

Ettekanne aines
Matemaatika seminar III:
“Lõplike elementide meetod”

Indrek Mandre <indrek@mare.ee>

24. november, 2009

Kokkuvõte

Poissoni võrrand. Variatsioonmeetodid ja Ritzi meetod. Lõplike elementide meetodi idee. Lihtne näide ühemõõtmelisel juhul. Baasfunktsioonidest. Numbriline näide. Lõplike elementide meetodi algoritmi formaalses definitsioon. Kahemõõtmeline juht. Lisamärkused.

1 Poissoni võrrand

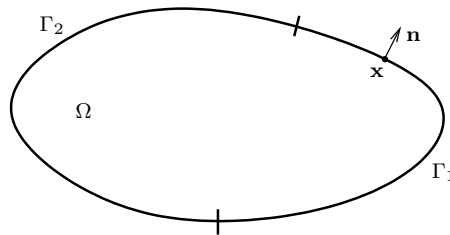
Vaatame võrrandit kujul

$$-\nabla \cdot (k(x)\nabla u(x)) = f(x). \quad (1)$$

Edaspidi eeldame, et $k(x) \geq k_0 > 0$. Klassikalises Poissoni võrrandis on $k(x) \equiv 1$ ja kahemõõtmelisel juhul saame seega

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = -f(x).$$

Võttes $f(x) = 0$ saame Laplace'i võrrandi. Meid huvitab selle võrrandi lahend mingis piirkonnas $\Omega \subset \mathbb{R}^N$ rajaga Γ (joonis 1).



Joonis 1: Piirkond Ω ja raja $\Gamma = \Gamma_1 \cup \Gamma_2$.

Ülesande lahendamiseks on antud ka rajatingimused, mis võivad olla kolme liiki:

1. Dirichlet' ehk olulised rajatingimused:

$$u(x) = g(x), \quad x \in \Gamma;$$

2. Neumanni ehk loomulikud rajatingimused:

$$k(x)\nabla u(x) \cdot \mathbf{n} = h(x), \quad x \in \Gamma.$$

3. Segaülesanne, kus raja lõpliku arvuga eri osadele on antud erinevat tüüpi rajatingimused:

$$\begin{aligned} u(x) &= g(x), & x \in \Gamma_1; \\ k(x)\nabla u(x) \cdot \mathbf{n} &= h(x), & x \in \Gamma_2; \end{aligned}$$

kus siis

$$\Gamma = \Gamma_1 \cup \Gamma_2, \quad \Gamma_1 \cap \Gamma_2 = \emptyset, \quad \Gamma_1 \neq \emptyset, \quad \Gamma_2 \neq \emptyset.$$

2 Ritzi meetod

Tuletades meelde eelmises seminaris räägitud variatsioonmeetodeid, vaatame kuidas käis ülesande lahendamine Ritzi meetodiga.

Valime mingi funktsiooni $w(x) \in C^2(\overline{\Omega})$, mis rahuldab Dirichlet' rajatingimusi ($w(x)|_{\Gamma_1} = g(x)$) ja saame uue otsitava funktsiooni z (ehk siis $u = w + z$). z kuulub ruumi D_A^0 :

$$D_A^0 = \{z \in C^2(\overline{\Omega}) : z(\Gamma_1) = 0, k(x)\nabla z(x) \cdot \mathbf{n}|_{\Gamma_2} = h(x) - k(x)\nabla w(x) \cdot \mathbf{n}\}.$$

Korrutame diferentsiaalvõrrandit (1) skalaarselt funktsiooniga $v \in D_A^0$:

$$-\int_{\Omega} \nabla \cdot (k(x)\nabla u(x)) v(x) dx = \int_{\Omega} f(x)v(x) dx. \quad (2)$$

Kasutades Greeni esimest valemit, saame vasaku poole lahti teha:

$$-\int_{\Gamma} k(\nabla u \cdot \mathbf{n})v d\Gamma + \int_{\Omega} k\nabla u \cdot \nabla v dx = \int_{\Omega} f v dx. \quad (3)$$

ehk

$$-\int_{\Gamma_1} k(\nabla u \cdot \mathbf{n})v d\Gamma - \int_{\Gamma_2} k(\nabla u \cdot \mathbf{n})v d\Gamma + \int_{\Omega} k\nabla u \cdot \nabla v dx = \int_{\Omega} f v dx.$$

Esimene integraal on 0 ja järgi jääb

$$-\int_{\Gamma_2} h(x)v(x) d\Gamma + \int_{\Omega} k(x)\nabla u(x) \cdot \nabla v(x) dx = \int_{\Omega} f(x)v(x) dx.$$

Edasi läheme üle energiaruumile H_A (mis on D_A^0 sulund ruumis $H^1(\Omega)$) ehk

$$H_A = \{z \in H^1(\Omega) : z(\Gamma_1) = 0\}$$

(rajatingimus $\nabla z(x) \cdot \mathbf{n}|_{\Gamma_2} = h(x) - \nabla w(x) \cdot \mathbf{n}$ langeb ära) ja saame üldistatud ülesande: leida $z \in H_A$, nii et $u = w + z$ rahuldab võrrandit

$$\int_{\Omega} k(x)\nabla u(x) \cdot \nabla v(x) dx = \int_{\Omega} f(x)v(x) dx + \int_{\Gamma_2} h(x)v(x) d\Gamma \text{ iga } v \in H_A \text{ korral.} \quad (4)$$

Ritzi meetodi korral otsime lahendit energiaruumi mingis n -mõõtmelises alamruumis H_A^n kujul

$$u = w + \sum_{j=1}^n c_j v_j, \quad (5)$$

kus $\{v_j\}$ on selle ruumi baas ja saame süsteemi

$$\int_{\Omega} k \nabla \left(w + \sum_{j=1}^n c_j v_j \right) \cdot \nabla v_i dx = \int_{\Omega} f v_i dx + \int_{\Gamma_2} k h v_i d\Gamma, \quad i = 1, 2, \dots, n \quad (6)$$

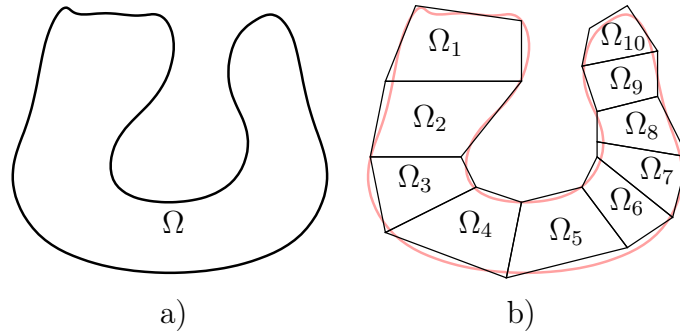
ehk

$$\sum_{j=1}^n c_j \int_{\Omega} k \nabla v_j \cdot \nabla v_i dx = \int_{\Omega} f v_i dx + \int_{\Gamma_2} k h v_i d\Gamma - \int_{\Omega} k \nabla w \cdot \nabla v_i dx, \quad i = 1, 2, \dots, n.$$

Tegu on n tundmatuga lineaarse võrrandisüsteemiga. Selle lahendamisel saame koefitsendid lahendile (5).

3 Lõplike elementide idee

Variatsioonmeetodid on rakendatavad juhul, kui meil õnnestub leida funktsioon w ja baas energiaruumile H_A . Kui piirkond Ω ei ole triviaalne (joonis 2a), on see komplitseeritud.



Joonis 2: Mittetriviaalne piirkond (a) ja selle jagamine tükki (b).

Et leida lahendit antud keerulise piirkonna jaoks on idee jagada see piirkond lihtsa kujuga tükki (joonis 2b) ja siis iga tüki peal rakendada variatsioonmeetodit. Samas me otsime lahendit, mis oleks pidev ehk eri alampiirkondade kokkupuutejoontel omaks sama väärtust. Lõplike elementide meetodi idee on valida sellised baasifunktsioonid, et mingi elemendi Ω_e lahendi

$$u^e(x) = \sum_{j=1}^n c_j^e v_j^e(x) \quad (7)$$

kordajad c_j^e on funktsiooni u^e väärtused mingites kindlates selle alampiirkonna punktides ja funktsioonid $v_j^e(x)$ on interpoleerimisfunktsioonid (mis nende punktide vahel väärtusi interpoleerivad). Valides ühised punktid osapiirkondade nurkades ja tuues sisse lisatingimuse, et eri osapiirkondade samade punktide väärtused on võrdsed, saame ka tulemuse pidevuse tingimuse täidetud.

4 Lihtsustatud näide ühemõõtmelisel juhul

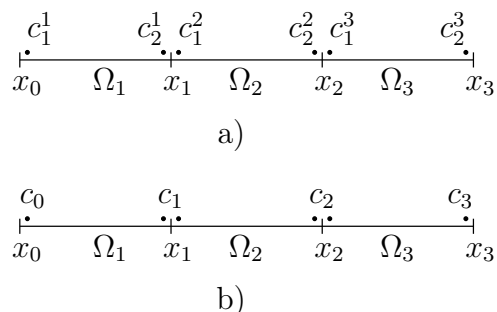
Et kogu idee selgitada, vaatame lihtsustatud näidet ühemõõtmelisel juhul. Olgu ülesanne leida u , mis rahuldaks piirkonnas Ω valemit

$$-\frac{d}{dx} \left(k \frac{du}{dx} \right) = f.$$

Jagame piirkonna Ω kolmeks osaks (joonis 3a) ja olgu ülesande rajatingimused antud kui:

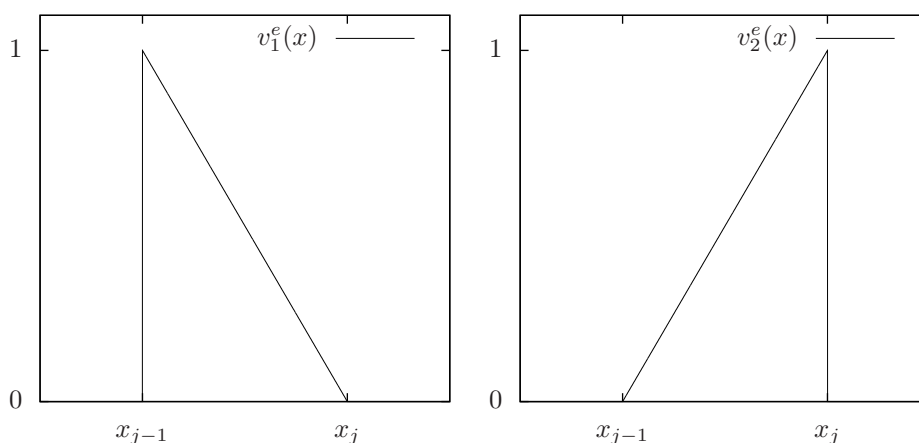
$$u(x_0) = g_0, \quad k \frac{du}{dx} \Big|_{x=x_3} = h_0.$$

Kasutame igas osapiirkonnas ainult kahte punkti.



Joonis 3: Lõplike elementide meetod, ühemõõtmeline juht.

Pidevuse jaoks peab $c_2^1 = c_1^2 = c_1$ ja $c_2^2 = c_1^3 = c_2$ ehk järgi jääb neli erinevat tundmatut (joonis 3b). Lisaks algtingimustest saame, et $c_0 = u_0$. Millised oleksid aga baasifunktsioonid? Antud juhul on tegu lihtsa lineaarse interpolatsiooniga ja baasifunktsioonid on näha joonisel 4.



Joonis 4: Lihtsaimad baasifunktsioonid ühemõõtmelisele alampiirkonnale Ω_j .

Et lahenduseni jõuda, rakendame Ritzi meetodit. Korrutame algse võrrandi (1) skalaarselt funktsiooniga v ja integreerime ositi (jõuame valemieni (3))¹. Iga elementidele Ω_e saame

$$\int_{x_{e-1}}^{x_e} \left(k \frac{du}{dx} \frac{dv}{dx} - fv \right) dx = k \frac{du}{dx} v \Big|_{x_e} - k \frac{du}{dx} v \Big|_{x_{e-1}} .$$

Tähistades

$$Q^e = k \frac{du}{dx} \Big|_{x_e} ,$$

saame

$$\int_{x_{e-1}}^{x_e} \left(k \frac{du}{dx} \frac{dv}{dx} - fv \right) dx = v(x_e)Q^e - v(x_{e-1})Q^{e-1} .$$

Nüüd asendame u antud elemendi otsitava lahendiga (7). Lisaks asendame v nende kahe erineva baasifunktsiooniga v_1^e ja v_2^e . Iga elemendi kohta saame kaks võrrandit:

$$\int_{x_{e-1}}^{x_e} \left[k \frac{d}{dx} \left(\sum_{j=1}^n c_j^e v_j^e \right) \frac{dv_1^e}{dx} - f v_1^e \right] dx = v_1^e(x_e)Q^e - v_1^e(x_{e-1})Q^{e-1} ,$$

$$\int_{x_{e-1}}^{x_e} \left[k \frac{d}{dx} \left(\sum_{j=1}^n c_j^e v_j^e \right) \frac{dv_2^e}{dx} - f v_2^e \right] dx = v_2^e(x_e)Q^e - v_2^e(x_{e-1})Q^{e-1} .$$

Edasi tähistame

$$K_{ij}^e = \int_{x_{e-1}}^{x_e} \left(k \frac{dv_j^e}{dx} \frac{dv_i^e}{dx} \right) dx$$

$$f_i^e = \int_{x_{e-1}}^{x_e} f v_i^e dx$$

¹Võrreldes valemite (7) valemiga (5) on w paigutatud juba sinna sisse kui üks baasifunktsioonidest. Kui Ritzi meetodil tehti võrrand kohe lahti ja liigutati w integraal paremale, siis siin me jätame ta süsteemi sisse ja liigutame paremale alles lõplikus maatrikskujus.

ning kirjutame kõik kuus võrrandit välja:

$$\begin{aligned}
\sum_{j=1}^n K_{1j}^1 c_j^1 &= f_1^1 + v_1^1(x_1)Q^1 - v_1^1(x_0)Q^0, \\
\sum_{j=1}^n K_{2j}^1 c_j^1 &= f_2^1 + v_2^1(x_1)Q^1 - v_2^1(x_0)Q^0, \\
\sum_{j=1}^n K_{1j}^2 c_j^2 &= f_1^2 + v_1^2(x_2)Q^2 - v_1^2(x_1)Q^1, \\
\sum_{j=1}^n K_{2j}^2 c_j^2 &= f_2^2 + v_2^2(x_2)Q^2 - v_2^2(x_1)Q^1, \\
\sum_{j=1}^n K_{1j}^3 c_j^3 &= f_1^3 + v_1^3(x_3)Q^3 - v_1^3(x_2)Q^2, \\
\sum_{j=1}^n K_{2j}^3 c_j^3 &= f_2^3 + v_2^3(x_3)Q^3 - v_2^3(x_2)Q^2.
\end{aligned}$$

Lihtsustame:

$$\begin{aligned}
K_{11}^1 c_0 + K_{12}^1 c_1 &= f_1^1 - v_1^1(x_0)Q^0, \\
K_{21}^1 c_0 + K_{22}^1 c_1 &= f_2^1 + v_2^1(x_1)Q^1, \\
K_{11}^2 c_1 + K_{12}^2 c_2 &= f_1^2 - v_1^2(x_1)Q^1, \\
K_{21}^2 c_1 + K_{22}^2 c_2 &= f_2^2 + v_2^2(x_2)Q^2, \\
K_{11}^3 c_2 + K_{12}^3 c_3 &= f_1^3 - v_1^3(x_2)Q^2, \\
K_{21}^3 c_2 + K_{22}^3 c_3 &= f_2^3 + v_2^3(x_3)Q^3.
\end{aligned}$$

Liidame valemid 2 ja 3 ning 4 ja 5. Siin saame taandada² osad paremal olevad elemendid:

$$\begin{aligned}
K_{11}^1 c_0 + K_{12}^1 c_1 &= f_1^1 - v_1^1(x_0)Q^0, \\
K_{21}^1 c_0 + (K_{22}^1 + K_{11}^2) c_1 + K_{12}^2 c_2 &= f_2^1 + f_1^2, \\
K_{21}^2 c_1 + (K_{22}^2 + K_{11}^3) c_2 + K_{12}^3 c_3 &= f_2^2 + f_1^3, \\
K_{21}^3 c_2 + K_{22}^3 c_3 &= f_2^3 + v_2^3(x_3)Q^3.
\end{aligned}$$

Järgmise sammuna asendame siia sisse algtingimused ($c_0 = g_0$ ja $Q^3 = h_0$). Saame:

$$\begin{aligned}
K_{11}^1 g_0 + K_{12}^1 c_1 &= f_1^1 - v_1^1(x_0)Q^0, \\
K_{21}^1 g_0 + (K_{22}^1 + K_{11}^2) c_1 + K_{12}^2 c_2 &= f_2^1 + f_1^2, \\
K_{21}^2 c_1 + (K_{22}^2 + K_{11}^3) c_2 + K_{12}^3 c_3 &= f_2^2 + f_1^3, \\
K_{21}^3 c_2 + K_{22}^3 c_3 &= f_2^3 + v_2^3(x_3)h_0.
\end{aligned}$$

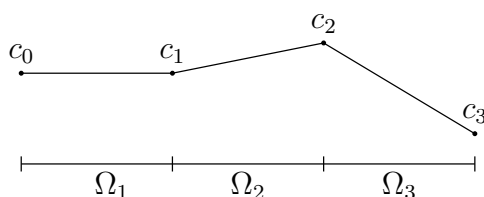
²Võrrandite summeerides saadud suurused $Q_-^e + (-Q_+^e)$ ei pruugi olla nullid, vaid võivad olla võrdsed väljaspoolt tuleva mõjuga (ette antud sisendandmed probleemi püstituses).

Meil on neli lineaarset võrrandit ja neli tundmatut $(c_1, c_2, c_3, Q^0)^3$. Nende leidmiseks tuleb antud lineaarvõrrandisüsteem lahendada. Tundmatute c_1, c_2, c_3 leidmiseks jääb järgi süsteem

$$\begin{bmatrix} K_{22}^1 + K_{11}^2 & K_{12}^2 & 0 \\ K_{21}^2 & K_{22}^2 + K_{11}^3 & K_{12}^3 \\ 0 & K_{21}^3 & K_{22}^3 \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix} = \begin{bmatrix} f_2^1 + f_1^2 - K_{21}^1 g_0 \\ f_2^2 + f_1^3 \\ f_2^3 + v_2^3(x_3)h_0 \end{bmatrix}. \quad (8)$$

Lahendades antud süsteemi, saame välja arvutada otsitava suuruse u ja ka tema tuletise kasutades valemeid (7).

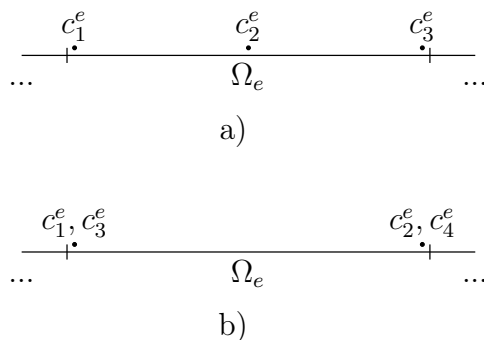
Kogu lahendi võimalik kuju on joonisel 5.



Joonis 5: Võimalik lahend, konstandid c_j määravad $u(x)$ väärtuse eri kohtades.

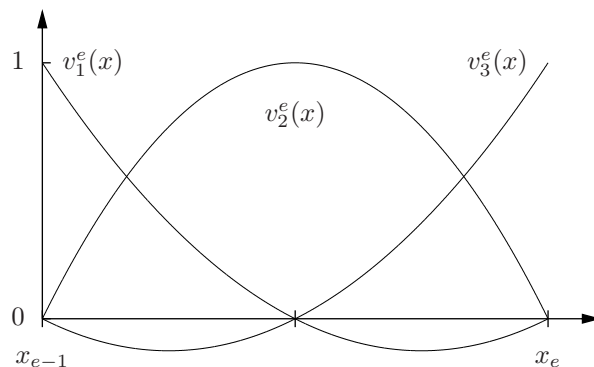
4.1 Baasifunktsioonidest

Eelmises näites piirdusime kahe punktiga elemendi kohta. Samas me võime elemendi sisse paigutada veel lisapunkte (joonis 6a). Ühe lisapunktiga saame kasutada Lagrange'i interpolatsioonifunktsioone (joonis 7).



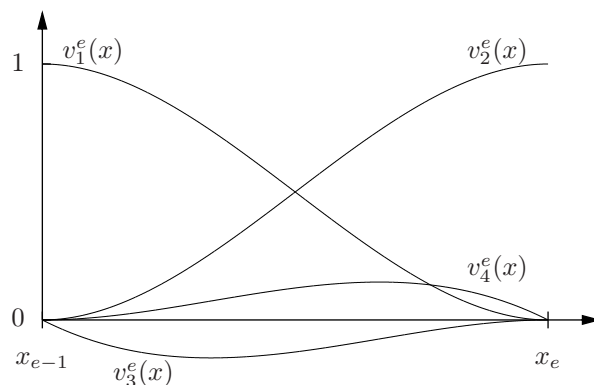
Joonis 6: Lisapunktide lisamine elementide sisse.

³Esimese võrrandi ära jätmisel jäävad ainult konstandid c_1, c_2, c_3 ja kolm võrrandit.



Joonis 7: Lagrange'i interpolatsioonifunktsioonid kolmele punktile.

Antud funktsioonide kasutamisega saadud lahend on pidev, samas tema tuletis ei ole pidev. Et saada pideva tuletisega (siledamat) tulemust võib kasutada Hermite'i interpoleerimisfunktsioone (joonis 8). Viimased interpoleerivad nii funktsiooni väärtust kui ka tema tuletist. Seega igas punktis on vaja määrata kaks väärtust - funktsiooni väärtus ja tema tuletis (joonis 6b).



Joonis 8: Hermite'i interpolatsioonifunktsioonid kahele punktile.

Erinevaid interpoleerimisfunktsioone erinevate arvu punktide ja ka suurema-dimensionaalsete ruumide jaoks võib leida raamatutest.

4.2 Kommentaarid

Ühemõõtmelist lõplike elementide meetodit võib kasutada ka diskreetsete süsteemide lahendamiseks (nihete leidmine vedrude süsteemis, pingete leidmine takistite ahelas), erinevates mehaanika probleemides nagu kandekaarte pingete arvutamisel, varda koormuse ja painde arvutustes jne. Lisaks saab lahendada ka omaväärtusülesandeid, kus diferentsiaalvõrrand teisendatakse lihtsaks maatriksite omaväärtusülesandeks.

Ajast sõltuvate probleemide lahendamisel kas lisatakse aeg juurde kui teine dimensioon või tehakse eeldus, et tulemusel on võimalik teha muutujate eraldus

$u(x, t) = T(t)X(x)$. Sellisel juhul lahendatakse ülesanne tavalise lõplike elementide meetodiga kandes muutuja t läbi valemite ja jõutakse HDV süsteemini, mille lahendamisel saadakse ajast sõltuvad koefitsendid $c_j(t)$.

4.3 Numbriline näide

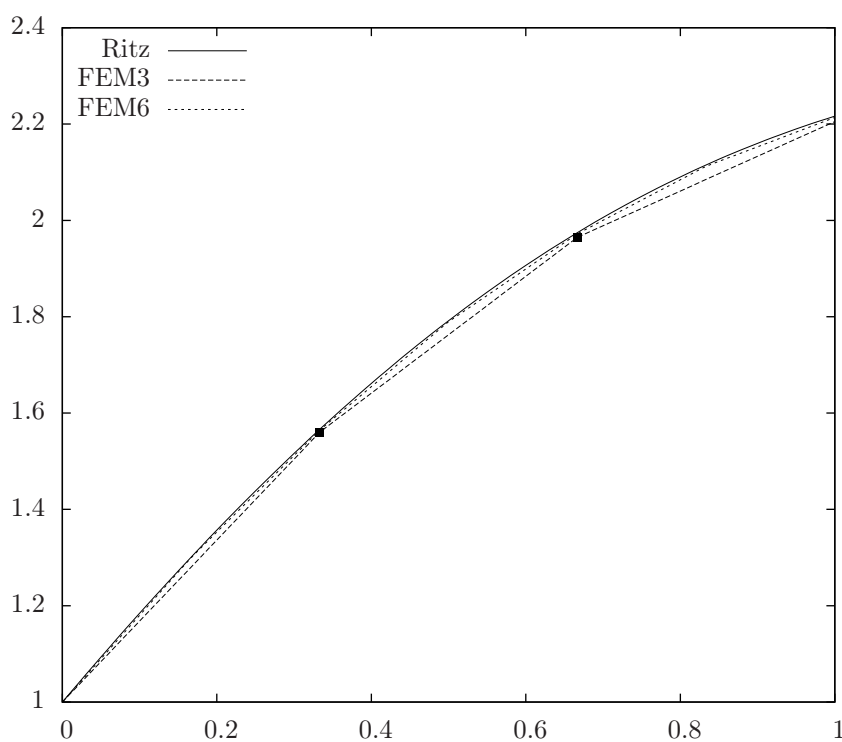
Vaatame ülesannet

$$-\frac{d^2u}{dx^2} - u = -x^2, \quad \Omega = (0, 1), \quad u(0) = 1, \quad u'(1) = 0.5.$$

Antud juhul on valemis vasakul poolel veel lisaelement $-u$ ja K_{ij}^e saab kuju

$$K_{ij}^e = \int_{x_{e-1}}^{x_e} \left(k \frac{dv_j^e}{dx} \frac{dv_i^e}{dx} - v_i^e v_j^e \right) dx.$$

Tulemuste võrdlused on näha joonisel 9 ja tabelis 1 (tulemuse genereerinud programmi kood on antud lisas A).



Joonis 9: Numbriline näide: Ritzi meetodi ja lõplike elementide lahend.

x	Ritz	FEM3	FEM6
0.0	1.000	1.000	1.000
0.1	1.184	1.168	1.180
0.2	1.357	1.336	1.353
0.3	1.516	1.504	1.512
0.4	1.662	1.641	1.655
0.5	1.792	1.763	1.790
0.6	1.907	1.884	1.899
0.7	2.006	1.989	2.000
0.8	2.090	2.061	2.084
0.9	2.160	2.133	2.153
1.0	2.216	2.205	2.214

Tabel 1: Numbriline näide: Ritzi meetodi ja lõplike elementide lahend.

5 Sammud lõplike elementide rakendamisel

Järgmine nimekiri kirjeldab süstemaatilist algoritmi lõplike elementide meetodi rakendamiseks [Reddy2].

1. Etteantud piirkonna diskretiseerimine elementideks.
 - (a) Piirkonna jagamine elementideks.
 - (b) Elementide ja nende sees olevate punktide nummerdamine.
 - (c) Elementide geomeetriliste omaduste (koordinaatide, pindalade, pikkuste) leidmine.
2. Elementide võrrandite tuletamine.
 - (a) Variatsioonmeetodite alusel elemendi jaoks võrrandi koostamine.
 - (b) Hinnangu
$$u^e = \sum_{i=1}^n c_i^e v_i^e$$
asendamine võrrandisse.
 - (c) Interpolatsioonifunktsioonide valik, koefitsientide K_{ij}^e ja f_i^e arvutamine.
3. Individuaalsete elementide võrrandite kokkupanek.
 - (a) Identifitseerida elementide vahelised ühised punktid ja kordajad.
 - (b) Sekundaarsete muutujate vaheliste seoste selgitamine elementide servadel.
 - (c) Elemendi võrrandite liitmine, lihtsustamine.
4. Rajatingimuste rakendamine.
5. Lahendi leidmine kokkupanud võrranditest.

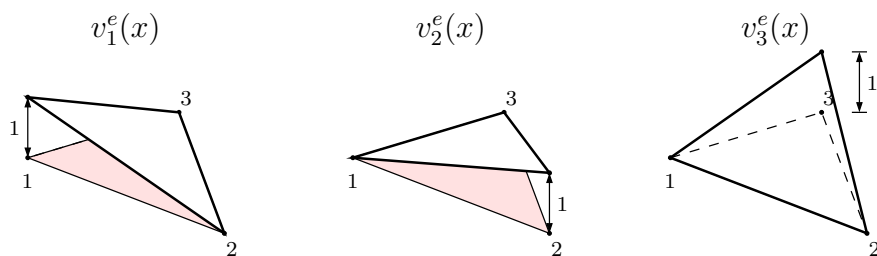
6. Lõpparvutused.

- (a) Otsitava lahendi väärtuste ja muude nendest tulenevate otsitavate suuruste arvutamine.
- (b) Tulemuse graafiline või numbriline esitamine ja interpretatsioon.

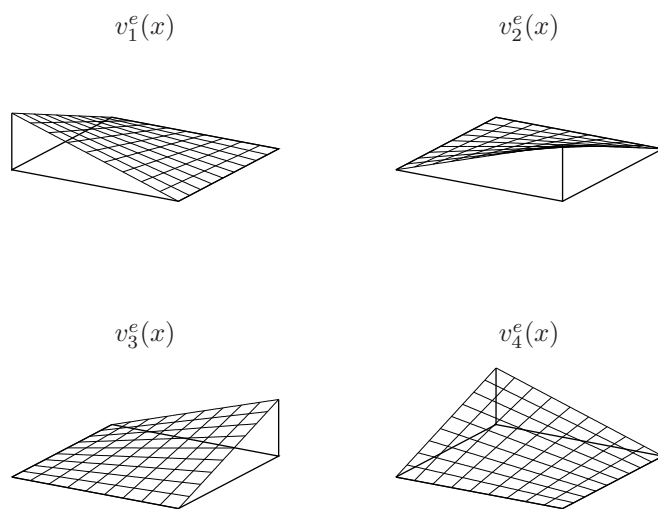
6 Lõplike elementide meetod kahes dimensioonis

Kahedimensionaalses ja suuremates ruumides piirkonna Ω diskretiseerimisel elementideks ei pruugi uus saadud $\cup \Omega_e$ olla sama mis Ω (joonis 2b). See toob tulemuses kaasa mingisuguse vea.

Millised võiksid olla samas elemendid? Kõige lihtsam on kasutada kolmnurki (joonis 10) või nelinurki (joonis 11). Samas ka keerulisemad kujundid on võimalikud, samas see teeb integraalid K_{jk}^e arvutamise raskemaks ja lisaks on raskendatud ka interpolatsioonifunktsioonide leidmine.



Joonis 10: Kolmnurksete elementide lihtsaimad baasifunktsioonid.



Joonis 11: Nelinurksete elementide lihtsaimad baasifunktsioonid.

7 Lisamärkused

1. Siin kasutasime Ritz'i meetodit, samas iga variatsioonmeetod (näiteks Galerkin'i meetod) sobib võrrandite koostamiseks.
2. Etteantud algoritmi sammud 1-6 on ühised igale probleemile. Tegu on süstemaatilise viisiga lahendada probleeme kasutades lõplike elementide meetodit.
3. Meetodit saab kasutada iga probleemi jaoks, mis on esitatav operaatorvõrrandiga

$$Au = f.$$

4. Koefitsientide K_{ij}^e ja f_i^e integreerimise võib samuti läbi viia numbriliselt. Sõltuvalt interpoleerimisfunktsioonidest võivad nad olla nii keerukad, et muud võimalust ei olegi.
5. Mudel tekitab kolme sorti vigu:
 - (a) Piirkonna viga, mis tekib piirkonna ebatäpsel esitamisel läbi elementide (näiteks piirkond on ümmargune, elemendid kandilised).
 - (b) Arvutusvead koefitsientide K_{ij}^e ja f_i^e arvutamisel ja üldiselt arvutite ebatäpse reaalarvude esituse tõttu.
 - (c) Hinnangu vead, mis tulenevad sellest, et me interpoleerime otsitavat suurust.
6. Sekundaarse muutuja pidevuse eeldus, mida me kasutasime muutujate Q_e taandamiseks, ei pea paika saadud tulemuses. S.t. kui

$$\left. \frac{du}{dx} \right|_{x_{e-}} + \left(- \left. \frac{du}{dx} \right|_{x_{e+}} \right) = 0,$$

siis

$$\left. \frac{du^e}{dx} \right|_{x_{e-}} + \left(- \left. \frac{du^{e+1}}{dx} \right|_{x_{e+}} \right) \neq 0.$$

7. Ritz'i meetodit läbi saadud maatriksvõrrandis (8) on süsteemimaatriks sümmeetriline. Lisaks on ta ka lintmaatriks, kus elemendid on diagonaalil või diagonaali lähedal.

Kasutatud kirjandus

[Janno1] Jaan Janno, "Variatsioonmeetodid", 2001.

[Reddy1] J.N.Reddy, "Applied Functional Analysis and Variational Methods in Engineering", 1986.

[Reddy2] J.N.Reddy, "An Introduction to the Finite Element Method, third edition", 2006.

[Janno2] Jaan Janno, "Arvutusmeetodid", 2008.

A MAXIMA CAS programmikood

A.1 ritz.m

```
1 f(x):=-x*x;
2 v(i):=x*x**(i-1);
3
4 Kij(i,j):=integrate(diff(v(j),x)*diff(v(i),x)-v(j)*v(i),x,0,1);
5 Fi(i):=integrate(v(i)*f(x)+v(i),x,0,1)+0.5*ev(v(i),x=1);
6
7 A: matrix(
8   [Kij(1,1), Kij(1,2), Kij(1,3), Kij(1,4), Kij(1,5), Kij(1,6)],
9   [Kij(2,1), Kij(2,2), Kij(2,3), Kij(2,4), Kij(2,5), Kij(2,6)],
10  [Kij(3,1), Kij(3,2), Kij(3,3), Kij(3,4), Kij(3,5), Kij(3,6)],
11  [Kij(4,1), Kij(4,2), Kij(4,3), Kij(4,4), Kij(4,5), Kij(4,6)],
12  [Kij(5,1), Kij(5,2), Kij(5,3), Kij(5,4), Kij(5,5), Kij(5,6)],
13  [Kij(6,1), Kij(6,2), Kij(6,3), Kij(6,4), Kij(6,5), Kij(6,6)]
14 );
15 B: matrix(
16   [Fi(1)],
17   [Fi(2)],
18   [Fi(3)],
19   [Fi(4)],
20   [Fi(5)],
21   [Fi(6)]
22 );
23
24 C: float(invert(A).B);
25 fun(x):=1+C[1][1]*v(1)+C[2][1]*v(2)+C[3][1]*v(3)+C[4][1]*v(4)+C[5][1]*v(5)+C[6][1]*v(6);
26
27 D: zeromatrix(101,2);
28 for k:0 thru 100 do (D[k+1][1]:float(k * 1/100), D[k+1][2]:fun(k * 1/100+1e-32))$
29 D:
30 outf: openw("ritz.dat");
31 write_data(D, outf, space);
32 close(outf);
33
34 D: zeromatrix(11,2);
35 for k:0 thru 10 do (D[k+1][1]:float(k * 1/10), D[k+1][2]:ev(fun(k * 1/10+1e-32), nouns)) $
36 D:
37 outf: openw("ritz_tabel.dat");
38 write_data(D, outf, space);
39 close(outf);
```

A.2 fem3.m

```
1 g0:1;
2 h0:0.5;
3 N:3.0;
4 h:1.0/N;
5 v(x,e,j):= if abs(e*h-h/2-x) <= h/2 + 1e-16 then
6   (if is(equal(j,1)) then ((e*h-x)/h) else (1-(e*h-x)/h)) else (0);
7 dv(x,e,j):= if abs(e*h-h/2-x) <= h/2 + 1e-16 then
8   (if is(equal(j,1)) then (-1/h) else (1/h)) else (0);
9 b:-1;
10 f(x):=-x*x;
11
12 Keij(e,i,j) := romberg('dv(x,e,j)*'dv(x,e,i)+b*'v(x,e,i)*'v(x,e,j), x, (e-1)*h, e*h);
13 Fei(e,j) := romberg(f(x)*'v(x,e,j), x, (e-1)*h, e*h);
14
15 A: matrix(
16   [Keij(1,1,1), Keij(1,1,2), 0, 0],
17   [Keij(1,2,1), Keij(1,2,2)+Keij(2,1,1), Keij(2,1,2), 0],
18   [0, Keij(2,2,1), Keij(2,2,2)+Keij(3,1,1), Keij(3,1,2)],
19   [0, 0, Keij(3,2,1), Keij(3,2,2)]
20 );
21 F: matrix(
22   [Fei(1,1)-v(0,1,1)*Q0],
23   [Fei(1,2)+Fei(2,1)],
24   [Fei(2,2)+Fei(3,1)],
25   [Fei(3,2)+v(1,3,2)*Q3]
26 );
27
28 c0:g0;
29 Q3:h0;
30
31 A: matrix(
32   [Keij(1,2,2)+Keij(2,1,1), Keij(2,1,2), 0],
33   [Keij(2,2,1), Keij(2,2,2)+Keij(3,1,1), Keij(3,1,2)],
34   [0, Keij(3,2,1), Keij(3,2,2)]
35 );
36 F: matrix(
37   [Fei(1,2)+Fei(2,1) - Keij(1,2,1)*c0],
38   [Fei(2,2)+Fei(3,1)],
39   [Fei(3,2)+v(1,3,2)*Q3]
40 );
41
42 res: invert(A).F;
43 c1: res [1][1];
44 c2: res [2][1];
45 c3: res [3][1];
46
47 fun(x):=c0*'v(x,1,1)+c1*'v(x,1,2)+c1*'v(x,2,1)+c2*'v(x,2,2)+c2*'v(x,3,1)+c3*'v(x,3,2);
```

```

48
49 D:zeromatrix(101,2);
50 for k:0 thru 100 do (D[k+1][1]:float(k * 1/100), D[k+1][2]:ev(fun(k * 1/100),nouns)) $
51 outf: openw("fem.dat");
52 write_data(D, outf, space);
53 close(outf);
54
55 D:zeromatrix(11,2);
56 for k:0 thru 10 do (D[k+1][1]:float(k * 1/10), D[k+1][2]:ev(fun(k * 1/10),nouns)) $
57 outf: openw("fem_tabel.dat");
58 write_data(D, outf, space);
59 close(outf);

```

A.3 fem6.m

```

1  g0:1;
2  h0:0.5;
3  N:6.0;
4  h:1.0/N;
5  v(x,e,j):= if abs(e*h-h/2-x) <= h/2 + 1e-16 then
6    (if is(equal(j,1)) then ((e*h-x)/h) else (1-(e*h-x)/h)) else (0);
7  dv(x,e,j):= if abs(e*h-h/2-x) <= h/2 + 1e-16 then
8    (if is(equal(j,1)) then (-1/h) else (1/h)) else (0);
9  b:-1;
10 f(x):=-x*x;
11
12 Keij(e,i,j) := romberg('dv(x,e,j)*'dv(x,e,i)+b*'v(x,e,i)*'v(x,e,j), x, (e-1)*h, e*h);
13 Fei(e,j) := romberg(f(x)*'v(x,e,j), x, (e-1)*h, e*h);
14
15 A:matrix(
16   [Keij(1,1,1), Keij(1,1,2), 0, 0,0,0,0],
17   [Keij(1,2,1), Keij(1,2,2)+Keij(2,1,1), Keij(2,1,2), 0,0,0,0],
18   [0, Keij(2,2,1), Keij(2,2,2)+Keij(3,1,1), Keij(3,1,2),0,0,0],
19   [0,0, Keij(3,2,1), Keij(3,2,2)+Keij(4,1,1), Keij(4,1,2),0,0],
20   [0,0,0, Keij(4,2,1), Keij(4,2,2)+Keij(5,1,1), Keij(5,1,2),0],
21   [0,0,0,0, Keij(5,2,1), Keij(5,2,2)+Keij(6,1,1), Keij(6,1,2)],
22   [0,0,0,0, 0, Keij(6,2,1), Keij(6,2,2)]
23 );
24 F:matrix(
25   [Fei(1,1)-v(0,1,1)*Q0],
26   [Fei(1,2)+Fei(2,1)],
27   [Fei(2,2)+Fei(3,1)],
28   [Fei(3,2)+Fei(4,1)],
29   [Fei(4,2)+Fei(5,1)],
30   [Fei(5,2)+Fei(6,1)],
31   [Fei(6,2)+v(1,6,2)*Q3]
32 );
33
34 c0:g0;
35 Q3:h0;
36
37 A:matrix(
38   [Keij(1,2,2)+Keij(2,1,1), Keij(2,1,2), 0,0,0,0],
39   [Keij(2,2,1), Keij(2,2,2)+Keij(3,1,1), Keij(3,1,2),0,0,0],
40   [0, Keij(3,2,1), Keij(3,2,2)+Keij(4,1,1), Keij(4,1,2),0,0],
41   [0,0, Keij(4,2,1), Keij(4,2,2)+Keij(5,1,1), Keij(5,1,2),0],
42   [0,0,0, Keij(5,2,1), Keij(5,2,2)+Keij(6,1,1), Keij(6,1,2)],
43   [0,0,0, 0, Keij(6,2,1), Keij(6,2,2)]
44 );
45 F:matrix(
46   [Fei(1,2)+Fei(2,1) - Keij(1,2,1)*c0],
47   [Fei(2,2)+Fei(3,1)],
48   [Fei(3,2)+Fei(4,1)],
49   [Fei(4,2)+Fei(5,1)],
50   [Fei(5,2)+Fei(6,1)],
51   [Fei(6,2)+v(1,6,2)*Q3]
52 );
53
54 res:invert(A).F;
55 c1:res[1][1]; c2:res[2][1];
56 c3:res[3][1]; c4:res[4][1];
57 c5:res[5][1]; c6:res[6][1];
58
59 fun(x):=
60   if x < 1 * h then (c0*'v(x,1,1)+c1*'v(x,1,2)) else (
61     if x < 2 * h then (c1*'v(x,2,1)+c2*'v(x,2,2)) else (
62       if x < 3 * h then (c2*'v(x,3,1)+c3*'v(x,3,2)) else (
63         if x < 4 * h then (c3*'v(x,4,1)+c4*'v(x,4,2)) else (
64           if x < 5 * h then (c4*'v(x,5,1)+c5*'v(x,5,2)) else
65             (c5*'v(x,6,1)+c6*'v(x,6,2)))));
66
67 D:zeromatrix(101,2);
68 for k:0 thru 100 do (D[k+1][1]:float(k * 1/100), D[k+1][2]:ev(fun(k * 1/100),nouns)) $
69 outf: openw("femx.dat");
70 write_data(D, outf, space);
71 close(outf);
72
73 D:zeromatrix(11,2);
74 for k:0 thru 10 do (D[k+1][1]:float(k * 1/10), D[k+1][2]:ev(fun(k * 1/10),nouns)) $
75 outf: openw("femx_tabel.dat");
76 write_data(D, outf, space);
77 close(outf);

```