

Jihočeská univerzita v Českých Budějovicích

Pedagogická fakulta – Katedra fyziky

Příprava internetových stránek s prezentací základních pojmů počítačové
fyziky

Bakalářská práce

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

Autor: Jaroslava Křížová

Anotace

Tato bakalářská práce si klade za cíl seznámit studenty středních škol se základními poznatky oboru počítačová fyzika. Práce je koncipována jako přehledový text, který si neklade za cíl poskytnout odborný základ pro vědeckou práci, ale je zaměřena především na pochopení základních pojmů. V úvodu se zabývám definicí počítačové fyziky. Kapitola Hardwarové a softwarové prostředky počítačové fyziky začíná stručným popisem architektury počítače a jeho historickým vývojem. Následují základy programování s krátkým přehledem programovacích jazyků. V kapitole Počítačové modelování podávám stručnou charakteristiku tohoto oboru, seznamuji čtenáře se základními pojmy a technikami modelování. V následující kapitole je čtenář seznámen s principy spojitého i hybridního modelování a také metodou Monte Carlo. Čtvrtá kapitola by měla být zajímavou ukázkou moderních trendů v počítačovém modelování. V posledních kapitolách se zabývám stručnou charakteristikou programů, které jsou v oblasti počítačového modelování nejčastěji používány (MATLAB, Fluent a COMSOL Multiphysics).

Abstract

This Bachelor's Thesis has undertaken to acquaint high school students with basic information in the field of computer physics. The work is conceived as a textual overview which does not aspire to provide a specialization base for scientific work, but is primarily aimed at understanding the basic terms. In the introduction, I deal with the definitions of computer physics. The chapter entitled Hardware and Software Devices of Computer Physics begins with a brief description of the architecture of the computer and its historical development. This is followed by the basics of programming with a short overview of programming languages. In the chapter entitled Computer Modelling, I present a brief characteristic of this field and acquaint the reader with the basic terms and techniques of modelling. In the following chapter, the reader is acquainted with the principles associated with hybrid modelling as well as the Monte Carlo method. The fourth chapter is intended to be an interesting demonstration of modern trends in computer modelling. In the last chapters, I deal with the brief characteristic of programs that are most commonly used in the field of computer modelling (MATLAB, Fluent and COMSOL Multiphysics).

Prohlášení

Prohlašuji, že svoji bakalářskou 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 bakalářské práce, a to v nezkrácené podobě elektronickou cestou ve veřejně přístupné části databáze STAG provozované Jihočeskou univerzitou v Českých Budějovicích na jejích internetových stránkách.

V Českých Budějovicích dne 23. 4. 2011

Jaroslava Křížová

Touto formou děkuji svému konzultantovi RNDr. Petru Bartošovi, Ph.D., za cenné rady a připomínky při zpracování mé práce.

Úvod	7
1 Hardwarové a softwarové základy počítačové fyziky	9
1.1 Hardwarové prostředky	9
1.2 Technický rozvoj počítačů.....	10
1.3 Softwarové prostředky	12
1.4 Zásady programování	14
2 Počítačové modelování	16
2.1 Metodologie modelování.....	16
2.2 Základní pojmy	16
2.3 Řešení fyzikálního problému metodami počítačového modelování.....	19
2.4 Techniky počítačového modelování	19
3 Metody Modelování.....	21
3.1 Metoda molekulární dynamiky	21
3.1.1 Metoda konečných diferencí.....	21
3.2 Verletova metoda	23
3.3 Metoda Monte Carlo	25
3.4 Metody spojitého modelování	28
3.4.1 Stručný popis spojitého modelování	28
3.4.2 Příklad spojitého modelu	28
3.5 Metody hybridního modelování.....	29
3.5.1 Energetický model	30
3.5.2 Prostorový model	30
3.5.3 Iterační model	30
4 Moderní trendy v počítačovém modelování	31
4.1 Neuronové sítě	31
4.2 Fuzzy logika.....	32
4.3 Distribuovaný výpočet	33
5 Programovací jazyk MATLAB	34
5.1 Výpočetní jádro.....	36
5.2 Grafický subsystém	36
5.3 Otevřená architektura	36
5.4 Pracovní nástroje.....	37

5.5	Toolboxy.....	38
5.6	Aplikační knihovna, simulink.....	38
6	Program Fluent.....	41
6.1.1	Síťování a numerika, paralelní zpracování.....	44
6.1.2	Dynamické a pohybující se sítě	45
6.1.3	Turbulence a akustika.....	46
6.1.4	Přenos tepla.....	46
6.1.5	Reakce proudu	47
6.1.6	Vícefázové proudění	47
6.1.7	Zpracování výsledků a export dat	48
6.1.8	Uživatelské nástroje	48
7	Program COMSOL Multiphysics.....	49
7.1	Příklad použití programu COMSOL Multiphysics®	50
8	Závěr	51
9	Použitá literatura.....	52

Úvod

Počítačová fyzika je poměrně nový obor, zabývající se vědeckým bádáním, jehož název vznikl spojením dvou slov – počítač a fyzika. Počítačová fyzika řeší metodiku fyzikálního problému na rozdíl od jiných směrů vědecké práce, kde je hlavní úkol objekt studia (např. fyzika plazmatu). Metodika fyzikálního problému se dělí na dva základní směry na experimentální a teoretický. [1]

Experimentální fyzika poskytuje data, nabízí podněty pro tvoření teorií a umožňuje tyto teorie testovat. Teoretická fyzika zobecňuje experimentální poznatky a navrhuje další experimenty k prohloubení znalostí, nebo dává podněty pro studium oblastí nových. Tyto dva směry se vzájemně dobře doplňují, ale postupem času a příchodu počítačů začal do těchto dvou oborů zasahovat i obor počítačová fyzika. Ta rozšířila vědecké bádání na tři navzájem se doplňující obory, tedy experimentální fyziku, teoretickou fyziku a počítačovou fyziku. Experimentální fyzika generuje data, využívá jak teoretickou tak počítačovou fyziku s tím, že počítačová fyzika provádí analýzu těchto dat, zabezpečuje řízení experimentálních zařízení, vytváří modely reálných procesů a na jejich základě analyzuje experimentální data a navrhuje nové experimenty. Vazba mezi počítačovou fyzikou a teoretickou fyzikou je taková, že teoretická fyzika dává rovnice pro řešení metodami počítačové fyziky a naopak interpretuje výsledky počítačovými metodami získané. Počítačová fyzika pak poskytuje teoretické fyzice pomoc při provádění rozsáhlých výpočtů a podobně jako experimentální fyzika dává prostřednictvím počítačových experimentů podněty pro vytváření nových teorií a pomáhá při jejich testování. [1]

Vysvětlení, co to vlastně počítačová fyzika je, není jednoduché. Ve shrnutí to můžeme definovat jako takové postupy řešení fyzikálních úloh, při kterých hraje počítač významnou roli. V současné době se bez počítače neobejde žádná laboratoř, ať jde jen o napsání textu, přes výpočty, uchování dat, atd. Nezapomínáme na počítačové simulace nebo symbolické manipulace, částečně i automatizace experimentu, apod. S dalším vývojem hardware se samozřejmě vytvářejí i nové oblasti počítačové fyziky, např. práce v počítačových sítích nebo studium paralelních procesů, vytváří také nové metody práce. Hranice tohoto oboru se s vývojem počítače neustále mění. [1]

Počítačová fyzika se nadále také vnitřně dělí na:

- Klasickou počítačovou fyziku
- Moderní počítačovou fyziku

Nejpoužívanější oblastí je počítačové modelování, protože nám pomáhá zároveň s experimentálním, ale i teoretickým studiu fyziky. Je také rozšířen do několika metodik, např. metoda Monte Carlo. [1]

Klasickou počítačovou fyziku dělíme do čtyř směrů:

- Počítačové modelování
- Počítačová grafika a vizualizace
- Zpracování obrazu
- Integrální transformace

Za moderní počítačovou fyziku můžeme pokládat aplikace neutronových sítí, fuzzy logiku, evoluční programování, atd. Zde se ale budeme spíše zabývat klasickou počítačovou fyzikou. [1]

Vývoj výpočetní techniky ve dvacátém století můžete vidět v tabulce 1. Další vývoj je zaměřen na integrované obvody, množství prvků v jednom obvodu dosahuje podle výkonu, až sto milionů.

generace	rok	konstr.prvky	výkon[op/s]	RAM[b]	SW
0.	1941	relé	10^1	10^3	stroj.kód
1.	1946	elektronky	10^2	10^4	assembler
2.	1960	polovodiče	10^3	10^5	FORTTRAN
2,5.	1964	mikromoduly	10^4	10^6	op. systémy
3.	1966	int. obvody	10^5	10^7	sdílení času
3,5.	1971	int. obvody	10^6	10^8	virt. paměti
mikroprocesory					
generace	rok	typové označení	výkon[MIPS]	šířka datové sběrnice [b]	hodinová frekvence [MHz]
3,5.	1971	4004	0,06	4	0,108
3,5.	1974	8080	0,64	8	2
3,5.	1978	8086	0,75	16	5 - 10
4.	1982	80286	10 - 100	16	6 - 20
4.	1985	80386	10 - 100	32	16 - 33
4.	1989	80486	10 - 100	32	25 - 100
5.	1993	Pentium	$10^3 - 10^9$	64	60 - 166
5.	1995	Pentium Pro	$10^3 - 10^9$	64	150 - 200
5.	1997	Pentium II	$10^3 - 10^9$	64	200 - 450
5.	1999	Pentium III	$10^3 - 10^9$	64	400 - 1400
5.	2000	Pentium 4	$10^3 - 10^9$	64	1400 - 3400
5.	2001	AMD Opteron	$10^3 - 10^9$	64	1600 - 3200
5.	2003	AMD Athlon 64	$10^3 - 10^9$	64	2600 - 3000
5.	2007	AMD Phenom	$10^3 - 10^9$	64	1800 - 2600
5.	2008	Phenom II	$10^3 - 10^9$	64	2500 - 3400
5.	2011	Sandy Bridge	$10^3 - 10^9$	64	2700 - 3400

Tabulka 1. Hlavní charakteristiky výpočetní techniky [3]

1.2 Technický rozvoj počítačů

V roce 1950 byly vymyšleny nové vědní disciplíny, obory a zcela jiný přístup k modelování. Druhá polovina 20. století se vyznačuje velkým rozmachem techniky, kdy nebylo možné řešit klasickým modelováním úkoly technické praxe na úrovni doby.

Období VTR (VTR je etapa velkého rozvoje techniky, počítačů, nových vědních disciplín, které začalo po roce 1950) přineslo pro řešení problémů modelováním kvalitativně nové technické prostředky pro informační technologie (získávání, uchovávání, zpracovávání a sdělování informací). Jsou to změny podmíněné elektronizací a komputizací, jako základními prvky VTR.[1]

Další vývoj lidského myšlení spojeného s modelováními je např. automatizace, počítačová podpora ale i psychologie.

Tyto vědní obory jsou důležité pro vymezení modelů a modelování na jiné úrovni vzhledem ke klasickému modelování technické praxe.

Samozřejmě ale zcela nový pohled na modelování přinesl počítač. Ten je označován za nejlepší přístroj, který člověk dostal do rukou. Pro matematika a mechanika

N. Mojsejeva byl počítač jako vynalezení parního stroje nebo dokonce objevení ohně.

Počátek počítačových strojů začíná v roce 1934, kdy ing. Konrad Zuse sestavuje mechanickou výpočetní pomůcku. S rokem 1937 H.H. Akien vytváří automatický počítač pro výpočty složitých rovnic. Další rok přinesl Zuse první počítač ještě s elektromechanickými prvky, 1939 H.H. Akien podepisuje s IBM kontakt na spolupráci vytvoření počítače. V roce 1941 je vyroben první reléový počítač, který vynalezl K. Zuse. Při roku 1943 přichází další reléový počítač, tentokrát je autorem Akien. Název počítače je MARK I. Následující rok 1944 se objevuje počítač MARK I, který byl využíván na Harvardské univerzitě. Přispíval k řešení Besselových funkcí, simulace atomové bomby, problematik obrany atd.

Rok 1943 až 1946 J. P. Eckert vyvíjí automatický počítač ENIAC, který byl 1000krát rychlejší než reléové počítače. Byl používán pro výpočty drah balistických střel. Symbolický rok 1945 J. von Neumann (1903 – 1957) vytvořil koncepci samočinného počítače EDVAC s pěti hlavními jednotkami: vstupem, řadičem, operační jednotkou, výstupem a s vnitřním řízením podle programu, uloženým v paměti. Vnitřní řízení bylo geniální myšlenkou, kterou dnes považujeme za samozřejmost.[1]

Vymyšlení první feritové paměti spojujeme s datem 1950. První sériově vyráběný počítač UNIVAC I přiřazujeme roku 1951. Končí průkopnická éra počítačů. Pro toto období je charakteristické, že počítače byly vytvářeny v jediném provedení a jejich vývoj trval dlouhou dobu. [1]

1.3 Softwarové prostředky

Mezi softwarové prostředky patří především programovací jazyk a operační systém, který slouží k ovládní procesu výpočtu.

Počítač je řízen strojovým kódem, ten je tvořen posloupností příkazů, které říkají řadiči počítače jaký výpočet má udělat, jaké argumenty použít, jak naložit s výsledkem operace a jak dále pokračovat. Strojový kód se píše zpravidla ve dvojkové soustavě. Zápis se liší podle počítače nebo použitého mikroprocesoru.

Protože psaní programu ve zdrojovém kódu je velmi složité, začalo se přemýšlet, jak tuto práci programátorům ulehčit. Jak nechat mechanický provoz počítači a uživateli pouze autorskou činnost. Tak se přešlo k vývoji programovacích jazyků.

První přišel jazyk symbolických adres – assembler. Nelze ho nazývat pravým programovacím jazykem, šlo o zrychlený zápis programu ve strojovém kódu. Měl však základní výhodu a tou byla rychlost výpočtu a plná kontrola nad činností počítače. Assembler je ale stále dost složitý na programování a proto se používá, jen když rychlost výpočtu je hlavním kritériem (např. řízení experimentu). Jinak se používají vyšší programovací jazyky.

Programovací jazyk – jedna instrukce znamená více instrukcí ve zdrojovém kódu.

Historie programovacích jazyků:

Kolem roku 1957 se objevily tři programovací jazyky, které sehrály v dalším vývoji významnou roli FORTRAN, COBOL a ALGOL. FORTRAN byl vyhovující pro vědecko-technické úlohy, COBOL pro ekonomické výpočty a ALGOL pro práci v oblasti matematiky.

Po těchto třech programovacích jazycích následně přišel prudký rozmach ve vývoji programovacích jazyků a do dnešní doby ještě neskončil. O tomto prudkém vývoji vypovídá i jejich počet – od roku 1980 jich bylo více než 2000.

Programovací jazyky dělíme:

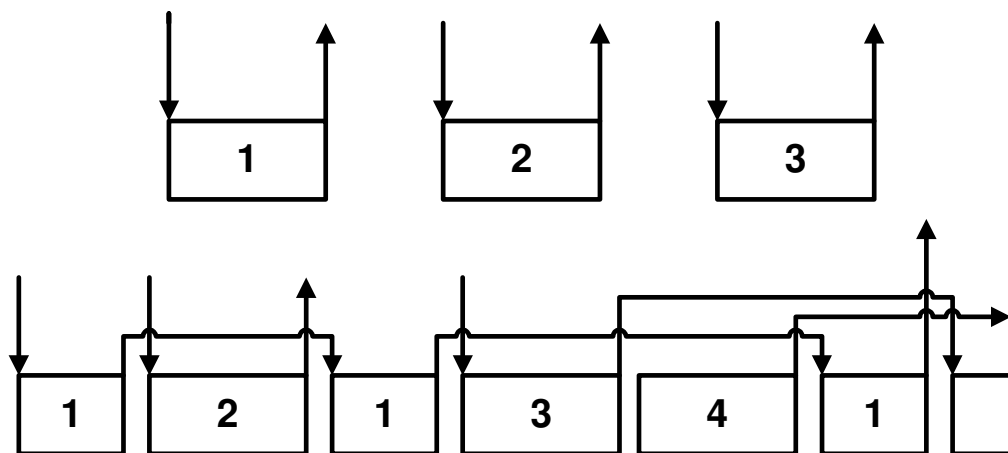
- Jazyky strojově orientované a problémově orientované (assembler)
- Jazyky kompilační a interpretační (C, Pascal)
- Jazyky zaměřené na vědecko-technické výpočty, ekonomii, ...(FORTRAN, ALGOL a COBOL) atd.

Z hlediska počítačové fyziky jsou nejrozšířenější programovací jazyky FORTRAN-90/95 a C. Jazyk FORTRAN byl vyvinut v roce 1957. Nejdříve FORTRAN-66, dále FORTRAN-77 (dnes používán ojedinele na některých pracovištích), nakonec FORTRAN-90 a poslední verze FORTRAN-95. Jazyk C vyvinut v roce 1978, jako standartní verzi známe ANSI C z roku 1988 s rozšířenou verzí C++.[1]

Dále se často využívají speciálně vyvinuté programy (MATLAB, COMSOL, FLUENT). Také bych ráda zmínila programovací jazyk Abaqus, který programuje systém SAP, jehož hojně využívají velké firmy pro komplexní řízení (účetnictví, výroba, nákup, prodej, sklady..)

Dále je potřeba zvolit i vhodný operační systém. Dříve byly fyzikální úlohy zapisovány ve strojovém kódu a ovládání počítače ručně přes operátorskou konzoli. S nástupem programovacích jazyků byly vyvinuty i operační systémy, které bylo možno předem řízeně naprogramovat. Mezi první řadíme MS-DOS, Windows a UNIX. Pro mikropočítače volíme Windows, LINUX. Jestliže máme k dispozici stanici s větší výpočetní technikou, většinou spoléháme na operační systém UNIX pro jeho jednoduchost, možnosti práce s více úlohami také více uživatelů může pracovat najednou. [1]

Významnou roli hraje i volba režimu práce. Na obrázku 2 jsou zobrazeny režimy práce.



Obr. 2. Režimy práce: individuální (nahore), multiprogramový (dole) [4]

Individuální režim – Uživatel udělá práci, vypne počítač. Přijde další uživatel, který pracuje stejným principem jako ten první. Je to nejstarší režim.

Dávkový režim – Pro zefektivnění práce zde využíváme operačního systému, který spustí následující výpočet pro dalšího uživatele hned po tom, co první uživatel svůj výpočet dokončí. Jsou tady eliminovány časové mezery individuálního režimu.

Multiprogramování – Při dávkovém režimu vždy nebylo využití počítače maximální. V režimu multiprogramování jsou v paměti vloženy programy více uživatelů.

Programy mají odlišné priority. Na obrázku 2. s nejvyšší prioritou č.1. dále č.2.atd. Tento režim plně využívá činnost procesoru. Při volné chvíli procesoru ho okamžitě využije k jiné operaci.

V laboratořích na superpočítačích se dnes setkáváme s režimem multiprogramování, ale v 50. letech to byl režim individuální. V 60. letech se přešlo na dávkový a dále v 70. až 80. letech to byl již mutliprogramovací režim. Co se týče osobních počítačů, zde se používá většinou individuální režim, ale ty výkonnější jsou také multiprogramovatelné. Jako příklad multiprogramování bych uvedla MetaCentrum (sdružení mezinárodního výzkumného prostředí, do kterého patří vysoké školy). Uživatelé MetaCentra mohou programovat dohromady jeden projekt.

1.4 Zásady programování

Dříve nebyl brán takový zřetel na úpravu programů. Proto byly často nepřehledné, špatně čitelné, neefektivní. Toto programování se nazývá divoké programování.

Strukturované programování vzniklo v druhé polovině 70. let. Program, který strukturovaný kód vyžaduje je např. Pascal. V průběhu několika let se ukázalo strukturované programování v počítačové fyzice velice efektivní. Statistickými výpočty vychází vzorec pro výpočet doby tvorby programu. N je počet instrukcí (délka), m je exponent, který se liší pro strukturované tak pro divoké programování.

$$T \approx K \cdot N^m \quad (1)$$

Programování top-down a bottom-up

Nejdříve si uděláme přípravu programu bez počítače. Zjednodušíme řešení tím, že ho rozdělíme do bloků. Po té přistoupíme k programování na počítači a skládáme v něm předem připravené bloky.

Přednostní používání určitých programových konstrukcí

Zde se máme vyvarovat vytváření nepřehledných programů. V zásadě používat příkazy, které dělají v programu přirozené bloky. Inspiraci najdeme v programovacím jazyku Pascal, který lze považovat za první jazyk využívající tzv. strukturované programování.

Modularita programu

Programovací jazyk by měl obsahovat konstrukce pro uzavřenou strukturu programu, jejíž obsah by se po určitém odladění již neměnil. Struktura tohoto programu se dá zavolat jen přesně určeným způsobem. Jde o tyto konstrukce: podprogram, blok a modul. [1]

2 Počítačové modelování

Modelování představuje jeden z nejobecnějších způsobů zobrazení vnějšího světa, prováděného ke zkoumání v něm existujících objektivních zákonitostí. Modelování je experimentální informační proces, při němž se zkoumanému systému (originálu, objektu, dílu) jednoznačně přiřazuje podle určitých kritérií jiný systém, fyzický nebo abstraktní, nazývaný model. Modelování dynamických systémů s přímým nebo nepřímým zpětným působením na zkoumaný objekt se nazývá simulací. [5]

2.1 Metodologie modelování

Metodologie modelování má za úkol najít společné hledisko modelů a to budoucích i existujících, které zprostředkovalo obecnou klasifikaci a systemizaci modelů a určilo jejich postavení v systému experimentálních prostředků. Vytváření jednotné metodologie ztěžuje izolovaný rozvoj jednotlivých druhů modelování a nesjednocenost informačních vlastností modelů.

Úroveň modelování zvyšuje rostoucí rozvoj kybernetiky. Velkým přínosem pro metodologii modelování přispělo rozšíření teorie systémů i na modelové systémy.

2.2 Základní pojmy

V této kapitole si vysvětlíme několik důležitých pojmů pro počítačovou fyziku. Zavedením pojmu *simulace* se vytřídily množiny modelů a to modely s významem pro rozvoj vědeckého bádání. To jsou modely dynamických systémů a procesů, modely mezních ustálených stavů, v nichž nacházíme ustálený systém. Nezáleží na tom, jestli s modelem pracujeme na počítači nebo s využitím jiných technických prostředků. Pojem simulace můžeme vždy nahradit obecněji za výraz modelování, obráceně však nikoliv. Pro pojem simulace je hlavním faktorem přenos informací získaných na simulačním modelu zpět na zkoumaný objekt.

Pojem *výpočet* nevyjadřuje vztah k řešenému procesu. Je to mechanický sled výpočetních operací.

Dále *podobnost* lze chápat jako jednoznačné vzájemné přiřazení mezi různými systémy v jejich struktuře, vlastnostech a chování. Fyzikální podobnost vyjadřuje

podobnost mezi systémy a procesy stejné fyzikální podstaty a zahrnuje vedle geometrické podobnosti podobnost parametrů a stavových veličin systému. Matematická podobnost vyjadřuje podobnost mezi systémy a procesy majícími stejný matematický podpis. Jde-li o fyzikálně odlišné systémy a procesy, nazývá se analogie. Kybernetická (funkcionální) podobnost vyjadřuje matematickou podobnost ve vnějším chování systémů. Podle uvedených tří druhů podobností lze modelování dělit na fyzikální, matematické a kybernetické. [1]

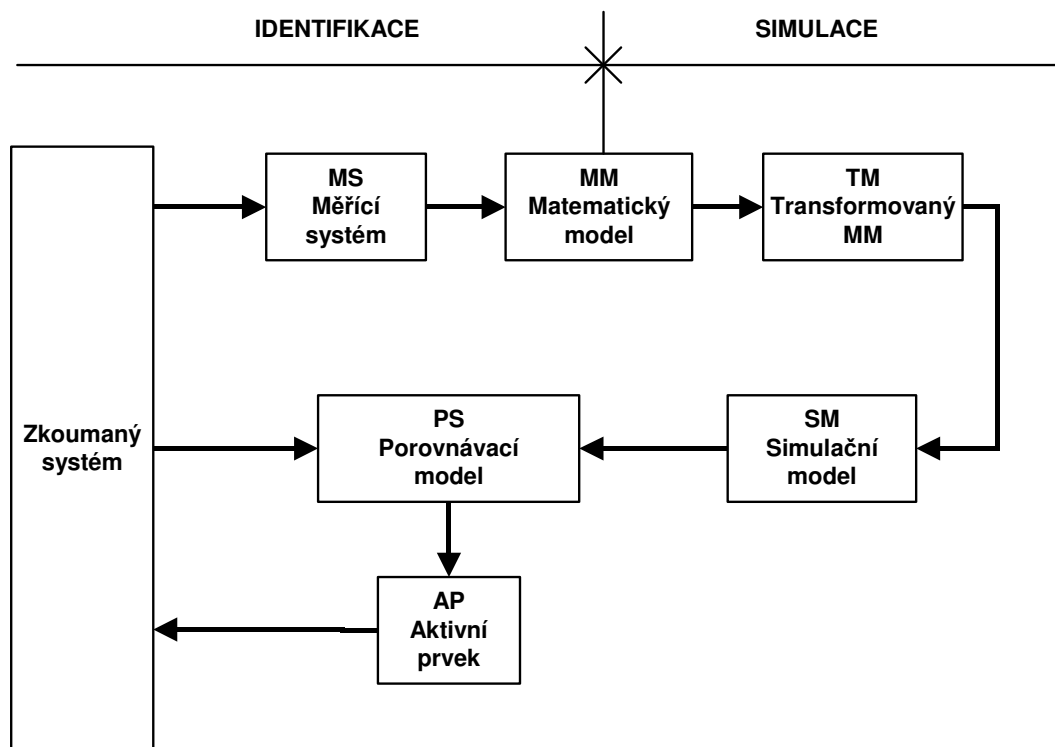
Pak *kybernetické modelování* využívá modely černé skříňky. Tento pojem představuje systém s vnitřní strukturou, o které nemáme informace. Víme pouze o jeho vnějším chování. Pojem černá skříňka zavedl N. Wiener. Skříňku můžeme mít i bílou. Je opakem černé skříňky a ta nám poskytuje informace o vnitřní struktuře systému a probíhajícím procesu. Samozřejmě můžeme zavést i skříňku s částečnými informacemi o vnitřním systému, která se nazývá *šedá skříňka*.

Důležitý pojem je také *experiment*, tím myslíme obecný pojem pro modelování, činnost, která směřuje k rozvoji vědy. Experiment má více druhů. Může být fyzický, myšlenkový, vědecký, výrobní, výukový atd. Pro rozvoj společnosti se nejvíce osvědčil experiment vědecký, který se opírá o modelování. Důležitá vlastnost experimentu je to, že se může opakovat za stejných podmínek.

Další podstatný pojem je *měření*, tímto pojmem máme na mysli informační proces, jehož základem je snímání a převádí danou informaci z objektu k uživateli. Následný pojem *experimentální technika* se týká měřící, výpočetní a řídicí techniky. S tím spojená *technická kybernetika* se zabývá vyšetřováním a řízením složitých systémů a *plán experimentu*, do kterého patří algoritmus experimentu, postup práce, vlastnosti práce atd. Pak důležitý *algoritmus experimentu* což je to souhrn jasně daných úkolů a jejich splnění vede k uskutečnění cíle experimentu. A za hlavní pojem máme *počítač*, který má tři základní úlohy v experimentu. Celkově pracuje s experimentálními daty, ovládá průběh experimentování a pracuje jako modelový prostředek.

Pro modelování má spolehlivé zjištění vlastností zkoumaného systému základní význam. S tím úzce souvisí věrohodnost matematického a simulačního modelu, a tedy i výsledků modelování. Proto identifikace představuje pro modelování a simulaci systému důležitou etapu, na níž do značné míry závisí účinnost modelování. Identifikace je experimentální způsob určování podstatných charakteristik systému a procesu umožňující sestavení matematického modelu. Charakteristikami systému se rozumí zejména různé fyzikální vlastnosti, struktura a parametry systému a v něm

probíhajícího procesu. Diagnostikou se rozumí zjišťování stavu systému, zejména odchylek od předpokládaného stavu. [1]



Obr. 3. Identifikace a simulace v procesu poznávání zkoumaného systému [6]

Na blokovém schématu (obr. 3) je postup identifikace zkoumaného dynamického systému a navazujícího procesu simulace. Podstatné informace o chování systému (zkoumaném objektu, originálu, díle) jsou získány přímým měřením pomocí měřícího systému. Vyústěním procesu identifikace je formulace matematického modelu.

Matematický model se transformuje do tvaru umožňujícího vytvořit zvolenými technickými prostředky model simulační. Z porovnání chování zkoumaného systému a simulačního modelu lze soudit na věrohodnost modelu získanou informaci využít ke zdokonalení originálu. [1]

Pro vytvoření modelu matematického máme na výběr ze dvou možností (obr. 2). První možností je přímá. U této možnosti se zaměřujeme na souhrn poznatků o chování zkoumaného objektu podaných jednak identifikací, jednak jako výsledek rozvoje informací v daném oboru. Mnohokrát se ale setkáváme s tím, že zásadní informace jsou nám poskytnuty nepřímou fyzikálním modelem.

2.3 Řešení fyzikálního problému metodami počítačového modelování

Modelování můžeme rozdělit do několika kroků:

1. Formulace problému

2. Vytvoření modelu

Zjednodušujeme studovaný jev, protože neznáme všechny vlastnosti jevu. Zahrnujeme do modelu základní rysy fyzikálního jevu. Méně podstatné věci zanedbáváme. Výsledkem správnosti si ale nejsme plně jisti, proto musíme modelování srovnat s experimentálními daty. Pokud jsme model příliš zjednodušili nebo zadali chybně, odrazí se tyto údaje i na výsledném modelování. Například si musíme dávat pozor na desetinná místa ve výsledcích.

3. Řešení modelu

Model řešíme matematickými postupy. Rovnicemi diferenciálními, parciálními, obyčejnými nebo i pravděpodobností pomocí matematické statistiky. Pomáháme si standardními matematickými postupy.

4. Srovnání výsledků modelování s experimentálními údaji

Zde porovnáváme výsledky přímého měření s výsledky modelování. [1]

2.4 Techniky počítačového modelování

- Částicové techniky

Zde se zabýváme jednotlivými částicemi, z kterých se studovaný jev skládá. Detailně zde studujeme atomy, ionty nebo elektrony. Když testujeme chování modelu, musíme se řídit srovnáváním výsledků modelování s experimentálními údaji, které jsme měli v předchozím odstavci. Tím sice část informací ztratíme, ale tato informace nám zůstane k dispozici a můžeme s ní dále pracovat, vracet se k ní a dále ji srovnávat s ostatními výsledky.

Podle popisu studovaného celku se zaměřujeme na tyto základní metody:

- Metoda molekulární dynamiky (numerické řešení pohybových rovnic)
- Metoda Monte Carlo (chování jednotlivých částic pomocí matematických zákonů pravděpodobnosti)

- Spojité modelování

U této techniky využíváme makroskopické úrovně. Plazma řešíme jako kontinuum s určitou teplotou, tlakem, složením, rychlostí proudění, atd. Zde se

řídíme zákony zachování energie, náboje atd. Z matematiky tady používáme diferenciální rovnice i parciální rovnice. [1]

- Hybridní modelování

Používáme kombinaci částicového a spojitého modelování s cílem eliminovat nedostatky těchto metod a podpořit jejich přednosti.

3 Metody Modelování

3.1 Metoda molekulární dynamiky

Molekulární dynamika zachycuje vývoj systému částic podle zákonů mechaniky, čímž zde rozumíme mechaniku klasickou. Budeme předpokládat, že integrační potenciál U je spojitý a diferencovaný a může obecně obsahovat jak mezičásticový potenciál, tak i vnější pole, např. síly, kterými působí na částice stěny nádoby. [7]

3.1.1 Metoda konečných diferencí

Je numerická metoda sloužící k simulaci průběhů napětí, deformací, vlastních frekvencí, proudění tepla, jevů elektromagnetismu, proudění tekutin atd., na vytvořeném fyzikálním modelu. MKP je užívána především pro kontrolu již navržených zařízení, nebo pro stanovení kritického (nejnamáhanějšího) místa konstrukce. Ačkoliv jsou principy této metody známy již delší dobu, k jejímu masovému využití došlo teprve s nástupem moderní výpočetní techniky.[8]

Při řešení pohybových rovnic dostaneme časový vývoj systému. V Newtonově znění se stýkají pohybové rovnice zrychlení částic přímo se silami.

$$\vec{r}_i = \frac{\vec{F}_i}{m_i}, \quad i = 1, \dots, N, \quad (2)$$

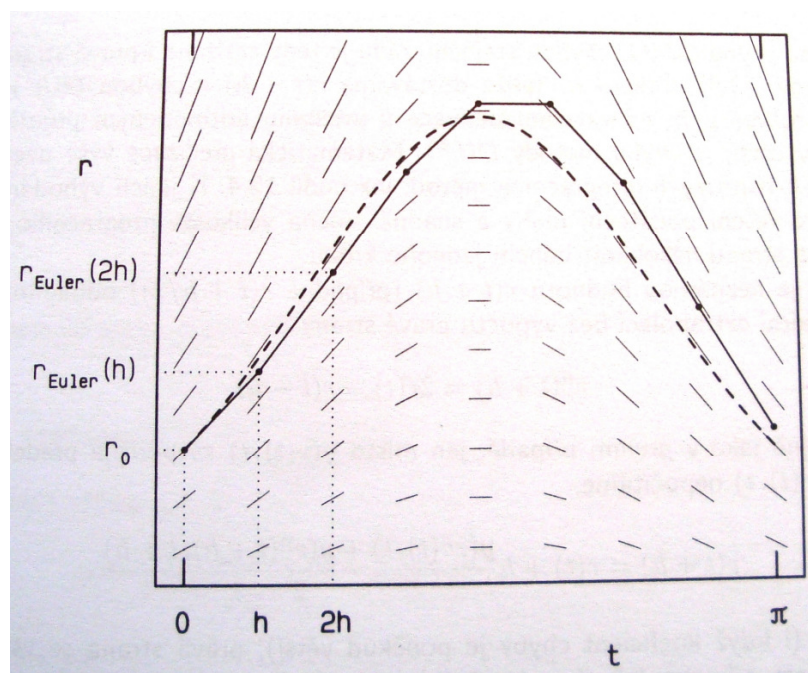
Označení šipky, kterou vidíme nad písmenem r , vyznačují druhou časovou derivaci. Tato rovnice (2) je soustavou $3N$ základních diferenciálních rovnic druhého řádu. K této rovnici samozřejmě ještě nesmíme zapomenout přidat počáteční podmínky. Tím myslíme všechny polohy r_i , rychlosti \dot{r}_i pro počáteční stav $t = t_0$.

Vyřešením této soustavy získáme trajektorii. To je funkce $r_i(t)$, $i = 1, \dots, N$, v nějakém časovém intervalu $[t_0, t_{\max}]$. Na počítači budeme tuto spojitou funkci psát v diskrétních bodech tvaru $t = t_0 + ih$. Zde i je nezáporné celé číslo a h je krátký časový krok, tak zvaný integrační krok. Kde h musí být tak dlouhý úsek, aby se v jeho průběhu měnily polohy síly a částic jen „trochu“. Postup, jenž je zde popisován nazýváme *metoda konečných diferencí*. [7]

Při řešení soustavy rovnic (2), kterou jsme řešili metodou konečných diferencí, máme k dispozici mnoho různých metod. Proč tedy mluvíme o této metodě jako o disciplíně? Zaprvé zjišťujeme, že počet rovnic soustavy je hodně velký (v dnešní době počítáme až miliony), to ale není ten nejhlavnější důvod. Co nás zajímá mnohem víc je tvar rovnic a v první řadě tvar pravé strany rovnic. Síla stojící na pravé straně je funkcí poloh všech částic a musíme ji počítat v každém kroku a to je velmi časově náročné, například výpočet pro párový potenciál zahrnuje výpočet pro $N(N-1)/2$ členů.

Kvůli této náročnosti jsou ostatní metody nežádoucí. Protože některé metody například počítají několikrát působící sílu na částice.

Ale při molekulárních simulacích z pravidla nepotřebujeme integrační krok měnit a to nám dává na výběr z mnoha dalších metod. V konkrétním případě typický molekulárně dynamický experiment s využitím nejvýkonnějších počítačů znázorňuje vývoj reálného systému po dobu řádově jen pikosekundy až nanosekundy (to představuje statisíce až milióny integračních kroků). Konečný výsledek je spočítán za několik dnů, nikoli za několik měsíců.



Obr.4: Eulerova metoda integrace diferenciální rovnice prvního řádu. Směry tenkých čárek vyznačují v každém bodě (t, r) hodnotu derivace r , čárkovaná křivka udává přesné řešení. [7]

3.2 Verletova metoda

Tato metoda se používá ve fyzice plazmatu, kde máme velký počet částic, ale hlavní je chování celého souboru částic. Zde využíváme méně přesné, ale rychlejší algoritmy. Menší přesnost nám vyrovnává následné průměrování trajektorií, kde získáváme makroskopické veličiny jako je například elektrický proud.

Algoritmus Verletovy metody pro řešení pohybových rovnic: [1]

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

$$\vec{r}_i^{k+1} = \vec{r}_i^k + \vec{v}_i^k \Delta t + \frac{1}{2m_i} \vec{F}_i^k \Delta t^2 \quad (3)$$

$$\vec{F}_i^{k+1} = \dots \quad i = 1, \dots, N \quad (4)$$

$$\vec{v}_i^{k+1} = \vec{v}_i^k + \frac{1}{2m_i} (\vec{F}_i^k + \vec{F}_i^{k+1}) \Delta t. \quad (5)$$

```

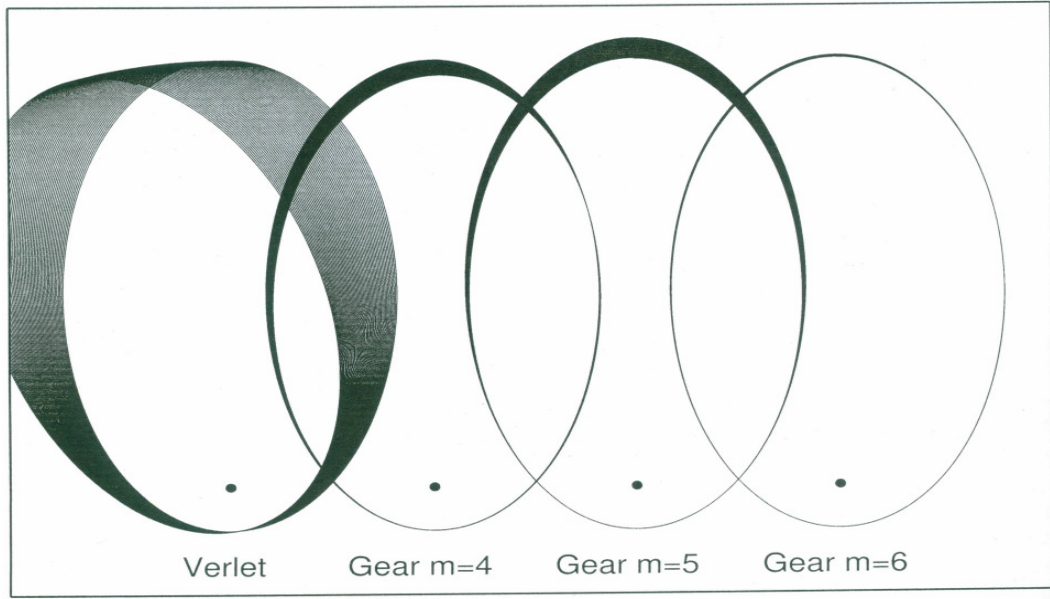
CONST      h=0.001      integrační krok
REAL t=0      čas a jeho počáteční hodnota
VECTOR r= (1 ,0)      poloha planety v čase t; vektor r se skládá ze složek (r1.x,r1.y)
VECTOR      rold, rnew
VECTOR      a           zrychlení
VECTOR      v = (0,0.5) počáteční rychlost
REAL f

Rold := r-h*v      jednodušší verze rovnice 3

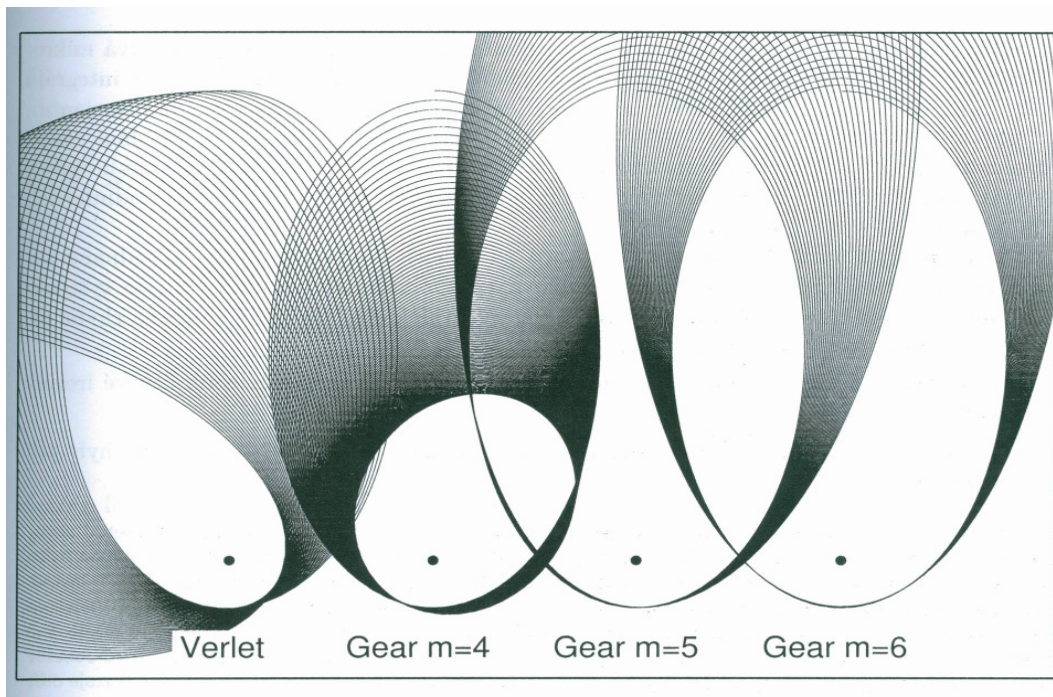
REPEAT
f := (r.x**2+r.y**2)**(3/2)
a:= -r/f
      rnew := 2*r-rold+h**2*a
      rold := r
      t := t+h
      Nakreslibod (r.x, r.y)
UNTIL t>10 skončí po 10 časových jednotkách (to jsou asi 4 oběhy)

```

Obr. 5: Algoritmus výpočet dráhy planet kolem Slunce Verletovou metodou v redukovaných jednotkách (gravitační konstanta I hmotnost planet jsou 1). Trajektorie je nakreslena na obrázcích 6 a 7. [9]



Obr. 6: Trajektorie planet kroužící kolem Slunce vypočítané různými metodami [10]



Obr. 7: jako obr. 6 ale s dvojnásobkem kroku [11]

3.3 Metoda Monte Carlo

Monte Carlo je třída algoritmů pro simulaci systémů. Jde o stochastickou metodu používající pseudonáhodná čísla. Typicky využívány pro výpočet integrálů, zejména vícerozměrných, kde běžné metody nejsou efektivní. Metoda Monte Carlo má široké využití od simulací experimentů přes počítání určitých integrálů až třeba řešení diferenciálních rovnic. Základní myšlenka této metody je velice jednoduchá, chceme určit střední hodnotu veličiny, která je výsledkem náhodného děje. Vytvoří se počítačový model tohoto děje a po proběhnutí dostatečného množství simulací se mohou data zpracovat klasickými statistickými metodami, třeba určit průměr a směrodatnou odchylku. [1]

Jako příklad pro vysvětlení principu metody MC uvedeme výpočet čísla π , je jednoduché a pro žáky velice motivující. Představme si, že máme čtverec o hraně velikosti 2, do něhož je vepsán kruh o maximálním obsahu (viz obr. 9). Strany čtverce necht' jsou rovnoběžné s matematickými osami kartézské soustavy souřadné, počátek souřadné soustavy leží ve středu kružnice. Každý bod uvnitř čtverce je definován uspořádanou dvojicí čísel $[\gamma_1, \gamma_2]$, přičemž $\gamma_i \in \langle -1, 1 \rangle$ je náhodné číslo. Podíl počtu čísel ležících uvnitř kruhu N_{kruh} a všech čísel generovaných uvnitř čtverce N_{celkem} je roven podílu obsahu kruhu S_{kruh} a obsahu čtverce $S_{ctverec}$, tj. [4]

$$\frac{N_{kruh}}{N_{celkem}} = \frac{S_{kruh}}{S_{ctverec}} = \frac{\pi r^2}{S_{ctverec}}. \quad (6)$$

Známe-li počet generovaných čísel a poloměr kruhu, pak lze pro číslo π psát

$$\pi = \frac{S_{ctverec} N_{kruh}}{r^2 N_{celkem}}. \quad (7)$$

```

polomer=1;
N_celkem=1000000;

gamma1=2*rand(N_celkem, 1)-1;
gamma2=2*rand(N_celkem, 1)-1;

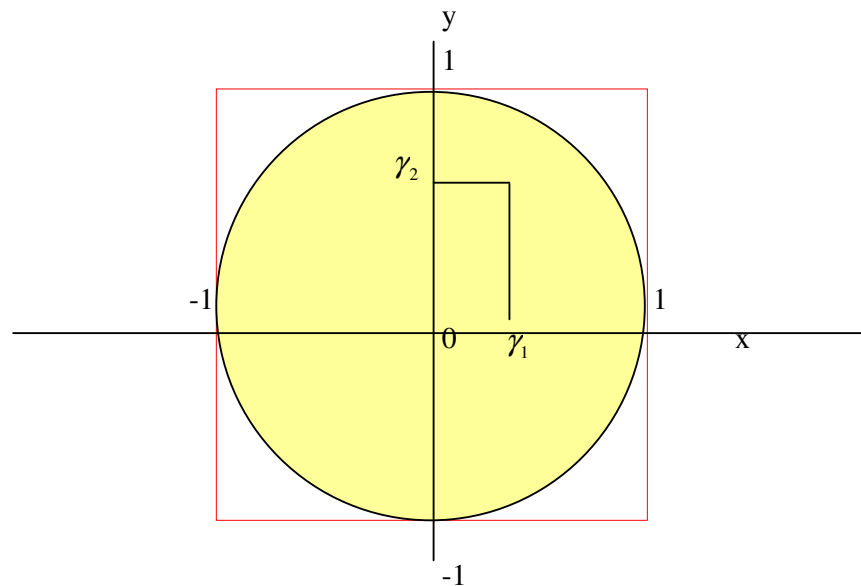
v_kruhu=find((gamma1.^2+gamma2.^2)<polomer.^2);
N_kruh=size(v_kruhu, 1);

pi=4*N_kruh/(N_celkem*polomer.^2)

```

Obr.8. Zdrojový kód [11]

Přibližná hodnota daná výpočtem činí po provedení 3,1419, výpočet trval na běžném počítači necelou sekundu.



Obr. 9: Výpočet π metodou Monte Carlo [11]

Nejstarší aplikací metody Monte Carlo je tzv. Buffonova úloha o jehle. V této úloze se určuje hodnota čísla π . Nakreslíme na vodorovné rovině rovnoběžky ve vzdálenosti d šířky a o délce l , která bude menší nebo rovna šířce rovnoběžek. Úspěšný pokus bude

takový, když náhodně hozená jehla protne rovnoběžku, ostatní pokusy budou neúspěšné. Pokus zopakujeme N -krát a určíme číslo M / N . M je počet úspěšných pokusů. Dalším počítáním najdeme limitní vztah mezi M / N pro nekonečný počet pokusů a délkami l a d . V konstantě úměrnosti v tomto vztahu je obsaženo číslo π . Pak zjištěním poměr M / N experimentálně určíme hodnotu konstanty. Tato metoda je ale značně nepřesná zvýšením počtu pokusů jsme schopni odhadnout hodnotu π 3,14 až po 100 000 pokusech. [1]

3.4 Metody spojitého modelování

3.4.1 Stručný popis spojitého modelování

Spojité model nahrazuje diskrétní částice spojitými veličinami, které popisují jejich chování jako tekutinu. Místo souřadnic konkrétních částic, jejich energií a podobně zavádíme odpovídající koncentraci, hustotu energie a podobně. Základním nedostatkem spojitých modelů je jejich z pravidla menší přesnost.

- Komplikovanější vytváření programů
- Zpravidla rychlejší výpočet než u částicových metod

3.4.2 Příklad spojitého modelu

Na stránkách www.fit.vutbr.cz vidíme názorný příklad spojitého modelování :

Jako příklad spojité simulace budeme uvažovat systém kola automobilu. Experiment bude sledovat odezvu kola na jednotkový skok. Tlumené kmitání kola je popsáno rovnicí:

$$M\ddot{x} + D\dot{x} + kx = F(t) \quad (8)$$

kde $F(t)$ je vstupní budicí funkce (např. jednotkový skok) a

k, D, M jsou konstantní parametry systému kola

Rovnici převedeme na soustavu diferenciálních rovnic prvního řádu

$$\dot{v} = \frac{(F - Dv - Ky)}{M} \quad (9)$$

$$\dot{y} = v$$

v je rychlost pohybu kola,

y je výchylka kola z klidové polohy,

Spojité model popisujeme propojením funkčních bloků - objektů modelu. Každý objekt je inicializován tak, že prvním parametrem jeho konstruktoru je jeho vstup. Na místě vstupu může být výraz, ve kterém lze použít objekty (jako proměnné) nebo číselné hodnoty (jako konstanty). [13]

3.5 *Metody hybridního modelování*

Se slovem *hybridní* se setkáváme v mnoha oblastech lidské činnosti. Zpravidla jej používáme tam, kde chceme zdůraznit, že jsme využili ty lepší vlastnosti z každé části celku - při křížení v zahradnictví, v automobilovém průmyslu a podobně. Jako snadno představitelný příklad nám poslouží *hybridní trolejbus*. Trolejbus i autobus všichni známe a víme, jaké jsou jejich výhody a nevýhody. Trolejbus je tichý a šetrný k městskému ovzduší, ale je závislý na trolejovém vedení. Autobus je nezávislý, ale v dopravní zácpě dokáže vydatně přispět ke smogu. Zkusme spojit výhody autobusu a trolejbusu. Od takového vozidla budeme čekat alespoň částečnou nezávislost i šetrnost k prostředí města. Řešení může mít různé podoby - například hybridní trolejbus. Do trolejbusu zkrátka zabudujeme spalovací motor. Ve městě bude trolejbus napájený z trolejového vedení a na vesnici pojede na naftu. Zdánlivě je to jednoduché, ale k úspěšnému výsledku vede nelehká cesta. Musíme sestrojít složitější převodovku, vyrovnat se s vyšší celkovou hmotností a podobně. S obdobnými problémy se setkáváme v modelování ve fyzice. Některé metody jsou velmi přesné a časově náročné, jiné dávají výsledky nesrovnatelně rychleji, ale s velkou chybou. Hybridní modely by měly vést k co nejlepším výsledkům v přiměřeném čase. Dosahují toho kombinací různých přístupů k jednomu fyzikálnímu problému. Obvykle však nelze převzít jen ty lepší vlastnosti, a proto musíme hledat kompromisní řešení. Hybridní modely jsou nejčastěji využívány ve fyzice nízkoteplotního plazmatu. Existuje několik typů, přičemž my si představíme tři nejčastější: [14]

3.5.1 Energetický model

Hranice mezi částicovým a spojitým modelem je dána nějakou významnou hodnotou energie elektronu. Elektrony s vyšší energií, než je tato prahová energie, podstupují nepružné excitační a ionizační srážky, což narušuje elektronovou rozdělovací funkci, která je pro nižší energie například maxwellovská. Elektrony tedy rozdělíme na „pomalé“ a „rychlé“ podle jejich energie. Tato metoda je velmi výhodná, pokud v pracovní oblasti jen velmi málo elektronů přesahuje prahovou energii danou vlastnostmi neutrálů. Pomalé elektrony jsou pak simulovány spojitě, rychle částicově. [14]

3.5.2 Prostorový model

V řadě řešených problémů pozorujeme, že jen relativně malá část zkoumané oblasti vyžaduje výpočet částicovým modelem. Nabízí se tedy myšlenka rozdělit pracovní oblast na dvě části. Tam, kde požadujeme přesnější výpočet, použijeme částicový model a zbytek oblasti popíšeme spojitým modelem. Pokud částicový model pracuje na malé oblasti, můžeme výpočet tímto způsobem výrazně urychlit. Stejně jako v energetickém hybridním modelu nemusí být délka časového kroku stejná pro obě části modelu. [14]

3.5.3 Iterační model

Jedná se o model s vnitřními reakcemi např.: pružný rozptyl, přenos náboje nebo chemické reakce. Tento model je vhodný především pro hledání pohybu částic v plazmatu. Příkladem takového problému je interakce substrátu s plazmatem v pozitivním sloupci doutnavého výboje. Metoda není vhodná pro řešení problémů s výraznými změnami v čase, například iontová implantace skokovou změnou napětí.

4 Moderní trendy v počítačovém modelování

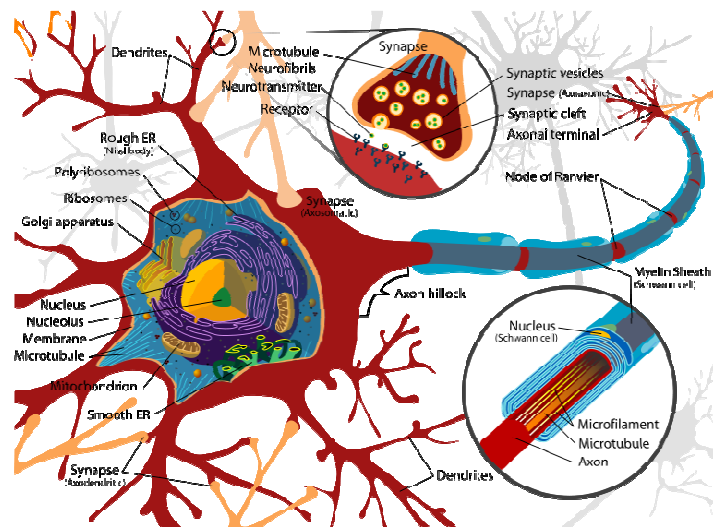
Technika prochází neustálým vývojem, proto si už dnes můžeme pomocí počítačového modelování představit například věci, jako je umělý neuron.

4.1 Neuronové sítě

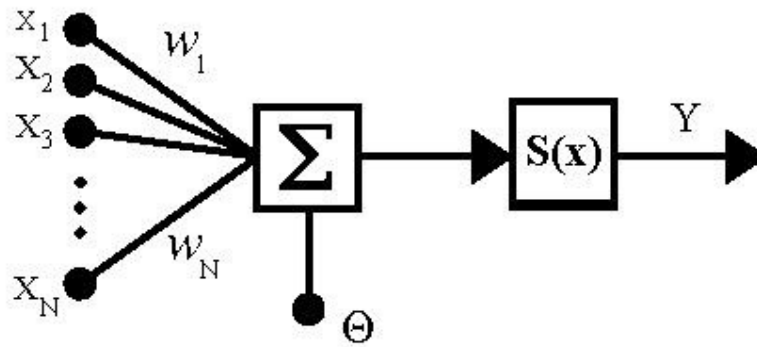
Neuronová síť je jedním z výpočetních modelů používaných v umělé inteligenci. Jejím vzorem je chování odpovídajících biologických struktur. Umělá neuronová síť je struktura určená pro distribuované paralelní zpracování dat. Je složena z umělých neuronů, jejichž vzor představuje biologický neuron. [15]

Neurony jsou vzájemně propojeny a navzájem si předávají signály a transformují je pomocí určitých přenosových funkcí. Neuron má libovolný počet vstupů, ale pouze jeden výstup. [16]

Neuronové sítě mají mnoho využití např. předvídání časových řad (burzovní index) a dokonce i filtrování spamů. Neuronové soustavy nám pomáhají i v lékařství, kdy pomocí nich můžeme lépe studovat fungování nervových soustav živých organismů. [15]



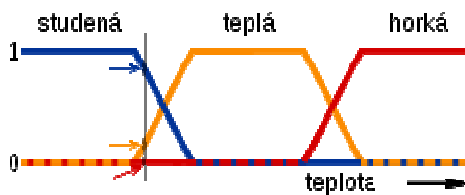
Obr. 10: Biologický neuron [16]



Obr.11: Umělý neuron [17]

4.2 Fuzzy logika

Na rozdíl od výrokové logiky používající pouze hodnoty 0 nebo 1 (pravda nebo nepravda) fuzzy logika pracuje se všemi hodnotami intervalu $j \langle 0;1 \rangle$.



Obr.12: Fuzzy logika [19]

Fuzzy v angličtině znamená neurčitý, z toho je odvozen název. Díky fuzzy logice si můžeme představit i pojmy jako je trochu nebo dost.

- Překlad fuzzy logiky -Anglické slovo fuzzy – chundelatý, chomáčkovitý, neostrý
- Vysvětlení pojmu stupeň příslušnosti - Funkce příslušnosti ve fuzzy logice umožňuje přiřadit příslušnost k množinám v rozmezí od 0 do 1, včetně obou hraničních hodnot. Fuzzy logika tak umožňuje matematicky vyjádřit pojmy jako „trochu“, „dost“ nebo „hodně“. Přesněji, umožňuje vyjádřit částečnou příslušnost k množině. Stupeň příslušnosti je často zaměňován s pravděpodobností. Tyto pojmy jsou ale rozdílné. Fuzzy hodnota je přiřazena funkcí příslušnosti k vágně definovaným množinám a nepředstavuje

pravděpodobnost nějakého jevu, stejně tak u ní nejde o možnosti, které mohou nastat a možnosti, které nastanou. [20]

Dalším vědeckým oborem , který pracuje s fuzzy logikou je kvantová fyzika využívající také stavy, kde je výsledek měření možný jen v rámci pravděpodobnosti. [20]

4.3 Distribuovaný výpočet

Distribuovaný výpočet je výpočet rozdělený na více menších, méně náročných úloh za účelem rychlejšího vyřízení požadavku předaného programu. Lze ho využít jen u výpočtů, jejichž algoritmus lze paralelizovat – převést na paralelní verzi, kdy vzájemně nezávislé části výpočtu běží současně.

Výpočet lze distribuovat buď na úrovni operačního systému s přesměrováním softwarových vláken na jiné členy clusteru, nebo přímo v režii programu, který se nainstaluje v podobě mnoha klientů na každý z počítačů tvořících cluster.

Mezi typické výpočty, které je vhodné řešit distribuovaně, patří analýza velkého množství statistických dat, „zpětné inženýrství“ DNA, modelování struktury proteinů, generování fraktálů nebo zkoumání vesmírného vlnění na přítomnost rádiového signálu mimozemšťanů (SETI). [22]

Další moderní trendy:

- Modelování chůze dětí trpících obrnou
- Modelování hvězd

5 Programovací jazyk MATLAB

MATLAB je integrované prostředí pro vědeckotechnické výpočty, modelování, návrhy algoritmů, simulace, analýzu a prezentaci dat, paralelní výpočty, měření a zpracování signálů, návrhy řídicích a komunikačních systémů. MATLAB je nástroj, jak pro pohodlnou interaktivní práci, tak pro vývoj širokého spektra aplikací.

Výpočetní systém MATLAB se během uplynulých let stal celosvětovým standardem v oblasti technických výpočtů a simulací ve sféře vědy, výzkumu, průmyslu i v oblasti vzdělávání. MATLAB poskytuje svým uživatelům nejen mocné grafické a výpočetní nástroje, ale i rozsáhlé specializované knihovny funkcí spolu s výkonným programovacím jazykem čtvrté generace. Knihovny jsou svým rozsahem využitelné prakticky ve všech oblastech lidské činnosti. [23]

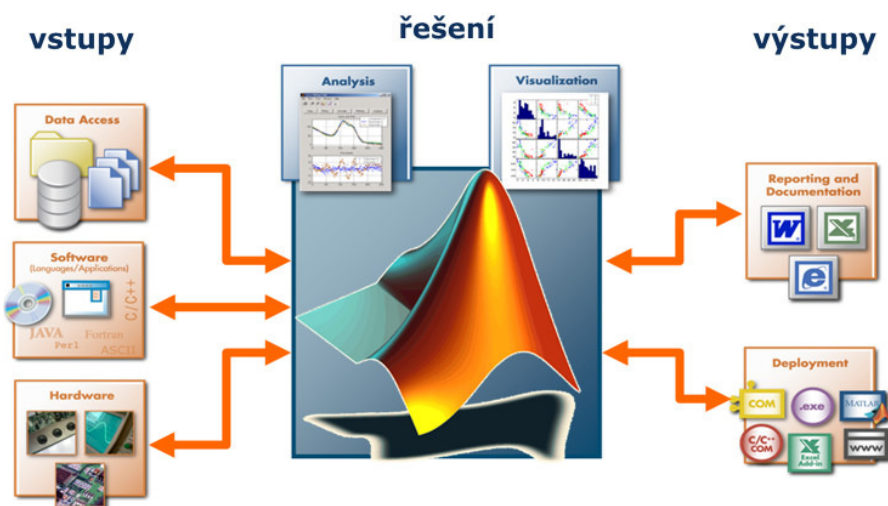
Detailní charakteristiku programu je možné nalézt na internetových stránkách www.humusoft.cz. MATLAB používají zejména ti, kteří se zabývají náročnými matematickými výpočty a nemají čas se věnovat jádru matematického problému. MATLAB vyniká také svou jednoduchostí oproti FORTRANU nebo programovacímu jazyku C a v neposlední řadě také velkým množstvím využití. Za největší přednost MATLABu vidíme velmi rychlé výpočetní jádro a vyhovující algoritmy, se kterými léta pracují vyhlášené firmy po celém světě. [23]

Výhody MATLABU můžeme shrnout do několika bodů:

- rychlé výpočetní jádro
- podpora paralelních výpočtů
- akcelerace výpočtů - JIT (Just in Time adaptive compilation)
- otevřený a rozšiřitelný systém
- působivá 2D a 3D grafika
- konfigurovatelné uživatelské rozhraní Matlab Desktop
- velké množství aplikačních knihoven
- programovací jazyk 4. generace
- objektové programování
- integrace s jazykem Java
- podpora vícerozměrných polí a uživatelsky definovaných datových struktur

- interaktivní nástroje pro tvorbu grafického uživatelského rozhraní
- interaktivní průvodce importem dat
- zvukový vstup a výstup, animace
- komunikace s externím přístrojovým vybavením (sériová linka, GPIB, VISA)
- výpočetní jádro pro programy psané ve Fortranu a jazyce C
- distribuce nezávislých uživatelských aplikací: překlad do jazyka C, runtime modul, WWW technologie
- rozsáhlá dokumentace v pdf nebo v on-line hypertextové formě

[22]



Obr.13 MATLAB schéma [23]

5.1 Výpočetní jádro

Když se zaměříme na již zmiňovanou nejdůležitější část MATLABu dozvídáme se, že máme možnost pracovat se základními operacemi (determinant, násobení, inverze atd.) tak i s pracnějšími typy například více rozměrná pole reálných nebo komplexních čísel. MATLAB také zvládne pracovat s poli buněk (jako matice), ale v tomto případě může být každý prvek jiného typu. Dále si také uživatel může určit přesnost ukládání dat dle jeho potřeby. MATLAB totiž standartně ukládá čísla ve dvojitě přesnosti.

V neposlední řadě využijeme MATLAB při práci s objekty, operacemi, polynomy, vyhledávání extrémů, výpočet střední hodnoty, atd. [23]

Práce s objekty nám dává možnost rozvětvit výpočetní prostředí o další datové typy, pro které můžeme dále zadávat nové funkce a operátory. Výpočetní jádro je realizováno s použitím základních matematických knihoven s podporou více jader. K tomu nám postačí základní verze MATLABu. [23]

5.2 Grafický subsystém

Grafika MATLABu je rozmanitá. Používá snadné zobrazení výsledků, které byly vypočítány. Dále zpracovává různé grafy funkcí. Můžeme se setkat jak s dvourozměrnými grafy s jednou proměnnou, tak trojrozměrné grafy pro funkce dvou proměnných, klasické koláčové grafy, histogramy a mnoho dalších. Tento subsystém dokáže u všech grafických znázornění okamžitě měnit vzhled a to ihned při vývoji práce tak i po ní. Když tvoříme světelné obrazce dá se zde použít stínování dopadajícího světla, různé animace atd. Potřebujeme-li obrázek dále ovládat MATLAB umožňuje do obrázku vložit tlačítka a mnoho dalšího. [23]

5.3 Otevřená architektura

Vlastností, která patrně nejvíce přispěla k rozšíření MATLABu, je jeho otevřená architektura. MATLAB je úplný programovací jazyk, to znamená, že uživatelé v něm mohou vytvářet funkce "šité na míru" pro jejich aplikace. Tyto funkce se způsobem volání nijak neliší od vestavěných funkcí a jsou uloženy v souborech v čitelné formě. Dokonce většina funkcí s MATLABem dodávaných je takto vytvořena a opravdu

vestavěné jsou jen funkce základní. To má dvě velké výhody: jazyk MATLABu je téměř neomezeně rozšiřitelný a kromě toho se uživatel může při psaní vlastních funkcí poučít z dodaných algoritmů. Navíc jsou takto koncipované funkce snadno přenosné mezi různými platformami, na kterých je MATLAB implementován. Všechny moduly systému doprovází rozsáhlá pdf i hypertextová on-line dokumentace, která uživatelům usnadňuje orientaci ve funkcích MATLABu. Otevřená architektura MATLABu inspirovala mnoho nezávislých firem k vývoji a distribuci vlastních produktů, které buď rozšiřují výpočetní prostředí MATLAB o další knihovny a nástroje nebo zajišťují propojení MATLABu s jinými specializovanými programy. [23]

5.4 Pracovní nástroje

MATLAB je přizpůsoben programování aplikací, má všechny potřebné příkazy pro psaní programů např. cykly , podmíněné příkazy, větvící příkazy atd. Již v MATLABu zmiňovaná jednoduchost a zvladatelnost řadí MATLAB k programovacím jazykům čtvrté generace, kde tvoření nekladou překážku ani velmi složité aplikace. MATLAB Desktop považujeme za základní nástroj systému. Do prostředí jsou vloženy nástroje jako prohlížeč adresářů a souborů, prohlížeč pracovního prostoru, okno historie příkazů, spouštěč aplikací atd. Uživatel si může sám určit rozměry a počet zobrazených nástrojů k jeho spokojenosti. Jestliže chceme tvořit aplikace, obrátíme se na nástroj pro vytváření uživatelských rozhraní, tam lze snadno utvořit a uspořádat ovládací prvky aplikace. Dalším pomocným nástrojem je nástroj pro import dat, který velmi pomůže při načítání dat z jakéhokoli zdroje, jako je např. tabulka, texty, obrázky animace atd. MATLAB se skvěle dorozumívá i s ostatními programovacími jazyky jako je C, FORTRAN nebo Java. Z programu Java dokonce můžeme objekty aplikovat v prostředí MATLAB. To je velkou výhodou zejména pro vyhotovování složitých grafických rozhraní. [23]

5.5 *Toolboxy*

Otevřená architektura MATLABu vedla ke vzniku knihoven funkcí, nazývaných toolboxy, které rozšiřují použití programu v příslušných vědních a technických oborech. Tyto knihovny, navržené a v jazyce MATLABu, nabízejí předzpracované specializované funkce, které je možno rozšiřovat, modifikovat, anebo jen čerpat informace z přehledně dokumentovaných algoritmů. [23]

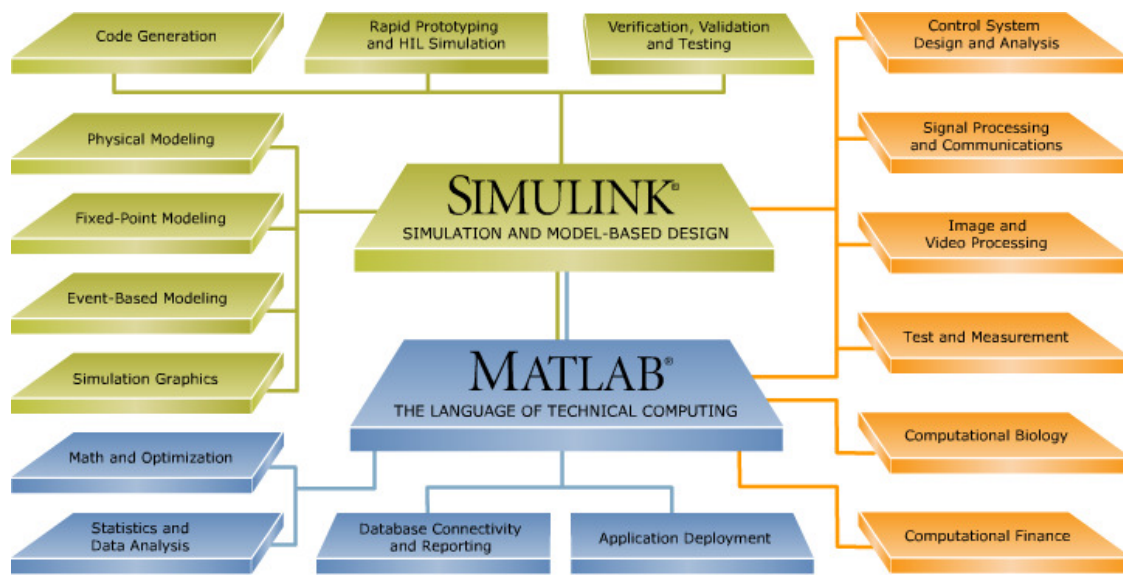
Oblasti využití jsou popsány také na stránkách www.humusoft.cz :

- Aplikovaná matematika
- Automatické řízení a regulace
- Zpracování signálu a komunikace
- Zpracování obrazu
- Měření a testování
- Výpočetní biologie
- Finanční modelování a analýza
- Modelování fyzikálních soustav

5.6 *Aplikační knihovna, simulink*

Otevřená architektura MATLABu a Simulinku vedla ke vzniku knihoven funkcí a bloků, nazývaných toolboxy a blocksety, které rozšiřují použití programů v příslušných vědních a technických oborech. [24]

Kliknutím na název knihovny (obr.14) nám MATLAB nabídne podrobný popis daného tématu i s názornými obrázky.



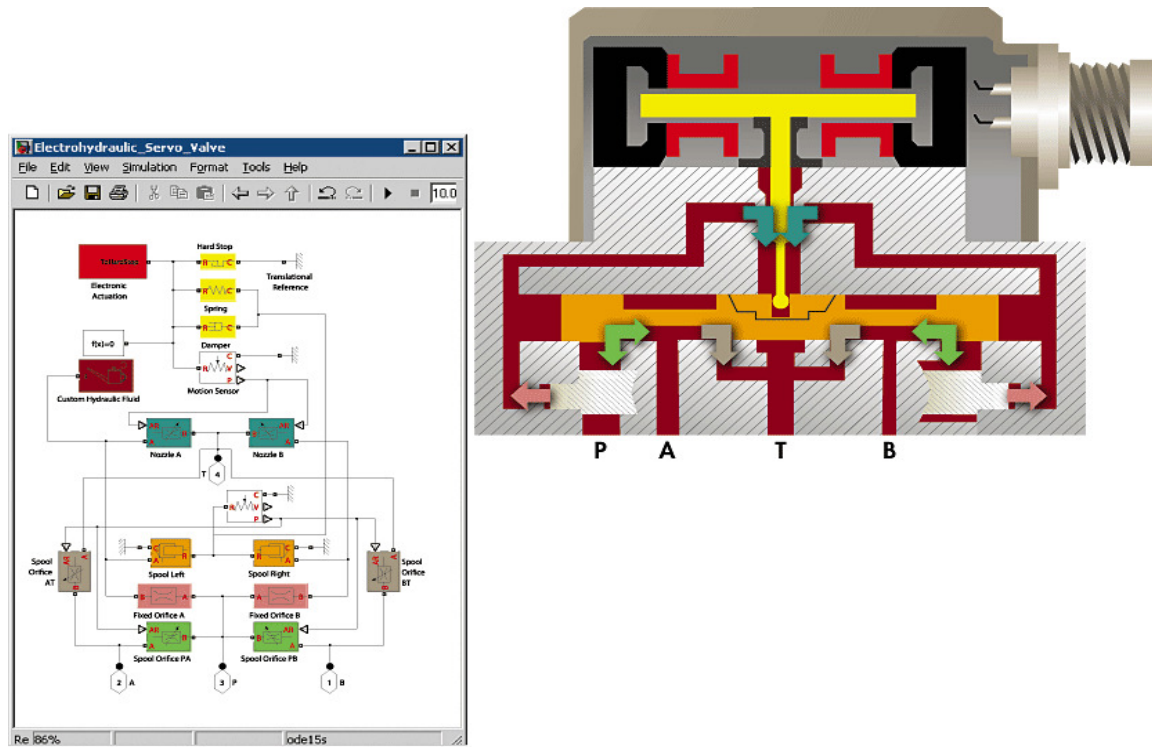
Obr.14 : Přehled aplikačních knihoven [24]

Viz následující příklad

Klikneme-li např. na fyzikální modelování (Physical Modeling) otevře nám MATLAB řadu příkladů modelování od modelování fyzikálních systémů přes modelování v mechanice, energetiky a elektroniky až po hydraulické soustavy atd.

Vybereme tedy hned první případ – modelování a simulace fyzikálních systémů a výsledek vypadá takto : [24]

Simscape rozšiřuje Simulink o nástroje pro modelování a simulace tzv. "multi-domain" systémů obsahujících propojení mechanických, elektrických a hydraulických komponent. Simscape využívá nový přístup k modelování systémů. Zavádí do simulačních schémat reálné fyzikální veličiny, jako jsou síly, momenty, napětí, proudy, tlaky, průtoky atd. Podobně jako při montáži reálného systému, vzniká model v Simscape grafickým propojením bloků, které přímo odpovídají fyzickým prvkům reálného systému. Bloky se spojují do sítě, ve které spojení mezi elementy odpovídají přenosům energie v systému. Tento přístup umožňuje systémy modelovat přímo popisem jejich fyzické struktury a odbourává se potřeba odvozování příslušných matematických vztahů mezi sledovanými veličinami. Simscape tyto vztahy generuje automaticky. [24]



Obr.15 : Model fyzikálního systému [25]

Simscape obsahuje knihovny základních mechanických, elektrických a hydraulických bloků, které umožňují vytvářet uživatelské bloky a schémata komplexnějších komponent systému. Využití Simscape je široké, uplatnění najde v automobilovém průmyslu, letectví, obraně, návrhu průmyslových a stavebních strojů a podobně.

Simscape obsahuje i vlastní jazyk, založený na jazyku MATLABu. Lze jej využít k tvorbě uživatelských komponent, rozšiřujících knihovny prvků v dané fyzikální doméně, nebo k tvorbě celých uživatelsky definovaných fyzikálních domén.

Jak tedy vidíme MATLAB zvládne všechny vědní obory je to nerozšířenější program na modelování a simulace, má skvělou grafiku s širokými možnostmi využití. Je poměrně jednoduchý na obsluhu a je prověřený a celosvětově nejpoužívanějším programem. [24]

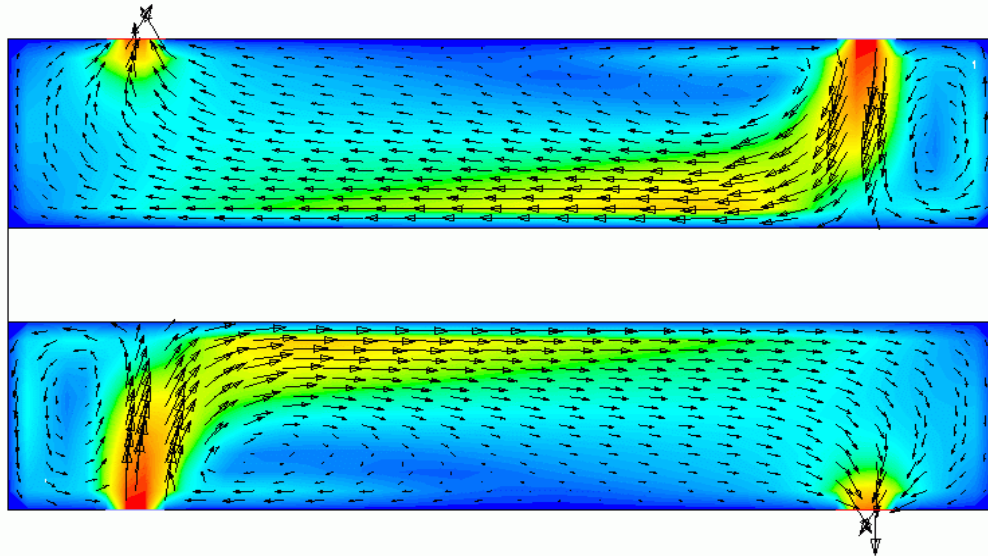
6 Program Fluent

FLUENT CFD (computational fluid dynamics) je systém pro numerické řešení problémů z oblasti dynamiky tekutin. Je to nový moderní program pro řešení úloh počítačového modelování. Jeho možnosti jsou rozmanité. Můžeme modelovat proudění plynů, přenos tepla, spalování, vnitřní i vnější obtékání, více fázové proudění, turbulence, probublávání ropné plošiny, výroba polovodičů, tok krve, až to možnosti modelovat multifyzikální úlohy. Možno použít v letectví, automobilovém průmyslu, chemickém inženýrství, technice životního prostředí atd.[26] [27]

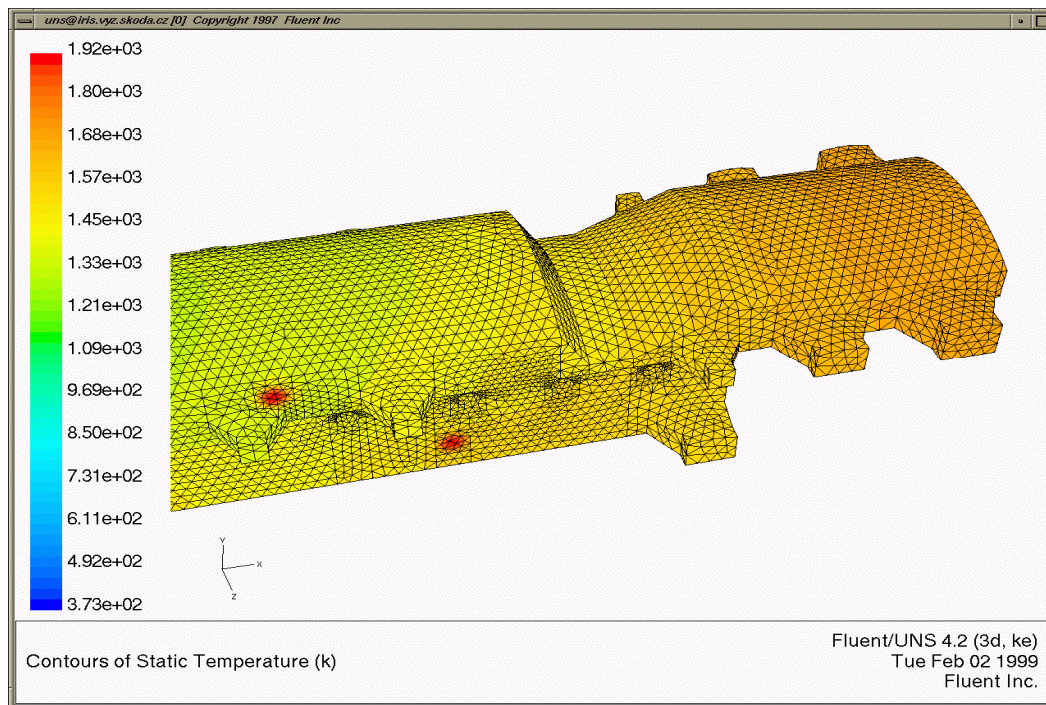
V dnešní době má mnoho společností úspěch díky využívání programu ANSYS FLUENT. Vylepšení řešiče nám nabízí flexibilní pohyb, deformaci sítě a rychlé a přesné výsledky. Uživatel má možnost zadávat funkce a tím ovlivňovat nové modely. Další funkce programu ANSYS FLUENT nám umožňují si výpočet kdykoli pozastavit a zkontrolovat výsledky či nastavení a po té pokračovat ve výpočtu ovšem jednom spuštění programu. Také datové soubory můžeme nahrát do programu ANSYS CFD-Post kde s pokročilými nástroji podrobně sledujeme a porovnáváme například jiné simulace vedle sebe.[26]

Využití výpočetního systému FLUENT pro řešení úloh dynamiky tekutin a zjištění rozložení teplot, tepelných toků, rychlosti proudění a tlaků v technologických systémech, kde dochází k přenosu tepla mezi proudící tekutinou a povrchem pevných těles. [28]

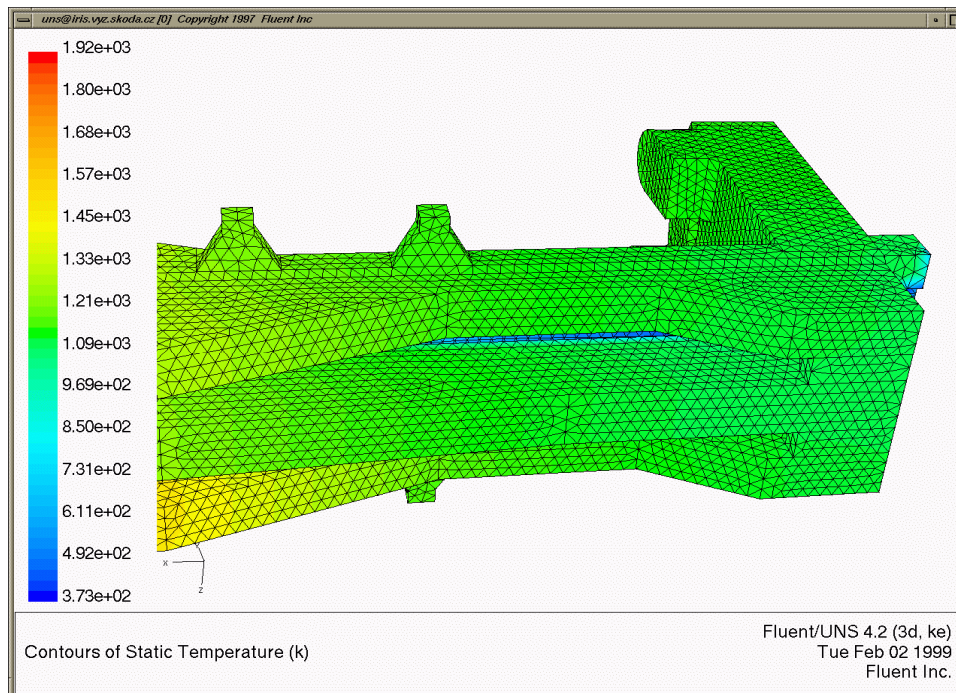
GAMBIT je integrovaný objemový modelář a preprocesor výpočetních sítí všech řešičů Fluent Inc. Umožňuje načtení geometrie z jiných formátů, top-down (od objemu k bodu) i bottom-up (od bodu k objemu) objemové modelování a automatické opravy importované geometrie. [29]



Obr.16: Zobrazení dynamiky tekutin [30]



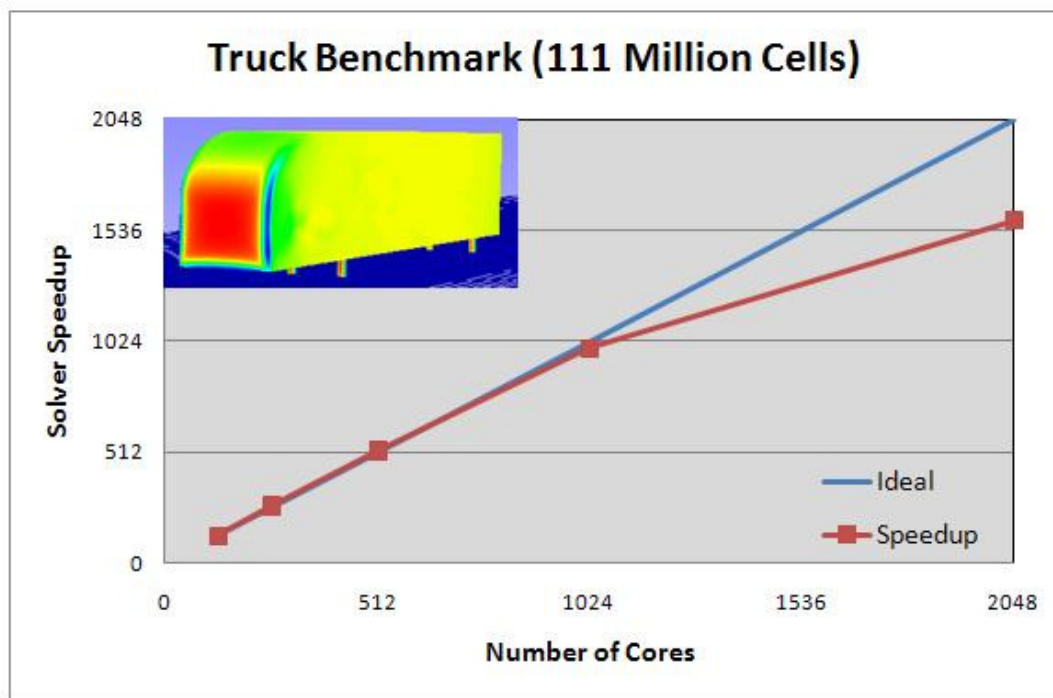
Obr. 17: GAMBIT- objemové modelování [31]



Obr. 18: GAMBIT- objemové modelování [32]

6.1.1 Síťování a numerika, paralelní zpracování

Zde se používá nestruturovaných sítí. Sítě mohou být tvořeny z částí ve tvaru trojúhelníků, čtverců, obdélníků pro 2D simulace, šestiúhelníků mnohoúhelníků pro 3D simulace. Pracuje spolehlivě se všemi fyzikálními modely a typy proudění – nestlačitelné, hypersonické, stacionární, nestacionární. Program nám poběží v kterémkoli z operačních systémů Windows, Linux či Unix. Průběh programu umožňuje vícejádrový a víceprocesorový stroj nebo počítačový klastr. [26]

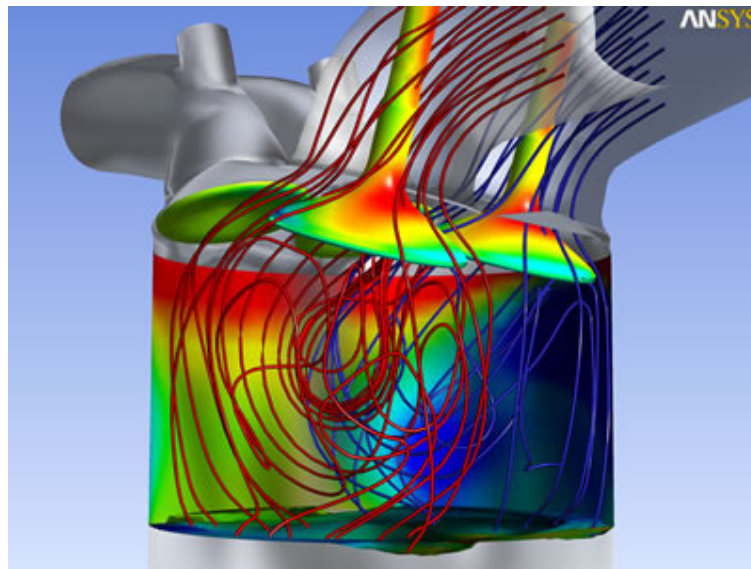


Obr.č. 19: Škálování výpočtu programem ANSYS FLUENT 12 je blízko ideálu až do 1 024 procesorů/jader a na 78% ideálního stavu při výpočtu na 2 048 procesorů/jader. [33]

6.1.2 Dynamické a pohybující se sítě

Další možností jak využít ANSYS FLUENT je simulace spalování a proudění například ve válcích motoru. Můžeme použít několik schémat pro výměnu sítě během výpočtu a mnoho dalšího. Na začátku před startem výpočtu musíme vytvořit jen počáteční síť.

Možnost využití dynamických sítí v programu ANSYS FLUENT vyhovuje potřebám náročných aplikací jako je simulace proudění a spalování ve válcích motoru nebo při uvolňování rakety z letadla. Lze použito několik různých schémat pro změnu sítě během výpočtu, včetně vrstvení, vyhlazování a přesíťování, při simulaci různých pohyblivých částí v jednom výpočtu. Před zahájením výpočtu je potřeba vytvořit pouze počáteční síť a zadat pohyb pohyblivých částí. Při simulacích s nenuceným pohybem můžeme využít řešič se šesti stupni volnosti. Pro dynamiku pohybu lodě na vlnách atd. [26]



Obr. 20: Modelování spalování v motoru užitím programu ANSYS FLUENT s modely pohybující se a deformující se sítě a programu ANSYS CFD-Post, který je určen pro post-procesing.[33]

6.1.3 Turbulence a akustika

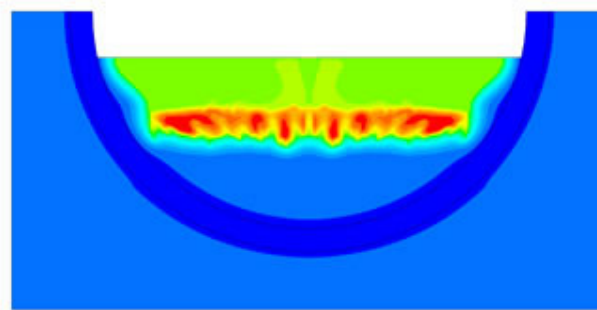
Pro zpracování modelů turbulence jsou různé verze časově středovaných k-epsilon modelů, modely k-omega a modely s řešením Reynoldsových napětí (RSM). Dnešní pokrok v modelování turbulence nás dovádí k řešení dalších modelů například: turbulentní přechodový model. [26]



Obr. 21: Vírové struktury generované koly přistávacího letadla.[33]

6.1.4 Přenos tepla

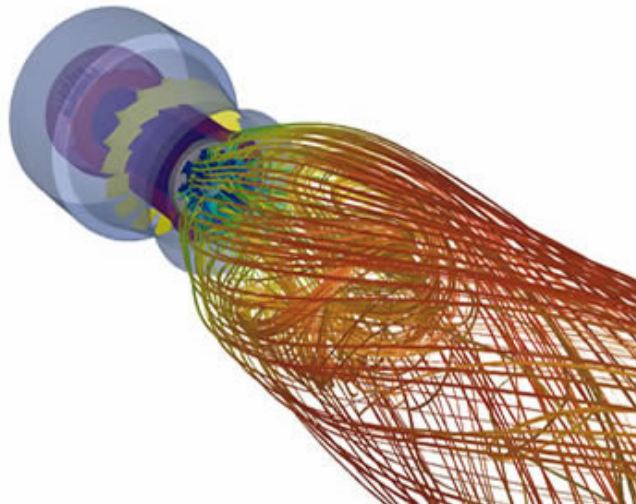
Přenos tepla je základním jevem v mnoha aplikacích proudění tekutin. ANSYS FLUENT má řadu modelů pro nastavení přenosu tepla konvekcí, kondukcí a radiací. Můžeme využít z celé řady možností například dva rozdílné modely pro simulaci výměníků tepla.



Obr. 22: Tavení a tuhnutí uranu v selhávajícím reaktoru.[33]

6.1.5 Reakce proudu

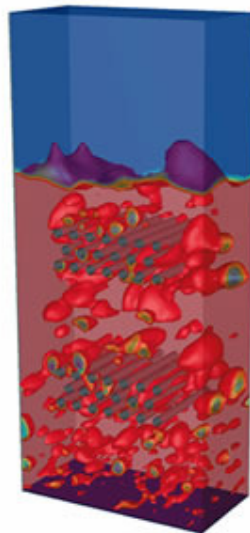
Programu ANSYS FLUENT je doménou, že modeluje speciálně chemické reakce v turbulentních podmínkách.



Obr. 23: Hořák pro spalování nízkých NO_x. [33]

6.1.6 Vícefázové proudění

Tady program ANSYS FLUENT dovoluje odborníkům pochopit podrobnosti o proudění uvnitř strojů, které se jinak velmi složitě objevují.



Obr.24: Bubliny ve fluidní vrstvě reaktoru. [33]

6.1.7 Zpracování výsledků a export dat

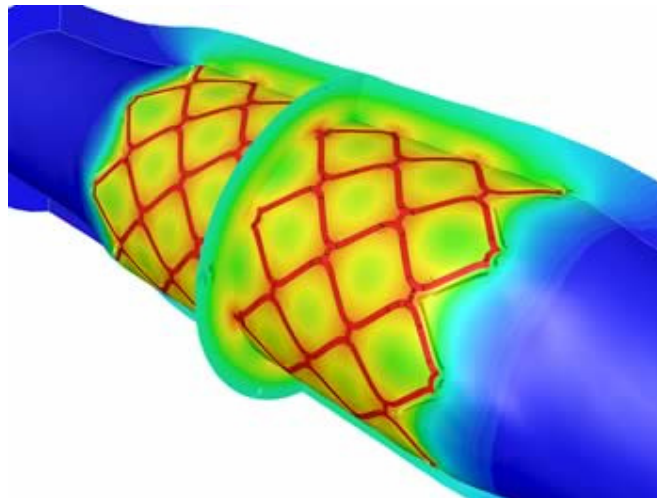
Výsledkem zpracovaných dat v programu ANSYS FLUENT může být pomocí zpracovávacích nástrojů animace, automatické reporty, grafika atd. Grafika je propracovaná a můžeme využít stínované a průhledné plochy, proudnice, vektory, kontury atd..



Obr. 25: Síť z mnohostěnů a rozložení tlaků na formuli F1 pro post-procesing s použitím programu ANSYS CFD-Post.[33]

6.1.8 Uživatelské nástroje

Funkce, které má možnost uživatel zadat jsou velmi zajímavé a hojně používané. Uživatel může upravovat a doplňovat program. Studijní materiály má k dispozici nebo profesionální technickou pomoc-například šablony.



Obr. 26 Anisotropní difuze léku od trubičkové výztuže do stěny kapiláry.[33]

7 Program COMSOL Multiphysics

COMSOL Multiphysics® je určen všem vývojářům, výzkumným i vědeckým pracovníkům a díky široké nabídce funkcí pro zobrazování vypočtených výsledků je určen také vysokým i specializovaným středním školám. Zájemci mají jedinečnou možnost názorně pronikat do podstaty fyzikálních procesů. Do řešení je možné zahrnout několik fyzikálních vlivů najednou (multifyzikální úlohy) a tak provádět komplexnější analýzu modelu. COMSOL Multiphysics® je těsně propojen s univerzálním nástrojem MATLAB® určeným pro vědecko-technické výpočty. Funkce tohoto nástroje je možné využívat například při kreslení geometrických tvarů, generování FEM sítí, při vlastním numerickém řešení nebo při konečném zpracování výsledků úlohy. [34]

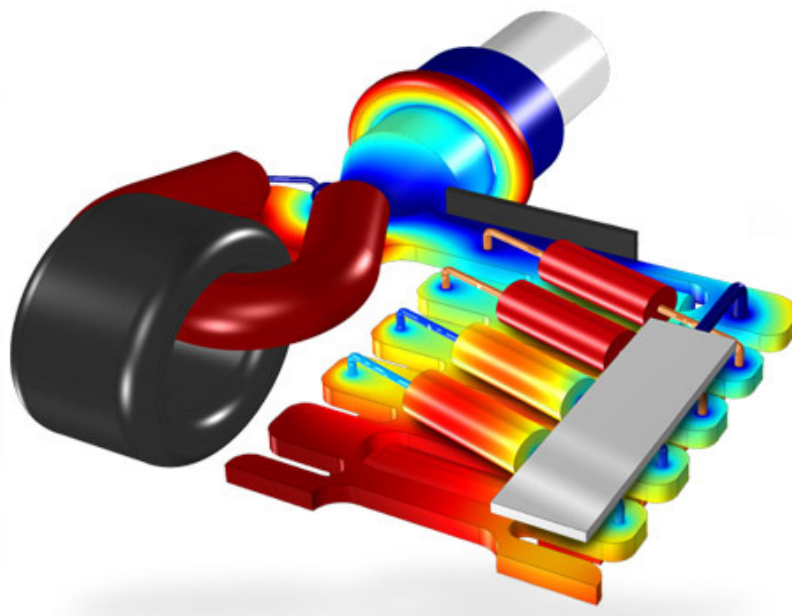
Tento program je úzce spojen s nástrojem MATLAB.

COMSOL Multiphysics® pracuje s parciálními diferenciálními funkcemi. Modelování v COMSOL Multiphysics® je podobné jako při obecném řešení parciálních diferenciálních rovnic (popis základních okrajových podmínek na plochách, definice prostředí, základní okrajové podmínky). V COMSOL Multiphysics® používáme grafický editor, v něm může být zobrazena např. strojní součást nebo reagující prostředí katalyzátoru. Uživatel si musí být vědom toho, jaké fyzikální vlivy na znázorněnou geometrii působí. Jde-li například o strojní součást určíme PDE z pružnosti a pevnosti. Máme-li případ proces zahřívání součásti volíme PDE popisující šíření tepla.

COMSOL Multiphysics® vlastní také knihovny parciálních diferenciálních rovnic popisující různé režimy. Když volíme požadovaný režim v grafickém editoru se nám automaticky objeví daná okna pro vyplnění vlastností oblastí a okrajových podmínek. Sem ale zadáváme fyzikální vlastnosti veličin např. tepelnou vodivost měrnou hustotu prostředí atd. COMSOL Multiphysics® spojuje několik aplikačních režimů do jednoho modelu bez nutnosti vygenerování přídatných kódů apod. To nám rozšiřuje možnosti v modelování kompletních úloh. S COMSOL Multiphysics® může uživatel vypracovávat své aplikační procesy použitím obecného tvaru PDE a slabých formulací pro rozdílné části modelu. Při zpracovávání těchto aplikací musíme odborně znát, co úloha obsahuje včetně matematického vyjádření. [34]

7.1 Příklad použití programu COMSOL Multiphysics®

Univerzální napájecí zdroj vytváří prostředí pro simulace a modelování výkonnosti rezistorů, kondenzátorů, tlumivek, cívek, motorů a senzorů. I když tato zařízení v první řadě charakterizuje elektromagnetismus, jsou také ovlivněny dalšími typy fyziky. Například tepelnými účinky, které mohou změnit elektrické vlastnosti materiálů, přičemž musíme uvažovat elektromechanické, elektrostatické i magnetické hodnoty v každém procesu navrhování. [35]



Obr.27 Univerzální napájecí zdroj [35]

8 Závěr

V práci jsem se pokusila seznámit čtenáře se základními poznatky počítačové fyziky. Jako základní text, ze kterého práce vychází, je použito skriptum prof. R. Hracha a kniha autorského kolektivu I. Nezbedy. Při sepisování práce byly dále využity především informace dostupné na internetových stránkách. Práce je koncipována jako přehledový text, který má za cíl doplnit popularizační formou vytvořenou internetovou prezentaci. Nekladla jsem si za cíl vytvoření čistě odborného textu, jelikož práce je určena laické veřejnosti a především pak studentům středních škol.

9 Použitá literatura

- [1] Rudolf Hrach, Počítačová fyzika I, PF UJEP Ústí nad Labem 2003
- [2] Rudolf Hrach, Počítačová fyzika I, PF UJEP Ústí nad Labem 2003 str.:7
- [3] Rudolf Hrach, Počítačová fyzika I, PF UJEP Ústí nad Labem 2003 str.:9
- [4] Rudolf Hrach, Počítačová fyzika I, PF UJEP Ústí nad Labem 2003 str.:12
- [5] Základy modelování SNTL- Nakladatelství technické literatury Praha 1989 str:12
- [6] Václav Franta, Základy modelování SNTL, Nakladatelství technické literatury Praha 1989 str.:14
- [7] I.Nezbeda, J.Kolafa, M.Kotrla, Úvod do počítačových simulací, Nakladatelství Karolinum Praha 2003
- [8]http://cs.wikipedia.org/wiki/Metoda_kone%C4%8Dn%C3%BDch_prvk%C5%A5
- [9] I.Nezbeda, J.Kolafa, M.Kotrla. Úvod do počítačových simulací, Nakladatelství Karolinum Praha 2003, str.:33
- [10] I.Nezbeda, J.Kolafa, M.Kotrla. Úvod do počítačových simulací, Nakladatelství Karolinum Praha 2003, str:38
- [11] I.Nezbeda, J.Kolafa, M.Kotrla. Úvod do počítačových simulací, Nakladatelství Karolinum Praha 2003, str:38
- [12] Tesař, J., Bartoš, P.: Metoda Monte Carlo a programovací jazyk MATLAB při přípravě učitelů na pedagogických fakultách, Technical computing Prague 2006, 14th Annual Conference Proceedings, p. 26, Praha, 2006, ISBN 80-7080-616-8.
- [13] <http://www.fit.vutbr.cz/~peringer/SIMLIB/doc/html-cz/node43.html>, 11.11.2010
- [14] Vojtěch Hrubý, Diplomová práce studium interakce plazma-pevná látka pomocí hybridního modelování 2009
- [15] http://cs.wikipedia.org/wiki/Neuronov%C3%A9_s%C3%ADt%C4%9B, 10.9.2010
- [16] http://cs.wikipedia.org/wiki/Neuronov%C3%A1_s%C3%AD%C5%A5, 8.10.2010
- [17] http://cs.wikipedia.org/wiki/Soubor:Complete_neuron_cell_diagram_en.svg, 11.11.2010
- [18] <http://cs.wikipedia.org/wiki/Soubor:NeuronModel.jpg>, 1.11.2010
- [19] http://cs.wikipedia.org/wiki/Soubor:Fuzzy_logic_temperature_cs.svg, 25.9.2010
- [20] http://cs.wikipedia.org/wiki/Fuzzy_logika, 10.12.2010
- [21] http://cs.wikipedia.org/wiki/Distribuovan%C3%BD_v%C3%BDpo%C4%8Det, 10.12.2010
- [23] <http://www.humusoft.cz/produkty/matlab/matlab/>, 11.11.2010
- [24] <http://www.humusoft.cz/produkty/matlab/aknihovny/>, 11.11.2010
- [25] http://www.humusoft.cz/produkty/matlab/aknihovny/simscape/ss_valve_wl_18843.gif, 1.12.2010
- [26] http://www.techsoft-eng.cz/sluzby/ansys_cfd/fluent/#dynamicke 13.10.2010
- [27] <http://metavo.metacentrum.cz/cs/docs/aplikace/software/fluent.html> 13.10.2010
- [28] <http://ttp.zcu.cz/cz/vyzkum/pocitacove-modelovani> 13.10.2010

- [29] <http://ttp.zcu.cz/cz/vyzkum/pocitacove-modelovani/pocitacove-modelovani-dynamiky-tekutin> 13.10.2010
- [30] <http://ttp.zcu.cz/cz/vyzkum/pocitacove-modelovani/pocitacove-modelovani-dynamiky-tekutin>, 13.10.2010
- [31] <http://ttp.zcu.cz/cz/vyzkum/pocitacove-modelovani/pocitacove-modelovani-dynamiky-tekutin>, 13.10.2010
- [32] <http://ttp.zcu.cz/cz/vyzkum/pocitacove-modelovani/pocitacove-modelovani-dynamiky-tekutin>, 13.10.2010
- [33] http://www.techsoft-eng.cz/sluzby/ansys_cfd/fluent/#reakce, 20.10.2010
- [34] <http://www.humusoft.cz/produkty/comsol/comsol/>, 25.10.2010
- [35] <http://www.comsol.com/products/acdc/>, 1.2.2011