

JIHOČESKÁ UNIVERZITA V ČESKÝCH BUDĚJOVICÍCH
PEDAGOGICKÁ FAKULTA
KATEDRA APLIKOVANÉ FYZIKY A TECHNIKY

VYTVOŘENÍ UČEBNÍCH PODPOR K PŘEDMĚTU
„POČÍTAČOVÉ MODELOVÁNÍ A VÝPOČTY
V TECHNICE“

DIPLOMOVÁ PRÁCE

Vedoucí práce: RNDr. Petr Bartoš, Ph. D.

Autor: Bc. Jana Fiktusová

Anotace:

Tato diplomová práce nabízí čtenáři základní přehled nejpoužívanějších postupů počítačového modelování a dostupného softwaru, který lze k tomuto účelu použít. Cílem práce není detailní popis jednotlivých postupů, ale spíše vytvoření přehledového textu, který může čtenář využít jako materiál k prvotní orientaci do dané problematiky. Práce je koncipována jako návrh témat přednášek předmětu „Počítačové modelování a výpočty v technice“.

Práce je rozdělena do tří tematických bloků. V prvním bloku je čtenář seznámen s programovacím jazykem MATLAB, s jeho možností použití ve studiu i praxi. Druhý blok seznamuje s CAD systémy a poslední blok prezentuje bližší použití jednotlivých softwarových aplikací při řešení vybraných modelových situací.

Annotation:

This diploma thesis offers to the reader the basic overview of the most used methods of computer simulation and accessible software which can be used for this purpose. The aim of the thesis is not a detailed description of individual procedures, but rather a creation of overview text which can be used by the reader as a material for the primary orientation to the given problems. The thesis is outlined as a proposal of the themes of lectures of the subject “Computer simulation and calculations in technology”.

The thesis is divided into three thematic blocks. In the first block, the reader is acquainted with the programming language MATLAB, with its possibility of application in study as well as in practice. The second block acquaints with the CAD systems and the last block presents a nearer usage of individual software applications during the solution of selected model situations.

Prohlášení:

Prohlašuji, že svoji diplomovou práci jsem vypracovala samostatně pouze s použitím pramenů a literatury uvedených v seznamu citované literatury.

Prohlašuji, že v souladu s § 47b zákona č. 111/1998 Sb. v platném znění souhlasím se zveřejněním své diplomové práce, a to v nezkrácené podobě, elektronickou cestou ve veřejně přístupné části databáze STAG provozované Jihočeskou univerzitou v Českých Budějovicích na jejích internetových stránkách, a to se zachováním mého autorského práva k odevzdanému textu této kvalifikační práce. Souhlasím dále s tím, aby toutéž elektronickou cestou byly v souladu s uvedeným ustanovením zákona č. 111/1998 Sb. zveřejněny posudky školitele a oponentů práce i záznam o průběhu a výsledku obhajoby kvalifikační práce. Rovněž souhlasím s porovnáním textu mé kvalifikační práce s databází kvalifikačních prací Theses.cz provozovanou Národním registrem vysokoškolských kvalifikačních prací a systémem na odhalování plagiátů.

V Českých Budějovicích dne 1. srpna 2012

.....

Bc. Jana Fiktusová

PODĚKOVÁNÍ:

Touto formou bych velice ráda poděkovala svému vedoucímu diplomové práce panu RNDr. Petru Bartošovi, Ph. D., za cenné návrhy, rady, připomínky a čas věnovaný konzultacím, které mi pomohly při vypracování mé diplomové práce.

OBSAH

ÚVOD	8
Návrh silabu předmětu Počítačové modelování a výpočty v technice	9
1 POČÍTAČOVÉ MODELOVÁNÍ	
1.1 Matematický model	10
1.1.1 Základní prvky modelu	10
1.1.2 Typy modelů	11
1.1.3 Postupy matematického modelování	12
1.1.4 Metodologie matematického modelování	12
1.2 Matematické modelování a ICT	12
1.3 Modelování 2D a 3D	13
1.4 Využití matematického modelování v praxi	14
1.5 Vybraný software pro matematické modelování na počítači	14
2 ZÁKLADY PRÁCE V PROGRAMU MATLAB	
2.1 Základní prvky systému MATLAB	15
2.2 Základní vlastnosti pracovního prostředí MATLAB	16
2.3 Základní prvky programovacího jazyka v prostředí MATLABu	17
2.4 Základní operace	18
2.4.1 Přiřazovací příkazy, konstanty a maticové proměnné	19
2.4.2 Vytváření matic a vektorů	19
2.4.3 Zápis dat do souboru a čtení ze souboru	19
2.5 Maticové operace	20
2.5.1 Základní maticové operace	20
2.5.2 Operace nad prvky matic	20
2.5.3 Práce s polynomy a interpolace	20
2.5.4 Práce se soubory a znakovými řetězci	21
2.5.5 Výstupní formát dat	21
2.6 Maticové funkce	21
2.6.1 Funkce na sestavení speciálních matic a vektorů	22
2.6.2 Řídící příkazy	22
2.7 Vizualizace – typy grafů	23
2.8 <i>M</i> – soubory	25

2.9	Toolboxy	26
2.10	Příklady k řešení	27
3	METODA MONTE CARLO	
3.1	Schéma řešení problému pomocí metody Monte Carlo	28
3.2	Stručné shrnutí pojmů statistiky	29
3.3	Použití metody Monte Carlo v procesu simulace	33
3.4	Příklad k řešení	34
4	GENEROVÁNÍ NÁHODNÝCH ČÍSEL	
4.1	Generování náhodných čísel	35
4.2.	Transportní problém	36
4.3	Rozehrání náhodné volné dráhy	38
4.4	Příklad k řešení	39
5	EULERŮV A VERLETŮV ALGORITMUS A METODA LEAP-FROG	
5.1	Použití metody molekulární dynamiky	40
5.2	Pracovní oblast, výpočet silového působení	40
5.3	Řešení pohybových rovnic	43
5.3.1	Eulerova metoda	43
5.3.2	Verletova metoda	43
5.3.3	Metoda Leap-Frog	44
5.4	Metoda P-I-C	44
5.5	Model plynu a difúze	45
5.6	Příklady k řešení	47
6	CAD SYSTÉMY	
6.1	Požadavky na CAD systémy	49
6.2	Rozdělení CAD systémů	50
7	SOLIDWORKS – 3D CAD SYSTÉM	
7.1	SolidWorks	53
7.2	Aplikace a balíčky pro SolidWorks	55
7.3	FloXpress	55
8	METODA KONEČNÝCH PRVKŮ, VEDENÍ TEPLA A PROUDĚNÍ TEKUTIN	
8.1	Metoda konečných prvků	57
8.1.1	Diferenciální rovnice – definice a základní pojmy	57

8.1.2 Metoda řešení parciálních diferenciálních rovnic – MKP	58
8.1.3 Nástin řešení příkladu pomocí MKP	61
8.1.4 Výhody a nevýhody výpočtových metod CAD systémů	62
8.2 Vedení tepla	62
8.3 Proudění tekutin	65
8.4 Počítačová dynamika tekutin	68
8.5 Příklad k řešení	69
9. SOFTWARE COMSOL MULTIPHYSICS	
9.1 COMSOL Multiphysics	70
9.2 Příklad k řešení	73
10 SOFTWARE ANSYS	
10.1 Software ANSYS Fluent	74
10.2 Software ANSYS GAMBIT	77
10.3 Software ANSYS TGrid	77
11 MODELOVÁNÍ DYNAMICKÝCH SÍTÍ	
11.1 Příklad tvorby sítě v programu ANSYS Fluent	79
12 DEFORMACE PŮSOBENÍM MECHANICKÉ SÍLY A V DŮSLEDKU TEPELNÉ ROZTAŽNOSTI	
12.1 Deformace působením mechanické síly a důsledkem tepelné roztažnosti	83
12.2 Příklad k řešení	86
13 FREEWAROVÉ APLIKACE – ALTERNATIVY KE KOMERČNÍM PRODUKTŮM	
13.1 Vybrané 2D CAD programy	87
13.2 Vybrané 3D CAD programy	89
13.3 Vizualizace a animace	90
13.4 Vybrané E programy	91
13.5 Další freewarové aplikace	92
13.6 Výhody a nevýhody značného množství freewarových CAx aplikací	92
14 Závěrečná přednáška – shrnutí, opakování a prezentace studentských prací	93
14.1 Nástin řešení seminární práce	93
ZÁVĚR	95
POUŽITÉ ZDROJE A LITERATURA	96

Úvod

Cílem této diplomové práce je vytvoření učebních podpor k předmětu Počítačové modelování a výpočty v technice. Již samotný název diplomové práce říká, že obsahem je zevrubné seznámení se softwarovými možnostmi počítačového modelování fyzikálních úloh a výpočtů v technice. Práce je zaměřena na metody, prostředky a aplikace počítačového modelování, zpracování a vizualizací dat, které jsou použitelné ve fyzice a v technické praxi při simulaci konkrétních složitějších úloh. Tyto metody se používají při řešení technických problémů, které jsou obtížně řešitelné nebo jsou neřešitelné jinými matematickými prostředky. Práce je členěna do tří tématických bloků, které jsou propojeny některými podkapitolami i mezi sebou. Zároveň se zabývá také rozčleněním do jednotlivých vyučovacích celků, kterých je čtrnáct (podle počtu týdnů v jednom semestru), pro vyučování jedenkrát v týdnu dvě hodiny přednáška a jedna hodina cvičení.

Hlavním tématickým celkem je přehled softwarů použitelných pro počítačové modelování, jejich zevrubný popis a možnosti použití při výpočtech v technické praxi.

Blok 1 obsahuje základní seznámení s počítačovým programem *MATLAB*, který je vhodný pro technické výpočty a použitelný ve všech odvětvích vědy a techniky. Pomocí jednoduché struktury a uživatelsky příjemného prostředí tohoto programovacího jazyka lze provádět i složité fyzikální a technické výpočty. Další částí tohoto bloku jsou základy metody *Monte Carlo*, která patří bezesporu mezi užitečné matematické postupy. Opírá se o pojmy z *pravděpodobnosti* a *statistiky* a umožňuje řešit problémy obtížně řešitelné tradičními metodami. Do bloku je zařazena i metoda molekulární dynamiky, která se zabývá studiem dynamiky částic, kde chování studovaných částic je popisováno pomocí pohybových rovnic. Celý tento blok je zaměřen hlavně na prezentaci využití programu *MATLAB*, kde příklady použití základních příkazů *MATLABU* jsou vysvětlovány na vybraných často používaných obratech počítačové fyziky.

Blok 2 je věnován přehledu CAD systémů vhodných pro použití v materiálovém inženýrství, automobilovém průmyslu, leteckém průmyslu, aj. Čtenář je stručně seznámen s programem *SolidWorks*, který je určen hlavně pro simulaci fyzikálních, technických a vědeckých modelů.

Blok 3 obsahuje seznámení s použitím dalších programů pro řešení problémů vedení tepla, proudění tekutin, deformace působením mechanické síly nebo v důsledku teplotní roztažnosti metodou sítí či konečných prvků. Jedná se o software *COMSOL Multiphysics* a *ANSYS Fluent*. Na závěr jsou zmíněny vybrané freewareové aplikace jako alternativy ke komerčním produktům.

Závěrečná hodina může být věnována prezentaci studentských projektů. Právě při zpracovávání těchto projektů studenti získávají nadhled na problematiku počítačového modelování. Techniky počítačového modelování, se kterými se student seznámil během kurzu, je možné použít v řadě jiných oborů a lze tudíž říci, že počítačové modelování je v podstatě samostatný obor.

Na konci každého bloku jsou nastíněny možné problémy k řešení ve cvičeních.

Na internetu i v knihovnách je dostupné velké množství materiálů zabývajících se problematikou počítačového modelování a nemělo by v práci význam je pouze přeformulovat. Proto jsem vybrané pasáže převzala, příslušným způsobem je cituji a doplňuji je jednotlivými komentáři. Text je dále doplněn o odkazy na literaturu, ve které je možné nalézt podrobnější informace.

Návrh sylabu předmětu Počítačové modelování a výpočty v technice.

Časová dotace: 2 + 1

Témata přednášek:

BLOK 1

1. Úvodní hodina – seznámení s obsahem studia v semestru, účel simulace, software pro technické výpočty – stručná charakteristika, ukázky.
2. MATLAB – základy programování – základní funkce a příkazy, ovládání příkazového okna, využívání zabudovaných funkcí, grafické prostředky.
3. Základy metody Monte Carlo.
4. Generátory polohy a rychlosti částic, rozehrání náhodné volné dráhy částic.
5. Eulerův a Verletův algoritmus a metoda Leap-Frog, metoda Particle in Cell (P-I-C), – model plynu a difúze.

BLOK 2

6. CAD systémy – materiálové inženýrství, automobilový průmysl, letectví, vesmír, ...
7. SolidWorks – seznámení se softwarem, základní obsluha programu

BLOK 3

8. Metoda konečných prvků,
– vedení tepla a proudění tekutin.
9. COMSOL Multiphysics – seznámení se softwarem.
10. ANSYS Fluent – metoda sítí pro řešení diferenciálních rovnic.
11. Modelování dynamických sítí.
12. Deformace působením mechanické síly a v důsledku teplotní roztažnosti.
13. Freewareové aplikace – alternativy ke komerčním produktům.
14. Závěrečná hodina – prezentace studentských projektů.

Požadavky na studenta:

Aktivní zvládnutí učiva v odpřednášeném rozsahu a daném tematickém plánu.

Doporučená literatura:

F. Dušek. *MATLAB a SIMULINK - úvod do používání*. Univerzita Pardubice, 2002. ISBN 80-7194-475-0.

J. Novák, I. Pultarová, P. Novák. *ZÁKLADY INFORMATIKY – počítačové modelování v Matlabu*. ČVUT Praha, 2005. ISBN 80-01-03308-2.

J. Nezbeda, J. Kolafa, M. Kotrla. *ÚVOD DO POČÍTAČOVÝCH SIMULACÍ*. MFF UK Praha, 1998.

R. Hrach. *POČÍTAČOVÁ FYZIKA I*. PF UJEP Ústí nad Labem, 2003

H. Vláčilová a kol. *SOLIDWORKS*. Computer Press, Brno 2006.

I. Nezbeda, J. Kolafa, M. Kotrla. *Úvod do počítačových simulací metody Monte Carlo a molekulární dynamiky*. PRAHA : Karolinum, 2003.

M. Guštar. *Generování náhodně proměnných veličin v metodě Monte Carlo*. 15.3.2000. ISBN 80-02-01344-1.

BLOK 1

KAPITOLA 1: POČÍTAČOVÉ MODELOVÁNÍ

Matematické modelování nesporně patří mezi vědecké metody používané nejen ve vědeckém světě, ale i v jiných oborech lidské činnosti, protože představuje nenahraditelný nástroj k poznávání a pochopení pozorovaných jevů reálného světa, dokáže napodobit chování zkoumaných systémů, simulovat je na technickém modelu a následně ovlivnit jeho chování požadovaným způsobem. Modelování bývá chápáno jako multifunkční, protože v sobě zahrnuje nejen matematiku a fyziku, ale také teorii systémů, teorii pravděpodobnosti, informatiku, výpočetní techniku, další přírodní vědy, výzkum a podobně.^[1]

Ve vědecké i technické praxi se často řeší právě nejrůznější úlohy, které lze popsat vhodným matematickým modelem. Tyto úlohy ve většině případů nelze řešit pomocí uzavřeného matematického tvaru, protože takové řešení je příliš obtížné či nevhodné pro praktickou aplikaci, a nebo vůbec takové řešení neexistuje. Proto se v současné době řešení naprosté většiny reálných problémů provádí pomocí výpočetní techniky a vhodných počítačových metod. Výhodou počítačových metod je velmi rychlé zpracování velkého objemu dat a tím možnost provádět velké množství simulací daného problému s různými hodnotami vstupních parametrů v krátkém čase. Výsledky získané pomocí výpočetní techniky jsou do značné míry závislé na zvolené numerické metodě a postupu výpočtu.^[1]

1.1 Matematický model

Matematický model je abstraktní popis reálného systému využívající matematický zápis a je vždy pouze přiblížením skutečnosti, zjednodušením reality. Procesy probíhající v reálném světě jsou velmi komplikované a při jejich popisu je nutné mnohé věci zjednodušovat a vybírat jen nejdůležitější části modelovaného problému. V simulaci je možné předdefinovat hodnoty jedné i více závislých proměnných, ne však s absolutní přesností, ale pouze jen s jistou pravděpodobností. Aby byl vytvořený model akceptovatelný, nesmí příliš zjednodušovat a zkracovat vlastnosti simulovaného systému, musí být co nejdělejší.

Proč používat modelování? Několik bodů, proč je použití modelování výhodné: Pro daný soubor faktů A není známa žádná teorie, ale tento soubor faktů má některé společné vlastnosti s jiným souborem faktů B, pro který již určitá teorie existuje, tzn. že strukturovaný soubor faktů B může teoreticky sloužit jako model daného souboru faktů A. Pro daný objekt zkoumání platí rozvinutá teorie, v níž je matematicky obtížné dojít k některým výsledkům; pojmy této teorie lze někdy interpretovat pomocí vhodného modelu daného objektu tak, že na tomto modelu můžeme dojít k požadovaným výsledkům snáze než v rámci původní teorie. Model umožňuje interpretovat některé teorie a tvoří tak spojovací článek mezi teorií a skutečností, danou teorií lze pomocí modelu doplňovat. Model umožňuje studium některých experimentálně nedostupných objektů, umožňuje experimentovat tam, kde experiment na reálném objektu zkoumání by byl nevhodný nebo by hrozilo jeho fyzické poškození. Model slouží jako standard ideálního nebo idealizovaného chování, na jehož pozadí lze posoudit realitu.^[2]

1.1.1 Základní prvky modelu^[2]

Matematický model popisuje systém pomocí proměnných a vztahů mezi nimi. Model se skládá ze tří základních skupin objektů, a to z proměnných a konstant, matematické struktury a řešení.

a) Proměnné a konstanty

řídící (nezávislé) konstanty – představují nejdůležitější procesy modelovaného systému,
parametry (konstanty) – nezávislé, hodnoty jsou určovány mimo modelovaný systém,
vstupní proměnné – ovlivňují model, hodnoty určeny mimo modelovaný systém,
výstupní proměnné – hodnoty jsou generovány stavem systému,
stavové proměnné – závislé na ostatních proměnných,
náhodné proměnné – představují neurčitost v modelu, hodnoty jsou určeny pravděpodobnostní funkcí nebo hustotou pravděpodobnosti

b) Matematické struktury

V matematických modelech se matematické struktury nazývají omezující podmínky.

Analytické struktury – soustavy rovnic, nerovnic, funkce,

geometrické struktury – model je popsán grafickými prostředky,

topologické struktury – vychází z matematické teorie grafů.

c) Řešení

Řešení matematického modelu můžeme dělit podle cíle, modelování na přípustné, maximální, minimální, optimální nebo alternativní.

1.1.2 Typy modelů^[2]

Rozdělení podle zahrnutí náhodných veličin do modelu:

- statické – neberou v úvahu časový vývoj systému, zobrazení se týká zvoleného časového intervalu,
- dynamické – popisují systém v průběhu času,
- deterministické – všechny proměnné, konstanty a funkce jsou nenáhodné veličiny nebo funkce,
- stochastické – obsahují náhodné veličiny a funkce.

Rozdělení modelů podle zobrazovaných vlastností originálu:

- substanciální – shodují se s originálem ve všech nebo v některých základních vlastnostech,
- strukturní – zobrazují především vnitřní uspořádání prvků, z nichž se skládá zkoumaný originál,
- funkční – zobrazující různé funkce respektive chování originálu,
- smíšené – kombinace předchozích případů.

Rozdělení modelů podle zobrazující funkce:

- materiální – fungují na základě objektivních zákonů nezávisle na činnosti lidí,
- ideální – fungují podle zákonů logiky, existují jen v činnosti lidí.

Rozdělení modelů podle funkčního použití technických prostředků:

- počítačová simulace – model je vytvořený čistě počítačovým programem, který nasimuluje abstraktní model určitého systému, zjistí, jak se bude systém chovat pro zadaná vstupní data a provede optimalizaci. Existuje mnoho různých druhů počítačových simulací, které ale mají jeden společný rys – pokouší se generovat vzorek reprezentativních scénářů pro model, ve kterém by byl kompletní výpočet všech možných stavů modelu nepřipustný nebo nemožný,
- simulace v technice – zde je simulace chápána jako napodobování určitých podmínek, fyzikálních nebo chemických dějů, případně napodobování chování výrobků nebo technických zařízení. Pro zajištění simulace je potřebné i určité technické vybavení, jako například klimatická komora, aerodynamický tunel, letecký simulátor, apod. Jistým druhem simulace je i vlastní technický výpočet. K simulaci se používají jak fyzické, tak i matematické modely. Fyzické modely většinou neumožňují rychlou a levnou analýzu problému, ale použijí se tam, kde by byl matematický model příliš složitý na řešení nebo by vůbec neexistoval,
- vědecké modelování – slouží nejen pro řešení problémů z praxe, ale je určeno hlavně k provádění nejrůznějších výzkumů a experimentů tam, kde by při provádění experimentu na reálném systému hrozilo jeho poškození. Vytvořený vědecký simulační model se následně stává počítačovým programem, kde je pro jeho sestavení nutná specifická algoritmizace, a kde nejdůležitější fází je způsob zachycení času. Vědecká simulace se v zásadě tedy dělí na simulaci spojitou, při které děj probíhá bez časových skoků a výsledky modelu se mění průběžně, a diskrétní, kde simulace vykazuje změny hodnot jen v případě, že nastane nějaká nová událost díky níž se změní hodnota některé proměnné.

Existuje ještě mnoho kategorií matematických modelů, například podle spojitosti máme modely spojitě a diskrétní, zaměříme-li se na linearitu, hovoříme o lineárních a nelineárních modelech, atd.

1.1.3 Postupy matematického modelování^[2]

Podle pozorování reálného objektu vytvoříme jeho abstraktní model. Navržený model vyřešíme a vzniknou nám poznatky o modelu, které se snažíme interpretovat s ohledem na reálný objekt. Tím získáme nové informace o objektu, které můžeme implementovat do dalšího výzkumu. Proces zkoumání systému začíná sběrem dat a pokračuje jejich vstupem do matematického modelu. Pak probíhá analýza výsledků řešení modelu. Výsledky by měly být v souladu s naměřenými hodnotami. Pokud tomu tak není, musí dojít k úpravě modelu a nové analýze.

Průběh matematického modelování lze rozdělit na čtyři etapy:

- 1) sestavení modelu – zápis vlastností a zkoumaných zákonitostí objektu pomocí matematických termínů,
- 2) matematická analýza modelu a získání teoretických důsledků,
- 3) zkoumání, zda je hypotetický model v souladu s praktickými poznatky a přijetí modelu,
- 4) následná analýza modelu v souvislosti s novými poznatky o zkoumaném objektu a případné vylepšení modelu.

Dalším důležitým problémem je identifikace koeficientu – určení hodnot parametrů vystupujících v modelu.

1.1.4 Metodologie matematického modelování^[2]

Vyvíjí se jako samostatná odborná specializace. Platí několik obecných zásad jak modelovat, které lze popsat následujícími kroky:

- a) *Identifikace systému*: zda se jedná o standardní nebo nový systém, výběr typu modelu, tvorba předpokladů.
- b) *Konstrukce modelu*: definice prvků, vazeb, struktury a chování systému, výběr významných částí podle účelu modelu, analýza citlivosti a neurčitosti.
- c) *Řešení modelu*: výpočet – analytické, numerické řešení, výběr optimálních řešení, verifikace řešení – kontrola výsledků.
- d) *Modifikace modelu*: vylepšení modelu, pokud není v souladu s modelovanou realitou.

1.2 Matematické modelování a ICT^[2]

V dnešní době se matematické modelování neobejde bez využití ICT. Postup matematického modelování je uveden na obrázku 1.1. Jde o permanentní proces sestávající z několika kroků s četnými zpětnými vazbami, který se několikrát opakuje.

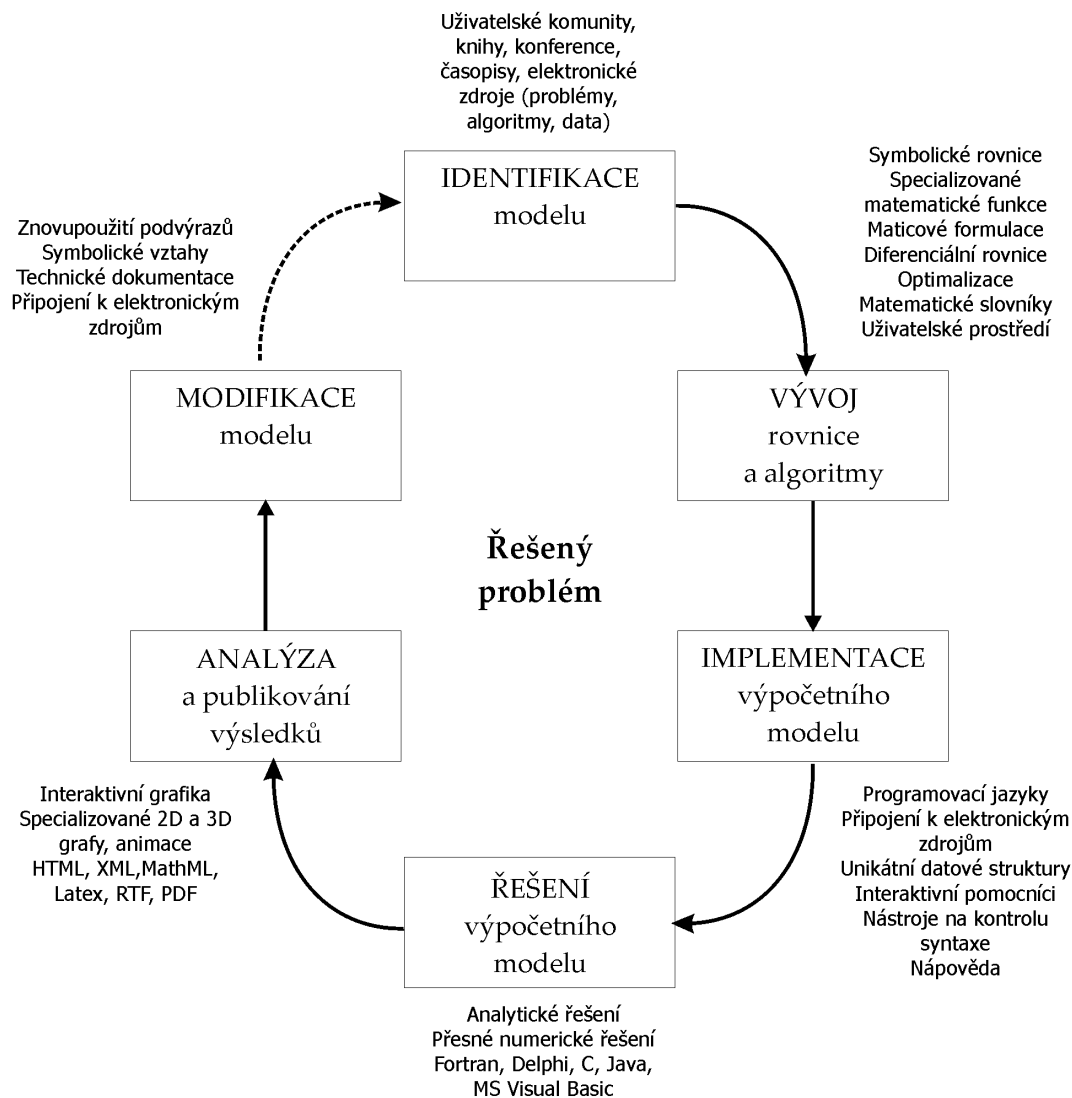
Prvním krokem při vytváření matematického modelu je identifikace jednotlivých složek s využitím odborné literatury a spoluprací s odbornou vědeckou komunitou. Velmi důležitá je správná matematická formulace zkoumaného problému, která vychází z analýzy systému a stanovených cílů řešení. Složitou realitu je třeba zjednodušit pro účely modelu a definovat prvky, jejich vazby, procesy, vstupy, výstupy a funkce.

Vývoj modelu začíná tvorbou předpokladů o fungování systému. Pokud jsou dostatečně precizní, mohou okamžitě vést k matematickým rovnicím popisujícím modelovaný systém. Pro konstrukci modelu je rozhodující účel, který sledujeme. Ten pak rozhoduje o tom, co budeme pokládat za významné a co zahrneme do modelu. Pokračujeme výběrem matematických rovnic a analýzou citlivosti a neurčitosti parametrů modelu.

Implementace modelu s využitím ICT zahrnuje naprogramování v příslušném programovacím jazyce, odladění, verifikaci a analýzu výpočetní složitosti programu.

Řešení modelu je možné získat dvěma způsoby: analyticky pomocí analytických matematických metod nebo numericky v případě, že problém neumíme analyticky řešit, či je to velmi obtížné. Pak využijeme k výpočtu různé simulační metody (např. Monte Carlo) a musíme také uvažovat o numerické stabilitě, konvergenci a chybě, která vznikne při výpočtu.

Následuje analýza řešení modelu. Kontrolujeme, zda výsledky souhlasí s chováním objektu, a využíváme k tomu různé vizualizační prostředky. Cílem testování modelu je prověření jeho správné struktury, vypovídacích schopností a odstranění formálních chyb. Výsledky modelu porovnááme s realitou, tj. s naměřenými údaji.



Obr. 1.1: Cyklus tvorby matematického modelu^[3]

1.3 Modelování 2D a 3D^[4]

a) **2D grafika** neboli dvojrozměrná grafika (má délku a šířku, nemá objem) se zabývá grafickou informací popsanou v rovině. Používají se dva nejznámější způsoby:

- *rastrová grafika*: je založena na matici informací, které popisují barvu každého bodu (pixelu) pomocí RGB systému. Nevýhodou tohoto zobrazování je ztráta detailů, protože čím víc detailů je třeba zachytit, tím je třeba větší počet bodů matice (nárůst velikosti výsledných dat) a s tím spojená omezená možnost zvětšování obrazu.

Rastrové editory: *Adobe Photoshop, Corel Photopaint, The GIMP, Paintshop Pro, MSpaint*, aj.

- *vektorová grafika*: neukládá informace o jednotlivých bodech, ale o přímkách, křivkách a polygonech, jejich tloušťce a barvě. Výhodou je možnost zmenšování a zvětšování obrazu bez ztráty kvality, protože paměťová náročnost je mnohem menší než u rastrového zobrazování. Využívá se zejména pro tvorbu ilustrací, animací, grafů a diagramů.

Vektorové editory: *Adobe Illustrator, CorelDraw, Inkspace, Sodipodi, Zoner Callisto*, aj.

b) **3D grafika** neboli trojrozměrná grafika vychází z vektorové 2D grafiky a geometrická data se neukládají pouze v rovině, ale v prostorové soustavě souřadnic. Základními geometrickými útvary jsou tzv. polygony, což jsou plošky, ze kterých jsou tvořeny všechny objekty. Používá se například při tvorbě animací, počítačových simulací nebo pro trojrozměrné zobrazení orgánů.

Základními pojmy 3D grafiky jsou:

- *modelování*: proces vytváření a tvarování 3D modelu podle dat získaných měřicím přístrojem z reálného procesu nebo na základě počítačové simulace,
- *texturování*: vytváření a mapování textur, které určují například průhlednost nebo lesk daného bodu na povrchu,
- *animace*: kromě samotného pohybu objektu definuje například barvu, zdroj světla, úhel pohledu a další prvky, které se mohou měnit v čase,
- *renderování*: vykreslení 3D obrazu ve dvojrozměrném zobrazení na základě modelu scény a dalších informací (textury, osvětlení, stínování či polohy pozorovatele).

1.4 Využití matematického modelování v praxi^[4]

- *Přírodní vědy*: veškerý vědecký výzkum se opírá hlavně o simulace zkoumaných přírodních jevů v oblastech chemie, biologie, genetického inženýrství, fyziky, farmaceutického průmyslu, jaderného výzkumu, aj.
- *Lékařství*: pomáhá simulovat operativní metody a vliv těchto použitých metod na průběh uzdravování pacienta, simulaci používají některé lékařské přístroje pro zobrazování po snímkování jako je CT nebo magnetická rezonance.
- *Letectví*: letecké simulátory pro výcvik pilotů hrají nezastupitelnou úlohu v bezpečnosti letecké dopravy nebo při výcviku vojenských bojových pilotů. Bez simulace se neobejdou ani leteční konstruktéři a vývojáři.
- *Doprava*: pomocí simulací pracovníci v dopravě navrhují nejefektivnější trasy pro přepravu jak v pozemní osobní či nákladní automobilové, vlakové a městské hromadné přepravě, tak letecké dopravě, modelování řízeného pohybu vozidel na křižovatkách, vyhledávání optimálního objezdu při zácpách, řešení optimálního rozložení parkovacích ploch, atd. Simulátory se ve značné míře využívají i při výcviku řidičů v autoškole.
- *Ekonomika a logistika*: simulace zahrnuje optimalizaci, plánování a řízení výroby, prověřování různých variant řešení s cílem eliminovat rizika a zefektivnit výrobní procesy.
- *Zemědělství*: v tomto odvětví se simulace využívají ke zjištění nejvyššího poměru zisku k nákladům vynaloženým na pěstování plodin nebo chovu dobytka.
- *Vojsko a policie*: zde se využívají simulátory od bojové strategie, šance na přežití, týmové spolupráce, přes deaktivaci bomb až po řízení nejmodernějších strojů.
- *Meteorologie*: využití simulací od dlouhodobé i krátkodobé předpovědi počasí až po možnost výskytu živelných katastrof jako je zemětřesení a následné tsunami.
- *Školství*: simulace pro zefektivnění výuky s ohledem na využívání laboratoří a poslucháren, optimální rozložení škol v regionu.
- *Ekologie*: simulace záplavových oblastí či znečišťování životního prostředí.
- *Urbanistika*: modelování městských sídlišť, jejich struktury a komunikací.
- *Sociologie*: modelování migrace obyvatel, vliv koncentrace lidí v městských sídlištích.
- *Bankovníctví*: simulace předpokládaného pohybu nebo vývoje na finančních a akciových trzích, problematika bankovního managementu.

1.5 Vybraný software pro matematické modelování na počítači^[1]

- *Matlab*: Matematický systém pro počítačové modelování ve vědě, technice a ekonomii s rozsáhlou sítí knihoven funkcí.
- *Simulink*: Matematický systém pro počítačovou simulaci reálných mechanických nebo elektronických systémů, popsaných pomocí matematických modelů.
- *Mathematica*: Software pro různé matematické výpočty a grafické znázornění výsledků.
- *Maple*: Software pro symbolické matematické výpočty.
- *MultiPhysics*: Software pro řešení složitých fyzikálně-inženýrských úloh z celé oblasti fyziky pomocí metody konečných prvků.

2 ZÁKLADY PRÁCE V PROGRAMU MATLAB^[5]

Produkt MATLAB byl vyvinutý a je neustále udržovaný společností The MathWorks, Inc. a je Protected by U. S. patents.

MATLAB je programové prostředí a skriptovací programovací jazyk pro vědeckotechnické numerické výpočty, počítačové modelování nejrůznějších fyzikálních a jiných dějů a jevů, návrhy a vývoj počítačových algoritmů, počítačové simulace, analýzu a prezentaci dat, měření a zpracování signálů, návrhy řídicích a komunikačních systémů, využitelných pro řešení různě složitých problémů, které jsou vždy podpořeny odpovídajícím matematickým modelem. V neposlední řadě lze také pomocí tohoto systému vytvářet samostatné aplikace s příjemným uživatelským grafickým prostředím.

Název MATLAB vznikl zkrácením slov MATrix LABoratory (volně přeloženo „laboratoř s maticemi“), což odpovídá skutečnosti, že klíčovou datovou strukturou při výpočtech v MATLABU jsou matice, resp. n -dimenzionální datová pole.^[6]

Program MATLAB lze tedy s úspěchem uplatnit v typických oblastech jako jsou:

- ☆ Matematické výpočty
- ☆ Vývoj algoritmů
- ☆ Modelování a simulace
- ☆ Analýza dat a vizualizace
- ☆ Vědecká a inženýrská grafika
- ☆ Vývoj aplikací včetně uživatelského rozhraní

Program MATLAB je multiplatformní a to znamená, že lze používat jak v operačních systémech Windows, tak i v Linuxu či Unixu.

2.1 Základní prvky systému MATLAB^[7]

a) Vývojové prostředí MATLABU

Tato skupina nástrojů umožňuje využívat funkce MATLABU, pracovat se soubory, proměnnými a funkcemi. Jde o samostatné uživatelské prostředí, které slouží k vývoji, ladění a spouštění vytvořených počítačových programů. Zahrnuje základní pracovní plochu, příkazové okno, historii příkazů, editor souborů, ladící procedury, nápovědu, atd.

b) Knihovna matematických funkcí

Jednou z nejdůležitějších součástí programu MATLAB jsou knihovny funkcí (ardesáře s m a mex soubory), které se nazývají *Toolboxy*. Ty obsahují vždy zpracovaný určitý obor (numerická matematika, analytická matematika, statistika) jako sjednocený soubor informací včetně dokumentace a příkladů. Je to rozsáhlý soubor nejrůznějších matematických algoritmů od nejzákladnějších matematických funkcí až po složité matematické algoritmy pro numerické řešení různých matematických problémů, jako jsou například numerické metody lineární algebry, řešení lineárních a nelineárních rovnic, řešení diferenciálních rovnic, interpolace, integrace, optimalizace funkcí, Fourierova transformace, atd. Systém umožňuje také vytváření vlastních dialogových oken a oken s kombinacemi grafů, tlačítek, listů a dalších objektů, kterým mohou být přiřazeny volané funkce v m -file.

c) Programovací jazyk

Jedná se o vyšší programovací jazyk, jehož základní strukturou je datové pole, neboli matice či vektor, se standardními příkazy pro běh programu, standardními datovými typy, funkcemi pro vstup a výstup a znaky objektově orientovaného programování. Systém obsahuje vlastní interpreter jazyka MATLAB, ve kterém je možné připravit jak dávkové soubory, tak i definovat nové funkce. Tyto funkce mohou být interpretovány buď přímo z textového souboru (m -file) nebo z předdefinované podoby souboru (p -file), systém dále umožňuje přidávat moduly (soubory

mex-file) zkomprimované do strojového kódu procesoru. Používají se programovací jazyky C, C++, Fortran nebo po použití *Matlab to C Computer* i funkce uložené v *m-file*.

d) Grafické prostředí

Základní funkce umožňují pro 2D a 3D zobrazení dat ve vektorové či maticové formě. Systém má mnoho funkcí pro 2D a 3D vizualizaci dat, zpracování obrazu, animací a prezentací grafiky. Pro grafický výstup je k dispozici velmi silná podpora tvorby uživatelského prostředí a vizualizace dat. Problémem není ani tisk grafů s doplněnou legendou a popiskami os.

e) Aplikační programové rozhraní

Tato knihovna obsahuje širokou skupinu funkcí a příkazů, které umožňují vzájemnou a oboustrannou spolupráci MATLABu s různými funkcemi a programy vytvořenými v externích programovacích jazycích. Zahrnuje prostředky pro volání podprogramů v MATLABu, volání výpočetního jádra MATLABu, předávání dat, apod.

2.2 Základní vlastnosti pracovního prostředí MATLAB^[7]

a) Spuštění programu

Program se spouští příkazem *matlab* z operačního systému, nebo poklepnem na ikonu v grafickém prostředí. Po spuštění se na pracovní ploše objeví prompt příkazové řádky `>>`. Příkazy napsané za tímto promptem jsou vykonány bezprostředně po stisknutí klávesy Enter. Příkaz, resp. několik příkazů, lze vykonat spuštěním dávky – tzv. *m-file*. *M-file* je sekvence příkazů, která je uložena v paměti jako programový soubor. Příponou tohoto souboru je povinně **.m*.

Po skončení výpočtu zůstanou proměnné uloženy v pracovní paměti. Znak `%` indikuje komentář v textu.

Úvodní obrazovka se zobrazí v takovém stavu, ve kterém byl MATLAB ukončen naposledy, tj. pamatuje si poslední nastavení.

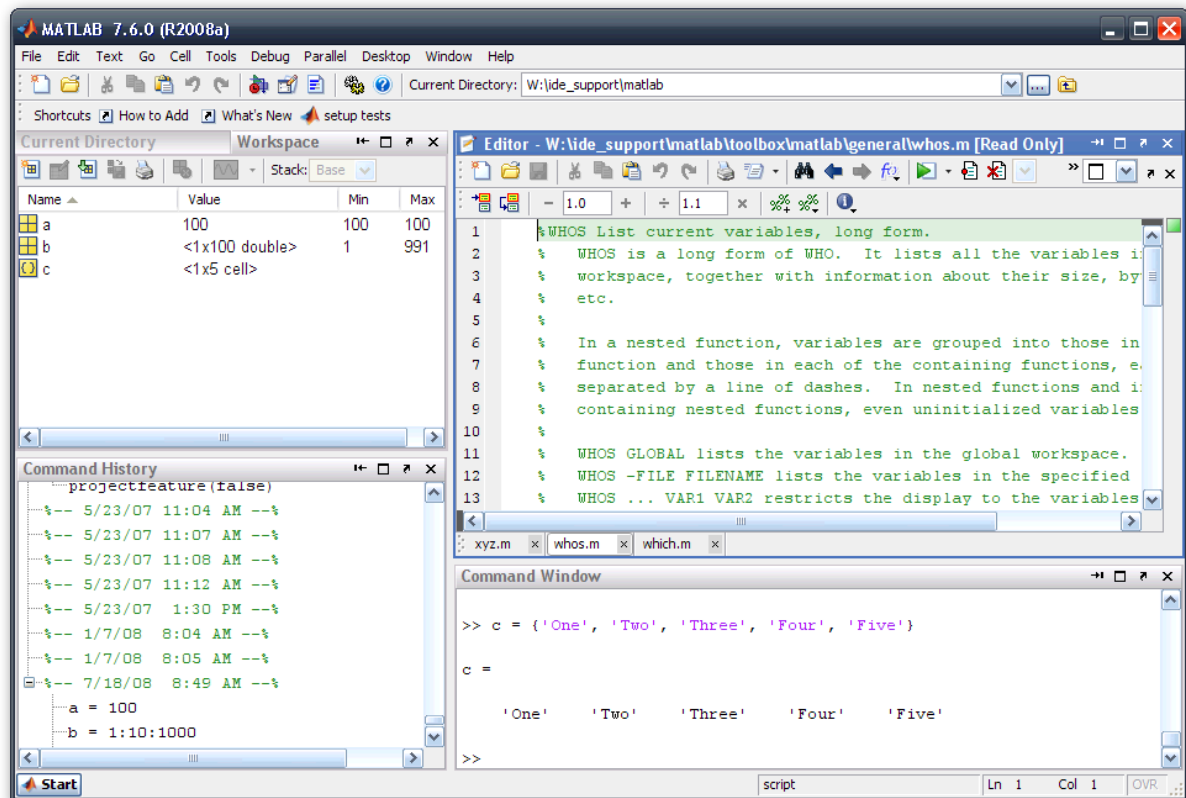
b) Ukončení programu

Program MATLAB ukončíme zavřením okna nebo příkazem *quit* či *exit*. Při opuštění programu ztrácíme proměnné uložené v paměti. Je-li třeba proměnné pro další použití uložit, provedeme uložení příkazem *save*. Tento příkaz uloží všechny proměnné do souboru na disk pod názvem *matlab.mat*, který pro další použití aktivujeme příkazem *load*. Příkazy *save* a *load* můžeme používat se jménem souboru, do kterého se data uloží, nebo i bez něj. Exportovaná a importovaná data jsou v ASCII kódu.

c) Pracovní plocha

Po spuštění programu se zobrazí základní obrazovka (Obr. 2.1), ve které uživatel ovládá program pomocí příkazů menu a jednotlivých dílčích oken. Základní obrazovka je graficky rozdělena na následující prvky:

- 1) Okno *Command Window* – práce v dialogovém režimu, okno je určeno pro zápis příkazů.
- 2) Okno *Workspace* – zobrazení všech dostupných proměnných.
- 3) Okno *Array Editor* – zobrazení proměnných v tabulce nebo jako graf, lze je zde editovat.
- 4) Okno *Current Directory* – zobrazení obsahu aktuálního adresáře. *Kontextové menu souborů* vyvolané pravým tlačítkem myši umožňuje standardní práci se soubory pracovního adresáře: přejmenování, kopírování, přesun, mazání, vytvoření nových adresářů a *M*-souborů.
- 5) Okno *Command History* – zobrazení historie použitých příkazů.



Obr. 2.1 – Základní obrazovka MATLABu^[8]

d) Hlavní menu

- 1) *File* – tvorba a otevření souboru, import, ukládání či tisk dat, ukončení programu, ...
- 2) *Edit* – obsahuje položky pro standardní editovací příkazy.
- 3) *Debug* – slouží pro ladění běhu programu.
- 4) *Desktop/View* – umožňuje nastavit vzhled pracovního prostředí.
- 5) *Window* – umožňuje přepínat mezi jednotlivými okny systému.
- 6) *Help* – umožňuje otevřít rozsáhou nápovědu k systému.

e) Nápověda

MATLAB má k dispozici jak stránky nápovědy v systému Windows tak řádkovou nápovědu v příkazovém okně (*Command Window*). Nápověda se vyvolá příkazem *help*. Tento příkaz vypíše na obrazovku seznam používaných operátorů, klíčových slov a vestavěných funkcí. Kompletní dokumentaci je možné vyvolat příkazem *helpdesk* a to ve formátu buď HTML nebo PDF. Součástí nápovědy jsou též demonstrační programy (*Demos*), které názorně ukazují použití jednotlivých funkcí na příkladech. U mnoha funkcí je též uveden plný zdrojový text, který je možné modifikovat.

2.3 Základní prvky programovacího jazyka v prostředí MATLABu^[7]

1) Základní prvky, které tvoří programovací jazyk

- a) *Symbols a skupiny symbolů* – jsou složeny z číslic, písmen a některých speciálních znaků. Slouží k označení operátorů, proměnných, apod.
- b) *Rezervovaná slova* – tzv. klíčová, mají přesně definovaný význam a nelze je měnit (*break, case, catch, continue, else, elseif, end, for, function, global, if, otherwise, persistent, return, switch, try, while*).
- c) *Datové typy* – určují vlastnosti daného datového prvku, uchovávání v paměti a interpretace v souvislosti s operátory jazyka.
- d) *Proměnné* – jsou jednoznačně určeny svými názvy, na které se program odkazuje při jejich volání. Názvy mají přesně daná pravidla a omezení. Zvláštní skupinu tvoří *speciální proměnné*,

které vracejí určitou specifickou číselnou hodnotu, například:

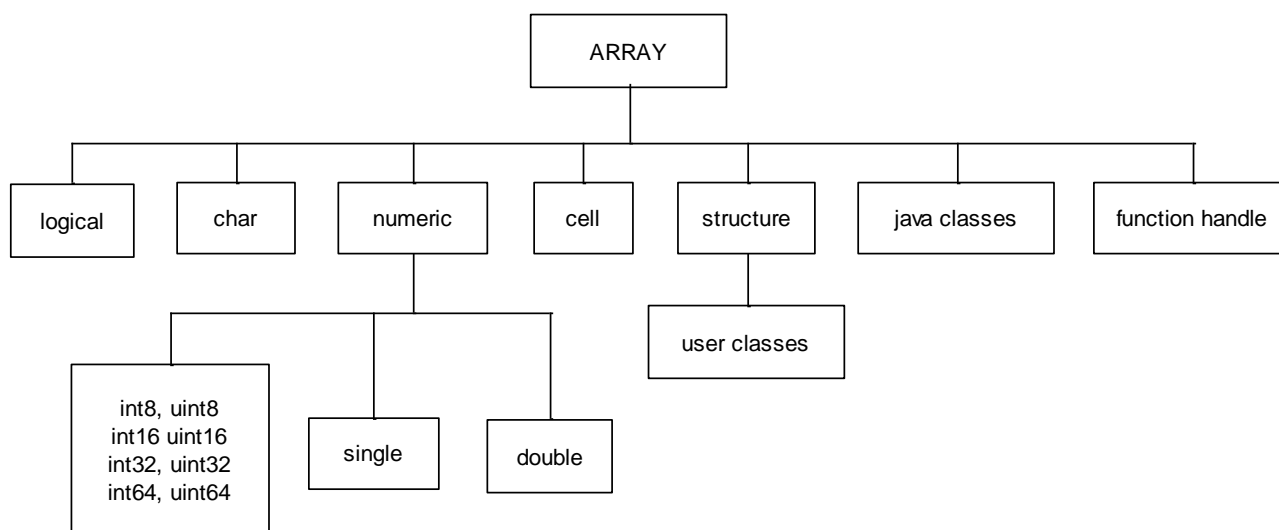
- *ans* – obsahuje poslední zobrazenou nepojmenovanou hodnotu,
- *eps* – určuje relativní přesnost při výpočtech,
- *inf* – reprezentace nekonečna, dělení nulou,
- *NaN* – neplatná číselná hodnota, je důsledkem nedefinovaných aritmetických operací typu *0/0*
- nebo *inf/inf*,
- *pi* – Ludolfovo číslo,
- *i, j* – imaginární jednotka.

e) *Procedury a funkce* – jsou logicky sdružené posloupnosti příkazů programovacího jazyka.

f) *Výrazy* – kombinace funkcí, operátorů, proměnných, které dávají jako výsledek hodnotu některého numerického datového typu nebo řetězce znaků.

2) Datové typy v MATLABu

Datový typ je základní charakteristikou datových prvků, určuje jejich vlastnosti a způsob datové prezentace v paměti počítače a způsob interpretace této hodnoty. V MATLABu existuje 15 základních datových typů (Obr. 2.2), které jsou vyjádřeny ve formě tzv. datového pole (*array*).



Obr. 2.2: Typy proměnných v MATLABu.^[7]

- Datový typ *logical* – logické datové pole. Obsahuje pouze logické hodnoty 1 a 0. Program vrací tento datový typ při použití relačních nebo logických operátorů a funkcí.
- Datový typ *char* – řetězec znaků nebo znak, kde každý znak má 16 bitů.
- Datové typy *int8*, *uint8*, *int16*, *uint16*, *int32*, *uint32*, *int64* a *uint64* – jsou celočíselné datové typy velikostí 8, 16, 32 a 64 bitů, mají menší nároky na velikost paměti.
- Datový typ *single* – reprezentuje číselné hodnoty v pohyblivé řádové čárce o jednoduché přesnosti.
- Datový typ *double* – reprezentuje číselné hodnoty v pohyblivé řádové čárce ve dvojnásobné přesnosti. Tento numerický datový typ je nejpoužívanější.
- Datový typ *cell* – pole buněk, které obsahují další pole, a tak umožňuje vzájemně vázat propojená data s různou velikostí a různým typem v jedné datové struktuře (indexovaná struktura).
- Datové pole *structure* – strukturované datové pole, které umožňuje soustřeďovat různá data do jedné proměnné a místo indexování se používají jména polí.

2.4 Základní operace^[7]

MATLAB je ve své podstatě zaměřen především na snadnou práci s maticemi. Základním typem proměnné je komplexní matice. Všechny další typy jsou pouze speciální případy – reálné matice, vektory a skaláry. Dimenze proměnné se dynamicky mění podle výsledku přiřazení na pravé straně.

2.4.1 Přiřazovací příkazy, konstanty a maticové proměnné

MATLAB nezná přísné deklarování typu proměnné jako některé jiné programovací jazyky (např. Pascal). Proměnné se nedeklarují, ale definují až při prvním přiřazení hodnoty. Přiřazovací příkaz má tvar: `>> proměnná = výraz`.

Proměnná začíná písmenem a skládá se z libovolného počtu písmen, číslic a podtržitek. Rozlišuje se prvních 19 znaků i malá a velká písmena.

MATLAB používá aritmetiku s pohyblivou řádovou čárkou a s relativní přesností *eps*, což odpovídá 16 dekadickým číslicím.

Reálná čísla jsou zadávána běžným způsobem v dekadickém nebo exponenciálním tvaru.

2.4.2 Vytváření matic a vektorů

Matrice je možné z prvků nebo matic skládat jejich vložení mezi *hranaté závorky*. Jednotlivé prvky (například čísla) se oddělují ve vodorovném směru znakem *mezera* nebo *čárka* a ve svislém směru znakem *konce řádky* nebo *středníkem*. Pro skládání prvků platí, že výsledná matice musí být čtvercová nebo obdélníková, musí obsahovat ve všech řádcích shodný počet základních prvků – komplexních nebo reálných hodnot. Skaláry potom vyjadřujeme jako matice rozměru (1 x 1) a vektory jako matice s jedním řádkem (1 x *n*) nebo jedním sloupcem (*n* x 1). Vektory mohou být zadávány prostým výčtem prvků nebo generováním tzv. dvojtečkovou notací. Důležitou vlastností je možnost skládat nejen číselné hodnoty a skalární proměnné, ale i možnost skládat a rozšiřovat matice. Skládání velkých matic ze *submatic* se řídí podobnými pravidly jako sestavování matic z prvků. Jediným omezením je rozměrová kompatibilita.

Opakem ke skládání prvků do matice je její indexování. K indexování matic se používají *kulaté závorky* mezi které se vloží indexy požadovaného prvku oddělené čárkou. Důležité je si uvědomit, že indexy se zadávají v pořadí řádek, sloupec, a indexování začíná prvkem (1,1). Je možné také pracovat i s vícerozměrnými poli (*N-D arrays*), pak jsou indexy uspořádány podle jednotlivých dimenzí pole. Pro indexaci sloupcových i řádkových vektorů stačí zadat do závorek pouze jednu hodnotu, dokonce na pole libovolné dimenze lze pohlížet jako na vektor vyplňující pole postupně od první k dalším dimenzím. Pro 2D matici to znamená složení jejích sloupců postupně pod sebe. Pokud by index přesahoval dosavadní dimenzi matice *a*, tato by se dynamicky zvětšila a dosud nedefinované prvky by byly rovny 0. Pokud bychom ale odkazovali na neexistující prvek matice na levé straně přiřazení, program by vygeneroval chybové hlášení. Stejně tak by se stalo i v případě nekladného indexu. Speciálním případem matice je *prázdná matice* označená [], např. `p=[]`.

Potřebujeme-li pracovat s komplexními čísly, použijeme zápis ekvivalentní algebraickému zápisu komplexních čísel. Využijeme k tomu předdefinovanou konstantu *i* nebo *j* – imaginární jednotku.

Při matematických výpočtech jsou často potřeba řady. Pro jednoduchý zápis vzestupných aritmetických řad (vektorů a intervalů) se používá symbol *dvojtečka* (:). Vzestupnou řadu (vektor) s inkrementem 1 od hodnoty *x* do hodnoty *y* lze napsat `x:y`. Je-li potřeba definovat jiný inkrement než 1, lze popužít zápis `x:D:y`, kde *D* je hodnota inkrementu.

Na závěr přehledu o maticích zmíníme ještě aspoň náznakem možné typy matic. Matice s reálnými nebo komplexními čísly může být buď plná (*full*) nebo řídká (*sparse*), která obsahuje velké množství nulových prvků. Matice textová, ta odpovídá poli textových řetězců v jiných systémech. Textové konstanty se zadávají vložení posloupnosti znaků mezi *apostrofy*. Je možné používat i celočíselná pole s prvky různé bitové délky *s* a *bez* znaménka (`int8`, `uint8`, `int16`, `uint16`, `int32`, `uint32`). Dále lze definovat struktury, pole prvků s rozdílnými typy (`cell array`) a definovat nové typy. Pro jednotlivé typy mohou být definovány specifické funkce podobné metodám v objektově orientovaných jazycích.

2.4.3 Zápis dat do souboru a čtení ze souboru

Systém MATLAB umožňuje ukládat a načítat matice *do* nebo *ze* souboru. Soubor může být buď textový nebo binární. Textový soubor obsahuje hodnoty pouze z jedné matice. Do binárního souboru lze uložit specifikované proměnné a nebo veškeré proměnné existující v daném okamžiku

v prostředí MATLABU.

Příkaz *save* použitý bez dalších parametrů uloží všechny použité proměnné v pracovním prostoru do souboru *matlab.mat* v aktuálním adresáři. Použijeme-li tento příkaz s jedním parametrem *nazev*, uloží se proměnné do souboru *nazev.mat*. Formát souboru je binární a speciální pro MATLAB, kam se ukládají i názvy proměnných. Uvedou-li se za název souboru ještě názvy proměnných, uloží se pouze vyjmenované proměnné. Příkaz *save* použitý s uvedením klíčového slova *-ascii* uloží data v textovém formátu. Tato varianta uložení proměnných umožňuje výměnu dat s jinými programy.

Příkaz *load* načte data ze souboru a uloží je do paměti do proměnné stejného názvu jako je název souboru (bez přípony). Data v načítaném souboru mohou být uložena jak v binárním tak v textovém formátu.

Pro práci s diskovými soubory jsou příkazy *dir*, *type*, *delete*, *chdir*, které jsou analogické stejnojmenným příkazům MS-DOSu.

K zaznamenání průběhu práce se po spuštění příkazu: `>> diary jmeno_souboru` do textového souboru nahraje záznam spouštěných příkazů a vše, co se vypisuje na obrazovku. Tento soubor uzavřeme příkazem `>> diary off` a opět spustíme příkazem `>> diary on`.

Pro tisk proměnných slouží příkazy *disp* (bez formátu) a *printf* (s formátem)

2.5 Maticové operace^[7]

2.5.1 Základní maticové operace

Základní operace, které lze na matice aplikovat, jsou sčítání a odečítání (+ a -), unární neboli tečkové plus a mínus (+ a -), maticové násobení (*), umocňování matice skalárem a skaláru maticí (^) a dělení matic. Systém umožňuje dělení i obdélníkových matic (nelze počítat přes funkci determinant), jsou k dispozici dvě operace, značené lomítkem (/) a zpětným lomítkem (\) tzv. pravé a levé maticové dělení. Některé operace je možné provádět i po prvcích (provádí se skalární operace mezi odpovídajícími prvky matic shodné dimenze).

Dále jsou k dispozici relační operace: rovnost (==), nerovnost (~=), menší (<), větší (>), menší rovno (<=) a větší rovno (>=), jejichž výsledkem jsou matice, ve kterých číselná hodnota 1 označuje pozici, pro které byla relace splněna. Ostatní prvky obsahují hodnotu 0. Matice v relaci musí být shodné dimenze, nebo alespoň jedna strana musí být skalár.

Základní operací s maticemi je vytvoření transponované matice. Získá se přidáním znaku *apostrof* za určení matice. Tím může být jak jméno již definované proměnné tak i složení prvků v hranatých závorkách. Transpozice komplexní proměnné znamená současně převod na komplexně sdružená čísla.

Priorita použitých operací je stejná jako v matematice. Dále se předpokládá, že operandy mají stejnou dimenzi.

2.5.2 Operace nad prvky matic (tečkové operace)

Tečka před matematickými operátory *.**, *./*, *.^*, značí součin (dělení, umocňování) vektoru provedený prvek po prvku, tzn. že matematická operace se provádí mezi prvky na sobě odpovídajících pozicích.

Operandy mohou být matice stejných rozměrů, pak se umocní vždy odpovídající dva prvky, nebo může být základ skalár a exponent vektor, nebo naopak základ je vektor a exponent je skalár.

Relační i logické operátory (& – AND, | – OR, ~ – NOT) spadají také do kategorie prvek po prvku. Relační operátory slouží k porovnávání prvků matic stejné dimenze. Výsledkem relační operace je matice nul a jedniček, kde 1 reprezentuje pravdu a 0 nepravdu.

Logickou hodnotu *false* představuje hodnota prvku 0, hodnotu *true* jiná hodnota než 0. Taktéž vracejí matice nul a jedniček.

2.5.3 Práce s polynomy a interpolace

MATLAB podporuje práci s polynomy. Využívá se vektorové reprezentace polynomů. Je zavedena určitá konvence uložení polynomu. Koefficient u nejvyšší mocniny polynomu je uložen

v prvním prvku vektoru a ostatní koeficienty u klesajících mocnin polynomu jsou umístěny postupně do dalších prvků. Je nutné napsat i nulové koeficienty.

Některé funkce pro práci s polynomy:

- >> $r = poly(a)$ – implementuje polynom jako řádkové vektory s koeficienty uspořádanými podle klesajících mocnin,
- >> $r = polyval(p,x)$ – vyčíslí polynom p pro všechny hodnoty x ,
- >> $r = conv(p,q)$ – násobení polynomů p a q ,
- >> $koreny = roots(p)$ – kořeny polynomu p ,
- >> $r = polyfit(x,y,3)$ – nalezení koeficientů polynomu 3. stupně prokládajícího data podle kritéria nejmenších čtverců,
- >> $interp1, interp2$ – nalezení přibližné hodnoty funkce v nějakém intervalu.

Funkce $poly$ a $roots$ jsou inverzní.

S proložením bodů pomocí polynomu úzce souvisí interpolace, což je nalezení hodnot závislosti v předepsaném místě mezi zadanými body. Bližší informace jsou k dispozici zadáním příkazu: >> *help polyfun*.

2.5.4 Práce se soubory a znakovými řetězci

Práci se znakovými řetězci podporuje MATLAB v dostatečném rozsahu. Práci s textem, tzn. se znakovými řetězci, se vlastně vůbec nevyhneme. Napíšeme-li příkaz, jde o textový řetězec zadaný z klávesnice, který musí MATLAB správně zpracovat. Číslo (různě dlouhá posloupnost číselných znaků) se převádí do vnitřního (binárního) kódu, kde se s ním provedou požadované matematické operace, pak se převede zpět do srozumitelného textu – řetězce znaků.

Řetězec, neboli pole znaků, je posloupnost jednotlivých znaků (číslic, písmen, speciálních znaků). Program chápe řetězec znaků jako vektor, kde prvky jsou jednotlivé znaky. Pro vytváření řetězců slouží *jednoduché uvozovky*, mezi které je umístěn daný řetězec znaků – *string*. Takto vytvořený řetězec znaků můžeme uložit do proměnné typu *string*.

Řetězce znaků lze ukládat i do matic. Zde je nutné, aby všechny řetězce měly stejnou délku, tzn. stejný počet znaků. Funkce *char* doplní jednotlivé řetězce o příslušný počet mezer a vytvoří matici, jejíž řádky tvoří uvedené řetězce. Přebytečné mezery lze zpětně odstranit použitím funkce *deblank*.

Se znakovou proměnnou nelze provádět matematické operace.

2.5.5 Výstupní formát dat.

Pokud jsou všechny prvky matice celá čísla, matice se zobrazuje ve formátu bez desetinné tečky. Pokud alespoň jeden prvek matice není celé číslo, je možné použít různých výstupních formátů, které se nastavují příkazem >> *format option*. Option zde může představovat *short*, *long*, *short e*, *long e*, *hex*, +. Pro dlouhé formáty se poslední číslice může zdát nesprávná, ale výstup je přesná dekadická reprezentace binárního čísla uloženého v paměti.

2.6 Maticové funkce^[7]

MATLAB podporuje funkce s různým počtem vstupních i výstupních parametrů. Při definici funkce jsou nadefinována formální jména vstupních a výstupních parametrů. Při volání funkce není nutné naplnit všechny vstupní parametry a ukládat hodnoty všech výstupních parametrů. V těle volané funkce jsou pak nepoužité vstupní parametry nedefinovány a funkce nesmí číst hodnoty těchto parametrů. K informaci o počtech parametrů slouží funkce *nargin* a *nargout*. Definice funkce musí být obsažena v první řádce souboru. Souvislý blok komentáře se označuje symbolem % před textem tohoto komentáře. Příkazy obsahující přiřazení nebo volání funkcí je ve funkcích vhodné ukončovat symbolem ;.

Libovolnou část programu je možno uložit do samostatného souboru s názvem shodným se jménem funkce a s příponou *.m a volat jej v jiném programu jako podprogram. Aby program našel příslušnou funkci (*m-file*), musí se nacházet v aktuálním adresáři obsaženém v cestě k souborům. Definované proměnné se přitom nemění.

Funkce se od podprogramu liší tím, že předává parametry, tj. do funkce se předají jen vstupní parametry a z funkce vystoupí jen výstupní parametry.

2.6.1 Funkce na sestavení speciálních matic a vektorů

MATLAB umožňuje generovat i speciální typy matic, jako jsou jednotková matice nebo matice nul. Tyto matice mají společné funkční volání $A = \text{JmenoMatice}([m\ n])$, kde A je matice o velikosti $m \times n$:

- *eye* – jednotková matice I ,
- *ones* – matice se všemi prvky rovnými 1,
- *zeros* – matice se všemi prvky rovnými 0,
- *rand* – matice s prvky, jež jsou náhodná čísla z rovnoměrného rozdělení,
- *randn* – matice s prvky, jež jsou náhodná čísla z normálního rozdělení,
- *diag* – vytvoří diagonální matici z původní matice,
- *det* – počítá determinant matice.

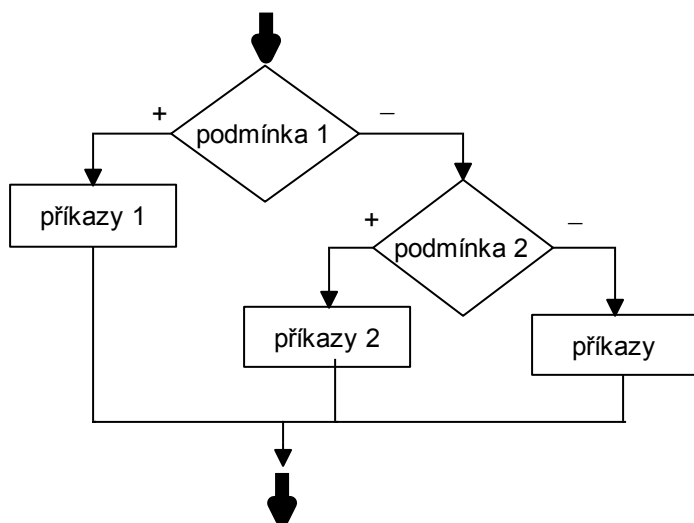
Proměnné, které se v průběhu práce vytvářejí, jsou ukládány v *pracovním prostoru*. Pro sledování a uvolňování aktuálně nadefinovaných proměnných slouží např. příkazy: `>> who` nebo `>> whos`. Oba dávají informaci o definovaných proměnných.

Často potřebujeme matice naplněné speciálními hodnotami, např. nejčastěji používaná matice nebo vektor jsou vytvořeny zadanými rozměry naplněnými konstantou. MATLAB poskytuje dvě možnosti: `>> zeros` – matice naplněná hodnotou 0 a `>> ones` – matice naplněná hodnotou 1.

2.6.2 Řídící příkazy

Je možné používat 4 základní programové konstrukce:

I. Neúplný a úplný podmíněný příkaz *if*.



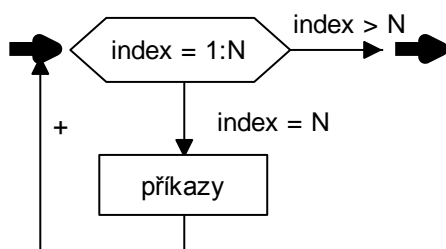
Podmíněný příkaz (větvení), který provede určitou posloupnost příkazů v návaznosti na splnění zadané logické podmínky (Obr. 2.3):

```
if podmínka 1
    příkazy 1
elseif podmínka 2
    příkazy 2
.
.
.
else
    příkazy
end
```

Obr. 2.3 – Vývojový diagram příkazu *if*.

II. Cykly.

a) Cyklus *for* s pevným počtem opakování:

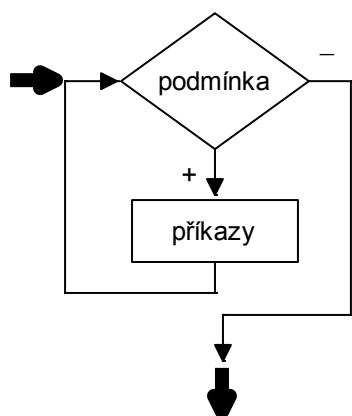


Příkaz cyklu, který provádí určitou posloupnost příkazů N -krát, kde N je předem definovaný počet cyklů (Obr. 2.4):

```
for index = start:krok:konec
    příkazy
end
```

Obr. 2.4 – Vývojový diagram příkazu *for*.

b) Cyklus *while* s podmínkou na začátku.

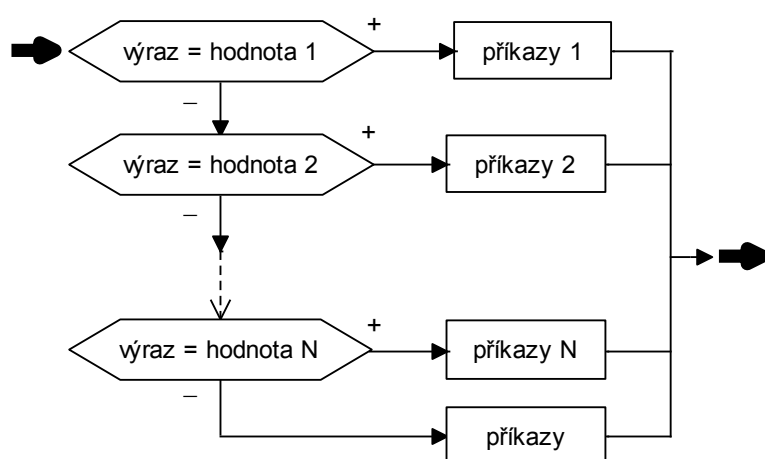


Příkaz podmíněného cyklu, který provádí opakovaně určitou posloupnost příkazů tak dlouho, dokud je splněna určitá logická podmínka (Obr. 2.5):

```
while podmínka
    příkazy
end
```

Obr. 2.5 – Vývojový diagram příkazu *while*.

III. Přepínač *switch*



Rozhodovací příkaz, který provede určitou posloupnost příkazů podle hodnoty, které nabývá daný výraz nebo proměnná (Obr. 2.6):

```
switch výraz
    case hodnota 1
        příkazy 1
    case hodnota 2
        příkazy 2
    .
    .
    otherwise
        příkazy
end
```

Obr. 2.6 – Vývojový diagram příkazu *switch*.

Příkaz *break* ukončí cyklus *for* nebo *while* a pokračuje za koncem cyklu, příkaz *return* slouží k opuštění funkce, program poté pokračuje v běhu, *end* ukončí provádění všech příkazů.

2.7 Vizualizace^[7]

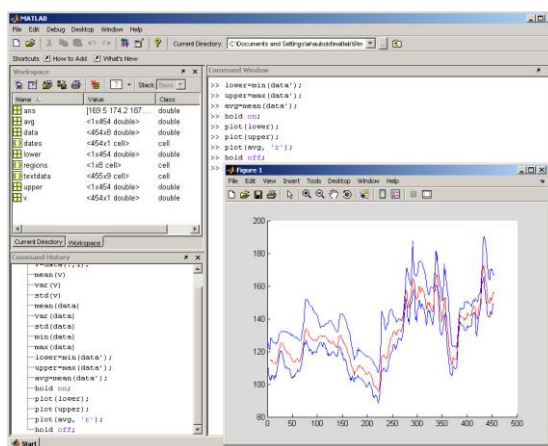
Grafický subsystém MATLABu je dobře strukturovaný a umožňuje snadnou manipulaci s grafy a dialogy na libovolné úrovni. Prostorem pro libovolný grafický výstup i vstup jsou grafická okna (funkce *figure*), kterých může být otevřeno téměř libovolné množství. Do každého okna je možné vkládat další grafické objekty (grafy, tlačítka, posuvné lišty, editační vstupy, výběrové lišty a ostatní prvky obvyklé pro uživatelská rozhraní). Výše uvedené funkce pro vykreslování průběhů a ploch jsou pouze funkce vyšší úrovně využívající objektových vlastností grafického subsystému. Příkaz *plot* zjednodušeně odpovídá tvorbě nového grafického okna, vložení objektu *axes* (osa grafu) a vložení jednotlivých lomených čar (*line*) do takto vytvořeného objektu. Pro tvorbu složitých dialogů je k dispozici kromě mnoha funkcí vyšší úrovně i kompletní návrhář dialogů a oken, který se spouští příkazem *guide*.

Postup při vytváření a úpravě grafů můžeme obecně rozdělit na následující kroky:

- 1) zpracování dat, která chceme zobrazit – výpočet, načtení ze souboru, atd.,
- 2) výběr grafického okna – příkaz *figure(číslo)*,
- 3) vykreslení grafu – specifikace typu grafu, barvy, atd.,
- 4) úprava grafu – nastavení rozmezí os pro zobrazení dat, popis souřadnicových os grafu, anotace grafu, atd.,
- 5) uložení grafu – vhodný výběr grafického souboru nebo tisk na tiskárnu.

a) Dvourozměrné grafy

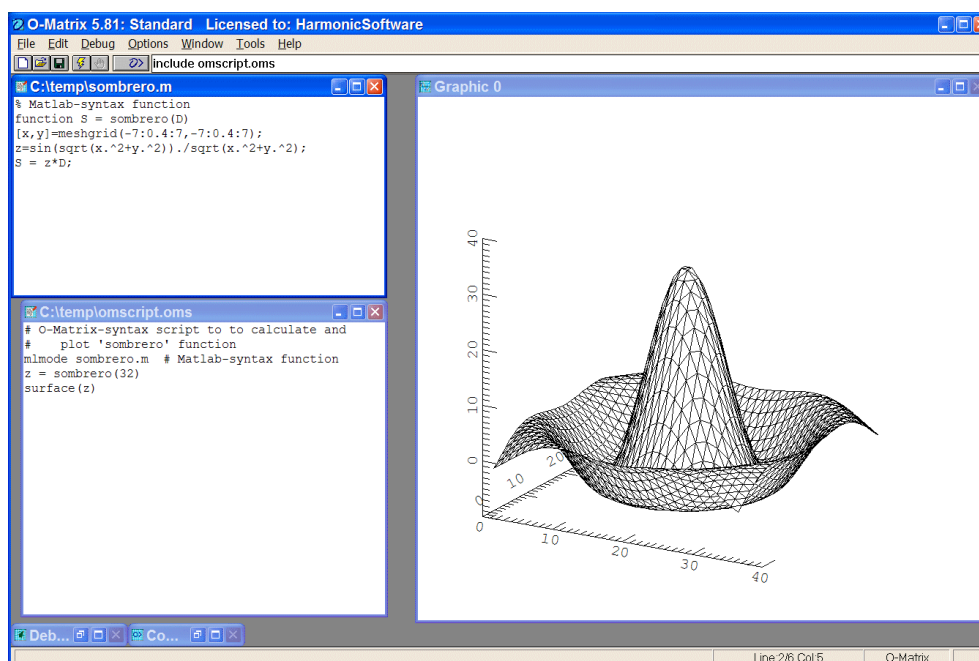
Nejjednodušším příkazem pro zobrazení závislosti dvou proměnných nebo průběhu jedné proměnné je příkaz *plot* (Obr. 2.7). Zobrazovaná data jsou předávána ve formě sloupcových vektorů. Obecně má příkaz podobu *plot(x,y S, ..)*, kde *x* a *S* mohou být vynechány. Transpozice na sloupcový vektor není v tomto případě nutná, příkaz *plot* je inteligentní. V takto vykresleném grafu bude mít osa *x* význam hodnoty indexu v předaném vektoru. Textový parametr *S* specifikuje barvu a způsob vykreslení průběhu. Skupinu parametrů *x*, *y*, *S* lze i několikrát opakovat. Příkazy *title*, *xlabel*, *ylabel* a *text* lze graf doplnit popisy. Výsledný graf lze vytisknout nebo exportovat do souboru. MATLAB umožňuje vykreslit více grafů do jednoho okna a to jak přes sebe (*hold on*) tak i vedle sebe a pod sebe (*subplot*). Je možné libovolně měnit měřítka os, používat logaritmické a semilogaritmické zobrazení.



Obr. 2.7: Ukázka tvorby 2D grafu. [9]

b) Třírozměrné grafy

Příkazy pro kreslení třírozměrných grafů (Obr. 2.8) jsou v základní podobě sice také jednoduché, ale pozornost je třeba věnovat přípravě dat pro vykreslení grafu nejen co se týká vlastních hodnot, ale mít také rozumné požadavky na jejich počet. Při tvorbě matice definující plochu v prostoru je nutné si uvědomit, že počet bodů narůstá s kvadrátem rozměru matice a že vykreslení plochy složené z velkého počtu bodů může v závislosti na zvoleném typu grafu a na rychlosti procesoru trvat i značnou dobu.

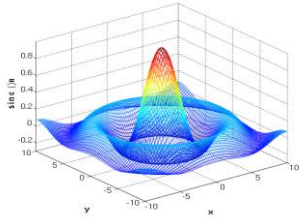
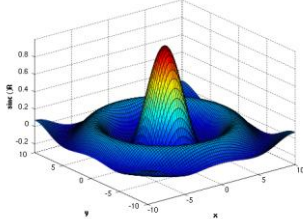


Obr. 2.8 – Ukázka 3D grafu. [10]

3D – grafy mohou být buď čárové nebo plošné (Obr. 2.9). Čárový graf v prostoru nakreslíme velmi jednoduše příkazem *plot3*, máme-li k dispozici tři vektory stejné délky, které obsahují

souřadnice x , y a z – průběh funkce. K vykreslení plošného grafu již potřebujeme minimálně matici hodnot funkce z a případně ještě vektory hodnot x a y definující síť, ve které jsou funkční hodnoty matice vypočteny.

Vlastní vykreslení prostorového pohledu se provede příkazem *mesh* (drátový model) nebo příkazem *surf* (stínovaná plocha).

3D drátové zobrazení	3D povrchové zobrazení
<pre>[X, Y] = meshgrid (-10:0,25:10,-10:0,25:10); f = sinc (sqrt((X/pi).^2+(Y/pi).^2)); mesh (X,Y,f); axis ([-10 10 -10 10 -0.3 1]) xlabel ('\bfx') ylabel ('\bfy') zlabel ('\bfsinc' ('\ bfR')) hidden off</pre>	<pre>[X, Y] = meshgrid (-10:0,25:10,-10:0,25:10); f = sinc (sqrt((X/pi).^2+(Y/pi).^2)); surf (X,Y,f); axis ([-10 10 -10 10 -0.3 1]) xlabel ('\bfx') ylabel ('\bfy') zlabel ('\bfsinc' ('\ bfR)) hidden off</pre>
	

Obr. 2.9 – Ukázka 3D grafu ve drátovém a povrchovém zobrazení.^[6]

c) Další typy grafů

Kromě standardních typů grafů uvedených v předchozích kapitolách nabízí MATLAB ještě další běžné typy grafů: histogram, koláčové 2D a 3D grafy, sloupcové 2D a 3D grafy, atd.

d) Formátování a popis grafu

Po vytvoření grafu je nutné provést základní grafické úpravy. Proměnná *LineStyle* je řetězec znaků, který umožňuje stanovit barvu značky a čáry, typ značky a typ čáry spojující datové body. Řetězec může být obecně složen z několika typů symbolů.

2.8 M - soubory^[7]

Existují dva typy *M-souborů*, script soubor a funkční soubor.

a) Script soubory

Script soubory jsou textové soubory s příponou **.m*, které umožňují provést sekvenci příkazů najednou, aniž by je bylo nutné zadávat z klávesnice. *Script* soubor může obsahovat i volání jiných *M-souborů* nebo sám může být rekurzivní. Používá se hlavně tam, kde potřebujeme opakovat činnost se stále stejnými daty nebo proměnnými určitých jmen. Vytváříme ho použitím textového editoru (Notepad, e-editor nebo použijeme vlastní editor MATLABU). Spustíme ho příkazem daným názvem daného souboru bez přípony. Proměnné, které byly před spuštěním scriptu definovány, lze ve scriptu použít. Proměnné vytvořené v rámci provádění scriptu zůstanou po jeho skončení zachovány.

b) Funkce

Funkce se od *script* souboru liší tím, že má vstupní argumenty a výstupní argumenty. Funkční soubor je v první řádce *M-souboru* uveden slovem *function*. Proměnné definované a používané uvnitř funkčního souboru jsou lokální a nemohou být používány globálně v paměťovém souboru MATLABU (jako proměnné u *scriptu*). Chceme-li ale použít globální proměnnou i v definované funkci, použijeme klíčové slovo *global* před názvem proměnné.

Funkce v MATLABU jsou značně flexibilní. Je možné vyvolat funkci nejen s různými typy vstupních parametrů (skalár, vektor, matice), ale i s různým počtem vstupních a výstupních parametrů. Uvnitř funkce je možné tuto skutečnost zjistit (funkce *nargin* a *nargout*) a např. podle přítomnosti výstupních parametrů měnit činnost funkce. Navíc použití objektů dovoluje měnit význam funkce podle typu objektu.

c) Editor

Součástí MATLABU je i vlastní editor, který zároveň slouží jako *debugger* (speciální prostředek usnadňující ladění vytvořených scriptů a funkcí. Praktickou možností je vykonání označených příkazů bez opuštění editoru.

MATLAB při ladění poskytuje ještě možnost zjistit časovou náročnost prováděných výpočtů. Jde o tzv. *profiler*. Požadavek na zaznamenávání se provede příkazem *profile on*, spustí se sledovaný script nebo sledovaná funkce, ukončení zaznamenávání se provede příkazem *profile off*. Příkaz *profile report* vygeneruje protokol v HTML formátu a pomocí prohlížeče ho zobrazí. Zobrazí se celkový přehled a detailní informace o jednotlivých funkcích (v této informaci je vidět i kolikrát se funkce vykonávaly a kolik času se spotřebovalo na jednotlivé řádky).

2.9 Toolbox (Tbx)

Jedná se o souhrn metod a vědomostí snad ze všech oborů, ve kterých se využívá exaktního matematického nebo číselného vyjadřování. Kromě knihoven pro statistiku, aproximace a prokládání funkcí a dalších obvyklých součástí i jiných matematických systémů, obsahuje balíky pro identifikaci, modelování a návrh řízení i složitých systémů, filtrace, komprese a zpracování obrazů a další. Některé balíky obsahují i předdefinované modely pro ověřování dynamických vlastností systémů. Většina balíků obsahuje i demonstrační příklady, s jejichž pomocí je možné snadno přejít k řešení vlastních problémů.

Toolboxy jsou určeny pro obecné použití, pro zpracování signálu a obrazu, pro návrh řízení, pro řešení problematiky v určité oblasti, a tudíž nejsou základní součástí nabídky MATLABU.

- **Control System Toolbox**^[11]

Control System Toolbox je aplikační knihovna, která rozšiřuje systém MATLAB o nástroje pro řídicí techniku a teorii systémů. Je určen pro analýzu a syntézu lineárních, časově invariantních dynamických systémů spojitých i diskrétních objektů (LTI). LTI funkce z oblasti analýzy a návrhu řídicích systémů využívají jak klasické přechodové charakteristiky, tak i popisy systémů ve stavovém prostoru jsou struktury popisující jednorozměrové i mnoharozměrové lineární systémy. Do LTI lze kromě popisu struktury systému uložit i mnoho dalších vlastností, jako je vzorkovací frekvence, pojmenování vstupních a výstupních signálů, a další uživatelská data. Tyto informace lze samostatně editovat a ukládat v kterémkoli časovém okamžiku a tak přehledně dokumentovat jednotlivé stavy systému během experimentu.

- **System Identification Toolbox**^[11]

System Identification Toolbox je určen pro vytváření matematických modelů systémů z naměřených dat. Poskytuje nástroje pro vytvoření matematických modelů dynamických systémů založené na sledování vstupních a výstupních dat. Pro práci využívá grafické uživatelské rozhraní, které usnadňuje práci při organizaci dat a modelů. Nástroje pro identifikaci poskytované v tomto toolboxu jsou určené pro užití v oblasti od návrhu regulátoru a zpracování signálu až k časové a vibrační analýze.

- **Optimization Toolbox**^[11]

Optimization Toolbox je určen pro minimalizaci a maximalizaci funkcí. Obsahuje funkce určené pro minimalizaci (nebo maximalizaci) obecných nelineárních funkcí. Také obsahuje funkce pro nelineární vyrovnaní řešení a funkce pro řešení problému nejmenších čtverců. Všeobecně je nástrojem pro řešení různých úloh statické optimalizace.

- **Signal Processing Toolbox**^[11]

Signal Processing Toolbox je určen pro syntézu filtrů. Je tedy určen ke zpracování signálů. Podporuje široké pole operací ke zpracování signálů od generování časových průběhů signálu po

návrh filtrů a jejich implementaci, parametrické modelování a spektrální analýzu. Toolbox nabízí dvě kategorie nástrojů, funkce pro zpracování signálů a grafické interaktivní nástroje.

Nástavbou MATLABu je *Simulink* – program pro simulaci a modelování dynamických systémů, který využívá algoritmy MATLABu pro numerické řešení především nelineárních diferenciálních rovnic.

2.10.1 Příklady

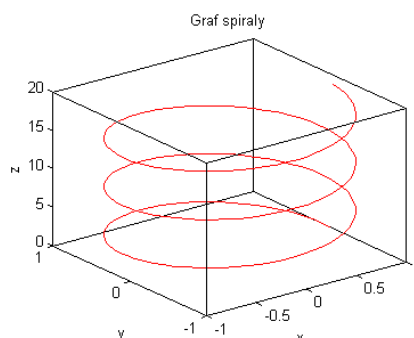
1) Příklad k řešení

Sestrojte graf funkce $z = \frac{\sin(r)}{r}$, kde $r = \sqrt{x^2 + y^2}$ je vzdálenost od počátku souřadnic v rovině xy . Diskutujte použitelnost grafu v různých možných zobrazeních.

2) Řešený příklad^[12]

Vytvoření třídimenzionálního grafu (Obr 2.10): spirála popsaná parametricky $x(t) = \cos(t)$, $y(t) = \sin(t)$, $z(t) = t$.

<code>>> t = 0:0.1:6*pi;</code>	parametr t ; rozsah odpovídá třem otáčkám spirály
<code>>> x = cos(t);</code>	x -ová a y -ová souřadnice opisují kružnici
<code>>> y = sin(t);</code>	
<code>>> z = t;</code>	z -ová souřadnice se mění lineárně
<code>>> plot3(x,y,z,'r')</code>	vykreslení grafu spirály (červeně) (Obr 2.10)
<code>>> box</code>	Orámování grafu – příkaz <code>box</code> pracuje jako přepínač, jeho opakování funkci střídavě zapíná a vypíná. Rovněž lze použít <code>box on</code> , <code>box off</code> .
<code>>> xlabel('x')</code>	Popis osy x
<code>>> ylabel('y')</code>	Popis osy y
<code>>> zlabel('z')</code>	Popis osy z
<code>>> title('Graf spirály')</code>	Přidání titulku



Obr 2.10: Graf spirály.^[12]

Pro vykreslení funkce $z = z(x,y)$ existuje několik příkazů; nejdůležitější jsou *mesh* (sít, síťový graf) a *surf* (surface - plocha).

KAPITOLA 3: METODA MONTE CARLO^[13]

Metoda Monte Carlo patří mezi užitečné matematické postupy. Vznikla z konkrétních požadavků na řešení složitých problémů fyziky, matematiky, ostatních přírodních věd, techniky, ekonomie, atd. Opírá se o pojmy z *pravděpodobnosti a statistiky* a umožňuje řešit problémy obtížně řešitelné tradičními metodami. Tato metoda je především mistrovskou ukázkou schopnosti matematicky modelovat a poté simulovat složité jevy, a matematickou formou dospět v některých případech k výsledku rychleji oproti tradičním postupům. Použití metody Monte Carlo je rozsáhlé, ale řešení problému pomocí této metody nemusí být vždy tou nejvýhodnější metodou jak z hlediska jednoduchosti, přesnosti i rychlosti výpočtu.

Metoda Monte Carlo je souhrn postupů umožňujících pomocí mnohonásobných náhodných pokusů získat řešení problémů a to nejen v počítačové fyzice. Tato metoda patří mezi částicové metody. Je to metoda stochastická, což znamená, že hledaný výsledek je získáván na základě počtu pravděpodobností. Princip metody je velmi starý a proto k realizaci náhodných pokusů vlastně nepotřebujeme pomoc výpočetní techniky.

3.1 Schéma řešení problému pomocí metody Monte Carlo

a) Analýza problému a vytvoření modelu

Cílem analýzy zkoumaného jevu je jeho popsání pomocí náhodné veličiny. Vytvoření modelu pak znamená zjednodušeně popsat zkoumaný jev pomocí konkrétní náhodné veličiny s daným oborem hodnot a rozdělením pravděpodobností a současně určit, která charakteristika této náhodné veličiny obsahuje námi hledaný výsledek.

b) Generování náhodné veličiny

Generování náhodné veličiny na počítači se provádí ve dvou krocích:

1. nagegenerování náhodné veličiny s určitým pevně daným rozdělením pravděpodobností,
2. přetransformování předešlé veličiny v hledanou náhodnou veličinu.

c) Statistické vyhodnocení výsledků

Předchozími kroky dostaneme pouze jednu realizaci náhodné veličiny. Proto musíme předchozí kroky opakovat, přičemž počet opakování musí být veliký. Získáme postupně hodnoty $\xi_1, \xi_2, \dots, \xi_N$. Tyto hodnoty podrobíme statistické analýze a podle formulace daného problému z nich získáme hledanou odpověď.

V případě experimentální realizace stochastického modelování bylo prokázáno, že chyba metody klesá s počtem pokusů N podle vztahu:

$$\mathcal{G} \sim \frac{1}{\sqrt{N}}.$$

Z předchozího vyplývá, že první krok je tvůrčí část, a následující kroky jsou rutinní částí experimentu.

Metoda Monte Carlo realizuje řešení problémů pomocí mnohokrát opakovaných náhodných pokusů. Odhady hledané veličiny se získávají statistickou cestou a mají tedy pravděpodobnostní charakter. Označíme v tomto případě odhady $\theta_1, \theta_2, \dots, \theta_N$ za hledané hodnoty veličiny θ , jež získáme statistickým zpracováním experimentálních dat. Požadujeme, aby v tomto případě veličina θ_n , kde n značí počet pokusů, která je náhodnou veličinou, při $n \rightarrow \infty$ konvergovala k hledané hodnotě θ podle pravděpodobnosti. Tím se rozumí splnění vztahu, aby pro libovolně malé $\varepsilon > 0$ platilo:

$$\lim_{n \rightarrow \infty} P(|\theta_n - \theta| < \varepsilon) = 1,$$

kde θ_n má charakter statistických odhadů a souvisí s hledanou hodnotou θ prostřednictvím pravděpodobnostních zákonitostí.

3.2 Stručné shrnutí pojmů statistiky

Nyní uvedeme stručný přehled základních pojmů z počtu pravděpodobnosti. Výběr témat je podřízen metodě Monte Carlo.

1) Náhodná veličina

Náhodné veličiny jsou veličiny, u které známe všechny hodnoty, kterých mohou nabývat, a pravděpodobnosti jejich nabytí, avšak nedovedeme předpovědět jejich hodnotu v konkrétním případě. Dělíme je na *diskrétní* a *spojité* podle toho, jakých hodnot mohou při realizacích nabývat.

Pro každou náhodnou veličinu se zavádí tzv. *zákon rozdělení náhodné veličiny*, který každé hodnotě nebo množině hodnot z určitého intervalu přiřazuje pravděpodobnost, že náhodná veličina této hodnoty nabude. Existují dvě varianty tohoto zákona:

a) Distribuční funkce

Přiřazuje každému reálnému číslu pravděpodobnost, že náhodná velikost nabude hodnot menších než toto číslo:

$$F(x) = P(\xi < x),$$

kde $F(x)$ je distribuční funkce, symbolem $P(y)$ se označuje pravděpodobnost výskytu jevu y a ξ je náhodná veličina.

Pravděpodobnost nemožného jevu je nulová a pravděpodobnost jistého jevu se rovná 1. Distribuční funkce $F(x)$ proto nabývá hodnot z intervalu $\langle 0,1 \rangle$, je neklesající a platí pro ni

$$P\{x_1 \leq \xi \leq x_2\} = F(x_2) - F(x_1) \quad \text{pro } x_1 < x_2.$$

b) Pravděpodobnost

Distribuční funkce má díky své definici $F(x) = P(\xi < x)$ integrální význam, shrnuje výslednou pravděpodobnost za určitý interval. Pokud chceme pracovat s konkrétními pravděpodobnostmi, musíme zavést odpovídající diferenciální veličiny, např. $p(x)$

$$p(x) = \frac{dF(x)}{dx} \quad \text{neboli} \quad F(x) = \int_{-\infty}^x p(y) \cdot dy.$$

Oba popisy jsou ekvivalentní, ale častěji se používá práce s pravděpodobnostmi, protože je bližší fyzikálnímu způsobu vyjadřování.

2) Typy náhodných veličin

a) Diskrétní náhodné veličiny

U tohoto typu náhodných veličin můžeme zákon rozdělení náhodné veličiny popsat množinou hodnot x_i a odpovídajícími pravděpodobnostmi p_i , kde:

$$p_i = P\{\xi = x_i\}$$
$$\xi = \begin{pmatrix} x_1 & x_2 & \dots & x_n \\ p_1 & p_2 & \dots & p_n \end{pmatrix}.$$

Hodnoty x_1 až x_n , kterých může náhodná veličina ξ nabývat, mohou být libovolná čísla. Pro pravděpodobnosti p_1 až p_n platí dvě omezení:

$$p_i > 0, \quad i = 1, 2, \dots, n$$

$$\sum_{i=1}^n p_i = 1.$$

Místo pravděpodobností p_i můžeme použít i distribuční funkci $F(x)$, která bude mít případně tvar skokové funkce.

b) Spojité náhodné veličiny

Spojitá náhodná veličina ξ nabývá hodnot x z nějakého konečného nebo nekonečného intervalu. Pro zápis můžeme použít jak distribuční funkci $F(x)$, která bude pro spojitou náhodnou veličinu spojitá, tak tzv. hustotu pravděpodobnosti náhodné veličiny ξ v bodě $x - p(x)$. Spojitá náhodná veličina ξ bude charakterizována:

$$\xi: x \in \langle a, b \rangle, \quad p(x),$$

kde hustota pravděpodobnosti $p(x)$ má tyto vlastnosti:

$$p(x) \geq 0, \quad x \in \langle a, b \rangle,$$

$$\int_{-\infty}^{\infty} p(y) \cdot dy = 1,$$

$$P\{x_1 \leq \xi \leq x_2\} = \int_{x_1}^{x_2} p(y) \cdot dy.$$

Můžeme pracovat i s vícerozměrnými náhodnými veličinami. Pro popis použijeme sdruženou distribuční funkci:

$$F(x, y) = P\{x < \xi, \eta < y\}.$$

3) Charakteristiky náhodných veličin

Náhodná veličina X je jednoznačně určena rozdělením pravděpodobnosti pomocí pravděpodobnostní funkce nebo distribuční funkce (popř. hustoty pravděpodobnosti). Tyto funkce jsou však často poměrně složité a jejich určení pracné. Proto je výhodné shrnout informace o náhodné veličině do několika čísel, které ji dostatečně charakterizují. Tato čísla nazýváme *číselné charakteristiky* a dělíme je:

a) Podle způsobu konstrukce na charakteristiky:

- momentové: jsou konstruovány na základě počátečního momentu μ_k nebo centrálního momentu ν_k ,
- kvantilové: jsou obvykle odvozeny pomocí distribuční funkce $F(x)$ a jsou určovány pro spojitou náhodnou veličinu, pro diskrétní náhodnou veličinu nebývá jejich určení jednoznačné,
- ostatní.

b) Podle toho, které vlastnosti rozdělení pravděpodobnosti charakterizují, na charakteristiky:

- polohy: $E(X)$, Me , Mo , kvantily. Určují jakýsi "střed", kolem něhož kolísají hodnoty náhodné veličiny X ,
- variability: $D(X)$, σ , Ukazují rozptýlenost hodnot náhodné veličiny kolem střední hodnoty,
- šikmosti a špičatosti: charakterizují průběh rozdělení náhodné veličiny X .^[14]

Někdy je vhodné použít jen zkrácenou formu zápisu udávající pouze základní vlastnosti náhodné veličiny. Pro tento účel byl navržen systém charakteristik – momenty náhodných veličin. Uvedeme ty nejpoužívanější:

c) Charakteristiky polohy

Jednou ze základních charakteristik každé náhodné veličiny je její *střední hodnota s ohledem na rozdělení pravděpodobnosti*, označuje se $E\xi$ a nejčastěji se nazývá očekávaná hodnota. Definiční vztahy pro diskrétní a spojitou náhodnou veličinu jsou:

$$E\xi = \sum_i x_i \cdot p_i,$$

$$E\xi = \int_{-\infty}^{\infty} x \cdot p(x) \cdot dx.$$

d) Charakteristiky variability

Udávají rozptyl možných hodnot náhodné veličiny ξ kolem její střední hodnoty $E\xi$. Definiční vztah společný pro diskrétní i spojité náhodné veličiny se nazývá *rozptyl*, *variance* nebo *disperze* a označuje se $D\xi$:

$$D\xi = E(\xi - E\xi)^2.$$

Tento vztah se ale častěji převádí do výhodnějšího ekvivalentního vyjádření:

$$D\xi = E(\xi^2) - (E\xi)^2.$$

Odvozená jednotka od rozptylu se nazývá *směrodatná odchylka* a označuje se σ_ξ :

$$\sigma_\xi = \sqrt{D\xi}.$$

e) Charakteristiky vyšších řádů

Definovat *centrální moment k-tého řádu* můžeme za předpokladu, že existuje $E\xi$ a má konečnou hodnotu:

$$\mu_k = E(\xi - E\xi)^k, \quad k = 0, 1, 2, \dots$$

Na základě momentu třetího řádu je definována *šikmost*:

$$\alpha_3 = \frac{\mu_3}{\sigma_3^3}.$$

U symetrických rozdělení je tato charakteristika nulová. Je-li kladná, je rozdělení pravděpodobnosti zešikmené doleva, je-li záporné tak doprava.

Normováním centrálního momentu 4. řádu definujeme *špičatost*:

$$\alpha_4 = \frac{\mu_4}{\sigma_4^4}.$$

Bude-li $\alpha_4 > 3$, bude studované rozdělení špičatější než normální rozdělení (Gaussovo rozdělení), pro menší hodnoty je rozdělení plošší.

4) Vybrané náhodné veličiny

Nejčastěji používaná rozdělení pro diskrétní veličiny jsou binomické rozdělení, Poissonovo rozdělení a rovnoměrné rozdělení pro diskrétní náhodnou veličinu, pro spojité veličiny rovnoměrné rozdělení pro spojitou náhodnou veličinu, Gaussovo a Maxwelllovo rozdělení.

a) Binomické rozdělení

Tímto rozdělením se řídí četnost náhodného jevu v n nezávislých pokusech, když v každém pokusu má výskyt jevu pravděpodobnost p , kde n je přirozené číslo a $p \in \langle 0, 1 \rangle$:

$$\xi = \begin{pmatrix} 0 & 1 & \dots & n \\ p_0 & p_1 & \dots & p_n \end{pmatrix}, \quad p_x = P(\xi = x) = \binom{n}{x} \cdot p^x \cdot (1-p)^{n-x}.$$

Základní charakteristiky rozdělení jsou:

$$\begin{aligned} E\xi &= n p & D\xi &= n p (1-p), \\ \alpha_3 &= \frac{1-2p}{\sqrt{n p (1-p)}} & \alpha_4 &= \frac{1-6p(1-p)}{n p (1-p)} + 3. \end{aligned}$$

b) Poissonovo rozdělení

Popisuje proces, při kterém studujeme četnost nějakého jevu v mnoha pokusech, když výskyt tohoto jevu v jednotlivých pokusech je jen velmi málo pravděpodobný. Z binomického rozdělení se získá limitním přechodem, kde $p \rightarrow 0$, $n \rightarrow \infty$ a $n \cdot p = \lambda$ konečné. λ je jediným parametrem Poissonova rozdělení.

$$\xi = \begin{pmatrix} 0 & 1 & \dots & n \\ p_0 & p_1 & \dots & p_n \end{pmatrix}, \quad p_x = P\{\xi = x\} = e^{-\lambda} \cdot \frac{\lambda^x}{x!} = \frac{1}{e^\lambda + 1} \cdot \frac{\lambda^x}{x!}.$$

Základní charakteristiky rozdělení jsou:

$$E\xi = \lambda \qquad D\xi = \lambda,$$

$$\alpha_3 = \frac{1}{\sqrt{\lambda}} \qquad \alpha_4 = \frac{1}{\lambda} + 3.$$

c) Rovnoměrné rozdělení pro diskrétní náhodnou veličinu

Náhodná veličina ξ s rovnoměrným rozdělením může nabývat m hodnot $1, 2, \dots, m$ se stejnými pravděpodobnostmi:

$$p = P(\xi = x) = \frac{1}{m}.$$

První dva momenty této veličiny jsou:

$$E\xi = \frac{m+1}{2} \qquad D\xi = \frac{m^2-1}{12}.$$

d) Rovnoměrné rozdělení pro spojitou náhodnou veličinu

Náhodná veličina ξ zavedená v intervalu $\langle a, b \rangle$ má rovnoměrné rozdělení tehdy, má-li v tomto intervalu konstantní hustotu pravděpodobnosti:

$$p(x) = \frac{1}{b-a} \quad x \in \langle a, b \rangle.$$

Základní charakteristiky rovnoměrného rozdělení jsou:

$$E\xi = \frac{a+b}{2} \qquad D\xi = \frac{(b-a)^2}{12},$$

$$\alpha_3 = 0 \qquad \alpha_4 = \frac{9}{5}.$$

e) Gaussovo rozdělení

Obecně bývá normální rozdělení pro popis daného jevu použitelné v těch případech, kdy na rozptyl hodnot náhodné veličiny působí současně velký počet nepatrných a navzájem nezávislých vlivů. Hustota pravděpodobnosti tohoto rozdělení je dána vztahem:

$$f(x) = \frac{1}{\sqrt{2\pi} \cdot \sigma} \cdot \exp\left[-\frac{(x-\mu)^2}{2\sigma^2}\right] \quad x \in (-\infty, \infty).$$

Jeho distribuční funkce je:

$$F(x) = \frac{1}{\sqrt{2\pi} \cdot \sigma} \cdot \int_{-\infty}^x e^{-\frac{(t-\mu)^2}{2\sigma^2}} dt.$$

Gaussovo rozdělení má dva parametry: μ a σ , kde $\sigma > 0$.

Základní charakteristiky Gaussova rozdělení jsou:

$$E\xi = \mu \qquad D\xi = \sigma^2 \qquad \alpha_3 = 0 \qquad \alpha_4 = 3.$$

f) Maxwelllovo rozdělení

Toto rozdělení, používané zejména v kinetické teorii plynů, má jeden parametr $a > 0$. Hustota Maxwelllova rozdělení je dána předpisem:

$$f(x) = \frac{2}{a^3 \sqrt{2\pi}} \cdot x^2 \cdot \exp\left(-\frac{x^2}{2a^2}\right) \quad x \in \langle 0, \infty \rangle.$$

Základní charakteristiky rozdělení jsou:

$$E\xi = \frac{3a}{\sqrt{2}} \qquad D\xi = 2a^2.$$

5) Vybrané limitní věty

Z teorie pravděpodobnosti jsou vybrány ty, na nichž je založeno statistické vyhodnocování experimentálních dat a modelování reálných procesů metodou Monte Carlo.

a) Zákon velkých čísel

Budou-li $\xi_1, \xi_2, \dots, \xi_n$ nezávislé náhodné veličiny se stejným rozdělením, pak jejich aritmetický průměr bude:

$$\bar{\xi} \cong \frac{1}{n} \cdot \sum_{i=1}^n \xi_i,$$

konverguje podle pravděpodobnosti k μ . Toto tvrzení znamená, že pro každé kladné ε platí:

$$\lim_{n \rightarrow \infty} P \left\{ \left| \frac{1}{n} \cdot \sum_{i=1}^n \xi_i - \mu \right| < \varepsilon \right\} = 1.$$

Pro větší počet nezávislých pozorování náhodné veličiny ξ můžeme proto jejich aritmetický průměr použít pro odhad střední hodnoty $E\xi$.

Rozptyl aritmetického průměru $\bar{\xi}$ je dán vztahem:

$$D\bar{\xi} = \frac{D\xi_1 + \dots + D\xi_n}{n^2} = \frac{\sigma^2}{n}.$$

Směrodatná odchylka $\sqrt{D\bar{\xi}}$ pak bude úměrná druhé odmocnině z počtu pozorování n .

b) Centrální limitní věta počtu pravděpodobnosti

Budou-li $\xi_1, \xi_2, \dots, \xi_n$ nezávislé náhodné veličiny se stejným rozdělením, které má střední hodnotu μ a rozptyl σ^2 , pak jejich součet má pro velká n přibližně Gaussovo rozdělení s parametry $N(n\mu, n\sigma^2)$.

3.3 Použití metody Monte Carlo v procesu simulace^[15]

Při řešení příkladů metodou Monte Carlo hledáme převážně hodnotu veličiny ξ , která souvisí s jistým náhodným procesem. Tento proces se modeluje na počítači, čímž se získají realizace některé náhodné veličiny ξ . Pomocí těchto realizací veličiny ξ se odhaduje s určitou přesností hodnota původně hledané veličiny ξ .

Obecně tedy můžeme charakterizovat metodu Monte Carlo takto:

- Výpočet je založen na modelování náhodných procesů a to i tehdy, když se řeší deterministický problém.
- Výsledky jsou zpracovávány statistickými metodami.
- K dosažení potřebné přesnosti je nutné mnohonásobné opakování simulací.

Z toho plyne, že:

- Musíme znát vlastnosti modelovaného náhodného procesu nebo náhodné veličiny. V praxi to znamená znát rozdělení pravděpodobnosti, které náhodné veličiny a procesy popisují. Tato rozdělení pravděpodobnosti se často určují empiricky. V některých problémech, jako například ve fyzice elementárních částic, jsou tato rozdělení odvozena teoreticky. Přihlížíme i k dalším požadavkům, jako je efektivita výsledného algoritmu nebo snadnost výpočtu.
- Musíme umět získat realizace libovolné náhodné veličiny. Nejprve získáme nezávislé realizace náhodné veličiny γ s rovnoměrným rozdělením v intervalu $(0,1)$ a tyto realizace pak

transformujeme na realizace veličin s požadovaným rozdělením. Efektivita algoritmů pro řešení těchto úloh rozhoduje většinou o efektivitě celého programu.

- 3) Výsledek má obvykle tvar intervalu, v němž hledaná hodnota s jistou pravděpodobností leží. Meze tohoto intervalu bývají určeny pomocí disperze nebo jiných momentů náhodné veličiny. Zde může být výhodné použít odhadů získaných jinými prostředky, například analyticky.
- 4) Zpravidla není možné stanovit předem potřebný počet simulací, určujeme je až v průběhu výpočtu na základě odhadů přesnosti.

Metoda Monte Carlo je často používána jako poslední možnost řešení problému, na kterém ostatní metody selhávají. A proto je nutné kontrolovat správnost algoritmu použitého programu všemi dostupnými prostředky.

3.4 Příklad k řešení:

Pomocí metody Monte Carlo určete objemy těchto těles s danými parametry: polokoule o poloměru 1, válce o poloměru podstavy 0,5 a výšce 1, čtyřstěnu s vrcholy $[0,0,0]$, $[1,0,0]$, $[0,0,1]$ a $[0,1,0]$, rotačního kužele o poloměru 1 a výšce 2. Hodnoty porovnejte s výsledky získanými pomocí obvyklých vzorců.

KAPITOLA 4: GENEROVÁNÍ NÁHODNÝCH ČÍSEL

4.1 Generátory náhodných čísel^[13]

Abychom mohli mluvit o řešení transportního problému týkajícího se generování polohy a rychlosti částic metodou Monte Carlo či případně pomocí programu MATLAB, připomeneme si nejdříve mechanismus generování náhodných čísel.

Generátor náhodných čísel je zařízení nebo procedura, která generuje náhodné číslo. Vstupní parametry mohou být od jaké hodnoty náhodné číslo začíná, jakou hodnotu nesmí peskočit a někdy i maximální rozpětí mezi vygenerovanými čísly. Jejich výhodou je reprodukovatelnost. Začneme-li za stejných počátečních hodnot, dostaneme stejnou posloupnost pseudonáhodných čísel.

Softwarové generátory obvykle generují pseudonáhodná čísla, která by správně měla být statistickými testy nerozeznatelná od skutečných náhodných čísel, nicméně jsou vypočtena deterministicky. Posloupnost nagenерованých pseudonáhodných čísel je vždy konečná, po uplynutí tzv. periody p se začne celá posloupnost opakovat. V rámci jedné periody se čísla vyskytují relativně náhodně, ale tato náhodnost není dokonalá.

Cílem teorie generátorů pseudonáhodných čísel je vybrat algoritmus jejich generování tak, aby oba rysy co nejméně rušily, tj. vytvořit posloupnosti s periodou výrazně delší (alespň 10x) než je v experimentu počet použitých náhodných čísel a v rámci jedné periody mít čísla rozdělena co nejnáhodněji. Vytvoření kvalitního generátoru pseudonáhodných čísel je složitá záležitost. Generátor je nejprve vybrán na základě nějakých teoretických předpokladů a pak dlouhodobě velice důkladně testován. Touto problematikou se převážně zabývají profesionálové (programátoři i matematici) a proto lze předpokládat, že generátory zabudované do překladačů hlavních programovacích jazyků dodávaných softwarovými firmami oba tyto požadavky v rozumné míře splňují.

Algoritmů zajišťujících generování náhodných čísel je mnoho a při nízkých nárocích na opravdovou náhodnost se používají funkce ze systémových knihoven.

Pro generování primární posloupnosti náhodných čísel lze použít jednoho ze dvou základních typů generátorů – fyzikální generátor nebo generátor pseudonáhodných čísel.

Jako základ fyzikálního generátoru náhodných čísel lze použít téměř libovolný fyzikální princip, který má náhodný charakter známých vlastností. Jako příklad lze uvést házení mincí nebo kostkou. K nejběžnějším v praxi používaným generátorům patří šumové generátory využívající vlastnosti polovodičového přechodu nebo kombinace radioaktivního zářiče a detektoru. Fyzikální generátory mají řadu nevýhod a proto se používají zřídka. Posloupnost generovaných čísel je neopakovatelná, což ztěžuje vývoj algoritmů, ladění programů a porovnávání výsledků testování. Stabilita vlastností generátorů je závislá na vnějších vlivech a je obtížně dlouhodobě udržitelná. Změny vlastností generátorů nejsou zjistitelné a mohou významně ovlivnit výsledky simulací. Skutečné chování generátorů se často liší od teoretického v důsledku výrobních tolerancí atd. Z popsaných důvodů jsou fyzikální generátory nahrazovány pseudonáhodnými generátory.^[15]

Generátory pseudonáhodných čísel jsou založeny na aritmetických procedurách využívajících rekurentní vzorce. Nejčastěji používané generátory využívají principu lineárního kongruentního generátoru (LCG), který je definován vztahem:

$$x_{i+1} = a_0 x_i + a_1 x_{i-1} + \dots + a_k x_{i-k} + b \pmod{M},$$

kde $k \geq 0$; a , b a M jsou vhodně zvolené konstanty a generovaná čísla jsou celá s rovnoměrným rozložením v rozsahu $0 \leq x_i < M$.^[13]

Vlastnosti takovéto posloupnosti závisí na volbě konstant a , b a M . Tyto konstanty se volí podle typu použitého počítače s ohledem na statistické vlastnosti generované posloupnosti a rychlosti generování. Ačkoli jsou známa některá doporučení z teorie čísel, která volbu konstant usnadňují, je teoretické odvození vlastností kongruenčních generátorů obvykle prakticky nemožné.^[16]

Konkretizací konstant v definičním vztahu můžeme vytvořit tři zjednodušené typy generátorů:

a) Aditivní – založené na vztahu:

$$x_{i+1} = x_{i-k_1} + x_{i-k_2} \pmod{M}.$$

Tyto generátory jsou velmi rychlé, kvalitní, ale jejich slabinou je velmi krátká perioda.

b) Multiplikační – založené na rekurentním vztahu prvního řádu s nulovou hodnotou konstanty b :

$$x_{i+1} = a_0 x_i \pmod{M}.$$

Tyto generátory jsou dostatečně rychlé, ale kvalita výsledku není dostačující.

c) Smíšené – založené na lineárním rekurentním vztahu:

$$x_{i+1} = ax_i + b \pmod{M}.$$

Převážná většina generátorů pseudonáhodných čísel vytváří jednotlivé hodnoty náhodné veličiny rovnoměrně rozdělené v intervalu $(0,1)$. Generátory se od sebe liší tím, zda hodnoty 0 a 1 patří do definičního intervalu. Před použitím konkrétního generátoru ve vlastním programu je třeba zjistit, zda chyby způsobené například vygenerováním náhodného čísla s nulovou hodnotou nebude prováděna operace dělení, protože tato chyba se při ladění programu hledá velmi těžko.

Generátory s posuvnými registry se v uživatelských programech používají jen vyjíměčně, protože jsou v podstatě předurčeny pro přímou hardwarovou realizaci. Integrovaný obvod zabudovaný přímo do počítače generující pseudonáhodná čísla je konstruován podle algoritmu pro generátor s posuvným registrem:

$$b_{i+1} = c_0 b_i + c_1 b_{i-1} + \dots + c_k b_{i-k} \pmod{2},$$

kde $k \geq 1$ a c_k nabývají pouze hodnot 0 nebo 1 (nejméně dvě hodnoty musí být nenulové), konstanta k se volí mnohem větší než u lineárních kongruenčních generátorů – zvýšení kvality generátoru. Je samozřejmě možné i tento generátor naprogramovat, pro dostatečnou efektivitu výsledného kódu je ale třeba mít programovací jazyk umožňující přímou manipulaci s bity, protože místo s celými čísly pracuje pouze s bity b .^[13]

Existují i snahy kvalitu generátorů pseudonáhodných čísel zvýšit, a to jak jejich periodu tak i náhodnost. Za tímto účelem lze kombinovat více generátorů, pokud možno založených na různých typech algoritmů. Jednou z možných cest je jedním generátorem naplnit větší množinu pseudonáhodných čísel a druhým generátorem z této množiny náhodně vybírat konkrétní číslo. První generátor pak nahradí již použité číslo novým a postup se opakuje. Sice se zvýší náhodnost generované posloupnosti čísel, současně se však zpomaluje výpočet a prakticky znemožňuje teoretickou analýzu generovaných výsledků.^[13]

Druhou skupinou náhodných čísel jsou čísla kvazináhodná. Posloupnost těchto čísel vůbec nevypadá jako náhodná čísla. Základní myšlenka generátorů kvazináhodných čísel je založena na požadavku co nejrovnoměrněji pokrýt interval $(0,1)$. I skutečná náhodná čísla pokrývají úsečku $(0,1)$ relativně rovnoměrně, ale s náhodnými fluktuacemi, které zde přítomné nejsou. Generátory kvazináhodných čísel mohou být založeny na libovolném prvočísle: $1/2, 1/4, 3/4, 1/8, 5/8, 3/8, 7/8, 1/16, \dots$ Použití kvazináhodných čísel v počítačové fyzice je velmi omezené. Nelze je použít tam, kde požadujeme náhodnost a nezávislost dvou po sobě jdoucích náhodných čísel (např. souřadnice náhodného bodu), neboť kvazináhodná čísla jsou v tomto ohledu přísně determinovaná. Pokud ale požadujeme od generátoru náhodná čísla, která budou zpracovávána od sebe odděleně, projeví se neexistence fluktuací v posloupnosti kvazináhodných čísel jako výhoda.^[13]

4.2 Transportní problém^[13]

Nyní se vrátíme k nejrozšířenějšímu problému částicového modelování ve fyzice, k transportnímu problému, při kterém studujeme polohu částice a její rychlost. Transportní problém lze popsat souborem algoritmů a na tuto formulaci pak převést značné množství nejrozličnějších fyzikálních úloh. Pokud budeme řešit transportní problém jinou metodou než metodou Monte

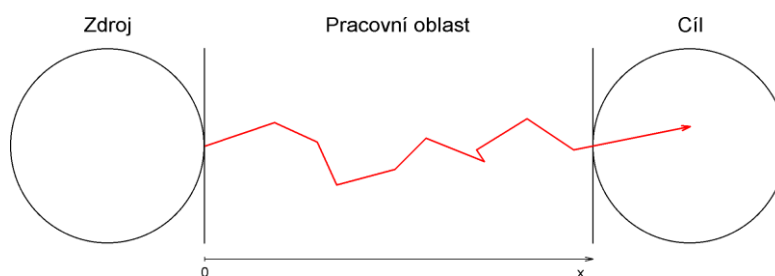
Carlo, musíme detailně znát celý problém transportu, abychom ho mohli nejlépe napodobit spojitým nebo deterministickým částicovým modelováním. Při počítačovém modelování musíme spolu s výběrem použitých algoritmů věnovat pozornost i datovým strukturám, kam budou v průběhu výpočtů pro každou sledovanou částici data ukládána.

Uvedeme si několik skupin údajů.^[13]

- prostorové souřadnice částic* – protože modely mohou být prostorově jednorozměrné nebo vícerozměrné, bude první skupina údajů obsahovat jednu až tři položky, například kartézské souřadnice x , y a z , nebo souřadnice v soustavě sférické. V každém případě musí skupina obsahovat minimálně jednu prostorovou souřadnici, která představuje jakousi řídicí osu (v algoritmech metody molekulární dynamiky má podobné postavení časová osa).
- rychlosti částic* – pro sledování průběhu transportu musíme mít k dispozici informace o okamžité rychlosti každé částice \bar{v} . V závislosti na rozměrech modelu může jít o složky v_x, v_y a v_z , případně v jiné soustavě vyjádřené jinak, nebo může jít také o absolutní hodnotu celkové rychlosti v spolu se složkou v_x . Datovou strukturu je nutné přizpůsobit co nejlépe řešenému problému tak, aby data byla ukládána co nejúsporněji a aby se s nimi dalo manipulovat co nejrychleji.
- další údaje* – pro řešení řady problémů uvedené skupiny údajů o částici stačí. Pro více druhů částic v souboru nás mohou zajímat i další charakteristiky těchto částic, například jejich hmotnost, náboj, stupeň excitace, atd. Tím datový soubor narůstá do objemu a je vhodné data optimalizovat tak, že zavedeme další integrální parametr a přiřazení konkrétních charakteristik se provede v programu jen jednou.

Při částicovém modelování musíme nejprve určit oblast, v níž budeme studovaný proces simulovat. Oblast bude tvořena třemi částmi, zdrojem částic, vlastní pracovní oblast, v níž se transport odehrává, a cílovou oblastí:

- zdroj částic* – představuje generátor částic, které vstupují do určené pracovní oblasti. Z hlediska datové struktury provádí zdroj naplnění příslušných datových struktur pro částice počátečními hodnotami, tj. hodnotu proměnné x nastaví na nulu, zadají se další hodnoty prostorových souřadnic y a z a počáteční energie těchto částic či jejich rychlosti.
- vlastní pracovní oblast* – část prostoru, kde dochází k interakcím částic, které procházejí látkovým prostředím (Obr. 4.1). Informace o počtu rozptylových procesů, jejich intenzitě a popis jejich vlivu na procházející částice získáme z odborné literatury nebo experimentů. Dostáváme charakterizující kvantitativně celkový průměrný vliv prostředí na částici, tj. výslednou střední volnou dráhu. Na jejím základě vytvoříme tzv. náhodnou volnou dráhu, po jejímž uražení se sledovaná částice srazí s jinou částicí a její trajektorie se zalomí. Pak z existujících rozptylových procesů vybereme typ konkrétní interakce, podle její charakteristiky změním rychlost částice \bar{v} , zopakujeme postup a tak vytvoříme celou trajektorii částice. Pokud se při srážce vytvoří další částice, zavedeme ji do datového souboru.



Obr. 4.1: Znázornění pracovní oblasti při modelování transportu částic.^[13]

- datové struktury* – zajištění v přirozeném modelu ukládání potřebných dat: všechny údaje příslušející jedné částici jsou ukládány do jednoho sloupce datové struktury, tj. počet sloupců odpovídá celkovému počtu částic. Pro každou částici do ní ukládáme několik skupin údajů, přičemž nejčastější jsou prostorové souřadnice, rychlosti částic, jejich hmotnost, náboj, stupeň excitace. Samozřejmě, že ne všechny z nich musí být použity.

d) *cílová oblast* – trajektorie částice může končit v cílové oblasti, záchytem nebo návratem do zdrojové oblasti. Obvykle zapisujeme jen počet částic, které pronikly do cílové oblasti a srovnáme je s počtem původně vypuštěných částic ze zdroje. Ztrátové procesy zbylých dvou způsobů ukončení trajektorie můžeme podle potřeby dále analyzovat.^[13]

4.3 Rozehrání náhodné volné dráhy^[13]

Rozehrání náhodné volné dráhy je základním úkolem při řešení transportního problému. Operujeme se dvěma pojmy:

- náhodná volná dráha ξ je vzdálenost, kterou urazí částice mezi dvěma sobě jdoucími interakcemi a jedná se o náhodnou veličinu,
- střední volná dráha λ je nenáhodné číslo udávající průměrnou vzdálenost mezi interakcemi a charakterizující tak prostředí, v němž transport probíhá. Střední volná dráha představuje první moment náhodné veličiny ξ a je mezi nimi obvyklý vztah:

$$\lambda = E\xi = \frac{1}{n} \cdot \sum_{i=1}^n \xi_i,$$

kde $E\xi$ je střední hodnota s ohledem na rozdělení pravděpodobností a nazýváme ji očekávaná hodnota nebo také matematické očekávání.

Předpokládáme-li, že střední volná dráha je konstantní, tj. že látkové prostředí je homogenní, a střední volná dráha nezávisí ani na dalších parametrech částice, můžeme použít pro generování jednotlivých realizací náhodných volných drah ξ_i jednoduchý vztah:

$$\xi_i = -\lambda \cdot \ln \gamma_i,$$

kde γ je rovnoměrně rozdělená náhodná veličina v intervalu $\langle 0,1 \rangle$. pokud ale není předpoklad $\lambda = konst.$ splněn, nemůžeme předchozí vztah použít.

Vztah pro rozehrání náhodné volné dráhy ξ :

Předpokládejme, že se částice pohybuje z bodu $x=0$ podél osy x a po trase se může srážet. Rozdělovací funkce náhodné volné dráhy ξ , $F(x)$, bude:

$$F(x) = P\{\xi < x\}.$$

Pravděpodobnost, že částice bude mít první srážku v intervalu $\langle x, x + \Delta x \rangle$, určíme dvěma způsoby:

- je-li Δx dostatečně malé, dostaneme s využitím distribuční funkce $F(x + \Delta x) - F(x)$ a
- tutéž pravděpodobnost můžeme vyjádřit pomocí účinného průřezu S : $[1 - F(x)] \cdot S \cdot \Delta x$, kde $1 - F(x)$ je pravděpodobnost, že částice doletí bez srážky do místa x , a $S \cdot \Delta x$ je pravděpodobnost srážky v intervalu $\langle x, x + \Delta x \rangle$.

Z obou těchto vztahů pro pravděpodobnost srážky vytvoříme rovnici a provedeme-li limitní přechod $\Delta x \rightarrow 0$, dostaneme pro rozdělovací funkci $F(x)$ vztah:

$$F(x) = 1 - \exp\left[-\int_0^x S(s) \cdot ds\right].$$

Odpovídající hustota pravděpodobnosti rozdělení náhodných volných drah ξ :

$$p(x) = S(x) \cdot \exp\left[-\int_0^x S(s) \cdot ds\right].$$

Náhodná volná dráha je spojitou náhodnou veličinou s oborem hodnot $x \in \langle 0, \infty \rangle$ a hustotou pravděpodobnosti $p(x)$. Známe-li celkový účinný průřez $S(x)$, můžeme tyto údaje využít k rozehrání konkrétních hodnot náhodných volných drah. V prostředí s konstantním celkovým

účinným průřezem S dostáváme zjednodušené výrazy pro hustotu pravděpodobnosti $p(x) = S \cdot \exp(-Sx)$ a rozdělovací funkci $F(x) = 1 - \exp(-Sx)$.

Mezi rozptylovými procesy v modelech některých fyzikálních jevů se vyskytují takové, při kterých vznikají nové částice a tak se trajektorie štěpí a studovaný model začne být mnohočásticový, to znamená, že z původní jedné stopy částice se vytvoří tzv. strom a tudíž je nutné v programu projít všechny jeho větve. Nejjednodušší, ale nejméně efektivní, je metoda, při které se simulují všechny interakce vytvářející strom a po uložení dat do paměti počítače se postupně analyzují. Zefektivnit celou proceduru metody procházející stromovou strukturou lze dvěma způsoby:

- a) Analýza po větvích – předpokládejme, že při první srážce vznikne další, druhá částice, jejíž počáteční data zapíšeme do datové struktury, a dále sledujeme první částici. Ta při další interakci vytvoří pro nás v pořadí již třetí částici, její data zapíšeme opět do datové struktury. Takto pokračujeme až do vzniku poslední částice, kterou hodláme sledovat. Jakmile původní částice opustí pracovní oblast, zpracujeme její data, tzn. že uložíme informaci o její energii nebo úhlu vylétující částice. Následně po zpracování dat této částice zrušíme část stromu příslušící této částici od posledního uzlu a smažeme i data této částice z datové struktury. Tím se vracíme k poslednímu uzlu a pokračujeme druhou větví vycházející z tohoto uzlu stromu. Pro jednoduchost předpokládáme, že se každá trajektorie při štěpení dělí na dvě, že trajektorie mezi štěpeními jsou stejně dlouhé a že vynecháváme srážky nevedoucí ke štěpení trajektorie. Budeme-li tento postup důsledně uplatňovat, postupně projdeme celou stromovou strukturou. Odstraňováním prázdných sloupců z datové struktury a přečíslováváním částic optimalizujeme práci počítače.
- b) Analýza po generacích – v tomto algoritmu se vstupující částice nezachovává, ale po první srážce se mění na částici dvě a částici tři. Tomu odpovídá vymazání údaje o první částici z datové struktury a zavedení dvou nových sloupců ve struktuře. Pak postupně sledujeme obě nově vzniklé větve a vytvoříme čtyři nové částice v těchto dvou větvích, opět s odpovídajícími změnami údajů v datové struktuře. Pokračujeme dál ve srážkách, kdy vznikají při větvení další částice, které budeme řadit do tzv. generací, tj. máme jednu částici generace I, dvě částice generace II, čtyři částice generace III, atd. Na popisovaném algoritmu je důležité to, že dříve než přejdeme k částicím další generace, zpracujeme všechny větve generace předchozí, zaniklé částice předchozích generací pravidelně vymazávat, ve struktuře uchovávat jen částice stejné generace a odstraňovat všechny prázdné sloupce, čímž uchováme rozumnou velikost datové struktury. Rozměr datové struktury bude roven počtu částic poslední generace před jejich opuštěním pracovní oblasti.

V prostředí, kde je relativně málo srážek vedoucích ke štěpení trajektorie, kdy je strom řídký a s dlouhými větvemi, se používá algoritmus analýzy po větvích. Naproti tomu algoritmus analýzy po generacích se používá v případě stromu s velkým počtem krátkých větví.

Programování modelů se štěpením trajektorie je velmi složité a náročné, takže je výhodnější se takovým modelům vyhnout. V některých případech to ale udělat nemůžeme, například při modelování vzniku laviny nabitých částic ve tvaru jiskry nebo blesku.

4.4 Příklady k řešení

1) Srážkové procesy – necht' v daném plynu dochází ke dvěma typům srážkových procesů – prvnímu s účinným průřezem 10^{-16} m^2 a k druhému s účinným průřezem $2 \times 10^{-16} \text{ m}^2$. Připravte funkci, která rozhodne o typu srážkového procesu, který nastane v okamžiku srážky.

2) Pohyb částic – zobrazte pohyb částice v kruhu o poloměru 1 cm, jestliže střední volná dráha je 1 mm, případně 5 mm. Velikost rychlosti částice se v průběhu experimentu nebude měnit. Její velikost zvolte tak, aby měl obrázek dobrou vypovídací hodnotu.

KAPITOLA 5: EULERŮV A VERLETŮV ALGORITMUS A METODA LEAP FROG^[13]

Řešení transportního problému je nejrozšířenějším problémem částicového modelování, což znamená, že studovaný jev popisujeme na základě chování souboru jeho dílčích částí. Je to metoda deterministická, při které řešíme soubor klasických pohybových rovnic.

5.1 Použití metody molekulární dynamiky

V této kapitole se budeme zabývat metodou molekulární dynamiky, která se s úspěchem používá pro velmi rozmanité spektrum problémů od studia atomů až po astronomii, což může představovat elektrony a ionty v plazmatu, atomy v plynech, hvězdy při studiu galaxií. Rozhodující je, že částice, jejichž chování pomocí pohybových rovnic popisujeme, představují relativně mikroskopickou úroveň studovaného jevu.

Obecný postup řešení modelování trajektorií částic je následující:

- Vytvoření co nejvěrnějšího modelu studovaného jevu.
- Popsání systému pomocí souboru N částic.
- Sestavení klasické pohybové rovnice pro všechny částice.
- Určení počátečních podmínek a vyřešení pohybových rovnic. Rozmezí časů probíhá v intervalu $\langle t_0, t_{\max} \rangle$. Cílem výpočtu je většinou nalezení trajektorie částice nebo polohy částice v daném čase.

V některých oblastech fyziky se metoda molekulární dynamiky používá i pro výpočet statických vlastností, například při sledování dráhy částic ve fázovém prostoru pro konečnou dobu t . Hledaná veličina bývá zpravidla vyjádřena ve formě střední hodnoty.

5.2 Pracovní oblast, výpočet silového působení

Nyní se podívejme na jednotlivé kroky při vytváření počítačového modelu. Pro praktické řešení fyzikálního problému je třeba zvolit pracovní oblast, odhadnout výsledné síly působící na jednotlivé částice souboru, určit počáteční stav částic, numericky vyřešit příslušné pohybové rovnice všech částic a případně statisticky zpracovat získané výsledné trajektorie.

Pracovní oblast

Pracovní oblast se bude lišit podle typu řešeného problému. Pro potřeby počítačového modelování je potřeba vždy zvolit určitou velikost oblasti (nelze simulovat nekonečně velkou oblast):

- Studium pohybu těles nebo částic, jejichž trajektorie leží v omezené oblasti. Pracovní oblast musí být tak velká, aby studovaná soustava byla celá umístěna v této oblasti (např. pohyb komety kolem Slunce).
- Studium chování velkého počtu částic zajímavých velký objem. Pracovní oblast musí mít měřítko podle charakteristiky systému a proto bude mnohem menší než celý objem prostoru naplněného částicemi, jelikož ji volíme jako jakýsi výřez do studované soustavy částic (např. víme, že elektrické pole dosahuje do vzdálenosti 10 cm, takže oblast musíme vymežit tak, aby tam toto pole bylo zahrnuto celé).

Tvar pracovní oblasti

Při volbě tvaru se řídíme symetrií řešeného problému.

- Jsou-li částice rozloženy pravidelně, tvar pracovní oblasti podřídíme tomuto rozložení, protože pak se zjednoduší popis procesů na hranici pracovní oblasti, např. studium pohybu elektronů v pevné látce, kdy tvar pracovní oblasti napodobuje zvětšenou elementární buňku krystalové mřížky.
- Nemá-li studovaný problém žádnou výraznou symetrii, volíme tvar pracovní oblasti co nejjednodušší, aby se zjednodušila i manipulace s daty při modelování. Například ve dvourozměrném případě budeme pracovat se čtvercem o hraně L , v trojrozměrném případě s krychlí o hraně L .

Velikost pracovní oblasti

Při volbě velikosti pracovní oblasti máme dvě kritéria: první výpočetní a druhé fyzikální.

- a) Současná výpočetní technika nám umožňuje pracovat řádově s miliony částic. Při menším počtu jsou příliš velké fluktuace a při příliš velkém počtu částic je výpočet příliš pomalý a nemusí stačit kapacita paměti počítače.
- b) Fyzikální kritérium vyplyne z charakteru silového působení v modelu. Rozeznáváme síly dalekodosahové, kdy jejich velikost se vzdáleností klesá s kvadrátem vzdálenosti (gravitační síla, elektrostatická síla, aj.), a krátkodosahové, které mají závislost na vzdálenosti mnohem silnější – r^{-7} (např. síly v pevných látkách).

Při volbě velikosti pracovní oblasti požadujeme, aby na částici umístěnou v jejím středu bylo silové působení od částic mimo pracovní oblast zanedbatelné, tj. aby na vzdálenosti $L/2$ klesly síly na nevýznamnou hodnotu. Ve většině případů nám současná aplikace obou kritérií určí interval, z kterého můžeme rozměr pracovní oblasti zvolit. Někdy však se obě kritéria navzájem vylučují a pak je nutné použít ekvivalent umělých obrátů z metody Monte Carlo (viz předcházející kapitoly).

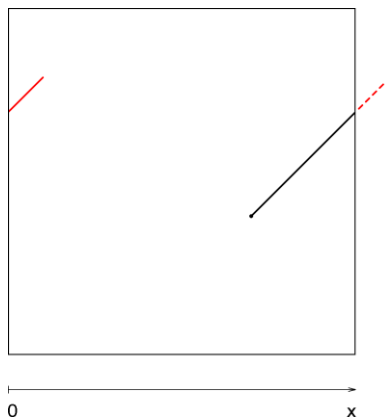
3) Volba počátečních a okrajových podmínek

Počáteční podmínka je obvykle jedna a definuje výchozí stav výpočtu na počátku řešení, to znamená v čase $t_0 = 0$, která může být získána naměřením počátečního stavu na fyzikálním modelu nebo vytvořena předchozí analýzou. Tuto podmínku můžeme obecně vyjádřit jako funkci f souřadnic x , y a z v čase t_0 , kdy se závislá proměnná u průběžně s časem mění, tedy $u = f(x, y, z)$.

Okrajové podmínky se obecně vyskytují v případech, kde jako nezávisle proměnné vystupují souřadnice. Tyto podmínky se vyskytují na hranicích tělesa a popisují základní charakteristiky tohoto tělesa. Okrajové podmínky mohou být konstantní v čase, tzn. stacionární, nebo časově proměnné, tzn. nestacionární, dále na dalších veličinách nezávislé, tzn. lineární, nebo závislé na různých veličinách, tzn. nelineární. Nesprávné zadání okrajových podmínek může vést k chybným výsledkům nebo k nestabilnímu řešení. Známe tři základní typy okrajových podmínek:

- okrajová podmínka 1. druhu (Dirichletova) – hodnota závisle proměnné v místě x_0 je známou funkcí ostatních souřadnic a času,
- okrajová podmínka 2. druhu (Neumannova) – hodnota derivace závisle proměnné podle jedné souřadnice je známou funkcí ostatních souřadnic a času,
- okrajová podmínka 3. druhu (Newtonova) – hodnota lineární kombinace hodnoty závisle proměnné u v bodě x_0 a její derivace podle x v místě x_0 je známou funkcí ostatních souřadnic a času.

V případě, že pracovní oblast je pouze výřezem z celého objemu naplněného částicemi, bude neustále docházet k vylétávání částic z pracovní oblasti a naopak vstupu částic z okolního prostoru. V modelu tento rovnovážný proces simulujeme pomocí cyklických okrajových podmínek – pokud částice pracovní oblast opustí, ihned vstoupí do pracovní oblasti zpět na odpovídajícím místě protilehlé stěny (Obr. 5.1).



Obr. 5.1: Cyklické okrajové podmínky ve směru osy x .^[13]

V kubické pracovní oblasti o hraně L každá souřadnice x , y a z každé částice musí ležet v rozmezí $\langle 0, L \rangle$. Sledujeme pouze hodnoty souřadnic a přičteme nebo odečteme hodnotu L k příslušné souřadnici, která z intervalu $\langle 0, L \rangle$ vybočí.

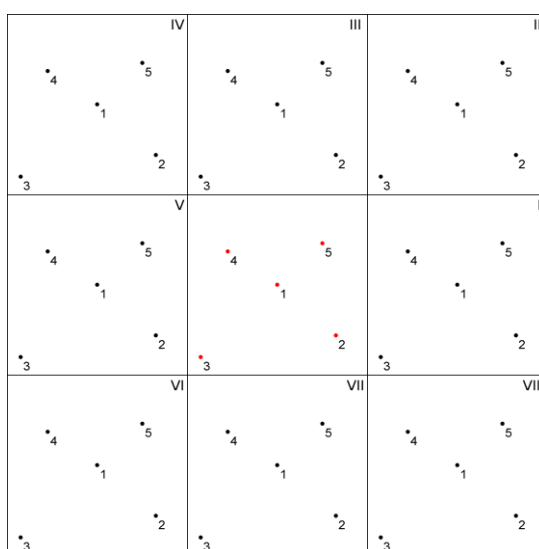
Při aplikaci cyklických okrajových podmínek klademe na studovanou soustavu částic určitá fyzikální omezení. V systému nesmí být přítomen žádný usměrněný tok částic a prostředí musí být zcela homogenní. Pokud tomu tak nebude, musíme vytvořit zdroj částic a jím kompenzovat úbytek částic opouštějících pracovní oblast.

Výpočet silového působení

Metoda molekulární dynamiky je založena na řešení pohybových rovnic. Proto musíme určit síly, které na každou částici působí. Předpokládáme, že pracovní oblast obsahuje N částic. Sílu působící na částici nacházející se poblíž pracovní oblasti najdeme jako vektorový součet silového působení od ostatních $N-1$ částic v pracovní oblasti.

Komplikovanější situace nastane pro částice ležící poblíž hranice pracovní oblasti, kde nemůžeme zanedbat silnější silové působení od blízkých částic na druhé straně hranice pracovní oblasti a je nutné jejich vliv započítat do výpočtu. To nelze udělat přímo – do modelu bychom zahrnuli vedle původních N částic i některé sousední částice a došlo by tím jen ke zvětšení pracovní oblasti, což problém neřeší.

Tento problém lze však řešit například kopírováním pracovní oblasti. Na obrázku 5.2 je pracovní oblast zobrazena bez označení ve středu a kolem ní je umístěno 8 kopií. Pracovní oblast obsahuje N částic (zde $N=5$). Kopie jsou označeny *I–VIII*, zůstávají svázané s původní pracovní oblastí, takže každý pohyb původní částice se promítne i do všech jejích obrazů.



Obr. 5.2: Obrazy pracovní oblasti pro výpočet silového působení.^[13]

Výsledné silové působení na konkrétní částici v původní pracovní oblasti se i nadále počítá jako vektorový součet působení od ostatních $N-1$ částic, přičemž se uvažují největší síly. V našem případě dostaneme:

$$\bar{F}_1 = \bar{f}_{12} + \bar{f}_{13} + \bar{f}_{14} + \bar{f}_{15},$$

kde všechny částice 2, 3, 4 a 5 vezmeme v původní pracovní oblasti.

Naproti tomu:

$$\begin{aligned}\bar{F}_2 &= \bar{f}_{21} + \bar{f}_{23} + \bar{f}_{24} + \bar{f}_{25}^{(VII)}, \\ \bar{F}_3 &= \bar{f}_{31} + \bar{f}_{32}^{(V)} + \bar{f}_{34}^{(VII)} + \bar{f}_{35}^{(VI)}, \\ \bar{F}_4 &= \bar{f}_{41} + \bar{f}_{42}^{(V)} + \bar{f}_{43}^{(III)} + \bar{f}_{45}^{(V)}, \\ \bar{F}_5 &= \bar{f}_{51} + \bar{f}_{52}^{(III)} + \bar{f}_{53}^{(II)} + \bar{f}_{54}^{(I)},\end{aligned}$$

kde horní index rozlišuje, do které oblasti částice patří, zda původní nebo některého jejího obrazu.

Při realizaci zjistíme, že každá částice nebo její obraz se v součtu popisujícím silové působení objeví právě jednou. Algoritmus hledání největších sil se tím zjednoduší. Nepracuje se přímo se silami, ale pouze se vzdálenostmi částic, neboť všechny typy sil ubývají se vzdáleností a

stačí proto najít $N-1$ nejbližších částic. Jelikož nepotřebujeme síly, nejedná se o absolutní hodnoty vzdáleností, ale jen o jejich pořadí, takže pracujeme jen s druhými mocninami vzdáleností. Všechny souřadnice všech obrazů částice se liší pouze přičtením nebo odečtením hodnoty L v rozdílech typu $x_i - x_j$, což lze v cyklu jednoduše realizovat.

5.3 Řešení pohybových rovnic

Deterministická metoda molekulární dynamiky je založena na řešení soustavy pohybových rovnic pro všechny částice typu

$$\bar{F}_i = m_i \cdot \bar{a}_i, \quad i = 1, \dots, N.$$

Diferenciální rovnice druhého řádu $\left(\bar{a} = \frac{d^2 \bar{r}}{dt^2} \right)$ převedeme na dvojnásobný počet rovnic prvního řádu (v proměnných \bar{r} a \bar{v}). Ty posléze převedeme na rovnice diferenční (spojitou časovou osu nahradíme diskrétní posloupností časů $t_0, t_1, \dots, t_{\max}$, kde $\Delta t = t_{k+1} - t_k$). Zvolíme počáteční podmínky pro polohu a rychlost všech částic v čase t_0 a dostaneme jednoduchý algoritmus pro řešení soustavy pohybových rovnic pro všech N částic.

Byla navržena celá řada algoritmů, z nichž nejjednodušší jsou metody:

5.3.1 Eulerova metoda řešení pohybových rovnic

- Počáteční podmínky: \bar{r}_i^0, \bar{v}_i^0 ,
- Přejchod z času t_0 do t_1, \dots ,
- Přejchod z času t_k do t_{k+1} ,

$$\bar{r}_i^{k+1} = \bar{r}_i^k + \bar{v}_i^k \Delta t + \frac{1}{2m_i} \bar{F}_i^k \Delta t^2,$$

$$\bar{v}_i^{k+1} = \bar{v}_i^k + \frac{1}{m_i} \bar{F}_i^k \Delta t \quad i = 1, \dots, N,$$

$$\bar{F}_i^{k+1} = \dots$$

Tato metoda je univerzální, nepřináší žádná fyzikální omezení. Její nevýhodou je, že je pouze prvního řádu přesnosti Δt .

Požadujeme-li vyšší přesnost řešení, použijeme kvalitnější algoritmy, které nabízí numerická matematika, například metody typu prediktor-korektor nebo Runge-Kuttovy metody vyšších řádů. Tyto algoritmy jsou ale pomalé, proto jsou vhodné pro menší počet částic N .

Tam, kde máme velký počet částic a rozhodující je pouze jejich chování jako celku, používáme méně přesné, ale za to rychlejší algoritmy. Nepřesnost výpočtu kompenzujeme obvyklým následným průměrováním trajektorií a tím získáme makroskopické veličiny jako je například tlak, elektrický proud, apod.

Jako vhodný kompromis se nabízí dva přibližně rovnocenné algoritmy druhého řádu přesnosti odpovídající modifikovaným Eulerovým metodám – Verletova metoda a metoda Leap-frog.

5.3.2 Verletova metoda řešení pohybových rovnic

- Počáteční podmínky: \bar{r}_i^0, \bar{v}_i^0 ,
- Přejchod z času t_0 do t_1, \dots ,
- Přejchod z času t_k do t_{k+1} ,

$$\bar{r}_i^{k+1} = \bar{r}_i^k + \bar{v}_i^k \Delta t + \frac{1}{2m_i} \bar{F}_i^k \Delta t^2,$$

$$\bar{F}_i^{k+1} = \dots \quad i = 1, \dots, N,$$

$$\bar{v}_i^{k+1} = \bar{v}_i^k + \frac{1}{m_i} (\bar{F}_i^k + \bar{F}_i^{k+1}) \cdot \Delta t.$$

Ve vztahu používáme současně starou a novou hodnotu síly \bar{F}_i^k a \bar{F}_i^{k+1} , takže potřebujeme současně dva vektory, což zvyšuje nároky na kapacitu paměti počítače.

5.3.3 Leap-frog metoda řešení pohybových rovnic

- Počáteční podmínky: $\bar{r}_i^0, \bar{v}_i^{1/2}$,
- Přejchod z času t_0 do t_1, \dots ,
- Přejchod z času t_k do t_{k+1} ,

$$\bar{r}_i^{k+1} = \bar{r}_i^k + \bar{v}_i^{k+1/2} \Delta t + \frac{1}{2m_i} \bar{F}_i^k \Delta t^2,$$

$$\bar{F}_i^{k+1} = \dots \quad i = 1, \dots, N,$$

$$\bar{v}_i^{k+3/2} = \bar{v}_i^{k+1/2} + \frac{1}{m_i} \bar{F}_i^{k+1} \cdot \Delta t.$$

Slabinou tohoto algoritmu je fakt, že pro nastartování potřebujeme počáteční hodnoty polohy a rychlosti v různých časech. Protože tyto hodnoty nebývají z experimentů k dispozici, musíme hodnotu rychlosti $\bar{v}_i^{1/2}$ dopočítat z původní hodnoty \bar{v}_i^0 .

Oba předchozí algoritmy mají oproti Eulerově metodě změněné pořadí výpočtu nových hodnot \bar{F}_i a \bar{v}_i , což přináší fyzikální omezení pro algoritmus $\bar{F}_i^{k+1} = \dots$, kde $i = 1, \dots, N$, protože jsou použitelné pouze pro řešení problémů, kde síla nezávisí na rychlosti částice, jako například pro síly gravitační povahy nebo elektrostatické, nikoli však pro úplnou Lorentzovu sílu. Pokud potřebujeme studovat pohyb nabitých částic v magnetickém poli, musíme použít buď pomalou Eulerovu metodu nebo algoritmy zvlášť vyvinuté pro tento případ.

5.4 Metoda P-I-C (Particle-In-Cell)

Metoda P-I-C je jednou z nejoblíbenějších algoritmů používaných ve fyzice plazmatu. Jde o hybridní simulace, částice se pohybují volně v prostoru v souladu s pohybovou rovnicí, pole jsou ale známá jen ve vrcholech předem dané mříže. Částice tak interaguje nikoli se všemi ostatními částicemi, ale se středním polem generovaným celým souborem částic. Označíme-li N počet částic v simulaci, sníží tento postup výpočetní náročnost z N^2 na $N \log N$. Každá částice představuje v mnoha simulacích celý shluk skutečných částic. ^[17]

Formulace problému

Analyzováním algoritmů na řešení pohybových rovnic z hlediska časové náročnosti zjistíme, že převážnou část doby program stráví při výpočtu síly \bar{F}_i . Sílu působící na i -tou částici počítáme podle vztahu:

$$\bar{F}_i = \bar{F}_i^{\text{ext}} + \sum_{\substack{j=1 \\ j \neq i}}^N \bar{f}_{ij} \quad i = 1, \dots, N,$$

kde \bar{F}_i^{ext} je síla působící na částici z externího zdroje a druhý člen rovnice představuje vzájemné silové působení souboru částic.

V případě přítomnosti pouze elektrostatického pole můžeme externí sílu vyjádřit pomocí intenzity tohoto pole:

$$\bar{F}_i^{\text{ext}} = q_i \bar{E}^{\text{ext}}(\bar{r}_i)$$

kde $\bar{E}_i^{\text{ext}}(\bar{r}_i)$ je intenzita lokálního pole v místě i -té částice \bar{r}_i a q_i je náboj této částice. Jelikož tato intenzita pole závisí jen na poloze i -té částice, bude první člen v soustavě rovnic pro výpočet celkového silového působení \bar{F}_i lineárně záviset na počtu částic N , $O(N)$. O efektivitě výpočtu rozhoduje druhý člen rovnice \bar{F}_i . Zde musíme rozlišit případ dalekodosahových a krátkodosahových sil.

Metoda Particle-In-Cell

Základní myšlenka těchto algoritmů spočívá v tom, že vzájemné silové působení ve vztahu pro sílu \bar{F}_i nahradíme ekvivalentním vztahem pro \bar{F}_i^{ext} . K tomu účelu musíme najít intenzitu lokálního elektrického nebo gravitačního pole \bar{E}^{lok} a vztah pro výsledné silové působení je:

$$\bar{F}_i(\bar{r}_i) = \bar{F}_i^{\text{ext}}(\bar{r}_i) + \bar{F}_i^{\text{lok}}(\bar{r}_i) \quad i = 1, \dots, N.$$

V této metodě navíc k obvyklé diskretizaci časové osy (s krokem Δt) provedeme ještě diskretizaci prostoru. Pracovní oblast rozdělíme na soustavu menších čtvercových buněk s hranou Δx . Původní soubor N částic se nám tak rozdělí mezi jednotlivé buňky.

Další postup výpočtu intenzity lokálního pole pro elektrostatické pole (obdobně bude probíhat i pro pole gravitační):

1. Elektrický náboj všech částic v buňce složíme do jedné výsledné částice ve středu buňky q_{ij} , kde (ij) jsou souřadnice příslušné buňky. Celkový elektrický náboj v buňce q_{ij} můžeme získat buď pomocí metody NGP (Nearest Grid Point), což je prosté sečtení všech nábojů v buňce a přenesení výsledného náboje do středu buňky, nebo pomocí metody CIC (Cloud In Cell), kde náboj není bodový, ale má neurčitý tvar, který obvykle zasahuje do více buněk úměrně svému objemu v příslušné buňce. Pro jednoduchost předpokládáme, že nábojový útvar má konstantní hodnotu a tvar stejný jako buňka.
2. K prostorové hustotě náboje ρ_{ij} od nábojů q_{ij} přejdeme vydělením objemem buňky.
3. Vyřešíme Poissonovu rovnici a dostaneme hodnoty elektrického potenciálu v každé buňce

$$U_{ij}: \Delta U = -\frac{\rho}{\epsilon_0}.$$

4. Diferenčními schémata typu $(E_{ij})_x = \frac{U_{i+1,j} - U_{i-1,j}}{2\Delta x}$ přejdeme od elektrostatického potenciálu hledané intenzitě elektrického pole $\bar{E}_{ij}^{\text{lok}}$ v jednotlivých buňkách.

Metoda P-I-C znamená z hlediska výpočtu trajektorií souboru vzájemně interagujících částic pouze dílčí algoritmus pro nalezení silového působení na i -tou částici, \bar{F}_i . Tuto sílu dosadíme do pohybových rovnic a provedeme zintegrování celé soustavy rovnic v požadovaném časovém intervalu $\langle t_0, t_{\text{max}} \rangle$.

P-I-C algoritmus je vhodný k popisu vln a nestabilit v plazmatu, přepojení magnetických indukčních čar, ohřevu plazmatu, interakce laserového paprsku s plazmatem či sledování vývoje turbulencí. ^[17]

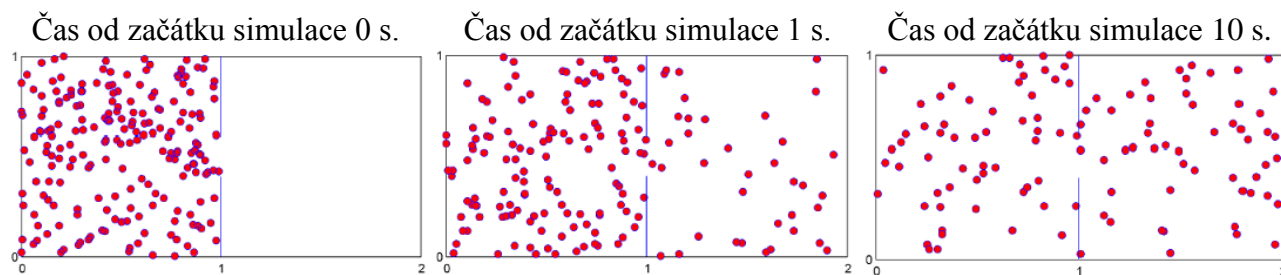
5.5 Model plynu a difúze ^[18]

1) Zadání úlohy

Zadání úlohy modelu difúze částic neutrálního plynu obsahuje algoritmus pro pohyb částic, srážkové procesy mezi molekulami, jevy na okraji pracovní oblasti atp. Při částicovém modelování hledáme trajektorie obvykle velkého množství částic pohybujících se podle základních pravidel dynamiky. Každá z těchto trajektorií je popsána soustavou diferenciálních rovnic.

Předpokládejme, že máme pracovní oblast rozdělenou na dvě stejně velké oblasti pro částice nepropustnou překážkou. V obou oblastech máme plyn o stejné koncentraci, v levé polovině této oblasti pak umístíme navíc N částic plynu, jehož difúzi do pravé části pracovní oblasti budeme sledovat. Pro jednoduchost předpokládejme, že všechny tyto částice můžeme považovat za body zanedbatelné hmotnosti a že všechny částice mají v průběhu experimentu stejnou rychlost. Dále předpokládejme, že problém řešíme jako dvoudimenzionální případ.

V čase t_0 odstraníme část překážky tak, jak je to patrné z obrázku 5.3. Spustíme výpočet a sledujeme, jak částice pronikají z levé strany pracovní oblasti do pravé, až se počty částic na obou stranách přibližně vyrovnají. Změnou vstupních parametrů (střední volná dráha částic mezi srážkami, jejich rychlost, velikost štěrbiny atp.) tak můžeme simulovat vlastnosti jevu nazývaného difúze.



Obr. 5.3: Rozložení částic v pracovní oblasti v čase 0 s, 1 s a 10 s od počátku výpočtu. ^[18]

2) Popis počítačového modelu

a) Zdrojový kód tohoto modelu bude sestávat z několika základních částí:

1. Část, v níž zadáme všechny potřebné parametry modelu a dopočítání ostatních fyzikálních veličin.
2. Rozehrání počátečních poloh, rychlostí a náhodných volných drah jednotlivých částic.
3. Eulerův algoritmus.
4. Ošetření událostí na okraji pracovní oblasti a na překážce.
5. Zavedení srážkových procesů v plynu.
6. Grafický výstup provedené simulace.

b) Zadání počátečních parametrů a dopočítání ostatních veličin:

V běžném plynu bývá koncentrace částic vysoká a rychlosti částic dosahují poměrně velkých hodnot. Pro potřeby takové simulace by bylo potřeba volit pracovní oblast velmi malých rozměrů a mít k dispozici výkonný počítač. Protože tento požadavek nebývá většinou splněn a provedení simulace s reálnými parametry tudíž nemusí přinést očekávaný efekt, je výhodnější, když počáteční parametry zvolíme „nefyzikálně“ tak, aby byly dobře pozorovatelné podstatné jevy studovaného jevu. Například volíme velikost hrany pracovní oblasti 1 m, rychlost částic $5 \text{ m}\cdot\text{s}^{-1}$, střední volnou dráhu částic mezi srážkami 0,5 m a počet sledovaných částic 200.

c) Rozehrání počátečních poloh, rychlostí a náhodných volných drah jednotlivých částic:

Počáteční polohy částic jsou určeny nagenováním bodu ve čtverci pomocí funkce *rand*. Směr rychlosti v polárních souřadnicích můžeme zadat nagenováním náhodného úhlu γ nebo lze také využít postupu, kdy ve čtverci se středem v bodě $[0, 0]$ a hranami o délce 2 nagenovujeme bod, otestujeme, zda se tento bod nachází uvnitř kružnice o poloměru 1 a posléze normujeme souřadnice x a y vzdáleností tohoto bodu od středu r . Takto získané složky poté využijeme k rozehrání náhodného směru vektoru rychlosti.

d) Předpis pro rozehrávání náhodné volné dráhy získáme ze vzorce pro pravděpodobnost Poissonova jevu a lze psát:

$$\lambda_i = -\lambda_{\text{str}} \ln \gamma,$$

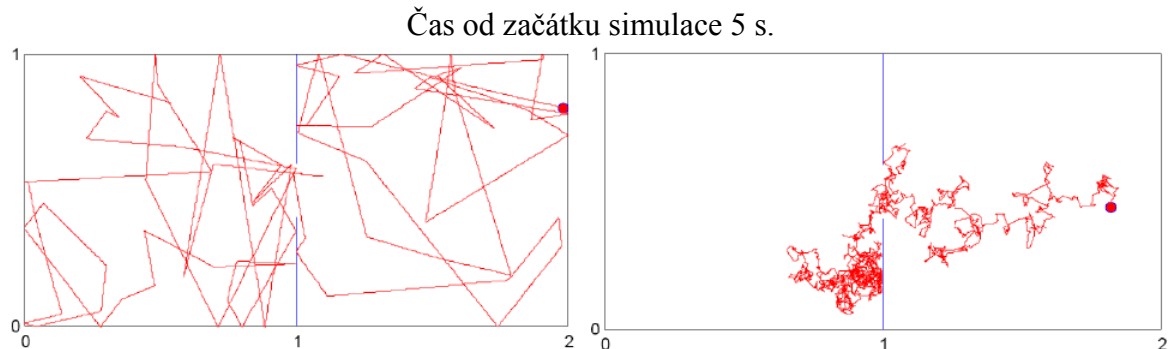
kde λ_i je střední volná dráha i -té částice, λ_{str} střední volná dráha daného typu částic a γ je náhodné číslo s rovnoměrným rozdělením na intervalu $(0,1)$.

e) Pohyb částic:

Pohyb jednotlivých částic uvnitř pracovní oblasti lze realizovat několika postupy numerické matematiky (Obr. 5.4). Nejjednodušším postupem je tzv. Eulerův algoritmus, který diskretizuje rovnici pro polohu a rychlost částice takto:

$$\begin{aligned} \bar{r} &= \bar{r}_0 + \bar{v}t + \frac{1}{2}\bar{a}t^2 & \bar{r}^{k+1} &= \bar{r}^k + \bar{v}^k \Delta t + \frac{1}{2m} \bar{F}^k \Delta t^2 \\ \bar{v} &= \bar{v}_0 + \bar{a}t & \bar{v}^{k+1} &= \bar{v}^k + \frac{\bar{F}^k}{m} \Delta t \\ \bar{F} &= \dots & \bar{F}^{k+1} &= \dots \end{aligned}$$

Pohyb každé částice je nutné sledovat zvlášť s daným časovým krokem Δt .



Obr 5.4: Trajektorie jedné částice v případě střední volné dráhy 0,5 metru (levý obrázek) a 0,01 metru (pravý obrázek).^[18]

Dále je v modelu nutné ošetřit chování částice na okraji pracovní oblasti. V našem případě předpokládáme, že okraj pracovní oblasti představuje pro částici nepřekonatelnou překážku (stěnu, od které se částice odrazí). V tomto případě tedy postačí pouze vyhledat ty částice, které v daném časovém kroku opustily pracovní oblast, a změnit znaménko složky rychlosti, která je kolmá ke stěně na opačné. Při vhodně zvoleném časovém kroku lze zanedbat vzdálenost, po kterou se částice pohybovala mimo pracovní oblast.

f) Srážkové procesy:

V plynu dochází k neustálým srážkám mezi částicemi. V našem modelu realizujeme srážky tak, že na počátku výpočtu přidělíme každé částici náhodnou volnou dráhu, od níž v průběhu výpočtu postupně odečítáme uraženou vzdálenost. V okamžiku, kdy daná částice urazí celou náhodnou volnou dráhu, realizujeme srážku, která spočívá v nagenování nového náhodného směru, a zároveň částici přidělíme novou hodnotu náhodné volné dráhy. Právě tento parametr charakterizuje rychlost šíření částic do volného prostoru. Provedeme-li simulaci pro různé parametry, umožníme vytvoření názorné představy o faktorech ovlivňujících rychlost difúze částic.

g) Grafický výstup:

Program MATLAB je velmi dobře známý svým kvalitním grafickým výstupem. Oproti jiným programovacím jazykům umožňuje vytvoření obrázku zadáním jednoho příkazu. Jelikož náš model je nastaven tak, aby průběžně vykresloval pohyb částic v pracovní oblasti, je nejprve vykreslen graf z počátečních hodnot polohy a následně jsou tyto hodnoty aktualizovány příkazem *set*. Překreslení obrázku je vynuceno příkazem *drawnow*.

5.5 Příklady k řešení

Odhadněte charakteristickou velikost pracovní oblasti pro následující systémy:

- pohyb Země kolem Slunce,
- pohyb elektronu kolem jádra atomu,
- studium vlivu přílivu a odlivu na erozi pobřežních oblastí,
- elektrické nabíjení povrchu kosmické stanice ve vesmíru,
- studium jaderných zařízení užívaných pro výrobu elektrické energie,
- studium kvantových systémů,
- studium pohybu hvězd v galaxii.

V pracovní oblasti tvaru čtverce o straně délky L náhodně rozmístěte 10000 částic. Poté naprogramujte funkci, která umožní aproximovat velikost síly působící na částici pomocí algoritmu Particle-in-Cell (PIC).

V MATLABu připravte funkce, které umožní vypočítat novou polohu soustavy částic po provedení jednoho časového kroku. K výpočtu použijte Eulerův algoritmus, Verletův algoritmus a algoritmus Leap-Frog. Algoritmy použijte ke studiu pohybu planety Země kolem Slunce, Jupiteru kolem Slunce, Neptunu kolem Slunce. Úlohu řešte pro různě velké časové kroky a sledujte, jak se projevuje nefyzikální ohřev.

BLOK 2

KAPITOLA 6: CAD SYSTÉMY

CAD, z angličtiny *computer-aided design*, česky *počítačem podporované projektování*, nebo míněno na obecný CAD systém jako *computer-aided drafting – počítačem podporované kreslení*. Jde o velkou oblast IT, která zastřešuje širokou činnost navrhování.

CAD aplikace vždy obsahují grafické, geometrické, matematické a inženýrské nástroje pro kreslení plošných výkresů a modelování objektů a dějů reálného světa. Pokročilejší řeší výpočty, analýzy a řízení systémů (výroby, zařízení). Blízkým příbuzným je také oblast počítačových vizualizací, protože virtuální 3D návrhy jsou často klientům prezentovány ve formě fotorealistických vizualizací.

6.1 Požadavky na CAD systémy^[19]

CAD systémy jsou nástroje pro modelování, animaci a renderování, ve kterých je možné vytvářet interaktivní prezentace, průchozí vizualizace např. interiérů domů a počítačové hry, vše přímo pomocí interního grafického editoru s možností doplnění kódem v některém objektově orientovaném programovacím jazyce.

Interface

Interface je neuvěřitelně efektivní, intuitivní a umožňuje tvořit modely a animace rychle, přirozeně a bez nutnosti intenzivně přemýšlet, kde najít jakou funkci skrytou v několikátém submenu.

Hlavními atributy interface jsou:

- plně přenastavitelná pracovní plocha,
- rozdělení do oken pro modelování, animační křivky, outliner, nelineární videostřih, editování UV map, animování postav (pose editor, NLA editor), souborový manažer, atd.,
- databázový systém umožňující optimální management scény, instance a dynamické propojování projektů v různých souborech,
- lokalizace do několika jazyků, včetně možnosti zapnout co vše má být lokalizováno a co ponecháno v angličtině (např. tlačítka anglicky, vysvětlující popisek v jiném jazyce) a možnosti vytvářet si vlastní jazykové sady,
- zabudovaný textový editor sloužící k poznámkám a programování skriptů,
- interface je stejný na všech platformách.

Modelování

- práce s polygony, plochami, křivkami, metabally, vektorovými fonty (TrueType, PostScript, OpenType),
- povrchy s editovatelnou ostrotí a oblastí hran,
- editování polygonálního meshe s volitelnou selekcí vertexů, hran nebo faců,
- boolens operace pro mesh,
- editovací funkce jako extrude, bevel, cut, spin, screw, warp, subdivide, noise, smoots, ...,
- možnost doprogramovat si modelační nástroje dle potřeby.

Animace

- deformační armatury (kosti - skeletony) s dopřednou i inverzní kinematikou (FK, IK), autoskinning a interaktivní nastavování vah deformačních skupin,
- několik typů constraints pro rigging,
- pose editor,

- editor nelineární animace (NLA) , automatizace posunu postavy ze zacyklenou animací chůze (walkcycle) podél definované cesty (path),
- animace vertex keys a relative vertex keys (obdoba morph targets) s ovládacími posuvníky,
- particle efekty s deformátory podle větru, gravitace, magnetickou přitažlivostí či odpuzováním a detekcí kolizí,
- SoftBodies (např. simulace látek) s detekcí kolizí,
- animovatelná deformace lattice,
- podpora "motion curve" i tradičního key-frame editování,
- podpora zvuku a nástrojů pro synchronizaci zvuku a obrazu,
- možnost doprogramovat si animační nástroje případně „řízené animace“ dle potřeby.

Render

- možnost výběru z renderovacích enginů,
- oversampling, motion blur, postprodukční efekty (glow, zblur...) fields, nečtvercové pixely,
- environment mapy, halo, lens flare, mlha...,
- několik materiálových shaderů pro difusní a specularitní kanál,
- rendering pro efekt vytažených okrajů,
- procedurální textury,
- množství exportních skriptů do dalších raytracerů,
- UV editor s několika metodami pro unwrap, atd.

Interaktivní aplikace - realtime 3D/tvorba her

- grafický editor pro naprogramování logiky aplikace bez nutnosti programovat,
- detekce kolizí a simulace dynamiky,
- přístup do enginu přes skripty pro složitější logiku, umělou inteligenci apod.,
- podpora všech povrchových módů včetně průhlednosti, animovatelných reflexních map, apod.
- přehrávání her a interaktivních 3D aplikací bez kompilování a předpočítání,
- audio,
- multi-layering scén pro plovoucí interface, aj.

Soubory a podporované formáty

- všechna data projektu se ukládají do jediného souboru,
- formát, který podporuje kompresi, digitální podpisy, zakódování, dopřednou i zpětnou kompatibilitu a může být použit jako knihovna, do níž přistupujete z jiného souboru,
- umožňuje export/import do/z co možná největšího počtu formátů,
- vytvoření samospustitelných souborů (.exe) s interaktivními 3D aplikacemi, hrami apod., nebo přehrávání ve webovském prohlížeči s příslušným pluginem,
- podpora co největšího množství platform (Windows, Linux, Mac OS, atd.).

6.2 Rozdělení CAD systémů^[20]

a) Obecné:

- 2D nebo 3D, objemové a povrchové.
 - AutoCAD (Autodesk)
 - ArchiCAD (Graphisoft)
 - Allplan (Nemetschek)
 - IntelliCAD

- Microstation (Bentley)
- ProgeCAD
- SketchUp (Google)
- Spirit
- TurboCAD
- ZWCAD

b) Specializované, určené pro:

- strojírenství:
 - CAM (Computer Aided Design): při znalosti konkrétního soustruhu a technologie obrábění je tento software schopen navrhnout dráhy nástroje při soustružení, popřípadě další aspekty výroby, v ideálním případě tak, aby současně byla výroba co nejefektivnější, tedy měla co nejmenší energetický a materiální vstup a průmyslový odpad při co nejvyšší produkci (NC programy, Rapid prototyping).
 - CAE (Computer Aided Engineering): slouží pro podporu analýzy, simulace, návrhu, výroby, plánování, diagnostiky a oprav, umožňuje rychlejší návrh, optimalizaci, zajištění robustnosti. Elektro CAE systém s technickou podporou poskytuje svým uživatelům tzv. katalogové přístrojové databáze. Každý elektrotechnický přístroj pro CAE systémy je tvořen objektem a technickým – datovým záznamem.
 - AutoCAD: – výkonný flexibilní software pro 2D a 3D projektování, konstruování, modelování a tvorbu dokumentace pro projektanty a konstruktéry zabývajícími se návrhy v oblasti strojírenských součástí a konstrukcí, stavební projekcí a architekturou, mapováním a terénními úpravami, atd. Umožňuje řízení projektů od fáze konceptu až po jeho dokončení pomocí výkonných nástrojů pro dokumentaci díky nástrojům pro automatizaci, správu a úpravy. Existuje v mnoha modifikacích, každá z nich je určena pro konkrétní druh práce. Základní verze obsahuje jen velmi omezený počet knihoven, nepodporuje některé pokročilejší funkce, nemá integrované žádné programovací rozhraní a neobsahuje aplikační nadsavby. Pro požadované plněhodnotné použití je potřeba zakoupit k základní verzi speciální knihovny a programové nadstavby.^[21]
 - CADKEY – je označován jako hybridní modelář umožňující provádět strojní konstruktérské, designérské či analytické práce, obsahuje velmi dobré funkce pro tvorbu klasických výkresů a jejich tisk, nabízí prostředí pro řešení geometrických úloh ve 3D prostoru, funkce pro vyhodnocení hmotových vlastností drátové geometrie, stínované zobrazení, správu vytvářených sestav, tvorbu vlastních maker a nadstaveb, obsahuje knihovny i specifických a speciálních strojních součástí (např. ozubená a řetězová kola) a pracuje s ČSN normami.^[22]
 - SurfCam: je jedním z nejrozšířenějších CAM systémů, který se používá pro řízení CNC technologií. Dovoluje naprogramování řízení frézek (2 až 5 os), soustruhů, vyvrtávaček a dalších obráběcích strojů. Umožňuje přípravu všech operací od hrubovacího přes dokončovací až ke zbytkovému obrábění, ověřuje vygenerovanou dráhu nástroje a porovná vyrobený modul s původním počítačově vytvořeným modelem. Je schopen velmi kvalitně exportovat i importovat datové formáty z jiných různých CAD/CAM systémů.^[22]
 - SolidWorks: plně parametrický plošný a objemový modelář je vynikajícím CAD řešením v oblasti strojírenství a průmyslového designu. Disponuje exportem i importem z mnoha datových formátů, umožňuje práci s rozsáhlými sestavami a automaticky generuje výkresovou dokumentaci. Pro tvorbu modelu je použit 3D prostor, problém technických pohledů a řezů systém řeší za uživatele. Ten se může pak zabývat jen přemýšlením o funkci, vzhledu a vlastnostech navrhovaného výrobku.^[22]

- CATIA, ELCAD, RUPLAN, AUCOPLAN, SPAC a nejnovější platformy Engineering Base), Space Claim, Alibre design, Inventor, NX, Pro/Engineer, Solid Edge, SolidWorks, T-Flex, VISI CAD, aj.
- stavebnictví a architektura:
 - AEC (Architecture Engineering Construction),
 - BIM (Building Information Model): objektově orientovaný 3D projekční software. Umožňuje použití všech běžných pracovních postupů, od jednoduchých 2D kreseb až po vytvoření virtuálního modelu budovy.
 - CAAD (Computer-aided architectural design),
 - ArchiCAD (Graphisoft), Allplan (Nemetschek), ArCon, AutoCAD Architecture (Autodesk), Bricscad, Revit Architecture (Autodesk), Speedikon (IEZ).
- potrubní systémy a technické zařízení budov:
 - Allplan (Nemetschek), Allklima, AutoCAD MEP (Autodesk), Revit MEP (Autodesk), AutoCAD Plant 3D (Autodesk).
- liniové a dopravní stavby:
 - AutoCAD Civil 3D, InRoad, InRail, RoadPAC.
- správa nemovitostí:
 - FM (Facility Management),
 - Allfa (Nemetschek), AutoCAD Map 3D (Topobase).
- elektrotechnika:
 - PCB (printed circuit boards),
 - EDA (Electronic design automation),
 - OrCAD, ProfiCAD, Racal/Redac, EAGLE.
- územní plánování a geografie:
 - GIS (geographic information systems),
 - Arcinfo, AutoCAD Map 3D (Autodesk), Mapinfo, Allplan (Nemetschek).

KAPITOLA 7: SOLIDWORKS – 3D CAD SYSTÉM

Je jedním z nejpoužívanějších profesionálních strojírenských 3D CAD systémů. K dispozici jsou vynikající nástroje pro modelování dílů a sestav, tvorbu výrobní dokumentace, kontrolu správnosti návrhu, nástroje pro správu dat, vyhodnocení dopadu na životní prostředí i možnost provádění pevnostních analýz, analýz přenosu tepla, proudění a tolerancí. Nabízí výkonné objemové i plošné modelování, vertikální nástroje pro plechové díly, práci s neomezeně rozsáhlými sestavami a automatické generování výrobních výkresů. Přesnou a efektivní práci zajišťuje nejširší paleta základních, pokročilých i specializovaných nástrojů, obsahuje speciální technologii pro práci s velkými sestavami a generování rozsáhlých výkresů. Samozřejmostí jsou i rozsáhlé možnosti pro využití existující dokumentace jako převod ze 2D výkresů do 3D modelů nebo převodníky různých druhů přenosových formátů.^[23]

7.1 SolidWorks

a) Uživatelské rozhraní

Uživatelské rozhraní je velmi příjemné, logicky uspořádané a vizuálně přehledné pracovní prostředí, které inovativně kombinuje moderní ovládací prvky (jako je CommandManager neboli pás karet, který logicky seskupuje podobné nástroje) s osvědčeným systémem roletových menu. Prostředí je zcela uživatelsky přizpůsobitelné, vlastní nastavení umožňuje nakonfigurovat klávesové zkratky, gesta myši, panel zástupců či veškeré ikony. Pro snazší orientaci a zrychlení práce slouží strom prvků (PropertyManager), podokno úloh či kontextové nástroje. Ovládání se založené na technologii SWIFT, která snižuje potřebu takových manuálních zásahů a operací, při kterých si uživatel není dopředu jistý správným postupem, významně šetří čas a umožňuje věnovat se samotnému procesu navrhování a ne ovládání systému. SWIFT automaticky provádí diagnózu problémů spojených s pořadím prvků, vazbami skic, použitím kót, atd.^[23]

b) Zobrazení modelů technologií *RealView*

Tato technologie ovlivňuje zásadním způsobem kvalitu zobrazení modelů v SolidWorks. Umožňuje v reálném čase zobrazovat realistické, stínované a plně dynamické modely dílů či sestav. Pro použití RealView se nic nenastavuje, stačí nástroj jen zapnout a na model použít některý z materiálů SolidWorks. RealView používá společnou databázi vzhledů a prostředí, které reprezentují použitý materiál a okolní scénu, a pro snazší přístup jsou umístěny v podokně úloh.^[23]

c) Konstrukční systém

Jedním z požadavků na moderní konstrukce je maximální úspora materiálů a energií při zachování plné funkčnosti výrobku. Dimenzování namáhaných částí, pohonů a zdrojů tepla lze s použitím analytických nástrojů snadno ověřit a optimalizovat. Výhodou těchto nástrojů je i možnost ověřit v krátké době několik možných řešení. Díky jedinečným analytickým nástrojům systému SolidWorks můžete znát vlastnosti výrobků i bez výroby prototypů a následného testování. Nástroj *ToolAnalyst* dovoluje ověřit tolerance na celé sestavě a vyhledat místa, kde jsou tolerance příliš nízké nebo naopak příliš vysoké. K dispozici jsou rozsáhlé možnosti pro úpravu uživatelského rozhraní i nastavení vlastních stylů výkresů přesně dle uživatelských požadavků či norem. Dodržení nastavených stylů lze kontrolovat pomocí nástroje *Design Checker*. Nástroj *3DVia Composer* umožňuje snadnou tvorbu technických ilustrací včetně pohyblivých prezentací. K typickým představitelům technické ilustrace patří obrázkové montážní návody, zobrazení jednotlivých technologických operací, rozhozené pohledy sestav či znázornění funkčnosti zařízení.^[24]

d) Skicář

Inteligentní skicář obsahuje množství užitečných nástrojů od základní tvorby po simulaci

mechanismů. Inteligentní nástroj *SketchXpert* a funkce kontrola skici pro vybraný prvek pomáhají nalézt a vyřešit případné chyby ve skicích, jako jsou například duplicitní entity či nespojitě obrysy. Do skic lze také vkládat importovanou geometrii a využít tak starší 2D dokumentaci k tvorbě nových modelů.^[24]

e) Tvorba součástí

Při tvorbě dílů je možné kombinovat výhody objemového a plošného modelování včetně použití prvků z knihoven či inteligentních součástí (Obr. 7.1). Tvorba ve 3D prostředí poskytuje uživateli jasnou představu o tvaru a velikosti dílu. Pro minimalizaci možných chyb v návrhu slouží nástroje pro kontrolu a analýzu návrhu. Mezi tyto nástroje patří analýza průniků součástí, detekce kolizí, kontrola sousostí děr či podřezání. Návrh odlitek a vstřikovaných dílů usnadní funkce pro nalezení dělicích křivek a siluet, inteligentní analýza úkosů i analýza síly stěn. Díky sadě specializovaných nástrojů pro formy a ve spojení s vynikajícími nástroji pro tvarové modelování je systém SolidWorks mimořádně vhodný pro tvorbu vložek forem či částí modelových zařízení i pro efektivní práci s válcovanými profily. Tyto profily je možné snadno vkládat na jednotlivé entity skic a řídit spojení v rozích včetně mezer pro svary.^[24]



Obr. 7.1: Ukázka součástky.^[24]

f) Sestavy

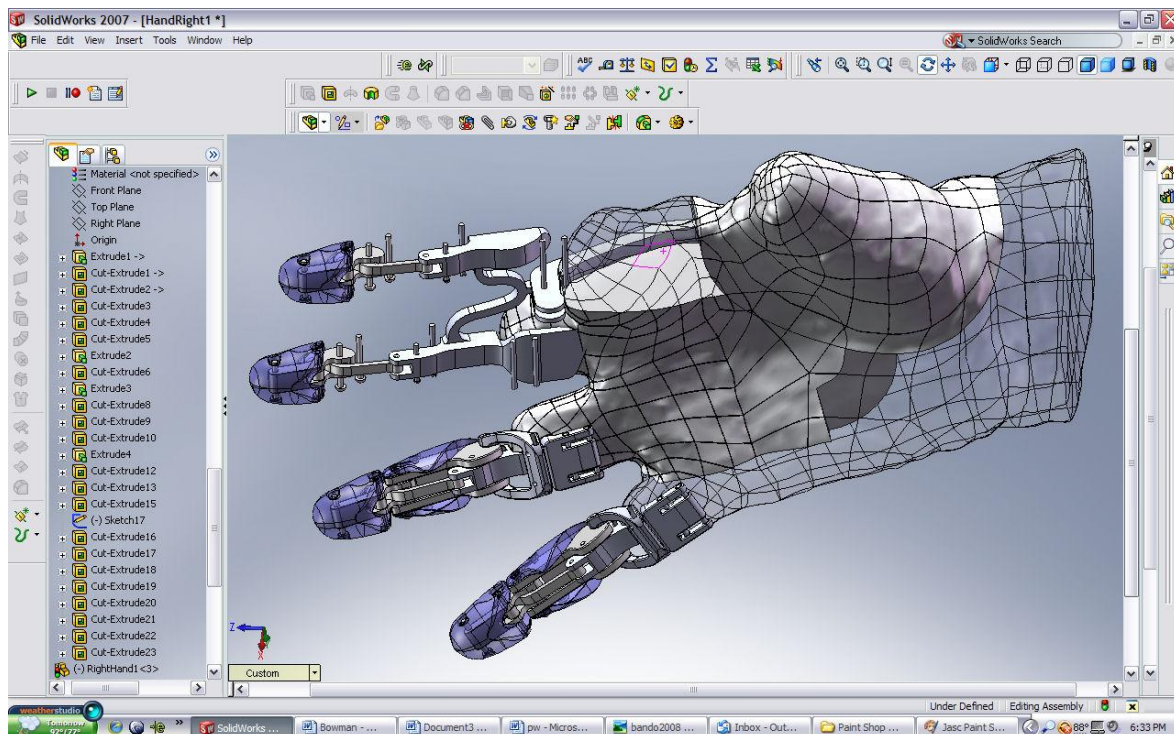
V systému SolidWorks lze vkládat do sestav součásti a definovat jejich polohu pomocí vazeb. Součásti, které nemají pomocí vazeb definované všechny stupně volnosti, je možné tažením myši přemístit a jednoduše tak ověřit pohyb mechanismů. Postupným přidáváním standardních i speciálních vazeb do sestavy se omezuje pohyb součásti v rámci sestavy a tím vlastně definuje kinematiku. Součásti, které tvoří logické celky, můžeme seskupovat do podsestav. Seznam dílů a podsestav, vazeb, prvků a referenční geometrie se nachází ve stromě, jehož obsah se snadno prohledává a filtruje. Pro větší přehlednost se také můžou libovolné součásti skrývat, zprůhledňovat, vizualizovat pomocí stavů zobrazení nebo filtrovat pokročilými výběrovými pravidly. Pro detailní analýzy chování pohyblivých sestav slouží nástroj *SolidWorks Motion*.^[24]

g) Knihovny návrhů

Pro snazší dostupnost často používaných dílů, prvků či popisů lze vybrané objekty uložit do knihovny návrhů. Při opětovném použití stačí objekt pouze přetáhnout do aktuálního dokumentu. Knihovna návrhů je součástí uživatelského prostředí a nachází se po pravé straně v podokně úloh. Knihovny normalizovaných součástí obsahují kromě standardních typů spojovacího materiálu i ozubená kola, hřebenové tyče a válcované profily.^[24]

h) Výkresová dokumentace

Výkresová dokumentace je založena na využití již existujících modelů součástí. Na listy výkresů je možné jednoduchým přetažením z palety pohledů vložit základní pohledy a následně definovat vlastnosti pohledů, jako jsou zobrazení skrytých a tečných hran či vyjmutí součástí z řezů. Automatické vyplnění rohových razítek a kusovníků či vložení kót a dalších typů popisů významně přispívá k produktivitě tvorby výkresů a omezení vzniku možných chyb. Všechny součásti návrhu, tedy modely dílů, sestav a jejich výkresy (Obr. 7.2), jsou vzájemně propojeny, je tak zajištěno, že dokumentace je vždy aktuální.^[24]



Obr. 7.2: Ukázka 3D zobrazení^[25]

7.2 Produktová řada SolidWorks nabízí následující balíčky a aplikace:^[23]

SolidWorks Standard – nabízí ucelené a robustní řešení pro 3D CAD navrhování, včetně modelování plechových dílů a forem, vytváření sestav, mechanismů a aktuální kompletní výkresové dokumentace. K dispozici je celá řada nástrojů pro ověřování návrhů.

SolidWorks Professional - rozšiřuje balíček Standard o výkonné nástroje pro zvýšení produktivity a zlepšení komunikace. To zahrnuje i PDM systém umožňující správu dat.

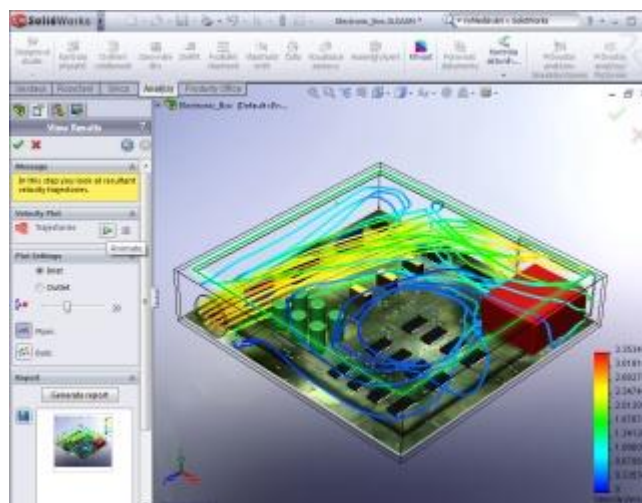
SolidWorks Premium – přidává nástroje pro pokročilé a přesné strukturální a pohybové simulace. Obsaženy jsou také nástroje pro navrhování potrubních systémů a kabelových svazků.

SolidWorks Sustainability – rozšiřuje produktovou řadu SolidWorks o nástroje pro podporu ekologického navrhování.

7.3 FloXpress^[23]

1) Simulační nástroj FloXpress

FloXpress je simulační nástroj určený pro základní analýzu proudění vzduchu nebo vody. S jeho pomocí se rychle získávají informace o proudění v uzavřeném prostoru modelu, který obsahuje jeden vstupní a jeden výstupní otvor (Obr. 7.3).



Obr. 7.3: Ukázka obrazovky simulující proudění^[23]

FloXpress je součástí každé instalace SolidWorks, do jehož prostředí je stejně jako plnohodnotný Flow Simulation beze zbytku integrován.

Veškeré zadávání a zpracování dat probíhá v přehledném průvodci, s jehož pomocí můžeme snadno a rychle zadat tekutinu, parametry simulace, vstupní a výstupní otvor. Díky ukázkám, příkladům a vysvětlujícímu textu průvodce usnadňuje správné pochopení celého procesu simulace.

Jakmile jsou splněny vstupní podmínky geometrie modelu a zadána všechna potřebná data, FloXpress umožní spuštění simulace. Po jejím skončení si můžeme prohlédnout trajektorie proudění (proudnice) mezi vstupním a výstupním otvorem. Na základě získaných výpočtů můžeme v daném návrhu nalézt a opravit problematické oblasti.

Vstupní a výstupní data^[23]

Při zadávání potřebných dat volíme mezi vodou a vzduchem, určíme vstupní a výstupní otvor, vybíráme metriku pro výpočet proudění – tlak, objemový nebo hmotnostní průtok, můžeme změnit směr proudění – záměna vstupu a výstupu, volíme animaci proudění tekutin, znázornění proudnic pomocí kuliček nebo křivek, automatickou generaci reportů a zachycení obrazovky.

2) Flow Simulation – Analýza proudění^[24]

V SolidWorks Flow Simulation lze jednoduše simulovat proudění kapalin a plynů. Obsahuje výkonný nástroj na výpočet proudění tekutin (CFD – computational fluid dynamics), který umožňuje rychle a jednoduše simulovat proudění tekutiny, sdílení tepla a výpočet tlaku v kapalině a další jevy, které mohou zásadně ovlivnit návrh výrobku.

- Dokáže zkoumat proudění uvnitř i okolo zařízení nebo obě možnosti zároveň pomocí kombinace interního a externího proudění. Spojí proudění s tepelnou analýzou, simulující sdílení tepla prouděním, vedením a zářením, nalezne nejlepší rozměry nebo podmínky přívodu a vývodu, které splní cíle návrhu, jako je výkon, pokles tlaku nebo rychlost. Zahrne do výpočtu složité jevy jako jsou pórovitost materiálů, kavitace, vlhkost,... Vyřeší problémy proudění neneutronovských tekutin, jako je krev, olej a plast. Simuluje rotaci oběžných kol a ventilátorů za použití rotující oblasti. Simuluje změny teplot použitím povrchových nebo objemových tepelných zdrojů a zohledňuje sluneční záření. Sleduje chování částic unášených v toku kapalin a plynů. Použije časově a místně závislé okrajové podmínky a tepelné zdroje.
- Využívá schémata řezů pro prozkoumání rozložení výsledných množství, včetně rychlosti, tlaku, vířivosti, teploty a hmotnostního podílu.
- Dokáže měřit výsledky v jakémkoliv místě pomocí čidel.
- Provádí grafické znázornění různých výsledků na jakémkoliv skice SolidWorksu.
- Sestaví seznam výsledků a automaticky exportuje data do aplikace Microsoft Excel.
- Prozkoumá trajektorii toků uvnitř nebo kolem modelu animovanými pruhy, šipkami 3D, trubkami nebo kuličkami.

3) Electronic Cooling a HVAC

Electronic Cooling vyhodnocuje teplotní vlastnosti komponentů a požadavky na chlazení. Obsahuje nástroje pro simulaci ohřívání v joulech, kompaktní modul se dvěma tepelnými odpory, modul tepelné trubice a PCB generátor.

Modul HVAC vyhodnocuje pohyb vzduchu a plynu v pracovním a životním prostředí. Zahrnuje upřesňující modelování záření, dovolené parametry a rozsáhlou databázi stavebních materiálů.

8.1 Metoda konečných prvků

Metoda konečných prvků (Finite Element Method – FEM) je numerická metoda pro řešení soustav diferenciálních rovnic. Touto metodou je zpravidla možné řešit problémy, které nelze úspěšně řešit klasickými postupy. Metoda pokrývá celou šíři fyzikálních aplikací: statika, dynamika, akustika, teplo, elektromagnetické pole, elektrostatika, piezoelektrické jevy a proudění, sloužící k simulaci průběhů napětí, deformací, vlastních frekvencí, proudění tepla, jevů elektromagnetismu, proudění tekutin, atd. na vytvořeném fyzikálním modelu.^[26] Pro svoji efektivnost a univerzálnost je v posledních letech velice oblíbená a je základním prostředkem při řešení problému v řadě komerčních softwarových produktů.

Myšlenka diskretizace kontinua do velkého množství konečných prostorových nebo i plošných prvků vhodného tvaru s parametry určenými v jednotlivých uzlových bodech, přinesla možnost nalezení řešitelného matematického modelu s přijatelnou chybou. Postup řešení je založen jednak na úplné eliminaci diferenciální rovnice (stacionární úlohy), nebo na převedení parciálních diferenciálních rovnic na ekvivalentní obyčejnou diferenciální rovnici, jež je následně řešena standardními postupy.^[27]

Existuje celá řada možných postupů, všechny s určitými výhodami i nevýhodami. Metoda konečných prvků je rozumnou volbou pro řešení parciálních diferenciálních rovnic na složitých oblastech, jakými jsou například automobily či potrubní rozvody, nebo v případě, kdy požadovaná přesnost se mění po celé oblasti řešení. Například při simulaci místního počasí na Zemi je mnohem důležitější dosáhnout přesné predikce nad zemí než nad oceánem, to je požadavek, který je právě dosažitelný metodou konečných prvků.^[28]

Pro simulace pomocí metody konečných prvků je možné využít řady počítačových programů – např. COMSOL Multiphysics, Ansys, Cosmos, MSC/Marc, MCS/Nastran, a dalších.

8.1.1 Diferenciální rovnice – definice a základní pojmy^[29]

Abychom si mohli vysvětlit princip metody konečných prvků, musíme si zopakovat základní pravidla řešení parciálních diferenciálních rovnic. Tyto rovnice nám pomáhají velmi účinně zvládnout složité fyzikální a technické úlohy.

a) Řešení každého problému se skládá ze tří základních kroků:

1. sestavení rovnice,
2. řešení rovnice,
3. interpretace a využití výsledků.

b) Diferenciální rovnice dělíme podle vybraného kritérií na:

- obyčejné a diferenciální
- eliptické, parabolické a hyperbolické,
- stacionární (bezčasové) a evoluční (časové),
- lineární a nelineární, apod.

c) **Obyčejná diferenciální rovnice** je taková rovnice, ve které se vyskytuje neznámá funkce jedné proměnné a její derivace. Úkolem je najít všechna řešení rovnice nebo najít řešení, které splňuje nějaké doplňující (počáteční nebo okrajové) podmínky.

Příklad obyčejné diferenciální rovnice: $\frac{ds}{dt} = v(t)$

d) Parciální diferenciální rovnice je taková rovnice, ve které se vyskytuje neznámá funkce více proměnných a její parciální derivace.

Příklad parciální diferenciální rovnice *prvního* řádu funkce u proměnných x, y :

$$\frac{\partial u}{\partial x} + \frac{\partial u}{\partial y} = 0.$$

Typickým příkladem rovnice prvního řádu je rovnice zákona zachování hmoty, která říká, že celkové množství hmoty je konstantní.

Příklad parciální diferenciální rovnice *druhého* řádu funkce u proměnných x, y :

$$\frac{\partial u}{\partial x} \frac{\partial u}{\partial y} + \frac{\partial^2 u}{\partial x \partial y} + u^2 = 0.$$

Jako příklad rovnice druhého řádu můžeme uvést Laplaceovu rovnici, rovnici pro kmity struny nebo rovnici pro vedení tepla. Všechny uvedené parciální diferenciální rovnice jsou lineární.

Každou parciální diferenciální rovnici druhého řádu můžeme převést na tzv. *kanonický* tvar. Při zápisu jsou použity symboly pro nezávisle proměnné x, y a pro hledanou funkci u :

$$\frac{\partial^2 u}{\partial x^2} = f\left(x, y, u, \frac{\partial u}{\partial x}, \frac{\partial u}{\partial y}\right).$$

Řád parciální diferenciální rovnice je určen nejvyšší derivací, která se v rovnici vyskytuje.

Řešením parciální diferenciální rovnice je funkce, která dosazena společně se svými parciálními derivacemi do rovnice tuto rovnici splňuje. Někdy se řešení vymezuje úžeji. Za řešení rovnice se považují pouze takové funkce, které splňují rovnici a mají spojité parciální derivace, vyskytující se v rovnici.

Lineární parciální diferenciální rovnice je taková rovnice, ve které se neznámá funkce a všechny její derivace vyskytují v první mocnině, současně neobsahuje žádný součin derivace a funkce, ani žádný součin derivací. Jinak je rovnice *nelineární*. Speciálním případem nelineární rovnice je rovnice *kvazilineární*.

8.1.2 Metoda řešení parciálních diferenciálních rovnic – metoda konečných prvků^[26]

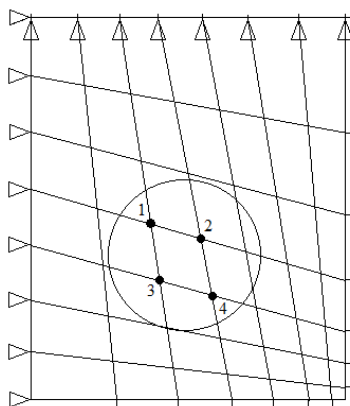
Metoda konečných prvků je obecnou metodou numerického řešení parciálních diferenciálních rovnic a sestává se z těchto základních kroků:

1. Diskretizace

Analyzovanou strukturu rozdělíme na podoblasti, tzv. konečné prvky, které se vzájemně nepřekrývají a jejichž sjednocení zahrnuje všechny body analyzované oblasti. Jedná se o diskretizaci modelu – nahrazení nekonečného objemu modelu konečným počtem prvků, resp. uzlových bodů. Abychom mohli parametry analyzované struktury v prostoru konečného prvku považovat za konstantní, měly by být velmi malé. Velikost a tvar konečných prvků nejsou nijak omezeny. Analyzovaná oblast se dělí v 1D prostoru na úsečky, ve 2D prostoru na trojúhelníky a ve 3D prostoru se vytváří síť ze čtyřtěsnů, tzv. tetrahedronů.

Na obrázku 8.1 je v principu znázorněn model složený z obdélníkových *prvků*, tzv. elementů. Černými body v průsečících jsou znázorněny *uzly* nebo také *uzlové body*, které určují místa

spojení jednotlivých prvků. Zatížení nebo potlačení stupňů volnosti lze definovat pouze prostřednictvím těchto uzlů.



Obr. 8.1: Model složený z obdélníkových prvků.^[26]

2. Aproximační funkce

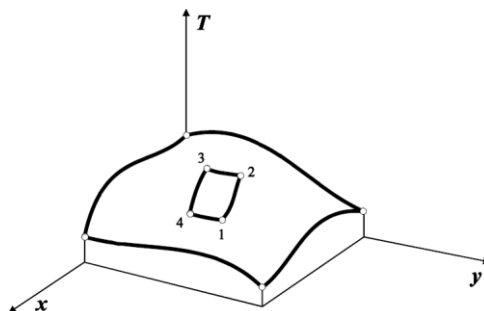
Po rozdělení analyzované oblasti na jednotlivé konečné prvky můžeme formálně vyjádřit aproximaci hledané neznámé funkce nad plochou každého konečného prvku. Přitom neznámé řešení obvykle aproximujeme lineární kombinací zvolených *bázových funkcí* a *neznámých koeficientů*. Fyzikální vlastnosti tělesa, posunutí, napětí, teplota, atd. lze nahradit funkcí prostorových souřadnic. Tato funkce se nazývá *aproximační funkcí* nebo také *funkcí tvaru*. Bázová funkce je funkce, která tvoří jakousi základnu, tzv. *bázi*, pro aproximaci hledané neznámé funkce.

Řešení problému získáme vyřešením maticové rovnice pro vektor neznámých aproximačních koeficientů. Dosadíme-li tyto koeficienty do formální aproximace, dostaneme aproximaci hledané funkce v každém bodu každého konečného prvku. Sjednocením aproximací nad všemi konečnými prvky získáme celkovou aproximaci ve všech bodech prostoru, nad kterým jsme řešení hledali.

Neznámou funkci $T(x, y)$ (Obr. 8.2) nahradíme v jednotlivých uzlech aproximační funkcí, která musí mít tolik členů, kolik má prvek uzlů. Pro obdélníkový prvek vznikne polynom čtvrtého stupně:

$$\Phi = a_1 + a_2x + a_3xy + a_4y,$$

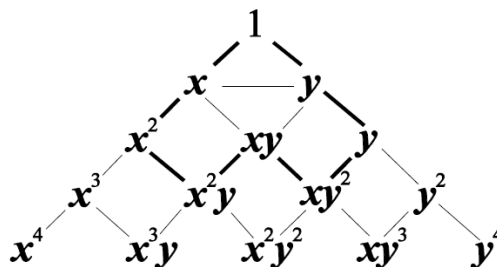
který se snaží přiblížit k funkci $T(x, y)$. Koeficienty rovnice a_i získáme na základě řešení polynomu pro všechny čtyři uzly obdélníkového tvaru, to znamená, že řešíme soustavu čtyř rovnic o čtyřech neznámých.



Obr. 8.2: Zobrazení funkce $T(x, y)$.^[26]

Polynomy s prvky s více uzly získáme z Pascalova trojúhelníku (Obr. 8.3). Rozhodnutí, který prvek se kterou aproximační funkcí použít, nemá jednoznačná pravidla. Jeden prvek dává více

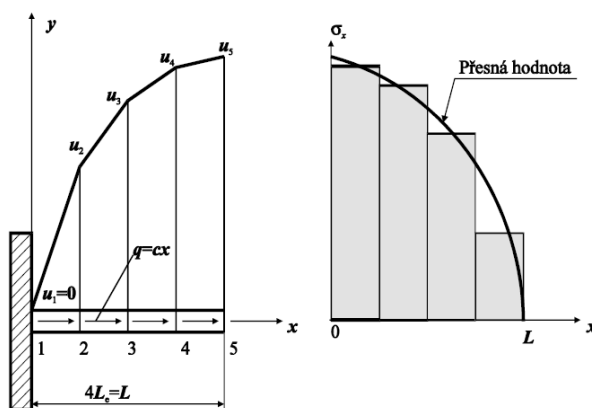
či méně přesný výsledek v závislosti na tvaru, okrajových podmínkách a druhu analýzy. Obecně ale platí, že s rostoucím počtem uzlů roste i přesnost aproximační funkce a tím i celkového výsledku analýzy. Čím více uzlů prvků a počet samotných prvků, tím větší náročnost na kapacitu a výkon počítače.



Obr. 8.3: Pascalův trojúhelník.^[26]

3. Interpolace (Obr. 8.4)

Interpolace je postup, kterým se přibližně určuje hodnota funkce $f(x)$ v bodě $x \in (a, b)$, známe-li její hodnoty v jiných bodech intervalu $\langle a, b \rangle$. Malá část složitého pole může být modelována jednoduchým polem.



Obr. 8.4: Interpolace.^[26]

4. Sestavení a řešení rovnic, zadání okrajových podmínek.

V metodě konečných prvků je pro analýzu hledanou hodnotou posunutí v jednotlivých uzlech a odtud jsou pak odvozeny další výsledky: napětí, deformace, atd. Za základ byl zvolen princip minimální potenciální energie, který říká, že ze všech možných hodnot posunutí (x, y, z) daných okrajovými podmínkami je nejpravděpodobnější ta, pro kterou bude celková potenciální energie tělesa minimální. Tím určíme vlastně potenciální energii všech uzlových stupňů volnosti pole posunutí, které je definované interpolací z posunutí v jednotlivých uzlech. Minimum potenciální energie získáme, položíme-li první derivaci rovnou nule. Řešením je pak rovnice s argumenty odpovídajícími matici tuhosti prvku vektoru ekvivalentního zatížení.

Vektor posunutí prvku uvnitř prvku je interpolován z vektoru uzlových posunutí pomocí funkce tvaru, kde forma funkce tvaru rozhoduje o kvalitě řešení. Každý stupeň volnosti ve vektoru posunutí prvku $\{d\}$ je obsažen v globálním vektoru posunutí $\{D\}$, který zahrnuje každý stupeň volnosti tělesa.

Potenciální energie tělesa je funkcí globálního vektoru posunutí. Zjistíme-li minimum, získáme základní statickou rovnici metody konečných prvků:

$$\{R\} = \{D\}[K],$$

kde $[K]$ je matice tuhosti prvku a $\{D\}$ je globální vektor posunutí.^[26]

5. Finální výstup

Většinou se jedná o grafické zobrazení na modelu s výpisem důležitých hodnot.

Metoda konečných prvků v sobě zahrnuje celou řadu možností pro řešení parciálních diferenciálních rovnic a proto není možné uvádět všechny tyto možnosti.

8.1.3 Nástin řešení příkladu pomocí metody konečných prvků (ANSYS)^[30]

Stěžejním problémem při určování elektrodynamických sil je výpočet elektromagnetického pole. Jak již bylo řečeno, analyticky se tento úkol provádí velmi obtížně. Základním matematickým aparátem jsou i v tomto případě Maxwellovy rovnice, jejich řešení však probíhá numericky pomocí výpočetní techniky. Výsledkem řešení není jen jedna hodnota fyzikální veličiny (elektromagnetická indukce, intenzita), ale rozložení elektromagnetického pole na celé oblasti modelu. Tím lze získat mnohem ucelenější představu, a odhalit jinak skryté souvislosti. Numerické metody jsou tak fakticky jedinou možností, jak postihnout silové působení ve složitých tvarech a konfiguracích proudovodných drah s maximální komplexností.

Analýza fyzikálního pole je zpravidla rozdělena do tří základních etap (Obr. 8.5):

a) PreProcessing

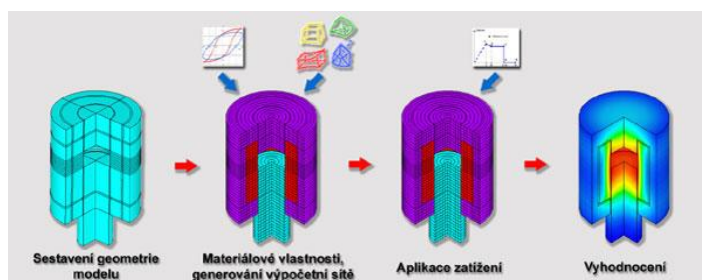
V této fázi dochází k vytváření modelu a definici jeho geometrických rozměrů. Následuje volba materiálových vlastností a generování výpočetní sítě. Většinou se zde aplikují i okrajové podmínky (kolmost a rovnoběžnost magnetických toků, neohraničenost okolního prostoru) a zatížení (úbytek napětí, proud, proudová hustota). Model může být jedno, dvou či trojrozměrný (1D, 2D, 3D).

b) Solution

Zde probíhá volba typu analýzy (statická, harmonická, transientní), výběr řešiče optimalizovaného pro dané fyzikální pole a nastavení požadované přesnosti. Podle typu analýzy se pak volí výpočetní časy či frekvence, způsob zápisu a tisku výsledků atd. Ve většině výpočetních programů je tato část plně automatizována.

c) PostProcessing

V této závěrečné části se provádí vyhodnocení řešené úlohy. K dispozici bývá několik možností grafické interpretace výsledků, z nichž nejpoužívanější je zobrazení mapy elektromagnetického pole, či vynesení závislosti elektromagnetických veličin (na čase, rozměru, teplotě, rychlosti a pod.).



Obr. 8.5: Nastínění základního postupu v podobě diagramu^[30]

Výše uvedený postup je naznačen pouze okrajově a vychází z obecných principů řešení numerických modelů. Tyto postupy jsou většinou shodné u všech výpočetních programů, a pokud se liší, pak jsou velmi dobře osvětleny v příslušných obslužných manuálech.

8.1.4 Výhody a nevýhody výpočtových metod CAD systémů^[27]

Výhody:

- Zkrácení vývojového času.
- Snížení výrobních nákladů na zavedení do výroby.
- Zvyšování kvality.
- Inovace, dodržování norem, ...

Nevýhody:

- Vysoká náročnost na výpočetní techniku.
- Nutná vysoká teoretická odbornost uživatele.
- Vysoká pořizovací cena.

8.2 Vedení tepla

a) Rovnice pro vedení tepla^[31]

Právě pomocí parciálních diferenciálních rovnic druhého řádu se dá popsat mnoho fyzikálních problémů.

Je lineární parciální diferenciální rovnice druhého řádu:

$$\frac{\partial u}{\partial t} - \frac{\partial^2 u}{\partial t^2} = 0.$$

Charakteristická rovnice pro rovnici vedení tepla je:

$$(dt)^2 = 0.$$

a charakteristiky jsou tedy přímky: $t = \text{konst.}$

Rovnice pro vedení tepla je parabolická, má jednu soustavu charakteristik a každým bodem prochází jedna charakteristika. Vyskytuje se většinou jako matematický model různých časových fyzikálních zákonitostí. Proměnné v parabolických rovnicích se obvykle vážou k fyzikálním proměnným jako jsou čas t a polohová proměnná x . Řešení úlohy spočívá v nalezení řešení $u(x, t)$ v čase $t > 0$ a počáteční hodnota $u(x, 0)$ je dána předem.

Rovnice pro vedení tepla pro funkci $T = T(x, y, z, t)$ ve tvaru:

$$\frac{\partial T}{\partial t} c \rho = \lambda \Delta T$$

popisuje přenos tepla nebo difúzi v tekutinách v neustáleném stavu. Funkce $T = T(x, y, z, t)$ označuje teplotu tělesa v bodě $[x, y, z]$ v okamžiku t , λ je tzv. koeficient pro vedení tepla, c je měrná tepelná kapacita, ρ je hustota tělesa a $\Delta T = \frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} + \frac{\partial^2 T}{\partial z^2}$, kde Δ je tzv Laplaceův operátor.

Je-li pole *stacionární*, tzn. v ustáleném stavu, platí $\frac{\partial T}{\partial t} = 0$ a tedy $\Delta T = 0$.

Za řešení této rovnice v oblasti Ω prohlásíme každou funkci T , která má v Ω spojitě derivace podle x, y, z druhého řádu, spojitou derivaci $\frac{\partial T}{\partial t}$ a která splňuje rovnici $\frac{\partial T}{\partial t} c \rho = \lambda \Delta T$.

b) Stručná analýza problému vedení tepla^[32]

V daném tělese se pod vlivem okrajových podmínek vytvoří teplotní pole. Okrajové podmínky tvoří například jeho ohřev nebo ochlazení. Znalost tohoto teplotního pole říká, ve kterých místech daného tělesa jsou největší teplotní gradienty, tzn. kde dochází k nejprudším teplotním změnám, které mohou mít za následek porušení materiálu v důsledku vzniklých teplotních napětí. K vyřešení teplotního pole je nutná znalost diferenciální rovnice energie – rovnice vedení tepla.

Vezmeme za základ kontrolní objem tvaru krychle o stranách Δx , Δy a Δz . Do tohoto kontrolního objemu některé tepelné toky vstupují a některé z něho vystupují. Dále může být uvnitř kontrolního objemu tepelná energie generována (například průchodem elektrického proudu) a také akumulována (například déletrvajícím ohřevem), tzn. že teplota tělesa je proměnná co do polohy uvnitř tělesa, tak v čase.

Energetická bilance:

tepelný tok vstupující do kontrolního objemu vedením ve všech třech směrech x, y, z	–	tepelný tok vystupující z kontrolního objemu vedením ve všech třech směrech x, y, z	+	tepelná energie vznikající v kontrolním objemu, tzv. vnitřní zdroj	=	rychlost akumulace vnitřní energie kontrolního objemu
---	---	---	---	--	---	---

S použitím Fourierova zákona dostaneme diferenciální rovnici vedení tepla:

$$\rho c \frac{\partial T}{\partial t} = \frac{\partial}{\partial x} \left(\lambda \frac{\partial T}{\partial x} \right) + \frac{\partial}{\partial y} \left(\lambda \frac{\partial T}{\partial y} \right) + \frac{\partial}{\partial z} \left(\lambda \frac{\partial T}{\partial z} \right) + S.$$

Rovnice vedení tepla je přímým výsledkem obdržným aplikací gradientního zákona pro tok dané veličiny a 1. zákona termodynamiky (tj zákona zachování energie), S je zdroj vnitřní energie.

c) Stacionární vedení tepla^[31]

Těleso si můžeme představit jako dlouhou tenkou tyč, která je kromě konců celá tepelně izolovaná. V tomto tělese si vytvoříme síť kontrolních objemů, pro 1D situaci uvažujeme jednotkovou tloušťku tělesa ve směru y a z . Z toho plyne, že velikost kontrolního objemu je $V = \Delta x \cdot 1 \cdot 1$. Integrací předešlé rovnice přes daný kontrolní objem a zavedením profilu veličiny φ mezi dvěma sousedními body dostaneme předpoklad pro výpočet prvních derivací.

Obecnější tvar následující rovnice je základem pro standardní tvar všech rovnic. Z ní je možné odvodit rovnice pro 2D a 3D, či pro okrajové kontrolní objemy, v nichž se musí objevit vliv okrajových podmínek:

$$a_p \cdot T_p = \sum a_{nb} + b,$$

kde index nb značí sousední uzlové (síťové) body a suma je myšlena všech sousedních uzlových bodů.

Základní pravidla

Splníme-li následující pravidla při odvozování diskretizačních rovnic, zaručeně splníme platnost dvou základních předpokladů a to předpoklad fyzikálně reálného řešení a předpoklad celkové bilance pro řešené veličiny.

1) Konzistence na hranicích kontrolního objemu:

je-li hranice společná dvěma sousedním kontrolním objemům, musí být tepelný tok tekoucí přes tuto hranici popsán stejným výrazem pro oba kontrolní objemy.

2) Kladné koeficienty:

všechny koeficienty a_p a sousední a_{nb} musí být vždy kladné. Pokud dojde v jednom uzlovém bodě ke vzrůstu teploty, pak pokud zůstanou ostatní podmínky nezměněny, musí dojít v sousedním bodě také ke vzrůstu teploty a ne k jejímu poklesu.

3) Záporná směrnice linearizovaného zdroje:

je-li zdroj linearizován ve tvaru $\bar{S} = S_C + S_p T_p$, musí být $S_p \leq 0$, kde S_C je konstantní část zdroje, S_p určuje koeficient u teploty T_p a tato teplota je konstantní v celém kontrolním objemu.

4) Suma sousedních koeficientů $\sum a_{nb}$ musí být rovna koeficientu a_p :

zvýšíme-li okrajové teploty o libovolnou konstantu c , všechny teploty v daném tělese musí vzrůst o stejnou konstantu.

Okrajové podmínky

Při 1D zobrazení tyče s jedním naznačeným kontrolním objemem bude na každém jejím konci jeden uzlový koncový okrajový bod. Rozlišují se tedy vnitřní uzlové body a okrajové uzlové body. Sousední uzlové body obsahují různé teploty. Značí to, že ve dvou rovnicích se objeví hodnoty teplot okrajových uzlových bodů a právě přes tyto uzlové body se přenáší dovnitř tělesa informace o okrajových podmínkách.

Následující diferenciální rovnice popisuje 1D stacionární vedení tepla s vnitřním zdrojem:

$$\frac{d}{dx} \left(\lambda \frac{dT}{dx} \right) + S = 0.$$

kde S je zdrojový člen. Zdroj je velmi často funkcí řešené teploty.

Existují tři typické okrajové podmínky:

- 1) daná okrajová teplota (Dirichletova podmínka),
- 2) daný tepelný tok hranicí – okrajem tělesa (Neumannova podmínka),
- 3) tepelný tok hranicí daný pomocí teploty okolí a součinitele přestupu tepla (Newtonova podmínka).

Pokud je dána okrajová podmínka 1. druhu, nepotřebujeme další rovnici. Pokud ale máme danu podmínku 2. nebo 3. druhu, nejsou dány hraniční teploty a musí se tedy odvodit další rovnice pro okrajové teploty – na levém i pravém okraji, respektive na všech okrajích pro vícerozměrnou situaci.

Pro 2D zobrazení postupujeme při odvozování diskretizační rovnice stejně jako pro 1D stacionární vedení tepla, kdy byla integrována příslušná rovnice v hranicích 1D kontrolního objemu. Vnitřní uzlový bod má nejen sousední uzlové body jako u 1D úlohy v ose x , ale i uzlové body v ose y . Parciální diferenciální rovnice pro 2D stacionární vedení tepla s vnitřním zdrojem má tvar:

$$\frac{\partial}{\partial x} \left(\lambda \frac{\partial T}{\partial x} \right) + \frac{\partial}{\partial y} \left(\lambda \frac{\partial T}{\partial y} \right) + S = 0.$$

Celou rovnici budeme integrovat v mezích kontrolního objemu. Dále pak platí, že tepelný tok vystupující z kontrolního objemu ve směru x Q_{-x} ponížený o tepelný tok vstupující do kontrolního objemu ve směru x Q_{+x} a tepelný tok vystupující z kontrolního objemu ve směru y Q_{-y} ponížený o tepelný tok vstupující do kontrolního objemu ve směru y Q_{+y} plus teplo generované v kontrolním objemu, tzv. zdroj tepla Q_{zt} , se musí rovnat nule.

Bilanční rovnice popsaná v předešlém odstavci, bude vypadat takto:

$$Q_{-x} - Q_{+x} + Q_{-y} - Q_{+y} + Q_{zt} = 0.$$

Dále předpokládáme, že teploty mezi uzlovými body se mění po částech lineárně spojitě.

Pro 3D stacionární vedení tepla platí zjednodušená rovnice ve tvaru:

$$\frac{\partial}{\partial x} \left(\lambda \frac{\partial T}{\partial x} \right) + \frac{\partial}{\partial y} \left(\lambda \frac{\partial T}{\partial y} \right) + \frac{\partial}{\partial z} \left(\lambda \frac{\partial T}{\partial z} \right) + S = 0.$$

Opět budeme integrovat přes kontrolní objem. Nyní přibýly ještě uzlové body v ose z . Bilanční rovnice bude mít tvar:

$$Q_{-x} - Q_{+x} + Q_{-y} - Q_{+y} + Q_{-z} - Q_{+z} + Q_{zt} = 0.$$

O úlohách, kde jsou fyzikální parametry, např. tepelná vodivost je funkcí řešené proměnné teploty nebo vnitřní zdroj energie je funkcí teploty, mluvíme jako o úlohách nelineárních. Při řešení takových úloh předpokládáme, že daná soustava rovnic je lineární a nelinearitu zahrneme tím, že soustavu rovnic řešíme iteračním postupem, při němž se respektuje nelineární chování koeficientů. Hrubý algoritmus řešení:

1. Odstartuje se řešení s odhadnutým nebo jiným způsobem určeným teplotním polem, tj. hodnotami teploty ve všech uzlových bodech.
2. Koeficienty diskretizačních rovnic se počítají s těmito odhadnutými teplotami.
3. Následně se řeší jmenovitě lineární soustava algebraických rovnic a tím se získá nové teplotní pole, tj. nové hodnoty teplot T v uzlových bodech.
4. S tímto lepším odhadem teplotního pole se vracíme do kroku 2 a celý proces se opakuje tak dlouho, až následující iterace nezpůsobí žádnou výraznou změnu v hodnotách teploty T .

Tento konečný stav, kdy žádný další iterační krok nepřinese již žádnou změnu v řešené teplotě, označujeme jako konvergenci iterací. Toto konvergované řešení je správným řešením nelineárních rovnic i když bylo dosaženo metodami pro lineární rovnice.

d) Nestacionární vedení tepla^[31]

Jedná se o situaci, kdy teplotní pole je v čase proměnné. Teplotní pole je popsáno rovnicí:

$$\frac{\partial}{\partial x} \left(\lambda \frac{\partial T}{\partial x} \right) = \rho c \frac{\partial T}{\partial t}.$$

Pro zjednodušení předpokládáme, že ρc je konstantní. Při řešení nestacionárního teplotního pole se dá říci, že úkolem je najít teplotní pole $t + \Delta t$ pro dané teplotní pole T v čase t , tzn. že řešení nestacionárních úloh postupuje vždy ve směru rostoucího času. Pro řešení diskretizační algebraické rovnice z diferenciální rovnice použijeme stejný postup jako pro 1D stacionární problém, integrujeme jednak v mezích kontrolního objemu a jednak v časovém intervalu od t do $t + \Delta t$.

8.3 Proudění tekutin

a) Modelování proudění tekutin^[33]

Numerické modelování mnoha fyzikálních jevů je úzce spojeno s modelováním určité formy pohybu matematickými prostředky. Pohyb tekutin souvisí s řešením nejrůznějších problémů, daných fyzikálním modelem:

- laminární a turbulentní proudění v jednoduchých i složitých geometriích,
- stlačitelné a nestlačitelné proudění,
- stacionární, nestacionární a přechodové proudění,
- vícefázové proudění, proudění s volnou hladinou, proudění s pevnými částicemi, bublinami, resp. kapkami,
- proudění porézním prostředím, atd.

K řešení proudění je možno využít komerční programové systémy, jako je Fluent, CFX a další. Nezastupitelnou úlohou uživatele je znalost hydromechaniky, termomechaniky a dalších věd podle složitosti problému. U programu Fluent resp. CFX je třeba vědět, s jakými tvary konečných objemů se bude pracovat. Z toho vyplývá volba hustoty sítě, jaká aproximační schémata bude vhodné použít. U dynamiky mít představu o charakteru časové závislosti jednotlivých veličin a z toho vyplývající velikosti časového kroku, apod. Neméně významnou částí je vyhodnocení výsledků, které je obzvláště obtížné u trojrozměrných úloh. Je optimální mít k dispozici alespoň orientační hodnoty počítaných veličin, ideální je srovnání výsledků s experimentem.

Proudění kapalin je možno rozdělit podle několika hledisek.^[34]

Dělení podle fyzikálních vlastností tekutiny

Proudění ideální (dokonalé) tekutiny:

- Potenciální proudění (nevířivé).

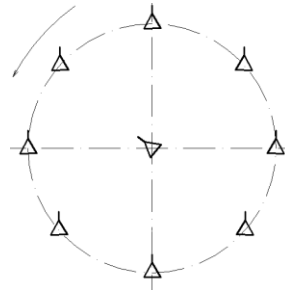
Částice tekutiny se pohybují přímočaře nebo křivočaře po dráhách tak, že vůči pozorovateli se neotáčejí kolem vlastní osy (Obr. 8.6). Natočení částice na křivé dráze je kompenzováno stejně velkým natočením částice kolem vlastní osy, ale v opačném smyslu. Mezi potenciální proudění patří rovněž potenciální vír, u něhož částice krouží kolem vřového vlákna potenciálně s výjimkou částice, která tvoří vlákno (Obr. 8.7).

- Vířivé proudění.

Částice tekutiny se vůči pozorovateli natáčejí kolem vlastních os (Obr. 8.8).



Obr. 8.6: Potenciální proudění



Obr. 8.7: Potenciální vír



Obr. 8.8: Vířivé proudění

Proudění skutečné (vazké) tekutiny:

- Laminární proudění.

Částice tekutiny se pohybují v tenkých vrstvách, aniž se přemísťují po průřezu (Obr. 8.8).

- Turbulentní proudění.

Částice tekutiny mají kromě podélné rychlosti také turbulentní (flukтуаční) rychlost, jíž se přemísťují po průřezu.

Dělení podle kinematických hledisek

Dělení proudění dle uspořádání v prostoru:

- Proudění je třírozměrné neboli prostorové (3D), jestliže veličiny, např. rychlost, závisí na poloze v prostoru $v = v(x, y, z)$.
- Proudění dvourozměrné neboli rovinné (2D) je charakterizované veličinami, jako je např. rychlost, závisí na poloze v rovině (příkladem je osově symetrické proudění v potrubí) $v = v(x, y)$.
- Proudění jednorozměrné (1D) předpokládá závislost počítaných veličin na poloze na křivce (příkladem je proudění v potrubních systémech) $v = v(s)$.

Dělení proudění podle závislosti na čase:

- Proudění ustálené (stacionární) nezávisí na čase $v \neq v(t)$; $\frac{\partial}{\partial t} = 0$.
- Proudění neustálené (nestacionární) je proudění, u něhož veličiny jsou závislé na čase, $v = v(x, y, z, t)$; $v = v(s, t)$; $v = v(t)$.

b) Přenos hmoty, hybnosti, tepla při neizotermním proudění nestlačitelné tekutiny

Základní fyzikální zákony popisující proudění jsou zákony zachování hmotnosti, hybnosti, tepla případně dalších skalárních veličin. Jsou vyjádřeny Navier-Stokesovými rovnicemi spolu s rovnicí kontinuity a popisují laminární i turbulentní režim proudění.

Rovnice pro zachování hmotnosti, hybnosti a rovnice energie v obecné konzervativní formě tvoří systém parciálních diferenciálních rovnic, přitom všechny rovnice lze formálně vyjádřit zápisem:

$$\underbrace{\frac{\partial(p\zeta)}{\partial t}}_{\text{akumulace}} = - \underbrace{\frac{\partial}{\partial x_j}(p u_j \zeta)}_{\text{konvekce}} + \underbrace{\frac{\partial}{\partial x_j} \left[\alpha_\zeta \frac{\partial \zeta}{\partial x_j} \right]}_{\text{difúze}} + \underbrace{S_\zeta}_{\text{zdroj}},$$

kde p je tlak, α_ζ je inverzní turbulentní Prandtlovo číslo, ζ je proměnná a členy na pravé straně jsou postupně *konvektivní*, *difúzní* a *zdrojový člen*, proto se rovnice nazývá také konvekčně-difúzní rovnice. Pokud ζ představuje teplotu nebo jinou skalární veličinu, pak se jedná o lineární rovnici druhého řádu, pokud ζ představuje složku rychlosti, lze tuto rovnici lze považovat za nelineární rovnici druhého řádu.

c) Okrajové a počáteční podmínky, podmínky vstupu a výstupu

Okrajová podmínka definuje teplotu na počátku a konci oblasti, okrajové a počáteční podmínky pro teplotu jsou platné pro obecnou proměnnou ζ . Okrajové podmínky nemusí být jen konstantní veličiny, ale mohou nabývat hodnot definovaných funkcí, tabulkou, apod. (konstanta, polynom. funkce, derivace podle normály, po částech lineární funkce, kombinace polynom. a po částech lineární funkce).

Podmínky vstupu a výstupu pro dvě průtočné hranice mohou nastat pouze následující základní kombinace okrajových podmínek: vstupní rychlost – výstupní rychlost, vstupní rychlost – výstupní statický tlak, vstupní celkový tlak – výstupní statický tlak. Kombinace vstupní rychlosti a výstupní rychlosti nemůže nastat, protože rychlost na druhém vstupu se počítá z rovnice spojitosti. Při uvažování rovnice energie se zadává navíc hodnota teploty.

Definují se podmínky na průtočných hranicích, podmínky na stěně, podmínky symetrie a osové symetrie a periodické podmínky.

d) Klasifikace proudění skutečných kapalin

Proudění skutečných kapalin může být klasifikováno jako *laminární* nebo *turbulentní* proudění. U turbulentního proudění bylo na základě experimentálních měření zjištěno, že na stěnách potrubí nebo obtékaného tělesa vzniká vrstva kapaliny s laminárním pohybem, tzv. *laminární podvrstva*, jejíž tloušťka je několik desetin milimetrů. Těsně za laminární podvrstvou je *přechodová vrstva* mezi laminární podvrstvou a *turbulentním jádrem*, které tvoří další oblast turbulentního proudu. Laminární podvrstva a přechodová vrstva tvoří turbulentní mezní vrstvu. Stanovení typu proudění není zcela jednoduché a jednoznačné a záleží na zkušenostech řešitele.

e) Vytvoření sítě

Numerická metoda konečných objemů je založena na vytvoření systému nepřekrývajících se elementů, konečných objemů. Původně byla metoda konečných objemů postavena na konečných objemech tvaru obdélníků a křivočarých čtyřúhelníků ve dvourozměrném případě a kvádrů nebo obecných šestistěnů v trojrozměrných úlohách. Takto vytvořená síť se nazývá *strukturovaná síť*. Zásadním pravidlem je, že hranice prvků musí sousedit s jedinou hranicí sousedního elementu, nelze tedy libovolně zhušťovat síť (je analogií pro metodu konečných diferencí včetně možnosti použití indexování). Také výsledná výpočtová oblast je pak kvádr nebo obdélník. V současné době se používá tzv. *nestruturovaná síť*. Konečným objemem je ve 3D kvádr, čtyřstěn, prizmatický a pyramidový prvek, jehož výhody již byly ověřeny v úlohách pružnosti, řešených metodou konečných prvků. Výše vyjmenované prvky se v současné době mohou kombinovat, čímž se získá optimální síť, kde v okolí stěny jsou použity čtyřúhelníky a kvádry (pro výpočet z hlediska přesnosti jsou optimální) a v dalších oblastech, kde nedochází z důvodu existence mezní vrstvy k velkým gradientům řešených veličin, se použijí zbývající prvky. Ty zajistí snadnou změnu hustoty sítě. Takto vytvořený vysíťovaný model je vyexportován do souboru s příponou .msh a tím je ukončena všechna práce v programu Fluent Gambit.

8.4 Počítačová dynamika tekutin^[35]

Modelování Computational Fluid Dynamics (CFD) – počítačová dynamika tekutin – poskytuje výkonný nástroj inženýrům, výzkumníkům a jiným technickým odborníkům ke znázornění pochodů, probíhajících ve většině typů zařízení v technické výrobě.

1) CFD modelování

Software počítačové dynamiky tekutin (CFD) umožňuje numerické řešení základních fyzikálních rovnic (jinak nazývaných jako Navier-Stokesovy rovnice), které popisují zachování hmoty, momentu a energie. Řešením těchto rovnic v několika tisících definovaných bodech na výpočtové síti, vytvořené tak, aby se blížila geometrii modelované složky zařízení nebo systému, jsou programy CFD schopny účinně simulovat základní děje, jako je tok tekutin, přenos hmoty a tepla a chemické reakce. A když se zkombinuje modelování CFD s jinými typy softwaru modelování, jsou uživatelé schopni nejen porozumět chování tekutin v systému, ale analyzovat a předpovídat odezvu struktury, únavu a vibraci strojního zařízení. Modelování CFD se nyní používá na urychlení práce při návrhu zařízení, na zkrácení doby návrhu, urychlení uvedení výrobku na trh a omezení nákladů na výrobu zmenšením nutného rozsáhlého vývoje prototypu a jeho fyzického testování.

a) Jak CFD pracuje

Software poskytuje vizuální znázornění, které usnadňuje uživateli analýzu prakticky každého typu toku v reálných podmínkách, předpovídat řadu podmínek procesu, jako jsou reakční rychlosti, míchání, odtržení proudu, rozložení materiálů, rozložení teploty nebo tlaku a rozložení smykové rychlosti. Uživatelé dále využívají stálých pokroků v přímé tvorbě sítě, která je nedílnou součástí modelování CFD, a softwaru zpracování po procesu, který se využívá na jemné doladění analýzy a vizuální znázornění výsledků modelování. Mnoho současných softwarových programů poskytuje automatickou kontrolu chyb, která může zjistit nesrovnalosti ve vstupních datech a sestavení modelu a upozorní uživatele před přechodem k dalšímu kroku. Podobně prvek „uzdravení geometrie“ pomáhá mnohým současným řešitelům CFD automaticky zjistit „ne optimální“ vlastnosti geometrie, jako jsou malé mezery, nesprávné okraje a velmi malé plochy, čímž se zlepšuje jednotnost použité geometrie a přesnost výsledků získaných modelováním.

b) Tvorba sítě

Aby se urychlila a automatizovala tvorba sítě, jsou současné programy na tvorbu sítě schopné přijímat výkresy z počítačového návrhu složky prakticky z každého komerčního CAD programu a použít je jako počáteční vstup pro tvorbu sítě. Uživatel musí určit několik základních velikostí buněk a jejich rozřídění (podle návodu uvedeného v programu) a software potom automaticky provede celou práci k vytvoření a vyhlazení sítě. Současné síťovací programy nyní například používají předem definované stavební bloky a normalizované šablony stylů sítě, založené na různých geometrických prvcích.

c) Software po ukončení procesu, ke zpracování výsledků modelu

Schopnosti dnešního propracovaného softwaru po ukončení procesu pomáhají účinněji prozkoumat velké neustálené sady dat a smysluplněji ozřejmit kritická hlediska složitých simulací. Nástroje po ukončení procesu pomáhají uživateli vytvářet barevné grafy s velkým rozlišením a animace, představující veličiny jako jsou vektory rychlosti, vrstevnice tlaku a čáry konstantních vlastností v proudu tekutiny. Uživatelé se při CFD analýzách obvykle zajímají o práci se značně menším objemem cílových dat, vztažených k jednomu zvláštnímu hledisku modelu. Je-li například předmětem určité studie CFD chování povrchového napětí, jsou pro analýzu po ukončení procesu potřebná jen data o povrchovém napětí. Typické datové výstupy zahrnují grafické prvky jako jsou izoplochy, řezné roviny, výpočetní roviny a/nebo dráhy částic, nebo mohou být data seřazena podél určitých čar nebo povrchů, aby se mohly získat důležité délkové, plošné a hmotnostní průměry.

d) Spolupráce CFD s jinými způsoby modelování

V současnosti se modelování CFD stále častěji spojuje s jinými typy softwaru modelování, aby se spojilo modelování toku tekutin s jinými fyzikálními typy modelování a simulace. Když se

například spojí modely CFD s jinými modely analýzy napětí do plné spolupráce, mohou uživatelé vykonat složitější analýzy interakce tekutina-struktura (fluid structure interaction – FSI).

2) Software pro dynamiku tekutin^[36]

Řešením společnosti ANSYS pro dynamiku tekutin je komplexní sada produktů pro modelování proudění a dalších souvisejících fyzikálních jevů. ANSYS má dva světově nejrespektovanější CFD produkty pro simulaci proudění – ANSYS Fluent a ANSYS CFX. Produkty umožňují kvalifikovaně a rychle řešit požadavky na bezpečnost výrobku, jeho spolehlivost a udržitelnost. ANSYS nabízí nekomplexnější a nejprovázanější sadu moderních CFD nástrojů, která je na trhu k dispozici. Uživatel si může vybrat, který z těchto řešičů použije pro danou analýzu.

Software ANSYS pro simulaci dynamiky tekutin nabízí bezkonkurenční šíři a hloubku v modelování proudění a souvisejících fyzikálních dějích. Velice snadno lze získat výsledky vazkého a turbulentního proudění, interní nebo externí aerodynamiky.

Technologie ANSYS pro dynamiku tekutin vyniká v modelování turbulentního proudění díky špičkové sadě turbulentních modelů, které pokrývají i ty nejsložitější typy proudění. Predikce turbulentního proudění může být rozšířena i o modelování šíření hluku vyvolaném prouděním tekutiny. Přenos tepla lze uvažovat včetně sálání.

K dispozici je celá řada metod a numerických schémat, které zajistí, aby byly výpočty účinné a výsledky přesné. Nástroje zahrnují řešiče metody konečných objemů, využívající segregované i spřažené metody pro obecné modelování proudění tekutin, ale i řešič metody konečných prvků pro viskoelastické proudění tekutin. Lze použít plně nestrukturované hybridní sítě, které mohou obsahovat všechny typy používaných buněk včetně mnohostěnů. Při výpočtu je možné provádět manuální i automatické adaptace sítě pro dosažení velmi přesných výsledků.

8.5 Příklad k řešení

Proveďte metodou konečných prvků a metodou sítí simulaci prostupu tepla zdi z plných cihel o tloušťce 40 centimetrů. Teplota na vnitřní stěně zdi je 20°C, teplota na vnější stěně – 20°C. Testujte vliv volby vstupních parametrů na rozložení teploty. Simulujte úlohu i pro případ, kdy je na vnější, na vnitřní nebo obou stěnách umístěna izolace z polystyrenových desek o tloušťce 5 centimetrů. Využijte získaných informací k diskuzi o výhodnosti uspořádání daného způsobu izolace.

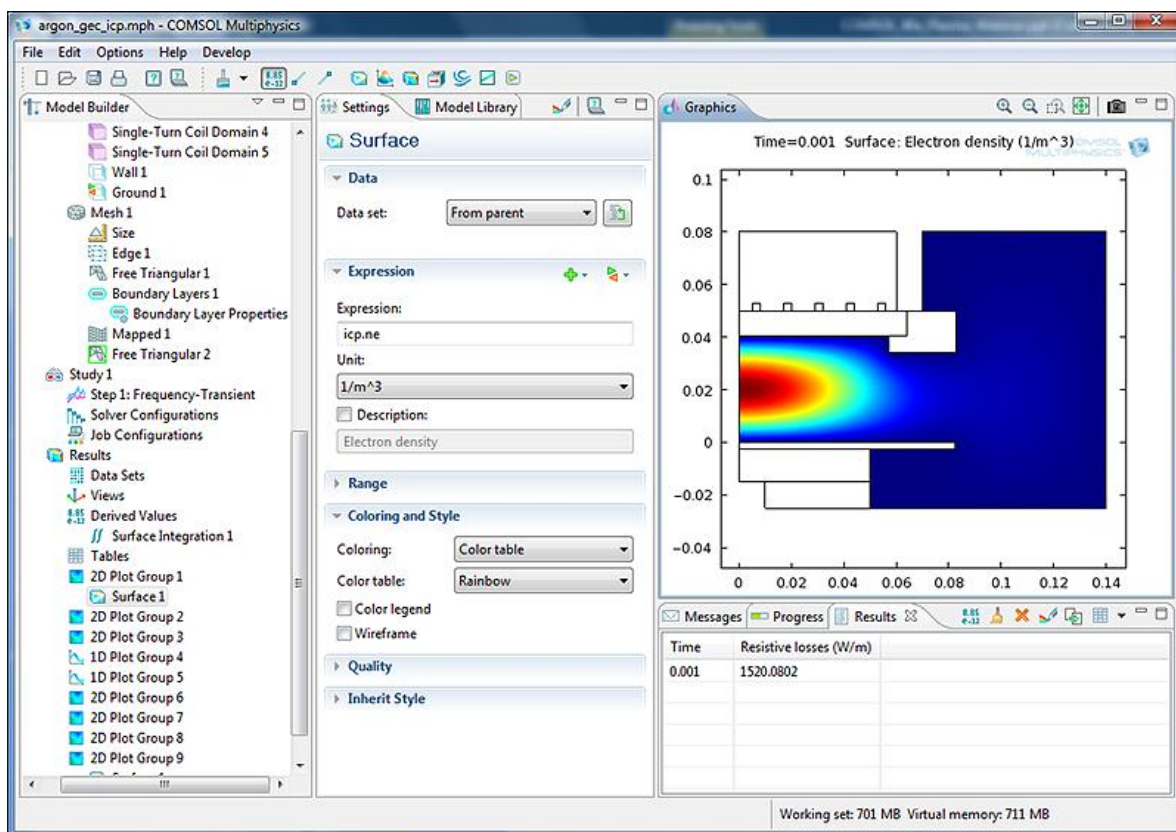
KAPITOLA 9: SOFTWARE COMSOL MULTIPHYSICS

9.1 COMSOL Multiphysics^[37]

COMSOL Multiphysics je softwarový nástroj, který umožňuje modelovat fyzikální děje a řešit fyzikální úlohy popsané parciálními diferenciálními rovnicemi metodou konečných prvků. Svě využití nachází hlavně u výzkumných a vědeckých pracovníků ve vědeckotechnických oblastech, vyhledáván je i vývojáři. Díky široké nabídce funkcí pro zobrazování vypočítaných výsledků je vhodný také pro vysoké a specializované střední školy. Hlavní výhodou tohoto programu je možnost kombinace několika fyzikálních rozhraní parciálních diferenciálních rovnic do jednoho komplexního multifyzikálního modelu. Tohoto propojení jednotlivých rovnic se dosahuje přímo v uživatelském rozhraní programu a tak není zapotřebí vytvářet jakékoliv přídavné kódy nebo skriptové soubory.

COMSOL Multiphysics nabízí tzv. rozšířenou multifyziku, což je možnost promítnutí již vypočítaných dat z jedné části geometrie do jiné její části, nebo dokonce do jiné geometrie bez ohledu na prostorovou dimenzi modelu. Řešení jedné úlohy může zahrnovat několik fyzikálních vlivů najednou a tím je možné provádět komplexnější analýzu modelu.

Postup při modelování úlohy je obdobný jako obecné řešení parciálních diferenciálních rovnic metodou konečných prvků a na základě definice prostředí, které tyto rovnice popisují, a zadání okrajových podmínek na plochách, hranách nebo bodech v daném geometrickém modelu. Řešený geometrický model je zobrazen v grafickém editoru (obr. 9.1) a může zobrazovat například zahřívání tepelný radiátor, proud vzduchu v aerodynamickém tunelu či zatíženou strojní součástku. Je nutné znát, jaké fyzikální vlivy na zobrazenou geometrii působí, a podle toho volit správné parciální diferenciální rovnice. Zvolenou parciální diferenciální rovnicí popisující modelovanou aplikaci nazýváme fyzikálním rozhraním.



Obr. 9.1: Pracovní prostor programu COMSOL^[38]

COMSOL Multiphysics obsahuje knihovny předdefinovaných parciálních diferenciálních rovnic, které jsou základem jednotlivých fyzikálních rozhraní. Výběrem rozhraní se automaticky zpřístupní dialogová okna, kam se zadávají potřebné vlastnosti oblastí, počáteční či okrajové podmínky. Nejedná se o matematické vyjádření problému, ale o definování takových fyzikálních veličin, jako jsou například vlastnosti uvažovaného materiálu (měrná hustota, kinematická viskozita

nebo tepelná vodivost), a u okrajových či počátečních podmínek se jedná například o rychlost kapaliny na vstupu, vyzařování tepla do okolí či předepsání teploty na hranici.

Předdefinovaná fyzikální rozhraní jsou určena k řešení úloh z oblasti pevnosti a pružnosti, dynamiky tekutin, nejaderných plazmových reaktorů, elektromagnetismu, přestupu tepla, chemických reakcí, povrchové úpravy materiálů, atd. Systém je tzv. otevřený, což znamená možnost úpravy předdefinovaných rovnic podle svých potřeb a požadavků nebo zavedení vlastních partiálních diferenciálních rovnic pomocí dostupných fyzikálních rozhraní.

a) Pracovní postup při modelování úlohy:

1. Geometrie zkoumaného modelu

Geometrii zkoumaného modelu můžeme vytvořit CAD nástroji v grafickém editoru nebo jej můžeme importovat z jiného externího CAD systému. Program je schopen načítat geometrické soubory ve formátech STL a VRML, které definují model povrchovou sítí, 2D soubory v DXF formátu a modely popsané 3D sítí ve formátu NASTRAN. Načítání dalších geometrických dat zajišťuje specializovaný modul a jeho nadstavby.

2. Zadání okrajových podmínek a vlastností oblastí

Pro řešení úlohy je nezbytnou podmínkou zadání okrajových podmínek a vlastností oblastí v modelu. Každé geometrické těleso je rozděleno na několik částí, což právě umožňuje zadávat různým částem modelu různé okrajové podmínky. Částem geometrie, jako jsou oblasti, plochy (ve 3D), hrany nebo body (ve 2D), mohou být přiřazeny proměnné, výrazy a nebo funkce. Při zadávání vlastností subdomén je k dispozici knihovna materiálů i chemických prvků. Vytvářený model může obsahovat několik oblastí a každé z nich lze přiřadit vlastnost rozdílného prostředí nebo materiálu – vlastnosti mechanické, tepelné nebo chemické. Uživatel si může přidávat do připravené materiálové knihovny další materiály nebo si může vytvořit knihovnu vlastní.

3. Generování výpočetní sítě

Geometrický model s nastavenými okrajovými podmínkami je připraven pro generování FEM sítě, v jejíchž uzlových bodech budou vypočtena potřebná data. Síť může být generována automaticky nebo lze vlastnosti sítě ovlivňovat nastavováním různých parametrů ve zvolených částech modelu. V jednom modelu lze nastavit několik variant sítí s různým typem a řádem elementů. Vytvořené varianty souvisí také s použitým typem řešiče pro danou úlohu.

4. Řešení modelu

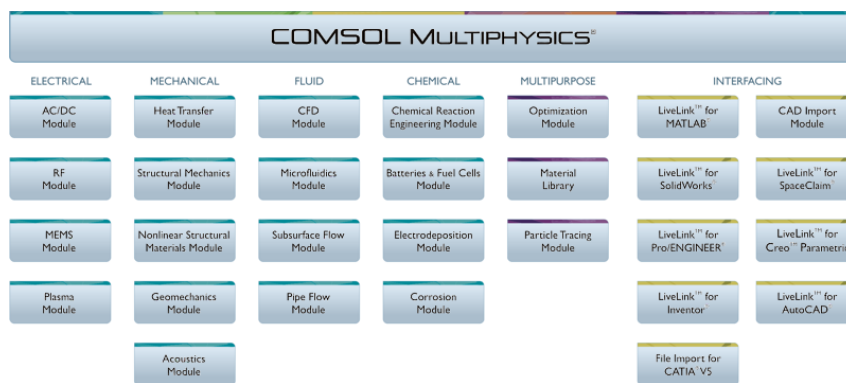
Pro řešení modelu obsahuje program několik typů řešičů, které řeší lineární i nelineární úlohy, úlohy ve frekvenční a časové oblasti nebo úlohy se zvoleným parametrem. Pro řešení soustav lineárních rovnic jsou k dispozici přímé řešiče UMFPACK a SPOOLES, iterační řešič GMRES a také řada předpodmiňovačů. Řešení úlohy může být spuštěno z grafického rozhraní COMSOL Multiphysics. Pokud je úloha popsána v textovém *M*-souboru, lze k jejímu řešení využít příkazového řádku programu MATLAB spuštěním tohoto souboru. Dalším způsobem řešení může být zpracování úlohy v dávce.

5. Konečné zpracování výsledků

Multifyzikální úlohy obsahují různé typy vypočtených proměnných, které lze ve zvolených jednotkách zobrazovat současně pomocí barevných map, izochar, izoploch, proudnic, šipek, částic nebo řezů. Úlohy řešené v čase lze snadno animovat s možností zápisu do formátu AVI, Quick Time, GIF nebo Flash. Jakákoliv řešení je možné pro další zpracování exportovat do jednoduchých textových souborů. Export vytvořených obrázků a grafů je samozřejmostí. Celý model může být exportován v datové struktuře do prostředí MATLABU nebo zapsán do textového *M*-souboru.

b) Některé vybrané moduly obsažené v programu COMSOL Multiphysics^[37]

Struktura systému COMSOL Multiphysics (Obr. 9.2):



Obr. 9.2: Struktura systému COMSOL Multiphysics^[37]

- *Module* – modul obsahuje hotové nástroje k simulaci základních procesů na elektrodách a v elektrolytech u různých typů baterií, akumulátorů a palivových článků.
- *CFD Module* – modul je určený k simulaci úloh z oblasti dynamiky tekutin.
- *Heat Transfer Module* – modul je určený k modelování úloh z oblasti přestupu tepla vedením, prouděním i sáláním.
- *MEMS Module* – modul obsahuje multifyzikální aplikace z oblasti strukturální mechaniky a elektromagnetismu s možností jejich provázání s dalšími fyzikálními aplikacemi, jako je například přestup tepla.
- *Microfluidics Module* – modul vhodný pro řešení různých druhů proudění v mikrokanálech. Jedná se především o dynamiku tekutin v zařízeních, jejichž rozměr je v řádech mikro- nebo nanometrů.
- *Structural Mechanics Module* – modul analyzuje součástky a subsystemy, ve kterých je třeba řešit deformace v důsledku zatížení, řeší úlohy v pružnosti a pevnosti v mechanice tuhých těles, jako jsou například deformace při zatížení nebo dislokační vztahy v lineární i nelineární formě.
- *Chemical Reaction Engineering Module* – řeší úlohy z oblasti chemie a chemického průmyslu.
- *Nonlinear Structural Materials Module* – nadstavbový modul rozšiřuje možnosti modelování problémů z oblasti pružnosti a pevnosti o nelineární materiálové moduly.
- *Pipe Flow Module* – jedná se o nadstavbový specializovaný modul určený k simulacím jevů v potrubních sítích a kanálech. Do modelu lze zahrnout jak proudění, tak i přestup tepla v tekutině, případně akustiku v rozvodných sítích.
- *Plasma Module* – slouží k řešení všech typů nejaderných plazmatických reaktorů.
- *RF Module* – modul je určený k modelování elektromagnetického vlnění vysokých frekvencí. Dále umožňuje simulovat mikrovlnná zařízení nebo součásti z optiky.
- *Subsurface Flow Module* – slouží k modelování podpovrchového proudění a nabízí další aplikace k analýze procesů pod zemským povrchem.
- *Material Library* – je nadstavbový modul, který poskytuje pohodlnější definování prostředí (subdomény).
- *Optimization Module* – umožňuje optimalizaci úloh a modelů vytvořených pomocí všech nadstavbových modulů.
- *Particle Tracing Module* – je určený k výpočtu trajektorií jednotlivých částic například v proudící tekutině.
- *CAD Import Module* – slouží k importu a úpravě geometrie pro potřeby FEM analýzy.

COMSOL Multiphysics obsahuje ještě další nadstavbové moduly, které jsou podrobně popsány v manuálu tohoto programu.

c) Použití programu COMSOL Multiphysics v praxi

Na následující webové adrese je velmi detailně popsáno vytvoření modelu součástky umožňující připojení stejnosměrného proudu od transformátoru k elektrickému zařízení. Model simuluje tepelné zatížení této součástky Joulovým teplem. Sběrnice je vyrobena z mědi a šrouby jsou titanové. Volba materiálů je velmi důležitá, protože titan má nižší elektrickou vodivost než měď a tudíž bude mít vyšší proudovou hustotu. Cílem simulace je přesně spočítat tepelné zatížení této součástky, její tepelnou roztažnost a vliv proudění vzduchu na její ochlazování. Celou simulaci provází krok za krokem program Wizard.^[39]

<http://www.comsol.eu/products/tutorials/introduction/page2/> .

Další velmi dobře vytvořenou prezentací je simulace pod názvem Plasma Model for a GEC ICP Reactor na webové adrese (viz níže), kde autor svůj postup provází ukázkami spojenými se slovním komentářem (Obr. 9.1).^[40]

<http://www.comsol.eu/products/tutorials/Plasma%20Model%20for%20a%20GEC%20ICP%20Reactor%20-%20Tutorial/> .

Oblasti využití

Akustika, antény, biologie, bioinženýrství, chemické reakce, difuze, ekologie, elektromagnetismus, životní prostředí, dynamika tekutin, palivové články, geofyzika, prostup tepla, matematika, aplikované PDEs, MEMS, mikrovlnná zařízení, nanotechnologie, optika a fotonová fyzika, fyzika, proudění pórovitými látkami, kvantová mechanika, polovodičové prvky, pružnost a pevnost, transportní jevy, šíření vln, poroelasticita, elektrochemické reakce, podpovrchové proudění, chromatografie...

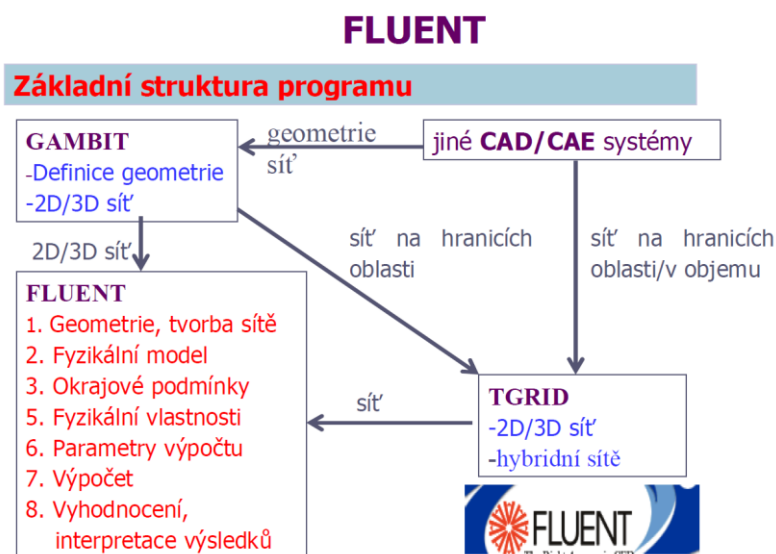
9.2 Příklad k řešení

Pomocí programu COMSOL Multiphysics simulujte rozložení teplot ve zdivu v okolí okenního otvoru a jeho výplně.

10.1 Software ANSYS Fluent

Computational Fluid Dynamics (CFD) je jednou z oblastí mechaniky tekutin, která používá numerické metody a algoritmy k řešení a analyzování problémů, které zahrnují proudění tekutin. Probíhající výzkum přináší software, který zvyšuje přesnost a rychlost simulace složitých problémů, jako jsou jevy v oblasti rychlosti blížící se rychlosti zvuku nebo s výskytem turbulentních toků. Základem téměř všech problémů CFD je Navier-Stokesova rovnice, která definuje každé jednotlivé fázové proudění tekutin, nebo Eulerovy rovnice, pokud nepočítáme viskozitou.^[42]

ANSYS Fluent je program, který obsahuje fyzikální modely poskytující široké možnosti potřebné k modelování proudění vzduchu, přenosu tepla či turbulence až po návrh a výrobu polovodičů. Je využíván jako integrální součást designu a optimalizační fáze vývoje produktu. Rozšířená technologie tohoto nástroje poskytuje rychlé a přesné výsledky, flexibilní pohyb a pokročilé techniky síťování a současně vyšší paralelní rozšiřitelnost. Uživatelem definované funkce umožňují implementaci nových uživatelských modelů a jejich značnou úpravu dle požadavků uživatele. Interaktivní řešení, nastavení výpočtu a integrované vyhodnocování výsledků umožňuje kdykoli pozastavit výpočet, prohlédnout si výsledky, upravit nastavení a poté pokračovat ve výpočtu v jednom spuštění programu. Tyto nástroje dovolují uživatelsky zjednodušený přenos dat a sdílení výsledků mezi aplikacemi (např. použití výsledků z proudění tekutin jako okrajové podmínky pro strukturální mechanické simulace).



Obr. 10.1: Zobrazení základní struktury programu.^[33]

1) Síťování, numerika a paralelní zpracování^[33]

Program využívá technologie nestrukturované sítě. Síť může být vytvořena z elementů ve tvaru čtyřstěnů a trojúhelníků v případě 2D simulací, šestistěnů, čtyřstěnů, mnohostěnů, prizmatických a pyramidových buněk pro 3D simulace (Obr 10.1). Propracovaná numerika programu zajišťuje přesné výsledky na jakékoli kombinaci nestrukturované sítě. Program běží efektivně se všemi fyzikálními modely a typy proudění – stacionární i nestacionární, nestlačitelné i hypersonické. Pokročilá schopnost paralelního běhu výpočtu může být využita téměř na jakékoli platformě (Windows, Linux i Unix).

Přehled metod řešení parciálních diferenciálních rovnic^[33]

a) **Diferenční metoda** je nejstarší všeobecně známou metodou řešení diferenciálních rovnic, která je použita v ilustrativním příkladu vedení tepla v tyči. Spočívá v nahrazení derivací diferencními podíly použitím Taylorova rozvoje, odvozením diferencních rovnic a jejich řešením.

b) **Metoda konečných objemů** spočívá stručně řečeno ve třech základních bodech:

- dělení oblasti na diskrétní objemy užitím obecné křivočaré sítě,
- bilancování neznámých veličin v individuálních konečných objemech a diskretizace,

- numerické řešení diskretizovaných rovnic.

Fluent definuje diskretní konečné objemy užitím *non-staggered schématu*, kdy všechny proměnné jsou uchovávány ve středech konečných objemů. V současné době se začíná prosazovat v řešení proudění také *metoda konečných prvků*, která spočívá v těchto bodech:

- násobení diferenciální rovnice bázovými funkcemi,
- dělení oblasti na trojúhelníkové nebo čtyřúhelníkové prvky ve dvourozměrné oblasti (2D) a čtyřstěny resp. šestistěny ve třírozměrné oblasti (3D),
- integrace přes konečné elementy založená na variačním principu,
- minimalizace reziduálů.

Speciální metodou je *spektrální metoda* vhodná pro periodické proudění v jednoduchých oblastech (Taylorovy víry vznikající v mezeře mezi koncentrickými válci, z nichž jeden rotuje).

Rovnice řešené ve Fluentu jsou rozšířením předchozích na třídímenzionální křivočarý souřadný systém. Každá iterace sestává z kroků, které jsou popsány následovně:

- pohybové rovnice pro neznámé složky rychlosti jsou řešeny s užitím hodnot tlaků tak, aby se aktualizovalo rychlostní pole,
- rychlosti určené v předchozím bodě nemohou splňovat rovnici kontinuity, proto se určují tzv. tlakové korekce a následně korekce i rychlostního pole,
- pomocí nových hodnot rychlostí se řeší rovnice pro turbulentní energii a disipaci,
- řeší se další rovnice pro určení teploty a dalších skalárních veličin,
- aktualizují se fyzikální vlastnosti kapalin (např. viskozita),
- kontrola konvergence.

2) Výběr interpolačního schématu^[33]

Fluent ukládá složky rychlosti a skalární veličiny v geometrických středech konečných objemů definovaných sítí. Z důvodu výpočtového procesu jsou potřebné hodnoty těchto veličin na hranicích konečných objemů. Tyto hodnoty jsou získány interpolací, přitom si lze vybrat mezi následujícími třemi variantami lišícími se řádem přesnosti (vzestupně):

- mocninová interpolace,
- kvadratická upwind interpolace (QUICK),
- interpolace druhého řádu/centrální diference.

Při velkých změnách tlaků a průtoků je vhodné rozpočítat úlohu s nejnižším řádem přesnosti (což je předdefinováno) a po několika iteracích využít vyšší řád přesnosti (pro proudění se zavíráním, s přenosem tepla, disipací apod.).

3) Konvergence^[33]

Residuály

Při simulaci proudění pomocí programu Fluent je velmi důležité získat konvergentní řešení. Mírou konvergence jsou reziduály, které představují maximum rozdílu dvou odpovídajících si veličin ve stejném bodě sítě ve dvou po sobě následujících iteracích. Residuály jsou vyhodnocovány pro všechny počítané veličiny v každém kroku iterace a zobrazovány pro vybrané veličiny.

Urychlení konvergence

Konvergence je ovlivněna mnoha faktory, jako je počáteční odhad, velký počet buněk, relaxační faktor atd. Pro urychlení konvergence se navrhuje využít *počátečního odhadu* proměnných významných pro proudění, což je nejlepší způsob, jak začít řešit úspěšně úlohu. V opačném případě jsou všechny veličiny definovány inicializací, často jsou pokládány rovny nule na počátku výpočtu. Nejvýznamnější příklady nastavení počátečních podmínek jsou:

- teplota pro problémy řešící přenos tepla při užití stavové rovnice,
- rychlost při velkém počtu buněk,

- teplota i rychlost při řešení přirozené konvekce,
- proudění s reakcí, kdy je dobré nastavit teplotu i hmotnostní podíly.

4) Technika step by step^[33]

Důležitou technikou k urychlení konvergence je *technika step by step* (postupně od jednoduché úlohy ke složitější). Při řešení problému s přenosem tepla je dobré začít výpočet z izotermního proudění, při řešení reagujícího proudění z proudění bez reakce se zahrnutím příměsí. Problém se nadefinuje nejprve celý a teprve potom se vyberou proměnné, pro které se vyřeší počáteční stav.

Relaxace

Z důvodu nelinearity diferenciálních rovnic není obecně možné získat hodnoty všech proměnných řešením původně odvozených aproximačních diferenčních schémat. Konvergence lze však dosáhnout užitím relaxace, která redukuje změny každé proměnné v každé iteraci.

5) Dynamické a pohybující se sítě^[33]

Možnost využití dynamických sítí v programu vyhovuje potřebám náročných aplikací jako je např. simulace proudění a spalování ve válcích motoru. Je možné použít několik různých schémat pro změnu sítě během výpočtu, včetně vrstvení, vyhlazování a přesítování, při simulaci různých pohyblivých částí v jednom výpočtu. Před zahájením výpočtu je potřeba vytvořit pouze počáteční síť a definovat pohyb pohyblivých částí. Zabudovaný nástroj má až šest stupňů volnosti, které lze využít při simulaci.

6) Turbulence^[33]

Program nabízí jedinečnou šíři modelů turbulence, jako jsou různé verze časově středovaných k-epsilon modelů, modely k-omega a modely s řešením Reynoldsových napětí (RSM). Současný pokrok v modelování turbulence vede k implementaci dalších modelů jako je turbulentní přechodový model, který je významný pro detailní modelování přechodu od laminárního k turbulentnímu proudění vyskytující se v blízkosti stěn, zatímco řešení turbulence v přechodových nestabilitách (silná separace) je bez explicitní sítě a také závislosti na časovém kroku.

7) Vícefázové proudění^[33]

Program nabízí i modelování vícefázového proudění. Modely umožňují pochopit detaily o proudění uvnitř zařízení, které je obtížné zjistit jiným způsobem. Nabízí plný Eulerův vícefázový model s řešením transportních rovnic pro každou fázi, ale i jednodušší a ekonomičtější přístup s řešením rovnic pro směs fází. Oba modely lze použít pro simulaci s chováním částic v proudu tekutiny.

8) Zpracování výsledků a export dat^[33]

Nástroje pro zpracování výsledků jsou používány pro tvorbu grafických výstupů, animací a automatických reportů, které umožňují snadný výstup vypočtených výsledků. Stínované a průhledné plochy, proudnice, vektory, kontury, uživatelsky definované proměnné a tvorba scény, to je jen několik výstupů, které jsou k dispozici (Obr. 10.2). Data mohou být využita pro další analýzy.



Obr 10.2: Síť z mnohostěnů a rozložení tlaků na formuli F1.^[41]

9) Uživatelské nástroje^[33]

K dispozici je komplexní dokumentace a několik výukových materiálů, dále také profesionální technická pomoc, další přídatné moduly pro specializované aplikace např. pro simulaci magnetohydrodynamiky.

10.2 Software ANSYS GAMBIT^[41]

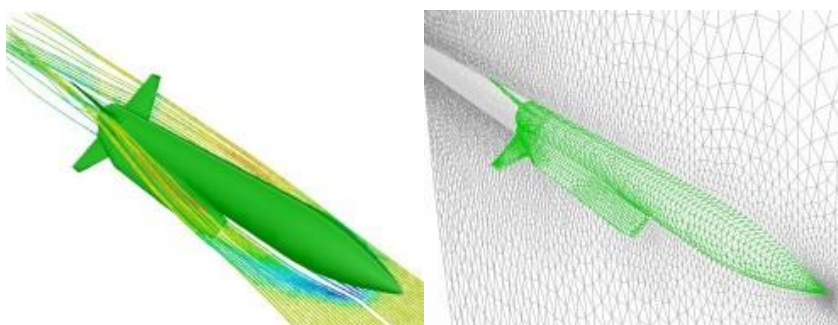
Program spojuje tvorbu geometrie se síťováním a tím je uceleným nástrojem pro preproceing. Pokročilé nástroje pro editaci příkazů umožňují editovat a současně přehrávat příkazy a tím i vytvářet skripty pro parametrické studie. Program využívá možnosti načtení dat z CAD programů, „čištění“ geometrie, dekompozici objemů a nástrojů pro tvorbu sítí. Spojením těchto činností se stává jedním z nejjednodušších, z nejrychlejších a nejpřímějších preprocesorovým nástrojem.

(Poznámka: Preprocesor je počítačový program, který zpracovává vstupní data tak, aby výstup mohl dále zpracovávat jiný program. Preprocesor je často používán pro předzpracování zdrojového kódu před vlastní kompilací. Druh a míra předzpracování závisí zejména na schopnostech preprocesoru. Většina preprocesorů je relativně jednoduchá, zvládá nahrazování textu a jednoduchá makra. Existují též sofistikované preprocesory, případně plně rozvinuté programovací jazyky.)

1) Tvorba geometrie a sítě

Jako nejmodernější preprocesor pro analýzu obsahuje mnoho nástrojů pro tvorbu geometrie a sítě ve výkonném, flexibilním, plně integrovaném a jednoduše ovladatelném grafickém prostředí (Obr 10.3).

Použitím „virtuální“ geometrie a vyspělých nástrojů na „začišťování“ geometrie jsou importované geometrie z jiných systémů rychle překonvertovány na oblasti pro výpočet proudění a vedení tepla. Obsáhlá sada vysoce automatizovaných nástrojů pro tvorbu sítě zaručuje tvorbu nejlepší sítě: strukturované, multiblokové, nestrukturované či hybridní.



Obr 10.3: Ukázka vytvořené sítě.^[41]

2) Inteligentní síťování

Nástroje pro síťování pomohou s dekompozicí geometrie pokud chceme použít strukturovanou síť nebo chceme mít dokonalou kontrolu nad sítí a používat převážně síť složenou ze šestistěnů. Trojúhelníková povrchová síť a čtyřúhelníková prostorová síť mohou být jednoduše vytvořeny v jednom prostředí spolu s pyramidovými buňkami, okrajovými buňkami u stěn pro vytvoření hybridní sítě. Navíc je možné využít nástrojů pro plošné a prostorové zahušťování sítí v kritických oblastech a tím umožnit simulaci proudění úzkými škvírami a oblastmi s velkými gradienty.

10.3 Software ANSYS TGrid^[41]

Specializovaný preprocesor určený k tvorbě nestrukturovaných objemových sítí pro komplexní geometrie.

1) Tvorba nestrukturované objemové sítě

Nabízí tvorbu objemových sítí s využitím čtyřstěnů, pyramid a prizmatických vrstev ze šestistěnů a trojbokých hranolů. To znamená daleko větší flexibilitu při tvorbě sítě, především u složitějších tvarů a geometrií. Robustní a automatizované algoritmy výrazně šetří čas a vytváří dokonalé sítě pro analýzu.

2) Import dat

Nástroje pro zpracování výsledků v programu ANSYS Fluent jsou používány pro tvorbu grafických výstupů, animací a automatických reportů, které umožňují snadný výstup vypočtených výsledků z CFD programu. Stínované a průhledné plochy, proudnice, vektory, kontury, uživatelsky definované proměnné a tvorba scény, to je jen několik výstupů, které jsou k dispozici.

Plošné a objemové sítě mohou být také importovány z programu ANSYS, CATIA, I-DEAS, NASTRAN, PATRAN, Pro/ENGINEER, Hypermesh a dalších.

3) Nástroje

Obsáhlý soubor nástrojů slouží k vylepšení importované plošné sítě a spojování jednotlivých částí sítě při importu z několika souborů. Obsahuje nástroje pro detailní studii sítě umožňující prověřování velikosti sítě a také její kvality, provádí pokročilé objemové síťování hybridní sítě složené z hexa, tetra, pyramidových a prizmatických prvků a 2D síťování s trojúhelníkovou a čtyřúhelníkovou sítí.

Hlavní vlastnosti programu jsou:

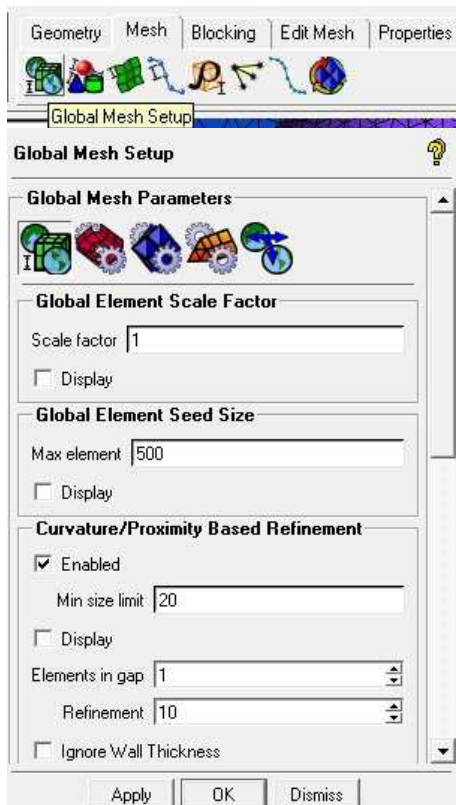
- tvorba trojúhelníkové a čtyřúhelníkové plošné sítě ve 2D,
- tvorba tetra, paramiových a prizmatických prvků ve 3D,
- použití pokročilých metod pro vrstvení prizmat u obtékaných ploch (mezní vrstvy),
- použití Delaunayovy triangulace pro trojúhelníkovou a čtyřstěnovou síť,
- nástroje pro opravu/vylepšení mezní vrstvy,
- nástroje pro manipulaci se zónami.

KAPITOLA 11: MODELOVÁNÍ DYNAMICKÝCH SÍTÍ^[42]

11 Příklad tvorby sítě v programu ANSYS Fluent (model vozidla)

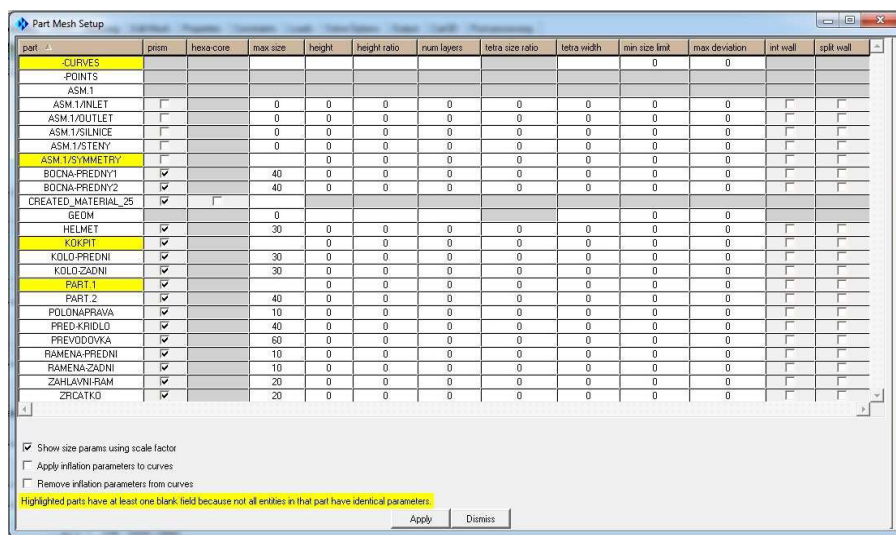
1) Nastavení sítě

Nejdříve pomocí funkce: *Mesh/Global Mesh setup* určíme globální nastavení sítě (Obr. 11.1).



Obr. 11.1: Nastavení největší velikosti sítě.^[42]

V *Global Mesh Setup* nastavíme největší velikost sítě, Max element: 500. To znamená, že tak velká bude buňka sítě v tunelu u vozidla. Tuto síť ještě musíme zmenšit pro přesnější výpočty a můžeme ji nastavit pro všechny plochy zvlášť (Obr. 11.2). Proto jsme museli na začátku rozdělit vozidlo na menší plochy. Zmenšení sítě provedeme pomocí funkce: *Mesh/Part Mesh Setup*. Pro menší části použijeme jemnější síť.

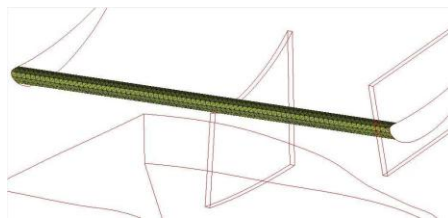


Obr. 11.2: Nastavení části sítě.^[42]

2) Povrchová síť

Na zadním křídle je třeba nastavit nastavit přesnější výpočet povrchové sítě na náběžné hraně (Obr. 11.3), a to pomocí funkce *Compute Mesh/Surface Mesh Only*, kde za input byla zadaná

náběžná hrana.



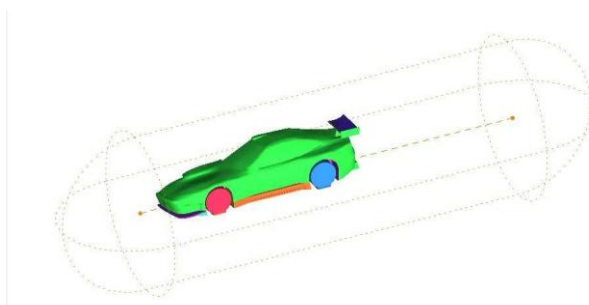
Obr. 11.3: Povrchová síť na náběžné hraně.^[42]

3) Hustota sítě

Pro blízké okolí auta musíme nastavit jemnější síť i před křídlo, a to pomocí funkce *Mesh/Create Mesh Density*.

Hustota sítě (Obr. 11.4) definuje objemovou oblast s volitelnou velikostí sítě, tam kde neexistuje geometrie, např. před křídlem. Můžeme použít i v případě, kdy nemáme aktuální geometrii modelu. Síťové uzly nejsou omezeny na hustotě objektu. Geometrie se mohou vzájemně protínat. Nastavenou hustotu sítě můžeme ještě dále zvyšovat. Hustota sítě se vždy dělí na nejmenší nastavení velikosti. Nastavení velikosti:

- size – velikost buňky sítě,
- ratio – poměr expanze od hustoty objektu,
- width – velikost vybrané oblasti.



Obr. 11.4: Ukázka nastavení hustoty sítě.^[42]

4) Výběr typu sítě

Budeme používat *Tetra/Mixed* objemovou síť a pracujeme metodou *Robust (Octree)* (Obr. 11.5).

Vlastnosti zvolené metody jsou:

- zachová tetra objemy,
- jde použít, když máme komplexní tvar,
- jde použít, když nechceme trávit příliš mnoho času s opravou geometrie.

5) Vlastní postup výpočtu

Stačí nastavit odpovídající velikost sítě na geometrii:

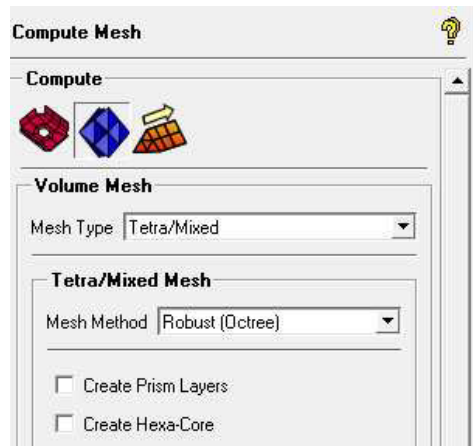
- části geometrie modelu,
- povrchu,
- křivek.

6) Výpočet sítě

Po spouštění *Compute Mesh* můžeme vybírat ze tří možností:

- plošná síť,
- objemová síť,
- prisky.

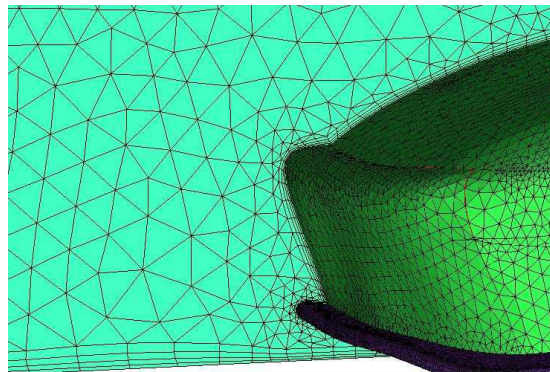
Poté, co zvolíme objemovou síť, vybíráme typ sítě, v našem případě *tetra/mixed*. Dále zvolíme metodu pro síťování (*robust*). Musíme ještě nastavit, aby počítal i předem nastaveným povrchovým sítím.



Obr. 11.5: Nastavení pro výpočet sítě.^[42]

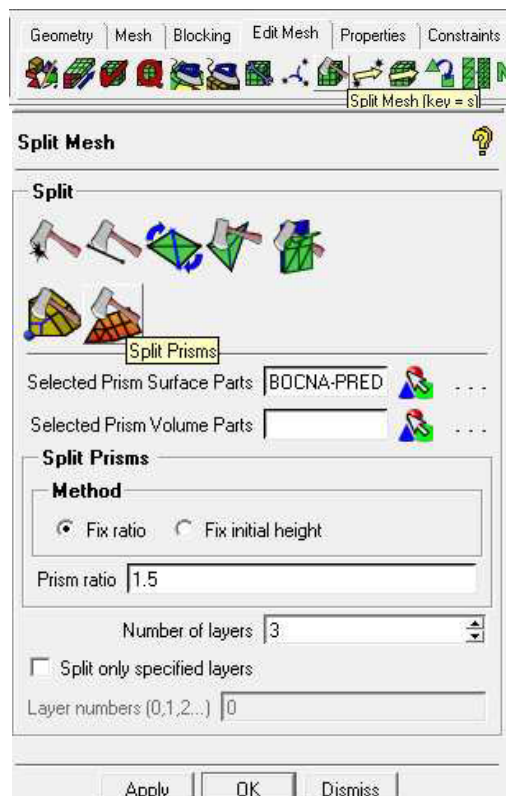
7) Prismata

Pro správnou simulaci proudění v mezní vrstvě musíme nastavit prismatické buňky při síťování. *Mesh/Global Mesh Setup* (Obr. 11.6).



Obr. 11.6: Ukázka prismatic.^[42]

Prismata jsou dobře viditelná na obrázku 11.6, jde o rovnoběžné vrstvy s vozidlem. Prismata ještě musíme opravit: *Edit Mesh/Split Mesh/ Split Prisms* (Obr. 11.7)



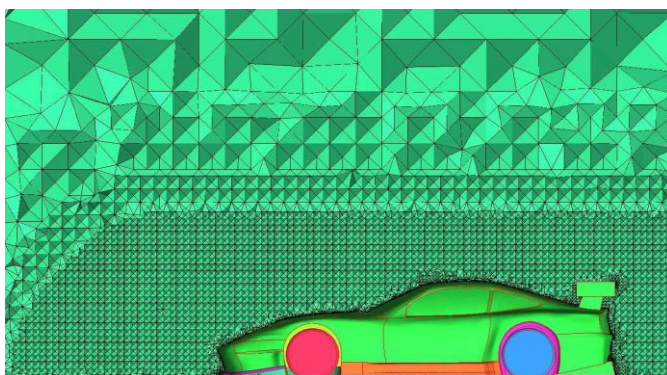
Obr. 11.7: Nastavení rozdělení prismatic.^[42]

Pokud je vyžadováno více prismatických vrstev, je rychlejší vytvoření jedné "tlusté" vrstvy, kterou dále rozdělíme pomocí funkce *mesh editing*. Rozdělení pak probíhá následovně:

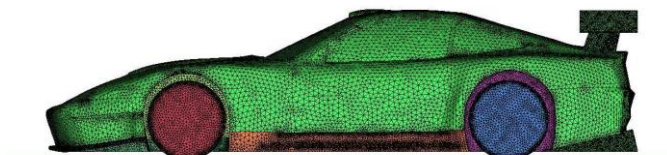
- Fix ratio (Fix poměr): První vrstva je rozdělena tak, aby její výsledné vrstvy používaly daný poměr růstu.
- Fix initial high (Fix počáteční výška): První vrstva je rozdělena tak, že výška buňky u stěny nové vrstvy vychází z dané výšky.

8) Výsledná síť

Na obrázku 11.9 je výsledná objemová síť. Je dobře patrné, jak se vzdáleností zvětšují buňky sítě. Na obrázku 11.10 je znázorněna povrchová síť vozidla.



Obr. 11.9: Objemová síť.



Obr. 11.10: Povrchová síť.

KAPITOLA 12: DEFORMACE PŮSOBENÍM MECHANICKÉ SÍLY A V DŮSLEDKU TEPELNÉ ROZTAŽNOSTI

12.1 Deformace působením mechanické síly a důsledkem tepelné roztažnosti^[43]

Pomocí mechaniky a jejích základů může čtenář získat potřebnou představu o možných napětích v deformovaných součástech vlivem vnějších sil. Strojáři v praxi čelí vznikajícím deformacím a trhlinám a v této chvíli postupují po svém – zvolí pevnější materiál, změní konstrukci, snaží se změnit působící síly aj. Někteří postupují svým citem a jiní se vrací k výpočtům. Následují zkoušky a další hledání optimálních řešení. Téměř vždy je dodatečná úprava spojená se zvýšením ceny.

Stále více se rozšiřující softwarové aplikace v oblasti CAD rázně postoupily z plošného prostředí do prostoru a nabízejí pro práci technika tzv. 3D modeláře. Společnost Autodesk, jako jedna z největších SW společností na světě, jej pojmenovala Inventor a svůj název odvodila od latinského slova invence – důvtip, vtipný nápad, vynalézavost a důmysl – *Autodesk Inventor Professional*. Obsahuje pracovní modul pro technické výpočty CAE metodou konečných prvků společnosti ANSYS Inc.

Co umí modul FEM/MKP v Autodesk Inventor Professionalu:

- 1) aplikovat zatížení a vazby (reakční síly) na součást,
- 2) měnit hrubost sítě – volba hustoty sítě elementů,
- 3) volba typu analýzy – zátěžová, modální nebo obě,
- 4) volba různých režimů v zobrazení včetně barev a jejich rozestupů,
- 5) simulace chování součástí – animace,
- 6) generování zpráv.

Co neumí modul FEM/MKP v Autodesk Inventor Professionalu

- 1) modul neumí výpočty na sestavách, ale jen na jednotlivých modelech,
- 2) umožňuje použití materiálů, které mají stejnorodé vlastnosti ve všech směrech v prostoru a lineárně závislé,
- 3) je určen pro výpočty statických úloh, kdy se zatížení s časem nemění,
- 4) nelze řešit nestabilní úlohy – např. pruty namáhané na vzpěr,
- 5) kompatibilita – modul potřebuje k plné funkci spolupracovat s jiným systémem, kterým je ANSYS nebo jeho jednodušší verze ANSYS Workbench od společnosti ANSYS Inc.

Základní úlohou je tzv. *přímá úloha pružnosti*, která se dá formulovat následovně: pro těleso se známou geometrií, materiálem, zatížením a vazbami k okolí určujeme jeho deformaci a napjatost označované jako napěťová analýza, je předpokladem k hodnocení mezních stavů konstrukce.

Vztahy obecné pružnosti představují v obecné prostorové statické úloze systém 15 rovnic, postačující spolu s okrajovými podmínkami k určení 15 neznámých funkcí posuvů (15 neznámých funkcí proměnných x , y , z), přetvoření a napětí. Jedná se o:

- a) tři posuvy v osách u , v , w ,
- b) šest přetvoření – poměrná prodloužení ε_x , ε_y , ε_z ,
– zkos pravoúhlých stěn γ_{xy} , γ_{yz} , γ_{zx} ,
- c) šest napětí – v osách σ_x , σ_y , σ_z ,
– v rovinách na ně kolmých τ_{xy} , τ_{yz} , τ_{zx} .

Tyto funkce jsou navzájem vázány systémem obecných rovnic pružnosti, které musí být splněny uvnitř řešené oblasti. Jsou to rovnice rovnováhy, rovnice geometrické a rovnice fyzikální. Na hranici řešené oblasti musí pak být splněny předepsané okrajové podmínky.

a) Poměrné prodloužení vlivem osově síly $\varepsilon = \frac{\Delta l}{l_0}$, kde Δl je rozdíl délky a l_0 je původní délka.

b) Zkos $\gamma = \frac{\varepsilon}{h}$, kde h je výška elementu.

c) Rovnice rovnováhy

Tyto rovnice jsou podmínkami rovnováhy základního vnitřního prvku, na který kromě složek napětí působí vnější objemová síla (např. gravitační) o složkách o_x, o_y, o_z [$\text{N}\cdot\text{m}^{-3}$]. Představují vzájemnou vazbu mezi složkami napětí, která musí být splněna vždy bez ohledu na typ materiálu, velikost deformací apod. Uvádím je pro případ statického zatěžování:

$$\frac{\partial \sigma_x}{\partial x} + \frac{\partial \tau_{xy}}{\partial y} + \frac{\partial \sigma_{xz}}{\partial z} + o_x = 0, \quad \frac{\partial \tau_{xy}}{\partial x} + \frac{\partial \sigma_y}{\partial y} + \frac{\partial \tau_{yz}}{\partial z} + o_y = 0, \quad \frac{\partial \tau_{xz}}{\partial x} + \frac{\partial \tau_{yz}}{\partial y} + \frac{\partial \sigma_z}{\partial z} + o_z = 0.$$

d) Rovnice geometrické

Jedná se o vztahy vytvářející vazbu mezi složkami posuvů a přetvoření:

$$\varepsilon_x = \frac{\partial u}{\partial x}, \quad \varepsilon_y = \frac{\partial v}{\partial y}, \quad \varepsilon_z = \frac{\partial w}{\partial z},$$
$$\gamma_{xy} = \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}, \quad \gamma_{yz} = \frac{\partial v}{\partial z} + \frac{\partial w}{\partial y}, \quad \gamma_{zx} = \frac{\partial w}{\partial x} + \frac{\partial u}{\partial z}.$$

e) Fyzikální vztahy

Představují vztah mezi deformací a napjatostí. Opět je uvedeme v nejběžnějším tvaru pro lineárně pružný materiál (v oblasti platnosti Hookova zákona $\sigma = E \cdot \varepsilon$), jehož vlastnosti jsou určeny dvěma nezávislými materiálovými konstantami: modulem pružnosti v tahu – E (Youngův modul pružnosti) a Poissonovo číslem – μ .

Modul pružnosti ve smyku G není nezávislou materiálovou veličinou a lze jej určit:


$$G = \frac{E}{2 \cdot (1 + \mu)}.$$


Ve spojitosti s analytickým řešením je obvyklá diferenciální formulace a deformační nebo silový přístup k výběru nezávislých funkcí pružnosti.


U metody konečných prvků jako numerické metody pak jednoznačně převládá variační formulace a deformační přístup – hovoříme o deformační variantě metody konečných prvků, kde primární neznámé jsou funkce posuvů. Jako u všech numerických metod je u metody konečných prvků zásadní požadavek konvergence – při zhušťování sítě konečných prvků se numerické řešení musí blížit k řešení odpovídajícího spojitého problému.


Pro prvky metody konečných prvků je exaktně dokázána konvergence, a to monotonní konvergence zdola. Znamená to, že vypočtené posuvy jsou při stejném zatížení obecně menší než skutečné, diskretizovaný model je tedy tužší než spojitý. Zvyšováním počtu prvků zvyšujeme poddajnost výpočtového modelu.

f) Zatěžující síly a momenty

 Síla [N] – lze definovat velikost, směr ve třech složkách souřadného systému a umístění. Jejím působištěm může být plocha, hrana nebo bod. Pokud působištěm nemůže být celá plocha, je nutné vytvořit dodatečně malé plošky. Sílu lze umístit i na sražení, do otvoru a na zaoblení. Podmínkou je možnost označit požadovaný útvar na součásti. Na stejnou plochu může působit více sil.


 Tlak [MPa] – lze definovat jeho velikost. Jedná se o sílu na plochu, vyžaduje definování tlaku označení všech ploch, na které tlak působí. Na stejnou plochu může působit více tlaků. Tlak na plochu je zadáván označením konkrétní plochy a volbou velikosti tlaku.


 Moment [N.mm] – lze definovat velikost, směr a umístění. Moment může mít charakter ohybového i krouťícího momentu a je skutečně důležitá jeho orientace vůči ploše.

 Zatížení ložiska [N] – lze definovat velikost, směr a umístění. Aplikuje zatížení s typickým směrem a tvarem působení jen ve směru zatížení.

Zatížení tělesa [m/s^2] – lze definovat orientaci zemské gravitace vůči souřadnému systému nebo ploše. Zrychlení může mít i hodnotou odlišnou od $9,81 m/s^2$, které může vycházet z odstředivých sil. Je nutné opět zadání směru.

g) Reakční síly a momenty – vazby na okolí

 Pevná vazba – je schopná zachycovat normálové a tečné síly. Její účinky lze přirovnat k pevnému připevnění na podložku – přivařením, přišroubováním aj. Můžeme definovat nulový pohyb součásti vůči okolí nebo dovolit malý pohyb, můžeme předepsat dovolenou volnost v určitém rozsahu, můžeme povolit nulový pohyb, nenulový pohyb nebo bez možnosti pohybu.

 Hladká vazba – umožňuje pohyb po pomyslné rovné nebo zakřivené ploše bez omezení rozsahu. Součást s touto vazbou se v ploše dotyku s podložkou může posouvat v rovině plochy a ještě otáčet, nemůže se vnořit do podložky. Vazba zabraňuje pohybu v normálovém směru k ploše. Hladká vazba je někdy nazývána také jako volná nebo ideální.

Poznámka k účelu použití:

V případě nosníků je právě tato vazba nutná pro zachycení radiálních sil, ale umožňující přibližování nebo oddalování reakcí. Pokud bychom toto neumožnili, vznikla by dodatečná napětí v materiálu. Právě z tohoto důvodu jsou mosty, potrubí ale i hřídele uloženy na jedné straně pevně, ale na druhé straně umožňují měnit vzdálenost reakcí.

h) Pevnostní analýza a její možnosti

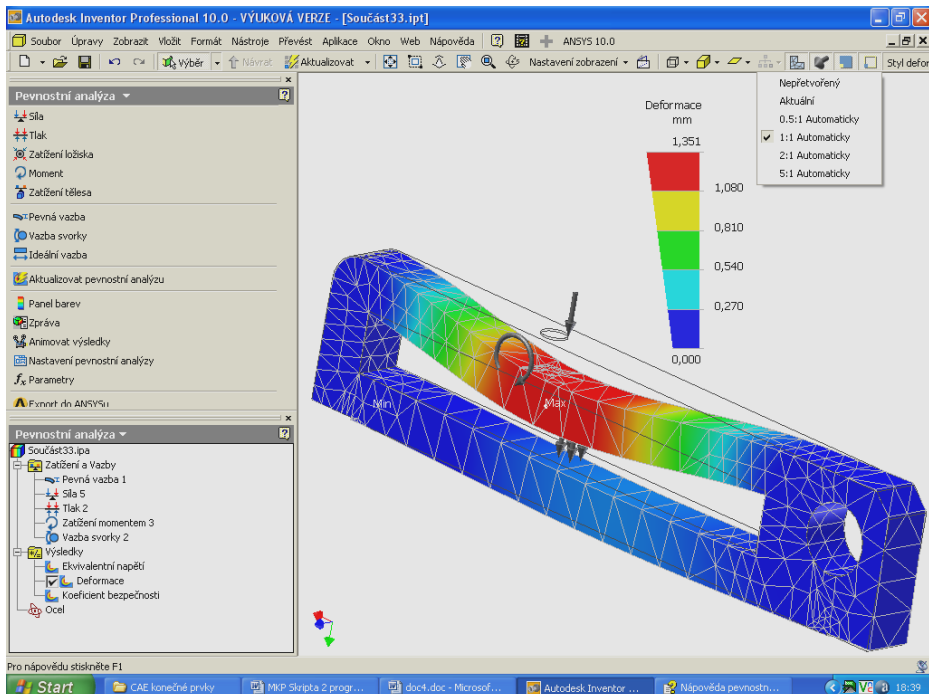
Spustíme vlastní pevnostní analýzu stiskem tlačítka Aktualizovat pevnostní analýzu (Obr. 8.1). Modul FEM od společnosti ANSYS se spustí a doba dokončení práce je závislá na možnostech počítače a komplikovanosti výpočtu.

Základními výsledky pro posouzení součásti je zjištění *ekvivalentního napětí*, *deformací* a *koeficientu bezpečnosti*.

Získané výsledky z analýzy napětí:

- 1) součást je výrazně deformována od osamělé síly. Z hlediska napětí se zřejmě nejedná o hodnověrné napětí, protože je v působišti síly a toto místo jsme definovali my. Můžeme ho zvětšit, zaoblit, aj.,
- 2) místa koncentrátorů napětí jsou v oblasti síly nahoře a dole a v místech ostrých přechodů a hran,
- 3) kritické hodnoty napětí nebyly s ohledem na barevnou stupnici a materiál dosaženy,
- 4) minimální napětí je pod otvorem, protože podepření jsme zvolili v díře a ne v malé plošce pod ní,
- 5) dolní část nosníku je zatížena rovnoměrným tlakem a je souměrně prohnutá,
- 6) vazba svorky nedoformuje otvor.

Při zjišťování deformace se nám zobrazí místa s různou úrovní deformace (Obr. 12.1).



Obr. 12.1: výsledné zobrazení. [43]

Pro možnost předvedení na počítačích bez nainstalovaného produktu Autodesk Inventor Professional lze záznam z pohybu zatěžované součásti uložit jako soubor AVI (*.avi), který je spustitelný z běžných prohlížečů.

12.2 Příklad k řešení

Simulujte pomocí metody sítí změnu teploty v tyči vyrobené z hliníku, z oceli, z plastu, ze dřeva. Konce tyčí jsou udržovány na různých, ale konstantních teplotách. Tepelné ztráty z ostatních částí tyče zanedbejte. Úlohu řešte také v programu COMSOL Multiphysics.

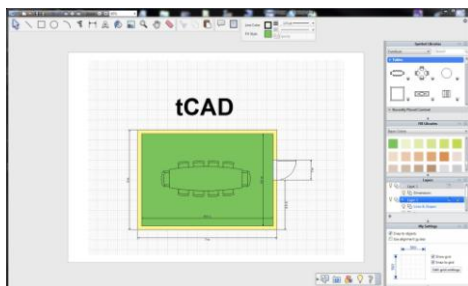
KAPITOLA 13: FREEWAROVÉ APLIKACE – ALTERNATIVY KE KOMERČNÍM PRODUKTŮM^[44]

Plné verze všech CAX systémů jsou obvykle příliš drahé, ale v nabídce webových stránek lze najít programy, které jsou volně šiřitelné a mnohé z nich jsou i plně funkční.

Free CAD – přehled některých vybraných bezplatných CAX systémů a prohlížečů

13.1 Vybrané 2D CAD programy

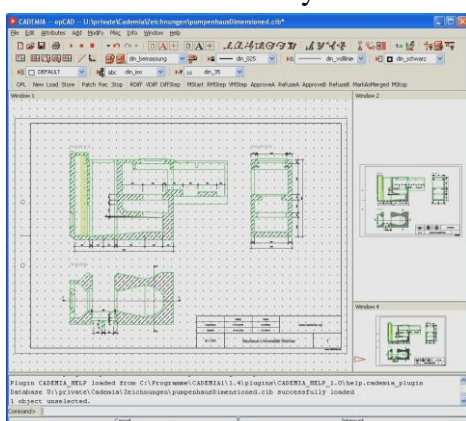
- Project Cooper



SW určený pro kreslení 2D plánů a schémat ve zvoleném měřítku pomocí základních kreslicích nástrojů, symbolů a příkladů kreseb. Má správu hladin, barev, typů čar a šrafování. Výkres můžeme i okótovat. I když je práce s ním jednoduchá a intuitivní, postrádáme důležité editační nástroje. Obsahuje-li výkres objekty ve výkresovém prostoru, je načtení souboru uloženého ve formátu DWG v AutoCADu problematické. Názvy hladin

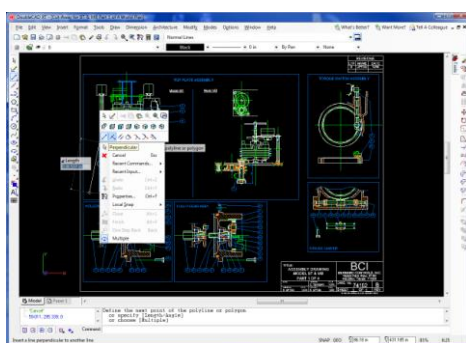
jsou dost nepřehledné a program jich vytváří zbytečně moc. Program se hodí pro uživatele jako vstup do světa CADu.

- CADEMIA Community



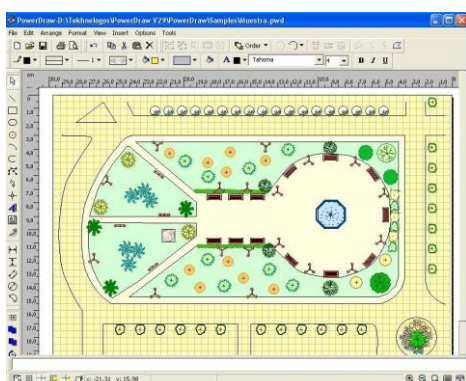
Tento software byl původně navržený pro školní účely a vývoj. Je modulárně řešený a nabízí množství možností pro rozšíření funkčnosti. Číst a zapisovat dokáže ale jen do vlastního formátu. Je napsaný v jazyku JAVA a tak je nezávislý na nainstalovaném operačním systému. Jako plně funkční CAD systém v pravém slova smyslu však velké uplatnění nemá.

- DoubleCAD XT – freeware



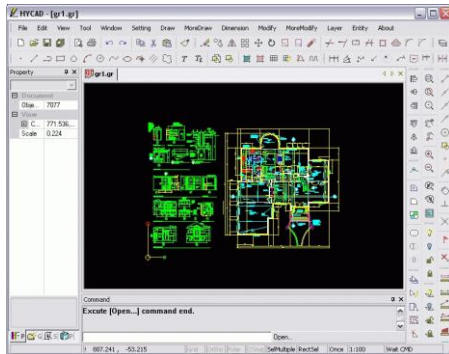
Tento software je velmi podobný AutoCADu a i když je způsob práce přece jen trochu rozdílný, uživatel zvyklý používat AutoCAD se v něm dokáže velmi rychle zorientovat. SW DoubleCAD XT výkresy uložené v AutoCADu spolehlivě otvírá, zobrazuje a ukládá. Používá fonty SHX, chybí typy čar a možnost vlastního šrafování, obsahuje ale postačující výběr šrafovacích vzorů. Nevýhodou je, že program nemá možnost jakéhokoliv rozšíření pomocí LISPU či vývojářských metod. Free verze nemá žádná omezení a je dobře použitelná v reálné praxi.

- PowerDraw



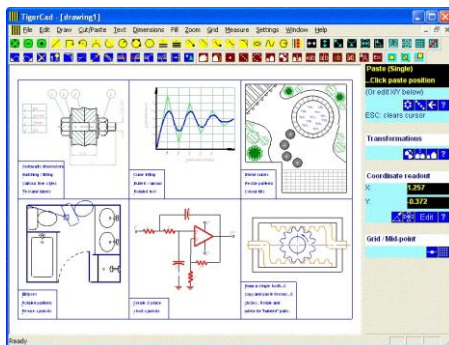
PowerDraw je jednoduchá 2D vektorová aplikace. Disponuje nástroji na 2D kreslení a editaci. Umožňuje vytváření bloků a jejich ukládání do kružnice a mnohé další funkce. Podporuje import do formátu WMF a bitmapy.

- **HYCAD**



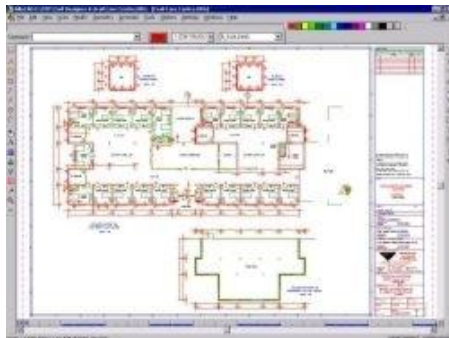
HYCAD je jednoduchý a zajímavý CAD program, má mnoho funkcí, je lehce použitelný pro jednoduchou tvorbu náčrtů, ovládání je dost podobné AutoCADu, neumí načítat DWG formát a formát DFX načítá s problémy. Umí automaticky kótovat, ale nastavit automat na kótování je problematické. Neumožňuje vkládání do bloku a následné uložení/vyvolání jako blok. Ve verzi 5.35 není možno používat relativní rozměry při kreslení (všechny vzdálenosti musí být počítány od 0,0) a neumí ani kresbu pod zadaným úhlem pomocí zadávání délky a úhlu třeba úsečky.

- **TigerCad**



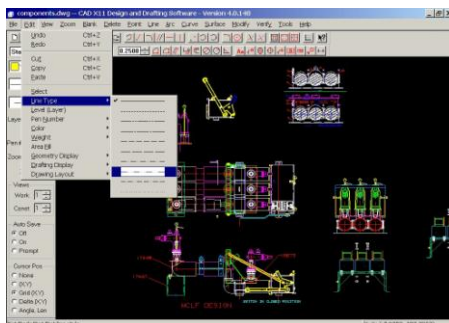
Kompletně bezplatné řešení pro jednoduché technické výkresy. Je ideální pro návrhy zahrad, případně pro jiné technické obory, jako je strojírenství, elektrotechnika, plány místností či obecné diagramy. Nefunkční v operačním systému Win7 – nelze zadávat příkazy a bliká obrazovka.

- **AllyCAD FREWARE**



AllyCAD je profesionální 2D CAD systém, který je známý výkonem a jednoduchostí použití. Je oproti jiným freewarovým již popsaným systémům velmi rychlý při překreslování, uchopování nebo načítání výkresů. AllyCAD plně podporuje čtení a zápis AutoCADovských systémů CWG, DXF, Caddie CEX a Arcview SHP. Tato verze nabízí plnou funkčnost s jediným omezením: velikost ukládaného souboru může dosahovat maximálně 150 kB.

- **CAD X11 FREE**



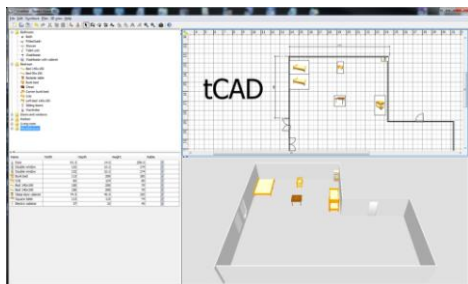
Je ideální pro kreslení a plošné modelování. Obsahuje 3D drátěné modelování ploch, DXF, 2D kreslení s kótováním ANSI, ISO a JIS, geometrické tolerance, podporu TrueType fontů, dynamickou rotaci, násobné pohledy, kontrolu dat, geometrickou analýzu, šrafovy, hladiny, podporu tiskáren ve Windows a PDF výstup. Při stahování je nutné zažádat e-mailem oficiálního poskytovatele o heslo.

- **BlueCAD**

Kreslení v BlueCAD je jednoduché a díky jeho intuitivnímu uživatelskému rozhraní i rychlé. Obsahuje více jak 100 funkcí – šrafovy, texty, bloky, typy čar, hladiny, asociativní kóty, import/export DXF, a další. Navíc když chceme přidat nové vlastnosti do systému a rozšířit jeho funkce je možné vytvářet nové příkazy s použitím programovacího jazyka C++.

13.2 Vybrané 3D CAD programy

- Sweet Home 3D



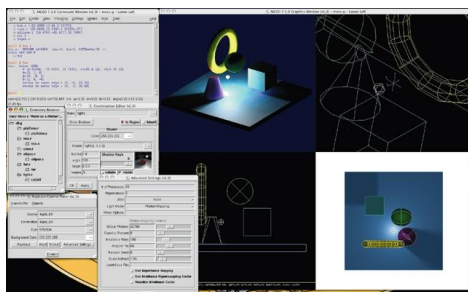
Sweet Home 3D je volně šiřitelný program, který pomáhá při navrhování designu interieru. Nábytek se umísťuje po domě ve 2D plánu s okamžitým 3D náhledem. Práce s ním je sice celkem pohodlná, ale dost pomalá. Můžete kreslit zdi, vkládat okna a zařízení. I když lze používat pouze existující zařízení – rozměry lze přizpůsobit. Díky tomu je to celkem použitelný sw na volné chvíle a dá se s ním skutečně navrhnout základní design interieru. Snad ještě informace že obsahuje i funkce pro základní kótování.

- Breeze Designer



Breeze Designer je 32-bitový bezplatný 3D modelovací a navrhovací nástroj. Umožňuje 3D modelování těles a částečně i ploch. Poskytuje i základní animaci. Pro rendrování se však doporučuje používat jiné free nástroje.

- BRL-CAD

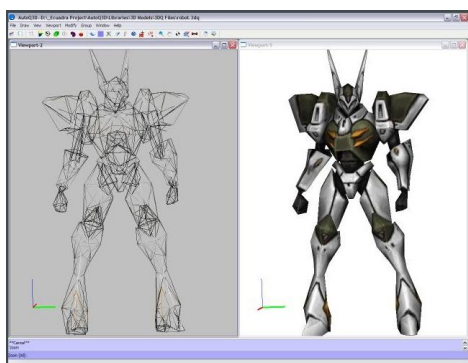


BRL-CAD je výkonný konstrukční 3D objemový modelovací systém, který obsahuje interaktivní editor geometrie, ray tracing podporu pro rendrování a geometrickou analýzu a další progresivní nástroje.

- Rhinoceros Trial Cz

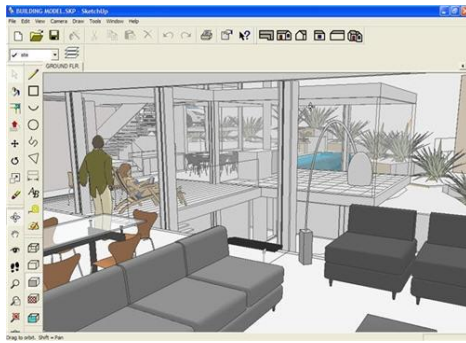
Rhinoceros® je silný programový nástroj pro 3D modelování a koncepční design. K reprezentaci dokonale hladkých křivek a ploch využívá NURBS geometrii. NURBS (NonUniform Rational B-Splines) je matematická reprezentace 3D geometrie, která umožňuje přesně popsat jakýkoliv tvar od jednoduché 2D čáry, oblouku, kružnice nebo křivky až po nejsložitější 3D plochy a tělesa organických tvarů. Díky jejich flexibilitě a přesnosti můžete používat NURBS modely v libovolném procesu od ilustrace po animaci nebo sériovou výrobu. Trial verze umožňuje 25 uložení (počet spuštění není omezen). Poté přestane ukládat a exportovat, ostatní funkce zůstanou zachovány.

- AutoQ3D Community



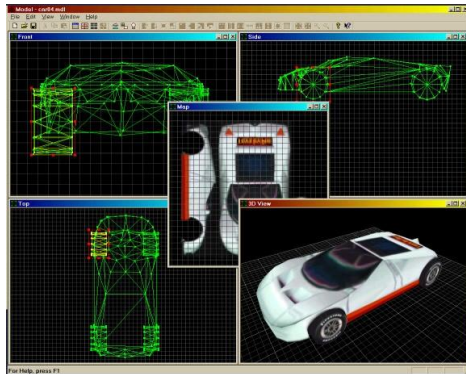
Kompletní aplikace pro 3D modelování. AutoQ3D může být bezplatně používán pro zobrazování, vytváření a publikování ve 3D zobrazení. Import do formátů OBJ, 3DS, DXF a MD2 umožňuje jen placená verze.

- Google SketchUp



Google SketchUp v6 je free verze pro nekomerční použití. Slouží pro jednoduché prostorové modelování. Ovládání je intuitivní a velmi jednoduché. Umožňuje jednoduše mapovat textury a nasvítit objekt. Dokáže importovat CAD soubory 3DS, DWG a DXF. Exportovat do těchto formátů však už neumí.

- Ohmikron 3D Model Editor



Ohmikron 3D Model Editor byl navržen pro zvýšení rychlosti 3D grafiky a pro lepší import 3D modelů do nových SW aplikací. Nabízí přesnou kontrolu nad mapováním textur každého vytvořeného 3D objektu v čase. Místo vytvoření modelu a následné aplikace textur na model, vytváří se model kreslením přímo na texturu. Textury jsou uloženy v souborech modelu a není problém oddělit modely od textur. Je možné rychle vytvářet i objemy vysunutím a rotací souřadnic mapy automaticky aplikovanými během vytváření. Tento software je výkonný, jednoduchý na ovládání a je určený návrhářům

designu. Při instalaci je nutné zadat registrační klíč, který je možné obdržet od společnosti Ohmikron a software bude fungovat 90 dní, pak je nutné požádat o nový registrační klíč.

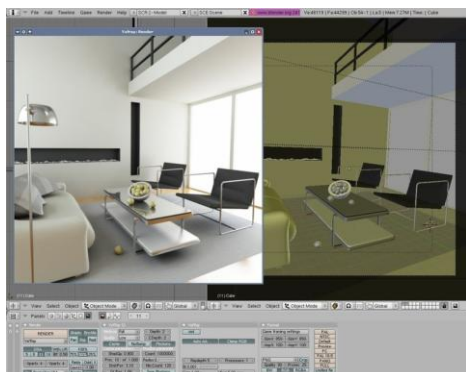
13.3 Vizualizace a animace

- TrueSpace



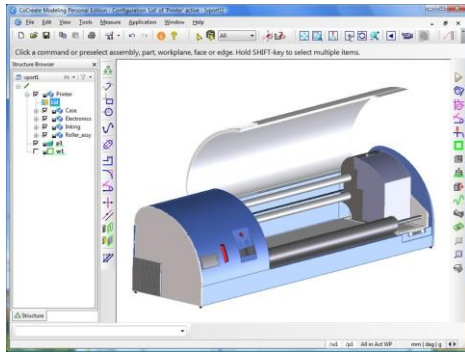
Po jednoduché registraci je možné trueSpace 7.6 stáhnout zcela zdarma. Jeho kvalita sice samozřejmě nedosahuje velikánů jako XSI, Maya či 3ds max, nicméně spokojeni budou všichni, kteří si chtěli trueSpace třeba jen vyzkoušet. Aplikace je plně vybavená, výkonná pro tvorbu realistické 3D grafiky. Umožňuje modelování, morphing, aplikaci textur a světla, animace a rendrování 3D objektů v reálném čase. Podporuje export grafiky do her a Virtual Earth (BHV, SCN, COB, CAN, WRL 1.0 & 2.0, DXF, ASC, 3DS, STL), import snímků řady formátů (BVH, DXF, SCN, COB, SOB, WRL 1.0, GZ, WRZ, DXF, 3DS, ASC, PRJ, LWO, OBJ, LWB, IOB, GEO, STL).

- Blender



Blender je multiplatformní open source aplikace zaměřená na vytváření 3D modelů, animací, postprodukční činnost a v neposlední řadě interaktivních aplikací. Disponuje plně fotorealistickým renderingem a jedná se skutečně o jedinečné a komplexní řešení vhodné i pro profesionální použití. I když se jedná o free verzi, je plně funkční a kompletní.

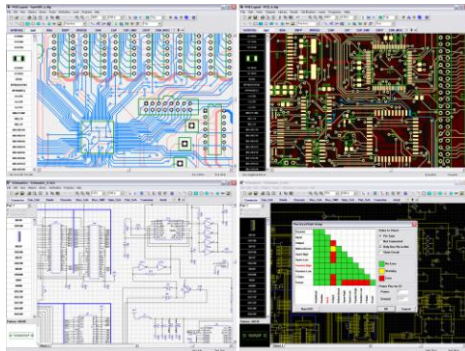
- CoCreate Modeling Personal Edition



Je plně funkčním strojírenským 3D CAD řešením, jež je zajímavé např. proto, že na rozdíl od většiny ostatních CADů nejsou v něm vytvářené modely vázány na historii vzniku. Tím pádem tento program mimochodem nabízí širokou interoperabilitu s daty pocházejícími z jiných aplikací. Jedinou nevýhodou bezplatně uvolněné „osobní edice“ je fakt, že tato dovolí konstruovat sestavy čítající maximálně šedesát dílů a výstupy ukládá ve formátu nekompatibilním s komerční verzí (při zakoupení komerční licence však lze tyto soubory odemknout a používat i v plné verzi). Import: IGES, STEP, DWG, DXF. Export: VRML, STL.

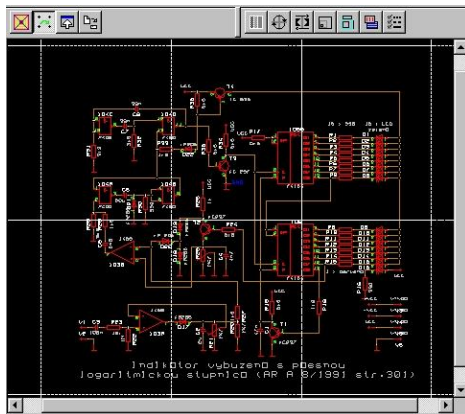
13.4 Vybrané E programy

- DipTrace Freeware



Program je určen pro návrh plošných spojů. Obsahuje výkonný a přesný autorouter, který dokáže spoje navrhnut s velkou přesností. Vytvořené obrazce je možné ručně upravit podle svých požadavků. Umožňuje i kreslení elektrotechnických schémat s následným exportem spojů do návrhového programu. Obsahuje všechny komponenty komerční verze, ale pracuje s omezením na 250 přípojných míst součástek a nepodporuje export do DFX. Tato licence nesmí být používána na komerční účely.

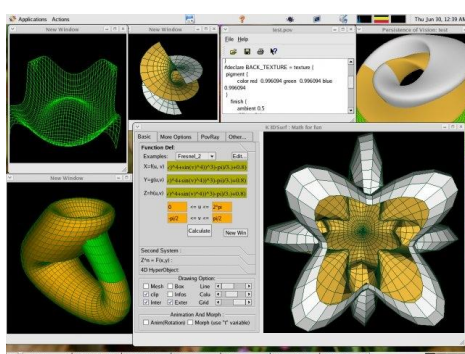
- LSD 2000



Návrhový systém pro elektroniku LSD2000 je software určený pro návrh elektrických zařízení a vytváření technické dokumentace ze všech oblastí elektrotechniky a elektroniky. Umožňuje pohodlné kreslení slaboproudých i silnoproudých schémat, tvorbu potřebných schematických značek a tvorbu pomocných symbolů. V návaznosti na schéma lze vytvářet návrh spojového obrazce pro výrobu desek plošných spojů, doplňovat a upravovat potřebné patice, technologické symboly. Nakreslená schémata i spojové obrazce je možné zpracovávat řadou postprocesorových programů pro různá zařízení různých typů. Volně šiřitelná verze umožňuje načtení, prohlížení a

zpracovávání libovolně rozsáhlých návrhů. Pokud však návrh překračuje kterýkoliv z uvedených limitů (40 objektů, 10 bloků, ...) je zablokována funkce ukládání návrhu na disk do souboru. Uživatel je o této skutečnosti informován periodicky se objevujícím informačním oknem.

- K3DSurf



K3DSurf je program určený pro tvorbu matematických modelů ve třech, čtyřech, pěti a šesti rozměrech. Tyto matematické objekty zobrazuje s implicitními nebo explicitními parametrickými rovnicemi, kterými mohou být křivky nebo plochy. Vizualizuje i animuje. Generuje soubory sítí, které zobrazují tvar matematického modelu.

13.5 Další freewarové aplikace

Free2Design, A9CAD, APM Graph LV, FelixCAD LT, CadStd Lite, JFDRAW, CADE, Draft IT v2, ThanCAD, SagCAD, Tlinea, VectorEngineer Quick-tools, JustCad, Autodesk AliasStudio Personal Learning Edition 2009, Strata 3D, Silo Learning Version 2.1.1, Wings3D, myStudio, Varkon 1.19C, Autodesk Maya Personal Learning Edition 2009, K-3D, DesignWorkshop Lite, Behemot Graphics Editor, Freyja 3d, POV-Ray, ppModeler, OGRE 1.6 RC1, 3D Canvas, Alibre Design Xpress, KOPMAS-3D Light CZ, gCAD 3D, Open CASCADE, DeskArtes Expert Demo, Solid Edge 2D Drafting, Minos, PowerSHAPE-e 8080, APM Structura 3D LV – FEM analýza, eMachineShop, MEDUSA4 Personal, OCTREE, Allplan BIM – studentská verze, Archicad pro studenty, MicroStation Powerdraft XM pro studenty, CYCAS, PCB123, EAGLE Light Edition, ExpressPCB, Kicad 2008-08-25C, OrCAD – Demo, FreePCB, Pad2Pad, eCabinet Systems, GEOTOPIA Generátor reliéfů map, GRASS GIS, Surface Evolver, aj.

13.6 Výhody a nevýhody značného množství free CAx aplikací

Nevýhody

- ukládání všech dat (sítě, animace, textury,...) je možné jen do jednoho souboru či jediného formátu,
- neobsahují kompletní knihovny nebo nemají možnost používat vlastní aplikace či návrhy,
- některé aplikace neumožňují uložení hotového projektu,
- nejsou podporované některými platformami OS,
- některé vyžadují při instalaci zadání klíče či hesla, o které je nutné zažádat distributora,
- některé jsou po instalaci plně funkční jen omezenou dobu, pak se stávají nefunkčními,
- některé neumožňují tisk či při tisku stránku deformují nebo přetiskují vodoznakem,
- některé neumožňují tvorbu realistické 3D grafiky v reálném čase,
- existuje jich velké množství, každý něco umí/neumí a je komplikované se v nich orientovat.

Výhody

- jsou bezplatné a nevyžadují registraci,
- po vyzkoušení free verze je možné zakoupit registrační klíč pro plnou aktivaci aplikace,
- vždy si uživatel může najít takovou aplikaci, která splní jeho očekávání.

KAPITOLA 14: ZÁVĚREČNÁ HODINA – SHRNUTÍ, OPAKOVÁNÍ A PREZENTACE STUDENTSKÝCH PRACÍ

Při této závěrečné přednášce je vhodné celkově shrnout probranou látku, vyzdvihnout přednosti prezentovaných vybraných softwarových aplikací, zkompletovat poznatky o výhodách a nevýhodách freewarových programů a zdůraznit použitelnost nabytých poznatků.

V předcházejících přednáškách a cvičeních si studenti vybrali či sami navrhli náměty na samostatnou práci. Tyto práce v žádném případě neřeší matematicky, ale vyžaduje se po nich komplexní náhled na problematiku těchto námětů v návaznosti na probranou látku a nástin vhodných řešení, tzn. že práce by měla obsahovat obecný rozbor řešené problematiky.

14.1 Nástin řešení seminární práce

Návrh a simulace RLC obvodu^[45]

- Formulace problému

Použit systém vhodný pro simulaci vybraných jevů v elektrických obvodech obsahujících rezistory, cívky, kondenzátory a zdroje elektrického napětí. Systém musí umožňovat uživatelský návrh zapojení a jeho následnou časovou analýzu v reálném čase, prováděnou pro každou změnu charakteristických hodnot u použitých prvků obvodu. Výstupem systému budou informace o napěťových a proudových poměrech panujících v daném zapojení.

- Vytvoření modelu

Při vytváření počítačového modelu je potřeba nejdříve zformulovat zjednodušený popis zkoumaného systému a charakterizovat závislosti výstupů systému na jeho vstupech. Následně vzniklý abstraktní model zapsat formou programu. Model obvodu je popsán schématem a pomocí datové struktury logického zapojení obvodu. Pro počítačové zpracování a samotný výpočet je nezbytný popis logického členění schématu, při kterém se ignoruje skutečné rozložení prvků obvodu na ploše. Pro úspěšnou analýzu postačuje znalost parametrů jednotlivých prvků obvodu a znalost vzájemného propojení mezi prvky. U každého prvku obvodu se tedy musí uchovávat jeho poloha na kreslicí ploše, jednoznačný identifikátor, charakteristické vlastnosti a způsob propojení.

- Analýza

Metodu analýzy je možné chápat jako konkrétní postup od sestaveného modelu obvodu, většinou ve formě soustavy rovnic, po získání cíle analýzy, tj. určité vlastnosti originálního systému. Analýza zahrnuje rozbor a ověření vlastností daného modelu. Typy výsledků analýzy mohou být v různých tvarech, které závisí na použitých algoritmech a míře jejich abstrakce. Symbolické výsledky analýzy jsou ve formě matematických vzorců s proměnnými, čímž je přehledně zobrazena míra závislosti výsledku na potřebných proměnných a v případě potřeby je možné dosáhnout semisymbolického i numerického výsledku dosazením patřičných hodnot za proměnné.

- Simulace

Prvotním cílem simulace časové analýzy je zjištění časových průběhů signálů v obvodu, přičemž by měla umožnit zobrazení všech možných časových průběhů, které lze odvodit na základě napětí a proudů, a dále zaznamenávat jednorázové přechodové děje. Reakce obvodu na daný budicí signál závisí nejen na tomto signálu, ale i na počátečních podmínkách, v nichž se obvod nachází v okamžiku připojení budicího signálu. Tyto počáteční podmínky jsou implicitně nastaveny na nulu a teprve při přerušení časové analýzy tvoří neznámé veličiny počáteční podmínky pro další průběh simulace. Na samém začátku simulace dochází k optimalizaci zapojení, redukci pro výpočet nadbytečných prvků obvodu a tím k zjednodušení výsledného logického zapojení. Program může rozpoznat krátký obvod v zapojení a tak část obvodu, která nemůže ovlivnit chování celku, není zahrnuta do výpočtu.

- Konečné zobrazení výsledků

Výsledek může být znázorněn pomocí grafu, zobrazené průběhy napětí a proudu neslouží k přesnému zjištění velikosti vykreslených průběhů na jednotlivých prvcích obvodu, nýbrž

k porovnání fázových posuvů napětí a proudu u jednotlivých prvků obvodu. Pro výstup slouží export schématu do souboru v některém možném formátu (např. BMP). Další formou výstupu je export vykreslovaných průběhů zobrazených signálů.

- Závěr

Do zapojení je možné vkládat pouze ideální součástky a výsledné zapojení lze přiblížit reálnému pouze omezeným způsobem. Ke každé ideální součástce je možné připojit další ideální součástky, které popisují parazitní jevy v daném reálném prvku obvodu, např. sériové připojení ideální cívky popisující indukčnost vývodů reálného rezistoru k ideálnímu rezistoru. Při časové analýze nebývají uvažovány přechodové jevy vznikající při nabíjení kondenzátoru nebo při změně velikosti proudu tekoucího cívkou.

Zde by stálo za úvahu najít a popsat vhodné applety znázorňující řešený problém. Jako ukázkou uvádím následující www adresy:

http://www.walter-fendt.de/ph14cz/combrlc_cz.htm

<http://www.lon-capa.org/~mmp/kap23/RCL/app.htm>

<http://www.cco.caltech.edu/~phys1/java/phys1/lrc/index.html>

<http://phet.colorado.edu/en/simulations/category/physics>

http://www.bartol.udel.edu/~seckel/courses/Physics%202008/Lecture%20Notes/chap%2033/AF_3313.swf

14.2 Další návrhy seminárních prací

- Seznámení se s problematikou modelování dopravy, nalezení problémových míst a jejich diagnostika. Kdy se proces modelování a simulace využívá při hledání potenciálních problémových míst na dopravní síti a jak umožňuje přípravu jejich možného řešení, posuzování dopadu dopravních staveb na životní prostředí, ověření správnosti nebo opodstatněnosti daného návrhu.
- Protipovodňová opatření a jejich účinnost. V důsledku extrémních povodní z nedávných let a jejich medializací vzrostl zájem společnosti na zmírnění škod, které by v budoucnu mohly povodně opět způsobit. Pro návrh a dimenzování protipovodňových opatření na specifické podmínky jejich umístění lze využít nástroje hydrologického modelování. Cílem je objasnění vlivu a určení účinnosti různých scénářů protipovodňových opatření během extrémních srážko-odtokových situací na zvoleném povodí.
- Předpověď počasí je z vědeckého hlediska jedním z nejsložitějších fyzikálních úkolů. K jeho řešení existuje řada metod, ale žádná z nich není stoprocentní. V současnosti existují předpovědi jak všeobecné, tak speciální pro různé obory hospodářství, námořníkům, letcům, stavařům, aj. Předpovědi pro všeobecnou potřebu obsahují minimum kvantitativních informací a podávají charakteristiku očekávaného vývoje počasí.
- Simulace mechanického namáhání rotující součástky,
- grafická vizualizace hluku,
- kartografické metody modelování měst,
- 3D modely a jejich využití pro modelování městského klimatu,
- modelování tepelných ztrát, aj.

ZÁVĚR

Diplomovou práci jsem pojala jako koncepční přehled o simulacích v rámci softwaru pro počítačové modelování a technické výpočty. Není to učebnice počítačového modelování, nejde o samotné vytváření a modelování rovnic, ani to není návod pro práci s jednotlivými softwarovými prostředky. Jde v ní o to, aby její čtenář získal základní náhled na problematiku počítačového modelování, aby byla pomůckou pro rozvíjení základních znalostí o softwaru používaného pro technické výpočty.

Kapitoly diplomové práce jsem seřadila tak, aby se čtenář zorientoval v nabídce softwarových produktů dostupných na trhu a dokázal zvolit alternativu, která bude vhodná pro jeho potřeby. Předpokládám, že pro budoucí práci budou absolventi používat plnohodnotné profesionální verze uváděných programů a ne jen jejich freewareové aplikace. Měli by ale znát návaznost jednotlivých programovacích segmentů, aby se vyvarovali nutných vedlejších přechodových operací, které by jejich práci mohly zkreslit nebo zpomalit. Freewareové verze, na které ve své práci odkazují, jsou dostačující na to, aby studenti získali zkušenosti, jak se v tom kterém produktu pracuje.

Pro vypracování této práce bylo nutné stáhnout velké množství dat z webu a prostudovat také dostupné materiály z knihovny. Na webu se nachází velké množství studijního materiálu, existuje nepřehledné množství různých stránek pojednávajících o problémech řešených nebo zmiňovaných v této práci. Tyto stránky nabízí nejen přímé články a příspěvky publikované prostřednictvím tohoto média, ale je možné stáhnout i různé příručky, skripta a akademické práce. Při zpracovávání těchto podkladů jsem se setkala jak s velmi precizně a pečlivě vytvořenými příspěvky, tak bohužel i s informacemi zavádějícími nebo i přímo nepravdivými, kdy jejich zpětné ověření bylo časově náročné. Bylo třeba velmi pečlivě vybrat i autory těchto příspěvků a je dobře, že web nabízí i možnost zjistit hodně o těchto autorech. Proto mohu jen doporučit, aby před vlastním použitím dat stahovaných z webu si student ověřil relevantnost takového zdroje.

Byla bych ráda, kdyby tato diplomová práce posloužila účelu, pro který byla zpracována, tedy aby pomohla při výuce studentů, a doufám, že moje snažení bude ku prospěchu a přínosem pro všechny čtenáře.

POUŽITÁ LITERATURA A WWW STRÁNKY

- [1] Počítačové modelování. *Fyzikální webové stránky - webFyzika* [online]. [cit. 2012-03-24]. Dostupné z: <http://webfyzika.fsv.cvut.cz/> .
- [2] NOVOTNÁ, Markéta. *Počítačové modelování znečištění v přírodě s využitím Maple*. Brno, 2010. Bakalářská práce. Masarykova univerzita v Brně. Vedoucí práce prof. RNDr. Jiří Hřebíček, CSc.
- [3] HŘEBÍČEK, Jiří, Zdeněk POSPÍŠIL a Jaroslav URBÁNEK. *Úvod do matematického modelování s využitím Maple*. Vyd. 1. Brno: Akademické nakladatelství CERM, 2010. 118 s. ISBN 978-80-7204-691-1.
- [4] Modelování 2D a 3D. In: *Ulož.to* [online]. 2011 [cit. 2012-06-22]. Dostupné z: <http://www.uloz.to/xhpD7XJ/14-modelovani-2d-a-3d-docx> .
- [5] Matlab, laboratoř nejen pro matematiky. PÍŠA, Pavel. *Cmp.felk.cvut.cz* [online]. 2012 [cit. 2012-06-22]. Dostupné z: http://cmp.felk.cvut.cz/~pisa/Public/ST_matlab.html .
- [6] MATLAB. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2012-06-22]. Dostupné z: <http://cs.wikipedia.org/wiki/MATLAB> .
- [7] Základy informatiky – Počítačové modelování v Matlabu; Ing. Jiří Novák, Ph.D., RNDr. Ivana Pultarová, Ing. Pavel Novák; ČVUT v Praze 2005; ISBN 80-01-03308-2.
- [8] Matlab Central. THE MATHWORKS, Inc. *Mathworks.com* [online]. 1994 - 2012 [cit. 2012-06-20]. Dostupné z: <http://blogs.mathworks.com/community/2008/07/21/graphical-user-interfaces-are-just-plain-better/>.
- [9] Aqua Phoenix. *Aquaphoenix.com* [online]. 2012 [cit. 2012-06-22]. Dostupné z: <http://www.aquaphoenix.com/lecture/matlab9/page2.html> .
- [10] Using a Matlab. *Omatrix.com* [online]. 2012 [cit. 2012-06-22]. Dostupné z: <http://www.omatrix.com/mlfunction.html> .
- [11] Popis systému Matlab a jeho toolboxů. *Www.352.vsb.cz* [online]. 2012 [cit. 2012-06-22]. Dostupné z: http://www.352.vsb.cz/uc_texty/autorizmatlab/kap1.html .
- [12] Třidimenzionální grafy. *Eamos.pf.jcu.cz* [online]. 2012 [cit. 2012-06-22]. Dostupné z: http://eamos.pf.jcu.cz/amos/kat_fyz/externi/kat_fyz_1272/3d_graf/3d_grafy.htm
- [13] HRACH, Rudolf. *Počítačová fyzika I.* Ústí nad Labem: Pedagogická fakulta UJEP, 2003.
- [14] DOLEŽALOVÁ, Jarmila. *Pravděpodobnost a statistika*. Ostrava: Vysoká škola báňská – Technická univerzita, 2005.
- [15] VIRIUS, Miroslav. *Aplikace matematické statistiky: metoda Monte Carlo*. Vyd. 3. Praha: ČVUT, 1998. ISBN 80-010-1779-6. Skripta. ČVUT.
- [16] GUŠTAR, Milan. *Generování náhodně proměnných veličin v metodě Monte Carlo*. 15.3.2000. ISBN 80-02-01344-1.
- [17] Simulace.pdf. In: *Aldebaran.cz* [online]. 2012 [cit. 2012-06-25]. Dostupné z: <http://www.aldebaran.cz/produkty/tpla/ukazky/simulace.pdf>
- [18] TESAŘ, Jiří a Petr BARTOŠ. VYUŽITÍ MATLABU JAKO MOTIVAČNÍHO PROSTŘEDKU. In: <http://dsp.vscht.cz/> [online]. 2008 [cit. 2012-06-29]. Dostupné z: http://dsp.vscht.cz/konference_matlab/MATLAB08/prispevky/111_bartos.pdf
- [19] Blender3D.cz. <http://www.blender3d.cz/> [online]. 2005 [cit. 2012-07-20]. Dostupné z: <http://www.blender3d.cz/drupal/?q=charakteristika>
- [20] Computer aided design. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001-2012 [cit. 2012-07-19]. Dostupné z: http://cs.wikipedia.org/wiki/Computer_aided_design

- [21] AutoCAD. In: Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2001-2012 [cit. 2012-07-21]. Dostupné z: <http://cs.wikipedia.org/wiki/AutoCAD>
- [22] SURFCAM a CAD systémy CADKEY a SolidWorks. MM Průmyslové spektrum. 2003, č. 1. Dostupné z: <http://www.mmspektrum.com/clanek/surfcam-a-cad-systemy-cadkey-a-solidworks.html>
- [23] SolidWorks: 3D CAD navrhování. *SolidVision* [online]. 2012 [cit. 2012-08-03]. Dostupné z: <http://www.solidvision.cz/solidworks/>
- [24] SolidWorks. 3E PRAHA ENGINEERING [online]. 2009 - 2012 [cit. 2012-08-03]. Dostupné z: <http://www.3epraha.cz/SolidWorks/proc-solidworks/vyhody-systemu-solidworks>
- [25] MINGS, Josh. 3D Scanning Goodness. More Fingers For Your Nubbin's Via SolidWorks. *Solidsmack.com* [online]. 2009 [cit. 2012-08-03]. Dostupné z: <http://solidsmack.com/design/3d-scanning-solidworks-medical-prosthetics/>
- [26] Metoda konečných prvků: panel 1 - 3. In: <Http://www.345.vsb.cz/> [online]. 2012 [cit. 2012-06-27]. Dostupné z: http://www.345.vsb.cz/jirihruby/Vmt/MKP_panel1-3.pdf
- [27] Metoda konečných prvků. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001 [cit. 2012-06-26]. Dostupné z: http://cs.wikipedia.org/wiki/Metoda_kone%C4%8Dn%C3%BDch_prvk%C5%AF
- [28] Numerické simulace: Metoda konečných prvků. *Ústav nauky o dřevě* [online]. 2012 [cit. 2012-06-27]. Dostupné z: <http://wood.mendelu.cz/cz/sections/FEM/?q=node/32>
- [29] TEORIE DIFERENCIÁLNÍCH ROVNIC. In: *EAMOS* [online]. 2012 [cit. 2012-06-29]. Dostupné z: http://eamos.pf.jcu.cz/amos/kat_fyz/externi/kat_fyz_0062/kapitola1.pdf
- [30] Metoda konečných prvků v elektrotechnické praxi. In: Ing. KAČOR Ph.D., Petr. <Http://homen.vsb.cz/~kac37/> [online]. 2012 [cit. 2012-06-28]. Dostupné z: http://homen.vsb.cz/~kac37/_num_met_Kacor/
- [31] Teorie diferenciálních rovnic. In: *EAMOS* [online]. 2012 [cit. 2012-08-03]. Dostupné z: http://eamos.pf.jcu.cz/amos/kat_fyz/externi/kat_fyz_0062/kapitola2-3.pdf
- [32] JÍCHA, Miroslav. *Počítačové modelování úloh vedení tepla a proudění*. 1. vyd. Brno: VUT Brno, 1991, 121 s. ISBN 80-214-0364-0.
- [33] KOZUBKOVÁ, Milada. *Modelování proudění tekutin: FLUENT, CFX* [Scripta]. Ostrava, 2008 [cit. 4.7.2012]. Dostupné z: <http://www.338.vsb.cz/PDF/Kozubkova-Fluent.pdf>
- [34] DRÁBKOVÁ, S. a kol. *Mechanika tekutin. Ostrava: VŠB-TU Ostrava, 2007. 248 s.* (Elearningová učebnice). ISBN 978-80-248-1508-4.
- [35]ACHEMA 2009: Počítačová dynamika tekutin. *CHEMagazín*. 2009, XIX, č. 2, s. 2. Dostupné z: http://www.chemagazin.cz/userdata/chemagazin_2010/file/chxix_2_cl6.pdf
- [36] TechSoft Engineering, spol. s r.o. *Www.techsoft-eng.cz* [online]. 2006-2007 [cit. 2012-02-20]. Dostupné z: <http://www.techsoft-eng.cz/>
- [37] Humusoft.cz: Technické výpočty, řídicí technika, simulace ,... <Http://www.humusoft.cz/> [online]. 1991 - 2012 [cit. 2012-06-26]. Dostupné z: <http://www.humusoft.cz/produkty/comsol/>
- [38] Designworldonline.com. <Http://www.designworldonline.com/> [online]. 2010 [cit. 2012-06-26]. Dostupné z: <http://www.designworldonline.com/articles/6141/314/COMSOL-introduces-the-Plasma-Module.aspx> .
- [39] Comsol.eu: Introduction to COMSOL Multiphysics 4: Building a model. <Http://www.comsol.eu/> [online]. 1998 - 2012 [cit. 2012-06-26]. Dostupné z: <http://www.comsol.eu/products/tutorials/introduction/page2/> .

- [40] Comsol.eu: Introduction to COMSOL Multiphysics 4: Building a model.
Http://www.comsol.eu/ [online]. 1998 - 2012 [cit. 2012-06-26]. Dostupné z:
<http://www.comsol.eu/products/tutorials/Plasma%20Model%20for%20a%20GEC%20ICP%20Reactor%20-%20Tutorial/>
- [41] TechSoft Engineering, spol. s r.o. *Www.techsoft-eng.cz* [online]. 2006-2007 [cit. 2012-02-20].
Dostupné z: <http://www.techsoft-eng.cz/>
- [42] TÁRNOK, Gábor. *Aerodynamická interakce dvou vozidel*. Brno, 2011. Diplomová práce.
VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ, Fakulta strojního inženýrství. Vedoucí práce Ing.
JAN VANČURA.
- [43] FINK, Milan a Ladislav ŘEZNÍČEK. *FEM/MKP - Základy použití metody konečných prvků pro technické výpočty v programu Autodesk Inventor Professional*. Trutnov, 2006. Dostupné z:
<http://www.spstrutnov.cz/o-skole/projekty/technicke-vypocty/skripta-fem-mkp.pdf> . Scripta.
SPŠ, Školní 101, Trutnov.
- [44] TCAD. *Server technických CAD řešení* [online]. 1993 - 2012 [cit. 2012-07-19]. Dostupné z:
<http://free.tcad.cz/>
- [45] SCHULZ, Martin. *Návrh a simulace RLC obvodů*. Brno, 2007. Bakalářská práce. Masarykova univerzita v Brně.