

České vysoké učení technické v Praze
Fakulta stavební
Katedra mechaniky



Procesor G-code pro simulaci 3D tisku

Diplomová práce

Bc. Jan Voříšek

Studijní program: Stavební inženýrství
Obor studia: Konstrukce a dopravní stavby
Vedoucí práce: Prof. Dr. Ing. Bořek Patzák

Praha, 2019



ZADÁNÍ DIPLOMOVÉ PRÁCE

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: Voříšek	Jméno: Jan	Osobní číslo: 423675
Zadávací katedra: katedra mechaniky, K132		
Studijní program: Stavební inženýrství		
Studijní obor: Konstrukce a dopravní stavby		

II. ÚDAJE K DIPLOMOVÉ PRÁCI

Název diplomové práce: Procesor G-code pro simulaci 3D tisku	
Název diplomové práce anglicky: G-code processor for simulation of 3D printing	
Pokyny pro vypracování: Na základě specifikace jazyka G-code používaného jako řídicí kód 3D tiskáren vypracovat softwarový preprocesor, který bude generovat vstup vhodný pro následnou numerickou simulaci výrobního procesu.	
Seznam doporučené literatury: G-code specification, https://reprap.org/wiki/G-code , 2018	
Jméno vedoucího diplomové práce: Prof. Dr. Ing. Bořek Patzák	
Datum zadání diplomové práce: 3.10.2018	Termín odevzdání diplomové práce: 6.1.2019 <i>Údaj uveďte v souladu s datem v časovém plánu příslušného ak. roku</i>
Podpis vedoucího práce	Podpis vedoucího katedry

III. PŘEVZETÍ ZADÁNÍ

Beru na vědomí, že jsem povinen vypracovat diplomovou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací. Seznam použité literatury, jiných pramenů a jmen konzultantů je nutné uvést v diplomové práci a při citování postupovat v souladu s metodickou příručkou ČVUT „Jak psát vysokoškolské závěrečné práce“ a metodickým pokynem ČVUT „O dodržování etických principů při přípravě vysokoškolských závěrečných prací“.

Datum převzetí zadání	Podpis studenta(ky)
-----------------------	---------------------

Prohlášení

Prohlašuji, že jsem svou diplomovou práci napsal samostatně a výhradně s použitím citovaných pramenů. Souhlasím se zapůjčováním práce a jejím zveřejňováním.

V Praze dne 6. ledna 2019

.....
Bc. Jan Voříšek

Poděkování

Na tomto místě bych rád poděkoval Prof. Dr. Ing. Bořkovi Patzákovi za jeho odborné rady, připomínky a trpělivost během konzultací. Mé poděkování samozřejmě patří i rodině, přátelům a všem ostatním, kteří mě po celou dobu studia podporovali.

Název práce: Procesor G-code pro simulaci 3D tisku

Autor: Bc. Jan Voříšek

Katedra: Katedra mechaniky

Vedoucí diplomové práce: Prof. Dr. Ing. Bořek Patzák

Abstrakt Tato diplomová práce se zabývá vytvořením nástroje, který umožňuje tvorbu modelu pro numerickou simulaci výrobního procesu 3D tisku. Práce popisuje princip 3D tisku, používané technologie a výrobní proces. Základem vytvořeného nástroje je emulátor 3D tisku nad digitálním rastrem tvořeným z prostorových krychlí o volitelném rozlišení a volitelné délce časového kroku. Interpretací vstupního souboru pro tiskárnu (ve formátu G-code) je zjištěn čas a míra vyplnění tiskovým materiálem jednotlivých krychlí rastrového modelu.

Klíčová slova: 3D tisk, G-code, emulace, MKP, model

Title: G-code processor for simulation of 3D printing

Author: Bc. Jan Voříšek

Department: Department of Mechanics

Supervisor: Prof. Dr. Ing. Bořek Patzák

Abstract The main objective of this thesis is to develop the software emulator of a 3D printer facilitating the creation of a numerical model for simulating the printing process. The thesis describes the principle of 3D printing, current technologies and overall production process. The emulator is based on a discrete voxel model with variable edge length and variable time step. By gradual processing of a 3D printer input file (in a G-code format) the volume of deposited material is determined for each voxel as a function of time.

Keywords: 3D printing, G-code, emulation, FEM, model

Obsah

Abstrakt	ix
1 Úvod	1
1.1 Motivace	1
1.2 Cíle práce	2
2 3D tisk	3
2.1 Výrobní proces	4
2.1.1 Modelování	5
2.1.1.1 Dostupný software	5
2.1.1.2 Výstupní formáty	6
2.1.2 Slicing	7
2.1.3 Tisk	8
2.1.4 Post-processing	8
2.2 Přehled technologií	8
2.2.1 Stereolitografie	9
2.2.2 Selective laser sintering	10
2.2.3 Fused deposition modelling	11
2.2.3.1 Materiály	11
2.3 G-code	12
2.3.1 Pohybové příkazy	12
2.3.1.1 Nejpoužívanější příkazy	13
2.3.2 Speciální příkazy	13
2.4 Problémy	13
2.4.1 Smršťování	14
2.4.2 Přehřívání	14
3 Procesor G-code	15
3.1 Vstupní zpracování	16
3.2 Zpracování jednotlivých příkazů	16
3.2.1 Pohybové příkazy	17
3.2.1.1 Délka trvání pohybu	18
3.2.1.2 Výpočet šířky depozice	19
3.2.1.3 Výpočet informací o vrstvách	19
3.3 Aktivace kostek v časových krocích	20
3.3.1 Přidělení objemu elementům rastrového modelu	20
3.3.1.1 Určení půdorysných souřadnic depozice	20
3.3.1.2 Určení elementů pro vyplnění	21
3.3.1.3 Průnik vytištěného objemu s kostkami modelu	22
3.3.1.4 Výpočet plochy vytištěného obrazce	24
3.3.1.5 Rozdělení objemu dle výšky a tloušťky vrstvy	25
3.3.1.6 Překryv navazujících depozic	25
3.3.2 Přerozdělení objemu u přeplněných prvků	27
3.4 Výstup vhodný pro MKP software	28
3.4.1 Výstupní formát JSON	28
3.4.2 Formátování uzlů	29

3.4.3	Formátování prvků	29
4	Příklady	31
4.1	Tiskárna Creality Ender 3	32
4.2	Příklad 1: Krychle 20 × 20 × 20 mm	33
4.2.1	Postupná aktivace elementů	34
4.3	Příklad 2: Krychle 20 × 20 × 20 mm řídce vyplněná	36
4.3.1	Postupná aktivace elementů	37
4.4	Příklad 3: Žába	39
4.4.1	Postupná aktivace elementů	40
5	Závěr	43
5.1	Motivace pro další výzkum	43
	Literatura	45
	Seznam obrázků	47
	Seznam tabulek	49
	Seznam algoritmů	51
	Přílohy	53
A	Obsah přiloženého CD	55
A.1	Spuštění aplikace	55
A.2	Sestavení aplikace	55

1 Úvod

Technologie 3D tisku umožňuje vytvořit reálný objekt na základě prostorového počítačového modelu. Jedná se o proces tzv. aditivní výroby, kdy je objekt realizován postupně po jednotlivých tenkých vrstvách. Narozdíl od tradiční substraktivní výroby (např. obrábění kovů) jde o relativně novou technologii, jejíž historie sahá do 80. let minulého století. V případě použití plastu jako tiskového materiálu se jedná o přímou konkurenci vstřikování plastů do forem.

Historicky první technologií 3D tisku je tzv. stereolitografie patentovaná v roce 1984. Tato technologie, původně určená pro rychlou tvorbu prototypů, si rychle našla i další uplatnění např. ve výrobě speciálních zubních rovnátek na míru [1]. V dnešní době je nejpožívanější technologií tzv. *fused deposition modelling* (FDM)¹. V roce 2005 byl založen otevřený projekt 3D tiskárny – RepRap (Replication Rapid Prototyper) [2]. Myšlenkou bylo vytvořit tiskárnu, která dokáže vytisknout co největší množství vlastních dílů. Součástky, které tiskem vyrobit nelze (šrouby, motory aj.) jsou levné a běžně dostupné všude na světě. Tento projekt svou otevřeností umožnil další rozvoj komerčního i nekomerčního 3D tisku. Zejména komerčnímu rozvoji napomohlo též vypršení patentů na FDM v roce 2009. Tiskárna pro domácí použití se dnes dá pořídit již za několik tisíc korun. Původní myšlenka tvorby prototypů zůstává, ale výsledkem 3D tisku jsou dnes i plnohodnotné součástky běžně používané v automobilovém, leteckém i strojném průmyslu [3].

Celý proces 3D tisku začíná tvorbou prostorového modelu požadovaného tvaru a rozměrů. Existuje celá řada programů umožňujících rychlé 3D modelování, někdy se využívá i moderních technologií jako je laserové skenování již existujících objektů. Existují také celé online databáze, ze kterých lze zdarma stáhnout modely určené k 3D tisku. Prostorový model se následně musí ve speciálním programu připravit k tisku. Na základě požadované výšky vrstvy a dalších vstupních parametrů se vytvoří soubor s instrukcemi pro tiskárnu. Poté je některou ze zvolených technologií aditivní výroby objekt postupně po vrstvách vytištěn. Zejména u technologie FDM je k dispozici celá řada materiálů o různých vlastnostech, ze kterých můžeme tisknout. Po skončení tisku je hotový objekt běžně připraven k použití. Někdy je ovšem nutné s ním ještě dále pracovat. Běžné jsou různé povrchové úpravy v závislosti na použitém materiálu a technologii.

1.1 Motivace

Proces 3D tisku ovlivňuje celá řada faktorů a i malá chyba může vést k selhání tisku. Častým důvodem neúspěchu jsou problémy spojené se špatnou kalibrací 3D tiskárny. Může se jednat například o špatně vyrovnanou tiskovou podložku, špatně nastavenou teplotu horkého konce nebo nekvalitní či poškozený filament.

I při dokonalé kalibraci tiskárny, eliminaci vibrací jejího rámu a použití kvalitního filamentu můžeme mít problémy s nadměrnými deformacemi tisknutého objektu. Ty vycházejí z problematiky vedení tepla v materiálu, respektive z deformací způsobených velkými teplotními gradienty (viz obr. 1). Běžně se v praxi používají různá lepidla pro zabránění

¹Jedná se o nejpopulárnější technologii 3D tisku, kdy z horké trysky vytéká termoplast, který tvrdne a spojí se s předchozí vrstvou.

zdvihání rohů z podložky. Pro tisk některých materiálů (např. ABS) se používá vyhřívaná podložka nebo dokonce uzavření celé tiskárny do komory pro zvýšení teploty exteriéru. Řešení těchto problémů není zatím podloženo numerickými simulacemi, ale spíše se jedná o metodu „pokus omyl“.

Úroveň volně dostupných programů umožňuje připravit prostorový model k tisku v řádu několika málo minut. Přesto v současné době neexistuje žádný volně dostupný nástroj, který by se napojil do procesu 3D tisku a umožnil numerickou simulaci výrobního procesu 3D tisku. Existuje pouze několik programů (např. gCodeViewer [4]), které umí zjistit základní informace o tisku a vizualizovat dráhu pohybu tiskové hlavy.



(a) Zvedání rohů



(b) Delaminace na rozhraní vrstev

Obrázek 1: Typické problémy související s deformací rozdílem teploty. Převzato z [5].

1.2 Cíle práce

Hlavním cílem této práce je vytvoření nástroje pro tvorbu numerického modelu pro analýzu 3D tisku metodou konečných prvků. Model bude tvořen osmiuzlovými prvky tvaru krychle. Hlavními vstupními parametry pro tvorbu toho modelu budou délka strany krychle a délka časového kroku. Nástroj by měl poskytnout data potřebná pro postupnou aktivaci jednotlivých prvků a zjištění jejich poměrného zaplnění. Výstup by měl být snadno exportovatelný do současného konečně-prvkového software.

Při sestavování zjednodušeného modelu budou též zjištěny informace o průběhu 3D tisku, jako je např. odhadovaná doba tisku, spotřeba materiálu, výšky jednotlivých vrstev atd.

2 3D tisk

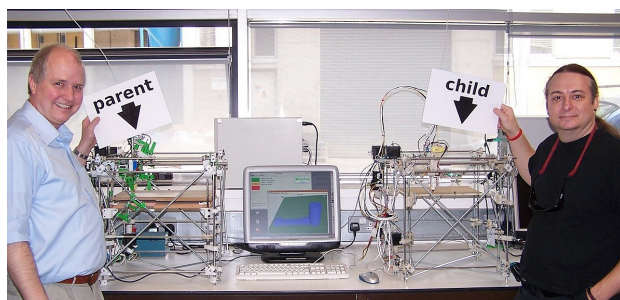
3D tisk je proces výroby reálného objektu na základě jeho digitální předlohy ve formě prostorového modelu. Původním účelem této technologie byla rychlá výroba prototypů, s tím souvisí původní a dodnes používané označení *Rapid Prototyping*. Dnes se používá termín aditivní výroba jako protiklad k subtraktivním výrobním metodám (např. obrábění).

Historie 3D tisku sahá do 80. let minulého století, kdy byla v roce 1984 Charlesem W. Hullem patentována technologie stereolitografie. V roce 1987 spatřila světlo světa první komerční 3D tiskárna založená na této technologii (viz obr. 2). V průběhu let vzniklo několik dalších technologií 3D tisku. Jejich společnou charakteristikou je postupná výroba objektu po jednotlivých tenkých vrstvách. Nejrozšířenější metodou je v současné době *Fused Deposition Modelling* (FDM), ze které vychází i téma této práce. Některé vybrané metody jsou popsány v podkapitole 2.2.



Obrázek 2: SLA-1 – První komerční 3D tiskárna na světě. Převzato z: [6].

Rok 2005 je pravděpodobně nejdůležitějším milníkem pro rozvoj a rozšíření 3D tisku do dnešní podoby. Na univerzitě v Bathu byl Adrianem Bowyerem založen otevřený projekt 3D tiskárny RepRap (viz obr. 3). Zároveň v roce 2009 vypršely patenty na technologii FDM.



Obrázek 3: RepRap – Tiskárna schopná replikovat své plastové části. Převzato z [2].

Přestože prvotní použití 3D tisku jako způsobu výroby prototypů přetrvává dodnes, používá se v současné době 3D tisk i k výrobě plnohodnotných součástek. Hlavní výhodou 3D tisku je, že zdánlivě komplikované geometrie se realizují úplně stejně snadno jako jednoduché tvary (např. krychle). Obecně je možné použít široké množství materiálů (plast, kovy atd.).



Obrázek 4: Příklad využití 3D tisku v praxi. Převzato z [7, 8, 9].

V případě použití termoplastů se jedná o zajímavou alternativu k běžné výrobě plastových výrobků metodou vstřikování do forem (*injection moulding*). Zásadní výhodou oproti konvenční technologii je absence potřeby forem. Ty bývají finančně velmi nákladné, komplikované na návrh a jejich výroba je časově náročná. Na druhou stranu, pokud potřebujeme vyrábět velké množství kusů, nemá tradiční vstřikování plastů konkurenci.

2.1 Výrobní proces

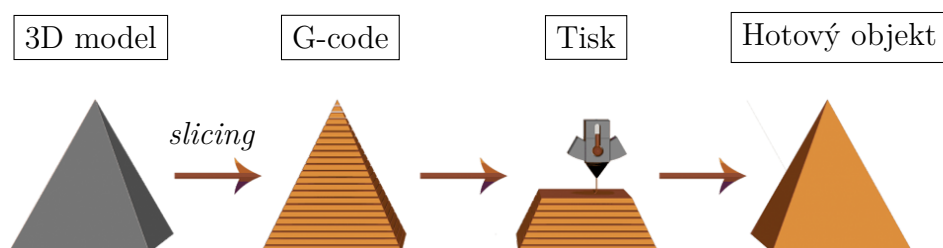
Úspěšný 3D tisk je zajištěn sledem několika na sebe navazujících procesů.

Ve fázi *pre-processingu* se nejprve navrhne prostorový model objektu. K tomu není potřeba žádný software určený přímo pro 3D tisk. Postačí jakýkoliv program, který umožňuje export modelu v některém z podporovaných formátů (2.1.1.2).

Po vytvoření prostorového modelu na počítači je potřeba ho rozdělit po výšce na jednotlivé tenké vrstvy volitelné výšky. Tato fáze *pre-processingu* se nazývá *slicing*. U složitějších objektů, zejména pokud se jejich geometrie mění po výšce, je potřeba často vytvořit tzv. dočasné podpěry – i to je úlohou *sliceru*. Výstupem je řídicí kód s instrukcemi pro 3D tiskárnu (tzv. soubor *G-code*).

Samotný tisk v závislosti na velikosti objektu a zvolené výšce vrstvy běžně trvá desítky minut, ale výjimkou nejsou ani tisky trvající desítky či stovky hodin. Proces tisku bývá označován jako *processing*.

S hotovým objektem se často v tzv. fázi *post-processingu* ještě dále pracuje. Běžně jde o odstranění dočasných podpěr nebo o úpravu povrchu pro dosažení požadované estetičnosti.



Obrázek 5: Schéma výrobního procesu 3D objektu. Převzato z [10].

2.1.1 Modelování

Model objektu se nejprve navrhne pomocí CAD software nebo se získá například laserovým 3D skenováním. Při navrhování je důležité zohlednit minimální tloušťku částí modelu, kterou jsme schopni 3D tiskem realizovat. Ta je v případě FDM omezena zejména průměrem použité trysky, ale i požadovanou pevností vytištěných objektů. Příliš tenké stěny jsou náchylné na delaminaci v místě spojení jednotlivých vrstev.

2.1.1.1 Dostupný software

Existuje celá řada komerčních programů i jejich bezplatných alternativ. Využít lze prakticky jakýkoliv software, který umožňuje export do některého z běžně používaných formátů (2.1.1.2).

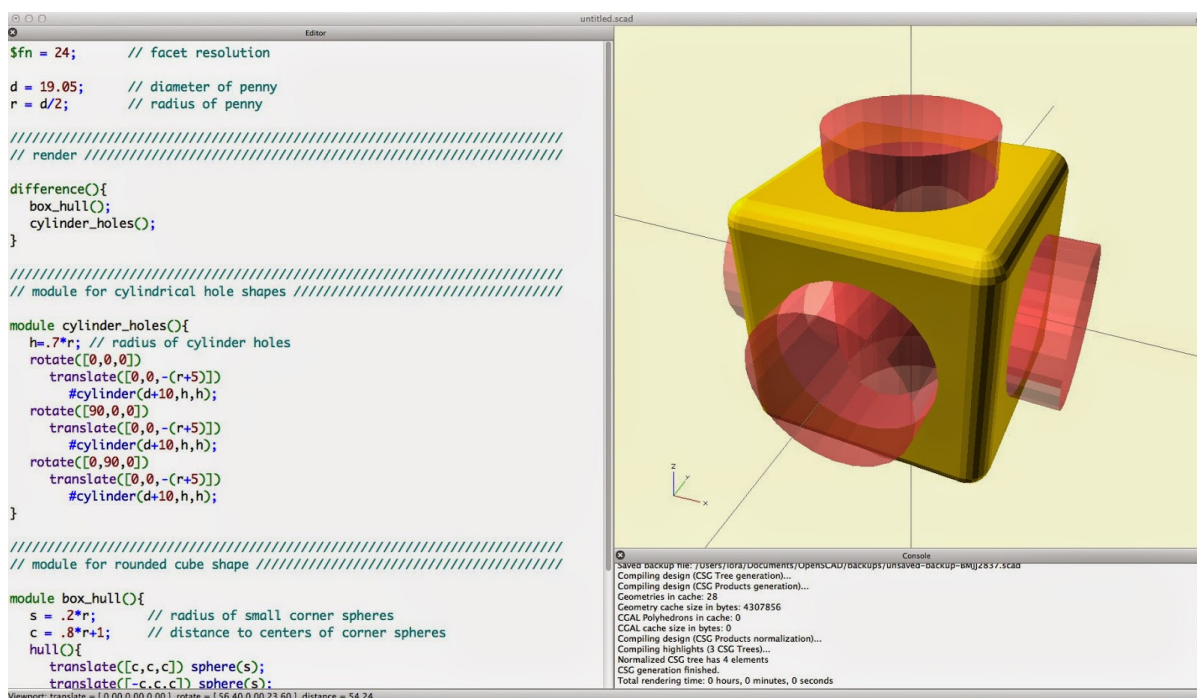
Komerční software

Pokročilejší uživatelé mohou použít například placený software Solidworks, Rhino 3D nebo SketchUp. Lze použít i z inženýrské praxe dobře známé produkty firmy Autodesk (AutoCAD, Inventor)[11].

Bezplatný software

Pro úspěšný 3D tisk není nutné používat pokročilé placené programy pro 3D modelování. Zejména pro začátečníky a pro tvorbu jednoduchých modelů plně dostačují programy dostupné zdarma jako např. TinkerCAD nebo FreeCAD.

Zajímavou alternativou ke klasickému modelování je použití volně dostupného software OpenSCAD (viz obr. 6), ve kterém je možné geometrii objektu „programovat“. Tím lze snadno vytvářet parametrické modely komplikovaných tvarů vzniklé například sjednocením či rozdílem dílčích prostorových objektů.



Obrázek 6: OpenSCAD

2.1.1.2 Výstupní formáty

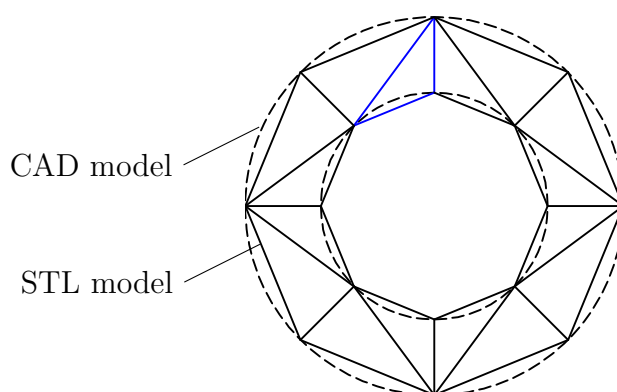
Pro zpracování *slicerem* je potřeba vyexportovat geometrii 3D modelu v některém z běžně podporovaných formátů.

Formát STL

Formát STL (Stereolithography) původně vznikl pro potřeby software firmy 3D Systems. Jak jeho název napovídá, užívá se pro objekty realizované stereolitografií, ale pro svou jednoduchost byl přejat i pro další metody tisku (zejména FDM). Stále se jedná o nejrozšířenější formát uložení modelu objektu pro 3D tisk.

Formát STL popisuje pouze geometrickou reprezentaci trojrozměrného objektu. V souboru se nacházejí informace o povrchu, který je popsán pomocí povrchové triangulace a příslušných normálových vektorů. Protože popisuje geometrii objektu pouze pomocí trojúhelníků, při exportu do tohoto formátu jsou zakřivené úseky rozděleny na několik přímých dílků, jak je patrné z obrázku 7. Soubor STL zároveň neobsahuje žádné informace o barvách, materiálu či jiných parametrech modelu.

Soubor může být reprezentován přímo v textové podobě nebo binárně. Pro úsporu místa se používá binární reprezentace.



Obrázek 7: Princip reprezentace mezikruží ve formátu STL

Formát OBJ

Soubor OBJ (Wavefront Object File) narozdíl od STL umožňuje popis geometrie pomocí jiných prvků než trojúhelníků. Prvek není omezen třemi body, ale povrch je možné sestavovat například z čtyřúhelníků nebo složitějších polygonů. Zároveň tento formát podporuje i několik typů křivek (NURBS, Cubic Hermite spline a Bézier curve) pro popis zakřivených oblastí.

Formát je běžnými slicery podporován a tak lze model snadno držet v jednom souboru pro 3D tisk i jiné využití.

Formát VTK

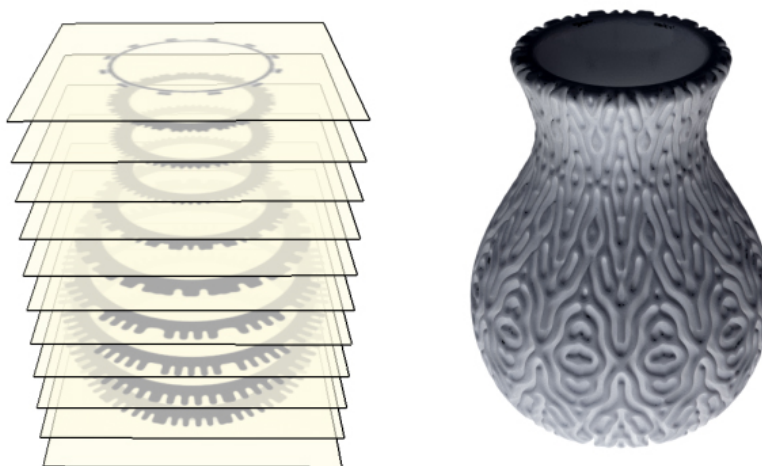
Formát VTK (Visualisation Toolkit) je široce rozšířený zejména v oblasti konečně-prvkových programů pro export výsledků a jejich post-processing např. v programu ParaView. Umož-

ňuje popis složitější geometrie pomocí nelineárních prvků (kvadratický čtyřstěn, hexahedron atd.). Také je možné kromě pouhé geometrie uložit i další parametry modelu jako např. informace o materiálu.

Slicery tento formát zatím běžně nepodporují, ale je možné použít některý z volně dostupných nástrojů pro vzájemný převod mezi formáty [12].

2.1.2 Slicing

Hotový 3D model je potřeba připravit k tisku ve specializovaném programu zvaném *slicer*. Prostorový model je rozřezán na tenké vrstvy obvykle stejné tloušťky. Některé programy dokonce umožňují tloušťky jednotlivých vrstev spočítat automaticky tak, aby byla geometrie modelu po výšce respektována co nejpřesněji.



Obrázek 8: Slicing - proces rozřezání objektu na vrstvy po výšce. Převzato z: [13].

Výška vrstvy je sice stěžejním parametrem pro *slicing*, protože na ní přímo závisí doba tisku. Výsledný tisk ovšem ovlivňuje i celá řada dalších vstupních parametrů. V případě technologie FDM se jedná o průměr použitého filamentu, průměr trysky a charakteristiky konkrétní tiskárny – např. maximální rychlost pohybu, zrychlení atd.

Zajímavou charakteristikou 3D tisku je, že není nutné tisknout objekty jako 100% vyplněné. Existují různé vzory, kterými může být objekt vyplněn. Určením vzoru a procenta vyplnění zmenšíme množství potřebného materiálu a také dobu tisku. Je třeba mít na paměti, že výběr konkrétního vzoru a procenta vyplnění má přímý vliv na pevnost výsledného objektu.

Slicer navrhne po rozdělení na jednotlivé vrstvy optimální dráhu pohybu tiskové hlavy. V případě FDM je taktéž pro každý pohyb určeno potřebné množství materiálu (délku filamentu) pro jeho depozici. Potřebné množství filamentu je určeno požadovanou šířkou jedné depozice, ale přímo závisí i na průměru trysky a průměru použitého filamentu.

Výstupem *sliceru* je soubor příkazů pro 3D tiskárnu ve formátu G-code. Obsahuje nejen instrukce pro pohyb tiskové hlavy v prostoru, ale v případě technologie FDM i délku vytlačeného filamentu v jednotlivých pohybech. Zároveň nastavuje důležité parametry tiskárny – teplotu horkého konce, vyhřívání podložky, aktuální rychlost pohybu atp.

2.1.3 Tisk

Pro postupnou realizaci objektu po vrstvách můžeme využít celou řadu různých technologií aditivní výroby (viz podkapitola 2.2). Tyto technologie se liší použitými materiály, rychlostí i cenou. Tiskárna je řízena souborem G-code vytvořeným v procesu slicing (viz 2.1.2).

Výška jednotlivých vrstev se v závislosti na použité technologii pohybuje běžně okolo 0,1 – 0,4 mm. Jedná se o zásadní parametr určující časovou náročnost tisku. Při použití tenkých vrstev sice dosáhneme vzhledově dokonalejších objektů, ale doba tisku může vzrůst z řádu minut do řádu hodin. Efektivnější řešení bývá použití větší tloušťky vrstvy a následná úprava výsledného povrchu některou z metod uvedených v 2.1.4.

Tisk umožňuje realizovat tvarově komplikované objekty stejně snadno jako objekty jednoduché. To je zásadní výhoda oproti konvenčním technologiím, jakým je např. vstříkování plastů do forem.

2.1.4 Post-processing

Po skončení tisku většinou stačí vytištěný objekt po vychladnutí oddělit od podložky. V závislosti na tvarové složitosti výtisku či požadované povrchové úpravě může být nutné s výtiskem dále pracovat.

Při tisku převyšujících částí u geometricky složitých objektů jsou tyto části podepřeny dočasnými podpěrami. Pokud jsou správně umístěny, dají se jednoduše mechanicky odstranit (odlomit) bez narušení hlavního objektu. V případě použití 3D tiskárny s více tryskami lze použít pro dočasné podpory speciální materiál. Například použitím PVA (polyvinylalkohol) můžeme podpory po dokončení tisku rozpustit ponořením do vody.

Nedílnou součástí 3D tisku je při použití běžných trysek (průměr okolo 0,4mm) viditelná mikrostruktura. Hladkého povrchu lze dosáhnout různě. Často se jedná o broušení, leštění nebo různé epoxidové nátěry. U termoplastů rozpustných v acetonu (např. ABS) se dá krátkým vystavením jeho výparům dosáhnout dokonale hladkého a lesklého povrchu.

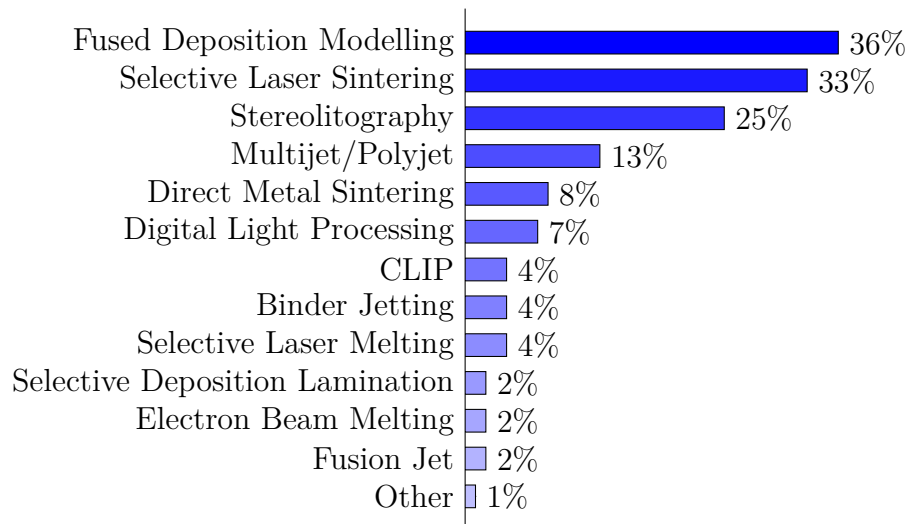
2.2 Přehled technologií

Jednotlivé technologie 3D tisku můžeme rozlišit podle typu použitého materiálu a způsobu jeho zpracování.

Práškové materiály Silný zdroj tepla či světla je použit pro spojování tenkých vrstev prášku do žádaného tvaru. Materiálem mohou být plasty (nylon, polyamid), keramická hmota, sklo, ale i kovy. Práškové materiály jsou zpracovávány technologiemi *selective laser sintering* (slinování plastů) a *direct metal sintering* (slinování kovů).

Tekuté polymery Osvit speciální pryskyřice laserem zajišťuje její tvrdnutí v tenkých vrstvách. Toho využívají technologie stereolitografie nebo *digital light processing*.

Termoplasty Termoplast je postupně taven a vytlačován tryskou různého průměru. Toho využívá nejrozšířenější technologie tisku – *fused deposition modelling*.

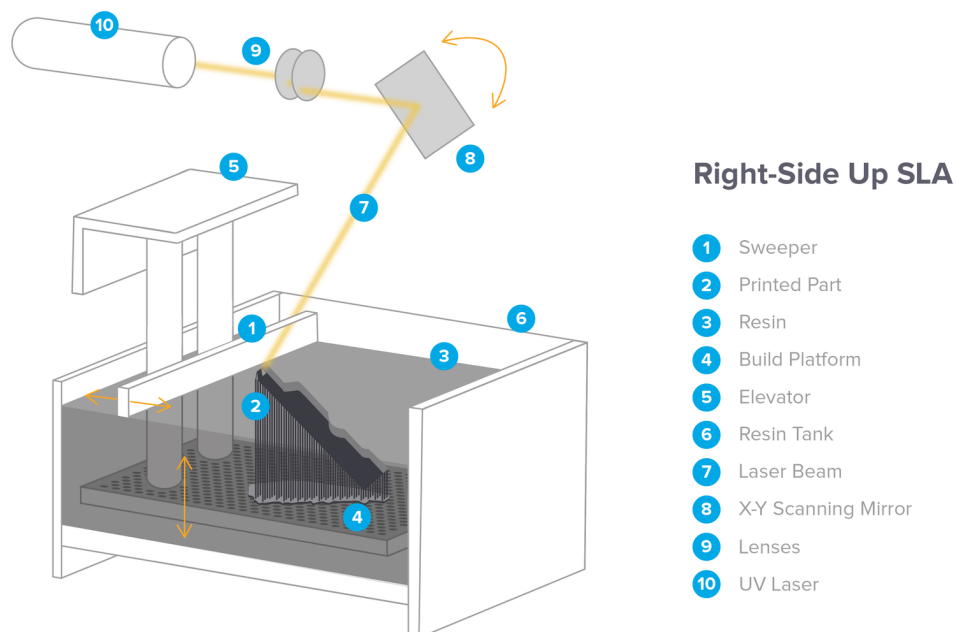


Obrázek 9: Statistika používaných technologií 3D tisku. Převzato z [14].

2.2.1 Stereolitografie

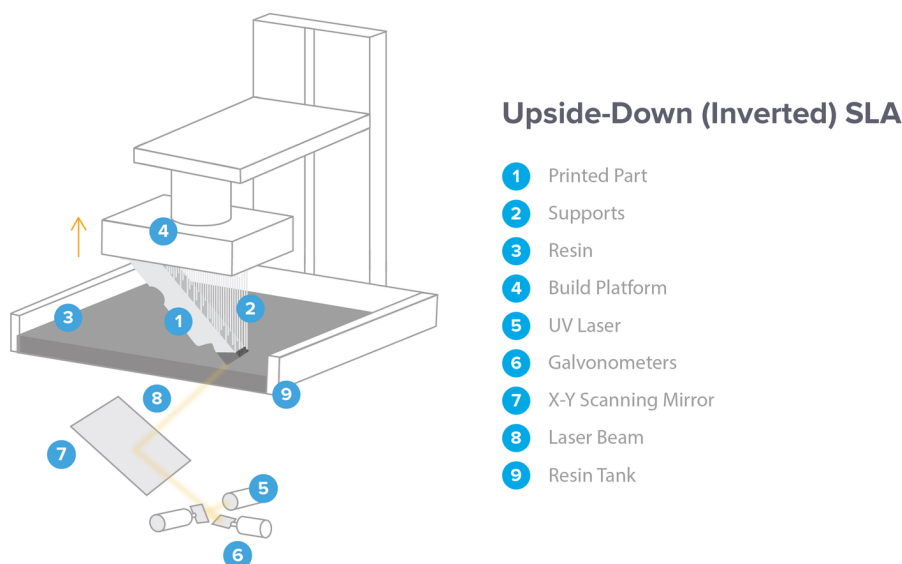
Stereolitografie je první a dodnes široce rozšířenou technologií 3D tisku. Počátky výzkumu sahají do 70. let minulého století, ale až v roce 1984 byl Chuckem Hullem podána žádost o patent, která byla schválena v roce 1986.

První stereolitografické tiskárny byly založeny na principu, kdy se objekt realizuje po vrstvách odspodu tak, že plocha vrstvy pryskyřice osvětlená laserem tvrdne a platforma postupně klesá do nádrže s pryskyřicí. Nevýhodou je potřeba velké nádrže na pryskyřici. Tiskárny založené na této technologii jsou větší, dražší a složitější na údržbu. Jedná se proto o řešení používané výhradně v průmyslové aplikaci.



Obrázek 10: Stereolitografie – „Right-Side Up“. Převzato z [1].

Další možností je tisk „vzhůru nohama“, kdy se laserem ozařuje spodní část nádoby s tenkou vrstvou tekuté pryskyřice. Po vytištění jedné vrstvy platforma s objektem stoupne vzhůru a dojde k doplnění fotopolymerní pryskyřice. Tato metoda nevyžaduje plnou nádrž s pryskyřicí a bývá obvyklá u menších domácích tiskáren.



Obrázek 11: Stereolitografie – „Upside-Down“. Převzato z [1].

2.2.2 Selective laser sintering

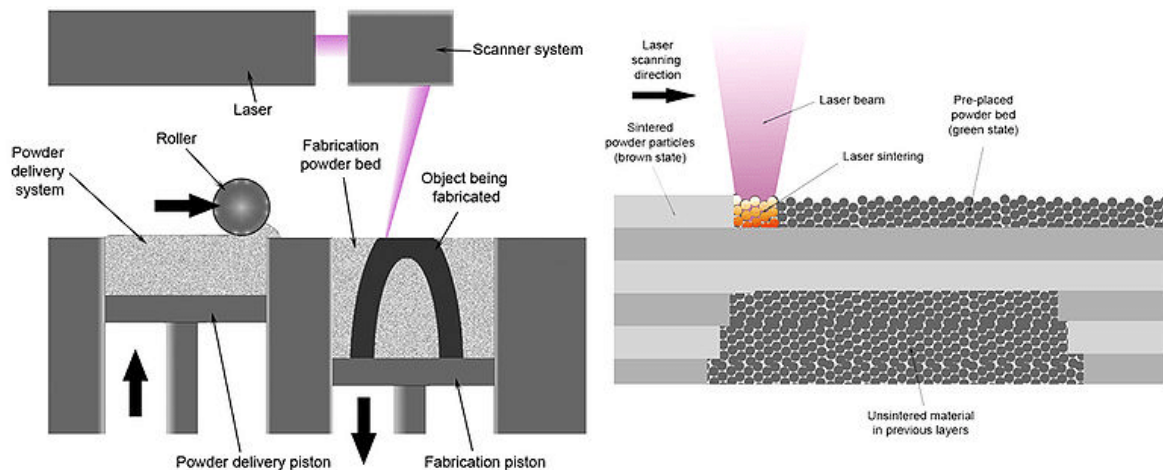
Historie laserového slinování sahá do 80. let minulého století. Technologie vznikla na Texaské univerzitě v Austinu [15].

Pro slinování práškového materiálu používá laser o vysokém výkonu. Obvykle se jedná o plast (nylon nebo polyamid), ale je možné použít např. sklo nebo kovy (*direct metal sintering*).

Princip výroby objektu postupně po vrstvách je shodný s dalšími metodami 3D tisku. Práškový materiál je umístěný v komoře a přehřátý na teplotu nižší než je jeho teplota tání. Díky tomu stačí k roztavení prášku menší výkon laseru. Po dokončení vrstvy platforma poklesne, je doplněn materiál a pokračuje se vytvrzováním další vrstvy.

Hlavní výhodou oproti jiným technologiím je, že nepotřebuje u složitých geometrií žádné dočasné podpěry. Objekt je od začátku z principu samonosný. Výsledkem laserového slinování mohou být objekty s mechanickými vlastnostmi blížícími se objektům vyrobeným tradičně vstřikováním do forem. Tato technologie je využívána v leteckém a kosmickém průmyslu k výrobě prototypů i finálních součástek.

Trh domácích 3D tiskáren na bázi laserového slinování není vzhledem k vysoké ceně a použití výkonných laserů nijak rozšířený.



Obrázek 12: Selective laser sintering. Převzato z [15].

2.2.3 Fused deposition modelling

Fused deposition modelling, je současně nejrozšířenější metodou 3D tisku. Tato technologie je z otevřeného projektu RepRap známá též jako *Fused filament fabrication*. Od vypršení patentů v roce 2009 dochází k velkému komerčnímu rozvoji této technologie.

Princip spočívá v tavení plastu nebo jiného materiálu ve formě vlákna uvnitř tiskové hlavy. Tento tzv. filament je poté z horké trysky vytlačován na podložku či předchozí vrstvu, ke které přilne.

Hlavní výhodou tohoto způsobu 3D tisku je nízká pořizovací cena 3D tiskáren a relativně jednoduchá obsluha. Nevýhodou jsou komplikace spojené s nepřilnavostí k podložce a deformace způsobené teplotními rozdíly v materiálu.

2.2.3.1 Materiály

Technologie FDM používá jako materiál (filament) termoplast ve formě tiskové struny. Běžně používané průměry jsou 1,75 a 3,00 mm. Nejčastěji používanými materiály jsou termoplasty ABS (Akrylonitrilbutadienstyren) a PLA (polyaktická kyselina). Existuje celá řada dalších materiálů s odlišnými fyzikálními vlastnostmi (flexibilní, magnetické ap.). Přehled běžně používaných filamentů je uveden v tabulce 1.

Vlastnost	ABS	PLA	Nylon	PETG
Tahová pevnost [MPa]	40	65	40–85	53
Koef. tepelné roztažnosti [$^{\circ}\text{C}^{-1}$]	$90 \cdot 10^{-6}$	$68 \cdot 10^{-6}$	$95 \cdot 10^{-6}$	$60 \cdot 10^{-6}$
Hustota [g/cm^3]	1,04	1,24	1,10	1,23
Teplota trysky [$^{\circ}\text{C}$]	220–250	190–220	220–270	230–250
Max. teplota použitelnosti [$^{\circ}\text{C}$]	98	52	80–95	73

Tabulka 1: Přehled používaných materiálů – FDM. Převzato z [16].

2.3 G-code

G-code je jazyk pro automatizované řízení strojů pomocí počítače [17]. Vznikl původně v laboratořích MIT již v 60. letech minulého století a jedná se o velmi jednoduchý jazyk. Sled po sobě jdoucích příkazů a jejich parametrů určuje přesně chování řízeného stroje a jeho nástrojů v čase. V případě 3D tiskárny se jedná zejména o ovládání pohybu tiskové hlavy v kartézské soustavě souřadnic a její rychlosti, teploty horkého konce a množství vytlačeného materiálu. Na rozdíl od obráběcích strojů se G-Code pro 3D tisk až na výjimky (např. kalibrace) nepíše ručně.

Ačkoliv pro G-code řídicí obráběcí stroje existuje několik standardů, nejsou tato pravidla napříč širokým spektrem výrobců dodržována [17]. Význam jednotlivých příkazů či parametrů se může dokonce úplně lišit. Stejně peripetie trápí i moderní 3D tiskárny, kde v zásadě každý větší výrobce má svůj standard. Při exportu ze *sliceru* je důležité mít nastavený formát G-Code kompatibilní s konkrétní 3D tiskárnou. Výrobce obvykle na přiložené SD kartě či CD dodává doporučený *slicer*. V této práci je respektován G-Code dle specifikace otevřeného projektu RepRap [17].

```

M107 <--- Větrák 0%
M104 S205 <----- Nastavit teplotu horkého konce na 205 °C
G28 <----- Návrat do referenčního bodu
G1 Z5 F5000 <--- Posun do Z=5 mm rychlostí 5000 mm/min

; Filament gcode <----- Komentář

; set temperature and wait for it to be reached <----- Komentář
M109 S205 <----- Nastavení teploty horkého konce na 205°C
G21
G90
M82
G92 E0
G92 E0
G1 Z0.350 F7800.000 <--- Posun do Z=0,35 mm rychlostí 7800 mm/min
G1 E-2.00000 F2400.00000 <--- Zatažení 2 mm filamentu rychl. 2400 mm/min

```

Obrázek 13: Příklad skutečného souboru G-code

Příkazy G-Code jsou uloženy v textovém souboru na jednotlivých řádcích. Tyto příkazy se v zásadě dají rozdělit na příkazy související s pohybem (G) a speciální příkazy (M). Přehled běžně se vyskytujících příkazů je uveden dále, podrobná tabulka všech používaných příkazů je uvedena např. v [17].

2.3.1 Pohybové příkazy

Pohyb tiskové hlavy probíhá nejčastěji ve vodorovné pracovní rovině XY . Svislá souřadnice Z slouží obvykle pro postup na vyšší vrstvu. Pohyb v ose E způsobuje vytlačování filamentu z trysky (přírůstek E) nebo jeho zatažení (úbytek E) před netiskovým posunem. Pohyby jsou vykonávány aktuálně nastavenou rychlostí dle parametru F . Každá

tiskárna má své maximální hodnoty rychlosti pro jednotlivé osy, pokud by na základě G-code byla některá z maximálních hodnot překročena, proběhne pohyb pomaleji – rychlost všech os zůstane synchronizována lineární interpolací.

2.3.1.1 Nejpoužívanější příkazy

G0 [X#][Y#][Z#][F#][E#]

Rapid move je dle specifikace pohyb na zadané souřadnice co nejvyšší možnou rychlostí. Tento typ pohybu není v 3D tisku využíván, resp. je s ním běžně zacházeno jako s příkazem G1 (dle [17]).

G1 [X#][Y#][Z#][F#][E#]

Feed move je pohyb do souřadnic X, Y, Z s posunem filamentu E definovanou rychlostí F . Pohyb všech os je synchronizován.

G2, G3 [X#][Y#][I#][J#][F#][E#]

Arc move je pohyb po kružnici se středem v I, J do cílových souřadnic X, Y ve směru nebo proti směru hodinových ručiček. Většina současných *slicerů* generuje místo tohoto pohybu několik přímých segmentů pomocí příkazů G1. Důvodem je, že nejpoužívanější vstupní formát souboru pro slicer (STL) ukládá geometrii pomocí trojúhelníků. Zpětný výpočet kružnicových částí by byl zbytečně náročný a náchylný na chyby.

2.3.2 Speciální příkazy

Kromě pohybových příkazů existuje celá řada speciálních příkazů. Mezi nejdůležitější patří příkazy pro nastavení teploty horkého konce a případně vyhřívané podložky. Typické je též nastavení rychlosti větráku pro chlazení tištěného objektu (zejména pro materiál PLA).

M104 [S#]

Nastavení teploty S horkého konce. Tiskárna nečeká na dosažení požadované teploty.

M106 [P#][S#]

Nastavení rychlosti S pro větrák P .

M109 [S#]

Nastavení teploty S horkého konce. Před dalším pokračováním tisku musí být dosaženo požadované teploty (tiskárna čeká).

M190 [S#]

Nastavení teploty S vyhřívané podložky. Před dalším pokračováním tisku musí být dosaženo požadované teploty (tiskárna čeká).

2.4 Problémy

V kapitole 1.1 byly nastíněny problémy způsobené vysokými rozdíly teplot při 3D tisku. Níže jsou tyto problémy rozebrány podrobněji.

2.4.1 Smršťování

Při tisku z materiálu citlivého na změny teplot jakým je např. ABS dochází smršťováním k deformaci tištěného objektu. Materiál vycházející z trysky má teplotu okolo 230 °C, teplota podložky běžně přesahuje 100 °C. Tyto teploty několikanásobně přesahují teplotu interiéru, ve kterém jsou tiskárny provozovány. Smršťování je způsobeno postupným chladnutím objektu v oblasti mezi podložkou a právě tištěnou vrstvou na teplotu blíží se okolní teplotě. U větších objektů vlivem smršťování může dojít až k úplnému odlepení objektu od tiskové podložky. To následně vede buď k úplnému selhání tisku nebo dotištění zdeformovaného objektu.

Ideálním řešením tohoto problému by bylo zajištění postupného rovnoměrného chladnutí celého tisknutého objektu. To sice z principu úplně možné není, ale lze se tomuto stavu přiblížit uzavřením tiskárny do komory, která umožní ohřev okolního vzduchu na vyšší teplotu. U malých běžně používaných 3D tiskáren s vyhřívanou podložkou se jejich uzavřením do komory dosahuje teploty okolo 50 °C.

Problém přilnavosti se dá řešit použitím speciálních adhezních povrchů tiskové podložky nebo použitím různých lepidel. Stále ovšem mohou vznikat deformace způsobené nerovnoměrným smršťováním, případně mohou v materiálu zůstat větší reziduální napětí snižující celkovou pevnost objektu.

2.4.2 Přehřívání

Příliš vysoká rychlost tisku může někdy vést k deformacím, které jsou způsobeny nedostatečně rychlým chladnutím předchozích vrstev. K tomu může snadno dojít při tisku štíhlých a vysokých objektů. Tiskový materiál opouští trysku o teplotě běžně přesahující 200 °C. Pokud je při tisku vrstva ještě příliš horká a tudíž v (polo)tekutém stavu, může dojít při výtisku navazující vrstvy k nadměrným deformacím, které mohou vést i k úplnému selhání tisku všech dalších navazujících vrstev.

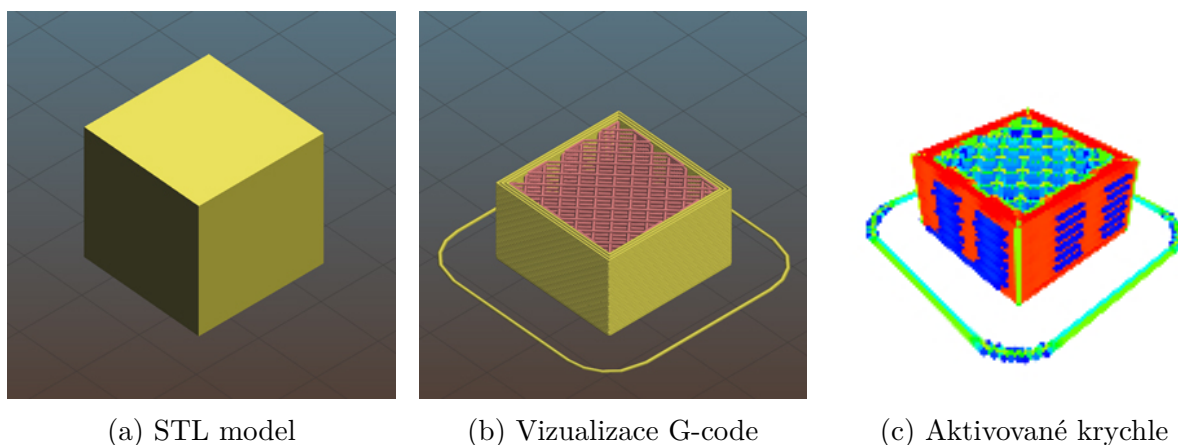
Řešením je zpomalení pohybu tiskové hlavy v problematických vrstvách. To však slicery běžně neumožňují. Konečně-prvková analýza konkrétního modelu by mohla pomoci určit maximální rychlost pohybu v problematických vrstvách. Tento problém také lze řešit snížením teploty trysky, ale jen do určité míry. Pokud nebude tato teplota dosahovat bodu tání konkrétního materiálu, nebude z trysky vycházet žádný materiál.

3 Procesor G-code

V této kapitole je podrobně popsán postup numerické emulace 3D tisku na rastrovém modelu. Vlastní model je vytvořen nad digitálním rastrem tvořeným z prostorových krychlí o volitelném rozlišení a v principu jde o převod vektorového modelu G-code na rastrový model vhodný pro numerickou simulaci.

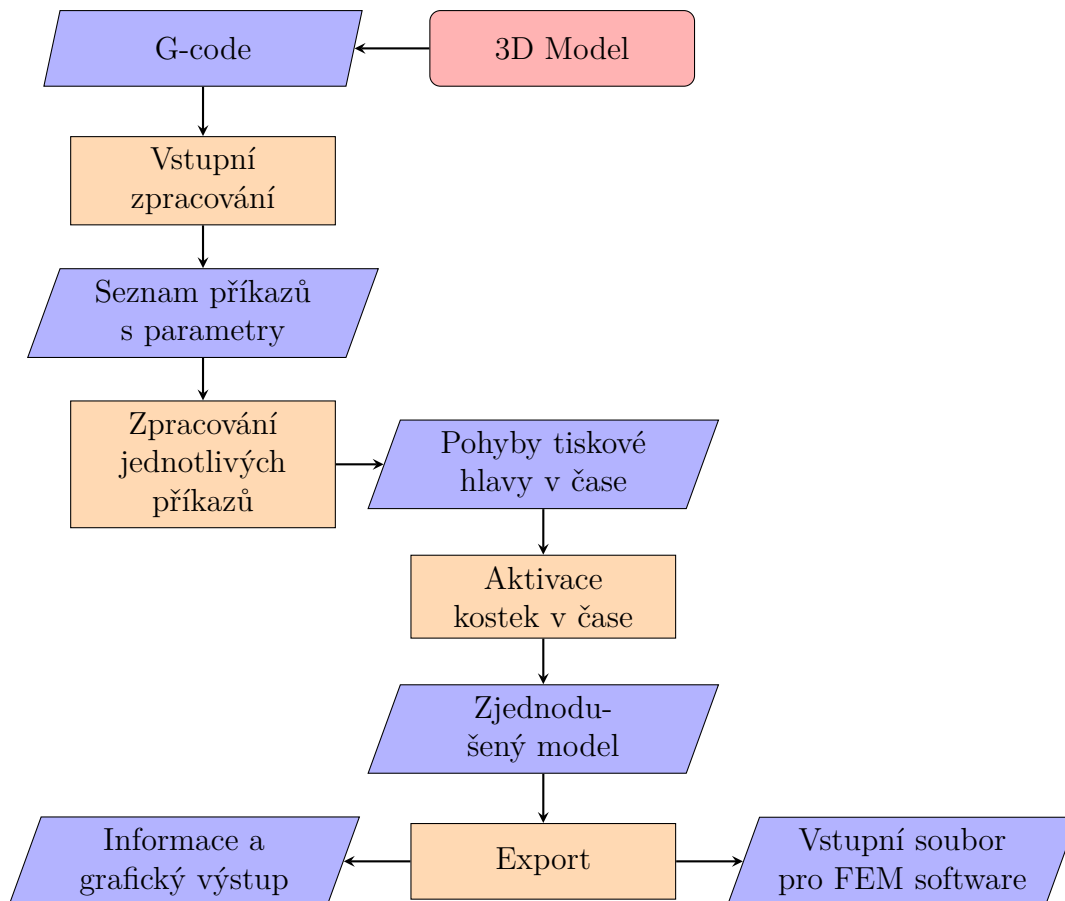
Nejprve je vždy potřeba zpracovat vstupní soubor G-code podle kapitoly 3.1. Následně můžeme ze seznamu příkazů sestavit pohyby tiskové hlavy a pro každý pohyb vyhodnotit informace o uražené vzdálenosti, času jeho trvání a objemu vytisknutého materiálu, jak je podrobně popsáno v kapitole 3.2. Z těchto informací lze sestavit rastrový prostorový model. Volbou časového kroku jsme schopni zjistit procentuální zaplnění a čas aktivace každé krychle modelu v libovolném kroku. Rastrový model s historií přiděleného objemu je vhodný pro následný import do konečně-prvkových programů a následný výpočet.

Na obrázku 14 je ukázán výsledek emulace 3D tisku spodní poloviny krychle $20 \times 20 \times 20$ mm na rastrovém modelu z krychlí o délce hrany 0,5 mm, originální STL model a vizualizace G-code z programu Slic3r.



Obrázek 14: Převod vektorového modelu G-code na rastrový model z krychlí

Konkrétní implementace realizovaná v jazyce JavaScript (Node.js) je přiložena na disku CD. Program byl zpracován jako interaktivní webová aplikace. Ke spuštění tohoto programu tedy stačí jakýkoliv běžně dostupný webový prohlížeč (Firefox, Chrome). Součástí je uživatelské rozhraní, které pohodlně umožňuje import souboru G-code a jeho následné zpracování dle požadovaných parametrů. Nástroj poskytuje i vizualizace tiskových pohybů a graficky emuluje postupnou aktivaci jednotlivých krychlí rastrového modelu. Díky volbě prostředí webového prohlížeče je obsluha výsledného programu snadná a intuitivní.



Obrázek 15: Schéma fungování procesoru G-code

3.1 Vstupní zpracování

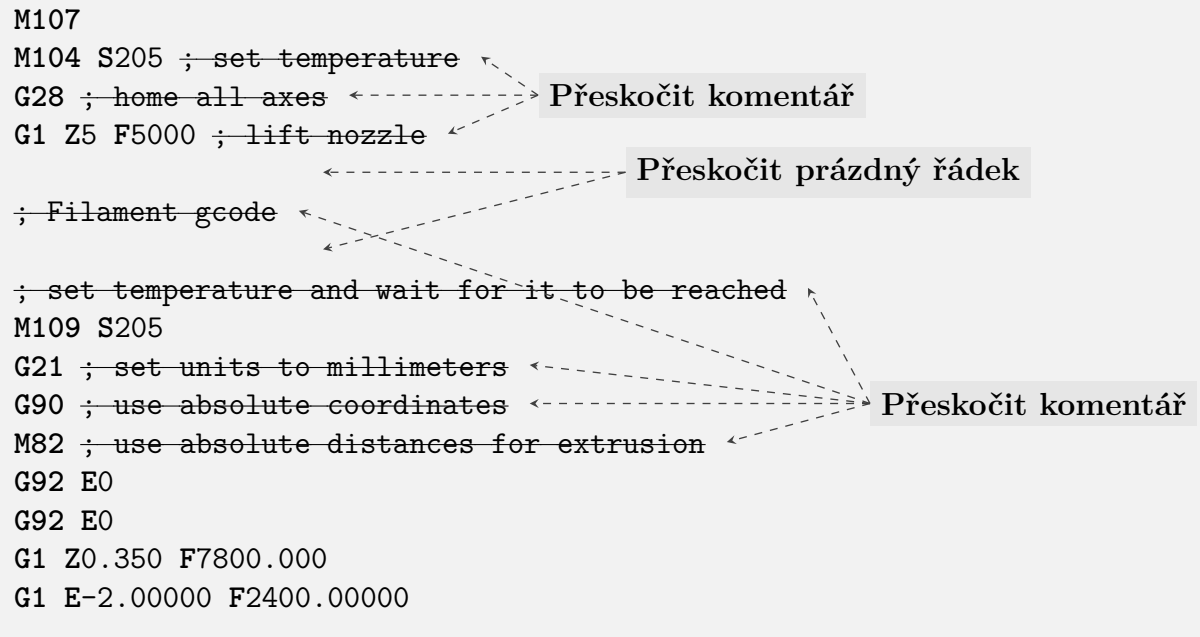
Soubor *G-code* obsahuje v textové podobě příkazy umístěné na jednotlivých řádcích, jak je ukázáno na obr. 16. Tento soubor je tedy pro další zpracování načten do paměti a převeden na seznam jednotlivých řádků. Při konkrétní implementaci se při rozdělování na řádky nesmí zapomenout na to, že kontrolní znak označující konec řádku se liší napříč operačními systémy (Unix – `\n`, Windows – `\r\n`).

Při zpracovávání jednotlivých řádků můžeme prázdné řádky rovnou přeskočit. Pokud se na řádku vyskytuje znak `;` (komentář), můžeme zbylý obsah řádku, počínaje tímto znakem, ignorovat. Na ostatních řádcích se program pokusí najít platný příkaz G nebo M. Pokud je nalezen, zpracují se postupně všechny parametry uvedené na daném řádku.

Výsledkem načtení vstupního souboru je seznam příkazů G a M včetně jejich odpovídajících parametrů.

3.2 Zpracování jednotlivých příkazů

Jednotlivé příkazy jsou zpracovávány postupně v takovém pořadí, v jakém byly načteny do seznamu v předcházejícím kroku. Každý příkaz (G1, G28, M107 atd.) má přiřazenou svou vlastní zpracovávající funkci. Pro simulaci je stěžejní podpora pohybových a tiskových příkazů G1, protože jím je definováno množství a umístění vytlačeného termoplastu. Pokud je funkce pro daný příkaz definována, je spuštěna a jako argument je jí předán seznam

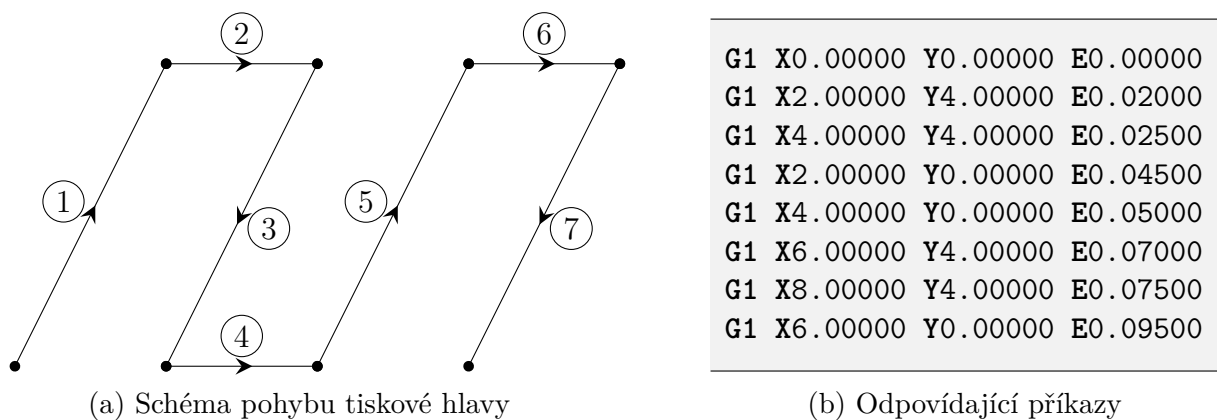


Obrázek 16: Princip zpracování souboru G-code

parametrů daného příkazu. Tímto systémem je možné emulátor tisku přizpůsobit i pro další metody 3D tisku nebo ho jinak rozšířit.

3.2.1 Pohybové příkazy

Pro další analýzu jsou nejdůležitějšími příkazy ovlivňující pohyb tiskové hlavy v prostoru a posun *filamentu*. Do nového seznamu budou postupně přidávány pohyby tiskárny včetně některých vypočtených charakteristik daného pohybu (délka posunu, doba trvání, vytištěná délka *filamentu* aj.). Průběžným procházením jednotlivých příkazů jsou také zjištěny další užitečné informace jako např. celkový objem *filamentu*, celková doba tisku a informace o jednotlivých vrstvách (výška a doba tisku).



Obrázek 17: Zpracování pohybových příkazů G-code

Jednotlivé pohybové příkazy jsou postupně ukládány do seznamu. Přehled ukládaných informací je ukázán v tab. 2. Na základě tohoto seznamu lze provést sestavení rastrového

modelu resp. výpočet postupné aktivace a zaplnění jednotlivých elementů. Zpracování pohybových příkazů stačí provést pro každý G-code pouze jednou.

Informace	Označení
Souřadnice počátečního bodu	A
Souřadnice koncového bodu	B
Uražená vzdálenost	L_i
Čas na začátku pohybu	T_{start}
Čas na konci pohybu	T_{end}
Použitá délka filamentu	ΔE_i
Šířka depozice	w_i

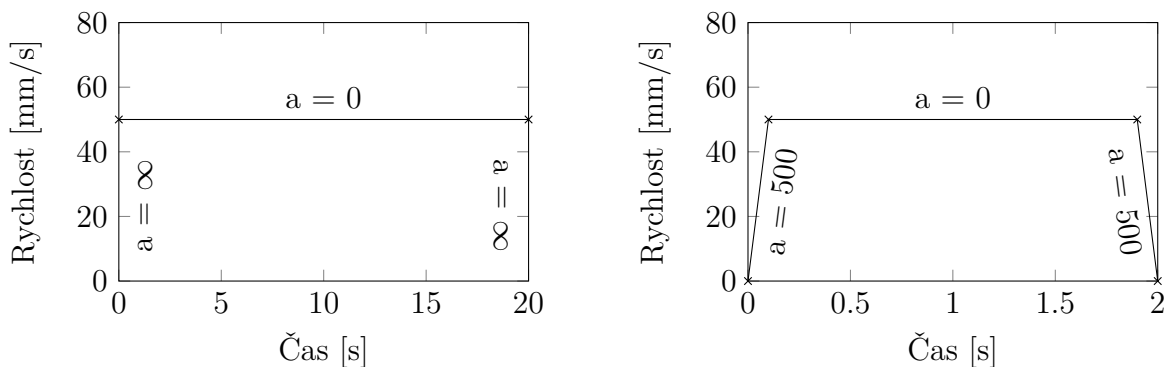
Tabulka 2: Informace o jednotlivých pohybových příkazech

V následujících podsekcích je uveden postup určení všech důležitých informací o jednotlivých pohybech.

3.2.1.1 Délka trvání pohybu

Rychlost pohybu tiskové hlavy je určena parametrem F (Feedrate). Jedná se o hodnotu rychlosti zadanou v mm/min (případně inch/min), kterou chceme dosáhnout při pohybu mezi dvěma body.

V nejjednodušším případě se dá uvažovat průběh rychlosti konstantní roven aktuální hodnotě *feedrate*. Užití konstantního průběhu zanedbává zrychlování a zpomalování na začátku a konci pohybu. Výsledný čas tisku bude podhodnocen.



(a) Konstantní rychlost

(b) Zohlednění zrychlování/zpomalování

Obrázek 18: Průběh rychlosti pohybu po jeho délce

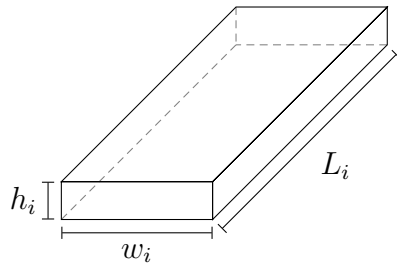
Pro přesnější výpočet je možné na začátku a konci každého pohybu zohlednit lineární nárůst rychlosti resp. konstantní průběh zrychlení (obr. 18b).

Procesor G-code umožní zvolit jeden z těchto způsobů výpočtu. Při zadání zrychlení 0 mm/s² bude program uvažovat nekonečnou hodnotu zrychlení, výchozí hodnotou je obvyklých 500 mm/s². Ve 4. kapitole je pro několik modelů porovnává skutečně naměřená doba tisku s výpočtem uvažujícím oba průběhy rychlosti.

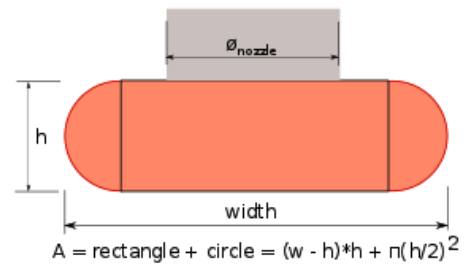
Na základě přehledu o délce tisku jednotlivých vrstev a celkové délce tisku bude možné lépe odhadnout časový krok pro vytvoření konečně-prvkového modelu.

3.2.1.2 Výpočet šířky depozice

Pro každý pohyb, při kterém dochází k depozici materiálu, je potřeba určit tvar a rozměry příčného řezu této depozice. Jedná se v zásadě o opačnou úlohu k procesu *slicingu*, kdy *slicer* určuje kolik je potřeba vytlačit materiálu z trysky na základě požadované šířky depozice. Jednotlivé *slicery* běžně pracují s jednoduchým příčným řezem tvaru obdélníku, ale například Slic3r uvažuje zaoblení rohů dle obr. 20.



Obrázek 19: Geometrie depozice



Obrázek 20: Příčný řez depozice dle programu Slic3r. Převzato z: [18].

Aby byl postup určení šířky depozice co nejvíce univerzální a nezávislý na konkrétním *sliceru*, bude pro výpočet šířky depozice použit zjednodušený příčný řez tvaru obdélníku (viz obr. 19). V prostoru jde o kvádr, jehož délka je rovna délce pohybu L_i a výška tloušťky aktuální vrstvy h_i . Jediným neznámým rozměrem je šířka w_i . Objem vstupující do trysky spočítáme na základě známé délky spotřebovaného filamentu ΔE , a jeho kruhového průřezu o průměru D .

$$V_{in} = \frac{\pi D^2}{4} \Delta E$$

$$V_{out} = w_i h_i L_i$$

$$V_{in} = V_{out}$$

Neznámou šířku depozice w_i získáme jejím vyjádřením z rovnosti objemů:

$$w_i = \frac{\pi D^2}{4 h_i L_i} \Delta E$$

3.2.1.3 Výpočet informací o vrstvách

Proces tisku probíhá po jednotlivých vrstvách realizovaných ve vodorovné rovině XY. Pohyb ve svislé ose Z slouží pouze k přechodu do vyšší vrstvy. Pokud je tedy při pohybu zjištěn posun ve svislém směru, určí se z rozdílu souřadnic tloušťka další vrstvy. Pro předchozí vrstvu se v tomto okamžiku uloží informace o celkové době tisku této vrstvy a délce spotřebovaného filamentu. Pro každou vrstvu je také kromě její tloušťky uložena také hodnota souřadnice z jejího horního povrchu. Přehled informací zjištěných o jednotlivých vrstvách je uveden v tab. 3.

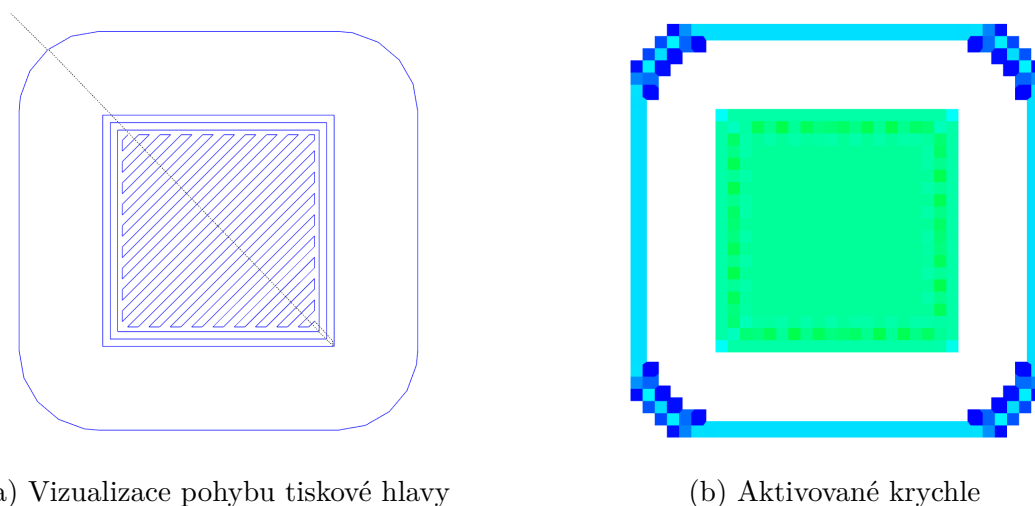
Informace	Označení
Tloušťka vrstvy	h_i
Absolutní výška vrstvy	z_i
Doba tisku vrstvy	t_i
Čas na konci tisku vrstvy	t_{end}

Tabulka 3: Informace o jednotlivých vrstvách

3.3 Aktivace kostek v časových krocích

Dalším krokem je aktivace a postupné zaplňování kostek rastrového modelu v jednotlivých okamžicích dle zvoleného časového kroku Δt . Je potřeba projít postupně všechny pohyby tiskárny a rozdělit objem každého pohybu do odpovídajících kostek zjednodušeného modelu.

Obecně jde o problém promítnutí vytištěného objemu (kvádrů) do rastrového modelu a určení příslušných objemových podílů pro každý tiskový pohyb. Výsledek aktivace celé jedné vrstvy modelu krychle je ukázán na obr. 21.



Obrázek 21: Ukázka zpracovaného G-code jedné vrstvy krychle

Při přidělení objemu do jednotlivých kostek je také potřeba zkontrolovat, zda se v dané kostce nenachází více než 100% objemu. Případný přebytek je třeba rozdělit rovnoměrně do sousedních krychlí.

3.3.1 Přidělení objemu elementům rastrového modelu

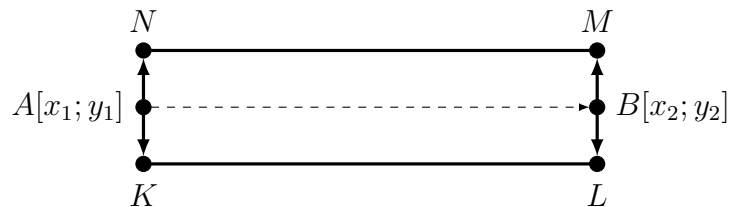
Pro každý pohyb tiskové hlavy je potřeba určit přidělení natištěného objemu do jednotlivých krychlí rastrového modelu. Toho lze dosáhnout spočtením půdorysného průniku depozice a jednotlivých půdorysů elementů modelu (čtverců). Zjištěná plocha průniku se poté rozdělí po výšce do zasažených krychlí.

3.3.1.1 Určení půdorysných souřadnic depozice

Pro každý tiskový pohyb známe jeho vektor posunutí a vypočtenou šířku depozice. Pro výpočet půdorysného sjednocení obrazců je potřeba vypočítat přesné souřadnice obdélní-

kového půdorysu depozice.

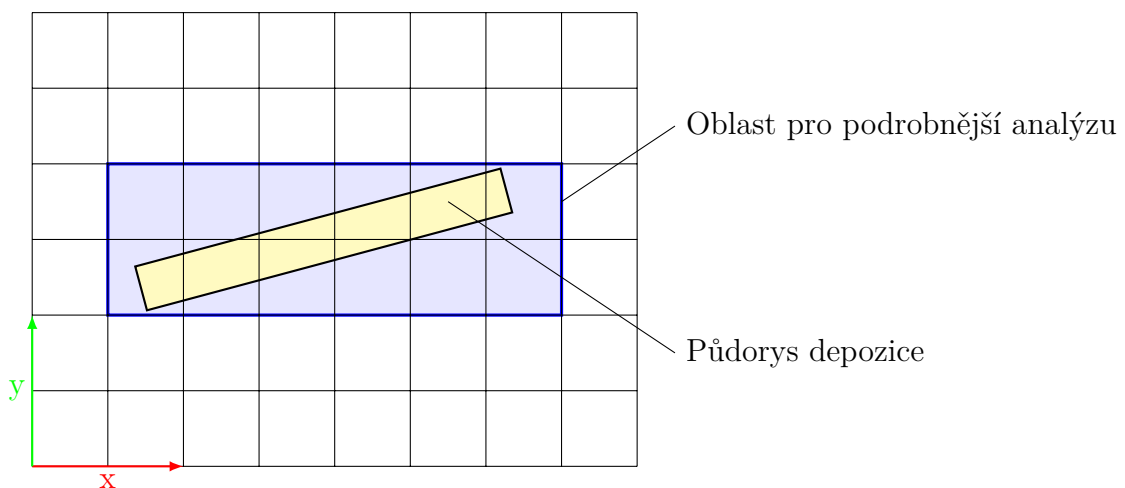
Potřebujeme přičíst a odečíst k počátečnímu a koncovému bodu pohybu vektor kolmý k vektoru pohybu o velikosti poloviny šířky depozice. V rovině lze vektor kolmý na vektor pohybu získat prohozením jeho souřadnic a změnou znaménka jedné z nich. Tento kolmý vektor je následně potřeba znormalizovat a jeho složky vynásobit polovinou šířky depozice. Tento vektor již můžeme přičíst a odečíst k počátečnímu a koncovému bodu pohybu. Tím je určen obdélníkový půdorys depozice.



Obrázek 22: Určení půdorysných souřadnic depozice

3.3.1.2 Určení elementů pro vyplnění

Určení průniku dvou polygonů (deponovaný objem a krychle rastru) je výpočetně náročný proces, je třeba tedy omezit výpočet jen na ty oblasti rastru, kde takový průnik může nastat. K vymezení použijeme tzv. *bounding box* – obdélník ohraničující půdorys depozice v rovině XY. Ten se určí výpočtem maximálních a minimálních souřadnic x a y půdorysu depozice. Tyto souřadnice se poté převedou na celočíselné indexové souřadnice. Následně stačí identifikovat prvky rastru obsažené v *bounding boxu*, to je však velmi levná operace. Tím jsou určeny konkrétní elementy, pro které je potřeba analyzovat přesný průnik s půdorysným obrazcem depozice.

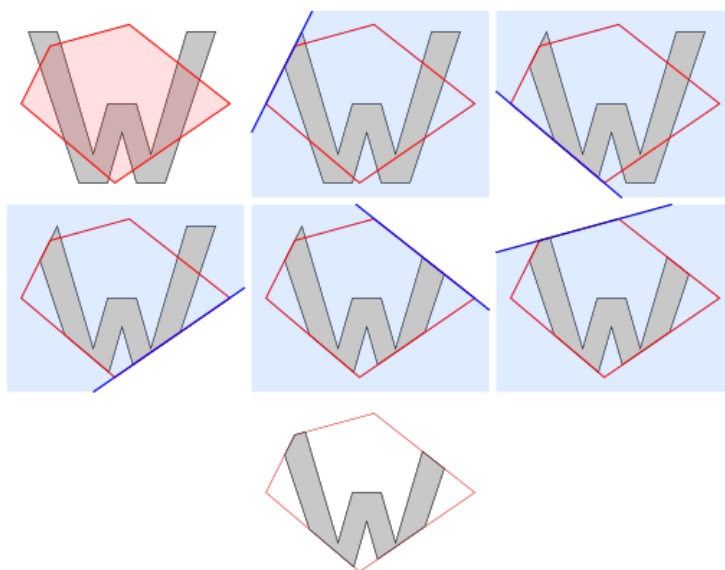


Obrázek 23: Určení elementů pro vyplnění

Jak je vidět na obr. 23, ne všechny elementy nacházející se v ohraničujícím obdélníku musejí být vyplněny. Pro takové elementy musí algoritmus z následující podsečky vrátit nulový průnik.

3.3.1.3 Průnik vytištěného objemu s kostkami modelu

Jak bylo naznačeno na obrázku 23, je potřeba nějakým způsobem spočítat půdorysný průnik vytištěného objemu a jednotlivých krychlí zjednodušeného modelu. Z matematického hlediska se jedná o sjednocení dvou konvexních polygonů (konkrétně obdélníku a čtverce). Pro tento účel existuje celá řada různých algoritmů. V této práci byl použit algoritmus *Sutherland–Hodgman* [19], který funguje na principu postupného ořezávání jednoho polygonu (*subjekt*) nekonečnými hranami toho druhého (*ořezový polygon*). Jako subjekt je uvažován vždy čtvercový půdorys krychle a ořezovým polygonem je půdorys depozice. Souřadnice ořezového polygonu musí být pro správné fungování algoritmu orientovány proti směru hodinových ručiček.



Obrázek 24: Sutherland-Hodgeman – Princip fungování algoritmu

Algoritmus 1 Sutherland-Hodgeman [19]

```

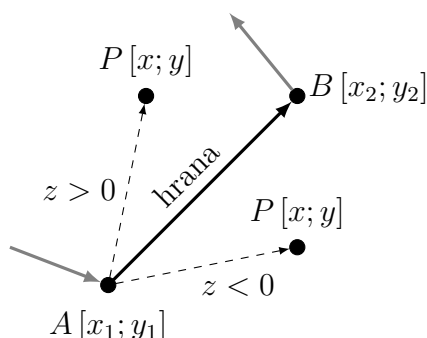
1: List outputList = subjectPolygon;
2: for Edge clipEdge in clipPolygon do           ▷ Iterace přes hrany ořezového polygonu
3:   List inputList = outputList;                 ▷ Ořez polygonu z předchozího kroku
4:   outputList.clear();
5:   Point S = inputList.last;
6:   for Point E in inputList do                 ▷ Iterace všech bodů subjektu
7:     if E inside clipEdge then
8:       if S not inside clipEdge then
9:         outputList.add(intersection(S, E, clipEdge));
10:      outputList.add(E);
11:     else if S inside clipEdge then
12:       outputList.add(intersection(S, E, clipEdge));
13:     S = E;

```

Funkce inside

Pro potřeby algoritmu 1 je nutno definovat funkci *inside*. Ta slouží ke ověření, zda bod ořezávané oblasti (subjektu) leží „uvnitř“ ořezového polygonu od hrany řezu. Protože výše zmíněný algoritmus pracuje s konvexním ořezovým polygonem, lze zjistit pozici bodu spočtením vektorového součinu.

Hrana ořezového polygonu je definována dvěma body AB , testovaný vrchol ořezávané oblasti je označen jako P . Výsledkem vektorového součinu AB a AP je vektor na ně kolmý a svislý, protože vektory AB a AP leží ve vodorovné rovině XY viz obrázek 26. Jedinou nenulovou souřadnicí výsledného vektoru je jeho svislá složka z .



Obrázek 25: Určení, zda-li je bod vpravo či vlevo od přímky

Hodnota z svým znaménkem určuje, na které straně od hrany (vektoru) AB se nachází bod P .

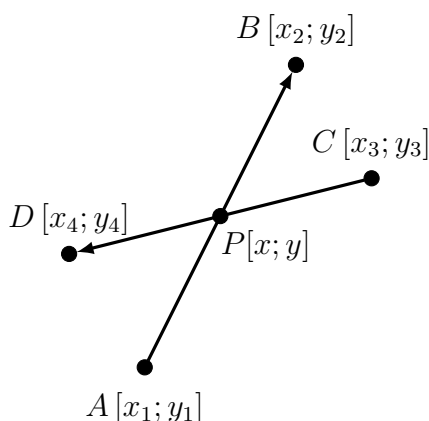
$$z = (x_2 - x_1)(y - y_1) - (y_2 - y_1)(x - x_1)$$

- $z = 0$: Bod P leží na hraně AB
- $z < 0$: Bod P leží vpravo od hrany AB
- $z > 0$: Bod P leží vlevo od hrany AB

Pro mnohoúhelník definovaný proti směru hodinových ručiček znamená vlevo \rightarrow uvnitř.

Funkce intersection

Zároveň musíme definovat i funkci *intersection* pro výpočet průsečíku dvou hran z algoritmu 1. Matematicky se jedná o úlohu výpočtu průsečíku dvou přímek. V tomto případě nemusíme ověřovat, zda průsečík leží na těchto přímkách. Funkce *intersection*, se použije ve výše zmíněném algoritmu pouze, pokud průsečík hran skutečně existuje.



Obrázek 26: Průsečík dvou přímek (definovaných úsečkami)

Pro první přímku AB a neznámý bod P dosadíme do obecné rovnice přímky $ax + by + c = 0$ a tím získáme rovnice pro neznámé a, b, c . Protože P leží na přímce AB , je determinant matice soustavy roven nule. Determinant této matice vyjádříme rozvojem podle řádku s neznámými hodnotami x a y :

$$0 = \begin{vmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x & y & 1 \end{vmatrix} = \begin{vmatrix} y_1 & 1 \\ y_2 & 1 \end{vmatrix} x - \begin{vmatrix} x_1 & 1 \\ x_2 & 1 \end{vmatrix} y + \begin{vmatrix} x_1 & y_1 \\ x_2 & y_2 \end{vmatrix} = 0$$

Protože zároveň P leží i na přímce CD platí analogicky:

$$0 = \begin{vmatrix} x_3 & y_3 & 1 \\ x_4 & y_4 & 1 \\ x & y & 1 \end{vmatrix} = \begin{vmatrix} y_3 & 1 \\ y_4 & 1 \end{vmatrix} x - \begin{vmatrix} x_3 & 1 \\ x_4 & 1 \end{vmatrix} y + \begin{vmatrix} x_3 & y_3 \\ x_4 & y_4 \end{vmatrix} = 0$$

Koeficienty submatic 2×2 známe. Získali jsme soustavu rovnic o dvou neznámých x a y . Jejím vyřešením získáme souřadnice průsečíku dvou přímek:

$$x = \frac{(x_1 y_2 - y_1 x_2)(x_3 - x_4) - (x_1 - x_2)(x_3 y_4 - y_3 x_4)}{(x_1 - x_2)(y_3 - y_4) - (y_1 - y_2)(x_3 - x_4)}$$

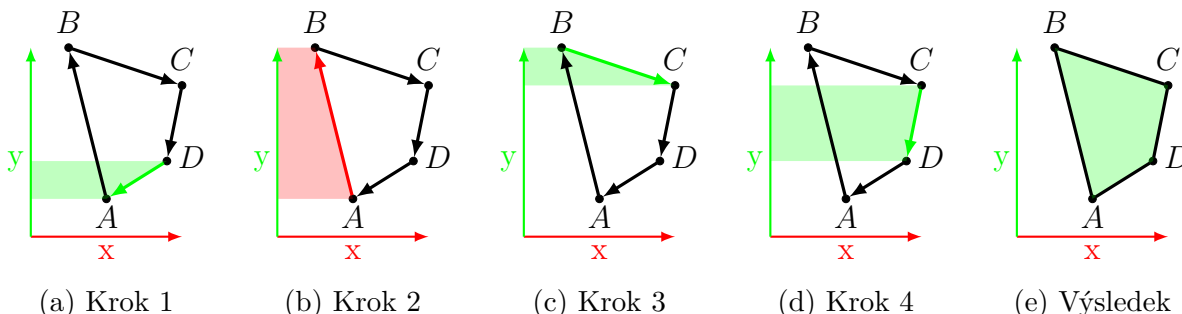
$$y = \frac{(x_1 y_2 - y_1 x_2)(y_3 - y_4) - (x_1 - x_2)(x_3 y_4 - y_3 x_4)}{(x_1 - x_2)(y_3 - y_4) - (y_1 - y_2)(x_3 - x_4)}$$

Jmenovatel je pro obě neznámé stejný. Pokud je roven nule, jsou řešené přímky rovnoběžné či totožné. Při vyhodnocení výrazů by v tomto případě došlo k dělení nulou. Na to je nutné dát pozor zejména při konkrétní implementaci tohoto algoritmu.

3.3.1.4 Výpočet plochy vytištěného obrazce

Pro polygony získané postupně v předchozím kroku je pro každý z nich nutné určit jeho plochu, neboť ta vyjadřuje poměrné zaplnění krychliček rastru. Protože se hrany těchto polygonů vzájemně neprotínají, můžeme použít například algoritmus známý jako *shoelace formula* [20].

Jedná se o postupné sčítání ploch jednotlivých hran polygonu vzhledem k ose Y . Z obrázku 27 je patrné, že plocha toho obrazce je rovna průměru souřadnic X dané hrany vynásobeným rozdílem jeho souřadnic Y . Znaménko dílčí plochy je určeno orientací hrany ve směru Y .



Obrázek 27: Princip výpočtu plochy polygonu

Výsledná plocha může být záporná. Znaménko závisí na tom, zda je polygon definován po směru či proti směru hodinových ručiček. Stačí tedy výsledek převést na jeho absolutní hodnotu.

Algoritmus 2 Shoelace formula [20]

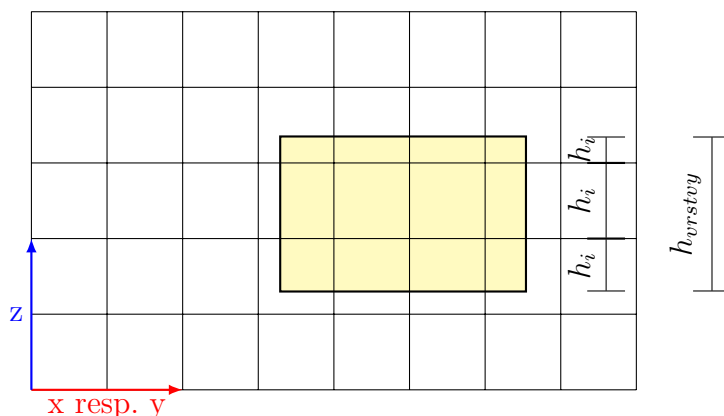
```

1: function POLYGONAREA(points, numPoints)
2:    $area = 0;$  ▷ Celková plocha polygonu
3:    $j = numPoints - 1;$  ▷ Index posledního bodu je předchozí pro první iteraci
4:   for  $i = 1, numPoints$  do
5:      $area = area + (points[j].x + points[i].x) \cdot (points[j].y - points[i].y);$ 
6:     ▷ Plocha úsečky vzhledem k ose y vč. znaménka
7:      $j = i;$  ▷ j je index předchozího bodu
8:    $area = abs(area/2);$  ▷ Výsledná plocha může vyjít záporná
9:   return  $area;$ 

```

3.3.1.5 Rozdělení objemu dle výšky a tloušťky vrstvy

Známe indexové souřadnice elementů z hlediska půdorysu. Pro přidělení objemu musíme zjistit indexové souřadnice elementů ve směru svislé osy Z. Ze známé aktuální výšky a tloušťky vrstvy se podle postupu na obr. 30 snadno přidělí odpovídající část výšky vrstvy odpovídajícím svislým indexovým souřadnicím.



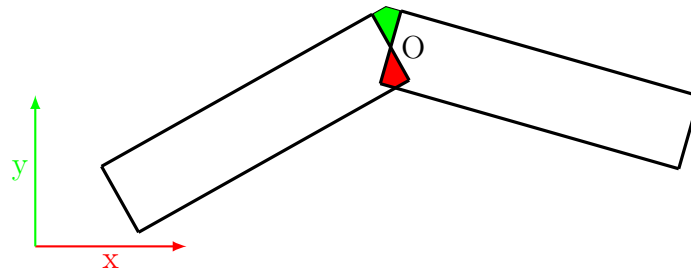
Obrázek 28: Rozdělení vytištěného objemu ve svislé ose

Nyní známe všechny tři indexové souřadnice krychlí společně s přesně definovaným objemem. Přidělený objem se vypočítá pro každý element vynásobením plochy sjednocení půdorysu depozice s danou krychlí a částí výšky vrstvy h_i , která dané krychli připadá.

3.3.1.6 Překryv navazujících depozic

Při uvažování geometrie depozice jako jednoduchého kváдру dochází pro na sebe navazující tiskové pohyby k překryvu jejich půdorysů. To při praktickém výpočtu vede na nesprávné dvojnásobné přidělení objemu (viz červená oblast na obr. 29).

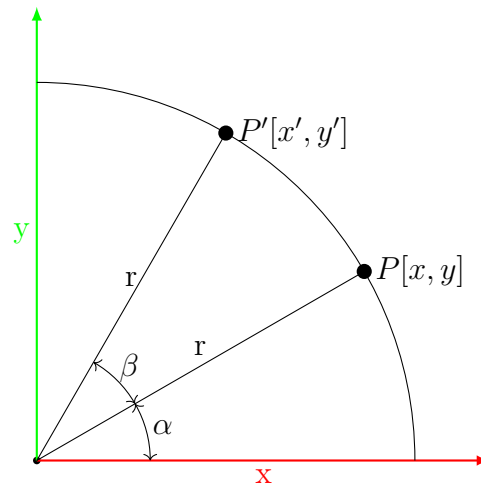
Překryv navazujících depozic se snadno spočte užitím dříve definovaného algoritmu 1. Polovina objemu spadajícího do červené oblasti bude odečtena mechanismem z předchozí podkapitoly užitím záporné hodnoty půdorysné plochy. Tento odečtený objem bude vložen do zelené oblasti otočením polygonu překryvu okolo bodu společného pro oba navazující pohyby o 180° .



Obrázek 29: Oprava překryvu navazujících pohybů

Otočení polygonu okolo bodu

Otočení polygonu okolo bodu O o požadovaný úhel β se provede v několika dílčích krocích. Nejprve se polygon posune, tak aby bod O ležel v počátku souřadného systému. Tento posun je určen vektorem T . Následně se jednotlivé body polygonu otočí o požadovaný úhel okolo počátku souřadného systému. Po otočení se polygon posune zpět o vektor $-T$.



Obrázek 30: Otočení bodu o úhel okolo počátku

Otočení bodu P okolo počátku probíhá po kružnici o poloměru r . Souřadnice bodu před otočením jsou určeny v závislosti na úhlu α :

$$x = r \cos \alpha$$

$$y = r \sin \alpha$$

Souřadnice bodu P po otočení o úhel β jsou:

$$x' = r \cos(\alpha + \beta)$$

$$y' = r \sin(\alpha + \beta)$$

Užitím vztahů pro goniometrické funkce můžeme otočené souřadnice vyjádřit jako:

$$\begin{aligned}
 x' &= r(\cos \alpha \cdot \cos \beta - \sin \alpha \cdot \sin \beta) \\
 &= r \cos \alpha \cdot \cos \beta - r \sin \alpha \cdot \sin \beta \\
 y' &= r(\sin \alpha \cdot \cos \beta + \cos \alpha \cdot \sin \beta) \\
 &= r \sin \alpha \cdot \cos \beta + r \cos \alpha \cdot \sin \beta
 \end{aligned}$$

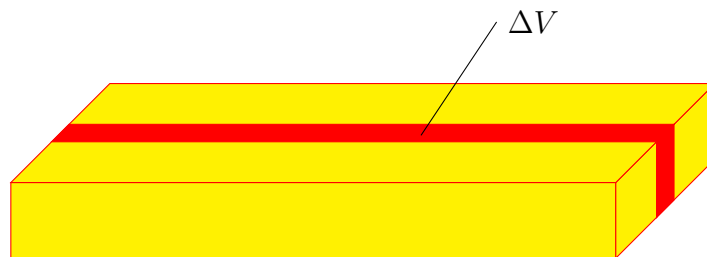
Nyní můžeme dosadit původní souřadnice bodu P :

$$\begin{aligned}
 x' &= x \cos \beta - y \sin \beta \\
 y' &= y \cos \beta + x \sin \beta
 \end{aligned}$$

Toto otočení je provedeno pro všechny body polygonu. Následně jsou body posunuty zpět translačním vektorem $-T$.

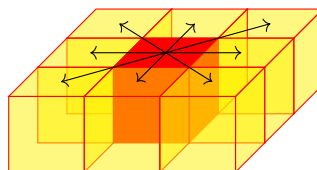
3.3.2 Přerozdělení objemu u přeplněných prvků

Původní předpoklad, že objemové zaplnění elementů rastru přesahující 100% bylo vyřešeno postupem uvedeným v 3.3.1.6, se nepotvrdil. V praxi se ukázalo, že *slicery* běžně produkují takový G-Code, že dochází k mírnému překryvu tiskových pohybů. To by znamenalo, že elementy rastrového modelu by mohly být zaplněny na více než 100%.



Obrázek 31: Překryv depozic

Při zaplňování jednotlivých krychlí se kontroluje, zda není překročeno objemové vyplnění 100%. V případě přeplnění je přebytečný objem ΔV rovnoměrně rozdělen do sousedních elementů ve vodorovné rovině XY , jak je naznačeno na obr. 32. Přebytečný objem je přerozdělen pouze do elementů, které ještě nejsou zaplněné.



Obrázek 32: Přerozdělení objemu u přeplněného elementu

3.4 Výstup vhodný pro MKP software

Předchozím postupem jsme získali informace o postupné aktivaci jednotlivých elementů a jejich objemovém zaplnění v čase. Pro další MKP analýzu je potřeba připravit soubor, který bude obsahovat všechny důležité informace. Tento soubor by mělo být možné jednoduchým zpracováním převést na formát podporovaný konkrétním konečně-prvkovým programem (např. OOFEM).

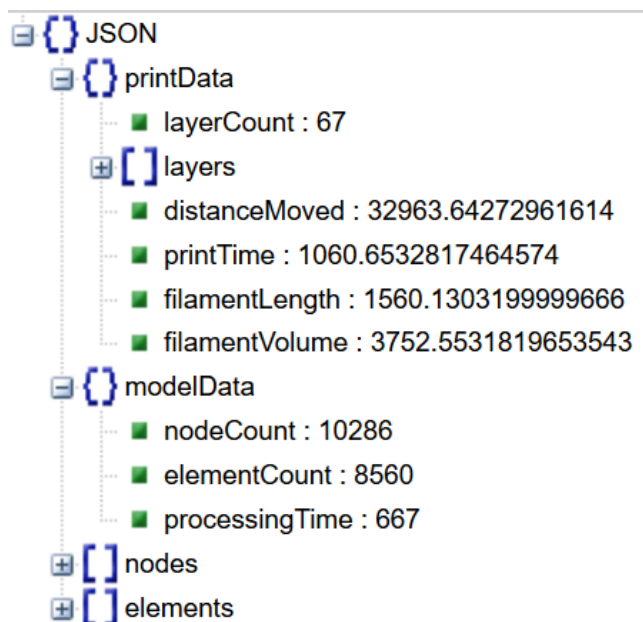
3.4.1 Výstupní formát JSON

Pro uložení vytvořeného rastrového modelu bude použit formát JSON². Jedná se o datový formát nezávislý na operačním systému určený pro přenos dat. Tento formát umožňuje přenos čísel, řetězců, objektů a polí. Obsah je uložen v textovém souboru a není nijak komprimován. Všechny běžně používané jazyky disponují funkcemi pro práci s tímto formátem, případně jsou dostupná otevřená řešení ve formě knihoven.

V poli **printData** jsou uloženy důležité informace o samotném tisku (viz obr. 33). Těmi jsou celková uražená vzdálenost **distanceMoved**, celkový odhadovaný čas tisku **printTime**, délka spotřebovaného filamentu **filamentLength** a celkový vytištěný objem **filamentVolume**.

V poli **modelData** je uložen celkový počet unikátních uzlů **nodeCount** a celkový počet krychelných elementů **elementCount**. Také je zaznamenán čas **processingTime**, který byl potřeba pro emulaci tisku nad rastrovým modelem.

Pole **nodes** obsahuje všechny unikátní uzly a pole **elements** seznam všech elementů.



Obrázek 33: Vizualizace výstupu procesoru G-code

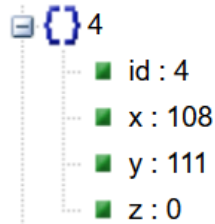
²JavaScript Object Notation – Přestože je název tohoto formátu odvozen z jazyka JavaScript, jedná se o velmi používaný datový formát napříč různými jazyky a platformami.

3.4.2 Formátování uzlů

Jednotlivé uzly jsou definovány svým unikátním číslem **id** a prostorovými souřadnicemi **x**, **y**, **z**, jak je vidět na obr. 34.

Dosud nebylo v G-code procesoru s jednotlivými uzly nijak uvažováno. Jejich souřadnice se zjistí ze seznamu aktivovaných elementů resp. z indexových souřadnic jednotlivých elementů. Při vytváření jednotlivých uzlů se musí postupovat tak, aby nebyly v modelu stejné uzly zdvojeny.

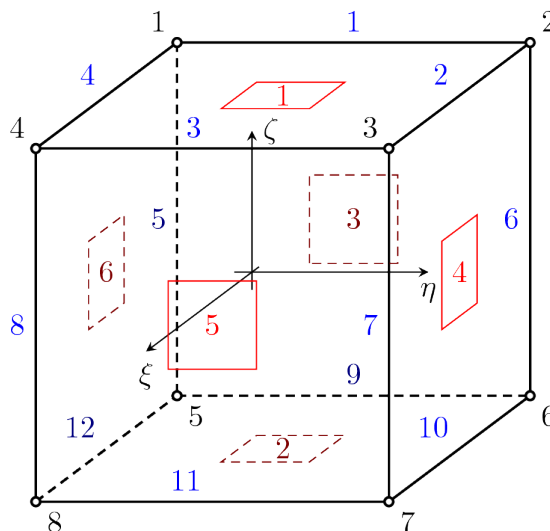
Unikátní číslo **id** uzlu je využito pro definování jednotlivých elementů (viz obr. 36).



Obrázek 34: Ukázka uložených informací o uzlu

3.4.3 Formátování prvků

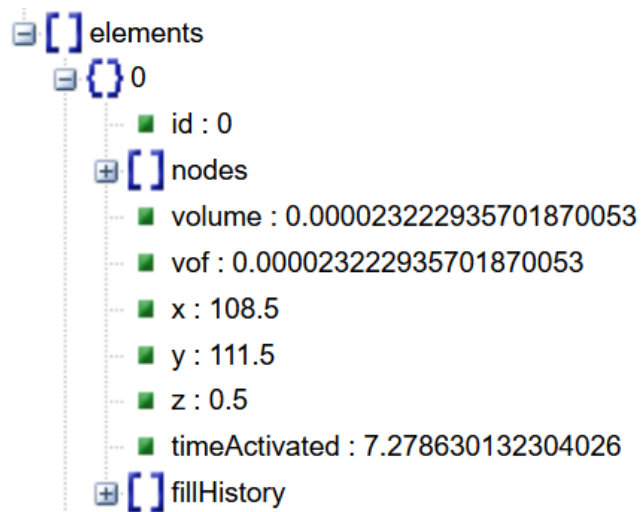
Prvek pro výpočet transportního problému vedení tepla je v případě této práce uvažován jako izoparametrický osmiuzlový hexahedron (krychle). Jednotlivé prvky jsou definovány čísly uzlů, kterými jsou tvořeny. Pořadí uzlů je uvažováno v souladu s prvkem dostupným v programu OOFEM (dle obr. 35).



Obrázek 35: Brick1ht – Element pro transportní problém vedení tepla. Převzato z: [21].

Číslo **id** udává unikátní číslo každého elementu, v poli **nodes** je uloženo 8 různých uzlů definujících element resp. jejich **id** (pořadí viz obr. 35). Informace o celkovém vyplnění elementu je zaznamenána v mm^3 jako **volume** a v procentech jako **vof**. Souřadnice **x**, **y**, **z** elementu udávají pozici jeho těžiště. Ta se může hodit při dalším zpracování např. pro rychlé vykreslení prvku. Zároveň je uložen čas aktivace prvku **timeActivated**. V poli

fillHistory je potom uložena kompletní historie postupného vyplňování. Je tedy možné z jednoho souboru zrekonstruovat situaci pro libovolnou délku časového kroku.



Obrázek 36: Ukázka uložených informace o elementu

4 Příklady

V této kapitole je analyzováno několik objektů, které byly zároveň vytištěny na skutečné tiskárně Creality Ender 3. Nejprve byl zvolen model krychle, který byl v jednom případě uvažován jako kompletně vyplněný a ve druhém případě vyplněn na 20% pomocí *rectilinear infill*. Jako třetí příklad byl vybrán model žáby pro svou geometrickou složitost, který byl uvažován s vyplněním 50%.

Pro přípravu tiskového souboru G-code byl použit program Slic3r. Přehled důležitých parametrů použitých při *slicingu* je uveden v tab. 4.

Základní informace	
Průměr trysky	0,40 mm
Průměr filamentu	1,75 mm
Materiál	ABS
Teplota trysky	235°C
Teplota vyhřívané podložky	105°C
Vrstvy	
První vrstva	0,30 mm
Ostatní vrstvy	0,30 mm
Rychlost pohybu	
Netiskový pohyb	130 mm/s
Perimetr	60 mm/s
Výplň	80 mm/s
Zrychlení při pohybu	500 mm/s ²

Tabulka 4: Důležité parametry sliceru pro přípravu tiskových souborů příkladů

Kromě přehledu základních informací jako je odhadovaná celková doba tisku a množství spotřebovaného filamentu je dle postupu v kapitole 3 vygenerován v jednotlivých časových krocích rastrový model. Jsou uvažovány 3 různé velikosti krychlí o délkách hran 2, 1 a 0,5 mm. Časový krok je ve všech příkladech zvolen jako $\approx 1/12$ odhadovaného celkového času tisku. Tento krok byl zvolen pro názornou vizualizaci postupné aktivace prvků ve 12 krocích, vhodná délka kroku pro analýzu MKP je otázkou dalšího výzkumu. Postupná aktivace prvků je srovnána pro jednotlivá dělení rastru v tabulce. Současně je uvedena orientační časová náročnost pro sestavení jednotlivých rastrových modelů na běžně dostupném PC.

Užitím současně dostupných programů pro analýzu tiskových souborů G-code (gCodeViewer [4], G-Code Analyser [22]) je srovnána celková doba tisku a množství spotřebovaného filamentu. Tento software však bohužel neumožňuje tvorbu modelů vhodných pro analýzu MKP.

4.1 Tiskárna Creality Ender 3

Tiskárna Ender 3 použitá ke skutečnému zhotovení srovnávaných objektů ukazuje na současnou dostupnost 3D tisku i v domácích podmínkách. Přes svou nízkou cenu³ je její součástí dokonce i vyhřívaná podložka. Díky tomu je možné bez problémů tisknout ze všech běžně používaných materiálů (PLA, ABS). Technické parametry této tiskárny jsou uvedeny v tab. 5.



Obrázek 37: Creality Ender 3

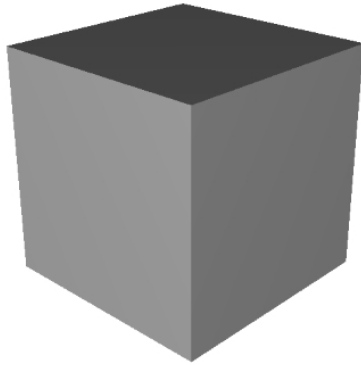
Tisková plocha	220 × 220 × 250 mm
Průměr trysky	0,4 mm
Min. výška vrstvy	0,1 mm
Přesnost tisku	±0,1 mm
Průměr filamentu	1.75 mm
Filament	PLA, ABS, TPU
Max. rychlost tisku	200 mm/s
Max. zrychlení pohybu	500 mm/s ²
Vyhřívaná podložka	ano, max. 110 °C
Přenos dat	SD karta, USB

Tabulka 5: Ender 3 – Technické parametry

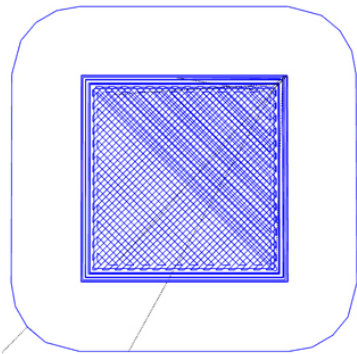
³V listopadu 2018 se cena této tiskárny na internetu pohybovala okolo necelých 4500 Kč.

4.2 Příklad 1: Krychle 20 × 20 × 20 mm

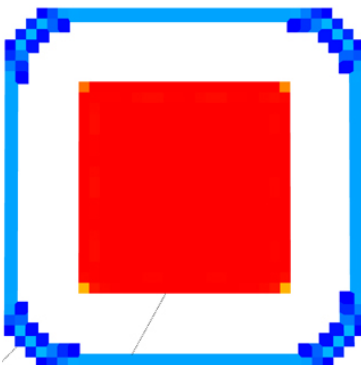
Jako první byl zpracován model krychle 20 × 20 × 20 mm. Jedná se prakticky o tvarově nejjednodušší objekt, který lze vytisknout. Často se používá pro kalibraci a kontrolu správného fungování 3D tiskárny.



(a) STL model



(b) G-code 1. vrstvy



(c) Aktivace 1. vrstvy


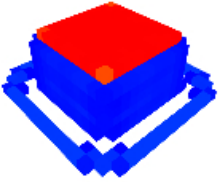
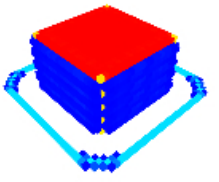
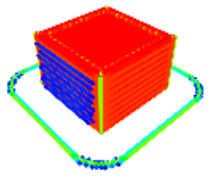
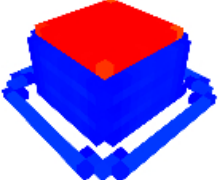
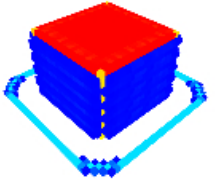
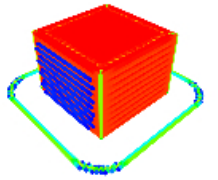
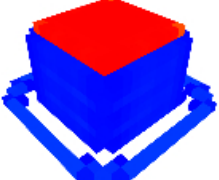
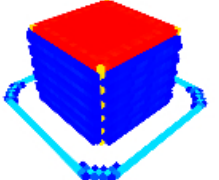
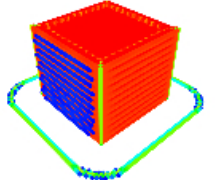
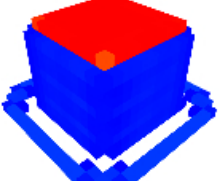
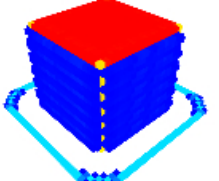
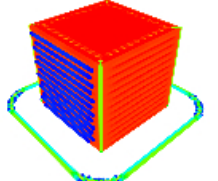
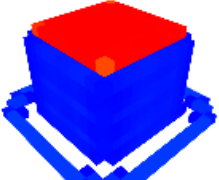
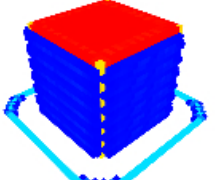
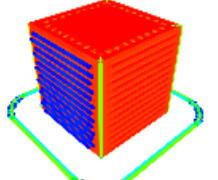

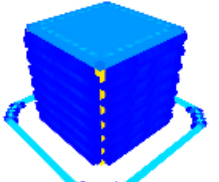
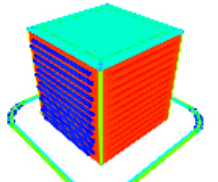
Obrázek 38: Příklad 1

STL model	
Vnější rozměr	20 × 20 × 20 mm
Objem	8 000 mm ³
Trojúhelníky	12
G-code	
Celkový čas tisku	
Processor G-code ($a = \infty$)	1 166 s
Processor G-code ($a = 500$)	2 100 s
G-Code Analyser ($a = 500$)	1 964 s
gCodeViewer ($a = \infty$)	1 367 s
Naměřená hodnota	1 740 s
<i>a je zrychlení v mm/s²</i>	
Délka filamentu	
Processor G-code	3 394 mm
G-Code Analyser	3 390 mm
gCodeViewer	3 390 mm
Vytištěný objem	
Processor G-code	8 163 mm ³
G-Code Analyser	8 160 mm ³
gCodeViewer	8 153 mm ³
Rastrový model	
2 × 2 × 2	
Počet uzlů	1 984
Počet elementů	1 395
Příprava modelu	833 ms
1 × 1 × 1	
Počet uzlů	11 407
Počet elementů	9 179
Příprava modelu	1 660 ms
0,5 × 0,5 × 0,5	
Počet uzlů	72 162
Počet elementů	66 116
Příprava modelu	5 118 ms

Tabulka 6: Souhrn informací

4.2.1 Postupná aktivace elementů

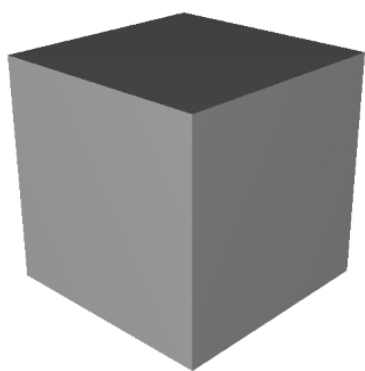
Objemové vyplnění krychle [%]			
Čas [s]	$2 \times 2 \times 2$	$1 \times 1 \times 1$	$0,5 \times 0,5 \times 0,5$
176			
352			
528			
704			
880			
1056			

Objemové vyplnění krychle [%]			
0  50 100			
Čas [s]	$2 \times 2 \times 2$	$1 \times 1 \times 1$	$0,5 \times 0,5 \times 0,5$
132,			
1 408			
1 584			
1 760			
1 936			
2 100			

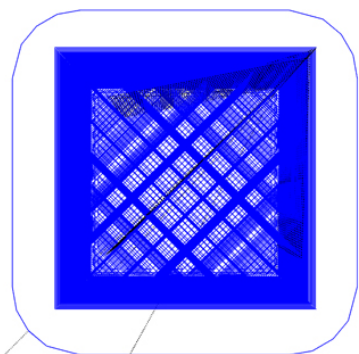
Tabulka 7: Postupná aktivace elementů

4.3 Příklad 2: Krychle $20 \times 20 \times 20$ mm řídce vyplněná

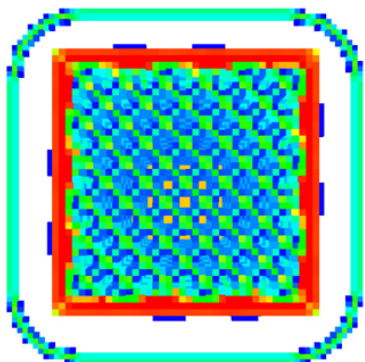
Druhým příkladem je krychle $20 \times 20 \times 20$ mm. Není ale použito 100% vyplnění objemu jako v 1. příkladě. Je použit tzv. 20% „rectilinear infill“. V takovém případě je vnitřek krychle vyplněn sérií rovnoběžných pohybů jejichž směr se liší ob vrstvu o 90° .



(a) STL model



(b) G-code v polovině tisku



(c) Aktivace v polovině tisku

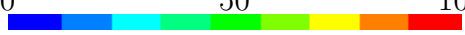
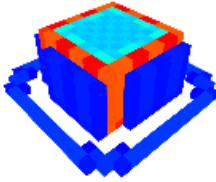
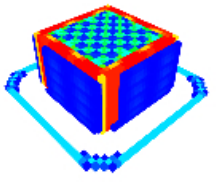
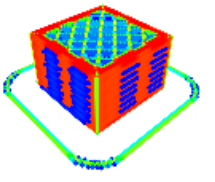
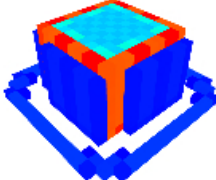
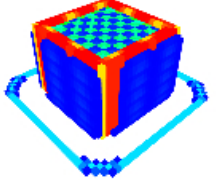
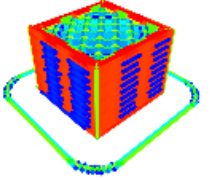
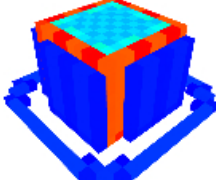
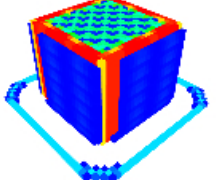
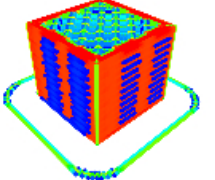

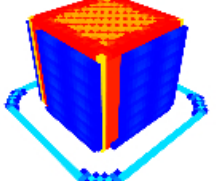
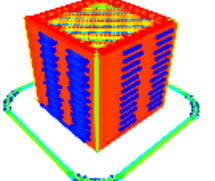

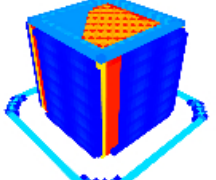
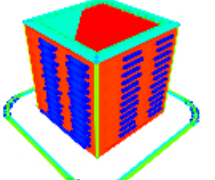

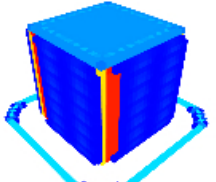
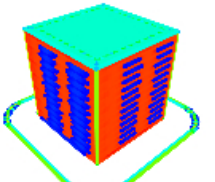
Obrázek 39: Příklad 2

STL model	
Vnější rozměr	$20 \times 20 \times 20$ mm
Objem	$8\,000$ mm ³
Trojúhelníky	12
G-code	
Celkový čas tisku	
Processor G-code ($a = \infty$)	701 s
Processor G-code ($a = 500$)	1 061 s
G-Code Analyser ($a = 500$)	1 028 s
gCodeViewer ($a = \infty$)	905 s
Naměřená hodnota	900 s
<i>a je zrychlení v mm/s²</i>	
Délka filamentu	
Processor G-code	1 560 mm
G-Code Analyser	1 560 mm
gCodeViewer	1 556 mm
Vytisknutý objem	
Processor G-code	$3\,753$ mm ³
G-Code Analyser	$3\,750$ mm ³
gCodeViewer	$3\,743$ mm ³
Rastrový model	
$2 \times 2 \times 2$	
Počet uzlů	1 984
Počet elementů	1 395
Příprava modelu	833 ms
$1 \times 1 \times 1$	
Počet uzlů	xxxx
Počet elementů	560
Příprava modelu	1 660 ms
$0,5 \times 0,5 \times 0,5$	
Počet uzlů	71 022
Počet elementů	57 860
Příprava modelu	2 491 ms

Tabulka 8: Souhrn informací

4.3.1 Postupná aktivace elementů

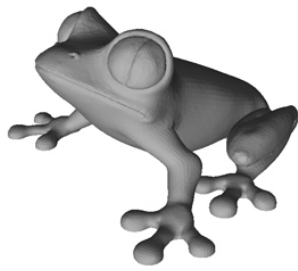
Objemové vyplnění krychle [%]			
Čas [s]	$2 \times 2 \times 2$	$1 \times 1 \times 1$	$0,5 \times 0,5 \times 0,5$
89			
178			
267			
356			
445			
534			

Objemové vyplnění krychle [%]			
0  100			
Čas [s]	$2 \times 2 \times 2$	$1 \times 1 \times 1$	$0,5 \times 0,5 \times 0,5$
623			
712			
801			
890			
979			
1061			

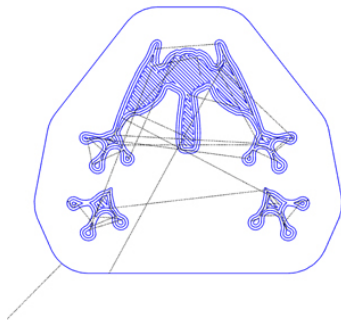
Tabulka 9: Postupná aktivace elementů

4.4 Příklad 3: Žába

Posledním příkladem je model žáby. Tento geometricky komplikovaný objekt se v 3D tisku používá jako zkušební výtisk. Pro urychlení výtisku je použito vyplnění 50% *rectilinear infillem*.



(a) STL model



(b) G-code 1. vrstvy



(c) Aktivace 1. vrstvy


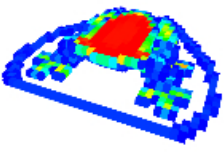
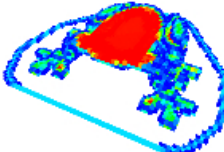
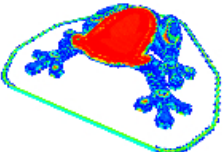
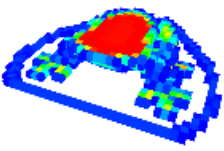
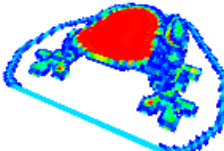
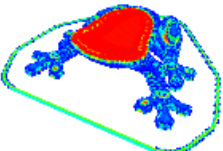
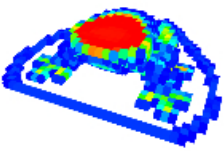
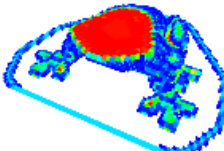
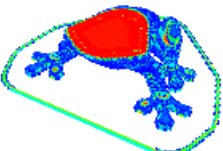
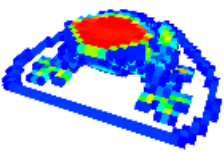
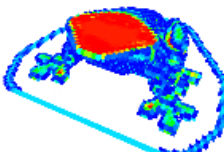
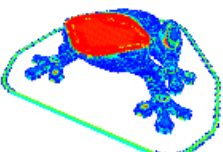
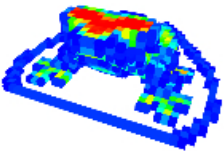
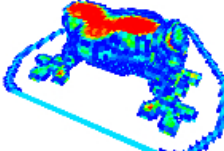
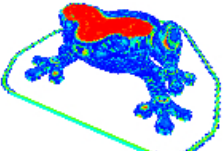
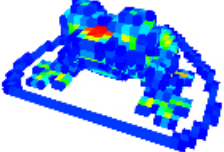
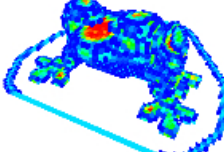
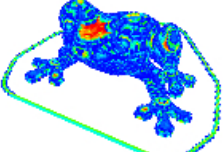
Obrázek 40: Příklad 3

STL model	
Vnější rozměr	51,23 × 45,21 × 25,01 mm
Objem	10 460 mm ³
Trojúhelníky	46 708
G-code	
Celkový čas tisku	
Processor G-code ($a = \infty$)	1 770 s
Processor G-code ($a = 500$)	5 103 s
G-Code Analyser ($a = 500$)	4 812 s
gCodeViewer ($a = \infty$)	1 823 s
Naměřená hodnota	4 593 s
<i>a je zrychlení v mm/s²</i>	
Délka filamentu	
Processor G-code	3 288 mm
G-Code Analyser	3 284 mm
gCodeViewer	3 290 mm
Vytištěný objem	
Processor G-code	7 909 mm ³
G-Code Analyser	7 899 mm ³
gCodeViewer	7 910 mm ³
Rastrový model	
2 × 2 × 2	
Počet uzlů	2,165
Počet elementů	3 516
Příprava modelu	833 ms
1 × 1 × 1	
Počet uzlů	17 972
Počet elementů	13 570
Příprava modelu	2 667 ms
0,5 × 0,5 × 0,5	
Počet uzlů	110 373
Počet elementů	95 022
Příprava modelu	8 031 ms

Tabulka 10: Souhrn informací

4.4.1 Postupná aktivace elementů

Objemové vyplnění krychle [%]			
Čas [s]	$2 \times 2 \times 2$	$1 \times 1 \times 1$	$0,5 \times 0,5 \times 0,5$
426			
852			
1278			
1704			
2130			
2556			

Objemové vyplnění krychle [%]			
0  50 100			
Čas [s]	$2 \times 2 \times 2$	$1 \times 1 \times 1$	$0,5 \times 0,5 \times 0,5$
2 982			
3 408			
3 834			
4 260			
4 686			
5 103			

Tabulka 11: Postupná aktivace elementů

5 Závěr

Předložená práce se zabývá návrhem a implementací softwarového nástroje, který umožní simulovat proces 3D tisku nad rastrovým modelem a poskytnout výstup vhodný pro generování vstupního modelu pro numerickou simulaci procesu tisku. Motivací práce je řada problémů, které vznikají především vysokými teplotními gradienty v tištěném výrobku.

V 2. kapitole práce představuje současně používané technologie 3D tisku, princip výrobního procesu a řídicí kód 3D tiskáren ve formátu G-code. V kapitole 3 je následně podrobně vysvětlen postup tvorby rastrového modelu tvořeného prostorovými krychlemi o volitelné délce hrany a volitelné délce časového kroku. Dále jsou představeny základní algoritmy potřebné pro simulaci řídicího kódu nad rastrovým modelem. Pro každý element rastru je zjištěn čas a míra vyplnění tiskovým materiálem. Vytvořený softwarový nástroj umožňuje export do souboru ve formátu JSON. Tento soubor obsahuje všechny informace potřebné pro sestavení vstupního souboru pro analýzu metodou konečných prvků (např. v programu OOFEM).

V závěru práce je ilustrováno použití vytvořeného nástroje na třech různých modelech. Pro vytvoření tiskových souborů byly použity parametry tiskárny Ender 3 a materiál ABS. U každého z modelů je názorně ukázána postupná aktivace a vyplňování jednotlivých elementů rastrového modelu ve 12 krocích pro 3 různé délky hrany krychle. Všechny objekty byly na této tiskárně skutečně vytištěny a tak byl porovnán odhadovaný čas tisku s realitou. Lze konstatovat, že odhadovaná doba tisku bez uvažování zrychlování a zpomalování je příliš podhodnocena. Naopak při uvažování lineárního průběhu zrychlení je dosaženo vyšších než naměřených hodnot o 10-20%. Pro přesnější výsledek by bylo potřeba analyzovat zdrojový kód firmware konkrétní tiskárny a emulovat pohyb přesně stejným způsobem. Spotřebovaná délka filamentu a vytištěný objem byl v souladu napříč porovnávanými nástroji.

5.1 Motivace pro další výzkum

Počet elementů se u příkladů v kapitole 4 pohybuje okolo 70 000 – 100 000 při použití rastru z krychlí $0,5 \times 0,5 \times 0,5$ mm. Dá se předpokládat, že analýza nestacionárního problému vedení tepla bude na takové síti pomocí běžně dostupných PC časově náročnou, ale řešitelnou operací. Obecně bude žádoucí analyzovat proces 3D tisku pomocí termokamery. Na základě výsledků měření by mělo být možné určit potřebnou jemnost rastru i délku časového kroku a kalibrovat numerický model tak, aby bylo numerickou simulací dosaženo výsledků srovnatelných se skutečností.

Cílem dalšího výzkumu by mohlo být vytvoření výpočetního nástroje, který by se napojil do současného procesu 3D tisku — např. integrováním do některého z volně dostupných *slicerů*. Tento nástroj by následně mohl pomoci s optimalizací generovaného řídicího kódu 3D tiskáren.

Literatura

1. FORMLABS. *Stereolithography 3D Printing: From the 1980s to Now* [online] [cit. 6.11.2018]. Dostupné z: <https://formlabs.com/blog/history-of-stereolithography-3d-printing>.
2. REPRAP. *About — RepRap*, [online] [cit. 6.12.2018]. Dostupné z: <https://reprap.org/wiki/About>.
3. JOSHI, Sunil C.; SHEIKH, Abdullah A. 3D printing in aerospace and its long-term sustainability. *Virtual and Physical Prototyping*. 2015, roč. 10, č. 4, s. 175–185. Dostupné z DOI: 10.1080/17452759.2015.1111519.
4. USTYANTSEV, Alex. *gCodeVisualizer - a web-based visual gcode viewer and analyzer* [online] [cit. 11.12.2018]. Dostupné z: <https://github.com/hudbrog/gCodeViewer>.
5. TYSON, Ed. *Prints Pillowing, 3D Printer Stringing and Layers Splitting – 3 Worst 3D Printer Problems Resolved* [online] [cit. 27.12.2018]. Dostupné z: <https://rigid.ink/blogs/news/pillowing-stringing-and-splitting-what-they-are-and-how-to-stop-them>.
6. 3D SYSTEMS. *Our Story* [online] [cit. 2.12.2018]. Dostupné z: <https://www.3dsystems.com/our-story>.
7. G2G COMMUNITIES. *3D Printed Cog* [online] [cit. 6.11.2018]. Dostupné z: <https://g2gcommunities.org/sustainable-3d-printing/attachment/3d-print-cog/>.
8. QUARTZ. *You can now walk across the world's first 3D-printed steel bridge* [online] [cit. 6.11.2018]. Dostupné z: <https://qz.com/1432622/a-robot-built-the-worlds-first-3d-printed-steel-bridge/>.
9. 3DNATIVES. *China creates their first 3D printed bridges* [online] [cit. 6.11.2018]. Dostupné z: <https://www.3dnatives.com/en/china-3d-printed-bridge310720174/>.
10. MY 3D CONCEPTS. *How 3D Printing Works* [online] [cit. 17.12.2018]. Dostupné z: <http://my3dconcepts.com/explore/how-3d-printing-works/>.
11. BIEHLER, John; FANE, Bill. *3D Printing with Autodesk: Create and Print 3D Objects with 123D, AutoCAD and Inventor*. Que Publishing, 2014. ISBN 978-0-7897-5328-1.
12. SCHLÖMER, Nico. *meshio* [online] [cit. 16.12.2018]. Dostupné z: <https://github.com/nschloe/meshio>.
13. OZEKI. *Introduction to 3D printing* [online] [cit. 19.12.2018]. Dostupné z: <http://www.ozeki.hu/index.php?owpn=1018>.
14. SCULPTEO. *The State of 3D Printing* [online] [cit. 11.12.2018]. Dostupné z: https://www.sculpteo.com/media/ebook/State%20of%203DP%202017_1.pdf.
15. SAKLY, Adnene. *Fabrication additive de pièces à base d'alliages métalliques complexes*. 2013. Dostupné z DOI: 10.13140/RG.2.1.3370.4721. Disertační práce.

16. SIMPLIFY3D. *3D Printing Material Properties Table - Compare Top Filaments* [online] [cit. 2.1.2019]. Dostupné z: <https://www.simplify3d.com/support/materials-guide/properties-table/>.
17. REPRAP. *G-code — RepRap*, [online] [cit. 18.10.2018]. Dostupné z: <https://reprap.org/wiki/G-code>.
18. HODGSON, Gary; RANELLUCCI, Alessandro; MOE, Jeff. *Slic3r Manual: Flow Math* [online] [cit. 11.12.2018]. Dostupné z: <https://manual.slic3r.org/advanced/flow-math>.
19. SUTHERLAND, Ivan E.; HODGMAN, Gary W. Reentrant Polygon Clipping. *Commun. ACM*. 1974, roč. 17, č. 1, s. 32–42. ISSN 0001-0782. Dostupné z DOI: 10.1145/360767.360802.
20. BRADEN, Bart. The Surveyor's Area Formula. *The College Mathematics Journal*. 1986, roč. 17, č. 4, s. 326–337. ISSN 07468342, 19311346. ISSN 07468342, 19311346. Dostupné také z: <http://www.jstor.org/stable/2686282>.
21. BOŘEK, Patzák. *OOFEM Element Library Manual – Elements for Transport problems (TM Module)* [online] [cit. 24.12.2018]. Dostupné z: <http://www.oofem.org/resources/doc/elementlibmanual/html/elementlibmanualsu17.html#x23-770003.3>.
22. GCODEANALYSER.COM. *G-Code Analyser* [online] [cit. 2.1.2019]. Dostupné z: <http://www.gcodeanalyser.com/>.

Seznam obrázků

1	Typické problémy související s deformací rozdílem teploty	2
2	SLA-1 – První komerční 3D tiskárna na světě	3
3	RepRap – Tiskárna schopná replikovat své plastové části	3
4	Příklad využití 3D tisku v praxi	4
5	Schéma výrobního procesu 3D objektu	4
6	OpenSCAD	5
7	Princip reprezentace mezikruží ve formátu STL	6
8	Slicing - proces rozřezání objektu na vrstvy po výšce	7
9	Statistika používaných technologií 3D tisku	9
10	Stereolitografie – „Right-Side Up“	9
11	Stereolitografie – „Upside-Down“	10
12	Selective laser sintering	11
13	Příklad skutečného souboru G-code	12
14	Převedení vektorového modelu G-code na rastrový model z krychlí	15
15	Schéma fungování procesoru G-code	16
16	Princip zpracování souboru G-code	17
17	Zpracování pohybových příkazů G-code	17
18	Průběh rychlosti pohybu po jeho délce	18
19	Geometrie depozice	19
20	Příčný řez depozice dle programu Slic3r	19
21	Ukázka zpracovaného G-code jedné vrstvy krychle	20
22	Určení půdorysných souřadnic depozice	21
23	Určení elementů pro vyplnění	21
24	Sutherland-Hodgeman – Princip fungování algoritmu	22
25	Určení, zda-li je bod vpravo či vlevo od přímky	23
26	Průsečík dvou přímek (definovaných úsečkami)	23
27	Princip výpočtu plochy polygonu	24
28	Rozdělení vytištěného objemu ve svislé ose	25
29	Oprava překryvu navazujících pohybů	26
30	Otočení bodu o úhel okolo počátku	26
31	Překryv depozic	27
32	Přerozdělení objemu u přeplněného elementu	27
33	Vizualizace výstupu procesoru G-code	28
34	Ukázka uložených informací o uzlu	29
35	Brick1ht – Element pro transportní problém vedení tepla	29
36	Ukázka uložených informace o elementu	30
37	Creality Ender 3	32
38	Příklad 1	33
39	Příklad 2	36
40	Příklad 3	39

Seznam tabulek

1	Přehled používaných materiálů – FDM	11
2	Informace o jednotlivých pohybových příkazech	18
3	Informace o jednotlivých vrstvách	20
4	Důležité parametry sliceru pro přípravu tiskových souborů příkladů	31
5	Ender 3 – Technické parametry	32
6	Příklad 1 – Souhrn informací	33
7	Příklad 1 – Postupná aktivace elementů	35
8	Příklad 2 – Souhrn informací	36
9	Příklad 2 – Postupná aktivace elementů	38
10	Příklad 3 – Souhrn informací	39
11	Příklad 3 – Postupná aktivace elementů	41

Seznam algoritmů

1	Sutherland-Hodgeman [19]	22
2	Shoelace formula [20]	25

Přílohy

A Obsah přiloženého CD

Přiložené CD obsahuje následující složky:

<code>/examples-gcode</code>	soubory G-code použité k analýze příkladů v kapitole 4
<code>/gcode-processor</code>	procesor G-code vč. zdrojových kódů
<code>/gcode-processor/src</code>	spustitelná verze
<code>/gcode-processor/js</code>	zdrojové kódy v jazyce JavaScript
<code>/gcode-processor/scss</code>	soubory kaskádových stylů (CSS) definující design ovládacích prvků aplikace

A.1 Spuštění aplikace

Vytvořený nástroj je připraven jako webová aplikace, proto pro jeho spuštění stačí otevřít soubor `index.html` umístěný v adresáři `gcode-processor/dist/`.

A.2 Sestavení aplikace

Jazyk JavaScript není sice kompilovaným jazykem, přesto je třeba výsledný program z jednotlivých souborů sestavit. Tak jako v Pythonu je k dispozici balíčkovací systém pip, prostředí Node.js používá analogický systém npm.

Pro sestavení aplikace je nutné adresář `gcode-processor/` okopírovat na disk. Následující příkazy je nutné spustit v tomto adresáři, kde je dostupný řídicí soubor `package.json`. Před sestavením je nutné spustit příkaz `npm install`. K dispozici je vývojářský režim spustitelný příkazem `npm run dev`. Pro sestavení verze určené koncovému uživateli slouží příkaz `npm run build`. V počítači musí být pro toto sestavení nainstalováno prostředí Node.js.