University logo

# Modelling of dynamic network objects: new approaches and adaptation challenges for future HPC systems

author name + institute

## PhD focus

- Parallelization approaches
- Adaptation to heterogeneous parallel computing architectures
- Code optimization
- Solvers based on parallel numerical methods – block difference numerical methods (BDM)
- Simulation model development with DSL

## Introduction

- Complex dynamic systems are considered as objects of research, design, automatization, monitoring and control in many subject areas: mechanical engineering, coal mining, metallurgy and other
- Modelling and simulation are needed
- Typically such systems could be represented as dynamic network objects (DNO) – description via graphs, models with different complexity levels (concentrated or distributed parameters)
- Because of computational complexity HPC resources are used for simulation
- High heterogeneity of modern HPC resources allows to use different programming models and so to make modelling process maximum efficient

## Challenges

- Parallelization of common used sequential solvers or even their replacement with new and parallel ones
- Code optimization and adaptation to different heterogeneous hardware
- Simplification of the model description and configuration process for the user, who is usually an expert in specific subject area, but not a software/hardware specialist

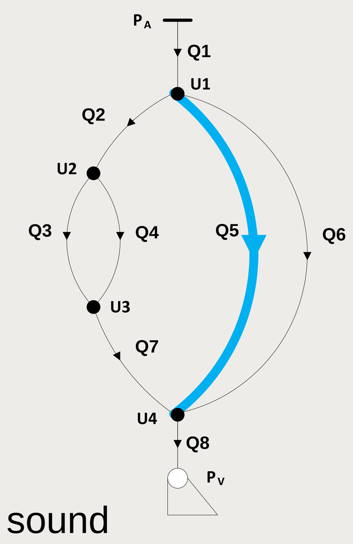
## Proposed approach

- New parallel solvers based on block difference numerical methods (BDM)
- Different code optimisation techniques and other tuning mechanisms
- Usage of domain specific languages (DSL) to separate model description and specific hardware optimisation parts

## Simulation model

- Coal mine air ventilation system
- DNO with distributed parameters (DNODP)
- Test model: 8 branches, 6 nodes
- Equation system for each *j*-branch:

$$\begin{cases} -\dfrac{\partial P_j}{\partial \xi} = r_j Q_j^2 + \dfrac{\rho}{F_j}\dfrac{\partial Q_j}{\partial t} + r_j(\xi_p, t) Q_j^2 \\ -\dfrac{\partial P_j}{\partial t} = \dfrac{\rho a^2}{F_j}\dfrac{\partial Q_j}{\partial \xi} \end{cases}$$



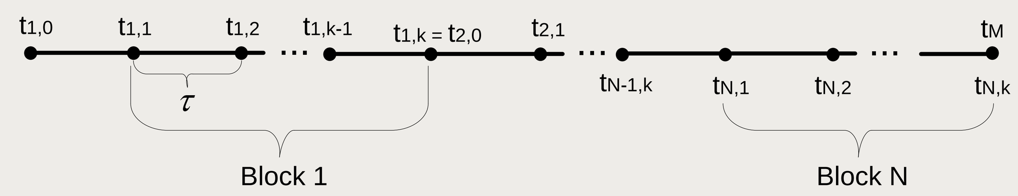

$P_A$ – atmosphere pressure
$P_V$ – ventilator pressure
$P_j$, $Q_j$ – pressure, airflow
$r$ – flow resistance
$F_j$ – cross-section area
$\rho$ – air density
$a$ – speed of sound
$r_j(\xi_p,t)$ – adjustable resistance (on $\xi_p$-coordinate)

Boundary values (for each *wi*-node)

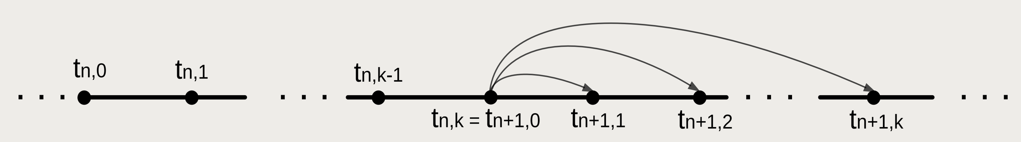
- Internal $\quad -\dfrac{\partial P_{wi}}{\partial t} = \dfrac{\rho a^2}{F_{wi}}\dfrac{\partial Q_{wi}}{\partial \xi}$
- ventilators in *j*-branch (active elements) $\quad P_{wi} = P_{AEJ}(Q_J)$
- atmosphere $\quad P_{wi} = P_{ATM} = const$

## Block difference methods

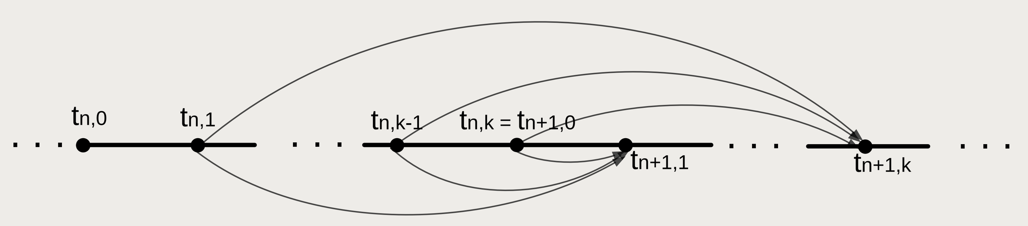
- Cauchy-problem: $x' = f(t,x), \quad x(t_0) = x_0$
- BDM: general decomposition schema (N blocks, k points)




- BDM: 1-step methods



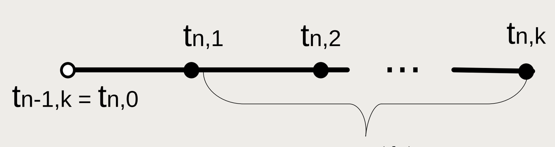

- BDM: multi-steps methods




- BDM: general formula (m-steps, k-points)

$$u_{n,i} = u_{n,0} + i\tau\left[\sum_{j=1}^{m} b_{i,j}F_{n,j-m} + \sum_{j=1}^{k} a_{i,j}F_{n,j}\right] \qquad i=\overline{1,k} \quad n=\overline{1,N} \quad F_{n,j} = f(t_n + j\tau, u_{n,j})$$

- **BDM4 (m=1, k=4) – PhD research focus**



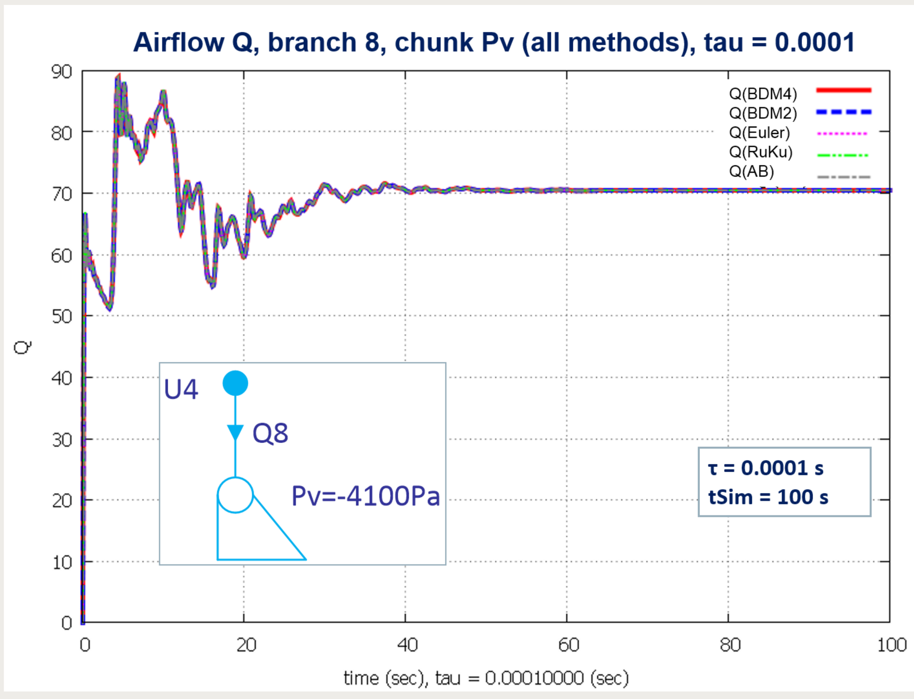

$$T_n^{(k)}$$

$N$ – blocks
$F$ – right side of equation
$a,b$ – coefficients
$\tau$ – time step
$i$ – point in block

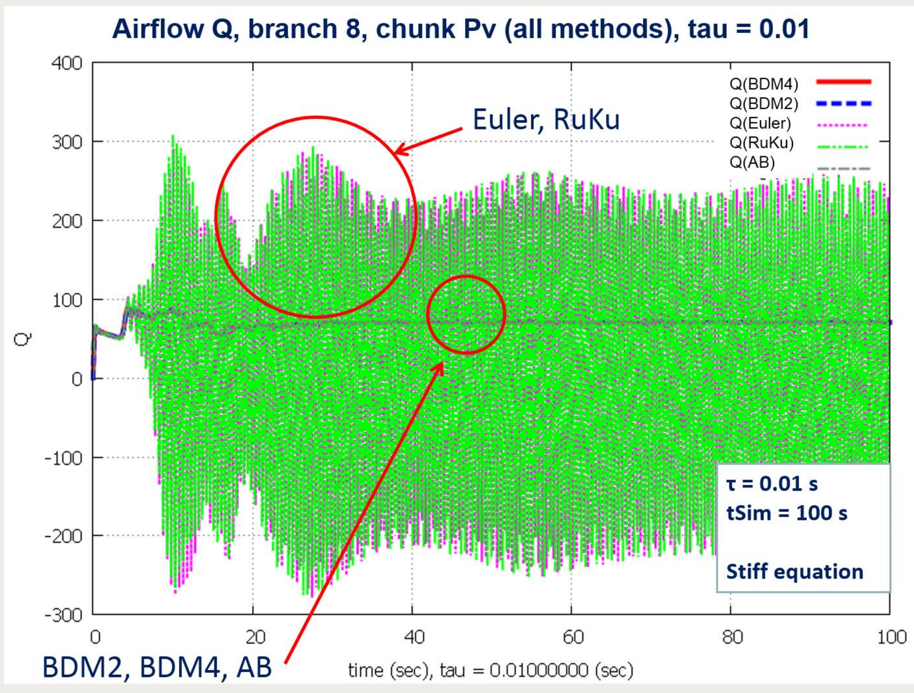
$$u_{n,i} = u_{n,0} + i\tau\left[b_i F_{n-1,0} + \sum_{j=1}^{k} a_{i,j}F_{n,j}\right] \qquad i=\overline{1,k} \quad n=\overline{1,N}$$

## First experiments, BDM

- DNODP, sequential solver
- Numerical methods:
  AB (Adams-Bashforth), RuKu (Runge-Kutta), EU (Euler), BDM2 (2 points), BDM4 (4 points)
- Time step: tau = 0.0001 s




- Time step: tau = 0.01 s




- Critical time step, computational time

| | $Tau_{critical}$, s | Computational time, s | Steps | Max iterations (for implicit BDM methods) |
|---|---|---|---|---|
| Eu | 0.0001 | 3.96 | 1000000 | |
| AB | 0.01 | 0.06 | 10000 | |
| RuKu | 0.0001 | 14.89 | 1000000 | |
| BDM2 | **0.03** | **0.14** | 1667 | 9 |
| BDM4 | **0.03** | **0.23** | 834 | 6 |

## Conclusion

**Block difference methods provide:**

- ➢ High accuracy, convergence
- ➢ Less computational steps
- ➢ Good relation “accuracy-speed”

**Further investigation, parallelization:**

- Parallelization of the BDM-solvers using diverse programming models
- Parallelization on the graph level varying the granularity of parallel processes
- Different code optimization techniques

**Further investigation, DSL:**

- Usage of AnyDSL framework as starting point (with “Impala” language inside)
- Creation of DNO-model in DSL, adaptation to different heterogeneous architectures
- Research in the direction of automatic model optimization and adaptation

Author: name + email