

Gradu Amaierako Lana  
Gradu bikoitza Fisikan eta Ingeniaritza Elektronikoan

# Giza indarraren ereduaren bidezko oinezkoen simulazioak: bi noranzkoko fluxua eta larrialdi-egoerak

Egilea:  
Ibon Barrio Ugidos  
Zuzendaria:  
Josu Mirena Igartua Aldamiz





# Gaien Aurkibidea

<b>1</b>	<b>Oinezkoen jokaeraren azterketa</b>	<b>1</b>
1.1	Testuinguru historikoa . . . . .	1
1.2	Ezaugarriak eta ebidentziak . . . . .	1
1.3	Ereduak . . . . .	2
<b>2</b>	<b>Social Force Model (SFM)</b>	<b>4</b>
2.1	Sarrera . . . . .	4
2.2	Eredua . . . . .	4
2.2.1	Indar abiarazlea . . . . .	5
2.2.2	Oinezkoen arteko indarra . . . . .	5
2.2.3	Hormek eragindako indarra . . . . .	6
2.2.4	Bestelako gaiak eta aldakuntzak . . . . .	6
2.3	Anisotropia . . . . .	7
2.4	Ereduaren hedapena: larrialdi kasuak . . . . .	8
2.4.1	Testuingurua . . . . .	8
2.4.2	Larrialdietarako eredua . . . . .	8
2.5	Mugak eta arazo intrintsekoak . . . . .	9
2.6	Datu esperimentalak eta ereduaren kalibrazioa . . . . .	10
2.6.1	Konstanteen zehaztapena eta aukeraketa . . . . .	11
<b>3</b>	<b>Neurtutako magnitudeen definizioak</b>	<b>13</b>
3.1	Batez-besteko denbora . . . . .	13
3.2	Polarizazioa . . . . .	13
3.3	Biskositatea . . . . .	14
3.3.1	Lerro-biskositatea . . . . .	14
3.3.2	Espazioaren menpeko biskositatea . . . . .	14
<b>4</b>	<b>Problemaren ebazpena</b>	<b>16</b>
<b>5</b>	<b>Esperimentuak</b>	<b>19</b>
5.1	Bi noranzkoko korridorea . . . . .	19
5.1.1	K0. Esperimentua: lerroen eraketa . . . . .	19
5.1.2	K1. Esperimentua: Korridorearen zabaleraren eragina . . . . .	20
5.1.3	K2. Esperimentua: Polarizazioa eta biskositatea . . . . .	21
5.1.4	Anisotropiaren eragina . . . . .	22
5.1.5	Oinezko kopuruaren eragina . . . . .	24
5.1.6	K7. Esperimentua: Oztopoen eragina . . . . .	25
5.2	Larrialdi-egoera: gela baten ebakuazioa . . . . .	27
5.2.1	Problemaren definizioa eta planteamendua . . . . .	27
5.2.2	Emaitzak eta analisisa . . . . .	30

---

<b>Ondorioak eta etorkizuneko lana</b>	<b>35</b>
<b>Bibliografia</b>	<b>37</b>
<b>A Programak</b>	<b>39</b>
A.1 Bi noranzkoko fluxua . . . . .	39
A.1.1 SFM funtzioa . . . . .	39
A.1.2 Eboluzioa . . . . .	43
A.2 Larrialdi egoera: gela baten ebakuazioa . . . . .	46
A.2.1 Erabilitako pakete eta libreriak . . . . .	46
A.2.2 Hasierako egoeraren definizioa . . . . .	46
A.2.3 Oztopen eta oinezkoen distantzia minimoa . . . . .	48
A.2.4 SFM funtzioa . . . . .	49
A.2.5 Runge-Kutta 4 . . . . .	52
A.2.6 Eboluzioa . . . . .	52
A.2.7 Programa nagusia . . . . .	55
<b>B Runge-Kutta metodoa</b>	<b>57</b>
<b>C Esperimentuen irudiak: larrialdi-egoerak</b>	<b>59</b>



# Sarrera

Oinezkoentzako inguruneak pertsonak oinez mugitzeko zonalde publiko edo pribatuak dira. Hauen adibide dira metroetako pasabideak, parkeak, aireportuak edo merkataritza guneak, esaterako. Maiz toki hauek bete egiten dira, daukaten eskari handia dela eta. Horrek oinezkoen segurtasuna arriskuan jartzen du, bai ohiko baldintzetan baita larrialdi kasuetan ere.

Lan honen motibazioa fisikaren ezagutzak segurtasuna bezalako gai inportante baterako erabiltzea da. Hasierako ideia soziofisikarekin lotuta zegoen zerbait egitea zen. Izan ere, oso erakargarria iruditzen zitzaidan fisika estatistikoa erabiliz, banakoa, egitura sozial baten unitate gisa, honek beste pertsonekin zituen interakzioen ondorioz sortutako fenomenoak aztertzea. Arlo honetan, hamaika aplikazio oso interesante daude: iritziaren dinamika, hautesleen botoarekin lotutako ereduak lantzen dituen; hierarkien eraketa, banakoen berdintasunetik desberdintasuna nola sortzen den aztertzen duena edo hizkuntzen dinamika, esaterako, hizkuntzak nola sortu, eraldatu eta galtzen diren analizatzea helburu duena. Lan hau soziofisikaren barruan jendetzaren dinamikaren azterketaren alorrean koka daiteke. Arlo horretan, oinezkoen dinamikaz gain, ikuskizun baten ondoren jendetzak aldi berean eta frekuentzia berarekin nola txalotzen duen ere azter daiteke adibidez. Hala ere, aipatu dudan moduan, oinezkoen dinamikaren azterketak segurtasuna hobetzeko harreman estua zeukanez, ongizatearen funtsezko oinarria den esparru horretan fisikaren bidez laguntzea interesgarria iruditzen zitzaidan [2].

Lan honen helburua da oinezkoen dinamika aztertzen duen eredu bat lantzea. Besteak beste, eredu horren formulazio matematikoa, aplikagarritasun-esparruak, mugak eta datu esperimentalekiko erlazioa aztertuko dira. Gainera, programazio-lengoai baten laguntzarekin, Python alegia, eredu hau frogatzeko eta oinarritzko ebidentzia esperimentalak betetzen diren ala ez ikusteko zenbait simulazio egingo da.

Oinezkoen dinamika aztertzeaz gain, lan honek Python programazio lengoaiaren erabilpenean trebatzea, eta datu-egitura eta konputazio-egituraren ezagutzan oinarritutako programazio metodologia indartzea du helburu. Gainera zehazki Python Numpy libreria [22] ezagutzea eta horrek eskeintzen dituen erreztasunez jabetzeko xedea du.

Bestalde, argi dago gaur egun enpresa askotan talde-lanak pisu handia daukala. Ondorioz, komunikatzeko eta egindako programa bertsio diferenteak sailkatzeko eta partekatzeko erreminta egokiak izatea ezinbesteko bilakatu da. Hori burutzea eskaintzen duen aplikazio bat Jupyter Notebook-a [23] da. Zientzian datuen analisirako asko hedatzen ari den baliabidea da eta plataforma eta programa ugariarekin bateragarria da gainera. Bere garrantziaren erakusgarria da duela gutxi Nature aldizkari ospetsuak plataforma honen inguruan argitaratutako artikulua. Lan honek web-aplikazio potente hau, kodea eta testu azalpenak tartekatzen dituen, erabiltzen ikastea du ere badu jomuga.

Lan hau bi ataletan banatuta dago: lehenengo partean korridore batean binoranzkoko fluxua aztertuko da. Pasilloak daukan geometriaren, oinezko kopuruaren, oztopoen eta anisotropiaren arabera polarizazioa, zeharkatzeko bataz besteko denbora eta biskositatea aztertuko dira. Esparru horretan oinezkoek bere ingurumena ez dutela modu berean norabide guztietan hautematen fenomenoarekin dago lotuta anisotropia. Bigarren atalean erabilitako ereduak hedatuko da zenbait aldaketarekin larrialdi-kasuak aztertzeko asmoz. Gela bat larrialdi-egoeran irtenarazteko atetik gertu oztopoak ipintzeak daukan eragina ikustea da zati honen xedea. Geometria diferenteak, hots, oztopo desberdinen kokapen diferenteak aztertuko dira.

# 1. Kapituluua

## Oinezkoen jokaeraren azterketa

Lehenengo kapitulu honetan oinezkoen jokaeraren inguruko oinarrizko ideia batzuk ikusiko dira. 1.1 atalean ibiltarien dinamikaren azterketaren testuinguru historikoa laburki azalduko da. Ondoren, 1.2 atalean, oinezkoek ibiltzean jarraitzen dituzten patroiak eta ezaugarriak ikusiko dira. Bukatzeko, 1.3 atalean, ezaugarri horiek erreproduzitzeko eta adierazteko, historian zehar proposatu diren eredu berri emango da.

### 1.1 Testuinguru historikoa

Oinezkoen fluxuen azterketa ez da gauza berria, bereziki eraikinen plangintzaren eta suteen babesaren arloetan. Hala ere, oinezkoen dinamikaren azterketaren jatorria 50. hamarkadan koka dezakegu [1]. Garai horretan zuzenezko behaketaren, argazkien eta time lapse filmen bitartez egiten zen.

Lehenengo eredu fisikoa 1971. urtean proposatu zen [3]. Horretan, oinezkoen fluxuak gasen eta fluidoan antzekoak zirela kontsideratzen zen eta, ondorioz, harreman zuzena zuten Navier-en Stokes-en ekuazioekin. Horrek arazo handia zeukan, izan ere ez zituen aintzat hartzen talkak saihesteko maniobrak, zeintzuetan energia eta momentua kontserbatzen ez diren. Horrez gainera, ez zuten oinezkoen natura pikortasuna kontsideratzen: oinezko bakoitzak bolumen zehatz bat okupatzen du eta gainerakoak ezin dira horretan sartu. Horren guztiaren ondorioz, azken urteotan eredu mikroskopikoak izan dira arrakastasuenak.

### 1.2 Ezaugarriak eta ebidentziak

Pentsa daiteke oinezkoen portaerak ordenik ez duela, baina bai ordea arrazoi eta helburu bat. Hala ere, fluxuetan patroiak agertzen direla ikusi da. Oinezkoen dentsitatearen arabera egoera gaseosoak (ibiltari kopuru txikia eta libreki higitu), fluidoak (oztopoduna) eta solidoak (ibiltari kopuru oso handia eta ia higitu ezin direnean) ikusi dira.

Ibiltarien portaera hiru mailatan deskribatzen da: "estrategikoa", eginbeharrekoen zerrenda bat erabakitzea; "taktikoa", epe laburreko erabakiak dira eta maila estrategikoan erabakitutako zereginei garrantzia ematen zaie; gainera, ibilbide bat zehazten da, hemuga eta tarteko jomugak dituen; azkenik, "operatiboa", hartutako erabakiak exekutatzeko dira.

Ibiltarien jokaeraren ezaugarri nagusietako bat da gune pertsonalaren efektua. Oinezkoak deseroso sentitzen dira beste ibiltariengatik gertu higitu behar direnean. Edward Hall an-

tropologoak 1959. urtean proxemika kontzeptua azaltzen du, zeinaren arabera 4 distantzia interpersonal daude: **norberarena**, **pertsonala**, **soziala** eta **publikoa**. Horiek kulturaren eta erlijioaren arabera aldatzen dira: Asiakoek, adibidez, espazio pertsonal gutxiago behar dute ibiltzeko.

Beste ezaugarrietako bat da pertsonen auto-antolatzeak daukaten portaera [4]. Hori egoera desberdinetan ikusi da. Fluxua noranzko bakarrekoa denean eta homogeneoa, hau da, ibiltari guztiek ezaugarri berdinak dituztenean, oinezkoak ilaretan antolatzen dira. Izan ere, bata bestea aurreratzea ekiditen du homogeneotasun horrek. Noranzko bakarreko fluxu heterogeneoaren kasuan, aldiz, abiadura edo norabide-aldaketak oinezko baten egoera hobetu dezake. Dena dela, onura handia ez bada, oinezkoak nahiago du abiadura moteldu eta beste baten atzetik joan, horrela oztopoekin talka egiteko probabilitatea murrizten baita. Teoriak, beraz, abiadura bereko fluxu homogeneoen eraketak aurreratzen ditu. Bukatzeko, bi noranzkoko fluxuen kasuan, noranzko bereko lerro dinamikoak eratuz antolatzen dira oinezkoak. Kasu optimoa korridorea bi gunetan banatzen denean gertatzen da: zonaldetako batean noranzko bereko lerroak korrante fluxu bat eratzen dute eta, bestean, kontrako noranzkokoak. Modu honetan korranteen arteko marruskadura bi korranteen arteko mugan baino ez da gertatzen. Korridoreen bidegurutzetan lerro diagonalen eraketa ere ikusi da. Interesgarria da aipatzea, korrante dinamiko hoiek autoek zirkulatzen duten alde berean eratzen direla eta beraz, herrialdeen arabera aldatu egiten dira.

Laburbilduz esan daiteke: kulturalki menpekoak (ibiltzeko nahiago duten aldea, espazio pertsonalaren zabalera) diren patroi kolektiboak eratzen dituztela pertsonen (lerroak, oszilazioak). Hau da, tokian tokiko elkarrekintzen biden patroi globalak sortzen dira. Patroi hauek dentsitate arruntetan eraginkorrak (auto-antolatu) izaten dira. [16]

### 1.3 Ereduak

Mekanika estatistikoan eta dinamika molekularrean dago jendetzaren dinamikaren azterketaren filosofia. Beraz, bizidunak izango ez balira bezala tratatzen dira oinezkoak. Horren justifikazioa da oinezko bakar baten eragina jendetza osoan oso txikia dela. Hau da, zarata estokastikotzat jo daiteke dinamika determinista zehatz bati gainezarrita.

Ikuspegiaren arabera ereduak hainbat taldetan sailka ditzakegu. Esaterako, eskala kontuan hartzen badugu:

- **Makroskopikoak:** Oinezkoen kopurua oso handia denean egiten da eta abiadura edo presioa bezalako aldagaiak izaten dira aztergai.
- **Mikroskopikoak:** Gizabanakoa bera aztertzen da eta horrek ingurunearekin dituen interakzioak. Oinezko bakoitzaren berezko informazioarekin (sexua, adina, pisua, abiadura,..) lan egiteko ahalbidetzen du.

Makroskopikoaren eta mikroskopikoaren arteko tarteko eskala bat ere badago, **mesoskopikoa** deritzona. Hori pertsona-talde ezaugarrietara bideratuta dago.

Aldagaiak har ditzaketen balioen arabera ere egin daiteke eredu sailkapena:

- **Diskretuak:** Orokorrean, bai espazioan bai denboran diskretuak izaten dira. Oinezkoa modelatzen dute bere posizioa hurrengo moduan eguneratuz:

$$\mathbf{x}_{t+1} = f(\mathbf{x}_t, \beta) \quad (1.1)$$

$\mathbf{x}_t$   $t$  aldiuneko egoera bektorea da eta  $\beta$  ereduaren parametroak dira. Eredu horien abantailarik handiena da konputazio aldetik azkarrak eta eraginkorrak direla. Hala ere, ibiltarien kopurua asko handituz gero konputagailuaren memoria erraz agortu daiteke.

- **Jarraituak:** Denboran jarraitua den funtzio batez definitzen dute posizioa:

$$\frac{d\mathbf{x}}{dt} = f(\mathbf{x}(t), \beta) \quad (1.2)$$

non  $\mathbf{x}(t)$  zenbakizko integrazio metodoen bidez kalkulatzen den. Egonkortasuna eta emaitza egokiak bermatzeko  $\Delta t$  denbora-pausua txikia hartzea halabeharrezkoa da.

Ibiltarien konportamoldea aztertzeko, besteak beste, hurrengo ereduak proposatu dira:

- **Gasen zinematika:** Abiadura banaketak eta dentsitateak kontsideratzen dituzte baina numerikoki ahalegin handia eskatzen dute. Mota honetakoa izan zen Hoogendoorn and Bovy (2000) ikerlariak proposatutako ereduak.
- **Fluidoaren dinamika:** Talka uhinak (fluidoaren mekanikan deritzo soinuaren abiadurarekin baino azkarrago hedatzen den presio uhinari, presio-diferentzia handiak eta tenperaturaren igoera dakartzana) aztertzeko egokiak eta eraginkorrak dira. Hala ere, jendetzaren natura diskretoa eta pikortasuna ez dute kontuan hartzen. Izan ere, oinezkoak korronteen bitartez adierazten dituzte. Mota honetakoak izan ziren Henderson (1971) eta Hughes (2000) ikerlarien ereduak; biak makroskopikoak eta jarraituak.
- **Cellular automata (CA):** Lan honen aztergai den Social Force ereduarekin batera CA motako ereduak izan dira arrakastatsuenak. CA ereduak konputazionalki oso eraginkorrak dira baina espazioan diskretizatzeak muga handiak ipintzen ditu fluxuen azterketan. Klüpfel (2003) eta Schadschneider (2002) ikerlariak, esaterako, mota honetako ereduak garatu zituzten.

Lan honetan **Social Force Model (SFM)** ereduak aztertuko da, mikroskopikoa eta jarraitua dena. Gainera, eredu honen bidezko azterketa egingo da. Hurrengo kapituluan honen nondik norakoak azalduko dira.

## 2. Kapituluua

# Social Force Model (SFM)

Kapitulu honetan oinezkoen dinamika aztertzeke lan honetan erabiliko den ereduaren, Social Force Model (SFM)-aren, oinarri teorikoa azalduko da. Ereduaren funtsa eta mugapenak aztertuko dira. Gainera, ereduaren hedapena egingo da larrialdi kasuak ere deskribatzeko.

### 2.1 Sarrera

Eredu hau 1995. urtean proposatu zuten Helbing eta bere kideek [10]. Oinezkoetan eragiten duten giza-indarrek, *social force*, irudizko indarretan dago oinarrituta. Giza indarrak giza-elkarrekintzen ondorioz (fisikako elkarrekintza edo eremuen orde) azelerazioa edo dezelazio efektua deskribatzen dituzten bektoreak dira. Indar horiek oinezkoaren higidura zehazten dute.

Oinezkoen arteko elkarrekintzak, auto-antolamendu efektuak eta jendetzaren presio handiak era errealistan deskribatzen ditu SFM ereduak. Bere desabantaila handiena da konputazionalki esfortzu handia eskatzen duela.

### 2.2 Eredua

Oinezkoen higidura ekuazio diferentzial arrunt baten bidez deskribatzen da:

$$m_i \frac{d\vec{v}_i}{dt} = \vec{F}_{tot} + \xi \quad (2.1)$$

Adierazpen horretan  $i = 1, 2, \dots, N$  oinezkoa adierazten du;  $m_i$  da  $i$ . oinezkoaren masa. Newtonen bigarren legea dugu,  $\xi$  gai osagarri batekin. Fluktuazioak deskribatzeko "sartzen" da hori, oinezkoak ohiko jarreratik urruntzen direneko jokabidea adierazteko (ingurunearekin entretentzen direnean, giza-jokaerek irregularitasun edota imperfekzio-gradua daukate,...). Lan honetan gai hori ez da kontuan hartuko.  $\vec{F}_{tot}$  ingurunearen efektua deskribatzen du eta  $i$ . oinezkoarengan eragiten du. Hainbat atalez osatuta dago:

$$\vec{F}_{tot} = \vec{F}_i^d + \sum_w \vec{F}_i^w + \sum_{i \neq j} \vec{F}_{ij}^r + \sum_g \vec{F}_{ig}^a \quad (2.2)$$

2.2 ekuazioan  $\vec{F}_i^d$  oinezkoaren helbururantz zuzendutako higidura indarra da;  $\sum_w \vec{F}_i^w$  da  $w$  objektuak (hormak, kaleko oztupoak)  $i$  oinezkoan eragindako indarra;  $\sum_{i \neq j} \vec{F}_{ij}^r$   $j$  oinezkoak  $i$  oinezkoan eragindako indar alderatzailea eta  $\sum_g \vec{F}_{ig}^a$  pertsona edo objektu bati erakarpema da. Azken gai hori lan honetan ez da kontuan hartuko.

### 2.2.1 Indar abiarazlea

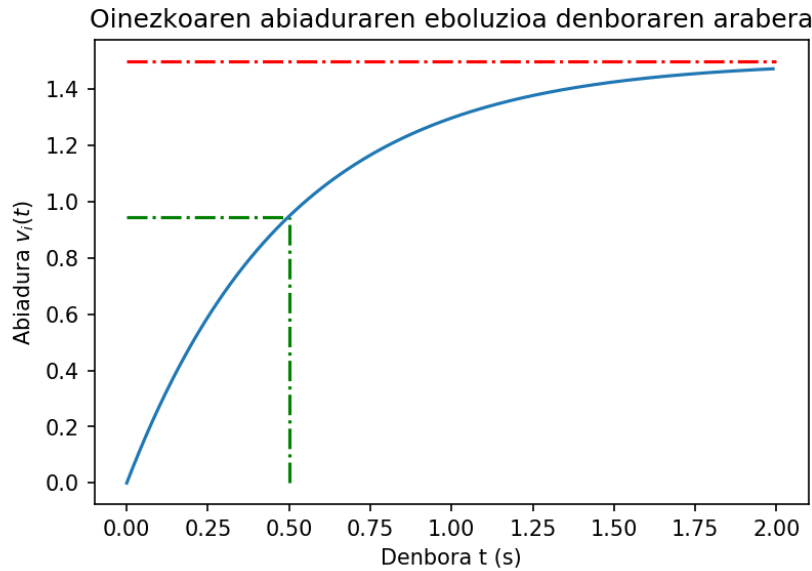
Bere tarteko helmuetara eta amaierako helmugara zuzentzen du oinezkoa indar honek. Honako formakoa:

$$\vec{F}_i^d(t) = m \cdot \frac{v_i^0(t) \vec{e}_i^0(t) - \vec{v}_i(t)}{\alpha_i} \quad (2.3)$$

2.3 adierazpenean  $v_i^0$  eta  $e_i^0$  oinezkoak desiratutako abiadura eta noranzkoa diren.  $\alpha$  erlajazio denbora da eta 2.1 irudian ikus daiteke bere esanahia. Ibiltaria  $\vec{r}_i$  posiziora joan nahi badu desiratutako noranzkoaren bektore unitarioa honako hau da:

$$\vec{e}_i^0 = \frac{\vec{r}_t - \vec{r}_i}{\|\vec{r}_t - \vec{r}_i\|} \quad (2.4)$$

Oinezkoak  $t$  aldiunean daukan abiadura da. Azken hori desiratutako abiadura baino handiagoa edo txikiagoa izan daiteke baina 2.3 ekuazioari erreparatzen badiogu,  $v_i^0$  eta  $e_i^0$  desiratutako abiadura eta noranzkoak berreskuratzen ditu  $\alpha_i$  denbora karakteristikoan, 2.1 irudian ikusten den bezala.



2.1. Irudia:  $\alpha = 0.5$  denbora karakteristikoa eta  $v^0 = 1.5m/s$ -ko desiratutako abiadura duen oinezko batentzat, bere abiadura denboraren funtzioan. Kolore berdearekin  $\alpha = 0.5$  segunduan oinezkoak desiratutako abiaduraren  $0.63 \cdot v_i^0$  lortzen duela ikusten da. Gorriz, desiratutako abiadura markatuta dago, limitean hauxe lortzen du oinezkoak.

2.1 irudian ageri den bezala, denbora pasatu ahala oinezkoak bere desiratutako abiadura lortuz doa. Gainera,  $\alpha$  denbora karakteristikoa pasatu eta gero, bere desiratutako abiaduraren  $(1 - 1/e)v_i^0 \approx 0.63 \cdot v_i^0$  dauka.

### 2.2.2 Oinezkoen arteko indarra

Ibiltarien arteko elkarrekintza deskribatzen du indar honek. Dakigun moduan, oinezkoek ibiltzeko espazio propio bat edukitzen dute eta, ondorioz, indar hori alderatzailea da. Hurrengo forma dauka:

$$\vec{F}_{ij}^r(d_{ij}) = A_i e^{-\frac{r_{ij} - d_{ij}}{B_i}} \vec{e}_{ij} \quad (2.5)$$

$A_i$  eta  $B_i$  konstanteak dira. Berreturako  $r_{ij} = r_i + r_j$  da  $i$  eta  $j$  oinezkoen erradioen batura eta  $d_{ij} = \|\vec{r}_i - \vec{r}_j\|$ ,  $i$  eta  $j$  oinezkoen masa zentroen arteko distantzia. Ikusiko den moduan, larrialdi kasuak aztertzen direnean, oinezkoen arteko indar horren adierazpenean bi gai gehiago sartu beharko dira.

Lan honetan, karga konputazionala murrizteko, indar alderatzailea hurrengo moduan definituko da:

$$\vec{F}_{ij}^r(d_{ij}) = \begin{cases} A_i e^{\frac{r_{ij}-d_{ij}}{B_i}} \vec{e}_{ij} & d_{ij} < R_p \\ 0 & \text{bestela} \end{cases} \quad (2.6)$$

$nR_p$  alderatze erradioa den. Horrek esan nahi du, oinezko bat beste ibiltari baten alderatze erradio horren kanpo baldin badago, ez diola inongo indarrik eragingo (ez du ikusiko). Aipatzekoa da azken indar hau ez jarraitua dela. Gainera, oinezkoak errealitatean gertatzen ez den bezala, haien artean oso gertu egon daitezke eta gainezarri, 2.6 aiderazpena  $d_{ij} = 0$  puntuan finitua delako. Baliteke oinezkoen arteko elkarrekintza hau  $\vec{F}_{ij}^r \propto 1/d_{ij}$  forma daukan indar batez deskribatzea eta modu honetan  $d_{ij} \rightarrow 0$ ,  $\vec{F}_{ij}^r \rightarrow \infty$  da. Hala ere, bornatuak ez diren funtzio hauek erabiltzeak arazoak ekar ditzake ere numerikoki ezegonkorrak direlako eta ondorioz, metodo implizituen beharra dago.

Aipatzekoa da lan honetan 2.5 ekuazioan aurkeztutako esponenzialaren formak espezifikazio zirkularra jarraitzen duela. Formulazio hau sinpleena eta konputazionalki eraginkorra da, isotropikoa delako. Gainera, SimWalk paketeak erabiltzen du. Hala ere, bestelako espezifikazioak ere existitzen dira, espezifikazio eliptikoa, esaterako. Honen oinarria pertsonen ibiltzeko bere abiaduraren noranzkoan espazio handiagoa behar dutela da, hurrengo pausua norabide horretan emango dutelako. [16]

### 2.2.3 Hormek eragindako indarra

Oinezkoak horma eta oztopoetatik urruntzen saiatzen dira. Fenomeno hau deskribatzeko honako indarra daukagu:

$$\vec{F}_i^W = A_{i,w} e^{\frac{r_i-d_{iW}}{B_{i,w}}} \quad (2.7)$$

Oinezko-oinezko elkarrekintzaren oso antzekoa da. Hala ere, desberdintasun garrantzitsu bat dago: ibiltariak erradio karakteristiko bat daukaten zirkuluak bezala deskribatzen diren bitartean, horma eta oztopoak elementu puntualak bezala hartzen dira.

Hormek egindako indarra hurrengo modu honetan ere definitu daiteke:

$$\vec{F}_i^{wall} = \begin{cases} F_{Wall} (1 - \frac{R_{wall}}{d}) \vec{n} & d < R_{Wall} \\ 0 & \text{bestela} \end{cases} \quad (2.8)$$

Adierazpen horretan  $\vec{n}$  hormatik ateratzen den bektore perpendikularra da,  $F_{Wall}$  indar alderatzailearen intentsitatea eta  $d$ ,  $i$ . oinezkoaren eta hormaren arteko distantzia.

### 2.2.4 Bestelako gaiak eta aldakuntzak

Jatorrizko 1995. urteko ereduari aldaketa bat sar daiteke. SFM arabera abiadurak ez dauka inolako mugapenik. Horrek arazoak ekar ditzake; izan ere, bi ibiltari elkarrekiko oso gertu daudenean, indar alderatzaile oso handiak agertzen dira eta beraz, abiadura erraldoiak.



Ondorioz, abiadura maximo bat zehaztea ezinbestekoa da, errealistak ez diren efektu horiek ekiditzeko. Beraz, abiadura:

$$\vec{v}_i(t) = \vec{v}_i \cdot g\left(\frac{v_{max}}{\|\vec{v}_i\|}\right) \quad (2.9)$$

da. 2.9 adierazpeneko  $g$  funtzioa hurrengo moduan definitzen da:

$$g\left(\frac{v_{max}}{\|\vec{v}_i\|}\right) = \begin{cases} 1 & \|\vec{v}_i\| \leq v_{max} \\ \frac{v_{max}}{\|\vec{v}_i\|} & \text{bestela} \end{cases} \quad (2.10)$$

Honek esan nahi du indar erresultantea kalkulatu ondoren oinezkoaren abiaduraren moduloak  $v_{max}$  gainditzen badu, orduan abiadurak  $x$  eta  $y$  osagaien arteko erlazioa kontserbatuko du baina modulua lortu dezakeen maximoaren berdina izango da.

## 2.3 Anisotropia

Oinezkoek ez dute bere ingurunea homogeneoki norabide guztietan hautematen [6]. Hau da, oinezkoek, aleak edo molekulak ez bezala, aurreko eta atzeko partea daukate. Hauxe da ibiltariak daukaten ezaugarri propioetako bat.

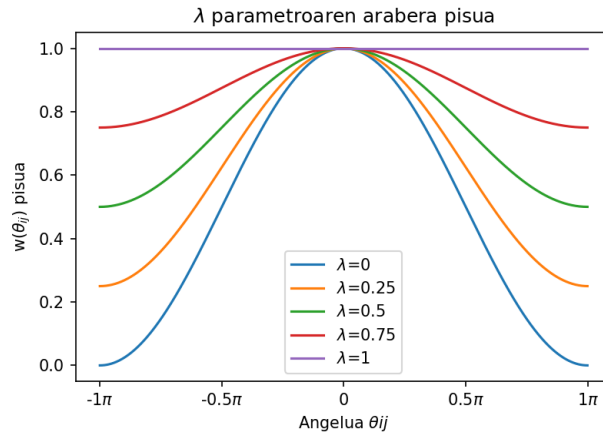
Gure ibiltzeko modua ikusten dugunaren arabera da eta argi eta garbi, gure aurrez aurre dagoenak eragin handiagoa dauka. Anisotropia hau kontuan hartzeko, oinezkoen arteko elkarrekintza, hots, 2.6 adierazpena, angeluaren menpekoea den pisu batez  $w(\theta_{ij})$  biderkatzen da:

$$w(\theta_{ij}) = \lambda + (1 - \lambda) \frac{1 + \cos(\theta_{ij})}{2} \quad \text{non } \theta_{ij} \in [-\pi, \pi] \quad (2.11)$$

2.11 formulako  $\theta_{ij} = \theta_{r_i, r_j}$  angeluak honako adierazpena du:

$$\cos(\theta_{ij}) = \frac{\mathbf{v}_i}{\|\mathbf{v}_i\|} \cdot \frac{-(\mathbf{r}_i - \mathbf{r}_j)}{d_{ij}} \quad (2.12)$$

2.2 irudian ikus daitekeen moduan,  $\lambda = 1$  denean, norabide guztiak guztiz baliokideak dira.  $\lambda = 0$  denean, ordea, aurrean dauden oinezkoek ekarpen handiagoa daukate atzean daudenean baino.



2.2. Irudia: Anisotropia

Datu esperimentalen kalibrazioaren bidez  $\lambda \approx 0.1$  proposatu da.

## 2.4 Ereduaren hedapena: larrialdi kasuak

Aurreko ataletan ohiko baldintzetan oinezkoen dinamika aztertzeke ereduaz azaldu da. Atal honetan ikusiko da orain arte azaldutako SFM ereduaz nola hedatu larrialdi kasuetarako.

Objektu edo subjektu askoren elkarrekintza ez-linealaren ondorioz bat-batean sortzen den portaera berria da larrialdia. Hurrengo ezaugarriak ditu ikara-larrialdi batek:

- Dentsitate handia
- Ohi baino azkarrago saiatzeko dira mugitzen oinezkoak
- Ibiltariak bultzaka hasten dira eta elkarrekintzak fisikoak bilakatzen dira
- Ateetan zehar igarotzeko prozesuan ez da inolako ordenarik mantentzen
- Metro karratuko 7 oinezko baino gehiago daudenean jendetza masa fluidotzat jo daiteke. Hau da, elkarrekintzen uhinak hedatu egiten dira

### 2.4.1 Testuingurua

Jende asko leku zehatz batean biltzen denean arriskua handitu egiten da. Gainera, istripu asko larrialdi kasuetan gertatzen dira. Jendetza ezbeharren zenbait adibide honako hauek dira:

- 2010. urtean Alemanikao Duisberg hirian Love Parade jaialdian 19 hildako eta 100 zauritu egon ziren. [9]
- 2015. urtean Mecarako peleginazioan ikara ihesaldi batek 2070 pertsonen bizitzarekin bukatu zuen. [8]

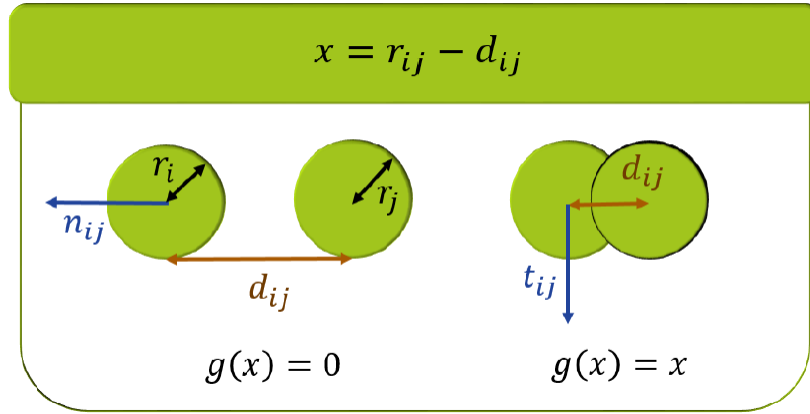
Hori guztia dela eta, egoera hauek deskribatzen duten eredu egokiak izatea ezinbestekoa da. 2000. urtean, Dirk Helbing, Illés Farkas eta Tamás Vicsek-ek jatorrizko SFM-aren bertsio berri bat proposatu zuten, larrialdi kasuetarako moldatzeko [7].

### 2.4.2 Larrialdietarako ereduak

Oinezkoen arteko elkarrekintzan, ohiko 2.5 ekuazioari beste bi gai gehitu behar zaizkio: [16][7]

$$\vec{F}_{ij}^r(d_{ij}) = A_i e^{\frac{r_{ij}-d_{ij}}{B_i}} \vec{e}_{ij} + k g(r_{ij} - d_{ij}) \vec{n}_{ij} + \kappa g(r_{ij} - d_{ij}) \Delta v_{ji}^t \vec{t}_{ij} \quad (2.13)$$

Adierazpen horretan  $k$  eta  $\kappa$  konstanteak dira.  $g(x)$  funtzioa  $g(x) = x$  da  $x > 0$  denean eta  $g(x) = 0$  bestela.  $\vec{n}_{ij} = (n_{ij}^1, n_{ij}^2) = ((\vec{r}_i - \vec{r}_j)/d_{ij})$ ,  $j$  oinezkotik  $i$ . oinezkora zuzenduta dagoen bektore normala da eta  $\vec{t}_{ij} = (-n_{ij}^2, n_{ij}^1)$ , ordea,  $\vec{n}_{ij}$  bektorearekiko perpendikularra. Gainera,  $\Delta v_{ji}^t = (\vec{v}_j - \vec{v}_i) \cdot \vec{t}_{ij}$  abiadura tangenzialen arteko diferentzia da.



2.3. Irudia: 2.13 adierazpenaren  $g$  funtzioaren esangura. Irudi hau [13] lanetik hartuta dago.

Ikusten den moduan 2.13 ekuazioaren gai guztiek  $r_{ij} - d_{ij}$  gaiarekiko menpekotasuna daukate. Hori positiboa baldin bada, orduan oinezkoak ez daude bata bestearen gainean. Negatiboa bada, aldiz, oinezkoak gainezartzen dira. Ibiltariak gainezartzen ez direnean,  $x = r_{ij} - d_{ij} > 0$  da eta beraz, soilik gai esponentziala daukagu, 2.5 ekuazioa berreskuratuz.

2.13 ekuazioaren bigarren gaia,  $kg(r_{ij} - d_{ij})\vec{n}_{ij}$ , gorputz indarra da. Malguki baten forma dauka,  $k$  konstante elastikoduna eta oinezkoak aldentzen diren heinean linealki indargabetzen da. Bere funtzioa oinezkoak elkarrengandik oso hurbil daudenean, haien arteko konpresio horri aurka egitea da. Bukatzeko, hirugarren gaia,  $\kappa g(r_{ij} - d_{ij})\Delta v_{ji}^t \vec{t}_{ij}$ , labainketa frikzioa da, pertsonak gainezartzen direnean hididura tangenziala zein zaila den adierazten duena.

Larrialdi-kasuetan, oinezkoen arteko elkarrekintzari bi gai horiek sartzeaz gain, oinezko-horma interakzioari, hots, 2.7 ekuazioari ere, bi gai baliokide gehitu behar zaizkio:

$$\vec{F}_i^w(d_{ij}) = A_i e^{-\frac{r_i - d_{iW}}{B_i}} + kg(r_i - d_{iW})\vec{n}_{iW} - \kappa g(r_i - d_{iW})(\vec{v}_i \vec{t}_{iW})\vec{t}_{iW} \quad (2.14)$$

Komentatu den bezala, oztupoak puntualak kontsideratzen dira, eta ondorioz ez daukate ez erradio ez abiadura propiorik. Honen ondorio zuzena da 2.14 ekuazioaren azken gaiak zeinu negatiboa izatea, 2.13 adierazpenean gertatzen ez den bezala.

## 2.5 Mugak eta arazo intrintsekoak

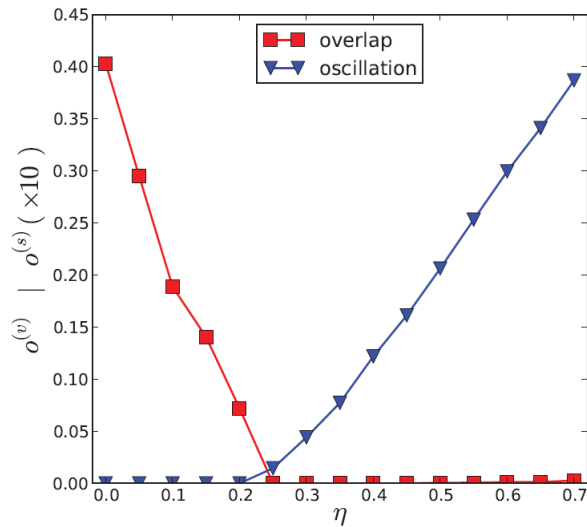
SFM oinezkoen dinamika deskribatzeko eredu aproposa da. Alabaina, atal honetan ikusiko den moduan zenbait mugapen eta arazo du:

Hasteko, oinezkoek batzuetan egoera egonkorak eratzen dituzte. Hauek adibidez, ikusiko den moduan, oinezko-kopurua txikia denean eta ate bat zeharkatu nahi dutenean gerta daitezke atearen ondoan. Indar netoa zerotik oso hurbil dago eta oinezkoek cluster bat eratzen dute, denbora luzez aurrera eta atzera egin ezinik geratzen dena. Kasu horiek ekiditeko edota ibiltari bat pilatutako ibildide batean dagoenean bestelako bide bat topatzeko, indar fluktuazioak beharrezkoak dira, hau da, estokastikoa den indar gai bat.

Beste arazoetako bat akzio-erreakzio printzipioarekin dago lotuta. Eredu hau Newtonen dinamikan dago oinarrituta, baina ez du Newtonen 3. legea betetzen. Izan ere, anisotropiaren definizioan ikusi den moduan, orokorrean oinezko batek ez dio berdin eragiten bere atzean edo

aurrean kokatuta dagoen beste ibiltari bati ( $\lambda = 1$  kasuan soilik betetzen da akzio-erreakzio printzipioa).

Larrialdi-kasuetan azaldu den bezala, oinezkoak gainezarri daitezke eta hau beste arazo bat da, bolumenaren eskusio printzipioa betetzen ez delako. Hori indar alderatzaile txikiak daudenean gertatzen da. Hala ere, gainezarpen hauek oso handiak ez direnena, gorputzaren deformazio elastiko bezala kontsidera ditzakegu. Gainezarmenarekin zuzenean lotuta dagoen beste problema bat oszilazioak dira. Batzuetan, indar alderatzaile oso handien kasuan, ibiltari bat oszilatzeko has daiteke, aurrera eta atzera eginez. Orokorrean oszilazioak agertzen direla esango da oinezko batek atzera egiten duenean (hau larrialdi kasuetan errealista ez delako). Bi fenomeno hauek, oszilazioak eta gainezarmena, guztiz ezabatzea ezinezkoa da giza indarren ereduetan. Hortaz, indar alderatzailea asko handituta gainezarmena ekidingo litzateke, baina kontrari oszilazioak indartuz. Indar alderatzailea txikituz, ordea, kontrakoa: oszilazioak desagertuko lirateke baina ibiltariak are gehiago gainezarriko lirateke. [21] lanean esaterako fenomeno hauetariko bakoitza kuantifikatzeko ratio bat definitzen da.



2.4. Irudia: Oszilazio eta gainezarpen ratioak indar alderatzailearen magnitudearen arabera. Irudi hau [21] lanetik hartuta dago.

2.4 irudian  $\eta$  hizkiaren indar alderatzailearen magnitudea adierazten da. Ikus daitezke balio baterako oszilazioak eta gainezarmenak minimizatzen direla, baina bi ratioak inoiz ez dira nulua. Beraz,  $A_i$  eta  $B_i$  balio batzuetarako, emaitza arrazongarriak lortzen dira, oszilazio eta gainezarmen kopuru minimoarekin. Balio egoki horiek zeintzuk diren eta orokorrean erudian agertzen diren beste parametro guztiak lortzeko prozedura, hurrengo atalean azalduko da laburki.

## 2.6 Datu esperimentalak eta ereduaren kalibrazioa

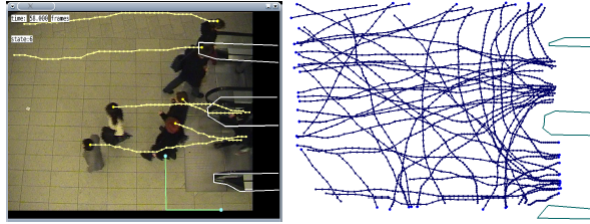
Garrantzi handikoak dira datu esperimentalak. Izan ere, eredu matematikoetarako informazioa eta kalibraziorako ezinbestekoak dira. Baina gainera modelizazio kontzeptu berriak garatzeko ere erabilgarriak izan daitezke.

Esperimentu ugari egin dira laborategi baldintzetan eredu matematikoak kuantitatiboki baliazgarriak diren frogatzeko. Ereduak kalibratzeko, esaterako, ereduaren parametroak eta

osagaiak aldatzen dira, simulazioak Weidmann-ek proposatutako fluxu dentsitate diagramak lortu arte:

$$v = v_0 \cdot \left\{ 1 - \exp \left[ -\gamma \cdot \left( \frac{1}{\rho} - \frac{1}{\rho_{max}} \right) \right] \right\} \quad (2.15)$$

2.15 adierazpenean  $\gamma = 1.913P/m^2$ ,  $v_0 = 1.34m/s$  eta  $\rho_{max} = 5.4P/m^2$  dira. Fluxu dentsitate horiei oinarritzko diagramak deritze eta oso erabiliak dira oinezkoen dinamikaren deskribapenean. Batez besteko fluxua, abiadura eta dentsitatea erlazionatzen ditu. Dena dela, eztabaida irekian dago, lortutako zenbait datu esperimental oinarritzko diagrama hauekin bat ez datoze-lako. Hori neurketa-metodologiak eta aztertutako geometriak oinarritzko diagrametan menpe-kotasuna duelako gertatzen da. Egun 2.15 oinarritzko ekuazioa erabiltzen da, bestelako datu esperimentalik erabiltzen ez bada. Ondorioz, esan daiteke benetan oinezkoen instalazioen konfigurazio egokiak lortzeko, kasu bakoitzerako oinarritzko diagrama propioa behar dela.[12]



2.5. Irudia: Irudi hau [12] lanetik hartuta dago.

Datu esperimentalak lortzeko metodo zabalduetako bat bideoaren bidezko jarraipena da. Oinezkoen ibilibideak kamera baten bidez (edo sentsoare infragorrien bidez beste kasu askotan) grabatu egiten dira gero plano bidimentsional batean adierazteko, 2.5 irudian ikusten den bezala.

### 2.6.1 Konstanteen zehaztapena eta aukeraketa

Oinezkoen interakzioa oinarritzkoak diren bi parametroen arabera zehazten da:  $A_i$  eta  $B_i$ . Konstante hauek ondo aukeratzea ezinbestekoa da ibiltarien portaera errealista izateko, bai mikroskopikoki bai makroskopikoki.

Balioak datu esperimentalen eta eredu matematikoen arteko konparazioaren bidez lortu dira. Hala ere, aztertzen den bibliografiaren arabera diferentea da eta gainera batzuetan nahas-garria izan daiteke definizio diferentek erabiltzen direlako. Hortaz,  $A_i$  izenarekin ezagutzen da energia potentzial, potentzial, indar eta azelerazio moduan definitzen den parametroei. Hauek desberdintzeko lan honetan erabiliko den notazioa hurrengo izango da:

- Energia potentziala:

$$U_{ij}^r(d_{ij}) = \tilde{A}_i e^{-\frac{r_{ij}-d_{ij}}{B_i}} \quad \text{non} \quad \tilde{A}_i = [ML^2/T^2]$$

- Potentziala:

$$V_{ij}^r(d_{ij}) = \frac{\tilde{A}_i}{m} e^{-\frac{r_{ij}-d_{ij}}{B_i}} = \hat{A}_i e^{-\frac{r_{ij}-d_{ij}}{B_i}} \quad \text{non} \quad \hat{A}_i = [L^2/T^2]$$

- Indarra, kontuan hartuta  $F = -\nabla_{d_{ij}} U_{ij}^r(d_{ij})$  dela:

$$\vec{F}_{ij}^r(d_{ij}) = \frac{\hat{A}_i}{B_i} e^{\frac{r_{ij}-d_{ij}}{B_i}} \vec{e}_{ij} = A_i e^{\frac{r_{ij}-d_{ij}}{B_i}} \vec{e}_{ij} \quad \text{non} \quad A_i = [ML/T^2]$$

- Azelerazioa, masarekin zatituz:

$$\vec{a}_{ij}^r(d_{ij}) = \frac{A_i}{m} e^{\frac{r_{ij}-d_{ij}}{B_i}} \vec{e}_{ij} = A_{i,m} e^{\frac{r_{ij}-d_{ij}}{B_i}} \vec{e}_{ij} \quad \text{non} \quad A_{i,m} = [L/T^2]$$

Lan desberdinetan honako parametroen balioak proposatu dira [19]:

### 2.1. Taula: Lan desberdinetan erabilitako parametroen balioak

Lana	$\hat{A}_i(m^2/s^2)$	$B_i(m)$
Helbing and Molnár (1995)	2.1	0.3
Helbing (1997)	2.1	0.3
Helbing et al. (2000)	2	0.08
Stucki (2003)	1.5	1.71
Lakoba et al. (2005)	2	0.5
Hoogendoorn and Daamen (2005)	1.92	0.16
Yongxiang et al. (2016)	1.34	0.08

Lan honetan koste konputazionala murrizteko zenbait parametro oinezko guztientzat berdinak kontsideratu dira. Beste ezer esaten ez bada, kontsideratutako balioak hurrengoak dira:[11][12]

### 2.2. Taula: Erabili diren parametroen balioak

Parametroa	Balioa	Unitatea
$\alpha$	0.5	s
$R_p$	2.5	m
$r_i$	[0.25,0.35]	m
$v_i^0$	[1.35,1.8]	m/s
m	58	kg
h	0.085	1
Korridorea		
$\hat{A}$	2.25	$m^2/s^2$
B	0.35	m
$\hat{A}_W$	10	$m^2/s^2$
$B_W$	0.2	m
Larrialdi egoera		
$\hat{A}$	1.34	$m^2/s^2$
B	0.08	m
k	819.62	$kg/s^2$
$\kappa$	510.49	$kg/(ms)$

## 3. Kapitulumua

# Neurtutako magnitudeen definizioak

Atal honetan, oinezkoen fluxua kuantifikatzeko, zenbait parametro definituko da [14]:

### 3.1 Batez-besteko denbora

Oinezko batek korridorea zeharkatzeko denbora  $t_c^i$  bada,  $i$  oinezko baten batez-besteko denbora honakoa da:

$$T_m^i = \frac{\sum_{i=1}^K t_c^i}{K} \quad (3.1)$$

Adierazpen horretan  $K$  da oinezko batek pasilloa zeharkatzen duen aldi kopurua. Ondoren, oinezko guztientzako batez-bestekoa egin daiteke:

$$T_m = \frac{1}{N} \sum_i^N T_m^i \quad (3.2)$$

### 3.2 Polarizazioa

Higiduraren norabidearekiko batz besteko desbiderapena da denboraren menpekia den polarizazioa  $p(t)$ . Parametro horrek ordenaren/desordenaren berri ematen du. Aipatzekoa da testuinguru honetan fisikako beste arlo batzuetan gertatzen ez den bezala, polarizazio nulua ordenarekin lotuta dagoela (oinezko guztiak paraleloan ibiliz) eta polarizazio maximoa, 1 baliokoa, desordenarekin. Polarizazioa definitu ahal izateko hurrengo definizioak behar dira:

- $\theta_i(t) \in (-\pi, \pi]$ ,  $i$ . oinezkoaren higidura-angelua da:

$$\tan(\theta_i(t)) = \frac{\vec{v}_i(t) \cdot \vec{e}_y(t)}{\vec{v}_i(t) \cdot \vec{e}_x(t)} \quad (3.3)$$

- Oinezko-talde guztiaren batez besteko higitze-angelua:

$$\tan(\theta(t)) = \frac{\langle \vec{v}(t) \cdot \vec{e}_y(t) \rangle_N}{\langle \vec{v}(t) \cdot \vec{e}_x(t) \rangle_N} \quad (3.4)$$

non  $\langle \vec{v}(t) \cdot \vec{e}_\xi(t) \rangle_N$   $N$  oinezkoentzako batz bestekoa den:

$$\langle \vec{v}(t) \cdot \vec{e}_\xi(t) \rangle_N = \frac{\sum_{j=1}^N \vec{v}(t) \cdot \vec{e}_\xi}{N}, \quad \xi \in (x, y) \quad (3.5)$$

- $d(\theta_i(t), \theta(t)) = \min_{k \in \mathbb{Z}} |\theta_i(t) - \theta(t) - 2k\pi| [0, \pi]$  tartean kokatutako  $\vec{v}_i(t)$  eta batz besteko higitze-norabidearen arteko angelua da.

Horiek guztiak kontuan hartuta, denboraren menpekoa den polarizazioa honako eran definitzen da:

$$p(t) = \frac{1}{N} \sum_{j=1}^N d(\theta_j(t), \theta(t)) \quad (3.6)$$

Askotan esanguratsua izaten da polarizazio horren denboran batz-bestekoa:

$$P = \langle p(t) \rangle_t = \left\langle \frac{1}{N} \sum_{j=1}^N d(\theta_j(t), \theta(t)) \right\rangle_t \quad (3.7)$$

### 3.3 Biskositatea

Oinezkoen fluxuaren bi biskositate mota definituko dira:

#### 3.3.1 Lerro-biskositatea

Substantzia batek jariatzeko duen erraztasunaren berri ematen du biskositateak. Fluxua handia baldin bada, orduan, biskositate txikia dago eta alderantziz.

Lerro-biskositatea definitzeko, sistemaren biskositateak denboran zehar daukan eboluzioari buruz berri ematen duena, hasteko ibilbide askea definitu behar da:

$$\Lambda_i(d_{ij}) = \begin{cases} \sum_{\Phi} d_{ij} & \text{if } \Phi \neq 0 \\ R_p & \text{if } \Phi = 0 \end{cases} \quad (3.8)$$

Adierazpen horretan  $\Phi$  da  $i$  ibiltari baten aldarapen erradioaren barruan dagoen oinezko kopurua. Ibilbide askeak,  $\Lambda_i$  delakoak, adierazten du  $i$  pertsona batek ibiltzeko daukan erraztasuna (besteekin topo egin gabe). Hori kontuan hartuta, lerro-biskositatea honako hau da:

$$\mu(t) = \frac{P(t)}{\langle v \rangle \Lambda}, \quad t \in [t_o, T_f] \quad (3.9)$$

Adierazpen honetan  $P(t)$  eta  $\langle v \rangle$  batz besteko polarizazioa eta abiadura dira, hurrenez hurren. Hasierako iragankorreko efektuak gainditzeko  $t_0$ -ren balioak behar bezain handia izan behar du. Deinizioa hau kualitatiboki eraz interpreta daiteke: hasteko, polarizazioarekiko menpekotasuna behar da. Izan ere, oinezko guztiak norabide berean higitzen baldin badira, hots, egitura bat baldin badago (lerroak eratuta), fluxua handiagoa izatea espero da. Aldiz, ibiltariak nor bere norabidean higituz badoa, inolako ordenarik gabe, kasu horretan fluxua txikiagoa da. Bestalde, abiaduraren batez bestekoarekin lotutako gai bat behar da, abiadura handitzean fluxua handiagoa izango da eta. Bukatzeko, ibilbide askeak oinezkoek libreki higitzeko gaitasuna definitzen du. Beraz, ibilbide askea handia bada, biskositate txikia dago.

#### 3.3.2 Espazioaren menpeko biskositatea

Espazioaren zein puntuetan fluxua handia edo txikia den adierazten du espazioaren menpeko biskositateak. Hasteko, oinezkoen dentsitate-funtzio orokor bat definitu da:

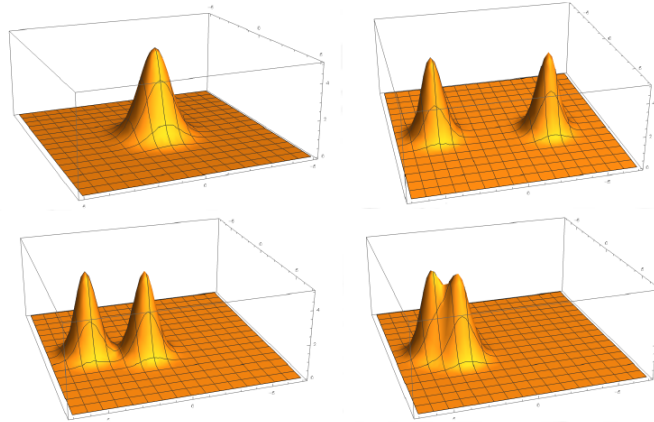
$$\Gamma(x, y) = \sum_i K_{\tau}^i(x, y) \quad (3.10)$$



Adierazpen honetan oinezko guztientzat egiten da batukaria eta  $K_\tau^i(x, y)$  da batukari-kernela da, hau da, zenbait propietate betetzen dituen funtzio integragarri eta periodikoa. Kasu honetan, funtzio hori honakoa da:

$$K_\tau^i(x, y) = \frac{1}{\pi\tau} \exp \left[ \frac{-((x - p_x^i)^2 + (y - p_y^i)^2)}{\tau} \right] \quad (3.11)$$

Adierazpen honetan  $p_x^i$  eta  $p_y^i$  dira ibiltari bakoitzaren  $x$  eta  $y$  posizioak, hurrenez hurren.

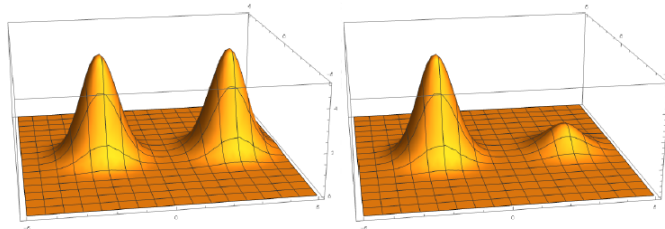


3.1. Irudia: Oinezkoen dentsitate funtzioaren irudia. Goiko ezkerreko irudian oinezko bat adierazten da. Besteetan, aldiz, bi ibiltariren dentsitate funtzioaren eboluzioa oinezkoak hurbiltzen diren heinean. [14] erreferentziatik hartutako irudia.

Dentsitate funtzioa,  $\Gamma$ , erabiliz eta banako bakoitzaren ekarpena bere abiaduraren menpekoa den pisu batez biderkatuz menpeko biskositatea lortzen da:

$$\chi(x, y) = \sum_i \frac{1}{v_i + \gamma} K_\tau^i(x, y) \quad (3.12)$$

Adierazpen honetan  $\gamma$  da aukeratutako konstante bat,  $\chi$  funtzioa  $\infty$  ez joateko  $v_i \rightarrow 0$  doanean. Kasu honetan  $\tau = 1$  eta  $\gamma = 0.2$  aukeratu dira.



3.2. Irudia: Espazioaren menpeko biskoistatearen bi irudi. Ezkerreko grafikan abiadura bereko bi oinezkoen kasua ageri da; eskumakoan, aldiz, abiadura desberdinekoak. [14] erreferentziatik hartutako irudia.

## 4. Kapitulum

# Problemaren ebazpena

Atal honetan ebatzi beharreko problemaren planteamendua eta ebazteko metodoa azalduko da.

Helburua da 2.1 bigarren mailako ekuazio diferentzial bektoriala hurrengo forman idaztea:

$$\dot{\vec{y}} = f(\vec{y}, t) \quad (4.1)$$

Hau da, kasu honetan, bigarren ordenako ekuazio diferentzial bektoriala denez, oinezko bakoitzarentzat, lehen ordenako 2 ekuazio diferentzial bektorial akoplatutan bihurtuko da. Gainera, bektore bakoitzak 2 osagai dituenek, guztira 4 ekuazio dira ibiltari bakoitzeko. Kontuan hartu behar da problemaren  $N$  ibiltari izango direla, beraz  $4N$  ekuazio guztira. Hurrengo forman idatziko da egoera bektorea:

$$\vec{y} = [x_1 \quad v_1^x \quad y_1 \quad v_1^y]^T \quad (4.2)$$

Eta ekuazio-sistema, 4.1 forman:

$$\begin{bmatrix} \dot{x} \\ \dot{v}_1^x \\ \dot{y}_1 \\ \dot{v}_1^y \\ \vdots \\ \dot{v}_N^{0y} \end{bmatrix} = \begin{bmatrix} \vec{F}_{ix}^d + \sum_w \vec{F}_{ix}^w + \sum_{i \neq j} \vec{F}_{ij,x}^r \\ \vec{F}_{iy}^d + \sum_w \vec{F}_{iy}^w + \sum_{i \neq j} \vec{F}_{ij,y}^r \\ \vdots \\ \vec{F}_{Ny}^d + \sum_w \vec{F}_{Ny}^w + \sum_{i \neq j} \vec{F}_{Nj,y}^r \end{bmatrix} \quad (4.3)$$

Kasu honetan, 4.4 egoera bektorea hedatu da oinezko bakoitzaren berezko informazioa gordezko. Hurrengo moduan definituko da:

$$\vec{y} = \underbrace{[x_1 \quad v_1^x \quad y_1 \quad v_1^y \quad \eta_1 \quad r_1 \quad v_1^{0x} \quad v_1^{0y} \dots]}_{1. \text{ oinezkoa}} \quad x_N \quad v_N^x \quad y_N \quad v_N^y \quad \eta_N \quad r_N \quad v_N^{0x} \quad v_N^{0y}]^T \quad (4.4)$$

Ikusten denez, oinezko bakoitzak 8 osagai ditu orain:  $x, v_x, y, v_y$  posizio eta abiadura bektoreen proiezioak  $x$  eta  $y$  ardatzetan;  $\eta$  aldagai konstantea noranzkoa adierazten duena. Bi noranzkoko fluxua simulatzen den kasurako esaterako,  $\eta = 0$  bada ibiltariak ezkerretik eskumara mugitu nahi du;  $\eta = 1$  bada, eskumatik ezkerreara  $\eta = 2$  bada, oinezkoa gelditu geratzen da dagoen posizioan. 5. gaia da ibiltariaren erradioa. Bukatzeko, 4.4 egoera bektorearen azken bi gaiak oinezko bakoitzaren desiratutako abaidurak adierazten dituzte.

Egoera-bektorean gehitu diren azken gai horiek konstanteak dira (desiratutako abaidura ez da denboraren menpekua izango eta ez erradioa ez noranzkoa ez dira aldatuko). Ondorioz, ekuazio-sistemak hurrengo forma hartzen du:

$$\begin{bmatrix} \dot{x} \\ v_1^x \\ \dot{y}_1 \\ v_1^y \\ \eta_1 \\ \dot{r}_1 \\ v_1^{\dot{0}x} \\ v_1^{\dot{0}y} \\ \vdots \\ v_N^{\dot{0}y} \end{bmatrix} = \begin{bmatrix} v_1^x \\ \vec{F}_{ix}^d + \sum_w \vec{F}_{ix}^w + \sum_{i \neq j} \vec{F}_{ij,x}^r \\ v_1^y \\ \vec{F}_{iy}^d + \sum_w \vec{F}_{iy}^w + \sum_{i \neq j} \vec{F}_{ij,y}^r \\ 0 \\ 0 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \quad (4.5)$$

Sistema horrek  $8N$  ekuazio dauzka. Ebazteko laugarren ordenako Runge-Kutta metodoa, ekuazio diferentzialen ebazpenerako zenbakizko metodo oso zabaldua, erabiliko da. Horren azalpen laburtua B eranskinean aurkitu daiteke. Hala ere, SFM testuinguruan, Eulerren metodo esplizitua erabiltzen da maiz, baina, kasu honetan, Euler baztertu da sorrarazten duen errore handiak direla eta.

A eranskinean egindako kodea kontsultatu daiteke. Gainera, programan zehar erabilitako funtzioak euren artean nola erlazionatzen diren adierazten duen eskema ere A eranskinean dago. Bertan frogatzen den legez, ekuazio diferentziala ebazteko SFM ereduak emandako indarrak kalkulatu behar dira lehenengo. SFM-aren zenbakizko implementazioaren pseudokodea 4.1 taulan ikus daiteke.

Prozedura hori A eranskinetako eboluzioa-funtzioaren bidez nahi den  $N_{pausu}$  aldiunetan egi-ten da. Bestalde, hasierako egoera sortzeko abiarazi funtzio bat sortu da. Horretan, oinezko guztiei hasierako posizioa, abiadura, erradioa, noranzkoa eta desiratutako abiadura esleitzen zaie. Oinezko horiek gela edo korridorearen barruan hasieratu behar dira. Gainera, korridorearen kasuan, bi aukera existitzen dira: oinezkoak hausazko posizio bateatik abiarazi edo oinezkoak korridorearen muturretatik hasieratu.

## 4.1. Taula: SFM ereduko implementazio numerikoaren pseudokodea

---

**Algoritmoa:** SFM pseudokodea

---

```

0: for all Oinezko  $i$  do
1:   if noranzkoa=2                                Oinezkoa geldi baldin badago
2:      $F_i = 0$ 
3:   else
4:      $F_i^d = 0$ 
4:      $F_i^j = 0$ 
5:      $F_i^W = 0$ 
6:     atea kalkulatu
7:     tarteko helburuak kalkulatu
8:     if  $x_i, y_i <$  tarteko helburua      tarteko helburuak zeharkatu ez baditu
9:        $F_i^d \leftarrow$  tarteko helmugara
10:    else if  $x_i, y_i <$  atea                oraindik atea zeharkatu ez badu
11:       $F_i^d \leftarrow$  atera
12:    else atea zeharkatu badu                atea zeharkatu badu
13:       $F_i^d \leftarrow$  aurrera
14:    for all Oztopo  $W$  do                    oztopo-oinezko elkarrekintza
15:       $f_i^W$  kalkulatu
16:       $F_i^W = F_i^W + f_i^W$ 
17:    for all Oinezko  $j \neq i$  do              oinezko-oinezko elkarrekintza
18:       $f_i^j$  kalkulatu
19:       $F_i^j = F_i^j + f_i^j$ 
20:     $F_i = F_i^d + F_i^W + F_i^j$ 

```

## 5. Kapitulumua

# Esperimentuak

Ereduaren frogapena egiteko aztertutako esperimentuak azaltzen dira atal honetan. Esperimentu horien emaitzak eta analisisa egin da. Bi esperimentu mota desberdin egin da: 5.1 atalean bi noranzkoko korridorea azalduko da. Horretan SFM eredu arrunta erabili da. Esperimentu horiek K letrarekin izendatuko dira. Bestalde, 5.2 atalean, gela baten ebakuazioa aztertu da larrialdi egoeran. Kasu horretan, SFM larrialdietarako hedatutako eredu erabili da. Esperimentu horiek L letrarekin bereiziko dira.

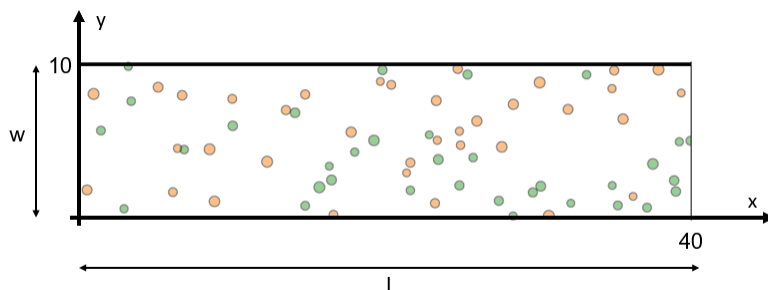
### 5.1 Bi noranzkoko korridorea

Bi noranzkoko korridorea aztertu da. 2.2 taulan eredurako erabilitako parametroen balioak jasotzen dira. Beste ezer esaten ez bada, erabili diren korridorearen dimentsioak hauek izan dira:

$$\Omega = l \times w = [0 \text{ m}, 40 \text{ m}] \times [0 \text{ m}, 10 \text{ m}] \quad (5.1)$$

Pasilloaren irudia ageri da 5.1 irudian. Aipatu behar da oinezko kopuru handia simulatzeak karga konputazionala asko handitzen duela. Ondorioz, mugalde baldintza periodikoak hartu dira kontuan, hots:

$$x(l) = x(0) \quad (5.2)$$



5.1. Irudia: Bi noranzkoko korridorea aztertzeke erabili den geometria. Gainera, ibiltarien hasierako egoera bat ageri da.

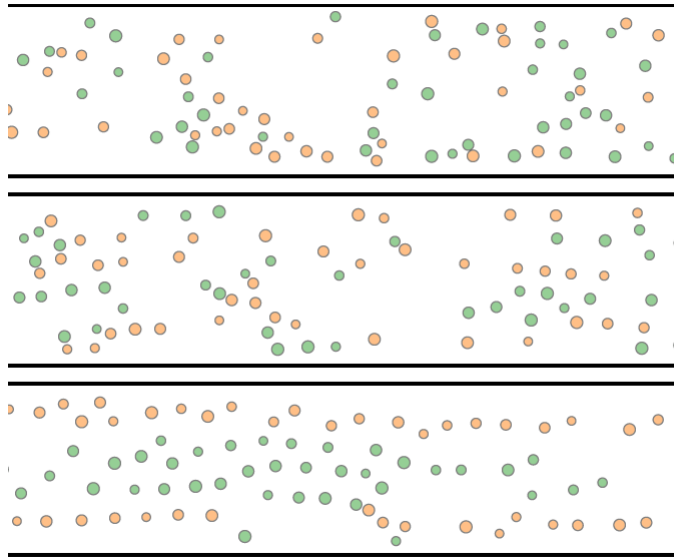
#### 5.1.1 K0. Esperimentua: lerroen eraketa

Oinezkoek lerrokatzeke joera daukate 1.2 atalean aipatu den moduan. Lehenengo esperimentu honetan ereduak eta egindako kodeak fenomeno hori erreproduzitzen duen egiaztatu da. Hurrengo parametroak erabili dira simulazioa egiteko:

5.1. Taula: K0. Esperimentuan erabili diren parametroen balioak

Parametroa	Balioa
$h$	0.085
$N_{pausua}$	500
$N$	80
$N_{left}$	40
$\lambda$	0.1

5.1 taulan  $N_{pausua}$  pausu-kopurua eta  $h$  denbora pausua, hots,  $h = (t_N - t_0)/N_{pausua}$  dira. Bestalde,  $N$  guztira korridorean dagoen oinezko-kopurua eta  $N_{left}$  ezkeretik eskumara mugitzen den oinezko-kopurua. Beraz, kontroako noranzkoan  $N - N_{left}$  dago. Bukatzeko,  $\lambda$  anisotropiaren parametroa da.



5.2. Irudia: Hasierako aldiunean, 70. aldiunean eta amaieran korridorearen irudia K0. Esperimentuan

5.2 irudian ikus daitekeen moduan, hasierako hausazko konfiguraziotik, amaierako aldiunean ibiltariak lerrokatuta daude. Lerro horien noranzko bereko ibiltariak osatzen dituzte. Beraz, emaitza bat dator 1.2 ataleko ebidentzietan aipatu denarekin. Hala ere, esan beharra dago aukeratutako muga-baldintzek emaitza hau azkarrago eta era egonkorrago batean lortzea ahalbidetzen dutela. Izan ere, lerroak korridorearen muturretan ez dira apurtzen hausazko era batean sartzen diren oinezkoekin.

### 5.1.2 K1. Esperimentua: Korridorearen zabalerearen eragina

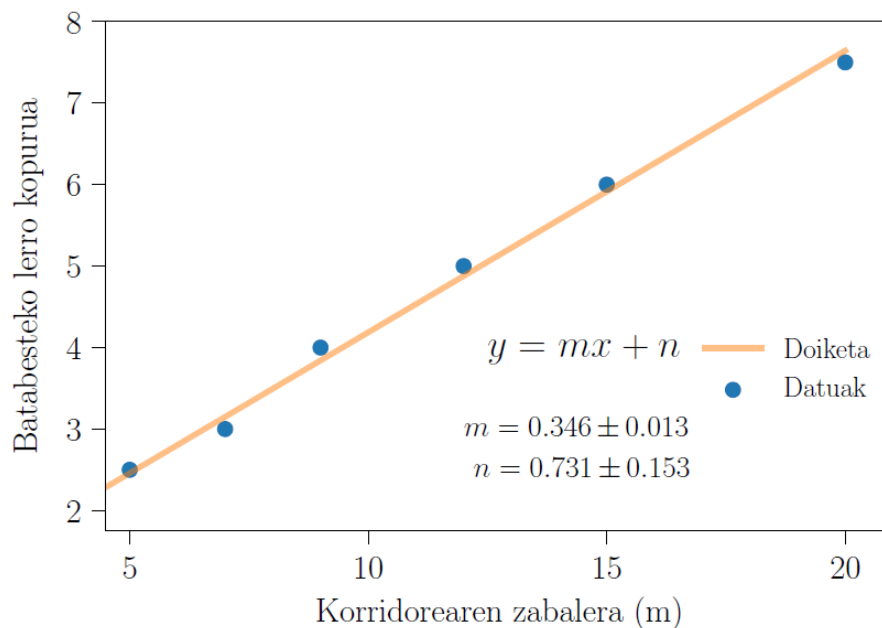
K0. Esperimentuan oinezkoak lerrokatzen direla ikusi da. K1. Esperimentu honetan eratzten den lerro kopuru hori korridorearen ( $w$ ) zabalerearekin nola aldatzen den aztertu da. Azterketa sinplifikatzeko eta baldintza berberak mantentzeko, oinezko dentsitatea mantendu da kasu guztietan zabalera aldatzen joan den heinean. Batzuetan ez dira lerro zuzen zehatzak osatu eta horregatik zaila izan da interpretazioa egitea.

Erlazio lineala dago batez besteko lerro kopuruaren eta korridorearen zabalerearen artean 5.3 ikus daitekeenez. Lortutako zuzenak hurrengo forma du:

$$lerro = (0.35 \pm 0.01)w + (0.7 \pm 0.1) \quad (5.3)$$

Emaitza hori hein handi batean bat dator [10] artikuluan lortzen den zuzenarekin. Lortzen den maldaren balioa errore tartearen barruan dago. Jatorriko ordenatua, ordea, gutxigatik kanpo gelditzen da. [10] artikuluekin honako diskrepantziak lortu dira:

$$D_{m_2}(\%) = \frac{|0.36 - 0.35|}{0.36} = \%3 \quad D_{m_1}(\%) = \frac{|0.59 - 0.7|}{0.59} = \%16 \quad (5.4)$$



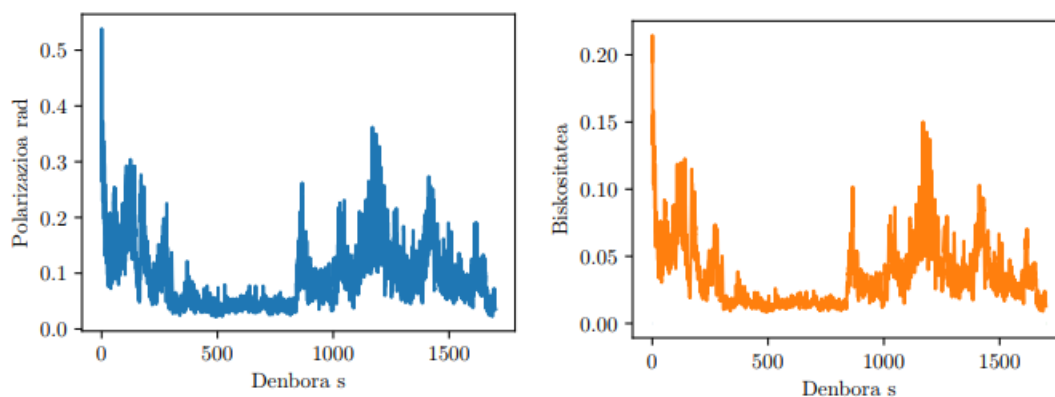
5.3. Irudia: Bi noranzkoko korridorean eraturako lerro kopurua pasilloaren zabalerearen  $w$  funtzioan

### 5.1.3 K2. Esperimentua: Polarizazioa eta biskositatea

Polarizazioa eta lerro biskositatea aztertuko dira esperimentu honetan. Simulazioa egiteko 5.2 taulako parametroak erabili dira.

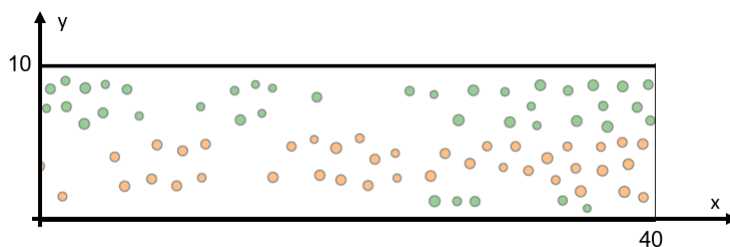
5.2. Taula: K2. Esperimentuan erabili diren parametroen balioak

Parametroa	Balioa
$h$	0.085
$N_{pausua}$	20000
$N$	80
$N_{left}$	40
$\lambda$	0.1



5.4. Irudia: K2. Esperimentuko polarizazioa eta biskositatea aldiuneeen funtzioan (20000 step edo pausu)

Lortutako polarizazioa eta biskositatea 5.4 irudian ikus daitezke.  $t = 0$  aldiuneean ikus daitezke denbora tarte txiki bat zeinean  $p(t)$  polarizazioa eta biskositatea azkar txikiagotzen diren. Ondoren, egonkortu egiten da eta balio baten inguruan fluktuatu egiten du. Hala ere, batzuetan berriro polarizazioa eta biskositatea hazi egiten dira, baina inoiz ez da hasierako polarizazio edo biskositate mailak berreskuratzen. Hasierako beherakadak sistemarentzako hasierako konfigurazioa aproposena ez dela adierazten du. Polarizazioa txikiagoa izateak oinezkoak ordenatuago (lerrokatuago) daudela adierazten du. 5.5 irudian behatu daitezke.



5.5. Irudia: K2. Esperimentuaren amaierako egoera. Ibiltariak hasierako hauzazko egoeran baino ordenatuago, lerrokatuta, daudela ikus daitezke

#### 5.1.4 Anisotropiaren eragina

Orain anisotropiaren eragina korridorea zeharkatzeko batez-besteko denboran, polarizazioan eta biskositatean aztertuko da. Horretarako, **K3. Esperimentua** eta **K4. Esperimentua** egin dira. Esperimentu bietan parametro berberak erabili dira  $N_{pausu}$  kopurua izan ezik. Erabilitako parametroen balioak 5.3 eta 5.4 tauletan daude. Ikus daitezkeenez  $\lambda$ -ren balioa 0.1 eta 0.9 tarte bitartean hartu da, 0.1-eko tartearekin.  $\lambda$ -ren balio bakoitzerako 10 simulazio egin dira esperimentu bakoitzean estatistika egin ahal izateko.

Lortutako emaitzak 5.6 irudian behatu daitezke.

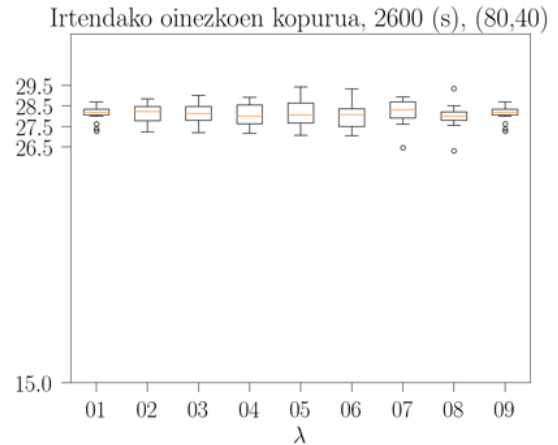
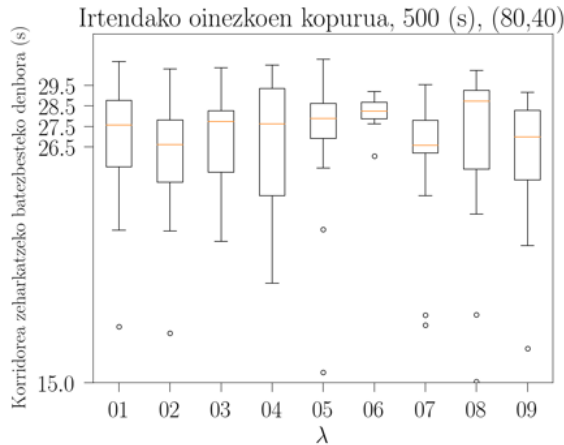


5.3. Taula: K3. Esperimentuko balioak

Parametroa	Balioa
$h$	0.085
$N_{pausua}$	500
$N$	80
$N_{left}$	40
$\lambda$	(0.1, 0.9)

5.4. Taula: K4. Esperimentuko balioak

Parametroa	Balioa
$h$	0.085
$N_{pausua}$	2600
$N$	80
$N_{left}$	40
$\lambda$	(0.1, 0.9)



5.6. Irudia: Korridorea zeharkatzeko batz besteko denbora  $\lambda$ -ren balio desberdinetarako K3 (ezkerrean) eta K4 (eskuman) esperimentuetarako. Lortutako balioak kutxa diagrama baten bidez irudikatu dira.

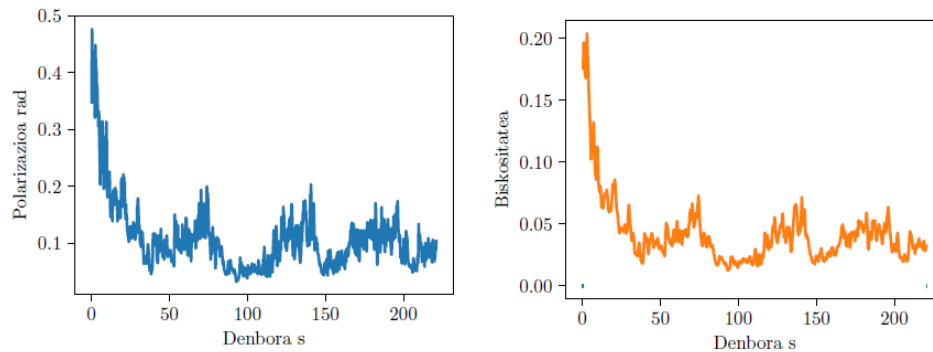
Hasteko, esan beharra dago korridorea zeharkatzeko lortutako batez besteko denborak,  $T_m \approx 28s$  ingurukoak, zentzukoak direla, korridorearen luzera eta ibiltarien batz besteko abiadurak  $l = 40 m$  eta  $v_i^0 \in [1.35, 1.8]$  izanik.

Bestalde, 5.6 irudian ikus daiteke pasabidea zeharkatzeko batez-besteko denborak ez daukela  $\lambda$  parametroarekiko menpekotasun nabarmenik. Emaitza hori ez dator bat [14] lanean lortutako ondorioarekin. Lan horretan,  $\lambda$  parametroaren balioa zenbat eta handiagoa izan orduan eta txikiagoa da batez besteko denbora. Izan ere, printzipioz,  $\lambda$  handiago batek oinezkoek atzetik indar handiagoak eragitea sorrarazten du. Ondorioz, indarrak handiagoak badira, oinezkoek abiadura handiagoak lortzen dituzte. Hori hein batean egia da, baina gogoratu behar da abiaduraren balioa mugatuta dagoela eta oinezkoek desiratutako abiadura bat dutela  $\lambda$ -ren edozein baliorako. Hori dela eta, anisotropiaren balio desberdinen artean abiadura diferentzia antzeko da eta beraz, zeharkatze denbora.

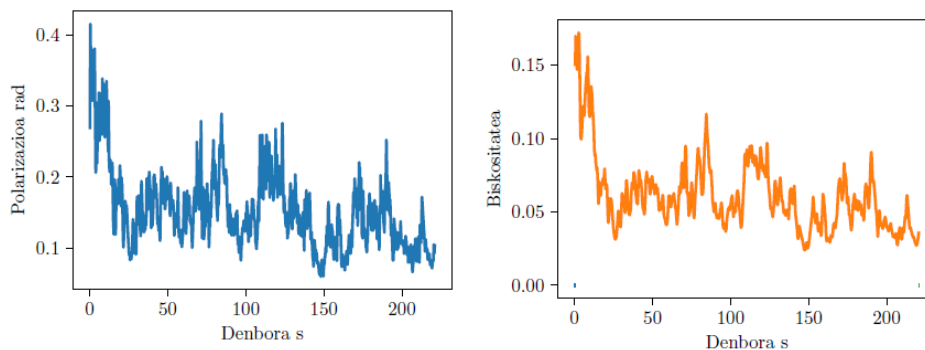
Aipagarria da K3 eta K4 esperimentuen artean desberdintasun nabarmena dagoela: bietan erdiesten diren medianaren balioak oso antzekoak dira baina K3-n datuetan minimoaren eta maximoaren arteko diferentzia askoz handiagoa da. Gauza bera gertatzen da kutxaren tamainarekin. Datuen %50-a barne hartzen duten kutxa horiek handiagoak dira K3-ren kasuan. Hauxe gertatzen da  $N_{pausua}$  pausu kopurua 5 aldiz inguru txikiagoa delako. Hortaz, hasierako egoerak datuetan menpekotasun handia dauka. Hasieran ibiltariak desordenatuta daude eta horregatik zeharkatzeko denbora diferenteak lortzen dira. Behin lerrokatzen direla, batez-besteko denbora horiek askoz homogenoagoak dira.

Bukatzeko, ohikoak ez diren balioak bi kasuetan aurkitu daitezke, baina, K3-ren kasuan horiek nabarmenagoak dira. Argi dago horrelako esperimendu bat egin eta emaitza *onak* lortzeko  $N_{pausu}$  pausu kopuruak handia izan behar duela.

Polarizazioa eta biskositatea baita ere  $\lambda$ -ren arabera aztertu dira. K4 esperimentuko bi  $\lambda$ -ren muturreko balio hartu dira hori erakusteko. 5.7 eta 5.8 alderatzen baldin badira polarizazioa handiagoa da anisotropiaren balioa handiagoa den kasurako. Emaitza hori bat dator [6] laneko emaitzarekin. Honen arabera: ibiltariak atzeko oinezkoak *sentitzen* dituztenean lerrotatze okerrak lortzen dituzte.



5.7. Irudia:  $\lambda = 0.1$  kasuan iterazioetako batean K4. Esperimentuan lortutako polarizazioa eta biskositatea



5.8. Irudia:  $\lambda = 0.9$  kasuan iterazioetako batean K4. Esperimentuan lortutako polarizazioa eta biskositatea

### 5.1.5 Oinezko kopuruaren eragina

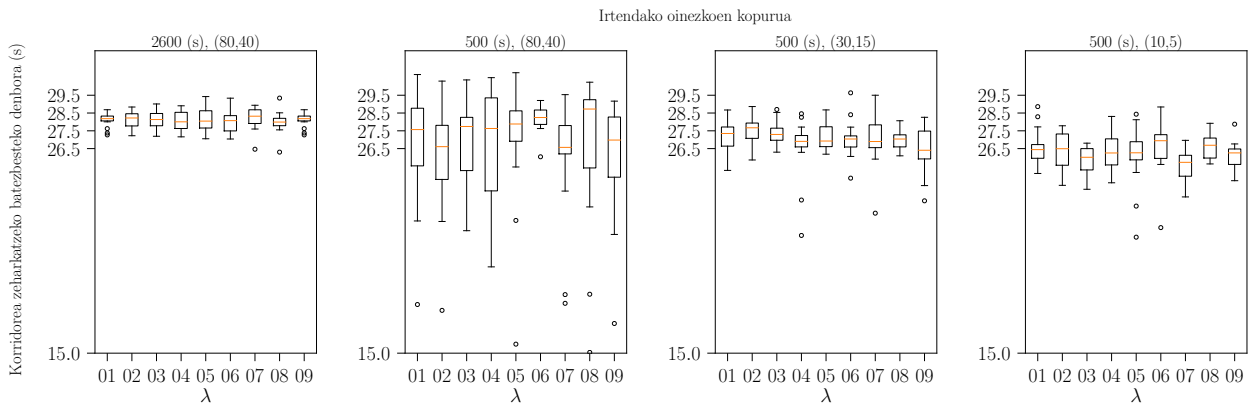
Oinezko kopuruaren eta batez besteko zeharkatze denboraren arteko erlazioa aztertzeko **K5. Esperimentua** eta **K6. Esperimentua** egin dira. Horiek, aurreko K3 eta K4 kasuen moduan,  $\lambda$  parametroaren balio desberdinetarako egin dira eta hauen balio bakoitzeko 10 alditan errepikatu da estatistika edukitzeko. Baina kasu honetan, guztira 80 oinezko egon beharrean, kopuru hori murriztu egin da:  $N = 30$  (K5. Esperimentuan) eta  $N = 10$  (K6. Esperimentuan). Erabilitako parametroak 5.4 eta 5.5 tauletan jasotzen dira.

5.5. Taula: K5. Esperimentuko balioak

Parametroa	Balioa
$h$	0.085
$N_{pausua}$	500
$N$	30
$N_{left}$	15
$\lambda$	[0.1, 0.9]

5.6. Taula: K6. Esperimentuko balioak

Parametroa	Balioa
$h$	0.085
$N_{pausua}$	500
$N$	10
$N_{left}$	5
$\lambda$	[0.1, 0.9]



5.9. Irudia: K3, K4, K5 eta K6 esperimientuetarako batz besteko zeharkatze-denbora  $\lambda$  parametroaren arabera

5.9 irudian ageri den moduan, zenbat eta oinezko gutxiago izan, orduan eta txikiagoa da zeharkatzeko batz-besteko denbora, korridorearen tamaina mantentzen den heinean. Datu hori bat dator intuitiboki pentsa litekeenarekin. Izan ere, oinezko gutxiago badago, gutxiago desbideratu beharko dira eta, ondorioz, azkarrago zeharkatuko dute pasilloa.

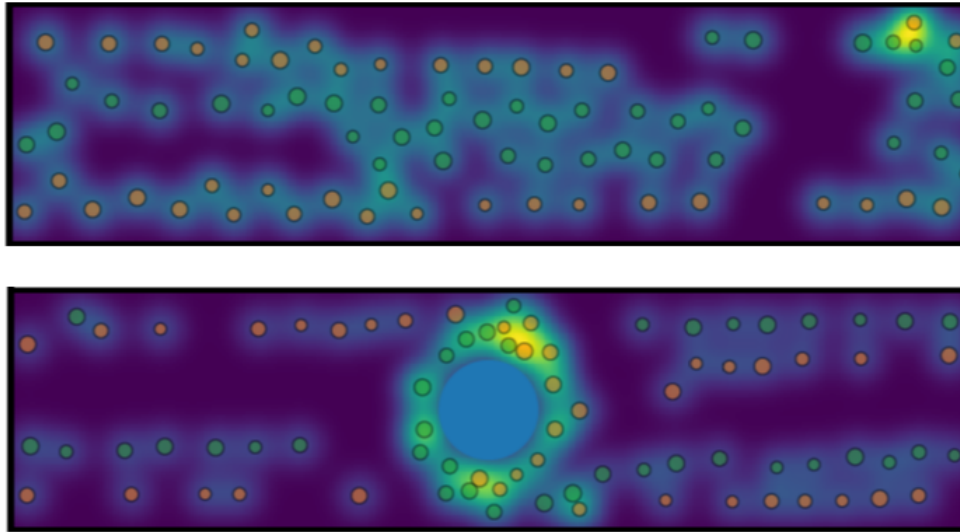
Azkeneko K5 eta K6 esperimientuetan lortutako balioak K4-n baino kontzentratuago daude, 5.9 irudian ikus daitekeenez, nahiz eta pausu kopurua berdina izan. Hori gertatzen da K5 eta K6 esperimientuetan oinezko kopurua txikiagoa delako. Ondorioz, ibiltariak gutxitan *ikusten* dituzte gainerako oinezkoak eta balio nahiko homogeenok lortzen dira zeharkatze-denboran.

### 5.1.6 K7. Esperimentua: Oztopoen eragina

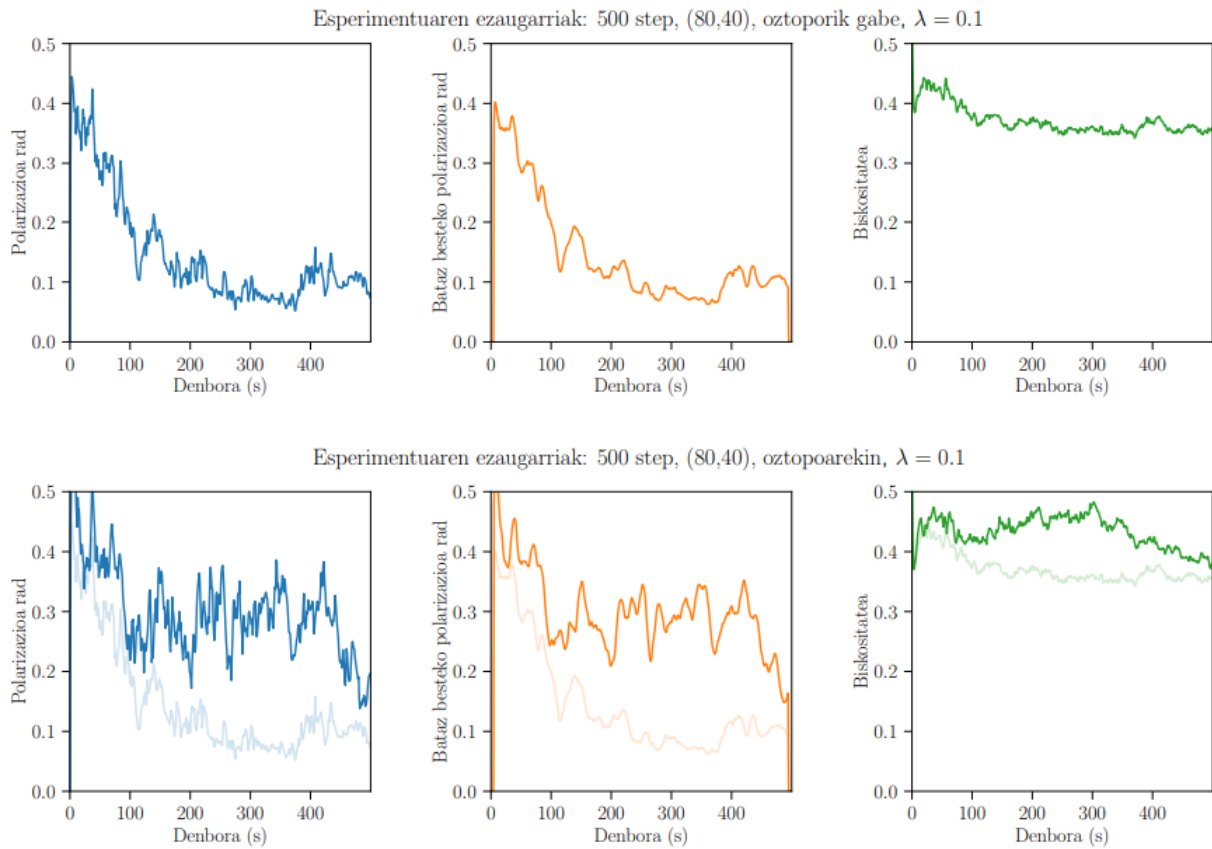
Korridorearen fluxuan oztopoen duten eragina aztertzeko K7. Esperimentua burutu da. Horretan, korridorearen erdian  $(x, y) = (20, 5)$  posizioan  $r = 2$  m-ko erradiodun zutabe bat ipini da. Egoera hori K0. Esperimentuko balioekin egin da, hots, 5.2 taulan agertzen diren parametroekin. Modu horretan, oztopodun eta oztoporik gabeko kasuak alderatu dira.

Hasteko espazioaren menpeko biskositatea aztertu da. Lortutako emaitza 5.10 irudian ageri da. Ikus daitekeen moduan, zutabedun kasuak ez du lortzen fluxua hobetzea. Izan ere, oztopoaren inguruan oinezkoak pilatzen dira: zutabeak lerroak apurtzen ditu.

Berori ikus daiteke polarizazioaren, batz-besteko polarizazioaren eta biskositatearen grafiketan, 5.11 irudia. Alegia, oztopodun konfigurazioan lortzen diren egoera egonkorreko balioak ia hirukoitzak dira: hiru aldiz desordenatuago daude.



5.10. Irudia: K7. Esperimentua: Espazioaren menpeko biskositatea zutaberik gabeko pasabidearen (goian) eta zutabedun (behean) pasabidearen kasuan



5.11. Irudia: K7. Esperimentua: Polarizazioa, batz-beste polarizazioa eta biskositatea zutaberik gabeko (goian) eta zutabedun (oztoporik gabekoarekin gainezarrita, behean) kasuetan.

## 5.2 Larrialdi-egoera: gela baten ebakuazioa

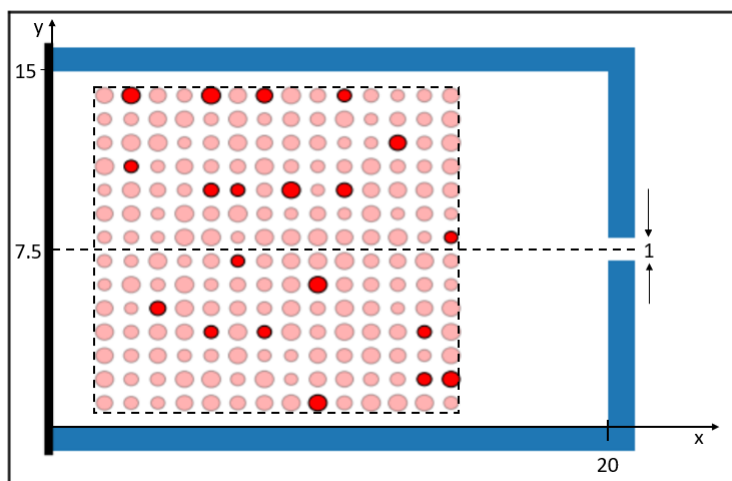
Atal honetan ate bakarreko gela baten ebakuazioa aztertu da. Helburua da oztopoek prozesu honetan nola eragiten duten ikustea. Konfigurazio ezberdinak aztertu dira.

### 5.2.1 Problemaren definizioa eta planteamendua

Gelaren dimentsioak honako hauek dira:

$$\Omega = [0 \text{ m}, 20 \text{ m}] \times [0 \text{ m}, 15 \text{ m}] \quad (5.5)$$

eta atea 5.12 irudian ikus daitekeen moduan,  $(x, y) = (20, 7.5)$  posizioan dago,  $d_w = 0.5 \text{ m}$ -ko zabalarekin. 5.12 irudian gelaren oinarritzko geometria ageri da.



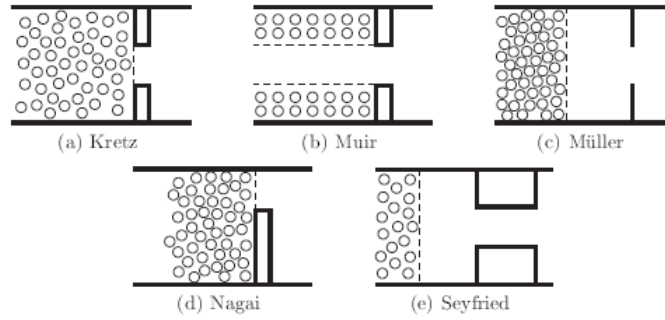
5.12. Irudia: Larrialdi kasuak aztertzeke erabiliko den gelaren geometria.

Oinezkoak hasierako egoeran  $(x \in [1, 14] \times y \in [1, 14])$  karratuaren barruan kokatuta daude, antzetzoki batera, adibidez, zutik begira egongo balira bezala. Espazio hori sare baten bidez diskretizatzen da,  $1 \text{ m}^2$ -ko tokiak izateko (pertsona bakoitzak ikuskizunean dagoen bitartean daukan espazioa). Guztira, beraz,  $14 \times 14 = 196$  toki daude.

Oinezkoak hasierako egoeran antolatzeke forma hau, 5.13 irudiko (e) kasuaren antzekoa da. Ikus daitekeen moduan beste konfigurazio forma posibleak daude eta bakoitzak egoera diferente bat adieraz dezake. Esaterako, irudiko (b) egoera, hegazkin edo autobus baten ebakuaziorekin lotu daiteke.

Bi motatako simulazioak egin dira:

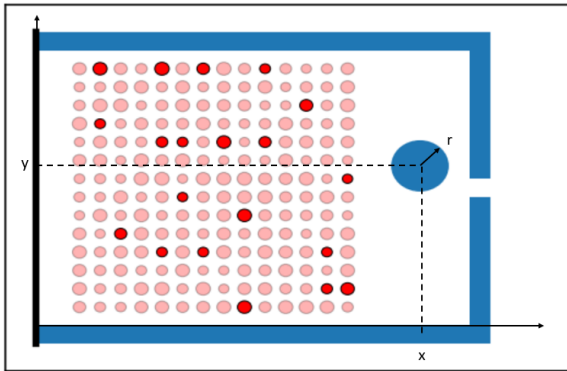
- Beteta: 196 guneak beteta daude, ondorioz 196 oinezkoekin.
- Erdi-beteta: gelan 100 oinezko bakarrik daude, beraz, soilik gune batzuk daude beteta.



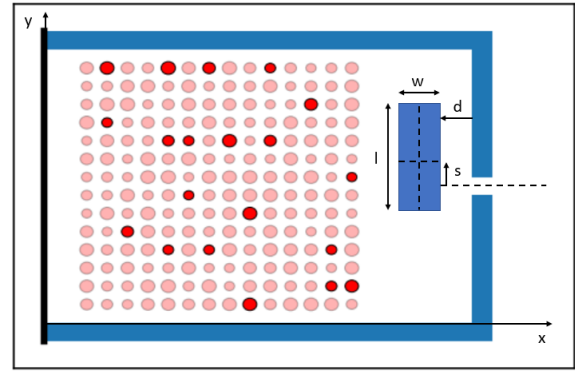
5.13. Irudia: Ebakuazioa aztertzeko oinezkoak hasieran kokatzeko proposatutako konfigurazio ezberdinak.

Gela definitzen duten hornez aparte zenbait oztopo definitu da. Bi motatakoak aintzakotzat hartu dira:

- Zutabeak: geometria zirkularrekoak. Zentroko  $(x, y)$  posizioaren eta erradioaren  $(r)$  bidez parametrizatu dira. Ikusi 5.14 irudia.
- Hormak: geometria errektangeluarrekoak. Horien luzeerak  $(l)$ , zabaleraz  $(w)$ , atea dagoen hormarekiko  $x$  ardatzeko distantziak  $(d)$  eta atearen ardatzetik  $y$  altuerak  $(s)$  zehazten dute bere posizioa. Ikusi 5.15 irudia.



5.14. Irudia: Zutabe erako oztopo baten parametrizazioa  $x, y$  eta  $r$  aldagaien bitartez



5.15. Irudia: Horma erako oztopo baten parametrizazioa  $l, d, w$  eta  $s$  aldagaien bitartez.

Indar alderatzaileak arreta handia erakarri du arlo honetan eta maiz aztertu da 2 kapitulu zehar ikusi den moduan. Izan ere, espezifikazio eliptikoa proposatu edota indar honek oszilazio-gainezarmen efektuetan duen eragina ikusi da. Indar abiarazlea, aldiz, ez da horrenbeste ikertu. Indar abiarazlea definitzen duen 2.3 ekuazioa sinplea dirudien arren, ez dago argi desiratutako norabidea nola aukeratu. Hori dela eta, oinezkoak punu bakar batera zuzentzean, esaterako, pilaketak eratzen dira (bereziki korridoreak estutzean edota korridoreen izkinetan).

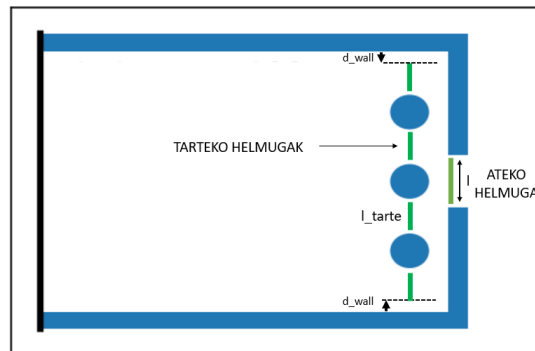
Lan honetan oinezkoak atera zuzentzeko irizpidea honakoa izan da. Atearen posizioa:

$$x = 20.0, y = \left( 7.5 - \frac{d_w}{2}, 7.5 + \frac{d_w}{2} \right) \quad (5.6)$$

puntuen bitartean baldin badago,  $d_w$  atearen zabalera izanik, atearen planoan honako puntuak dituen lerroa eratu da (ikus 5.16 eta 5.17 irudietak  $l$  lerro berdea):

$$l = \left( x = 20.0, y = \left( 7.5 - \frac{d_w}{2} + r_i, 7.5 + \frac{d_w}{2} - r_i \right) \right) \quad (5.7)$$

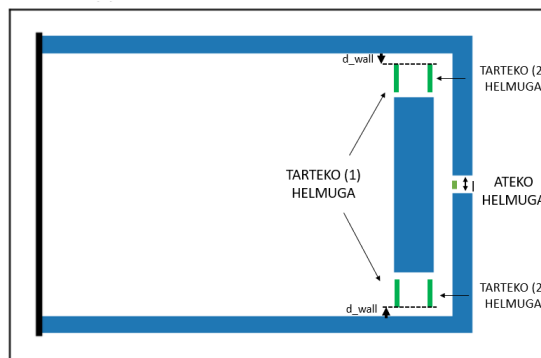
Aldiune bakoitzean  $i$ . oinezkoa eta  $l$  lerroaren arteko distantzia minimoko puntua kalkulatzeko da eta bertara zuzentzen da ibiltaria. Prozedura hori eraginkorra da eta emaitza errealistak lortzen dira. Baina kontuan hartu behar da, askotan, oinezkoarentzat atea ez dagoela ikusgai. Ondorioz, tarteko helmugak ipini behar dira. Zutabeak dituen konfigurazio batean tarteko helmuga horiek 5.16 irudian ageri dira eta hormen kasuan, 5.17 irudian.



5.16. Irudia: Tarteko helmugak zutabeen konfigurazi baterako. Zutabeen zentroak dauden planoan daude kokatuta

Ibiltariak ez dira zuzenean hormen, oztopoen edo atearen mugetara zuzentzen 5.16 eta 5.17 irudietan ikus daiteen moduan. Horien artean oinezkoen erradio maximoa, hots,  $\max(r_i) = 0.35 \text{ m}$ -ko distantzia uzten da. Balio hori bai oztopoetan, bai goiko eta beheko hormetan bai atearen kasuan aintzakotzat hartzen da esperimendu guztietarako.

Alabaina, atea zentroan dagoenez, oinezkoak normalean zentrorra joateko joera daukate. Horregatik, kodea fintzeko asmoz, goiko eta beheko hormekiko  $d_{wall}$  parametroa definitu da (ikus 5.16 edo 5.17). Aldagai horiek tarteko helmuga horiek gelaren goiko eta beheko hormekiko zein distantziatara hasten diren adierazten du. Simulazioetan  $d_{wall} = 0.35$  hartu den arren, etorkizunean parametro honen balioa aldatzearen eragina aztertuko da.



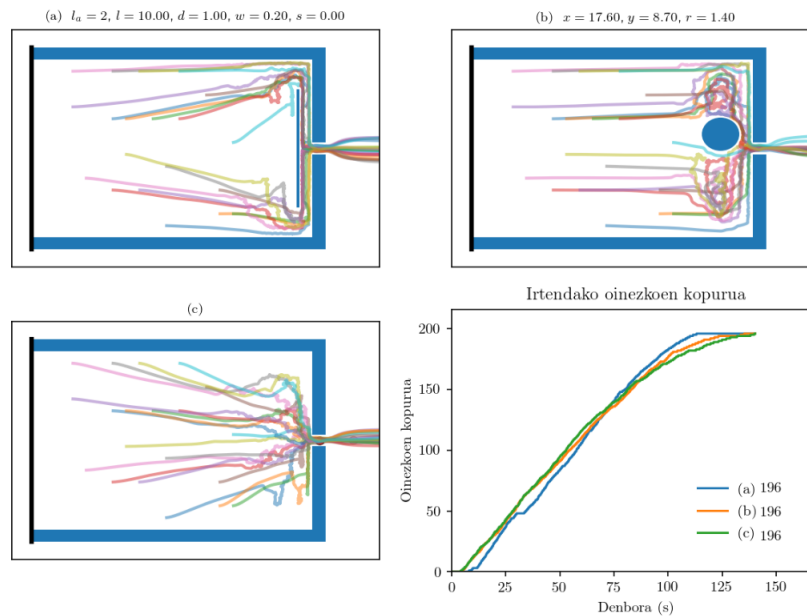
5.17. Irudia: Tarteko helmugak hormen konfigurazio baterako

### 5.2.2 Emaitzak eta analisisa

Atal honetan [17] erreferentzian agertzen den konfigurazioa hartu da oinarri moduan. Bertan 3 egoera aztertzen dira: oztoporik gabeko kasua, horma bat ate ondoan kokatuta eta zutabe bat ate ondoan ipinita. Artikulu horretan horma eta zutabe horien posizio optimoa bilatzen da eboluzio diferentziala, *Differential evolution (DE)*, optimizazio-metodoaren bitartez. Lan honetan, posizio optimo horien antzeko balioak aintzakotzat hartuta, lehenengo esperimentuan egoera hori errepikatuko da. Esperimendu horri L0.Esperimentua deituko zaio. Egindako kodearekin beste ikerketetan lortzen diren emaitzekin bateragarriak diren ondorioak lor daitezkeela ikustea da helburua. Behin hori frogatuta, hurrengo esperimenduetan, zutabe eta hormen posizioa aldatuko da eta beren kopurua ere bai.

Azterketa egiteko esperimendu bakoitzean hiru egoera diferente aztertuko da, euren arteko konparaketa egin ahal izateko. Esperimendu bakoitza 4 grafikatan laburtuko da: lehenengo 3 irudiak egoera bakoitzaren konfigurazioa eta parametroak adieraziko dira. Bertan hausaz aukeratutako zenbait oinezkok jarraitutako ibilbidea markatuta geldituko da. 4. Irudi batean, konfigurazio bakoitzean irtendako ibiltari-kopurua denboraren funtzioan adieraziko da.

#### L0. Esperimentua



5.18. Irudia: L0. Esperimentua N=196 oinezkoen kasuan

5.18 irudian ikus daitezkeen moduan, N=196 oinezkoen kasuan, (a) egoera, hau da, horma daukan konfigurazioa da eraginkorrena. Hala ere, aipatzekoa da hasiera batean egoera hori dela txarrena, zeren eta oinezkoek denbora luzeago behar duten atera heltzeko, egin beharreko distantzia handiagoa delako. Bestalde, printzipioz, harritzekoa izan daiteke oztoporik gabeko konfigurazioa txarrena izatea. Era erraz batean azaltzen da: oztopoek fluxua ordenatzea lortzen dute eta, beraz, oinezkoak arinago ateratzea lortzen dute. Emaitza [17] artikuluan agertzen den ondorioarekin bateragarria da.

Aipatu beharrekoa da denbora pasa ahala oinezkoen irtete-ratioa txikiagotzen dela (5.18-ko 4. grafikoko malda txikiagotuz doa denboraren funtzioan). Hauek batzuetan atearen ondoan erutzen diren *clusterren* ondorio dira. Izan ere, aldiune batzuetan ez aurrera ez atzera gelditzen dira *cluster* horiek.

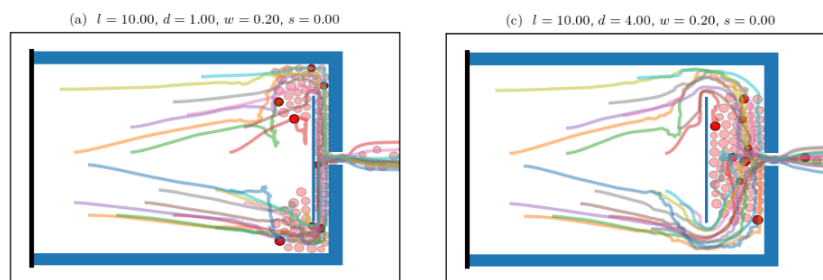


$N=100$  kasuan, aldiz,  $C$  eranskinean ikus daitekeen moduan, zutabearen (b) konfigurazioa da efizienteena. Hori ez da harritzekoa. Izan ere,  $N=196$  oinezkoen kasua, 5.18 irudiko laugarren grafikari erreparatzen bazaio, oinezko kopurua 100 denean 3 egoerak nahiko parekatuta daude. Ondorioz, hasierako kokapenaren arabera bata edo bestea izan daiteke eraginkorrena.

Behin ikusita  $N=196$  oinezkoen kasuan horma bat ate aurrean ipintzea onurak dituela, horma horren posizioa eta tamaina aldatu da bi faktore horien menpekotasuna ikusteko. Hurrengo esperimentuak egin dira:

- **L1. Esperimentua:** Hormaren  $d$  distantzia, hau da, atearikiko  $x$  distantzia aldatu da. Gainerako parametroak,  $L0$ . Esperimentukoaren berdinak izan dira.  $d = 1m$  ( $L0$ . Esperimentukoa),  $d = 2m$  eta  $d = 4m$  kontsideratu dira.

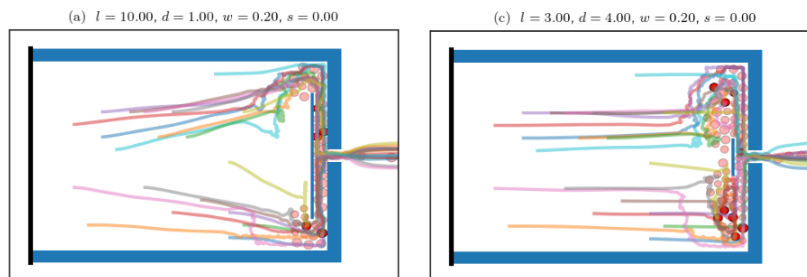
Esperimentu honetan argi ikusten da (ikus  $C$  eranskina), bai  $N=196$  bai  $N=100$  kasuan,  $L0$ . Esperimentuko konfigurazioa eraginkorrena dela. Hau da, atetik gertu ipini behar da horma fluxua ordenatzea nahi badugu. Izan ere, horma urrun ipinita,  $d = 4m$ -tara esaterako, fluxua ez da ordenatzea lortzen eta oztoporik gabeko kasuan bezala, oinezkoek osatutako erdizirkunferentzia bat osatzen da atearen aurrean 5.19 irudian ikus daitekeen moduan.



5.19. Irudia: L1. Esperimentua  $N=196$  oinezkoen kasuan  $t=58s$ -ko egoera

- **L2. Esperimentua:** Esperimentu honetan hormaren luzera aldatu da gainera parametroak mantenduz.  $l = 10m$ ,  $l = 6m$  eta  $l = 3m$  aintzat hartu da.

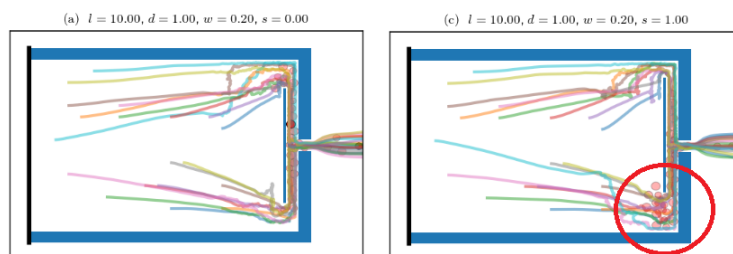
$C$  eranskinean ikus daitekeen modua, berriro ere,  $L0$ . Esperimentuko balioa, hots,  $l = 10m$ , da efizienteena. Luzera txikiagoko hormek ez dute emaitza onik ematen. 5.20 irudian ikus azter moduan, horma luzea denean, oinezkoak zig-zag formazioan kokatzen dira bi hormen artean. Horma txikiaren kasuan, aldiz, banan-banan antolatzen dira eta horregatik irtetzen abiadura motelagoa da.



5.20. Irudia: L2. Esperimentua  $N=196$  oinezkoen kasuan  $t=72.6s$ -ko egoera

- **L3. Esperimentua:** 3. Esperimentu honetan hormaren atearen ardatzarekiko  $s$  posizioa aldatu da. L0. Esperimentuan justu atearen simetrikoki zegoen kokatuta horma ( $s = 0$ ). Orain, beste bi kasu aztertu dira, horma gora  $s = 0.5m$  eta  $s = 1m$  eramanez.

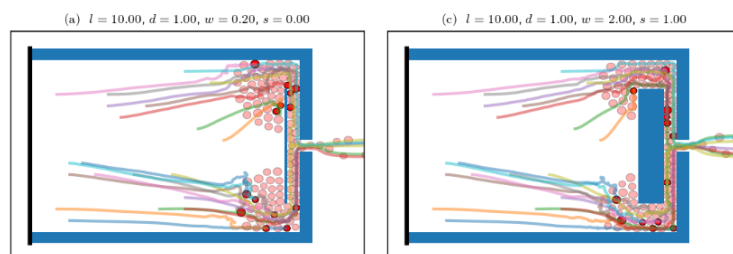
C eranskinean ageri den moduan, horma simetria ardatzetik desplazatzeak ez du laguntzen. Izan ere, gora eramatean, beheko partean oinezko gehiago pilatuko da eta horiek irtetzeko denbora luzeagoa beharko dute 5.21 grafikari ikus daitekeen moduan.



5.21. Irudia: L3. Esperimentua N=196 oinezkoren kasuan  $t=85s$ -ko egoera

- **L4. Esperimentua:** Hormen konfigurazio desberdinekin bukatzeko, 4. Esperimentu honetan, hormaren  $w$  zabalera aldatu da. L0. Esperimentuko  $w = 0.2m$ -etaz gain,  $w = 1m$  eta  $w = 2m$  aztertuko da.

Kasu honetan horma zabalagoak lagungarriak direla ikusi da (C). Izan ere, fluxua are gehiago ordenatzen da, ez soilik  $y$  norabidean baizik eta zabalera horregatik  $x$  norabidean ere (ikusi 5.22).

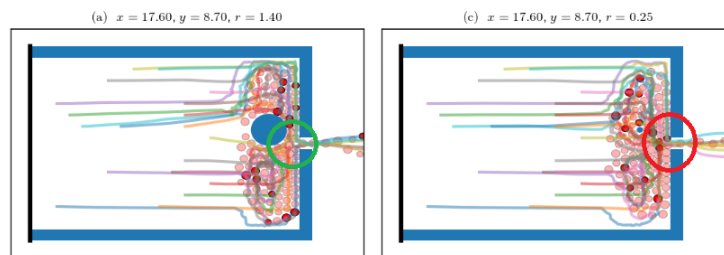


5.22. Irudia: L4. Esperimentua N=196 oinezkoren kasuan  $t=42.5s$ -ko egoera

Orain zutabe bakarraren parametroak aldatu dira euren eragina aztertzeko. Betiere L0. Esperimetuko zutabearen balioak hartuko dira oinarritzat, hots,  $x = 17.6$ ,  $y = 8.7$  eta  $r = 1.4m$ .

- **L5. Esperimentua:** Esperimentu honetan zutabearen erradioa aldatu da.  $r = 1.4m$ -ko erradioaz gain,  $r = 0.5m$  eta  $r = 0.25m$  hartu da.

C eranskinean behatu ahal den moduan, zutabearen erradioaren murrizketak ez du laguntzen fluxua ordenatzen. Izan ere, 5.23 grafikari aztertu daitekeen moduan,  $r = 0.25m$ -ko egoeran ate ondoan ez da fluxua ordenatzen.



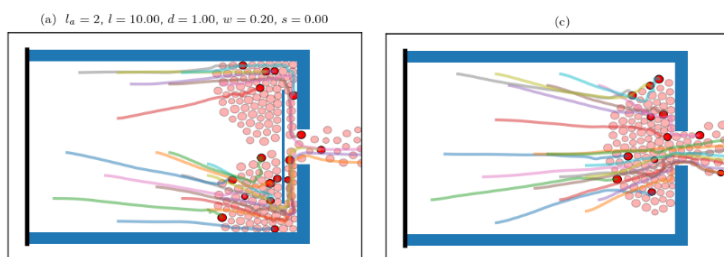
5.23. Irudia: L5. Esperimentua  $N=196$  oinezkoen kasuan  $t=45.9s$ -ko egoera

- **L6. Esperimentua:** Oraingoan zutabearen  $y$  posizioa aldatu da. L0. Esperimentuan zutabe hori ez zen justo atearen aurrean ipintzen ( $y = 8.7$ ). Kasu honetan, ate aurrean, hots,  $y = 7.5$ , eta  $y = 10$  posizioan ipini da ere. Esperimentu honetan, konfigurazio desberdinen artean ez da diferentzia nabarmenik behatu eta kasu guztietan antzeko irtete denborak lortu dira.
- **L7. Esperimentua:** Zutabearen posizio desberdinekin bukatzeko, zutabea  $x$  ardatzean zehar mugitu da,  $x = 16.6$ ,  $x = 17.6$  eta  $x = 17.85$  hartuz. Berriro ere, L6. Esperimentuan gertatzen zen bezala, ez da aipagarriak diren diferentziak aztertu.

Orain arte atearen zabalera  $1m$ -koa izan da. Aztertu den hurrengo esperimentuan, **L8 Esperimentuan**, L0. Esperimentuaren parametro berdinak hartu dira baina atearen zabalera  $2m$ -koa.

Kasu honetan oztoporik gabeko konfigurazioa da eraginkorrena. Hori gertatzen da kasu honetan horma oso gertu dageolako esperimentu hontan dagoen atearen zabalerrako. 5.24 irudiari erreparatuz, hormak irtetzen diren oinezko kopurua mugatzen du. Izan ere, ibiltarien bi ilara sortzen dira, baina atearen zabalera aldiunero 3 oinezko irtetzea ahalbidetzen du. Ondorioz, horma  $d$  distantzia handiago batera ipini beharko litzateke. Antzeko egoera bat gertatzen da zutabearekin.

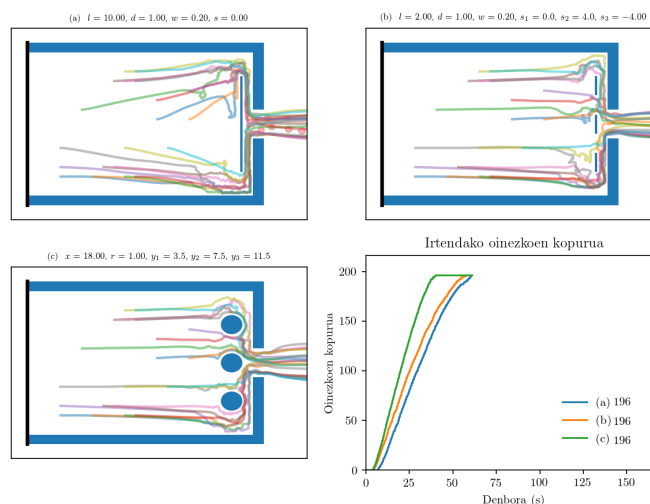
Esperimentu honetan beraz argi ikusten da, besteak beste, hormaren  $d$  distantzia eta atearen zabalera erlazionatuta daudela.



5.24. Irudia: L8. Esperimentua  $N=196$  oinezkoen kasuan  $t=13.77s$ -ko egoera

Bukatzeko beste bi esperimentu egingo dira baina kasu honetan hormen eta zutabeen kopurua handituz. Modu honetan egindako kodeak daukan aplikagarritasun handia ikus daiteke.

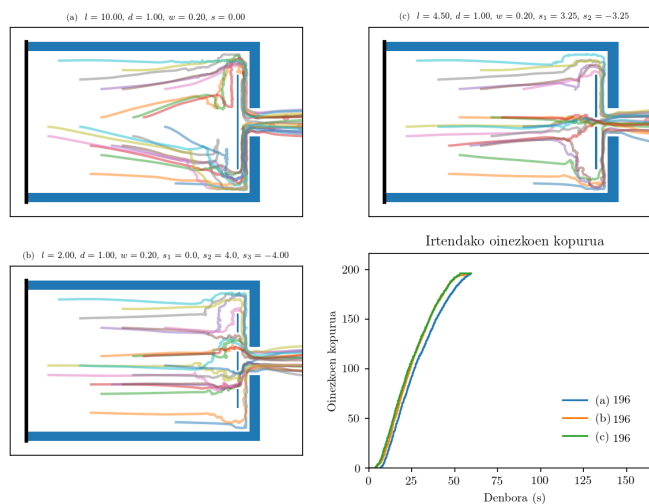
**L9. Esperimentua** Esperimentu honen helburua zutabe luze bat ipini beharrean, antzeko papera jokatzen duten 3 zutabe edo horma txikiagok nola jokatzen duten ikustea da. Konfigurazioa zehazki 5.25 irudian agertzen da.



5.25. Irudia: L9. Esperimentua N=196 oinezkoen kasuan

5.25 grafikari erreparatzen badiogu, argi ikusten da 3 zutabeak efizienteenak direla. Hori baliteke L8 esperimentuan azaldu den atearen zabalerarekin berriro ere zerikusia izatea. Izan ere, kasu honetan berriro ere atearen zabalera  $d_w = 2m$ -koa da. Hala ere, bai lerrokatutako 3 zutabeek zein hormatxoek fluxua ordenatzea lortzen dute era efiziente batean, oinezkoak lau taldetan banatzen baitituzte.

**L10. Esperimentua:** Azkenego esperimentu honetan justo atearen altueran horma bat izan beharrean, espazio librea uztea erabaki da bertatik oinezkoak igarotzeko. Ondorioz, horma luzea bi zatitan banandu da hauetako konfigurazio batean. Beste biak ohiko horma luzea eta L9. Esperimentuko lerrokatutako 3 hormatxoak izango dira (ikusi 5.26).



5.26. Irudia: L10. Esperimentua N=196 oinezkoen kasuan

Emaitzari erreparatzen bazaio lerrokatutako 3 hormatxoek eta 2 hormek emaitza oso antzekoa lortzen dute eta beti horma luzeak baino hobea.

# Ondorioak eta etorkizuneko lana

Atal honetan lanean zehar lortutako ondorioak eta etorkizunean garatzeko gaiak azalduko dira. Hasteko, ondorioak honako hauek izan dira:

- Bi noranzkoko fluxuaren kasuan:
  - oinezkoak noranzko bereko lerroetan antolatzen dira
  - lerro-kopurua korridorearen zabalera araberakoa da
  - anisotropiak ez dauka ibiltarien zeharkatzeko batez-besteko denboran eraginik. Bai, ordea, polarizazioan. Zenbat eta anisotropia handiagoa izan, orduan eta polarizazio handiagoa lortzen da
  - batez-besteko zeharkatze denborak behera egiten du geometria mantentzen bada eta oinezko kopurua gutxituta
  - korridorearen zentroan ipinitako  $2 m$ -ko erradiodun zutabeak fluxuaren ordenaketan ez du lagundu. Hala ere, oztopoen beste konfiguraziorik ez da aztertu, agian helburu honetarako egokiak litezkeenak.
- Larrialdi-kasuko gelaren ebakuazioari dagokionez:
  - atearen ondoan oztopoak ipintzeak fluxua ordenatzen laguntzen du eta, ondorioz, eraginkorragoak dira larrialdietan *tokiak* husteko
  - Horma bakarraren kasuan atearekiko simetrikoki kokatu behar da, atearen zabalera araberak distantziara. Izan ere, atearen zabalera hormaren atearekiko  $d$  distantzia horretan eragin handia du. Horrez gain, hormak luzeera eta zabalera handikoak izan behar du, fluxua hobeto ordenatze aldera. Kasu zehatz honetan, parametro egokiak  $l = 10 m$ ,  $w = 2 m$ ,  $s = 0 m$  eta  $d = 1 m$  izan dira, hormaren zabalera  $d_w = 1 m$  izanik
  - Zutabearen kasuan erradio txikiko oztopoek ez dute eraginik.  $r = 1.34 m$  izan da aztertutako parametroen artean egokiena. Bestalde, atetik hurbil kokatzea komenigarria da, baina ez da ikusi eragin handirik  $x, y$  posizio horretan
  - egindako esperimentuetan horma luzeen ordez, lerrotuta dauden horma txikiak edo zutabeak efizienteagoak izan dira

Hala ere, lortutako emaitza hauek guztiak kualitatiboak dira eta benetan gela bateko konfigurazio optimoena bilatzeko egoera ugari aztertu beharko lirateke. Horretarako oztopoak definitzen dituzten parametro guztiak kontua hartu beharko lirateke aldi berean eta programazio generikoa edo, eboluzio diferentziala bezalako optimizazio-teknika baten bitartez, oinezkoen hustutze-denbora minimizatu.

Erabilitako programari eta inguruneari dagokienean, alde batetik, Jupyter Notebook aplikazioak daukan erabilgarritasuna frogatu da. Bereziki, gaur egun, talde-lanak garrantzi handia daukala, programatzeko oso aukera ona ematen baitu. Izan ere, programa-kodea eta azalpenak era xume batean tartekatzeko aukera eskeintzen du. Gainera, aplikazio desberdinak dituzten pakete eta libreriak erabiltzeko erraztasuna ematen du.

Bestalde, Python lengoiarekin trebatzeko aukera ona izan da. Aipatzekoak dira bereziki `numpy.array`-ek ematen duten malgutasuna. Konputazionalki ohiko listak baino askoz azkarragoak dira eta zientzia arloan oso erabiliak.

Etorkizunari dagokionez, egindako lanak izan dezakeen aplikagarritasuna medio, lanean sakontzen jarraitzeko asmoa daukat. Adibidez, orain, hildakoekin lanean nabil. Izan ere, gela baten ebakuazioan askotan eror daitezkeen oinezkoak daude eta, beraz, horien eragina aztertzea ezinbestekoa da.

Egindako kodea oraindik hainbat alderditan orokortu daiteke:

- Ereduan ez da fluktuaziorik aintzat hartu. Horiek lagungarriak izan daitezke, adibidez, *clusterrak* saihesteko.
- Ereduan ez da oinezkoen arteko erakarpena aintzat hartu
- Egindako kodeak oztopoak edonon kokatzea ahalbidetzen du baina ez tarteko helmugak
- Lan honetan korridoreak eta gelak soilik aztertu dira. Bestelako geometria konplexuak aztertzeko, bidegurutzeak esaterako, orokotu liteke
- Python Objetoetara zuzendutako lengoaia dela aprobetxatuz, elementuak (ibiltariak, geometriak, oztopoak eta abar) objektu moduan definitu litezke dagozkien parametroekin

Bukatzeko, ereduaren aplikagarritasuna bestelako sistemetan azter daiteke. Bi noranzko fluxuaren egoera, adibidez, errepideetako autoen edota zainetako zelulen fluxuarekin lotu daiteke. Zelulen berezko parametroak (abiadura, tamaina, etab) definituz eta kolesterola, esaterako, oztopotzat hartuz, arlo horretan aplikatu liteke eredu eta hori ere bai buruan daukat.

# Bibliografía

- [1] Hankin, B. D., Wright, R. A. (1958). *Passenger flow in subways*, *Operational Research Quarterly*, 9, 81.
- [2] Castellano, C., Fortunato, S. and Loreto, S. 2009. *Statistical physics of social dynamics*
- [3] Henderson, L. F., 1971, *Nature* 229, 381.
- [4] Hoogendoorn, S.P and Daamen, W. 2005. *Traffic and Granular Flow* Springer Berlin Heidelberg, 1st edition.
- [5] ETH Zurich (Swiss Federal Institute of Technology in Zurich), Prof. Dr. rer. nat. Dirk Helbing Chair of Sociology, in particular of Modeling and Simulation <http://webarchiv.ethz.ch/soms/teaching/>
- [6] Gulikers L, Evers J, Muntean A and Lyulin A. *The effect of perception anisotropy on particle systems describing pedestrian flows in corridors*. *Journal of Statistical Mechanics*. 2013; 24 pages.
- [7] Helbing, D., Farkas, I. and Vicsek, T. (2000). *Simulating dynamical features of escape panic*, *Physical Review Letters*, 84, 1240.
- [8] October 29, (2015), Death toll in Saudi haj disaster at least 2,070: Reuters tally. Retrieved August 1, 2017 from <http://www.reuters.com>.
- [9] Stampede at German Love Parade festival kills 19. [Internet]. London: BBC; 25 July 2010. <http://www.bbc.com/news/world-europe-10751899>.
- [10] Helbing, D. and Molnar, P. (1995). *Social force model for pedestrian dynamics*, *Physical Review E*, 51, 4282.
- [11] Steiner, A., Philipp, M. and Schmid, A. (2007). *Parameter Estimation for a Pedestrian Simulation Model*, 7th Swiss Transport Research Conference, Ascona.
- [12] Johansson, A., Helbing, D. and Shukla, P. K. (2007). *Specification of the Social Force Pedestrian Model by Evolutionary Adjustment to Video Tracking Data*, *Advances in Complex Systems*, 10, 271.
- [13] Trigo López, M (2017). *Universality of the Fundamental Diagram in Pedestrian Dynamics. A study based on social force models*.
- [14] Johansson, A (2016). *Bi-Directional flow in the Social Force Model*
- [15] SimWalk: Simulazio Softwarea <https://www.simwalk.com/>
- [16] Gonzalez Arostico, J.D (2011) *Estudio de modelos para tráfico peatonal y su aplicación a pasillos en diferentes escenarios*

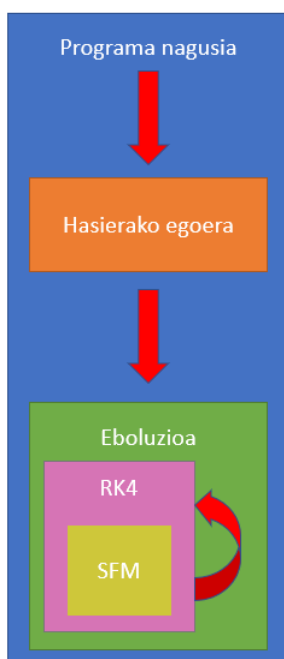
- 
- [17] Yongxiang Zhaoa, Meifang Li, Xin Luc, Lijun Tian, Zhiyong Yud, Kai Huanga, Yana Wanga and Ting Li (2016) *Optimal layout design of obstacles for panic evacuation using differential evolution*
- [18] Klüpfel, H. L. (2003) *A Cellular Automaton Model for Crowd Movement and Egress Simulation*, PhD Thesis, University Duisburg-Essen, Germany.
- [19] A. Steiner, M. Philipp and A. Schmid (2007) *Parameter Estimation for a Pedestrian Simulation Model*, 7<sup>th</sup> Swiss Transport Research Conference
- [20] [https://en.wikipedia.org/wiki/Runge%E2%80%93Kutta\\_methods](https://en.wikipedia.org/wiki/Runge%E2%80%93Kutta_methods)
- [21] M. Chraibi *Validated force-based modeling of pedestrian dynamics*
- [22] <https://www.numpy.org/>
- [23] <https://jupyter.org/>



## A. Eranskina

# Programak

Programaren funtzio desberdinek honako eskema hau jarraitzen dute:



A.1. Irudia: Programan dauden funtzioek jarraitzen duten oinarrizko eskema

Atal honetan erabilitako programak ageri dira.

### A.1 Bi noranzkoko fluxua

Korridorean bi noranzkoko fluxua aztertzeko erabilitako SFM eta eboluzioa funtzioak ageri dira. Zenbait funtzio gehiago ere erabili da baina larrialdi-egoeren berdinak direnez, solik atal horretan daude.

#### A.1.1 SFM funtzioa

Funtzio honek SFM ereduaren implementazioa du egoera arrunt baterako (larrialdi-egoera ez dena). Oinezko bakoitzean eragindako indarrak kalkulatu ditu. Gainera, aldiuneko polarizazioa eta biskositatea ere bueltatzen ditu. Horretarako aldiuneko egoera bektorea,

oinezko kopurua, ezkerretik eskumara doan ibiltari kopurua, korridorearen geometria eta anisotropia jasotzen ditu argumentu moduan.

```
def socialforce(t, y, kop, numberleft, geometria, landa):
    with_horma = True
    if geometria[3] == 0.0:
        with_horma = False

    def hormara(horma, y_i0, y_i1, y_i2, y_i3, v0, alfa, x_target):

        minimo=100
        for k in horma:
            d = abs(k - y_i2)
            if d < minimo:
                minimo = d
                retorna=k

        desx=(x_target-y_i0)
        desy=(retorna-y_i2)
        vec=np.array([desx, desy])
        desx=((vec/np.sqrt(vec@vec))*v0)@np.array([1,0])
        desy=((vec/np.sqrt(vec@vec))*v0)@np.array([0,1])
        abiarazleax=(desx-y_i1)/alfa
        abiarazleay=(desy-y_i3)/alfa

        return abiarazleax, abiarazleay

alfa=0.5 #erlajazio denbora

#Beste oinezkoen eraginez alderatzailea:
A=6.43
B=0.35
Ahorma=50
Bhorma=0.2
Rp=2.5 #ze distantziatik aurrera indarra nulua

h=geometria
#oztopoak = oztopoak

#Deribatuaq gordetzeko bektorea
F=np.empty(8*kop)

#BISKOSITATEA kalkulatzeko aldagaiak
j=-1 #kontadorea
vbatazbeste=np.zeros(kop) #bataz-besteke abiadurak gordetzeko bektorea
ibilbideaskea=np.zeros(kop) #ibilbide askea gordetzeko bektorea

ibilaskea=0 #jendetza guztiaren ibilbide askea gorde
bisk=0 #biskositatea gordetzeko

vx=np.empty(numberleft) #oinezko bakoitzaren abiaduraren x osagaia gordetzeko
vy=np.empty(numberleft) #oinezko bakoitzaren abiaduraren y osagaioa gordetzeko
tang=np.empty(numberleft)
pol=np.empty(kop)

#i. oinezkoak pairatzen dituen indarrak kalkulatzeko:
for i in range(0,8*kop,8):
    if(y[i+4])==2: #oztopo bat baldin bada
        F[i:i+8]=[0,0,0,0,0,0,0,0]
    else:
        j=j+1
        #Balio desberdinak hasieratu behar ditugu
        repulsivex=0
        repulsivey=0
        anisotrop=0
        landa=landa #anisotropia aztertzeko: (0,1) balioen artean har ditzake
        peopleinvicinity=0 #zenbat pertsona zirkunferentziaren barruan

#ABIARAZLEA:
zutab=geometria[: -8]
```

```

v0=np.sqrt(y[i+6]**2+y[i+7]**2)
hutsuneak=list()
if with_horma=False and geometria[2]!=0:

    behe=np.arange(0.0+0.35,zutab[1]-zutab[2]-0.35,0.1)
    goi=np.arange(zutab[-3]+zutab[-2]+0.35,15.0-0.35,0.1)
    for s in range(len(behe)):
        hutsuneak.append(behe[s])

    for w in range(0,len(zutab)-4,4):
        l=np.arange(zutab[w+1]+zutab[w+2]+0.35,zutab[w+5]-zutab[w+6]-0.35,0.1)
        for f in range(len(l)):
            hutsuneak.append(l[f])

    for c in range(len(goi)):
        hutsuneak.append(goi[c])

elif with_horma=True:
    behe=np.arange(0.0+0.35,zutab[1]-0.35,0.1)
    goi=np.arange(zutab[-1]+0.35,15.0-0.35,0.1)
    for s in range(len(behe)):
        hutsuneak.append(behe[s])

    for w in range(0,len(zutab)-4,4):
        l=np.arange(zutab[w+3]+0.35,zutab[w+5]-0.35,0.1)
        for f in range(len(l)):
            hutsuneak.append(l[f])

    for c in range(len(goi)):
        hutsuneak.append(goi[c])

if (y[i+4]==0):
    if with_horma:
        if (y[i]>geometria[2]):
            abiarazleax=(y[i+6]-y[i+1])/alfa
            abiarazleay=(y[i+7]-y[i+3])/alfa
        else:
            if y[i]<geometria[0]:
                abiarazleax, abiarazleay = hormara(hutsuneak,y[i],y[i+1],y[i+2],
                y[i+3],v0,alfa,geometria[0])
            else:
                abiarazleax, abiarazleay = hormara(hutsuneak,y[i],y[i+1],y[i+2],
                y[i+3],v0,alfa,geometria[2])
        else:
            if (geometria[2]==0): #oztoporik gabeko kasua
                abiarazleax=(y[i+6]-y[i+1])/alfa
                abiarazleay=(y[i+7]-y[i+3])/alfa
            else:
                if (y[i]>geometria[0]):
                    abiarazleax=(y[i+6]-y[i+1])/alfa
                    abiarazleay=(y[i+7]-y[i+3])/alfa
                else:
                    abiarazleax, abiarazleay = hormara(hutsuneak,y[i],y[i+1],y[i+2],
                    y[i+3],v0,alfa,zutab[0])
elif (y[i+4]==1):
    if with_horma:
        if (y[i]<geometria[2]):
            abiarazleax=(y[i+6]-y[i+1])/alfa
            abiarazleay=(y[i+7]-y[i+3])/alfa
        else:
            if y[i]<geometria[0]:
                abiarazleax, abiarazleay = hormara(hutsuneak,y[i],y[i+1],y[i+2],
                y[i+3],v0,alfa,geometria[0])
            else:
                abiarazleax, abiarazleay = hormara(hutsuneak,y[i],y[i+1],y[i+2],
                y[i+3],v0,alfa,geometria[2])
        else:
            if (geometria[2]==0): #oztoporik gabeko kasua
                abiarazleax=(y[i+6]-y[i+1])/alfa
                abiarazleay=(y[i+7]-y[i+3])/alfa
            else:

```

```

        if (y[i]<geometria[0]):
            abiarazleax=(y[i+6]-y[i+1])/ alfa
            abiarazleay=(y[i+7]-y[i+3])/ alfa
        else:
            abiarazleax , abiarazleay = hormara(hutsuneak ,y[i] ,y[i+1],y[i+2],
            y[i+3],v0, alfa , zutab[0])

for w in range(0, len(h) ,4):
    if (h[w+3]>h[w+1]):
        peddistance , obstaclex , obstacley = pointRectangleDistance(y[i] , y[i+2],
        h[w] , h[w+1], h[w+2], h[w+3])
    else:
        peddistance , obstaclex , obstacley =pointCircleDistance(y[i] ,y[i+2],
        h[w] ,h[w+1],h[w+2])

    vec=np. array ([y[i]-obstaclex ,y[i+2]-obstacley ])
    vec=vec/peddistance

    robstacle=y[i+5]
    repulsivex = repulsivex + Ahorma*np. exp((robstacle-peddistance)/Bhorma)
    *(vec@np. array ([1 ,0]))

    repulsivey = repulsivey + Ahorma*np. exp((robstacle-peddistance)/Bhorma)
    *(vec@np. array ([0 ,1]))

for k in range(0, kop*8 ,8):
    if (k!=i):
        vec=np. array ([y[i]-y[k] ,y[i+2]-y[k+2]])
        peddistance = np. sqrt (vec@vec)
        vec=vec/peddistance
        r0=y[i+5]+y[k+5]
        if (peddistance < Rp):
            peopleinvicinity=peopleinvicinity+1
            ibilbideaskea [j]=ibilbideaskea [j]+peddistance

            v=np. array ([y[i+1],y[i+3]])
            v=v/np. sqrt (v@v)
            a=-np. array ([y[i]-y[k] ,y[i+2]-y[k+2]])/peddistance
            z=np. cos (v@a)

            anisotrop = landa + (1-landa)*(1+z)*0.5
            r0=y[i+5]+y[k+5]
            repulsivex = repulsivex + anisotrop*A*np. exp((r0-peddistance)/B)
            *(vec@np. array ([1 ,0]))

            repulsivey = repulsivey + anisotrop*A*np. exp((r0-peddistance)/B)
            *(vec@np. array ([0 ,1]))
            #Abiaduraren balioa mugatuta dago:

v0=3.0
if (np. sqrt (y[i+1]**2+y[i+3]**2)>v0):
    vec=np. array ([y[i+1],y[i+3]])
    y[i+1]=((vec/np. sqrt (vec@vec))*v0)@np. array ([1 ,0])
    y[i+3]=((vec/np. sqrt (vec@vec))*v0)@np. array ([0 ,1])

if (y[i+4]==0):
    if y[i+1]==0:
        vx[j]=0.01
    else:
        vx[j]=y[i+1]

    vy[j]=y[i+3]

F[i:i+8]=[y[i+1], abiarazleax+repulsivex ,y[i+3], abiarazleay+repulsivey ,0 ,0 ,0 ,0]

#BISKOSITATEAREN KALKULUA

```

```

vbatzbeste[j]=np.sqrt(y[i+1]**2+y[i+3]**2) #batz beste abiadura gorde

if (peopleinvicinity!=0):
    ibilbideaskea[j]=ibilbideaskea[j]/peopleinvicinity #ibil askea normalizatuta
else:
    ibilbideaskea[j]=Rp #erradio barruan ez badago oinezkorik,

#POLARIZAZIOAREN KALKULUA
#Behin ibilbideaskea[j] bektorean oinezko bakoitzaren ibilbideaskea kalkulatu
#dugula, orain jendetza guztiarena kalkulatu dugu

#Polarizazioaren kalkulurako

tang=vy/vx
tang_mean=sum(vy)/sum(vx) #batz beste tangentea
ang=np.arctan(tang)
ang_mean=np.arctan(tang_mean) #batz beste angelua
ang=ang-ang_mean
ang=np.arcsin(np.sin(ang)) #Gure angelua -pi eta pi tartean egoteko
ang=abs(ang)
pol=sum(ang)/len(ang)

ibilaskea=np.mean(ibilbideaskea)
bisk=1/(ibilaskea*np.mean(vbatzbeste)) #biskositatea kalkulatzeko

return F, bisk, pol

```

## A.1.2 Eboluzioa

Programa honek zehaztutako denbora tartean hasierako egoeraren oinezko guztien eboluzioa kalkulatu du.

```

def eboluzioa(tN, N, y0, numberpeople, leftx, rightx, bottony, upy, geometriak,
landa, oztopoak, option=1, numberobstacle=0):

    y00 = y0.flatten()

    t0 = 0 #hasierako aldiunea
    tN = tN #amaierako aldiunea
    N = N #####
    h = (tN - t0) / N #pausua

    denbora = np.zeros(N)
    posizioax = np.zeros(N)
    posizioay = np.zeros(N)
    posizioax2 = np.zeros(N)
    posizioay2 = np.zeros(N)
    biskositatea = np.zeros(N)
    polarizazioa = np.zeros(N)
    linebisk = np.zeros(N)

    meantime = np.zeros([numberpeople,10])
    meantimekont = np.zeros(numberpeople)

    spacebis = 0 #Biskositate espaziala gordetzeko aldagaia

    #guztira = np.zeros(N)
    guztira = list()

    oztopoak = oztopoak

    def figure_dinamika(figure, axis, left_x_lim, right_x_lim, left_y_lim, right_y_lim,
axline_1, axline_2, oztopoak,):

        axis.set_xlim(left_x_lim, right_x_lim);
        axis.set_ylim(left_y_lim, right_y_lim);
        axis.axhline(y=axline_1, color='k');
        axis.axhline(y=axline_2, color='k');
        axis.yaxis.set_major_locator(ticker.NullLocator());

```

```

axis.xaxis.set_major_locator(ticker.NullLocator());
axis.axhline( 0.00, 0.00, 1.00, color='k');
axis.axhline(10.00, 0.00, 1.00, color='k');
#axis.set_aspect("equal", adjustable="datalim");

for oztopo in oztopoak:
    if oztopo[0] == 'rectangle':
        Rec(axis, oztopo[1], oztopo[2], oztopo[3], oztopo[4])
    else:
        Cir(axis, oztopo[1], oztopo[2], oztopo[3])

def figure_datuak(figure, axis, left_x_lim, right_x_lim, left_y_lim, right_y_lim):
    axis.set_title('Irtendako_oinetzkoen_kopurua');
    axis.set_xlabel('Denbora_s');
    axis.set_ylabel('Oinezkoen_kopurua');
    axis.set_xlim(left_x_lim, right_x_lim); #x ardatza definitu
    axis.set_ylim(left_y_lim, right_y_lim); #y ardatza definitu

colors = list(mcolors.TABLEAU_COLORS.values())

for i in range(N):
    time = datetime.now()
    t0, y00, bisk, pol = rk4(socialforce, h, y00, t0, numberpeople, numberleft,
    geometriak, landa)
    denbora[i] = t0
    biskositatea[i] = bisk
    polarizazioa[i] = pol

    print('****', '{:4d}'.format(i)+'_iterazioa', '*****', '{:07.2f}'.
    format(denbora[i]), '*****Time_elapsed_calculations:_', datetime.now() - time)

    time = datetime.now()

    fig, axes = plt.subplots(nrows=1, ncols=1, figsize=(8,6), );# sharex=True, sharey=True,;
    ax0 = axes

    figure_dinamika(fig, ax0, 0.00, 40.00, -15.00, 25.00, 0.00, 10.00, oztopoak,)
    #Hemen aldatzen da irudiaren tamaina

    for j in range(0, numberpeople*8, 8):
        if (y00[j+4]==2): #obstacle bada
            axes.add_artist(plt.Circle((y00[j], y00[j+2]), y00[j+5], color='k'));

        elif (y00[j+4]==0): #ezkerrera badao blue
            if (y00[j+6]==1.7): #stress bada
                axes.add_artist(plt.Circle((y00[j], y00[j+2]), y00[j+5], color='p'));

            else:
                axes.add_artist(plt.Circle((y00[j], y00[j+2]), y00[j+5], color='b')); #

        else: #eskumara badao red
            if (y00[j+6]==-1.7): #stress bada
                axes.add_artist(plt.Circle((y00[j], y00[j+2]), y00[j+5], color='p'));

            else:
                axes.add_artist(plt.Circle((y00[j], y00[j+2]), y00[j+5], color='r')); #

    if (y00[j]>rightx):
        k=int(j/8)
        if (meantimekont[k]==0):
            meantime[k, int(meantimekont[k])] = t0
        else:
            meantime[k, int(meantimekont[k])] = t0
            guztira.append(t0-meantime[k, int(meantimekont[k]) - 1])

        meantimekont[k] = meantimekont[k] + 1
        y00[j] = leftx
    elif (y00[j]<leftx):
        k=int(j/8)
        if (meantimekont[k]==0):
            meantime[k, int(meantimekont[k])] = t0

```

```

        else:
            meantime[k, int(meantimekont[k])] = t0
            guztira.append(t0 - meantime[k, int(meantimekont[k] - 1)])
            meantimekont[k] = meantimekont[k] + 1
            y00[j] = rightx

print('****', '{:4}'.format('')+ 'XXXXXXXXXX', 'XXXXXXXXXX', '{:7}'.format(''),
      '****Time_elapsed_representations:', datetime.now() - startTime)

plt.tight_layout();
plt.savefig(os.path.join(os.path.expanduser("~"), "Desktop", "gif1", str(i) + ".png"),
            format="png", dpi = 150);
plt.close(fig=fig);

#GIF FITXATEGIA EGITEKO KODEA: _____
#main = os.path.join(os.path.expanduser("~"), "Desktop", "gif1")
#images = list() #lista batean irudi desberdinak gordetzen ditugu
#for i in range(N):
#    images.append(imageio.imread(os.path.join(main, str(i) + ".png")))
#imageio.mimsave(os.path.join(main, "korridoreaber.gif"), images, duration=0.1)

fig, ax=plt.subplots();
ax.plot(denbora[1:], polarizazioa[1:]);
ax.set_ylabel("Polarizazioa_rad");
ax.set_xlabel("Denbora_s");
plt.savefig(os.path.join(os.path.expanduser("~"), "Desktop", "gif1", "polarizazioa.png"),
            format="png", dpi = 150);

plt.close(fig=fig);

#Bataz besteko polarizazioa kalkulatzeko
gelakopurua=int(0.5/h)

fig2, ax2=plt.subplots();
polbb=np.zeros(N)
for i in range(gelakopurua, len(polbb)-gelakopurua):
    polbb[i]=sum(polarizazioa[i-gelakopurua:i+gelakopurua])/(2*gelakopurua)

ax2.plot(denbora[:gelakopurua], polbb[:gelakopurua], linestyle='—');
ax2.plot(denbora[gelakopurua:-gelakopurua], polbb[gelakopurua:-gelakopurua]);
ax2.plot(denbora[-gelakopurua:], polbb[-gelakopurua:], linestyle='—');

ax2.set_ylabel("Bataz besteko_polarizazioa_rad");
ax2.set_xlabel("Denbora_s");
plt.savefig(os.path.join(os.path.expanduser("~"), "Desktop", "gif1", "bbpolarizazioa.png"),
            format="png", dpi=150);

plt.close(fig=fig2);

linebisk=polbb*biskositatea
fig3, ax3=plt.subplots()
ax3.plot(denbora[:gelakopurua], linebisk[:gelakopurua], linestyle='—');
ax3.plot(denbora[gelakopurua:-gelakopurua], linebisk[gelakopurua:-gelakopurua]);
ax3.plot(denbora[-gelakopurua:], linebisk[-gelakopurua:], linestyle='—');
ax3.set_ylabel("Biskositatea")
ax3.set_xlabel("Denbora_s")
plt.savefig(os.path.join(os.path.expanduser("~"), "Desktop", "gif1", "biskositatea.png"),
            format="png", dpi=150)

plt.close(fig=fig3)

if (len(guztira)!=0):
    ratio=sum(guztira)/len(guztira)
else:
    ratio="Oraindik_oinezko_bakar_batek_ez_du_korridore_guztia_zeharkatu"
return ratio

```

## A.2 Larrialdi egoera: gela baten ebakuazioa

### A.2.1 Erabilitako pakete eta libreriak

Atal honetan larrialdi egoerako kasua garatzeko erabilitako programa eta paketeak ageri dira. Horrez gainera, nola izendatu diren ere ikus daiteke.

```
import numpy as np
import matplotlib.pyplot as plt
import os
import imageio
import random
from matplotlib import animation
from mpl_toolkits.mplot3d import Axes3D
from decimal import Decimal
import matplotlib.patches as patch
from matplotlib.patches import Rectangle, Circle
import matplotlib.ticker as ticker
from datetime import datetime
import random as rd
import matplotlib.colors as mcolors
import math as ma
from matplotlib import rc
from pathlib import Path
```

### A.2.2 Hasierako egoeraren definizioa

Hasierako egoera definitzen duen funtzioa. Oinezko guztiei dagozkien parametroak esleitzen zaizkie, hala nola, posizioak, abiadurak, erradioak, desiratutako abiadurak, etab. Funtzio honek oinezko kopurua, ezkeretik eskumara doan oinezko kopurua, estresatutako (abiadura handiagoa) daukan oinezko kopurua, gelaren dimentsioak, informazioa gordeko den konputagailuaren helbidea,

```
def abiarazi(numberpeople, numberleft, numberstress, leftx, rightx, bottony, upy, path,
            banaketa=0, option=1, numberobstacle=0):

    numberpeople=numberpeople+numberobstacle

    y0 = np.zeros((numberpeople, 8))
    y00 = np.zeros((numberpeople, 9))

#####
    x=np.zeros((numberpeople))
    y=np.zeros((numberpeople))

    if banaketa == 0:
##### 1 ##### hasierakoposizioax
        #Hasierako posizioak zehazteko bi aukera : 1)Option=0: muturretatik 2)Option=1: random
        posizioa=0

        if (option==0):

            for index, value in enumerate(x):
                if (index<=numberleft):
                    x[index] = leftx
                else:
                    x[index] = rightx
            elif (option==1):
                for index, value in enumerate(x):
                    posizioa=(rightx - leftx) * np.random.random() + leftx
                    if (posizioa not in x):
                        x[index] = posizioa
            else:
                print("Parametroren_bat_ez_da_egokia")

##### 2 ##### hasierakoposizioay
        posizioa=0

        for index, value in enumerate(y):
```



```

        posizioa=(upy - bottony) * np.random.random() + bottony
        if (posizioa not in y):
            y[index] = posizioa

x = np.linspace(1.00, 14.00, round(numberpeople**0.5)) # number_people**0.5
y = np.linspace(1.00, 14.00, round(numberpeople**0.5))
X, Y = np.meshgrid(x, y)
X = X.reshape((np.prod(X.shape),))
Y = Y.reshape((np.prod(Y.shape),))
x, y = X, Y

else:
    ## generate the grid
    x_grid = np.linspace(1.00, 14.00, 14)#round(numberpeople**0.5) # number_people**0.5
    y_grid = np.linspace(1.00, 14.00, 14)#round(numberpeople**0.5)
    X, Y = np.meshgrid(x_grid, y_grid)
    X = X.reshape((np.prod(X.shape),))
    Y = Y.reshape((np.prod(Y.shape),))

    for index, value in enumerate(x):
        pair = [0,0]
        while pair in [[i,j] for i,j in zip(x,y)]:
            pair = [np.random.choice(X), np.random.choice(Y)]
        x[index] = pair[0]
        y[index] = pair[1]
        #print('x= ',x)
        #print('y= ',y)

##### 3 ##### noranzkoa
noranzkoa=np.zeros((numberpeople))

hautatuak=list() #hemen gordetzen ditugu zein indizedun oinezkoak dira antisocial
hautatuak=random.sample(range(numberpeople), numberobstacle)
for index, value in enumerate(noranzkoa):
    if (index in hautatuak):
        noranzkoa[index] = 2
    else:
        if (index<=numberleft):
            noranzkoa[index] = 0
        else:
            noranzkoa[index] = 1

##### 4 ##### personal
persphere = np.zeros((numberpeople))
lista = np.linspace(0.25, 0.35, 10)
persphere = [np.random.choice(lista) for i in np.arange(numberpeople)]

##### 5 ##### hasierakovx
hvx=np.zeros((numberpeople))

for index, value in enumerate(hvx):
    if (index<=numberleft):
        hvx[index] = 1.0
    else:
        hvx[index] = -1.0

##### 6 ##### amaierakovx
avx=np.zeros((numberpeople))

hautatuak=np.zeros((numberpeople))
hautatuak=random.sample(range(numberpeople), numberstress)
for index, value in enumerate(avx):
    if (index<=numberleft):
        if (index in hautatuak):
            avx[index] = 1.5*np.random.choice(np.arange(0.9,1.2,0.05))
        else:
            avx[index] = 1.5*np.random.choice(np.arange(0.9,1.2,0.05))
    else:
        if (index in hautatuak):
            avx[index] = -1.5*np.random.choice(np.arange(0.9,1.2,0.05))

```

```

        else:
            avx[index] = -1.5*np.random.choice(np.arange(0.9,1.2,0.05))

##### 7 ##### hasierakovy
hvy=np.zeros((numberpeople))

##### 8 ##### amaierakovy
avy=np.zeros((numberpeople))

index = 0
for i,j in zip(y0, y00):
    i[:] = x[index], hvx[index], y[index], hvy[index],
        noranzkoa[index], persphere[index], avx[index], avy[index]
    j[:] = index, x[index], hvx[index], y[index], hvy[index],
        noranzkoa[index], persphere[index], avx[index], avy[index]
    index += 1
    filename = path + '/hasierako_egoera.csv'
np.savetxt(filename, y00, delimiter=',', fmt='%8.3f')
return y0

```

### A.2.3 Oztopoen eta oinezkoen distantzia minimoa

Oztopoak zirkuluen eta errektangeluen bitartez daude parametrizatuta. Hurrengo funtzio hauek, (x,y) posizioan kokatuta dagoen oinezko baten eta zirkulu edo errektangelu baterako distantzia minimoa kalkulatzeko du. Horrez gainera, distantzia minimoa ematen duen zirkuluaren edo errektangeluaren puntua bueltatzen du.

```

def pointCircleDistance(x,y,x1,y1,r):
    v=np.array([x1-x,y1-y])
    d=np.sqrt(v@v)-r
    xzir=x+((v/np.sqrt(v@v))*d)@np.array([1,0])
    yzir=y+((v/np.sqrt(v@v))*d)@np.array([0,1])
    return d, xzir, yzir

def pointRectangleDistance(x, y, x1, y1, x2, y2):
    if (x < x1) :
        dx = x1 - x;
        if (y < y1):
            dy = y1 - y;
            return np.sqrt(dx * dx + dy * dy),x1,y1;
        elif (y > y2) :
            dy = y - y2;
            return np.sqrt(dx * dx + dy * dy),x1,y2;
        else :
            return dx,x1,y;

    elif (x > x2) :
        dx = x - x2;
        if (y < y1) :
            dy = y1 - y;
            return np.sqrt(dx * dx + dy * dy),x2,y1;
        elif (y > y2) :
            dy = y - y2;
            return np.sqrt(dx * dx + dy * dy),x2,y2;
        else :
            return dx,x2,y;

    else :
        if (y < y1) :
            return y1 - y,x,y1;

        elif (y > y2) :
            return y - y2,x,y2;

        else :
            return 0.0,x,y;

```

## A.2.4 SFM funtzioa

Funtzio honek oinezko bakoitzean eragindako indarra kalkulatu du. Aldiunea, oinezko guztien egoera bektorea, gelaren geometria deskribatzen duen lista, dwall hormarekiko distantzia parametroa eta hildakoak egonogo diren aldiunea jasotzen ditu parametro moduan. 4.1 ekuazioak emandako  $\vec{y}$  bueltatzen du.

```
def socialforce(t, y, kop, geometria, dwall, konthil):
    with_horma = True
    if geometria[3] == 0.0:
        with_horma = False

    def hormara(horma, y_i0, y_i1, y_i2, y_i3, v0, alfa, x_target):

        minimo=100
        for k in horma:
            d = abs(k - y_i2)
            if d < minimo:
                minimo = d
                retorna=k

        desx=(x_target-y_i0)
        desy=(retorna-y_i2)
        vec=np.array([desx, desy])
        desx=((vec/np.sqrt(vec@vec))*v0)@np.array([1,0])
        desy=((vec/np.sqrt(vec@vec))*v0)@np.array([0,1])
        abiarazleax=(desx-y_i1)/alfa
        abiarazleay=(desy-y_i3)/alfa

        return abiarazleax, abiarazleay

    def atera(y_i0, y_i1, y_i2, y_i3, v0, alfa, x_target):

        desx=(x_target-y_i0)
        desy=0
        vec=np.array([desx, desy])
        desx=((vec/np.sqrt(vec@vec))*v0)@np.array([1,0])
        desy=((vec/np.sqrt(vec@vec))*v0)@np.array([0,1])
        abiarazleax=(desx-y_i1)/alfa
        abiarazleay=(desy-y_i3)/alfa

        return abiarazleax, abiarazleay

#PARAMETROAK:
#Abiarazlea:
alfa=0.5 #erlajazio denbora
#Beste oinezkoen eraginez alderatzailea:
r0=0.5
A=17.22
B=0.08
kb=14.13 #bodycompression
ks=8.8 #sliding friction
h=geometria

#Deribatuaq gordetzeko bektorea
F=np.empty(kop*8)

#HILDAKOAK
noiz=100
zein=[17,23,40]
if (noiz==konthil):
    for d in zein:
        y[4+8*d]==2

#Kontadorea
j=-1
#i. oinezkoak pairatzen dituen indarrak kalkulatzeko:
for i in range(0,8*kop,8):
    if (y[i+4]==2: #oztopo bat baldin bada
        F[i:i+8]=[0,0,0,0,0,0,0,0]
    else:
        j=j+1
```

```

#Balio desberdinak hasieratu behar ditugu
repulsivex=0
repulsivey=0
bodycompressionx=0
bodycompressiony=0
frictionx=0
frictiony=0
anisotrop=0
landa=0.1 #anisotropia aztertzeko: (0,1) balioen artean har ditzake

#ABIARAZLEA:

atea= np.arange( geometria[-1]+0.35, geometria[-7]-0.35, 0.005)

v0=np.sqrt(y[i+6]**2+y[i+7]**2)

zutab=geometria[: -16]
hutsuneak=list()
if with_horma==False and geometria[2]!=0:

    behe=np.arange(0.0+dwall, zutab[1]-zutab[2]-0.35,0.1)
    goi=np.arange(zutab[-3]+zutab[-2]+0.35,15.0-dwall,0.1)
    for s in range(len(behe)):
        hutsuneak.append(behe[s])

    for w in range(0, len(zutab)-4,4):
        l=np.arange(zutab[w+1]+zutab[w+2]+0.35, zutab[w+5]-zutab[w+6]-0.35,0.1)
        for f in range(len(l)):
            hutsuneak.append(l[f])

    for c in range(len(goi)):
        hutsuneak.append(goi[c])

elif with_horma==True:
    behe=np.arange(0.0+dwall, zutab[1]-0.35,0.1)
    goi=np.arange(zutab[-1]+0.35,15.0-dwall,0.1)
    for s in range(len(behe)):
        hutsuneak.append(behe[s])

    for w in range(0, len(zutab)-4,4):
        l=np.arange(zutab[w+3]+0.35, zutab[w+5]-0.35,0.1)
        for f in range(len(l)):
            hutsuneak.append(l[f])

    for c in range(len(goi)):
        hutsuneak.append(goi[c])

if(y[i]<20.0): #atea oraindik zeharkatu ez badu

    if with_horma:
        if (y[i]>geometria[2]):
            abiarazleax, abiarazleay =
            hormara(atea, y[i], y[i+1], y[i+2], y[i+3], v0, alfa, 20.0)
        else:
            if y[i]<geometria[0]:
                abiarazleax, abiarazleay =
                hormara(hutsuneak, y[i], y[i+1], y[i+2], y[i+3], v0, alfa, geometria[0])
            else:
                abiarazleax, abiarazleay =
                hormara(hutsuneak, y[i], y[i+1], y[i+2], y[i+3], v0, alfa, geometria[2])

    else:
        if (geometria[2]==0): #oztoporik gabeko kasua
            abiarazleax, abiarazleay =
            hormara(atea, y[i], y[i+1], y[i+2], y[i+3], v0, alfa, 20.0)
        else:
            if (y[i]>geometria[0]):
                abiarazleax, abiarazleay =
                hormara(atea, y[i], y[i+1], y[i+2], y[i+3], v0, alfa, 20.0)
            else:

```

```

        abiarazleax , abiarazleay =
        hormara(hutsuneak ,y [ i ] ,y [ i +1] ,y [ i +2] ,y [ i +3] ,v0 , alfa , zutab [0])

else: #atea jadatau badu
    abiarazleax , abiarazleay = atera(y [ i ] ,y [ i +1] ,y [ i +2] ,y [ i +3] ,v0 , alfa ,40.0)

    #OBSTACLE INDARRA (x,y) center rectangle

for w in range(0 , len (h) ,4):
    if (h[w+2]==0.0 and h[w+2]<h[w]):
        repulsivex = 0.0
        repulsivey = 0.0
    else:
        if (h[w+3]>h[w+1]):
            peddistance , obstaclex , obstacley =
            pointRectangleDistance (y [ i ] , y [ i +2] , h [w] , h [w+1] , h [w+2] , h [w+3])
        else:
            peddistance , obstaclex , obstacley =
            pointCircleDistance (y [ i ] , y [ i +2] ,h [w] ,h [w+1] ,h [w+2])

vec=np. array ([y [ i ]-obstaclex , y [ i +2]-obstacley ])
vec=vec/peddistance

robstacle=y [ i +5]
if (robstacle>peddistance): #evacuation kasuetarako
    bodycompressionx=
    bodycompressionx+(robstacle-peddistance)*kb*(vec@np. array ([1 ,0]))
    bodycompressiony=
    bodycompressiony+(robstacle-peddistance)*kb*(vec@np. array ([0 ,1]))

    vecperp=np. array ([ obstacley -y [ i +2] ,y [ i ]-obstaclex ])
    vecperp=vecperp/np. sqrt (np. dot (vecperp , vecperp))
    abiadif=[-y [ i +1] , -y [ i +3]]
    frictionx=frictionx+
    (robstacle-peddistance)*ks*((vecperp@abiadif)*vecperp)@np. array ([1 ,0])
    frictiony=frictiony+
    (robstacle-peddistance)*ks*((vecperp@abiadif)*vecperp)@np. array ([0 ,1])

    repulsivex = repulsivex +
    A*np. exp ((robstacle-peddistance)/B)*(vec@np. array ([1 ,0]))
    repulsivey = repulsivey +
    A*np. exp ((robstacle-peddistance)/B)*(vec@np. array ([0 ,1]))

for k in range(0 , kop*8 ,8):
    if (k!=i):
        vec=np. array ([y [ i ]-y [k] , y [ i +2]-y [k+2]])
        peddistance = np. sqrt (vec@vec)
        vec=vec/peddistance

        if (peddistance < 2): #bestela indarrak nuluak

            v=np. array ([y [ i +1] ,y [ i +3]])
            v=v/np. sqrt (v@v)
            a=-np. array ([y [ i ]-y [k] , y [ i +2]-y [k+2]])/peddistance
            z=np. cos (v@a)

            anisotrop = landa + (1-landa)*(1+z)*0.5

            r0=y [ i +5]+y [k+5]
            if (r0>peddistance): #evacuation kasuetarako
                bodycompressionx=bodycompressionx+
                (r0-peddistance)*kb*(vec@np. array ([1 ,0]))
                bodycompressiony=bodycompressiony+
                (r0-peddistance)*kb*(vec@np. array ([0 ,1]))

                vecperp=np. array ([y [k+2]-y [ i +2] ,y [ i ]-y [k]])
                vecperp=vecperp/np. sqrt (np. dot (vecperp , vecperp))
                abiadif=[y [k+1]-y [ i +1] ,y [k+3]-y [ i +3]]
                frictionx=frictionx+
                (r0-peddistance)*ks*((vecperp@abiadif)*vecperp)@np. array ([1 ,0])

```

```

frictiony=frictiony+
(r0-peddistance)*ks*((vecperp@abiadif)*vecperp)@np.array([0,1])

repulsivex = repulsivex +
anisotrop*A*np.exp((r0-peddistance)/B)*(vec@np.array([1,0]))
repulsivey = repulsivey +
anisotrop*A*np.exp((r0-peddistance)/B)*(vec@np.array([0,1]))

totalx=abiarazleax+repulsivex+bodycompressionx+frictionx
totaly=abiarazleay+repulsivey+bodycompressiony+frictiony

#Abiaduraren balioa mugatuta dago:
v0=np.sqrt(y[i+6]**2+y[i+7]**2)
if (np.sqrt(y[i+1]**2+y[i+3]**2)>v0):
    vec=np.array([y[i+1],y[i+3]])
    y[i+1]=((vec/np.sqrt(vec@vec))*v0)@np.array([1,0])
    y[i+3]=((vec/np.sqrt(vec@vec))*v0)@np.array([0,1])

F[i:i+8]=[y[i+1],totalx,y[i+3],totaly,0,0,0,0]

return F

```

## A.2.5 Runge-Kutta 4

Runge-Kutta 4 zenbakizko metodoa implementatzen duen funtzioa. Metodo honek hurrengo parametroak hartzen ditu argumentu moduan: f funtzioa, h iterazio bakoitzeko urrats-luzera, y egoera bektorea, t aldiunea, oinezkoen kopurua, oztopoen kokapena gordetzen duen lista, dwall parametroa eta konthil hildakoak agertzea nahi dugun aldiunea. Funtzioak egoera aldagai berria eta aldiunea bueltatzen ditu.

```

def rk4(f, h, y, t, numberpeople, geometria, dwall, konthil):

    emaitza=f(t, y, numberpeople, geometria, dwall, konthil)

    k1 = f(t, y, numberpeople, geometria, dwall)
    k2 = f(t + h / 2.0, y + h / 2.0 * k1, numberpeople, geometria, dwall, konthil)
    k3 = f(t + h / 2.0, y + h / 2.0 * k2, numberpeople, geometria, dwall, konthil)
    k4 = f(t + h, y + h * k3, numberpeople, geometria, dwall, konthil)

    t=t+h
    y= y + h * (k1 + 2.0 * k2 + 2.0 * k3 + k4) / 6.0

    return t,y

```

## A.2.6 Eboluzioa

Programa honek zehaztutako denbora tartean hasierako egoeraren oinezko guztien eboluzioa kalkulatzen du. Amaierako aldiunea, pausu kopurua, hasierako egoera, gelaren eta oztopoen geometria jasotzen ditu parametro moduan.

```

def eboluzioa(tN, N, y0, number, leftx, rightx, bottony, upy, geometriak, tituluak,
            path, dwall, option=1, numberobstacle=0):

    y00 = y0.flatten()
    y1 = y0.flatten()
    y2 = y0.flatten()

    t0 = 0 #hasierako aldiunea
    tN = tN #amaierako aldiunea

    N = N #####
    h = (tN - t0) / N #pausua

    spacebis=0 #Biskositate espaziala gordetzeko aldagai
    kont = 0

```

```

kont1 = 0
kont2 = 0
kont1 = np.zeros([N])
kont11 = np.zeros([N])
kont12 = np.zeros([N])
denbora = np.zeros([N])

tituluak = tituluak

kont3 = 0.0 ##### kontadorea, hiru kasuetan ibitari guztiak ateratakoak,
                #beste bost ziklo egiten da BREAKa egin baino lehen

def figure_tokia(figure, axes, left_x_lim, right_x_lim, left_y_lim, right_y_lim,
                 geometriak, tituluak):

    def figure_tokikoak(figure, axis, geometria):

        axis.add_patch(geometria);

    plt.rc('text', usetex=True)

    r=0
    for axis in axes:
        geo_berezia=geometriak[r][: -16]
        #geo_berezia=geometriak[r][:]

        for w in range(0, len(geo_berezia), 4):
            if (geo_berezia[w+3]!=0.00):
                figure_tokikoak(fig, axis, Rectangle((geo_berezia[w:w+2]),
                geo_berezia[w+2]-geo_berezia[w], geo_berezia[w+3]-geo_berezia[w+1]));
            else:
                figure_tokikoak(fig, axis, Circle((geo_berezia[w:w+2]),geo_berezia[w+2]));
            r=r+1

    for index, i in enumerate(axes):
        i.set_xlim(left_x_lim, right_x_lim);
        i.set_ylim(left_y_lim, right_y_lim);
        i.set_title(tituluak[index]);
        i.yaxis.set_major_locator(ticker.NullLocator());
        i.xaxis.set_major_locator(ticker.NullLocator());
        i.set_aspect(20./22.5);
        i.autoscale(enable=False, axis='both');
        i.plot([-1.00, -1.00], [-1.00, 16.00], color='k', lw=3);

        i.add_patch(Rectangle((geometriak[index][ -8], geometriak[index][ -7]),
        geometriak[index][ -6]-geometriak[index][ -8],
        geometriak[index][ -5]-geometriak[index][ -7]));

        i.add_patch(Rectangle((geometriak[index][ -4], geometriak[index][ -3]),
        geometriak[index][ -2]-geometriak[index][ -4],
        geometriak[index][ -1]-geometriak[index][ -3]));

        i.add_patch(Rectangle((geometriak[index][ -16], geometriak[index][ -15]),
        geometriak[index][ -14]-geometriak[index][ -16],
        geometriak[index][ -13]-geometriak[index][ -15]));

        i.add_patch(Rectangle((geometriak[index][ -12], geometriak[index][ -11]),
        geometriak[index][ -10]-geometriak[index][ -12],
        geometriak[index][ -9]-geometriak[index][ -11]));

    def figure_datuak(figure, axis, left_x_lim, right_x_lim, left_y_lim, right_y_lim,):

        plt.rc('text', usetex=True)
        tick_left_y_lim, tick_right_y_lim = 0, right_y_lim+round(0.1*right_y_lim)
        tick_scale_y = (tick_right_y_lim - tick_left_y_lim)/20.
        ticks_y = [i*20 for i in np.arange(0, tick_scale_y, 1)]

        tick_left_x_lim, tick_right_x_lim = 0, right_x_lim+round(0.2*right_x_lim)
        tick_scale_x = (tick_right_x_lim - tick_left_x_lim)/10.
        ticks_x = [i*10 for i in np.arange(0, tick_scale_x, 1)]

```

```

axis.set_title(r'\textm{Irtendako_oinezkoen_kopurua}');
axis.set_xlabel(r'\textm{Denbora_(s)}');
axis.set_ylabel(r'\textm{Oinezkoen_kopurua}');
axis.set_xlim(left_x_lim, right_x_lim); #x ardatza definitu
axis.set_ylim(left_y_lim, right_y_lim+round(0.1*right_y_lim)); #y ardatza definitu

colors = list(mcolors.TABLEAU_COLORS.values())

y00_ = np.zeros((numberpeople*8, N))
y1_ = np.zeros((numberpeople*8, N))
y2_ = np.zeros((numberpeople*8, N))

if number == numberpeople:
    number = ma.ceil(numberpeople*0.10) # ibiltari denen %10 aukeratzten da segitzeko
    ibiltari_segitu = sorted(rd.sample(list(np.arange(numberpeople)), number,))
else:
    ibiltari_segitu = sorted(rd.sample(list(np.arange(numberpeople)), number,))

np.savetxt(path+'selected.csv', ibiltari_segitu, delimiter=',', fmt='%8.3f')

for i in range(N):
    time = datetime.now()
    t, y00, bisk = rk4(socialforce, h, y00, t0, numberpeople, geometriak[0],dwall,i)
    y00[:,i] = y00
    t, y1, bisk = rk4(socialforce, h, y1, t0, numberpeople, geometriak[1],dwall,i)
    y1[:,i] = y1
    t, y2, bisk = rk4(socialforce, h, y2, t0, numberpeople, geometriak[2],dwall,i)
    y2[:,i] = y2
    denbora[i] = i*h

    time = datetime.now()

    fig, axes = plt.subplots(nrows=2, ncols=2, figsize=(8,6), );
    ax0, ax1, ax2, ax3 = axes.flat

    three_axes = [axis for axis in axes.flatten()[:-1]]

    for j in range(0,numberpeople*8,8):
        if j in [8*i for i in ibiltari_segitu]:
            ax0.plot(y00_[j,:i],y00_[j+2,:i], lw=2, alpha=0.5)#, color=colors[])
            ax1.plot(y1_[j,:i], y1_[j+2,:i], lw=2, alpha=0.5)#, color=colors[])
            ax2.plot(y2_[j,:i], y2_[j+2,:i], lw=2, alpha=0.5)#, color=colors[])

            ax0.add_patch(Circle((y00[j], y00[j+2]), y00[j+5], edgecolor='k',
                                facecolor='r', alpha=1.0, linewidth=0.50))

            ax1.add_patch(Circle((y1[j], y1[j+2]), y1[j+5], edgecolor='k',
                                facecolor='r', alpha=1.0, linewidth=0.50))

            ax2.add_patch(Circle((y2[j], y2[j+2]), y2[j+5], edgecolor='k',
                                facecolor='r', alpha=1.0, linewidth=0.50))
        else:
            three_axes[0].add_patch(Circle((y00[j], y00[j+2]), y00[j+5], edgecolor='k',
                                            facecolor='r', alpha=0.3, linewidth=0.50))
            three_axes[1].add_patch(Circle((y1[j], y1[j+2]), y1[j+5], edgecolor='k',
                                            facecolor='r', alpha=0.3, linewidth=0.50))
            three_axes[2].add_patch(Circle((y2[j], y2[j+2]), y2[j+5], edgecolor='k',
                                            facecolor='r', alpha=0.3, linewidth=0.50))
        if (y00[j]>20.1):
            kont=kont+1
        if (y1[j]>20.1):
            kont1=kont1+1
        if (y2[j]>20.1):
            kont2=kont2+1

    kontl [i] = kont
    kontl1 [i] = kont1
    kontl2 [i] = kont2

figure_tokia(fig, three_axes, leftx, rightx, bottony, upy, geometriak, tituluak)

```



```

figure_datuak (fig, ax3, 0, tN, 0, numberpeople, );

ax3.plot(denbora[:i], kont1[:i], label=r'\textrm{(a)}');
ax3.plot(denbora[:i], kont11[:i], label=r'\textrm{(b)}');
ax3.plot(denbora[:i], kont12[:i], label=r'\textrm{(c)}');
#ax3.legend(loc='lower right');

ax3.legend(bbox_transform=ax3.transData, bbox_to_anchor=(tN*0.85, numberpeople*0.40),
ncol=1, borderaxespad=0, frameon=False, fontsize=10);

text_kont1, text_kont11, text_kont12 = "{:4d}".format(int(kont1[i])),
"{:4d}".format(int(kont11[i])), "{:4d}".format(int(kont12[i]))

ax3.text(tN*0.90, numberpeople*0.33, r'\textrm{%s}' % text_kont1, ha='right')
ax3.text(tN*0.90, numberpeople*0.23, r'\textrm{%s}' % text_kont11, ha='right')
ax3.text(tN*0.90, numberpeople*0.13, r'\textrm{%s}' % text_kont12, ha='right')

if kont==numberpeople and kont1==numberpeople and kont2==numberpeople:
    kont3 += 1.0
    if kont3 == 7.0:
        break

kont, kont1, kont2 = 0, 0, 0

plt.tight_layout();
#filename =
plt.savefig(path + '/images/' + str(i)+'.png', format="png", dpi = 150);
plt.close(fig=fig);

print('****', '{:4d}'.format(i)+' iterazioa', '*****', '{:07.2f}'.format(denbora[i]),
'{:07.2f}'.format(kont1[i]), '{:07.2f}'.format(kont11[i]), '{:07.2f}'.format(kont12[i]),
'*****Time_elpased:', datetime.now() - startTime)

return denbora, kont1, kont11, kont12, y00_, y1_, y2_

```

## A.2.7 Programa nagusia

Larrialdi kasuetako programa nagusia ageri da. Hasieran esperimentuak izango dituen parametroen balioak, izenburuak, datuak gordetzeko konputagailuaren helbidea eta aztertuko diren geometria diferenteak definitzen dira. Gero, abiarazi funtzioari deitzen zaio hasierako egoera lortzeko eta hau definituta dagoelarik, eboluzioa funtzioari dei egiten zaio. Gainera, iterazio bakoitzak zenbateko denbora tardatzen duen ikusteko komandoa ageri da ere. Egoera aldagai guztiak aldiune bakoitzean dituen matrizeak ere gorde egiten dira testu fitxategietan.

```

##### EXPERIMENT #####
number = 1
exp = 'exp_' + str(number) + '_'

delta_t = 0.085

N = 2000
tN = round(N*delta_t)
persons = 196
numberpeople = persons
banaketa = 0
selected = 20
numberleft = persons
numberstress = 0
leftx = -2.5
rightx = 25.00
bottony = -2.5
upy = 17.5
dwall=0.35

name_experiment = exp+str(N)+'_' +str(persons)+'_' +str(selected)+'_' +str(banaketa)+'_'

geometria_h1 = [18.72, 1.51,18.92,
13.49, -1.00,15.00,

```

```

                21.00,16.00, -1.00,-1.00,
                21.00, 0.00,20.00,
                8.00,21.00,15.00,
                20.00,0.00,21.00, 7.00]
geometria_h2 = [18.32, 1.51,18.92, 13.49,
                -1.00,15.00,21.00, 16.00,
                -1.00,-1.00,21.00, 0.00,
                20.00, 8.00,21.00, 15.00,
                20.00,0.00,21.00, 7.00]
geometria_h3 = [17.92, 1.51,18.92,13.49,
                -1.00,15.00,21.00,16.00,
                -1.00,-1.00,21.00, 0.00,
                20.00, 8.00,21.00,15.00,
                20.00,0.00,21.00, 7.00]

tituluak = [r '\textrm{(a)}\_Horma\_x=18.92$, \_d=0.2$',
            r '\textrm{(b)}\_Horma\_x=18.92$, \_d=0.4$',
            r '\textrm{(c)}\_Horma\_x=18.92$, \_d=0.8$']

path = name_experiment[:-1]

##### generating the directories and files to store the data

directories = Path(path + '/images')
directories.mkdir(parents=True, exist_ok=True)

##### Experiment starts

startTime = datetime.now()
print('\nTime_elapsed:_', startTime,2*'\n')

y0 = abiarazi(numberpeople, numberleft, numberstress, leftx, rightx, bottony, upy,
              path, banaketa=banaketa, option=1, numberobstacle=0)

denbora, kontl, kontl1, kontl2, y00_, y1_, y2_ = hasierakoegoera
(tN, N, y0, selected, leftx, rightx, bottony, upy,
 [geometria_h1, geometria_h2, geometria_h3], tituluak, path,dwall, option=0)

print('\nTime_elapsed:_', datetime.now() - startTime)

## Experiment finishes

data = []
for i in zip(denbora, kontl, kontl1, kontl2):
    data.append(i)
    np.savetxt(path+'/data_'+name_experiment[:-1]+'_'.csv', np.array(data),
               delimiter=',', fmt='%8.3f')

np.savetxt(path+'/data_y0_'+name_experiment[:-1]+'_'.csv', np.array(y00_),
           delimiter=',', fmt='%8.3f')

np.savetxt(path+'/data_y1_'+name_experiment[:-1]+'_'.csv', np.array(y1_),
           delimiter=',', fmt='%8.3f')

np.savetxt(path+'/data_y2_'+name_experiment[:-1]+'_'.csv', np.array(y2_),
           delimiter=',', fmt='%8.3f')

```

## B. Eranskina

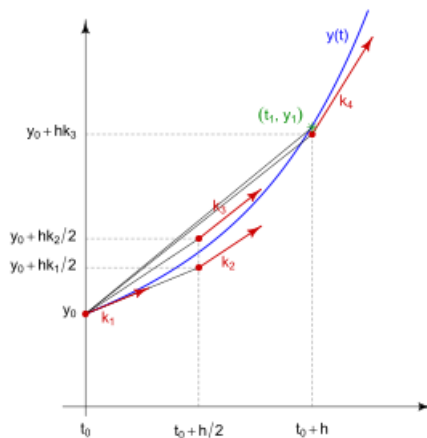
# Runge-Kutta metodoa

Runge-Kutta 4 ekuazio diferentzial arrunten soluzioen hurbilketarako iteraziozko zenbakizko metodoa da, zehazki, hasierako balioko problemaren ebazpenerako. Maiz erabiltzen da deribatu analitikorik behar ez duelako eta Taylor-en garapenean ordena handiko metodoen baliokidea den zehaztasuna lor daitekelako.

Ekuazio diferentzial arrunten problema hurrengo moduan idazten da:

$$y' = f(t, y) \quad y(t_0) = y_0 \quad (\text{B.1})$$

hau da, funtzioaren hasierako balore bat eta funtzioaren eta  $t$  aldagai aske baten menpeko beste funtzio bat  $f(t,y)$  emanik,  $y(t)$  kalkulatu behar da  $t$  tarte baterako.



B.1. Irudia: RK4 metodoan erabiltzen diren malda desberdinak. [20] erreferentziatik hartutako irudia.

$f$  funtzioa  $[t_i, t_{i+1}]$  tartean 4 aldiz ebaluatutako  $f$  funtzioaren balio ezberdinenen batezbesteko bat izango da:

$$y_{i+1} = y_i + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4) \quad (\text{B.2})$$

non

$$k_1 = f(t_i, y_i)$$

$$k_2 = f(t_i + h/2, y_i + hk_1/2)$$

$$k_3 = f(t_i + h/2, y_i + hk_2/2)$$

$$k_4 = f(t_i + h, y_i + hk_3)$$

RK4 4. ordenako metodoa denez, trunkatze errore lokala, hots, iterazio bakoitzean egiten dena,  $O(h^5)$  ordenakoa da eta metatutako errore globala, iterazio askoren ondoriozkoa,  $O(h^4)$  ordenakoa. Hauxe Eulerren metodoan baino askoz hobea da. Eulerren trunkatze errore lokala  $O(h^2)$ -koa da eta metatutakoa  $O(h)$ -koa, lehen ordenakoa baita.

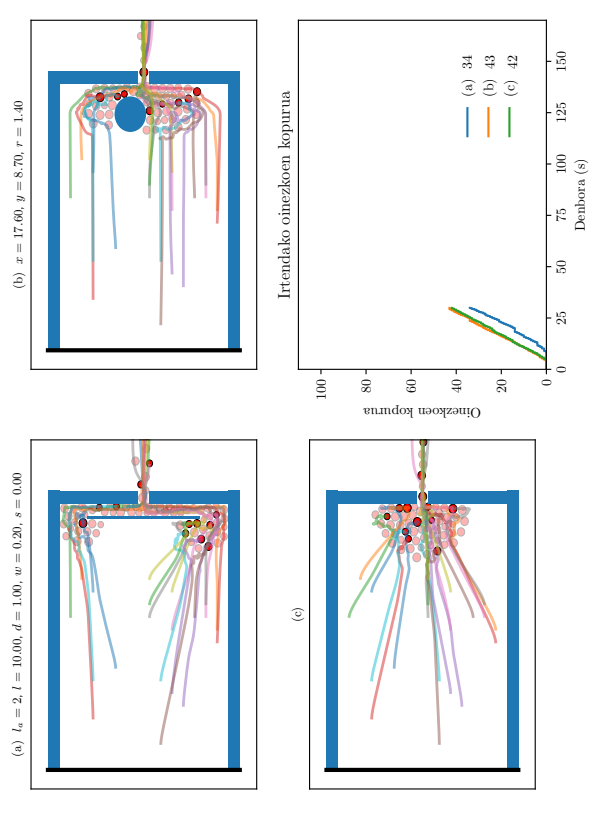
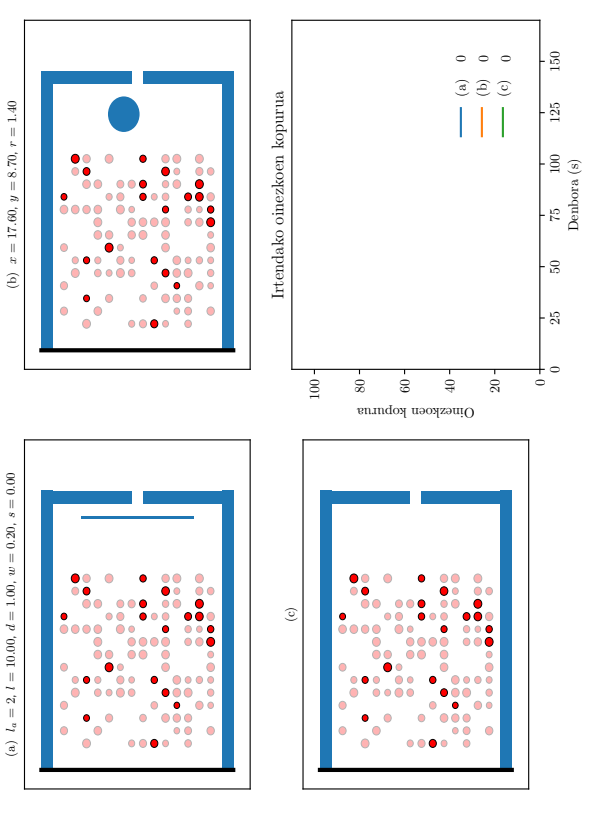
## C. Eranskina

# Esperimentuen irudiak: larrialdi-egoerak

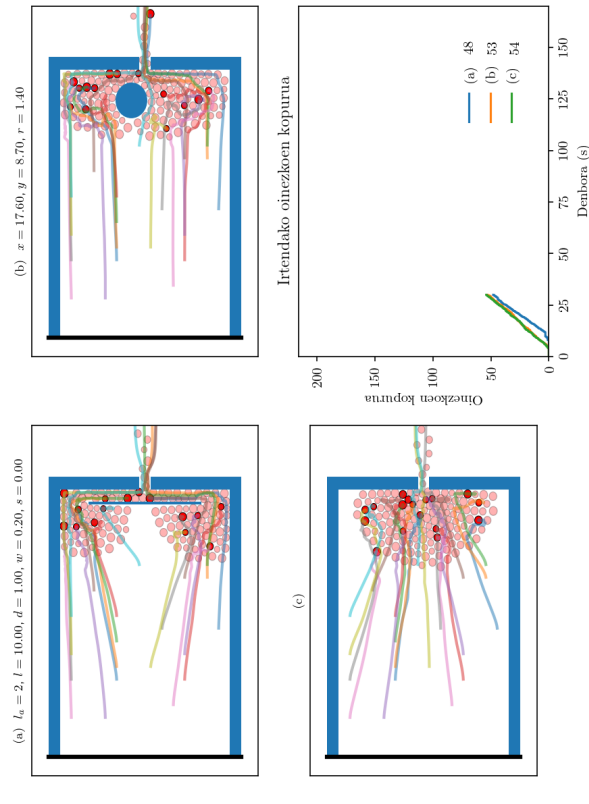
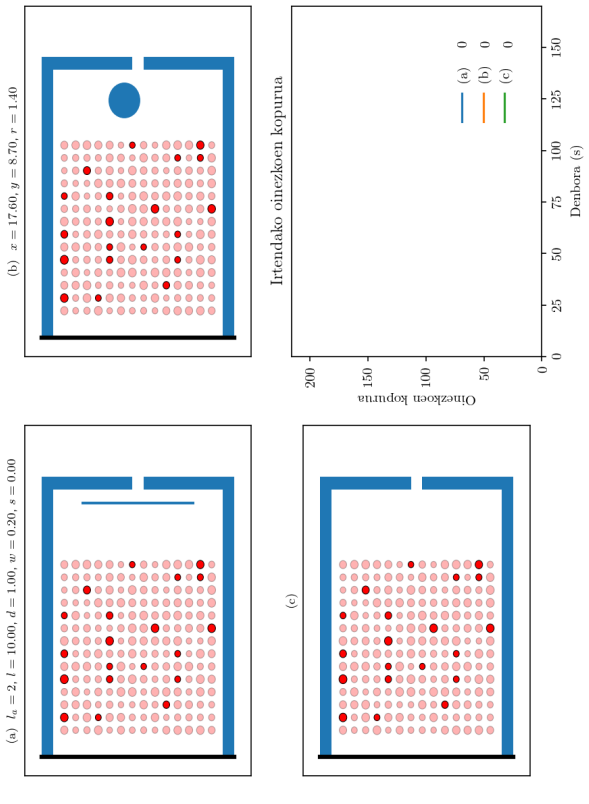
Eranskin honetan gelaren ebakuaziorako egindako esperimentu guztien emaitzak eta tarteko zenbait irudi dago. Esperimentu bakoitza  $N = 196$  eta  $N = 100$  oinezkorekin egin denez, bakoitzarentzat bi zutabe egongo da. Gainera, hasierako, tarteko ( $t = 30s$ ) eta amaierako egoerak adieraziko dira. Esperimentuaren oztopoen kokapen zehatza irudian bertan ageri da.

I0. Esperimntua

N=100

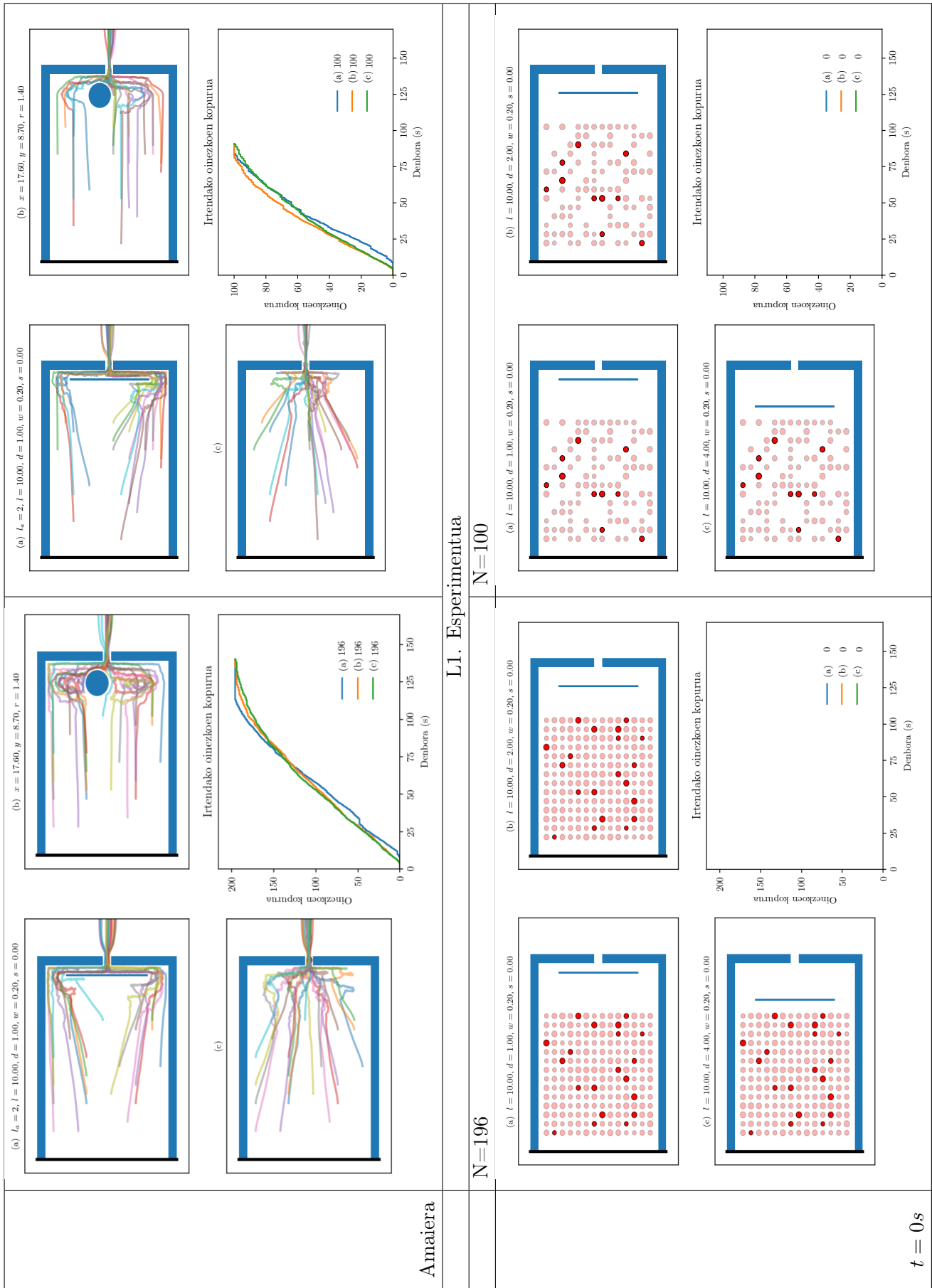


N=196

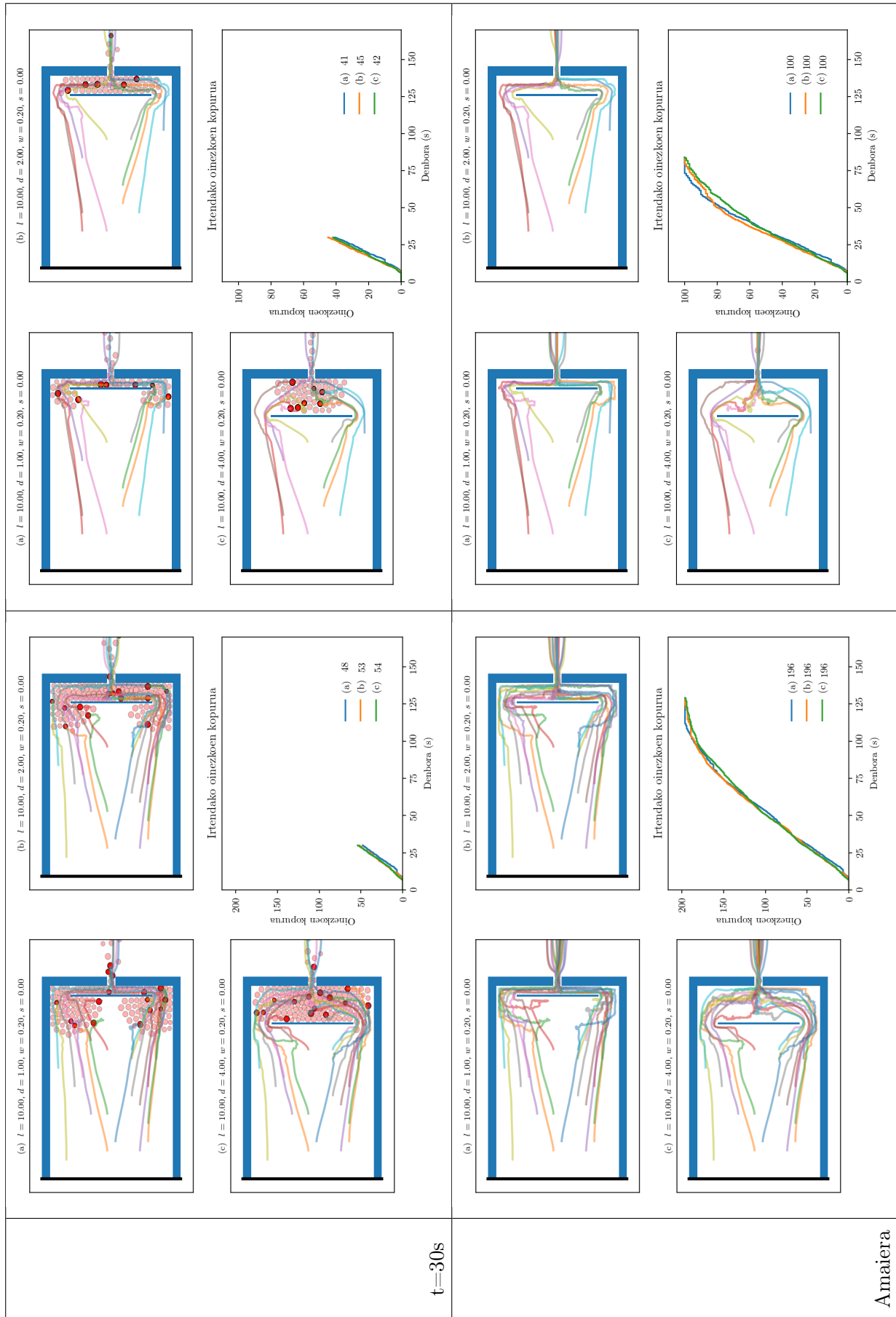


t = 0s

t = 30s



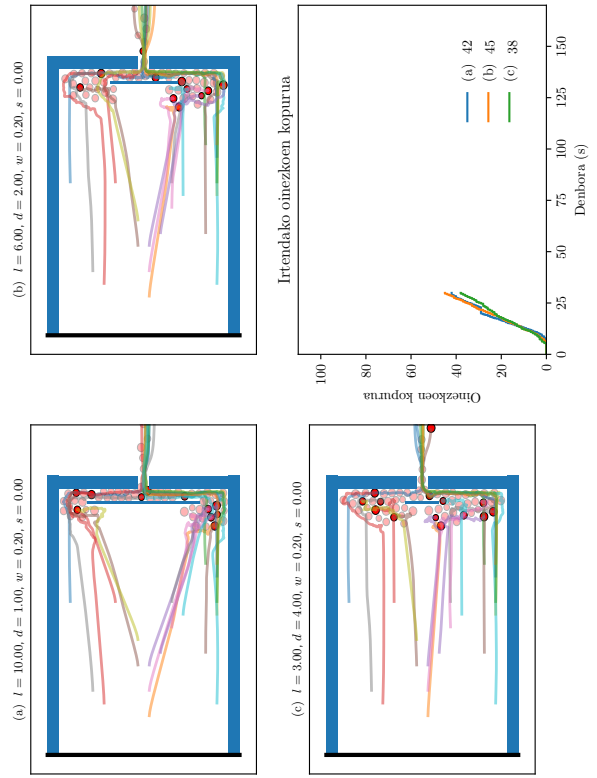
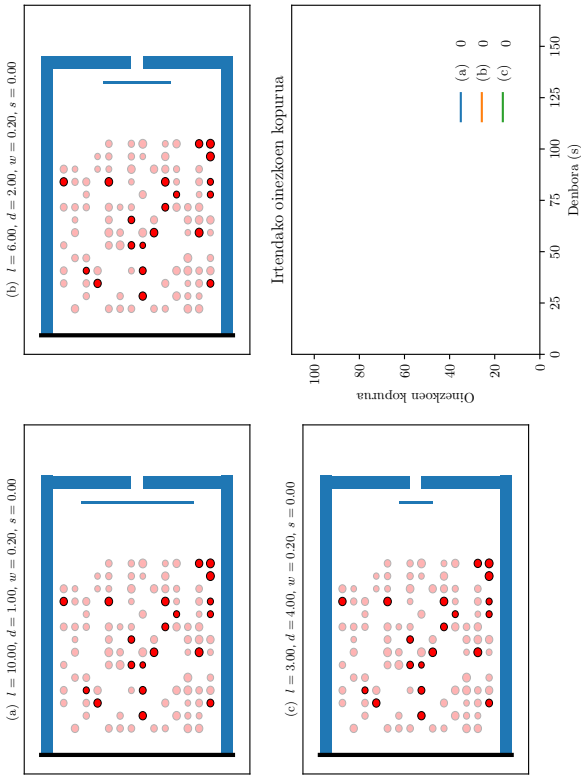
$t = 0s$



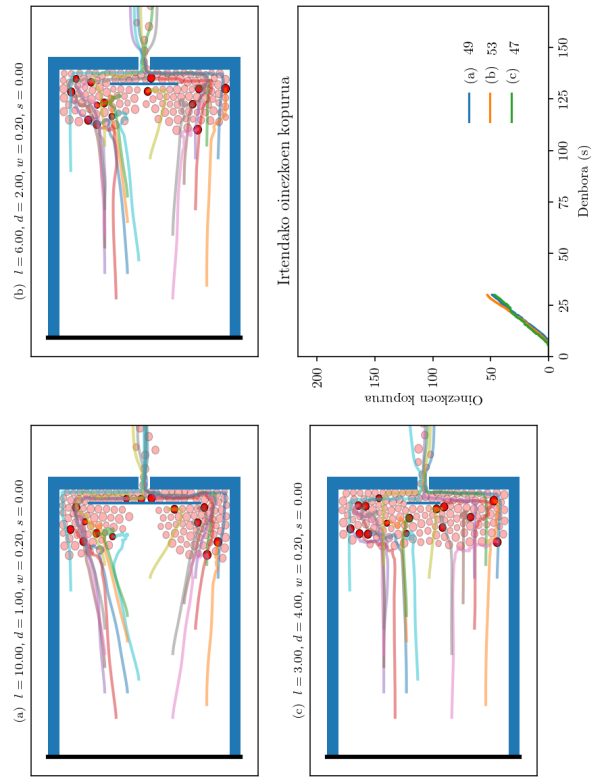
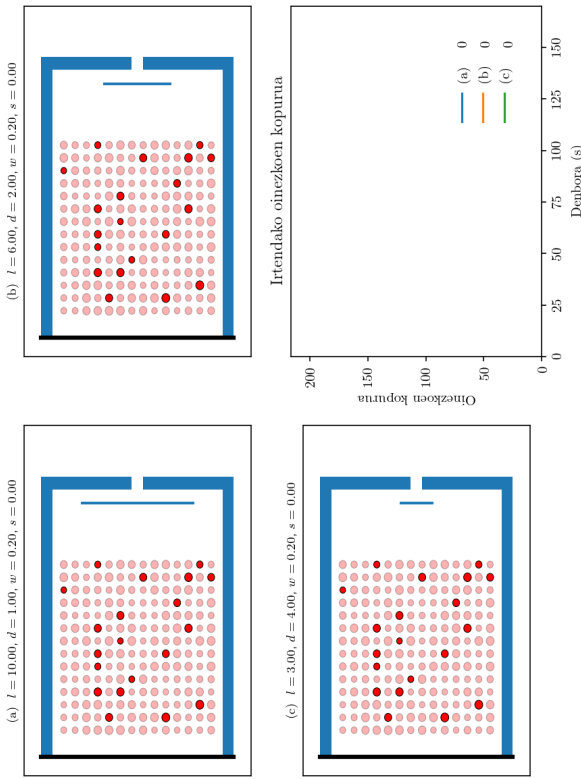


L2. Esperimentua

N=100

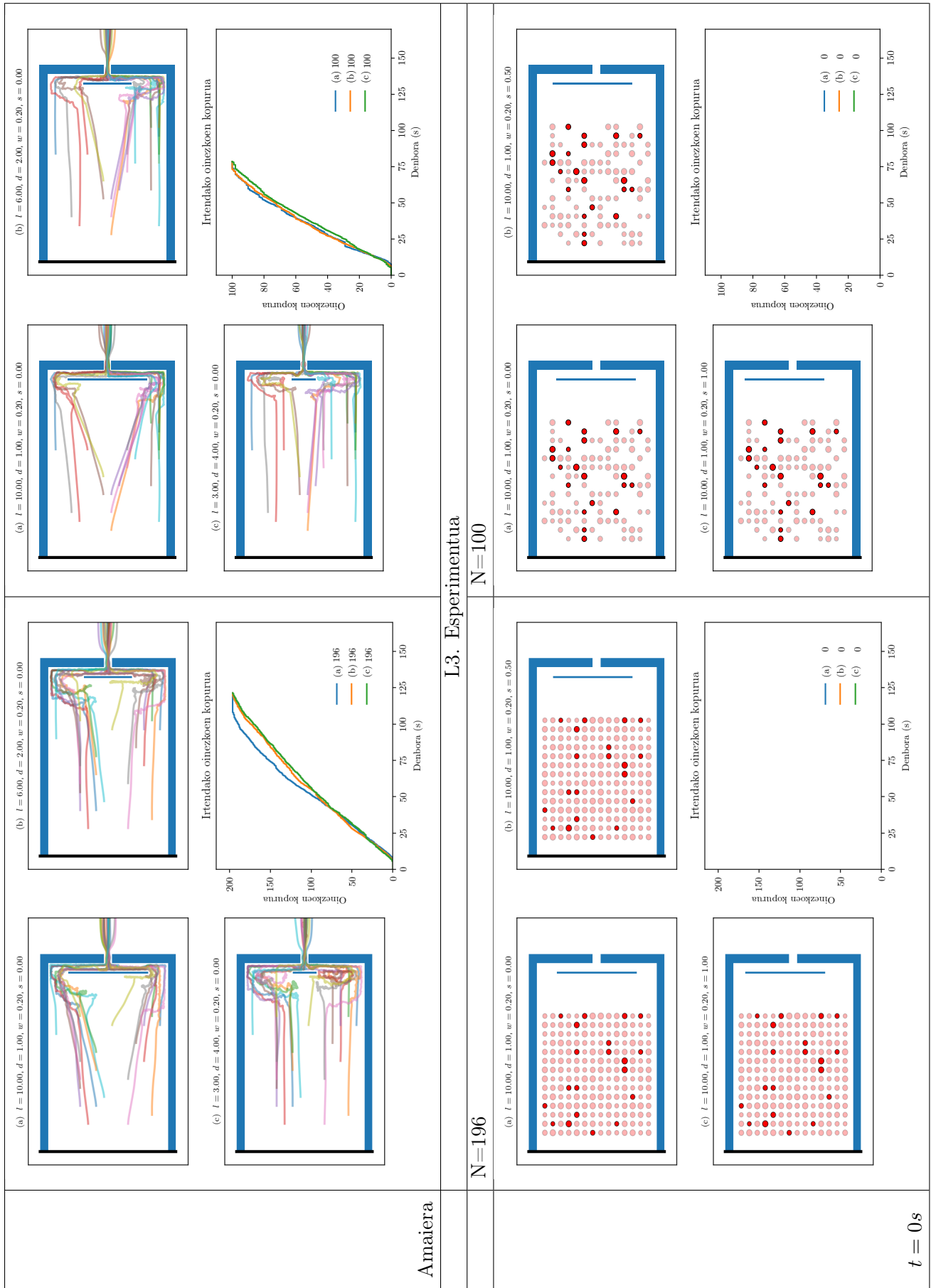


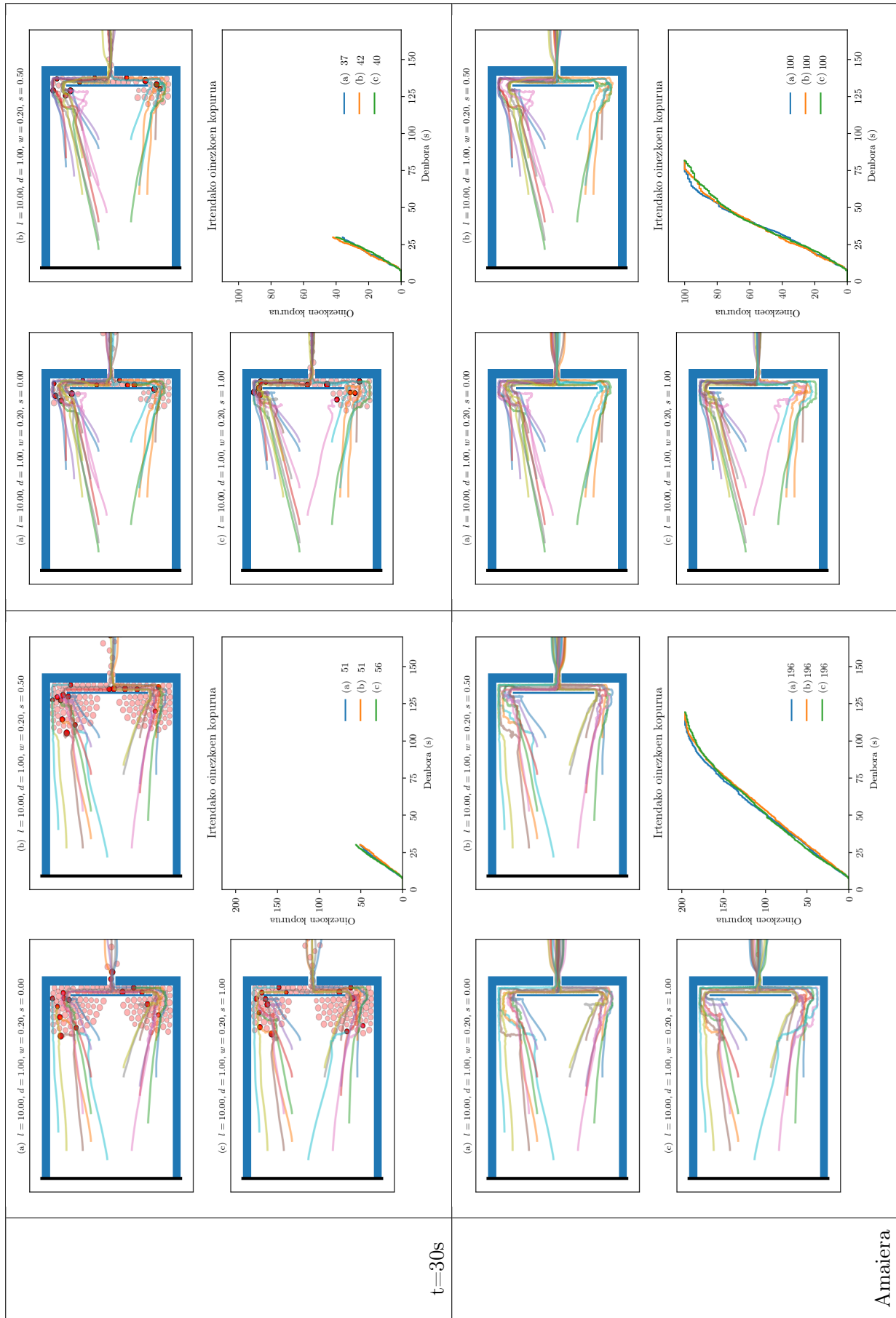
N=196



$t = 0s$

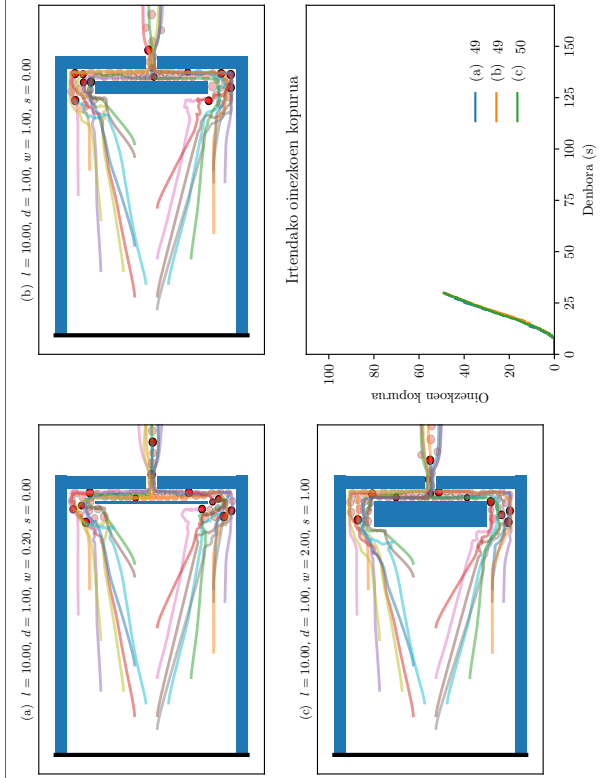
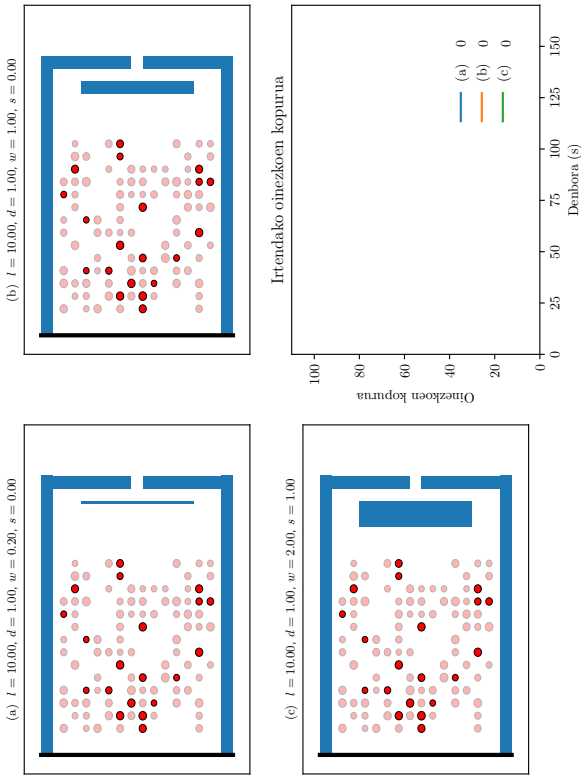
$t = 30s$



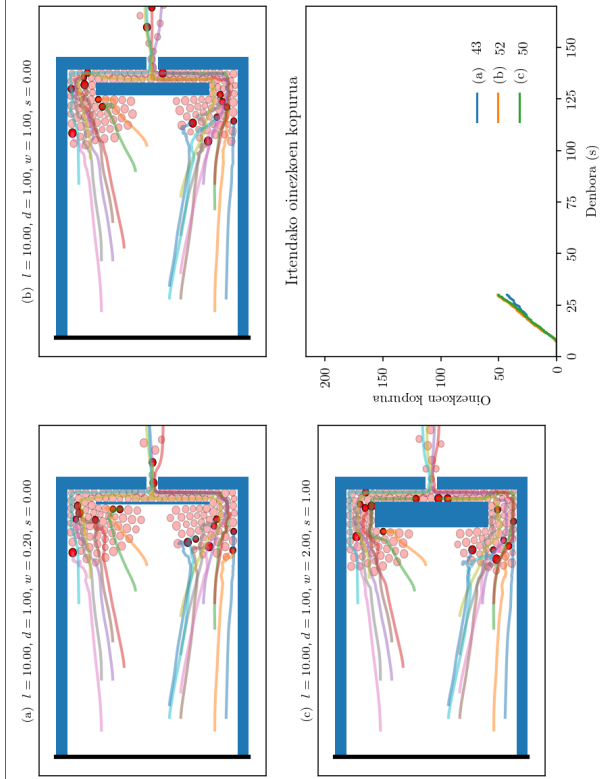
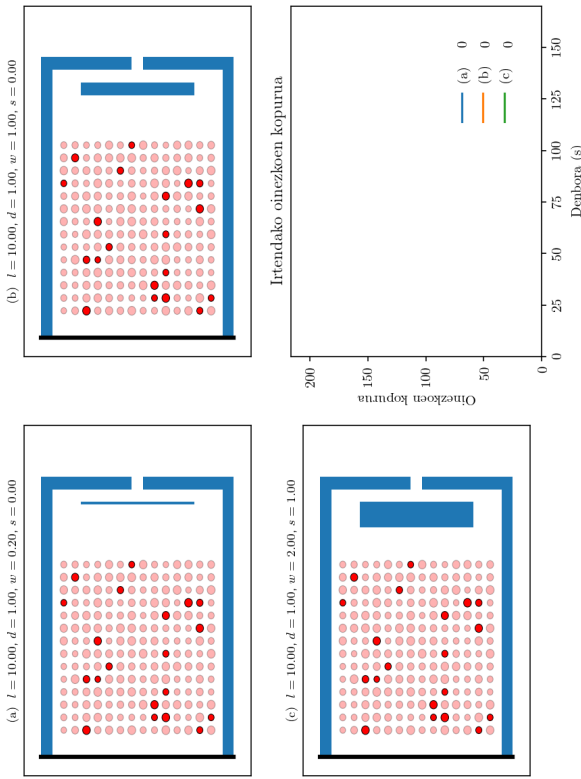


I4. Esperimentua

N=100

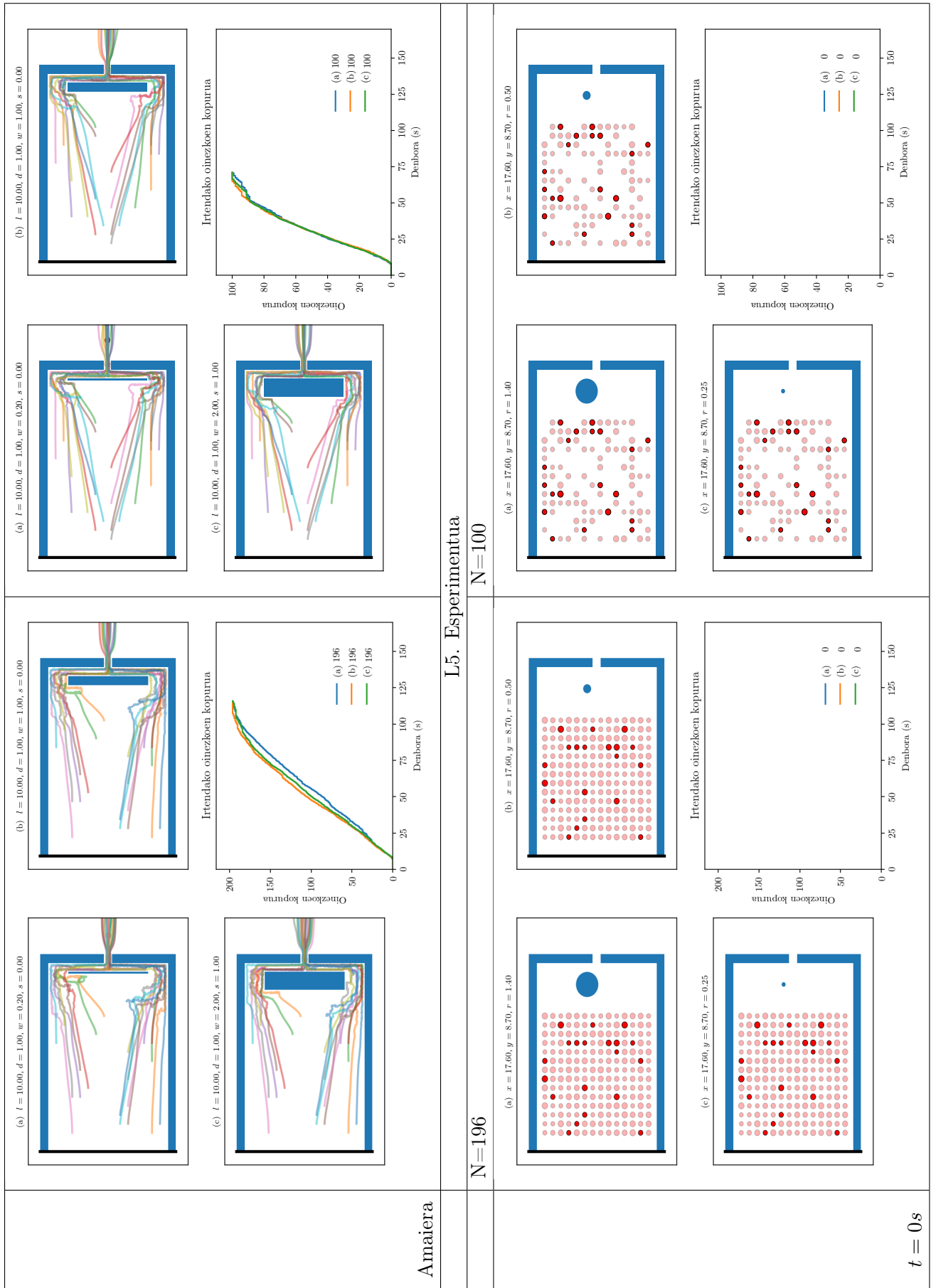


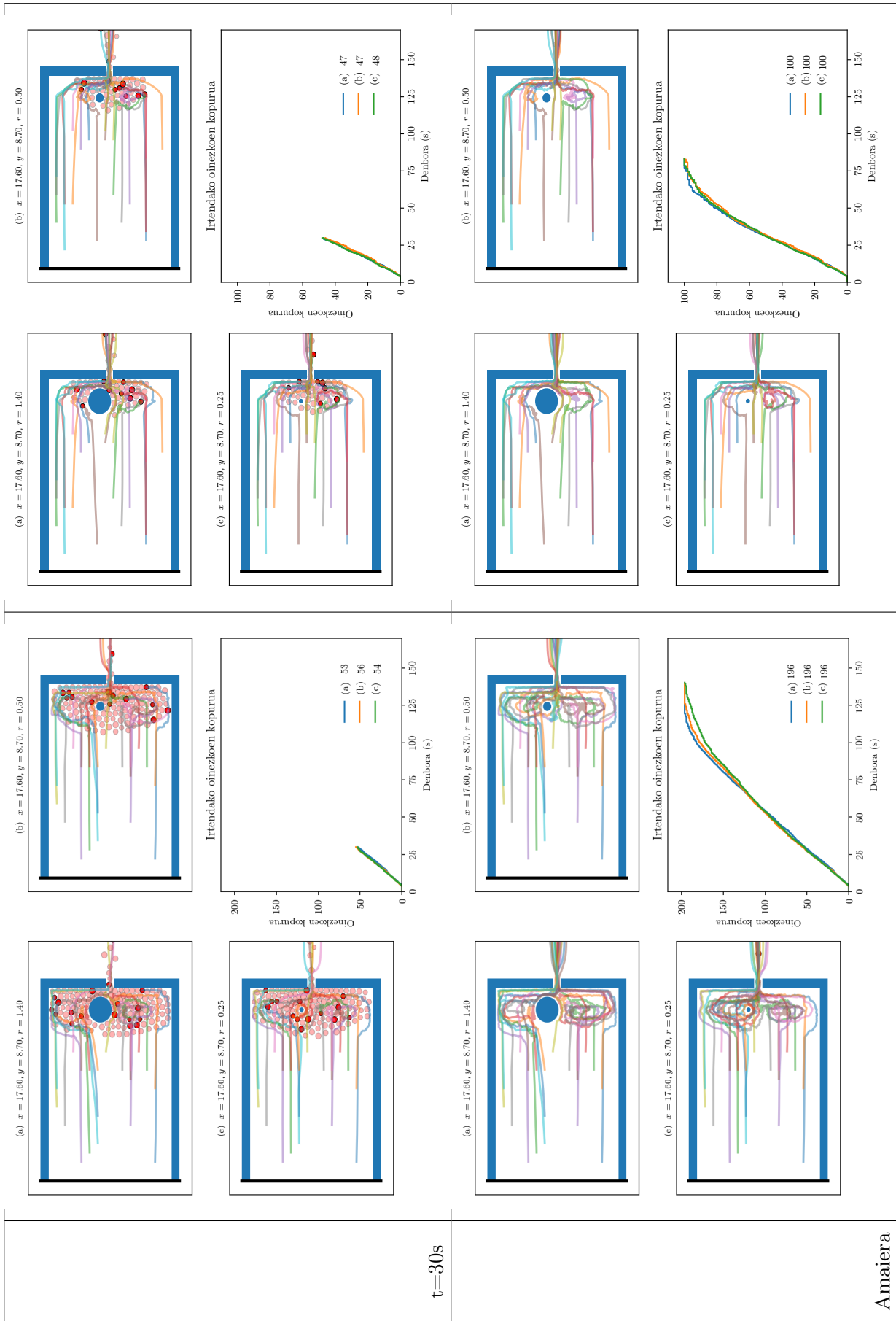
N=196



$t = 0s$

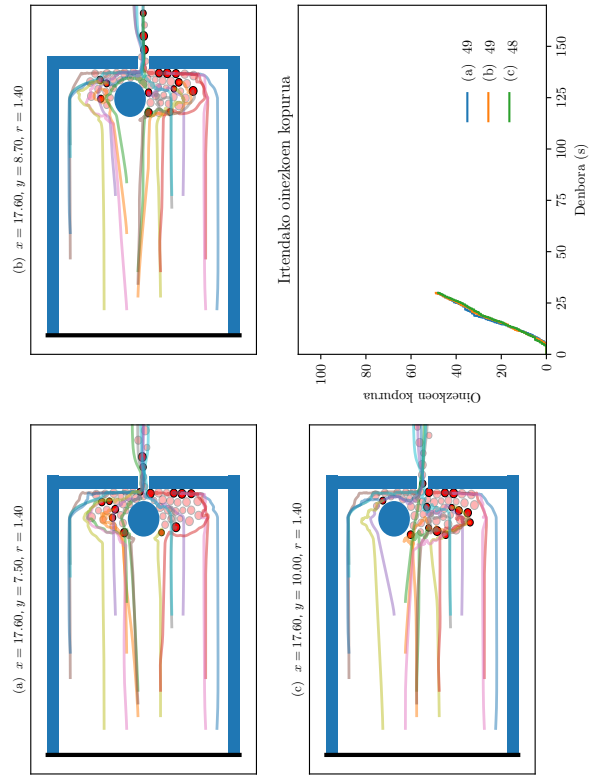
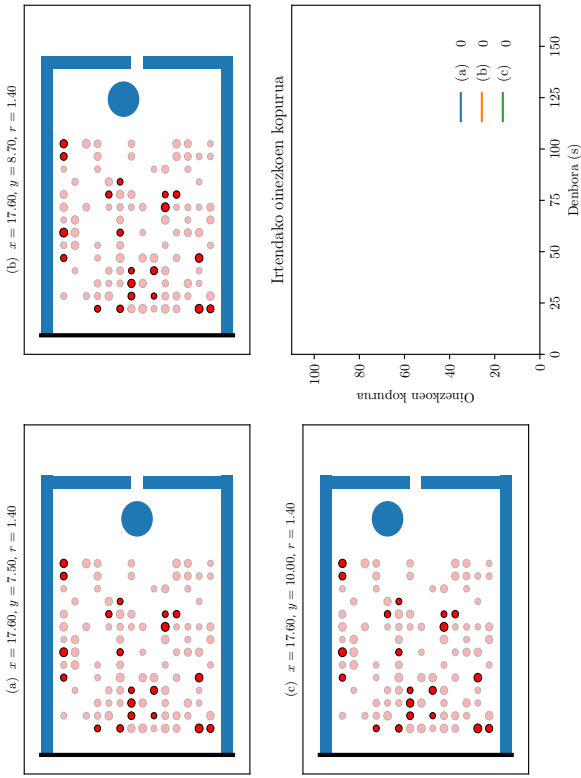
$t = 30s$



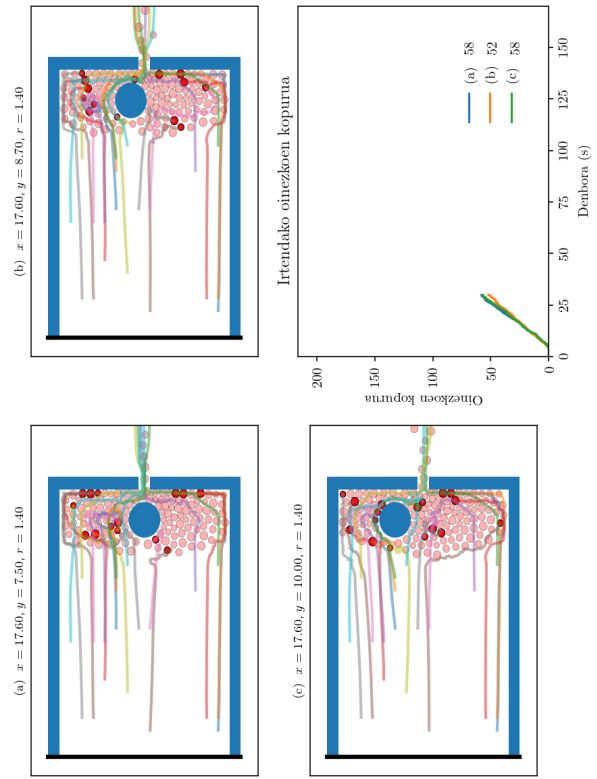
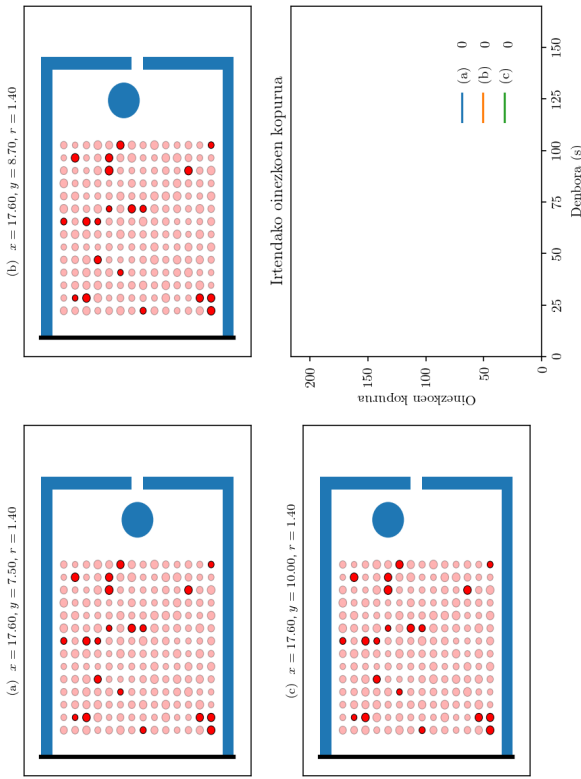


L6. Esperimentua

N=100

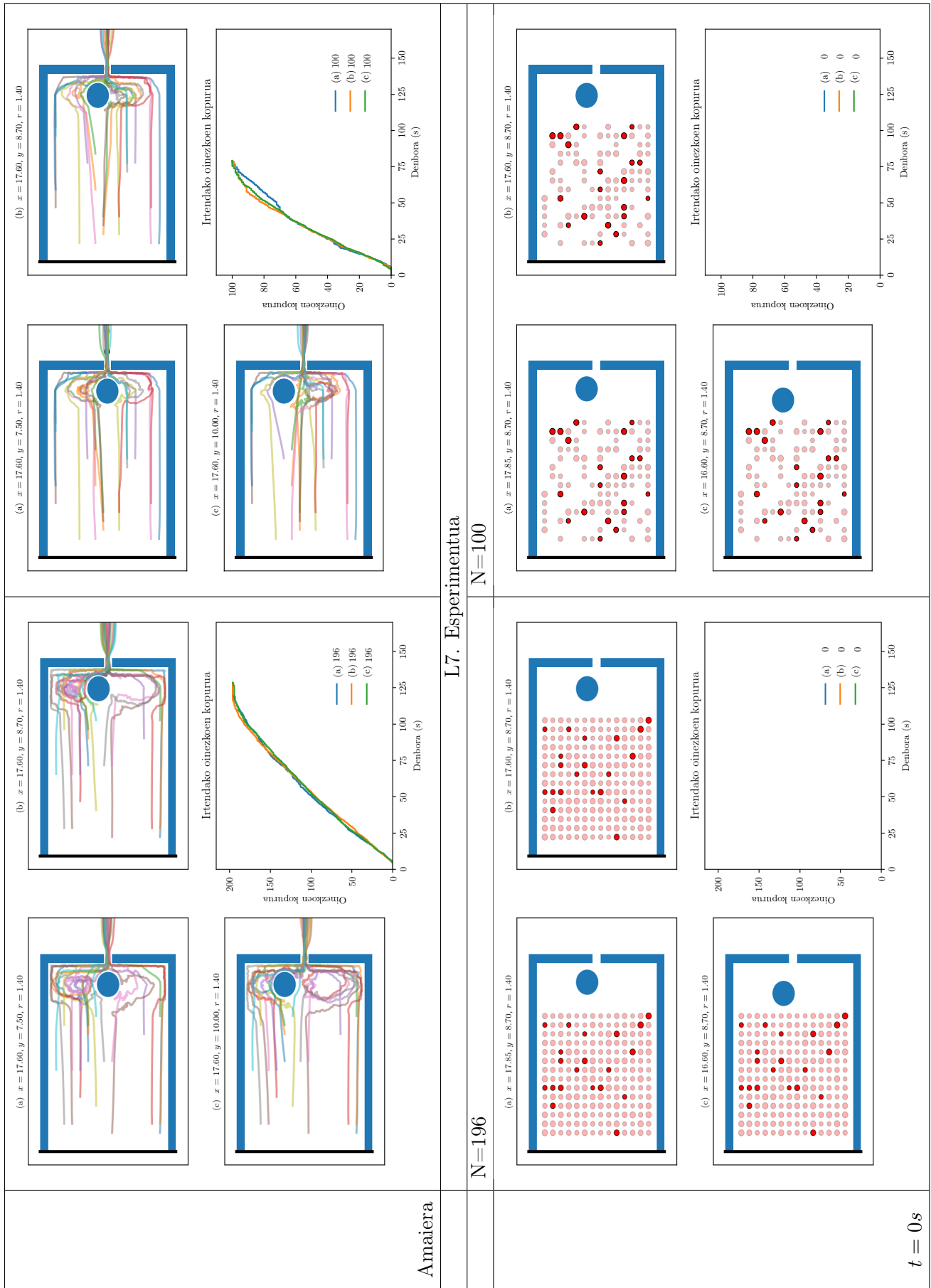


N=196

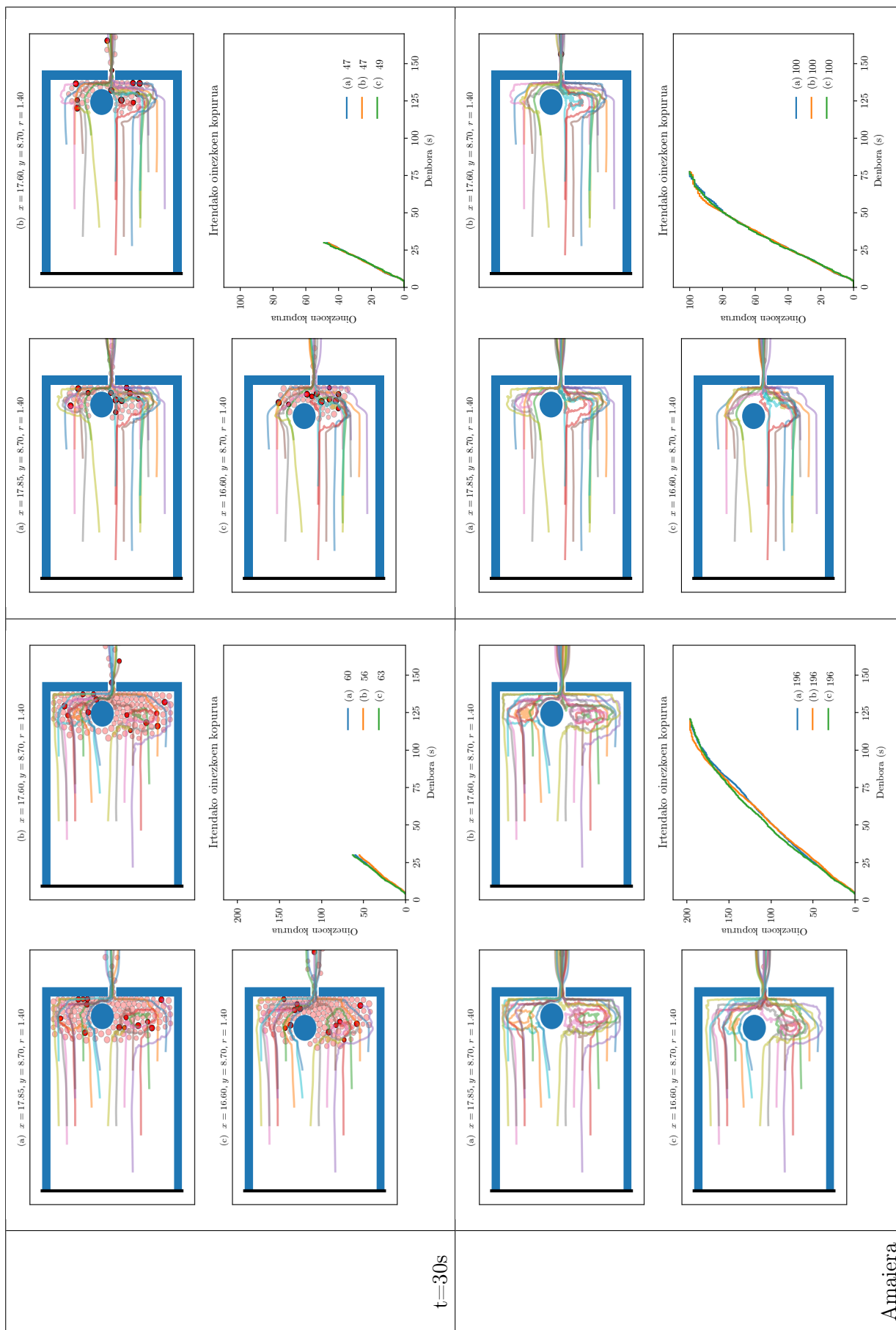


t = 0s

t = 30s

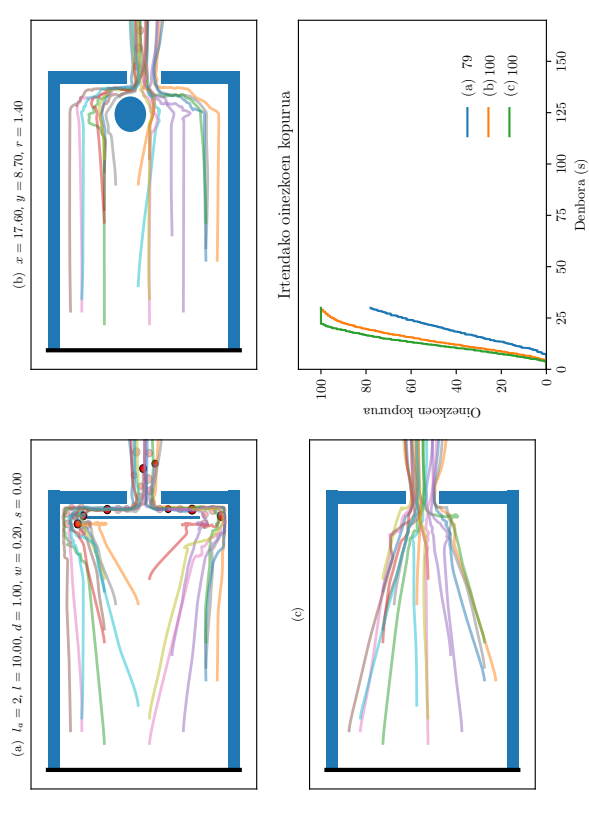
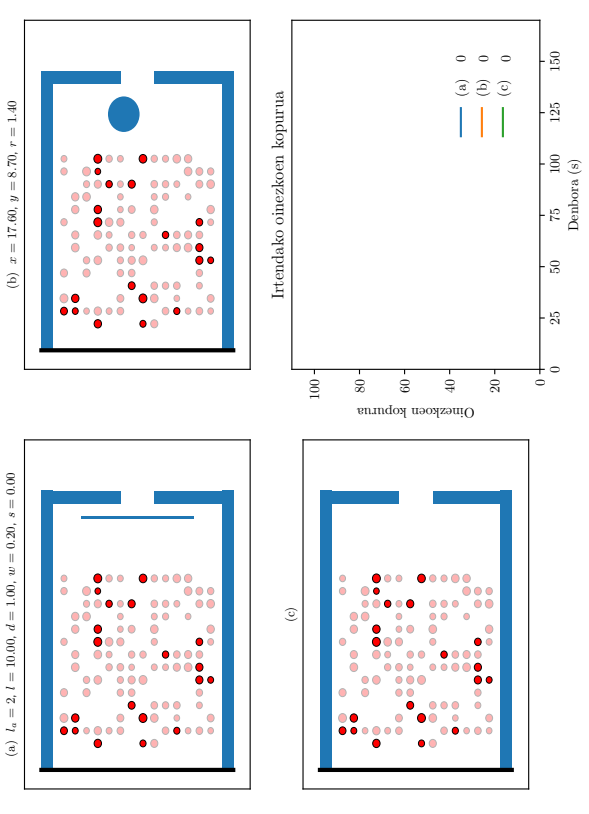




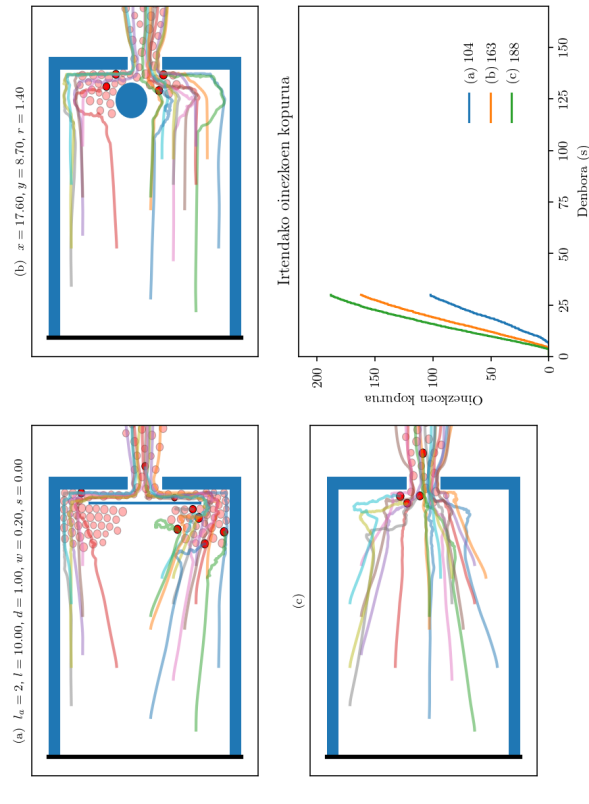
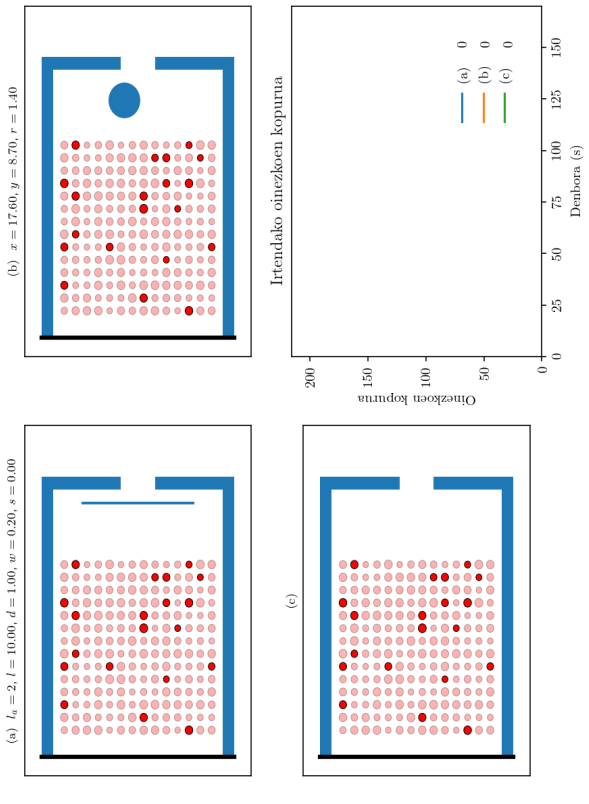


I.8. Esperimentua

N=100

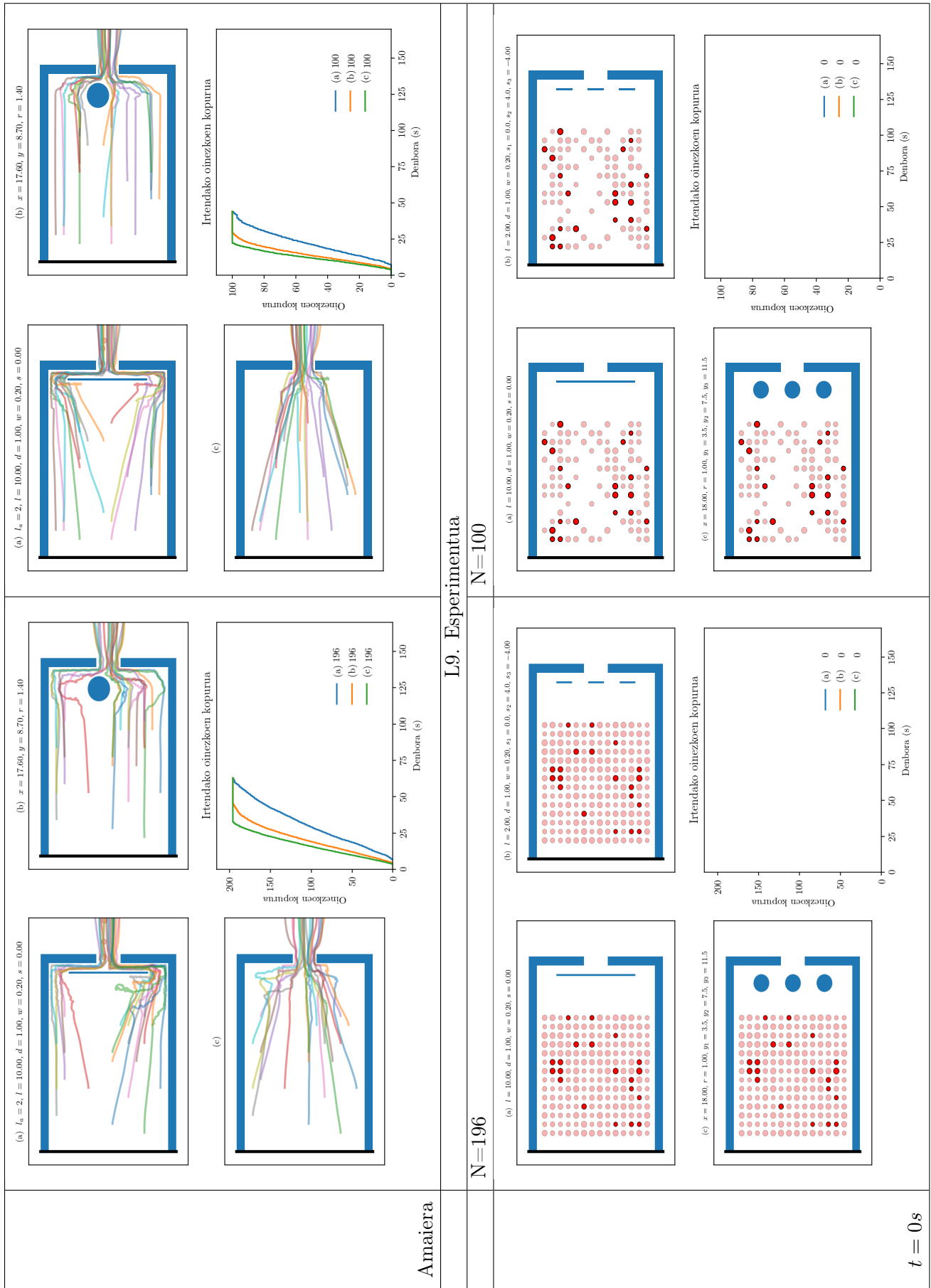


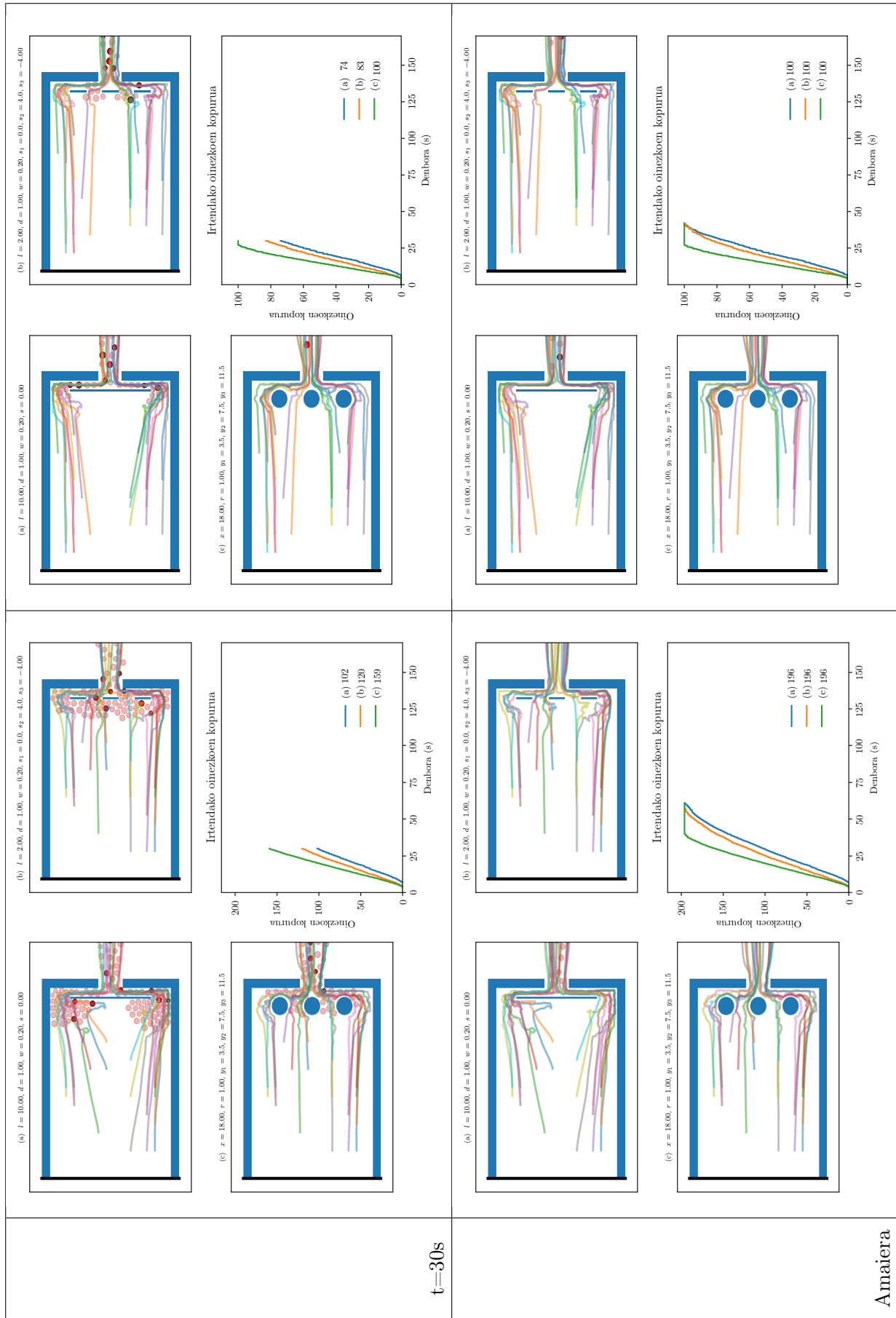
N=196



$t = 0s$

$t = 30s$



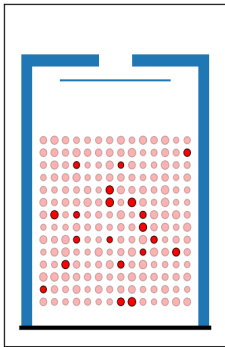


L10. Esperimentua

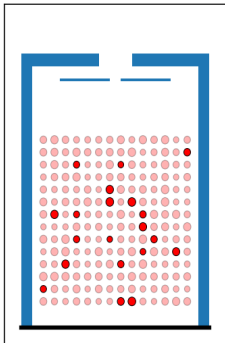
N=100

N=196

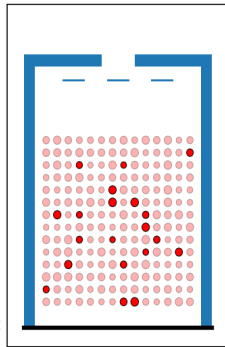
(a)  $I = 10.00, d = 1.00, w = 0.20, s_1 = 0.00, s_2 = -4.00$



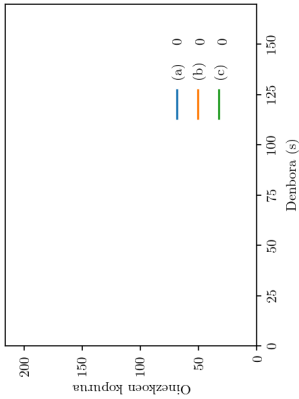
(c)  $I = 4.50, d = 1.00, w = 0.20, s_1 = 3.25, s_2 = -3.25$



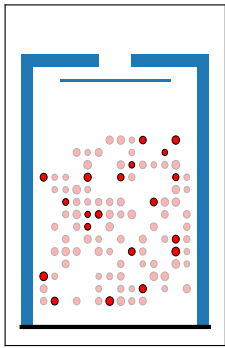
(b)  $I = 2.00, d = 1.00, w = 0.20, s_1 = 0.0, s_2 = 4.0, s_3 = -4.00$



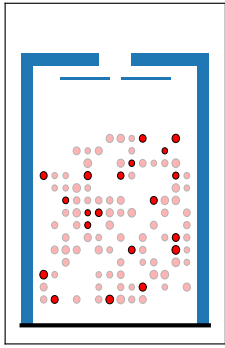
Irtendako oinezkoen kopurua



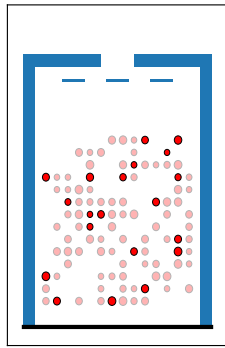
(a)  $I = 10.00, d = 1.00, w = 0.20, s_1 = 0.00, s_2 = 0.00$



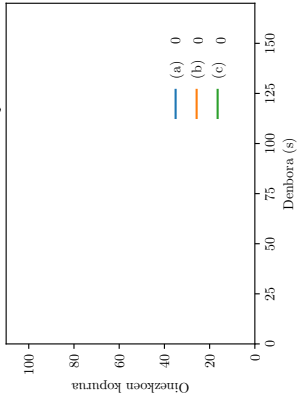
(c)  $I = 4.50, d = 1.00, w = 0.20, s_1 = 3.25, s_2 = -3.25$



(b)  $I = 2.00, d = 1.00, w = 0.20, s_1 = 0.0, s_2 = 4.0, s_3 = -4.00$

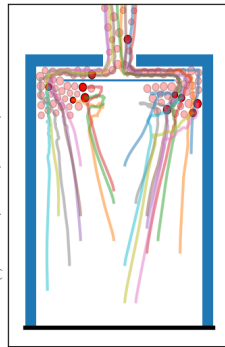


Irtendako oinezkoen kopurua

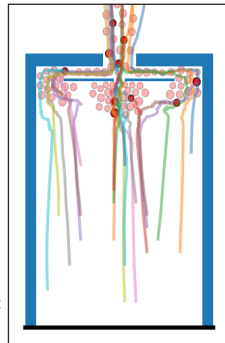


$t = 0s$

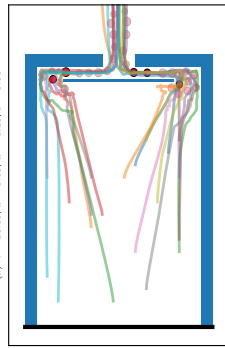
(a)  $I = 10.00, d = 1.00, w = 0.20, s_1 = 0.00, s_2 = -4.00$



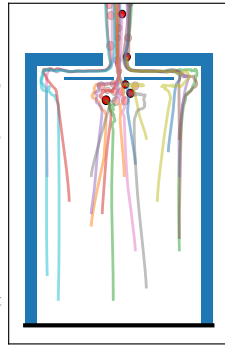
(c)  $I = 4.50, d = 1.00, w = 0.20, s_1 = 3.25, s_2 = -3.25$



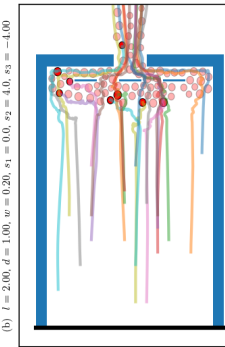
(b)  $I = 2.00, d = 1.00, w = 0.20, s_1 = 0.0, s_2 = 4.0, s_3 = -4.00$



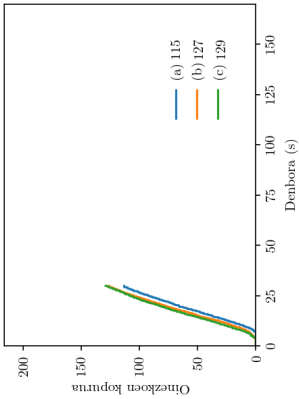
(c)  $I = 4.50, d = 1.00, w = 0.20, s_1 = 3.25, s_2 = -3.25$



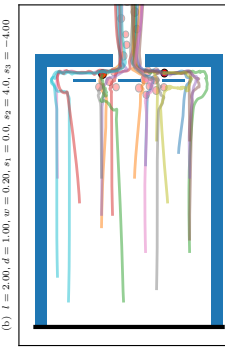
(b)  $I = 2.00, d = 1.00, w = 0.20, s_1 = 0.0, s_2 = 4.0, s_3 = -4.00$



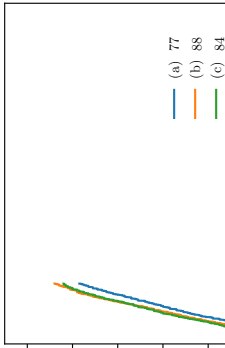
Irtendako oinezkoen kopurua



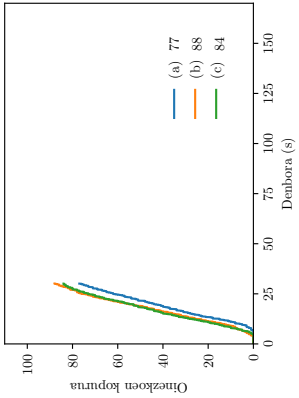
(a)  $I = 10.00, d = 1.00, w = 0.20, s_1 = 0.00, s_2 = 0.00$



(c)  $I = 4.50, d = 1.00, w = 0.20, s_1 = 3.25, s_2 = -3.25$



Irtendako oinezkoen kopurua



$t = 30s$

