

Eötvös Loránd Tudományegyetem

Természettudományi Kar

Fizika TDK dolgozat



Axionmező szimulálása adaptív ráccsal

Tóth Ábel

ELTE TTK Fizika szak BSc 2. évfolyam

Témavezető:

Katz Sándor

ELTE, Elméleti Fizika Tanszék

Kivonat

Peccei és Quinn az erős CP probléma feloldására javasolták az axion nevű hipotetikus részecskét, ami, ha megfelelő a tömege, a sötét anyag meghatározó alkotóeleme is lehet. Az axionikus sötét anyag energiájának egy része a részecskét leíró komplex skalármező topologikus defektusaiban, nagyon kicsiny hurokban tárolódik. Emiatt a releváns méretek között akár harminc nagyságrend különbség is felléphet, ami megnehezíti a mező időbeli fejlődésének numerikus szimulálását, szélsőséges esetekben egyenletes ráccsal egyáltalán nem kapható pontos eredmény. A kutatásom során egy adaptív rácsot használó szimulációt írtam, amivel két dimenzióban a skálakülönbség ellenére is pontosan modellezhető az axionmező fejlődése. A szimuláció alapjául a quadtree algoritmus szolgál, aminek előnye egyéb (pl. háromszög-) rácsokkal szemben, hogy egyszerűen általánosítható három dimenzióra, ahol octree algoritmusnak hívják. Ez az algoritmus széles körben használt 2D és 3D grafikához, illetve videójáték motorokhoz, de legjobb tudásom szerint ez az első alkalmazása mezőelméleti szimulációra. A szimuláció megalkotásában a legnagyobb kihívást egy olyan diszkretizáció megtalálása jelentette, amivel az energia a rács időbeli változása ellenére is jól megmarad. A dolgozatomban kifejtem az erre adott megoldásomat, ismertetem az algoritmus működését, végül olyan folyamatokra is bemutatom a szimuláció jogosságát, amikre egyenletes rács már csak nehezen alkalmazható.

Tartalomjegyzék

1. Bevezetés	3
2. Elméleti háttér	4
2.1. Az axiondinamikát leíró egyenletek	4
2.2. Topologikus defektusok	5
3. Az algoritmus működése	6
3.1. Létező módszerek a mező szimulálására	6
3.2. A quadtree algoritmus bevezetése	7
3.3. A használt elnevezések	8
3.4. A rács időbeli változtatása	9
3.5. Mezőértékek időbeli léptetése	10
4. Diszkretizáció	11
4.1. Egyenletes rács diszkretizációja	11
4.2. Szempontok a diszkretizáció kiválasztásához	13
4.3. Az általam használt diszkretizáció	13
4.4. A szélső pontok rögzítése	17
5. Axionmező szimulálása	19
5.1. Kezdőfeltétel	19
5.2. Frontális ütközés lefolyása	21
5.3. Nem frontális ütközés	24
6. Adaptív és egyenletes rács összehasonlítása	26
6.1. A leggyorsabb futáshoz szükséges paraméterek meghatározása	26
6.2. A komplexitás elméleti meghatározása	28
6.3. A futásidők kísérleti meghatározása	28
6.4. Összehasonlítás másik adaptív ráccsal	30
7. Összefoglalás	32
8. Kitekintés	32
A. A szimulációról és az ábrázoló szoftverről	36
B. Egy cella teljes Hamilton-függvénye	37
C. A futásidők mérési adatai	38

1. Bevezetés

Az erős kölcsönhatás a fundamentális kölcsönhatások egyike, a standard modell, azon belül is a kvantum-színdinamika (QCD) egyenletei írják le. Ezek alapján semmilyen mechanizmus nem erőltet rá CP szimmetriát¹ [1], a Lagrange-sűrűségfüggvényében megjelenő szimmetriasértő tag együttthatója akármennyi lehet. Mérések azonban ennek nagyságrendjét 10^{-9} alattinak határozták meg [2], és mérési pontosságon belül az erős kölcsönhatás CP szimmetriát mutat. Az erős CP probléma arra keresi a választ, hogy miért jelenik meg egy olyan szimmetria, amit a kölcsönhatást leíró egyenleteink nem indokolnak. Peccei és Quinn erre a felvetésre adtak egy elegáns választ, egy új részecske, az axion létezésének feltételezésével [1].

Az axiont az erős CP probléma kontextusában kezdték eredetileg vizsgálni, azonban gyengén, de nagy hatótávolságban is kölcsönható részecskeként felmerült, hogy a sötét anyag egyik, akár fő alkotóeleme is lehet [3]. Ha ez valóban így van, akkor az, hogy mekkora hányada lehet a sötét anyagnak, csupán egyetlen paramétertől (az axion tömegétől) függ, amit numerikusan, szimuláció segítségével lehet meghatározni. Az e téren való kutatást motiválja, hogy a mai technológiával már léteznek olyan elrendezések, amivel, ha ez a paraméter ismert, akkor kimutatható (vagy esetleg cáfolható) lenne az axion létezése. A szimulálást azonban megnehezíti, hogy a szimulált objektumok karakterisztikus energiaskálája tipikusan $10^{10} - 10^{15} \text{ GeV}$, míg a rendszer méretét jellemző Hubble-skála $H \approx 10^{-17} \text{ GeV}$, amik között így akár 30 nagyságrend különbség is lehet [4].

A tipikus módszer mezőelméleti szimulációkra egy nagy, egyenletes rács felvétele, ennek pontjait kell időben léptetni. Ez az út azonban a drasztikus skálakülönbség miatt nem járható, hiszen az a rács, ami kellően részletesen felbontja a részecskéket, már abszurd mennyiségű pontot tartalmaz. Emiatt egyenletes rács helyett adaptív rácsot érdemes használni, vagyis olyat, ami a nagy energiasűrűség közelében több adatpontot, attól távol pedig csak néhányat tartalmaz. Az adaptív rács előnye, hogy a skálakülönbség növelésével a szükséges pontok száma nem köbösen (vagy két dimenzióban négyzetesen), hanem logaritmikusan nő, így lehetségessé válik szélsőséges esetek kezelése. Hátránya azonban, hogy míg az egyenletes rács esetében az időbeli léptetés egyszerű összefüggésekből számolható, addig az adaptív rács esetében az adatpontok elhelyezkedésének bonyolult szerkezete miatt a diszkretizáció már egyáltalán nem triviális, és a legtöbb eljárás az energiamegmaradás instabilitásához vezet. További nehézséget jelent, hogy egyenletes rácsra egyszerű leapfrog léptetéssel² garantálható, hogy az energia

¹C-"charge"-töltéskonjugálás, P-"parity"-térbeli tükrözés, és T-"time"-időtükrözés, illetve ezek kombinációi alapvető szimmetriák, amikkel bizonyos rendszerek rendelkezhetnek. A tudomány jelen állása szerint az egész világegyetem rendelkezik CPT szimmetriával, de minden más kombináció sérül valahol.

²A leapfrog algoritmus egymáshoz képest fél időközzel eltolva lépteti a mező értékét, illetve annak

hibája az időléptetés négyzetével arányos, de adaptív esetben a rácspontok mozgatása az időköztől független hibát okoz, így az energiamegmaradás pontossága nem növelhető csupán az időköz csökkentésével.

A dolgozatom célja egy általam írt adaptív algoritmus bemutatása, ami két dimenzióban áthidalja a fenti problémákat és sokkal gyorsabb a hasonló pontosságú egyenletes rácús szimulációnál, és mindemellett könnyen általánosítható három dimenzióra. Ennek megfelelően a hangsúly az algoritmus működésének és az ehhez használt diszkretizációnak a részletezésén, illetve a szimuláció sebességének és pontosságának az egyenletes rácúval való összehasonlításán lesz, a szimulálások eredményei (azon túl, hogy visszaadják az irodalmi értékeket) a dolgozat céljának szempontjából mellékesek.

2. Elméleti háttér

2.1. Az axiondinamikát leíró egyenletek

Az erős kölcsönhatás Lagrange-sűrűségfüggvényének általános felírásában a mérésekkel gyakorlatban is kimutatott $\mathcal{L}_{QCD}$ tagon kívül szerepelhet egy CP szimmetriát megsértő tag is³ [1]:

$$\mathcal{L} = \mathcal{L}_{QCD} + \Theta \frac{g^2}{32\pi^2} F^{\mu\nu} \tilde{F}_{\mu\nu} \quad (1)$$

Kísérleti eredmények szerint a CP szimmetria nem sérül, $|\Theta| < 10^{-9}$. A paraméter rejtélyesen kis értékére nyújt magyarázatot Peccei és Quinn elmélete szerint egy komplex skalármező, amihez a

$$\mathcal{L}_\phi = \partial_\mu \phi^* \partial^\mu \phi - V_k \quad (2)$$

Lagrange-sűrűségfüggvény tartozik, és

$$\Theta = \arg(\phi) \quad (3)$$

Így (1) az alábbi formában írható fel [5]:

$$\mathcal{L} = \mathcal{L}_{QCD} + \arg(\phi) \frac{g^2}{32\pi^2} F^{\mu\nu} \tilde{F}_{\mu\nu} + \mathcal{L}_\phi + \dots \quad (4)$$

ahol még a ϕ mező és a standard modell kapcsolatát leíró egyéb modellfüggő tagok is megjelennek.

időderiváltjának értékét, bővebben a 3. fejezetben mutatom be.

³A dolgozatban használt egységrendszer a $c = \hbar = k_B = 1$, a metrikus tenzor szignatúrája $[+, -, -, -]$, és az Einstein-féle összegzési konvenciót használom.

A (2) egyenletben szereplő potenciál, V_k értéke:

$$V_k = \lambda (\phi\phi^* - f_a^2)^2 \quad (5)$$

ahol λ egy dimenziótlan konstans. Ez egy negyedfokú potenciál, amelynek minimuma van f_a -nál, és globális $U(1)$ szimmetriával rendelkezik, így a Θ fázis bármennyi lehet. Azonban, ha kezdetben a $\phi = 0$ mezőértéknél, az $U(1)$ szimmetriát teljesítve helyezkedik el minden pont, akkor a szimmetria spontán megtörik a tér pontjaiban, ahogy felvesznek egy-egy fázist a potenciálminimum felé haladva. Erről és ennek következményeiről bővebben a (2.2) fejezetben lesz szó.

A (4) Lagrange-sűrűségfüggvény átírható az $a = f_a\Theta$, változóra, ezt a mezőt hívjuk axionnak. Ekkor a kvantum-színdinamika axionnal való gyenge kölcsönhatása miatt megjelenik egy új, V_χ potenciál. Ez tartalmazza a hőmérsékletfüggő $\chi(T)$ állandót, a topologikus szuszceptibilitást, ami magas hőmérsékleten $T^{-7,5}$ -nel arányosan cseng le [6], így csak alacsony energiánál játszik szerepet:

$$V_\chi = \chi(T) (1 - \cos(\arg(\phi))) \quad (6)$$

Szemléletesen ez a potenciál megdönti az (5) egyenlet negyedfokú potenciálgörbáját a $\Theta = 0$ irányba. Így hosszú távon a tér pontjainak nagy része a nulla fázis környékén fog elhelyezkedni, ez magyarázhatja a kísérletileg kimutatott alacsony értéket.

2.2. Topologikus defektusok

Az (5) potenciál következményeként a rendszerben megjelenhetnek topologikus defektusok [7]. Nagy hőmérsékleten (6) potenciál járuléka elhanyagolható, így $\phi = 0$ kezdeti feltétel mellett a tér pontjai véletlenszerű fázist vesznek fel, ahogy a mező abszolútértéke a potenciálminimumhoz tartozó f_a értékhez közelít. Ekkor azonban előfordulhat, hogy tér egy zárt kontúrján végighaladva a mező fázisa összesen $2k\pi$ növekedést (vagy csökkenést) szenved, ahol $k > 0$ egész szám. A kontúrt folytonosan kicsinyítve oly módon, hogy a fázisugrás megmaradjon, találni fogunk egy olyan pontot, ahol $\phi = 0$, hiszen a mező folytonosan változik. Ebben a pontban és közvetlen környezetében az energiasűrűség nagyon magas, és a fázis szingulárisává válik. Az ilyen egydimenziós defektusokat, amik körüli kontúron a fázis ugrik, húroknak hívják. Ezeket a fázisugrás előjelének függvényében egy $\pm k$ töltésű részecskének tekinthetjük.

Ezek azért topologikus defektusok, mert folytonos transzformációval nem szüntethetők meg [8]. Ennek következménye, hogy (6) potenciál bevezetésével, ahogy a fázisértékek nagy része a 0-hoz közelít, a hurok körül megmarad a fázis teljes spektruma, egészen addig, amíg össze nem ütközik egy ellentétes előjelű töltéssel rendelkező másik húrral. Két dimenzióban a hurok pontszerű részecskéként jellennek meg, ezeket vortexeknek is

nevezik. Mivel az axionmező esetében általában három dimenzióban és húrokról beszélünk, ezért a két dimenzióban fellépő pontszerű topologikus defektusokra is így szoktak hivatkozni. Két dimenzióban bizonyos számolások analitikusan is elvégezhetőek, két távoli részecske egymásra kifejtett hatására például teljesülnek a kétdimenziós Maxwell egyenletek, így alkalmazható elektromágneses analógia [9].

3. Az algoritmus működése

3.1. Létező módszerek a mező szimulálására

Egy mező számítógéppel való szimulálásához a tér kontinuum pontja közül véges sokat kell kiválasztanunk, és ezekre kell a számolásokat elvégezni. A triviális megközelítés a pontok egyenletes felvétele. Ez egy időben állandó (n dimenziós) négyzetrács, aminek segítségével nagyon jó energiatartás mellett követhető a mező fejlődése, amennyiben kellően sűrű a felbontása. Emellett a mezőnek az időbeli léptetése igen egyszerű, a hiba pedig a léptetési időköz csökkentésével csökkenthető. Továbbá, egyenletes ráccsal szimuláláshoz a mező lokalitása miatt jól használható GPU alapú optimalizálás, amivel nagyon sok számolás párhuzamosan történik, jelentősen csökkentve a futásidőt. A megközelítés hátulütője, hogy amennyiben a "kellően sűrű" felbontás rácsállandója túl kicsi a mért térfogat (vagy 2D-ben a terület) méreteihez képest, úgy a futásidő rohamosan nőni fog. A szükséges pontok száma 2D-ben négyzetesen, 3D-ben köbösen arányos a vizsgált terület vagy térfogat oldalhosszával, így éppen az axionok esetében, ahol a skálakülönbség akár 30 nagyságrend is lehet, ez az út teljesen járhatatlan.

Az adaptív rácsok lényege, hogy a felbontás mindenhol csak akkora, amekkora *lokálisan* szükséges. Ennek következtében a szükséges adatpontok száma nem L^2 -tel vagy L^3 -bel, hanem $\log L$ -lel arányos (bővebben a 6.2 fejezetben), ahol L a vizsgált területet vagy térfogatot jellemző dimenziótlan lineáris hosszúság. Bár ez megoldja az egyenletes rács legnagyobb problémáját, de emellett elveszíti annak összes fent említett előnyét is (tehát a felbontásnak időben változnia kell, a hiba nem csökkenthető le bármeddig az időlépés csökkentésével, a számolások igen bonyolulttá válnak), így a szimulálás minden szempontból sokkal nehezebbé válik.

Az adaptív ráccsal való szimulálásra több megközelítés is létezik [11]. Az első, a h-adaptív rács, egy empirikusan felvett hibabecslő függvényen alapul [12, 13, 14]. A szimuláció először egy durva, adaptív ráccsal végez egy időlépést, majd a hibabecslő függvények segítségével meghatározza, hogy hol nagy a hiba, és ezeken a helyeken lokálisan növeli a rács felbontását, csökkentve a hibát. A hibabecslést és pontosítást újra és újra elvégzi, amíg el nem éri a megfelelő pontosságot, majd ezzel a nem egyenletes ráccsal lépteti az időt. A második, az r-adaptív rács alapja a nem egyenletes rács

leképezése egy speciális egyenletes rácsra, és ott elvégezni a számolást[15, 16, 17]. Létezik továbbá p-adaptív rács is, ami a h-adaptívhoz nagyon hasonló, de speciálisan a végeselem módszerrel⁴ végzett diszkretizáció esetén működik csak [19].

Egy másfajta megközelítés az ún. "smoothed particle method". Ebben a koordináták együtt mozognak a "részecskékkal", vagy itt a nagy energiasűrűségű térrészekkel[20, 21, 22]. Ezen módszerek mindegyikének megvan a maga nehézsége, ami miatt az alkalmazásuk nem egyszerű [11], ezt itt nem részletezem.

3.2. A quadtree algoritmus bevezetése

Az általam írt szimuláció adaptív rácsot használ, de egyik korábban említett kategóriába sem illik igazán. A rácsponthoz felvételének hátterében a *quadtree* algoritmus áll, ami a felbontás fraktálszerű, rekurzív növelésén alapszik (hasonlóan igaz ez a quadtree többdimenziós általánosításaira is) [23, 24]. Az egyszerűség kedvéért ennek működését két dimenzióban fogom ismertetni, de több dimenzióban is ugyanúgy alkalmazhatók az alapelvek.

Az algoritmus az adaptív rács megtalálására egy nagyon durva $N_0 \cdot N_0$ -as egyenletes rácsból indul ki. Ezután minden régióhoz (a rács pontjai által alkotott elemi kisnégyzethez) egy fitness-függvény segítségével hozzárendel egy értéket, és ahol a legmagasabb ez az érték, azt a régiót felbontja négy kisebb, tipikusan szintén négyzet alakú régióra. Ezekhez az új négyzetekhez is hozzárendel egy fitness értéket, és megint felbontja a legnagyobb fitness-ű felbontatlan régiót, és így tovább, amíg eléri a szükséges felbontást. A szimuláció eredménye egy fa-struktúrában tárolt régiók halmaza⁵. Megjegyzendő, hogy általában $N_0 = 1$ határeset használt, hiszen az algoritmus úgyis megtalálja a legnagyobb fitness-ű régiókat, de az axionmező szimulálásához érdemes $N_0 \geq 10$ -et választani, mert ez az alapváz különböző N_0 -al való tesztelések alapján segíti az energia pontosabb megmaradását.

Ezen algoritmus (illetve a három dimenziós változata, az octree) különböző formáinak elterjedt a kereskedelmi használata két-, illetve háromdimenziós grafikához és videójáték motorokhoz⁶, de általánosan alkalmazott egyebek közt robotikában, mo-

⁴A végeselem módszer parciális differenciálegyenletek közelítő megoldására használt módszer, amit elsősorban mérnökök alkalmaznak mechanikai problémákra. A módszer alapja bonyolult alakú testek véges elemekre való bontása majd ezekkel való számolás [18].

⁵A quadtree algoritmusnak több fajtája is létezik, az általam leírtak a régió- és a pont-régió quadtree algoritmusra igazak, a szimulációmban ezek közül utóbbit használom. Más változatok nem feltétlenül négyzet alakú régiókkal dolgoznak, vagy régiók helyett pontokat vagy éleket tárolnak el, és egyéb eltérések is lehetnek a különböző implementációk között. Az alapgondolat, hogy hely szerint rekurzívan felbontva a teret új és új adatokat kapunk, amiket egy fastruktúrában tárolunk, általánosan igaz.

⁶Pl. a Cube 2 Sauerbratuen játék szinte kizárólag octree-n alapul, az id Tech 6 motor octree-kben tárolt voxeleket használt, az OGRE nevű 3D grafika renderelő scene-manager implementációja is octree-vel fut.

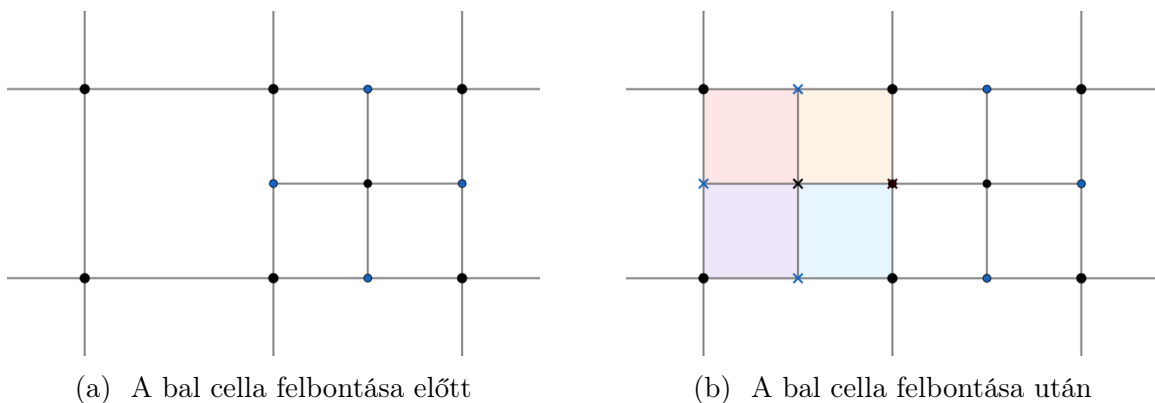
lekuláris modellezéshez és 3D nyomtatáshoz. Mindemelett a fizika több területén is használt: hidrodinamikai és asztrofizikai szimulációkhoz, illetve időjárás-előrejelzéshez [25, 26, 27].

3.3. A használt elnevezések

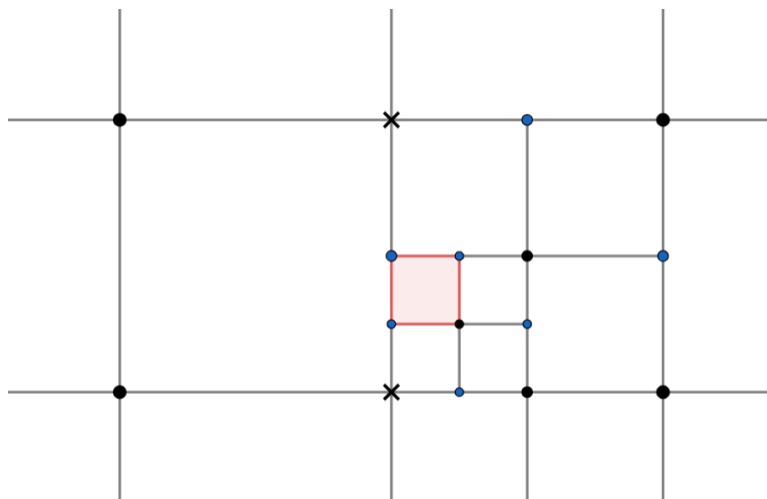
A továbbiakban az axionmező szimulálásakor a quadtree felbontás elvégzése után kapott elemi régiókat *celláknak* hívom. Azok a négyzetek, amik egyszeresen fel vannak bontva négy cellára, azok *egyszeresen felbontott cellák*. Ha csak az egyik negyedében lévő négyzet tovább van bontva még egyszer, akkor a cella már nem egyszeresen felbontott. Legyenek *őscellák* a kezdeti $N_0 \cdot N_0$ -as durva rács rácsponthai által határolt, akár azóta már továbbbontott négyzetek.

Nevezzük *pontoknak* az adatpontokat, amik egy adott időpontban értelmezett struktúrák, és a tulajdonságaik egy rögzített x és y koordináta, illetve egy adott időpillanatban ehhez a helyhez tartozó mezőérték, illetve impulzusérték. Minden cellához definíció szerint négy pont tartozik (ami a négy sarokban van), mert új cellák képzésekor (vagyis egy korábbi cellák felbontásakor) mindig felvesszük az új cellák sarkaiba kerülő pontokat új adatpontként. A felbontás menetét a 1. ábra illusztrálja.

Tehát egy cella felbontásakor öt új pont és négy új cella keletkezik, míg az eredeti cella egyszeresen felbontottá válik (de a cellát tartalmazó nagyobb cella immár



1. ábra. Egy cella felbontása és következményei. A bal oldali cella felbontásával öt új pont jelent meg, amiket pöttyök helyett "x"-ek jelölnek. Beszínezve látható a keletkező négy új cella. Azon pontok, amik négy cellának is a sarkai (tehát középső pontok) azok fekete színnel, amik pedig csak kettőnek (vagyis szélső pontok) azok kékkel vannak jelölve. Külön figyelmet érdemes fordítani az ábrákon középen megjelenő pontnak. Mivel ez a pont a bal cella felbontása előtt is létezett, már tartozott hozzá két cella, aminek ő a sarka. A bal cella felbontása után már négy ilyen cella is van, így szélső (kék) pontból középső (fekete) ponttá alakul. A felbontás után a pont pöttyel és "x"-szel is meg van jelölve, ezzel mutatva, hogy a régi, jobboldali, és az új, baloldali celláknak is sarka.



2. ábra. Illegális felbontás. Az ábrán többféle módon megállapítható hogy ez a felbontás nem lehetséges:

- Az "x"-szel jelölt két pont közti szakasz az egyik oldalról felbontatlan, a másik oldalról kétszeresen is felbontott.
- Az ábrán találunk szomszédos szélső (kék) pontokat
- A pirossal jelölt cellában három szélső pont is van.

Ezen három állítás egyike sem lehet igaz egy szabályos felbontásra.

biztosan nem csak egyszeresen felbontott). Vegyük észre, hogy ugyan minden cellához tartozik négy pont, de nem minden ponthoz tartozik négy cella (aminek az a pont a sarka). Ez alapján megkülönböztetünk középső és szélső pontokat, amik rendre négy, illetve kettő különböző cellának a sarkai. A quadtree algoritmust úgy végeztem el, hogy olyan cellát, aminek bármelyik sarka szélső pont, nem bontok tovább (akármekkora is az energiája), egészen addig amíg a szomszédos cellák felbontása miatt már nem szélső pont egyik sarka sem. Így semmilyen körülmény között sem lehet két szomszédos cella érintkező éle mentén több mint kettővel nagyobb felbontás az egyik oldalon, ahogy a 2. ábra mutatja. Ez máshogy is megfogalmazható: két szomszédos pont (olyan pontpár, ami egy cella két szomszédos sarkát alkotja) közül nem lehet mindkettő szélső, azaz egy szélső pont minden szomszédja középső pont (lásd 2. ábra).

3.4. A rács időbeli változtatása

Az axion szimulálása esetében a nagy energiájú húrokra szeretnénk ránagyítani, ezért a fitness-függvényre értelemszerű választás a cellában (régióban) tárolt energia. Az axionmező időben természetesen változik, és vele változik az ideális felbontás geometriája is. Ezért minden egyes időben léptetés után megkeressük a legnagyobb energiájú felbontatlan cellát és a legkisebb energiájú, (pontosan) egyszer felbontottat. Ha előbbi energiája nagyobb, akkor azt felbontjuk, utóbbinak pedig visszafejtjük a felbontását, így a rács leköveti az energiasűrűség változását. A felbontást a korábban tárgyalt módon

végezhetjük, a visszafejtés során pedig ugyanez történik, csak visszafelé: felszabadítjuk a memóriában (töröljük) a benne lévő négy elemi cella és az ezek sarkaiként felvett öt pont helyét. A kezdeti, $N_0 \cdot N_0$ -as durva rács cellái, az őscellák mindig megmaradnak, mert ők nem egy korábbi cella felbontásával keletkeztek.

Ahhoz, hogy minden egyes felbontással és visszafejtéssel 5-5 pontot vegyünk fel, illetve töröljünk ki, némely pontot kétszer tárolunk el (lásd az 1b ábrán a középső pont). Ezek értékeit természetesen összhangban tartjuk. A pontok törlése és behelyezése miatt az is változhat, hogy melyik pont középső és melyik szélső, ezt szintén számon kell tartani. Így tehát állandó minden időpillanatban az egyszerre eltárolt pontok száma, ezt N_{max} -nak nevezzük. Ez a szám az adaptív rács felbontását jellemzi.

A szimuláció ezen lépése - pontok felvétele és törlése - olyan hibát okoz, ami az időköz csökkentésével nem csökkenthető (ellentétben a mezőértékek időbeli léptetésével, lásd (3.5)), ezért ezt lehetőleg minél ritkábban akarjuk elvégezni. Ezért gyakorlatban nem végzem el, ha a legnagyobb energiájú elemi cella energiája csupán egy kicsit nagyobb, mint a legkisebb energiájú egyszeresen felbontott celláé, csak akkor, ha már k -szor nagyobb. Ha k túl kicsi, akkor túl gyakran végezzük el ezt a nem energiatartó lépést, ha túl nagy, akkor nem követi le elég jól a felbontás az energiaváltozást. Elméleti alapon nehéz megmondani, hogy mekkora legyen ez az érték, de több érték kipróbálása után $k = 3.5$ alkalmasnak tűnt, ezt használtam minden szimulációnál.

3.5. Mezőértékek időbeli léptetése

A mező időbeli fejlesztéséhez bevezetjük a pontokra jellemző változónak a mezőérték mellett azok időbeli deriváltjait is, ezzel elsőrendűvé téve a mozgásegyenletet. Így áttérünk a kanonikus egyenletekre. Ezek pontos alakjáról a következő fejezetben lesz szó.

A mezőt az alábbi három lépésben fejlesztem:

1. Minden pontban a mező értékét léptetem $p \cdot \frac{dt}{2} - vel$, ahol p a mező időbeli deriváltja az adott pontban, dt pedig az egész szimulációra jellemző állandó, az időlépés nagysága.
2. Minden pontban meghatározom (az új mezőértékek alapján) a p értékek idő szerinti deriváltjait, majd léptetem p -t ennek dt -szerezésével.
3. Megismétlem az első lépést; minden pont mezőértékét fejlesztem az új p értékek $\frac{dt}{2}$ -szerezésével.

A három lépés elvégzése után minden pont minden paramétere dt idővel fejlődött. E három lépés ismételtetése az úgynevezett leapfrog algoritmus.

A leapfrog algoritmus előnye az ugyanennyi lépésből álló Euler-integrálással szemben, hogy másodrendű módszer, vagyis az időlépés négyzetével arányos a hibája (az

energiaváltozás) [28]. A Runge-Kutta és Yoshida algoritmusok még magasabb rendűek, de több lépésből is állnak, ami lassítja a szimulációt. Meg kell jegyezni azonban, hogy amennyiben az időparamétert is adaptívvá tesszük, akkor a leapfrog algoritmus már nem lesz stabil [29], így ilyenkor más módszert kell használni.

A szimuláció tehát elvégez egy leapfrog léptetést, majd kiszámolja a fitness-függvényeket. Ha szükséges, akkor változtatja a rácsot, majd újra lépteti az időt a leapfrog algoritmussal.

4. Diszkretizáció

4.1. Egyenletes rács diszkretizációja

Egyenletes rács esetében több egyszerű diszkretizáció is alkalmazható. Ezek közül most bemutatom azt, amit később a szimulációhoz használtam. Itt, és a fejezet minden további alfejezetében $f_a = 1$ egységrendszerben végeztem a számolásokat, de a ϕ mező átskálázásával tetszőleges f_a -ra átszámolhatók az eredmények.

A $p = \dot{\phi}$ időderiváltat felvesszük új változóként, és áttérünk a kanonikus egyenletekre⁷. A folytonos mezőre a Hamilton-sűrűségfüggvény:

$$\mathcal{H}_\phi = |p|^2 + \left| \frac{\partial \phi}{\partial x} \right|^2 + \left| \frac{\partial \phi}{\partial y} \right|^2 + \lambda (|\phi|^2 - 1)^2 \quad (7)$$

Egyenletes rács mellett a triviális diszkretizációval a hely szerinti deriváltat a megfelelő oldali szomszéd mezőértékkel való különbségből lehet számolni. Legyen a rácsban a szomszédos pontok közötti távolság $2d$. Így egy pontban a diszkretizált Hamilton-sűrűségfüggvény:

$$\begin{aligned} \mathcal{H}(x, y) = & |p(x, y)|^2 + \left| \frac{\phi(x + 2d, y) - \phi(x, y)}{2d} \right|^2 \\ & + \left| \frac{\phi(x, y + 2d) - \phi(x, y)}{2d} \right|^2 + \lambda (|\phi(x, y)|^2 - 1)^2 \end{aligned} \quad (8)$$

Feltehetjük, hogy ez a sűrűség az egyes pontok körül felvehető $(2d)^2$ területű képzetbeli négyzeten belül állandó. Így a teljes, "felintegrált" Hamilton-függvény:⁸:

⁷Ugyan p -vel jelölöm, de ez nem az általánosított impulzus, az ehhez képest egy területszorzóval rendelkezik. Mégis ebben fejezzük ki az egyenleteket, mert a szimuláláshoz ez a praktikus változó.

⁸Valójában a vizsgált térrész véges léte miatt nem pont ez a végeredmény. Adott határfeltételek mellett kiszámolható, hogy hogyan módosul ez a kifejezés a szélek mentén.

$$H = \sum_{j=1}^{N_{max}} 4d^2 \left(|p(x_j, y_j)|^2 + \left| \frac{\phi(x_j + 2d, y_j) - \phi(x_j, y_j)}{2d} \right|^2 + \left| \frac{\phi(x_j, y_j + 2d) - \phi(x_j, y_j)}{2d} \right|^2 + \lambda (|\phi(x_j, y_j)|^2 - 1)^2 \right) \quad (9)$$

ahol N_{max} a pontok száma a rácsban.

Ebből kiszámolhatók a mozgásegyenletek. Ezzel a diszkretizációval teljesül a stabil energiamegmaradás, a hiba a leapfrog algoritmusnak megfelelően a dt időköz négyzetével arányos.

Ezzel már készen vagyunk az egyenletes rács diszkretizációjával, de szeretnénk egy cellára is felírni a Hamilton-függvényt. Továbbra is feltételezzük, hogy a (8) egyenletben a kinetikus és a potenciális (a λ -t tartalmazó) tagok az egyes pontok körül állandóak. Ahhoz, hogy az egy cellára vonatkozó Hamilton-függvényben csak a cella sarkaiban lévő mezőértékek szerepeljenek, egy kicsit megváltoztatjuk ezt a feltételezést a hely szerinti deriváltas tagokra. Ezekben a tagokban két szomszédos pont, vagyis mondhatni egy él szerepel, úgyhogy tegyük fel azt, hogy ez a tag az él körül állandó. Minden élre képzeletben ráillesztünk egy olyan négyzetet, aminek az az él egy középvonala, és ezen belül vesszük állandónak a Hamilton-sűrűség megfelelő irányú deriváltas tagját. Külön a függőleges és külön a vízszintes élekre ezek a $(2d)^2$ területű négyzetek átfedés nélkül lefedik a teljes síkot, így ezt a sűrűségeloszlást felösszegezve ugyanazt a Hamilton-függvényt kapjuk, mint amit a korábbi eloszlással számoltunk. Így viszont könnyen felírhatjuk egy cellára a Hamilton-függvényt. A cella sarkait tetszőleges körüljárás szerint 1-től 4-ig indexelve ez:

$$H_{cella} = \sum_{j=1}^4 d^2 \left(|p_j|^2 + \lambda (|\phi_j|^2 - 1)^2 \right) + \frac{1}{2} |\phi_2 - \phi_1|^2 + \frac{1}{2} |\phi_3 - \phi_2|^2 + \frac{1}{2} |\phi_4 - \phi_3|^2 + \frac{1}{2} |\phi_1 - \phi_4|^2 \quad (10)$$

A cellás felírás előnye, hogy ez már egyszerűen alkalmazható adaptív rácsra is. A (10) kifejezés ugyanis adaptív rács esetében is kiszámolható minden cellára, és definiálhatjuk a teljes Hamilton-függvényt ezek összegeként. A mozgásegyenletek ebből is kiszámolhatóak, annyi különbség lesz, hogy más-más cellákból jövő tagok különböző területekkel lesznek beszorozva.

Ez a diszkretizáció adaptív rácsra azonban nem lesz jó. Minden cella felbontásánál és visszafejtésénél ugyanis megváltozik az energia, ami nem csökkenthető az időköz csökkentésével. Az energiamegmaradás így instabil és igen hamar elszáll az energia. Adaptív rács esetében tehát óvatosabban kell megválasztani a diszkretizációt.

4.2. Szempontok a diszkretizáció kiválasztásához

Amint láttuk, a diszkretizációhoz kell, hogy tartozzon egy olyan interpoláció, amivel az új pontok berakása esetén az energia jól (ideális esetben egzaktul) megmarad, ez az első számú szempont a megfelelő diszkretizáció felírásához.

Amikor ez teljesül, még akkor is lehet azonban instabil az energiamegmaradás. Ez egyrészt azért történhet, mert a középső és a szélső pontok eredendően mások. A középső pontokra ugyanúgy igaz az a stabilitás, ami az egyenletes rács pontjaira is teljesül. Szélső pontok körül ellenben nincsen szimmetrikusan minden irányban információ a mezőről. Így viszont máshogyan fognak mozogni az ottani mezőértékek, tapasztalat szerint nagy amplitúdóval oszcillálva a szomszédos pontokban lévő mezőértékekhez képest. Ez azt is jelenti, hogy a szélső pontok közelében nagyon más az energiaeoszlás, mint amennyit csak a szomszédos középső pontok mezőértékei alapján számolnánk ki. Emiatt a cellák visszafejtése nagy hibát fog okozni. A cellák visszafejtése információvesztéssel jár, így egzakt energiamegmaradás nem garantálható, de törekednünk kell az ebből származó hiba csökkentésére.

A leapfrog algoritmus, amíg nincsen rácsváltoztatás, dt^2 -tel arányos hibát okoz adott idő alatt, azonban semmi sem szabja meg, hogy a konstans szorzó mennyi legyen. Az instabilitás egy másik forrása az lehet, hogy a rácsváltoztatás során egy konstans hiba bejövetele mellett ez a szorzó is megváltozhat. Annak, hogy mennyi információt veszítünk a cellák visszafejtésénél, erre is lehet hatása. Ezért érdemes bevezetni a 3.4 fejezet végén említett $k > 1$ együtthatót, hogy minél kisebb energiájú, minél simább mezőeloszlású cellákat fejtsünk csak vissza.

Végül egy gyakorlati szempontot is vegyünk figyelembe. A cél az, hogy olyan algoritmust írjunk, amivel minél hatékonyabban szimulálható a mező időbeli változása. Ehhez arra kell törekednünk, hogy a korábbi szempontok mellett az összefüggéseink a lehető legegyszerűbbek, így a lehető legkisebb számításigényűek legyenek, hogy minél nagyobb skálakülönbség (így minél több szükséges adatpont) esetén is még elfogadható legyen a futásidő.

4.3. Az általam használt diszkretizáció

A quadtree algoritmus a teret különböző méretű, diszjunkt (négyzet alakú) cellákra bontja. Tekintsünk egy ilyen cellát, és egyelőre tegyük fel, hogy ennek egyik sarka sem szélső pont. A cella sarkaiban a mezőérték legyen valamilyen körüljárás szerint ϕ_1 , ϕ_2 , ϕ_3 és ϕ_4 , ezek időderíaltjai pedig p_1 , p_2 , p_3 és p_4 . Legyen a négyzet középpontján a mezőérték a sarkokbeliek számtani közepe, illetve legyen az időderíalt abszolútérték-négyzete a sarokbeli abszolútérték-négyzetek számtani közepe:

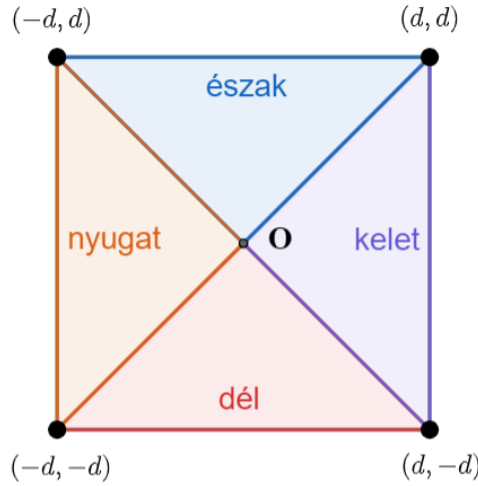
$$\phi_m = \frac{\phi_1 + \phi_2 + \phi_3 + \phi_4}{4} \quad (11)$$

$$|p_m|^2 = \frac{|p_1|^2 + |p_2|^2 + |p_3|^2 + |p_4|^2}{4} \quad (12)$$

Ez természetesen nem definiálja p_m -et, hiszen a komplex argumentuma ez alapján akármennyi lehet, de a számolás további részében erre nem is lesz szükség. Fontos megjegyezni, hogy ezt a pontot nem vesszem fel új változóként, és nem léptetem, csupán a számoláshoz kell.

A cellát a középső pont segítségével négy egybevágó háromszög alakú régióra bontjuk a 3. ábra szerint, ezeket az elhelyezkedésük alapján déli, északi, keleti és nyugati régióknak hívjuk. Legyen az origó a középpontban, és tekintsük a keleti, illetve északi irányokat a pozitívnak, a cella oldalhosszát pedig nevezzük $2d$ -nek. Így a cella sarkai mind a $(\pm d, \pm d)$ koordinátáknál vannak. A háromszög alakú régiók csúcsaiban mért ϕ és $|p|^2$ értékeket ismerjük, a régiókban azt tesszük fel, hogy ezek lineárisan változnak. Így például a déli régióban, ha az 1-es index a délkeleti, a 2-es a délnyugati pontot jelöli:

$$\phi_{dél}(x, y) = \phi_m + \frac{y \cdot (2\phi_m - \phi_1 - \phi_2) + x \cdot (\phi_2 - \phi_1)}{2d} \quad (13)$$



3. ábra. A cellát négy háromszögre bontjuk, amiket az égtájak szerint nevezünk el. A modellünkben a ϕ mezőérték lineárisan változik ezeken belül, a középpontban pedig a négy sarokbeli értékek számtani közepét feltételezzük. Hasonlóan teszünk fel a p időderiváltak abszolútérték-négyzetéről. A középpontot origónak véve a csúcsok koordinátái $(\pm d, \pm d)$, ahol az x koordináta az ábrán jobbra, az y felfelé nő.

$$|p_{dél}|^2(x, y) = |p_m|^2 + \frac{y \cdot (2|p_m|^2 - |p_1|^2 - |p_2|^2) + x \cdot (|p_2|^2 - |p_1|^2)}{2d} \quad (14)$$

Ezek az összefüggések megadják az interpolációt is, amit az új pontok berakásakor azok értékének meghatározásához használnunk kell ahhoz, hogy a mező ne változzon meg, és így az energia is állandó maradjon. Az élek felezőpontjaiban felvett pontokban a mezőértékek tehát a megfelelő két sarokbeli értékek számtani közepe, míg a cella középpontjában a négy sarok számtani közepe. Hasonló mondható el a mező időderiváltjának abszolútérték-négyzetéről is. Az időderiváltak interpolálásához még a fázist is meg kell adni, arról azt tesszük fel, hogy a mezőértékhez hasonlóan lineárisan változik.

A régió járulékat a Hamilton-függvényhez az alábbi integrállal kapjuk:

$$H_{dél} = \iint \mathcal{H}_{dél} = \int_{-d}^0 dy \int_y^{-y} dx \left(|p_{dél}|^2 + \left| \frac{\partial \phi_{dél}}{\partial x} \right|^2 + \left| \frac{\partial \phi_{dél}}{\partial y} \right|^2 + \lambda (|\phi_{dél}|^2 - 1)^2 \right) \quad (15)$$

A többi háromszög régióra ugyanez a számolás hasonlóan elvégezhető, az egy cellára vonatkozó teljes Hamilton-függvény a B. függelékben megtalálható.

Ezt a Hamilton-függvény által megadott energiát (és ezzel együtt a teljes cella, illetve teljes térrész energiáját is) három tagra bonthatjuk. Az időderiváltakat tartalmazó E_{kin} kinetikus tagra, A hely szerinti deriváltakat tartalmazó E_{der} deriváltas tagra, és az (5) potenciálból származó E_{pot} tagra. A felintegrált, kibontott formában az utóbbi kettőt úgy különböztethetjük meg, hogy a deriváltas tagban kiesik a cella területe, a potenciálos tagban pedig nem.

A teljes diszkretizált Hamilton-függvény megadásához a kapott kifejezést az összes cellára összegezni kell, ezt hívjuk a továbbiakban H -nak. A gyorsulások kiszámolásához formálisan tekinthetjük a mezőértéket és annak komplex konjugáltját minden pontban két független mennyiségnek. Így a Hamilton-egyenletekből egy adott k indexű pontra:

$$\dot{\phi}_k = p_k \quad (16)$$

$$\dot{p}_k \cdot A_k = -\frac{\partial H}{\partial \phi_k^*} \quad (17)$$

ahol A egy új változó, ami az aktuális felbontásra és a pontra jellemző. Mivel a tényleges általánosított impulzus helyett a mező időderiváltjára írtuk át a Lagrange-függvényt, ezért ezek valójában nem pont a Hamilton-egyenletek, azonban nagyon hasonlóak, egy-egy konstans szorzó a különbség.

A cellára vonatkozó, függelékben kiírt H_{cella} Hamilton-függvény kényelmes tulaj-

donsága, hogy bármely sarkában értelmezett p szerinti parciális deriváltja $d^2 \cdot p$ lesz, így, ha a k jelű pontra annak a négy cellának, aminek ő a sarka az oldalhosszaik $2d_1$, $2d_2$, $2d_3$ és $2d_4$, akkor

$$A_k = d_1^2 + d_2^2 + d_3^2 + d_4^2 \quad (18)$$

ami természetesen a rács időbeli változása során változhat.

A H_{cella} Hamilton-függvény megadja az egy cellára vonatkozó teljes energiát. Mivel ez egy nagyon hosszú és nagy számításigényű kifejezés, ezért nem célszerű minden egyes léptetés után meghatározni a cellák pontos energiáját a quadtree algoritmus fitness-értékeihez. Ezért inkább egy, az energiát közelítő egyszerű fitness-függvényt használok az alábbi alakban:

$$\begin{aligned} f = & d^2 (|p_1|^2 + |p_2|^2 + |p_3|^2 + |p_4|^2) \\ & + d^2 \lambda \left((|\phi_1|^2 - 1)^2 + (|\phi_2|^2 - 1)^2 + (|\phi_3|^2 - 1)^2 + (|\phi_4|^2 - 1)^2 \right) \\ & + \frac{1}{4} \left(3 \sum_{j=1}^4 |\phi_j|^2 - \sum_{k=2}^4 \sum_{j=1}^{k-1} (\phi_j \phi_k^* + \phi_k \phi_j^*) \right) \end{aligned} \quad (19)$$

Ez a felírás abból származik, hogy a viszonylag rövid kinetikus, és deriváltas tagja megegyezik az általam használt diszkretizációval egzaktul kiszámolhatóban szereplő ilyen tagokkal. A potenciális tagot azonban egy gyorsabban kiszámolható rövid képletre cseréltem, méghozzá a (az egyenletes ráccsal végzett diszkretizációra vonatkozó) energiaképlet (10) potenciális tagjára.

Természetesen ez az összefüggés különböző körülmények között különböző mértékben közelíti a tényleges energiát. A lényeg azonban az, hogy tartalmaz időderiváltas tagokat, hely szerinti deriváltas tagokat és potenciális tagokat is, így ezek közül, ha bármelyik nagy, akkor az összeg is nagy lesz. Valójában nem fontos, hogy a fitness-függvény éppen az energia legyen, elég, hogyha olyan, hogy a hurok közelében magas az értéke, így ott sűrű a felbontás. Összehasonlítások azt mutatták, hogy ezzel a fitness-függvénnyel, illetve az energiával, mint fitness-függvénnyel végzett adaptív felbontás nagyon hasonló, csupán néhány, relatíve kis energiájú cella felbontottságánál van eltérés. Ezzel az egyszerűsítéssel viszont a futásidő szignifikánsan (akár 40%-kal) csökkenthető, míg a szimuláció pontossága csak nagyon egyenletes és alacsony energiasűrűség esetén csökken, amikor már a pontos energiát használva is nagyobb hibákat kapnánk (a hibák nagyságáról később lesz szó).

4.4. A szélső pontok rögzítése

Ha figyelembe vesszük a szélső pontokat, akkor látszólag csak annyit kell változtatnunk az előbbieken, hogy (18) képlet kiszámolásánál csak két cella hossza fog szerepelni. Azonban, ha ezt tesszük, akkor a szélső pontok mozgásegyenletében csak tőlük egyik oldalra lévő pontok szerepelnek, így "nem látják", ami tőlük a másik oldalra van, és nem megfelelően mozognak. Figyelembe lehet venni egy ilyen szélső pontot a szomszédos cellában az ottani a pontot érintő háromszög alakú régiót elfelezve, oly módon, hogy mindkét új háromszögnek csúcsa legyen ez a szélső pont, és ezekre felintegrálni a Hamilton-sűrűségfüggvényt. Így már szerepelni fog a szélső pontoktól minden irányban lévő pont, de itt is igaz lesz, hogy az egyik oldalról más súllyal szerepelnek a közeli pontok, mint a másik oldalról. Ilyenkor viszont sosem úgy fognak fejlődni a szélső pontok, mint a szomszédos középső pontok. Ennek eredményeként a mező fejlődése során ezen pontok nagy amplitúdóval fognak oszcillálni, és ez az energia instabilitásához vezet. Így minden új szélső pont egyre növeli az időegységenkénti hibát.

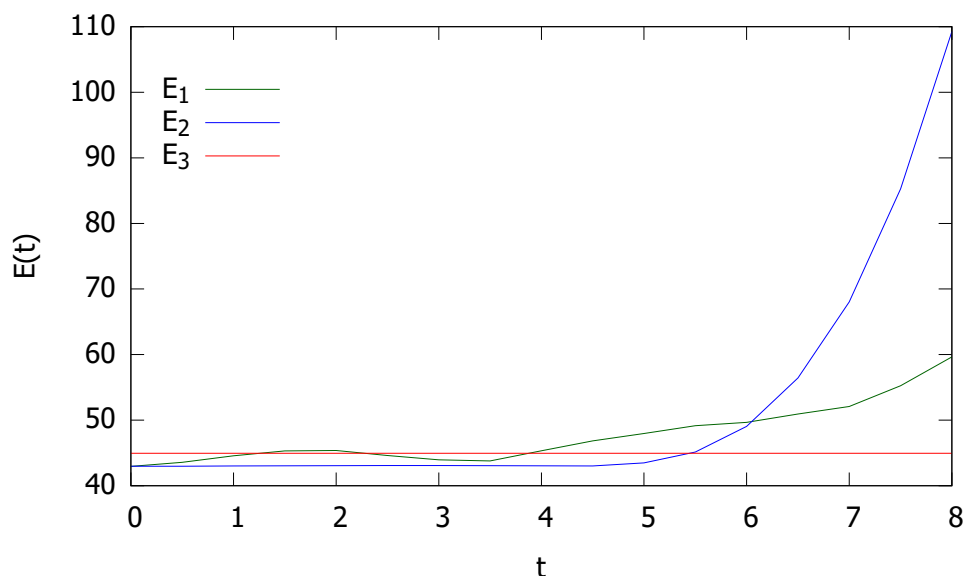
Emiatt más megközelítést használok: a szélső pontokat nem tekintem független változóknak. Tekintsünk egy szélső pontot, illetve azt a két szomszédját, amik között húzódó él felezőpontján van. Az ezekhez tartozó mezőértékek legyenek ϕ_{sz} , ϕ_a és ϕ_b , ezek időderiváltjai pedig p_{sz} , p_a és p_b . Rögzítsük a szélső ponthoz tartozó adatokat a szomszédjaihoz az alábbi módon:

$$\phi_{sz} = \frac{\phi_a + \phi_b}{2} \quad (20)$$

$$\begin{aligned} |p_{sz}|^2 &= \frac{|p_a|^2 + |p_b|^2}{2} \\ \arg p_{sz} &= \frac{\arg p_a + \arg p_b}{2} \end{aligned} \quad (21)$$

Ezek az összefüggések összhangban vannak (13) és (14) interpolációkkal, de most egzaktul megadtuk a mező időderiváltját az extra egyenlettel. A (20) és (21) azonosságokat a szimuláció során minden időpillanatban kikényszerítjük, és így a szélső pontokat fix pontoknak is nevezhetjük. Természetesen a mezőértékek lerögzítésének következtében azok már nem független változók, így azon cellákra, amikre legalább az egyik sarok fix⁹ módosulni fog a Hamilton-függvény, méghozzá úgy, hogy a fix pontok helyére a fenti kifejezéseket kell helyettesíteni. Ily módon egy cellának a Hamilton-függvényébe bekerülhet olyan pont is, ami nem a cella sarka (ϕ_a és ϕ_b egyike ilyen). Ezért (18) összefüggés, és így a mozgásegyenlet is módosulhat a szélső pontok közelében, méghozzá a következő módon: ha egy szélső pontnak az a illetve b indexű pontoktól való távolsága

⁹egy cellának csak nulla, egy, vagy két sarka lehet fix, több nem, hiszen úgy végeztük a cellák felbontását, hogy két szomszédos pont sosem szélső pont, így nem is fix.



4. ábra. Egy egyszerű folyamat során az összenergia változása különböző módon diszkretizált mezőkkel. E_1 a 4.1 fejezet végén leírt diszkretizációt használó szimuláció energiája. E_2 és E_3 ugyanez a 4.3 fejezetben leírt diszkretizációra rendre a szélső pontok rögzítése nélkül, illetve azok rögzítésével. Az első kettő esetben 2.5-ször annyi pontot engedtem lerakni, mint a harmadikban, hogy egy kicsit tovább maradjanak stabilak, de még ez sem segít sokat. Ez a mozgás ugyanezekkel a kezdőfeltételekkel a 6. és 7. ábrákon kerül részletes bemutatásra.

d , akkor ehhez a két ponthoz tartozó (18) egyenlet jobb oldalához hozzáadódik $0.5 \cdot d^2$. Emiatt a gyorsulások számolásánál esetszétválasztást kell elvégezni.

Nézzük meg, hogy mennyire jó ez a diszkretizáció a többi említettthez képest. A dolgozat során egy húr és egy antihúr frontális ütközését fogom vizsgálni. A 4. ábrán egy ilyen folyamatra ábrázoltam az összenergia megváltozását a 4.1 fejezet végén leírt (adaptív) diszkretizáció használatával, a 4.3 fejezetben leírt diszkretizációval a szélső pontok lerögzítése nélkül, majd végül ugyanezzel, de a szélső pontok lerögzítése mellett. Amint látható, az első két diszkretizáció nagyon instabil, és hamar elszáll velük az összenergia, annak ellenére is, hogy ezeknél 2.5-ször annyi adatpontot engedtem meg, mint a harmadiknál. Eközben a harmadik, a dolgozatban használt diszkretizáció stabilan energiatartó (ez alatt az idő alatt az energia relatív megváltozása egy század százalék alatt van). Az ábrázolt folyamat esetében legalább $t \approx 30$ -ig (kicsivel az ütközés utánig) mindenképpen el kell futtatni a szimulációt, az tehát elfogadhatatlan, hogyha az energia csak $t \approx 5$ -ig marad meg. Ez a tény szükségessé teszi a bonyolult diszkretizáció használatát a nagy számításigény árán is.

5. Axionmező szimulálása

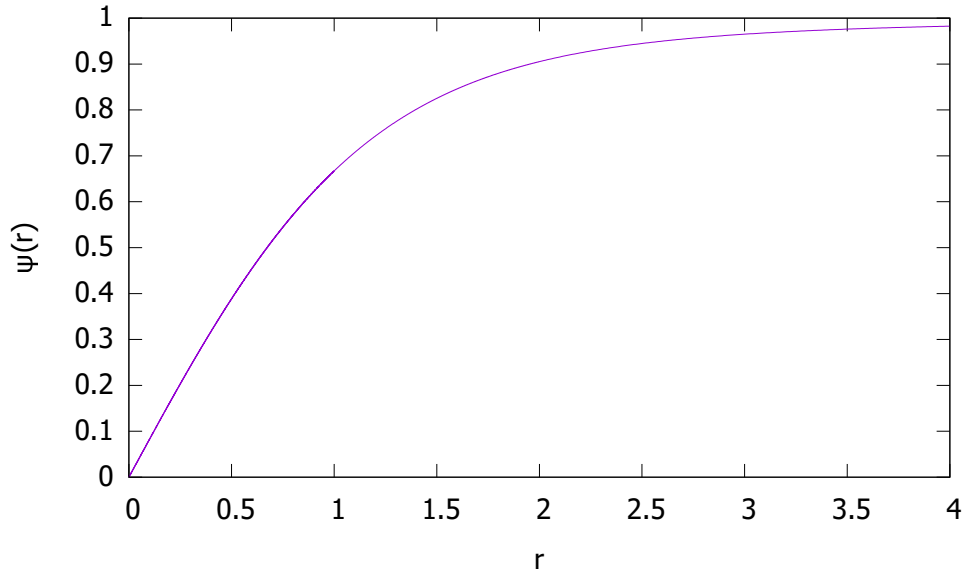
5.1. Kezdőfeltétel

Kettő dimenzióban egy húr egy pontszerű képződmény a lokális potenciálmaximumon ($\phi = 0$), aminek a környezetében gyorsan lecseng a potenciál és a mezőértékek abszolútértéke távol f_a -hoz konvergál. Feltételezhetünk forgásszimmetriát, így a (nem diszkretizált) mozgásegyenletekből megkapható a rendszer "álló húr" megoldása. Ehhez polárkoordinátákkal írtuk fel a mezőt a centrum körül, és használtuk a $\phi_h(r, \varphi) = \psi(r) e^{i\varphi}$ ansatzot, ahol $\psi(r)$ egy valós függvény. Ekkor a

$$\psi''(r) = 2\lambda\psi(r) (\psi^2(r) - f_a^2) + \frac{\psi(r)}{r^2} - \frac{\psi'(r)}{r} \quad (22)$$

differenciálegyenlet teljesül, amely numerikusan megoldható. A peremfeltételek $\lim_{r \rightarrow 0} \psi(r) = 0$ és $\lim_{r \rightarrow \infty} \psi(r) = f_a$. $\lambda = 1$ paraméter mellett az f_a egységében mért $\psi(r)$ függvény az 5. ábrán látható. A továbbiakban a mezőérték mindig f_a egységében értendő, és a λ paraméter értéke mindenhol 1. A húrhoz rendelhetünk egy karakterisztikus sugarat, amin belül a hatása még jól érzékelhető. Az ábra alapján ezt $r_h = 3$ -nak választottam.

Két dimenzióban egy húrhoz megfeleltethetünk egy töltést, ami azt határozza meg, hogy a húr pozitív körüljárása esetén (előjelesen) hányszor 2π fázisugrás történik. Ha csak ± 1 "töltésű" húrokkal foglalkozunk, akkor megkülönböztethetünk egyetlen pozitív



5. ábra. A $\phi_h(r, \varphi) = \psi(r) e^{i\varphi}$ ansatz-cal kapott húrmegoldáshoz tartozó $\psi(r)$ függvény. A húr "mérete" egy nem egzaktul definiált mennyiség, azt a sugarat jelenti, amilyen távol a centrumtól már "közel" 1 a mező abszolútértéke, az ábra alapján ezt $r_h = 3$ -nak választottam.

és egyetlen negatív "töltésű" hűrt, amiket hívhatunk rendre húrnak és antihúrnak [10]:

$$\begin{aligned}\phi_h^+(r, \varphi) &= \psi(r) e^{i\varphi} \\ \phi_h^-(r, \varphi) &= \psi(r) e^{-i\varphi}\end{aligned}$$

Gyakran a pontosan kiszámolt kezdőfeltétel helyett egy kellően kisimított véletlenszerű kezdő-feltételt vizsgálnak [30]. Ez a dolgozat azonban szimulációm bemutatásáról, és más szimulációval való összehasonlításáról szól, így célszerű megismételhető, és ismert rendszert szimulálnunk. Ezért a dolgozatban kettő azonos nagyságú, de ellentétes töltéssel rendelkező húr ütközését vizsgálom. Írjuk át a húr körül kiszámolt mezőértéket helyvektortól függő alakba, vagyis legyen $\phi_h(\mathbf{r}) = \phi(r, \varphi)$. Így a kezdőfeltétel kettő, $\mathbf{R}_0$ és $-\mathbf{R}_0$ helyen lévő, ellentétes töltésű húr esetén:

$$\phi_0(\mathbf{r}) = \phi_h^+(\mathbf{r} - \mathbf{R}_0) \cdot \phi_h^-(\mathbf{r} + \mathbf{R}_0) \quad (23)$$

Ha két ilyen húr távolról hat egymásra akkor az analóg az elektromos töltések viselkedésével két dimenzióban [9], így ellentétes "töltésű" hűrok távolról is vonzzák egymást. A számítógép azonban csak adott pontosságig tudja kezelni a mezőértéket, ezért túl távoli hűrok esetén a szimulációban nem érzékelnek egymás jelenlétét. Ezért nagyobb távolság esetén muszáj kezdősebességet is adni a hűroknak. Ez úgy tehető meg, hogy egy húr közelében (ameddig már a számbábrázolás pontossága állandó abszolút értékű mezőt lát) az alábbi kezdeti momentumot adjuk:

$$p_h(\mathbf{r}) = \dot{\phi}_h(\mathbf{r}) = \frac{B}{dt} (\phi_h(\mathbf{r} + d\mathbf{r}) - \phi_h(\mathbf{r})) \quad (24)$$

ahol B egy paraméter, dt a szimuláció időlépése, és egy léptetés alatt a húr (és környezete) $d\mathbf{r}$ vektorral mozog odébb. A másik húr közelében $d\mathbf{r}$ helyett $-d\mathbf{r}$ -t rakva érhetjük el, hogy egymás felé menjenek.

Az általam végzett szimulációban egy négyzet alakú véges V területet vizsgálok, az origó pedig ennek a középpontja. $\mathbf{R}_0$ -t úgy választom, hogy a két húr az egyik nagyátlón legyen, $d\mathbf{r}$ -t pedig úgy, hogy ezek egymás felé induljanak el. Ez a durván kiszabott kezdősebesség csak akkor szimulálja jól egy húr haladását (és akkor sem tökéletesen), ha a rács felbontása nagyon jó. Ez azért van, mert a nem folytonos felbontás mellett az "álló húr" megoldás nem egzakt, így kis zavarok lesznek az időderiváltakban. Ahhoz, hogy ezek hatása ne legyen túl nagy, az időléptetést igen kicsire kell venni. Ekkor viszont az időderivált értékek nagyon nagyok lesznek, ami ezeknek a hibáját hangsúlyozza ki. Ezért van szükség a B paraméterre, így kis dt -vel és kis B értékek mellett alacsonyan tarthatók az időderiváltak. Mivel $A \neq 1$, ezért nem tökéletesen tolódik el a húr egy léptetés alatt, így $d\mathbf{r}$ nem lehet túl nagy.

Célszerű bevezetni B helyett a $v = B|\mathbf{dr}|/dt$ mennyiséget, ami a húr centrumának sebessége lesz. Mivel $c = 1$, ezért természetesen $v < 1$. Ha v túl nagy, akkor a kezdőfeltételnél a Lorentz-kontrakciót is figyelembe kell venni. Ezt elkerülendő, kisebb sebességeket vizsgáltam.

A véges térfogat miatt be kell vezetni egy határfeltételt is. Erre a választott húrgeometria miatt célszerű választás az alábbi, "antiperiodikus konjugált" határfeltétel:

$$\phi(\mathbf{r}) = \phi^*(-\mathbf{r}), \quad \text{ha } \mathbf{r} \in \partial V \quad (25)$$

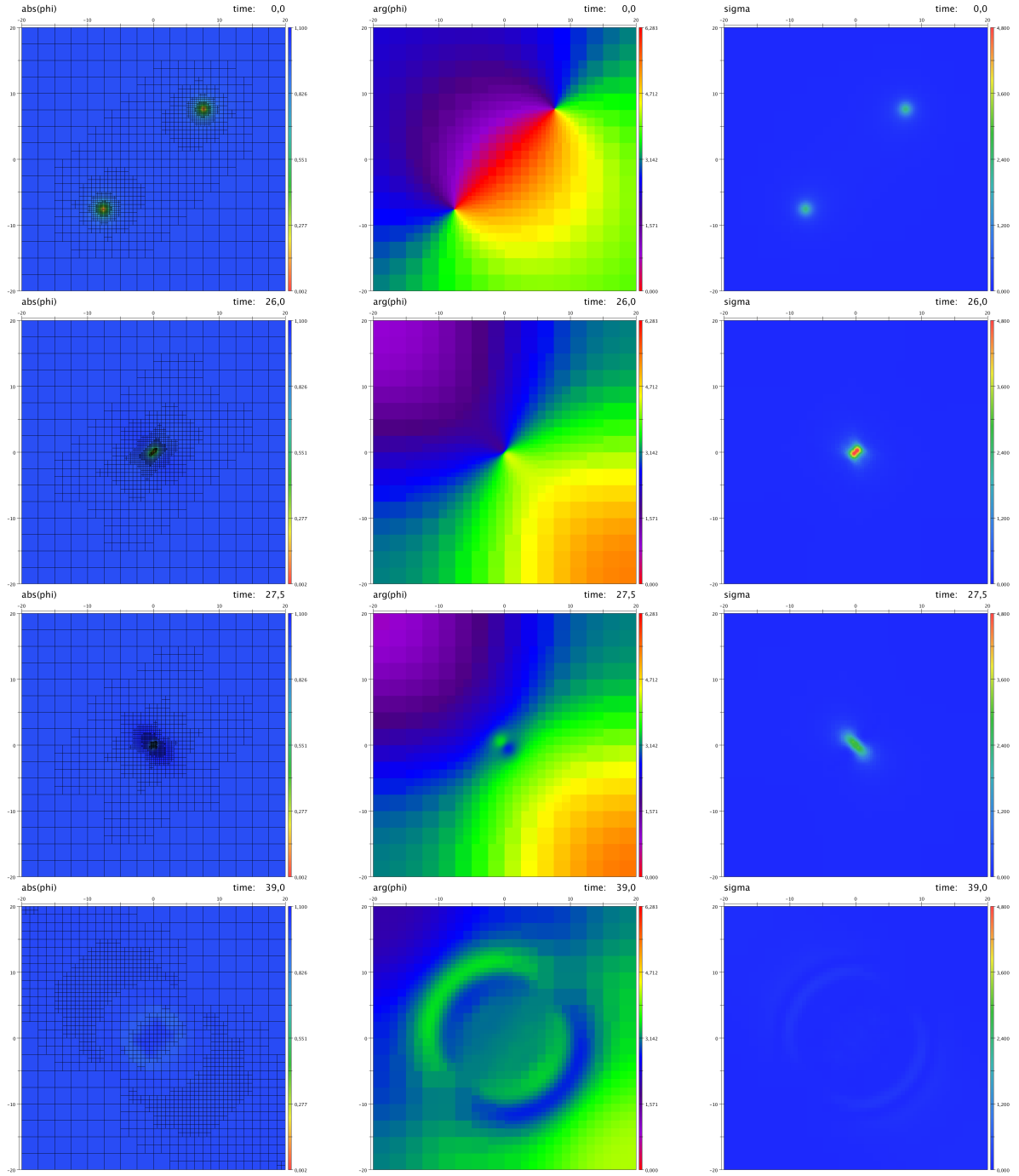
Mivel itt az origóra középpontosan szimmetrikus a vizsgált terület (egy $L \cdot L$ területű négyzet), ezért $\mathbf{r}$ és $-\mathbf{r}$ mindig a négyzet két ellentétes határpontjára mutatnak.

5.2. Frontális ütközés lefolyása

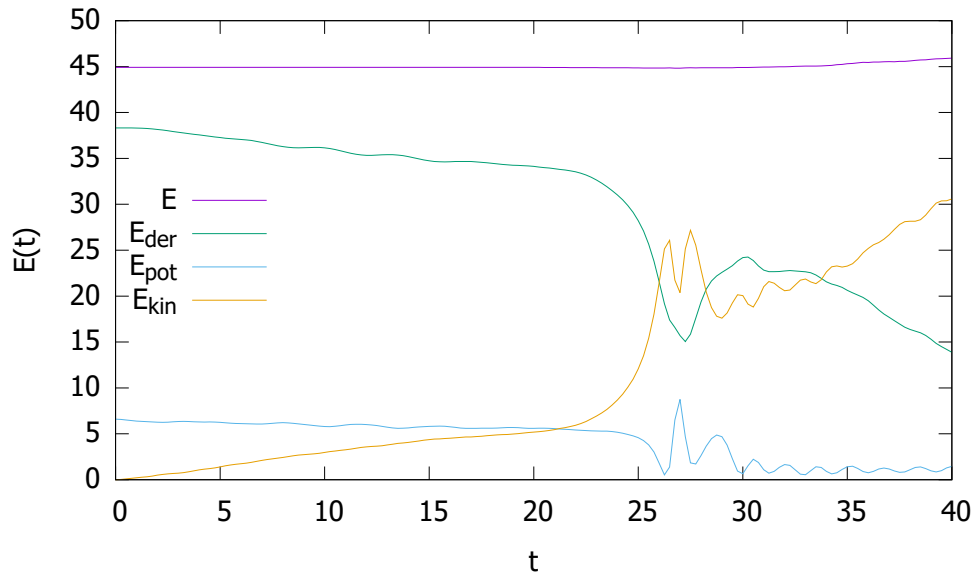
Minden általam végzett szimulációban $f_a = \lambda = 1$. A hűrt és az antihűrt egymástól mindig $2|\mathbf{R}_0| = 0.38 \cdot L$ távolságból indítom (L a vizsgált terület oldalhossza), méghozzá úgy, hogy a vizsgált négyzet alakú terület egyik nagyátlóján, a középpontra szimmetrikusan helyezkednek el. A dolgozatban főként frontális ütközéseket vizsgállok, így, ha más nincs jelezve, $\mathbf{dr} = (dr_x, dr_y) = (1, 1)$. A v sebességparaméter a kezdősebesség nélküli esetben zérus, bármely egyéb esetben $v = 0.25$ értékre veszem fel. Az egyes szimulációkban szabad paraméterek tehát a vizsgált terület L oldalhossza, a dt időköz, és a felbontás. A felbontást az egyenletes rács esetében egyetlen N számmal jellemzem, ami egyetlen sorban lévő pontok száma (összes pontok száma $N \cdot N$). Adaptív rács esetén a felbontást $N_{max} = N_0 \cdot N_0 + N_{extra}$ alakban adom meg, vagyis, ha egy $16 \cdot 16$ -os durva felbontással indultam, majd 2000 extra pontot helyezek be az első quadtree algoritmussal, akkor azt úgy jelölöm, hogy $N_{max} = 16 \cdot 16 + 2000$.

Először kezdősebesség nélkül, viszonylag közletről elindított húr és antihúr ütközését néztem. A kezdőfeltétel szerint a húrok távolságának és méretének aránya $\eta = 2|\mathbf{R}_0|/r_h = 8.6$. A rendszer fejlődését a 6. ábra mutatja három különböző nézetben.

Az első sor a kezdőfeltételt mutatja. A bal oldalon a mező abszolút értékét láthatjuk, az első ábrán megbizonyosodhatunk arról, hogy az η arányszám tényleg jól jellemzi a rendszert. A középső oszlop a $\Theta = \arg \phi$ értéket ábrázolja, megfigyelhetjük, hogy a húrok körül valóban megjelenik a $\pm 2\pi$ fázisugrás. Jobb oldalon a mező σ energiasűrűsége van felrajzolva, ami minden cellára a benne lévő összenergia és a területének a hányadosa. Valóban teljesül, hogy a húrok közvetlen közelében magas, távol pedig közel zérus az energiasűrűség. $t = 26$ időegység eltelése után a két húr összeütközik. Látható, ahogy megszűnik a két fázisugrás, az energia pedig egy pontban sűrűsödik össze. Ezután az energia szétsugárzik, a legnagyobb intenzitással az ütközésre merőleges irányban. A középső oszlopban be van rajzolva az adaptív rács felbontása is. Megfigyelhető, hogy



6. ábra. Két kezdősebesség nélkül indított közeli húr ütközése 2 dimenzióban. A bal oldalon a ϕ mezőérték abszolútértéke, középen az argumentuma, jobb oldalon az energiasűrűség. A bal oszlopban a rács felbontása is jelölve van. A színekódok minden kisábra jobb oldalán látható, a méretskála pedig az ábrák bal és felső oldalán, minden a szöveg szerinti dimenziótlanított egységekben. A szimuláció paramétereit: $L = 40$, $dt = 0.05$, $N_{max} = 16 \cdot 16 + 2000$ (további információk a 5.2 fejezet első bekezdésében).



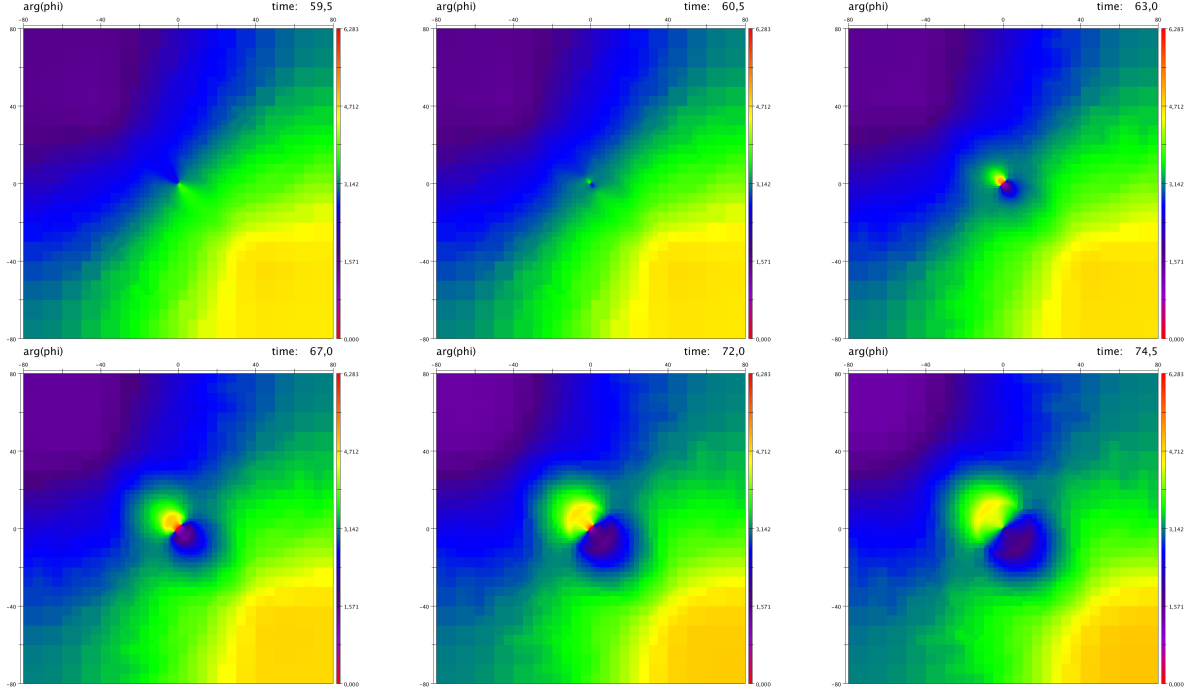
7. ábra. Az összenergia és az energia egyes komponenseinek alakulása a 6. ábrán mutatott szimuláció során.

követi a húrokat, majd az energiakisugárzását is. A látványosság kedvéért a szükségesnél sokkal jobban felbontottam a teret, de a hasonló energiatartáshoz harmadennyi rácspontra is elegendő.

A folyamat ugyanígy néz ki az η paramétertől függetlenül, az egyetlen különbség, hogy a bal és jobb oldali oszlopban kisebb pöttyöknek látszódnak a húrok. Emiatt a középső ábrán a rácsfelbontás is sűrűbb kell, hogy legyen a húroknál, így több pontra van szükség.

Az energia különböző komponenseinek időbeli változását megtekinthetjük a 7. ábrán. Mivel nem volt kezdősebesség, ezért a kinetikus tag nulláról indul, majd ahogy egyre jobban vonzza egymást a két húr és egyre gyorsabban haladnak egymás felé, elkezd növekedni. A potenciális tag alig változik, ahogy közeledik a két húr, így ez a növekedés a deriváltas tag kárára történik. Az ütközés pillanatában a potenciális tag szinte nullára csökken, a kinetikus pedig nagyon megnő. Itt megfigyelhetünk egy érdekességet, ugyanis ez egymás után kétszer megtörténik, mintha kettőször is bekövetkezne az ütközés, erre később visszatérünk. Ezután az energia szétsugárzódik, a mező egyre egyenletesebbé válik, így a deriváltas és potenciális energia egyre csökken, és az energia nagy része már kinetikus.

Megfigyelhetjük, hogy az összenergia egy kicsit megváltozott a folyamat során. Ez azért van, mert amikor már egyenletes az energiaeloszlás, akkor semmi előny nem származik az adaptív rácsból, a hátrányai viszont mind kihangsúlyozódnak. $t = 31$ -ig, ameddig már minden ütközés körüli esemény befejeződött, és megkezdődött az energia szétsugárzása, de az még nem *túl* egyenletes, az energia hibája mindössze 0.1%.



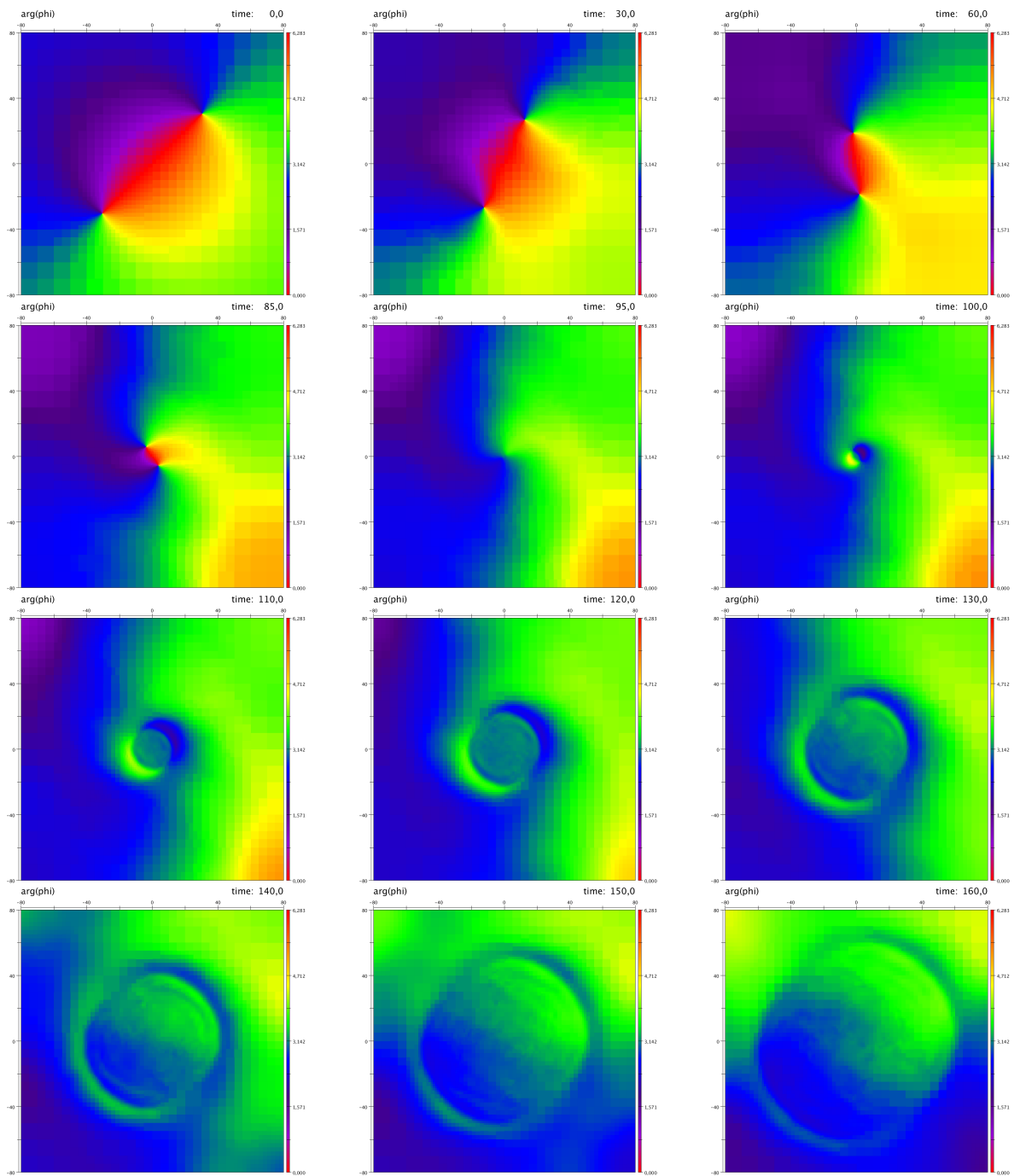
8. ábra. Az ütközést követő jelenség, a "többszörös annihiláció". Az idő balról jobbra majd fentről lefelé telik. Az ütközést követően újra megjelenik két húr, egy kicsit távolodnak, majd elkezdnek közeledni egymáshoz, míg végül újfent annihilálódnak. A szimuláció paraméterei: $L = 160$, $dt = 0.025$ $N_{max} = 16 \cdot 16 + 5000$.

Az ütközés környékén történő különös viselkedés, a többszörös annihiláció kevésbé vizsgált jelenség, az irodalomban csak 2020-ból származik róla említés, egy orosz szakdolgozatban [11]. A név beszédes, itt valóban az történik, hogy az annihiláció egymás után kis idővel kétszer is végbemegy. A jelenség látványosabb nagyobb sebességű ütközés esetén, ezért egy kezdősebességgel, távolabbról elindított húr-antihúr pár esetén mutatom be a 8. ábrán. A legelső ábra mutatja, hogy a fázisugrások eltűnnek az ütközés pillanatában, de a harmadik ábrán már egyértelműen kivehetően újra megjelentek. A két húr egyre lassabban távolodik, majd újra annihilálódik.

5.3. Nem frontális ütközés

Nem frontális ütközés esetén a hurok "bespiráloznak" középre, és úgy ütköznek össze. Egy ilyen folyamat időbeli lefolyása az $\arg \phi$ térben a 9. ábrán látható. A két húr itt messzebből indítottam, és $\mathbf{dr} = (dr_x, dr_y) = (1, 0.1)$ eltolásvektorral állítottam be a kezdősebességet. Az ábrán jól megfigyelhető egyfajta forgás, amit a hurok mozgása okoz. Ez még látványosabb lenne olyan húr-antihúr párra, amik az ütközés előtt akár többször is megkerülik egymást a "spirálozás" során. Ehhez azonban nagy kezdősebesség szükséges, és a 5.1 fejezetben leírt módszerrel csak kis, sebességet lehet adni.

A második ábrán megfigyelhető egy kis zaj a hurok kezdőhelyének közelében, ami a



9. ábra. Nem frontális ütközés esetén a mező argumentumának fejlődése. Az idő balról jobbra majd fentről lefelé telik. A szimuláció paramétereit: $L = 160$, $dt = 0.025$, $N_{max} = 16 \cdot 16 + 5000$, $\mathbf{dr} = (1, 0.1)$.

következő ábrákon is egy kis zavart okoz a forgó "csápok" alakjában. Ez a kezdősebesség korábban leírt tökéletlenségei miatt történik, a kezdősebességgel indított frontális ütközésnél szintén megtalálható. Ennek a zavarnak a hibára való hatására a következő fejezetben térünk ki.

6. Adaptív és egyenletes rács összehasonlítása

6.1. A leggyorsabb futáshoz szükséges paraméterek meghatározása

Azt szeretnénk meghatározni, hogy mennyivel gyorsabb (ha egyáltalán) az adaptív rácsú szimuláció az egyenletes rácsúnál. A futási idő azonban függ a felvett pontok számától (felbontástól) és az időköztől is, emellett ezeknek a függvényében a szimuláció pontossága is változik. Az igazságos összehasonlításhoz egy adott rendszert mindkét szimulációval a lehető leggyorsabban kell lefuttatni, amivel a szimuláció még egy előre meghatározott mértékig pontos.

Az adaptív rács esetében ez igen egyszerű. A dt időköz, ha elég kicsi, akkor nem változtat a pontosságon (összenergia időbeli megváltozásán), mert a hiba első számú forrása a cellák felbontásának visszafejtése lesz. Egy adott kezdőfeltételből indulva egy kis dt mellett lefuttattam a szimulációt különböző N_{max} értékekre (N_{max} az eltárolt pontok száma). Ezután kiválasztottam azt az N_{max} értéket, amire már kellően pontos az eredmény, majd dt -t addig növeltem, amíg ez a pontosság első rendben nem változik. A "kellő pontosságot" 3%-nak választottam meg, a folyamat teljes lefolyása alatt ezen belül kell, hogy maradjon a hiba.

Egyenletes rácsnál a pontossághoz nem használható csupán az energiamegmaradás, hiszen az a leapfrog integrálás miatt szinte tökéletesen teljesül, még olyankor is, amikor a felbontás elégtelensége miatt a szimuláció helytelen eredményt ad. Ezért egy szimulálás pontosságát úgy teszteltem, hogy megnéztem ugyanazt a folyamatot egy pontosabb felbontásra, és, amennyiben ugyanolyan karakterisztikát mutatnak (pl. az ütközés ugyanannyi idő alatt következik be), akkor elégséges volt a rács felbontása. Így kiválasztható a szimuláció felbontása, amit itt a maximális pontok száma helyett annak gyökével, a rács egy sorában szereplő pontok N számával jellemezek: $N_{max} = N \cdot N$. A szimuláció pontosságának térbeli felbontástól való függését a 10. ábra szemlélteti. Itt különböző L -ek mellett ábrázoltam az ütközésig eltelt T időt $N = 200, 400, 600, 800$ és 1000 mellett. Egyre nagyobb skálakülönbség mellett egyre rosszabb eredményt adnak a kisebb rácsok, az $N = 200$ -as ráccsal nem is történik már ütközés $L \geq 250$ -re.

Ha minden N -re megnézzük azt a legnagyobb L -et, amire a szimuláció még kellően pontos (vagyis az ütközési idő ugyanannyi, mint nagyobb rácsokkal) akkor egy egyszerű összefüggést kapunk:

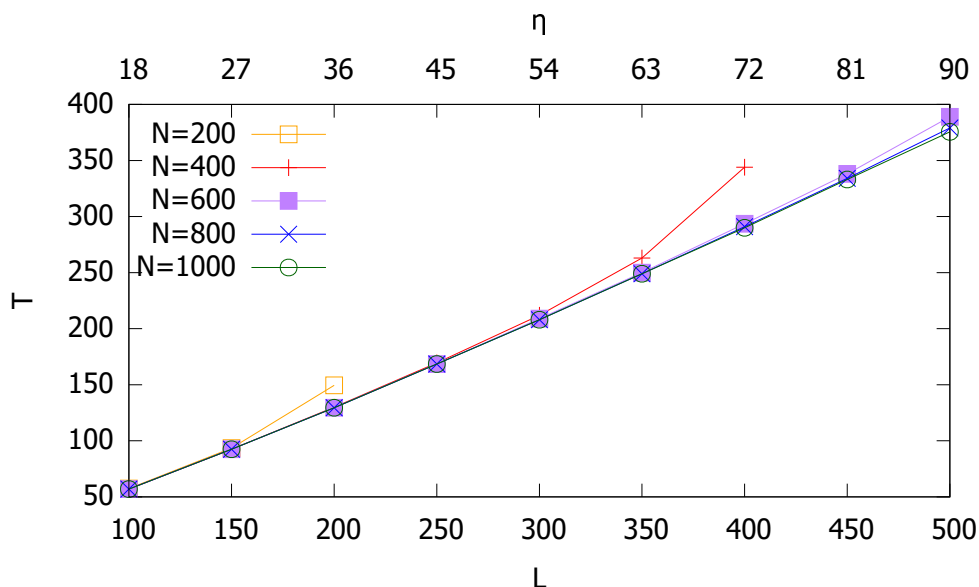
$$L_{max}(N) = N/2 \quad (26)$$

L_{max} és N között lineáris összefüggés van, tehát egy cella maximum vastagsága $dx = 0.5$ lehet a hurok egymástól mért távolságától függetlenül. A hurok méretét (ami

λ -val állítható) nem változtattam, vagyis egy hűrt körülvevő $2r_h \cdot 2r_h = 6 \cdot 6$ területű négyzetet mindig legalább $12 \cdot 12$ -es felbontással kell vizsgálni.

Ugyan az energia jól megmarad, dt nem lehet tetszőlegesen nagy. A rácsállandótól függően egy adott értéke felett a szimuláció instabillá válik¹⁰. Ezért dt -t lassan növelve kerestem annak legnagyobb lehetséges értékét. Ez alapján az instabilitási határ ($\lambda = 1$ és $dx = 0.5$ esetén) $dt = 0.32$ volt, ez még stabil, ennél nagyobb már nem. Azonban az épphogy csak stabil dt mellett nagyobb hibákat kapunk, úgyhogy $dt = 0.2$ -t használtam a pontos eredményért. Ezzel a legnagyobb relatív kilengések az összenergia értékében 1-2% körül voltak.

Ugyanígy adaptív rács esetében is igaz, hogy a szükséges legkisebb cella mérete L -től független, így ott is választható egy állandó dt . Kísérleti úton ez $dt = 0.075$. Ez azért kisebb az egyenletes rács időlépésénél, mert a rosszabb stabilitás miatt nagyobb felbontásig kell elmenni hasonló pontosság eléréséhez.



10. ábra. Az ütközésig szükséges idő L függvényében különböző felbontású egyenletes rácsokkal szimulálva. Az ábra tetején az η skáláját is jelöltük, ami csak egy konstans szorzófaktorral tér el L -étől.

¹⁰Több stabilitási feltétel is létezik, amik különböző differenciálegyenletek esetén teljesülnek, az egyik legáltalánosabb a CFL-kritérium [31], ami az információterjedés sebességét korlátozza. Gyakorlatban dt korlátját kísérletileg lehet meghatározni, hiszen az instabil szimuláció hamar elszáll és fizikailag triviálisan helytelen eredményt ír ki.

6.2. A komplexitás elméleti meghatározása

Informatikában egy algoritmus komplexitása azt jelenti, hogy hogyan függ az erőforrásigénye (leggyakrabban számításigénye vagy futásideje) az adathalmaz nagyságát jellemző paramétertől. A mi esetünkben azt akarjuk tudni, hogy L paraméter milyen függvénye szerint változik a futásidő. Ez a nézet azért hasznos, mert így meg tudjuk határozni, hogy nagy skálakülönbségek esetén milyen szimuláció lesz belátható időn belül lefuttatható.

Egyenletes rács esetében a szimuláció komplexitásának elméleti meghatározása igen egyszerű. A (26) összefüggés szerint a felvett pontok $N_{max} = N^2$ száma négyzetesen függ L -től. Az ütközésig fogunk szimulálni, és az addig eltelt (szimulációbeli) idő nagyjából lineárisan változik L -lel, lásd a 10. ábra alsó határgörbéjén. A pontok léptetését minden időpontban és minden pontra el kell végezni, vagyis a számításigény komplexitása $\mathcal{O}(L^3)$. A futásidő a számításigénnyel arányos, így erre is ugyanez a függés várható. Ez egy igen gyors növekedés, ez teszi lehetetlenné az algoritmus használatát nagy skálakülönbségre (különösen, hogy 3D-ben már $\mathcal{O}(L^4)$ a függés).

Az adaptív rács komplexitását nehezebb meghatározni, mert függ a mező jellegétől is. Tökéletesen egyenletes energiaszórás (pl vákuum) esetén az adaptív algoritmus ideális esetben éppen egyenletes rácsot fog adni. (Az ideális eset itt azt jelenti, hogy N_{max} ehhez éppen megfelelő, ha például eggyel több lenne, akkor az egyenletes rácsra képest még egy cellát fel fog bontani.) Az egyenletes rácsra pedig az imént számoltuk ki az L^2 -es térbeli komplexitást. Az általam vizsgált rendszerben azonban nem ez a helyzet. A hurok száma és mérete állandó, így ugyanakkora nagyságú legkisebb négyzetekig kell lemenni a kellően jó felbontáshoz, L -től függetlenül. Az egyetlen különbség, hogy nagyobb terület esetén többször kell negyedelni ennek eléréséhez, méghozzá $\log_4 L^2$ -szer. Minden cella felbontásával, vagyis minden negyedeléssel öttel nő a pontok száma. Mivel a konstans szorzókat a komplexitásban nem vesszük figyelembe, így a térbeli felbontás komplexitása $\mathcal{O}(\log L)$, vagyis az idő beleszámolásával együtt a teljes rendszeré $\mathcal{O}(L \log L)$. Ez sokkal lassabban nő, ezért merülhet fel használata nagy skálakülönbség esetén is. Három dimenzióban a hurok már egydimenziósak, így $L \log L$ -lel arányos pont kell a térbeli felbontáshoz, és $L^2 \log L$ a teljes komplexitás.

Ezek a számolások természetesen csak elméletiek, és csupán nagyvonalakban adják meg a futásidőt, a gyakorlati szimulálás során más szempontok is szerepet játszanak.

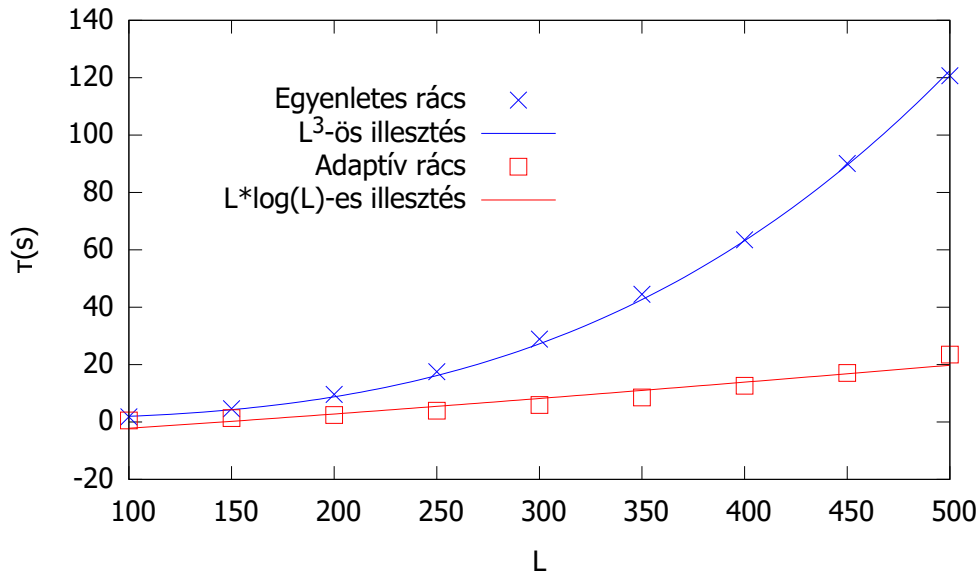
6.3. A futásidők kísérleti meghatározása

A szükséges indítási paraméterek ismeretében kísérletileg feltárható a τ futásidő L -től való függése. A 11. ábrán L függvényében ábrázoltam a hurok ütközéséig szükséges futásidőt egyenletes és adaptív rácsra is. A szimulációkat a saját otthoni laptopomon

futtattam, így a pontos futásidőknek nincs jelentősége, csupán az L -től függés a fontos. A futásidőbe nincs beleszámolva a kezdőfeltétel (és adaptív esetben a kezdőrács) felvétele, sem az adatok deklarálása, és a memóriakezelés. A szimulációkhoz használt összes paraméter és mérési adat a C függelékben látható.

A másik fő hibaforrás a kezdősebesség tökéletlenségéből fakad. Az egyenletes rácsnál annak nagyon nagy stabilitása miatt ez nem jelent problémát, az adaptív rács esetében viszont ahhoz vezet, hogy a hurok sebessége nem pont annyi lesz, mint amennyinek lennie kellene. Ez okoz hibát az energiában, és az ütközés (szimulációbeli) idejében is. Emiatt az ütközés idő nem pont egyforma az adaptív és az egyenletes rács esetében, ezért valójában csak az ütközés *körülbelüli* időpontjáig szimulálok. Ez a jelenség, hogy az ütközés nem ugyanakkor történik a két szimuláció esetében, kizárólag akkor jön elő, ha adok a rendszernek kezdősebességet. Ha nem, akkor akár közeli, akár viszonylag távoli a két húr, az ütközés pontosan ugyanakkor történik. Ez tehát kizárólag abból fakad, hogy az adaptív rácspan ez a fajta kezdősebesség máshogy "értelmeződik", mint egyenletes rács esetén.

A mért időkre illesztettem az imént megadott elméleti képleteket. Az adaptív adat-sorra az illeszkedő görbe egyenlete $\tau_a(L) = (-6 \pm 2) + (0.0083 \pm 0.0009) \cdot L \log L$, míg az egyenletesre $\tau_e(L) = (1 \pm 0.6) + (9.7 \pm 0.1) \cdot 10^{-7} \cdot L^3$. Az egyenletes rács görbéje igen jól illeszkedik, a konstans eltolás, amit a szisztematikus hibák okoznak, kicsi. A leg-



11. ábra. A szimulációk lefutásához szükséges τ idő másodpercben az ütközésig¹¹ L hosszparaméter függvényében. Az egyenletes rács $\mathcal{O}(L^3)$ komplexitású, az adaptív rács pedig $\mathcal{O}(L \log L)$ -es, ezek az illesztések is az ábrán láthatók.

¹¹Valójában csak az ütközés körülbelüli idejéig, lásd a törzsszöveget. A kétfajta szimulációt természetesen minden L -re egyforma szimulációbeli ideig futtattam.

jobb illeszthető hatványfüggvény hatványa 2.8 ± 0.02 lenne, tehát az elméleti számolás körülbelül helyes jóslást adott.

Az adaptív rácsnál az elméleti becslés eleve csak nagy L esetén volt érvényes, ha túl kevés pont van, akkor a pontosság nem lesz annyira jó, mint amit az elmélet megkövetelne, illetve, ha túl nagyok a hurok L -hez képest, akkor már nem elég $\log L$ -lel arányos darabszámú pont a felbontáshoz. Emiatt a konstans eltolás itt nem csak szisztematikus hibákat tartalmaz, és ezért negatív. Mindemellett megfigyelhetünk egy enyhe görbülést is a pontokban a majdnem lineáris $L \log L$ illesztéshez képest. Ezt a kezdősebesség rossz megadása okozza. Ahogy L -t növeljük, nő az ütközésig eltelt szimulációbeli idő. Emiatt a kezdősebességet nagyobb pontossággal kell meghatározni, hogy a hiba ne halmozódjon fel a hosszú úton, és 3% alatt maradjon. A kezdősebesség pontosabb beépítéséhez a mezőt egy kicsit részletesebben fel kell bontani, vagyis választott N_{max} -nak jobban kell nőnie, mint az elméleti $\log L$. Ez a növekmény elég kicsi ahhoz, hogy a dt -t nem szükséges csökkenteni, (a $dt = 0.075$ minden L -re jól működött,) ahhoz viszont elég nagy, hogy a 11. ábrán a piros pontoknak az illesztett (majdnem) egyeneshez képesti elgörbüléséhez vezessen.

Ennek a hibának a javításához óvatosabban felvett kezdőfeltételre van szükség. Ez elérhető például úgy, hogy a kezdőfeltételt időben visszafele léptetjük egy kis disszipáló tag berakása mellett, ezzel "kisimítva" a mezőt.

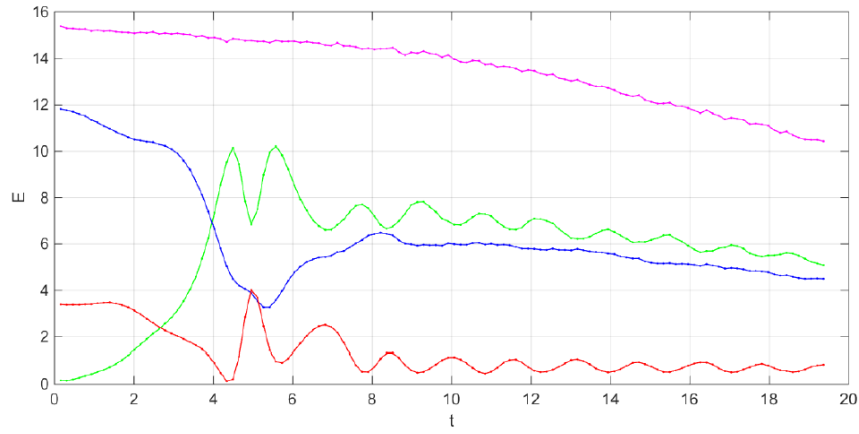
6.4. Összehasonlítás másik adaptív ráccsal

Érdemes összehasonlítani a szimuláció pontosságát mások által írt adaptív rácsú szimulációkkal. Három dimenzióban több adaptív ráccsal való szimulálással foglalkozó cikk is létezik [32, 33], két dimenzióban azonban csak a nyilvánosan nem elérhető [11] orosz szakdolgozatban találtam ábrát az ő adaptív szimulációjának energiamegmaradásáról, úgyhogy ezt fogom használni az összehasonlításra.

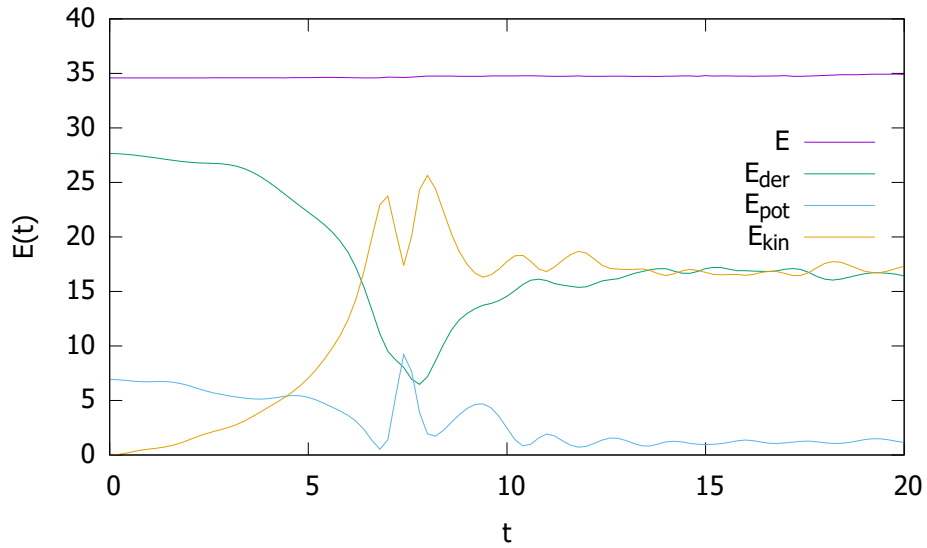
A szerző a szimulációjában az energiasűrűség szerint, Delaunay-háromszögeléssel vett fel pontokat, majd ezeket a hurokkal együtt mozgatta az időben. A diszkretizációhoz a végeselem módszert használta, vagyis a teret háromszögekre bontotta a pontok között. Ez a módszer szintén $L \log L$ -es komplexitású, azonban az energiát igen rosszul tartja.

A 12. ábrán ennek a szakdolgozatnak az ő szimulációjáról készült energia-ábráját hasonlítom össze az én szimulációmmal, ugyanolyan kezdőfeltétel mellett készült ábrával. Más egységrendszer (más λ) miatt az energiák nagysága nem egyforma, de térben és időben úgy skálázta a szerző az ábrát, mintha a $\lambda = 1$ egységrendszerben számolt volna, így összehasonlíthatók az eredmények.

Az orosz dolgozat ábráján az látszik, hogy már erre a nagyon kicsi skálakülönbségre



(a) A [11] dolgozat ábrája az energiáról



(b) Saját ábra az energiáról

12. ábra. A [11] szakdolgozat ábrája az ő adaptív szimulációjának az energiatartásáról és saját ábra az enyémről. A kezdőfeltételben a hurok nagyon közelről, $2R_0 = 5.48$ távolságról ($\eta = 0.97$), kezdősebesség nélkül indulnak. Az energiaskála más egységrendszer miatt nem egyforma, azonban az időskála egyezik. A felső ábrán a összenergia erősen változik, az alsón 1%-on belül állandó marad.

($\eta = 0.97$) is gyorsan csökken az összenergia. Az én algoritmusom erre, és ennél jóval nagyobb skálakülönbségekre is stabilan tartja az energiát.

7. Összefoglalás

Az axion egy hipotetikus részecske, mely a fizika két nyitott kérdését is megválaszolhatja. A kísérleti kimutatása azonban igen nehéz, így előbb szimulációk segítségével minél pontosabban meg kell határozni a részecske tömegét. Ehhez az axionmező időbeli fejlődését kell vizsgálni, és a húroknak nevezett topologikus defektusok számának alakulását az univerzum tágulása során. A szimulálást a tradicionális módszerrel, egyenletes rácsot használva azonban lehetetlenné teszik a hatalmas skálakülönbségek, amik megjelenhetnek a hurok és a vizsgált térfogat méretei között.

A munkám során egy általam kifejlesztett és megírt adaptív rácsú szimulációt mutatok be, ami két dimenzióban sokkal jobban kezeli ezt a problémát. Ennek alapjául egy olyan algoritmust alkalmaztam, amit a mezőelmélet területén még nem használtak, viszont más, hasonló algoritmusokkal szemben az az előnye, hogy egyszerűen általánosítható három dimenzióra is.

A program megírása során meglepően nehéznek bizonyult egy olyan diszkretizációt találni, amivel az energia a rács időbeli mozgatása ellenére is stabilan megmarad. A dolgozatomban bemutatom az egyszerűnek tűnő diszkretizáció helytelenségét, és levezetek helyette egy olyat, ami jóval bonyolultabb, de stabil energiatartást eredményez.

Az algoritmust egymás felé elindított centrális húr-antihúr ütközéssel teszteltem. Erre megmutatom, hogy a szimulációm futásideje $L \log L$ módon függ a skálakülönbséggel arányos dimenziótlan hosszparaméterről, míg a hagyományos módszer esetében ugyanez a függés L^3 . Mindemellett a programom futásideje már viszonylag alacsony skálakülönbségre is kisebb.

8. Kitekintés

Más, axionmező szimulálásával foglalkozó dolgozatok és cikkek két dimenzióban általában megvizsgálják a rendszert véletlenszerű kezdőfeltétellel is, és elemzik a hurok számának időbeli változását. A modellt tovább lehet fejleszteni a korai univerzum tágulásának beépítésével, amihez egy extra taggal kell kiegészíteni a Lagrange-függvényt. Ennek a dolgozatnak a célja magának az új módszernek bemutatása, de az eredmények alapján mindenképpen érdemes ezeket a szimulációkat elvégezni. A témában kutatás végcéljához, az axion tömegének meghatározáshoz, három dimenzióban kell megtenni ugyanezt. A három dimenzióba átlépés legtöbb adaptív rács esetén igen nehéz, de a quadtree algoritmus általánosítása nagyon egyszerű. Ehhez újra kell számolni a diszkretizációt, és a több változó hosszabb kifejezést fog eredményezni, de feltehetően a fent leírt módszerrel számolva három dimenzióban is stabil energiatartás adódik. A több tagból álló képlet és a több lehetséges szélső pont miatti nagyobb esetszétválasztás las-

sítani fogja a szimulációt, de ha az energiamegmaradás teljesül, akkor az L^4 helyetti $L^2 \log L$ -es távolágfüggvény miatt továbbra is sokkal jobb lesz az egyenletes rácsnál. Az ígéretes eredmények alapján mindenképpen érdemes feltárni ezeket a későbbiekben.

Köszönetnyilvánítás

Köszönöm témavezetőmnek, Katz Sándornak szakmai segítségét és támogatását. Köszönöm Tóth Bálintnak a szimuláció debugolásához nyújtott segítségét, és Hervay Ben-
cének, aki bemutatta nekem a quadtree algoritmust.

A Kulturális és Innovációs Minisztérium ÚNKP-22-1 kódszámú Új Nemzeti Kiváló-
sági Programjának a Nemzeti Kutatási, Fejlesztési és Innovációs Alapból finanszírozott
szakmai támogatásával készült.



Hivatkozások

- [1] Peccei, Roberto D.; Quinn, Helen R. "CP Conservation in the Presence of Pseudo-
particles" Phys. Rev. Lett. 38, 1440 (1977)
- [2] Pospelov P., Ritz A. Electric dipole moments as probes of new physics. Annals Phys.
318, 119–169 (2005).
- [3] Shellard E., Battye R. Origin of dark matter axions, Physics Reports. 1998. Vol.
307, no. 1. P. 227-234.
- [4] M. Gorghetto, E. Hardy and G. Villadoro, Axions from Strings: the Attractive
Solution, JHEP 07 (2018), 151
- [5] Peccei, R. D. The Strong CP Problem and Axions. Lect. Notes Phys. 741, 3–17
(2008).
- [6] Borsanyi, S. et al. Calculation of the axion mass based on high-temperature lattice
quantum chromodynamics. Nature 539, 69–71 (2016).
- [7] Moore G. D. Intercommutation of U(1) global strings. (2016). [arXiv:1604.02356
[hep-ph]].
- [8] A. Vilenkin, E.P.S. Shellard. Cosmic strings and other topological defects. Camb-
ridge University Press, (1994).
- [9] M. Fleury, L. & D. Moore, G. Axion string dynamics I: 2+1D. JCAP 05 (2016),
005.

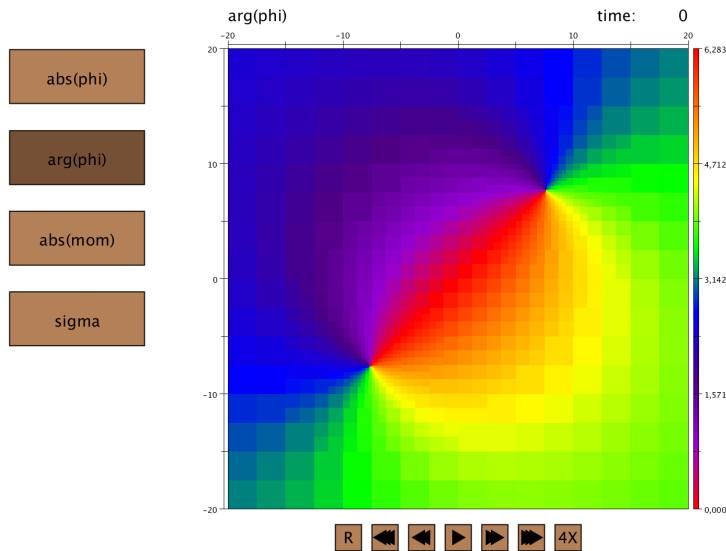
- [10] Hecht, Matthew W. and DeGrand, Thomas A. Radiation patterns from vortex-antivortex annihilation. *Phys. Rev. D.* (1990) Vol. 42. P. 519 - 528.
- [11] Boytsov Alexander Vladimirovich "Simulation of axion strings" Saint Petersburg State University (2020).
- [12] Gropp W. D. Local uniform mesh refinement with moving grids. *SIAM Journal on Scientific and Statistical Computing.* (1987) Vol. 8, no. 3. P. 292-304.
- [13] R.A. Trompert and J.G. Verwer. A static-regridding method for twodimensional parabolic partial differential equations. *Applied Numerical Mathematics.* (1991) Vol. 8, no. 1. P. 65-90.
- [14] Marsha J Berger and Joseph Oliger. Adaptive mesh refinement for hyperbolic partial differential equations *Journal of Computational Physics.* (1984) Vol. 53, no. 3. P. 484-512.
- [15] Zegeling P. Theory and application of adaptive moving grid methods *Adaptive Computations: Theory and Algorithms* Ed. by T. Tang and J. Xu. Beijing : Science Press, (2007) P. 279-332.
- [16] Budd, Chris J. and Huang, Weizhang and Russell, Robert D. Adaptivity with moving grids *Acta Numerica.* (2009) Vol. 18.
- [17] Weizhang Huang, Robert D. Russell. *Adaptive Moving Mesh Methods.* Springer, (2011).
- [18] Daryl L. Logan A first course in the finite element method. Cengage Learning. (2011).
- [19] Rupak Biswas, Karen D. Devine and Joseph E. Flaherty. Parallel, adaptive finite element methods for conservation laws *Applied Numerical Mathematics.* (1994) Vol. 14, no. 1. P. 255-283.
- [20] R.A. Gingold; J.J. Monaghan. Smoothed particle hydrodynamics: theory and application to non-spherical stars. *Mon. Not. R. Astron. Soc.* 181 (3): 375–89. (1977).
- [21] L.B. Lucy, A numerical approach to the testing of the fission hypothesis. *Astron. J.* 82: 1013–1024. (1977).
- [22] G.R.Liu, M.B. Liu. *Smoothed Particle Electrodynamics, A Meshfree Particle Method.* Singapore : World Scientific Publishing Co. Pte. Ltd., (2003).
- [23] Finkel, R. A.; Bentley, J. L. "Quad trees a data structure for retrieval on composite keys". *Acta Informatica.* 4 (1): 1–9. (1974).
- [24] Samet, H. "The quadtree and related hierarchical data structures". *ACM Computing Surveys.* ACM. 16 (2): 187–260. (1984).
- [25] Neumann, P., Neckel, T. A dynamic mesh refinement technique for Lattice Boltzmann simulations on octree-like grids. *Comput Mech* 51, 237–253 (2013).

- [26] Saftly, W. and Camps, P. and Baes, M. and Gordon, K. D. and Vandewoude, S. and Rahimi, A. and Stalevski, M. "Using hierarchical octrees in Monte Carlo radiative transfer simulations" *Astron. Astrophys.* (2013).
- [27] Chaowei Yang, Huayi Wu, Qunying Huang, Zhenlong Li, Jing Li "Using spatial principles to optimize distributed computing for enabling the physical science discoveries" *Proceedings of the National Academy of Sciences* vol. 108 n. 14 p 5498-5503, (2011).
- [28] I.P. Omelyan and I.M. Mryglod and R. Folk. Symplectic analytically integrable decomposition algorithms: classification, derivation, and application to molecular dynamics, quantum and celestial mechanics simulations *Computer Physics Communications*. (2003). Vol. 151, no. 3. P. 272-314.
- [29] Skeel, R. D., "Variable Step Size Destabilizes the Störmer/Leapfrog/Verlet Method", *BIT Numerical Mathematics*, Vol. 33, (1993), p. 172–175.
- [30] Katona D. "Axion hűrok klasszikus térelméleti szimulációja" *ELTE TTK* (2018).
- [31] R. Courant, K. Friedrichs, H. Lewy, *Über die partiellen Differenzengleichungen der mathematischen Physik*, *Mathematische Annalen* 100 (1928) 32–74
- [32] M. Buschmann, J. W. Foster, A. Hook, A. Peterson, D. E. Willcox, W. Zhang and B. R. Safdi, "Dark matter from axion strings with adaptive mesh refinement", *Nature Commun.* 13 (2022) no.1, 1049
- [33] A. Drew, E.P.S.Shellard, "Radiation from global topological strings using adaptive mesh refinement: Methodology and massless modes", *Phys. Rev. D* 105 (2022) no.6, 063517

A. A szimulációról és az ábrázoló szoftverről

A szimulációt (mind az adaptívat, mind az egyenleteset) a gyors futás érdekében `c++` nyelven írtam. Nem használtam ezekhez semmilyen fizika engine-t vagy libraryt, hogy a lehető legalappsztűbben irányíthassan a szimulációt. Az adaptív szimuláció futás-idejének jelenleg 98%-át a mező időbeli léptetése, azon belül is (17) kiszámolása jelenti, így további optimalizálásra leginkább annak minél gyorsabban számolható alakban való felírása ad lehetőséget.

Az ábrázoláshoz Java nyelven írtam egy ábrázolószoftvert, innen exportáltam képeket a dolgozatba. Ebben a szoftverben lehetőség van a leszimulált folyamatot tetszőleges időpillanatban megnézni, vagy videóként lejátszani, mindezt több különböző nézetben. Egy kép a programról a 13. ábrán látható.



13. ábra. Az ábrázolóprogramról egy kép. A bal oldalon lehet változtatni a nézetet, az alsó gombokkal lehet az időben előre vagy hátra lépegetni, videószerű lejátszást kezdeni, a lejátszás sebességét állítani, illetve visszamenni a $t = 0$ időhöz

B. Egy cella teljes Hamilton-függvénye

$$\begin{aligned}
H = \lambda d^2 \bigg(& \frac{109\phi_1^2(\phi_1^*)^2}{720} + \frac{7\phi_1^2\phi_1^*\phi_3^*}{360} + \frac{\phi_1^2\phi_1^*\phi_4^*}{18} + \frac{\phi_1^2\phi_1^*\phi_2^*}{18} + \frac{\phi_1^2(\phi_3^*)^2}{720} + \frac{\phi_1^2\phi_3^*\phi_4^*}{180} + \frac{\phi_1^2\phi_3^*\phi_2^*}{180} + \frac{11\phi_1^2(\phi_4^*)^2}{720} \\
& + \frac{\phi_1^2\phi_4^*\phi_2^*}{120} + \frac{11\phi_1^2(\phi_2^*)^2}{720} + \frac{19\phi_1(\phi_1^*)^2\phi_2}{480} + \frac{19\phi_1(\phi_1^*)^2\phi_4}{480} + \frac{67\phi_1(\phi_1^*)^2\phi_m}{720} + \frac{\phi_1\phi_1^*\phi_2\phi_3^*}{144} \\
& + \frac{\phi_1\phi_1^*\phi_2\phi_4^*}{144} + \frac{41\phi_1\phi_1^*\phi_2\phi_2^*}{720} + \frac{\phi_1\phi_1^*\phi_3^*\phi_4}{144} + \frac{11\phi_1\phi_1^*\phi_3^*\phi_m}{360} + \frac{41\phi_1\phi_1^*\phi_4\phi_4^*}{720} + \frac{\phi_1\phi_1^*\phi_4\phi_2^*}{144} \\
& + \frac{7\phi_1\phi_1^*\phi_4^*\phi_m}{120} + \frac{7\phi_1\phi_1^*\phi_m\phi_2^*}{120} - \frac{3\phi_1\phi_1^*}{4} + \frac{\phi_1\phi_2(\phi_3^*)^2}{1440} + \frac{\phi_1\phi_2\phi_3^*\phi_4^*}{720} + \frac{\phi_1\phi_2\phi_3^*\phi_2^*}{144} + \frac{\phi_1\phi_2(\phi_4^*)^2}{1440} \\
& + \frac{\phi_1\phi_2\phi_4^*\phi_2^*}{144} + \frac{19\phi_1\phi_2(\phi_2^*)^2}{480} + \frac{\phi_1(\phi_3^*)^2\phi_4}{1440} + \frac{\phi_1(\phi_3^*)^2\phi_m}{240} + \frac{\phi_1\phi_3^*\phi_4\phi_4^*}{144} + \frac{\phi_1\phi_3^*\phi_4\phi_2^*}{720} + \frac{\phi_1\phi_3^*\phi_4^*\phi_m}{72} \\
& + \frac{\phi_1\phi_3^*\phi_m\phi_2^*}{72} - \frac{\phi_1\phi_3^*}{12} + \frac{19\phi_1\phi_4(\phi_4^*)^2}{480} + \frac{\phi_1\phi_4\phi_4^*\phi_2^*}{144} + \frac{\phi_1\phi_4(\phi_2^*)^2}{1440} + \frac{\phi_1(\phi_4^*)^2\phi_m}{48} + \frac{7\phi_1\phi_4^*\phi_m\phi_2^*}{360} \\
& - \frac{\phi_1\phi_4^*}{4} + \frac{\phi_1\phi_m(\phi_2^*)^2}{48} - \frac{\phi_1\phi_2^*}{4} + \frac{11(\phi_1^*)^2\phi_2^2}{720} + \frac{(\phi_1^*)^2\phi_2\phi_3}{1440} + \frac{(\phi_1^*)^2\phi_2\phi_m}{48} + \frac{(\phi_1^*)^2\phi_3^2}{720} \\
& + \frac{(\phi_1^*)^2\phi_3\phi_4}{1440} + \frac{(\phi_1^*)^2\phi_3\phi_m}{240} + \frac{11(\phi_1^*)^2\phi_4^2}{720} + \frac{(\phi_1^*)^2\phi_4\phi_m}{48} + \frac{(\phi_1^*)^2\phi_m^2}{18} + \frac{\phi_1^*\phi_2^2\phi_3^*}{120} + \frac{\phi_1^*\phi_2^2\phi_4^*}{180} \\
& + \frac{\phi_1^*\phi_2^2\phi_2^*}{18} + \frac{\phi_1^*\phi_2\phi_3\phi_3^*}{144} + \frac{\phi_1^*\phi_2\phi_3\phi_4^*}{720} + \frac{\phi_1^*\phi_2\phi_3\phi_2^*}{144} + \frac{7\phi_1^*\phi_2\phi_3^*\phi_m}{360} + \frac{\phi_1^*\phi_2\phi_4^*\phi_m}{72} + \frac{7\phi_1^*\phi_2\phi_m\phi_2^*}{120} \\
& - \frac{\phi_1^*\phi_2}{4} + \frac{7\phi_1^*\phi_3^2\phi_3^*}{360} + \frac{\phi_1^*\phi_3^2\phi_4^*}{180} + \frac{\phi_1^*\phi_3^2\phi_2^*}{180} + \frac{\phi_1^*\phi_3\phi_3^*\phi_4}{144} + \frac{11\phi_1^*\phi_3\phi_3^*\phi_m}{360} + \frac{\phi_1^*\phi_3\phi_4\phi_4^*}{144} + \frac{\phi_1^*\phi_3\phi_4\phi_2^*}{720} \\
& + \frac{\phi_1^*\phi_3\phi_4^*\phi_m}{72} + \frac{\phi_1^*\phi_3\phi_m\phi_2^*}{72} - \frac{\phi_1^*\phi_3}{12} + \frac{\phi_1^*\phi_3^2\phi_4^*}{120} + \frac{7\phi_1^*\phi_3^*\phi_4\phi_m}{360} + \frac{\phi_1^*\phi_3^*\phi_m^2}{15} + \frac{\phi_1^*\phi_4^2\phi_4^*}{18} + \frac{\phi_1^*\phi_4^2\phi_2^*}{180} \\
& + \frac{7\phi_1^*\phi_4\phi_4^*\phi_m}{120} + \frac{\phi_1^*\phi_4\phi_m\phi_2^*}{72} - \frac{\phi_1^*\phi_4}{4} + \frac{7\phi_1^*\phi_4^*\phi_m^2}{90} + \frac{7\phi_1^*\phi_m^2\phi_2^*}{90} - \frac{2\phi_1^*\phi_m}{3} + \frac{11\phi_2^2(\phi_3^*)^2}{720} + \frac{\phi_2^2\phi_3^*\phi_4^*}{180} \\
& + \frac{\phi_2^2\phi_3^*\phi_2^*}{18} + \frac{\phi_2^2(\phi_4^*)^2}{720} + \frac{7\phi_2^2\phi_4^*\phi_2^*}{360} + \frac{109\phi_2^2(\phi_2^*)^2}{720} + \frac{19\phi_2\phi_3(\phi_3^*)^2}{480} + \frac{\phi_2\phi_3\phi_3^*\phi_4^*}{144} + \frac{41\phi_2\phi_3\phi_3^*\phi_2^*}{720} \\
& + \frac{\phi_2\phi_3(\phi_4^*)^2}{1440} + \frac{\phi_2\phi_3\phi_4^*\phi_2^*}{144} + \frac{19\phi_2\phi_3(\phi_2^*)^2}{480} + \frac{\phi_2(\phi_3^*)^2\phi_m}{48} + \frac{\phi_2\phi_3^*\phi_4^*\phi_m}{72} + \frac{7\phi_2\phi_3^*\phi_m\phi_2^*}{120} - \frac{\phi_2\phi_3^*}{4} \\
& + \frac{\phi_2(\phi_4^*)^2\phi_m}{240} + \frac{11\phi_2\phi_4^*\phi_m\phi_2^*}{360} - \frac{\phi_2\phi_4^*}{12} + \frac{67\phi_2\phi_m(\phi_2^*)^2}{720} - \frac{3\phi_2\phi_2^*}{4} + \frac{109\phi_3^2(\phi_3^*)^2}{720} + \frac{\phi_3^2\phi_3^*\phi_4^*}{18} \\
& + \frac{\phi_3^2\phi_3^*\phi_2^*}{18} + \frac{11\phi_3^2(\phi_4^*)^2}{720} + \frac{\phi_3^2\phi_4^*\phi_2^*}{120} + \frac{11\phi_3^2(\phi_2^*)^2}{720} + \frac{19\phi_3(\phi_3^*)^2\phi_4}{480} + \frac{67\phi_3(\phi_3^*)^2\phi_m}{720} + \frac{41\phi_3\phi_3^*\phi_4\phi_4^*}{720} \\
& + \frac{\phi_3\phi_3^*\phi_4\phi_2^*}{144} + \frac{7\phi_3\phi_3^*\phi_4^*\phi_m}{120} + \frac{7\phi_3\phi_3^*\phi_m\phi_2^*}{120} - \frac{3\phi_3\phi_3^*}{4} + \frac{19\phi_3\phi_4(\phi_4^*)^2}{480} + \frac{\phi_3\phi_4\phi_4^*\phi_2^*}{144} + \frac{\phi_3\phi_4(\phi_2^*)^2}{1440} \\
& + \frac{\phi_3(\phi_4^*)^2\phi_m}{48} + \frac{7\phi_3\phi_4^*\phi_m\phi_2^*}{360} - \frac{\phi_3\phi_4^*}{4} + \frac{\phi_3\phi_m(\phi_2^*)^2}{48} - \frac{\phi_3\phi_2^*}{4} + \frac{11(\phi_3^*)^2\phi_4^2}{720} + \frac{(\phi_3^*)^2\phi_4\phi_m}{48} \\
& + \frac{(\phi_3^*)^2\phi_m^2}{18} + \frac{\phi_3^*\phi_4^2\phi_4^*}{18} + \frac{\phi_3^*\phi_4^2\phi_2^*}{180} + \frac{7\phi_3^*\phi_4\phi_4^*\phi_m}{120} + \frac{\phi_3^*\phi_4\phi_m\phi_2^*}{72} - \frac{\phi_3^*\phi_4}{4} + \frac{7\phi_3^*\phi_4^*\phi_m^2}{90} + \frac{7\phi_3^*\phi_m^2\phi_2^*}{90} \\
& - \frac{2\phi_3^*\phi_m}{3} + \frac{109\phi_4^2(\phi_4^*)^2}{720} + \frac{7\phi_4^2\phi_4^*\phi_2^*}{360} + \frac{\phi_4^2(\phi_2^*)^2}{720} + \frac{67\phi_4(\phi_4^*)^2\phi_m}{720} + \frac{11\phi_4\phi_4^*\phi_m\phi_2^*}{360} - \frac{3\phi_4\phi_4^*}{4} \\
& + \frac{\phi_4\phi_m(\phi_2^*)^2}{240} - \frac{\phi_4\phi_2^*}{12} + \frac{(\phi_4^*)^2\phi_m^2}{18} + \frac{\phi_4^*\phi_m^2\phi_2^*}{15} - \frac{2\phi_4^*\phi_m}{3} + \frac{\phi_m^2(\phi_2^*)^2}{18} - \frac{2\phi_m\phi_2^*}{3} + 4 \bigg) \\
& + \frac{3}{4} \sum_{j=1}^4 |\phi_j|^2 - \frac{1}{4} \sum_{k=2}^4 \sum_{j=1}^{k-1} (\phi_j\phi_k^* + \phi_k\phi_j^*) + d^2 (p_1p_1^* + p_2p_2^* + p_3p_3^* + p_4p_4^*)
\end{aligned}$$

C. A futásidők mérési adatai

A 11. ábrában a szimulációkhoz használt pontos paraméterek és mérési adatok az 1. táblázatban láthatók. L hosszúságparaméterrel T szimulációbeli időegységig tartott a futtatás. Az összes adatpont száma az adaptív esetben $N_{max} = N_0 \cdot N_0 + N_{extra}$, ahol $N_0 = 10$ mindenhol, N_{extra} pedig a táblázatban látható. Az egyenletes rács esetében a pontok száma $N_{max} = N^2$. A mért adat, τ a valós időben a futásidő másodpercben.

L	Adaptív rács			Egyenletes rács			T
	$\tau[s]$	N_{extra}	dt	$\tau[s]$	N	dt	
100	0.55	1000	0.075	1.82	200	0.2	
150	1.36	1650	0.075	4.61	300	0.2	
200	2.41	2200	0.075	9.60	400	0.2	
250	3.95	2900	0.075	17.5	500	0.2	
300	5.92	3600	0.075	28.9	600	0.2	
350	8.53	4400	0.075	44.5	700	0.2	
400	12.6	5700	0.075	63.5	800	0.2	
450	17.1	6550	0.075	90.1	900	0.2	
500	23.5	8050	0.075	120.7	1000	0.2	

1. táblázat. A szimulálásokhoz használt paraméterek a 11. ábrán.

A mért idő pontos értéke akár drasztikusan módosulhat a számítógépen a háttérben zajló folyamatok függvényében. Az összes ilyen jellegű hiba hatásának csökkenése érdekébe minden adatot háromszor mértem, ezeket átlagoltam. Az összes szimulációt közvetlenül egymás után futtattam, közben másra nem használtam a laptopot. Mindenféle Windows-os folyamatok így is bezavarhatnak a mérésekbe, ez a kisebb időkre igen látványos 1-2 másodperc növekményt okozott a másik mérésekhez képest. Ezeket az időket emiatt újramértem.

Meg kell jegyezni emellett, hogy GPU optimalizációval az egyenletes rácsú szimuláció futásideje nagyban gyorsítható. Ez azonban csak egy konstans szorzóval javítja, vagyis nagyobb skálán ugyanúgy csak adaptív szimuláció használható.