Navigálás a lebegőpontos tengereken: a bitenkénti reprodukálhatóságtól a csökkentett pontosságú számításokig

Siklósi Bálint

A PhD Disszertáció tézisei

Témavezető:

Dr. Reguly István Zoltán, PhD

Pázmány Péter Katolikus Egyetem

Roska Tamás Műszaki és Természettudományi Doktori Iskola

Budapest, 2025

1. Bevezetés

A lebegőpontos aritmetika a modern tudományos számítások egyik alappillére. Az időjárásmodellezéstől a mérnöki szimulációkig lehetővé teszi a bonyolult, folytonos valós rendszerek digitális közelítését. Azonban a lebegőpontos műveletek természetüknél fogva korlátozott pontosságúak, kerekítési hibákat tartalmaznak, és nem asszociatívak. Ezek a korlátok két fő kihívásként jelennek meg a nagy teljesítményű számításokban: az eredmények reprodukálhatóságának hiánya, valamint a numerikus pontosság és a teljesítményhatékonyság közötti feszültség.

Ez a disszertáció mindkét problémát két összefüggő témán keresztül vizsgálja. Az első a bitpontos reprodukálhatóság – az a garancia, hogy a számítások minden futtatás, a legtöbb hardverplatform vagy párhuzamosítási szint esetén bitről bitre azonos eredményt adnak. A reprodukálhatóság különösen fontos a tudományos kontextusokban, ahol az eredményeknek megbízhatónak, ellenőrizhetőnek és hordozhatónak kell lenniük. Mégis, ezt gyakran aláássa a párhuzamos számítás, ahol a lebegőpontos műveletek sorrendje kiszámíthatatlan. A második téma a csökkentett és vegyes pontosságú számítás, amely nagyobb teljesítményt és energiahatékonyságot ígér az alacsonyabb pontosságú reprezentációk szelektív alkalmazásával. Azonban ez a teljesítmény növekedés a numerikus pontatlanság elfogadhatatlan mértékű növekedésének kockázatával járhat.

A kutatás célja e két kihívás vizsgálata nagyléptékű számítógépes szimulációk kontextusában. A disszertáció általánosítható megoldásokat mutat be, amelyeket olyan domén-specifikus keretrendszerekben valósítottam meg, mint az OP2 és az OPS, ezek a reprodukálhatóság és a pontosság közötti kompromisszumokat úgy kezelik, hogy közben nem áldozzák fel a skálázhatóságot vagy a produktivitást. A módszereket referenciaproblémákon és ipari szintű szimulációs megoldókon teszteltem, beleértve a Rolls-Royce Hydra CFD alkalmazást és az OpenSBLI könyvtárat, ezáltal széles körű és gyakorlati értékelési környezetet biztosítva.

2. Módszerek és eszközök

2.1. Lebegőpontos aritmetika és pontossági kihívások

A lebegőpontos aritmetika alapvető szerepet játszik a tudományos számításokban, mivel lehetővé teszi a valós számok digitális ábrázolását és kezelését. Egy lebegőpontos szám tipikusan az alábbi formában reprezentálható:

$$(-1)^{elojel} \times mantissza \times alap^{karakterisztika}$$

ahol az *előjel* bit a szám előjelét jelzi, a *mantissza* határozza meg a szám pontosságát, az *alap* általában 2, míg a *karakterisztika* (vagy exponens) a nagyságrendet méretezi. Az IEEE 754 szabvány [1] meghatározza a leggyakrabban használt formátumokat, például a fél pontosságú (16 bites), egyszeres pontosságú (32 bites) és dupla pontosságú (64 bites) lebegőpontos számokat, egyensúlyban tartva a dinamikus tartományt és a pontosságot.

A lebegőpontos aritmetika ugyanakkor természeténél fogva korlátokat is bevezet, különösen a kerekítési hibákat és a műveletek nem-asszociativitását [2, 3]. Mivel az eredményeket a formátumhoz kell kerekíteni, a műveletek sorrendje befolyásolja a végső eredményt, ami változékonyságot okozhat párhuzamos számítások során, ahol a műveletek sorrendje nem determinisztikus. Az Aero benchmark (OP2) esetében például ugyanannak a konjugált gradiens megoldónak az eltérő számú MPI folyamattal történő futtatása kis, de mérhető eltéréseket eredményez az eredményekben a kerekítések és a nem-asszociativitás miatt [4].

Ez a nem-determinizmus megnehezíti a reprodukálhatóságot, amely viszont alapvető fontosságú a hibakeresés, az ellenőrzés és a tudományos érvényesítés szempontjából. A bitpontos reprodukálhatóság – amely garantálja, hogy az eredmények azonosak maradnak minden futtatás során, függetlenül a párhuzamosítás mértékétől – megköveteli a kerekítési változékonyság forrásainak kontrollálását vagy kiküszöbölését, különösen az

összegzési műveletek során. Olyan megközelítések, mint a ReproBLAS [5], reprodukálható összeadásokat valósítanak meg oly módon, hogy a lebegőpontos összeadásokat gondosan kezelik, így garantálva a bitpontos azonosságot a végrehajtási sorrendtől függetlenül.

A tudományos alkalmazások egyre növekvő számítási igényei motiválták a csökkentett és vegyes pontosságú lebegőpontos aritmetika kutatását a teljesítmény, a memóriahasználat és az energiahatékonyság javítása érdekében.

A csökkentett pontosságú számítás kisebb bitszámú lebegőpontos formátumokat alkalmaz (például 16 bites fél pontosságú vagy akár 8 bites formátumokat) a számok reprezentálására, ami jelentősen csökkentheti a memóriaigényt, és növelheti az áteresztőképességet az alacsony pontosságú műveletekre optimalizált hardvereken. Ugyanakkor a pontosság csökkentése megnöveli a kerekítési hibákat, és szűkíti az ábrázolható értéktartományt, ami hátrányosan befolyásolhatja a numerikus pontosságot és stabilitást.

A vegyes pontosságú aritmetika különböző pontosságú lebegőpontos formátumokat kombinál egyetlen számítás során, hogy egyensúlyt teremtsen a pontosság és a hatékonyság között. Például az időközi számításokat alacsonyabb pontossággal végezhetjük, míg a kritikus összegzéseket vagy korrekciókat magasabb pontossággal. Az iteratív algoritmusok, például a gradienscsökkenés vagy más ismétlődő megoldók, különösen alkalmasak a vegyes pontosság alkalmazására, mivel a kezdeti durva közelítések fokozatosan finomíthatók magasabb pontosságú lépésekkel.

Ez a megközelítés kihasználja a modern nagy teljesítményű számítógépes rendszerek – többek között GPU-k és mesterséges intelligencia gyorsítók – hardveres képességeit, amelyek gyakran lényegesen nagyobb áteresztőképességet biztosítanak az alacsony pontosságú műveletekhez. A vegyes pontosság gondos alkalmazásával jelentős gyorsulás és energiafogyasztás-csökkenés érhető el anélkül, hogy az összesített megoldásminőség romlana.

Ebben a disszertációban a csökkentett és vegyes pontosságú számítás

lehetőségeit ezen kompromisszumok mentén vizsgáljuk: hogyan érhető el számítási hatékonyság növekedés a numerikus pontosság és reprodukálhatóság elfogadható szinten tartása mellett.

2.2. OP2 keretrendszer

Az OP2 egy domén-specifikus absztrakciós keretrendszer párhuzamos, rendezetlen térhálón végzett számításokhoz [6]. Lehetővé teszi a fejlesztők számára, hogy a számításokat magas szintű, platformfüggetlen módon írják le térháló halmazokon és leképezéseken. Az OP2 futtatókörnyezet automatikusan optimalizált párhuzamos kódot generál különböző architektúrákra, beleértve a többmagos CPU-kat és GPU-kat is.

Az OP2 támogatja az elosztott memóriás párhuzamosságot MPI-n keresztül, valamint az osztott memóriás párhuzamosságot OpenMP vagy CUDA segítségével, ezáltal biztosítva a skálázható teljesítményt heterogén HPC rendszereken. Az OP2-t sikeresen alkalmazták tudományos alkalmazásokban, például áramlástanban és végeselem-módszerekben [4].

2.3. OPS és OpenSBLI

Az OPS (Oxford Parallel library for Structured mesh solvers) hasonló absztrakciós keretrendszer, amely rendezett térhálós stencil-alapú számításokat céloz meg [7]. Magas szintű API-t biztosít, és többféle párhuzamos backendet támogat (MPI, OpenMP, CUDA), lehetővé téve a hordozható és hatékony végrehajtást változatos HPC platformokon.

Az OpenSBLI az OPS-re épül, és automatizálja a magas rendű véges differenciás megoldók generálását összenyomható áramlási problémákhoz [8]. Szimbolikusan leírt problémaspecifikációkat fordít optimalizált, párhuzamos OPS-alapú kóddá, elősegítve a numerikus módszerekkel való gyors fejlesztést és kísérletezést.

Az OP2, OPS és OpenSBLI együtt a tudományos számítások modern megközelítéseit képviselik, amelyek az absztrakcióra, automatizálásra és

teljesítményhordozhatóságra helyezik a hangsúlyt, kezelve mind a rendezetlen, mind a rendezett térhálós problémákat. Magas szintű API-juk lehetővé teszi, hogy a kutatók a numerikus algoritmusokra koncentráljanak anélkül, hogy alacsony szintű kódot kellene módosítaniuk az eltérő architektúrákhoz. Ez a felelősségi körök szétválasztása támogatja a kódgenerálás és optimalizálás automatizálását, biztosítva, hogy az alkalmazások fenntarthatóak és hatékonyak maradjanak a folyamatosan fejlődő HPC platformokon.

3. Új tudományos eredmények

Ebben a fejezetben a disszertáció fő tudományos hozzájárulásait tézisek formájában mutatom be. Minden téziscsoport egy-egy kutatási területet emel ki, míg az egyes tézispontok konkrét eredményeket írnak le.

Téziscsoport I. – Algoritmusok reprodukálható lebegőpontos műveletekhez nem strukturált térhálókon értelmezett számításokon.

Tézis I.1. Az OP2 DSL absztrakciójából kiindulva megmutattam, hogy mely lebegőpontos műveletek – például a párhuzamos redukciók, az indirekt memóriaelérések, valamint a nemdeterminisztikus végrehajtási sorrend – felelnek a reprodukálható tulajdonság lehetséges megsértéséért. Megmutattam, hogy mely lépésekkel – például ideiglenes tömbalapú akkumulációval, determinisztikus gráfszínezéssel és a ReproBLAS használatával – lehet biztosítani, hogy ezek a műveletek mégis reprodukálható eredményt adjanak.

A kutatás ezen részében az OP2 DSL nyelven írt rendezetlen térhálós alkalmazásokra jellemző számítási mintákat elemeztem. Ezek a minták tipikusan indirekt memóriaműveleteket és akkumuláló műveleteket tar-

talmaznak – különösen redukciókat és olvasás-írás típusú műveleteket –, amelyek különösen érzékenyek a párhuzamos környezetekben fellépő nem-determinizmusra. Rendszeresen azonosítottam az OP2-ben előforduló azon lebegőpontos művelettípusokat, amelyek bitpontos reprodukálhatósági hibákhoz vezethetnek, különös tekintettel azokra a redukciókra (pl. OP_INC és OP_RW), amelyek megosztott adatokkal dolgoznak.

A problémák megoldására két fő algoritmikus megközelítést javasoltam. Először egy ideiglenes tömbön alapuló felhalmozási sémát vezettem be, amely biztosítja, hogy a növekmények globálisan egyedi elemazonosítók alapján, fix és determinisztikus sorrendben kerüljenek alkalmazásra.

Másodszor, a globális redukciók reprodukálhatóságát az OP2-be integrált ReproBLAS könyvtár segítségével biztosítottam, amely determinisztikus lebegőpontos összegzést tesz lehetővé a szálak számától, a folyamatok elosztásától vagy a redukciós fa szerkezetétől függetlenül. Ezeket a módszereket úgy valósítottam meg, hogy az OP2 absztrakcióját csak minimális mértékben kellett módosítani, így a DSL nyújtotta termelékenységi előnyök megmaradtak, miközben a determinisztikus kimenet biztosíthatóvá vált.

A módszerek hatékonyságát olyan tesztesetekben mutattam be, mint az Aero és az MG-CFD, ahol az eredmények nem-determinisztikus viselkedése láthatóan csökkent vagy teljesen megszűnt különböző folyamatelosztások mellett.

Tézis I.2. Bevezettem egy új algoritmust, amely elosztott, partícionált gráfokon biztosít reprodukálható színezést, függetlenül a partíciók számától. Az algoritmus M. Osama eredetileg GPU-kra kidolgozott gráfszínezési módszerén alapul, és azt terjeszti ki teljes determinisztikusságot biztosítva az elosztott környezetre. Megmutattam, hogy ezek az algoritmusok – az ideiglenes tömbalapú akkumuláció, a determinisztikus gráfszínezés és a ReproBLAS-alapú redukciók – hatékonyan leképezhetők különböző párhuzamosítási stratégiákra. Demonstráltam, hogy a módszerek közel optimális teljesítményt nyújtanak

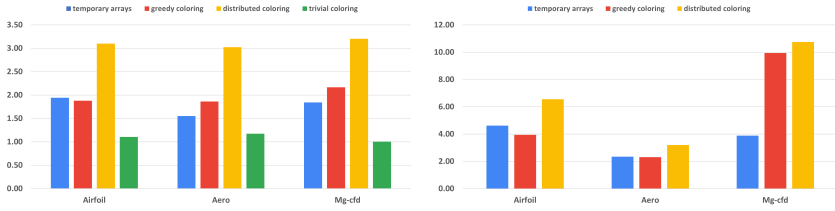
többmagos processzorokon, elosztott rendszereken és GPU-kon, valamint iparilag reprezentatív alkalmazásokon keresztül a gyakorlati skálázhatóságot is igazoltam.

A gráf színezés egy széles körben alkalmazott technika a párhuzamos számítástechnikában, amely megakadályozza a versenyhelyzeteket a megosztott adatok frissítésekor. A hagyományos színezési módszerek azonban gyakran függnek a térháló partícionálási stratégiájától, ami variabilitást eredményezhet, ha a partíciók száma vagy konfigurációja változik. Ez a tézispont egy új elosztott színezési algoritmus fejlesztését írja le, amely garantálja a reprodukálhatóságot a térháló partícionálásától függetlenül.

Az algoritmus egy meglévő párhuzamos színezési módszert bővít ki egy determinisztikus sorrenddel, amely globális elemazonosítók és hash-alapú döntéssegítők alkalmazásán alapul. Ez a tervezés biztosítja, hogy az eredményül kapott színezési hozzárendelések konzisztensek maradjanak, még akkor is, ha a térhálót más számú MPI folyamatra osztják, vagy ha a terheléselosztás miatt újrarendezik.

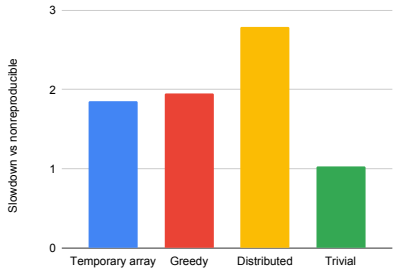
A színezés reprodukálhatósága több OP2 alkalmazásban, köztük a Hydra CFD kódban is ellenőrizve lett, ahol a párhuzamos végrehajtásban a konzisztencia fenntartása kulcsfontosságú a hibakeresés és a validálás szempontjából. Továbbá, az algoritmus általános, és más gráf alapú alkalmazásokban is újrahasznosítható az OP2-n kívül, így robusztus eszközt biztosít a determinisztikus párhuzamos végrehajtáshoz nem struktúrált doménekben.

A reprodukálhatóság érvényesítésének egyik központi kihívása a determinisztikus végrehajtás által bevezetett teljesítménybeli többlet minimalizálása. Ennek érdekében a disszertációban kifejlesztett módszerek úgy lettek megtervezve, hogy hatékonyan integrálódjanak az OP2 kódgenerálásába és párhuzamos backendjeibe. Ez a tézispont továbbá a reprodukálhatósági technikák teljesítményének értékelését és skálázhatósági elemzését részletezi különböző architektúrák alkalmazása során, beleértve a megosztott memóriás CPU-kat, elosztott memóriás klaszte-



(a) (b)

1. ábra. A különböző módszerek lassulásának hatása a nem reprodukálható verzióhoz képest. (**a**) 40 MPI-s folyamat használata a Cirrus gépen; (**b**) Egy MPI+CUDA GPU folyamat használata a Cirrus-GPU gépen.



2. ábra. A Hydra lassulása, amelyet 8M hálón, 20 iteráción, a Cirrus-CPU gépen mértem.

reket és CUDA-támogatott GPU-kat.

A reprodukálható redukciók, ideiglenes tömbös felhalmozási módszerek és színezési séma hatásait több OP2 alkalmazáson mértem. Az 1. és a 2. ábrák a nem reproduktálható alapváltozathoz képesti lassulásokat mutatják. Különösen a Hydra alkalmazást [9] - egy ipari szintű CFD megoldót - használtam a reprodukálhatósági infrastruktúra skálán történő tesztelésére. Az eredmények azt mutatják, hogy bár bizonyos többlet elkerülhetetlen, sok esetben az még mindig elfogadható határokon belül marad, különösen, ha figyelembe vesszük a determinisztikus szimulációs kimenetek előnyeit.

GPU rendszereken, ahol a versenyhelyzetek különösen nehezen kezelhetők, a javasolt technikák sikeresen alkalmazásra kerültek az OP2 CUDA backendjének segítségével. A reprodukálható végrehajtást Nvidia V100 gyorsítókon értem el anélkül, hogy jelentős átalakításokat kellett volna végezni a felhasználói kódban. Ez bemutatja a javasolt algoritmusok gyakorlati megvalósíthatóságát és hordozhatóságát több valós számítási környezetben.

A téziscsoporthoz kapcsolódó publikációk: [J1], [C1], [C2], [C3], [C4].

Tézis II.

Olyan módszertant vezettem be, amely strukturált rácsos alkalmazásokban lehetővé teszi a numerikus pontosság szisztematikus csökkentését, és annak teljesítményre, illetve numerikus pontosságra gyakorolt hatásának mérését. Ezt egy reprezentatív turbulens szimuláción (Taylor–Green örvény) alkalmazva megmutattam, hogy a 32 bites lebegőpontos számábrázolás teljesítménynövekedést eredményez, miközben a pontosság továbbra is elfogadható marad, míg a 16 bites ábrázolás már jelentősen módosítja a fizikai következtetéseket. Erre alapozva az OpenSBLI keretrendszert úgy bővítettem, hogy támogassa a kevert pontosságú számításokat: lehetővé tettem, hogy meghatározott állapotváltozók és átmeneti tárolók alacsonyabb pontosságon legyenek számolva és tárolva. Ez lehetőséget ad a kevert pontosságú konfigurációk kontrollált, kvantitatív kiértékelésére. Megmutattam, hogy gondosan megválasztott 16 és 32 bites ábrázolások kombinációjával az eredmények pontossága megőrizhető, miközben modern CPU és GPU architektúrákon is számottevő futásidő-csökkenés érhető el.

A disszertáció ezen része a csökkentett pontosságú számítások tudományos szimulációkban történő alkalmazásának felmerülő lehetőségét tárgyalja, amelyet nagymértékben a hardverfejlesztések indítanak el. A modern GPU- és CPU-platformok egyre növekvő támogatása a 16- és 32

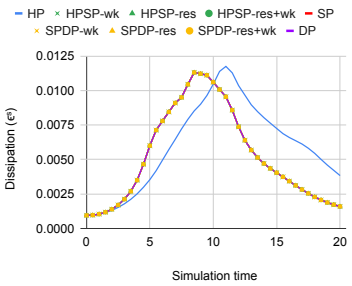
bites lebegőpontos műveletekhez motiválta azt a módszertant, amelyet kifejlesztettem annak rendszeres értékelésére, hogy miként befolyásolják az ilyen pontosságcsökkentések a szimuláció pontosságát és futási teljesítményét.

A módszertant egy kanonikus áramlástani problémára alkalmaztam: a Taylor-Green örvényre, egy 3D-s, időben változó turbulens tesztreferenciára [10]. Ez a probléma jól alkalmas érzékenységi elemzésre, mivel bonyolult, nemlineáris dinamikát mutat, amely fokozhatja a numerikus hibákat, és kiemelheti az alacsonyabb pontosságú beállítások instabilitását.

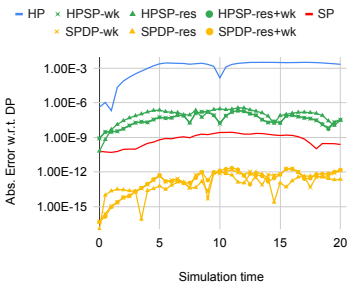
Az OpenSBLI-t használtam kódgenerálásként és az OPS-t a párhuzamos végrehajtás hátttereként, és három különböző pontosságú változatot valósítottam meg: dupla pontosságú (FP64), egyes pontosságú (FP32) és feles pontosságú (FP16) verziókat. A pontosságot globális mennyiségek, például a kinetikus energia disszipációja és a numerikus hiba normái alapján értékeltem, míg a teljesítményt futási idő per iteráció és memóriahasználat formájában mértem.

Ahogy az a 3. ábrán és az 1.–3. táblázatokban látható, az eredmények megerősítik, hogy az FP32 pontosság jó egyensúlyt biztosít, jelentős sebességnövekedést (különösen GPU platformokon) elérve, miközben megőrzi a fizikai szempontból pontos viselkedést. Azonban a teljes FP16 pontossággal futtatott szimulációk nagymértékű eltéréseket mutattak kulcsfontosságú mennyiségekben, ami módosított vagy fizikailag nem helyes eredményeket eredményezett, különösen a csúcs disszipáció közelében. Ez megerősíti azt a következtetést, hogy bár a 16 bites aritmetika bizonyos helyi környezetekben alkalmazható, nem alkalmazható vakon egy teljes CFD kódra anélkül, hogy veszélyeztetné a tudományos érvényességet.

Fontos megjegyezni, hogy a fejlesztett infrastruktúra nem specifikus a Taylor-Green örvényre. Az OpenSBLI szimbolikus és backend-agnosztikus tervezése révén ugyanaz a módszertan újrahasználható más CFD modellek elemzésére is, amelyek véges különbségek diszkretizációt



(a) Disszipáció



(b) A disszipáció abszolút különbsége, összehasonlítva a dupla pontossággal

3. ábra. A TGsym alkalmazás numerikus pontossága különböző pontosságú szinteken. A téháló mérete $= 256^3$, $M = 0.5$, $Re = 800$. A szimulációkat 8000 iteráción keresztül futtatták az alapértelmezett módszerrel.

alkalmaznak struktúrált térhálókon. Így ez a munka megalapozza a jövőbeli kutatásokat, amelyek a pontosságtudatos modellezési gyakorlatokat célozzák meg a nagy teljesítményű szimulációs munkafolyamatokban.

A teljes csökkentett pontosságú szimulációk eredményeire alapozva, a tézis folytatása egy finomabb és rugalmasabb megközelítést mutat be: a kevert pontosságú számításokat. Ahelyett, hogy minden változóra és műveletre ugyanazt a numerikus formát alkalmaznánk, az itt alkalmazott stratégia az volt, hogy a pontosság szintjét szelektíven rendeljük hozzá, figyelembe véve az egyes komponensek numerikus szerepét és stabilitási érzékenységét.

Ennek lehetővé tételéhez kiterjesztettem az OpenSBLI kódgenerálási infrastruktúráját, hogy támogassa a változó-specifikus pontosságvezérlést. Ehhez módosítani kellett a szimbolikus frontendet, a változó típusú deklarációkat, valamint az OPS kernel kódot, biztosítva, hogy az ideiglenes tömbök, munkabufferek és konzervált állapotváltozók mind eltérő lebegőpontos pontossággal legyenek definiálva. A megvalósítás támogatja az olyan kombinációkat, mint az FP64 a konzervált változók

	Alapértelmezett		Storesome	
	futási idő	sebességnövekedés	futási idő	sebességnövekedés
HP	42.43 ms	2.43 ×	18.67 ms	2.69 ×
HPSP	47.71 ms	2.16 ×	22.13 ms	2.27 ×
SP	56.95 ms	1.81 ×	25.00 ms	2.01 ×
SPDP	74.02 ms	1.39 ×	37.60 ms	1.34 ×
DP	103.21 ms	1.00 ×	50.31 ms	1.00 ×

1. táblázat. Futási idő per iteráció és sebességnövekedés összehasonlítva a dupla pontosságú futási idővel a TGsym alkalmazásnál az alapértelmezett és storesome generálási módszerekkel. Térháló mérete=256^3, $M = 0.5$, Re=800, 800 iteráció. A mérések egyetlen NVIDIA A100-SXM4-40GB GPU-val és egy AMD EPYC™ 7763 (Milan) CPU-val történtek.

számára, és FP32 vagy FP16 az köztes műveletekhez.

A Taylor-Green örvény problémát újra tesztkörnyezetként használva, több kevert pontosságú konfigurációt értékeltem, például FP64 állapotváltozókat FP32 maradványokkal, vagy FP32 állapotot FP16 ideiglenes változókkal. A teljesítmény mérőszámok azt mutatják, hogy ezek a konfigurációk jelentős sebességnövekedéseket kínálnak anélkül, hogy a tiszta FP16 futtatások során tapasztalt szimulációs pontosságot elveszítenénk. A futási idő per iteráció és a memóriahasználat javulásait az 1.–a 3. táblázatok összegzik, amelyek több mint 2×-es sebességnövekedést mutatnak, miközben a kulcsfontosságú fizikai mennyiségekben elhanyagolható hűségvesztést tapasztalunk.

A tézishez kapcsolódó publikációk: [J2], [C5], [C6].

A tézishez kapcsolódó puplikáció, a benyújtás időpontjában még elbírálás alatt: [J3]

4. Hatás és Alkalmazások

A disszertációban kifejlesztett módszertanok két kritikus kihívást céloznak meg a számítástudományban: a numerikus reprodukálhatóságot és a pontosságtudatos számítást. Ezek a hozzájárulások közvetlen alkal-

	Alapértelmezett		Storesome	
	futási idő	sebességnövekedés	futási idő	sebességnövekedés
HP	68.48 ms	5.71 ×	21.39 ms	8.12 ×
HPSP	105.22 ms	3.71 ×	47.31 ms	3.67 ×
SP	185.23 ms	2.11 ×	65.05 ms	2.67 ×
SPDP	249.11 ms	1.57 ×	123.22 ms	1.41 ×
DP	390.71 ms	1.00 ×	173.68 ms	1.00 ×

2. táblázat. Futási idő per iteráció és sebességnövekedés összehasonlítva a dupla pontosságú futással a TGsym alkalmazásnál alapértelmezett és Storesome generálási módszerekkel. Térháló mérete=256^3, $Minf = 0.5$, Re=800, 800 iteráció. A mérések egy Intel Xeon Platinum 8592+ CPU-val történtek.

	Alapértelmezett		Storesome	
	Memória	Nyereség	Memória	nyereség
HP	2.21 GB	4.00 ×	1.09 GB	4.00 ×
HPSP	2.72 GB	3.25 ×	1.60 GB	2.72 ×
SP	4.41 GB	2.00 ×	2.17 GB	2.00 ×
SPDP	5.43 GB	1.63 ×	3.19 GB	1.36 ×
DP	8.83 GB	1.00 ×	4.35 GB	1.00 ×

3. táblázat. A memóriahasználat a TGsym alkalmazásnál és a memória nyereség a dupla pontosságú futással összehasonlítva. Méret=256^3.

mazásokat találnak ipari és akadémiai szimulációs munkafolyamatokban.

Reprodukálhatósági technikák a nem struktúrált hálós alkalmazásokhoz lehetővé teszik a determinisztikus eredményeket ipari CFD kódokban (pl. légi- és energetikai rendszerek), ahol a szabályozási előírások megkövetelik az egységes eredményeket a hardver konfigurációk között. Az OP2 integráció lehetővé teszi a bit-szintű reprodukálhatóságot a párhuzamos teljesítmény feláldozása nélkül, így javítva a hibakeresési és tanúsítási folyamatokat. Különösen figyelemre méltó, hogy az aerosztruktúra-tervezésben használt HYDRA CFD megoldó az OP2-t alkalmazza nagyszabású szimulációkhoz, ahol a determinisztikus viselkedés kulcsfontosságú mind az mérnöki validáció, mind a szabályozói jóváhagyás szempontjából.

Kevert pontosságú stratégiák illeszkednek a modern GPU/gyorsító architektúrákhoz, kihasználva az alacsonyabb pontosságú egységeket a sebességnövelés érdekében, miközben a pontosságtudatos algoritmusokkal fenntartják a pontosságot. Ez gyorsabb tervezési ciklusokat tesz lehetővé az aerodinamikai és égéstermék modellezésben, különösen hasznos a valós idejű döntéstámogatás és a nagy paramétertanulmányok esetén. Egy figyelemre méltó példa az OpenSBLI alkalmazása a JAXA (Japán Űrkutatási Ügynökség) számára végzett nagyfelbontású aerofoil buffeting szimulációk során, ahol a gyártási futások három-négy hétig tartanak 120 Nvidia V100 GPU használatával [11]. Az ilyen esetekben bármilyen teljesítményjavulás – mint amit a kevert pontosságú technikák kínálnak – jelentősen csökkentheti a számítási költségeket.

Ezeknek a fejlesztéseknek az OP2/OPS rendszerekbe való beágyazása biztosítja azok azonnali használhatóságát: a fejlesztők reprodukálhatóságot és pontosság-vezérlést nyernek anélkül, hogy újra kellene írniuk az alkalmazás logikáját. Az infrastruktúra lehetővé teszi a jövőbeli *adaptív pontosságú* megközelítéseket is, ahol a szimulációk dinamikusan állítják be a pontosságot a helyi hibák becslései alapján.

Ezek a gyártásra kész megoldások a következő három alapvető elvet

egyensúlyozzák: konzisztencia, hatékonyság és hozzáférhetőség – megfelelve a tudományos számítások magas teljesítményű igényeinek, miközben csökkentik az alulfinanszírozott kutatócsoportok számára a hozzáférési akadályokat.

Mesterséges intelligencia alapú eszközök használata

A disszertáció megírása során mesterséges intelligencia alapú eszközöket is igénybe vettem. Különösen a ChatGPT (OpenAI), a Perplexity.ai, valamint a GitHub Copilot szolgáltak segítségül technikai leírások megfogalmazásában, a szöveg nyelvi és stilisztikai javításában, valamint a kódfejlesztési munkafolyamatok gyorsításában. A dolgozatban szereplő eredmények, következtetések és tudományos érvek teljes mértékben a saját munkámat tükrözik, az említett eszközöket kizárólag a szöveg érthetőségének és a munkafolyamat hatékonyságának támogatására használtam.

Köszönetnyilvánítás

Szeretném megköszönni a témavezetőmnek, Reguly Istvánnak az elhivatott támogatását ezen az úton – legyen szó kód hibakereséséről, a disszertáció megformálásáról, vagy személyes bátorításáról a nehéz időkben. A türelme és megértése valóban nagy különbséget tett.

Hálás vagyok Gihan R. Mudalige-nak (Warwicki Egyetem) és Neil Sandham-nek (Southamptoni Egyetem), hogy meleg szeretettel fogadtak külföldi tartózkodásom alatt, és kollégaként kezeltek. Köszönetem illeti Pushpender K. Sharma-t és David Lusher-t is a kutatásom fizikai és kódolási aspektusainak értékes hozzájárulásaikért és segítségükért.

Külön köszönet a Robinson családnak, hogy helyet adtak nekem otthonukban Southamptonban – a kedvességetek teret adott arra, hogy

valóban a munkámra koncentrálhassak.

A többi PhD hallgatónak és barátomnak – különösen Daninak, Attilának és az “Ebéd délben?” csapatnak – köszönöm a beszélgetéseket, támogatást és jókedvet. Hálás vagyok a lakótársaimnak, Daninak, Annának, Tominak és Dalmának a napi társaságért.

A családomnak: a csendes, de folyamatos támogatásotok többet jelentett számomra, mint bárhogyan is kifejezhetném. Különösen édesapámra emlékezem, aki ezen az úton elhunyt – büszkeségét magammal hozom.

Mindazoknak, akik velem osztoztak az életben ezen az úton, különösen Ágotának – köszönöm a türelmet és gondoskodást. És mindenkinek, aki kisebb-nagyobb módon támogatott: köszönöm.

Külön köszönet illeti az adminisztratív munkatársakat, különösen Vida Tivadarnét, és a nemzetközi kapcsolatok irodáját a segítségükért. Ezt a munkát az EKÖP, ÚNKP, Erasmus+, OTKA és Zsedrovits Tamás támogatása tette lehetővé, akinek időben felajánlott ösztöndíja segített, hogy folytathassam a kutatásomat, amikor a legnagyobb szükségem volt rá.

Köszönöm mindenkinek.

A szerző publikációi

[J1] B. Siklósi, G. R. Mudalige, and I. Z. Reguly, "Enabling bitwise reproducibility for the unstructured computational motif", Applied Sciences, vol. 14, no. 2, 2024, issn: 2076-3417. doi: 10.3390/app14020639. [Online]. Available: https://www.mdpi.com/2076-3417/14/2/639

[J2] D. J. Lusher, A. Sansica, N. D. Sandham, J. Meng, B. Siklósi, and A. Hashimoto, "Opensbli v3.0: High-fidelity multi-block transonic aerofoil cfd simulations using domain specific languages on gpus", Computer Physics Communications, vol. 307, p. 109406, 2025, issn: 0010-4655. doi: https://doi.org/10.1016/j.cpc.2024.109406 [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0010465524003291.

[J3] B. Siklosi, P. K. Sharma, D. J. Lusher, I. Z. Reguly, and N. D. Sandham, "Reduced and mixed precision turbulent flow simulations using explicit finite difference schemes", Future Generation Computer Systems, 2025, Under review.

[C1] B. Siklósi, I. Z. Reguly, and G. R. Mudalige, "Bitwise reproducible task execution on unstructured mesh applications", in 2020 20th IEEE/ACM International Symposium on Cluster, Cloud and Internet Computing (CCGRID), 2020, pp. 889–892. doi: 10.1109/CCGrid49817.2020.00015.

[C2] B. Siklósi, "Bitwise reproducible execution of unstructured mesh applications", in PhD Proceedings Annual Issues of the Doctoral School, Faculty of Information Technology and Bionics, vol. 16, 2021, pp. 165–168.

[C3] B. Siklósi, "Bitwise reproducible execution of unstructured mesh applications", in PhD Proceedings Annual Issues of the Doctoral School, Faculty of Information Technology and Bionics, vol. 15, 2020, pp. 161–164.

[C4] B. Siklósi, I. Z. Reguly, and G. R. Mudalige, "Bitwise reproducible execution of unstructured mesh applications", Jedlik Laboratories

Reports, vol. 9, no. 2, pp. 13–19, 2020.

[C5] B. Siklósi, "Achieving mixed precision computing with the help of domain specific libraries", in PhD Proceedings Annual Issues of the Doctoral School, Faculty of Information Technology and Bionics, vol. 17, 2022, pp. 187–189.

[C6] B. Siklósi, "Utilizing the op2 domain specific library for adaptive multi-precision computing", in PhD Proceedings Annual Issues of the Doctoral School, Faculty of Information Technology and Bionics, vol. 18, 2023, pp. 141–144.

[O1] B. Siklosi, I. Z. Reguly, and G. R. Mudalige, "Heterogeneous cpu-gpu execution of stencil applications", in 2018 IEEE/ACM International Workshop on Performance, Portability and Productivity in HPC (P3HPC), 2018, pp. 71–80. doi: 10.1109/P3HPC.2018.00010.

[O2] B. Keömley-Horváth, G. Horváth, P. Polcz, et al., "The design and utilisation of pansim, a portable pandemic simulator", in 2022 First Combined International Workshop on Interactive Urgent Supercomputing (CIW-IUS), 2022, pp. 1–9. doi: 10.1109/CIW-IUS56691.2022.00006.

Hivatkozások

[1] „Ieee standard for floating-point arithmetic," *IEEE Std 754-2019 (Revision of IEEE 754-2008)*, pp. 1–84, 2019.

[2] D. Goldberg, „What every computer scientist should know about floating-point arithmetic," *ACM Comput. Surv.*, vol. 23, p. 5–48, 3 1991.

[3] O. Villa, D. Chavarría-Miranda, V. Gurumoorthi, A. Marquez, and S. Krishamoorthy, „Effects of floating-point non-associativity on numerical computations on massively multithreaded systems," in *CUG Proceedings*, 5 2009.

[4] O. Zienkiewicz, R. Taylor, and J. Zhu, *The Finite Element Method: its Basis and Fundamentals (Seventh Edition)*. Oxford: Butterworth-Heinemann, seventh edition ed., 2013.

[5] J. Demmel, P. Ahrens, and H. D. Nguyen, „Efficient reproducible floating point summation and blas," Tech. Rep. UCB/EECS-2016-121, EECS Department, University of California, Berkeley, 06 2016.

[6] G. Mudalige, M. Giles, I. Reguly, C. Bertolli, and P. Kelly, „Op2: An active library framework for solving unstructured mesh-based applications on multi-core and many-core architectures," 2012 Innovative Parallel Computing (InPar), pp. 1–12, 2012.

[7] I. Z. Reguly, G. R. Mudalige, and M. B. Giles, „Loop Tiling in Large-Scale Stencil Codes at Run-Time with OPS," *IEEE Transactions on Parallel and Distributed Systems*, vol. 29, no. 4, pp. 873–886, 2018.

[8] D. J. Lusher, S. P. Jammy, and N. D. Sandham, „OpenSBLI: Automated code-generation for heterogeneous computing architectures applied to compressible fluid dynamics on structured grids," *Computer Physics Communications*, vol. 267, p. 108063, 2021.

[9] I. Z. Reguly, G. R. Mudalige, C. Bertolli, M. B. Giles, A. Betts, P. H. Kelly, and D. Radford, „Acceleration of a full-scale industrial cfd application with op2," *IEEE Transactions on Parallel and Distributed Systems*, vol. 27, no. 5, pp. 1265–1278, 2016.

[10] S. P. Jammy, C. T. Jacobs, and N. D. Sandham, „Performance evaluation of explicit finite difference algorithms with varying amounts of computational and memory intensity," *Journal of Computational Science*, vol. 36, p. 100565, 2019.

[11] D. J. Lusher, A. Sansica, N. D. Sandham, J. Meng, B. Siklósi, and A. Hashimoto, „OpenSBLI v3.0: High-Fidelity Multi-Block Transonic Aerofoil CFD Simulations using Domain Specific Languages on GPUs," *Computer Physics Communications*, p. 109406, 2024.