



TECHNICKÁ UNIVERZITA V LIBERCI
Fakulta mechatroniky, informatiky
a mezioborových studií ■

Numerické modelování proudění tekutiny metodou spektrálních elementů

Bakalářská práce

Studijní program: B3901 – Aplikované vědy v inženýrství
Studijní obor: 3901R055 – Aplikované vědy v inženýrství
Autor práce: **Ota Procházka**
Vedoucí práce: doc. Ing. Petr Šidlof, Ph.D.





TECHNICAL UNIVERSITY OF LIBEREC
Faculty of Mechatronics, Informatics
and Interdisciplinary Studies ■

Numerical Simulation of Fluid Flow Using Spectral Element Method

Bachelor thesis

Study programme: B3901 – Applied Sciences in Engineering
Study branch: 3901R055 – Applied Sciences in Engineering
Author: **Ota Procházka**
Supervisor: doc. Ing. Petr Šidlof, Ph.D.





Zadání bakalářské práce

Numerické modelování proudění tekutiny metodou spektrálních elementů

Jméno a příjmení: **Ota Procházka**
Osobní číslo: M16000102
Studijní program: B3901 Aplikované vědy v inženýrství
Studijní obor: Aplikované vědy v inženýrství
Zadávající katedra: Ústav nových technologií a aplikované informatiky
Akademický rok: **2019/2020**

Zásady pro vypracování:

1. Nastudujte základy dynamiky tekutin konkrétně obtékání těles při různých Reynoldsových číslech. Seznamte se se standardními numerickými metodami používanými pro simulace proudění tekutin metodou konečných diferencí, metodou konečných prvků a metodou konečných objemů. Nastudujte princip Metody spektrálních elementů.
2. Nainstalujte open-source knihovnu NEK5000 a vyzkoušejte si práci v této knihovně spuštěním tutoriálů.
3. Realizujte v knihovně NEK5000 simulaci obtékání válce při nízkých a středních Reynoldsových číslech. Porovnejte výsledky s dostupnými benchmarkovými daty. Vyzkoušejte paralelizaci výpočtu pomocí knihovny MPI a vyhodnoťte škálovatelnost (zrychlení) při využití různého počtu výpočetních jader.
4. Shrňte získané zkušenosti a zhodnoťte práci v knihovně NEK5000.

Rozsah grafických prací:
Rozsah pracovní zprávy:
Forma zpracování práce:
Jazyk práce:

dle potřeby
30-40 stran
tištěná/elektronická
Čeština



Seznam odborné literatury:

- [1] White F. M. (2006), Fluid Mechanics, McGraw-Hill.
- [2] Brennen C. E. (2006), Internet Book on Fluid Dynamics, Caltech, <http://brennen.caltech.edu/fluidbook/content.htm> (Online)
- [3] Versteeg H., Malalasekera W. (2007), An Introduction to Computational Fluid Dynamics. The Finite Volume Method, Pearson Education Limited.
- [4] Patera A. T. (1984), A spectral element method for fluid dynamics: Laminar flow in a channel expansion, Journal of Computational Physics 54 (3), pp. 468-488
- [5] NEK5000 a fast and scalable high-order solver for computational fluid dynamics, <https://nek5000.mcs.anl.gov/> (Online)

Vedoucí práce:

doc. Ing. Petr Šidlof, Ph.D.
Ústav nových technologií a aplikované informatiky

Datum zadání práce:

9. října 2019

Předpokládaný termín odevzdání:

18. května 2020

prof. Ing. Zdeněk Plíva, Ph.D.
děkan

L.S.

Ing. Josef Novák, Ph.D.
vedoucí ústavu

Prohlášení

Byl jsem seznámen s tím, že na mou bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb., o právu autorském, zejména § 60 – školní dílo.

Beru na vědomí, že Technická univerzita v Liberci (TUL) nezasahuje do mých autorských práv užitím mé bakalářské práce pro vnitřní potřebu TUL.

Užiji-li bakalářskou práci nebo poskytnu-li licenci k jejímu využití, jsem si vědom povinnosti informovat o této skutečnosti TUL; v tomto případě má TUL právo ode mne požadovat úhradu nákladů, které vynaložila na vytvoření díla, až do jejich skutečné výše.

Bakalářskou práci jsem vypracoval samostatně s použitím uvedené literatury a na základě konzultací s vedoucím mé bakalářské práce a konzultantem.

Současně čestně prohlašuji, že tištěná verze práce se shoduje s elektronickou verzí, vloženou do IS STAG.

Datum:

Podpis:

Abstrakt

Tato práce se zaměřuje na metody numerických simulací proudění tekutin. Cílem této práce je otestovat open-source solver NEK5000 využívající metodu spektrálních elementů a realizovat v něm simulaci obtékání válce, které je častým vzorovým případem a lze ji využít k verifikaci a validaci metody ve srovnání s jinými metodami.

Klíčová slova: nestlačitelné proudění, obtékání, simulace, dynamika tekutin, Reynoldsovo číslo, metoda spektrálních elementů

Abstract

This work is focused on numerical simulation methods of fluid flow. The target of this work is to try using an open-source solver called NEK5000 which employs the spectral elements method and to implement flow over a cylinder, which is a common example suitable for verification and validation of methods.

Keywords: incompressible flow, simulation, fluid dynamics, Reynolds number, spectral element method

Poděkování

Poděkování patří především vedoucímu mé bakalářské práce, panu doc. ing. Petru Šidlovi, Ph.D., za vedení, návrh tématu, konzultace a čas, který se mnou strávil. Dále přátelům, kteří neváhali se mnou sdílet své znalosti. Adamu Peklákovi za tipy k duálním operačním systémům a používání linuxu, Karolíně Sedláčkové za informace k Pythonu a Overleaf, Josefu Musilovi a Sabině Bednářové za inspirující diskuze o numerických metodách, rodičům za korekturu textu, Martině Hanzlíkové a Glebovi Pokatilovovi za vzájemnou podporu při psaní práce.

Obsah

Seznam zkratek	8
1 Úvod	9
2 Dynamika tekutin	10
2.1 Tekutiny	10
2.1.1 Definice	10
2.1.2 Vlastnosti	10
2.2 Popis proudění	12
2.2.1 Lagrangeova metoda	12
2.2.2 Eulerova metoda	13
2.2.3 Vektorové pole, trajektorie, proudnice	13
2.3 Obtékání těles	13
2.3.1 Reynoldsovo číslo	13
2.3.2 Odporové síly	15
2.3.3 Strouhalovo číslo	17
2.4 Navier-Stokesovy rovnice	17
2.5 Numerické metody výpočtu	18
2.5.1 Výpočetní síť	18
2.5.2 Metoda konečných diferencí	19
2.5.3 Metoda konečných prvků	20
2.5.4 Metoda konečných objemů	20
2.5.5 Spektrální metoda	20
2.5.6 Metoda spektrálních elementů	20
2.6 Galerkinova metoda	22
3 Testování solveru využívajícího metodu spektrálních elementů	24
3.1 Volba solveru	24
3.2 Nektar++	24
3.3 Použitý hardware a software	25
3.4 Kompilace, tutoriály	25
3.5 Benchmarkový problém	26
3.6 Realizace simulace a práce s knihovnou	28
3.7 Výsledky simulace	34
3.7.1 Stacionární případ $Re=20$	35
3.7.2 Nestacionární případ $Re=100$	36

3.8 Paralelní škálování	37
4 Závěr	41

Seznam zkratek

EU	Evropská unie
MIT	Massachusetts Institute of Technology
GPG	GNU Privacy Guard
PDR	Parciální diferenciální rovnice
CFD	Computational Fluid Dynamics
MIRA	Multi-Cluster/Sector Initial Rapid Assessment
MKD	Metoda konečných diferencí
MKP	Metoda konečných prvků
MKO	Metoda konečných objemů
SM	Spektrální metoda
MSE	Metoda spektrálních elementů
GPU	Graphics Processing Unit
CPU	Central Processing Unit
DoF	Degrees of Freedom

1 Úvod

Tato práce vznikla v návaznosti na práci profesora Anthonyho T. Patera z MIT, který ve svém článku [18] představil metodu spektrálních elementů aplikovanou na Navier-Stokesovy rovnice popisující proudění kontinua. Jde o další metodu numerické simulace proudění vedle široce využívaných metod konečných objemů a konečných prvků. Spočívá v propojení obecnosti metody konečných prvků a přesnosti spektrální metody.

V ČR se metodou spektrálních elementů zabývá Jan Pech, Ph.D., z Ústavu termomechaniky Akademie věd České republiky, který se v současnosti specializuje na modelování nestability při obtékání zahřívaných těles. Ve své dřívější práci realizoval simulaci obtékací válce ve 2D [13]. Jeho práce tématem úzce souvisí s touto prací, provádí v ní ale simulace na extrémně hrubé síti o 9 elementech při vysokých řádech polynomů (až 49). Zde je přinesena další sada výsledků pro automaticky generovatelné, nestrukturované, trojúhelníkové sítě různých jemností. Zároveň obsahuje zhodnocení knihovny implementující spektrální/hp metody NEK5000, vysvětlení, proč byla místo ní použita jiná knihovna NEKTAR++ a popis práce s touto knihovnou.

Tato práce je strukturována do čtyř kapitol. První kapitolou je samotný úvod. Druhá kapitola pokračuje teorií dynamiky tekutin zahrnující stručný popis numerických metod. Ve třetí kapitole následuje vysvětlení práce s knihovnou a tabulky s výsledky. Čtvrtou kapitolou je závěr obsahující shrnutí a zhodnocení práce.

2 Dynamika tekutin

V této části se zaměříme na vybranou teorii k mechanice tekutin potřebnou k pochopení podstaty proudění a jeho numerických simulací. Dále pak na samotné metody numerických simulací.

2.1 Tekutiny

2.1.1 Definice

Tekutina je látka složená z molekul, mezi kterými je oproti pevným látkám větší vzdálenost a působí mezi nimi slabší vazebné síly. Netvoří krystalickou strukturu, ale volně se pohybují. V dynamice tekutin ale přistupujeme k tekutinám makroskopicky. Hned od začátku jde tedy o abstrakci. Ta ovšem v makroměřítku velmi dobře funguje. Podle [5] je tekutina látka, která se po nespécifikovanou dobu deformuje při působení neizotropních sil. Na druhou stranu pevné látky jsou takové, které se při působení stálých sil deformují a po čase se ustálí a deformace ustanou. Je zřejmé, že tato definice není zcela jednoznačná právě kvůli nespécifikovanému času. Na době pozorování může záviset rozhodnutí o zařazení látky při daných podmínkách mezi pevné látky, nebo tekutiny. To je ale v pořádku, protože to skutečně odráží realitu. Pohled na věc, že látka je jedno, nebo druhé nemusí být vždy zcela přesný. Dobrým příkladem je zemský plášť, který přenáší seismické vlny a přestává se pod působením stálých sil deformovat z krátkodobého hlediska jako pevná látka, ale zároveň umožňuje pohyb a deformaci litosferických (tektonických) desek jako tekutina. Proto popis látky jako pevné nebo tekuté závisí na době pozorování deformace.

2.1.2 Vlastnosti

V návaznosti na zavedení tekutiny jako kontinua pracujeme i s veličinami jako s kontinuálními. Určujeme je v bodech prostoru, přestože jsou to makroskopické veličiny a jejich spojitý charakter neodpovídá částicové povaze hmoty.

Nyní si zavedeme některé veličiny používané k popisu tekutin v klidu. Nejprve **tlak**. Uvažujeme plochu o obsahu S umístěnou do statické tekutiny. Pak tlak působící na tuto plochu p je daný podílem síly F působící kolmo na plochu a velikostí plochy S , tedy $p = \frac{F}{S}$. Tlak je makroskopická veličina. Podle Newtonova zákona je způsoben normálovými impulzy nárazů molekul na jednotkovou plochu tělesa za jednotku času, což je z delšího časového hlediska konstantní. Také může být tlak

daný tokem hybnosti skrz plochu. Pomocí toho můžeme definovat tlak v kterékoli oblasti v tekutině i bez přítomnosti tělesa. Záleží to ovšem na orientaci plochy. Ve statických případech to nevádí, protože náhodný pohyb molekul je ve všech směrech stejný, tudíž tlak na směr nezávisí. V proudící tekutině je pohyb neisotropní.

Dále používáme veličinu zvanou **tenzor napětí**. Ta je stejně jako tlak v jednotkách Nm^{-2} . Jde o tenzor druhého řádu 3×3 . Působí-li na tekutinu v různých bodech a směrech síly $\vec{F}_1, \dots, \vec{F}_N$, potom na infinitezimální plochu procházející jakýmkoli bodem tekutiny působí vektor napětí \vec{T} , který svírá obecný úhel ϕ s normálou plochy \vec{n} . Protože ale záleží na natočení plochy, popíšeme napětí v bodě pro tři plochy kolmé na osy kartézských souřadnic pomocí trojsložkových vektorů

$$\vec{T}_1 = (\sigma_{1,1}, \sigma_{1,2}, \sigma_{1,3}),$$

$$\vec{T}_2 = (\sigma_{2,1}, \sigma_{2,2}, \sigma_{2,3}),$$

$$\vec{T}_3 = (\sigma_{3,1}, \sigma_{3,2}, \sigma_{3,3}).$$

Pomocí těchto hodnot lze určit i napětí libovolně orientované plochy procházející tímtož bodem s normálou \vec{n} jako

$$T_i = \sigma_{ji}n_j, \quad (2.1)$$

Pokud je tekutina v rovnovážném stavu, plyne ze zachování momentu, že tenzor je symetrický a stačí tedy k popisu napětí v bodě šest hodnot místo původních devíti.

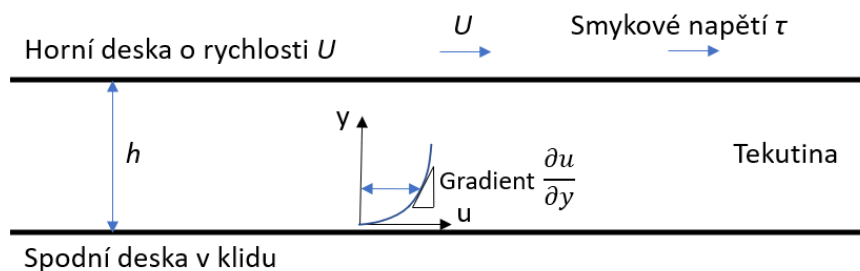
Teplota se udává v kelvinech K a vychází z Boltzmanovy konstanty. Souvisí s pohybem mikroskopických částic (hlavně molekul) a jejich kinetickou energií. Často proudění aproximujeme jako nestlačitelné, izotermické. Tehdy je teplota konstantní.

Hustota ρ je definována jako $\rho = \frac{m}{V}$. Je proto v jednotkách kg/m^3 . V reálných tekutinách je hustota funkcí teploty a tlaku $\rho(T, p)$. Často je teplota v celém objemu tekutiny stejná. Pak můžeme závislost hustoty na teplotě zanedbat. Při řešení proudění kapalin nebo pomalého proudění plynů (přibližně do hranice Machova čísla $M = 0,3$) můžeme zanedbat i závislost na tlaku, pak $\rho = \text{konst}$ a jde o tzv. nestlačitelné proudění. Tím se zabýváme v této práci.

Viskozita (jinak také vazkost) je další vlastnost tekutin. Projevuje se pouze při proudění, nikoli pokud jsou tekutiny v klidu. Charakterizuje vnitřní tření kapaliny způsobené přitažlivými silami působícími mezi částicemi. V ideálních tekutinách je viskozita nulová, v reálných vyšší. Projevuje se zpomalováním proudění a mezi laiky je často zaměňována za hustotu. V [5] viskozitu vysvětlují na myšleném experimentu - proudění mezi dvěma rovnoběžnými nekonečnými deskami vzdálenými od sebe h , přičemž spodní je nehybná a horní se pohybuje rychlostí U (viz Obr. 2.1).

Tenká vrstva tekutiny při povrchu stěny má rychlost té stěny. Jde o tzv. no-slip podmínku. Na horní desku musí působit síla ve směru jejího pohybu, aby udržovala svou rychlost U . Sílu vztaženou k jednotce plochy nazýváme „smykové napětí“ τ . To je přenášeno mezi vrstvami tekutiny napříč celou oblastí a je všude stejné. Když si označíme vzdálenost od spodní desky souřadnicí y , viz obrázek 2.1, a rychlost tekutiny jako $u(y)$, tak smykové napětí vyvolává rychlostní gradient $\frac{\partial u}{\partial y}$ a viskozita μ je vlastnost tekutiny určující poměr mezi smykovým napětím a gradientem rychlosti

$$\tau = \mu \frac{\partial u}{\partial y}. \quad (2.2)$$



Obrázek 2.1: Laminární proudění mezi dvěma nekonečnými rovnoběžnými deskami, rychlostní profil

Při analýze dalších jevů se často používá **kinematická viskozita**, která je podílem viskozity a hustoty

$$\nu = \mu / \rho. \quad (2.3)$$

Udáváme ji v jednotkách m^2/s . To jsou obecně jednotky pro jakoukoli difuzivitu - míru schopnosti částic tekutiny se pohybovat.

2.2 Popis proudění

Pro matematický popis proudění se používají dvě metody: Lagrangeova a Eulerova.

2.2.1 Lagrangeova metoda

Jednou z možností popisu proudění je Lagrangeova metoda. Spočívá v tom, že pozorujeme konkrétní malý element tekutiny a popíšeme, jak se jeho poloha x_i , tlak p , teplota T atd. mění v čase t .

$$x_i(t), \quad p(t), \quad T(t)$$

Nyní můžeme snadno určit rychlost jako $u_i(t) = \frac{dx_i}{dt}$. Pochopitelně takových elementů je v tekutině mnoho a jiný element bude mít v čase t obecně jiné vlastnosti. Proto při snaze popsat celé proudění musíme vybrat všechny elementy a rozlišit je třeba podle polohy X_i v nějakém počátečním čase

$$x_i(X_i, t), \quad p(X_i, t), \quad T(X_i, t).$$

Nevýhodou této metody je, že není praktická k popisu proudění v konkrétním bodě prostoru, např. při povrchu obtékaného tělesa. Výhodou je, že Lagrangeův popis je vhodný pro aplikaci zákonů Newtonovské fyziky (zákon zachování hmotnosti, pohybových zákonů) a zákonů termodynamiky. Aplikace těchto zákonů je klíčová při sestavování rovnic pro popis pohybu tekutin.

2.2.2 Eulerova metoda

Druhým způsobem, jak popsat proudění, je pozorovat vybraný bod v prostoru a sledovat časový vývoj stavu tekutiny v tomto bodě. Bod určíme pomocí kartézského souřadného systému jako x_1, x_2, x_3 (nebo x_i) a dále určujeme rychlost u_i , tlak p , teplotu T a další veličiny, jak se vyvíjí v čase ve všech bodech.

$$u_i(x_i, t), \quad p(x_i, t), \quad T(x_i, t) \quad (2.4)$$

Protože oba přístupy popisují proudění kompletně, je zřejmé, že jeden popis lze převést na druhý. Dalším krokem je výše zmíněné zákony potřebné pro výpočty proudění převést do tvaru využitelného pro Eulerův popis. K tomu je potřeba vztah mezi časovými derivacemi v každém z popisů. Časovou derivaci v Lagrangeově popisu (tzv. materiálovou derivaci) označíme $\frac{D}{Dt}$, časovou derivaci v Eulerově popisu $\frac{\partial}{\partial t}$. Potom platí

$$\frac{D}{Dt} = \frac{\partial}{\partial t} + (u \cdot \nabla). \quad (2.5)$$

2.2.3 Vektorové pole, trajektorie, proudnice

Jedním způsobem, jak lze graficky znázornit proudící tekutinu, je vykreslit některé z vektorů rychlostního **vektorového pole**. Toto pole má rychlostní vektor v každém bodě, ale to samozřejmě není možné zakreslit. Vyobrazíme tedy jen vybrané vektory pomocí šipek. Orientace šipky určuje směr, délka velikost rychlosti. S vhodně zvolenou hustotou rozmístění a poměrem délky šipky a velikosti rychlosti dostáváme přehledný obraz, který nám umožňuje udělat si představu o tom, jak vypadá proudění v nějakém čase.

Dalším způsobem jsou **trajektorie** - křivky tvořené množinou bodů, kterými prošel Lagrangeův element tekutiny (viz odstavec 2.2.1).

Proudnice zase odpovídají Eulerovu popisu (viz odstavec 2.2.2). Jsou to takové křivky, ke kterým je v každém bodě tečný vektor rychlosti. Pro ustálené proudění jsou trajektorie a proudnice totéž.

2.3 Obtékání těles

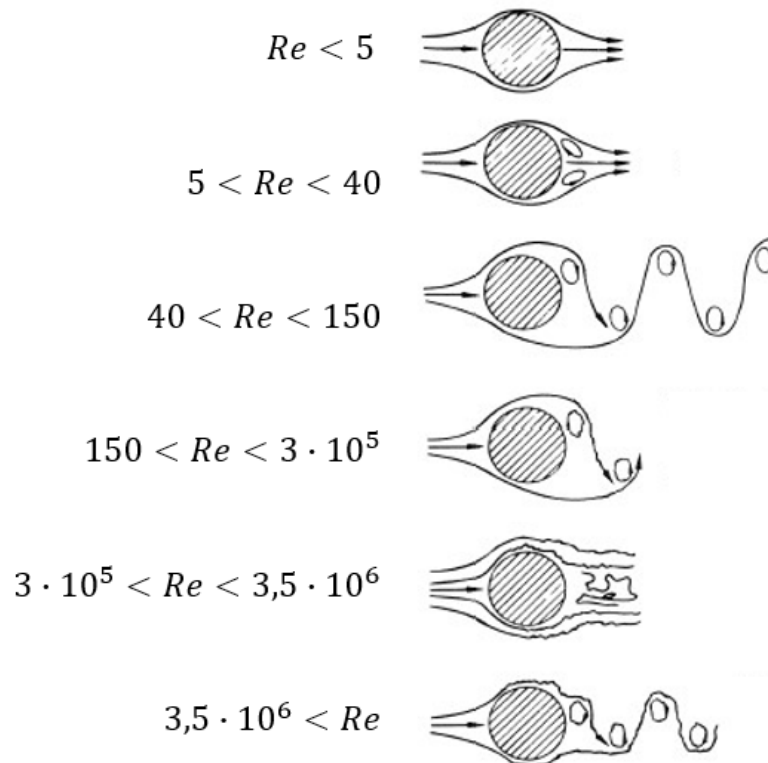
Když umístíme objekt do toku tekutiny (nebo ekvivalentně objekt necháme pohybovat tekutinou), začne být tekutinou obtékán.

2.3.1 Reynoldsovo číslo

Jak je vysvětleno v [22], Reynoldsovo číslo Re je bezrozměrné podobnostní číslo učené následovně

$$Re = \frac{VL}{\nu}, \quad (2.6)$$

kde V a L jsou charakteristická rychlost a charakteristický rozměr a ν kinematická viskozita podle vztahu 2.3. Podobnostní čísla se v mechanice tekutin používají ke srovnání dvou systémů.



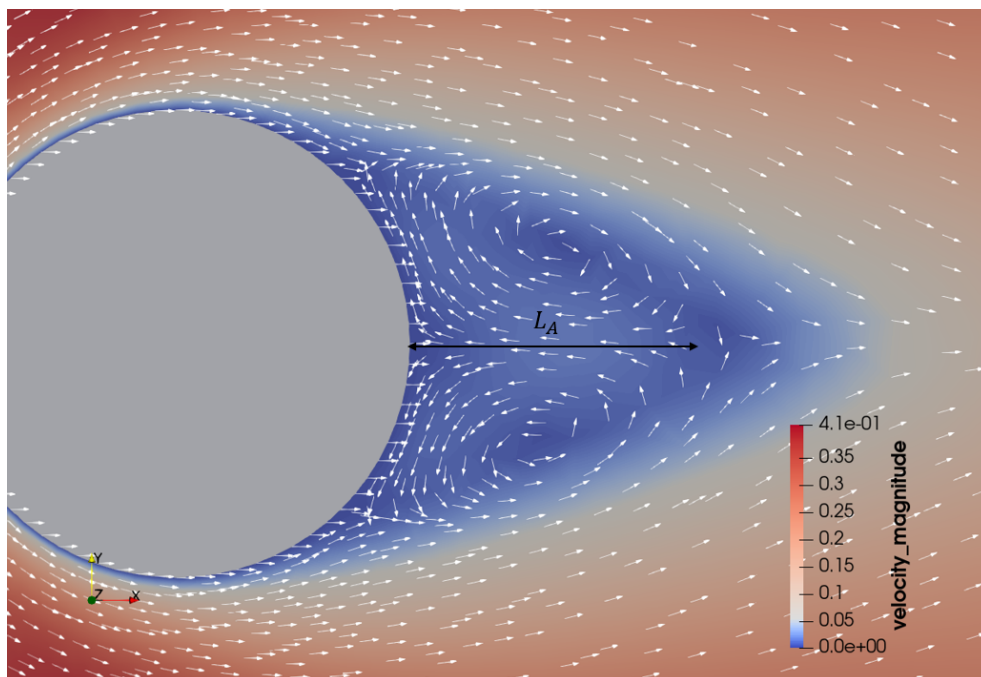
Obrázek 2.2: Charakter proudění při různých Re [4]

Charakteristická rychlost může být např. průměrná rychlost na přítoku, charakteristický rozměr může být průměr trubice nebo obtékaného válce, charakteristický čas může být převrácená hodnota frekvence odtrhávání vírů. Je to konvenčně dáno pro konkrétní případy. Dosazením charakteristických veličin do Navier-Stokesových rovnic získáme vztah, kde vystupují výrazy, které použijeme jako podobnostní čísla. V této práci využíváme Reynoldsovo a Strouhalovo, ale existují třeba ještě Machovo, Prandtlovo a další.

Typy proudění pro různá Reynoldsova čísla jsou naznačeny na obrázku 2.2. Podle charakteru proudění v oblasti za tělesem (tzv. úplav) říkáme, že jsou tělesa obtékána v různých režimech. Nízká Re indikují creep proudění s minimálním vlivem setrvačných účinků. Také se mu říká Stokesovo proudění. Při $Re > 5$ vzniká za tělesem dvojice vírů, kde dochází ke zpětnému proudění. Pro hodnoty Reynoldsova čísla $40 < Re < 150$ je proudění laminární a za tělesem vzniká vzorová řada. S $Re > 150$ roste šance vzniku turbulentního (vířivého) proudění charakteristického silnými, náhodnými fluktuacemi. Postupně je nejprve přechodné a od $Re = 300$ pak v úplavu plně turbulentní. Po překročení Re okolo $3 \cdot 10^5$ se naruší laminární okrajová vstava při povrchu tělesa. Proudění za tělesem je chaotické. Při Re odpovídajícímu $3,5 \cdot 10^6$

se opět objevuje turbulentní vzorová řada.

V režimu s dvěma víry ($5 < Re < 40$) za obtékaným tělesem vzniká tzv. **recirkulační zóna**. V této oblasti je nižší tlak a dochází tam ke zpětnému proudění. Za neměnných podmínek se velikost oblasti ustálí. Zkoumání délky recirkulační zóny L_A , viz obrázek 2.3, a rozdílu tlaků před a za tělesem využíváme v této práci k porovnání výsledků simulace s výsledky simulací využívajících jiné metody.



Obrázek 2.3: Rychlostní vektorové pole s $Re=20$

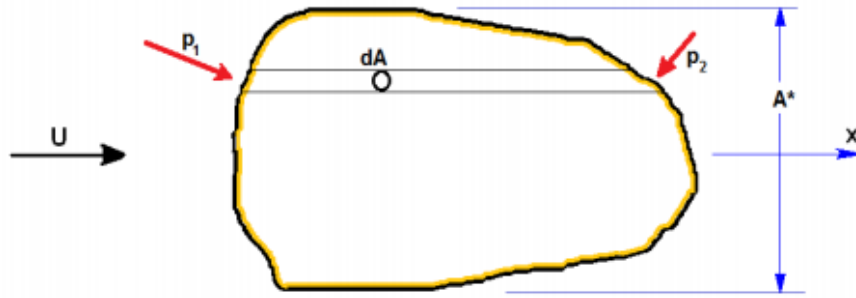
2.3.2 Odporové síly

Ze zkušenosti víme, že na obtékaná tělesa působí síly. Rozlišujeme dvě a říkáme jim aerodynamická odporová (drag force) F_D a aerodynamická vztlaková (lift force) F_L . K nim si zavedeme bezrozměrné koeficienty

$$c_D = \frac{2F_D}{\rho U^2 A^*}, \quad c_L = \frac{2F_L}{\rho U^2 A^*}, \quad (2.7)$$

kde ρ je hustota tekutiny v referenčním bodě a A^* je plocha průřezu. Obvykle jde o plochu čelního průmětu ze směru přitékajícího proudu.

Podle příčiny vzniku rozlišujeme síly na další dvě skupiny - síly způsobené rozdílem tlaků a síly způsobené smykovým napětím působícím na povrch. Uvažujme stacionární proudění s rychlostí U konstantní před tělesem, které je obtékáno. Objem tělesa rozdělíme na elementární válce o průřezu dA s osami rovnoběžnými se směrem proudění (viz Obr. 2.4) a označíme tlak působící na čelní podstavu p_1 a tlak působící na podstavu na straně úplavu p_2 . Pak síla ve směru proudění způsobená



Obrázek 2.4: Tlaky přispívající k aerodynamické odporové síle [5]

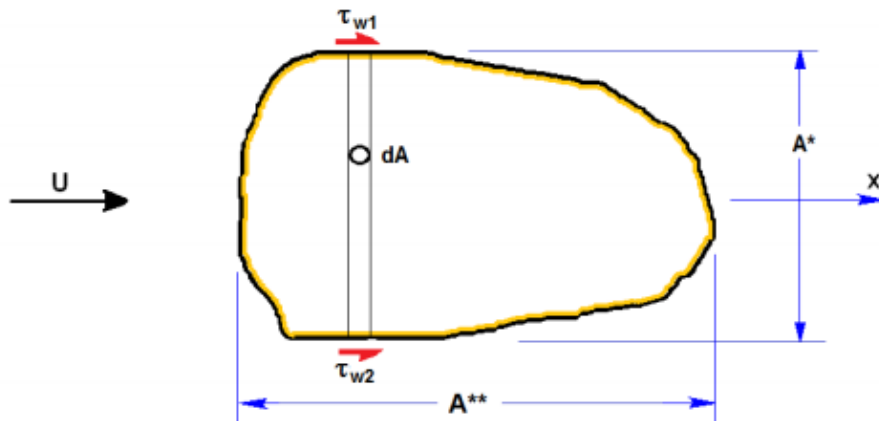
tlakem $F_{D,p}$ bude

$$F_{D,p} = \int_{A^*} (p_1 - p_2) dA. \quad (2.8)$$

dosazením do vztahu 2.7 dostaneme

$$c_{D,p} = \frac{1}{A^*} \int_{A^*} \frac{(p_1 - p_2)}{\frac{1}{2}\rho U^2} dA = \int_{A^*} (c_{p1} - c_{p2}) \frac{dA}{A^*}. \quad (2.9)$$

Podobně určíme sílu působící ve směru proudění způsobenou smykovým napětím. Elementární válce jsou tentokrát orientovány vertikálně kolmo na směr proudění (viz Obr. 2.5).



Obrázek 2.5: Smykové napětí přispívající k aerodynamické odporové síle [5]

Smykové tření působící na podstavy pojmenujeme τ_{w1} a τ_{w2} . Sílu pak vyjádříme jako

$$F_{D,\tau} = \int_{A^{**}} (\tau_1 + \tau_2) dA, \quad (2.10)$$

kde dA je průřez elementárních válců a A^{**} je obsah plochy průmětu tělesa ve směru

os el. válců. Dále určíme koeficient pro tuto sílu:

$$c_{D,\tau} = \frac{2}{A^*} \int_{A^{**}} \frac{(\tau_1 + \tau_2)}{\rho U^2} dA = \frac{2A^{**}}{A^*} \int_{A^{**}} \frac{(\tau_1 + \tau_2)}{\rho U^2} \frac{dA}{A^*}. \quad (2.11)$$

Za povšimnutí stojí, že narozdíl od 2.9 zde vystupuje faktor $\frac{2A^{**}}{A^*}$. Je z toho tedy vidět, že i když smykové napětí je mnohem menší než rozdíl tlaků, může být $F_{D,\tau}$ dominantní, platí-li $A^{**} \gg A^*$. To např. platí pro aero/hydrodynamická tělesa.

Dynamické vztlakové síly vyjádříme obdobně jako

$$F_{L,p} = \int_{A^{**}} (p_1 - p_2) dA \quad (2.12)$$

pro vztlakovou sílu způsobenou rozdílem tlaků a následně její koeficient jako

$$c_{L,p} = \frac{1}{A^*} \int_{A^{**}} \frac{(p_2 - p_1)}{\frac{1}{2}\rho U^2} dA = \frac{A^{**}}{A^*} \int_{A^{**}} (c_{p2} - c_{p1}) \frac{dA}{A^{**}}. \quad (2.13)$$

Zde nám poměr $\frac{A^{**}}{A^*}$ napovídá, že pro aero/hydrodynamická tělesa může být z dvojice sil vyvolaných rozdílem tlaků větší dynamická vztlaková než tažná. Zbývá poslední - dynamická vztl. síla způsobená smykovým napětím. Ta se opět vyjádří analogicky, tak rovnou shrneme sekci o odporových silách tím, že výsledná F_D a F_L se spočítají jako prostý součet jejich složek

$$F_D = F_{D,p} + F_{D,\tau}, \quad F_L = F_{L,p} + F_{L,\tau}. \quad (2.14)$$

2.3.3 Strouhalovo číslo

Další číslo používané k popisu proudění je Strouhalovo číslo

$$St = \frac{Df}{U} = \frac{D}{T U} = \frac{u_{osc}}{U}, \quad (2.15)$$

kde D je charakteristiký rozměr, U rychlost proudění a f je frekvence separace vírů. V alternativních vyjádřeních je T doba periody a $u_{osc} = D/T$ oscilační rychlost. Opět se jedná o podobnostní bezrozměrné číslo a souvisí s oscilací proudícího systému. Zápis, kde je Strouhalovo číslo vyjádřeno jako podíl oscilační rychlosti a rychlosti proudění, se často používá v simulacích s periodicky pohybující se geometrií, jako je například při tvorbě zvuku, vlnění křídel letadla a plavání zvířat.

2.4 Navier-Stokesovy rovnice

Navier-Stokesovy rovnice se v literatuře objevují v mnoha formách. Záleží na tom, jaká přijmeme zjednodušení a jaké předpoklady. Pro účely této práce použijeme rovnice pro nestlačitelné proudění tak, jak jsou uvedeny v [1]

$$\frac{\partial u}{\partial t} + u \cdot \nabla u = -\nabla p + \nu \nabla^2 u + f \quad (2.16)$$

$$\nabla \cdot u = 0,$$

kde u je rychlost, p je tlak a ν kinematická viskozita. První vztah vychází z Newtonova pohybového zákona (zachování hybnosti), je to nelineární parciální diferenciální rovnice druhého řádu. Taková rovnice obecně nemá analytické řešení. Pro některé případy proudění bylo analytické řešení nalezeno. Nicméně i pro ostatní případy lze nalézt řešení numericky s využitím počítačové techniky. Druhý vztah je zákon zachování hmoty (divergence rychlosti je nula, tedy tekutina nevzniká ani nezániká).

2.5 Numerické metody výpočtu

Numerické metody nabyly na obrovském významu s rozvojem výpočetní techniky. Samotný název už napovídá, že jde o výpočty používající přibližné číselné hodnoty. Důležité je si uvědomit, že použití počítačů obnáší aritmetiku s konečnou přesností. Podle způsobu uložení má každá hodnota omezený počet platných číslic, dochází tedy vždy k nějaké chybě, jejíž velikost je ovšem možné určit. Typické jsou pro numerické simulace algoritmy, které zahrnují mnoho opakování, a operace s velkým množstvím dat. Proto jsou k nim počítače prakticky nutností. Přestože to není jediná aplikace, budeme se držet tématu výpočtů dynamiky tekutin - Computational Fluid Dynamics (CFD).

Numerická analýza tekutin nachází uplatnění např. v letectví, automobilovém průmyslu, energetice, chemickém průmyslu, jaderném průmyslu, podmořském průmyslu. Uplatní se také v biomedicině, chladících systémech, vytápění, ventilaci, klimatizaci...

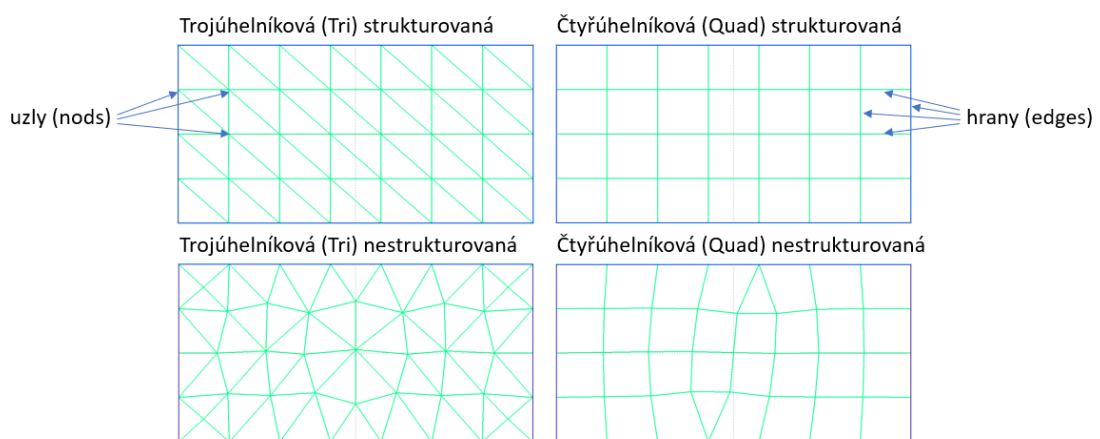
CFD hledá řešení Navier-Stokesových rovnic 2.16. Jsou tři základní metody řešení: metoda konečných diferencí (MKD), metoda konečných prvků (MKP) a metoda konečných objemů (MKO). Podrobněji jsou okomenovány dále. Kromě nich existuje spousta dalších metod jako třeba metoda spektrálních elementů (MSE). Všechny metody ale mají společné, že bývají doprovázeny experimenty. Ty jsou důležité pro ověření správnosti simulace. Často se také postupuje tak, že simulaci „ladíme“ podle experimentálních dat. Můžeme tak třeba docílit správnosti simulace a zároveň dopočítat další informace, které měřit neumíme. Pro představu řekněme, že měříme průběh rychlosti na několika místech proudu. Když sestavíme simulaci tak, aby dávala stejný průběh, máme zároveň informace pro rychlost a tlak v celé oblasti a ještě např. dopočítáme síly působící na obtékané těleso.

2.5.1 Výpočetní síť

Při řešení postupujeme tak, že výpočetní oblast (doménu) tzv. zasiťujeme [6] - rozdělíme na malé podoblasti (elementy). Elementy mohou mít různý tvar (viz Obr. 2.6). Mohou mít dokonce zakřivené hrany a v jedné síti je možné kombinovat více typů

elementů. Síť může být strukturovaná, nestrukturovaná, nebo hybridní (kombinace předchozích dvou). Strukturovaná síť je díky pravidelnosti méně náročná na paměť. Také je lepší z hlediska konvergence a výpočetní časové náročnosti. I tak se častěji používají nestrukturované sítě, protože je mnohem snazší a rychlejší je vygenerovat.

Vytvoření vhodné sítě je klíčové pro správnost a efektivitu simulace. Jemnost sítě přímo ovlivňuje výpočetní náročnost. Nedostatečně jemná síť může způsobit numerické chyby, nebo dokonce nemusí docházet ke konvergenci vůbec. Často síť zjemňujeme lokálně tam, kde očekáváme větší gradient řešených veličin, a dosahujeme tak přesnějších výsledků, aniž bychom přicházeli o příliš mnoho efektivity. Výběr typu sítě a její sestavení je složité téma a to z něj činí samostatný předmět výzkumu.



Obrázek 2.6: Různé typy 2D sítí

Když jsou výpočty hotové, postup pro jednotlivé metody se opět schází při interpretaci jejich výsledků. Používá se postprocessing vizualizace, kdy nám specializované programy (např. Paraview) zobrazí výsledky graficky pomocí barevné škály nebo nástrojů popsaných v sekci 2.2.3. Pro časově závislé děje je možné, aby grafickým výstupem byla animace simulovaného procesu. To pomáhá uživateli lépe pochopit, co se v procesu děje.

2.5.2 Metoda konečných diferencí

Metody konečných diferencí používají Taylorův rozvoj k aproximaci jednotlivých členů v parciální diferenciální rovnici (PDR) [5]. Tím získáme hodnoty stavových veličin a veličin popisujících proudění v uzlech sítě a okolních bodech. Jak to provést závisí na typu aproximovaných rovnic. Narážíme na fakt, že PDR je možné dělit na eliptické, parabolické a hyperbolické. Situace běžného subsonického proudění je popsána pomocí eliptických. V případě složitějšího transonického proudění se prolíná použití eliptických a hyperbolických.

2.5.3 Metoda konečných prvků

Zákony zachování, ze kterých vychází popis dynamiky tekutin, jsou obecně v integrálním tvaru. S přijetím jistých předpokladů je můžeme převést na parciální diferenciální rovnice. Ty v metodě konečných prvků (MKP) převedeme na tzv. **slabou formulaci**. V [16] je popsáno, jak lze přecházet z integrální formy na slabou formulaci bez přijetí předpokladů pro sestavení diferenciálních rovnic a zachovat tak obecnější platnost. Dále je prostorovou a časovou **diskretizací** vygenerován systém algebraických rovnic. Z nich vzniklé matice jsou obvykle řídké a symetrické. To je výhodné z hlediska efektivity a paměťové náročnosti. Výhodou je také flexibilita této metody. S využitím trojúhelníkových elementů (2D) je možné diskretizovat jakkoli geometricky složité problémy. V jednotlivých elementech aproximujeme reálnou fyzikální veličinu použitím tzv. **bázových funkcí** [10].

2.5.4 Metoda konečných objemů

Tato metoda je v CFD zdaleka nejpoužívanější. Využívají ji například oblíbené řešiče OpenFOAM a ANSYS Fluent. Parciální diferenciální rovnice jsou převedeny na algebraické rovnice pomocí integrace přes jednodivé elementy. Systém vzniklých rovnic je pak vyřešen a získáváme hodnoty závislých proměnných pro každý element [14].

2.5.5 Spektrální metoda

Spektrální metoda (SM) vybočuje od dosavadních metod tím, že nevyužívá diskretizaci domény. Podle [11] je považována za rychlou a efektivní metodu s vysokou přesností zejména pro časově periodické proudění při obtékání těles. Řešení hledáme jako superpozici bázových funkcí, obvykle za využití Fourierových, Chebyshevových nebo Legendreových řad. Je ovšem limitována na jednoduché geometrie [12].

2.5.6 Metoda spektrálních elementů

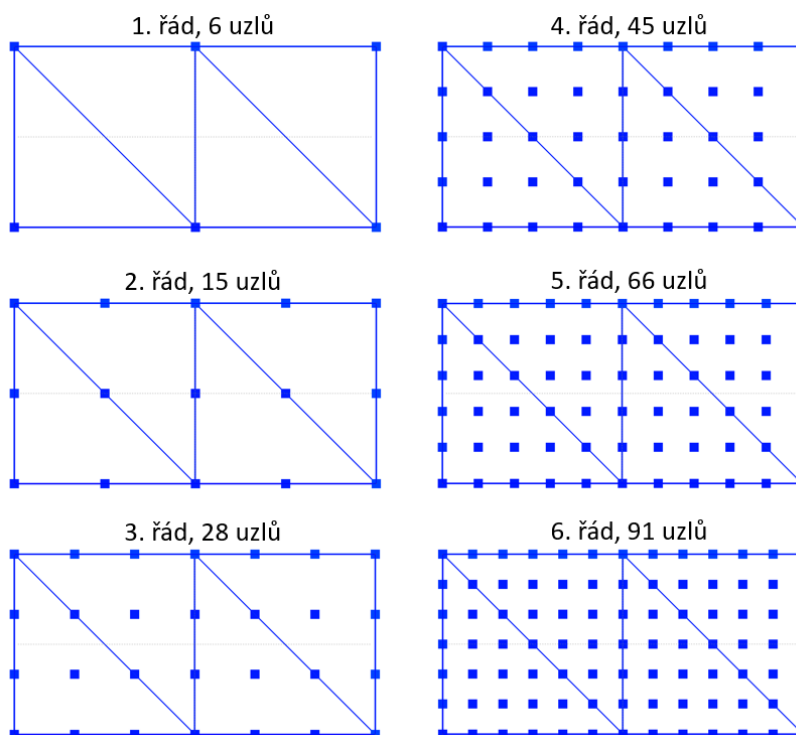
Metodu spektrálních elementů (MSE) publikoval roku 1984 Anthony T. Patera ve svém článku [18]. Jak název napovídá, jedná se o kombinaci MKP a SM. Cílem je využít toho nejlepšího, co každá z metod nabízí, a propojit tak geometrickou flexibilitu metody konečných prvků s přesností a efektivitou spektrální metody.

Výpočetní oblast rozdělíme na elementy, jako při MKP, ale tentokrát může být síť hrubější. Zatímco u klasické MKP je přibližné řešení tvořeno lineární kombinací po částech lineárních funkcí (nebo polynomů nízkého řádu), u MSE se používají Lagrangeovy polynomy vysokého řádu, což kompenzuje snížení počtu uzlů při zvětšení elementů. Jak přibývají uzly s rostoucím řádem polynomu je ukázáno na jednoduché 2D síti o 4 elementech na obrázku 2.7 níže. Kdyby ale byly uzly ekvidistantně rozmístěny jako na obrázku, docházelo by k **Rungeovu jevu**. To je výskyt velkých chyb při aproximaci funkce polynomem na okrajích intervalu. Naznačeno to je na obrázku 2.8 pro červeně vyznačenou funkci $f(x) = \frac{1}{1+25x^2}$ prokládanou modře vyznačeným polynomem. Tato chyba zůstává velká i při zvolení ještě vyšších

řádů polynomu. Zmenšit ji však můžeme chytrou volbou rozmístění prokládaných bodů tak, aby byly více koncentrované u okrajů. V MSE využíváme **Chebyshevovy kolokační body**. Na obrázku 2.8 vidíme jejich rozmístění a mnohonásobně menší chybu aproximace. Rozmístění vychází z goniometrických identit. Funkci $u(x)$ reprezentujeme v i -tém 1D elementu jako Lagrangeův interpolant přes $N_x^i + 1$ bodů

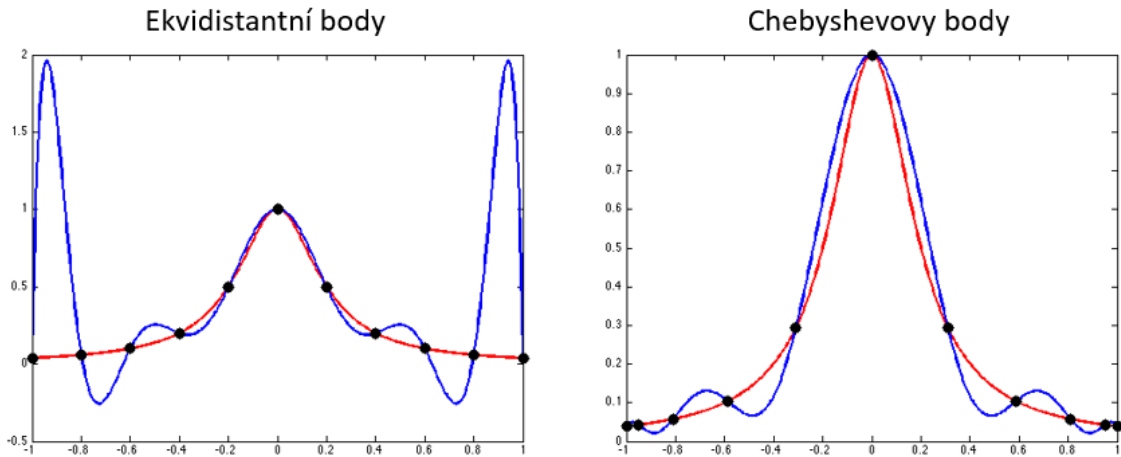
$$x_j^i = \cos \frac{\pi j}{N_x^i}, \quad j = 0, 1, 2, \dots, N_x^i. \quad (2.17)$$

To je výhodná volba, ale ne jediná. Alternativně se k výběru bodů používá Gauss-Lobattova kvadratura.



Obrázek 2.7: Ekvidistantně rozmístěné uzly přibývající s řádem polynomu 1 až 6

Ani metody konečných prvků nemusí zůstat u aproximace lineárními funkcemi. Jedná se o MKP vyšších řádů. h -verze MKP je taková, kde máme daný stupeň báze funkce a přesnost aproximace měníme parametrem h - rozměrem elementu. p -verze MKP naopak má fixovanou jemnost sítě a zpřesňování provádíme změnou řádu polynomu P . V obou případech je možné provést zpřesnění lokálně (velikost elementu a stupně polynomů se můžou pro jednotlivé elementy napříč sítí lišit). Propojením těchto dvou verzí je hp -verze MKP, kde je možné měnit oboje. Je to v podstatě totéž jako metoda spektrálních elementů s tím, že MSE využívá specifické body, jak je popsáno výše. Koeficienty vzniklé expanze následně hledáme pomocí projekce vážených reziduí.



Obrázek 2.8: Rungeho jev při aproximaci funkce polynomem 10. řádu

2.6 Galerkinova metoda

Galerkinova metoda je používána v rámci MKP a MSE k převádění na slabou formulaci. Ukážeme si ji na příkladu lineární diferenciální rovnice

$$L(u) = f \quad (2.18)$$

na oblasti Ω . Rovnici 2.18 vynásobíme tzv. testovací funkcí v a zintegrujeme přes celou oblast Ω :

$$\int_{\Omega} vL(u)dx = \int_{\Omega} vfdx, \quad \forall v \in V. \quad (2.19)$$

Hledáme pak řešení $u \in U$, přičemž U je prostor zkušebních a V testovacích funkcí. To se dá přeformulovat jako

$$a(v, u) = (v, f), \quad \forall v \in V, \quad (2.20)$$

kde (v, f) je skalární součin v a f . Dalším krokem je diskretizace. Namísto hledání řešení u v prostoru nekonečné dimenze U budeme hledat přibližné řešení u^δ v prostoru konečné dimenze $U^\delta \subset U$. Řešení reprezentujeme jako lineární kombinaci bázevých funkcí Φ_n , které tvoří prostor U^δ , např.

$$u^\delta = \sum_{n \in N} \Phi_n \hat{u}_n. \quad (2.21)$$

Nový problém je: Nalezněte $\hat{u}_n (n \in N)$, které splňuje

$$\sum_{n \in N} a(\Phi_m, \Phi_n) \hat{u}_n = (\Phi_m, f), \quad \forall m \in N. \quad (2.22)$$

V maticové formě

$$A\hat{u} = \hat{f}, \quad (2.23)$$

kde \hat{u} je vektor koeficientů \hat{u}_n , A je matice s elementy

$$A[m][n] = a(\Phi_m, \Phi_n) = \int_{\Omega} \Phi_m L(\Phi_n) dx, \quad (2.24)$$

a \hat{f} je vektor daný jako

$$\hat{f}[m] = (\Phi_m, f) = \int_{\Omega} \Phi_m f dx. \quad (2.25)$$

3 Testování solveru využívajícího metodu spektrálních elementů

3.1 Volba solveru

Existuje více knihoven s implementovanými metodami spektrálních elementů nebo MKP vyšších řádů. V zadání byl navržen NEK5000.

Knihovnu NEK5000 [17] se podařilo úspěšně zkompilovat a spustit v ní vzorové simulace. Jako velký problém se ale ukázalo jak generování komplexnějších výpočetních sítí, tak jejich vkládání z jiného software. A vlastně jakékoli další změny. Na vině je nedostatečná dokumentace a málo početná uživatelská komunita, následkem čehož není kde zjišťovat odpovědi na vyvstávající otázky a jak proniknout do složitého systému souborů definujících simulaci.

Rozhodl jsem se proto přejít po doporučení vedoucího práce na jiný solver. Alternativní solvery jsou např.:

- 1) Semtex (implementuje 2D MSE s Fourierovým rozšířením do třetího směru [3])
- 2) HERMES [20] (nabízí značné zlepšení efektivity opoti MKP pro 2D problémy)
- 3) DUNE [9]
- 4) deal.II [2]
- 5) NEKTAR++

Zvolil jsem NEKTAR++. Je zdarma dostupný na stránkách [1] a už na první pohled je zpracovaný lépe než Nek5000 a také využívá metodu spektrálních elementů, je tedy adekvátní náhradou a práce byla provedena na něm.

3.2 Nektar++

Nektar++ je open-source knihovna sloužící k numerickému simulování fyzikálních problémů. Je napsaný v jazyce C++. Umožňuje řešit parciální diferenciální rovnice, využívá metodu spektrálních elementů a má dobré vlastnosti z hlediska škálovatelnosti, viz sekce 3.8. Podle [7] poskytuje lepší vlastnosti než klasické MKP jak výpočetně za využití klasických CPU, tak i z hlediska paměťové náročnosti a využívání dalších výpočetních platforem jako vícejádrové GPGPU (general-purpose graphics processing units). Nabízí diskretizaci spektrálních elementů libovolného řádu pro jednu, dvě i tři dimenze. Má implementovanou Galerkinovu spojitou, nespojitou a hybridní nespojitou metodu projekce. Podporuje různé lineární solvery a předpodmiňovače a i různé časové integrace. První verze Nektaru++ byla vydána v květnu

roku 2006 a je stále aktivně vyvíjen týmy z Imperial College London a University of Utah. V práci je použita současná nejnovější verze 5.0.0 z prosince 2019. Zahrnuje několik solverů:

- 1) Acoustic Solver
- 2) Advection-Diffusion-Reaction Solver
- 3) Cardiac Electrophysiology Solver
- 4) Compressible Flow Solver
- 5) Incompressible Navier-Stokes Solver
- 6) Linear Elasticity Solver
- 7) Pulse Wave Solver
- 8) Shallow Water Solver

Zároveň umožňuje vytváření dalších solverů. V této práci se zaměřujeme na konkrétní z existujících - Incompressible Navier-Stokes Solver.

3.3 Použitý hardware a software

Veškerá práce byla provedena na notebooku HP ENVY x360 15-u200nc vybaveném procesorem Processor Intel(R) Core(TM) i5-5200 CPU @ 2.20GHz, 2201 MHz, 2 Core(s), 4 Logical Processor(s), 8 GB RAM, 256 GB SSD diskem a grafickou kartou Intel® HD 5500.

Přestože Nektar++ lze používat také skrze Windows, OS X a Fedoru, tak distribuce Linuxu založené na Debian a Ubuntu jsou optimální a nejpoužívanější. Operační systém používaného notebooku byl upgradován na Windows 10 Enterprise za účelem využití vestavěného nástroje Hyper-V sloužícího k vytváření přidáných virtuálních systémů v hostitelském systému na jednom zařízení. Ukázalo se ale, že je teprve ve vývoji a nenabízí snadno dostupným způsobem funkce potřebné pro běžné fungování (běží pomalu, problém s přenosem souborů mezi hostitelem a virtuálním systémem, problémy s rozlišením atd.) Naopak jako spolehlivý se ukázal Oracle VM VirtualBox 6.0. Jako operační systém virtuálního zařízení byl použit Ubuntu 18.04.3 LTS bionic. Dalšími použitými programy jsou Gmsh pro vytváření geometrií a sítí, Paraview pro vizualizaci výsledků simulací.

3.4 Kompilace, tutoriály

Nektar++ je open-source. Nabízí tedy zdrojové kódy přímo ke stažení. Pro Debian a Ubuntu nabízí také předkompilované binární soubory. Ty je vhodné použít, chce-li uživatel využívat solver bez dalšího vyvíjení kódu. Je to tedy ideální volba pro případ této práce. Instalace byla provedena následujícím postupem v terminálu:

1. Aktualizujeme a nainstalujeme GPG:

```
sudo apt-get update && apt-get install wget gpg
```

2. Stáhneme klíč pro ověření z odkazu na stránky Nektaru++:

```
sudo wget http://www.nektar.info/nektar-apt.gpg -O - |
sudo apt-key add -
```

3. Do seznamu apt zdrojů /etc/apt/sources.list.d přidáme odkaz na Nektar++ pro požadovanou distribuci:

```
sudo su -c "echo 'deb http://www.nektar.info/ubuntu-
bionic bionic contrib' > nektar.list"
```

4. Nakonec aktualizujeme a spustíme instalaci:

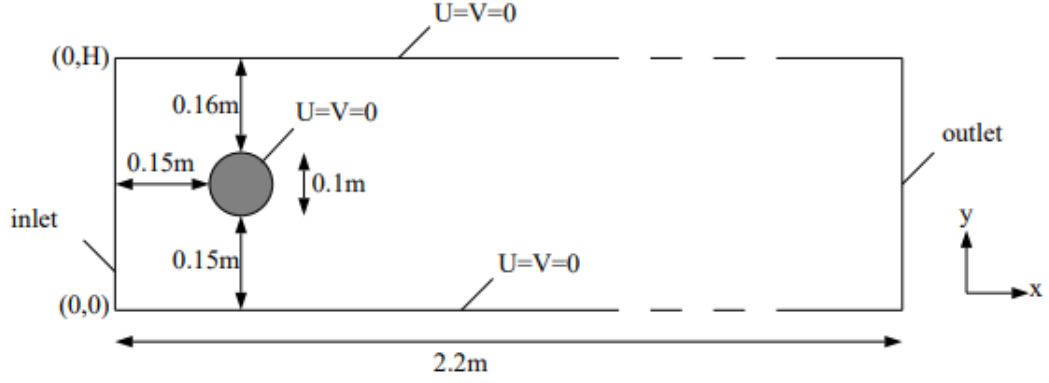
```
sudo apt-get update && sudo apt-get install nektar++
```

Po instalaci je k dispozici řada tutoriálů. Několik se nachází na stránkách [1] v sekci GETTING STARTED / TUTORIALS. Tyto tutoriály jsou podrobně rozebrané včetně matematického popisu problému, práce se sítí, určení počátečních a okrajových podmínek, zvolení časových parametrů, řešení rovnice, použité projekce atd. Pomocí těchto tutoriálů uživatel ověří správnost instalace a udělá si představu o práci s knihovnou. Možnosti knihovny jsou dále popsány v uživatelské příručce (User guide) dostupné na stejných stránkách v sekci DOWNLOADS. Příručka zároveň obsahuje popis hotových vzorových příkladů (Examples) pro jednotlivé solvery. Tyto připravené příklady najdeme ve webovém repozitáři GitLab.

3.5 Benchmarkový problém

Pro ověření správnosti výsledků a pro porovnání s dalšími metodami bylo využito benchmarkových dat publikovaných v [19]. Ty definují společné zadání pro několik různých případů laminárního proudění okolo válce. Na řešení se podílelo celkem 17 vědeckých skupin, které řešily některé ze zadaných případů za využití různých numerických metod a přístupů. Jejich výsledky a výpočetní časy jsou uvedeny v tomto článku. Účelem bylo najít odpovědi na řadu otázek ohledně postupů při simulacích proudění, které byly v době vzniku článku (1996) aktuální a důležité pro udání směru, kterým se ohledně používaných metod vydat.

Já jsem se zaměřil na dva dvoudimenzionální případy: stacionární proudění při $Re = 20$ a nestacionární při $Re = 100$. Oba případy mají společnou geometrii (viz Obr. 3.1). $H=0,41$ m je šířka kanálu a $D=0,1$ m je průměr válce. Tekutiny jsou považovány za izotermické, nestlačitelné a jsou tedy popsány Navier-Stokesovými rovnicemi:



Obrázek 3.1: Geometrie [19]

$$\frac{\partial U_i}{\partial x_i} = 0 \quad (3.1)$$

$$\rho \frac{\partial U_i}{\partial t} + \rho \frac{\partial}{\partial x_j} (U_j U_i) = \rho \nu \frac{\partial}{\partial x_j} \left(\frac{\partial U_i}{\partial x_j} + \frac{\partial U_j}{\partial x_i} \right) - \frac{\partial P}{\partial x_i},$$

kde t je čas, ρ hustota, ν kinematická viskozita, $(x_1, x_2, x_3) = (x, y, z)$ jsou kartézské souřadnice, P je tlak a $(U_1, U_2, U_3) = (U, V, W)$ jsou složky rychlosti. Velikosti kinematické viskozity a hustoty jsou dány jako $\nu = 10^{-3} \text{ m}^2\text{s}^{-1}$ a $\rho = 1 \text{ kg/m}^3$.

Na vstupu je plně vyvinutý parabolický profil rychlosti. Střední rychlost je tedy určena jako

$$\bar{U}(t) = 2U(0, H/2, t)/3. \quad (3.2)$$

Aerodynamická odporová a aerodynamická vztlaková síla jsou dány jako

$$F_D = \int_S (\rho \nu \frac{\partial v_t}{\partial n} n_x - P n_x) dS, \quad F_L = - \int_S (\rho \nu \frac{\partial v_t}{\partial n} n_y + P n_y) dS. \quad (3.3)$$

S je obsah válce, n_x a n_y jsou složky normálového vektoru k ploše S , v_t je tečná složka rychlosti k S a $t = (n_y, -n_x)$ tečný vektor. Odporový a vztlakový koeficient jsou

$$c_D = \frac{2F_D}{\rho \bar{U}^2 D}, \quad c_L = \frac{2F_L}{\rho \bar{U}^2 D}. \quad (3.4)$$

Strouhalovo číslo je dáno jako

$$St = \frac{Df}{\bar{U}}, \quad (3.5)$$

kde f je frekvence odtrhávání vírů. Délka recirkulační zóny je

$$L_a = x_r - x_e, \quad (3.6)$$

kde $x_e = 0,25$ je x-ová souřadnice konce válce a x_r x-ová souřadnice konce recirkulační zóny. Dále určujeme rozdíl tlaků ΔP :

$$\Delta P = \Delta P(t) = P(x_a, y_a, t) - P(x_e, y_e, t). \quad (3.7)$$

Zde $(x_a, y_a) = (0, 15; 0, 2)$ jsou souřadnice začátku a $(x_e, y_e) = (0, 25; 0, 2)$ souřadnice konce válce. Rychlostní okrajová podmínka na vstupu je

$$U(0, y) = 4U_m y(H - y)/H^2, \quad V = 0, \quad (3.8)$$

kde U_m je maximální rychlost. Pro stacionární případ $U_m = 0,3 \text{ ms}^{-1}$, pro nestacionární $U_m = 1,5 \text{ ms}^{-1}$.

Pro vyhodnocení stacionární simulace je potřeba vypočítat následující hodnoty po ustálení: c_D , c_L , L_A , ΔP .

Pro nestacionární: c_{Dmax} , c_{Lmax} , St , ΔP v čase $t = t_0 + 1/2f$, přičemž t_0 je čas při dosažení c_{Lmax} .

3.6 Realizace simulace a práce s knihovnou

Simulaci vytváříme v **.xml souboru** (Extensible Markup Language). Je to datový typ podobný HTML, který ale používá vlastní tagy pro vytvoření objektů a dat v objektech. Protože jsou XML soubory formátovány jako textové dokumenty, mohou být otevřeny a upravovány ve většině textových editorů. Mohou být čteny uživatelem i počítačem. Soubor musí vždy začínat root-ovským tagem `<NEKTAR>`. Tomu ještě může předcházet verze a kódování souboru `<?xml version="1.0" encoding="utf-8"?>`. Struktura souboru vypadá takto:

```
<?xml version="1.0" encoding="utf-8"?>
<NEKTAR>
  <EXPANSIONS>
</EXPANSIONS>
  <CONDITIONS>
</CONDITIONS>
  <GEOMETRY>
</GEOMETRY>
</NEKTAR>
```

Pro přehlednost může být soubor rozdělen na více souborů (typicky geometrie a síťování v jednom souboru, zbytek ve druhém souboru). Každý z těchto souborů musí obsahovat root-ovský tag `<NEKTAR>`, ve kterém jsou umístěny libovolné podkategorie z celkové struktury.

V sekci EXPANSIONS specifikujeme polynomy aproximující řešení. Můžeme volit různý řád a různý způsob rozmístění prokládaných bodů. To můžeme navolit odlišně pro jednotlivé části domény a také zvlášť pro rychlost a tlak.

```

<EXPANSIONS>
  <E COMPOSITE="C[1]" NUMMODES="5" FIELDS="u,v,p" TYPE="
    MODIFIED" />
</EXPANSIONS>

```

V sekci GEOMETRY je zadána dimenze elementů a dimenze prostoru, ve kterém existují. Dále obsahuje seznam bodů, hran, povrchů, elementů, zakřivení, kompozitů (označených částí geometrie, které jsou využity později např. při zadávání okrajových podmínek) a domén. Notace je následující:

```

<GEOMETRY DIM="2" SPACE="2">

  <VERTEX>
    <V ID="0"> 0.0 0.0 0.0 </V>
    <V ID="1"> 0.5 0.0 0.0 </V>
    ...

  </VERTEX>

  <EDGE>
    <E ID="0"> 0 1 </E>
    <E ID="1"> 1 2 </E>
    ...

  </EDGE>

  <ELEMENT>
    <Q ID="0"> 0 3 5 2 </Q>
    <Q ID="1"> 1 4 6 3 </Q>
    ...

  </ELEMENT>

  <COMPOSITE>
    <C ID="0"> Q[0-3] </C>
    <C ID="1"> E[0,1,10,11] </C> <!-- Walls -->
    <C ID="2"> E[2,7] </C> <!-- Inflow -->
    <C ID="3"> E[4,9] </C> <!-- Outflow -->
  </COMPOSITE>

  <DOMAIN> C[0] </DOMAIN>

</GEOMETRY>

```

Body jsou dány kartézskými souřadnicemi, hrany dvojicemi bodů, Tri-elementy trojicemi bodů, Quad-elementy čtveřicemi bodů, kompozity nejčastěji výčtem hran nebo elementů, doména pomocí kompozitů. (Popis není kompletní, ale je dostatečný pro sítě použité v této práci a k ukázaní způsobu zápisu.) Pro jemné sítě je výhodou možnost uložení seznamů těchto prvků v komprimované binární podobě.

Příkaz NekMesh slouží k převedení sítě vytvořené v Gmsh s příponou .msh do formátu čitelného pro Nektar++, tedy .xml. Používá se v příkazovém řádku následovně:

```
NekMesh file .msh newFile.xml
```

Defaultně komprimuje obsah do binárního formátu. Pokud chceme čitelný ASCII formát, provedeme to takto:

```
NekMesh file .msh newFile.xml:xml:uncompress
```

V sekci CONDITIONS definujeme parametry a okrajové podmínky, které určují podstatu a způsob řešení problému. V následující části jsou popsány podseky této sekce s příklady částí kódů využitých v této práci.

1. SOLVERINFO - Zde volíme typ rovnice, typ solveru, projekci, časovou integrační metodu a další.

```
<SOLVERINFO>
  <I PROPERTY="SolverType" VALUE="
    VelocityCorrectionScheme" />
  <I PROPERTY="EQTYPE" VALUE="UnsteadyNavierStokes" />
  <I PROPERTY="EvolutionOperator" VALUE="SkewSymmetric"
    />
  <I PROPERTY="Projection" VALUE="Galerkin" />
  <I PROPERTY="TimeIntegrationMethod" VALUE="IMEXOrder1
    " />
</SOLVERINFO>
```

2. PARAMETERS - Dle typu solveru zadáváme fyzikální a časové parametry (např. kinematickou viskozitu, časový krok, počet časových kroků).

```
<PARAMETERS>
  <P> TimeStep = 0.001      </P>
  <P> NumSteps = 8000      </P>
  <P> IO_CheckSteps = 8000 </P>
  <P> IO_InfoSteps = 100   </P>
  <P> Kinvis = 0.001       </P>
  <P> Re = 20              </P>
```

```
</PARAMETERS>
```

3. VARIABLES - Tady definujeme proměnné. Každé přiřadíme jméno a číslo.

```
<VARIABLES>
  <V ID="0"> u </V>
  <V ID="1"> v </V>
  <V ID="2"> p </V>
</VARIABLES>
```

4. GLOBALSYSSOLNINFO - V této části volíme způsob pro řešení globálního systému, předpokmiňovač a toleranci konvergence. V případě Incompressible Navier-Stokes Solveru můžeme parametry nastavit pro rychlost a tlak odlišně.
5. BOUNDARYREGIONS - S využitím kompozitů definovaných v sekci GEOMETRY zde přiřadíme čísla oblastem, pro které budeme určovat okrajové podmínky.

```
<BOUNDARYREGIONS>
  <B ID="0"> C[2] </B> <!-- Inflow -->
  <B ID="1"> C[3] </B> <!-- Outflow -->
  <B ID="2"> C[4] </B> <!-- Wall -->
  <B ID="3"> C[5] </B> <!-- Cylinder -->
</BOUNDARYREGIONS>
```

6. BOUNDARYCONDITIONS - Každému definovanému regionu z předchozího bodu určíme okrajové podmínky

```
<BOUNDARYCONDITIONS>
  <REGION REF="0"> <!-- Inflow -->
    <D VAR="u" VALUE="1.2*y*(0.41-y)/0.1681" />
    <D VAR="v" VALUE="0" />
    <N VAR="p" USERDEFINEDTYPE="H" VALUE="0" />
  </REGION>
  <REGION REF="1"> <!-- Outflow -->
    <N VAR="u" VALUE="0" />
    <N VAR="v" VALUE="0" />
    <D VAR="p" VALUE="0" />
  </REGION>
  <REGION REF="2"> <!-- Wall -->
    <D VAR="u" VALUE="0" />
    <D VAR="v" VALUE="0" />
    <N VAR="p" USERDEFINEDTYPE="H" VALUE="0" />
  </REGION>
  <REGION REF="3"> <!-- Cylinder -->
```



```

    <D VAR="u" VALUE="0" />
    <D VAR="v" VALUE="0" />
    <N VAR="p" USERDEFINEDTYPE="H" VALUE="0" />
  </REGION>
</BOUNDARYCONDITIONS>

```

Další sekce na stejné úrovni jako CONDITIONS je sekce FILTERS - umožňuje vypočítat různé kvantify z výsledků simulace jak se vyvíjí v čase, například jednoduše hodnotu proměnné v nějakém bodě (sledování tlaku před a za obtékaným válcem), nebo složitější výpočty (aerodynamické síly). Uvedené veličiny bylo potřeba počítat pro porovnání s benchmarkovými daty.

```

<FILTERS>

  <FILTER TYPE="AeroForces">
    <PARAM NAME="OutputFile">DragLift </PARAM>
    <PARAM NAME="OutputFrequency">1000 </PARAM>
    <PARAM NAME="Boundary"> B[3] </PARAM>
  </FILTER>

  <FILTER TYPE="HistoryPoints">
    <PARAM NAME="OutputFile">TimeValues </PARAM>
    <PARAM NAME="OutputFrequency">1000 </PARAM>
    <PARAM NAME="Points">
      0.15 0.20 0
      0.25 0.20 0
    </PARAM>
  </FILTER>

</FILTERS>

```

Některé z uvedených údajů není nutné uvádět. V tom případě je zvolena defaultní možnost. Je nad rámec této práce podrobně rozebírat všechny možnosti knihovny. Jsou zde uvedeny hlavně obecné informace a informace, které byly potřebné k realizaci zadané simulace viz odstavec 3.5.

Příkaz **FieldConvert** nám poslouží především ke konvertování binárních výstupů simulací .fld nebo .chk do formátu .vtu/.dat/.plt čitelného pro software k vizualizaci a post-processingu (ParaView/VisIt/Tecplot). Příkaz v základní podobě pak vypadá následovně:

```
FieldConvert in1.xml in2.fld out.vtu
```

Často převádíme velké množství souborů, použijeme tedy for cyklus, např. takto:

```
for i in {0..100}; do FieldConvert in1.xml in2_${i}.chk  
out3_${i}.vtu; done
```

Simulaci spustíme příkazem:

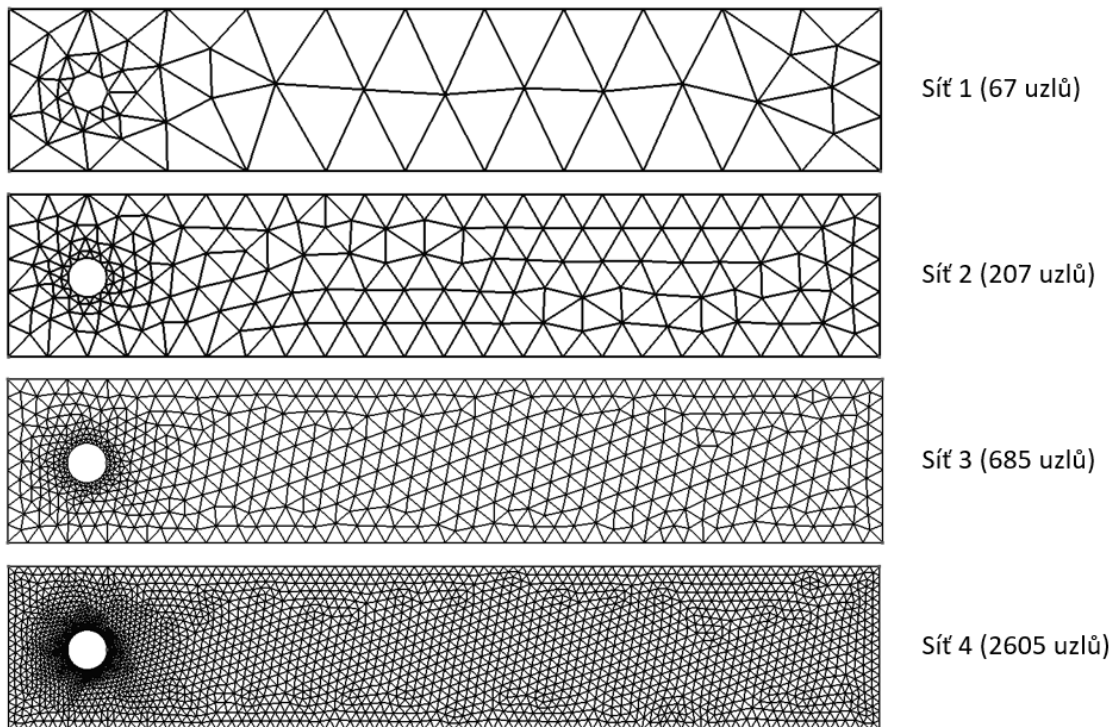
```
IncNavierStokesSolver mesh.xml conditions.xml
```

Zadáme-li jej s parametrem -v, vypíše nám počet stupňů volnosti a shrnutí voleb v SOLVERINFO.

3.7 Výsledky simulace

Simulace byla prováděna na čtyřech trojúhelníkových nestrukturovaných sítích různé jemnosti (viz Obr. 3.2). U všech sítí byla simulace odzkoušena pro různé řády polynomu. Poslední dva řádky každé tabulky uvádí rozmezí výsledků z benchmarkových dat pro snadné porovnání. Všechny simulace byly provedeny s 8000 časovými kroky o délce 0,001 s. Vysvětlivky k následujícím tabulkám s přehledem výsledků:

- řád - řád polynomu
- prostorové proměnné - stupně volnosti (DoF) (3 pro každý uzel - tlak a složky rychlosti)
- t - čas výpočtu



Obrázek 3.2: Použité výpočetní sítě

3.7.1 Stacionární případ $Re=20$

Tabulka 3.1: síť 1

řád	prostorové proměnné	c_D	c_L	L_a [m]	ΔP [Pa]	t [s]
1	201	5,3155	2,5517	-	0,1397	24
2	696	5,0041	1,3305	-	0,1301	58
3	1485	5,2037	0,4490	-	0,1127	77
4	2568	5,1628	0,1328	-	0,1189	102
5	3945	5,2914	0,06375	0,0845	0,1166	142
6	5616	5,2357	0,05675	0,0845	0,1159	187
12	21816	5,3175	0,02023	0,0780	0,1170	732
16	38496	5,3298	0,01437	0,0790	0,1174	1557
20	59880	5,3348	0,01206	0,0790	0,1177	2855
30	133920	5,3392	0,01025	0,0790	0,1180	10306
	spodní mez	5,57	0,0104	0,0842	0,1172	
	horní mez	5,59	0,0110	0,0852	0,1176	

Tabulka 3.2: síť 2

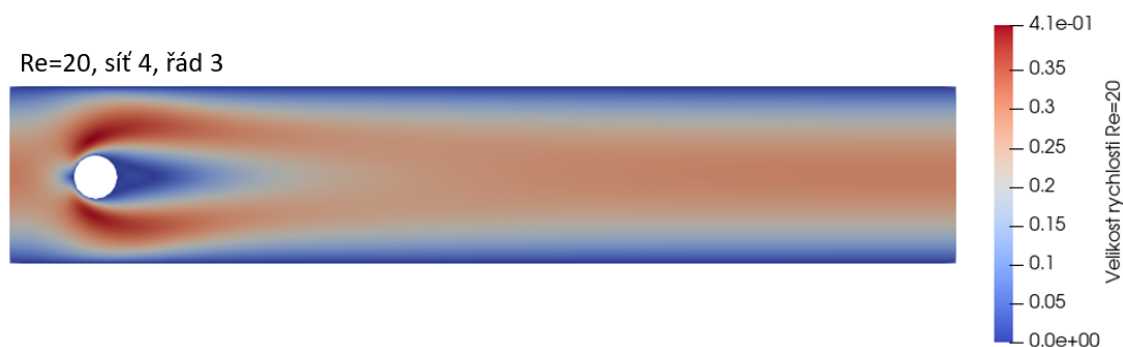
řád	prostorové proměnné	c_D	c_L	L_a [m]	ΔP [Pa]	t [s]
1	621	5,7407	0,1102	-	0,1355	58
2	2277	5,6845	0,0636	-	0,1353	164
3	4968	5,4688	0,0106	0,082	0,1176	239
4	8694	5,4434	0,0092	0,081	0,1145	329
5	13455	5,4676	0,0080	0,083	0,1148	508
6	19251	5,4686	0,0091	0,082	0,1152	660
12	75762	5,4907	0,0096	0,083	0,1150	2596
	spodní mez	5,57	0,0104	0,0842	0,1172	
	horní mez	5,59	0,0110	0,0852	0,1176	

Tabulka 3.3: síť 3

řád	prostorové proměnné	c_D	c_L	L_a [m]	ΔP [Pa]	t [s]
1	685	5,3341	-0,0476	-	0,1199	165
2	7815	5,5809	0,0044	0,079	0,1190	645
3	17280	5,5378	0,0090	0,084	0,1175	811
4	30450	5,5332	0,0100	0,083	0,1169	1291
5	47325	5,5425	0,0100	0,083	0,1169	1641
6	67905	5,5449	0,0100	0,084	0,1168	2167
10	187275	5,5520	0,0105	0,084	0,1168	5699
	spodní mez	5,57	0,0104	0,0842	0,1172	
	horní mez	5,59	0,0110	0,0852	0,1176	

Tabulka 3.4: síť 4

řád	prostorové proměnné	c_D	c_L	L_a [m]	ΔP [Pa]	t [s]
1	2605	5,3813	7.4998	0.082	0.1171	639
2	30450	5,5549	0,0078	0,083	0,1168	2214
3	67905	5,5456	0,0101	0,084	0.1170	3382
4	120180	5,5459	0,0104	0,084	0.1169	4804
10	745050	5,5543	0,0105	0,084	0,1168	25886
	spodní mez	5,57	0,0104	0,0842	0,1172	
	horní mez	5,59	0,0110	0,0852	0,1176	



Obrázek 3.3: Vizualizace výsledného proudění po ustálení pro Re=20

3.7.2 Nestacionární případ Re=100

Vysvětlivky k této části výsledků:

- diverg. - řešení diverguje
- error - solver nedopočítal simulaci

Tabulka 3.5: síť 1

řád	prostorové proměnné	c_D	c_L	L_a [m]	ΔP [Pa]	t [s]
1	201	diverg.	-	-	-	-
2	696	-	-	-	-	53
3	1485	-	-	-	-	74
4	2568	-	-	-	-	100
5	3945	4,4050	1,2133	0,3326	2,8831	131
6	5616	3,7141	1,3553	0,3125	2,4227	173
7	7581	error	-	-	-	-
8	9840	error	-	-	-	-
	spodní mez	3.2200	0,9900	0,2950	2,4600	
	horní mez	3.2400	1,0100	0,3050	2,5000	

Tabulka 3.6: síť 2

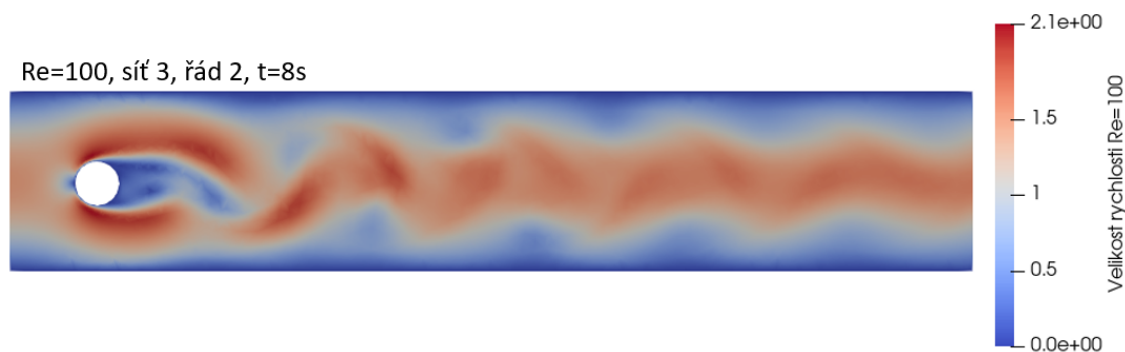
řád	prostorové proměnné	c_D	c_L	L_a [m]	ΔP [Pa]	t [s]
1	621	-	-	-	-	55
2	2277	2,8348	0,9185	0,3030	2,3082	148
3	4968	3,6270	1,1939	0,2941	2,7002	219
4	8694	error	-	-	-	-
	spodní mez	3.2200	0,9900	0,2950	2,4600	
	horní mez	3.2400	1,0100	0,3050	2,5000	

Tabulka 3.7: síť 3

řád	prostorové proměnné	c_D	c_L	L_a [m]	ΔP [Pa]	t [s]
1	685	-	-	-	-	205
2	7815	3,3755	0,7950	0,3125	2,2702	498
3	17280	error	-	-	-	-
4	30450	error	-	-	-	-
	spodní mez	3.2200	0,9900	0,2950	2,4600	
	horní mez	3.2400	1,0100	0,3050	2,5000	

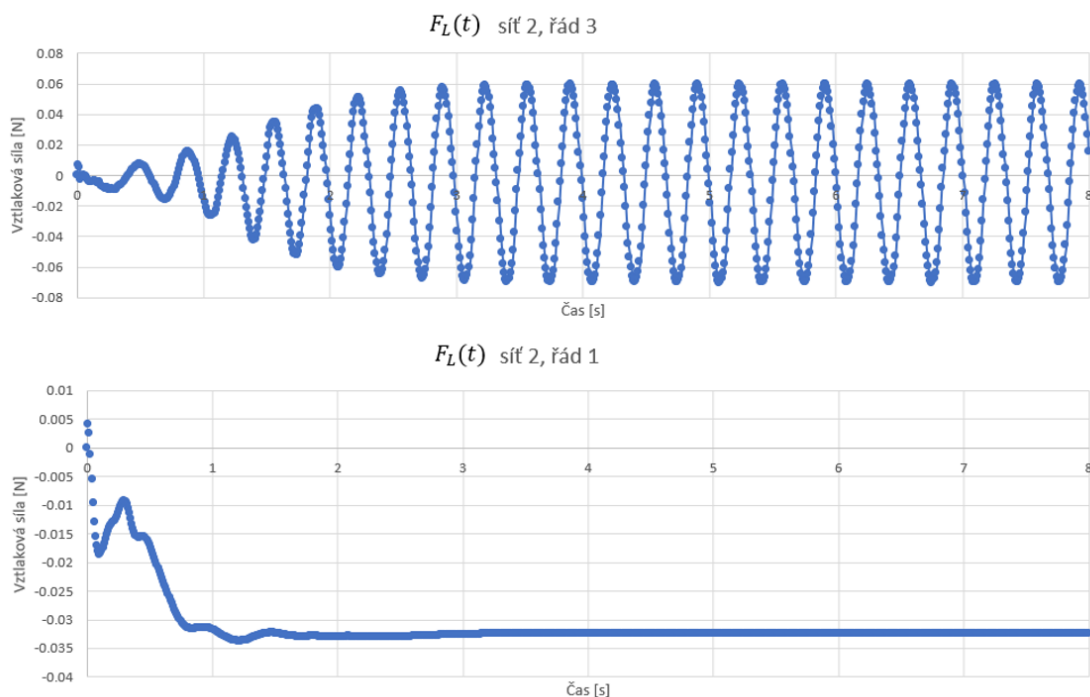
Tabulka 3.8: síť 4

řád	prostorové proměnné	c_D	c_L	L_a [m]	ΔP [Pa]	t [s]
1	2605	2,7157	0,4991	0,3226	2,0822	566
2	30450	error	-	-	-	-
3	67905	error	-	-	-	-
4	120180	error	-	-	-	-
	spodní mez	3.2200	0,9900	0,2950	2,4600	
	horní mez	3.2400	1,0100	0,3050	2,5000	

Obrázek 3.4: Vizualizace výsledného proudění pro Re=100 v čase $t = 8$ sekund

3.8 Paralelní škálování

Při využití dvou procesorů se výpočetní čas simulací ve všech odzkoušených případech prodlužoval oproti času výpočtu na jednom procesoru, viz tabulka 3.9. Příčin může být celá řada. Pravděpodobně problém s kompilací nebo komunikací virtuál-



Obrázek 3.5: Průběh vztlakové síly pro $Re=100$

ního systému s hostitelským. I vlivem vyhlášení nouzového stavu při tvorbě práce nebyla k dispozici jiná zařízení ani výpočetní clustery, na kterých by škálovatelnost bylo možné odzkoušet.

Tabulka 3.9: Výpočetní časy pro 120 časových kroků pro jedno a dvě výpočetní jádra

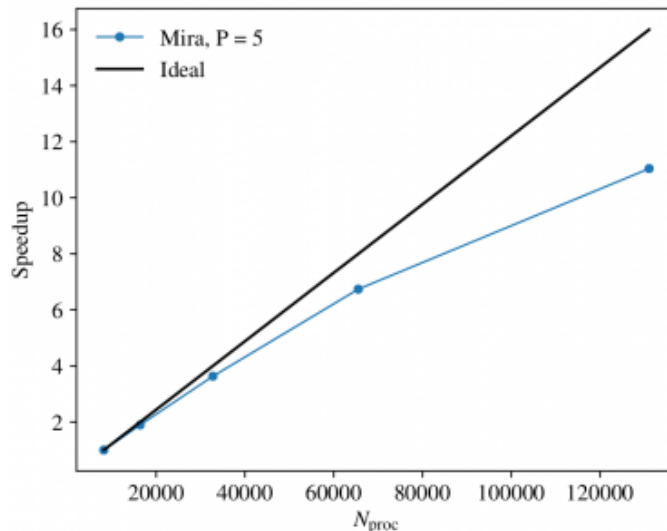
řád	síť 1		síť 2		síť 3	
	1 CPU	2 CPU _s	1 CPU	2 CPU _s	1 CPU	2 CPU _s
1	1 s	1 s	1 s	2 s	3 s	5 s
2	2 s	2 s	3 s	5 s	9 s	18 s
3	2 s	2 s	3 s	7 s	16 s	30 s
4	2 s	3 s	7 s	10 s	25 s	43 s
5	3 s	4 s	10 s	13 s	37 s	59 s
6	4 s	6 s	14 s	20 s	55 s	83 s
7	5 s	7 s	20 s	25 s	80 s	111 s

Práce zabývající se škálovatelností NEKTARu++ ovšem existují [15, 8]. Z nich vyplývá, že paralelní výpočty v Nektaru++ jsou velmi efektivní. Problémem je např. použití XML vstupního formátu při definici rozsáhlých sítí. Není totiž přímá možnost číst pouze vybrané části XML souboru a při počátečním dělení sítě na menší části je potřeba alespoň jednou přečíst celý soubor. V detailních sítích na komplexních geometriích, které mívají počet elementů v řádu milionů, to vede k velké paměťové náročnosti a delšímu výpočetnímu času nutnému k vytvoření rozdělení sítě.

Na stránkách [1] je prezenován výsledek testování paralelizace (viz Obr. 3.6 na výpočtením clusteru Mira až po 131 000 jader se skvělou efektivitou při simulaci nestlačitelného proudění okolo geometrie formule 1. Mira je superpočítač v Argonne National Laboratory. Patří k nejvýkonnějším počítačům na světě a má téměř 800 000 výpočetních jader. Černá přímka zobrazuje ideální případ, kdy je závislost mezi počtem jader a zrychlením výpočetního času lineární. Teoreticky, zdvojnásobíme-li počet jader, čas se zkrátí na polovinu. Prakticky to tak ale nefunguje, protože část výkonu je nutná pro rozdělení výpočtu mezi jádra a pro jejich vzájemnou komunikaci, a dokonce po překročení nějakého počtu jader dochází ke zpomalování výpočtu. Blíže realitě je speed up dle Amdahlova zákona

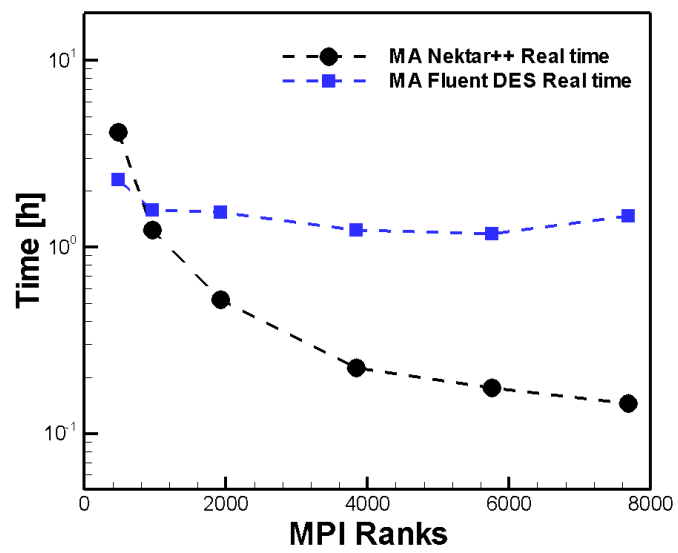
$$S = \frac{1}{(1 - P) + \frac{P}{n}} = \frac{T}{T(s)}, \quad (3.9)$$

kde P je Paralelizovatelná část algoritmu, n je počet procesorů, T je doba výpočtu před zvýšením počtu procesorů a $T(s)$ je doba výpočtu po zvýšení počtu CPUs. Za T je vzána doba prvního výpočtu (při použití 8 192 CPUs).



Obrázek 3.6: Škálovatelnost [1]

Dále uvádíme srovnání [21], jak si stojí Nektar++ oproti komerčně využívanému Fluentu s implementovanou metodou konečných objemů. Porovnán je zde výpočetní čas v závislosti na počtu použitých jader, viz Obr. 3.7. Výpočty byly provedeny na superpočítači Cray XC40. V Nektar++ byl použit třetí řád polynomu pro rychlost a druhý pro tlak. Podmínky simulací byly odlišné. Obě začínají v čas 0,11 s a končí v čas 0,16 s, ale Nektar++ počítal s parametry $Re=100$, časový krok 10^{-5} , počet časových kroků 5000, zatímco Fluent (DES) měl $Re=6.3 \cdot 10^6$, časový krok 10^{-4} , počet časových kroků 500. V [21] je také uvedeno, že není žádný významný rozdíl v časech pro výpočty při obou různých Re .



Obrázek 3.7: Porovnání škálovatelnosti solverů Nektar++ a Fluent [21]

4 Závěr

Jak je naznačeno v sekci 3.1, knihovnu NEK5000, kterou bylo původně zamýšleno využít, je problematické používat jen na základě dostupné dokumentace. Ta se ukázala jako zcela nedostatečná. Oproti tomu Nekar++ splnil všechna očekávání a podal přesné výsledky pro stacionární 2D proudění nestlačitelné tekutiny kolem válce v uzavřeném kanálu při $Re=20$.

V každé tabulce s výsledky v sekcích 3.7.1 a 3.7.2 je v posledních dvou řádcích uveden odhadovaný interval, ve kterém se podle autorů [19] může nacházet přesné řešení. Je vidět, že pro stacionární případ ($Re=20$) se výsledky s rostoucím počtem stupňů volnosti (DoF) ustálí na hodnotách, které neleží přesně v těchto intervalech. Je to pravděpodobně způsobeno tím, že při použití hrubé trojúhelníkové sítě není dostatečně přesně aproximován zakřivený tvar válce. V případě nejméně jemné použité sítě (viz Obr. 3.2) počítáme vlastně obtékání osmiúhelníku, což se liší od kruhu podstatně. Tomuto vysvětlení odpovídá také fakt, že výsledky jemnějších sítí se ustálily blíže intervalům.

Závěrem tedy je, že pro použití vysokých řádů metody spektrálních elementů je vhodnější ručně připravit hrubější sítě se zakřivenými hranami. To by mohl být další předmět zkoumání v návaznosti na tuto práci.

Pro nestacionární případ byl problém, že při překročení počtu stupňů volnosti okolo 7500 - 8000 solver řešení nedopočítal. Pro simulace s nízkým počtem DoF do přibližně 2000 - 3000 bylo řešení natolik nepřesné, že určování srovnávaných parametrů by ani nedávalo smysl. Je to vidět na Obr. 3.5, kde oba grafy znázorňují průběh vztahové síly. Horní je poměrně přesný, zatímco u spodního s méně DoF není možné stanovit frekvenci pro výpočet Strouhalova čísla a ani c_{Dmax} , c_{Lmax} a ΔP o ničem nevypovídají, nejsou proto určeny.

Při zkoumání výhod použití vyšších řádů polynomů oproti zjemňování sítě na této sadě výsledků není zřejmá převaha ani jednoho z přístupů.

Pro testování paralelizace nebyl k dispozici vhodný hardware. Spouštění simulace na dvou jádrech vykazovalo časově delší výpočet než na jednom výpočetním jádře, viz tabulka 3.9. Dohledané zdroje ale uvádí skvělé výsledky při testování na vysoce výkonných výpočetních clusterech o tisících i desetitisících jádrech.

Literatura

- [1] NEKTAR++ spectral/hp element framework. <https://www.nektar.info/>. Accessed: 2020-06-05.
- [2] W. Bangerth, R. Hartmann, and G. Kanschat. Deal.ii—a general-purpose object-oriented finite element library. *ACM Trans. Math. Softw.*, 33(4):24–es, August 2007.
- [3] H.M. Blackburn and S.J. Sherwin. Formulation of a galerkin spectral element–fourier method for three-dimensional incompressible flows in cylindrical geometries. *Journal of Computational Physics*, 197(2):759 – 778, 2004.
- [4] Robert D. Blevins. *Flow-induced vibration*. New York: Van Nostrand Reinhold, 1990.
- [5] Christopher E Brennen. An internet book on fluid dynamics. *INDEX for INTERNET BOOK ON FLUID DYNAMICS*, 2006.
- [6] O. C., R. L. Zienkiewicz, and Taylor. *The finite element method : vol. 1: basic formulation and linear problems*. McGraw-Hill, 1994.
- [7] CD Cantwell, D Moxey, A Comerford, A Bolis, G Rocco, G Mengaldo, Grazia D de, S Yakovlev, J-E Lombard, D Ekelschot, B Jordi, H Xu, Y Mohamied, C Eskilsson, B Nelson, P Vos, C Biotto, RM Kirby, and SJ Sherwin. Nektar++: An open-source spectral/hp element framework. *Computer Physics Communications*, 192:205–219, 2015.
- [8] A Comerford, C Cantwell, SJ Clifford, and SJ Sherwin. Scaling the nektar++ spectral/hp element framework to large clusters.
- [9] A. Dedner, R. Klöfkor, and M. Nolte. A generic interface for parallel and adaptive discretization schemes: abstraction principles and the dune-fem module. pages 165 – 196, 2010.
- [10] Miloslav Feistauer, Jiří Felcman, and Ivan Straškraba. *Mathematical and computational methods for compressible flow*. Oxford University Press on Demand, 2003.
- [11] Arathi Gopinath and Antony Jamesony. Time spectral method for periodic unsteady computations over two and three dimensional bodies. 01 2005.

- [12] S. J., Sherwin, G. E., and Karniadakis. Tetrahedral hp finite elements : Algorithms and flow simulations. 1996.
- [13] Pech Jan. 2d simulation of flow behind a heated cylinder using spectral element approach with variable coefficients. *Programs and Algorithms of Numerical Mathematics*, 17, 2015.
- [14] Fadl Moukalled, Luca Mangani, and Marwan Darwish. *The Finite Volume Method in Computational Fluid Dynamics: An Advanced Introduction with OpenFOAM® and Matlab®*, volume 113. 10 2015.
- [15] David Moxey, Chris D. Cantwell, Yan Bao, Andrea Cassinelli, Giacomo Castiglioni, Sehun Chun, Emilia Juda, EhsanKazemi Kazemi, Kilian Lackhove, Julian Marcon, Gianmarco Mengaldo, Douglas Serson, Michael Turner, Hui Xu, Joaquim Peiró, M. Kirby Robert, and Spencer J. Sherwin. Nektar++: Enhancing the capability and application of high-fidelity spectral/hp element methods. 17, 2019.
- [16] C. W. OSEEN. *Neuere Methoden und Ergebnisse in der Hydrodynamik*. Leipzig, Akademische Verlagsgesellschaft m. b. h., 1927.
- [17] J. Mullen H. Tufo J. Lottes S. Kerkemeier P. Fischer, J. Kruse. NEK5000 open source spectral element cfd solver. <https://nek5000.mcs.anl.gov/>. Accessed: 2020-06-05.
- [18] Anthony T. Patera. A spectral element method for fluid dynamics: Laminar flow in a channel expansion. In *Numerical Modeling of Seismic Wave Propagation: Gridded Two-way Wave-equation Methods*. Society of Exploration Geophysicists, 01 2012.
- [19] M. Schäfer, S. Turek, F. Durst, E. Krause, and R. Rannacher. Benchmark computations of laminar flow around a cylinder. *Flow Simulation with High-Performance Computers II*, 48, 1996.
- [20] Tomáš Vejchodský, Pavel Šolín, and Martin Zítka. Modular hp-fem system hermes and its application to maxwell’s equations. *Mathematics and Computers in Simulation*, 76(1):223 – 228, 2007. Mathematical Modelling and Computational Methods in Applied Sciences and Engineering.
- [21] Sebastian Wagner. Strong scalability test of an automotive use case performed with nektar++ and fluent (des).
- [22] F.M. White. *Fluid Mechanics*. McGraw-Hill series in mechanical engineering. McGraw-Hill, 2008.