



Pedagogická  
fakulta  
Faculty  
of Education

Jihočeská univerzita  
v Českých Budějovicích  
University of South Bohemia  
in České Budějovice

Jihočeská univerzita v Českých Budějovicích  
Pedagogická fakulta  
Katedra aplikované fyziky a techniky

Diplomová práce

# Užití numerického kódu FLASH v plazmové astrofyzice

Vypracoval: Bc. Jaroslav Brož  
Vedoucí práce: RNDr. Petr Jelínek, Ph.D.

České Budějovice 2013

*Anotace:* Diplomová práce se zabývá použitím numerických počítačových kódů při simulacích v plazmové astrofyzice. V úvodu se zabýváme základní charakteristikou Slunce, dále se zaměříme na sluneční koronu a problém ohřevu sluneční korony. V následující části se pak věnujeme programům pro simulace v plazmové astrofyzice, jejich instalaci a zobrazování výsledků pomocí vizualizačního software. V závěru práce je předvedeno užití tohoto software na vzorovém příkladě a vlastní simulaci, která provádí simulaci šíření impulsně generovaných vln ve slunečních koronálních smyčkách.

*Abstract:* My diploma thesis is focused on the use of numerical computer codes for simulation in plasma astrophysics. They will learn the basic characteristics of the Sun, a closer focus on the solar corona and the solar corona heating problem. The following section is devoted to simulation software in plasma astrophysics, their installing and displaying the results using the visualization software. In the conclusion is demonstrated using this software on a model example and a simulation that performs simulation of impulsively generated waves in solar coronal loops.

## **Prohlášení**

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.

## **Poděkování**

Touto formou děkuji svému vedoucímu práce RNDr. Petru Jelínkovi, Ph.D. za cenné rady, připomínky a pomoc při zpracování mé diplomové práce.

# Obsah

1. Úvod.....	7
2. Slunce.....	8
2.1. Základní charakteristiky Slunce .....	8
2.2. Složení Slunce .....	9
2.3. Sluneční jádro .....	9
2.4. Vrstva v zářivé rovnováze .....	10
2.5. Tachoklina.....	10
2.6. Konvektivní zóna.....	10
2.7. Fotosféra.....	11
2.8. Sluneční skvrny .....	12
2.9. Chromosféra .....	12
2.10. Koróna.....	12
2.11. Problém ohřevu sluneční korony.....	13
2.11.1. Sluneční mikroerupce .....	14
2.11.2. Magnetoakustické vlny.....	14
2.11.3. Ohmický ohřev.....	14
3. Programy pro výpočty a simulace v plazmové astrofyzice.....	15
3.2. Programy používané pro výpočty .....	15
3.3. FLASH .....	15
3.3.1. Základní vlastnosti FLASHe .....	15
3.4. Instalace programu FLASH.....	17
3.4.1. Požadavky na výběr OS .....	17
3.4.2. Přepínání mezi uživateli .....	18
3.4.3. Příprava před instalací.....	18
3.4.4. Instalace kompilátoru jazyka G++ .....	18
3.4.5. Test správné instalace G++.....	19
3.4.6. Instalace JRE - Java Runtime Environment.....	19

3.4.7.	Intel® Fortran Compiler .....	19
3.4.8.	MPICH2.....	21
3.4.9.	HDF5 .....	23
3.4.10.	Instalace programu FLASH.....	25
3.5.	Vizualizační software .....	26
3.5.1.	VisIt.....	27
3.5.2.	IDL.....	27
3.6.	Instalační skripty.....	30
4.	Výpočty úloh .....	33
4.1.	Výpočet testové úlohy .....	33
4.1.1.	Počáteční podmínky .....	34
4.1.2.	Nastavení prostředí pro výpočty, sestavení kódu a spuštění výpočtu .....	36
4.1.3.	Spouštěcí skript .....	37
4.1.4.	Zobrazení výsledků .....	37
4.2.	Vlastní simulace .....	42
4.2.1.	Výpočetní model .....	43
4.2.2.	Počáteční podmínky .....	44
4.2.3.	Nastavení parametrů výpočtu.....	46
4.2.4.	Spuštění simulačních výpočtů .....	48
4.2.5.	Výsledky výpočtů.....	49
5.	Závěr.....	54
	Literatura.....	55
	Seznam obrázků .....	57
	Seznam tabulek .....	58
	Příloha 1 - nastavení proměnných prostředí v souboru <i>BASHRC</i> . .....	59
	Příloha2 - Výpis upraveného souboru Makefile.h.....	60

# 1. Úvod

Tématem této práce je použití numerických kódů používaných v plazmové astrofyzice, obzvláště pak při simulacích dějů ve sluneční fyzice. V úvodu se seznámíme se základními vlastnostmi a složením Slunce a s problémem ohřevu sluneční korony.

Další část je věnovaná programům pro počítačové simulace, blíže se zaměřuje na program FLASH, jeho instalaci a konfiguraci. Jsou zde také uvedeny příklady programů pro zobrazování výsledků počítačové simulace.

V poslední části této diplomové práce ověřujeme správnost kroků instalace z předchozí kapitoly na testovacím příkladě, který simuluje rekonekci magnetických silokřivek. Závěr práce je věnován výpočtu vlastního příkladu, který se zaměřuje na šíření vln v koronálních smyčkách Slunce.

## 2. Slunce

### 2.1. Základní charakteristiky Slunce

Slunce je Zemi nejbližší hvězda, která se nachází v centru sluneční soustavy a je hlavním zdrojem energie sluneční soustavy. Slunce je součástí spirálovité galaxie Mléčné dráhy, v rámci vnitřního okraje ramene Orionu. Oběžná dráha Slunce od středu mléčné dráhy se pohybuje přibližně okolo 24 000 – 26 000 světelných let od středu galaxie.

Podle hvězdné klasifikace je Slunce označeno jako G2V. Tato klasifikace je založena na spektrálních charakteristikách vyzařovaného záření a G2 znamená, že povrchová teplota Slunce se pohybuje okolo 5700 K. Označení V zařazuje Slunce mezi hvězdy hlavní posloupnosti.

Střední vzdálenost Země od Slunce je přibližně 149,6 miliónů kilometrů, což je vzdálenost definovaná jako 1 astronomická jednotka (AU). Na této střední vzdálenosti trvá světlu okolo 8 minut a 19 sekund, než překoná tuto vzdálenost ze Slunce na Zemi.

Slunce vzniklo společně se sluneční soustavou přibližně před 5 miliardami let z mezihvězdného mraku molekul vodíku. Toto mračno se postupně smršťovalo, až došlo k následnému zhroucení a díky dostatečné střední teplotě se iniciovala fúzní termonukleární reakce.

Chemické složení Slunce je tvořeno především z 92,1 % vodíku, 7,8 % helia a 0,1 % dalších těžších prvků jako je C, N, O, Si, Ne, Mg, S, Fe.

Další základní vlastnosti Slunce jsou uvedeny v tabulce 1.

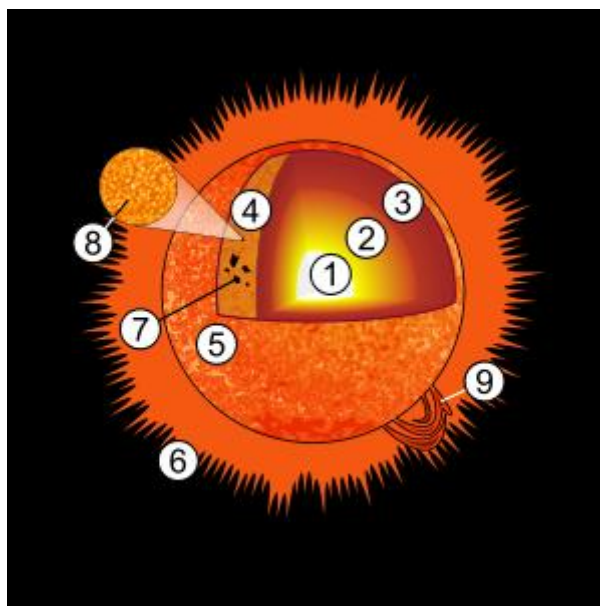
**Tab. 1:** Základní fyzikální vlastnosti Slunce

Průměr	1392000 km
Hmotnost	$1,989 \times 10^{30}$ kg
Průměrná hustota	1 409 kg/m <sup>3</sup>
Tíhové zrychlení na povrchu	274,0 m/s <sup>2</sup>
Úniková rychlost	617,7 km/s <sup>1</sup>
Perioda synodické rotace(rovník)	27,3 dnů
Vzdálenost od Země 1 AU	149 597 870 km
Zářivý výkon	$3,8 \times 10^{26}$ W
Stáří	$4,6 \times 10^9$ let



## 2.2. Složení Slunce

Na obrázku 1 je zobrazen průřez Sluncem, kde jsou znázorněny jednotlivé jeho vrstvy.



**Obr. 1:** Znázornění struktury Slunce, 1. Jádru, 2. Vrstva v zářivé rovnováze, 3. Konvektivní zóna, 4. Fotosféra, 5. Chromosféra, 6. Koróna, 7. Sluneční skvrny, 8. Granulace, 9. Protuberance; Převzato z [1].

## 2.3. Sluneční jádro

Jádru Slunce se nachází uprostřed Slunce asi do 20 % až 25 % jeho poloměru. Jádru je složeno z horkého stlačeného plynu v plazmatickém skupenství, jehož hustota se pohybuje okolo  $150 \text{ g/cm}^3$ . Je to nejteplejší část Slunce, teplota se zde pohybuje okolo  $15\,000\,000 \text{ K}$ .

Jádru produkuje energii pomocí termojaderné fúze, kde se slučují jádra vodíku na helium. Zbytek Slunce je pak zahříván touto energií, která je přenášena z jádra ven.

Energie produkovaná jádrem za jednotku času se mění se vzdáleností od středu Slunce. Ve středu slunečního jádra činí podle modelu  $276,5 \text{ W/m}^3$ . Přibližně v 19 % slunečního průměru, na hranici vrstvy v zářivé rovnováze, teplota jádra klesne na  $10\,000\,000 \text{ K}$  a produkce energie termojadernou fúzí na asi  $6,9 \text{ W/m}^3$ , to odpovídá 2,5 % maxima. Přesto je v této oblasti vyprodukováno asi 91 % celkové energie Slunce. Dál od středu Slunce termojaderná fúze ustává úplně [2].

Rychlost jaderné fúze závisí především na tlaku, který vytváří gravitace. Tento tlak, který způsobuje gravitace je kompenzován tlakem plazmatu při vysoké teplotě. Mírné zvýšení termojaderné fúze způsobí, že se jádru začne více zahřívát, začne se rozpínat, tím se sníží jeho hustota a sníží se tím i rychlost jaderné fúze. V opačném případě, když se jádru Slunce trochu ochladí, jeho objem se

zmenší a vzroste hustota a tím se zvýší i rychlost jaderné fúze na potřebnou velikost. Tímto způsobem se udržuje rychlost jaderné fúze v jádře v rovnováze.

#### **2.4. Vrstva v zářivé rovnováze**

Tato vrstva je druhá v pořadí směrem od středu Slunce. Vrstva v zářivé rovnováze se nachází mezi slunečním jádrem a konvektivní zónou, přibližně ve vzdálenosti 20 až 70 % průměru Slunce.

Vrstva v zářivé rovnováze je vnitřní vrstva v slunci, kde se energie šíří směrem ven od jádra převážně zářením, než prouděním. Energie v této zóně se šíří ve formě elektromagnetického záření jako fotony.

Hmota v této vrstvě je stlačena tak, že fotony mohou urazit jen malou vzdálenost, než jsou absorbovány nebo se srazí s jinou částicí. Z tohoto důvodu trvá průměrně 171 000 let, než energie z jádra projde vrstvou v zářivé rovnováze. Průchodem přes tuto vrstvu se teplota plazmatu sníží z 15 miliónů K, která se vyskytuje u jádra, na 1,5 miliónu K, která se vyskytuje na přechodu s konvektivní zónou.

Vrstva v zářivé rovnováze rotuje jako celek, nebyla v ní pozorována diferenciální rotace.

#### **2.5. Tachoklina**

Je to v pořadí třetí vrstva od středu Slunce. Tachoklina je přechodová vrstva mezi vnitřní vrstvou v zářivé rovnováze a venkovní vrstvou konvektivní zóny. Nedávné výsledky měření z helioseismologie ukazují, že se tachoklina nachází přibližně v 70 % průměru Slunce a její tloušťka je okolo 0,04 % slunečního poloměru.

V této oblasti dochází k velké změně rotace plazmatu. Vnější vrstva, konvektivní zóna, rotuje jako normální tekutina, u pólů rotuje pomalu a na rovníku rotuje rychle. Naproti tomu, vnitřní vrstva v zářivé rovnováze rotuje jako pevné těleso.

V této přechodové vrstvě Slunce dochází vlivem velikých změn v rychlosti plazmatu ke generování rozsáhlého magnetického pole Slunce. Termín tachoklina byl vytvořen jako analogie termínu termoklina, což jsou oceánské proudy vznikající na rozmezí dvou vrstev s rozdílnými teplotami.

#### **2.6. Konvektivní zóna**

Konvektivní zóna je vnější vrstva Slunce. Tato vrstva začíná přibližně na 70 % průměru Slunce a její tloušťka je okolo 200 000 km. Sluneční plazma v této vrstvě není již dostatečně stlačená, aby zde mohlo docházet k přenosu tepla zářením. Hlavním přenosem tepla se tak stává proudění neboli konvekce.

Toto proudění se v konvektivní zóně projevuje jako tepelné sloupce, které vynášejí horké plazma k povrchu Slunce. Jakmile se plazma na povrchu ochladí, padá dolů k základně konvektivní zóny, aby zde získala více tepla z horní části vrstvy v záživé rovnováze. Tyto tepelné sloupce jsou Rayleigh–Bénardovy buňky, proto mají tendenci mít tvar šestihraného hranolu.

Turbulentní proudění v této vnější části sluneční vrstvy vytváří magnetické pole celého povrchu Slunce.

## **2.7. Fotosféra**

Fotosféra je hraniční vrstvou slunečního tělesa a můžeme ji tedy z mnoha důvodů považovat za sluneční povrch. Hvězdy, s výjimkou neutronových hvězd, nemají pevný povrch, nicméně fotosféra bývá často používána k popisu viditelného slunečního povrchu. Je to vrstva, ve které plyn přestává být průhledný pro fotony. Tato vrstva navazuje na konvektivní zónu a proto děje, které probíhají v konvektivní vrstvě, se projeví na chování fotosféry.

Jde o vrstvu, v porovnání s rozměry celého Slunce velmi tenkou. Podle různých zdrojů má tloušťku 200 až 500 km. Přesto právě tato vrstva je předmětem slunečních pozorování od samého počátku pozorování Slunce, a proto je nejlépe prozkoumaná [3].

Průměrná teplota sluneční fotosféry se pohybuje mezi 4500 až 6000 K, průměrná efektivní teplota se pohybuje okolo 5800K. Fotosféra je tak nejchladnější oblastí Slunce. Hustota plazmatu v této vrstvě je okolo  $2 \cdot 10^{-4} \text{ kg/m}^3$ .

Sluneční fotosféra se skládá z konvenčních buněk, kterým se říká granule. Každá má v průměru asi 1000 km. Jsou to projevy konvektivních proudů horkého plazmatu, kdy teplejší plyn v centru této buňky stoupá vzhůru a chladnější zase padá po okrajích dolů.

Každá granule má životnost pouze okolo 8 minut, výsledkem je pak neustále měnící se tvar povrchu Slunce. Měření ukazují, že se pohyby v granulích probíhají především vertikálně s amplitudou rychlosti přibližně 1 km/s. Níže v nitru se granule seskupují do tzv. supergranulí, které mají v průměru až 30 000 km a délku života až 24 hodin. Pohyby v nich se odehrávají především horizontálně s amplitudou 500 m/s. O gigantických konvektivních buňkách, jež vznikají na samém dně konvektivní zóny, se mluví již od šedesátých let a jejich existence doposud nebyla spolehlivě prokázána. Takové buňky by měly mít rozměry kolem 200 000 km s horizontálními pohyby s amplitudou několika desítek m/s a životností více než týden. Jejich existence je ve fotosféře ještě více skryta, než existence supergranulí [3].

## **2.8. Sluneční skvrny**

Sluneční skvrny jsou temné chladné oblasti na povrchu Slunce, které mají teplotu okolo 4000 K a jsou způsobené intenzivními magnetickými poli prorážejícími přes povrch Slunce.

Skvrny se obvykle vyskytují ve skupinách, výjimečně osamoceně. Jejich vzhled a velikosti jsou rozmanité. Větší a vyvinutější skvrny se skládají z velmi tmavého jádra (umbry) a světlejšího polostínu (penumbry).

Životnost slunečních skvrn je také různá. Jsou známy skvrny, které zmizely po několika hodinách, stejně jako víme o rekordmanech, kteří přežili pět otoček Slunce, tedy téměř půl roku.

Skvrny se nacházejí především v tzv. královských pásech, které leží přibližně od 40° jižní heliografické šířky až po 40° severní heliografické šířky. Poloha, kde se objevují sluneční skvrny, také závisí na fázi aktivity Slunce. S nástupem nového cyklu se skvrny objevují ve větších vzdálenostech od rovníku. Končící cyklus je charakterizován mimo jiné koncentrací skupin skvrn velmi blízko slunečního rovníku [4].

## **2.9. Chromosféra**

Chromosféra je nejnižší část sluneční atmosféry, která se nachází nad fotosférou a její tloušťka je přibližně 2000 km. Její název pochází z řeckého slova  $\chi\rho\omega\sigma$  -barva a tento název vychází ze skutečnosti, že tato vrstva má červenou barvu. Tato červená barva je způsobená tím, že vizuální spektrum chromosféry dominuje v temně červené oblasti H-alfa spektrálních čar vodíku.

Tato vrstva nemůže být normálně pozorována bez speciálního vybavení, protože není běžně vidět kvůli mnohonásobně většímu jasů fotosféry. Může být pozorována během zatmění Slunce nebo pomocí zvláštních úzkopásmových optických filtrů pracujících v oblasti H-alfa spektrálních čar. Tato technika pozorování zobrazí chromosféru velmi jasně a mnoho observatoří ji používá k pozorování chromosféry.

## **2.10. Koróna**

Koróna je horní část sluneční atmosféry, která volně přechází do meziplanetárního prostoru. Měřeními bylo zjištěno, že teplota v této oblasti dosahuje hodnot okolo 1 500 000 K a hustota částic v koruně se pohybuje okolo  $10^{11}$  částic/m<sup>3</sup>. Koróna může být pozorována pouze během slunečního zatmění, kdy je hlavní záření ze slunečního povrchu zastíněno průchodem Měsíce nebo pomocí speciálních přístrojů.

Díky vysokým teplotám koróna vyzařuje záření v oblasti od UV spektra až po rentgenovou oblast. Korónu můžeme dělit na několik částí, podle toho v jakém spektru záření vyzařuje [5].

- **E koróna (emisní)** je to vlastní záření korony. Jsou v ní obsaženy emisní čáry způsobené přechody ve vysoce ionizovaných kovech. Tyto čáry jsou možné jen za vysokých teplot milionů kelvinů.
- **K koróna (kontinuum)** je způsobena rozptylem slunečního světla na volných elektronech.
- **F koróna (Fraunhoferova)** je způsobena rozptylem slunečního světla na prachových částicích padajících z meziplanetárního prostoru na Slunce, charakteristické jsou absorpční čáry.

Sluneční koróna nemá vnější hranici a plynule přechází do vesmíru a vytváří tak sluneční vítr, proud nabitých částic, který se rozpíná zhruba do vzdálenosti 100 AU.

Tento proud nabitých částic ze Slunce zaplavuje celou sluneční soustavu a skládá se zejména z protonů, elektronů a alfa částic (jádra hélia). Typická rychlost těchto částic při průletu kolem Země je okolo 500 km/s, teplota 3 eV (30 000 K) a koncentrace několik protonů na m<sup>3</sup>. Částice vylétávající v polárním směru Slunce mají vyšší rychlost, přibližně 750 km/s a nazýváme je rychlý sluneční vítr [5].

Sluneční vítr proudí ze Slunce všemi směry a tvoří v mezihvězdném prostředí „bublinu“ z těchto částic, tzv. Heliosféru, oblast kam zasahuje vliv Slunce. Heliosféra končí hraniční vrstvou, jejíž vnější část se nazývá heliopauza. Je to oblast, kde se vyrovná tlak částic slunečního větru s tlakem částic z okolních hvězd a tlaku galaktického větru. Je to tedy hraniční oblast mezi heliosférou a mezihvězdným prostorem. Jedná se o oblast, kde končí sluneční soustava.

### ***2.11. Problém ohřevu sluneční korony***

Průměrná efektivní teplota povrchu Slunce pohybuje okolo 5800 K. Je jasné, že teplota směrem od Slunce by se měla snižovat, přesto bylo měřeními zjištěno, že teplota ve sluneční koróně dosahuje hodnot okolo 1 500 000 K. Takto vysoké teploty v koróně ukazují, že tato oblast musí být permanentně ohřívána jinými mechanismy, než jen přímým ohřevem od fotosféry, jinak by plazma v koróně vychladla v průběhu několika hodin.

Příčina vysoké teploty v této vrstvě není do dnešních dnů zcela jasně vysvětlena. Teorií o tom, co by takto vysokou teplotu mohlo způsobovat, je celá řada. Do dnešní doby ale přežily jen tři. Protože žádný mechanismus neposkytuje samostatně dostatečný výkon k trvalému ohřevu koróny, musí se na něm podílet všechny tři mechanismy najednou [5].

### **2.11.1. Sluneční mikroerupce**

Sluneční erupce jsou výrony plazmatu způsobené rekonekcí magnetických siločar. Při průměrné erupci se uvolní energie  $10^{25}$  J, která zahřeje okolní plazma, vymrští ho a způsobí vzplanutí až v RTG oboru. Takových velkých erupcí je ale nesmírně málo na to, aby viditelně zahřály koronu. Podle teorie by mělo docházet i k tzv. mikroerupcím, jejichž energetická bilance je sice o dva řády nižší, ale jejich počet by měl být enormní a tak by měly mikroerupce přispívat k ohřevu korony významnou měrou. V místě, kde siločary míří proti sobě (tzv. X bod), dojde k přepojení a uvolnění energie [5].

### **2.11.2. Magnetoakustické vlny**

Zvuk se plazmatem šíří mnohem komplikovaněji než v běžném prostředí. Energie vibrujících iontů se přelévá mezi chaotickou složkou (tlakovou energií), uspořádanou složkou (kinetickou energií kmitajících iontů) a energií magnetického pole. Zvuk se šíří anizotropně ve třech vlnoplochách (rychlé, pomalé a Alfvénově vlně). Asi nejjednodušší je Alfvénova vlna, kdy dochází k příčnému rozvlnění magnetických siločar a k přenosu energie zvukové vlny podél siločar Alfvénovou rychlostí. Právě Alfvénovy vlny mohou vynášet energii v podobě zvukových vln z povrchu podél siločar až vysoko do korony. Podle prvních odhadů by ani tento mechanismus neměl být sám dostatečný k ohřevu korony [5].

### **2.11.3. Ohmický ohřev**

Pohybující se nabitě částice znamenají různé elektrické proudy tekoucí plazmatem. Vzhledem k tomu, že plazma má konečnou vodivost, vždy bude docházet k disipaci energie Jouleovým ohřevem. Odhady opět ukazují, že samotný Jouleův ohřev je v řídké koruně málo účinný a nemůže být samostatně zodpovědný za ohřev korony [5].

## 3. Programy pro výpočty a simulace v plazmové astrofyzice

### 3.2. Programy používané pro výpočty

V dnešní době je mnoho programů, které je možno používat pro simulace. Liší se především v možnostech, jaké problémy jsou schopny řešit a v jakém programovacím jazyku jsou napsány. Softwaru, který se používá v astrofyzice pro modelování, simulace a analýzu, jsou věnované internetové stránky **Astro-Code Wiki** [6]. V tabulce 2 je seřazen přehled nejvíce používaných programů.

### 3.3. FLASH

Pro výpočty problémových úloh v této diplomové práci byl zvolen program FLASH. Byl vybrán pro jeho široké možnosti použití a řadu již vytvořených příkladů. Jeho částečnou nevýhodou může být složitější instalace.

#### 3.3.1. Základní vlastnosti FLASHe

FLASH je software určený pro numerické výpočty. Je modulárně přizpůsobitelný kód schopný zvládnout počítat obecné problémy stlačitelného proudění v astrofyzikálních prostředích.

- Jedná se o modulární program. FLASH byl navržen tak, aby uživatelům dovolil konfigurovat počáteční a okrajové podmínky při výpočtech, měnit algoritmy, a přidat nové fyzikální efekty s minimálním úsilím.
- Je adaptivní: FLASH používá knihovnu PARAMESH k řízení blokově strukturované adaptivní mřížky, používá prvky pro větší rozlišení pouze tehdy, pokud je to potřeba.
- Může počítat paralelně: FLASH používá Message-Passing Interface (MPI) knihovnu pro dosažení přenositelnosti a škálovatelnosti na různých druzích paralelních počítačů.

Aktuální verze kódu FLASH je nyní v jeho 4. verzi. Je to veřejně dostupný, vysoce výkonný kód, který je modulární a rozšiřitelný software. FLASH se skládá z vzájemně spolupracujících modulů, které mohou být kombinovány ke generování různých aplikací. Architektura FLASHe dovoluje mít libovolně mnoho alternativních implementací těchto komponent, které mohou existovat nebo se zaměřovat jedna s druhou. Toto je jednoduchý a elegantní mechanismus, který umožňuje přizpůsobit funkce kódu bez nutnosti zasahovat do zdrojových kódů jádra programu.

Aktuální verze programu FLASH implementuje fyzikální algoritmy pro řešení fyzikálních problémů zejména hydrodynamiky a magnetohydrodynamiky.

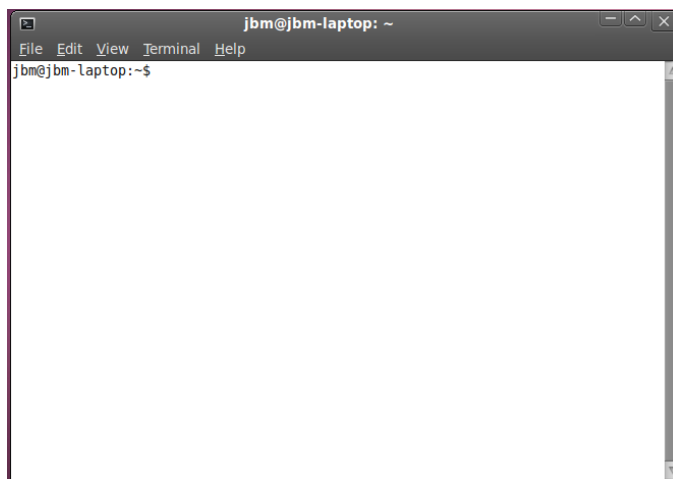
**Tab. 2:** Přehled programů používaných pro výpočty v astrofyzice.

Jméno	Aktuální verze	První vydání	Napsáno v	Typ	MHD	Záření	Vlastní gravitace	SR	GR	Rozměry	Souřadnice	Paralelní výpočty
Athena	3.0 (2007)	2000	C	FV	x	-	-	-	-	1D, 2D, 3D	kart., cyl.	MPI
Enzo	1.0.1 (2007)	2004	C++, F77	FV & FD	-	x	x	-	-	1D, 2D, 3D	kart.	MPI
FLASH	2.3 (2003), 3beta (2007)	1998	F90, C	FV	x (2D, bez sfér.)	-	x	x	-	1D, 2D, 3D	kart., cyl., sfér. (bez 3D, MHD)	MPI
HARM		2003	C	FV	x	-	-	x	x	1D, 2D	kart., cyl., sfér.	-
Heracles				FV	-	x	-	-	-	1D, 2D, 3D	kart., cyl., sfér.	-
Nirvana	3.2.1 (2007)	1997	C	FV	x	-	x	-	-	2D, 3D	kart.	MPI (s omezeními)
Pencil		2001	F90	FD	x	FLD (exp.)	x	-	-	1D, 2D, 3D	kart.	MPI
Pluto	3.1.0 (2010)	2006	C	FV/FD	x	x	-	x	-	1D, 2D, 3D	kart., cyl., sfér.	MPI
Ramses	3.06 (2010)		F90	FV	x	-	x	-	-	1D, 2D, 3D	kart., cyl., sfér.	MPI
VAC	4.51 (2005)	1996	F90/Perl	FV	x	x	x	-	-	1D, 2D, 3D	kart., cyl., sfér.	MPI,HPF nebo OpenMP
ZEUS	MP/2 2.1.2 (2007)	začátek 80.let	F90	FD	x	FLD	x	-	-	1D, 2D, 3D	kart., cyl., sfér.	MPI (verze - MP)



### 3.4. Instalace programu FLASH

Protože program FLASH pracuje pouze pod OS Linux, byl vytvořen jednoduchý a přehledný manuál instalace tohoto numerického kódu. Instalace v OS Linux se většinou provádí z příkazového řádku, viz obr. 2, zadáváním příkazů a ne v grafických prostředích, jak je tomu např. v OS Windows. Tento stručný návod vznikl z toho důvodu, aby člověk, který se nesetkal s instalací programů do Linuxu, byl schopen pomocí tohoto manuálu nainstalovat program FLASH a vizualizační software sám.



Obr. 2: Okno příkazového řádku v OS Linux.

#### 3.4.1. Požadavky na výběr OS

Protože program FLASH pracuje jen pod OS Linux, byl výběr zúžen na tento operační systém. Program lze tedy spustit v zásadě na jakékoliv Linuxové distribuci, ale nakonec jako nejvhodnější distribuce Linuxu byl vybrán UBUNTU. Jako hlavní důvody tohoto výběru jsou:

- OS, který je přívětivý k začínajícímu uživateli Linuxu, obsahuje grafické rozhraní.
- Většina instalačních balíčků je dostupná pro tuto distribuci Linuxu. Pro tuto distribuci jsou také vizualizační programy zkompileované již jako instalační balíčky.

Jako veze této linuxové distribuce UBUNTU byla vybrána **10.04 Lucid Lynx**, která je označována jako LTS, což znamená, že má prodlouženou dobu podpory. Všechny postupy a instalační skripty popsané v této diplomové práci pak byly ověřeny na tomto typu OS.

### 3.4.2. Přepínání mezi uživateli

V průběhu následujících kroků instalace se budeme muset přepínat do rolí různých uživatelů. Z běžného uživatele se do role uživatele **root** přepneme zadáním příkazu:

```
sudo su
```

pro dokončení tohoto příkazu budeme vyzváni pro zadání hesla k účtu tohoto uživatele. Toto heslo jsme zadali při instalaci OS Ubuntu.

Pro přepnutí zpět do role normálního uživatele můžeme zadat příkaz

```
exit
```

Další možností je uzavřít okno terminálu a znovu jej otevřít.

### 3.4.3. Příprava před instalací

Aby byl postup instalace programu FLASH nebo vizualizačních programů stejný, je nutné si vytvořit na disku složku, kde budou uloženy všechny instalační balíčky programů.

```
mkdir /tmp/install_flash/
```

Dále k instalaci budeme potřebovat mít stažené následující soubory se zdrojovými kódy programů. Verze níže uvedených souborů se mohou lišit podle aktuální stažené verze.

- **l\_fcompxe\_intel64\_2011.1.107.tgz** – Intel Fortran Compiler
- **mpich2-1.1.1p1.tar.gz** – Prostředí MPI pro paralelní výpočty
- **hdf5-1.6.10.tar.bz2** – HDF5 Hierarchical data format, potřebné k výpočtům
- **FLASH3.3.tar** – zdrojový kód FLASH, verze 3.3.

Dále je potřeba vytvořit adresář **ASTRO\_SIM**, kam budeme instalovat samotný FLASH3.3 a další vizualizační programy. Tento adresář vytvoříme v domovském adresáři uživatele:

```
mkdir $HOME/ASTRO_SIM/
```

### 3.4.4. Instalace kompilátoru jazyka G++

Intel fortran compiler potřebuje pro svou činnost mít nainstalované knihovny kompilátoru jazyka G++,C,C++. Pro nainstalování tohoto kompilátoru je třeba zadat následující příkaz:

```
sudo apt-get install g++
```

po dobehnutí instalace ověříme ještě funkčnost kompilátoru g++.

### 3.4.5. Test správné instalace G++

Pro kontrolu, jestli je kompilátor C++ a G++ správně nainstalován zadáme následující příkazy:

```
c++
```

```
Dostaneme odpověď: c++: no input files
```

```
g++
```

```
Dostaneme odpověď: g++: no input files
```

### 3.4.6. Instalace JRE - Java Runtime Environment

V Linuxu UBUNTU není standardně nainstalováno prostředí JRE pro spouštění aplikaci v Javě, je nutné ho před instalací Intel Fortran Compileru doinstalovat. Instalace se provede následujícím příkazem:

```
sudo apt-get install default-jre
```

#### 3.4.6.1. Test správné instalace JRE

Test, jestli je správně nainstalováno prostředí JRE pro běh aplikací Java provedeme následujícím příkazem:

```
java -version
```

```
Dostaneme odpověď:
```

```
java version "1.6.0_24"
```

```
OpenJDK Runtime Environment (IcedTea6 1.11.5) (6b24-1.11.5-0ubuntu1~10.04.2)
```

```
OpenJDK 64-Bit Server VM (build 20.0-b12, mixed mode)
```

### 3.4.7. Intel® Fortran Compiler

**Fortran** (je zkratka slov **FOR**mula a **TRAN**slator) je imperativní programovací jazyk, který byl navržen firmou IBM v 50. letech 20. století a používá s především pro vědecké výpočty a numerické aplikace. Fortran se brzy stal jedničkou mezi programovacími jazyky a více než půl století se využíval například pro výpočty předpovědi počasí, analýzu pevných částic a další fyzikální a chemické výpočty [7].

### 3.4.7.1. Instalace Intel® Fortran Compileru

Zdrojové kódy pro nekomerční použití je možné stáhnout na stránkách firmy Intel® Software [8]. Instalace se musí provádět jako uživatel root. Po přepnutí uživatele rozbalíme zkomprimovaný soubor s instalací Fortran Compileru následujícím příkazem:

```
tar -xvzf l_fcompxe_intel64_2011.1.107.tgz -C /tmp/instar_flash
```

Po dekompresi se přepneme do instalačního adresáře pomocí příkazu:

```
cd /tmp/instal_flash/l_fcompxe_intel64_2011.1.107
```

V adresáři spustíme instalaci pomocí příkazu:

```
./install.sh
```

Během instalace budeme vyzváni, abychom zvolili adresář, kam se má nainstalovat Fortran Compiler. Tento adresář nebudeme měnit, necháme ten, který nám instalátor sám nabízí, tj.: **/opt/intel/composerxe-2011.1.107**

V dalším kroku instalace budeme, po přečtení licenčních podmínek, vyzváni k zadání licenčního čísla. Toto licenční číslo je možné získat pro nekomerční použití tohoto softwaru, na stránkách Intel® Software [9]. Po zadání budeme opět sledovat instrukce k úspěšnému dokončení celé operace.

Po dokončení instalace je potřeba nastavit proměnné prostředí pro správný běh tohoto programu. Tyto proměnné jsou uloženy v souboru bashrc.

Přepneme se do role „běžného uživatele“ a pomocí textového editoru Gedit upravíme soubor bashrc, napíšeme následující příkaz:

```
gedit $HOME/.bashrc
```

Otevře se okno s editorem a v něm obsah souboru **bashrc**. Na konec souboru vložíme následující řádky, které se liší podle verze OS Ubuntu.

Pro 64-bit verzi Ubuntu je potřeba nastavit tyto proměnné:

```
#Intel Fortran Composer  
source /opt/intel/bin/compilervars.sh intel64
```

Pro 32-bit verzi Ubuntu je potřeba nastavit tyto proměnné:

```
#Intel Fortran Composer  
source /opt/intel/bin/compilervars.sh ia32
```

Uložíme soubor, a aby se projevíly změny v nastavení proměnných prostředí, je třeba okno terminálu zavřít a znovu otevřít.

### 3.4.7.2. Test správné instalace Intel® Fortran Compileru

Pro vyzkoušení správné instalace Fortran compileru napíšeme příkaz:

```
ifort
```

Pokud je Fortran úspěšně nainstalován a máme správně nastavené lokální proměnné, měli bychom obdržet následující řádku:

```
ifort: command line error: no files specified; for help type "ifort -help"
```

### 3.4.8. MPICH2

Je specifikace (protokol) pro podporu paralelního řešení výpočetních problémů v počítačových clusterech. MPICH2 je velmi výkonná a široce přenositelná implementace standardu Message Passing Interface (dále jen MPI) – oba standardy MPI-1 a MPI-2. Konkrétně se jedná o rozhraní pro vývoj aplikací, které je založené na zasílání zpráv mezi jednotlivými uzly jako specifikace pro uživatelské rozhraní a message-passing knihovnamy pro paralelní výpočty. Je navrženo pro širokou skupinu výrobců paralelních počítačů, programátorů a vývojářů aplikací jako standard pro paralelní výpočty [10], [11].

#### 3.4.8.1. Instalace MPICH2

Instalace MPICH2 je podrobně popsána v instalačním manuálu [12]. V instalaci budeme postupovat víceméně podle tohoto návodu.

Rozbalíme komprimovaný soubor s instalací MPICH2 do instalačního adresáře pomocí tohoto příkazu:

```
tar -xzf mpich2-1.1.1pl.tar.gz -C /tmp/install_flash
```

Nyní se opět přihlásíme jako uživatel root. V adresáři **opt** vytvoříme podadresář *mpich2* pomocí příkazu:

```
mkdir /opt/mpich2
```

Následující příkaz, který vytvoří soubor Makefile, napíšeme jako jeden řádek:

```
/tmp/flashinstal/mpich2-1.1.1pl/configure prefix=/opt/mpich2/ --enable-f90 --enable-threads=multiple FC=/opt/intel/bin/ifort F90=/opt/intel/bin/ifort
```

Po vytvoření souboru Makefile napíšeme následující příkaz, který přeloží zdrojové kódy do binárních souborů:

```
make
```

Překlad bude trvat delší dobu a po úspěšném ukončení program MPICH2 nainstaluje příkazem:

```
make install
```

Po instalaci je potřeba doplnit do souboru **bashrc** proměnné pro správný běh tohoto programu, opět jako normální uživatel. Na konec souboru přidáme následující řádek:

```
#MPICH2
export PATH=/opt/mpich2/bin:$PATH;
```

Uložíme soubor, a aby se projevily změny v nastavení proměnných prostředí, je třeba okno terminálu zavřít a znovu otevřít.

### 3.4.8.2. MPD daemon

Podle manuálu je potřeba pro správný běh nastavit spuštění daemonu *mpd*, který je součástí programu MPICH2. Daemon je program, který běží v pozadí a čeká na události, které nastanou, reaguje na ně a poskytuje služby. Přepneme se opět jako uživatel root a napíšeme příkazy:

```
cd $HOME
touch .mpd.conf
chmod 600 .mpd.conf
echo „secretword=mr45-j9z“ > .mpd.conf
```

Tyto příkazy vytvoří konfigurační soubor pro mpd. Pro spuštění procesu na pozadí se zadá následující příkaz:

```
mpd &
```

Při spuštění procesu se do okna terminálu vypíše podobná informace, jako tato:

```
[1] 15671
```

### 3.4.8.3. Test správné instalace MPICH2 a MPD daemonu

Pro kontrolu správnosti instalace můžeme vyzkoušet následující příkazy:

```
which mpd
```

Dostaneme odpověď: **/opt/mpich2/bin/mpd**

```
which mpicc
```

Dostaneme odpověď: **/opt/mpich2/bin/mpicc**

```
which mpiexec
```

Dostaneme odpověď: **/opt/mpich2/bin/mpiexec**

```
which mpirun
```

Dostaneme odpověď: **/opt/mpich2/bin/mpirun**

```
mpdtrace
```

Dostaneme odpověď: **ubuntu**

```
mpdallexit
```

Tento příkaz uzavře vše. Pokud jsme obdrželi správné odpovědi, je MPICH2 nainstalován v pořádku.

### 3.4.9. HDF5

HDF (Hierarchical Data Format) je souhrnný název formátu souborů a knihoven určených pro ukládání a organizování velkého množství numerických dat. Původně byl vyvinutý v Národním centru pro superpočítačové aplikace, v současnosti je podporován neziskovou skupinou HDF, jejímž posláním je zajistit další vývoj HDF5 technologií a pokračující dostupnosti dat uložená v HDF. HDF technologie řeší problémy, jak organizovat, ukládat, zpřístupňovat, analyzovat, sdílet a chránit data vzhledem k jejich obrovskému nárůstu velikosti a složitosti [13, 14].

#### 3.4.9.1. Instalace HDF5

Instalace je podobná instalaci MPICH2. Nejdříve rozbalíme archiv **hdf5-1.6.10.tar.bz2** do adresáře */tmp/install\_flash*. Pro rozbalení použijeme následující příkaz:

```
tar -xjvf hdf5-1.6.10.tar.bz2 -C /tmp/install_flash
```

Nyní se musíme přihlásit jako uživatel *root*. V adresáři *opt* vytvoříme podadresář *hdf5* pomocí příkazu:

```
mkdir /opt/hdf5
```

V dalším kroku musíme nakonfigurovat instalaci HDF5 podle prostředí našeho počítače. Následující příkaz napíšeme jako jeden řádek:

```
/tmp/flashinstall/hdf5-1.6.10/configure prefix=/opt/hdf5/  
FC=/opt/intel/bin/ifort F90=/opt/intel/bin/ifort FX=/opt/intel/bin/ifort
```

Když skončí konfigurace úspěšně, napíšeme následující příkaz, který sestaví zdrojové kódy do binárních spustitelných souborů:

```
make
```

Překlad bude trvat delší dobu. Jestliže sestavení programu neskončilo žádnou chybou, můžeme napsat následující příkaz, který nainstaluje HDF5.

```
make install
```

Tímto příkazem nainstalujeme HDF5 a po jeho úspěšné instalaci je potřeba doplnit do souboru **bashrc** proměnné pro správný běh tohoto programu, opět jako normální uživatel.

Následující řádka by měla být doplněna pod řádkou source /opt/intel/bin/compilervars.sh intel64:

```
export LD_LIBRARY_PATH=/opt/intel/lib:$LD_LIBRARY_PATH
```

Nakonec souboru přidáme následující řádky:

```
#HDF5  
  
export LD_LIBRARY_PATH=/opt/hdf5/lib:$LD_LIBRARY_PATH  
  
export LD_RUN_PATH=/opt/hdf5/lib:$LD_RUN_PATH  
  
export INCLUDE=/opt/hdf5/include:
```

Uložíme soubor, a aby se projevily změny v nastavení proměnných prostředí, je třeba okno terminálu zavřít a znovu otevřít.

### 3.4.9.2. Test správné instalace HDF5

Pro test správné instalace HDF5 napíšeme následující příkaz:

```
make check-install
```

Dostaneme odpověď: **ok**

Pro další kontrolu správné instalace HDF5 můžeme ukončit okno terminálu a znovu spustit, pokud okno terminálu nevypiše chybové hlášení, bashrc jsme upravili dobře.



### 3.4.10. Instalace programu FLASH

Při instalaci programu FLASH budeme postupovat podle manuálu, který je ke stažení na stránkách FLASH center [15][16].

Pomocí příkazu `tar` rozbalíme soubor **FLASH3.3.tar** do adresáře `home/ASTRO_SIM`:

```
tar -xvf FLASH3.3.tar -C $HOME/ASTRO_SIM/
```

Tím je celá instalace v podstatě hotová. Nyní musíme ještě program FLASH nakonfigurovat.

#### 3.4.10.1. Soubor Makefile.h

Soubor `Makefile.h` se využívá při sestavení spustitelného kódu ze zdrojových kódů programu FLASH. Je to předpis pro sestavení, který obsahuje názvy cest ke knihovnám potřebným pro sestavení kódu a pak i pro jeho běh.

Při spuštění skriptu **setup**, který nakonfiguruje FLASH pro výpočty, prohledává tento skript adresář **sites** a hledá adresář se souborem `Makefile`, který se jmenuje podle hostname počítače.

Pro vytvoření souboru `Makefile` specifického pro počítač, kde pracujeme, musíme postupovat podle následujících instrukcí. Nejdříve musíme zjistit hostname počítače, na kterém pracujeme. Hostname je unikátní jméno identifikující počítač v síti. Toto jméno zjistíme následujícím příkazem:

```
echo $HOSTNAME
```

Nyní musíme vytvořit adresář **sites/<hostname>**, kde **<hostname>** je jméno našeho počítače, které jsme zjistili předcházejícím příkazem.

```
mkdir sites/$HOSTNAME
```

Nyní nás čeká úprava souboru `Makefile.h`, který budeme při výpočtech používat. Přepneme se tedy do adresáře **/sites**:

```
cd /sites
```

Zde se nachází několik adresářů, z nichž je pro nás nejdůležitější následující:

```
flash.uchicago.edu
```

Z tohoto adresáře si zkopírujeme soubor *Makefile.h* do vlastního adresáře, který jsme vytvořili podle hostname našeho počítače. Soubor *Makefile.h* zkopírujeme nyní pomocí tohoto příkazu:

```
cp sites/flash.uchicago.edu/Makefile.h sites/$HOSTNAME
```

Nyní je potřeba tento soubor upravit a doplnit zde cesty k potřebným knihovnám pomocí textového editoru, to je asi nejtěžší bod instalace programu FLASH.

Máme dvě možnosti, jak postupovat při doplňování hodnot do tohoto souboru.

První možnost je ručně vyhledat názvy adresářů s potřebnými knihovnami. Pro snadnější identifikaci, kde se nacházejí potřebné knihovny, lze využít skript **FLASH3.3/bin/suggestMakefile.sh**, který prohledá celý systém a zobrazí názvy míst a adresářů nalezených knihoven.

Další možnost, jak doplnit správné hodnoty do souboru Makefile.h je, doplnit hodnoty podle výpisu souboru, který je v příloze 2 této práce. Pokud bylo postupováno stejně podle předchozích bodů instalace popsané v této práci, měly by být hodnoty v tomto souboru identické.

### 3.4.10.2. Test správné instalace FLASH

Jako test správné instalace zkusíme nakonfigurovat a sestavit vzorový příklad, stejný, jako používají v manuálu. Napíšeme příkazy:

```
./setup Sedov -auto  
  
cd object/  
  
make
```

Pokud se konfigurace ukončí a vidíme jako poslední řádek slovo "SUCCESS", pravděpodobně jsme FLASH3.3 správně nainstalovali

## 3.5. Vizualizační software

Pro zobrazení výsledků provedených výpočtů je možné použít mnoho programů, jak už bezplatné programy nebo i komerční placený software. Na internetových stránkách astro-sim.org [17] je přehledně zpracován přehled volného i komerčního software používaného pro grafické zobrazení dat z vědeckých výpočtů.

Jako hlavní program pro vizualizaci jsem zvolil bezplatný program VisIt. Pro porovnání zobrazení výsledků v této práci budou použity výstupy z komerčního vizualizačního programu IDL od firmy Exelis.

Na jeden počítač je možné nainstalovat více vizualizačních programů, je možné mít pro vizualizaci nainstalované oba programy. VisIt je vhodný pro rychlé zobrazení výsledků a docela dobře se v něm vytvářejí animace z výpočtů. Je též jednodušší na instalaci, než IDL.

### 3.5.1. VisIt

VisIt je nástroj, pomocí kterého můžeme interaktivně paralelně zobrazovat a analyzovat vědecká data na Unix a PC platformách. Pomocí VisIt je možné rychle data zobrazit, udělat z nich animaci, manipulovat s nimi a výsledek snadno uložit. VisIt obsahuje mnoho nástrojů, kterými můžeme zobrazit data různými způsoby. Může být použit pro zobrazení skalárních a vektorových polí definovaných ve dvou, nebo třech dimenzích (2D nebo 3D), strukturovaná a nestrukturovaná pole. VisIt je navržen tak, aby zvládl zacházet se sadou dat v řádech terabajtů, ale zvládne i zacházet se sadou dat v rozměru kilobajtů.

#### 3.5.1.1. Instalace VisIt

Instalační soubor je možné stáhnout na internetových stránkách projektu VisIt [18]. Na těchto stránkách je také možné stáhnout uživatelskou příručku a další dokumentaci k tomuto vizualizačnímu programu. Jako nejvyšší možná verze, kterou je možno stáhnout a bude pracovat na OS Ubuntu 10.04LTS je verze VisIt 2.3.2. [19].

Instalace programu VisIt je velmi jednoduchá. Jedná se vlastně jen o rozbalení instalačního balíku a zadání příkazu pro jeho spuštění. Zadáním následujícího příkazu rozbálíme soubory programu VisIt na správné místo, tj. do adresáře ASTRO\_SIM. Přepneme se do adresáře, kde máme stažený soubor s instalací a zadáme tento příkaz a pak se přepneme do adresáře programu:

```
tar -xvf visit2_3_2.linux-x86_64-ubuntu8.tar.gz -C $HOME/ASTRO_SIM/  
cd $HOME/ASTRO_SIM/visit2_3_2.linux-x86_64/bin/
```

Pro to, aby VisIt byl schopen číst soubory s daty speciálně vytvořené v programu FLASH2 nebo FLASH3, musíme VisIt spustit z příkazové řádky příkazem s tímto parametrem:

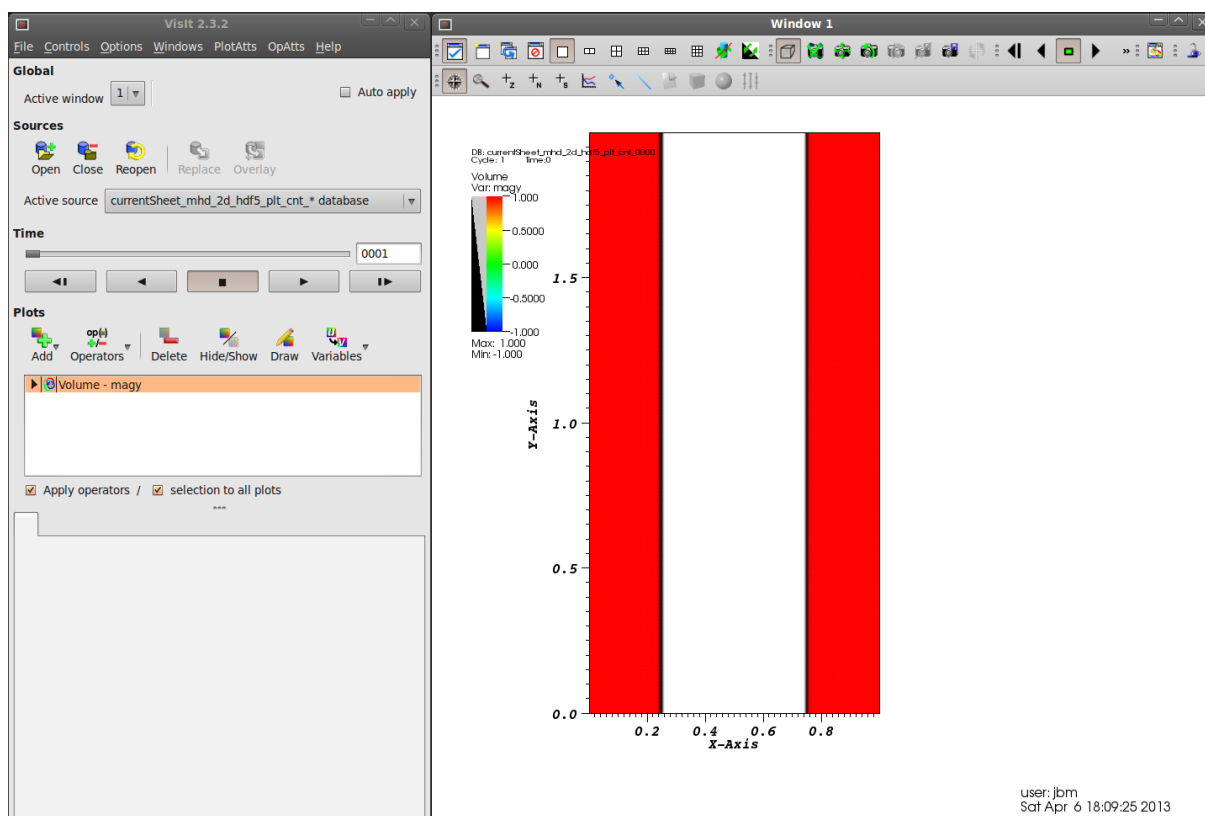
```
./visit -assume_format FLASH
```

Po zadání tohoto příkazu se otevře okno s programem VisIt, viz obrázek 3.

### 3.5.2. IDL

Dalším programem s grafickým rozhraním, pomocí kterého jdou zobrazovat výsledky, je program IDL firmy Exelis [20]. Tento software je komerční, proto není zadarmo. Program je primárně určen pro zobrazování dat z jazyku IDL (interactive data language), ale zadáním parametrů při jeho spuštění je možné zobrazovat v něm i data z programu FLASH.

Pro potřeby instalace je potřeba soubor se zdrojovými kódy instalace, zabalené do souboru **idl\_linux.tar.gz**



**Obr. 3:** Okno programu VisIt.

### 3.5.2.1. Instalace IDL

Nejdříve musíme vytvořit adresářovou strukturu pro instalaci. Instalace programu IDL se provádí pod uživatelem root. V terminálu vytvoříme pomocí následujícího příkazu v adresáři **/usr/local** podadresář **itt**:

```
mkdir /usr/local/itt
```

Dále rozbalíme komprimovaný soubor se zdrojovými kódy do vytvořeného adresáře. V adresáři, kde máme připravený soubor s instalací, napíšeme:

```
tar -xvzf idl_linux.tar.gz -C /usr/local/itt
```

Po rozbalení by měl tento adresář **/usr/local/itt** obsahovat následující čtyři adresáře: **idl**, **idl706**, **license**, **install**.

Přepneme se do tohoto adresáře a v příkazovém řádku napíšeme následující příkaz, který program IDL nainstaluje:

```
cd /usr/local/itt
./install
```

Po skončení instalace je potřeba doplnit do souboru **bashrc** proměnné, opět jako normální uživatel. Na konec souboru doplníme tyto řádky:

```
#IDL
IDL_DIR="/usr/local/itt/idl"; export IDL_DIR
XFLASH3_DIR="/home/a/FLASH3.2/tools/fidlr3.0"; export XFLASH3_DIR
IDL_PATH="${XFLASH3_DIR}:${IDL_DIR}:${IDL_DIR}/lib"; export IDL_PATH
IDL_PATH="${XFLASH3_DIR}:${IDL_PATH}"; export IDL_PATH
```

Uložíme soubor, a aby se projevily změny v nastavení proměnných prostředí, je třeba okno terminálu zavřít a znovu otevřít.

Při instalaci na 64 bitový OS je dále potřeba vytvořit soubor **arch** a nakopírovat jej na správné místo podle následujících příkazů. Pomocí tohoto souboru zjistí vizualizační software IDL, že běží na 64-bitovém prostředí. V domácím adresáři napíšeme postupně následující příkazy, jako normální uživatel. Budeme vyzváni k zadání hesla pro uživatele root.

```
touch arch
echo "uname -m" > arch
sudo chmod 755 arch
```

Pomocí následujícího příkazu přesune tento soubor do adresáře /bin/arch .

```
sudo mv arch /bin/
```

Tím je celá instalace IDL hotová.

### 3.5.2.2. Spuštění IDL

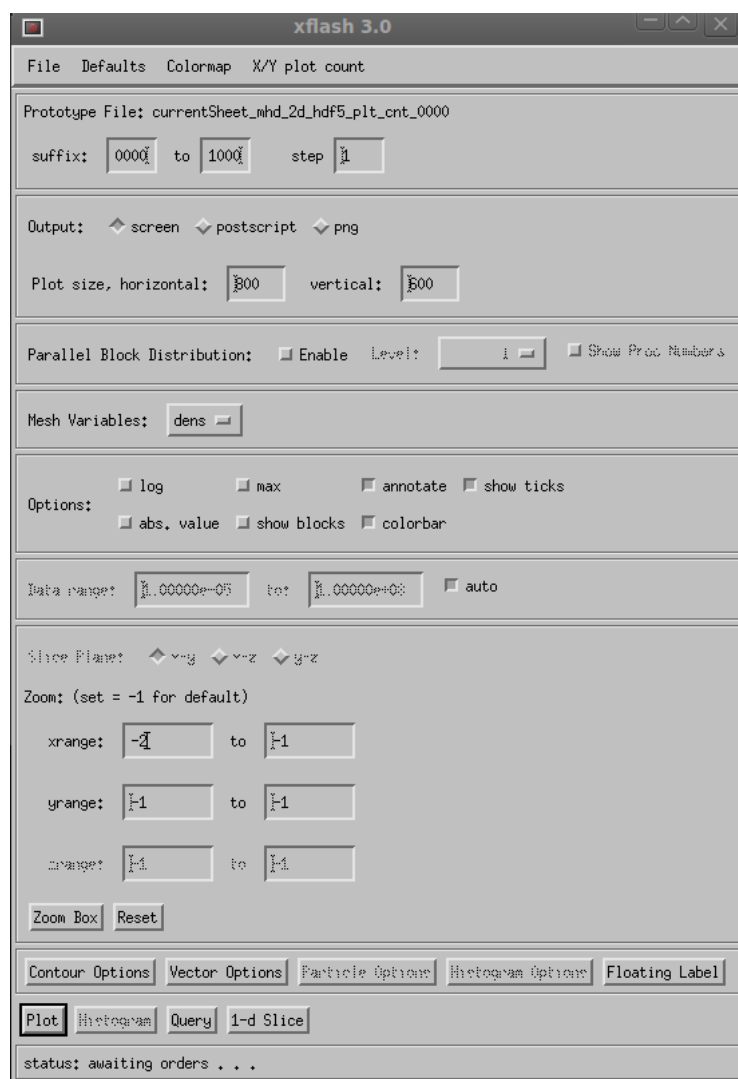
Pro spuštění IDL se nejdříve musíme přepnout do adresáře **FLASH3.3/tools/fidlr3.0** a napsat příkaz s tímto parametrem:

```
cd $HOME/ASTRO_SIM/FLASH3.3/tools/fidlr3.0/
idl start_linux.pro
```

Pro spuštění hlavního okna programu napíšeme příkaz:

```
IDL> xflash3
```

Otevře se hlavní grafické okno programu IDL, jak je zobrazené na obrázku 4.



**Obr. 4:** Spuštěné okno programu IDL

Abychom se dostali na normální příkazový řádek po uzavření programu IDL, musíme napsat tento příkaz:

```
IDL> exit
```

### 3.6. *Instalační skripty*

Pro usnadnění instalace pro méně zkušené uživatele Linuxu byly vytvořeny skripty, které provedou jednotlivé kroky pro instalaci prostředí FLASH a všech programů potřebných jeho spuštění. Použitím těchto skriptů odpadne zdlouhavé zadávání dlouhých příkazů při instalaci, dojde k urychlení instalace a přípravy prostředí pro spuštění programu FLASH.

Nejdříve je potřeba vytvořit adresář **/tmp/install\_flash**, do kterého nakopírujeme všechny zabalené soubory s instalacemi Intel Fortran Compileru, MPICH2, HDF5 a FLASH 3.3, které budeme potřebovat při instalaci. Adresář vytvoříme příkazem:

```
mkdir /tmp/install_flash
```

V adresáři **/tmp/install\_flash** dále musíme vytvořit adresář **SCRIPTS**, do kterého nakopírujeme jednotlivé instalační skripty.

```
mkdir /tmp/install_flash/SCRIPTS
```

Následující skripty je potřebné spouštět jako normální uživatel, během instalace budeme vyzváni k zadání hesla pro uživatele root.

Pro instalaci Intel Fortran Compileru se spustí instalační skript následujícím příkazem (v adresáři SCRIPT):

```
./installfortran.sh
```

Tento skript rozbálí soubor s instalací, vytvoří potřebné adresářové struktury a spustí instalaci Intel Fortran Compileru. Po skončení instalace doplní tento skript do proměnných prostředí potřebné proměnné.

Pro instalaci MPICH2 se spustí instalační skript následujícím příkazem:

```
./installmpich2.sh
```

Tento skript rozbálí soubor s instalací, vytvoří potřebné adresářové struktury a spustí instalaci MPICH2. Po skončení instalace vytvoří soubor s heslem pro démona MPD a doplní do proměnných prostředí potřebné hodnoty.

Pro instalaci HDF5 se spustí instalační skript následujícím příkazem:

```
./installhdf5.sh
```

Tento skript rozbálí soubor s instalací, vytvoří potřebné adresářové struktury a spustí instalaci HDF5. Po skončení instalace doplní tento skript do proměnných prostředí potřebné hodnoty.

Pro instalaci programu FLASH 3.3 se spustí instalační skript následujícím příkazem:

```
./installflash.sh
```

Tento skript v domovském adresáři vytvoří adresář s názvem **ASTRO\_SIM**, rozbálí do něj soubor s instalací, vytvoří adresář podle HOSTNAME a nakopíruje do něj vzorový soubor makefile.h potřebný pro kompilaci programu FLASH.

Pro instalaci vizualizačního programu IDL můžeme spustit instalační skript následujícím příkazem:

```
./installidl.sh
```

Tento skript vytvoří potřebné adresářové struktury a rozbalí do nich soubor s instalací. Po skončení instalace doplní do proměnných prostředí potřebné hodnoty.



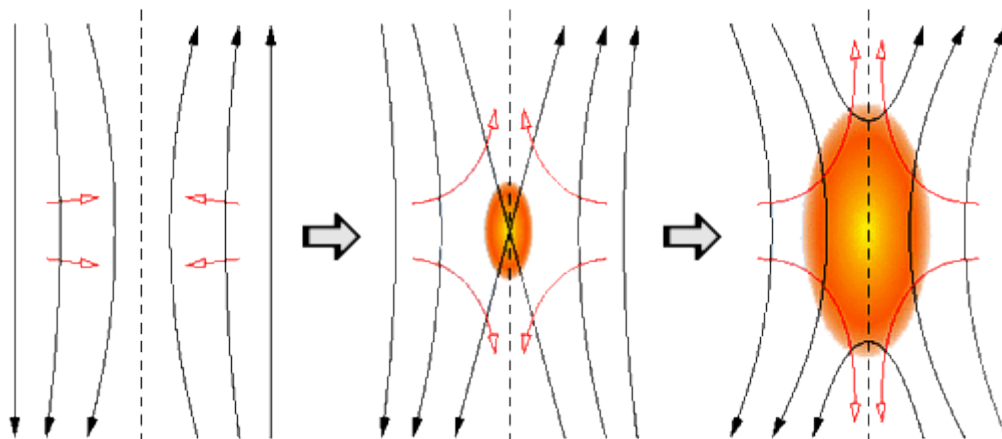
## 4. Výpočty úloh

V poslední části této diplomové práce budou provedeny některé výpočty týkající se vln ve sluneční koruně. Jako testovací příklad byl zvolen kód pro rekonekci magnetického pole.

### 4.1. Výpočet testové úlohy

Jako testovací příklad jsme zvolili příklad CurrentSheet, který je součástí vzorových příkladů distribuovaných přímo v instalaci programu FLASH. Tento vzorový příklad simuluje rekonekci magnetického pole, která je blízka řešení problému ohřevu korony. Výpočet testové úlohy slouží také zároveň jako ověření správného nainstalování všech součástí programu FLASH.

Magnetická rekonekce je základní proces, který probíhá v plazmatu a dochází při něm k rychlému uvolnění energie plazmatu [21]. Jde o jev, s jehož pomocí se daří vysvětlovat řadu dosud nepochopených dějů v plazmatu. Mezi ně patří například náhlá vzplanutí v koruně Slunce, jevy na hranici magnetosféry Země a meziplanetárního prostoru, jevy v atmosférách magnetarů, stejně tak jako jevy v laboratorním plazmatu [22].

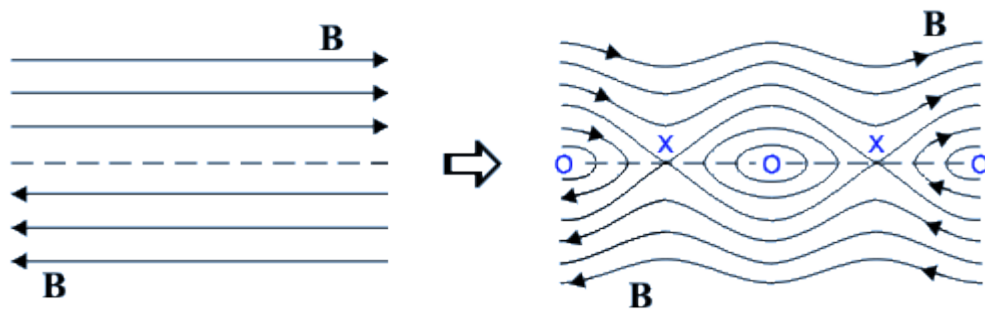


**Obr. 5:** Průběh rekonekce magnetických silokřivek, převzato z [23].

Obdobně, jako elektrický proud si hledá nejkratší, nejvýhodnější cestu, může dojít k tomu, že magnetické silokřivky prudce změni svou dosavadní topologii do jiné, energeticky výhodnější podoby. Na obrázku 5 je znázorněna situace, kdy se v plazmatu k sobě přiblíží dvě oblasti magnetického pole s opačně orientovanými silokřivkami, v této oblasti vzniká tzv. difúzní region. Ten je charakteristický velmi nízkou hodnotou magnetického pole. Právě zde dojde ke změně topologie magnetických silokřivek, jejich přepojení do nové konfigurace s nižší energií. Makroskopický pohyb plazmatu je při přepojení také ovlivněn. Před přepojením se plazma pohybuje kolmo na silokřivky směrem do difúzního regionu (červené šipky na obrázku). Po

přepojení je plazma vytlačováno ve směru původní orientace magnetických siločar. Podle tvaru magnetických siločar se střed difúzního regionu, ve kterém je nulové pole, někdy nazývá neutrální bod typu X. V třírozměrné situaci tvoří hodnoty nulového magnetického pole celou křivku. V některých situacích způsobí nestability opakované přepojení magnetických siločar s periodicky se opakujícími body nulového pole tvaru X a O, jak je znázorněno na obrázku 6.

Při přepojení tečou v difúzním regionu velké elektrické proudy, které zahřívají plazma. Energie magnetického pole je transformována do tepelné energie plazmatu. Horké plazma nadbytečnou energii intenzivně vyzařuje do okolí i v RTG oboru [24].



**Obr. 6:** Vlevo opačně orientovaná magnetická pole, vpravo je zobrazena rekonekce magn. pole s periodicky se opakujícími body nulového pole tvaru X a O. Převzato z [23].

#### 4.1.1. Počáteční podmínky

Tento testovací příklad slouží především pro ověření instalace, proto nebudeme měnit simulační ani počáteční parametry, které jsou již nastaveny v definičních souborech tohoto příkladu.

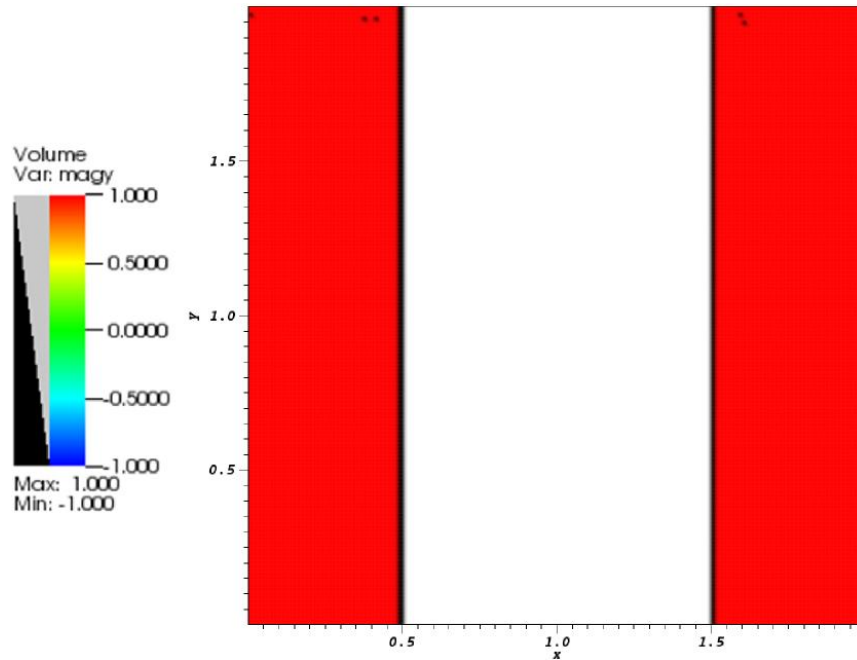
Zvolený testovací příklad simuluje 2D rekonekci magnetického pole a řeší ji pomocí ideálních MHD rovnic.

Na obrázku 7 jsou znázorněny počáteční podmínky pro simulaci. Jsou zde dvě magnetické pole orientované proti sobě.

Tyto pole inicializujeme s následujícími podmínkami:

$$B_y = \begin{cases} B_0, 0 \leq x < 0,5 \\ -B_0, 0,5 \leq x < 1,5 \\ B_0, 1,5 \leq x < 2,0 \end{cases} \quad (1)$$

Kde  $B_0 = 1$ . Ostatní složky magnetického pole ( $B_x$ ,  $B_z$ ) mají hodnotu 0. Výpočetní box, ve kterém budou prováděny simulace má rozměry  $[0, 2] \times [0, 2]$ . Všechny veličiny v tomto příkladu jsou bezrozměrné.



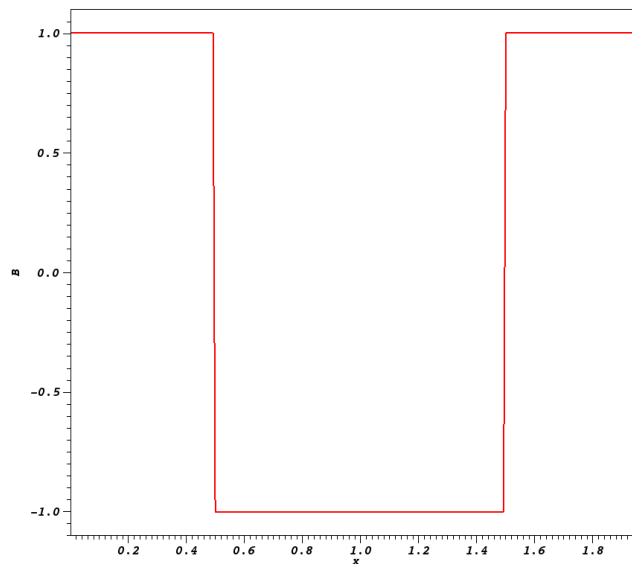
**Obr. 7:** Zobrazení simulačního boxu na počátku simulace a barevné stupnice hodnot. V čase  $t=0$  s.

Z barevné stupnice hodnot magnetického pole, která je zobrazena vlevo na obrázku 7, vidíme, že červená barva znamená hodnotu magnetického pole 1.0 a bílá barva -1.0. To znamená, že jsou tato pole orientovaná proti sobě. Ke stejným hodnotám magnetického pole dospějeme, pokud provedeme řez simulačním boxem ve směru osy  $x$ , jak je zobrazeno na obrázku 8.

V tomto simulačním boxu vygenerujeme pulz rychlosti ve směru osy  $x$ . Rychlost složky  $x$  je generována vztahem (2)

$$x = U_0 \cdot \sin 2\pi y \quad (2)$$

kde  $u_0 = 0.1$  a ostatní složky rychlosti mají hodnotu 0. Hustota plazmatu má jednotkovou velikost a tlak je  $p = 0,1$ . Doba, po kterou bude probíhat výpočet je nastavena v inicializačních parametrech tohoto příkladu na 4 s.



**Obr. 8:** Počáteční nastavení simulačního boxu - řez simulačním boxem ve směru osy x.

#### 4.1.2. Nastavení prostředí pro výpočty, sestavení kódu a spuštění výpočtu

Před spuštěním výpočtů je potřeba provést několik kroků, které jsou nutné pro správný běh programu FLASH. Nejdříve je potřeba smazat obsah adresáře **object**, aby v něm nezůstaly soubory z předchozích výpočtů. To se provede příkazem:

```
rm -f object/*
```

Dalším příkazem nakonfigurujeme FLASH pro výpočty úlohy, kterou potřebujeme. Soubory s nastavením a podmínkami pro výpočty vždy v jednom adresáři, v tomto případě, pro naši zvolenou úlohu, jsou v adresáři **magnetoHD/CurrentSheet**.

Příkaz **setup** podle konfiguračních souborů nakopíruje všechny potřebné moduly, knihovny a nastavení do adresáře **object**.

```
./setup magnetoHD/CurrentSheet -auto -2d +usm -maxblock=1000 -noclobber
```

Přepneme se do adresáře **object**:

```
cd object/
```

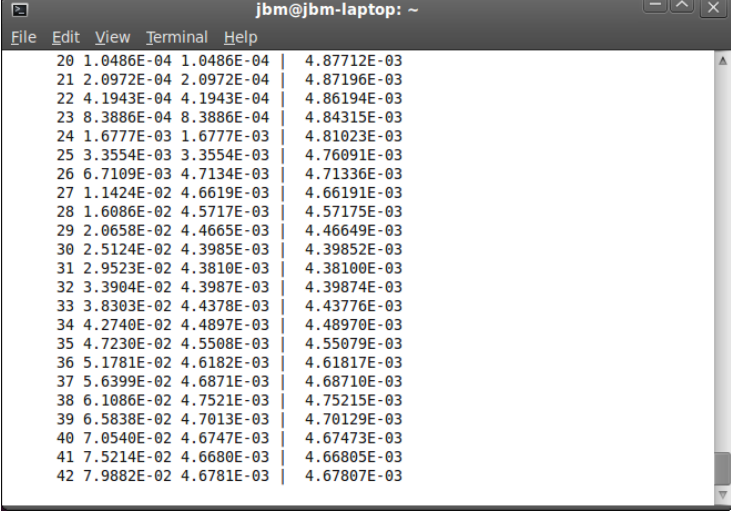
Následujícími příkazy sestavíme ze zdrojových kódů, které jsou v adresáři **object**, spustitelný kód:

```
make
mpd &
```

```
mpirun -np 1 ./flash3
```

Parametrem `-np <np>` volíme počet procesů, které se budou používat při výpočtech. V tomto případě byl zvolen jen jeden.

Po zadání příkazu na spuštění výpočtu budou v okně terminálu přibývat řádky s jednotlivými kroky výpočtu, jak je zobrazeno na obr. 9.



```
jbm@jbm-laptop: ~
File Edit View Terminal Help
20 1.0486E-04 1.0486E-04 | 4.87712E-03
21 2.0972E-04 2.0972E-04 | 4.87196E-03
22 4.1943E-04 4.1943E-04 | 4.86194E-03
23 8.3886E-04 8.3886E-04 | 4.84315E-03
24 1.6777E-03 1.6777E-03 | 4.81023E-03
25 3.3554E-03 3.3554E-03 | 4.76091E-03
26 6.7109E-03 4.7134E-03 | 4.71336E-03
27 1.1424E-02 4.6619E-03 | 4.66191E-03
28 1.6086E-02 4.5717E-03 | 4.57175E-03
29 2.0658E-02 4.4665E-03 | 4.46649E-03
30 2.5124E-02 4.3985E-03 | 4.39852E-03
31 2.9523E-02 4.3810E-03 | 4.38100E-03
32 3.3904E-02 4.3987E-03 | 4.39874E-03
33 3.8303E-02 4.4378E-03 | 4.43776E-03
34 4.2740E-02 4.4897E-03 | 4.48970E-03
35 4.7230E-02 4.5508E-03 | 4.55079E-03
36 5.1781E-02 4.6182E-03 | 4.61817E-03
37 5.6399E-02 4.6871E-03 | 4.68710E-03
38 6.1086E-02 4.7521E-03 | 4.75215E-03
39 6.5838E-02 4.7013E-03 | 4.70129E-03
40 7.0540E-02 4.6747E-03 | 4.67473E-03
41 7.5214E-02 4.6680E-03 | 4.66805E-03
42 7.9882E-02 4.6781E-03 | 4.67807E-03
```

**Obr. 9:** Průběh výpočtu simulace.

### 4.1.3. Spouštěcí skript

Pro usnadnění spuštění programu a automatizaci všech předchozích kroků, aby se nemusely dokola vypisovat dlouhé příkazy, byl vytvořen spouštěcí skript:

```
./runflash.sh
```

Spuštěním tohoto skriptu se provedou všechny kroky, které byly popsány výše, zkontroluje se, zda běží daemon MPD, popřípadě se spustí a nakonec se spustí výpočty nakonfigurované úlohy.

### 4.1.4. Zobrazení výsledků

Výsledky výpočtů se ukládají do souborů **plot** v adresáři **FLASH3.3/object**, které můžeme potom graficky zobrazit pomocí vizualizačního software. V tomto adresáři vzniknou soubory **currentSheet\_mhd\_2d\_hdf5\_plt\_cnt\_xxxx**, kde **xxxx** je pořadové číslo souboru. Počet souborů s výsledky výpočtů je různý a závisí na nastavení simulačních parametrů příkladu.

#### 4.1.4.1. Zobrazení pomocí programu VisIt

Program VisIt můžeme spustit postupem popsáním v kapitole 3.5.2.2., ale pro urychlení práce byl také vytvořen skript pro jeho snadné spouštění.

```
./runvisit.sh
```

Tento skript spustí program VisIt již s potřebnými parametry a nastaví pracovní adresář na **FLASH3.3/object**, kde program FLASH ukládá soubory s vypočtenými výsledky. Odpadá tak dlouhé hledání souborů s výsledky v adresářové struktuře.

#### 4.1.4.2. Zobrazení výsledků pomocí programu IDL

Dále byl vytvořen skript pro snadné spouštění vizualizačního software IDL.

```
./runidl.sh
```

Tento skript spustí program IDL s parametry pro zobrazování dat z programu FLASH. Pro spuštění hlavního okna programu napíšeme příkaz:

```
IDL> xflash3
```

Otevře se hlavní grafické okno programu IDL.

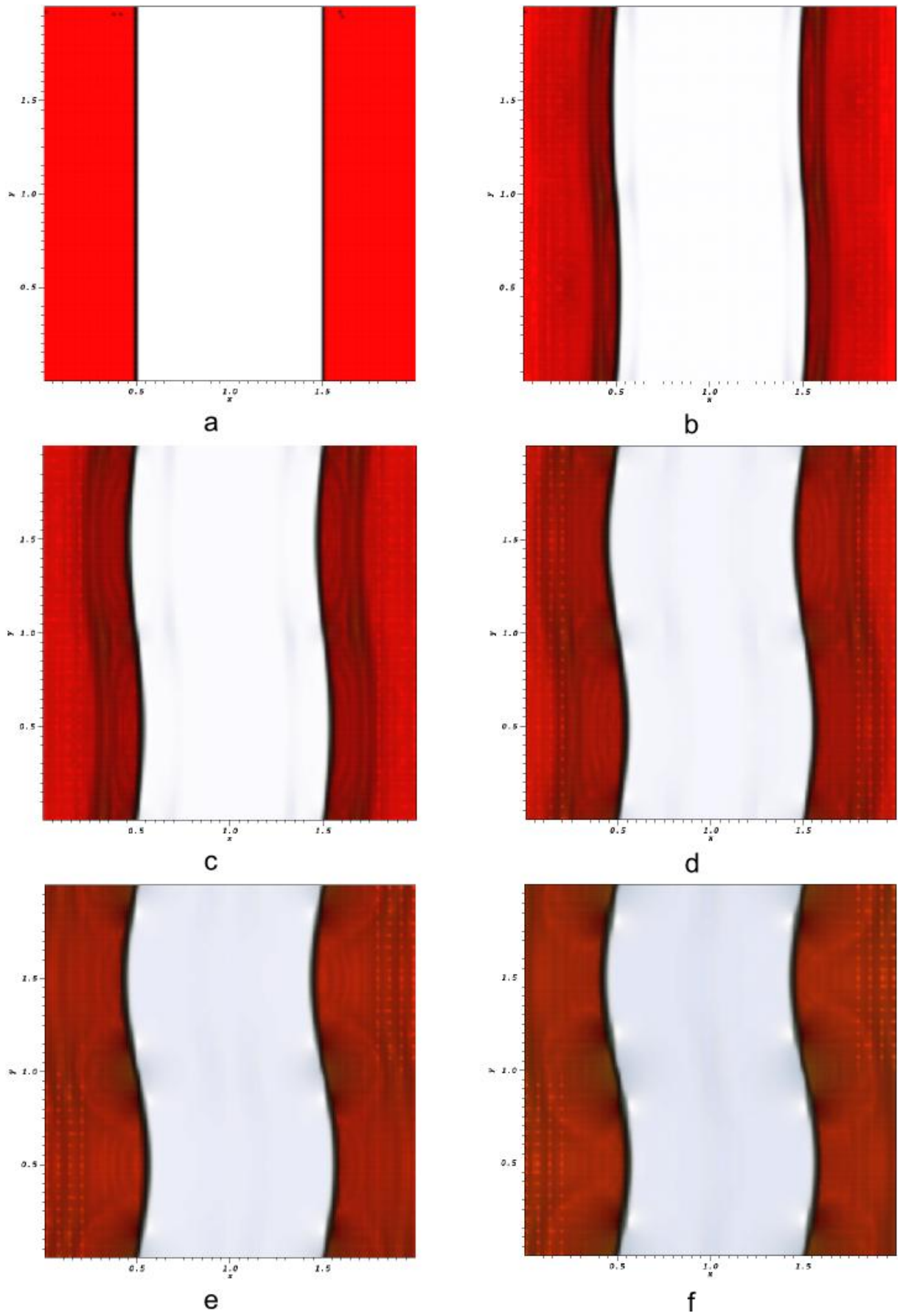
#### 4.1.4.3. Grafické zobrazení

Následující obrázky byly vytvořeny pomocí programu VisIt. Na obrázku 10 až 12 jsou zobrazeny postupně kroky průběhu simulace. Pro lepší zobrazení průběhu simulace, a aby byl lépe vidět vývoj bodů, kde dochází k magnetické rekonekci, byl zvolen krok zobrazení snímků po 0,1 s.

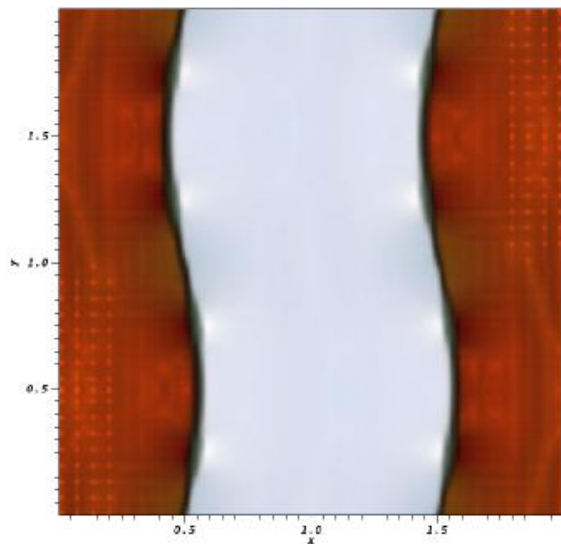
Na obrázcích je vidět, jak rozvlákněním magnetického pole vznikají malé ostrůvky nulových bodů, které se posouvají podél magnetických polí a spojují se do větších ostrovů nulového pole tvaru O. Mezi těmito ostrovy dochází k vytvoření bodů X a dojde zde ke spojení těchto dvou magnetických polí.

Z obrázků je patrné, že zde došlo k opakovanému přepojení magnetických siločar a vytvořili se zde také periodicky se opakující body nulového pole tvaru X a O.

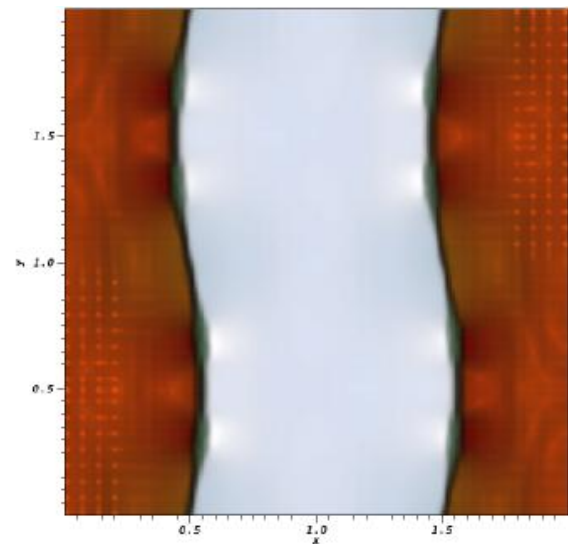
Pro porovnání zobrazení grafických výstupů z různých programů, byl vytvořen obrázek 13 v programu IDL. Na tomto obrázku zobrazen průběh simulace v čase  $t = 4$  s, což odpovídá obrázku 11i.



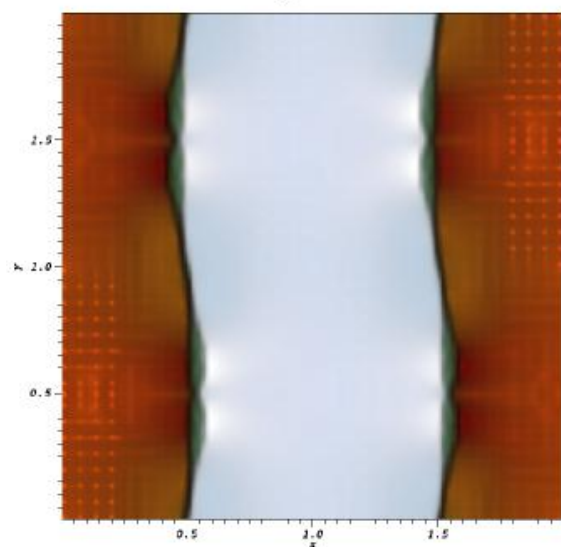
**Obr. 10:** Zobrazení výsledků simulace CurrentSheet pomocí programu VisIt. Zobrazení v krocích po 0,1 s, čas  $t = 0 - 0,5$  s.



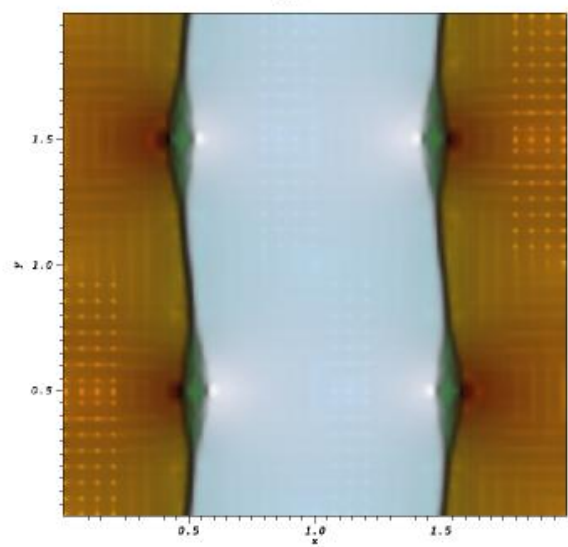
g



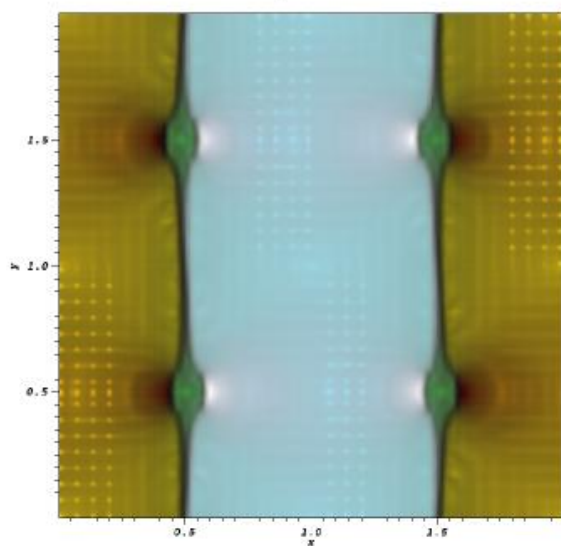
h



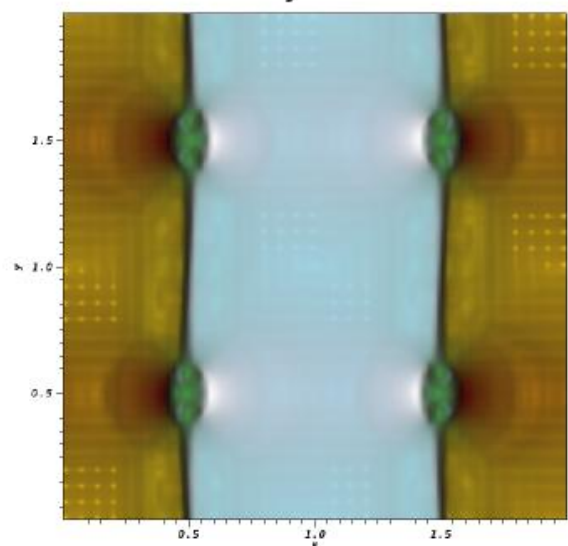
i



j



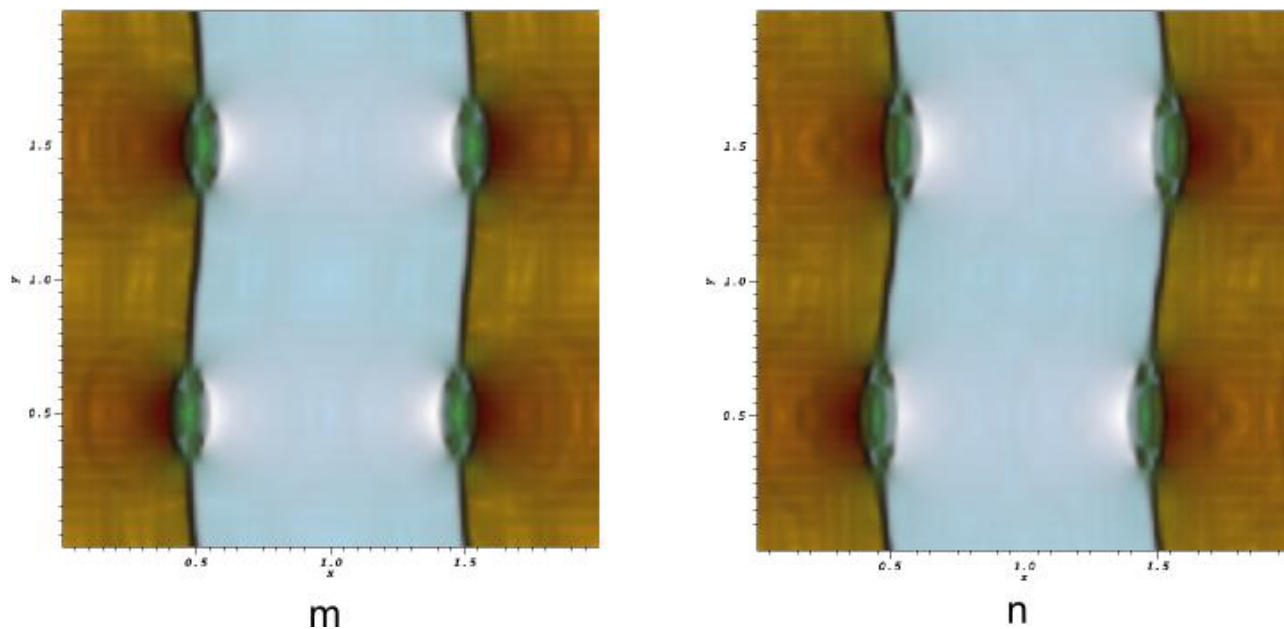
k



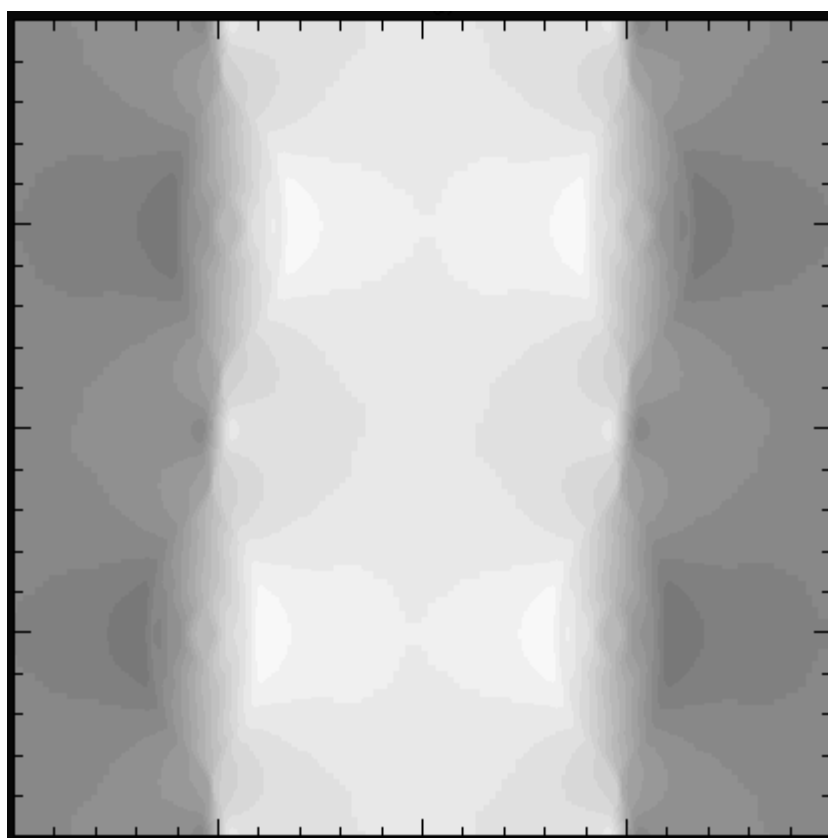
l

**Obr. 11:** Zobrazení výsledků simulace CurrentSheet pomocí programu VisIt. Zobrazení v krocích po 0,1 s, čas  $t = 0,6 - 1,1$  s.





**Obr. 12:** Zobrazení výsledků simulace CurrentSheet pomocí programu VisIt. Zobrazení v krocích po 0,1 s, čas  $t = 1,2 - 1,3$  s.



**Obr. 13:** Zobrazení výsledků pomocí programu IDL. Zobrazení v čase  $t = 4$  s.

## 4.2. *Vlastní simulace*

V poslední části této diplomové práce budou provedeny některé výpočty a simulace týkající se vln v koronálních smyčkách ve sluneční koruně.

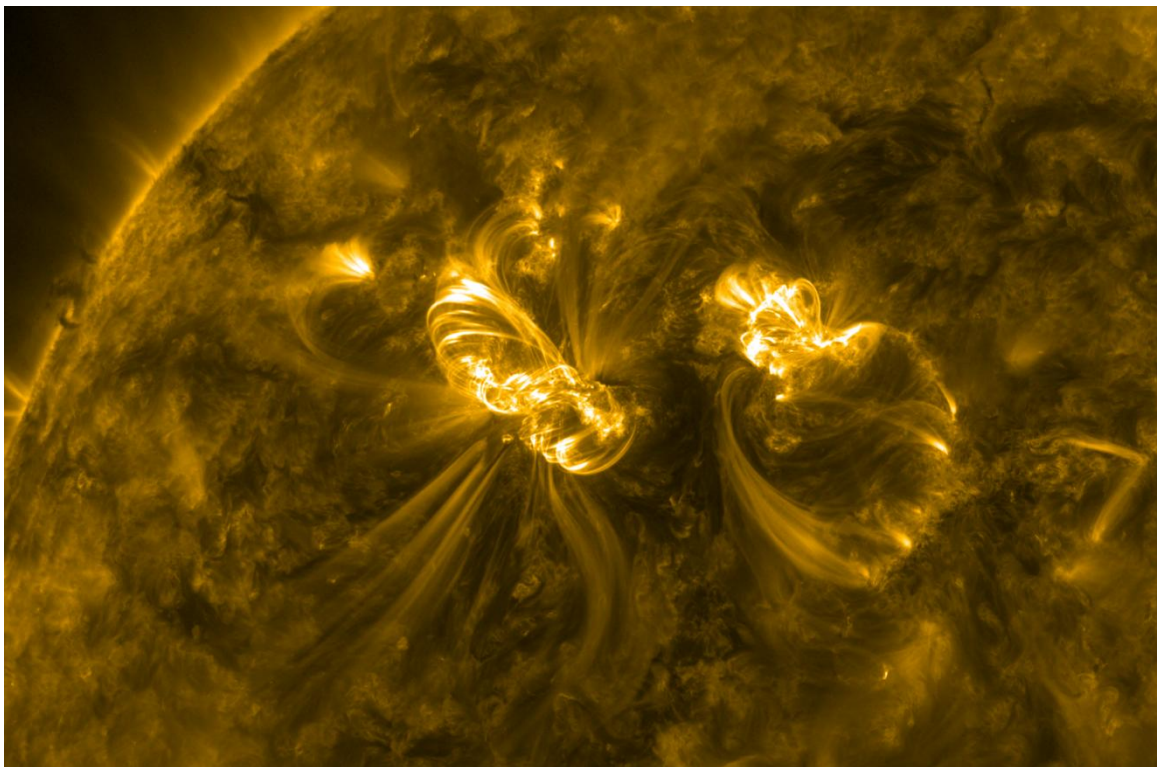
Vlny a oscilace ve slunečních koronálních smyčkách jsou pozorovány a studovány, ať už teoreticky nebo i pozorováním, již dlouhou dobu. Teprve s příchodem nových družic, jako je SUMER (SOHO) a TRACE, které mají dobré prostorové rozlišení a vysoce citlivé přístroje, bylo možné detekovat různé druhy oscilací v koronálních smyčkách. Třebaže teorie MHD kmitů je rozvíjena již několik desetiletí, pouze nová pozorování z kosmických sond poskytují údaje v dostatečném časovém rozlišení a poskytují velké množství údajů, dovolující kvantitativní aplikaci teoretických vztahů popisujících MHD vlny. Tyto nové objevy daly vzniknout nové disciplíně, známé jako koronální seismologie.

Výzkum v této oblasti je důležitý ve spojení s problémem koronálního ohřevu, urychlováním slunečního větru a koronální seismologií, protože by to mohlo vysvětlit některé dosud neobjasněné procesy ve sluneční fyzice.

Nově byly v koruně identifikovány oscilace koronálních smyček, protuberancí a plazmatu v aktivních oblastech nad slunečními skvrnami. Pozorované oscilace obsahují příčné a stojaté magnetoakustické vlny. Byly také pozorovány různé druhy rychlých vln: horizontální a vertikální „kink“ vlny a také rychlé „sausage“ vlny. Pomocí Dopplerových posuvů spektra lze pozorovat také rychlostní výkyvy šíření vln v plazmatu způsobené torzními Alfvénovskými vlnami [25, 26].

Nedávno objeveno i šíření MHD vln s pohybujícími se uzly. Šíření těchto MHD vln je v takovémto případě zapříčiněno impulsivně generovanými vzruchy v časovém měřítku rychlejším než je Alfvénova doba nebo času přeběhu akustických vln napříč strukturou. Šíření pomalých módů MHD vln (akustických rychlostí) bylo detekováno v koronálních smyčkách sondami TRACE a SoHO/EIT. Byly převážně iniciovány s třiminutovou periodou v blízkosti slunečních skvrn nebo s pětiminutovou periodou v klidných oblastech. Tyto akustické vlny se šíří nahoru smyčkou od její základny a jsou rychle utlumeny. Vlny s nejvyšším výskytem ve sluneční koruně, jsou postupně MHD vlny, které jsou komplexní a šíří se sféricky po celém slunečním povrchu následně po erupcích anebo koronálním výronu hmoty [27, 28, 29].

Na obrázku 14 jsou vyfotografovány tzv. poerupční koronální smyčky, které vznikají jako doprovodný efekt po slunečních erupcích. Těmito smyčkami se šíří impulzně generované vlny od základny smyčky vzhůru.



**Obr. 14:** Poerupční koronální smyčky na Slunci. Snímek pořízen dne 7.3.2012 v 05:32:36 UT sondou SDO

Cílem této simulace bude vytvořit jednoduchý model koronální smyčky, kde vygenerujeme impuls a budeme pozorovat, jak se tento impuls šíří.

#### 4.2.1. Výpočetní model

V plasmové fyzice existuje několik výpočetních metod, pomocí kterých je možné popsat a řešit daný problém. V tomto výpočetním modelu popisujeme dynamiku plazmatu v koronálních smyčkách pomocí následujících ideálních MHD rovnic [30].

$$\frac{\partial \rho}{\partial t} = -\vec{\nabla} \cdot (\rho \vec{v}) \quad (3)$$

$$\rho \frac{\partial \vec{v}}{\partial t} + \rho (\vec{v} \cdot \nabla) \vec{v} = -\nabla p + \frac{1}{\mu_0} (\vec{\nabla} \times \vec{B}) \times \vec{B} \quad (4)$$

$$\frac{\partial \vec{B}}{\partial t} = \vec{\nabla} \times (\vec{v} \times \vec{B}) \quad (5)$$

$$\frac{\partial U}{\partial t} = -\nabla \vec{S} \quad (6)$$

$$\nabla \cdot \vec{B} = 0 \quad (7)$$

Kde  $\rho$  je hustota,  $v$  je rychlost proudění plazmatu,  $p$  je tlak plynu a  $\mathbf{B}$  je magnetické pole.

Hustota energie v plazmatu  $U$  je dána vztahem

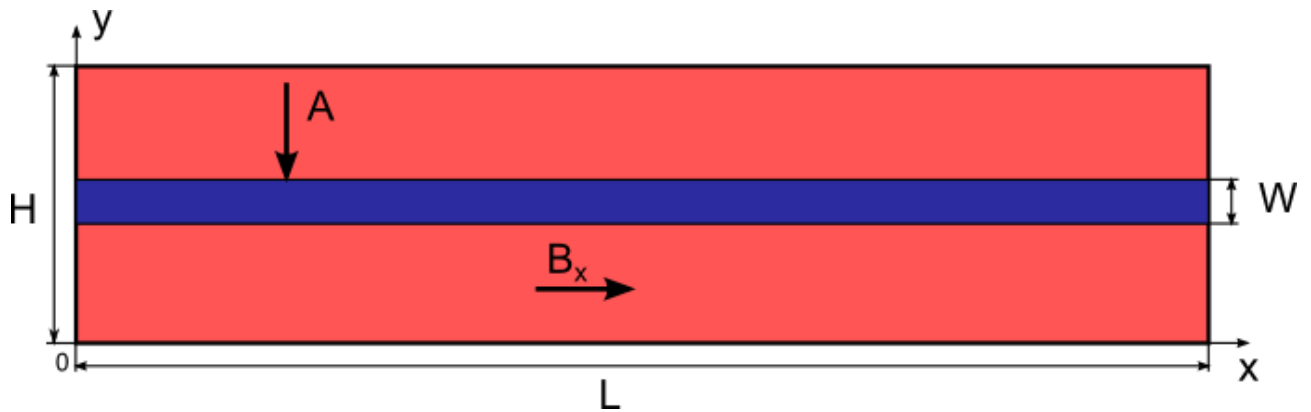
$$U = \frac{p}{\lambda - 1} + \frac{\rho}{2} \vec{v}^2 + \frac{B^2}{2\mu_0} \quad (8)$$

Kde  $\lambda$  je adiabatický koeficient  $\lambda = 5/6$ . Vektor proudění  $\mathbf{S}$  je vyjádřen vztahem

$$\mathbf{S} = \left( U + p + \frac{B^2}{\mu_0} \right) \cdot \mathbf{v} - (\mathbf{v} \cdot \mathbf{B}) \frac{\mathbf{B}}{\mu_0} \quad (9)$$

#### 4.2.2. Počáteční podmínky

Na obrázku 15 je zobrazen model koronální smyčky, který je založen na reálném modelu koronální smyčky. Pro naše potřeby počítačové simulace zobrazen jako dvourozměrný slab (slab - hustotní pás). Je to vlastně projekce tří rozměrného objektu koronální smyčky do dvou rozměrného objektu. Když se podíváme ze shora na tuto smyčku, tak se její projekce na plochu promítne jako úzký pás. Výpočty budeme provádět v simulačním boxu, ve kterém jsou dvě oblasti s různou hustotou.

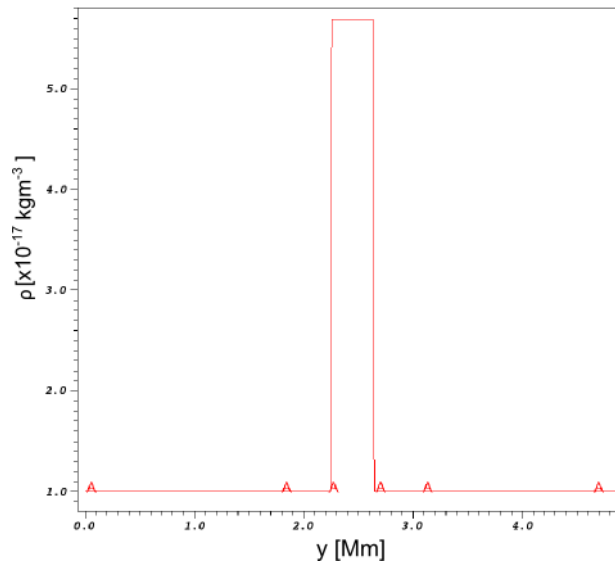


**Obr. 15:** Nákres 2D slabu v simulačním boxu a počátečních podmínek.

Uprostřed tohoto boxu je pás s větší hustotou (modrý pás), označíme jí jako hustotu interní  $\rho_i$ . Na okrajích je hustota menší (červený pás), označíme jí jako hustotu externí  $\rho_E$ . Mezi interní a externí hustotou v tomto pásu platí následující vztah:

$$\rho_i = d \cdot \rho_E \quad (10)$$

Hustota v celé délce pásu má stejnou velikost, a pokud bychom provedli řez přes celou šířku simulačního boxu ve směru osy  $y$ , měla by hustota plazmatu průběh, jaký je zobrazen na obrázku 16.



**Obr. 16:** Průřez 2D slabem ve směru osy  $x$ , zobrazení průběhu hustoty v čase  $t = 0$  s.

S tím také souvisí tlak plazmatu v tomto pásu, pro který platí vztah z rovnice (11).

$$p = \frac{\rho \cdot k \cdot T}{m} \quad (11)$$

Kde  $p$  je tlak,  $T$  je teplota,  $m$  je hmotnost,  $k$  je Stefan-Boltzmannova konstanta.

Pro naše podmínky platí, že  $k$ ,  $T$ ,  $m$  jsou v tomto případě konstanty a z toho vztahu (11) vyplývá, že tlak je přímo úměrný hustotě. Kdybychom udělali průřez tímto hustotním pásem, který zobrazuje tlak v plazmatu, měl by stejný průběh jako hustota.

Tento pás leží v magnetickém prostředí, kde magnetické pole je dáno tzv. parametrem  $\beta$ . Pro tento parametr platí známý vztah z rovnice podle (10).

$$\beta = \frac{p}{p_{mag}} = \frac{2\mu_0 p}{B^2} \quad (12)$$

Magnetické pole je souběžné s osou  $x$ , a je konstantní v celé simulační oblasti. Pro naše účely pozorování šíření vln v koronálních smyčkách v tomto hustotním pásu vygenerujeme v bodě A, viz obr. 15, rychlostní pulz ve směru osy  $x$  a budeme pozorovat, jak se jím šíří vlna. Pulz rychlosti je vygenerován dle vztahu [31]:

$$v_y = A_0 \cdot y \cdot \exp\left(-\frac{x^2}{\lambda_x^2}\right) \cdot \exp\left(-\frac{y^2}{\lambda_y^2}\right) \quad (13)$$

Kde  $A_0$  je počáteční amplituda pulzu. Šířky pulzů ve směru os x a y jsou  $\lambda_x = 1,5\text{Mm}$  a  $\lambda_y = 0,5\text{Mm}$ . Tento pulz vygenerujeme na začátku výpočetní simulace v čase  $t = 0$  s.

Simulační box, ve kterém budeme pozorovat šíření vln má rozměry  $H = 4,8$  Mm a  $L = 30$  Mm. Přitom slab je ve směru osy y podle obrázku rozdělen na dvě poloviny, pás s hustotou  $\rho_i$  je umístěn doprostřed a šířka tohoto pásu je  $w = 0,2$  Mm.

Rychlosti šíření zvukových a alfvénových vln uvnitř a vně slabu byly pro tuto simulaci nastaveny na následující hodnotu:

- Alfvénovská rychlost vně slabu  $v_{Ae} = 2,5$  km/s
- Rychlost šíření zvuku vně slabu  $v_{Se} = 0,2$  km/s
- Alfvénovská rychlost uvnitř slabu  $v_{Ai} = 1,0$  km/s
- Rychlost zvuku uvnitř slabu  $v_{Si} = 0,3$  km/s

### 4.2.3. Nastavení parametrů výpočtu

Pro nastavení výpočtů simulací je důležitý obsah adresáře, ze kterého skript **setup**, podle nastavení v konfiguračních souborech, nakopíruje všechny potřebné moduly, knihovny a nastavení do adresáře **object**. V našem případě je to adresář **Magneto HD/slab\_2d**.

V tomto adresáři je obsaženo několik souborů. Pro nastavení parametrů výpočtů jsou nejdůležitější následující dva soubory.

#### 4.2.3.1. Soubor flash.par

V tomto souboru se nacházejí především simulační parametry. To znamená parametry, které ovlivňují průběh a výstupy ze simulace. Pro naše potřeby patří mezi nejdůležitější parametry:

Velikost výpočetního boxu pro 2D výpočty

- **xmin** - minimální hodnota na ose x
- **xmax** – maximální hodnota na ose x
- **ymin** - minimální hodnota na ose y
- **ymax** - maximální hodnota na ose y

Parametry výstupních souborů

- **run\_comment** – poznámka k příkladu
- **log\_file** – název souboru, kam se ukládá hlášení o průběhu výpočtů, popřípadě chybové hlášení

- **basenm** – základní jméno souboru s výstupními daty.
- **tmax** - maximální doba výpočtu

Parametry výstupních kontrolních souborů

- **checkpointFileIntervalTime** – doba po kterou se zapisují kontrolní soubory (check file chk)

Parametry výstupních kontrolních souborů.

- **plotfileIntervalTime** - doba po jaké se zapisují plot soubory. Tyto soubory jsou výstupem ze simulace. Pokud chceme větší rozlišení, můžeme zapisovat po menší době. Můžeme zapisovat po 0,5 s nebo 0,1 s.
- **plot\_var\_n** – těmito parametry se mění, jaké veličiny chceme zobrazovat ve výstupním souboru plot. Tyto veličiny jsou nadefinované pro tento problém, nemůžeme je v podstatě měnit. Jsou zde nadefinované rychlosti (velx, vely, velz), hustota (dens), tlak (pres), magnetické pole (magx, magy, magz) a celkové magnetické pole (vecz).

Resizement, mění se zde hustota mřížky

- **nblockx** - Počet bloků na osu x
- **nblocky** - Počet bloků na osu y
- **lrefine\_min** - minimální resizing
- **lrefine\_max** - maximální resizing

#### 4.2.3.2. Soubor Simulation\_initBlock.F90

Dalším souborem, kde jsou nadefinované parametry pro výpočty, je soubor Simulation\_initBlock.F90. Jedná se především o popis fyzikálních parametrů výpočtu.

Parametry simulačního boxu

- **r0** - definice rozměrů jednotek délky v jednotkách m.
- **V0** – definice rozměrů jednotek rychlosti (Alfvénovskou rychlostí) v jednotkách m/s.
- **t0** - definice jednotek času.

Definice rychlostí v různých

- **c\_Ae** - Alfvénovská rychlost v okolním prostředí, v jednotkách V0
- **c\_se** - Rychlost zvuku v okolním prostředí, v jednotkách V0
- **c\_A0** - Alfvénovská rychlost ve slabu, v jednotkách V0
- **c\_s0** – Rychlost zvuku ve slabu, v jednotkách V0

Vlastnosti slabu

- **x\_slab** – pozice slabu na ose x
- **slab\_width** - Šířka slabu

Parametry inicializačního pulzu

- **angle\_V** – úhel působení pulzu ve směru osy X.
- **A\_p** - relativní amplituda tlakového pulzu
- **A\_rho** - amplituda hustotního pulzu
- **A\_Vx** - amplituda rychlostního pulzu ve směru osy x, v jednotkách  $c_{Ae}$
- **A\_Vy** - amplituda rychlostního pulzu ve směru osy y, v jednotkách  $c_{Ae}$
- **A\_Vz** - amplituda rychlostního pulzu ve směru osy z, v jednotkách  $c_{Ae}$
- **x0** - pozice počátečního pulzu ve směru osy x
- **y0** - pozice počátečního pulzu ve směru osy y
- **wpx** - šířka počátečního pulzu ve směru osy x, jednotky  $10^6$  m
- **wpy** - šířka počátečního pulzu ve směru osy y, jednotky  $10^6$  m

#### 4.2.4. Spuštění simulačních výpočtů

Pro spuštění výpočtů a provedení všech potřebných kroků před jejich spuštěním můžeme opět použít spouštěcí skript **runflash.sh**. Stačí do něj doplnit řádek s příkazem, který nastaví program pro výpočty tohoto problému. V textovém editoru gedit otevřeme soubor runflash.sh a doplníme do něj příkaz:

```
./setup magnetoHD/slab_2D -auto -2d +usm -maxblock=5000 -noclobber
```

Předchozí podobu příkazu **setup** můžeme v tomto souboru také ponechat a novou doplnit hned na řádek pod původní. Stačí před tento původní příkaz doplnit znak # a příkaz se již nebude spouštět. Tento postup je vhodný, pokud bychom chtěli opět spouštět původní příklad. Pak stačí odmazat křížek před příkazem a doplnit jej na jiném řádku, který již nechceme spouštět.

Nyní stačí výpočty spustit příkazem:

```
./runflash.sh
```



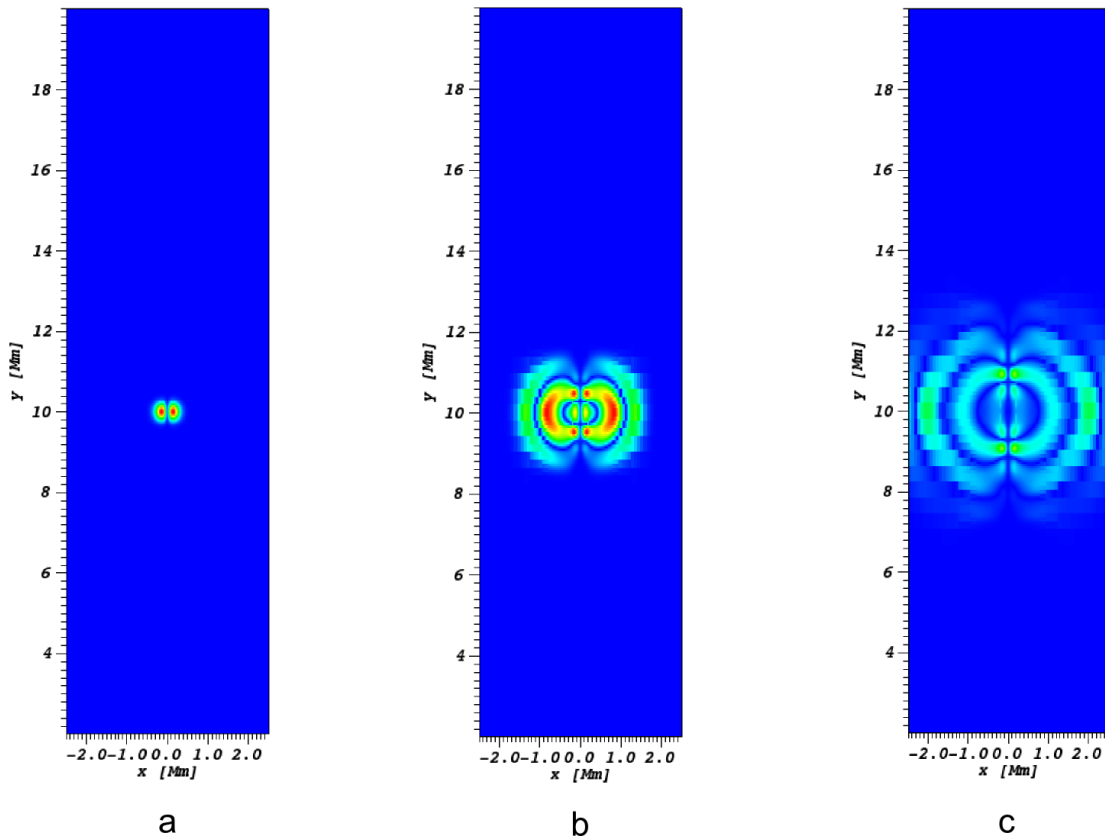
### 4.2.5. Výsledky výpočtů

Všechny obrázky z výstupních souborů byly opět vytvořeny pomocí programu VisIt. V adresáři **object/** vzniklo 30 souborů s názvem **Arcade\_hdf5\_plt\_cnt\_xxxx**, ve kterých jsou uloženy výsledky simulací po krocích, které jsme nadefinovali parametrem **plotfileIntervalTime**. V našem případě tedy po 0,5 s.

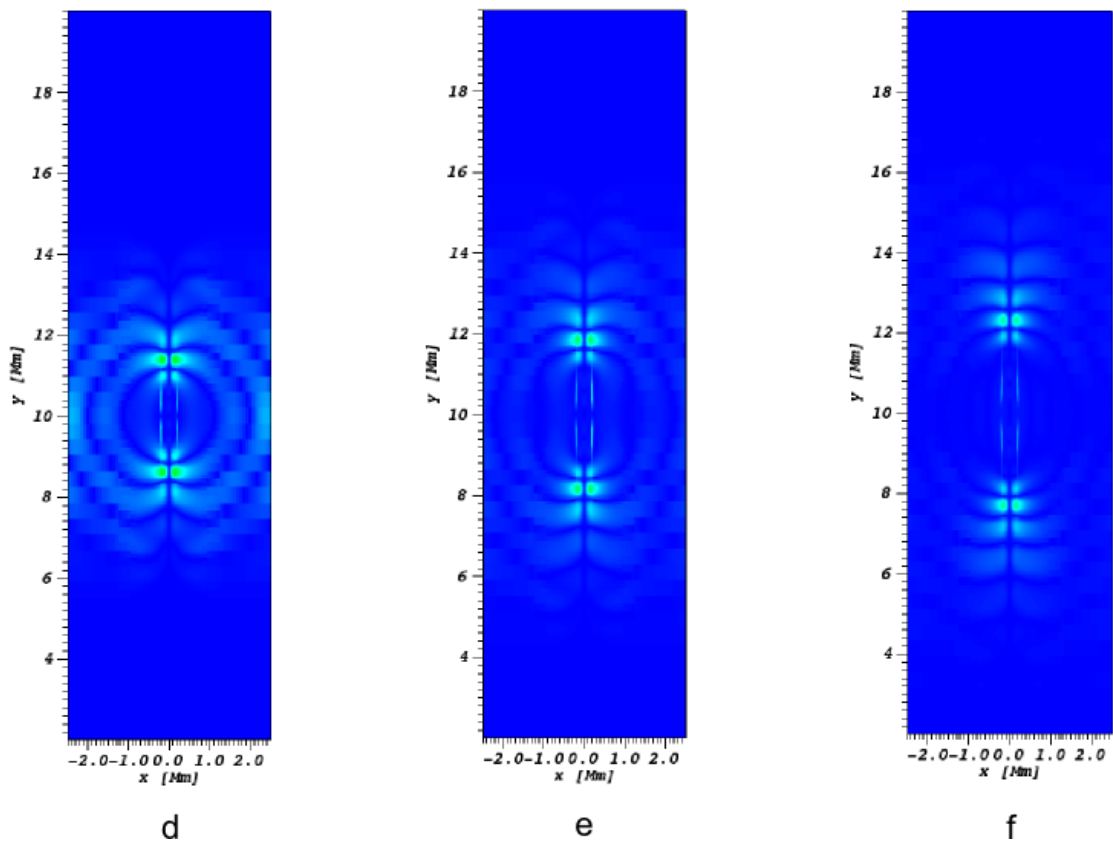
Na obrázku 17 až 19 je zobrazen postupné šíření vlny 2D slabem v různých časových intervalech. Graf zobrazuje celkovou rychlost šíření vlny v plazmatu. Pro lepší detail šíření vlny je grafu zobrazen pouze počátek celého 2D slabu o délce 20 Mm.

Na obrázku 20 jsou znázorněny grafy s průběhem vlny v čase 0 – 5 s, která postupuje 2D slabem. Tyto grafy byly získány tak, že byl proveden řez 2D slabem ve směru osy y. Řez byl proveden ve vzdálenosti +0,1 Mm přes celou délku slabu.

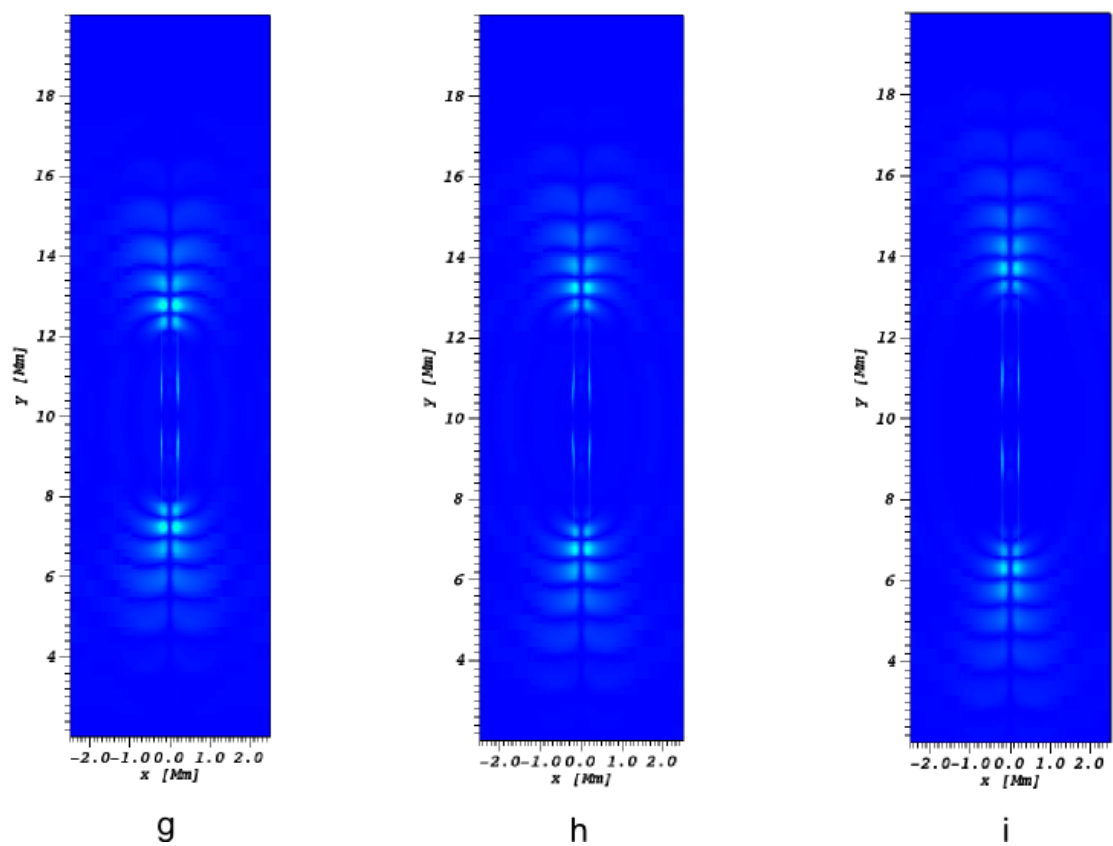
Pro porovnání, jak se šíří impulsně generovaná vlna 2D slabem různé šířky, byly provedeny další výpočty s nastavením šířky slabu na hodnoty 0,5 Mm a 0,8 Mm. Tyto simulace jsou zobrazeny na obrázcích 21-23, pro lepší přehlednost jsou tyto průběhy ze stejného časového okamžiku seřazeny na obrázcích vedle sebe.



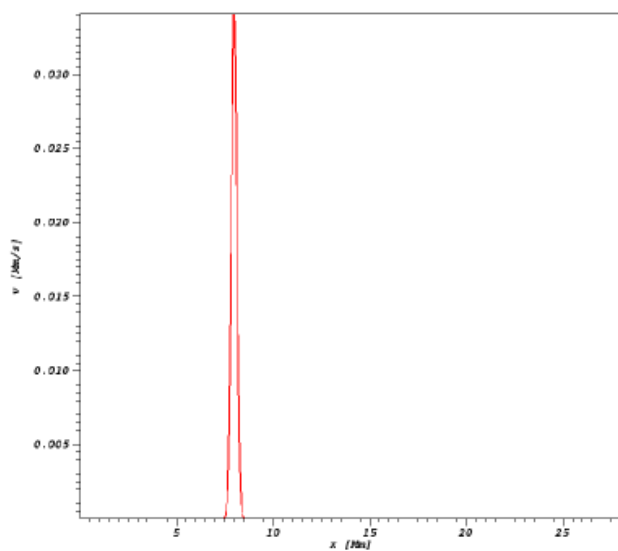
**Obr. 17:** Zobrazení šíření impulsně generované vlny 2D slabem v časech: a) 0 s; b) 0,5 s; c) 1 s;



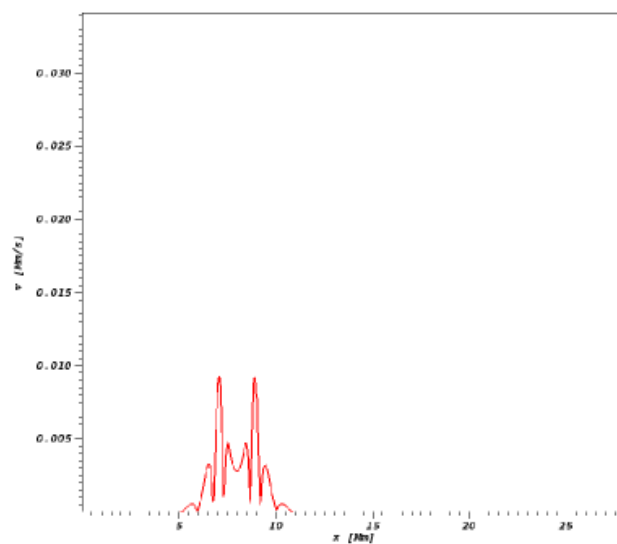
**Obr. 18:** Zobrazení šíření impulzně generované vlny 2D slabem v časech: d) 1,5 s; e) 2 s; f) 2,5 s;



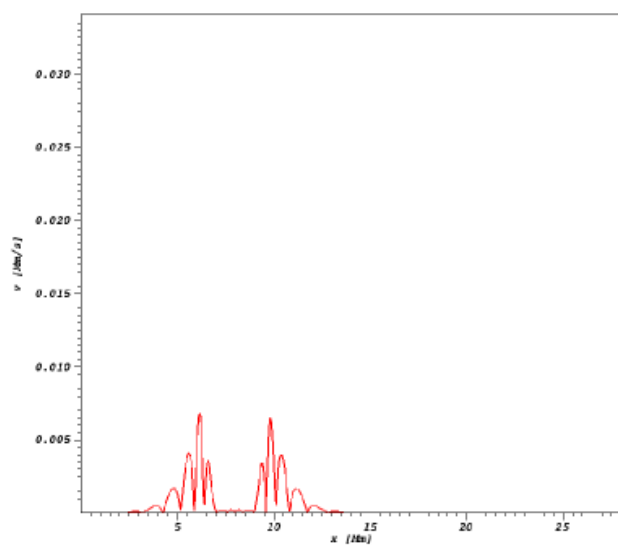
**Obr. 19:** Zobrazení šíření impulzně generované vlny 2D slabem v časech: g) 3 s; h) 3,5 s; i) 4 s;



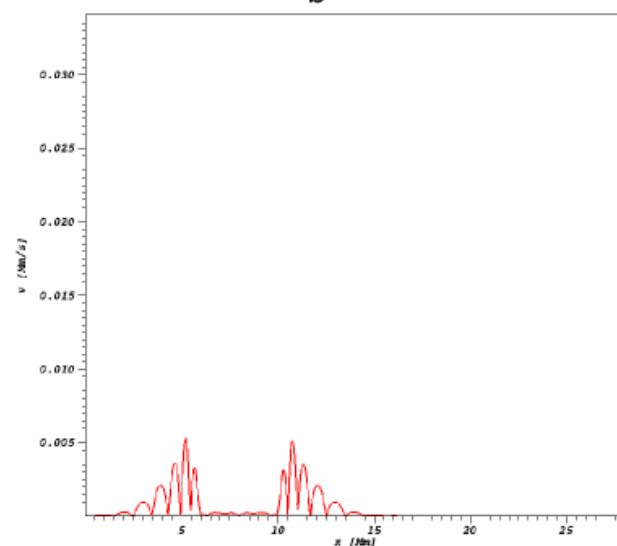
a



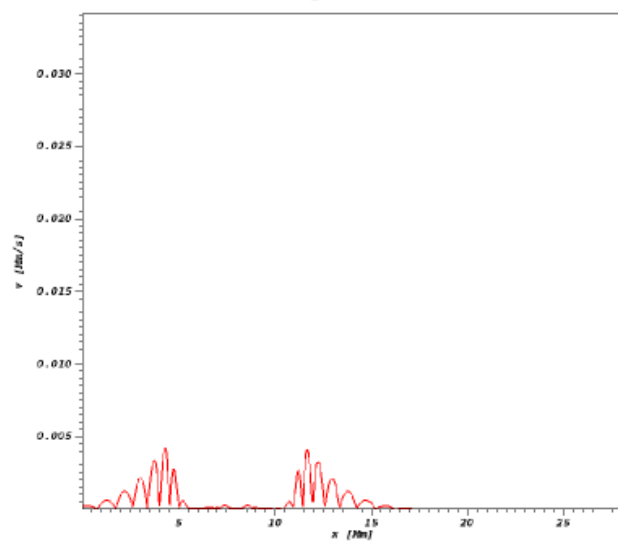
b



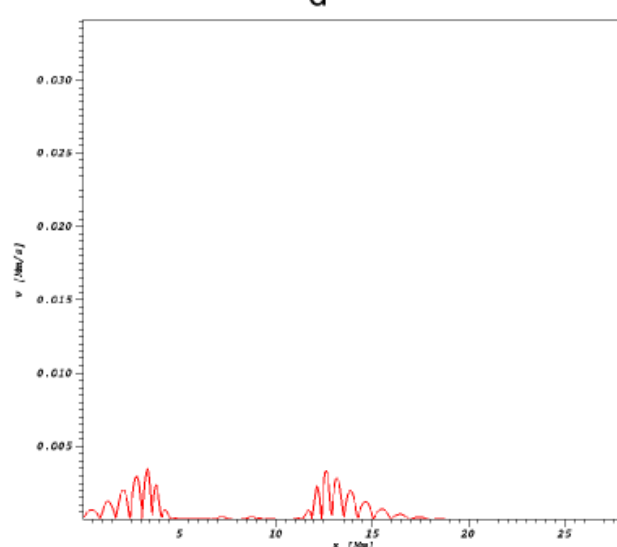
c



d

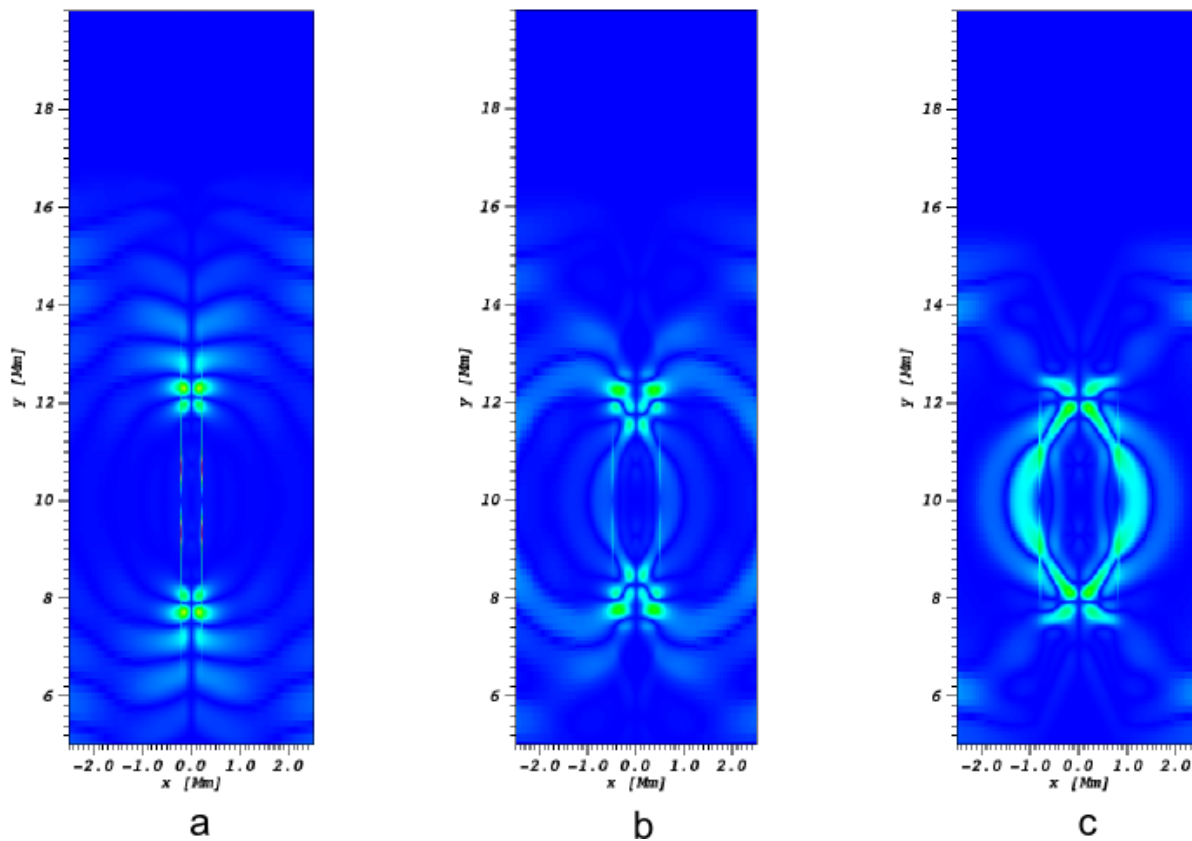


e

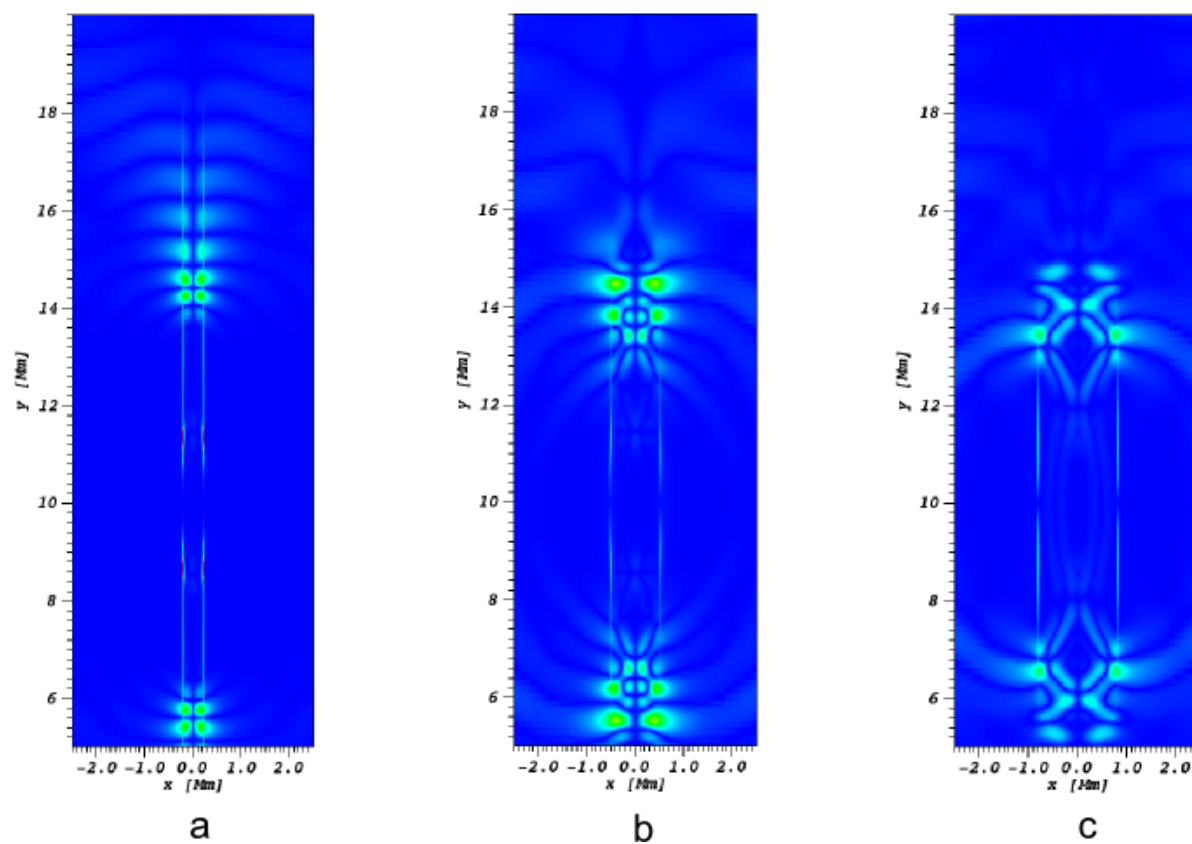


f

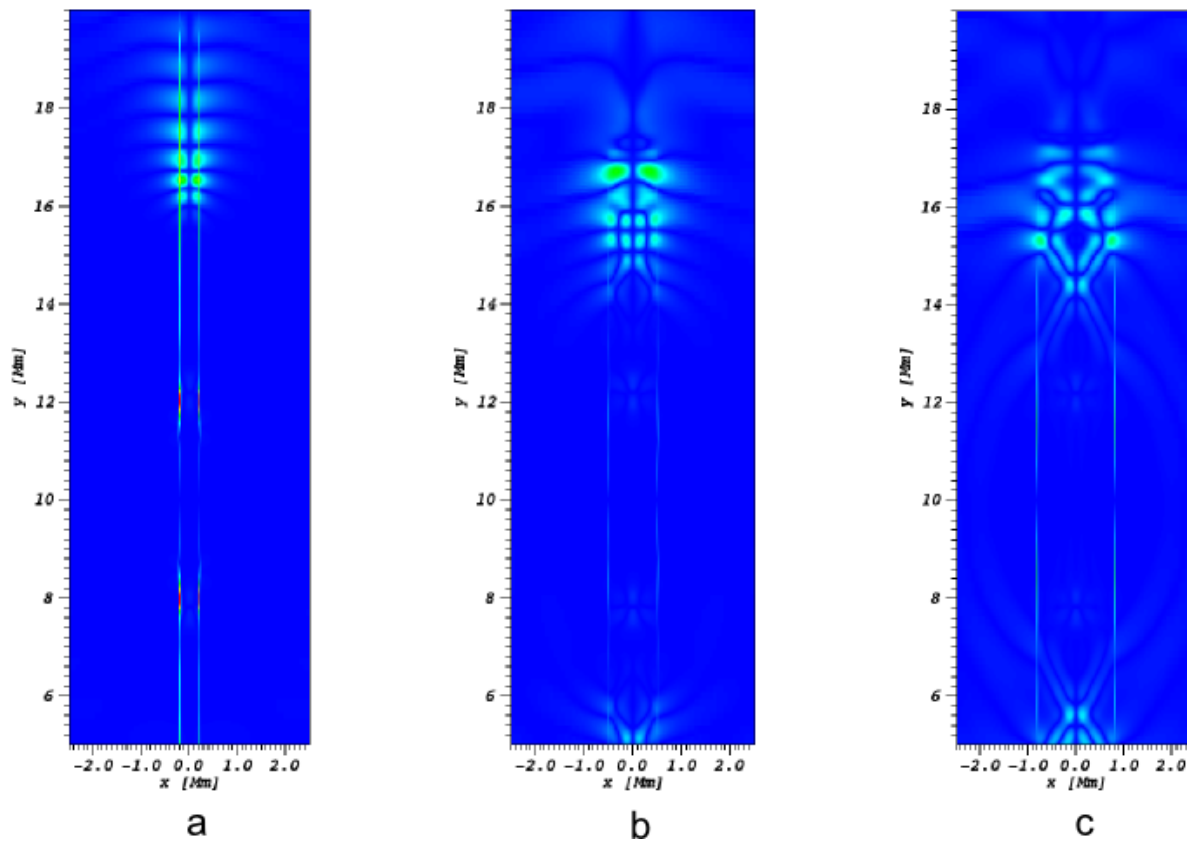
**Obr. 20:** Řez 2D slabem - zobrazení postupující vlny v časech: a) 0 s; b) 1 s; c) 2 s; d) 3 s; e) 4 s; f) 5 s;



**Obr. 21:** Porovnání šíření vlny slabem různých šířek a)  $w = 0,2$  Mm; b)  $w = 0,5$  Mm; c)  $w = 0,8$  Mm. Zobrazení v čase  $t = 2,5$  s.



**Obr. 22:** Porovnání šíření vlny slabem různých šířek a)  $w = 0,2$  Mm; b)  $w = 0,5$  Mm; c)  $w = 0,8$  Mm. Zobrazení v čase  $t = 5,0$  s.



**Obr. 23:** Porovnání šíření vlny slabem různých šířek a)  $w = 0,2$  Mm; b)  $w = 0,5$  Mm; c)  $w = 0,8$  Mm. Zobrazení v čase  $t = 7,5$  s.

## 5. Závěr

V úvodu této práce byly shrnuty základní vlastnosti a složení Slunce, detailněji se zaměřila na problém ohřevu sluneční korony.

V další části byl vytvořen přehledný manuál na instalaci numerického kódu FLASH, vizualizačního software VisIt, IDL a všech knihoven potřebných pro jejich spuštění.

V rámci tohoto manuálu byly také vytvořeny instalační skripty pro urychlení instalace, skripty pro konfiguraci, spouštění výpočtů a spuštění vizualizačního software s potřebnými parametry.

V závěrečné části byla pomocí testovací úlohy ověřena správnost kroků instalace všech programů z předchozí kapitoly. Dále byla vytvořena vlastní úloha, která se simuluje šíření vln v koronálních smyčkách Slunce. Byly provedeny výpočty podle nastavených počátečních podmínek a výsledky zobrazeny pomocí vizualizačního software VisIt.

Pomocí této práce je schopen případný zájemce o tuto oblast si nainstalovat potřebný software pro simulace a vytvořit svůj vlastní simulační problém, nebo rozšířit již vytvořenou úlohu například o vliv gravitačního pole.

## Literatura

- [1] Obrázek vrstev Slunce, [http://solarcellcentral.com/sun\\_page.html](http://solarcellcentral.com/sun_page.html), 29. 4. 2013
- [2] Zirker, Jack B. (2002). *Journey from the Center of the Sun*. Princeton University Press. pp. 15–34. ISBN 978-0-691-05781-1.
- [3] <http://www.asu.cas.cz/~svanda/prace/prednasky/surfsol.pdf>, 29. 4. 2013
- [4] [http://www.vzdelavaci-institut.info/?q=system/files/Slunecni\\_aktivita-o\\_vlivu\\_nejblizsi\\_hvezdy\\_na\\_Zemi-Petr\\_Kulhanek.pdf](http://www.vzdelavaci-institut.info/?q=system/files/Slunecni_aktivita-o_vlivu_nejblizsi_hvezdy_na_Zemi-Petr_Kulhanek.pdf), 29. 4. 2013
- [5] Ohřev sluneční koróny, [http://www.aldebaran.cz/bulletin/2009\\_09\\_cor.php](http://www.aldebaran.cz/bulletin/2009_09_cor.php), 29. 4. 2013
- [6] <http://www.astrosim.net/code/doku.php?id=home>, 29. 4. 2013
- [7] Fortran - <http://cs.wikipedia.org/wiki/Fortran>, 29. 4. 2013
- [8] Download Intel® Fortran Composer XE 2011 for Linux, <http://software.intel.com/en-us/articles/intel-composer-xe/>, Dostupné z internetu 20.1.2011
- [9] Registrace Intel® Fortran Composer XE 2011 for Linux, <https://registrationcenter.intel.com/RegCenter/AutoGen.aspx?ProductID=1523&AccountID=&EmailID=&ProgramID=&RequestDt=&rm=NCOM>, 29. 4. 2013
- [10] MPICH2, Home page, <http://www.mcs.anl.gov/research/projects/mpich2/>, 29. 4. 2013
- [11] Instalační balíčky programu MPICH2 pro UBUNTU 10.04, <http://www.mpich.org/downloads/>, 29. 4. 2013
- [12] MPICH2, Installer's Guide, <http://ftp.mcs.anl.gov/pub/mpi/mpich2-doc-install.pdf>, 29. 4. 2013
- [13] HDF5, Main page, <http://www.hdfgroup.org/>, 29. 4. 2013
- [14] HDF5, Download, <http://www.hdfgroup.org/ftp/HDF5/prev-releases/hdf5-1.6.10/src/>, 29. 4. 2013
- [15] FLASH, Home Page, <http://flash.uchicago.edu/site/>, 29. 4. 2013
- [16] FLASH 3.3., User's Manual, [http://flash.uchicago.edu/website/codesupport/flash\\_3\\_ug\\_3p3.pdf](http://flash.uchicago.edu/website/codesupport/flash_3_ug_3p3.pdf), 29. 4. 2013
- [17] Seznam programů pro grafickou vizualizaci dat, [http://astro-sim.org/index.php?option=com\\_content&task=view&id=26&Itemid=44](http://astro-sim.org/index.php?option=com_content&task=view&id=26&Itemid=44), 29. 4. 2013
- [18] VisIt, Home page, <https://wci.llnl.gov/codes/visit/home.html>, 29. 4. 2013
- [19] VisIt 2.3.2 Linux - x86\_64, 64 bit Ubuntu, Download, <https://wci.llnl.gov/codes/visit/executables.html>, 29. 4. 2013
- [20] IDL, fa. Exelis <http://www.exelisvis.com/ProductsServices/IDL.aspx>, 29. 4. 2013
- [21] Current sheet, <http://www.astro.virginia.edu/VITA/ATHENA/cs.html>, 29. 4. 2013

- [22] Magnetic Reconnection and Thin Current Sheets, <http://spp.astro.umd.edu/SpaceWebProj/Projects%20n%20research%20pages/Magnetic%20Recon%20n%20thin%20cur%20sheets.htm>, 29. 4. 2013
- [23] Průběh rekonekce magnetických silokřivek, <http://www.aldebaran.cz/astrofyzika/plazma/reconnection.html>, 29. 4. 2013
- [24] Magnetická rekonekce, <http://www.aldebaran.cz/astrofyzika/plazma/reconnection.html>, 29. 4. 2013
- [25] M. Karlický, P. Jelínek, H. Mészárosová, Magnetoacoustic waves in the narrowband dm-spikes sources, *Astronomy and Astrophysics* 529, A96, 2011.
- [26] NASA, Closing in on a Hot Mystery, [http://www.nasa.gov/centers/goddard/news/topstory/2007/hinode\\_alfven\\_waves.html](http://www.nasa.gov/centers/goddard/news/topstory/2007/hinode_alfven_waves.html), 29. 4. 2013
- [27] BEDNÁŘ, V.: Vlny a oscilace v koronálních smyčkách. KAFT, PF JCU, České Budějovice, 2010
- [28] Aschwanden, M.: *Physics of Solar Corona*. Praxis Publishing Ltd, Chichester, UK, 2005.
- [29] Kulsrud, R. M.: *Plasma Physics for Astrophysics*. Princeton University Press, Princeton, New Jersey, 2005.
- [30] Chung, T. J.: *Computational Fluid Dynamics*. Cambridge University Press, NY, USA, 2002.
- [31] M. Karlický, P. Jelínek, Impulsively Generated Wave Trains in a Solar Coronal. *IEEE Trans. Plasma Sci.* 38, No. (9), 2243-2248, 2010.



## Seznam obrázků

Obr. 1: Znázornění struktury .....	9
Obr. 2: Okno příkazového řádku v OS Linux.....	17
Obr. 3: Okno programu VisIt. ....	28
Obr. 4: Spuštěné okno programu IDL.....	30
Obr. 5: Průběh rekonekce magnetických silokřivek.....	33
Obr. 6: Opačně orientovaná magnetická pole, Rekonekce magn. pole s periodicky se opakujícími body nulového pole tvaru X a O.....	34
Obr. 7: Zobrazení simulačního boxu na počátku simulace .....	35
Obr. 8: Počáteční nastavení simulačního boxu - řez simulačním boxem ve směru osy x.....	36
Obr. 9: Průběh výpočtu simulace. ....	37
Obr. 10: Zobrazení výsledků simulace CurrentSheet pomocí programu VisIt. ....	39
Obr. 11: Zobrazení výsledků simulace CurrentSheet pomocí programu VisIt.....	40
Obr. 12: Zobrazení výsledků simulace CurrentSheet pomocí programu VisIt.....	41
Obr. 13: Zobrazení výsledků pomocí programu IDL.....	41
Obr. 14: Poerupční koronální smyčky na Slunci.....	43
Obr. 15: Nákres 2D slabu v simulačním boxu a počátečních podmínek.....	44
Obr. 16: Průřez 2D slabem ve směru osy x, zobrazení průběhu hustoty.....	45
Obr. 17: Zobrazení šíření impulzně generované vlny 2D slabem .....	49
Obr. 18: Zobrazení šíření impulzně generované vlny 2D slabem .....	50
Obr. 19: Zobrazení šíření impulzně generované vlny 2D slabem .....	50
Obr. 20: Řez 2D slabem - zobrazení postupující vlny .....	51
Obr. 21: Porovnání šíření vlny slabem různých šířek. ....	52
Obr. 22: Porovnání šíření vlny slabem různých šířek .....	52
Obr. 23: Porovnání šíření vlny slabem různých šířek .....	53

## Seznam tabulek

Tab. 1: Základní fyzikální vlastnosti Slunce .....	8
Tab. 2: Přehled programů používaných pro výpočty v astrofyzice.....	16

## Příloha 1 - nastavení proměnných prostředí v souboru *BASHRC*.

```
# Intel Fortran Compiler

source /opt/intel/bin/compilervars.sh intel64

export LD_LIBRARY_PATH=/opt/intel/lib:$LD_LIBRARY_PATH

#MPICH

PATH=/opt/mpich2/bin:$PATH; export PATH

#HDF5

export LD_LIBRARY_PATH=/opt/hdf5/lib:$LD_LIBRARY_PATH

export LD_RUN_PATH=/opt/hdf5/lib:$LD_RUN_PATH

export INCLUDE=/opt/hdf5/include:

#IDL

IDL_DIR="/usr/local/itt/idl"; export IDL_DIR

XFLASH3_DIR="/home/petr/ASTRO_SIM/FLASH3.2/tools/fidlr3.0"; export

XFLASH3_DIR

IDL_PATH="${XFLASH3_DIR}:${IDL_DIR}:${IDL_DIR}/lib"; export IDL_PATH

IDL_PATH="${XFLASH3_DIR}:${IDL_PATH}"; export IDL_PATH
```

## Příloha2 - Výpis upraveného souboru Makefile.h

V tomto příkladu jsou nastaveny cesty podle umístění knihoven podle instalace popsané v této práci.

```
# FLASH makefile definitions for ix86-64 Linux (gfortran compiler)

#-----

# Set the HDF/HDF5 library paths -- these need to be updated for your system
#-----

HDF4_PATH =
HDF5_PATH = /opt/hdf5

ZLIB_PATH =

PAPI_PATH =

PAPI_FLAGS =

MPI_PATH = /opt/mpich2

MPE_PATH =

#-----

# Compiler and linker commands

#

# Use the MPICH wrappers around the compilers -- these will automatically
# load the proper libraries and include files. Version of MPICH prior
# to 1.2.2 (?) do not recognize .F90 as a valid Fortran file extension.
# You need to edit mpif90 and add .F90 to the test of filename extensions,
# or upgrade your MPICH.

#-----

FCOMP = /opt/mpich2/bin/mpif90
CCOMP = /opt/mpich2/bin/mpicc
#CPPCOMP = /usr/local/mpich-1.2.7pl/gfortran/bin/mpiCC

LINK = /opt/mpich2/bin/mpif90 -B static

# pre-processor flag

PP = -D

#-----

# Compilation flags

#
```

```

# Three sets of compilation/linking flags are defined: one for optimized
# code, one for testing, and one for debugging. The default is to use the
# _OPT version. Specifying -debug to setup will pick the _DEBUG version,
# these should enable bounds checking. Specifying _TEST is used for
# flash_test, and is set for quick code generation, and (sometimes)
# profiling. The Makefile generated by setup will assign the generic token
# (ex. FFLAGS) to the proper set of flags (ex. FFLAGS_OPT).
#-----
FFLAGS_OPT = -c -r8 -i4 -O3
FFLAGS_DEBUG = -c -r8 -i4
FFLAGS_TEST = -c -r8 -i4

F90FLAGS =

CFLAGS_OPT = -c -O2
CFLAGS_DEBUG = -c -g
CFLAGS_TEST = -c -O2

# if we are using HDF5, we need to specify the path to the include files
CFLAGS_HDF5 = -I ${HDF5_PATH}/include
CFLAGS_MPI = -I$(MPI_PATH)/include
#-----

# Linker flags
#
# There is a separate version of the linker flags for each of the _OPT,
# _DEBUG, and _TEST cases.
#-----
LFLAGS_OPT = -r8 -i4 -Vaxlib -o
LFLAGS_DEBUG = -r8 -i4 -Vaxlib -o
LFLAGS_TEST = -r8 -i4 -Vaxlib -o
#-----

# Library specific linking
#
# If a FLASH module has a 'LIBRARY xxx' line in its Config file, we need to
# create a macro in this Makefile.h for LIB_xxx, which will be added to the
# link line when FLASH is built. This allows us to switch between different

```

```

# (incompatible) libraries. We also create a _OPT, _DEBUG, and _TEST
# library macro to add any performance-minded libraries (like fast math),
# depending on how FLASH was setup.
#-----
LIB_OPT =
LIB_DEBUG =
LIB_TEST =
#LIB_HDF4 = -lmfhdf -ldf -ljpeg -lz
LIB_HDF5 = -L $(HDF5_PATH)/lib -lhdf5
LIB_PAPI =
LIB_MATH = -ldfftw -ldrfftw
LIB_MPI = -L $(MPI_PATH)/lib -lmpich -lmpichf90 -lmpe -lpthread
LIB_NCMPPI =
LIB_MPE =
#-----
# Additional machine-dependent object files
#
# Add any machine specific files here -- they will be compiled and linked
# when FLASH is built.
#-----
MACHOBJ =
#-----
# Additional commands
#-----
MV = mv -f
AR = ar -r
RM = rm -f
CD = cd
RL = ranlib
ECHO = echo

```