



**FAKULTA
STROJNÍ
ČVUT V PRAZE**

Ústav automobilů, spalovacích motorů
a kolejových vozidel

Rychlostní pole ve válci motoru – srovnání
experimentálních a simulačních výsledků

In-cylinder velocity field – comparison
of experimental and simulation results

DIPLOMOVÁ PRÁCE
2022

Bc. Filip Labuta

Studijní program: NDTT Dopravní a transportní technika
Studijní obor: N071TSPM Spalovací motory
Vedoucí práce: Ing. Petr Hatschbach, Csc.

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Labuta** Jméno: **Filip** Osobní číslo: **466695**
Fakulta/ústav: **Fakulta strojní**
Zadávací katedra/ústav: **Ústav automobilů, spalovacích motorů a kolejových vozidel**
Studijní program: **Dopravní a transportní technika**
Specializace: **Spalovací motory**

II. ÚDAJE K DIPLOMOVÉ PRÁCI

Název diplomové práce:

Rychlostní pole ve válci motoru - srovnání experimentálních a simulačních výsledků

Název diplomové práce anglicky:

In-cylinder velocity field - comparison of experimental and simulation results

Pokyny pro vypracování:

Zpracujte postup pro analýzu, porovnání a prezentaci výsledků experimentů a numerických simulací proudění ve válci motoru:

- Proveďte průzkum a zvolte vhodný jednotný analytický a prezentační softwarový nástroj, ověřte možnosti importu dat.
- Jako testovací příklad zvolte stacionární profukovací zkoušku sacích kanálů hlavy spalovacího motoru a zobrazení rychlostního pole v rovinném řezu.
- Zpracujte přehled veličin z výsledků numerické simulace proudění a z měření rychlostního pole, proveďte rozbor možností jejich porovnání.
- Použijte dostupná experimentální data z měření rychlostního pole pomocí termoanemometru a PIV.
- Pro data z numerických simulací použijte existující výsledky a proveďte vlastní simulace. Seznamte se používáním CFD programu AVL FIRE a proveďte výpočty případů odpovídajících experimentálním výsledkům.
- Vypracujte postupy, resp. skripty pro generování porovnatelných parametrů, grafů a jiných forem prezentací na základě dat z různých zdrojů s využitím zvoleného softwaru. Ověřte a ukažte jejich použitelnost na testovacích příkladech.

Seznam doporučené literatury:

Jméno a pracoviště vedoucí(ho) diplomové práce:

Ing. Petr Hatschbach, CSc. ústav automobilů, spalovacích motorů a kolejových vozidel FS

Jméno a pracoviště druhé(ho) vedoucí(ho) nebo konzultanta(ky) diplomové práce:

Datum zadání diplomové práce: **20.04.2022**

Termín odevzdání diplomové práce: **13.07.2022**

Platnost zadání diplomové práce: _____



Ing. Petr Hatschbach, CSc.
podpis vedoucí(ho) práce



doc. Ing. Oldřich Vítek, Ph.D.
podpis vedoucí(ho) ústavu/katedry




doc. Ing. Miroslav Španiel, CSc.
podpis děkana(ky)

III. PŘEVZETÍ ZADÁNÍ

Diplomant bere na vědomí, že je povinen vypracovat diplomovou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací. Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v diplomové práci.

27.4.2022
Datum převzetí zadání


Podpis studenta



Anotační záznam

- Jméno autora: Bc. Filip Labuta
- Název práce: Rychlostní pole ve válci motoru – srovnání experimentálních a simulačních výsledků
- Anglický název: In-cylinder velocity field – comparison of experimental and simulation results
- Rozsah práce: 69 stran
56 obrázků
- Akademický rok: 2021/2022
- Ústav: 12 120 Ústav automobilů, spalovacích motorů a kolejových vozidel
- Studijní program: NDTT Dopravní a transportní technika
- Vedoucí práce: Ing. Petr Hatschbach, Csc.
- Klíčová slova: Rychlostní pole ve válci motoru, Vizualizace a analýza dat, Vírové a průtokové charakteristiky, Automatizace vyhodnocení dat, Skriptování, Experiment a simulace
- Key words: In-cylinder velocity field, Experiment and simulation, Vortex and flow characteristics, Automation of data evaluation, Data visualization and analysis, Scripting



Abstrakt

Cílem této práce je tvorba vhodného srovnání rychlostních polí ve válci motoru získaných z experimentálního měření a numerických simulací. V textu jsou tyto metody popsány, jakožto i například formát vstupních dat. Práce je zaměřena zejména na automatizaci procesu vyhodnocení formou programování skriptů, které data ideálně samostatně zpracují a vyhodnotí ve formě přehledných grafů. Součástí je přehled několika vizualizačních a analyzačních softwarů, přičemž jeden z nich je vybrán a širě popsán. To znamená taktéž popis použitých funkcí nebo na co si dát pozor.

Klíčová slova:

Rychlostní pole ve válci motoru, Experiment a simulace, Vírové a průtokové charakteristiky, Automatizace vyhodnocení dat, Vizualizace a analýza dat, Skriptování

Abstract

The aim of this work is to create a suitable comparison of in-cylinder velocity fields obtained from experimental measurements and numerical simulations. These methods are described in the text, as well as, for example, the format of the input data. The work is mainly focused on the automation of the evaluation process in the form of programming scripts, which ideally process and evaluate the data independently in the form of well-arranged graphs. It includes an overview of several visualization and analysis software, one of which is selected and described in more detail. This also means a description of the functions used or what to look out for.

Key words:

In-cylinder velocity field, Experiment and simulation, Vortex and flow characteristics, Automation of data evaluation, Data visualization and analysis, Scripting



Čestné prohlášení

Prohlašuji, že jsem diplomovou práci na téma: “Rychlostní pole ve válci motoru – srovnání experimentálních a simulačních výsledků” vypracoval samostatně s použitím odborné literatury a pramenů uvedených v seznamu zdrojů, jenž tvoří poslední kapitolu této práce.

V Praze dne: 19.6.2022

.....

(podpis autora)



Poděkování

Velmi bych chtěl tímto poděkovat svému vedoucímu a konzultantovi Ing. Petru Hatschbachovi za opravdu velice vstřícný přístup, jakožto i za výborné vedení této práce, ale také za přínosné rady a poznatky. Nesmím taktéž opomenout značnou pomoc rodiny a přátel, kterým tímto moc děkuji.



Obsah

Seznam použitých zkratk.....	9
Seznam použitých symbolů a veličin.....	10
Úvod	11
1. Teoretická část	12
1.1. Testovaný objekt – Profukovací zkouška hlavy motoru.....	12
1.2. Experimentální měření.....	13
1.2.1. HWA – Měření metodou žhaveného drátku.....	13
1.2.2. PIV – Měření metodou snímkování proudících částic	14
1.3. Numerické simulace	15
1.3.1. Metoda k- ζ -f	15
1.3.2. Metoda PANS.....	15
1.4. Průtokové a vírové vlastnosti sacích a výfukových kanálů.....	16
1.4.1. Objemový a hmotnostní průtok	16
1.4.2. Průtokový součinitel	17
1.4.3. Průtokové číslo	18
1.4.4. Uzavírací číslo kanálu.....	19
1.4.5. Vírová čísla	20
1.4.5.1. Tok momentu hybnosti.....	20
1.4.5.2. Tumble a swirl číslo	21
1.5. Analyzační softwary	22
1.5.1. Excel	22
1.5.2. MATLAB.....	23
1.5.3. ParaView	24
1.5.4. Tecplot	24
1.5.5. VisIt	25
1.5.6. Gnuplot.....	26
1.5.7. NI DIAdem – vybraný software	27
1.5.8. Stručný přehled zkoumaných softwarů	28
2. Praktická část.....	29
2.1. Výstupní (surová) data.....	29
2.1.1. Výstupní data z experimentu	29
2.1.1.1. HWA měření.....	29
2.1.1.2. PIV měření.....	30
2.1.2. Výstupní data ze simulace.....	30
2.1.2.1. Doporučení před simulacemi – AVL FIRE	31
2.1.2.2. Výstupní data ze simulace v AVL FIRE	31



2.2.	Nahrání dat do DIAdemu.....	33
2.3.	Skriptování.....	34
2.3.1.	Základní import dat.....	34
2.3.2.	Přepočet surových dat na souřadnice a veličiny	37
2.3.3.	Výpočet požadovaných integrálních veličin.....	39
2.3.4.	Uspořádání dat pro přehlednost	41
2.3.5.	Vytvoření vektorů pro zobrazení rychlostních polí	42
2.3.6.	Skriptování více simulací – proměnné otevření ventilů.....	44
2.3.7.	Skriptování experimentů	45
2.4.	Vizualizace – grafické výstupy a vhodné grafy	46
2.4.1.	Základní vyobrazení rychlosti průtoku v axiálním směru	48
2.4.2.	Zobrazení izočár, umístění hodnot a max/min hodnoty	50
2.4.3.	Zobrazení vektorů rychlosti.....	52
2.4.4.	Zobrazení rozdílových map – diference grafů.....	56
2.4.5.	Spojnicový graf, tabulka a různé detaily	57
2.5.	Sestavené vzorové skripty – 3 typy	59
2.6.	Sestavené vzorové grafické výstupy.....	60
2.6.1.	Základní grafické srovnání simulací dvou hlav motorů.....	60
2.6.2.	Grafické srovnání simulací a experimentů.....	61
2.6.3.	Grafické srovnání při proměnném zdvihu ventilů	64
2.6.4.	Animace	65
2.7.	Zpracování a analýza jiného typu dat	66
3.	Závěr	67
	Seznam použité literatury.....	68
	Seznam použitých obrázků.....	69
	Seznam příloh.....	70



Seznam použitých zkratk

CFD	Computational Fluid Dynamics – Počítačová simulace proudění tekutin
ČVUT	České vysoké učení technické v Praze
DNS	Direct Navier-Stokes, v některých zdrojích také Direct Numerical Solution
EU	Evropská Unie
GIF	Graphics Interchange Format – Formát umožňující například animaci
HWA	Hot Wire Anemometry – Měření metodou žhaveného drátku
LES	Large Eddy Simulation
PDF	Portable Document Format – Přenosný formát dokumentů
PIV	Particle Image Velocimetry – Měření snímkováním proudících částic
PPT	Microsoft Power Point – Prezentační nástroj od Microsoftu
PANS	Partially Averaged Navier Stokes
RANS	Reynolds Averaged Navier Stokes
SW	Software



Seznam použitých symbolů a veličin

Značení	Popis veličiny	Jednotka
Latinská abeceda		
A	Plocha, průřez	m^2
h	Měrná entalpie	J/kg
k	Turbulentní kinetická energie	m^2/s^2
p	Tlak	Pa
r	Měrná plynová konstanta	J/kg/K
v	Rychlost (proudění)	m/s
Řecká abeceda		
κ	Poissonova konstanta	1
ρ	Hustota	kg/m^3



Úvod

Motivací této práce bylo shromáždění postupů vyhodnocení a srovnání dat získaných buďto z měření různých veličin, či během provádění numerických simulací různorodých modelů. Při práci s velkým počtem dat totiž není mnohdy žádný jednotný a ucelený postup pro jejich následné zpracování. Myšlený uživatel tak nad touto problematikou tráví opakovaně zbytečně dlouhý čas při snaze o vhodné prezentování výsledků práce na projektech, zkouškách, certifikacích, testech a podobných záležitostech.

Tato práce se tak zabývá možnostmi zpracování, vyhodnocení a následnou prezentací výsledků z dat v přehledné formě, zároveň je menší rešerší programů, které by mohly posloužit těmto účelům. Čtenář následně bude mít větší přehled o možnostech analyzačních softwarů a nebude muset spoléhat pouze na vybraný analyzační software. Ve zvoleném softwaru bude proveden průzkum možností výstupů a návodů k tomu příslušejících tak, aby byla uživateli urychlena a usnadněna práce s velkým počtem dat, či aby mohl snadno navázat na poznatky podané v následujícím textu.

Testovaným objektem, jemuž se bude následující text věnovat, je stacionární profukovací zkouška hlavy motoru, kde se sledují rychlostní pole ve válci motoru. Tato profukovací zkouška byla prováděna v laboratorních podmínkách, tedy experimentálně a následně tedy i pomocí numerické simulace. Simulace i experimenty byly prováděny za použití daných okrajových podmínek, ale o tom bude pojednáno v textu dále. Část textu je taktéž zaměřena i na postup přípravy simulace, respektive i na různé poznatky získané během práce se softwarem, zde tedy s CFD softwarem AVL FIRE.

Sledované dvě hlavy motoru (sloužící zde jako objekty pro zpracování a srovnání dat) se konstrukčně podobají. První hlava je používána na motorech (například mimo území Evropy), tedy splňující normu jinou než tu z EU. Druhá hlava slouží na motorech, které zase naopak splňují emisní normy platné v EU, příkladem může být již starší norma EURO 5. Detailnější popis těchto hlav motorů bude popsán v textu dále.

Při zpracovávání dat je také vyžadováno dosažení ideálně plné automatizace procesu vyhodnocení dat, a to pokud možno bez nutnosti zásahu uživatele. To by znamenalo, že daný software si data sám načte, přepočte, či dopočte, vyhodnotí a následně vyexportuje výstup například ve formě PDF či PPT.



1. Teoretická část

V teoretické části této práce bude nahlíženo na obecný pohled a popis problematiky, již se daný text zabývá. To znamená uvedení čtenáře (pokud možno nenásilnou formou) do toho, co je zde předmětem srovnávání, až po analýzu výsledných dat ve formě například jednoduchých a přehledných grafů, či obrázků a tabulek.

1.1. Testovaný objekt – Profukovací zkouška hlavy motoru

Sledovaným objektem, na němž jsou testovány možnosti prezentace výsledných dat v přijatelné formě, je profukovací zkouška hlavy motoru. V této zkoušce se vyhodnocují rychlostní pole, jak vychází z názvu této práce. Z rychlostních polí jsou dále odvozené průtokové vlastnosti jako hmotnostní průtok, moment hybnosti, průtokový součinitel či vírové vlastnosti, jež jsou zde určeny z předem daného průřezu kolmého k ose válce.

Na obrázku 1 jsou vyobrazeny tři výpočetní sítě, a to dvě z motoru s přímým vstřikem paliva (červené sací kanálky – dvě zleva) a nepřímým vstřikem paliva (modré sací kanálky – vpravo), jež jsou vyobrazeny v programu FIRE. Jsou zde použity 2 stejné sítě lišící se však ve velikostech elementů výpočetní sítě pro další možnost porovnání výsledků při prezentaci. Větší síť nepřímého vstřiku zde byla zmenšena ve FIRE na rozměry menší sítě přímého vstřiku. Rozměry všech třech sítí jsou tak zde dány vrtáním $D = 74,5$ mm.



Obrázek 1: Pohled na sítě z motorů s přímým a nepřímým vstřikem paliva

Experimentální data (tedy ta skutečně naměřená v laboratorních podmínkách na měřicím stanovišti) byla naměřena metodami HWA a PIV. Metodou HWA bylo měřeno na hlavě



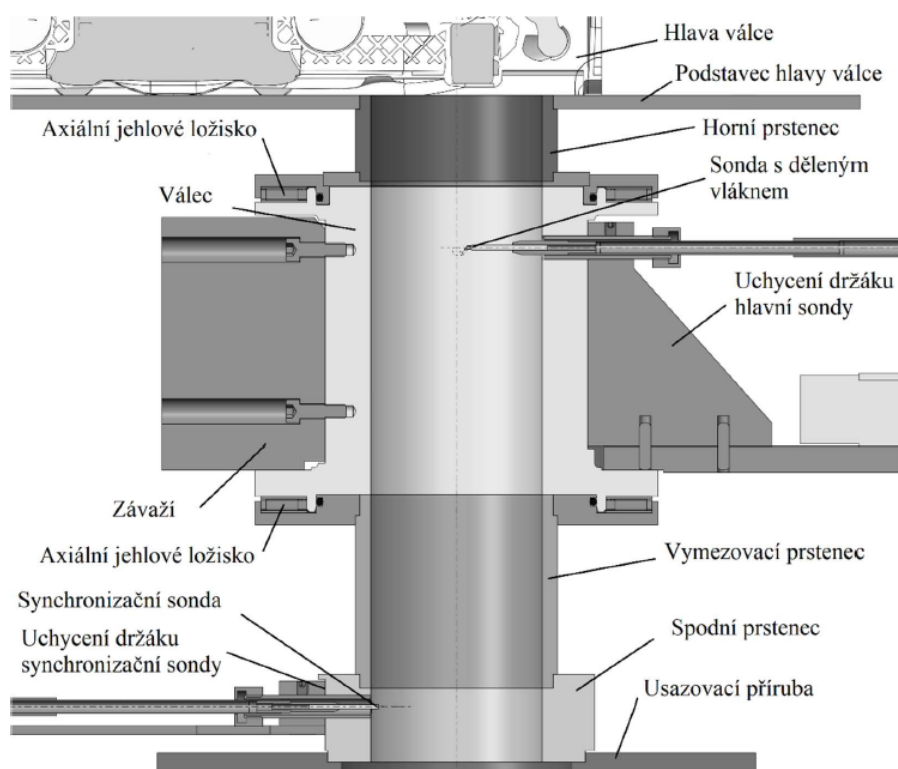
s přímým vstřikem a metodou PIV naopak průtok hlavy s nepřímým vstřikem. V následujících kapitolách budou podrobněji popsány stanoviště a modely o kterých celá tato práce pojednává.

1.2. Experimentální měření

Měření rychlostního pole při profukování hlav motorů bylo prováděno dvěma metodami: HWA a PIV. Profukování probíhalo na již uvedených hlavách motorů s přímým a nepřímým vstřikem paliva. Měřená rovina je dána do stejné poměrové vzdálenosti od roviny mezi hlavou motoru a válcem motoru, tedy měřicí rovina je ve vzdálenosti $0,6D$ od hlavy.

1.2.1. HWA – Měření metodou žhaveného drátku

První zmíněnou experimentální metodou, při níž byla měřena rychlostní pole ve válci motoru, byla metoda HWA prováděná v tomto případě na hlavě motoru s přímým vstřikem. Tato metoda se jeví jako poměrně časově náročná, protože je nutné traverzování sondou po průměru. Jednodušeji řečeno se sonda pootáčí a posouvá po průměru válce tak, aby bylo zachyceno proudění ve více bodech plochy myšleného průřezu (viz obrázek 2).



Obrázek 2: Řez měřicím stanovištěm – HWA [1]

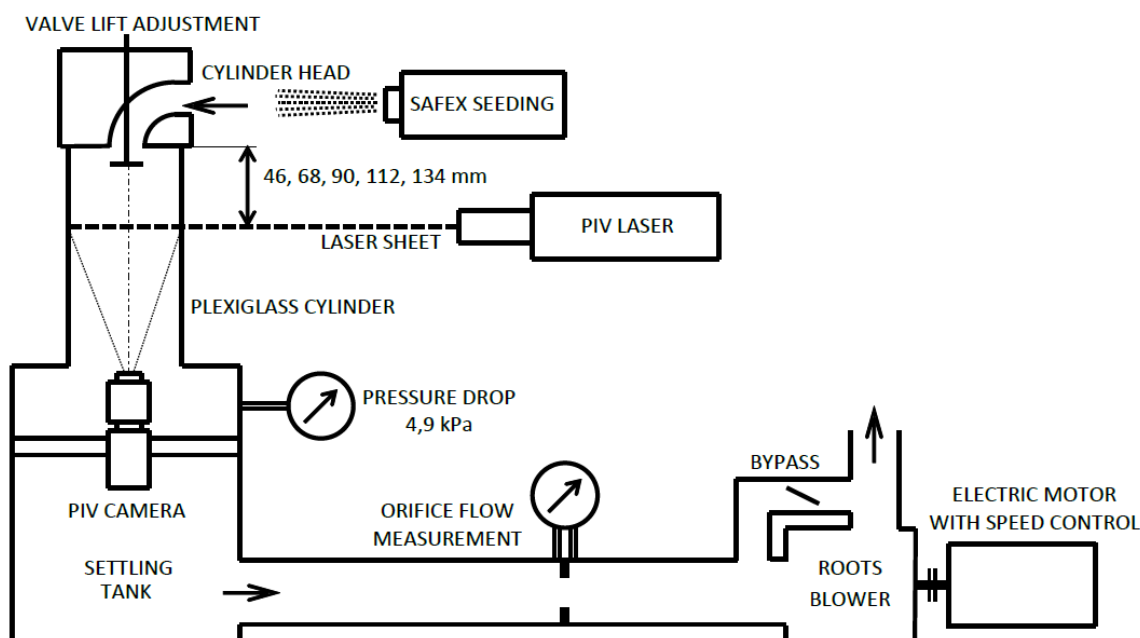
Problematické se tak zdá hodnocení pravdivosti výsledných údajů o rychlostních polích, jelikož zde může docházet k velkému ovlivnění měření pohybem sondy, či samotnou přítomností sondy. Ostatně díky turbulentnímu proudění je často nutný jistý počet opakování,



aby se minimalizovala chyba měření a časová náročnost tudíž nepříjemně stoupá. Výsledky tím pádem nejsou zcela příznivé, jak bude možné zpozorovat ve výsledných datech dále. Sonda zde použitá má dva na sobě nezávislé tenké niklové filmy, díky nimž zvládne měřit dvě složky rychlosti v třírozměrném proudění. Jakmile totiž dojde k natočení proudu, dochází k rozdílu v napětí na těchto filmech, což se projeví změnou průtočné hustoty v daném směru. [1]

1.2.2. PIV – Měření metodou snímkování proudících částic

Druhou experimentální metodou je měření za pomoci snímkování proudících částic, přičemž je snímán pohyb částic (změna polohy v čase), tudíž je vyhodnocován vektor rychlosti proudění každé částice. Měření touto metodou bylo prováděno na hlavě motoru s nepřímým vstříkem paliva.



Obrázek 3: Schéma měřicího stanoviště PIV [16]

Výhodou této metody je značná přesnost bez vlivů sondy na rozdíl od metody HWA, kde je sonda přítomna v měřeném průřezu. Jinými slovy se jedná o bezkontaktní měření bez vlivů měřicího zařízení. Taktéž je zajištěna dobrá opakovatelnost, zejména díky snímání celého průřezu najednou, ale také vysokorychlostnímu snímkování celé plochy.

Nevýhodou však může být, že nelze zachytit složku rychlosti, která směřuje přímo ke kameře, zde tedy axiální složku ve směru osy z . Bez této složky se poté nedá určit objemový průtok a veličiny z něho odvozené. Dají se zde tak pozorovat jevy kolmé ke kameře, jako je například swirl. [16]



1.3. Numerické simulace

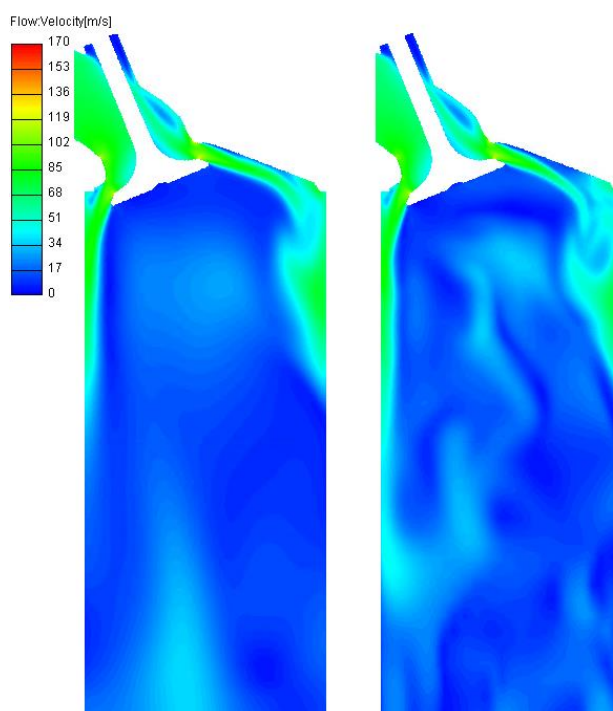
Další sadou dat pro srovnávání (jak už z názvu práce vyplývá) jsou hodnoty z rychlostních polí ve válci motoru získané pomocí numerických simulací (zde provedených v CFD programu AVL FIRE). Pro výpočet byly zadány stejné podmínky jako při experimentu, čímž jsou myšleny charakteristiky týkající se například tlaku, či teploty. Pro srovnávání bylo nadále využito dvou odlišných výpočtových metod k - ζ - f a PANS, jenž budou dále popsány podle příslušného softwaru.

1.3.1. Metoda k - ζ - f

Tato metoda spadající do rodiny RANS, byla vyvinuta v roce 2004 jako modifikace vířivo-viskozitního modelu založeného na eliptickém relaxačním modelu z roku 1991. Tato „defaultní“ metoda v softwaru AVL FIRE ve výsledku dosahuje lepší numerické stability. V názvu k označuje turbulentní kinetickou energii, ζ normalizovanou velikost rychlosti a f eliptickou relaxační funkci. [2] [4] [7] [11] [17]

1.3.2. Metoda PANS

Metoda PANS je hybridní metoda, jenž byla navržena roku 2006. Využívá plynulé změny výpočetního přístupu od RANS po DNS v závislosti na velikosti elementů dané sítě (respektive jemnosti sítě). Hrubá síť využívá metody RANS, kdežto zjemňováním plynule dochází k přechodu na model LES a při výrazném zjemnění sítě až na metodu DNS.



Obrázek 4: Porovnání metod k - ζ - f a PANS na hlavě s nepřímým vstřikováním paliva – rychlostní pole v m/s



Na obrázku 4 je vidět, že model k- ζ -f (na rozdíl od PANS) nezachycuje tak dobře proudění dál od stěny a hlavní proud se více tiskne ke stěně válce.

Turbulentní model PANS je však schopen pracovat pouze ve stacionárních režimech, kdy nedochází ke změnám geometrie, či okrajových podmínek. Přístup PANS je ještě stále předmětem výzkumu a spolupráce s výzkumnými týmy na univerzitách. [2] [7] [9] [11] [13] [17]

1.4. Průtokové a vírové vlastnosti sacích a výfukových kanálů

Tato kapitola bude zaměřena na veličiny, které jsou odvozeny zejména z rychlostních polí ve válci motoru, a na vlastnosti odvozené například z hustoty proudící tekutiny, která je snadno k dispozici u numerických simulací. U experimentálních měření není možnost změřit lokální hustotu, a tak je potřeba brát konstantní hodnotu v celém průřezu dle daných podmínek.

1.4.1. Objemový a hmotnostní průtok

Výchozím pozorovaným parametrem je průtok, jenž je brán podle základních zákonů termodynamiky o zachování hmoty. Podle tohoto zákona je známo, že časová derivace hmotnosti je rovna nule. To jinými slovy znamená, že daná hmota směsi sice mění svou podobu (například během procesu spalování), ale ve výsledku bude hmota vstupující do systému rovna hmotě výstupní.

Daná hmota jednoduše nevyrchá do neznáma. Hmotnostní tok, jenž vychází z rovnice kontinuity, je dán následovně:

$$\dot{m}_1 = \dot{m}_2 = \rho_1 \cdot A_1 \cdot v_1 = \rho_2 \cdot A_2 \cdot v_2 = konst. \quad (1)$$

$$\dot{V}_1 = \dot{V}_2 = A_1 \cdot v_1 = A_2 \cdot v_2 = konst. \quad (2)$$

Objemový tok dle rovnice 2 je tedy hmotnostním tokem z rovnice 1, kdy je předpokládána neměnná hustota. Z toho vychází, že v průřezu sacích ventilů bude větší rychlost než v průřezu umístěném ve volném prostoru válce. Tento průřez bude dán kruhovou plochou o průměru vrtání válce.

Skutečné proudění však bude ovlivněno například ztrátami způsobenými škrcením, ale to již budou řešit parametry v následujících kapitolách, které tyto jevy sledují. [6]



1.4.2. Průtokový součinitel

Průtokový součinitel je bezrozměrné číslo, které je definované jako poměr skutečného a teoretického (teoreticky možného, bezztrátového) hmotnostního průtoku. Skutečný hmotnostní průtok je změřen experimentálně, či je určen z výsledků numerické simulace.

Teoretický hmotnostní průtok je dán stavovými parametry proudícího média (tlak, teplota, hustota), tlakovým spádem a průtočným průřezem. Referenční průtočný průřez by měl být ten nejmenší, ale při proudění kanály a mezerou mezi ventilem a sedlem jeho určení nemusí být jednoznačné.

V praxi se používají dvě definice, dva způsoby určení referenčního průtočného průřezu a podle toho je pak výsledkem buď průtokový součinitel nebo průtokové číslo.

Prvním zmíněným je tedy průtokový součinitel, jenž je vztahován k mezeře mezi okrajem ventilu a ventilovým sedlem, čímž vzniká válcová plocha (viz obrázek 5 dále). Velikost této válcové plochy je tak úměrná zdvihu ventilu. K této geometrické ploše je následně přiřazen hmotnostní průtok.

$$S_V = \pi \cdot D_V \cdot L_V \quad (3)$$

Geometrická plocha S_V , k níž je průtok vztažen, je dána dle rovnice 3, kde D_V je vnitřní průměr ventilového sedla a L_V značí zdvih ventilu. Vztah pro průtokový součinitel je dán následovně:

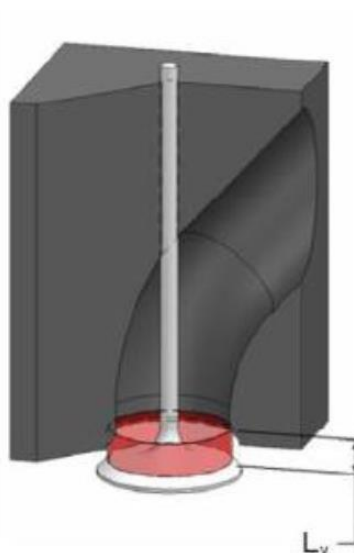
$$C_d = \frac{\dot{m}_{skut}}{\dot{m}_{d\ teor}} = \frac{\dot{m}_{skut}}{N_V \cdot \rho \cdot S_V \cdot v_2} \quad (4)$$

N_V je počet sacích ventilů a v_2 uvažovaná rychlost, která je odvozena z první termodynamické věty, tedy včetně uvažování stlačitelnosti tekutiny. Konkrétně se bude jednat o jednoduchý tvar pro adiabatické proudění bez přívodu technické energie.

$$h_1 + \frac{v_1^2}{2} = h_2 + \frac{v_2^2}{2} \quad (5)$$

Jedná se tedy o součet entalpie a kinetické energie plynu, přičemž je uvažována nulová vtoková rychlost, tedy rychlost v_1 . Entalpie ideálního plynu je zde dána součinem měrné tepelné kapacity při stálém tlaku c_p a teploty T v Kelvinech:

$$h = c_p \cdot T \quad (6)$$



Obrázek 5: Odvozená geometrická plocha průtokového součinitele [5]

Měrná tepelná kapacita je poté dána vztahem:

$$c_p = \frac{\kappa \cdot r}{\kappa - 1} \quad (7)$$

Adiabatickou (izoentropickou) změnu lze popsat Poissonovým vztahem:

$$\frac{T_2}{T_1} = \left(\frac{p_2}{p_1}\right)^{\frac{\kappa-1}{\kappa}} \quad (8)$$

Po dosazení těchto vztahů do rovnice 5 lze získat Saint Vénantovu-Wantzelovu rovnici, (tedy vztah pro rychlost v_2), který uvažuje stlačitelnost ideálního plynu: [4] [5] [6]

$$v_2 = \sqrt{\frac{2 \cdot \kappa}{\kappa - 1} \cdot r \cdot T_1 \cdot \left(1 - \left(\frac{p_2}{p_1}\right)^{\frac{\kappa-1}{\kappa}}\right)} = \sqrt{\frac{2 \cdot \kappa}{\kappa - 1} \cdot \frac{p_1}{\rho} \cdot \left(1 - \left(\frac{p_2}{p_1}\right)^{\frac{\kappa-1}{\kappa}}\right)} \quad (9)$$

1.4.3. Průtokové číslo

Průtokové číslo je velice podobné svou definicí průtokovému součiniteli s tím rozdílem, že uvažovaný referenční průřez teoretického hmotnostního průtoku je odlišný.

Průřez se zde odvíjí od kruhové plochy kanálu, tudíž nezávisle na zdvihu ventilu. Jedná se o plochu kruhu (viz obrázek 6 dále) o již uvedeném vnitřním průměru ventilového sedla D_V bez uvažování dříku ventilu, tedy následovně:

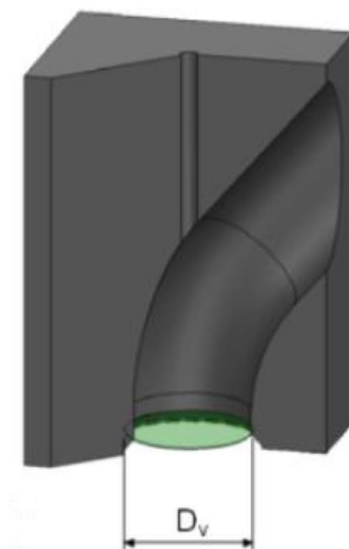
$$S_p = \frac{\pi \cdot D_V^2}{4} \quad (10)$$



Výsledný poměr (tedy průtokové číslo) je dán následovně:

$$C_f = \frac{\dot{m}_{skut}}{\dot{m}_{f teor}} = \frac{\dot{m}_{skut}}{N_V \cdot \rho \cdot S_P \cdot v_2} \quad (11)$$

Výstupní rychlost v_2 je zde odvozena stejně jako v případě rovnice 9 pro průtokový součinitel. [5] [6]



Obrázek 6: Uvažovaný průřez kanálu bez dráku ventilu [5]

1.4.4. Uzavírací číslo kanálu

Uzavírací číslo kanálu je jednoznačný vztah mezi průtokovým součinitelem a průtokovým číslem znázorňující odlišnost v uvažovaných průřezech teoretického průtoku. Jedná se o poměr zdvihu ventilu ke vnitřnímu průměru ventilového sedla.

$$\sigma = \frac{C_f}{C_d} = \frac{S_V}{S_P} = \frac{4 \cdot L_V}{D_V} \quad (12)$$

Průtokový součinitel se liší zejména svým uvažovaným průřezem teoretického hmotnostního toku, který je definován zdvihem ventilu. Průtokové číslo má tuto plochu danou neměnným průřezem kanálu.

Tím, jak se mění zdvihy ventilů, se značně odlišují i tyto součinitelé a o to větší důraz je tak kladen při analýze na oba parametry. [5] [6]

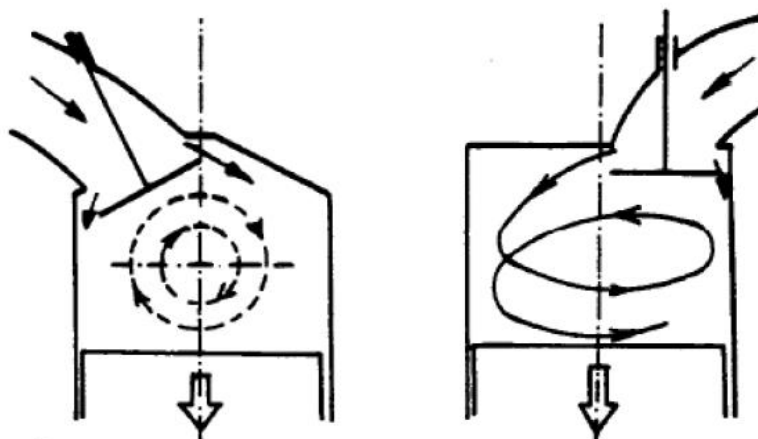


1.4.5. Vírová čísla

Vírová čísla, která zohledňují vírové vlastnosti ve válci, popisují tvorbu užitečných turbulentních efektů. Díky turbulentnímu proudění dochází k intenzivnějšímu mísení tekutin, ale také například k rychlejšímu prohoření směsi, které by jinak trvalo neúměrně dlouhou dobu.

Základními vírovými pohyby ve válci, jež projdou dále v textu analýzou, jsou tumble a swirl. V případě tumble se jedná o vír s osou rotace kolmou k ose válce, na jehož průběh má značný vliv umístění kanálu v hlavě motoru. Tumble je běžnou součástí u střešového tvaru hlavy motoru.

V případě swirl se jedná o tečnou rotaci náplně s osou rotace válce motoru. Tohoto jevu je kupříkladu dosaženo vyosením kanálu oproti ose válce, čímž vzniká moment hybnosti. Dobře je tento jev pozorovatelný v případě dvou sacích ventilů, kdy jeden ventil zůstane uzavřen. Vliv na swirl má také tvar ventilu či ventilového sedla. [2] [5] [6]



Obrázek 7: Tumble a swirl [5]

1.4.5.1. Tok momentu hybnosti

Vířivý proud je možné charakterizovat momentem hybnosti náplně ve válci. Při stacionární profukovací zkoušce, na kterou je zaměřena analýza v této práci, je možné naměřit tok momentu hybnosti v daném průřezu válce. Tento tok je dán následovně:

$$\dot{B} = \int (\vec{r} \times \rho \vec{u}) \vec{u} \cdot dA \quad (13)$$

Kde \vec{r} je poziční vektor (myšlen jako kolmá vzdálenost od vektoru k ose) a \vec{u} je vektor rychlosti. Měřené místo, kde je uvažována plocha, je průřezem kolmým k ose z o průměru rovnému vrtání válce D . Podle metodiky ČVUT je posun v ose z od roviny hlavy válce roven $0,6D$ pro vyhodnocení tumble (pro vyhodnocení swirl je tato vzdálenost $1,75D$). [6] [10]



1.4.5.2. Tumble a swirl číslo

Pro tumble je tok momentu hybnosti $\dot{\vec{B}}$ charakterizován spojením složek x a y následovně:

$$\dot{B}_x = \int y \rho u_z^2 dA \quad \dot{B}_y = - \int x \rho u_z^2 dA \quad (14)$$

$$\dot{B}_T = \sqrt{\dot{B}_x^2 + \dot{B}_y^2} \quad (15)$$

Pro swirl je tok momentu hybnosti $\dot{\vec{B}}$ charakterizován pomocí složky z následovně:

$$\dot{B}_S = \dot{B}_z = \int (x \cdot u_y - y \cdot u_x) \rho u_z dA \quad (16)$$

Kde u_x , u_y a u_z jsou vektory rychlosti v jednotlivých osách, x a y jsou kolmé vzdálenosti od vektorů rychlosti vztahované k osám. Tok momentu hybnosti \dot{B}_T či \dot{B}_S lze získat pomocí měření či z CFD simulací. V případě této analýzy budou integrály počítány numericky jako sumy přes danou plochu.

Při uvažování rotace náplně ve válci (jakožto pevného tělesa) lze vyjádřit tok momentu hybnosti takto:

$$\dot{B} = \dot{I} \cdot \omega = \dot{m} \cdot r_I^2 \cdot \omega \quad (17)$$

Kde \dot{I} je moment setrvačnosti toku, ω je úhlová rychlost pevného tělesa a r_I je poloměr otáčení (pro kruhový válec nebo disk je dán jako $r_I^2 = D^2/8$). Úhlové rychlosti náplně ekvivalentní pevnému tělesu jsou pro tumble a swirl dány:

$$\omega_T = \frac{8 \cdot \dot{B}_T}{\dot{m} \cdot D^2} \quad \omega_S = \frac{8 \cdot \dot{B}_S}{\dot{m} \cdot D^2} \quad (18)$$

Bezrozměrné vírové parametry jsou definovány jako poměr mezi úhlovou rychlostí víru a fiktivní úhlovou rychlostí motoru ω_e . Tuto fiktivní úhlovou rychlost lze odvodit z rovnice pro střední pístovou rychlost $u_{z \text{ mean}}$ po menších modifikacích:

$$u_{z \text{ mean}} = 2Zn_e = \frac{\dot{m}}{\rho \frac{\pi D^2}{4}} \quad (19)$$

$$\omega_e = 2\pi n_e = \frac{4 \cdot \dot{m}}{\rho \cdot D^2 \cdot Z} \quad (20)$$

Kde n_e jsou otáčky klikového hřídele a Z je zdvih.



Po přepsání rovnic 18 a 20 lze odvodit tumble a swirl čísla následovně:

$$TN = \frac{\omega_T}{\omega_e} = \frac{2 \cdot Z \cdot \rho \cdot \dot{B}_T}{\dot{m}^2} \quad (21)$$

$$SN = \frac{\omega_S}{\omega_e} = \frac{2 \cdot Z \cdot \rho \cdot \dot{B}_S}{\dot{m}^2} \quad (22)$$

Pro důslednější porovnání výsledků mezi různými motory z hlediska jejich konstrukce je zavedeno redukované tumble a swirl číslo:

$$TN_r = TN \cdot \frac{D}{Z} = \frac{2 \cdot D \cdot \rho \cdot \dot{B}_T}{\dot{m}^2} \quad (23)$$

$$SN_r = SN \cdot \frac{D}{Z} = \frac{2 \cdot D \cdot \rho \cdot \dot{B}_S}{\dot{m}^2} \quad (24)$$

Redukce těchto čísel je definována pro případ „čtvercového“ motoru, kdy je vrtání rovno zdvihu. Redukovaná čísla tak nemají závislost na zdvihu, která je u statické profukovací zkoušky zavádějící. [6] [10]

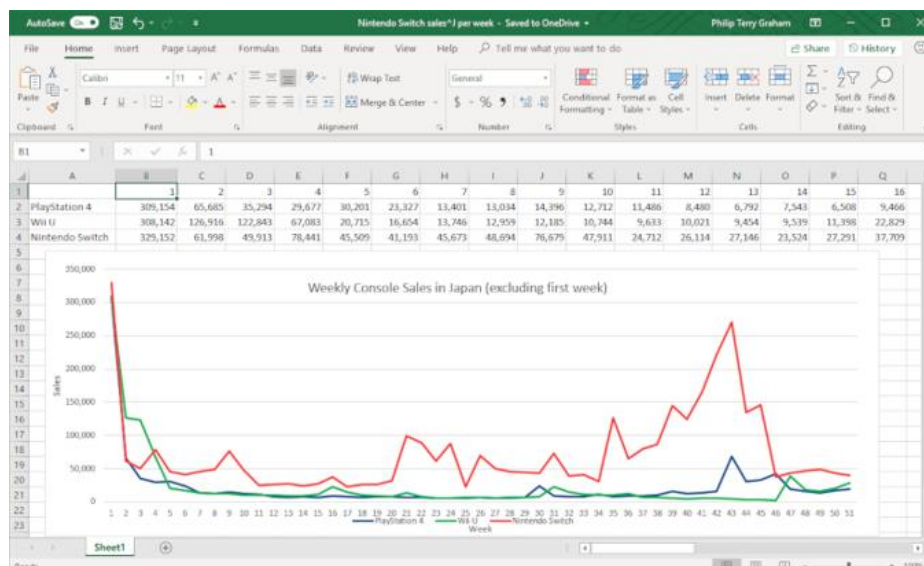
1.5. Analyzační softwary

V této kapitole je uvedena v přehledu kratší rešerše několika programů, jež slouží k analýze a následné vizualizaci surových dat. Je zde stručný přehled vybraných programů, ale těchto podobných existuje celá řada. Nutno podotknout, že je potřeba vybírat i podle formátu výstupních dat, která je nutno vyhodnotit. V této práci se však jedná o jednoduchá data s příponou *.dat*, *.txt* či například *.csv* – comma separated values.

1.5.1. Excel

Prvním a asi tím nejnámějším programem je Microsoft Excel (neboli zkráceně Excel). Jedná se v podstatě o nejrozšířenější tabulkový procesor na trhu, jenž je dostupný na několika platformách od počítačů až po chytré mobilní telefony. Excel je totiž součástí balíčku Microsoft Office, jenž zahrnuje i další nástroje jako je Word či Powerpoint.

Excel je velice uživatelsky přívětivý software s velikou podporou, s možnostmi organizace dat a s uplatněním matematických, logických, vyhledávacích, odkazovacích, textových či převodních funkcí (například ve formě nepřímého odkazu). Výstup lze následně propojit s dokumenty Word, či rovnou prezentovat ve formě jednoduchých dat či grafů. Značnou výhodou je česká lokalizace. Software je placený, ale jak již bylo zmíněno, tak součástí balíčku je hned několik nástrojů, což určitě přidává na hodnotě.

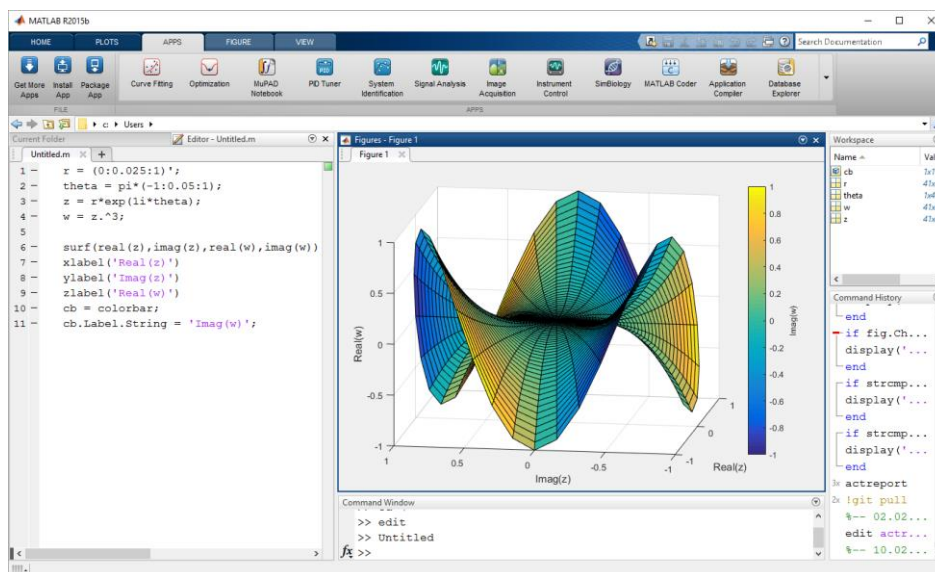


Obrázek 8: Pohled na prostředí Excel [14]

Problém však nastává s velkým počtem dat, kdy excel už je pomalý a náročný na výkon. Možností jsou zde také často jenom jednoduché grafy ve 2D formátu. Je využíván v této práci, zejména kvůli problémům s nejednotným formátováním výstupních dat u experimentálního měření, a tudíž musí být data ještě před vyhodnocením upravena. [14]

1.5.2. MATLAB

Druhým softwarem je MATLAB, jenž je známý například na univerzitách. Program slouží jako velice schopný programovací nástroj i pro tvoření komplexních aplikací. MATLAB vyžaduje dobrou znalost tohoto programovacího jazyka, jenž je vyvíjen v C, C++, Fortran či Javě. Náročnost je zde tedy poměrně vysoká s uživatelsky nepřívětivým prostředím bez českého jazyka.



Obrázek 9: Pohled na prostředí MATLAB [14]



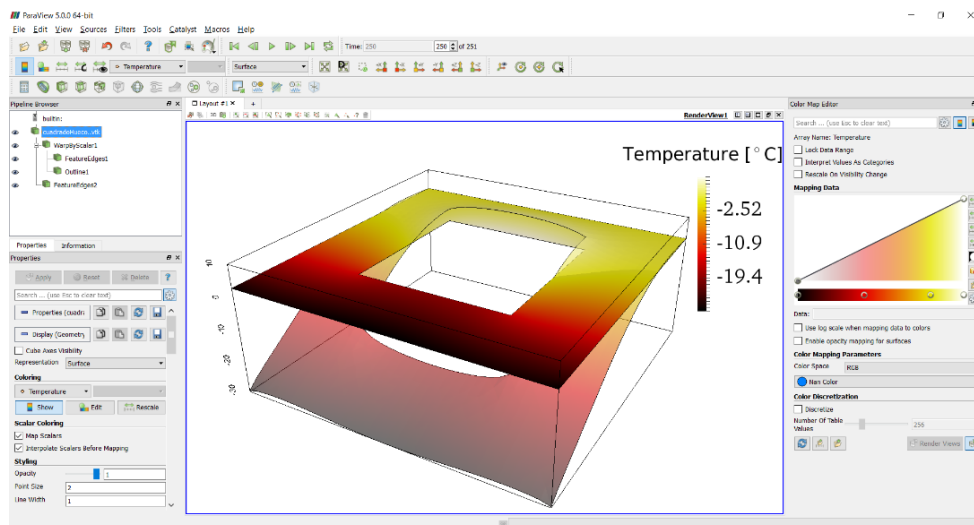
Výhodami jsou velice dobré analyzační funkce, možnosti skriptování, vykreslování velice povedených 2D či 3D grafů, možnosti nahlédnutí do knihoven či vzorů a rozsáhlá uživatelská komunita či podpora.

Nevýhodami však mohou být větší pořizovací náklady na software, velikost softwaru na disku a již zmíněná náročnost programu pro uživatele. [14]

1.5.3. ParaView

Tento program není placený. Je dostupný pod BSD licenci, která vyžaduje například uvedení autora a informace o licenci. Je zde umožněna práce s velkým počtem dat interaktivní 3D formou či programováním pomocí skriptů. Software je poměrně rozšířen na univerzitách, v laboratořích či průmyslu, byl taktéž oceněn za vysokou rychlost zpracování dat.

Programování je na bázi jazyků C, C++, Fortran a Python, který je v této aplikaci využíván zejména. Uživatelské prostředí může být pro někoho složitější jako i samotná instalace programu. Obecně se jedná spíše o složitější software pro vědecké účely. Výhodami jsou pokročilé možnosti analyzačních a vizualizačních nástrojů. [14] [15]



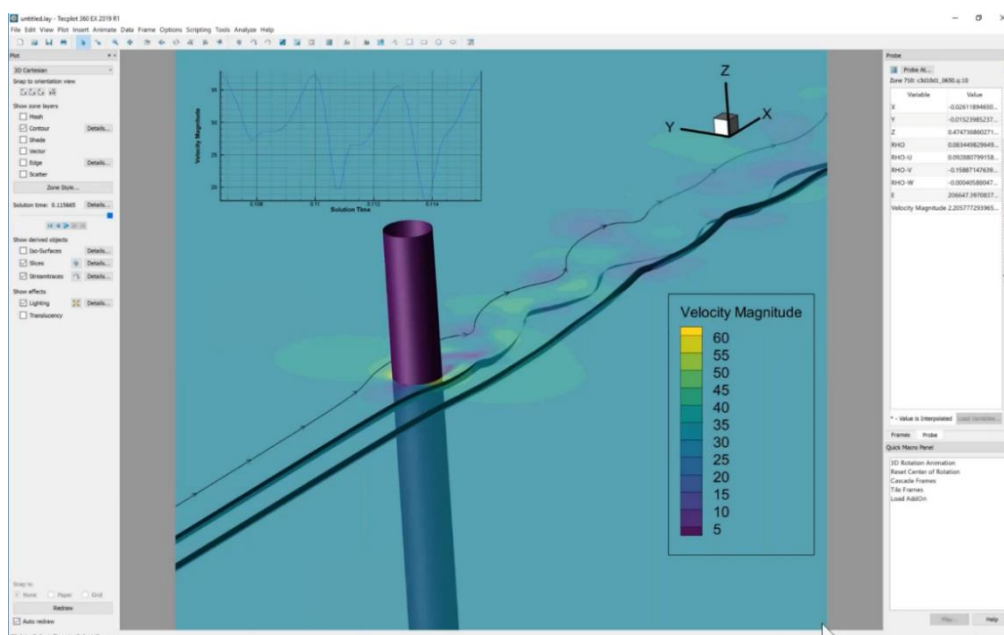
Obrázek 10: Pohled na prostředí ParaView [14]

1.5.4. Tecplot

Dalším vizualizačním a analyzačním SW je placený Tecplot, o němž tvůrci mluví jako o jednoduše použitelném SW s velkými možnostmi vizualizace. Zajímavostí je vylepšená kvalita obrázků 2D či 3D díky možnostem renderování. Dokáže pracovat s velkým množstvím dat, přičemž je možné vyobrazovat řezy, izočáry, proudnice, či vizualizovat data pomocí animací. Výhoda se jeví v automatickém rozklíčování vírových jader.



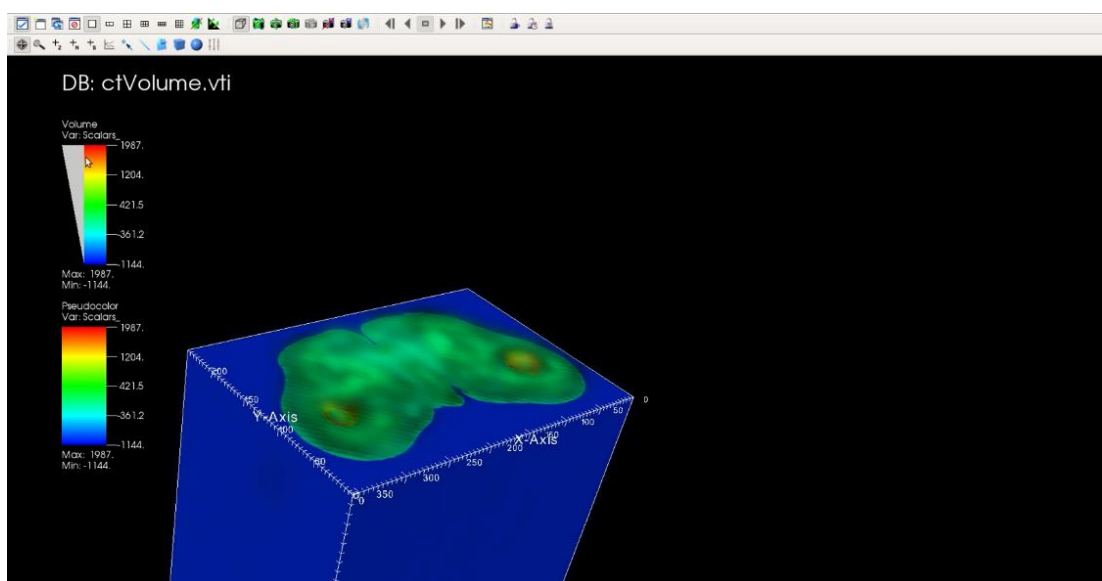
Práce v softwaru může být komplikovaná díky složitým příkazům a kódování, jenž je vyvíjeno v Pythonu. Přesto se v uživatelském prostředí celkem dobře orientuje. Vyobrazení je poté pomocí jazyka Fortran. [14] [15]



Obrázek 11: Pohled na prostředí Tecplot [15]

1.5.5. VisIt

VisIt je z řady open source programů, opět dostupný pod BSD licenci jako ParaView. Umožňuje také analýzu a vizualizaci dat ve 2D či 3D formátech (a to i pomocí animací či renderování) pro vysokou kvalitu zobrazování. Součástí programu je velká komunita, ve které jsou uvedeny různé příklady na něž se dá navázat, či je rovnou využít.



Obrázek 12: Pohled na prostředí VisIt [15]



Program je vyvíjen v programovacích jazycích C, C++, Python a Java, pričomž v Python se odehráva skriptování (Ize tak podchytiť chod programu).

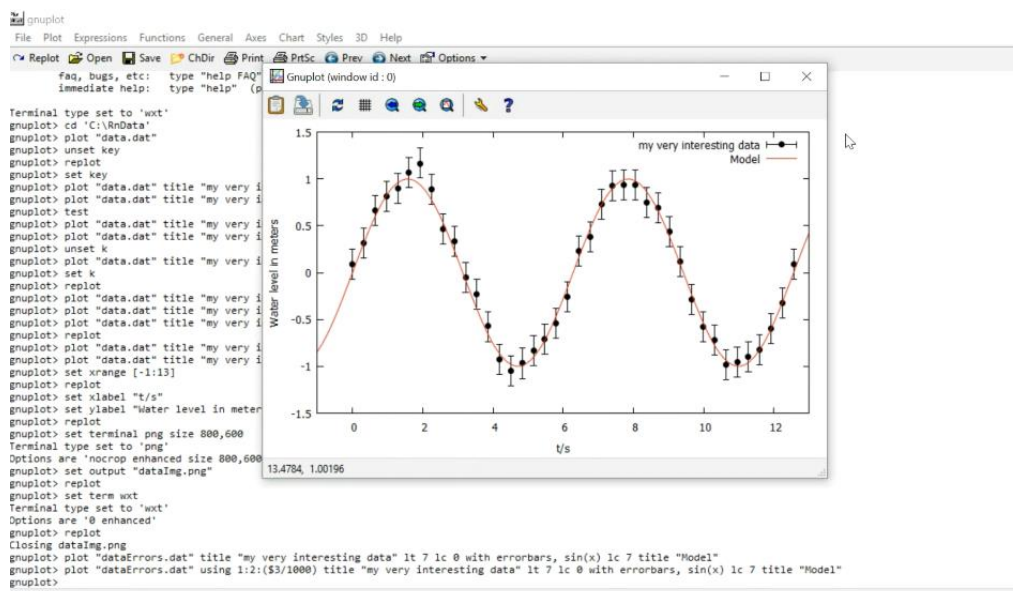
Všetchny plug-iny ve formě grafických prostředí či databází jsou nahrávány dynamicky během práce s daty, a tudíž je zaručena rychlejší odezva SW. Ostatně další plug-iny je možné nainstalovat do programu dle potřeby, kde pomocí grafického nástroje je možnost si vytvořit své vlastní plug-iny. [14] [15]

1.5.6. Gnuplot

Zde se tentokrátě jedná o volně dostupný grafický nástroj, pričomž uživatel musí dodržovat autorská práva daná na stránkách SW. Gnuplot je řízen programovacím řádkem, a tudíž je vyžadována znalost v příkazech a kódování, které vychází z jazyků C, C++, Lua, ale také Python či Java.

Gnuplot slouží například v Octave, což je opět volně dostupný software na provádění číselných výpočtů a je do určité míry kompatibilní s MATLABem. Ostatně prostředí Gnuplot softwaru se podobá MATLABu.

Možností je skriptování, jedná se však o základní programovací nástroj. K dispozici je mnoho dokumentací, tutoriálů a návodů často i v českém jazyce. [14] [15]



Obrázek 13: Pohled na prostředí Gnuplot [15]



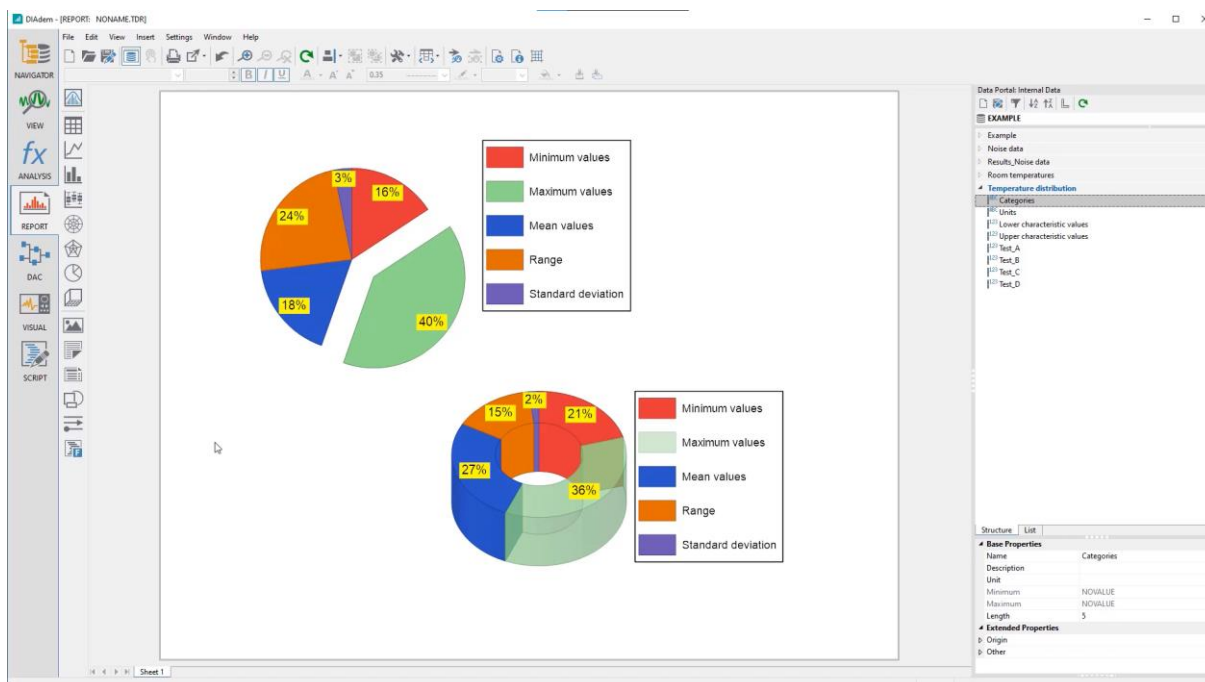
1.5.7. NI DIAdem – vybraný software

Vybraným softwarem je DIAdem od NI – National Instruments, což je placený, avšak velice schopný a sofistikovaný analyzační a vizualizační program. Program je velice efektivní, a přitom jednoduchý na práci se širokou škálou funkcí.

Například import dat různých formátů je zde podpořen díky možnosti vytvoření plug-inu pro specifické datové soubory. Software tak například automaticky rozliší, co jsou hodnoty a co je název těchto hodnot. Skriptování je zde následně v jazycích Python či Visual Basic Script pro automatizaci procesu vyhodnocení dat.

Značnou výhodou je možnost nahrávání práce uživatele v programu, jež jí následně přeneše do spustitelného skriptu. Není tak požadována vysoká znalost programovacího jazyka. Spoluprací s LabVIEW lze dosáhnout ještě větší automatizace, kdy prakticky stačí pouze měřit a programy kooperací mezi sebou rovnou vyexportují požadovaný výstup dle požadavku uživatele. Součástí DIAdemu je mnoho tutoriálů, návodů, vzorových příkladů či velká komunita.

Nevýhodou však může být absence češtiny či nemožnost nahrání všech prací, které uživatel provádí v programu. Problémem může také být počítání mezi různými skupinami či soubory, kdy v podstatě jedna skupina vychází z jednoho nahraného souboru. [14] [15]



Obrázek 14: Pohled na prostředí DIAdem [15]



1.5.8. Stručný přehled zkoumaných softwarů

V tabulce níže je shrnutí zkoumaných softwarů objektivním, ale i subjektivním pohledem různých uživatelů. Na jiné uživatele tak může nějaký z těchto softwarů působit odlišně, než je zde uvedeno. Obecně je však pohled na tyto softwary brán optikou člověka se strojním vzděláním. [14] [15]

Název	Výhody	Nevýhody	Placený?
Excel	Součást Office, jednoduchost, velká komunita, dominance na trhu, je na mnoha platformách	Problém s velkým počtem dat	ano
MATLAB	Velice schopný nástroj, možnost tvorby komplexních aplikací	Velikost SW, značná složitost	ano
ParaView	Interaktivní 3D či programová práce s daty	Složitější SW	ne
Tecplot	Uživatelské prostředí, renderování, automatické rozklíčování vírových jader	Složitě příkazy a kódy pro vyhodnocování	ano
VisIt	Tvoření plug-inů, renderování	Složitost programovacích jazyků	ne
Gnuplot	Basic programovací nástroj s velkou podporou	Podoba v programování s MATLAB – složitost	ne
DIAdem	Jednoduchý a efektivní, možnost nahrávání skriptu – automatizace procesu, podpora	Není čeština, vše do skriptu nelze nahrát, problémová práce mezi skupinami a více soubory	ano



2. Praktická část

V praktické části bude proveden náhled na práci s daty za pomoci vybraného softwaru DIAdem od společnosti National Instruments za menší pomoci Excelu. DIAdem byl vybrán zejména díky možnostem nahrávání práce uživatele v programu (neboli jinými slovy nahrávání makra s převodem do skriptu).

2.1. Výstupní (surová) data

První věc, která bude uživatele zajímat před analýzou a vyhodnocením, bude formát výstupních dat z měření či výpočtů. V následujících kapitolách tak bude nahlíženo na tuto problematiku. Přibude taktéž (již dříve avizovaný) menší náhled do práce s CFD programem FIRE od AVL.

2.1.1. Výstupní data z experimentu

Experimentální měření nebývají složena z velkého počtu dat (pokud jsou uvažovány střední hodnoty z vysokofrekvenčního snímání), ani nedisponují takovým množstvím informací jako výpočtové modely. V numerických simulacích je možné v podstatě sledovat cokoliv a kdekoliv. Přes relativní jednoduchost dat je však občas nutné datové soubory před vyhodnocením upravovat, jakož o tom bude pojednáno dále.

2.1.1.1. HWA měření

Metoda měření žhaveným drátkem je toho dobrým příkladem. Zde je měření postupné, kdy dochází k traverzování sondou po průměru válce a je tudíž naměřena hrubší síť s menším počtem dat obsahujících několik informací. Na důsledky přítomnosti sondy v proudícím toku bylo již poukázáno, problémem je zde datový výstup.

```
# Rotat = probe rotation
# Dist = radial immersion
# V-axi = axial vector magnitude
# V-tng = tangential vector magnitude

# Rotat   Dist   V-axi   V-tng   File
# [deg]   [mm]   [kg/(s.m^2)][kg/(s.m^2)] number

ZONE T = "AT_Rp000", I = 46, F = POINT

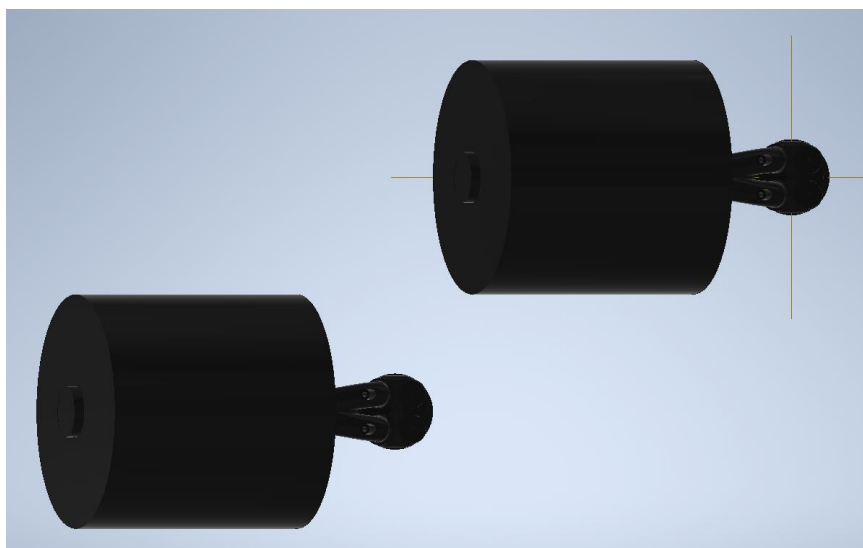
020      000      4.3      -1.2     1214202
020      000      4.5      -0.8     1214204
020      003      5.7      -0.5     1214206
020      003      5.7      -0.5     1214208
020      006     15.0       7.8     1214210
020      006     14.5       7.6     1214212
```

Obrázek 15: Surová data z HWA měření



2.1.2.1. Doporučení před simulacemi – AVL FIRE

Důležitou součástí přípravy modelu je určení jeho souřadnic, například jejich vhodné umístění do roviny mezi hlavou a válcem motoru. Případnou úpravu souřadnic je nutné provést předem v CAD programu, protože v AVL FIRE už není možnost dalších úprav.



Obrázek 18: Příklad extrémního posunu od základních souřadnic

Na obrázku 18 je zobrazen extrémní případ, kdy je válec mimo osu x a y , přičemž není zahrnuta ještě osa z . Samozřejmostí je možnost jiného natočení v prostoru, které je pro výstup taktéž nežádoucí.

Dalším doporučením je připravení výpočetní sítě v novějším FIRE M, kde je práce se síťováním mnohem snazší oproti konvenčnímu FIRE, ostatně v budoucnu bude postupný přechod na novější FIRE M. Simulace je poté možné provádět také ve FIRE M, ale kvůli problémům s nynější kompatibilitou byly výpočty a výstupy prováděny ve FIRE Workflow Manager.

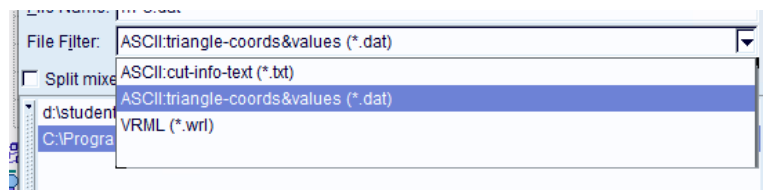
Potíží u novějších verzí softwaru však je to, že chybí například návody do staršího Workflow Manager ohledně profukovací zkoušky, ty tak byly dohledány ze starších verzí.

2.1.2.2. Výstupní data ze simulace v AVL FIRE

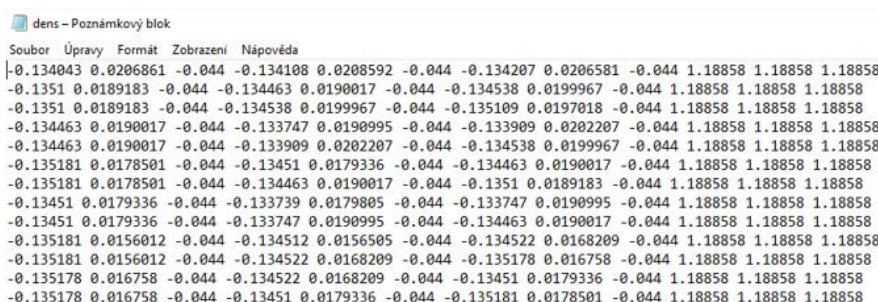
Výstup z AVL FIRE je možné získat v různých formátech, přičemž je vhodné použít formátu se souřadnicemi určujícími trojúhelníky. Export těchto dat je možný po rozkliknutí řezu modelu, nebo příslušného řezu pod *Impress-Cuts* ve stromu projektu. Po vyskočení příslušného okna lze pod záložkou *Info* exportovat data pod tlačítkem *Export*. Bude se jednat o soubory s příponou *.dat*, obsahujícími vždy tři souřadnice tří bodů trojúhelníku, tedy celkem 9 sloupců x , y a z .



První tři sloupce vyznačují souřadnice trojúhelníku bodu *A*, další bodu *B*, a nakonec bodu *C*. Poslední tři sloupce poté označují velikost požadované veličiny v každém bodě trojúhelníku. Náhled na výběr formátu a data je na obrázcích 19 a 20.



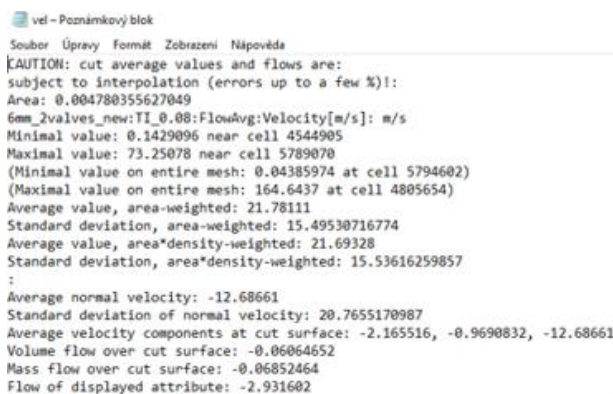
Obrázek 19: Požadovaný formát dat z FIRE



Obrázek 20: Pohled na data hustoty z FIRE

Použitím zmiňovaného datového formátu lze dosáhnout jednoduchého vyhodnocení jednotlivých ploch, k nimž jsou dány hodnoty. Snadno tak může být například spočten průtok v jednotlivých elementech.

Užitečný je taktéž textový soubor (viz obrázek 21), který vypíše hodnoty v daném řezu. Tento soubor lze vyexportovat použitím formátu *txt* (cut-info-text). V souboru jsou vypsána rozpětí dané veličiny (zde celkové rychlosti), její střední hodnota, nebo také objemový či hmotnostní průtok v daném průřezu. Řez ale lze udělat jen rovinou kolmou k nějaké souřadné ose. Proto je nutné kontrolovat, jestli řez touto rovinou nezahrne i oblast, ve které tyto hodnoty vyhodnocovat nechceme. Např. tento řez kolmý k ose válce ve vzdálenosti $0,6D$ zahrnuje i část vstupní nádoby.



Obrázek 21: Vypis důležitých hodnot, zde společně s celkovou rychlostí



2.2. Nahrání dat do DIAdemu

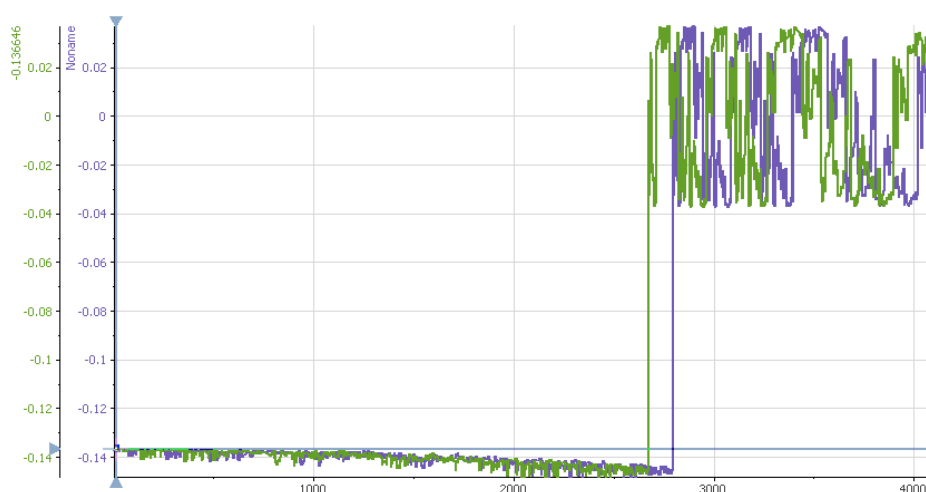
Data, která budou předmětem analýzy a vizualizace, jsou aktuálně připravena k importu do zvoleného softwaru DIAdem. Hodnoty naměřené při experimentech jsou zde ve formátu *csv*, tudíž stačí vybrat autodetekci *csv* souborů z nabídky podporovaných formátů dat. Data ze souboru se shromáždí do skupiny (*group*) a jednotlivé veličiny do kanálů (*channels*). Je třeba si dát pozor u *csv* souborů z českého prostředí, protože například *csv* soubor z Excelu vypíše místo teček desetinné čárky a místo čárek dělá mezery formou středníků.

Co se týká dat z numerických výpočtů, zde z AVL FIRE, je nezbytné vytvoření datového plug-inu. Je zapotřebí přiřadit názvy jednotlivým sloupcům tak, aby nebyly názvy převzaty z hodnot. Jak je vidět na obrázku 22, skupina i kanály jsou pojmenované a zároveň software neignoruje počáteční hodnoty, které jinak bere jako názvy.

Name	Noname	-0.136646	
Group name	dens	-0.136817	
Unit			
Length	7538	7414	
Channel Contents			
1	-0.136816	-0.136646	
2	-0.136646	-0.136646	
3	-0.136646	-0.136646	
4	-0.136646	-0.136646	
5	-0.136646	-0.136859	
6	-0.136646	-0.136859	
7	-0.137183	-0.136646	

Obrázek 22: Data s a bez plug-inu

Data si je poté možné prohlédnout pod záložkou *VIEW* (viz obrázek 23), kde je možnost vidět, že došlo k posunu hodnot (zmiňované ignorování hodnot).



Obrázek 23: Posun hodnot ze dvou zmiňovaných datových skupin

Dále bude popsáno naskriptované importování dat do DIAdemu.

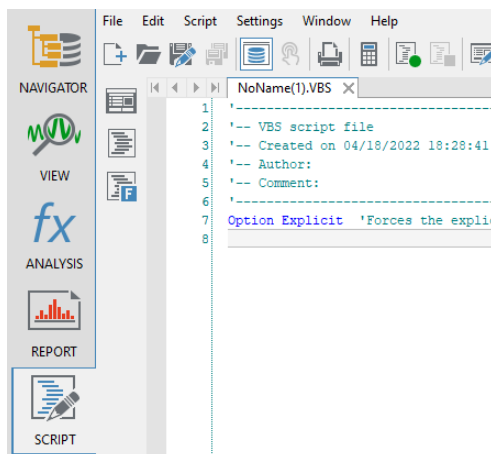


2.3. Skriptování

Tato kapitola bude popisovat skriptovací část ze softwaru DIAdem, jež je poslední záložkou v programu. První záležitostí skriptu bude zmiňovaný import dat do programu.

2.3.1. Základní import dat

Na obrázku 24 jsou patrné záložky *ANALYSIS* a *REPORT*, přičemž pod těmito záložkami jsou funkce, které budou použity až po nahrání dat.



Obrázek 24: Záložky DIAdem, otevřen *SCRIPT*

Dále je výpis ze skriptu, kde se určí proměnná *MyFolders* (a její velikost), která určuje cesty k datům. Po nadefinování funkce už je možno vyvolávat data s danými názvy, které je vhodné dodržovat nebo alespoň si určit jednotné názvosloví podle svého uvážení. Data, jak je vidět, jsou nahrávána přes vytvořený datový plug-in pro data z FIRE. Rozměr proměnné *MyFolders* je 2, jelikož je použito číslování 0 a 1, což tedy dává dohromady 2 hodnoty.

```
'Opens MyFolders directory - nastavení cest ke složkám s daty
Dim MyFolders ()
-----
Sub InitMyFolders
    ReDim MyFolders(2)
    MyFolders(0) = "C:\analyza\data\"
    MyFolders(1) = "C:\analyza\"
End Sub
Call InitMyFolders

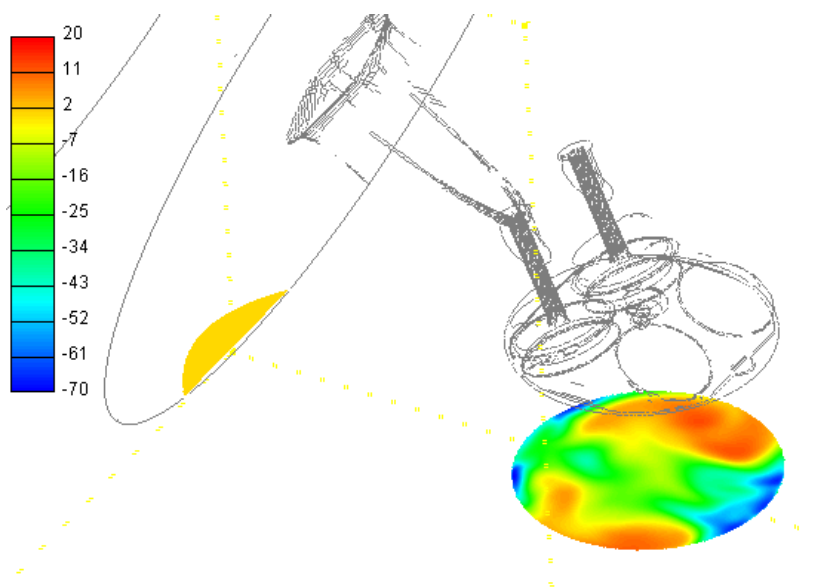
''Import dat ''
'Načte data z FIRE
Call DataFileLoad(MyFolders(0) &"vel_U.dat", "DAT_DataPlugin_AVLFire",
"Load|ChnXYRelation")
Call DataFileLoad(MyFolders(0) &"vel_V.dat", "DAT_DataPlugin_AVLFire",
"Load|ChnXYRelation")
Call DataFileLoad(MyFolders(0) &"vel_W.dat", "DAT_DataPlugin_AVLFire",
"Load|ChnXYRelation")
Call DataFileLoad(MyFolders(0) &"dens.dat", "DAT_DataPlugin_AVLFire",
"Load|ChnXYRelation")
```



Pro import dat například z experimentů, které jsou zde ve formátu *csv*, je potřeba použít datového plug-inu pro *csv*. To tedy znamená pouze změnit název plug-inu a případně název datových souborů (samozřejmě společně s očíslováním rozměru proměnné).

Spuštěním tohoto skriptu nastane požadovaný import dat. Data zde vytvoří 4 skupiny s celkem 48 kanály určujícími vždy tři body trojúhelníku a tři hodnoty dané veličiny. Ve druhém rozměru proměnné *MyFolders* je vhodné nahrát obecné záležitosti, jako je například zmiňovaná okrajová kružnice (vymezení válce) či grafické prostředí. Koneckonců, všechna data a soubory si lze uložit do jedné složky, není to však vhodné kvůli přehlednosti.

Vhodné je si data také odfiltrovat tak, aby byla pouze data z válce, nikoliv společně s ovlivňujícím řezem z uklidňovací nádoby, jak je vidět na obrázku 25. Zahrnutí tohoto prostoru mimo sledovanou oblast je způsobeno řezem (rovinou *xy*), jenž prochází celým modelem.



Obrázek 25: Pohled na řez válcem a zároveň i nádobou – axiální složka rychlosti v *m/s*

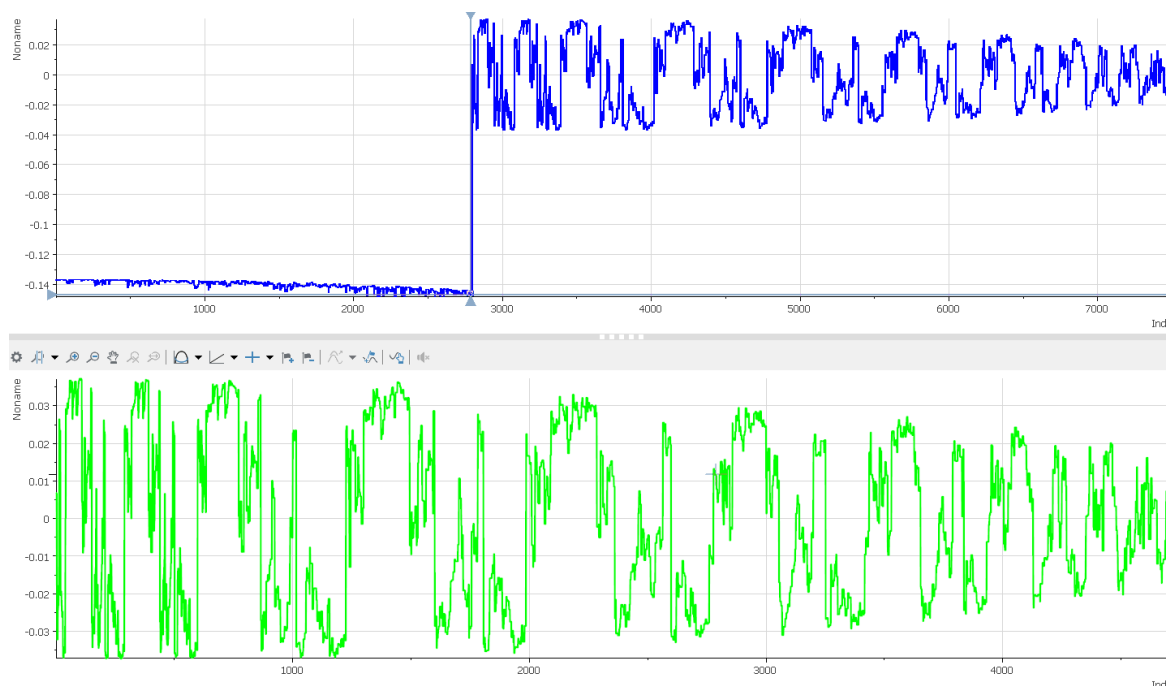
Problém je toho docílit, jelikož se hodnoty velikostí překrývají s hodnotami z prostoru válce (samozřejmě kromě hodnot souřadnic, které jsou jiné). Možností je data seřadit podle souřadnic, zleva od nádoby až doprava po stěnu válce. Po seřazení, které je zde výchozím uspořádáním po exportu z FIRE, je možné provést odstranění. Příkazem níže se tak smažou počáteční hodnoty všech kanálů v daném intervalu (ve výpisu níže je to 3000 hodnot).

```
'Ošetření dat - smazání dat ze vstupní nádoby - určit např. ve VIEW  
Call DataBlDel("'vel_U/Noname' - 'dens/NoName_11'", 1, 3000, True)
```

Název před lomítkem odkazuje na skupinu a za lomítkem na kanál (názvy lze také nahradit číselnými odkazy v hranatých závorkách). DIAdem defaultně pojmenovává ve vytvořeném pluginu kanály „*Noname*“ a indexuje je dle pořadí sloupců v datovém souboru.



Na obrázku 26 je vidět příklad filtrace souřadnice x_A (bod A trojúhelníka). Nutné je si data předem prohlédnout například v záložce *VIEW* a určit počet hodnot ke smazání. Samozřejmě toto není nutné provádět, pokud bude soubor obsahovat pouze relevantní data určená k vyhodnocení.



Obrázek 26: Rozdíl v hodnotách před smazáním (nahore) a po smazání (dole) – filtrace

Výhodné může na druhou stranu být vymezení tohoto úseku nevyhovujících dat až po přepočtu na jednotlivé veličiny. Při prvotním nahrání dat se totiž jedná pouze o matoucí zmeť sloupců s číselným značením. Výsledných dat je ale poté mnohem více s ještě horší přehledností.

Další možností, jak odfiltrovat tyto hodnoty, je redukované nahrávání dat. Data je možné si předem projít přímo v datovém souboru (viz obrázek 27). Bude ale problém nalézt vyznačený přechod z nulových hodnot. Příhodné tudíž bude provedení této analýzy v softwaru a až poté toto pásmo vymezit (jako na obrázku 26).

```
-0.145351 0.134656 -0.044 -0.145873 0.132826 -0.044 -0.145471 0.132977 -0.044 0.0134851 0.011533 0.0140141  
-0.145351 0.134656 -0.044 -0.145471 0.132977 -0.044 -0.144923 0.133736 -0.044 0.0134851 0.0140141 0.0149128  
-0.145351 0.134656 -0.044 -0.144923 0.133736 -0.044 -0.145292 0.134631 -0.044 0.0134851 0.0149128 0.0138436  
0.00676391 0.0863844 -0.044 0.00803343 0.0866415 -0.044 0.00799092 0.0868368 -0.044 3.07035 3.23847 3.6918  
0.00676391 0.0863844 -0.044 0.00799092 0.0868368 -0.044 0.00672754 0.086581 -0.044 3.07035 3.6918 3.49316  
-0.0177692 0.0902743 -0.044 -0.0176736 0.09045 -0.044 -0.0195196 0.0915261 -0.044 13.306 14.0143 17.3775
```

Obrázek 27: Přechod mezi nulovými daty a daty relevantními

Závěrem je tedy nutné podotknout, že je zde potřeba změnit cesty k datům a počet hodnot, které jsou určeny k filtraci neboli k redukcii. Pokud však není potřeba data filtrovat, tak tento příkaz samozřejmě odpadá.



2.3.2. Přepočet surových dat na souřadnice a veličiny

Souřadnice z dat byly doposud ve formě tří bodů trojúhelníku, ty budou nyní sečteny a poděleny třemi, aby se získala jednotná střední hodnota souřadnic. V tomto případě už jsou data odfiltrována/ošetřena, tudíž bude počítáno jen s potřebnými hodnotami:

$$x = \frac{x_A + x_B + x_C}{3} \quad y = \frac{y_A + y_B + y_C}{3} \quad (25)$$

První přepočet se týká souřadnic x a y (viz rovnice 25), poté jednotlivých složek rychlosti U , V a W , a nakonec k tomu je zde ještě hustota ρ , značená jako rho . Jelikož jsou souřadnice v metrech, jsou dále násobené tisícem, aby se získaly v milimetrech (viz skript níže).

```
'Přepočet na souřadnice x,y a rychlosti U,V,W + hustota rho
'x y - souřadnice
Call Calculate ("Ch ("vel_U/x") =
(Ch ("vel_U/Noname_9") + Ch ("vel_U/Noname_10") + Ch ("vel_U/Noname_11")) / 3)
Call Calculate ("Ch ("vel_U/y") =
(Ch ("vel_U/Noname_1") + Ch ("vel_U/Noname_4") + Ch ("vel_U/Noname_7")) * 1000 / 3)

'U V W - složky rychlosti
Call Calculate ("Ch ("vel_U/U") =
(Ch ("vel_U/Noname_9") + Ch ("vel_U/Noname_10") + Ch ("vel_U/Noname_11")) / 3)
Call Calculate ("Ch ("vel_U/V") =
(Ch ("vel_V/Noname_9") + Ch ("vel_V/Noname_10") + Ch ("vel_V/Noname_11")) / 3)
Call Calculate ("Ch ("vel_U/W") =
(Ch ("vel_W/Noname_9") + Ch ("vel_W/Noname_10") + Ch ("vel_W/Noname_11")) / 3)

'rho - hustota
Call Calculate ("Ch ("vel_U/rho") =
(Ch ("dens/Noname_9") + Ch ("dens/Noname_10") + Ch ("dens/Noname_11")) / 3)
```

Veškeré souřadnice, veličiny a další spočítané hodnoty jsou zde uspořádány do jedné skupiny. Seskupení je z toho důvodu, aby bylo jednodušší počítání ve skriptu a taktéž kvůli přehlednosti (ve skriptu, nikoli v datech!).

Seskupení lze docílit odkazováním pomocí kódu `Ch(“skupina/kanál“)` ve funkci `Call Calculate`. Ta zde slouží například k práci mezi různými skupinami, běžně se totiž pracuje v jedné skupině.

Druhá část přepočtu je vztahována k určení vah jednotlivých hodnot formou normalizace příslušných ploch trojúhelníků, jež jsou dány právě třemi body. K určení plochy je určen Heronův vzorec z rovnice 26, kterým lze vypočítat obsah trojúhelníku z délek jeho stran.



$$S = \sqrt{s(s-a)(s-b)(s-c)} \quad (26)$$

$$s = \frac{a+b+c}{2} \quad (27)$$

Kde S je plochou trojúhelníku a malé s je poté jeho polovičním obvodem. Ve skriptu je však poloviční obvod značen malým o z důvodu programování (byly by stejné názvy).

Ve skriptu dále je poté pohled na tento výpočet a zároveň i provedená normalizace těchto ploch s intervalem od 0 do 1. [14]

```
'Obsah trojúhelníků sítě - Heronův vzorec
  Call Data.Root.Channelgroups.Item("vel_U").Channels.AddCalculationChannel("a",
"= sqrt((Noname_3-Noname_6)^2+(Noname_4-Noname_7)^2)",,2)
  Call Data.Root.Channelgroups.Item("vel_U").Channels.AddCalculationChannel("b",
"= sqrt((Noname-Noname_6)^2+(Noname_1-Noname_7)^2)",,2)
  Call Data.Root.Channelgroups.Item("vel_U").Channels.AddCalculationChannel("c",
"= sqrt((Noname-Noname_3)^2+(Noname_1-Noname_4)^2)",,2)
  Call Data.Root.Channelgroups.Item("vel_U").Channels.AddCalculationChannel("o",
"= (CH("vel_U/a")+CH("vel_U/b")+CH("vel_U/c"))/2",,2)
  Call Data.Root.Channelgroups.Item("vel_U").Channels.AddCalculationChannel("S",
"= sqrt(CH("vel_U/o")*(CH("vel_U/o")-CH("vel_U/a"))*(CH("vel_U/o")-
CH("vel_U/b"))*(CH("vel_U/o")-CH("vel_U/c")))",,2)

'Normalizování ploch trojúhelníků - rozmezí 0 až 1
  Set ChnResult = ChnNormalize("vel_U/S", "vel_U/NormS")
```

Díky normalizaci je tak možné provést vážený průměr, jelikož výpočetní síť není po průměru rovnoměrná. Síť je totiž zhuštěna u stěn válce, a tudíž by například výsledný objemový tok byl ovlivněn značnou chybou.

Normalizace je zde jako funkce, možností je však použití vzorce, tedy dělení všech hodnot hodnotou nejvyšší.

V poslední části je výpočet celkových rychlostí ze tří a ze dvou složek rychlosti. To znamená celková rychlost v třírozměrném prostoru a celková rychlost v rovině xy . Některé odkazy na kanály ve výpočtech nefungují dle předpokladů, je proto nutné použít kódu $CH(“skupina/kanál“)$, což je přímý odkaz na kanál v určité skupině.

```
'Výpočty z dat - celková rychlost UVW (VEL) a UV
  Call
Data.Root.Channelgroups.Item("vel_U").Channels.AddCalculationChannel("VEL", "=
sqrt(CH("vel_U/U")^2 + CH("vel_U/V")^2 + CH("vel_U/W")^2)",,8)
  Call
Data.Root.Channelgroups.Item("vel_U").Channels.AddCalculationChannel("UV", "=
sqrt(CH("vel_U/U")^2 + CH("vel_U/V")^2)",,8)
```



Další fází je výpočet požadovaných integrálních veličin, jako jsou toky a z nich odvozené charakteristiky proudění ve válci motoru.

2.3.3. Výpočet požadovaných integrálních veličin

V této výpočetní fázi integrálních veličin je nutné určit proměnné. Proměnnými jsou myšleny zdvih, poloměr uložení ojnicního ložiska na klikové hřídeli, či vrtání válce, tedy průměr.

Dále je potřeba znát průměr ventilového sedla sacích ventilů, počet sacích ventilů a jejich zdvih, či tlakový spád zde vzniklý pro výpočet rychlosti proudění v sacích kanálech. Zde je však uvažována stlačitelná tekutina, tudíž je potřeba znát referenční tlak, tedy vstupní či výstupní.

```
'''Zadání proměnných!!!
Call Calculate ("Ch ("vel_U/Z") = 0.09 ")           'Zdvih v m
Call Calculate ("Ch ("vel_U/D") = 0.08 ")           'Vrtání - průměr v m
Call Calculate ("Ch ("vel_U/Dv") = 0.025 ")         'průměr ventilového sedla v m
Call Calculate ("Ch ("vel_U/Lv") = 0.006 ")         'Zdvih ventilu v m
Call Calculate ("Ch ("vel_U/p1") = 100000 ")        'Vstupní tlak v Pa
Call Calculate ("Ch ("vel_U/deltap") = 4900 ")      'Tlakový spád v Pa
Call Calculate ("Ch ("vel_U/Nv") = 2 ")             'Počet sacích ventilů
```

Když je například hustota dána jako konstantní (zejména u experimentů), je nutné napřímo zadat číselnou hodnotu hustoty do vzorců (například funkcí Find and Replace, kterou lze vyvolat zkratkou *CTRL+F*).

Při zadání proměnné a následném násobení s kanály o více hodnotách se převedou na tuto jednu hodnotu, jakou by měla v tomto případě hustota. Tím by pak samozřejmě nešlo počítat integrované veličiny.

Proměnné jsou tedy určeny, nyní je možné vypočítat požadované integrální veličiny odvozené z objemového toku, respektive hmotnostního toku. V následujících vztazích jsou použity rovnice 1, 2 a 28 pro výpočet toků a střední hodnoty (s ohledem na váhu hodnot).

```
'Váha hustoty, střední hustota, Objemový a hmotnostní tok
Call
Data.Root.Channelgroups.Item("vel_U").Channels.AddCalculationChannel("rhovaha", "=
CH("vel_U/rho") * CH("vel_U/NormS")", , 11)
Call
Data.Root.Channelgroups.Item("vel_U").Channels.AddCalculationChannel("strrho", "=
cch("vel_U/rhovaha", 4) / cch("vel_U/NormS", 4)", , 12)
Call
Data.Root.Channelgroups.Item("vel_U").Channels.AddCalculationChannel("V_tok", "=
(CH("vel_U/W") * CH("vel_U/S")", , 12)
Call
Data.Root.Channelgroups.Item("vel_U").Channels.AddCalculationChannel("M_tok", "=
CH("vel_U/rho") * (CH("vel_U/W") * CH("vel_U/S")", , 12)
```



Střední hustota je zde vypočítána z váženého průměru dle následujícího vzorce:

$$\bar{x} = \frac{\sum_{i=1}^n w_i x_i}{\sum_{i=1}^n w_i} \quad (28)$$

Kde x_i jsou dané hodnoty (zde hustoty) a w_i je vždy odpovídající váha hodnoty. Váhu hodnot zde znázorňují předem určené normalizované plochy trojúhelníků. Pod kódem `cch("skupina/kanál", "číslo funkce)` uvedeném ve skriptu se skrývají různé funkce.

V tomto případě je uvedena funkce číslo 4, což je funkce sumární. Jedná se tedy o podíl sum součinnů vah s hodnotami a sum samotných vah. Ve výsledku to dává požadovaný vážený průměr.

V další části skriptu s výpočtem je již dříve popsáný moment toku hybnosti pro swirl a tumble. Integrace je zde zajištěna sumou jednotlivých toků, čímž je myšlenka integrace splněna (myšlené sumy ploch pod křivkou).

```
'Jednotlivé toky momentu hybnosti - zde pro swirl - Bsi=(x*V-Y*U)*rho*W*Plocha
  Call Data.Root.Channelgroups.Item("vel_U").Channels.AddCalculationChannel("Bsi",
  "= (CH("vel_U/x")*CH("vel_U/V")) -
  CH("vel_U/y")*CH("vel_U/U"))/1000*CH("vel_U/rho")*CH("vel_U/W")*CH("vel_U/
  S")", , 13)
  Call Data.Root.Channelgroups.Item("vel_U").Channels.AddCalculationChannel("Bs",
  "= cch("vel_U/Bsi", 4)", , 14)

'Tumble Bx = y * rho * W^2 * S      By = Bx = x * rho * W^2 * S
  Call Data.Root.Channelgroups.Item("vel_U").Channels.AddCalculationChannel("Bxi",
  "= CH("vel_U/y")*CH("vel_U/rho")*(CH("vel_U/W"))^2*CH("vel_U/S")/1000", , 15)
  Call Data.Root.Channelgroups.Item("vel_U").Channels.AddCalculationChannel("Byi",
  "= CH("vel_U/x")*CH("vel_U/rho")*(CH("vel_U/W"))^2*CH("vel_U/S")/1000", , 16)
  Call Data.Root.Channelgroups.Item("vel_U").Channels.AddCalculationChannel("Bt",
  "= sqrt((cch("vel_U/Bxi", 4))^2 + (cch("vel_U/Byi", 4))^2)", , 17)
```

Výpočet toků momentů hybnosti je splněn, nyní je potřeba vypočíst swirl a tumble čísla.

```
'Swirl a Tumble čísla
  Call Data.Root.Channelgroups.Item("vel_U").Channels.AddCalculationChannel("SN",
  "= 2*Z*strrho*Bsi/(cch("vel_U/M_tok", 4))^2", , 18)
  Call Data.Root.Channelgroups.Item("vel_U").Channels.AddCalculationChannel("TN",
  "= 2*Z*strrho*Bt/(cch("vel_U/M_tok", 4))^2", , 19)
'Redukovaná
  Call Data.Root.Channelgroups.Item("vel_U").Channels.AddCalculationChannel("SNr",
  "= 2*D*strrho*Bsi/(cch("vel_U/M_tok", 4))^2", , 18)
  Call Data.Root.Channelgroups.Item("vel_U").Channels.AddCalculationChannel("TNr",
  "= 2*D*strrho*Bt/(cch("vel_U/M_tok", 4))^2", , 19)
```

Ta jsou vypočítána dle již uvedených vzorců, zároveň i s vyhodnocením redukováných hodnot na „čtvercové“ motory. To znamená, že se zdvih rovná vrtání. V těchto vztazích je použito středních hodnot hustoty a celkového toku vzniklého ze sumy jednotlivých toků po průřezu.



Tímto jsou určeny vírové charakteristiky ve smyslu swirl a tumble. Další výpočet je zaměřen na průtokové charakteristiky jako jsou součinitele. V tomto případě je zde výpočet potřebné výstupní rychlosti počítán včetně uvažování stlačitelnosti.

Co se týče průtokového součinitele a průtokového čísla, ty jsou brány z teoretického průtoku proudícího přes sací ventily. Skutečný průtok je zde uvažován v průřezu válce, kde probíhá zkoumání rychlostních polí, buďto tedy měřeními či simulacemi.

```
'Průtokové charakteristiky
'Výstupní rychlost
  Call Data.Root.Channelgroups.Item("vel_U").Channels.AddCalculationChannel("v2",
"= sqrt(7*p1/strrho*(1-(((p1-deltap)/p1)^(0.4/1.4))))",,11)

'Průtokový součinitel
  Call Data.Root.Channelgroups.Item("vel_U").Channels.AddCalculationChannel("Sv",
"= PI*Dv*Lv",,11)
  Call Data.Root.Channelgroups.Item("vel_U").Channels.AddCalculationChannel("Cd",
"= cch("vel_U/M_tok",4)/(CH("Nv")*strrho*Sv*v2)",,11)

'Průtokové číslo
  Call Data.Root.Channelgroups.Item("vel_U").Channels.AddCalculationChannel("Sp",
"= PI*Dv^2/4",,11)
  Call Data.Root.Channelgroups.Item("vel_U").Channels.AddCalculationChannel("Cf",
"= cch("vel_U/M_tok",4)/(CH("Nv")*strrho*Sp*v2)",,11)

'Uzavírací číslo
  Call
  Data.Root.Channelgroups.Item("vel_U").Channels.AddCalculationChannel("Sigma", "=
CH("vel_U/Cf")/CH("vel_U/Cd")",,11)
```

V tomto případě je referenčním tlakem tlak na vstupu a od něj se poté odečítá tlakový spád. Tím je získán tlak výstupní. Poissonova konstanta κ je zde uvažována 1,4 (tedy pro dvouatomové plyny), a tudíž je zde výpočtový vzorec rychlosti zjednodušen. Dále už je postupováno podle již uvedených vzorců s třemi výslednými hodnotami.

Shrnutím je zde tedy nutné změnit proměnné před spuštěním skriptu, zejména geometrii, která bude nejspíše rozdílná. Ostatní příkazy není nutné měnit a lze je tak nechat beze změn.

2.3.4. Uspořádání dat pro přehlednost

Nyní bude text zaměřen na přenos spočtených veličin do jednotné skupiny. Vytvoří se tak nová skupina, kde budou všechna data uspořádána pro pozdější jednodušší analýzu a vizualizaci (viz skript dále).

```
'Pouze přenos kanálu do jiné skupiny (základní skupiny budou smazány)
'Základní souřadnice a veličiny
  Call Calculate("Ch("hlava1/x")= Ch("vel_U/x")")
  Call Calculate("Ch("hlava1/y")= Ch("vel_U/y")")
  Call Calculate("Ch("hlava1/W")= Ch("vel_U/W")")
'Rychlosti celkové ve 3D a 2D
  Call Calculate("Ch("hlava1/VEL")= Ch("vel_U/VEL")")
  Call Calculate("Ch("hlava1/UV")= Ch("vel_U/UV")")
```

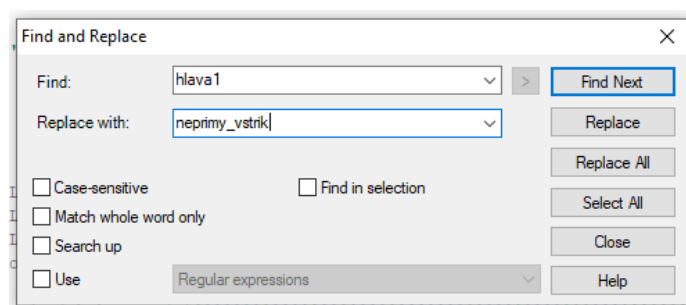


```
'Swirl a Tumble
  Call Calculate ("Ch("hlava1/SN") = Ch("vel_U/SN")")
  Call Calculate ("Ch("hlava1/TN") = Ch("vel_U/TN")")
'Swirl a Tumble redukovaný
  Call Calculate ("Ch("hlava1/SNr") = Ch("vel_U/SNr")")
  Call Calculate ("Ch("hlava1/TNr") = Ch("vel_U/TNr")")
'Objemový a hmotnostní tok
  Call Calculate ("Ch("hlava1/V_tok") = Ch("vel_U/V_tok")")
  Call Calculate ("Ch("hlava1/M_tok") = Ch("vel_U/M_tok")")
'Průtokový souč., číslo a uz. číslo
  Call Calculate ("Ch("hlava1/Cd") = Ch("vel_U/Cd")")
  Call Calculate ("Ch("hlava1/Cf") = Ch("vel_U/Cf")")
  Call Calculate ("Ch("hlava1/Sigma") = Ch("vel_U/Sigma")")
```

Dle příkazů *Calculate* na předchozím skriptu jsou výsledné veličiny přeneseny do jednotné skupiny. Nyní už je možné předchozí data smazat tak, aby zůstala pouze ta výsledná potřebná.

```
'Smaže přebytečná data (groups) - z nich jsou přepočítané souřadnice a složky rychlostí
Dim oElementList
  Set oElementList = data.CreateElementList()
'Smazání přebytečných skupin - groups
  Call oElementList.Add(Data.Root.ChannelGroups("vel_U"))
  Call oElementList.Add(Data.Root.ChannelGroups("vel_V"))
  Call oElementList.Add(Data.Root.ChannelGroups("vel_W"))
  Call oElementList.Add(Data.Root.ChannelGroups("dens"))
Data.Remove(oElementList)
```

V příkazech s proměnnou *oElementList* (na skriptu výše) jsou minulé a nyní už tedy nepotřebná data smazána. Potřeba je zde pouze změnit název skupiny, do které budou data seskupena. Změnu názvu lze jednoduše docílit klávesovou zkratkou *Ctrl+F*. Použitím této zkratky vyskočí okno, kde je již možno daný text ve skriptu přepsat na jiný dle potřeby (viz obrázek 28).



Obrázek 28: Hledání a změna textu

2.3.5. Vytvoření vektorů pro zobrazení rychlostních polí

Nyní už se jedná víceméně o nadstavbu na základní veličiny. Vektory je v případě DIAdemu možno zobrazit pomocí dvou bodů, tedy počátečního a koncového. Nutností je tak přičíst k dané souřadnici rychlosti (viz příkazy dále), přičemž počátek je určen samotnou souřadnicí a konec vektoru tímto součtem (tím je dáno umístění a rozdílem bodů velikost vektoru).



V první fázi jsou opět nahrána data s rychlostmi o třech složkách U , V a W . Rozdíl je zde však v tom, že data jsou nahrána v předem daném kvantu hodnot (zde čistě nahodile v počtu 2000). Interval je určen z toho důvodu, že při zobrazení všech hodnot (zde v základu přes 7 tisíc hodnot) přestává být vizualizace rychlostních polí čitelná.

Když se data nahrají v omezeném počtu, pokračuje již zmiňovaný výpočet dvou bodů jednotlivých vektorů. Po spočítání se nakonec opět smažou nepotřebné skupiny pomocí předem definované proměnné.

```
'Znovu načtení dat, ale přepočítání na vektory - méně hodnot
Call DataFileLoadRed(MyFolders(0)&"vel_U.dat", "DAT_DataPlugin_AVLFire", "",
"IntervalCount", 2000, 8, 0, 0)
Call DataFileLoadRed(MyFolders(0)&"vel_V.dat", "DAT_DataPlugin_AVLFire", "",
"IntervalCount", 2000, 8, 0, 0)
Call DataFileLoadRed(MyFolders(0)&"vel_W.dat", "DAT_DataPlugin_AVLFire", "",
"IntervalCount", 2000, 8, 0, 0)

Call Calculate("Ch("hlava1/xU1")=
(Ch("vel_U/Noname_Mean")+Ch("vel_U/Noname_3_Mean")+Ch("vel_U/Noname_6_Mean"))
*1000/3")
Call Calculate("Ch("hlava1/xU2")=
(Ch("vel_U/Noname_9_Mean")+Ch("vel_U/Noname_10_Mean")+Ch("vel_U/Noname_11_Mean
"))/3 + Ch("hlava1/xU1") ")

Call Calculate("Ch("hlava1/yV1")=
(Ch("vel_V/Noname_1_Mean")+Ch("vel_V/Noname_4_Mean")+Ch("vel_V/Noname_7_Mean"
))*1000/3")
Call Calculate("Ch("hlava1/yV2")=
(Ch("vel_V/Noname_9_Mean")+Ch("vel_V/Noname_10_Mean")+Ch("vel_V/Noname_11_Mean
"))/3 + Ch("hlava1/yV1") ")

Call Calculate("Ch("hlava1/zW1")=
(Ch("vel_W/Noname_2_Mean")+Ch("vel_W/Noname_5_Mean")+Ch("vel_W/Noname_8_Mean"
))*1000/3")
Call Calculate("Ch("hlava1/zW2")=
(Ch("vel_W/Noname_9_Mean")+Ch("vel_W/Noname_10_Mean")+Ch("vel_W/Noname_11_Mean
"))/3 + Ch("hlava1/zW1") ")

Call oElementList.Add(Data.Root.ChannelGroups("vel_U"))
Call oElementList.Add(Data.Root.ChannelGroups("vel_V"))
Call oElementList.Add(Data.Root.ChannelGroups("vel_W"))
Data.Remove(oElementList)
```

Tímto je momentálně pokryt základní skript zpracovávající simulační data jedné profukované hlavy pro následnou analýzu a vizualizaci.

Pokud bude požadavek analýzy více provedených simulací, tak asi nejvhodnějším postupem je změna cesty k datům. Dalším postupem bude zkopírování příkazů pro nová data s tím, že se budou muset upravit podle dat k nim příslušejícím. To se týká například proměnných ve formě zdvihů či vrtání.



```
'Práce s dalšími daty  
'Změna cesty k datům  
MyFolders(0) = "C:\analyza\data2\"
```

Po příkazu výše bude opakovaně použit skript uvedený v předchozích kapitolách. Nabízí se proto různé možnosti automatického opakování řádků bez nutnosti opětovného vypisování příkazů. V tomto případě, kdy se však musí některé hodnoty měnit ručně, to není uskutečnitelné.

2.3.6. Skriptování více simulací – proměnné otevření ventilů

Jak již bylo naznačeno, při práci s více daty se nabízí opakovací funkce. Problém však nastává nejenom při samotném zjištění, jak tohoto dosáhnout, ale taky jak si poradit s proměnnými.

Potíž nastává, když je nutné zajistit opakování vybraných segmentů skriptu. Podobná funkce opakování řádků však nebyla zjištěna, a tak je nutné si poradit jinak. Skript je tak potřeba kopírovat pro stejná data několikrát, čímž se zhoršuje přehlednost a zvyšuje časová náročnost. Ostatně v případě simulace modelu s proměnným zdvihem je potřeba i text ve skriptu přepisovat, což s opakovacími funkcemi nejde dohromady.

Ve většině případů je možné si vytvořit samostatné výstupy a ty poté srovnávat mezi sebou, případně použít méně objektů k porovnání (zde hlav motorů).

Případ použití opakování skriptu je například u proměnného otevření ventilů, jenž zde bylo zkoumáno na hlavě s nepřímým vstřikem paliva. Na této hlavě byly zkoumány nejen proměnné zdvihy ventilů, ale také různé počty otevření sacích ventilů (respektive zde otevření jednoho či dvou ventilů). Skript je tím pádem dosti dlouhý a méně přehledný, je však součástí vypracovaných vzorů v příloze. Čtenář tak může použít stávající skript či na něj navázat dle libosti.

Pro částečné zjednodušení je zde uvažovaná střední hustota pro všechny případy stejná. Dále proměnné týkající se zdvihu ventilů a počtu otevřených ventilů jsou rovnou zapsány ve vzorcích. Celkové rychlosti v prostoru a ploše jsou taktéž vynechány (včetně určení vektorů). Jedná se zde tedy v podstatě o skript vyhodnocující vírové a průtokové charakteristiky 18 různých simulací (9 různých zdvihů jednoho a dvou ventilů).

Pro vyhodnocovanou složku z dat (tedy rychlost) je zde určen pouze jeden sloupec oproti třem v předchozí kapitole. Tato skutečnost je nejspíše dána starší verzí SW, z kterého byla data odebrána.



```
'Spočtení souřadnic x a y + složka rychlosti
Call DataFileLoad(MyFolders(0)&"velocity_W_1mm_2s.dat", "DAT_DataPlugin_AVLFire",
"Load|ChnXYRelation")
  Call Calculate("Ch(""2Valves/x_1mm"")=
(Ch(""velocity_W_1mm_2s/Noname"")+Ch(""velocity_W_1mm_2s/Noname_3"")+Ch(""velocity_
W_1mm_2s/Noname_6""))*1000/3")
  Call Calculate("Ch(""2Valves/y_1mm"")=
(Ch(""velocity_W_1mm_2s/Noname_1"")+Ch(""velocity_W_1mm_2s/Noname_4"")+Ch(""velocit
y_W_1mm_2s/Noname_7""))*1000/3")
  Call Calculate("Ch(""2Valves/W_1mm"")= Ch(""velocity_W_1mm_2s/Noname_9""))

Call DataFileLoad(MyFolders(0)&"velocity_W_1mm_1s.dat", "DAT_DataPlugin_AVLFire",
"Load|ChnXYRelation")
  Call Calculate("Ch(""1Valve/x_1mm"")=
(Ch(""velocity_W_1mm_1s/Noname"")+Ch(""velocity_W_1mm_1s/Noname_3"")+Ch(""velocity_
W_1mm_1s/Noname_6""))*1000/3")
  Call Calculate("Ch(""1Valve/y_1mm"")=
(Ch(""velocity_W_1mm_1s/Noname_1"")+Ch(""velocity_W_1mm_1s/Noname_4"")+Ch(""velocit
y_W_1mm_1s/Noname_7""))*1000/3")
  Call Calculate("Ch(""1Valve/W_1mm"")= Ch(""velocity_W_1mm_1s/Noname_9""))
```

Další zjednodušení je v jednotném odfiltrování dat, tedy zde smazání irrelevantních hodnot z prostoru uklidňovací nádoby. Každý soubor zde obsahuje jiný počet dat, a to v tomto případě v hodnotách pohybujících se kolem 33 tisíc. Nicméně počátečních přibližně 1880 hodnot je potřeba odfiltrovat.

Po nahlédnutí do počtu dat (počtu řádků) byl však vybrán jednotný počet určený k vymazání, a to ten nejmenší. Data v některých souborech jsou tak navíc a vzniká tudíž jistá, ale téměř zanedbatelná chyba při vyhodnocování průtoků (menší než 0,01 %).

```
'Ošetření dat - smazání dat ze vstupní nádoby - určit počet dat
Call DataBDel("'velocity_W_1mm_2s/Noname' - 'velocity_U_9mm_1s/Noname_9'", 1,
1876, True)
```

Skript se dále podobá předchozí kapitole, tudíž dochází k nahrání dat, přepočtu plošek, výpočtu požadovaných integrálních hodnot, a nakonec ke smazání přebytečných skupin.

2.3.7. Skriptování experimentů

Co se týká experimentů, tak bylo již zmíněno, že nelze změřit všechny veličiny. V případě měření pomocí anemometru byly vyhodnoceny dvě složky rychlosti (axiální a tangenciální) a co se týká snímkování v metodě PIV, tak opět dvě složky, ale pouze v jedné rovině (radiální a tangenciální). Tím pádem je potřeba kombinovat již uvedené skripty pro tyto případy, kdy není vše k dispozici.

Ve formátování dat je však opět rozdíl, protože veškeré naměřené veličiny z experimentů jsou shromážděny do jednoho souboru. To taktéž znamená kratší a jednodušší skript. Výjimkou však může být již zmiňovaný výstup dat z HWA měření, kdy je data nutné upravovat před dalším zpracováním.



Na skriptu dále je vidět jednoduchý příkaz nahrávající soubory dat, kde u PIV je navíc přepočít vektorů, jenž je možný díky dvou vyhodnoceným složkám rychlosti v rovině. U HWA metody je zde rychlost násobená -1 z důvodu opravy nesouhlasné orientace osy z .

```
'Práce s dalšími daty
'Změna cesty k datům
MyFolders(0) = "C:\analyza\exp_HWA\"

'Načte data z experimentu novějšího - žhavený drátek - HWA
    Call DataFileLoad(MyFolders(0)&"exp_HWA.csv", "CSV", "Load|ChnXYRelation")
    Call
Data.Root.Channelgroups.Item("exp_HWA").Channels.AddCalculationChannel("W", "=
CH("v-axi")*(-1)",,10)

'Práce s dalšími daty
'Změna cesty k datům
MyFolders(0) = "C:\analyza\exp_PIV\"

'Načte data z PIV experimentu
    Call DataFileLoad(MyFolders(0)&"exp_PIV.csv", "CSV", "Load|ChnXYRelation", "",
"IntervalCount", 3000, 8, 0, 0)
    Call
Data.Root.Channelgroups.Item("exp_PIV").Channels.AddCalculationChannel("xU", "=
CH("U[m\s]") + CH("x (mm) [mm]"),,8)
    Call
Data.Root.Channelgroups.Item("exp_PIV").Channels.AddCalculationChannel("yV", "=
CH("V[m\s]") + CH("y (mm) [mm]"),,8)
    Call
Data.Root.Channelgroups.Item("exp_PIV").Channels.AddCalculationChannel("UV", "=
sqrt((CH("U[m\s]"))^2 + (CH("V[m\s]"))^2)",,8)
```

Případné složité názvosloví veličin je možné modifikovat změnou názvu přímo v textovém souboru dat nebo následně ve skriptu pomocí následující funkce:

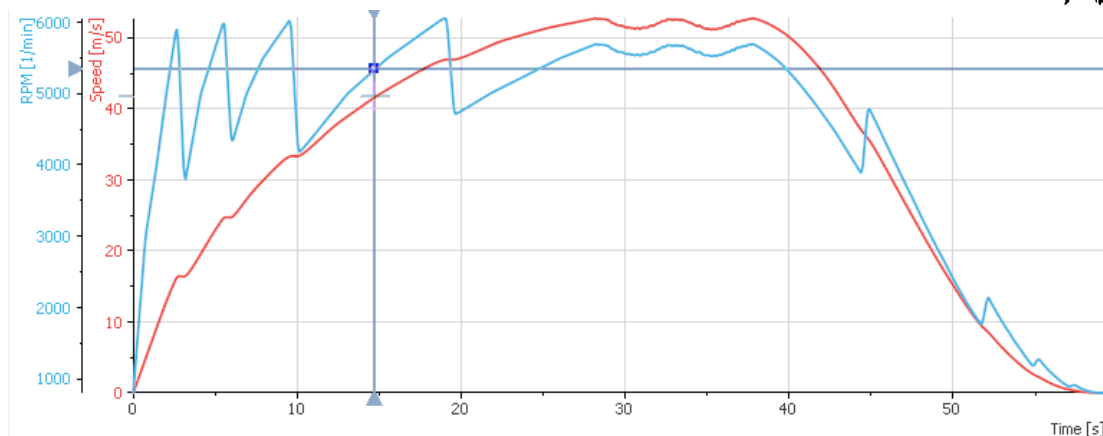
```
Data.Root.ChannelGroups(8).Channels("x (mm) [mm]").Name = "x"
```

Kdy v tomto případě v 8. skupině dojde k přejmenování složité pojmenované veličiny "x (mm) [mm]" na „x“. Následně budou popsány ještě modifikace těchto skriptů ve vytvořených vzorech. Data jsou tímto připravena k následné vizualizaci a analýze.

2.4. Vizualizace – grafické výstupy a vhodné grafy

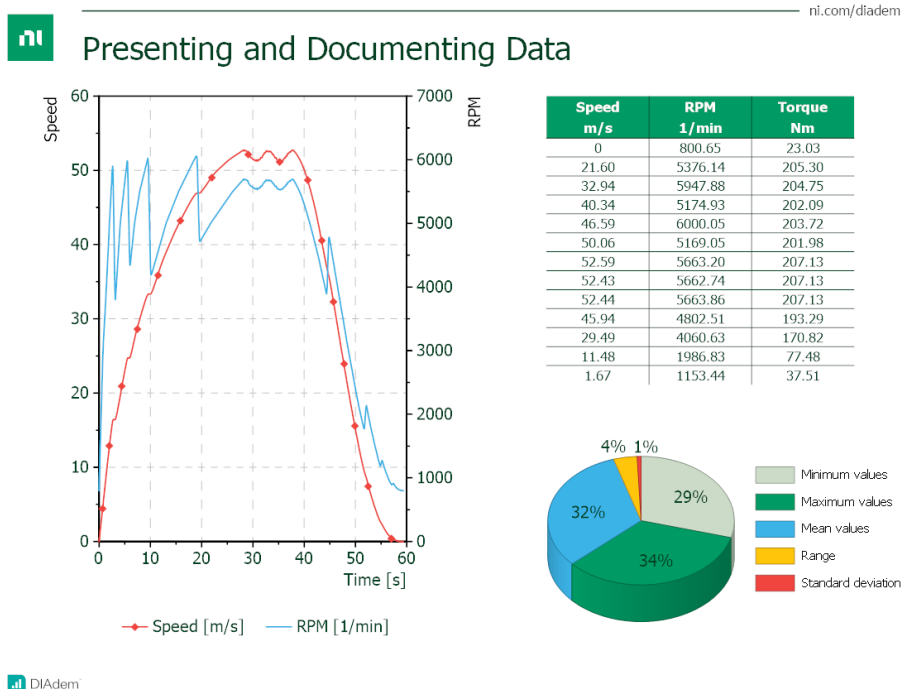
V minulých kapitolách byla data přepracována a přepočítána, čímž byly získány potřebné veličiny ke zkoumání. Nyní je nutné data nějakým způsobem vizualizovat pro následnou analýzu, jelikož vyhodnocování dat formou procházení kupy čísel není úplně vhodné.

K těmto účelům může například sloužit záložka *VIEW*, kde je možno si data rychle prohlédnout a zhodnotit, přičemž tato sekce slouží spíše jenom pro rychlá srovnání či nahlédnutí do obsáhlých dat. Vhodné je využít této sekce například k vymezení dat vhodných k filtraci.



Obrázek 29: Graf otáček a rychlosti ze vzorového příkladu – sekce VIEW

Pro účely nejen profukovací zkoušky, kdy je zkoumán třírozměrný prostor je však určitě vhodnější použít *REPORT* sekci. Ze sekce *VIEW* je možné si převést grafy a různé podobné výstupy do *REPORT* sekce. V *REPORT* sekci je poté možné nejrůznějšími způsoby vizualizovat svá data a následně vyexportovat ve vyhovujícím formátu.



Obrázek 30: Náhled na výstup ze vzorového příkladu – sekce REPORT

Na obrázku 30 je vidět výsledný možný výstup (zde například ve formě obrázku), na němž je dvourozměrný graf s tabulkou a koláčovým grafem. Možností je si tyto grafy naskriptovat ve skriptovací sekci, vhodnější je však si zvlášť vytvořit vlastní vizualizační prostředí, a to poté jednoduše nahrát přes skript. Součástí této práce však budou již vypracovaná vzorová vizualizační prostředí.

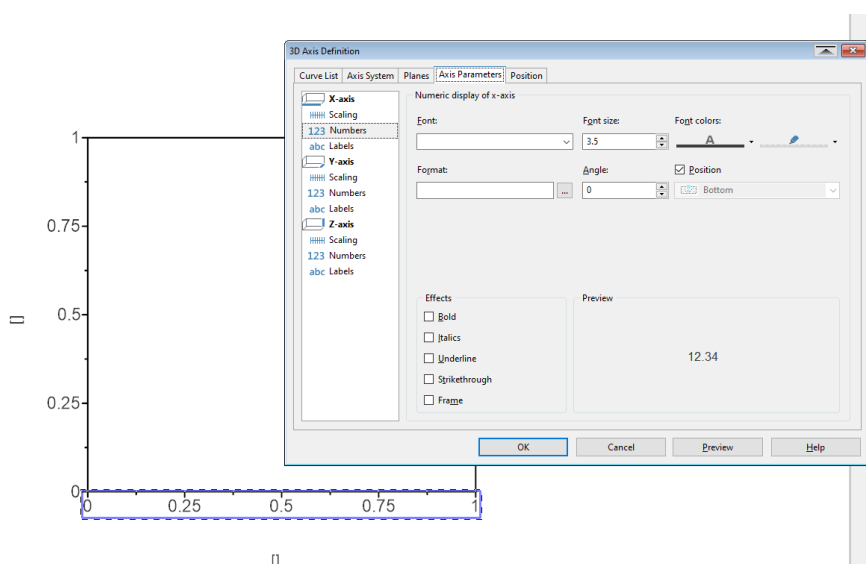


```
'Otevření a export report layout - Vizualizační prostředí  
Call Report.NewLayout()  
Call Report.LoadLayout(MyFolders(1) & "Vizualizacni_prostredi.TDR")  
Call Report.Refresh  
Call Report.Sheets.ExportToPDF(MyFolders(1) & "Vystup.PDF", FALSE)
```

Ve výše uvedených příkazech se vymaže staré prostředí, nahraje se z dané složky nové, učiní se aktualizace prostředí, a nakonec se provede výstup ve formě PDF. Pro účely profukovací zkoušky hlavy motoru bude použito jiných grafů, a to zejména třírozměrných.

2.4.1. Základní vyobrazení rychlosti průtoku v axiálním směru

Po otevření nového prostředí je vhodné si vložit trojrozměrný graf z nabídky *3D Axis Systems* ► *3D Axis System from Above*, tedy zobrazení z vrchu. Pro úpravu velikosti grafu je dobré použít pravé tlačítko myši a přes *Display...* je možné si upravit rozměry, umístění, či například intervaly na osách a co a jak na nich bude vyobrazeno.



Obrázek 31: Úprava velikosti nebo umístění grafu a příslušných os

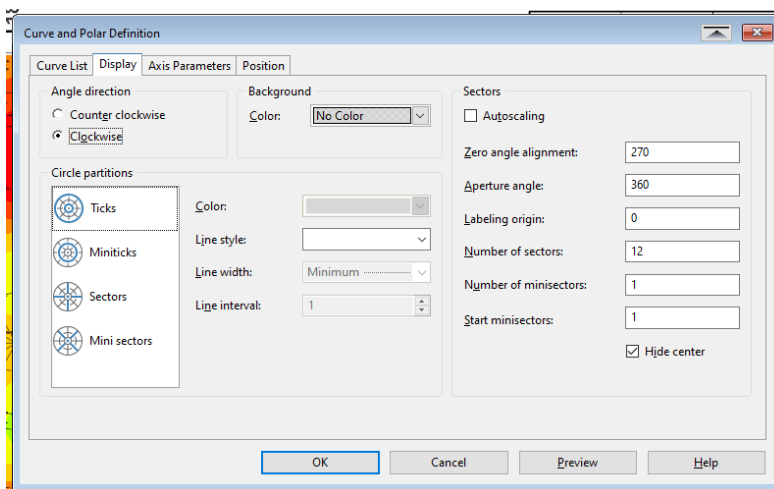
Na záložce *Curve List* je možné si přes znak + vložit typ grafu, přičemž zde zatím bude použit *Characteristic diagram*. V tomto grafu se pro *X* a *Y* kanály použijí přepočtené souřadnice a pro *Z* kanál zkoumaná veličina, zde axiální složka rychlosti.

Odkazy na kanály lze psát číselnými příkazy v hranatých závorkách například $[1]/[2]$, kdy dojde k otevření druhého kanálu z první skupiny. Druhou možností je vypsání přímo názvu skupiny a kanálu, tedy například *hlava1/x*. Ostatně tyto možnosti je možné kombinovat (například před lomítkem bude číslice v závorkách a za lomítkem přímo název skupiny).

Dále bude přidána polární soustava souřadnic přes *Polar Axis Systems* ► *Polar Axis System with 360° Aperture Angle*, tedy vyobrazení úhlů, které je vhodné zobrazit zejména u experimentálních měření. Na záložce *Display* je možno upravit zobrazení a rozmístění

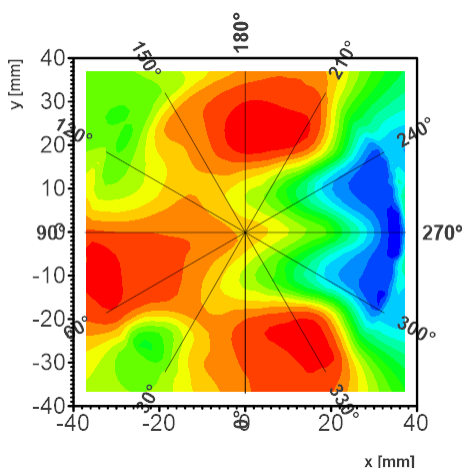


jednotlivých segmentů a na záložce *Axis Parameters* upravit popisky os či přidat odkazy na určité texty (viz obrázek 32).



Obrázek 32: Úprava polárního souřadnicového systému

V této chvíli je připraven graf s osami rozměrů zkoumaného prostoru, v tomto případě válce motoru, jsou dány popisky os a vyobrazeny úhly.



Obrázek 33: První zobrazení grafu axiální rychlosti

V grafu na obrázku však není vymezen prostor válce a není známa velikost zkoumané veličiny, to se však dá snadno napravit.

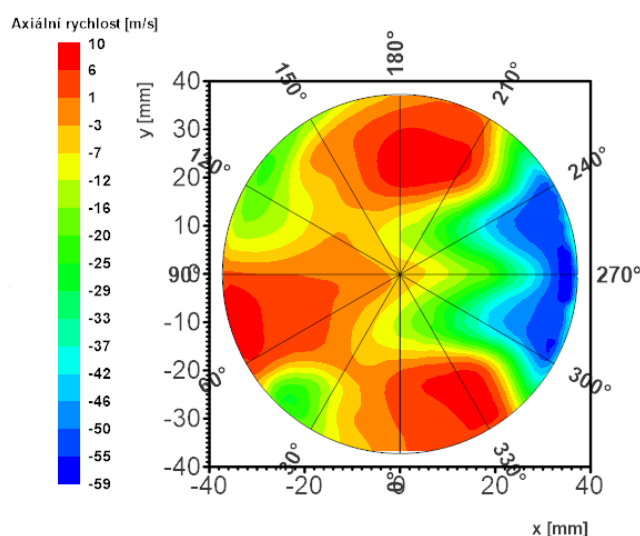
Jako první bude vymezen válec (kruhový řez), což lze v *Curve List* a přes tři tečky s popisem *Set Curve Parameters* se přes záložku *Boundary Curve* načte okrajová křivka. Vymezit okraj lze i tak, že program si podle dat sám křivku určí, přičemž v tomto případě vychází křivka dosti kostrbatě. Příhodnější je tedy možnost určení křivky z předem nahrané skupiny okrajové kružnice.



Je potřeba vybrat možnost, aby se oblast touto křivkou uzavřela, tedy vybráním možnosti *Clip to boundary curve*. Další možnosti jako tloušťka čáry či její barva je už možno měnit dle libosti.

Po každé úpravě se dají zobrazit změny (bez uložení) tlačítkem *Preview*. Pro přehled o velikosti zobrazované veličiny je potřeba zobrazit barevnou legendu, jež je k dispozici po zaškrtnutí *Color Legend* opět v záložce *Curve List*. Rozkliknutím *Settings...* u *Color Legend* je možné tuto legendu upravit a přizpůsobit svým potřebám.

Znovu je možnost zobrazit text, který odkazuje například na skupinu či kanál. Výsledkem okrajové křivky a barevné legendy může být tento graf na obrázku 34.



Obrázek 34: Příklad jednoduchého pohledu v grafu na axiální rychlost ve válci

2.4.2. Zobrazení izočár, umístění hodnot a max/min hodnoty

V předchozí kapitole byl uveden postup tvorby jednoduchého třírozměrného grafu vyobrazujícího axiální rychlost průtoku. Pro kvalitnější analýzu je přesto možné využít další možnosti jako jsou izočáry, či přehled, v kterých místech byl měřen, respektive počítán průtok.

Izočáry mohou být vloženy a spravovány opět přes tři tečky v *Curve Parameters* ► *Isolines*. Je třeba zaškrtnout *Isolines* a v záložce *General* zaškrtnout ve zdroji pro hodnoty jejich automatické určení – *Determine automatically*.

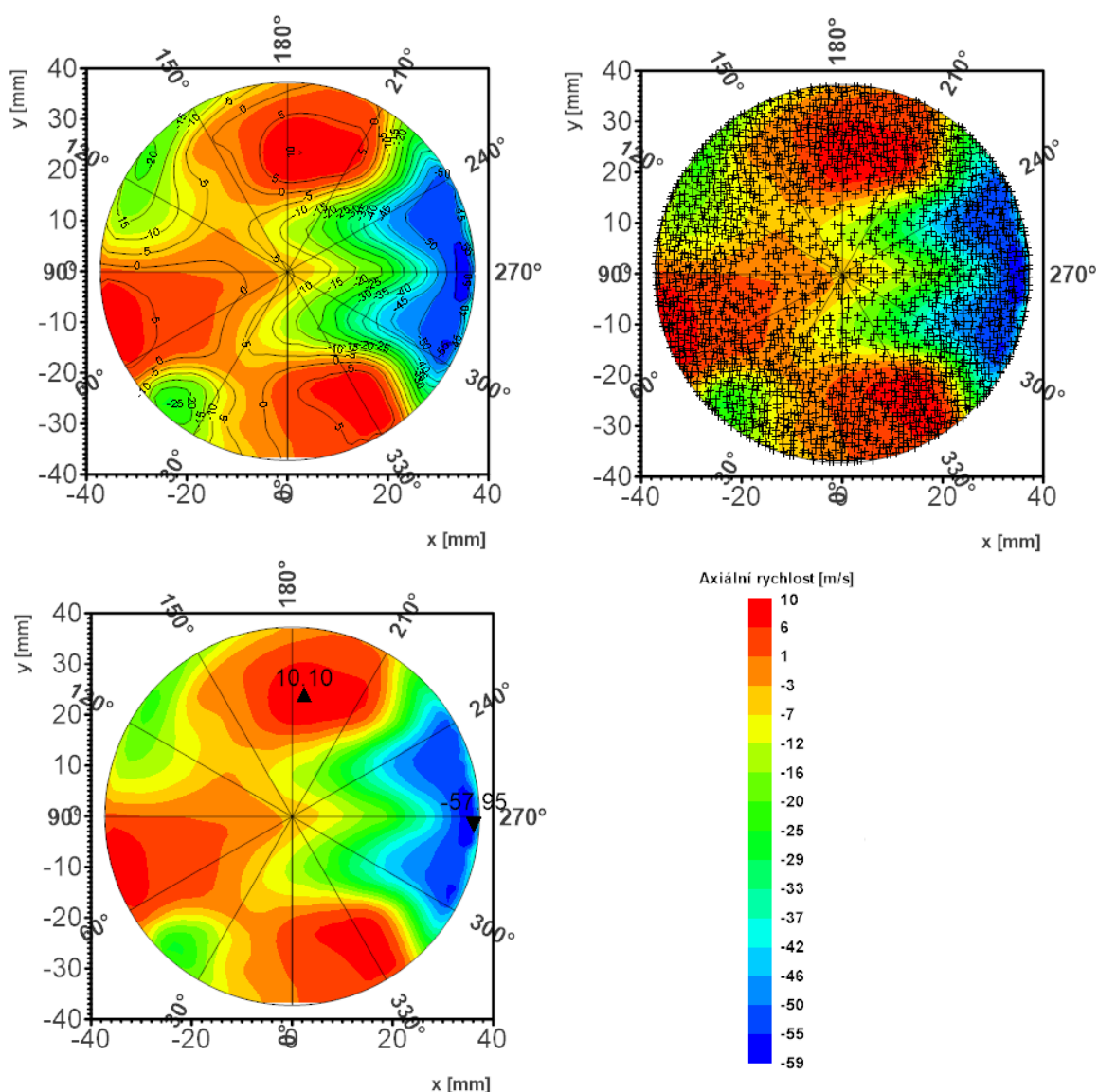
Na dalších záložkách je možné vložit číselné hodnoty izočár, jejich četnost, vzhled a velikost čar a písma a jiné. Přičemž po zaškrtnutí *Extended isolines* se zpřístupní záložka s rozšířenými možnostmi, kde je možno určit izočáry vyhrazené pro dané hodnoty s jinými tvary a barvami, pokud je potřeba.



Na záložkách jako *Surface*, *Interpolation* či *Hyperbola* není třeba momentálně nic měnit a zbývá tedy jediná přístupná záložka *Points*. Na ní je možno zobrazit měřené body či vypočtené hodnoty (i jejich umístění), ale taktéž maximální či minimální hodnoty, což je velice přínosné.

V případě simulací je zobrazení vypočtených bodů spíše na škodu, jelikož není poté v podstatě přes velký počet bodů nic vidět. Co se týká experimentů, tam je možné měřená místa zobrazit, jelikož jich bývá o dost méně. Četnost hodnot je možné zredukovat pro zlepšenou přehlednost (u simulací jich přesto bude mnoho). Redukci lze provést zaškrtnutím *Data reduction values* (není tak potřeba upravovat nahrávaný soubor).

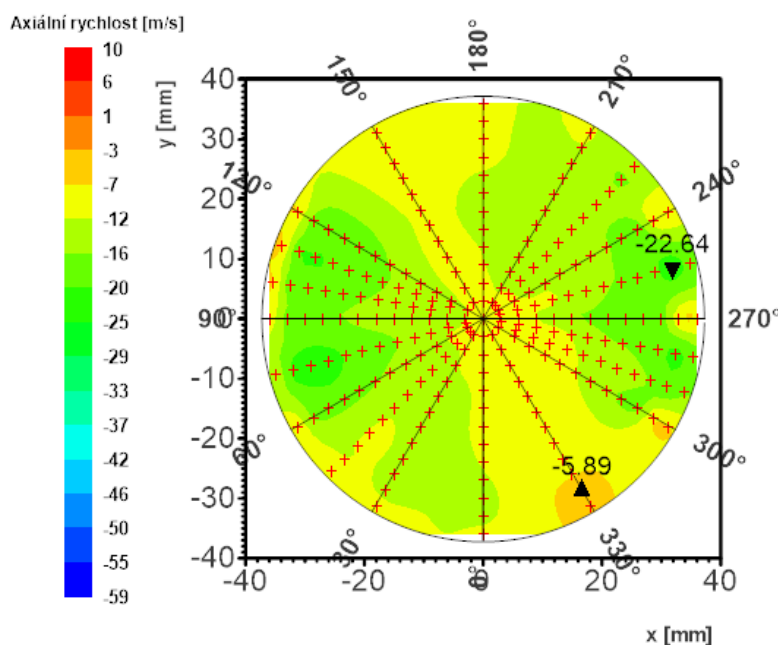
Dále je možnost upravovat velikost textu, barvy a symboly označující tyto body, či jestli budou vypisovány hodnoty, nebo jenom zobrazeny symboly (viz obrázek 35).



Obrázek 35: Zleva zhora: příklad zobrazení izočár, vypočítaných míst a min/max hodnot



Bylo zpozorováno, že v některých případech je potřeba označit redukci dat pro správné zobrazení maximálních a minimálních hodnot. Ty dokonce někdy nejdou s některými funkcemi dohromady a ukazuje se například jenom minimální hodnota, tudíž je potřeba s tímto počítat. Lepší zobrazení daných bodů (zde z experimentu HWA) je v grafu na obrázku 36, přičemž je vidět, že poměrně velké plochy jsou dopočítány, respektive aproximovány programem.



Obrázek 36: Příklad zobrazení měřených bodů v experimentu HWA

Ovlivnit, jak program data dopočítává, tedy jak interpoluje zde plochu danou rychlostmi, lze opět v záložce *Curve List* ► *Data Check*. Vyskočí okno, ve kterém je možné zadat jak relativní (odchylka v procentech z celé oblasti) či absolutní (odchylka z dílčích hodnot) toleranci zobrazovaných hodnot, tak také například upravit jejich rozmezí. Uživatel musí s tímto pracovat opatrně a sledovat, jak se chová graf při změnách tolerancí, jelikož se velice nelineárně poté mění celý graf.

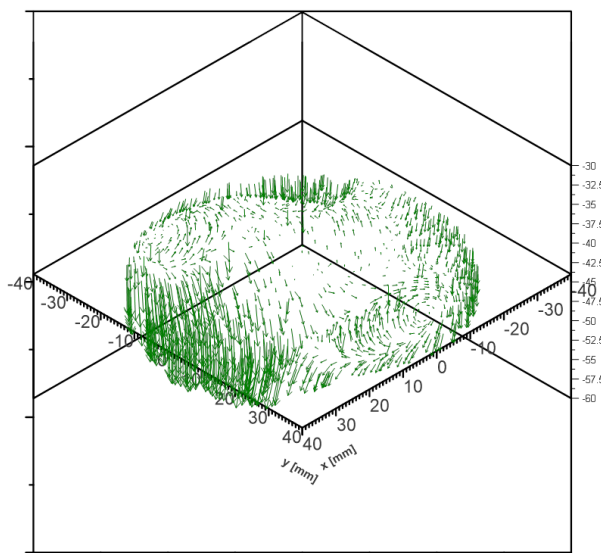
Mimochodem úhlové osy zde zobrazené přímo kopírují souřadnice dané měřením HWA metodou.

2.4.3. Zobrazení vektorů rychlosti

V této kapitole bude nahlíženo na postup zobrazení vektorů rychlosti daných profukovací zkouškou hlavy motoru. Ve skriptovací části byly vektory předem napočítány v tom smyslu, že byly určeny čtyři souřadnice dvourozměrných vektorů, přičemž dvě souřadnice vymezují počátek a další dvě konec vektoru.

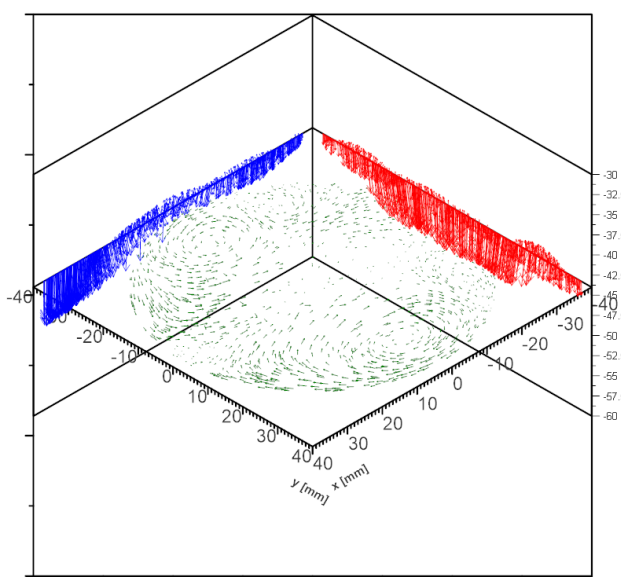


Nabízí se možnost zachycení vektorového pole v třírozměrném prostoru, ale jak je vidno na obrázku 37, tak takovéto zobrazení není vůbec přehledné, tudíž je zbytečné. Různým pootáčením souřadného systému se situace nijak nelepší.



Obrázek 37: Vektory v třírozměrném prostoru

Jinou variantou je zobrazení vektorových polí v jednotlivých plochách. Na obrázku 38 je toho následovně využito tak, že plocha xy je zelená, xz je modrá a yz červená.

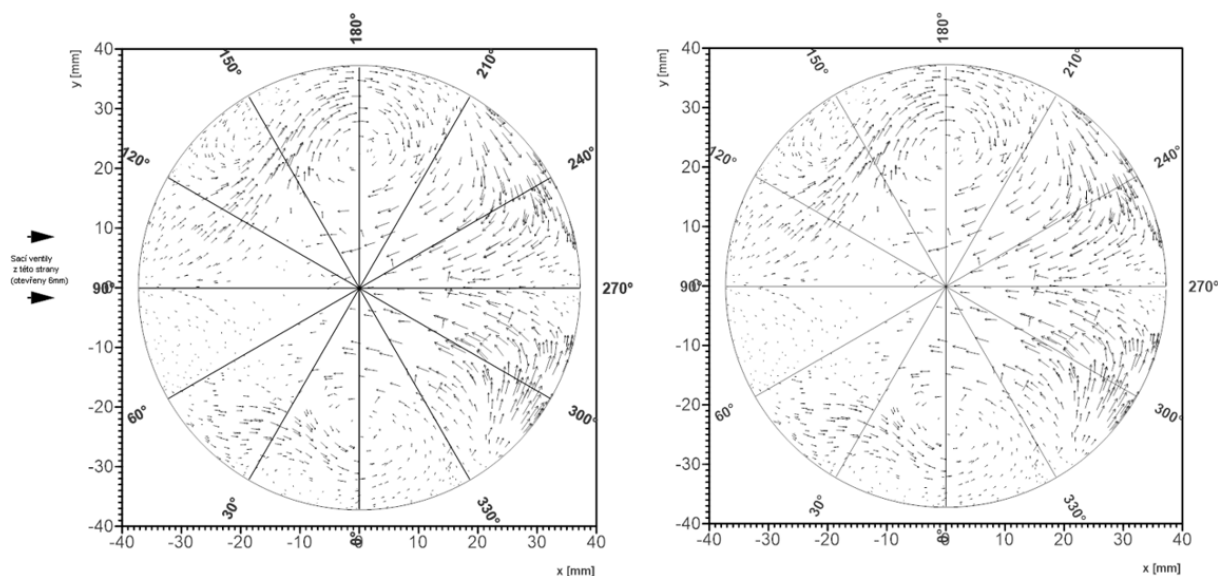


Obrázek 38: Zobrazení vektorů v jednotlivých plochách

Zde je rozložení vektorů do jednotlivých ploch daleko přehlednější, přičemž v plochách xz a yz jsou pouze specifické 2D grafy s konstantní souřadnicí (zde y a x). Vzniklé vektorové pole tak zobrazuje všechny složky v jedné ploše beze změny jedné souřadnice. Nejprínosnějším zobrazením je tak rovina xy s průmětem vektorů rychlosti.

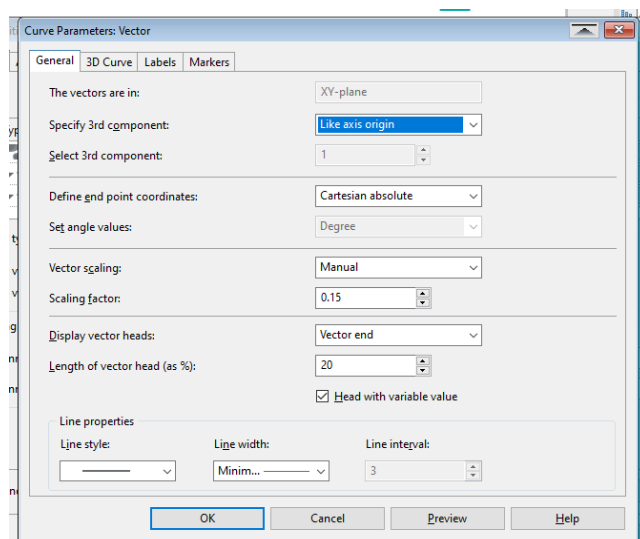


Dále bude následovat postup tvorby plošného vektorového pole v ploše xy . V záložce *Curve List* se přidá nové grafické pole *Vector*, kde se zapíšou počáteční a konečné souřadnice připravených vektorů, určí se dvou či třírozměrná pole a barva vektorů. Na obrázku 39 je srovnání při zadání dvou a tříložkových vektorů, přičemž je vidět, že dle očekávání vychází stejně – složka W se v ploše xy nemůže zobrazit.



Obrázek 39: Vlevo vektory U a V složek, vpravo vektory U , V a W složek

Díky této skutečnosti je možnost vypustit počítání třetích rozměrů vektorového pole. Po rozkliknutí tří teček je možnost nastavovat vektorová pole, přičemž pro dvourozměrné uspořádání v ploše xy je vhodné mít nastavení podobné tomu na obrázku 40.



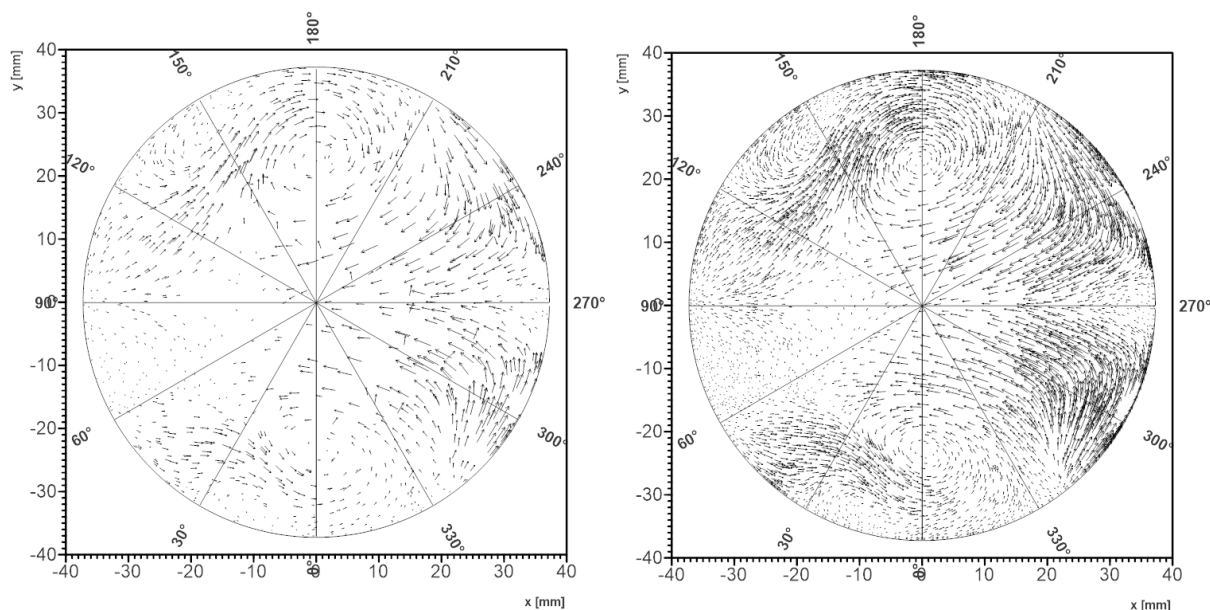
Obrázek 40: Nastavení vektorového pole

To znamená nechat třetí komponentu dle zdrojové osy, souřadnice nechat kartézské absolutní a ostatní už víceméně dle uvážení či vhodnosti zobrazení pro každého uživatele.



V tomto případě však byly nastaveny manuálně velikosti vektorů se zmenšením na 0,15krát původní velikosti a upraven konec s variabilní velikostí šipky vektoru.

Nikde se však nedá určit počet zobrazovaných vektorů, proto je ve skriptu počet načtených dat omezen na dané maximum. To znamená, že program ze souboru dat nahraje, například při počtu 10 tisíc dat se zadáním intervalu 1 tisíc, každou desátou hodnotu. Na obrázku 41 je tak zachycen pohled na vektory ze všech dat a z jejich zmenšeného intervalu.

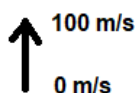


Obrázek 41: Srovnání vektorových polí vlevo ze 2 tisíc a vpravo ze 7 tisíc hodnot

Interval je možné určit jakýkoliv, například pro jemnější zobrazení vektorového pole. Větší počet dat například v počtu 7 tisíc hodnot (jako zde na obrázku) je pořád ještě přijatelný, přičemž při hodnotách přes 30 tisíc už je to nepoužitelné.

Co se týká informace o velikosti rychlosti daného vektoru, tam je nutné použít například barevné pole celkové rychlosti v ploše určené ze složek rychlosti U a V.

DIAdem možnost zobrazení příslušné legendy neumožňuje, a tak je potřeba si pomoci jinými způsoby. Jiný způsob než barevné mapy může být přidání nové hodnoty vektoru určujícího například velikost 100 m/s a může být umístěn mimo graf, například vpravo od něj (viz obrázek 42). Vektor by například mohl být takto zadán přímo do datového souboru. Tato možnost je na rozdíl od barevné mapy však velice komplikovaná a nepomáhající.



Obrázek 42: Myšlený vzorový vektor znázorňující poměrovou velikost rychlosti



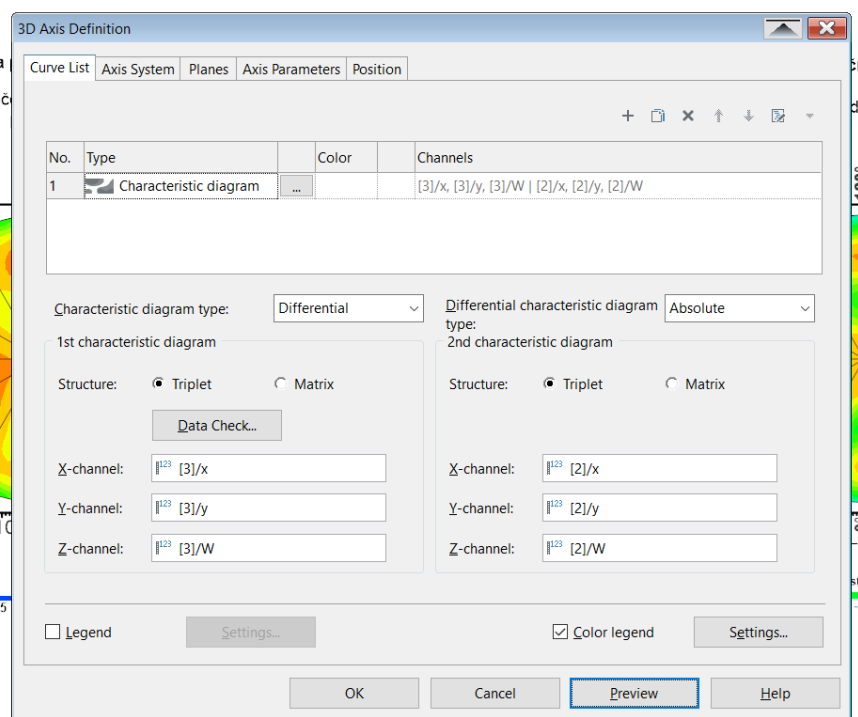
Barevnou mapu s vektory bude možno vidět v kapitole se sestavenými vzorovými grafickými výstupy.

2.4.4. Zobrazení rozdílových map – difference grafů

Velice užitečným nástrojem je přímé srovnání dvou grafů tak, že se od sebe odečtou barevné mapy těchto dvou grafů. Tohoto rozdílu je dosaženo diferencí výsledných aproximovaných barevných map, nikoli odečítáním přímých hodnot.

Tento způsob diferencování map velice elegantně řeší problém s jiným počtem dat zejména při srovnávání simulace a experimentu. Na druhou stranu je toto srovnání (například u modelů s hrubou sítí) velice orientační, a nelze tak na toto porovnání plně spoléhat.

Postup tvorby difference je poměrně jednoduchý. V záložce *Curve List* (viz obrázek 43) je potřeba změnit typ charakteristického diagramu na *Differential* a typ tohoto grafu na *Absolute*, tedy absolutní rozdílovou mapu. Rozdíl grafů je zde myšlen zprava doleva, v tomto případě se tedy od pravého grafu (z druhé skupiny) odečítá levý graf (ze třetí skupiny).

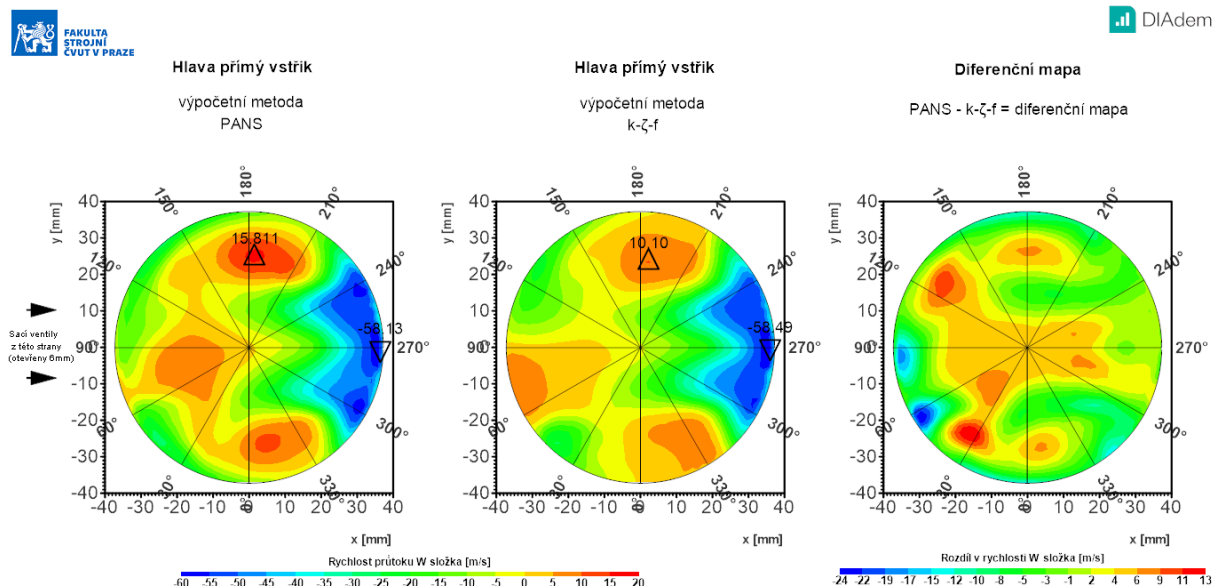


Obrázek 43: Nastavení diferenční mapy



Výsledkem je diferenční mapa s fixním intervalem vypočtených hodnot. Například na obrázku 44 jsou znázorněny mapy složky rychlosti W z výsledků simulace se dvěma výpočetními metodami (PANS a k - ζ - f) a jejich diferenční mapa.

Diference je tedy zde myšlena jako levý graf minus pravý graf, přičemž v nastavení je to naopak (pravý minus levý, na to je si nutné dávat pozor!).



Obrázek 44: Příklad zobrazení rychlostní složky W metodami PANS a k - ζ - f a diferenční mapy

2.4.5. Spojnicový graf, tabulka a různé detaily

Spojnicový graf může sloužit k rychlému srovnání výsledných hodnot, například vírových charakteristik tumble a swirl při proměnném otevření ventilů. Tvorba samotného spojnicového grafu, tedy grafu daného dvojicí souřadnic, není nijak složitá.

Získání dvojice souřadnic už je však složitější, jelikož jsou zde každá tumble a swirl čísla vypočtena zvlášť, a tak je nelze jednoduše spojit dohromady. Ke spojení hodnot z více kanálů do jednoho slouží funkce *Concatenate*, přičemž názvy kanálů a skupin musí být stejné, a skupiny musí mít pouze jiné indexování. Funkci *Concatenate* tak nelze použít, a proto je třeba hodnoty jednotlivě kopírovat do společného kanálu pomocí skriptu.

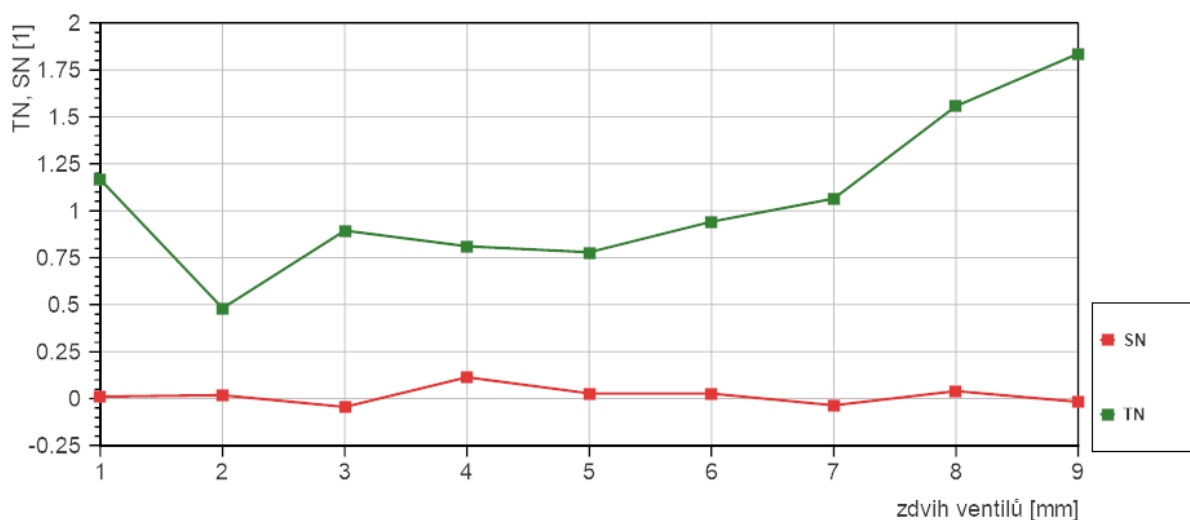
Výhodou je daleko lepší přehled nad pohybem dat oproti složitějším a nic neříkajícím příkazům, které ale naopak obsahují méně textu. Příkazy jsou tedy nahrazeny jednoduchými příkazy kopírovat a vložit, kdy se číselně určí pozice, na kterou se jednotlivé hodnoty kopírují.



```
'Přesun hodnot z několika skupin do jedné  
'TN 2 ventily  
Call DataBlClpCopy("2Valves/TN_1mm", 1, 1)  
Call DataBlClpPaste("2Valves/TNall", 1, 0, True)  
Call DataBlClpCopy("2Valves/TN_2mm", 1, 1)  
Call DataBlClpPaste("2Valves/TNall", 2, 0, True)  
Call DataBlClpCopy("2Valves/TN_3mm", 1, 1)  
Call DataBlClpPaste("2Valves/TNall", 3, 0, True)
```

Ve skriptu výše je pohled na příkazy přesouvající tumble čísla pro zdvihy ventilů 1, 2 a 3 mm. Tato možnost je tedy zjednodušenou verzí funkce *Concatenate*. Touto formou lze kopírovat i několik hodnot najednou tak, že se změní druhá hodnota uvedená v závorce na požadovaný počet hodnot. První hodnota značí umístění výchozí hodnoty.

Na obrázku 45 je zobrazen příklad průběhu tumble a swirl čísel v závislosti na velikosti zdvihu ventilů. Jedná se o průběh na hlavě motoru s nepřímým vstřikováním paliva s otevřením obou sacích ventilů.

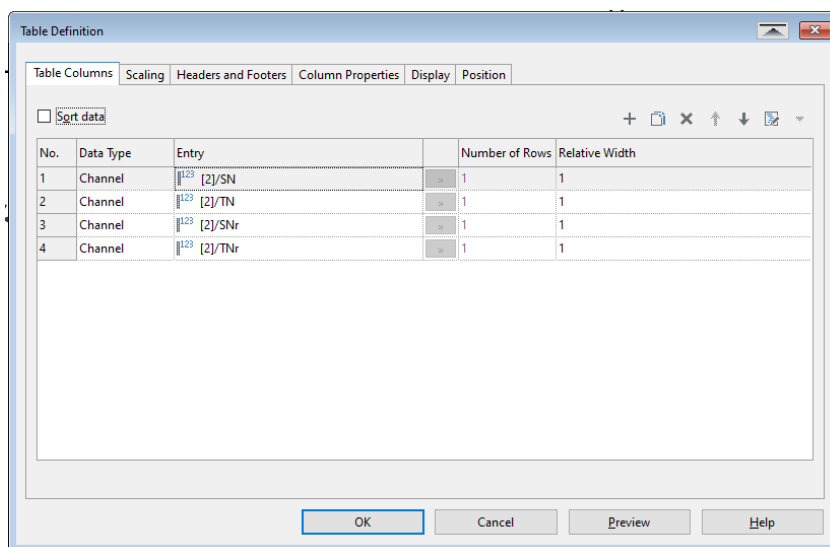


Obrázek 45: Příklad průběhu tumble a swirl čísel na hlavě s nepřímým vstřikováním paliva

Nastavení grafu je obdobné již uvedenému třírozměrnému grafu, přičemž jsou zde upraveny osy a je přidána mřížka.

Co se týká různých detailů jako jsou popisky, tak u nich je možné použít například odkazovacích příkazů. Takto je možné vypsát názvy skupin nebo kanálů, zejména pro dosažení větší automatizace procesu vyhodnocení dat. Obrázky je možné jednoduše zkopírovat a vložit klávesovými zkratkami nebo přes panel *Insert*.

Tvorba tabulky pro vypsání jednotlivých charakteristických hodnot je možná jednoduchou formou pod ikonkou *Tables* ► *2D Tables* ► *2D Table with Horizontal and Vertical Separators*. V záložce *Table Columns* se vloží vstupní hodnoty, které se mají zobrazovat, přičemž jsou typu *Channel* (viz obrázek 46).



Obrázek 46: Nastavení tabulky

V záložce *Scaling* se nastaví *Automatically increasing Table Length* a 3 % výška stránky. V nastavení hlavičky a zápatí už jenom nastavit zobrazení názvu kanálu a ostatní věci nechat automaticky, nebo změnit dle vlastního uvážení, jelikož software zobrazuje například malá písmena a čísla (v závislosti na hodnotě). Výsledkem může tak být tabulka na obrázku 47.

SN	TN	SNr	TNr
0.0264	2.86	0.0246	2.67

Obrázek 47: Příklad tabulky s tumble a swirl čísly včetně redukováných čísel

V případě nejasností je vždy možné nahlédnout do nastavení ve vzorech, či pod tlačítkem *Help* se dá ve většině případu vše dohledat.

2.5. Sestavené vzorové skripty – 3 typy

Sestavené skripty obsahují již uvedené příkazy, proto zde budou pouze stručně popsány jejich funkce. Lze si je upravit dle libosti či je překombinovat na jiný typ dat, obecně tedy budou k dispozici tři typy skriptů.

První typ slouží pro jednoduché porovnání dvou datových sad, například tedy hlavy s přímým a nepřímým vstřikováním paliva, přičemž je možnost jednoduše přidat či odebrat další data.

U druhého typu vzoru bude možnost vyhodnocovat data například ze 6 simulací a 2 experimentů, kdy u dvou simulovaných hlav budou dvě výpočetní metody. Součástí příkladu je také 1 simulace s hrubou sítí a 1 simulace se skutečnými rozměry (nezmenšené



na rozměry hlavy s přímým vstřikem). K těmto 6 simulacím tedy přibudou ještě 2 experimentální měření, a to HWA metodou na hlavě s přímým vstřikováním paliva a PIV metodou na hlavě s nepřímým vstřikováním paliva.

Součástí také bude skript jenom pro 6 simulací, kdy bude možné skript kombinovat s experimenty nebo dalšími simulacemi.

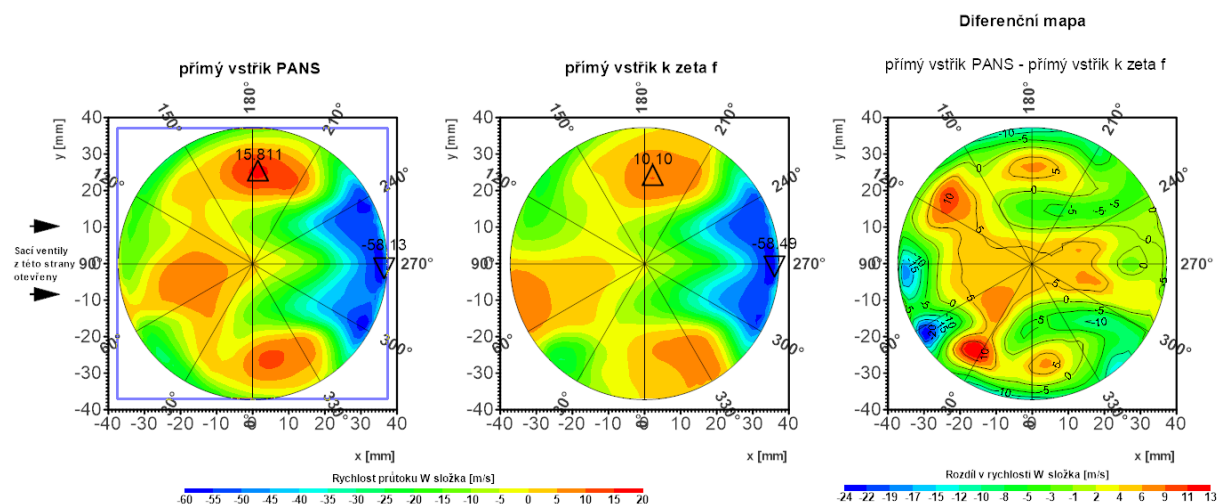
Třetí typ poté srovnává měnící se charakteristiky při proměnném otevření ventilů. Jedná se tedy o stejný model, kde se mění jak zdvih ventilů, tak i počet otevřených ventilů.

2.6. Sestavené vzorové grafické výstupy

Ke každému z uvedených typových skriptů přísluší grafický výstup tak, aby se zobrazily všechny relevantní hodnoty určené k analýze, případně je možné si cokoliv posléze upravit.

2.6.1. Základní grafické srovnání simulací dvou hlav motorů

Sestavený grafický výstup ze srovnání dvou hlav motorů, případně dvou výpočetních metod jedné hlavy, může sloužit k rychlému a jednoduchému srovnání. K dispozici je srovnání základních veličin jako jsou axiální rychlost či rychlost v ploše. Na obrázku 48 je vidět příklad srovnání hlav s přímým vstřikováním (hlava typu 1) a dvou výpočetních metod PANS a k- ζ -f.

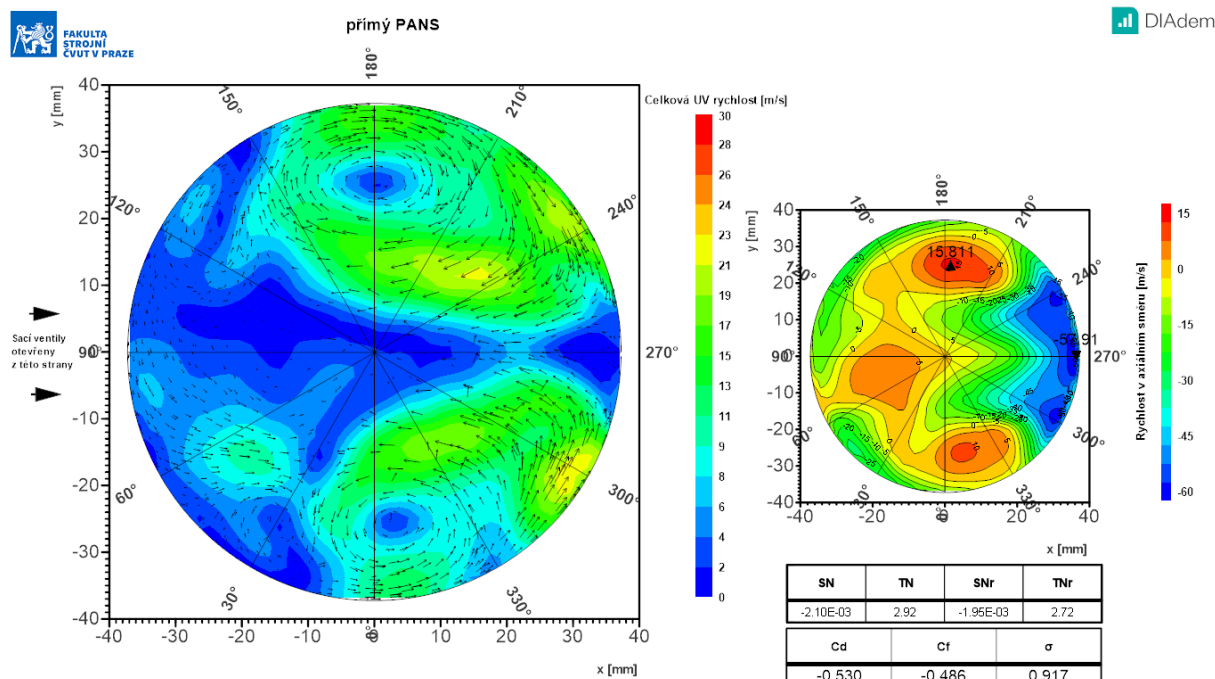


Obrázek 48: Příklad srovnání axiálních rychlostí dvou výpočetních metod

Jednalo se o stejné výpočetní sítě, ale jiné výpočetní metody. V diferenční mapě je poté možné vidět rozdíl ve výpočtech.

Dalším možným výstupem je srovnání celkové rychlosti kolmé k axiální, která je mimo jiné zobrazená pomocí vektorů. Dále se také srovnávají například toky, vírové charakteristiky

či průtokové vlastnosti, jenž je možné zobrazit například u jednotlivých simulací zvlášť jako na obrázku 49.



Obrázek 49: Příklad pohledu na jednotlivou simulaci a vlastnosti

Zde jsou vypsány tyto hodnoty v tabulce, ale například u proměnného zdvihu ventilů je hodnot více a lze je tak přehledně uspořádat do grafu. V takovéto podobě to bude možné vidět dále v textu. Součástí tohoto vzoru je tak v základní konfiguraci srovnání dvou hlav, průtočných rychlostí, toků, vírových a průtočných vlastností.

2.6.2. Grafické srovnání simulací a experimentů

V tomto případě se jedná o obdobné srovnání, přičemž zde je 6 různých simulací 2 zkoumaných hlav. K simulacím se přidávají 2 experimenty, HWA k přímému vstříku a PIV k nepřímému vstříku.

U experimentů je méně informací o průtoku (některé veličiny nelze běžně získat na rozdíl od rozsáhlejších simulací), tudíž v těchto případech se jedná pouze o vizualizaci určitých veličin. U této HWA metody lze tak díky měření zobrazit jen dvě složky rychlosti – axiální a tangenciální.

U PIV metody zase obecně nelze měřit složku směřující ke kameře, a tak lze pouze zobrazit rychlost radiální a tangenciální. Díky chybějícím složkám rychlosti a jiným veličinám tak nelze zcela vyhodnotit vírové a průtokové charakteristiky.

Simulace 6

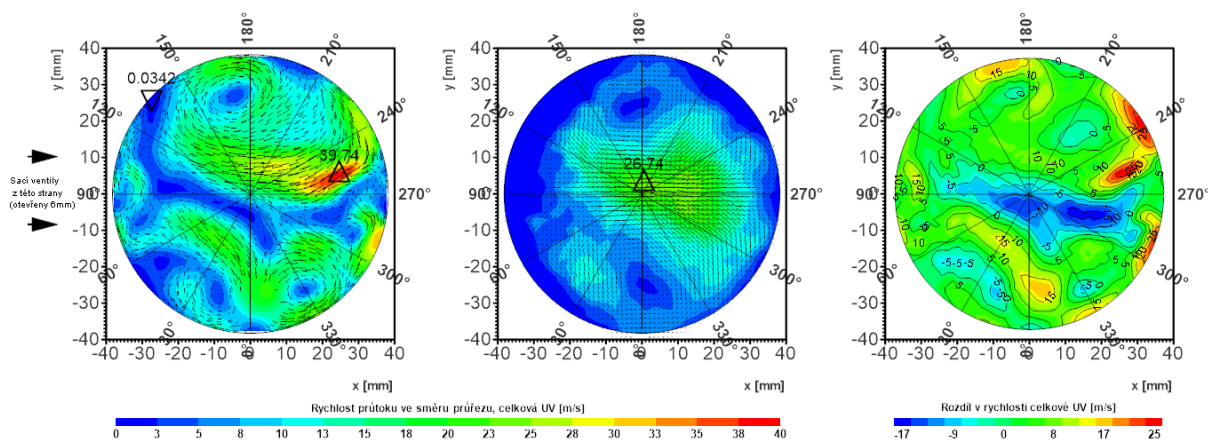
nepřímý vstřík PANS skutečné rozměry

Experiment 8

exp_PIV

Simulace 6 - experiment 8 = diferenční mapa

nepřímý vstřík PANS skutečné rozměry - exp_PIV



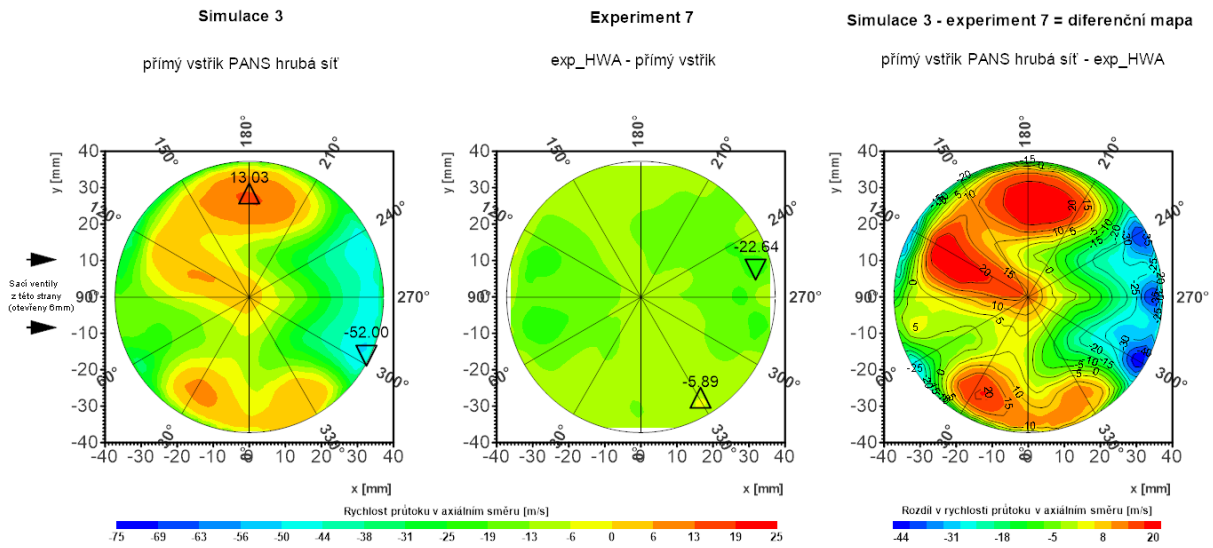
Obrázek 50: Příklad srovnání simulace a PIV experimentu u hlavy s nepřímým vstříkem

Na obrázku 50 je příklad srovnání PIV experimentu a simulace provedené metodou PANS, přičemž se jedná o hlavu s nepřímým vstříkáváním (hlava typu 2). Simulační výsledky obsahují přes 30 tisíc hodnot a experimentální přes 7 tisíc. U těchto výsledků je možnost poměrně dobré analýzy, přesto je vždy nutné počítat s jistou chybou.

Srovnávanou veličinou je zde celková rychlost v ploše, tedy výsledná rychlost z tangenciální a radiální složky. Tyto složky jsou zde doplněny o vektorová pole. Rozdíl těchto grafů ve formě diferenční mapy využívá pro přehlednost izočáry, které lze vypnout.

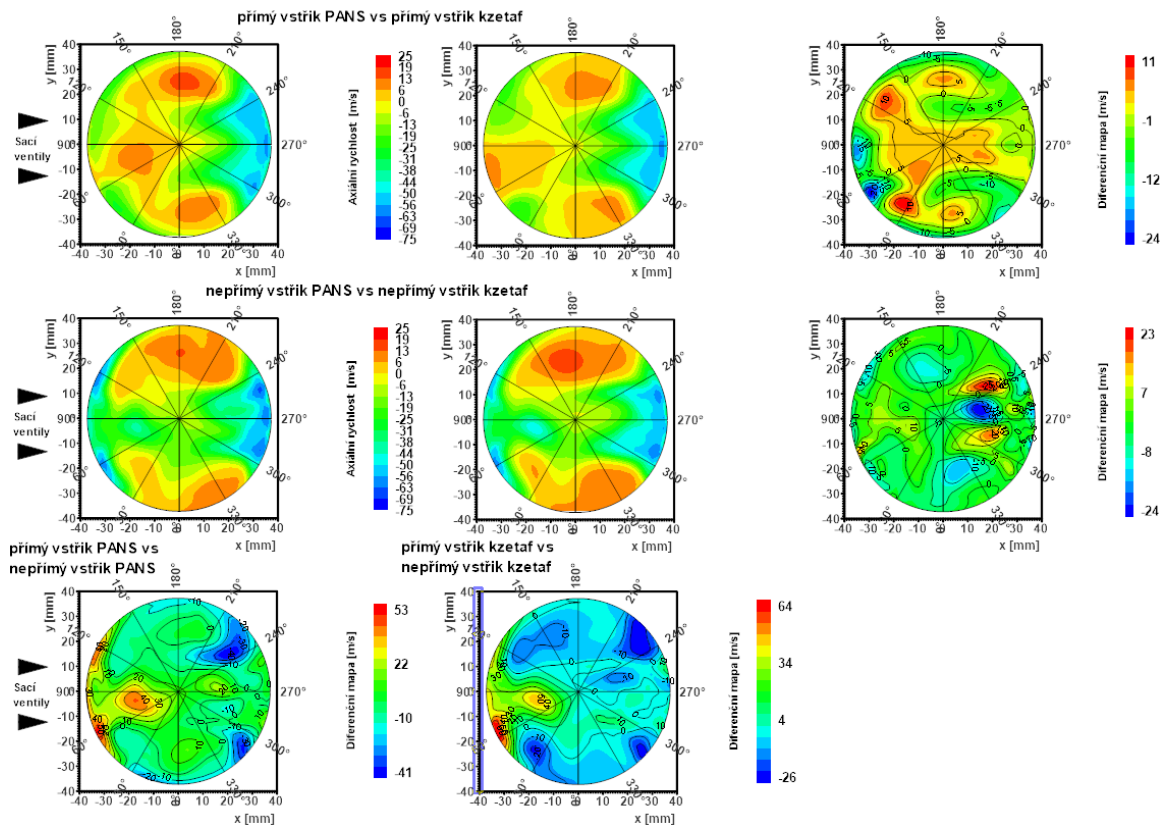
Na obrázku 51 je naopak příklad srovnání hlav typu 1. Jedná se o hrubou síť modelu spočteného metodou PANS a experiment vyhodnocený metodou HWA.

Použitím hrubé sítě se však stejně dosáhne velmi jednoduše téměř o řád vyššího počtu hodnot než u HWA metody. Chyba je zde u simulace i experimentu vyšší než u hlavy typu 2.



Obrázek 51: Příklad srovnání simulace a HWA experimentu u hlavy s přímým vstřikem

Na obrázku 52 je vidět příklad srovnání 4 simulací 2 typů hlav a 2 výpočetních metod, kdy vzniknou celkem 4 diferenční mapy ze stejných hlav motorů a ze stejných výpočetních metod.

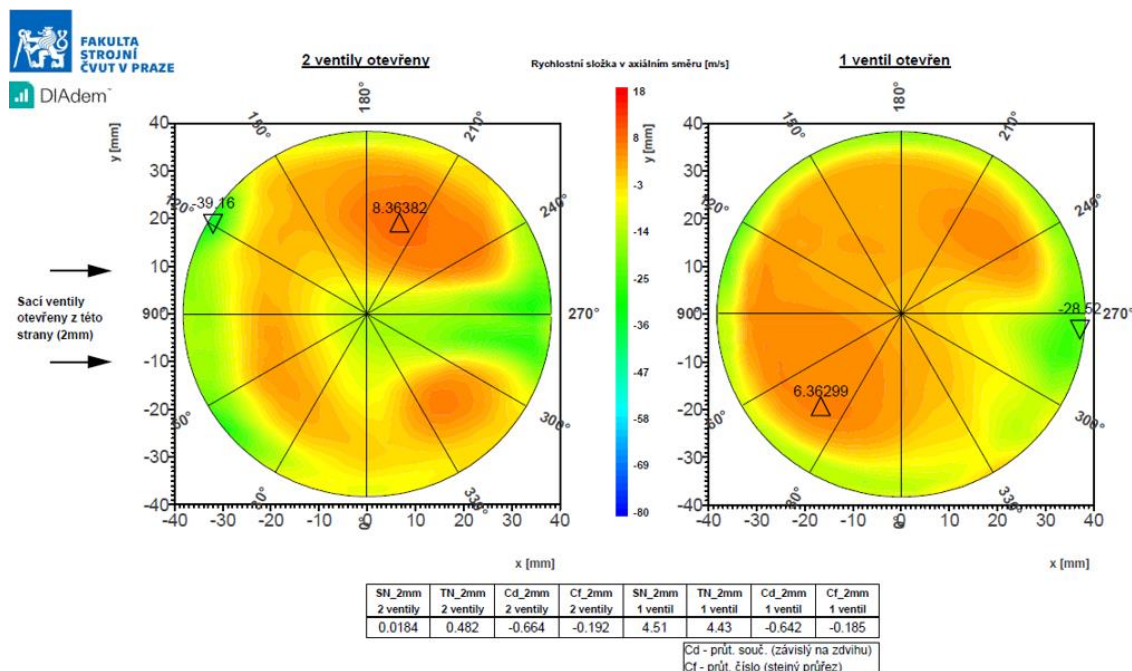


Obrázek 52: Příklad přímého srovnání 2 hlav a 2 výpočetních metod



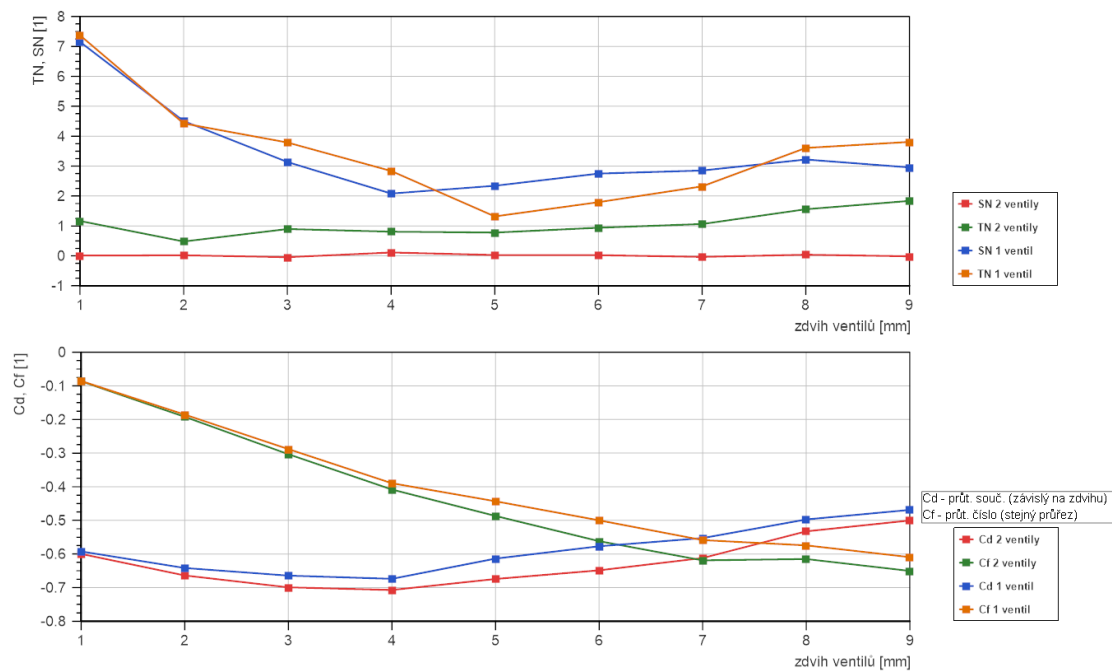
2.6.3. Grafické srovnání při proměnném zdvihu ventilů

V tomto případě je možné zobrazit proměnný zdvih ventilů či jiný libovolný proměnný parametr modelu. Zde v příkladu na obrázku 53 (snímek pro zdvih 2 mm) jsou vypsány vírové a průtokové charakteristiky a srovnání při otevření jednoho či dvou sacích ventilů.



Obrázek 53: Příklad výstupu z proměnného otevření ventilů – zdvih 2 mm

V tomto grafickém výstupu jsou taktéž přítomny grafy srovnávající vírové a průtokové charakteristiky v závislosti na proměnném zdvihu ventilů (viz obrázek 54).



Obrázek 54: Příklad grafu vírových a průtokových charakteristik měnících se v závislosti na zdvihu ventilů

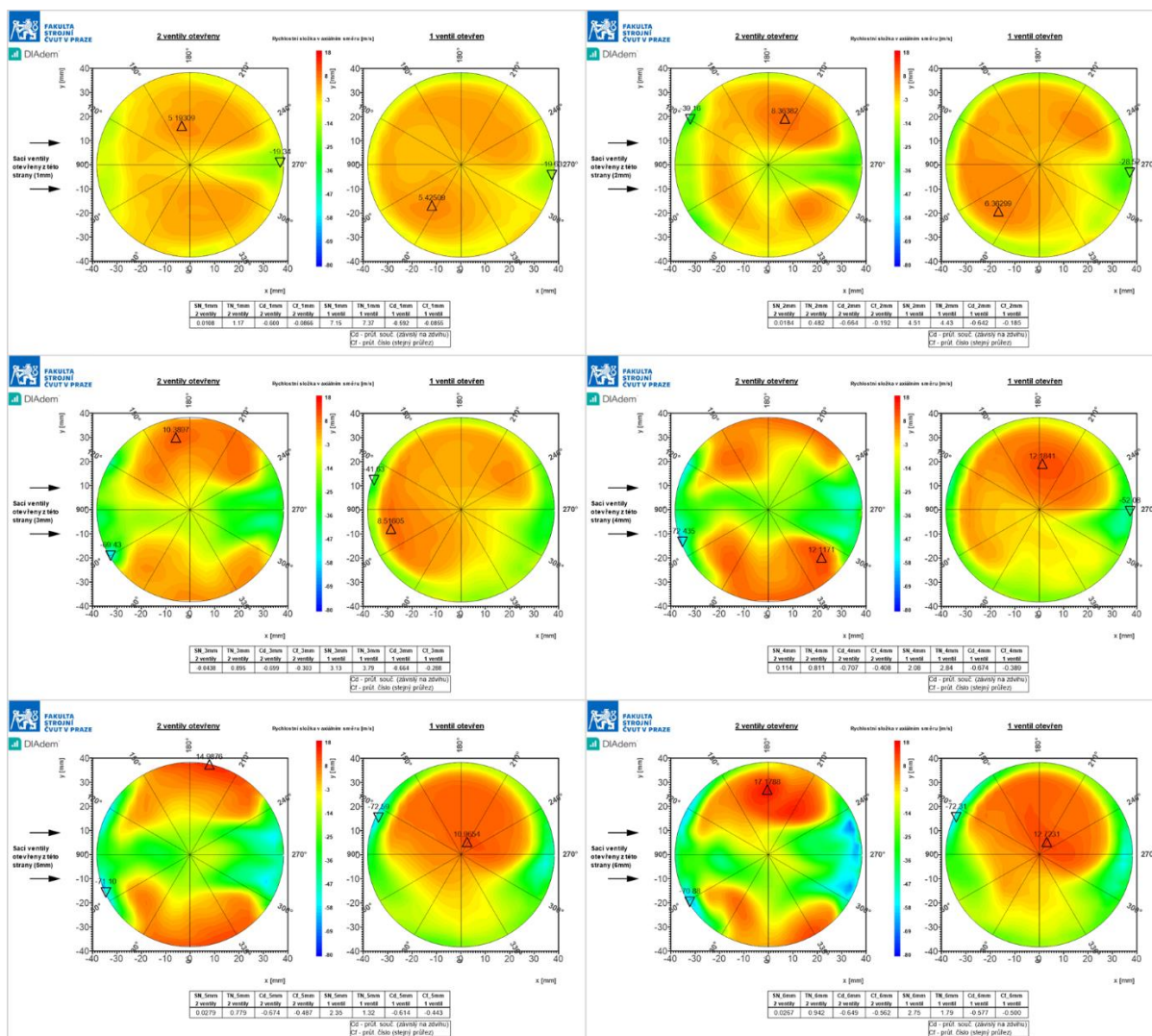


2.6.4. Animace

Vhodná forma zobrazení proměnných parametrů je pomocí animace či videa, která však není k dispozici v programu NI DIAdem. Nicméně, když si uživatel přeje vytvořit animované grafy, tak není nic jednoduššího než použít internetových funkcionalit.

Příkladem takové stránky je třeba *ezgif.com*, kde například ze souboru PDF lze vytvořit pohyblivý obrázek GIF, nebo rovnou video, které lze pozastavit v libovolné části. Po nahrání souboru lze animaci upravit, například oříznutím, přidáním textů či efektů, nebo úpravou prodlevy snímků tak, aby výsledný formát vyhovoval požadavkům.

Výsledkem může být například animace změny zdvihu ventilů v rozmezí 1 až 9 mm, zde znázorněna zobrazením jednotlivých snímků na obrázku 55 (zde 1–6 mm zdvihu). Uvedené vytvořené příklady budou v kompletní formě k dispozici v příloze včetně skriptů.



Obrázek 55: Snímky příkladu vizualizace proměnného zdvihu jednoho a obou ventilů, zde v rozmezí 1-6 mm

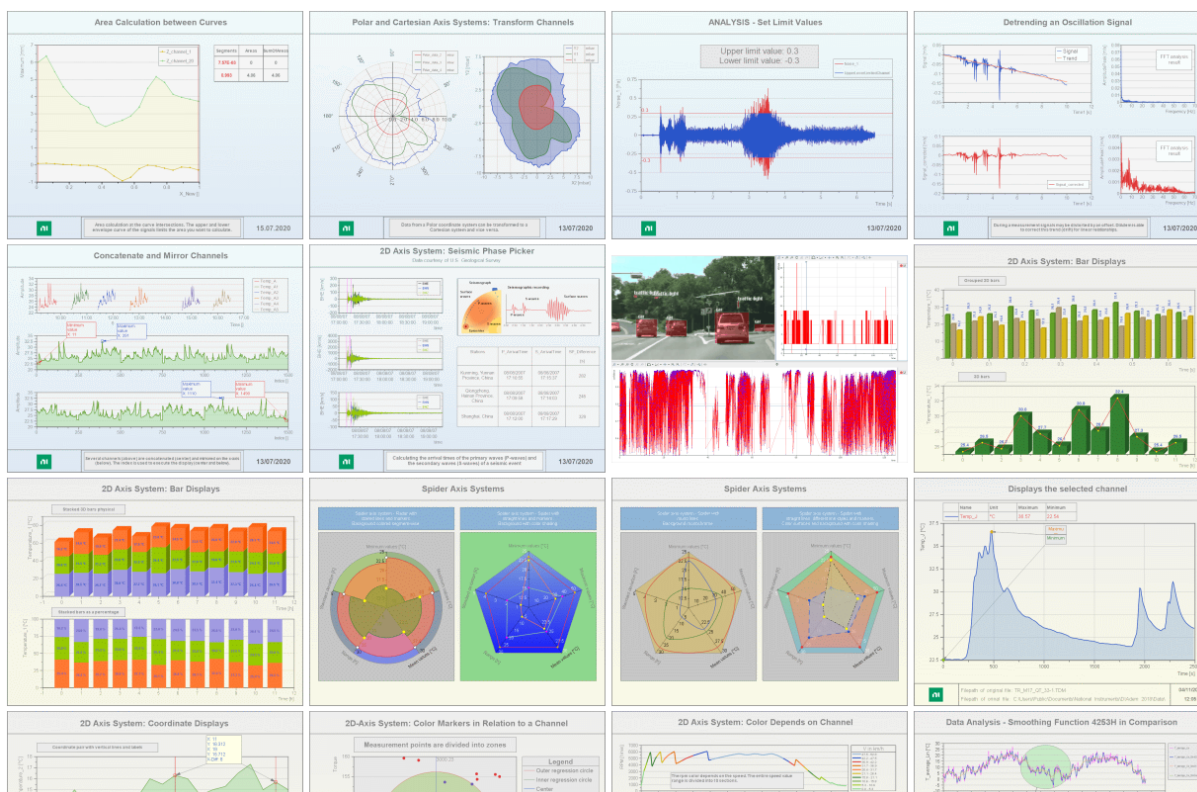


2.7. Zpracování a analýza jiného typu dat

Vypracované skripty a vizualizační prostředí lze modifikovat na různé profukovací zkoušky či podobná měření. Pokud je však potřeba provést analýzu jiného typu dat, než pro které byl tento text určen, je nutné nahlédnout do *Help* či návodů.

Program ve svých návodech nabízí přehled mnoha možností jak provést skript, či data zobrazit. Pokud datové soubory umožňují možnosti indexování, lze využít i možnosti opakování skriptů pomocí funkcí (jako jsou například *while* či *loop*). Skript taktéž umožňuje tvorbu grafických výstupů ve skriptovacím prostředí, což může být ale často mnohem složitější než interaktivní tvorba přímo v *REPORT*. Ostatně některé možnosti práce s programem jsou i v návodech na *youtube* přímo od vývojářů. Uživatel si například může nalézt funkci v záložce *ANALYSIS* a po jejím rozkliknutí přes tlačítko *Help* nahlédnout, jak funkce vlastně pracuje.

Dobrým příkladem je nalezení vhodného příkladu v programu (*Help* ► *Example*), který bude vyhovovat nebo částečně odpovídat požadavkům na vytyčená data. *Example* si lze potom různě upravit dle libosti. Samozřejmostí je zvolení správného datového formátu a případné úpravy dat ve skriptu. Teoreticky při potřebě zobrazení hotových dat lze mít skript pouze na nahrání dat a vizualizačního prostředí, které se poté přeneso například do formy PDF.



Obrázek 56: Krátký přehled exemplů v NI DIAdem



3. Závěr

Na počátku textu byla popsána daná profukovací zkouška hlavy motoru prováděná experimentálně v laboratoři a simulačně na počítači. Byly zkoumány dvě hlavy motorů, jedna s přímým vstřikováním a druhá s nepřímým vstřikováním. U obou hlav byly provedeny simulace dvěma výpočetními metodami (PANS a k- ζ -f), přičemž co se jedná experimentů, tam byla u každé hlavy použita pouze jedna měřicí metoda. U hlavy s přímým vstřikem byla použita HWA metoda a u hlavy s nepřímým vstřikem PIV metoda.

Převážná část této práce byla však věnována možnostem zpracování výsledků simulace, experimentální měření totiž většinou neumožňují tak rozsáhlou analýzu, a to hlavně díky omezeným možnostem proveditelnosti měření. Nicméně experimentální měření je součástí příkladů srovnávání, kde se setkávají přímo experimenty a simulace, a vyhodnocují se jejich rozdíly. U hlavy s nepřímým vstřikem je vidět podobnost simulace a experimentu, zejména díky velkému počtu vyhodnocených dat, ale také díky PIV metodě neovlivňující průtok.

Závěrem tak lze říct, že byl proveden postup tvorby datové analýzy na profukovací zkoušce hlavy motoru. K vhodnému zpracování byl po širokém rozboru vybrán SW DIAdem, v němž se snadno pracuje, ale hlavně umožňuje nahrávání makra.

Značnou součástí této práce je totiž tvorba spustitelného skriptu. Nahrávání makra znamená, že uživatel pracuje v SW a program vepisuje tuto práci do spustitelného skriptu, přičemž při velkém počtu dat je žádoucí použití přímého odkazování v příkazech. Program poté načítá jiná data než uživatel požaduje, a hledání možných chyb tak program neusnadňuje.

Formátování dané programovacím jazykem je také nutné dodržovat, program uživatele ale většinou na takovou skutečnost upozorní. Hotový skript tak vytyčená data načte a zpracuje, až nakonec vyexportuje požadovaný grafický výstup, který je například ve formátu PDF.

Nabízí se samozřejmě i jiné cesty jak zpracovat data, například použitím jiného softwaru nebo jiným grafickým výstupem. Součástí rozboru v této práci byla i rešerše dalších programů (placených i dostupných zdarma), kdy v některých lze přímo tvořit animace.

NI DIAdem je ale oproti jiným SW možné spojit s nástrojem LABView, se kterým může dobře spolupracovat například při měření. Spojením těchto programů by tak mohla být dosažena značná automatizace procesu vyhodnocení dat z experimentů. Data ze simulací se ale budou muset stále exportovat zvlášť.



Seznam použité literatury

- [1] ŠIMÁNA K.: *Příprava experimentálního zařízení pro měření rychlosti ve válci termoanemometrickou sondou*. Praha, 2019. Bakalářská práce. FS ČVUT, Ústav automobilů, spalovacích motorů a kolejových vozidel.
- [2] URUBA V.: *Turbulence*. FS ČVUT v Praze, 2009. ISBN 978-80-01-05600-4.
- [3] MACEK J., KLIMENT V.: *Spalovací turbíny, turbodmychadla a ventilátory*. Nakladatelství ČVUT, 2006. ISBN 80-01-03529-8.
- [4] HOFMAN D.: *CFD simulace průtokových vlastností sacích kanálů hlavy válce v programu Creo*. Praha, 2020. Bakalářská práce. FS ČVUT, Ústav automobilů, spalovacích motorů a kolejových vozidel.
- [5] SUCHÝ M.: *Měření a vyhodnocení průtokových a vírových vlastností kanálů spalovacího motoru*. Praha, 2018. Bakalářská práce. FS ČVUT, Ústav automobilů, spalovacích motorů a kolejových vozidel.
- [6] VYŠTEIN O.: *Měření průtokových vlastností hlavy válců motoru EA211*. Praha, 2020. Bakalářská práce. FS ČVUT, Ústav automobilů, spalovacích motorů a kolejových vozidel.
- [7] VALSKÝ J.: *Výzkum proudění směsi ve spalovacím prostoru vozidlového čtyřválcového zážehového motoru*. Praha, 2017. Diplomová práce. FS ČVUT, Ústav automobilů, spalovacích motorů a kolejových vozidel.
- [8] SCHMIDT J.: *Přípravek pro měření tumbly ve válci spalovacího motoru*. Praha, 2018. Diplomová práce. FS ČVUT, Ústav automobilů, spalovacích motorů a kolejových vozidel.
- [9] VLČEK P., JIROUT T.: *Modelování turbulentního proudění*. [Přednáška z konference]. Ústav procesní a zpracovatelské techniky, FS ČVUT, Praha, 2013. Dostupné z <http://chps.fsid.cvut.cz/pt/2013/pdf/3509.pdf>
- [10] HATSCBACH P., VÍTEK O., TICHÁNEK R.: *In-cylinder flow characterization using vorticity based parameters* [online]. Publikace Vol. 18 No. 1 (2021). FS ČVUT, Ústav automobilů, spalovacích motorů a kolejových vozidel, Praha, 2018. [vid. 2.5.2022]. Dostupné z <https://ojs.cvut.cz/ojs/index.php/MECCA/article/view/7698>
- [11] *AVL, Software Documentation 2020*. AVL List, 2020. Dostupné ze spouštěcí obrazovky SW AVL
- [12] *NI DIAdem Documentation, Help, Examples 2021*. National Instruments, 2021. Dostupné v SW NI DIAdem
- [13] AVRAHAM T.: *Pots and PANS*. Článek z blogu, <https://cfdisrael.blog/> [online]. Tom Avraham, 2017. [vid 10.5.2022]. Dostupné z <https://cfdisrael.blog/2017/03/29/pots-and-pans/>



- [14] *Wikipedia* [online]. Otevřená encyklopedie, česká i anglická verze. Wikimedia Foundation, Inc. [vid. 8.5.2022]. Dostupné z <https://www.wikipedia.org/>
- [15] *Youtube* [online], Internetový server pro sdílení videí. Google Inc. [vid 9.5.2022]. Dostupné z <https://www.youtube.com/>
- [16] HATSCHBACH P.: *Měření rychlostních polí pro ověření metod kvantifikace velkých vírových struktur ve válci spalovacího motoru*. Konferenční příspěvek. ISBN 978-80-87117-16-3, p. 14-19. Prague, 31st SYMPOSIUM ON ANEMOMETRY, Institute of Hydrodynamics CAS, 2017.
- [17] *CFD Online* [online]. CFD komunita od roku 1994, články, informace, diskuze, novinky apod. [vid 6.5.2022]. Dostupné z <https://cfd-online.com/>

Seznam použitých obrázků

Obrázek 1: Pohled na sítě z motorů s přímým a nepřímým vstřikem paliva	12
Obrázek 2: Řez měřicím stanovištěm – HWA [1]	13
Obrázek 3: Schéma měřicího stanoviště PIV [16]	14
Obrázek 4: Porovnání metod k- ζ -f a PANS na hlavě s nepřímým vstřikem paliva – rychlostní pole v m/s ..	15
Obrázek 5: Odvozená geometrická plocha průtokového součinitele [5]	18
Obrázek 6: Uvažovaný průřez kanálu bez dířku ventilu [5]	19
Obrázek 7: Tumble a swirl [5].....	20
Obrázek 8: Pohled na prostředí Excel [14].....	23
Obrázek 9: Pohled na prostředí MATLAB [14]	23
Obrázek 10: Pohled na prostředí ParaView [14].....	24
Obrázek 11: Pohled na prostředí Tecplot [15].....	25
Obrázek 12: Pohled na prostředí VisIt [15].....	25
Obrázek 13: Pohled na prostředí Gnuplot [15].....	26
Obrázek 14: Pohled na prostředí DIAdem [15]	27
Obrázek 15: Surová data z HWA měření	29
Obrázek 16: Excel sloužící k úpravě dat	30
Obrázek 17: Surová data z PIV měření	30
Obrázek 18: Příklad extrémního posunu od základních souřadnic	31
Obrázek 19: Požadovaný formát dat z FIRE	32
Obrázek 20: Pohled na data hustoty z FIRE	32
Obrázek 21: Výpis důležitých hodnot, zde společně s celkovou rychlostí	32
Obrázek 22: Data s a bez plug-inu	33
Obrázek 23: Posun hodnot ze dvou zmiňovaných datových skupin	33
Obrázek 24: Záložky DIAdem, otevřen SCRIPT.....	34
Obrázek 25: Pohled na řez válcem a zároveň i nádobou – axiální složka rychlosti v m/s.....	35
Obrázek 26: Rozdíl v hodnotách před smazáním (nahore) a po smazání (dole) – filtrace.....	36
Obrázek 27: Přejít mezi nulovými daty a daty relevantními	36
Obrázek 28: Hledání a změna textu	42
Obrázek 29: Graf otáček a rychlosti ze vzorového příkladu – sekce VIEW.....	47
Obrázek 30: Náhled na výstup ze vzorového příkladu – sekce REPORT	47



Obrázek 31: Úprava velikosti nebo umístění grafu a příslušných os	48
Obrázek 32: Úprava polárního souřadnicového systému	49
Obrázek 33: První zobrazení grafu axiální rychlosti	49
Obrázek 34: Příklad jednoduchého pohledu v grafu na axiální rychlost ve válci	50
Obrázek 35: Zleva zhora: příklad zobrazení izočár, vypočítaných míst a min/max hodnot	51
Obrázek 36: Příklad zobrazení měřených bodů v experimentu HWA	52
Obrázek 37: Vektory v třírozměrném prostoru	53
Obrázek 38: Zobrazení vektorů v jednotlivých plochách	53
Obrázek 39: Vlevo vektory U a V složek, vpravo vektory U, V a W složek	54
Obrázek 40: Nastavení vektorového pole	54
Obrázek 41: Srovnání vektorových polí vlevo ze 2 tisíc a vpravo ze 7 tisíc hodnot	55
Obrázek 42: Myšlený vzorový vektor znázorňující poměrovou velikost rychlosti	55
Obrázek 43: Nastavení diferenční mapy	56
Obrázek 44: Příklad zobrazení rychlostní složky W metodami PANS a k- ζ -f a diferenční mapy	57
Obrázek 45: Příklad průběhu tumble a swirl čísel na hlavě s nepřímým vstřikováním paliva	58
Obrázek 46: Nastavení tabulky	59
Obrázek 47: Příklad tabulky s tumble a swirl čísly včetně redukováných čísel	59
Obrázek 48: Příklad srovnání axiálních rychlostí dvou výpočetních metod	60
Obrázek 49: Příklad pohledu na jednotlivou simulaci a vlastnosti	61
Obrázek 50: Příklad srovnání simulace a PIV experimentu u hlavy s nepřímým vstřikem	62
Obrázek 51: Příklad srovnání simulace a HWA experimentu u hlavy s přímým vstřikem	63
Obrázek 52: Příklad přímého srovnání 2 hlav a 2 výpočetních metod	63
Obrázek 53: Příklad výstupu z proměnného otevření ventilů – zdvih 2 mm	64
Obrázek 54: Příklad grafu vírových a průtokových charakteristik měnicích se v závislosti na zdvihu ventilů	64
Obrázek 55: Snímky příkladu vizualizace proměnného zdvihu jednoho a obou ventilů, zde v rozmezí 1-6 mm.	65
Obrázek 56: Krátký přehled exemplů v NI DIAdem	66

Seznam příloh

Skripty

Reporty (vizualizační prostředí) + příklady v PDF

Převodní excel k HWA metodě

Soubor kružnice

Datový plug-in pro AVL FIRE