



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA STAVEBNÍ

FACULTY OF CIVIL ENGINEERING

ÚSTAV STAVEBNÍ MECHANIKY

INSTITUTE OF STRUCTURAL MECHANICS

**ISOGEOMETRICKÁ ANALÝZA A JEJÍ POUŽITÍ
V MECHANICE KONTINUA**

ISOGEOMETRIC ANALYSIS AND APPLICATIONS IN CONTINUUM MECHANICS

DIPLOMOVÁ PRÁCE

DIPLOMA THESIS

AUTOR PRÁCE

AUTHOR

Bc. Martin Ladecký

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. PETRA ROZEHNALOVÁ, Ph.D.

BRNO 2018



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

FAKULTA STAVEBNÍ

Studijní program	N3607 Stavební inženýrství
Typ studijního programu	Navazující magisterský studijní program s prezenční formou studia
Studijní obor	3608T001 Pozemní stavby
Pracoviště	Ústav stavební mechaniky

ZADÁNÍ DIPLOMOVÉ PRÁCE

Student	Bc. Martin Ladecký
Název	Isogeometrická analýza a její použití v mechanice kontinua
Vedoucí práce	Ing. Petra Rozehnalová, Ph.D.
Datum zadání	31. 3. 2017
Datum odevzdání	12. 1. 2018

V Brně dne 31. 3. 2017

prof. Ing. Drahomír Novák, DrSc.
Vedoucí ústavu

prof. Ing. Rostislav Drochytka, CSc., MBA
Děkan Fakulty stavební VUT

PODKLADY A LITERATURA

J. A. Cottrell, T. J. R. Hughes, Y. Bazilevs, *Isogeometric Analysis: Toward Integration of CAD and FEA*, John Wiley & Sons, 2009.

Alexandre Ern, Jean-Luc Guermond, *Theory and Practice of Finite Elements*

ZÁSADY PRO VYPRACOVÁNÍ

Student se seznámí s problematikou isogeometrické analýzy, NURBS křivek a ploch. Prostuduje vlastnosti analýzy a výsledky bude aplikovat na vhodné úlohy z mechaniky kontinua. Práce bude doplněna implementací a testovacími výpočty.

STRUKTURA DIPLOMOVÉ PRÁCE

VŠKP vypracujte a rozčleňte podle dále uvedené struktury:

1. Textová část VŠKP zpracovaná podle Směrnice rektora "Úprava, odevzdávání, zveřejňování a uchování vysokoškolských kvalifikačních prací" a Směrnice děkana "Úprava, odevzdávání, zveřejňování a uchování vysokoškolských kvalifikačních prací na FAST VUT" (povinná součást VŠKP).
2. Přílohy textové části VŠKP zpracované podle Směrnice rektora "Úprava, odevzdávání, zveřejňování a uchování vysokoškolských kvalifikačních prací" a Směrnice děkana "Úprava, odevzdávání, zveřejňování a uchování vysokoškolských kvalifikačních prací na FAST VUT" (nepovinná součást VŠKP v případě, že přílohy nejsou součástí textové části VŠKP, ale textovou část doplňují).

Ing. Petra Rozehnalová, Ph.D.
Vedoucí diplomové práce

ABSTRAKT

Táto práca sa zaoberá riešením úloh mechaniky kontinua Isogeometrickou metódou. Jedná sa o pomerne mladý postup, ktorý spája výhody presnej NURBS geometrie a robustnosť klasickej metódy konečných prvkov. Metódu sme v práci podrobne rozobrali a popísali na riešení rovinatej Poissonovej okrajovej úlohy. Isogeometrickú metódu sme implementovali v programe MatLab a jednotlivé algoritmy sme priložili k textu.

KLÚČOVÉ SLOVÁ

Isogeometrická analýza-IGA, Galerkinova metóda, NURBS geometria, Poissonova okrajová úloha, mechanika kontinua

ABSTRACT

This thesis deals with solving the problems of continuum mechanics by method of Isogeometric analysis. This relatively young method combines the advantages of precise NURBS geometry and robustness of the classical finite element method. The method is described on procedure of solving a plane Poissons boundary value problem. Solver is implemented in MatLab and algorithms are attached to the text.

KEYWORDS

Isogeometric analysis-IGA, Galerkin method, NURBS geometry, Poissons boundary value problem, continuum mechanics

ISOGEOMETRICKÁ ANALÝZA A JEJÍ POUŽITÍ V MECHANICE KONTINUA:
diplomová práce. Brno: Vysoké učení technické v Brně, Fakulta stavební, Ústav
stavební mechaniky, 2017/2018. 65 s. Vedúci práce bol Ing. PETRA ROZEHNALOVÁ, Ph.D.

Prehlásenie:

Prehlasujem, že som svoju diplomovú prácu na tému „ISOGOMETRICKÁ ANALÝZA A JEJÍ POUŽITÍ V MECHANICE KONTINUA“ vypracoval samostatne pod vedením vedúceho diplomovej práce, využitím odbornej literatúry a ďalších informačných zdrojov, ktoré sú všetky citované v práci a uvedené v zozname literatúry na konci práce.

Ako autor uvedenej diplomovej práce ďalej prehlasujem, že v súvislosti s vytvorením tejto diplomovej práce som neporušil autorské práva tretích osôb, najmä som nezasiahol nedovoleným spôsobom do cudzích autorských práv osobnostných a/nebo majetkových a som si plne vedomý následkov porušenia ustanovenia § 11 a nasledujúcich autorského zákona č. 121/2000 Sb., o právu autorskom, o právach súvisiacich s právom autorským a o zmeně niektorých zákonov (autorský zákon), vo znení neskorších predpisov, vrátane možných trestnoprávných dôsledkov vyplývajúcich z ustanovenia časti druhej, hlavy VI. diel 4 Trestného zákoníka č. 40/2009 Sb.

Brno

.....

Bc. Martin Ladecký
(podpis autora)

Podakovanie:

Za finančnú podporu ďakujem Grantovej agentúre českej republiky, projekt číslo 17-04150J.

Vedúcej diplomovej práce, Pani Ing. PETRE ROZEHNALOVEJ, Ph.D. z katedry matematiky VUT FAST v BRNE, by som v prvom rade chcel poďakovať za priateľský prístup. Ten mi v mnohých oblastiach uľahčil prekonávanie komplikácií pri tvorbe tejto práce. Ďalej za odborné vedenie, konzultácie a množstvo času, ktoré mi venovala. Za odborné rady v oblasti matematiky, a mnoho užitočných rád pri koncipovaní práce. Taktiež za obrovskú trpezlivosť v snahe pomôcť mi pochopiť Isogeometrickú metódu a metódu konečných prvkov.

Ďalej by som sa chcel poďakovať Doc. Ing. JÁNOVI ZEMANOVI, Ph.D. z katedry mechaniky FSv ČVUT v PRAHE, za odborný dohľad nad celkovým priebehom. Konzultácie, počas ktorých mi poskytol množstvo cenných rád. Takisto za revíziu práce a pomoc pri úprave textu.

Rovnako veľká vďaka patrí mojej priateľke Veronike Panáčkovej. Jej láska a neustála podpora umožnila vznik tejto práce.

V neposlednom rade chcem poďakovať svojim rodičom, ktorí ma vždy neúnavne podporovali a poskytli mi tak skvelé zázemie po celú dobu štúdia.

OBSAH

1	Úvod	1
2	Galerkinova metóda	3
2.1	Diferenciálna formulácia	3
2.2	Slabá formulácia	4
2.3	Galerkinova metóda	5
3	Postup výpočtu	8
3.1	Lokalizované bázové funkcie	8
3.2	Lokálna matica tuhosti	9
3.3	Algoritmus výpočtu	11
4	Oblasť Ω ako NURBS plocha	13
4.1	Motivácia	13
4.2	NURBS krivka	14
4.3	NURBS bázové funkcie	16
4.3.1	B-spline báza	16
4.3.2	NURBS báza	19
4.4	NURBS plocha	23
5	Isogeometrická analýza-IGA	28
5.1	Isoparametrický prístup	29
5.2	Element v Isogeometrickej metóde	31
5.3	Derivácie NURBS bázových funkcií	34
5.4	Jakobiho matica	38
5.5	Výpočet vektora pravých strán	38
5.6	Výpočet matice tuhosti	42
5.7	Vyriešenie sústavy	44
6	Aplikácie metódy na úlohy mechaniky kontinua	45
6.1	Voľné krútenie prútu nekruhového prierezu	47
6.2	Rovinná napätosť	53
7	Záver	55
A	Príloha	57
A.1	Transformácia súradnej sústavy	57
A.1.1	Transformácia integrálov	57

A.1.2	Parciálne derivácie zložených funkcií	58
A.1.3	Zostavenie lokálneho vektoru pravých strán	60
A.1.4	Zostavenie lokálnej matice tuhosti	62
B	Príloha	65
B.1	Gaussove body a váhy	65

ZOZNAM OBRÁZKOV

3.1	Funkcie s lokálnym nosičom.	9
3.2	Rozdelenie oblasti Ω na elementy.	10
3.3	Algoritmus výpočtu	12
4.1	Všeobecná rovinná krivka.	14
4.2	NURBS krivka.	15
4.3	Uniformné B-spline funkcie.	17
4.4	Neuniformné B-spline funkcie.	18
4.5	Závislosť NURBS krivky na hodnote váhového koeficientu.	20
4.6	Závislosť NURBS funkcií na hodnote váhového koeficientu.	21
5.1	Zostrojenie Ω transformáciou $\hat{T} : \hat{\Omega} \rightarrow \Omega$	29
5.2	Členenie $\hat{\Omega}$ na elementy $\hat{\Omega}^e$	31
5.3	Postupná transformácia elementu.	39
6.1	Skrútený prizmatický prút.	47
6.2	NURBS bazové funkcie k popisu prierezu Ω	48
6.3	Členenie profilu na elementy	48
6.4	Riadiaca sieť pre medzikružie.	49
6.5	Približné výsledky Poissonovej okrajovej úlohy.	50
6.6	Štvorcová stena.	53
6.7	Relatívna odchýlka riešenia.	54
6.8	Riešenie úlohy rovinného napätia.	54
A.1	Transformácia sústavy	58
A.2	Transformácia elementu z parametrickej do kartézskej súradnej sústavy.	61
A.3	Postupná transformácia elementu.	62

1 ÚVOD

Pred pár rokmi sme nastúpili na stavebnú fakultu začali sme sa zoznamovať s inžinierkou prácou. Už po pár semestroch sme sa začali rozdeľovať na projektantov, ktorí navrhovali tvary budov a tých čo sme mali analyzovať ich návrhy. Projektanti pracovali výlučne v CAD-ových systémoch, ktoré umožňovali krásne a presne reprezentovať ich myšlienky. My sme sa naproti tomu učili modelovať v konečne prvkových programoch, ktoré dokázali ich návrhom vdýchnuť fyzikálne správanie. Aj keď sme spracovali na jednom projekte, museli sme si každý vytvoriť svoj vlastný model. Zabezpečiť aktualizáciu modelu pri každej zmene, bolo priam nemožné. Už vtedy sme sa pýtali, čím to je, že nemôžeme pracovať spolu. Prečo nemožno vytvárať geometricky presné objekty vhodné k vizualizácii a súčasne posudzovať fyzikálne vlastnosti v jednom, a tom istom programe. Postupom času sem sa pýtať prestali a adoptovali sme názor, že to jednoducho nejde.

Prekvapenie pre mňa nastalo, keď som sa približne pred rokom dostal ku knihe „Isogeometric analysis - Toward Integration of CAD and FEA¹“ z roku 2009. V nej pán Thomas J.R.Hughes so svojimi spoluautormi tvrdili opak. Už v úvode písali, že prišli na spôsob, akým premostiť priepasť medzi CAD a MKP softvérmi. Ako spojiť proces návrhu a analýzy, čím by sa ušetrili nespočetné hodiny práce pri vytváraní duplicitných modelov. Táto myšlienka ma nadchla natoľko, že som sa rozhodol venovať Hughesovej metóde a zvolil som si ju ako tému na diplomovú prácu.

Isogeometrická analýza, v literatúre často označovaná aj IGA, je spôsob akým pristupovať k riešeniu fyzikálnych problémov. Nejedná sa ani tak o konkrétnu numerickú metódu, ale skôr o možnosť ako vylepšiť už existujúce postupy. Isogeometrický prístup je aplikovateľný na Galerkinovu, Ritzovu, kolokačnú a mnohé iné metódy. Značný potenciál si začínajú uvedomovať mnohí odborníci. Nie je preto divu, že jej v posledných rokoch venujú veľkú pozornosť. Podľa našich informácií zatiaľ neexistuje komerčný softvér, ktorý by využíval isogeometrickú technológiu. Vzhľadom na to, že sa jedná o pomerne novú metódu sa tomu nemôžeme čudovať.

Hughesovu knihu [3] vrelo doporučujeme. Musíme však priznať, že nám nestačila a museli sme siahnuť aj po inej literatúre. To nás viedlo k rozhodnutiu napísať túto prácu ako učebný text. Snažili sme sa čo najjednoduchšie a najzreteľnejšie popísať celý proces. Od nastolenia problému popísaného diferenciálnou rovnicou, až po jej vyriešenie. V texte sú priložené aj kusy kódu z programu MatLab, ktorý môže čitateľovi slúžiť ako vôdzka pri snahe o vlastnú implementáciu.

Po týchto úvodných slovách si v druhej kapitole dáme za cieľ vyriešiť Poissonovu okrajovú úlohu. Bude sa jednať o parciálnu diferenciálnu rovnicu, riešenú na rovin-

¹FEA-Finite element analysis je anglické označenie pre konečne prvkovú analýzu.

nej oblasti. Tá má širokú škálu využitia nielen v mechanike kontinua, ale aj iných oblastiach vedy. Nám poslúži ako názorný príklad vďaka svojej jednoduchosti. Postupne odvodíme slabú formuláciu a ukážeme ako k riešeniu pristupuje Galerkinova metóda. Tá k hľadaniu riešenia využíva bázové funkcie. Od výberu bázových funkcií závisí správnosť riešenia. Postupu a vlastnostiam bázových funkcií budeme venovať tretiu kapitolu. V štvrtej na chvíľu odbočíme a zameriame sa na popis geometrie pomocou NURBS funkcií. NURBS geometria sa využíva v mnohých CAD-ových systémoch a podľa [3] a [9] patrí medzi štandardy v počítačovej grafike. NURBS funkcie sú stavebnými kameňmi Isogeometrického prístupu. Okrem geometrie ich využijeme ako bázové funkcie na hľadanie riešenia. Tomu sa však budeme venovať až v piatej kapitole. Tá bude obsahovať konkrétne body postupu z kapitoly tri a na jej konci by mal byť čitateľ schopný vyriešiť Poissonovu okrajovú úlohu. V predposlednej kapitole ukážeme jej fyzikálnu interpretáciu a priložíme aj riešenie vektorovej úlohy lineárnej pružnosti. V závere sa pokúsime prácu zhrnúť a vypichnúť najdôležitejšie body.

2 GALERKINOVA METÓDA

Diferenciálne rovnice hrajú významnú rolu v inžinierskej praxi. Dokážeme pomocou nich popísať mnohé prírodné deje a následne tento popis aplikovať na konkrétny inžiniersky problém. Od schopnosti vyriešiť diferenciálne rovnice, tak priamo závisí naša schopnosť riešiť inžinierske úlohy. Nanešťastie, neexistuje univerzálny kľúč, podľa ktorého by sme sa boli schopní popasovať s ľubovoľnou rovnicou. Preto bolo vymyslených mnoho postupov ako sa dopracovať k výsledku rôznych úloh.

Jedným zo spôsobov ako pristupovať k riešeniu je Galerkinova metóda. Tá vychádza z nahradenia diferenciálnej formulácie úlohy jej integrálnou formou. Integrálnu formu zvykneme nazývať aj slabá formulácia. Následne slabú formuláciu diskretizujeme, čím jej riešenie prevedie na sústavu algebraických rovníc. Riešenie sústavy sa dá zväčša jednoducho algoritmizovať. To je jedným z dôvodov prečo je táto metóda tak populárna a masovo využívaná. Stala sa základom mnohých metód. Patria medzi ne aj metóda konečných prvkov a nami popisovaná metódy isogeometrickej analýzy.

V tejto kapitole si Galerkinovu metódu popíšeme, a prejdeme všetky spomínané kroky. Princíp si uvedieme na jednoduchej okrajovej úlohe, Poissonovej parciálnej diferenciálnej rovnici v rovine. Na takejto úlohe, by mal byť princíp ľahko viditeľný a pre čitateľa pochopiteľný. Riešenie všeobecnejších úloh nájdete napríklad v [1].

2.1 Diferenciálna formulácia

Poissonovou parciálnou diferenciálnou rovnicou môžeme popísať ustálený stav rôznych fyzikálnych dejov [1]. Ako sme spomenuli, bude sa jednať o dvoj-dimenzionálnu úlohu a tak si zadefinujeme oblasť na ktorej ju budeme riešiť.

Definícia 2.1 (*Oblasť*). Buď $\Omega \subset \mathbb{R}^2$ ohraničená otvorená množina, potom sa Ω nazýva oblasť s hranicou Γ .

Môžeme prejsť k samotnej formulácii úlohy.

Diferenciálna formulácia. Pre danú funkciu $f : \Omega \rightarrow \mathbb{R}$, spojitú na Ω nájdite funkciu $u : \Omega \rightarrow \mathbb{R}$, takú aby spĺňala diferenciálnu rovnicu

$$-\Delta u = f \quad \text{na } \Omega \tag{2.1}$$

a okrajovú podmienku

$$u = 0 \quad \text{na } \Gamma.$$

Funkciu u nazývame klasickým riešením okrajovej úlohy ak $u \in C^2(\Omega)$, tzn. u je spojitá na oblasti Ω vrátane svojich prvých, a druhých parciálnych derivácií.

Takéto zadefinovanie úlohy sa taktiež často označuje ako silná, či klasická formulácia. Hľadanie klasického riešenia býva veľmi často komplikované, ak nie nemožné. Ak aj poznáme klasické riešenie pre rovnicu a pravú stranu na určitej oblasti, často sa stráca ak sa oblasť zmení. V praxi sa taktiež často vyskytujú nespojité funkcie f . Klasická formulácia s nimi nepočíta a preto drvivú väčšinu inžinierskych úloh nie sme schopní takto vyriešiť [2]. Musíme preto pristúpiť k preformulovaniu úlohy.

2.2 Slabá formulácia

Slabá formulácia úlohy, z predchádzajúcej kapitoly, nám umožní oslabiť požiadavky na riešenie a dovoľí aplikovať ľubovolnejšiu funkciu f . V prvom kroku prenásobíme ľubovoľnou tzv. testovacou funkciou $v \in V$ obe strany rovnice (2.1)

$$-\Delta uv = fv, \quad \forall v \in V,$$

pričom V volíme ako množinu funkcií $V = \{v : v \in C^1(\Omega), v|_{\Gamma} = 0\}$. Sú to spojité funkcie na Ω vrátane prvých parciálnych derivácií, ktoré sú na hranici Ω nulové. Poznamenajme, že takáto úprava nemá vplyv na riešenie úlohy. Vzťah ostáva ekvivalentný s rovnicou (2.1). Následne integrujme obe strany rovnice cez Ω ,

$$-\int_{\Omega} \Delta uv \, dx = \int_{\Omega} fv \, dx, \quad \forall v \in V.$$

Užitím Greenovej vety sa nám integrál na ľavej strane rozpadne na

$$\int_{\Omega} \nabla u \cdot \nabla v \, dx - \int_{\Gamma} \nabla uv \cdot n \, ds = \int_{\Omega} fv \, dx, \quad \forall v \in V.$$

Ak uvážime, že všetky $v \in V$ sú nulové na hranici, vypadne nám krivkový integrál z ľavej strany a obdržíme vzťah

$$\int_{\Omega} \nabla u \cdot \nabla v \, dx = \int_{\Omega} fv \, dx, \quad \forall v \in V. \quad (2.2)$$

Všimnime si, že vo vzťahu (2.2) je u už len raz derivovaná. Na úkor toho ale zvyšujeme nároky na testovaciu funkciu v . Tá musí mať spojité parciálne derivácie prvého rádu. To však nerobí problém, keďže všetky $v \in V$ spĺňajú túto podmienku. S výhodou sme si oslabili požiadavky na hľadané riešenie u . Funkcia u musí mať spojité parciálne derivácie prvého rádu a musí spĺňať Dirichletovu okrajovú podmienku. Môžeme preto zadefinovať množinu prípustných riešení $S = \{u : u \in C^1(\Omega), u|_{\Gamma} = 0\}$,

ako množinu funkcií, ktoré spĺňajú podmienky kladené na slabé riešenie. Dá sa dokázať, že stačí ak u a v budú mať po častiach spojité prvé parciálne derivácie. To znamená, že v vyberáme z množiny testovacích funkcií

$$V = \{v : v \in PC^1(\Omega), v|_{\Gamma} = 0\},$$

a u hľadáme v množine prípustných riešení

$$S = \{u : u \in PC^1(\Omega), v|_{\Gamma} = 0\}.$$

Môžeme prejsť k slabej formulácii úlohy.

Slabá formulácia. Nájdite také $u \in S$, ktoré spĺňa integrálny vzťah (2.2) pre všetky $v \in V$.

Poznamenajme, že za určitých podmienok je možné dokázať, že slabé riešenie sa rovná klasickému. Keďže máme slabú formuláciu, môžeme pristúpiť ku Galerkinovej metóde.

2.3 Galerkinova metóda

Doposiaľ sme sa stále pojednávali o presnom riešení u , rovnice (2.2), ktoré sme schopní nájsť v nekonečne veľkej množine funkcií S . Vieme teda kde riešenie hľadať, otázka nastáva ako ho hľadať. Galerkinova metóda prichádza s myšlienkou aproximovať presné riešenie u funkciou u^h . To konkrétne lineárnou kombináciou v tvare

$$u \approx u^h = \sum_{j=1}^N c_j \varphi_j. \quad (2.3)$$

Budeme sa teda snažiť aproximovať slabé riešenie u pomocou lineárnej kombinácie N známych básových funkcií φ_j . Koefficienty c_j kombinácie (2.3), predstavujú neznáme parametre, ktoré budeme chcieť stanoviť. Keďže nám množina prípustných riešení S splýva s množinou testovacích funkcií V , môžeme básové funkcie zvoliť z V . Dosadením aproximácie u^h do slabého riešenia (2.2) dostávame rovnosť

$$\int_{\Omega} \nabla \left(\sum_{j=1}^N c_j \varphi_j \right) \cdot \nabla v \, dx = \int_{\Omega} f v \, dx, \quad \forall v \in V,$$

ktorú vďaka linearite integrálu môžeme upraviť na

$$\sum_{j=1}^N c_j \int_{\Omega} \nabla \varphi_j \cdot \nabla v \, dx = \int_{\Omega} f v \, dx, \quad \forall v \in V. \quad (2.4)$$

Pri lepšom pohľade vidíme, že sa jedná o rovnicu s N neznámymi koeficientami c_j , ktorá musí platiť pre všetky funkcie $v \in V$.

Takto sme sa vysporiadali s jedným nekonečnom a hľadáme už len N neznámych koeficientov c_j . Máme však stále nekonečne veľa funkcií, pre ktoré vzťah (2.4) musí platiť. Ponúka sa nám teda myšlienka, vybrať si z množiny V len N funkcií. Takýto set funkcií sme už raz vybrali pre aproximáciu a tak zvolme na testovanie množinu

$$V^h = \{\varphi_1, \varphi_2, \dots, \varphi_N\}$$

obsahujúcu spomínané bázové funkcie. Dosadením do (2.4) dostávame

$$\sum_{j=1}^N c_j \int_{\Omega} \nabla \varphi_j \cdot \nabla \varphi_i \, dx = \int_{\Omega} f \varphi_i \, dx, \quad i = 1, 2, \dots, n, \quad (2.5)$$

čo predstavuje sústavu N lineárnych rovníc o N neznámych. Takúto sústavu sme za určitých okolností schopní riešiť. Jej vyriešením dostávame neznáme koeficienty c_j , a môžeme zostaviť funkciu

$$u^h = \sum_{j=1}^N c_j \varphi_j, \quad (2.6)$$

aproximujúcu nami hľadané riešenie u . Takto Galerkinova metóda pristupuje k hľadaniu riešenia. Z praktického hľadiska si uvedieme ešte maticový zápis sústav a zaviedeme názvoslovie, ktoré sa časom ustálilo.

Vzťah (2.5) vyjadruje sústavu N lineárnych rovníc a N neznámych, ktorú zapíšeme maticovo ako

$$\begin{bmatrix} \int_{\Omega} \nabla \varphi_1 \cdot \nabla \varphi_1 \, dx & \int_{\Omega} \nabla \varphi_2 \cdot \nabla \varphi_1 \, dx & \dots & \int_{\Omega} \nabla \varphi_N \cdot \nabla \varphi_1 \, dx \\ \int_{\Omega} \nabla \varphi_1 \cdot \nabla \varphi_2 \, dx & \int_{\Omega} \nabla \varphi_2 \cdot \nabla \varphi_2 \, dx & \dots & \int_{\Omega} \nabla \varphi_N \cdot \nabla \varphi_2 \, dx \\ \vdots & \vdots & \vdots & \vdots \\ \int_{\Omega} \nabla \varphi_1 \cdot \nabla \varphi_N \, dx & \int_{\Omega} \nabla \varphi_2 \cdot \nabla \varphi_N \, dx & \dots & \int_{\Omega} \nabla \varphi_N \cdot \nabla \varphi_N \, dx \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_N \end{bmatrix} = \begin{bmatrix} \int_{\Omega} f \varphi_1 \, dx \\ \int_{\Omega} f \varphi_2 \, dx \\ \vdots \\ \int_{\Omega} f \varphi_N \, dx \end{bmatrix}.$$

Maticu na ľavej strane zvykneme označovať \mathbf{K} a voláme ju globálna matica tuhosti. Je symetrická a môže nadobúdať určité vlastnosti (napr. riedkosť, pozitívna definitnosť) v závislosti na voľbe bázových funkcií. Jej prvky spočítame ako integrály

$$K_{i,j} = \int_{\Omega} \nabla \varphi_j \cdot \nabla \varphi_i \, dx. \quad (2.7)$$

Vektor na pravej strane zvykneme označovať \mathbf{F} , a voláme ho zaťažovací vektor či

vektor pravých strán. Jeho prvky spočítame ako integrály

$$F_i = \int_{\Omega} f \varphi_i dx. \quad (2.8)$$

Vektor koeficientov \mathbf{c} predstavuje neznáme parametre často označované ako stupne voľnosti. Prvky vektora neznámych dostaneme vyriešením sústavy

$$\mathbf{K}\mathbf{c} = \mathbf{F}. \quad (2.9)$$

Oboznámili sme sa s Galerkinovou metódou, ktorá nám ponúka zaujímavý pohľad na to ako sa postaviť k riešeniu okrajovej úlohy. Existuje mnoho variácií Galerkinovej metódy, napr. Bubnov-Galerkinova, Petrov-Galerkinova [3]. Zväčša ide o spôsob voľby množín, z ktorých testovacie a aproximačné funkcie vyberáme. My sme zvolili Bubnov-Galerkinovu metódu. Pri tejto metóde volíme aproximačné aj testovacie funkcie z jednej množiny, tak ako bolo naznačené v texte. V nasledujúcej kapitole sa budeme venovať postupu výpočtu. Ten ovplyvňujú vlastnosti básových funkcií. Správna voľba báze prináša výhody pri riešení sústavy, ale aj pri zostavovaní matice tuhosti a zaťažovacieho vektora.

3 POSTUP VÝPOČTU

V predchádzajúcej kapitole, sme si vysvetlili ako budeme pristupovať k riešeniu okrajovej úlohy (2.1). Povedali sme si, že zovšeobecnené riešenie aproximujeme lineárnou kombináciou N bázových funkcií. Koefficienty lineárnej kombinácie (2.6) získame vyriešením sústavy (2.9).

Takáto sústava rovníc môže mať tisíce neznámych. Je zrejmé, že jej zostavenie a následné riešenie sa stáva extrémne výpočtovo náročné. Naznačili sme, že vhodnou voľbou bázových funkcií môžeme dosiahnuť vlastnosť zvanú „riedkosť“ matice sústavy. Riedke matice sa vyznačujú tým, že obsahujú veľké množstvo nulových prvkov. Na riešenie sústav s riedkymi maticami existujú veľmi efektívne numerické metódy. Výhodnou je aj fakt, že netreba zostavovať všetky prvky matice. Za zmienku stojí aj nižšia potreba pamäte, keďže do nej stačí ukladať len nenulové prvky.

Z týchto dôvodov sa s veľkou obľubou využívajú úzko lokalizované bázové funkcie. Typickým príkladom je aj metóda konečných prvkov, ktorá patrí v dnešnej dobe medzi najrozšírenejšie numerické metódy na riešenie fyzikálnych problémov. Lokalizované bázové funkcie využíva aj metóda isogeometrickej analýzy, a preto im budeme venovať nasledujúcu kapitolu.

3.1 Lokalizované bázové funkcie

Za lokalizované funkcie budeme považovať také funkcie, ktoré sú síce definované na celej oblasti Ω , no nenulové sú len na omedzenej podoblasti. Príklady takýchto funkcií môžete vidieť na obr. 3.1. Aby sme s týmito funkciami mohli pracovať, zdefinujme si nasledujúci pojem.

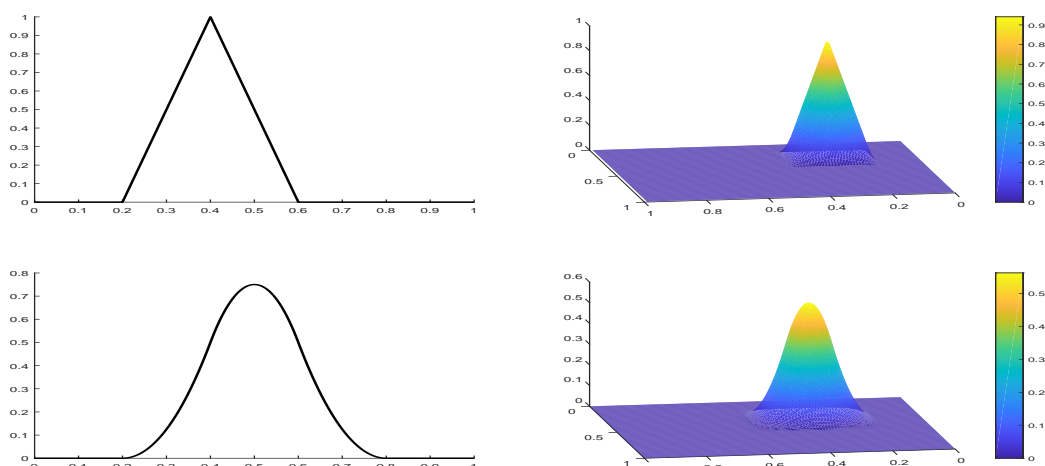
Definícia 3.1 (*nosič*). Buď Ω oblasť v \mathbb{R}^n a body $\mathbf{x} \in \Omega$ pre ktoré je funkcia $f : \mathbb{R}^n \rightarrow \mathbb{R}$ definovaná, potom množinu $\text{supp}(f) \subseteq \Omega$ na ktorej $f(\mathbf{x}) \neq 0$, nazveme nosič funkcie f .

Voľne povedané, nosič je množina bodov, v ktorých je funkcia nenulová. Dôsledok znalosti nosiča si ukážeme pri zostavovaní matice tuhosti \mathbf{K} . Jej prvky počítame ako integrály súčinu gradientov dvoch bázových funkcií cez Ω ,

$$K_{i,j} = \int_{\Omega} \nabla \varphi_j \cdot \nabla \varphi_i \, dx.$$

Predpokladajme, že gradienty $\nabla \varphi_i$ a $\nabla \varphi_j$ majú len lokálny nosič na Ω , a teda

$$\text{supp}(\nabla \varphi_i) \cap \text{supp}(\nabla \varphi_j) = \Omega_s \subset \Omega. \quad (3.1)$$



Obr. 3.1: Príklady funkcií s lokálnym nosičom. Vľavo funkcie jednej premennej. Vpravo funkcie dvoch premenných na jednotkovej oblasti.

Vieme že integrál môžeme rozdeliť na

$$\int_{\Omega} \nabla \varphi_j \cdot \nabla \varphi_i \, dx = \int_{\Omega_s} \nabla \varphi_j \cdot \nabla \varphi_i \, dx + \int_{\Omega - \Omega_s} \nabla \varphi_j \cdot \nabla \varphi_i \, dx.$$

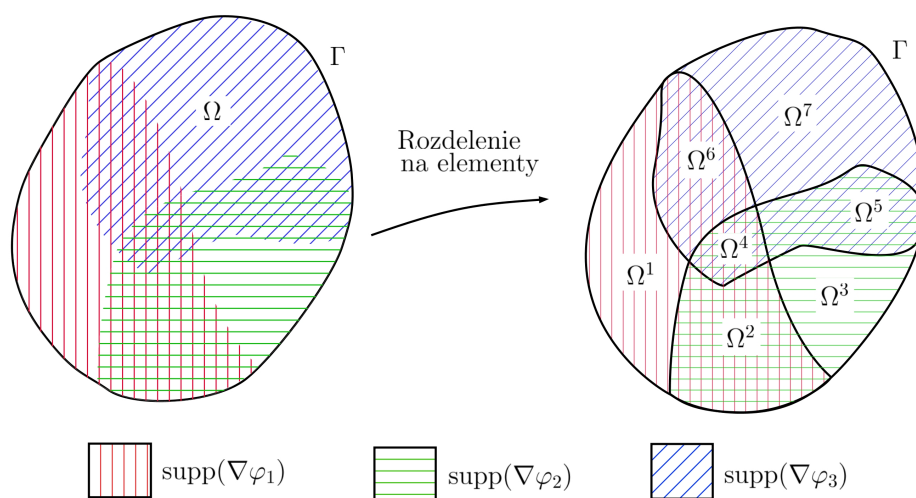
Keďže súčin $\nabla \varphi_j \cdot \nabla \varphi_i$ na $\Omega - \Omega_s$ bude nulový, potom aj integrál tohto súčinu cez $\Omega - \Omega_s$ bude nulový. Z toho môžeme usúdiť, že stačí integrovať cez Ω_s a teda

$$K_{i,j} = \int_{\Omega_s} \nabla \varphi_j \cdot \nabla \varphi_i \, dx.$$

Z týchto úvah môžeme vyvodiť určité následky. Pri numerickej integrácii nemá zmysel voliť integračné body z oblastí mimo Ω_s . To vedie k zvýšeniu presnosti a zefektívneniu numerickej integrácie. Ešte jednoduchší prípad nastane, ak je podmnožina Ω_s prázdna. Pri získaní hodnoty týchto prvkov matice \mathbf{K} nemusíme vôbec integrovať a prehlásime ich za nulové. Analogicky by sme uvažovali pri vektore pravých strán \mathbf{F} . Znalosťou nosičov jednotlivých bázových funkcií si môžeme ušetriť mnoho úsilia pri zostavovaní sústavy (2.9).

3.2 Lokálna matica tuhosti

Predstavme si súbor úzko lokalizovaných bázových funkcií so známymi nosičmi na Ω . Ako sme si pred chvíľou ukázali, pri zostavovaní prvkov matice \mathbf{K} a vektora \mathbf{F} nemusíme integrovať cez celú Ω . Často stačí integrovať cez omnoho menšiu podmnožinu Ω_s , ktorá predstavuje prienik nosičov daných funkcií. Rozdelme si preto oblasť Ω na podoblasti.



Obr. 3.2: Rozdelenie oblasti Ω na elementy podľa nosičov gradientov bázových funkcií, popísané v definícii 3.2.

Definícia 3.2 (*sieť*). Rozdelme Ω na m uzavretých podmnožín Ω^e pre ktoré platí:

- $\bigcup_{e=1}^m \Omega^e = \bar{\Omega}$.
- Dve rozdielne oblasti Ω^k a Ω^l môžu mať spoločný len vrchol alebo časť hranice.

Potom toto delenie nazvime sieť a prvky siete Ω^e elementy.

Oblasť Ω máme rozčlenenú sieťou na elementy. Ak vieme, ktoré bázové funkcie majú na jednotlivých elementoch nosiče môžeme využiť linearitu integrálov a príspevky z elementov postupne pričítať. Prvok globálnej matice tak spočítame ako

$$K_{ij} = \int_{\Omega_s} \nabla \varphi_j \cdot \nabla \varphi_i \, dx = \sum_{\{e: \bigcup \Omega^e = \Omega_s\}} \int_{\Omega^e} \nabla \varphi_j \cdot \nabla \varphi_i \, dx.$$

Najskôr si teda spočítame lokálny príspevok na elemente

$$K_{\text{lok } ij}^e = \int_{\Omega^e} \nabla \varphi_j \cdot \nabla \varphi_i \, dx, \quad (3.2)$$

ktorý následne pripočítame k prislúchajúcemu prvku K_{ij} . Aby sme takýto postup mohli uskutočniť musíme vedieť, ktoré bázové funkcie φ_i majú na elemente nosič. Preto ešte pred začatím zostavovania matice tuhosti vytvoríme pole spojitosti, ktoré každému elementu priradí bázové funkcie, ktoré na ňom majú nosič. Analogicky budeme postupovať pri zostavení vektoru pravých strán, ktorého lokálny príspevok

bude

$$F_{\text{lok } i}^e = \int_{\Omega^e} f \varphi_i \, dx. \quad (3.3)$$

Konkrétny podobu siete si popíšeme v kapitole 5. Aby si už teraz dokázal čitateľ ľahšie predstaviť takéto delenie a jeho výhody, pokúsili sme sa ho znázorniť na obr. 3.2. Predstavme si tri funkcie $\varphi_1, \varphi_2, \varphi_3$, ktorých gradienty majú známe lokálne nosiče na oblasti Ω . Ak si túto oblasť rozumne rozdelíme na elementy umožní nám to spočítať prvok matice K_{12} ako

$$K_{12} = \int_{\Omega_s} \nabla \varphi_1 \cdot \nabla \varphi_2 \, dx = \int_{\Omega^2} \nabla \varphi_1 \cdot \nabla \varphi_2 \, dx + \int_{\Omega^4} \nabla \varphi_1 \cdot \nabla \varphi_2 \, dx.$$

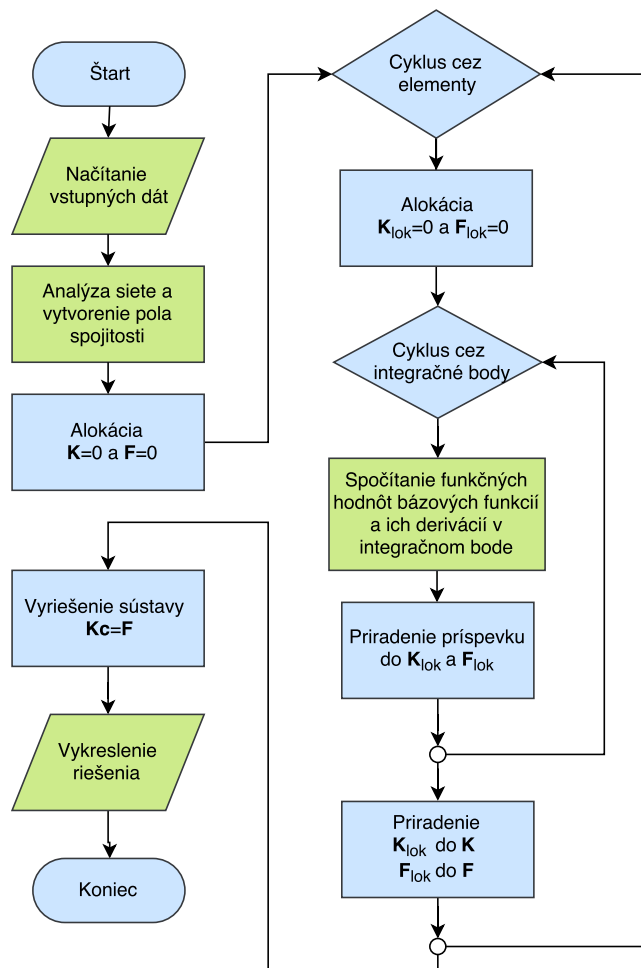
Výpočtová náročnosť zostavovania matice \mathbf{K} a vektoru \mathbf{F} takýmto prístupom, pri použití lokálnych bázových funkcií, sa značne zníži. Rozdelenie oblasti Ω na elementy je základným krokom pri metóde konečných prvkov a ako si ukážeme, využíva sa aj pri isogeometrickej analýze.

Doposiaľ sme sa pri hľadaní riešenia venovali čisto vlastnostiam bázových funkcií. Elementy sme si zatiaľ popísali ako geometricky abstraktné podmnožiny. Takýto popis sme zvolili zámerne, keďže predpokladáme, že čitateľ sa stretol s metódou konečných prvkov. Pravdepodobne si pod pojmom element predstavuje trojuholník či štvoruholník. Takouto abstraktnejšou definíciou sme chceli čitateľa priviesť k myšlienke, že v isogeometrickej metóde nebude za element voliť „pekný“ geometrický výsek z Ω , ale oblasť definovanú nosičmi bázových funkcií. Ako si neskôr ukážeme, elementy v isogeometrickej metóde môžu nadobudnúť omnoho exotickéjšie tvary.

3.3 Algoritmus výpočtu

Ukázali sme si aké výhody prináša použitie lokálnych bázových funkcií. Na záver tejto kapitoly by sme chceli uviesť postup výpočtu. Na začiatku musíme previesť analýzu siete a vytvoriť pole spojitosti. Následne na každom elemente zostaviť lokálne prvky matice tuhosti a vektoru pravých strán. Tie priradiť do globálnych \mathbf{K} a \mathbf{F} . Vyriešiť sústavu a následne zo získaných koeficientov a bázových funkcií zostaviť približné riešenie.

Tento postup sme popísali diagramom na obr. 3.3, ktorého originálnu verziu nájdete na 95. strane v [3]. Zaujímavosťou je, že analogický diagram sa dá použiť pri metóde konečných prvkov. Zelenou farbou sú zvýraznené časti algoritmu, ktoré stačí zmeniť v existujúcom konečne prvkovom algoritme aby sa stal isogeometrický.



Obr. 3.3: Diagram znázorňujúci postup výpočtu. Zelenou farbou sú zvýraznené kroky, ktoré stačí upraviť v konečne prvkovom algoritme keď chceme implementovať isogeometrický prístup.

4 OBLASŤ Ω AKO NURBS PLOCHA

Doposiaľ sme sa v tejto práci venovali tomu, akým spôsobom budeme pristupovať k riešeniu Poissonovej úlohy na oblasti Ω . Zatiaľ sme si nepovedali ako bude oblasť Ω vyzeráť, a už vôbec nie ako ju matematicky popíšeme. Ponechajme preto Galerkinovu metódu a riešenie úlohy na chvíľu bokom a zamerajme sa na matematický vyjadrenie oblasti Ω . Isogeometrická metóda užíva k popisu oblasti NURBS plochu. Ako sa neskôr ukáže, táto voľba nieje náhodná ale hrá veľkú rolu pri samotnom riešení našej úlohy (2.1).

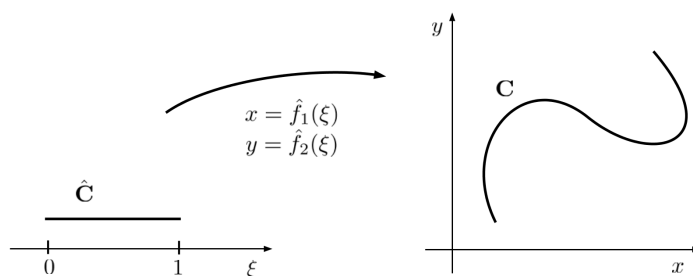
NURBS je matematický aparát, pomocou ktorého môžeme reprezentovať krivky, plochy či objekty. Dokáže efektívne popísať mnohé tvary, ktoré sa následne dajú jednoducho lokálne modifikovať. Vďaka svojej štruktúre zvládne popísať geometrii komplexných objektov pomocou malého množstva dát. Pre svoje vlastnosti je v dnešnej dobe NURBS geometria považovaná za štandard v počítačovej grafike. Svedčí o tom aj množstvo komerčných programov, ktoré pracujú na jeho báze, napr. Maya, Cinema 4D, Rhinoceros, SolidWorks.

Ešte predtým než si ukážeme ako popísať oblasť pomocou NURBS, musíme si vysvetliť ako vytvoriť B-spline krivky. Ich zobecnením sa dostaneme k NURBS krivkám, ktoré tvoria základ pre popis plôch. Chceli by sme čitateľa upozorniť, že NURBS geometriu popisujeme pre potreby isogeometrickej analýzy. Jej komplexnejšie vysvetlenie nájdete v [9].

4.1 Motivácia

Na začiatok si povedzme, k čomu slúži a čím bol motivovaný vývoj metód na popis geometrie. V druhej polovici minulého storočia nastala snaha preniesť proces navrhovania v strojárstve, stavebníctve ale aj iných oblastiach, z papiera do počítačov. Spoločnosť si uvedomovala potenciál, no bolo potrebné vymyslieť efektívny spôsob ukladania a manipulácie s geometrickými objektami.

Predstavme si, že máme krivku, ktorej tvar potrebujeme uchovať a vedieť ju kedykoľvek znova zostaviť či upraviť. Uložiť kartézské súradnice $[x_i, y_i]$ každého bodu krivky by sa nám asi nepodarilo. Núka sa myšlienka vyjadriť jej tvar pomocou matematických funkcií s voliteľnými parametrami. Parametre funkcie by sme zvolili tak, aby čo najlepšie popisovala žiadaný tvar krivky. Vhodným adeptom sa stali polynómy, ktorých tvar sa mení zmenou koeficientov. Vieme si asi predstaviť, aké náročné by bolo vyladiť tvar polynómu so stovkami koeficientov. Z toho dôvodu sa prešlo na popis po častiach polynomický. To znamená, že v určitom bode sa mení popis krivky, čím sa značne uľahčí dosiahnutie požadovaného tvaru.



Obr. 4.1: Krivka ako zobrazenie C v kartézskej súradnej sústave, intervalu \hat{C} z parametrickej sústavy.

Meniť abstraktné koeficienty a snažiť sa dosiahnuť požadovaný tvar bolo pre dizajnéra dosť nepraktické. Hľadali sa rôzne spôsoby ako s krivkou pracovať a jednoducho meniť jej tvar. Postupom času sa prišlo na spôsob, akým tieto koeficienty nahradiť kartézskymi súradnicami riadiacich bodov \mathbf{p}_i . Priamo, zmenou polohy bodu v priestore, zmeniť tvar krivky je veľmi intuitívne. Okrem toho je takýto spôsob efektívny z hľadiska množstva dát, ktoré si potrebujeme zapamätať. Stačí uložiť len súradnice riadiacich bodov \mathbf{p}_i a vieme dopočítať ktorýkoľvek bod ležiaci na krivke.

4.2 NURBS krivka

Predtým ako sa dostaneme k NURBS krivkám si musíme vysvetliť čo rozumieme pod pojmom rovinná krivka.

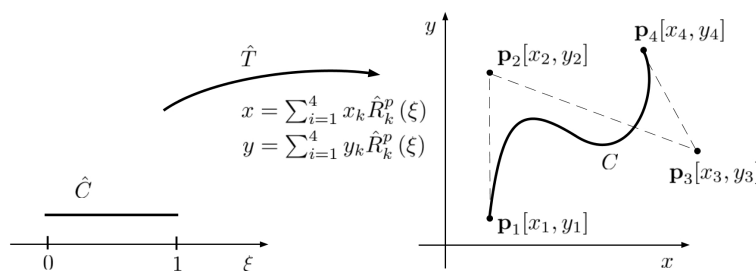
Definícia 4.1 (*rovinná krivka*). Nech $C : \hat{C} \rightarrow \mathbb{R}^2$ je zobrazenie kde $C = [\hat{f}_1, \hat{f}_2]$ pričom $\hat{f}_1, \hat{f}_2 : \mathbb{R} \rightarrow \mathbb{R}$. Nech \hat{C} je uzavretý interval a ku každému bodu $\xi \in \hat{C}$ je priradený bod \mathbf{x} o súradniciach $[x, y] = [\hat{f}_1(\xi), \hat{f}_2(\xi)] \in \mathbb{R}^2$ tak, že platí:

- Zobrazenie C je prosté na celom \hat{C} .
- Funkcie $\hat{f}_1(\xi), \hat{f}_2(\xi)$ sú na \hat{C} spojité.

Zobrazenie C tak nazvime rovinná krivka.

Touto definíciou sme sa snažili formálnejšie povedať, že krivku budeme popisovať parametricky. To znamená, že súradnice $[x_k, y_k]$ jej bodov v kartézskej sústave vieme vyjadriť ako funkcie parametra ξ . Krivku môžeme chápať aj ako obraz intervalu \hat{C} . Tento interval budeme v parametrickej sústave zobrazovať ako úsečku jednotkovej dĺžky. Tá sa následne pomocou vhodných transformačných vzťahov

$$x = \hat{f}_1(\xi), \quad y = \hat{f}_2(\xi), \quad (4.1)$$



Obr. 4.2: Krivka C zostrojená pomocou lineárnej kombinácie NURBS bázových funkcií $\hat{R}_k^p(\xi)$ a kartézskych súradníc x_k, y_k bodov riadiaceho polygónu. Na obrázku vidíme, že ku konštrukcii C boli použité štyri riadiace body, a teda aj štyri bázové funkcie.

zobrazí do požadovaného tvaru v kartézskej sústave, obr. 4.1. V závislosti na voľbe funkcií \hat{f}_1 a \hat{f}_2 hovoríme o druhu krivky. V našom prípade NURBS krivke.

U NURBS krivky volíme \hat{f}_1 a \hat{f}_2 ako lineárnu kombináciu bázových funkcií

$$x = \hat{f}_1(\xi) = \sum_{k=1}^n x_k \hat{R}_k^p(\xi), \quad y = \hat{f}_2(\xi) = \sum_{k=1}^n y_k \hat{R}_k^p(\xi), \quad (4.2)$$

kde x_k, y_k sú kartézské súradnice k -tého riadiaceho bodu \mathbf{p}_k . Tieto body tvoria tzv. riadiaci polygón znázornený čiarkovanou čiarou na obr. 4.2. $\hat{R}_k^p(\xi)$ je k -ta NURBS bázová funkcia, ktorú si čoskoro detailnejšie rozoberieme. Bázou budeme myslieť množinu všetkých bázových funkcií. Horný index p bude označovať polynomický stupeň bázových funkcií.

Zo vzťahov (4.2) rovnako vyplýva, že počet bázových funkcií \hat{R}_k^p je rovnaký ako počet riadiacich bodov \mathbf{p}_k . Krivku popíšeme ako vektorovú funkciu,

$$\mathbf{C}_{[2 \times 1]}(\xi) = \begin{bmatrix} x(\xi) \\ y(\xi) \end{bmatrix} = \begin{bmatrix} \sum_{k=1}^n x_k \hat{R}_k^p(\xi) \\ \sum_{k=1}^n y_k \hat{R}_k^p(\xi) \end{bmatrix} = \sum_{k=1}^n \mathbf{p}_k \hat{R}_k^p(\xi),$$

parametra ξ , ktorý nadobúda hodnoty $\langle 0; 1 \rangle$.

Pri počítačovej implementácii je veľmi ošzné pracovať s maticami. Preto je nespornou výhodou, že sa súradnice krivky dajú vyjadriť ako výsledok maticového súčinu

$$\mathbf{C}_{[2 \times 1]}(\xi) = \mathbf{P}_{[2 \times n]} \hat{\mathbf{R}}_{[n \times 1]}^p(\xi),$$

kde $\mathbf{P}_{[2 \times n]}$ nazývame vektor bodov riadiaceho polygónu. Po rozpísaní nadobúda tvar

$$\mathbf{P}_{[2 \times n]} = \begin{bmatrix} \mathbf{p}_1 & \mathbf{p}_2 & \dots & \mathbf{p}_n \end{bmatrix},$$

a $\hat{\mathbf{R}}_{[n \times 1]}^p$ je stĺpcový vektor NURBS bázových funkcií

$$\hat{\mathbf{R}}_{[n \times 1]}^p(\xi) = \left[\hat{R}_1^p(\xi) \quad \hat{R}_2^p(\xi) \quad \dots \quad \hat{R}_n^p(\xi) \right]^T. \quad (4.3)$$

Vysvetlili sme si, ako budeme zostavovať krivky. Môžeme prejsť k popisu samotných NURBS bázových funkcií.

4.3 NURBS bázové funkcie

Skratku NURBS by sme mohli preložiť ako *Neuniformný racionálny bázový splajn*. Prečo bázový, sme si už vysvetlili. Zvyšné dva pojmy sa priamo viažu na charakter bázy. Racionálnosť naznačuje, že bázové funkcie sú racionálne. Budú vyjadrené vo forme podielu dvoch polynómov. Neuniformita popisuje štruktúru uzlového vektora. Na to, aby sme boli schopní vysvetliť tieto charakteristiky, musíme najskôr predstaviť Bspline funkcie. Tie slúžia ku konštrukcii B-spline krivky, ktorá je predchodcom našej NURBS krivky. Transformačné vzťahy sú analogické s (4.2). Vo vzťahu sa len zamenia NURBS bázové funkcie za B-spline bázové funkcie

$$x = \hat{f}_1(\xi) = \sum_{k=1}^n x_k \hat{B}_k^p(\xi), \quad y = \hat{f}_2(\xi) = \sum_{k=1}^n y_k \hat{B}_k^p(\xi).$$

4.3.1 B-spline báza

Splinom označujeme krivku, ktorá je definovaná po častiach. Volne povedané to znamená, že je krivka rozdelená na úseky a v každom úseku sa chová nezávisle. B-spline krivky tento charakter dedia po B-spline bázových funkciách. Tie sú podľa [3] definované rekurzívnym vzťahom

$$\hat{B}_k^p(\xi) = \frac{\xi - \xi_k}{\xi_{k+p} - \xi_k} \hat{B}_k^{p-1}(\xi) + \frac{\xi_{k+p+1} - \xi}{\xi_{k+p+1} - \xi_{k+1}} \hat{B}_{k+1}^{p-1}(\xi). \quad (4.4)$$

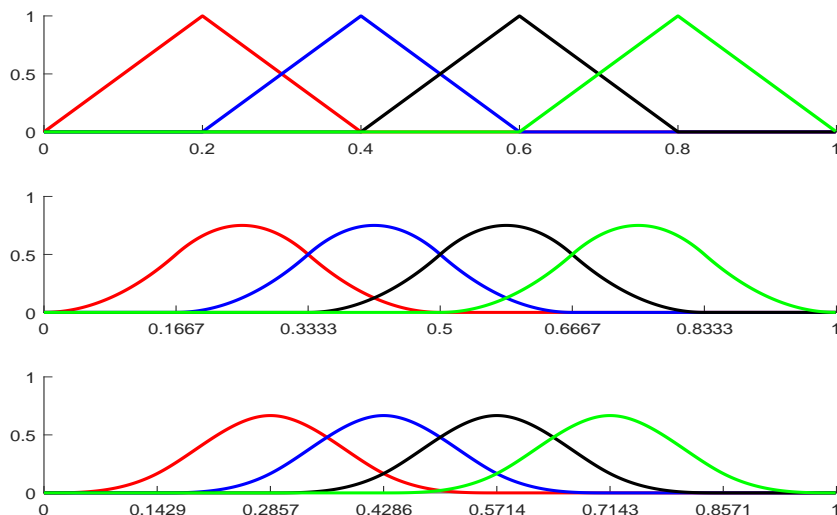
Všimnime si, že vzťah sa vždy odvolá sám na seba so zníženým indexom p . Tento postup sa ukončí v okamihu keď $p = 0$, rozhodnutím podmienky

$$\hat{B}_k^0(\xi) = \begin{cases} 1 & \text{ak } \xi_k \leq \xi < \xi_{k+1} \\ 0 & \text{inak.} \end{cases} \quad (4.5)$$

Koeficienty ξ_k sú body uzlového vektora

$$\Xi = \{\xi_1, \xi_2, \dots, \xi_{n+p+1}\},$$

ktorý je definovaný ako neklesajúca postupnosť hodnôt parametra ξ .



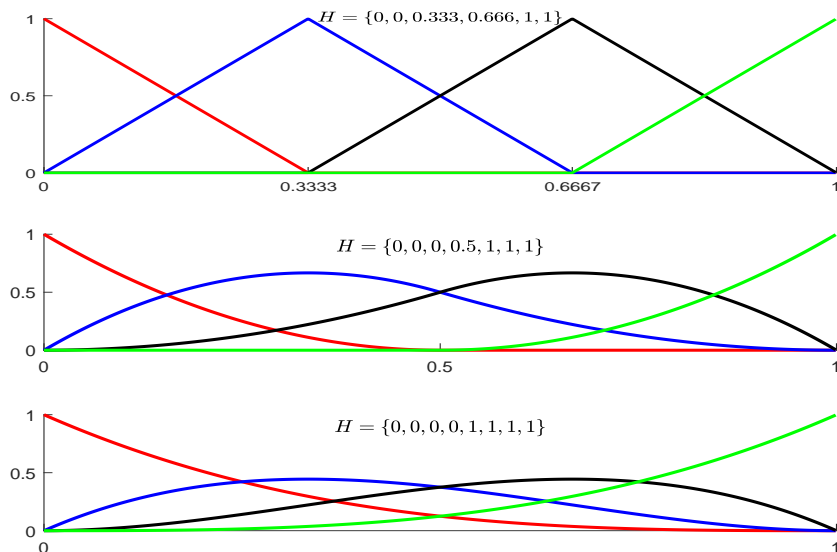
Obr. 4.3: Priebeh B-spline bázových funkcií stupňa $p = 1, 2, 3$, zostrojených pomocou uniformných uzlových vektorov. Hodnoty uzlových bodov sú vyznačené na vodorovnej ose.

Počet uzlových bodov je pevne zviazaný s počtom bázových funkcií n , a ich polynomickým stupňom p . V prípade, keď sú uzlové body rovnomerne rozmiestnené, všetky funkcie \hat{B}_k majú rovnaký priebeh, sú len posunuté. Báza zostavená pomocou takéhoto vektora Ξ sa nazýva uniformná báza.

Ak zvolíme uzly $\xi_1 = 0$ a $\xi_{n+p+1} = 1$, môžeme hodnoty Ξ chápať ako body, pomocou ktorých rozdelíme interval \hat{C} . Na obrázku 4.3 môžeme vidieť chovanie štyroch funkcií v závislosti na polynomickom stupni p . Vidíme, že funkcia \hat{B}_k^p má nosič len na intervale $(\xi_k; \xi_{k+p+1})$. Následkom toho dochádza k spomínanej zmene predpisu po intervaloch. Vzťahy (4.1) sú síce rovnaké pre celý interval \hat{C} , na každom úseku (ξ_k, ξ_{k+1}) sú však nenulové iné funkcie $\hat{B}_k^p, \hat{B}_{k+1}^p, \dots, \hat{B}_{k+p+1}^p$. To znamená, že zmenou súradníc riadiaceho bodu \mathbf{p}_k zmeníme len zobrazenie úseku, na ktorom má \hat{B}_k^p nosič. Vďaka tomu sa dá krivka pozmeniť lokálne. Môžeme tak upravovať detail v určitej časti bez toho, aby sme pozmenili celkový tvar.

Uniformná báza má jeden zásadný nedostatok. Jej použitie má za následok, že naša krivka nezačína v bode \mathbf{p}_1 a nekončí v \mathbf{p}_n . Z toho dôvodu bolo nutné upraviť bázové funkcie. To sa docielilo použitím *Neuniformného* uzlového vektora Ξ . Zistilo sa, že ak prvky $\xi_1 = \xi_2 = \dots = \xi_{1+p}$ majú rovnakú hodnotu, krivka bude začínaf v bode \mathbf{p}_1 . Analogicky, ak sú totožné posledné prvky $\xi_n = \xi_{n+1} = \dots = \xi_{n+p+1}$, krivka končí v poslednom riadiacom bode \mathbf{p}_n . Bázové funkcie zostrojené podľa takýchto uzlových vektorov sú znázornené na obr. 4.4.

Takýmto počínom dosiahneme požadované chovanie krivky. Nastane však zádrh v chode rekurzívneho vzťahu (4.4). V menovateli prvého aj druhého sčítanca môže nastať delenie nulou. Ak by nastal takýto prípad položíme zlomok rovný nule. Opa-



Obr. 4.4: Priebeh B-spline bázových funkcií stupňa $p = 1, 2, 3$, zostrojených pomocou neuniformných uzlových vektorov.

kovanie uzlových hodnôt nieje výsostným právom prvých a posledných prvkov vektora Ξ . Pokiaľ sa nám v uzlovom vektore vyskytne uzol $\xi_k = \xi_{k+1} = \dots = \xi_{k+m-1}$, budeme o ňom hovoriť ako o m násobnom uzle. Ak dosiahne uzol ξ_k násobnosť $m = p$, jemu prislúchajúca bázová funkcia \hat{B}_k^p v ňom dosiahne hodnotu 1 a krivka bude v tomto bode interpolovať príslušný riadiaci bod \mathbf{p}_k . Ak by jeho násobnosť bola väčšia ako $m + 1$, nosič funkcie \hat{B}_k^p by degradoval na nulový interval. Tým by ona a jej príslúchajúci riadiaci bod stratili vplyv na tvar C . Stali by sa nadbytočné.

Násobnosť uzlov mení tvar krivky. Okrem toho má zásadný vplyv na jej regulárnosť. Krivka je v bode \mathbf{x} regulárna n -tého rádu práve vtedy, keď existujú derivácie transformačných vzťahov (4.2) až do rádu n , pričom aspoň jedna musí byť nenulová [8]. Pod týmto pojmom si môžeme predstaviť stupeň hladkosti krivky. Ak nemajú funkcie (4.2) v danom bode ξ_k prvú deriváciu, značí to že C bude v danom bode \mathbf{x} spojitá, ale môže v ňom nastať ostrý zlom. Čím bude väčší počet nenulových derivácií n , tým bude mať C v danom bode väčšiu hladkosť. Derivovateľnosť transformačných vzťahov závisí od derivovateľnosti bázových funkcií. Obecne platí, že \hat{B}_k^p má v m -násobnom uzle ξ_k , $p - m$ spojitých derivácií. Vzťahy pre výpočet derivácií si uvedieme neskôr.

Algoritmus pre výpočet $\hat{B}_k^p(\xi)$

Tento algoritmus slúži k stanoveniu funkčnej hodnoty k -tej B-spline bázovej funkcie v bode ξ . B-spline funkcia bude polynomickeho stupňa p a zostrojená pomocou neuniformného uzlového vektora Ξ .

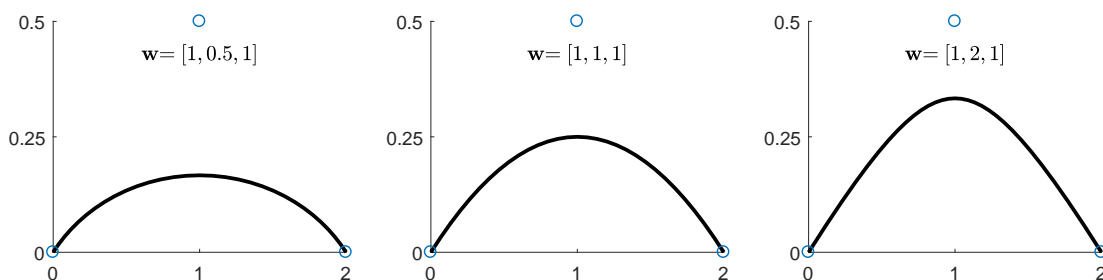
```

function B=Bspline_basis(xi,k,p,Xi,n)
%% Input
% xi - xi(1,1) one specific point in parametric space
% k - k(1,1) index of B-spline basis function
% p - p(1,1) polynomial order of B-spline basis function
% Xi - Xi(1,n+p+1) knot vector
% n - n(1,1) number of Basis functions
%% Output
% B - B(1,1) value of k-th B-spline basis function in xi
%% Cox-de Boor recursion formula
if p == 0
    if (xi >= Xi(k)) && (xi < Xi(k+1))
        B=1;
    elseif (xi==Xi(end)) && (k==n)
        B=1;
    else
        B=0;
    end
elseif Xi(k+p)== Xi(k)
    if Xi(k+p+1)== Xi(k+1)
        B=0;
    else
        B=((Xi(k+p+1)-xi)/((Xi(k+p+1)-Xi(k+1))))...
        .*Bspline_basis(xi,k+1,p-1,Xi,n);
    end
elseif Xi(k+p+1)== Xi(k+1)
    B=((xi-Xi(k))/(Xi(k+p)-Xi(k)))...
    .*Bspline_basis(xi,k,p-1,Xi,n);
else
    B=((xi-Xi(k))/(Xi(k+p)-Xi(k)))...
    .*Bspline_basis(xi,k,p-1,Xi,n)+...
    ((Xi(k+p+1)-xi)/((Xi(k+p+1)-Xi(k+1))))...
    .*Bspline_basis(xi,k+1,p-1,Xi,n);
end
end

```

4.3.2 NURBS báza

Krivka zostrojená pomocou B-spline bázy dostačuje pri popise mnohých inžinierskych tvarov. Nedokáže však presne popísať niektoré kuželosečky. Medzi ne



Obr. 4.5: Priebeh NURBS krivky v závislosti na hodnote váhového koeficientu w_2 . Kvadratická krivka je zostrojená podľa uzlového vektora $\Xi = [0, 0, 0, 1, 1, 1]$ a troch riadiacich bodov $\mathbf{p}_1 = [0, 0]^\top$, $\mathbf{p}_2 = [1, 0.5]^\top$, $\mathbf{p}_3 = [2, 0]^\top$, znázornených modrými kruhmi.

radíme kružnicu, elipsu, parabolu a hyperbolu [12]. Tieto tvary sú architektmi a projektantmi veľmi obľúbené, a preto sa s nimi často stretávame pri analýze. Ich modelovanie bolo dosiahnuté pridaním váhových koeficientov. Množinu váhových koeficientov $w_k \in \mathbb{R}^+$ budeme označovať $\mathbf{w}_{[n \times 1]} = \{w_1, w_2, \dots, w_n\}$. V literatúre zameranej na geometriu, napr.[12], sú koeficienty $\mathbf{w}_{[n \times 1]}$ považované za projektívne súradnice riadiacich bodov $\mathbf{P}_{[2 \times n]}$. To uľahčuje zapamätanie praktického významu týchto koeficientov zobrazeného na obrázku 4.5. Čím je w_k väčšie, tým sa krivka C viac približuje k danému bodu \mathbf{p}_k .

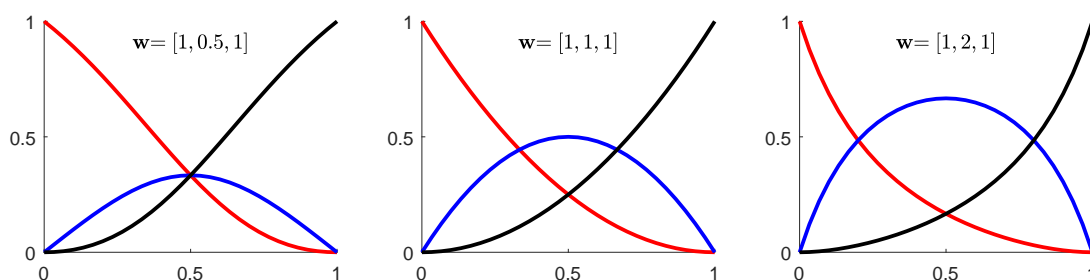
My sa skôr prikloníme k [3] a budeme w_k chápať ako váhu bázeovej funkcie $\hat{B}_k^p(\xi)$. To nám umožní NURBS bázeové funkcie vyjadriť vo forme podielu

$$\hat{R}_k^p(\xi) = \frac{\hat{B}_k^p(\xi) w_k}{\sum_{\hat{k}=1}^n \hat{B}_{\hat{k}}^p(\xi) w_{\hat{k}}}. \quad (4.6)$$

Vo vzťahu (4.6) vidíme, že koeficientom w_k pre násobíme B-spline funkciu $\hat{B}_k^p(\xi)$. Ich súčin následne podelíme sumou všetkých bázeových funkcií $\hat{B}_k^p(\xi)$ pre násobených príslušnými koeficientami w_k . Čím bude koeficient w_k väčší, tým bude konkrétna funkcia $\hat{R}_k^p(\xi)$ nadobúdať väčších funkčných hodnôt, vid. obr. 4.6. Tento efekt spôsobí spomínané približovanie sa krivky k riadiacemu bodu \mathbf{p}_k , ktorý je vidieť na obr. 4.5.

Vzťah pre výpočet vektora NURBS bázeových funkcií (4.3) si môžeme zapísať aj maticovo

$$\hat{\mathbf{R}}_{[n \times 1]}^p(\xi) = \frac{\hat{\mathbf{B}}_{[n \times 1]}^p(\xi) \odot \mathbf{w}_{[n \times 1]}}{\hat{\mathbf{B}}_{[n \times 1]}^{p^\top}(\xi) \mathbf{w}_{[n \times 1]}},$$



Obr. 4.6: Priebeh NURBS bázových funkcií závislosti na hodnote váhového koeficientu w_2 . Funkcie sú stupňa $p = 2$, zostrojené podľa uzlového vektora $\Xi = [0, 0, 0, 1, 1, 1]$. Pomocou týchto funkcií boli zostrojené krivky na obr. 4.5.

kde $\hat{\mathbf{B}}_{[n \times 1]}^p$ je stĺpcový vektor

$$\hat{\mathbf{B}}_{[n \times 1]}^p(\xi) = [\hat{B}_1^p(\xi) \quad \hat{B}_2^p(\xi) \quad \dots \quad \hat{B}_n^p(\xi)]^\top \quad (4.7)$$

všetkých n B-spline bázových funkcií v bode ξ . Prvky vektora

$$\mathbf{w}_{[n \times 1]} = [w_1 \quad w_2 \quad \dots \quad w_n]^\top$$

sú tvorené váhovými koeficientami w_k . Operátorom \odot označujeme Hadamardov súčin. Ten môžeme pre dve matice \mathbf{A} , \mathbf{B} rozmerov n, m zapísať ako $(\mathbf{A} \odot \mathbf{B})_{kl} = A_{kl}B_{kl}$. Ako sa neskôr ukáže, pri konštrukcii plochy budeme potrebovať vektor B-spline bázových funkcií. Z toho dôvodu uvedieme algoritmus pre výpočet $\hat{\mathbf{B}}_{[n \times 1]}^p(\xi)$ a nie vektora NURBS bázových funkcií.

Algoritmus na zostavenie $\hat{\mathbf{B}}_{[n \times 1]}^p(\xi)$

Tento algoritmus slúži k stanoveniu vektora hodnôt všetkých B-spline bazových funkcií v bode ξ . B-spline funkcie sú polynomickeho stupňa p a zostrojené pomocou neuniformného uzlového vektora Ξ .

```
function Bvec=Bspline_vector(xi,p,Xi)
%% Input
% xi - xi(1,1) one specific point in parametric space
% p - p(1,1) polynomial order of B-spline basis function
% Xi - Xi(1,n+p+1) knot vector
%% Output
% Bvec - Bvec(size(Xi,2)-p-1,1) B-spline functions vector in xi
%% Algorithm
Bvec=zeros(size(Xi,2)-p-1,1);
for k=1:size(Xi,2)-p-1
    Bvec(k,1)= Bspline_basis(xi,k,p,Xi,size(Xi,2)-p-1);
end
end
```


4.4 NURBS plocha

V predchádzajúcej kapitole sme si vysvetlili ako popíšeme rovinnú krivku pomocou NURBS bázových funkcií a riadiacich bodov. Popis plochy bude veľmi podobný.

Definícia 4.2 (*plocha*). Nech $S : \hat{S} \rightarrow \mathbb{R}^2$ je zobrazenie kde $S = [\hat{f}_1, \hat{f}_2]$ pričom $\hat{f}_1, \hat{f}_2 : \mathbb{R}^2 \rightarrow \mathbb{R}$. Nech $\hat{S} \subset \mathbb{R}^2$ je oblasť a ku každému bodu $[\xi, \eta] \in \hat{S}$ je priradený bod $[x, y] = [\hat{f}_1(\xi, \eta), \hat{f}_2(\xi, \eta)] \in \mathbb{R}^2$ tak, že platí:

- Zobrazenie S je prosté na celom \hat{S} .
- Funkcie $\hat{f}_1(\xi, \eta), \hat{f}_2(\xi, \eta)$ sú na \hat{S} spojité.

Zobrazenie S tak nazvime plocha.

Podobne ako pri krivke, sme sa touto definíciou snažili povedať, že plochu S budeme popisovať parametricky. To znamená, že súradnice $[x, y]$ jej bodov v kartézskej sústave vieme vyjadriť ako funkcie parametrov ξ, η . Plochu môžeme analogicky chápať ako obraz oblasti \hat{S} . Tú budeme v parametrickej sústave zobrazovať ako štvorec s stranou dĺžky jedna. Ten sa následne pomocou transformačných vzťahov

$$x = \hat{f}_1(\xi, \eta), \quad y = \hat{f}_2(\xi, \eta), \quad (4.8)$$

zobrazí na požadovanú oblasť v kartézskej sústave. Transformačné vzťahy budú opäť vo forme funkčnej rady

$$x = \sum_{k=1}^n \sum_{l=1}^m x_{kl} \hat{R}_{kl}^{p,q}(\xi, \eta), \quad y = \sum_{k=1}^n \sum_{l=1}^m y_{kl} \hat{R}_{kl}^{p,q}(\xi, \eta),$$

kde x_{kl}, y_{kl} sú kartézske súradnice bodov \mathbf{p}_{kl} . Tie budú tentokrát vytvárať takzvanú sieť riadiacich bodov. $\hat{R}_{kl}^{p,q}(\xi, \eta)$ predstavuje NURBS bázovú funkciu dvoch parametrov ξ, η . Tá vznikne opäť z B-spline bázových funkcií upravených váhovými koeficientami ako

$$\hat{R}_{kl}^{p,q}(\xi, \eta) = \frac{\hat{B}_k^p(\xi) \hat{B}_l^q(\eta) w_{kl}}{W(\xi, \eta)}. \quad (4.9)$$

kde

$$W(\xi, \eta) = \sum_{\hat{k}=1}^n \sum_{\hat{l}=1}^m \hat{B}_{\hat{k}}^p(\xi) \hat{B}_{\hat{l}}^q(\eta) w_{\hat{k}\hat{l}} \quad (4.10)$$

je váhová funkcia. Váhové koeficienty w_{kl} prislúchajú funkciám $\hat{R}_{kl}^{p,q}(\xi, \eta)$ a budú spôsobovať efekt „prťahovania“ sa plochy k bodu \mathbf{p}_{kl} .

Ako vyjadriť $\hat{B}_k^p(\xi)$ sme si vysvetlili v predchádzajúcej kapitole. Analogicky zostrojíme funkciu $\hat{B}_l^q(\eta)$ premennej η , polynomickeho stupňa q , z uzlového vektora $\mathbf{H} = \{\eta_1, \eta_2, \dots, \eta_{m+q+1}\}$. Ako vidíme, funkcií premennej η nemusí byť rovnaký počet a dokonca môžu mať aj odlišný polynomickeý stupeň ako tie premennej ξ . Každéj funkcií vzniknutej ich súčinom $\hat{B}_{kl}^{p,q}(\xi, \eta) = \hat{B}_k^p(\xi) \hat{B}_l^q(\eta)$ priradíme príslušný váhový koeficient w_{kl} . Samotné súradnice bodu plochy vypočítame ako

$$\mathbf{S}_{[2 \times 1]}(\xi, \eta) = \sum_{k=1}^n \sum_{l=1}^m \mathbf{p}_{kl} \hat{B}_{kl}^{p,q}(\xi, \eta),$$

kde $\mathbf{S}_{[2 \times 1]}(\xi, \eta)$ je vektor kartézskych súradníc bodu.

Zo spomínaných dôvodov si uvedme ešte maticový zápis. Bázové funkcie \hat{B}_l^q si zoradme analogicky s (4.7) do vektora

$$\hat{\mathbf{B}}_{[m \times 1]}^q = [\hat{B}_1^q(\eta) \quad \hat{B}_2^q(\eta) \quad \dots \quad \hat{B}_m^q(\eta)]^\top.$$

Maticovým súčinom týchto dvoch vektorov dostávame maticu

$$\hat{\mathbf{B}}_{[n \times m]}^{p,q} = \hat{\mathbf{B}}_{[n \times 1]}^p \hat{\mathbf{B}}_{[m \times 1]}^{q\top} = \begin{bmatrix} \hat{B}_{11}^{p,q}(\xi, \eta) & \hat{B}_{12}^{p,q}(\xi, \eta) & \dots & \hat{B}_{1m}^{p,q}(\xi, \eta) \\ \hat{B}_{21}^{p,q}(\xi, \eta) & \hat{B}_{22}^{p,q}(\xi, \eta) & \dots & \hat{B}_{2m}^{p,q}(\xi, \eta) \\ \vdots & \vdots & \vdots & \vdots \\ \hat{B}_{n1}^{p,q}(\xi, \eta) & \hat{B}_{n2}^{p,q}(\xi, \eta) & \dots & \hat{B}_{nm}^{p,q}(\xi, \eta) \end{bmatrix}.$$

Táto matica predstavuje súbor nm B-spline bázových funkcií definovaných na štvorcovej oblasti \hat{S} v parametrickej súradnej sústave. Aby sme z nich mohlo spraviť NURBS bázové funkcie, budeme potrebovať nm váhových koeficientov. Zapišeme ich do matice

$$\mathbf{w}_{[n \times m]} = \begin{bmatrix} w_{11} & w_{12} & \dots & w_{1m} \\ w_{21} & w_{22} & \dots & w_{2m} \\ \vdots & \vdots & \vdots & \vdots \\ w_{n1} & w_{n2} & \dots & w_{nm} \end{bmatrix}, \quad (4.11)$$

kde w_{kl} je váha príslušnej $\hat{R}_{kl}^{p,q}$ bázovej funkcie. Maticu NURBS bázových funkcií zostavíme ako

$$\hat{\mathbf{R}}_{[n \times m]}^{p,q} = \frac{\hat{\mathbf{B}}_{[n \times m]}^{p,q} \odot \mathbf{w}_{[n \times m]}}{W} = \begin{bmatrix} \hat{R}_{11}^{p,q}(\xi, \eta) & \hat{R}_{12}^{p,q}(\xi, \eta) & \dots & \hat{R}_{1m}^{p,q}(\xi, \eta) \\ \hat{R}_{21}^{p,q}(\xi, \eta) & \hat{R}_{22}^{p,q}(\xi, \eta) & \dots & \hat{R}_{2m}^{p,q}(\xi, \eta) \\ \vdots & \vdots & \vdots & \vdots \\ \hat{R}_{n1}^{p,q}(\xi, \eta) & \hat{R}_{n2}^{p,q}(\xi, \eta) & \dots & \hat{R}_{nm}^{p,q}(\xi, \eta) \end{bmatrix}, \quad (4.12)$$

kde

$$W = \hat{\mathbf{B}}_{[n \times m]}^{p,q} : \mathbf{w}_{[n \times m]}.$$

Symbol $:$ značí Frobeniov súčin ktorý môžeme pre dve matice \mathbf{A}, \mathbf{B} rozmerov n, m zapísať ako $\mathbf{A} : \mathbf{B} = \sum_{k=1}^n \sum_{l=1}^m A_{kl} B_{kl}$.

Na záver, súradnice bodu plochy spočítame ako

$$x(\xi, \eta) = \mathbf{P}_{x[n \times m]} : \hat{\mathbf{R}}_{[n \times m]}^{p,q}(\xi, \eta), \quad y(\xi, \eta) = \mathbf{P}_{y[n \times m]} : \hat{\mathbf{R}}_{[n \times m]}^{p,q}(\xi, \eta),$$

kde $\mathbf{P}_{x[n \times m]}, \mathbf{P}_{y[n \times m]}$ označujeme matice súradníc radiacich bodov. Tie obsahujú x -ové, resp. y -ové kartézské súradnice bodov \mathbf{p} radiacej siete. Tieto matice môžeme rozpísať ako

$$\mathbf{P}_{x[n \times m]} = \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1n} \\ x_{21} & x_{22} & \dots & x_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ x_{m1} & x_{m2} & \dots & x_{mn} \end{bmatrix}, \quad \mathbf{P}_{y[n \times m]} = \begin{bmatrix} y_{11} & y_{12} & \dots & y_{1n} \\ y_{21} & y_{22} & \dots & y_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ y_{m1} & y_{m2} & \dots & y_{mn} \end{bmatrix}.$$

Poznamenajme, že pri implementácii budeme mať tieto súradnice uložené v jednom objekte $\mathbf{P}_{[n \times m \times 2]}$, ktorého prvky sú $P_{kl1} = x_{kl}$ a $P_{kl2} = y_{kl}$.

Algoritmus na zostavenie $\hat{\mathbf{R}}_{[n \times m]}^{p,q}(\xi, \eta)$

Tento algoritmus slúži k stanoveniu matice hodnôt všetkých NURBS bázových funkcií v bode $[\xi, \eta]$. NURBS funkcie sú polynomického stupňa p, q a zostrojené pomocou neuniformných uzlových vektorov Ξ, \mathbf{H} a matice váhových koeficientov $\mathbf{w}_{[n \times m]}$.

```
function [Rmat]= NURBS_Basis2D(xi,eta,p,q,Xi,Eta,w)
%% Input
% xi - xi(1,1) Xi coordinate of specific point in parametric space
% eta - eta(1,1) Eta coordinate of specific point in parametric space
% p - p(1,1) order of basis fun in xi direction
% q - q(1,1) order of basis fun in eta direction
% Xi - Xi(1,n+p+1)knot vector in parameter space
% Eta - Eta(1,m+q+1)knot vector in parameter space
% w - w(n,m) matrix of weight coefficients
%% Output
% Rmat - Rmat(n,m) values of NURBS basis functions in [xi,eta]
%% Algorithm
Bvec_xi=Bspline_vector(xi,p,Xi); % Bspline vectors in xi direction
Bvec_eta=Bspline_vector(eta,q,Eta); % Bspline vectors in eta direction
W=(Bvec_xi'*w*Bvec_eta); % Weight function
Rmat=((Bvec_xi*Bvec_eta').*w)./(W);
end
```

Algoritmus na výpočet kartézskych súradníc bodov NURBS plochy

Tento algoritmus slúži k výpočtu kartézskych súradníc bodov NURBS plochy. Výsledkom je množina bodov, ktoré majú v parametrickej sústave súradnice $\xi = T_i, \eta = Q_j$. Súradnice spočítame Frobeniovym súčinom matice NURBS bazových funkcií a matice súradníc bodov riadiaceho polygónu.

```
function [S]=NURBS_Surface(T,Q,p,q,Xi,Eta,w,P)
%% Input
% T - T( ,1) Vector of xi coordinates of points in parametric space
% Q - Q( ,1) Vector of eta coordinates of points in parametric space
% p - p(1,1) order of basis fun in xi direction
% q - q(1,1) order of basis fun in eta direction
% Xi - Xi(1,n+p+1) knot vector in parameter space
% Eta - Eta(1,m+q+1) knot vector in parameter space
% W - W(n,m) matrix of weight coefficients
% P - P(n,m,3) matrix of control point coordinates
%% Output
% S - S(size(T,2),size(Q,2),3) vector of coordinates of surface point
%% Algorithm
S=zeros(size(T,2),size(Q,2),3); % size allocation
for i=1:size(T,2) % loop over all points in xi direction
    for j=1:size(Q,2) % loop over all points in eta direction
        Rmat=NURBS_Basis2D(T(i),Q(j),p,q,Xi,Eta,w); % NURBS basis matrix
        for k= 1:size(P,3) % loop over all coordinations
            S(i,j,k)=sum(sum(Rmat(:,:).*P(:,:,k))); % Frobenius inner product
        end
    end
end
end
end
```

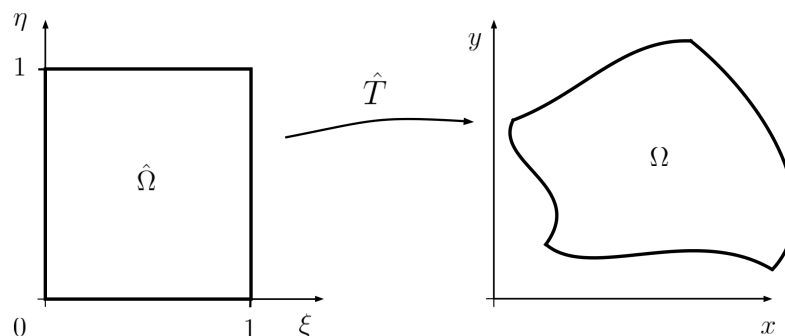
5 ISOGEOMETRICKÁ ANALÝZA-IGA

Všeobecne povedané, Isogeometrická metóda sa využíva na numerické riešenie problémov popísaných diferenciálnymi rovnicami. Jedná sa o pomerne mladý postup, ktorý autori Thomas J.R. Hughes s kolegami predstavili svetu publikáciou [6] v roku 2005. Odvtedy sa Isogeometrickej metóde začala venovať značná pozornosť. Pre jej veľký potenciál využitia v praxi, nielen v akademickej sfére, o ňu každým rokom vzrastá záujem spoločnosti.

Vývoj Isogeometrickej metódy bol motivovaný doposiaľ stále existujúcou priepasťou medzi geometrickým modelovaním pomocou CAD (Computer Aided Design) systémov, a numerickou analýzou konečne prvkovými softvérmi. Zatiaľ čo pri návrhu dizajnu vznikne CAD model, pri analýze je potrebný nový model vytvorený sieťou konečných prvkov. Existuje mnoho programov, ktoré automatizujú generovanie konečne prvkovej siete na základe CAD modelu. Pri komplikovanej geometrii však vyžadujú ľudský zásah a manuálnu úpravu siete. To vytvára časové nároky, ktoré sa pri komplexných modeloch odhadujú až na 80% času samotnej analýzy [3]. Miesto analyzovania výsledkov a optimalizácií je tak väčšina času venovaná duplikácií existujúceho CAD modelu v inom programe. Okrem toho, že vytvorenie siete je časovo náročné, geometrická reprezentácia pomocou klasických prvkov je len približná. Polygonálna aproximácia geometrie môže mať za následok značné nepresnosti vo výsledkoch, a tak znehodnocovať samotnú analýzu [3].

Hughesova metóda prepája výhody metódy konečných prvkov a možnosti presného geometrického popisu v CAD systémoch. Vďaka tomu, že Isogeometrická metóda využíva popis geometrie pomocou NURBS funkcií, nieje nutné vytvárať druhý model a samotná analýza sa môže vykonávať na NURBS modele. NURBS geometria patrí, už dnes pre svoje výhody medzi najpoužívanejšie v CAD systémoch. Designer tak nebude Isogeometrickou metódou ukrátený o možnosti pri návrhu tvarov, na ktoré bol doposiaľ zvyknutý. Analytik však ušetrí mnoho času, ktorý by strávil vytváraním konečne prvkového modelu.

Isogeometrická metóda je v mnohom príbuzná s klasickou metódou konečných prvkov. Jednou z podobností je aj izoparametrický prístup. O ňom si povieme v nasledujúcej kapitole, kde sa budeme zaoberať voľbou bazových funkcií. Postupne si v kapitole prejdeme dôležité témy, potrebné k zostaveniu sústavy a nájdeniu riešenia úlohy (2.1).



Obr. 5.1: Zostrojenie Ω transformáciou $\hat{T} : \hat{\Omega} \rightarrow \Omega$ z parametrickej do kartézskej súradnicovej sústavy.

5.1 Isoparametrický prístup

Isoparametrický prístup je dobre známy a často využívaný v metóde konečných prvkov. Jeho základná myšlienka je vo využití rovnakých bazových funkcií na parametrický popis geometrie, a aproximáciu riešenia u úlohy (2.1). Začal spoužívať z viacerých dôvodov. Jedným z nich bola neschopnosť zdefinovať aproximačné bazové funkcie φ_j na zložitejšej geometrii elementov. Druhým bolo zásadné zjednodušenie integrácie.

V klasickej metóde konečných prvkov sa vychádzalo z aproximačných funkcií, ktoré sa následne použili aj k popisu Ω . U isogeometrickej metódy bol vývoj opačný. NURBS funkcie boli najskôr použité k reprezentácii geometrie v CAD systémoch, a až následne sa zvolili ako aproximačná báza pre analýzu. Výhody z použitia isoparametrického prístupu však ostávajú rovnaké.

Spomeňme si na kapitolu 2.3. V nej sme si popísali, ako budeme riešiť úlohu (2.1) na Ω . Povedali sme si, že Ω popíšeme ako NURBS plochu. Zostrojenie rovinatej NURBS plochy môžeme chápať ako transformáciu $\hat{T} : \hat{\Omega} \rightarrow \Omega$, ktorá pomocou transformačných funkcií $f_1(\xi, \eta)$, $f_2(\xi, \eta)$ transformuje $\hat{\Omega}$ z parametrickej na Ω v kartézskej súradnej sústavy. Transformácia \hat{T} je znázornená na obr.5.1. Transformačné funkcie $f_1(\xi, \eta)$, $f_2(\xi, \eta)$ z transformačných vzťahov

$$x(\xi, \eta) = f_1(\xi, \eta) \quad y(\xi, \eta) = f_2(\xi, \eta), \quad (5.1)$$

sú v našom prípade vo forme lineárnej kombinácie NURBS funkcií $\hat{R}_{kl}^{p,q}(\xi, \eta)$ defi-

novaných na $\hat{\Omega}$ a súradníc bodov x_{kl}, y_{kl} riadiaceho polygónu,

$$f_1(\xi, \eta) = \sum_{k=1}^n \sum_{l=1}^m x_{kl} \hat{R}_{kl}^{p,q}(\xi, \eta), \quad f_2(\xi, \eta) = \sum_{k=1}^n \sum_{l=1}^m y_{kl} \hat{R}_{kl}^{p,q}(\xi, \eta). \quad (5.2)$$

Ďalej sme si povedali, že na Ω hľadáme približné riešenie u^h . Aproximáciu u^h sme hľadali vo forme funkčnej rady

$$u \approx u^h = \sum_{j=1}^N c_j \varphi_j,$$

ktorá predstavuje lineárnu kombináciu básových funkcií φ_j a neznámych koeficientov c_j , ktoré hľadáme. Izoparametrický prístup spočíva v tom, že si definujeme básovú funkcie φ_j pomocou funkcií, ktoré sme použili k transformácii $\hat{T} : \hat{\Omega} \rightarrow \Omega$. Básovú funkcie zvolíme ako

$$\varphi_j = R_j^{p,q}(x(\xi, \eta), y(\xi, \eta)) = \hat{R}_{kl}^{p,q}(\xi, \eta).$$

Vzťah medzi indexmi j a k, l sme zvolili ako

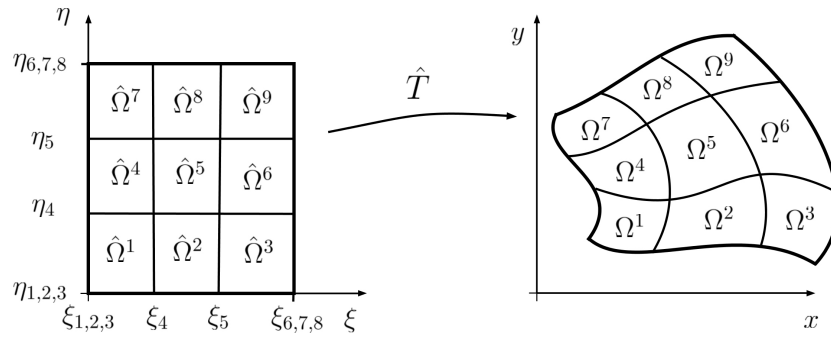
$$j = k + (l - 1)n. \quad (5.3)$$

To odpovedá číslovaníu prvkov matice NURBS funkcií (4.12) po stĺpcoch. Spôsob, akým určíme ich poradie nemôže hrať rolu v správnosti výsledku. Nezabudnime, že vyberáme len také $R_j^{p,q}(\mathbf{x}(\boldsymbol{\xi}))$, ktoré spĺňajú okrajovú podmienku na hranici Γ . Z implementačných dôvodov je však efektívnejšie očíslovať všetky NURBS funkcie a až následne si vybrať tie, ktoré vyhovujú podmienkam množiny V^h .

Približné riešenie budeme teda hľadať ako lineárnu kombináciu

$$u \approx u^h = \sum_{j=1}^N c_j R_j^{p,q}(\mathbf{x}(\boldsymbol{\xi})).$$

Keď sme pojednávali o NURBS funkciách, ukázali sme si, že majú lokálny nosič. O funkciách s lokálnym nosičom sme si rozprávali v kapitole 3. Dopracovali sme sa k názoru, že je výhodné si Ω rozdeliť na elementy Ω^e . Toto rozdelenie je vhodné zvoliť na základe nosičov aproximačných básových funkcií. Keďže už vieme, aké básovú funkcie použijeme, môžeme si konkretizovať čo budeme považovať za element.



Obr. 5.2: Členenie $\hat{\Omega}$ na elementy $\hat{\Omega}^e$ a následná transformácia na Ω na elementy Ω^e . Pri použití NURBS funkcií stupňa $p = q = 2$ by uzlové vektory takéhoto členenia boli $\Xi = \mathbf{H} = \left[0, 0, 0, \frac{1}{3}, \frac{2}{3}, 1, 1, 1\right]$.

5.2 Element v Isogeometrickej metóde

O sieti a elementoch sme si už niečo povedali v kapitole 3.2. Popis delenia na elementy si však teraz trochu konkretizujeme. Už vieme, že oblasť Ω môžeme chápať ako transformáciu $\hat{T} : \hat{\Omega} \rightarrow \Omega$ z parametrickej do kartézskej súradnej sústavy. Popísali sme si, že pri konštrukcii NURBS funkcií sú zásadné uzlové vektory Ξ a \mathbf{H} . Z uzlových vektorov vieme vyčítať, ako sa mení nosič jednotlivých NURBS funkcií na $\hat{\Omega}$. Keďže sme si aproximačné bázové funkcie definovali podľa NURBS funkcií na $\hat{\Omega}$, môžeme toho využiť.

Rozdelme si $\hat{\Omega}$ na elementy $\hat{\Omega}^e$ podľa hodnôt uzlových vektorov Ξ a \mathbf{H} . Príklad rozdelenia štvorcovej oblasti sme znázornili na obrázku 5.2. Vidíme, že $\hat{\Omega}$ je rozdelený štruktúrovanou sieťou na obdĺžniky $\hat{\Omega}^e$. Toto rozdelenie sa nám následne transformáciou \hat{T} preniesie na Ω . Elementy Ω^e môžu nadobudnúť rôzne tvary a ich hranica môže byť všeobecnejšia, oproti tomu ako sme zvyknutí v metóde konečných prvkov. V parametrickej sústave sa však každý element Ω^e zobrazí ako obdĺžnik $[\xi_k, \xi_{k+1}] \times [\eta_l, \eta_{l+1}]$.

Elementy budeme číslovať v zmysle, aký je znázornený na obrázku 5.2. Číslo elementu môžeme dopočítavať na základe indexov súradníc ľavého dolného rohu elementu $\hat{\Omega}^e$. Ten má súradnice $[\xi_k, \eta_l]$, čo sú uzlové body vektorov Ξ a \mathbf{H} . Pre číslo elementu platí

$$e = (l - q - 1)(n - p) + (k - p),$$

pokiaľ $p + 1 \leq k \leq n$ a $q + 1 \leq l \leq m$. Keďže zo samotného čísla e by sme ne-

boli schopní zistiť konkrétne hranice elementu, musíme vytvoriť pole PEV , ktoré každému e priradí indexy k a l . Pre náš prípad je znázornené v tab.5.1. Toto pole, spočítané z uzlových vektorov a polynomických indexov p, q , uložíme vo forme matice. Vieme, že funkcia $\hat{R}_{kl}^{p,q}$ má nosič len na oblasti $(\xi_k; \xi_{k+p+1}) \times (\eta_l; \eta_{l+q+1})$. Táto

e	1	2	3	4	5	6	7	8	9
k	3	4	5	3	4	5	3	4	5
l	3	3	3	4	4	4	5	5	5

Tab. 5.1: Pole PEV , ktoré každému elementu $\hat{\Omega}^e$ priraduje koeficienty k a l uzlových bodov. Tieto uzlové body predstavujú súradnice ľavého dolného rohu elementu $\hat{\Omega}^e$.

oblasť sa vždy skladá maximálne z $(p+1)(q+1)$ elementov. Na jednom elemente Ω^e má nosič $(p+1)(q+1)$ bázových funkcií. Podstatnejší je pre nás však fakt, že ľubovoľná NURBS funkcia $\hat{R}_{kl}^{p,q}$ zdieľa svoj nosič maximálne s $(2p+1)(2q+1)$ NURBS funkciami, vrátane seba samej. To znamená, že šírka pásu matice tuhosti \mathbf{K} je rovnaká ako u metódy konečných prvkov¹[3]. Od toho sa odvíja aj náročnosť riešenia sústavy (2.9), ktoré nebude komplikovanejšie ako pri použití klasickej konečne prvkovej báze.

Algoritmus pre zostavenie PEV .

Pole PEV slúži na určenie ľavého dolného vrchola $\hat{\Omega}^e$.

```
function PEV=PEV(p,q,Xi,Eta)
%% Input
% p,q - p,q(1,1) orders of basis functions
% Xi,Eta - Xi(1,n+p+1),Eta(1,m+q+1) knot vectors
%% Output
% PEV - PEV (2,number of elements) par.coord. of left down e.corner
%% Algorithm
PEV=zeros(2,(size(Xi,2)-2*p-1)*(size(Eta,2)-2*q-1));e=1;
for l=q+1:size(Eta,2)-q-1
    for k=p+1:size(Xi,2)-p-1
        PEV(1,e)=k;PEV(2,e)=1;
        e=e+1;
    end
end
end
```

¹Platí keď použijeme konečne prvkovú bázu a NURBS bázu rovnakého polynomického stupňa.

Algoritmus pre zostavenie poľa spojitosti.

Pole spojitosti ESN bude určovať ktoré bázové funkcie majú na elemente nosič.

```
function [ESN] =el_support(p,q,Xi,Eta)
%% Input
% p,q - p,q(1,1) orders of basis functions
% Xi,Eta - Xi(1,n+p+1),Eta(1,m+q+1) knot vectors
%% Output
% ESN-ESN(N_bfpel,N_el) Numbers of BFs which have support on elements
%% Algorithm
N_bfpel=(p+1)*(q+1); % Number of basis funtcions per element
N_bf_xi=(size(Xi,2)-p-1); % Number of basis funtcions in xi dir.
N_el=(size(Xi,2)-2*p-1)*(size(Eta,2)-2*q-1); % Number of elements
N_el_xi=(size(Xi,2)-2*p-1); % Number of elements
N_el_ni=(size(Eta,2)-2*q-1); % Number of elements
ESN=zeros(N_bfpel,N_el);
for i=1:N_el_ni
    for j=1:N_el_xi
        for k=1:(q+1)
            ESN((k-1)*(p+1)+1:k*(p+1),j+(i-1).*N_el_xi)=...
                (j+(i-1+k-1)*N_bf_xi+1:j+p+(i-1+k-1)*N_bf_xi);
        end
    end
end
end
end
```

5.3 Derivácie NURBS bázových funkcií

Pre zostavovanie matice tuhosti budeme potrebovať parciálne derivácie NURBS bázových funkcií. Tak ako sú NURBS odvodené z B-spline funkcií aj derivácie NURBS vychádzajú z derivácií B-spline funkcií. Pre výpočet derivácie B-spline funkcie využijeme znalosti Cox-de Boorovho rekurzívneho vzťahu (4.4). Prvú deriváciu spočítame ako

$$\hat{B}_{k,\xi}^p(\xi) = \frac{p}{\xi_{k+p} - \xi_k} \hat{B}_k^{p-1}(\xi) - \frac{p}{\xi_{k+p+1} - \xi_{k+1}} \hat{B}_{k+1}^{p-1}(\xi). \quad (5.4)$$

Z nej sa následne spočítajú parciálne derivácie NURBS jednoducho, užitím vzorca pre parciálnu deriváciu podielu² vzťahu (4.9), ako

$$\hat{R}_{kl,\xi}^{p,q}(\xi, \eta) = \frac{\left(\hat{B}_{k,\xi}^p(\xi) \hat{B}_l^q(\eta) w_{kl}\right) W(\xi, \eta) - \left(\hat{B}_k^p(\xi) \hat{B}_l^q(\eta) w_{kl}\right) W_{,\xi}(\xi, \eta)}{\left(W(\xi, \eta)\right)^2},$$

kde

$$W_{,\xi}(\xi, \eta) = \sum_{\hat{k}=1}^n \sum_{\hat{l}=1}^m \hat{B}_{\hat{k},\xi}^p(\xi) \hat{B}_{\hat{l}}^q(\eta) w_{\hat{k}\hat{l}}$$

je parciálna derivácia váhovej funkcie (4.10). Analogicky by sme spočítali aj parciálnu deriváciu podľa premennej η . Derivácie vyšších rádov sa dajú odvodiť podobne a môžeme ich nájsť napríklad v [3].

Maticový zápis je podobný ako pre nederivovaných funkciách. Vektor prvých derivácií všetkých B-spline funkcií bude

$$\hat{\mathbf{B}}_{,\xi[n \times 1]}^p(\xi) = \left[\hat{B}_{1,\xi}^p(\xi) \quad \hat{B}_{2,\xi}^p(\xi) \quad \dots \quad \hat{B}_{n,\xi}^p(\xi) \right]^\top.$$

Jeho súčinom s vektorom $\hat{\mathbf{B}}_{[m \times 1]}^q(\eta)$ dostávame maticu parciálnych derivovaných B-spline funkcií,

$$\hat{\mathbf{B}}_{,\xi[n \times m]}^{p,q}(\xi, \eta) = \hat{\mathbf{B}}_{,\xi[n \times 1]}^p(\xi) \hat{\mathbf{B}}_{[m \times 1]}^{q\top}(\eta).$$

Finálny vzťah výpočtu parciálne derivovaných NURBS bázových funkcií je

$$\hat{\mathbf{R}}_{,\xi[n \times m]}^{p,q} = \frac{\hat{\mathbf{B}}_{,\xi[n \times m]}^{p,q} \odot \mathbf{w}_{[n \times m]} W_{[1 \times 1]} - \hat{\mathbf{B}}_{[n \times m]}^{p,q} \odot \mathbf{w}_{[n \times m]} W_{,\xi[1 \times 1]}}{W_{[1 \times 1]}^2}, \quad (5.5)$$

²Ak $f(x, y) = \frac{g(x, y)}{h(x, y)}$, potom $f_{,x}(x, y) = \frac{g_{,x}(x, y)h(x, y) - g(x, y)h_{,x}(x, y)}{(h(x, y))^2}$.

kde

$$W_{,\xi[1 \times 1]} = \hat{\mathbf{B}}_{,\xi[n \times m]}^{p,q} : \mathbf{w}_{[n \times m]}.$$

Sekcia s algoritmami bude tentokrát trochu dlhšia. Obsahuje kód na výpočet k -tej derivovanej B-spline funkcie, vektora všetkých derivovaných funkcií a matice parciálne derivovaných NURBS funkcií pre plochu.

Algoritmus pre výpočet $\hat{B}_{k,\xi}^p(\xi)$

Tento algoritmus slúži k stanoveniu funkčnej hodnoty k -tej derivovanej B-spline bázeovej funkcie v bode ξ . Derivovaná B-spline funkcia je polynomickeho stupňa $p-1$ a zostrojená pomocou neuniformného uzlového vektora Ξ .

```
function dB=dBspline_basis(xi,k,p,Xi,n)
%% Input
% xi - xi(1,1) one specific point in parametric space
% k - k(1,1) index of B-spline basis function
% p - p(1,1) polynomial order of B-spline basis function
% Xi - Xi(1,n+p+1) knot vector
% n - n(1,1) number of Basis functions
%% Output
% dB - dB(1,1) value of k-th derivatived B-spline basis function in xi
%% Cox-de Boor recursion formula
if p == 0
    dB=0;
elseif Xi(k+p)== Xi(k)
    if Xi(k+p+1)== Xi(k+1)
        dB=0;
    else
        dB=-((p)/((Xi(k+p+1)-Xi(k+1)))).*Bspline_basis(xi,k+1,p-1,Xi,n);
    end
elseif Xi(k+p+1)== Xi(k+1)
    dB=((p)/(Xi(k+p)-Xi(k))).*Bspline_basis(xi,k,p-1,Xi,n);
else
    dB=((p)/(Xi(k+p)-Xi(k))).*Bspline_basis(xi,k,p-1,Xi,n)-...
        ((p)/((Xi(k+p+1)-Xi(k+1)))).*Bspline_basis(xi,k+1,p-1,Xi,n);
end
end
```

Algoritmus na zostavenie $\hat{\mathbf{B}}_{\xi[n \times 1]}^p(\xi)$

Tento algoritmus slúži k stanoveniu vektora hodnôt všetkých derivovaných B-spline báзовých funkcií v bode ξ . Derivované B-spline funkcie je polynomického stupňa $p - 1$ a zostrojené pomocou neuniformného uzlového vektora Ξ .

```
function dBvec=dBspline_vector(xi,p,Xi)
%% Input
% xi - xi(1,1) one specific point in parametric space
% p - p(1,1) polynomial order of B-spline basis function
% Xi - Xi(1,n+p+1) knot vector
%% Output
% dBvec - dBvec(1,size(Xi,2)-p-1) derived B-spline basis function in xi
%% Algorithm
dBvec=zeros(size(Xi,2)-p-1,1);
for k=1:size(Xi,2)-p-1
    dBvec(k,1)= dBspline_basis(xi,k,p,Xi,size(Xi,2)-p-1);
end
end
```

Algoritmus na zostavenie $\hat{\mathbf{R}}_{,\xi[n \times m]}^{p,q}(\xi, \eta)$, $\hat{\mathbf{R}}_{,\eta[n \times m]}^{p,q}(\xi, \eta)$

Tento algoritmus slúži k stanoveniu matíc hodnôt derivovaných NURBS bázo-
vých funkcií v bode $[\xi, \eta]$. Matica $\hat{\mathbf{R}}_{,\xi[n \times m]}^{p,q}(\xi, \eta)$ obsahuje funkcie derivované podľa ξ
a $\hat{\mathbf{R}}_{,\eta[n \times m]}^{p,q}(\xi, \eta)$ podľa η .

```
function [dRmat_xi,dRmat_eta]= dNURBS_Basis2D(xi,eta,p,q,Xi,Eta,w)
%% Input
% xi,eta - xi,eta(1,1) xi,eta coordinates of specific point in parametric space
% p,q - p,q(1,1) orders of basis functions
% Xi,Eta - Xi(1,n+p+1),Eta(1,m+q+1) knot vectors
% w - w(n,m) matrix of weight coefficients
%% Output
% dRmat_xi - dRmat_xi(n,m) derived Rmat basis by xi
% dRmat_eta- dRmat_eta(n,m) derived Rmat basis by eta
%% Algorithm
Bvec_p=Bspline_vector(xi,p,Xi); % Bsplines in xi direction
Bvec_q=Bspline_vector(eta,q,Eta); % Bsplines in eta direction
dBvec_p=dBspline_vector(xi,p,Xi); % derived Bsplines in xi direction
dBvec_q=dBspline_vector(eta,q,Eta); % derived Bsplines in eta direction
W=Bvec_p'*w*Bvec_q; % Weight function
W_xi=dBvec_p'*w*Bvec_q; % derived Weight function by xi
W_eta=Bvec_p'*w*dBvec_q; % derived Weight function by eta
dRmat_xi=((dBvec_p*Bvec_q')*.w)*W-((Bvec_p*Bvec_q')*.w)*W_xi/(W^2);
dRmat_eta=((Bvec_p*dBvec_q')*.w)*W-((Bvec_p*Bvec_q')*.w)*W_eta/(W^2);
end
```

5.4 Jakobiho matica

K zostaveniu matice tuhosti a vektora pravých strán potrebujeme integrovať cez transformovanú integračnú oblasť. Pri tom si musíme dať pozor na určité komplikácie. Jednou z nich je potreba jakobiánu, čiže determinantu Jakobiho matice transformácie. Túto problematiku sme rozobrali v prílohe A. Ešte predtým, ako prejdeme k samotnému zostaveniu matice tuhosti a vektora pravých strán, si musíme ukázať, ako vypočítať Jakobiho maticu v isogeometrickej metóde.

Všeobecne je Jakobiho matica definovaná z transformačných vzťahov (5.1), ako

$$\mathbf{J}(\xi, \eta) = \begin{bmatrix} f_{1,\xi}(\xi, \eta) & f_{1,\eta}(\xi, \eta) \\ f_{2,\xi}(\xi, \eta) & f_{2,\eta}(\xi, \eta) \end{bmatrix}, \quad (5.6)$$

ktorej členy sú parciálne derivácie transformačných funkcií (5.2). Keďže vieme, že tieto funkcie sú lineárnou kombináciou NUBRS bázových funkcií a súradníc riadiacich bodov, ich parciálne derivácie budú

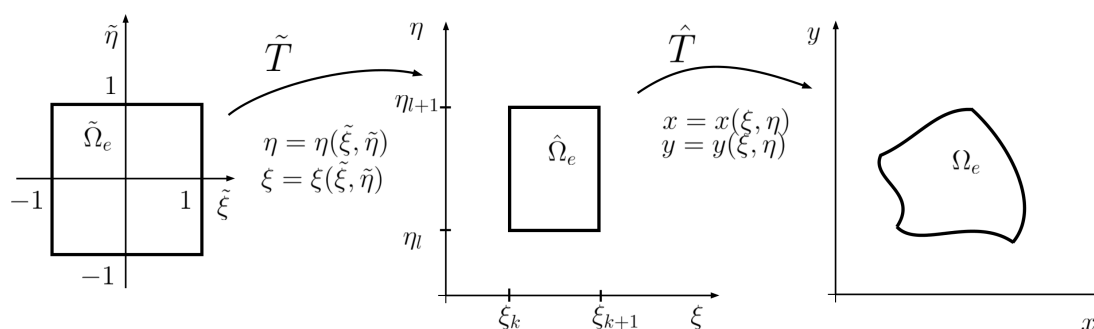
$$\begin{aligned} f_{1,\xi}(\xi, \eta) &= \sum_{k=1}^n \sum_{l=1}^m x_{kl} \hat{R}_{kl,\xi}^{p,q}(\xi, \eta), & f_{1,\eta}(\xi, \eta) &= \sum_{k=1}^n \sum_{l=1}^m y_{kl} \hat{R}_{kl,\eta}^{p,q}(\xi, \eta), \\ f_{2,\xi}(\xi, \eta) &= \sum_{k=1}^n \sum_{l=1}^m x_{kl} \hat{R}_{kl,\xi}^{p,q}(\xi, \eta), & f_{2,\eta}(\xi, \eta) &= \sum_{k=1}^n \sum_{l=1}^m y_{kl} \hat{R}_{kl,\eta}^{p,q}(\xi, \eta). \end{aligned}$$

V maticovom zápise spočítame Jakobiho maticu ako

$$\mathbf{J} = \begin{bmatrix} \mathbf{P}_{x[n \times m]} : \hat{\mathbf{R}}_{,\xi[n \times m]}^{p,q} & \mathbf{P}_{x[n \times m]} : \hat{\mathbf{R}}_{,\eta[n \times m]}^{p,q} \\ \mathbf{P}_{y[n \times m]} : \hat{\mathbf{R}}_{,\xi[n \times m]}^{p,q} & \mathbf{P}_{y[n \times m]} : \hat{\mathbf{R}}_{,\eta[n \times m]}^{p,q} \end{bmatrix}.$$

5.5 Výpočet vektora pravých strán

Na začiatku kapitoly 5.1 sme naznačili, že izoparametrický prístup so sebou nesie značné uľahčenie integrácie. Teraz by sme radi vysvetlili, v čom toto zjednodušenie spočíva. K zostaveniu matice tuhosti a vektora pravých strán potrebujeme integrovať. Dohodli sme sa, že integrály (2.7) a (2.8) spočítame postupne, najskôr na elementoch a následne príspevky z každého Ω^e sčítame. Integrovať cez oblasť elementu Ω^e je dosť náročné. Jeho hranica nadobúda nepravidelné tvary a bola by popísaná lineárnou kombináciou NURBS funkcií. Z toho dôvodu je výhodnejšie integrovať na elementoch $\hat{\Omega}^e$ v parametrickej sústave. Ako sme spomenuli, tie majú vždy obdĺžnikový tvar a tak nie je problém s medzami integrálu, keďže sú konštantné. Prvky



Obr. 5.3: Postupná transformácia elementu.

lokálneho vektora pravých strán spočítame ako

$$F_{\text{lok } i}^e = f \int_{\Omega^e} R_i^{p,q} dx dy = f \int_{\hat{\Omega}^e} \hat{R}_{kl}^{p,q} |J| d\xi d\eta. \quad (5.7)$$

Pre uľahčenie uvažujeme len konštantnú funkciu f a preto sme si dovolili ju vyňať pred integrál. J je jakobián transformácie, ktorého výpočet sme si ukázali v predchádzajúcej kapitole.

Je zrejmé, že integrál budeme aproximovať numerickým výpočtom. Vhodná metóda na integrovanie polynómov je Gaussova kvadratura. Laicky povedané, vyčíslime integrovanú funkciu v určitom počte integračných bodov ip , funkčné hodnoty pre násobíme váhovými koeficientami w_{ip} a sčítame. Presnosť integrácie závisí z veľkej miery od počtu integračných bodov a dokáže presne spočítať integrál polynómu určitého stupňa. Keďže NURBS nie sú polynómy ale racionálne funkcie, Gaussova metóda nefunguje ideálne. Navzdory tomu, sa nám pri integrácií NURBS funkcie $\hat{R}_{kl}^{p,q}(\xi, \eta)$ osvedčilo použiť Gaussovú kvadratúru o stupeň vyššiu, než je najvyšší z indexov p, q . Efektívne integračné metódy pre isogeometrickú metódu sú ešte stále predmetom výskumu [3].

Polohy integračných bodov, a ich váhy sú tabelované pre integráciu na tzv. prirodzenej štvorcovej oblasti $\tilde{\Omega} [-1, 1] \times [-1, 1]$. Preto, aby sme ich mohli použiť, musíme transformovať prirodzenú oblasť $\tilde{\Omega}$ na $\hat{\Omega}$, viz obr. 5.3. Na túto transformáciu sme zvyknutí z metódy konečných prvkov. Označme ju $\tilde{T} : \tilde{\Omega} \rightarrow \hat{\Omega}$. Jej transformačné vzťahy $\xi(\tilde{\xi}, \tilde{\eta}), \eta(\tilde{\xi}, \tilde{\eta})$ sú rovnaké, ako pri použití bilineárneho štvoruholníkového izoparametrického prvku. Nebudeme preto vzťahy uvádzať a nájdete ich v [4], časti pojednávajúcej o bilineárnom izoparametrickom prvku.

Dopad pri implementácii je taký, že nemôžeme zabudnúť na jakobián \tilde{J} transformácie \tilde{T} a prepočet integračných bodov. Prvok lokálneho vektora pravých strán

spočítame ako

$$F_{\text{lok}i}^e = f \int_{\hat{\Omega}^e} \hat{R}_{kl}^{p,q} |J| \, d\xi \, d\eta \approx f \sum_{ip=1}^{N_{ip}} w_{ip} \hat{R}_{kl}^{p,q} \left(\xi \left(\tilde{\xi}_{ip}, \tilde{\eta}_{ip} \right), \eta \left(\tilde{\xi}_{ip}, \tilde{\eta}_{ip} \right) \right) |J| |\bar{J}|. \quad (5.8)$$

Súradnice $\tilde{\xi}_{ip}, \tilde{\eta}_{ip}$ integračných dohou ip sa rovnako ako ich počet N_{ip} odvíjajú od stupňa Gassovej kvadratury. V prílohe B sme pripravili súradnice a váhy w_{ip} na použitie v podobe MatLabovskej funkcie.

Algoritmus na zostavenie vektora pravých strán \mathbf{F}

Tento algoritmus slúži na zostavenie vektora pravých strán \mathbf{F} . Vektor bude zostavený postupne, cyklom cez všetky elementy. Na elemente sa vyčíslia súradnice vrcholov elementu $\hat{\Omega}^e$. Následne prejde cyklus cez integračné body $\tilde{\xi}_{ip}, \tilde{\eta}_{ip}$. V nich sa spočíta Jakobiho matica $\bar{\mathbf{J}} \left(\tilde{\xi}_{ip}, \tilde{\eta}_{ip} \right)$ a následne sa prepočítajú na $\xi_{ip} \left(\tilde{\xi}_{ip}, \tilde{\eta}_{ip} \right), \eta_{ip} \left(\tilde{\xi}_{ip}, \tilde{\eta}_{ip} \right)$. V týchto bodoch sa spočíta Jakobiho matica $\mathbf{J} \left(\xi_{ip}, \eta_{ip} \right)$ a vyčíslia sa matica $\hat{\mathbf{R}}_{[n \times m]}^{p,q}$. Tá sa podľa vzťahu (5.3) prepočíta na vektor. Prevedie sa cyklus, v ktorom sa vyčíslia všetky nenulové prvky a pripočítajú sa do globálneho vektora. Na záver sa celý vektor prenásobí konštantou funkciou pravej strany f .

```

function [F_glob]=F_load(p,q,Xi,Eta,w,B,f,ESN)
%% Gauss quadrature points and weights
[Gauss_ip,w_ip]=Gauss(max(p,q)+1);
N_ip=size(Gauss_ip,2); % Number of integration points
%% Mesh analysis
N_el=(size(Xi,2)-2*p-1)*(size(Eta,2)-2*q-1); % Number of elements
N_bpl=(p+1)*(q+1); % Number basis funtcions per element
N_bf=(size(Xi,2)-p-1)*(size(Eta,2)-q-1); % Total number of BFs
ENA=PEV(p,q,Xi,Eta); % Parametric elements boudandaries
%% F-Load vector
F_glob=zeros(N_bf,1);
for el=1:N_el % over all elements
PEC=[Xi(ENA(1,el)) Eta(ENA(2,el)); Xi(ENA(1,el)+1) Eta(ENA(2,el));...
     Xi(ENA(1,el)+1) Eta(ENA(2,el)+1);Xi(ENA(1,el)) Eta(ENA(2,el)+1)];
for ip=1:N_ip
xi_bar=Gauss_ip(1,ip);
eta_bar=Gauss_ip(2,ip);
J_bar=[(-1+eta_bar)/4 (1-eta_bar)/4 (1+eta_bar)/4 (-1-eta_bar)/4;...
       (-1+xi_bar)/4 (-1-xi_bar)/4 (1+xi_bar)/4 (1-xi_bar)/4]*PEC;
xi=((PEC(3,1)-PEC(1,1))./2).*Gauss_ip(1,ip)+((PEC(3,1)+PEC(1,1))./2);
eta=((PEC(3,2)-PEC(1,2))./2).*Gauss_ip(2,ip)+((PEC(3,2)+PEC(1,2))./2);
[dRmat_xi,dRmat_eta]= dNURBS_Basis2D(xi,eta,p,q,Xi,Eta,w);
J=[sum(sum(dRmat_xi(:,:,:).*B(:,:,1)))...
   sum(sum(dRmat_eta(:,:,:).*B(:,:,1)))...
   sum(sum(dRmat_xi(:,:,:).*B(:,:,2)))...
   sum(sum(dRmat_eta(:,:,:).*B(:,:,2)))];
Rmat=NURBS_Basis2D(xi,eta,p,q,Xi,Eta,w);
Rv=Rmat(:)';
for i=1:N_bpl
F_lok_i=(w_ip(ip).*Rv(1,ESN(i,el)))*abs(det(J))*abs(det(J_bar));
F_glob(ESN(i,el))=F_glob(ESN(i,el))+F_lok_i;
end
end
end
F_glob=f*F_glob;
end

```

5.6 Výpočet matice tuhosti

Maticu tuhosti budeme zostavovať analogicky s vektorom pravých strán. Musíme si však dať pozor na parciálne derivácie, z ktorých sa spočítajú gradienty. Výpočet lokálnych prvkov sme si odvodili ako

$$K_{\text{lok } ij}^e = \int_{\Omega^e} \nabla R_j^{p,q} \cdot \nabla R_i^{p,q} \, dx \, dy = \int_{\Omega^e} \begin{bmatrix} R_{j,x}^{p,q} \\ R_{j,y}^{p,q} \end{bmatrix} \cdot \begin{bmatrix} R_{i,x}^{p,q} \\ R_{i,y}^{p,q} \end{bmatrix} \, dx \, dy.$$

Parciálne derivácie NURBS bázových funkcií vzhľadom k premenným x a y odvodíme pomocou Jakobihu matice transformácie ako

$$\begin{bmatrix} R_{j,x}^{p,q} \\ R_{j,y}^{p,q} \end{bmatrix} = (\mathbf{J}^\top)^{-1} \begin{bmatrix} \hat{R}_{kl,\xi}^{p,q} \\ \hat{R}_{kl,\eta}^{p,q} \end{bmatrix}.$$

Postupné odvedenie tohoto vzťahu nájdete v prílohe A. Uvážením tejto skutočnosti a transformačných postupov vysvetlených v predchádzajúcej kapitole sa dostávame k integrálu

$$K_{\text{lok } ij}^e \approx \sum_{ip=1}^{N_{ip}} w_{ip} \left(\mathbf{J}^{-\top} \begin{bmatrix} \hat{R}_{kl,\xi}^{p,q} \\ \hat{R}_{kl,\eta}^{p,q} \end{bmatrix} \right) \cdot \left(\mathbf{J}^{-\top} \begin{bmatrix} \hat{R}_{kl,\xi}^{p,q} \\ \hat{R}_{kl,\eta}^{p,q} \end{bmatrix} \right) |J| |\bar{J}|.$$

Algoritmus na zostavenie matice tuhosti \mathbf{K}

Tento algoritmus slúži na zostavenie matice tuhosti \mathbf{K} . Matica bude zostavená postupne, cyklom cez všetky elementy. Na elemente sa vyčíslia súradnice vrcholov elementu $\hat{\Omega}^e$. Následne prejde cyklus cez integračné body $\tilde{\xi}_{ip}, \tilde{\eta}_{ip}$. V nich sa spočíta Jakobihu matica $\bar{\mathbf{J}}(\tilde{\xi}_{ip}, \tilde{\eta}_{ip})$ a následne sa prepočítajú na $\xi_{ip}(\tilde{\xi}_{ip}, \tilde{\eta}_{ip}), \eta_{ip}(\tilde{\xi}_{ip}, \tilde{\eta}_{ip})$. V týchto bodoch sa spočíta Jakobihu matica $\mathbf{J}(\xi_{ip}, \eta_{ip})$ a vyčíslia sa matice $\hat{\mathbf{R}}_{,\xi}^{p,q}[n \times m]$ a $\hat{\mathbf{R}}_{,\eta}^{p,q}[n \times m]$. Tie sa podľa vzťahu (5.3) prepočítajú na vektory. Prevedie sa cyklus v ktorom sa vyčíslia všetky nenulové prvky a pripočítajú sa do globálnej matice tuhosti.

```

function [K_glob]=K_stiffness(p,q,Xi,Eta,w,B,ESN)
%% Gauss quadrature points and weights
[Gauss_ip,w_ip]=Gauss(max(p,q)+1);
N_ip=size(Gauss_ip,2); % Number of integration points
%% Mesh analysis
N_el=(size(Xi,2)-2*p-1)*(size(Eta,2)-2*q-1); % Number of elements
N_bfpel=(p+1)*(q+1); % Number of basis funtcions per element
N_bf=(size(Xi,2)-p-1)*(size(Eta,2)-q-1); % Total number of bfs
ENA=PEV(p,q,Xi,Eta); % Parametric elements boudandaries
%% K-Stiffness matrix
K_glob=zeros(N_bf,N_bf);
for el=1:N_el % over all elements
PEC=[Xi(ENA(1,el)) Eta(ENA(2,el)); Xi(ENA(1,el)+1) Eta(ENA(2,el));...
     Xi(ENA(1,el)+1) Eta(ENA(2,el)+1);Xi(ENA(1,el)) Eta(ENA(2,el)+1)];
for ip=1:N_ip % over all integration points
xi_bar=Gauss_ip(1,ip); eta_bar=Gauss_ip(2,ip);
J_bar=[(-1+eta_bar)/4 (1-eta_bar)/4 (1+eta_bar)/4 (-1-eta_bar)/4;...
       (-1+xi_bar)/4 (-1-xi_bar)/4 (1+xi_bar)/4 (1-xi_bar)/4]*PEC;
xi=((PEC(3,1)-PEC(1,1))./2).*Gauss_ip(1,ip)+((PEC(3,1)+PEC(1,1))./2);
eta=((PEC(3,2)-PEC(1,2))./2).*Gauss_ip(2,ip)+((PEC(3,2)+PEC(1,2))./2);
[dRmat_xi,dRmat_eta]= dNURBS_Basis2D(xi,eta,p,q,Xi,Eta,w);
J=[sum(sum(dRmat_xi(:,:).*B(:,:,1)))...
   sum(sum(dRmat_eta(:,:).*B(:,:,1)))...
   sum(sum(dRmat_xi(:,:).*B(:,:,2)))...
   sum(sum(dRmat_eta(:,:).*B(:,:,2)))];
J_invtrans=inv(J)';
Rv_xi = dRmat_xi(:)'; Rv_eta = dRmat_eta(:)';
for i=1:N_bfpel
for j=1:N_bfpel
delta_Rj=[Rv_xi(1,ESN(j,el));Rv_eta(1,ESN(j,el))];
delta_Ri=[Rv_xi(1,ESN(i,el));Rv_eta(1,ESN(i,el))];
K_lok=w_ip(ip)*(J_invtrans*delta_Rj)'*(J_invtrans*delta_Ri)...
      *abs(det(J))*abs(det(J_bar));
K_glob(ESN(i,el),ESN(j,el))=K_glob(ESN(i,el),ESN(j,el))+K_lok;
end
end
end
end
end
end
end

```

5.7 Vyriešenie sústavy

Dostávame sa k záveru. Maticu tuhosti \mathbf{K} a vektor pravých strán \mathbf{F} máme zostavené. Nemôžeme zabudnúť na to, že sme pri ich zostavení nebrali ohľad na okrajové podmienky. To znamená, že sme počítali s bázovými funkciami, ktoré nie sú na hranici Γ nulové. Preto musíme z \mathbf{K} a \mathbf{F} odstrániť riadky a stĺpce, v ktorých sa tieto funkcie nachádzajú. Ostáva už len vyriešiť sústavu lineárnych rovníc

$$\mathbf{K}\mathbf{c} = \mathbf{F}, \quad (5.9)$$

čím získame hľadaný vektor \mathbf{c} . Jeho prvky sú koeficienty lineárnej kombinácie bázových funkcií

$$u \approx u^h = \sum_{i=1}^N c_j R_j^{p,q}(\mathbf{x}(\boldsymbol{\xi})),$$

ktorá predstavuje aproximáciu nami hľadaného riešenia úlohy (2.2).

Stručne sme sa pokúsili popísať princíp, akým metóda funguje a ukázali sme možnosť jej implementácie. Nami podaná implementácia jednoznačne nieje najefektívnejšia. Mnoho postupov v nej sa dá uľahčiť a sme si toho vedomí. Išlo nám skôr o predstavenie názorného algoritmu, kde sú všetky postupy zreteľné. Pri zefektívnení algoritmov by mnohé kroky ostali nečitateľné a predpokladáme, že by čitateľa skôr zmietli ako mu uľahčili pochopenie problému. V nasledujúcej, predposlednej kapitole by sme radi ukázali praktické aplikácie úlohy. Najskôr uvidíme fyzikálnu interpretáciu problému popísaného Poissonovou rovnicou a následne jej riešenie. Neskôr sa vrhneme na lineárnu pružnosť, a to konkrétne problém rovinného napätia.

6 APLIKÁCIE METÓDY NA ÚLOHY MECHANIKY KONTINUA

Jedným z oborov mechaniky kontinua pevnej fázy, je teória pružnosti. Jej základnou úlohou, ktorej riešením sa budeme zaoberať, je tzv. priama úloha pružnosti. Tá sa snaží pre teleso so známou geometriou, materiálom, zaťažením a väzbami na okolie určiť posuny, pretvorenia a napätia. Určenie pretvorení a napätí je predpokladom k následnému posudzovaniu konštrukcie v praxi. Predpokladáme, že teória pružnosti je čitateľovi dobre známa zo základných kurzov. Ak nie, môže sú nájsť v publikácii [13]. Nebudeme ju preto rozoberať dopodrobna a zopakujeme len niektoré pojmy, ktoré sú nevyhnutné k uceleniu textu.

Vobecnej priestorovej úlohe hľadáme pätnásť neznámych funkcií premenných x, y, z . Tie môžeme rozdeliť do skupín na:

- tri posunutia u, v, w ,
- šesť deformácií $\varepsilon_{xx}, \varepsilon_{yy}, \varepsilon_{zz}, \varepsilon_{yz}, \varepsilon_{xz}, \varepsilon_{xy}$,
- šesť napätí $\sigma_{xx}, \sigma_{yy}, \sigma_{zz}, \sigma_{yz}, \sigma_{xz}, \sigma_{xy}$.

Tieto neznáme funkcie sú navzájom zviazané systémom všeobecných rovníc pružnosti, ktoré musia byť splnené vo vnútri telesa. Jedná sa o rovnice rovnováhy, rovnice fyzikálne alebo konštitutívne a geometrické rovnice. Aby bola úloha jednoznačná, musia byť na okraji telesa predpísané okrajové podmienky.

Rovnice rovnováhy predstavujú podmienku rovnováhy elementárneho prvku telesa, na ktorý pôsobia zložky napätia a objemové sily (napríklad gravitačné), ktorých zložky v smere x, y, z značíme X, Y, Z . Predstavujú vzájomnú väzbu medzi zložkami napätí, ktorá musí byť splnená vždy bez ohľadu na typ materiálu, veľkosť deformácií atď. Pokiaľ uvažujeme statické úlohy, a zanedbávame tak zotrvačné sily, majú tieto rovnice tvar

$$\begin{aligned} \frac{\partial \sigma_{xx}}{\partial x} + \frac{\partial \sigma_{xy}}{\partial y} + \frac{\partial \sigma_{xz}}{\partial z} + X &= 0, \\ \frac{\partial \sigma_{yx}}{\partial x} + \frac{\partial \sigma_{yy}}{\partial y} + \frac{\partial \sigma_{yz}}{\partial z} + Y &= 0, \\ \frac{\partial \sigma_{zx}}{\partial z} + \frac{\partial \sigma_{zy}}{\partial y} + \frac{\partial \sigma_{zz}}{\partial z} + Z &= 0. \end{aligned} \tag{6.1}$$

Rovnice geometrické vzájomne zväzujú zložky posunov a pretvorení. Všeobecne existuje v mechanike kontinua viacero spôsobov ako definovať tieto vzťahy. V lineárnej pružnosti, kde uvažujeme len malé deformácie sú v tvare

$$\begin{aligned} \varepsilon_{xx} &= \frac{\partial u}{\partial x}, & \varepsilon_{yy} &= \frac{\partial v}{\partial y}, & \varepsilon_{zz} &= \frac{\partial w}{\partial z}, \\ \varepsilon_{yz} &= \frac{\partial v}{\partial z} + \frac{\partial w}{\partial y}, & \varepsilon_{xz} &= \frac{\partial u}{\partial z} + \frac{\partial w}{\partial x}, & \varepsilon_{xy} &= \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}. \end{aligned} \quad (6.2)$$

Rovnice fyzikálne často označované aj ako konštitutívne vzťahy popisujú relácie medzi pretvorením a napätosťou. Pre teleso z lineárne pružného, izotropného materiálu sú vyjadrené zovšeobecným Hookovým zákonom

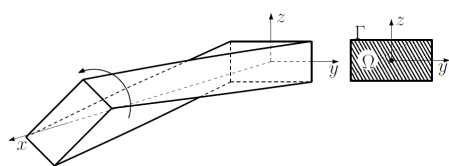
$$\begin{aligned} \varepsilon_{xx} &= \frac{1}{E} [\sigma_{xx} - \mu (\sigma_{yy} + \sigma_{zz})], & \varepsilon_{yz} &= \frac{1}{2G} \sigma_{yz}, \\ \varepsilon_{yy} &= \frac{1}{E} [\sigma_{yy} - \mu (\sigma_{xx} + \sigma_{zz})], & \varepsilon_{xz} &= \frac{1}{2G} \sigma_{xz}, \\ \varepsilon_{zz} &= \frac{1}{E} [\sigma_{zz} - \mu (\sigma_{xx} + \sigma_{yy})], & \varepsilon_{xy} &= \frac{1}{2G} \sigma_{xy}. \end{aligned} \quad (6.3)$$

V rovniciach vystupujú tri konštanty. Youngov modul pružnosti E , Poissonov súčiniteľ priečnej kontrakcie μ a modul pružnosti v šmyku $G = \frac{E}{2(1+\mu)}$.

Okrajové podmienky Aby mal uvedený systém rovníc riešenie, musíme ich doplniť okrajovými podmienkami. Tie v mechanike kontinua rozdeľujeme na geometrické a silové. Geometrickými podmienkami predpíšeme posun časti povrchu telesa a fyzikálnymi môžeme zaviesť silové zaťaženie na povrchu.

6.1 Voľné krútenie prútu nekruhového prierezu

Jednou z možných aplikácií Poissonovej úlohy je problematika krútenia prútu. S takto namáhanými konštrukčnými prvkami sa často stretávajú stavební ale aj strojní inžinieri. Tomuto typu namáhania najlepšie vzdorujú prúty kruhového prierezu. Môžu však nastať prípady, kedy ich nemožno použiť a musíme siahnuť po všeobecnejších profiloch. Všeobecne sa označujú ako prúty s nekruhovým profilom. Na rozdiel od kruhových v nich dochádza k deplanácii prierezu. Pri voľnom krútení nie je deplanácii bránené a tak nevznikajú normálové napätia v smere osi prútu.



Obr. 6.1: Skrútený prizmatický prút,[5].

Uvažujme homogénny prizmatický prút, ktorého strednica je rovnobežná s osou x . Jeho prierez Ω leží v rovine y, z . Prút je na začiatku $x = 0$ votknutý a jeho opačný koniec $x = \ell$ je pootočený o uhol $\ell\alpha$, viď. obr. 6.1. Všeobecné riešenie vychádza z dvoch základných predpokladov. Jedným, je spomínaná absencia normálových napätí $\sigma_{xx} = \sigma_{yy} = \sigma_{zz} = 0$ a druhým, že sa každý prierez pootočí okolo osi prútu ako tuhý celok, pričom sa priečny tvar prierezu nemení.

Matematické odvodenie vzťahov popisujúcich tento dej, ktoré nájdete v [5], vychádza z rovníc pre priestorovú napätosť telesa a vedie k zavedeniu tzv. Airyho funkcie napätia $\Phi(y, z)$. Pokiaľ môžeme profil prútu modelovať ako jednoducho súvislú oblasť Ω s hranicou Γ , môžeme problém voľného krútenia popísať Poissonovou okrajovou úlohou

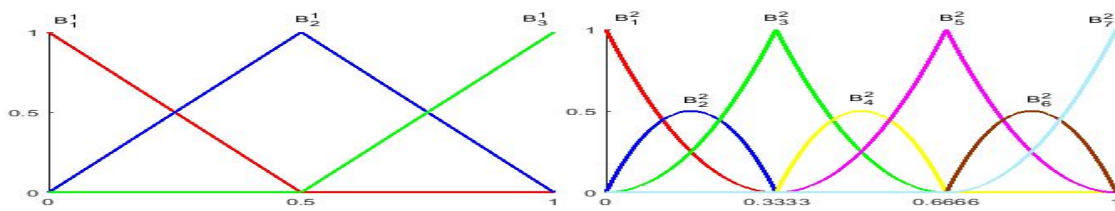
$$\begin{aligned}\Delta\Phi &= -2 \quad \text{na } \Omega, \\ \Phi &= 0 \quad \text{na } \Gamma,\end{aligned}$$

pre Airyho funkciu napätia Φ , s homogénnou Dirichletovou okrajovou podmienkou. Vyriešením tejto úlohy sme následne, podľa vzťahov

$$\sigma_{xy} = \alpha G \frac{\partial\Phi(y, z)}{\partial z}, \quad \sigma_{xz} = -\alpha G \frac{\partial\Phi(y, z)}{\partial y},$$

schopní dopočítať šmykové napätia vznikajúce v priereze. Symbolom G označujeme spomínaný modul pružnosti v šmyku a α je uhol natočenia na jednotku dĺžky prútu.

Vzhľadom na metódu, ktorej sa táto práca venuje, sme si zvolili prierez prútu ako výsek medzikruží s polomerami $r = 1$ a $R = 2$, znázornený na obrázku 6.4. Okrem zaujímavého geometrického tvaru prierezu Ω bola táto voľba motivovaná znalosťou analytického riešenia, ktoré je dopodrobna rozobraté v [5] a mohli sme tak správnosť



Obr. 6.2: NURBS bázové funkcie. Vľavo lineárne ($p=1$) zostrojené podľa uzlového vektora \mathbf{H} a vpravo kvadratické ($q=2$) podľa uzlového vektora \mathbf{H} .

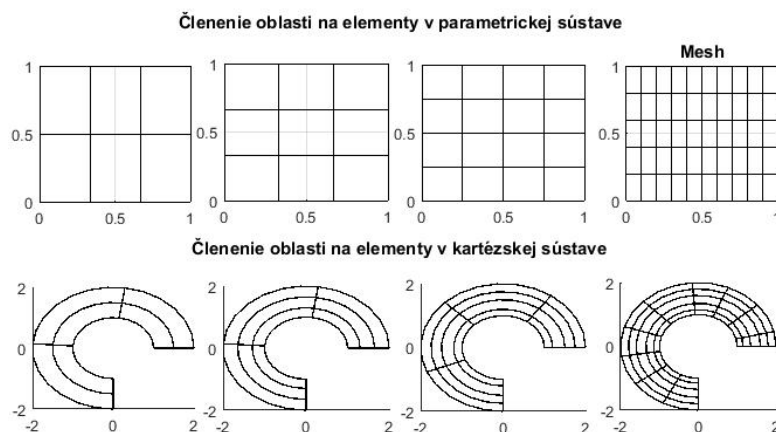
nášho riešenia overiť.

Túto oblasť sme popísali ako NURBS plochu a podľa spomínaného postupu rozčlenili na elementy. Pri najhrubšom delení, ktoré je znázornené na obrázku 6.3 vľavo, je oblasť Ω rozdelená na šesť nenulových elementov podľa uzlových vektorov

$$\mathbf{H} = \left\{ 0, 0, \frac{1}{2}, 1, 1 \right\}, \quad \mathbf{H} = \left\{ 0, 0, 0, \frac{1}{3}, \frac{1}{3}, \frac{2}{3}, \frac{2}{3}, 1, 1, 1 \right\}.$$

Z počtu opakovaní počiatočných a konečných uzlových hodnôt môžeme vyčítať, že bázové funkcie sú v smere ξ lineárne ($p = 1$) a η kvadratické ($q = 2$). B-spline bázové funkcie $\hat{B}_k^p(\xi)$ a $\hat{B}_l^q(\eta)$, ktoré tvoria základ NURBS funkciám $\hat{R}_{kl}^{p,q}(\xi, \eta)$ podľa vzťahu (4.9), sú vykreslené na obr. 6.2. Pre takéto delenie sme na obrázku 6.4 okrem oblasti Ω , znázornili čiarkovane aj riadiacu sieť a označili jednotlivé riadiace body. Ich súradnice uložené v objekte $\mathbf{P}_{[n \times m \times 2]}$ spolu maticou váhových koeficientov $\mathbf{w}_{[n \times m]}$ nájdete na konci kapitoly v sekcii s kódom.

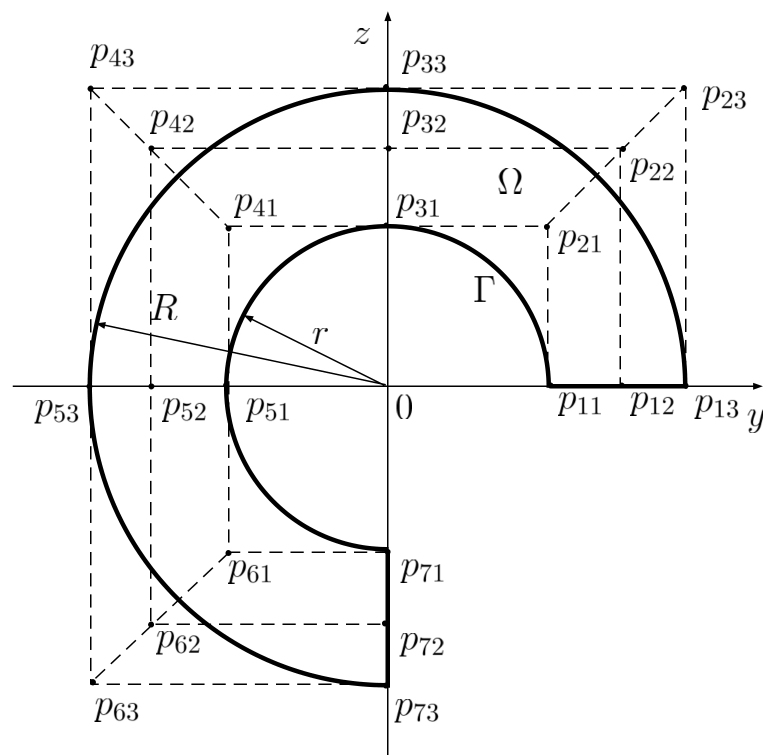
Pre zvýšenie presnosti a možnosť posúdenia konvergencie sme výpočet vykonali



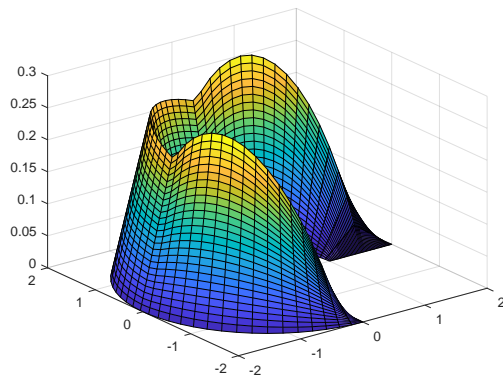
Obr. 6.3: Obrázok znázorňuje spôsob delenia oblasti $\hat{\Omega}$ (hore) a Ω (dole) v závislosti na počte elementov.

na rôznych sieťach. Stupne bázových funkcií sme zachovali a zvyšovali sme jemnosť delenia. Takáto úprava siete sa v anglickej literatúre označuje ako *h-refinement* kde *h* symbolizuje, že sa mení veľkosť prvku. V hornej časti obrázka 6.3 je zobrazené postupné členenie oblasti $\hat{\Omega}$ v parametrickej sústave. Jemnosť delenia sa zľava doprava zväčšuje, a počet elementov stúpa. V dolnej časti obrázka 6.3 sú zobrazené prislúchajúce členenia oblasti Ω v kartézskej sústave. Môžeme si všimnúť, že geometria profilu Ω sa vôbec nemení a aj pri použití malého počtu prvkov (obr. 6.3 vľavo dole) sme schopní ju vyjadriť presne.

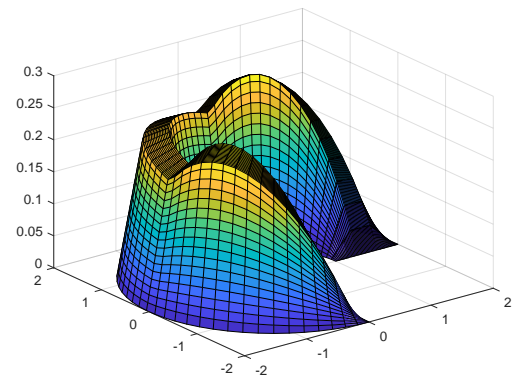
Pre jednotlivé delenia sme zostavili potrebné matice isogeometrickou metódou a približne našli Airyho funkciu napätia Φ ako riešenie Poissonovej úlohy. Je zrejmé, že čím je sieť jemnejšia, tým je počet bázových funkcií väčší a riešenie presnejšie. Konkrétne výsledky výpočtu, pre jednotlivé delenia, sme zobrazili na obrázku 6.5.



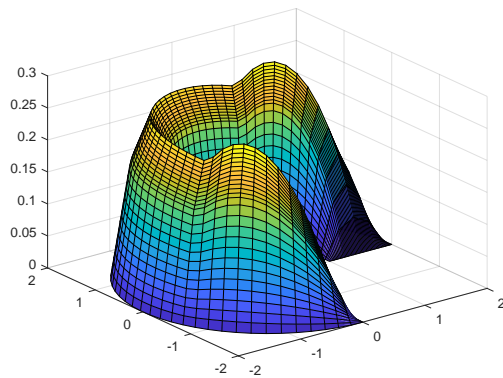
Obr. 6.4: Profil Ω v tvare výseku medzikruží znázornený hrubou plnou čiarou spolu s riadiacou sieťou vyznačenou tenkou čiarkovanou čiarou.



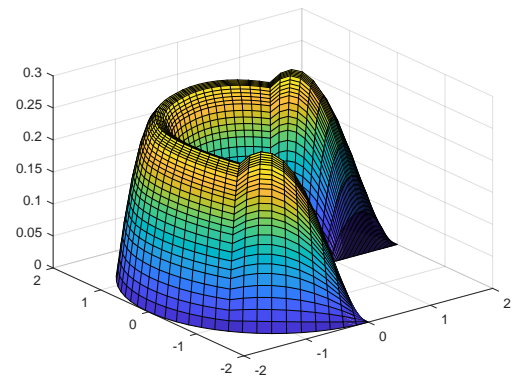
(a) 2x3 elementov



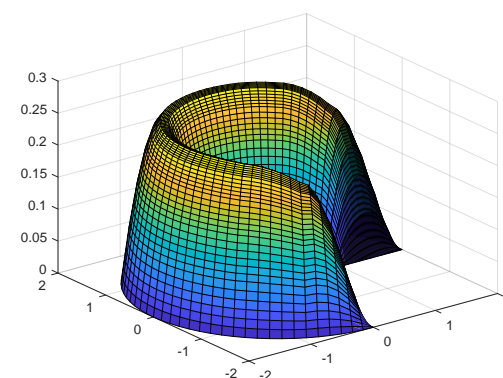
(b) 3x3 elementov



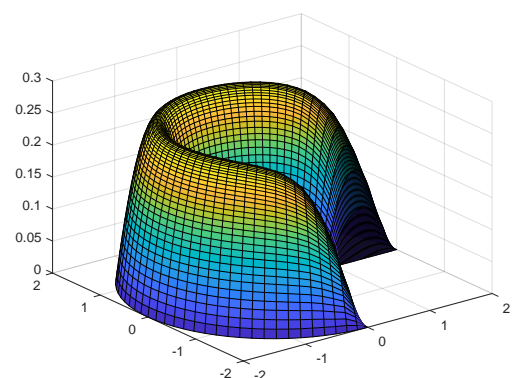
(c) 4x4 elementov



(d) 5x10 elementov



(e) 10x10 elementov



(f) 20x20 elementov

Obr. 6.5: Približné výsledky Poissonovej okrajovej úlohy získané pomocou isogeometrickej metódy v závislosti na počte elementov.

Algoritmus pre zavedenie Dirichletovej okrajovej podmienky

Tento algoritmus roztriedi bázové funkcie na tie podľa toho či splňujú Dirichletovu okrajovú podmienku $u = 0$. Vyhovujúce zaradí do vektora IB a nevyhovujúce do BOB.

```
function [BOB,IB] =Direchlet(p,q,Xi,Eta)
%% Input
% p,q - p,q(1,1) orders of basis functions
% Xi,Eta - Xi(1,n+p+1),Eta(1,m+q+1) knot vectors
%% Output
% BOB- Numbers of BFs with support on boundary
% IB- Numbers of BFs without support on boundary
%% Algorithm
N_bf=(size(Xi,2)-p-1)*(size(Eta,2)-q-1); % Number of basis funtcions
N_bf_xi=(size(Xi,2)-p-1); % Number of basis funtcions in xi dir.
BOB=[(1:1:N_bf_xi) (N_bf_xi+1:N_bf_xi:N_bf)...
(2*N_bf_xi:N_bf_xi:N_bf) (N_bf : -1 :N_bf-N_bf_xi) ];
BOB=sort(BOB);
BOB=unique(BOB);
IB=setdiff(1:N_bf,BOB);
end
```

Algoritmus pre riešenie Poissonovej okrajovej úlohy.

Tento algoritmus obsahuje vstupné údaje, postup riešenia a následné spracovanie výsledkov.

```

%% INPUT
P(:,:,1)=[ 1  1.5  2; 1  1.5  2; 0 0 0;-1 -1.5 -2;...
          -1 -1.5 -2;-1 -1.5-2; 0 0 0]';
P(:,:,2)=[ 0  0  0;  1  1.5  2;  1  1.5  2;  1  1.5  2;...
          0  0  0; -1 -1.5 -2; -1 -1.5 -2]';
w(:,:,)= [  1  1  1; sqrt(2)/2 sqrt(2)/2 sqrt(2)/2;...
          1  1  1; sqrt(2)/2 sqrt(2)/2 sqrt(2)/2;...
          1  1  1; sqrt(2)/2 sqrt(2)/2 sqrt(2)/2;1 1 1]';
Xi=[0 0 1/2 1 1 ];
Eta=[0 0 0 1/3 1/3 2/3 2/3 1 1 1];
p=1;q=2;
f=-2; % load
%% ASSEMBLY K,F
K_stiffness=K_stiffness(p,q,Xi,Eta,w,P,el_support(p,q,Xi,Eta));
F_load=F_load(p,q,Xi,Eta,w,P,f,el_support(p,q,Xi,Eta));
%% BOUNDARY CONDITIONS
[BOB,IB] =Direchlet(p,q,Xi,Eta);
K_stiffness(:,BOB)=0;K_stiffness(BOB,:)=0;
F_load(BOB)=0;
%% SOLVING SYSTEM OF LIN. EQUATIONS
C=K_stiffness(IB,IB)\F_load(IB);
%% POST PROCES
C_vek=zeros(size(F_load,1),1);
C_vek(IB) = C;
C_mat(:,:,)=reshape(C_vek(:,1),size(P,1),size(P,2));
P(:,:,3)=C_mat(:,:,);
T=(0:1/(50):1);
Q=(0:1/(50):1);
S=NURBS_Surface(T,Q,p,q,Xi,Eta,w,P);
%% PLOT SOLUTION
figure('Name','Solution'); hold on
surf(S(:,:,1),S(:,:,2),S(:,:,3));
title('Solution');
mesh=surf(P(:,:,1),P(:,:,2),zeros(size(P(:,:,1))));
set(mesh,'facecolor','none')

```

6.2 Rovinná napätosť

Pre konštrukcie, ktorých rozmer v smere osi z je omnoho menší ako zvyšné dva x, y , pristupujeme často k idealizácii na dvojrozmernú úlohu. Typickým stavbárskym prípadom takejto konštrukcie je stena. Tú môžeme reprezentovať len jej strednicou Ω . Za stenu považujeme plošnú konštrukciu, ktorá je zatažená len v rovine svojej strednice.

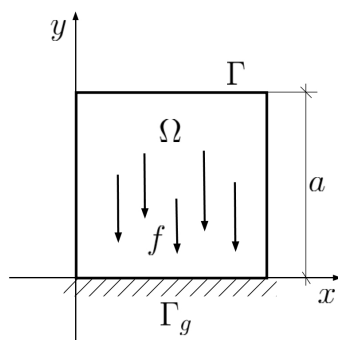
Keďže deformácii steny kolmo na strednicu nie je nijak bránené $\varepsilon_{zz} \neq 0$, nemôžu v tomto smere vznikáť napätia $\sigma_{zz} = \sigma_{xz} = \sigma_{yz} = 0$. Od šmykových napätí závisí skosenie a teda $\varepsilon_{xz} = 0, \varepsilon_{yz} = 0$. Po postupnom dosadení týchto rovností do vzťahov (6.3), (6.2) a (6.1) nám z rovníc rovnováhy podľa [7] vyplynie sústava

$$\begin{aligned} G \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right) + G \frac{1 + \mu}{1 - \mu} \left(\frac{\partial^2 u}{\partial xy} + \frac{\partial^2 v}{\partial xx} \right) + X &= 0, \\ G \left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} \right) + G \frac{1 + \mu}{1 - \mu} \left(\frac{\partial^2 u}{\partial yx} + \frac{\partial^2 v}{\partial yy} \right) + Y &= 0, \end{aligned}$$

dvoch parciálnych diferenciálnych rovníc. Tentokrát už teda nepôjde o skalárny problém, ako tomu bolo v predchádzajúcej úlohe. Budeme hľadať dve funkcie $u(x, y)$ a $v(x, y)$, ktoré budú popisovať posun každého bodu strednice $x, y \in \Omega$ v smere os x a y . Tieto rovnice doplníme geometrickými okrajovými podmienkami

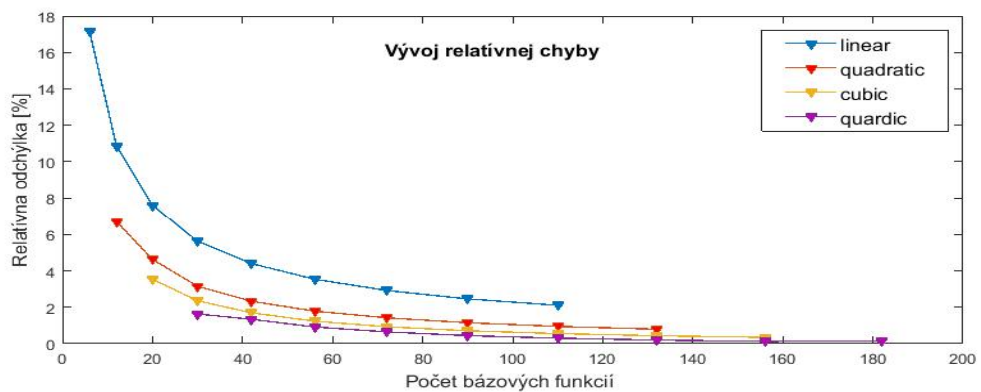
$$v = 0 \quad \text{na} \quad \Gamma_g, \quad u = 0 \quad \text{na} \quad \Gamma_g,$$

ktorými stenu stabilizujeme v priestore. Pre náš prípad sa jedná o zabránenie posunu v oboch smeroch na časti hranice $\Gamma_g \subseteq \Gamma$.



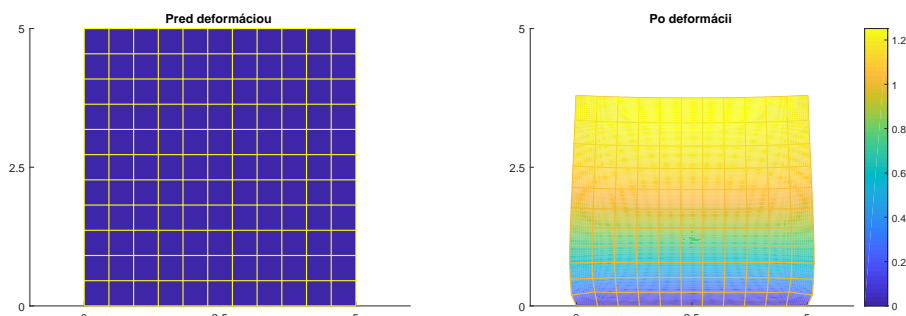
Obr. 6.6: Štvorcová stena uložená na okraji Γ_g zatažená objemovými silami f .

V takomto prípade na aproximáciu oboch funkcií u, v použijem tie isté bazové funkcie. Ak by sa okrajové podmienky pre hľadané posuny nezhodovali, museli by sa odlišovať aj aproximačné funkcie. Uvažujeme štvorcovú strednicu Ω s dĺžkou strany $a = 5\text{m}$ znázornenú na obr. 6.2. Materiál je homogénny izotropný s modulom pružnosti $E = 1000$ a poissonovým súčiniteľom $\mu = 0.2$. Zatažená je výhradne objemovými silami $f = [0, -100]$. Tie zvyčajne reprezentujú sily gravitačné no pre testovacie účely ich môžeme zvoliť ľubovoľne. Deformácia steny, spolu s tromi zložkami napätí sú vyobrazené na obr. 6.8. Popísať štvorcovú Ω pomocou NURBS plochy nieje nijak náročné. Na tomto príklade by

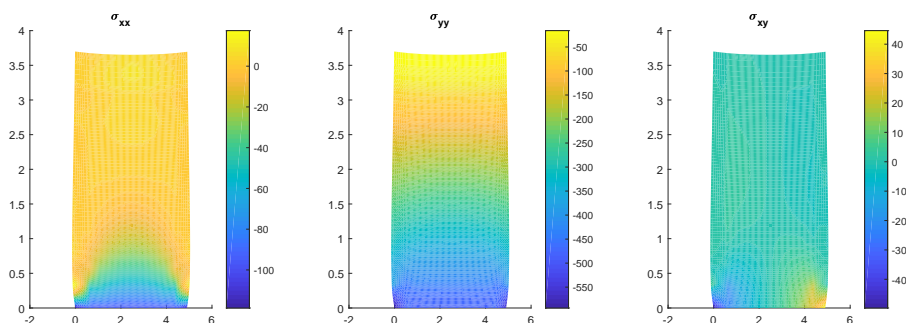


Obr. 6.7: Závislosť relatívnej odchýlky približného riešenia isogeometrickou metódou od analytického riešenia v percentách, na počte a polynomicom stupni bazových funkcií.

sme však chceli demonštrovať konvergenciu metódy. Pre zvýšenie presnosti aproximácie sme postupne zjemňovali delenie domény na elementy Ω^e . Presnosť riešenia sme vyjadrili relatívnou odchýlkou v percentách a vzťahli k počtu bazových funkcií. Tento postup sme opakovali pre rôzne polynomicke stupne NURBS báze aby sme ukázali, ako metódy konverguje pre konkrétne stupne $p = q$.



(a) Strednica steny pred a po deformácií.



(b) Jednotlivé zložky napätia.

Obr. 6.8: Riešenie úlohy rovinného napätia pomocou isogeometrickej analýzy.

7 ZÁVER

Primárnym cieľom tejto diplomovej práce bolo oboznámenie sa s Isogeometrickou metódou a možnosťami jej aplikácie na úlohy mechaniky kontinua. Úmyslom bolo čitateľovi jednoducho vysvetliť v čom spočíva isogeometrický prístup a pripraviť návod na jej aplikáciu.

Na začiatku sme si zaviedli Poissonovu parciálnu diferenciálnu rovnicu na rovinatej oblasti Ω , ktorou môžeme popisovať rôzne fyzikálne deje. Diferenciálnu (silnú) formuláciu sme následne previedli na integrálnu (slabú) a hľadali jej približné riešenie Galerkinovou metódou. Keďže sme chceli aplikovať Isogeometrický prístup, ktorý spočíva v použití NURBS bázových funkcií na aproximáciu riešenia a súčasne popis geometrie oblasti Ω , venovali sme jednu kapitolu popisu geometrie. NURBS geometriu sme sa snažili rozpísať čo najpodrobnejšie, keďže aj pre nás predstavovala niečo úplne nové a nepredpokladáme že by sa s ňou čitatelia niekedy predtým stretli. V nasledujúcej kapitole sme opäť vrátili k hľadaniu približného riešenia. Popísali sme aproximačné funkcie a postup pri zostavovaní matíc tuhosti a vektora pravých strán. V poslednej kapitole o praktických aplikáciách sme Poissonovej úlohe vdýchli fyzikálnu podstatu a dokončili sme jej riešenie. K záveru sme priložili aj riešenie vektorového problému, konkrétne úlohy rovinatej napätosti.

V závere niektorých kapitol sme k textu prikladali implementačný kód v logickej náväznosti na aktuálne prebraný problém. Kód je vždy vo forme MatLabovskej funkcie a mal by čitateľa sprevádzať a napovedať mu pri jeho vlastnej snahe o implementáciu.

Isogeometrickú metódu sa nám podarilo pochopiť a úspešne implementovať na skalárny aj vektorový problém mechaniky kontinua. Tým považujeme ciele práce za splnené. Či sa nám podarilo metódu zrozumiteľne popísať a vytvoriť tak text, ktorý by uľahčil začiatky pri jej štúdiu, už môže posúdiť len samotný čitateľ.

LITERATURA

- [1] Pavel Drábek a Gabriela Holubová. *Parciální diferenciální rovnice. Západočeská univerzita v Plzni*, 2011.
- [2] Jiří Bouchala. *Variační metody. Matematika pro inženýry 21. století*, 2012.
- [3] J. Austin Cottrell, Thomas J. R. Hughes, and Yuri Bazilevs. *Isogeometric Analysis: Toward Integration of CAD and FEA*. Wiley Publishing, 1st edition, 2009. ISBN 0470748737, 9780470748732.
- [4] CSc. Doc. ing. Jindřich Petruška. *Počítačové metody mechaniky 2. Ústav mechaniky teles FSI VUT v Brně*, 2013.
- [5] Jan Francu, Petra Nováčková, and Přemysl Janíček. Torsion of a non-circular bar. *Engineering Mechanics*, 19(1):45 – 60, 2012.
- [6] T.J.R. Hughes, J.A. Cottrell, and Y. Bazilevs. Isogeometric analysis: Cad, finite elements, nurbs, exact geometry and mesh refinement. *Computer Methods in Applied Mechanics and Engineering*, 194(39):4135 – 4195, 2005. ISSN 0045-7825. doi:<https://doi.org/10.1016/j.cma.2004.10.008>. URL <http://www.sciencedirect.com/science/article/pii/S0045782504005171>.
- [7] D.L. Logan. *A First Course in the Finite Element Method, SI Version*. Cengage Learning, 2011. ISBN 9781133715788. URL <https://books.google.sk/books?id=7eoKAAAAQBAJ>.
- [8] D. Martišek. *Matematické principy grafických systémů*. Littera, 2002. ISBN 9788085763195. URL <https://books.google.sk/books?id=7YkYtwAACAAJ>.
- [9] Les Piegl and Wayne Tiller. *The NURBS Book*. Springer-Verlag, London, UK, UK, 1995. ISBN 3-540-55069-0.
- [10] K. Rektorys. *Variational Methods in Mathematics, Science and Engineering*. 1980.
- [11] Karel Rektorys and Others. *Přehled užití matematiky I, II*. SNTL, 1988.
- [12] Bc. Jana Slabáková. *Modelování nurbs křivek a ploch v projektivním prostoru. Diplomová práce, FSI VUT v Brně*, 2009.
- [13] Svatopluk Šmirák. *Pružnost a plasticita I*. CERM, 1995. ISBN 80-7204-468-0.

A PRÍLOHA

A.1 Transformácia súradnej sústavy

Pri Isogeometrickej metóde, ale aj u iných isoparametrických metód, definujeme prvok (element) v parametrickej súradnej sústave. V tej spravidla definujeme bá-
zové funkcie a prvok v nej nadobúda jednoduchý geometrický tvar. Následne prvok transformujeme do fyzikálnej súradnej sústavy, v ktorej nadobudne svoj skutočný tvar. Tento proces býva často označovaný ako mapovanie elementu. Takýto prístup prináša mnohé výhody (napr. pri numerickej integrácii), no nesie so sebou určité úskalia. Považujeme preto za prínosné na tieto problémy poukázať a navrhnúť ich riešenie. Začneme trochu teoretickejšie a následne budeme jednotlivé poznatky aplikovať. Ako názorný príklad sme zvolili zostavenie vektoru pravých strán a matice tuhosti pri riešení Poissonovej rovnice z kapitoly 2.

A.1.1 Transformácia integrálov

Definícia A.1 (*Integračná oblasť*). Buď $\Omega \subset \mathbb{R}^2$ uzavretá ohraničená množina, potom sa Ω nazýva integračná oblasť.

Definícia A.2 (*Jakobiho matica*). Nech

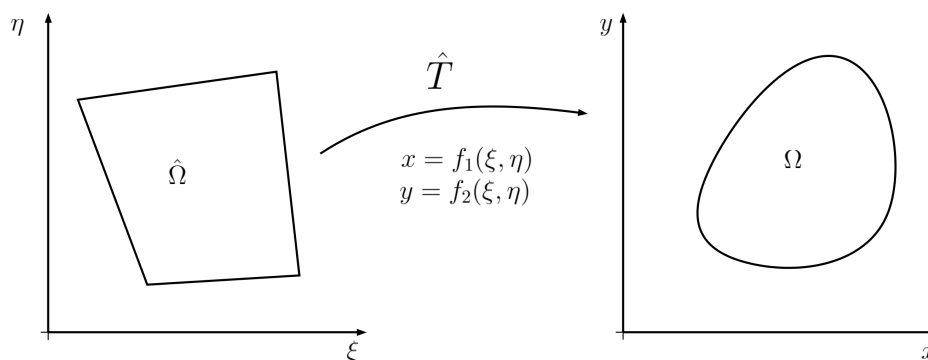
$$x = f_1(\xi, \eta), \quad y = f_2(\xi, \eta) \quad (\text{A.1})$$

sú funkcie dvoch premenných ξ, η , ktoré majú na $\hat{\Omega}$ spojité parciálne derivácie, potom maticu týchto parciálnych derivácií nazveme Jakobiho maticou

$$\mathbf{J}(\xi, \eta) = \begin{bmatrix} \frac{\partial f_1(\xi, \eta)}{\partial \xi} & \frac{\partial f_1(\xi, \eta)}{\partial \eta} \\ \frac{\partial f_2(\xi, \eta)}{\partial \xi} & \frac{\partial f_2(\xi, \eta)}{\partial \eta} \end{bmatrix}. \quad (\text{A.2})$$

Definícia A.3 (*jakobián*). Nech $\hat{T} : \mathbb{R}^2 \rightarrow \mathbb{R}^2$ je zobrazenie kde $\hat{T} = [f_1, f_2]$ pričom $f_1, f_2 : \mathbb{R}^2 \rightarrow \mathbb{R}$. Nech $\hat{\Omega} \subseteq D\hat{T}$, kde $D\hat{T}$ značí definičný obor \hat{T} , je oblasť a ku každému bodu $[\xi, \eta] \in \hat{\Omega}$ je rovnicami (A.1) priradený bod $[x, y] = [f_1(\xi, \eta), f_2(\xi, \eta)] \in \mathbb{R}^2$ tak, že platí:

- Ak $\hat{T}(\hat{\Omega}) = \Omega$, potom Ω je oblasť v \mathbb{R}^2 .
- Zobrazenie \hat{T} je vo vnútri množiny prosté.
- Funkcie (A.1) a ich parciálne derivácie sú na $\hat{\Omega}$ spojité.



Obr. A.1: Transformácia sústavy

Potom hovoríme, že transformačné rovnice (A.1) transformujú oblasť $\hat{\Omega}$ na oblasť Ω . Zobrazenie \hat{T} sa tak nazýva transformácia a determinant Jakobiho matice (A.2) označujeme ako jakobián transformácie \hat{T}

$$J(\xi, \eta) = \det(\mathbf{J}(\xi, \eta)) = \begin{vmatrix} \frac{\partial f_1(\xi, \eta)}{\partial \xi} & \frac{\partial f_1(\xi, \eta)}{\partial \eta} \\ \frac{\partial f_2(\xi, \eta)}{\partial \xi} & \frac{\partial f_2(\xi, \eta)}{\partial \eta} \end{vmatrix}. \quad (\text{A.3})$$

Veta A.1 (*Transformácia integrálu*). *Nech $\hat{\Omega}$ zobrazí pomocou rovníc (A.1) na Ω a funkcie $f_1(\xi, \eta)$, $f_2(\xi, \eta)$ majú spojité parciálne derivácie na $\hat{\Omega}$. Ďalej nech pre každý bod $[\xi, \eta] \in \hat{\Omega}$ je Jakobián (A.3) rôzny od nuly a funkcia $g = g(x, y)$ je spojitá na Ω . Potom podľa [11] platí*

$$\int_{\Omega} g(x, y) \, dx \, dy = \int_{\hat{\Omega}} g(f_1(\xi, \eta), f_2(\xi, \eta)) |J(\xi, \eta)| \, d\xi \, d\eta. \quad (\text{A.4})$$

Poznámka A.1. Transformáciou dosiahneme zmenu integračnej oblasti, čo nám mnohokrát uľahčí výpočet samotného integrálu. V našom prípade musíme poznamenať aj to, že pri zostavovaní matíc tuhosti, hmotnosti a vektoru pravých strán máme integrovať bázové funkcie $g(x, y)$ definované na Ω , avšak poznáme len predpis funkcie $\hat{g}(\xi, \eta)$ na $\hat{\Omega}$ a zobrazenie pomocou ktorého sa zobrazí na $g(x, y)$. Jednou z možností ako sa vysporiadať s týmto problémom je práve transformácia integrálu.

A.1.2 Parciálne derivácie zložených funkcií

V poznámke A.1 sme naznačili, že pri výpočtoch používame integrál (A.4). Ten popisuje integráciu funkcie $g(x, y)$ transformovanej z Ω na $\hat{\Omega}$. V praxi sa stretávame so situáciou kedy je potrebné integrovať parciálne derivácie tejto funkcie, vzhľadom

k premenným pôvodnej súradnej sústavy, konkrétne výrazy typu

$$\int_{\Omega} \left(\frac{\partial g(f_1(\xi, \eta), f_2(\xi, \eta))}{\partial f_1} + \frac{\partial g(f_1(\xi, \eta), f_2(\xi, \eta))}{\partial f_2} \right) |J(\xi, \eta)| \, d\xi \, d\eta.$$

Naviac sa ukáže, že tieto parciálne derivácie nepoznáme a budeme ich musieť vyjadriť pomocou derivácií funkcie \hat{g} vzhľadom k novým premenným (ξ, η) . Ako sa s touto situáciou vysporiadať nám napovie nasledujúca veta.

Veta A.2 (*Parciálna derivácia zloženej funkcie*). *Nech funkcie $x = f_1(\xi, \eta)$, $y = f_2(\xi, \eta)$ sú v bode $[\xi_0, \eta_0]$ diferencovateľné a nech funkcia $g = g(x, y)$ je diferencovateľná v odpovedajúcom bode $[x_0, y_0]$. Potom, podľa [10], aj zložená funkcia*

$$\hat{g}(\xi, \eta) = g(f_1(\xi, \eta), f_2(\xi, \eta))$$

je diferencovateľná, ako funkcia premenných (ξ, η) , a platí vzťah

$$\begin{aligned} \frac{\partial \hat{g}}{\partial \xi} &= \frac{\partial g}{\partial f_1} \frac{\partial f_1}{\partial \xi} + \frac{\partial g}{\partial f_2} \frac{\partial f_2}{\partial \xi}, \\ \frac{\partial \hat{g}}{\partial \eta} &= \frac{\partial g}{\partial f_1} \frac{\partial f_1}{\partial \eta} + \frac{\partial g}{\partial f_2} \frac{\partial f_2}{\partial \eta}. \end{aligned} \tag{A.5}$$

Predstavme si situáciu kedy sú nám známe derivácie $\frac{\partial \hat{g}}{\partial \xi}$, $\frac{\partial \hat{g}}{\partial \eta}$, no potrebujeme vyjadriť $\frac{\partial g}{\partial f_1}$, $\frac{\partial g}{\partial f_2}$. Vzťahy (A.5) si môžeme zapísať v maticovom tvare

$$\begin{bmatrix} \frac{\partial \hat{g}}{\partial \xi} \\ \frac{\partial \hat{g}}{\partial \eta} \end{bmatrix} = \begin{bmatrix} \frac{\partial f_1}{\partial \xi} & \frac{\partial f_2}{\partial \xi} \\ \frac{\partial f_1}{\partial \eta} & \frac{\partial f_2}{\partial \eta} \end{bmatrix} \begin{bmatrix} \frac{\partial g}{\partial f_1} \\ \frac{\partial g}{\partial f_2} \end{bmatrix}.$$

Vidíme, že vo vzťahu medzi vektormi parciálnych derivácií sa objavu skôr zavedená transponovaná Jakobihova matica \mathbf{J}^\top , (A.2). Opäť môžeme predpokladať, že jakobián transformácie je vždy rôznyi od nuly, čiže vieme, že Jakobihova matica je regulárna. Z lineárnej algebry vieme, že pre násobením inverznou transponovanou Jakobihovou maticou, získavame vzťahy pre výpočet vektoru $\begin{bmatrix} \frac{\partial g}{\partial f_1} \\ \frac{\partial g}{\partial f_2} \end{bmatrix}^\top$

$$\begin{bmatrix} \frac{\partial g}{\partial f_1} \\ \frac{\partial g}{\partial f_2} \end{bmatrix} = (\mathbf{J}^\top)^{-1} \begin{bmatrix} \frac{\partial \hat{g}}{\partial \xi} \\ \frac{\partial \hat{g}}{\partial \eta} \end{bmatrix}.$$

Keďže je Jakobiho matica rozmerov 2×2 , je možné jednoducho vyjadriť jej inverziu¹

$$\begin{bmatrix} \frac{\partial g}{\partial f_1} \\ \frac{\partial g}{\partial f_2} \end{bmatrix} = \frac{1}{\det(\mathbf{J}^\top)} \begin{bmatrix} \frac{\partial f_2}{\partial \eta} & -\frac{\partial f_2}{\partial \xi} \\ -\frac{\partial f_1}{\partial \eta} & \frac{\partial f_1}{\partial \xi} \end{bmatrix} \begin{bmatrix} \frac{\partial \hat{g}}{\partial \xi} \\ \frac{\partial \hat{g}}{\partial \eta} \end{bmatrix}.$$

Je známe, že $\det(A) = \det(A^\top)$ z čoho vyplýva $\det(\mathbf{J}^\top) = J$. Ak uvažime, že $\frac{1}{J} = J^{-1}$ a následne vzťah roznásobeníme, dostávame

$$\begin{aligned} \frac{\partial g}{\partial f_1} &= J^{-1} \left(\frac{\partial \hat{g}}{\partial \xi} \frac{\partial f_2}{\partial \eta} - \frac{\partial \hat{g}}{\partial \eta} \frac{\partial f_2}{\partial \xi} \right), \\ \frac{\partial g}{\partial f_2} &= J^{-1} \left(\frac{\partial \hat{g}}{\partial \eta} \frac{\partial f_1}{\partial \xi} - \frac{\partial \hat{g}}{\partial \xi} \frac{\partial f_1}{\partial \eta} \right). \end{aligned} \quad (\text{A.6})$$

Dospeli sme k finálnym vzťahom pre výpočet neznámych derivácií $\frac{\partial g}{\partial f_1}$, $\frac{\partial g}{\partial f_2}$ pomocou známych $\frac{\partial \hat{g}}{\partial \xi}$, $\frac{\partial \hat{g}}{\partial \eta}$.

Odvodili sme si niekoľko vzťahov, ktorých využitie si ukážeme na príkladoch v nasledujúcich dvoch podkapitolách.

A.1.3 Zostavenie lokálneho vektoru pravých strán

Pri riešení okrajovej úlohy v kapitole 5 sme si odvodili, ako spočítať jednotlivé členy potrebné na zostavenie sústavy lineárnych rovníc. Prvý si uvedieme výpočet k -tého prvku lokálneho vektoru pravej strany. S uvažovaním konštantnej funkciu f na prvku sme dostali dvojný integrál cez oblasť elementu

$$F_{\text{lok } k}^e = f \iint_{\Omega^e} R_k(x, y) \, dx \, dy.$$

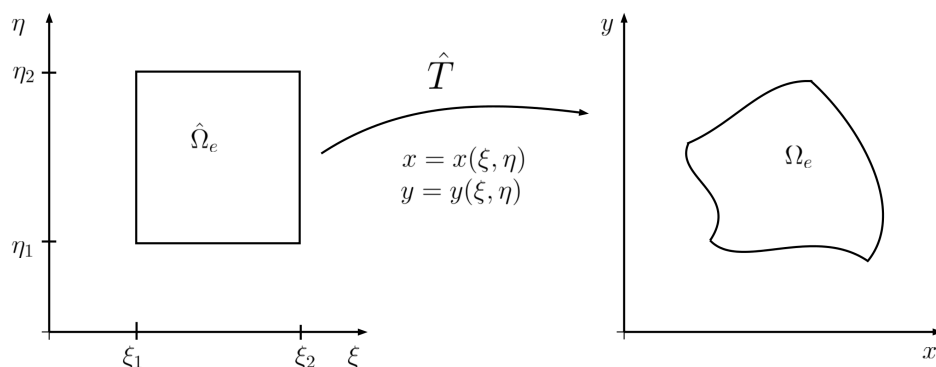
Integrujeme k -tu báзовú funkciu R_k cez oblasť elementu Ω^e . Index $k = 1, \dots, n$ značí lokálne číslo báзовej funkcie, ktorá je nenulová na danom elemente a n je počet takých funkcií.

Integrujeme cez dvojrozmernú oblasť Ω^e , ktorá môže nadobúdať obecné tvary. Ako sme v poznámke (A.1) naznačili zavedieme transformáciu \hat{T} ktorá pomocou transformačných vzťahov²

$$x = x(\xi, \eta), \quad y = y(\xi, \eta) \quad (\text{A.7})$$

¹ $\mathbf{A}^{-1} = \begin{bmatrix} a & b \\ c & d \end{bmatrix}^{-1} = \frac{1}{\det \mathbf{A}} \begin{bmatrix} d & -b \\ -c & a \end{bmatrix} = \frac{1}{ad-bc} \begin{bmatrix} d & -b \\ -c & a \end{bmatrix}.$

²Konkrétnu transformáciu a transformačné vzťahy sme si zaviedli v kapitole 5. Kvôli prehľadnosti a zachyteniu podstaty problému ich však ponecháme v tvare (A.7).



Obr. A.2: Transformácia elementu z parametrickej do kartézskej súradnej sústavy.

transformuje obdĺžnikovú oblasť $\hat{\Omega}^e$ na obecnjšiu oblasť Ω^e . Keďže poznáme transformačné vzťahy, z definície (A.3) vieme spočítať jakobián transformácie J , a môžeme aplikovať vzťah (A.4) pre transformáciu integrálu. Dostávame opäť dvojný integrál

$$F_{\text{lok } k}^e = f \iint_{\Omega^e} R_k(x, y) \, dx \, dy = f \iint_{\hat{\Omega}^e} R_k(x(\xi, \eta), y(\xi, \eta)) |J(\xi, \eta)| \, d\xi \, d\eta,$$

tentokrát cez obdĺžnikovú oblasť $\hat{\Omega}$.

Transformáciou sme vyriešili problém s komplikovanou integračnou oblasťou. Súčasne sme tiež vyriešili druhý problém a to, že nepoznáme priamo predpis funkcie $R_k(x, y)$ ale len predpis $R_k(x(\xi, \eta), y(\xi, \eta))$. Pre danú transformáciu \hat{T} sme v kapitole (5) zaviedli analytické vyjadrenie bázových funkcií

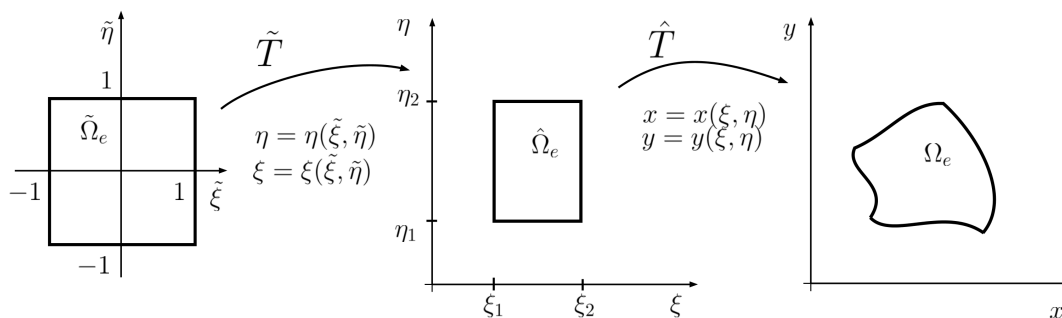
$$\hat{R}_k(\xi, \eta) = R_k(x(\xi, \eta), y(\xi, \eta)).$$

Dostávame integrál bázovej funkcie \hat{R}_k , prenášobenej absolútnou hodnotou jakobiánu, cez oblasť elementu v parametrickej súradnej sústave

$$F_{\text{lok } k}^e = f \iint_{\hat{\Omega}^e} \hat{R}_k(\xi, \eta) |J(\xi, \eta)| \, d\xi \, d\eta.$$

Je jasné, že integrácia sa v praxi musí vykonávať nejakou numerickou metódou. My sme zvolili jednu z najrozšírenejších, Gaussovu kvadraturu. Tá sa zvyčajne odvodzuje pre štvorcovú oblasť $\tilde{\Omega}(-1, 1) \times (-1, 1)$, my však integrujeme cez obdĺžnikovú oblasť $\hat{\Omega}^e(\xi_1, \xi_2) \times (\eta_1, \eta_2)$ a preto zavedieme transformáciu \tilde{T} . Tentokrát pomocou transformačných vzťahov

$$\xi = \xi(\tilde{\xi}, \tilde{\eta}), \quad \eta = \eta(\tilde{\xi}, \tilde{\eta}),$$



Obr. A.3: Postupná transformácia elementu.

transformujeme oblasť $\tilde{\Omega}$ na $\hat{\Omega}$. To sa prejaví tým, že v integrály pribudne jakobián transformácie \tilde{J} . Na finálny tvaru integrálu

$$F_{\text{lok } k}^e = f \iint_{\tilde{\Omega}^e} \hat{R}_k(\xi(\tilde{\xi}, \tilde{\eta}), \eta(\tilde{\xi}, \tilde{\eta})) |J(\xi, \eta)| |\tilde{J}(\tilde{\xi}, \tilde{\eta})| d\tilde{\xi} d\tilde{\eta},$$

môžeme použiť tabelované hodnoty integračných bodov a váh Gaussovej kvadratury. Celkovo sme teda element dva krát transformovali čo môžeme vidieť na obr. A.3.

A.1.4 Zostavenie lokálnej matice tuhosti

Pri odvodení členov vektoru pravých strán sme si ukázali, ako transformovať integrály bázových funkcií. Teraz k tomu pridáme poznatky o parciálnych deriváciách zložených funkcií, ktoré sa vyskytujú pri výpočte členov lokálnej matice tuhosti. Odvodením v kapitole 5 sme dospeli k dvojnému integrálu cez oblasť elementu

$$K_{\text{lok } kl}^e = \iint_{\Omega^e} \frac{\partial R_k(x, y)}{\partial x} \frac{\partial R_l(x, y)}{\partial x} + \frac{\partial R_k(x, y)}{\partial y} \frac{\partial R_l(x, y)}{\partial y} dx dy.$$

Integrand tvoria parciálne derivácie jednotlivých bázových funkcií R_k, R_l . Indexy $k, l = 1, \dots, n$ značia lokálne čísla bázových funkcií, ktoré sú nenulové na danom elemente a n je počet takých funkcií.

Prevedieme rovnaký postup transformácie, aký sme si popísali v druhom odseku predchádzajúcej kapitoly. Dostávame dvojný integrál

$$K_{\text{lok } kl}^e = \iint_{\hat{\Omega}^e} \left(\frac{\partial R_k(x, y)}{\partial x} \frac{\partial R_l(x, y)}{\partial x} + \frac{\partial R_k(x, y)}{\partial y} \frac{\partial R_l(x, y)}{\partial y} \right) |J(\xi, \eta)| d\xi d\eta$$

cez obdĺžnikovú oblasť elementu $\hat{\Omega}^e$.

Problém nastáva v tom, že nepoznáme predpis parciálnych derivácií našich bá-

zových funkcií vzhľadom k premenným x, y . Poznáme však funkciu

$$\hat{R}_k(\xi, \eta) = R_k(x(\xi, \eta), y(\xi, \eta)),$$

zadefinovanej na oblasti $\hat{\Omega}^e$. Z nej sme schopný vyjadriť parciálne derivácie vzhľadom k premenným ξ a η

$$\begin{aligned}\frac{\partial \hat{R}_k}{\partial \xi} &= \frac{\partial R_k}{\partial x} \frac{\partial x}{\partial \xi} + \frac{\partial R_k}{\partial y} \frac{\partial y}{\partial \xi}, \\ \frac{\partial \hat{R}_k}{\partial \eta} &= \frac{\partial R_k}{\partial x} \frac{\partial x}{\partial \eta} + \frac{\partial R_k}{\partial y} \frac{\partial y}{\partial \eta}.\end{aligned}$$

V podkapitole A.1.2 sme si vysvetlili ako z týchto vzťahov vyjadriť potrebné parciálne derivácie.

$$\begin{aligned}\frac{\partial R_k}{\partial x} &= J^{-1} \left(\frac{\partial \hat{R}_k}{\partial \xi} \frac{\partial y}{\partial \eta} - \frac{\partial \hat{R}_k}{\partial \eta} \frac{\partial y}{\partial \xi} \right), \\ \frac{\partial R_k}{\partial y} &= J^{-1} \left(\frac{\partial \hat{R}_k}{\partial \eta} \frac{\partial x}{\partial \xi} - \frac{\partial \hat{R}_k}{\partial \xi} \frac{\partial x}{\partial \eta} \right).\end{aligned}$$

Tie sme podľa vzťahu (A.6) schopný vyjadriť pomocou známych parciálnych derivácií funkcií \hat{R} a parciálnych derivácií transformačných rovníc (A.7). Dosadením do predchádzajúceho integrálu a jednoduchými algebraickými úpravami dostaneme

$$K_{\text{lok } kl}^e = \iint_{\hat{\Omega}^e} \left(a_1 \frac{\partial \hat{R}_k}{\partial \xi} \frac{\partial \hat{R}_l}{\partial \xi} - a_2 \left(\frac{\partial \hat{R}_k}{\partial \xi} \frac{\partial \hat{R}_l}{\partial \eta} + \frac{\partial \hat{R}_k}{\partial \eta} \frac{\partial \hat{R}_l}{\partial \xi} \right) + a_3 \frac{\partial \hat{R}_k}{\partial \eta} \frac{\partial \hat{R}_l}{\partial \eta} \right) J^{-1} d\xi d\eta.$$

Koeficienty a_1, a_2, a_3 sú zostavené z parciálnych derivácií transformačných vzťahov

$$a_1 = \left(\frac{\partial y}{\partial \eta} \right)^2 + \left(\frac{\partial x}{\partial \eta} \right)^2, \quad a_2 = \left(\frac{\partial y}{\partial \xi} \frac{\partial y}{\partial \eta} + \frac{\partial x}{\partial \xi} \frac{\partial x}{\partial \eta} \right), \quad a_3 = \left(\frac{\partial y}{\partial \xi} \right)^2 + \left(\frac{\partial x}{\partial \xi} \right)^2.$$

Môžeme si povšimnúť, že ide o rovnaké členy aké obsahuje Jakobiho matica \mathbf{J} , ktorú musíme počítať kvôli jakobiánu. Tohoto poznatku využijeme a vyjadríme si koeficienty pomocou prvkov Jakobiho matice

$$a_1 = J_{2,2}^2 + J_{2,1}^2, \quad a_2 = J_{1,2}J_{2,2} + J_{1,1}J_{2,1}, \quad a_3 = J_{1,2}^2 + J_{1,1}^2,$$

kde dolné indexy značia koordináty prvku v matici.

Ako bolo spomenuté v predchádzajúcej kapitole, integrály spočítame numericky

na prirodzenej oblasti $\tilde{\Omega}$. V integrále

$$K_{\text{lok } kl}^e = \iint_{\tilde{\Omega}^e} \left(a_1 \frac{\partial \hat{R}_k}{\partial \xi} \frac{\partial \hat{R}_l}{\partial \xi} - a_2 \left(\frac{\partial \hat{R}_k}{\partial \xi} \frac{\partial \hat{R}_l}{\partial \eta} + \frac{\partial \hat{R}_k}{\partial \eta} \frac{\partial \hat{R}_l}{\partial \xi} \right) + a_3 \frac{\partial \hat{R}_k}{\partial \eta} \frac{\partial \hat{R}_l}{\partial \eta} \right) J^{-1} |\tilde{J}| \, d\tilde{\xi} \, d\tilde{\eta},$$

pribudol jakobián \tilde{J} transformácie \tilde{T} .

B PRÍLOHA

B.1 Gaussove body a váhy

Algoritmus pre Gaussovu kvadratúru

Tento algoritmus vráti súradnice a váhové koeficienty Gaussových integračných bodov v závislosti na polynomickej stupni p .

```
function [Gauss_ip,wi]=Gauss(p)
if p==1
    Gauss_ip=[ 0.0000000000  0.0000000000]';
    wi=[      4.0000000000]';
elseif p==2
    Gauss_ip=[-0.5773502691 -0.5773502691; -0.5773502691  0.5773502691;...
              0.5773502691  0.5773502691;  0.5773502691 -0.5773502691]';
    wi=[      1.0000000000; 1.0000000000;  1.0000000000; 1.0000000000]';
elseif p==3
    Gauss_ip=[-0.7745966692 -0.7745966692;  0.0000000000 -0.7745966692;...
              0.7745966692 -0.7745966692; -0.7745966692  0.0000000000;...
              0.0000000000  0.0000000000;  0.7745966692  0.0000000000;...
              -0.7745966692  0.7745966692;  0.0000000000  0.7745966692;...
              0.7745966692  0.7745966692]';
    wi=[      0.3086419750; 0.4938271600;  0.3086419750; 0.4938271600;...
          0.7901234560; 0.4938271600;  0.3086419750; 0.4938271600;...
          0.3086419750]';
elseif p==4
    Gauss_ip=[-0.8611363115 -0.8611363115; -0.3399810435 -0.8611363115;...
              0.3399810435 -0.8611363115;  0.8611363115 -0.8611363115;...
              -0.8611363115 -0.3399810435; -0.3399810435 -0.3399810435;...
              0.3399810435 -0.3399810435;  0.8611363115 -0.3399810435;...
              -0.8611363115  0.3399810435; -0.3399810435  0.3399810435;...
              0.3399810435  0.3399810435;  0.8611363115  0.3399810435;...
              -0.8611363115  0.8611363115; -0.3399810435  0.8611363115;...
              0.3399810435  0.8611363115;  0.8611363115  0.8611363115;]';
    wi=[      0.1210029932; 0.2268518518;  0.2268518518; 0.1210029932;...
          0.2268518518; 0.4252933030;  0.4252933030; 0.2268518518;...
          0.2268518518; 0.4252933030;  0.4252933030; 0.2268518518;...
          0.1210029932; 0.2268518518;  0.2268518518; 0.1210029932;]';
end
```

