

Hochleistungsrechnen auf Grafikkarten für innovative Automotiveanwendungen – Simulation der Wasserdurchfahrt eines Serienfahrzeugs

Thorsten Grahs[†], Christian Janßen[★]

[†]move-csc UG , Inst. Scientific Computing, TU Braunschweig

[★] Inst. f. Fluiddynamik u Schiffstheorie, TU Hamburg-Harburg

Innovative Anwendungen der Strömungssimulation (CFD) in der Produktentwicklung – Einsatzgebiete, Methoden, Grenzen

19. März 2013, Wiesbaden

move-csc.de

LBM auf
GPGPUs

Th. Grahs,
Ch. Janßen

Rechnen auf
Grafikkarten

LBM

Anwendung

Resumee

1 Rechnen auf Grafikkarten

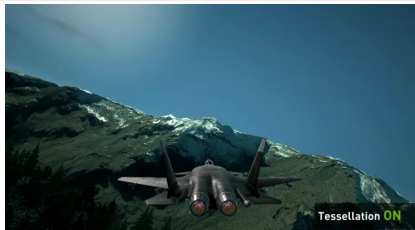
2 LBM

3 Anwendung

4 Resumee

General Purpose Graphical Processing Units (GPGPUs)

- High Performance Computing auf frei programmierbaren Grafikkarten (Unified shader approach, XBox 2005)
- 2006 erste Tesla Serien von NVIDIA
- Programmiermodell (CUDA - Compute Unified Device Architecture)
- Seit 2009 Open CL (Open Computing Language)



CPU vs. GPU

Chip-Design

- CPU optimiert f. seriellen Programmablauf (single thread)
- GPU optimiert für massive parallelen Datenbehandlung (multiple thread)
- Der Grafikkarte ist es egal, ob sie Pixeldaten aufbereitet (tesseliert, transformiert, rendert)
- oder wissenschaftliche Berechnungen anstellt.



Massiver paralleler Datendurchsatz

Hohe Anforderungen an Rechenleistung (real time rendering)

- Programmiermodell
Vektorrechnern nachempfunden
(SIMD)
- Möglichst viele Threads parallel
abzuarbeiten
- Durchsatz orientierter Ansatz
- Hohe Taktrate des Datenbusses



⇒ **Exzellent** geeignet für massives paralleles Rechnen

NVIDIA Quadro 6000

Kleiner Rechencluster in der Workstation

- CUDA cores: 448
- Gigaflops (SP): 1030.4
- Gigaflops (DP): 515.2
- Gesamt-Framebuffer: 6 GB GDDR5
- Speicher-Bandbreite: 144 GB/s

Kosten: ~ 3.500 €



Physikalischer Hintergrund

Kinetische Gastheorie

- Makroskopische Eigenschaften werden durch molekulare Bewegung und Kollision beschrieben
- Betrachte Partikelkollision (Mikroskala)
- Partikelverteilungsfunktion $f(t, \mathbf{x}, \xi)$
- Wahrscheinlichkeit eine Partikels zum
 - Zeitpunkt t
 - Ort \mathbf{x}
 - mit Geschwindigkeit ξanzutreffen



Ludwig Boltzmann (1872)

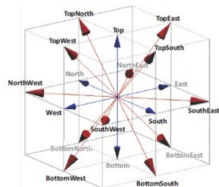
$$\frac{Df}{Dt} = \left[\frac{\partial}{\partial t} + \boldsymbol{\xi} \frac{\partial}{\partial \mathbf{x}} + \mathbf{F} \frac{\partial}{\partial \boldsymbol{\xi}} \right] f(t, \boldsymbol{\xi}, \mathbf{x}) = \Omega$$

- 7 Variablen: Zeit, Ort, Geschwindigkeit

Diskrete Boltzmann-Gleichung

$$\frac{df_i(t, \mathbf{x})}{\partial t} + \mathbf{e}_i \frac{\partial f_i(t, \mathbf{x})}{\partial x_j} = \Omega_i$$

- Beschränkung der Partikelbewegung auf diskreten Geschwindigkeitsraum
- \implies 4 Variablen: Zeit, Ort
- \mathbf{e}_i : Diskreter Geschwindigkeitsvektor
- D3Q19 Modell



- Die diskrete Boltzmann-Gleichung kann weiter diskretisiert werden mit
 - Finite-Volumen-Methode (FVM)
 - Finite-Element-Methode (FEM)
 - nahezu jedem anderen Diskretisierungsansatz
- Diskretisierung mit der Finiten-Differenzen-Methode (FDM) auf einem kartesischen Gitter führt zur

Gitter-Boltzmann-Gleichung (lattice Boltzmann-equation)

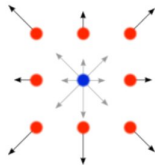
$$f_i(t + \Delta t, \mathbf{x}_i + \mathbf{e}_i \Delta t) = f_i(t, \mathbf{x}_i) + \Omega_i$$

- Zerlegung in **Kollisionsschritt**

$$\bar{f}_i(t, \mathbf{x}_i) = f_i(t, \mathbf{x}_i) + \Omega_i$$

- und **Propagationsschritt**

$$f_i(t + \Delta t, \mathbf{x}_i + \mathbf{e}_i \Delta t) = \bar{f}_i(t, \mathbf{x}_i)$$



Verschiedene Kollisionsmodell vorhanden

- Bhatnagar, Gross, Krook (BGK)-Modell
- Multiple Relaxation Time (MRT)-Modell
- ...

Expandierte LB-Gleichung

Mit entsprechender Wahl der Relaxationszeit und

- Chapman-Enskog-Entwicklung
(Entwicklung um Parameter ε (Knudsen-Zahl)) und
- Taylor-Entwicklung

erhält man Lösungen der Navier-Stokes-Gleichung mit

$$p = \rho c_S^2 = c_S^2 \sum_{i=0}^{18} f_i, \quad v = \frac{1}{\rho_0} \sum_{i=0}^{18} \mathbf{e}_i f_i, \quad \text{mit} \quad c_S = \frac{\Delta x}{\Delta t \sqrt{3}}$$

Vorteile LBM

- Einfache Gittergenerierung
- Geringere numerische Viskosität
- Kein Lösen einer Poisson-Gleichung
- Spannungstensor lokal verfügbar
- Einfachere Parallelisierung

Nachteile LBM

- Dynamische Gitterverfeinerung kompliziert
- Nicht-viskose Strömungen schwer zu simulieren
- Nicht echt inkompressibel
- Probleme bei hohe Dichteverhältnissen
(Mehrphasenströmungen)

Warum?

LBM-Datenstruktur

- kartesisches Gitter
- keine aufwendigen Nachbarschaftsbeziehungen notwendig
- einfache Parallelisierung

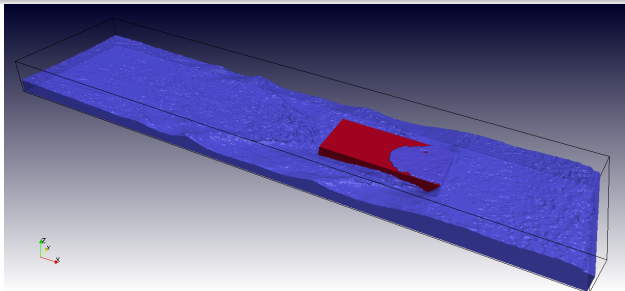
GPGPUs

- hohe Taktrate
- massive lokale Parallelität

⇒ Speedup von 20 gegenüber FV-Methoden möglich

Projektstudie für OEM

- Simulation Kanaldurchfahrt eines Serienfahrzeug
- Interesse: Rechenzeiten LBM-GPGPU
 - bewegter komplexer Fahrzeuggeometrie
 - freier Oberfläche
- Hintergrund:
 - Rechenzeit FVM mehrere Wochen



3D-LBM-GPGPU-code

elbe

efficient lattice boltzmann environment

- Dr.-Ing. Christian Janßen
PostDoc am Inst. Schiffstheorie & Fluiddynamik
TU Hamburg-Harburg
- Freie Oberfläche [LBM + VOF (Volume-Of-Fluid)]
- Validiert in Promotion Janßen (TU Braunschweig, 2010)

Erweiterungen in dem Projekt

- Gitternachführung/Remeshing der bewegten
Fahrzeuggeometrie auf der GPGPU

Rechengebiet

- Kanal: 20.m lang, 5 m breit, 3,75 m hoch
- äquidistantes, cartesisches Gitter
 $\Delta x = \Delta y = \Delta z = 0,025 \text{ m} \implies 24 \text{ Mill. Zellen}$
- Simulierte Zeit: 3 Sekunden; $\Delta t = 0,0000375 \text{ Sekunden}$

- Rechenzeit: 6 Stunden
- Hardware:
NVIDIA Quadro 6000 GPGPU
(Kosten: $\sim 3.500,00 \text{ Euro}$)
- Speedup: ~ 70
(gegenüber aktueller Core-Arch.)



move-csc.de

LBM auf
GPGPUs

Th. Grahs,
Ch. Janßen

Rechnen auf
Grafikkarten

LBM

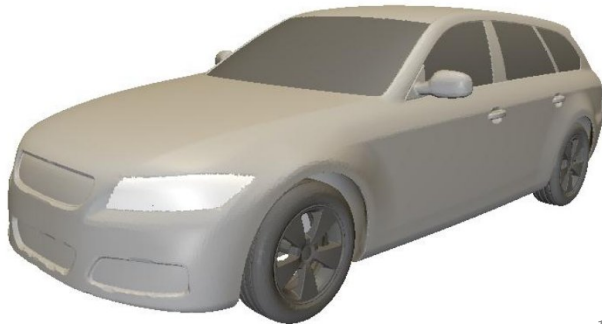
Anwendung

Resümee

DrivAer, TU München

Heft, A. I., Indinger, T. , Adams, N. A.

Introduction of a New Realistic Generic Car Model for
Aerodynamic,
Michigan, USA, : Paper 2012-01-1068, 2012



move-csc.de

LBM auf
GPGPUs

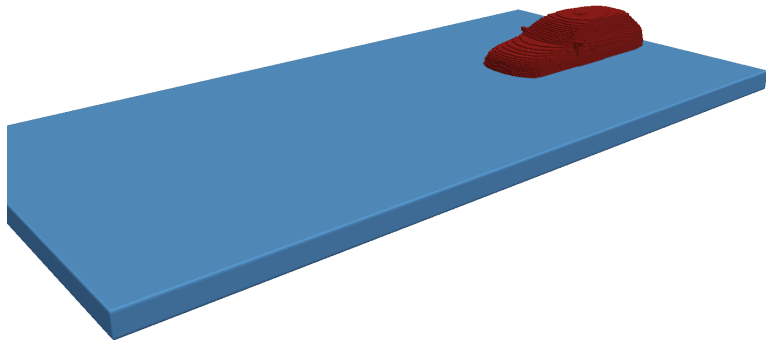
Th. Grahs,
Ch. Janßen

Rechnen auf
Grafikkarten

LBM

Anwendung

Resümee



Dramatische Reduzierung der Simulationszeit

- Simulation auf Grafikkarten ermöglichen dramatischen Zeitgewinn in der Produktentwicklung
 - Tage statt Wochen
 - Stunden statt Tagen
 - Workstation statt Cluster
- Besonders effektiv in Kombination mit LBM
- Die Integration komplexer Multiphysics-Simulationen in den Produktentwicklungsprozess rückt in greifbare Nähe
 - Bewegte Geometrien
 - Mehrphasensimulation/Freie Oberflächen
 - Aeroakustiksimulation
 - Optimierung (DOE)

move-csc.de

LBM auf
GPGPUs

Th. Grahs,
Ch. Janßen

Rechnen auf
Grafikkarten

LBM

Anwendung

Resumee

Geplante Erweiterungen

- Implementierung von Multi-GPUs-Ansatz
 - Integration von adaptive Gitterverfeinerung/local time stepping
 - Integration von Tropfenmodell
 - Elaboriertes Turbulenzmodell
-
- Aussenaerodynamik
 - Aeroakustiklöser

Vielen Dank für die Aufmerksamkeit!

Fragen?