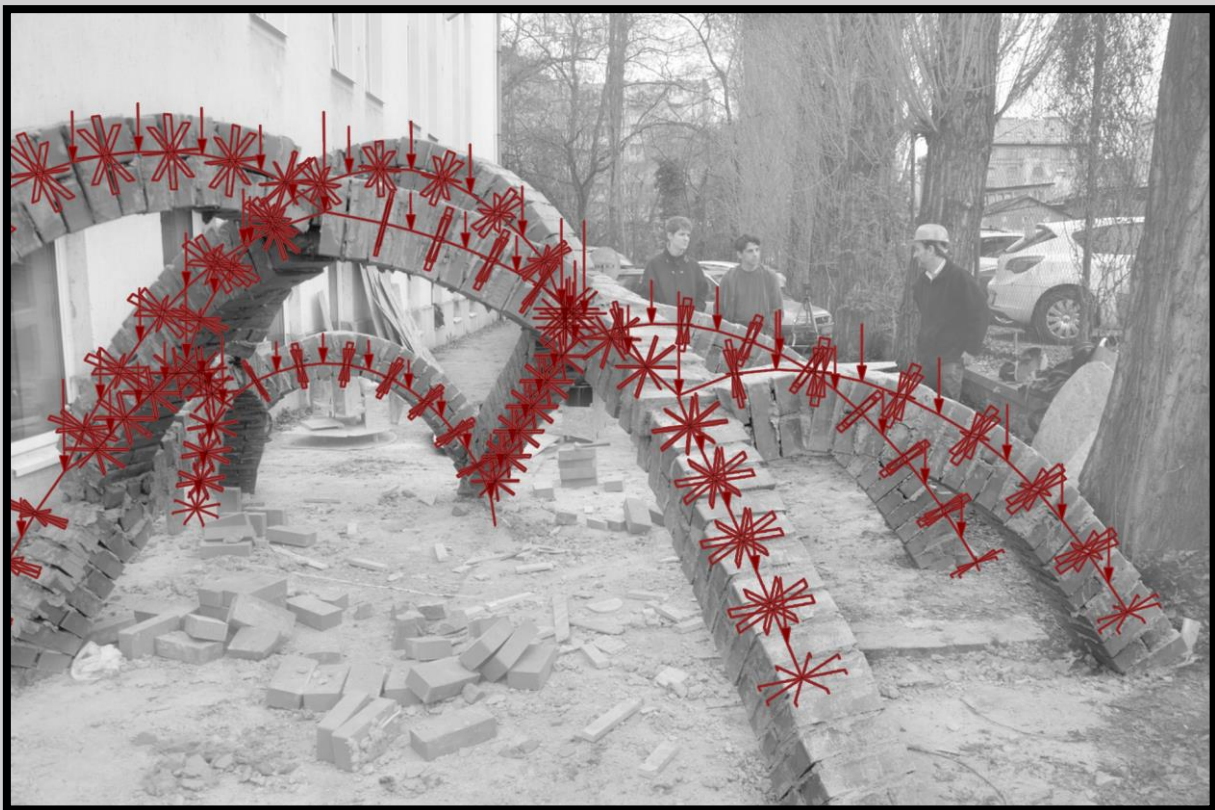


ÍVRENDSZEREK TÖNKREMENTELE

– törőteher számítása végesselemes módszerrel



TDK 2021.

Szerző: Varga Kristóf R.

Konzulens: Dr. Ther Tamás

TARTALOMJEGYZÉK

BEVEZETÉS	2
ELMÉLETI HÁTTÉR	4
PROGRAMOZÁSI KÖRNYEZET	6
A PROGRAM FELÉPÍTÉSE	7
EREDMÉNYEK	19
ZÁRÓ GONDOLATOK	23
FELHASZNÁLT IRODALOM	23

BEVEZETÉS

A középkori (elsősorban gótikus) boltozatok szerteágazó fejlődését vizsgálva sokszor felmerül a tartószerkezeti innováció vélelme. Ez olykor triviálisnak tűnhet, egyszerű elvek mentén bizonyítható is, főleg, ha hosszabb, jelentősen eltérő korszakok jellegzetességeit hasonlítjuk össze. Sok esetben viszont nehezen megállapítható, hogy melyik megoldás eredményez jobb, erősebb szerkezetet, s különösen igaz ez a késő-gótika egyre összetettebb megoldásaira. Tekintve, hogy számos más tényező is szerepet játszott az említett építmények morfológiai alakulásában – gondolhatunk itt az akkor még jóval korlátozottabb kivitelezési módszerek fejlődésére, az egyre ornamentálisabb szerkezeti kialakításokra, vagy akár az egyes régiók jellegzetességeire, például az egyenes záródáshoz való ragaszkodásra – egyáltalán nem evidens, hogy adott formai eszköz milyen és mekkora változást hozott mechanikai értelemben. Tovább árnyalja a képet a konkrét geometriában való bizonytalanság: felmérésekből ugyan ismertek a fennmaradt épületállományban lévő boltozatok, ám az építési pontatlanság és az eltelt idő alatt elszenvedett alakváltozások miatt nem feltétlen azonosítható be egyértelműen az eredeti, elvi szándék. A különböző felületi geometriák márpedig különböző erőjátékot eredményeznek, és adott morfológiai családon belül is sokféle variációt találhatunk. Érdekes például belegondolni, hogy nagyságrendileg azonos bordamennyiséget tartalmazó csillagboltozatok esetén mely bordaelhelyezés tekinthető hatékonyabbnak. Nos, ezek a kérdések inspiráltak arra, hogy ezt valamilyen módon érdemes lenne mélyebben megvizsgálni: megnézni, hogy adott boltozatcsaládok geometriai paramétereinek korlátai mit jelenthetnek a szerkezet mechanikai tulajdonságainak tekintetében, ez alapján pedig több családot összehasonlítani, akár különböző hatásokra, teherelrendezésekre.



[f.1.] Bishop West's Chantry Chapel, Ely Cathedral, Cambridgeshire

Ez természetesen egy meglehetősen sok mechanikai problémát magában foglaló téma, ezért szükségeltetik némileg leszűkíteni, részfeladatokra bontani. Ha a bordás boltozatokra tekintünk, a bordák közti felületelem például sok modellezési – és így tartószerkezeti – kérdést felvet. Együtt viseli-e a terhet a bordákkal, vagy hierarchikusnak tekinthető, és csak a terhek elosztását befolyásolja; esetleg milyen merevítő funkciói lehetnek, gátol-e bizonyos hajlítási vagy csavarodási elmozdulásokat, stb. Ezen bonyodalmaktól ideiglenesen megszabadulva, az a döntés született, hogy első körben kizárólag a bordahálózat, mint önálló ívtartók rendszere lesz a vizsgálat tárgya.

A feladat még így is összetett, hiszen többszörösen határozatlan tartókról beszélünk, a repedések megnyílása, így a statikai modell megváltozása miatt erős nemlineáris viselkedéssel. A vizsgálat célja a törőteher megállapítása különböző, az önsúly mellett fellépő, jellemzően aszimmetriát okozó terhelések hatására.

A problémára egyébként létezik tisztán geometriai megoldás, amely egyszerű, síkbeli tartó esetén vélhetően a legkönnyebben elvégezhető – egy integrálással meghatározható alakú görbét (a nyomásvonalat) egy zárt intervallumon belül vett szabad paraméter által (horizontális erőkomponens), affinitással transzformálva megkaphatjuk az eredményt (az alapján, hogy a görbe minden pontja belül esik-e egy zárt alakzaton) – nevezetesen, hogy a szerkezet elbírja-e az adott terhet, avagy összedől. Viszont, ha több, egymásra terhelő ívet veszünk, minden változóhoz újabb intervallumnyi változó rendelődik hozzá, ráadásul, ha mindezt térben szeretnénk elvégezni, akár több tucat íven, akkor azt látjuk, hogy a megoldás összetettsége nagyságrendileg arányos lesz a feladat összetettségével.

Éppen ezért kezelhetőbb megoldásnak tűnt egy végeelemes program konstruálása, amely legfeljebb elemszámát tekintve növekszik, arányosan a hozzáadott rudak számával, ám függetlenül azok kapcsolatától.

A módszer már csak azért is adta magát, mert amilyen szintű pontosságra és parametrizálhatóságra szükség lesz a későbbiekben a geometriai objektumokat (elsősorban görbéket) illetően, ésszerű választásnak tűnik a McNeel által fejlesztett Rhino CAD szoftver használata, és a hozzá kifejlesztett Grasshopper programozási környezet. Ebben pedig már létezik beépített végeelemes kiegészítő, a Karamba 3D, tehát, amennyiben az ott található lineáris rúdmodell használatával létre lehet hozni egy számunkra megfelelő, nemlineáris számításra képes programot, ami tetszőleges ívrendszer és tehereset elemzésére univerzálisan alkalmas, akkor egyetlen programon belül elvégezhetőek a vizsgálatok. Ez vált tehát a kutatás első fázisának közvetlen céljává.

ELMÉLETI HÁTTÉR

A következőkben röviden összefoglalom azon alap gondolatokat és előfeltevéseket, amelyek a program működésének megértéséhez szükségesek.

Elsőként tekintsük egy két végén befogott, egyszerű, síkbeli ívtartó mechanikai viselkedését. Amennyiben az ívet egyetlen, görbe tengelyű rúdnak tekintjük, amely merev testként viselkedik, a kétszeres befogás miatt a kiinduló állapotban a szerkezet háromszorosan határozatlan.

Ha ezt a rudat olyan erő terheli, melynek intenzitása megegyezik a görbe alakjának második deriváltjával, akkor a rúdban kizárólag normálerő ébredhet, ennek megfelelően pedig a tönkremenetel legfeljebb anyagi jellegű lehet – eltekintve most a stabilitásvesztés lehetőségétől. Minden olyan esetben viszont, amikor ettől eltérő terhet kap (ide tartozhat sok esetben maga az önsúly is), a tartóban nyírás és hajlítás is ébred. Ennek vizsgálatához Heyman három közelítéssel él [1]:

i.) A kövek nem csúszhatnak el egymástól.

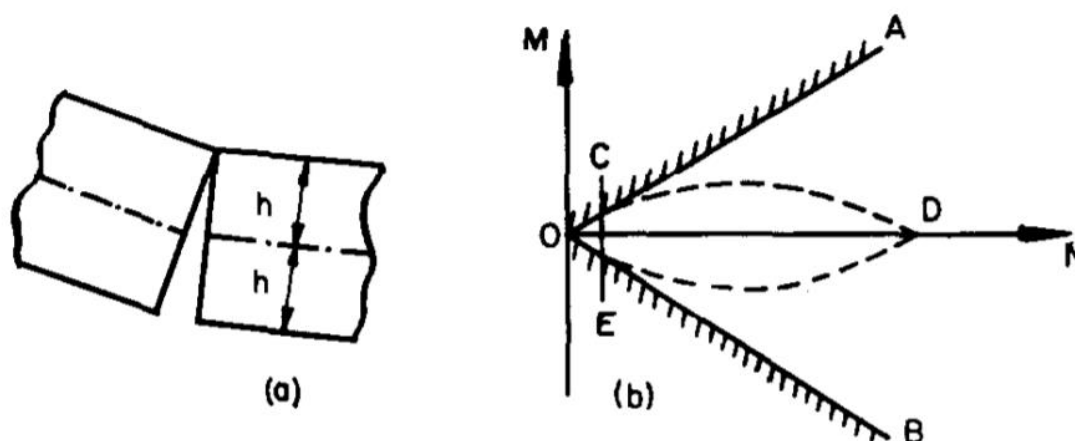
Coulomb megállapításai szerint [2] a boltíveket alkotó elemek súrlódási ellenállása olyan nagy, hogy a kövek nem tudnak egymáson elcsúszni, az ehhez tartozó nyírás tönkremeneteltől tehát eltekinthetünk.

ii.) A köveknek nincs húzószilárdsága.

Ez a feltevés a biztonság javára téved. Bizonyos szerkezeteknél ugyanis megoldható, hogy húzást is át tudjon adni, erre maga Heyman is kitér, viszont a legtöbb boltív esetében egy ésszerű közelítés, hiszen előfordulhat, hogy a köveket szárazon falazták, vagy egyszerűen gyenge habarccsal. Így tehát, bár magának a kőnek lehet húzószilárdsága, a kövek közti kapcsolatoknak vélhetően nincs; ott a szerkezet szabadon berepedhet.

iii.) A kövek nyomószilárdsága végtelen.

Ez a feltevés a biztonság kárára téved. Alapja, hogy általában a normálfeszültség elég alacsony ahhoz, hogy a számítást ez ne befolyásolja. Viszont, ha konkrétan nyomásra nem is menne tönkre a szerkezet, a nyomás nagysága csökkentheti a maximálisan elérhető külpontosságot, ezáltal pedig a keresztmetszet nyomatéki teherbírását.



[f.2.] keresztmetszet határállapota (Heyman [1], p.5.)

Ugyanis, ha a ii.) pontban foglaltak szerint nincs húzószilárdság, az egyetlen mód a hajlítás felvételére a külpontos nyomás. Véges nyomószilárdság esetén viszont a nyomóerő átadása nem lehetséges egyetlen ponton, szükséges hozzá egy, a normálerő nagyságával lineárisan arányos felület. Világos tehát, hogy minél nagyobb az ébredő erő, annál kevesebb hely marad a keresztmetszetben a külpontosság biztosítására, és arányosan annál kevesebb nyomatékot bír.

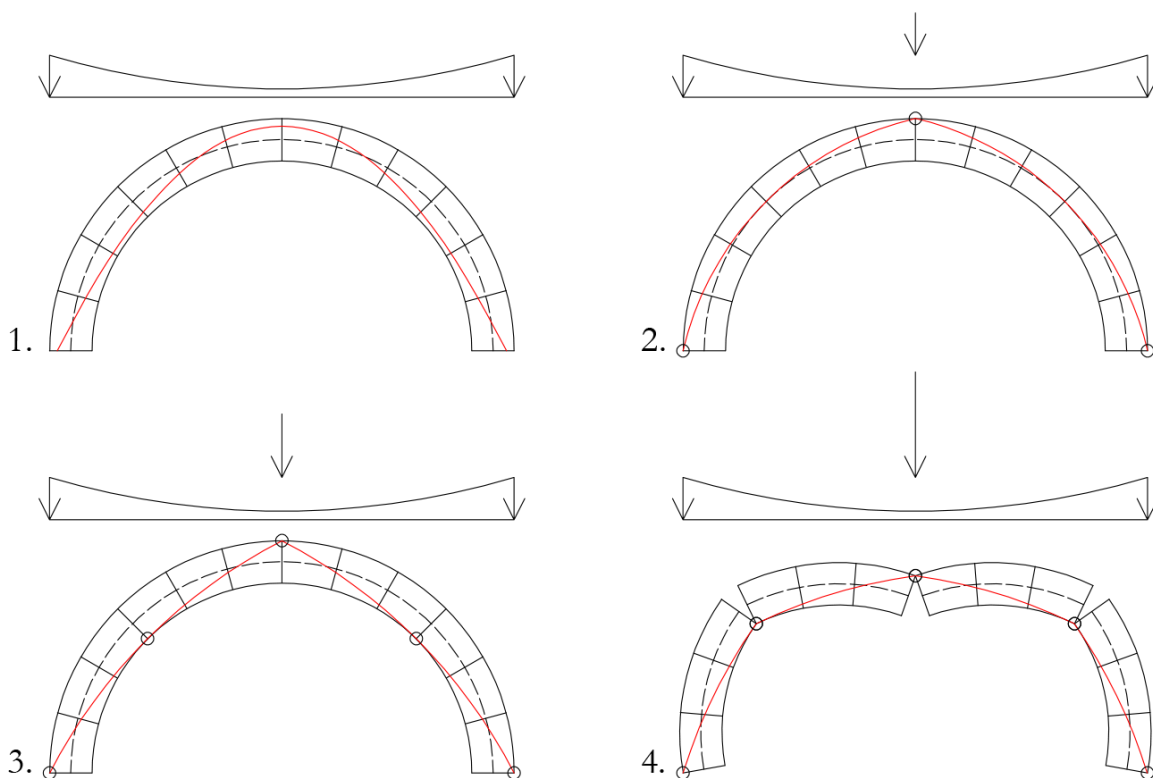
Ennek ellenére ez a közelítés megengedhető, mert amíg a jellemző normálerő nagyságrendileg a tizede a teherbírásnak, addig az ábrán látható OCE háromszögön belül marad a keresztmetszet igénybevétele, így a lineáris és parabolikus függvény közt nincs túl nagy eltérés.

Ilyen közelítések mellett a keresztmetszet a hajlítási teherbírásának elérésekor (tehát amikor a nyomásvonal érinti a keresztmetszet szélét) megnyílik a külpontos erővel szemközi oldalt, szabad elfordulást biztosítva. Ilyen képlékeny csuklók kialakulásával vesztí el sorra a szabadságfokokat, amíg mechanizmussá nem alakul – ekkor a szerkezet egésze tönkremegy, mint arra már Pippard és Baker [3] is rámutatott.

Mivel a program segítségével nem közvetlenül a tönkremenetel pillanatát fogjuk megtalálni, hanem lépésenként haladunk az újabb és újabb csuklók kialakulásához, a Heyman-i elvek mellett egy további lényeges közelítéssel kell élnünk:

iv.) A keresztmetszet repedései nem zárulnak.

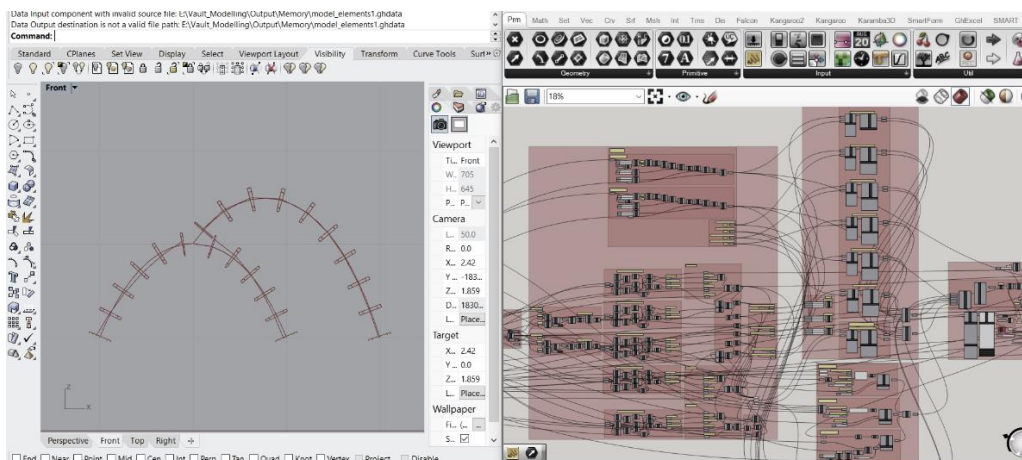
A valóságban ugyanis a képlékeny csuklók helyzete nem feltétlenül állandó az időben. Meglévő repedések zárulnak, és nyílnak máshol, ám ezt számítással lekövetni meglehetősen bonyolult volna, ezért ettől eltekintünk.



[f.3.] boltív tönkremenetelének elvi koncepciója növekvő koncentrált teher hatására, a nyomásvonal ábrázolásával

PROGRAMOZÁSI KÖRNYEZET

Mielőtt továbblépnénk a program működésére, röviden bemutatnám a használt szoftvereket.



[f.4.] képernyőkép a használt program felületéről

McNeel: Rhinoceros

A Rhino egy 3D modellezésre alkalmas CAD szoftver. Mint azt korábban Boronkay Gábor is részletesen kifejtette tanulmányában¹, a modellezés egyik legnagyobb erősségét a NURBS (Non-Uniform Rational B-Spline) adja, amely lehetővé teszi görbék és felületek precíz megadását. Ennek köszönhetően elsősorban 3D nyomtatásban és alkatrészgyártáshoz kapcsolódó iparágakban terjedt el, esetünkben viszont egy másik egyedülálló tulajdonsága volt releváns, nevezetesen, hogy egyik kiegészítője segítségével talán a legkönnyebben parametrizálható és programozható CAD szoftver a piacon.

Grasshopper

A Grasshopper egy C#-ban írt kiegészítő, amely gyakorlatilag egy vizuális programozási felület. A Rhino és a GH objektumai könnyedén megfeleltethetők egymásnak, átvihetők egyik szoftverből a másikba, ráadásul a Rhino élő megjelenítést biztosít minden GH-ben definiált elemnek, ezáltal egyszerűvé téve a programkód ellenőrzését. A felület könnyen kezelhető – gyakorlatilag programozási nyelv tanulása nélkül, pusztán az ahhoz szükséges logika és módszertan alkalmazásával könnyen elsajátítható a használata. Mindezeknek köszönhetően kiválóan alkalmas szoftver az ilyen jellegű – félig grafikus, félig számítási – problémák megoldására.

C# programnyelv

A Grasshopper több programozási nyelvnek is teret biztosít, ha mégis szükségessé válna scriptelni. Egyes funkciókat továbbra is egyszerűbb ilyen módon megírni, ráadásul vannak kifejezetten fontos, de GH-ben közvetlenül nem interpretált parancsok, mint például a FOR ciklus, amelynek visszacsatolása elengedhetetlen bizonyos programok megírásához.

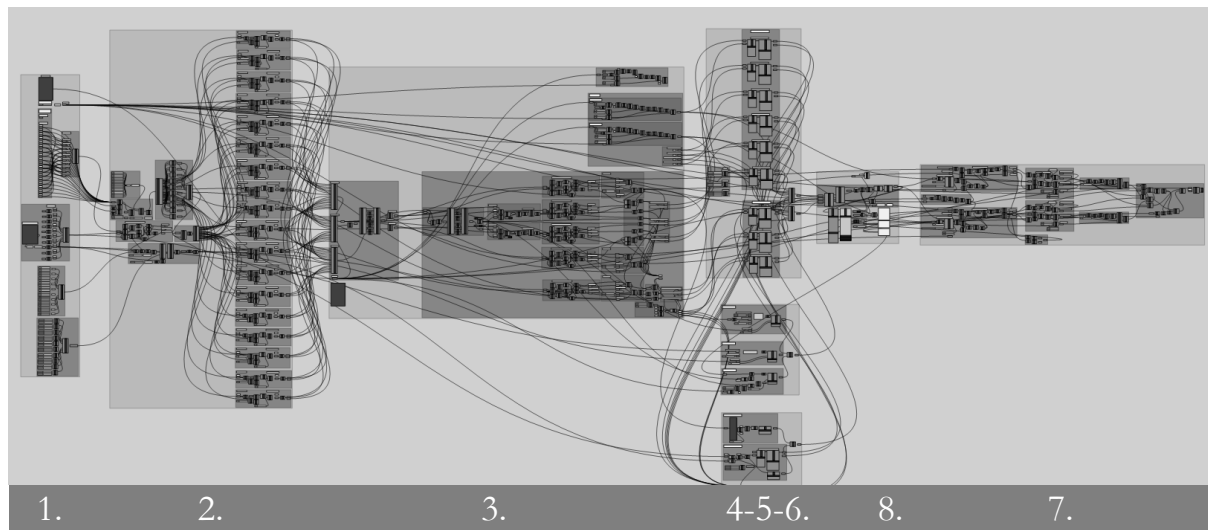
Karamba 3D

A Karamba 3D egy a Grasshopper számos belső kiegészítője közül. A program alapvetően szerkezettervezéshez való végeselemes modulokból áll: rúd- és felületelemek tartalmazó modellek létrehozása, azok lineáris és nemlineáris számítása, kiegészülve az összes megszokott paraméterrel – támaszok, terhek, anyagjellemzők, merevségek stb. definiálásával.

¹ Boronkay Gábor: Tartószerkezet tervezés evolúciós eljárásokkal (TDK 2018.)

A PROGRAM FELÉPÍTÉSE

PROGRAMTÉRKÉP



[f.5.] képernyőkép a programról az egyes modulok számozásával

GEOMETRIAI MODELL KONSTRUÁLÁSA

1. Ívrendszer definiálása
2. Csomópontok definiálása
3. Diszkretizáció

VÉGESELEMES MODELL KONSTRUÁLÁSA

4. Összetett rúdelemek definiálása
5. Támaszok definiálása
6. Terhek definiálása
7. Számítás
8. Vezérlőmodul (a program indításához, eredmény kijelzéshez)

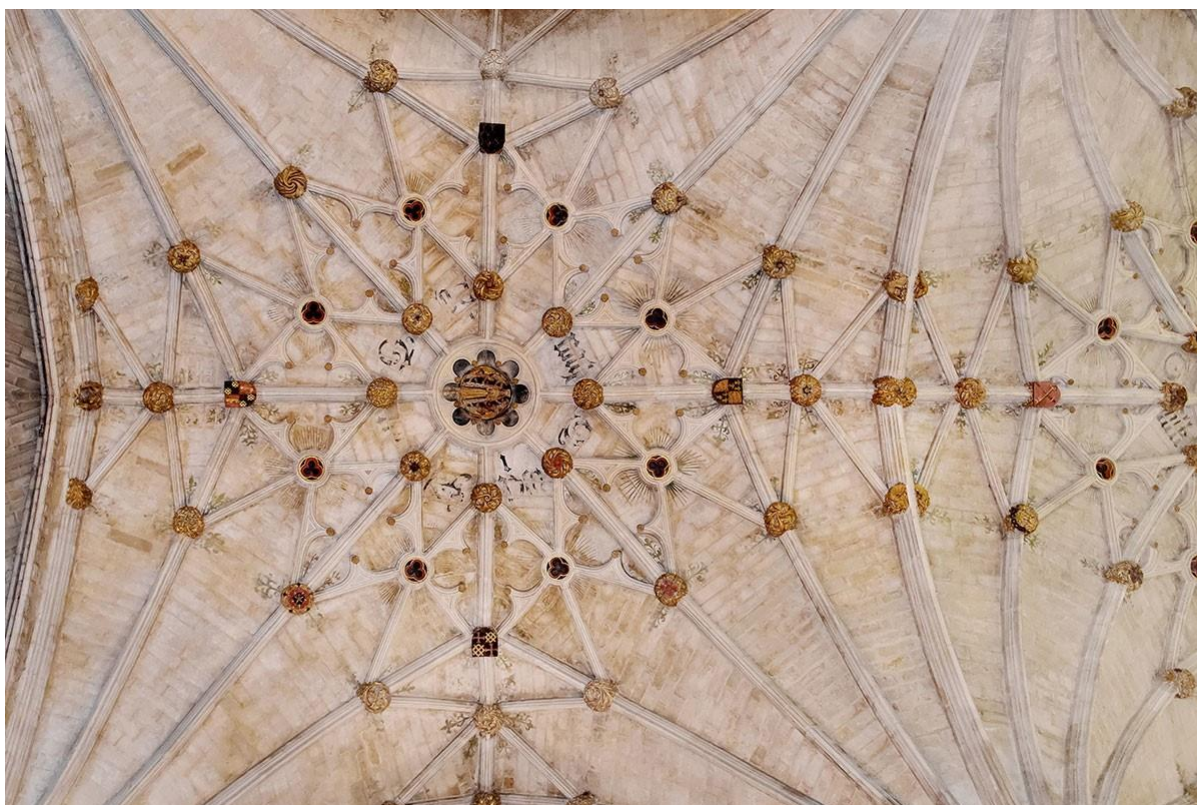
GEOMETRIAI MODELL KONSTRUÁLÁSA

1. Ívrendszer definiálása

Első lépésként a programnak be kell olvasnia az ívrendszer geometriáját. Optimális megoldás lehet eleve Grasshopper-ben definiálni a görbéket, egyéni parametrizálással biztosítva a számunkra előreláthatólag szükséges változtatások lehetőségét. A program kiegészítői közt fellelhetők alakkeresésre alkalmas algoritmusok is, így adott esetben a tervezési fázis közvetlenül összekapcsolható a szerkezet ellenőrzésével.

Egy másik opció lehet, hogy a Rhino felületén hozzuk létre az ívrendszert alkotó görbéket, ebben az esetben viszont kizárólag ott tudjuk módosítani ezeket, fogópontok segítségével. Ez a megoldás akkor lehet előnyös, ha előre ismert, esetleg más CAD szoftverben már megrajzolt geometriát használunk, a kompatibilitásnak köszönhetően ugyanis a modell könnyedén importálható.

Ha a program beolvasta a görbéket, létre kell hoznia egy adatstruktúrát, ami a későbbi hivatkozást lehetővé teszi. Ez alapvetően az egyes ívek számozását, valamint orientációját (kezdő- és végpontok definiálását; tetszőleges pontra felírt kísérő triéder számítását) jelenti. Ha Grasshopper-ben készített ívrendszerünk van, akkor ezt akár rögtön meg is adhatjuk manuálisan. Importált modell esetén kis elemszámnál ugyanúgy működhet, ráadásul a manuális számozás sokat segíthet az átláthatóságban az eredmények ellenőrzésekor. Magasabb elemszámnál erre érdemes volna egy külön algoritmust írni, ami automatikusan kezeli az ebből adódó szabadságfok változásokat a VEM modellben.



[f.6.] Winchesteri katedrális boltozata – csomóponti elemek kialakítása

2. Csomópontok definiálása

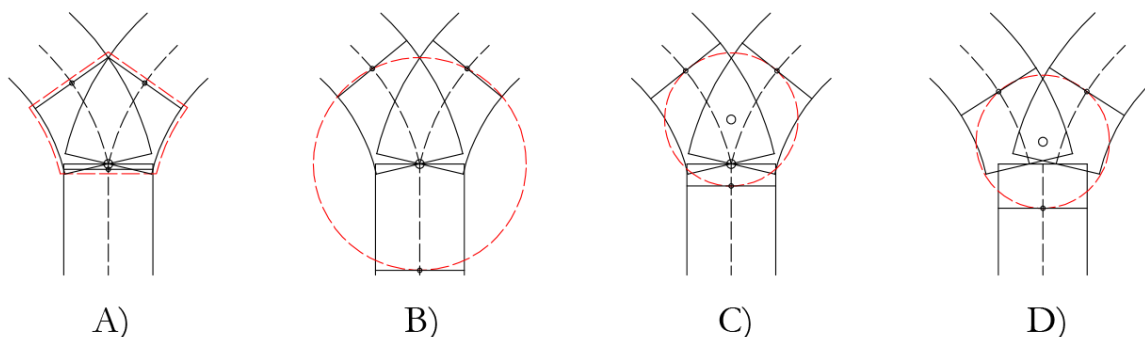
2.1. Pontok megadása

Összetettebb ívrendszerek esetén az ívek találkozási pontjaiban csomópontokat kell megadnunk. Erre megint írható olyan egyszerű algoritmus, ami az ívek geometriáján végrehajt egy ütközésvizsgálatot, és ezek alapján automatikusan megtalálja a csomópontokat. Erre ezidáig nem volt igény, kisebb elemszámok esetén könnyen beolvashatók a pontok manuálisan. Speciális esetekben ráadásul kizárólag ez működik, például, ha tengelyvonal helyett zsaluzási vonal görbáját adjuk meg, hiszen ekkor az ívek párhuzamos eltolásakor megszűnnek az ívek metszéspontjai.

2.2. Csomóponti elemek kiterjedése

Mindaddig, amíg az íveket (tér-)görbének tekintjük, normálsíkon vett kiterjedés nélkül, a csomópontok is tekinthetők pontszerűnek. Az ívek keresztmetszetének definiálásakor azonban ezek összemetsződése problémákat okozhat. Ezt elkerülendő, a csomóponti elemekhez is kiterjedést kell rendelnünk. Ez a megoldás egyébként nem áll távol a valóságtól, a metsződéseknél kialakuló csomópont gyakran egyetlen köelemből állt [f.6.].

Ha az íveknek keresztmetszetet adunk, megtalálhatók a felületek metszetszögéből. A minimális csomóponti méretet az adja, ha minden ív-páros metszetén a görbe legtávolabbi pontján áthaladó normálsíkkal elmessük az eredeti görbét [f.7. A]. Közelítőeljárásnak viszont az is megfelel, ha olyan gömböt definiálunk, amely ezeket a pontokat mind magában foglalja. Ekkor az ívekből való levágás egységes hosszú [f.7. B] – ez a középpont utólagos módosításával korrigálható [f.7. C].



[f.7.] csomóponti elem kiterjedésének különböző definíciói

2.3. Ívek módosítása

A csomóponti elemek kiterjedéséből adódóan az ívekből egy darabot levágunk – a csomóponthoz csatlakozó kezdő- vagy végpontjai módosulnak. Látható az is, hogy amennyiben az ívek tengelyvonalai, mint görbék nem is rendelkeznek elméleti metszésponttal, ez nem fog problémát jelenteni, hiszen mindkét ív külön-külön csatlakozik a csomóponti elemhez, mint térbeli objektumhoz [f.7. D].

3. Diszkretizáció

3.1. Ívek felbontása

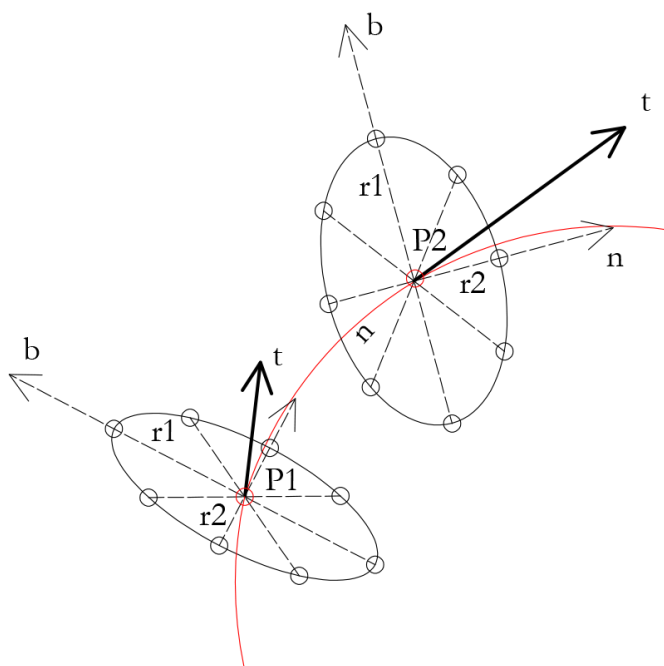
Az ívek megfelelő diszkretizálása több okból is lényeges. Egyrészt maga a modell egyenes rúdelemekkel számol, tehát a görbe rúdtengelyek geometriai közelítése már önmagában is megkövetel valamiféle felbontást. Másrészt viszont adódik egy statikai követelmény is, hiszen a keresztmetszetek ellenőrzésére éppen ezen rúdelemek találkozásában lesz lehetőség, tehát a felbontás közvetlenül a vizsgált keresztmetszetek számát fogja megadni.

Az általános eljárás, hogy egy maximális ívhosszt kell megadni paraméterként, a program pedig ez alapján minden ívet felbont. Megadható minden ívre manuálisan is a darabszám, ám ekkor tisztában kell lenni az adatstruktúrában rögzített csomópont és rúd sorszámozással. Ennek a programtesztelés megkönnyítésén túl a koncentrált terhek megfelelő elhelyezésében lesz szerepe.

3.2. Keresztmetszet definiálása és felbontása (3D)

A keresztmetszet szempontjából számunkra az adott irányban vett maximális kiterjedés lesz lényeges. Ez síkbeli tartók esetében nem más, mint maga az intrados és extrados vonala; egy adott keresztmetszetre vizsgálva ez mindig két pontot jelent. Térbeli rendszerben viszont a keresztmetszet széle egy zárt síkgörbe, amely pontokra redukálásához ismételten egy részletezettségi paramétert kell bevezetnünk.

A modellben használt keresztmetszet egy olyan ellipszis, amelynek kis- és nagytengelye a bi- és főnormális irányában helyezkedik el. A tengelyek nagysága definiálható egységesen, ívenként vagy akár keresztmetszetenként is, mivel a számolásnál releváns keresztmetszet felvétele az ívek felbontása után történik (a csomópont definiálásakor használtuk ugyan a keresztmetszetet, ott érdemes az íven belül maximálisan előfordulóval közelíteni, vagy pedig előre megmodellezni és importálni). Ezt követően az ellipszist előre megadott tengelyszám alapján egyenlő hosszúságú darabokra osztva kapjuk meg az ívdarabok leendő csatlakozásaként szolgáló keresztmetszeti pontokat.



[f.8.] keresztmetszetek definiálása az ívszegmenseken, különböző paraméterekkel a kísérő triéder mentén

VÉGESELEMES MODELL KONSTRUÁLÁSA

4. Összetett rúdelemek definiálása

4.1. Rúdelemek az ívekben

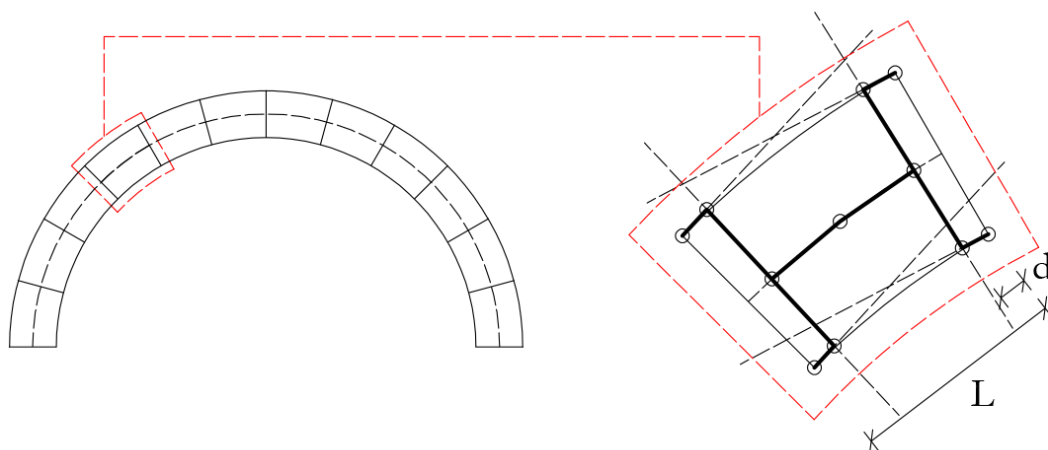
Az első elméleti közelítés az volna, ha a görbe rúdtengelyt egy tetszőleges részletezettségű törtvonallal lekövetnénk, ahol minden egyes szakasz egyetlen rúdelem. Egy ilyen modell igénybevételeire (a felbontás mértékétől függően, természetesen) nagyságrendileg jó eredményeket kapnánk.

A probléma a korábban már említett képlékeny csuklók kialakulásával lenne: ha ugyanis ebben a modellben csuklót alakítunk ki két szegmens között, az egyáltalán nem tud majd nyomatékot felvenni, miközben a célunk éppen az, hogy a keresztmetszet a csukló kialakulását követően is képes legyen ennek viselésére.

Mivel a keresztmetszet és a nyomásvonal viszonya valójában a külpontosságot korlátozza, a legegyszerűbb megoldás geometriailag felvenni ezt a nyomatékot: egy külpontos csuklóval biztosítani, hogy a kapcsolatban átadódó normálerő minden esetben a nyomatéki ellenállást adja át igénybevételeként a rúd tengelyére.

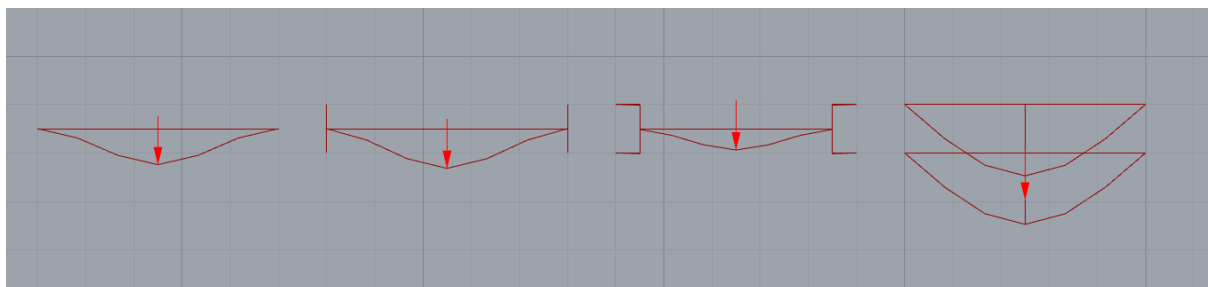
Ennek kivitelezésére két módszer áll rendelkezésünkre: újabb rúdelemek hozzáadása a modellhez, vagy pedig meglévők elvétele. Előbbi előnye, hogy a modell letisztultabb marad, a nyomaték iránya pontosabban megadható, a létrehozott rúdelemek száma pedig nagyságrendileg kisebb, ezáltal csökkentve a számítási kapacitás igényét. A program jelenlegi verziójában nem ez került alkalmazásra, viszont a Grasshopper környezetében hasonlóképp megvalósítható volna.

A másik (programozás szempontjából lényegesen egyszerűbb) megoldás, ha a rúdelemek közti mereven befogott kapcsolat helyett ugyanazt a szabadságfokot több, előre definiált külpontos csuklóval biztosítjuk. Ekkor a keresztmetszetben fellépő nyíró- és normálerő a csuklókban lokálisan ébredő megfelelő erők összege, míg a nyomaték a normálerők és külpontosságok vektoriális szorzatainak összege. Amikor képlékeny csukló kialakulását szimuláljuk, a keresztmetszetben lévő csuklók közül az az egy marad meg, amelyiket a nyomásvonal érinti, és az összes többit kivesszük a rendszerből az adott keresztmetszet repedését imitálva.



[f.9.] ívszegmens csomópontjainak geometriai összefüggései

Mivel a Karamba3D jelenleg nem rendelkezik kapcsolati erőket mérő modullal, ezeket a lokális igénybevételeket a rúdban ébredő erőkből kell leolvasnunk. Ennek megkönnyítése érdekében a csuklókhöz kapcsolódó rudak tangenciálisan kerültek elhelyezésre. Egy másik, elemszámot tekintve előnyösebb megoldás lehetne, hogy a keresztmetszeti ellipszisben radiálisan kapcsolódnak be ezek a rudak, viszont ekkor a nyíróerő egyik komponense válna normálerővé, és fordítva, megnehezítve a lekérdezést – ráadásul a két kapcsolódó keresztmetszet rúdelemeinek elkülönítése sem volna evidens a program számára.

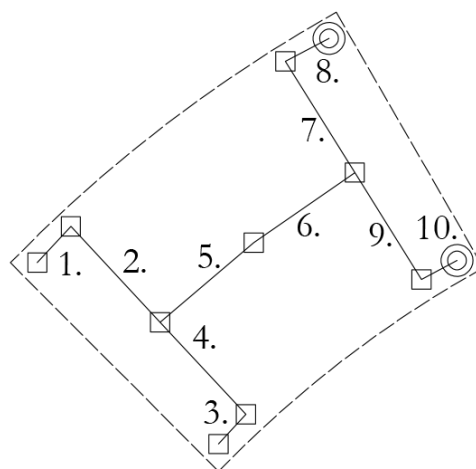


[f.10.] különböző ívszegmens geometriák elmozdulásai (balról jobbra: egyszerű rúd; $d=0$; $d=L/10$; $d=L/2$)

Az ívszegmens, mint több rúdból álló összetett elem konstruálásakor nem szabad elfelejteni, hogy a program elmozdulások alapján számolja majd az ébredő erőket, tehát csak akkor adhat reális eredményt, ha az elem elmozdulásának jellege nem tér el túlzottan az egyszerű egyenes rúdelem elmozdulásaitól. Éppen ezért tesztelni kell, hogy a tangenciális rudak miatt bevezetett új paraméter mennyiben befolyásolja a végeredményt. Az első elméleti közelítésben használt egyszerű, törtvonalas modellel összevetve azt kapjuk, hogy kizárólag a $d=L/2$ érték esetében látható szignifikáns eltérés. A programban végtelenül kis érték megadható, a számítások $L/20$ -as beállítással készültek.

A rúdvégek merevsége az összes összekötő rúd kezdő és végpontján merev befogásnak tekinthető. A tangenciális, két ívszegmens határán elhelyezkedő rudak esetén pedig a szegmens induló csomópontjánál merev, a végződő csomópontoknál pedig csuklós a kapcsolat. Ez az orientáció determinálja majd a támaszok és csomópontok kapcsolati kialakításait – ezért volt fontos korábban tisztázni az ívek irányultságát.

Rúdelemek sorszama, elnevezése	beamID
1. kezdő ág felső csomóponti rúd	[USB1]
2. kezdő ág felső összekötő rúd	[USB2]
3. kezdő ág alsó csomóponti rúd	[LSB1]
4. kezdő ág alsó összekötő rúd	[LSB2]
5. közbenső ág kezdő rúd	[MSB]
6. közbenső ág végződő rúd	[MEB]
7. végződő ág felső csomóponti rúd	[UEB1]
8. végződő ág felső összekötő rúd	[UEB2]
9. végződő ág alsó összekötő rúd	[LEB2]
10. végződő ág alsó csomóponti rúd	[LEB1]

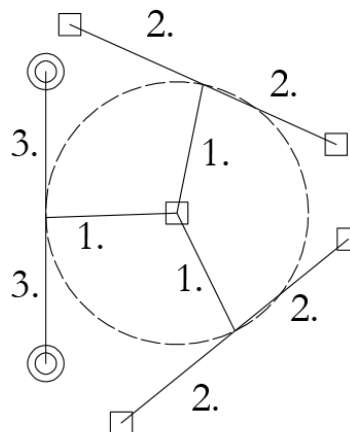


[f.11.] ívdarabot alkotó rúdelemek

4.2. Rúdelemek a csomópontokban

A csomóponti elemek esetében az ívszegmensekhez hasonlóan több rúdelemet kell definiálni. Itt már nincs jelentősége a csuklókhoz kapcsolódó elemek irányának, mivel a csomópontban nem vizsgáljuk a keresztmetszet tönkremenetelét. Ez azt is jelenti, hogy az eredeti ív tengelyvonala a csomópont általi visszavágásig nem mehet tönkre a számítás során. Mint az már említésre került, az ívszegmensekhez kapcsolódó rudak végződésénél kialakított merevség mindig a csatlakozó ívszegmens irányának függvénye – ha az ívdarab csuklós, akkor a csomópont befogott és fordítva.

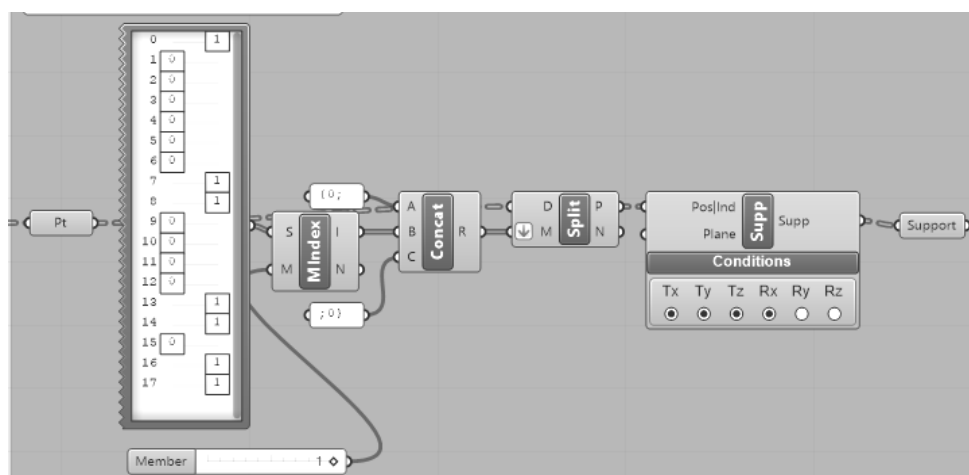
Rúdelemek sorszáma, neve	beamID
1. belső rúd	[MB]
2. merev összekötő rúd	[CSP]
3. csuklós összekötő rúd	[CSN]



[f.12.] rúdelemek a csomópontban

5. Támaszok definiálása

Támaszokról ezidáig nem beszéltünk, ezek megadása elvi szinten a csomópontokkal ellentétben kizárólag manuálisan történhet, hiszen a geometriából nem következik egyértelműen, hol szeretnénk a szerkezetet megtámasztani. Természetesen beépíthető volna olyan alapbeállítás, hogy mindig az ívrendszer szabad végződéseinél (tehát ahol nincs csomópont) definiáljon támaszt, de ez bizonyos esetekben a csomópontokhoz hasonló korrekciót igényelne – pl. egymással nem metsződő rúdtengelyek esetén. A támaszmerevségek megadása a csomópontokéhoz hasonló módon történik: a csatlakozó elemek orientációjának függvényében lehet merev vagy csuklós kapcsolat.



[f.13.] támaszok manuális definiálása a programban

6. Terhek definiálása

A terhek megadása a programban viszonylag triviális. A diszkretizáció miatt koncentrált terhekkel dolgozunk, amelyek az ívszegmensek középpontjában hatnak, ezért is szükséges ott az összetett elemen belül csomópontot definiálni.

Terhek létrehozásához csupán meg kell adni az erő nagyságát, irányát, valamint azt az ívdarabot, amin hat – illetve megadható konkrét pont is, amire a program kiszámolja, hogy az adott felbontás mellett ez melyik ívdarabot jelenti majd.

Az eddig használt modelleken három tehetípus került megadásra:

i.) Önsúly az íveken

$G(\text{rúd}) [\text{kN}] = \text{térfogatsúly} [\text{kN}/\text{m}^3] * \text{keresztmetszet területe} [\text{m}^2] * \text{ív hossz} [\text{m}]$, ahol

a térfogatsúly, mint anyagjellemző szabad paraméterként megadható, a keresztmetszetet a geometriánál korábban már definiáltuk, az ívhosszt pedig a program számolja a diszkretizáció alapján minden ívdarabhoz.

ii.) Önsúly a csomópontokon

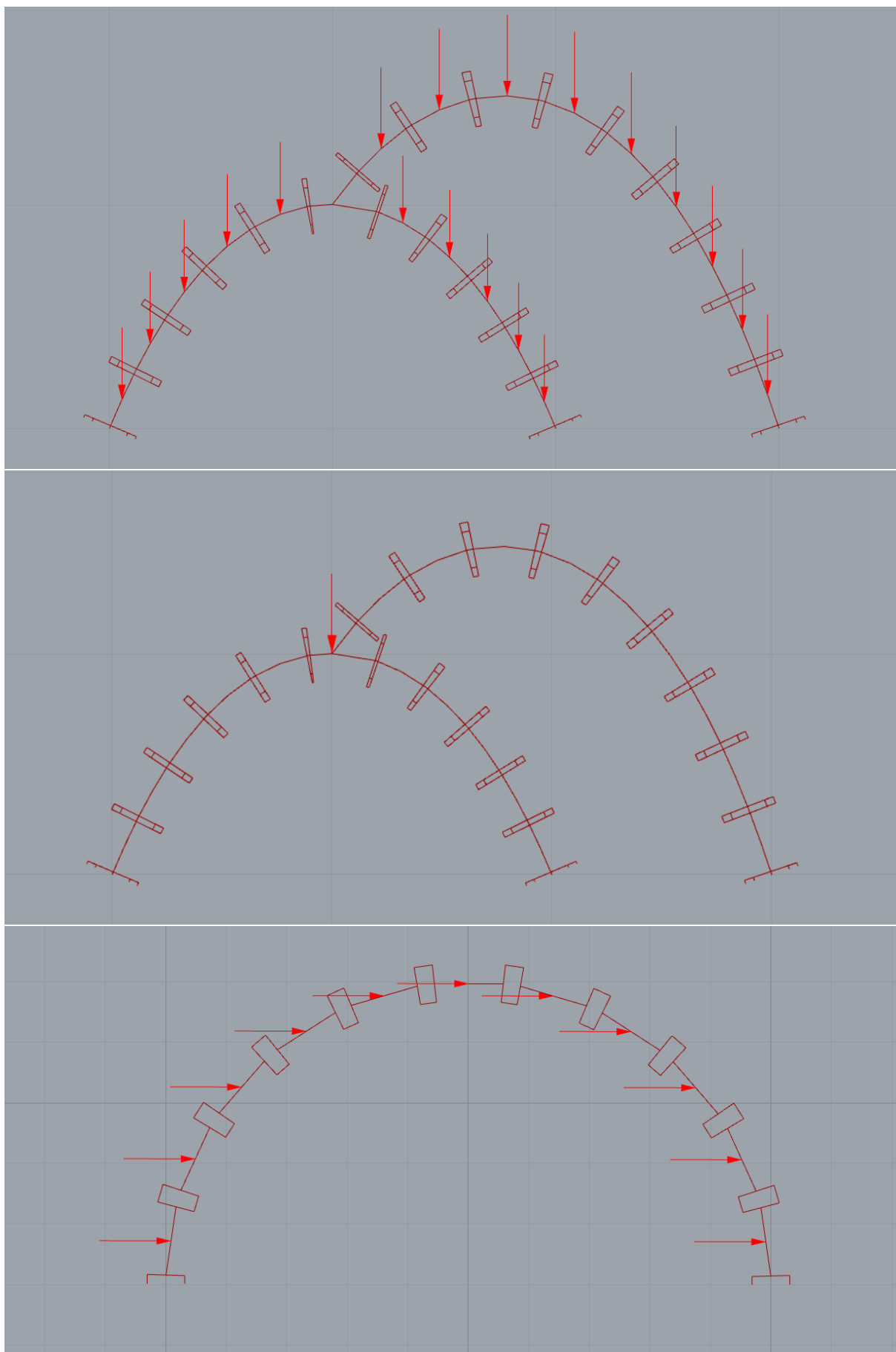
$G(\text{csp}) [\text{kN}] = \text{térfogatsúly} [\text{kN}/\text{m}^3] * \text{térfogat} [\text{m}^3]$, ahol

a térfogatsúly a fenti esettel megegyezik, a térfogat számítása viszont összetettebb – ha a levágott ívdarabok hosszának összegét vennénk, megszorozva a keresztmetszettel, akkor az átfedések miatt túl nagy erőt kapunk, ezt valahogy korrigálni kellene. A pontos megoldás az lenne, ha már a csomópont kiterjedésekor valahogy kiszámolnánk a csatlakozó rudak uniójának térfogatát, viszont a program jelenlegi verziójában egyszerűbb közelítésnek tűnt a vágóelemként definiált gömb térfogatát számolni, ez kisebb felülbecslést jelent az első megoldásnál, a későbbiekben pedig bármikor pontosítható.

iii.) Változó teher

A nyomásvonal alakja adott teheralakra állandó, ezért, amennyiben nem egy statikus állapotot akarunk modellezni, hanem egy külső hatást, amire a nyomásvonal kellő mértékben kimozdul a keresztmetszetből, szükséges egy változó intenzitású erő is, aminek az állandó terhekkel vett szuperpozíciójából adódik majd a változó nyomásvonal is. Az erő paramétere módosítható manuálisan, valamint egy C# modul segítségével, amely gombnyomásra elindul (előre megadott lépésekben) és megáll.

Változó terhek közül az egyik tipikus példa az ív egyetlen pontján ható koncentrált teher volt (2-3. példa), valamint az önsúllyal arányosított, változó oldalirányú teher a tartó egészén (1. példa) – ez lényegében egy eltolás vizsgálat, ami a földrengésre való ellenőrzés egy fajtája.

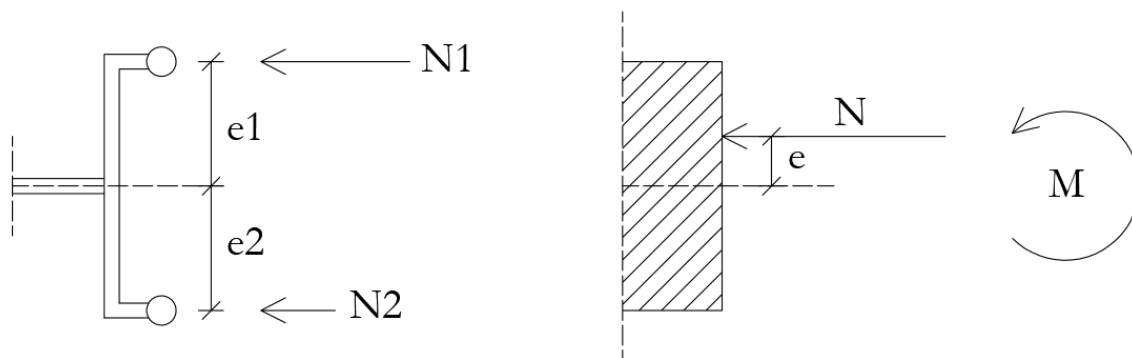


[f.14.] terhek a programban (fentről lefelé: i.) ii.) és iii.)

7. Megoldás menete

7.1. Síkbeli tartó esetén

Minden keresztmetszetben két csukló helyezkedik el, egy az intradoson és egy az extradoson – ezekben mérjük az ébredő normálerőt.



[f.15.] csuklóknban ébredő erők redukálása a keresztmetszetre

$$N = N_1 + N_2$$

$$M = N_1 \cdot e_1 - N_2 \cdot e_2 = N_1 \cdot v - N_2 \cdot v = (N_1 - N_2) \cdot v$$

$$e = M/N = [(N_1 + N_2) / (N_1 - N_2)] \cdot v$$

Ezt a vizsgálatot a program az összes definiált ívszegmens kezdő és végződő keresztmetszetére elvégzi. Mivel a teher növelése nem folytonos, hanem diszkrét ugrásokkal történik, előfordulhat, hogy a számítás lineáris része túlfut azon a ponton, amikor egy keresztmetszetnek be kellene repednie. Ebből következik, hogy adott esetben egyszerre több keresztmetszet is berepedne, sokszor a valóságnak ellentmondó módon, például a kialakítandó csukló közvetlen környezetében, hiszen magasfokú diszkretizáció mellett a nyomásvonal ott is kiugrott a megengedett tartományból. Éppen ezért a program mindig kiválasztja a legnagyobb külpontossággal rendelkező keresztmetszetet, és kizárólag ott hajt végre módosítást, egyszerre mindig egyetlen szabadságfokot véve ki a rendszerből.

A kiválasztott, leginkább igénybe vett keresztmetszetre a következőket hajtja végre:

- i.) Ha $|e| \leq v$, akkor a keresztmetszet működik. A számítás (vagyis a változó teher növelése) folytatódhat a keresztmetszet változtatása nélkül.
- ii.) Ha $e > v$, akkor a nyomásvonal az extradoson kívülre ugrott, tehát az intrados felől berepedt. Ekkor a program eltávolítja az alsó csuklót a rendszerből.
- iii.) Ha $e < -v$, akkor a nyomásvonal az intradoson belülré ugrott, tehát az extrados felől berepedt. Ekkor a program eltávolítja a felső csuklót a rendszerből.

[Megjegyzendő, hogy a bevezetésben foglalt irányelvek alapján elméletileg az $N < 0$ feltételnek is teljesülnie kell, ezt jelenleg a program nem ellenőrzi, mivel az eddig használt teheresetekben ez adta magát.]

Mivel a Grasshopper nem engedélyezi a paraméterek ciklikus visszacsatolását, a csuklók eltávolítása a következőképpen történik:

A modell összeállítása után, de még az elmozdulások számítása előtt egy modul beolvas egy külön memóriába kimentett, Boolean értékeket tartalmazó listát. Ez a lista rendre minden egyes definiált elemre megmondja, hogy aktív, vagy nem, és ezt a modellt visszük tovább – így a számításba már egyedül az aktív elemek kerülnek majd be.

Amikor tehát kivesszünk egy csuklót (valójában két ívdarabot összekötő rúdelemet), a program ennek szándékát rögzíti a külső memóriában tárolt listán, és a legközelebbi beolvasás alkalmával ennek megfelelően alakítja át a modellt.

A beolvasás összeköthető a változó terhet szabályozó elemmel, így csak akkor növeli a terhet a program, ha a modell állapotfüggvénye frissült. Ez természetesen lassítja a működést, tehát mindenképp érdemes megjegyezni, hogy a jelenlegi verzió futása még nem optimális: a későbbiekben ezt a visszacsatolást érdemes egy C# modullal helyettesíteni, ahol az egész számítás és modellszabályozás beépíthető egy FOR-ciklusba, anélkül, hogy külső fájlba való mentés-beolvasást használnánk.

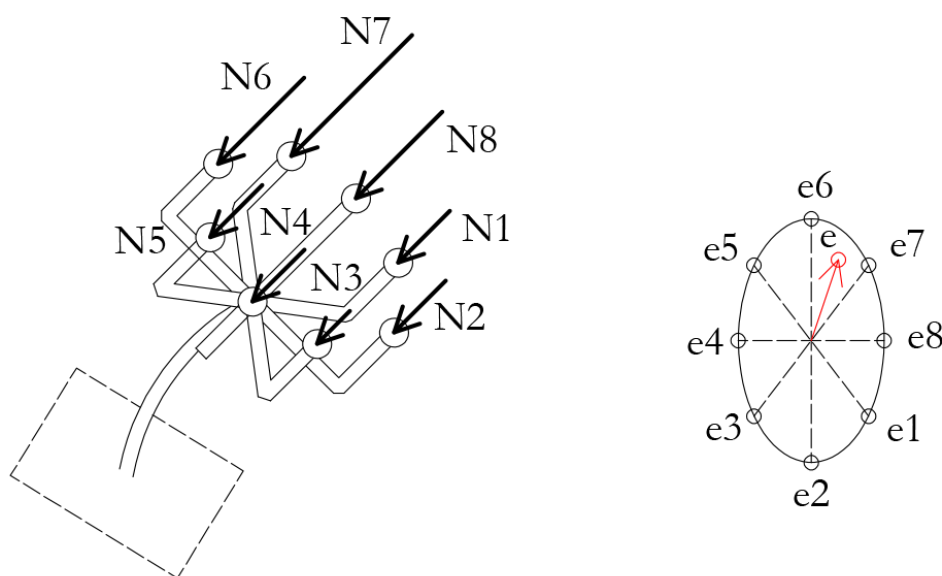
7.2. Térbeli tartó esetén

Térbeli modell esetében a hajlítási igénybevétel ellenőrzése alig különbözik. Mint az a diszkretizáció kérdésénél már előkerült, ilyenkor a keresztmetszet határa nem két pont, hanem egy zárt görbe, véges sok csuklót definiálva. A normálerő a fentiekhez hasonlóan mérhető, viszont a lokális külpontosságok nem skalármennyiségek, hanem a keresztmetszet síkjában, annak középpontjára felírt lokális koordinátarendszerben vett helyvektorok.

$$\underline{\mathbf{N}} = \Sigma \underline{\mathbf{N}}_i$$

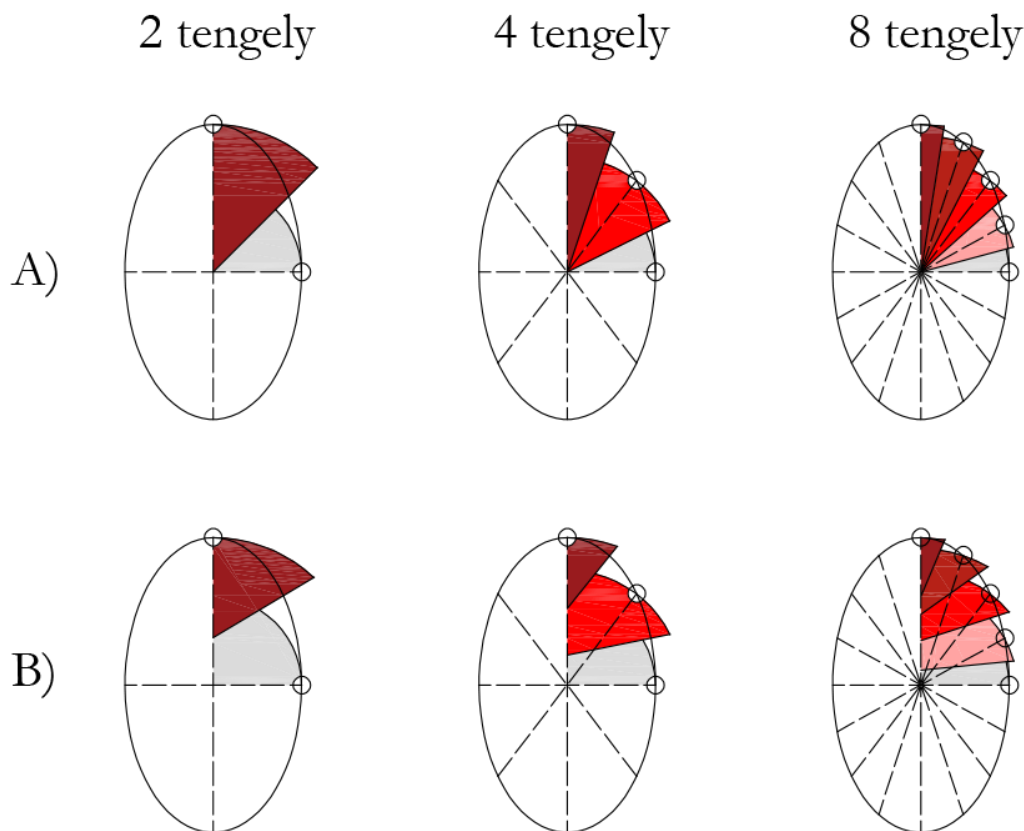
$$\underline{\mathbf{M}} = \Sigma (\underline{\mathbf{N}}_i \times \underline{\mathbf{e}}_i)$$

$$\underline{\mathbf{e}} = \underline{\mathbf{M}}/\underline{\mathbf{N}} = [e_x; e_y; e_z]$$



[f.16.] térbeli rúdvég kapcsolati erőinek keresztmetszetre való redukálása

Ha megkaptuk a keresztmetszet egyetlen erőre redukált igénybevételét, és ezzel a külpontosság mértékét, még mindig nem teljesen egyértelmű, mihez kell hasonlítanunk. Mivel maguk a csuklók előre meghatározott helyen vannak, a megnyílás iránya is ezekhez tud csak igazodni. A legegyszerűbb közelítőeljárás, ha megkeressük, melyik csukló van a legközelebb a ponthoz (szögfelező [f.17. A], távolság [f.17. B], stb. alapján) és annak a külpontosságához hasonlítjuk. A módszer pontossága megfelelő keresztmetszeti diszkretizáció mellett elfogadható lesz.



[f.17.] keresztmetszeti csuklók hatásköre a valós külpontosság elhelyezkedésére

A ferde hajlításon kívül azonban egy másik új igénybevétel is megjelenik: a csavarás. A csuklóban ébredő nyírással eddig nem foglalkoztunk, mivel csak a kövek kicsúszását befolyásolták volna, amit elhanyagoltunk. A csavarás viszont egy sokkal jelentősebb hatás, ami valós tönkremenetelnél is előfordulhat, és erősen függ a geometriai kialakítástól. Ennek részletes vizsgálatát még nem tartalmazza a program – eddig csupán konstans csavarási merevségek voltak definiálva. Amennyiben figyelembe vesszük a súrlódást, vélhetően megadható olyan csavarási ellenállás, ami a külpontossághoz hasonlóan a normálerő függvénye – és így visszacsatolható a modellbe.

Összegezve tehát, a számítás során tehát nem a rugalmas alakváltozásokkal foglalkozunk, hanem a képlékeny csuklók által kialakuló mechanizmussal – minden megnyíló repedés új rúdszerkezetet eredményez, a szerkezet tehát deformációk nélküli formájában frissül.

EREDMÉNYEK

1. Példa

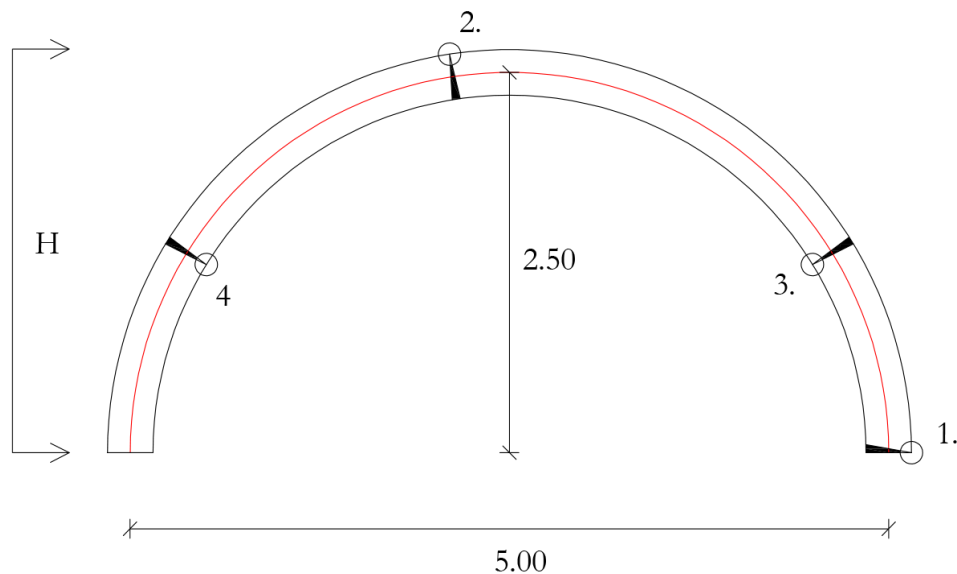
Síkbeli, egyszerű félköríves boltív önsúlyra és növekvő oldalirányú teherre.

Diszkretizáció: 11

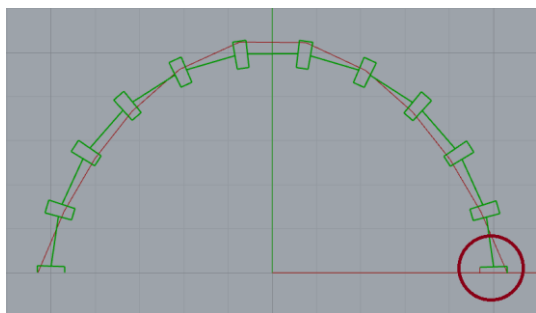
Fesztávolság: 5.0 m (középvonalak közt)

Keresztmetszet: 60 cm × 30 cm

Térfogsúly: 24 kN/m³ (G = 6.25 kN elemenként)



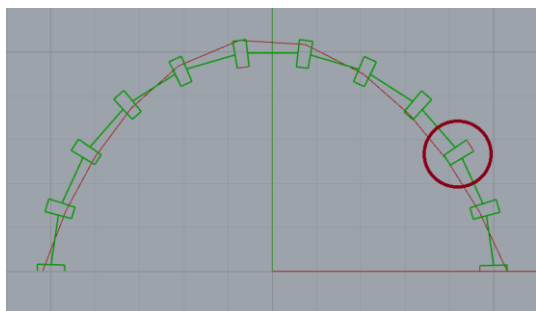
1. csukló kialakulása: $H = G * 0.45\%$



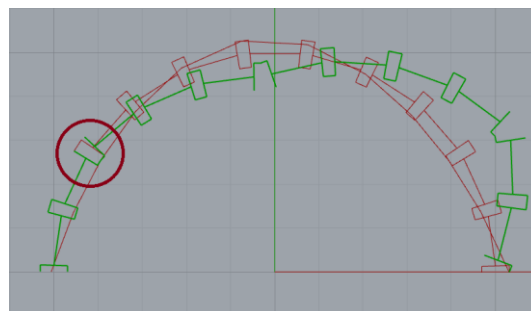
2. csukló kialakulása: $H = G * 2.05\%$



3. csukló kialakulása: $H = G * 5.7\%$



4. csukló kialakulása: $H = G * 7.07\%$



Jelmagyarázat: piros diszkretizált vonal: deformálatlan rúttengely
piros görbe: nyomásvonal aktuális helyzete (a deformálatlan alakhoz képest)
zöld vonal: elmozdult ív

2. Példa

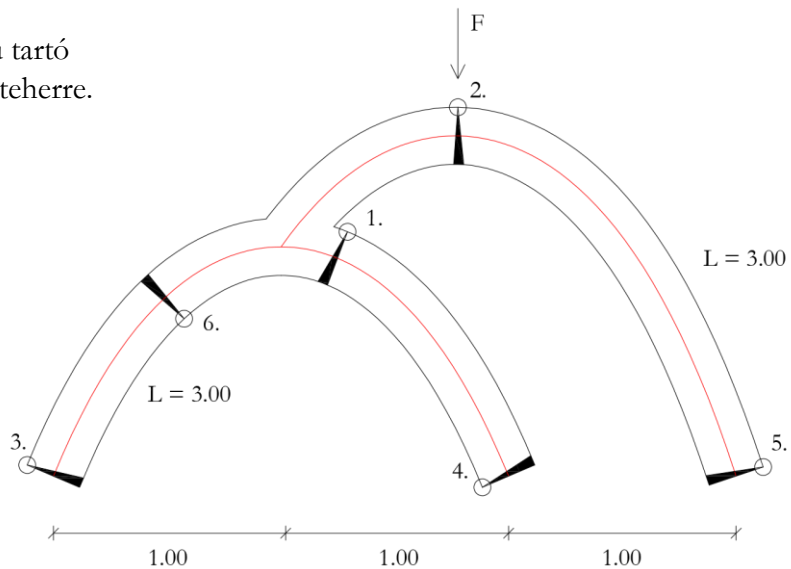
Síkbeli, összetett, láncgörbe alakú tartó
 önsúlyra és növekvő koncentrált teherre.

Diszkretizáció: [5;5;9]

Fesztávolság: 2.0 m + 1.0 m

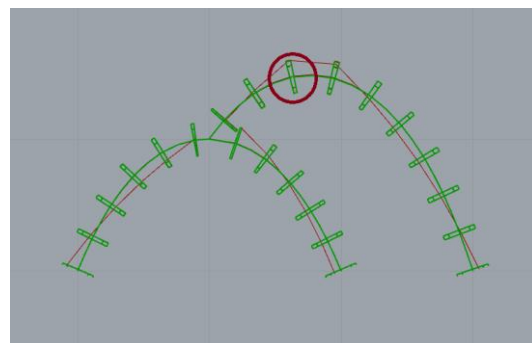
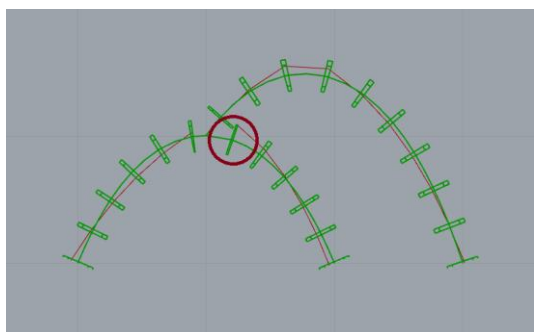
Keresztmetszet: 25 cm × 20 cm

Térfogsúly: 16 kN/m³



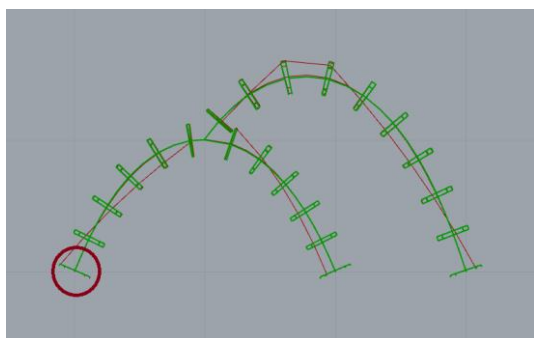
1. csukló kialakulás: $F = 0.6 \text{ kN}$

2. csukló kialakulás: $F = 2.1 \text{ kN}$



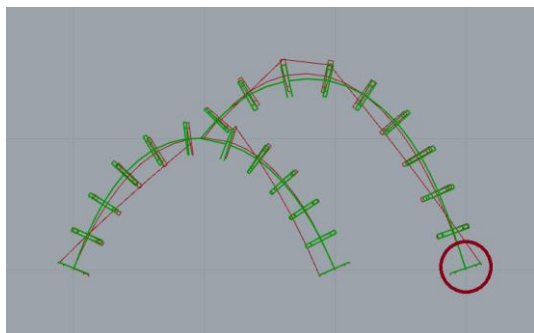
3. csukló kialakulás: $F = 4.0 \text{ kN}$

4. csukló kialakulás: $F = 9.0 \text{ kN}$



5. csukló kialakulás: $F = 14 \text{ kN}$

6. csukló kialakulás: $F = 17.5 \text{ kN}$



3. Példa

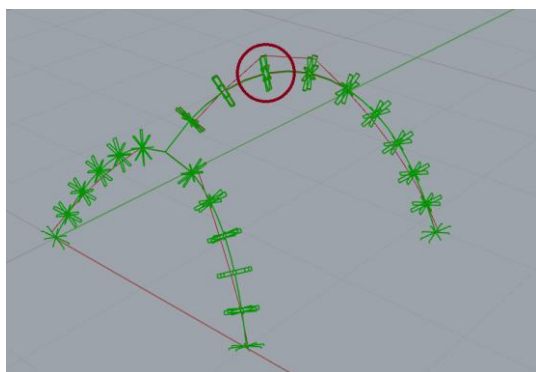
Térbeli, összetett, láncgörbe alakú tartó önsúlyra és növekvő koncentrált teherre.

Az alapgeometria megegyezik az előzővel a következő módosítások mellett:

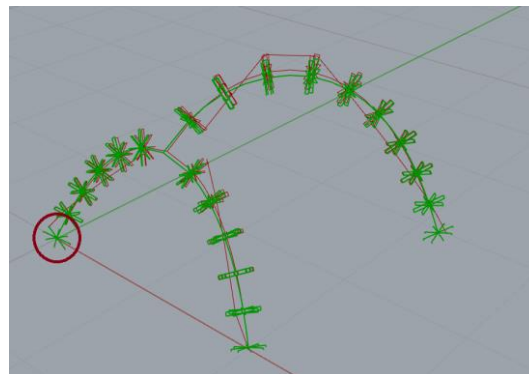
A kisebb ívet 18° -ban megdöntjük a két támasza által meghatározott tengely körül, a nagyobb ívet pedig beforgatjuk 63° -ban függőleges tengely körül. Diszkretizáció, keresztmetszet, térfogatsúly nem változik.

Érdeemes megfigyelni, hogy bár a térbeli esetben elegendő négy képlékeny csukló a mechanizmussá váláshoz, a kedvezőbb geometriai kialakítás (megfelelő szögben való forgatások) hatására mégis ritkábban alakulnak ki, összességében növelve a törőteher értékét.

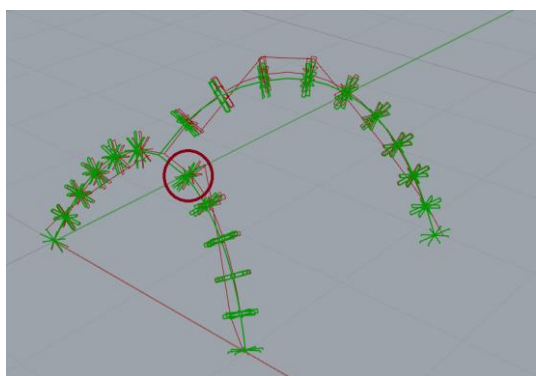
1. csukló kialakulása: $F = 2 \text{ kN}$



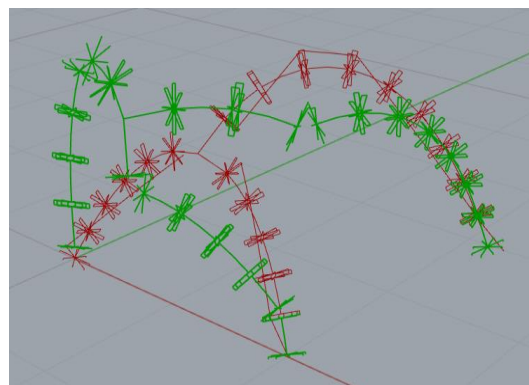
2. csukló kialakulása: $F = 10 \text{ kN}$



3. csukló kialakulása: $F = 11 \text{ kN}$



4. csukló kialakulása: $F = 23 \text{ kN}$

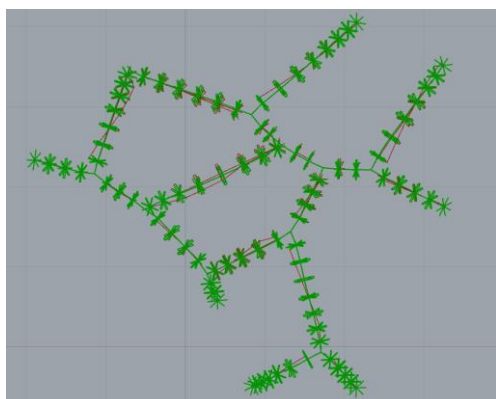


Bónusz Példa

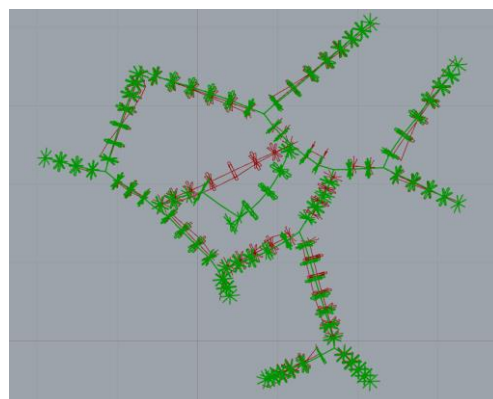
Térbeli, többszörösen összetett tartó önsúlyra és növekvő koncentrált teherre.

Az előző példák egyszerűsége részben amiatt is volt fontos, hogy a megoldások könnyen ellenőrizhetők legyenek akár manuálisan, akár más programok segítségével. 2016-ban azonban a Szilárdságtani és Tartószerkezeti Tanszék Alkotóheti programjának keretein belül épült egy meglehetősen összetett szerkezet, mely összetöréséről részletes dokumentáció áll rendelkezésre.

10. csukló kialakulása: $F = 12 \text{ kN}$



18. csukló kialakulása: $F = 16 \text{ kN}$



Ebben az esetben a programnak még nem sikerült jól eltalálnia a pontos tönkremenetelt. A képen látható, hogy az összedőlés pillanatában a középső ív még a helyén van, és az oldalsó az, amely kiesése elindítja a folyamatot, míg a program által számolt verzióban fordítva. Az látható, hogy a csavarási merevségekre érzékeny a számítás, ami az előző példákban nem okozott gondot, hiszen a geometria úgy volt kialakítva, hogy ilyen igénybevétel ne nagyon lépjen fel. Szintén érzékeny pont lehet a csomópontok környezetében az ívek visszavágásának mértéke, és ehhez kapcsolódóan a diszkrétizáció. A közbenső csuklók kialakulása ugyanakkor mutat hasonlóságot a valósággal, és a nyomásvonal kitéréseinek iránya is biztató. Ez a példa tehát további vizsgálatokat igényel, ami a későbbiekben segítheti majd az összetettebb térbeli tartók helyes számítását. (Valóságban mért törőteher: 7.7 kN)



[f.18.] szerkezet a tönkremenetel pillanatában

ZÁRÓ GONDOLATOK

Mindenképp fontosnak tartom megjegyezni, hogy a program még bőven fejlesztés alatt áll. Többször említettem, hogy egyes funkciók, melyek jelenleg csak manuálisan érhetőek el, a későbbiekben könnyedén automatizálhatók; hogy a futási idő, vagy az elemszám még több módon javítható; hogy egyes helyeken talán nem elég pontos közelítésekkel dolgozik, és így tovább.

A program lényegi része viszont, az a mechanikai vizsgálat, amely elvégzésére létre lett hozva, alapvetően működik. Összetettebb térbeli rendszerek esetén még vannak megválaszolandó kérdések, de ott sem áll távol attól, hogy elérje azt, amit síkbeli tartók esetén sikerült.

Összegezve tehát, a program képes tetszőleges geometriájú ívrendszerek tönkremenetelét vizsgálni Heyman-i elvek alapján. Ez azért is lényeges, mert bár a végeselemes számítás módja miatt van némi jelentősége a merevségeknek, ám mindaddig, amíg nagyságrendileg reális értékek között mozog, ez nem befolyásolja a tönkremenetel módját, és a törőteher értékét is alig. Nem tudok róla, hogy volna jelenleg hasonlóan kevés nemfizikai paraméterrel dolgozó program, mindenesetre a későbbiekben érdemes lehet az eredményeket összehasonlítani más megoldási módszerekkel, például DEM környezetben végzett vizsgálatokkal.

Az egyes képlékeny csuklók kialakulásának időpillanatai (teherértékei) kimenthetők a törőteherhez hasonlóan, így végigkövethető a folyamat egészen a mechanizmus kialakulásáig, az eredmények pedig könnyedén visszaellenőrizhetők akár szerkesztéssel is.

Egy ilyen programmal már lehetőség nyílik arra, hogy nagy mennyiségben, sok paraméter mentén teszteljünk ívrendszereket különböző hatásokra, ami egy meglehetősen nagy lépés a kezdetben megfogalmazott szándék, a gótikus bordás boltozatok osztályozása felé. A program optimalizálása mellett tehát ez tekinthető a kutatás várható folytatásának.

FELHASZNÁLT IRODALOM

[1] Jacques Heyman: The Stone Skeleton, International Journal of Solids and Structures, 1966. Vol. 2, pp. 249 to 279.

[2] C.A. Coulomb: Essai sur une application des règles de maximis et minimis a quelques problèmes de statique, relatifs à l'architecture. Mémoires de Mathématique et de Physique présentés à l'Académie Royale des Sciences, p. 343. (1773.)

[3] A. J. S. Pippard and J. F. Baker: The Analysis of Engineering Structures, 2nd Edition, Edward Arnold (1943.)