An Efficient Parallel Algorithm for Numerical Solution of Low Dimension Dynamics Problems

S.G. Orlov, A.K. Kuzin, N.N. Shabrov

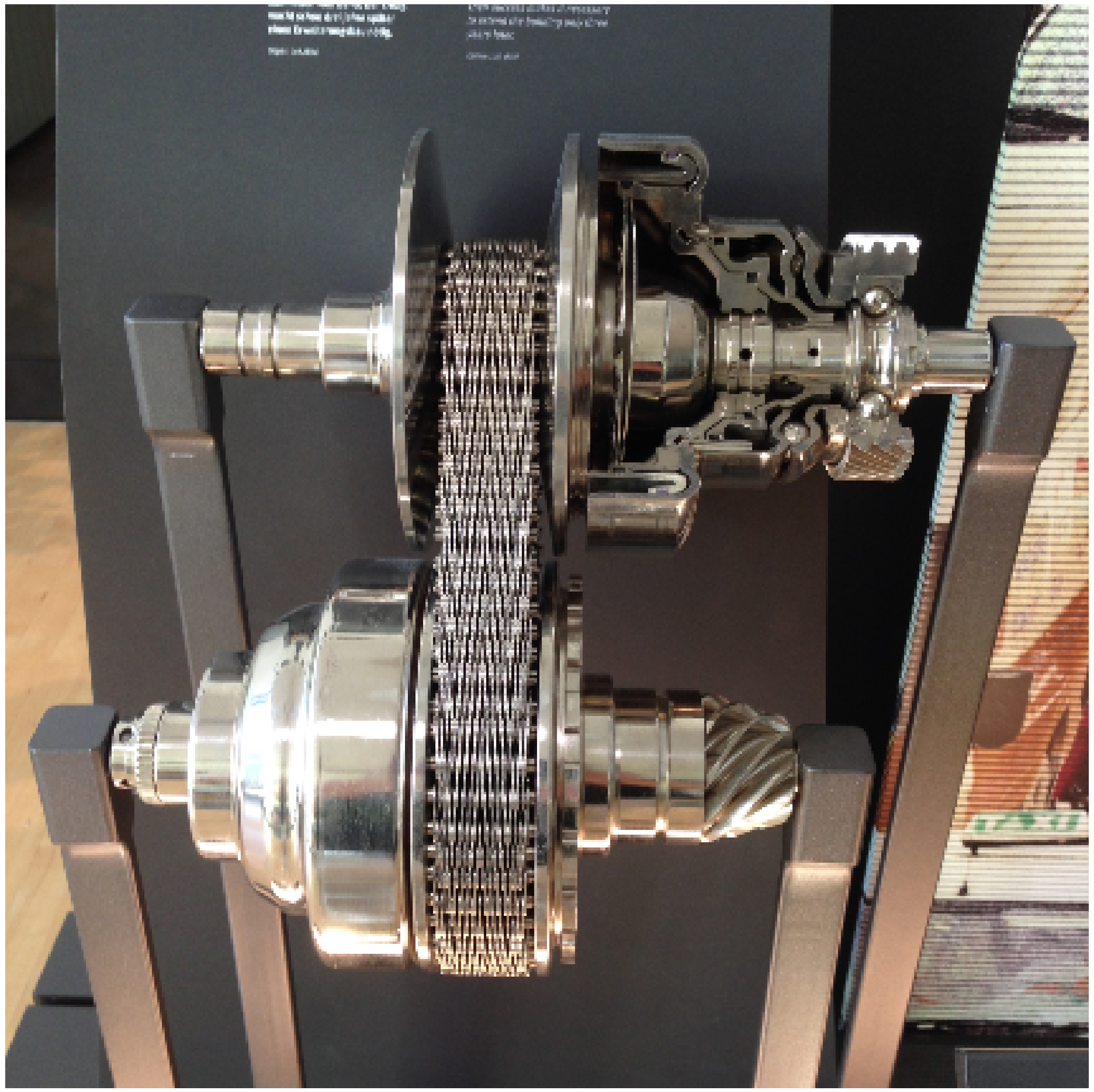
Computer technologies in engineering dept.

Peter the Great St. Petersburg Polytechnic University

Russian Supercomputing Days
Sept. 24-25 2018, Moscow, Russia

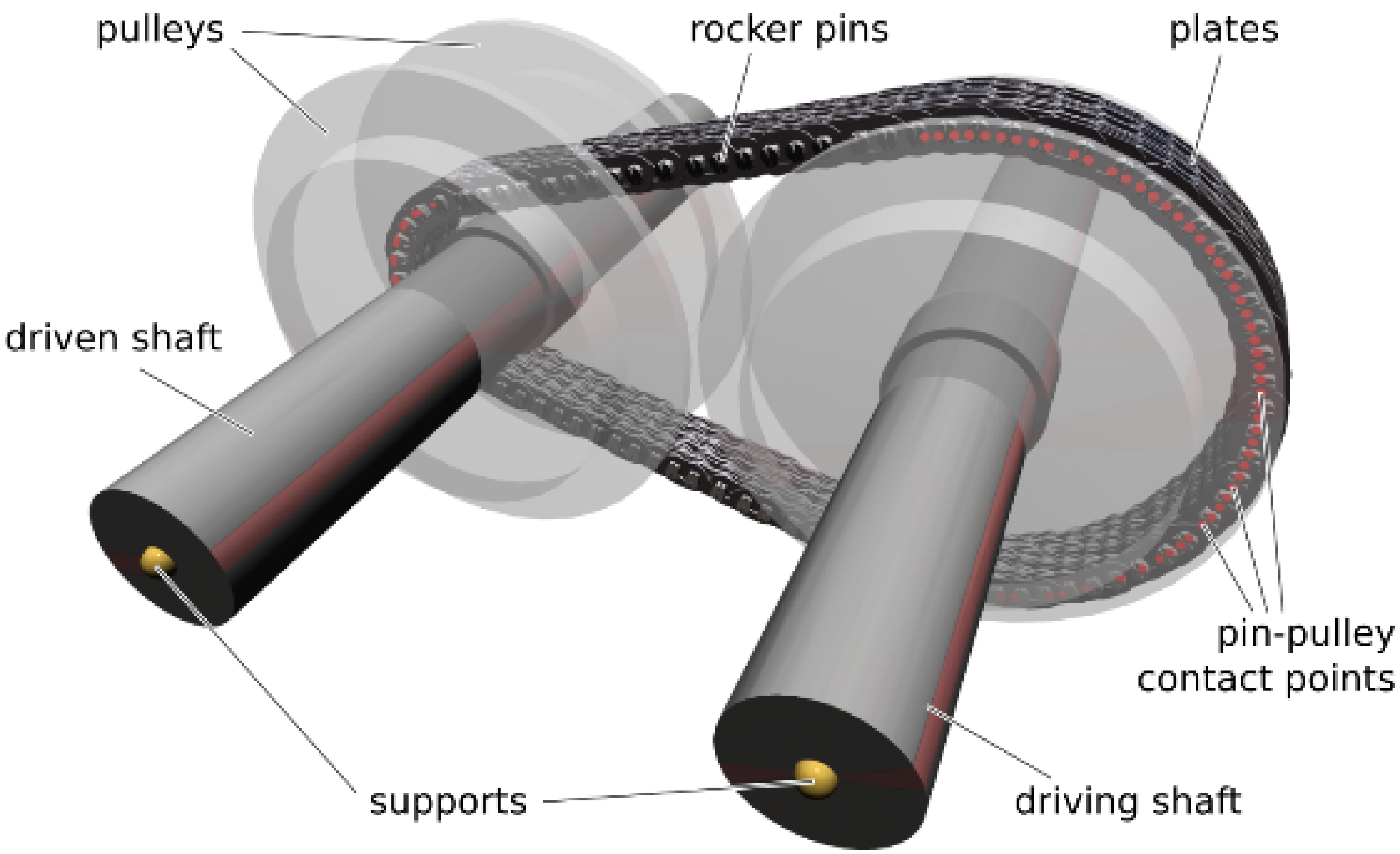
Model overview

Real device

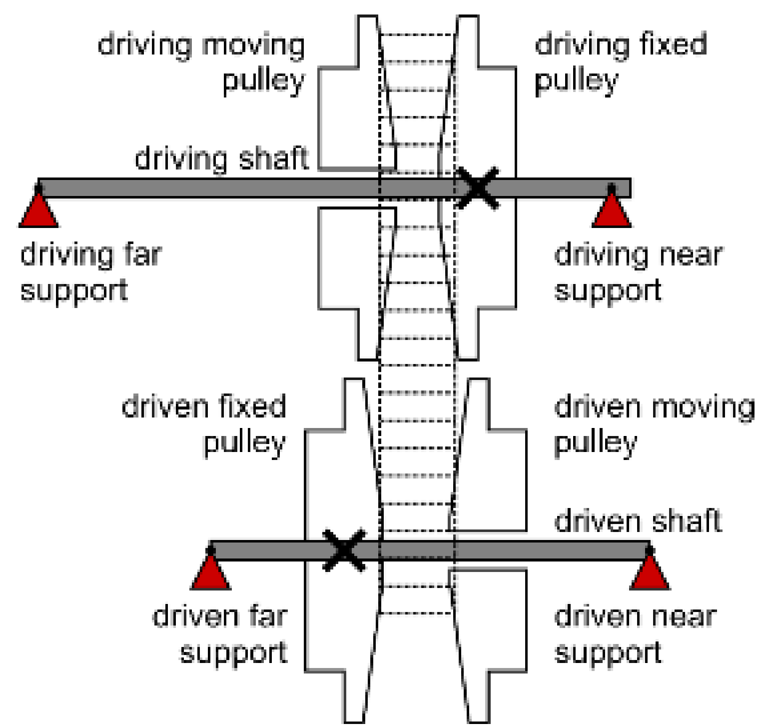


Model overview

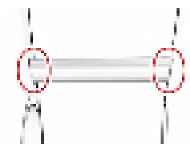
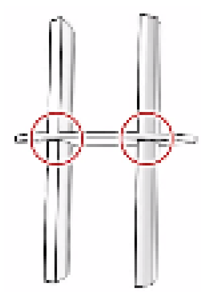
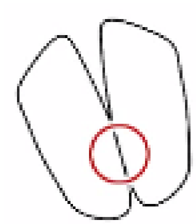
3D view

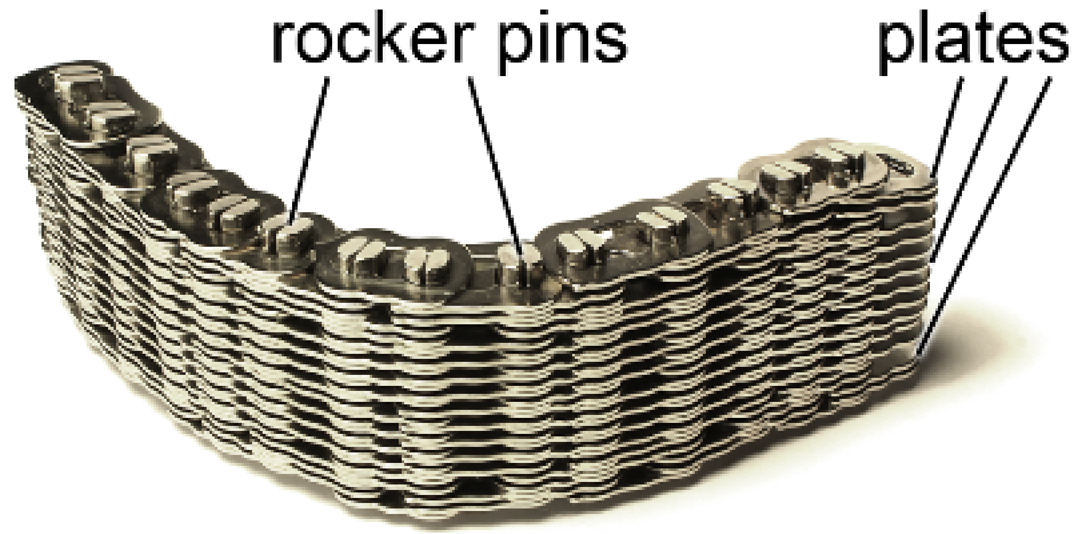


top view



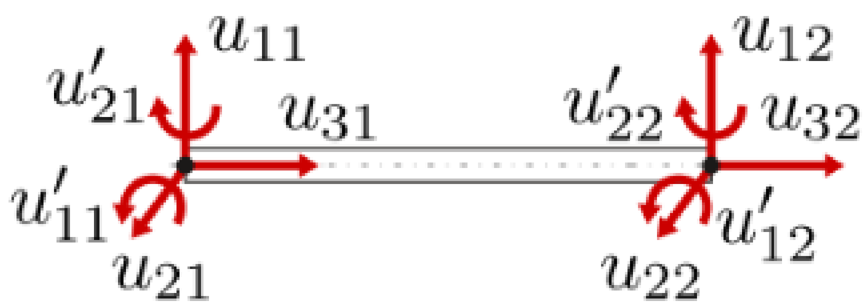
Model overview

- The chain consists of plates and rocker pins
- Each pin has two halves rolling over each other
- There are many contact interactions
  - pin — pulley 
  - pin — plate 
  - pin — pin 

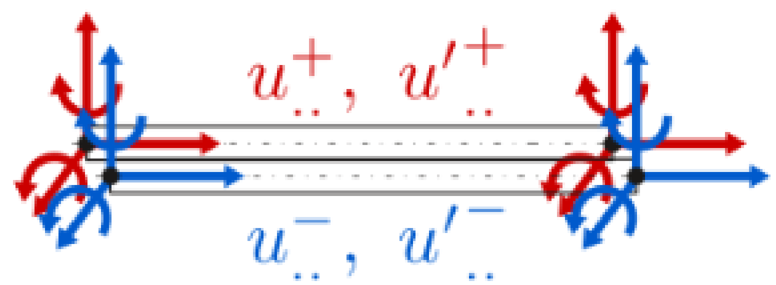


Model overview

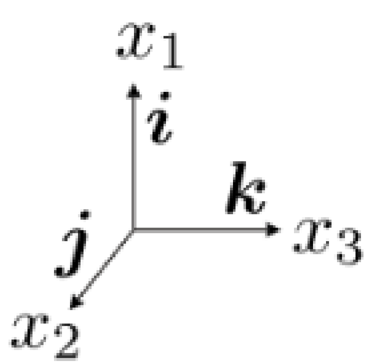
21 generalized coordinates per chain link



pin half



pin



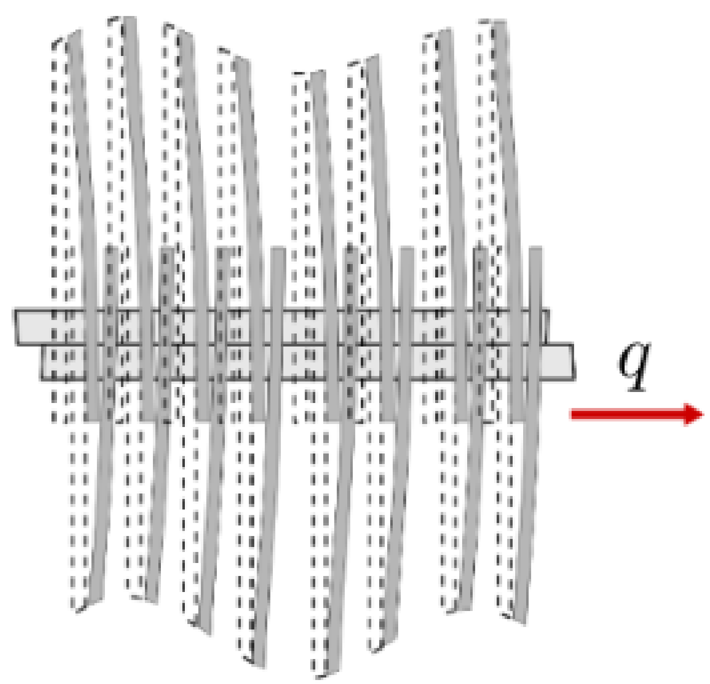


plate pack

Equations of motion

- Lagrange equations: $\frac{d}{dt}\frac{\partial L}{\partial \dot{q}} - \frac{\partial L}{\partial q} = \tilde{Q}$
- lead to $\mathbf{A}(q)\,\ddot{q} = F(t, q, \dot{q}\,)$
  - The inertia matrix $\mathbf{A}$ is sparse block-diagonal
  - Sometimes it really depends on q
- In the normal form, ODE system is $A\dot{x} = f(t, x)$
  - $\begin{matrix} q \equiv u \\ \dot{q} \equiv v \end{matrix}, \quad A = \begin{bmatrix} I & 0 \\ 0 & \mathbf{A} \end{bmatrix},$
  - $x = \begin{bmatrix} u \\ v \end{bmatrix}, \quad f = \begin{bmatrix} v \\ F(t, u, v) \end{bmatrix}$

Integration Scheme

- Solving IVP for

$$A\dot{x} = f(t,x) = [v, F(t,u,v)]^T \quad \Rightarrow \quad \dot{x} = \tilde{f}(t,x)$$

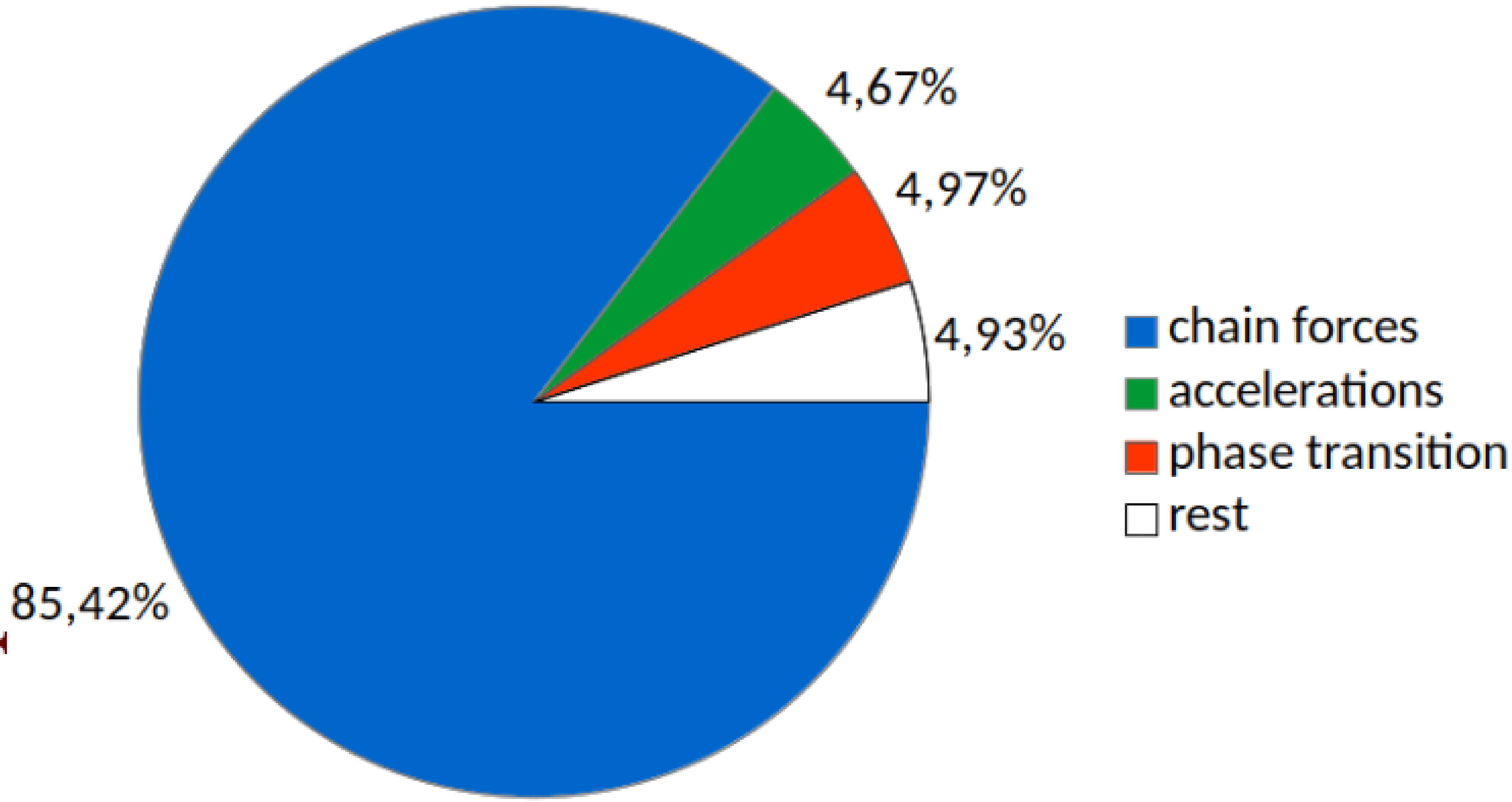
- Currently using explicit RK4 scheme

$$\begin{aligned}
k_1 &= \tilde{f}\left(t^{(n)}, x^{(n)}\right), \\
k_2 &= \tilde{f}\left(t^{(n)} + \tfrac{h}{2}, x^{(n)} + \tfrac{h}{2}k_1\right), \\
k_3 &= \tilde{f}\left(t^{(n)} + \tfrac{h}{2}, x^{(n)} + \tfrac{h}{2}k_2\right), \\
k_4 &= \tilde{f}\left(t^{(n)} + h, x^{(n)} + hk_3\right), \\
x^{(n+1)} &= x^{(n)} + \tfrac{h}{6}\left(k_1 + 2k_2 + 2k_3 + k_4\right).
\end{aligned}$$

- Each evaluation of $\tilde{f} \Rightarrow$ one evaluation of $F(t,u,v)$ and Cholesky reverse pass for A
- Each integration step ends with phase transition procedure

Parallelization

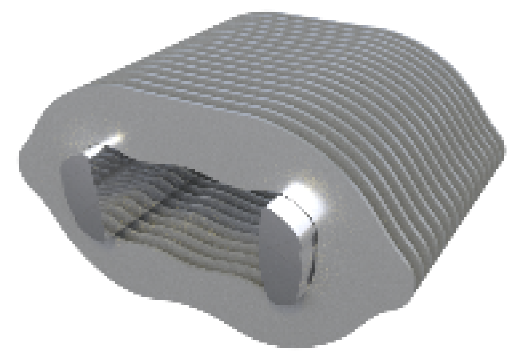
Time distribution in sequential code

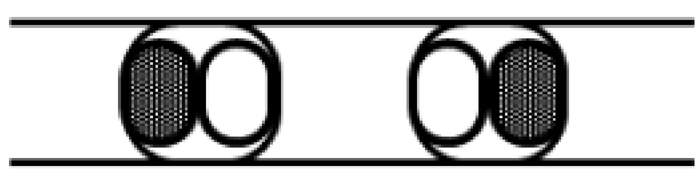


Parallelization

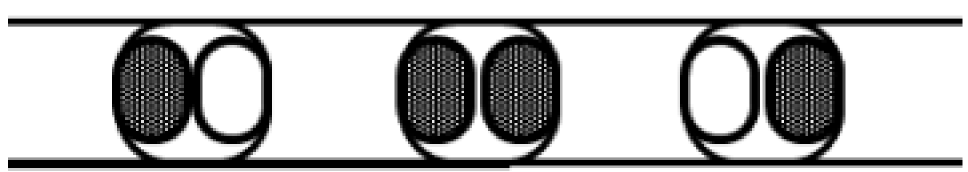
of chain forces calculation

Possible parallelization grains arise from forces kinds:





- Link forces
  - Forces due to link plates deformations

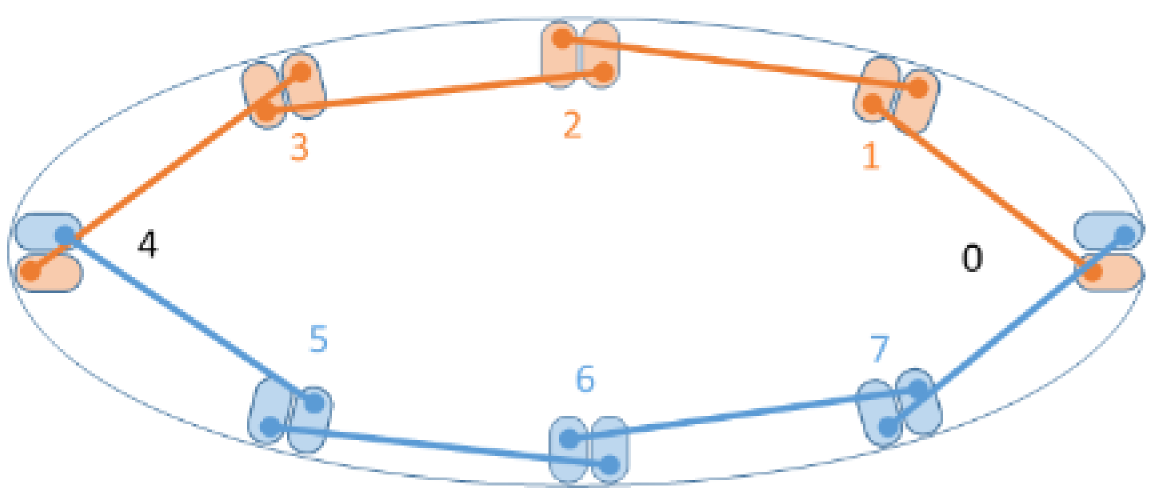


- Pin forces
  - Contact interactions between pin halves

Parallelization

of chain forces calculation

Workload distribution in the case of 2 threads



Thread 1	Thread 2
Independent calculation of pin/link forces for pins:	
1, 2, 3	5, 6, 7
Barrier	
Independent calculation of pin forces for:	
Pin 0	Pin 4

Simulation Results

CVT model with chain containing 84 pins considered.

The resulting system of ODEs consists of approx. 3600 equations.

$2 * 10^{-3}$ s of model time require 7min.22s of processor time in sequential mode.

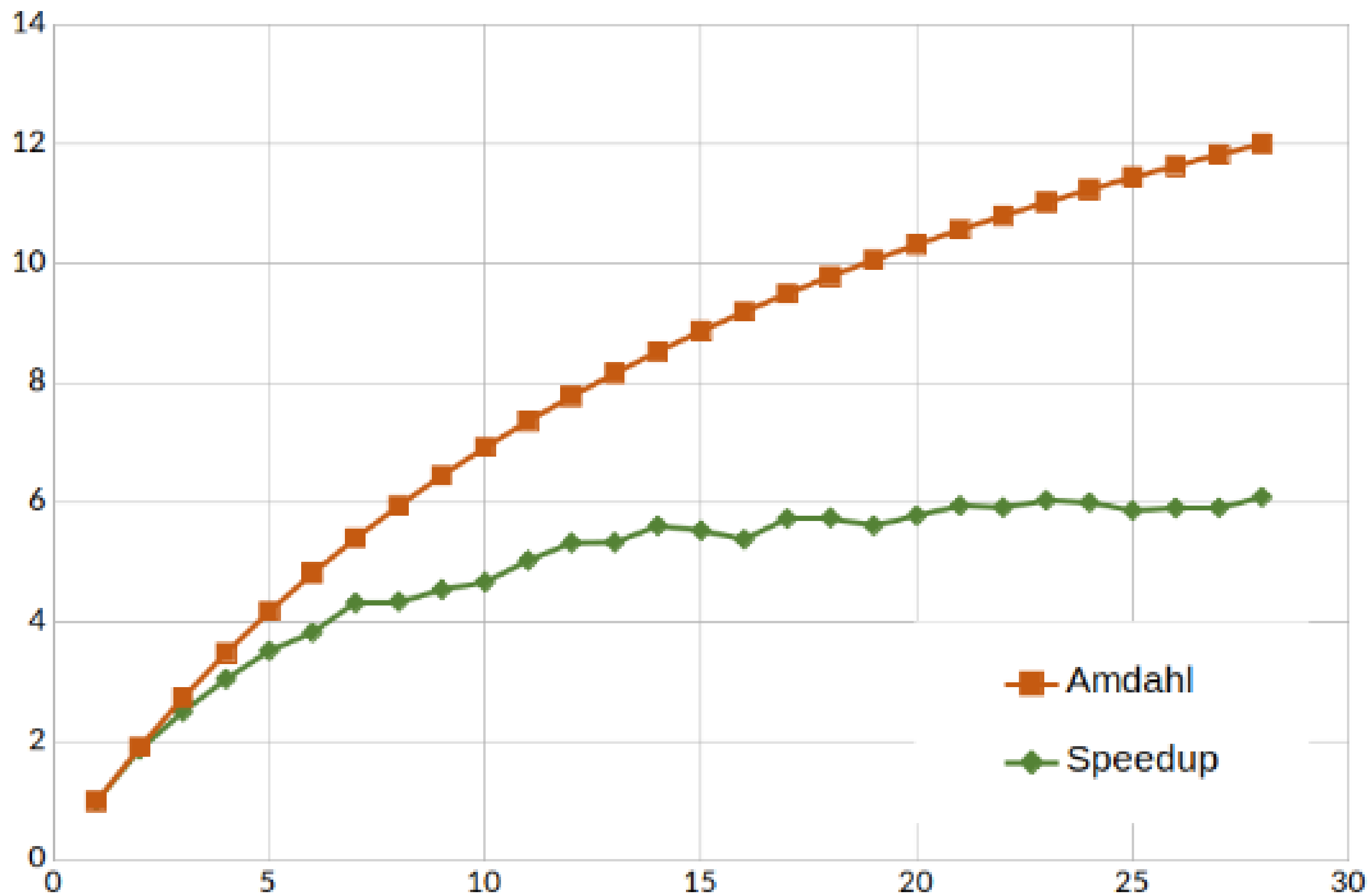
"Polytechnic RSK Tornado"

of Supercomputer Center Polytechnic of SPbPU

Cores per socket	14
NUMA Nodes	2
CPUs	Intel Xeon CPU E5-2697 v3 2.60GHz
Linux	CentOS Linux release 7.0.1406 (Core)
C++ compiler	Intel 2017.5.239

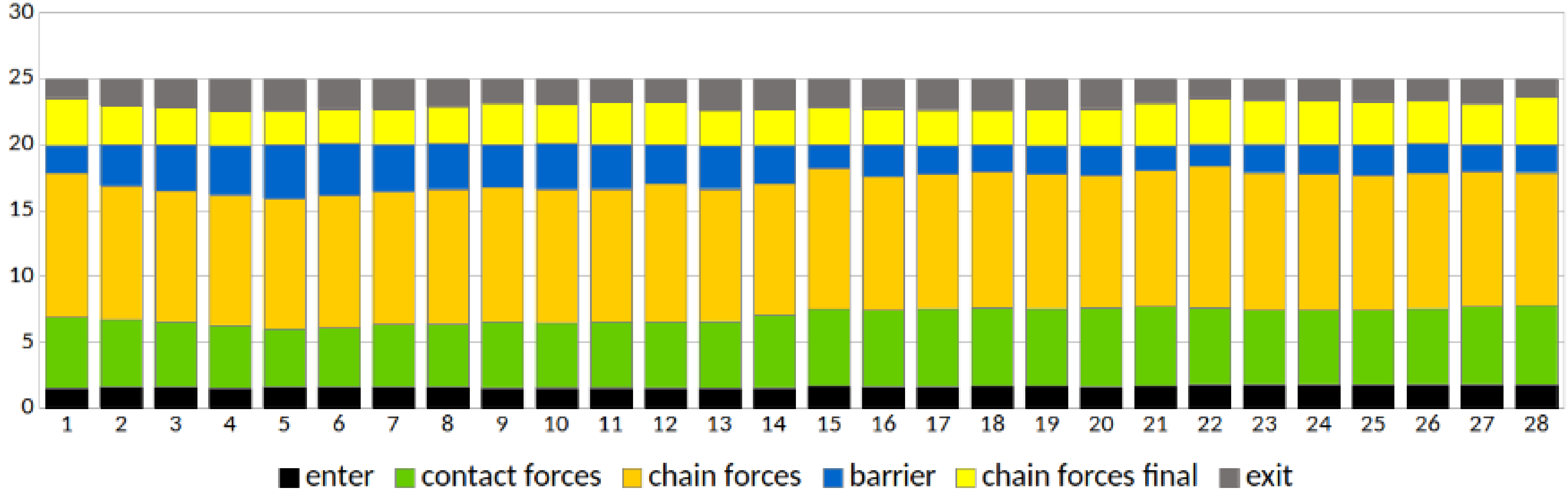
Parallelization

The speedup of the whole simulation as a function of thread count



Amdahl's law curve corresponds the fraction of serial work equal to 5%

Parallelization



- There is imbalance between threads in contact forces calculation
- Significant piece of time is spent in OpenMP code:

enter	7%
barrier	11%
exit	8%

Conclusions

- Maximal speedup of the whole simulation equal to 6 achieved for given hardware
- Significant scalability can be achieved only up to 10-12 threads. Further thread count increment leads to saturation.
- One of the sources of load imbalance is contact forces calculation.
- Significant overhead time presents at the start and the end of OpenMP section when the number of threads is large.

Thank you