# DIPLOMADOLGOZAT

TANÁCS KORNÉL gépészmérnöki szak

Gödöllő 2023



# Magyar Agrár- és Élettudományi Egyetem Szent István Campus Gépészmérnöki Szak

## Szárítóberendezésben kialakuló szemcsemozgási folyamatok modellezése

Belső konzulens:	Dr. Keppler István	
	egyetemi tanár	
Külső konzulens:	Dr. Varga Attila	
	szimulációs mérnök	
Készítette:	Tanács Kornél	
	VAQA8U	
	levelező tagozat	
Intézet/Tanszék:	Műszaki Intézet,	
	Gépszerkezettani Tanszék	

Gödöllő 2023

# TARTALOMJEGYZÉK

1. Bevezetés és célkitűzések	1
2. Szárítás és szárító berendezések a mezőgazdaságban	2
2.1. Szárítás	2
2.2. Berendezések	
2.2.1. Gravitációs rendszerű szárítók	4
3. Szemcsés halmazok	
4. Diszkrét elemes modellezés	14
4.1. A DEM kialakulása	14
4.2. A DEM fogalma	16
4.3. A modell elemei	16
4.4. Mozgásegyenlet	
4.5. Időintegrálási eljárások	
4.6. Csillapítások	
4.7. Kapcsolati modellek	
4.7.1. Lineárisan rugalmas kapcsolati modell	
4.7.2. Hertz-Mindlin csúszásmentes kapcsolati modell	
4.8. Alkalmazások	
4.9. A YADE DEM bemutatása	
4.9.1. A program használata	
5. Kalibráció	
5.1. A használt anyagok és módszer	
5.2. A kalibráció menete	
5.3. Az eredmények feldolgozása	
6. Szárítóberendezés szimulációja	
6.1. A súrlódási tényező hatása az együtt haladásra	
6.2. A súrlódási tényező hatása a sebességeloszlásra	
7. Összefoglalás	
8. Irodalomjegyzék	
9. Summary	
10. Mellékletek	
10.1. A kalibráció szimulációja	59
10.2. A szárítóberendezés szimulációja	
10.3. Nyilatkozatok	74
10.4. További ábrák	77

### 1. BEVEZETÉS ÉS CÉLKITŰZÉSEK

A mezőgazdasági termények szárításánál gyakori probléma az egyenetlen száradás, amikor a halom egyik része teljesen megszárad, vagy akár túl is száradhat, míg a másik része még nedves. Az egyenetlen szárítás jelentős veszteségeket és költségnövekedést eredményez. Ezeknek a veszteségeknek a minimalizálásához szükséges az anyagáramlási folyamatok megértése a szárítóberendezésekben. Ehhez használható a diszkrét elemes módszer (DEM), amelyet kifejezetten a szemcsés anyagok modellezésére hoztak létre.

A diplomamunkám témája a mezőgazdaságban található szárítóberendezések belső kialakításának vizsgálata, geometriai paramétereinek változtatása és annak hatása a szárítási folyamatra, valamint a szárítón belüli anyagmozgási folyamatokra. A dolgozatban bemutatom az általam elkészített kalibrációs és szimulációs programokat, melyeknek a célja a műszaki gyakorlat számára megfelelő pontossággal rendelkező, a szárítási folyamatot jól közelítő modell készítése, aminek szimulációs igényei minimálisak, mind idő, mind számítástechnikai szemszögből.

# 2. SZÁRÍTÁS ÉS SZÁRÍTÓ BERENDEZÉSEK A MEZŐGAZDASÁGBAN

#### 2.1. Szárítás

A szárítás számos iparágban létfontosságú, ami nélkül a végtermék előállítása és tárolása szinte lehetetlen lenne. A szárítási folyamatok energiaigényesek és gyakran környezetszennyezőek, mivel fosszilis tüzelőanyagokat használnak hőenergiaként. A szárítás az iparosodott országokban az energia 15-30%-át fogyasztja. Az Egyesült Államokban, Angliában, Franciaországban, Kanadában és Magyarországon az energiaigény 10-15%-át, Dániában és Németországban pedig 20-25%-át teszi ki. A világgazdaság növekedésével ezek a számok is növekedni fognak. (Mujumdar & Beke, 2002)

Ezért a szárítási technológia fejlesztőinek energiahatékony eljárásokat kell megvalósítaniuk. A mérnöknek követnie kell a technológiai fejlődést, és nyitottnak kell lennie az új tudományos felfedezésekre, hogy elérje ezt a célt. Ismertsége ellenére a szárítási kutatások csak néhány évtizeddel ezelőtt kezdődtek. Az élelmiszeripar, a mezőgazdaság, a biotechnológia, a gyógyszeripar, a vegyipar és a fafeldolgozás mindmind alkalmazza a szárítást a gyártási folyamatokban, mivel a nedvességtartalom károsíthatja az anyagokat a feldolgozás és a tárolás során. (Mujumdar & Beke, 2002)

Szárításról akkor beszélünk, amikor a szilárd vagy gáznemű anyagok fázisváltozás során nedvességet veszítenek, az anyagok nedvesség- és szárazanyag-tartalma megváltozik és félig szilárd vagy folyékony nyersanyagokból szilárd termékké alakítja azokat. Az energiaátadásnak négy típusa van: konduktív, konvektív, sugárzásos hőközlés és térfogati hőközlés. Az ipari szárítók több mint 85%-át a konvektív szárítók teszik ki. A szárítás levegővel vagy füstgáz-levegő keverékkel történik. (Mujumdar & Beke, 2002)

A konvektív szárítási eljárás levegő vagy füstgázok segítségével hőenergiát juttat a szárító kamrába az anyag nedvességének elpárologtatására. A hőenergiát hordozó gáz, más néven szárítóközeg, feladata a hőenergia elnyelése és átadása, valamint a nedves termény által kibocsátott vízgőz felvétele és a szárítóhelyiségből való kivezetése. A

legtöbb konvektív szárítási folyamathoz légköri nyomást használnak, ez az atmoszférikus szárítás. (Mujumdar & Beke, 2002)

#### 2.2. Berendezések

A következőkben a mezőgazdaságban használt szemes-termény szárítókat és azok felépítését mutatom be.

A termény fölösleges víztartalmának eltávolítására használt szárítótér általában egy vagy több azonos szárítózónából áll. Szerkezeti kialakításuk szerint a szárítókat három nagy csoportba sorolhatjuk: a kényszer anyagmozgatású, a gravitációs anyagtovábbítású toronyszárítók és a vastagrétegű szárítók. (Mujumdar & Beke, 2002)

A kényszer anyagmozgatású szárítókban egy vízszintesen elhelyezkedő, villanymotorral hajtott szárítófelületen (szalagok, tálcák) mozgatják a terményt a berendezésen belül. A szárítás a továbbítás sebességétől, a rétegvastagságtól és a szárítási hőmérséklettől függ, ezek segítségével szabályozható. (Mujumdar & Beke, 2002)

A gravitációs toronyszárítók alapelve, hogy a termény a saját súlyánál fogva halad végig a torony felső betápláló részétől a kitároló szerkezetig. Ez a mozgás lehet folyamatos és szakaszos. A szárítóközeg általában adott sorminta szerint kialakított légvezetékeken keresztül van átvezetve a nedves terményen. A száradás sebessége a szárítóközeg hőmérsékletével és mennyiségével, valamint a kitároló szerkezeten áthaladó térmennyiség beállításával változtatható. Különböző belső kialakításuk szerint megkülönböztetünk: aknás, csörgedeztető és terményoszlopos berendezéseket. (Mujumdar & Beke, 2002)



ábra Gravitációs anyagmozgatású terményoszlopos szárítótér kialakítások

 a) perforált lemezzel határolt oszlop, b) zsalunyílással ellátott oszlop,
 c) cikcakkban vezetett oszlop, d) váltakozó légátvezetésű oszlop
 1—légáram, 2—anyagáram (Mujumdar & Beke, 2002)

A vastagrétegű szárítók a termény tárolására is szolgálhatnak. Itt több méter vastagságban található a száradó anyag, így sokkal több időt igényel a szárítás is, mint az előzőekben említett kényszertovábbítású és gravitációs anyagmozgatású szárítóknál. (Mujumdar & Beke, 2002)(Mujumdar & Beke, 2002)

#### 2.2.1. GRAVITÁCIÓS RENDSZERŰ SZÁRÍTÓK

A Cambell-szárító engedélye szolgált a hagyományos keresztáramú terményszárítók helyi gyártásának alapjául, amelyek az országban meglévő szárítóberendezések többségét teszik ki. Ezek a szárítók csörgedeztető rendszerű, keresztáramlásos, aknás berendezések. Elsősorban kukorica szárítására szánták őket, azonban alkalmasak hüvelyesek és olajos magvak szárítására is, ahogy azt a gyakorlat bizonyította. Előnyük a meglehetősen egyszerű kialakításukban rejlik, viszont némileg magas az energiaigényük és a szárításuk sem egyenletes. Elsődleges elemei (2. ábra) a betárolóegység, az előtároló, a szárítótorony, a kitárolóegység, szellőztetőegységek, valamint a tüzelő- és vezérlőberendezéseknek helyet adó épület. (Mujumdar & Beke, 2002)



 2. ábra A klasszikus keresztáramú szárító főbb szerkezeti egységei (ikertelepítés) l—felhordó, betároló, 2—előtároló, 3—torony, 4—meleglevegő—kamra,
 5—vezérlóterem, 6—tűztér, 7—meleglevegő—ventilátor, 8—hűtőlevegő—ventilátor (Mujumdar & Beke, 2002)

A nedves terményt serleges elevátorral és ejtőcsővel szállítják az előtároló tartályba, a fogadógaratból. Az előtároló tartály két célt szolgál. Segít biztosítani, hogy a szárítótorony folyamatosan telítve legyen, és segít megakadályozni, hogy a szárítóközeg elszökjön a torony tetejéből. A szárítótoronyban háromszög keresztmetszetű, alul nyitott keresztirányú légcsatornák találhatók. A forrólevegőkamra bemeneti oldala a légcsatornákhoz csatlakozik, a kimeneti végeken pedig billenő ajtók vannak felszerelve. A termény szárítás közben függőlegesen ereszkedik lefelé, aminek sebessége a kiadagoló sebességének változtatásával szabályozható. A kiürítő egységet egy előre meghatározott ideig szakaszosan működtetik, és ürítési ciklusok hossza állítható. (Mujumdar & Beke, 2002)

Bár a hűtési és a szárítási zóna technikailag nem különül el egymástól, a légcsatornák elrendezése és az a tény, hogy mindkét terület nyomás alatt van, megakadályozza a

hűtőközeg és a szárítóközeg jelentős keveredését. Az égetőegység közegét egy kettős beömlésű radiál ventilátor szívja be a meleglevegő-kamrába. A hűtőventilátorok a torony szabadtéri része alatt helyezkednek el, és egy közös tengely hajtja őket, melyek a hűtőközeg mozgatására szolgálnak. (Mujumdar & Beke, 2002)

A szárítóközeg ventilátor és az égetőmű egy térben található. A hőenergiát két szabályozott, olaj- vagy gázégő állítja elő, elkülönülő saját hengeres égéstérben. A hőellátó épület égő oldala nyitott felületű a környezeti levegőt a főventilátor innen tudja beszívni, a felmelegítéshez. (Mujumdar & Beke, 2002)

Az égőket vezérlő rendszer, az elektromos egységek reteszelése, a bejövő szárítóközeg hőmérsékletét, és az utószárító zóna végén lévő anyag hőmérsékletét mérő hőmérők, a légveszteségkapcsoló, a hővédelem és a leürítés jelzéseit mind a gépezet automatizált rendszer részét képezik. A tüzelőberendezéssel egy épületben, de egy másik helyiségben található automatika és a különböző beavatkozó eszközök vezérlőpanelje. (Mujumdar & Beke, 2002)

Az elsősorban a beruházási költségeket és a szerkezeti egyszerűséget fontosnak tartó szemlélet elhagyását az 1970-es évek energiaválsága okozta. A szárítók fejlesztésének alapvető kritériumaként az energiacsökkentés és a termék értéke lett a trend. A hagyományos keresztáramú szárítók százainak energiatakarékos modellekké történő átalakítása ennek az eredménye volt. (Mujumdar & Beke, 2002)

A különböző tervezési megközelítések alapjául két alap ötlet szolgált. Az egyik az, hogy a szárítózónán áthaladó közeget egy visszafordító kamra segítségével visszavezetik a szárító terményen keresztül, ahelyett, hogy a szabadba engednék. Ennek eredményeként a szárítóból kilépő levegő páratartalma közel 100%-os, szemben a korábbi kialakításoknál alkalmazott nagyjából 40%-os páratartalommal. A beállítás csak úgy valósítható meg, ha az adott berendezésen kisebbre méretezzük az áramlási keresztmetszetet és a terményréteg ellenállását növeljük. A gép teljesítménye így némileg csökken. A másik változtatás azon az alapelven nyugszik, hogy az utószárítási zónából alacsony páratartalommal és magas hőmérséklettel kilépő levegőt egy adapterrel fel lehet fogni és vissza lehet vezetni a tűztérbe, attól függően, hogy a szárítmány milyen nedvességtartalommal rendelkezik a kezdetben. Ebben az esetben, ha a szárítót megfelelően tervezték meg, akkor a kifelé távozó levegő szinte telített, mégsem következik be teljesítményveszteség, (a légtechnikai jellemzői nem változnak jelentősen.) A hűtőközeg recirkulációját mindkét lehetőség magába foglalja, azonban ebben az esetben a recirkulált közeg tisztítása kulcsfontosságú. (Mujumdar & Beke, 2002)

A keresztáramú szárítók fejlesztésének egyik eredménye az eltolt terelőlapokból álló tölcsérre hasonlító belső szárítótorony kialakítású szárítók megjelenése. Ennek előnye, hogy a szárítóanyag egyenletesebben keveredik, és a szárítóközeg bizonyos részeken nem érintkezik a terménnyel, ami javítja a hő- és nedvességmező homogenitását. Hátránya, hogy eltömődéshez vezethet, és megnöveli a berendezés költségét. (Mujumdar & Beke, 2002)

A legtöbb oszlopos szárító egy serleges felhordót használ a nedves terménynek a gép tetején lévő előtároló részbe történő szállítására. Az anyag egyenletes adagolása egy csigának köszönhetően történik. A termény a saját súlya miatt halad lefelé az előszárító zóna felső részében egy gabonaoszlopot alkotva. A termény a tényleges előszárító alján két oszlopra oszlik, cik-cakk alakban. A ventilátor a fűtőegység által termelt forró levegőt a kialakult oszlopok közötti résbe nyomja. (Mujumdar & Beke, 2002)

A száradó terményoszlop a berendezésen különböző sebességgel halad át a csigás kitároló berendezésnek köszönhetően, aminek vezérlő óraművel a kiürítési gyakorisága módosítható. A szárító működése és energetikai tulajdonságai kiválóak és mivel a belseje egyszerűen tisztítható, karbantartható, ezért a vetőmagszárításban is kedvelt megoldás. (Mujumdar & Beke, 2002)

Az első generációs gravitációs szárítók általában környezetvédelmi szempontoknak nem teljesen feleltek meg. Különösen a por és a szálló törmelék kibocsátásának csökkentése érdekében egy légkamrát alakítottak ki, ahol ciklofan ventilátorokat alkalmaztak, amik a kiemelkedő hatásfok mellett, kellően szűrik a távozó levegőt és a szárító zajszennyezését is mérsékli, lehetővé téve annak lakott területen való alkalmazását is. (Mujumdar & Beke, 2002)

A szárítók második generációjával szemben a szárított termék minősége és az energiahatékonyság, valamint az univerzális használhatóság volt a három legfontosabb fejlesztési követelmény. A szárítóközeg egyedi, testre szabott légvezetésével és annak regenerálásával (újra felmelegítésével) alacsony energiafogyasztást ér el kiváló teljesítmény mellett. (Mujumdar & Beke, 2002)



3. ábra Energiatakarékos univerzális szemestermény-szárító felépítése (Mujumdar & Beke, 2002)

A nedves terményt egy serleges felhordó mozgatja a befogadó garatból az előtárolóba. A kitároló szerkezet sebesség beállítása után az anyag a gravitáció segítségével halad lefelé a toronyban, amely aknás csörgedeztető rendszerű és három függőleges részre (két szárítási zóna és egy hűtési zóna) van osztva. A cellás kitároló egységek a hűtőzóna alatt találhatók, melyeket egy villanymotor működtet, aminek fordulatszáma állítható. (Mujumdar & Beke, 2002)

A szárításhoz szükséges hőenergiát két folyamatos üzemű gáz- vagy olajégő biztosítja. A környezetből beszívott közeg előmelegítésére a hengerek között légcsatornával ellátott égőkamra kéthengeres kialakítású. A két tűztér különálló négyzetes alakú burkolatokban van elhelyezve. A forró szárítóközeget a bal oldali szellőzőegység ventilátora szívja be a keverőkamrába. A hűtési zóna, ahonnan a hűtőközeg beáramlik és keveredik a forró közeggel, szintén a keverőkamrához csatlakozik. A ventilátor az így keletkezett szárítóközeget a légkamrába hajtja, ahol az az alsó szárítózónába kerül. A közeg kilép a terményréteget, és a jobb oldali légcsatornán keresztül a keverőtérbe jut, miután áthaladt egy gyűjtőkamrán. A ventilátor szívóereje szabályozza a keverőkamra légkeringését. (Mujumdar & Beke, 2002)

A jobb oldali tűztéren átnyomott hígított füstgáz összekapcsolódik az utószárítóból kilépő szárítóközeggel, és ez a folyamat regenerálja a szárítóközeget. A felső szárítózónán keresztül haladva ez a meleg levegő a szabadba kerül. Ennek megfelelően a szárító a fent említett alapvető regenerációs üzemmódban működik, de a szárítási igényektől függően a hőellátó rendszere és a légcsatornában lévő vezérlő és szabályozó elemek a hagyományos keresztáramú, hűtőlégvisszavezetéssel kevert kereszthuzat megvalósítására is alkalmazhatók. szárítandó tulajdonságainak А anyag figyelembevételével kialakított technika a légáram és a légmennyiség, valamint a szárítóközeg hőmérsékletének változtatásával a szabályozási lehetőségek igen széles skálájához igazítható. (Mujumdar & Beke, 2002)

A 4. ábra egy kompakt szárítót ábrázol, amely a regeneráló-recirkulációs vízelvonási módszerre épül. Csörgedeztető rendszerű terménymozgatás és kereszt-ellenáramú szárítást alkalmaz, a légcsatornák elhelyezése révén. A kompakt kialakításnak köszönhetően alacsony zajszint garantált, és a porkibocsátás minimálisra csökkenthető. A készülék pneumatikus ürítési technológiával rendelkezik, amely folyamatosan, finoman szabályozott ürítést tesz lehetővé. (Mujumdar & Beke, 2002)



4. ábra Regeneráló-recirkulációs kompakt szárító (Mujumdar & Beke, 2002)

A hagyományos technológia mellett a szárítók harmadik generációja számos tervezési változást is tartalmaz. Alternatívaként a hűtőközeg és a használt szárítóközeg egy része is újrahasznosítható. Ez a választás különösen akkor előnyös, ha a kezdeti nedvességtartalom magas. (Mujumdar & Beke, 2002)

A szárító hőellátó egysége figyelemre méltó, ugyanis választható a forró vízzel, gőzzel, gázzal vagy olajjal történő fűtés is. Az utóbbihoz rendelhető nyomás-, vagy légporlasztásos, vagy atmoszférikus mátrixégő. A gáz- vagy olajtüzelésű változatnál a közvetlen és közvetett üzem, vagy ezek kombinációja is lehetséges (5. ábra). (Mujumdar & Beke, 2002)



5. ábra Kombinált tűztér direkt és indirekt üzemmódhoz. (Mujumdar & Beke, 2002)

Az üzemmódválasztó alkalmazásával a tűztérhez csatlakoztatott diffúzor zárható (közvetett üzem) vagy nyitható (közvetlen üzem). A tűztér a többrétegű palástnak köszönhetően többhuzamú hőcserélő felületet hoz létre. A levegő- és füstgázkamrák hőálló rozsdamentes acélból készültek. A gyári adatok szerint a közvetlen hőellátás hatásfoka elérheti a 97%-ot, míg indirekt üzemmódban a 90%-ot! (Mujumdar & Beke, 2002)



6. ábra Légcsatorna kialakítás hőlándzsákkal és állítható kilépőéllel (Mujumdar & Beke, 2002)

Az üzem légcsatorna-rendszere ötletesen van kialakítva (6. ábra). A meleglevegőkamra felőli zárt csatornafedelek egyedi kialakításának köszönhetően megakadályozzák a helyi kondenzációt, az úgynevezett hőlándzsák segítségével, melyek használatával friss meleglevegő juttatható a használt szárítóközegbe. A kiemelkedően magas és ingadozó kezdeti nedvességértékek esetén ez a megoldás különösen hasznos. A kreatív megközelítés segítségével a forrólevegő-csatorna alsó kilépőélének magassága meghatározott határok között szabályozható, így elkerülhető a termény kiömlése vagy túlfolyása. Ez a megközelítés különösen hasznos a recirkuláció alkalmazása esetén, mivel lehetővé teszi a levegő sebességének kismértékű növelését, ami teljesen vagy részben ellensúlyozhatja a magasabb relatív páratartalmú visszakevert levegő teljesítménycsökkentő hatását. Ezenkívül a kilépő keresztmetszet megváltoztatásával változó mértékű turbulencia hozható létre, ami segíti az intenzívebb és homogénebb hő- és anyagáramlást. További előnye, hogy a csatorna széle körül összegyűlt szennyeződések könnyebben eltávolíthatók. A kimeneti szárítóközeg tisztítása és a zajkibocsátás csökkentése érdekében ciklonos porgyűjtő és abszorpciós hangtompító is beépíthető. (Mujumdar & Beke, 2002)

#### 3. SZEMCSÉS HALMAZOK

A szárító berendezésekben kialakuló szemcsemozgási körülmények modellezéséhez a szemcsés anyagok mechanikai viselkedésének jó ismerete szükséges.

Számos területen találkozunk szemcsés anyaghalmazokat és a mérnökök nehezen tudják ezeket modellezni, mivel bizonyos körülmények között szilárd anyagként viselkednek (terhet viselnek, megtartják alakjukat) és máskor folyadékra hasonlító anyagként viselkedik. (Csizmadia, 2009)

A szemcsés anyagok elméleti vizsgálatában alkalmazott közelítések és egyszerűsítések nagy eltérésekhez vezetnek a becslések és a mért eredmények között. A mechanikai modell kiválasztása a legnehezebb része ezeknek a vizsgálatoknak. Sem a folyadékok mechanikája, sem a szilárd testeké nem alkalmas teljesen a szemcsés halmazok modellezésére. Általánosan két módszerrel írhatók le. A klasszikus kontinuummechanikából származó szilárd testek mechanikájával, valamint a diszkrét elemes módszerrel, ami az egyes részecskék mozgásegyenleteinek felírásával modellezi a szemcsék viselkedését. (Bagi, 2007; Csizmadia, 2009)

A szemcsés anyag a kontinuum modellben feltételezhetően folyamatosan tölti ki a térfogatot, így a rendszert jellemző mechanikai mennyiségek leírására folytonos függvények használhatók. A folytonos függvények a halmaz állapotának leírására is használhatók, és differenciálegyenletek segítségével kiszámítható az állapotváltozás, miközben figyelembe kell venni az adott anyag jellemzőit. A szemcsés halmazok anyagviselkedése nemlineáris és nem rugalmas, ami azt jelenti, hogy összenyomásuk során rugalmassági modulusuk folyamatosan változik, és nem térnek vissza a kezdeti állapotukba. (Keppler, Szemcsés anyagok természetes boltozódása, 2006)

A kontinuum alapú numerikus módszerek közül a végeselemes modellezés volt az egyik legnépszerűbb technika erre a célra, de számos problémával jár a nem folytonos problémák modellezésekor. Mivel a végeselemes modellezés elsősorban a folytonosságon alapul, nem tudja pontosan modellezni a nem folytonos viselkedést, és ehelyett nagyon összetett modelleket kell használni ilyen körülmények között. A halmaz szétválása és újbóli összeállása jelenti a legnagyobb modellezési kihívást, mivel ezeket kontinuum modellel nagyon nehéz szimulálni. Ennek eredményeképpen közelítéseket kell keresni és használni.(Bagi, 2007; Fleissner és mtsai., 2007)

### 4. DISZKRÉT ELEMES MODELLEZÉS

Az előzőek alapján belátható, hogy az általános kontinuum-mechanikai alapú mérnöki számítási eljárások a szemcsehalmazok jellegzetességeit vagy egyáltalán nem, vagy csak igen nehézkesen tudják modellezni. Az ilyen problémák numerikus vizsgálatára az 1970-es években jöttek létre az úgynevezett diszkrét elemes modellek, amelyeket a számítógépek gyors fejlődésének eredményeként a mérnöki gyakorlatban az 1990-es évekre már széles körben alkalmaztak. (Bagi, 2007)

#### 4.1. A DEM kialakulása

A diszkrét elemes modellezés (DEM) olyan numerikus módszer, amellyel a szemcsés anyagok, például kőzetek, talajok, porok és szemcsék viselkedését lehet szimulálni különböző körülmények között. A DEM fejlődése az 1950-es évek elejére vezethető vissza, amikor a számítógépes szimulációk először váltak elérhetővé.

Az 1970-es években a kutatók elkezdték alkalmazni a módszert a talajmechanika és a kőzetmechanika tanulmányozására, azzal a céllal, hogy megértsék a talajok és kőzetek viselkedését különböző terhelési körülmények között. A DEM első sikeres alkalmazása valós problémára a részecskeáramlás szimulációja volt egy forgó dobban, amelyet Cundall és Strack végzett 1979-ben.

Azóta a DEM-et széles körben használják számos területen, többek között a geotechnikai mérnöki, bányamérnöki, vegyészmérnöki és anyagtudományi területeken. Az elmúlt évtizedekben jelentős előrelépés történt mind a DEM elméleti alapjaiban, mind a számítási technikákban.

A DEM egyik legfontosabb előrelépése a részecskék közötti erőket leíró érintkezési modellek kifejlesztése volt. A legegyszerűbb érintkezési modell azt feltételezi, hogy a részecskék rugalmas ütközéseken keresztül lépnek kölcsönhatásba egymással, de ennél kifinomultabb modelleket is kidolgoztak, amelyek figyelembe veszik a súrlódás, az adhézió és a deformáció hatásait.

A DEM másik fontos fejlődése a párhuzamos számítástechnika használata volt a nagyobb és összetettebb rendszerek szimulálására. A nagy teljesítményű számítástechnikai erőforrások rendelkezésre állása lehetővé tette a szemcsés anyagok valósághűbb szimulációját, nagyobb részecskeszámokkal, összetettebb geometriákkal és valósághű részecsketulajdonságokkal.

A szemcsés anyagok viselkedésének pontos szimulálásához a DEM segítségével részletesen modellezni kell az egyes részecskék közötti kölcsönhatásokat. Ez általában az egyes részecskék mozgásegyenleteinek megoldását és a szomszédos részecskékkel való érintkezés miatt rájuk ható erők kiszámítását jelenti. Ezek a számítások számításigényesek lehetnek, mivel a mozgásegyenletek numerikus integrálását, valamint nagyszámú részecskére vonatkozó gyakori érintkezési detektálást és erőszámításokat igényelnek.

Emellett a szemcsés anyagok összetett viselkedést mutathatnak, mint például a részecskék átrendeződése, szegregáció és áramlás, amelyek pontos megragadásához kifinomult érintkezési modellekre, konstitutív modellekre és numerikus algoritmusokra lehet szükség. Ezek a modellek és algoritmusok iteratív számításokat és összetett numerikus eljárásokat foglalhatnak magukban, ami tovább növeli a DEM-szimulációk számítási igényeit.

Ráadásul a DEM-ben szimulálandó szemcsés rendszerek mérete is nagyon eltérő lehet, a több ezer részecskét tartalmazó laboratóriumi léptékű kísérletektől a több millió vagy akár több milliárd részecskét tartalmazó ipari léptékű folyamatokig. A rendszer méretének növekedésével a DEM-szimulációk számítási követelményei jelentősen megnőhetnek, ami nagy teljesítményű számítási erőforrásokat igényel a részecskék nagy számának és a közöttük lévő kölcsönhatások összetettségének kezeléséhez.

Az elmúlt években a nyílt forráskódú szoftverek fejlődése is jelentős szerepet játszott a DEM fejlődésében. Számos nyílt forráskódú DEM szoftvercsomagot, például a LIGGGHTS-t, a YADE-t és a MercuryDPM-et fejlesztették ki és bocsátották a kutatóközösség rendelkezésére, megkönnyítve az együttműködést, a tudásmegosztást és a DEM továbbfejlesztését.

Összességében a DEM fejlesztését az az igény vezérelte, hogy jobban megértsük a szemcsés anyagok viselkedését, amelyek kulcsszerepet játszanak számos ipari folyamatban és természeti jelenségben. A számítási erőforrások folyamatos javulásával a DEM valószínűleg még szélesebb körben fog elterjedni és kifinomultabbá válni az elkövetkező években.

#### 4.2. A DEM fogalma

Minden diszkrét elemes modell két alapvető komponensből áll: egymástól elkülönülő elemekből és az egymással való érintkezésükből eredő kapcsolatokból. A modellek egy részében az elemek megfeleltethetők a vizsgált anyagoknak vagy szerkezetnek, de ez nem minden esetben megoldható vagy szükséges. Példaként egy homokréteg megbízható modellezéséhez elegendő lehet egy néhány tízezer szemcséből készült modell a valóságnak megfelelő több millió helyett. Ez persze akkor lehetséges, ha az anyagjellemzők megfelelően meghatározottak. (Bagi, 2007)

Diszkrét elemes modellnek akkor tekintünk egy numerikus eljárást, ha egymástól egyértelműen elkülöníthető elemekből épül fel és ezek az elemek önálló elmozdulási szabadságfokokkal rendelkeznek úgy, hogy a modell képes követni az elemek véges nagyságú eltolódásait, elfordulásait és deformációit. Az elemek között új kapcsolatok jöhetnek létre és a már meglévő kapcsolatok megszűnhetnek. (Bagi, 2007)

Az elemek egymástól független mozgása, egy alapvető különbség a végeselemes modellekhez képest, ahol az elmozdulásmezők a végeselemek érintkezési pontjaiban különféle folytonossági követelményeket teljesítenek. Ezen túlmenően, amíg a szokásos kontinuum-mechanikai alapú keret- vagy törésmechanikai szoftverek legfeljebb az elemek közötti kapcsolatok megszűnését tudják csak követni, egy diszkrét elemes modell az új kapcsolatok létrejöttét is az állapotváltozási folyamat szerves részeként kezeli. (Bagi, 2007)

#### 4.3. A modell elemei

A diszkrét elemes modellezés első lépése a rendszer geometriai modelljének elkészítése, azaz a diszkrét elemek helyének és alakjának, valamint a rendszer geometriájának a megadása. (Bagi, 2007)

A különféle DEM szoftverekben leggyakrabban alkalmazott elemek alakja nagyon változatos. Síkbeli szimulációkhoz használhatunk poliédereket, elliptikus elemeket, köröket (7. ábra).



7. ábra Gyakori elem-alakok síkbeli modellezés esetén (Bagi, 2007)

A térbeli szimulációkhoz pedig gömböket vagy akár gömbökből összetett alakzatokat.



8. ábra Gömbökből összetett gabonaszem (Keppler és mtsai., 2012)

A poliéder alakú elemek helyzetét a sarokpontjaival, míg kör vagy gömb alakú elemekét a középpont és az elem sugarának megadásával lehet definiálni. Összetett alakzatoknál az elem lokális koordinátarendszerében definiálva az elem alakját, majd megadva a vizsgálat globális koordinátarendszerében a lokális koordinátarendszer origójának helyét és a lokális tengelyek irányát, az elem alakja és helyzete egyértelműen leírható. (Bagi, 2007)

A modell által szimulált mechanikai viselkedés két dologtól függ: a geometriai jellemzőktől és az elemek és azok kapcsolatának mechanikai anyagjellemzőitől függ. Ahogy a végeselemes modellekben laboratóriumban meghatározott jellemzőket (pl. rugalmassági modulus, Poisson-tényező) rendelünk az egyes elemekhez, ugyanúgy a diszkrét elemes modellek kialakítása esetén is tényleges fizikai mérések alapján adjuk meg és ellenőrizzük a szükséges anyagi paramétereket. (Bagi, 2007)

#### 4.4. Mozgásegyenlet

A diszkrét elemes modellek a vizsgált állapotváltozási folyamatot kis elmozdulás növekmények sorozataként állítják elő. Az egyes elmozdulás-növekményekre a legtöbb modell esetén érvényesnek tekintjük a kis elmozdulások feltételezését, és a tényleges mozgások ezeknek a kis növekmények összeadásából jönnek létre. (Bagi, 2007)



9. ábra Egy részecske valódi pályája (kék), és számított útja (piros) (Bagi, 2007)

Egy számítási ciklus lényege a rendszer egy elmozdulás-növekményének meghatározása egy adott tehernövekményre, amelyet nem csak külső erők, hanem az elemekre ható még kiegyensúlyozatlan erők is okozhatnak. A ciklus elején

- ismert az elemek helyzete, geometriája, és a kapcsolatok geometriája (pl. normálirány);
- ismert az elemek és kapcsolatok statikai állapota (pl. az épp aktuális kapcsolati erők);
- ismertek az elemek és a kapcsolatok aktuális anyagjellemzői.

A modell ekkor meghatározza a ciklus során létrejövő elmozdulás-növekményeket, majd frissíti a geometriai, topológiai, statikai és anyagi jellemzőket. Ezután következhet az újabb számítási ciklus.

Az egyes módszerek különbsége abban rejlik, hogy egy adott eljárás milyen egységeknek (csomópontoknak, elemeknek) milyen fajta elmozdulásait (eltolódásait, elfordulásait, deformációit) tekinti a modell viselkedésére jellemzőnek. Ebből következik, hogy a modell mozgásegyenlete, ami a számítások alapját képezi más-más alakban írható fel különböző fizikai és geometriai tulajdonságú elemeket használva.

A leggyakrabban használtak a következők:

- végtelen merev elemek
- tartományokra osztott deformálható elemek
- deformálható, de részekre nem osztott elemek.

Az általam használt szoftverben lévő modellben az elemek végtelen merevek és a kölcsönhatások diszkrét, pontszerű tartományokban jönnek létre. Ebben az esetben a modellt N számú merev térbeli elem alkotja, amelyek alakjukat és méretüket nem változtatják. A számítások megkezdéskor ismerjük:

- az elemek helyzetét
- az elemek között átadódó (kapcsolati) erőket és nyomtatékokat
- az elemekre ható külső erőket (pl. gravitáció, közegellenállásból származó erőket)
- továbbá időléptetéses eljárások esetén ismert a szemcsék kezdősebessége is.

A p-edik elem elmozdulásvektora a referenciapont eltolódásának komponenseiből és az elem referenciapont (általában a tömegközéppont) körüli elfordulásának komponenseiből áll (Bagi, 2007):

$$\boldsymbol{u}^{p}(t) = \begin{bmatrix} u_{x}^{p}(t) \\ u_{y}^{p}(t) \\ u_{z}^{p}(t) \\ \varphi_{x}^{p}(t) \\ \varphi_{y}^{p}(t) \\ \varphi_{z}^{p}(t) \end{bmatrix}.$$
(1)

Ezt mindegyik elemre felírva, megkapjuk az egész rendszer elmozdulásvektorát (Bagi, 2007):

$$\boldsymbol{u}^{p}(t) = \begin{bmatrix} \boldsymbol{u}^{1}(t) \\ \boldsymbol{u}^{2}(t) \\ \vdots \\ \boldsymbol{u}^{N}(t) \end{bmatrix}.$$
 (2)

Az iterációs lépésekben megfelelően kicsi elmozdulások esetén a teljes rendszer sebesség- és gyorsulásvektora (Bagi, 2007):

$$\boldsymbol{v}(t) = \frac{d\boldsymbol{u}(t)}{dt}, \qquad \boldsymbol{a}(t) = \frac{d^2\boldsymbol{u}(t)}{dt^2}.$$
(3)

A p-edik elem sebesség- és gyorsulásvektora (Bagi, 2007):

$$\boldsymbol{v}^{p}(t) = \begin{bmatrix} v_{x}^{p}(t) \\ v_{y}^{p}(t) \\ v_{z}^{p}(t) \\ \omega_{x}^{p}(t) \\ \omega_{y}^{p}(t) \\ \omega_{z}^{p}(t) \end{bmatrix}, \qquad \boldsymbol{a}^{p}(t) = \begin{bmatrix} a_{x}^{p}(t) \\ a_{y}^{p}(t) \\ a_{z}^{p}(t) \\ \varepsilon_{x}^{p}(t) \\ \varepsilon_{y}^{p}(t) \\ \varepsilon_{z}^{p}(t) \end{bmatrix}.$$
(4)

A fentiek alapján a *p*-edik elem mozgásegyenlete (Bagi, 2007):

$$\boldsymbol{M}_{p}(t)\boldsymbol{a}_{p}(t) = \boldsymbol{f}_{p}(t, \boldsymbol{u}(t), \boldsymbol{v}(t)),$$
(5)

amelyben az  $f_p$  vektor függ a p elem és p elemmel érintkező többi elem elmozdulásától és sebességétől is. A  $M_p$  mátrix a p elem tömegét és inerciáit tartalmazza és mivel az elemek fordulásai módosítják az inerciát ezért az  $M_p$  mátrix időfüggő (kivéve gömbszimmetrikus elemek esetén).

Hasonlóan az elmozdulásokhoz, hogyha minden szemcsére felírjuk a mozgásegyenletet, akkor a teljes rendszer mozgásegyenletét kapjuk meg (Bagi, 2007):

$$\boldsymbol{M}(t)\boldsymbol{a}(t) = \boldsymbol{f}(t, \boldsymbol{u}(t), \boldsymbol{v}(t)), \tag{6}$$

ahol M(t) az ún. inerciamátrix, az  $M_p(t)$  blokkokból összeállított blokkdiagonálmátrix. A mozgásegyenlet mindig azonos alakú, az alkalmazott elemek típusától független. (Bagi, 2007)

#### 4.5. Időintegrálási eljárások

Az időintegrálási eljárások során az ismert kezdeti állapotból kiindulva, kis időlépéseken keresztül kapjuk meg a rendszer mozgását törekedve arra, hogy a mozgásegyenlet (5) minden vizsgált  $t_i$  időpontban minél pontosabban teljesüljön. (Bagi, 2007)

Differenciálegyenletek kezdetiérték-feladatainak numerikus megoldására többféle módszer létezik. A megoldásra használt explicit eljárás lényege, hogy a  $t_{i+1}$ időponthoz tartozó u és v értékeket a korábbi időpontokhoz (pl. a legegyszerűbb esetben a  $t_i$  időponthoz) tartozó, már kiszámított u és v értékekből, a  $t_i$  időpontban felírva állítják elő a mozgásegyenletet. Az explicit eljárások nagy előnye az implicit eljárásokkal szemben, hogy jóval egyszerűbb és gyorsabb számítást tesznek lehetővé. Hátránya viszont, hogy a számítások során elkövetett hibák összegződnek, és a számított viselkedés egyre messzebbre kerülhet a modellezni kívánt valós folyamattól, mivel nem ellenőrzik minden időlépés végén a mozgásegyenlet teljesülését. Az implicit eljárások ezzel szemben minden vizsgált időlépés végén addig iterálnak, amíg a mozgásegyenlet kellő pontossággal nem teljesül, és csak ezután lépnek tovább a következő időlépés vizsgálatára. Ennek következtében az eredményeik jóval megbízhatóbbak, bár a számítások időigényesebbek is, mint az explicit eljárások esetén.(Bagi, 2007)

A megoldásra használt explicit eljárás lényege, hogy a  $t_{i+1}$  időponthoz tartozó u és v értékeket a korábbi időpontokhoz (pl. a legegyszerűbb esetben a  $t_i$  időponthoz) tartozó, már kiszámított u és v értékekből, a  $t_i$  időpontban felírva állítják elő a mozgásegyenletet.

A korábbiakban említettek szerint tehát a mozgásegyenlet általános alakú, merev szemcsék estén 6 skaláregyenletből áll. Az egyenletek numerikus megoldása a centrális differenciák módszerével történik, ami lényegében az explicit Euler-módszer egy javított változata. Egy  $\Delta t(t_i, t_{i+1})$  időintervallum vizsgálatához a következő adatokra van szükségünk (Bagi, 2007):

- elemek helyzetére  $t_i$ -kor;
- az előző intervallumra vonatkozó átlagsebességekre (v<sub>i+1</sub>);
- t<sub>i</sub>-kor az elemre ható külső erőkre, melyekből kiszámolhatjuk a rendszer minden egyes elemére a rájuk ható eredőerőket és eredőnyomatékokat, a referenciapontra redukálva

A fentieket figyelembe véve a *p*-edik szemcse mozgásegyenletének diszkretizált alakja a következő (Bagi, 2007):

$$\boldsymbol{M}_{i}^{p} \frac{\boldsymbol{\nu}_{i+\frac{1}{2}}^{p} - \boldsymbol{\nu}_{i-\frac{1}{2}}^{p}}{\Delta \boldsymbol{t}} = \boldsymbol{f}_{i}^{p}.$$
 (7)

Másképp:

$$\boldsymbol{v}_{i+\frac{1}{2}}^{p} = \boldsymbol{v}_{i-\frac{1}{2}}^{p} + \Delta \boldsymbol{t} \cdot \left(\boldsymbol{M}_{i}^{p}\right)^{-1} \cdot \boldsymbol{f}_{i}^{p}.$$
(8)

A p elem új helyzete a vizsgált intervallumra jellemző átlagsebesség ismeretében (Bagi, 2007):

$$\boldsymbol{u}_{i+\frac{1}{2}}^{p} = \boldsymbol{u}_{i}^{p} + \Delta \boldsymbol{t} \cdot \boldsymbol{v}_{i+\frac{1}{2}}^{p}.$$
(9)

Az explicit eljárások hátránya, hogy a számításokban a  $\Delta t$  alatt keletkező elmozdulás-növekményeket a számítás törvényszerűen túlbecsüli a pontos megoldáshoz képest. Ez miatt bizonyos esetekben túl nagy belső erők keletkezhetnek az elmozdulások irányban és ezek az erők a következő iterációnál mintha "visszalöknék" a rendszert, a pontos megoldáshoz tartozó helyzetnél is jobban. Így lényegében egyfajta oszcilláló mozgás jön létre a pontos megoldás körül. Ha például egy rendszernek egyensúlyi állapotba kellene kerülnie, akkor ez a számítási eljárás azt eredményezi, hogy a rendszer az egyensúlyi helyzet körül rezeg. Ennek a jelenségnek a minimalizálására a modellekben korlátozzák a felvehető időlépés hosszát, másrészt különféle csillapításokat alkalmaznak.(Bagi, 2007)

#### 4.6. Csillapítások

A csillapítások alkalmazásának az egyik célja, hogy a rendszer mozgási energiáját csökkentsék, az egyensúlyi állapot elérésének érdekében. A másik, hogy a valóságban lejátszódó energiaveszteséget megfelelően közelítsék. A különféle csillapítások közül a legelterjedtebbek a lokális és kapcsolati viszkózus csillapítások. A lokális csillapítás lényege, hogy a mozgásegyenlet (5) jobb oldalon lévő tagjához (elemekre ható erőkhöz) hozzáadunk egy, az elem sebességével ellentétes irányú csillapító erőt, melynek nagysága a megfelelő erőkomponens  $\alpha$ -szorosa.

Példaként a *p* elem mozgásegyenletének első skalár egyenlete csillapítás nélkül (Bagi, 2007):

$$v_{x,i+\frac{1}{2}}^{p} = v_{x,i-\frac{1}{2}}^{p} + \Delta t \cdot \frac{1}{m^{p}} \cdot f_{x,i}^{p}$$
(10)

és lokális csillapítással:

$$v_{x,i+\frac{1}{2}}^{p} = v_{x,i-\frac{1}{2}}^{p} + \Delta t \cdot \frac{1}{m^{p}} \cdot f_{x,i}^{p} - \alpha \cdot \left| f_{x,i}^{p} \right| \cdot \frac{v_{x,i-\frac{1}{2}}^{p}}{\left| v_{x,i-\frac{1}{2}}^{p} \right|}$$
(11)

Az  $\alpha$  tényezőt a felhasználó választhatja meg. Ez a fajta csillapítás azoknak az elemeknek csökkenti a gyorsulását leginkább, amelyek a legtávolabb vannak az egyensúlyi állapot elérésétől. Ebből ered a "lokális" név is. Az egyik tulajdonsága, hogy nem függ az elemek sebességének nagyságától, ezáltal egyensúlyi helyzet akkor is kialakulhat, ha állandó sebesség mellett áramlanak a szemcsék.(Bagi, 2007)

Kapcsolati viszkózus csillapítás esetén a kapcsolati erők mindegyik komponenséhez hozzáadunk egy-egy olyan erőt (ill. nyomatékot), melynek iránya olyan, hogy lassítsa a relatív sebességet és nagysága arányos a kapcsolati relatív elmozdulás-sebesség megfelelő komponenseinek nagyságával.

A normálerőhöz hozzáadódó komponens nagysága (Bagi, 2007):

$$|D_n| = C_n \cdot v_{n,i-\frac{1}{2}}^c$$
(12)

Az energiadisszipáció valóságos folyamatához ez a fajta csillapítás áll a legközelebb.

#### 4.7. Kapcsolati modellek

Mivel a kapcsolatok pontszerűen kicsinyek, koncentrált erőt (és esetleg koncentrált nyomatékot) adhat át egymásnak két érintkező elem. Rendeljünk minden kapcsolathoz egy-egy lokális (n, t, w) koordinátarendszert, amelyben az n tengely a c kapcsolatnál a két elem közös érintősíkjának normálisa, t és w pedig az érintősíkban lévő, egymásra merőléges irányok. Ha p a kapcsolat első, q a második eleme, akkor az n vektor p-ből kifelé és q belseje felé mutat. Ebben a koordinátarendszerben felírhatjuk a kapcsolat első elemére ható kapcsolati erő, illetve nyomaték vektorát, egyetlen vektorba összefoglalva (Bagi, 2007):

$$\boldsymbol{S}^{c} = \begin{bmatrix} N^{c} \\ T^{c}_{t} \\ T^{c}_{w} \\ M^{c}_{n} \\ M^{c}_{t} \\ M^{c}_{w} \end{bmatrix}.$$
(13)

Minden kapcsolathoz más-más lokális koordinátarendszer tartozik. Az  $N^c$  erő a kapcsolatban átadódó normálerő,  $T_t^c$  és  $T_w^c$  a kapcsolati nyíróerő két komponense,  $M_n^c$  a két elem között átadódó csavarónyomaték (*n* tengely körüli relatív elfordulásból),  $M_t^c$  és  $M_w^c$ , a hajlítónyomaték két komponense.(Bagi, 2007)

A kapcsolat mechanikai anyagmodellje egyrészt azt írja le, hogy ezek az erők és a nyomatékok hogyan függenek a két elem relatív elmozdulásától, másrészt teherbírási korlátokat fogalmazhat meg, továbbá, hogy képlékeny állapotba kerülése után a további terhelés esetén hogyan viselkedik a kapcsolat. Az érintkezést alkotó két anyagi pont relatív elmozdulása (Bagi, 2007):

$$\boldsymbol{\delta}^{c} = \begin{bmatrix} \boldsymbol{\delta}_{n}^{c} \\ \boldsymbol{\delta}_{t}^{c} \\ \boldsymbol{\delta}_{w}^{c} \\ \boldsymbol{\theta}_{n}^{c} \\ \boldsymbol{\theta}_{t}^{c} \\ \boldsymbol{\theta}_{w}^{c} \end{bmatrix} = \begin{bmatrix} \boldsymbol{u}_{n}^{qc} \\ \boldsymbol{u}_{t}^{qc} \\ \boldsymbol{u}_{w}^{qc} \\ \boldsymbol{\varphi}_{n}^{qc} \\ \boldsymbol{\varphi}_{n}^{qc} \\ \boldsymbol{\varphi}_{w}^{qc} \end{bmatrix} = \begin{bmatrix} \boldsymbol{u}_{n}^{pc} \\ \boldsymbol{u}_{t}^{pc} \\ \boldsymbol{u}_{w}^{pc} \\ \boldsymbol{\varphi}_{n}^{pc} \\ \boldsymbol{\varphi}_{n}^{pc} \\ \boldsymbol{\varphi}_{w}^{pc} \end{bmatrix}.$$
(14)

#### 4.7.1. LINEÁRISAN RUGALMAS KAPCSOLATI MODELL

A legegyszerűbb mechanikai modell, ami minden ilyen eljárásban megtalálható, mint lehetséges opció, a lineárisan rugalmas, Coulomb-súrlódás kapcsolat. A normálerő itt csak nyomás lehet, és a nagysága az érintkezést alkotó két pont *n* irányú relatív eltolódási komponensével arányos (Bagi, 2007):

$$N^c = k_N^c \delta_n^c; \ N^c \le 0 \tag{15}$$

Mikor a normál erő pozitívvá válik (a két elem eltávolodik egymástól), a kapcsolat megszakad.

A kapcsolati nyíróerő az érintőirányú relatív eltolódással arányos és nagyságának a Coulomb-feltétel szab határt (Bagi, 2007):

$$\begin{bmatrix} T_t^c \\ T_w^c \end{bmatrix} = k_T^c \begin{bmatrix} \delta_t^c \\ \delta_w^c \end{bmatrix}; \sqrt{(T_t^c)^2 + (T_w^c)^2} \le -\nu \cdot N^c$$
(16)

Ha a nyíróerő elérné ezt az értéket, a relatíveltolódás-növekmény a továbbiakban már állandó erő mellett történik (zérus nyírási merevséggel), amíg a relatív eltolódás iránya meg nem változik.

#### 4.7.2. HERTZ-MINDLIN CSÚSZÁSMENTES KAPCSOLATI MODELL

Másik gyakori kapcsolattípus a Hertz-Mindlin-féle közelítés. A kapcsolat normálmerevsége a Hertz-modell szerint becsülhető (függ a nyomóérőtől). R sugarú gömbök esetén (Bagi, 2007):

$$k_N = \frac{3}{2} \frac{\left(R \cdot N\right)^{\frac{1}{3}}}{\left(\frac{3(1-\mu)^2}{4E}\right)^{\frac{2}{3}}}$$
(17)

A normálerő növekménye és a normálirányú relatív eltolódás növekménye között megfogalmazható az előzőhöz hasonló összefüggés(Bagi, 2007):

$$\Delta N^c = k_N^c \Delta \delta_n^c; \ N^c \le 0 \tag{18}$$

A kapcsolat nyíróerejére (a nyomóerőtől függ) Cattaneo (1938), Mindlin (1949), illetve Mindlin és Deresiewicz (1953) munkái alapján ad becslést. Ezekben a cikkekben a levezetések speciális esetekre vonatkoznak, így csak durva közelítésnek tekinthetők.

#### 4.8. Alkalmazások

A diszkrét elemes modellezést számos különböző probléma szimulálására alkalmazhatjuk, a nagy dinamikájú rendszerektől, mint például a szemcsés áramlások, a lassú dinamikájú rendszerekig, mint például a terhelt kontinuumok.(Fleissner és mtsai., 2007)

A dinamikusan mozgó határokkal rendelkező szemcsés áramlásra példaként egy részecske által hajtott vízikerék szimulációját láthatjuk az 10. ábraán. A részecskék közötti tartós kétoldalú kölcsönhatások bevezetésével, amelyek egy bizonyos terhelésnél megtörnek, lehetőség nyílik a kontinuum modellezésére. Ezt a megközelítést egy ortogonális vágási folyamat szimulációján keresztül mutatjuk be, ahol a DEM nyilvánvaló előnyökkel rendelkezik a FEM-hez képest, mivel a forgácsot lehetővé tevő részecske-szétválás szerepel a modellben. Ugyanezeket a kölcsönhatásokat használjuk rugalmas membránok modellezésére is, amely az utolsó példán látható. (Fleissner és mtsai., 2007)



10. ábra Pillanatképek egy kereket hajtó szemcsék szimulációjáról. A részecskék a sebességük szerint vannak színezve. (Fleissner és mtsai., 2007)

A részecskeszimuláció és a többtestes rendszer összekapcsolásának képességét demonstrálandó, szimulációs példának egy vízkereket választottunk. A szimulációban a testek merev makroszemcsékként vannak ábrázolva, amelyek háromszög alakú felületelemekből állnak, amelyeket a szimulációban egyedi részecskékként kezelünk. A vízikerék, amelynek tömegközéppontja a forgástengelyen van, statikus keretben van rögzítve. Az alkalmazott hajtónyomaték negatívan arányos a kerék szögsebességével, így egy rövid átmeneti fázis után a kerék szögsebessége elér egy állandósult állapotot. Ahhoz, hogy a kereket állandó részecskeárammal tudjuk táplálni, egy részecskebeáramlás és egy kiáramlás készült. A részecskék a szimuláció során folyamatosan törlődnek és jönnek létre, hogy memória- és CPU-költséget takarítsunk meg. A vízikerék mozgását az modellben lévő testek mozgásegyenleteinek integrálásával számítjuk ki, a részecskéket erőelemeknek tekintve. (Fleissner és mtsai., 2007)

A DEM-ben vizsgálható vágási folyamatok szimulálására az ortogonális vágás esetét vettük figyelembe. Ez a vágás viszonylag egyszerű esete kétdimenziós feszültségállapottal. Első lépésként a munkadarabot reprezentáló szilárd testet szabályos, kocka rácsba rendezett, azonos gömbök tömegeként generáljuk. Ezután minden szomszédos gömböt viszkoelasztikus-plasztikus erőkkel kötünk össze, ezáltal

egy törhető és szemcsés szilárdtest háromdimenziós modelljét hozzuk létre, amelyet merev, törhetetlen gömb alakú részecskék összetétele alkot. A munkadarabot háromszögek által reprezentált szerszámmal megmunkáljuk. Ha az anyag szakítószilárdságát elérjük, az anyagot összetartó erők eltávolításra kerülnek, és nem állnak vissza. A köztes állapotok néhány pillanatfelvétele az 11. ábraán látható. (Fleissner és mtsai., 2007)



11. ábra Forgács leválasztás szimulációja. (Fleissner és mtsai., 2007)

A részecskék viszkoelasztikus elemekkel való összekapcsolásával olyan membránok szimulálhatók, amelyek eloszló tömegét gömb alakú részecskék reprezentálják. A kötött részecskék között csak a rugós csillapító erők hatnak, míg a nem kötött részecskék között bármilyen más érintkezési törvény, például a Hertz-törvény vagy egy súrlódási modell alkalmazható. A szimulációs példában tetszőleges, de bonyolult és nem konvex 3D-modelleket láthatunk a membránnal kölcsönhatásba lépő geometriaként. A 12. ábraán egy 8300 részecskéből álló membrán látható, amely kölcsönhatásba lép egy merev szoborral. A membrán a gravitáció hatására az arcra esik, és rugalmasan deformálódik, annak alakját követve. (Fleissner és mtsai., 2007)



12. ábra Szimuláció egy rugalmas membránról, ahogy egy szobor arcára hullik. A részecskék sebességük szerint vannak színezve. (Fleissner és mtsai., 2007)

A bemutatott példák egymástól igen eltérők és rajtuk keresztül belátható, hogy milyen széles körben alkalmazható a gépészmérnöki gyakorlatban a diszkrét elemes modellezés.

#### 4.9. A YADE DEM bemutatása

A szimuláció elkészítéséhez a YADE DEM programot fogom használni, amit a következőkben bemutatok.

A YADE (Yet another dynamic engine) egy nagymértékben bővíthető, nyílt forráskódú keretrendszer a diszkrét elemek módszerét alkalmazó numerikus modellekhez. A szoftver C++ nyelven íródott, és a nagyfokú rugalmasság és teljesítmény érdekében Python interfésszel rendelkezik. Ez lehetővé teszi az egymással kölcsönhatásban lévő részecskék millióinak szimulációját. A YADE Linux-alapú operációs rendszereken használható, és elérhető az olyan elterjedt disztribúciók hivatalos tárolóiban, mint az Ubuntu és a Debian; a forráskód szabadon elérhető a GitHubon, és a GPL2 nyílt forráskódú licenc alatt áll. A YADE első verziója 2004-ben jelent meg. A kódot 2007 és 2010 között refaktorálták és konzisztensebbé tették. (Haustein és mtsai., 2017)

A szimuláció beállításához, vezérléséhez és utófeldolgozásához Pythont használnak. A Python kötéseket a BoostPython könyvtáron keresztül valósítják meg. Ez az értelmezett programozási nyelv elrejti a felhasználó elől a viszonylag összetett C++ implementációt. Ez teszi a YADE-et oktatási célokra alkalmas szoftverré, és számos egyetem sikeresen használja a DEM alapjainak bemutatására a hallgatók számára. A YADE rendelkezik egy IPython alapú interaktív héjjal. Emellett lehetőség van a YADE modulként történő importálására a Python szkriptbe. (Haustein és mtsai., 2017)

A YADE sok metaprogramozási technikát használ (kódgenerálás a fordítási idő alatt makrók és sablonok segítségével), ami leegyszerűsíti az új érintkezési törvények vagy motorok programozását, de megnehezíti a kód hibakeresését. A sablonok használata garantálja a szigorú típusparadigmát, de növeli a fordítási időt. (Haustein és mtsai., 2017)

A párhuzamosítás nagyon fontos része minden részecske alapú szimulációs szoftvernek. A YADE OpenMP (nyílt többprocesszoros) technikákat használ a legtöbb fontos ciklusban, hogy növelje a számítási sebességet a többprocesszoros rendszereken. A szimulációk önálló alkalmazásként és kötegelt üzemmódban is futtathatók, ami alkalmassá teszi a szoftvert a természettudományok és a mérnöki tudományok területén végzett parametrikus vizsgálatokra. (Haustein és mtsai., 2017)

A YADE számos külső, kötelező és opcionális függőséggel rendelkezik. A függőségek megnehezítik a fordítási folyamatot egyes specifikus platformokon, de csökkentik a szoftver kódbázisát, növelve annak olvashatóságát és csökkentve a kódduplikációt. A YADE fordításához szükséges legfontosabb könyvtárak a következők. (Haustein és mtsai., 2017)

Eigen (könyvtár lineáris algebrához), QGlviewer (grafikus felület), VTK (postprocessing eszköz), számos Python könyvtár, CGAL (geometriai csomag, opcionális) stb. Továbbá a Boostot a kód számos helyén használjuk, főként Python kötéseket biztosítva és néhány numerikus módszerhez. (Haustein és mtsai., 2017)

Az inline dokumentáció igény szerint generálódik a fordítási folyamat során. A Sphinxet használjuk ezekre a célokra és a dokumentáció PDF, HTML és EPUB formátumban történő generálására. (Haustein és mtsai., 2017)

A YADE folyamatos integrációt használ a 'yadedaily' Deb-csomagok építésével számos Ubuntu és Debian verzióhoz. Egy, az Univ. Grenoble Alpes által üzemeltetett buildbotot használunk a kód újrafordítására minden egyes commit után és éjszakánként a fordítási hibák észlelésére. Ez a buildbot minden egyes fordítás után egység- és funkcionális teszteket is futtat. (Haustein és mtsai., 2017)

A YADE-nek aktív közössége van, amely fejleszti a kódot, és segít a felhasználóknak a kérdéseik megválaszolásában. A Launchpad infrastruktúrát kommunikációs platformként és hibakövetőként is használják. (Haustein és mtsai., 2017)

#### 4.9.1. A PROGRAM HASZNÁLATA

A program használatának menetét egy kód bemutatásával szeretném ismertetni. A használni kívánt modulok meghívása az első lépés, mint például a plot modul, amit grafikon készítésére használhatunk, vagy a pack, ami a részecskék létrehozásához szükséges. A pack modullal lehet több gömbből álló szemcsét (clump) generálni, amit a kukoricaszem közelítésére használok.

Következő lépésben a használt anyagokhoz paramétereit adom meg, amik a szimuláció során szükségesek lesznek. Az itt elkészített anyag profilokat tudjuk hozzárendelni a létrehozott geometriákhoz.

A testeket és részecskéket a geometriájuk megadásával definiáljuk és hozzuk létre. Másik megoldás lehet, hogy a használni kívánt testeket egy arra alkalmas CAD szoftverben elkészítjük és importáljuk az ymport modul segítségével. A megadott testeket meg kell hívni a szimulációban, hogy számoljon velük a program. A generált testeket a YADE háromszögek segítségével közelíti, aminek felbontása megválasztható.

Ugyanitt hozzuk létre a több gömbből álló testeket, clump-okat a nem szabályos részecskék vagy testek közelítésére használhatjuk vagy olyan esetekre, ahol egy anyag törését szeretnék vizsgálni. A következő lépés a motorok (engines) megadása. Ezek olyan előre megírt fizikai motorok, függvények, amiket a program megadott időközönként (akár minden számítási ciklusban) futtat. Beállíthatók a használni kívánt kontakt modellek, a gravitációs tér, a testek mozgatásához/forgatásához használt motorok, a szimuláció befoglaló méretei, a vizsgálni kívánt adatok kinyeréséhez használt motorok, valamint saját függvényeket is meghatározgatunk.

### 5. KALIBRÁCIÓ

A kölcsönhatásokat szabályozó mikromechanikai paraméterek megtalálása nem triviális feladat. A DEM kalibrálására nincs robusztus módszer, az eredmények nagymértékben függnek a mikromechanikai paraméterektől. Emiatt a paraméterek meghatározásának szokásos módja a részecskék viselkedésének szimulációján alapul. A paramétereket addig kell módosítani, amíg az alkalmazott mikromechanikai paraméterek az aktuális modellben ugyanazt a makromechanikai viselkedést eredményezik, mint a valóságban. Ehhez különböző technikákat alkalmazhatók, például nyíróvizsgálat, de a legegyszerűbb megoldás az a rézsűszög vizsgálatán alapuló kalibráció. Ennek a vizsgálatnak a legfontosabb tulajdonsága, hogy nem szükséges hozzá semmilyen speciális készülék, így könnyen és gyorsan elvégezhető. Megadott átmérőjű hengert kell feltölteni a vizsgálni kívánt anyaggal, majd adott sebességgel felemelni azt. A szemcsés anyag kiáramlik és kúpszerűen szétterül, az így kialakuló szöget mérve kapjuk meg a rézsűszöget.



13. ábra A kukoricaszemekből kialakul halom. (Bablena és mtsai., 2021)

#### 5.1. A használt anyagok és módszer

Egy tipikus, henger emelésével végzett kúpszög kalibrációs vizsgálat eredményei a szemcseméret és a vizsgálóberendezés méretei tekintetében változatlanok, azonban a

vizsgálóberendezés méretei kohéziós vagy elaszto-plastikus anyagok esetében fontossá válnak. Feltételezzük, hogy ebben az esetben a megfelelő kalibráció elvégzéséhez elegendő és megbízható ez a vizsgálat. Elvégeztük a mérést és a kialakult halmaz alapján mért kúpszögek a következők:

1 táblázat A kísérlet során mért kúnszögek

1. tablazat 11 kilberret bortan mert k	apszogen	
α <sub>1</sub> = 19.2 [°]	α <sub>2</sub> = 18.8 [°]	α <sub>3</sub> = 17.4 [°]

Az elérni kívánt szög pedig ezeknek az átlaga:

$$\alpha_{\acute{a}tl} = \frac{\alpha_1 + \alpha_2 + \alpha_3}{3} = \frac{19.2 + 18.8 + 17.4}{3} = 18.467 \, [^{\circ}]$$
(19)

Tehát a szárítóberendezés szimulációja elkészítése előtt fontos tudni a kukorica szemcsék súrlódási tényezőjét. Ennek meghatározásához előbb egy kalibrációs szimuláció programozása és futtatására van szükség. Itt meg kell adni az anyagra jellemző paramétereket:

- sűrűség •
- Young-modulus
- Poisson-tényező •
- súrlódási tényező •

A súrlódási tényező viszont nem ismert, így iterációval kell meghatározni. A kalibráláshoz több különböző anyagú testet alkalmaztam. A szimulációk során használt anyagtulajdonságokat a következő táblázat (2. táblázat) tartalmazza. A nem feltüntetett értékekből a YADE alap (default) értékeit használtam.

orációhoz felhasznált anyagok paraméterei				
	Kukorica	Acél	Beton	
Sűrűség $\left[\frac{\text{kg}}{\text{m}^3}\right]$	1180	7500	2400	
Young-modulus $\left[\frac{N}{m^2}\right]$	210·10 <sup>6</sup>	210·10 <sup>6</sup>	80·10 <sup>6</sup>	
Poisson-tényező [-]	0.31	0.3	0.2	
Súrlódási tényező [rad]	-	0.5	0.5	

2. táblázat Kali
A Poisson-tényezőket, az acél sűrűségét, a beton sűrűségét az irodalomból vettem. Az acél és a beton nyírási modulus értékei a nagyságrendekkel kisebbek, mivel ebben az esetben a vizsgált anyag csak a részecske, így ezeknek az értékeknek minimális hatása van a vizsgált eredményekre, de a számítási idő valóban érzékeny a nyírási modulus értékekre és így nagyobb időlépteket lehet alkalmazni a gyorsabb futtatás eléréséhez. (Bablena és mtsai., 2021)



14. ábra A kukoricaszem geometriája

A kukoricaszem valódi alakját egy összetett elem segítségével közelíthető mivel az nem gömb és a diszkrét elemes modellek általában gömb alakú elemekkel dolgoznak. Erre a közelítésre alkalmasak a korábban már említett "clump" elemek, amiket a program egyként kezel, annak ellenére, hogy több gömb összeillesztéséből képezhetők. A kukoricaszemek három gömbfelületből készültek, a gömbök sugara R = 4,5 mm volt, a súlypontját pedig a program automatikusan számolja az összeillesztett gömbökből.

A hengergeometria méretei a 15. ábraán láthatók. A kalibrációban a mérésekkel megegyező átmérőjű hengert alkalmazok. A henger magassága a mérésekben h = 25 [*cm*] volt, és teljesen tele volt, így a kukoricakupac magassága is azonos volt. A részecskék leülepedése miatt a szimulációban a henger magasságának nagyobbnak kell lennie, mint a valóságban, nagyjából 3-4-szer akkora, ez h = 75 [*cm*]. Ezzel a h magassággal a szimulációs szemcse halmaz körülbelül az eredeti halom magassága volt, mint a mérések során. A padló betonból a henger pedig acélból modellezzem.



15. ábra A henger méretei

### 5.2. A kalibráció menete

A fent említett hengert a kiinduló állapotban fel kell tölteni az elkészített szemcsékkel. A szemcsék számát és egy szemcse tömegét a szoftver kiszámította, így a szemcsék össztömege adott. Mivel a YADE clumpok használata esetén a nem nulla súrlódású anyagokat nehézkesen tudja tömör halmazokba helyezni, és a kitöltendő tér csak téglatest formájában adható meg, ezért indokolt, hogy a programot futtatni kell és megvárni, hogy a gravitáció segítségével tömörödjenek az elemek.



16. ábra A kiindulási állapot

Minden egyes részecske állapotát adott időközönként, ebben az esetben 5 valós másodpercenként, vizsgálja a szkript, így amikor a halmaz eléri az egyensúlyi állapotot, ami 1 [s] szimulációs idő után történik. Az egyensúly elérése elindítja a programban szereplő következő lépést, ami a hengert elkezdi emelni  $v_{henger} = 0.15 \left[\frac{m}{s}\right]$  sebességgel, amíg nem emelkedett fel a talajtól 25 [cm] magasra, ahol megáll.



17. ábra Az ülepített elemek

A részecskék eközben a padlón szétterülnek. Ahhoz, hogy kiszűrjem a túl távol guruló elemeket a vizsgálatból, egy ún. "Domain limiter" motort is futtatok, amivel megadhatók a vizsgálni kívánt tér befogó méretei. Az alapnak 36 x 36 [*cm*]-t adtam meg. Az ezen kívülre kerülő elemeket a program törli, a továbbiakban nem számol velük. Ez egyrészt felgyorsítja a számítást, ahogy egyre kevesebb elemet kell figyelembe venni, másrészt az így kialakuló kúp maximális átmérője is adott, a vizsgálati határok által megszabott 36 [*cm*]. A kalibráció akkor ér véget, mikor a szemcsék mozgása megáll és újra nyugalmi állapotba kerülnek.



18. ábra A henger emelkedése folyamat közben

A valódi méréstől eltérően a kúpszög mérését nem grafikus úton végeztem. A programban elkészítettem az eddig említett motorok mellett egy olyan vizsgálati parancsot, ami minden valós 60 másodpercben ellenőrzi és dokumentálja a clump halmaz befogó méreteit x, y és z irányban, ebből különösképp fontos a befoglaló téglatest felső sarokpontjának z koordinátáját, amivel meghatározható a legmagasabban elhelyezkedő elem legmagasabb pontjának pozíciója. Ennek segítségével tehát rögzítésre kerül:

- a vizsgálat időpontja szimulációban lévő virtuális idő szerint,
- a halmaz magassága,
- a halmaz alapterülete,
- valamint az adott pillanatban mérhető, YADE-ban használatos, "unbalancedForce" nagysága, amivel az egyensúlyi állapothoz való konvergálás ellenőrizhető.

A dokumentált adatokat a program a kalibráció befejeztével elmenti egy szöveges dokumentumba, egy előre meghatározott mappába.



19. ábra A nyugalmi állapot a kalibráció végén

A kalibrálást, ahogy említettem iterációra használom, tehát a súrlódási tényezőt vizsgáltam. Kiindulásként 0.1-es értékkel és innen kezdtem el felfele haladni 0.5-ig, először 0.1-es léptékkel, közben figyelve, hogy a kialakult kúpszög mikor áll legközelebb a valós mért értékhez. Ezeknek az eredményei a következő táblázatban (3. táblázat) látható.

Súrlódási tényező [rad]	Idő [s]	Magasság [mm]	Sugár [mm]
0.10	17.01	49.23	187.19
0.20	8.77	79.52	185.60
0.30	10.10	94.89	181.87
0.40	17.53	100.38	179.16
0.50	18.70	116.12	176.24

3. táblázat A kalibrálási adatok 0.1-0.5 között

A 0.2-nél nagyobb értékek esetén már a kapott szög már meghaladta a keresendőt, így leszűkítettem a vizsgált tartományt és a 0.1 és 0.2 közti értékeket kezdtem el ellenőrizni, 0.02-es lépésekkel haladva. A kapott eredmények a következő táblázatban (4. táblázat) láthatók:

Súrlódási tényező [rad]	Idő [s]	Magasság [mm]	Sugár [mm]
0.12	14.93	52.51	186.83
0.13	11.99	57.13	186.79
0.14	11.29	60.36	186.71
0.15	9.87	64.73	185.63
0.16	9.99	67.84	186.34
0.18	8.80	72.83	186.40

4. táblázat A kalibrálási adatok 0.12-0.18 között

### 5.3. Az eredmények feldolgozása

A kalibráció célja az 20. ábraán látható α szög meghatározása. A következőkben összefoglalom ennek a lépéseit.



20. ábra A kúpszög meghatározása

Ahogy az előző fejezetben említettem, a kalibrációs program segítségével kinyertem a halmaz legmagasabb pontjának koordinátáját, valamint sugarát a szimuláció végén elért nyugalmi pozícióban.



21. ábra A kalibrációk nyugalmi állapotai

A magasságot egyszerűen megkapjuk, a halmaz "sugarának" pedig az x, y tengely menti kiterjedéseinek átlagát veszem, mint közelítés, és ezt a méretet menti ki a program a további felhasználáshoz. Ezeknek az adatoknak a segítéségével pedig könnyen kiszámolható az egyes esetekben kapott kúpszög:

$$\alpha = \tan^{-1} \left( \frac{h}{\frac{D}{2}} \right) \tag{20}$$

Az eredmények ellenőrzéséhez szükséges még a relatív hiba megállapítása is:

$$H = \frac{|\alpha_k - \theta_i|}{\alpha_k} \cdot 100 \tag{21}$$

A következő táblázatban (5. táblázat) összefoglaltam a kalibrációs adatokból kiszámított eredményeket. Ezek alapján belátható, hogy 0.14-es súrlódási tényezővel kaptam a valósághoz legközelebb álló értéket. Ezután megvizsgáltam a 0.13 és 0.15 értékeket (ahogy az a fenti 21. ábraán is látható), hogy azokkal a hiba növekedni kezdett újra, így a szárítóberendezés szimulációjának futtatásához is a 0.14-es értéket állítottam be.

Súrlódási tényező [rad]	Idő [s]	Magasság [mm]	Sugár [mm]	Szög [rad]	Szög [°]	Relatív hiba [%]
0.10	17.01	49.23	187.19	0.26	14.73	20.21
0.12	14.93	52.51	186.83	0.27	15.70	14.99
0.13	11.99	57.13	186.79	0.30	17.01	7.91
0.14	11.29	60.36	186.71	0.31	17.92	2.98
0.15	9.87	64.73	185.63	0.34	19.22	4.10
0.16	9.99	67.84	186.34	0.35	20.00	8.33
0.18	8.80	72.83	186.40	0.37	21.34	15.57
0.20	8.77	79.52	185.60	0.40	23.19	25.60
0.30	10.10	94.89	181.87	0.48	27.55	49.21
0.40	17.53	100.38	179.16	0.51	29.26	85.84
0.50	18.70	116.12	176.24	0.58	33.38	80.76

5. táblázat A kalibrációból nyert adatokból számolt eredmények

A kapott eredményeket 0.1-0.2 tartományban ábrázoltam is a következő (1. diagram) grafikonon. Jól látható, hogy a súrlódási tényező növekedésével egyenesen arányos a kialakult halmaz magassága. Ez olyan esetekben lehet hasznos, mikor nincs idő elvégezni a teljes kalibrációs folyamatot a vizsgált tartomány minden értékére. Ilyenkor elegendő lehet nagyobb léptékkel haladva bizonyos értékek vizsgálata és azok

kiértékelését követően egy egyenes felvétele. A kapott egyenes egyenlete alapján a további kúpok magasságainak mérésével megfelelő közelítést kaphatunk a súrlódási kúpszögeikre.



1. diagram A kialakult kúp magasságok a súrlódási tényező függvényében

A kalibráció során kapott 0.14 [*rad*] súrlódási tényező viszonylag kicsi. Ezt az okozhatja, hogy míg a kukoricaszem sima, addig a gömbökből összerakott kukoricaszemen "dudorok" vannak, amik összeakadhatnak egymással és ez a hatás olyan mintha nagyobb lenne a szemcsék között a súrlódás.

# 6. SZÁRÍTÓBERENDEZÉS SZIMULÁCIÓJA

A szimuláció célja a szárító belsejében található lamella súrlódási tényezőjének vizsgálata a szaritasi folyamatra, valamint a szárítón belüli anyagmozgási folyamatokra.



22. ábra A szimuláció során használt keresztmetszet

A szárító berendezés CAD modelljét készítettem el először Solid Edge segítségével az 1 [mm] vastagságú lemez anyagból, hiszen az érintkező felületek szempontjából nem lényeges az anyag vastagság és minél kevesebb háromszögből kell közelíteni a geometriát annál jobb a szimuláció szempontjából. A modellt elegendő volt egyszer elkészíteni, mivel a vizsgált súrlódási tényezőt már a YADE-n belül módosítom az anyag paramétereinél. A szárító felső részét a valódi magasságának többszörösére méreteztem, az előzőekben a kalibrációnál említett feltöltési folyamat miatt és gravitációs ülepítés miatt. Ezután elkészítettem a kitároló egység ajtajait, külön testekként, hogy a YADE-ben a geometria többi részétől függetlenül mozgathatók legyenek. Továbbá mivel csak a keresztmetszet vizsgálatára összpontosít a feladatom, az arra merőleges, azt lezáró, falakat is külön testként kellett kezelnem, hogy a későbbiekben egy nulla súrlódású anyagként tudjam őket használni a program futtatása közben. A szimulációban használt modell a következő ábrán (23. ábra) látható.



23. ábra A szárítóberendezés modelljének méretei

Következő lépésként a modell-t exportáltam STL (STereoLithography) formátumként, ami lényegében a testet háromszög alakú felületelemekre darabolja és azokat listaként tartalmazza. Ez a lépés elengedhetetlen mivel a YADE igen kevés fájl kiterjesztést támogat a külső geometriák importálásához.



24. ábra A szimuláció során használt lamella méretei

A következőkben bemutatom a szimuláció programját, a Python-ban készült kódját, a főbb pontokat kiemelve. A teljes szkript kommentálva megtalálható a mellékletben.

Ahogy azt az előzőekben is tettem és említettem, itt is az anyagok definiálásával kezdem a program elkészítését. A kalibráció eredményeit felhasználva a kukorica szemcséket helyettesítő clump-oknak a 0.14 [*rad*]-os súrlódási feszültséget adom meg. A vizsgált keresztmetszetre merőleges határoló falaknak, egy módosított acél anyagot használok, aminek a súrlódási tényezője nulla, hogy ezeknek a határoknak minimalizáljam a hatását a szimuláció során. A szárító keresztmetszetét ábrázoló belső

geometriának az előzőekben is használt acél tulajdonságokat alkalmazom újra, a súrlódási tényező kivételével, amit a 0.1-0.2 tartomány között vizsgálok. A használt anyagok paraméterei a következő táblázatban (6. táblázat) találhatók. Ahogy a kalibrációnál ittt is a nem feltüntetett értékekből a YADE alap (default) értékeit használtam.

	Kukorica	Acél	Határ
Sűrűség $\left[\frac{kg}{m^3}\right]$	1180	7500	7500
Young-modulus $\left[\frac{N}{m^2}\right]$	$210.10^{6}$	210·10 <sup>6</sup>	210·10 <sup>6</sup>
Poisson-tényező [-]	0.31	0.3	0.3
Súrlódási tényező [rad]	0.14	vált.	0

6. táblázat A szimulációban használt anyagok paraméterei

A testeket hozzáadom a szimulációhoz, a hozzájuk rendelt anyagokkal és ezek után indítható a szimuláció. A kiindulási allapot a következő ábrán (25. ábra) látható.



25. ábra A kiindulási állapot 47

A feltöltést egy lépésben végzem, a teljes teret kitöltve elemekkel, amiknek egy kis időre van szükségük, hogy kellően leülepedjenek. A fő szempont, hogy tömörödött állapotban a halmaz szintje a lamella teteje felett helyezkedjen el. A nyugalmi állapot elérése, készít egy festett sávot a halmazban a további vizsgálatokhoz, majd ez követően indítja a kitárolási ciklust.



26. ábra A színezett sáv az ülepedés után

A modellszárítóban, aminek én az egyszerűsített verzióját szimulálom, az anyag kitárolása egy forgó henger segítségével történik.

A henger állandó szögsebességgel 180°-ot fordul a tengelye körül, mely során kiüríti a benne lévő magmennyiséget, majd rövid szünet után további 180°-os fordulat után visszatér a kiindulási helyzetébe, újból megtelik, majd pedig ismétlődik a folyamat. A forgó alkatrészek egyenletes kopása miatt a henger nem egy irányban forog, hanem váltott irányban, egyszer óramutató járása szerint, egyszer ellentétesen. A numerikus modellben ezen geometria és a hozzá tartozó kinematikai jellemzők megadása nehézkes lett volna, így a kör keresztmetszet helyett egy téglalap keresztmetszetű kitároló egységet használtunk, ahol az ürítő nyílás a kitároló hengeréhez hasonlóan 50 mm szélességű, a téglalap keresztmetszete pedig megegyezik a henger kör alapjának területével, azaz

$$A = \frac{d^2 \cdot \pi}{4} = \frac{91,6^2 \cdot \pi}{4} = 6589,93 \ mm^2,$$
(21)

ebből a téglalap b magassága (a vízszintes méret a = 50 mm):

.

$$A = a \cdot b$$
$$b = \frac{A}{a} = 131,8 mm. \tag{22}$$

A numerikus modellben az ürítés a kitároló egységet fedő lap elmozdulásával kezdődik. A zárólap konstans, v = 0.05 m/s sebességgel halad vízszintes irányban 1 s időtartamig, ezalatt a kitároló egység teljesen kinyílik, a szárítóból pedig beáramlanak a szemcsék.



27. ábra A kitároló egységbe áramló szemcsék nyitott felső zárólap mellett

0,5 s várakozás után a záróegység a megadott v sebességgel, ellentétes irányban mozogva 1 s alatt bezáródik. Mivel a kitároló egységben található szemcséknek a magmozgás szempontjából már nincs jelentősége, ezért a felső lap záródása után a kitároló egységben lévő szemcséket egyszerűen törlöm a szimulációból és a program nem számol velük a továbbiakban.



28. ábra A kitároló egységben lévő szemcsék törlésének a pillanata

Innentől pedig a ciklus kezdődik elölről. A felső zárólap a modellszárítóban található hengerhez hasonlóan az egyik nyitási ciklusban jobbra, a következő ciklusban pedig balra kezdi a nyitást. Egy nyitási ciklus 3 s alatt megy végbe, de mivel váltott irányban történik a nyitás, ezért egy adott irányú nyitási ciklus ismétlési ideje 6 s.

A kód írása szempontjából egy további egyszerűsítést is bevezettem. A várakozás helyett a felső lap egyszerűen tovább mozog a várakozási idő alatt is, így a várakozási idő fele alatt folytatja az útját, a másik fele alatt pedig már elindul az ellenkező irányba. Ezt a folyamatos mozgást egy egyszerűbb feltétel rendszert igényel, mint a mozgás megállítása és újra indítása, és számítási teljesítmény szempontjából elhanyagolható. A szimuláció során 4 nyitás-zárás ciklus megy végig, ezután a program megáll. Itt már egyértelműen látható, hogy a festett sávok milyen alakot vesznek fel. A YADE egy úgynevezett vtu kiterjesztésű fájlformátumba menti el az elemek adatait a megadott időközönként. Azért, hogy a szimulációról egyszerűen tudjak a későbbiekben a folyamatot ábrázoló videókat készíteni, a virtuális (szimuláción belüli) időt használtam erre. Minden virtuális másodperc alatt 30-szor mentettem ki az elemek paramétereit, amivel lényegében egy 30 képkocka per másodperces felvételt tudok generálni. A posztprocesszálást a ParaView nevű szoftverrel végeztem.

A kiértékeléshez minden egyes szimuláció esetén úgy vettem figyelembe a festett csík pozícióját, hogy az adott sáv egy lamellán végig menjen, de a festett szemcsék még ne hulljanak le az adott lamelláról.



29. ábra A festett sáv változása a négy nyitási ciklus elején (súrlódási tényező 0.4)

A lamella súrlódási tényezőjének vizsgálatára egy egyenetlenséget jellemző elmozdulási arányszámot használok:

$$\xi = \frac{y_{max} - y_{min}}{y_{max}} \tag{23}$$

Ahol:

- y<sub>max</sub> a legmagasabb pozícióban elhelyezkedő szemcse súlypontjának y koordinátája,
- y<sub>min</sub> a legalacsonyabb pozícióban elhelyezkedő szemcse súlypontjának y koordinátája.

#### 6.1. A súrlódási tényező hatása az együtt haladásra

Ezek az y koordináták a szoftverből egyszerűen lekérdezhetőek. Az y koordináták segítségével így minden vizsgált súrlódási tényező esetén meghatároztam a ξ hányadost a kezdeti állapot és harmadik nyitás közötti időtartamban, az eredmények az alábbi táblázatban találhatók.

7. táblázat A színezett sáv elmozdulásából számított ξ értékek a harmadik nyitásnál

Súrlódási tényező [rad]	y <sub>max</sub> [m]	y <sub>min</sub> [m]	ξ[-]
0	0,578417	0,517906	0,10461
0,2	0,633345	0,581871	0,08127
0,4	0,685765	0,591209	0,13788
0,6	0,744631	0,596211	0,19932
0,8	0,74558	0,592105	0,20585
1	0,747177	0,591264	0,20867

Az értékeket grafikusan ábrázolva, megfigyelhető, hogy szinte egyenes arányosság áll fent a súrlódási tényező nagysága és az elmozdulási arány között. Ez látható a következő diagramon (2. diagram):



2. diagram A lamella súrlódási tényezőjének változása a  $\xi$  függvényében

Belátható tehát, hogy a lamella súrlódási tényezőjének növelése, nagy mértékben lassítja a szemcsék fal menti mozgását és rontja az együtt haladást. Ez kifejezetten rontja a szárítás hatékonyságát. A következőkben vizsgáljuk meg grafikus módon a ParaView segítségével a szárítóban létrejövő sebességeloszlásokat.

#### 6.2. A súrlódási tényező hatása a sebességeloszlásra

A teljes modulban kialakuló mozgásviszonyokat vizsgálva látható, hogy a halmaz lefelé haladása közben helyenként feltorlódik, lelassul és sok szemcse teljesen mozdulatlan, a lamella felületén.



30. ábra A lamella menti sebességeloszlás (súrlódási tényező 0.6)

A fenti képet (30. ábra) tekintve a szemcse sebességeloszlását vizsgálva még szembetűnőbbé válik a sebességeloszlás egyenetlensége. Jól látható a falhatás, a súrlódó fal menti anyagréteg sebessége jelentősen lecsökken a faltól távolabbi rétegekhez képest. A lamella felületén is kialakul egy réteg, ami nagyon lassan vagy egyáltalán nem halad a szárítóban lefelé. A lamella alatt található részek, főleg a törés közeli részek is ehhez hasonlók, de még erőteljesebbek, boltozatok alakulnak ki. A faltól távolodva a mozdulatlan rétegek mennyisége csökken. A középvonal mentén szinte egy csatorna alakul ki, ahol az anyag szabadabban tud áramlani. Ezek mind olyan jelenségek, amik miatt a szemcsék nem egyenlő időt töltenek el a szárítóban, ami egyenetlen szárításhoz vezet.



31. ábra További sebességeloszlások különböző súrlódási tényező mellett (balról jobbra: 0, 0.4, 0.8, 1)

A 31. ábra is jól szemlélteti a súrlódás és a sebesség eloszlás közötti kapcsolatot és a fal hatás felerősödését.

#### 7. ÖSSZEFOGLALÁS

A diplomamunkám célja egy szárítóberendezés belsejében lezajló anyagmozási folyamat vizsgálatára alkalmas diszkrét elemes modell készítése és az azzal kapott eredmények vizsgálata volt.

Az irodalomkutatásban feltártam az egyenletes szárítás fontosságát és annak eléréséhez tett műszaki fejlesztéseket. Ismertettem a mezőgazdaságban fellelhető szárítóberendezések kialakulását és technikai fejlődését különböző példákon keresztül, kitérve azok előnyeire és a hozzájárulásukra a szárítási javítás szemszögéből. Bemutattam a diszkrét elemes modellezés kialakulását, számítási alapjait és felhasználási módjait, területeit. Áttekintettem a YADE DEM szoftver tulajdonságait és alap szintű alkalmazását.

Elkészítettem egy kalibrációs, amivel megfelelően közelíteni tudtam az alapjául szolgáló valóságban is végrehajtott mérést és aminek segítségével kellően pontos értéket tudtam meghatározni a kukoricaszem súrlódási tényezőjének.

A kalibráció segítségével mért súrlódási tényezőt felhasználva tudtam elkészíteni a szárítóberendezés szimulációját, ami a fő feladatom volt a diplomamunkában. A vizsgálat során nyert adatokat feldolgozva sikeresen bemutattam a szárítási folyamatra károsan ható jelenségeket és azok összefüggését a lamella súrlódási tényezőjével.

A kalibráció során kapott 0.14 [*rad*] súrlódási tényező viszonylag kicsi. Ezt az okozhatja, hogy míg a kukoricaszem sima, addig a gömbökből összerakott kukoricaszemen "dudorok" vannak, amik összeakadhatnak egymással és ez a hatás olyan mintha nagyobb lenne a szemcsék között a súrlódás.

Az elkészült kalibrációt és szimulációt is lehetett volna tovább finomítani és van lehetőség a tovább fejlesztésre és pontosításra. Ennek egyik módja lehet kukoricaszem több gömbből álló közelítésével a valósághoz közelebb álló részecske készítése, viszont ez nagymértékben megnövelné a számítási igényét, a már így is 10 órán keresztül futó programnak. Másik lehetőség a fejlesztésre a lamellára merőleges határoló falak kicserélése periodikus peremekre a nulla súrlódási tényezővel rendelkező anyagú falak helyett. Továbbá vizsgálni lehetne a különböző kontakt modelleket és azok paramétereit, hogy milyen és mekkora hatással vannak a szimulációs eredményekre.

## 8. IRODALOMJEGYZÉK

- Bagi, K. (2007). *A diszkrét elemek módszere*. Budapest: BME Tartószerkezetek Mechanikája.
- Csizmadia, B. (2009). Sajátos anyagmodellek alkalmazása a mezőgépészetben. Gödöllő: FVM Mezőgazdasági Gépesítési Intézet.
- Fleissner, F., Gaugele, T., & Eberhard, P. (2007). Applications of the discrete element method in mechanical engineering. *Multibody System Dynamics*.
- Keppler, I. (2006). Szemcsés anyagok természetes boltozódása. Gödöllő: Szent István Egyetem.
- Keppler, I., Kocsis, L., Oldal, I., Farkas, I., & Csatár, A. (2012). Grain velocity distribution in a mixed flow dryer. *Advanced Powder Technology*, 824-832.
- Mujumdar, A. S., & Beke, J. (2002). *Gyakorlati szárítás*. Budapest: Szaktudás Kiadó Ház.

#### 9. SUMMARY

The aim of my thesis was to build a discrete element model for the investigation of the material mixing process inside a dryer and to study the results obtained.

In the literature search, I explored the importance of uniform drying and the technical improvements made to achieve it. I have described the development and technical evolution of drying equipment in agriculture through various examples, highlighting their advantages and their contribution to drying improvement. I presented the development, computational basis, applications and fields of discrete element modelling. I reviewed the features and basic level application of the YADE DEM software.

I have prepared a calibration, with which I was able to adequately approximate the underlying real-world measurement and with which I was able to determine a sufficiently accurate value of the coefficient of friction of the corn grain.

Using the coefficient of friction measured by the calibration, I was able to prepare a simulation of the drying plant, which was my main task in the thesis. By processing the data obtained during the test, I successfully demonstrated the phenomena that have a detrimental effect on the drying process and their correlation with the coefficient of friction of the lamella.

The coefficient of friction of 0.14 [rad] obtained in the calibration is relatively small. This may be due to the fact that while the corn kernel is smooth, the corn kernel assembled from the kernels has "bumps" that can get stuck together and this effect is like a higher friction between the kernels.

The calibration and simulation could be further refined and there is room for further improvement and refinement. One way to do this could be to make a particle closer to reality by approximating a corn grain with several kernels, but this would greatly increase the computational demand on a program that already runs for 10 hours. Another possible improvement is to replace the boundary walls perpendicular to the geometry with periodic boundaries instead of walls with a zero friction coefficient material. Furthermore, the different contact models and their parameters could be investigated to see how and how much they affect the simulation results.

Keywords: discrete element method, DEM, mixed flow dryer, simulation

## 10. MELLÉKLETEK

Mellékletként csatoltam a kalibráció és a szimuláció szkriptjeit, a szükséges nyilatkozatokat és további ábrákat a szimuláció egyes eseteiről.

### 10.1. A kalibráció szimulációja

# ------ Import Modules ------

from yade import pack, plot, utils

#from yade import qt
#qt.View()

utils.readParamsFromTable(frictAng = 0.1)

from yade.params import table

#----- Materials -----

# grain particle

densG = 1180 EG = 80e6 nuG = 0.31 frictAngG = table.frictAng

idGrain = O.materials.append(FrictMat(density=densG, young=EG, poisson=nuG, frictionAngle=frictAngG, label='grain'))

# steel cylinder

densS = 7500 ES = 210e6 nuS = 0.3 frictAngS = 0.5

idSteel=O.materials.append(FrictMat(density=densS, young=ES, poisson=nuS, frictionAngle=frictAngS, label="steel"))

# concrete floor

denC = 2400 EC = 210e6 nuC = 0.2 frictAngC = 0.5

idConcrete=O.materials.append(FrictMat(density=denC, young=EC, poisson=nuC, frictionAngle=frictAngC, label="concrete"))

# ------ Geometry ------

# steel cylinder

cylinder = O.bodies.append(geom.facetCylinder((0, 0, 0.375), 0.10, 0.75, segmentsNumber=36, wallMask=5, material=idSteel))

# concrete floor

# create floor with plane

floor = O.bodies.append(wall((0, 0, 0), axis = 2, sense = 0))

# grain

```
cloud = pack.SpherePack()
grain = pack.SpherePack([ ((0, 0, 0), radius), ((0.005, 0.0015, 0), radius), ((0.005, -
0.0015, 0), radius)])
```

cloud.makeClumpCloud((-0.065, -0.065, 0), (0.065, 0.065, 0.7), [grain])
cloud.toSimulation(material=idGrain)

cyl = O.bodies[0]

# ------ Engines ------

# domain limiter diameter
domainDia = 0.36
domainRad = domainDia/2

# create folder for outputs
path = 'CALIBRATION'
folder = 'frictAngG-'+str(frictAngG)
name = 'frictAngG-'+str(frictAngG)+'-'+O.tags['d.id']
folderName = os.path.join(path, folder, name)
os.makedirs(folderName)

O.engines = [

ForceResetter(),

InsertionSortCollider([Bo1 Sphere Aabb(), Bo1 Facet Aabb(),

Bo1\_Wall\_Aabb()]),

InteractionLoop(

[Ig2\_Sphere\_Sphere\_ScGeom(), Ig2\_Facet\_Sphere\_ScGeom(), Ig2\_Wall\_Sphere\_ScGeom()],

[Ip2\_FrictMat\_FrictMat\_FrictPhys()],

[Law2\_ScGeom\_FrictPhys\_CundallStrack()]

),

NewtonIntegrator(gravity=(0, 0, -9.81), damping=0.4, label='newton'),

TranslationEngine(ids=cylinder, velocity=0.15, translationAxis=(0, 0, 1), dead=True, label='start'),

TranslationEngine(ids=cylinder, velocity=0, translationAxis=(0, 0, 1), dead=True, label='stop'),

DomainLimiter(dead=False, lo=(-domainRad, -domainRad, -0.01), hi=(domainRad, domainRad, 1), mask=1, iterPeriod=100, label='domain'),

PyRunner(dead=True, command='stop()', iterPeriod=100, label='stop'),

# VTKRecorder(fileName='3d-vtk-', recorders=['all'], iterPeriod=10),

# save data from Yade's own 3d view

# qt.SnapshotEngine(fileBase='3d-', iterPeriod=100, label='snapshot'),

VTKRecorder(realPeriod = 60, fileName = folderName + '/frictAng-'+str(frictAngG)+'-'+O.tags['d.id']+'-', recorders = ['all'], label = 'vtkrecorder'),

# call the checkUnbalanced function (defined below) every 5 seconds, can only run once

PyRunner(command='startCyl()', realPeriod=1, label='check'),

# check when cylinder movement can be stopped every 5 seconds, can only run once

PyRunner(command='stop()', realPeriod=1),

# call the addPlotData function every 100 steps for graphs
#PyRunner(command='addPlotData()', iterPeriod=100),

# call the addPlotData function every 60 seconds for data mining
PyRunner(command='dataMiner()', realPeriod=(60)),

# bounding box size print every 60 seconds to check height of pile
PyRunner(command='printdim()', realPeriod=(60))

]

# ------ Simulation parameters ------

O.dt = 0.5 \* PWaveTimeStep()

O.trackEnergy = True O.timingEnabled = True

def startCyl():

if O.time > 1:

if unbalancedForce() < 0.05:

start.dead=False

print('Starting cylinder')
check.command = 'stopCyl()'

def stopCyl():

```
if cyl.state.pos[2] > 1:
    start.dead=True
    stop.dead=False
    print('Stopping cylinder')
    check.command = 'checkUnbalanced()'
```

```
def checkUnbalanced():
```

```
if unbalancedForce() < 0.05:
```

```
plot.saveDataTxt(folderName+'/frictAng-'+str(frictAngG)+'-
```

'+O.tags['d.id']+'.txt',

vars=('time', 'height', 'radius', 'unbalanced'))

O.pause()

def printdim():

(xdim, ydim, zdim)=aabbDim()

print("Time:", O.time, "Height:", zdim, "Radius:", ((xdim+ydim)/2)/2, "Balance:", unbalancedForce()) def dataMiner():

(xdim, ydim, zdim)=aabbDim()

plot.addData(time=O.time, unbalanced=unbalancedForce(), height=zdim, radius=((xdim+ydim)/2)/2, \*\*O.energy)

def addPlotData():

plot.addData(i=O.iter, unbalancedForce=unbalancedForce(), \*\*O.energy)

plot.plots = {'time': ('unbalanced',)}

#plot.plot()

O.saveTmp()

# run simulation, can be run in batch
O.run()
waitIfBatch()

10.2. A szárítóberendezés szimulációja
# ------ Import Modules -----from yade import pack, plot, utils, ymport

#from yade import qt

#qt.View()

utils.readParamsFromTable(frictAng = 0)

from yade.params import table

# ------ Materials ------

# grain particle

densG = 1180 EG = 80e6 nuG = 0.31 frictAngG = 0.14

idGrain = O.materials.append(FrictMat(density=densG, young=EG, poisson=nuG, frictionAngle=frictAngG, label='grain'))

# steel dryer

densS = 7500 ES = 210e6 nuS = 0.3 frictAngS = table.frictAng idSteel=O.materials.append(FrictMat(density=densS, young=ES, poisson=nuS, frictionAngle=frictAngS, label="steel"))

# bounding walls

idBounds = O.materials.append(FrictMat(density=0, young=ES, poisson=nuS, frictionAngle=0, label='bounds'))

# ------ Geometry ------

# top door

topdoor = O.bodies.append( ymport.stl( "door\_top.stl", color=(239,174,45), material = idSteel, wire = True, fixed=True, scale=0.001))

# bottom door

botdoor = O.bodies.append( ymport.stl( "door\_bot.stl", color=(239,174,45), material = idSteel, wire = True, fixed=True, scale=0.001))

# dryer

dryer = O.bodies.append( ymport.stl( "dryer.stl", color=(239,174,45), material = idSteel, wire = True, fixed=True, scale=0.001))

# boundary

bound = O.bodies.append( ymport.stl( "bound.stl", color=(239,174,45), material = idBounds, wire = True, fixed=True, scale=0.001))

# grain radius = 0.0045

```
cloud = pack.SpherePack()
grain = pack.SpherePack([ ((0, 0, 0), radius), ((-0.0015, 0, 0.005), radius), ((0.0015, 0, 0.005), radius)])
```

```
# add every clump
cloud.makeClumpCloud((-0.09, -0.03, 1), (0.09, 0.03, 4), [grain])
cloud.toSimulation(material = idGrain, color=(75/255, 145/255, 255/255))
```

```
top = O.bodies[0]
bot = O.bodies[12]
```

# ------ Engines ------

# create folder for outputs
path = 'SIMULATION'
program = '0425'
folder = 'frictAng-'+str(frictAngS)
name = 'frictAng-'+str(frictAngS)+'-'+O.tags['d.id']
folderName = os.path.join(path, program, folder, name)
os.makedirs(folderName)

```
O.engines = [
```

ForceResetter(),

InsertionSortCollider([Bo1\_Sphere\_Aabb(), Bo1\_Facet\_Aabb(),

Bo1\_Wall\_Aabb()]),

InteractionLoop(

[Ig2\_Sphere\_Sphere\_ScGeom(), Ig2\_Facet\_Sphere\_ScGeom(), Ig2\_Wall\_Sphere\_ScGeom()],

[Ip2\_FrictMat\_FrictPhys(frictAngle=MatchMaker(matches=((idGrain, idGrain, frictAngG), (idSteel, idGrain, frictAngS))))],

[Law2\_ScGeom\_FrictPhys\_CundallStrack()]

),

NewtonIntegrator(gravity=(0, 0, -9.81), damping=0.4, label='newton'),

TranslationEngine(ids=topdoor, velocity=0.05, translationAxis=(1, 0, 0), dead=True, label='topPositive'),

TranslationEngine(ids=topdoor, velocity=-0.05, translationAxis=(1, 0, 0), dead=True, label='topNegative'),

```
TranslationEngine(ids=topdoor, velocity=0, translationAxis=(1, 0, 0), dead=True, label='topStop'),
```

DomainLimiter(dead=True, lo=(-0.12, -0.8, 0.132), hi=(0.12, 0.8, 3.5), mask=1, iterPeriod=1, label='discharge'),

VTKRecorder(virtPeriod = 1/30, fileName = folderName + '/frictAng-'+str(frictAngS)+'-'+O.tags['d.id']+'-', recorders = ['all'], label = 'vtkrecorder'),

# call the checkUnbalanced function (defined below) every 5 seconds, can only run once

PyRunner(command='checkBalance()', iterPeriod=1, label='check'),

]
# ------ Simulation parameters ------

O.dt = 0.8 \* PWaveTimeStep()

O.trackEnergy = True O.timingEnabled = True

cycle = 0

```
def checkBalance():
```

if O.time > 2:

# after balance start opening the top door - positive direction
if unbalancedForce() < 0.06:</pre>

check.command = 'colorBand()'

def colorBand():

for b in O.bodies:

if isinstance(b.shape, Sphere): if 0.775 < b.state.pos[2] < 0.8: b.shape.color = (255/255, 200/255, 0/255) check.command = 'topMtoP()'

def topMtoP():

# move top door from the middle to the positive side topStop.dead=True topPositive.dead=False discharge.dead=True print(O.time, 'Moving Top from Middle to Positive side') check.command = 'topPtoM()' def topPtoM():

# if the top door reaches the positive side ( full length + waiting time )
if top.state.pos[0] > 0.05:

# move top door from the positive side to the middle topPositive.dead=True topNegative.dead=False print(O.time, 'Moving Top from Positive side to Middle') check.command = 'emptyUnit1()'

## def emptyUnit1():

# if the top door reaches the middle

if top.state.pos[0] < -0.025:

# stop the movement

topPositive.dead=True

topNegative.dead=True

topStop.dead=False

# empty discharge unit

discharge.dead=False

check.command = 'topMtoN()'

## def topMtoN():

# move top door from the middle to the negative side topStop.dead=True topNegative.dead=False discharge.dead=True print(O.time, 'Moving Top from Middle to Negative side') check.command = 'topNtoM()'

### def topNtoM():

# if the top door reaches the negative side ( full length + waiting time )
if top.state.pos[0] < -0.1:</pre>

# move top door from the negative side to the middle topNegative.dead=True topPositive.dead=False print(O.time, 'Moving Top from Negative side to Middle') check.command = 'emptyUnit2()'

## def emptyUnit2():

# if the top door reaches the middle

if top.state.pos[0] > -0.025:

# stop the movement

topPositive.dead=True

topNegative.dead=True

topStop.dead=False

# empty discharge unit

discharge.dead=False

global cycle

cycle += 1

if cycle < 4:

check.command = 'topMtoP()'

else:

O.pause()

O.saveTmp()

# run simulation, can be run in batch
O.run()
waitIfBatch()

# 10.3. Nyilatkozatok

#### NYILATKOZAT

#### a diplomadolgozat nyilvános hozzáféréséről és eredetiségéről

	A hallgató neve:	Tanács Kornél
	A Hallgató Neptun kódja:	VAQA8U
*	A dolgozat címe:	Szárítóberendezésben kialakuló szemcsemozgási folyamatok modellezése
	A megjelenés éve:	2023
	A konzulens tanszék neve:	Műszaki Intézet, Gépészettani Tanszék

Kijelentem, hogy az általam benyújtott záródolgozat/szakdolgozat/diplomadolgozat/portfólió12 egyéni, eredeti jellegű, saját szellemi alkotásom. Azon részeket, melyeket más szerzők munkájából vettem át, egyértelműen megjelöltem, s az irodalomjegyzékben szerepeltettem.

Ha a fenti nyilatkozattal valótlant állítottam, tudomásul veszem, hogy a Záróvizsga-bizottság a záróvizsgából kizár és a záróvizsgát csak új dolgozat készítése után tehetek.

A leadott dolgozat, mely PDF dokumentum, szerkesztését nem, megtekintését és nyomtatását engedélyezem.

Tudomásul veszem, hogy az általam készített dolgozatra, mint szellemi alkotás felhasználására, hasznosítására a Magyar Agrár- és Élettudományi Egyetem mindenkori szellemitulajdonkezelési szabályzatában megfogalmazottak érvényesek.

Tudomásul veszem, hogy dolgozatom elektronikus változata feltöltésre kerül a Magyar Agrárés Élettudományi Egyetem könyvtári repozitori rendszerébe.

Kelt: 2023. év 05. hó 02. nap

Tandis Korull

#### Hallgató aláírása

#### **KONZULTÁCIÓS** NYILATKOZAT

A Tanács Kornél (név) (hallgató Neptun azonosítója: VAQA8U) konzulenseként nyilatkozom arról, hogy a záródolgozatot/szakdolgozatot/diplomadolgozatot/portfólió1 áttekintettem, a hallgatót az irodalmi források korrekt kezelésének követelményeiről, jogi és etikai szabályairól tájékoztattam.

A záródolgozatot/szakdolgozatot/diplomadolgozatot/portfóliót a záróvizsgán történő védésre javaslom / nem javaslom<sup>2</sup>.

A dolgozat állam- vagy szolgálati titkot tartalmaz:

igen <u>nem</u>\*3

Kelt: 2023. év május hó 3. nap

Skypler Job Belső konzulens

A megfelelő dolgozattípus meghagyása mellett a többi típus törlendő.
 A megfelelő aláhúzandó.
 A megfelelő aláhúzandó.



Szent István Campus, Gödöllő Cím: 2100 Godolló, Páter Károly utea 1. Tel.: +36-28/522-000 Honlap: <u>https://godollo.uni-mate.hu</u>

1. sz. függelék – Témaválasztási lap

# ZÁRÓDOLGOZAT / SZAKDOLGOZAT / DIPLOMADOLGOZAT\* TÉMAVÁLASZTÁSI LAP

Hallgató tölti ki	!			
Hallgató neve: T	anács Kornél	Neptu	n kódja: VAQA8U	
Szak: Gépészmé	rnöki			
Képzési hely*:	Budai Camp	us / Georgikon Campus (H	Keszthely) / Kaposvári Campus	
	Károly Róbe	rt Campus (Gyöngyös) / S	zent István Campus (Gödöllő)	
	Szarvasi Kép	zési Hely		
Képzési szint*:	FOSZK/BA/	BSc / MA/MSc/SZTK Évf	folyam: 2. évfolyam	
Tagozat*: Nappali / Levelező / Távoktatás				
Szakirány(ok) /	Specializáció(1	<)*: Műszaki feilesztő		
Hallgató e-mail címe; tanacskornel@gmail.com				
Külső konzulen: Diploma- , Szak Szárítóberendez Kelt: 2023. év 0	s neve, beosztá - vagy Záródol ésben kialakul 5. hó 02. nap	sa és munkahelye: Dr. Varga Igozat témája: ó szemcsemozgási folyamato	Attila, szimulációs mérnök ok modellezése Tanaja Komel	
Belső kor	zulens	Külső konzulens	Hallgató	
Intézetigazgató megküldendő a tanulm <i>Témaválasztáss</i> Kelt:	/ Tanszékveze ányi ügyintézőnek és al egyetértek/n év	ztő/ Szakfelelős tölti ki! – aláí : a befogadó intézet adminisztrációjának em értek egyet* hó nap	rúst követően a témaválasztási lap elektronikuss t.	
			Szakfelelős/Szakkoordinátor	
A dolgozat témá	t befogadom/n	em fogadom be*		
Kelt:	év	hó nap		
			Intézetigazgató/Tanszékvezető*	
*Kérjük a megfelelő **Amennyiben az ér dolgozik, akkor az in campus koordinátora	t aláhúzni! intett szak és a térr tézet adott campuso (ennek hiányában a	nát kiadó intézet vezetője nem a hall n illetékes tanszékének vezetője – (e szakfelelős) írja alá.	gató képzési helye szerint illetékes campuso nnek hiányában az intézetigazgató), ill. a szr	

# 10.4. További ábrák



32. ábra A festett sáv változása a négy nyitási ciklus elején (súrlódási tényező 0)



33. ábra A festett sáv változása a négy nyitási ciklus elején (súrlódási tényező 0.2)



34. ábra A festett sáv változása a négy nyitási ciklus elején (súrlódási tényező 0.4)



35. ábra A festett sáv változása a négy nyitási ciklus elején (súrlódási tényező 0.6)



36. ábra A festett sáv változása a négy nyitási ciklus elején (súrlódási tényező 0.8)



37. ábra A festett sáv változása a négy nyitási ciklus elején (súrlódási tényező 1)



38. ábra Sebességeloszlás (súrlódási szög 0)



39. ábra Sebességeloszlás (súrlódási szög 0.2)



40. ábra Sebességeloszlás (súrlódási szög 0.4)



linVelLen

41. ábra Sebességeloszlás (súrlódási szög 0.6)



42. ábra Sebességeloszlás (súrlódási szög 1)