

EÖTVÖS LORÁND TUDOMÁNYEGYETEM  
Természettudományi Kar

# A TÖMEGMÁTRIX KÖZELÍTÉSEI

## BSc SZAKDOLGOZAT

**Írta:**

Kovács Tamás  
Matematika BSc  
Matematikai elemző szakirány

**Témavezető:**

Dr. Izsák Ferenc  
Egyetemi Docens  
Alkalmazott Analízis és  
Számításmatematikai Tanszék



**ELTE**  
EÖTVÖS LORÁND  
TUDOMÁNYEGYETEM

Budapest, 2024

## **KÖSZÖNET NYILVÁNÍTÁS**

Szeretnék köszönetet mondani mindenkinek, akitől segítséget kaptam ennek a dolgozatnak a megírásához, különösen témavezetőmnek Dr. Izsák Ferencnek.

# Tartalomjegyzék

<b>1. Bevezetés</b>	<b>5</b>
<b>2. Matematikai bevezetés</b>	<b>6</b>
2.1. Tömegmátrixszal kapcsolatos alapfogalmak . . . . .	6
2.2. Mátrixok összehasonlításához használt fogalmak . . . . .	7
2.3. Kvadratúrák bevezetése . . . . .	8
<b>3. Eredmények</b>	<b>9</b>
3.1. Tömegmátrix elsőfokú bázisfüggvények esetén . . . . .	9
3.2. Tömegmátrix magasabb fokú bázisfüggvények esetén . . . . .	11
3.3. Tömegmátrix két dimenzióban . . . . .	12
3.3.1. Elsőfokú bázisfüggvények két dimenzióban . . . . .	12
3.4. Általánosítás tetszőleges háromszögrácsra baricentrikus koordinátákkal . . . . .	14
3.4.1. Baricentrikus koordináták bevezetése . . . . .	14
3.4.2. Bázisfüggvények felírása baricentrikus koordinátákkal . . . . .	15
3.4.3. Tömegmátrix kiszámolása baricentrikus koordinátákkal . . . . .	16
3.4.4. Konkrét tömegmátrix kiszámítása két dimenzióban . . . . .	18
3.5. Összegzéssel való közelítés . . . . .	18
3.6. Legjobban közelítő diagonális mátrix keresése . . . . .	20
3.6.1. Alapeset . . . . .	20
3.6.2. Kiterjesztés több változótól függő mátrixokra . . . . .	20
3.6.3. $T$ elemeinek optimális kiválasztása . . . . .	21
3.6.4. Kísérlet $k = 1$ esetben . . . . .	21
3.6.5. Kísérlet kétdimenziós esetben . . . . .	22
3.7. Diagonális $M$ előállítása . . . . .	23
3.7.1. Egy kvadratúra pontjainak naiv megközelítése . . . . .	23
3.7.2. Megoldás Gauss–Lobatto-kvadratúrával egydimenziós esetben . . . . .	24

3.7.3.	Spektrál elem módszer . . . . .	28
3.8.	Tömegmátrix közelítése háromszögrácson lineáris bázisfüggvényekkel . . . . .	29
3.9.	Gauss–Lobatto-kvadrátúra háromszögrácson . . . . .	30
<b>4.</b>	<b>Összefoglalás</b>	<b>31</b>
<b>5.</b>	<b>Függelék</b>	<b>32</b>
5.1.	Segédfüggvények . . . . .	32
5.1.1.	harom_diag_matrix . . . . .	32
5.1.2.	tomegmatrix_2d_linar . . . . .	32
5.2.	Összegzéssel való közelítés kódjai . . . . .	33
5.2.1.	osszegzessel_valo_kozelites . . . . .	33
5.2.2.	main_osszegzessel_valo_kozelites . . . . .	33
5.2.3.	main_osszegzessel_valo_kozelites_2d . . . . .	33
5.3.	Legjobban közelítő diagonális mátrix keresésének kódjai . . . . .	34
5.3.1.	kulonbseg_normajat_szamolo . . . . .	34
5.3.2.	T_optimalizalo_linaris . . . . .	34
5.3.3.	main_T_optimalizalas . . . . .	35
5.4.	kulonbseg_normajat_szamolo_2d . . . . .	35
5.4.1.	T_optimalizalo_linaris_2d . . . . .	35
5.4.2.	main_T_optimalizalas_2d . . . . .	36
5.5.	A trapézformulával kapott eredmény vizsgálatának a kódja . . . . .	36

# 1. Bevezetés

A tömegmátrix fogalmát a végeselem-módszer kapcsán definiáljuk. A végeselem-módszer egy numerikus közelítést ad parciális differenciálegyenletek megoldására. Ezt eredetileg mérnöki feladatok numerikus megoldásaira (például tartószerkezeteknél a rúderő és a torzió számolására) vezették be.

Ha például egy parabolikus feladat megoldásának közelítését  $\sum_{j=1}^{N_h} u_j(t)\phi_j(x)$  alakban keressük, akkor

$$\sum_{j=1}^{N_h} \dot{u}_j(t) \int_{\Omega} \phi_j \phi_i d\Omega + \sum_{j=1}^{N_h} u_j(t) a(\phi_j, \phi_i) = \int_{\Omega} f(t) \phi_i d\Omega, i = 1, 2, \dots, N_h.$$

Itt  $\phi_i$ -k adott bázisfüggvények,  $u_j(t)$ -k az ismeretlen együtthatófüggvények. Ez az egyenlet felírható mátrixokkal a következő módon.

- Készítünk egy  $\mathbf{u}$  vektort a következő módon  $\mathbf{u} = (u_1(t), u_2(t), \dots, u_{N_h}(t))^T$ .
- Az  $\dot{u}_j$ -hez tartozó  $\int_{\Omega} \phi_j \phi_i d\Omega$  integrálok adják az  $M_{i,j}$  elemeket.
- Az  $u_j$ -hez tartozó  $a(\phi_j, \phi_i)$ -k fogják megadni  $A_{i,j}$  elemeit.

Ezen jelöléseket használva kapjuk, hogy

$$\mathbf{M}\dot{\mathbf{u}}(t) + \mathbf{A}\mathbf{u}(t) = \mathbf{f}(t)$$

Itt  $\mathbf{A}$ -t merevségi mátrixnak hívják és  $\mathbf{M}$ -et tömegmátrixnak. Látható, hogy az egyenlet megoldásához kéne tudnunk invertálni  $\mathbf{M}$ -et, vagy minden időlépésben ezzel egy egyenletet megoldani. Ez a gyakorlatban nem mindig lehetséges, mert ha már  $10^6$  nagyságrendű mátrixunk van, azt nagyon időigényes lenne invertálni. Ezért helyettesíteni szeretnénk ezt a tömegmátrixot egy hozzá hasonló diagonális mátrixszal, így könnyen és gyorsan invertálható lesz. A hullámegyenletek numerikus megoldásánál szinte mindig ezt az elvet követik [8].

Ebben a dolgozatban az ilyen közelített tömegmátrixok létrehozásának módszereit fogom vizsgálni. Összehasonlítom majd ezeket annak alapján, hogy milyen pontosak és hogy milyen gyorsak.

Először azt fogom felírni, hogyan néznek ki a bázisfüggvények egy és két dimenzióban. Majd három közelítési módszert fogok bemutatni. A sorösszegzést, az eredeti tömegmátrixhoz legközelebb álló diagonális mátrix megkeresését és a tömegmátrix előállítását olyan módon, hogy az eleve diagonális legyen.

## 2. Matematikai bevezetés

Itt fogom összegyűjteni azokat a definíciókat és állításokat, amiket a dolgozat későbbi részeiben fel fogok használni.

### 2.1. Tömegmátrixszal kapcsolatos alapfogalmak

**2.1. Definíció** (Lagrange-interpoláció). A Lagrange-interpolációval az  $f$  függvényt közelítjük egy intervallumon polinommal. Az intervallumon felvett  $\{x_1, x_2, \dots, x_{n+1}\}$  pontokhoz tartozik egy-egy  $f(x_i)$  függvényérték, és olyan  $n$ -ed fokú polinomot konstruálunk, amely ugyanezt az értéket veszi fel minden egyes pontban. Ez a következő alakban állítható elő:

$$p(x) = \sum_{i=1}^n f(x_i) L_i(x)$$

ahol  $L_i$  az  $x_i$ -hez tartozó Lagrange-polinom

$$L_i(x) = \prod_{j=1, j \neq i}^n \frac{x - x_j}{x_i - x_j}$$

Definiálom a Lagrange-típusú bázisfüggvényeket, mert többször ezek fogják felépíteni a tömegmátrixot.

**2.2. Definíció** (Lagrange-típusú bázisfüggvény). Egy  $\Omega$  tartományt felosztunk  $\{x_1, x_2, \dots, x_n\}$  osztópontokra. Ezután minden  $x_i$  osztóponthoz hozzárendelünk egy  $b_i : \Omega \rightarrow \mathbb{R}$  bázisfüggvényt úgy, hogy minden  $i = 1, 2, \dots, n$  esetén a következők teljesüljenek:

- $b_i$  folytonos  $\Omega$ -án.
- $b_i(x_i) = 1$  és  $b_i(x_j) = 0 : \forall j \neq i, x_j \in \{x_1, x_2, \dots, x_n\}$ .
- A  $\{x_j, x_{j+1}, \dots, x_i, \dots, x_{j+k+1}\} \in \{x_1, x_2, \dots, x_n\}$  pontokon végzett Lagrange-interpoláció eredménye  $b_i$ .

Az így létrehozott  $b_j$  bázisfüggvény  $k$ -ad fokú Lagrange-típusú bázisfüggvény.

**2.3. Definíció** (Tömegmátrix). Az  $\mathbf{M}$  tömegmátrix elemeit  $b_i$  bázisfüggvények szorzatának integráljaként definiáljuk az alábbi módon:

$$\mathbf{M}_{j,k} = \int_{\Omega} b_j b_k.$$

A definícióból következik, hogy a tömegmátrix szimmetrikus.

## 2.2. Mátrixok összehasonlításához használt fogalmak

Először a  $p$ -vektornormát vezetem be, mert fel fogom használni egy mátrixnorma indukálásához.

**2.4. Definíció** ( $p$ -vektornorma). Egy  $\mathbf{x} \in \mathbb{R}^n$  vektor  $p$ -normája a következő:

$$\|\mathbf{x}\|_p = \sqrt[p]{\sum_{i=1}^n |x_i|^p}.$$

Ebből a vektornormából indukálhatunk a  $p$ -mátrixnormát.

**2.5. Definíció** ( $p$ -mátrixnorma). Egy tetszőleges  $M \in \mathbb{R}^{n \times n}$  mátrix  $p$ -normáját a  $p$ -vektornorma indukálja:

$$\|M\|_p = \max_{\mathbf{x} \neq 0} \frac{\|M\mathbf{x}\|_p}{\|\mathbf{x}\|_p}$$

Ezek a normák közül a  $p = 2$  speciális esetet fogom vizsgálni és felhasználni, ennek a normának a neve a spektrálnorma.

Ehhez és a következő definícióhoz is használom a  $\lambda_j(M)$  jelölést az  $M$  mátrix  $j$ -edik, valamint a  $\lambda_{\max}(M)$  jelölést a legnagyobb sajátértékére.

**2.6. Definíció** (spektrálnorma). Egy  $M$  szimmetrikus mátrix 2-es normáját spektrálnormának is hívják és az alábbival egyenlő:

$$\|M\|_2 = \max_{\mathbf{x} \neq 0} \frac{\|M\mathbf{x}\|_2}{\|\mathbf{x}\|_2} = \sqrt{\lambda_{\max}(M^*M)} = \lambda_{\max}(M).$$

Spektrálnormát nagyon könnyű számolni nemnegatív elemű diagonális mátrixban, mert a legnagyobb abszolút értékű eleme a spektrálnormája.

Bevezetek egy másik mátrixnormát is, így össze tudom hasonlítani a két norma által adott eredményt.

**2.7. Definíció** (Frobénusz-norma). Az  $M$  mátrix Frobénusz-normája a következő:

$$\|M\|_F = \sqrt{\sum_{i,j} (M_{i,j})^2} = \sqrt{\text{tr}(M^*M)} = \sqrt{\sum_{i=1}^n \lambda_i^2(M)}.$$

Ezeknek a normáknak az ekvivalenciájáról szól a következő tétel.

**2.1. Tétel** (mátrixnormák ekvivalenciája). Véges dimenziós vektortérben minden  $\|\cdot\|_a$  és  $\|\cdot\|_b$  mátrixnormára léteznek olyan pozitív konstansok, hogy

$$c_1 \|A\|_a \leq \|A\|_b \leq c_2 \|A\|_a.$$

Így a két mátrixnorma közötti különbség legfeljebb egy konstans tényező, ezért mindegy, melyik mátrixnormát használjuk.

Az előbbi tétel miatt elméletben nem számít, hogy melyik normát használjuk fel, de gyakorlatban majd látható lesz, hogy a megfelelő mátrixnorma megválasztása nagyon fontos lehet.

## 2.3. Kvadraturák bevezetése

**2.8. Definíció** (Legendre-polinomok). *A következő egyenlőséggel adott*

$$P_n(x) = \sum_{k=0}^n \binom{n}{k} \binom{n+k}{k} \left(\frac{x-1}{2}\right)^k$$

*n-ed fokú polinomot n-edik Legendre-polinomnak nevezünk.*

**2.1. Állítás.** *A Legendre-polinomok egy ortogonális függvényrendszert alkotnak, minden gyökük valós és a  $[-1, 1]$  intervallumba esik.*

A Legendre-polinomok szükségesek a Gauss-kvadraturák alappontjainak és súlyainak kiszámolásához.

**2.9. Definíció** (Gauss-kvadratura). *A Gauss-kvadratura egy olyan kvadratura, ahol minden  $x_i$  osztópontot és  $\omega_i$  súlyok megfelelő megválasztásával  $2n-1$ -ed fokú polinom integrálját pontosan közelíthetjük.*

$$\int_a^b f(x) dx \approx \sum_{i=1}^n \omega_i f(x_i).$$

Bevezetem Kronecker-delta függvényt, amit fel fogok használni majd a többdimenziós esetben.

**2.10. Definíció** (Kronecker-delta függvény). *Ez a függvény 1, ha  $i$  és  $j$  egyenlők, különben 0, azaz*

$$\delta : \mathbb{R}^2 \rightarrow \{0, 1\}, \delta(i, j) = \begin{cases} 1, & \text{ha } i = j \\ 0, & \text{ha } i \neq j. \end{cases}$$

Másik jelölése a  $\delta_{ij}$ . A továbbiakban ezt fogom használni.



### 3. Eredmények

#### 3.1. Tömegmátrix elsőfokú bázisfüggvények esetén

Az egyik legegyszerűbb példa ilyen mátrixra úgy kapható, hogy a bázisfüggvények csak szakaszonként lineáris elemekből állnak.

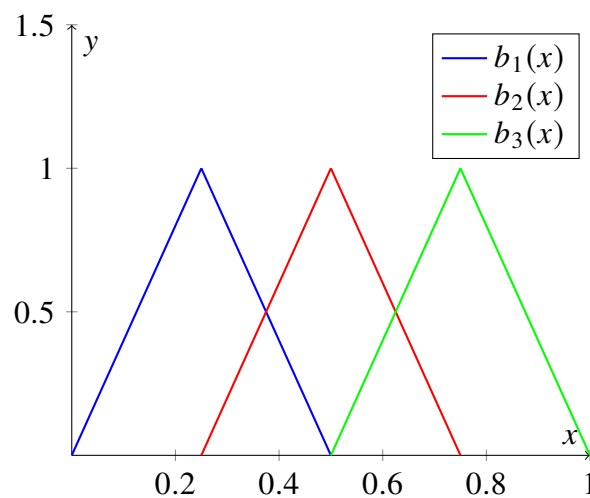
Példaként tekintsük a  $[0, 1]$  intervallumon egyenletes elosztással felvett  $n$  db osztópontot  $h$  távolságra egymástól. Minden osztópontozhoz rendelünk egy  $b_i$  bázisfüggvényt.

Egy ilyen  $b_i$  bázisfüggvény az  $x_{i-1}$  pontban 0,  $x_{i-1}$  és  $x_i$  között lineárisan nő,  $x_i$ -ben 1,  $x_i$  és  $x_{i+1}$  között lineárisan csökken és  $x_{i+1}$ -ben 0, minden más helyen 0. Ennek formális felírása a következőképpen néz ki:

$$b_i(x) = \begin{cases} 0 & : x < x_{i-1} \\ 1 + \frac{x-x_i}{h} & : x_{i-1} \leq x < x_i \\ 1 & : x = x_i \\ 1 - \frac{x-x_i}{h} & : x_i < x \leq x_{i+1} \\ 0 & : x > x_{i+1} \end{cases}$$

Mivel egyenletesen  $h$  távolságra vannak egymástól az  $x_i$  pontok, ezért  $x_{i+1} = x_i + h$ , így az előző függvény megadásában  $x_{i-1}$ -et és  $x_{i+1}$ -et ki lehet fejezni csak  $x_i$ -vel és  $h$ -val.

Ezek a bázisfüggvények ilyen egyszerű alakú „sátorfüggvények” lesznek:



1. ábra. Lineáris bázisfüggvények.

Ezekből a bázisfüggvényekből ki tudjuk számolni a hozzájuk tartozó tömegmátrixot. Az  $M$  tömegmátrix  $(i, j)$ -edik elemét a következőképpen számolom ki:

$$M_{i,j} = \int_0^1 b_i(x) * b_j(x) dx.$$

Néhány különböző esetre lehet szétválasztani az  $M$  mátrixhoz szükséges integrálásokat.

- Ha  $i = j$ , vagyis az elem a mátrix diagonálisán van, akkor

$$\begin{aligned} & \int_0^1 b_i(x) * b_i(x) dx = \\ & = \int_0^{x_{i-1}} b_i(x) * b_i(x) dx + \int_{x_{i-1}}^{x_i} b_i(x) * b_i(x) dx + \\ & + \int_{x_i}^{x_{i+1}} b_i(x) * b_i(x) dx + \int_{x_{i+1}}^1 b_i(x) * b_i(x) dx, \end{aligned}$$

amelybe könnyen be lehet helyettesíteni  $b_i(x)$  definíciója alapján. Azaz kapjuk, hogy

$$\begin{aligned} & \int_0^1 b_i(x) * b_i(x) dx = \int_0^{x_{i-1}} 0 * 0 dx + \int_{x_{i-1}}^{x_i} \left(1 + \frac{x - x_i}{h}\right) * \left(1 + \frac{x - x_i}{h}\right) dx + \\ & + \int_{x_i}^{x_{i+1}} \left(1 - \frac{x - x_i}{h}\right) * \left(1 - \frac{x - x_i}{h}\right) dx + \int_{x_{i+1}}^1 0 * 0 dx = \\ & = 0 + \int_{x_{i-1}}^{x_i} \frac{x^2}{h^2} + 2x\left(\frac{1}{h} - \frac{x_i}{h^2}\right) + \left(\frac{x_i^2}{h^2} - \frac{2x_i}{h} + 1\right) dx + \int_{x_i}^{x_{i+1}} \frac{x^2}{h^2} - 2x\left(\frac{1}{h} + \frac{x_i}{h^2}\right) + \left(\frac{x_i^2}{h^2} + \frac{2x_i}{h} + 1\right) dx + 0 = \\ & felhasználva, hogy  $x_{i+1} = x_i + h$  \\ & = \left[\frac{x^3}{3h^2} + x^2\left(\frac{1}{h} - \frac{x_i}{h^2}\right) + x\left(\frac{x_i^2}{h^2} - \frac{2x_i}{h} + 1\right)\right]_{x_{i-1}}^{x_i} + \left[\frac{x^3}{3h^2} - x^2\left(\frac{1}{h} + \frac{x_i}{h^2}\right) + x\left(\frac{x_i^2}{h^2} + \frac{2x_i}{h} + 1\right)\right]_{x_i}^{x_i+h} = \\ & = \frac{h}{3} + \frac{h}{3} = \frac{2}{3} * h \end{aligned}$$

Így kapjuk, hogy az  $M$  mátrix diagonális elemeinek értéke mind  $\frac{2h}{3}$ .

- Ha  $|i - j| = 1$ , akkor  $x_i$  és  $x_j$  szomszédosak. Ebben az esetben hasonlóan számoljuk az előző esethez, viszont itt csak egy olyan intervallum lesz, ahol az integrál nem nulla. Nézzük a  $j = i - 1$  esetet, ebben az esetben ez az intervallum  $[x_{i-1}, x_i]$ .

$$\begin{aligned} & \int_0^1 b_i(x) * b_{i-1}(x) dx = \int_{x_{i-1}}^{x_i} b_i(x) * b_{i-1}(x) dx = \\ & \int_{x_{i-1}}^{x_i} \left(1 + \frac{x - x_i}{h}\right) * \left(1 - \frac{x - x_{i-1}}{h}\right) dx = \end{aligned}$$

Itt ismét kihasználom azt, hogy  $x_{i-1} = x_i - h$

$$\begin{aligned} & \int_0^1 b_i(x) * b_{i-1}(x) dx = \int_{x_i-h}^{x_i} \left(1 + \frac{x - x_i}{h}\right) * \left(1 - \frac{x - x_i + h}{h}\right) dx = \\ & \int_{x_i-h}^{x_i} \frac{x^2}{h^2} + x\left(\frac{2x_i}{h^2} - \frac{1}{h}\right) + \left(\frac{x_i^2}{h^2} - \frac{x_i^2}{h^2}\right) dx = \left[-\frac{x^3}{3h^2} + x^2\left(\frac{2x_i}{2h^2} - \frac{1}{2h}\right) + x\left(\frac{x_i}{h} - \frac{x_i^2}{h^2}\right)\right]_{x_i-h}^{x_i} = \\ & = \frac{h}{6} \end{aligned}$$

Mivel a  $b_i$  bázisfüggvények szimmetrikusak az  $x_i$  pontra, ezért  $j = i + 1$  esetén az előző eredményt felhasználva:

$$\int_0^1 b_i(x) * b_{j=i+1}(x) dx = \int_0^1 b_{j=i+1}(x) * b_{j-1=i}(x) dx = \frac{h}{6}.$$

- Ha nincs olyan  $x$ , hogy  $b_i(x) \neq 0$  és  $b_j(x) \neq 0$ , akkor a két függvény szorzata 0, így az integráljuk is 0. Ez az eset akkor áll fenn, ha  $|i - j| \geq 2$ .

Mivel az  $M$  tömegmátrix minden eleme tartalmazza a  $h$ -t vagy 0, ezért kiemelek  $h$ -t a mátrixból. Ezután a tömegmátrix így fog kinézni:

$$M = \begin{bmatrix} \frac{2}{3} & \frac{1}{6} & 0 & \cdots & 0 \\ \frac{1}{6} & \frac{2}{3} & \frac{1}{6} & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \frac{1}{6} & \frac{2}{3} & \frac{1}{6} \\ 0 & \cdots & 0 & \frac{1}{6} & \frac{2}{3} \end{bmatrix} * h$$

### 3.2. Tömegmátrix magasabb fokú bázisfüggvények esetén

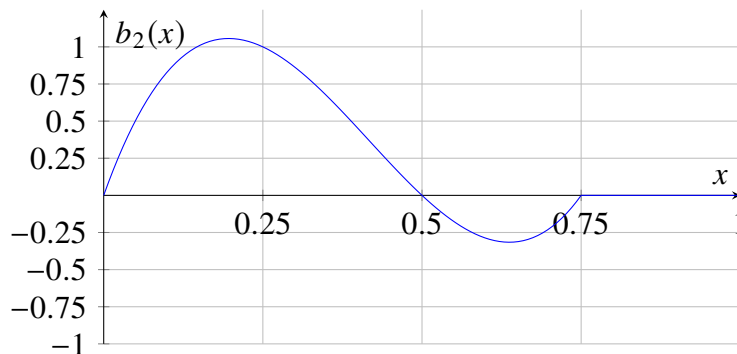
A magasabb fokú  $b_i$  bázisfüggvényeket interpolációval számolhatjuk ki.

Lagrange-interpolációval elkészíthetünk egy folytonos  $b_i$ -t. Ekkor az alappontok legyenek az  $x_i$  és mellette lévő  $k$  db pont  $x_a$  ponttal kezdve, így  $k$ -ad fokú bázisfüggvényeket kapunk. Mivel  $b_i(x_j) = 0$  minden  $j \neq i$  esetén, ezért csak az  $x_i$ -hez tartozó Lagrange-polinomot kell néznünk. [7]

$$b_i(x) = \prod_{j=a, j \neq i}^{a+k+1} \frac{x - x_j}{x_i - x_j}$$

Példa egy ilyen bázisfüggvényre  $k = 3$  esetben. Az alappontok legyenek  $x_1 = 0$ ,  $x_2 = 0.25$ ,  $x_3 = 0.5$ ,  $x_4 = 0.75$ ,  $x_5 = 1$ . Ekkor  $b_2$ -t felírhatjuk a következőképpen:

$$b_2(x) = \frac{x - x_1}{x_2 - x_1} \frac{x - x_3}{x_2 - x_3} \frac{x - x_4}{x_2 - x_4} = \frac{x}{0.25} \frac{x - 0.5}{-0.25} \frac{x - 0.75}{-0.5} = 32x^3 - 40x^2 + 12x$$



Egy érdekes megfigyelés, hogy  $b_2$  maximuma nagyobb, mint 1.

### 3.3. Tömegmátrix két dimenzióban

Ha  $\Omega \in \mathbb{R}^2$ , akkor  $\Omega$  síkját szakaszok helyett véges sok kisebb síkidomra osztjuk fel. Ezeket a síkidomokat jelöljük  $S_k$ -val. Legyen  $N_\Omega$  ezeknek a száma. Ezek az  $S_k$ -kat úgy hozzuk létre, hogy az alábbi állítások igazak legyenek, ha  $i \neq j$ :

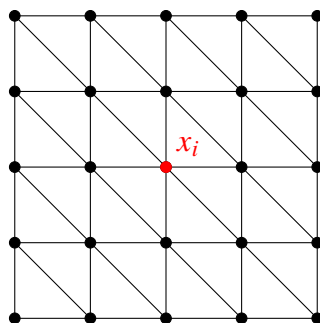
- Minden  $i, j$  párosra pontosan egy állítás igaz:

- $S_i \cap S_j$  pontosan egy csúcs.
- $S_i \cap S_j$  pontosan egy teljes oldal.
- $S_i \cap S_j$  üres.

- $S_i \neq \emptyset$

- $\bigcup_{k=1}^{N_\Omega} S_k = \Omega$

Az  $S_k$ -k csúcsaiban lesznek az  $x_i$  pontok, amelyekhez tartozni fognak a  $b_i$  bázisfüggvények. A legegyszerűbb példa erre egy négyzet rácsa, aminek az oldalai párhuzamosak  $x$  és  $y$  tengellyel. Ennek egy gyakorlatban hasznosabb változata az, amikor ezt a rácsot háromszögekre osztjuk fel. [4]



Általános esetben ennek a háromszöges hálónak az elemei nem feltétlenül hasonlítanak egymásra és nem egyforma méretűek.

#### 3.3.1. Elsőfokú bázisfüggvények két dimenzióban

Az előző ábrán lévő piros pont legyen  $x_i$  és a hozzá tartozó bázisfüggvény  $b_i$ . A bázisfüggvények alaptulajdonságai itt is igazak, így  $b_i(x_i) = 1$  és  $b_i(x_j) = 0 : \forall j \neq i, j \in \{1, 2, \dots, N_h\}$ . Ebben az esetben  $x_i$ -nek 6 szomszédja van, ezért 6 darab részfüggvényből fog állni. Ezek a részfüggvények egy-egy  $S_k$  háromszögon lesznek értelmezve.

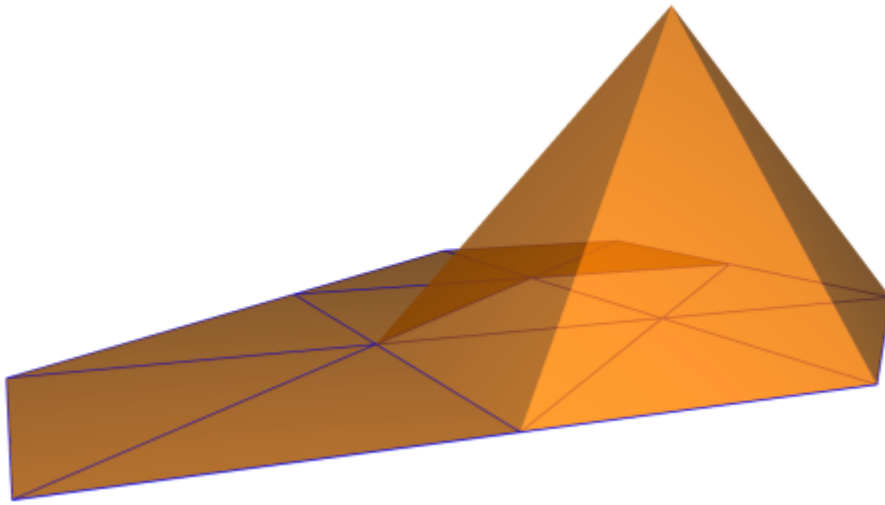
Legyenek  $S_k$  csúcsai  $A_i = (x_i, y_i)$ ,  $A_u = (x_u, y_u)$  és  $A_v = (x_v, y_v)$ . Lagrange-interpolációval ki lehet számolni  $b_i$ -nek az ehhez a háromszöghöz tartozó  $\phi_i^k$  rész-bázisfüggvényét. Mivel  $i \neq j$  pontokban nulla, ezért csak egy Lagrange-féle bázispolinomból fog állni.

$$\phi_i^k(x, y) = \begin{cases} \frac{(x-x_v)(x-x_u)(y-y_v)(y-y_u)}{(x_i-x_v)(x_i-x_u)(y_i-y_v)(y_i-y_u)} & \text{if } (x, y) \in S_k \\ 0 & \text{if } (x, y) \notin S_k \end{cases}$$

Ezekből már definiálható a bázisfüggvény.

**3.1. Definíció** (kétdimenziós bázisfüggvény háromszögrács esetén). Az  $x_i$  ponthoz tartozó  $b_i$  kétdimenziós bázisfüggvényt a háromszögrács  $S_k$  elemein értelmezett  $\phi_i^k$  rész-bázisfüggvények alkotják.

$$b_i(x, y) = \sum_{j=1}^{N_\Omega} \phi_i^j(x, y)$$



2. ábra. Kétdimenziós  $b_i$  bázisfüggvény háromszögrácson. Forrás:[4].

Ha a rácspontokat bal felülről soronként számozzuk, akkor az  $x_i$  szomszédai a következő csúcsok.  $x_{i-n-1}, x_{i-n}, x_{i-n+1}, x_{i-1}, x_{i+1}, x_{i+n-1}, x_{i+n}, x_{i+n+1}$ .  $b_i$  bázisfüggvénynek 6 háromszög oldala van, minden  $\int_\Omega b_i b_j$  számolásnál ennek a kúpnak kettő lapjában metszik egymást.

Az eredményül kapott  $M \in \mathbb{R}^{n^2 \times n^2}$  tömegmátrix máshogy fog kinézni, mint az egydimenziós esetben. Az  $i$ -edik sora a következő:

$$[0 \cdots 0, \int_\Omega b_i b_{i-n-1}, \int_\Omega b_i b_{i-n}, 0 \cdots 0, \int_\Omega b_i b_{i-1}, \int_\Omega b_i b_i, \int_\Omega b_i b_{i+1}, 0 \cdots \\ \cdots \int_\Omega b_i b_{i+n-1}, \int_\Omega b_i b_{i+n}, \int_\Omega b_i b_{i+n+1}, 0 \cdots 0]$$

Ha egy háromszög három csúcsa  $P_1, P_2$  és  $P_3$  akkor ezen a háromszögen belül így néznének ki a bázisfüggvényeik.

$P_1$  pontra:

$$b_1(x, y) = \frac{(x-x_2)(y_3-y) - (x_3-x)(y-y_2)}{(x_2-x_3)(y_1-y_3) - (x_1-x_3)(y_2-y_3)}$$

$P_2$  pontra:

$$b_2(x, y) = \frac{(x - x_3)(y_1 - y) - (x_1 - x)(y - y_3)}{(x_2 - x_3)(y_1 - y_3) - (x_1 - x_3)(y_2 - y_3)}$$

$P_3$  pontra:

$$b_3(x, y) = \frac{(x - x_1)(y_2 - y) - (x_2 - x)(y - y_1)}{(x_2 - x_3)(y_1 - y_3) - (x_1 - x_3)(y_2 - y_3)}$$

### 3.4. Általánosítás tetszőleges háromszögrácsra baricentrikus koordinátákkal

Amikor egy tetszőleges felületet háromszögrácscsal akarunk felosztani (vagy akár csak közelíteni), ebben a rácsban a háromszögek akár mindegyike különbözhet egymástól. Ezért kell egy megoldás arra, hogy bármilyen háromszögeknél ugyanúgy tudjuk számolni a bázisfüggvényeket. Erre fogjuk használni a baricentrikus koordinátákat.

#### 3.4.1. Baricentrikus koordináták bevezetése

Ehhez nézzük egy valódi  $S_k$  háromszög három csúcsát:  $A^1, A^2$  és  $A^3$  -at. Ezekkel felírható bármilyen másik koordináta. De nem csak háromszögre fogom definiálni, hanem szimplexekre általánosan.

**3.2. Definíció** (Baricentrikus koordináta). *Legyenek  $A^1, A^2, \dots, A^{n+1}$  pontok egy  $n$ -dimenziós affin térben és ezek a pontok legyenek affin függetlenek. Ekkor bármelyik  $P$  pont felírható a következő módon:*

$$\overrightarrow{OP} = \lambda_1 \overrightarrow{OA^1} + \lambda_2 \overrightarrow{OA^2} + \dots + \lambda_{n+1} \overrightarrow{OA^{n+1}},$$

ahol a  $\lambda$  súlyok összege 1, azaz  $\sum_{i=1}^{n+1} \lambda_i = 1$ . A fenti egyenlőségben szereplő  $\lambda_i$  együtthatókat nevezzük a  $P$  pont baricentrikus koordinátáinak.

Be kell látnunk, hogy ezek a koordináták nem függenek az  $O$  pont kiválasztásától. Legyen  $O'$  egy  $O$ -tól különböző pont és a  $\lambda_i$  súlyok maradjanak ugyanazok, mint a definícióban. Ekkor  $P'$  legyen a következő pont:

$$\overrightarrow{O'P'} = \sum_{i=1}^{n+1} \lambda_i \overrightarrow{O'A^i}.$$

$$\begin{aligned} \overrightarrow{O'P'} &= \sum_{i=1}^{n+1} \lambda_i (\overrightarrow{O'O} + \overrightarrow{OA^i}) = \sum_{i=1}^{n+1} \lambda_i \overrightarrow{O'O} + \sum_{i=1}^{n+1} \lambda_i \overrightarrow{OA^i} = \sum_{i=1}^{n+1} \lambda_i \overrightarrow{O'O} + \sum_{i=1}^{n+1} \lambda_i \overrightarrow{OA^i} = \\ &= \left( \sum_{i=1}^{n+1} \lambda_i \right) * \overrightarrow{O'O} + \sum_{i=1}^{n+1} \lambda_i \overrightarrow{OA^i} = \overrightarrow{O'O} + \overrightarrow{OP} = \overrightarrow{O'P}, \end{aligned}$$

vagyis valóban  $P' = P$ .

**3.1. Tétel.** *Minden  $n$ -dimenziós  $P$  pontnak létezik egyértelmű baricentrikus koordinátája.*

Ha  $A^j$  koordinátái  $(x_1^j, x_2^j, \dots, x_n^j)$  és  $P$  koordinátái  $(x_1, x_2, \dots, x_{n+1})$ . Ekkor felírhatjuk az alábbi egyenletrendszerrel arra, hogy megkapjuk a koordinátákat.

$$\begin{cases} \lambda_1 + \lambda_2 + \dots + \lambda_{n+1} = 1 \\ \lambda_1 x_1^1 + \lambda_2 x_1^2 + \dots + \lambda_{n+1} x_1^{n+1} = x_1 \\ \lambda_1 x_2^1 + \lambda_2 x_2^2 + \dots + \lambda_{n+1} x_2^{n+1} = x_2 \\ \vdots \\ \lambda_1 x_n^1 + \lambda_2 x_n^2 + \dots + \lambda_{n+1} x_n^{n+1} = x_n \end{cases}$$

Így bármely  $P$  pont baricentrikus koordinátáit ki tudjuk számolni.

Legyen  $\lambda_i(P)$  egy olyan függvény, amely a  $P$  helyen ezen pont  $i$ -edik baricentrikus koordinátáját adja az  $S_k$  háromszög csúcsainak rendszerében. A baricentrikus koordináták definíciójából következik, hogy  $\lambda_i(A^j) = \delta_{ij}$ .

### 3.4.2. Bázisfüggvények felírása baricentrikus koordinátákkal

Egy  $x_i$  ponthoz tartozó  $b_i$  bázisfüggvényt több  $\phi_i^k$  rész-bázisfüggvények összegére bontjuk. Ahol  $\phi_i^k$  egy bázisfüggvény egy olyan  $S_k$  háromszögön, aminek  $x_i$  az egyik csúcsa és mindenhol máshol 0.

Itt az a célom, hogy ezeken a háromszögeken belül meghatározzam a  $\phi_i$  rész-bázisfüggvényeket.

Amikor lineáris bázisfüggvényt keresünk, akkor az  $S_k$  háromszög  $A^i$  csúcshoz tartozó  $\phi_i$  bázisfüggvény a  $\lambda_i$ . Mivel a  $\lambda_1, \lambda_2, \lambda_3$  függvények bázist alkotnak a kétváltozós elsőfokú polinomok terén, ezért a bázisfüggvények felírhatók csak a  $\lambda_j$  függvényekből.

Ha magasabb fokú polinomot szeretnénk bázisfüggvénynek felvenni, akkor a háromszögön belül új képzeletbeli pontokat kell felvennünk ahhoz, hogy meg tudjunk adni egy  $\lambda$ -ákból álló bázist.

Ez másodfokú függvényeknél a következőképpen történik. A következő függvények bázist alkotnak a kétváltozós másodfokú polinomok terén:

$$\lambda_1^2, \lambda_2^2, \lambda_3^2, \lambda_1 \lambda_2, \lambda_1 \lambda_3, \lambda_2 \lambda_3.$$

Ennek a szabadsági foka 6. Ezért a Lagrange-interpolációhoz 6 pont kell. Ezek a pontok lehetnek a háromszög csúcsai és az oldalainak felezőpontjai. Mivel baricentrikus koordinátákat használunk, ezek a koordináták minden háromszögre ugyanazok:  $(\frac{1}{2}, \frac{1}{2}, 0)$ ,  $(\frac{1}{2}, 0, \frac{1}{2})$  és  $(0, \frac{1}{2}, \frac{1}{2})$ .

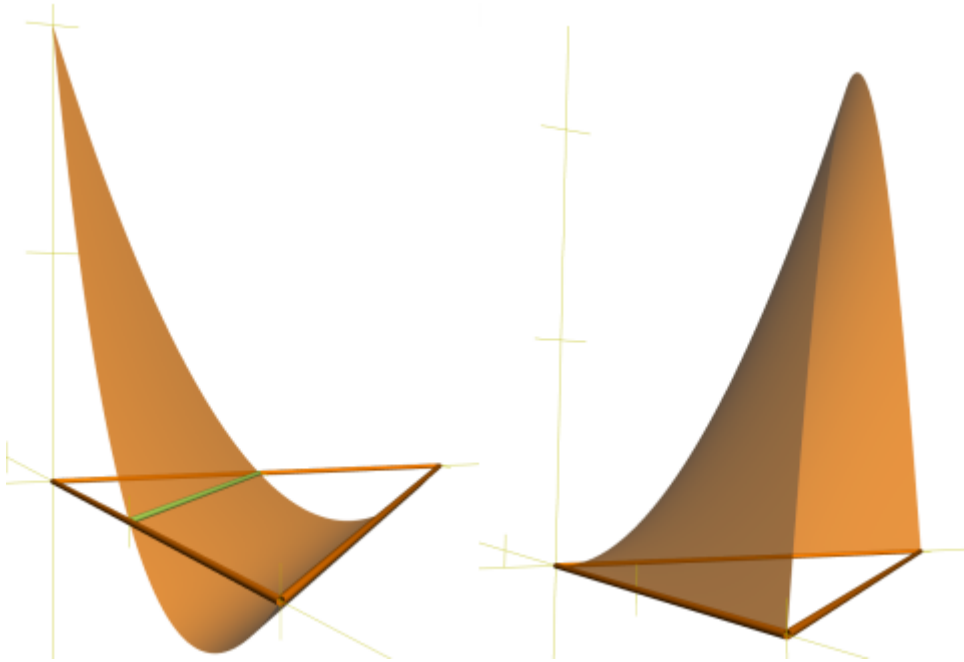
Ezért a bázisfüggvények itt is felírhatók csak a  $\lambda_j$  függvények segítségével.

Most, hogy definiáltuk a bázisfüggvényeket egy háromszögön belül, be kell látnunk, hogy ezeknek az összege is folytonos lesz.

Nézzünk két szomszédos háromszöget  $S_k$ -t és  $S_{k'}$ -t. A közös pontjaik  $A^1$  és  $A^2$ . Hozzájuk



3. ábra. Példa a hozzáadott pontok elhelyezkedéséről  $k = 1, 2, 3$ . Forrás:[3].



4. ábra. Kétdimenziós  $\phi_i$  bázisfüggvény egy háromszög elemen. Forrás:[4].

tartoznak  $\phi$  és  $\phi'$  bázisfüggvények, ahol  $\phi(A^1) = \phi'(A^1)$  és  $\phi(A^2) = \phi'(A^2)$ . A közös élüket felírhatjuk a következő módon  $P = \mu A^1 + (1 - \mu)A^2$ , ahol  $\mu \in [0, 1]$ . Ahhoz, hogy ezen a szakaszon folytonos legyen az egész bázisfüggvény, az alábbi feltételnek kell teljesülnie:  $\phi(P) = \phi'(P)$  minden  $P = \mu A^1 + (1 - \mu)A^2$  re. Mivel  $\phi(P) - \phi'(P) = 0$ -nak a bal oldala egy elsőfokú polinom, amely  $\mu$ -tól függ. Ennek az egyenletnek két megoldását tudjuk biztosan  $P = A^1$ -et, amikor  $\mu = 1$  és a  $P = A^2$ -et, amikor  $\mu = 0$ . Mivel most egy elsőfokú polinomnak két különböző helyen is nulla, ez azt jelenti, hogy a polinom a konstans nulla. Így látszik, hogy  $\phi - \phi' = 0$  az egész szakaszon. Ebből következik hogy  $b_i$  folytonos az  $S_k \cup S_{k'}$  halmazon. [4]

### 3.4.3. Tömegmátrix kiszámolása baricentrikus koordinátákkal

Hozzunk létre egy olyan  $S^R$  referencia-háromszöget, ahol egyszerűbben tudunk számolni. Legyen ez a háromszög az, amelynek csúcsai  $R_1 = (0, 0)$ ,  $R_2 = (1, 0)$  és  $R_3 = (0, 1)$ . Ekkor bármely  $S_k$ -hoz létezik pontosan egy olyan affin leképezés, amelyre  $F_k(R_j) = A_k^j$ .



Ekkor tetszőleges  $P \in S^R$  pont megkapható a következő alakban:

$$F_k(P) = \lambda_1^R(P)A_k^1 + \lambda_2^R(P)A_k^2 + \lambda_3^R(P)A_k^3.$$

Itt a  $\lambda_j^R$  a referencia-háromszöghöz tartozó baricentrikus koordináta. Ez felírható a következő módon:  $\lambda_j^R(P) = \lambda_j(F_k(P))$ . Ezután felírhatjuk a baricentrikus koordinátákat egyszerűen az eredeti koordinátákkal:  $\lambda_1^R = 1 - x_1 - x_2$ ,  $\lambda_2^R = x_1$  és  $\lambda_3^R = x_2$ .

Ekkor egy  $f$  függvény integrálása egy tetszőleges  $S_k$ -n a következőképpen néz ki:

$$\int_{S_k} f = 2T_k \int_0^1 \int_0^{1-\lambda_2} f d\lambda_3^R d\lambda_2^R.$$

Itt  $T_k$  az  $S_k$  háromszög területe. Korábban egydimenziós esetben láttuk, hogy a tömegmátrix meg van szorozva  $h$ -val, ahol  $h$  az intervallumok mérete, ezért  $M$  lineárisan kötődött a felosztásunk méreteihez. Mivel itt most már megjelenik a háromszög területe, így látszik hogy a tömegmátrix már négyzetesen fog függeni a felosztásunk méreteitől.

Ezek alapján ki tudjuk számolni a tömegmátrixot. Elsőként számoljuk ki a nem diagonális tagokat egy háromszögen.

$$\int_{S_k} b_2 b_3 dx = \int_{S_k} \lambda_2 \lambda_3 dx = \int_{S^R} \lambda_2^R \lambda_3^R * J dx^R =$$

Itt  $J$  egy konstans Jacobi-mátrix, amelyek a determinánsa  $2T_k$ .

$$\begin{aligned} &= 2T_k \int_{S^R} x_1^R x_2^R dx^R = 2T_k \int_0^1 \int_0^{1-x_1^R} x_1^R x_2^R dx_2^R dx_1^R = 2T_k \int_0^1 \frac{x_1^R - 2x_1^{R2} + x_1^{R3}}{2} dx_1^R = \\ &= 2T_k \left( \frac{1}{4} - \frac{1}{3} + \frac{1}{8} \right) = \frac{T_k}{12} \end{aligned}$$

Mivel két bázisfüggvény több háromszögben is metszheti egymást, ezeknek a háromszögeknek a halmazát jelöljük  $S^{i,j}$ -vel. Ekkor:

$$M_{ij} = \int_{\Omega} b_i b_j = \sum_k^{S^{i,j}} \frac{T_k}{12}.$$

Ugyanígy kiszámoljuk a főátló elemeinek értékeit.

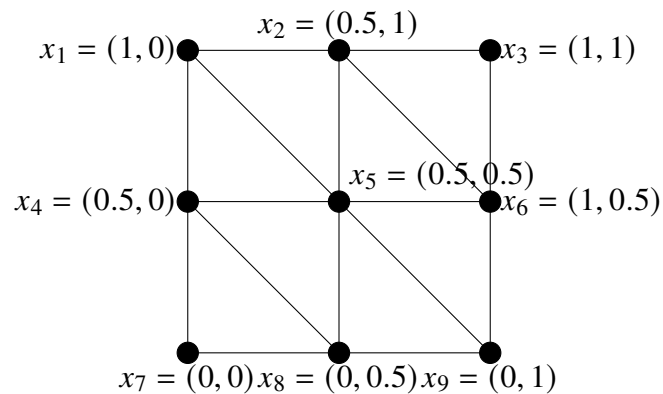
$$\begin{aligned} &\int_{S_k} b_2 b_2 dx = \int_{S_k} \lambda_2 \lambda_2 dx = \int_{S^R} \lambda_2^R \lambda_2^R * J dx^R = \\ &= 2T_k \int_{S^R} x_1^R x_1^R dx^R = 2T_k \int_0^1 \int_0^{1-x_1^R} x_1^R x_1^R dx_2^R dx_1^R = 2T_k \left( \frac{1}{3} - \frac{1}{4} \right) = \frac{T_k}{6}. \end{aligned}$$

Ekkor a tömegmátrix elemei felírhatóak a következő módon:

$$M_{ii} = \int_{\Omega} b_i b_i = \sum_k^{S^{i,i}} \frac{T_k}{6}.$$

### 3.4.4. Konkrét tömegmátrix kiszámítása két dimenzióban

Az alábbi rácson lévő lineáris bázisfüggvények által meghatározott tömegmátrixot fogom kiszámolni az előzőekben elvégzett számítások alapján.



Ekkor  $T_k = \frac{1}{8} = 0.125$

$$M_{2D} = \begin{bmatrix} \frac{2}{6} & \frac{1}{12} & 0 & \frac{1}{12} & \frac{2}{12} & 0 & 0 & 0 & 0 \\ \frac{1}{12} & \frac{1}{6} & \frac{1}{12} & 0 & \frac{2}{12} & \frac{2}{12} & 0 & 0 & 0 \\ 0 & \frac{1}{12} & \frac{1}{6} & 0 & 0 & \frac{1}{12} & 0 & 0 & 0 \\ \frac{1}{12} & 0 & 0 & \frac{3}{6} & \frac{2}{12} & 0 & \frac{1}{12} & \frac{1}{12} & 0 \\ \frac{2}{12} & \frac{2}{12} & 0 & \frac{2}{12} & \frac{6}{6} & \frac{2}{12} & 0 & \frac{2}{12} & \frac{2}{12} \\ 0 & \frac{2}{12} & \frac{1}{12} & 0 & \frac{2}{12} & \frac{3}{6} & 0 & 0 & \frac{1}{12} \\ 0 & 0 & 0 & \frac{1}{12} & 0 & 0 & \frac{1}{6} & \frac{1}{12} & 0 \\ 0 & 0 & 0 & \frac{2}{12} & \frac{2}{12} & 0 & \frac{1}{12} & \frac{2}{6} & \frac{1}{12} \\ 0 & 0 & 0 & 0 & \frac{2}{12} & \frac{1}{12} & 0 & \frac{1}{12} & \frac{2}{6} \end{bmatrix} * \frac{1}{8}$$

Erre a mátrixra igaz a 3.3.1. szakaszban megadott sorképlet.

## 3.5. Összegzéssel való közelítés

Mátrixok közti távolság meghatározására alkalmasan választott mátrixnormákat használunk. Vagyis a  $T \approx M$  mátrix távolsága  $M$ -től  $\|M - T\|$ . Ez nyilván nagyban függ attól, hogy milyen normát használunk. A következőkben én a spektrálnormát és a Frobeniusz-normát fogom használni.

**3.3. Definíció** (Összegzéssel való közelítés). Adott  $M$  tömegmátrixot úgy közelítjük, hogy a  $\tilde{M} \approx M$  mátrix  $i$ -edik sorában a diagonális elem értéke az  $M$   $i$ -edik sorának összege, mindenhol máshol pedig nulla.

Az így kapott  $\tilde{M}$  egy diagonális mátrix, amelyek nincs negatív eleme.

A függvény, amivel ezt számolom (ld. 5.2.1. szakasz) nem a beépített szumma függvényt használja, mert ez az implementáció gyorsabb volt erre az esetre.

Az így kapott  $\tilde{M}$  mátrixot három szempont alapján vizsgálom meg:

- Távolsága az eredeti  $M$  mátrixtól. Ezt a távolságot így számolom ki:  $\|\tilde{M} - M\|$ .
- A két mátrix normájának az aránya:  $\frac{\|\tilde{M}\|}{\|M\|}$ .
- Az előállításukhoz szükséges futtatási időket is összehasonlítom.

Mindhárom vizsgálatot elvégzem Frobeniusz-normával és spektrálnormával is. A vizsgálathoz a 3.1. pontban alkotott  $M$  mátrixot használom. Az ehhez tartozó programkódot az 5.2.2. szakasz tartalmazza.

mátrix mérete	$\ \tilde{M} - M\ _{\text{frob}}$	$\frac{\ \tilde{M}\ _{\text{frob}}}{\ M\ _{\text{frob}}}$	futás idő(mp)
$10^1 \times 10^1$	0.1856	0.4497	0.001296
$10^2 \times 10^2$	0.0620	0.1415	0.001733
$10^3 \times 10^3$	0.0197	0.0447	0.076509
$10^4 \times 10^4$	0.0062	0.0141	13.087047

1. táblázat. Összegzéssel való közelítés elemzése Frobeniusz-normával.

mátrix mérete	$\ \tilde{M} - M\ _2$	$\frac{\ \tilde{M}\ _2}{\ M\ _2}$	futás idő(mp)
$10^1 \times 10^1$	1.0137	0.0968	0.000548
$10^2 \times 10^2$	0.0100	1.0002	0.002720
$10^3 \times 10^3$	0.0010	1	0.240373
$10^4 \times 10^4$	0.0001	1	47.856800

2. táblázat. Összegzéssel való közelítés elemzése spektrálnormával.

Az adatokból látszik, hogy spektrálnormával számolva a közelítés pontosabbnak tűnik. Mivel a  $\frac{\|\tilde{M}\|_2}{\|M\|_2}$  hányados 1-hez nagyon közeli, vagyis a két mátrix egyforma méretű. Az új mátrix távolsága az eredetitől sokkal kisebb a spektrálnorma esetén. De spektrálnormát számolni 3-szor annyi ideig tartott kiszámolni.

Ugyanezt a közelítést elvégeztem a 3.4.4. szakaszban legyártott  $M_{2D}$  mátrixra is.

norma típusa	mátrix mérete	$\ \tilde{M} - M\ _2$	$\frac{\ \tilde{M}\ _2}{\ M\ _2}$	futás idő(mp)
Frobénius	$9 \times 9$	0.1549	0.3568	0.001944
spektrál	$9 \times 9$	0.1240	0.4596	0.001431

3. táblázat. Összegéssel való közelítés elemzése kétdimenziós lineáris bázisfüggvények esetén.

### 3.6. Legjobban közelítő diagonális mátrix keresése

Egy másik hozzáállás az, amikor az eredeti  $M \in \mathbb{R}^{n \times n}$  mátrixból nem egy adott képlet segítségével számoljuk ki az eredményt, hanem egy hozzá „közeli” másik mátrixot választunk.

#### 3.6.1. Alapeset

Ez a feladat egy optimalizálási feladattá alakul át, ahol  $T$  mátrix megfelelő megválasztásával a következőt kell meghatározni:  $\min_T (\|T - M\|)$ , ahol  $T$  diagonális.

Először keressünk egy olyan típusú  $T$  mátrixot, amelynek minden diagonális eleme ugyanaz. Jelölje ezt az elemet  $a$ . Ekkor a feladat sokkal egyszerűbb lesz, mert így csak egyetlen  $a \in \mathbb{R}$  változó van, amelyre nézve optimalizálni kell:

$$\min_a (\|T - M\|).$$

#### 3.6.2. Kiterjesztés több változótól függő mátrixokra

Ezt lehet javítani azzal, hogy több különböző értéket is megengedünk a  $T$  mátrixban. A változók száma attól függ, hogy hányad fokú bázisfüggvényeket használtunk az eredeti tömegmátrix létrehozásánál, mert a  $k$ -ad fokú bázisfüggvények által alkotott  $M$ -ben a főátló egy oldalán  $k$  db átlóban is szerepelnek nem nulla értékek. A tömegmátrix definíciójából tudjuk, hogy az szimmetrikus.

Tegyük fel, hogy  $n \geq 2k + 1$ . Ez a feltétel azért kell, hogy mind a  $k + 1$  darab különböző sor benne legyen a mátrixban. Ha ez nem áll fenn, akkor  $\lceil \frac{n}{2} \rceil \leq k + 1$ , ezért az  $M$  mátrixnak ebben az esetben nincs minden lehetséges sortípusa. Ekkor  $k + 1$  helyett az  $\lceil \frac{n}{2} \rceil$  értéket kellene használni mindenhol.

Mivel egy diagonális tömegmátrix összes eleme nemnegatív, ezért  $p_i \geq 0 \forall i \in \{1, 2, \dots, k + 1\}$ .

Ha az előző feltételek teljesülnek, akkor  $T$  a következőképpen néz ki:

$$T = \begin{bmatrix} p_1 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 \\ 0 & p_2 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 \\ 0 & 0 & \ddots & 0 & \cdots & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & p_{k+1} & \cdots & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & p_{k+1} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \cdots & 0 & \ddots & 0 & 0 \\ 0 & 0 & 0 & 0 & \cdots & 0 & 0 & p_2 & 0 \\ 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & p_1 \end{bmatrix}$$

A közelítő diagonális mátrix.

### 3.6.3. $T$ elemeinek optimális kiválasztása

Most, hogy megvan  $T$  felépítése, ami  $k + 1$  darab változótól függ, ezt felhasználva fel tudjuk írni az optimalizálási feladatot:

$$\min_{p_1, p_2, \dots, p_{k+1}} (\|\mathbf{T} - \mathbf{M}\|).$$

### 3.6.4. Kísérlet $k = 1$ esetben

A kísérlethez alapértéknek veszem azt az  $\mathbf{M}$  tömegmátrixot, amit a 3.1. szakaszban létrehoztam. Ennek a mátrixnak a generálására szolgál az 5.1.1. pontban megadott függvény. Ebben az esetben  $T$  csak  $p_1$  és  $p_2$  változóktól függ. A függvény, aminek az eredményét minimalizálni szeretnénk, a  $\|\mathbf{T} - \mathbf{M}\|$ , ennek az implementációja: 5.3.1. szakaszban található.

Ennek a két függvénynek a felhasználásával írtam meg a  $T$  mátrixot optimalizáló függvényt, amely argumentumnak megkapja, hogy mekkora mátrixszal dolgozzon és azt, hogy milyen normát használjon fel. Ezen Matlab-függvény kódjának (ld. 5.3.2. pont) 5. sorában nem csak a "kulonbseg\_normajat\_szamolo" függvény eredményét próbálom optimalizálni, hanem hozzá kellett adnom ezt a két tagot, hogy biztosítsák a  $p_i$  értékek nemnegatív tulajdonságát:

$$(|p_i| - p_i) * 10^n.$$

Ezt azért kellett megtenni, mert a Matlabba beépített "fminsearch" függvénynek nincs olyan paramétere, amivel be lehetne állítani, hogy  $p_1$  és  $p_2$  pozitívak legyenek. Ugyan létezik az "fmincon" függvény, amelyet épp erre a célra írtak, és ott lehet alsó határt adni a változóknak, de sajnos nincs jól optimalizálva arra az esetre, amikor hosszabb számításokat kell végeznie. Amikor ezt használtam, a  $10^3 \times 10^3$  méretű mátrixokra ugyanolyan eredményeket adott, mint az "fminsearch", de  $10^4 \times 10^4$  méretű mátrixoknál már a minimum értéket 0.9999-nek mondta, ami nyilván nem egy jó eredmény.

A korábbi, összegzéssel való közelítés módszerének (ld. 3.5. szakaszban) eredményét itt felhasználtam a  $p_1$  és  $p_2$  kezdőértékeként, így egy potenciális megoldástól indul el és próbál nála jobbat találni.

mátrix mérete	$(p_1, p_2)$	minimum érték	futásidő(mp)
$10^1$	$(8.4 * 10^{-3}, 1 * 10^{-2})$	$2.001 * 10^{-1}$	0.017266
$10^2$	$(1.0348 * 10^{-5}, 8.6334 * 10^{-5})$	$7.00 * 10^{-2}$	0.005520
$10^3$	$(2.7362 * 10^{-5}, 1.2910 * 10^{-6})$	$2.24 * 10^{-2}$	0.107205
$10^4$	$(6.8050 * 10^{-6}, 5.5386 * 10^{-8})$	$7.1 * 10^{-3}$	63.433361

4. táblázat. A kísérlet eredménye Frobeniusz-normával.

mátrix mérete	$(p_1, p_2)$	minimum érték	futás idő(mp)
$10^1$	$(7.1 * 10^{-3}, 7.9 * 10^{-3})$	$9.47 * 10^{-2}$	0.009258
$10^2$	$(1.4038 * 10^{-4}, 2.9184 * 10^{-5})$	$1 * 10^{-2}$	0.061824
$10^3$	$(2.6444 * 10^{-6}, 7.8892 * 10^{-7})$	$9.9999 * 10^{-4}$	10.597412
$10^4$	$(2.7371 * 10^{-7}, 2.8346 * 10^{-8})$	$1.8775 * 10^{-4}$	1532.163519

5. táblázat. A kísérlet eredménye spektrálnormával.

Ezekből az adatokból több következtetés is levonható. Az első ilyen következtetés az, hogy spektrálnormát számolni nagy mátrixoknál sokkal időigényesebb, mint Frobeniusz-normát, így a jelenlegi állapotban  $10^5$ -es nagyságrendű mátrixra már nem futna le belátható időn belül. A másik következtetésem az, hogy a spektrálnormánál könnyebb jó  $T$ -ket találni, mivel a minimum érték itt mindig kisebb volt, mint a Frobeniusz-normánál. Itt látszik az, hogy ugyan elméletileg a mátrixnormák ekvivalenciájának tétele alapján mindegy, hogy melyik normát választjuk, de a gyakorlatban egyéb szempontok szerint vannak sokkal kedvezőbb mátrixnormák.

### 3.6.5. Kísérlet kétdimenziós esetben

A 3.4.4. szakaszban létrehozott mátrixhoz szeretném megtalálni a hozzá legközelebb álló diagonális mátrixot. A mátrix első 5 sora különbözik egymástól, ezért a  $k = 4$  esetet próbálom majd optimalizálni.

Az optimalizálás kiindulópontjait itt is a megfelelő sorok összegeiből kapott diagonális mátrix adja.

norma típusa	$p_1$	$p_2$	$p_3$	$p_4$	$p_5$	min. ért.	futás idő(mp)
Frobeniusz	0.0416	0.0521	0.0208	0.0625	0.1250	0.0926	0.007783
spektrál	0.0620	0.0672	0.0277	0.0879	0.1308	0.0543	0.019993

6. táblázat. A legjobban közelítő diagonális mátrix elemei két különböző norma szerint kétdimenziós esetben.

Ebben az esetben láthatjuk, hogy a minimum érték sokkal kisebb, mint az összegzésnél. Ezért ez egy sokkal jobb közelítési mód ebben az esetben. Ebből azt a következtetést lehet levonni, hogy az összegzés elég pontos egyszerűbb szerkezetű mátrixok esetén. De ez az optimalizálási feladat sokkal jobb eredményeket tud elérni bonyolultabb mátrixokon.

### 3.7. Diagonális $M$ előállítása

Az eddigi tömegmátrix közelítési módszerek egy meglévő  $M$  mátrixhoz próbáltak megfelelő diagonális mátrixot találni. Egy más hozzáállás az, hogy próbáljuk meg úgy létrehozni az eredeti tömegmátrixot, hogy az eleve diagonális legyen. Ezt az osztópontok megfelelő kiválasztásával akarjuk elérni.

**3.4. Definíció** (Gauss–Lobato-kvadratura). *Ez a kvadratura a Gauss-kvadratura egy fajtája. Abban különbözik az eredetitől, hogy az integrálás intervallumának a végpontjai is kvadratura-pontok. A többi kvadratura pontjai a  $P_{n-1}$  Legendre-polinom deriváltjának a gyökei.*

**3.1. Állítás.** *A fenti módszer pontos az összes legfeljebb  $2n - 3$ -ad fokú polinom esetén.*

Képletet is adhatunk a kvadratura pontosságára a  $[-1, 1]$  intervallumon a [1] és [6] klasszikus munkák alapján:

$$\int_{-1}^1 f(x) dx = \frac{2}{n(n-1)}(f(-1) + f(1)) + \sum_{i=2}^{n-1} \omega_i f(x_i) + R_n,$$

ahol a súlyokat a következő képlettel számoljuk  $1 < i < n$  esetén:

$$\omega_i = \frac{2}{n(n-1)(P_{n-1}(a_i))^2}.$$

Itt  $P_{n-1}$  az  $n - 1$ -ed fokú Legendre-polinom, továbbá

$$R_n = \frac{-n(n-1)^3 2^{2n-1} ((n-2)!)^4}{(2n-1)((2n-2)!)^3} f^{(2n-2)}(\epsilon).$$

A definícióból következik, hogy  $a_1$  és  $a_n$  súlyai  $\omega_1 = \omega_n = \frac{2}{n(n-1)}$ .

Mivel ez a módszer eredetileg csak a  $[-1, 1]$  intervallumra van definiálva, ezért át kell definiálni egy tetszőleges  $[a, b]$  intervallumra. Az  $a_i$  pontokat ez esetben úgy kaphatjuk meg, ha az eredeti  $P_n$  Legendre-polinomot eltoljuk, és az  $[a, b]$  intervallumra képezzük. Ekkor a

$$P_n^{[a,b]}(x) = P_n\left(\frac{2}{b-a}\left(x - \frac{a+b}{2}\right)\right)$$

polinom gyökeit kell kiszámolnunk.

A súlyok számolásánál ennek megfelelően a  $P_{n-1}^{[a,b]}$  függvényeket kell használnunk.

Ekkor a kvadratura felírható lesz a következő módon:

$$\int_a^b f(x) dx \approx \frac{b-a}{2} \sum_{i=1}^n \omega_i f(a_i).$$

#### 3.7.1. Egy kvadratura pontjainak naiv megközelítése

Amikor  $\Omega$  egydimenziós, akkor az  $\int_{\Omega} b_i(x)b_j(x) dx$  integrált könnyen közelíthetjük kvadraturákkal.

Ha ennek a kvadraturának az  $a_i$  pontjait úgy választjuk meg, hogy az  $x_i$  pontokra essenek, akkor könnyű lesz kiszámolni. Ugyanis a  $b_i$  tulajdonsága az, hogy  $x_i$ -ben 1 és  $x_{j \neq i}$  esetén 0. Az összes ilyen bázisfüggvény esetén igaz ez  $\Omega$  mindegyik osztópontjára. Ekkor kvadraturánk így néz ki:

$$M_{i,j} = \int_{\Omega} b_i b_j = \sum_{k=1}^{K_n} b_i(a_k) * b_j(a_k) * \omega_s,$$

Ahol  $K_n$  a kvadratura pontok száma. Mivel az  $a_k$  pontokat úgy vettük fel, hogy az  $x_i$  pontokra essenek, ezért ez az egyenlet felírható a következő alakban:

$$M_{i,j} = \int_{\Omega} b_i b_j = \sum_{i=1}^n b_i(x_i) * b_j(x_i) * \omega_i.$$

Ez az összeg csak akkor lesz nem nulla, ha  $b_i$  és  $b_j$  egyszerre nem nulla értéket vesznek fel. De ez az  $x_i$  pontokon nézve csak akkor eshet meg, ha  $i = j$ . Ebből következik, hogy  $M$  diagonális lesz. Így  $M$  kiszámításához már csak az  $\omega_i$  súlyokat kell meghatározni.

Egy módszer az  $\omega_i$  súlyok kiszámolására egy lineáris egyenletrendszer megoldása. Ez az egyenletrendszer a következő módon épül fel. Az alapötlet az, hogy van  $n$  darab pontunk, amin egy integrált akarunk közelíteni és  $n$  db  $\omega$  súlyunk, amiket ki akarunk számolni. Akkor számolhatjuk úgy, hogy legalább  $n - 1$ -ed fokú polinomra pontos legyen. Ezt így írhatjuk fel:

$$\mathbf{V} * \omega = b$$

$$\mathbf{V}\omega = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ x_1 & x_2 & x_3 & \cdots & x_n \\ x_1^2 & x_2^2 & x_3^2 & \cdots & x_n^2 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_1^{n-1} & x_2^{n-1} & x_3^{n-1} & \cdots & x_n^{n-1} \end{bmatrix} \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \\ \vdots \\ \omega_n \end{bmatrix} = \begin{bmatrix} \int_{\Omega} 1 dx \\ \int_{\Omega} x dx \\ \int_{\Omega} x^2 dx \\ \vdots \\ \int_{\Omega} x^{n-1} dx \end{bmatrix}$$

Itt a  $V$  mátrix az ún. Vandermonde-mátrix.

De ez a megoldás sajnos nem jó, mert a  $\omega_i$  súlyok közötti különbség nagyon nagy lesz és  $n$ -et minél jobban növeljük, annál jobban oszcillálnak. Ez azt eredményezi, hogy a kvadraturával kapott eredmények nagyon pontatlanok lesznek.[9]

Habár ez a módszer nem jó, de azt bemutatja, hogyha olyan kvadraturát veszünk, amelynek alappontjai az eredeti  $x_i$  pontok, akkor a tömegmátrixunk diagonális lesz.

### 3.7.2. Megoldás Gauss–Lobatto-kvadraturával egydimenziós esetben

Az előző gondolatnál egy sokkal jobb megoldás az, ha fordítva dolgozunk. Először választjuk a kvadratura pontokat, és csak utána választjuk ki az  $x_i$  pontjainkat. Ehhez használjuk fel a Gauss–Lobatto-kvadraturát. Ekkor az  $a_i$  kvadratura pontok az  $n$ -ed fokú Legendre-polinom gyökei lesznek. Most az  $x_i$  pontokat választjuk meg úgy, hogy egybeessenek az  $a_i$  pontokkal.



Első lépésként  $\Omega$ -át tetszőlegesen szakaszokra bontjuk. Az összes ilyen szakaszon belül ugyanannyi  $x_i$  pontot fogunk felvenni a következő módon.

Egy tetszőleges  $[a, b]$  szakaszon a Gauss–Lobatto-szabályhoz választunk  $a_i$  kvadrátúra pontokat, majd az  $x_i$  pontokat ugyanoda helyezzük el. A kvadrátúra pontjai a korábban definiált  $P_{m-1}^{[a,b]}$  Legendre-polinom deriváltjának a gyökei, valamint az  $a$  és  $b$  pont. Itt  $m$  a kvadrátúra pontok száma  $[a, b]$ -n.

A súlyokat a következő képpel lehet kiszámolni:

$$\omega_i = \begin{cases} \frac{2}{m(m-1)[P_{m-1}^{[a,b]}]'^2} & i \neq 1, m \\ \frac{2}{m(m-1)} & i = 1, m \end{cases}$$

Az előbb elhelyezett  $x_i$  pontokhoz létre kell hoznunk  $b_i$  bázisfüggvényeket. Ezt megtehetjük Lagrange-interpolációval: egy  $x_i$  ponthoz vesszük az összes többi pontot, ami vele egy  $[a, b]$  intervallumban van.

Itt legyen  $b_i(x_i) = 1$  és minden más  $x_j$  pontban nulla. Ekkor kapunk egy  $m-1$ -ed fokú Lagrange-bázispolinomot, amely konstans 0 az  $[a, b]$  intervallumon kívül. Ha  $x_i$  pont két intervallum határán, van akkor mindkét intervallumon elvégezzük ezt és az összegük fogja alkotni  $b_i$ -t.

Ennek a konstrukciónak köszönhetően, ha két bázisfüggvény tartójának metszete üres vagy egy pontból áll, akkor a szorzatuk az összes kvadrátúra pontban 0. Ezért ezután csak egyes intervallumokat, és olyan bázisfüggvényeket vizsgálunk, amelyek itt nem nullák.

Most a  $b_1$  és a  $b_m$  bázisfüggvények az intervallum határain nem nullák, így folytonosságuk miatt a szomszédos intervallumukban nem nullaként értelmezve vannak. Viszont ez két különböző intervallumon történik, ezért  $b_1$  és  $b_m$  csak a közös intervallumon egyszerre nem 0. Ezt belátva a bázisfüggvények szorzatának az integrálját szűkíteni lehet.

$$\int_{\Omega} b_i b_j = \int_a^b b_i b_j$$

Itt fel lehet írni  $M$  tömegmátrix elemeit Gauss–Lobatto-kvadrátúrával.

$$M_{ij} = \int_{\Omega} b_i b_j = \int_a^b b_i b_j = \frac{b-a}{2} \sum_{k=1}^n \omega_k b_i(a_k) * b_j(a_k)$$

Az eddigi konstrukcióval elértük azt, hogy az  $a_k$  pontok éppen az  $x_i$  pontok legyenek. Ennek köszönhetően a  $b_i(a_k) * b_j(a_k)$  szorzat csak akkor lehet nem 0, ha  $i = j = k$ .

Ezért  $M$  minden nem diagonális eleme 0 lesz. A diagonális elemek egyenlete pedig felírható egyszerűbben:

$$M_{ii} = \int_a^b b_i b_i = \frac{b-a}{2} \omega_i = \frac{b-a}{2} \frac{2}{m(m-1)[P_{m-1}^{[a,b]}]'^2} = \frac{b-a}{m(m-1)[P_{m-1}^{[a,b]}]'^2(x_i)}$$

Ha  $x_i$  egy intervallum határa, akkor a két intervallum hosszát összeadjuk:

$$M_{ii} = \int_a^b b_i b_i = \frac{(b_1 - a_1) + (b_2 - a_2)}{2} \omega_i = \frac{(b_1 - a_1) + (b_2 - a_2)}{2} \frac{2}{m(m-1)} =$$

$$= \frac{(b_1 - a_1) + (b_2 - a_2)}{m(m - 1)}$$

Nézzük meg egy konkrét példán keresztül:

$\Omega$  legyen a  $[0, 1]$  intervallum. Ezt osszuk fel három részre: a  $[0, 0.5]$ ,  $[0.5, 0.6]$  és a  $[0.6, 1]$  intervallumokra. Legyen  $m$  értéke is 4.

Az  $a_i$  pontok kiszámítása:

- A  $[0, 0.5]$  intervallumhoz tartozó Legendre-polinom:

$$P_3^{[0,0.5]}(x) = 0.5(5(4(x - 0.25))^3 - 3(4(x - 0.25)))$$

$$\dot{P}_3^{[0,0.5]}(x) = 24 - 240x + 480x^2$$

Ennek a gyökei  $a_1 = 0, a_2 = \frac{1}{4} - \frac{1}{4\sqrt{5}}, a_3 = \frac{1}{4} + \frac{1}{4\sqrt{5}}$  és  $a_4 = 0.5$ .

- A  $[0.5, 0.6]$  intervallumhoz tartozó Legendre-polinom:

$$P_3^{[0.5,0.6]}(x) = 0.5(5(20(x - 0.55))^3 - 3(20(x - 0.55)))$$

$$\dot{P}_3^{[0.5,0.6]}(x) = 18120 - 66000x + 60000x^2$$

Ennek a gyökei  $a_4 = 0.5, a_5 = 0.527639, a_6 = 0.572361$  és  $a_7 = 0.6$ .

- A  $[0.6, 1]$  intervallumhoz tartozó Legendre-polinom:

$$P_3^{[0.6,1]}(x) = 0.5(5(5(x - 0.8))^3 - 3(5(x - 0.8)))$$

$$\dot{P}_3^{[0.6,1]}(x) = 592.5 - 1500x + 937.5x^2$$

Ennek a gyökei  $a_7 = 0.6, a_8 = 0.710557, a_9 = 0.889443$  és  $a_{10} = 1$ .

A hozzájuk tartozó  $\omega_i$ -k számolásához ki kell számolni minden intervallumnak a  $P_4^{[a,b]}$  Legendre-polinomját.

- A  $[0, 0.5]$  intervallumhoz tartozó Legendre-polinom:

$$P_4^{[0,0.5]}(x) = \frac{1}{8}(35(4(x - 0.25))^4 - 30(4(x - 0.25))^2 + 3) .$$

Ezeket az értékeket behelyettesítve a képletbe a következő súlyokat kapjuk:  $\omega_1 = \frac{1}{6}, \omega_2 = 4.166, \omega_3 = 4.166$  és  $\omega_4 = \frac{1}{6}$ .

- A  $[0.5, 0.6]$  intervallumhoz tartozó Legendre-polinom:

$$P_4^{[0.5,0.6]}(x) = \frac{1}{8}(35(20(x - 0.55))^4 - 30(20(x - 0.55))^2 + 3) .$$

Ezeket az értékeket behelyettesítve a képletbe a következő súlyokat kapjuk:  $\omega_1 = \frac{1}{6}, \omega_2 = 4.1662, \omega_3 = 4.1662$  és  $\omega_4 = \frac{1}{6}$ .

- A  $[0.6, 1]$  intervallumhoz tartozó Legendre-polinom és deriváltja:

$$P_4^{[0.6,1]}(x) = \frac{1}{8}(35(5(x-0.8))^4 - 30(5(x-0.8))^2 + 3).$$

Ezeket az értékeket behelyettesítve a képletbe a következő súlyokat kapjuk:  $\omega_1 = \frac{1}{6}$ ,  $\omega_2 = 4.1665$ ,  $\omega_3 = 4.1665$  és  $\omega_4 = \frac{1}{6}$ .

Ekkor a tömegmátrix a következő:

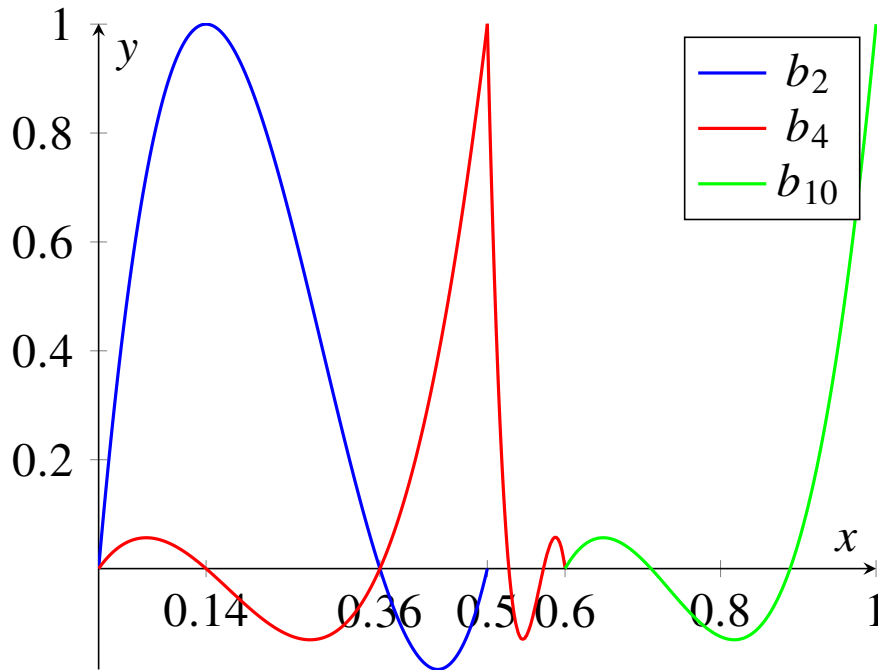
$$M = \begin{bmatrix} \frac{1}{24} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1.0415 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1.0415 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{20} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.2083 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.2083 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{24} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.83324 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.83324 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{30} \end{bmatrix}$$

Ugyan a tömegmátrixhoz nem kellett konkrétan kiszámolni a bázisfüggvényeket, de én kiszámoltam hármát, hogy szemléltetni tudjam, pontosan hogyan is néznek ki.

$$b_2 = \begin{cases} 16.1803x - 77.082x^2 + 89.4427x^3 & \text{ha } x \in [0, 0.5] \\ 0 & \text{ha } x \notin [0, 0.5] \end{cases}$$

$$b_4 = \begin{cases} 2x - 20x^2 + 40x^3 & \text{ha } x \in [0, 0.5) \\ 1 & \text{ha } x = 0.5 \\ 906.006 - 4810.03x + 8500.06x^2 - 5000.04x^3 & \text{ha } x \in (0.5, 0.6] \\ 0 & \text{ha } x \notin [0, 0.6] \end{cases}$$

$$b_{10} = \begin{cases} -29.625 + 124.375x - 171.875x^2 + 78.1251x^3 & \text{ha } x \in [0.6, 1] \\ 0 & \text{ha } x \notin [0.6, 1] \end{cases}$$



### 3.7.3. Spektrál elem módszer

Ezt a módszert több dimenzióban használjuk. A klasszikus esetben ehhez az  $\Omega$  tartományt egy téglalapokból vagy téglatestekből álló hálóra bontjuk.

A többdimenziós bázisfüggvények elkészítéséhez felhasználjuk az egyváltozós bázisfüggvényeket, méghozzá vektoriálisan szorozzuk őket. Így az  $x_i$  ponthoz tartozó bázisfüggvényt így kapjuk meg:  $\phi_i(x, y, \dots) = b_{i_x}(x) \otimes b_{i_y}(y) \otimes \dots$

Első lépésként az  $\Omega$  tartományt tetszőleges részekre bontjuk. Ezeken a részeken mind elvégezzük a következő lépéseket.

Az integrálok értékét Gauss–Lobatto-kvadraturával közelítjük. Az előző szakaszban(3.7.2.) használt módhoz hasonlóan fogjuk kiválasztani a kvadratura pontjait. A kvadratura pontokat úgy számoljuk ki, hogy szétválasztjuk egydimenziós esetekre. Ekkor 3.7.2. szakaszban meghatározott módon már ki tudjuk számolni a pontokat.

A súlyok kiszámítását is ugyanígy egydimenziós esetekre bontással végezzük el. Majd az egy ponthoz tartozó súlyokat átlagoljuk. Ezután a kvadratura így néz ki:

$$\int_{\Omega} \phi_i \phi_j = \sum_{k=1}^{N_h} \phi_i(x_k) * \phi_j(x_k) * \omega_k$$

Itt a 3.7.2. szakaszban bemutatott okok miatt csak a diagonális elemek maradnak meg. Ezek így fognak kinézni:

$$M_{ii} = \omega_i * \frac{|b_i \text{ saját része}|}{|\Omega|},$$

ahol  $|\cdot|$  a megfelelő dimenzióban értelmezett Lebesgue-mérték (azaz általában terület vagy térfogat).

A módszerben használt pontok az  $\Omega$  tartományt pontjai  $n$ -dimenziós kockákra bontják. Ezzel az lehet a gond, hogy kockákkal nehéz lefedni az eredeti testet. Ezért általában pontatlanabb lesz a közelítésnek ezen része annál, mintha háromszögekkel vagy tetraéderekkel dolgoznánk.[2] [10]

### 3.8. Tömegmátrix közelítése háromszögrácson lineáris bázisfüggvényekkel

Az a célunk, hogy a 3.4.3. szakaszban magadott bázisfüggvények integrálját olyan kvadraturával számoljuk ki, hogy diagonális mátrixot kapjunk, ahogy a 3.7.2. szakaszban.

Ha lineáris bázisfüggvényeket használunk, akkor a tömegmátrix elemeinek kiszámításához a kettős integrállá való átírás után a trapézsabályt tudjuk alkalmazni. Viszont ez a módszer nem fog pontos eredményt adni, hanem csak egy közelítést. Mivel a trapézsabály nem pontos másod- és harmadfokú polinomokra. Ilyen polinomok pedig szerepelnek ebben az integrálásban, a  $\phi_i * \phi_j$  másodfokú és az  $\frac{1-x_1^R}{2}(\phi_i(x_1, 0)\phi_j(x_1, 0))$  pedig harmadfokú polinom lesz.

$$\begin{aligned} \int_{S_k} b_i b_j dx &= \dots = 2T_k \int_0^1 \int_0^{1-x_1^R} \phi_i^R \phi_j^R dx_2^R dx_1^R \approx \\ &\approx 2T_k \int_0^1 \frac{1-x_1^R}{2} (\phi_i(x_1, 0)\phi_j(x_1, 0) + \phi_i(x_1, 1-x_1^R)\phi_j(x_1, 1-x_1^R)) dx_1^R \approx \\ &\approx 2T_k \frac{1}{2} * \left[ \frac{1-0}{2} (\phi_i(0, 0)\phi_j(0, 0) + \phi_i(0, 1-0)\phi_j(0, 1-0)) + \right. \\ &\quad \left. + \frac{1-1}{2} (\phi_i(1, 0)\phi_j(1, 0) + \phi_i(1, 1-1)\phi_j(1, 1-1)) \right] = \\ &= 2T_k \frac{1}{2} \left[ \frac{1}{2} (\phi_i(0, 0)\phi_j(0, 0) + \phi_i(0, 1)\phi_j(0, 1)) + \frac{0}{2} (\phi_i(1, 0)\phi_j(1, 0) + \phi_i(1, 0)\phi_j(1, 0)) \right] \\ &= 2T_k \frac{1}{4} (\phi_i(0, 0)\phi_j(0, 0) + \phi_i(0, 1)\phi_j(0, 1)) \end{aligned}$$

Mivel a  $(0, 0) = A_1^R$  és  $(0, 1) = A_3^R$  pontok a háromszög csúcsai, ezért  $\phi_i$  és  $\phi_j$  nem lehetnek egyszerre nem nullák rajtuk, ha  $i \neq j$ . Ezért a nem diagonális pozíciókban a kvadratura 0 értéket ad.

Ha  $j = i$ , akkor pedig

$$\int_{S_k} \phi_i \phi_j = \frac{T_k}{2}.$$

Így kaptuk a tömegmátrix egy diagonális közelítését, amelyet a háromszögrács elemeinek a területei határoznak meg.

A 3.4.3. szakaszban lévő rácsra alkalmazom ezt a módszert. Majd utána az így kapott  $\tilde{M}$  közelített tömegmátrixot összehasonlíjuk az eredeti tömegmátrixszal.

$$\tilde{M} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1.5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.5 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1.5 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1.5 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.5 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1.5 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} * \frac{1}{8}$$

Az így kapott  $\tilde{M}$  közelített tömegmátrixot összehasonlítjuk az eredeti tömegmátrixszal.

Norma típusa	mátrix mérete	$\ \tilde{M} - M\ $	$\frac{\ \tilde{M}\ }{\ M\ }$
Frobénius	$9 \times 9$	0.3954	2.7690
Spektrál	$9 \times 9$	0.2651	2.3638

7. táblázat. A trapézformulával létrehozott tömegmátrix összehasonlítása az eredetivel.

### 3.9. Gauss–Lobatto-kvadratúra háromszögrácson

Amikor magasabb fokú bázisfüggvényekkel akarunk dolgozni, akkor az  $S_k$  háromszögekben fel kell venni képzeletbeli pontokat (ld. 3.4.3. szakasz). Ha ekkor a pontokat a Gauss–Lobatto-kvadratúra szerint vesszük fel, akkor a mátrix diagonális lesz ugyanúgy, mint a 3.7.2. szakaszban. Ez a cikk például erről szól: [5].

## 4. Összefoglalás

Ebben a dolgozatban bemutattam, hogy a tömegmátrix fogalma hogyan kerül elő a végelem-módszerben. Bemutattam, hogy miért jobb, ha a tömegmátrix diagonális. Bevezettem a tömegmátrixok közelítését, aminek az a célja, hogy előállítson egy diagonális tömegmátrixot úgy, hogy az közel van az eredetihez.

Ezt követően a tömegmátrix definíciójához vezető bázisfüggvények felépítésével foglalkoztam. Első lépésnek felírtam azt, hogyan nézzenek ki az egydimenziós lineáris bázisfüggvények, majd az általuk alkotott tömegmátrixot is kiszámoltam. Ezután felírtam azt, hogyan néznek ki a magasabb fokú polinomok által alkotott bázisfüggvények. Egy fontos példa esetén ábrát is készítettem ehhez.

A következő bővítés, amit megtettem az volt, hogy kétváltozós lineáris bázisfüggvényeket vezettem be. Itt láttuk, hogy az eredményt nagyban befolyásolja az, hogy a számítási tartományt milyen alakzatokra osztjuk fel. Ehhez bemutattam, hogy egy megadott háromszögrácson hogyan néz ki a tömegmátrix egy sora. Ehhez az esethez is készült ábra.

A következő lépés az volt, hogy bevezettem a baricentrikus koordinátákat és megmutattam, hogy bármilyen háromszögekből álló rácstra hogyan kell kiszámolni a lokális bázisfüggvényeket. Kiszámoltam egy kis háromszögrácson vett lineáris bázisfüggvények által felépített tömegmátrixot.

Ezután a közelítési módszerekkel foglalkoztam. A módszerek közül az első a sorösszegzés módszere volt. Matlabban írt kód segítségével készítettem táblázatokat ezen eljárás pontosságának vizsgálatára. A következő módszer egy optimalizációs feladatmegoldás. Itt próbáltam létrehozni egy diagonális mátrixot, ami legközelebb van az eredeti tömegmátrixhoz. Az eredményeket összegyűjtöttem egy hasonló táblázatban, mint az összegzésnél tettem.

Az utolsó közelítési módszer ez eredeti tömegmátrix előállításával foglalkozik. A célja, hogy úgy válasszuk ki a bázisfüggvényeket és a végelem-módszer pontjait, hogy a tömegmátrix diagonális legyen. Ezt úgy éri el, hogy a tömegmátrix értékeihez szükséges integrálokat kvadratúrákkal közelíti, még hozzá úgy, hogy a kvadratúra pontjai egybeesnek a végelem-módszer pontjaival. A módszert bemutattam egydimenziós esetben, és végigszámoltam egy konkrét tömegmátrixot.

Bemutattam, hogyan néz ki ez a módszer kétdimenziós esetben, ha a tartomány téglákra vagy téglalapokra van felosztva; ez a spektrál elem módszer. Megmutattam háromszögrács esetén is, hogy egy ilyen közelítés hogyan néz ki.

Végül azzal foglalkoztam, hogy ha nem lineáris bázisfüggvényeket szeretnénk alkalmazni a háromszögrácson, akkor milyen módon lehetne a tömegmátrix diagonális. Ennek tárgyalásához egy friss cikket utaltam, mert ez már túlmutat a jelen dolgozat a keretein.

## 5. Függelék

Összegyűjtöttem a Matlabban írt programkódokat, amelyeket felhasználtam a dolgozat megírásához.

### 5.1. Segédfüggvények

A következő két pontban felsorolom azokat a függvényeket, amelyek az eredeti tömegmátrixokat hozzák létre, ezeket felhasználtam több módszer értékeléséhez is.

#### 5.1.1. három\_diag\_matrix

Az alábbi függvény hozza létre a 3.1. szakaszban elkészített tömegmátrixot.

---

```
1 function [eredmeny_matrix] = három_diag_matrix(meret,diag_sulyok)
2
3 egyesek=ones(meret,1);
4 eredmeny_matrix = spdiags([diag_sulyok(1)*egyesek, diag_sulyok(2)*egyesek, ...
5     diag_sulyok(3)*egyesek],[-1,0,1],meret,meret);
6 eredmeny_matrix = eredmeny_matrix*(1/meret);
7
8 end
```

---

#### 5.1.2. tomegmatrix\_2d\_linar

Ebben a függvényben tároltam el a 3.4.4. szakaszban létrehozott tömegmátrixot. Így könnyebben fel tudtam használni a későbbi kódokban.

---

```
1 function [M] = tomegmatrix_2d_linar()
2 M = [
3     2/6, 1/12, 0, 1/12, 2/12, 0, 0, 0, 0;
4     1/12, 3/6, 1/12, 0, 2/12, 2/12, 0, 0, 0;
5     0, 1/12, 1/6, 0, 0, 1/12, 0, 0, 0;
6     1/12, 0, 0, 3/6, 2/12, 0, 1/12, 1/12, 0;
7     2/12, 2/12, 0, 2/12, 6/6, 2/12, 0, 2/12, 2/12;
8     0, 2/12, 1/12, 0, 2/12, 3/6, 0, 0, 1/12;
9     0, 0, 0, 1/12, 0, 0, 1/6, 1/12, 0;
10    0, 0, 0, 2/12, 2/12, 0, 1/12, 2/6, 1/12;
11    0, 0, 0, 0, 2/12, 1/12, 0, 1/12, 2/6
12 ]*(1/8);
13 end
```

---



## 5.2. Összegzéssel való közelítés kódjai

A következők a 3.5. szakaszban lévő módszert implementáló kódok.

### 5.2.1. osszegzessel\_valo\_kozelites

Ez a függvény végzi el a sorok összegzését és visszaadja az új közelített diagonális mátrixot.

---

```
1 function [eredmeny_matrix] = osszegzessel_valo_kozelites(input_matrix)
2
3 [meret,~] = size(input_matrix);
4 sor_osszeg = zeros(meret,1);
5 [sor_index,oszop_index] = find(input_matrix);
6 for index = 1: meret
7     sor_osszeg(sor_index(index)) = sor_osszeg(sor_index(index)) +
8         input_matrix(sor_index(index),oszop_index(index));
9
10 eredmeny_matrix = spdiags(sor_osszeg(:),0,meret,meret);
11
12 end
```

---

### 5.2.2. main\_osszegzessel\_valo\_kozelites

Ennek a kódnak a futtatása adja vissza azokat az értékeket, amelyeket az 1. és a 2. táblázatban mutatok be.

---

```
1 n=input("Hanyadik 10 hatvany legyen a matrix merete?");
2 norma_tipus = input("Milyen tipusu normat használjon?");
3
4 M = három_diag_matrix(10^n,[1/6,2/3,1/6]);
5
6 tic
7 T = osszegzessel_valo_kozelites(M);
8 K = T-M;
9 hanyados = norm(full(T))/norm(full(M),norma_tipus)
10 kulonbseg = norm(full(K),norma_tipus)
11 toc
```

---

### 5.2.3. main\_osszegzessel\_valo\_kozelites\_2d

Ez a 5.2.2. szakaszban lévő kód módosított változata. Így ez a kód a 3.4.4. szakaszban megadott tömegmátrixszal dolgozik. A kód eredményei a 3. táblázatban vannak.

---

```
1 norma_tipus = input("Milyen tipusu normat használjon?");
2
3 M = tomegmatrix_2d_linar();
```

```

4
5 tic
6 T = osszegzessel_valo_kozelites(M);
7 K = T-M;
8 hanyados = norm(full(T))/norm(full(M),norma_tipus)
9 kulonbseg = norm(full(K),norma_tipus)
10 toc

```

---

### 5.3. Legjobban közelítő diagonális mátrix keresésének kódjai

Itt a 3.6. szakaszban bemutatott módszert implementáló kódok találhatóak.

#### 5.3.1. kulonbseg\_normajat\_szamolo

Ennek a függvénynek az eredményét fogjuk majd optimalizálni.

```

1 function [eredmeny] = kulonbseg_normajat_szamolo(p_1,p_2,M,norma_tipus)
2
3 [meret,] = size(M);
4 vector = ones(meret)*p_2;
5 vector(1) = p_1;
6 vector(end) = p_1;
7 T = diag(vector);
8 eredmeny = norm((T-M),norma_tipus);
9
10 end

```

---

#### 5.3.2. T\_optimalizalo\_linaris

Ez a függvény oldja meg az optimalizálási problémát az "fminsearch" függvény segítségével.

```

1 function [p1,p2,minimum_ertek] = T_optimalizalo_linaris(n,norma_tipus)
2 M = harom_diag_matrix(10^n,[1/6,2/3,1/6]);
3 p_1_kezdo_ertek = 5/6*(1/10^n);
4 p_2_kezdo_ertek = 1*(1/10^n);
5 [otimalis_parameterek, minimum_ertek] = fminsearch(@(x)
6     kulonbseg_normajat_szamolo(x(1), x(2),M,norma_tipus) +
7     (abs(x(1))-x(1))*10^n + (abs(x(2))-x(2))*10^n, [p_1_kezdo_ertek,
8     p_2_kezdo_ertek]);
9 p1 = otimalis_parameterek(1);
10 p2 = otimalis_parameterek(2);
11 end

```

---

### 5.3.3. main\_T\_optimalizalas

Ennek a kódnak a futtatása adja vissza azokat az értékeket, amelyeket a 4. és az 5. táblázatban mutatok be.

---

```
1 n=input("Hanyadik 10 hatvany legyen a matrix merete?");
2 norma_tipus = input("Milyen tipusu normat használjon?");
3 tic
4 [p1,p2,minimum_ertek] = T_optimalizalo_linaris(n,norma_tipus)
5 toc
```

---

### 5.4. kulonbseg\_normajat\_szamolo\_2d

A 5.3.1. szakaszban lévő kód kétdimenziós esetre módosítva.

---

```
1 function [eredmeny] = kulonbseg_normajat_szamolo_2d(p_1, p_2, p_3, p_4,
2         p_5,M,norma_tipus)
3
4 vector = [p_1, p_2, p_3, p_4, p_5, p_4, p_3, p_2, p_1];
5 T = diag(vector);
6 eredmeny = norm((T-M),norma_tipus);
7 end
```

---

#### 5.4.1. T\_optimalizalo\_linaris\_2d

A 5.3.2. szakaszban lévő kód kétdimenziós esetre módosítva.

---

```
1 function [p1,p2,p3,p4,p5,minimum_ertek] =
2         T_optimalizalo_linaris_2d(norma_tipus)
3 M = tomegmatrix_2d_linar();
4 p_1_kezdo_ertek = 2/3*(1/8);
5 p_2_kezdo_ertek = 1*(1/8);
6 p_3_kezdo_ertek = 1/3*(1/8);
7 p_4_kezdo_ertek = 11/12*(1/8);
8 p_5_kezdo_ertek = 2*(1/8);
9 [otimalis_parameterek, minimum_ertek] = fminsearch(@(x)
10         kulonbseg_normajat_szamolo(x(1), x(2), x(3), x(4), x(5),M,norma_tipus) +
11         (abs(x(1))-x(1))*10 + (abs(x(2))-x(2))*10 + (abs(x(3))-x(3))*10
12         +(abs(x(4))-x(4))*10 +(abs(x(5))-x(5))*10, [p_1_kezdo_ertek,
13         p_2_kezdo_ertek,p_3_kezdo_ertek,p_4_kezdo_ertek,p_5_kezdo_ertek]);
14 p1 = otimalis_parameterek(1);
15 p2 = otimalis_parameterek(2);
16 p3 = otimalis_parameterek(3);
17 p4 = otimalis_parameterek(4);
18 p5 = otimalis_parameterek(5);
```

### 5.4.2. main\_T\_optimalizalas\_2d

Ennek a kódnak a futtatása adja vissza azokat az értékeket, amelyeket a 6. táblázatban használtam fel.

---

```
1 norma_tipus = input("Milyen tipusu normat használjon?");
2 tic
3 [p1,p2,p3,p4,p5,minimum_ertek] = T_optimalizalo_linaris_2D(norma_tipus)
4 toc
```

---

## 5.5. A trapézformulával kapott eredmény vizsgálatának a kódja

Itt a 3.8. szekcióban létrehozott mátrixot hasonlítom össze az eredeti tömegmátrixszal, amit a 3.4.4. szakaszban számoltam ki.

Ennek a kódnak a futtatása adja vissza azokat az értékeket, amelyeket a 7. táblázatban használtam fel.

---

```
1 diagonalis_elemek = [1, 1.5, 0.5, 1.5, 3, 1.5, 0.5, 1.5, 1];
2 M_trapez = diag(diagonalis_elemek)*(1/8);
3
4 M = tomegmatrix_2d_linar();
5
6 norm(M_trapez-M, 'fro')
7 norm(M_trapez-M, 2)
8 norm(M_trapez, 'fro')/norm(M, 'fro')
9 norm(M_trapez, 2)/norm(M, 2)
```

---

## Hivatkozások

- [1] Milton Abramowitz and Irene Ann Stegun (eds.), *Handbook of mathematical functions with formulas, graphs, and mathematical tables*, ninth reprint with additional corrections of tenth original printing with corrections (december 1972); first ed. ed., Applied Mathematics Series, vol. 55, United States Department of Commerce, National Bureau of Standards; Dover Publications, Washington D.C.; New York, June 1983.
- [2] S. Geevers, W. A. Mulder, and J. J. W. van der Vegt, *New higher-order mass-lumped tetrahedral elements for wave propagation modelling*, SIAM J. Sci. Comput. **40** (2018), no. 5, A2830–A2857. MR 3852719
- [3] Hans Petter Langtangen, *Finite element method (fem) documentation*, [https://hplgit.github.io/INF5620/doc/pub/sphinx-fem/.\\_main\\_fem009.html](https://hplgit.github.io/INF5620/doc/pub/sphinx-fem/._main_fem009.html), Accessed: 2024-05-30.
- [4] Hervé Le Dret, *Master 1 lecture notes 2011–2012: Numerical approximation of pdes*, Lecture Notes, January 2012, Sorbonne universités.
- [5] Cuiyun Liu and Bo Liu, *Symmetric and asymmetric gauss and gauss–lobatto quadrature rules for triangles and their applications to high-order finite element analyses*, Journal of Computational and Applied Mathematics **437** (2024), 115451.
- [6] William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery, *Numerical recipes 3rd edition: The art of scientific computing*, 3rd ed., Cambridge University Press, 2007.
- [7] Dr. Tamás Péter, Bojtos Attila, Décsei-Paróczy Annamária, and Dr. Fekete Róbert Tamás, *Végeselem módszerek*, BME MOGI, 2014.
- [8] Alfio Quarteroni, *Numerical models for differential problems*, third ed., MS&A. Modeling, Simulation and Applications, vol. 16, Springer, Cham, 2017. MR 3702005
- [9] Gilbert Strang, *The geometry of linear equations | mit 18.06sc linear algebra, fall 2011*, YouTube, 2023, Accessed: 2024-05-25.
- [10] S.J. Thomas and R. Loft, *Parallel spectral element atmospheric model*, Parallel Computational Fluid Dynamics 2000 (C.B. Jenssen, H.I. Andersson, A. Ecer, N. Satofuka, T. Kvamsdal, B. Pettersen, J. Periaux, and P. Fox, eds.), North-Holland, Amsterdam, 2001, pp. 331–337.

# NYILATKOZAT

**Név:** Kovács Tamás

**ELTE Természettudományi Kar, szak:** Matematika

**NEPTUN azonosító:** QEQXNM

**Szakedolgozat címe:**

A tömegmátrix közelítései

A **szakedolgozat** szerzőjeként fegyelmi felelősségem tudatában kijelentem, hogy a dolgozatom önálló szellemi alkotásom, abban a hivatkozások és idézések standard szabályait következetesen alkalmaztam, mások által írt részeket a megfelelő idézés nélkül nem használtam fel.

Budapest, 2024. Június 3.



\_\_\_\_\_  
*a hallgató aláírása*