A GPU cluster optimized multigrid scheme for computing unsteady incompressible fluid flow

Dr. György Tegze

Hungarian Academy of Sciences
Wigner Research Centre for Physics
Budapest

22 June 2015

introduction

- code optimization - pushing the algorithm to limits
- algorithm selection and design
- outlook for 3D simulation
- multi-billion grid points
- 2D simulations

compute system

hardware resources

- arithmetics (FLOPS count)
- memory: 40+ GB (4GB + overhead/billion discrete variable)
- memory bandwidth
- interconnect (distributed systems)

In advance .... FLOPS count is rarely a bottleneck

Formulating physics for computers

our work scheme:

- formulate a physical phenomenon as partial differential equations (PDEs)
- series expansion of the solution function
- set of linear algebraic equations (with specific properties)
- billions of discrete variables
- solving the algebraic equations

most steps are generic: signal/image processing, pattern recognition, variational optimization etc.

Choosing an expansion

Fourier expansion

- effective on a single GPU (trivial discrete eqs.) : amount of memory is limiting
- multi GPU: consecutive FFT and transpose: interconnect bandwidth is limiting
- multi GPU: compute copy overlap: interconnect latency is limiting

Taylor expansion

- single GPU: can be effective (amount of memory is limiting)
- multi GPU: compute copy overlap is effective

Optimizing for memory footprint

pays off

- memory size is often a limitation
- significant factor in system cost
- potentially decrease bandwidth demand - faster computations

methods of optimization

- decreasing the number of variables
- advanced grid generation: decrease number of meshpoints
- decreasing the memory overhead of solving the linear system

The incompressible NS equation

fundamental eq. for momentum and mass conservation

$$\frac{\partial \mathbf{v}}{\partial t} = -\mathbf{v} \cdot (\nabla \otimes \mathbf{v}) + \frac{\eta}{2} \nabla \cdot (\nabla \mathbf{v} + \nabla \mathbf{v}^T) - \nabla p \qquad (1)$$

$$0 = -\nabla \cdot \mathbf{v} \qquad (2)$$

variables:

- 3 variables in 2D (velocity components + pressure)
- each variable is used to compute a component
- compute all variables in a single kernel (if possible)
- few arithmetic operations/discrete variable (finite difference)

The solution strategy: Chorin's projection method

decomposing pressure as: $p^{t+1} = p^t + \delta p$

predicting velocity - trivial linear system

$$\mathbf{v}^* = \mathbf{v}^t - \Delta t[\mathbf{v}^t \cdot (\nabla \otimes \mathbf{v}^t) + \eta \Delta \otimes \mathbf{v}^t] - \nabla p^t \qquad (3)$$

Substituting $\mathbf{v}^{t+1} = \mathbf{v}^* - \nabla \delta p$ into Eq. (2)

pressure equation - matrix equations

$$0 = \nabla \cdot \mathbf{v}^* + \nabla^2 \delta p \qquad (4)$$

correcting velocity

$$\mathbf{v}^{t+1} = \mathbf{v}^* - \nabla \delta p \qquad (5)$$

Solving large linear systems

direct solvers: Gaussian elimination and its variants

- excessive cost for large systems $O(N^3)$

iterative solvers

- GMRES: $O(N log(N))$ complexity, fast convergence, data dependence - complex schemes and compute code
- CG and variants: $O(N)$ complexity, gradients must be stored
- Gauss-Seidel, Jacobi, SOR: simple, low memory usage, $O(L^2)$ complexity

hybrid solvers

- Gauss-Seidel+multigrid: low memory usage + $O(L)$! complexity

The multigrid method

multiresolution
discretization

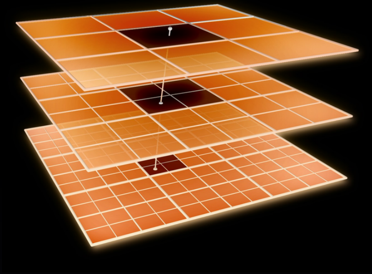
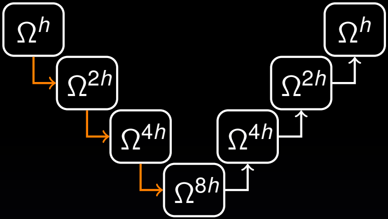


Fig. by Marius Sucan

- residual is relaxing on multiple wavelength
- faster convergence

multigrid cycle

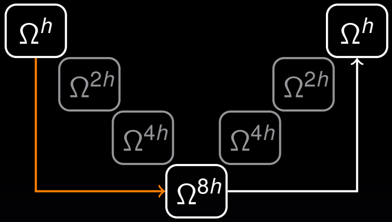


1. GS iteration
2. downsampling
3. resampling

decreasing system size - interconnect latency limit

Decreasing the number of iterations

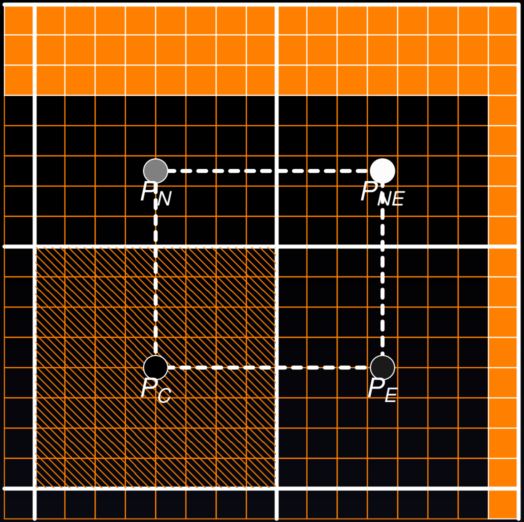
sparse multigrid cycle



- we assume that not all discretization levels are equally important
- we try to bypass some levels

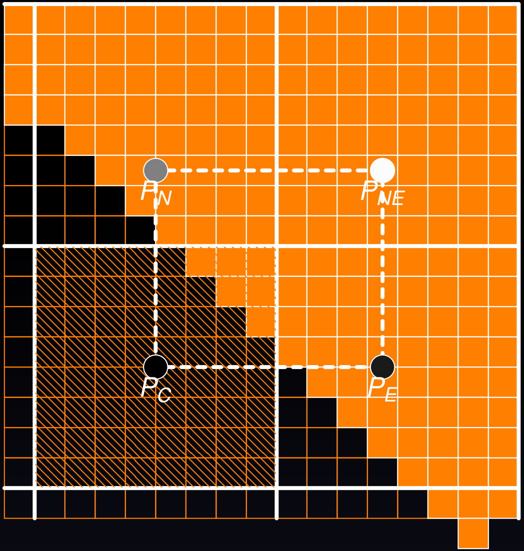
- arithmetic cost slightly increased
- iteration count decreased to $1/5$ of the full cycle
- further tricks to decrease aritmetic cost

Geometric multigrid (GMG)



- re-discretization of the contiuum equations
- using the same stencil as on the finer grids
- inconsistent: no guarrantee that more iterations on the coarse grid will result better prediction on the fine grid

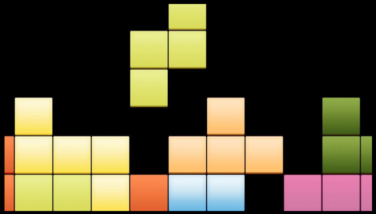
GMG sloped boundary



- no generic Laplacian stencil that provides consistency
- excessive iterations
- makes cycling strategy more complicated
- difficult to avoid convergence stagnation

Additive Correction Multigrid (ACM)

- downsampling: Coarse discrete eqs. are simple sum of the discrete eqs.
- no re-discretization at coarse levels
- converging to the fine grid solution, on all levels
- consistent



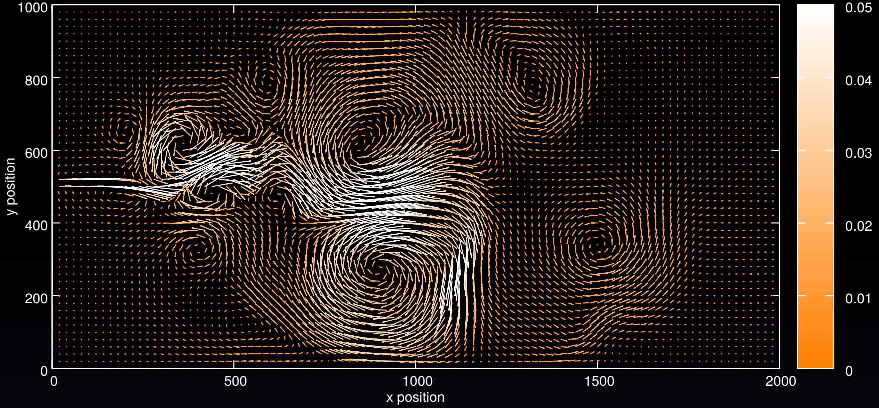
- ACM: FD stencil fitting to the problem!

benchmark: 2D turbulence

- Soap film flowing through a comb.
- flow patterns on various scales
- chaotic

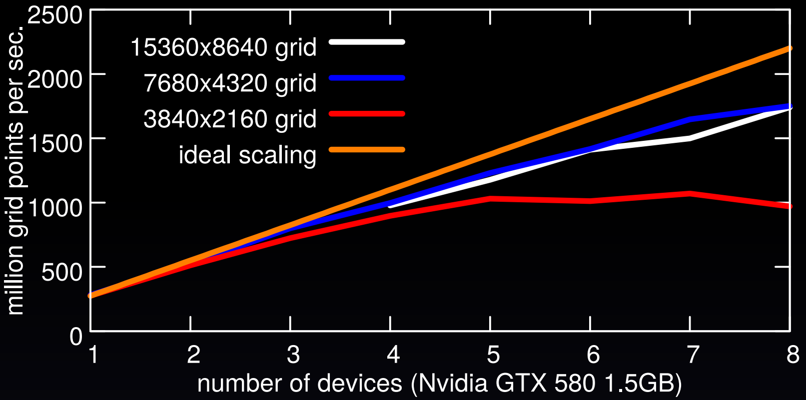
M.A. Rurgers: Soap film turbulence

2D jet



- Fluid jet mimics a gap in the comb $-->$ equally spaced jets
- eddy size growing (multiple scales to resolve)

benchmark: 2D turbulence



Boffetta 2009 32k^2 pseudo-spectral

Acknowledgement

- Prof. Laszlo Granasy for support and discussions
- Mate Nagy, and the Wigner GPU Lab for supporting us with hardware and programming advices

This work has been supported by the EU FP7 Collaborative Project "EXOMET" (contract no. NMP-LA-2012-280421, co-funded by ESA), and by the ESA MAP project "GRADECET" (Contract No. 4000104330/11/NL/KML). G. Tegze is a grantee of the János Bolyai Scholarship of the Hungarian Academy of Sciences.