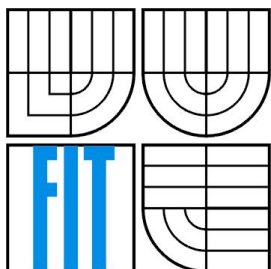




VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ
FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

GENEROVÁNÍ DRÁHY NÁSTROJE PŘI 3 OSÉM FRÉZOVÁNÍ 3D POLYGONÁLNÍCH MODELŮ

TOOL PATH GENERATION FOR 3 AXES POLYGONAL MODELS MILLING

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

Jan Nedvědický

VEDOUCÍ PRÁCE
SUPERVISOR

doc. Ing. Přemysl Kršek, Ph.D.

BRNO 2012

Abstrakt

Tato bakalářská práce se věnuje problematice generování dráhy pro 3 osé frézování na základě polygonálních modelů. Úvodní část se zabývá rozбором základních pojmů a témat souvisejících s řešenou úlohou. Hlavní část práce řeší postupy a problémy při implementaci jednoduchého generátoru dráhy nástroje. Probrány jsou jednoduché strategie frézování založené na pohybu nástroje v rovině. Součástí práce je implementace zásuvného modulu aplikace OpenFlipper, ve kterém jsou použity a prakticky ověřeny popsání postupy.

Abstract

This bachelor's thesis deals with tool path generation for 3-axis milling based on 3D polygonal models. First part introduces basic terms and related topics. Main part is devoted to description of techniques and problems related to implementation of simple tool path generator. Simple milling strategies based on tool movement restricted in plane are discussed. This thesis includes implementation of plugin for OpenFlipper application where described techniques are used and practically verified.

Klíčová slova

CNC frézování, dráha nástroje, frézovací strategie, polygonální model

Keywords

CNC milling, tool path, milling strategy, polygonal model

Citace

Jan Nedvědický: Generování dráhy nástroje při 3 osém frézování 3D polygonálních modelů, bakalářská práce, Brno, FIT VUT v Brně, 2012

Generování dráhy nástroje při 3 osém frézování 3D polygonálních modelů

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením doc. Ing. Přemysla Krška, Ph.D. a v seznamu literatury uvedl všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
Jan Nedvědický
14.5.2012

Poděkování

Rád bych tímto poděkoval vedoucímu své bakalářské práce doc. Ing. Přemyslu Krškovi, Ph.D. za vedení práce a cenné náměty a podporu při její tvorbě.

© Jan Nedvědický 2012

Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.

Obsah

Obsah.....	1
1 Úvod.....	2
2 Rozbor.....	3
2.1 Obrábění.....	3
2.2 Numerické řízení.....	6
2.3 Strategie CNC Frézování.....	6
2.4 Specifikace zadání.....	9
3 Návrh.....	11
3.1 Paralelní strategie frézování.....	11
3.2 Horizontální strategie frézování.....	21
4 Implementace.....	33
4.1 OpenFlipper, OpenMesh.....	33
4.2 CGAL - Computational Geometry Algorithms Library.....	34
4.3 Struktura zdrojových kódů.....	34
5 Výsledky.....	36
5.1 Paralelní strategie.....	36
5.2 Horizontální strategie.....	38
6 Závěr.....	40
Literatura.....	41
Seznam příloh.....	42
Příloha 1. Ukázky vygenerovaných drah.....	43
Příloha 2. Návod k přiložené aplikaci.....	46
Příloha 3. Nastavení prostředí pro překlad.....	48

1 Úvod

Rozvoj výpočetní techniky se v poslední době výrazně promítá také ve strojírenství. Počítačem řízené stroje (CNC – computer numeric control) se stávají běžnou součástí výrobních hal. Dříve přesnost, kvalita a rychlost výroby závisela na zručnosti a zkušenostech obráběče. Dnes však hraje určující roli program, který obrábění řídí. S narůstající složitostí tvarů vyráběných součástí roste také složitost CNC programů. V určitém bodě již ani není v lidských silách ve formě programu popsat dostatečně přesně tvar součástí. V této chvíli přichází na řadu specializovaný software CAM (Computer Aided Manufacturing), který na základě počítačového modelu součásti vygeneruje odpovídající program pro CNC stroj.

Tato práce se zabývá rozbořem možností generování dráhy nástroje pro tříosé CNC frézování a implementací jednoduchého generátoru dráhy na základě polygonálního modelu. Generování optimální cesty nástroje je netriviální úkol, protože se musí vzít v úvahu množství různých faktorů ovlivňujících přesnost a rychlost obrábění a také životnost obráběcího nástroje. Profesionální CAM software je proto tvořen rozsáhlými a komplikovanými systémy, které nabízejí velké množství specializovaných frézovacích strategií. Je zřejmé, že v rámci této práce není možné vytvořit systém porovnatelný s průmyslově používaným softwarem. Cílem práce je především vyřešit základní problémy při implementaci generátoru dráhy pro frézování. Implementovat jednoduché frézovací strategie a umožnit jejich případné přizpůsobení konkrétním účelům.

V kapitole 2 jsou rozebrány základní pojmy problematiky frézování a obrábění obecně. Kapitola 3 řeší jednotlivé problémy při generování dráhy podle vybraných frézovacích strategií. Kapitola 4 se zabývá implementací generátoru dráhy pro frézování. Obsahuje popis knihoven použitých při implementaci a struktury zdrojových kódů. V kapitole 5 jsou probrány výsledky dosažené při generování dráhy na základě různých modelů a jsou popsány vlastnosti implementace uvedených postupů. Závěrečná 6. kapitola se zabývá zhodnocením dosažených výsledků, návrhy pro vylepšení používaných postupů a možnými směry dalšího pokračování projektu.

2 Rozbor

Problematika CNC frézování i obrábění samotného je rozsáhlé téma, jehož důkladnější rozbor spadá mimo rozsah této práce. Uvedeme zde proto jen několik základních pojmů, které souvisí s řešeným tématem. Soustředíme se především na frézování a přístupy k jeho numerického řízení.

2.1 Obrábění

Obrábění chápeme jako proces, jehož činností vznikne z polotovaru požadovaný výrobek. Typicky se jedná o odebrání přebytečného materiálu, tak abychom dosáhli požadovaného tvaru. Pojmem polotovar rozumíme objekt před začátkem obrábění. Objekt, který se nachází v procesu obrábění je označován jako obrobek. Dokončením obrábění vzniká z obrobku konečný výrobek.

Obrábění

- třískové
 - rotační
 - frézování
 - soustružení
 - vrtání
 - posuvné
 - hoblování
 - obrážení
- netřískové
 - laser
 - vodní paprsek
 - plasma
 - další méně rozšířené

Pravděpodobně nejrozšířenějším druhem obrábění je obrábění třískové, kde bříty nástroje postupně odřezávají z obrobku malé množství materiálu (třísky). V souvislosti s rozvojem techniky se však stále častěji můžeme setkat také s dalšími druhy obrábění, kdy se k opracování obrobku používá například laserový paprsek nebo proud vody o vysokém tlaku.

Vynecháme třískové obrábění kde se využívá vratného přímého pohybu (hoblování a obrážení) a soustředíme se na třískové obrábění, kde je hlavní řezný pohyb představován rotací. Podle vzájemných pohybů obrobku a nástroje rozlišujeme rotační obrábění na vrtání, soustružení a frézování. [1] [2]

2.1.1 Vrtání

Vrtání se používá pro tvorbu děr v plném materiálu. Hlavní rotační řezný pohyb zde vykonává nástroj (vrták). Vedlejší pohyb je omezen ve směru osy nástroje. Při vrtání se totiž samotného procesu oddělování materiálu účastní pouze spodní hrana vrtáku. Vrtáky jsou typicky šroubovitého tvaru, který zajišťuje hladký odvod třísek ven z díry.

2.1.2 Soustružení

Soustružením vznikají výrobky rotačních tvarů. Obrobek tvoří nejčastěji tyčový materiál, který je uchycen v rotujícím sklíčidle. Hlavní řezný pohyb je tedy vykonáván obrobkem. Požadovaného tvaru se dosahuje zanořováním nástroje (břitu) do rotujícího obrobku. Nástroj je typicky omezen v pohybu ve vodorovné rovině, přičemž je jeho posuv řízen nezávisle v podélném a příčném směru. Pohyb nástroje je typicky zajišťován pomocí šroubů s definovaným stoupáním. Otočením o daný počet otáček je tak možné dosáhnout přesné délky posuvu. [1] [2]



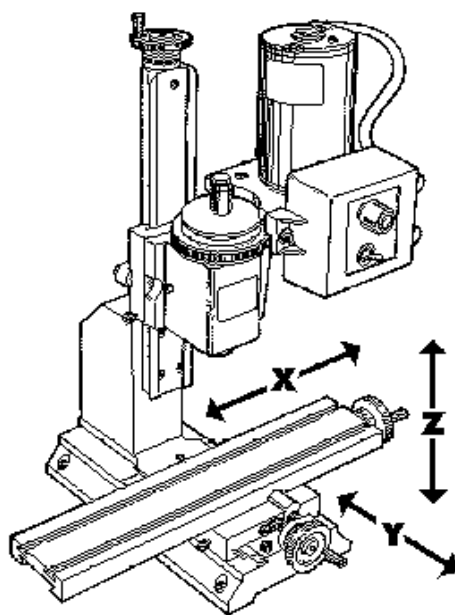
Obrázek 1: Soustružení. Zdroj: <http://www.frontierpolymer.com/pic/Lathe.jpg>.

2.1.3 Frézování

Při frézování je polotovár obráběn rotujícím nástrojem, který se podle daných pravidel pohybuje vzhledem k obrobku. Podle konstrukce stroje se toho nejčastěji dosahuje pevným uchycením obrobku, přičemž veškerý pohyb obstarává vřeteno s nástrojem. Výjimkou však nejsou ani stroje, kde dochází ke kombinaci pohybu pracovního stolu s uchyceným obrobkem a vřetene. V tomto případě je

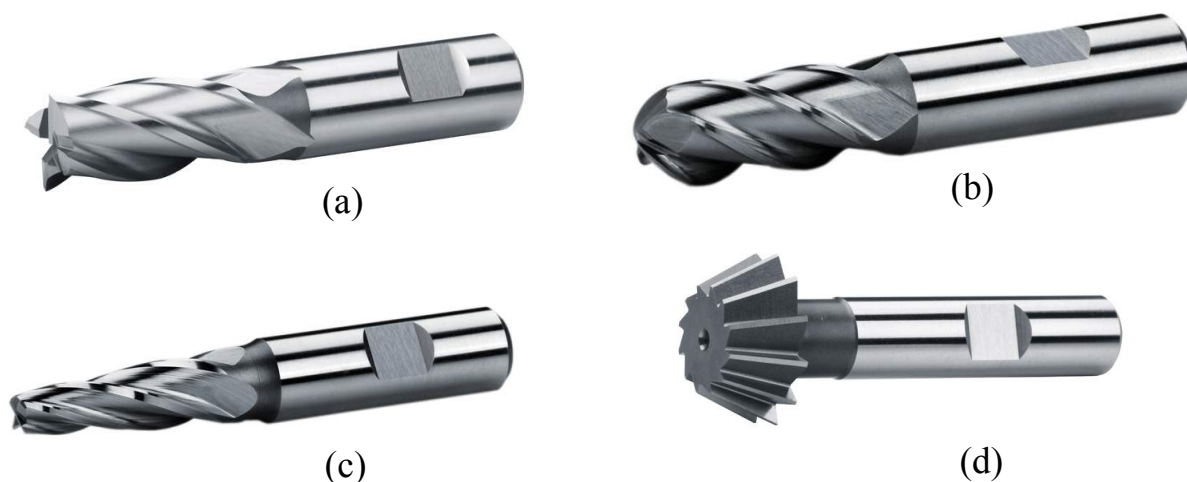
typicky pohyb nástroje omezen do dvou os a pohybem pracovního stolu se dosahuje posunu podél třetí osy.

Podle počtu stupňů volnosti nástroje vzhledem k obrobku rozlišujeme tříosé až pětiosé frézování. Zbývající šestý stupeň volnosti představuje hlavní řezný pohyb – rotace nástroje kolem osy. Nejjednodušší je tříosé frézování, kde dochází pouze k posuvnému pohybu nástroje podél tří os v prostoru (X, Y a Z). Vzhledem k tomu, že je v tomto případě stálá orientace nástroje (vektor osy nástroje se nemění), je však také značně omezen možný tvar výrobku. Možným rozšířením tohoto přístupu je upnutí obrobku do vřetena jako v případě soustružení. Řízením natočení obrobku pak může nástroj pohybující se ve třech osách vytvořit složitější tvary. Častěji se však v případě většího počtu řízených os setkáváme s frézováním pětiosým. Základ je stejný jako u tříosého frézování. Vřeteno je zde však doplněno o mechanismus umožňující naklápění nástroje podél dalších dvou os. Tím je dosaženo větší volnosti při návrhu tvaru součástí. Porovnejme možnosti frézování podle počtu řízených os na polotvaru ve tvaru krychle, do jejíchž stran chceme vyhloubit kapsy. Tři osy nám umožní opracovat pouze horní stranu krychle. Uchycením krychle do řízeného vřetene získáme možnost opracovat 4 strany krychle. Pětiosé frézování nám umožní opracovat celou krychli kromě spodní strany, kterou je uchycena k pracovnímu stolu stroje.



Obrázek 2: Schéma jednoduché tříosé frézky.
Zdroj: <http://www.sherline.com/images/millfig2.gif>

Pro frézování se podle podle konkrétní situace používají různé typy nástrojů. Liší se tvarem velikostí i počtem břitů. Pokud uvedeme nejčastější rozdělení podle geometrického tvaru jsou typicky používány frézy válcové a kulové. Pro zvláštní použití pak existují také například frézy kotoučové a frézy speciálních tvarů. [1] [2]



Obrázek 3: Válcová (a), kulová (b), kuželová (c) a úhlová (d) fréza.

Zdroj: Katalog společnosti ZPS-FRÉZOVACÍ NÁSTROJE, <http://www.zps-fn.cz/katalog-frez/>.

2.2 Numerické řízení

Numerické řízení obráběcích strojů umožňuje ve větší míře automatizovat výrobu a dosáhnout výrazně lepší opakované přesnosti v porovnání s ručním obráběním. Využívá se u obou nejčastějších typů rotačního třískového obrábění – soustružení i frézování. V případě potřeby je možné stroj určený pro CNC frézování použít s odpovídajícím nástrojem také pro vrtání.

Počítačového řízení obrábění je dosaženo použitím takových posuvů, u kterých je možné elektronicky definovat vzdálenost pohybu. Nejjednodušší posuv je realizován skrze šroub, kterým otáčí krokový motor. Krokový motor se otáčí postupně po diskrétních krocích. Počet kroků spolu se stoupáním šroubu pak udává uraženou vzdálenost. Existují samozřejmě další možnosti posuvu. Z pohledu programátora stroje však mezi nimi není rozdíl.

2.3 Strategie CNC Frézování

Základním problémem řešeným při generování dráhy nástroje pro CNC frézování je určit dráhu daného bodu nástroje (typicky jeho špičky) tak, aby se nástroj samotný dotýkal povrchu modelu, ale

nikdy se pod něj nezanořil. Protože budeme pracovat ve třech osách, musíme se také zabývat vhodným uspořádáním takovéto trajektorie v prostoru.

Strategie frézování určuje principy a pravidla, podle kterých je vytvořena trajektorie nástroje v prostoru tak, aby byly splněny požadavky kladené na danou strategii. Společnosti zabývající se vývojem CAM softwaru se často snaží odlišit od konkurence podporou množství speciálních strategií. Zde proto rozebereme jen několik strategií, které podporuje naprostá většina dostupného softwaru.

Z hlediska úlohy lze strategie frézování typicky rozdělit do dvou skupin. První skupinou jsou hrubovací strategie a druhou jsou dokončovací strategie. Případnou třetí skupiny by mohly tvořit strategie se speciálním určením. Toto rozdělení však není striktní, konkrétní strategie lze často použít pro hrubování i dokončování přímo nebo jen s nepatrnými úpravami.

Hrubování

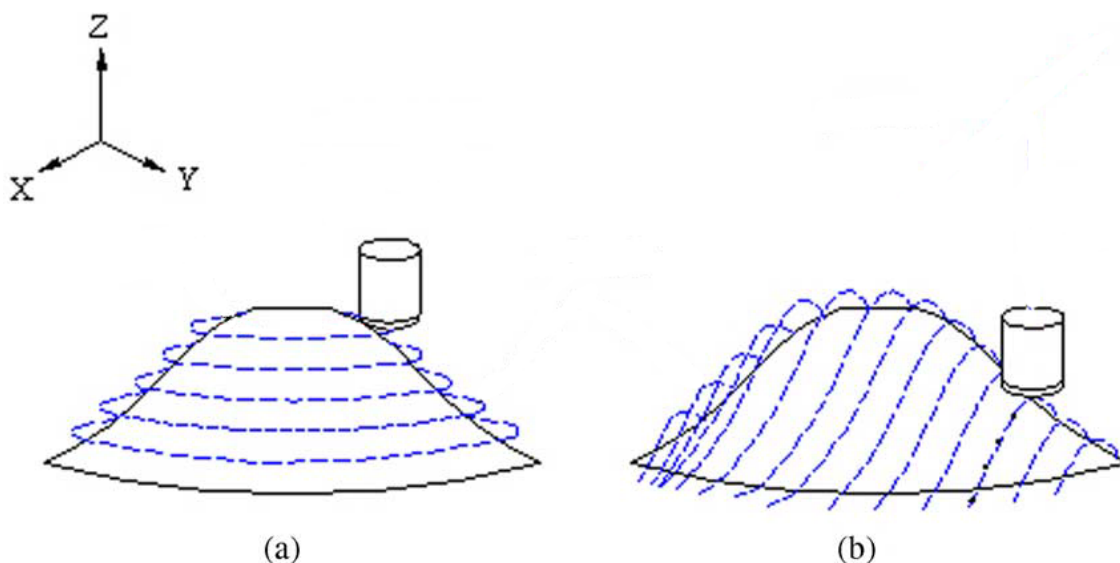
Hrubování předchází fázi dokončování. Cílem hrubování je efektivně odebrat co největší množství přebytečného materiálu. Používají se tedy typicky nástroje většího průměru. Vstupem je surový materiál a výsledkem je hrubý tvar frézovaného objektu. Je však třeba zajistit, aby veškerý zbývající přebytečný materiál bylo možné odebrat ve fázi dokončování. Uživatel tedy musí kvalifikovaně rozhodnout, jaké množství přebytečného materiálu může na výrobku po hrubování zůstat a podle toho určit parametry hrubování, případně hrubování rozdělit do více průchodů.

Dokončování

Cílem fáze dokončování je vyhladit tvar vzniklý hrubováním a vytvořit konečný povrch výrobku. Oproti hrubování se typicky používají menší nástroje, které umožňují opracovat výrobek do větších detailů.

2.3.1 Strategie založené na rovnoběžných rovinách

Nejjednodušší strategie frézování generují dráhu v prostoru postupně po rovnoběžných rovinách, které jsou podél modelu rozmístěny s konstantním krokem. Podle orientace těchto rovin vzhledem k orientaci nástroje budeme rozlišovat paralelní a horizontální strategie. V základu se jedná o strategie dokončovací. Od dráhy paralelního i horizontálního dokončování však lze odvodit také trajektorii nástroje pro odpovídající hrubovací strategii.



Obrázek 4: Horizontální (a) a paralelní (b) frézovací strategie. Zdroj [3].

Horizontální strategie

Horizontální frézovací strategie (obr. 4.a), někdy také zvaná strategie s konstantní hladinou Z, vychází z dráhy nástroje podél modelu, přičemž je pohyb nástroje omezen do horizontální roviny. Pokud budeme uvažovat uzavřené modely bude trajektorie nástroje v horizontální rovině vždy tvořit uzavřenou křivku. Výslednou dráhu získáme vhodným propojením drah po sobě jdoucích rovin.

Paralelní strategie

Paralelní strategie (obr. 4.b), někdy též nazývána rastrová nebo „cik-cak“ strategie, je založena na generování dráhy v rovinách rovnoběžných s osou nástroje (\rightarrow paralelní). Typicky se použijí roviny rovnoběžné s rovinou XZ nebo YZ. Dráha nástroje v takovéto rovině je otevřená, začíná vždy na jednom konci modelu a končí na druhém. Výslednou dráhu je pak vhodné skládat tak, aby docházelo k pohybu v jedné rovině jedním směrem a v sousední rovině opačným směrem. Jednoduchou modifikací můžeme získat strategii hrubovací. V závislosti na délce břitu frézy omezíme maximální hloubku zanoření a provedeme vygenerování dráhy. V dalším kroku povolíme zanoření vyšší a znovu vygenerujeme dráhu. Tímto způsobem postupně skládáme jednotlivé dráhy za sebe až dojdeme na nejnižší úroveň modelu. Tím, že v každém kroku omezíme hloubku zanoření omezíme množství odebíraného materiálu v jednom průchodu a tím také zabráníme případnému poškození nástroje.

2.3.2 Další často používané strategie

Odvrtávání

Odvrtávání je typicky hrubovací strategie. K odebrání materiálu v tomto případě dochází pouze při pohybu podél osy Z. K přesunům podél os X a Y dochází vždy mimo kontakt nástroje s obrobkem. Výhodou je, že v tomto případě působí síly ve směru osy nástroje, který nejlépe odolává právě takovému namáhání.

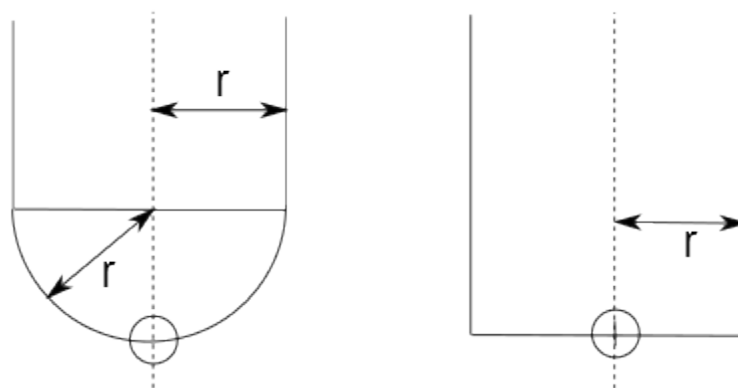
Spirálové frézování

Výstupem této strategie je dráha podél modelu tvořená jedinou spirálou. Výhodou je zde plynulé zanořování frézy. Jedná se o dokončovací strategii.

2.4 Specifikace zadání

V této práci se zaměříme na generování dráhy pro tříosé CNC frézování pomocí kulových a válcových nástrojů. Vstupem bude polygonální model ve formátu STL a požadovaným výsledkem bude dráha nástroje popsána G-kódem ve formátu kompatibilním se systémem LinuxCNC [4]. Cílem je implementace základních strategií frézování – paralelní a horizontální dokončování a od nich odvozené hrubování.

Dále bude v dalším popisu předpokládána následující orientaci os stroje. Pracovní stůl je tvořen rovinou XY a osa nástroje je rovnoběžná s osou Z. Osa Z je orientována tak, že s klesáním nástroje se hodnota souřadnice Z snižuje. V úvahu budeme brát pouze nejjednodušší tvary nástrojů, které jsou znázorněny na obrázku 5. Případný různý průměr stopky a břitů nebudeme uvažovat. Na obrázku je vyznačena také špička nástrojů. Polohu nástroje v prostoru budeme vždy určovat souřadnicemi tohoto vztažného bodu.



Obrázek 5: Uvažované tvary nástrojů a polohy vztažných bodů.

3 Návrh

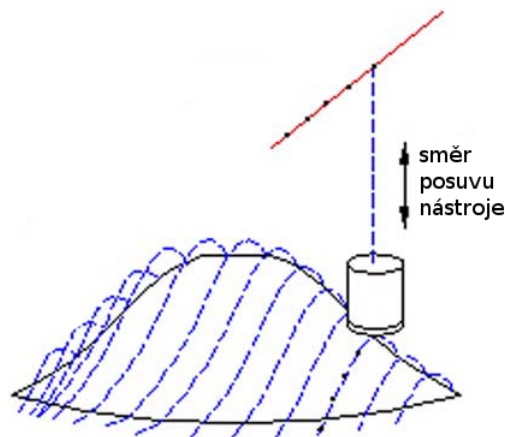
Zabýváme se jednoduchými strategiemi frézování, založenými na spojení dráhy nástroje z jednotlivých rovin. Základním problémem je tedy určit trajektorii nástroje v rovině. Prvotní myšlenkou bylo vyjít z řezu modelu danou rovinou. Tento přístup se ukázal problematický v tom, že neuvažoval možnost kontaktu nástroje s částmi modelu mimo oblast řezu. Navíc bral v úvahu pouze plochu trojúhelníků, ze kterých se skládá povrch modelu. Dráha nástroje podél trojúhelníku je však složitější, jak bude popsáno později. Protože NC stroje typicky rozumí jen drahám složených z přímých segmentů a kruhových oblouků, bylo by při tomto přístupu třeba některé části dráhy aproximovat.

Existuje několik dalších možných přístupů k řešení. Jedním z nich, je generování dráhy na základě výškové mapy [5], kterou získáme spouštěním nástroje do bodu dotyku podél jeho osy. Generování kompletní výškové mapy je však výpočetně náročné, především při vyšší požadované přesnosti. Výrazného zrychlení je možno dosáhnout případnou akcelerací pomocí grafické karty [6] [7]. V této práci však použijeme jiných postupů popsaných v [8], [9] a [3] s tím rozdílem, že některé dílčí problémy budou řešeny odlišným způsobem.

3.1 Paralelní strategie frézování

Tato metoda spočívá v proložení celého objemu modelu množinou rovnoběžných rovin, které jsou rovnoběžné s osou nástroje. Vzdálenost mezi jednotlivými rovinami je dána určeným krokem. Orientace rovin se typicky volí rovnoběžně s rovinou XZ, případně YZ. Není však vyloučena ani jiná orientace, pokud dodržíme podmínku rovnoběžnosti s osou nástroje. Základním stavebním prvkem je tak trajektorie nástroje při pohybu podél modelu v rovině. Vhodným spojením trajektorií z jednotlivých rovin dostaneme již konečný výsledek.

Pokud hledáme trajektorii nástroje ležící v rovině rovnoběžné s nástrojem bude výsledkem vždy otevřená křivka začínající na jednom okraji modelu a končící na druhém okraji. Promítnutím uvažované roviny rovnoběžné s osou nástroje do roviny XY získáme vodící přímku (červená přímka na obrázku 6). Pokud budeme schopni vypočítat výšku nástroje (souřadnici Z) postupně pro každý bod této přímky, tak aby se nástroj právě dotýkal modelu získáme tím samotnou trajektorii.



Obrázek 6: Princip výpočtu dráhy paralelní strategie. Zdroj [3].

Je zřejmé, že nebude možné uvažovat všechny body této vodící přímky, jichž je nekonečné množství. Uvažovanou rovinu tedy proložíme vhodnou množinou přímek rovnoběžných s osou nástroje [8] (svislá přerušovaná čára na obr. 6). V případě, že dokážeme určit bod, ve kterém se nástroj pohybující se po konkrétní svislé přímce dotkne modelu, dostaneme bod konečné trajektorie nástroje v této rovině. Jinak řečeno, k souřadnicím X a Y musíme dopočítat souřadnici Z , tak aby se nástroj umístěný na souřadnicích $[X, Y, Z]$ dotýkal modelu. Spojením bodů získaným na po sobě jdoucích svislých přímkách v rovině získáme samotnou trajektorii. Tento princip je vyobrazen na obrázku 6.

Uvedeným postupem sice nevznikne trajektorie kopírující naprosto přesně povrch modelu, ale vhodnou volbou délky kroku, případně jeho adaptivní změnou na základě tvaru modelu bude možné dosáhnout uspokojivých výsledků. Zřejmou výhodou je poměrně jednoduchý algoritmus, který poskytuje konzistentní výsledek nezávisle na složitosti tvaru modelu.

3.1.1 Souřadnice Z nástroje při dotyku s modelem pro dané souřadnice X a Y

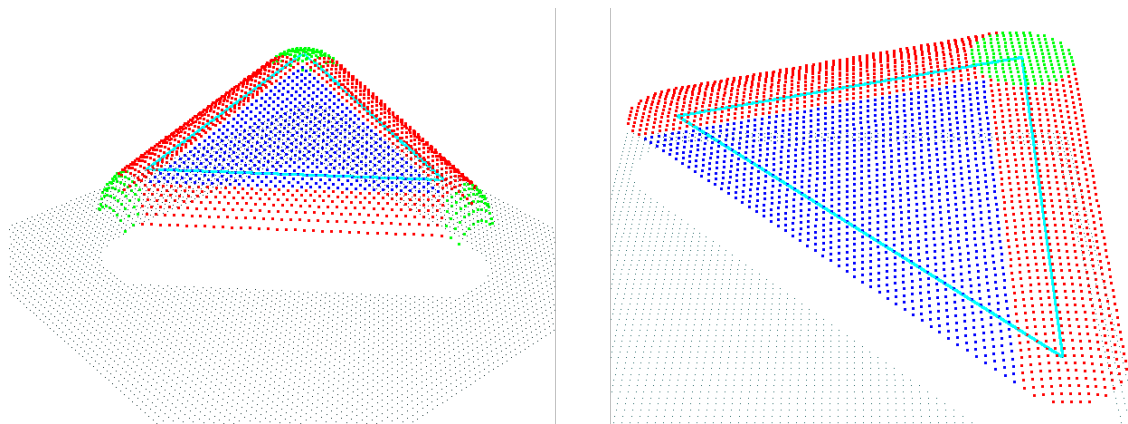
Uvažujeme povrchem reprezentované modely, kde jediným základním stavebním prvkem jsou trojúhelníky. Jedním z nejdůležitějších problémů při generování dráhy pro polygonální modely je tedy určit vzájemnou polohu samostatného trojúhelníku a frézy. Chceme-li určit dráhu tak, aby se nástroj nezanořil do objemu modelu musíme zaručit, aby se fréza dotýkala trojúhelníku vždy nejvýše v jednom bodě. Fréza může být v kontaktu s trojúhelníkem skrz jeho vrchol, jeho stranu nebo jeho

plochu. Tyto případy je potom třeba řešit samostatně [8][9]. Navíc je vzhledem k odlišné geometrii nutné uvažovat tyto problémy zvlášť pro kulovou a válcovou frézu.

Výpočet souřadnice Z spočívá v podstatě v umístění nástroje na dané souřadnice X a Y a jeho spuštění dolů směrem k modelu až dojde k dotyku s některým z trojúhelníků tvořících povrch modelu. Je tedy nutné určit množinu trojúhelníků, které se mohou dostat do kontaktu s nástrojem. Vzhledem k tomu, že nástroj se pohybuje ve směru osy Z, jedná se o trojúhelníky přímo pod nástrojem. Tedy takové jejich vzdálenost od osy nástroje (na souřadnicích X, Y) je menší než jeho poloměr. Následně určíme výšku nástroje (souřadnici Z) při dotyku s každým z těchto trojúhelníků. Výsledná výška potom bude nejvyšší hodnota vypočtená z dané množiny trojúhelníků.

3.1.2 Kontakt nástroje s trojúhelníkem

Fréza může být v kontaktu s trojúhelníkem skrze jeho vrchol, stranu nebo plochu. K těmto případům je třeba přistupovat jednotlivě, protože jejich geometrické řešení je odlišné. Budeme-li hledat odpovídající souřadnici Z pro nástroj umístěný na daných souřadnicích X a Y, určíme postupně souřadnici Z v místě kontaktu s plochou, stranami a vrcholy. Získáme tak až 7 různých hodnot, přičemž konečným výsledkem bude nejvyšší z takto vypočtených hodnot.



Obrázek 7: Možnosti kontaktu kulové (vlevo) a válcové (vpravo) frézy s trojúhelníkem.
Zdroj: <http://www.anderswallin.net/2010/03/drop-cutter-again/>;
<http://www.anderswallin.net/2010/03/spherical-drop-cutter/>.

Na obrázku 7 jsou zobrazena umístění špičky nástroje v prostoru v případě kontaktu s různými částmi trojúhelníku. Pokud budeme v dalším textu odkazovat na pojem „bod kontaktu“ myslí se tím právě bod, na kterém se nachází **špička nástroje** při kontaktu s trojúhelníkem. Bílé jsou označeny body bez kontaktu s trojúhelníkem. Modrou barvou je vyznačena množina bodů, na kterých se nástroj dotýká plochy trojúhelníku, červené body značí kontakt se stranou a zelené s vrcholem. Byť to

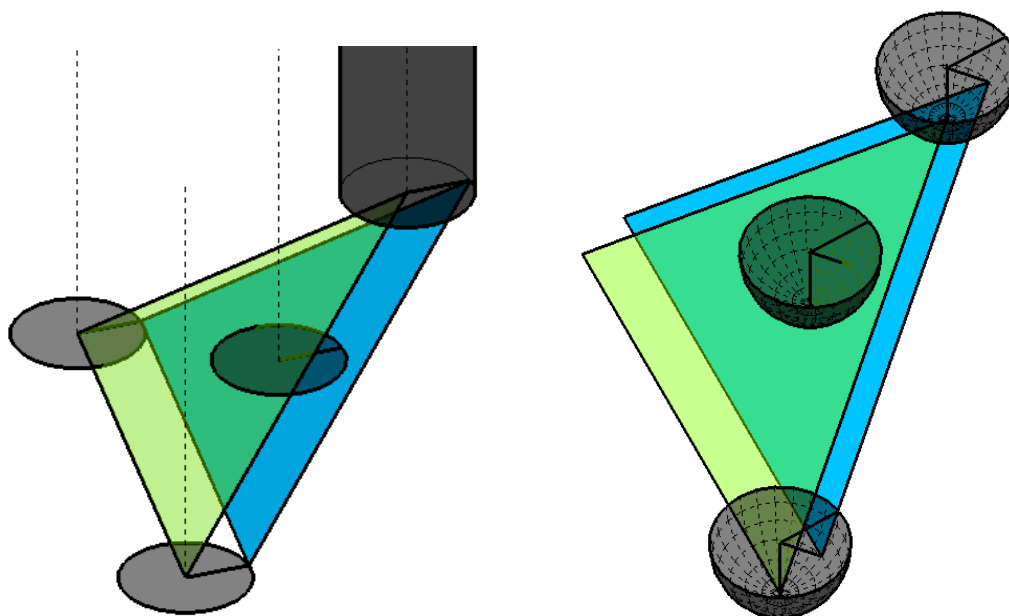
z obrázku nemusí být zcela zřejmé, je třeba si uvědomit, že body kontaktu s plochou a stranami, respektive vrcholy, neleží v jedné rovině. Například body kontaktu s vrcholem (zelené) v případě válcové frézy leží v horizontální rovině a body kontaktu s plochou (modré) v rovině rovnoběžné s plochou trojúhelníku.

3.1.3 Kontakt nástroje s plochou trojúhelníku

Jako první budeme řešit kontakt s plochou trojúhelníku. Přestože vycházíme z prací [8] a [9], budeme k řešení některých problémů přistupovat odlišně. Máme-li k dispozici podpůrné knihovny pro práci se základními geometrickými tvary, získáme tak často názornější řešení, které bude možno navíc použít i v hledání trajektorie v rovině kolmé na nástroj. Při hledání kontaktu frézy a plochy trojúhelníku použijeme právě vlastní přístup.

Známe-li bod, ve kterém se nástroj dotýká plochy trojúhelníku, můžeme jednoznačně určit také bod, na kterém se nachází špička nástroje. Poloha špičky nástroje vůči bodu dotyku bude sice záviset na tom, jestli se jedná o kulovou nebo válcovou frézu, ale pro daný nástroj bude toto přiřazení jednoznačné. Tím se kontakt nástroje s plochou trojúhelníku odlišuje od kontaktu se stranami a vrcholy. Jak je zřejmé z obrázku 7, dotyku nástroje s vrcholem (tedy jedním bodem) totiž náleží mnoho možností polohy nástroje (zelené body).

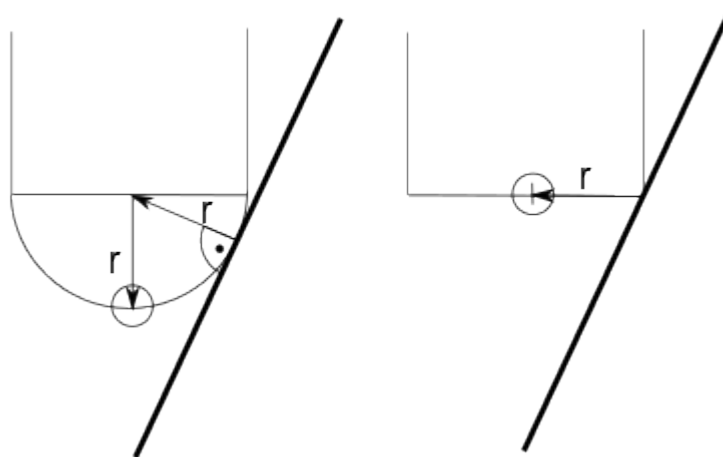
Vzhledem k jednoznačnému přiřazení mezi bodem dotyku na ploše trojúhelníku a pozicí



Obrázek 8: Dotyk s plochou trojúhelníku a odpovídající poloha nástroje.

Vraťme se zpět k řešenému problému, kdy máme zadány souřadnice X , Y osy nástroje a hledáme souřadnici Z , tak aby se nástroj dotýkal plochy trojúhelníku. Jak lze odvodit z obrázku 8, k dotyku s plochou trojúhelníku dojde v případě, že osa nástroje protne výše popsany obraz trojúhelníku a poloha špičky nástroje bude určena právě tímto průsečíkem. Zbývá nám tedy jen úkol odvodit tento obraz pro válcovou a kulovou frézu o daném průměru.

Na obrázku 9 je nakreslen stejný případ jako na obrázku 8, ovšem z pohledu kolmého na plochu trojúhelníku a naznačen je pouze původní trojúhelník bez jeho obrazu. Je zde patrné, že chceme-li získat umístění špičky nástroje pro kulovou frézu, musíme bod dotyku posunout o poloměr nástroje ve směru normálového vektoru trojúhelníku a následně o poloměr nástroje ve směru osy nástroje. V případě válcové frézy získáme směr posunutí promítnutím normálového vektoru trojúhelníku do roviny XY (jinak řečeno vynulujeme jeho souřadnici Z). Délka posunutí opět odpovídá poloměru nástroje. Obraz trojúhelníku poté sestrojíme posunutím jeho vrcholů podle právě uvedených pravidel.



Obrázek 9: Odvození polohy špičky nástroje od bodu dotyku

Samotné určení průsečíku obrazu trojúhelníku a osy nástroje se skládá z výpočtu průsečíku roviny obrazu a osy nástroje. Následně je třeba otestovat, jestli vypočtený průsečík leží uvnitř nebo vně trojúhelníku. Tyto základní geometrické operace obstarává knihovna CGAL, není je tedy třeba podrobněji rozebírat.

3.1.4 Kontakt nástroje se stranou trojúhelníku

Připomeňme si řešený problém. Uvažujeme nástroj na daných souřadnicích X , Y a hledáme souřadnici Z tak, aby se nástroj dotýkal strany trojúhelníku. K řešení kontaktu nástroje a strany trojúhelníku je třeba přistupovat odlišně od kontaktu s plochou. Ke každému bodu dotyku totiž může obecně příslušet více možností polohy nástroje.

Je důležité si uvědomit, že v tomto kroku řešíme dotek s vnitřní částí úsečky, tvořící stranu trojúhelníku, nikoliv s koncovými body (tedy vrcholy). Tyto případy budou řešeny samostatně v dalším kroku. Tím se nám zjednoduší řešení v tom, že nebude třeba uvažovat zakončení úsečky. Představme si pro ilustraci množinu bodů, které představují střed koule, jež se dotýká úsečky. Neboli také množinu bodů v konstantní vzdálenosti od této úsečky. Tyto body budou tvořit válcovou plochu podél úsečky a tato plocha bude na koncích úsečky uzavřena dvěma polokoulemi. V tomto kroku budeme uvažovat pouze vnitřní část úsečky – válcovou plochu, bez zakončovacích polokoulí (ty se budou řešit v rámci dotyku nástroje s vrcholem trojúhelníku).

Základním předpokladem, aby se vůbec nástroj mohl do kontaktu se stranou dostat je, že vzdálenost osy nástroje a úsečky tvořící stranu trojúhelníku musí být menší než je poloměr nástroje. Pokud je tato vzdálenost větší, není třeba se s danou stranou zabývat, protože ke kontaktu dojít nemůže. Vzhledem k odlišnému tvaru bude třeba řešit samostatně případ válcové a kulové frézy.

Případ 1: Válcová fréza

Nejdříve se budeme věnovat problému z pohledu válcové frézy. Spodní část válcové frézy tvoří rovná kruhová plocha, která se při spouštění nástroje ve směru jeho osy dostane do kontaktu s modelem vždy jako první. Řešení tedy omezíme pouze na tuto plochu. Vnitřní část úsečky navíc může být v kontaktu jen s krajovou kružnicí. Řešení tedy budeme moci ještě více omezit pouze na krajovou kružnici. Existuje jen jeden případ, kdy by se vnitřní body úsečky mohly při spouštění nástroje podél osy dotknout jiné části nástroje než je spodní kruhová hrana nástroje – úsečka kolmá na osu nástroje. V tomto případě je buď alespoň jeden z krajních bodů úsečky ve vzdálenosti větší než poloměr nástroje od jeho osy – můžeme použít níže popsaný postup, nebo oba krajní body leží v menší vzdálenosti – můžeme zanedbat, protože správnou výšku určíme v dalším kroku při určování kontaktu s vrcholy.

V podstatě celý následující postup výpočtu výšky bodu dotyku pro válcovou frézu je převzat z práce [8]. Na začátku je vhodné posunout celý problém tak, aby osa nástroje byla umístěna na souřadnicích $X=Y=0$. Vzhledem k tomu, že hledáme pouze souřadnici Z tak se tímto posunutím

řešení nezmění. Vstupem je úsečka představující stranu trojúhelníku, definovaná krajními body $\mathbf{p}_1(x_1, y_1, z_1)$ a $\mathbf{p}_2(x_2, y_2, z_2)$, kterou lze vyjádřit rovnicí 1.

$$\mathbf{p}(q) = (1-q) \cdot \mathbf{p}_1 + q \cdot \mathbf{p}_2, \quad q \in \langle 0; 1 \rangle \quad (1)$$

Jak je vysvětleno výše jediný možný případ dotyku vnitřní části úsečky je se spodní hranou nástroje. Promítnutím nástroje a úsečky do roviny XY se nám problém zjednoduší na hledání průsečíku kružnice (obraz hrany nástroje) a přímky. Tento průsečík je možné dosadit zpět do původní rovnice přímky v prostoru a tím získáme souřadnici Z odpovídající danému průsečíku a tedy i bodu dotyku. Kombinací rovnice kružnice se středem $[0, 0]$ o poloměru nástroje a rovnice úsečky (rovnice 1) po promítnutí do roviny XY získáme po úpravě rovnicí 2. Jejich jedinou neznámou je parametr úsečky q .

$$((x_2 - x_1)^2 + (y_2 - y_1)^2)q^2 + 2(x_1(x_2 - x_1) + y_1(y_2 - y_1))q + x_1^2 + y_1^2 - R^2 = 0 \quad (2)$$

Na základě řešení rovnice 2 můžeme dojít k několika možnostem. Pokud rovnice má dvě různá reálná řešení, budeme muset následně určit to, kterému odpovídá větší výška nástroje. Má-li rovnice 1 řešení, žádná komplikace nenastává. Pokud rovnice nemá reálné řešení, úsečka kružnici neprotne a můžeme tedy pokračovat dalším krokem. Dosazením vypočtené hodnoty parametru q do rovnice 1 pro souřadnici Z dostaneme požadovanou výšku dotyku nástroje s uvažovanou úsečkou. Ještě před tím je však třeba zkontrolovat hodnotu tohoto parametru. Řešením rovnice 2 totiž dostaneme hodnotu obecně z celé množiny reálných čísel. Platné hodnoty parametru pro uvažovanou přímku však leží pouze v intervalu $\langle 0, 1 \rangle$. Hodnoty parametru q mimo tento interval znamenají průsečík kružnice a přímky definované body \mathbf{p}_1 a \mathbf{p}_2 , který však leží mimo úsečku definovanou těmito body. Hodnoty parametru q mimo interval $\langle 0, 1 \rangle$, tedy nesmí být brány v potaz.

Případ 2: kulová fréza

K problému dotyku kulové frézy a stranu trojúhelníku bude třeba přistupovat výrazně odlišně od válcové frézy. Na rozdíl od válcové frézy, kde je spodní strana nástroje tvořena rovnou plochou, bude plocha, která se může potenciálně dostat do kontaktu s úsečkou, ve tvaru polokoule. Stejně jako v případě dotyku s plochou trojúhelníku se odchýlíme od postupů uvedených v článkách [8] a [9] a použijeme vlastní řešení.

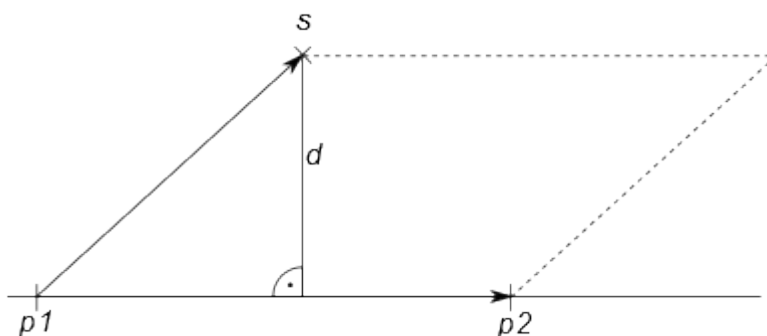
Pro zjednodušení se soustředíme na hledání středu kulové plochy nástroje. Známe-li tento střed, můžeme polohu špičky získat jednoduchým posunutím ve směru osy nástroje o délku poloměru. Na začátku znovu využijeme posunutí celého problému tak, aby osa nástroje byla umístěna

na souřadnicích $X=Y=0$, jelikož ve směru osy Z k posunu nedochází, nezmění se tím ani výsledná výška.

Vstupem bude již odpovídajícím způsobem posunutá úsečka představující stranu trojúhelníku definovaná krajními body $p_1(x_1, y_1, z_1)$ a $p_2(x_2, y_2, z_2)$. Hledáme souřadnici z_s středu koule o poloměru R , tak aby se dotýkala přímky dané body p_1 a p_2 . Pokud bude tento dotyk ležet mezi body p_1 a p_2 , bude se jednat o dotyk vnitřní části dané úsečky a tedy vypočtená hodnota z_s bude platnou hodnotou výšky nástroje. Nástroj se po počátečním posunutí bude pohybovat podél osy Z , souřadnice středu koule tedy budou ve tvaru $[0, 0, z_s]$.

Povrch koule je tvořen množinou bodů, které jsou od středu koule ve vzdálenosti dané poloměrem koule. Má-li se povrch koule dotýkat dané přímky p_1p_2 musí být střed koule od této přímky právě ve vzdálenosti dané poloměrem koule. Hledáme tedy bod na ose Z , který je v dané vzdálenosti od přímky p_1p_2 .

Prvním krokem bude odvodit vztah pro určení vzdálenosti bodu od přímky. Mějme tedy přímku určenou body p_1 a p_2 a bod s . Potřebujeme určit jejich vzdálenost. Body p_1 , p_2 a s určují rovnoběžník vyznačený na obrázku 10. Vzdálenost bodu s a přímky p_1p_2 zde představuje výšku tohoto rovnoběžníku. Využijeme toho, že obsah plochy rovnoběžníku lze vyjádřit pomocí součinu délky strany a odpovídající výšky a také pomocí velikosti vektorového součinu stran rovnoběžníku (rovnice 3). Úpravou této rovnice získáme vztah pro výpočet vzdálenosti bodu s od přímky p_1p_2 (rovnice 4).

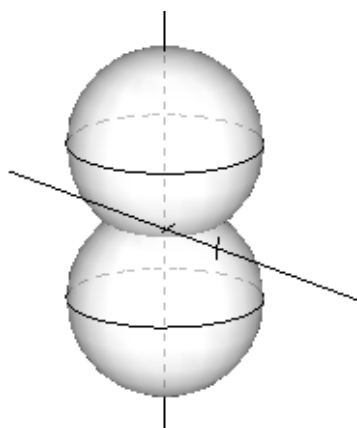


Obrázek 10: Vzdálenost přímky a bodu.

$$S = \|p_1 p_2\| \cdot d = \|\vec{p}_1 \vec{p}_2 \times \vec{p}_1 \vec{s}\| \quad (3)$$

$$d = \frac{\|\vec{p}_1 \vec{p}_2 \times \vec{p}_1 \vec{s}\|}{\|p_1 p_2\|} \quad (4)$$

Ze vztahu 4 již můžeme vyjít při určování výšky středu kulové plochy nástroje. Vzdálenost d položíme rovnu poloměru nástroje. Body p_1 a p_2 jsou koncové body testované strany. Nástroj se pohybuje po ose Z , tedy jedinou neznámou bude souřadnice Z bodu s (středu kulové plochy nástroje). Řešení rovnice 4, kde vystupuje jako neznámá hodnota z_s povede na kvadratickou rovnici, jejíž dvě možná řešení odpovídají i prostorové reprezentaci, která je vyobrazena na obrázku 11. Zde jsou znázorněny dvě možnosti dotyku přímky a koule, za předpokladu, že možnost pohybu je omezena podél osy Z . Nemá-li rovnice reálné řešení, přímka p_1p_2 leží od osy nástroje ve vzdálenosti větší než je poloměr, tím pádem ke kontaktu dojít nemůže.



Obrázek 11: Možnosti dotyku koule se přímkou při pohybu podél osy nástroje

Je patrné, že jeden z bodů dotyku leží na horní polokouli. Tvar kulové frézy však tvoří pouze spodní polovina kulové plochy. Ze dvou vypočtených hodnot z_s je tedy třeba vybrat tu vyšší, která odpovídá středu horní koule z obrázku 11 a tím pádem dotyku s přímkou v její spodní polovině.

Zjistíme-li platnou pozici středu kulové plochy při kontaktu s přímkou, musíme ještě zkontrolovat, jestli odpovídající bod dotyku leží mezi body p_1 a p_2 , tedy na zkoumané straně trojúhelníku. Polohu bodu dotyku zjistíme například tak, že vytvoříme rovinu kolmou na úsečku p_1p_2 , která prochází zjištěným středem koule. Průsečík takové roviny a přímky p_1p_2 pak představuje bod dotyku. Úplně posledním krokem je určení polohy špičky nástroje na základě polohy středu kulové plochy, které se provede prostým odečtením poloměru nástroje (tj. posunutím proti směru osy Z).

3.1.5 Kontakt nástroje s vrcholem trojúhelníku

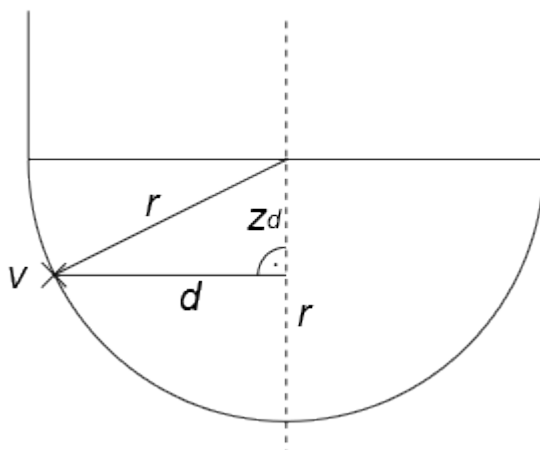
Posledním možným případem kontaktu nástroje a trojúhelníku je dotyk s jeho vrcholem. Vzhledem k různému tvaru bude třeba znovu přistupovat jednotlivě k případu válcové a kulové frézy. Stejně

jako v předchozím případě budeme pro zjednodušení řešit problém po posunutí, tak aby osa nástroje splývala s osou Z. Odvození je oproti předchozím případům poměrně primitivní, takže poměrně přesně odpovídá i postupům popsaných v pracích, ze kterých vycházíme.

Začneme jednodušším případem – válcovou frézou. Při spouštění podél osy nástroje se nástroj může dostat do kontaktu s vrcholem trojúhelníku jen skrze spodní kruhovou plochu. Podmínkou dotyku tedy je, aby vzdálenost vrcholu V (v_x, v_y, v_z) a osy nástroje byla menší než poloměr nástroje. Platí-li tedy nerovnost 5, bude se válcová fréza dotýkat vrcholu trojúhelníku V a to přímo ve výšce v_z . Spodní plocha nástroje je totiž rovnoběžná s rovinou XY, proto bude i špička nástroje ve stejné výšce jako je vrchol, s nímž je nástroj v kontaktu.

$$\sqrt{v_x^2 + v_y^2} < R \quad (5)$$

Řešení kontaktu vrcholu a kulové frézy bude mírně složitější. Pro test, jestli k dotyku s vrcholem vůbec dojde můžeme znovu použít vztah 5. Odpovídající polohu špičky nástroje při případném dotyku však bude třeba odvodit z obrázku 12. Osa nástroje splývá s osou Z. Vzdálenost d vrcholu V a osy nástroje tedy bude určena vztahem 6. Přičtením vzdálenosti z_d k souřadnici v_z dostaneme výšku středu kulové plochy, následným odečtením poloměru získáme výšku špičky nástroje (rovnice 7).



Obrázek 12: Dotyk kulové frézy a vrcholu.

$$d = \sqrt{v_x^2 + v_y^2} \quad (6)$$

$$\begin{aligned} z &= v_z + z_d - r \\ z &= v_z + \sqrt{r^2 - (v_x^2 + v_y^2)} - r \end{aligned} \quad (7)$$

3.2 Horizontální strategie frézování

Podobně jako při řešení paralelních rovin budeme vycházet z množiny rovin, tentokrát kolmých na osu nástroje, kterými s konstantním krokem proložíme celý objem modelu. Budeme uvažovat každou z těchto rovin samostatně, přičemž pro každou z nich určíme trajektorii nástroje podél modelu. Vhodným propojením trajektorií z těchto rovin získáme trajektorii konečnou.

Pokud uvažujeme uzavřené prostorové modely, bude nástroj pohybující se v rovině kolmé na nástroj při opisování tvaru modelu vždy tvořit uzavřenou křivku. Postup popsany při řešení roviny rovnoběžné s nástrojem tedy nebude možné bez zásadních změn použít.

Pro řešení problému použijeme postup popsany v [3] s tím rozdílem, že některé dílčí problémy budou řešeny odlišným způsobem. Výhoda tohoto postupu spočívá v tom, že lze použít část funkcionality implementované v rámci řešení paralelních rovin. Algoritmus vychází z průsečíku modelu a roviny kolmé na nástroj. V případě uzavřených polygonálních modelů bude průsečík tvořen uzavřeným mnohoúhelníkem. Máme-li k dispozici tento průsečík, pokračujeme určením offset vektorů pro každý z vrcholů mnohoúhelníku. Vlastnosti offset vektorů se budou lišit u konvexních a konkávních úhlů příslušného vrcholu. Bod trajektorie nástroje získáme tak, že do konkrétního vrcholu průsečíku umístíme špičku nástroje a ve směru offset vektoru posunujeme nástroj do té míry, až se ztratí kontakt nástroje a modelu. V tomto místě se nacházíme v bodě konečné trajektorie nástroje. Celou trajektorie získáme postupným průchodem všech vrcholů mnohoúhelníku podle výše popsaného postupu. I zde se tedy uplatňuje jistý posuv nástroje. Tentokrát však nikoliv ve směru osy nástroje, ale ve směru offset vektoru, který leží v dané rovině řezu a je tedy na osu nástroje kolmý. V tomto kroku využijeme některých postupů uvedených v kapitole 3.1. Princip tohoto postupu je vyobrazen na obrázku 13.



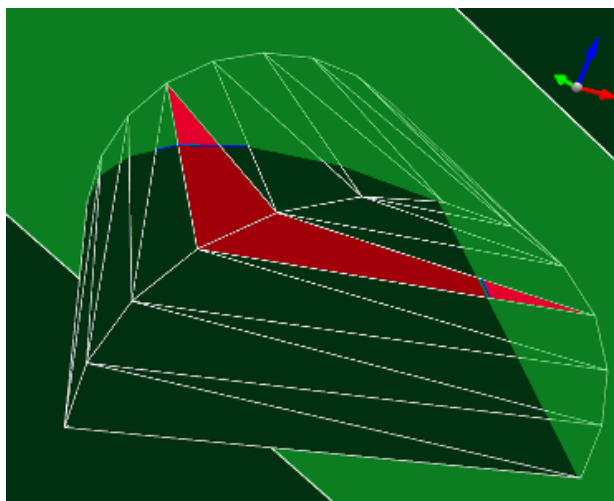
Obrázek 13: Princip výpočtu dráhy horizontální strategie. Zdroj [3].

Části takto získané trajektorie se však mohou za určitých podmínek protínat. Je tedy nutné takové průniky najít a odstranit. Řez roviny a modelu může být tvořen několika samostatnými mnohoúhelníky, čímž získáme několik samostatných trajektorií nástroje. Je tedy nutné vyřešit také případné protnutí samostatných trajektorií.

3.2.1 Průsečík roviny a modelu

Prvním krokem algoritmu je určení průsečíku modelu a roviny kolmé na osu nástroje. Uvažujeme modely reprezentované povrchem, který je dán množinou trojúhelníků. Řešení průsečíku roviny a takovýchto modelů, importovaných z STL souborů, se věnuje např. práce [10]. My však využijeme vlastního postupu, kde využijeme možností poskytovaných knihovnou OpenMesh pro reprezentaci polygonálních sítí. Knihovna OpenMesh po importu souboru STL zajistí rekonstrukci povrchu z množiny samostatných trojúhelníkových ploch a umožní efektivní procházení takového povrchu tím, že poskytuje odkazy na sousední plochy, hrany a vrcholy. Více informací i knihovně OpenMesh naleznete v kapitole 4.1.

Řez uzavřeného trojúhelníkového polygonálního modelu rovinou bude tvořen jedním nebo několika uzavřenými mnohoúhelníky. Slovem řez budeme mít dále na mysli výsledný mnohoúhelník, termínem průsečík pak budeme typicky myslet průsečík roviny řezu a jednoho z trojúhelníků povrchu modelu. Každou ze stran mnohoúhelníku řezu bude představovat průsečík uvažované roviny a jednoho z trojúhelníků tvořících povrch modelu. Algoritmus tedy spočívá v prvním kroku v nalezení libovolného trojúhelníku, který je protnut rovinou řezu. Tento trojúhelník se vloží do množiny zpracovaných trojúhelníků a jeho průsečík se uloží jako první strana řezu. Dále se bude postupovat podél modelu, tak, že v bezprostředním okolí aktuálně zpracovávaného trojúhelníku najdeme další trojúhelník protnutý rovinou řezu. Pokud tento trojúhelník nenáleží do množiny již zpracovaných trojúhelníků uložíme jeho průsečík jako další segment řezu a pokračujeme v hledání dalších protnutých trojúhelníků v jeho okolí. Algoritmus úspěšně skončí tím, že dojdeme zpět k prvnímu protnutému trojúhelníku. Tím také dojde k uzavření polygonu řezu. Je však třeba projít ještě zbývající trojúhelníky povrchu modelu, které dosud nebyly zpracovány, protože se řez může skládat z více samostatných mnohoúhelníků.



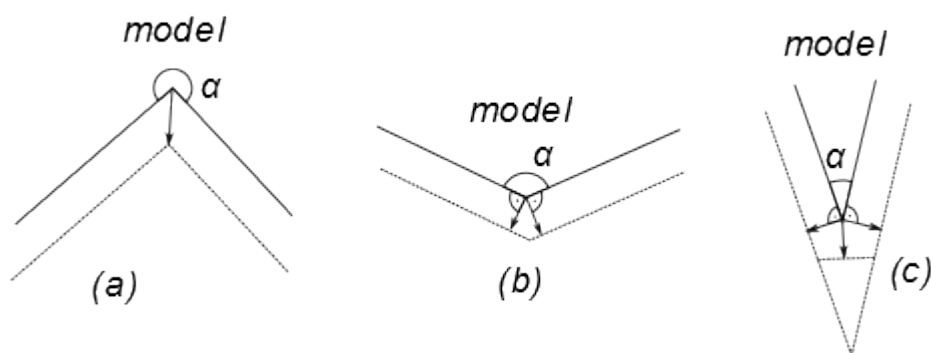
Obrázek 14: Problém skládání řezu průchodem sousedících trojúhelníků.

V některých případech však může výše uvedený postup selhat, jak je znázorněno na obrázku 14. Je zde jasně vidět, že přestože jsou zvýrazněné trojúhelníky v bezprostředním sousedství, části řezu náležící těmto trojúhelníkům už po sobě přímo nenásledují. Výše uvedený algoritmus je tedy třeba doplnit o test, jestli průsečíky roviny s aktuálním a kandidátním trojúhelníkem sdílejí jeden z koncových bodů. Alternativním řešením je uvažovat pouze takové okolní trojúhelníky, které s aktuálním trojúhelníkem sousedí skrze hranu nebo vrchol, protnutý rovinou řezu.

Jeden uzavřený mnohoúhelník řezu rozděluje rovinu na dvě části – vnitřní a vnější. Jedna z nich vždy náleží objemu modelu a druhá znamená volný prostor. Pro další výpočty je třeba zajistit, aby byl směr definování mnohoúhelníku řezu konzistentní vzhledem k umístění vnitřní části modelu. Tedy aby v případě řezu definovaného proti směru hodin náležela ohraničená oblast vždy objemu modelu a ohraničená oblast řezu definovaného po směru hodin naopak představovala volný prostor.

3.2.2 Určení offset vektorů

Následujícím krokem algoritmu je určení offset vektorů pro každý z vrcholů řezu. Tyto vektory budou určovat směr posuvu nástroje od daného vrcholu. Nástroj se má pohybovat v dané rovině řezu, která je kolmá na osu nástroje, všechny offset vektory tedy musí ležet v této rovině. Podle [3] je vhodné rozdělit generování offset vektorů na tři případy, podle velikosti vnitřního úhlu, který svírají přilehlé strany. Na obrázku 15 jsou tyto případy rozkresleny. Plná čára vyznačuje část řezu, přerušovaná čára značí část trajektorie nástroje.



Obrázek 15: Offset vektorů v závislosti na vnitřním úhlu vrcholu.

Nejjednodušší je případ konkávního vrcholu (obr. 15.a, $\alpha \geq \pi$). Offset vektor získáme, jako průměr normálových vektorů přilehlých stran. Bod trajektorie získáme přímo posunem nástroje ve směru tohoto offset vektoru až dojdeme do bodu, kde se nástroj právě dotýká modelu.

Ke konvexním vrcholům (obr. 15.b, $\alpha < \pi$) bude třeba přistupovat odlišně. Matematicky přesná trajektorie nástroje by byla tvořena přímými segmenty podél stran a tyto segmenty by spojovala část kružnice se středem ve vrcholu. Při pohybu po obloukové části by však byl nástroj neustále v kontaktu s vrcholem a vlivem oscilací by mohlo dojít k jeho stržení. Vhodnějším přístupem bude prodloužení těchto přímých segmentů až do jejich průniku, kde bude nový bod trajektorie nástroje. S tímto přístupem si navíc vystačíme při definici trajektorie pouze s úsečkami. Ke každému vrcholu tedy budou náležet 2 offset vektory, který budou mít směr normálových vektorů přilehlých stran.

Pokud bychom však použili výše popsaný postup také u ostře konvexních vrcholů (obr. 15.c), vycházel by bod trajektorie ve velké vzdálenosti od vrcholu, což znamená zvýšené riziko nežádoucí kolize s jinou částí modelu. U vrcholů s malým vnitřním úhlem ([3] uvádí jako vhodnou mez $\alpha < \pi/2$) je tedy vhodné doplnit předcházející postup o vygenerování třetího offset vektoru a v bodě získaném posunem podél tohoto vektoru vygenerovat přímku na offset vektor kolmou. Výsledné body trajektorie budou potom tvořeny průsečíkem této přímky a prodlouženými segmenty, který jsme získali postupem popsaným v předchozím odstavci. Třetí offset vektor získáme stejně jako v případě konkávního vrcholu jako průměr normálových vektorů přilehlých stran.

Pokud bychom převzali takto popsaný postup z práce [3] beze změn, mohlo by v případě dlouhých stran mnohoúhelníku řezu dojít až k odstranění částí modelu, které ve vyšších úrovních přesahují aktuální řez. Body dotyku se totiž počítají pouze ve vrcholech řezu a dráha vzniká pouhým jejich spojením. Body dotyku je tedy třeba s vhodným krokem počítat také podél stran řezu. Tím dojde k detekci a vygenerování správné dráhy okolo případných kolizí podél stran řezu.

3.2.3 Kontakt nástroje s modelem

Podobně jako v případě rovin rovnoběžných s osou nástroje budeme opět hledat bod dotyku nástroje a modelu při posunu nástroje. Tentokrát však nikoliv při posunu podél osy nástroje, ale podél daného offset vektoru. Máme určen polygon řezu a ke každému z jeho vrcholů náleží jeden nebo několik offset vektorů. Posunem podél každého z těchto offset vektorů získáme jeden bod konečné trajektorie nástroje. Tento bod bude ležet na polopřímce určené vrcholem řezu a offset vektorem. Průchodem množiny kandidátních trojúhelníků povrchu modelu určíme bod, ve kterém se nástroj při pohybu podél offset vektoru dotkne právě uvažovaného trojúhelníku. Z těchto bodů vybereme bod v největší vzdálenosti od vrcholu řezu, který bude již bodem konečné trajektorie nástroje. Stále přitom vycházíme z práce [3], která jako kandidátní trojúhelníky určuje ty, které leží v půdorysu nástroje, jehož špička se nachází na zkoumaném vrcholu řezu.

Při určování dotyku nástroje a trojúhelníku budeme opět samostatně řešit jeho plochu, strany a vrcholy. Pro zjednodušení výpočtu provedeme posun systému *nástroj + trojúhelník + vrchol řezu + offset vektor* tak, aby vrchol řezu ležel v počátku souřadného systému a následně otočení tak, aby offset vektor směřoval podél osy Y. K této transformaci je možné určit transformaci inverzní a tak nakonec převést výsledek zpět do původní orientace. Provedeme-li tuto transformaci, budou souřadnice X a Z hledaného bodu dotyku vždy nulové. Jedinou neznámou bude souřadnice Y, která bude představovat vzdálenost špičky nástroje při dotyku s trojúhelníkem a zkoumaného vrcholu řezu.

Pohyb nástroje podél offset vektoru kolmého na osu nástroje se oproti pohybu podél osy nástroje liší tím, že se nástroj může dostat do kontaktu s modelem kromě spodní kruhové plochy, respektive polokulové plochy, také skrze svoji válcovou část. Při zkoumání dotyků s jednotlivými částmi trojúhelníku se odchýlíme od postupu v článku [3] a použijeme postupy popsané v kapitole 3.1 s rozšířením o detekci dotyku s válcovou plochou.

3.2.4 Kontakt nástroje s plochou trojúhelníku

Po výše popsané transformaci se špička nástroje pohybuje po ose Y. Při tomto pohybu nemůže nastat situace aby se válcová plocha nástroje dostala do kontaktu s vnitřní plochou trojúhelníku. Vnitřní plochy trojúhelníku se může dotknout pouze spodní kruhová hrana válcové frézy nebo spodní polokulová plocha kulové frézy. Pokud by měl být trojúhelník v dotyku s válcovou plochou nástroje bude to vždy skrze jeho stranu nebo vrchol. Jediný speciální případ, kdy toto neplatí je trojúhelník, jehož normálový vektor kolmý na osu nástroje. V takovém případě se však válcové plochy bude dotýkat také nejméně jedna ze stran trojúhelníku, takže není nutné toto řešit v rámci dotyku s plochou.

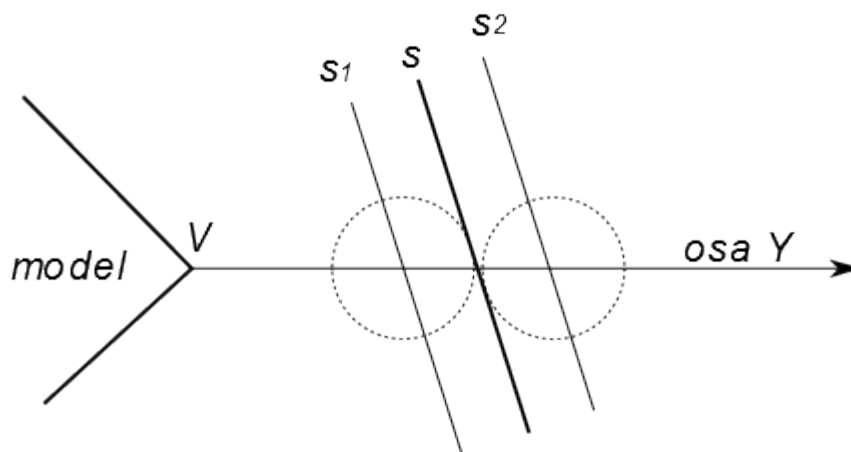
Řešení tedy můžeme beze změn převzít z kapitoly 3.1.3 s tím, že výsledným bodem dotyku bude průsečík obrazu trojúhelníku a osy Y.

3.2.5 Kontakt nástroje se stranou trojúhelníku

V případě řešení dotyku nástroje a hrany trojúhelníku už budeme muset uvažovat i možnost dotyku s válcovou plochou nástroje. Při zkoumání dotyku nástroje tedy budeme uvažovat zvlášť válcovou plochu a kulovou plochu v případě kulové frézy nebo kruhovou spodní hranu v případě válcové frézy. Řešení válcové plochy bude společné pro válcovou i kulovou frézu.

Válcová plocha nástroje

Na obrázku 16 je rozkreslen problém dotyku nástroje se stranou trojúhelníku po promítnutí do roviny XY. V levé části je naznačena část mnohoúhelníku řezu a vrchol V , ze kterého vycházíme. Špička nástroje se pohybuje po polopřímce určené tímto vrcholem a offset vektorem (který po transformaci splývá s osou Y). Kolmým posunutím uvažované strany s o poloměr nástroje na obě strany dostaneme úsečky s_1 a s_2 . Výsledný střed nástroje při dotyku se stranou s bude ležet na průsečíku osy Y a strany s_1 nebo s_2 . Hledáme vnější dotyk, takže hledaný bod bude ten s větší hodnotou souřadnice Y.



Obrázek 16: Kontakt nástroje a strany trojúhelníku.

Takové řešení by platilo pro nekonečnou válcovou plochu. Válcové plocha se však při pohybu špičky nástroje podél osy Y nachází pouze v oblasti $Z > 0$ v případě válcové frézy a $Z > \text{poloměr nástroje}$ v případě kulové frézy. Musíme tedy určit také souřadnici Z bodu dotyku a zjistit, jestli splňuje tyto podmínky. To provedeme převedením řešení do prostoru. V takovém případě budou oba

směry posunutí strany s na ni kolmé a zároveň rovnoběžné s rovinou XY (jinými slovy souřadnice Z se při posunu nezmění). Délka posunutí zůstává rovna poloměru nástroje. Výsledné středy válcové plochy, při jejím dotyku s uvažovanou stranou s , budou potom ležet na průsečíku stran s_1 a s_2 a roviny YZ . Vnější dotyk strany zajistíme výběrem průsečíku s větší hodnotou souřadnice Y . Následně podle souřadnice Z ověříme, jestli bod dotyku leží ve válcové oblasti podle tvaru nástroje. Pokud k dotyku nedojde v oblasti válcové plochy nebo vůbec nedojde k průniku přímky s_1 nebo s_2 s rovinou YZ , potom je strana s v pozici, kdy ke kontaktu s válcovou plochou vůbec nedojde.

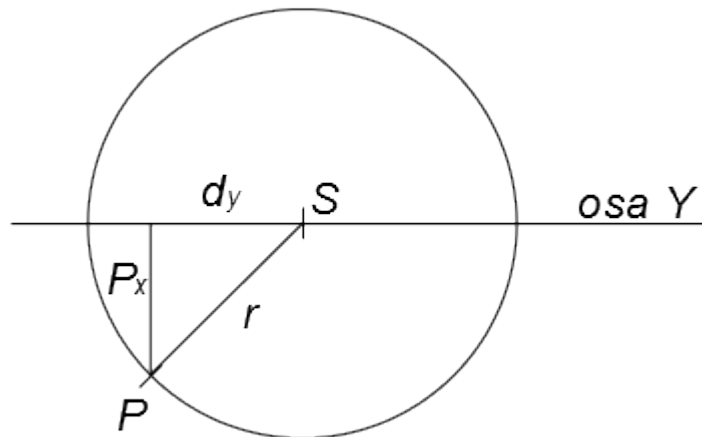
Kulová část kulové frézy

Řešení dotyku kulové plochy kulové frézy a strany trojúhelníku bude naprosto stejné jako v kapitole 3.1.4. Jedinou změnou bude, že zaměníme osu nástroje za osu Y . Obecně můžeme opět získat 2 výsledky (vnitřní dotyk a vnější dotyk). Vybereme vždy vnější dotyk, tedy ten s větší hodnotou souřadnice Y . Protože kapitola 3.1.4 řeší dotyk přímky a koule jako celku, je možné, že výsledkem bude dotyk v horní polokouli, která je však skryta uvnitř stopky nástroje. Podobně jako v případě válcové plochy musíme provést test, jestli dotyk leží v oblasti spodní polokulové plochy, tedy zda platí, že $0 < z_d < \text{poloměr nástroje}$.

Kruhá hřana válcové frézy

Válcová fréza se při pohybu podél osy Y může dotknout vnitřní části úsečky pouze válcovou plochou (popsáno výše) nebo spodní kruhovou hranou. K dotyku se spodní plochou dojít nemůže, protože výška nástroje (souřadnice Z) zůstává konstantní. Tím se nynější situace liší od té popsané v kapitole 3.1.4 a bude proto nutné odvodit jiné řešení.

Hledáme tedy případný dotyk úsečky (strany trojúhelníku) a spodní kruhové hrany nástroje, která leží v rovině XY . Tento dotyk musí tedy ležet také v rovině XY . Jediným kandidátem na bod dotyku bude průsečík roviny XY a zkoumané úsečky. Při pohybu nástroje podél osy Y vytvoří spodní kruhová hřana tímto pohybem pás v rovině XY o šířce průměru nástroje. Pokud průsečík úsečky a roviny XY bude ležet uvnitř tohoto pásu, dojde k dotyku spodní hrany nástroje a úsečky právě v tomto bodě. Zbývá tedy na základě známého bodu dotyku určit odpovídající střed nástroje na ose Y .



Obrázek 17: Odvození středu nástroje od bodu dotyku.

Známe tedy bod dotyku $P = [p_x, p_y, 0]$ a hledáme střed $S = [0, s_y, 0]$ nástroje o poloměru r . Z obrázku 17 je patrné, že souřadnice s_y středu S bude dána vztahem 8. Druhý platný střed bychom získali také odečtením délky d_y , to by se ovšem jednalo o vnitřní dotyk, o který nám nejde.

$$\begin{aligned} s_y &= p_y + d_y \\ s_y &= p_y + \sqrt{r^2 - p_x^2} \end{aligned} \quad (8)$$

3.2.6 Kontakt nástroje s vrcholem trojúhelníku

Podobně jako v případě dotyku strany trojúhelníku i zde budeme muset uvažovat případný dotyk s válcovou plochou tvořící stopku kulové frézy. Naopak se nám zjednoduší řešení válcové frézy, protože při pohybu nástroje ve vodorovné rovině je jedinou částí, která se může dotknout bodu v prostoru právě válcová plocha nástroje.

Válcová plocha nástroje

Hledáme střed nástroje, který se dotýká vrcholu trojúhelníku v ploše modelu, přičemž pohyb nástroje (respektive jeho špičky) je omezen podél osy Y. Zde pozor na rozlišení pojmů vrchol trojúhelníku a vrchol řezu – vrchol mnohoúhelníku řezu, pomocí kterého je spolu s offset vektorem definována polopřímka, na níž je omezen pohyb špičky nástroje. Po příslušné transformaci odpovídá vrcholu řezu počátek souřadného systému a offset vektor je orientován kladným směrem osy Y.

Při pohybu válcové frézy podél osy Y je válcová plocha nástroje omezena na kladné hodnoty souřadnice Z. Pokud je tedy vrchol trojúhelníku pod touto mezí (jeho souřadnice Z je menší než nula) ke kontaktu s nástrojem dojít nemůže a není třeba se daným vrcholem trojúhelníku dále zabývat.

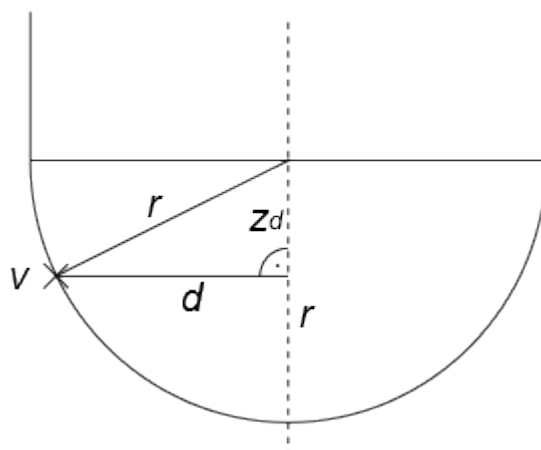
Podobně i v případě kulové frézy je její válcová část omezena, tentokrát však nikoliv ve výšce $Z > 0$, ale ve výšce dané poloměrem nástroje ($Z > r$). Pokud provedeme jako první tento test souřadnice Z , můžeme bez rizika dále uvažovat dotyk s nekonečnou válcovou plochou.

Pohybem nekonečné válcové plochy o poloměru r podél osy Y vznikne v prostoru oblast ohraničená dvěma rovinami ve vzdálenosti r od roviny YZ . Pokud bod leží mimo tuto oblast, k dotyku nedojde. Pokud bod leží v této oblasti, bude dotyk dán přímo polohou vrcholu trojúhelníku. Poloha středu nástroje bude následně určena vztahem 8, kde za bod dotyku P dosadíme souřadnice zkoumaného vrcholu trojúhelníku.

Protože se vrchol nemůže při vodorovném pohybu nástroje dostat do kontaktu se spodní částí válcové frézy stačí uvažovat pouze možnost dotyku s válcovou plochou.

Kulová plocha nástroje

V případě kulové frézy o poloměru r , jejíž špička se pohybuje podél osy Y , leží (polo)kulová oblast vždy mezi hladinami $Z = 0$ a $Z = r$. Prvním krokem při zkoumání kontaktu kulové části frézy a vrcholu trojúhelníku bude tedy test, jestli vrchol leží v oblasti kulového tvaru nástroje. Tím bez složitých výpočtů vyřadíme velké množství testovaných vrcholů. V dalším postupu budeme pracovat s kulovou plochou bez omezení, takže testem hladiny vrcholu vyřadíme i případné dotyky s vrchní polokoulí, která je ukryta uvnitř stopky nástroje.



Obrázek 12: Dotyk kulové frézy a vrcholu.

Při určování dotyku kulové plochy s vrcholem opět vyjdeme z obrázku 12 z kapitoly 3.1.5. Špička nástroje nyní bude ukotvena na ose Y a přerušovaná čára bude představovat rovinu YZ . Řezem kulové plochy vodorovnou rovinou v hladině vrcholu $V(v_x, v_y, v_z)$ bude kružnice o poloměru d . Aby se vrchol dotkl této kružnice musí být jeho vzdálenost od roviny YZ menší než poloměr d .

Pokud tedy platí vztah 10 vrchol se nástroje dotkne a pozici špičky nástroje odvodíme vztahem 8, kde bod dotyku P nahradíme vrcholem V a poloměr nástroje r nahradíme poloměrem d (rovnice 9).

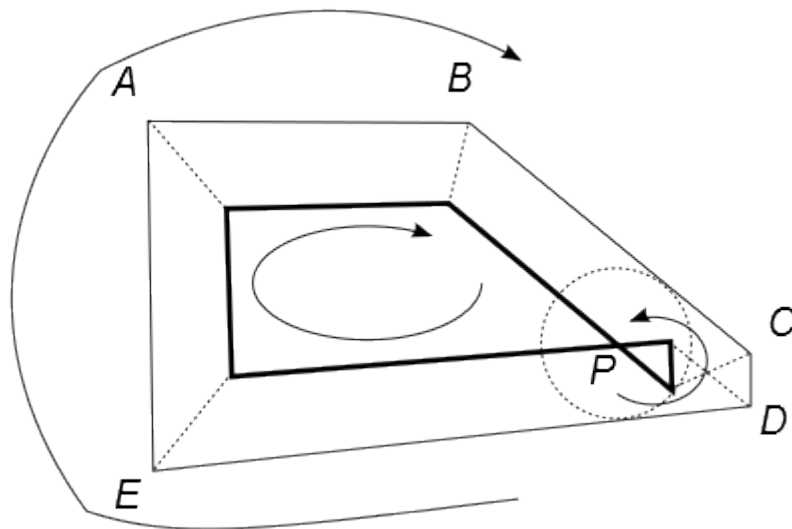
$$d = \sqrt{r^2 - (r - v_z)^2} \quad (9)$$

$$\begin{aligned} |x_v| &< d \\ |x_v| &< \sqrt{r^2 - z_d^2} \\ |x_v| &< \sqrt{r^2 - (r - v_z)^2} \end{aligned} \quad (10)$$

3.2.7 Odstranění křížení trajektorie

Dráha nástroje odvozená od polygonálního řezu může za určitých podmínek protínat samu sebe. Typicky se tak stává v úzkých nekonvexních částech, do kterých se nástroj kvůli svému průměru nemůže dostat. Pokud by trajektorie zůstala bez odstranění křížení, vedlo by to v místech křížení k poškození výsledného výrobku. Jeden z možných způsobů odstranění křížení je naznačen v [3]. My však použijeme odlišný přístup.

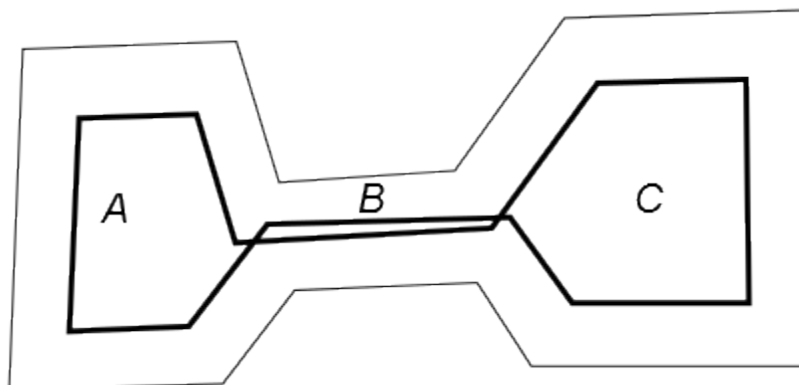
Dráha nástroje je v našem případě tvořena množinou navazujících úseček. Příklad takové křížící se dráhy je vyznačen tučnou čarou na obrázku 18. Tenká spojitá čára představuje řez modelu, ze kterého vychází celý postup generování dráhy. Přerušované čáry mající směr offset vektoru spojují bod dráhy nástroje s výchozím vrcholem řezu. Přerušovaná kružnice se středem v průsečíku dráhy P vyznačuje tvar nástroje v tomto bodě.



Obrázek 18: Křížící se dráha nástroje.

Detekce křížení je jednoduchá záležitost. Stačí projít všechny úsečky dráhy a pokud dojde k průniku dvou úseček, které nenásledují přímo po sobě, nalezneme bod překřížení – P . Z obrázku 18 je patrné, že při pohybu podél části dráhy (tučná čára), tvořené body PCD by došlo k nežádoucímu zanoření nástroje do obrobku. Je tedy třeba provést odstranění této části, které se provede následovně. Procházíme drahou až narazíme na křížení – bod P , jehož polohu si zapamatujeme. Následně z dráhy odstraňujeme všechny vrcholy, na které narazíme (C , D), až dojdeme opět do bodu křížení. Na místo odstraněných vrcholů vložíme průsečík P . Tím jsme odstranili nežádoucí oblast. Problém ovšem nastává v případě startu algoritmu nikoliv z bodu A ale například z bodu C . V takovém případě by došlo k odstranění platné části dráhy. Platná část dráhy je vždy definována ve stejném směru jako výchozí řez. Naopak neplatná část je určena opačným směrem. Tohoto lze využít k určení, kterou z částí odstranit. Směr určíme součtem vektorových součinů všech po sobě následujících úseček dané části. Pokud bude mít takto získaný výsledek stejné znaménko jako v případě polygonu řezu, mají oba mnohoúhelníky stejný směr a jedná se tedy o platnou část.

Výše popsaný algoritmus bude dobře pracovat v případě jednoduchého případu křížení. Na obrázku 19 je však naznačen složitější případ. Křížením zde vznikly 3 oblasti – A , B a C . Pokud bychom spustili výše popsaný algoritmus z části A , došlo by k odstranění neplatné části B , ale také platné části C . V takovýchto případech je třeba podle počtu samostatných platných oblastí vytvořit odpovídající počet kopií původní dráhy a spustit algoritmus se startem v každé z platných částí. Výsledkem bude odpovídající počet samostatných drah.



Obrázek 19: Složitější případ křížení dráhy nástroje.

Podobně je třeba zamyslet se nad případným křížením různých drah v rámci jedné roviny řezu. V práci [3] je doporučeno tento problém řešit jako průnik mnohoúhelníků. V případě protínajících se vnějších drah bude výsledkem obvod mnohoúhelníku vzniklého sloučením kolidujících mnohoúhelníků. V případě křížení vnitřní dráhy (tvořící dutinu v modelu) a vnější dráhy (tvořící výstupek v dutině) získáme výsledek jako obvod plochy vzniklé odečtením plochy mnohoúhelníku tvořícího vnější dráhu od plochy mnohoúhelníku určeného vnitřní drahou.

4 Implementace

Vzhledem k předpokládaným náročným výpočtům nad rozsáhlými daty byl jako efektivní programovací jazyk zvolen jazyk C++. Dalším důvodem pro tento výběr byla také dostupnost podpůrných knihoven pro práci s geometrickými daty a nástrojů pro vizualizaci výsledků. Celý systém pro generování dráhy nástroje na základě polygonálních modelů je implementován jako zásuvný modul aplikace OpenFlipper ve verzi 1.1 [11]. Operace nad základními geometrickými tvary zajišťuje knihovna CGAL.

Aplikace byla implementována a testována v operačním systému Windows 7 v 64bitové verzi. Aplikace nepoužívá žádné nepřenositelné funkce, proto by bez výrazných změn měla pracovat i na dalších systémech podporovaných aplikací OpenFlipper a knihovnou CGAL. Překlad ani správná funkčnost však v jiných operačních systémech testována nebyla.

4.1 OpenFlipper, OpenMesh

OpenFlipper je multiplatformní aplikace s otevřeným zdrojovým kódem v jazyce C++ navržená pro zpracování a vizualizaci geometrických dat. Je založena na frameworku Qt a pro rozšíření o vlastní funkce lze použít systém zásuvných modulů frameworku Qt. O datovou reprezentaci geometrických dat se stará knihovna OpenMesh [12]. Použitím aplikace OpenFlipper získáme prostředí pro vizualizaci výsledků s již hotovou podporou uživatelského rozhraní.

OpenMesh [12] je knihovna jazyka C++ zajišťující efektivní datovou reprezentaci polygonálních sítí a jednoduché rozhraní pro jejich procházení a manipulaci. Knihovna podporuje načítání modelů z několika datových formátů. Mezi nimi je také formát STL, o který nám jde především. Přestože OpenMesh podporuje obecně polygonální sítě, v této práci se vzhledem k omezení formátu STL budeme zabývat pouze trojúhelníkovými sítěmi. Topologie a geometrie polygonální sítě je v OpenMesh uložena pomocí struktury založené na polohranách (angl. halfedge), které vzniknou rozdělením hrany na dvě orientované části. Tato struktura umožňuje efektivní procházení sítě, tím že poskytuje přímé reference na nejbližší okolí. Podrobnější informace je možné dohledat v dokumentaci knihovny na webových stránkách [12].

4.2 CGAL - Computational Geometry Algorithms Library

CGAL je knihovna jazyka C++, která poskytuje komplexní algoritmy a datové struktury pro práci s různými geometrickými objekty. Přestože knihovna obsahuje rozsáhlé možnosti, využijeme z nich v podstatě jen funkce pro práci se základními objekty (body, přímky, trojúhelníky, ...). Reprezentace geometrie v prostředí počítače se potýká s omezenou přesností. Z tohoto důvodu poskytuje knihovna CGAL možnost použít knihoven pro přesné výpočty. To s sebou ovšem nese výrazné zpomalení výpočtů. Pokud si budeme omezené přesnosti vědomi, vystačíme si s dvojitou přesností čísel v pohyblivé řádové čárce. Podrobnější informace a dokumentaci je možné nalézt na webových stránkách knihovny CGAL [13]. Dobrým výchozím zdrojem informací při prvních krocích s touto knihovnou je také dokument [14].

4.3 Struktura zdrojových kódů

Veškeré zdrojové kódy řešeného projektu se nacházejí ve složce **Plugin-MillCNC**. Zbytek je tvořen zdrojovými kódy samotné aplikace OpenFlipper a několika základních zásuvných modulů. Z archivu aplikace OpenFlipper byla odstraněna většina modulů, které nesouvisí s řešeným tématem. Důvodem je zpřehlednění uživatelského rozhraní a také značná úspora času při překladu.

Zdrojový kód zásuvného modulu pro generování dráhy se skládá následujících souborů:

- **CMakeLists.txt** – Konfigurace projektu pro překladový systém CMake.
- **millcnc.hh** a **millcnc.cc** – Základní třída projektu. Zajišťuje komunikaci s aplikací OpenFlipper a vykreslování výsledků. Definiuje uživatelské rozhraní a obsahuje základní obecné algoritmy generování dráhy.
- **CGALcfg.hh** – Definice výpočetního jádra knihovny CGAL a typů používaných geometrických objektů.
- **MCgeom.hh** a **MCgeom.cc** – Funkce pro převod mezi geometrickými reprezentacemi knihoven OpenMesh a CGAL a balík funkcí pro manipulaci s geometrií, které nejsou poskytovány knihovnou CGAL.
- **MillTool.hh** a **MillTool.cc** – Definice obecné třídy nástroje a od ní odvozené třídy kulového a válcového nástroje. Zde jsou implementovány algoritmy pro generování dráhy, které závisí na tvaru nástroje.

- **ToolPath.hh** a **ToolPath.cc** – Třída reprezentující dráhu nástroje. Poskytuje funkce pro vykreslování dráhy, přístup k částem dráhy skrze jednotlivé vrcholy nebo úsečky. Zajišťuje konečný výstup ve formě g-kódu.

Uvedené třídy tvoří následující hierarchii, která je do systému aplikace OpenFlipper napojena skrze rozhraní BaseInterface.

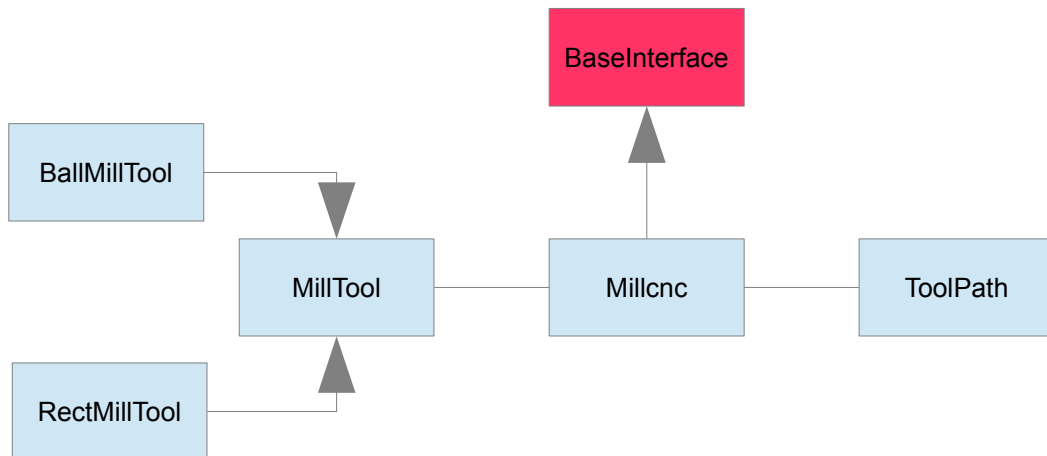


Diagram 1: Diagram tříd aplikace.

5 Výsledky

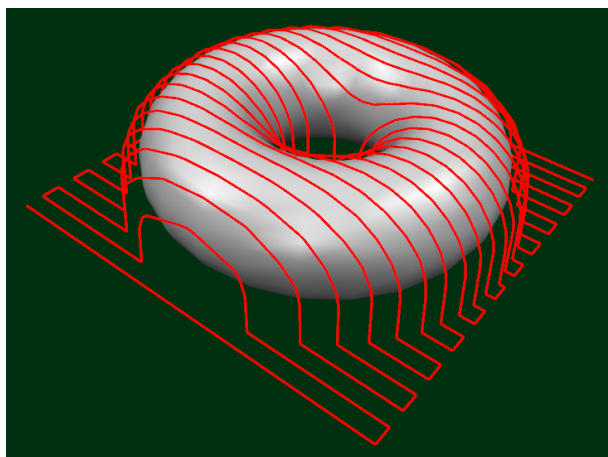
Horizontální a paralelní frézovací strategie byly implementovány samostatně a podle odlišných postupů. Z tohoto důvodu probereme samostatně také dosažené výsledky. Při hodnocení kvality výstupu je třeba si uvědomit, že cílem nebylo vytvořit profesionální nástroj k okamžitému použití. Přiložená aplikace slouží spíše k demonstraci popsaných postupů a tvoří základ pro případnou implementaci pokročilejších algoritmů.

Než se dostaneme k popisu výsledků implementace obou strategií uvedeme některé společné znaky. Implementace obou strategií se vyznačují poměrně dlouhou dobou generování dráhy. Přičemž doba generování roste v závislosti na složitosti modelu (počtu polygonů) a požadované přesnosti (délce kroku, vzdálenosti mezi rovinami). Je zde poměrně dost prostoru pro případné optimalizace, efektivita však nebyla klíčovým požadavkem, proto k ní nebylo soustředěno více úsilí. Další společnou vlastností je velká délka vygenerovaného programu pro NC stroj. Strategie totiž často generují body dráhy, které leží na jedné přímce a daly by se vypustit. To značně zvyšuje velikost výsledného souboru a může také znamenat pomalejší zpracování na straně obráběcího stroje.

Grafické příklady vygenerovaných drah nástroje naleznete v příloze. V této kapitole budou uvedeny a rozebrány pouze některé speciální případy.

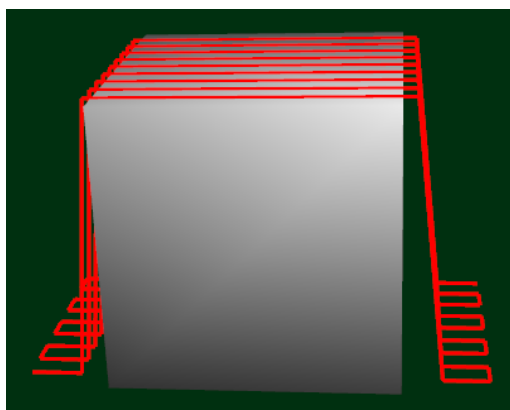
5.1 Paralelní strategie

Pokud odhlédneme od výpočetní náročnosti, výsledky implementované paralelní strategie odpovídají předpokladům. Paralelní dokončování i odvozené hrubování dává konzistentní výsledky i při generování dráhy na základě modelů složitých tvarů. Použit je možné bez omezení jak kulové tak válcové nástroje. Vezmeme-li v úvahu vlastnosti dané pevně danou vzdáleností rovin a pevně daným krokem, se kterým se vypočítává dráha v rámci roviny, můžeme považovat obě paralelní strategie za reálně použitelné.



Obrázek 20: Paralelní strategie – model s nedostupnými částmi.

Z obrázku 20 je zřejmé, že algoritmus z principu zvládá dobře i generování dráhy pro modely, kde je část modelu ve stínu jiné části. V tomto případě se jedná spodní polovinu toroidu. Pokud bychom použili postupy generování vycházející z řezu modelu, byl by problém mimo jiné s řešením právě takovýchto případů, kdy je část modelu zakryta jinou částí.



Obrázek 21: Paralelní strategie – vliv dlouhého kroku na svislé plochy.

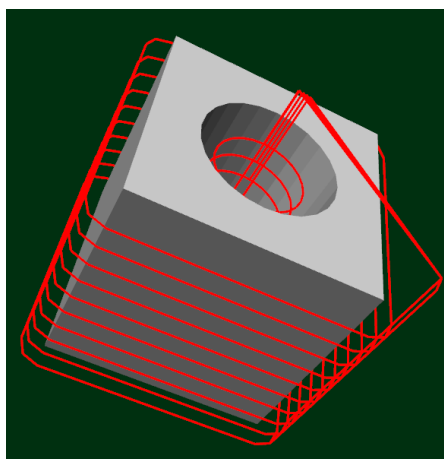
Na obrázku 21, kde je jako model uvažována jednoduchá krychle, je jasně patrná nepříjemná vlastnost generování dráhy s větší délkou konstantního kroku. A sice, že nikdy nedostaneme dokonale svislé části trajektorie nástroje a to ani podél svislých ploch. Chybu je sice možné omezit volbou kratšího kroku, ale nikdy nelze zcela eliminovat. Možným řešením je návrh systému, který by v průběhu generování detekoval problémové úseky a podle okolností upravoval délku kroku.

5.2 Horizontální strategie

Popsaný postup generování dráhy pro horizontální strategii se ukázal jako nepříliš dokonalý. U jednoduchých modelů sice produkuje správné a odpovídající výsledky. V případech složitějších modelů však přestávají být výsledky konzistentní a pro reálné nasazení by bylo třeba několik dodatečných úprav. Stejně jako v případě paralelní strategie je i implementace horizontální strategie poměrně výpočetně náročná. Vzhledem ke složitějším algoritmům je zde však tato vlastnost výraznější.

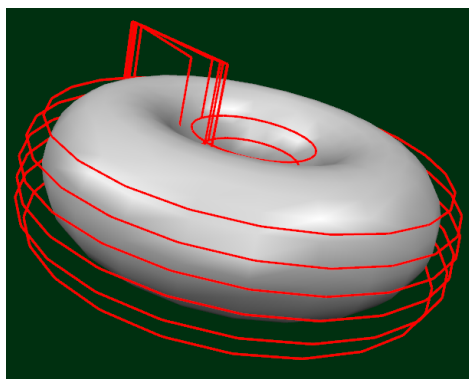
Protože se nezdařilo vytvořit dostatečně robustní implementaci generování dráhy v jedné samostatné horizontální rovině, nebylo spojení drah z jednotlivých rovin věnováno větší úsilí a je provedeno pouze primitivním a neefektivním způsobem. Na konci dráhy v jedné rovině dochází ke zdvihu nástroje nad nejvyšší úroveň modelu a přesun nad počáteční bod dráhy další v rovině. Zde dojde ke spuštění nástroje a dál se pokračuje podél vygenerované dráhy.

Křížení v rámci jedné dráhy jsou řešena podle popsaného postupu, který pracuje podle očekávání. Namísto řešení možného křížení více drah v jedné rovině byla práce soustředěna na odladění generování dráhy samotné, které se však nepodařilo dotáhnout do zdárného konce. Proto krok detekce a odstranění křížení více drah v rámci jedné roviny v implementaci zcela chybí.



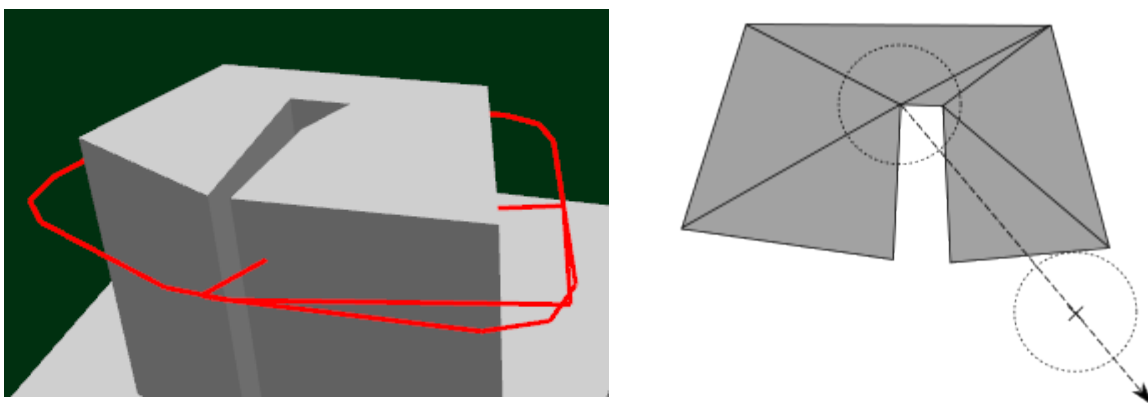
Obrázek 22: Výsledek horizontální strategie pro jednoduché tvary.

Na obrázku 22 je výsledná trajektorie pro jednoduchý model. Je zde patrné, že dráha nástroje odpovídá očekávání. Algoritmus si dobře poradí i s více samostatnými drahami v rámci jedné roviny (podél vnější krychlové části i podél vnitřní válcové části).



Obrázek 23: Horizontální strategie – problém zakrytých částí modelu.

Na obrázku 23 je patrná nepříjemná vlastnost generování dráhy na základě řezu modelu. Dochází zde k nesprávnému určení dráhy pro části modelu, které jsou překryty jinou částí (v tomto případě se jedná o spodní část toroidu). Při frézování podle dráhy ve spodní části by došlo k poškození tvaru modelu ve vyšších úrovních.



Obrázek 24: Horizontální strategie – problém úzkých dutin.

Jedním z problémů, se kterými si algoritmus generování dráhy v horizontální rovině již z principu neporadí jsou dutiny užší než je průměr nástroje (obr. 24). Tento problém si popíšeme na pravé části tohoto obrázku. Je zde vyobrazen výpočet bodu trajektorie pro vrchol řezu uvnitř dutiny. Podle [3] hledáme nejvzdálenější dotyk nástroje s množinou trojúhelníků, které alespoň z části leží ve stínu nástroje umístěného na pozici výchozího vrcholu. V tomto případě se jedná o všechny trojúhelníky horní plochy. Vypočtený bod trajektorie leží zcela mimo dutinu. Výsledkem je neplatná část dráhy, která je patrná v levé části obrázku.

6 Závěr

Cílem této bakalářské práce bylo navrhnout a implementovat jednoduchý generátor dráhy nástroje pro tříosé CNC frézování na základě 3D polygonálních modelů. Vybrány byly jednoduché strategie pro paralelní a horizontální frézování. První část práce se věnuje základnímu přehledu problematiky obrábění a jeho numerického řízení. Jádro práce popisuje navržené postupy generování dráhy nástroje a řešení souvisejících problémů. V poslední části se nachází informace o implementaci a použitých knihovnách a zhodnocení dosažených výsledků.

Práce vychází především z publikovaných algoritmů. K řešení dílčích problémů však bylo často přistupováno originálním způsobem. Modifikována byla hlavně řešení geometrických problémů dotyku nástroje a různých částí trojúhelníku. Důvodem bylo omezení možných chyb použitím prověřených funkcí knihovny CGAL a možnost použití jednoho postupu pro obě uvažované strategie.

Problematika generování dráhy nástroje v prostoru se ukázala být poměrně složitým tématem, především v porovnání s generováním dráhy pro vyřezávání plošných tvarů, se kterým má autor dřívější zkušenost. Hned na začátku bylo jasné, že výsledný produkt nebude moci konkurovat komerčně dostupným CAM systémům. Přiložená aplikace tak především demonstruje vlastnosti vybraných postupů generování dráhy a nesnaží být v praxi okamžitě použitelným produktem.

Výsledná dráha nástroje generovaná pomocí paralelní strategie splnila očekávání a pracuje podle předpokladů i v případě složitějších modelů. Pokud vezmeme v úvahu vlastnosti dané generováním dráhy s konstantním krokem a konstantní vzdáleností rovin řezu, můžeme tuto strategii pokládat ze reálně použitelnou. Naproti tomu postup použitý při generování dráhy horizontální strategie se ukázal být méně robustní, přičemž zjištěné problémy jsou uvedeny v předcházející kapitole. Byť základní myšlenka není zcela mylná, dotažení postupu do funkční podoby by vyžadovalo větší množství další práce.

Přestože implementace paralelní strategie dává poměrně dobré výsledky, výrazného zlepšení by mohlo být dosaženo doplněním určování vhodné délky kroku v rámci roviny podle aktuálního tvaru modelu. Prioritou práce na horizontální strategii frézování by v případě dalšího pokračování projektu mělo být dotažení algoritmu generování dráhy v horizontální rovině do funkční podoby i pro složitější modely. Dalším krokem by mělo být vytvoření odvození dráhy pro hrubování. Celkově lepších výsledků horizontálního dokončování by po doladění generování dráhy v rámci roviny mohlo být dosaženo adaptivním určování vzdálenosti jednotlivých hladin řezu podle aktuálního tvaru modelu. Pokud bychom požadovali reálně použitelný produkt, byla by také nezbytná celková optimalizace implementace a výsledného programu pro CNC stroj.

Literatura

- [1] KOCMAN, Karel. *Technologie obrábění*. Brno: Akademické nakladatelství CERM s.r.o., 2001. ISBN 80-214-1996-2.
- [2] VLACH, Bohumil. *Technologie obrábění a montáží*. 1. vyd. Praha: SNTL - Nakladatelství technické literatury, Redakce strojírenské a metalurgické literatury. 1990. ISBN 80-030-0143-9.
- [3] CHUANG, Chen-Ming a Hong-Tzong YAU. A new approach to z-level contour machining of triangulated surface models using fillet endmills. *Computer-Aided Design*. 2005, roč. 37, č. 10, s. 1039-1051. ISSN 00104485. DOI: 10.1016/j.cad.2004.10.005.
- [4] *LinuxCNC: Software for realtime control* [online]. [cit. 2012-04-28]. Dostupné z: <http://www.linuxcnc.org/>
- [5] SAITO, Takafumi a Tokiichiro TAKAHASHI. NC machining with G-buffer method. *ACM SIGGRAPH Computer Graphics*. 1991, roč. 25, č. 4, s. 207-216. ISSN 00978930. DOI: 10.1145/127719.122741.
- [6] CARTER, Jeremy A., Thomas M. TUCKER a Thomas R. KURFESS. 3-Axis CNC Path Planning Using Depth Buffer and Fragment Shader. *Computer-Aided Design and Applications*. 2008, roč. 5, č. 5, s. 612-621. ISSN 16864360. DOI: 10.3722/cadaps.2008.612-621.
- [7] INUI, Masatomo. Fast inverse offset computation using polygon rendering hardware. *Computer-Aided Design*. 2003, roč. 35, č. 2, s. 191-201. ISSN 00104485. DOI: 10.1016/S0010-4485(02)00052-0.
- [8] HWANG, Ji Seon a Tien-Chien CHANG. Three-axis machining of compound surfaces using flat and filleted endmills. *Computer-Aided Design*. 1998, roč. 30, č. 8, s. 641-647. ISSN 00104485. DOI: 10.1016/S0010-4485(98)00021-9.
- [9] CHUANG, C.-M., C.-Y. CHEN a H.-T. YAU. A Reverse Engineering Approach to Generating Interference-Free Tool Paths in Three-Axis Machining from Scanned Data of Physical Models. *International Journal of Advanced Manufacturing Technology*. 2002, roč. 19, č. 1, s. 23-31. ISSN 0268-3768. DOI: 10.1007/PL00003965.
- [10] HUANG, S.-H., L.-C. ZHANG a M. HAN. An Effective Error-Tolerance Slicing Algorithm for STL Files. *The International Journal of Advanced Manufacturing Technology*. 2002, roč. 20, č. 5, s. 363-367. ISSN 0268-3768. DOI: 10.1007/s001700200164.
- [11] *OpenFlipper - An Open Source Geometry Processing and Rendering Framework* [online]. [cit. 2012-04-28]. Dostupné z: <http://www.openflipper.org/>
- [12] *OpenMesh* [online]. [cit. 2012-04-28]. Dostupné z: <http://www.openmesh.org/>
- [13] *CGAL - Computational Geometry Algorithms Library* [online]. [cit. 2012-04-28]. Dostupné z: <http://www.cgal.org/>
- [14] GIEZEMAN, Geert-Jan, Remco VELTKAMP a Wieger WESSELINK. *Getting started with CGAL* [online]. Utrecht University. 1999. Dostupné z: http://graphics.stanford.edu/courses/cs368-00-spring/manuals/CGAL_Tutorial.pdf
- [15] CHUA, Chee Kai, Kah Fai LEONG a Chu Sing LIM. *Rapid prototyping: principles and applications*. 2nd ed. New Jersey: World Scientific Publishing Co. Pte. Ltd., 2003. ISBN 98-123-8120-1.

Seznam příloh

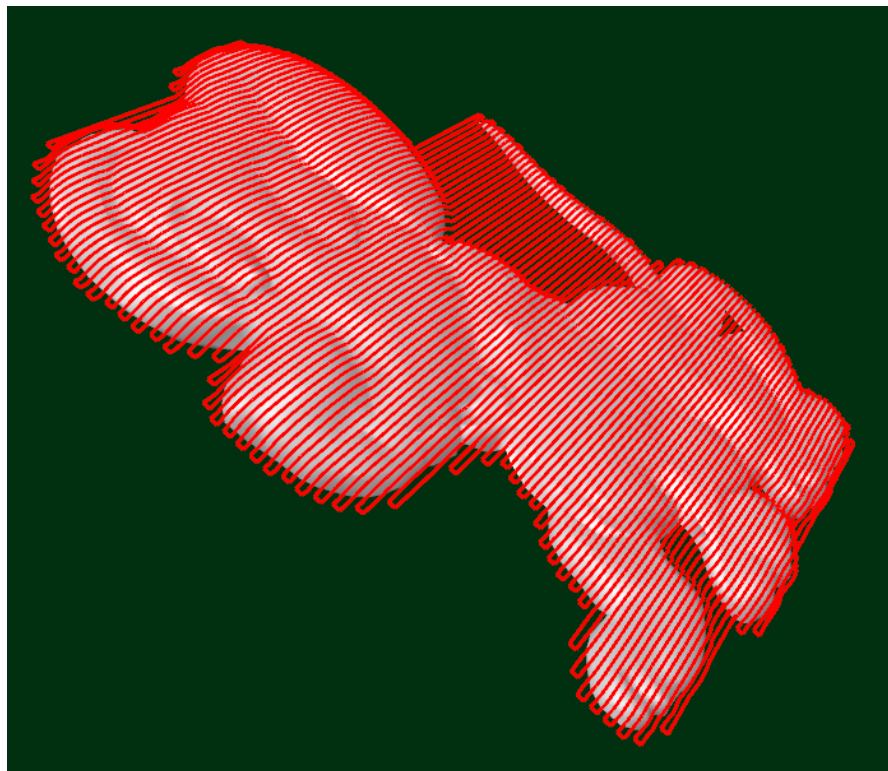
Příloha 1. Ukázky vygenerovaných drah

Příloha 2. Návod k přiložené aplikaci

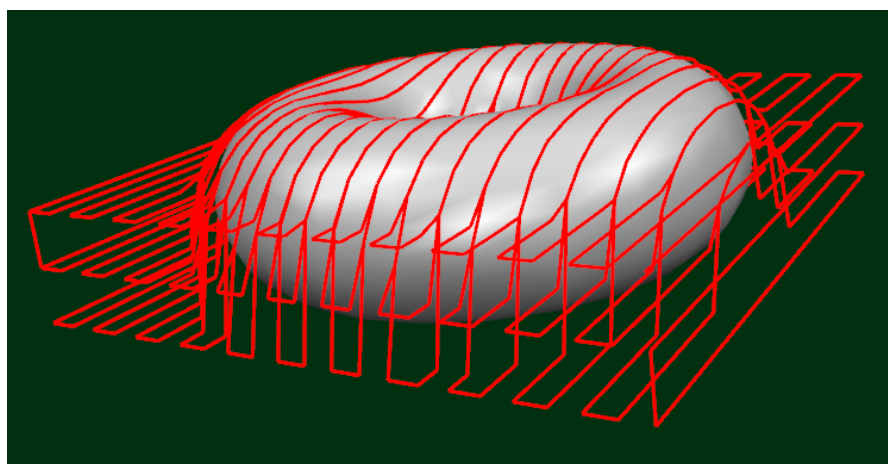
Příloha 3. Nastavení prostředí pro překlad

Příloha 4. CD se zdrojovými texty a binární podobou aplikace

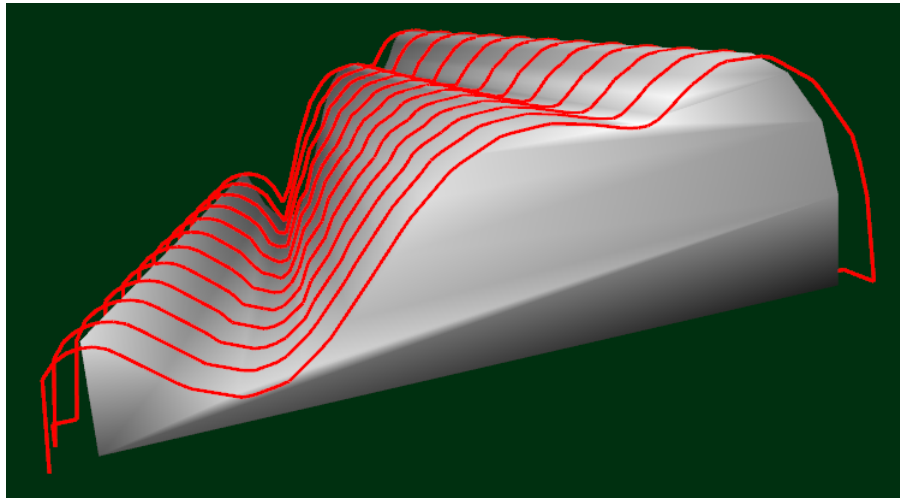
Příloha 1. Ukázky vygenerovaných drah



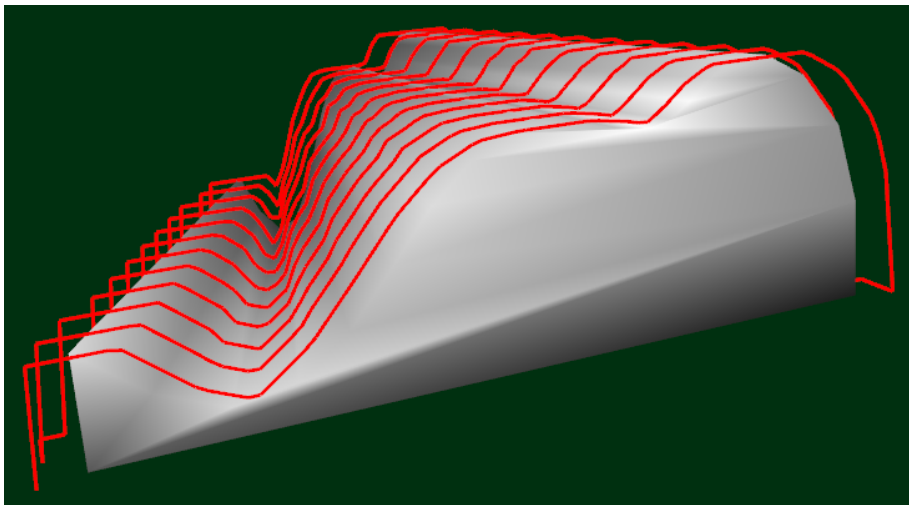
little_dog.stl (zdroj: <http://www.vectorclip3d.com/>): 100x110x8 mm, Parallel finish, Flat Mill 2 mm, Planes spacing 1 mm, Step size 1 mm.



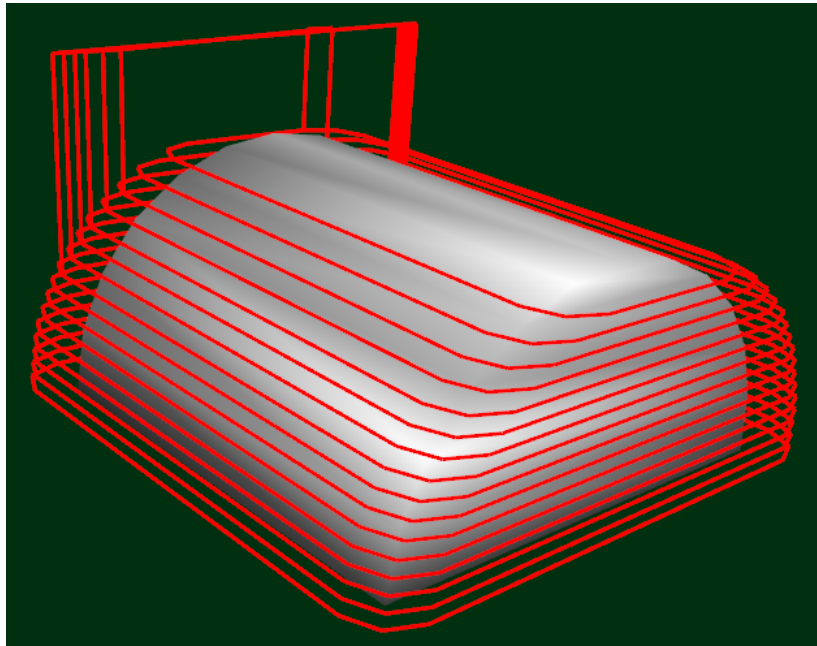
toroid.stl: 10x10x6 mm, Parallel rough, Ball Mill 2 mm, Planes spacing 1 mm, Step size 1 mm, Max cut 2 mm.



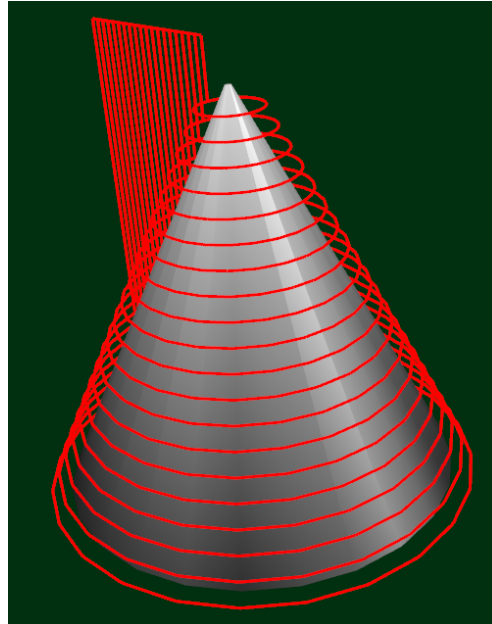
waves.stl: 15x15x5 mm, Parralel finish, Ball mill 1,5 mm, Planes spacing 1 mm, Step size 0,2 mm.



waves.stl: 15x15x5 mm, Parralel finish, Flat mill 1,5 mm, Planes spacing 1 mm, Step size 0,2 mm.



***halfcylinder.stl:** 15x15x5 mm, Horizontal finish, Flat mill 2 mm, Planes spacing 0,3 mm, Step size 1 mm.*

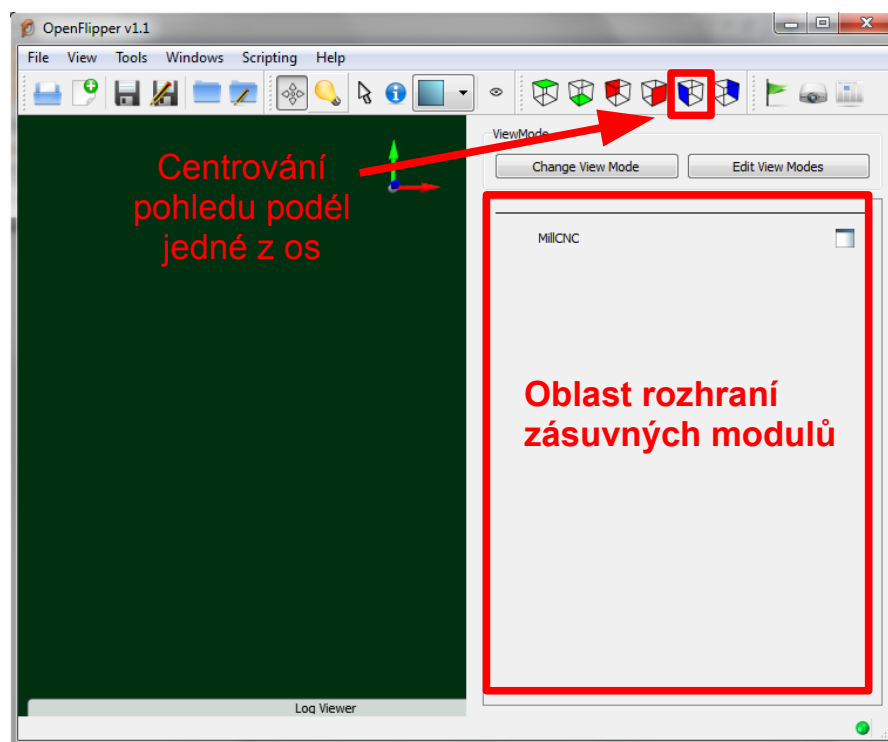


***cone.stl:** 20x20x20 mm, Horizontal finish, Flat mill 2 mm, Planes spacing 1 mm, Step size 1 mm.*

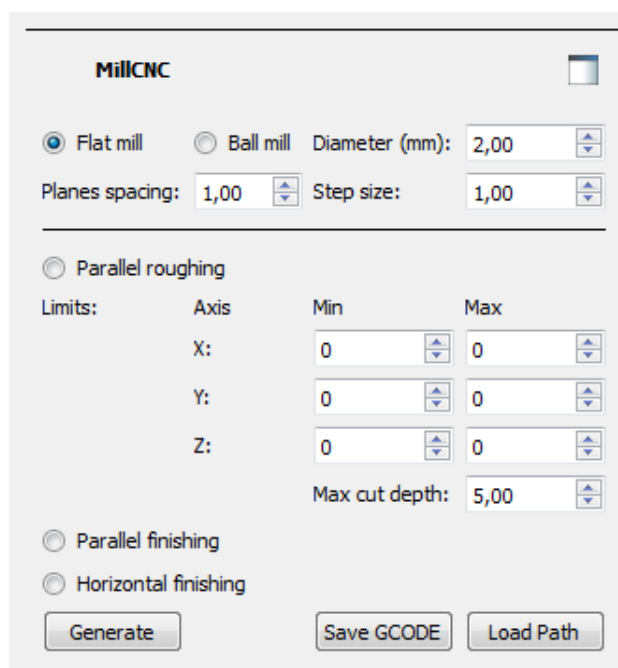
Příloha 2. Návod k přiložené aplikaci

Přiložená aplikace v binární podobě vyžaduje pro spuštění balíček *Microsoft Visual C++ 2008 Redistributable*. V případě, že ve vašem počítači tento software chybí, nainstalujte jej z přiloženého souboru `vc_redist_x86.exe`.

Po spuštění programu (`OpenFlipper.exe`) se zobrazí následující okno (obrázek 1). V horní části se nachází panel nástrojů. Dominující zelená oblast slouží pro vykreslování modelu. Model do aplikace načtete příkazem *Load Object* z nabídky *File*. V případě, že je v aplikaci již nějaký model nahrán je nutné jej nejprve příkazem *File* → *Clear All* odstranit. Pokud model po načtení nezobrazí správně, vycentrujte pohled příkazem *View* → *View all* nebo kliknutím na níže vyznačenou ikonu v panelu nástrojů. V pravé části je červenou barvou vyznačena oblast rozhraní zásuvných modulů. Kliknutím na položku *MillCNC* zobrazíte volby pro nastavení parametrů generování dráhy (obrázek 2).



Obrázek 1: Hlavní okno aplikace.



Obrázek 2: Rozhraní pro nastavení parametrů generátoru.

Na obrázku 2 je zobrazeno rozhraní zásuvného modulu. Všechny uvedené délky jsou v milimetrech. V horní části rozhraní je nabídnuta volba tvaru nástroje a jeho poloměr. Těsně nad vodorovným oddělovačem se nachází nastavení základních parametrů, které jsou společné všem strategiím. Hodnota *Planes spacing* určuje vzdálenost mezi rovinami řezu a hodnota v poli *Step size* představuje délku kroku, se kterým se generuje dráha v rámci roviny.

Přepínače v prostřední části pod oddělovačem slouží k volbě strategie. V případě volby paralelního hrubování je třeba zadat meze hrubovaného regionu v části *Limits* a hloubku frézování v jednom průchodu v poli *Max cut depth*.

Po nastavení všech parametrů vygenerujete dráhu nástroje kliknutím na tlačítko *Generate*. Výpočet dráhy může podle složitosti modelu a požadované přesnosti, dané hodnotami *Planes spacing* a *Step size*, trvat až několik minut. Po dokončení se v prostoru modelu zobrazí náhled dráhy. Vygenerovanou dráhu můžete uložit kliknutím na tlačítko *Save GCODE*. Výsledkem bude program kompatibilní se systémem LinuxCNC. Dříve vygenerovanou a uloženou dráhu můžete načíst kliknutím na tlačítko *Load Path*.

Příloha 3. Nastavení prostředí pro překlad

Následující návod je z větší části pouze přeloženým výtahem z kapitoly „Building OpenFlipper“ z vývojářské dokumentace aplikace OpenFlipper (<http://openflipper.org/Documentation/OpenFlipper-1.1/buildingOpenFlipper.html>).

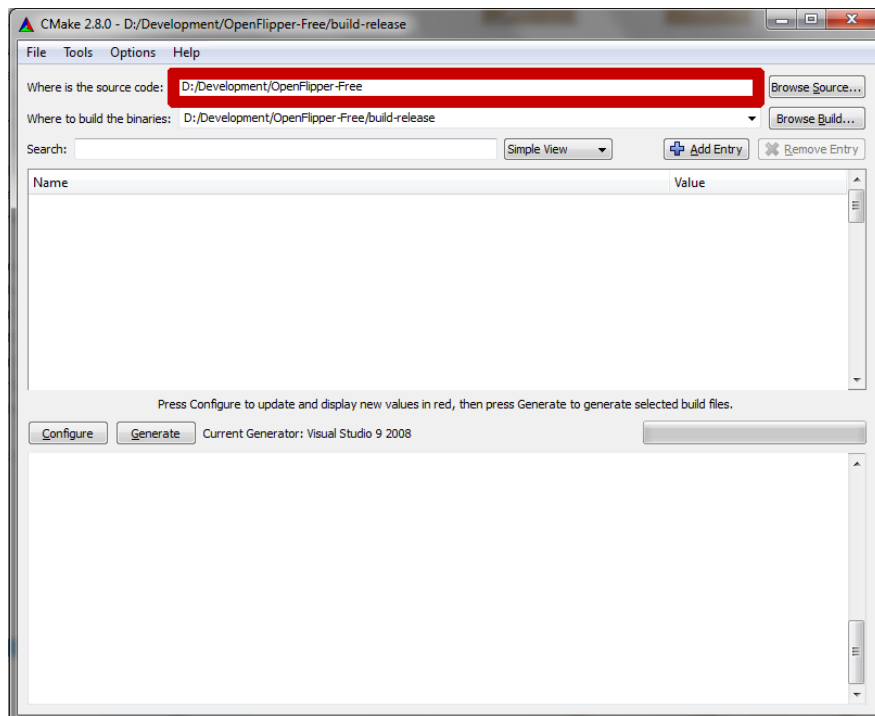
Potřebné knihovny a software

- Qt \geq 4.7.3 (<http://qt.nokia.com/downloads>)
- GLUT (<http://www.opengl.org/resources/libraries/glut/>)
- GLEW \geq 1.6 (<http://glew.sourceforge.net>)
- CMake \geq 2.8.4 (<http://www.cmake.org/cmake/resources/software.html>)

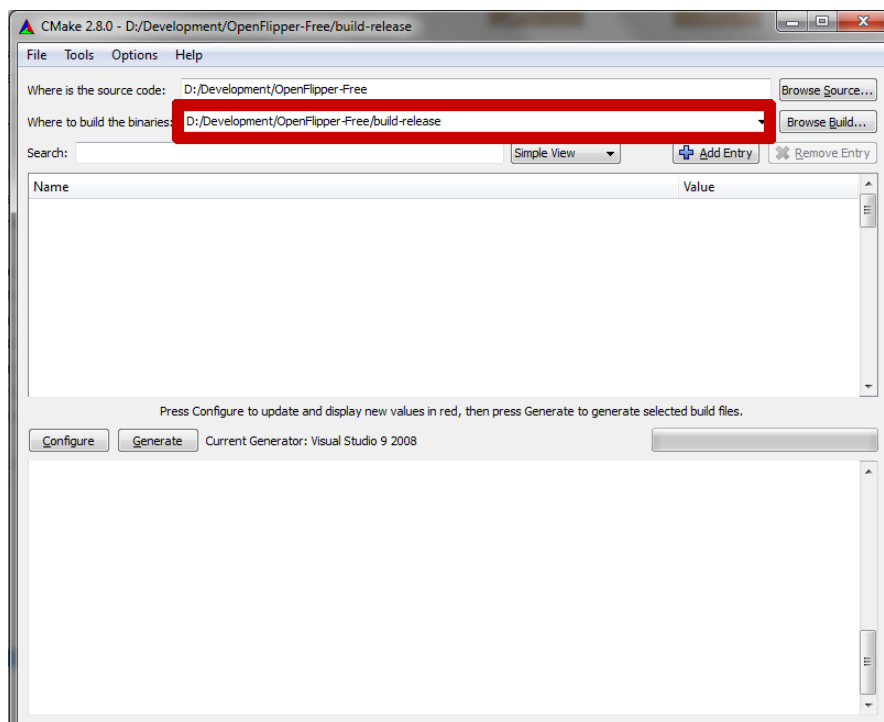
Překlad aplikace OpenFlipper ve Windows

- Nainstalujte Visual Studio 2008 (jiná verze není pro překlad uvedené verze aplikace podporována).
- Nainstalujte framework Qt 4.7.3 nebo novější ve verzi pro Visual Studio.
- Nastavte proměnnou prostředí s názvem QTDIR na adresář instalace Qt.
- Připravte si zdrojové kódy projektu.
- Stáhněte knihovnu glew a nainstalujte ji, nejlépe do `c:\libs\glew`. Složka `c:\libs\glew\lib` by měla obsahovat soubory DLL a LIB. Hlavičkové soubory by měly být umístěny v `c:\libs\glew\include\gl`.
- Přesuňte soubory ve složce `glut-3.7` tak, aby všechny soubory LIB a DLL byly ve umístěny ve složce `c:\libs\glut-3.7\lib` a soubor `glut.h` v `c:\libs\glut-3.7\include\gl\glut.h`.
- Nainstalujte systém CMake pro windows.

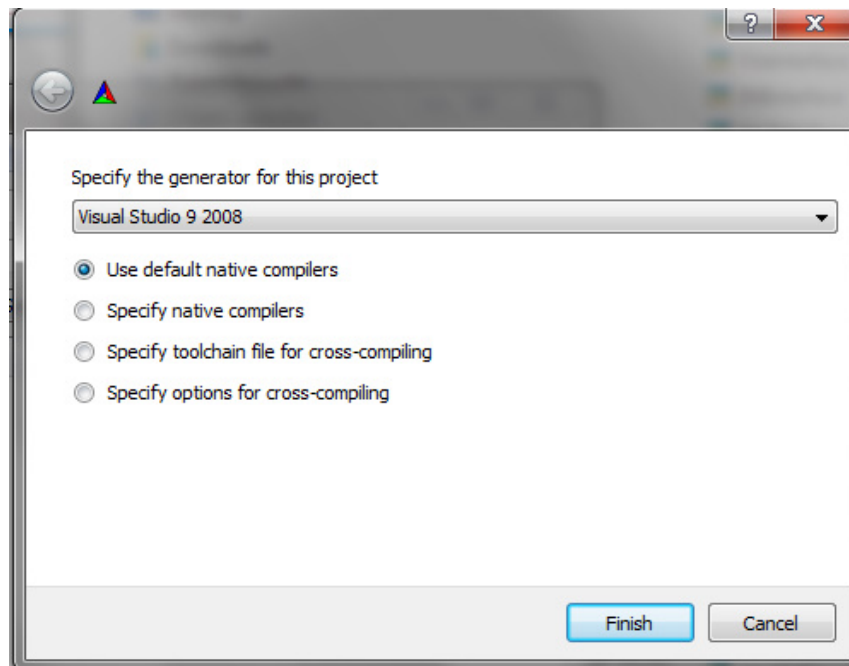
- Spust'te grafickou nastavbu CMake (cmake-gui) a jako umist'eni zdrojov'ych k'od' vyberte kořenový adresář zdrojov'ych k'od' projektu.



- Vyberte libovolný adresář pro umístění překladačového prostředí.



- Klikněte na tlačítko Generate.
- Vyberte používané překladové prostředí (Visual Studio 2008).



- Pokud konfigurace selže z důvodu chybějících knihoven nebo hlavičkových souborů, bude ve spodní části vypsáno chybové hlášení. Cesty můžete v cmake-gui upravit ručně. Některé z proměnných jsou všach zobrazeny pouze v pokročilém režimu zobrazení (Advanced Mode). Režimy zobrazení můžete měnit pomocí voleb v horní části okna.
- Znovu klikněte na tlačítko configure, dokud nebudou zobrazeny žádné červeně zvýrazněné řádky nebo chyby.
- Kliknutím na tlačítko Generate vytvoříte projektové soubory pro Visual Studio
- Ve zvoleném adresáři překladového prostředí bude vytvořen soubor Visual Studio solution (OpenFlipper.sln).
- Nyní je po otevření souboru OpenFlipper.sln možné přeložit aplikaci. Zásuvný modul generování dráhy však navíc používá knihovnu CGAL. Je tedy nutné projekt upravit tak, aby ji správně používal.
- CGAL používá knihovnu Boost, kterou nainstalujete z příloženého souboru boost_1_47_setup.exe.
- Ve Visual Studiu zkontrolujte nastavení cest v Tools → Options → Projects and Solutions → VC++ Directories. Cesty se mohou lišit v závislosti na vašem prostředí.
 - include: <Boost> (C:\Program Files\boost\boost_1_47)
 - include: <CGAL>\include (C:\Program Files\CGAL-3.7\include)

- include: <CGAL>\auxiliary\gmp\include (C:\Program Files\CGAL-3.7\auxiliary\gmp\include)
 - include: <QT>\include (C:\Qt\4.7.1\include)
 - include: <QT>\include\QtCore (C:\Qt\4.7.1\include\QtCore)
 - include: <QT>\include\QtGui (C:\Qt\4.7.1\include\QtGui)

 - library: <CGAL>\lib (C:\Program Files\CGAL-3.7\lib)
 - library: <QT>\lib (C:\Qt\4.7.1\lib)
 - library: <Boost>\lib (C:\Program Files\boost\boost_1_44\lib)
 - library: <CGAL>\auxiliary\gmp\lib (C:\Program Files\CGAL-3.7\auxiliary\gmp\lib)
- V nastavení projektu MillCNC přidejte knihovny libgmp-10.lib a libmpfr-4.lib a zakažte knihovny gmp-vc90-mt-gd.lib a mpfr-vc90-mt-gd.lib (v případě konfigurace release potom gmp-vc90-mt.lib a mpfr-vc90-mt.lib).

