The Potential for Real-time Computational Fluid Dynamics via GPU acceleration

Alistair Revell
The University of Manchester

Overview

- GPU accelerated CFD
- Defining Realtime Simulation and its potential
- The numbers: a recent example and current status
Overview

- GPU accelerated CFD

- Defining Realtime Simulation and its potential

- The numbers: a recent example and current status
Development of CFD algorithms on GPU
- Originally driven by animating realistic physics
- Increasing application to particle-based methods

Initial activity
- Appreciation of the challenge
- Difficulties porting codes

Approach changed
- Codes designed ground-up
- Range of CFD methods
- Pyfr, SPHysics, Sailfish

Renewed potential

![Graph showing the growth of different methods over time](image)
Overview of LBM

- Lattice Boltzmann Makes use of Statistical Mechanics
  - In this room there are billions of molecules hitting us at speeds of order 400m/s!
  - Do we feel them? Do we need to know the behaviour of each molecule?

- In LBM a collection of particles is represented by a distribution function
  - bridges scales by considering a collection of particles as a unit

- There are also some great advantages for GPU
  - Navier Stokes: non-linear and non-local
  - Lattice Boltzmann is linear and local
  - perfect for parallelization on many core architectures
Gaming vs Physics (~15 years)

- Realism vs Accuracy
- Detail vs. Developing Intuition
- Cost, Speed & Convenience
- Potential for closer collaboration?

- PhysX FleX
  instantaneous on a single GPU

- STAR-CCM+
  300M cells IDDES
  order 100,000 CPU hours

- ~1M cells steady RANS
  order 100 CPU hours
Overview

- GPU accelerated CFD
- Defining Realtime Simulation and its potential
- The numbers: a recent example and current status
Defining Simulation Speed

- Game physics is instantaneous
  - Engineering Simulation is not!

- ‘Realtime’ has a clear definition
  - Interactive is open to interpretation

WALL CLOCK TIME
SIMULATION TIME

- Conventional CFD: order 10000 : 1
- GPU accelerated CFD: possibly 100-1000:1
- ‘interactive’
Conventional vs Realtime CFD

- Typical CFD design has 2 loops
  - design loop
  - solution loop

- pre-processing main bottleneck

- bottlenecks also in data transfer
  - increasingly for larger calculations

- interactive CFD can have 1 loop
  - geometry modification, solution and visualization in a single loop

- various means of interacting
  - input devices, augmented reality

- data can’t be saved/transferred
  - faster to view in situ
Most obvious is creation of a virtual environment
- e.g. for training
- realtime is important

But realtime isn’t necessary for all applications
- developing intuition, interactive design
Visualization output

- A range of techniques available from GPU libraries
  - Contour flood (of e.g. velocity magnitude)
  - colormap is stored on GPU to speed up visualisation
- e.g. Imaged Based Flow Visualisation (van Wijk, 2002)
  - simulates advection of particles through a flow field

Volume Rendering too:
- libraries exist from Nvidia
- e.g. Nicolas Delbosc’s work at Univ. Leeds
  - see his Youtube account:
Kinect input: virtual windtunnel

- Input geometry can be obtained from any source
  - E.g. we demonstrate with a Microsoft Kinect (Mawson 2013)
  - Toolkit enables rapid integration with the flow solver

- Kinect Fusion
Human Systems: examples

Teaching/Debugging tool
- Used currently in syllabus and in science fairs
- Provides direct understanding

Interactive Design concept
- Reduce design-engineer loop
- Modest aims at this stage

Surgical Training
- Geometry captured in advance
- Force visualised in realtime

SPH by Guo et al 2015 Comp Animation and virtual worlds
Automated Systems

- **Realtime is generally more important in these cases**
  - Used as part of an environmental monitoring system

- **Forecasting: Faster than Realtime**
  - e.g. extremely local weather forecasting
  - Early warning system: predicting path of contaminant

- **Ability to incorporate other sensors and ‘autocorrect’ simulation**
Automated systems: examples

Work on Data centre cooling at Leeds
- identify need for different levels of cooling in realtime
- simulated flow at a Reynolds number of 10,000, using
  - LBM simulations on a single Tesla K40: 0.34 s per second
  - Fluent on a CPU server across 16 nodes: 7 minutes per second

Contaminant tracking
- potential to be used to track spread of contaminant/pollutant/fire
- in combination with other sensors

Flight Management
- use as a local wind speed prediction tool for drones
- sensing buildings
- combining with forecasts
Data reduction

- when detailed flow information is not required the challenge is to reduce/extract meaningful data on the fly
Overview

- GPU accelerated CFD

- Defining Realtime Simulation and its potential

- The numbers: a recent example and current status
Discretisation of LBM

- The LBE is discretised as follows:

\[ f_i(r + c_i \Delta t, t + \Delta t) = f_i(r, t) + \frac{\Delta t}{\tau} [f_i^{eq}(r, t) - f_i(r, t)] \]

- and is used together with a specific set of discrete velocities defined as:
  - \( D^n Q^m \): for \( n \) dimensions and \( m \) discrete velocities
  - In our work we use

**D2Q9**

- \( w_0 = \frac{4}{9} \)
- \( w_{1-4} = \frac{1}{9} \)
- \( w_{5-8} = \frac{1}{36} \)

**D3Q19**

- \( w_0 = \frac{12}{36} \)
- \( w_{1-6} = \frac{2}{36} \)
- \( w_{7-18} = \frac{1}{36} \)
The algorithm for LBM can now be defined as follows:

1. Initialise
2. Compute equilibrium function
3. Collide
4. Stream
5. apply Boundary conditions
6. Compute Macroscopic values
7. Output data
LBM Validation - 1: Poiseuille flow

2nd order convergence up to floating point precision in 3D

in DP memory limit hit before floating point error
LBM Validation - 2 : LDC

2D Lid Driven Cavity

3D Lid driven Cavity

Centreline $u$ profiles for Re=100, 400 and 1000
Code optimised for GPU (Mawson 2013)

<table>
<thead>
<tr>
<th>Feature</th>
<th>FK104: K5000M</th>
<th>GK110: K20c</th>
</tr>
</thead>
<tbody>
<tr>
<td>cores (SMX x cores/SMC)</td>
<td>1344 (7 x 192)</td>
<td>2496 (13 x 192)</td>
</tr>
<tr>
<td>regs / thread</td>
<td>63</td>
<td>255</td>
</tr>
<tr>
<td>DRAM</td>
<td>4GB</td>
<td>4.7GB</td>
</tr>
<tr>
<td>SP/DP ratio</td>
<td>24:1</td>
<td>3:1</td>
</tr>
<tr>
<td>Peak performance (single precision)</td>
<td>1.6 TFLOPS</td>
<td>3.5 TFLOPS</td>
</tr>
<tr>
<td>DRAM Bandwidth</td>
<td>66 GB/s (measured)</td>
<td>143 GB/s (measured)</td>
</tr>
</tbody>
</table>
Optimization steps

- fold arrays flat
- write arrays using f direction first and rely on ‘un-coallesced’ access
- Change algorithm order to reduce read/write of data during loop

**PUSH**
1. initialise
2. compute forces
3. compute $f^{(eq)}$
4. collide (local)
5. stream (non-local)
   - i.e. requires synchronisation
6. impose bcs.
7. compute macroscopic quantities

**PULL**
1. initialise
2. compute forces
3. compute $f^{(eq)}$
4. collide
5. stream
   - i.e. read values from host into new location
6. impose bcs.
7. compute macroscopic quantities
Overall performance-3D

- Peak 814 MLUPS K20c; ~92% of bandwidth scaled performance
- For realtime CFD this means a resolution of $160^3$ at a refresh rate of 200fps
- Main limitation was on board memory
- Current hardware is a factor 2-3 faster with larger memory
Challenges for realtime

- **True realtime is > 24 frames/second**
  - so graphics output interval is 40ms
  - Visualisation, used carefully and on board is not restrictive

- **Data output must be minimal**
  - 1000 MLUPS and higher, 1GB-1TB data can be generated per second
  - e.g. from Delbosc (2015): for $128^3$ LDC the following are per iteration

<table>
<thead>
<tr>
<th>Time</th>
<th>Task</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.4 ms</td>
<td>LBM</td>
</tr>
<tr>
<td>6.8 ms</td>
<td>GPU to CPU</td>
</tr>
<tr>
<td>1700 ms</td>
<td>write data (total 24GB / second)</td>
</tr>
<tr>
<td>0.1 ms</td>
<td>display results using OpenGL</td>
</tr>
</tbody>
</table>

- **LBM structure imposes limitations**
  - high numbers of registers in LBM (e.g. in 3D 19 pops, 4 macro + other integers)
  - memory requirements also limit domain size on GPU
  - new developments: e.g. *Link-Wise Artificial Compressibility* (Asinari, Obrecht)
Current status

- DNS (blue line) and LES (red line) vs. Reynolds number
- Realtime (green line) and Interactivity (yellow line)

Capacity for LES/DNS:
- solid lines based on desktop with 4 K20s
- ~ peak 3000 MLUPS

Memory limitation:
- corresponding to ~15M points

Dashed lines indicate Titan:
- multi GPU LBM implementation (Robertson & Mattila 2015)
- 16384 GPUs
- ~ peak 1 800 000 MLUPS
- i.e. 2 x 10^{12} LUPS

Numbers based on LDC calculations and classic scaling laws, also using results in the literature.
Conclusions

- ‘Realtime’ CFD is on the horizon
  - hardware in the loop, automated CFD analysis for predictive purposes
  - currently up to Re~10^4 for LES on a desktop with multiGPU

- Use of interactive CFD is increasing
  - range of applications for virtual environments, learning, testing & design

- Main focus has been on LBM
  - Many challenges remain: wall modelling, reduction of memory overhead

Acknowledgements

- UKCOMES: UK Consortium for Mesoscale Engineering Sciences (Grant No. EP/L00030X/1)
- Others at University of Manchester: George Lever, Adrian Harwood
- Dr. Mark Mawson, STFC
- Dr. Nic Delbosc, Dr Jon Summers (Univ Leeds)
- Dr. Christian Obrecht (Lyon)