

Zrýchlenie výpočtu splajnových povrchov pomocou redukovaného algoritmu

Zoltán Szoplák

3Ib, 2017-2018

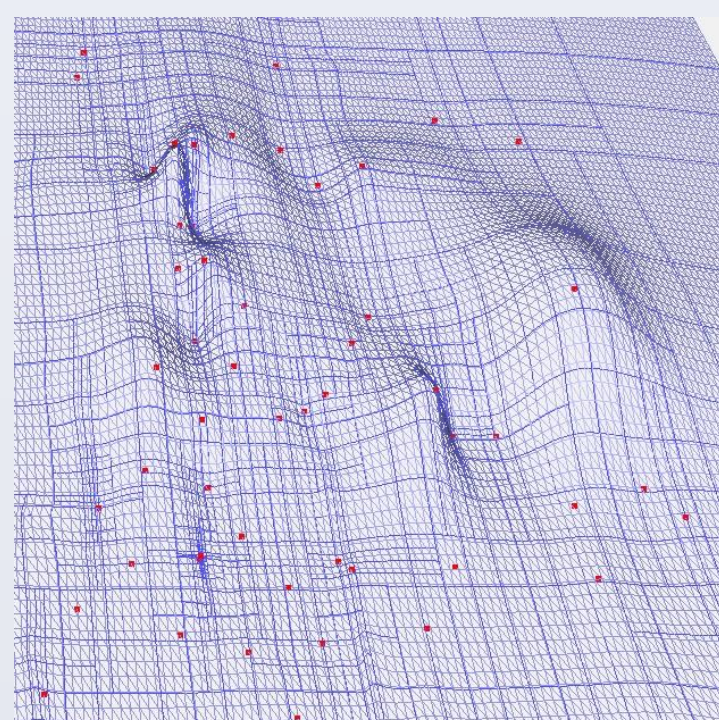
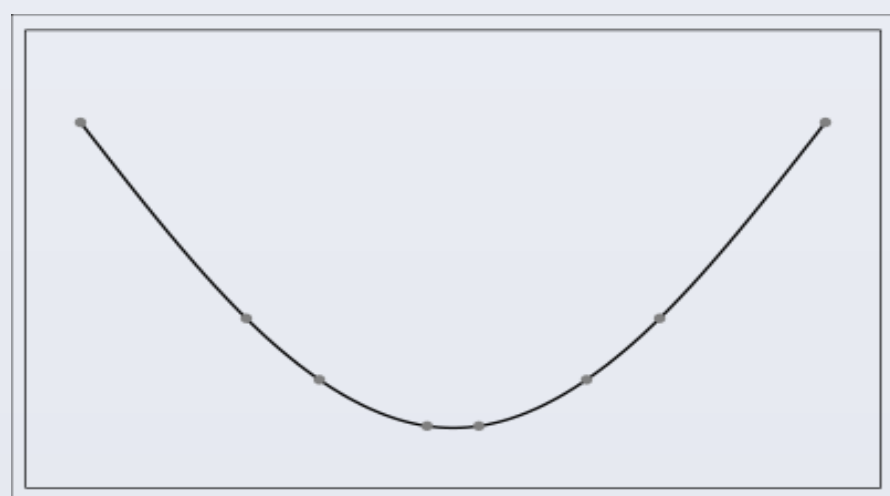
Abstrakt

Splajny a splajnové povrchy sa v informatike mnohokrát používajú na modelovanie či aproximáciu dát a taktiež sú všadeprítomné v počítačovej grafike, hlavne v CAD systémoch. Z tohoto dôvodu sa usilujeme nájsť algoritmus, ktorý by dokázal riešiť trojdiagonálne sústavy rovníc potrebných na ich interpoláciu čo najrýchlejšie. Naše riešenie spočíva vo využití viacerých algoritmov, vrátane nedávno navrhutej metódy, ktorá umožňuje explicitne vypočítať polovicu neznámych koeficientov.

Splajny a splajnové povrchy

Splajny sú špeciálne typy funkcií, ktoré sú definované polynomiálmi jednotlivo po častiach. Môžu byť jeden- alebo viacrozmerné.

V informatike sú často využívané pri modelovaní dát, hlavne v oblasti počítačovej grafiky ako reprezentácia kriviek kvôli jednoduchosti ich konštrukcie, presnosti a rýchlosti ich vyhodnocovania, ako aj schopnosti vytvárania komplexnejších útvarov pomocou interaktívnych nástrojov. V našej práci sa budeme hlavne zaoberať splajnovými povrchmi, teda trojrozmernými splajnami.

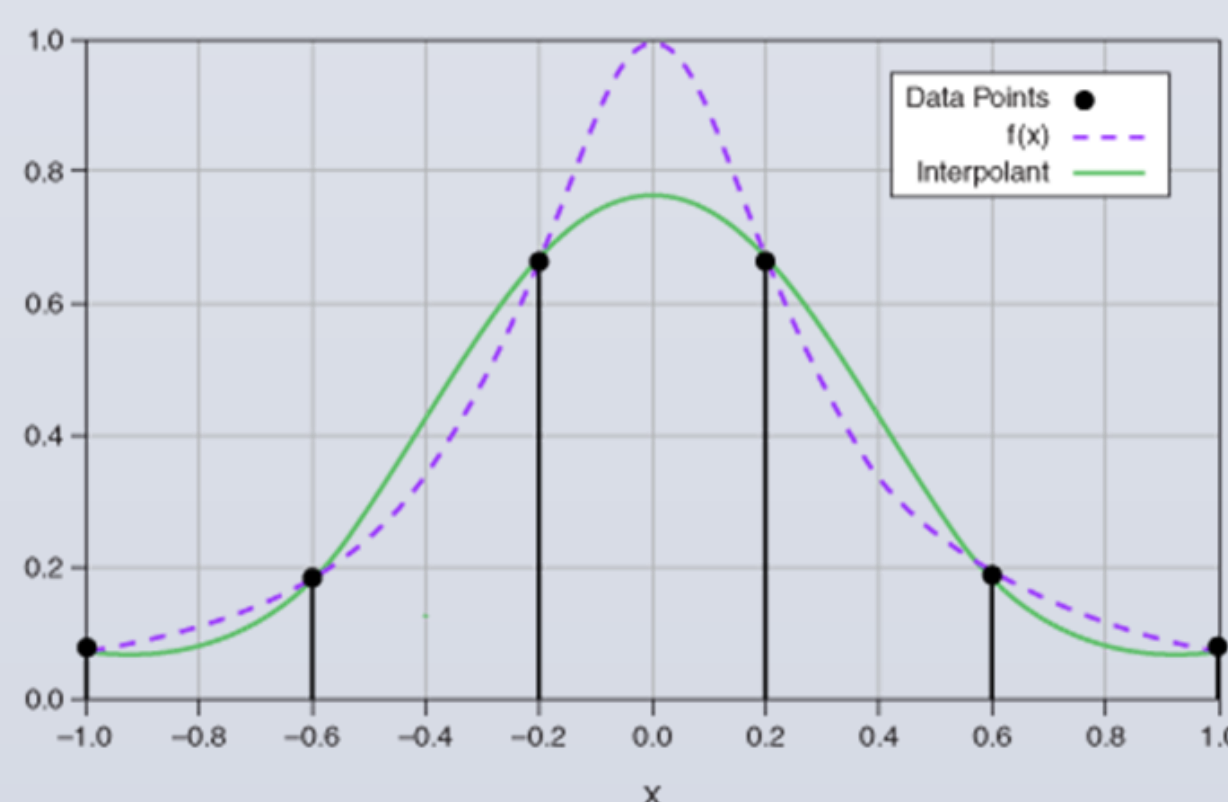
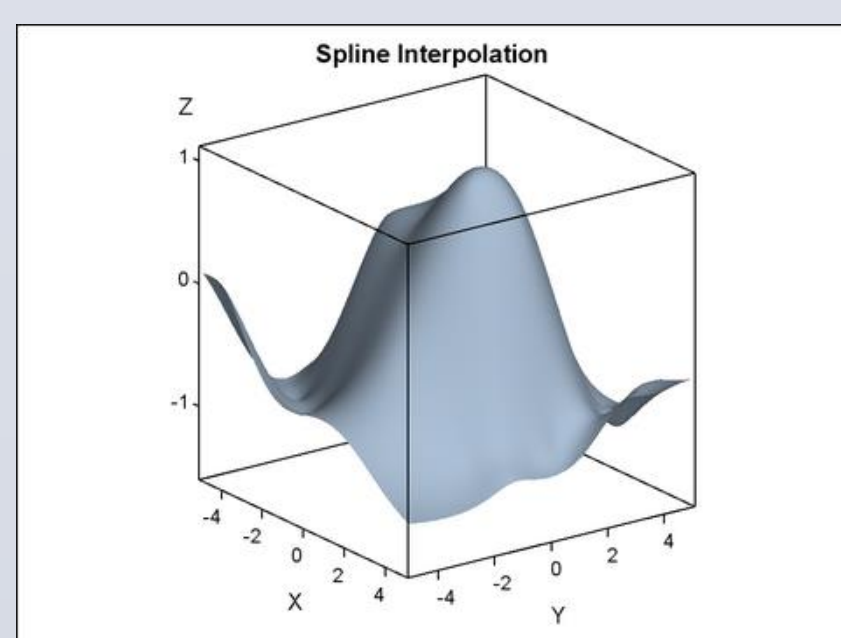


Interpolácia

Interpoláciou nazývame odhad dát na určitom intervale na základe už existujúcich.

Existuje mnoho typov interpolácie (lineárna, polynomiálna). Naša práca používa splajnovú.

Splajnová interpolácia používa polynómy nízkeho stupňa, ktoré produkujú nízke chyby ako polynomiálna interpolácia a zároveň je jednoduchšia na vyhodnocovanie. Takže sa vyhne problému oscilácie. Splajnová interpolácia pozostáva z rozdelenia danej funkcie na intervaly, kde hodnoty krajných bodov sú vopred známe, a nájdenie polynómov korešpondujúcich s jednotlivými intervalmi pre každý z nich. Naš algoritmus je určený na interpoláciu splajnového povrchu, o ktorom platí niekoľko pravidiel. Z dôvodu zachovania hladkosti funkcie sa požaduje, aby sa hodnota derivácie každého polynómu v krajnom bode zhodovala s hodnotou derivácie polynómu v susednom intervale v rovnakom krajnom bode. Úroveň hladkosti (kontinuity) hovorí o tom, koľkokrát je možné funkcie differencovať, aby sa spomínaný vzťah zachoval. Budeme pracovať s bikubickým povrchom, teda povrchom s úrovňou hladkosti 2, ktorý je navyše uniformný, čo znamená že deliace body sú od seba ekvidištančne položené.



Trojdiagonálne matice a sústavy

Trojdiagonálne sústavy rovníc v sebe obsahujú trojdiagonálne matice, ktoré sa vyznačujú tým, že iba prvky, ktoré sa nachádzajú na diagonále alebo riadok nad či pod ňou majú nenulové hodnoty. Zatiaľ čo štandardné systémy matíc sa väčšinou riešia Gaussovou eliminačnou metódou, čo je operácia so zložitou $O(n^3)$, pre trojdiagonálne systémy existujú algoritmy, ktoré ich riešia v lineárnom čase. Sú nimi napr. Thomasov algoritmus či LU dekompozícia. Avšak pri simuláciách a vykreslovaní v reálnom čase môže byť aj sekvenčný algoritmus zložitosti $O(n)$ časovo náročný. Väčšina mtód zlepšenia dnes spočíva v zjednodušení, ako aj paralelizácii spomínaného výpočtu.

$$A = \begin{pmatrix} a_{11} & a_{12} & 0 & 0 & 0 & 0 & 0 & 0 \\ a_{21} & a_{22} & a_{23} & 0 & 0 & 0 & 0 & 0 \\ 0 & a_{32} & a_{33} & a_{34} & 0 & 0 & 0 & 0 \\ 0 & 0 & a_{43} & a_{44} & a_{45} & 0 & 0 & 0 \\ 0 & 0 & 0 & a_{54} & a_{55} & a_{56} & 0 & 0 \\ 0 & 0 & 0 & 0 & a_{65} & a_{66} & a_{67} & 0 \\ 0 & 0 & 0 & 0 & 0 & a_{76} & a_{77} & 0 \end{pmatrix}$$

LU rozklad

LU rozklad alebo dekompozícia je sekvenčný algoritmus na rozdelenie riešenia trojdiagonálnej sústavy na dve menšie podproblémy s lineárnou zložitou. Majme trojdiagonálnu maticu A. LU dekompozícia alebo LU rozklad nám ju umožní rozdeliť na dve matice, L – Lower a U – Upper, pričom platí $A=LU$

Majme trojdiagonálnu sústavu $Ax=d$, po LU dekompozícii $LUx=d$.

Tento problém sa dá rozdeliť na dva podproblémy: $Ux=y$ a $Ly=d$. Tieto rovnice pozostávajú už len z dvojdiagonálnych matíc a dajú sa riešiť v lineárnom čase.

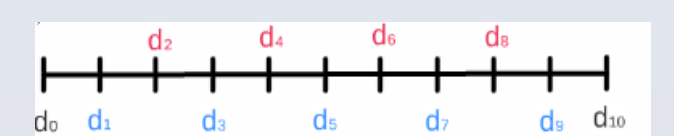
$$[A] = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \dots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nr} \end{bmatrix} = \begin{bmatrix} 1 & 0 & \dots & 0 \\ \ell_{21} & 1 & \dots & 0 \\ \vdots & \vdots & \dots & \vdots \\ \ell_{n1} & \ell_{n2} & \dots & 1 \end{bmatrix} \begin{bmatrix} u_{11} & u_{12} & \dots & u_{1n} \\ 0 & u_{22} & \dots & u_{2n} \\ \vdots & \vdots & \dots & \vdots \\ 0 & 0 & \dots & u_{nr} \end{bmatrix}$$

ABM algoritmus

Na účely paralelizácie využijeme algoritmus ABM - Austin Berndt Moulton. Tento algoritmus má za úlohu rozdeliť trojdiagonálnu maticu na niekoľko menších matíc. Najskôr sa n riadkov trojdiagonálnej matice rozdelí medzi p processormi. Samotný algoritmus pozostáva z 3 fáz: V prvej fáze každý processor pozmení prvý a posledný riadok, ktorý mu patrí. V druhej fáze získané trojdiagonálne sústavy riešime sekvenčným algoritmom (LU rozklad). V tretej fáze si každý processor dosadí hodnoty neznámych vypočítané v druhej fáze do svojej sústavy rovníc, čím získa vlastnú nezávislú trojdiagonálnu sústavu. Celý tento proces po vykonaní rozdelenia môže byť teda plne paralelizovaný bez potreby synchronizácie a komunikácie medzi vláknami.

Redukovaný algoritmus

Štandardný algoritmus Carl de Boor rieši každú deriváciu pre každý index s využitím trojdiagonálnej sústavy. Z výskumu doc. Töröka sa však ukázalo, že ak sa jedná o uniformné kubické splajny/bikubické povrchy, tak týmto spôsobom stačí vypočítať iba každú páru deriváciu a hodnotu zvyšných premenných zistíme triviálne.

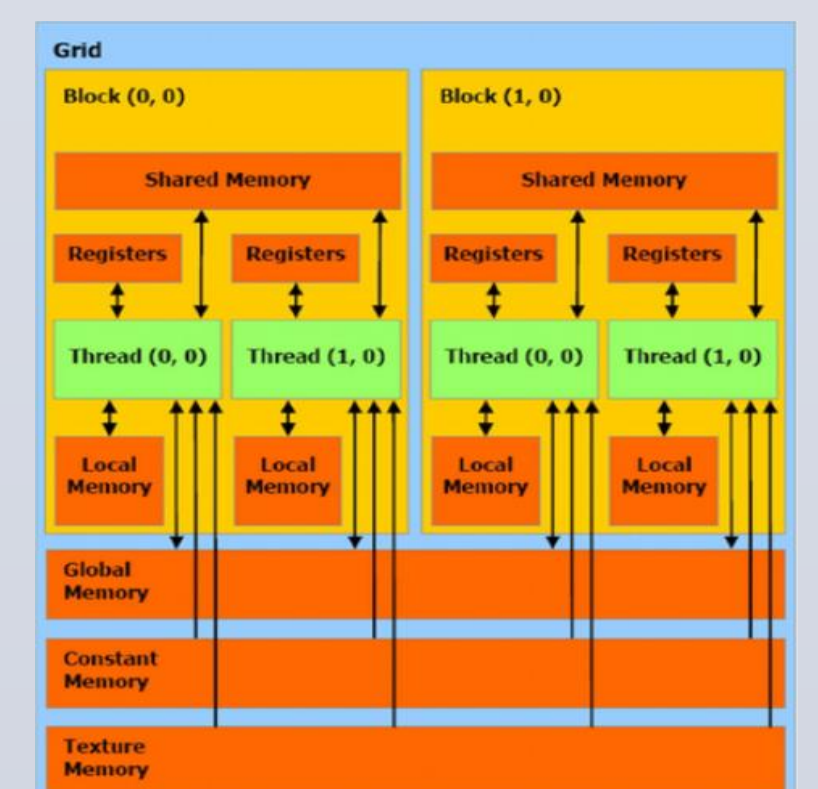
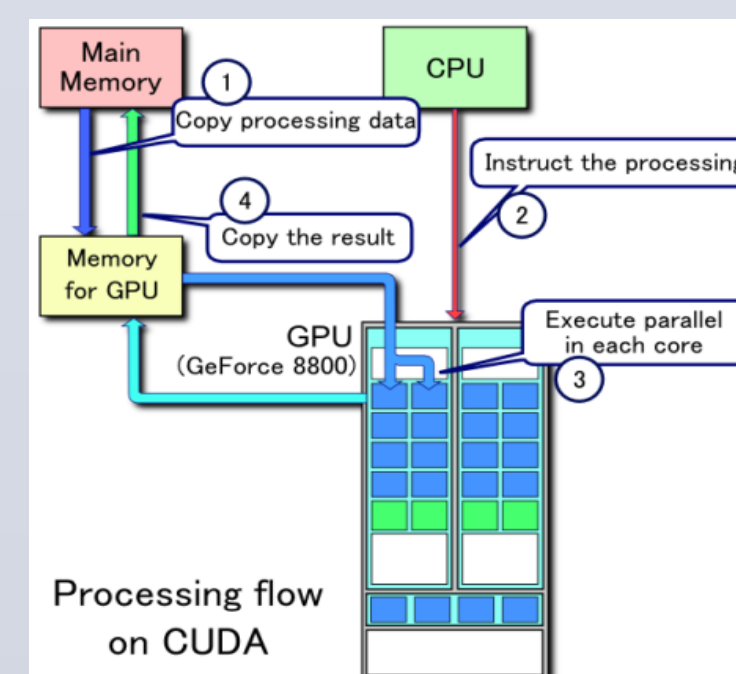


Výpočet

Pri použití finálnej formy redukovaného algoritmu pre bikubický uniformný splajnový povrch dokážeme zredukovať našu sústavu rovníc o takmer polovičný počet neznámych, ktoré sa potom počítajú pomocou algoritmu ABM, kde sa jednotlivé menšie sústavy paralelne počítajú cez sériový algoritmus LU rozklad. Pri párných hodnotách sa jedná a triválny výpočet rovníc, ktoré sú absolútne nezávislé od seba a teda môžu byť plne paralelizovateľné.

Nvidia CUDA

CUDA (Compute Unified Device Architecture) je API od firmy Nvidia, ktorá umožňuje priamu komunikáciu s GPU. Táto platforma je určená na paralelizáciu výpočtov, ako aj manažment pamäte a podporuje programovacie jazyky ako C, C++ a Fortran.



Záver

Pri testovaní urýchlenia na zariadení Intel Core i5-7200, 8 GB DDR4 pamäte vybavenom grafickou kartou NVIDIA GeForce 940MX s 2 GB VRAM sme zaznamenali 1,57 násobné zrýchlenie pri meraní na bikubickom splajnovom povrchu s rozmermi mriežky 100x100 s náhodne generovanými hodnotami z. Merané hodnoty sa môžu líšiť pri zmene vstupných hodnôt ako aj pri použití zariadenia s odlišnými systémovými parametrami.