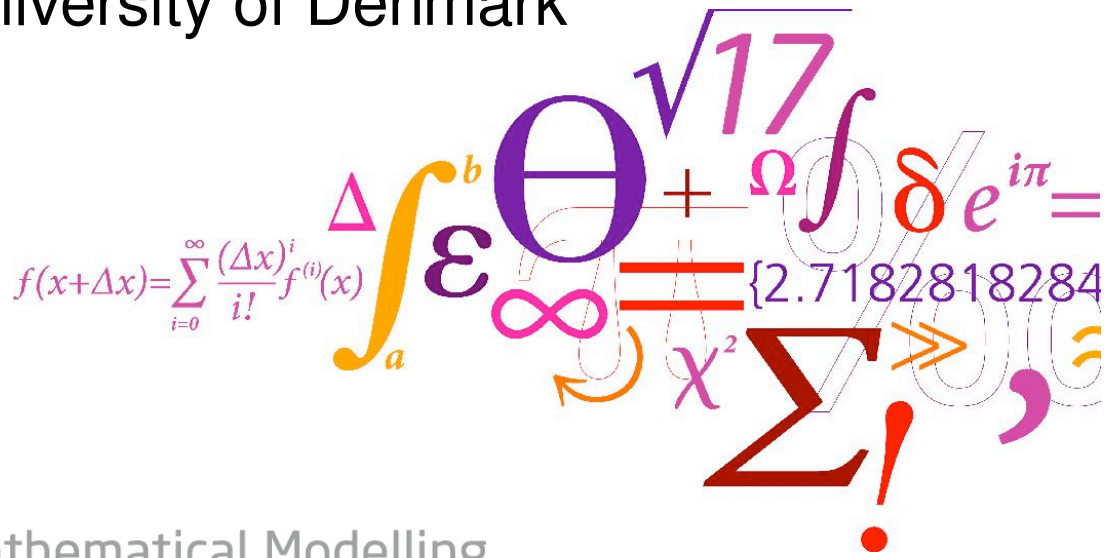
Scientific Computations on GPUs

First results from GPULab @ DTU Informatics

Bernd Dammann
Associate Professor
Scientific Computing – DTU Informatics
Technical University of Denmark



DTU Informatics
Department of Informatics and Mathematical Modelling

Technical University of Denmark

Students, 2008

	Admitted	Graduating	Total enrollment
Total	**1,737**	**1,414**	**6,270**
BEng	509	268	1,787
BSc	715	391	2,192
MSc	513	755	2,291

International students, 2008

Exchange students admitted	560
Summer University students	75
International MSc students admitted	258*
DTU students abroad	162

* Includes all Nordic citizens and other non-Danish citizens with a permanent residence permit in Denmark.

Staff and PhD fellows (head counts, 2007)	
Professors	129
Associate professors and senior researchers	739
Assistant professors, researchers and postdoctoral fellows	443
Research assistants	29
Other scientific personnel	156
Total scientific personnel	**1,496**
Technical and administrative personnel in institutional departments	1,788
Total staff in institutional departments	**3,284**
Administration and campus service	309
PhD fellows, including from other institutions	909
Total institutional staff	**4,502**

Outline

- GPULab @ DTU Informatics
  - Motivation
  - Goal
  - People
  - Equipment
- Projects & First results
- Future

Why do we work with GPUs?

- We cannot neglect the development
- Our “customers” will ask for it
- DTU Informatics has the expertise in-house:
  - Scientific Computing & HPC
  - Computer Graphics
  - Embedded Systems Engineering
- We can attract new students (MSc & BSc):
  “GPU computing is a hot topic”

GPULab @ DTU Informatics

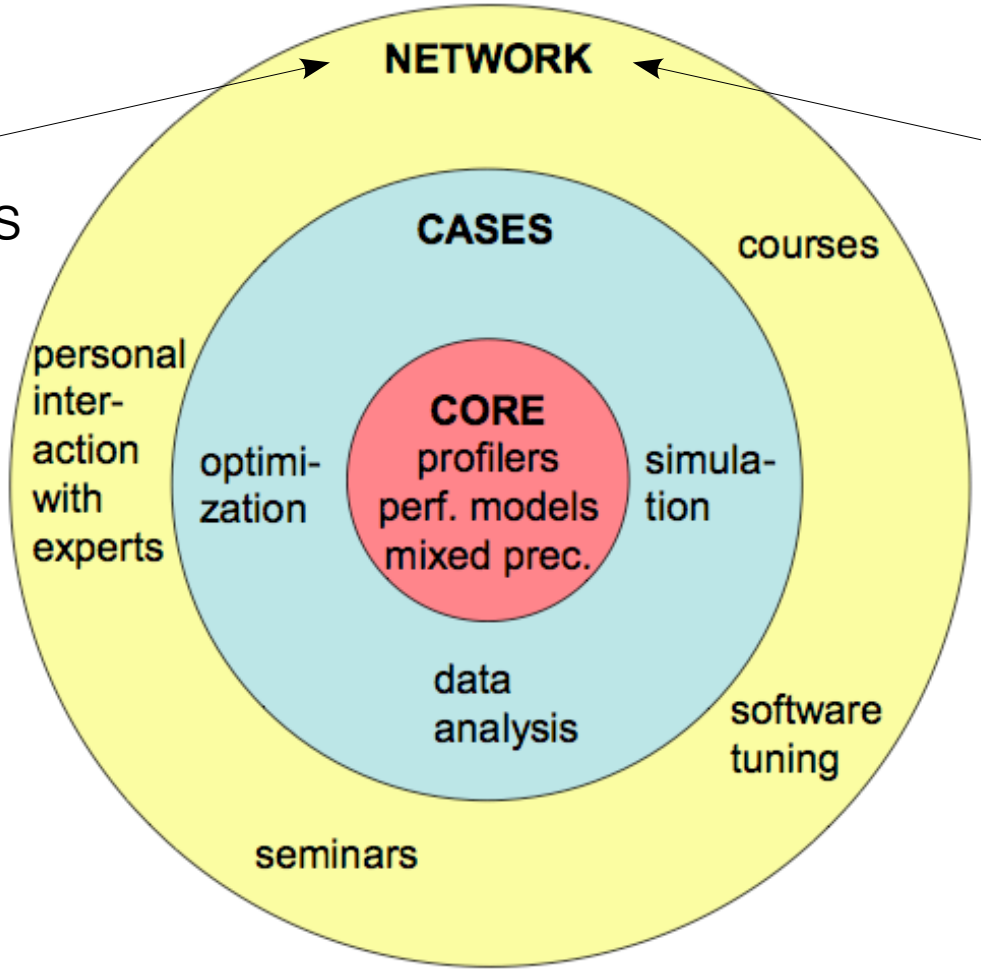
Research Proposal:

"Desktop Scientific Computing on Consumer Graphics Cards"

- Proposal was accepted by "The Danish Council for Independent Research – Technology and Production Sciences", December 2009
- 2 PhD positions & 1 Postdoc
- Project start: April 1, 2010

GPULab @ DTU Informatics

Goal:



Industry:
- QuantumWise A/S
- Brüel & Kjær
- DHI Group
- MOSEK Aps
- NVIDIA(?)
- ...

Academics:
- Brown University
- Rice University
- INRIA
- RWTH Aachen
- Univ. of Antwerp
- Copenhagen Univ.
- Aalborg Univ.
- ...

GPULab @ DTU Informatics

People:

- Prof. Per Chr. Hansen
- Assoc. Prof. John B. Jørgensen
- Assoc. Prof. Bernd Dammann
- Assistant Prof. Allan Peter Engsig-Karup
- Assistant Prof. Jeppe Revall Frisvad
- Postdoc Hans Henrik Brandenborg Sørensen
- 2 PhDs
- MSc & BSc students

Scientific GPU Computing for Dynamical Optimization

Test problem: distribution and control of electrical power via a “smart power-grid” through the use of model predictive control (MPC).

Computational bottleneck: matrix computations – we aim to develop high-performance algorithms that overcome this bottleneck.

Our developments will also produce advances in other areas of constrained optimization, convex optimization, and MPC.

Candidates for this PhD position:

- master in CSE, applied mathematics, or engineering
- programming skills in HPC and/or multi-core
- familiar with optimization algorithms and/or matrix computations.

DTU Informatics GPULab

Our equipment:

- 1 PC with
  - quad-core CPU, 4 GB RAM
  - Nvidia GeForce GTX280, 1 GB RAM
- 1 PC with
  - quad-core CPU, 12 GB RAM
  - Nvidia Tesla C1060, 4 GB RAM
  - Nvidia GeForce 9500GT, 1 GB RAM
- Linux 2.6 kernel, with CUDA
- students develop on laptops with CUDA cards

Projects @ GPULab

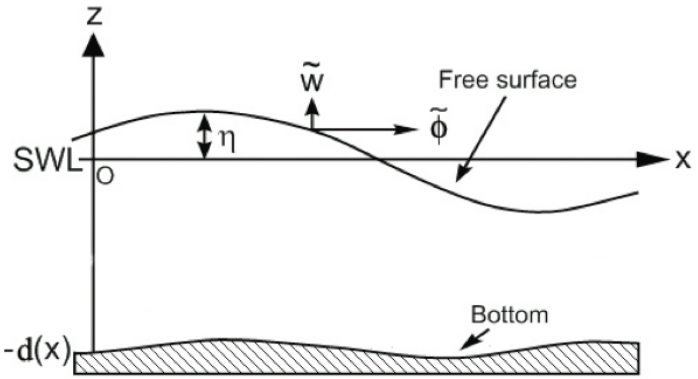
Existing and planned projects:

- "Solver for non-linear water waves"
- "Sound algorithms on multi-core CPUs and GPU: A comparison study"
- "Modelling the CUDA architecture"
- "Fast computational methods for high-resolution ODF problems on many-core systems"
- more multi-core vs. GPU comparisons

Non-linear water waves

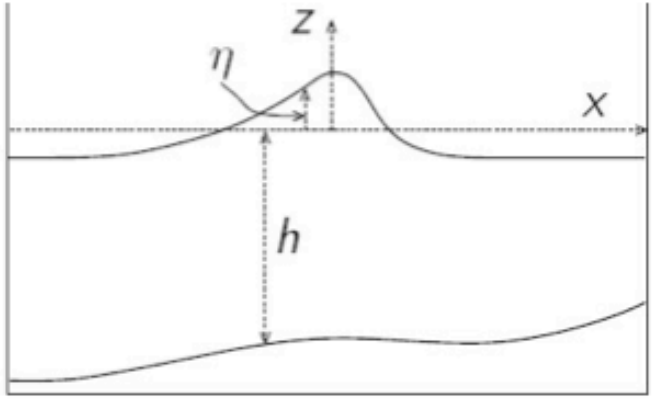
- Wave loadings on ships, offshore, platforms
- Influence of the bottom interaction in coastal regions
- Seakeeping & manouevring
- ...

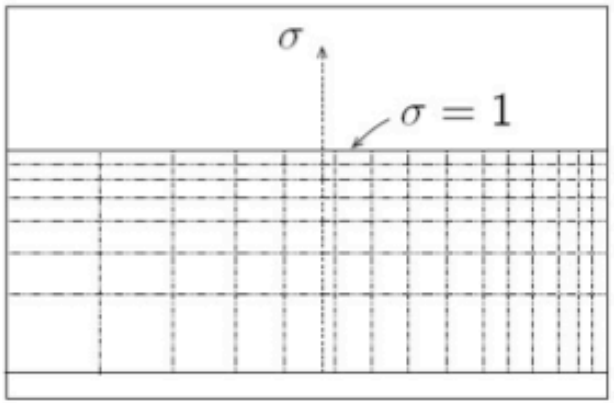
Non-linear water waves



Apply a sigma transform to the vertical coordinate:

Physical domain ⇒ Computational Domain





Non-linear water waves

- There are many ways to solve this problem
- Our choice: a combination of
  - GMRES
  - Pre-conditioning
  - Multigrid
- Robust and efficient

- Work presented here: linear water waves, only

Non-linear water waves

GPU limitations:

- Tesla C1060 has 4 GB of memory
- This allows us to solve the problem for grid sizes up to 8193 x 8193
- For this problem size one needs to solve a linear system of over 67 mio equations

Non-linear water waves

Choice of algorithm:

- Most important aspect: scalability
- Algorithms that perform well on a single-core CPU may perform much worse on – or may not even be suitable for – a GPU
- Avoiding memory access collisions in the algorithm is one of the most important aspects

Non-linear water waves

Development environment:

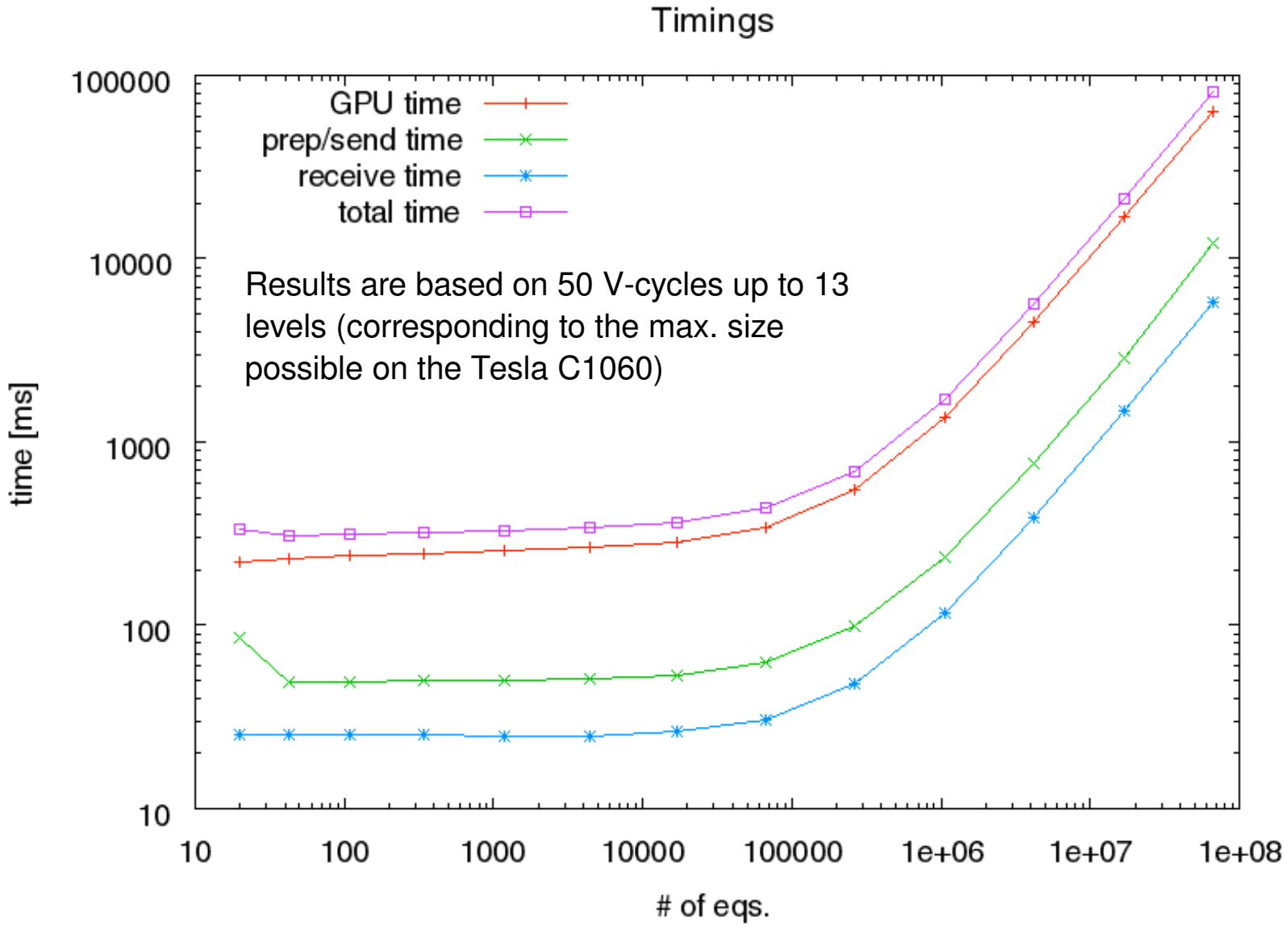
- Programing languages:
  - C#
  - CUDA
- Libraries:
  - CUDA.NET
  - .Net framework (Windows)
  - Mono framework (Linux)

Non-linear water waves

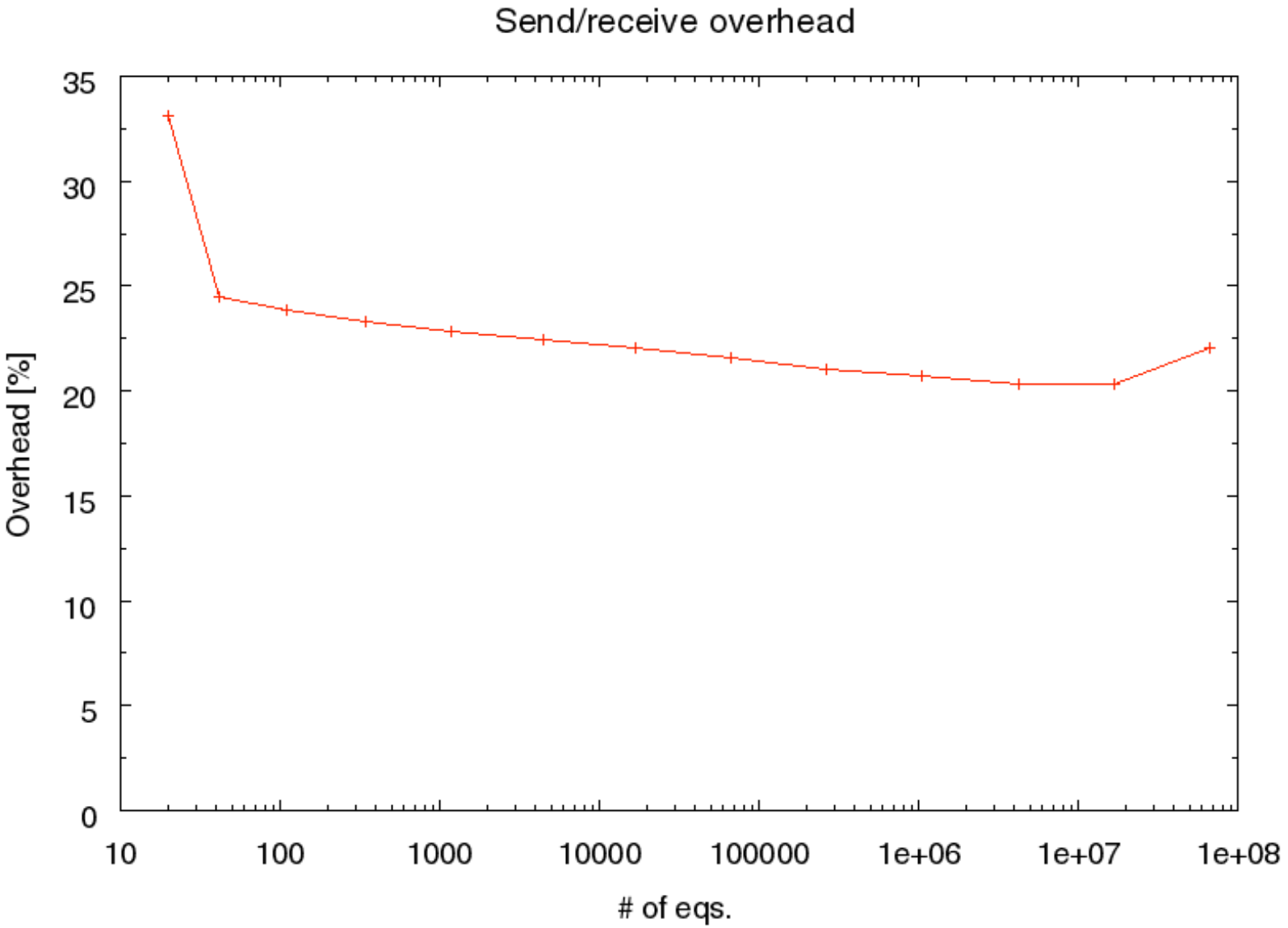
CUDA issues:

- An incorrectly executed kernel does not give an error. Instead it may crash the system!
- Amount of arguments passed to a kernel is limited, forcing the programmer to use workarounds and produce less readable code.
- Many other annoying issues like compiler errors
- Despite all that it is much more pleasant to work with CUDA compared to working with shaders.

Non-linear water waves

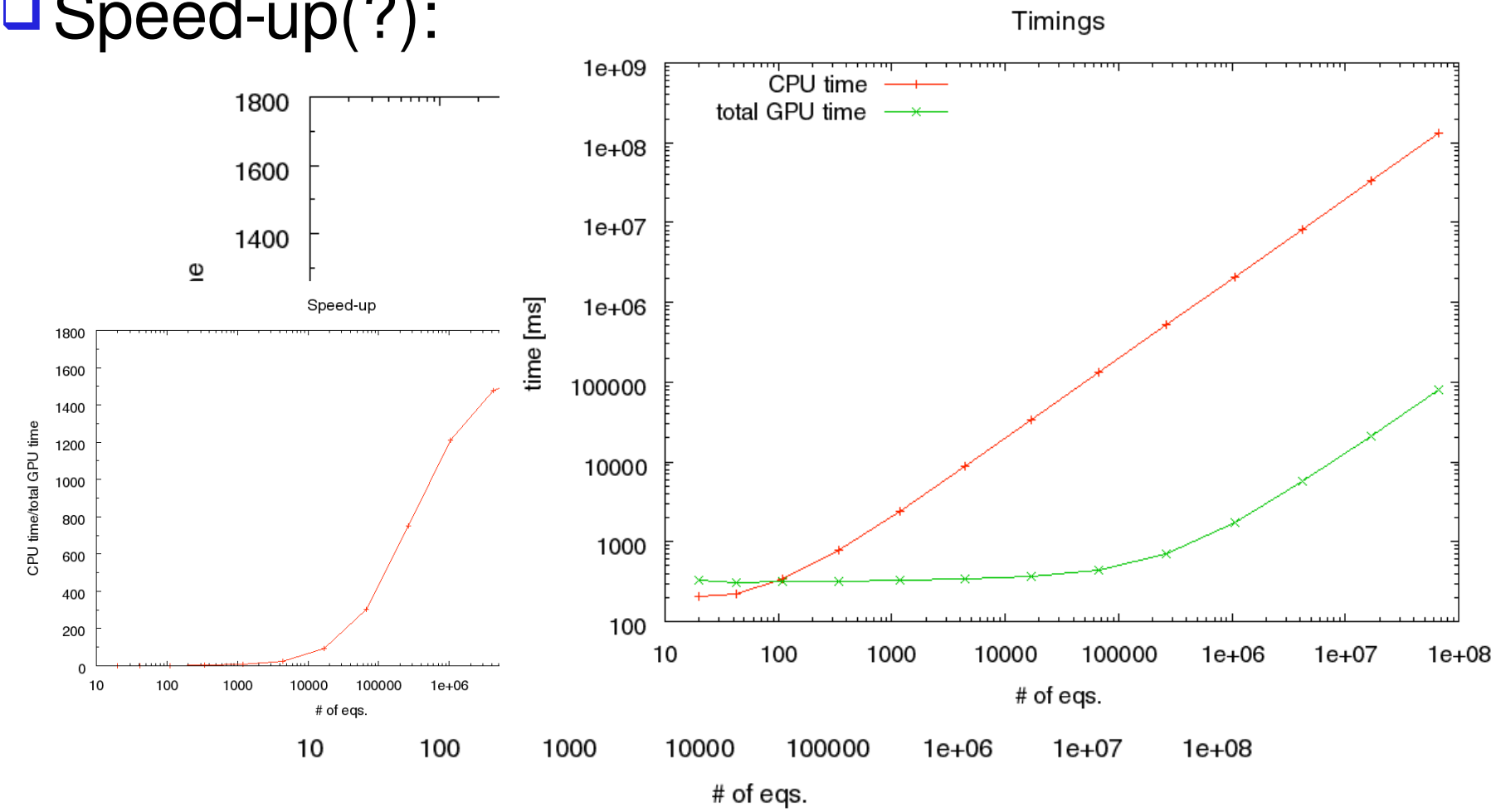


Non-linear water waves

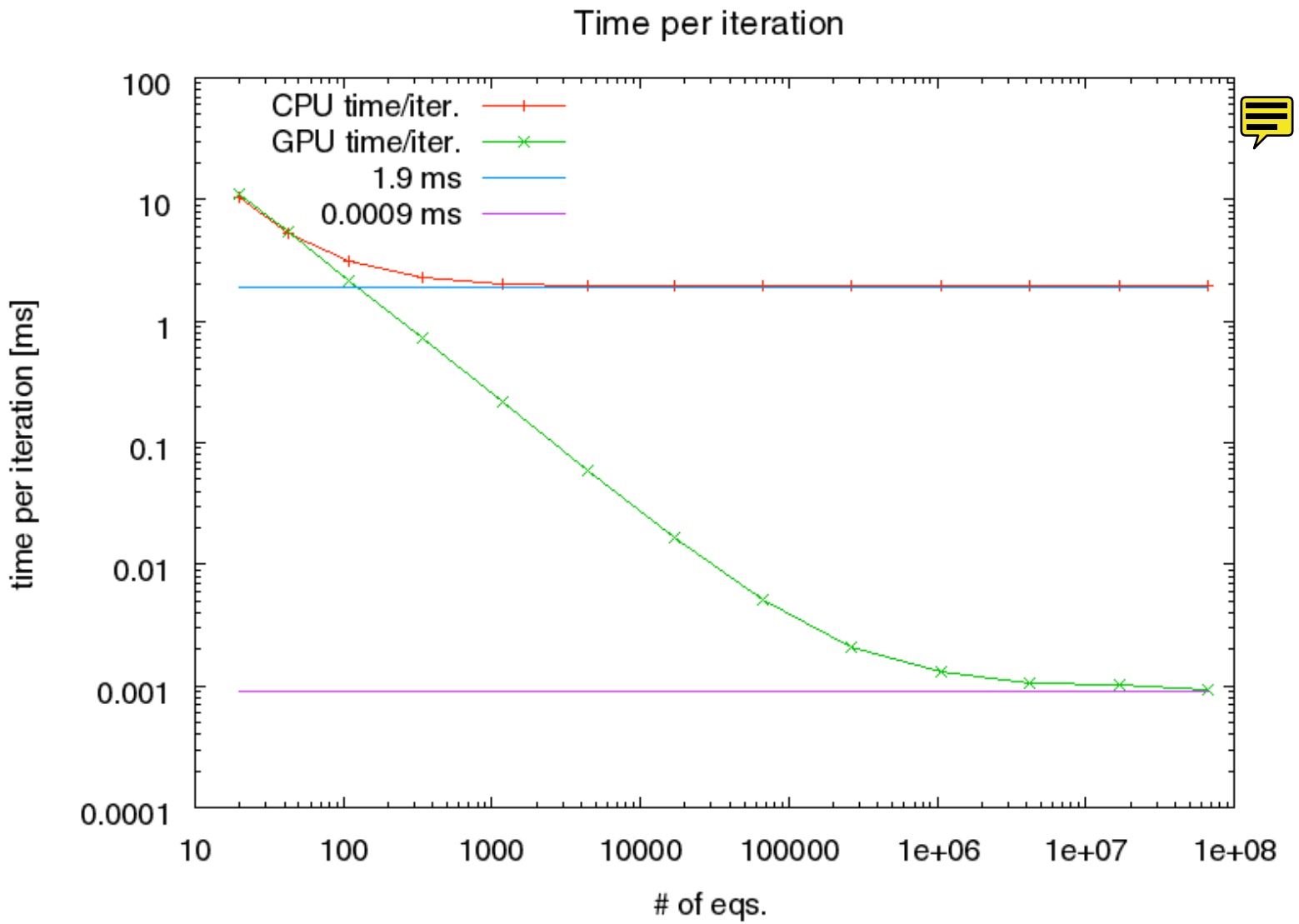


Non-linear water waves

- Speed-up(?):



Non-linear water waves



Non-linear water waves

Summary:

- Results look very promising
- 67 mio. equations compared to 2.6 mio on CPU
- Speed-ups obtained are not very realistic (CPU implementation in C#/Mono) – would be better to compare to the original Fortran implementation.
- send/receive overhead is constant at around 25% for all problem sizes – can be improved since we now use more data than needed for development

The future ...

"Prediction is very difficult, especially if it is about the future." -- Niels Bohr (1885-1962)

- In the past:
  - we have had "accelerators" before: transputers, etc
  - specialized hardware – not a big market.
  - who remembers them today?
- GPUs are a based on a mass market product
- GPU computing will (probably) not go away
- … but it will develop/change

The future ...

Challenges:

- re-design of algorithms:
  - re-think memory access
  - re-consider algorithms (decisions made in the past)
- better code development tools are needed
- we will need standards:
  - to assure portability
  - long-term perspective of our applications

The future ...

Strategies:

- use existing libraries and tools whenever possible
- develop new strategies for 'special cases'
- re-think algorithms with respect to the new additional resources
- make use of our in-house expertise and the external networks

Thank you!

http://gpulab.imm.dtu.dk/