

SZAKDOLGOZAT

Tar Gergő
Fehérvári Zoltán

Debrecen
2010

DEBRECENI EGYETEM
INFORMAIKAI KAR

Íj modellezése végeelem módszerrel

Témavezető:
Dr. Tóth László
egyetemi adjunktus

Készítette:
Tar Gergő
Fehérvári Zoltán
Mérnök informatikus

Debrecen
2010

Tartalomjegyzék

1. Bevezetés	2
2. Az íj működése	3
2.1. Egy egyszerű íjmodell	3
2.2. Az egyenes íj	4
2.3. A visszacsapó íj	5
3. Végeselem módszer	7
3.1. Végeselem módszer fizikája	7
3.1.1. Newtoni mechanika	7
3.1.2. Részecskék rendszere	8
3.1.3. Mozgás egyenletek leírása általános koordinátákkal	12
3.1.4. D’Alambert elv	12
3.1.5. Lagrange függvény	14
3.1.6. Példa a Lagrange függvény alkalmazhatóságára	15
3.1.7. Rugalmasságtan	16
3.2. Végeselem programozás	19
3.2.1. Előfeldolgozás	20
3.2.2. Peremfeltételek definiálása	21
3.2.3. Algebrai egyenlet megoldása	21
3.2.4. Eredmények kiértékelése	21
4. Végeselem módszer megvalósítása Javaban [3]	22
4.1. Végeselem analízis főbb lépései	22
4.2. A végeselem egyenlet	22
4.2.1. Galerkin módszer	23
4.2.2. Variációs megközelítés	25
4.2.3. Példa alakfüggvény meghatározására	26
4.3. A program működése	28
4.3.1. A modell felépítéséhez szükséges adatok	29
4.3.2. A modellen végbemenő erőhatások definiálásához szükséges adatok	29
4.3.3. Eredmények és a modell kirajzolása	30
4.3.4. Egyenes íj modell	33
5. Összefoglalás	34
6. Saját munkák leírása	34
7. Köszönet nyílvánítás	34

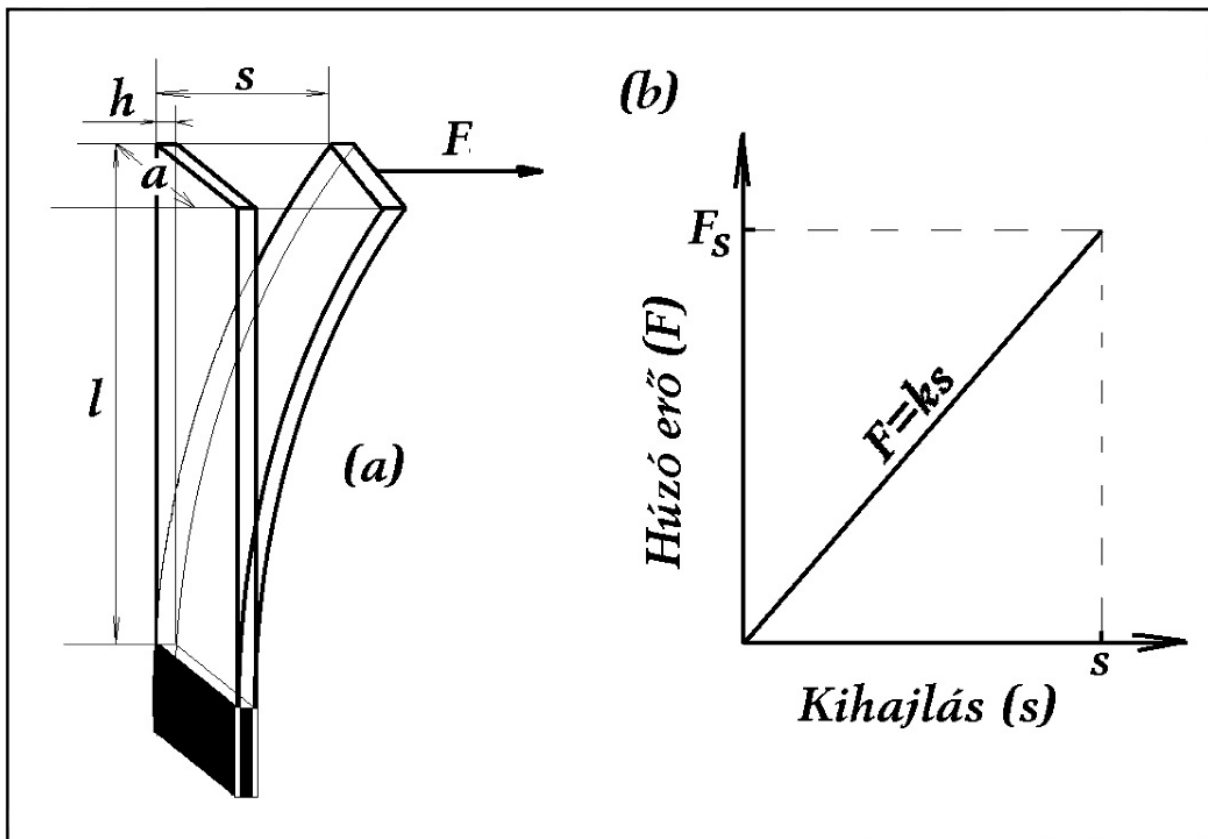
1. Bevezetés

A dolgozatunkban az egyenes, illetve a visszacsapó íj működésével foglalkoztunk. Az íjak viselkedésének leírása kapcsán foglalkoztunk a végeelem módszerrel és a végeelem programozással. A végeelem programozást a Java programozási nyelven keresztül valósítottuk meg, egy egyszerű modellen keresztül bemutattuk a program működését, illetve modelleztünk egy kezdetleges, fából készült rudat. Továbbá terveink között szerepel egy bonyolultabb egyenes, illetve visszacsapó íjmodell végeelem módszerrel történő modellezése, illetve a statikus viselkedés mellett a dinamikus viselkedés vizsgálata is. A felhasználtuk benne a Newtoni mechanikát, az általános koordinátákbeli egyenleteket, a Lagrange függvényeket és a rugalmasságtant is. Alkalmaztuk a Java 3D-t az eredmények szemléltetésére.

2. Az íj működése

2.1. Egy egyszerű íjmodell

A bonyolultabb szerkezetű visszacsapó íj tárgyalása előtt célszerű tisztában lennünk az egyszerű egyenes íj működésének főbb törvényszerűségeivel. [2]



1. ábra. Egyszerű modell [2]

Ennek érdekében tekintsünk egy igen egyszerű modellt. Vegyünk egy fél íjkart, azaz egy l hosszúságú, mindenütt egyenlő keresztmetszetű homogén anyageloszlású rudat, amelyet az egyik végénél befogva függőleges helyzetben rögzítünk, a másik végét pedig vízszintes irányban F erővel húzzuk. Ekkor, - ha a súlyerőktől eltekintünk és csak a kisméretű kihajlások (s) eseteivel foglalkozunk - az íjkar erőtvényére az adódik, hogy:

$$F = ks,$$

ahol

$$k = \frac{E ah^3}{4 l^3}$$

ahol F : húzó erő, k : az erőgörbe meredeksége, vagy másképpen a rugómerevség, E : nyújtási modulus (anyagi minőségtől függ), a : a rúd szélessége, h : a rúd vastagsága, l : a rúd hossza, valamint s : a rúd kihajlása. Látható, hogy az íjkar hajlása (s) egyenesen arányos a feszítő erővel (F).

Az íj erejét a k rugómerevség határozza meg, amit a linearitás miatt az egy adott kihajlásánál (s) mérhető feszítő erővel (F_s) is megadhatunk. A k viszont, amint már láthattuk, függ az íj anyagától (E), valamint annak alakjától (a, h, l), mégpedig úgy, hogy a harmadik hatványra emelés miatt különösen érzékeny az íjkar vastagságának (h) és hosszúságának (l) a megváltozására.

Az alkalmazás szempontjából nézve az íjakat az energiaviszonyaikkal is jellemezni kell. Egyik ezek közül a kifeszítés során az íjba fektetett munka ($W_{feszítés}$), vagyis az íj energiátároló képessége, a vizsgált fél íjkar esetén a következő alakú:

$$W_{feszítés} = \int_0^s F(s)ds = \int_0^s ksds = \frac{ks^2}{2}$$

Azonban a lövés során a nyílvesszőnek átadott energia ($W_{vessző}$) a fellépő energiaveszteségek ($W_{veszteség}$) miatt kevesebb lesz a befektetettnél ($W_{feszítés}$):

$$W_{feszítés} = W_{vessző} + W_{veszteség}$$

Így az íj hatásfoka (η):

$$\eta = \frac{W_{vessző}}{W_{feszítés}} 100\%$$

A veszteséget ($W_{veszteség}$) számos tényező okozhatja, melyek közül a két legjelentősebb; az íjkarok gyorsítására fordított munka, valamint magában az íj anyagában pl. az íjkarban és az idegben- történő energiaelnyelődés. Ezek csökkenthetők, ha a megfelelő alapanyag kiválasztása mellett csökkentjük az íjkarok tömegét, illetve a tehetetlenségi nyomatékát. Azaz, az íjkaroknak könnyűnek kell lenni és a végek felé a megfelelő módon vékonyodniuk kell.

Ezek után a kirepülő vessző energiájának ($W_{vessző}$), vagy másképpen az átütőerejének és tömegének (m) ismeretében kiszámolható a vessző kezdősebessége (v), amitől viszont a lőtávolság függ:

$$W_{vessző} = W_{feszítés} - W_{veszteség}$$

azaz,

$$\frac{1}{2}mv^2 = ks^2 - W_{veszteség}$$

ebből a vessző kezdősebessége (v) az ideális, veszteség nélküli esetben ($W_{veszteség} = 0$):

$$v = s\sqrt{\frac{k}{m}}$$

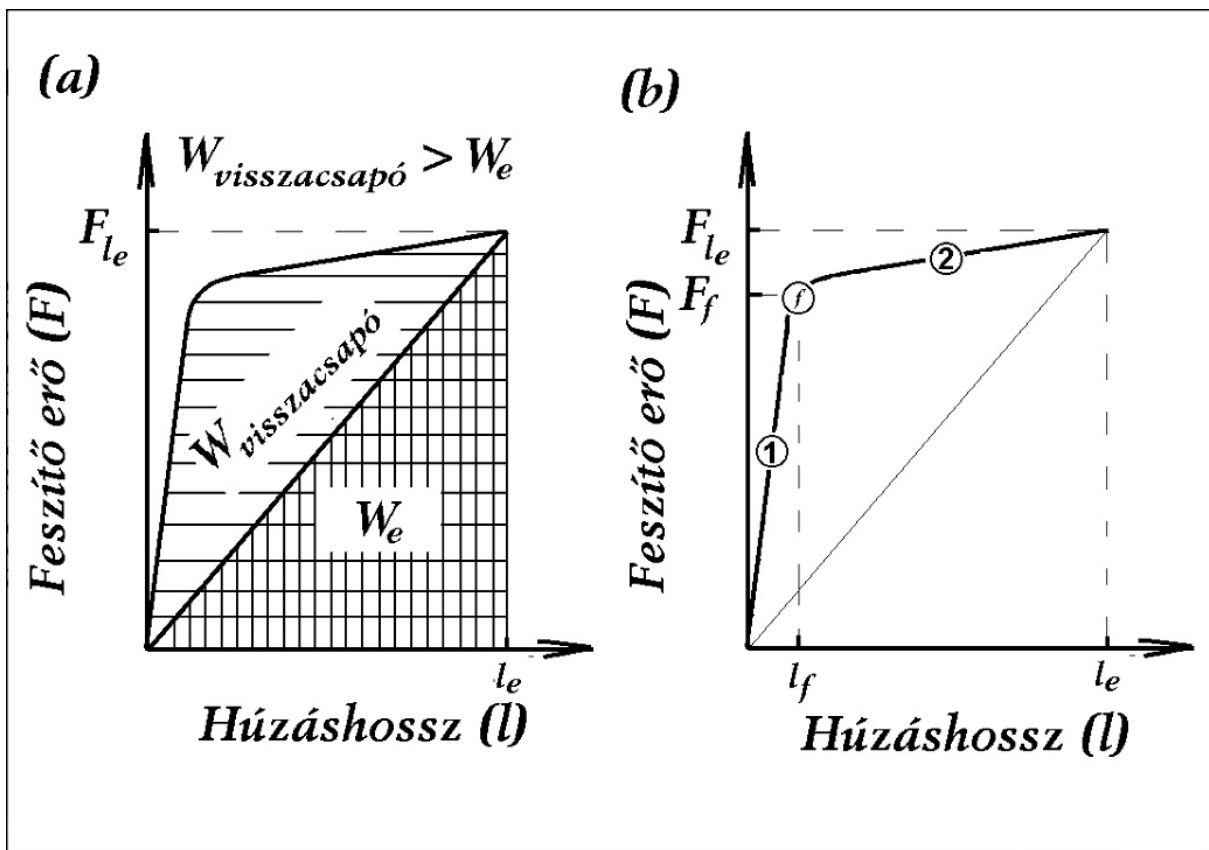
-nek adódik.

Mindezidáig egy erősen leegyszerűsített, "elméleti" egyenes íjról beszéltünk, ami azonban alkalmas volt arra, hogy a működésének az alapelveit megértsük.

2.2. Az egyenes íj

Nézzük most meg, hogy hogyan viselkedik egy valódi egyenes íj. Mivel az egyenes íj paraméterei (E, a, h, l), szemben az előbb tárgyalt esettel, az íjkarok mentén helyről helyre változhatnak és az íjkar végeire ható erő iránya sem állandó, ezért az egyenes íj erőtvénye nem írható fel az 1. egyenlethez hasonló egyszerű alakban. Ezért, vagyis a feladat bonyolultsága miatt, kellett később használnunk a végeelem módszert.

Ennek ellenére a viselkedése kezdetben hasonló az előző esetben már látottakhoz. Vagyis, erőgörbéje a feszítés kezdetén jó közelítéssel egyenesnek mondható, később a meredekség (k) rohamosan növekedni kezd, majd az íj bekeményedik, és az egyre nagyobb feszítőerő következtében el is törhet. Ezt a tartományt tehát érdemes elkerülni a használat során, amit tovább erősíti az a tény, hogy egy a bekeményedési



2. ábra. A visszacsapó íj erőgörbéje és energiátároló képessége, összehasonlítva az egyenes íjjal.[2]

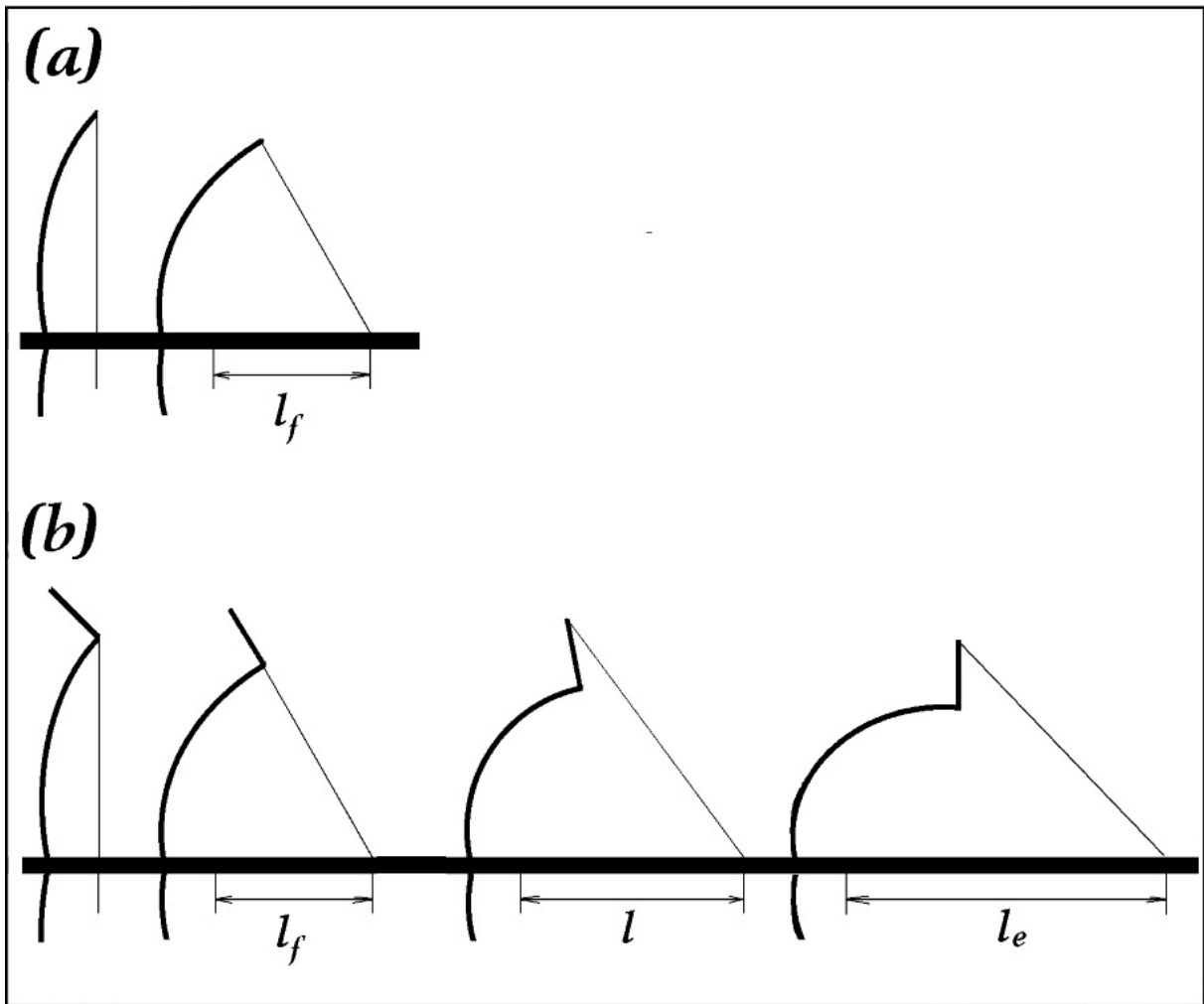
pontján túlhúzott íjban kevesebb energia ($W_{kemény}$) halmozódik fel, mint egy másik, azonos húzáshosszon ugyanakkora feszítőerővel megtartott, de csak az erőgörbe közel lineáris szakaszán működtetett íjban. Nevezük ez utóbbit ekvivalens egyenes íjnak (W_e).

A megkívánt erősségű (F_l) íjat tehát úgy célszerű elkészíteni, hogy azon tartomány húzáshossza (l), amelyen az erőgörbe még közel lineáris futású, az éppen az íjász húzáshosszához igazodjon. Ezzel egyben már láthatók az egyenes íj korlátai is. Vagyis, az átütőerő ($W_{vessző}$) növelése a veszteségek ($W_{veszteség}$) csökkentése mellett a húzáshossz (l), illetve az adott húzáshosszhoz tartozó feszítő erő (vagyis az F_l , azaz a k) növelésével érhető csak el. Ennek viszont határt szab az ember fizikai felépítése; karhosszúsága és ereje.

2.3. A visszacsapó íj

Az azonos feszítőerő (F_l) és húzáshossz (l) melletti átütőerő ($W_{vessző}$) növelésére megoldás lehet egy olyan íj készítése, amelynek erőgörbéje ($F_{visszacsapó} = F_{visszacsapó}(l)$) a megkívánt húzáshosszon végig a fentebb már definiált ekvivalens egyenes íj erőgörbéje ($F_e = F_e(l)$) felett halad.

Ennek érdekében először is készítsünk egy olyan rövid egyenes íjat, amelynek az erőgörbéje meredekebben indul, mint az összehasonlításképpen használt ekvivalens egyenes íjé. Ha egy ilyen kemény íj karjainak a végére előrehajló merev szarvakat szerelünk, akkor létezik egy l_f húzáshossz, ameddig az



3. ábra. Íj működés[2]

íjat megfeszítve az ideg még rásimul az íjkarok végére szerelt merev szarvakra, és ezért a viselkedése az erőgörbe szempontjából az f fordulópontig gyakorlatilag megegyezik a rövid kemény egyenes íjével. Az l_f húzáshossz után azonban az ideg eltávolodik a szarvaktól, aminek következtében megnő az erőkar és az erőgörbe meredeksége csökken, majd végül az l_e húzáshossznál eléri az ekvivalens egyenes íjnak megfelelő végső F_{l_e} feszítőerőt. Hogy az erőgörbe e közel vízszintes szakasza minél laposabb legyen, azt az íjszarvak hosszúságának, illetve az íjkarok által erős hajlításnak kitett íjkar végek "keménységének" a megfelelő csökkentésével érhetjük el. Az íjkar végének "keménységét" például az által is csökkenthetjük, ha az a vége felé megfelelően vékonyodik. Egy bizonyos húzáshossz után, természetesen a visszacsapó íj is felkeményedik, amely tartományt, az egyenes íjnál már tárgyalt két ok miatt itt is célszerű elkerülni.

3. Végeelem módszer

A végeelem módszer (Finite Element Method) egy olyan numerikus módszer, mely segítségével az eredetihez nagyon hasonló modellt tudunk létrehozni. A modellt felosztjuk tetszőleges számú csomópontra, illetve a csomópontokat összekötő vonalak által határolt elemekre. Az egyes elemek mentén a keresett attribútum értékét előre felvett paramétereket tartalmazó függvényekkel közelítjük, majd a szomszédos elemek határai mentén adott elvek alapján összeillesztjük.

A végeelem módszer onnan származik, hogy szükség volt megoldást találni a komplex rugalmassági és szerkezeti elemzés problémákra mind polgári mind repüléstechnikai tervezés szempontjából. A kifejlesztése visszavezethető Alexander Hrennikoff (1941) és Richard Courant (1942) munkásságára.

3.1. Végeelem módszer fizikája

Elsőnek is összefoglaljuk a végeelem módszer működéséhez szükséges fizikai fogalmakat, illetve összefüggéseket.[1] A klasszikus mechanikából indulunk ki. Ezen összefüggéseken keresztül juthatunk el az új működését szemléltető megoldásokhoz (statikus, dinamikus), mivel ezeket az egyenleteket analitikus formában nem lehet megoldani, ezért numerikus megoldásra lesz szükségünk.

3.1.1. Newtoni mechanika

Klasszikus mechanikaként is ismert. A testek mozgásának leírásával és az azokat okozó törvényekkel foglalkozik. A mozgások leírására Euklideszi geometriát használ. A testek leírására tömegpontokat használunk. Egy tömegpont jellemezhető: helyvektor, tömeg, sebesség, gyorsulás.

A továbbiakban nézzük a számunkra fontos képleteket és összefüggéseket!

Sebesség: A sebesség az elmozdulás idő szerinti deriváltja.

$$\vec{v} = \frac{d\vec{r}}{dt} = \dot{\vec{r}}$$

Gyorsulás: A gyorsulás a sebesség idő szerinti deriváltja. Ami egyenlő az elmozdulás idő szerinti második deriváltjával.

$$\vec{a} = \frac{d\vec{v}}{dt} = \frac{d^2\vec{r}}{dt^2} = \ddot{\vec{r}}$$

Lendület: A lendület a tömeg és a sebesség szorzata.

$$\vec{p} = m\vec{v} = m\dot{\vec{r}}$$

Newton 2. törvénye: Egy pontszerű test gyorsulása egyenesen arányos a testre ható, a gyorsulással azonos irányú erővel, és fordítottan arányos a test tömegével. Ami egyenlő a lendület idő szerinti deriváltjával.

$$\vec{F} = m\vec{a} = m \frac{d\vec{v}}{dt} = \frac{dm\vec{v}}{dt} = \frac{d\vec{p}}{dt} = m\ddot{\vec{r}}$$

Impulzusmomentum: Egy tömegpont impulzusmomentuma egyenlő a forgásközéppontból a tömegközéppontra mutató vektor és az impulzus vektoriális szorzatával.

$$\vec{L} = \vec{r} \times \vec{p}$$

Forgatónyomaték: A forgatónyomaték az impulzusmomentum idő szerinti deriváltja.

$$\vec{N} = \frac{d\vec{L}}{dt} = \vec{r} \times \frac{d\vec{p}}{dt} = \vec{r} \times \vec{F}$$

Munka (Változó nagyságú és irányú erő munkája): Kezdő pontból (1) a végpontba (2) vezető út munkája F erő hatása alatt:

$$W_{12} = \int_1^2 \vec{F} d\vec{s}$$

Mozgási energia (kinetikus energia): A mozgási energia egy test mechanikai mozgásából eredő munkavégző képessége. Egy test mozgási energiája egyenlő azzal a munkával, amit nyugalmi állapotból kell kifejtsen hogy elérje a kívánt sebességet és forgást.

$$T \equiv \frac{mv^2}{2}$$

Potenciális energia (helyzeti energia): Az az energia, amellyel egy test rendelkezik potenciális erőterben.

$$F = -\nabla V(\vec{r})$$

ahol V a potenciális energia.

A munka és a mozgási energia fogalmából következik, hogy

$$W_{12} = \int_1^2 \vec{F} d\vec{s} = \int_1^2 m \vec{a} d\vec{s} = \int_1^2 m \vec{v} \frac{d\vec{v}}{ds} ds = \int_1^2 m \vec{v} d\vec{v} = m \int_1^2 \vec{v} d\vec{v} =$$

$$\frac{1}{2}mv_2^2 - \frac{1}{2}mv_1^2 = T_2 - T_1$$

A potenciális energia fogalmából következik, hogy

$$W_{12} = \int_1^2 \vec{F} d\vec{s} = V_1 - V_2$$

Az előző két összefüggésből következik a következő:

$$T_1 + V_1 = T_2 + V_2$$

3.1.2. Részecskék rendszere

Mivel a modelleket is csomópontokból és elemekből építjük be, ezért az eddigi összefüggéseket definiálnunk kell a részecskék rendszerében is.

Részecskére ható erő: Egy részecskére ható erőnél megkülönböztethetjük a részecskére ható belső, illetve a külső erőket.

$$F_i = \dot{p}_i$$

$$F_i = \sum_j F_{ji} + F_i^{(e)}$$

A rendszerre ható erő: Összegezzük a rendszerre ható belső, illetve külső erőket.

$$\sum_i F_i = \sum_{i,j} F_{ji} + \sum_i F_i^{(e)} = \sum_{i < j} (F_{ji} + F_{ij}) + \sum_i F_i^{(e)}$$

A kölcsönhatás törvénye (akció-reakció törvénye) miatt az $(F_{ji} + F_{ij})$ kifejezés elhagyható. Ha egy testre egy másik test F erővel hat, akkor a másik test ugyanakkora, de ellenkező irányú erővel hat az előbbire.

$$F_{ij} = -F_{ji}$$

Ezért:

$$\sum_i F_i = \sum_i F_i^{(e)} = \sum_i \dot{p}_i = \frac{d^2}{dt^2} \sum_i m_i r_i$$

A rendszer tömegközéppontja: A tömegközéppont az a pont, mely úgy viselkedik, mintha a rendszer tömege ebbe a pontba lenne koncentrálna.

Egy részekből álló rendszer tömegközéppontjának helye:

$$R = \frac{\sum m_i r_i}{\sum m_i} = \frac{\sum m_i r_i}{M}$$

A tömegközéppont fogalmából következik, hogy

$$M\ddot{R} = \sum_i F_i^{(e)} \equiv F^{(e)}$$

A teljes rendszer lendülete:

$$P = \sum_i p_i = \sum_i m_i \dot{r}_i = M\dot{R}$$

Megfigyelhetjük, hogy ha a rendszer lendületét idő szerint deriváljuk, akkor pontosan $F^{(e)}$ -t kapjuk

$$\dot{P} = M\ddot{R} = F^{(e)}$$

A teljes rendszer impulzus momentuma:

$$L = \sum_i L_i = \sum_i r_i \times p_i$$

Mivel

$$\dot{p}_i = F_i = \sum_j F_{ji} + F_i^{(e)}$$

Ezért, ha a rendszer impulzusát idő szerint deriváljuk, a következőt kapjuk:

$$\dot{L} = \sum_{i,j} r_i \times F_{ji} + \sum_i r_i \times F_i^{(e)}$$

A kölcsönhatás törvénye (akció-reakció törvénye) miatt a következő kifejezés elhagyható:

$$\sum_{i,j} r_i \times F_{ji}$$

Így megkapjuk a rendszer forgatónyomatékát:

$$\dot{L} = \sum_i r_i \times F_i^{(e)} = \sum_i N_i^{(e)} = N^{(e)}$$

A teljes rendszer impulzus momentuma leegyszerűsítve a tömegközéppont segítségével:

Egy részecske helyzete a tömegközéppontban:

$$r_i = r'_i + R$$

Egy a tömegközépponttól r távolságra lévő részecske sebessége:

$$v'_i = \dot{r}'_i$$

Tömegközéppont sebessége:

$$v = \dot{R} = v_i - v'_i$$

Tömegközéppontban lévő részecske sebessége:

$$v_i = v + v'_i$$

És igazak a következő állítások:

$$M = \sum_i m_i \quad \sum_i m_i r_i = 0 \quad \sum_i m_i v_i = 0$$

Ezekből az összefüggésekből következik, hogy

$$L = \sum_i r_i \times p_i = \sum_i (r'_i + R) \times m_i (v'_i + v) = R \times Mv + \sum_i r'_i \times m_i v'_i$$

Munka:

$$W_{12} = \sum_i \int_1^2 F_i ds_i$$

$$W_{12} = T_2 - T_1$$

Kinetikus energia: A tömegközéppont segítségével itt is két darabra tudjuk bontani az egyenletet, egy a tömegközéppontba koncentrált, illetve a tömegközéppont köré koncentrált mozgásra.

$$T = \sum_i \frac{1}{2} m_i v_i^2$$

$$T = \sum_i \frac{1}{2} m_i (v + v'_i) \rightarrow \frac{1}{2} M v^2 + \sum_i \frac{1}{2} m_i v_i'^2$$

Potenciális energia: A rendszer potenciális energiájának kiszámolásánál figyelembe kell vennünk a rendszerre ható belső, illetve külső erőket!

Külső erők:

$$F_i^{(e)} = -\nabla_i V_i$$

$$\sum_i \int_1^2 F_i^{(e)} ds_i = -\sum_i \int_i^2 \nabla_i V_i ds_i = -\sum_i (V_i)_2 + \sum_i (V_i)_1$$

Belső erők:

$$F_{ji} = -\nabla_i V_{ij}$$

$$\sum_{i,j} \int_1^2 F_{ji} ds_i = \frac{1}{2} \sum_{i,j} \int_1^2 [F_{ji} ds_i + F_{ij} ds_j] = -\frac{1}{2} \sum_{i,j} \int_1^2 [\nabla_i V_{ij} ds_i + \nabla_j V_{ij} ds_j]$$

A kölcsönhatás törvénye miatt igazak a következő állítások:

$$V_{ij} = V_{ij}(|r_i - r_j|) \quad \nabla_i V_{ij} = -\nabla_j V_{ij} = \nabla_{ij} V_{ij} \quad ds_i - ds_j = dr_{ij}$$

Ezért:

$$\sum_{i,j} \int_1^2 F_{ji} ds_i = -\frac{1}{2} \sum_{i,j} \int_1^2 \nabla_{ij} V_{ij} dr_{ij} = -\frac{1}{2} \sum_{i,j} (V_{ij})_2 + \frac{1}{2} \sum_{i,j} (V_{ij})_1$$

Vagy:

$$\sum_{i,j} \int_1^2 F_{ji} ds_i = (V^{(I)})_1 - (V^{(I)})_2$$

Ahol belső erőkkel meghatározott potenciális energia:

$$V^{(I)} \equiv \frac{1}{2} \sum_{i,j} V_{ij}$$

Így a rendszer potenciális energiája:

$$V \equiv V^{(e)} + V^{(I)} = \sum_i V_i + \frac{1}{2} \sum_{i,j} V_{ij}$$

3.1.3. Mozgás egyenletek leírása általános koordinátákkal

Számtalan mechanikai probléma megoldása lényegesen egyszerűsödik, ha derék-szögű koordináta-rendszer koordinátái helyett más koordinátákat választunk. Ha az n számú anyagi pontból álló mechanikai rendszer r számú kényszerfeltétele holonom, vagyis $f_k(x_1, \dots, x_{3n}, t) = 0$ (ahol $k = 1, 2, \dots, r$) alakban adható meg, a $3n$ derékszögű koordináta közül csupán $3n - r$ független, azaz a rendszer szabadsági foka: $f = 3n - r$. Az ilyen rendszer helyzetét f számú, egymástól független q_1, q_2, \dots, q_f adat egyértelműen meghatározza.

$$r_i = r_i(q_1, q_2, q_3, \dots, t)$$

A derékszögű koordináták helyett olyan, a rendszer f szabadsági fokával megegyező számú ún. általános koordinátát (q_1, \dots, q_f) vezetünk be, amelyek használatakor a kényszerfeltételek már eleve teljesülnek.

pl.:

Az (x, y, z) koordináták helyett választható általános koordináták a θ és ϕ polár-koordináták:

$$x = c \sin\theta \cos\phi$$

$$y = c \sin\theta \sin\phi$$

$$z = c \cos\theta$$

Ismerjük tehát a mozgás egyenletét:

$$m_i \ddot{r}_i = F_i = F_i^{(e)} + \sum_j F_{ji}$$

ahol a $F_i^{(e)}$ a rendszer ható külső erőket, $\sum_j F_{ji}$ a rendszerre ható belső erőket jelenti.

Hogyan tudnánk átalakítani a mozgás egyenletét az általánosított koordináták segítségével?

Megoldás: Lagrange függvény!

3.1.4. D'Alambert elv

A Lagrange függvény értelmezéséhez a D'Alambert elvből kell kiindulnunk.

A részecskék mindegyik mozgást végez.

Virtuális elmozdulás:

$$r_i \rightarrow r_i + \delta r_i \quad q_j \rightarrow q_j + \delta q_j$$

$$\delta r_i = \sum_j \frac{\partial r_i}{\partial q_j} \delta q_j$$

A newtoni egyenletet egyensúlyi egyenlet alakjában írjuk fel:

$$F_i = \dot{p}_i \rightarrow F_i - \dot{p}_i = 0$$

$$F_i = F_i^{(a)} + f_i$$

$F_i^{(a)}$: alkalmazott erő

$$F_i^{(a)} = F_i^{(a)}(r_1, r_2, \dots, r_i, \dots, r_N, t)$$

f_i : kényszererő

A kényszererő általában nem végez munkát:

$$\text{- mozgás merőleges az erőre: } f_i \delta r_i = 0$$

- kivétel: súrlódás

Így tehát:

$$F_i^{(a)} + f_i - \dot{p}_i = 0$$

Megszorozzuk az egyenletet δr_i -vel, illetve összegezzük az egészet i szerint.

$$\sum_i (F_i^{(a)} - \dot{p}_i) \delta r_i = 0$$

A kényszererő kiesik, mivel:

$$f_i \delta r_i = 0$$

Mivel a virtuális elmozdulás:

$$\delta r_i = \sum_j \frac{\partial r_i}{\partial q_j} \delta q_j$$

Ezért:

$$\sum_i \left(F_i^{(a)} \sum_j \frac{\partial r_i}{\partial q_j} \delta q_j - \dot{p}_i \sum_j \frac{\partial r_i}{\partial q_j} \delta q_j \right) = 0$$

Igaz, hogy:

$$\sum_i F_i \sum_j \frac{\partial r_i}{\partial q_j} \delta q_j = \sum_j Q_j \delta q_j \rightarrow Q_j \equiv \sum_i F_i \frac{\partial r_i}{\partial q_j}$$

Q_j : általánosított erő

$$\sum_j Q_j \delta q_j - \sum_i \dot{p}_i \sum_j \frac{\partial r_i}{\partial q_j} \delta q_j = 0$$

Mivel $\dot{p}_i = m_i \ddot{r}_i$, ezért:

$$\begin{aligned} \sum_j Q_j \delta q_j - \sum_{i,j} m_i \ddot{r}_i \frac{\partial r_i}{\partial q_j} \delta q_j &= 0 \\ \ddot{r}_i \frac{\partial r_i}{\partial q_j} &\rightarrow \frac{d}{dt} \left[\frac{\partial}{\partial \dot{q}_j} \left(\frac{v_i^2}{2} \right) \right] - \frac{\partial}{\partial q_j} \left(\frac{v_i^2}{2} \right) \end{aligned}$$

$$\begin{aligned} \sum_j Q_j \delta q_j - \sum_{i,j} m_i \left\{ \frac{d}{dt} \left[\frac{\partial}{\partial \dot{q}_j} \left(\frac{v_i^2}{2} \right) \right] - \frac{\partial}{\partial q_j} \left(\frac{v_i^2}{2} \right) \right\} \delta q_j &= 0 \\ \sum_j Q_j \delta q_j - \sum_j \left\{ \frac{d}{dt} \left[\frac{\partial}{\partial \dot{q}_j} \sum_i \frac{m_i v_i^2}{2} \right] - \frac{\partial}{\partial q_j} \sum_i \frac{m_i v_i^2}{2} \right\} \delta q_j &= 0 \\ T &\equiv \sum_i \frac{m v_i^2}{2} \\ \sum_j \left\{ \left[\frac{d}{dt} \left(\frac{\partial T}{\partial \dot{q}_j} \right) - \frac{\partial T}{\partial q_j} \right] - Q_j \right\} \delta q_j &= 0 \end{aligned}$$

D'Alembert elve értelmében a mozgó testre ható erők a tehetetlenségi erővel együtt egyensúlyban vannak. Ezt az egyensúlyi állapotot statikai egyensúlytól megkülönböztetve kinetikai egyensúlynak nevezzük.

3.1.5. Lagrange függvény

A Lagrange-függvény a fizikai rendszerek állapotát jellemző mennyiség. A mechanikában a kinetikus (mozgási) és a potenciális energia különbsége adja a Lagrange-függvényt.

$$\begin{aligned} \frac{d}{dt} \left(\frac{\partial L}{\partial \dot{q}_j} \right) - \frac{\partial L}{\partial q_j} &= 0 \\ L(q, \dot{q}, t) &\equiv T - V \end{aligned}$$

ahol T a rendszer kinetikus, V pedig a potenciális energiája.

q_j : általános koordináta

\dot{q}_j : általános koordináta időszerinti deriváltja

$$\sum_j \left\{ \left[\frac{d}{dt} \left(\frac{\partial T}{\partial \dot{q}_j} \right) - \frac{\partial T}{\partial q_j} \right] - Q_j \right\} \delta q_j = 0$$

Ha q_j -k függetlenek:

$$\frac{d}{dt} \left(\frac{\partial T}{\partial \dot{q}_j} \right) - \frac{\partial T}{\partial q_j} = Q_j$$

Konzervatív erőterben:

$$F_i = -\nabla_i V$$

Általánosított erő:

$$\begin{aligned} Q_j &\equiv \sum_i F_i \frac{\partial r_i}{\partial q_j} = - \sum_i \nabla_i V \frac{\partial r_i}{\partial q_j} = - \sum_i \frac{\partial V}{\partial r_i} \frac{\partial r_i}{\partial q_j} = - \frac{\partial V}{\partial q_j} \\ \frac{d}{dt} \left(\frac{\partial T}{\partial \dot{q}_j} \right) - \frac{\partial T}{\partial q_j} &= - \frac{\partial V}{\partial q_j} \\ \frac{d}{dt} \left(\frac{\partial T}{\partial \dot{q}_j} \right) - \frac{\partial (T - V)}{\partial q_j} &= 0 \end{aligned}$$

Tegyük fel, hogy $V = V(q_j, \dot{q}_j, t)$ nem tartalmazza a \dot{q}_j -t, azaz $\frac{\partial V}{\partial \dot{q}_j} = 0$ ekkor írhatjuk, hogy

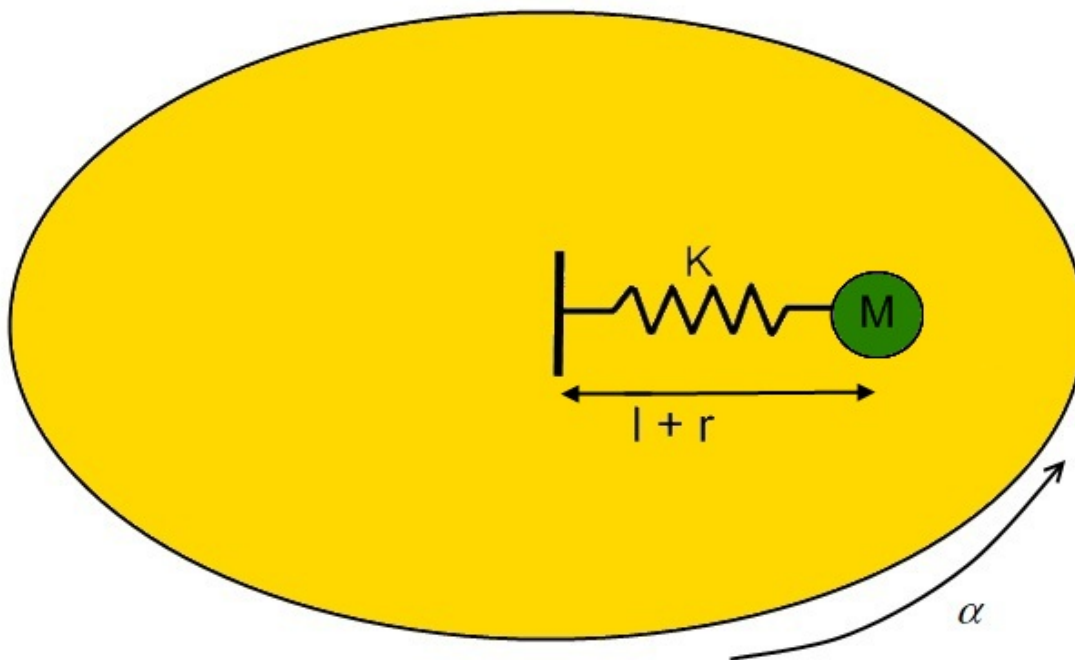
$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{q}_j} \right) - \frac{\partial L}{\partial q_j} = 0$$

ahol

$$L = T(q_j, \dot{q}_j, t) - V(q_j, t)$$

3.1.6. Példa a Lagrange függvény alkalmazhatóságára

Nézzünk egy egyszerű példát a Lagrange függvény alkalmazhatóságára.



4. ábra. Példa rezgőmozgásra

A fent megadott adatok segítségével az x, y -t a következőképpen tudjuk átalakítani:

$$x = (l + r) \cos \alpha t$$

$$y = (l + r) \sin \alpha t$$

Ki tudjuk számítani a rendszer potenciális és kinetikus energiáját:

$$T = \frac{m}{2} \{ \dot{x}^2 + \dot{y}^2 \} = \frac{m}{2} \{ \dot{r}^2 + (l + r)^2 \alpha^2 \}$$

$$V = \frac{K}{2}r^2$$

Ezekből következnek a következők:

$$L = \frac{m}{2} \left\{ \dot{r}^2 + (l+r)^2 \alpha^2 \right\} - \frac{K}{2}r^2$$

A rendszer Lagrange egyenlete:

$$\begin{aligned} \frac{d}{dt} \left(\frac{\partial L}{\partial \dot{r}} \right) - \frac{\partial L}{\partial r} &= m\ddot{r} - m\alpha^2(l+r) + Kr = 0 \\ m\ddot{r} + (K - m\alpha^2) \left(r - \frac{m\alpha^2 l}{K - m\alpha^2} \right) &= 0 \end{aligned}$$

Ha $K > m\alpha^2$, akkor harmonikus mozgást kapunk, ahol $\omega = \sqrt{\frac{K-m\alpha^2}{m}}$

Az ingás központját elmozdítja: $\frac{m\alpha^2 l}{K-m\alpha^2}$

Ha $K < m\alpha^2$, akkor a mozgás exponenciális lesz.

Ha $K = m\alpha^2$, akkor a sebesség állandó lesz.

3.1.7. Rugalmasságtan

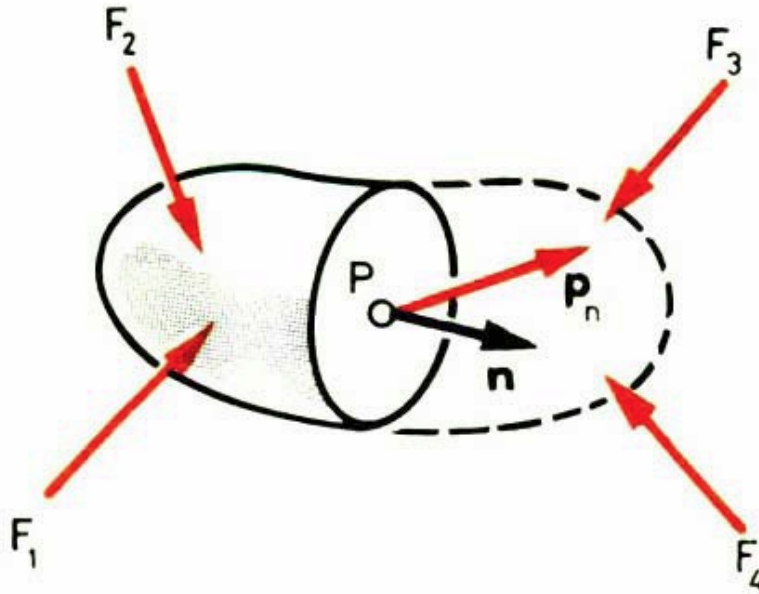
A rugalmasságtan a testek terhelésekre adott válaszát vizsgálja. A testet homogénnek, izotópnak és lineárisan rugalmasnak tekintjük, továbbá terhelés hatására deformációt szenved, így rugalmassági tényezők köthetők a terhelés előtti geometriához.

Feszültségvektor: P_n feszültségvektor a testben terhelés hatására fellépő, felület mentén megoszló belső erőrendszer sűrűségvektora (intenzitásvektora).

Vizsgáljuk a test állapotát a P pontban, amin át egy n normálisú síkkal a testet két részre vágjuk. A metszet egyik oldalán lévő részre a másik oldalán lévő rész által kifejtett hatást általános megoszló erőrendszerként modellezünk, e megoszló erőrendszer iránya és intenzitása a P pontban: a P -hez és az n normálisához tartozó feszültségvektor iránya és nagysága.

Feszülstégtenzor: σ feszülstégtenzor segítségével bármely n irányhoz számíthatjuk ki a megfelelő feszültségvektort:

$$P_n = \sigma n$$



5. ábra. Ketté vágott test [4]

Valamely (x,y,z) koordináta rendszerben a feszültség tenzor mátrixa:

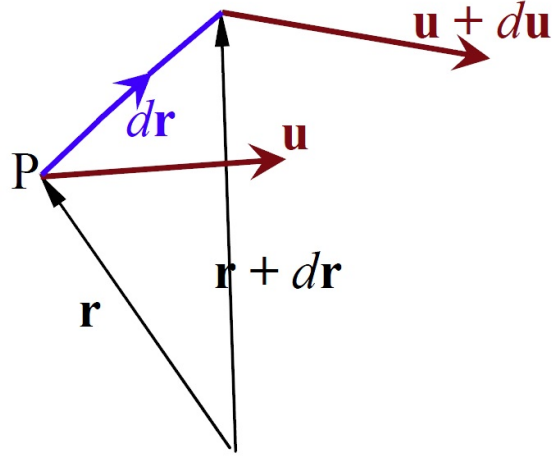
$$\sigma = \begin{bmatrix} \sigma_x & \tau_{xy} & \tau_{xz} \\ \tau_{yx} & \sigma_y & \tau_{yz} \\ \tau_{zx} & \tau_{zy} & \sigma_z \end{bmatrix}$$

Ahol az egyes sorok rendre az x , az y , és a z normális metszetekhez tartozó feszültségvektorok komponenseit tartalmazzák:

$$P_x = \begin{bmatrix} \sigma_x \\ \tau_{yx} \\ \tau_{zx} \end{bmatrix}; \quad P_y = \begin{bmatrix} \tau_{xy} \\ \sigma_y \\ \tau_{zy} \end{bmatrix}; \quad P_z = \begin{bmatrix} \tau_{xz} \\ \tau_{yz} \\ \sigma_z \end{bmatrix}$$

Alakváltozás: A vizsgált test P pontja (melynek helyvektora r) elmozdul, eltolódásvektora u ; a P ponthoz közeli, $r + dr$ helyvektorú pont szintén elmozdul, eltolódása $u + du$:

$$u = \begin{bmatrix} u \\ v \\ w \end{bmatrix} \quad u + du = \begin{bmatrix} u + du \\ v + dv \\ w + dw \end{bmatrix}$$



6. ábra. Alakváltozás [4]

Az u eltolódásmező gradienstensorával bármely dr esetén meghatározható, hogy mekkora lesz az r és a $r+dx$ pontok eltolódása közötti du különbség:

$$du = \begin{bmatrix} du \\ dv \\ dw \end{bmatrix} = \begin{bmatrix} \frac{\partial u}{\partial x} & \frac{\partial u}{\partial y} & \frac{\partial u}{\partial z} \\ \frac{\partial v}{\partial x} & \frac{\partial v}{\partial y} & \frac{\partial v}{\partial z} \\ \frac{\partial w}{\partial x} & \frac{\partial w}{\partial y} & \frac{\partial w}{\partial z} \end{bmatrix} \begin{bmatrix} dx \\ dy \\ dz \end{bmatrix} = B^T dr$$

Az r és az $r+dr$ helyvektorú pontok eltolódása két ok miatt tér el egymástól:

- a P pont környezete deformálódik (du^{def})
- a P környezete merevtest szerkezetű elfordulást végez (du^{rot})

du^{def} és du^{rot} külön-külön is kifejezhetők, ha a B gradienstenzort szétválasztjuk szimmetrikus és aszimmetrikus összetevőjére ($B = B^{def} + B^{rot}$):

$$du = (B^{def} + B^{rot})^T dx = (B^{def})^T dr + (B^{rot})^T dx = du^{def} + du^{rot}$$

Alakváltozástenzor: ϵ az eltolódásgradiens-tenzor szimmetrikus része, P környezetének deformációját fejezi ki:

$$\epsilon = B^{def} = \begin{bmatrix} \frac{\partial u}{\partial x} & \frac{1}{2} \left(\frac{v}{x} + \frac{u}{y} \right) & \frac{1}{2} \left(\frac{w}{x} + \frac{u}{z} \right) \\ \frac{1}{2} \left(\frac{u}{y} + \frac{v}{x} \right) & \frac{\partial v}{\partial y} & \frac{1}{2} \left(\frac{w}{y} + \frac{v}{z} \right) \\ \frac{1}{2} \left(\frac{u}{z} + \frac{w}{x} \right) & \frac{1}{2} \left(\frac{v}{z} + \frac{w}{y} \right) & \frac{\partial w}{\partial z} \end{bmatrix} = \begin{bmatrix} \epsilon_x & \frac{y_{xy}}{2} & \frac{y_{xz}}{2} \\ \frac{y_{xy}}{2} & \epsilon_y & \frac{y_{yz}}{2} \\ \frac{y_{xz}}{2} & \frac{y_{yz}}{2} & \epsilon_z \end{bmatrix}$$

Statikai egyenletek: A Cauchy-egyenletek a test belsejében lévő elemi hasábok egyensúlyát fejezik ki,

összefüggést teremtenek a test belsejében ébredő feszültségek és a testre ható tömegerők között:

$$\frac{\partial \sigma_x}{\partial x} + \frac{\partial \tau_{xy}}{\partial y} + \frac{\partial \tau_{xz}}{\partial z} + g_x = 0$$

$$\frac{\partial \tau_{yx}}{\partial x} + \frac{\partial \sigma_y}{\partial y} + \frac{\partial \tau_{yz}}{\partial z} + g_y = 0$$

$$\frac{\partial \tau_{zx}}{\partial x} + \frac{\partial \tau_{zy}}{\partial y} + \frac{\partial \sigma_z}{\partial z} + g_z = 0$$

Virtuális elmozdulás: Virtuális elmozdulás rendszernek nevezzük azt, amikor a szerkezet egy tetszőleges geometriailag lehetséges elmozdulásrendszerének változatlan geometriai peremfeltételek mellett képezett differenciálisan kicsiny megváltoztatása. Egy erőrendszer akkor és csak akkor statikailag lehetséges, ha bármely virtuális elmozdulásrendszeren végzett munkája zérus:

$$\delta W_{külső} + \delta W_{belső} = 0$$

ahol a tényleges külső erőknek a virtuális elmozdulásokon végzett munkája:

$$\delta W_{külső} = f^T \delta e + \int_{(S)} q^T \delta u dS + \int_{(V)} g^T \delta u dV$$

és a tényleges feszültségeknek a virtuális alakváltozásokon való belső munkája:

$$\delta W_{belső} = - \int_{(V)} \sigma^T \delta \epsilon dV$$

Hook törvény: Azt fejezi ki, hogy valamely testre ható erő és az általa előidézett deformáció egyenes arányban vannak egymással.

$$\sigma_{ij} = \sum_{kl} c_{ijkl} \epsilon_{kl}$$

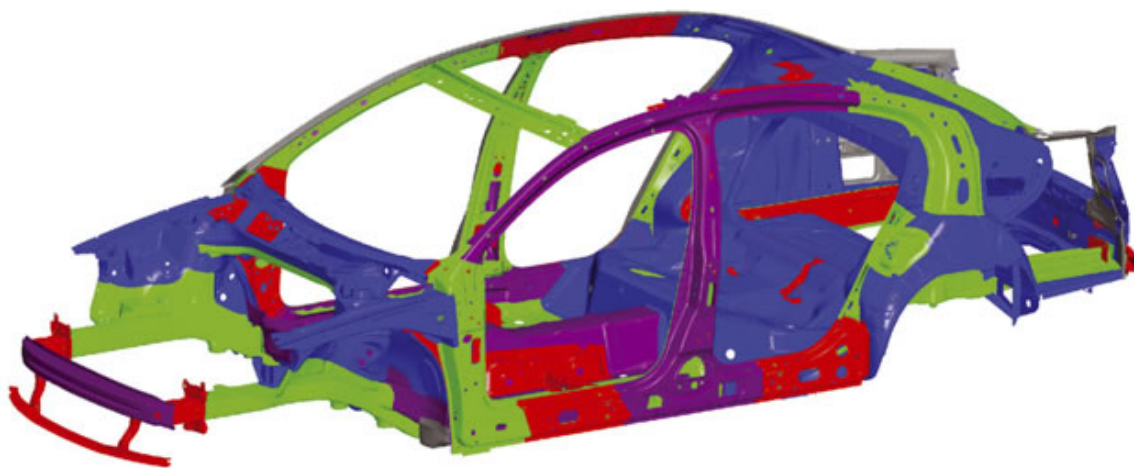
3.2. Végeelem programozás

A végeelem módszer sikerének a titka, hogy a teljes numerikus folyamat implementálható a számítógépre. Ezt az tette lehetővé, hogy mára már olyan hardver eszközök állnak rendelkezésünkre, melyek a nagy modellek feldolgozását is lehetővé teszik. Emellett olyan professzionális termékek jelentek meg, amelyek a számítógéppel segített műszaki tervezés és elemzés fázisait komplex módon tartalmazzák. E szoftverek segítségével a mérnök párbeszédés formában képes kezelni a geometriát, létre tudja hozni a végeelem

modellt, meg tudja oldani az előállított algebrai feladatot és végül hatékonyan, szemléletes formában ki tudja értékelni az eredményeket. Mindez a számítógép virtuális világában történik anélkül, hogy egyáltalán készült volna valamiféle fizikailag megfogható, mérhető darab vagy prototípus. Ráadásul az eredmények birtokában a folyamat újratekeshető azokkal a módosításokkal, melyektől javulást remélünk. Továbbiakban nézzük meg a végeselem módszer feldolgozási folyamat főbb lépéseit.

3.2.1. Előfeldolgozás

Ebben a fázisban történik a véges elem háló előállítás. Maga az elemháló felépítése lehet négyzetekből vagy háromszögekből. Ez a folyamat a legmunkaigényesebb, bár számos algoritmus segítheti a mérnök munkáját. A feladat szempontjából lényeges részeket sűrűbb, míg a nem annyira lényegesebbeket ritkább felosztással valósítják meg. A kész végeselem háló mérete és minősége nagyban befolyásolja az eredményen



7. ábra. Volkswagen karosszéria végeselem modelle[5]

pontosságát, valóságosságát. Míg a 80-as években néhány ezer elem, csomópont alkotott egy modellt, mára több millió elemet is használnak, amikhez nincs szükség szuper számítógépekre.

3.2.2. Peremfeltételek definiálása

A kiindulási paramétereket és adatokat kell megadni ebben a részben. Ilyen paramétereke például a szerkezeti terheltség, koncentrált erők, felületi terhelés, térfogati terhelés, időfüggő terhelés, gerjesztés vagy akár folyadékáramlás.

3.2.3. Algebrai egyenlet megoldása

Ez a fázis a legjelentősebb, segítséget ad a mérnöknek, hiszen több millió műveletet hajt végre a számítógép, melyre az ember nem lenne képes. A szoftver előállítja az algebrai egyenlet mátrixait, és azt az algebrai egyenletet, amely approximáció eredménye, amit megoldva közelítő eredményt kapunk az adott fizikai folyamatra. A futási idő erősen függ a feladat típusától és a modell méretétől. Például a "crash" alkalmazások, ahogy gépjárművek viselkedését vizsgálják a töréskeresztek alatt. Ezeknél a teszteknel 500 ezer feletti elem szám az általános, mivel ez egy időfüggő folyamat ezért a számítás összetettebb folyamat. Fontos a precizitás, mivel életek múlhatnak a számítások pontosságán.

3.2.4. Eredmények kiértékelése

Ez a lépés az utolsó a végeelem feldolgozásban. Ennél a fázisnál is nélkülözhetetlen a számítógép, a sok számítás miatt. A vizuális megjelenítés fontos az értékelés és lehetséges módosítások szempontjából. A kapott paraméterek érték tartományát felosztva és hozzá színt rendelve egyszerűen lehet szemléltetni a modellben fellépő feszültségeket. A felhasználó így könnyen eldöntheti, hogy a feszültség eloszlás megfelelő-e vagy szükséges módosítás a modellen. Ebben a fázisban lényeges a gyors és hatékony kiértékelés.

4. Végeelem módszer megvalósítása Javaban [3]

Elsőnek is ismertetjük, hogy mik is pontosan a végeelem analízis főbb lépései. Majd bemutatunk két végeelem egyenletet megoldó módszert, a Galerkin módszert és az úgynevezett variációs megközelítést.

4.1. Végeelem analízis főbb lépései

1. Tartomány diszkretizálás.

A véges elemek a csomópontoknál csatlakoznak, ezek felosztásra kerülnek különböző megoldástartományokra. A nagy mennyiségű adat miatt, a végeelem hálót egy preprocessorprogram generálja. A háló csomóponti koordinátákból és elemkapcsolásokból áll.

2. Interpolációs funkciók használata.

Az elemen keresztül interpolálásra kerülnek a mezőváltozók. Az interpolációs funkciók polinomokat választanak ki. Egy polinom értéke függ az elemet felépítő csomópontoktól.

3. Elemmátrixok és vektorok kiszámítása.

A mátrix egyenletek kiszámítása utána tudjuk meg, hogy az ismeretlen funkciójú csomóponti értékek milyen ismert paraméterekkel kapcsolhatóak össze. Erre különböző számítási módok használatosak, ilyen például a variációs megközelítés és a Galerkin módszer is.

4. Elemegyenletek összegyűjtése.

Az összegyűjtött elemegyenletek egyesítésével létre jön a globális egyenletrendszer. Az egyenletrendszer megoldása előtt megadásra kerülnek a rendszer kerületi feltételei.

5. Globális egyenletrendszer megoldása.

A globális egyenletrendszer jellemzően szórványos, szimmetrikus és pozitív-definit. Közvetlen és ismétlődő módszerek alkalmazhatóak a megoldására. A keresett függvény csomóponti értékei adják a rendszer megoldását.

6. További eredmények kiszámítása.

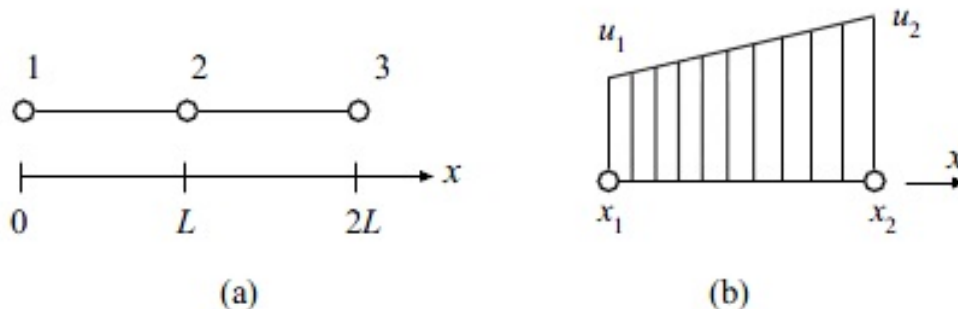
Sok esetben további paramétereket kell kiszámítani.

4.2. A végeelem egyenlet

A végeelem egyenlet megfogalmazására több átalakítási mód is lehetséges. Mi a Galerkin módszert és a variációs megközelítést mutatjuk be. Ha a probléma fizikai megvalósítása differenciál egyenletként ismert, akkor a Galerkin módszer használatos. Ha a fizikai probléma funkcionálisan minimalizált, akkor általában

a variációs formula alkalmazandó.

Mindkét módszert bemutatjuk a következő egyszerű egydimenziós példán keresztül:



8. ábra. Két egydimenziós lineáris elem (a) és egy elemen belüli függvény interpoláció(b)

4.2.1. Galerkin módszer

A példában szereplő két lineáris végelem figyelembe vételével a következő differenciálegyenletet kell megoldanunk:

$$a \frac{d^2 u}{dx^2} + b = 0, \quad 0 \leq x \leq 2L,$$

a következő kerületi feltételek figyelembe vételével:

$$u|_{x=0} = 0,$$

$$a \frac{du}{dx} \Big|_{x=2L} = R,$$

ahol $u=u(x)$ egy ismeretlen függvény.

Nézzünk meg egy darab végelemet (6. ábra (b)). Az elem két darab csomópontból áll, az $u(x)$ függvény közelítése a következőképpen fog kinézni:

$$u = N_1 u_1 + N_2 u_2 = [N] \{u\},$$

$$[N] = [N_1 \ N_2],$$

$$\{u\} = \{u_1 \ u_2\},$$

ahol N_i az úgynevezett alak funkció

$$N_1 = 1 - \frac{x - x_1}{x_2 - x_1},$$

$$N_2 = \frac{x - x_1}{x_2 - x_1},$$

amelyek $u(x)$ függvény interpolációjára használatosak. Az u_1 és u_2 csomóponti értékek olyan ismeretlenek, amelyeket a diszkrét globális egyenletrendszer fog meghatározni. Ha a csomóponti értékekkel és alaki függvényekkel kifejezett u -t behelyettesítjük a differenciálegyenletbe, akkor a következő alakot kapjuk

$$a \frac{d^2}{dx^2} [N] \{u\} + b = \psi,$$

ahol ψ egy nem nulla maradék. A Galerkin módszer maradékminimalizálást tesz lehetővé, a fenti egyenlet az adott elem szerint integrálva, az alak függvények és a feltételek figyelembe vételével a következőképpen fog kinézni:

$$\int_{x_1}^{x_2} [N]^T a \frac{d^2}{dx^2} [N] \{u\} dx + \int_{x_1}^{x_2} [N]^T b dx = 0,$$

az egyenlet részenkénti integrálása a differenciálegyenlet diszkrét formájához vezet

$$\int_{x_1}^{x_2} \left[\frac{dN}{dx} \right]^T a \left[\frac{dN}{dx} \right] dx \{u\} - [N]^T b dx - \begin{Bmatrix} 0 \\ 1 \end{Bmatrix} a \frac{du}{dx} \Big|_{x=x_2} + \begin{Bmatrix} 1 \\ 0 \end{Bmatrix} a \frac{du}{dx} \Big|_{x=x_1} = 0,$$

$$[k] \{u\} = \{f\},$$

$$k = \int_{x_1}^{x_2} \left[\frac{dN}{dx} \right]^T a \left[\frac{dN}{dx} \right] dx,$$

$$\{f\} = \int_{x_1}^{x_2} [N]^T b dx + \begin{Bmatrix} 0 \\ 1 \end{Bmatrix} a \frac{du}{dx} \Big|_{x=x_2} + \begin{Bmatrix} 1 \\ 0 \end{Bmatrix} a \frac{du}{dx} \Big|_{x=x_1},$$

Szilárdtest mechanikában $[k]$ -t merevségmátrixnak, az $\{f\}$ -t pedig tehervektornak nevezik. A szóban forgó két véges elem esetén a merevség a mátrixok és a tehervektorok könnyen kiszámolhatóak:

$$[k_1] = [k_2] = \frac{a}{L} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}$$

$$\{f_1\} = \frac{bL}{2} \begin{Bmatrix} 1 \\ 1 \end{Bmatrix}, \quad \{f_2\} = \frac{bL}{2} \begin{Bmatrix} 1 \\ 1 \end{Bmatrix} + \begin{Bmatrix} 0 \\ R \end{Bmatrix}$$

A fenti kapcsolatok adják a végeselem egyenleteit a két különálló végeselemnek. A két elemmel és három csomóponttal rendelkező tartománynak a globális egyenletrendszere a végeselem egyenletekből származtatható. A mi esetünkben az elemek a csomópontoknál globálisan hatnak egymásra. Az összekapcsolt globális egyenletrendszer:

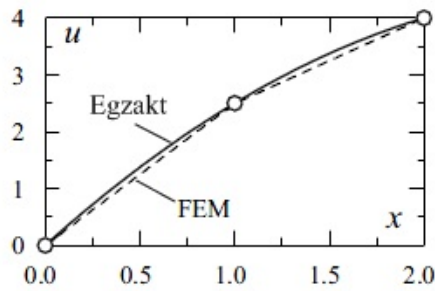
$$\frac{a}{L} \begin{bmatrix} 1 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 1 \end{bmatrix} \begin{Bmatrix} u_1 \\ u_2 \\ u_3 \end{Bmatrix} = \frac{bL}{2} \begin{Bmatrix} 1 \\ 2 \\ 1 \end{Bmatrix} + \begin{Bmatrix} 0 \\ 0 \\ R \end{Bmatrix},$$

A peremfeltételek alkalmazása után $u(x=0) = 0$, a globális egyenletrendszer végső megjelenési formája:

$$\frac{a}{L} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 2 & -1 \\ 0 & -1 & 1 \end{bmatrix} \begin{Bmatrix} u_1 \\ u_2 \\ u_3 \end{Bmatrix} = \frac{bL}{2} \begin{Bmatrix} 0 \\ 2 \\ 1 \end{Bmatrix} + \begin{Bmatrix} 0 \\ 0 \\ R \end{Bmatrix},$$

Amikor érvényes az $u_1 = 0$ peremfeltétel, akkor az egyenletrendszer mátrixának első sorába nullák kerülnek, ugyanúgy ezt történik az egyenlet jobb oldalával is, nullák kerülnek a mátrix első oszlopába is, illetve a főátlóba egy egységérték kerül. Az u_i csomóponti értékek lesznek a lineáris algebrai egyenletrendszer megoldásának eredményei. Az u értéke a végelemen belül bármely pontnál kiszámolható az alaki függvények segítségével.

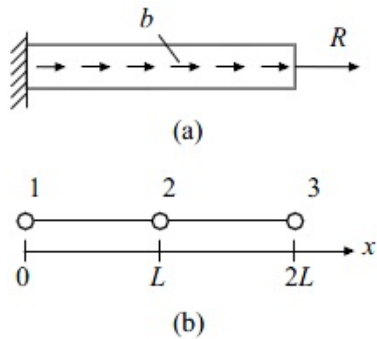
A differenciálegyenlet végelem megoldása ($a = 1$, $b = 1$, $L = 1$, $R = 1$):



9. ábra. A végelem megoldás és az egzakt megoldás összehasonlítása

A megoldás egy másodfokú függvény. Egy pontosabb megoldást úgy tudunk kapni, hogy növeljük az egyszerű elemek számát vagy bonyolultabb felépítésű elemeket használunk.

4.2.2. Variációs megközelítés



10. ábra. Egydimenziós rúd, ami egy elosztott terhelésnek, illetve egy koncentrált tehernek van kitéve

Itt is az előzőekben felvázolt differenciálegyenletből indulunk ki, a kerületi feltételek, illetve egy $a=EA$ paraméter figyelembe vételével. Az $a=EA$ paraméter az egydimenziós rúd feszültségét írja le, A : tartomány keresztmetszete, E : valamilyen anyagnak a Young-modulusa. Az egydimenziós modellt egy elosztott terhelésnek (b) és egy koncentrált tehernek (R) vetjük alá. Ilyen és ehhez hasonló problémák oldhatóak meg a variációs megközelítés segítségével.

A differenciálegyenletünk így fog kinézni a változtatások után:

$$\Pi = \int_L \frac{1}{2} a \left(\frac{du}{dx} \right)^2 dx - \int_L b u dx - R u|_{x=2L},$$

$$u|_{x=0} = 0$$

Az u alakfüggvény figyelembe vételével írhatjuk:

$$\Pi_e = \int_{x_1}^{x_2} \frac{1}{2} a u^T \left[\frac{dN}{dx} \right]^T \left[\frac{dN}{dx} \right] u dx - \int_{x_1}^{x_2} u^T [N]^T b dx - u^T \begin{Bmatrix} 0 \\ R \end{Bmatrix}$$

A Π minimum feltétele:

$$\delta \Pi = \frac{\partial \Pi}{\partial u_1} \delta u_1 + \dots + \frac{\partial \Pi}{\partial u_n} \delta u_n = 0$$

ami ekvivalens

$$\frac{\partial \Pi}{\partial u_i} = 0, \quad i = 1..n$$

A Π u_i -k szerinti differenciálisa végül a következő egyen súlyi végeselem egyenletre vezet :

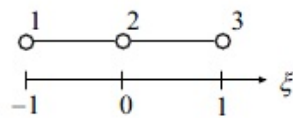
$$\int_{x_1}^{x_2} \left[\frac{dN}{dx} \right]^T EA \left[\frac{dN}{dx} \right] dx \{u\} - \int_{x_1}^{x_2} [N]^T b dx - \begin{Bmatrix} 0 \\ R \end{Bmatrix}$$

Ez a kifejezés egybeesik azzal az egyenlettel, amit a Galerkin módszernél kaptunk.

4.2.3. Példa alakfüggvény meghatározására

Nézzük a három csomóponttal rendelkező egydimenziós, másodfokú elemre fennálló alak függvényeket, a lokális koordináta-rendszer segítségével.

Megoldás:



11. ábra. Három csomóponttal rendelkező egydimenziós másodfokú elem

Az alak függvényekkel bármely terület kifejezhető az elemen belül:

$$u(\xi) = \sum N_i u_i, \quad i = 1, 2, 3.$$

A közelítő függvények minden csomópontnál egyenlők kell, hogy legyenek a csomóponti értékekkel

$$u(-1) = u_1, \quad u(0) = u_2, \quad u(1) = u_3,$$

Mivel az elem három csomópontból áll, ezért az alak függvények másodfokú polinomok lehetnek. Az N_1 a következőképpen írható le:

$$N_1 = \alpha_1 + \alpha_2 \xi + \alpha_3 \xi^2$$

α_i ismeretlen együtthatók a következő egyenletrendszerből nyerhetők ki:

$$N_1(-1) = \alpha_1 - \alpha_2 + \alpha_3 = 1,$$

$$N_1(0) = \alpha_1 = 0,$$

$$N_1(1) = \alpha_1 + \alpha_2 + \alpha_3 = 0$$

A megoldás: $\alpha_1 = 0, \alpha_2 = -1/2, \alpha_3 = 1/2$. Így, az alak függvény N_1 :

$$N_1 = -\frac{1}{2}\xi(1 - \xi)$$

hasonlóan határozzuk meg az N_2, N_3 alak függvényeket:

$$N_2 = 1 - \xi^2,$$

$$N_3 = \frac{1}{2}\xi(1 + \xi).$$

Elkerülhetjük az egyenletrendszer megoldását, ha leírjuk az alak függvény keresett formáját a következőképpen:

$$N_i = a_1(a_2 + \xi)(a_3 + \xi)$$

a_1, a_2 és a_3 együtthatók abból a feltételből határozhatóak meg, hogy az alak függvények a saját csomópontjuknál azzal egyenlő értéket, minden más csomópontnál nulla értéket kapnak. Például könnyű megkapnunk N_1 alak függvény a_2 és a_3 együtthatóját, ha 0-val egyenlővé tesszük a kifejezéseket:

$$\xi = 0 : a_2 + 0 = 0,$$

$$\xi = 1 : a_3 + 1 = 0.$$

Az $a_2 = 0, a_3 = -1$, így az N_1 alak függvény megjelenési formája:

$$N_1 = a_1 \xi(-1 + \xi)$$

Az a_1 együttható a feltétel alapján meghatározva:

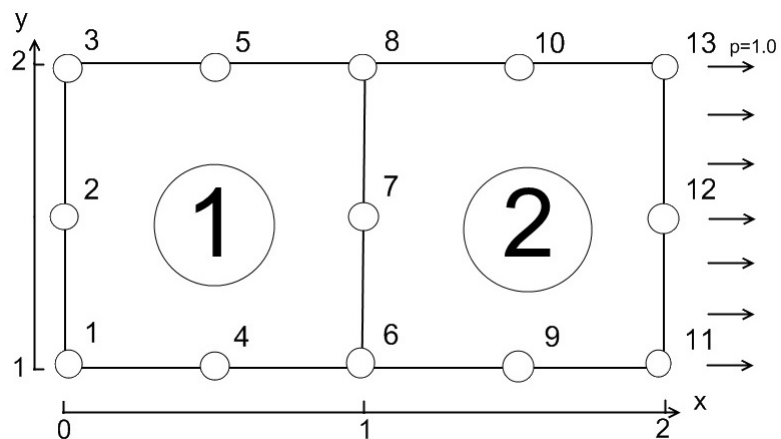
$$\xi = -1 : a_1(-1)(-1 - 1) = 1.$$

így $a_1 = 1/2$ és az N_1 alak függvényt meg is találtuk.

4.3. A program működése

A program, illetve a programban felépített modell teljes egészében Gennadiy Nikishkov: Programming Finite Elements in Java című könyvén alapul.

Nézzük a program működését egy egyszerű kétdimenziós modellen keresztül. Adott egy 2 elemből, elemenként 8 csomópontból álló modell, amelyet az x irányba szeretnénk megnyújtani, $p=1.0$ intenzitással.



12. ábra. Két 8 csomópontú véges elemből álló modell

4.3.1. A modell felépítéséhez szükséges adatok

Elsőnek is tisztáznunk kell, hogy milyen adatok is szükségesek a modellünk felépítéséhez.

Meg kell tudnunk adni a modellhez szükséges csomópontok számát, azok koordinátáját, a modellben lévő véges elemek számát és azok felépítését. Egy véges elem felépítéséhez tudnunk kell, hogy az adott elem milyen anyagból áll, hány csomópontból épül fel - a mi esetünkben ez 8 db csomópontot jelent - illetve, hogy a csomópontok hogyan, milyen sorrendben kapcsolódnak egymáshoz. Ezeket mind egy bemeneti fájlban adjuk meg, amely a végeselem hálót generálja, ennek a fájlnek a neve .mesh. A .mesh fájl következőképpen fog kinézni:

```
nNod = 13 nEl = 2 # Csomopontok es elemek szama
# csomóponti koordinatak
nodCoord = 0 0 0 0,5 0 1 0,5 0 0,5 1
           1 0 1 0,5 1 1 1,5 0 1,5 1
           2 0 2 0,5 2 1
# Elem adatok: elem tipus, anyag, kapcsolatok
elCon = QUAD8 1 1 4 6 7 8 5 3 2
        QUAD8 1 6 9 11 12 13 10 8 7
end
```

13 csomópontunk és 2 elemünk van, egy elem 8 csomópontból áll (típusuk: QUAD8), az anyagi tulajdonságnál az adott anyag neve van meg adva, ezt még a véges elem háló előtt definiáljuk. Az anyagi tulajdonságot a Young-modulus, Poisson tényező és a hőáglási együtharó segítségével definiáljuk.

4.3.2. A modellen végbemenő erőhatások definiálásához szükséges adatok

A modellünkön végbemenő erőhatás definiálására két további fájlra lesz szükségünk.

A .load fájlban adjuk meg, hogy melyik elemnél, elemeknél, és az adott elem, elemek mely csomópontjainál, ott milyen erővel történik az elmozdulás, kétdimenziós modell esetén ez az elmozdulás csak az x vagy y irányba történhet (nyújtás, összenyomás, húzás, tolás). Illetve itt adjuk meg az egyes csomópontok hőmérsékletét is. A fájl tehát a következőképpen néz ki:

```
loadStep = 1
# Felületi terhelés: irány, elemszam, elek szama
# csomopontok száma, el csomopontok szama, intenzitas
surForce = x 2 3 11 12 13 1 1 1
# csomóponti hőmérséklet
nodTemp = 10 10 10 10 10 10 10 10 10 10 10 10 10
end
```

A .fem fájlban definiáljuk, hogy pontosan milyen típusú problémával is állunk szemben. Itt adjuk meg, hogy milyen anyagokból épüljön fel a modellünk. Itt definiálhatjuk a kényszerfeltételeket, a mi

esetünkben a modell végpontjait rögzítenünk kell, hogy az ne mozduljon el. Engedézhajlítjuk, hogy az elmozdulásnál figyelmebe vegye e a különböző csomópontok hőmérsékletét.

```
stressState = PLSTRESS # Sik feszultsegi allapot
thermalLoading = Y # Hoterheles engedelyezes
# Vegeselem halo
includeFile f.mesh
# Anyagi tulajdonsagok:
material = 1 1 0,3 0,1
# Kényszerek
constrDispl = x 0,0 2 1 -3
constrDispl = y 0,0 5 1 4 6 9 11
end
# Load case
includeFile f.load
end
```

A probléma típusa lehet, háromdimenziós (THREED), illetve kétdimenziós (PLSTRESS). Mint már fentebb említettük, az anyagi tulajdonságot a Young-modulus, Poisson eloszlás és a hőtáglási együtható segítségével definiálhatjuk. Kényszerfeltételeket megadása úgy történik, hogy elsőnek megadjuk az irányt (x), majd egy értéket (0.0), meg adjuk, hogy az adott irányban hány darab csomópontot szeretnénk rögzíteni, végül felsoroljuk a rögzíteni kívánt csomópontok számát. A .fem fájlban hivatkoznunk kell a .mesh fájlra, a véges elem háló miatt, illetve a .load fájlra az elmozdulás miatt.

4.3.3. Eredmények és a modell kirajzolása

Ha mindegyik bemeneti fájlal meg vagyunk és minden adat a helyén van, akkor lefutathatjuk a .fem fájlt. A fájl lefutása után kapunk egy .fem.lst fájlt, amely az eredményeket tartalmazza, a mi esetünkben ez így fog kinézni:

Elmozdulások

Node	ux	uy
1	1.666667e-65	2.951815e-80
2	6.666667e-65	3.500000e-01
3	1.666667e-65	7.000000e-01
4	1.000000e+00	3.373503e-80
5	1.000000e+00	7.000000e-01
6	2.000000e+00	2.951815e-80
7	2.000000e+00	3.500000e-01
8	2.000000e+00	7.000000e-01
9	3.000000e+00	0.000000e+00
10	3.000000e+00	7.000000e-01
11	4.000000e+00	-6.325319e-80
12	4.000000e+00	3.500000e-01
13	4.000000e+00	7.000000e-01

Stresses

```
El    1  sxx syy sxy szz epi
      1,00000000  0,00000000  0,00000000  0,00000000  0,00000000
      1,00000000  0,00000000 -0,00000000  0,00000000  0,00000000
      1,00000000  0,00000000 -0,00000000  0,00000000  0,00000000
      1,00000000 -0,00000000  0,00000000  0,00000000  0,00000000
El    2  sxx syy sxy szz epi
      1,00000000  0,00000000 -0,00000000  0,00000000  0,00000000
      1,00000000 -0,00000000  0,00000000  0,00000000  0,00000000
      1,00000000 -0,00000000 -0,00000000  0,00000000  0,00000000
      1,00000000  0,00000000  0,00000000  0,00000000  0,00000000
```

A listában felsorolásra kerülnek az elmozdult koordináták, illetve, hogy az egyes elemeknél milyen feszültségek tapasztalhatóak.

A programban lehetőségünk van kirajzoltatni a modelleket, a Java 3D segítségével. Igaz, hogy a modellünk csak kétdimenziós, de a programban lehetőségünk van háromdimenziós modellek építésére is, illetve a kirajzolás teljes mértékben a Java 3D-re épül.

A modell kirajzoltatása miatt szükségünk van még további 2 bemeneti fájlra.

Készítenünk kell egy .res fájlt, amely az elmozdulásokat és az elemeknél fellépő feszültségeket tartalmazza, ez a fájl ugyanúgy néz ki mint az előbb tárgyalt lista fájl, amely az eredményeket tartalmazza. Illetve kell még nekünk egy .vis fájl, amely a kirajzolásért lesz felelős.

Nézzük, hogy az eredeti modellünkönél ezek, hogy fognak kinézni.

```
# f.res fajl
Elmozdulások
```

```
Node ux uy
```

```
1 0 0
2 0 0,5
3 0 1
4 0,5 0
5 0,5 1
6 1 0
7 1 0,5
8 1 1
9 1,5 0
10 1,5 1
11 2 0
12 2 0,5
13 2 1
```

Stresses

```
El  1 sxx syy sxy szz epi
    1,00000000  0,00000000  0,00000000  0,00000000  0,00000000
    1,00000000  0,00000000 -0,00000000  0,00000000  0,00000000
    1,00000000  0,00000000 -0,00000000  0,00000000  0,00000000
    1,00000000 -0,00000000  0,00000000  0,00000000  0,00000000
El  2 sxx syy sxy szz epi
    1,00000000  0,00000000 -0,00000000  0,00000000  0,00000000
    1,00000000 -0,00000000  0,00000000  0,00000000  0,00000000
    1,00000000 -0,00000000 -0,00000000  0,00000000  0,00000000
    1,00000000  0,00000000  0,00000000  0,00000000  0,00000000
```

```
# f.vis fajl
meshFile = f.mesh
resultFile = f.res
parm = ux
showEdges = Y
showNodes = Y
nContours = 8
fmin = 0,85
fmax = 3.0
end
```

A .vis fájl a .res és .mesh fájlok megadásása után az ux paraméter segítségével rajzolja ki a modellt. Nyolc féle színt használunk a különböző feszültségek jelölésére.

Az eredeti modellünk tehát a következőképpen fog kinézni a beavatkozás előtt és után:



13. ábra. Az eredeti modell



14. ábra. Modell a nyújtás után

4.3.4. Egyenes íj modell

A programban egy egyszerű rúdat hajlítunk meg és figyeljük annak viselkedését. A rúd anyaga teljes egészében fából (tölgy) és az ennek megfelelő rugalmassági adatból épül fel, 65 cm hosszú, a modell vastagsága fokozatosan csökken.

Ezen a modellen keresztül egy egyszerű íj viselkedését is megvizsgálhatjuk, elég csak az íjnak az egyik felét elkészítenünk, mivel markolattól lefelé, illetve felfelé is hasonlóan viselkedik húzáskor az íj, illetve modellezés szempontjából is sokkal könnyebb ennek megvalósítása. Tehát ezt nézhetjük, úgy mintha lenne egy tölgyfából készült 130 cm hosszúságú egyenes íjjunk, amelynek markolattól (3 cm) a húrtartóig (1,5 cm) fokozatosan csökken a vastagsága. A modell 2041 darab koordinátából és 596 darab véges elemből áll. A programban ez így néz ki:



15. ábra. A modell

A végeselemek számának növelésével jobb eredmény kapható. A modellt y irányban elhajlítjuk $p=0,1$ intenzitással, úgy mintha egy íjat feszítenénk. A modellen fellépő feszültség szintek különböző színekkel vannak jelölve. A modell a hajlítás után a következőképpen fog kinézni:



16. ábra. A modell hajlítás után

Látható, hogy egy ilyen egyszerű modellnek sem egyszerű a felépítése, rengeteg koordinátából és összekapcsolásokból áll.

5. Összefoglalás

A dolgozatunkban az egyenes, illetve a visszacsapó íj működésével foglalkoztunk. Az íj viselkedésének leírása kapcsán foglalkoztunk a végeelem módszerrel és a végeelem programozással. A végeelem módszert annak fizikáján keresztül mutattuk be, a Newtoni mechanikából kiindulva, az általános koordináták segítségével eljutottunk a D’Alambert elvig, majd megismerkedtünk a Lagrange függvény-nyel, illetve foglalkoztunk a rugalmasságtannal is. A végeelem programozást a Java programozási nyelven keresztül valósítottuk meg, bemutattuk, hogy milyen módszereket is alkalmaz a program ennek megvalósítása érdekében (Galerkin módszer, Variációs megközelítés). Egy egyszerű modellen keresztül bemutattuk a program működését, illetve építettünk egy kezdetleges, fából készült rudat, amelyet sikerült elhajlítani.

Továbbá terveink között szerepelt egy bonyolultabb egyenes, illetve visszacsapó íjmodell végeelem módszerrel történő modellezése is. Szerkezetük bonyolultsága miatt, ezen modellek nem készültek el.

6. Saját munkák leírása

A munkamegosztás a következőképpen zajlott

Tar Gergő:

- Végeelem módszer fizikája
- Végeelem módszer megvalósítása Javában
- Egyszerű modell felépítése

Fehérvári Zoltán

- Íj működésének vizsgálata
- Végeelem programozás
- Egyszerű modell felépítése

7. Köszönet nyílvánítás

Köszönetünket fejezzük ki a témavezetőknek, Dr. Tóth Lászlónak a dolgozatunk elkészítésében nyújtott segítségéért és útmutató tanácsaiért.

Hivatkozások

- [1] MASAHIRO MORII, *Physics 151 : Mechanics, Fundamental ideas of classical mechanics including contact with modern work and applications.*, Professor of Physics, Harvard University, 2003
- [2] TÓTH LÁSZLÓ, *A visszacsapó íj*, A Debreceni Déri Múzeum Évkönyve 2004
- [3] GENNADIY NIKISHKOV, *Programming Finite Elements in Java*
- [4] BAGI KATALIN, BOJTÁR IMRE, TARNAI TIBOR, *Rugalmasságtan alapkérdések segédlet*, 2006
- [5] http://www.volkswagen.hu/volkswagen_koeruel/innovacio/kutatas/szimulacios_rendszerek/vegeselem_modszer/