



M Ű E G Y E T E M 1 7 8 2

Budapesti Műszaki és Gazdaságtudományi Egyetem
Természettudományi Kar

TDK DOLGOZAT

Brachyterápiás célú GPU alapú Monte Carlo szoftver fejlesztése

Klausz Milán

Témavezető: Dr. Légrády Dávid
egyetemi docens
BME Nukleáris Technikai Intézet,
Nukleáris Technika Tanszék

BME

2014

Önállósági nyilatkozat

Alulírott, Klausz Milán, a Budapesti Műszaki és Gazdaságtudományi Egyetem fizikus MSc szakos hallgatója kijelentem, hogy ezt a dolgozatot meg nem engedett segédeszközök nélkül, önállóan, a témavezető irányításával készítettem, és csak a megadott forrásokat használtam fel. Minden olyan részt, melyet szó szerint, vagy azonos értelemben, de átfogalmazva más forrásból vettem, a forrás megadásával jelöltem.

Budapest, 2014. október 27.

.....
Klausz Milán

Tartalomjegyzék

Bevezetés	1
1. A program leírása	2
1.1. A program működése	2
1.2. A program kezelése	3
2. A program fejlesztése	5
2.1. A pontosság növelése	5
2.2. A kód optimalizálása	6
2.3. A Woodcock módszer és a sugárkövetés	6
3. A sugárkövetés és a Woodcock módszer összehasonlítása	9
3.1. Első kölcsönhatás egy egyenes mentén	9
3.2. Pontforrás szimulációja	12
3.2.1. Homogén vízfantom	13
3.2.2. Tüdőt és csontot tartalmazó rendszer	17
3.3. Kiterjedt sugárforrás szimulációja	20
3.3.1. Összehasonlítás az Oncentra-Brachy szoftverrel	21
3.3.2. Futási idők összehasonlítása	23
3.3.3. Figure of Merit kiterjedt forrásnál	24
Összefoglalás és kitekintés	27

Bevezetés

A rákos megbetegedések második leggyakrabban alkalmazott kezelési eljárása a sugárterápia. Ennek egyik módja a közel-besugárzás, más néven brachyterápia, amelynél a sugárforrást közvetlenül a daganatba vagy annak környezetébe helyezik el. A kezelés elengedhetetlen része a besugárzás megtervezése, amit számítógéppel hajtanak végre. A jelenleg használatban lévő tervezőprogramok vízfantomban mért adatok alapján határozzák meg a dózisteret. Ez egyben azt is jelenti, hogy nem veszik figyelembe a testüregekben lévő levegőt vagy éppen a csontokat. Erre megoldást jelenthetne egy Monte Carlo alapú szimulációs program, ezek hosszú futási ideje viszont ellehetetleníti az orvosi célú felhasználásukat. A számítástechnika – és azon belül a grafikuskártyák – fejlődésével azonban mára már ez a probléma is kiküszöbölhetővé vált. A modern grafikuskártyák hatalmas számítási teljesítménye, kiegészülve az újszerű GPU programozással, lehetőséget nyújt a megfelelő gyorsaságú programok írására.

A céloom tehát egy olyan grafikuskártyán futó Monte Carlo szimulációs program létrehozása, amely elegendően pontos és gyors ahhoz, hogy brachyterápiás tervezésre alkalmas legyen. TDK dolgozatomban a BSc-s szakdolgozatom keretén belül lefektetett alapok továbbfejlesztésén dolgoztam. A hangsúlyt a program pontosságának növelésére és gyorsítására fektettem.

1. fejezet

A program leírása

A szakdolgozati munkám során létrehoztam egy kezdetleges brachyterápiás tervezésre alkalmas szoftvert, a hozzá tartozó kezelőfelülettel, és validáltam a program működését pont- és kiterjedt forrásokra. Az eredmények nagyon biztatóak lettek, de a futási idők messze túl nagyok voltak ahhoz, hogy a programot ténylegesen használni lehessen besugárzás tervezésére. Ennek legfőbb oka az volt, hogy a program alapját egy PANNI névre hallgató, pozitronemissziós tomográfia Monte Carlo módszerrel történő modellezésére alkalmas, grafikuskártyára implementált szoftver adta. (Cserkaszky, 2013) Ebből kifolyólag ugyanis a program olyan Monte Carlo módszereket használ, melyek PET szimulációhoz optimálisak.

A következőkben röviden ismertetem a program működését és a kezelésére alkalmas felületet, melyekről részletesebben a szakdolgozatomban lehet olvasni. (Klausz, 2014)

1.1. A program működése

A létrehozandó brachyterápiás tervezőprogramnak alkalmasnak kell lennie arra, hogy orvosi képalkotóeszközökből nyert szöveti eloszlásinformációk és a forrásgeometriák ismerete alapján dózisteret számoljon. A programnak tehát első lépésben be kell tudnia olvasni a vizsgálni kívánt anyageloszlást, és a felhasználó igényének megfelelően kell kialakítania benne a forráseloszlást. Ezután a forrásanyagok és a szövetek paramétereit, hatáskeresztmetszeteit felhasználva a fotontranszport Monte Carlo szimulációjával kell becsülnie a dóziseloszlást. Az eredményeket megfelelő pontossággal, a lehető legrövidebb idő alatt kell megkapnia, majd azokat a felhasználó számára értelmezhető módon megjelenítenie. Ennek megfelelően a program működéséhez alapvetően négyféle fájlra van szükség: az anyag- és forráseloszlást megadó fájlokra, a felhasznált anyagok hatáskeresztmetszeteit tartalmazó fájlokra, illetve a program futtatásához elengedhetetlen paraméterek megadását szolgáló úgynevezett konfigurációs fájlra.

A szimulálható geometriai elrendezések azonos méretű elemi téglatesteből, úgynevezett voxelekből épülnek fel. Az anyag- és forráseloszlások definiálása voxelenként történik, az erre szolgáló fájlokban. A felhasznált anyagok határkeresztmetszeteit tartalmazó fájlok az XCOM programmal készített fájlok. (NIST, 2014) Ezeket, és az eloszlásokat leíró fájlok elérési útvonalait kell megadni többek között a konfigurációs fájlban. Itt kell megadni továbbá az anyagok sűrűségét, a vizsgálandó geometria adatait, és az eredményfájlok kiírási helyét is.

A program a beolvasott fájlok alapján az indítandó fotonokat listákba rendezi a párhuzamos futtatás érdekében, elvégzi a fotontranszportot, és az eredményeket kiírja az erre létrehozott fájlba.

A fotonok indítása történhet egy adott voxel teljes térfogatából, vagy egy kiválasztott pontból is. Az irányorsorolás alapvetően egyenletes, de szükség esetén könnyedén módosítható. A szabad úthossz sorsolása Woodcock módszerrel történik. A transzport során implicit befogást (implicit capture) használ a kód, így minden kölcsönhatás során történik fotoeffektus és Compton-szórás is. Előbbi során a fotonhoz rendelt súly, utóbbinál a foton energiája csökken, illetve szóródásnál a haladási irány is változhat. Az új irány meghatározása táblázatos módszerrel történik, a Klein-Nishina formula felhasználásával.

1.2. A program kezelése

Mivel a program végső célja az orvosi célú felhasználás, mindenképpen szükség van egy olyan kezelőfelületre, amellyel az könnyen irányítható, a programkód beható ismerete nélkül is. Ennek létrehozására a Matlab környezetet választottam. Egy olyan függvényt fejlesztettem ki, amely képes a bemenő fájlok létrehozására, a program elindítására és eredményeinek beolvasására, illetve azok feldolgozására és mentésére is.

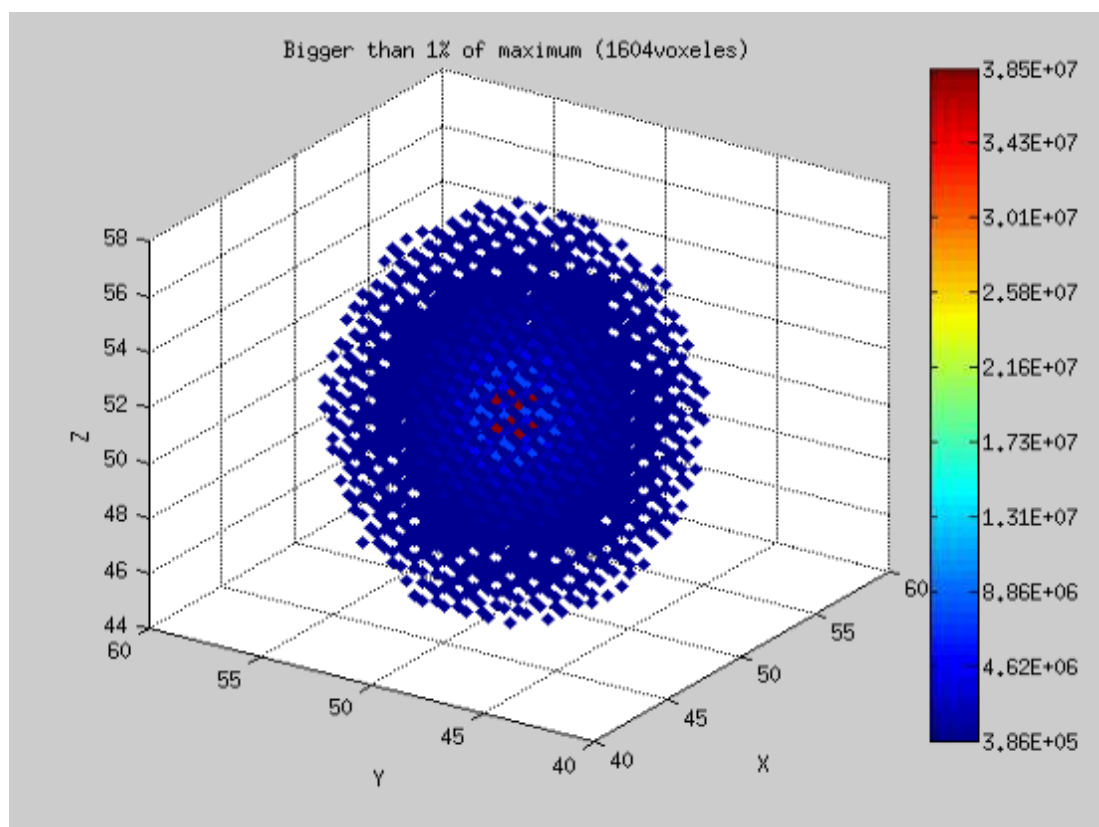
Az elérhető beállítások a geometria határai milliméterben megadva és az ezekre vonatkozó felosztások, azaz a voxelek számai. Az anyag és forráseloszlások létrehozhatók manuálisan – azaz voxelenként megadva –, vagy téglatest alakú források esetén a méretek és az anyagok megadásával, a geometria közepére illesztett – igény szerint borítással ellátott – forrás modellek automatikusan is elkészíthetőek.

Az eredmények feldolgozása két lépésben zajlik. Először a Matlab függvény egy statisztikát készít, hogy számszerű értékeket tudjon szolgáltatni, majd különböző módokon ábrázolja az adatokat, hogy a felhasználó vizuális információkat is nyerjen a futtatás eredményeiről.

A statisztikában szerepelnek többek közt a vizsgált geometriát leíró mennyiségek, a kiszökött fotonok száma és összenergiája, valamint a program futási ideje is.

Az eredmények ábrázolása kétféleképpen zajlik: két-, illetve három dimenzióban. A kétdimenziós ábrákon kiválasztott egyenesek mentén mért nyers, illetve különféleképpen normált adatok láthatóak. A háromdimenziós ábrázolást egy plot3c nevű függvény végzi, mely különböző színek segítségével jeleníti meg az értékeket. Egy energia-színskálát is elhelyez az ábra oldalán, így egy forgatható, nagyítható és egyéb módokon is manipulálható háromdimenziós képet ad az eredmények térbeli eloszlásáról. (Theune, 2005)

Az 1.1. ábrán demonstrációs jelleggel egy pontforrásból származó fotonok által homogén vízfantomban leadott energiák látszanak. Az átláthatóság kedvéért csak a maximális érték 1%-ánál nagyobb adatokat jelenítettem meg.



1.1. ábra. Pontforrás tere homogén vízfantomban.

2. fejezet

A program fejlesztése

A program legnagyobb problémája jelenleg a hosszú futási idő. Elsősorban ezen próbáltam meg segíteni a kód optimalizálásával, és egy új szabadúthossz-sorsoló módszer leprogramozásával. Mindeközben észrevettem, hogy bizonyos indított fotonszám fölött egyes voxelekben már nem nő tovább az elnyelt energia értéke, hanem egy egységes értékben maximalizálódik. Igyekeztem felkutatni ennek a hibának a forrását, és megfelelő megoldást találni rá.

2.1. A pontosság növelése

A voxelek említett telítődését az okozta, hogy egy 'float' típusú változóban – pontosabban vektorban – tárolta a program az egyes elemi térrészekben elnyelt energiák összegét, és kölcsönhatásonként adta hozzá az új értékeket. Mivel irídium esetén 380 keV-os energiával indultak a fotonok, ez volt a legnagyobb energiaérték, amely egy lépésben hozzáadódhatott a már eltárolt összeghez. A 'float' számábrázolásról tudni kell, hogy hatványkitevős felíráson alapszik, azaz a (2.1) képletnek megfelelően tárolja a számokat.

$$e \cdot a \cdot 2^x, \tag{2.1}$$

ahol 'e' az előjel, 'a' az úgynevezett mantissza, és 'x' a kitevő. A 'float' számoknál 1 biten tárolódik az előjel, 23 biten a mantissza, és 8 biten az előjeles kitevő.

Ez azt eredményezi, hogy ha 380-at adunk mindig egy ilyen típusú változóhoz, és az 'x' kitevő eléri a 8-at, akkor az 'a' – és így az eltárolt szám – nem nő tovább. Ennek megfelelően az $a=2^{23}$, az $x=2^8$, vagyis 2^{31} lesz az így elérhető maximális érték, amit a tapasztalat is megerősített.

Ennek a problémának a megoldására az eredmények tárolására használt tömb típusát 'float'-ról 'double'-re változtattam. Ezzel a probléma megoldódott, hiszen így az 'a' érték

52 biten tárolódik, ami több mint elég a program helyes működéséhez.

A program viszont az NVIDIA CUDA platformján íródott, mely nem támogatja az összeadást végző 'atomicAdd' függvény használatát 'double' változókkal. Erre a függvényre azért van szükség, hogy a párhuzamosan futó szálak helyesen tudjanak ugyanarra a memóriacímre adatot írni. Azt a megoldást találtam, hogy hozzáadtam a kódhoz az NVIDIA által javasolt, 'double' értékek összeadására is képes 'atomicAdd' függvényt. (NVIDIA, 2009) Az átalakítások eredménye sajnos az lett, hogy a program nagyjából 10%-kal lelassult, viszont ez mindenképpen szükséges áldozat volt a pontos működéshez.

2.2. A kód optimalizálása

A szakdolgozati munkám célja a program létrehozása és hitelesítése volt, ami felemész-tette a rendelkezésemre álló időt, így nem igazán került sor a kód optimalizálására. Több helyen is sikerült olyan átalakításokat végezni, amiktől mérhetően felgyorsult a program.

Ilyen volt például az, hogy az implicit befogás és a Compton-szórás során leadott energiákat korábban két külön lépésben írta a program megfelelő memóriacímre, az 'atomicAdd' függvény segítségével, de mostanra ez már egy lépésben valósul meg. Ebből kifolyólag az azonos memóriacímre író párhuzamos szálaknak kevesebb időt kell várniuk egymásra, hogy befejezzék az összeadást.

Ezen kívül a program szerkezetét több helyen úgy alakítottam át, hogy kevesebb fel-tételes elágazás szerepeljen benne, ami rontaná a párhuzamos futtatás hatásfokát.

Összességében az optimalizálás eredménye az lett, hogy a 'double' eredménytárolásra való áttérés ellenére a program futási ideje hosszú szimulációk esetén körülbelül 40%-kal csökkent.

2.3. A Woodcock módszer és a sugárkövetés

Az optimalizálással elért futási idő csökkenése remek ugyan, de már a szakdolgozat készítése közben kiderült, hogy a fő problémát a szabad úthossz sorsolás és a részecs-kék mozgásának követése okozza. A program leírásában szerepelt, hogy ezt a folyamatot a program a Woodcock módszerrel végzi, mely nagyon érzékeny a felhasznált anyagok hatáskeresztmetszetére. Például 10^{10} foton szimulációja $20 \times 20 \times 20$ cm³-es homogén víz-fantomban 153,8 másodpercig tart, míg ha akármilyen kis mennyiségű irídium is található a rendszerben, akkor ez majdnem 4,6 órát vesz igénybe. Ebből tehát látszik, hogy hiába

az optimalizálással nyert majdnem 3 óra – korábban ez a futtatás 7,5 óráig tartott – nagyságrendi gyorsulás csak a módszer felváltásával érhető el. Az új módszer, melyet programoztam és kipróbáltam, a sugárkövetés (ray-tracing). A továbbiakban ennek a verifikálását és a Woodcock módszerrel történő összehasonlítását fogom taglalni, de előbb nézzük meg, hogy hogyan is működnek ezek.

A Woodcock módszer

A Woodcock módszer során a program meghatározza adott energián az összes anyag teljes hatáskeresztmetszetei közül a legnagyobbat, ez lesz a majoráns hatáskeresztmetszet. Ezután elindul egy ciklus, amely először a foton helye és energiája alapján meghatározza a fotont tartalmazó voxel anyagának hatáskeresztmetszetét. A következő lépésben szabad úthosszt sorsol a (2.2) egyenlet alapján, a majoráns hatáskeresztmetszetet használva, majd a (2.3) összefüggés alapján dönti el, hogy újraindul-e a ciklus. Ha az egyenlőtlenség teljesül, akkor újraindul, ha nem, akkor véget ér a szabad úthossz sorsolás.

$$\Lambda = \frac{\ln(\text{rand}(0, 1))}{\Sigma_m}, \quad (2.2)$$

ahol ' Λ ' a szabad úthossz, ' Σ_m ' a majoráns hatáskeresztmetszet, ' $\text{rand}(0, 1)$ ' pedig egy 0 és 1 közötti, egyenletes valószínűséggel sorsolt véletlen szám.

$$\text{rand}(0, 1) \geq \sigma_t / \Sigma_m, \quad (2.3)$$

ahol ' σ_t ' az előző lépésben a fotont tartalmazó voxel anyagának teljes hatáskeresztmetszete.

Ez a módszer remekül működik olyan rendszereknél, ahol nincsen kiugróan magas hatáskeresztmetszetű anyag. Ilyenkor ugyanis a majoráns hatáskeresztmetszet nagy lesz, és olyan közegen is nagyon kis lépésekkel halad keresztül, ahol az anyag ezt egyáltalán nem indokolná. Nem véletlen, hogy a PANNI ezt a módszert alkalmazza, hiszen PET szimuláció során a fotonok a testben lévő különböző szöveteken haladnak keresztül, melyek között nincsen nagy hatáskeresztmetszet különbség. Brachyterápia során viszont fémtokban lévő sugárforrást juttatnak a szervezetbe, melynek a hatáskeresztmetszete többszöröse a vízének, és ettől rendkívül lecsökken a módszer hatékonysága. Előnyös tulajdonság viszont, hogy a geometria felosztása elméletileg nem befolyásolja a szimuláció időtartamát, így nagy felbontás is könnyen elérhető.

A sugárkövetés

A sugárkövetés első lépéseként a program egy mértékegység nélküli szabad úthosszt sorsol a (2.4) egyenlet szerint, majd egy cikluson belül addig lépteti a fotont a voxelek falain keresztül, amíg teljesül a (2.5) egyenlőtlenség. Amikor ez az egyenlőtlenség már nem áll fenn, akkor az utolsó lépés hosszát a (2.6) egyenlet adja meg.

$$\Lambda = -\ln(\text{rand}(0, 1)), \quad (2.4)$$

ahol ' Λ ' a szabad úthossz, ' $\text{rand}(0, 1)$ ' pedig egy 0 és 1 közötti, egyenletes valószínűséggel sorsolt véletlen szám.

$$\sum_{i=1}^x (\Sigma_{t_i} \cdot l_i) < \Lambda, \quad (2.5)$$

ahol ' Σ_{t_i} ' a teljes hatáskeresztmetszete annak az anyagnak, melyben az i -edik lépésben halad, és ' l_i ' az a távolság, amennyit megtesz ebben az anyagban a következő falig.

$$d = \frac{\Lambda - \sum_{i=1}^{n-1} (\Sigma_{t_i} \cdot l_i)}{\Sigma_{t_n}}, \quad (2.6)$$

ahol ' d ' az utolsó lépésben megtett úthossz, ' Σ_{t_n} ' pedig annak az anyagnak a teljes hatáskeresztmetszete, melyben halad.

A program minden lépésben meghatározza a foton helyzete és haladási iránya alapján, hogy melyik falhoz ér el legközelebb. Ehhez praktikusán csak a haladási irányának megfelelő 3 fal síkját vizsgálja meg.

A program sebességét némiképp növeli, hogy ha valamelyik sebességkomponens 0, hiszen akkor csak a többi irányban keresi a következő metszéspontot.

Ennek a módszernek az a nagy előnye a Woodcockkal szemben, hogy a felhasznált anyagok elvileg nem befolyásolják nagy mértékben a szimuláció időtartamát. Hátránya viszont, hogy a geometria nagyobb felbontásával megnő a számítási igény a transzport során, és ezzel együtt a futási idő is.

A kérdés tehát az, hogy melyik módszer vezet gyorsabban megfelelően pontos eredményhez a Brachyterápiában fontos elrendezésekben. Erre a kérdésre keresem a válasz a következő fejezetben.

3. fejezet

A sugárkövetés és a Woodcock módszer összehasonlítása

3.1. Első kölcsönhatás egy egyenes mentén

Az új algoritmus verifikálása és tesztelése érdekében analitikusan is számolható modellt hoztam létre, mely a 3.1. ábrán látható. A kép bal oldalán lévő voxel közepéből, egy pontból indultak a fotonok, a nyílnak megfelelő irányban, egy egyenes mentén. A kékekkel jelölt kockákban víz, a szürkében pedig irídium volt. Azt vizsgáltam, hogy hány foton lép először kölcsönhatásba az anyaggal az irídium, illetve az azt követő víz tartalmú voxelben. A két módszerrel kapott eredményeket egymással és az analitikus számolásból kapott értékekkel hasonlítottam össze.



3.1. ábra. Az egy egyenes menti szimuláció elrendezése.

A (3.1) egyenlet adja meg annak a valószínűségét, hogy egy foton nem lép kölcsönhatásba a homogén, ' Σ ' makroszkopikus hatáskeresztmetszettel rendelkező anyaggal, ' x ' távolságon belül.

$$P(x) = \exp(-\Sigma \cdot x). \quad (3.1)$$

A voxelek élei 2 mm-esek voltak, a víz makroszkopikus hatáskeresztmetszete a vizsgált 380 keV-os fotonokra nézve $1,086 \cdot 10^{-1}$ 1/cm, az irídiumé pedig 5,290 1/cm. Ezeket az adatokat felhasználva, a (3.1) egyenlet segítségével kiszámolható, hogy a fotonok

61,8346%-ának az irídiumban, $7,065 \cdot 10^{-1}\%$ -ának pedig a vizsgált vízkockában kell először kölcsönhatásba lépnie.

A Monte Carlo szimuláció hatékonyságára jellemző adat a 'FoM' (Figure of Merit), melyet a (3.2)–(3.4) egyenletek segítségével lehet kiszámolni:

$$FoM = \frac{1}{r^2 \cdot T}, \quad (3.2)$$

ahol 'T' a szimuláció időtartama másodpercben mérve, 'r²' pedig a tapasztalati szórásnégyzet, melyet a következő összefüggés ad meg:

$$r^2 = \frac{\Sigma(w \cdot D)^2}{(\Sigma(w \cdot D))^2} - \frac{1}{N}, \quad (3.3)$$

ahol 'w' a súlyfüggvény, 'D' a detektorfüggvény, 'N' pedig az indított részecskeszám. Mivel 'w·D' jelen esetben mindig 1, ezért a következőképpen egyszerűsödik az egyenlet:

$$r^2 = \frac{1}{M} - \frac{1}{N}, \quad (3.4)$$

ahol 'M' a vizsgált voxelben először kölcsönható fotonok száma.

A 'FoM' azért jó mérőszám, mert figyelembe veszi, hogy egy Monte Carlo kódnak nem csak pontosnak, de gyorsnak is kell lennie, vice versa.

A sugárkövetés és a Woodcock módszer első összehasonlításaként 10^7 és $1,754 \cdot 10^{11}$ közötti indított fotonszámok esetén vizsgáltam meg az analitikusan számolt 'M/N' értéktől való abszolút relatív eltéréseket ('D') és a 'FoM'-okat. Az irídium tartalmú voxelre kapott eredmények a 3.1. táblázatban, míg a víz tartalmúra kapottak a 3.2. táblázatban találhatóak.

Az adatokból jól látszik, hogy a kézzel számolt eredménytől való relatív eltérések irídium esetén sugárkövetésnél jelentősen kisebbek – habár a Woodcock módszernél is csupán 0,5% körüliek – a víznél viszont általában a Woodcock módszerrel kaptam közelebbi értékeket – még ha az eltérések rendkívül alacsonyak is. A 'FoM' értékeket tekintve viszont teljesen egyértelmű a helyzet; sugárkövetésnél minden esetben nagyobb értékeket kaptam. Ez adott futási idő mellett kisebb szórást, vagy adott szórást esetén kisebb futási időt jelet, tehát Woodcock módszerhez képest jelentős javulást hozott az új eljárás. A 'FoM' értéke mindkét esetben egy adott értékhez konvergál, ami elvárás, tekintve, hogy a szórásnégyzet fordítottan, a futási idő pedig egyenesen arányos az indított fotonszámmal. Ez a konvergencia figyelhető meg a 3.2. és 3.3. ábrákon. A következőkben mindig

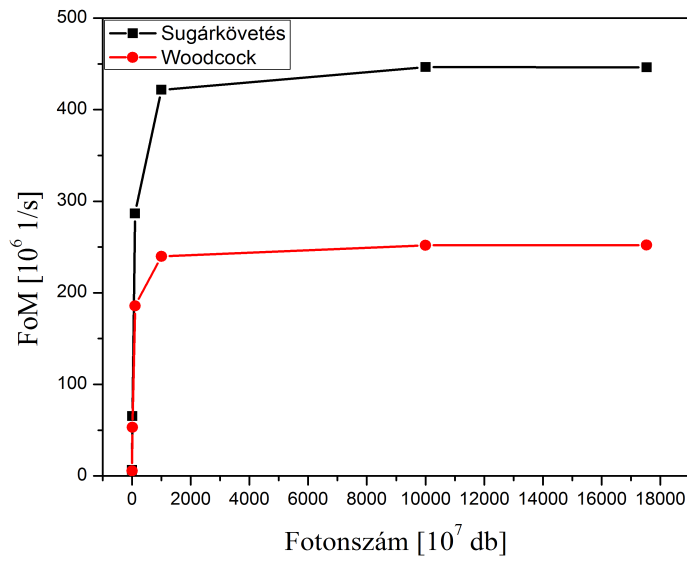
ügyeltem arra, hogy állandósult 'FoM' érték esetén végezzem az összehasonlítást, hiszen ekkor megbízhatóan az adatok.

	N	M	D [%]	T [s]	R ²	FoM [1/s]
Sugárkövetés	10 ⁷	6,186·10 ⁶	0,0450	2,466	6,165·10 ⁻⁸	6,578·10 ⁶
	10 ⁸	6,184·10 ⁷	0,0008	2,509	6,172·10 ⁻⁹	6,458·10 ⁷
	10 ⁹	6,183·10 ⁸	0,0147	5,651	6,175·10 ⁻¹⁰	2,866·10 ⁸
	10 ¹⁰	6,183·10 ⁹	0,0107	38,40	6,174·10 ⁻¹¹	4,218·10 ⁸
	10 ¹¹	6,183·10 ¹⁰	0,0110	362,7	6,174·10 ⁻¹²	4,466·10 ⁸
	1,754·10 ¹¹	1,084·10 ¹¹	0,0107	636,5	3,520·10 ⁻¹²	4,462·10 ⁸
Woodcock	10 ⁷	6,216·10 ⁶	0,5187	2,894	6,089·10 ⁻⁸	5,675·10 ⁶
	10 ⁸	6,215·10 ⁷	0,5171	3,097	6,089·10 ⁻⁹	5,303·10 ⁷
	10 ⁹	6,215·10 ⁸	0,5097	8,839	6,090·10 ⁻¹⁰	1,858·10 ⁸
	10 ¹⁰	6,215·10 ⁹	0,5047	68,50	6,091·10 ⁻¹¹	2,397·10 ⁸
	10 ¹¹	6,215·10 ¹⁰	0,5048	651,7	6,091·10 ⁻¹²	2,519·10 ⁸
	1,754·10 ¹¹	1,090·10 ¹¹	0,5050	1142	3,473·10 ⁻¹²	2,521·10 ⁸

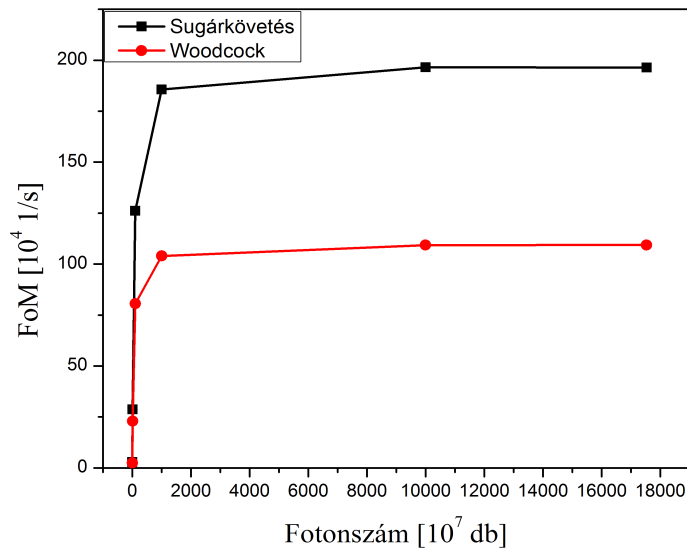
3.1. táblázat. Az irídium tartalmú voxelre kapott szimulált és számított adatok.

	N	M	D [%]	T [s]	R ²	FoM [1/s]
Sugárkövetés	10 ⁷	7,093·10 ⁴	0,0045	2,466	1,400·10 ⁻⁵	2,897·10 ⁴
	10 ⁸	7,062·10 ⁵	0,0004	2,509	1,406·10 ⁻⁶	2,835·10 ⁵
	10 ⁹	7,080·10 ⁶	0,0023	5,651	1,403·10 ⁻⁷	1,262·10 ⁶
	10 ¹⁰	7,079·10 ⁷	0,0022	38,40	1,403·10 ⁻⁸	1,857·10 ⁶
	10 ¹¹	7,080·10 ⁸	0,0024	362,7	1,403·10 ⁻⁹	1,966·10 ⁶
	1,754·10 ¹¹	1,242·10 ⁹	0,0024	636,5	7,998·10 ⁻¹⁰	1,964·10 ⁶
Woodcock	10 ⁷	7,050·10 ⁴	0,0024	2,894	1,408·10 ⁻⁵	2,453·10 ⁴
	10 ⁸	7,060·10 ⁵	0,0007	3,097	1,406·10 ⁻⁶	2,296·10 ⁵
	10 ⁹	7,069·10 ⁶	0,0006	8,839	1,405·10 ⁻⁷	8,054·10 ⁵
	10 ¹⁰	7,071·10 ⁷	0,0010	68,50	1,404·10 ⁻⁸	1,040·10 ⁶
	10 ¹¹	7,071·10 ⁸	0,0010	651,7	1,404·10 ⁻⁹	1,093·10 ⁶
	1,754·10 ¹¹	1,240·10 ⁹	0,0009	1142	8,008·10 ⁻¹⁰	1,094·10 ⁶

3.2. táblázat. A víz tartalmú voxelre kapott szimulált és számított adatok.



3.2. ábra. Figure of Merit a fotonszám függvényében, az irídium tartalmú voxelben.



3.3. ábra. Figure of Merit a fotonszám függvényében, a víz tartalmú voxelben.

3.2. Pontforrás szimulációja

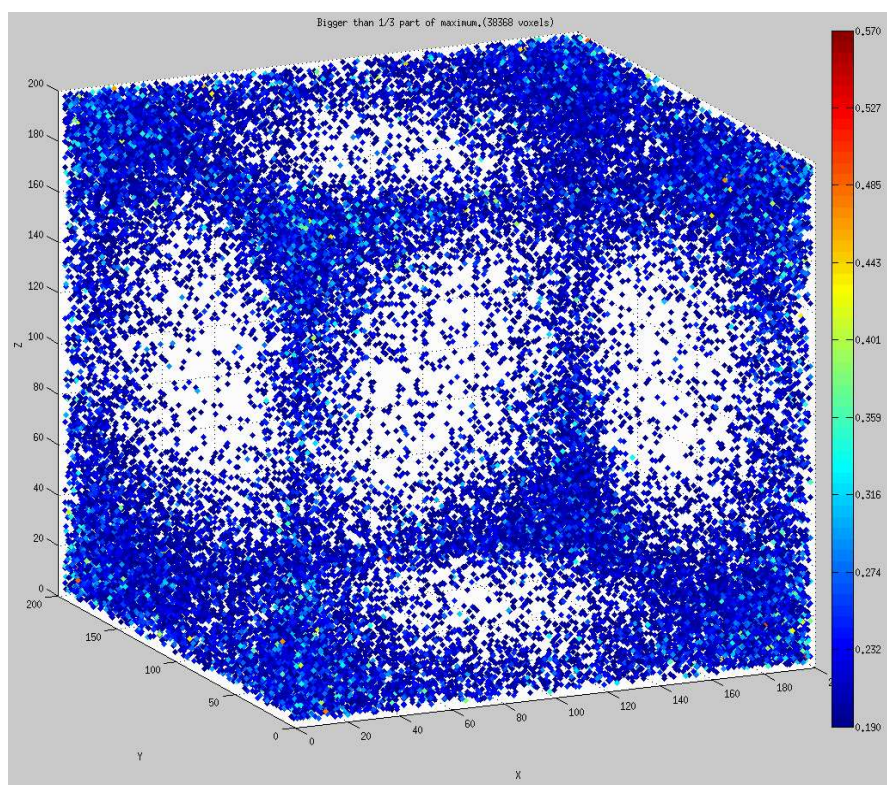
Az első háromdimenziós futtatásokat pontforrásokkal végeztem. A fotonok a geometria közepén lévő voxel közepéből indultak, véletlenszerű irányba, egyenletes eloszlással.

3.2.1. Homogén vízfantom

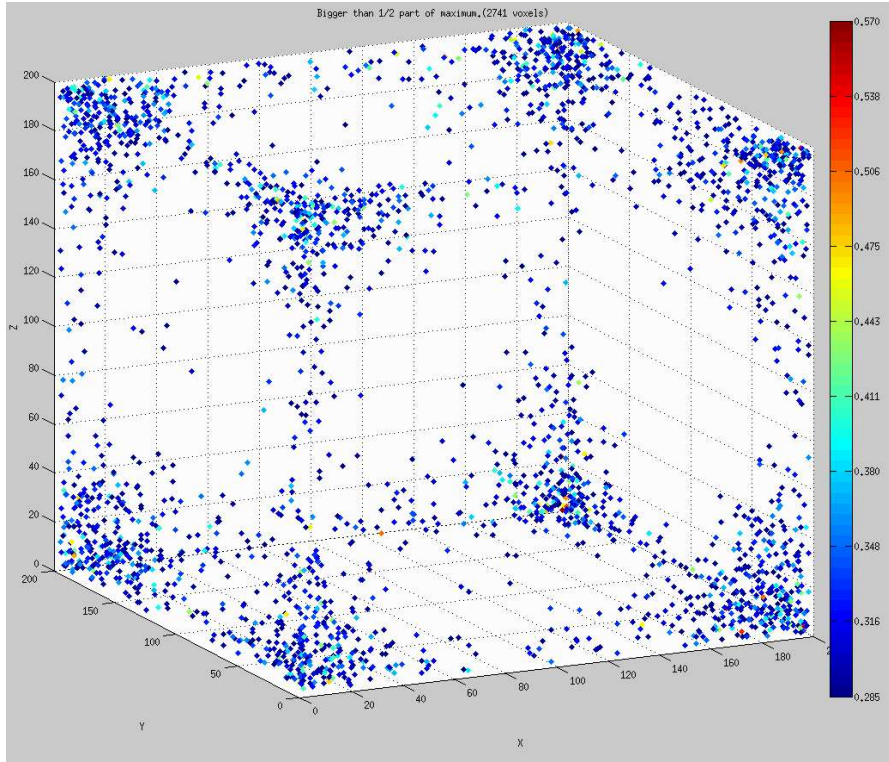
Az első vizsgálatokat $20 \times 20 \times 20 \text{ cm}^3$ -es homogén vízfantomban végeztem. A szimulált fotonok darabszáma 10^8 és 10^{10} között volt, melyek a $200 \times 200 \times 200$ voxelre felosztott térfogat közepén lévő voxelből indultak.

A szakdolgozatom keretén belül, hasonló elrendezésben már összehasonlítottam a Woodcock módszerrel kapott értékeket a hivatalosnak tekinthető MCNP-s futtatások eredményeivel. (Brown, 2009) Akkor kiderült, hogy 10^8 – jelenleg is vizsgált – 380 keV-os foton esetén, a középső 40 mm-es tartományon – ahol a legkisebb a statisztikus hiba – az átlagos relatív eltérés kevesebb, mint 2%. Emellett a kiszökött fotonok számában és a teljes térfogatban elnyelt energiában a relatív eltérés kevesebb, mint 0,5%, illetve 0,2%. Ezen kis értékek miatt a sugárkövetést közvetlenül a Woodcock módszerrel hasonlítottam össze.

Először megnéztem, hogy mekkorák az új módszerrel a voxelekben elnyelt energiák abszolút relatív eltérései a korábbiakhoz képest 10^{10} indított foton esetén. Azért, hogy átlátható ábrákat kapjak, csak a maximális relatív eltérés 1/3 illetve 1/2 részénél nagyobb eltéréseket jelenítettem meg, ezek láthatóak a 3.4. és 3.5. ábrákon.



3.4. ábra. A maximális eltérés 1/3 részénél nagyobb relatív eltérések.



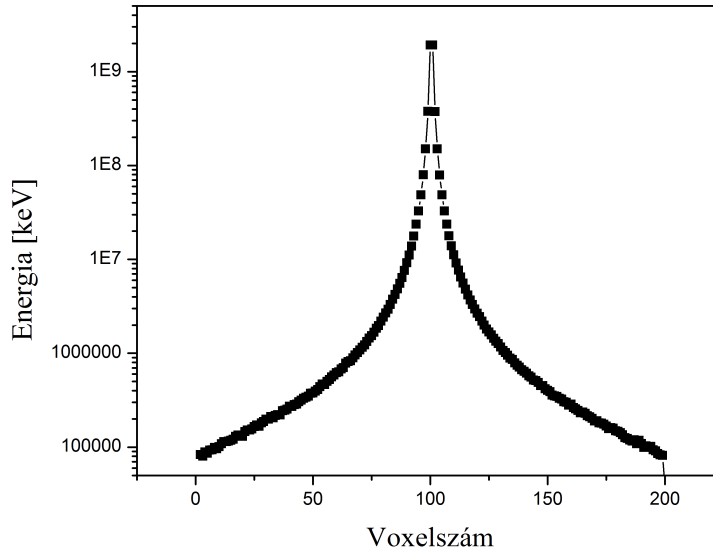
3.5. ábra. A maximális eltérés 1/2 részénél nagyobb relatív eltérések.

Bár a kétdimenziós képek nem adják vissza azt a térhatást, melyet az ábra forgatásával el lehet érni a Matlabban, azért látható, hogy a nagyobb eltérések a középponttól legtávolabbi részeken, a sarkokban vannak. Egy középtengely mentén a voxelekben elnyelt energiák a 3.6. ábrán látható módon csengenek le, amiből látható, hogy a széleken majdhogynem elhanyagolható értékek vannak a közeli tartományokhoz képest. Ráadásul az eltérések mindenhol kisebbek, mint a (3.4) egyenlet alapján becsülhető hibák, tehát a különbség csupán statisztikus hiba.

Amiben viszont jelentős különbség van, az a két verzió futási ideje. A 3.3. táblázatban olvasható értékek egyértelműen igazolják a várakozásokat, vagyis azt, hogy a sugárkövetés ebben az elrendezésben, a futási idő tekintetében jócskán alulmarad a Woodcock módszerhez képest.

	Woodcock	Sugárkövetés
Indított fotonok darabszáma	Futási idő [s]	
10^8	5,213	59,99
10^9	18,03	315,5
10^{10}	152,8	2962

3.3. táblázat. Futási idők homogén vízfantomban.



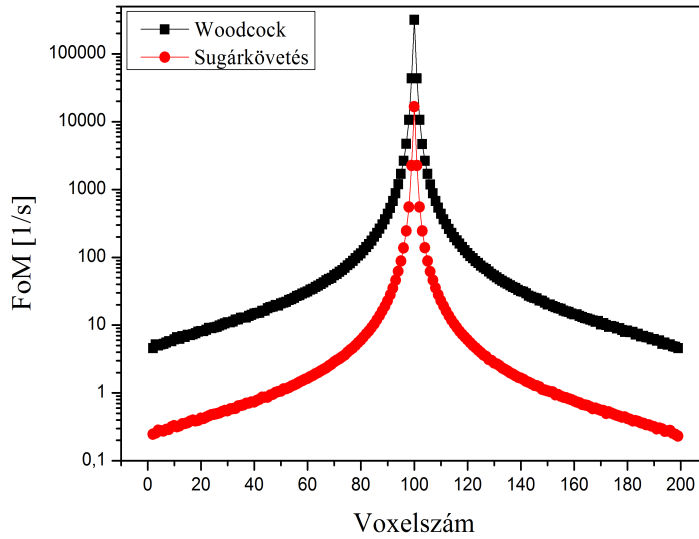
3.6. ábra. Az x-tengely mentén a voxelekben elnyelt energiák $y=100$, és $z=100$ mellett.

A kisebb futási idő önmagában nem jelenti azt, hogy egy kód jobb lenne a másiknál. Ha túl nagy a szórása egy eljárásnak, akkor könnyen előfordulhat, hogy a lassabb módszer hamarabb vezet megfelelő pontosságú eredményhez. Ennek megfelelően a korábban (3.2) és (3.4) egyenletekkel definiált 'FoM' érték eloszlását hasonlítottam össze azonos indított fotonszám mellett sugárkövetés és Woodcock módszer esetén. A szórásnégyzet képletében szereplő 'w' jelenleg az implicit befogásnál használt foton súlya, 'D' pedig a foton energiája.

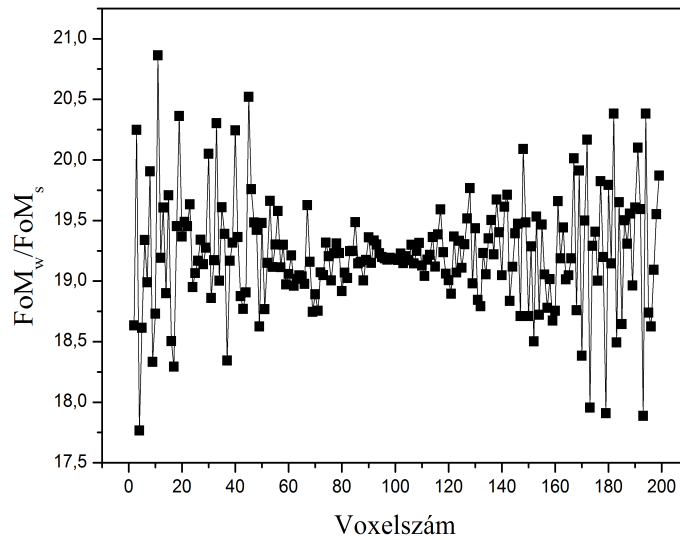
A 'FoM' értékek minden voxelben nagyobbak a Woodcock módszernél, mint a sugárkövetésnél. Az X-középtengely mentén kapott értékek a 3.7. ábrán láthatóak. Ha a 3.8. ábrán látható, Woodcock módszerrel kapott 'FoM' értékek sugárkövetéssel kapott értékeivel vett hányadosát vizsgáljuk, akkor azt tapasztaljuk, hogy a hányados csak kis mértékben változik a voxelszám függvényében. Ez – megvizsgálva a (3.4) egyenlettel definiált tapasztalati szórásnégyzeteket – nem meglepő, hiszen alig tapasztalható különbség köztük, ami azt jelenti, hogy a 'FoM' értékek hányadosával tulajdonképpen a 3.5. képlet szerinti értéket számoljuk ki. Ez elég jól megegyezik a 3.8. ábrán látható középértékekkel. Mindez nem meglepő, hiszen a homogén vízben vett majoráns hatáskeresztmetszet a víz hatáskeresztmetszete, és ugyanezt az értéket használja a sugárkövetés is végig, tehát ebben az elrendezésben a két módszer egybe esik. Különbség ennek megfelelően csak a futási időkből van.

$$\frac{\frac{1}{T_w \cdot r_w^2}}{\frac{1}{T_s \cdot r_s^2}} \approx \frac{1}{T_w \cdot r^2} = \frac{T_s}{T_w} = 19.26, \quad (3.5)$$

ahol ' T_w ' és ' T_s ' a Woodcock módszeres és a sugárkövetéses szimuláció futási ideje, ' r_w^2 ' és ' r_s^2 ' pedig a tapasztalati szórásnégyzete.



3.7. ábra. A Figure of Merit értékek az X-középtengely mentén, homogén vízfantomban.



3.8. ábra. A Figure of Merit értékek hányadosa az X-középtengely mentén, homogén vízfantomban

A homogén vízfantomban végzett szimulációk tehát azt az eredményt hozták, hogy a sugárkövetés ugyanolyan jó eredményeket ad, mint a Woodcock módszer, ráadásul nagyon hasonló szórással, tehát a kód jól működik. Futási időben viszont a várakozásoknak megfelelően nem veszi fel a versenyt a másik módszerrel, mely kifejezetten az ilyen homogén közegekben a leggyorsabb. Ezekből kifolyólag a 'FoM' értékekben a Woodcock módszer a jobb, tehát ezzel kaphatunk gyorsabban, megfelelő pontosságú értékeket.

3.2.2. Tüdőt és csontot tartalmazó rendszer

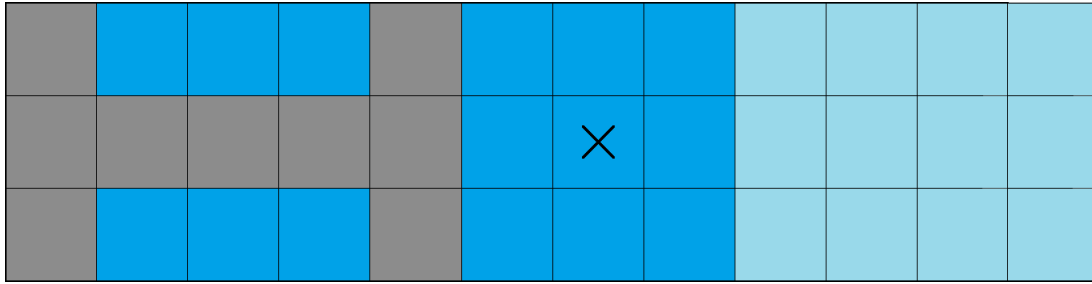
A valósághoz közelebbi esetet próbáltam szimulálni úgy, hogy a rendszerben nem csak víz, de tüdő és csont is jelen volt.

A tüdőt, mint $0,5 \text{ g/cm}^3$ sűrűségű vizet modelleztem, mivel 50% lágyszövetet, és 50% levegőt tartalmaz. A csont sűrűségét $1,85 \text{ g/cm}^3$ -nek vettem, a hozzátartozó hatáske-resztmetszet fájlt pedig az XCOM programmal hoztam létre a 3.4. táblázatban látható összetétel szerint. (NIST, 2014) A futtatásokat 10^{10} fotonnal végeztem, melyek egy voxel közepéből indultak. A teljes rendszer továbbra is $20 \times 20 \times 20 \text{ cm}^3$ -es volt, 1 mm-es felbontással. Szinte a teljes térfogatot víz töltötte ki, de az X-Y síkban – a Z-tengely közepénél – 9 voxel anyagát csontnak, 12-t pedig levegőnek állítottam be. Ezek elrendezése a 3.9. ábrán látható.

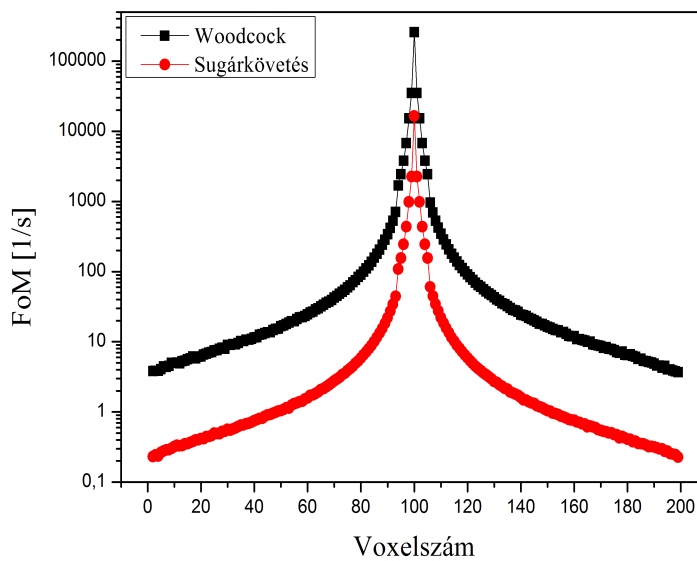
Kémiai elem	Tömegarány
Hidrogén	0.063984
Szén	0.278000
Nitrogén	0.027000
Oxigén	0.410016
Magnézium	0.002000
Foszfor	0.070000
Kén	0.002000
Kalcium	0.147000

3.4. táblázat. A csont összetétele.

A szimuláció lefutása Woodcock módszer esetén 188,9 másodpercet, sugárkövetésnél 2935 másodpercet vett igénybe. A két módszerrel az X középtengely mentén kapott 'FoM' értékek a 3.10. ábrán, ezek hányadosa pedig a 3.11. ábrán látható.

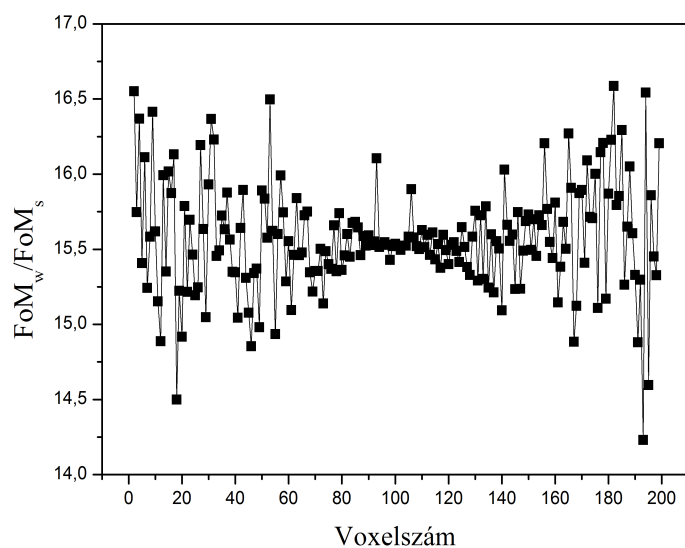


3.9. ábra. A levegőt és csontot tartalmazó szimuláció elrendezése. Az 'X' a forrás helye. Színjelölések: szürke - csont, kék - víz, türkiz - levegő.

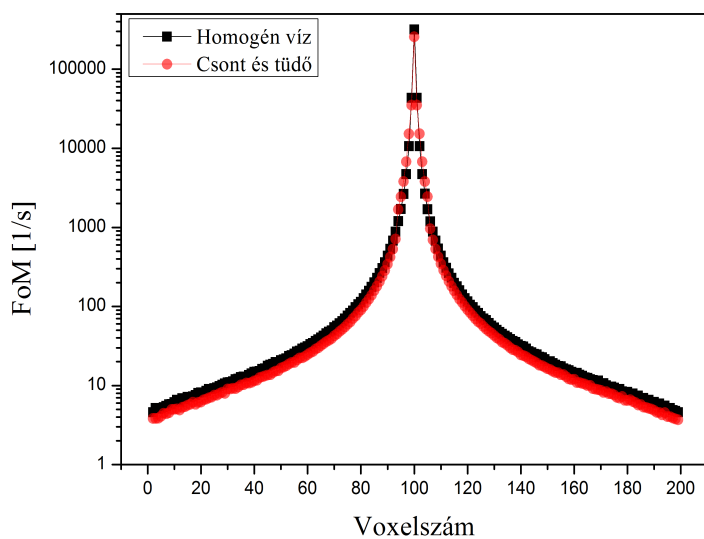


3.10. ábra. A Figure of Merit levegőt és csontot tartalmazó szimuláció esetén, az X-középtengely mentén.

A kapott görbék hasonlítanak a homogén vízben kapottakhoz, de érzékelhető a csont és a tüdő jelenléte. A 'FoM' értékek hányadosánál érdekes módon kiugró értékek vannak a 93. és a 106. voxelekben, vagyis a csont és a tüdő utáni első voxelekben. Ha összehasonlítjuk a homogén vízfantomban kapott adatokkal a 'FoM' hányadosokat, akkor látható, hogy csökkent a Woodcock módszer előnye. Ennek az okát úgy kerestem meg, hogy a 3.12. és 3.13. ábrákon egymás mellett jelenítettem meg a csontot és tüdőt tartalmazó, és csak vízből álló rendszerekben kapott 'FoM' értékeket az X-középtengely mentén mindkét módszer esetén. A két ábrából az derül ki, hogy a Woodcock módszer hatékonyságának csökkenése okozta a 'FoM'-ok arányának csökkenését a valósághibb rendszerben.

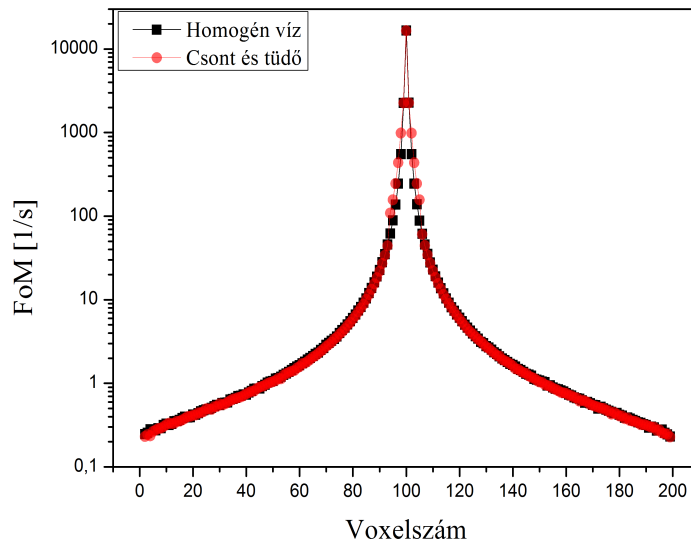


3.11. ábra. A Figure of Merit értékek hányadosa levegőt és csontot tartalmazó szimuláció esetén, az X-középtengely mentén.



3.12. ábra. A Figure of Merit woodcock módszerrel, homogén vízben, valamint levegőt és csontot tartalmazó szimuláció esetén, az X-középtengely mentén.

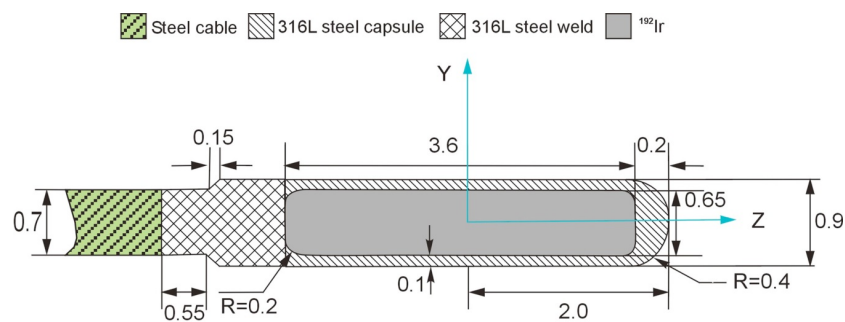
A végső következtetés tehát az, hogy a Woodcock módszer előnye csökkent a sugárkövetéssel szemben, a homogén vízben végzett futtatásokhoz képest, de még mindig jelentősen nagyobb 'FoM' értékeket produkál. A csont miatt megnövekedett a majoráns hatáskeresztmetszet, tehát nem volt elég ahhoz, hogy lerontsa a Woodcock módszer hatékonyságát, így ebben az elrendezésben is jobbnak bizonyult a sugárkövetésnél.



3.13. ábra. A Figure of Merit sugárkövetéssel, homogén vízben, valamint levegőt és csontot tartalmazó szimuláció esetén, az X-középtengely mentén.

3.3. Kiterjedt sugárforrás szimulációja

Szem előtt tartva, hogy a program célja a Brachyterápiás besugárzás tervezése, utolsó lépésként egy kiterjedt sugárforrás dózisterét vizsgáltam meg mindkét módszer segítségével. A szimulált forrás a Nucletron cég ^{192}Ir tartalmú mHDR-v2 forrása volt, mely a 3.14. ábrán látható. A „316L steel capsule” nével jelölt anyag $8,02 \text{ g/cm}^3$ sűrűségű acél, melynek összetétele: 68% Fe, 17% Cr, 12% Ni, 2% Mn és 1% Si. Az irídium mag sűrűsége $22,42 \text{ g/cm}^3$. (Perez-Calatayud, 2012)



Nucletron HDR ^{192}Ir model mHDR-v2 source

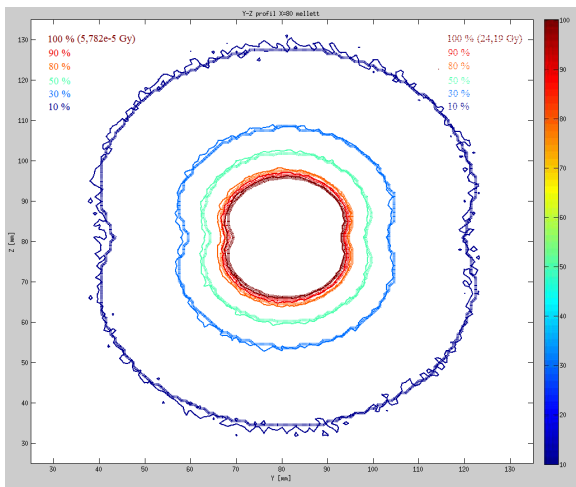
3.14. ábra. Nucletron mHDR-v2 forrás anyagai és méretei (mm).

3.3.1. Összehasonlítás az Oncentra-Brachy szoftverrel

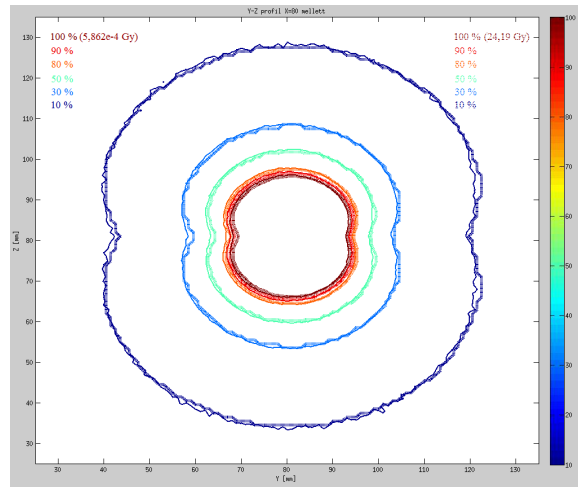
A programom eredményeit először validálásként az Országos Onkológiai Intézetben is használt, Oncentra-Brachy nevű szoftverrel készült szimuláció eredményeivel hasonlítottam össze. Ezek az adatok egy DICOM fájlban álltak rendelkezésemre, melyben a forrás középpontú, $160 \times 160 \times 160$ darab 1 mm^3 -es tartomány dóziseloszlása volt megadva. A Matlab függvényvel beolvastam a fájlt, és a saját futtatási eredményeimből kivágott azonos tartománnyal hasonlítottam össze, 10^9 és 10^{10} foton szimulációja mellett. A kis mennyiségű forrásborítást elhanyagoltam, így a modellem négy darab egymás melletti, 1 mm élhosszúságú irídium kockából állt. A majoráns hatás keresztmetszet akkor is az irídiumé lett volna, ha figyelembe veszem a borítás jelenlétét. A szakdolgozatom keretén belül már elvégeztem a Woodcock módszerrel történő összehasonlítást, így most a sugárkövetéssel végeztem a szimulációt.

Az adatokat a forrástól 10 mm -re mért dóziszra szokás normálni, mivel azonban a kapott fájlban 15 mm -en belül ugyanazok voltak az értékek, ezért az adatsorokat a középponttól 16 mm -re kapott értékekre normáltam. Az eredményekből az 3.15.–3.17-es ábrákon látható metszeteket készítettem. Szaggatott vonal jelöli a hivatalos, folytonos pedig a saját programommal kapott izodózis görbéket 100% , 90% , 80% , 50% , 30% és 10% -os értékeknél.

A kezdetleges forrásmodell ellenére az eltérések – melyek főleg a kisimultabb eredményt adó 10^{10} fotonos szimulációknál látszanak – igen kicsik, igazolva a program működésének helyességét. A Woodcock módszer is hasonlóan jó eredményeket produkált, viszont ahogyan az korábban is szerepelt, rettenetesen hosszú futási idők alatt; 45 perc 10^9 , és több, mint 7 óra 10^{10} foton esetén. Ennek oka az irídium kiugróan magas hatás keresztmetszete, melyet majoráns hatás keresztmetszetként használ fel a program, és így nagyon kis lépésekkel halad a térfogatban azután is, hogy a forrást már maga mögött hagyta.

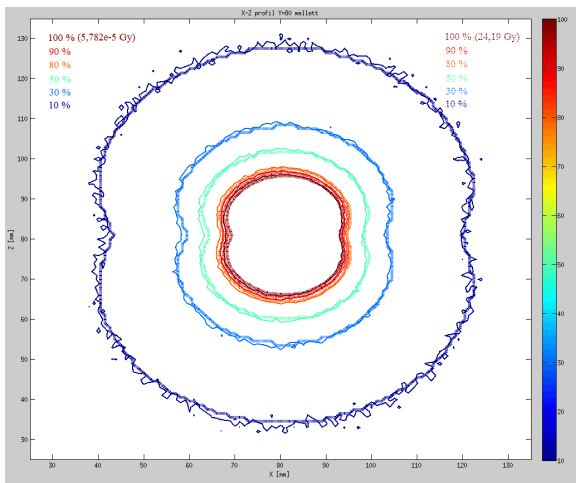


(a) 10^9 foton.

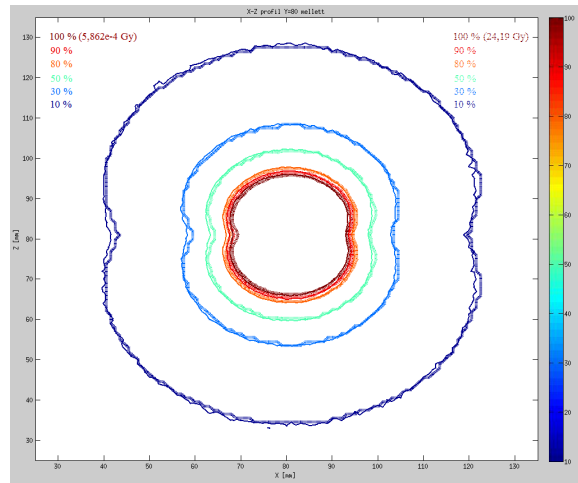


(b) 10^{10} foton.

3.15. ábra. Y-Z profil az X-tengely közepénél, mHDR-v2 forrás szimulációnál.

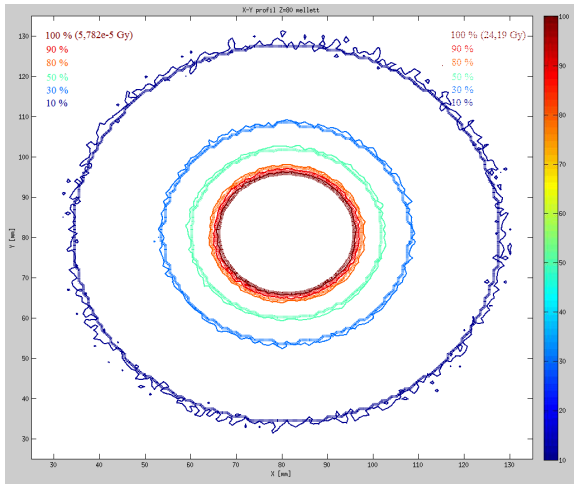


(a) 10^9 foton.

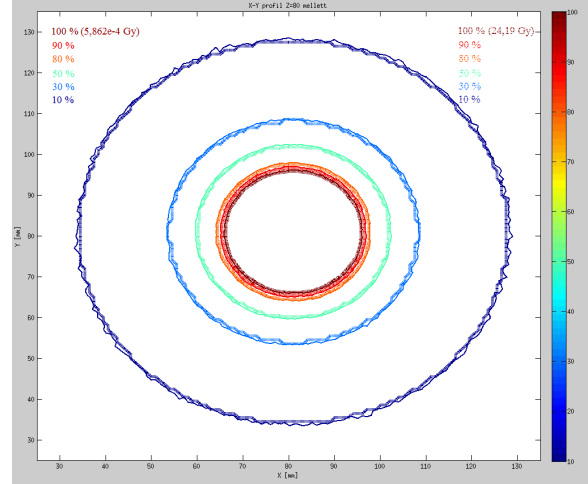


(b) 10^{10} foton.

3.16. ábra. X-Z profil az Y-tengely közepénél, mHDR-v2 forrás szimulációnál.



(a) 10^9 foton.



(b) 10^{10} foton.

3.17. ábra. X-Y profil az Z-tengely közepénél, mHDR-v2 forrás szimulációnál.

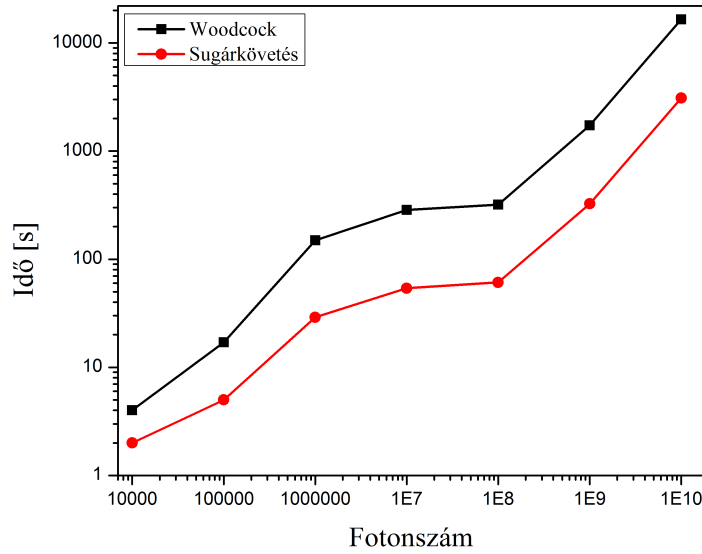
3.3.2. Futási idők összehasonlítása

Az előbb használt forrásmodellel végeztem futtatásokat 10^4 és 10^{10} közötti indított fotonszámokkal, Woodcock módszerrel és sugárkövetéssel, hogy kiderüljön, melyik a gyorsabb. A másodpercre kerekített eredmények a 3.5. táblázatban és a 3.18. ábrán láthatóak.

Indított fotonok darabszáma	Woodcock	Sugárkövetés
	Futási idő [s]	
10^4	4	2
10^5	17	5
10^6	150	29
10^7	285	54
10^8	319	61
10^9	1727	326
10^{10}	16541	3092

3.5. táblázat. Futási idők kiterjedt forrásban.

A program optimalizálásának köszönhetően a Woodcock módszerrel történő szimulációk időtartama lecsökkent ugyan, de a vizsgált elrendezésben meg sem közelíti a sugárkövetés eredményeit. Továbbra is az látszik, hogy a Woodcock módszer legkevésbé sem ideális, ha nagy hatáskeresztmetszet különbségek vannak a rendszerben.



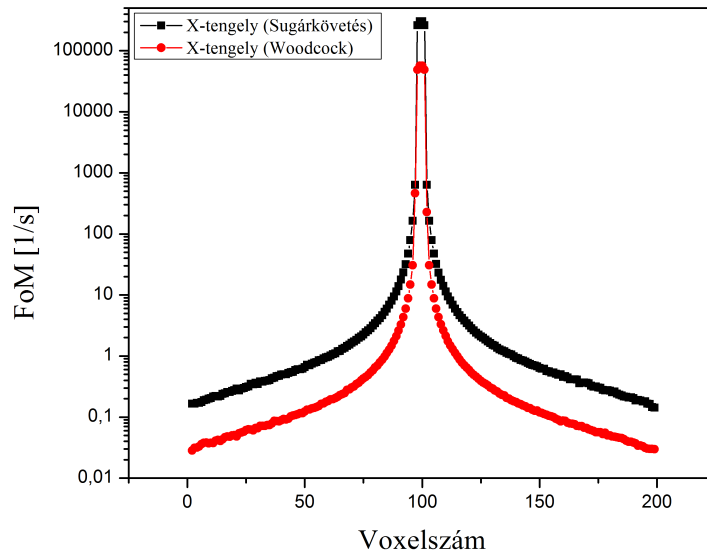
3.18. ábra. Futási idők kiterjedt forrásban.

3.3.3. Figure of Merit kiterjedt forrásnál

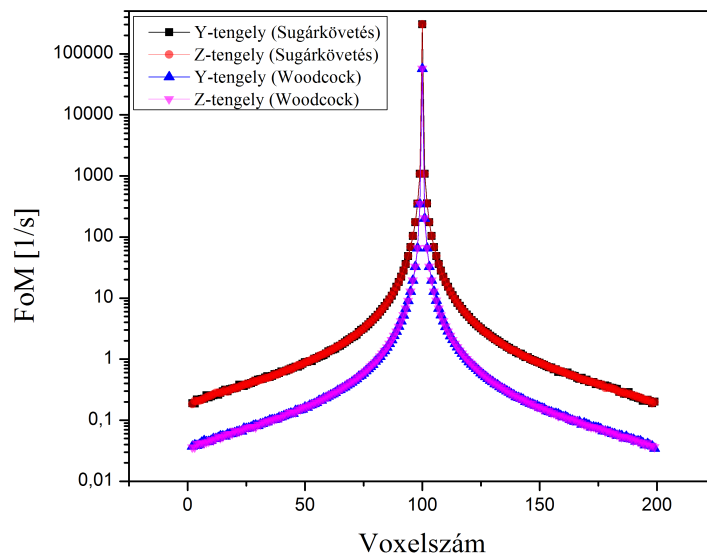
Az összehasonlítást a 10^{10} indított fotonos szimulációkon végeztem. A korábbiakhoz hasonlóan ábrázoltam az X-középtengely mentén kapott 'FoM' értékeket, de ez esetben az Y- és Z-középtengely menti értékeket is megjelenítettem. Utóbbi kettő egy grafikonon szerepel, mivel a rendszer hengersizmetrikus. A kapott adatok a 3.19. és a 3.20. ábrákon láthatóak.

Mindkét ábrán az látható, hogy a sugárkövetés esetén magasabbak a 'FoM' értékek, így amikor a 'FoM' értékek hányadosát ábrázoltam, akkor a sugárkövetéssel kapott értékeket osztottam ez a Woodcock módszerrel kapottakkal. Ezek láthatók a 3.21. és a 3.22. ábrákon.

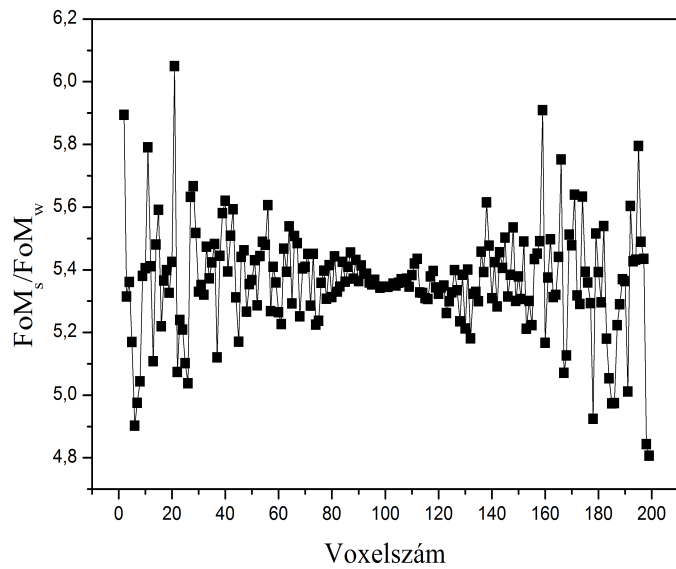
Az eredmények teljesen egyértelműek, a vizsgált elrendezésben a sugárkövetés nem csak gyorsabb, de nagyobb 'FoM' értékeket is produkál. Ez azt jelenti, hogy az új módszerrel hamarabb lehet elérni a kívánt pontosságú eredményeket olyan rendszerben, ahol az irídiumhoz hasonlóan nagy hatáskeresztmetszetű forrás, vagy éppenséggel forrásborítás van jelen.



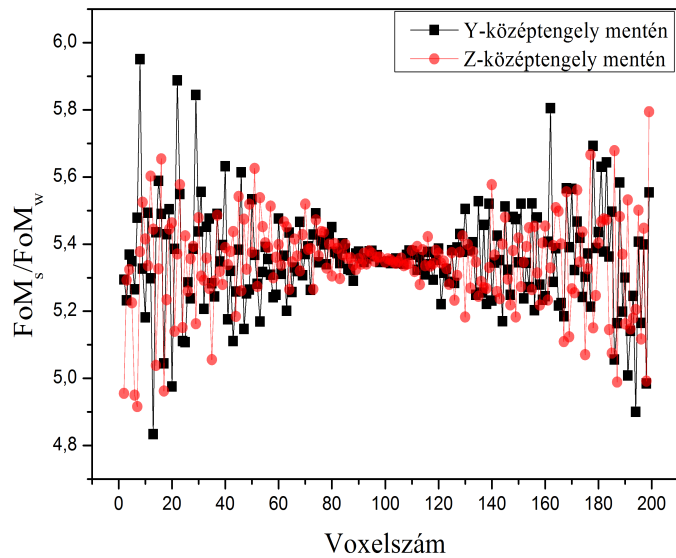
3.19. ábra. Figure of Merit a voxelszám függvényében, kiterjedt forrásnál, az X-középtengely mentén.



3.20. ábra. Figure of Merit a voxelszám függvényében, kiterjedt forrásnál, az Y- és a Z-középtengely mentén.



3.21. ábra. A Figure of Merit értékek hányadosa kiterjedt forrás szimulációja esetén, az X-középtengely mentén.



3.22. ábra. A Figure of Merit értékek hányadosa kiterjedt forrás szimulációja esetén, az Y- és Z-középtengely mentén.

Összefoglalás és kitekintés

A TDK munkám keretén belül:

- Sikerült kiküszöbölöm a szakdolgozatom során létrehozott programban jelentkező energiátárolási problémát, és egyúttal növelni a számolás pontosságát.
- A kód optimalizálásával 40%-kal csökkentettem a futási időt.
- Implementáltam a sugárkövetést, mint új szabad úthossz sorsoló eljárást.
- Sikeresen validáltam az új módszer eredményeit pont- és kiterjedt forrásokra.
- Összehasonlítottam a sugárkövetést és a Woodcock módszert.

Az összehasonlítás a várt eredményeket hozta; a sugárkövetés nem tudta felvenni a versenyt a Woodcock módszerrel olyan elrendezésekben, ahol nem volt kiugróan magas hatáskeresztmetszetű anyag a rendszerben. Mind a homogén vízfantomnál, mind a csontot és tüdőt tartalmazó elrendezésben nagyobb Figure of Merit értékeket, és alacsonyabb futási időket tudott elérni a Woodcock módszer. Az alkalmazás szempontjából legfontosabb szimulációban viszont a új módszer beváltotta a hozzáfűzött reményeket, és jobb eredményeket produkált, mint a régi metódus. Mivel a Brachyterápia során mindig lesz a rendszerben nagy hatáskeresztmetszetű forrásanyag vagy borítás, egyértelmű, hogy a Woodcock módszer jelen formájában nem alkalmazható a tervezőrendszerben.

Nem szabad viszont ilyen könnyedén elvetni ezt az eljárást, hiszen ha a foton már kijutott a forrásból, akkor az emberi test szöveteiben talán a legjobb megoldás lenne a szabad úthossz sorsolására. Ha feltesszük, hogy ha egy, a forrásból kilépő foton visszatér a forrásba, akkor újra kilépni már nem tud, akkor fel lehetne osztani a vizsgált rendszert különböző térrészekre, melyekben különböző majoráns hatáskeresztmetszetek lehetnek. Így megtarthatnánk a módszert az előnyös tulajdonságaival együtt, és kiküszöbölnénk

a gyenge pontját. A feltevés nem is tűnik túl merésznek, hiszen a forrásból másodjára kilépő foton valószínűleg elhanyagolható járulékot adna a forrás közelében, ahol a legtöbb foton megfordul.

Ami a sugárkövetést illeti, nem lehetünk maradéktalanul elégedettek vele, hiszen bár a Woodcock módszernél jobban működik Brachyterápiához hasonló környezetben, de a program futási ideje még mindig túl hosszú ahhoz, hogy besugárzás-tervezéshez lehessen használni.

Azonban ezt az eljárást többféleképpen is tovább lehet fejleszteni. Feltehetően a jelenlegi kód további optimalizálásával is lehet időt nyerni, hiszen a program szinte a teljes futási időt ennek az eljárásnak a végzésével tölti. Emellett viszont olyan módszerekkel lehet kiegészíteni a kódot, mint a 'track length estimator', vagy a 'next event estimator'. Ezek a módszerek kihasználnák azt a számítást, amit a voxeleken keresztül való léptetéssel végzünk, és egy szabad úthosszon belül is adna le energiát a foton. Első körben azért nem alkalmaztam ilyen eljárást, mert féltő, hogy egyszerre sok foton próbálna ugyanazokba a memóriacímekbe adatot írni, és sok idő menne el azzal, hogy a 2.1-es fejezetben említett 'atomiAdd' függvény arra vár, hogy a többi foton befejezze a leadott energia rögzítését.

A következő feladat tehát a módszerek továbbfejlesztése és kipróbálása, hogy a klinikumban is használható sebességet érjen el a program. Ezután kezdődhet majd meg egy, a forrás- és anyageloszlás rugalmasabb kezelését engedélyező program megvalósítása, vagy éppen a grafikus kezelői felület létrehozása.

Az eddigi eredmények nagyon ígéretesek, bízom benne, hogy sikerül megtalálni a megfelelő módszereket, és végül orvosi alkalmazásba állítani a programot. Mindenképpen szeretnék tovább dolgozni a fejlesztésen, mivel továbbra is nagy lehetőséget látok benne, és szakmai szempontból is rendkívül hasznosnak találom.

Irodalomjegyzék

- [1] Á. Cserkaszky, Á. Szlávecz, D. Légrády, G. Tolnai: *Forward Monte Carlo Based image reconstruction software for Positron Emission Tomography*, corresponding to code Revision 664 General description, 2013. november 11.
- [2] Klausz Milán: *Brachyterápiás célú GPU alapú Monte Carlo szoftver programozása*, Bsc. Szakdolgozat, 2014
- [3] <http://www.nist.gov/pml/data/xcom/>,
letöltés ideje: 2014. május 18., 21:41
- [4] <http://www.mathworks.com/matlabcentral/fileexchange/5735-plot3c>,
letöltés ideje: 2014. május 24., 02:57
- [5] NVIDIA, Cuda Programming Guide, Version 2.3.1, 2009. augusztus 26.
- [6] F. B. Brown, J.S. Bull, J.T. Goorley, A. Sood, J.E. Sweezy: *MCNP5-1.51 Release Notes*, LA-UR-09-00384, 2009.
- [7] <http://physics.nist.gov/cgi-bin/Star/compos.pl?refer=ap&matno=119>,
letöltés ideje: 2014. október 25., 16:37
- [8] Perez-Calatayud J., Ballester F., Das R.K., DeWerd L.A., Iddott G.S., Meigooni A.S., Ouhib Z., Rivard M.J., Sloboda R.S., Williamson J.F.: Dose calculation for photon-emitting brachytherapy sources with average energy higher than 50 keV: report of the AAPM and ESTRO, *Med Phys*, 2012; 39(5):2904–29.