



M Ű E G Y E T E M 1 7 8 2

# Tudományos Diákköri Konferencia



Diffúziós VVER zónaszámítások az SPNDYN  
végelem-módszeren alapuló reaktorfizikai kóddal

**Illés Gergely**  
COZJ2Z

Témavezető:  
Babcsány Boglárka  
tudományos segédmunkatárs  
BME Nukleáris Technikai Intézet

BME  
2021



# Tartalomjegyzék

<b>1. Bevezetés</b>	<b>1</b>
<b>2. A számítások háttere és motivációja</b>	<b>2</b>
2.1. Az SPNDYN diffúziós modulja . . . . .	2
2.2. Az SPNDYN rövid története és jelenlegi képességei . . . . .	6
2.3. A teszteléshez használt benchmarkok . . . . .	7
<b>3. Számítási eredmények</b>	<b>9</b>
3.1. Seidel benchmark . . . . .	9
3.1.1. A kétdimenziós modell specifikációja . . . . .	9
3.1.2. Hálófüggetlenség vizsgálata a kétdimenziós modellen . . . . .	10
3.1.3. Konvergenciaparaméterek vizsgálata a kétdimenziós modellen . . . . .	12
3.1.4. A kétdimenziós modell eredményei . . . . .	14
3.1.5. A háromdimenziós modell specifikáció . . . . .	15
3.1.6. Hálófüggetlenség vizsgálata a háromdimenziós modellen . . . . .	16
3.1.7. A háromdimenziós számítási eredmények kiértékelése . . . . .	17
3.2. Dyn001 visszacsatolásmentes időfüggő benchmark . . . . .	19
3.2.1. Geometria . . . . .	19
3.2.2. Csoportállandók időbeli változása . . . . .	21
3.2.3. Teljesítményszámítás és kezdeti eredmények . . . . .	22
3.2.4. A "rod cusping" jelensége és lehetséges kezelései . . . . .	23
3.2.5. Végeredmények . . . . .	26
<b>4. Összegzés és kitekintés</b>	<b>32</b>
<b>5. Irodalomjegyzék</b>	<b>33</b>
<b>6. Függelék</b>	<b>35</b>



## Alkalmazott jelölések

$\Psi(\underline{r}, \Omega, E, t)$	vektorfluxus
$\Phi(\underline{r}, E, t)$	skalárfluxus
$\underline{r}$	helyvektor
$\Omega$	térszög
$t$	idő
$v$	neutronsebesség
$\Sigma_s(E' \rightarrow E)$	szórási magfüggvény
$f(E)$	hasadási spektrum
$\nu$	hasadásonként keletkező neutronok átlagos száma
$S$	külső forrás
$D$	diffúziós állandó
$\Sigma_t$	totális makroszkopikus hatáskeresztmetszet
$\Sigma_f$	hasadási makroszkopikus hatáskeresztmetszet
$\Sigma_a$	abszorpciós makroszkopikus hatáskeresztmetszet
$\Sigma_s$	szórási makroszkopikus hatáskeresztmetszet
$k_{\text{eff}}$	effektív sokszorozási tényező
$Q$	forrástag
$g$ alsó index	$g$ -edik energiacsoportra vonatkozó mennyiség, melyet ekkor csoportállandónak hívunk
$\Sigma_R$	kiszórási csoportállandó
$C_i$	$i$ -edik későneutron-csoport anyamag koncentrációja
$\lambda_i$	$i$ -edik későneutron-csoport bomlásállandója
$\beta_i$	$i$ -edik későneutron-csoport részaránya

## Általános megjegyzések

A dolgozatban több helyen is elő fognak fordulni angol nyelvű kifejezések. Az egyik ilyen azért maradt angol nyelvű, mert a magyar szakirodalomban nincs megfelelője, a többi helyen viszont egy szakirodalomból átvett ábra jelölései, melyek csak azért jelennek meg a szövegben, hogy az ábrán beazonosítható legyen az adott görbe, így a kifejezés megértése nem szükséges.

A dolgozatban végeelem-módszerről és nodális kódokról is lesz szó, a nódus kifejezés pedig mindkét módszernél megjelenik, de mást jelent. A végeelem-módszer esetén a háló egy csomópontját, míg a nodális módszeres értelmében a diszkretizált geometria egy térfogatát jelöli. Ahol nagyon zavaró lett volna a kétféle jelentés keveredése, ott a nodális módszeres értelemben vett nódust igyekeztem a geometriai térfogat kifejezéssel megfogalmazni. Ahol viszont egyértelmű, hogy melyik jelentésről van szó, mint például az ábrák feliratai, ott meghagytam a szakirodalomban megszokott nódus kifejezést.



# 1. Bevezetés

Mint a tudomány számos területén, úgy a reaktorfizikában is igen jelentős szerepe van a számítógépes szimulációnak. A reaktorfizikai modellezés közel 50 éves múltra tekint vissza és módszerei együtt fejlődtek a számítógépes kapacitások növekedésével. A legegyszerűbb egydimenziós egycsoport diffúzióegyenlet megoldásától mára egészen addig eljutottunk, hogy Monte-Carlo-módszerekkel már a sztochasztikus viselkedést is lehet modellezni. Persze utóbbihoz a jelenlegi legfejlettebb technika szükséges, egy átlagos személyi számítógéppel a determinisztikus transzportelmélet magasabb rendű közelítései oldhatók meg. Ennek jegyében a Budapesti Műszaki és Gazdaságtudományi Egyetem Nukleáris Technikai Intézetében is egy az  $SP_3$  egyenletrendszer megoldására szolgáló kód fejlesztése folyik. Az SPNDYN [1] fejlesztése jelenleg a validációs és verifikációs folyamatban van, melybe én a BSc szakdolgozatom [2] keretében kapcsolódtam be. Akkor a feladatom az SPNDYN stacionárius diffúziós moduljának VVER reaktortípusra történő verifikációja volt, mely során elfogadható eredményeket kaptam, így az SPNDYN ezen modulja már alkalmas lehet VVER zónaszámítások elvégzésére, így VVER-440 vagy a későbbiekben akár VVER-1200 reaktortípusok reaktorfizikai elemzésére is.

Természetesen reaktorfizikai elemzések során nem csak az állandósult állapotra lehetünk kíváncsiak, hanem különböző tranziensek lefolyására is. Ehhez pedig egy tranziens számítások elvégzésére alkalmas reaktorfizikai kódra van szükség, legtöbb esetben egy termohidraulikai kóddal csatolva. De legalább szubcsatorna kód kell hozzá. Az SPNDYN rendelkezik egy tranziens diffúziós modullal is, csatolása termohidraulikai rendszerkóddal pedig folyamatban van, így a szakdolgozatom keretében elkezdett munkát az SPNDYN ezen moduljának tesztelésével folytatom MSc tanulmányaim elején, és az eddig elért eredményeimet szeretném bemutatni jelen TDK dolgozatban.

A dolgozatban először áttekintem az SPNDYN alapjául szolgáló fizikai, matematikai és numerikus modelleket. Röviden bemutatom, hogy hol tart az SPNDYN csatolása az APROS termohidraulikai rendszerkóddal és hogyan kapcsolódik ehhez a folyamathoz az én munkám. Emellett a dolgozat elején arról is lesz szó, hogy milyen VVER reaktorokra vonatkozó időfüggő benchmarkfeladatok érhetőek el és ezek közül melyek valósíthatók meg az SPNDYN jelenlegi kódverziójával.

A dolgozat fő részét az elvégzett benchmarktesztek eredményeinek ismertetése képezi. Itt először statikus benchmarkok eredményeit fogom ismertetni, mely egyfajta felvezetésként fog szolgálni az ezt követő dinamikus benchmarkhoz. A stacionárius esetben egy hálófüggetlenségi vizsgálatot követően hasonlítom össze az SPNDYN-nel kapott eredményt a referenciamegoldásokkal. Ennek során látni fogjuk, hogy még diffúziós közelítés esetén is komoly számítógépes kapacitásigénnyel rendelkezik egy-egy összetettebb probléma vége-selem-módszerrel történő megoldása. A dinamikus benchmarknál az eredmények referenciamegoldásokkal való összehasonlítása mellett a felmerült problémák és megoldásaik is nagy hangsúlyt fognak kapni.

## 2. A számítások hátttere és motivációja

Bár a dolgozat lényege a benchmarkok elvégzése, nagyon fontos az elméleti alapokat, és az SPNDYN kód működését is megérteni, hiszen ez elengedhetetlen az eredmények értékeléséhez. Emellett azt is érdemes megnézni, hogy hogyan kapcsolódik be a munkám a kódfejlesztés folyamatába, illetve miért pont ezeket a benchmarkfeladatokat végeztem el. Ebben a fejezetben ezeket fogom kifejteni.

### 2.1. Az SPNDYN diffúziós modulja

Az atomreaktorokban önfenntartó lánreakció valósul meg, szabályozott körülmények között. A folyamat leírásához szükséges legfontosabb mennyiség a neutronfluxus, így a reaktorfizika legfontosabb feladata ennek minél pontosabb meghatározása. A neutronfluxus meghatározásához szükséges alapegyenlet a neutronokkal végbemenő fizikai folyamatok alapján felírható transzportegyenlet:

$$\frac{1}{v} \frac{\partial \Psi(\underline{r}, E, \underline{\Omega}, t)}{\partial t} = -\underline{\Omega} \text{grad} \Psi(\underline{r}, E, \underline{\Omega}, t) - \Sigma_t(E) \Psi(\underline{r}, E, \underline{\Omega}, t) + Q(\underline{r}, E, \underline{\Omega}, t), \quad (1)$$

ahol a forrástag:

$$Q(\underline{r}, E, \underline{\Omega}, t) = \int_{4\pi} \int_0^\infty \Sigma_s(E' \rightarrow E, \underline{\Omega}' \rightarrow \underline{\Omega}) \Psi(\underline{r}, E, \underline{\Omega}, t) dE' d\Omega' + \frac{f(E)}{4\pi} \int_{4\pi} \int_0^\infty \Sigma_f(E') \Psi(\underline{r}', E', \underline{\Omega}', t) dE' d\Omega' + S(\underline{r}, E, \underline{\Omega}, t). \quad (2)$$

A fenti neutrontranszport-egyenlet egy olyan integro-differenciál egyenlet, amelynek megoldása igen bonyolult, analitikus megoldása csak nagyon speciális esetekben létezik. A neutrontranszport leírására így két lehetőség marad:

- Monte-Carlo-módszerrel sztochasztikus elven,
- a transzporfolyamatok közelítéseivel kapott egyenletekkel determinisztikus módon.

Ebből a Monte-Carlo-módszer a lényegesen pontosabb, hiszen ott fizikai folyamatok közvetlen szimulálására van lehetőség. Így ezt tulajdonképpen nem is kell közelítő módszernek tekinteni. Hátránya viszont, hogy rendkívül erőforrás-igényes, még a mai számítógépes technológiai szint mellett is nagyon sokáig tart egy-egy ilyen szimuláció. Így ipari alkalmazásoknál, ahol a modellezés célja elsősorban a skalárfluxus és az effektív sokszorozási tényező meghatározása, a determinisztikus módszerek alkalmazása a jellemző.

A determinisztikus módszereknél a transzportegyenletben először a szögfüggést közelítik:

- $S_N$  módszer esetén  $\underline{\Omega}$ -t diszkrétizáljuk, ezzel a szög szerinti integrálok diszkrét szummázásokra cserélhetőek.
- $P_L$  módszer esetén a szög szerinti függést gömbfüggvények szerint kifejtjük, de az egyenleteket csak  $l \leq L$  rendig oldjuk meg.



Az SPNDYN-ben ezek közül az (egyszerűsített) SP<sub>3</sub> módszer, illetve a diffúzió elmélet (P<sub>1</sub>+más alább részletezett közelítések) lett implementálva. Én csak az SPNDYN diffúziós moduljával foglalkoztam, így az SP<sub>3</sub> módszert itt nem szeretném jobban kifejteni.

A diffúzióelmélet alapja a P<sub>1</sub> közelítés, azaz a szögfüggő neutronfluxus egy izotrop és egy lineárisan anizotrop fluxusmomentum lineáris kombinációjaként írható fel:

$$\Psi(\underline{r}, E, \underline{\Omega}, t) = \frac{1}{4\pi} \Phi(\underline{r}, E, t) + \frac{3}{4\pi} \underline{\Omega} J(\underline{r}, E, t) . \quad (3)$$

Emellett csak az izotrop szórásat vesszük figyelembe, illetve feltételezzük hogy  $\frac{\partial J}{\partial t} \approx 0$ . Utóbbi segítségével kapjuk a Fick-törvényt:

$$\underline{J}(\underline{r}, E, t) = -D(E) \text{grad} \Phi(\underline{r}, E, t) . \quad (4)$$

Ezen feltételek alkalmazásával kapjuk az általános diffúzióegyenletet:

$$\frac{1}{v} \frac{\partial \Phi(\underline{r}, E, t)}{\partial t} = \text{div} [D(E) \text{grad} \Phi(\underline{r}, E, t)] - \Sigma_t(E) \Phi(\underline{r}, E, t) + Q_0(\underline{r}, E, t) , \quad (5)$$

ahol az izotrop forrástag:

$$Q_0(\underline{r}, E, t) = \int_0^\infty \Sigma_s(E' \rightarrow E) \Phi(\underline{r}, E', t) dE' + f(E) \int_0^\infty \nu \Sigma_f(E') \Phi(\underline{r}, E', t) dE' + S(\underline{r}, E, t) . \quad (6)$$

Ezzel már csak egyetlen probléma van: az energiára és az attól függő mennyiségekre mint folytonos változók tekintettünk. A numerikus számításokhoz ezeket még energiában diszkretizálni kell, azaz át kell térni kevés- vagy többscsoport struktúra alkalmazására. Ennek alapja a reaktorfizika alaptétele, mely szerint homogén reaktorokban a határfelületektől távol a tér- és energiafüggés szétválasztható:

$$\Phi(\underline{r}, E, t) = \Phi(\underline{r}, t) \cdot \Psi(E) . \quad (7)$$

Itt a  $\Psi(E)$  energiaspektrumot különböző lassulásmodellek segítségével kaphatjuk meg. Ezáltal pedig, első közelítésben a teljes energiatartományra kiintegrálva, kapjuk az egycsoport-közelítést, melyben már csak egyetlen differenciálegyenletet kell megoldani. Összetett (különböző anyagi minőségű zónákat tartalmazó) reaktor esetén azonban ez nem használható kellő pontossággal. Ezért a teljes 0-10 MeV-ig terjedő energiatartományt véges számú intervallumra osztjuk fel, az ezekre történő integrálással pedig csoportállandókat, illetve csoportfluxusokat definiálunk. Így az egy integro-differenciálegyenlet helyett egy olyan differenciálegyenlet-rendszert kapunk, mely alkalmas diszkretizációt követően numerikusan már jól kezelhető. Kísérleteknél, például, ha a spektrum meghatározása a cél, akár 600 energiacsoportot is definiálnak, ipari alkalmazásoknál viszont a limitált számítási kapacitások miatt a kevéscsoport elméletek (2-6) használata a domináns. Igazából a könnyűvízes, termikus reaktorok szempontjából releváns jelenségeket már a kétcsoport-elmélet is egészen jól visszaadja, ennek megfelelően ipari körülmények között ez a leggyakrabban alkalmazott közelítés. Ekkor az alábbi két neutronenergia-csoportot definiáljuk:

1. Epitermikus csoport:  $E > 0.625$  eV,
2. Termikus csoport:  $E < 0.625$  eV.

A számítógépes kapacitások fejlődésével egyre inkább elterjedt a négycsoport elmélet alkalmazása is [3]. Ennél a csoportok:

1. csoport: a 0.8 MeV feletti energiájú neutronok,
2. csoport: a rezonanciák feletti energiák,
3. csoport: a rezonanciák tartománya,
4. csoport: termikus energiák.

Az SPNDYN képes kezelni több energiacsoportot is, én viszont csak kétszoport számításokat végeztem vele, mivel az elvégzett benchmarkok arra vonatkoztak.

Ha stacionárius számításokat szeretnénk végezni, akkor virtuálisan kritikussá tesszük a reaktort, azaz  $\frac{\partial \Phi(\underline{r}, E, t)}{\partial t} := 0$ , cserébe bevezetjük az effektív sokszorozási tényezőt:  $\nu \rightarrow \frac{\nu}{k_{\text{eff}}}$ . Ezzel megkaptuk a megoldandó diffúziós kevéscsoport differenciálegyenlet-rendszert:  $\forall g \in G$ -re

$$0 = \text{div} [D_g(\underline{r}) \text{grad} \Phi_g(\underline{r})] - [\Sigma_g^a(\underline{r}) + \Sigma_g^R(\underline{r})] \Phi_g(\underline{r}) + \sum_{g' \neq g} \Sigma_{g' \rightarrow g}^s \Phi_{g'}(\underline{r}) + \frac{f_g}{k_{\text{eff}}} \sum_{g'=1}^G \nu \Sigma_{g'}^f \Phi_{g'}(\underline{r}). \quad (8)$$

Időfüggő számítás esetén, az időfüggő tag megtartása mellett, a későneutronokkal is foglalkozni kell, azaz ekkor a későneutronanyag-koncentrációkra vonatkozó egyenletekkel kapcsolt időfüggő diffúzióegyenletet kell megoldanunk:

$\forall g \in G$ -re

$$\frac{1}{v_g} \frac{\partial \Phi_g(\underline{r}, t)}{\partial t} = \text{div} [D_g(\underline{r}, t) \text{grad} \Phi_g(\underline{r}, t)] - [\Sigma_g^a(\underline{r}, t) + \Sigma_g^R(\underline{r}, t)] \Phi_g(\underline{r}, t) + Q_g(\underline{r}, t). \quad (9)$$

$$Q_g(\underline{r}, t) = \sum_{g' \neq g} \Sigma_{g' \rightarrow g}^s \Phi_{g'}(\underline{r}) + (1 - \beta) f_g \sum_{g'=1}^G \nu \Sigma_{g'}^f \Phi_{g'}(\underline{r}) + \sum_{i=1}^N f_{gi} \lambda_i C_i(\underline{r}, t). \quad (10)$$

$\forall i \in K$ -ra

$$\frac{\partial C_i(\underline{r}, t)}{\partial t} = -\lambda_i C_i(\underline{r}, t) + \beta_i \sum_{g'=1}^G \nu \Sigma_{g'}^f \Phi_{g'}(\underline{r}). \quad (11)$$

Ennek az egyenletrendszernek a megoldását az SPNDYN térben folytonos Galerkin-módszeren alapuló végeelem-modellezéssel, időben pedig végesdifferencia, azon belül is  $\theta$  módszerrel végzi el. (Megjegyzés: térbeli megoldásra az SPNDYN rendelkezik hibrid-végelem modullal is, de én azt most nem használtam.)

A végeelem-módszer lényege, hogy a vizsgált térrészt elemekre osztjuk, a keresett megoldásfüggvényt pedig az ezeket meghatározó csomópontokban értékeljük ki, a köztes részekben pedig ezek alapján interpolálunk. Az egyenleteknek a gyenge alakját írjuk föl, azaz egy súlyfüggvénnyel besorozva kiintegráljuk őket az egyes elemeken belül. A Galerkin-módszer esetén ez a súlyfüggvény megegyezik az interpolációs polinommal. Itt a nullára rendezett egyenletalakot integráltuk ki, azonban nem azt követeljük meg, hogy az integrál minden elemre legyen nulla, hanem itt egy maradéktaggal tesszük egyenlővé az integrált. Ezután az így kapott elemmátrixokból felépítjük a globális mátrixegyütthatókat, mely során megköveteljük, hogy a maradéktagok járuléka globálisan legyen minimális. Ezután a megoldás során még különböző peremfeltétel-típusokat is fegyelembbe lehet

venni. Ezzel meg is kaptuk a megoldandó lineáris algebrai egyenletrendszer. A megoldás során az SPNDYN fix forrással kiszámolja a neutronfluxus-eloszlást (belső iteráció), mely egy lineáris algebrai egyenletrendszer megoldását jelenti. A rendszer méretéből adódóan ez csak iteratív úton lehetséges. (Az optimális iteratív megoldó kiválasztásával a számítási időigény jelentősen csökkenthető, lásd [2]-ben végzett vizsgálataim.) Ezután a neutronfluxus-eloszlás alapján stacionárius számítás esetén meghatározza  $k_{\text{eff}}$ -et, illetve újraszámolja a forrást (külső iteráció, hatványiteráció). Az új forrással aztán újabb fluxust számol, amiből pedig forrást ... Mindezt addig, amíg a lépések közt az egyes mennyiségek változása kisebb nem lesz egy előre beállított konvergencia-kritériumnál. Az iteráció konvergenciáját, illetve  $k_{\text{eff}}$  számítását a reaktorfizikai normálás biztosítja.

Az időbeli diszkretizálás során az időbeli differenciáloperátort egyszerűen differenciáhányadossal közelítjük, tehát tulajdonképpen egy végesdifferencia módszerről van szó, annyi különbséggel, hogy a  $\theta$  módszerben a  $p$  időpillanatbeli értékeket  $(1 - \theta)$ -val,  $p+1$ -beli értékeket  $\theta$ -val szorozzuk. Ha tehát  $\theta=0$ , akkor az explicit Euler séma adódik, ha  $\theta \in (0,1]$ , akkor implicit sémát kapunk. Speciális esete a  $\theta = 0.5$ , vagyis a Crank-Nicolson séma, mely másodrendű konvergenciájú, így a számítások során én ezt használtam.

A tranziens számításokhoz szükség van kezdeti feltételre is, mely az esetek többségében egy kritikus reaktorállapot. Az SPNDYN ezt úgy valósítja meg, hogy a kapott hálóval és csoportállandókkal először egy stacionárius számítást végez, melynek eredményeként kapott neutronfluxus-eloszlás lesz a kezdeti feltétel, illetve az összes anyag és energiacsoport hasadási hatáskeresztmetszetét leosztja a statikus effektív sokszorozási tényezővel, ezzel biztosítva a kritikus kezdeti állapotot.

Természetesen nem az SPNDYN az első kód, amely diffúziós közelítést implementál. Az első ilyen NTI-ben fejlesztett kód a DIMITRI [4] volt. Ez háromdimenziós, időfüggő, diffúziós számításokra alkalmas és mind térben, mind időben végesdifferencia-sémát használ. Egy NTI-s diplomamunka keretében, szintén a témavezetőm vezetésével jött létre a DIREMO [5-7], mely szintén háromdimenziós, időfüggő, diffúziós számításokat tesz lehetővé. Ez az SPNDYN-hez hasonlóan térben végeselem-módszert alkalmaz, időben pedig végesdifferencia-sémát. Érdekessége, hogy a reaktorfizikai mellett termikus és mechanikai modullal is rendelkezik, azaz kifejezetten alkalmas különböző visszacsatolások vizsgálatára. Csatolt termohidraulikai folyamatok vizsgálatára képes az Atomenergia Kutatóintézet által fejlesztett KIKO3D [8] kód is, mely transzport- vagy diffúziós közelítésben old meg háromdimenziós, időfüggő problémákat, szintén nodális módszerekkel. Magyar viszonylatban kiemelendő még a C-PORCA [9] kód, mely a Paksi Atomerőmű Zrt. saját fejlesztése, és a négy atomerőművi blokk valós idejű monitorozását támogatja. Érdekessége, hogy háromdimenziós, időfüggetlen, diffúziós számításokat végez úgynevezett hibrid végeselem-módszerrel. Ez azt jelenti, hogy radiálisan végeselem-módszert használ, axiálisan viszont analitikusat. Hogy külföldi példát is említsek, a nemzetközi, nyílt forráskódú Milonga [10] kód is háromdimenziós, időfüggő, diffúziós számításokat végez, melyhez az SPNDYN-hez hasonló végeselem-módszer mellett a végestérfogatok módszerét is képes használni.

## 2.2. Az SPNDYN rövid története és jelenlegi képességei

Mint arról az előző fejezetben szó volt a BME Nukleáris Technikai Intézetében régóta folyik reaktorfizikai kódfejlesztés. Ez kezdetben csak diffúziós közelítést jelentett, a számítógépes kapacitások növekedésével viszont Monte-Carlo-módszeren alapuló kódokkal is elkezdtek számításokat végezni, illetve felmerült az igény egy diffúziósnál magasabb rendű determinisztikus transzportközelítés implementálására is. Utóbbi megvalósítása témavezetőm doktori munkájaként [2] kezdődött el. Az SPNDYN fejlesztése kezdetben az időfüggetlen  $SP_3$  egyenletrendszer megoldására irányult, végeelem-módszerrel. Később a térbeli diszkretizációs lehetőség ki lett egészítve egy a C-PORCA kódban alkalmazotthoz hasonló hibrid-végelem modullal is. Aztán készült hozzá egy időfüggetlen diffúziós modul is, hogy a diffúziós és  $SP_3$  számítások pontossága összehasonlítható legyen. Az időfüggés kezelése tehát nem volt cél, viszont különböző hallgatói munkák során időfüggő diffúzióegyenletet megoldó kódok is készültek az NTI-ben, melyeket támogatandó az SPNDYN is ki lett egészítve egy időfüggő diffúziós és  $SP_3$  végeelem modullal. Most tehát az SPNDYN képes a diffúziós vagy  $SP_3$  közelítést alkalmazni végeelem vagy hibrid-végelem módszerrel stacionárius számítás esetén, illetve diffúziós/ $SP_3$  közelítést és végeelem-módszert időfüggő számítás esetében.

Jelenleg az NTI részvételével zajló ECC-SMART projekt kapcsán folyamatban van az SPNDYN csatolása az APROS termohidraulikai rendszerkóddal. A két program közti kommunikáció már ki lett építve, a csatolás befejezéséhez már csak egyes részprogramok hiányoznak, mint például egy hatékony teljesítményszámító modul, így a csatolás várhatóan hamarosan elkészül. Ezzel pedig az SPNDYN komolyabb elemzések elvégzésére is alkalmassá válhat majd, illetve megnyílik a lehetőség a tranziensek és visszacsatolások modellezésére is.

Azonban mielőtt az elkészült csatolt kódrendszerrel számításokat végezhetnénk, először meg kell róla győződni, hogy az egyes részegységek jól működnek-e, ennek részeként pedig az SPNDYN diffúziós végeelem modulját is verifikálni kell. Ezt célozták a szakdolgozatom keretében végzett VVER reaktortípusra vonatkozó stacionárius számítások, az időfüggő modul verifikációja viszont még esedékes.

Emellett azt is fontos megjegyezni, hogy a stacioner modul korábban már rengeteg esetben bizonyított, az én feladatom specifikusan VVER reaktorokra vonatkozó diffúziós benchmarkszámítások esetében történő programtesztelésre terjedt ki. Ezzel szemben az időfüggő modul teljeskörű tesztelése mindeddig még nem történt meg. Eredményeinek analitikus megoldásokkal történő összehasonlítása megtörtént, illetve a BME Oktatóreaktorán történt mérések is lettek már modellezve vele [11], de olyan komplex feladatot, mint egy VVER-440 reaktorban lezajló tranziens még nem végeztek vele. Így a jelen TDK dolgozat keretében általam elvégzett munka nem csak a VVER reaktortípusra történő verifikációról szól, hanem az ilyen komplexitású feladatoknál óhatatlanul is felmerülő problémák megoldásáról is, hogy mire az APROS-SPNDYN csatolás elkészül, az időfüggő diffúziós modul átfogóbban tesztelve legyen.

### 2.3. A teszteléshez használt benchmarkok

Jelen dolgozat keretében specifikusan VVER-440 típusú reaktorokra vonatkozó benchmarkfeladatokat mutatok be. A végrehajtott benchmarkok forrása az AER (Atomic Energy Research) [12], mely egy kilenc ország 27 intézményét (köztük egyetemeket, erőműveket és kutatóközpontokat) tömörítő szervezet. Célja a VVER típusú reaktorok fejlesztéséhez és biztonságos üzemeltetéséhez szükséges kutatások koordinálása, az elért eredmények megosztása. Az AER számos VVER reaktorokra vonatkozó dinamikus és statikus benchmarktesztet dolgozott már ki, melyek specifikációi a KFKI<sup>1</sup> [13] weboldaláról elérhetőek. Ezek közül lettek kiválasztva a következőkben ismertetett statikus és dinamikus benchmarkproblémák. Hozzájuk általában több referenciaeredmény is elérhető, melyek szintén diffúziós számítások eredményei, így valóban jó összehasonlítási alapot jelentenek az SPNDYN-nek.



Az elérhető statikus benchmarkok mindegyike végrehajtható az SPNDYN-nel, viszont az előző részben ismertetett limitációk miatt a 6 elérhető dinamikus benchmarkból jelenleg még csak egyet sikerült végrehajtani, így ezen a ponton érdemesnek látom bemutatni, hogy miről is szólnak ezek a dinamikus benchmarkok és milyen továbbfejlesztésekre lenne szükség a végrehajtásukhoz. Mindegyik dinamikus benchmark VVER-440 reaktort modellez három dimenzióban, így ezt nem fogom külön kiemelni.

Megjegyzés: Az összes ismertetett és végrehajtott benchmark specifikációja elérhető a [13] weblapról, így a továbbiakban ezekre külön nem fogok hivatkozni.

A Dyn001 benchmark egy aszimmetrikus szabályozókazetta-kilökődést modellez. A tranziens kritikus állapotról, de alacsony teljesítményről indul, a kilökődést pedig vészleállítás követi, azaz a szabályzó és biztonságvédelmi kazetták elindulnak lefelé a zónában. A tranziens során a teljesítmény nem megy túl magasra, így visszacsatolások figyelembe vételére nincs szükség. A benchmark elvégzéséhez a geometria és a tranziens adatai mellett a szükséges csoportállandók is adottak.

A Dyn002 az előzőhöz hasonló aszimmetrikus szabályozókazetta-kilökődést modellez, viszont ebben nincs vészleállítás, így a teljesítmény jelentősen megnő. Az üzemanyagból nincs hőelvonás, így a visszacsatolások csupán egy egyszerű képlettel megadott Dopplervisszacsatolás formájában vannak figyelembe véve.

A Dyn003 szintén aszimmetrikus szabályozókazetta-kilökődést modellez, viszont már minden visszacsatolást figyelembe véve, azaz ehhez már teljes termohidraulikai modellezés szükséges. A geometrián, csoportállandókon és a tranziens adatain kívül itt a termohidraulikai paraméterek is adottak.

A Dyn004 benchmark kezdeti állapota egy leállított reaktor (minden szabályozókazetta lent, erősen bórsavas víz). A tranziens során egy véletlen szivattyúbeindulás hideg, bórsavmentes vizet juttat a zónába, ami így prompt szuperkritikussá válik. Ezután a zónába ismét bórsavas víz kerül, mely hatására a zóna mélyen szubkritikus állapotba kerül. Ehhez a benchmarkhoz már csak a geometria, a termohidraulikai paraméterek, illetve a tranziens két állapota közt lévő reaktivitáskülönbség az adott, a csoportállandókat a tesztet végzőknek kell meghatározniuk.

<sup>1</sup>Magyar Tudományos Akadémia, Központi Fizikai Kutatóintézet

A Dyn005 és Dyn006 benchmarkok kezdeti állapota egy teljes teljesítményen üzemelő reaktor, a tranzienszt pedig 200%-os csőtörés okozza. (A két benchmark a csőtörés helyében tér el.) Ezeknél már nem elegendő a zónát modellezni, hanem teljes primer,- illetve szekunderköri modell szükséges. Ezek geometriai és termohidraulikai adatai adottak, azonban a csoportállandókat a tesztet végzőknek kell meghatározniuk.

Ezek alapján egyértelmű, hogy az utolsó négy benchmark majd akkor lesz releváns, ha elkészült az SPNDYN APROS-hoz való csatolása. A Dyn002 benchmark realizálásához arra lenne szükség, hogy az SPNDYN minden időlépésben kiszámolja a teljesítményeloszlást, illetve a zóna összteljesítményét. A teljesítmény számítása jelenleg csak a számítás után, egy másik programmal az eredményfájlok alapján történik, azaz ennek kivitelezése a programkód módosítását igényli. Ez önmagában nem lenne megoldhatatlan feladat, a gondot, mint azt egy későbbi fejezetben látni fogjuk, a teljesítményszámítás végeselem-módszer miatti komplexitása okozza. Így maradt a Dyn001, mint az egyetlen SPNDYN jelenlegi verziójával megoldható dinamikus benchmarkfeladat.

### 3. Számítási eredmények

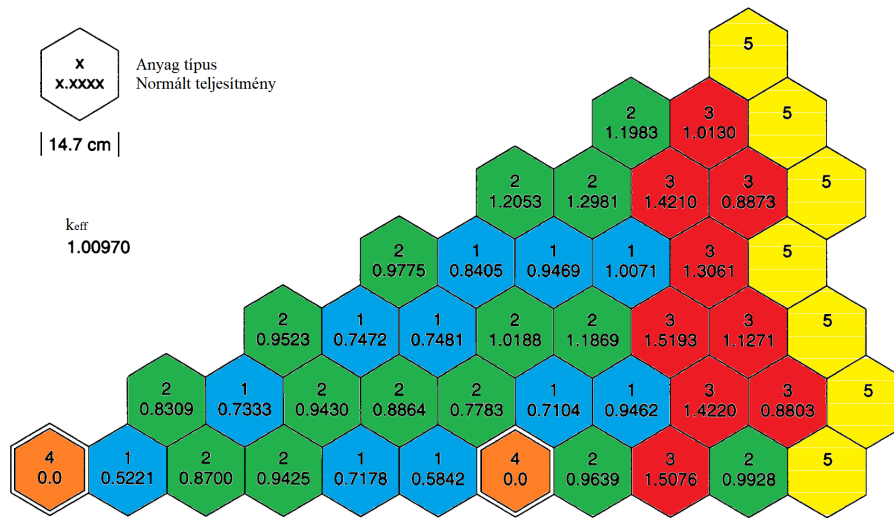
Az SPNDYN diffúziós moduljának VVER reaktorokra történő verifikációjához két benchmarkproblémát mutatok be ebben a részben. Az előző fejezetben kiválasztott dinamikus benchmark előtt annak stacioner verzióját is bemutatom, ugyanis annál sokkal részletesebben végig lehetett menni a benchmarktesztelés fő lépésein, illetve jól szemlélteti, hogy még egy stacionárius számítás is mennyire erőforrásigényes tud lenni a folytonos végelelem-módszert alkalmazó modullal. A tesztelést ennél hálófüggelenségi vizsgálattal kezdtem, majd az ennek eredményeként a legsűrűbb hálónál megkapott eredményeket vettem össze a referenciamegoldásokkal. A dinamikus benchmarknál az eredmények referenciamegoldással való összevetése mellett a felmerült problémák megoldása is nagy hangsúlyt fog kapni.

#### 3.1. Seidel benchmark

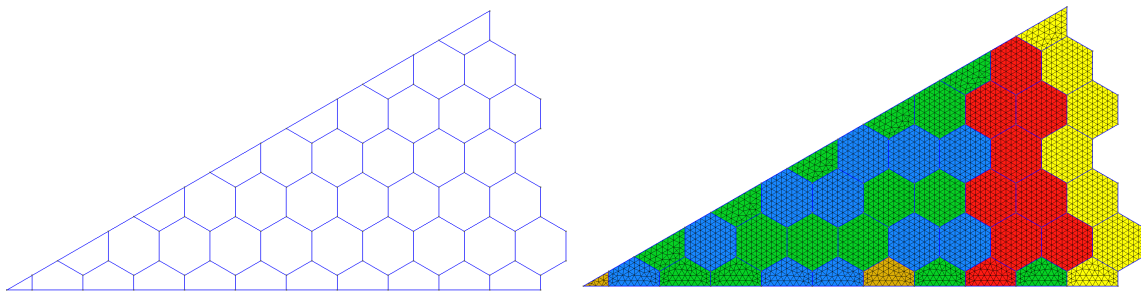
Először nézzük tehát az AER-FCM-001, vagyis Seidel benchmarkot, mely rendelkezik két és háromdimenziós változattal is. A kétdimenziós variáns lényegesen kisebb számítási kapacitásigénnyel rendelkezik, így megoldása során több paraméter vizsgálatára is lehetőség van. Ennél a benchmark részleteinek ismertetése és a hálófüggelenségi vizsgálat mellett az SPNDYN-ben alkalmazott konvergenciakritériumoktól való függést is vizsgálni fogom az eredmények diszkussziója előtt. Ezzel szemben a háromdimenziós változat realisabb képet ad az SPNDYN használhatóságáról, viszont a lényegesen nagyobb gépigény miatt kevesebb paraméter vizsgálható. Itt ismertetni fogom a benchmark részleteit, elvégzem a diszkretizációs hiba minimalizálását, majd a megmaradó hibák mellett kiértékelem az eredményt. Fontos, hogy ez a számítás még a szakdolgozatom [2] keretében lett elvégezve, és a szimulációk futtatásához ekkor csak a saját számítógépes kapacitásom állt rendelkezésre.

##### 3.1.1. A kétdimenziós modell specifikációja

A Seidel 2D benchmark egy VVER-440 blokkot modellez két dimenzióban. A zóna kazettánkénti homogén régiókból áll, melyekből ötféle van: három különböző üzemanyag, szabályozókazetta és reflektor. Ezek zónabeli elhelyezkedését, illetve a referenciamegoldást az 1. ábra szemlélteti. A VVER-440 blokk szimmetriája miatt elegendő a zóna 1/12 részét modellezni, amivel lényegesen csökkenthető a szimuláció számításigénye. Ennek megfelelően én is csak egy egy 1/12 zónarészletet modelleztem Gmsh-ban, mint ahogyan az a 2. ábrán is látható.



1. ábra. A Seidel 2D benchmark specifikációja.

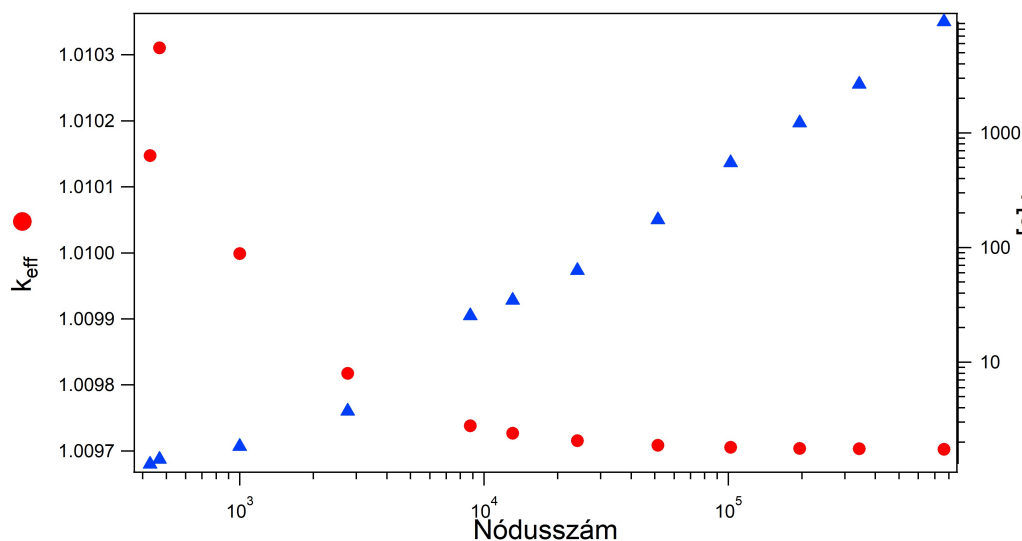


2. ábra. A Gmsh-val létrehozott geometria és háló a Seidel 2D benchmarkhoz.

### 3.1.2. Hálófüggetlenség vizsgálata a kétdimenziós modellen

A szimuláció végeredménye nem csak a szimulációhoz használt kódtól (ezáltal az alkalmazott fizikai, matematikai és numerikus modelltől), de a végelem-számításokhoz alkalmazott háló sűrűségétől is függhet. A diszkretizációból eredő számítási hiba minimalizálása érdekében először hálófüggetlenségi vizsgálatot kell végezni. Ehhez a Gmsh-ban a karakterisztikus elemméret változtatásával különböző sűrűségű hálókat generáltam. Ezután sorban mindegyikkel lefuttattam a szimulációt és figyeltem, hogyan változik az effektív sokszorozási tényező értéke. Az eredmények a 3. ábrán és az 1. táblázatban láthatóak.





3. ábra. A háló finomításának hatása a számítási időigényre és az effektív sokszorozási tényezőre a Seidel 2D benchmark esetében.

A benchmarkok specifikációjában ez PCM (azaz  $10^{-5}$ ) pontossággal van megadva, azaz a hálózfüggetlenség során is az a cél, hogy az eredmény ekkora pontosságúra kerekített értéke már ne változzon. Vizsgáltam a teljesítményeloszlást is, mely tekintetében a cél az, hogy két egymást követő hálónál a normált teljesítmények maximális változása kisebb legyen 0.1%-nál. Látható, hogy ezek a feltételek a két legsűrűbb háló esetén már bőven teljesülnek. Igazából már elég lett volna hamarabb is abbahagyni a vizsgálatot, de most szerettem volna demonstrálni, hogy egy ponton túl fölösleges sűríteni a hálót, hiszen nem kapunk érdemben pontosabb eredményt, viszont, mint az a 3. ábrán is jól látszik, a futásidő jelentősen megnő.

Em.	Nódusszám	$k_{\text{eff}}$	Futásidő [s]	$N_{\text{it}}$	$\Delta k_{\text{eff}}$	Max. $\Delta P$
8	429	1.010147	1.286	115		
6	469	1.010310	1.416	111	-0.000162	2.107%
4	999	1.009999	1.836	106	0.000311	2.615%
2	2768	1.009817	3.721	102	0.000181	1.660%
1	8774	1.009738	25.337	100	7.902E-05	0.628%
0.8	13110	1.009726	34.532	100	1.157E-05	0.100%
0.6	24094	1.009715	63.033	100	1.128E-05	0.086%
0.4	51449	1.009708	173.99	100	7.033E-06	0.054%
0.3	102166	1.009705	548.68 (9 perc)	99	3.038E-06	0.032%
0.2	196188	1.009703	1221.29 (20 perc)	100	1.393E-06	0.012%
1.5	343954	1.009703	2654.85 (44 perc)	100	6.924E-07	0.004%
0.1	763329	1.009702	9311.07 (2.6 óra)	99	6.005E-07	0.012%

1. táblázat. Hálózfüggetlenség vizsgálata a Seidel 2D benchmark esetén.  
(Jelölés: E-5  $\equiv \cdot 10^{-5}$ , Em. = Elemméret)

### 3.1.3. Konvergenciaparaméterek vizsgálata a kétdimenziós modellen

A számítás végeredményét nem csak a geometria diszkretizációjának hibája befolyásolhatja, hanem a nem megfelelően megválasztott konvergenciakritériumok is. Ezekből négy darab van:

- Belső iteráció konvergenciája. Alapértelmezett kritérium:  $\Delta\phi_{\text{fix forrás}} < 10^{-8}$
- Külső iteráción belül:
  - $k_{\text{eff}}$  konvergenciája. Alapértelmezett kritérium:  $\Delta k_{\text{eff}} < 10^{-7}$
  - Fluxus konvergenciája. Alapértelmezett kritérium:  $\Delta\phi < 10^{-3}$
  - Maximális iterációs szám. Alapértelmezett kritérium: 500

Ezek közül a belső iteráció és  $k_{\text{eff}}$  kritériumát érdemes együtt kezelni, hiszen a külső iterációban nincs értelme pontosabb eredményt várni, mint amit a belsőben megköveteltünk. Ezért általában  $k_{\text{eff}}$  kritériuma egy nagyságrenddel kisebb szokott lenni. A fluxus kritériuma pedig ezeknél több nagyságrenddel nagyobb szokott lenni, melynek pedig az oka, hogy ez konvergál a leglassabban, így ha túl szigorú feltételt szabunk meg rá, akkor lényegesen megnő az iterációs lépésszám. A maximális iterációs számra azért van szükség, hogy ha a megoldás nem lenne konvergens, a program akkor se fusson a végtelenségig. E tekintetben az alapértelmezett érték vizsgálat nélkül is elfogadható.

Az eredmény ezen paramétereiktől való függését viszonylag nagy, de állandó diszkretizációs hiba mellett végeztem, így az eredmények reprezentatívak, de nem kellett órákat várni egy-egy szimuláció lefutására. Alapvetően két dolgot vizsgáltam:

1. Milyen hatással van, ha mégis szigorúbb feltételt szabunk meg a fluxus konvergenciájának?
2. Mekkora belső (és ezzel külső) konvergenciakritériumra van szükség?

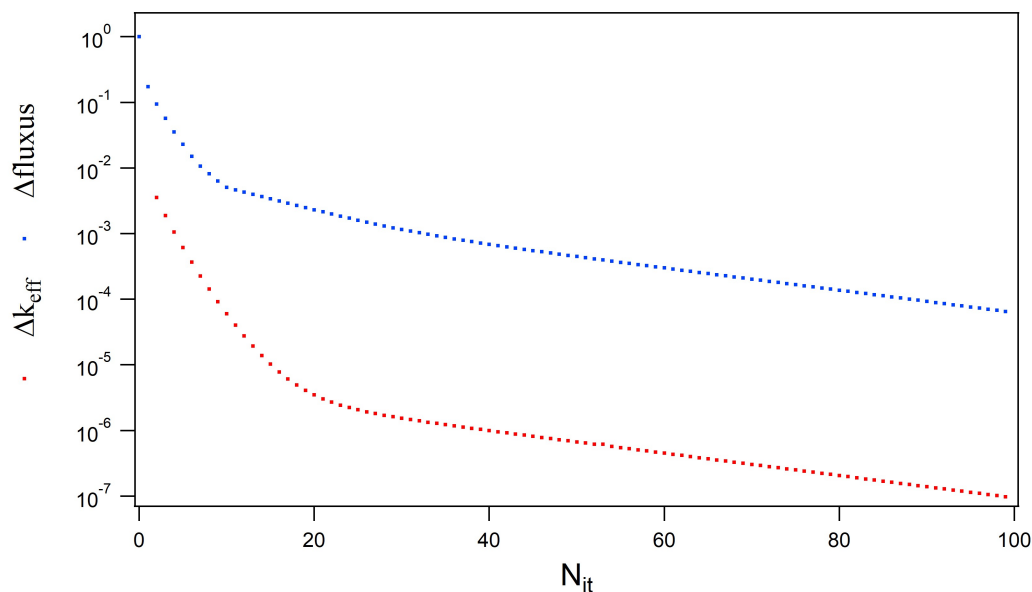
Az elvégzett számítások adatai és eredményei a 2. táblázatban láthatóak. Látható, hogy a fluxus kritériumának szigorításakor  $k_{\text{eff}}$  változása kisebb, mint az elvárt pontosság, viszont az iterációs lépésszám és ezzel a futásidő jelentősen nő. Azaz  $k_{\text{eff}}$  pontossága szempontjából elegendő az alapértelmezett fluxuskritérium. Meg kell azonban jegyezni, hogy a teljesítményeloszlás itt most nem lett vizsgálva, amit viszont befolyásolhat a fluxuskritérium.

Egy kicsit más a helyzet az iterációs kritériumokkal.  $k_{\text{eff}}$  változásának hullámzása azzal magyarázható, hogy egy darabig a fluxus kritériuma határozza meg a külső iterációs ciklus hosszát. Ekkor a különbség a belső iteráció pontosításával csak egyre kisebb mértékben változik. Ezután a külső iterációban a  $k_{\text{eff}}$ -re vonatkozó kritérium veszi át a leállító szerepet, ekkor a változásban egy nagyobb ugrás van. Ezután a belső és külső iteráció kritériumának további szigorításával már ismételtelen csak kis mértékben változik az eredmény. Ez pedig azt jelenti, hogy az alapértelmezett kritériumokat itt is elfogadhatjuk mint megfelelő kompromisszum a számítási pontosság és az elfogadható futásidő közt.

Kritériumok			Eredmények			
Belső	$k_{\text{eff}}$	Fluxus	$k_{\text{eff}}$	Futásidő [s]	$N_{\text{it}}$	$\Delta k_{\text{eff}}$
1.0E-08	1.0E-07	1.0E-03	1.0097154	63.2	100	
1.0E-08	1.0E-07	1.0E-04	1.0097154	59.9	100	0
1.0E-08	1.0E-07	1.0E-05	1.0097175	85.1	149	2.105E-06
1.0E-08	1.0E-07	1.0E-06	1.0097178	111.3	208	3.273E-07
1.0E-08	1.0E-07	1.0E-07	1.0097179	139.7	272	3.313E-08
1.0E-03	1.0E-02	1.0E-02	1.0093117	17.1	9	
1.0E-04	1.0E-03	1.0E-03	1.0096877	24.6	36	0.000376
1.0E-05	1.0E-04	1.0E-03	1.0096854	25.9	35	-2.283E-06
1.0E-06	1.0E-05	1.0E-03	1.0096850	27.8	34	-4.658E-07
1.0E-07	1.0E-06	1.0E-03	1.0096930	32.0	41	8.035E-06
1.0E-08	1.0E-07	1.0E-03	1.0097154	62.7	100	2.242E-05
1.0E-09	1.0E-08	1.0E-03	1.0097176	99.8	159	2.223E-06
1.0E-10	1.0E-09	1.0E-03	1.0097179	141.4	218	2.211E-07

2. táblázat. Konvergencia paraméterek vizsgálata a Seidel 2D benchmark 24094 nódusú hálóján.

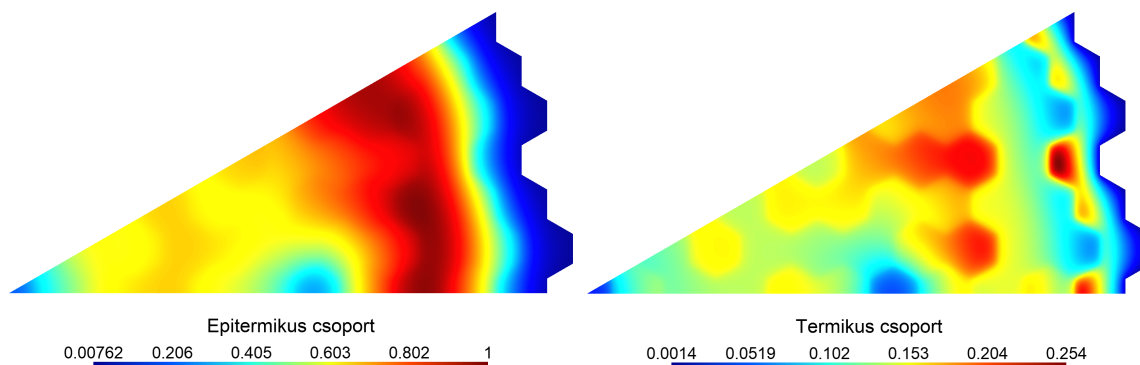
Mindez kicsit szemléletesebb, ha ábrázoljuk, hogyan alakul  $k_{\text{eff}}$  és a fluxus változása az iterációs lépésszám függvényében. Ezeket az értékeket az SPNDYN alából is kiírja a képernyőre, így nem volt nehéz módosítani rajta egy kicsit, hogy ezeket egy fájlba is kiírja, ahonnan már ábrázolhatóak. Az alapértelmezett kritériumokkal lefuttatott szimuláció esetén kapott grafikon a 4. ábrán látható. Ezen jól megfigyelhető, hogy a fluxus változása valóban jóval később ér el egy adott szintet, viszont az is megállapítható, hogy konvergenciájának állandósult üteme megegyezik  $k_{\text{eff}}$ -ével. Azaz csupán azért tűnik úgy, hogy a fluxus jóval lassabban konvergál, mert annál az állandósult ütemű szakasz egy nagyobb értékről kezdődik. A fluxus konvergenciája tehát gyorsítható, ha egy jó kezdeti fluxuseloszlás-bebecslést adunk meg az iteráció inicializálása során, amely most alapértelmezetten minden pontban egységnyi értéket vesz fel.



4. ábra. A konvergencia paraméterek változása az iteráció során a Seidel 2D benchmark esetében.

### 3.1.4. A kétdimenziós modell eredményei

Most, hogy beláttuk, hogy a diszkretizációs hiba minimális, és a konvergencia paraméterek is rendben vannak, rátérhetünk az eredmények diszkussziójára. A szimuláció végeredményének a legsűrűbb hálónál kapott értékeket vettem. Az effektív sokszorozási tényező  $k_{\text{eff}}=1.00970$  lett, ami megegyezik a referenciaeredménnyel. Ez alapján az SPNDYN igen jól szerepelt ezen a teszten. A másik összehasonlítható mennyiség a teljesítményeloszlás. Ezt az 5. ábrán látható neutronfluxus-eloszlások kazettánkénti kiintegrálásával és – a benchmark specifikáció szerint eljárva –  $\Sigma_f$ -fel beszorozva kaphatjuk meg. A referencia a normált teljesítményeket tartalmazza, ami annyit jelent, hogy az egyes kazetták teljesítményét elosztjuk az átlagos kazettateljesítménnyel. Ezek az értékek a referenciával összehasonlítva az A függelékben lévő 4. táblázatban találhatóak.



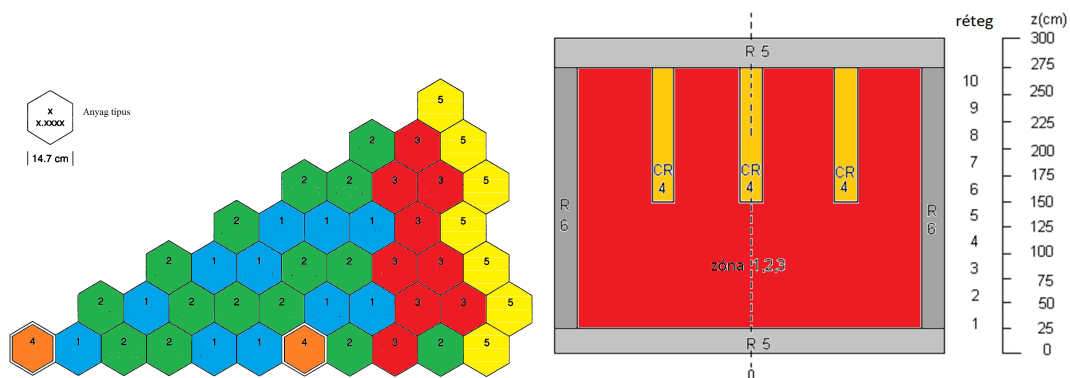
5. ábra. A Seidel 2D benchmark kiszámított neutronfluxus-eloszlásai a gyorscsoport fluxuseloszlás maximumával normálva.

Az eredményeket végignévezve megállapítható, hogy az SPNDYN ezen a téren is kiemelkedően jól teljesített. Az átlagos eltérés csupán 0.045%, és a maximális eltérés is csak 0.11% volt. Ennél kisebb eltérés nem igazán lenne elvárható, hiszen itt nem csupán a kód pontossága számít, hanem például az eredmények feldolgozása során alkalmazott numerikus integrálás is hozhat be valamekkora hibát.

Összességében tehát megállapíthatjuk, hogy az SPNDYN kiállta ezt a benchmark-tesztet.

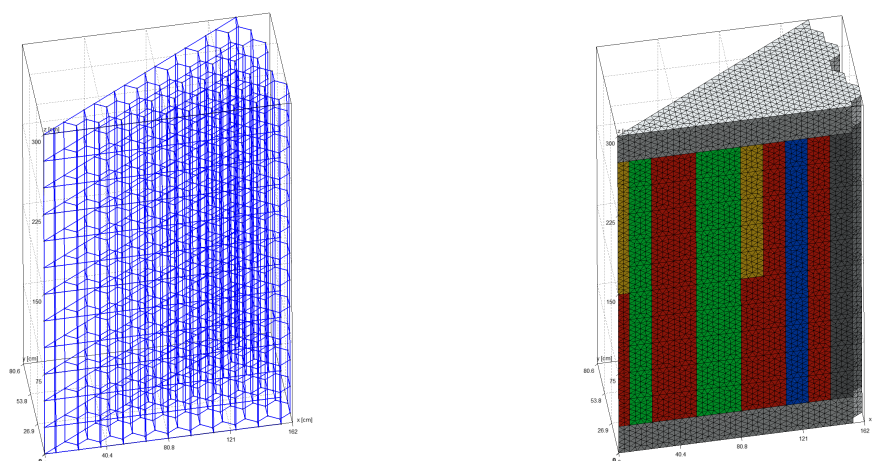
### 3.1.5. A háromdimenziós modell specifikáció

A Seidel 3D benchmark szintén egy VVER-440 reaktor zónáját modellezi, de már három dimenzióban. A zóna háromféle üzemanyagot és egyféle abszorbert tartalmaz, illetve alul és fölül is reflektorral van borítva, de eltérő anyaggal, így összesen hatféle homogén régió van. A szabályozókazetták félig vannak betolva a zónába. A modellezendő reaktor radiális és axiális metszete a 6. ábrán látható.



6. ábra. A Seidel 3D benchmark specifikációja.

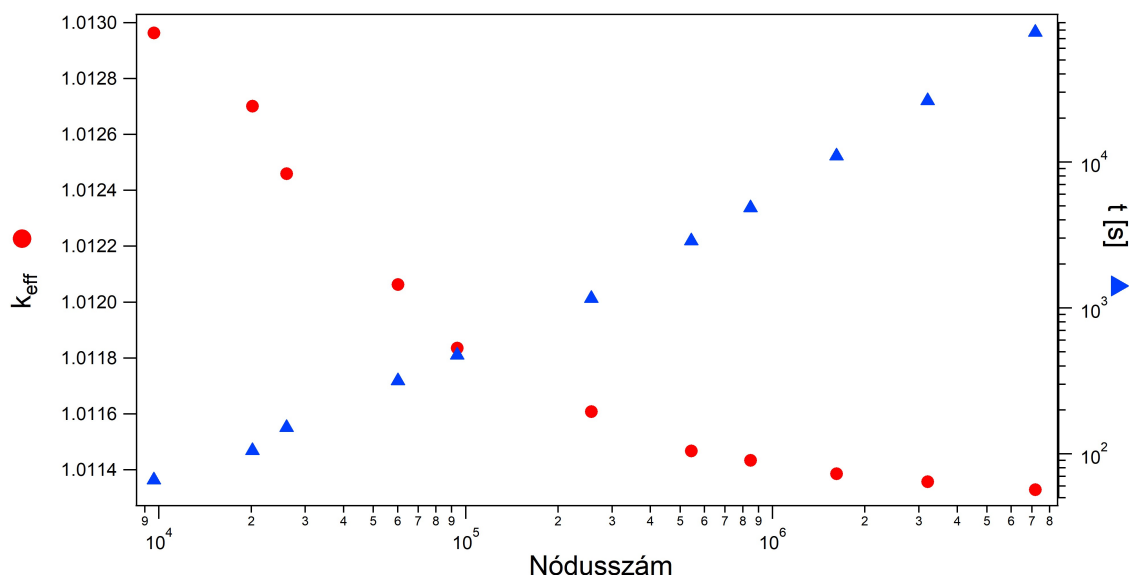
A VVER-440 blokk szimmetriája itt is megvan, azaz elegendő a zóna 1/12 részét modellezni, amivel lényegesen csökkenthető a szimuláció számításigénye. Ennek megfelelően én is csak egy ekkora részt készítettem el Gmsh-ban, mint ahogyan az a 7. ábrán is látható.



7. ábra. A Gmsh-val létrehozott geometria és háló a Seidel 3D benchmarkhoz.

### 3.1.6. Hálófüggetlenség vizsgálata a háromdimenziós modellen

A szimuláció végeredménye nem csak a szimulációhoz használt kódtól (ezáltal az alkalmazott fizikai, matematikai és numerikus modelltől), de a végeselem-számításokhoz alkalmazott háló sűrűségétől is függhet. A diszkretizációból eredő számítási hiba minimalizálása érdekében először hálófüggetlenségi vizsgálatot kell végezni. Ehhez a Gmsh-ban a karakterisztikus elemméretet változtattam és különböző sűrűségű hálókat generáltam. Ezután sorban mindegyikkel lefuttattam a szimulációt és figyeltem, hogyan változik az effektív sokszorozási tényező értéke. Az eredmények a 8. ábrán és a 3. táblázatban láthatóak.



8. ábra. A háló finomításának hatása a számítási időigényre és az effektív sokszorozási tényezőre a Seidel 3D benchmark esetében.

A cél a  $10^{-5}$ -es (1 pcm) pontosság elérése lett volna, azonban ez ennyi lépésből nem sikerült. Látható, hogy  $k_{\text{eff}}$  értéke konvergál egy hálófüggetlen értékhez, azonban nagyon lassan. Az eléréséhez még több hálósűrítési lépésre lenne szükség, ezek lefuttatásához azonban már nem áll rendelkezésemre elegendő számítási kapacitás. Látható, hogy az utolsó, 7 millió nódusú háló esetén a számítás időigénye már 21 óra volt. Itt a rendelkezésre álló 16 GB RAM már nem csak a globális mátrixegyütthetők felépítésénél, de a hatványiteráció során is maximálisan ki volt használva.

Megjegyzés: a teljesítményeloszlás konvergenciája általában szigorúbb kritérium  $k_{\text{eff}}$  konvergenciájánál, ráadásul számítása szintén időigényes, így azzal a hálófüggetlenség vizsgálata során az ismert limitációk miatt nem volt értelme foglalkozni.

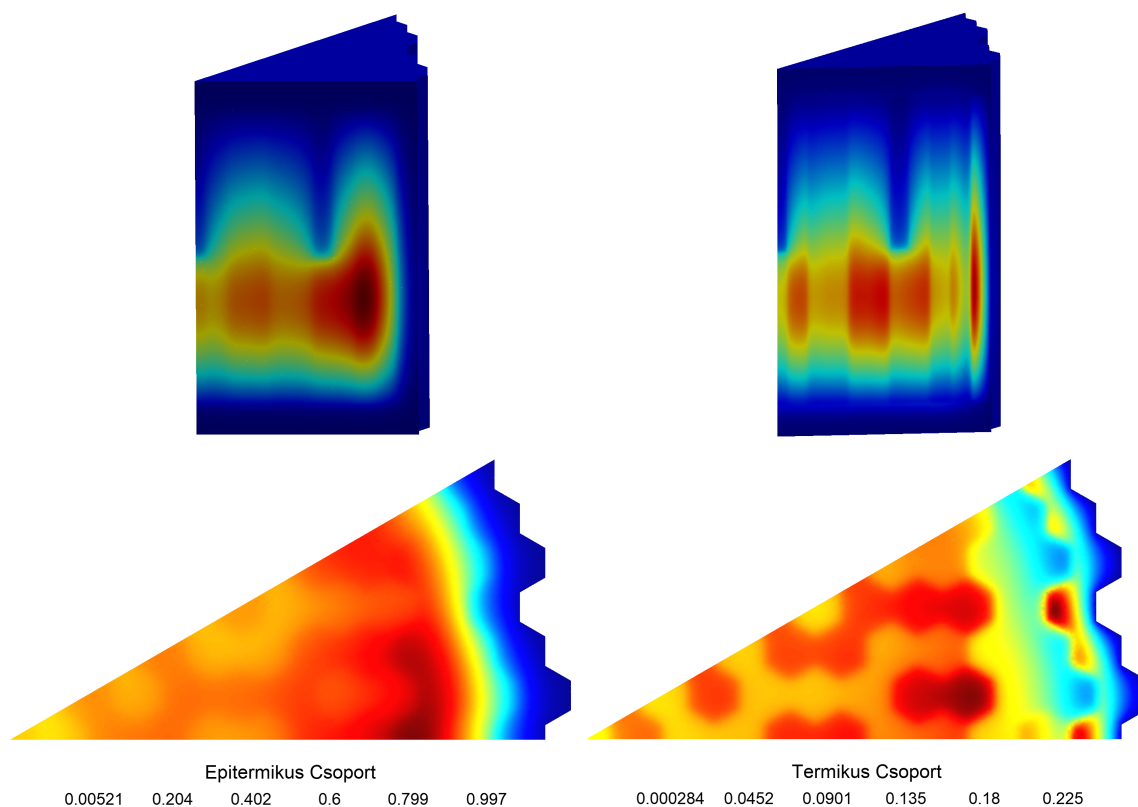
Emellett az is megjegyzendő, hogy pusztán a háló legenerálása is közel egy órát vett igénybe, szintén maximális erőforrás kihasználtság mellett. Mindez azt jelenti, hogy bár még nem értük el a hálófüggetlenséget, ilyen számítógépes kapacitás mellett nincs értelme folytatni a vizsgálatot. Azaz vagy beérjük ezzel a pontossággal, vagy nagyobb számítógépes kapacitásra kell váltani.

Elemméret	Nódusszám	$k_{\text{eff}}$	Futásidő [s]	$N_{\text{it}}$	$\Delta k_{\text{eff}}$
10	9630	1.012962	66.08	230	
8	20168	1.012701	105.05	232	0.000262
6	26069	1.012459	151.27	232	0.000241
4	60150	1.012063	315.27	232	0.000397
3	93922	1.011834	473.62	232	0.000229
2	256641	1.011608	1157.05	232	0.000227
1.5	544076	1.011467	2876.75	233	0.000141
1.25	848893	1.011433	4845.6	233	3.384E-05
1	1617807	1.011385	10994.04 (3 óra)	233	4.780E-05
0.8	3207606	1.011357	26309.3 (7 óra)	233	2.819E-05
0.6	7186332	1.011328	77233.3 (21 óra)	233	2.856E-05

3. táblázat. Hálófüggetlenség vizsgálata a Seidel 3D benchmark esetén.  
(Jelölés: E-5  $\equiv \cdot 10^{-5}$ )

### 3.1.7. A háromdimenziós számítási eredmények kiértékelése

Ezen opciókból nézzük az előbbit, azaz a továbbiakban az általam lefuttatott legsűrűbb, 7 millió nódusú hálóval kapott eredményeket ismertetem. Az eredmény  $k_{\text{eff}}=1.011329$  lett, ami +0.4 pcm-el tér el a referenciától (ami  $k_{\text{eff}}=1.011325$ ). Ez kifejezetten jó tekintve, hogy ez még nem a hálófüggetlen megoldás. Emellett megemlítendő, hogy a tendencia az volt, hogy a háló sűrítésével folyamatosan közelebb került az eredmény a referenciamegoldáshoz. Ez olyan szempontból jó, hogy ha ez már a hálófüggetlenség előtti utolsó lépés lenne, akkor a következő lépésben megkapott hálófüggetlen megoldás is 1 pcm-en belül lenne a referenciához képest. Viszont ha még több lépés kell a diszkrétizációs hiba minimalizálásához, akkor az is elképzelhető, hogy a referenciától való eltérés átmejj negatív irányúba és nőni kezd. Azaz a háló további sűrítésével ennél akár nagyobb eltérést is tapasztalhatnánk. Ilyen szempontból tehát eléggé szerencsétlen, hogy éppen itt kellett az említett számítógépes kapacitáslimitok miatt abbahagyni a vizsgálatot.



9. ábra. A Seidel 3D benchmark kiszámított neutronfluxus-eloszlásaia a gyorscsopor fluxuseloszlás maximumával normálva.

(Metszetek 120 cm-es magasságban, elemméret = 1-nél kapott eredmények.)

A 9. ábrán szemléltetett neutronfluxus-eloszlások alapján számított teljesítményeloszlás tekintetében is hasonló a helyzet, melynek referenciától való eltérése az A függelékben lévő 5. táblázatban látható. A rétegenkénti eloszlásban a maximális eltérés 0.49%, ami kifejezetten jó egyezésnek számít, tekintettel arra is, hogy a rendelkezésemre álló számítógépes kapacitással nem tudtam hálózfüggetlen eredményt elérni. Az abszolút értékben számolt átlagos eltérés viszont csak 0.06%, ami jelzi, hogy az SPNDYN tulajdonképpen itt is jól teljesít, csak a nem kielégítően sűrű hálózval van a probléma. A kazettánkénti eloszlásban egy picit jobb is a helyzet, ott a maximális eltérés 0.24%, az átlagos eltérés pedig csak 0.03%.

Összességében tehát elmondható, hogy az SPNDYN ezen a teszten is jól teljesített. Persze, hogy ez a kijelentés valóban megállja a helyét, a hálózfüggetlenség további vizsgálatára, és így a diszkretizációs hibától mentes megoldás kiértékelésére lenne szükség. Ehhez viszont nagyobb számítási kapacitásra van szükség.



## 3.2. Dyn001 visszacsatolásmentes időfüggő benchmark

Az előzőekben láttuk tehát, hogy hogyan néz ki és milyen nehézségekkel jár egy VVER-440 reaktor modellezése stacionárius esetben egy tisztán végeselem-módszert alkalmazó kóddal. Most térjünk rá az időfüggő modul tesztelésére a Dyn001 benchmarkfeladaton keresztül. Ez a rész már teljes egészében jelen félévben készült és a hosszabb szimulációk lefuttatásához egy egyetemi szervergépet is igénybe tudtam venni. Ebben a fejezetben először bemutatom a benchmarkhoz tartozó geometriai specifikációt, illetve a tranziens lefolyását és a csoportállandók ennek megfelelő paraméterezését. Ezt követően megnézzük, hogy milyen problémák merültek fel és ezekre milyen megoldásokat sikerült találni. Végezetül a kapott eredményeket összehasonlítom a referenciamegoldásokkal és levonom a következtetéseket.

### 3.2.1. Geometria

A Dyn001 benchmark tehát szintén egy VVER-440 zónát modellez, az anyagok elrendezése is nagyon hasonló a statikus esetben látotthoz. Háromféle üzemanyag van (1.6%, 2.4% és 3.6% dúsítású), egyféle abszorber és radiálisan, illetve axiálisan eltérő típusú reflektor. Ezek zónabeli elrendezése a 10. ábrán látható. A kezdeti állapotban a szabályzókazetták a zóna alja fölött 50 cm-rel helyezkednek el, a biztonságvédelmi rudak viszont teljesen ki vannak húzva a zónából.

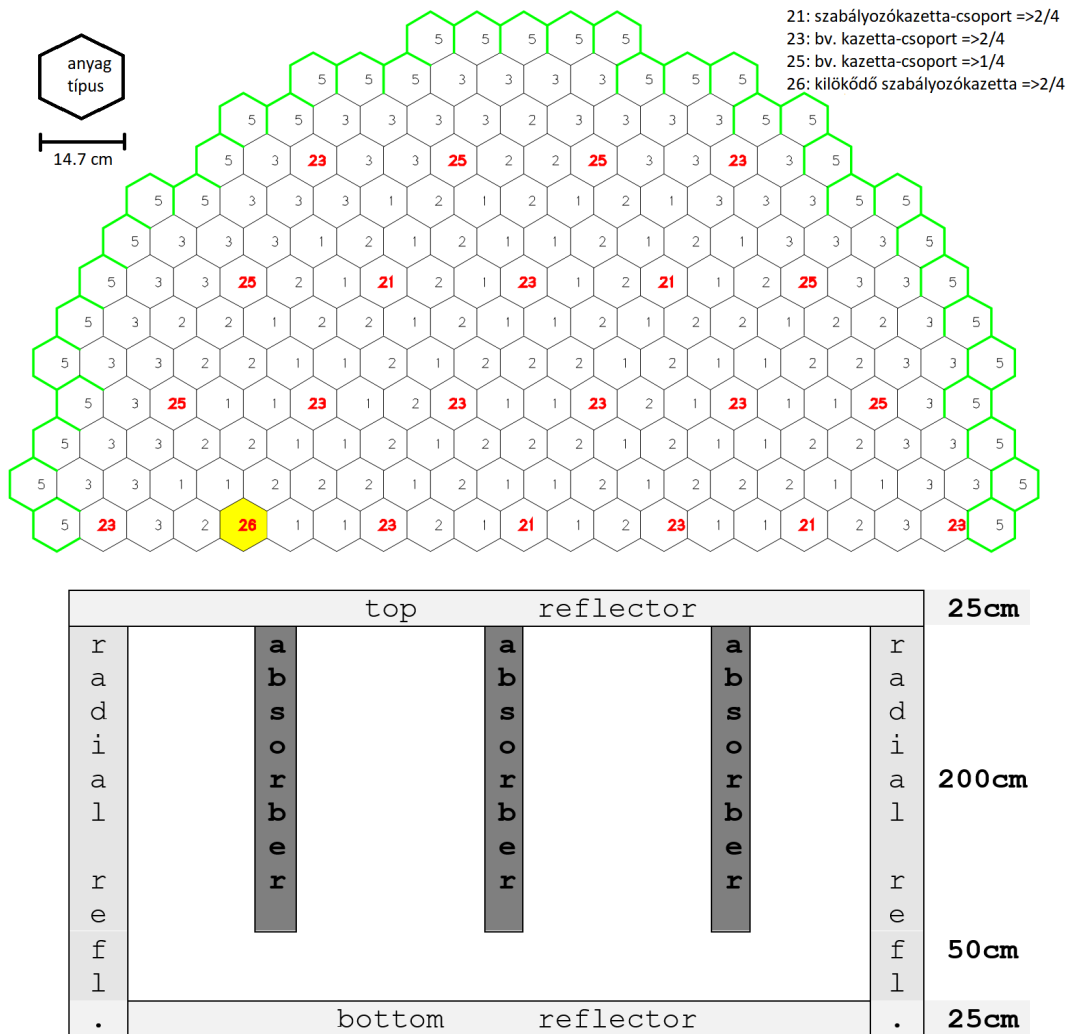
A tranziens során az egyik szélső szabályzókazetta fog kilöködni, ami megtöri a zóna kezdeti állapotban meglévő szimmetriáját. Ez pedig azt jelenti, hogy most nem lesz elegendő a zóna 1/12 részét modellezni. Ugyanakkor egy szimmetriasík így is megmarad, azaz a zóna felét célszerű modellezni.

A geometria felépítésekor figyelni kellett arra is, hogy a tranziens során a geometriában történő változások megfelelően lekövethetőek legyenek. A benchmark specifikációja szerint ez azt jelenti, hogy axiálisan 12 rétegre kell bontani a geometriát (10 réteg a zónának, 1-1 a reflektoroknak). A tesztet elsősorban nodális kódoknak tervezték, ugyanis a VVER üzemeltetők leggyakrabban ilyeneket használnak, illetve például a Paksi Atomerőműben hibrid-végeselem-módszeren alapulót, ami axiális megoldás szempontjából szintén a nodális kódokhoz hasonlatos. Ezeknek a kódtípusoknak elegendő ez az axiális felbontás, azonban tisztán végeselem-módszer esetén, mint amelyet az SPNDYN általam használt modulja is alkalmaz, hibás eredményekhez vezethet az, hogy túl nagy geometriai térfogatelemekben változnak a csoportállandók. Ezért én az SZBV (szabályozó és biztonságvédelmi) kazetták mozgásának jobb modellezhetősége érdekében a geometriát axiálisan 24 rétegre bontottam fel, melyből 20 réteg a zóna, 2-2 réteg pedig az alsó és felső reflektorok. Természetesen ezt nem csak a változásokat tartalmazó kazettákkal kell megcsinálni, hanem az összessel, különben a hálóban nem lenne biztosított az elemek csatlakozása. Így pedig a teljes geometria 5304 térfogathálóból tevődik össze.

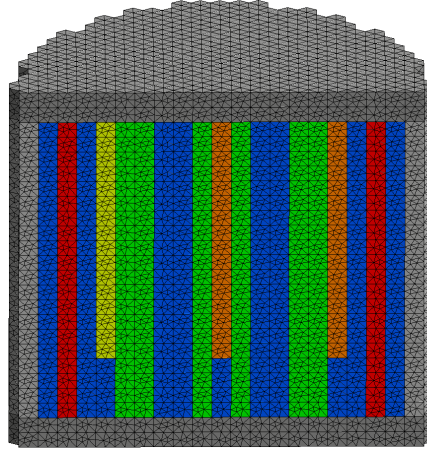
A geometria felépítésekor figyelni kellett arra is, hogy a szimuláció során a geometria és a háló fix lesz, csupán az egyes anyagok csoportállandói változhatnak. Ez pedig azt jelenti, hogy nem elég a specifikációban szereplő 6 anyagot definiálni, hanem minden olyan térfogat, amelyben változás történik, külön azonosítójú homogén régiót kell, hogy kapjon. Ezen persze valamelyest egyszerűsít, hogy például a biztonságvédelmi kazetták együtt mozognak, de így is kell 20-20 anyag a kilöködő szabályzókazettának, a szabá-

lyozókazettáknak és a kétféle biztonságvédelmi kazettának, így összesen 86-féle homogén régió lett a geometriában.

A számításokhoz két hálót generáltam, a továbbiakban h10-ként jelölt 63622 nódusból, a h6 pedig 203708 nódusból áll. Az előzőekben ismertetett benchmarkfeladat alapján egyértelmű, hogy ugyanilyen nódusszámokkal még az 1/12-es geometri esetén sem lenne hálófüggetlenséghez közeli a megoldás. Most ennyi nódus egy 1/2-es geometriában lesz elosztva, ami még nagyobb eltérésekhez vezethet. A későbbiekben ezt figyelembe véve kell majd értékelni az eredményeket.



10. ábra. A Dyn001 benchmark specifikációja.



11. ábra. A Gmsh-val létrehozott háló a Dyn001 benchmarkhoz.

### 3.2.2. Csoportállandók időbeli változása

A tranziens során először a 26-os számmal jelölt helyen lévő szabályozókazetta lökődik ki, 0 s és 0.08 s között konstans sebességgel. Ezután elkezdi felfutni a teljesítmény, melynek hatására 1 s-nál elkezdődik a vészleállítás. Ennek során a szabályozó és a biztonságvédelmi kazetták is ugyanazzal a konstans sebességgel mozognak, utóbbiak 10 s alatt érik el a zóna alját. A szimuláció célja a tranziens követése 0-6 s-ig. Ez idő alatt a szabályozókazetták el fogják érni a zóna alját, a biztonságvédelmi kazetták viszont csak a zóna feléig jutnak le. A csoportállandók időbeli paraméterezését ennek megfelelően kellett elvégezni.

Tegyük fel, hogy egy adott térfogatot a  $t_1$  időpillanatig teljes mértékben 1-es típusú anyag tölti ki, egy  $t_2 > t_1$  időpillanat után pedig teljes mértékben 2-es típusú. Ekkor a két időpillanat között a térfogat valahányad részét 1-es a többit pedig 2-es típusú anyag tölti ki, mely a végelem modellben nem megvalósítható állapot. Így a csoportállandó változást első közelítésben paraméterezhetjük úgy, hogy kiválasztunk egy  $t_1 \leq t' \leq t_2$  időpillanatot, melyig az 1-es anyagtípushoz tartozó csoportállandókat használjuk, utána pedig a 2-es anyagtípushoz tartozókat, azaz tetszőleges csoportállandóra:

$$\Sigma(t) = \Sigma^1 \cdot \Theta(t' - t) + \Sigma^2 \cdot \Theta(t - t') , \quad (12)$$

ahol  $\Theta(x)$  a lépcső (vagy egységugrás, "heavyside") függvény, melynek értéke 0, ha  $x < 0$  és 1, ha  $x \geq 0$ .

A kérdés itt az, hogy hogyan válasszuk meg  $t'$ -t. Az általam választott megoldás erre az lett, hogy legyen  $t' = \frac{t_1+t_2}{2}$ , azaz akkor történjen meg a csoportállandó-váltás, amikor az anyagok fele-fele arányban töltik ki a térfogatot. Ezzel egy kicsit nagyobb maximális teljesítményt fogunk kapni a valóságnál, hiszen később kezdenek becsúszni a kazetták, ugyanakkor az eltérés a biztonság szempontjából konzervatív, így elfogadhatónak tekinthető. Megjegyzés: látható, hogy ennél a paraméterezésnél a csoportállandó-változások egyszerűen lépcsőfüggvényekkel írhatók le, így ezt a későbbiekben hívjuk lépcső paraméterezésnek.

Meg kell még választani az időbeli felbontást is. Itt a mérlegelendő szempontok a tranziens lépéseinek megfelelő modellezése és a részletességgel növekvő számítási időigény. A kazettakilökődés során a kazetta 0.005 s alatt megy följebb egy rétegnnyivel 20 axiális szint feltételezése esetében, így én optimális időfelbontásnak a  $\Delta t = 0.005$  s-ot választottam.

### 3.2.3. Teljesítményszámítás és kezdeti eredmények

Elkészült tehát a geometria, illetve abból a hálók és a csoportállandók is megfelelően paraméterezve lettek, tehát a számítást le is lehetett futtatni. Az SPNDYN gond nélkül le is futott, az első megoldandó probléma az eredmények feldolgozásakor adódott.

Az SPNDYN az eredményeket ugyanis .msh fájlok formájában menti el, melyek az adott energiacsoport neutronfluxus-eloszlását (annak a nódusokban felvett értékét) tartalmazzák minden időpillanatban. Ezzel önmagában nincs is probléma, az eredmény így a Gmsh-val beolvasható, ábrázolható és feldolgozható. A nehézség az az, hogy nekünk nem a neutronfluxus-eloszlásokra van szükségünk, hanem a 3D teljesítmény-eloszlásra, azaz minden geometriai térfogatra ki kell integrálni a csoportfluxusokat és megszorozni a vonatkozó hasadási csoportállandóval, illetve a hasadásonként felszabaduló energiával. Megjegyzés: a benchmark feltételezi, hogy minden hasadás során ugyanakkora energia szabadul fel, így ezzel nem kell foglalkozni. Ilyen módon pedig egy relatív teljesítményt fogunk számolni a továbbiakban:

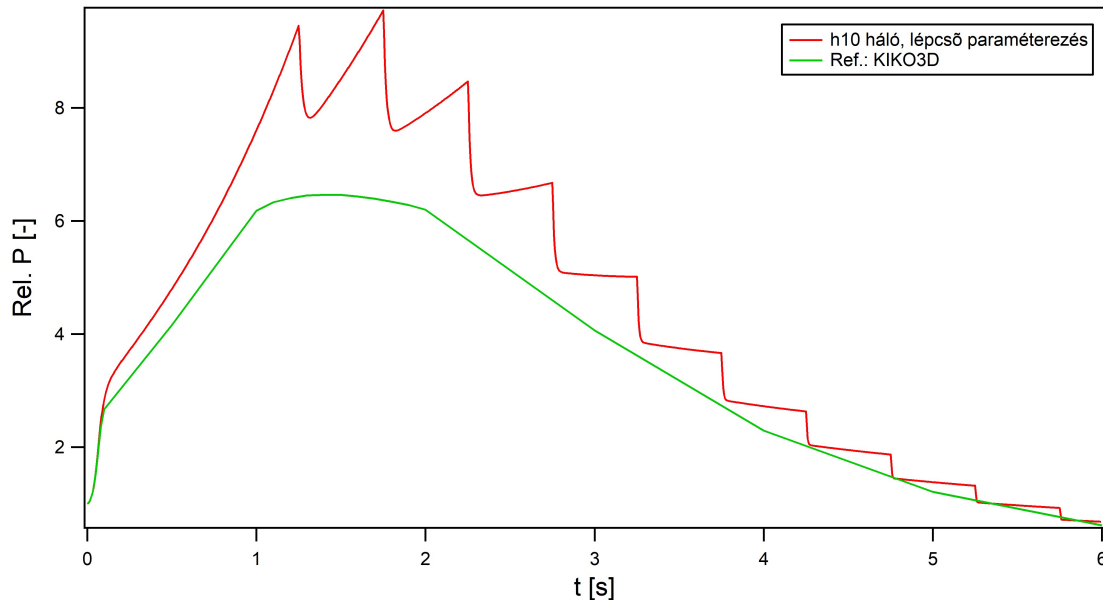
$$P_{rel}^i(t) = \sum_{g=1}^G \int_{V^i} \kappa \Sigma_g^i(t) \Phi_g(t) dV . \quad (13)$$

Ez képlet szintjén nem tűnik túl bonyolultnak, azonban hatékony numerikus megvalósítása nem olyan egyszerű, hiszen például a h6 háló nagyságrendileg 1.1 millió tetraéderből áll, melyekre integrálni kell. Az SPNDYN-nel végrehajtott egyik korábbi időfüggő számításához készült egy segédprogram, mely elvégzi az integrálásokat, és kis módosításokkal ezt a Dyn001 benchmark eredményfájljainak feldolgozására is sikerült alkalmassá tenni. A probléma vele az volt, hogy nem volt megfelelően optimalizálva, így a korábbi, lényegesen egyszerűbb (kisebb elemszámú) probléma esetén megfelelt, de most a futásideje a h10 háló esetén is várhatóan 70 óra körül lett volna. Ez nyilvánvalóan nem elfogadható annak tükrében, hogy az SPNDYN futásideje alig több mint 1 óra volt ugyanezzel a hálóval. Egyik megoldási lehetőség a meglévő program optimalizálása lenne, ami az APROS termohidraulikai rendszerkóddal való csatolás kapcsán már folyamatban is van. Pont ezért én nem szerettem volna ennek párhuzamosan nekiállni, helyette inkább készítettem egy saját teljesítményszámító programot. Ehhez azt használtam fel, hogy a Gmsh vezérelhető többek közt Python API-n keresztül is, a Gmsh pedig rendelkezik megfelelően optimalizált numerikus integrálóval. Azaz az általam írt programnak csak meg kell nyitnia az eredményfájlokat a Gmsh-val, ki kell választania a megfelelő térfogatot, le kell futtatnia a Gmsh integrálóját majd az eredményeket be kell szoroznia a hasadási hatáskeresztmetszetekkel és össze kell adnia az energiacsoportok járulékait. Mindezt egy körülbelül 100 soros program és a hozzá tartozó segédfájlok ( $\Sigma_g^f(t)$ , térfogatok Gmsh azonosítói, térfogatok anyagtípusa) formájában sikerült is megvalósítani. Ezzel pedig a teljesítmény számítása csak 45 percet vett igénybe, ami már egészen jó.

Ezzel el is készültek a referenciákkal összevethető eredmények. A benchmarkhoz nincs ajánlott referenciamegoldás, viszont többféle kód megoldása is elérhető. Ezek közül a KIKO3D és a DYN3D eredményei a [13] honlapról numerikus formában is elérhetők, így az összevetéshez elsősorban ezek használhatóak.

Elsőként nézzük a zónateljesítmény időbeli alakulását. Itt a kezdeti teljesítménnyel le lettek normálva az eredmények, így egy 1-ről induló relatív teljesítményt kaptunk az idő függvényében. A h10 hálóval,  $\Delta t = 0.005$  s-os időlépéssel és lépcső csoportállandó

paraméterezéssel kapott eredmény, illetve az egyik referenciamegoldás a 12. ábrán látható. Az már első ránézésre is megállapítható, hogy valami nincs rendben az eredménnyel. Amit az ábrán látunk, az a "rod cussing" jelensége. Hogy mi ez, és mit lehet ellene tenni, azt a következő részben fogom ismertetni.



12. ábra. A relatív teljesítmény időbeli változása a legritkább hálóval és lépcsős csoport-állandó paraméterezéssel.

### 3.2.4. A "rod cussing" jelensége és lehetséges kezelése

A "rod cussing" (magyarra talán rúdmozgás okozta csúcsosodásként fordítható le) az SPNDYN kapcsán most jelent meg először, azonban egy tranziens reaktorfizikai modellezésnél igen gyakran megjelenő jelenségről van szó. Oka az, hogy a szabályozókazetta neutronértékesége jelentősen eltér az üzemanyag kazettákétól, így ha a szabályozókazettákat túl nagy lépésekben, ugrásszerűen hozzuk be, akkor a reaktivásban és ezzel a neutronfluxusban is megjelenik egy nem fizikai, ugrásszerű válasz. Ez a jelenség nem reaktortípus specifikus és a főbb reaktorfizikai kódtípusok (nodális, végeselem, stb.) mindegyikénél előfordul, így már számos megoldási lehetőséget is kidolgoztak a jelenség hatásának csökkentésére. Persze az, hogy az egyes megoldások mennyire hatékonyak, és egyáltalán alkalmazhatóak-e az adott programban, az erősen függ a kód típusától. Ilyen szempontból a végeselem-módszeren alapuló kódok, mint az SPNDYN általam használt modulja eléggé rossz helyzetben vannak, ezeknél csak az egyszerűbb, de kevésbé hatékony megoldások alkalmazhatóak.

A legegyszerűbb megoldás, melyet én is alkalmaztam, az a térfogatokkal való súlyozás módszere. Ez annyit jelent, hogy a csoportállandókat nem lépcsőfüggvény szerint váltjuk át egyikről a másikra, hanem mikor egy geometriai térfogatban kétféle anyag van jelen, akkor azok csoportállandóit az általuk betöltött térfogatokkal súlyozva átlagoljuk. Figyelni kell viszont arra, hogy a diffúziós csoportállandó reciprokosan adódik össze, ennek megfelelően az átlagolást is reciprokosan kell végezni. Emellett a kazetták konstans se-

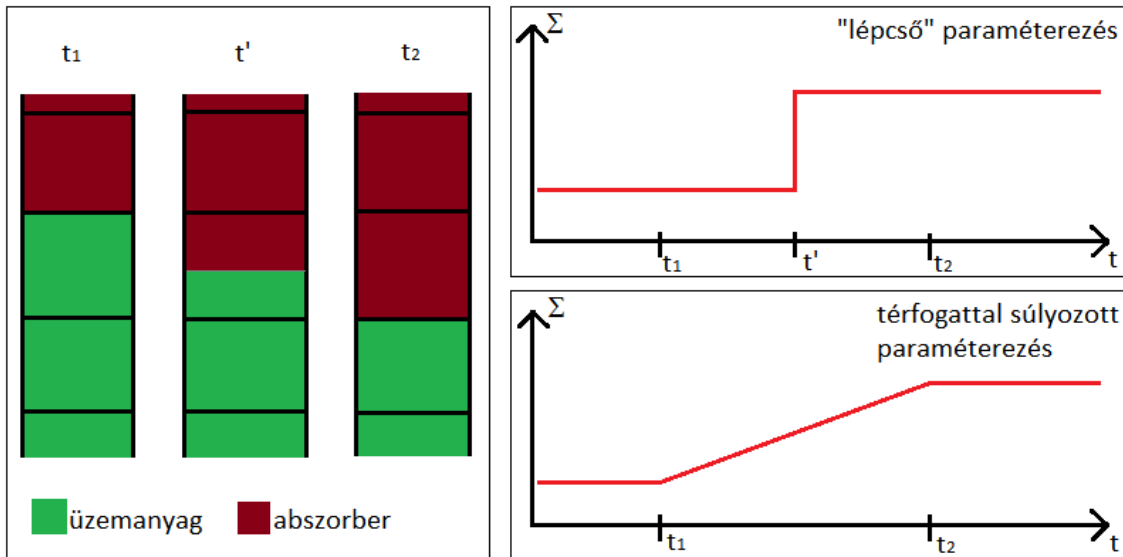
bességgel mozognak, így a térfogat szerinti súlyozást egyszerűen kifejezhetjük az idővel. Azaz a korábbi példa jelöléseit használva:

$$\Sigma(t) = \Sigma^1 \cdot \Theta(t_1 - t) + \left[ \frac{t_2 - t}{t_2 - t_1} \Sigma^1 + \frac{t - t_1}{t_2 - t_1} \Sigma^2 \right] \cdot \Theta(t - t_1) \cdot \Theta(t_2 - t) + \Sigma^2 \cdot \Theta(t - t_2) \quad (14)$$

és

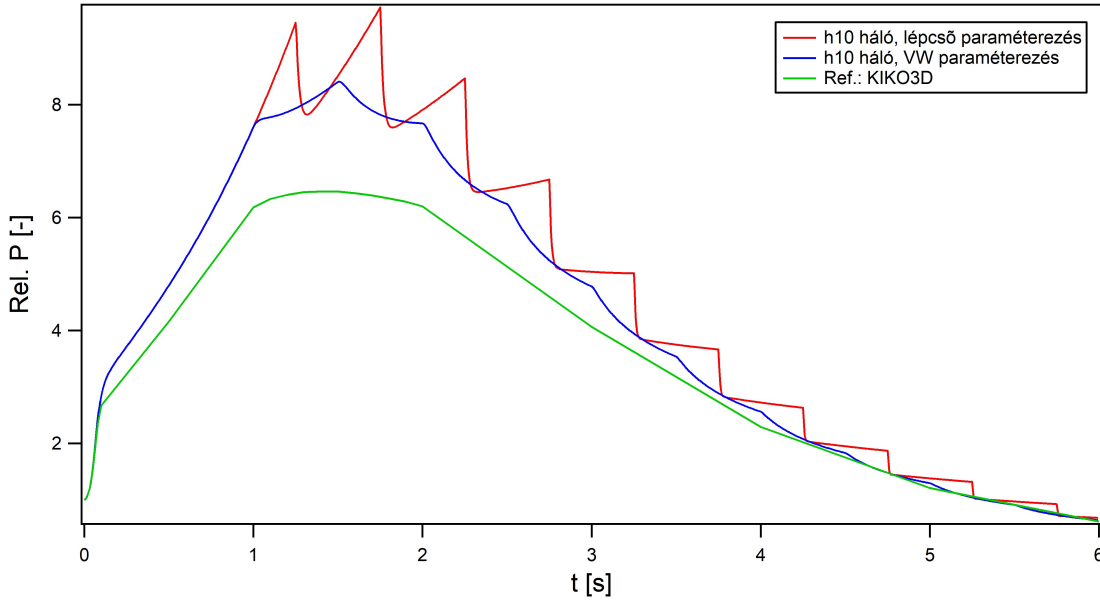
$$D(t) = D^1 \cdot \Theta(t_1 - t) + \frac{1}{\frac{t_2 - t}{t_2 - t_1} \frac{1}{D^1} + \frac{t - t_1}{t_2 - t_1} \frac{1}{D^2}} \cdot \Theta(t - t_1) \cdot \Theta(t_2 - t) + D^2 \cdot \Theta(t - t_2) \quad (15)$$

Hogy ez mit is jelent, azt a 13. ábrán próbáltam szemléltetni, ahol azt látjuk, ahogy a szabályozókazetta megy lefelé egy geometriai térfogatban és a pozíciójának, vagyis az időnek a függvényében hogyan változik az egyik csoportállandó a különböző paraméterezésekkel.



13. ábra. Szemléltető ábra a csoportállandók különböző paraméterezéseihez.

A térfogatokkal súlyozás módszerének egyértelmű előnye, hogy a csoportállandók paraméterezésével, az input adatok változtatásával megvalósítható, így semmilyen módosítást nem igényel a kódban. Emellett a szabályozókazetta így "valós időben" kerül a zónába, azaz a maximális teljesítményt így nem fogjuk felülbecsülni, mint a lépcső paraméterezéssel. Ami viszont a hátránya, az az, hogy az eredmény továbbra sem lesz kellően sima, mint azt a 14. ábrán is látjuk, a teljesítmény időbeli változásánál továbbra is maradnak fizikai tartalommal nem rendelkező csúcsok. Ennek magyarázata az, hogy bár kisebb lépésekben, de így is egy jóval nagyobb térfogatban változnak a csoportállandók mint kellene. Továbbá a gondot az abszorber kazetta neutronértékessége okozza, azaz nem csak az számít hogy mekkora térfogatot foglal el, hanem az is, hogy ott mekkora a neutronfluxus. Ez azért is probléma, mert egy abszorber környezetében általában erősen változó a neutronfluxus, azaz a térfogatokkal való súlyozás valójában eléggé rosszul fogja közelíteni a kevert csoportállandókat.



14. ábra. A relatív teljesítmény időbeli változása a legritkább hálóval és térfogattal súlyozott csoportállandó paraméterezéssel. (VW=Volume Weighted=térfogattal súlyozott)

Erre nyújt megoldást az egyel bonyolultabb lehetőség, a neutronfluxussal való súlyozás módszere [14]. Alapötlete nagyon egyszerű, súlyfaktorként használjuk a térfogat helyett a neutronfluxust. Persze ekkor az átlagolás összeadás helyett integrálás lesz, így például egy  $z$  irányban 1 magasságú geometriai térfogat esetén, melyet 0-tól  $h$ -ig 1-es anyag,  $h$ -tól 1-ig pedig 2-es anyag tölt ki:

$$\Sigma_g(t) = \frac{\int_0^{h(t)} \Phi_g(t,z) \Sigma^1 dz + \int_{h(t)}^1 \Phi_g(t,z) \Sigma^2 dz}{\int_0^1 \Phi_g(t,z) dz}, \quad (16)$$

Ezzel a módszerrel a térfogattal való súlyozáshoz képest már egészen jó eredményeket lehet kapni, ahogyan azt az [14] forrásban elvégzett tesztszámítások is mutatják. Persze teljesen jó megoldást ezzel sem kapunk, hiszen az a probléma továbbra is fennáll, hogy lényegesen nagyobb térfogatban változtatjuk a csoportállandókat mint kéne. Ami az SPNDYN szempontjából lényeges ezzel a módszerrel kapcsolatban az az, hogy ez elméletileg végeelem-módszer esetén is használható. Ugyanakkor implementálásához már nem lenne elég a csoportállandókat átparaméterezni, magát a számítási módszert kellene módosítani. Emellett, mint azt korábban már láttuk, a numerikus integrálás elég lassú tud lenni, ráadásul most azt is számon kéne tartani, hogy helyileg hol tart a kazetta az idő függvényében, és így mely elemekre kell integrálni. Utóbbi kérdést a strukturálatlan háló miatt gyakorlatilag nem is lehetne megválaszolni, így először is strukturált (hexaéderez vagy prizma) hálóra kellene váltani. Összességében tehát a neutronfluxussal való súlyozás módszere beépíthető lenne az SPNDYN-be is, a megvalósításának nehézsége viszont lényegesen túlmutat egy TDK dolgozaton.

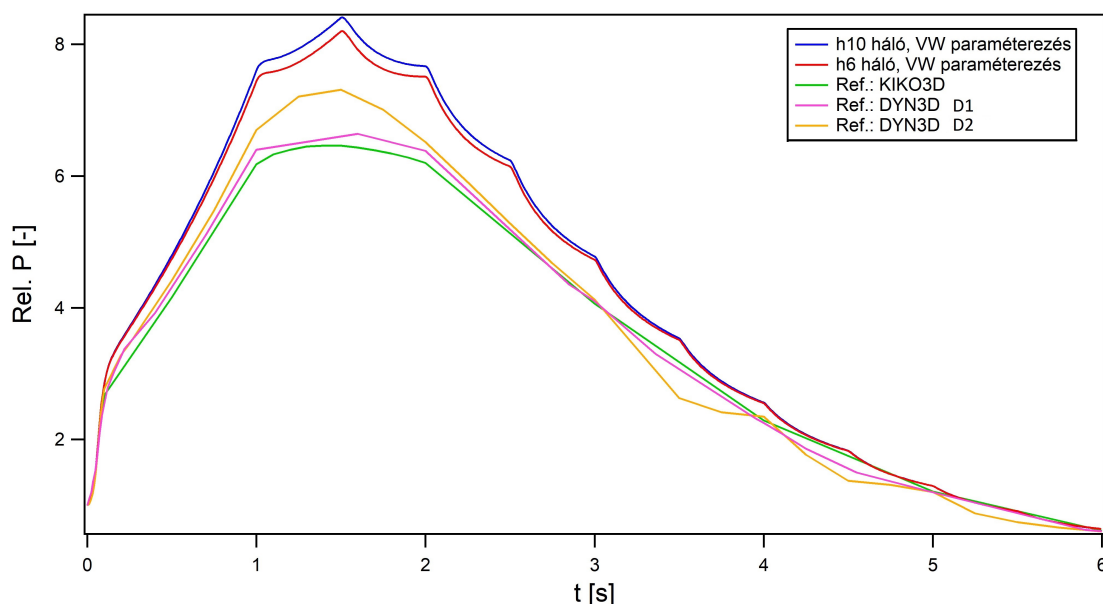
Léteznek olyan megoldások is, mint például a mozgó hálók módszere [15], melyek végeelem-módszer esetén nem alkalmazhatók, viszont nagyon nagy mértékben csökkentik a "rod cusping" effektust. Ezek általában azon alapulnak, hogy a nodális kódoknál

az abszorber kazettákat peremfeltételként, albedóval veszik figyelembe, az pedig mozgó háló segítségével megoldható, hogy a peremfeltételt a nódus egy közbenső vízszintes síkján vegyük figyelembe, keverést így csak a nódusok radiális csatlakozásainál lévő albedó peremfeltételeiben kell alkalmazni.

### 3.2.5. Végeredmények

Készen van tehát a geometria és megtörtént hálózás, a csoportállandók paraméterezését pedig a lehetőségekhez mérten optimálisan, térfogat szerinti súlyozással korrigáltuk, így nincs más hátra, mint az eredmények kiértékelése. Itt először is meg kell jegyezni, hogy a térfogat szerinti súlyozással a számítás során 1 s és 6 s között minden időpillanatban újra vannak számolva a globális mátrixok, ami a számítási idő drasztikus növekedéséhez vezet. Ezért ezek a számítások egy egyetemi serveren lettek lefuttatva, lényegesen nagyobb számítási kapacitás mellett. Ennek ellenére a h10 hálóval és  $\Delta t = 0.005$  s időlépéssel indított futtatás számítási időigénye 17 óra volt. Ennek fényében nyilvánvaló, hogy itt hálófüggetlenségi vizsgálatot nem lehetett végezni, a kiértékelést így a ritka háló által okozott hiba tudatában kell végrehajtani.

Bár komplett hálófüggetlenségi vizsgálatot nem csinálhattam, azt azért jó lenne tudni, hogy a probléma mennyire érzékeny a hálósűrűsége, és az eredmények milyen mértékben változnak a háló sűrítésével. Ennek eldöntésére a h6 hálóval és szintén  $\Delta t = 0.005$  s időlépéssel is le lett futtatva a benchmark. Ez a számítás már két és fél napot vett igénybe. A kapott eredmények és a numerikus formában elérhető referenciamegoldások a 15. ábrán láthatóak.



15. ábra. A relatív teljesítmény időbeli változása a h10 és h6 hálókval, térfogattal súlyozott csoportállandó paraméterezéssel.

Ahogy arra számítottunk, a sűrűbb hálóval egy kicsit más eredményt kaptunk, azonban a különbség nem jelentős. A teljesítmény felfutásának és lecsengésének szakasza



gyakorlatilag megegyezik, ahol különbség van, az a maximális teljesítmény. Ez pedig a referenciák irányába tolódott el, tehát feltételezhetjük, a háló sűrítésével még jobb eredményeket kapnánk. De mivel erre nincs lehetőség, a h6-os hálóval kapott eredményt el is fogadhatjuk végeredménynek.

Hasonlóan vizsgálni érdemes még az eredmény időlépéstől való függését is. Ehhez a h10 hálóval, de  $\Delta t = 0.01$  s időlépéssel is végeztem egy számítást. Az így kapott eredmények eltérése nagyon kicsi volt ( $<0.5\%$  a közös időpontoknál), ami alapján azt feltételezem, hogy a  $\Delta t = 0.005$  s időlépéssel kapott megoldás az időlépéstől függetlennek tekinthető, azt az alkalmazott fizikai, matematikai és numerikus modell vizsgálandó hibái mellett csak a nem kellően sűrű hálóból eredő hiba terheli. Megjegyzés: az időlépés csökkentésének vizsgálata is tervbe volt véve egy  $\Delta t = 0.0025$  s időlépéssel végzett szimuláció formájában, de a szervergéppel kapcsolatos váratlan problémák miatt erre végül nem volt lehetőség. Amennyiben addigra sikerül lefuttatni, a TDK előadásom alkalmával ki fogok térni arra, hogy az eredmény megfelelt-e a várakozásnak.

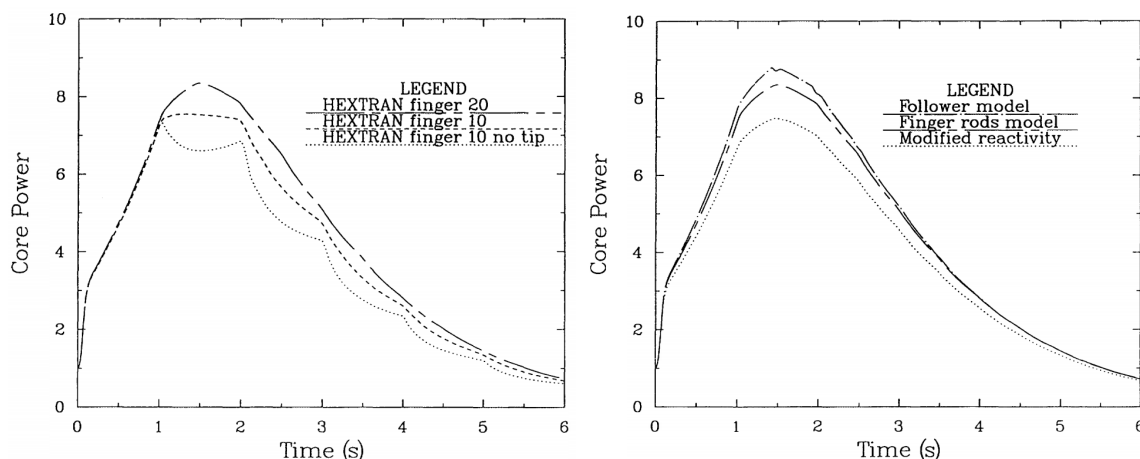
Ami a 15. ábrát megnézve egyértelműnek tűnik az az, hogy a referenciáknál nagyobb maximális teljesítményt kaptunk. Azonban mielőtt ezt hibának feltételeznék, meg kell jegyezni, hogy a Dyn001 benchmark a 15. ábrán szereplő kettőnél már jóval több kóddal végre lett hajtva, de csak ehhez a kettőhöz érhetőek el numerikus formában az eredmények. Ajánlott referenciamegoldás pedig nincs, mivel a különböző kódok megoldásai közt viszonylag nagy eltérések vannak és nem lehet kijelenteni azt, hogy melyik a jó. Nézzük csak meg a DYN3D-val készült két megoldás eltérését (lásd a 15. ábra). A D1-nél egy magasabb rendű homogenizációs eljárást alkalmaztak a "rod cusping" elkerülésére, viszont csak 10 rétegre bontották a zónát. Ezzel szemben a D2-nél több rétegre bontották a zónát, viszont egy egyszerűbb homogenizációs eljárást alkalmaztak, az eredmény pedig eléggé eltérő lett, ami azt mutatja, hogy a maximális teljesítmény igen érzékeny a számítások részleteire. Illetve azt is megfigyelhetjük, hogy a zóna 20 rétegre bontásával, ahogyan azt én is csináltam, a maximális teljesítmény felfelé tolódott el. Ennek az okának a kiderítésében jó támpont lehet a [16]-ös forrás, melyben szintén megoldásra kerül a Dyn001 benchmark és felmerül a maximális teljesítmény eltéréseinek kérdése. Ebben ezt azzal indokolják, hogy a kilökődött kazetta reaktivitásértékessége közel van a prompt kritikussághoz, így egy kis eltérés a dinamikus reaktivitásértékességben nagy változást okozhat a maximális teljesítményben.

Ahhoz tehát, hogy következtetéseket lehessen levonni az SPNDYN-nel készült megoldás megfelelőségével kapcsolatban, végig kell nézni a többi referenciamegoldást is, illetve meg kell vizsgálni, hogy milyen tényezőktől függ még a megoldás. Ehhez nézzük meg a HEXTRAN kóddal készült megoldás dokumentációját [17], melyben elég sok paraméter vizsgálatra került.

Ezek közül az első, ami az SPNDYN-t is érinti, az a későneutron-anyamagok mozgásának kezelése. Ugyanis a VVER-440 reaktor követőkazettás szabályzó és biztonságvédelmi kazettákkal rendelkezik, ami azt jelenti, hogy a szabályzókazetták nem belecúsznak az üzemanyag kazettába, mint a VVER-1000 és VVER-1200 típusoknál, hanem alul kitolják azt a zónából. Így a kazettakilökődés és a vészleállítás során valójában az üzemanyag a benne lévő későneutron-anyamagokkal együtt mozog, ezt pedig a specifikáció szerint modellezni is kellene. Az SPNDYN-nel erre viszont nincsen lehetőség, ami nyilvánvalóan az eltérések egy lehetséges forrása a többi kódhoz képest. Szerencsére ennek hatását a HEXTRAN kóddal már megvizsgálták, azaz készítették egy követőkazettás (16./jobb ol-

dali ábra "Follower model") és egy követőkazetta nélküli modellt is (16./jobb oldali ábra "Finger rods model"). Ezek alapján megállapítható, hogy a maximális teljesítmény ugyan eltér egy kicsit, de szignifikáns eltérés nincsen a két modell között, így ez feltételezhetően az SPNDYN esetén sem okoz túl nagy eltérést az eredményekben.

A második dolog amit vizsgáltak, az a kazettaértékesség változásának hatása. Ehhez a csoportállandók változtatásával egy nagyon kicsit csökkentették a kilökődő kazetta reaktivitásértékességét, melynek hatására, mint az a 16./jobb oldali ábrán látható ("Modified reactivity"), az elért maximális teljesítmény jelentősen lecsökkent, ami a [16]-ben tárgyalt elméletet erősíti.



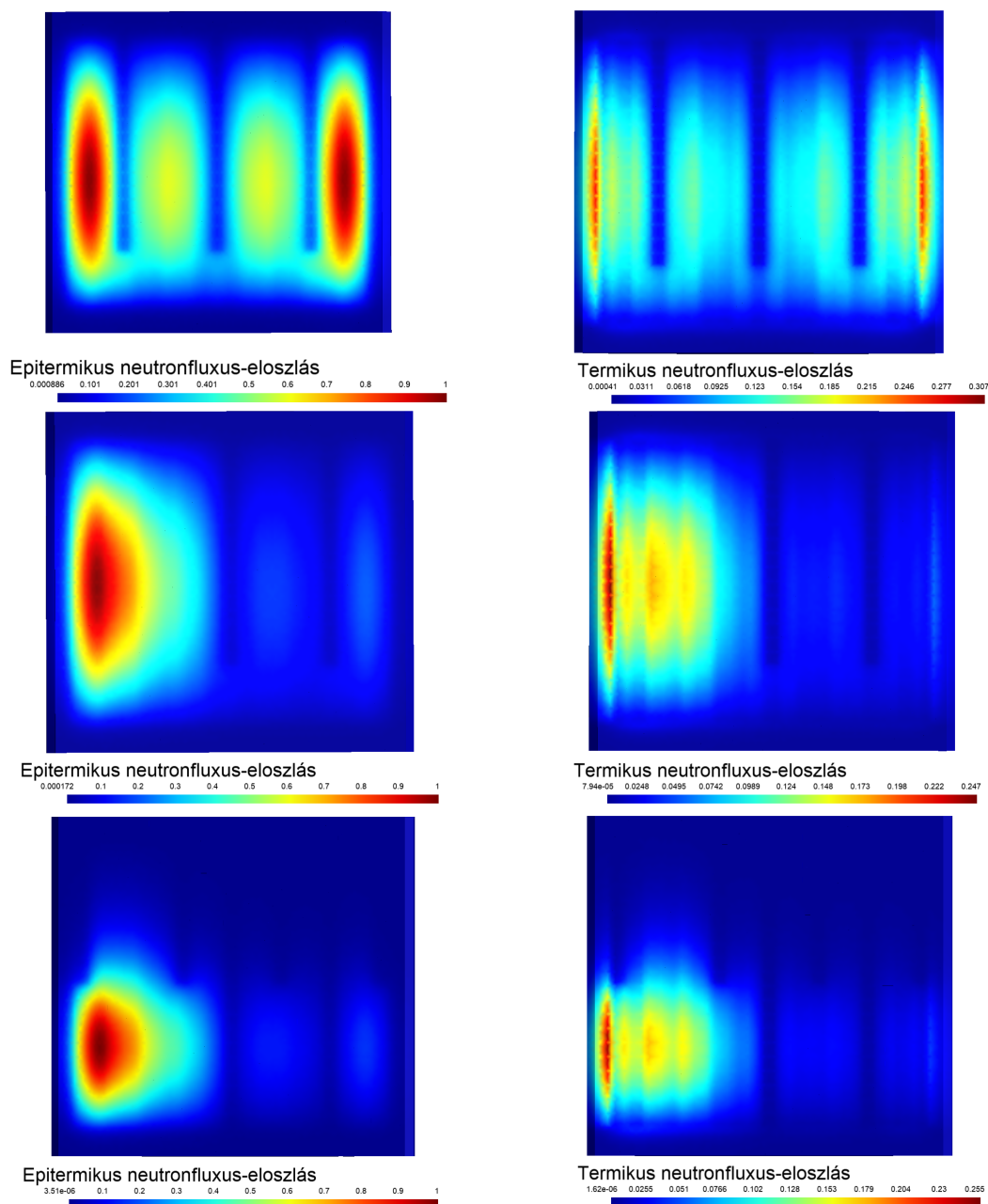
16. ábra. A HEXTRAN kóddal végzett számítások eredményei. A [17] forrásból átvett ábrák.

A harmadik dolog amit vizsgálnak, az a geometria axiális felbontása. Ehhez egy axiálisan 20 rétegre (16./bal oldali ábra "HEXTRAN finger 20") és egy 10 rétegre (16./bal oldali ábra "HEXTRAN finger 10") osztott geometriával kapott eredményeket hasonlították össze. Látható, hogy az eltérés elég jelentős, csupán 10 réteg esetén a görbe ellaposodik a maximumnál, a leállításnál pedig a kifinomultabb homogenizáció ellenére is megjelenik a "rod cusping". A 10 réteges modellnél kipróbálták a sima térfogattal súlyozott homogenizációt is, mellyel az ideálistól még az általam kapottnál is jobban eltérő görbe lett az eredménye. Ez tehát azt mutatja, hogy ha a térfogattal való súlyozásnál jobb homogenizációs eljárásra nincs lehetőség, akkor a geometria axiálisan több rétegre való felbontása mindenképpen indokolt.

Végezetül pedig a 16. ábrát megnézve még egy nagyon fontos következtetést le kell vonnunk. A HEXTRAN 20 réteges geometriával és követőkazettát figyelembe nem vevő modellel kapott eredménye gyakorlatilag megegyezik az SPNDYN-nel általam kapottal, az SPNDYN-nél lévő jelentősebb "rod cusping"-től eltekintve. Ez pedig azt jelenti, hogy az SPNDYN által szolgáltatott eredmény a teljesítmény időbeli alakulásának tekintetében elfogadható.

Természetesen nem a teljesítmény időbeli alakulása az egyetlen szempont, ami szerint a kódokat össze lehet hasonlítani. A benchmark specifikációja szerint a következő szempont a reaktivitás időbeli alakulása lenne a következő összehasonlítandó eredmény, de ez nem lehetséges, ugyanis az SPNDYN ezt nem számolja. Így az egyetlen ehhez hasonló

mennyiség, amit az SPNDYN-nel is számolni lehet, az a kilökődő kazetta reaktivitásértéksége stacionárius számítások alapján. Összesen háromféle stacionárius számítást végeztem, egy kezdeti állapotnak megfelelőt, egyet a kilökődött kazettával, egyet pedig a 6 s-nál lévő végállapotnak megfelelően. Az ekkor kapott normált neutronfluxus-eloszlásokat szemlélteti a 17. ábra. Ezek az ábrák, főleg a termikus neutronfluxus-eloszlások, nagyon jól szemléltetik azt is, hogy milyen nagy elemekből áll ez a háló, és ezekkel milyen pontossággal (vagyis inkább mekkora hibával) tudjuk közelíteni a fluxust.



17. ábra. A statikus futtatások neutronfluxus-eloszlásai. Fent: kezdeti állapot, középen: kilökődött kazetta, lent: végállapot. (A h6 háló eredményei, axiális keresztmetszetek a zóna közepén.)

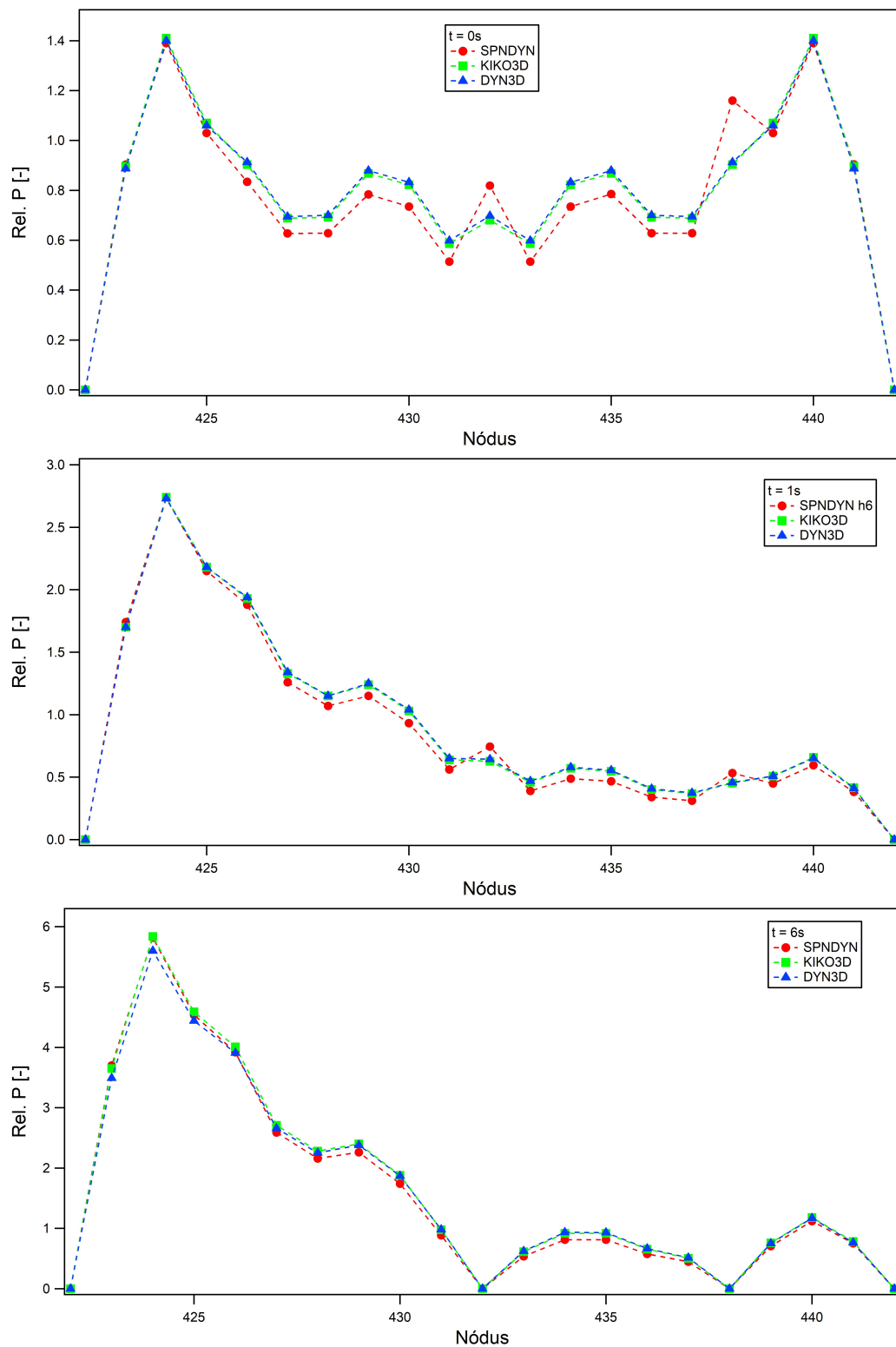
A h6 hálóval végzett stacionárius számítások esetén a kezdeti állapotban  $k_{\text{eff,kezd}} = 1.00248$ , a kilöködött kazettával  $k_{\text{eff,kint}} = 1.00717$ , a végállapotban pedig  $k_{\text{eff,vég}} = 0.99299$  adódott. Ezek alapján a kilöködött kazetta reaktivitásértékessége

$$\frac{k_{\text{eff,kint}} - k_{\text{eff,kezd}}}{k_{\text{eff,kezd}}} = 0.00467, \quad (17)$$

ami összehasonlítva a specifikációban kitűzött 0.00482 célértékkel egész jó egyezést mutat.

A következő összehasonlítási szempont a dinamikus 3D teljesítményeloszlások a szimuláció kezdetétől számított 0 s, 0.04 s, 0.08 s 1 s és 6 s időpontokban. Ezekhez összehasonlításként csak a numerikus eredményekkel is rendelkező KIKO3D és DYN3D jöhetett szóba. Ezek ilyen jellegű eredményei jó egyezést mutattak a specifikáció szerint, így a számszerű összehasonlítást én csak az egyikkel végeztem el, méghozzá a KIKO3D-vel. (Az ábrákon azért a DYN3D eredményei is szerepelni fognak.) A 3D teljesítményeloszlások összehasonlításához ismerni kell a geometriai térfogatok számozását is. Ez úgy néz ki, hogy a számozás a 10. ábrán a radiális metszeten balról-jobbra és fentről-lefelé, míg axiálisan letről felfelé halad. Fontos, hogy a referenciák 10 axiális rétegre vonatkoznak és az egyes időpontokban normálva vannak, azaz nekem a rétegek teljesítményét kettesével össze kellett adni normálás előtt. A 3D teljesítményeloszlásban az axiális reflektorok nincsenek benne, a radiálisak viszont 0.0 teljesítménnyel igen, így összesen 2210 térfogatra kell azt meghatározni.

A 3D teljesítményeloszlások összehasonlításához a specifikációban a referenciamegoldások egy metszete szerepel szemléltetésként, így én is ahhoz hasonló ábrákat készítettem, immár az SPNDYN eredményeivel kiegészítve. A 18. ábrán a harmadik rétegben (második a zónában) a középvonal (szimmetriatengely) mentén található térfogatok teljesítménye látható. Ezen a legfelső grafikon a  $t = 0$  s időpillanatbeli eloszlást mutatja, mely a számítás elején végzett stacionárius számítás eredménye. Azt a korábbiakban beláttuk, hogy a stacionárius diffúziós végeelem modul VVER-440 reaktor esetén jól működik, azaz a most kapott nagy eltérések a kezdeti állapotban a nem megfelelő hálósűrűségnek tudhatóak be. Ha megnézzük a későbbi időpontokban ugyanezt a metszetet, akkor látható, hogy a kezdetben fennálló különbségek az idővel nem változnak, az időbeli változásokat az SPNDYN megoldása is ugyanúgy követi le, azaz folyamatosan a háló okozta hibát látjuk. Persze ez a metszet nem teljesen reprezentatív, így a numerikus értékek mind a 2210 térfogatra és 5 időpontra vonatkozóan összehasonlításra kerültek (lásd B függelék). Ez alapján elmondható, hogy az eltérésekben a kezdeti állapotban fellelhető mintázat (alapvetően kis negatív eltérések és egy-egy kiugróan nagy pozitív az abszorberek környezetében) végig megmarad, illetve 6 s-nál az újonnan megjelent abszorberek környezetében szintén pozitív irányba tolódik az eltérés. Az összes térfogatot és időpontot figyelembe véve az abszolút értékben számolt átlagos eltérés 6.1 %, a maximális eltérés pedig 54.9 %. Mindez csak a kezdeti állapotot figyelembe véve 5.8 % és 48.2 %, ami azt jelenti, hogy az időfüggő modul hatására az átlagos eltérés csak 0.3 %-al emelkedik, a maximális eltérés pedig kevesebb mint 7%-al. Ezek alapján az a legvalószínűbb, hogy ezt valóban a túl ritka háló okozza és az SPNDYN időfüggő diffúziós végeelem modulja jól működik.



18. ábra. 3D teljesítményeloszlás különböző időpontokban alulról a harmadik rétegben a középvonal mentén. A kilökődő szabályozókazettát itt a 426-os számú nódus reprezentálja.

## 4. Összegzés és kitekintés

A dolgozat első részében bemutattam, hogy az SPNDYN jelenleg stacionárius esetben diffúziós és  $SP_3$  transzportközelítést képes használni, végeselem vagy hibrid-végeselem módszerrel, időfüggő esetben pedig a diffúziós és  $SP_3$  közelítéshez csak a végeselem-módszer az opció, illetve röviden az ezen módszerek elméleti hátterét is áttekintettem. Megnézem azt is, hogy az időfüggő modul teszteléséhez milyen VVER reaktorokra vonatkozó benchmarkfeladatok léteznek és ezek közül melyik oldható meg az SPNDYN jelenlegi képességeivel, illetve melyek a termohidraulikai kóddal való csatolást követően.

Ezután megoldottam a Seidel benchmarkot, mely egy VVER-440 zónát modellez. Ez ugyan egy stacionárius benchmark, ami még a szakdolgozatom keretében került megoldásra, ismertetése viszont elengedhetetlen volt az időfüggő változat során leírtak megértéséhez. Ennél ugyanis beláttam, hogy az SPNDYN VVER-440 reaktorok és stacionárius számítás esetén elfogadható eredményt ad, így a későbbiekben az időfüggő változatnál a kezdeti állapotban meglévő eltérés a túl ritka háló miatt van. Emellett ennél a hálófüggétlenségi vizsgálat során láttuk azt is, hogy még 1/12-es geometriában és stacionárius esetben is milyen nagy számítási kapacitásigénye van egy VVER-440 zóna modellezésének, így az időfüggő esetben a hálófüggétlenség vizsgálata már fel sem merülhet.

Az időfüggő változatnál aztán felépítettem egy jóval komplexebb geometriát, melyben már a csoportállandók paraméterezésével a kazettamozgásokat is le lehetett követni. Ehhez elkészítettem a csoportállandók paraméterezését is, melyet aztán a "rod cusping" jelenségével való szembesülést követően, a megoldási lehetőségeket mérlegelve korrigáltunk. Ezt követően az eredményeket a ritka háló okozta hiba tudatában összevettem a referenciaként elérhető megoldásokkal, mely során a tapasztalt eltérések okait is megpróbáltam megkeresni. Végül pedig levontam a következtetést, miszerint az SPNDYN időfüggő diffúziós végeselem modulja VVER-440 reaktortípusra elfogadható megoldást ad.

Persze ha majd valós elemzéseket is szeretnénk készíteni az SPNDYN-nel, akkor a most felmerült problémákra jobb megoldásokat kell keresni. A teljesítmény számítás optimalizálása, mint arra korábban is utaltam, folyamatban van, viszont a "rod cusping" jelenségét jelenleg ennél jobban nem lehet kezelni. Ennek megoldásában a magasabb rendű hálóelemek kezelése lehetne az első lépés. Ez amellet, hogy megteremtene a lehetőséget a fluxussal súlyozás módszerének alkalmazására, még a számítási időigényen is csökkenthetne ugyanolyan pontosság mellett.

Emellett azt sem szabad elfelejteni, hogy az SPNDYN rendelkezik egy hibrid-végeselem modullal is. Ez egy kevésbé elterjedt, de sokkal nagyobb potenciállal rendelkező eljárás, mint a sima végeselem-módszer, mind a számítási időigény csökkentése terén, mind pedig a "rod cusping" elkerülésére alkalmazható lehetőségek terén. Így érdemes azon is elgondolkozni, hogy a hibrid-végeselem modult is kiegészítsük-e a jövőben egy időfüggő megoldóval, illetve hogy magát a hibrid-végeselem módszert lehetne-e általánosítani más reaktortípusokra is. Jelenleg ugyanis a hibrid-végeselem módszer csak hatszöges kazetákból álló reaktorkonfigurációt képes kezelni.

## 5. Irodalomjegyzék

- [1] Babcsány Boglárka: Development of a finite-element-based neutron physics code system for the solution of the simplified  $P_3$  approximation to the neutron transport equation; PhD disszertáció, BME NTI, 2021.  
[http://dept.phy.bme.hu/phd/babcsany\\_boglarka/Babcsany\\_dissertation.pdf](http://dept.phy.bme.hu/phd/babcsany_boglarka/Babcsany_dissertation.pdf), Internet 2021. 10. 27.
- [2] Illés Gergely: VVER benchmarkszámítások az SPNDYN végelem-módszeren alapuló reaktorfizikai kóddal, BSc szakdolgozat, BME NTI, 2021.
- [3] Szatmáry Zoltán: Reaktorfizika mérnököknek; Egyetemi jegyzet, BME, 2007
- [4] Horváth András: Perturbációk elméleti és kísérleti vizsgálata a BME Oktatóreaktorán, BME NTI, 2014
- [5] Hajas Tamás: Az időfüggő diffúzióegyenletet végelem-módszerrel megoldó program fejlesztése, Tudományos Diákköri Konferencia dolgozat, BME NTI, 2018.
- [6] Hajas Tamás: Tranziens reaktorfizikai folyamatok vizsgálata saját fejlesztésű, végelem-módszeren alapuló diffúziós kóddal, Tudományos Diákköri Konferencia dolgozat, BME NTI, 2019.
- [7] Hajas Tamás: Az időfüggő diffúzióegyenlet megoldására szolgáló végelem-módszeren alapuló program fejlesztése és APROS rendszerkódhoz csatolása, Diplomamunka, BME NTI, 2020.
- [8] I. Panka , A. Keresztúri & C. J. Hegedűs (2007) Numerical Methods in the Kiko3D Three-Dimensional Reactor Dynamics Code, Transport Theory and Statistical Physics, 36:4-6, 381-419, DOI: 10.1080/00411450701468183
- [9] I. Pócs, Z. Kálya, T. Parkó, M. Horváth, S. Patai Szabó: C-PORCA 7: a nodal diffusion reactor calculation code to support off-line and on-line core analysis at Paks nuclear power plant; Kerntechnik, 84 (2019), pp. 228-241, 10.3139/124.190021
- [10] G. Theler: “Difusión de neutrones en mallas no estructuradas: comparación entre volúmenes y elementos finitos,” lecture note, Universidad Nacional de Buenos Aires & Instituto Balseiro, 2013.
- [11] Mészáros Péter: A BME Oktatóreaktorán végzett mérések modellezése a DIREMO, végelem-módszeren alapuló diffúziós kóddal, OTDK dolgozat, BME NTI, 2021.
- [12] Az AER honlapja: <http://www.aer-web.com/>, Internet 2021. 10. 25.
- [13] A KFKI honlapjának azon oldala, ahonnan a benchmark specifikációk letölthetők: <http://aerbench.kfki.hu/aerbench/>, Internet 2021. 10. 18.
- [14] Gonzalez-Pintor, S. and Verdú, G.: Correction of the rod cusping effect for a high order finite element method; International Conference on Mathematics and Computational Methods Applied to Nuclear Science and Engineering (M&C 2011) Rio de Janeiro, RJ, Brazil, May 8-12, 2011, on CD-ROM, Latin American Section (LAS) / American Nuclear Society (ANS) ISBN 978-85-63688-00-2
- [15] D. Gilbert, J.E. Roman, Wm. J. Garland, W.F.S. Poehlman. Simulating control rod and fuel assembly motion using moving meshes . Annals of Nuclear Energy, 35(2), pp. 291-303 (2008).
- [16] Gonzalez-Pintor, S.: Approximation of the Neutron Diffusion Equation on Hexagonal Geometries; PhD disszertáció, Universitat Politècnica de València, 2012.
- [17] Kyrki-Rajamaki R.: HEXTRAN VVER Reactor Dynamics Code for 3D Transients,

Proceedings of the 1st Symposium of AER, 1991

<https://www.osti.gov/etdeweb/servlets/purl/440063>, Internet 2021. 10. 25.



## 6. Függelék

### A) A két- és háromdimenziós teljesítményeloszlások referenciától való eltérései a Seidel benchmarknál

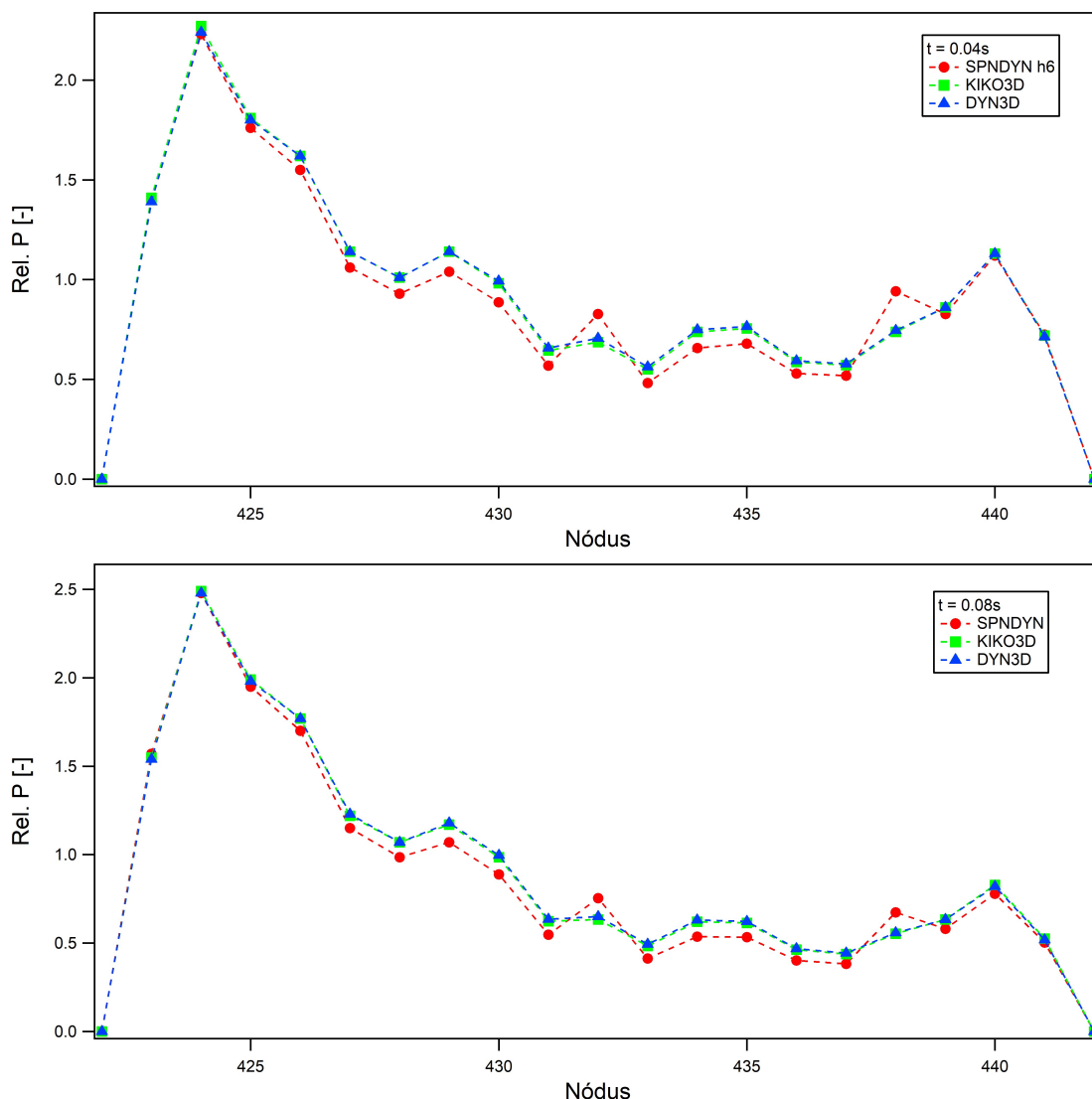
Azonosító	Típus	Teljesítmény	Ref. teljesítmény	Eltérés	
1	4	0.0000	0	0.0000	0.00%
2	1	0.5226	0.5221	0.0005	0.10%
3	2	0.8709	0.87	0.0009	0.13%
4	2	0.9434	0.9425	0.0009	0.10%
5	1	0.7183	0.7178	0.0005	0.07%
6	1	0.5843	0.5842	0.0001	0.02%
7	4	0.0000	0	0.0000	0.00%
8	2	0.9635	0.9639	-0.0004	-0.04%
9	3	1.5071	1.5076	-0.0005	-0.04%
10	2	0.9924	0.9928	-0.0004	-0.04%
11	2	0.8318	0.8309	0.0009	0.11%
12	1	0.7340	0.7333	0.0007	0.10%
13	2	0.9437	0.943	0.0007	0.08%
14	2	0.8869	0.8864	0.0005	0.06%
15	2	0.7784	0.7783	0.0001	0.02%
16	1	0.7101	0.7104	-0.0003	-0.04%
17	1	0.9458	0.9462	-0.0004	-0.04%
18	3	1.4214	1.422	-0.0006	-0.04%
19	3	0.8802	0.8803	-0.0002	-0.02%
20	2	0.9531	0.9523	0.0008	0.08%
21	1	0.7476	0.7472	0.0004	0.05%
22	1	0.7483	0.7481	0.0002	0.02%
23	2	1.0187	1.0188	-0.0001	-0.01%
24	2	1.1865	1.1869	-0.0004	-0.03%
25	3	1.5189	1.5193	-0.0005	-0.03%
26	3	1.1268	1.1271	-0.0003	-0.03%
27	2	0.9778	0.9775	0.0003	0.03%
28	1	0.8404	0.8405	-0.0001	-0.02%
29	1	0.9466	0.9469	-0.0003	-0.03%
30	1	1.0067	1.0071	-0.0004	-0.04%
31	3	1.3057	1.3061	-0.0004	-0.03%
32	2	1.2051	1.2053	-0.0002	-0.02%
33	2	1.2977	1.2981	-0.0004	-0.03%
34	3	1.4205	1.421	-0.0005	-0.04%
35	3	0.8871	0.8873	-0.0002	-0.02%
36	2	1.1977	1.1983	-0.0006	-0.05%
37	3	1.0127	1.013	-0.0002	-0.02%

4. táblázat. A Seidel 2D benchmark számított és a referenciaként rendelkezésre álló relatív teljesítményeinek összehasonlítása.

Az.	Réteg										Teljes kazetta
	1	2	3	4	5	6	7	8	9	10	
1	-0.09%	-0.14%	-0.12%	-0.12%	-0.12%	0.00%	0.00%	0.00%	0.00%	0.00%	-0.12%
2	-0.14%	-0.14%	-0.15%	-0.13%	-0.11%	-0.14%	-0.02%	-0.08%	0.16%	-0.07%	-0.24%
3	-0.14%	-0.08%	-0.11%	-0.05%	-0.05%	0.00%	0.04%	0.06%	0.17%	0.36%	-0.04%
4	-0.15%	-0.06%	-0.09%	-0.09%	-0.07%	-0.03%	-0.01%	0.01%	0.17%	0.33%	-0.05%
5	-0.16%	-0.06%	-0.07%	-0.09%	-0.11%	-0.03%	0.03%	-0.07%	0.14%	0.10%	-0.07%
6	-0.02%	-0.02%	-0.04%	-0.05%	-0.03%	0.02%	0.02%	0.02%	0.22%	-0.01%	0.00%
7	-0.12%	-0.09%	-0.06%	-0.05%	-0.01%	0.03%	0.07%	0.11%	0.09%	0.37%	-0.02%
8	-0.03%	-0.09%	-0.07%	0.00%	-0.02%	0.03%	0.04%	0.06%	0.03%	-0.06%	0.00%
9	-0.05%	-0.04%	-0.06%	-0.02%	0.01%	0.03%	-0.03%	0.04%	-0.01%	-0.16%	-0.02%
10	-0.01%	0.00%	-0.01%	-0.02%	0.03%	0.03%	0.13%	0.13%	0.08%	0.16%	0.04%
11	-0.15%	-0.03%	-0.10%	-0.03%	-0.07%	0.01%	-0.01%	-0.03%	0.11%	0.33%	-0.04%
12	0.05%	0.01%	-0.01%	0.01%	-0.01%	0.00%	0.07%	0.05%	0.02%	0.27%	0.02%
13	-0.10%	-0.08%	-0.03%	-0.07%	-0.05%	-0.03%	-0.03%	0.08%	0.20%	0.14%	-0.01%
14	-0.12%	-0.02%	-0.04%	0.01%	-0.01%	0.01%	0.04%	0.08%	0.10%	0.14%	0.00%
15	0.06%	-0.04%	0.01%	0.01%	0.00%	0.02%	0.10%	0.08%	0.09%	0.03%	0.04%
16	-0.01%	-0.06%	-0.08%	-0.03%	-0.06%	-0.06%	-0.10%	0.04%	-0.09%	-0.12%	-0.04%
17	0.08%	-0.01%	0.01%	0.02%	0.03%	-0.03%	0.06%	0.14%	0.15%	0.26%	0.00%
18	0.10%	-0.02%	0.00%	-0.01%	0.01%	0.07%	0.08%	0.11%	0.07%	0.07%	0.03%
19	-0.10%	-0.01%	-0.07%	-0.01%	-0.01%	0.01%	0.08%	0.02%	0.06%	-0.01%	0.01%
20	-0.07%	0.00%	-0.02%	-0.05%	0.00%	0.00%	0.02%	0.06%	0.23%	0.05%	0.01%
21	-0.06%	-0.12%	-0.09%	-0.05%	-0.02%	-0.04%	0.00%	0.11%	0.14%	0.21%	-0.03%
22	0.08%	0.01%	0.01%	-0.02%	-0.03%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
23	0.05%	-0.01%	-0.05%	0.00%	-0.01%	-0.08%	0.06%	-0.06%	0.08%	0.22%	-0.02%
24	-0.02%	0.01%	-0.01%	0.00%	0.04%	0.03%	0.10%	0.06%	0.19%	0.35%	0.04%
25	-0.07%	-0.08%	-0.09%	-0.06%	-0.02%	-0.04%	0.04%	0.08%	0.10%	-0.06%	-0.03%
26	-0.06%	-0.02%	-0.01%	-0.03%	-0.02%	0.05%	0.09%	0.06%	0.16%	0.06%	0.02%
27	0.02%	-0.09%	0.00%	-0.05%	0.00%	0.05%	0.03%	0.15%	0.05%	0.43%	0.02%
28	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
29	0.07%	0.01%	-0.02%	0.00%	-0.01%	0.04%	0.02%	0.07%	0.05%	0.26%	0.02%
30	0.03%	-0.07%	-0.05%	-0.07%	-0.03%	0.00%	0.00%	0.04%	0.04%	0.03%	-0.02%
31	0.03%	0.01%	-0.04%	-0.02%	0.01%	0.07%	0.06%	0.14%	0.23%	0.08%	0.03%
32	-0.01%	0.01%	-0.03%	-0.01%	0.01%	0.05%	0.09%	0.18%	0.26%	0.15%	0.04%
33	-0.06%	0.00%	-0.06%	0.01%	-0.03%	0.06%	0.02%	0.16%	0.21%	0.00%	0.03%
34	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
35	0.03%	0.00%	-0.02%	0.02%	0.01%	0.04%	0.09%	0.09%	0.13%	0.34%	0.04%
36	0.08%	-0.07%	-0.04%	0.00%	-0.01%	0.01%	0.06%	0.15%	0.13%	0.22%	0.02%
37	0.01%	-0.06%	-0.04%	-0.03%	0.02%	0.01%	0.07%	0.07%	0.26%	0.37%	0.03%
38	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
39	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
40	-0.12%	-0.03%	-0.07%	-0.06%	-0.01%	-0.02%	0.08%	0.13%	0.13%	-0.07%	-0.01%
41	0.03%	-0.06%	0.02%	-0.01%	0.02%	0.03%	0.08%	0.21%	0.20%	0.49%	0.05%
42	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
43	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
44	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%

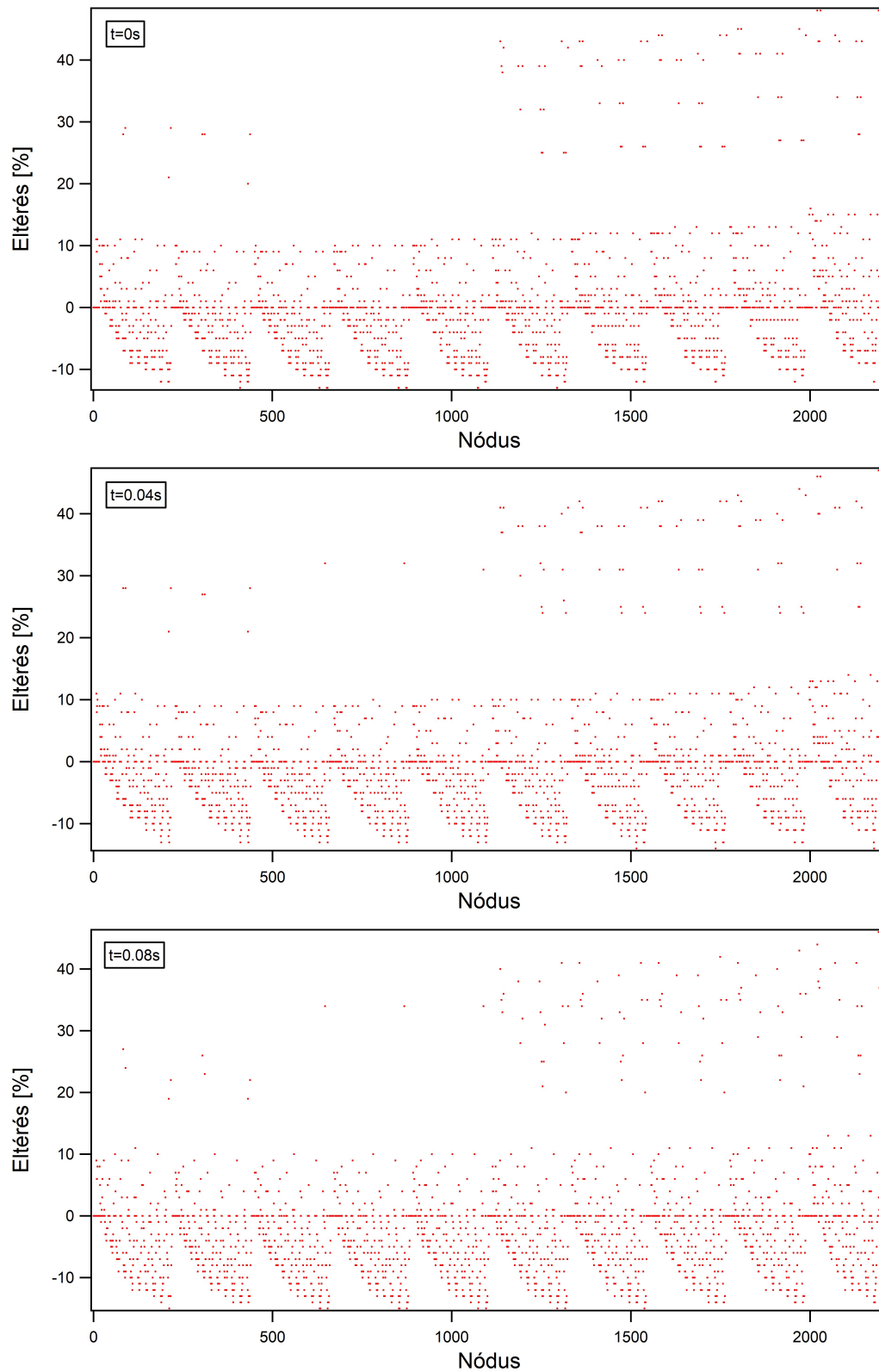
5. táblázat. A Seidel 3D benchmark számított és referencia teljesítményeinek összehasonlítása.

## B) 3D teljesítményeloszlás további ábrái a Dyn001 benchmarknál

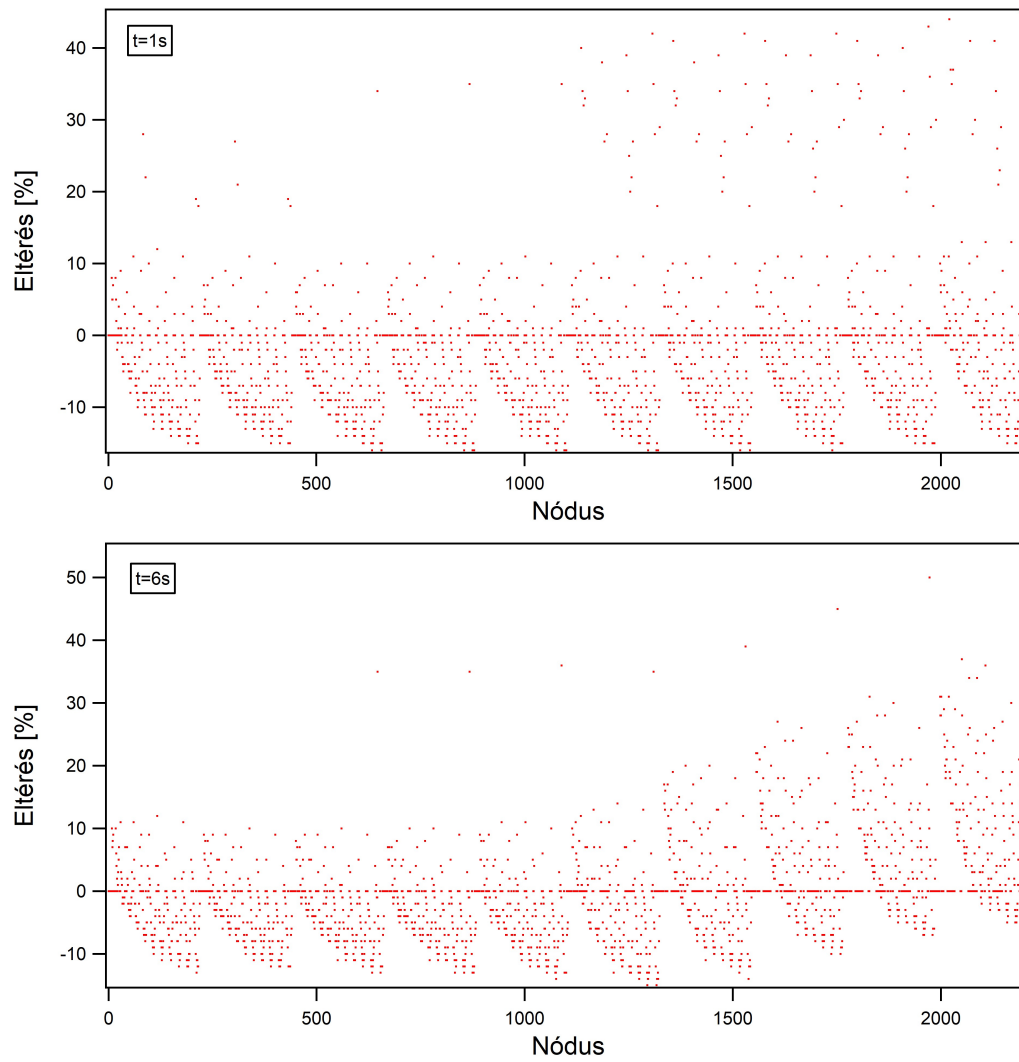


19. ábra. 3D teljesítményeloszlás két további időpontokban alulról a harmadik rétegben a középvonal mentén. A kilöködő szabályozókazettát itt a 426-os számú nódus reprezentálja.

Terjedelmi okokból a 3D teljesítményeloszlások eltéréseit az összes nódus és időpont tekintetében csak grafikus formában volt lehetőség közölni, de talán így szemléletesebb is. A nódusok számozása a 10. ábrán a radiális metszeten balról-jobbra és fentről-lefelé, míg axiálisan letről felfelé halad. Fontos, hogy a referenciák 10 axiális rétegre vonatkoznak és az egyes időpontokban normálva vannak, azaz a rétegek teljesítményét én kettesével összeadtam normálás előtt. A 3D teljesítményeloszlásban az axiális reflektorok nincsenek benne, a radiálisak viszont 0.0 teljesítménnyel igen, így itt összesen 2210 nódus teljesítményének KIKO3D referenciától való eltérése lett ábrázolva.



20. ábra. A 3D teljesítményeloszlás KIKO3D referenciához viszonyított eltérései a Dyn001 benchmarknál.



21. ábra. A 3D teljesítményeloszlás KIKO3D referenciához viszonyított eltérései a Dyn001 benchmarknál. (Folytatás.)