TémaLabor – GPGPU- általános számítások a GPUn, folyadékszimuláció

Mit is jelent az a GPGPU?

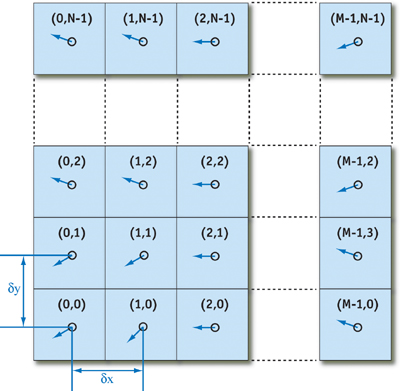
Alapvetően ha programozásról beszélünk mindenkinek elsőre az egyszálú (esetleg pár szálú) CPU-n futó programok jutnak az eszébe. Ez miért is jelenthet problémát? Vannak olyan feladatok, amelyek nagymértékben párhuzamosíthatóak, és ha ezt észre vesszük és ki is használjuk sokkal gyorsabb programokat érhetünk el. Mint tudjuk a CPU csak korlátozottan képes a többszálúságra így az olyan problémák esetén ahol egyszerű számításokat tudunk nagy számban párhuzamosan végezni érdemes a GPU-hoz fordulni. Ilyenkor beszélünk GPGPU alkalmazásról. (GPGPU = General-purpose computing on graphics processing units, azaz általános célú számítások a grafikus processzoron)

Az egyik ilyen GPGPU alkalmazási terület a fizikai szimulációk. Aki ismeretes az áramlástanban hallhatott már a Navier-Stokes egyenletekről. Mi a témalaborunk keretében ezek megértésére és alkalmazására fókuszáltunk.

Ahhoz persze hogy a GPUn ilyen dolgokat végezhessünk, természetesen szükség van a megfelelő eszközökre, gondolok itt mind a megfelelő videokártyákra, ill. a megfelelő driverekre, fejlesztő környezetekre. Alapvetően két különböző környezetet használhatunk: OpenCL, CUDA. Az OpenCL nevének megfelelően is elméletileg mindenféle gyártó kártyáján működik, míg ezzel ellentétben a CUDA csak Nvidia és azon belül csak megfelelő Nvidia kártyákkal működik. Mivel mi rendelkeztünk Nvidia kártyával így az utóbbi környezetet választottuk a fejlesztésre. (Munkák során kétféle kártyát használtunk: 1 egyszerűbb mobil gput: Nvidia GeForce GT 635m és egy jóval erősebb Nvidia GeForce GTX 1070 kártyát)

Fizikai alapok, avagy a Navier-Stokes egyenletek:

A folyadék állapotának leírására jó módszer ha a folyadékot egyfajta „négyzet-rácsban” képzeljük el, amelyben az adott pozícióban fellépő sebességet tároljuk (sebesség-vektormező)

x = (x, y) pozíció

u = (u,v) sebesség

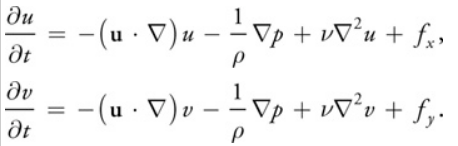
T idő

u(x, t) = (u(x, t), v(x, t))   
A Navier-Stokes egyenleteket Claude Navier és George Gabriel Stokes állították fel 1822-ben a folyékony anyagok mozgásának, áramlásának leírására. Alapvető elképzelésük az volt, hogy az anyagban fellépő feszültségnek két összetevője van: a folyékony anyag sebességgradiensével arányos diffúziós (vagyis egy a viszkozitást jellemző) kifejezés összetevőből és egy nyomás összetevőből áll. Az egyenletek mind elméleti mind gyakorlati haszonnal bírnak hiszen segítségükkel leírható pl. az időjárás, óceánokban áramlatok, repülőgépek szárnyai körül észlelt áramlás, vagy például csillagok galaxisokon belül leírt mozgása is. A repülőgépek és gépjárművek tervezése mellett használhatóak atmoszferikus szennyezés felmérésére, sőt akár véráram szimulálására is.

Az egyenletek:



Ahol *ρ* a folyadék sűrűsége, ν a viszkozitás, és **F** = (fx , fy ) minden egyéb külső erőt ír le. Emellett vegyük észre, hogy az első egyenlet lényegében két egyenletet takar, hiszen az u az egy vektor mennyiség, amit külön két részre lehet bontani:



Az egyenlet tagjai:

Advekció



Gondoljunk bele, hogy a folyadék hogyan szállíthat minden féle objektumot, sűrűséget a folyammal együtt. Például ha belecseppentünk a folyadékba egy kis tintát akkor az áramlatok ezt elszállítják. Ezt „előrehaladő”, „szállító” mozgást írja le az advekciós rész.

Nyomás



Mivel a folyadék molekulái tudnak áramolni mindenfele, egymással ütköznek is. Ha erőt fejtünk ki a folyadékban egy pontra akkor ez az erő nem fog rögtön végig terjedni a folyadékban. Az erőhöz közel lévő molekulák meglökik a távolabb lévőket és így egy nyomás alakul ki. Mivel a nyomás az egységnyi területre eső erő nagyságát írja le így könnyedén észre vehetjük hogy ez gyorsuláshoz vezet. (Newton első törvénye F = m\*a). Ez az egyenleti rész ezt írja le.

Diffuzió



Tudjuk tapasztalatból, hogy a vannak sűrűbb és folyékonyabb anyagok. Azt mondjuk, hogy a sűrű anyagoknak nagy a viszkozitásuk. A viszkozitás egy mérőszám, hogy a folyadék mennyire áll ellen az áramlásoknak. Ez az ellenállás sebességdiffúziót okoz.

Külső erők



Mindenféle egyéb külső erőt ír le, amelyek lehetnek lokálisak vagy globálisak (pl. gravitáció)

Látható, hogy a U2207.GIF szimbólum három különböző alkalmazását is használtuk. Ez az ún. nabla operátor. A három különböző dolog, amire használtuk: a gradiens, a divergencia, és a Laplace-operátor.

|  |  |  |
| --- | --- | --- |
|  | Definíció | Véges differencia alak |
| Gradiens: |  |  |
| Divergencia: |  |  |
| Laplace: |  |  |

Gradiens: parciális deriváltakból álló vektor

Divergencia: a vektormező forrásának (vagy elnyelésének) nagysága egy adott pontban. azaz a sebesség változása a folyadék egy adott pontját határoló felületen.

A második egyenlet (a folytonosság egyenlete: ) biztosítja, hogy a folyadék összenyomhatatlan

Laplace: A divergencia alkalmazása egy vektormező gradiensére. Diffúziós egyenleteknél gyakran megtalálható. U2207.GIF · U2207.GIF = U2207.GIF2. És ezzel meg is kaptuk a harmadik egyenletet.

Az egyenletek megoldása analitikus módon csak ritkán egyszerű esetekben történik. Általában és a mi esetünkben is egy numerikus, inkrementális megoldást alkalmazunk. (főleg hogy mi szeretnék RealTime megjeleníteni is, aminek eredményeképp kevés idő van a számításokra) Az egyenletek megoldása nem volt feladatunk a jelenlegi témalabor keretein belül így megkaptunk egy OpenCL környezetben megírt implementációját a problémának.

Alap implementáció OpenCL ben

Konvertálás OpenCL-ből Cuda-ba.

hasonlóságok,előnyök, hátrányok

optimalizálás/ monitorozás

Új funkciók

tetszőleges testek / falak – boundary Buffer

Állandóan mozgó közeg előállítása

A célunk az volt, hogy egy állandóan mozgó közeget hozzunk létre, amivel így jól szimulálható egy folyó áramlása vagy akár egy szélcsatorna is. Ehhez az első próbálkozásként minden négyzetrácson hozzáadtunk a sebességvektorhoz egy apró értéket, hogy ezzel biztosítsuk az közeg állandó mozgását. Azonban látható az ábrán is hogy nem a kívánt eredményt értünk el, hiszen így hogy minden mezőhöz hozzáadunk a falakba lényegében „belenyomjuk” a folyadékot és így alkalmatlan a különböző testek áramvonalasságának elemzésére:



Ezt követően próbáltuk ki azt a megoldást, hogy csak a képernyő szélén lévő pár oszlophoz adunk hozzá erőt és majd ez az egész közeget mozgásba fogja hozni.



Ez a megoldás sokkal jobbnak minősült a valóság szimulálására.

tetszőleges képből falak előállítása kép megjelenítése

Összefoglalás, jövőbeli tervek, felhasználási lehetőségek.