GPU acceleration of the Simple Cell Mapping algorithm

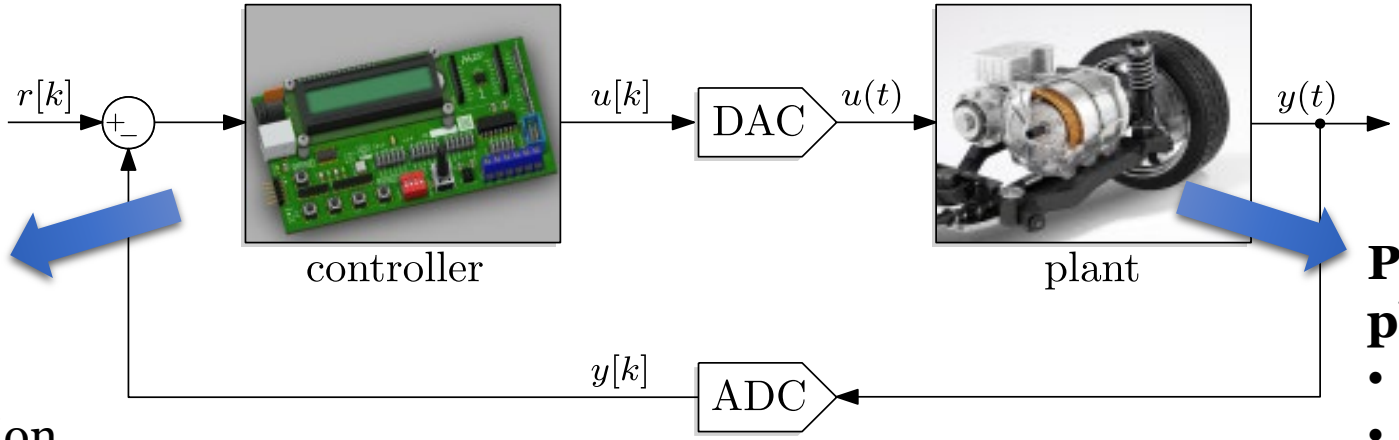
Tamás Haba, Csaba Budai

Department of Mechatronics, Optics and Mechanical Engineering Informatics
Faculty of Mechanical Engineering
Budapest University of Technology and Economics

GPU Day 2024

Introduction

- Structure of a mechatronic system:



Digital phenomena:
- Sampling
- Quantization
- Number representation

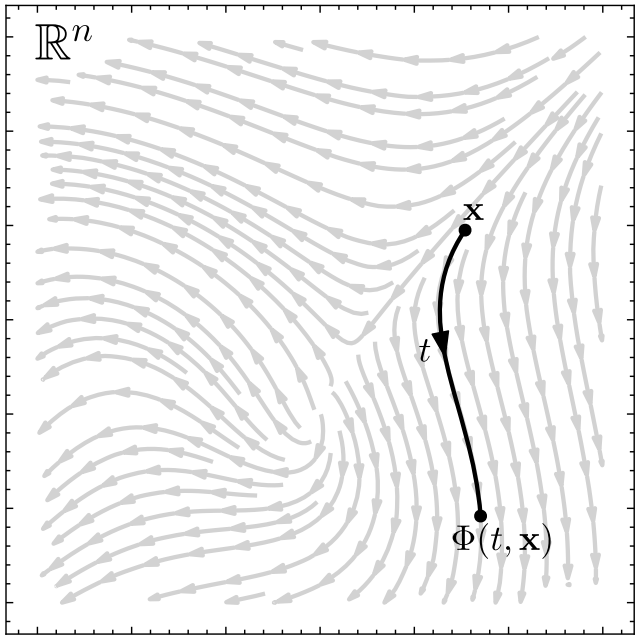
Properties of the physical system
- Friction effects
- Actuator saturation
- Nonlinearities

- Dynamical analysis
  - Linearization (locally)
  - Globally: various numerical methods → high computational costs
- This research: parallel computing application in mechatronics

Fundamentals of the cell mapping

Dynamical system

$\Phi\colon \mathbb{R} \times \mathbb{R}^n \to \mathbb{R}^n$

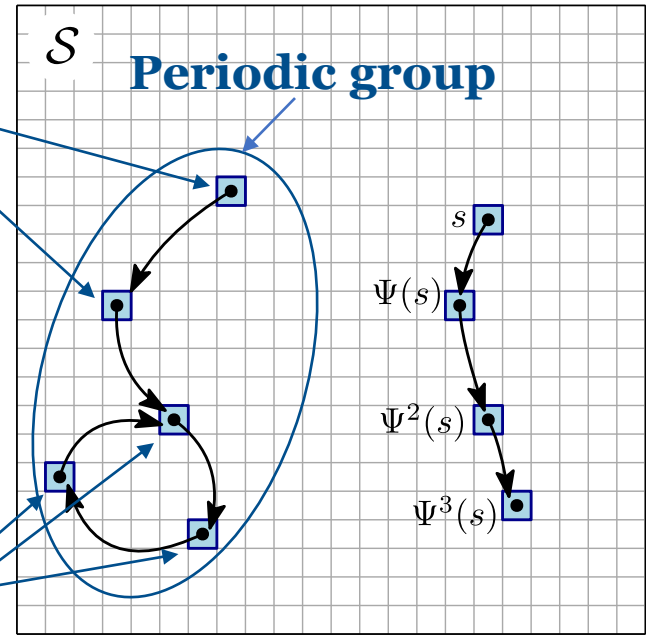


- Phase space: $\mathbb{R}^n$
- Flow map: Φ

Spatial and temporal discretization

Cell mapping

$\Psi\colon \mathcal{S} \to \mathcal{S}$



- Cell state-space: $\mathcal{S}$
- Cell-to-cell mapping: Ψ

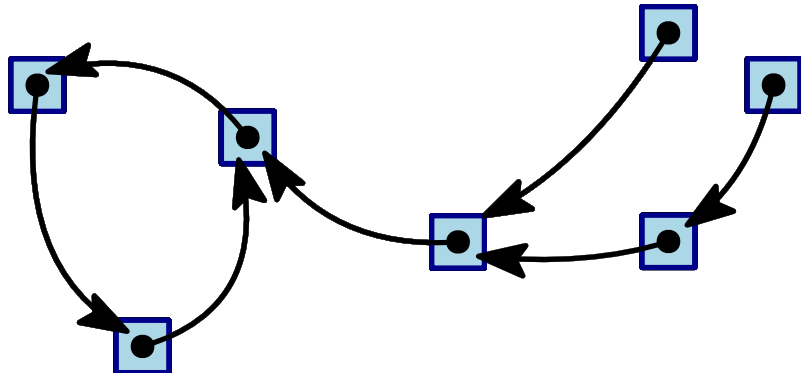
Simple Cell Mapping (SCM) algorithm

- Data structure of a cell:

```
struct Cell {
    unsigned int id;
    unsigned int group;
    Cell_type type;
    Cell_state state;
};
```

periodic/transient

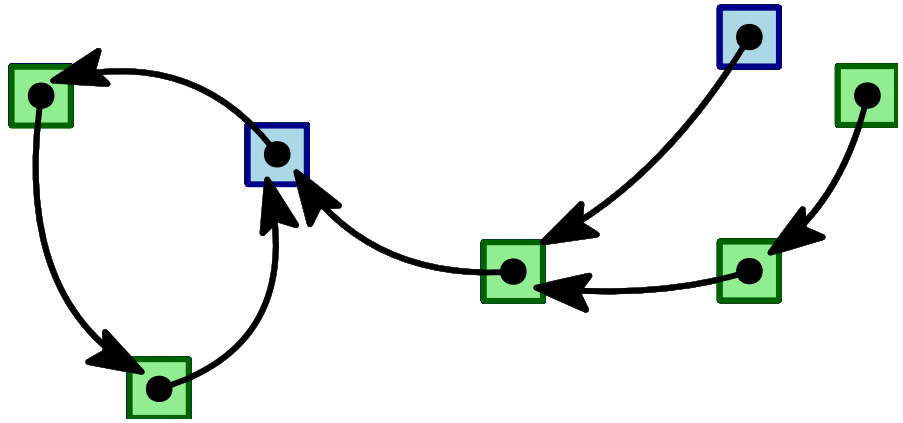
solved/under processing/unsolved



1. Start a cell sequence starting from an unsolved cell
2. For each cell in the sequence:
   - If unsolved: mark as under processing
   - If solved: append the new sequence to the previous solution
   - If under processing: new periodic group found

Challenges in parallellization

- **Idea: start solutions from multiple cells**
- Race conditions: solutions may merge
  - Memory access? (writing conflicting data to the same location)
  - Performance (more threads working on the same sequence)
  - Conflicting group IDs



Parallelization of the SCM algorithm

- Modified cell representation structure:

```
struct Cell {
    unsigned int id;
    unsigned int group;
    Cell_type type;
    Cell_state state;
};
```

```
struct Cell {
    Cell_type type;
    unsigned int group;
    Cell_state state[N_THREADS];
};
```

- Considerations:
  - Threads share the same cell array
  - Cell IDs are the array indices
  - Only the solution states are stored thread-by-thread
  - Group ID

Implementation in OpenCL

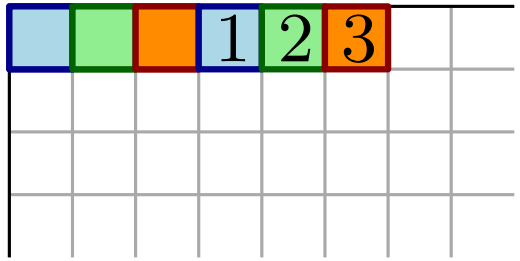
```
__kernel void solve(__global struct Cell *cells){
    // get global ID
    size_t gid = get_global_id(0);

    // calculate SCM solution
    unsigned int start = gid;
    unsigned int end = N_CELLS;
    for(unsigned int z=start; z<end; z+=N_THREADS){
        // ...

                    mem_fence(CLK_GLOBAL_MEM_FENCE);

        // ...
    }
}
```

- Cells in global memory
- Each work-item starts from a different cell
- Scanning through the cell state-space
- Synchronization during solving



Results

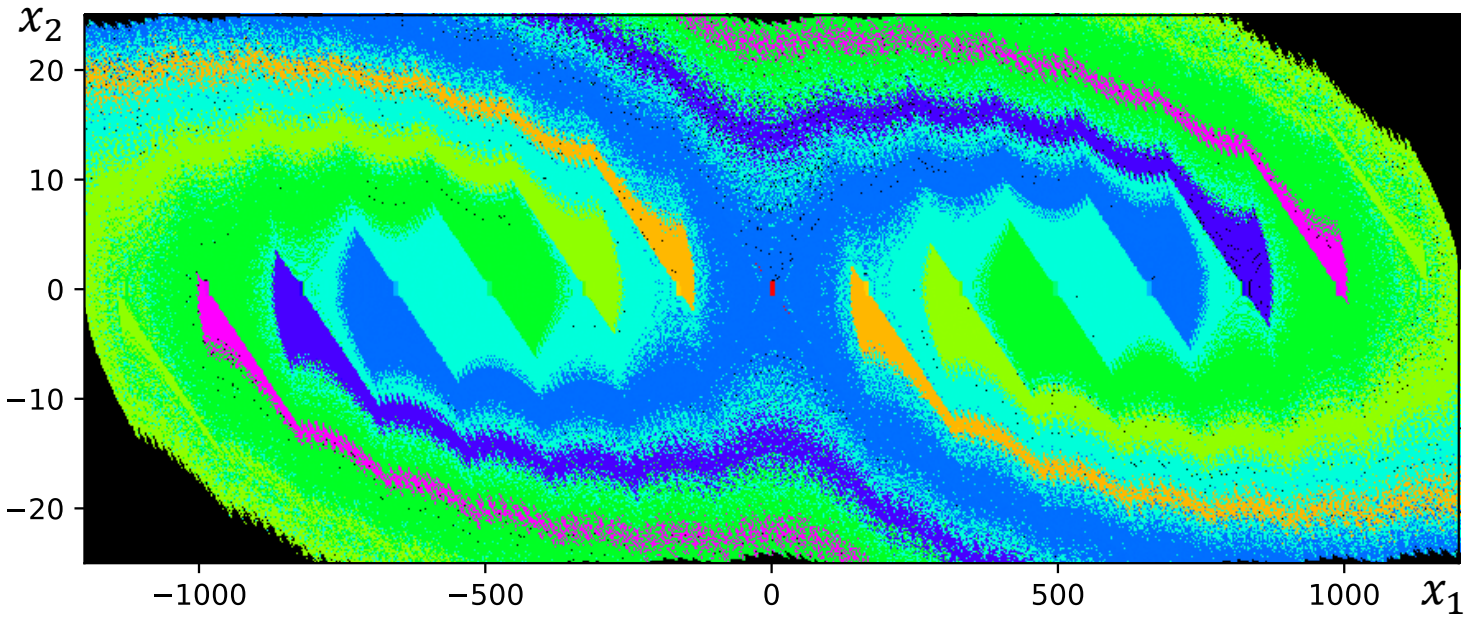
- Micro-chaos map
  - State-feedback: $u[k] = \mathrm{trunc}(-\mathbf{K}\,\mathbf{x}[k])$
  - State equation: $\mathbf{x}[k+1] = \mathbf{A}\,\mathbf{x}[k] + \mathbf{B}\,u[k]$
- SCM on a 1920×1080 cell state-space:

Parameters:

$$\mathbf{A} = \begin{bmatrix} 1{,}00304 & 1{,}00101 \\ 0{,}00609 & 1{,}00304 \end{bmatrix};$$

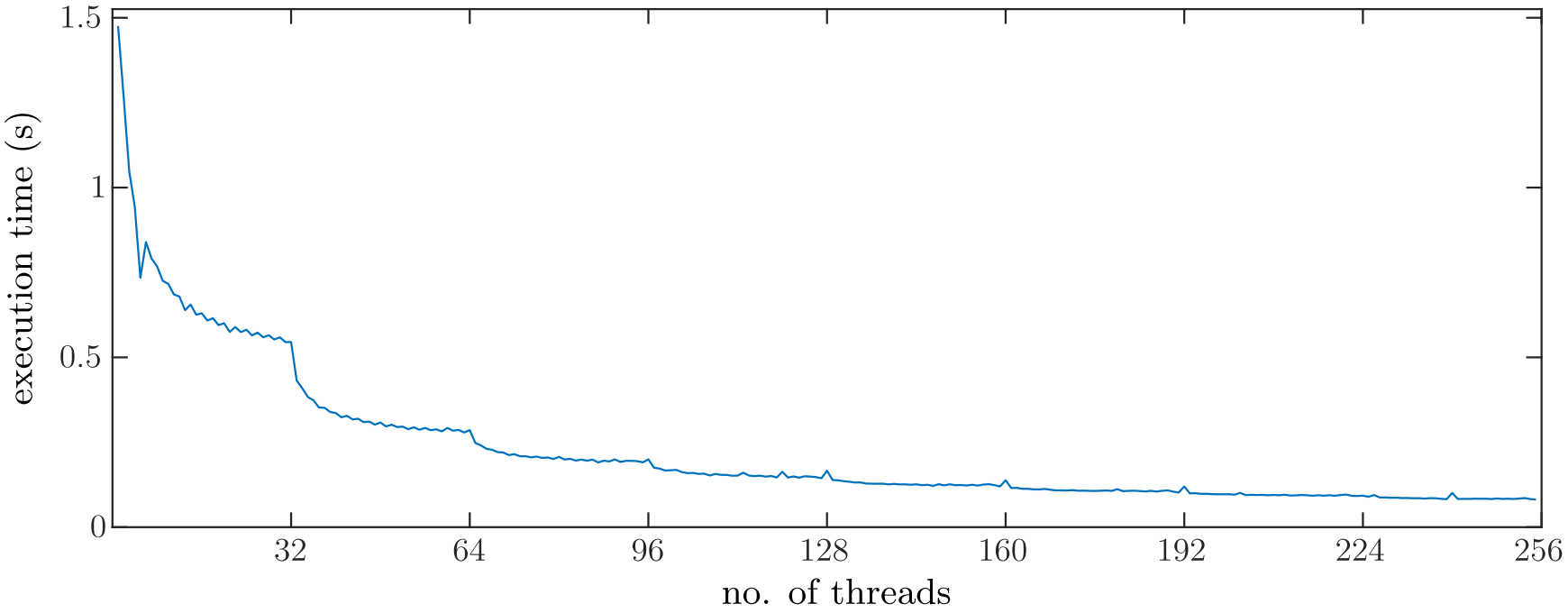
$$\mathbf{B} = \begin{bmatrix} 0{,}50025 \\ 1{,}00101 \end{bmatrix};$$

$$\mathbf{K} = [0{,}007 \quad 0{,}020]$$



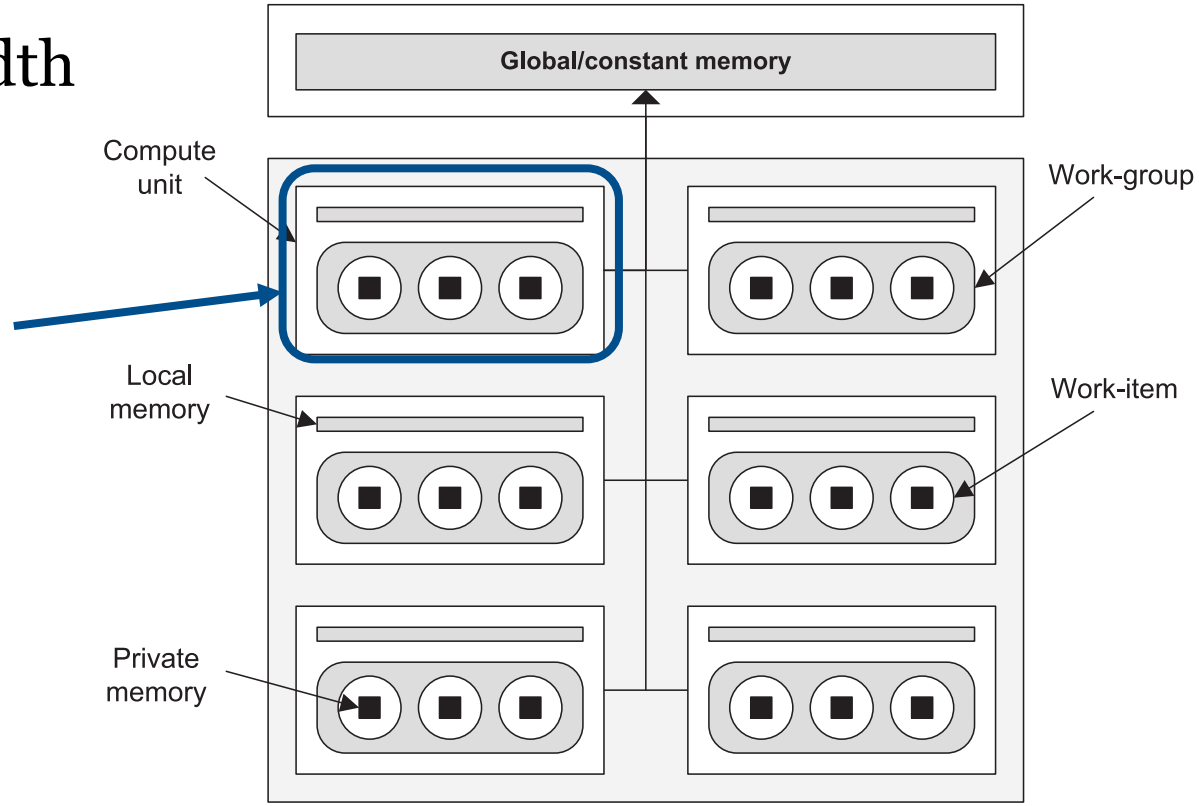
Results

- GPU: NVIDIA RTX 3050 Ti
- Scalability:

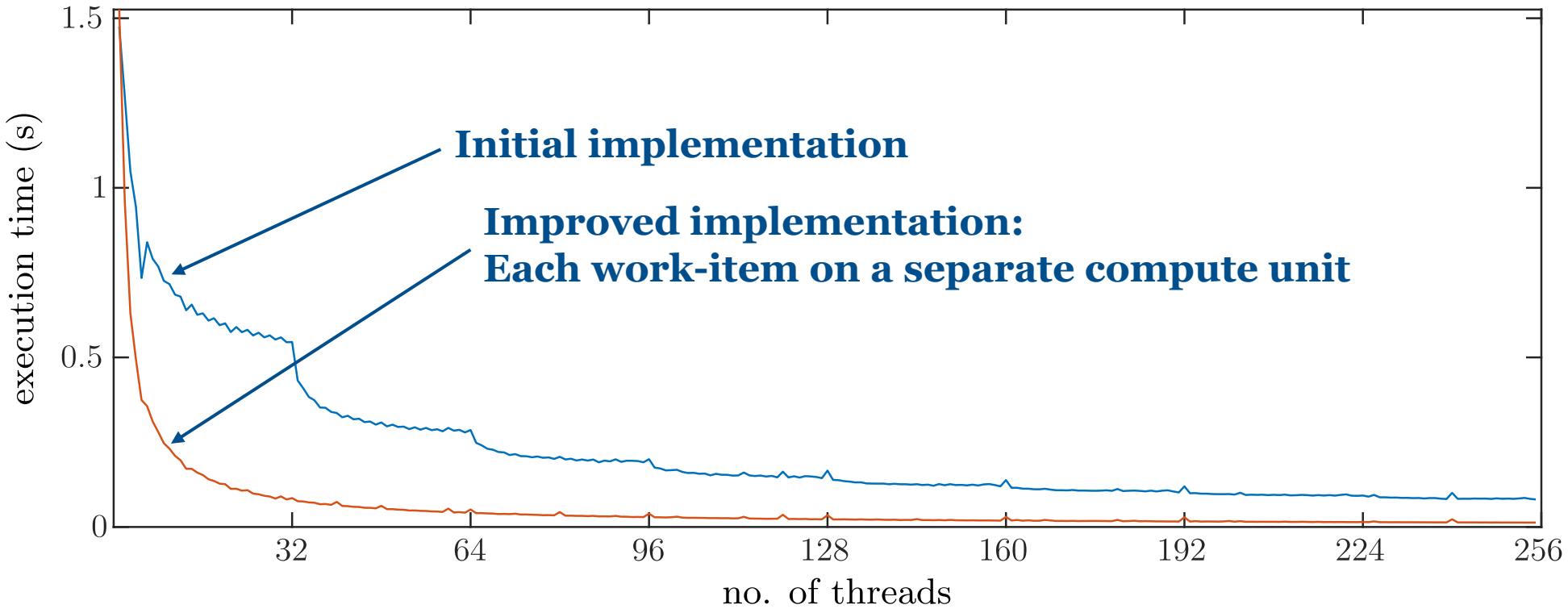


Further challenges

- Memory access
  - Global memory: limited bandwidth
  - Random access
- Thread divergence
  - Same WG, same Compute Unit
  - Shared program counter
  - Threads block each other at branches
  - Experiment: assign each Work-item to a separate Compute Unit

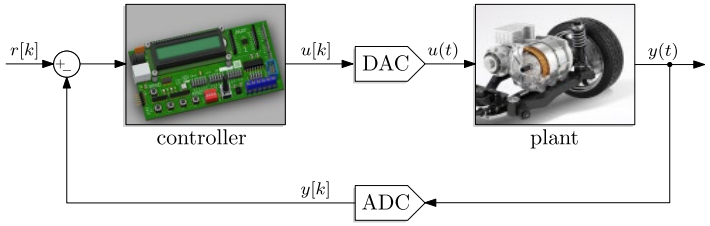


Results

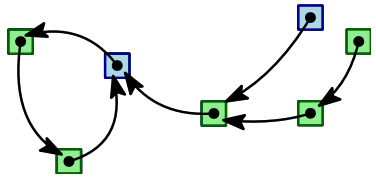

1.5
1
0.5
0
execution time (s)
32
64
96
128
160
192
224
256
no. of threads
Initial implementation
Improved implementation:
Each work-item on a separate compute unit

Conclusions

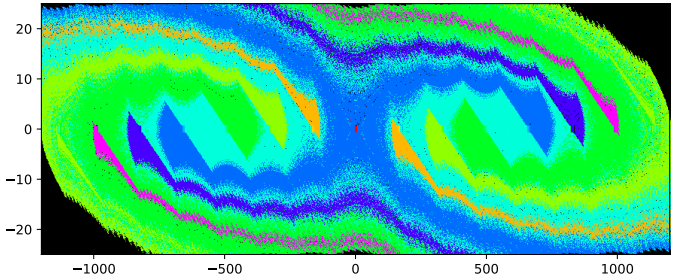
- HPC methods for analysis of mechatronic systems
- Goal: global dynamical analysis
- Parallelization of the Simple Cell Mapping (SCM) algorithm
- GPU implementation
- Test results on the micro-chaos map



Mechatronic system



GPU implementation of the SCM algorithm



Testing on the micro-chaos map

Thank you for your attention!

GPU acceleration of the Simple Cell Mapping algorithm

Tamás Haba, Csaba Budai

MINISTRY OF CULTURE AND INNOVATION

NATIONAL RESEARCH, DEVELOPMENT AND INNOVATION OFFICE HUNGARY

Új Nemzeti Kiválóság Program

Supported by the **ÚNKP-23-3-1-BME-63** New National Excellence Program of the Ministry for Culture and Innovation from the source of the National Research, Development and Innovation Fund.