Available online at www.sciencedirect.com

ScienceDirect

IFAC Papers Online
CONFERENCE PAPER ARCHIVE

IFAC-PapersOnLine 48-1 (2015) 260–261

A joint IMEX-MOR approach for Water Networks

Sara Grundel * **Lennart Jansen** **

* *Max-Planck Institute for Dynamics of Complex Technical Systems, Magdeburg, Germany (e-mail: grundel@mpi-magdeburg.mpg.de).*
** *Heinrich Heine Univeristät Düsseldorf, Germany (e-mail: lennart.jansen@uni-duesseldorf.de)*

Abstract: Modeling and Simulation of fluids in large network is a rather challenging problem. We provide an approach combining techniques in Model Order Reduction (MOR) and implicit-explicit (IMEX) integration to create efficient and stable simulations.

Keywords: Model Reduction, Integration, Differential Equations, Networks, Directed Graphs

1. INTRODUCTION

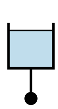
The simulation of fluids within a large network of pipes poses several mathematical challenges. Typically after spatial discretization the resulting mathematical system is a nonlinear differential algebraic system. Standard techniques are often slow due to the stiffness of the equation. We will show a several step process on how to improve on the timing. A first and major step in order to achieve stable and fast simulators for these problems is what we call the decoupling step. In that step we are able to model the system as a discrete index 1 DAE. This step is only possible due to the choosen discretization we use. Next we use a combination of Model Order Reduction (MOR) methods in order to create a smaller scale index 1 Differential Algebraic Equation. And last but not least we use an implicit-explicit (IMEX) integration method to reduce the time-step for the stiff nonlinear differential equation. We will only present a simplified network here which includes pipes, reservoirs and so called demand nodes. This system will actually result in an ODE which simplifies the discussion.

2. MODELING

It is common to define a connected and directed simple graph $G = G(V,E)$ representing the pipe network. This allows a more compact representation of the model equations. The set V are the nodes and E are the edges and we will describe the different node and edge elements in the following.

2.1 Node Elements

Reservoirs are water sources with unlimited capacity. Thus, we assume that they have a constant pressure p_s. Furthermore no balance equation holds at a reservoir, since an arbitrary amount of water may leave or enter the reservoir.



$$p = p_s$$

In contrast to reservoirs, **tanks** have limited capacity. Never the less, pressure can in- or decrease even though the tank is full or empty respectively. We will not talk about tanks in more details here.

A **demand node** has a given demand $q_s\colon I \longrightarrow \mathbb{R}_+$. Thus, the difference between the amount of water flowing towards a node, and the amount of water flowing away from the node has to be q

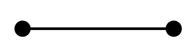
$$\sum_{i \in I_{in}} q^i_{in} - \sum_{i \in I_{out}} q^i_{out} = q_s(t)$$

with q^i_{in} and q^i_{out} being the incoming and outgoing flow of edges connected to the demand node, respectively. It is possible that $q_s(t) = 0$ for all $t \in I$.

2.2 Edge Elements

First we will discuss the **pipe** model. The behavior of a pipe is described by a continuity equation and an equation describing the movement inside a pipe. We consider circular pipes with diameter D, cross-section $A = \frac{\pi}{4}D^2$ and length ℓ. The independent variables are space $x \in [0,\ell] := \Omega \subset \mathbb{R}$ and time $t \in [t_0, T] := I \subset \mathbb{R}$. The time dependent variables are the mass flow $m\colon \Omega \times I \longrightarrow \mathbb{R}$ and the pressure $p\colon \Omega \times I \longrightarrow \mathbb{R}$. The parameters a, ρ and c depend on material properties of the pipe and the gas. α is the angle of the pipe and g is the gravitational constant. With this, we get the following partial differential equation, which describes the behavior in pipes

$$\begin{aligned} \frac{\partial p}{\partial t}(x,t) + \frac{a^2}{A}\frac{\partial m}{\partial x}(x,t) &= 0 \\ \frac{\partial m}{\partial t}(x,t) + A\frac{\partial p}{\partial x}(x,t) + \rho A g \sin\alpha + c|m(x,t)|m(x,t) &= 0 \end{aligned} \quad (1)$$

pipe:  Hyperbolic PDE

Further edge elements are **valves** and **pumps**, which we also omit in this extended abstract.

2.3 Network Model

From now on we consider a network with n_p many pipes and n_d many demand nodes and n_{rs} many reservoirs. For each pipe i, we get a flow $m_i\colon \Omega_i \times I \longrightarrow \mathbb{R}$, $\Omega_i = [x_{L_i}, x_{R_i}] \subset \mathbb{R}$, $I = [t_0, T]$ and a pressure $p_i\colon \Omega_i \times I \longrightarrow \mathbb{R}$ both depending on space $x \in \Omega_i$ and time $t \in I$. The node variables are $p_d\colon I \longrightarrow \mathbb{R}^{n_d}$ and

Peer review under responsibility of International Federation of Automatic Control.
10.1016/j.ifacol.2015.05.201

$p_{rs}\colon I \longrightarrow \mathbb{R}^{n_{rs}}$, the pressures at demand nodes and reservoirs. We define

$$m_L(t) = (m_i(x_{L_i},t))_{i\in\mathscr{A}_{pi}}, m_R(t) = (m_i(x_{R_i},t))_{i\in\mathscr{A}_{pi}}.$$
$$p_L(t) = (p_i(x_{L_i},t))_{i\in\mathscr{A}_{pi}}, p_R(t) = (p_i(x_{R_i},t))_{i\in\mathscr{A}_{pi}}.$$

m_L being the vector with all pipe flows at their tail-node and m_R the flow vector at their head-nodes and similarly for p_L and p_R. Note, that the node pressures coincide with the head- and tail pressures of the pipes. We call the vector of demand and reservoir pressures by p.

Last we define the following incidence matrices

$$A_R^{rs} \in \mathbb{R}^{n_p\times n_{rs}} (A_R^{rs})_{ij} = \begin{cases} 1 & \text{if reservoir node } j \text{ is head of pipe } i \\ 0 & \text{else} \end{cases}$$
$$A_L^{rs} \in \mathbb{R}^{n_p\times n_{rs}} (A_L^{rs})_{ij} = \begin{cases} -1 & \text{if reservoir node } j \text{ is tail of pipe } i \\ 0 & \text{else} \end{cases}$$
$$A_R \in \mathbb{R}^{n_p\times n_d}, (A_R)_{ij} = \begin{cases} 1 & \text{if demand node } j \text{ is head of pipe } i \\ 0 & \text{else} \end{cases}$$
$$A_L \in \mathbb{R}^{n_p\times n_d}, (A_L)_{ij} = \begin{cases} -1 & \text{if demand node } j \text{ is tail of flow } i \\ 0 & \text{else} \end{cases}$$

We can combine them all and get the full incidence matrix A

$$A := (A_R^{rs} + A_L^{rs} \;\; A_R + A_L) \in \mathbb{R}^{n_p\times + d_{rs} + n_d}$$

With the help of these matrices we can write the spatial discretized system of equations as

$$\dot{p}_R + D_\alpha(m_L - m_R) = 0 \quad (2)$$
$$\dot{m}_L + D_\beta A^T p + \gamma + G(m_L)m_L = 0 \quad (3)$$
$$A_R m_R + A_L m_L = q_{set} \quad (4)$$
$$A_R p_d = p_R \quad (5)$$
$$A_L p_d = p_L \quad (6)$$
$$A_R^{rs} p_{rs} = p_R \quad (7)$$
$$A_L^{rs} p_{rs} = p_L \quad (8)$$
$$p_{rs} = p_{set} \quad (9)$$

To obtain these equation it is crucial to chose a suitable spatial discretization. In particular the time derivative of the pressure is evaluated at the right end of the pipe and the time derivative of the flux at the left end. The size of the first two equation is the number of pipes, equation (4) the number of junction, (5,6,7,8) number of pipes and the last equation number of tanks. D_α is a diagonal matrix containing $\alpha_i = a_i^2/A_i/\ell_i$ on the diagonal and D_β similar with $\beta_i = A_i/\ell_i$. γ is a vector with $\gamma_i = \rho_i A_i g \sin\alpha_i$ and $G(m_L)$ is a diagonal matrix function such that $G(m_L)_i = c_i m_L^i$. The matrix A is the incidence matrix as described above. The vector q_{set} has an entry for every demand node showing the given demand at that particular node given by $q_s(t)$ and similar is p_{set} the vector of the given pressures p_s at the reservoirs. In the modeling of the graph it is crucial to pick the direction of the edges such that every demand node has a right end of a pipe. This is possible for any topology as long as one of the nodes in the graph is not a demand node which means in our case it has to be a reservoir. We furthermore want all pipes the end in a reservoir to end in a left node there.

3. DECOUPLING

By selecting a matrix A_{select} which picks one pipe for each node that has a right end in that given node we can rewrite the system of equation in the variables p_d and m_L.

$$\dot{p_d} + A_{select} D_\alpha(-Cq_{set} + CAm_L) = 0$$
$$\dot{m}_L + D_\beta A^T \begin{pmatrix} p_d \\ p_{set} \end{pmatrix} + \gamma + G(m_L)m_L = 0$$

We will explain in detail how we create the matrix C which is the crucial part in this decoupling process. This resulting ODE is of size $n_d + n_p$ and has the general structure

$$\dot{x} = Tx + g(x,t) = f(x,t), \quad (10)$$

where the matrix T is given by

$$T = \begin{pmatrix} 0 & A_{select}D_{alpha}CA \\ D_\beta(A_R + A_L)^T & 0 \end{pmatrix},$$

and the vector x is combined by p_d and m_L.

4. IMEX

In order to solve this stiff and nonlinear ODE we make use of implicit-explicit (IMEX) integration methods. This allow us to deal with the stiffness in an efficient way while not having to solve large-scale nonlinear problems. First order methods are of the flavor

$$\frac{x_{n+1} - x_n}{h} = (1-\gamma)Tx_n + \gamma Tx_{n+1} + g(x_n,t) \quad (11)$$

for $\gamma \in [0,1]$ and time setp h, which leads to the iteration

$$x_{n+1} = (1 - h\gamma T)^{-1}(x_n + (1-\gamma)Tx_n + hg(x_n).$$

We study convergence properties of that by analyzing the matrix T and the function g as well as the differences for several values of γ. If $\gamma = 0$ we get explicit Euler and if $\gamma = 1$ we get a combination of implicit Euler for the linear part an explicit Euler for the nonlinear part. We will also show the differences within this class of methods as well as the difference to second order methods following Ascher et al. (1995).

5. MODEL ORDER REDUCTION

On the resulting ODE (10) we use the Model Order Reduction techniques Proper Orthogonal Decomposition (POD) together with Discrete Empirical Interpolation (DEIM) by Chaturantabut and Sorensen (2010). POD is a projection-based method where we find a projection matrix W such that the solution of (10) $x \approx W\hat{x}$ for $\hat{x}$ in a lower dimensional space. The resulting low-dimensional ODE is then given by

$$\dot{\hat{x}} = W^T TW\hat{x} + W^T g(W\hat{x},t).$$

DEIM is then used to create a truely low-dimensional function approximating $W^T g(W\hat{x},t)$.

6. CONCLUSIONS

The combination of POD-DEIM with the IMEX integration results in a significant speedup of simulation time.

REFERENCES

Ascher, U.M., Ruuth, S.J., and Wetton, B.T. (1995). Implicit-explicit methods for time-dependent partial differential equations. *SIAM Journal on Numerical Analysis*, 32(3), 797–823.

Chaturantabut, S. and Sorensen, D.C. (2010). Nonlinear model reduction via discrete empirical interpolation. *SIAM J. Sci. Comput.*, 32(5), 2737–2764. doi:10.1137/090766498.