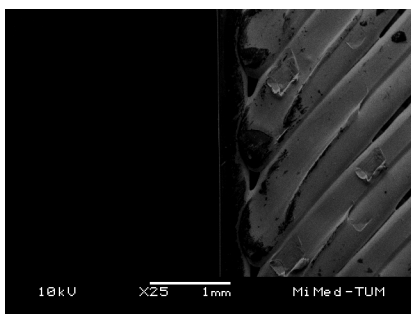
(d) von der Seite in Druckrichtung (e_z -Richtung)



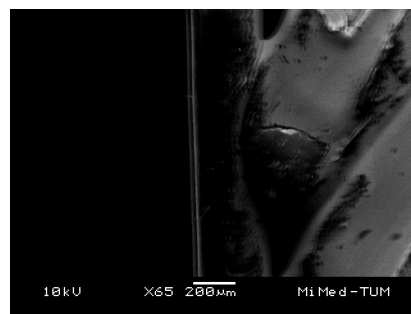
(e) oberste Schicht: deutlich erkennbare Lücken



(f) oberste Schicht: deutlich erkennbare Lücken

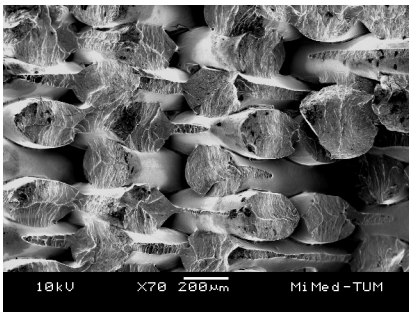


(g) oberste Schicht am Rand: Aufstauen von Material

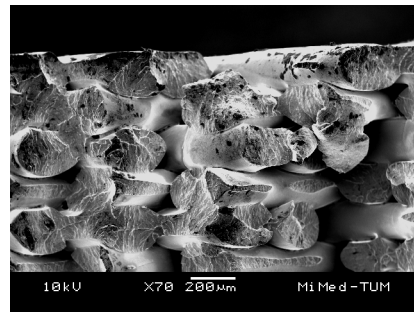


(h) oberste Schicht am Rand: Aufstauen von Material

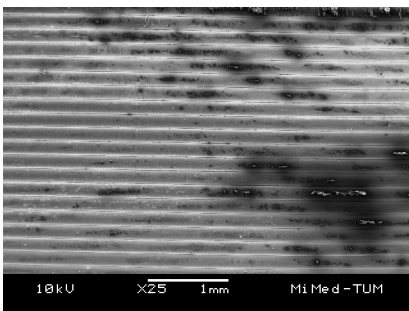
Abbildung 6.17: REM Aufnahmen von Zugstäben der Stratasys Fortus 360mc (Schichthöhe: $H = 0.254\text{ mm}$).



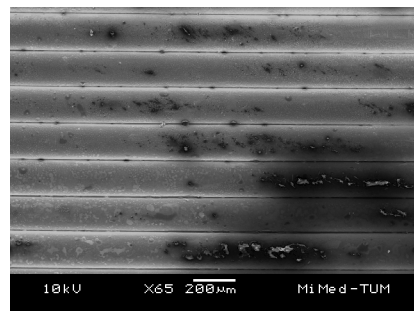
(a) Bruchstelle



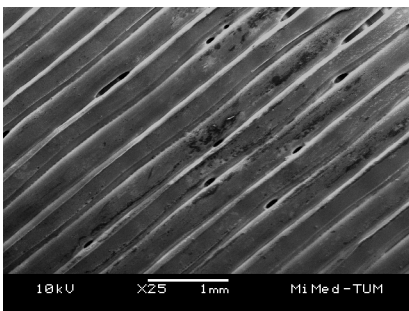
(b) Bruchstelle am Rand



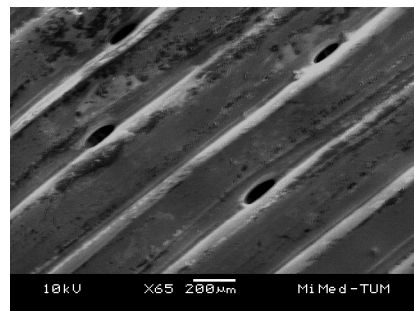
(c) von der Seite in Druckrichtung (e_z -Richtung)



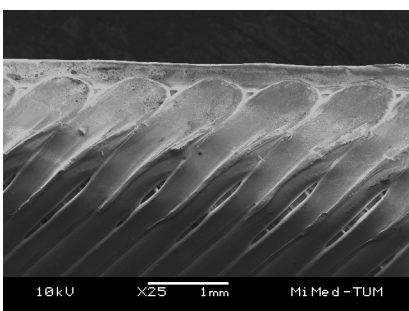
(d) von der Seite in Druckrichtung (e_z -Richtung)



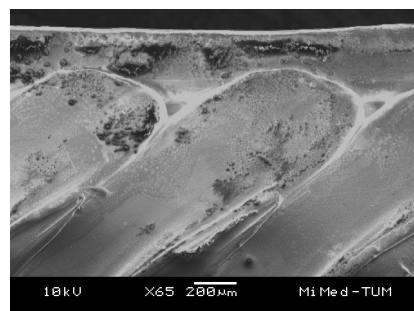
(e) oberste Schicht: deutlich erkennbare Lücken



(f) oberste Schicht: deutlich erkennbare Lücken

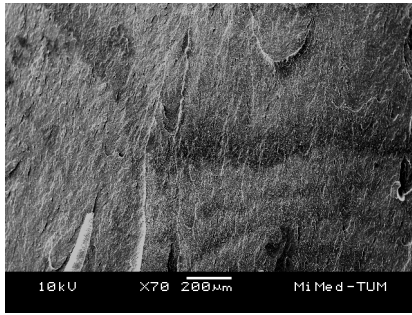


(g) oberste Schicht am Rand: Aufstauen von Material

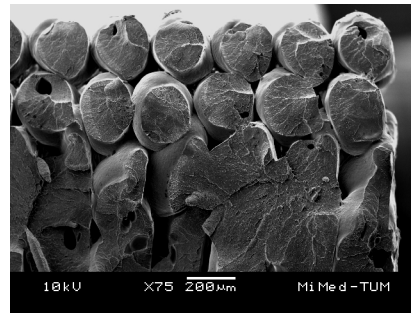


(h) oberste Schicht am Rand: Aufstauen von Material

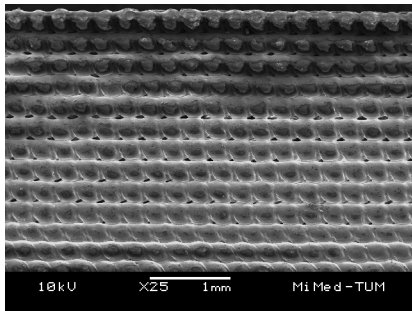
Abbildung 6.18: REM Aufnahmen von Zugstäben der Stratasys Fortus 360mc (Schichthöhe: $H = 0.178\text{ mm}$).



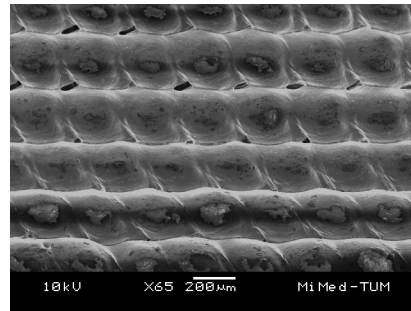
(a) homogene Bruchstelle: dichte Füllung, keine Tropfen erkennbar



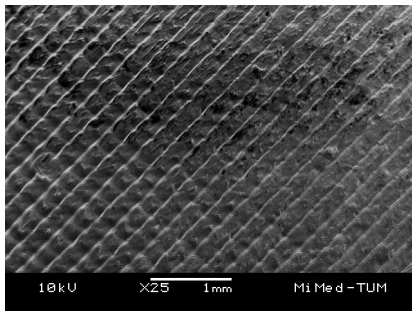
(b) Bruchstelle am Rand: durchtrennte Tropfen des Berandungsdrucks zu erkennen



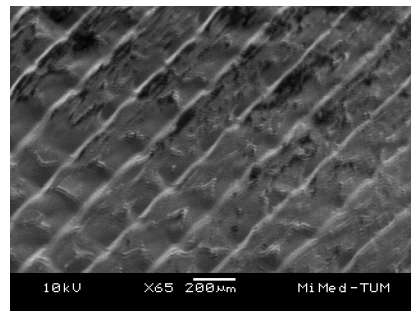
(c) von der Seite in Druckrichtung (e_z -Richtung)



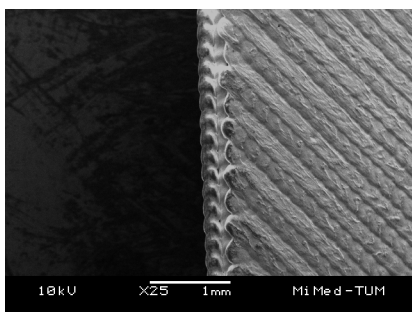
(d) von der Seite in Druckrichtung (e_z -Richtung)



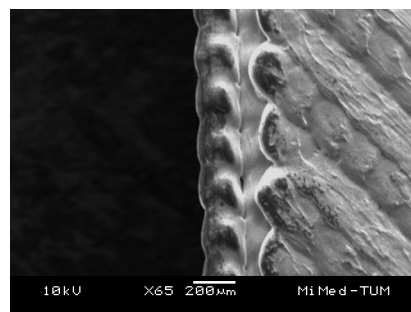
(e) oberste Schicht: homogene Füllung



(f) oberste Schicht: homogene Füllung



(g) oberste Schicht am Rand: kein Aufstauen von Material, da Leerfahrten zwischen Füllsegmenten möglich



(h) oberste Schicht am Rand: Füllung bindet sich mit innerer der beiden äußeren Konturen

Abbildung 6.19: REM Aufnahmen von Zugstäben des Tropfenerzeugers (Schichthöhe: $H = 0.265 \text{ mm}$).

7 Zusammenfassung und Ausblick

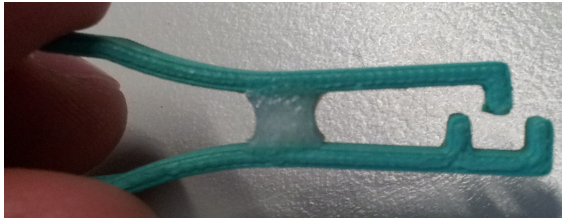
Die vorliegende Arbeit beschreibt die automatische Code-Generierung für einen neuen 3D-Druckprozess auf Tropfenbasis. Die Grundlage hierfür bildet auf der Hardware-Seite eine Maschine, die durch sequentielles Austragen von Tropfen aus Kunststoff in der Lage ist ein 3-dimensionales Bauteil Schicht für Schicht aufzubauen. Der Prozess ist daher den Rapid Prototyping Verfahren zuzuordnen und soll die Möglichkeit bieten schnell und kostengünstig Prototypen aus Kunststoff herzustellen, mit dem Ziel, diese auch als Funktionsteile verwenden zu können. Wesentlicher Bestandteil dieser Arbeit war dabei die Entwicklung einer prozessspezifischen Software, welche in der Lage ist aus CAD-Daten im STL-Format, automatisch Maschinencode (G-Code) zu generieren, der von der Maschinensteuerung ausgeführt werden kann um ein Bauteil zu drucken.

Das Hauptaugenmerk dieser Arbeit liegt in der algorithmischen Realisierung und mathematischen Darstellung aller implementierten Prozesse / Module vom Einlesen der STL-Daten bis hin zum Schreiben des fertigen Maschinencodes auf die Festplatte. Hierzu werden zunächst auf Basis mathematischer Definitionen alle für die Einzelprozesse relevanten mathematischen Grundlagen erläutert und anschließend deren Anwendung in den implementierten Modulen gezeigt. Der Gesamtprozess besteht dabei aus den Einzelprozessen „Importieren der STL-Daten“, „Slicen der importierten STL-Daten“, „Verschieben von Konturen für den Berandungsdruck“, „Füllen der Schichten“, „Optimierung der Daten für das Tropfenmodell“ und „Generieren von Maschinencode“. Alle Software-Module wurden in Microsoft Visual Studio 2010 in C++ programmiert. Die Integration aller Funktionalitäten zum Erzeugen von Maschinencode erfolgt über die definierte Schnittstelle einer DLL (Dynamic Link Library) in eine C#-Benutzeroberfläche, welche vom Entwickler des Druckprozesses zur Verfügung gestellt wird und Bestandteil des Gesamtsystems ist.

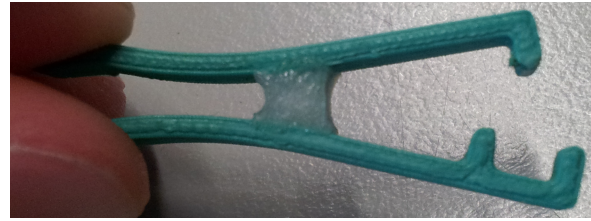
Mit Hilfe der im Rahmen dieser Arbeit entwickelten Software ist es möglich prozessspezifischen Maschinencode zu generieren und diesen zu drucken. Das Zusammenspiel zwischen Maschine, Druckprozess und erzeugtem NC-Programm ermöglicht es auf diese Weise Eigenschaften von Bauteilen während des Drucks positiv zu beeinflussen. Dies ist Grundlage dafür, gedruckte Prototypen auch als Funktionsbauteile verwenden zu können. Im Rahmen von Experimenten wurden daher für Probekörper die Oberflächenrauheit als optisches Kriterium und die Zugfestigkeit als mechanisches Kriterium bestimmt. Die Ergebnisse zeigen, dass die Oberflächenrauheit im Vergleich zum etablierten Fused Deposition Modeling Verfahren um teilweise 30 % verbessert werden konnte, wenn man die Mittenrauhwerte vergleicht. Auch bei der Zugfestigkeit der gedruckten Teile wurde mit 80 % des Zugfestigkeitswerts des verwendeten Kunststoffes ein beachtlicher Erfolg erzielt, so dass der Einsatz eines tropfenbasierten Verfahrens zur Herstellung von individuellen Funktionsbauteilen aus Kunststoff mehr als gerechtfertigt ist.

Trotz der erzielten Erfolge bezüglich der Festigkeit der gedruckten Teile und deren optischen Anmutung, gibt es einen sehr großen Spielraum für weitere Verbesserungen und Weiterentwicklungen. Diese betreffen unter anderem folgende Themen:

- Im Rahmen dieser Arbeit wurden die Druckdaten für jede einzelne Schicht getrennt optimiert. Es wurde kein Gedächtnis zwischen benachbarten Schichten entwickelt. Ist es



(a) Originalzustand der Klammer.



(b) Öffnen der Klammer.

Abbildung 7.1: Klammer bestehend aus einem biegbaren weichen Kunststoff und einem festen Kunststoff.

erst mal möglich in der Software gezielt Tropfen auf Lücke zu setzen bei benachbarten Schichten, so kann dies unnötige Lufteinschlüsse weiter minimieren und sich somit auf die Bauteilqualität / -festigkeit positiv auswirken. Hierbei müssen neben einer gezielten Entwicklung von neuen Software-Funktionalitäten auch Experimente durchgeführt werden, die weiteren Aufschluss darüber geben, wo Optimierungspotential besteht. Ein erster Schritt könnte beispielsweise die gezielte Detektion von Lufteinschlüssen sein.

- Der in dieser Arbeit verwendete Tropfenerzeuger ist nicht nur ein 3D-Drucker, sondern beinhaltet im Vergleich zum FDM-Verfahren auch eine Aufbereitungseinheit für verschiedenste Kunststoffe, wie dies auch bei den Spritzgießmaschinen der Fall ist. Durch die Kombinationen mehrerer solcher Aufbereitungseinheiten ist es möglich Kunststoffe mit verschiedenen Materialeigenschaften zu drucken. Abbildung 7.1 zeigt hierzu ein einfaches Beispiel einer gedruckten Hart-Weich-Kombination. Auf diese Weise können ganz neue Funktionsgruppen von Bauteilen gedruckt werden. Obwohl im Rahmen dieser Arbeit bereits ein erster automatischer Ansatz zum Druck solcher Bauteile entwickelt wurde, gibt es noch viel Arbeit um den beliebige Bauteile aus mehreren Materialien zu drucken. Auf der Software-Seite könnte dies unter anderem ein neues Datenformat sein, in dem jeder Komponente am Rechner bereits ein geeignetes Material zugewiesen wird.
- Die in dieser Arbeit entwickelte Software arbeitet zwar auf dem Tropfenmodell, allerdings geschieht dies schichtweise. Durch die Entwicklung neuer Baukonzepte kann es jedoch erforderlich werden, dass nicht nur Schichten sondern beispielsweise auch Sphären gedruckt werden können. Eine Herausforderung hierbei wäre die Entwicklung komplett neuer Baustrategien.

Literaturverzeichnis

- [1] Ahn, S.-H.; Montero, M.; Odell, D.; Roundy, S.; Wright, T. K. (2002): *Anisotropic Material Properties of Fused Deposition Modeling ABS*. Rapid Prototyping Journal, 8(4), pp. 248–257.
- [2] Ahn, D.; Kweon, J. H.; Kwon, S.; Song, J.; Lee, S. (2009): *Representation of Surface Roughness in Fused Deposition Modeling*. Journal of Materials Processing Technology, 209, pp. 5593–5600.
- [3] Anitha, R.; Arunachalam, S.; Radhakrishnan, P. (2001): *Critical Parameters Influencing the Quality of Prototypes in Fused Deposition Modelling*. Journal of Materials Processing Technology, 118, pp. 285–388.
- [4] Barth, F.; Mühlbauer, P.; Nikol, F.; Wörle, K. (1998): *Mathematische Formeln und Definitionen*. 7. überarbeitete Auflage, Bayerischer Schulbuch-Verlag, München, 1998.
- [5] Batchelder, J. S.; Comb, J.; Dahlin, T. J. (1999): *High Performance Rapid Prototyping System*. United States Patent US 5968561.
- [6] Bronstein, I. N.; Semendjajew, K. A.; Musiol, G.; Mühlig, H. (2008): *Taschenbuch der Mathematik*. 7. Auflage, Wissenschaftlicher Verlage Harri Deutsch GmbH, Frankfurt am Main, 2008.
- [7] Chang, C. C. (2004): *Direct Slicing and G-Code Contour for Rapid Prototyping Machine of UV Resin Spray Using Power SOLUTION Macro Commands*. The International Journal of Advanced Manufacturing Technology, 23(5-6), pp. 358 – 365.
- [8] Chen, X.; McMains, S. (2005): *Polygon Offsetting by Computing Winding Numbers*. In Proceedings of ASME 2005 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference, pp. 1 – 11.
- [9] Comb, J.; Dockter S. E.; Berens, P. A. (1999): *Rapid Prototyping Apparatus*. United States Patent US 5939008.
- [10] Crump, S. S. (1992): *Apparatus and Method for Creating Three-Dimensional Objects*. United States Patent US 5121329 A.
- [11] Dahlin, T. J.; Nelson, B. C.; Docken, K. D. (2000): *Rapid Prototyping System with Filament Supply Spool Monitoring*. United States Patent US 6022207.
- [12] Dhanik, S.; Xirouchakis, P. (2010): *Contour Parallel Milling Tool Path Generation for Arbitrary Pocket Shape Using a Fast Marching Method*. International Journal of Advanced Manufacturing Technology, 50, pp. 1101 – 1111.
- [13] Erhard, G. (2008): *Konstruieren mit Kunststoffen*, 4. Auflage, Carl Hanser Verlag, München.

- [14] Ehrenstein, G. W. (2007): *Mit Kunststoffen konstruieren*, 3. Auflage, Carl Hanser Verlag, München.
- [15] Hehl, K. (2010): *Vorrichtung zur Herstellung eines dreidimensionalen Gegenstandes*. German Patent DE 10 2009 030 099 B4.
- [16] Hull, C. W. (1986): *Apparatus for Production of Three-Dimensional Objects by Stereolithography*. United States Patent US 4575330.
- [17] Jibin, Z.; Weijun, L.; Wang, Y. (2004): *Study on Algorithm of Contours Path Generation for Robotic Prototyping*. In Proceedings of the IEEE International Conference on Robotics and Biomimetics, pp. 181 – 186.
- [18] Johannaber, F.; Michaeli, W. (2001): *Handbuch Spritzgießen*, 1. Auflage, Carl Hanser Verlag, München, Wien.
- [19] Lai, W.; Faddis, T.; Sorem, R. (2000) *Incremental Algorithms for Finding the Offset Distance and Minimum Passage Width in a Pocket Machining Tool Path Using the Voronoi Technique*. Journal of Materials Processing Technology, 100(1–3), pp. 30 – 35.
- [20] Lombardi, J. L.; Popovich, D.; Artz, G. J. (2000): *Water Soluble Rapid Prototyping Support and Mold Material*. United States Patent US 6070107.
- [21] Ma, W.; He, P. (1999): *An Adaptive Slicing and Selective Hatching Strategy for Layered Manufacturing*. Journal of Materials Processing Technology, 89–90, pp. 191 – 197.
- [22] Norm ASME B46.1 (2009): *Oberflächentextur (Oberflächenrauheit, -welligkeit und -struktur)*. Beuth Verlag, Berlin.
- [23] Norm DIN EN ISO 527-1 (2012): *Kunststoffe – Bestimmung der Zugeigenschaften – Teil 1: Allgemeine Grundsätze*. Beuth Verlag, Berlin.
- [24] Norm DIN EN ISO 527-2 (2012): *Kunststoffe – Bestimmung der Zugeigenschaften – Teil 2: Prüfbedingungen für Form- und Extrusionsmassen*. Beuth Verlag, Berlin.
- [25] Norm DIN EN ISO 3274 (1998): *Geometrische Produktspezifikationen (GPS) – Oberflächenbeschaffenheit: Tastschnittverfahren – Nenneigenschaften von Tastschnittgeräten*. Beuth Verlag, Berlin.
- [26] Norm DIN EN ISO 4287 (2010): *Geometrische Produktspezifikation (GPS) – Oberflächenbeschaffenheit: Tastschnittverfahren – Benennungen, Definitionen und Kenngrößen der Oberflächenbeschaffenheit*. Beuth Verlag, Berlin.
- [27] Norm DIN EN ISO 4288 (1998): *Geometrische Produktspezifikation (GPS) – Oberflächenbeschaffenheit: Tastschnittverfahren – Regeln und Verfahren für die Beurteilung der Oberflächenbeschaffenheit*. Beuth Verlag, Berlin.
- [28] Norm DIN 50125 (2009): *Prüfung metallischer Werkstoffe – Zugproben*. Beuth Verlag, Berlin.
- [29] Pandey, P. M.; Reddy, N. V.; Dhande, S. G. (2003a): *Slicing Procedures in Layered Manufacturing: a Review*. Rapid Prototyping Journal, 9(5), pp. 274 – 288.

- [30] Pandey, P. M.; Reddy, N. V.; Dhande, S. G. (2003b): *Real Time Adaptive Slicing for Fused Deposition Modelling*. International Journal of Machine Tools and Manufacture, 43(1), pp. 61 – 71.
- [31] Pei, E.; Campbell, R. I.; de Beer, D. (2011): *Entry-Level RP Machines: How Well Can They Cope with Geometric Complexity?*. Assembly Automation, 31(2), pp. 153–160.
- [32] Schwaiger, J.; Lueth, T. C.; Irlinger, F. (2013): *G-Code Generation for a New Printing Process Based on 3D Plastic Polymer Droplet Generation*. In Proc. of the ASME International Mechanical Engineering Congress & Exposition 2013, San Diego, California, USA, November 15-21.
- [33] Seo, M.; Kim, H.; Onosato, M. (2005): *Systematic Approach to Contour-Parallel Tool Path Generation of 2.5D Pocket with Islands*. Computer-Aided Design & Applications, 2(1-4), pp. 213 – 222.
- [34] Tata, K.; Fadel, G.; Bagchi, A.; Aziz, N. (1998): *Efficient Slicing for Layered Manufacturing*. Rapid Prototyping Journal, 4(4), pp. 151 – 167.
- [35] 3D Systems Inc. (1989): *Stereolithography Interface Specification*.
- [36] Vatani, M.; Rahimi, R.; Brazandeh, F.; Nezhad, A. S. (2009): *An Enhanced Slicing Algorithm Using Nearest Distance Analysis for Layer Manufacturing*. World Academy of Science, Engineering and Technology, 49, pp. 721 – 726.
- [37] Zäh, M. F. (2006): *Wirtschaftliche Fertigung mit Rapid-Technologien*. Carl Hanser Verlag, München, Wien.
- [38] Zhao, Q.; Huang, X.; Wang, C.; Yao, Y.; Hu, Q. (2011): *NC Tool Path Generation Approach Based on STL*. In International Symposium on IT in Medicine and Education, pp. 228 – 231.