International Series of Numerical Mathematics

Alexander Martin
Kathrin Klamroth
Jens Lang
Günter Leugering
Antonio Morsi
Martin Oberlack
Manfred Ostrowski
Roland Rosen
Editors

Mathematical Optimization of Water Networks

Birkhäuser

ISNM
International Series of Numerical Mathematics

Volume 162

For further volumes:
www.birkhauser-science.com/series/4819

Alexander Martin • Kathrin Klamroth •
Jens Lang • Günter Leugering • Antonio Morsi •
Martin Oberlack • Manfred Ostrowski •
Roland Rosen
Editors

Mathematical Optimization of Water Networks

Birkhäuser

Editors
Alexander Martin
Department of Mathematics
University of Erlangen-Nuremberg
Erlangen, Germany

Antonio Morsi
Department of Mathematics
University of Erlangen-Nuremberg
Erlangen, Germany

Kathrin Klamroth
Department of Mathematics and Informatics
University of Wuppertal
Wuppertal, Germany

Martin Oberlack
Fluid Dynamics
TU Darmstadt
Darmstadt, Germany

Jens Lang
Numerical Analysis and Scientific Computing
TU Darmstadt
Darmstadt, Germany

Manfred Ostrowski
Hydrology and Water Resources Engineering
TU Darmstadt
Darmstadt, Germany

Günter Leugering
Department of Mathematics
University of Erlangen-Nuremberg
Erlangen, Germany

Roland Rosen
CT T DE TC 3
Siemens AG
Munich, Germany

ISBN 978-3-0348-0435-6 ISBN 978-3-0348-0436-3 (eBook)
DOI 10.1007/978-3-0348-0436-3
Springer Basel Heidelberg New York Dordrecht London

Library of Congress Control Number: 2012942591

Mathematics Subject Classification: 90C11, 90C35, 35L02, 35L40, 45M05, 65K10, 90C30, 93C20, 93C83, 93C95

Printed on acid-free paper

Springer-Basel AG is part of Springer Science+Business Media (www.birkhauser-science.com)

Preface

Simulation and optimization methods, powerful tools that had their origin in Operations research, have for several decades been successfully applied to the field of water management. But only in a few rare cases has there been any feedback mechanism or direct cooperation with scientists in other areas of applied mathematics. Generally speaking, it has turned out that solving of real problems along narrow disciplinary lines has hardly led to significant innovations. Pushing the limits of these boundaries could provide a greatly widened scope for synergy and innovation. The development of complex model systems and their use for deriving optimal decisions in water management is taking place at a rapid pace. Still, simultaneous discoveries and developments in mathematics continue to be under-utilized due to the lack of coordination among disciplines. It is obvious that this failure of coordination and integration is a hindrance to expansion of progress in application of the knowledge we have gained. The purpose of this book is to help in overcoming this widespread condition.

The research and development measures described in this book comprise a practical reference for working in the area of water supply and sewage disposal. They have been carried out in cooperation with utility and disposal companies, infrastructure providers and planning authorities. Developments in the area of sewage disposal are of particular relevance, since optimal decisions on additional economically relevant investments for water pollution control will be made and implemented by national and regional guidelines and laws in the current and following years.

Water supply and drainage systems, as well as mixed water channel systems, are networks whose high dynamic is determined and/or affected by consumer habits with respect to drinking water, on the one hand, and by climate conditions, in particular rainfall, on the other hand. Besides the usual fluctuations in demand, water supply networks are subject to failures in supply and mains operation. The drains in mixed water channel systems, which usually run together in a wastewater treatment plant, consist of the drains of the water supply network, the so-called dry weather flow and the inflows from randomly occurring rainfalls. The ratio of the combined sewage flow vs. dry weather flow frequently is more than one hundred. From the point of view of hydrology and modeling, there is a need to work on both the flows

in open channels and the streams in closed tubes, within partially lumped systems as well as within algebraic, and/or differential-algebraic equations. In particular it is about hyperbolic systems of conservation equations with source terms that have found their way into the literature as Saint-Venant equations. It may happen though that, under certain flow conditions, equations in open channels and stream equations degenerate into parabolic and kinetic wave equations, and/or are presented as dead-time equations in a purely laminar case or as delay-differential equations. Status inquiries and/or estimates have to be applied when deciding on which model should be chosen in network operations.

This corresponds to a trust-region approach for hierarchic models. In the course of this approach, the optimization and control are supposed, on the basis of such dynamic systems, to be imposed on networks. The control strategy will thus show a closed loop character as well as an open loop character and will contain a continuous physical process shared along with discrete decision variables. They are thus complex multi-physical, multiscale, hybridically controlled and optimal dynamic systems on continuous networks.

The focus herewith is on two points. On one hand the dynamic is to be optimized with respect to practical relevant objectives and subject to state conditions. On the other hand, the technology developed in the first Step is supposed to locate maximum pressure in fresh water systems, and in sewage systems to determine, if necessary, transitions from running water to press water. These transition into sewage systems imply that channels fill up and the required and/or permitted discharge will be impeded. Pressure peaks in the fresh water system indicate a poor/faulty layout of the network. In both cases an attempt should be made, appropriate for the given topology, to avoid the occurrence of such indicators by means of control/steering on one hand and, if this doesn't prove to be satisfying, a design optimization on the other hand.

According to their size, water networks consist of hundreds or thousands of system elements. The processes that run continuously within the elements are one- to two-dimensional. Specific structures such as reservoirs or overflow valves make it even harder to describe such systems with mathematical models. In order to make these systems computable, the systems themselves must be aggregated and the physical fundamental equations needed to achieve efficient solution processes, must be simplified. While there are rather clearly structured, even though numerically challenging, equation systems for water supply networks, the models for simulation of combined sewers consist of a collection of extremely variable algorithms in terms of their physical properties and mathematical structure. The Saint-Venant-equations are a system of non-linear partial differential equations and are primarily based on the preservation of mass and impulse. The entire system is completed by appropriate initial- and boundary values, as well as by node conditions. The complexity of such systems rapidly reaches astronomic dimensions which can only be computationally treated by means of appropriate adaptive solution strategies. Adaptivity here takes effect on different levels: it includes individually adapted physical models with adaptive control of the spatial dimension and an efficient division of the network into areas of high and low dynamic, as well as control of spatial and temporal multi-scale

effects in numerical discretization by means of multilevel-methods. Corresponding decisions on models or their accuracy are determined by sensitivities that are being described by the corresponding adjoint systems.

Moreover, different types of decisions have to be taken in water management. The networks have to be optimized in terms of their topology and operation by targeting a variety of criteria. These criteria may for example be economic, social or ecological and may compete with each other. For this purpose practically applicable optimization methods are needed which are currently only partly available. Decisions can be of discrete or continuous nature, e.g. a decision has to be taken, if and where a system element in the network or in the planning area has to be established. In the event of a favorable decision, size, form and operation method have to be optimized. It is worth mentioning at this point that the further development, maintenance and replacement/substitution of urban drainage systems represent the most expensive civil infrastructure in Germany and is expected to continue to be a significant economical burden.

From the mathematical point of view these problems lead to network design problems including non-linear nodes and edge conditions. These problems on such a scale exceed the limits of today's methods. To solve such models the problems are approximated as partly linear or convex in such a manner that it is possible to take the qualitatively right decisions in terms of the network design (type and dimension of system elements). These approximations will also be key to further developments in choosing appropriate methods from the field of mixed-integer optimization in general and facility location problems in particular. The objective criteria relevant for optimization will be integrated into the analysis by using efficient methods of multicriterial optimization that need to be especially adapted. The disclosed solutions can be transferred to almost any water supply- or sewage system. In Germany alone, there are several thousand systems. An application on a European level or by any other industrial nation or consortium with similar standards of infrastructure, is basically also feasible.

The list of authors and participants of the project cover all relevant fields. With respect to mathematics, all necessary disciplines are available. This includes on one hand mixed-integer optimization, multi-objective and facility location optimization, numerics for cross-linked dynamic transportation systems and optimization as well as control of hybrid systems. According to flow dynamics, a long term experience in mathematical modeling of one- to three-dimensional flow processes on the basis of mixed hyperbolic-elliptical-parabolical systems exists. A close and direct connection to practical water management has been established by involving application-oriented engineering know-how out of the field of civil engineering. The interdisciplinary cooperation of partners from different scientific directions, together with people working in the industry, was the basis for directly applicable solutions of the problems that were addressed.

The industry partners came from the field of Industrial Solutions and Services, Water Technology (I&S WT) at Siemens AG, offering solutions for the control and visualization of all facilities in drinking water and waste-water systems. In close cooperation with the central goals, innovative solutions for integrated simulation of

processes and automization are being developed. As an industry partner, Siemens provides necessary data in order to pave the way for an integration of algorithmic results into commercial solutions.

Hessenwasser provides a public drinking water supply for around 2 million people in the metropolitan area of Frankfurt/Rhine-Main. The companies core competences include integrated groundwater management, abstraction, treatment, transportation and storage of water. Related services comprise groundwater monitoring, laboratory analysis and quality consulting, artificial recharge and irrigation water supply. In times of peak demand Hessenwasser supplies more than 400,000 m^3 of drinking water daily. Hessenwasser operates a distribution net of 415 km of transportation pipe which includes drinking-water reservoirs with a total volume of 343,000 m^3. Total water procurement amounts to a volume of about 100 mil. m^3/a. Sixty percent is produced in its own water works which are distributed throughout the region. Hessenwasser was founded in 2001 and is owned by the public multi-utility companies of the major cities Frankfurt, Wiesbaden and Darmstadt and of the district of Groß-Gerau.

Over recent years, Hessenwasser developed a hydraulic model for its complex cross-linked water distribution system in order to analyze the efficiency of the existing water system design and to cope with future needs. Our engineers gained a lot of expertise in advanced water distribution modeling and management. We supported the scientific approach to the project by technical advice and practical experience from our regional transportation network.

The chapters themselves cover the following topics:

In Chap. 1 the simulation of a water supply system on a mesoscale abstraction level is considered. The water network consists of storage tanks, pipes, pumps and valves. It is operated by the characteristics of the water supplier, the consumer and the pumps. For all network elements the modeling equations are given. They include mass and momentum conservation for pressurized pipe flow. For their numerical solution the method of lines is proposed. The discretization in space is based on a finite volume approach together with a local Lax-Friedrich splitting and central WENO reconstruction. Boundary and coupling conditions are implemented as algebraic equations. This leads to a system of differential-algebraic equations in time which is solved by a special Rosenbrock method. The paper ends with some typical simulation results of the network.

In Chap. 2 we consider the solution of the model equations of water supply networks and continuous optimal control tasks. We begin with the description of a simulation tool, in particular the numerical treatment of the water hammer equations. This includes a description of the implemented discretization scheme together with a stability and convergence analysis. As we will see, the applied scheme perfectly matches with the properties of the water hammer equations and thus builds a useful foundation for solution of the entire model equations as well as optimal control tasks. Then we consider the computation of sensitivity information, which is necessary for the application of gradient-based optimization techniques. Here, we follow a first-discretize approach to derive adjoint equations. Due to the special structure of the considered problems, very efficient algorithms can be applied. Finally, the chapter deals with the problem of singularities in the model equations of water supply

networks. Here, a physically motivated regularization approach is applied and also extended to be applicable in an adjoint calculus.

In Chap. 3 we introduce a mixed integer linear modeling approach for the optimization of dynamic water supply networks based on the piecewise linearization of nonlinear constraints. One advantage of applying mixed integer linear techniques is that these methods are nowadays very mature, that is, they are fast, robust, and are able to solve problems with up to a huge number of variables. The other major point is that these methods have the potential of finding globally optimal solutions or at least to provide guarantees of the solution quality. We demonstrate the applicability of this approach on examples networks.

In Chap. 4 we compare continuous nonlinear optimization with mixed integer optimization of water supply networks by means of a meso scaled network instance. We introduce a heuristic approach, which handles discrete decisions arising in water supply network optimization through penalization using nonlinear programming. We combine the continuous nonlinear and the mixed integer approach introduced in Chap. 3 to incorporate the solution quality. Finally, we show results for a real municipal water supply network.

Chapter 5 gives an overview of optimal control of sewer networks with dynamic process models. After introducing the method of model predictive control (MPC) and its requirements for optimization and process modeling, a focus is set on practical applications and the industrial viewpoint. An up-to-date sewer management system is introduced and used to illustrate industrial requirements and the mathematical challenges involved in it.

In practical application, open-channel or free-surface channel flow under the influence of gravity in sewers has traditionally been modeled with mathematical models based on one-dimensional governing equations of continuity and momentum—the so-called Saint-Venant equations. High volumetric flow rates or strong rains may lead to a transition from partial to fully filled cross sections in a sewer net, i.e. a free surface flow is no longer guaranteed. Hence the mathematical model of the Saint Venant equations loses its validity in whole or in parts of the channels and a transition occurs to the pressurized pipe equations. The main goal of Chap. 6 is to bring forward our knowledge about the process of changing the governing regime of fluid equations in the channel flow and to attempt to perform a general modeling tracking the movement of the transition interface between a free surface flow and the pressurized flow in a one-dimensional channel. Various flow cases with or without a moving transition are numerically investigated by means of the high-precision Discontinuous Galerkin Finite Element method. An exact knowledge of this event allows one to optimize the controlling of equipment and the operation in a sewer or design a new sewer correctly and effectively.

Chapter 7 introduces a software tool for MPC of sewer networks with a dynamic process model which is based on an interactive approach. A flexible optimizer, which implements local and global optimization methods, is connected to a dynamic sewer network model to evaluate the objective function values. Numerical results for a simple urban drainage network are presented, illustrating the functionality of the approach.

In Chap. 8 a hydrodynamic process model based on shallow water equations is discretized on 1D-networks with the method of finite volumes. Based on finite volumes, we replace algebraic coupling conditions by a consistent finite volume junction model. We use discrete adjoint computation for one-step Runge-Kutta schemes to generate fast and robust gradients for descent methods. We use the descent methods to generate an optimal control for an example network and discuss the computational results.

In Chap. 9 we compare the quality and generation performance of the optimal control sequence produced by the software frameworks BlueM.MPC and Lamatto.

In Chap. 10 we consider the goals and objectives arising in wastewater management in the context of a multiobjective analysis. This allows, among others, the individual consideration of (1) the overflow volume (i.e., the total amount of released water), (2) the pollution load in the released water, and (3) the cost of the generated control. Given a specific sewage network and data of typical inflow scenarios, a multiobjective offline analysis of the problem and, in particular, of the trade-off between the different goals, provides the decision maker with valuable information about the problem characteristics. This information can then be used to specify a suitable scalarized, single-objective optimization problem for the real-time optimal control that represents the decision maker's preferences in a best possible way. If an efficient solver for such scalarizations is available (which is the case for the problems considered here), this leads to an efficient online procedure that is justified by an extensive offline problem analysis. Even though the methods presented in this chapter were tailored for wastewater management problems, they are also applicable in the context of the other applications mentioned in this volume.

Erlangen, Germany — Alexander Martin
Wuppertal, Germany — Kathrin Klamroth
Darmstadt, Germany — Jens Lang
Erlangen, Germany — Günter Leugering
Erlangen, Germany — Antonio Morsi
Darmstadt, Germany — Martin Oberlack
Darmstadt, Germany — Manfred Ostrowski
Munich, Germany — Roland Rosen

Acknowledgements

We thank the BMBF (Bundesministerium für Bildung und Forschung, grants 03MAPAK1, 03LEPAK2, 03KLPAK3) for financial support.

Contents

Contributors

Kerstin Dächert Department of Mathematics and Informatics, Faculty of Mathematics and Natural Sciences, University of Wuppertal, Wuppertal, Germany

Björn Geißler Discrete Optimization (Lehrstuhl für Wirtschaftsmathematik), Friedrich-Alexander-Universität Erlangen-Nürnberg, Erlangen, Germany

Holger Hanss Siemens AG, I IS IN 1 WDC, Karlsruhe, Germany

Steffen Heusch Ingenieurhydrologie und Wasserbewirtschaftung, Technische Universität Darmstadt, Darmstadt, Germany

Johannes Hild Institut für Angewandte Mathematik (Lehrstuhl II), Friedrich-Alexander-Universität Erlangen-Nürnberg, Erlangen, Germany

Kathrin Klamroth Department of Mathematics and Informatics, Faculty of Mathematics and Natural Sciences, University of Wuppertal, Wuppertal, Germany

Oliver Kolb Numerical Analysis and Scientific Computing, Technische Universität Darmstadt, Darmstadt, Germany

Jens Lang Numerical Analysis and Scientific Computing, Technische Universität Darmstadt, Darmstadt, Germany

Günter Leugering Institut für Angewandte Mathematik (Lehrstuhl II), Friedrich-Alexander-Universität Erlangen-Nürnberg, Erlangen, Germany

Alexander Martin Discrete Optimization (Lehrstuhl für Wirtschaftsmathematik), Friedrich-Alexander-Universität Erlangen-Nürnberg, Erlangen, Germany

Saeid Moradi Ajam Strömungsdynamik, Technische Universität Darmstadt, Darmstadt, Germany

Antonio Morsi Discrete Optimization (Lehrstuhl für Wirtschaftsmathematik), Friedrich-Alexander-Universität Erlangen-Nürnberg, Erlangen, Germany

Martin Oberlack Strömungsdynamik, Technische Universität Darmstadt, Darmstadt, Germany

Manfred Ostrowski Ingenieurhydrologie und Wasserbewirtschaftung, Technische Universität Darmstadt, Darmstadt, Germany

Roland Rosen Siemens AG, CT T DE TC 3, Munich, Germany

Annelie Sohr Siemens AG, CT T DE TC 3, Munich, Germany

Gerd Steinebach Department of Electrical and Mechanical Engineering, Hochschule Bonn-Rhein-Sieg, Sankt Augustin, Germany

Yongqi Wang Strömungsdynamik, Technische Universität Darmstadt, Darmstadt, Germany

Chapter 1
Modeling and Numerical Simulation of Pipe Flow Problems in Water Supply Systems

Gerd Steinebach, Roland Rosen, and Annelie Sohr

Abstract In this chapter the simulation of a water supply system on a mesoscale abstraction level is considered. The water network consists of storage tanks, pipes, pumps and valves. It is operated by the characteristics of the water supplier, the consumer and the pumps. For all network elements the modeling equations are given. They include mass and momentum conservation for pressurized pipe flow. For their numerical solution the method of lines is proposed. The discretization in space is based on a finite volume approach together with a local Lax-Friedrich splitting and central WENO reconstruction. Boundary and coupling conditions are implemented as algebraic equations. This leads to a system of differential-algebraic equations in time which is solved by a special Rosenbrock method. The paper ends with some typical simulation results of the network.

1.1 Introduction

The main objective of water supply systems is the storage of water in reservoirs and the distribution to water consumers. The operation of a water supply system should ensure that enough water is available for consumption. On the other hand restrictions like maximum or minimum filling levels or pressure in the network must be regarded. Furthermore, the availability and the usage of water is highly time dependent. For these reasons, the simulation of the whole system and the optimization of its operation is desirable. In this contribution the simulation part is considered.

Typical components of a simplified water supply network are storage tanks, pipes, pumps and valves. In Sect. 1.2, a medium sized network consisting of such parts is introduced. The mathematical description of the network elements is given in Sect. 1.3. Due to the hyperbolic character of the flow equations, special methods must be applied for their numerical solution. The proposed approach is discussed in Sect. 1.4. Finally, some typical simulation results are presented in Sect. 1.5. The contribution ends with a short summary and some further remarks.

1.2 Example of a Water Supply System

Figure 1.1 shows a medium sized water supply system on a mesoscale abstraction level. It consists of 4 storage tanks, 16 pipes of 10.5 km total length, 3 pumps and

A. Martin et al. (eds.), *Mathematical Optimization of Water Networks*,
International Series of Numerical Mathematics 162,
DOI 10.1007/978-3-0348-0436-3_1,

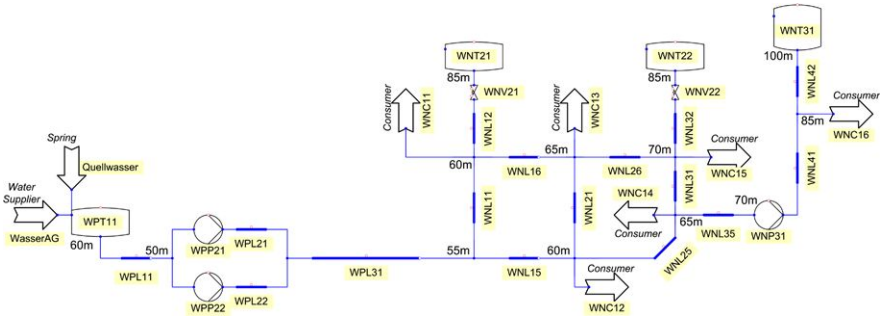


Fig. 1.1 Mesoscale water supply system

Table 1.1 Time dependent data for the operation of the water supply system

Date	Time	Supplier 1 m^3/s	Pump 1 state	Consumer 1 m^3/s	...
2000.09.30	24.00	0.028	0	0	
2000.10.01	1.00	0.028	0	0.017	
2000.10.01	1.30	0.056	1	0.017	
2000.10.01	2.00	0.056	1	0.017	...
2000.10.01	3.00	0.056	2	0.019	
2000.10.01	4.00	0.056	3	0.033	
2000.10.01	5.00	0.111	4	0.046	
2000.10.01	6.00	0.222	4	0.050	

2 valves. The storage tank at the left is fed by two water suppliers. The water in the tank is pumped by 2 pumps into the subnetwork located at higher altitudes. To this network, 3 further storage tanks and 6 consumers are connected. The operation of the pumps should ensure that the whole network is supplied with a sufficient amount of water and pressure at any time. The behaviour of the supplier, consumers, and the operation of the pumps are given by time dependent data. Table 1.1 shows an example of such input data for the model.

In Fig. 1.2 the input data for the operation of the water supply network is shown. The volume flow rates of the water suppliers and consumers are linearly interpolated from hourly given data of one day. Beside these data the operation states of the pumps must be given.

The simulation model is provided with this information, together with the length, diameters and altitudes of the pipes and the characteristics of the pumps and valves. With the model it should be possible to calculate the pressure in the network and the filling levels of the tanks.

The modeling, numerical solution, and some typical results are discussed in the following sections.

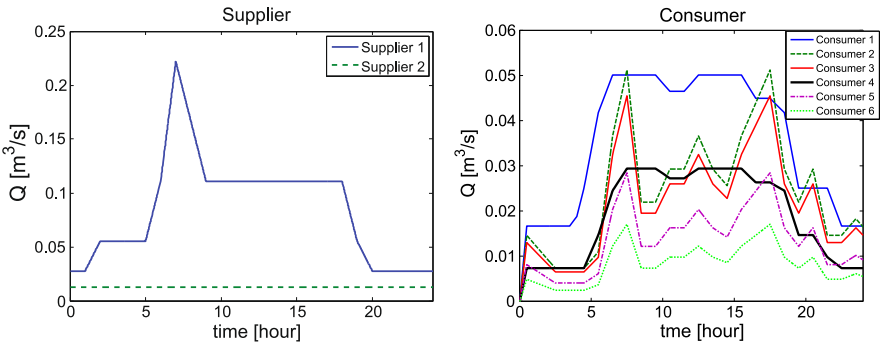


Fig. 1.2 Typical data for water supply and consumption

1.3 Modeling Equations

In water supply systems, the flow is usually dominated by pressure. On the other hand, in sewer networks, free surface flow conditions are found in the majority of the pipes and channels. Since the software concept of a company like Siemens for water and wastewater solutions should cover both cases, a unified modeling approach is desirable. In this section we start with the model equations for free surface flow and extend this model to cover pressure flow too.

1.3.1 Free Surface Flow

Free surface flow in channels or pipes is described by the well known Saint-Venant or shallow water equations (SWE) [15]:

$$\partial_t A + \partial_x Q = 0, \tag{1.1}$$

$$\partial_t Q + \partial_x \left(\frac{Q^2}{A} \right) + gA\partial_x z = -gAS_f. \tag{1.2}$$

$Q(x,t)$ denotes the volume flow along the channel or pipe, $A(x,t)$ the flooded cross-sectional area, g the gravitational acceleration, and $z(x,t)$ is the water surface elevation above a reference level, see Fig. 1.3.

This elevation z must be a given function of x and A: $z(x,t) = \tilde{f}(x, A(x,t))$. In the case of a trapezoidal geometry (see Fig. 1.4) the water depth h can be computed by the solution of $h(b_0 + h\tan(\alpha)) - A = 0$. The water elevation z is given by $z = h + S_0(x)$ with the bottom elevation S_0 of the channel.

If a circular pipe is considered, it is preferable to compute first the apex angle α from $A = \frac{r^2}{2}(\alpha - \sin(\alpha))$ and afterwards $h = r(1 - \cos(\frac{\alpha}{2}))$, see Fig. 1.5.

The friction S_f can be modeled by the Manning-Strickler formula

$$S_f = |u|u \frac{1}{K_{St}^2 r_{hyd}^{4/3}}$$

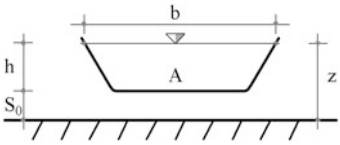


Fig. 1.3 Water surface elevation z, water depth h, cross-sectional area A, width b of the water level and bottom elevation S_0

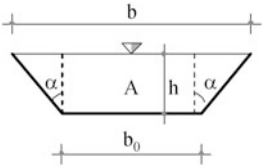


Fig. 1.4 Cross-sectional area A and apex angle α of a trapezoid

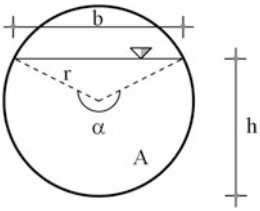


Fig. 1.5 Cross-sectional area A and apex angle α of a pipe

or the ansatz of Darcy-Weisbach [4]

$$S_f = |u|u\frac{\lambda}{8gr_{hyd}}.$$

In these formulae, u is the flow velocity given by $u = \frac{Q}{A}$, r_{hyd} is the hydraulic radius defined as the ratio of cross-section area A to the wetted perimeter U with $U = b_0 + \frac{2h}{\cos(\alpha)}$ for a trapezoidal and $U = r\alpha$ for a circular geometry. The formulae differ in the exponent of r_{hyd} and in the choice of the proportionality constants K_{St} and λ, which mainly depend on the roughness of the channel or pipe. A disadvantage of the laws is that they are valid only for steady ($\partial_t Q = 0$) and uniform ($\partial_x Q = 0$) flow conditions, but it is a common practice to apply them also to the general unsteady case. The advantage of the second formula is that λ has no physical dimension. For turbulent flow in a pipe

$$\lambda = \frac{1}{2.03^2 \log_{10}^2\left(\frac{K}{7.42r}\right)}$$

is a common approach, where K is the equivalent uniform grain roughness [6].

1.3.2 Pressure Flow

For free surface flow the density ρ of water is constant. (1.1), (1.2) arise from the conservation of mass and momentum given in it's original form

$$\partial_t(\rho A) + \partial_x(\rho Q) = 0, \tag{1.3}$$

$$\partial_t(\rho Q) + \partial_x\left(\frac{(\rho Q)^2}{(\rho A)}\right) + g(\rho A)\partial_x S_0 + A\partial_x p = -g(\rho A)S_f, \tag{1.4}$$

where (ρQ) is the mass flow. To consider pressure flow in a pipe, a non-constant density $\rho(x,t)$ is assumed. Furthermore, the flooded cross sectional area A does not depend on time and is identical to the whole pipe cross section denoted by $\bar{A}$, which may vary with space ($A = \bar{A}(x)$). The pressure p consists of the hydrostatic pressure and the overload pressure due to the assumed compressibility of water [3]:

$$p = \rho g h + \frac{1}{\beta}\frac{\rho - \rho_0}{\rho_0},$$

where β is the isothermal compressibility (also called compressibility coefficient) and ρ_0 is the density of water under free flowing conditions. It is assumed that $\rho_0 = 1000\frac{\text{kg}}{\text{m}^3}$ and $\beta = 5 \cdot 10^{-10}\frac{1}{\text{Pa}}$ [1].

With these assumptions, the equations for pressure and free surface flow read

$$\partial_t(\rho A) + \partial_x(\rho Q) = 0, \tag{1.5}$$

$$\partial_t(\rho Q) + \partial_x\left(\frac{(\rho Q)^2}{(\rho A)}\right) + g(\rho A)\partial_x z + \frac{A}{\beta\rho_0}\partial_x \rho = -g(\rho A)S_f. \tag{1.6}$$

In order to compute A and ρ from the state variable (ρA), we proceed as follows: If $\frac{(\rho A)}{\rho_0} > \bar{A}$, pressure flow exists with the density $\rho = \frac{(\rho A)}{\bar{A}}$. Otherwise, free surface flow is assumed with cross sectional area $A = \frac{(\rho A)}{\rho_0}$. An extension of this approach to pipes with an elastic tube wall is possible.

For each flow section, i.e. a single pipe or channel reach, boundary conditions and initial values must be prescribed. For pressure flow or assumed subcritical free surface flow, one upstream and one downstream boundary condition should be defined. Some implementation issues for the boundary conditions are discussed in Sect. 1.4.3 and more details concerning the coupling of several flow reaches can be found in [13].

1.3.3 Storage Tanks, Pumps and Valves

Storage tanks are assumed not to be under additional air pressure. Therefore a modeling by the conservation of volume is appropriate:

$$\frac{dV}{dt} = \sum_{i=1}^{n} Q_i(t).$$

$V(t)$ is the volume of the stored water in the tank and $Q_i(t)$, $i = 1, \ldots, n$, are volume flows into or out of the tank. These volume flows Q may depend on the dif-

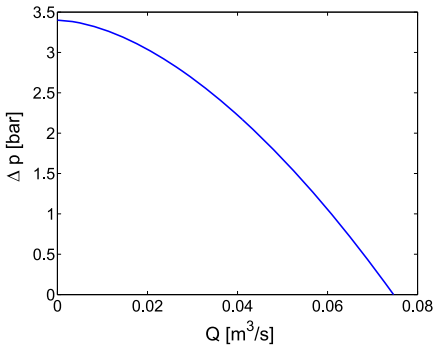


Fig. 1.6 Typical pump diagram

ference Δp between the hydrostatic pressure in the tank and the total pressure in an attached pipe:

$$Q = f\left(\rho_0 g h_{tank} - \left(\rho_{pipe} g h_{pipe} + \frac{1}{\beta}\frac{\rho_{pipe} - \rho_0}{\rho_0}\right)\right).$$

Such a function f could represent a valve, a pump or a weir.

A pump is characterized by an assumed simplified pressure-flow relation like

$$Q = \left(\frac{p_0 - \Delta p}{c}\right)^{1/1.7}.$$

This relation can be substituted by any other specific pump curve. Q is the volume flow through the pump and Δp is the pressure difference on both sides of the pump, which must be truncated in order to fulfill $0 \leq \Delta p \leq p_0$. p_0 and c are coefficients determining the pump load. Usually the pump can operate in several states (see Table 1.1) with different values of the coefficients. In state 0, the pump is off. Figure 1.6 shows a typical pump diagram.

A valve restricts the flow through it according to

$$\Delta p = \xi A Q|Q|$$

with valve coefficient ξ and valve cross-sectional area A. Again, Δp is the pressure difference between both sides of the valve and Q is the volume flow through the valve.

1.4 Numerical Solution

1.4.1 Method of Lines

The overall solution method is the method of lines (MOL) [9]. The MOL is based on the separation of the space discretization and the time integration. In the first step, the partial differential equations (PDEs) are discretized in space. The space

derivatives are approximated by suitable schemes which should be adopted to the properties of the PDEs [7].

In the case of hyperbolic problems, usually finite volume schemes are applied. This semi-discretization in space leads to a large system of ordinary differential equations (ODEs). A special feature of the network approach used in this chapter is the treatment of the boundary and coupling conditions. These conditions are modeled by algebraic equations, which must be added to the ODEs. Thus, a system of differential-algebraic equations (DAEs) is generated. Such DAEs require special solution methods, which usually are implicit or semi-implicit. In this chapter the Rosenbrock-Wanner method RODASP [12] is applied. Beside this method, of course, other schemes can be used like the popular BDF-methods [10] or newly developed two-step peer methods [14].

1.4.2 Space Discretization

In a standard conservation law approach, the equation

$$\partial_t q + \partial_x f(q) = 0 \tag{1.7}$$

for the unknown $q(x,t)$ with $q : [x_L, x_R] \times \mathbb{R}_{\geq 0} \to \mathbb{R}^n$, $f : \mathbb{R}^n \to \mathbb{R}^n$ is considered.

The space discretization is carried out by a finite volume approach. First, the space interval $[x_L, x_R]$ is divided into n finite volume cells, see Fig. 1.7.

In each cell i, $i = 1, \ldots, n$, the variable q_i is defined, which represents the average of the state variable $q(x,t)$ in this cell. Additionally, boundary variables q_L, q_R are defined, which are the boundary values of the state variable at the left or right side x_L, x_R. For simplification, the dependence of all variables on time t is not explicitely stated in this section.

The flux derivative $\partial_x f(q)$ in x_i is replaced by

$$\partial_x f\big(q(x_i)\big) \approx \frac{f_{i+1/2} - f_{i-1/2}}{\Delta x}$$

where $f_{i+1/2}$, $f_{i-1/2}$ are approximations of the flux function f evaluated at $q(x_{i+1/2})$, $q(x_{i-1/2})$ resp., and $x_{i+1/2}$, $x_{i-1/2}$ are the right and left cell boundaries of cell i. For hyperbolic conservation laws, it is very important that these approximations of $f_{i+1/2}$, $f_{i-1/2}$ include some kind of upwind information.

In this contribution a local Lax-Friedrich approach together with a central WENO reconstruction is applied [8]. The idea is the splitting of the flux function f into two components, which have exclusively positive or negative eigenvalues:

$$f(q) = \frac{1}{2}\Big(\underbrace{f(q) + |\lambda| q}_{=: f^+(q)} + \underbrace{f(q) - |\lambda| q}_{=: f^-(q)} \Big).$$

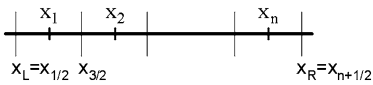


Fig. 1.7 Finite volume grid

Here λ should reflect the eigenvalue of $\frac{df}{dq}$ with the greatest absolute value. Now, f^+ is evaluated in the upstream and f^- in the downstream direction:

$$\begin{aligned} f(q_{i+1/2}) &= \frac{1}{2}\big(f\big(q^-_{i+1/2}\big) + |\lambda_{i+1/2}|q^-_{i+1/2} + f\big(q^+_{i+1/2}\big) - |\lambda_{i+1/2}|q^+_{i+1/2}\big) \\ &= \frac{1}{2}\big(f\big(q^-_{i+1/2}\big) + f\big(q^+_{i+1/2}\big) - |\lambda_{i+1/2}|\big(q^+_{i+1/2} - q^-_{i+1/2}\big)\big). \end{aligned}$$

The term

$$f^c_{i+1/2} = -\frac{1}{2}|\lambda_{i+1/2}|\big(q^+_{i+1/2} - q^-_{i+1/2}\big) \tag{1.8}$$

is called flux correction. $\lambda_{i+1/2}$ is chosen locally such that

$$|\lambda_{i+1/2}| = \max\Big\{\max_j\big|\lambda^j_i\big|, \max_j\big|\lambda^j_{i+1}\big|\Big\},$$

where λ^j_i is the jth eigenvalue of $\frac{df}{dq}$ at x_i.

The values $q^+_{i+1/2}, q^-_{i+1/2}$ must be reconstructed from the cell averages $q_1, \ldots, q_n$. The idea of the weighted essentially non-oscillatory (WENO) method is a linear combination of polynomials such that the weights are chosen according to its smoothness. These polynomials can be defined by [8]:

$$\begin{aligned} p^+_L(x) &= q_i + \frac{q_{i+1} - q_i}{\Delta x}(x - x_i), \\ p^+_C(x) &= q_{i+1} - \frac{1}{12}(q_{i+2} - 2q_{i+1} + q_i) + \frac{q_{i+2} - q_i}{2\Delta x}(x - x_{i+1}) \\ &\quad + \frac{q_{i+2} - 2q_{i+1} + q_i}{\Delta x^2}(x - x_{i+1})^2, \\ p^+_R(x) &= q_{i+1} + \frac{q_{i+2} - q_{i+1}}{\Delta x}(x - x_{i+1}). \end{aligned}$$

The smoothness indicators

$$\begin{aligned} S_L &= (q_{i+1} - q_i)^2, \\ S_C &= \frac{13}{3}(q_{i+2} - 2q_{i+1} + q_i)^2 + \frac{1}{4}(q_{i+2} - q_i)^2, \\ S_R &= (q_{i+2} - q_{i+1})^2 \end{aligned}$$

reflect the smoothness of p^+_L, p^+_C, p^+_R. Finally, $q^+_{i+1/2}$ is defined by

$$q^+_{i+1/2} = w_L p^+_L(x_{i+1/2}) + w_C p^+_C(x_{i+1/2}) + w_R p^+_R(x_{i+1/2})$$

with the weights

$$w_L = \frac{\alpha_L}{\alpha_L + \alpha_C + \alpha_R}, \qquad w_C = \frac{\alpha_C}{\alpha_L + \alpha_C + \alpha_R}, \qquad w_R = \frac{\alpha_R}{\alpha_L + \alpha_C + \alpha_R}$$

and

$$\alpha_L = \frac{1}{4}\frac{1}{(\epsilon + S_L)^p}, \qquad \alpha_C = \frac{1}{2}\frac{1}{(\epsilon + S_C)^p}, \qquad \alpha_R = \frac{1}{4}\frac{1}{(\epsilon + S_R)^p}.$$

The parameters $\epsilon = 10^{-6}$, $p = 0.6$ are chosen according to [8]. The definition of $q^-_{i+1/2}$ is similar. This approach gives a third order approximation for smooth solutions.

Unfortunately, (1.5), (1.6) with the state variable $q = ((\rho A), (\rho Q))^T$ are not in conservative form (1.7). The terms $g(\rho A)\partial_x z + \frac{A}{\beta\rho_0}\partial_x \rho$ destroy the conservation property and $g(\rho A)S_f$ introduces a source term. Well balanced schemes [2], which ensure that solutions of the steady state equations

$$0 = -\partial_x(\rho Q), \tag{1.9}$$

$$0 = -\partial_x\left(\frac{(\rho Q)^2}{(\rho A)}\right) - g(\rho A)\partial_x z - \frac{A}{\beta\rho_0}\partial_x \rho + g(\rho A)S_f \tag{1.10}$$

will be conserved, are desirable.

If $(g\rho A)\partial_x z + \frac{A}{\beta\rho_0}\partial_x \rho$ and $g(\rho A)S_f$ are discretized straightforward, this requirement cannot be fulfilled. By the following slight modification, it is at least possible to find a suitable discretization for the water at rest problem, which is defined by $\rho \equiv \rho_0$, $Q \equiv 0$, $\partial_x z \equiv 0$. (1.5) can be replaced by

$$\partial_t(\rho z) + \frac{1}{b}\partial_x(\rho Q) = 0 \tag{1.11}$$

with the width $b(x,t)$ of the water level, see Fig. 1.5. Then the flux-correction (1.8) for (1.11), (1.6) is given by

$$f^c_{i+1/2} = -\frac{1}{2}|\lambda_{i+1/2}|\left(\begin{pmatrix}(\rho z)\\(\rho Q)\end{pmatrix}^+_{i+1/2} - \begin{pmatrix}(\rho z)\\(\rho Q)\end{pmatrix}^-_{i+1/2}\right),$$

which vanishes equal to zero if the solution fulfills the water at rest problem. A combination of this and the common flux correction (1.8) is actually applied in the implementation. A fully well balanced discretization for the steady state (1.9), (1.10) could not be derived. As a consequence small oscillations occur in $Q(x,t)$ near transition locations where the flow is changing from sub- to supercritical or vice versa. For practical applications these oscillations are not relevant.

The eigenvalues of (1.5), (1.6) are given by $\lambda^f_{1/2} = u \pm \sqrt{\frac{gA}{b}}$ for free surface flow and $\lambda^p_{1/2} = u \pm \sqrt{\frac{1}{\beta\rho_0}}$ for pressure flow. The choice of $\lambda_{i+1/2}$ is a weighted linear combination of λ^f and λ^p.

1.4.3 Implementation of Boundary and Coupling Conditions

In order to implement the proposed space discretization, it is necessary to extrapolate the cell averages q_i of the state variables to the boundaries x_L and x_R, see Fig. 1.7. The most simple approach is to assume a piecewise constant solution $q(x)$ which fulfills

$$q_L = q_1, \qquad q_R = q_n.$$

A linear solution $q(x)$ leads to

$$q_L = \frac{1}{2}(3q_1 - q_2), \qquad q_R = \frac{1}{2}(3q_n - q_{n-1})$$

and a quadratic one to

$$q_L = \frac{1}{6}(11q_1 - 7q_2 + 2q_3), \qquad q_R = \frac{1}{6}(11q_n - 7q_{n-1} + 2q_{n-2}).$$

In most cases, the linear extrapolation gave the best results with respect to robustness and accuracy. These numerical boundary conditions are implemented as algebraic equations

$$\begin{aligned} 0 &= q_L - \frac{1}{2}(3q_1 - q_2), \\ 0 &= q_R - \frac{1}{2}(3q_n - q_{n-1}) \end{aligned}$$

for the unknowns q_L and q_R.

In the case of subcritical flow conditions, at each side, one of these numerical boundary conditions must be replaced by a physical one. If for instance the inflow $Q_0(t)$ at x_L and the water elevation $z_1(t)$ at the downstream end x_R are known, one ends up with the following boundary conditions:

$$\begin{aligned} 0 &= (\rho A)_L - \frac{1}{2}\big(3(\rho A)_1 - (\rho A)_2\big), \\ 0 &= Q_L - Q_0(t), \\ 0 &= z_R - z_1(t), \\ 0 &= (\rho Q)_R - \frac{1}{2}\big(3(\rho Q)_n - (\rho Q)_{n-1}\big). \end{aligned}$$

The same concept is applied for coupling conditions in the case of junctions or tributaries. Each physical coupling condition such as mass conservation or equal water elevation must replace one numerical boundary condition [13].

1.4.4 Solution of the Differential Algebraic Equations

The described semi-discretization in space together with the implementation of the boundary and coupling conditions leads to a large DAE-system of the type

$$My' = f(t, y); \quad y(t_0) = y_0$$

with a singular matrix M. Usually, these systems have index one, which guarantees a locally unique solution. In this chapter the solution method RODASP is applied. RODASP is a semi-implicit fourth order Rosenbrock-Wanner method with step size control especially adapted to DAEs [5, 12]. In each time step $t_n \to t_{n+1}$ with step size $h = t_{n+1} - t_n$, the Jacobian $DF = \frac{\partial f}{\partial y}(t_n, y_n)$ must be evaluated and $s = 6$ linear systems with matrix $E = (M - h\gamma DF)$ must be solved. Here, γ is a coefficient

of the method. It is very important that the sparsity of DF and E is considered. Otherwise an efficient solution of large DAE-systems is not possible.

The proposed method is not in all cases the most efficient one [11]. This is due to the fact that in contrast to e.g. BDF methods [5] or newly considered two-step peer methods [14] in each time step a new Jacobian must be evaluated. Nevertheless, RODASP is very robust and therefore it is the preferred method at present.

1.5 Simulation Results and Conclusions

The input data for the simulation run of the model has been shown in Sect. 1.2. Figure 1.8 shows the computed water levels and filling degrees in the four tanks. It is desirable that the filling degrees are in the range of 10–90 % of the maximum possible storage volumes.

In Fig. 1.9 the simulated water flow through the three pumps and the pressure evolutions at four locations within the network are presented. The water quantities depend on the operation states of the pumps, the associated pump curves and the pressure in the network elements. Sharp gradients in the water flows are generated

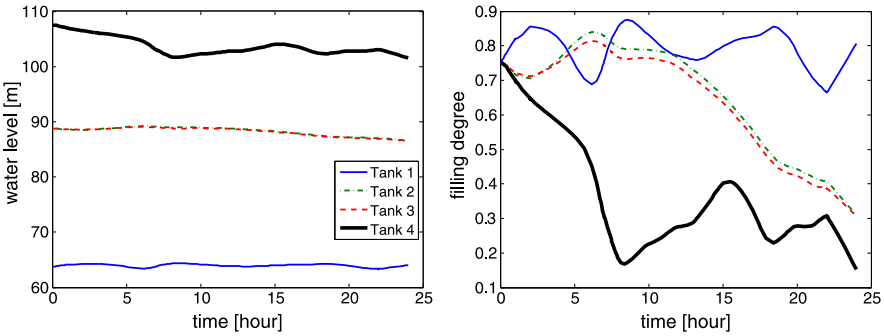


Fig. 1.8 Water levels and filling degrees in tanks

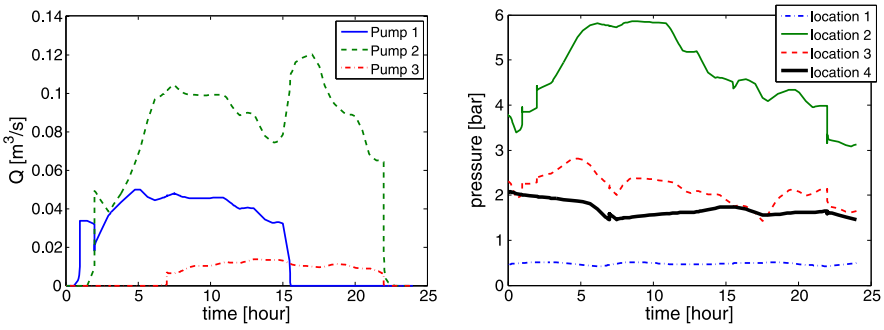


Fig. 1.9 Water volume flows through pumps and pressures at different locations in the network

when the pump states are switched. The pressure is shown just before (location 1) and behind (location 2) the first two pumps at altitudes of 50 m. Locations 3 and 4 at altitudes of 65 m resp. 85 m are before and behind the third pump.

These simulation results shall demonstrate that the modeling approach associated with the proposed numerical methods can successfully be applied to such networks.

For the space discretization a grid length of $\Delta x = 10$ m has been chosen.This choice leads to 1968 DAEs. From these equations, 1904 are ODEs and 64 are algebraic equations. The overall runtime for the simulation of 24 hours is 526 CPU-seconds on a standard PC in a MATLAB implementation. The RODASP integrator needs 197 successful and 54 failed time steps. Overall, 16282 function evaluations of the right hand side of the DAE-system including those for the computation of the Jacobian were necessary.

The time integration is sensitive to the operation of the pumps. If the pumps are switched on suddenly, the integrator fails in some cases. This failure is not restricted to RODASP, it can be observed for other integrators like `ode15s` or `ode23t` of MATLAB, too. Therefore, a temporary switch to an implicit Euler method is currently under investigation for such cases.

Nevertheless it is possible to simulate the presented water supply network sufficiently accurate in order to improve or even to optimize the operation rules.

References

1. W. Bohl, *Technische Strömungslehre* (Vogel, Berlin, 1998)
2. F. Bouchut, *Nonlinear Stability of Finite Volume Methods for Hyperbolic Conservation Laws and Well Balanced Schemes for Sources* (Birkhäuser, Basel, 2004)
3. C. Bourdarias, S. Gerbi, A finite volume scheme for a model coupling free surface and pressurised flows in pipes. J. Comput. Appl. Math. **209**(1), 109–131 (2007)
4. J.A. Cunge, F.M. Holly, A. Verwey, *Practical Aspects of Computational River Hydraulics* (Pitman, London, 1980)
5. E. Hairer, G. Wanner, *Solving Ordinary Differential Equations II, Stiff and Differential Algebraic Problems*, 2nd edn. (Springer, Berlin, 1996)
6. E. Heinemann, R. Feldhaus, *Hydraulik für Bauingenieure* (Teubner, Leipzig, 2003)
7. W. Hundsdorfer, J.G. Verwer, *Numerical Solution of Advection-Diffusion-Reaction Equations*, Springer Series in Comput. Math., vol. 33 (2003)
8. A. Kurganov, D. Levy, A third-order semidiscrete central scheme for conservation laws and convection-diffusion equations. SIAM J. Sci. Comput. **22**(4), 1461–1488 (2000)
9. W.E. Schiesser, *The Numerical Methods of Lines* (Academic Press, San Diego, 1991)
10. L.F. Shampine, M.W. Reichelt, The MATLAB ODE suite. SIAM J. Sci. Comput. **18**, 1–22 (1997)
11. G. Steinebach, Mathematische Modellbildung und numerische Methoden zur Strömungs-, Transport- und Reaktionssimulation in Netzwerken, in *Forschungsspitzen und Spitzenforschung, Innovationen an der Fachhochschule Bonn-Rhein-Sieg*, ed. by Ch. Zacharias et al. (Physica-Verlag, Heidelberg, 2008), pp. 151–163
12. G. Steinebach, P. Rentrop, An adaptive method of lines approach for modelling flow and transport in rivers, in *Adaptive Method of Lines*, ed. by A. Vande Wouver, Ph. Sauces, W.E. Schiesser (Chapman & Hall, London, 2001), pp. 181–205
13. G. Steinebach, S. Rademacher, P. Rentrop, M. Schulz, Mechanisms of coupling in river flow simulation systems. J. Comput. Appl. Math. **168**(1–2), 459–470 (2004)

14. G. Steinebach, R. Weiner, Peer methods for the one-dimensional shallow water equations with CWENO space discretization, Martin-Luther-Universität Halle-Wittenberg, Institut für Mathematik, Report No. 09 (2009)
15. J.J. Stoker, *Water Waves, the Mathematical Theory with Applications* (Interscience, New York, 1957)

G. Steinebach
Department of Electrical and Mechanical Engineering, Hochschule Bonn-Rhein-Sieg,
Grantham-Allee 20, 53757 Sankt Augustin, Germany
e-mail: gerd.steinebach@h-brs.de

R. Rosen (✉) · A. Sohr
Siemens AG, CT T DE TC 3, Otto-Hahn-Ring 6, 81730 Munich, Germany
e-mail: roland.rosen@siemens.com

A. Sohr
e-mail: annelie.sohr@siemens.com

Chapter 2
Simulation and Continuous Optimization

Oliver Kolb and Jens Lang

Abstract In this chapter we consider the solution of the model equations of water supply networks and continuous optimal control tasks. We begin with the description of our simulation tool in Sect. 2.1, in particular the numerical treatment of the water hammer equations. This includes the description of the implemented discretization scheme together with a stability and convergence analysis. As we will see, the applied scheme perfectly matches with the properties of the water hammer equations and thus builds a useful foundation for the solution of the entire model equations as well as optimal control tasks.

In Sect. 2.2 we consider the computation of sensitivity information, which is necessary for the application of gradient-based optimization techniques. Here, we follow a first-discretize approach to derive adjoint equations. Due to the special structure of the considered problems, very efficient algorithms can be applied.

Finally, Sect. 2.3 deals with the problem of singularities in the model equations of water supply networks. Here, a physically motivated regularization approach is applied and also extended to be applicable in an adjoint calculus.

2.1 Numerical Solution of the Model Equations

In this section, we describe our simulation tool, which numerically solves the underlying model equations. The main structure of this tool is described in Sect. 2.1.1. Here, we assume that the discretization of the model equations in time and space is given.

The water hammer equations are an integral part of the entire model of water supply networks. As we will see, this system of partial differential equations is hyperbolic. The numerical solution of hyperbolic PDEs demands great care regarding the discretization scheme, which crucially depends on the properties of the underlying equations. After recapitulating the basic properties of the water hammer equations in Sect. 2.1.2, we will describe the applied discretization scheme in Sect. 2.1.3 and give stability and convergence results.

2.1.1 Network Equations

The first step towards the solution of the model equations is an appropriate discretization. The treatment of the water hammer equations is described in detail in

A. Martin et al. (eds.), *Mathematical Optimization of Water Networks*,
International Series of Numerical Mathematics 162,
DOI 10.1007/978-3-0348-0436-3_2,

Sect. 2.1.3. The same time discretization is applied to the other components, modelled by algebraic and ordinary differential equations. The latter are discretized with one-step methods.

Let $t_0 < t_1 < \cdots < t_N$ be the time steps of the discretization. The application of the discretization schemes to the model equations yields a coupled system of (nonlinear) algebraic equations $E(y, u)$, which depends on state variables

$$y^T = \left(y(t_0)^T, y(t_1)^T, \ldots, y(t_N)^T\right),$$

like pressure head and flow rates, and control variables

$$u^T = \left(u(t_0)^T, u(t_1)^T, \ldots, u(t_N)^T\right),$$

e.g. the speed of pumps. Boundary and coupling conditions are already included in $E(y, u)$. Now, the simulation task consists of solving these equations for a given initial state $y(t_0)$ and control variables for all time steps. Due to the time-dependent structure, this set of equations can be partitioned and solved for $y(t_j)$ time step by time step ($j = 1, \ldots, N$), resulting in subsets of E of the form

$$F\left(t_{j-1}, t_j, y(t_{j-1}), y(t_j), u(t_{j-1}), u(t_j)\right) = 0.$$

While explicit dependencies may be solved in advance, the remaining (implicit) equations have to be solved with Newton's method. Here, we can exploit the sparsity structure of the underlying Jacobian matrix by using an appropriate solver for the sets of linear equations [7]. Unfortunately, the discretized model equations of water supply networks do not always yield unique solutions. The treatment of the underlying singularities is described in Sect. 2.3.

2.1.2 Properties of the Water Hammer Equations

The water hammer equations play an integral role in the modelling of water supply networks. The purpose of this section is to collect some properties of particular interest.

The water hammer equations are given by

$$\begin{aligned} &\frac{\partial}{\partial t} h + \frac{a^2}{gA} \frac{\partial}{\partial x} Q = 0, \\ &\frac{\partial}{\partial t} Q + gA \frac{\partial}{\partial x} h = -\lambda(Q) \frac{Q|Q|}{2dA} \end{aligned} \tag{2.1}$$

and can be written in the general form of a balance law

$$\frac{\partial}{\partial t} w + \frac{\partial}{\partial x} f(w) = g(w)$$

with $w = \binom{h}{Q}$ and

$$f(w) = \begin{pmatrix} \frac{a^2}{gA} Q \\ gAh \end{pmatrix}, \qquad g(w) = \begin{pmatrix} 0 \\ -\lambda(Q) \frac{Q|Q|}{2dA} \end{pmatrix}.$$

Obviously, $f(w)$ is linear in w and we have

$$\frac{\partial}{\partial x} f(w) = \underbrace{\begin{pmatrix} 0 & \frac{a^2}{ga} \\ gA & 0 \end{pmatrix}}_{= \frac{\partial}{\partial w} f(w)} \frac{\partial}{\partial x} w.$$

A short calculation yields

$$\lambda_{1,2} = \pm a$$

for the eigenvalues of $\frac{\partial}{\partial w} f(w)$. Thus, the water hammer equations form a hyperbolic system of PDEs with constant characteristic speeds.

Another important property of the water hammer equations is the dissipativity of the source term $g(w)$: The eigenvalues of

$$\frac{\partial}{\partial w} g(w) = \begin{pmatrix} 0 & 0 \\ 0 & -\frac{|Q|}{2dA} (\lambda'(Q) Q + 2\lambda(Q)) \end{pmatrix}$$

are

$$\mu_1 = -\frac{|Q|}{2dA} \big(\lambda'(Q) Q + 2\lambda(Q)\big) < 0, \qquad \mu_2 = 0.$$

In practical computations, often the stationary limit of the water hammer equations is used. Setting the time derivatives to zero yields that the discharge is constant (in space),

$$Q(x) \equiv Q, \tag{2.2}$$

and a linearly decreasing pressure head in flow direction,

$$h(x_1) - h(x_0) = -\lambda(Q) \frac{Q|Q|}{2gdA^2} (x_1 - x_0). \tag{2.3}$$

2.1.3 Implicit Box Scheme

For the discretization of the water hammer equations, we apply an implicit box scheme. The main drawback of explicit methods in the context of hyperbolic PDEs is the stepsize restriction due to the CFL condition, which is of the form

$$\Delta t \leq \alpha \frac{\Delta x}{\lambda_{max}} \tag{2.4}$$

with some positive $\alpha \in \mathbb{R}$. Here, λ_{max} denotes the spectral radius of the Jacobian matrix of the flux function. Regarding the water hammer equations, the CFL condition is very restrictive since λ_{max} equals the speed of sound (in water). Thus, very fine time discretizations would be necessary for stability reasons, while an appropriate resolution of the typically moderate dynamics in the daily operation of water supply networks would allow much larger time steps.

We now formulate the applied scheme for balance laws of the form

$$\frac{\partial}{\partial t} w + \frac{\partial}{\partial x} f(w) = g(w), \quad (x,t) \in \mathbb{R} \times \mathbb{R}_+ \tag{2.5}$$

with given initial data

$$w(x,0) = w_0(x), \quad x \in \mathbb{R}. \tag{2.6}$$

To approximate (weak) solutions of (2.5)–(2.6), we choose a spatial mesh size Δx, a time grid size Δt and introduce a piecewise constant function $\tilde{w}(x,t)$ defined by

$$\tilde{w}(x,t) = w_j^n \quad \text{for } (x,t) \in I_j \times J_n \tag{2.7}$$

with $I_j = [(j-0.5)\Delta x, (j+0.5)\Delta x)$ and $J_n = [n\Delta t, (n+1)\Delta t)$. For the computation of the approximate values $w_j^n \approx w(j\Delta x, n\Delta t)$, we consider the implicit box scheme

$$\begin{aligned} \frac{w_{j-1}^{n+1} + w_j^{n+1}}{2} = \frac{w_{j-1}^n + w_j^n}{2} - \frac{\Delta t}{\Delta x}\left(f\left(w_j^{n+1}\right) - f\left(w_{j-1}^{n+1}\right)\right) \\ + \Delta t \frac{g(w_{j-1}^{n+1}) + g(w_j^{n+1})}{2}. \end{aligned} \tag{2.8}$$

As initial conditions, we set

$$w_j^0 = \int_{I_j} w_0(x)\, dx. \tag{2.9}$$

When implementing this method for a scalar balance law on a finite grid $x_l < x_{l+1} < \cdots < x_{r-1} < x_r$, we get $r - l$ equations for $r - l + 1$ variables. So, we have to impose boundary conditions at exactly one boundary, depending on the characteristic direction, i.e., on the sign of f'. In order that the proposed scheme may work, we have to assume that the sign of f' does not change over the computational domain. The generalization for systems of balance laws is that the signature of the characteristic directions does not change. This assumption is often satisfied for subsonic flows and also holds for the water hammer equations (2.1).

We mention that for $g \equiv 0$ and $w^n, w^{n+1} \in L^1(\mathbb{Z})$, the scheme (2.8) is conservative. Moreover, it can be easily shown that the proposed scheme is exact in the stationary case (2.2)–(2.3) of the water hammer equations. Next, we give some further results for the applied scheme in the scalar case, which have already been published in [12], where also the proofs can be found.

First, it can be shown that the box scheme admits a unique solution in $L^1(\mathbb{Z})$ in every time step. For this, we assume $f' \geq \lambda_{min} > 0$. Analogously, Proposition 1 and the following propositions hold in the case $f' \leq -\lambda_{min} < 0$.

Proposition 1 (Existence and Uniqueness) *For* $w^n \in L^1(\mathbb{Z})$, $f, g \in C^1(\mathbb{R})$, $g(0) = 0$, $g' \leq 0$, $f' \geq \lambda_{min} > 0$ *and* $\frac{\Delta t}{\Delta x} \geq \frac{1}{2\lambda_{min}}$, *scheme* (2.8) *admits a unique solution* $w^{n+1} \in L^1(\mathbb{Z})$.

In Proposition 1, we have introduced the requirement $\Delta t \geq \Delta x/(2\lambda_{min})$. In contrast to the CFL condition (2.4), which determines an upper bound for the time grid size Δt, the implicit structure of the scheme leads to a lower bound for the time grid size.

Motivated by the well-known results of Kružkov [13], the following stability results can be shown:

Proposition 2 (Stability) *Let* $w^n, v^n \in L^\infty(\mathbb{Z}) \cap L^1(\mathbb{Z}) = L^1(\mathbb{Z})$, $f, g \in C^1(\mathbb{R})$, $g(0) = 0$ *and* $g' \leq 0$. *Then, scheme* (2.8) *has the following stability properties*:

(1) *If* $\frac{\Delta x}{\Delta t} \leq 2f' + \Delta x g'$, *then* $\|w^{n+1}\|_{L^\infty(\mathbb{Z})} \leq \|w^n\|_{L^\infty(\mathbb{Z})}$.
(2) *If* $\frac{\Delta x}{\Delta t} \leq 2f'$, *then* $\|w^{n+1} - v^{n+1}\|_{L^1(\mathbb{Z})} \leq \|w^n - v^n\|_{L^1(\mathbb{Z})}$ *and* $TV(w^{n+1}) \leq TV(w^n)$.

The requirement in (1) can even be weakened to $\frac{\Delta x}{\Delta t} \leq 2f'$ under mild additional assumptions. Next, in analogy to the Lax-Wendroff-Theorem (see e.g. [14], pp. 239ff.), it can be shown:

Proposition 3 *Let* $(w^{(k)})_{k\in\mathbb{N}}$ *be a sequence constructed by scheme* (2.8)–(2.9) *and converging in* $L^1_{loc}(\mathbb{R} \times \mathbb{R}_+)$ *with* $\Delta t^{(k)}, \Delta x^{(k)} \overset{k\to\infty}{\longrightarrow} 0$. *Then, the limit* $\hat{w} = \lim_{k\to\infty} w^{(k)}$ *is a weak solution of the Cauchy problem* (2.5)–(2.6).

Finally, assuming the stability properties

$$\begin{aligned} \left\|w^{n+1}\right\|_{L^\infty(\mathbb{Z})} &\leq \left\|w^n\right\|_{L^\infty(\mathbb{Z})}, \\ \left\|w^{n+1}\right\|_{L^1(\mathbb{Z})} &\leq \left\|w^n\right\|_{L^1(\mathbb{Z})}, \\ TV\left(w^{n+1}\right) &\leq TV\left(w^n\right), \end{aligned} \tag{2.10}$$

which can for instance be achieved by fulfilling the requirements of Proposition 2, convergence to the so-called entropy solution can be shown:

Proposition 4 (Convergence to Entropy Solution) *Let* $w_0 \in L^\infty(\mathbb{R}) \cap L^1(\mathbb{R})$, $f, g \in C^1(\mathbb{R})$, $g(0) = 0$, $g' \leq 0$, $f' \geq \lambda_{min} > 0$ *and* $TV(u_0) < \infty$. *Let* $(w^{(k)})_{k\in\mathbb{N}}$ *be a sequence constructed by scheme* (2.8)–(2.9), *fulfilling the stability properties* (2.10) *and with* $\Delta t^{(k)}, \Delta x^{(k)} \overset{k\to\infty}{\longrightarrow} 0$, *where* $r = \frac{\Delta t^{(k)}}{\Delta x^{(k)}} \geq \frac{1}{2\lambda_{min}}$. *Then, the limit*

$\hat{w} = \lim_{k\to\infty} w^{(k)}$ *exists in* $L^1_{loc}(\mathbb{R}\times\mathbb{R}_+)$ *and is the entropy solution of the Cauchy problem* (2.5)–(2.6).

2.2 Adjoint Calculus

The last section was concerned with the solution of the simulation task. The next step is to determine the control u in such a way that a given objective function is optimized while certain constraints have to be fulfilled. Therefore, we consider the following optimal control problem:

$$\begin{aligned} \min_u\ & f\big(y(u),u\big) \\ \text{s.t. } & g\big(y(u),u\big) \geq 0 \\ & h\big(y(u),u\big) = 0 \\ & u_{min} \leq u \leq u_{max} \end{aligned} \tag{2.11}$$

with state vector y, control vector u, objective function f, inequality constraints g and equality constraints h. The state vector is assumed to be a function of the control vector, that is, for a given control u the state y is uniquely determined. As described in Sect. 2.1.1, the state vector results from solving a (nonlinear) set of equations $E(y,u)=0$ for y. For this reason, the functions f, g and h can also be considered as functions solely depending on the control u. In fact, the state variables are not visible for the optimization tools we have linked to our software, their interface only contains the control variables.

We want to solve (2.11) with gradient-based optimization methods like DONLP2 [17, 18], IPOPT [19] and KNITRO [5]. While the simulation tool enables us to evaluate the objective function and the constraints for a given control u, we still have to provide sensitivity information for all functions with respect to the control. Adjoint calculus is a very efficient way to compute the so-called *reduced gradients*. In principle, there are two different ways to compute the desired information via adjoint equations as shown in Fig. 2.1. Starting with the model equations, one may first derive (analytically) adjoint equations and apply an appropriate discretization scheme afterwards. The second possibility is to derive adjoint equations based on the discretized model equations.

Since both approaches have their advantages and disadvantages, we apply both in our software. Nevertheless, there is a strong emphasis on the second approach because the necessary components are much easier to implement. Further details of this approach are provided in the following sections.

2.2.1 The First-Discretize Approach

We consider the computation of the reduced gradient of an arbitrary scalar function $f(y(u),u)$ via adjoint equations derived by a first-discretize approach. As above

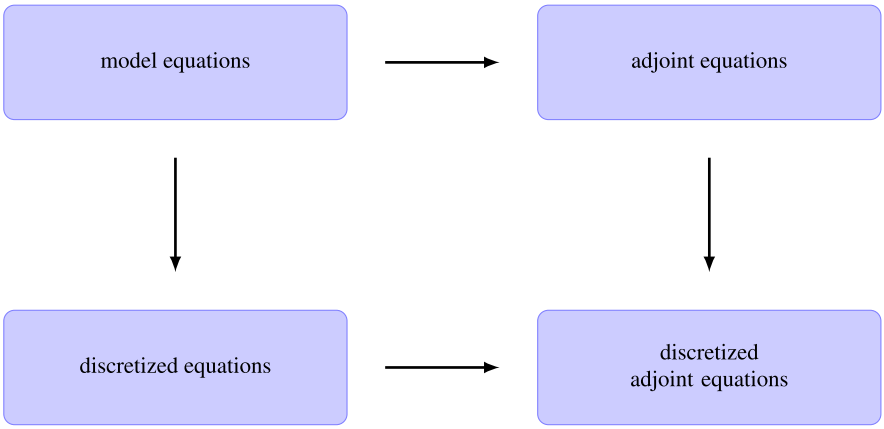


Fig. 2.1 Two ways from the model equations to the discretized adjoint equations

$y(u)$ is considered to be the unique solution of $E(y,u)=0$. Of course, the described procedure is not only valid for f being our objective function but any of the equality or inequality constraints in (2.11).

To derive the adjoint equations, which are necessary for the computation of $\frac{d}{du}f(y(u),u)$, we introduce the Lagrange function

$$L(y,u)=f(y,u)+\xi^T E(y,u), \tag{2.12}$$

where ξ is the so-called *adjoint state*. With $y=y(u)$, basic transformations of (2.12) lead to

$$\begin{aligned}\frac{d}{du}f\big(y(u),u\big)&=\frac{d}{du}L\big(y(u),u\big)-\frac{d}{du}\xi^T\underbrace{E\big(y(u),u\big)}_{=0}=\frac{d}{du}L\big(y(u),u\big)\\&=\underbrace{\frac{\partial}{\partial y}L\big(y(u),u\big)}_{\stackrel{!}{=}0\Rightarrow\xi}\frac{dy}{du}+\frac{\partial}{\partial u}L\big(y(u),u\big)=\frac{\partial}{\partial u}L\big(y(u),u\big)\\&=\frac{\partial}{\partial u}f\big(y(u),u\big)+\xi^T\frac{\partial}{\partial u}E\big(y(u),u\big).\end{aligned} \tag{2.13}$$

Thus, we have reduced the task of computing the total derivative of f with respect to u to the computation of the partial derivatives of f and E with respect to u and solving the system of *adjoint equations*:

$$\frac{\partial}{\partial y}L\big(y(u),u\big)=\frac{\partial}{\partial y}f\big(y(u),u\big)+\xi^T\frac{\partial}{\partial y}E\big(y(u),u\big)\stackrel{!}{=}0$$

$$\Leftrightarrow \quad \underbrace{\left(\frac{\partial}{\partial y}E\big(y(u),u\big)\right)^T}_{\text{independent of } f}\xi = -\left(\frac{\partial}{\partial y}f\big(y(u),u\big)\right)^T. \tag{2.14}$$

It is important to notice that (2.14) is a linear system and the matrix $\frac{\partial}{\partial y}E(y(u),u)$ is independent of the function f. Therefore, this matrix and any decomposition of it computed for solving (2.14) only needs to be computed once.

After having solved (2.14), we get our reduced gradient from (2.13):

$$\frac{d}{du}f\big(y(u),u\big) = \frac{\partial}{\partial u}f\big(y(u),u\big) + \xi^T \underbrace{\frac{\partial}{\partial u}E\big(y(u),u\big)}_{\text{independent of } f}.$$

Here, the matrix $\frac{\partial}{\partial u}E(y(u),u)$ is independent of f and therefore only needs to be computed once, independent of the number of computed gradients.

2.2.2 Application to Time-Dependent Problems

In the case of time-dependent control problems, the task (2.11) and therewith the reduced gradient (2.13) and the adjoint system (2.14) have a very special structure.

Let us begin with the state defining function $E(y,u)$. As described in Sect. 2.1.1, we start in a certain state $y(t_0) = y_0$. From any state $y(t_j)$, we come to the next state $y(t_{j+1})$ by solving a set of equations of the following form:

$$F\big(t_{old}, t_{new}, y(t_{old}), y(t_{new}), u(t_{old}), u(t_{new})\big) = 0. \tag{2.15}$$

Altogether, we have

$$E(y,u) = \begin{pmatrix} y(t_0) - y_0 \\ F(t_0, t_1, y(t_0), y(t_1), u(t_0), u(t_1)) \\ \vdots \\ F(t_{N-1}, t_N, y(t_{N-1}), y(t_N), u(t_{N-1}), u(t_N)) \end{pmatrix} = 0. \tag{2.16}$$

For the matrix in the adjoint system, we get

$$\frac{\partial}{\partial y}E(y,u) = \begin{pmatrix} I & & & & \\ A_1 & B_1 & & & \\ & A_2 & B_2 & & \\ & & \ddots & \ddots & \\ & & & A_N & B_N \end{pmatrix}$$

with

$$A_j = \frac{\partial}{\partial y_{old}}F\big(t_{j-1}, t_j, y(t_{j-1}), y(t_j), u(t_{j-1}), u(t_j)\big)$$

and

$$B_j = \frac{\partial}{\partial y_{new}} F\big(t_{j-1}, t_j, y(t_{j-1}), y(t_j), u(t_{j-1}), u(t_j)\big).$$

For an arbitrary scalar function f, the set of adjoint equations (2.14) then reads

$$\begin{pmatrix} I & A_1^T & & & \\ & B_1^T & A_2^T & & \\ & & B_2^T & \ddots & \\ & & & \ddots & A_N^T \\ & & & & B_N^T \end{pmatrix} \begin{pmatrix} \xi(t_0) \\ \xi(t_1) \\ \vdots \\ \xi(t_N) \end{pmatrix} = - \begin{pmatrix} \frac{\partial}{\partial y_0} f(y,u)^T \\ \frac{\partial}{\partial y_1} f(y,u)^T \\ \vdots \\ \frac{\partial}{\partial y_N} f(y,u)^T \end{pmatrix}. \tag{2.17}$$

Here, the partial derivatives $\frac{\partial}{\partial y_j}$ refer to the blockwise partitioning of the state vector according to the time steps.

Due to the blockwise bidiagonal structure of the matrix $\frac{\partial}{\partial y} E(y,u)$, the linear system (2.17) can be solved backwards in time and blockwise, reducing the size of the systems to be solved:

$$\begin{aligned} \xi(t_N) &= -\big(B_N^T\big)^{-1} \frac{\partial}{\partial y_N} f(y,u)^T, \\ &\vdots \\ \xi(t_j) &= -\big(B_j^T\big)^{-1} \left(\frac{\partial}{\partial y_j} f(y,u)^T + A_{j+1}^T \xi(t_{j+1}) \right), \\ &\vdots \\ \xi(t_0) &= -\left(\frac{\partial}{\partial y_0} f(y,u)^T + A_1^T \xi(t_1) \right). \end{aligned}$$

Besides the structure of the matrix $\frac{\partial}{\partial y} E(y,u)$, the right-hand side of (2.17) typically also features a special structure. Further benefit can be made out of the structure of $\frac{\partial}{\partial u} E(y,u)$. Similar to the structure of $\frac{\partial}{\partial y} E(y,u)$, we get

$$\frac{\partial}{\partial u} E(y,u) = \begin{pmatrix} 0 & \cdots & \cdots & \cdots & 0 \\ \frac{\partial}{\partial u_{old}} E_1 & \frac{\partial}{\partial u_{new}} E_1 & & & \\ & \frac{\partial}{\partial u_{old}} E_2 & \frac{\partial}{\partial u_{new}} E_2 & & \\ & & \ddots & \ddots & \\ & & & \frac{\partial}{\partial u_{old}} E_N & \frac{\partial}{\partial u_{new}} E_N \end{pmatrix},$$

where E_j abbreviates $F(t_{j-1}, t_j, y(t_{j-1}), y(t_j), u(t_{j-1}), u(t_j))$. Since the first block of rows equals zero, there is no need to compute $\xi(t_0)$ for the evaluation of the reduced gradient via (2.13). This result is not surprising, because the initial state

$y(t_0)$ is given and therefore does not depend on the control. Moreover, we usually have $\frac{\partial}{\partial u_{old}} E_j = 0$. In this case, $\frac{\partial}{\partial u} E(y, u)$ reduces to a block-diagonal matrix.

2.3 Singularities

As already mentioned in Sect. 2.1.1, the model equations of a water supply networks may contain non-unique solutions for certain constellations of elements and control states. Naturally, non-unique solutions cause problems when trying to solve the discretized model equations. In Newton's method, we are confronted with singular or at least ill-conditioned Jacobian matrices. But it is possible to introduce a physically reasonable regularization of the underlying matrices, which turns out to be a Tychonoff-like regularization. The presented results have already been published in [11].

2.3.1 Introduction

The first algorithm to determine pressure heads and flows for a networked system in the steady state case was published in 1936 [6]. Meanwhile, a variety of software packages has been implemented, e.g. KANET [1], STANET [2] and EPANET [15]. The latter one is released as freeware by the United States Environmental Protection Agency, broadly accepted, and often a core part of proprietary packages. But EPANET and also other codes have difficulties with certain constellations of control devices. Several problem cases have been published by Simpson in 1999 [16]. Meanwhile, the EPANET software copes with all of them but many recent publications still report about new cases where it fails or computes wrong results, e.g. [4, 9].

One underlying problem can be explained by a very simple example: Consider two closed valves as shown in Fig. 2.2. The equations modelling the pressure heads h_l and h_r and the flow rates q_l and q_r at the connection of the two valves are given as follows,

$$F(h_l, q_l, h_r, q_r) = \underbrace{\begin{pmatrix} h_l - h_r \\ q_l - q_r \\ q_l \\ q_r \end{pmatrix}}_{=: b(h_l, q_l, h_r, q_r)} \stackrel{!}{=} \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}. \tag{2.18}$$

Here, the pressure head between the two valves is not uniquely determined by the model equations. We only claim that h_l equals h_r. From the practical point of view, we might not be interested in the "real" pressure values between the two closed valves, but in a dynamic or quasi-stationary numerical simulation, we would expect the pressure variables to keep the same or at least similar values as in the previous time step.

Of course, one could cope with the non-uniqueness in the mentioned example but the situation becomes more difficult for large networks and especially when devices

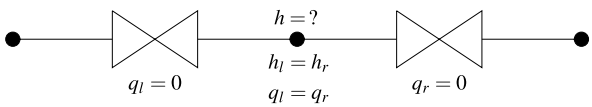


Fig. 2.2 Two closed valves—the model equations do not yield a unique solution

are state-controlled so that "truncated" parts are not known a priori. Anyway, we expect getting into trouble when solving the in general nonlinear model equations of a water supply network with Newton's method in situations like above.

The nature of the non-uniqueness in our small example is in close correlation with the Jacobian matrix of the model equations:

$$A(h_l, q_l, h_r, q_r) = \begin{pmatrix} 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}. \tag{2.19}$$

The Jacobian matrix is singular, and obviously, the vector $v_0 = (1, 0, 1, 0)$ is an element of the kernel of A. Thus, when solving $A\delta = b$ in Newton's method, we have to cope with the singularity of A, and moreover, an arbitrary multiple of v_0 could be added to any solution δ, which corresponds to arbitrary but equal values for h_l and h_r.

As already mentioned, from the practical point of view, we might not be interested in the pressure values between the two closed valves, but at least, we have to ensure that the underlying singularity does not impede the solution process of the whole system of model equations. Moreover, we would prefer the pressure variables to change as little as possible.

In literature, there are different approaches to handle the shown problem in general. Alvarez et al. [4] propose to add virtual tanks in the network. In [8], Deuerlein proposes a reformulation of the model equations in the form of a minimization problem. For the analysis of the resulting model, he also uses game theory. Hähnlein [10] uses basically the same modelling as we do. For solving the sets of linear equations in Newton's method, he applies singular value decomposition.

Regarding the example problem, solving $A\delta = b$ with an SVD has exactly the desired effect: We get the solution of the linear system of equations with minimal norm—the solution component in the nullspace of A (multiples of the vector v_0) equals zero. Since this property of using an SVD for solving sets of linear equations holds in a more general setting, singular value decomposition seems to be a good approach. While we get a solution of the linear system of equations (if one exists), the correction terms for the "critical variables" (here h_l and h_r) are kept small. Moreover, we may eliminate small singular values in the SVD of the matrix to stabilize the solution process. Besides all those advantages of using singular value decomposition, the price to pay is the huge computational effort.

Typically, there are various points in water supply networks where the pressure variables may not be uniquely determined and thus may be "critical" in the solution process. Without proper treatment, one gets very large correction terms δ in

Newton's method or no solution at all for the linear system of equations, and either usually leads to a failure of the method.

In the next section, we will present our approach to the solution of the introduced class of problems. In the general setting, parts of the solution of the underlying sets of linear equations are uniquely determined while there may be degrees of freedom in other parts. We want to keep those solution components as small as possible while maintaining a useful solution.

Our approach is based on the QR decomposition of a modified matrix $\tilde{A}$. Since we know the critical variables in our applications, the original matrix A is extended with additional rows in such a way that the resulting matrix has full rank. The basic idea behind this approach is to penalize corrections in critical variables. Although we increase the size of the underlying matrix, the application of a QR decomposition to the modified matrix compared to the SVD of the original matrix results in an enormous speed-up. But this is not the only contribution we make to the simulation task of water supply networks. Additionally, we use the decomposition of the modified matrix to efficiently compute sensitivity information with respect to a given target functional in Sect. 2.3.3.

2.3.2 Theoretical Analysis—Forward Direction

We consider the same setting as in Sects. 2.1.1 and 2.2: The discretization of the model equations of the whole water supply network yields a coupled system of nonlinear algebraic equations $E(y, u)$, which can be split up according to (2.16). During the solution process with Newton's method, we have to compute corrections δ by solving a linear system of equations of the form

$$A\delta = b \quad \Leftrightarrow \quad A\delta - b = 0 \tag{2.20}$$

with A being an $n \times n$ matrix. If A is singular (or ill-conditioned), we cannot make use of an LU decomposition of A in order to solve (2.20). Instead, we reformulate (2.20) as linear least squares problem

$$\min_{\delta} \|A\delta - b\|_2^2. \tag{2.21}$$

This problem can always be solved with a singular value decomposition of A. The SVD yields the (unique) solution δ^* of (2.21) where additionally $\|\delta^*\|_2$ is minimal among all solutions. In general, there is a residual $r^* = A\delta^* - b$.

Linear least squares problems can also be solved via a QR decomposition of the underlying matrix if it has full rank. Therefore, we consider the modified problem

$$\min_{\delta} \|\tilde{A}\delta - \tilde{b}\|_2^2 \tag{2.22}$$

with $\tilde{A} = \begin{pmatrix} A \\ B_s \end{pmatrix}$ and $\tilde{b} = \begin{pmatrix} b \\ 0 \end{pmatrix}$.

Here, B_s is a $k \times n$ matrix (with $k \leq n$) where in each row, there is exactly one nonzero entry $s > 0$ and at most one entry in every column, for example,

$$B_s = \begin{pmatrix} s & 0 & 0 & 0 & 0 \\ 0 & s & 0 & 0 & 0 \\ 0 & 0 & 0 & s & 0 \end{pmatrix}. \tag{2.23}$$

Let I_B be the set of column indices of the nonzero entries in B_s (in the example $I_B = \{1, 2, 4\}$). By adding additional rows to A, we can achieve that $\tilde{A}$ has full rank, and accordingly, the modified minimization problem (2.22) can be solved via a QR decomposition of $\tilde{A}$.

There are several advantages of using a singular value decomposition for solving the original problem (2.21):

1. If A is regular, $\delta^* = A^{-1}b$ and $r^* = 0$.
2. If A is not regular, $\|\delta^*\|_2$ is minimal among all solutions.
3. By eliminating small singular values, the solution process can be stabilized.

The main disadvantage of using SVD is the computational effort. Typically, the singular value decomposition is computed in two steps. First, the matrix is reduced to a bidiagonal matrix, and afterwards, the SVD of the bidiagonal matrix is computed by an iterative method up to a certain precision. In one of our real life applications, we have a 766×766 matrix with 1774 nonzero entries. For the computation of the SVD, 9.68 seconds are needed using MATLAB [3].

For the same example, the QR factorization of the corresponding modified matrix (1018×766 with 2026 nonzero entries) takes only 13 milliseconds. Hence, from the computational point of view, we prefer a QR decomposition to solve the modified problem (2.22) instead of solving (2.21) with an SVD. The results computed for the modified task (2.22) have to be measured in comparison to the three points given above. This is done in the following.

Let $\tilde{\delta}^*$ be the unique solution of (2.22) and $\tilde{r}^* = A\tilde{\delta}^* - b$. With

$$f(\delta) = \|\tilde{A}\delta - \tilde{b}\|_2^2 = \|A\delta - b\|_2^2 + s^2 \sum_{j \in I_B} \delta_j^2 = \|A\delta - b\|_2^2 + s^2 \|\delta_{I_B}\|_2^2 \tag{2.24}$$

we have

$$f\big(\tilde{\delta}^*\big) \leq f\big(\delta^*\big). \tag{2.25}$$

Inequality (2.25) yields for the corresponding residuals

$$\big\|\tilde{r}^*\big\|_2^2 \leq \big\|r^*\big\|_2^2 + s^2\big(\big\|\delta_{I_B}^*\big\|_2^2 - \big\|\tilde{\delta}_{I_B}^*\big\|_2^2\big) \leq \big\|r^*\big\|_2^2 + s^2\big\|\delta_{I_B}^*\big\|_2^2. \tag{2.26}$$

Thus, the maximum deviation of the Euclidean norm of the residual term $\tilde{r}^*$ from the possible minimum $\|r^*\|_2$ is limited and can be reduced by reducing s. In particular, if A is regular (or at least b is in the range of A), we have $\|r^*\|_2 = 0$ and

$$\big\|\tilde{r}^*\big\|_2 \leq s\big\|\delta_{I_B}^*\big\|_2. \tag{2.27}$$

Moreover, we get in the regular case:

$$A\left(\tilde{\delta}^* - \delta^*\right) = \tilde{r}^* \quad \Leftrightarrow \quad \tilde{\delta}^* - \delta^* = A^{-1}\tilde{r}^*. \tag{2.28}$$

Taking the Euclidean norm on both sides yields

$$\left\|\tilde{\delta}^* - \delta^*\right\|_2 \leq \left\|A^{-1}\right\|_2 \left\|\tilde{r}^*\right\|_2 \overset{(2.27)}{\leq} \left\|A^{-1}\right\|_2 s \left\|\delta^*_{I_B}\right\|_2 \leq \left\|A^{-1}\right\|_2 s \left\|\delta^*\right\|_2 \tag{2.29}$$

and finally

$$\frac{\|\tilde{\delta}^* - \delta^*\|_2}{\|\delta^*\|_2} \leq s\left\|A^{-1}\right\|_2 \tag{2.30}$$

for the relative error of $\tilde{\delta}^*$ compared to $\delta^* = A^{-1}b$. Note that $\tilde{\delta}^* = \delta^*$ if $\|\delta^*\|_2 = 0$.

Since $\|\tilde{\delta}^*_{I_B} - \delta^*_{I_B}\|_2 \leq \|\tilde{\delta}^* - \delta^*\|_2$, we also get from (2.29):

$$\frac{\|\tilde{\delta}^*_{I_B} - \delta^*_{I_B}\|_2}{\|\delta^*_{I_B}\|_2} \leq s\left\|A^{-1}\right\|_2. \tag{2.31}$$

Similar to above, note that $\tilde{\delta}^*_{I_B} = \delta^*_{I_B}$ if $\|\delta^*_{I_B}\|_2 = 0$.

In addition to the given results, (2.25) also yields

$$\left\|\tilde{\delta}^*_{I_B}\right\|_2^2 \leq \left\|\delta^*_{I_B}\right\|_2^2 - \frac{1}{s^2}\underbrace{\left(\left\|\tilde{r}^*\right\|_2^2 - \left\|r^*\right\|_2^2\right)}_{\geq 0} \leq \left\|\delta^*_{I_B}\right\|_2^2. \tag{2.32}$$

This means that regarding the indices I_B of the "correction terms" $\tilde{\delta}^*$ and δ^*, the correction induced by the QR decomposition of the modified matrix $\tilde{A}$ is not greater than the one induced by the SVD of the original matrix A. This is an important property since the set of indices I_B typically refers to "critical" variables of the problem, while the rest of the variables is supposed to be determined anyway.

So far, we have given quantitative results for our QR decomposition approach related to the first two advantages of using singular value decomposition. To give a quantitative result related to the third point, we consider the case $I_B = \{1, \ldots, n\}$ with $B_s = sI_n$, where I_n is the n-dimensional identity matrix.

Let $A = U\Sigma V^T$ be a singular value decomposition of A with

$$\Sigma = \begin{pmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_n \end{pmatrix}. \tag{2.33}$$

For the modified matrix $\tilde{A}$ we get

$$\tilde{A}^T\tilde{A} = \begin{pmatrix} A^T & B_s^T \end{pmatrix}\begin{pmatrix} A \\ B_s \end{pmatrix} = A^TA + B_s^2 = V\left(\Sigma^2 + B_s^2\right)V^T. \tag{2.34}$$

Hence, the singular values $\tilde{\sigma}_j$ $(j = 1, \ldots, n)$ of $\tilde{A}$ are given by

$$\tilde{\sigma}_j^2 = \sigma_j^2 + s^2. \tag{2.35}$$

In particular, we have $\tilde{\sigma}_j > \sigma_j$ and $\tilde{\sigma}_j \geq s > 0$.

While in the results above a smaller s is always preferred, here, the opposite is the case since an increase of the singular values leads to more stability. Thus, in practice, a trade-off has to be made.

2.3.3 Theoretical Analysis—Backward Direction

Let $f(y, u)$ be a (scalar) quantity of interest. As described in Sect. 2.2, we can efficiently compute sensitivity information with respect to the control u by solving adjoint equations. Due to the special structure of E, this can also be done time step wise, but backwards in time, and we finally have to solve linear systems of equations with the same matrices as in the forward direction, but transposed.

Thus, we have to solve systems of the form

$$A^T \xi = c. \tag{2.36}$$

In the whole section, we postulate that c is in the range of A^T. This has the following reason: The solution of the simulation process has degrees of freedom in the kernel $\ker(A)$ of A. Thus, regarding the quantity of interest f, it is reasonable to claim that the partial derivatives of f with respect to the state variables (in each time step) are perpendicular to $\ker(A)$, which is equivalent to being in the range of A^T. Additionally to the partial derivatives of f, c also may contain components from the preceding time step. This can only occur in parts of the network where the model contains temporal derivatives, but those parts do not suffer from the described problem of non-uniqueness since the state variables of consecutive time steps are linked here.

Let ξ^* be the solution of (2.36) of minimal Euclidean norm. Similar to Sect. 2.3.2, this can be computed by a singular value decomposition of A respectively A^T. It is natural to apply the QR decomposition of $\tilde{A}$ to solve the modified problem

$$\tilde{A}^T \begin{pmatrix} \xi \\ \mu \end{pmatrix} = c. \tag{2.37}$$

With the QR decomposition

$$\tilde{A} = \begin{pmatrix} \tilde{Q}_{11} & \tilde{Q}_{12} \\ \tilde{Q}_{21} & \tilde{Q}_{22} \end{pmatrix} \begin{pmatrix} \tilde{R} \\ 0 \end{pmatrix}, \tag{2.38}$$

where $\tilde{Q}_{11}$ and $\tilde{R}$ are $n \times n$ matrices, the solution of (2.37) of minimal norm can be written as

$$\begin{pmatrix} \tilde{\xi}^* \\ \tilde{\mu}^* \end{pmatrix} = \begin{pmatrix} \tilde{Q}_{11} \\ \tilde{Q}_{21} \end{pmatrix} \tilde{R}^{-T} c. \tag{2.39}$$

This results from the well-known fact that the columns of $\begin{pmatrix} \tilde{Q}_{12} \\ \tilde{Q}_{22} \end{pmatrix}$ form a basis of the kernel of $\tilde{A}^T$. With $\tilde{q}^* = A^T \tilde{\xi}^* - c$ we have

$$\|\tilde{\xi}^*\|_2^2 + \tfrac{1}{s^2}\|\tilde{q}^*\|_2^2 = \|\tilde{\xi}^*\|_2^2 + \|\tilde{\mu}^*\|_2^2 = \left\| \begin{pmatrix} \tilde{\xi}^* \\ \tilde{\mu}^* \end{pmatrix} \right\|_2^2 \leq \left\| \begin{pmatrix} \xi^* \\ 0 \end{pmatrix} \right\|_2^2 = \|\xi^*\|_2^2. \tag{2.40}$$

This yields

$$\|\tilde{q}^*\|_2^2 \leq s^2 \left(\|\xi^*\|_2^2 - \|\tilde{\xi}^*\|_2^2 \right) \leq s^2 \|\xi^*\|_2^2 \tag{2.41}$$

as upper bound for the residual with respect to the original equation (2.36). Analogously to Sect. 2.3.2, we get in the regular case

$$\frac{\|\tilde{\xi}^* - \xi^*\|_2}{\|\xi^*\|_2} \leq s \|A^{-T}\|_2 \tag{2.42}$$

for the relative error of $\tilde{\xi}^*$ compared to $\xi^* = A^{-T} c$.

References

1. KANET. http://kanet.iwg.uni-karlsruhe.de
2. STANET. http://www.stafu.de
3. MATLAB 7.5. The MathWorks Inc. (2007)
4. R. Álvarez, N.B. Gorev, I.F. Kodzhespirova, Y. Kovalenko, S. Negrete, A. Ramos, J.J. Rivera, Pseudotransient continuation method in extended period simulation of water distribution systems. J. Hydraul. Eng. **134**(10), 1473–1479 (2008)
5. R.H. Byrd, J. Nocedal, R.A. Waltz, Knitro: An integrated package for nonlinear optimization, in *Large-Scale Nonlinear Optimization* (2006), pp. 35–59
6. H. Cross, Analysis of flow in networks of conduits or conductors. University of Illinois Bulletin No. 286, 1936
7. T.A. Davis, *Direct Methods for Sparse Linear Systems*, Fundamentals of Algorithms, vol. 2 (Society for Industrial and Applied Mathematics, Philadelphia, 2006)
8. J. Deuerlein, Zur hydraulischen Systemanalyse von Wasserversorgungsnetzen, PhD thesis, U Karlsruhe, 2002
9. J. Deuerlein, A.R. Simpson, E. Gross, The never ending story of modeling control-devices in hydraulic systems analysis, in *Proceedings of Water Distribution Systems Analysis*, ASCE, 2008, p. 72
10. C. Hähnlein, Numerische Modellierung zur Betriebsoptimierung von Wasserverteilnetzen, PhD thesis, TU Darmstadt, 2008
11. O. Kolb, P. Domschke, J. Lang, Modified QR decomposition to avoid non-uniqueness in water supply networks with extension to adjoint calculus. Proc. Comput. Sci. **1**(1), 1421–1428 (2010)

12. O. Kolb, J. Lang, P. Bales, An implicit box scheme for subsonic compressible flow with dissipative source term. Numer. Algorithms **53**(2), 293–307 (2010)
13. S.N. Kružkov, First order quasilinear equations in several independent variables. Math. USSR Sb. **10**(2), 217–243 (1970)
14. R.J. LeVeque, *Finite Volume Methods for Hyperbolic Problems* (Cambridge University Press, Cambridge, 2002)
15. L.A. Rossman, *EPANET 2 Users Manual* (U.S. Environmental Protection Agency, Cincinnati, 2000)
16. A.R. Simpson, Modeling of pressure regulating devices: The last major problem to be solved in hydraulic simulation, in *Proceedings of 29th Annual Water Resources Planning and Management Conference*, ASCE, 1999
17. P. Spellucci, A new technique for inconsistent QP problems in the SQP method. Math. Methods Oper. Res. **47**(3), 355–400 (1998)
18. P. Spellucci, An SQP method for general nonlinear programs using only equality constrained subproblems. Math. Program. **82**(3), 413–448 (1998)
19. A. Wächter, L.T. Biegler, On the implementation of a primal-dual interior point filter line search algorithm for large-scale nonlinear programming. Math. Program. **106**(1), 25–57 (2006)

O. Kolb · J. Lang
Numerical Analysis and Scientific Computing, Technische Universität Darmstadt, Dolivostr. 15, 64293 Darmstadt, Germany

O. Kolb
e-mail: kolb@mathematik.tu-darmstadt.de

J. Lang (✉)
e-mail: lang@mathematik.tu-darmstadt.de

Chapter 3
Mixed Integer Optimization of Water Supply Networks

Antonio Morsi, Björn Geißler, and Alexander Martin

Abstract We introduce a mixed integer linear modeling approach for the optimization of dynamic water supply networks based on the piecewise linearization of nonlinear constraints. One advantage of applying mixed integer linear techniques is that these methods are nowadays very mature, that is, they are fast, robust, and are able to solve problems with up to a huge number of variables. The other major point is that these methods have the potential of finding globally optimal solutions or at least to provide guarantees of the solution quality. We demonstrate the applicability of our approach on examples networks.

3.1 Introduction

In this chapter we consider the optimization of dynamical water transport network management. The arcs or connections of a water supply network correspond to pipes or certain components such as valves or pumps. Within these components the physical laws describing the behavior involve partial differential equations, differential equations and nonlinear constraints. Additionally, such networks contain some kinds of switching components, that is, components which can be in different discrete states. In order to reflect discrete processes in a mathematical model, one typically introduces binary variables and maybe some additional constraints involving these variables. Dealing with (partial) differential equations in water network optimization problems tends in general to solving a discretized system of nonlinear constraints. One disadvantage of nonlinear programs is that they are only able to guarantee locally optimal solutions. Altogether, we are interested in globally optimal solutions of a mixed integer nonlinear optimization problem. In the presence of binary variables classical algorithms have to solve a (probably huge) number of such nonlinear programs to global optimality, which leads us to an intractable state of the art problem for realistic instances. In contrast, our approach is based on piecewise linearization of nonlinearities and the modeling of piecewise linear functions in terms of linear constraints. This brings us into a situation where we can apply global optimization algorithms from mixed integer linear programming to the optimization of water transport networks. Besides dynamic water supply network optimization, many engineering problems like supply chain network optimization, traffic problems or the problem of gas network optimization can be stated as such

A. Martin et al. (eds.), *Mathematical Optimization of Water Networks*,
International Series of Numerical Mathematics 162,
DOI 10.1007/978-3-0348-0436-3_3,

an optimization problem on a dynamic transport network and can thus be handled similarly [2, 3].

The remainder of this chapter is organized as follows: In Sect. 3.2, we introduce a network model suitable for dynamic water supply networks. In Sect. 3.3, we remind of the dynamics of fresh water in pipeline networks, which are mainly described by a coupled set of hyperbolic partial differential equations. In Sect. 3.4, we introduce our mixed integer nonlinear model for the optimization of water supply network management. Section 3.5 is dedicated to the approximation of nonlinearities by piecewise linear functions and their modeling in terms of linear constraints. Finally, we present some computational results in Sect. 3.6 before we conclude with a summary in Sect. 3.7. Note that we present in Chap. 3 further computational results of our approach in combination with results from the continuous nonlinear programming methods of Chap. 2.

3.2 Basic Network Model

We model a dynamic water transport network by means of a directed finite graph $G = (V, A)$. The set A of arcs is partitioned into different sets for the various components in the network, that is, a set of pipes A_P, a set of tanks A_T, a set of pumps A_U and a set of valves A_V.

The set V of vertices or nodes consists of a set V_I of intersection points of the segments (also called inner nodes) and a set of boundary nodes V_B, which further divides into a set V_S of suppliers, and a set V_C of consumers. Suppliers are considered as water delivering points and consumers reflect demands specified by the quantities flow and pressure. We denote the set of outgoing (ingoing) arcs from (to) the node $v \in V$ by δ_v^+ (δ_v^-).

The task is to route the water through the network to satisfy the consumers' demands such that the costs for the control elements, that is power consumption of the pumps, is minimized. Other objectives might be chosen to reflect real world applications more accurate, e.g. the minimization of a sum of electricity costs of pumps and water quality costs for supplied water from different vendors. A kind of supply guarantee, which is mapped to the adherence of a minimum tank level at all times, could be an alternative objective.

3.3 Flow in Pipelines

In water supply networks, we are dealing with pressurized water networks. Due to the incompressibility of water, pressure p can equivalently be expressed as elevation difference

$$\Delta h = \frac{p}{g\rho}, \tag{3.1}$$

where g is the gravitational acceleration and ρ is the constant water density. In water management, pressure is therefore often measured by the elevation above sea level, called the head h, which is the sum of the actual geodetic height and the elevation difference corresponding to the hydraulic pressure. Thus, a pressure of 5 bar at 100 m above sea level corresponds to a head $h \approx 151$ m. For this kind of network, the governing equations in all pipes $a \in A_P$ are the so-called water hammer equations [1]:

$$\frac{\partial h}{\partial t} + \frac{c^2}{gA}\frac{\partial Q}{\partial x} = 0, \tag{3.2}$$

$$\frac{\partial Q}{\partial t} + gA\frac{\partial h}{\partial x} = -\lambda \frac{Q|Q|}{2DA}, \tag{3.3}$$

where (h, Q) is the state vector consisting of the piezometric head and the flow. Here, g is the gravitational acceleration, c is the speed of sound in the pipe, A and D are the cross-sectional area and the diameter of the pipe. The term on the right-hand side of (3.3) models the influence of friction. The friction coefficient λ is implicitly given by the phenomenological formula of Prandtl-Colebrook (3.4),

$$\frac{1}{\sqrt{\lambda}} = -2\log_{10}\left(\frac{2.51}{\mathrm{Re}\sqrt{\lambda}} + \frac{k}{3.71D}\right), \tag{3.4}$$

where k is the roughness of the pipe. The variable Re is called Reynolds number and is defined by

$$\mathrm{Re} = \frac{D\rho|Q|}{\eta A}, \tag{3.5}$$

with dynamic viscosity of water η. The Reynolds number is used to characterize different flow regimes, such as laminar or turbulent flow. Laminar flow occurs at low Reynolds numbers and is characterized by smooth, constant fluid motion, while turbulent flow occurs at high Reynolds numbers, which tend to produce chaotic eddies and vortices. We should mention that the friction factor given by the formula of Prandtl-Colebrook (3.4) is only valid for turbulent flow. As any realistic flow within a pressurized fresh water transport system can be seen as turbulent, we neglect cases of laminar and transitional flow. However, from a mathematical point of view, including the non-turbulent flow case is no problem for our model and algorithms.

3.4 A Model for Dynamic Water Supply Network Optimization

To provide a basis for our mixed integer model we must attend to two main issues. First we need an appropriate finite-dimensional model to the water hammer equations (3.2), (3.3), which makes them suitable for numerical evaluation. As an adequate discretization scheme, we choose an implicit box schemes, which is known to work effectively, especially when using large step sizes. Large discretization step

sizes are one major point, since our goal is to solve a mixed integer program, where each discretization point introduces new variables (probably a huge number). Originally, box schemes have been introduced by Wendroff [12]. A survey of this scheme in combination with the water hammer equations can be found in Chap. 2. After a finite-dimensional problem, that is, a mixed integer nonlinear program, is obtained, the second main issue is how to deal with nonlinearities. Our approach is based on piecewise linearization of nonlinearities and the modeling of these piecewise linear functions in terms of linear constraints.

For our hydraulic model, we choose head variables h_v^n for each node $v \in V$ and flow variables for each arc $a \in A$ and time step $t_n \in \{t_0, \ldots, t_N\}$. Since a dynamic model, cannot assume to have constant flow within a pipe, we bring in two additional variables $Q_{a,v}^n$ and $Q_{a,w}^n$ representing the flow at the beginning and the end of a pipe $a = (v, w) \in A_P$ at time step $t_n \in \{t_0, \ldots, t_N\}$. Except for pipes, all other arcs are modeled to have no length, and so, we introduce only one flow variable Q_a^n per arc $a \in A \setminus A_P$ at time t_n. Each head and flow variable is bounded from below and above by some constants $\underline{h}_v^n$, $\overline{h}_v^n$ for $v \in V$, $\underline{Q}_{a,v}^n$, $\overline{Q}_{a,w}^n$ for $a \in A_P$, and $\underline{Q}_a^n$, $\overline{Q}_a^n$ for $a \in A \setminus A_P$ at time $t_n \in \{t_0, \ldots, t_N\}$. Typically these bounds represent some technical restrictions that are independent of time. A backward flow is indicated by a negative flow on a directed arc. The water supply into a network is represented by non-positive flow demand variables d_v^n for each source $v \in V_S$ and time step $t_n \in \{t_0, \ldots, t_N\}$. In the same manner, water withdrawal at each sink $v \in V_D$ is described by non-negative flow demand variables d_v^n for each time step $t_n = t_0, \ldots, t_N$.

3.4.1 Pipes

As we have seen, pipe dynamics are described by the water hammer equations (3.2) and (3.3). Applying the implicit box scheme from Chap. 2 to these partial differential equations yields the discretized equations

$$\frac{h_v^{n+1} + h_w^{n+1}}{2\tau_n} - \frac{h_v^n + h_w^n}{2\tau_n} + \frac{c^2}{gA} \cdot \frac{Q_{a,w}^{n+1} - Q_{a,v}^{n+1}}{L_a} = 0, \tag{3.6}$$

$$\begin{aligned} &\frac{Q_{a,v}^{n+1} + Q_{a,w}^{n+1}}{2\tau_n} - \frac{Q_{a,v}^n + Q_{a,w}^n}{2\tau_n} + gA\frac{h_w^{n+1} - h_v^{n+1}}{L_a} \\ &\quad = -\frac{1}{2DA}\left(\frac{\lambda(|Q_{a,v}^{n+1}|)Q_{a,v}^{n+1}|Q_{a,v}^{n+1}|}{2} + \frac{\lambda(|Q_{a,w}^{n+1}|)Q_{a,w}^{n+1}|Q_{a,w}^{n+1}|}{2}\right). \end{aligned} \tag{3.7}$$

Here, $\tau_n = t_{n+1} - t_n$ is the time step size. We choose a spatial step size equal to the length L_a of pipe a. Fortunately, the discretized continuity equation (3.6) is linear, so we can include it directly into our mixed integer linear program. The momentum equation (3.7) however contains nonlinear expressions. Therefore, we have to approximate the nonlinear terms by piecewise linear ones to incorporate them into a mixed integer model. We refer to Sect. 3.5 for further details on how

we build these piecewise linear approximations and how we include them into our model.

3.4.2 Tanks

In water supply networks, tanks are used for intermediate storage of water. Roughly speaking, tanks are filled during periods of low demand and emptied during the peak demand periods, although in general the situation is not as simple as that. Mainly for optimal controls this behavior might differ. In our model, the node w, corresponding to the head node w_a of a tank $a = (v, w) \in A_T$ is of degree one. This node w (and of course w_a) can be seen as physical point inside the tank, whereas v and v_a represent the physical situation in front of it. At first glance this model might be a little bit confusing, but it has the main advantage that we can represent tanks as arcs in our graph. Using the assumption that tanks have constant cross-sectional area A_a, we can derive the following relationship for the change of a tank's filling level over time:

$$\frac{d}{dt} h_w(t) = \frac{1}{A_a} Q_a(t). \tag{3.8}$$

A positive flow $Q_a(t)$ can be imagined as tank filling at time t, in contrast a negative flow represents an outflow. To integrate this ordinary differential equation into our model, we apply the implicit Euler discretization scheme and get the equations

$$\frac{h_w^{n+1} - h_w^n}{\tau_n} = \frac{1}{A_a} Q_a^{n+1} \tag{3.9}$$

for each time step. Furthermore, the discharge law

$$Q_a = C_a \operatorname{sgn}(h_v - h_w)\sqrt{|h_v - h_w|} \quad \text{or equivalently} \tag{3.10}$$

$$Q_a|Q_a| = C_a^2(h_v - h_w) \tag{3.11}$$

must hold. Here, C_a is a tank specific discharge coefficient. Since this law must hold at each point in time $t_n \in \{t_1, \ldots, t_N\}$, we get the system of equations

$$Q_a^n \left| Q_a^n \right| = C_a^2 \left(h_v^n - h_w^n \right). \tag{3.12}$$

Finally, we piecewise linearize the nonlinear terms $Q_a^n|Q_a^n|$ and add these approximated version of (3.12) to our mixed integer linear program.

3.4.3 Pumps

In pressurized networks, water is distributed by the fact that water flows from points of high pressure to points of lower pressure. Hence, one must increase the pressure

at certain parts of the network, e.g., when extracting ground-water or when water is transported uphill. Pumps are used to increase the pressure inside a water supply network. First of all, we introduce a binary variable s_a^n, which indicates if pump $a \in A_U$ is active or shut down at time t_n. Every active pump $a = (v, w) \in A_U$ increases the pressure by some controlled non-negative amount $\Delta h_a = h_w - h_v$. Generally, in water networks there are two basic types of pumps. Pumps can have fixed speed. Then, we can describe its pressure gain by the characteristic pump curve

$$\Delta h_a(t) = \alpha_a - \beta_a \cdot Q_a(t)^{\gamma_a}. \tag{3.13}$$

Here $\alpha_a > 0$ is the maximum possible pressure increase of the pump. The efficiency parameters $\beta_a > 0$ and $\gamma_a \geq 1$ are also pump-specific. An inactive/closed pump of either type must have a flow rate $Q_a = 0$ and its pressure differential Δh_a can be arbitrarily. In the case of fixed speed pumps, this is modeled by the two constraints

$$\underline{M}_a\big(1 - s_a^n\big) \leq \Delta h_a^n - \big(\alpha_a - \beta_a \cdot \big(Q_a^n\big)^{\gamma_a}\big) \leq \overline{M}_a\big(1 - s_a^n\big), \tag{3.14}$$

for appropriately chosen constants $\underline{M}_a$ and $\overline{M}_a$, e.g., $\underline{M}_a = h_w^{min} - h_v^{max} - \alpha_a$ and $\overline{M}_a = h_w^{max} - h_v^{min} - \alpha_a$. Again, in the case of $\gamma_a \neq 1$, we get a nonlinear term, which we approximate by a piecewise linearization. Obviously, the discretized version of (3.13) holds, if the pump is running, that is, $s_a^n = 1$ and the pressure difference Δh_a^n of the pump is arbitrary, if the pump is shut down, that is, $s_a^n = 0$ and $q_a^n = 0$. Now, we have to include additional constraints linking the values of Q_a^n and s_a^n.

$$\varepsilon s_a^n \leq Q_a^n, \tag{3.15}$$

$$Q_a^n \leq Q_a^{max} s_a^n + \varepsilon. \tag{3.16}$$

We can choose the parameter $\varepsilon > 0$ to be the minimal relevant non-zero flow. By constraint (3.15), a flow of less than ε implies the pump to be inactive ($s_a^n = 0$) and a positive flow of more than ε forces $s_a^n = 1$ by (3.16).

Other pumps can operate at variable speed. The technical model of these pumps involves the non-dimensional relative speed $\omega_a(t)$ of a pump $a \in A_U$ at time t. Then, the pressure increase Δh_a of a variable speed pump depends on two variables, the flow rate $Q_a(t)$ and the relative speed $\omega_a(t)$:

$$\Delta h_a(t) = \omega_a(t)^2 \left(\alpha_a - \beta_a \cdot \left(\frac{Q_a(t)}{\omega_a(t)}\right)^{\gamma_a}\right). \tag{3.17}$$

As we see, fixed speed pumps are a special case of variable speed pumps with constant relative speed $\omega_a(t) \equiv 1$. Indeed, we model variable speed pumps in exactly the same manner as fixed speed pumps, which reads as

$$\underline{M}_a\big(1 - s_a^n\big) \leq \Delta h_a^n - \left(\big(\omega_a^n\big)^2 \left(\alpha_a - \beta_a \cdot \left(\frac{Q_a^n}{\omega_a^n}\right)^{\gamma_a}\right)\right) \leq \overline{M}_a\big(1 - s_a^n\big). \tag{3.18}$$

Again, we have to replace the nonlinear terms by piecewise linear ones in order to include inequalities (3.18) in our mixed integer model.

3.4.4 Valves

In water supply networks four major kinds of valves are installed. The simplest type is a check valve (CV). A check valve is used to avoid backward flow, which can be modeled by $Q_a^n \geq 0$ for all $a \in A_{CV}$ and all time steps. The other valves have in common that they are controllable. For that purpose, we introduce binary variables $s_a^n \in \{0, 1\}$ for each valve $a \in A_V \setminus A_{CV}$ and time step $t_n \in \{t_0, \ldots, t_N\}$. The first class of controllable valves is called gate valves (GV). The flow through a gate valve is zero if it is closed and untouched otherwise. This condition is modeled by introducing

$$\underline{Q}_a s_a^n \leq Q_a^n \leq \overline{Q}_a s_a^n \quad \text{and} \tag{3.19}$$

$$\underline{M}_a\left(1 - s_a^n\right) \leq h_w^n - h_v^n \leq \overline{M}_a\left(1 - s_a^n\right), \tag{3.20}$$

for each gate valve $a = (v, w) \in A_{GV}$. Constraint (3.19) forces the flow rate Q_a^n to be zero, when the valve is closed ($s_a^n = 0$). For an opened valve, it follows from inequalities (3.20) that the head differential is zero. For appropriately chosen constants $\underline{M}_a$ and $\overline{M}_a$, for example $\underline{M}_a = \underline{h}_w - \overline{h}_v$ and $\overline{M}_a = \overline{h}_w - \underline{h}_v$, the pressure differential remains unaffected when the valve is closed. Another kind of valve, the flow control valve (FCV) is used to restrict the flow rate to be at most $\overline{Q}_a$ via

$$\varepsilon\left(1 - s_a^n\right) + Q_a^n \leq \overline{Q}_a \quad \text{and} \tag{3.21}$$

$$\overline{Q}_a s_a^n \leq Q_a^n. \tag{3.22}$$

For an ε indicating the smallest possible amount of positive flow, inequality (3.21) forces the flow control valve to be in an active state ($s_a^n = 1$) if $Q_a^n = \overline{Q}_a$ and becomes redundant for any $0 \leq Q_a^n < \overline{Q}_a$. By inequality (3.22), the valve is set to an open state ($s_a^n = 0$) if $Q_a^n < \overline{Q}_a$ and becomes redundant in the case $Q_a^n = \overline{Q}_a$. Note that these valves' states, in contrast to all others, depend on the flow only. Thus, the binary variables s_a^n may be viewed as state variables and not as control variables. In our model, an inactive flow control valve $a = (v, w) \in A_{FCV}$ causes no pressure loss, whereas this condition does not hold for active valves. This is modeled by

$$0 \leq h_v^n - h_w^n \leq M_a s_a^n, \tag{3.23}$$

for an appropriately large constant number M_a, e.g., $M_a = \overline{h}_v - \underline{h}_w$. So far, the presented valves only control the flow rate. Another class of valves is used to control the pressure—particularly pressure breaker valves (PBV). An active pressure breaker valve $a = (v, w) \in A_{PBV}$ ($s_a^n = 1$) provides a fixed pressure differential Δh at time t_n, that is,

$$\Delta h s_a^n \leq h_v^n - h_w^n \leq \Delta h \quad \text{and} \tag{3.24}$$

$$\underline{Q}_a s_a^n \leq Q_a^n \leq \overline{Q}_a s_a^n. \tag{3.25}$$

3.4.5 Flow Conservation

As introduced in Sect. 3.2 there are three different kinds of nodes in water supply networks. On the one hand, nodes are able to deliver or conduct water, but on the other hand they can be just intersection points of adjacent arcs. These different kind of tasks are summarized in our model by introducing a continuous variable d_v^n describing the demand of a node $v \in V$ at time $t_n \in \{t_1, \ldots, t_N\}$. Typically, a node does not change its type over time, such that we can unambiguously separate the set of nodes to suppliers or consumers. This partition is not essential, but it explains why the property being a supplier, consumer or junction is assigned to nodes and not to nodes at a given time. A consumer is identified by non-negative values of d_v^n, whereas suppliers have non-positive demand values d_v^n. For an accurate physical behavior within water supply network, we must ensure the conservation of mass at nodes. Formally, the conservation of mass of node $v \in V$ at time $t_n \in \{t_1, \ldots, t_N\}$ is expressed by

$$\sum_{a \in \delta_v^- \cap A_P} Q_{a,v}^n + \sum_{a \in \delta_v^- \setminus A_P} Q_a^n - \sum_{a \in \delta_v^+ \cap A_P} Q_{a,v}^n - \sum_{a \in \delta_v^+ \setminus A_P} Q_a^n = d_v^n. \tag{3.26}$$

Due to the instationary behavior of pipes we have to differentiate between pipes and the remaining components in (3.26). The outflow d_v^n of a node $v \in V$ at time $t_n \in \{t_1, \ldots, t_N\}$ is bounded by some time-dependent constants

$$\underline{d}_v^n \leq d_v^n \leq \overline{d}_v^n. \tag{3.27}$$

It is not unusual that $\underline{d}_v^n = \overline{d}_v^n$, that is, the demand is fixed for a certain time. Indeed the demand is fixed in general for consumers, whereas suppliers often have fixed head values to reflect the withdrawal from reservoirs or groundwater.

3.4.6 Further Transient Conditions

In addition to the behavior of different components described so far, we have to consider further conditions in our transient model. First, there are restrictions on the minimum runtime and downtime of each pump. Second, for practical purposes it is a good idea to include a terminal condition for the overall water volume flow of a network.

Consider a pump $a \in A_U$ with constant minimum runtime $R_a \in \mathbb{N}$ and downtime $r_a \in \mathbb{N}$. The adherence of a minimum runtime of a pump is then modeled by the inequalities

$$s_a^n - s_a^{n-1} \leq s_a^m, \quad \text{for } n+1 \leq m \leq \min\{n + R_a - 1, N\}. \tag{3.28}$$

The minimum downtime is modeled analogously by introducing the constraints

$$s_a^{n-1} - s_a^n \leq 1 - s_a^m, \quad \text{for } n+1 \leq m \leq \min\{n + r_a - 1, N\}. \tag{3.29}$$

We remark that a complete linear description of the polytope defined by (3.28) and (3.29) as well as an appropriate separation algorithm can be found in [8].

Since our optimization process is restricted to a finite time horizon, we have to ensure operational availability of the water network at the end of the considered time span. If we do not include a terminal condition, the optimization process tends to produce very low tank levels in the final time step. This might be a problem when the following time horizon is processed. To overcome this issue one could stipulate a minimum tank level in the last time step that ensures an operational network in the next horizon. More potential for optimization will be obtained by restricting the minimum volume of water in the last time step. The volume of water within a tank is calculated by

$$V_a^n = A_a\big(h_a^n - e_a\big), \tag{3.30}$$

where h_a is the head, A_a is the area and e_a is the elevation of tank a. We require the total water volume at the end of the considered time horizon to be at least as large as at the beginning. This can be modeled by the following linear inequality:

$$\sum_{a\in A_T} V_a^0 \le \sum_{a\in A_T} V_a^N. \tag{3.31}$$

Please note that the optimization process tends to fulfill (3.31) at equality, since transporting water through the network causes costs.

Another major point during the operational management of water supply networks is to ensure that delivered water is kept freshly at all time. For this reason on the one hand stagnant water in tanks has to be avoided and on the other hand tanks have to kept clean to prevent deposits. To reduce those problems a condition called breathing is claimed in practice. Breathing means that a tank has to be flushed and refilled a certain number of times during a time period. To express one part of that condition, namely the flushing, we introduce auxiliary binary variables b_a^n for each tank $a \in A_T$ and time step t_n and add the constraints

$$\underline{m}_a b_a^n \le h_a^n - h_a^{min} \le \overline{m}_a\big(1 - b_a^n\big) \tag{3.32}$$

and

$$\sum_{n=0}^{N} b_a^n \ge \underline{\xi}. \tag{3.33}$$

With appropriately chosen constants $\underline{m}_a := \underline{h}_a^n - h_a^{min}$ and $\overline{m}_a := \overline{h}_a^n - h_a^{min}$ a binary variable b_a^n is forced to one, if the tank level is less than or equal to a desired minimum level h_a^{min} within time step n, through (3.32). Otherwise the variable b_a^n is forced to zero. The requirement that water falls below the desired discharge level at least $\underline{\xi}$ times is obtained by (3.32). Analogously, the achievement that a tank level must raise $\overline{\xi}$ times to a desired maximum h_a^{max} is enforced.

3.4.7 Optimization Task

In the optimization of water distribution networks, a common objective is the fulfillment of consumer demands (w.r.t. flow and head) with minimal costs. The costs mainly arise from electric power consumption of pumps. The power consumption P_a of a single pump $a \in A_U$ of either class (fixed or variable speed) can be described by an equation of the form

$$P_a(t) = \omega_a(t)^3 \left(\bar{\alpha}_a - \bar{\beta}_a \cdot \frac{q_a(t)}{\omega_a(t)} \right), \tag{3.34}$$

which depends on the relative speed $\omega_a(t)$ and the throughput $q_a(t)$. The discretized version then reads as

$$P_a^n = \left(\omega_a^n\right)^3 \left(\bar{\alpha}_a - \bar{\beta}_a \cdot \frac{q_a^n}{\omega_a^n} \right). \tag{3.35}$$

Now, we have to consider that a pump has relative speed equal to zero (and hence no power consumption) if it is shut down. Therefore, we add the condition

$$\omega_a^n \leq s_a^n \overline{w}_a. \tag{3.36}$$

This inequality forces ω_a^n to be zero for inactive pumps and leads to $\omega_a^n \leq \overline{w}_a$ for a working pump. Again, we have to build piecewise linear approximations of the nonlinearities in (3.35). We remark that for fixed speed pumps, that is $\omega_a^n \in \{0, 1\}$, (3.35) is even linear. Finally, the minimum power consumption over the optimization time horizon $[t_0, t_N]$ is given by

$$\min \sum_{a \in A_U} \int_{t_0}^{t_N} P_a(t) \, \mathrm{d}t, \tag{3.37}$$

subject to all continuous state conditions describing the instationary behavior of (h, q). For our time-discretized model, we obtain

$$\min \sum_{a \in A_U} \sum_{n=0}^{N-1} \frac{\tau_n}{2} \left(P_a^n + P_a^{n+1} \right), \tag{3.38}$$

with time step size $\tau_n = t_{n+1} - t_n$, by applying the trapezoidal rule to (3.37).

3.5 Piecewise Linearization

In Sect. 3.4, we introduced the discretized momentum equation (3.7) as well as the constraints describing the pressure increase (3.18) and a pump's power consumption (3.35). All these equations involve nonlinear terms. In order to reflect the effects of these equations in a linear model, we build piecewise linearizations of all nonlinear

terms. In case of univariate nonlinear expressions, like in the discretized continuity equation, this leads us to an univariate piecewise linear function. In general approximating the nonlinearities in the characteristic pump curve equation (3.17) needs a two-dimensional triangulation. One way to avoid two-dimensional grids is obtained by a decomposition of nonlinear terms into elementary ones, followed by the application of the Binomial theorem to bivariate term of the form xy. The equivalence from the Binomial theorem to rewrite bivariate terms of the form xy to univariate once is depicted by the two equations

$$xy = \frac{1}{2}\left(z^2 - x^2 - y^2\right), \tag{3.39}$$

$$z = x + y. \tag{3.40}$$

According to that formula a bivariate term xy is replaced by three univariate nonlinear terms. For example the characteristic pump curve equation (3.17) can be rewritten into univariate nonlinear terms by such an iterative process. There is no obvious answer to the question, which way is the most efficient one, that is, whether or not to rewrite bivariate terms into multiple univariate separable terms or not. In our model we use the univariate representation for all pure terms of the form xy, such as for $\gamma_a \in \{0, 1, 2\}$ in (3.17).

In order to make a point about the quality of such an approximation, we must be able to measure the linearization errors. Doing this a-posteriori is quite simple: We solve some MIP approximation to optimality and evaluate the nonlinear expressions at the point where the optimum is attained. Unfortunately, this is not enough, whenever we want the solution to satisfy some predefined error bounds. In this case we must be able to estimate the linearization errors a-priori in order to avoid solving MIPs of increasing complexity (due to finer linearization grids) over and over again. To guarantee an a-priori given error bound to a nonlinear function it is essential to estimate the distance of a nonlinear function to a linear function over a simplex. In general this might be difficult or even intractable. Thus, we overestimate this error by computing the distance to a convex underestimator and a concave overestimator of the function. To measure the error between a convex function and a linear function obtained by interpolation the convex underestimator is given by the function itself and the overestimation error is zero. On the one hand we are certainly interested in tight estimators, since each simplex leads to new variables in our mixed integer program as we see in the following section. On the other hand it might be difficult or even impossible to find the tightest estimators. In fact the tightest estimators even lead to the exact linearization error. Due to this tradeoff the practicality of our approach mainly depends on the ability to find tight estimators for the given nonlinear functions. Due to the fact that the given nonlinearities in our proposed water supply network model are not that nasty, we are able to determine the point of maximal linearization error either directly (e.g., for terms of the form $x|x|$) or to use estimators proposed in the literature, like [5, 6]. Once we found a convex underestimator, we can identify a point where the maximal linearization error is obtained, by solving a pure convex optimization problem. For more mathematical details about

this procedure to control the error of piecewise linear approximations to nonlinear functions we refer to [4]. Now that we are able to control the linearization error over a simplex, it is obviously straightforward to control the overall linearization error. We simply add a point of maximum error to the vertex set of our triangulation and retriangulate the affected region. Repeating this process iteratively for all not yet checked simplices, leads to a piecewise linearization satisfying an overall error bound.

An appropriate way to model univariate piecewise linearizations by mixed integer programming known as incremental model is shown in Sect. 3.5.1. Nevertheless, our model might still contain non-separable bivariate terms, e.g., if $\gamma_a \notin \{0, 1, 2\}$ in (3.17). A generalization of the incremental model to higher dimensions is introduced in Sect. 3.5.2. For alternative approaches to the proposed model and further details on higher dimensional functions we refer to [4, 11].

3.5.1 Mixed Integer Model of a Univariate Piecewise Linearization

In order to describe a method for the incorporation of a piecewise linear function into a mixed integer linear program, we consider some (univariate) continuous nonlinear function $\phi : \mathbb{R} \to \mathbb{R}$, $\mathbf{x} \mapsto \phi(\mathbf{x})$. We further assume that this function has been approximated by a piecewise linear function $\tilde{\phi}$ as described in the last paragraph. Our method of choice to model a piecewise linear function is the incremental method, sometimes also called delta-method [9]. This model makes use of the fact that any point x in an interval $[\overline{x}_{i-1}, \overline{x}_i]$ can be written as $\overline{x}_{i-1} + \delta_i$ for $\delta_i \in [0, \overline{x}_i - \overline{x}_{i-1}]$. The linear function value can then be expressed as $y = \overline{y}_{i-1} + \frac{\overline{y}_i - \overline{y}_{i-1}}{\overline{x}_i - \overline{x}_{i-1}} \delta_i$, where $y_i := \tilde{\phi}(x_i)$ and $y_{i-1} := \tilde{\phi}(x_{i-1})$ (see Fig. 3.1). For the description of a piecewise linear function with n line segments this representation is used in the incremental model by introducing variables δ_i for each interval i such that $0 \le \delta_i \le \overline{x}_i - \overline{x}_{i-1}$. Then binary auxiliary variables z_i are introduced for each interval except the last to force the so-called "filling condition", that is, the condition $\delta_i > 0$ implies that δ_{i-1} is at its upper bound. We obtain the following model

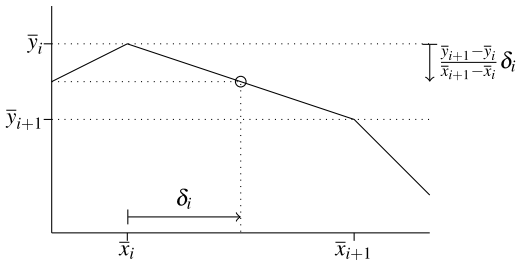
$$x = \overline{x}_0 + \sum_{i=1}^{n} \delta_i, \tag{3.41}$$

$$y = \overline{y}_0 + \sum_{i=1}^{n} \frac{\overline{y}_i - \overline{y}_{i-1}}{\overline{x}_i - \overline{x}_{i-1}} \delta_i, \tag{3.42}$$

$$\begin{aligned} (\overline{x}_{i-1} - \overline{x}_{i-2}) z_{i-1} &\le \delta_i \quad \text{for } i = 2, \dots, n, \\ \delta_i &\le (\overline{x}_i - \overline{x}_{i-1}) z_i \quad \text{for } i = 1, \dots, n-1, \\ z_i &\in \{0, 1\} \quad \text{for } i = 1, \dots, n-1. \end{aligned} \tag{3.43}$$

The validity of this model for any piecewise linear function defined by points $(x_i, y_i := \tilde{\phi}(x_i))$ for $i = 0, \dots, n$ can be seen as follows. Constraint (3.43) ensures

Fig. 3.1 Incremental method



that $\delta_k > 0$ forces all previous δ_i, that is, those with $i < k$, to their upper bounds. Through (3.42) all subsequent variables δ_i with $i > k$ are forced to zero. Hence $x \in [x_{k-1}, x_k]$ is described by

$$x = \overline{x}_0 + \sum_{i=1}^{n} \delta_i = \overline{x}_0 + \sum_{i=1}^{k-1} \delta_i + \delta_k = \overline{x}_{k-1} + \delta_k \tag{3.44}$$

as desired.

The quality of the incremental model was discussed by Padberg [10]. He studied the case where the objective function of a linear program is a piecewise linear function. In that case the formulation of the incremental method always yields an integral solution of the LP-relaxation whereas this is not the case for the standard textbook approach, the so-called convex combination method. In any case the polyhedron described by the incremental method is properly contained in the one described by the convex combination method. The method is generic and be readily incorporated in standard mixed integer linear models.

3.5.2 Mixed Integer Model of a Multivariate Piecewise Linearization

For the remainder of this section let $\phi : \mathbb{R}^d \to \mathbb{R}$ be an arbitrary continuous piecewise linear function. Thus the function ϕ does not necessarily have to be separable and of course it does not need to be convex and concave, respectively. First, we generalize the definition of piecewise linear functions to higher dimensions.

Definition 1 Let $\mathbb{D} \subset \mathbb{R}^d$ be a compact set. A continuous function $\phi : \mathbb{D} \to \mathbb{R}$ is called piecewise linear if it can be written in the form

$$\phi(x) = \phi_S(x) \quad \text{for } x \in S, \forall S \in \mathbb{S}, \tag{3.45}$$

with affine functions ϕ_S for a finite set of simplices $\mathbb{S}$ that partitions $\mathbb{D}$.

For the sake of simplicity we restrict ourselves to functions over compact domains, though some techniques can be applied to unbounded domains, too. Further-

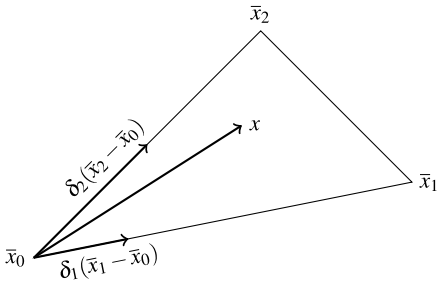


Fig. 3.2 A point $x = \overline{x}_0 + \delta_1(\overline{x}_1 - \overline{x}_0) + \delta_2(\overline{x}_2 - \overline{x}_0)$ inside a triangle

more our MIP techniques deal with continuous functions only. More information on a special class of discontinuous functions so-called lower semi-continuous functions can be found in [11]. In the literature it is common to be not as restrictive and require the domain S of each piece to be simply a polytope. However, since both definitions are equivalent and some of our approaches rely on simplicial pieces we go for the above definition.

According to Definition 1, we denote by $\mathbb{S}$ the set of simplices forming $\mathbb{D}$. The cardinality of this set is $n := |\mathbb{S}|$. The set of vertices of a single d-simplex S is denoted by $\mathbb{V}(S) := \{\overline{x}_0^S, \ldots, \overline{x}_d^S\}$. Furthermore $\mathbb{V}(\mathbb{S}) = \{\overline{x}_1^{\mathbb{S}}, \ldots, \overline{x}_m^{\mathbb{S}}\} := \cup_{S \in \mathbb{S}} \mathbb{V}(S)$ is the entire set of vertices of $\mathbb{S}$. As in the previous section our aim is to formulate a mixed integer linear model in which $y = \phi(x)$ for $x \in \mathbb{D}$ holds. According to the univariate case auxiliary binary variables will uniformly be denoted by z.

Now lets get to the point how the incremental method from Sect. 3.5.2 can be generalized to higher dimensional piecewise linear functions. The first major point for this generalization is that any point x^S inside a simplex $S \in \mathbb{S}$, as in the univariate case, can be expressed either as convex combination of its vertices or equivalently as $x^S = \overline{x}_0^S + \sum_{j=1}^d (\overline{x}_j^S - \overline{x}_0^S)\delta_j^S$ with $\sum_{j=1}^d \delta_i^S \leq 1$ and nonnegative $\delta_i^S \geq 0$ for $i = 1, \ldots, d$ (cf. Fig. 3.2).

When we look back to dimension one, we notice that the second main argument of this approach is that an ordering of the simplices holds in which the last vertex of any simplex is equal to the first vertex of the next one. A natural generalization of this approach to dimension $d \geq 2$ is therefore possible if an ordering of simplices with the following properties is available.

(O1) The simplices in $\mathbb{S} = \{S_1, \ldots, S_n\}$ are ordered in such a way that $S_i \cap S_{i+1} \neq \emptyset$ for $i = 1, \ldots, n-1$ holds and

(O2) for each simplex S_i its vertices $\overline{x}_0^{S_i}, \ldots, \overline{x}_d^{S_i}$ can be labeled such that $\overline{x}_d^{S_i} = \overline{x}_0^{S_{i+1}}$ holds for $i = 1, \ldots, n-1$.

An ordering of a two-dimensional example triangulation is illustrated in Fig. 3.3. Note that both properties are trivially fulfilled in the univariate case, where such an ordering is automatically given since the set of simplices $\mathbb{S}$ simply consists of a sequence of line segments. Based on properties (O1) and (O2) the first vertex $\overline{x}_0^{S_i}$ of simplex S_i is obtained by $\overline{x}_0^{S_i} = \overline{x}_0^{S_1} + \sum_{k=1}^{i-1} (\overline{x}_d^{S_k} - \overline{x}_0^{S_k})$.

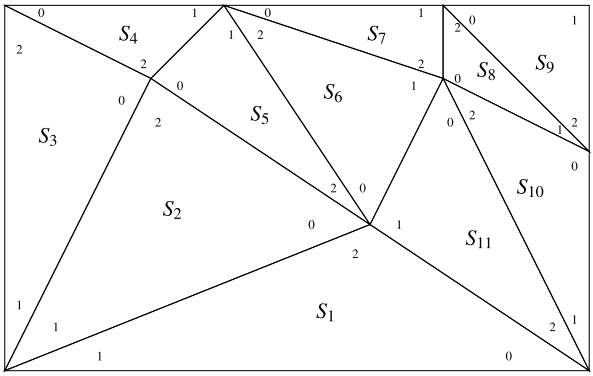


Fig. 3.3 Triangle ordering on a rectangular domain

Bringing this together with the representation of a point inside a single simplex as sum of a vertex and the rays spanning the simplex from that vertex, we get the generalized incremental model

$$x = \overline{x}_0^{S_1} + \sum_{i=1}^{n} \sum_{j=1}^{d} (\overline{x}_j^{S_i} - \overline{x}_0^{S_i}) \delta_j^{S_i}, \tag{3.46}$$

$$y = \overline{y}_0^{S_1} + \sum_{i=1}^{n} \sum_{j=1}^{d} (\overline{y}_j^{S_i} - \overline{y}_0^{S_i}) \delta_j^{S_i} \tag{3.47}$$

$$\sum_{j=1}^{d} \delta_j^{S_i} \leq 1 \quad \text{for } i = 1, \ldots, n, \tag{3.48}$$

$$\delta_j^{S_i} \geq 0 \quad \text{for } i = 1, \ldots, n \text{ and } j = 1, \ldots, d. \tag{3.49}$$

In addition the δ-variables have to satisfy a "generalized filling condition", that is, if for any simplex S_i a variable $\delta_j^{S_i}$ is positive then $\delta_d^{S_{i-1}} = 1$ must hold. Obviously we can conclude that a variable $\delta_j^{S_i}$ can only be positive if for all previous simplices $k = 1, \ldots, i-1$ the variables $\delta_d^{S_k}$ are equal to one. To enforce this condition we introduce auxiliary binary variables z and use the constraints

$$\sum_{j=1}^{d} \delta_j^{S_{i+1}} \leq z_i \quad \text{for } i = 1, \ldots, n-1, \tag{3.50}$$

$$z_i \leq \delta_d^{S_i} \quad \text{for } i = 1, \ldots, n-1. \tag{3.51}$$

Likewise the univariate case, the integrality of the polytope described by inequalities (3.48)–(3.51) together with the nonnegativity constraints $z_i \geq 0$ for $i = 1, \ldots, n-1$, is guaranteed as shown by Wilson [13].

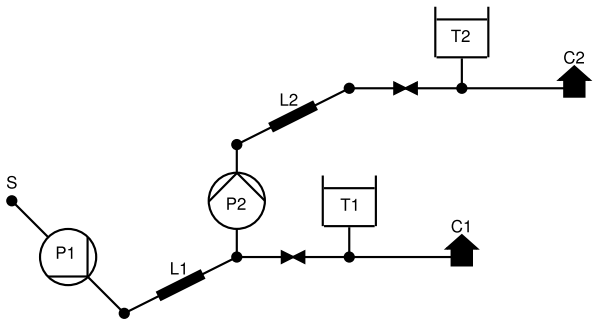


Fig. 3.4 A water network consisting of two pipes, two pumps, two tanks, one supplier and two consumers

3.6 Computational Results

In this section, we present two numerical results from water supply network optimization to illustrate the ability of our approach. Further computational results combined with continuous nonlinear solution techniques from Chap. 2 are shown in Chap. 4. To solve the underlying mixed integer linear problem, we used Gurobi 4.0 [7] as branch and cut solver. Afterward, we verified the computed results of the linearized model by simulation. To this end, we applied a simulation tool from Chap. 2, which solves the nonlinear model equations. All computations are carried out using 4 threads on a computer with two Six-Core AMD Opteron 2435 Processors and 64 GB of main memory.

3.6.1 Network 1

The first instance is based on a tree network consisting of one supplier, two consumers and two pipes (cf. Fig. 3.4). Each consumer has its own preceding tank associated with a valve that is able to cut off the consumer and tank from the remaining network. The pressure level at the source reservoir is fixed to 115 m and the total inflow is restricted by $2\frac{\mathrm{m}^3}{\mathrm{s}}$. Supplied water is transported through a 10 km pipeline overcoming an altitude difference of 50 m using pump P_1. The pump is able to increase the head by at most 150 m. From there water is distributed to tank T_1, consumer C_1 and tank T_2, consumer C_2. To obtain consumer T_2 it might be necessary to use pump P_2, which is identical to pump P_1, to overcome a difference of 50 m in altitude. The demand profile of both consumers is depicted in Fig. 3.5. Each tank has an area of 1000 m^2, a height of 20 m and is forced to breathe at least once. In this scenario breathing means to fall once below a filling level of 25 % as well as exceed a water level of 60 %. The tank levels are initially at 50 %. The described scenario is optimized over a time horizon of four hours using a time step size of 20 min. As objective function supply guarantee maximization is chosen, that is, to minimize the overall sum of tank levels below 60 %.

The resulting mixed integer linear program consists of 9781 constraints and 6805 variables, whereof 2989 are binary. An optimal solution was found after 28 s and

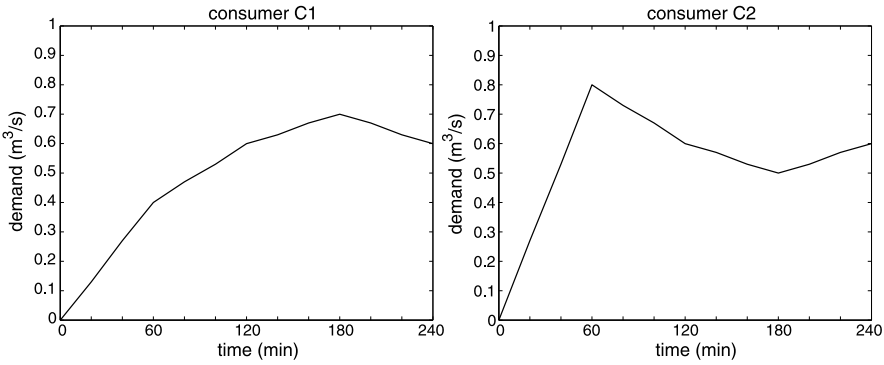


Fig. 3.5 Demand profile of consumers C_1 and C_2

Table 3.1 Control of the optimal solution found by branch and cut for the network given in Fig. 3.4

Time [min]	0	20	40	60	80	100	120
Pump P_1 power [MW]	–	3.7	–	–	–	–	–
Pump P_2 power [MW]	–	–	–	–	–	–	–

Time [min]	140	160	180	200	220	240
Pump P_1 power [MW]	–	3.5	3.5	3.4	3.4	3.6
Pump P_2 power [MW]	–	3.6	3.5	3.7	3.8	3.6

has an overall objective of 80.9955 m. The pump controls of this optimal solution is denoted in Table 3.1 and corresponding tank levels are shown in Fig. 3.6.

3.6.2 Network 2

Our second example on water supply network optimization is taken from a real-life application. The network consists of one source, four sinks, 20 pipes, three pumps and two intermediate tanks (cf. Fig. 3.7). The situation is as follows. Supplied water is pumped uphill and is then either stored in intermediate tanks on top of a hill or it is transmitted directly to the consumers. From the source, water can be taken at a constant pressure level of 130 m. Next to the source, three parallel pumps can operate at fixed speed to increase the pressure. Each pump can increase the pressure by at most 720 m. The power of a pump is restricted to 1 MW to 10 MW. The intermediate tanks are located at an elevation of 495 m and each of them has a cross-sectional area of 1573 m^2 and a height of 14.5 m. Both initial tank levels are set to 63 %. The consumers are located downhill at a height of 400 m, 300 m, 200 m and

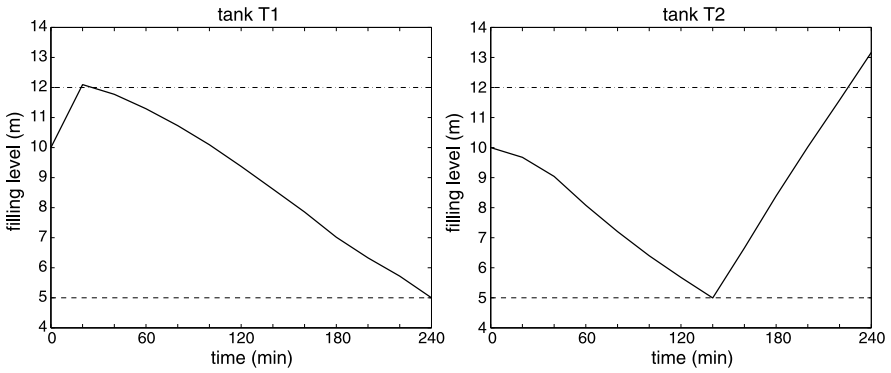


Fig. 3.6 Filling levels of tanks T_1 and T_2

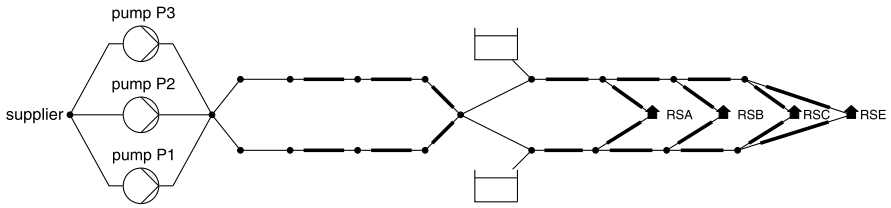


Fig. 3.7 A water network consisting of 20 pipes, three pumps, two tanks, one source and four sinks

100 m, respectively. The consumers' demand for sinks $i \in \{\text{RSA}, \text{RSB}, \text{RSC}, \text{RSD}\}$ is given by

$$d_i(t) = \begin{cases} \dfrac{\Omega_i}{4} t, & 0 \le t \le 4, \\ \Omega_i \left(0.9 + 0.1 \cos\left(\dfrac{2\pi}{20}(t-4)\right)\right), & 4 \le t \le 24, \end{cases} \tag{3.52}$$

with $\Omega_1 = 1$, $\Omega_2 = 0.8$, $\Omega_3 = 1$ and $\Omega_4 = 1.2$. Here Ω_i, and thus d_i, is measured in $\frac{\text{m}^3}{\text{s}}$ and t in hours (cf. Fig. 3.8). The given scenario is optimized over a horizon of one day with a time step size of one hour.

The constructed mixed integer linear program consists of 25000 constraints and 25077 variables, whereof 10839 are binary. An optimal solution was found after 694 s and has an overall power consumption of 145.51 MW. The resulting pump controls and corresponding tank levels of the optimal solution are shown in Table 3.2.

3.7 Conclusion

In this chapter, we have presented a network model which can be used to describe the dynamic transport within a water supply transport network. We have shown

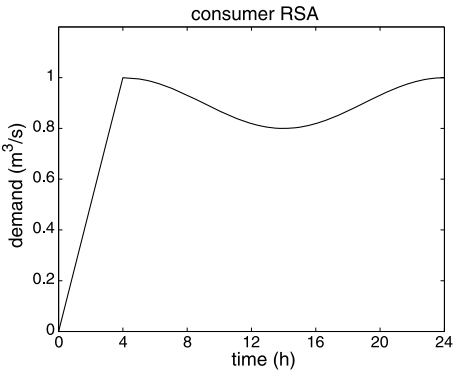


Fig. 3.8 Demand profile for consumer RSA

Table 3.2 Control of the optimal solution found by branch and cut for the network given in Fig. 3.7

Time [h]	0	1	2	3	4	5	6
Pump P_1 power [MW]	–	–	–	–	–	4.0	–
Pump P_2 power [MW]	–	–	–	–	3.6	4.0	–
Pump P_3 power [MW]	–	–	–	–	3.6	4.0	–
Tank T_1 level [m]	8.8	8.1	6.6	4.0	3.4	4.8	4.0
Tank T_2 level [m]	8.8	8.1	6.6	4.0	3.4	4.8	4.0

Time [h]	7	8	9	10	11	12	13
Pump P_1 power [MW]	4.0	4.0	3.3	4.1	3.4	–	4.1
Pump P_2 power [MW]	4.0	4.0	–	4.1	–	–	4.1
Pump P_3 power [MW]	4.0	4.0	–	4.1	–	–	4.1
Tank T_1 level [m]	4.3	6.4	6.9	8.1	8.8	6.7	7.6
Tank T_2 level [m]	4.3	6.4	6.9	8.1	8.8	6.7	7.6

Time [h]	14	15	16	17	18	19	20
Pump P_1 power [MW]	3.4	–	–	–	4.1	3.4	4.1
Pump P_2 power [MW]	–	3.7	3.7	–	4.1	–	4.1
Pump P_3 power [MW]	–	3.7	3.7	–	4.1	–	4.1
Tank T_1 level [m]	8.5	9.2	10.1	9.0	9.8	9.0	11.3
Tank T_2 level [m]	8.5	9.2	10.1	9.0	9.8	9.0	11.3

Time [h]	21	22	23	24
Pump P_1 power [MW]	3.4	4.1	3.4	4.1
Pump P_2 power [MW]	–	4.1	–	4.1
Pump P_3 power [MW]	–	4.1	–	4.1
Tank T_1 level [m]	11.8	12.5	12.6	13.3
Tank T_2 level [m]	11.8	12.5	12.6	13.3

how to use this network model to formulate optimization problems on water supply networks as mixed integer linear programs. Such problems involve nonlinear constraints, which cannot be incorporated into linear programs directly. To this end, we applied a method to approximate such constraints appropriately such that we are able to include them into our model. Finally, we presented two computational results showing that our approach is suitable for global optimization of dynamic water supply network management. A combination of our global mixed integer approach with local nonlinear programming approaches from Chap. 2 are presented in the following chapter.

References

1. J. Abreu, E. Cabrera, J. Izquierdo, J. García-Serra, Flow modeling in pressurized systems revisited. J. Hydraul. Eng. **125**, 1154–1169 (1999)
2. P. Domschke, B. Geißler, O. Kolb, J. Lang, A. Martin, A. Morsi, Combination of nonlinear and linear optimization of transient gas networks. INFORMS J. Comput. **23**, 605–617 (2011)
3. A. Fügenschuh, M. Herty, A. Klar, A. Martin, Combinatorial and continuous models for the optimization of traffic flows on networks. SIAM J. Optim. **16**, 1155–1176 (2006)
4. B. Geißler, A. Martin, A. Morsi, L. Schewe, Using piecewise linear functions for solving MINLPs, in *Mixed Integer Nonlinear Programming*, ed. by J. Lee, S. Leyffer. The IMA Volumes in Mathematics and its Applications, vol. 154 (Springer, Berlin, 2012), pp. 287–314
5. C.E. Gounaris, C.A. Floudas, Tight convex underestimators for $\mathbb{C}^2$-continuous problems: I. Multivariate functions. J. Glob. Optim. **42**, 69–89 (2008)
6. C.E. Gounaris, C.A. Floudas, Tight convex underestimators for $\mathbb{C}^2$-continuous problems: I. Univariate functions. J. Glob. Optim. **42**, 51–67 (2008)
7. Gurobi Optimization, Inc., Houston, Texas, USA. *Gurobi Optimizer Version 4.0*, 2010. Information available at URL http://www.gurobi.com
8. J. Lee, J. Margot, F. Margot, Min-up/min-down polytopes. Discrete Optim. **1**, 77–85 (2004)
9. H.M. Markowitz, A.S. Manne, On the solution of discrete programming problems. Econometrica **25**, 84–110 (1957)
10. M. Padberg, Approximating separable nonlinear functions via mixed zero-one programs. Oper. Res. Lett. **27**(1), 1–5 (2000)
11. J.P. Vielma, A.B. Keha, G.L. Nemhauser, Nonconvex, lower semicontinuous piecewise linear optimization. Discrete Optim. **5**(2), 467–488 (2008)
12. B. Wendroff, On centered difference equations for hyperbolic systems. J. Soc. Ind. Appl. Math. **8**(3), 549–555 (1960)
13. D. Wilson, Polyhedral methods for piecewise-linear functions. Ph.D. thesis in Discrete Mathematics, University of Kentucky, 1998

A. Morsi · B. Geißler · A. Martin
Discrete Optimization (Lehrstuhl für Wirtschaftsmathematik), Friedrich-Alexander-Universität Erlangen-Nürnberg, Cauerstr. 11, 91058 Erlangen, Germany

A. Morsi
e-mail: antonio.morsi@math.uni-erlangen.de

B. Geißler
e-mail: bjoern.geissler@math.uni-erlangen.de

A. Martin (✉)
e-mail: alexander.martin@math.uni-erlangen.de

Chapter 4
Nonlinear and Mixed Integer Linear Programming

Oliver Kolb, Antonio Morsi, Jens Lang, and Alexander Martin

Abstract In this chapter we compare continuous nonlinear optimization with mixed integer optimization of water supply networks by means of a meso scaled network instance. We introduce a heuristic approach, which handles discrete decisions arising in water supply network optimization through penalization using nonlinear programming. We combine the continuous nonlinear and the mixed integer approach introduced in Chap. 3 to incorporate the solution quality. Finally, we show results for a real municipal water supply network.

4.1 Introduction

As mentioned in the last chapter, optimization tasks for water supply networks typically contain not only continuous but also discrete control variables. The purpose of this chapter is to compare and to combine the optimization methods presented in the last two chapters to efficiently tackle such kind of problems. The basic idea behind the combination of both techniques is illustrated in Fig. 4.1.

Discrete optimization offers efficient methods to handle integer variables. Moreover, the applied algorithms deliver globally optimal solutions for the linearized problem. The developed method uses the discrete decisions from the solution found for the mixed integer linear problem as input for the nonlinear (continuous) optimization. In addition, the values of the continuous variables can additionally be used as initial guess here.

The key advantage, regarding the application, of nonlinear optimization techniques is that we can efficiently compute gradient information as described in Chap. 2. Thus, local optimization of the continuous variables is possible in comparatively short calculation times. Since we may consider the full nonlinear problem here, we can then give a "feedback" to the discrete optimization framework concerning the local error in the linearized problem. Further, we have developed a heuristic to find feasible solutions of the nonlinear mixed integer problem which is solely based on the solution of continuous optimization approach.

In Sect. 4.2, the heuristic approach is shortly described. In Sect. 4.3, we present results of discrete and nonlinear optimization techniques for an optimization task based on the network presented in Chap. 1. Finally, we show results for a real municipal water supply network in Sect. 4.4.

A. Martin et al. (eds.), *Mathematical Optimization of Water Networks*,
International Series of Numerical Mathematics 162,
DOI 10.1007/978-3-0348-0436-3_4,

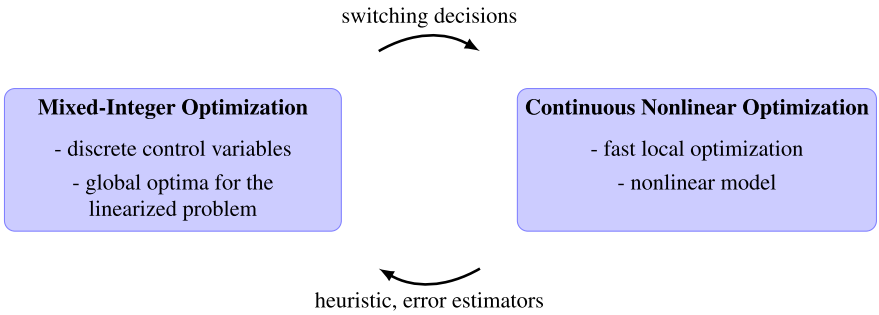


Fig. 4.1 Combination of discrete and nonlinear optimization techniques

4.2 Heuristic Approach

In this section, we describe a heuristic approach to find feasible solutions for mixed integer nonlinear optimal control problems by solving only continuous optimization problems. The presented technique has already been published in [5]. It strongly makes use of the underlying structure, how the discrete and continuous control variables are connected in the original problem: The feasible domain of the related element in the network typically consists of a single value {`minCtrl`}, which represents the switched off status, and an interval [`minCtrlWhenActive`, `maxCtrl`], in which the element may be controlled when it is active.

The first step is to relax this binary (on/off) decision and to consider the interval [`minCtrl`, `maxCtrl`] for the entire optimization horizon. Then, we add a penalty term to the objective function for each relaxed control variable. A prototype of our penalty functions is given by

$$p(x, x_m, y_m) = \begin{cases} y_m\left(2.5\left(\frac{x}{x_m}\right)^3 - 1.5\left(\frac{x}{x_m}\right)^5\right) & \text{if } \texttt{minCtrl} \le x \le x_m, \\ y_m\left(2.5\left(\frac{\texttt{mcwa}-x}{\texttt{mcwa}-x_m}\right)^3 - 1.5\left(\frac{\texttt{mcwa}-x}{\texttt{mcwa}-x_m}\right)^5\right) & \text{if } x_m \le x \le \texttt{mcwa}, \\ 0 & \text{otherwise}, \end{cases} \tag{4.1}$$

where `mcwa` abbreviates `minCtrlWhenActive`, and is plotted in Fig. 4.2. In the course of the optimization process, the position x_m and the height y_m of the peak of the penalty function are varied.

Figure 4.3 shows the basic algorithm from the view of a single penalty term. When the penalty function is initialized, the position x_m of the peak of the penalty function is set to x_0 and the maximum value y_m to y_0. After the run of the optimization tool, we check whether the relaxed control variable x is within one of the predefined fixing regions, that is, $x \le x_{off}$ or $x \ge x_{on}$. In this case, the binary decision is fixed accordingly. Otherwise, we increase the penalty term and move the position of the peak in the direction of the current control. Note that it is possible and also intended that x_m overtakes x. Then, the optimization tool is run again until all switching decisions are fixed or a maximum number of iterations is reached.

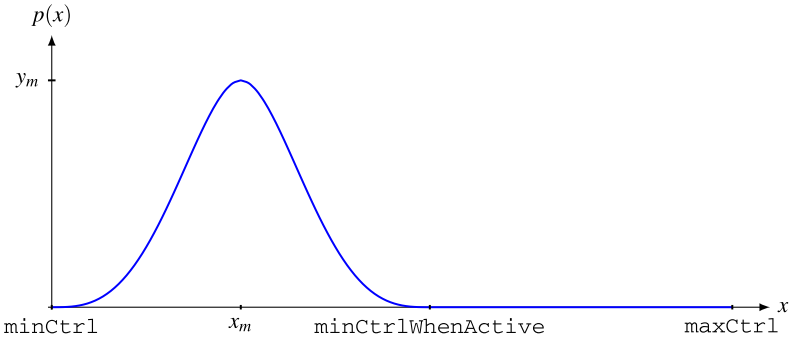


Fig. 4.2 Plot of a penalty function

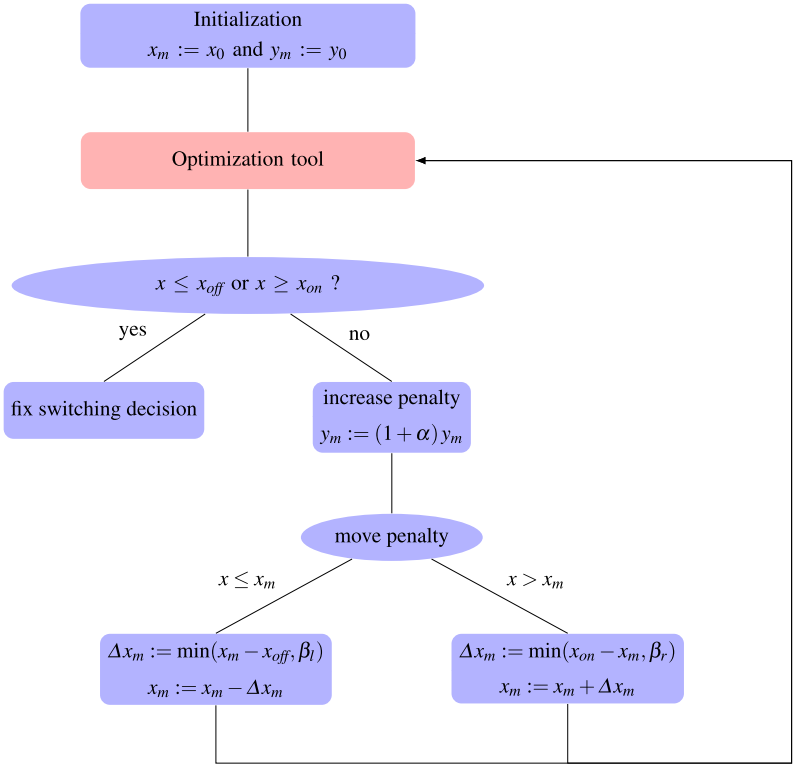


Fig. 4.3 Basic algorithm for the moving penalty function approach

Although our basic algorithm already may yield useful results, there are several challenges depending on the given task. One very important aspect is the choice of the parameters. For example, consider the parameters x_{off} and x_{on}. On the one hand, we would like to have large regions where the binary decisions get fixed as fast as

possible. But on the other hand, fixing the wrong variables too early might lead to an infeasible remaining task. As an improvement of our basic algorithm, we tackle the latter problem by introducing some kind of active set strategy, that is, if we cannot find a feasible solution of the remaining (relaxed) problem, we release some previously fixed binary variables which influence the violated constraints. Moreover, we reduce the size of the previously active fixing region around x_{off} or x_{on} by the factor ρ_{off} or ρ_{on} for every released control variable to avoid cycling between fixing and releasing.

Another strategy to improve our basic algorithm deals with the parameters β_l and β_r for the moving of the position of the penalty peak. Small values for β_l or β_r can result in lots of iterations of the algorithm. On the other hand, large values for the moving can result in jumping around the current solution without the penalty functions being able to affect the control in neither direction. Such situations typically occur if two or more control variables have to be influenced in opposite directions to get a feasible solution of the original (not relaxed) problem.

Our additional strategy is the following: After a specified number of steps into the same direction (x_m is always increased or decreased) we multiply β_r or β_l by a constant factor $\gamma_+ > 1$. Otherwise, if x_m jumps from one side of x to the other and back for a certain number of times, we decrease β_l and β_r by multiplication by a positive factor $\gamma_- < 1$.

4.3 Results for the Meso Network

In this section, we present computational results from the optimization of a medium size network that has already been introduced in Chap. 1. The network consists of two suppliers (cf. Fig. 4.4). One of them supplies water at a constant rate of 45 $\frac{\mathrm{m}^3}{\mathrm{h}}$, the other one is able to deliver water from 100 $\frac{\mathrm{m}^3}{\mathrm{h}}$ to 800 $\frac{\mathrm{m}^3}{\mathrm{h}}$. Supplied water flows through a tank with an area of 400 m^2 and a height of 5 m. Next to the tank two parallel pumps can increase the pressure appropriately, this is, a head difference of approximately 22 m to 40 m. The pumps are able to operate at a flow rate from 150 $\frac{\mathrm{m}^3}{\mathrm{h}}$ to 250 $\frac{\mathrm{m}^3}{\mathrm{h}}$ and 350 $\frac{\mathrm{m}^3}{\mathrm{h}}$ to 550 $\frac{\mathrm{m}^3}{\mathrm{h}}$, respectively. Therefrom water is distributed to five consumers and two identical tanks with a height of 5 m and an area of 500 m^2. A third pump has the ability to transport a flow rate of 40 $\frac{\mathrm{m}^3}{\mathrm{h}}$ to 65 $\frac{\mathrm{m}^3}{\mathrm{h}}$ to a downstream system at a higher level of altitude consisting of one consumer and one tank. This tank has an area of 25 m^2 and a height of 10 m. The demand profile for all consumers is shown in Fig. 4.5. We have to ensure that all tanks, except the first one, must breathe within the 24 hour optimization horizon. That means the filling level must fall below 20 % and it has to exceed a water level of 80 %. Besides, each tank has to reach at least its initial level of 75 % at the final time. The optimization uses a hourly based time step discretization. Our objective is to find a cost minimal (costs of pumps and water delivering costs of the suppliers) control of this scenario. We compute optimal solutions by the two proposed approaches, that is, the mixed integer linear programming approach from Chap. 3 and the heuristic method from Sect. 4.2.

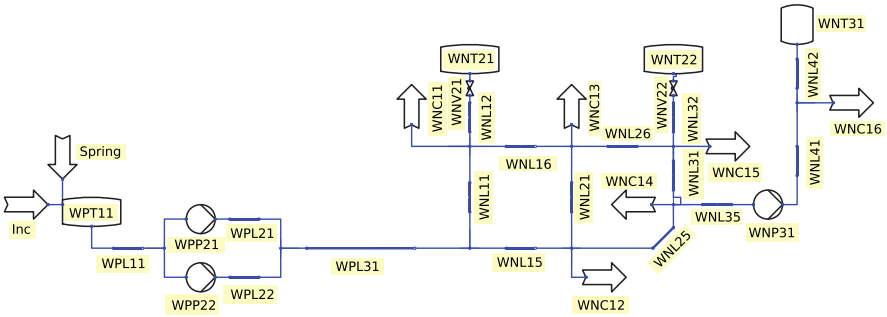


Fig. 4.4 Medium size network

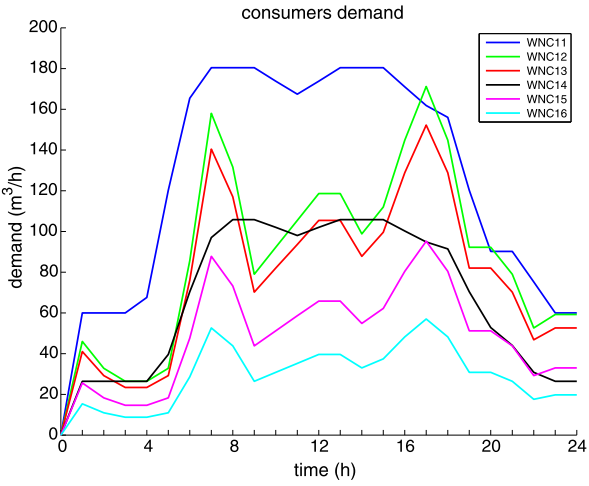


Fig. 4.5 Demand profile of consumers for meso network

The mixed integer linear program results in 25660 constraints and 19310 variables, from which 6401 are binary. An optimal solution was found by the Gurobi Optimizer 4.0 [3] within 41 s. The overall objective value of that solution is 140.9. Note that, we are thus able to ensure that (up to approximation errors) no solution with less objective value exists. Using this control as initial guess, the NLP solver carriers out some adjustments, due to piecewise linearization errors in the MIP, and eventually confirms our control after 8 s. The corresponding pump controls are plotted in Fig. 4.7 and corresponding tank levels are depicted in Fig. 4.6.

The heuristic searches for an optimal solution in the following way. In the first round it allows only three different control points over the complete horizon. This solution is taken as initial guess to the next iteration where six control points are admitted. In the third iteration 12 different control points are allowed and finally all 24 control points are taken into account. As nonlinear optimization solver we use IPOPT [8]. The heuristic obtains a solution with the same objective value of 140.9 within 99 s. The controls of that solution are shown in Fig. 4.9. In Fig. 4.8 the corresponding tank levels are presented.

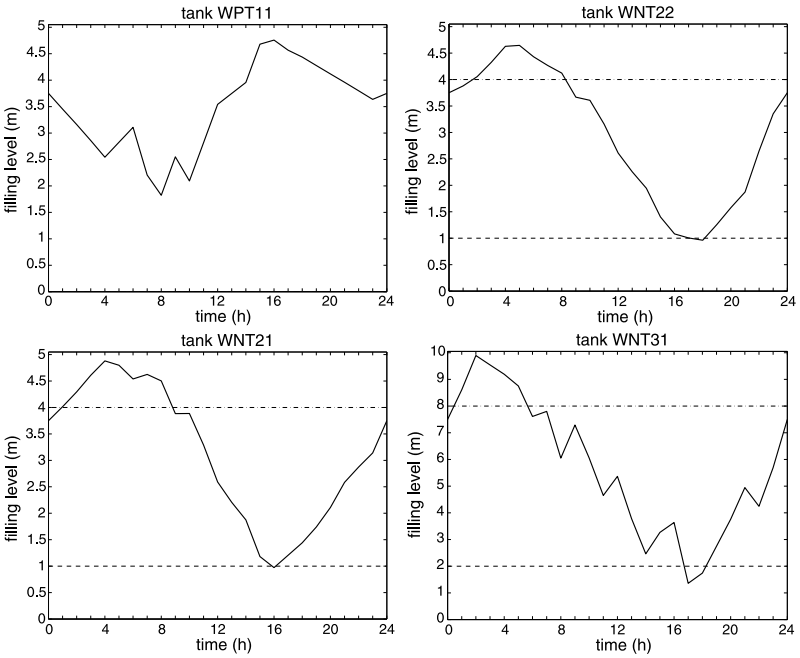


Fig. 4.6 Filling levels of tanks in solution obtained by MIP

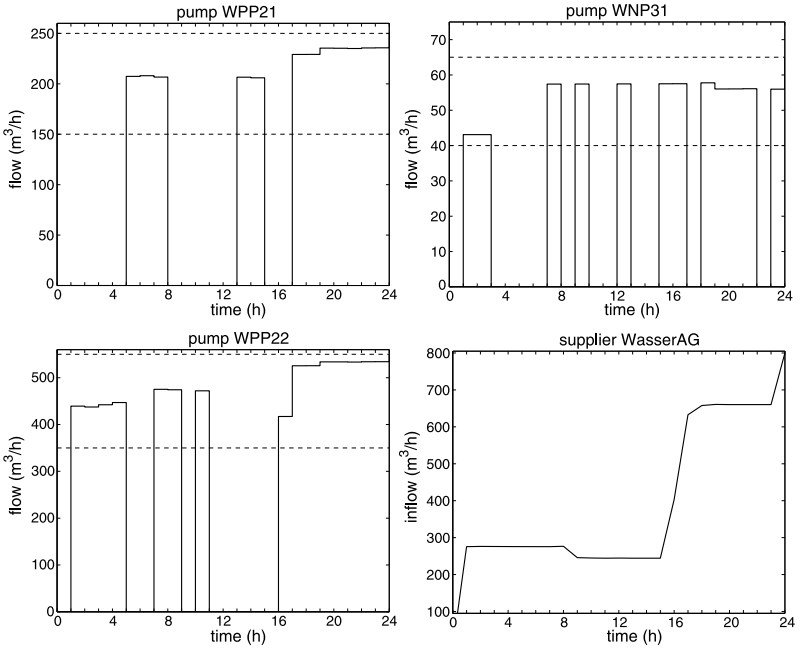


Fig. 4.7 Controls from solution obtained by MIP

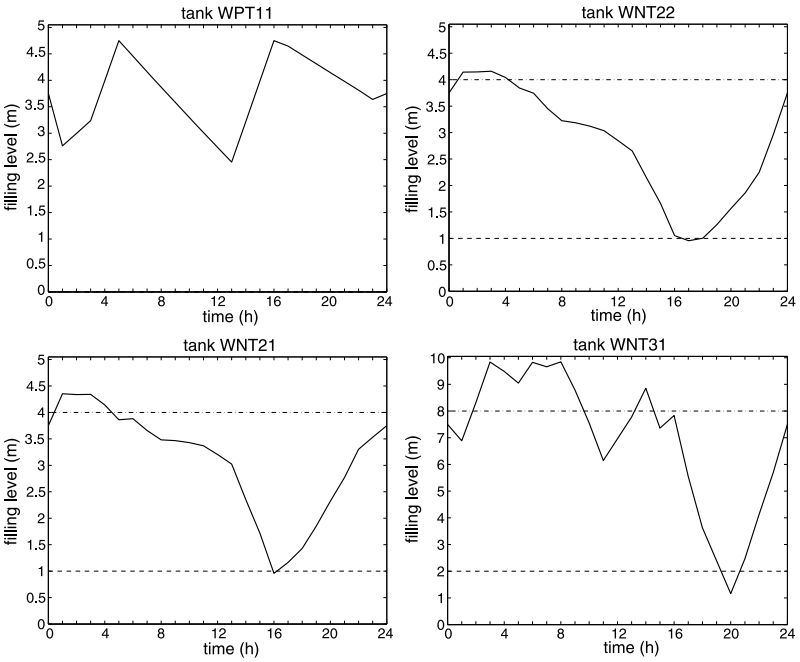


Fig. 4.8 Filling levels of tanks in solution obtained by heuristic

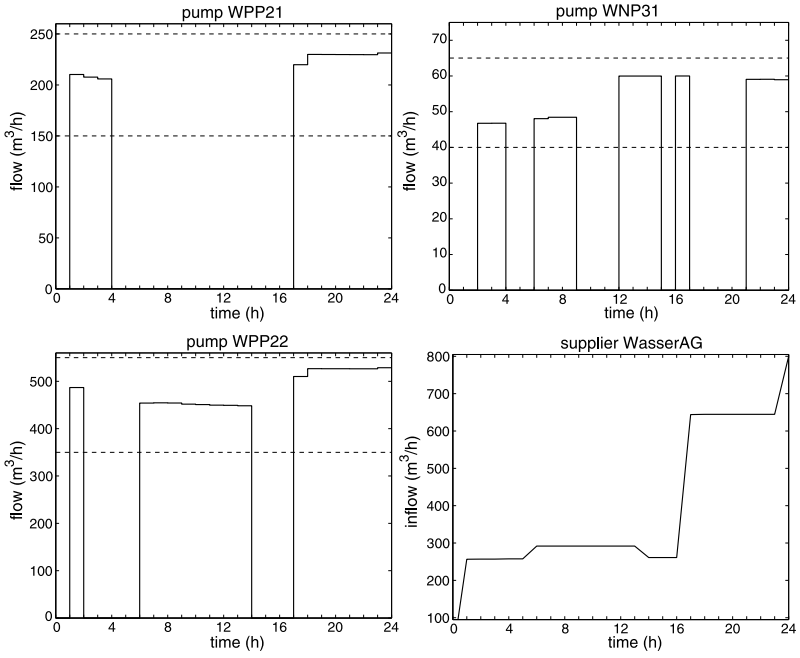


Fig. 4.9 Controls from solution obtained by heuristic

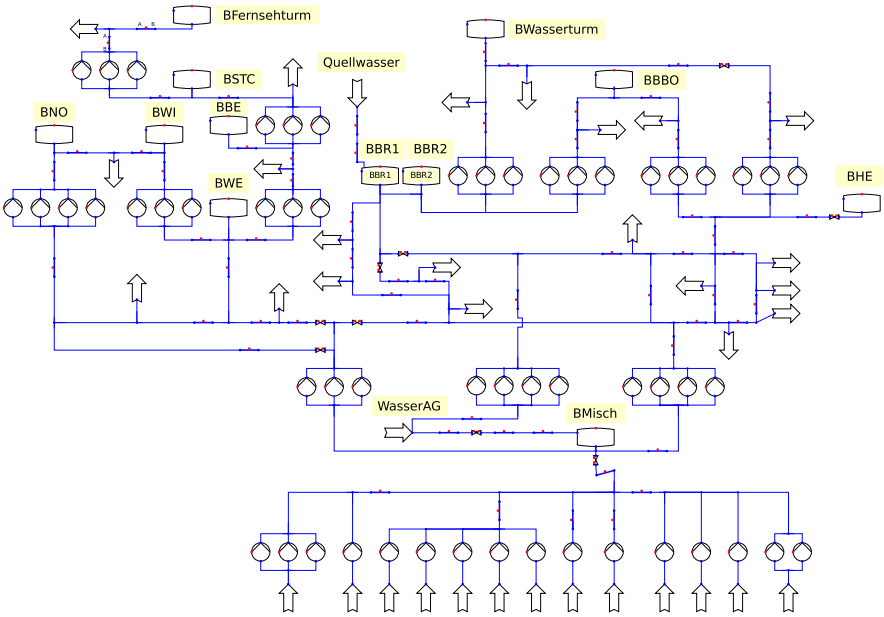


Fig. 4.10 Municipal water supply network provided by Siemens

4.4 Results for a Municipal Water Supply Network

In this section, we consider a (continuous) optimal control task for a real municipal water supply network. The underlying data has been provided by our industry partner Siemens and the presented results are also part of [4]. Figure 4.10 shows the topology of the network. For all consumers, the demand is given for 24 hours. The main difficulty for this network consists of the requirement that all twelve water tanks are supposed to breathe within the optimization horizon of 24 hours and to reach the initial filling level at the end of the optimization horizon. The objective is to minimize the entire costs, which consist of costs for the pumps and the delivery costs for water from the supplier WasserAG.

To reduce the complexity of the considered task, it is useful to cluster the groups of parallel pumps. Each group of several physical pumps is replaced by a single artificial pump, leading to characteristic curves that approximately reflect the behaviour of the whole group. This approach has already been successfully applied in [1] and reduces the complexity of the entire model tremendously.

We search for an optimal control on a control grid with grid points every hour. The temporal and spatial discretization parameters (for the implicit box scheme introduced in Chap. 2) are $\Delta t = 300$ s and $\Delta x \leq 1$ km. To have a good initial guess for the control variables, we first solved the problem on a coarser time grid: Here, we used $\Delta t = 3600$ s and six grid points in the control grid (every 4 hours).

To solve the (discretized) optimal control problem, we apply three gradient-based optimization tools as mentioned above: DONLP2 [6, 7], IPOPT [8] and KNI-

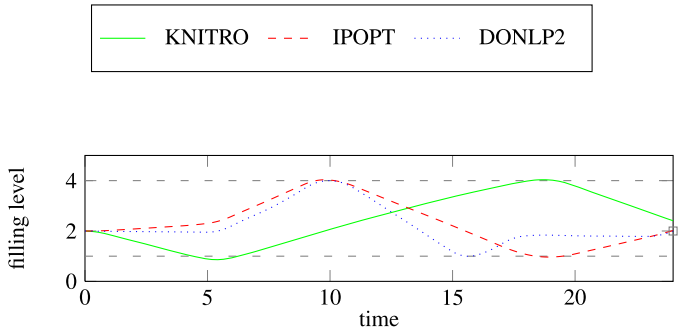


Fig. 4.11 Filling level of tank BBE with breathing and terminal constraint

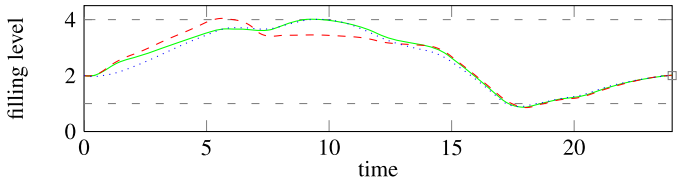


Fig. 4.12 Filling level of tank BBR1 with breathing and terminal constraint

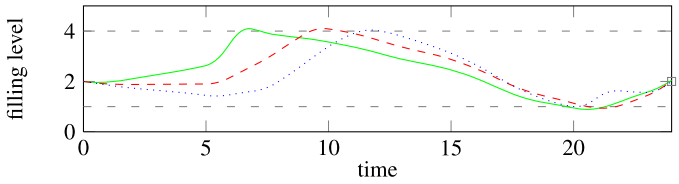


Fig. 4.13 Filling level of tank BNO with breathing and terminal constraint

TRO [2]. Probably due to the size of this problem, only the interior point methods of IPOPT and KNITRO (both with a limited-memory approximation of the Hessian) delivered feasible solutions within acceptable computation times. The best results concerning the computing time and also the objective function were achieved by IPOPT. The entire running time of IPOPT (computations for the initial guess plus the optimization on the fine grid) was 13 minutes. With KNITRO, the computing times and the results varied more strongly depending on the parameter settings. With computation times around 20 minutes, the optimal value of the objective function found by KNITRO is still between 1 % and 2 % above the value of the solution that is steadily found by IPOPT with various settings. Similar to KNITRO, also DONLP2 is quite sensitive towards parameter settings for this problem instance.

The course of the filling levels of four of the twelve water tanks in the network corresponding to the optimal solutions found by KNITRO, IPOPT and DONLP2 is plotted in Figs. 4.11, 4.12, 4.13, 4.14. Here, the horizontal (loosely) dashed lines

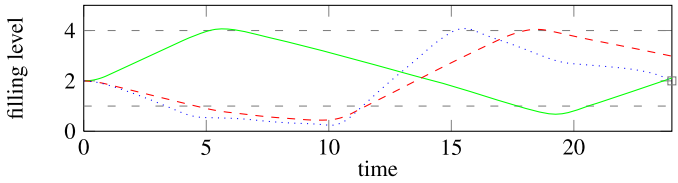


Fig. 4.14 Filling level of tank BSTC with breathing and terminal constraint

indicate the filling levels for the breathing constraints. The small box at $t = 24$ (hours) shows the filling level for the terminal constraint, which equals the initial filling level. The terminal constraint seems to be active for most of the tanks or the upper bound of the breathing constraint is fulfilled during the last time steps, which typically is a good indicator that a locally optimal solution has been found.

4.5 Conclusion

In this chapter, we presented a heuristic approach based on nonlinear programming methods. To incorporate a global solution quality guarantee to water supply network optimization problems, we have applied a combination of mixed-integer linear and nonlinear programming based algorithms. This way, we could benefit from the pros of both methods, namely the potential of mixed-integer programming to achieve globally optimal solutions and the potential of NLP techniques to achieve locally optimal and physically correct solutions. We presented solutions obtained in such a way on a meso scaled network instance and we showed optimization results for a real-life municipal water supply network. From the computational results, we conclude that the performance on both networks is suitable for a day ahead optimization of an operational management.

References

1. J. Burgschweiger, B. Gnädig, M.C. Steinbach, Optimization models for operative planning in drinking water networks. Optim. Eng. **10**(1), 43–73 (2009)
2. R.H. Byrd, J. Nocedal, R.A. Waltz, Knitro: An integrated package for nonlinear optimization, in *Large-Scale Nonlinear Optimization* (2006), pp. 35–59
3. Gurobi Optimization, Inc. *Gurobi Optimizer Version 4.0*. Houston, Texas, USA, 2010. Information available at URL http://www.gurobi.com
4. O. Kolb, Simulation and Optimization of Gas and Water Supply Networks, PhD thesis, Technische Universität Darmstadt, 2011
5. O. Kolb, P. Domschke, J. Lang, Moving penalty functions for optimal control with PDEs on networks, in *Progress in Industrial Mathematics at ECMI 2008* (Springer, Berlin, 2010), pp. 925–931
6. P. Spellucci, A new technique for inconsistent QP problems in the SQP method. Math. Methods Oper. Res. **47**(3), 355–400 (1998)

7. P. Spellucci, An SQP method for general nonlinear programs using only equality constrained subproblems. Math. Program. **82**(3), 413–448 (1998)
8. A. Wächter, L.T. Biegler, On the implementation of a primal-dual interior point filter line search algorithm for large-scale nonlinear programming. Math. Program. **106**(1), 25–57 (2006)

O. Kolb · J. Lang
Numerical Analysis and Scientific Computing, Technische Universität Darmstadt, Dolivostr. 15, 64293 Darmstadt, Germany

O. Kolb
e-mail: kolb@mathematik.tu-darmstadt.de

J. Lang
e-mail: lang@mathematik.tu-darmstadt.de

A. Morsi · A. Martin
Discrete Optimization (Lehrstuhl für Wirtschaftsmathematik), Friedrich-Alexander-Universität Erlangen-Nürnberg, Cauerstr. 11, 91058 Erlangen, Germany

A. Morsi
e-mail: antonio.morsi@math.uni-erlangen.de

A. Martin (✉)
e-mail: alexander.martin@math.uni-erlangen.de

Chapter 5
Optimal Control of Sewer Networks Problem Description

Steffen Heusch, Holger Hanss, Manfred Ostrowski, Roland Rosen, and Annelie Sohr

Abstract This chapter gives an overview of optimal control of sewer networks with dynamic process models. After introducing the method of model predictive control (MPC) and its requirements for optimization and process modeling a focus is set on practical applications and the industrial viewpoint. An up-to-date sewer management system is introduced and used to illustrate industrial requirements and the mathematical challenges involved in it.

5.1 Introduction

Urbanization, environmental protection and sustainable development are some of the major challenges in the 21th century. Nearly all industrial companies develop and deliver technology, products and solutions with respect to these megatrends. On top-level web pages clear statements and slogans can be found, e.g. “Green IT—Energy Efficient Solutions” (IBM), “Sustainability is integral to all aspects of our business” (ABB) and “A pioneer in intelligent infrastructure solutions” (Siemens). These are not only empty phrases as the example Siemens shows, which demand “Sustainability as the guiding principle of all their actions” [12]. This company wide strategy leads to a large and continuously growing environmental portfolio and investments in research and development.

Great importance of these research activities is attached to the management of water resources. Due to increasing legal requirements (e.g. to minimize negative impacts of urban wastewater systems on receiving water bodies), the field of water resources management is enjoying broad technological progress. Dynamic modeling of sewer networks has become a standard application in daily work by a broad community. Equipment for online-monitoring of runoff and water quality is substantially improving at the same time. Those developments are supported by advancements of computational power, such as disc storage size, processor speed and parallelization methods which allow storage of mass data as well as execution of extensive process models in application-oriented runtime. On the other hand, financial restrictions require intelligent solutions. In order to face these challenges, traditional boundaries between different disciplines are resolved, e.g. optimization methods and automa-

A. Martin et al. (eds.), *Mathematical Optimization of Water Networks*, International Series of Numerical Mathematics 162,
DOI 10.1007/978-3-0348-0436-3_5,

tion concepts play an increasing role in water resources management. Overall this leads to an increased consideration of Real Time Control (RTC) in urban drainage systems.

Not all networks are control worthy though. Control potential is generally available in heterogeneous sewer systems with low slopes and high in-line storage volumes. In the modeling process, flow dynamics has to be considered leading to the demand for hydrodynamic models based on shallow water equations in order to adequately simulate backwater effects.

Two control strategies are generally distinguished in RTC: Offline control and online control. Both strategies use a control time step in which control settings of the actuators in the sewer system are determined consecutively. In offline systems control decisions are derived in form of logical control rules during the design phase. In order to find these control rules for a huge variety of potential flow conditions many simulation runs are necessary, making this approach labor-intensive. Control decisions are saved in a database (e.g. in the form of if-then-rules) and are accessed during operation without significant time delay. In online systems control decisions are calculated online using simulation models to predict the system behavior and react online with an optimal control decision. Since prediction of future state developments of the system is an essential part of this approach, this method is known as *model predictive control (MPC)* or *optimal control* of sewer systems.

Dynamic models are generally regarded as infeasible because they are computationally highly demanding and therefore impractical for MPC. However, this opinion is usually based on the performance of flow and pollution routing models, which are approved in practical applications (engineers prerequisite), but which do not necessarily use latest mathematical expertise (mathematicians assumption). The work in this project complies with the different views on the topic and follows consequently two approaches: The mathematicians approach is based on innovative numerical solutions for flow routing calculations and derivative based optimization algorithms. The engineers approach is based on the application of approved flow routing models and on a flexible optimization module which enables the usage of both local and global search algorithms.

In this book, the topic *Optimal Control of Sewer Networks* is divided into four sections. This section gives a general introduction and the problems and constraints connected to the application of MPC of sewer networks with dynamic models. The following sections present the approaches and results of the working group which reflect the different views on the topic. Chapter 7 follows the engineers view, which is based on the prerequisite to use existing simulation models for flow routing and the conviction that global search algorithms are required for optimal control of extensive networks. A simple case study is introduced. Chapter 8 follows the mathematical view on the topic, which is based on the development of a new process model, which enables optimal control of dynamic systems. In Chapter 9 the achievements of the two approaches are compared by means of a more complex case study and a conclusion is given.

5.2 Technical Principles

5.2.1 Dynamic Flow Routing Modeling

Dynamic flow computations in sewer networks are based on the shallow water equations. In urban drainage management the usage of simulation software for dynamic flow routing is a common task for which many simulation models exist. Popular simulation models are SWMM (US EPA), InfoWorks (MWH Soft), MIKE Urban (DHI), SOBEK (Delft Hydraulics) and Hystem-Extran (itwh), but there are many other models used by operators, consultants and in administration. These models are usually commercial products developed and distributed by private companies, for which the source code is not available, varying in the usability of the software (GUI, tools for the preparation of reports and maps, GIS integration, import and export functions, etc.). SWMM is an exception since it is a free software for which the source is available. Most of these models apply finite difference methods (FDM) for the numerical solution of the shallow water equations. Software with other solution methods exist (e.g. DYNA, which uses FVM-methods as well as parallelization features [13]) but no detailed informations are published on the solution methods as well. Considerable differences in calculation time between those simulation models applying the FDM methods are not known.

5.2.2 Model Predictive Control

In MPC systems, control decisions are calculated online during operation. During the design phase only the control objective has to be formulated. Figure 5.1 shows the general workflow of MPC systems. Compared to offline control, in which the control decision is developed during the design phase on the base of a finite number of simulations, MPC has the advantage that it handles any given flow situation,

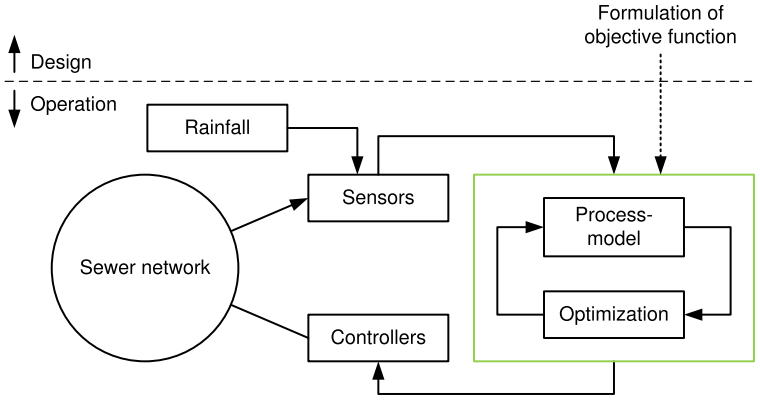


Fig. 5.1 Control algorithm for MPC applications

whereas offline systems are designed to handle a large but nevertheless incomplete set of flow situations. Thus, only online systems are theoretically capable to find optimal solutions for every situation.

For the derivation of the control decision in MPC systems a software is required which contains an optimization module and a dynamic process model to predict the flow in the sewer network. Generally, MPC systems are characterized by three principles:

1. Implementation of the receding horizon.
2. Explicit use of a process model to predict future state developments of the system.
3. Application of optimization algorithms to calculate optimal control settings.

Receding Horizon MPC systems implement the concept of receding horizon, in which the control action is obtained by solving a finite horizon optimal control problem consecutively at each sampling instant (i.e. after every control step). The optimization yields an optimal sequence of future settings for all controlled actuators but only the first control in this sequence is applied to the system. The time horizon moves on by the length of the control step and after updating the current states of the system the optimization starts over again.

Process Model The process model is required to predict future state developments of the system. Due to the formulated prerequisite of this research it consists of a dynamic model for flow routing in sewer networks. The process model represents both, the objective function and the constraints of the optimization problem.

For receding horizon applications two special requirements have to be fulfilled. First, the application must be able to save system states, i.e. flow conditions expressed as water levels and flow rates, and supply them as initial boundary conditions for following simulation runs. Second, it must be possible to apply time dependent flow control, using control structures like pump rates, weir heights or outlet openings.

Optimization Several optimization algorithms are available as stand-alone applications, which are interfaced with the process model to compute an optimal control. The common approach of all of these optimization algorithms is to improve the quality of one or more given controls with a sophisticated search strategy.

Optimization algorithms are often classified by their search strategy: Derivative-free optimization algorithms use the process model as a black-box and combine heuristic search strategies with sophisticated sampling routines to find improvements of a given control.

In the contrary, derivative-based algorithms rely heavily on sensitivity information, which has to be provided by the process model. These algorithms are in general more efficient than derivative-free strategies in finding improvements, but require local sensitivity analysis of the process model.

[11] attempted to classify optimization methods for RTC (Fig. 5.2) although they stress that it is almost impossible to come up with a strict taxonomy, since there

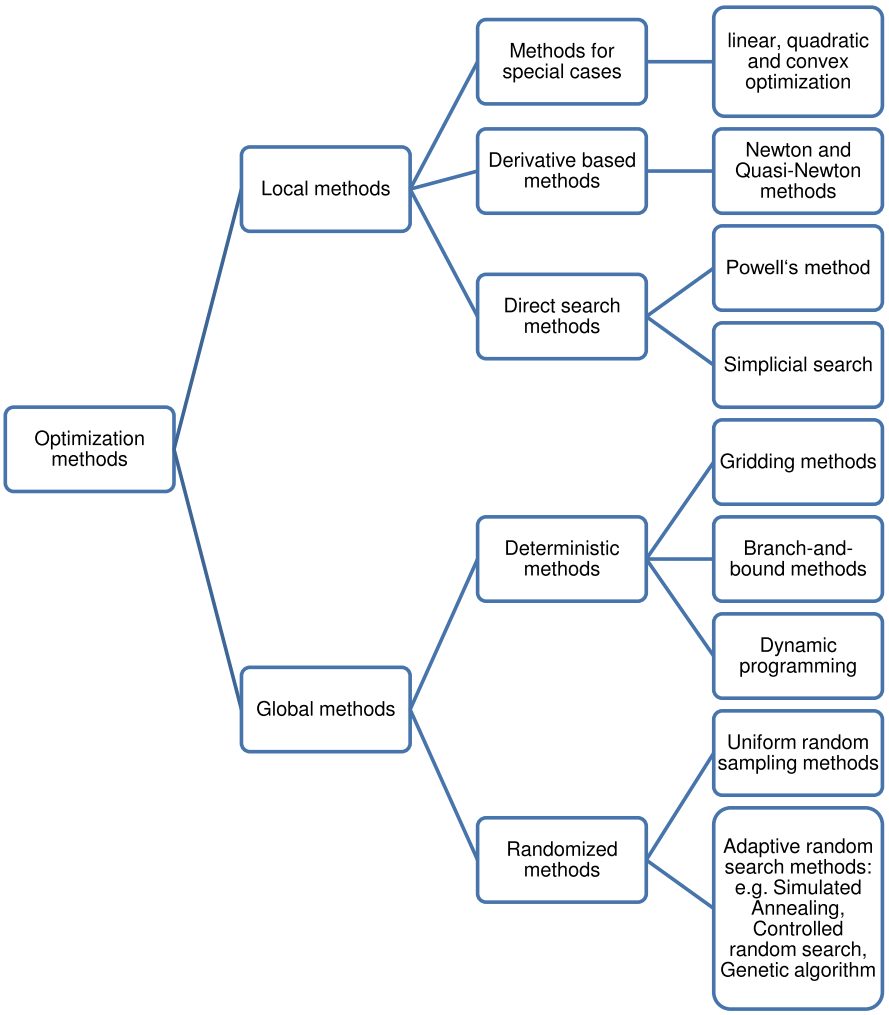


Fig. 5.2 Classification of optimization methods ([11], modified)

are many interrelations between the approaches and that a clear distinction between local and global methods cannot be made.

Setup of MPC Software Process model, optimization module and the operation of the moving horizon loop are managed by a software framework. Within this framework several time horizons have to be considered (Fig. 5.3):

Evaluation horizon: The evaluation horizon is the time span required to evaluate the objective function and therefore equals the simulation period of a single simulation run of the process model.

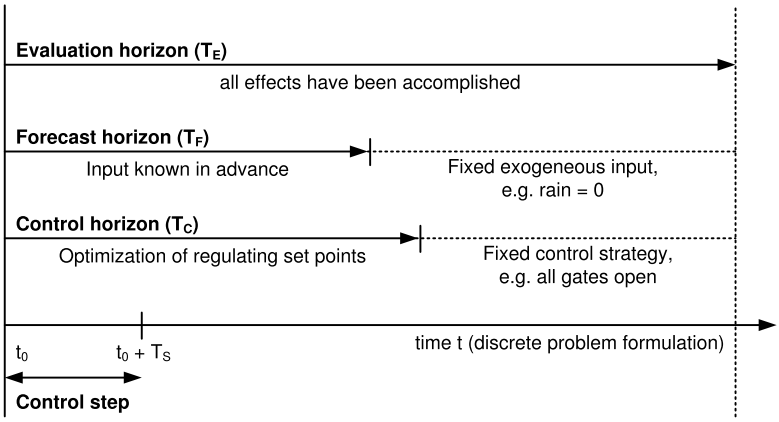


Fig. 5.3 Time horizons required for MPC ([9], modified)

Forecast horizon: The forecast horizon is the time span for which inflow boundary values can be computed. Inflow forecast is generally affected with uncertainties, whereas the degree of uncertainty depends usually on the forecast computation method. Rainfall-runoff models are a standard approach to compute inflows from the forecasted rain and can be improved by the evaluation of radar information. Since the forecast horizon might be shorter than the prediction horizon, the inflow boundary values for the remaining time span are estimated.

Control horizon: The control horizon is the time span for which control decisions for the actuators have to be computed with the optimization algorithm. If the control horizon is shorter than the prediction horizon, the control decisions in the remaining time span have to be extrapolated.

Control step: The control step is the sampling time during which all control settings have to be found by the optimization algorithm. After finding the optimal settings and delivering this information to the sewer system, the control horizon moves on by the length of the control step and the optimization process is initialized again with the latest sensor information. It is therefore essentially the time step of the control loop.

As optimization algorithms make excessive use of the process model, it is necessary to provide an interface which allows the optimizer to feed a flow control into the process model and to extract the state answer. In regard to the interface between the process model and the optimization module two different approaches are generally possible: They are either merged together presenting one optimization problem which can be solved explicitly or they are strictly separated demanding communication between them (e.g. with text files). The former approach is the traditional form of existing MPC software which usually utilizes derivative-based optimization algorithms requiring the calculation of numerical derivatives of the differential equations describing the flow process. Until now, these tools utilize simplified process models, mostly linear transfer functions, since the optimal control of hyperbolic differential

equations is still a research topic for which solutions were not yet time-efficiently available. By using these simplified process models the application of different optimization methods is possible. With such systems optimal control settings can be identified quickly, but the ability to model backwater effects has to be questioned.

5.3 Applications of MPC

Approaches for the application of simulation models and optimization methods for the operation of sewer networks were published in the US during the seventies of the 20th century. A first analysis on requirements for the optimal control of sewer network was given by [5]. Operators looked at RTC systems sceptically from the beginning, calling the method "cannot-engineering" [3] or simply refusing to believe that automatic control might be superior to manual operation in certain situations [10]. Even with advancements in monitoring and computing technologies over the last decades there is still a certain degree of prejudice in regard to the application of RTC in general but especially of MPC systems [4].

Implementation of RTC methods is definitely a complicated task in planning and operation but despite all skepticism—[1] for example state that RTC applications are rare because of the reliability of the required SCADA infrastructure as well as inadequate software and computing capacities—there are successful examples which prove that global control of sewer networks is feasible if the system has control potential. [15] list several cases in which investment costs of several millions of Euro were saved. Compilations of past and ongoing RTC projects are also given in [11] and [2].

Almost all of the listed RTC projects apply offline control methods though, i.e. hardly any MPC systems are in operation. The MPC system which is probably best documented within the scientific community is the global control system of the Quebec Urban Community (QUC) in Canada which is in operation since 1999 [8] and [7]. In order to reduce overflows from the combined sewer network into the St. Lawrence river, a control scheme with five control devices and a total controllable storage volume of 15.000 m^3 was implemented. For optimization a nonlinear programming algorithm is applied to a simplified process model based on linear transfer functions. This model is calibrated on-line using measured and computed flows (from a simultaneously running non-linear hydraulic model) using a Kalman filter. Compared to the static control, i.e. the situation prior to the installation of the global control system, reductions of overflows are estimated to be about 60 %.

Development of MPC software is usually connected to specific projects and companies. In Quebec a software is applied, which is developed by BPR-CSO, the company that is in charge for the whole planning of the MPC system. This software can be applied to other sewer systems but the peripheral requirements require the knowledge of BPR-CSO in order to use it. Another example of software which can be applied for MPC, is SIWA Sewer Management System by Siemens [6]. It is integrated in other existing software components from Siemens. Again, application

of the software demands expert knowledge from Siemens. More information of the software is given in the following section.

5.4 An Industrial Viewpoint

Simulation and optimization methods are widely used in industrial solutions for engineering and operation of plants and infrastructures. Typical applications in the engineering process are the validation of design concepts, the calculation of physical effects, and the optimization of design parameters. In the operation phase simulations run alongside the system operation to support the operator. Optimization methods are used on different levels, from management level (Enterprise Resource Planning ERP) to production control (Manufacturing Execution System MES) to process level (automation and control). These methods are realized in many industrial products, for example in Siemens product families SIROLL for metals and SIPAPER for the paper industry. On production planning level SIROLL "Speed Optimization System" coordinates all individual aggregates of a pickling line and tandem cold mill. Each part of the plant should run as fast as possible to increase throughput, but only as fast as necessary for quality reasons. With shorter time horizons applications of MPC methods are implemented in further SIROLL products, e.g. SIROLL HM and SIROLL Furnace Optimization, and SIPAPER products. So SIPAPER APC (Advanced process control) calculates set points for cost-optimal dosage of bleaching chemicals for single- and multi-level bleaching stages.

For water supply and wastewater disposal the modular Siemens SIWA Pipeline and Network Management Systems provide integrated solutions. The intelligent linkage of automation, IT infrastructure, measurement and control technology with management aspects creates the foundation for the centralized monitoring and optimized control of all operational processes. Exemplified by SIWA Sewer Management System practical and industrial requirements will be explained and, as a result, mathematical challenges will be formulated.

5.4.1 SIWA Sewer Management System

SIWA Sewer Management System is an innovative tool for sewer network control that uses optimization methods to calculate the best control interventions in sewer networks. It calculates best utilization of the storage volume and reduces the discharge of wastewater into natural bodies of water. Also the distribution of wastewater flow to the water treatment plant can be uniformed, which improves the water treatment performance.

Compared to typical MPC applications for closed loop control the evaluation horizon takes several hours due to the large dimension of sewers and the comparatively low flow velocity of wastewater. Furthermore by reason of the underlying

shallow water equations (hyperbolic partial differential equations) SIWA Sewer uses a nonlinear system model for the prediction. In order to allow a reusable product solution a component-oriented system architecture was chosen. Each plant component, e.g. sewers, storm water basins, throttle valves, pumps and weirs, delivers its process model in the form of non-linear equations to a central system of equations, which represents the equality constraints of the nonlinear optimization problem. From a physical-mathematical point of view, the translation of the hydrodynamic shallow-water equations to non-linear algebraic equations is important and a key factor for the power of the application as an optimal control solution. In SIWA Sewer Management System the finite-difference method is applied. Depending on the specific network design and especially on component type, in particular for sewers with and without storage capacity and overflow, the discretization in space and time is adapted to reach sufficient accuracy and performance. In the same component-oriented way the terms for the objective function, e.g. wastewater discharge, pump costs and uniform wastewater inflow to treatment plants, and the inequality constraints of optimization variables, e.g. non-negative flows, are gathered. This leads to the formulation of a large non-linear optimization problem, which will be solved with a SQP-solver which is customized for better performance and to avoid the calculation of local optima. The calculation results are set points for lower-level automation, for example for the control of throttle valves, weirs and pumps.

SIWA Sewer Management System is an online application. Therefore the look and feel of the human machine interface (HMI) is important for the acceptance among operators. Accordingly, a typical HMI solution for a control station application, here WinCC (SIMATIC PCS 7), is used (see Fig. 5.4).

The application shows the structure of the sewer network (or a part of it) and current operating states like filling levels and in- and outflows to storm water tanks (SWT). The diagram shows the calculated future filling level and discharge volume of the last basin before the sewage treatment plant. In this example, the incoming wastewater quantities are so high that despite optimization the filling level will reach 100 % and discharge will occur. Besides the visualization of optimization results and forecast of operating states, the application allows the input and modification of component parameters and optimization options.

5.4.2 Industrial Requirements and Mathematical Challenges

The success of a sewer control application depends not only on the power of mathematical optimization methods and the customer-oriented user interface, but also on the integration in the existing supervisory control and data acquisition (SCADA) system (see Fig. 5.5). Current measurement values and device states must be transmitted from SCADA to Sewer Management System. The values must be permanently validated and checked. Then the values can be used for an update of the internal states of the process part in the optimization model. After the optimization run, the results (the new set points) should be automatically transferred back to the

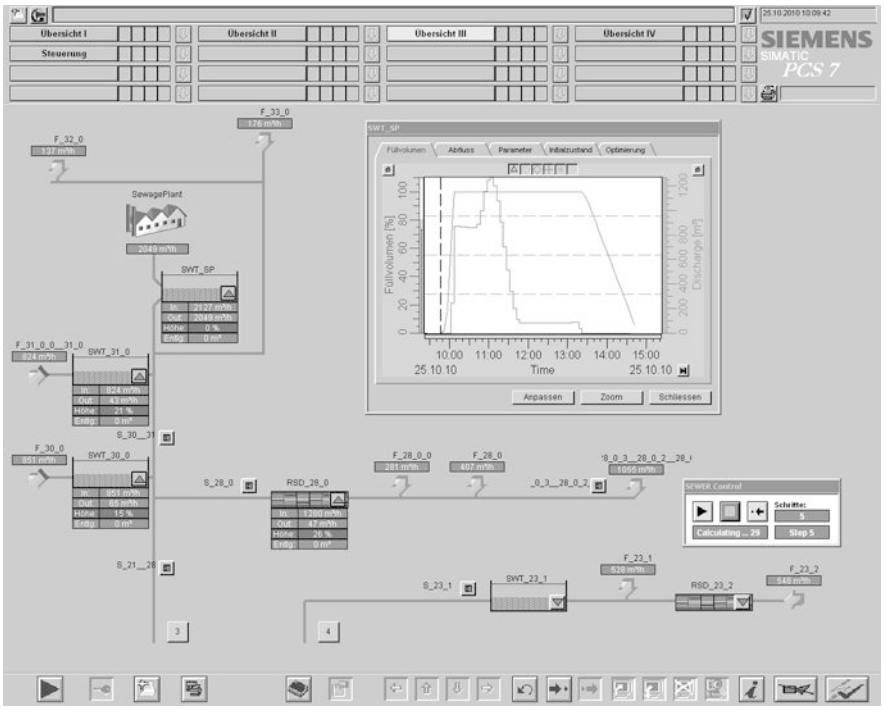


Fig. 5.4 Graphical user interface of SIWA sewer management system

SCADA system. A big difference for the operator but not from a technical or automation software aspect is how the result data will be used. This ranges from an operator recommendation to a full integration with automatic transfer of new set points to basic automation, e.g. to closed loop control of valves.

In all variants the breakdown and malfunction of involved components must be considered: False or outdated measurements and transmission failures from/to SCADA and the Sewer Management System. For this reason not only the optimization results for the next time interval in the receding horizon procedure, but also of several time intervals will be transferred and stored. In case of a failure replacement and default values are available.

This points to—independent from the realization in SIWA Sewer Management System—major challenges for future developments and research demand derived from it (see next section). Theoretically, the hyperbolic differential equations can be discretized in time and space with sufficient numerical accuracy. But this leads to a huge number of equations in the optimization problem, which cannot be solved with existing mathematical methods within the time interval which is fixed by the control step. Beside the general non-linear process behavior some discrete events occur in sewer systems. Decisions on operation and scheduling of (fixed speed) wastewater pumps must be determined. And from an engineering and industrial product point of view a reusable concept must be ensured. On the one hand, this requires a modu-

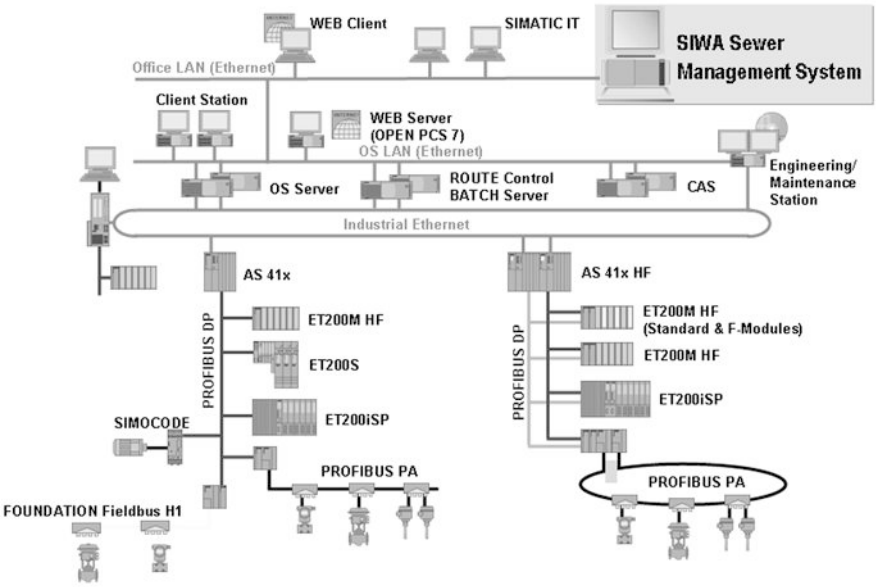


Fig. 5.5 Integration of a sewer control system in existing SCADA architecture

lar software structure (user interface, data base, algorithms and solver), on the other hand it implies a plant component-oriented library to realize per type/instance concept solutions for different sewer networks.

This leads to the following mathematical challenges:

- Performance of the mathematical optimizations. This relates to the required level of detail of the process model, the optimization algorithms and the use of multi-core processor architectures.
- Robust algorithms in dual sense. First, the algorithms must be reliable and stable. Second, the optimization must be robust against changes of variable values because many values are based on uncertain prognosis of rainfall and wastewater inflow to the sewer system.
- Consideration of discrete events in the process and in the optimization, e.g. calculation of pump schedules.
- Usage of component-oriented modeling approaches and reusable concepts for different sewer networks.

5.5 Practical Relevance and Research Demand

MPC is an established procedure in many technical disciplines. In urban drainage management it is well accepted for the control of waste water treatment plants, i.e. practical experiences and software infrastructure for these application already exist

and the industry aims to apply these knowledge to the sewer network in order to gain synergy effects in operation. Moreover, with current developments of computation power and enhancements in online monitoring of flows and pollutants due to legal requirements, it is conceivable that technical possibilities will favor MPC applications in near future.

MPC applications of sewer networks are so far not popular since the control constraints place restrictions on the degree of sophistication of the optimization algorithm, which in turn restricts the level of the mathematical process model used [14]. In other technical disciplines MPC is based on linear process models making optimization easy. But flow processes in sewer network are highly nonlinear and the proper mathematical description of these processes is based on hyperbolic differential equations requiring the application of numerical methods for their solution. These methods are time consuming and, furthermore, methods for the optimal control of hyperbolic systems are not available yet for derivative based optimization, since sensitivity analysis requires process models that generate smooth and robust state information.

In regard to the control algorithms, the bottleneck for MPC calculations are the time restrictions from the control step, since the optimal control decision for the following time interval has to be found within this available time period. Usually a control step of five minutes is applied. This value results from hydraulic as well as from practical considerations. From the hydraulic point of view a longer time step would minimize the possibilities to influence flows in respect to the defined objective function. For example, if the minimization of overflow volumes is penalized, control of storage volumes is a major factor. This task is getting more difficult with increasing length of the control horizon. From a practical point of view a shorter time step would utilize the technical control devices such as pumps and sluice gates too excessively.

As for the optimization algorithms, a number of authors state, that global search algorithms are required in order to assure that an optimal control setting is found. While this is true from a theoretical point of view, no proof has been given until now, that global algorithms are superior than local search algorithms in practical application.

In summary, within this research the following questions are under investigation:

- Development of methods for MPC with hydrodynamic process models to prove that such computations are generally possible.
- Development of a flow routing model based on shallow water equations which generates sufficiently smooth states for derivative-dependent optimization.
- Comparative study of local and global optimization algorithms for MPC.

The tools, which are developed during this work, are so far only for the application in simulation studies. Connections to real SCADA systems and networks have not been implemented. To be able to compare the results of the case studies, inflows to the sewer network where calculated before by means of rainfall runoff simulations.

Two case studies were investigated to prove the functionality of the developed software tools. The first system is an academical example which is used to prove

general software performance and the overall concept of the moving horizon system. The network is described in Sect. 7.3. The second system is a presentation of a real sewer network. An overview of the system and the constraints for the MPC calculations are given in Sect. 9.1.1. This system is more complex than the academical network. It is used to produce answers for the formulated questions and challenges. In both systems routing of flow and of a representative pollution parameter (COD—Chemical Oxygen Demand) is considered but for the MPC calculations only flow is regarded as optimization objectives. Pollutants were considered in multiobjective calculations in Chapter 10.

References

1. S. Darsono, J.W. Labadie, Neural-optimal control algorithm for real-time regulation of in-line storage in combined sewer systems. Environ. Model. Softw. **22**(9), 1349–1361 (2007)
2. DWA, *Advisory Leaflet DWA-M 180E: Framework for Planning of Real Time Control of Sewer Networks* (German Association for Water, Wastewater and Waste, Germany, 2005)
3. E. Englmann, J. Lohmann, W. Schilling, S. Schlegel, Instrumentation and control of water and wastewater treatment and transport systems. Korresp. Abwasser **33**(7), 559–562 (1986)
4. Helmut Gruening, Abflusssteuerung – quo vadis? KA, Wasserwirtsch. Abwasser Abfall **55**(4), 5 (2008)
5. J.W. Labadie, N.S. Grigg, P.D. Trotta, Minimization of combined sewer overflows by large-scale mathematical programming. Comput. Oper. Res. **1**(3–4), 421–435 (1974)
6. A. Pirsing, R. Rosen, B. Obst, F. Montrone, Einsatz mathematischer optimierungsverfahren bei der abflusssteuerung in kanalnetzen. Gas – Wasserfach, Wasser Abwasser **147**(5), 376–383 (2006)
7. M. Pleau, H. Colas, P. Lavallee, G. Pelletier, R. Bonin, Global optimal real-time control of the Quebec urban drainage system. Environ. Model. Softw. **20**(2005), 401–413 (2005)
8. M. Pleau, G. Pelletier, H. Colas, P. Lavallee, R. Bonin, Global predictive real-time control of Quebec urban community's westerly sewer network. Water Sci. Technol. **43**(7), 123–130 (2001)
9. W. Rauch, P. Harremoes, Genetic algorithms in real time control applied to minimize transient pollution from urban wastewater systems. Water Res. **33**(5), 1265–1277 (1999)
10. W. Schilling, 15 jahre kanalnetzsteuerung in den USA – was wurde erreicht? Korresp. Abwasser **33**(2), 147–151 (1986)
11. M. Schuetze, D. Butler, M. Bruce Beck, *Modelling, Simulation and Control of Urban Wastewater Systems* (Springer, Berlin, 2002)
12. Siemens, Siemens – annual report 2009, Technical report, 2010
13. R. Tandler, Ansaetze für eine parallele ueberstauberechnung von kanalnetzen. Korresp. Abwasser **41**(10), 1750–1761 (1994)
14. P.D. Trotta, J.W. Labadie, N.S. Grigg, Automatic control strategies for urban stormwater. J. Hydraul. Div. **103**(12), 1443–1459 (1977)
15. M. Weyand, W. Schilling, J. Broll-Bickhardt, Wirtschaftlichkeit und effektivitaet der kanalnetzsteuerung. Korresp. Abwasser **47**(2), 223–232 (2000)

S. Heusch · M. Ostrowski
Ingenieurhydrologie und Wasserbewirtschaftung, Technische Universität Darmstadt,
Petersenstr. 13, 64287 Darmstadt, Germany

S. Heusch
e-mail: heusch@ihwb.tu-darmstadt.de

M. Ostrowski (✉)
e-mail: ostrowski@ihwb.tu-darmstadt.de

H. Hanss
Siemens AG, I IS IN 1 WDC, Siemensallee 84, 76187 Karlsruhe, Germany
e-mail: holger.hanss@siemens.com

R. Rosen · A. Sohr
Siemens AG, CT T DE TC 3, Otto-Hahn-Ring 6, 81730 Munich, Germany

R. Rosen
e-mail: roland.rosen@siemens.com

A. Sohr
e-mail: annelie.sohr@siemens.com

Chapter 6
Modeling of Channel Flows with Transition Interface Separating Free Surface and Pressurized Channel Flows

Saeid Moradi Ajam, Yongqi Wang, and Martin Oberlack

Abstract In practical application open-channel or free-surface channel flow under the influence of gravity in sewers has traditionally been modeled with mathematical models based on one-dimensional governing equations of continuity and momentum—the so-called Saint Venant equations. High volumetric flow rates or strong rains may lead to the transition from partial to fully filled cross sections in a sewer net, i.e. a free surface flow is not guaranteed any more. Hence the mathematical model of the Saint Venant equations loses its validity in whole or in parts of the channels and a transition occurs to the pressurized pipe equations. The main goal of this work is to bring forward our knowledge about the process of changing the governing regime of the fluid equations in the channel flow and to attempt to perform a general modeling tracking the movement of the transition interface between a free surface flow and the pressurized flow in one-dimensional channels. Various flow cases with or without a moving transition are numerically investigated by means of the high-precision Discontinuous Galerkin Finite Element method. An exact knowledge of this event allows to optimize the controlling of equipment and the operation in a sewer or design a new sewer correctly and effectively.

6.1 Introduction

Sewer systems are designed usually to operate flows under free-surface flow conditions for normal rain events, i.e. the water is conveyed through the gravity and the air above the flow with standard atmospheric pressure does not affect the overall dynamics of the flow. The transition from the free surface to pressurized flow can be caused by intense rain events, failure of a pumping station or blockage of a pipeline, etc. Flows in a channel can be divided into three different cases: fully free surface flow, partially free surface flow an partially pressurized flow with a moving transition interface, as well as completely pressurized flow. In the normal situation, the flow in the whole channel is free surface flow. If the flow depth reaches the top of the closed channel due to strongly increasing volumetric flow rate, partially free surface and partially pressurized flow will occur with a moving infinitesimal thin section between the pressurized and the free surface flow domains. The moving transition interface develops and propagates towards the upstream direction. The downstream zone behind this interface is extended gradually. When the moving interface comes

A. Martin et al. (eds.), *Mathematical Optimization of Water Networks*,
International Series of Numerical Mathematics 162,
DOI 10.1007/978-3-0348-0436-3_6,

at the upstream end-boundary, the flow is pressurized everywhere in the whole channel. When the influx to the channel decreases, the sewer system starts losing water, and the free-surface flow region may emerge again close to the upstream end of the channel and the moving interface propagates towards the downstream end-boundary and the free-surface flow region expands gradually to the entire of the channel. Similar phenomena occur when the outflow of a channel flow decreases or increases. In practical applications gravity-driven free surface flows in sewers are traditionally modeled by the St. Venant equations based on one-dimensional governing equations of continuity and momentum, which lose their validity for the pressurized flow. To simulate the flow dynamic behaviors, it is important to capture the moving transition interface. The transition phenomena are highly dynamic and may cause serious operational problems such as Geysering, and structural damages in sewer systems as reported by [17] and [26]. The mathematical modeling of the transition between the free surface flow and the pressurized flow is of special importance for many practical applications such as optimization of controlling of the drainage systems, design of hydraulic structures and prediction of events in water sharing systems, sewer network and sewage pipelines, etc. The primary objective of this study is to present a theoretical method and a numerical technique for determining the moving interface (transition point) and simulating the channel flow including free surface and pressurized flow domains. In the present work, a mathematical model describing the motion of the transition interface is derived, which is demonstrated to be consistent with the Rankine-Hugoniot condition when at the transition point a jump occurs. An explicit Discontinuous Galerkin scheme is employed to solve the governing equations both in the free surface flow and in the pressured pipe flow.

6.2 Basic Equations

The mixed flow of free-surface and pressurized flows is a problem that appears in some hydraulic structures. In different flow regimes, the governing equations are different, which will be given below. An important task in the modeling of the mixed flow is to capture the motion of the transition point between free-surface and pressurized flows. If the transition point is discontinuous, its motion can be easily traced mathematically by the Rankine-Hugoniot condition.

6.2.1 Free Surface Flow

Flow in an open channel, in which the convection plays an important role, is in general unsteady and can be governed by the shallow water or Saint Venant equations, which were derived by the principles of the conservation of mass and the conservation of momentum. The free-surface flows may be classified into four regimes based on the values of the Froude number (Fr), representing the ratio of the flow velocity to the shallow water wave speed, and Reynolds numbers (Re) as follows

- Subcritical, Laminar: $Fr < 1.0$ and $Re \leq 500$.
- Supercritical, Laminar: $Fr > 1.0$ and $Re \leq 500$.
- Subcritical, Turbulent: $Fr < 1.0$ and $Re \geq 2000$.
- Supercritical, Turbulent: $Fr > 1.0$ and $Re \geq 2000$.

In the present we investigate only the turbulent cases of channel flows. In an unsteady state the volumetric flow rate varies as a function of time and position, therefore, all the hydraulic variables in a given cross section change as a function of time. Along the axial direction of the channel, all the flow quantities can be treated virtually as one-dimensional, i.e., a quasi one-dimensional flow assumption is adopted. Therefore we consider the local averaged cross-sectional velocity $v = v(x,t)$ and the elevation of the water above the channel bed $h = h(x,t)$ as the investigated dependent variables. The major assumptions in deriving the governing equations are the following:

- The pressure distribution in the vertical direction is hydrostatic.
- The channel bottom slope is small and fixed in time.
- The velocity is uniform within a cross section.
- Water is an incompressible and homogeneous fluid.
- The bottom and lateral walls of the channel are impermeable to water.

Under these assumptions the conservation equations can be reduced to the well-known Saint Venant equations

$$\begin{aligned} &\frac{\partial A}{\partial t} + \frac{\partial Q}{\partial x} = 0, \\ &\frac{\partial Q}{\partial t} + \frac{\partial}{\partial x}\left(\frac{Q^2}{A} + gI_1\right) = gA(S_0 - S_f) + gI_2, \end{aligned} \tag{6.1}$$

where as the independent variables t is time, and x is the coordinate along the axial direction of the channel, while as the dependent physical quantities A is the flow cross-section area and Q is the total flow rate. The bed slope S_0 is the spatial partial derivative of the bottom elevation $z_b(x)$, $S_0 = -\mathrm{d}z_b/\mathrm{d}x$, and g is acceleration due to gravity. The friction slope S_f is defined in terms of the Manning's roughness coefficient n as

$$S_f = \frac{n^2 Q|Q|}{A^2 R^{4/3}},$$

where R is the hydraulic radius defined as $R = A/\ell$, with ℓ representing the wetted perimeter. The hydraulic pressure I_1 and the wall pressure I_2 are determined by

$$I_1 = \int_0^{h(x,t)} \big(h(x,t) - y\big) B(x,y)\mathrm{d}y, \qquad I_2 = \int_0^{h(x,t)} \big(h(x,t) - y\big) \frac{\partial B(x,h)}{\partial x}\mathrm{d}y, \tag{6.2}$$

where $B(x,y)$ is the width of the channel, $B(x,h)$ the width of the water surface, and $h(x,t)$ is the water depth (see Fig. 6.1). If we assume that the cross section of

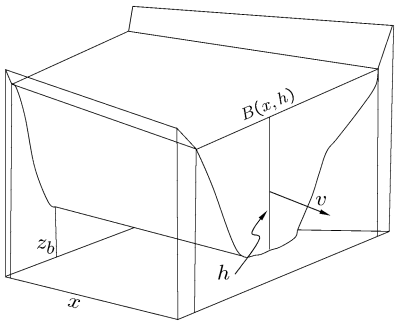


Fig. 6.1 A sketch of open channel flow

the channel is rectangular and uniform along the channel, i.e. $B = \text{const}$, the above equations can be reduced to

$$\begin{aligned} &\frac{\partial h}{\partial t} + \frac{\partial (vh)}{\partial x} = 0, \\ &\frac{\partial v}{\partial t} + \frac{\partial}{\partial x}\left(\frac{v^2}{2} + gh\right) = g(S_0 - S_f), \end{aligned} \tag{6.3}$$

and the hydraulic radius can be written as $R = (Bh)/(B + 2h)$.

The simplified one-dimensional shallow water equations (6.3) can be rewritten in vector form as

$$\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathbf{F}(\mathbf{U})}{\partial x} = \mathbf{S}(\mathbf{U}), \tag{6.4}$$

in which the vector $\mathbf{U}$ stands for the conserved physical variables, $\mathbf{F}(\mathbf{U})$ for the corresponding flux vector and $\mathbf{S}(\mathbf{U})$ for the source vector defined as follows

$$\mathbf{U} = \begin{bmatrix} h \\ q \end{bmatrix}, \qquad \mathbf{F}(\mathbf{U}) = \begin{bmatrix} q \\ \frac{q^2}{h} + \frac{gh^2}{2} \end{bmatrix}, \qquad \mathbf{S}(\mathbf{U}) = \begin{bmatrix} 0 \\ gh(S_0 - S_f) \end{bmatrix}, \tag{6.5}$$

where q indicates the flow rate per channel width, $q = vh = Q/B$. Some properties of the governing equations (6.4) can be investigated by its homogeneous quasilinear form

$$\mathbf{U}_t + \mathbf{F}'(\mathbf{U})\mathbf{U}_x = 0,$$

where $\mathbf{F}'(\mathbf{U})$ is the Jacobian matrix

$$\mathbf{F}'(\mathbf{U}) = \frac{\partial \mathbf{F}}{\partial \mathbf{U}} = \begin{bmatrix} 0 & 1 \\ -v^2 + gh & 2v \end{bmatrix}. \tag{6.6}$$

The eigenvalues λ_i, $(i = 1,\ 2)$ of the matrix $\mathbf{F}'(\mathbf{U})$, corresponding to the respective celerities $\mathrm{d}x/\mathrm{d}t$ of the two characteristics, can be obtained by

$$\det\left(\mathbf{F}'(\mathbf{U}) - \lambda \mathbf{I}\right) = 0 \quad \Rightarrow \quad \lambda_1 = v - \sqrt{gh}, \quad \lambda_2 = v + \sqrt{gh}. \tag{6.7}$$

Because the eigenvalues λ_i, $(i = 1, 2)$ are real and distinct, the equation system for shallow water flows is strictly hyperbolic. By linearizing these equations by considering very small perturbations at a constant depth h_0 and a constant velocity v_0, it can be observed that these perturbations move with velocities $v_0 + c_0$ and $v_0 - c_0$, where $c_0 = \sqrt{gh_0}$ is the wave celerity. These waves propagate at the speed $\pm\, c_0$ relative to the fluid [16]. Such properties of the hyperbolic system can be used to decompose the Saint Venant equation to two decoupled ordinary differential equations, which hold along characteristic curves, $\mathrm{d}x/\mathrm{d}t = \lambda_1$ and $\mathrm{d}x/\mathrm{d}t = \lambda_2$. They can be solved along the characteristic curves and transformed into a solution for the original equations. This method will be used to determine the solution of the Riemann problem, that will be discussed later. The celerity will also be used to define the stable condition of explicit numerical methods known as the CFL condition.

6.2.2 Shock Waves

If physical quantities change instantly in a very small spatial distance, these quantities can be considered as discontinuity functions mathematically. A discontinuity may occur in shallow water flows when the transition from subcritical flow ($Fr < 1$) occurs to supercritical flow ($Fr > 1$) such as the dam-breaking problem or vice versa e.g. in the case of hydraulic jump. The differential form of the Saint Venant equations does not apply to discontinuities. At the discontinuity the integral form of the conservation laws should be employed. The discontinuity propagates as a shock wave with speed w determined by the Rankine-Hugoniot condition

$$w = \frac{F(U_R) - F(U_L)}{U_R - U_L},$$

where U_L and U_R are, respectively, the left and the right values of the conserved variables immediately near the discontinuity.

6.2.3 Pressurized Flow

A pipe flow or pressurized flow is distinguished from an open channel or free surface flow by the absence of a free surface, i.e. the flow in a channel fills completely the cross section. Usually a pipe flow is driven by pressure gradient. The continuity and momentum equations for pressurized pipe flows in a rectangular channel can be expressed as (see e.g. [20])

$$\begin{aligned} &\frac{\partial}{\partial x}(vh) = 0, \\ &\frac{\partial v}{\partial t} + \frac{\partial}{\partial x}\left(\frac{v^2}{2} + \frac{P}{\rho}\right) = g\left(S_0 - f\frac{|v|v}{4gR}\right), \end{aligned} \tag{6.8}$$

where ρ is the water density, f the dimensionless Darcy-Weisbach friction factor, R the hydraulic radius and P the pressure.

If the channel height is constant, $h = \text{const.}$, these governing equations are reduced to

$$v = v(t), \qquad \frac{\partial v}{\partial t} + \frac{1}{\rho}\frac{\partial P}{\partial x} = g\left(S_0 - f\frac{|v|v}{4gR}\right). \tag{6.9}$$

6.3 Review of Existing Flow Regime Transition Models

The study of the transition between free-surface and pressure flow is of great importance in water sharing systems, sewer network and sewage pipelines, etc. In the last decades, many researches have investigated channel flows with transition point. These methods can be divided in two groups: (i) the methods under the assumption of constant water density, that is physically reasonable, and (ii) the methods with the assumption of variable water density, that is employed to modify the type of the system of governing equations in the pressurized flow regime and hence maintain the types of the governing equations unchanged in the whole flow domain. A short summary of the existing procedures concerning the transition problem is as follows:

- Assumption of incompressibility of water: ($\rho = \text{const.}$)
  - The *Rigid Column technique* is characterized by the assumption of a jump of the water depth in the transition interface and a constant water height in the free surface flow.
  - The *Preissmann slot approach* is characterized by assuming a virtual slot in the upper wall of the channel. The whole channel can be considered completely as free surface flow.
- Assumption of compressibility of water: ($\rho \neq \text{const.}$)
  - The *Shock Fitting method* is an extended form of the Method of Characteristics, in which the change of density of water and a jump of the variables in the transition point are assumed.
  - The derivation of a new system of equations for both different flow regimes based on the adoption of an elastic pipe and by employing the Riemann solver to determine the position of the transition point see [2].

More details of these techniques are given in the following subsections.

6.3.1 Rigid Column Technique

This class of flow regime transition models is proposed by [9, 26] and [23], among others. These models solve an ordinary differential equation (ODE) based on a momentum balance in a rigid column represented by the pressurized portion of the flow. In each time step, the ODE is solved and the velocity of the rigid column is

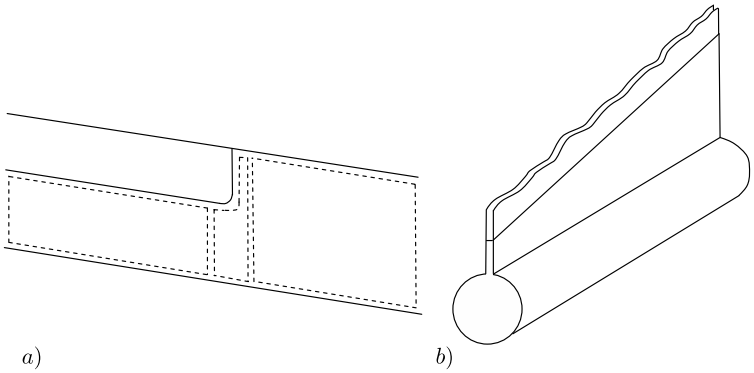


Fig. 6.2 Sketch of a rigid column pressurized flow with a jump at the transition (**a**) and a virtual Preissmann slot (**b**)

then updated. The location of the pressurization front is obtained using the continuity equation across the moving interface (see Fig. 6.2a). The method of the characteristics (MOC) technique is employed to calculate the free surface portion of the flow. The air phase as an air pocket is also modeled with compressible flow theory. These models may be restricted by a number of limitations, as described by [19] and [24]. If the relative velocity between the air and water phases exceeds a limit a large deformation in air pockets occurs that can cause instabilities. Another important limitation of this method is the assumption of the occurrence of a jump at the moving interface.

6.3.2 Preissmann Slot Technique

The Preissmann slot approach is applied in solving free surface and pressurized channel flows by [6, 7, 12] and [15]. The introduction of a hypothetical slot at the top of the pipe (see Fig. 6.2b) to simulate the pressurization was originally suggested by Preissmann in 1961 and employed by [7]. The slot assures that the free surface is preserved regardless of the pressure in the cross section and allows to analyze both free surface and pressurized flow by the St. Venant equations. The slot can be sized so that the wave celerity in the slot is equal to the water hammer wave speed and height of water is equal to the water hammer head. Because a single set of governing equations is employed throughout the whole flow domain, there is no need to track the transition interface between free surface and pressurized flow moving throughout the flow domain. The disadvantage of this model is the occurrence of numerical oscillations for transition from partially to completely full flows [6]. [8] applied the Preissmann slot technique to a combined sewer in Chicago. It was observed that this method became unstable for sharp slopes.

6.3.3 Shock Fitting Method

The shock fitting method is an alternative to model the transition in the mixed flow. [25] modified the Preissmann method by introducing a moving interface. [19] developed this idea using the method of characteristics. These models exploit the similarities between the Saint Venant equations and the equivalent mass and momentum equations for closed pipes. A shock-fitting technique approach is implemented in which the free surface and pressurized flow regimes are solved using equations that are appropriate to the particular regimes. The transition point is regarded as a moving discontinuity and tracked explicitly by means of jump conditions as an internal boundary between both flows. The drawback of this method is the modeling of complex wave interaction at the junctions in a sewer [21].

6.4 A New Flow Regime Transition Model

In this stage we attempt to deduce a general mathematical model to rigorously trace the moving transition point between the free surface flow and the pressurized pipe flow. The velocity of the transition point is a dynamic function of the flow states on its both adjacent sides. In this model, the transition condition between the free surface flow and the pressurized flow is deduced by tracking a moving infinitesimal thin section between the pressurized and the free surface flow domains. The transition velocity is determined for a finite slope of the water depth (without jump) near the transition point. It will be demonstrated that the Rankine-Hugoniot condition for the jump case is the limiting case of this formulation.

The present mathematical model can be considered as an extension of the *interface tracking* approaches. We will attempt to track the movement of the transition point by means of the transition conditions and the conservation laws for the control volume moving together with the transition point, without any additional assumption e.g. variable water density or a jump assumption in the transition point, as done in many existing methods.

Firstly, we employ a moving coordinate system fixed at the transition point,

$$\tilde{x} = x + wt \tag{6.10}$$

where $\tilde{x}$ is the space coordinate in the moving coordinate system and w is the velocity of the transition point, which will be determined. In general the velocity w is a function of time t. In our simulation the velocity of the transition point will be updated for each time step. Here we assume that w within a time step is unchanged. The position of the transition point will be calculated according to (6.10) in each time step. In the moving coordinate system, the governing equations of the

free surface flow (6.3) can be rewritten as

$$\begin{aligned}\frac{\partial h}{\partial t}+(v-w)\frac{\partial h}{\partial \tilde{x}}+h\frac{\partial v}{\partial \tilde{x}}&=0,\\ \frac{\partial v}{\partial t}+(v-w)\frac{\partial v}{\partial \tilde{x}}+g\frac{\partial h}{\partial \tilde{x}}&=F_{\tilde{x}},\end{aligned}\tag{6.11}$$

where $F_{\tilde{x}}$ is the sum of the friction force and the gravity in the flow direction. The continuity of velocity and water depth through the transition point is assumed. The case of a jump will be discussed later. This means that the flow state at the pressurized flow near the transition point, the water depth (equal to the height of the channel height) H and the flow velocity v_{pipe}, are considered as the boundary conditions for the free surface flow at the transition point $\tilde{x}=0^-$, i.e.,

$$h\big(\tilde{x}=0^-,t\big)=H,\qquad v\big(\tilde{x}=0^-,t\big)=v_{\text{pipe}}.\tag{6.12}$$

For the case of continuity, the v_{pipe} can be replaced by the velocity at the transition point in the free surface region v_1. The Taylor expansions of these variables in the free surface region near the transition point and for a small time variation can be written as

$$\begin{aligned}h(\tilde{x},t)&=h|_{(0^-,0)}+\tilde{x}\frac{\partial h}{\partial \tilde{x}}\bigg|_{(0^-,0)}+t\frac{\partial h}{\partial t}\bigg|_{(0^-,0)}+\tilde{x}t\frac{\partial^2 h}{\partial t\partial \tilde{x}}\bigg|_{(0^-,0)}+\frac{1}{2}\tilde{x}^2\frac{\partial^2 h}{\partial \tilde{x}^2}\bigg|_{(0^-,0)}\\&\quad+\frac{1}{2}t^2\frac{\partial^2 h}{\partial t^2}\bigg|_{(0^-,0)}+\cdots,\\ v(\tilde{x},t)&=v|_{(0^-,0)}+\tilde{x}\frac{\partial v}{\partial \tilde{x}}\bigg|_{(0^-,0)}+t\frac{\partial v}{\partial t}\bigg|_{(0^-,0)}+\tilde{x}t\frac{\partial^2 v}{\partial t\partial \tilde{x}}\bigg|_{(0^-,0)}+\frac{1}{2}\tilde{x}^2\frac{\partial^2 v}{\partial \tilde{x}^2}\bigg|_{(0^-,0)}\\&\quad+\frac{1}{2}t^2\frac{\partial^2 v}{\partial t^2}\bigg|_{(0^-,0)}+\cdots,\\ w(t)&=w|_{(0^-,0)}+t\frac{\partial w}{\partial t}\bigg|_{(0^-,0)}+\cdots=w_0+w_1t+\cdots.\end{aligned}\tag{6.13}$$

Applying the transition condition $(6.12)_1$ to $(6.13)_1$ yields the trivial conditions

$$\frac{\partial h}{\partial t}\bigg|_{(0^-,0)}=0,\qquad \frac{\partial^2 h}{\partial t^2}\bigg|_{(0^-,0)}=0,\quad \ldots.$$

Substituting the expansion series (6.13) to (6.11) and employing the conditions (6.12) yield the zero-order condition

$$(v_1-w_0)\frac{\partial h}{\partial \tilde{x}}\bigg|_{(0^-,0)}+H\frac{\partial v}{\partial \tilde{x}}\bigg|_{(0^-,0)}=0.$$

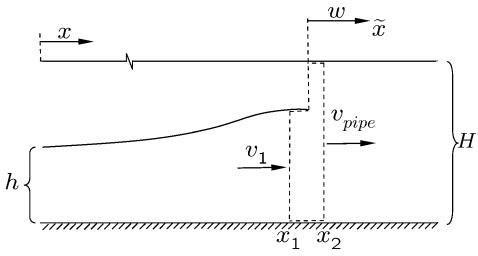


Fig. 6.3 A sketch of channel flow with the assumption of jump on transition point

Hence the velocity of the transition point for any time can be determined by

$$w_0 = v_1 + H \frac{\frac{\partial v}{\partial \tilde{x}}\Big|_{(0^-,0)}}{\frac{\partial h}{\partial \tilde{x}}\Big|_{(0^-,0)}} = v_1 + H \left.\frac{\frac{\partial v}{\partial \tilde{x}}}{\frac{\partial h}{\partial \tilde{x}}}\right|_{\text{transition point}}. \tag{6.14}$$

As pointed out above, in numerical simulations the velocity of the transition point is assumed to be constant within a time step and updated in each time step. It is taken $w \approx w_0$. This approach is similar to a Eulerian forward time step for the calculation of w. With this condition the motion of transition point can be directly tracked. In the above derivation the continuity of the velocity and the water depth at the transition point is assumed. In many practical applications the change of physical quantities through the transition point may be considerably large, i.e., a jump may be occur. Actually, in many methods capturing the moving transition interface such as Shock Fitting method or Rigid Column method the existence of a jump on the transition interface is presupposed in order to calculate the velocity of the transition point. For the case of a jump at the transition point, employing the continuity equation for the moving control volume at the transition point (Fig. 6.3), yields

$$(v - w)h\big|_{x_2} = (v - w)h\big|_{x_1}$$

where x_1 locates in the free surface region near the transition point whist x_2 is in the pressurized flow region. This relation can be written as

$$(v_{\text{pipe}} - w)H = (v_1 - w)h_1,$$

which results in the Rankine-Hugoniot condition, i.e.,

$$w = \frac{H v_{\text{pipe}} - h_1 v_1}{H - h_1}. \tag{6.15}$$

It can be easily demonstrated that this formulation of the velocity of the transition point is a limiting case of our result (6.14), if the gradients emerging in (6.14) are replaced by

$$\left.\frac{\partial v}{\partial \tilde{x}}\right|_{\text{transition point}} = \frac{v_{\text{pipe}} - v_1}{\Delta x}, \qquad \left.\frac{\partial h}{\partial \tilde{x}}\right|_{\text{transition point}} = \frac{H - h_1}{\Delta x},$$

which may be employed in numerical simulations when the relation (6.14) is used. Substituting these relations into (6.14) yields the same condition (6.15).

6.5 Discontinuous Galerkin Scheme for Numerical Simulation of the Shallow Water Equations

In many practical applications, e.g. in meteorology, weather-forecasting, oceanography, gas dynamics, turbulent flows among many others, convection plays an important role. The conservation laws for free surface flows are hyperbolic convection equations in the form

$$\begin{aligned} &\partial_t U + \partial_x F(U) = S(U), \quad \text{in } [L, R] \times [0, T], \\ &U(x, 0) = U_0(x). \end{aligned} \tag{6.16}$$

In general, these differential equations can be solved only numerically. Satisfactory numerical modeling of convection presents a well-known dilemma to the computational fluid dynamicist. On the one hand, traditional high-order schemes lead often to unphysical oscillatory behavior or disastrous non-convergence [4]. On the other hand, computations based on the classical first-order schemes often suffer from severe inaccuracies due to inherent numerical diffusions. Here we employ the Discontinuous Galerkin (DG) scheme with the TVD limiter to solve the shallow water equations. The idea is to employ the high-order DG scheme and simultaneously identify troubled cells where the oscillation may occur for which limiting is introduced [18].

For this purpose, the computational domain $\Omega \in [L, R]$ is discretized into N non-overlapping elements $I_j = [x_{j-1/2}, x_{j+1/2}]$ with boundary points $L = x_{1/2} < x_{3/2} < \cdots < x_{N+1/2} = R$ and the cell size $\Delta_j = x_{j+1/2} - x_{j-1/2}$. The approximation of the solution $U_h(x, t)$ over each cell I_j by using the kth order of polynomials can be written as

$$U_h(x, t) = \sum_{m=0}^{k} U_j^m(t) \Phi_j^m(x) \quad \text{for } x \in I_j, \tag{6.17}$$

where $\Phi_j^m(x)$ are the local base functions over I_j and k is the order of the approximation. To decouple the discretized equation system, we adopt the Gram-Schmidt polynomials GS^m, for which the lower-order functions ($m \leq 3$) are listed in Table 6.1, as local basis functions in the form

$$\Phi_j^m(x) = GS_j^m\left(\frac{2(x - x_j)}{\Delta_j}\right).$$

The orthogonal property of the basis functions can be written as follows

$$\int_{I_j} \Phi_j^m(x) \Phi_j^l(x) \mathrm{d}x = \delta_{ml}, \qquad \delta_{ml} = \begin{cases} 1, & m = l, \\ 0, & m \neq l. \end{cases} \tag{6.18}$$

Table 6.1 Basis polynomials of Gram-Schmidt's procedure

Order of polynomial	Basis function
0	$GS^0(x) = \frac{\sqrt{2}}{2}$
1	$GS^1(x) = \frac{\sqrt{6}}{2}x$
2	$GS^2(x) = \frac{3\sqrt{10}}{4}x^2 - \frac{\sqrt{10}}{4}$
3	$GS^3(x) = \frac{5\sqrt{14}}{4}x^3 - \frac{3\sqrt{14}}{4}x$

Substituting (6.17) into the conservation equation (6.16), multiplying by the local base function $\Phi_j^m(x)$, and then integrating the equation over the cell I_j yield

$$\int_{I_j} \partial_t U_h(x,t)\Phi_j^m(x)\mathrm{d}x + \int_{I_j} \partial_x F\big(U_h(x,t)\big)\Phi_j^m(x)\mathrm{d}x = \int_{I_j} S\big(U_h(x,t)\big)\Phi_j^m(x)\mathrm{d}x. \tag{6.19}$$

By means of the partial integration and the orthogonality of the base function (6.18), the evolution equation of the approximate solution for each order coefficient over each cell I_j, U_j^m, can be obtained in the form

$$\frac{\partial U_j^m}{\partial t} = L\big(U_h, \Phi_j^m\big) \quad \text{for } m = 0, 1, 2, \ldots. \tag{6.20}$$

Here the operator $L(U_h, \Phi_j^m)$ is defined as follows

$$\begin{aligned} L\big(U_h, \Phi_j^m\big) = &-\tilde{F}\big(U^L_{j+\frac{1}{2}}, U^R_{j+\frac{1}{2}}\big) + \tilde{F}\big(U^L_{j-\frac{1}{2}}, U^R_{j-\frac{1}{2}}\big) + \int_{I_j} F(U_h)\frac{\partial \Phi_j^m}{\partial x}\mathrm{d}x \\ &+ \int_{I_j} S(U_h)\Phi_j^m \mathrm{d}x, \end{aligned} \tag{6.21}$$

where $U^{L,\,R}_{j+\frac{1}{2}} = U_h(x^{L,\,R}_{j+\frac{1}{2}}, t)$ representing the left and right limits of the discontinuous solution U_h at the cell interface and $\tilde{F}(U^L, U^R)$ numerical flux function based on the exact or approximate Riemann solver, which will be discussed in the next section. The time discretization will be performed by a high-order Runge-Kutta scheme. The volume integrals emerging in (6.21) can be determined using the high-order accurate Gaussian quadrature rules, e.g.

$$\int_{I_j} F(U_h)\partial_x \Phi_j^m \mathrm{d}x = \Delta_j \sum_{i=1}^{N} \omega_i F\big(U_h(\xi_i, t)\big)\partial_x \Phi_j^m(\xi_i),$$

by a suitable choice of the points $\xi_i \in I_j$ and weights ω_i for $i = 1, \ldots, N$.

The projection of the initial condition by mth degrees of freedom can be written as

$$U_j^m(x,0) = \int_{I_j} U(x,0)\Phi_j^m(x)\mathrm{d}x.$$

6.6 Numerical Formulation of Fluxes

With the Discontinuous Galerkin method, the numerical solution is discontinuous with a jump at each cell interface, i.e., there exist two distinct values at the cell interface, U^L and U^R. To determine the single solution at the cell interface, or more precisely in the DG method, to obtain the expression for the flux on the cell interface, $\tilde{F}(U^L, U^R)$, the so-called Riemann problem is presented. The Riemann problem can be provided for a system of equations at the cell interface $x = 0$ in the form

$$U_t + F(U)_x = 0, \qquad U(x,0) = \begin{cases} U_L, & x < 0, \\ U_R, & x > 0, \end{cases} \tag{6.22}$$

where U_L and U_R are, respectively, the left and the right constant variables of the initial discontinuity at $x = 0$. The purpose is to find direct approximations to the function $F(U)$ at the discontinuity $x = 0$. There exists many procedures to do this. One of them is the approximate Riemann solver postulated by Harten, Lax and van Leer (HLL) [10].

Consider the control volume $[x_L, x_R] \times [U_L, U_R]$, displayed in Fig. 6.4, with $x_L \leq TS_L$ and $x_R \geq TS_R$, where S_L and S_R are the wave speeds, respectively, and T is a chosen time. Integrating the governing equation (6.22) in the control volume $ABCE$ yields

$$\begin{aligned} \int_{x_L}^{x_R} U(x,T)\mathrm{d}x &= \int_{x_L}^{x_R} U(x,0)\mathrm{d}x + \int_0^T F\big(U(x_L,t)\big)\mathrm{d}t - \int_0^T F\big(U(x_R,t)\big)\mathrm{d}t \\ &= x_R U_R - x_L U_L + T\big(F(U_L) - F(U_R)\big). \end{aligned} \tag{6.23}$$

The integral on the left-hand side can be split into three integrals, namely

$$\begin{aligned} \int_{x_L}^{x_R} U(x,T)\mathrm{d}x &= \int_{x_L}^{TS_L} U(x,T)\mathrm{d}x + \int_{TS_L}^{TS_R} U(x,T)\mathrm{d}x + \int_{TS_R}^{x_R} U(x,T)\mathrm{d}x \\ &= (TS_L - x_L)U_L + \int_{TS_L}^{TS_R} F\big(U(x_L,t)\big)\mathrm{d}t + (x_R - TS_R)U_R. \end{aligned} \tag{6.24}$$

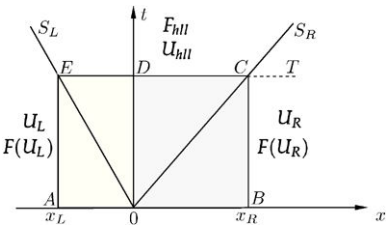


Fig. 6.4 Control volume $[x_L, x_R] \times [0, T]$ on x–t plane for the two-wave structure of the Riemann problem solution assumed in the HLL approach

Comparison of (6.24) with (6.23) gives

$$\int_{TS_L}^{TS_R} U(x,T)\mathrm{d}x = T\big(S_R U_R - S_L U_L + F(U_L) - F(U_R)\big).$$

Hence, the integral average of the exact solution of the Riemann problem between the slowest and fastest wave speeds at time T is

$$\frac{1}{T(S_R - S_L)} \int_{TS_L}^{TS_R} U(x,t)\mathrm{d}t = U_{hll} = \frac{S_R U_R - S_L U_L + F(U_L) - F(U_R)}{S_R - S_L}. \tag{6.25}$$

To estimate the corresponding numerical flux F_{hll} on the discontinuity, integrating the conservation law (6.22) on the control volume *AODE* yields

$$\int_{TS_L}^{0} U(x,T)\mathrm{d}x = -TS_L U_L + T(F_L - F_{0L}), \tag{6.26}$$

where F_{0L} is the flux $F(U)$ along the t-axis. Substituting the integrand by U_{hll} by (6.25) gives

$$F_{hll}^L = F_{0L} = F(U_R) + S_R(U_{hll} - U_R). \tag{6.27}$$

If the control volume *OBCD* is employed, a similar result is obtained, i.e.,

$$F_{hll}^R = T_{0R} = F(U_L) + S_L(U_{hll} - U_L). \tag{6.28}$$

It can be easily verified that $F_{hll}^L = F_{hll}^R = F_{hll}$ if the relation for U_{hll} in (6.25) is used. This yields the HLL flux as

$$F_{hll} = \frac{S_R F(U_L) - S_L F(U_R) + S_L S_R (U^R - U^L)}{S_R - S_L}.$$

The corresponding HLL numerical flux for the approximate Godunov method at the discontinuous cell interface is then given by

$$\tilde{F}(U_L, U_R) = \begin{cases} F(U_L), & 0 \le S_L, \\ F_{hll}, & S_L \le 0 \le S_R, \\ F(U_R), & 0 \ge S_R. \end{cases}$$

For the shallow water equations, the wave speeds can be determined by

$$S_L = \min\big(v_L + \sqrt{gh_L}, v^* - \sqrt{gh^*}\big), \qquad S_R = \min\big(v_R + \sqrt{gh_R}, v^* - \sqrt{gh^*}\big) \tag{6.29}$$

with

$$v^* = \frac{1}{2}(v_L + v_R) + \sqrt{gh_L} - \sqrt{gh_R}, \qquad \sqrt{gh^*} = \frac{1}{2}\big(\sqrt{gh_L} + \sqrt{gh_R}\big) - \frac{1}{4}(v_R - v_L), \tag{6.30}$$

where v_L, v_R, h_L and h_R are, respectively, the left and the right flow velocities and water depths at the cell interface. There exist more other possible Riemann solvers employed for the shallow water equations such as the Osher Riemann solver, the Roe Riemann solver etc. (see [22] and [3]).

6.7 Numerical Stability and Limiters

As shown in the previous part, in the DG method, an approximate solution $U_h(x,t)$ is constructed over each cell I_j. The simplest construction for $U_h(x,t)$ is a piecewise constant function, corresponding only to a first-order accurate DG scheme. For the first-order schemes, due to their inherent numerical diffusions, the solution near a discontinuity is often smeared out and hence the discontinuity can hardly be captured. To obtain a better accuracy a better construction of $U_h(x,t)$, i.e. a higher-order DG scheme, may be employed, for example, a piecewise linear function with a nonzero slope. It is well-known from Godunov's Theorem [21] that *all schemes of accuracy greater than one will produce spurious oscillations in the vicinity of discontinuities*. Many practical works based on various high-order numerical schemes have confirmed this theorem. The limiter schemes may be the most familiar and useful procedure to restrict or suppress such spurious oscillations in the vicinity of discontinuities when the high order of accuracy is employed in numerical methods. To detect the regions where spurious oscillations may occur, a so-called discontinuity detector was suggested by [14]. According to [14] smooth DG solutions of hyperbolic conservation laws exhibit a strong super-convergence phenomena at outflow boundaries $\partial\Omega^+$ of Ω_j such that

$$\frac{1}{|\partial\Omega_j^+|}\int_{\partial\Omega_j^+}(Q_j-q)\mathrm{ds}=\mathscr{O}\big(\Delta x^{2k+1}\big),$$

where q is the exact solution, Q_j is the numerical solution of q on Ω_j and k is the employed DG order. To construct the detector, consider the jump in Q_j across the inflow boundaries $\partial\Omega^-$ of the domain Ω_j and examine

$$D_j=\int_{\partial\Omega_j^-}(Q_j-Q_{nb_j})\mathrm{ds}=\int_{\partial\Omega_j^-}(Q_j-q)\mathrm{ds}+\int_{\partial\Omega_{nb_j}^+}(q-Q_{nb_j})\mathrm{ds},$$

which yields $D_j=\mathscr{O}(\Delta x^{k+2})$ for a smooth solution. If q is discontinuous at $\partial\Omega_j$, then either or both of $q-Q_j$ and $q-Q_{nb_j}$ are $\mathscr{O}(1)$. Hence,

$$D_j=\begin{cases}\mathscr{O}(\Delta x^{k+2}), & \text{if } q \text{ is smooth on } \partial\Omega_j,\\ \mathscr{O}(\Delta x), & \text{if } q \text{ is discontinuous on } \partial\Omega_j.\end{cases} \tag{6.31}$$

According to this feature of D_j, we can construct a detector by normalizing D_j by [1]

$$\mathscr{D}_j = \frac{|\int_{\partial\Omega_j^-}(Q_j - Q_{nb_j})\mathrm{ds}|}{\Delta x^{\frac{(k+1)}{2}}|\partial\Omega_j^-|\|Q_j\|}.$$

Using (6.31), $\mathscr{D}_j \to 0$ when either $\Delta x \to 0$ or $k \to \infty$ in smooth regions, whereas $\mathscr{D}_j \to \infty$ near a discontinuity. Hence, the detector scheme is

$$\begin{cases} q \text{ is discontinuous,} & \text{if } \mathscr{D}_j > 1, \\ q \text{ is smooth,} & \text{if } \mathscr{D}_j < 1. \end{cases} \tag{6.32}$$

After detecting the discontinuous regions by means of the above detector, a suitable slope limiter can be employed to reconstruct the approximate solution $U_h(x,t)$ only in regions where it is needed. This is a way of maintaining high accuracy in smooth regions. This step is achieved by limiting the coefficients U_j^m of the DG approximate solution (6.17) [4]. Starting with the highest-oder coefficient $m = k$, we replace U_j^m with

$$\bar{U}_j^m = \mathrm{minmod}\left(U_j^m, \frac{(U_{j+1}^{m-1} - U_j^{m-1})}{2}, \frac{(U_j^{m-1} - U_{j-1}^{m-1})}{2}\right),$$

where the minmod function is defined by

$$\mathrm{minmod}(a,b,c) = \begin{cases} \mathrm{sgn}(a)\min(|a|,|b|,|c|), & \text{if } \mathrm{sgn}(a) = \mathrm{sgn}(b) = \mathrm{sgn}(c), \\ 0, & \text{otherwise.} \end{cases}$$

Roughly speaking, U_j^m corresponds to the m-order derivative of the solution, so it will be compared to the forward and backward differences of the $(m-1)$-order derivative, which are alternative approximations to the m-order derivative. The slope limiter is applied adaptively. The highest-order coefficient is first limited. The limiter is then applied to successively lower-order coefficients when the next higher-order coefficient on the interval is changed by the effect of limiting. Hence the limiting procedure is applied only to higher-order coefficients for which it is needed.

6.8 Test Problems and Numerical Results

In this section the performance of the numerical code employing the DG method will be demonstrated and some examples will be discussed.

6.8.1 Error Analysis for the Linearized Shallow-Water Equations with Smooth Initial Conditions

One of the most important criteria for a numerical method is to estimate its numerical error. For the nonlinear shallow water equations no analytical solution is available. For this purpose, we investigate the numerical solution of the linearized shallow water equations in the form

$$\frac{\partial h}{\partial t} + h_0 \frac{\partial v}{\partial x} = 0, \qquad \frac{\partial v}{\partial t} + g \frac{\partial h}{\partial x} = 0, \quad x \in [0, 1], \tag{6.33}$$

whose analytical solutions are attainable. By comparing the analytical solutions with the corresponding numerical results, the numerical performance of the code can be evaluated. For the continuous initial conditions

$$h(x, 0) = \tilde{H} \cos(-kx), \qquad v(x, 0) = \frac{\omega}{k h_0} h(x, 0), \tag{6.34}$$

the exact solutions are in the form of traveling waves

$$h(x, t) = \tilde{H} \cos(\omega t - kx), \qquad v(x, t) = \frac{\omega}{k h_0} h(x, t), \tag{6.35}$$

where $\tilde{H}$ is the wave amplitude, k is the wave number, ω the angular frequency, and $h_0 = \frac{\omega^2}{k^2 g}$ is the still-water depth with the assumption of $h \ll h_0$ [11]. To quantitatively discriminate how well the numerical code can describe the present linearized equations an error measure for the physical variable u (or h) is introduced by the error of the root mean square (rms)

$$E = \frac{\sqrt{\sum_j^N \int_{I_j} (U_h - U_{exact})^2 \mathrm{d}x}}{\sqrt{\sum_j^N \int_{I_j} (U_{exact})^2 \mathrm{d}x}},$$

where U_h is the numerical value in the cell j obtained with the k-order DG scheme and U_{exact} is the corresponding exact value in the cell j. The global error with the parameters $\tilde{H} = 1$, $k = 1$, $\omega = 1$ at the time point $t = 5$ is displayed in Fig. 6.5 for the variable h with reference of different orders of the DG scheme, k, and various numbers of the cell, N. For the given continuous initial conditions (6.34) no limiting is necessary according to the discontinuity detector (6.32). As expected, the numerical error decreases with the increase of k and N. It can also been seen that an approximate relation of $E \sim N^{-p}$ is valid where $p \approx k$ with some small deviations. For the variable u similar results can be obtained.

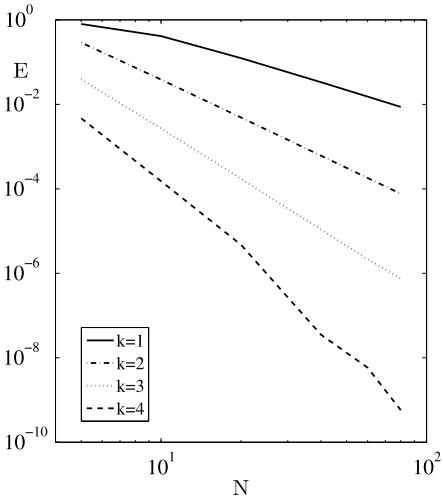


Fig. 6.5 Global rms error as a function of order of the DG approximation and number of element with the continuous initial conditions for the linearized shallow water equations

6.8.2 *Numerical Stability for the Linearized Shallow-Water Equations with Discontinuous Initial Conditions*

As indicated in Sect. 6.7, using the cell reconstruction technique with high-order accuracy may cause spurious oscillations near discontinuities, and hence possible instability. To suppress possible numerical oscillations, limiting may be needed in some identified cells. To investigate this behavior of the high-order DG method, the linear problem (6.33) for $x \in [0, 20]$ is investigated for discontinuous initial conditions

$$h(x,0) = h_L \mathscr{H}(10-x) + h_R \mathscr{H}(x-10), \qquad u(x,0) = 0, \tag{6.36}$$

where $\mathscr{H}(x)$ is a Heaviside function. The initial water depth is assumed as $h_L = 1$ for $x < 10$ and $h_R = 0.5$ for $x > 10$. The celerity is $c = 3\ \mathrm{ms}^{-1}$ and the eigenvalues are $\lambda_1 = -2$ and $\lambda_2 = 4$. The analytical solutions take the form [21]

$$\begin{aligned} h(x,0) &= 0.5\mathscr{H}(-x+3t+10) + 0.25\mathscr{H}(x-3t-10) \\ &\quad + 0.5\mathscr{H}(-x-3t+10) + 0.25\mathscr{H}(x+3t-10), \\ u(x,0) &= 1.6334\mathscr{H}(-x+3t+10) + 0.8167\mathscr{H}(x-3t-10) \\ &\quad - 1.6334\mathscr{H}(-x-3t+10) - 0.8167\mathscr{H}(x+3t-10). \end{aligned} \tag{6.37}$$

Figure 6.6 shows the corresponding numerical and analytical solutions, respectively, for the water depth $h(x,t)$ (left panel) and the velocity $u(x,t)$ (right panel) at time $t = 1$. The numerical solutions are obtained with the DG order $k = 2$ and the grid number $N = 80$. Comparison of the numerical results with the analytical solutions indicates that the DG method together with the limiter can yield fairly accurate

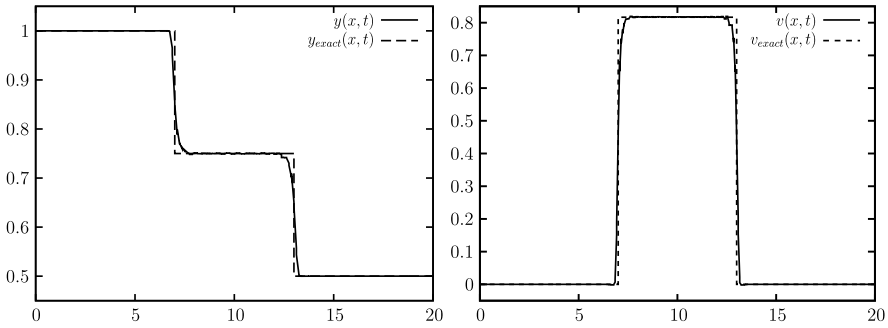


Fig. 6.6 Numerical- and analytical solutions of the water depth h (*left panel*) and the velocity v for the linearized shallow water equations

numerical results for discontinuous solutions. Visible smearings exist only near the discontinuities due to the limiting.

The behavior of the global rms error for variable h is displayed in Fig. 6.7 as a function of the DG order and the grid number with the discontinuous initial conditions (6.36) for the linearized shallow water equations at time $t = 1.5$. It can be seen that the errors decrease with increasing grid number not only for the DG scheme with the limiting in the cell reconstruction but also for the cases without limiting. In general, the numerical error with limiting is substantially smaller than that without limiting. The high-order scheme (e.g. $k \geq 3$ for the present case) will cause the occurrence of instability, if no limiting is used. Therefore, the numerical simulations for the DG scheme without limiting are performed only up to the second order. Furthermore, the relation between the error and the DG approximation order, $E \sim N^{-p}$, is valid with an obviously smaller value for p than the DG order k when a limiting must be employed. For sufficiently large value of k, the accuracy E is less dependent on k. The reason for this is that, in order to suppress numerical oscillations, the necessary limiting eliminates all higher-order effects. For problems with discontinuous solutions, simply going to high-order schemes does not necessarily produce a proportionate increase in accuracy.

6.8.3 Dam Breaking in a Rectangular Channel

In the last two subsections we investigated the linearized shallow water problems with smooth or discontinuous solutions in order to compare the numerical results with the known analytical solutions. In this subsection we investigate the performance of the DG method on the non-linear shallow water equations (6.3) for a horizontal channel of uniform rectangular cross section under negligible friction. As initial conditions, two water bodies with two uniform water levels (h_L and h_R with $h_L > h_R$), both at rest, are firstly separated by a flap at position $x = x_0$. When the flap is instantaneously removed at $t = 0$ (dam breaking), the so-called rarefaction

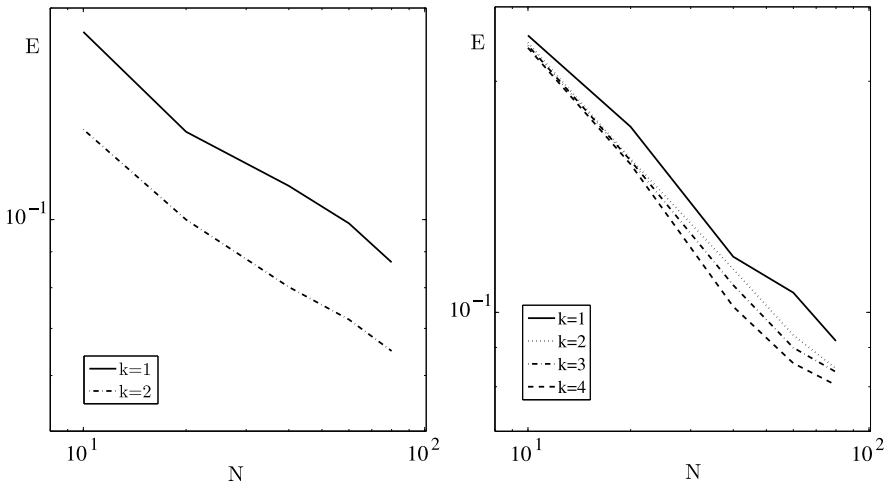


Fig. 6.7 Global rms error for variable h as a function of order of the DG approximation and number of element with the discontinuous initial conditions, without (*left panel*) and with limiting (*right panel*), respectively

wave and shock wave appear, which travel, respectively, to the left into the deep water region and the right into the shallow portion of the fluid. The right-facing shock wave raises the water depth abruptly, whilst the left-facing rarefaction wave reduces the height of the water level.

We assume that the wave phenomenon is correctly governed by the shallow water equations. The flow states after dam breaking are numerically simulated for a channel length $L = 100$ m, the flap position $x_0 = 40$ m, the initial water levels $h_L = 0.6$ m, $h_R = 0.2$ m. The numerical results for water depth and volumetric flow rate are displayed in Fig. 6.8, respectively for three different time points after dam breaking. The third-order DG scheme is employed with limiting where it is necessary. By means of the limiting to the third-order DG scheme the wave traveling can be well modeled without emergence of numerical oscillations.

To compare with the known results in [13], a further simulation with the similar conditions is performed. The channel length is taken as $L = 2000$ m, the dam locates at $x_0 = 1000$ m, as well as the initial water levels are assumed $h_L = 20$ m, $h_R = 5$ m. The second-order DG numerical results of the water depth and the volumetric flow rate at time $t = 50$ s are compared in Fig. 6.9, and a good agreement is demonstrated.

A further test example for the non-linear shallow water equations is based on an initial jump of the flow velocity instead of the jump on the water depth. The initial conditions are assumed as

$$h(x,0) = 0.4, \qquad u(x,0) = \begin{cases} 0.3, & x < 40, \\ -0.3, & x \geq 40. \end{cases} \tag{6.38}$$

The numerical results for $h(x,t)$ and $Q(x,t)$ at a given time point are shown in Fig. 6.10. For this case two shock waves are formed which travel from the initial

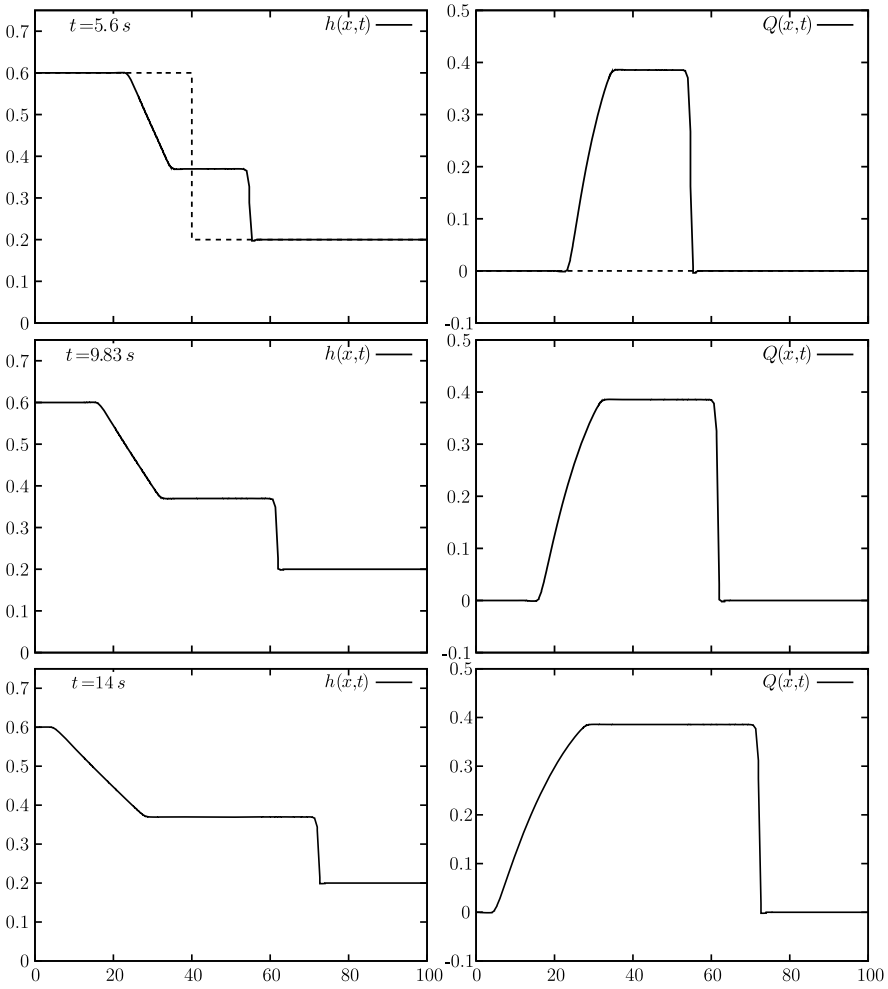


Fig. 6.8 Numerical solution of the dam-break problem using the third-order DG scheme with limiting, shown for the water depth (*left panels*) and the volumetric flow rate (*right panels*) at three time points, respectively. The *dashed lines* indicate the initial conditions

jump point in opposite directions. This effect is fairly similar to the traffic approaching a red traffic light.

As pointed out before, there exist various formulations for the numerical flux. Here we will compare the performances of two different formulations, respectively, postulated by Harten Lax and van Leer (HLL) introduced in Sect. 6.6 and employed in all other examples and Lax-Friedrichs (LF). The Lax-Friedrichs numerical flux is defined (see e.g. [5]) by

$$F_{LF}(U_L, U_R) = \frac{1}{2}\big[F_L + F_R - C(U_R - U_L)\big],$$

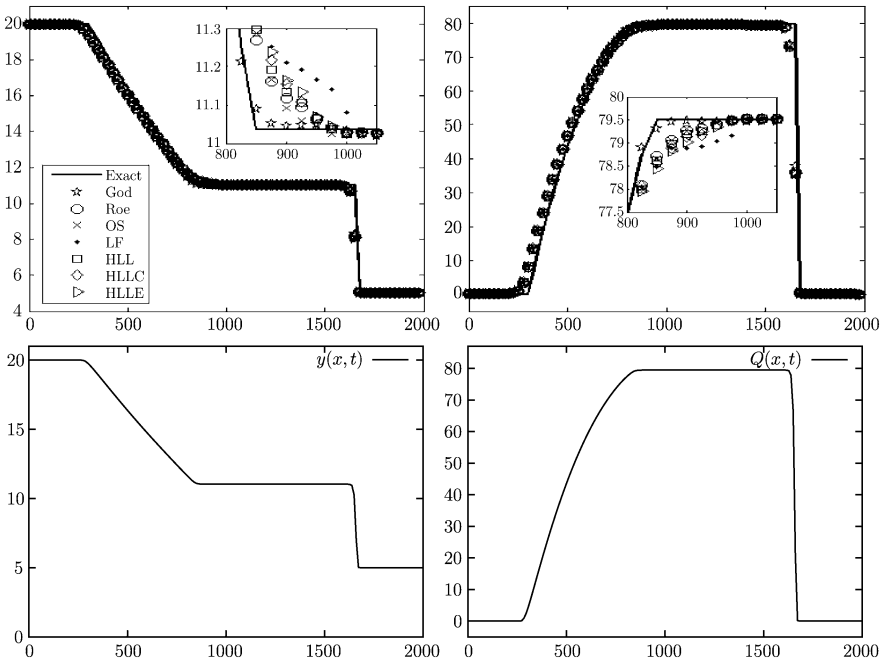


Fig. 6.9 Comparison of numerical results of the second-order DG scheme (*below panels*) with those in [13] (*above panels*), respectively, for the water depth (*left panels*) and the volumetric flow rate (*right panels*). The *dashed lines* indicate the initial conditions

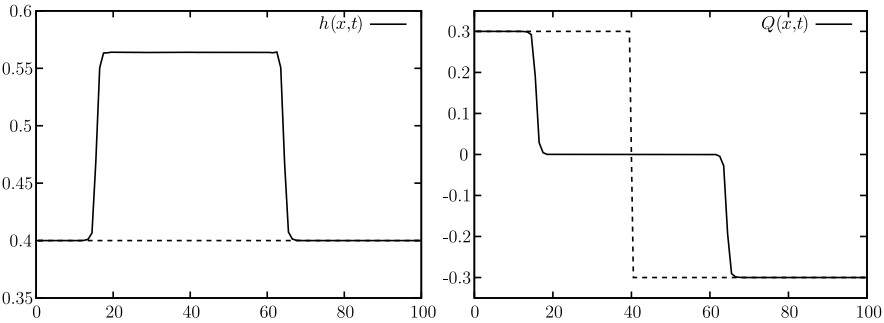


Fig. 6.10 Shock waves for an initial jump in the velocity. The *dashed lines* indicate the initial conditions and the *solid lines* for the states at $t = 12.5$ s

where

$$C = \max\left(|v_L - \sqrt{gh_L}|, |v_R - \sqrt{gh_R}|, |v_L + \sqrt{gh_L}|, |v_R + \sqrt{gh_R}|\right).$$

For this purpose the dam-breaking problem as presented in Fig. 6.8 is investigated but with a much worse resolution. The numerical results of the third-order DG

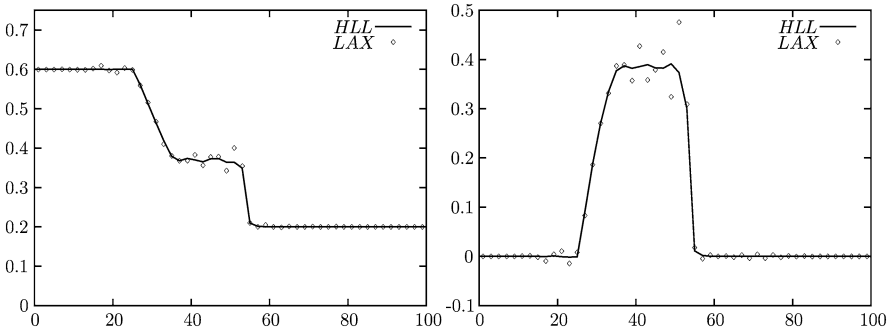


Fig. 6.11 Numerical performances of the HLL and Lax-Friedrichs fluxes for a dam-breaking problem, the left is height of water and the right is the flow rate at $t = 5.7$ s

scheme are shown in Fig. 6.11. It can be observed that the LF flux yields obvious oscillations. If a sufficiently fine resolution is employed, the difference of the both numerical fluxes will becomes negligible.

6.8.4 Channel Flow with the Moving Transition Separating Free-Surface and Pressurized Flow Regions

A main goal of this research is to investigate channel flows including the transition between the free-surface flow and the pressurized flow. Based on the mathematical model postulated in Sect. 6.4, two flow cases with the transition are numerically investigated.

The first test example is the flow in an inclined channel with the inclination angle of $\alpha = 4°$, the length of $L = 100$ m and the initial water depth of 0.1 m at the left end. The right end of the channel is closed. In a region near the right end the maximum height of the channel is confined to $H = 0.2$ m. The water is initially still and the water surface is horizontal. From $t = 0$, a steady volumetric flow rate of $Q(x = 0, t) = 0.1\ \mathrm{m}^{-3}\,\mathrm{s}^{-1}$ is imposed at the left end of the channel. Numerical results of the water depth are displayed in Fig. 6.12, respectively for different sequent time points. It can be observed that when the height jump reaches the right end of the channel, a transition interface separating the free-surface flow from the fully filled flow is formed. This transition interface moves back towards the left due to the incessantly imposed flow rate from the left. The movement of the transition can be well captured.

As the other test case, a horizontal channel of $L = 100$ m with both closed ends is initially separated by a flap at $x = 40$ m into two parts with different water depths $h_L = 0.6$ m and $h_R = 0.2$ m. The channel has a height of $H = 1$ m for $x \leq 60$ m and $H = 0.45$ m for $x > 60$ m, as displayed in Fig. 6.13. The flap is removed instantaneously at time $t = 0$. The distribution of the water depth along the channel is depicted in Fig. 6.13 for different sequent time points. When the shock wave

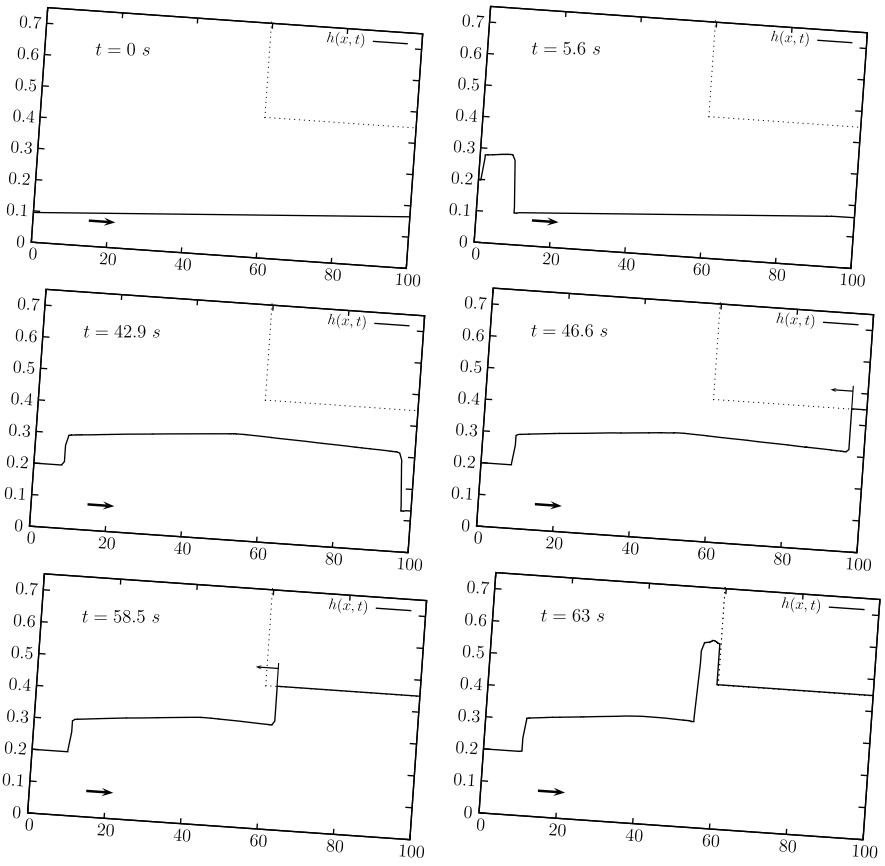


Fig. 6.12 Distribution of the water depth at different time points for a steady flow rate at the left end of the channel. Numerical simulation is performed with the third-oder DG scheme. The *arrow* denotes the motion direction of the transition point at time $t = 46.68$ s and $t = 58.5$ s

reaches the right end, due to the blockage of the wall a transition interface is formed which moves towards the left in the upstream direction. When the transition point is moving back to $x = 60$ m, a large height jump occurs as a result of the canceled limitation of the channel height. This shock wave travels further towards the left. Due to the newly formed right-facing traveling rarefaction wave, a new transition interface is formed which moves towards to the right in the downstream direction.

6.9 Concluding Remarks

In this paper, the shallow water channel flows have been investigated numerically by the Discontinuous Galerkin method. To restrict spurious numerical oscillations

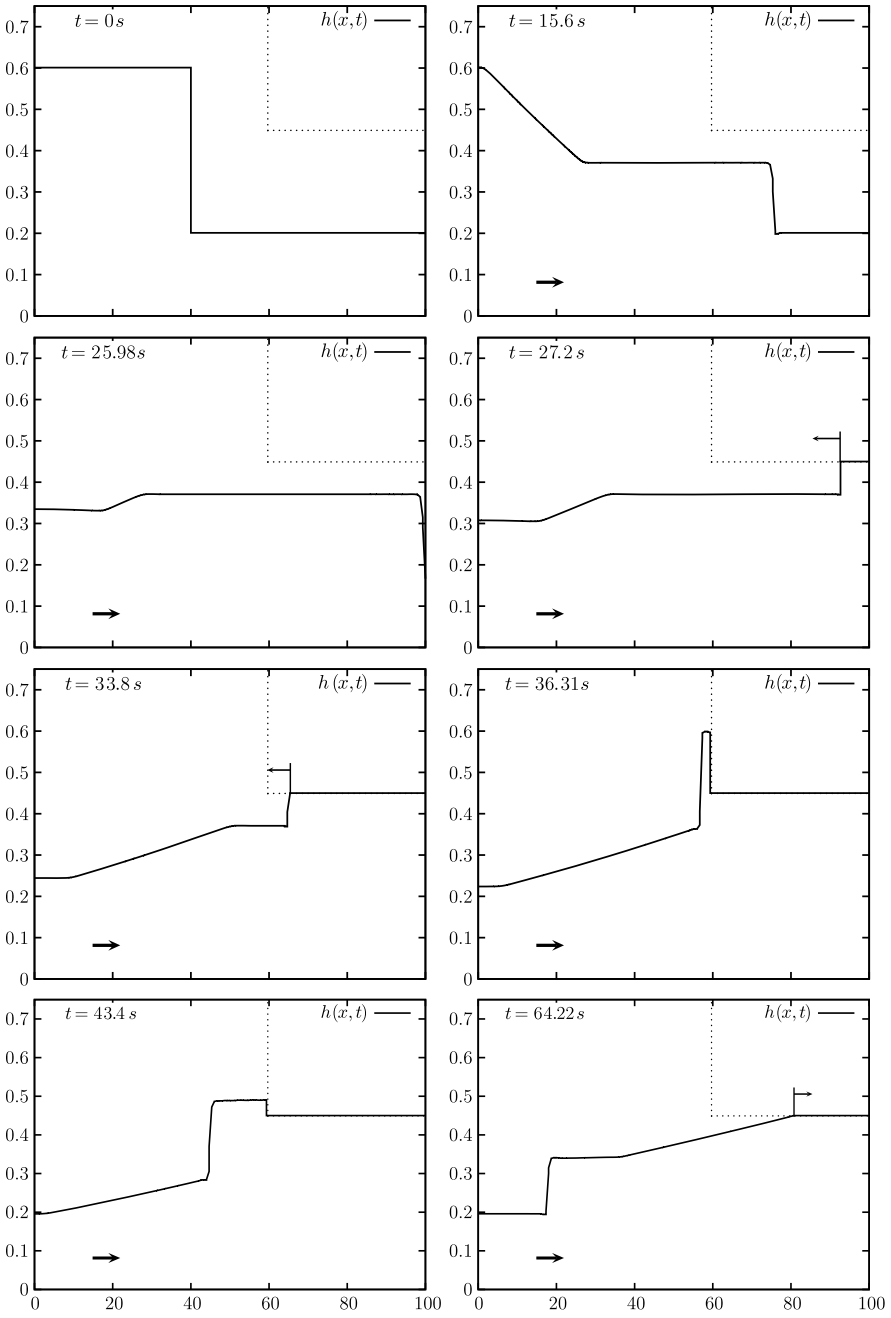


Fig. 6.13 Distribution of the water depth for different times after the flap separating two different water depths is removed. The *arrow* denotes the motion direction of the transition point at time $t = 27.2$ s, $t = 33.8$ s and $t = 64.22$ s

emerging in the high-order DG schemes for the cases of discontinuity or large gradients, a so-called discontinuity detector is used to detect the troubled regions, in which the limiting was employed to high-order DG coefficients. In this way, the high-order DG scheme is maintained in smooth regions, and in discontinuities the accuracy as high as possible is retained. For the emergence of a transition interface between the free surface flow and pressurized flow, the movement of the transition was theoretically and numerically investigated. Numerical results have demonstrated a good performance of the DG code and the modeling of the transition.

References

1. S. Adjerid, K.D. Devine, J.E. Flaherty, L. Krivodonova, A posteriori error estimation for discontinuous Galerkin solution of hyperbolic problems. Comput. Methods Appl. Mech. Eng. **191**, 1097–1112 (2002)
2. C. Bourdarias, S. Gerbi, A finite volume scheme for a model coupling free surface and pressurised flows in pipes. J. Comput. Appl. Math. **209**, 109–131 (2007)
3. C.E. Castro, E.F. Toro, A Riemann solver and upwind methods for a twophase flow model in non conservative form. Int. J. Numer. Methods Fluids **50**(3), 275–307 (2006)
4. B. Cockburn, S.-Y. Lin, C.-W. Shu, TVB Runge Kutta local projection discontinuous Galerkin finite element method for conservation laws III. J. Comput. Phys. **84**, 90–113 (1989)
5. B. Cockburn, C.W. Shu, Runge Kutta discontinuous Galerkin methods for convection dominated problems. J. Sci. Comput. **16**(3) (2001)
6. J. Cunge, B. Mazaudou, Mathematical modelling of complex surcharge systems: Difficulties in computation and simulation of physical situations, in *Int. Conf. on Urban Storm Drainage* (1984), pp. 363–374
7. J.A. Cunge, M. Wegner, Application au cas dune galerie tantot en charge. Houille Blanche **1** (1964)
8. Q. Guo, C.C.S. Song, Surging in urban storm drainage systems. J. Hydraul. Eng. **116**, 1523–1537 (1990)
9. M.A. Hamam, J.A. McCorquodale, Transient conditions in the transition from gravity to surcharged sewer flow. Can. J. Civ. Eng. **9**, 189–196 (1982)
10. A. Harten, P.D. Lax, B. van Leer, On upstream differencing and Godunov-type schemes for hyperbolic conservation laws. SIAM Rev. **25**(1), 35–61 (1983)
11. J.S. Hesthaven, T. Warburton, *Nodal Discontinuous Galerkin Methods: Algorithms, Analysis, and Applications* (Springer, New York, 2000)
12. Z. Ji, General hydrodynamic model for sewer channel networks systems. J. Hydraul. Eng. **124**(3), 307–315 (1998)
13. G. Kesserwani, R. Ghostine, J. Vazquez, A. Ghenaim, R. Mose, Riemann solvers with Runge Kutta discontinuous Galerkin schemes for the 1d shallow water equations. J. Hydraul. Eng. **134**(2), 243–255 (2008)
14. L. Krivodonova, J. Xin, J.F. Remacle, N. Chevaugeon, J.E. Flaherty, Shock detection and limiting with discontinuous Galerkin methods for hyperbolic conservation laws. Appl. Numer. Math. **48**, 323–338 (2004)
15. A. Leon, Improved Modeling of Unsteady Free Surface Pressurized and Mixed Flows in Storm Sewer Systems. Phd thesis, University of Illinois at Urbana Champaign, 2007
16. R.J. Leveque, *Finite Volume Methods for Hyperbolic Problems*. (Cambridge University Press, Cambridge, 2004
17. J. Li, A. McCorquodale, Modeling mixed flow in storm sewers. J. Hydraul. Eng. **125**(11), 1170–1180 (1999)

18. D. Schwanenberg, R. Liem, J. Köngeter, Runge Kutta discontinuous Galerkin methods for convection dominated problems. IWW, Aachen, Germany **16**(3) (2001)
19. C.C.S. Song, J.A. Cardle, K.S. Leung, Transient mixed flow models for storm sewers. J. Hydraul. Eng. **179**, 1487–1504 (1983)
20. V.L. Streeter, E.B. Wylie, Water hammer and surge control. J. Hydraul. Div. **95**, 79–101 (1973)
21. E.F. Toro, *Shock Capturing Methods for Free Surface Shallow Flows* (Wiley, New York, 2001)
22. E.F. Toro, *Riemann Solvers and Numerical Methods for Fluid Dynamics* (Springer, Berlin, 2009)
23. J.G. Vasconcelos, S.J. Wright, Surges associated with air expulsion in near horizontal pipelines, in *ASME JSME Joint Fluids Engrg. Conference*, Honolulu, HW (2003), pp. 2897–2905
24. J.G. Vasconcelos, S.J. Wright, *Numerical Modeling of the Transition Between Free Surface and Pressurized Flow in Storm Sewers* (CHI Publications, Canada, 2004)
25. D.C. Wiggert, Transient flow in free surface, pressurized systems. J. Hydraul. Div. **98**(1), 11–27 (1972)
26. F. Zhou, F.E. Hicks, P.M. Steffler, Transient flow in a rapidly filling horizontal pipe containing trapped air. J. Hydraul. Eng. **128**, 625–634 (2002)

S. Moradi Ajam · Y. Wang · M. Oberlack
Strömungsdynamik, Technische Universität Darmstadt, Petersenstr. 30, 64287 Darmstadt, Germany

S. Moradi Ajam
e-mail: moradi@fdy.tu-darmstadt.de

Y. Wang
e-mail: wang@fdy.tu-darmstadt.de

M. Oberlack (✉)
e-mail: oberlack@fdy.tu-darmstadt.de

Chapter 7
Optimal Control of Sewer Networks Engineers View

Steffen Heusch and Manfred Ostrowski

Abstract This chapter introduces a software tool for MPC of sewer networks with a dynamic process model which is based on an interative approach. A flexible optimizer, which implements local and global optimization methods, is connected to a dynamic sewer network model to evaluate the objective function values. Numerical results for a simple urban drainage network are presented, illustrating the functionality of the approach.

7.1 Prerequisites

Dynamic modeling of flows in sewer networks is state of the art in urban drainage planning for the evaluation of the hydraulic functionality of the network. Network data is usually available in a high resolution giving information for every junction and every conduit. Traditional objectives of these computations are the verification that design rainfalls can be drained by the network without flooding, but with increasing computational power long-term simulations with real rainfall data are getting more popular as well leading to a better understanding of flow processes in the sewer system.

The application of optimization methods is common practice in urban drainage management. With regard to sewer networks, frequent tasks are the design of new storm sewer networks, the scheduling of sewer rehabilitation strategies or the optimization of flow rates of devices such as pumps or orifices. Since optimization tasks require long-term simulations in most cases, the development of faster process models would certainly be beneficial for the engineers daily work. However, the development of up-to-date software with latest mathematical methods is not the engineers duty and therefore not a top priority from their point of view. Because many modeling datasets for hydraulic models from various projects already exist, it is rather desirable to build software modules for optimization, which utilize existing software for process modeling. The usage of approved software makes it easier to convince operators to use such methods and gives a higher credibility to projects. Therefore software development concentrates in many cases on the optimization module which is engineered in such a way that it can cope with the given conditions from the already existing process model.

A. Martin et al. (eds.), *Mathematical Optimization of Water Networks*,
International Series of Numerical Mathematics 162,
DOI 10.1007/978-3-0348-0436-3_7,

In regard to the optimization, experience shows that no single method is suitable for all problems. Thus, all kind of optimization methods can be found (an overview of applied methods for RTC in general is given in Chap. 5), and usually the constraints of the problem at hand determine the methods which are feasible for application. For MPC applications, however, a number of authors (e.g. [1] and [5]) state, that global search algorithms are required in order to assure that an optimal control setting is found, and that local search algorithms are therefore not sufficient. No proof for this assumption has been given until now.

The first step of this work is the development of a software tool for MPC which is based on an existing process model for dynamic flow routing in sewer networks and which enables the usage of different optimization algorithms. Based on this, the general performance of dynamic flow routing approaches for MPC are analyzed and the differences between local and global optimization algorithms are investigated.

7.2 Modeling Approach

The developed application uses SWMM5 [7] as process model and BlueM.Opt for optimization. Both software modules run separately and communicate by text files. Usage of text files for data input and results is a common feature in urban drainage modeling requiring appropriate interfaces for information exchange of optimization parameters (input data generated by the optimizer) and objective values (result data generated by the process model). Due to this separation this approach can be characterized as simulation based parameter optimization. The optimization is not based on numerical derivations of the dynamic flow equations and is therefore a derivative-free method (black-box-optimization).

For MPC, an additional software is required which controls the receding horizon process and supplies informations for the data flow. Setup and workflow of the whole software package is illustrated in Fig. 7.1. Information for the control time step and for the durations of the three horizons (Fig. 5.3) are given in the MPC module which starts the application. Once the process has been started the same workflow applies to every control step: The optimization module generates datasets which are simulated by the hydraulic process model. Simulation results from the process model are returned to the optimizer that evaluates them in regard to the defined objective function and consequently generates a new dataset to continue with the optimization process.

7.2.1 MPC Controller

The software developed for the control of the MPC process is called BlueM.MPC since it makes use of the optimizer BlueM.Opt. It is programmed in C-Sharp and runs on Windows requiring merely the .NET Framework 2.0 which is freely available.

BlueM.MPC controls the work flow of the MPC process and is basically a data manager. Application of the software is simple since the user interface only consists

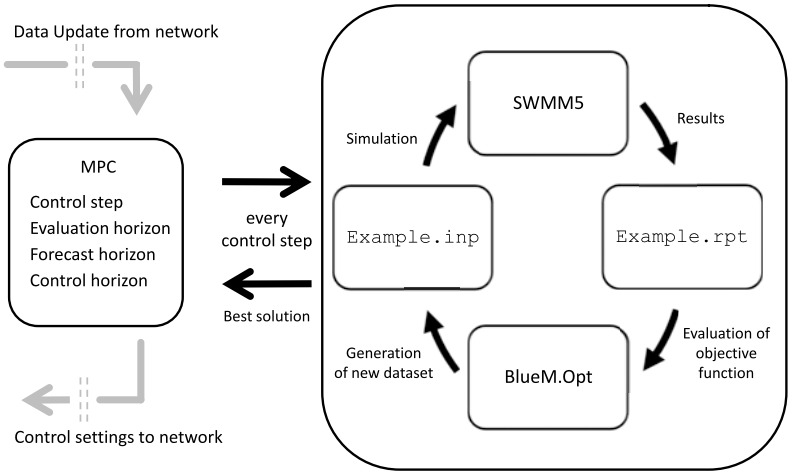
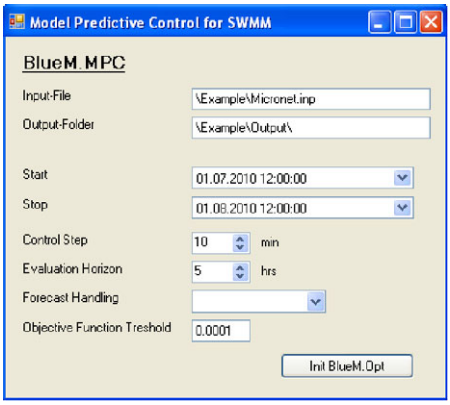


Fig. 7.1 Setup of the MPC software

Fig. 7.2 User form for BlueM.MPC



of one form with mostly preformatted textboxes which the user is required to fill in (Fig. 7.2). After setting all required information the application proceeds with initializing the optimizer. The user form for BlueM.Opt opens (Fig. 7.3) and demands informations for the optimization algorithm. MPC simulations start after pressing the Start/Continue-Button.

In addition to the informations of the two user forms for BlueM.MPC and BlueM.Opt, the whole application requires only three text files:

- SWMM input file describing the process model.
- Input file for the optimizer BlueM.Opt comprising informations for the optimization parameters (boundaries for the control variables and formatting instructions) and the objective function.

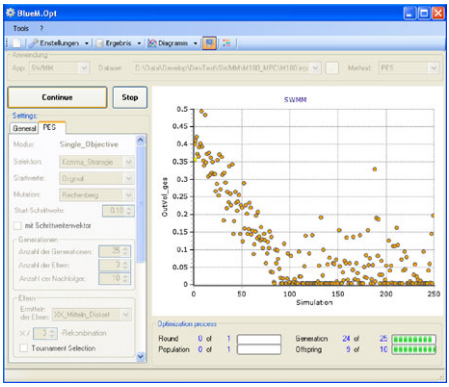


Fig. 7.3 User form for BlueM.Opt

- Input file for BlueM.MPC comprising informations on the forecasted inflow. Each node in the hydraulic model can be assigned to a forecast horizon. Additionally, uncertainties in the inflows can be considered in order to simulate less reliable future inflow forecasts.

The following informations are required for the user form of BlueM.MPC:

Inputfile The full path and name of the SWMM input file that stores the process model data.

Output folder The path of the folder in which all documentation and result files are copied in.

Start Starting date and time of the MPC simulation.

Stop Ending date and time of the MPC simulation.

Control step The time span for the control step (see Fig. 5.3) in minutes.

Evaluation horizon The time span of the evaluation horizon (see Fig. 5.3) in hours.

Forecast handling The method which is applied to determine the inflow values for the time difference between the evaluation horizon and the forecast horizon (see Fig. 5.3). The user can choose between three options: Set all values to zero, use average values from the forecast horizon or use the last value of the forecast horizon.

Objective function threshold This option can be applied to avoid extensive calculation times in cases where the objective function is almost minimal. A threshold value can be set, which is compared to the calculated value of the objective function of the first simulation run. If the objective function value is smaller than the threshold value, iterations for this control step will be terminated.

7.2.2 Process Model

SWMM 5 [7] is used as process model within BlueM.MPC. It is a dynamic rainfall-runoff simulation model which can be used for single event or long-term continuous simulation of runoff quantity and quality from primarily urban areas. SWMM 5 is

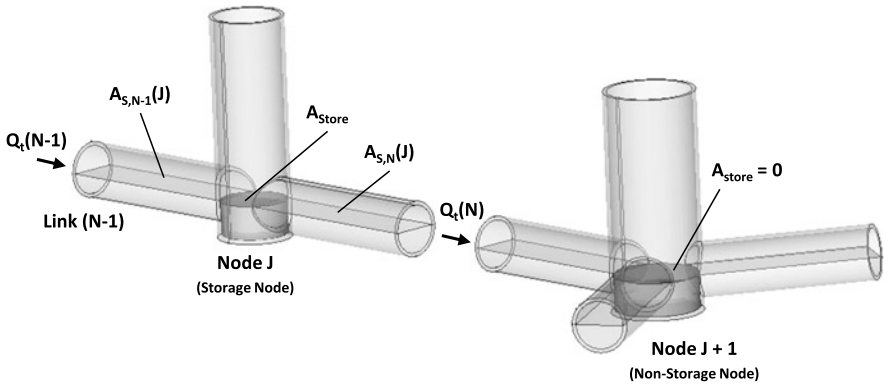


Fig. 7.4 Node-junction-representation in SWMM 5 ([6], modified)

an open source software developed and funded by US-EPA. It is used worldwide and approved in practice. Straightforward contact to the developers as well as a huge community running an active mailing-list ensure a sustainable development of the software. Currently the update version SWMM 5.0.018 is implemented in BlueM.MPC. Implementation of a new version is possible without changes as long as the SWMM 5 input and result file have the same format. Changes of the source code of SWMM 5 were not made. SWMM 5 complies with the requirements for receding horizon applications: Storage and allocation of system states is possible as well as the application of time dependent settings for flow control structures.

SWMM 5 solves the *Shallow Water Equations* using an iterative multi-step finite difference scheme based on a node-link representation as shown in Fig. 7.4. The following paragraphs are based on the explanations in [6].

The continuity equation

$$\partial_t A + \partial_x Q = 0 \tag{7.1}$$

is applied at nodes of the system (e.g., manholes, storage nodes) and the momentum equation

$$\partial_t Q + \partial_x \left(Q^2/A\right) + gA\partial_x H + gAS_f + gAh_L = 0 \tag{7.2}$$

is applied along links (e.g., pipes, channels). For the calculations an additional continuity relationship for the nodes that connects two or more conduits is required, which is given by

$$\partial_t H = \frac{\sum Q}{A_{store} + \sum A_s} \tag{7.3}$$

where A_{store} is the surface area of the node itself, $\sum A_s$ is the surface area contributed by the conduits connected to the node, and $\sum Q$ is the net flow into the

node contributed by all conduits. (7.1), (7.2) and (7.3) are converted into an explicit set of finite difference formulas leading to

$$Q_{t+\Delta t} = \frac{Q_t + g\bar{A}(H_1 - H_2)\Delta t/L + 2\bar{V}(\bar{A} - A_t) + \bar{V}^2(A_2 - A_1)\Delta t/L}{1 + \frac{gn^2|\bar{V}|\Delta t}{k^2\bar{R}^{4/3}} + \frac{\sum_i K_i|V_i|\Delta t}{2L}} \tag{7.4}$$

in which $\bar{A}$, $\bar{R}$ and $\bar{V}$ are average values in the conduit, the index i represents values at location i along the conduit and indices 1 and 2 denote values at the up- and downstream end of the conduit. The equation solved for the water head at each node is:

$$H_{t+\Delta t} = H_t + \frac{\Delta Vol}{(A_{store} + \sum A_s)_{t+\Delta t}} \tag{7.5}$$

where ΔVol is the net volume flowing through the node over the time step.

SWMM 5 solves (7.4) and (7.5) using a method of successive approximations with under relaxation.

When the water level in a node exceeds the crown of the highest conduit connected to it, a node is defined to be in a surcharged condition. Under this condition the surface area contributed by any conduit would be zero and (7.3) would no longer be applicable for non-storage nodes. SWMM 5 solves this problem by using an alternative nodal continuity condition which is expressed in the form of a perturbation equation:

$$\sum(Q + \partial_H Q \Delta H) = 0. \tag{7.6}$$

Solving for ΔH yields:

$$\Delta H = \frac{-\sum Q}{\sum \partial_H Q} \tag{7.7}$$

where $\partial_H Q$ follows from (7.4)

$$\partial_H Q = \frac{-g\bar{A}\Delta t/L}{1 + \frac{gn^2|\bar{V}|\Delta t}{k^2\bar{R}^{4/3}} + \frac{\sum_i K_i|V_i|\Delta t}{2L}}. \tag{7.8}$$

Every time equation (7.7) is applied to update the head of a surcharged node, (7.4) is recalculated to provide flow updates for the conduits connecting to the node until some convergence criterion is met. These surcharge iterations are integrated into the usual set of iterations. That is, whenever heads need to be computed in the successive approximation scheme, (7.7) is used in place of (7.5).

For supercritical flow conditions SWMM 5 limits the flow to be no greater than the normal flow for the current flow depth at the upstream end of the conduit. Boundary conditions at outfalls may be represented as free outfalls (i.e., flowing through critical depth) or a specified stage-time relationship.

Hydraulic structures such as pumps, orifices and weirs are modeled as links that connect a pair of nodes. Flow through these links is computed as a function of the heads at their end nodes. These flows are calculated during the flow evaluation step after the flows through all the conduits are computed.

For pumps, the user is required to define a pump curve which specifies flow as a function of the inlet node volume, inlet node depth, or the head difference between the inlet and outlets nodes.

For orifices, a classical orifice equation is used when the orifice is fully submerged:

$$Q = C_d A \sqrt{2gh}. \tag{7.9}$$

A modified weir equation is used when the orifice is submerged by a fraction f.

$$Q = C_d A \sqrt{2gD} f^{1,5}. \tag{7.10}$$

A denotes the area and D is the height of the full orifice opening, while h is the head across the orifice. The surface area contribution of the orifice to its end nodes is based on the depth of water in the orifice and the equivalent pipe length:

$$L = 2\Delta t \sqrt{gD}. \tag{7.11}$$

For weirs, the following classical equation is used to compute flow as a function of head h across the weir while the weir is not fully submerged:

$$Q = C_w L_w h^n \tag{7.12}$$

where C_w is the discharge coefficient of the weir, L_w is the length of its opening, and n is a exponent that depends on the type of weir (e.g. transverse or side-flow). When the weir becomes completely submerged the orifice equation is used to predict flow as a function of the head across the orifice. Weirs do not contribute any surface area to their end nodes.

7.2.3 Optimization

BlueM.Opt is an optimization software which was developed at the Section of Engineering Hydrology and Water Management (ihwb) at the Technische Universitaet Darmstadt. Based on the experience, that (1) no single model or simulator can fit all applications and (2) no single optimization method is suitable for all problems, it was designed as a generic framework for simulation based optimization that facilitates a problem driven coupling of appropriate models and optimization algorithms. BlueM.Opt integrates various optimization algorithms via strategy pattern [2] and supports simulators that use text based input files. Figure 7.5 cites the

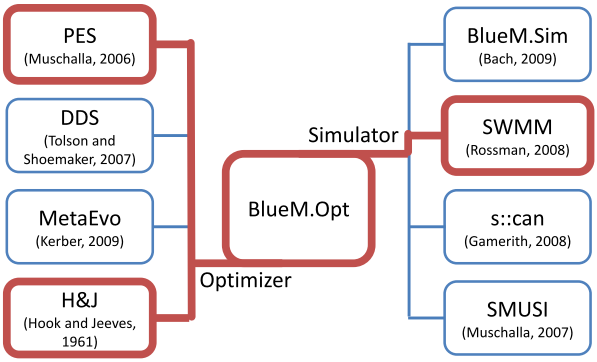


Fig. 7.5 BlueM.Opt

currently available optimization algorithms and simulators for parametric optimizations. BlueM.Opt is freeware and the source code is available for research cooperation (www.bluemodel.org). The software comes with a GUI enabling the user to monitor the optimization progress and to analyze and postprocess the results which is especially useful for multiobjective optimizations with evolutionary algorithms. Within this study, a basic evolutionary strategy algorithm (PES) and a hill climbing method from Hooke and Jeeves were selected for optimization. With these two algorithms one local and one global search algorithm is chosen giving the possibility to analyze the postulated theory, that global algorithms are superior for MPC.

PES Parametric evolutionary strategies (PES) mimic evolutionary principles to search optimum solutions by using the process of recombination, mutation and selection. ES belong to the global optimization procedures which are particular suitable for complex parametric optimization problems. They do not make assumptions on the continuity of the objective function, do not require information on its derivatives and they allow for the consideration of linear and nonlinear constraints. In BlueM.Opt, the method is called PES (parametric evolutionary strategy). It was implemented by Muschalla [4] for multiobjective optimizations of integrated urban wastewater systems. It enables population based optimizations and comprises different methods for recombination, mutation and selection. A flow chart of the algorithm is given in Fig. 7.6.

Hooke and Jeeves The Hooke and Jeeves algorithm [3] is a deterministic local optimization method. It is a direct search method, also called “pattern search”, and can be viewed as a matching part of gradient-based search techniques. The pattern search algorithm is a wide-spread local optimization method because of its excellent convergence characteristics. For application, only an initial step size is required.

The algorithm performs repeatedly two types of search routines: an exploratory search and a pattern search. At each iteration, it first defines a pattern of points by moving each parameter one by one, so as to optimize the current objective function. The entire pattern of points is then shifted to a new location. This new location

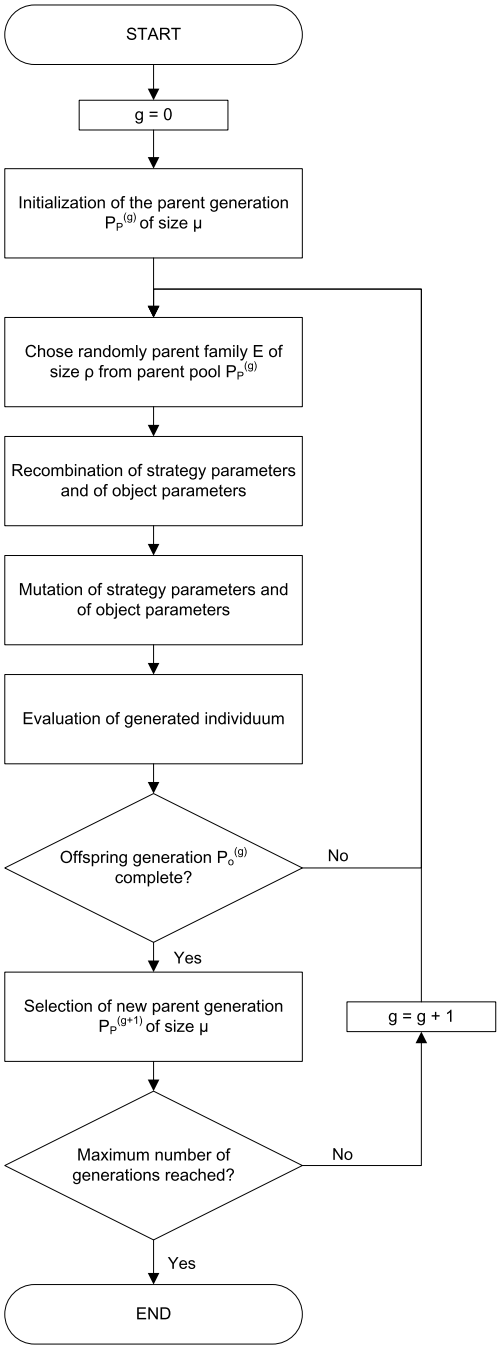


Fig. 7.6 Flow chart of ES strategy

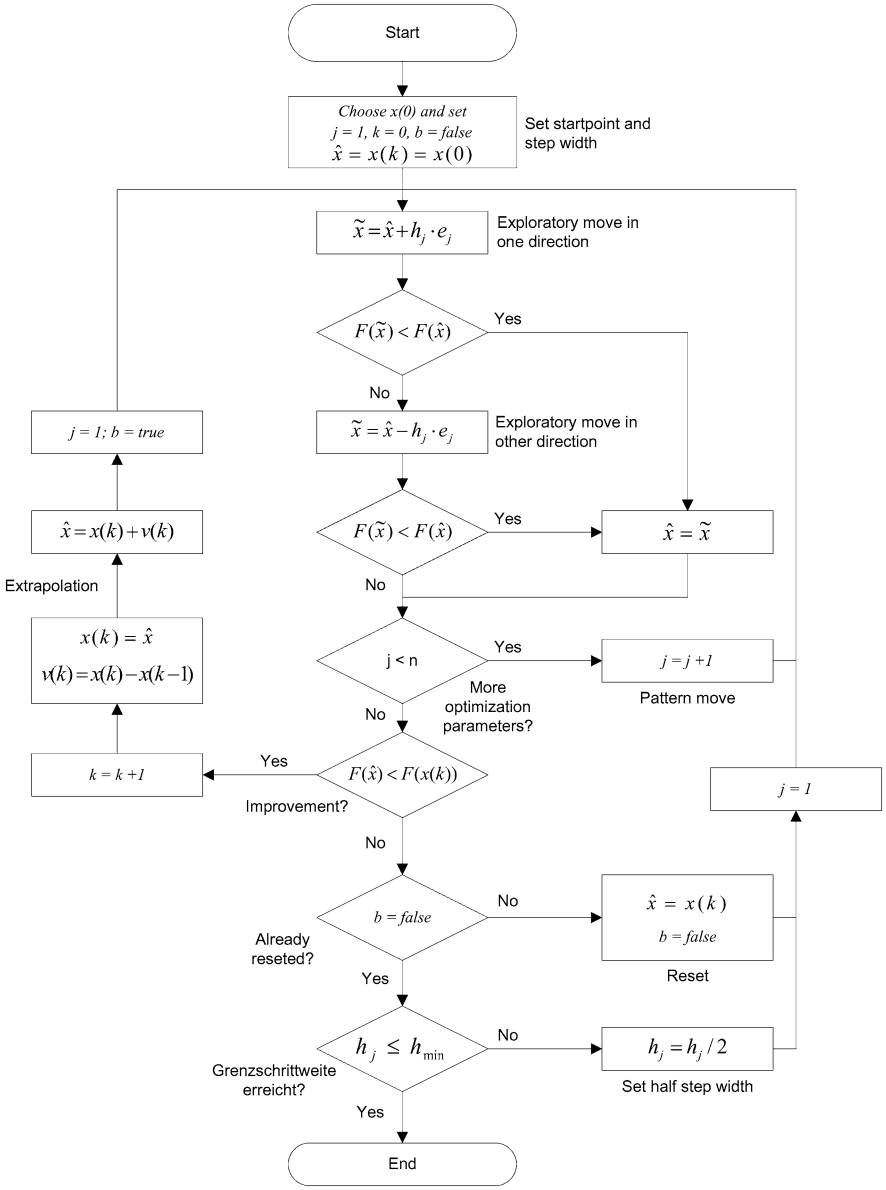


Fig. 7.7 Flow chart of Hooke and Jeeves algorithm

is determined by extrapolating the line from the old base point in the parameter space to the new base point. The step sizes in this process are constantly adjusted to decrease towards the respective optimum. A flow chart of the algorithm is given in Fig. 7.7.

7.2.4 Summary

With BlueM.MPC, a free ready-to-use software was developed, which is characterized by the following features: The process model SWMM 5 is a widely-used software, for which many datasets are already available and which is familiar to many engineers. In addition, the optimization module BlueM.Opt enables the application of different optimization algorithms making it possible to analyze and compare their performance in regard to the specific MPC application.

BlueM.MPC was developed for simulation studies and uses already supplied inflow data, i.e. rainfall-runoff-computations are not part of the process model. With its low demands for application it is an excellent tool for the analysis of the control potential of a sewer network.

Based on its iterative optimization approach, the ultimate questions for MPC applications are: How many simulation runs are possible within one control step and how many simulation runs are necessary within one control step to end up with satisfactory control decisions? The number of possible simulation runs within one control step is depending on those factors which influence the duration of a single simulation run:

1. The size of the sewer network, i.e. the number of nodes and links.
2. The routing time step of the dynamic calculations.
3. The length of the evaluation horizon which determines the time span of a single simulation run.

The number of necessary simulation runs depends on the following objectives:

1. The number of optimization variables, which is depending on the number of controllers in the network.
2. The length of the control horizon and its discretization.
3. The optimization algorithm and its capability to find control settings of sufficient quality within the given time.

7.3 Numerical Results

A simple academic network was selected to test general functionalities of the software. The system is taken from a demonstration software for real time control which is distributed by the German Association for Water, Wastewater and Waste (DWA) [8]. It contains two subcatchments and two storage basins (Fig. 7.8). Outflow from the upstream storage basin (B01) is routed to the lower storage basin (B02) from where it is pumped to the treatment plant. Basin B01 has a volume of 1.600 m^3, B02 of 1.800 m^3. Outflow from B02 to the treatment plant is constant at 120 l/s.

Both subcatchments (S01 and S02) have an impervious area of 60 ha. Catchment characteristics for the calculation of runoff generation are identical. For the calculation of COD-concentrations different coefficients for buildup and washoff were

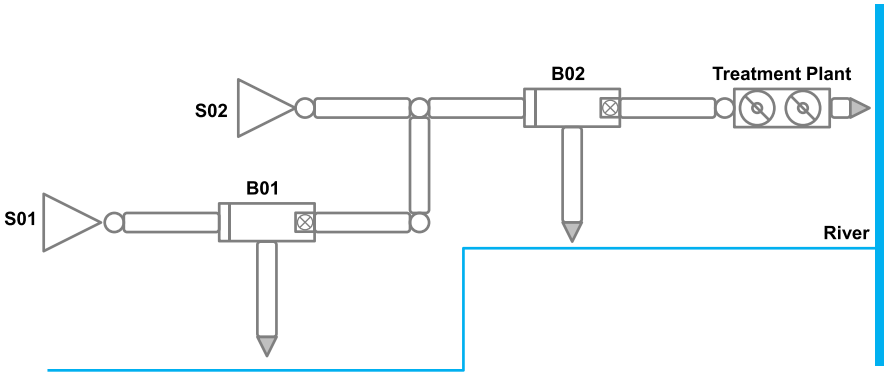


Fig. 7.8 Network of case study 1

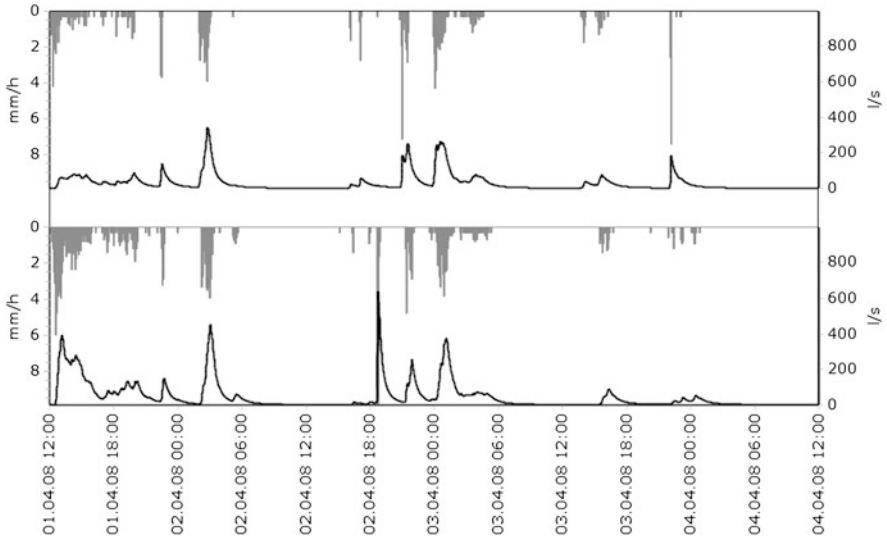


Fig. 7.9 Time series of rainfall and runoff from both subcatchments (*top*: subcatchment S02, *bottom*: subcatchment S01)

applied representing different land use scenarios in the subcatchments. In order to generate uneven flow conditions with a certain degree of control potential in such a small network different rainfall loadings were used. For control simulations a period of three days with rainfall heights of 13.5 mm (S01) and 21.4 mm (S02) was selected. Figure 7.9 shows time series for rainfall and runoff from both subcatchments.

Definition of the objective function for MPC calculations includes minimization of the overflows from both storage basins. Additional penalty functions were applied

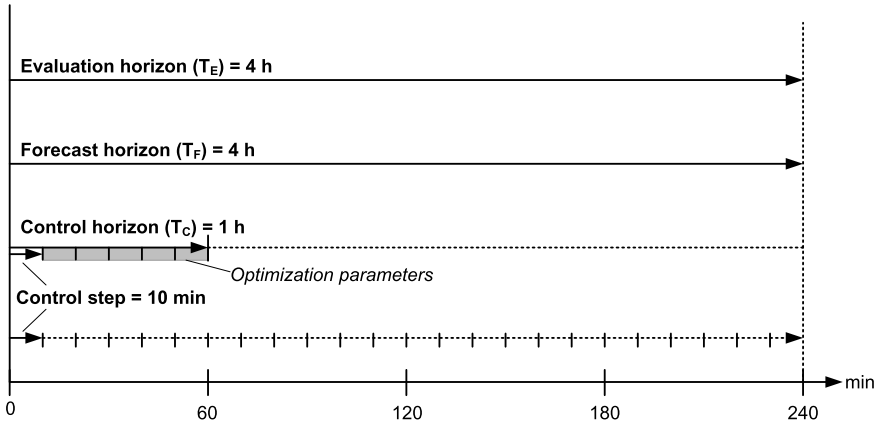


Fig. 7.10 Time horizons applied for the academical network

Table 7.1 Outflow volumes for simulation of academical network

Setting	Reference system	MPC-HJ	MPC-PES
Outflow B01	231 m^3	7 m^3	60 m^3
Outflow B02	1.127 m^3	835 m^3	783 m^3
Total outflow	1.358 m^3	842 m^3	843 m^3

to enforce the emptying of B01. Control of the system is induced by variation of the pump flow from B01 which can vary between a minimum outflow of 0 l/s and a maximum outflow of 90 l/s. Ideal conditions were assumed for the simulations. For the receding horizon strategy a prediction horizon of 4 hours was applied and it was assumed that inflow predictions are perfect. The control horizon was set to 1 hour with a control step of 10 minutes. With these assumptions the problem consists of five optimization variables during each control step. An overview of the applied time horizons is given in Fig. 7.10.

In order to evaluate the performance of the MPC simulations, results for the reference system are required in which no active control is applied and the design outflow of B01 is 45 l/s. For the selected three day simulation period the total overflow volume is 1.358 m^3 with B01 discharging 231 m^3 and B02 discharging 1.127 m^3 into the receiving water body. Figure 7.11 shows water levels, pump rates and overflows for B01 and B02. At B01, overflow occurs at two events between 00:00 2008/04/03 and 06:00 2008/04/03. At B02, three overflow events are recorded between 14:00 2008/04/01 and 06:00 2008/04/02.

Results for the MPC simulations are given in Table 7.1, an overview of water levels, pump rates and overflow are given in Figs. 7.12 (Hooke and Jeeves) and 7.13

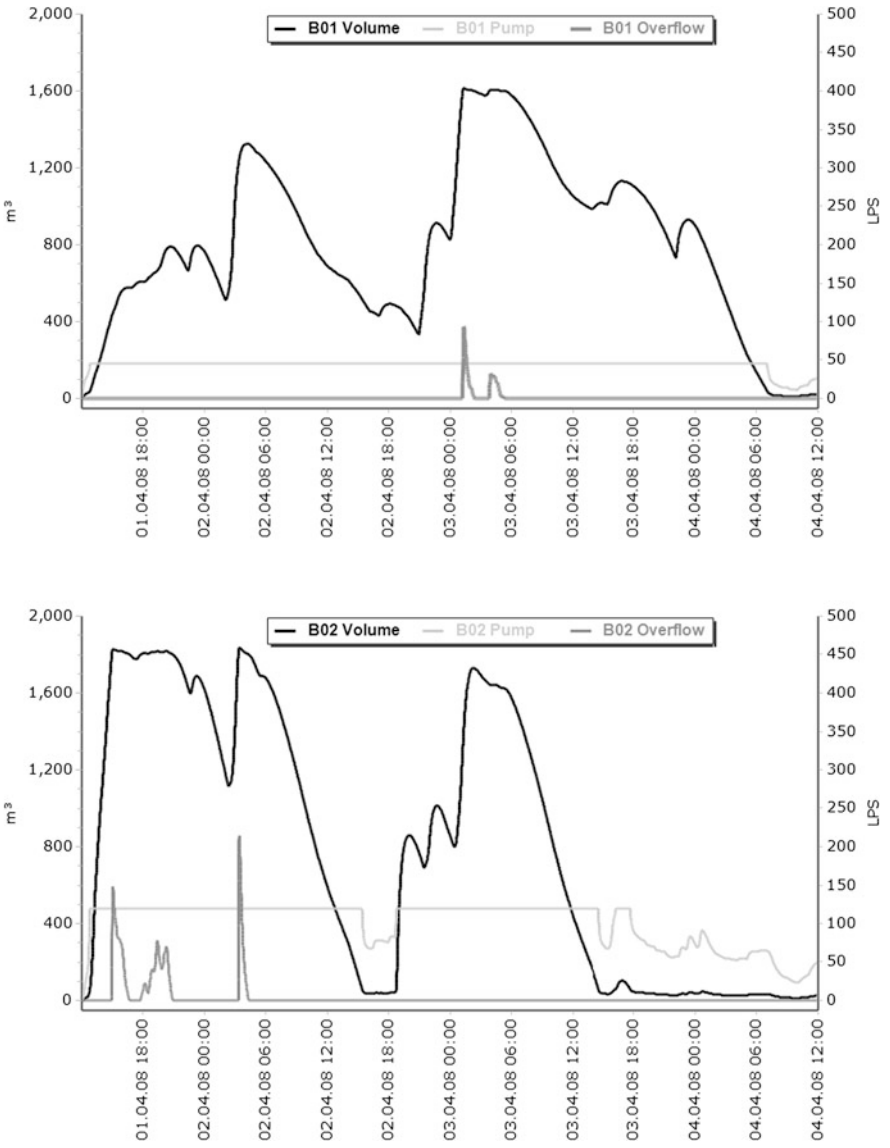


Fig. 7.11 Water levels (*black*), pump flows (*light grey*) and overflows (*dark grey*) of reference system

(PES). The computations demonstrate the general functionality of the MPC system. Simulations with both, local and global optimization algorithm, result in lower overflow volumes compared to the reference system. The differences between local and global algorithms are only marginal though. However, considering the simple structure of the system, this was expected.

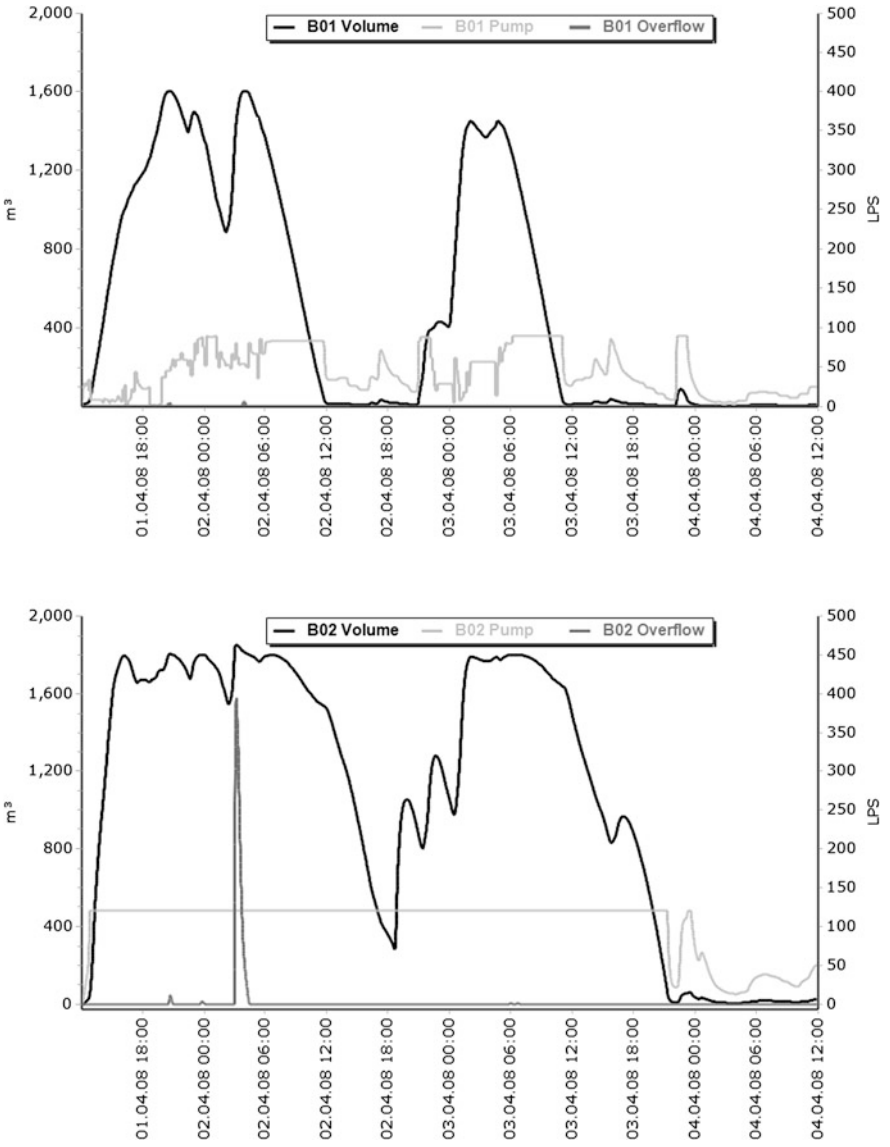


Fig. 7.12 Water levels (*black*), pump flows (*light grey*) and overflows (*dark grey*) of MPC simulation with local search algorithm (Hooke and Jeeves)

Both algorithms manage to avoid the first two overflows at B02, recorded in the reference system between 14:00 2008/04/01 and 22:00 2008/04/01 by decreasing outflow from B01 and using the available storage volume. Furthermore, the overflow recorded at B01 between 00:00 2008/04/03 and 06:00 2008/04/03 is avoided by emptying the basin in advance.

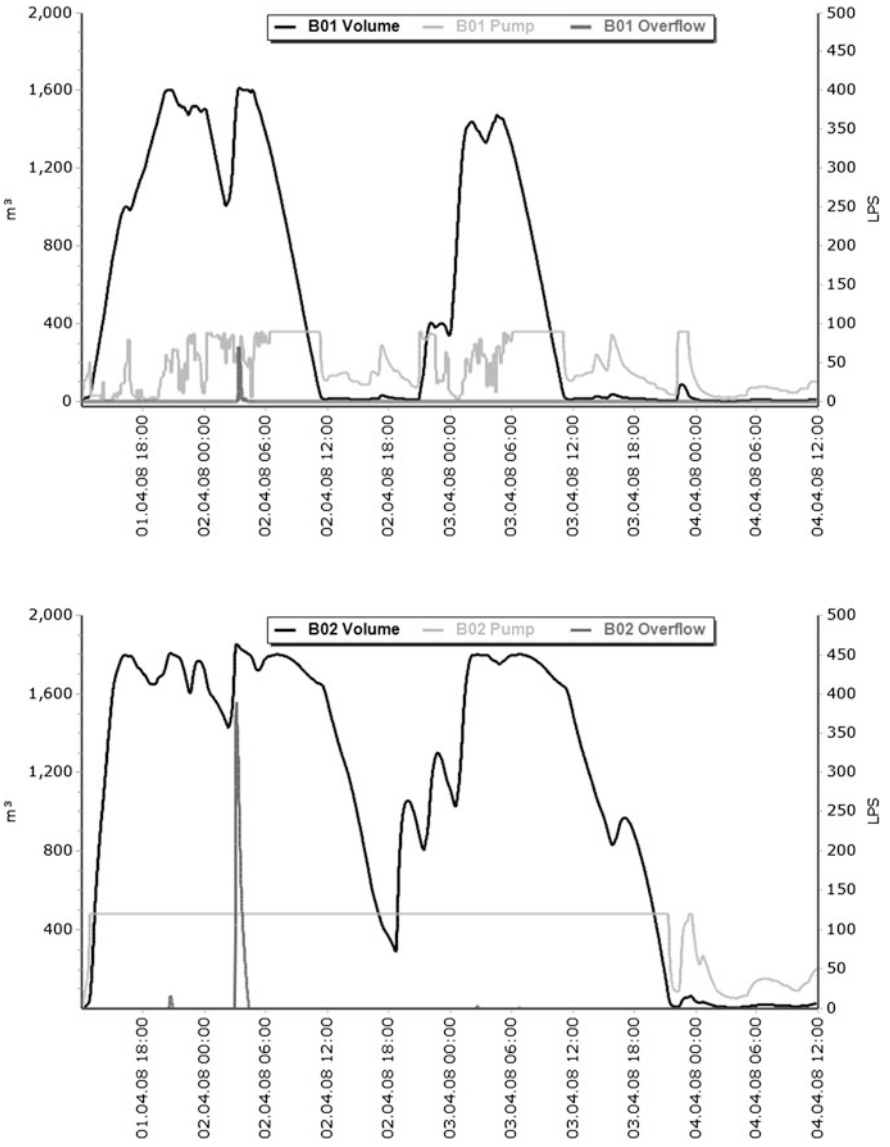


Fig. 7.13 Water levels (*black*), pump flows (*light grey*) and overflows (*dark grey*) of MPC simulation with global search algorithm (PES)

References

1. D. Butler, M. Schutze, Integrating simulation models with a view to optimal control of urban wastewater systems. Environ. Model. Softw. **20**(4), 415–426 (2005)

2. V. Gamerith, D. Muschalla, P. Koenemann, G. Gruber, Pollution load modelling in sewer systems: An approach of combining long term online sensor data with multi-objective auto-calibration schemes. Water Sci. Technol. **59**(1), 73 (2009)
3. R. Hooke, T.A. Jeeves, 'Direct search' solution of numerical and statistical problems. J. ACM **8**(2), 212–229 (1961)
4. D. Muschalla, *Evolutionaere multikriterielle Optimierung komplexer wasserwirtschaftlicher Systeme*, Mitteilungen des Instituts fuer Wasserbau und Wasserwirtschaft der TU Darmstadt, Bd. 137 (Darmstadt, 2006)
5. W. Rauch, P. Harremoes, Genetic algorithms in real time control applied to minimize transient pollution from urban wastewater systems. Water Res. **33**(5), 1265–1277 (1999)
6. L. Rossman, Storm Water Management Model Quality Assurance Report: Dynamic Wave Flow Routing. Technical Report, US EPA, 2006
7. L. Rossman, Storm Water Management Model, Version 5.0. User Manual, US EPA, 2008
8. M. Schuetze, M. Ogurek, A. Messmer, M. Scheer, Kanalnetzdemonstrator, 2007

S. Heusch · M. Ostrowski
Ingenieurhydrologie und Wasserbewirtschaftung, Technische Universität Darmstadt,
Petersenstr. 13, 64287 Darmstadt, Germany

S. Heusch
e-mail: heusch@ihwb.tu-darmstadt.de

M. Ostrowski (✉)
e-mail: ostrowski@ihwb.tu-darmstadt.de

Chapter 8
Real-Time Control of Urban Drainage Systems

Johannes Hild and Günter Leugering

Abstract A hydrodynamic process model based on shallow water equations is discretized on 1D-networks with the method of finite volumes. Based on the finite volumes we replace algebraic coupling conditions by a consistent finite volume junction model. We use discrete adjoint computation for one step Runge-Kutta schemes to generate fast and robust gradients for descent methods. We use the descent methods to generate an optimal control for an example network and discuss the computational results.

8.1 Introduction

Simulation and optimization of shallow water flows on networks is not an easy task: We have to face numerical instabilities resulting from flaws in the modeling process, strong nonlinearity and even discontinuous behavior of the process states at common rates, and the general ill-conditioning of the underlying mathematical problems.

Nevertheless we managed to develop a time-efficient simulation model for shallow water flow on networks, which is not only capable to treat strong velocities and discontinuous shock waves, but also generates sufficient smooth state solutions, which we require for the optimization algorithm.

We start this chapter by introducing the governing system equations in Sect. 8.2 and discretize the equations in space using the finite volume method in Sect. 8.3. Together with the model equations for special structures, this discretization process leads to a system of ordinary differential equations for the finite volume states, which will be the state restrictions for our time-continuous optimization problem introduced in Sect. 8.6.

Next we discretize this optimization problem in time and use the method of Lagrange multipliers to present the gradient of the Lagrange function as a function of the system's adjoint state in Sect. 8.7. In Sect. 8.9 we display and discuss the numerical results for optimal control of the academic case study network introduced in Sect. 7.3.

8.2 Shallow Water Equations on Networks

Traditionally, an urban drainage network is modelled as a mathematical, directed graph $\mathscr{G} = (\mathscr{V}, \mathscr{E})$ consisting of a set of vertices $v_i \in \mathscr{V}$ representing channel junc-

A. Martin et al. (eds.), *Mathematical Optimization of Water Networks*,
International Series of Numerical Mathematics 162,
DOI 10.1007/978-3-0348-0436-3_8,

tions and a set of edges $e_{i,j} \in \mathscr{E}$ representing prismatic sewer channels, where $e_{i,j}$ connects the vertex v_i to the vertex v_j. We interpret each $e_{i,j} \in \mathscr{E}$ as a 1-dimensional space domain $e_{i,j} = (a_{i,j}, b_{i,j}) \subset \mathbb{R}$. Furthermore we consider a general time interval $T := [t^0, t^1]$ for all edges. For a vertex v_j we define the index set of ingoing channels $\mathscr{I}_{in}(j) := \{i \in \mathbb{N} : e_{i,j} \in \mathscr{E}\}$, likewise we define the index set of outgoing channels $\mathscr{I}_{out}(j) := \{k \in \mathbb{N} : e_{j,k} \in \mathscr{E}\}$.

Next, we introduce the *flow state* y describing the behavior of the liquid along the channels. On each edge $e_{i,j}$ we define:

$$y_{i,j} := \begin{cases} e_{i,j} \times T \to \mathbb{R}^+ \times \mathbb{R} =: \Omega_y, \\ (x,t) \mapsto y_{i,j}(x,t) = \binom{A_{i,j}(x,t)}{Q_{i,j}(x,t)} \end{cases}$$

where $A_{i,j}(x,t)$ is the *wetted cross-sectional area* and $Q_{i,j}(x,t)$ is the *flow rate* of the liquid.

We want the flow states $y_{i,j}(x,t)$ on each edge to fulfill the *Shallow Water Equations*:

$$\partial_t y_{i,j}(x,t) + \partial_x F_{i,j}\big(y_{i,j}(x,t)\big) = S_{i,j}(y_{i,j}, t) \tag{8.1}$$

where

$$F_{i,j}(y_{i,j}) := \begin{pmatrix} Q_{i,j} \\ \frac{Q_{i,j}^2}{A_{i,j}} + \eta_{i,j}(A_{i,j}) \end{pmatrix} \tag{8.2}$$

is called *flux function* and

$$S_{i,j}(y_{i,j}, t) := \begin{pmatrix} s_M(t) \\ s_P(A_{i,j}, Q_{i,j}) \end{pmatrix} \tag{8.3}$$

is a *source function* consisting of a mass source s_M, like lateral flow, and a momentum source s_P, like friction and down-hill acceleration.

The function $\eta_{i,j}$ is called *hydrostatic pressure function* and defined as

$$\eta_{i,j}(A) := g \int_0^{h(A)} \big(h(A) - z\big) w_{i,j}(z)\, \mathrm{d}z \tag{8.4}$$

where $h(A)$ is the free surface height of the liquid in the channel $e_{i,j}$, while $w_{i,j}(h(A))$ is the corresponding width of the channel profile and $g \approx 9.81$ is the gravity constant. Some of these quantities are depicted in Fig. 8.1.

Next we look at the boundary states for system (8.1): The boundary states at initial time, namely $y_{i,j}(x, t^0)$, are given. But the boundary states at the channel boundaries, $y_{i,j}(a_{i,j}, t)$ and $y_{i,j}(b_{i,j}, t)$, have to fulfill the so-called *coupling conditions*, which come into effect, if two or more channels meet at a common vertex v_j:

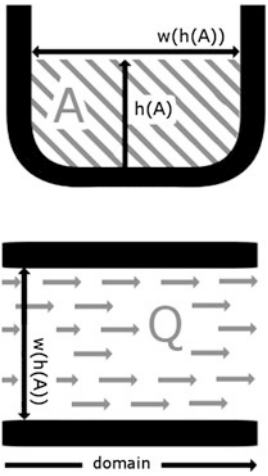


Fig. 8.1 Cross-sectional view (*top*), and bird view (*bottom*) of a channel

The most important of these conditions is *Kirchhoff's junction rule*, which guarantees that no mass is lost as the liquid flows across the vertices v_j:

$$\sum_{i \in \mathscr{I}_{in}(j)} Q_{i,j}(b_{i,j}, t) = \sum_{k \in \mathscr{I}_{out}(j)} Q_{j,k}(a_{j,k}, t) \quad \text{for all } j : v_j \in \mathscr{V},\ t \in T. \tag{8.5}$$

In general, Kirchhoff's junction rule is completed with another coupling condition like *continuity of free surface height*:

$$\begin{aligned} &h\big(A_{i,j}(b_{i,j}, t)\big) = h\big(A_{j,k}(a_{i,j}, t)\big) \\ &\text{for all } i, j, k: \quad i \in \mathscr{I}_{in}(j) \wedge k \in \mathscr{I}_{out}(j),\ t \in T \end{aligned} \tag{8.6}$$

or *continuity of particle velocity*:

$$\frac{Q_{i,j}(b_{i,j}, t)}{A_{i,j}(b_{i,j}, t)} = \frac{Q_{j,k}(a_{i,j}, t)}{A_{j,k}(a_{i,j}, t)} \quad \text{for all } i, j, k : i \in \mathscr{I}_{in}(j) \wedge k \in \mathscr{I}_{out}(j). \tag{8.7}$$

8.3 Finite Volume Discretization

In order to solve system (8.1) numerically, we discretize the edges $e_{i,j} \in \mathscr{E}$ into sets of disjoint intervals of space-constant state. This approach is called *finite volume method* (see for example [6, pages 64f]). This method is conservative per construction and is well-suited for the mixture of subcritical and supercritical flow, which occurs frequently in urban drainage systems. The numerical scheme is derived as follows:

For simplicity we first assume a network consisting only of a single channel $e := (a, b)$ and try to solve (8.1) with initial state $y(x, t^0)$ and admissible boundary

conditions $y(a,t)$ and $y(b,t)$. We divide the space e into a partition of $\mu = 1,\dots,m$ open intervals X^μ, the so-called *finite volume elements* or *grid cells*. Per definition each interval X^μ is centered around the *element midpoint* $x^\mu \in e$, so we get

$$X^\mu = \left(x^\mu - \frac{1}{2}\left|X^\mu\right|, x^\mu + \frac{1}{2}\left|X^\mu\right|\right) =: \left(a^\mu, b^\mu\right) \quad \text{for all } \mu = 1,\dots,m \tag{8.8}$$

In addition we get the following common boundary points:

$$a^\mu = \begin{cases} a & \text{if } \mu = 1, \\ b^{(\mu-1)} & \text{if } \mu = 2,\dots,m \end{cases}$$

and

$$b^\mu = \begin{cases} b & \text{if } \mu = m, \\ a^{(\mu+1)} & \text{if } \mu = 1,\dots,m-1. \end{cases}$$

We now assume that the state $y(x,t)$ can be approximated by a staircase function $\bar{y}(x,t)$ defined on the finite volume elements as:

$$\bar{y} := \begin{cases} \bigcup_{\mu=1}^m X^\mu \times T \to \Omega_y, \\ (x,t) \mapsto \bar{y}^\mu(t) \quad \text{for } x \in X^\mu \end{cases}$$

where $y^\mu(t)$ is a space-constant value on each finite volume element X^μ.

We plug this staircase function into the shallow water equations (8.1), and if we integrate (8.1) over some element X^μ, we get

$$\int_{X^\mu} \partial_t \bar{y}(x,t) + \partial_x F\left(\bar{y}(x,t)\right) - S\left(\bar{y}(x,t),t\right) \mathrm{d}x = 0. \tag{8.9}$$

Because the staircase function is constant on the finite volume element, this is equivalent to

$$\partial_t \bar{y}^\mu(t) + \frac{F(\bar{y}(b^\mu,t)) - F(\bar{y}(a^\mu,t))}{|X^\mu|} - S\left(\bar{y}^\mu(t),t\right) \mathrm{d}x = 0 \tag{8.10}$$

whereas the element boundary values $\bar{y}(a^\mu,t)$, $\bar{y}(b^\mu,t)$ are yet to be defined. In fact it is only necessary to define $F(\bar{y}(a^\mu,t))$ and $F(\bar{y}(b^\mu,t))$, but we have to respect the *conservation condition*

$$F\left(\bar{y}\left(a^{\mu+1},t\right)\right) = F\left(\bar{y}\left(b^\mu,t\right)\right) \tag{8.11}$$

as otherwise our solution approach is not conservative. An intuitive choice for the flux is therefore: $F(\bar{y}(a^{\mu+1},t)) = F(\bar{y}(b^\mu,t)) = F(f(\bar{y}^\mu(t),\bar{y}^{\mu+1}(t)))$, where $f : \Omega_y \times \Omega_y \to \Omega_y$ maps two neighbored finite volume states onto some "average"

state. We can combine the mapping f with the flux function F to introduce the *numerical flux*

$$\bar{F} := \begin{cases} \Omega_y \times \Omega_y \to \mathbb{R}^2, \\ (\bar{y}^\mu, \bar{y}^{\mu+1}) \mapsto \bar{F}(\bar{y}^\mu, \bar{y}^{\mu+1}) = F(f(\bar{y}^\mu, \bar{y}^{\mu+1})) \end{cases} \tag{8.12}$$

which in the following fulfills the *consistency condition* to the continuous flux F from (8.1):

Definition 1 Let y^μ, y^ν both in Ω_y be the discrete states of the neighbored finite volume elements X^μ, X^ν. A numerical flux function $\bar{F} : \Omega_y \times \Omega_y \to \mathbb{R}^2$ is consistent to a flux function $F : \Omega_y \to \mathbb{R}^2$ if

$$\bar{F}(y, y) = F(y) \quad \text{for all } y \in \Omega_y \tag{8.13}$$

and if there exists a constant $L > 0$ such that

$$\left\| \bar{F}\left(y^\mu, y^\nu\right) - F(y) \right\| \leq L \max\left(\left\| y^\mu - y \right\|, \left\| y^\nu - y \right\| \right) \tag{8.14}$$

for $y^\mu, y^\nu \in B_\epsilon(y)$ with $\epsilon > 0$ sufficient small.

We then formulate:

Problem 1 (Finite Volume ODE on Edges) For a sequence of $\mu = 1, \ldots, m$ finite volume elements $X^\mu_{i,j}$ partitioning a channel $e_{i,j}$ we look for finite volume states $y^\mu_{i,j}(t)$ with $t \in T$ fulfilling:

$$\partial_t y^\mu_{i,j}(t) = \frac{\bar{F}_{i,j}(y^{\mu-1}_{i,j}(t), y^\mu_{i,j}(t)) - \bar{F}_{i,j}(y^\mu_{i,j}(t), y^{\mu+1}_{i,j}(t))}{|X^\mu|} + S_{i,j}\left(y^\mu_{i,j}(t), t\right) \tag{8.15}$$

with the boundary fluxes

$$\bar{F}_{i,j}\left(y^0_{i,j}(t), y^1_{i,j}(t)\right) =: \bar{F}^a_{i,j}, \tag{8.16}$$

$$\bar{F}_{i,j}\left(y^m_{i,j}(t), y^{m+1}_{i,j}(t)\right) =: \bar{F}^b_{i,j} \tag{8.17}$$

and the initial conditions $y^\mu_{i,j}(t^0)$ given.

Lax and Wendroff (compare [4], and [6, page 240]) have shown that, if the solution of Problem 1 is stable, this solution is a approximation of a weak solution of system (8.1) on the channel $e_{i,j}$.

There are multiple choices for defining a numerical flux function $\bar{F}$ consistent to the shallow water flux F, a good overview with detailed discussion can be found in [6, pages 67f]. In this work we stick with Godunov's flux (see [8, page 152]). To evaluate Godunov's flux for the states y^μ, $y^{\mu+1}$, we first have to find the solution $y^{\mu,\mu+1}(x, t)$ of the shallow water Riemann problem:

Problem 2 For $y^{\mu}, y^{\mu+1} \in \Omega_y$ given, find $y^{\mu,\mu+1} : X^{\mu} \cup X^{\mu+1} \times T \to \Omega_y$ solving system (8.1) under the piecewise constant initial conditions

$$y^{\mu,\mu+1}(x,t^0) := \begin{cases} y^{\mu} & \text{for } x \in X^{\mu}, \\ y^{\mu+1} & \text{for } x \in X^{\mu+1}. \end{cases} \tag{8.18}$$

The solution $y^{\mu,\mu+1}(x,t)$ is always a similarity solution:

$$y^{\mu,\mu+1}(x,t) \equiv const. \quad \text{if } \frac{x}{t} \equiv const. \tag{8.19}$$

and we are especially interested in the value

$$y_0^{\mu,\mu+1} := y^{\mu,\mu+1}(x,t)\big|_{\frac{x}{t}=0}. \tag{8.20}$$

For $t \in T$ Godunov's numerical flux is therefore

$$\bar{F}\big(y^{\mu-1}, y^{\mu}\big) = F\big(y_0^{\mu,\mu+1}\big). \tag{8.21}$$

8.4 Finite Volume Junctions

After discretizing the edges $e_{i,j} \in \mathscr{E}$ in finite volume elements, we look at the vertices $v_j \in \mathscr{V}$, where multiple edges $e_{i,j}$ and $e_{j,k}$ meet. We denote the last element in $e_{i,j}$ as $X_{i,j}^m$ and its state is $y_{i,j}^m$. Accordingly we denote $X_{j,k}^1$ as the first element of $e_{j,k}$ and its state is $y_{j,k}^1$. All of these finite volume elements are neighbored to vertex v_j and their states fulfill (8.15).

Problem 1 requires, that the fluxes $\bar{F}_{i,j}^b$ and $\bar{F}_{j,k}^a$ at the boundaries are given a priori, but of course we want to select them physically correct and consistent to the shallow water equations.

First of all we assume that the channel profiles of the inflowing channels match the shape profiles of the outflowing channels:

$$\sum_{i \in \mathscr{I}_{in}(j)} w_{i,j} = \sum_{k \in \mathscr{I}_{out}(j)} w_{j,k} =: w_j. \tag{8.22}$$

Then we demand, that the union of the inflowing and outflowing boundary fluxes behaves like shallow water in a prismatic channel:

$$\sum_{i \in \mathscr{I}_{in}(j)} \bar{F}_{i,j}^b = \sum_{k \in \mathscr{I}_{out}(j)} \bar{F}_{j,k}^a = \bar{F}_j\Big(\sum_{i \in \mathscr{I}_{in}(j)} y_{i,j}^m, \sum_{k \in \mathscr{I}_{out}(j)} y_{j,k}^1 \Big). \tag{8.23}$$

Furthermore we want the boundary states $y^m_{i,j}$ and $y^1_{j,k}$ to fulfill the following continuity conditions:

$$\left.\begin{aligned} h(A^m_{i,j}) &= h(A^m_{i',j}) \\ \frac{Q^m_{i,j}}{A^m_{i,j}} &= \frac{Q^m_{i',j}}{A^m_{i',j}} \end{aligned}\right\} \quad \text{for all } i, i' \in \mathscr{I}_{in}(j), \tag{8.24}$$

$$\left.\begin{aligned} h(A^1_{j,k}) &= h(A^1_{j,k'}) \\ \frac{Q^1_{j,k}}{A^1_{j,k}} &= \frac{Q^1_{j,k'}}{A^1_{j,k'}} \end{aligned}\right\} \quad \text{for all } k, k' \in \mathscr{I}_{out}(j). \tag{8.25}$$

The conditions (8.23)–(8.25) are easy to treat, as the following proposition shows:

Proposition 1 *Let* $|X^m_{i,j}| =: |X^{in}_j| \in \mathbb{R}^+$ *for all* $i \in \mathscr{I}_{in}(j)$ *and likewise let* $|X^1_{j,k}| =: |X^{out}_j| \in \mathbb{R}^+$ *for all* $k \in \mathscr{I}_{out}(j)$. *Let* $y^{in}_j, y^{out}_j : T \to \Omega_y$ *fulfill*

$$\partial_t y^{in}_j = \frac{\sum_{i \in \mathscr{I}_{in}(j)} \bar{F}_{i,j}(y^{m-1}_{i,j}, y^m_{i,j}) - \bar{F}_j(y^{in}_j, y^{out}_j)}{|X^{in}_j|} + \sum_{i \in \mathscr{I}_{in}(j)} S_{i,j}\big(y^m_{i,j}\big), \tag{8.26}$$

$$\partial_t y^{out}_j = \frac{\bar{F}_j(y^{in}_j, y^{out}_j) - \sum_{k \in \mathscr{I}_{out}(j)} \bar{F}_{j,k}(y^1_{j,k}, y^2_{j,k})}{|X^{out}_j|} + \sum_{k \in \mathscr{I}_{out}(j)} S_{j,k}\big(y^1_{j,k}\big). \tag{8.27}$$

For y^{in}_j *let* h^{in}_j *be the corresponding free surface height and likewise* h^{out}_j *the corresponding free surface height for* y^{out}_j. *Then the boundary states*

$$\left.\begin{aligned} y^m_{i,j} &:= \begin{pmatrix} \int_0^{h^{in}_j} w_{i,j}(z)\,\mathrm{d}z \\ \int_0^{h^{in}_j} \frac{Q^{in}_j}{A^{in}_j} w_{i,j}(z)\,\mathrm{d}z \end{pmatrix} \\ y^1_{j,k} &:= \begin{pmatrix} \int_0^{h^{out}_j} w_{j,k}(z)\,\mathrm{d}z \\ \int_0^{h^{out}_j} \frac{Q^{out}_j}{A^{out}_j} w_{j,k}(z)\,\mathrm{d}z \end{pmatrix} \end{aligned}\right\} \tag{8.28}$$

fulfill (8.23)–(8.25).

Proof Condition (8.24)–(8.25) follows directly from (8.28). Because of (8.22) we verify

$$y^{in}_j = \sum_{i \in \mathscr{I}_{in}(j)} y^m_{i,j}, \tag{8.29}$$

$$y^{out}_j = \sum_{k \in \mathscr{I}_{out}(j)} y^1_{j,k} \tag{8.30}$$

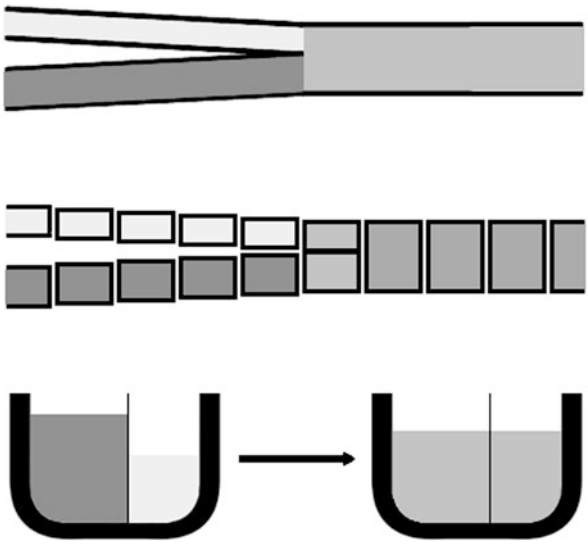


Fig. 8.2 Channel conjunction (*top*), finite volume discretization (*middle*), cross-sections at junction (*bottom*)

and using (8.15) leads to

$$0 = \partial_t \left(y_j^{in} - \sum_{i \in \mathscr{I}_{in}(j)} y_{i,j}^m \right) = -\bar{F}_j\left(y_j^{in}, y_j^{out}\right) + \sum_{i \in \mathscr{I}_{in}(j)} \bar{F}_{i,j}^b \tag{8.31}$$

and likewise

$$0 = \partial_t \left(y_j^{out} - \sum_{k \in \mathscr{I}_{out}(j)} y_{j,k}^1 \right) = \bar{F}_j\left(y_j^{in}, y_j^{out}\right) - \sum_{k \in \mathscr{I}_{out}(j)} \bar{F}_{j,k}^a \tag{8.32}$$

which leads directly to condition (8.23). □

This proposition gives no explicit description of the fluxes $\bar{F}_{i,j}^a, \bar{F}_{j,k}^b$, but these are not needed, as (8.26) is sufficient for the numerical computations.

An example for this setting with two channels at the inflow and one at the outflow of a vertex is given in Fig. 8.2.

8.5 The Finite Volume Network Problem

With Problem 1 and Proposition 1 we have gathered all tools to compute shallow water flow on networks numerically, as long as (8.22) holds on the channel junctions. But in practical applications this is not sufficient, as on the one hand (8.22) doesn't hold in general on all channel junctions and on the other hand, we still have to face the flow of the liquid across special structures like pumps, weirs and storage containers, which cannot be described by shallow water equations.

Nevertheless we found an elegant way to unite all of these special structures together with the shallow water network approach into one model. Looking at (8.15) and (8.26) we realize that both equations share the same structure: The development of the state y in the finite volume element X depends on the one hand on the inner source term $S(y)$. On the other hand, state y is influenced by the balance of the inflowing and outflowing flux functions at the boundary of element X. The idea is now, to replace some of the numerical flux functions $\bar{F}$ with some other functions $F_{\mu,\nu}$, describing the behavior of the liquid across pumps, weirs and other special structures.

In this process we lose the representation of the domain as a graph $\mathscr{G} = (\mathscr{V}, \mathscr{E})$ of 1-dimensional channels and channel junctions, but instead gain a new graph $\mathscr{G} = (\mathscr{X}, \mathscr{F})$ with the finite volume elements $X_\mu \in \mathscr{X}$ functioning as vertices and the flux functions $f_{\mu,\nu} \in \mathscr{F}$ representing flux exchange edges. For convenience the indices κ, μ, ν for the finite volume elements are now placed in the subscripts and the index sets change to $\mathscr{I}_{in}(\nu) := \{\kappa \in \mathbb{N} : f_{\kappa,\mu} \in \mathscr{F}\}$ and $\mathscr{I}_{out}(\nu) := \{\nu \in \mathbb{N} : f_{\mu,\nu} \in \mathscr{F}\}$. We define:

Definition 2 The locally Lipschitz continuous function

$$F_{\mu,\nu} : \begin{cases} \Omega_y^2 \times T \to \mathbb{R}^2 \\ (y_\mu, y_\nu, t) \mapsto F_{\mu,\nu}(y_\mu, y_\nu, t) \end{cases} \tag{8.33}$$

describing the flow at time $t \in T$ from a finite volume element X_μ with state $y_\mu \in \Omega_y$ into a neighbored element X_ν with state y_ν is called *general flux function*. Furthermore the locally Lipschitz continuous function

$$S_\mu : \begin{cases} \Omega_y \times T \to \mathbb{R}^2 \\ (y_\mu, t) \mapsto S_\mu(y_\mu, t) \end{cases} \tag{8.34}$$

describing the internal change of the state y_μ in an element X_μ is called *general source function*.

These general flux and source functions are highly flexible in design and are able to describe especially extensive settings in urban drainage modeling. Some examples for general flux functions are:

$$F_{\mu,\nu}(y_\mu, y_\nu, t) = \begin{pmatrix} q \\ 0 \end{pmatrix} \quad \text{(pump with constant pump rate } q\text{)}, \tag{8.35}$$

$$F_{\mu,\nu}(y_\mu, y_\nu, t) = \bar{F}(y_\mu, \vec{0}) \quad \text{(weir overflow)}, \tag{8.36}$$

$$F_{\mu,\nu}(y_\mu, y_\nu, t) = F\left(\begin{pmatrix} A_\mu \\ 0 \end{pmatrix}\right) \quad \text{(impulse loss at wall)}. \tag{8.37}$$

We end this section with our final model equation:

Problem 3 (Finite Volume Network Problem) Let $\mathscr{X}$ be the set of finite volume elements connected by general flux functions $F_{\mu,\nu}$ from Definition 2. We look for the states $y_\mu : T \to \Omega_y$ associated with $X_\mu \in \mathscr{X}$, fulfilling

$$\partial_t y_\mu = S_\mu(y_\mu, t) + \frac{\sum_{\kappa \in \mathscr{I}_{in}(\mu)} F_{\kappa,\mu}(y_\kappa, y_\mu, t) - \sum_{\nu \in \mathscr{I}_{out}(\mu)} F_{\mu,\nu}(y_\mu, y_\nu, t)}{|X_\mu|} \tag{8.38}$$

with initial conditions $y_\mu(t^0) \in \Omega_y$ given.

Reviewing Definition 2, we can use existence and uniqueness theorems for systems of ordinary differential equations (see for example [2, page 332]) to make sure that Problem 3 is well-defined. Furthermore we require (8.38) to only hold in the weak sense, e.g. $y_\mu \in C^0(T; \Omega_y)$ fulfills

$$\begin{aligned}\int_T - y_\mu \partial_t \psi_\mu \, \mathrm{d}t - y_\mu\big(t^0\big)\psi_\mu\big(t^0\big) \\ = \int_T \left(S_\mu(y_\mu) + \frac{\sum_{\kappa \in \mathscr{I}_{in}(\mu)} F_{\kappa,\mu}(y_\kappa, y_\mu) - \sum_{\nu \in \mathscr{I}_{out}(\mu)} F_{\mu,\nu}(y_\mu, y_\nu)}{|X_\mu|} \right) \psi_\mu \, \mathrm{d}t\end{aligned}$$

for all test functions $\psi_\mu \in \{f \in C^1(T; \Omega_y) : f(t^1) = 0\}$.

8.6 The Optimal Control Problem

In the last section we completed the development of our process model, which allows us to predict sewer system flow for given initial and boundary settings. With the help of controllable structures, like movable weirs or adjustable pumps, we are able to influence this sewer system flow in a small range during runtime. These controllable structures are modelled as general flux functions, where some constants (like weir height) or time-dependent behavior (like pump curves) are replaced by time-dependent *control functions*.

$$u_{\mu,\nu} : \begin{cases} T \to [u^0, u^1] =: \Omega_u \subset \mathbb{R}, \\ t \mapsto u_{\mu,\nu}(t). \end{cases} \tag{8.39}$$

This replacement allows us to influence the flux exchange behavior of the corresponding general flux function, which is therefore called *controllable flux function*:

$$F_{\mu,\nu} : \begin{cases} \Omega_y^2 \times T \times \Omega_u \to \mathbb{R}^2, \\ (y_\mu, y_\nu, u_{\mu,\nu}, t) \mapsto F_{\mu,\nu}(y_\mu, y_\nu, u_{\mu,\nu}, t). \end{cases} \tag{8.40}$$

If we plug these controllable flux functions into (8.38), we induce a function $\mathscr{Y}$, which maps all $r \in \mathbb{N}$ control functions of the network gathered in $U \in$

$L^\infty(T;\Omega_u^r)$ onto the corresponding $\mu = 1,\dots,m$ state functions $(y_1,\dots,y_m) =: Y \in C^0(T;\Omega_y^m)$. Then we formulate:

Problem 4 (ODE-Constrained Optimal Control Problem) For a finite volume network with time-dependent state vector $(y_1,\dots,y_m)=:Y \in C^0(T;\Omega_y^m)$ controlled by $r \in \mathbb{N}$ control functions $U \in L^\infty(T;\Omega_u^r)$ we want to find the optimal control $U^* \in L^\infty(T;\Omega_u^r)$ and corresponding optimal state $Y^* \in C^0(T;\Omega_y^m)$ solving

$$\min_{Y,U} J(Y,U) := \int_T G\big(Y(t),U(t)\big)\,\mathrm{d}t + \tilde{G}\big(Y\big(t^1\big)\big) \tag{8.41}$$

subject to

$$\left.\begin{aligned} \partial_t Y &= H(Y,U,t) \\ Y(t^0) &\in \Omega_y \quad \text{given} \end{aligned}\right\} \tag{8.42}$$

where G and $\tilde{G}$ are objective functions, and the μth component of $H(Y,U,t)$ is

$$\partial_t y_\mu = S_\mu(y_\mu,t) + \frac{\sum_{\kappa\in\mathscr{I}_{in}(\mu)} F_{\kappa,\mu}(y_\kappa,y_\mu,u_{\kappa,\mu}) - \sum_{\nu\in\mathscr{I}_{out}(\mu)} F_{\mu,\nu}(y_\mu,y_\nu,u_{\mu,\nu})}{|X_\mu|}. \tag{8.43}$$

Problem 4 is a terminal-cost optimal control problem and is subject to well-known results from optimal control theory. One important theorem is *Pontryagin's minimum principle* (also known as *maximal principle*) (see [1, page 284f] or [5, page 239]), that states necessary optimality conditions for a solution of Problem 4.

First, we introduce the $\mu = 1,\dots,m$ *adjoint states* (or costates) $w_\mu \in C^0(T;\mathbb{R})$, which are gathered into the adjoint vector $W := (w_1, w_2, \dots, w_m)$.

Then we define the *Hamiltonian function*

$$\mathscr{H} : \begin{cases} \Omega_y^m \times \mathbb{R}^m \times \Omega_u^r \times T \to \mathbb{R} \\ (Y,W,U,t) \mapsto \mathscr{H}(Y,W,U) = G(Y,U) + W^\top H(Y,U,t) \end{cases} \tag{8.44}$$

which is used in the following theorem:

Theorem 1 (Pontryagin's Minimum Principle) *For an optimal control U^* solving Problem 4 with optimal state Y^*, there exists an adjoint state $W^* \in C^0(T;\mathbb{R}^m)$ such that the Hamiltonian system*

$$\partial_t Y(t) = \partial_W \mathscr{H}\big(Y(t),W(t),U(t),t\big), \tag{8.45}$$

$$\partial_t W(t) = -\partial_Y \mathscr{H}\big(Y(t),W(t),U(t),t\big), \tag{8.46}$$

$$Y\big(t^0\big) \in \Omega_y \quad \textit{given and} \quad W\big(t^1\big) = \partial_Y \tilde{G}\big(Y\big(t^1\big)\big)$$

is solved, and for every $t \in T$ the optimal control $U^*(t)$ fulfills

$$\mathscr{H}\big(Y^*(t),W^*(t),U^*(t),t\big) = \min_{U\in\Omega_u^r} \mathscr{H}\big(Y^*(t),W^*(t),U,t\big). \tag{8.47}$$

This theorem allows us to find solutions of Problem 4 by finding solutions of (8.47), where Y^*, W^* solve (8.45). The adjoint equation (8.47) is linear in W and therefore a unique solution exists. It is possible to derive necessary and sufficient optimality conditions for a solution $U^*(t)$ of (8.47) and solve them together with system (8.45) numerically, but in this project we used a direct approach, e.g. we first discretize Problem 4 and then we derive and solve the optimality conditions for the discrete problem as it is shown in the following section.

8.7 Discrete Optimal Control Problem

In order to derive the discrete optimal control problem, we first divide the continuous time horizon $T := (t^0, t^1)$ into a set of $n+1$ discrete and equidistant time nodes $T^\tau := \{\tau_0, \tau_1, \ldots, \tau_{n-1}, \tau_n\}$ with constant distance Δt and $\tau_0 := t^0$, $\tau_{n+1} := t^1$. We assume to solve (8.42) for a finite volume network consisting of the $\mu = 1, \ldots, m$ finite volume elements $X_\mu \in \mathscr{X}$ supporting the states $y_\mu^{\tau_t} \in \Omega_y$ at time step $\tau_t \in T^\tau$. We gather these network element states $y_\mu^{\tau_t}$ to the state vector $Y^{\tau_t} \in \Omega_y^m$. Furthermore we introduce the set $Y^\tau := \{Y^{\tau_1}, \ldots, Y^{\tau_n}\}$ of all state vectors at all time steps. The state Y^{τ_0} is not included in this set as it is already determined.

Likewise, we discretize our control function $U \in \Omega_u^r$ on the same discrete time horizon T^τ into control vectors U^{τ_t} which are gathered as $U^\tau := \{U^{\tau_1}, \ldots, U^{\tau_n}\}$.

The constraint system (8.42) is approximated by an iterative scheme of the kind

$$Y^{\tau_{t+1}} = Y^{\tau_t} + \Delta t\big((1-\alpha)H\big(Y^{\tau_t}, U^{\tau_t}, \tau_t\big) + \alpha H\big(Y^{\tau_{t+1}}, U^{\tau_{t+1}}, \tau_{t+1}\big)\big) \tag{8.48}$$

where the μth component of H is

$$\begin{aligned} &H_\mu^\tau\big(Y^\tau, U^\tau, \tau_t\big) \\ &\quad := S_\mu\big(y^{\tau_t}, \tau_t\big) \\ &\qquad + \frac{\sum_{\kappa \in \mathscr{I}_{in}(\mu)} F_{\kappa,\mu}(y_\kappa^{\tau_t}, y_\mu^{\tau_t}, u_{\kappa,\mu}^{\tau_t}, \tau_t) - \sum_{\nu \in \mathscr{I}_{out}(\mu)} F_{\mu,\nu}(y_\mu^{\tau_t}, y_\nu^{\tau_t}, u_{\mu,\nu}^{\tau_t}, \tau_t)}{|X_\mu|} \end{aligned} \tag{8.49}$$

and $U^{\tau_t}, U^{\tau_{t+1}}$ as well as Y^{τ_t} are given.

The parameter $\alpha \in \{0, \frac{1}{2}, 1\}$ specifies the chosen iterative scheme, and it is easy to see that $\alpha = 0$ resembles the explicit Euler method, $\alpha = 1$ implies the implicit Euler method and for $\alpha = \frac{1}{2}$ we get the trapezoidal rule.

After discretizing the state constraint of Problem 4 we now want to discretize the cost functional (8.41). Let us assume that we have computed a discrete state vector Y^{τ_t} for $t = 1, \ldots, n$ using (8.48) for some $\alpha \in \{0, \frac{1}{2}, 1\}$.

Based on the states Y^τ and control U^τ we approximate the continuous cost functional through a discrete version:

$$J(Y, U) \approx J^\tau\big(Y^\tau, U^\tau\big) := \Delta t \sum_{i=1}^{n} G\big(Y^{\tau_i}, U^{\tau_i}\big) + \tilde{G}\big(y^{\tau_n}\big). \tag{8.50}$$

Next we define the residual function

$$R\left(Y^{\tau_t}, Y^{\tau_{t-1}}, U^{\tau_t}, U^{\tau_{t-1}}\right) := Y^{\tau_t} - Y^{\tau_{t-1}} - \Delta t\left((1-\alpha)H\left(Y^{\tau_{t-1}}, U^{\tau_{t-1}}, \tau_{t-1}\right) + \alpha H\left(Y^{\tau_t}, U^{\tau_t}, \tau_t\right)\right) \tag{8.51}$$

with Jacobian matrix

$$\nabla R\left(Y^{\tau_t}, Y^{\tau_{t-1}}, U^{\tau_t}, U^{\tau_{t-1}}\right) = \begin{pmatrix} \mathbb{1} & -\Delta t\alpha\nabla_{Y^{\tau_t}} H(Y^{\tau_t}, U^{\tau_t}, \tau_t) \\ -\mathbb{1} & -\Delta t(1-\alpha)\nabla_{Y^{\tau_{t-1}}} H(Y^{\tau_{t-1}}, U^{\tau_{t-1}}, \tau_{t-1}) \\ & -\Delta t\alpha\nabla_{U^{\tau_t}} H(Y^{\tau_t}, U^{\tau_t}, \tau_t) \\ & -\Delta t(1-\alpha)\nabla_{U^{\tau_{t-1}}} H(Y^{\tau_{t-1}}, U^{\tau_{t-1}}, \tau_{t-1}) \end{pmatrix}$$

and see at once that $R(Y^{\tau_{t+1}}, Y^{\tau_t}, U^{\tau_{t+1}}, U^{\tau_t}) = 0$ is equivalent to (8.48). We end up with a finite dimensional optimization problem:

Problem 5 (Discrete Optimal Control Problem) Find $U^\tau \in (\Omega_u^r)^n$ solving

$$\min_{U^\tau} J^\tau\left(Y^\tau, U^\tau\right) := \Delta t \sum_{i=1}^{n} G\left(Y^{\tau_i}, U^{\tau_i}\right) + \tilde{G}\left(y^{\tau_n}\right) \tag{8.52}$$

subject to

$$R\left(Y^{\tau_t}, Y^{\tau_{t-1}}, U^{\tau_t}, U^{\tau_{t-1}}\right) = 0 \quad \text{for all } t = 1, \ldots, n \tag{8.53}$$

and Y^{τ_0} given.

We want to use the Lagrangian technique for Problem 5 to construct necessary optimality conditions.

The corresponding Lagrangian function is defined as:

$$\Lambda\left(Y^\tau, U^\tau, W^\tau\right) := J^\tau\left(Y^\tau, U^\tau\right) - W^\tau \cdot \begin{pmatrix} R(Y^{\tau_1}, Y^{\tau_0}, U^{\tau_1}, U^{\tau_0}) \\ \vdots \\ R(Y^{\tau_n}, Y^{\tau_{n-1}}, U^{\tau_n}, U^{\tau_{n-1}}) \end{pmatrix}$$

where $W^\tau := [W^{\tau_1}, W^{\tau_2}, \ldots, W^{\tau_n}]$ is the row vector of Lagrangian multipliers and each $W^{\tau_t} := [w_1^{\tau_t}, \ldots, w_m^{\tau_t}]$ is divided in m multiplier sets associated with the finite volume elements.

Next we try to find stationary points of the Lagrangian. Sufficient conditions for these stationary points are (compare [3, page 124]):

$$\nabla_{Y^\tau} \Lambda\left(Y^\tau, U^\tau, W^\tau\right) = 0, \tag{8.54}$$

$$\nabla_{U^\tau} \Lambda\left(Y^\tau, U^\tau, W^\tau\right) = 0, \tag{8.55}$$

$$\nabla_{W^\tau} \Lambda\left(Y^\tau, U^\tau, W^\tau\right) = 0. \tag{8.56}$$

We look at the left side of (8.54) which is a gradient (row vector) consisting of subvectors $\nabla_{Y^{\tau_t}} \Lambda(Y^\tau, U^\tau, W^\tau)$.

We get

$$\begin{aligned}
&\nabla_{Y^{\tau_t}} \Lambda\big(Y^\tau, U^\tau, W^\tau\big) \\
&\quad = \Delta t \nabla_{Y^{\tau_t}} G\big(Y^{\tau_t}, U^{\tau_t}\big) - W^\tau \cdot \begin{pmatrix} 0 \\ \vdots \\ 0 \\ \mathbb{1} - \Delta t \alpha \nabla_{Y^{\tau_t}} H(Y^{\tau_t}, U^{\tau_t}, \tau_t) \\ -\mathbb{1} - \Delta t (1-\alpha) \nabla_{Y^{\tau_t}} H(Y^{\tau_t}, U^{\tau_t}, \tau_t) \\ 0 \\ \vdots \\ 0 \end{pmatrix}
\end{aligned}$$

for $t = 1, \ldots, n-1$. And for $t = n$ we get

$$\begin{aligned}
&\nabla_{Y^{\tau_n}} \Lambda\big(Y^\tau, U^\tau, W^\tau\big) \\
&\quad = \Delta t \nabla_{Y^{\tau_n}} G\big(Y^{\tau_n}, U^{\tau_n}\big) + \nabla_{Y^{\tau_n}} \tilde{G}\big(Y^{\tau_n}\big) - W^{\tau_n} \cdot \big(\mathbb{1} - \Delta t \alpha \nabla_{Y^{\tau_n}} H\big(Y^{\tau_n}, U^{\tau_n}, \tau_n\big)\big).
\end{aligned}$$

Now if we define $W^{\tau_{n+1}}$ as solution of

$$W^{\tau_{n+1}} \cdot \big(\mathbb{1} + \Delta t (1-\alpha) \nabla_{Y^{\tau_t}} H\big(Y^{\tau_t}, U^{\tau_t}, \tau_t\big)\big) = \nabla_{Y^{\tau_n}} \tilde{G}\big(Y^{\tau_n}\big) \tag{8.57}$$

(8.54) can be decomposed into the *discrete adjoint equation* for W^τ:

$$\begin{aligned}
&W^{\tau_t} \cdot \big(\mathbb{1} - \Delta t \alpha \nabla_{Y^{\tau_t}} H\big(Y^{\tau_t}, U^{\tau_t}, \tau_t\big)\big) \\
&\quad = W^{\tau_{t+1}} \cdot \big(\mathbb{1} + \Delta t (1-\alpha) \nabla_{Y^{\tau_t}} H\big(Y^{\tau_t}, U^{\tau_t}, \tau_t\big)\big) + \Delta t \nabla_{Y^{\tau_t}} G\big(Y^{\tau_t}, U^{\tau_t}\big).
\end{aligned} \tag{8.58}$$

It is easy to show that there exists a $\Delta t > 0$ such that the matrix

$$\mathbb{1} - \Delta t \nabla_{Y^{\tau_t}} H\big(Y^{\tau_t}, U^{\tau_t}, \tau_t\big)$$

is regular due to diagonal dominance, and therefore (8.58) is solvable. Furthermore (8.58) is consistent to the continuous adjoint equation (8.47) from Pontryagin's minimum principle.

The next condition for a stationary point of the Lagrangian is (8.55), which equals a row vector consisting of entries

$$\begin{aligned}
&\nabla_{U^{\tau_t}} \Lambda\big(Y^\tau, U^\tau, W^\tau\big) \\
&\quad = \Delta t \nabla_{U^{\tau_t}} G\big(Y^{\tau_t}, U^{\tau_t}\big) + \big(\alpha W^{\tau_{t-1}} + (1-\alpha) W^{\tau_t}\big) \cdot \Delta t \nabla_{U^{\tau_t}} H\big(Y^{\tau_t}, U^{\tau_t}, \tau_t\big).
\end{aligned} \tag{8.59}$$

In our model we placed the control only into the flux functions, which reduces the $\mu = 1, \ldots, m$ parts of $\nabla_{U^{\tau_t}} H(Y^{\tau_t}, U^{\tau_t}, \tau_t)$ to

$$\left(\nabla_{U^{\tau_t}} H\left(Y^{\tau_t}, U^{\tau_t}, \tau_t\right)\right)_\mu = \frac{\sum_{\kappa \in \mathscr{I}_{in}(\mu)} \nabla_{U^{\tau_t}} F^{\tau_t}_{\kappa,\mu} - \sum_{\nu \in \mathscr{I}_{out}(\mu)} \nabla_{U^{\tau_t}} F^{\tau_t}_{\mu,\nu}}{|X_\mu|}. \quad (8.60)$$

Last but not least (8.56) simply equals (8.53). We use this results and some basic knowledge for the box constraints of the control to propose the necessary optimality conditions for Problem 5:

Proposition 2 *Let Δt be small enough such that $\mathbb{1} - \Delta t \nabla_{Y^{\tau_t}} H(Y^{\tau_t}, U^{\tau_t}, \tau_t)$ is regular and choose $\alpha \in \{0, \frac{1}{2}, 1\}$. Then for the solution U^τ_* of Problem 5 there exists a unique optimal state vector Y^τ_* solving*

$$Y^{\tau_{t+1}} = Y^{\tau_t} + \Delta t\left((1-\alpha) H\left(Y^{\tau_t}, U^{\tau_t}, \tau_t\right) + \alpha H\left(Y^{\tau_{t+1}}, U^{\tau_{t+1}}, \tau_{t+1}\right)\right) \quad (8.61)$$

with Y^{τ_0} given. And there exists a unique adjoint vector W^τ_* solving the discrete adjoint equation

$$\begin{aligned} &W^{\tau_t} \cdot \left(\mathbb{1} - \Delta t \alpha \nabla_{Y^{\tau_t}} H\left(Y^{\tau_t}, U^{\tau_t}, \tau_t\right)\right) \\ &\quad = W^{\tau_{t+1}} \cdot \left(\mathbb{1} + \Delta t (1-\alpha) \nabla_{Y^{\tau_t}} H\left(Y^{\tau_t}, U^{\tau_t}, \tau_t\right)\right) + \Delta t \nabla_{Y^{\tau_t}} G\left(Y^{\tau_t}, U^{\tau_t}\right) \end{aligned} \quad (8.62)$$

with

$$W^{\tau_{n+1}} \cdot \left(\mathbb{1} + \Delta t (1-\alpha) \nabla_{Y^{\tau_t}} H\left(Y^{\tau_t}, U^{\tau_t}, \tau_t\right)\right) := \nabla_{Y^{\tau_n}} \tilde{G}\left(Y^{\tau_n}\right). \quad (8.63)$$

Furthermore each component of the gradient of the Lagrangian, with the tth component being

$$\begin{aligned} &\nabla_{U^{\tau_t}} \Lambda\left(Y^\tau, U^\tau, W^\tau\right) \\ &\quad = \Delta t \nabla_{U^{\tau_t}} G\left(Y^{\tau_t}, U^{\tau_t}\right) + \left(\alpha W^{\tau_{t-1}} + (1-\alpha) W^{\tau_t}\right) \cdot \Delta t \nabla_{U^{\tau_t}} H\left(Y^{\tau_t}, U^{\tau_t}, \tau_t\right) \end{aligned} \quad (8.64)$$

fulfills one of the following conditions:

$$\nabla_{U^{\tau_t}} \Lambda\left(Y^\tau, U^\tau, W^\tau\right) = 0 \quad (8.65)$$

or

$$U^{\tau_t} = u^0 \quad \text{and} \quad \nabla_{U^{\tau_t}} \Lambda\left(Y^\tau, U^\tau, W^\tau\right) > 0 \quad (8.66)$$

or

$$U^{\tau_t} = u^1 \quad \text{and} \quad \nabla_{U^{\tau_t}} \Lambda\left(Y^\tau, U^\tau, W^\tau\right) < 0. \quad (8.67)$$

Proof Without proof. □

8.8 Design Variables

In the last section we assumed that the control function U^τ is discretized on the same time grid as the state function and that these discrete control points U^{τ_t} are independent from each other. This assumption is only of theoretical nature as we want to represent U^τ by a low-dimensional set of $k = 1, \ldots, p$ design variables π^k which are gathered as $\Pi = [\pi^1, \ldots, \pi^p]$.

In consequence, each U^{τ_t} is a function of Π. We use this in our Lagrangian to get

$$\begin{aligned}
&\Lambda\big(Y^\tau, U^\tau(\Pi), W^\tau\big) \\
&\quad := J^\tau\big(Y^\tau, U^\tau(\Pi)\big) - W^\tau \cdot \begin{pmatrix} R(Y^{\tau_1}, Y^{\tau_0}, U^{\tau_1}(\Pi), U^{\tau_0}(\Pi)) \\ \vdots \\ R(Y^{\tau_n}, Y^{\tau_{n-1}}, U^{\tau_n}(\Pi), U^{\tau_{n-1}}(\Pi)) \end{pmatrix}.
\end{aligned}$$

Since $\nabla_{Y^\tau}\Lambda, \nabla_{W^\tau}\Lambda$ in principle stay the same, we only have to apply the chain rule for the gradient $\nabla_\Pi \Lambda$:

$$\begin{aligned}
&\nabla_\Pi \Lambda\big(Y^\tau, U^\tau(\Pi), W^\tau\big) \\
&\quad = \nabla_{U^\tau}\Lambda \cdot \nabla_\Pi U^\tau \\
&\quad = \Delta t\big(\nabla_{U^{\tau_t}} G^\tau\big(Y^{\tau_t}, U^{\tau_t}(\Pi)\big) \\
&\qquad + \big(\alpha W^{\tau_{t-1}} + (1-\alpha) W^{\tau_t}\big)\nabla_{U^{\tau_t}}\big) H\big(Y^{\tau_t}, U^{\tau_t}(\Pi)\big) \cdot \nabla_\Pi U^\tau.
\end{aligned}$$

8.9 Numerical Results

In order to verify the applicability of the theoretical derivations of the previous sections we implemented the process model from Problem 3, and the gradient and objective evaluation routines from Sect. 8.7 in a C++ environment called *Lamatto*. We chose the full implicit scheme for the computation of the state and adjoint equation, e.g. $\alpha = 1$. For descending we used a projected gradient algorithm with Armijo line search (see [7, pages 58 and 66]), which is chosen for its simple structure and easy implementation.

With this tools at hand we run several simulations and generated an optimal control strategy for the academical network introduced in Sect. 7.3. First, we verified the quality of our process model by comparing the hydrographs of *Lamatto* with the hydrographs of the process model *SWMM*, which is introduced in Sect. 7.2. Figure 8.3 shows the hydrographs of the flow rate entering the storage basin B02.

We see that the flow rates generated by *SWMM* and *Lamatto* differ in the following way: At high inflowing loads, the peaks of the *Lamatto* flow are significantly lower than the peaks of the *SWMM* flow. We can explain this different behavior by looking at the flows through the branches S01 and S02, which merge together

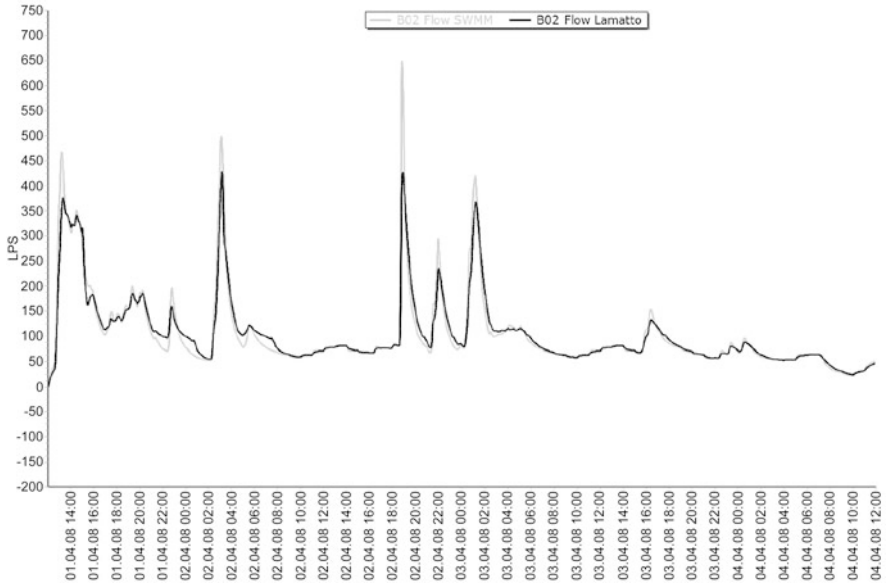


Fig. 8.3 Flow entering B02 with *SWMM* and *Lamatto*

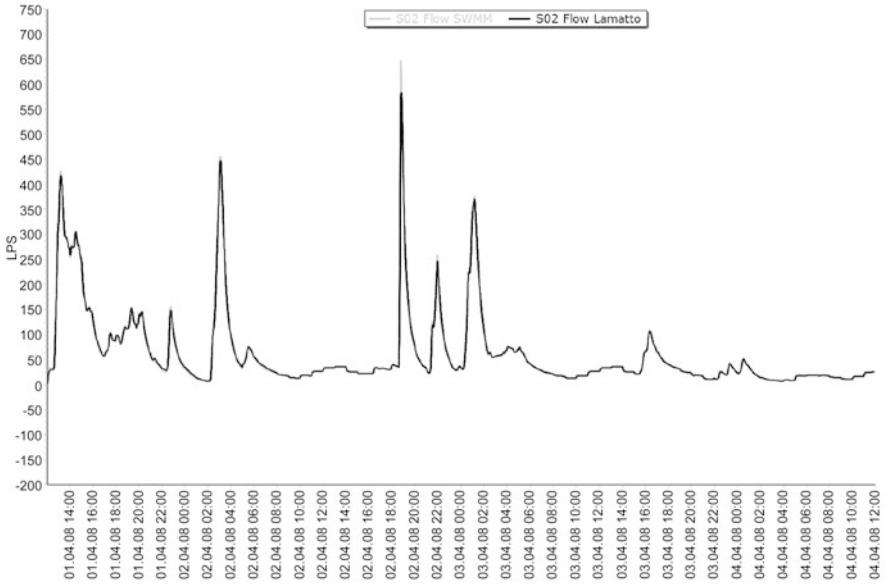


Fig. 8.4 Flow of S02 branch with *SWMM* and *Lamatto*

to form the flow entering B02. In Fig. 8.4 we see the hydrographs of the flows of branch S02 and observe that there are only small differences at high loads between

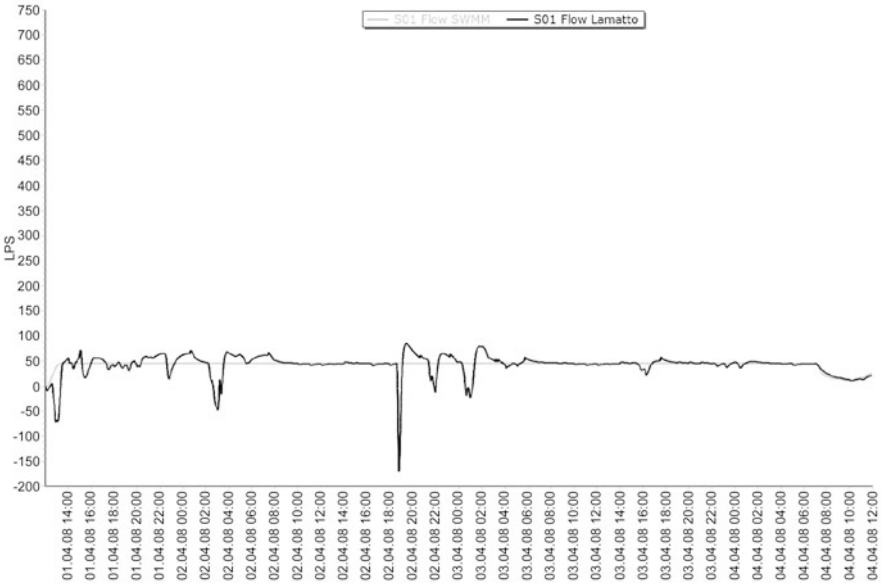


Fig. 8.5 Flow of S01 branch with *SWMM* and *Lamatto*

SWMM and *Lamatto*. But in Fig. 8.5, depicting the flow of the branch S01, the two process models show different behavior:

SWMM just adds the constant pump flow originating from B01 to the branch flow in S02 to generate the flow for storage basin B02. *Lamatto* instead splits the high load entering from branch S02 into two parts, where one part is running into storage basin B02 and the other one spreads *backwards* in upstream direction along branch S01. This effect is also known as *backwater effect*.

These differences are caused by the numerical solution methods and the way the methods handle flow rates in a conduit. The finite volume method used in Lamatto calculates multiple flow rates at different locations within a conduit: Flow rates at the pipe inflow and the pipe outflow are given as well as average flow rates. It is therefore capable to consider backwater effects independent of the spatial discretization (as seen in Fig. 8.5 which shows the flow rates at the end of the pipe directly before the junction). The finite difference method in SWMM, in contrast, uses only one flow rate for each conduit representing the average flow rate in the pipe. Since the spatial discretization is coarse in this case study the conduit length is too large to account for backwater effects.

It is obvious that both process models, *SWMM* and *Lamatto*, show different computational behavior, especially for high loads. We therefore cannot compare the minimal overflow costs resulting from the optimization directly, but have to rely on relative comparison. As both process models generate flow states with a equidistant time step of 60 second, we added up the flow rates leaving the system at B01 and B02 at each time step to evaluate the cost function and then we used the optimization techniques to generate optimal controls in the receding horizon setting introduced

Table 8.1 Comparison of optimization performance of *SWMM* and *Lamatto*

Setting	Costs in B01	Costs in B02	Total costs
SWMM, U_R	231 m^3	1127 m^3	1358 m^3
SWMM, U_S	7 m^3	835 m^3	842 m^3
Lamatto, U_R	231 m^3	887 m^3	1118 m^3
Lamatto, U_L	231 m^3	499 m^3	730 m^3

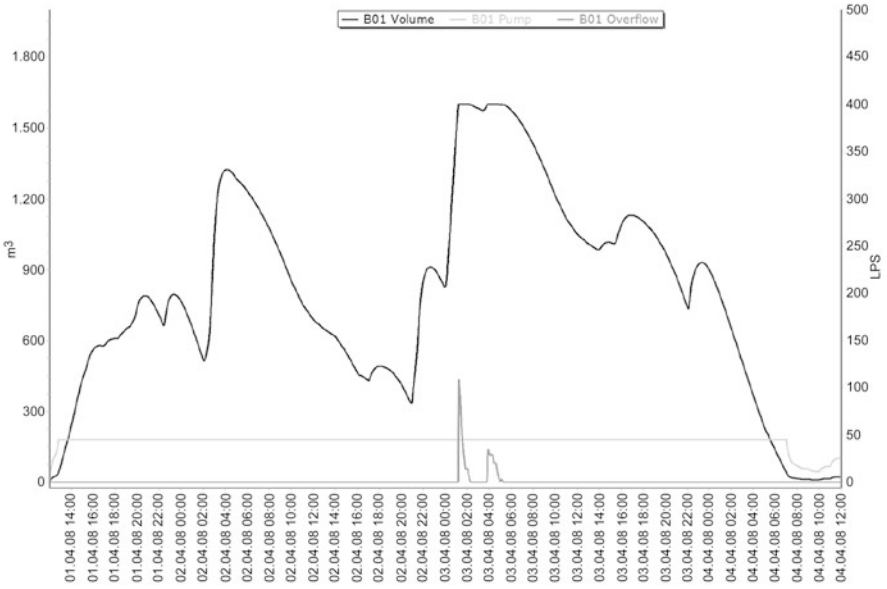


Fig. 8.6 Volume, pump flow and overflow in basin B01 with reference control

in Sect. 7.2. The results are presented in Table 8.1. The controls are named as follows: U_R is a reference control consisting of a pump with a constant pump rate of 45 l/s, which is chosen as an initial guess with respect to practical experience from the engineers. U_S is the control generated with the *SWMM* process model and likewise U_L is generated with the *Lamatto* process model. From the first two lines of Table 8.1 we conclude that the control U_S reduces the costs in the *SWMM* process model by 38 %.

Lets look at the last two lines: For the default control U_R the costs at conduit S212 computed with *Lamatto* are less than those computed from *SWMM*, which is a consequence of the backwater effect. The optimal control U_L for *Lamatto* reduces these costs by 35 %. An overview of the flow states generated with *Lamatto* in the controllable storage basin B01 and basin B02 is given in Figs. 8.6, 8.7, 8.8, 8.9, where volume, pump flow and overflow at both storage basins B01 and B02 is depicted for the reference and optimal control.

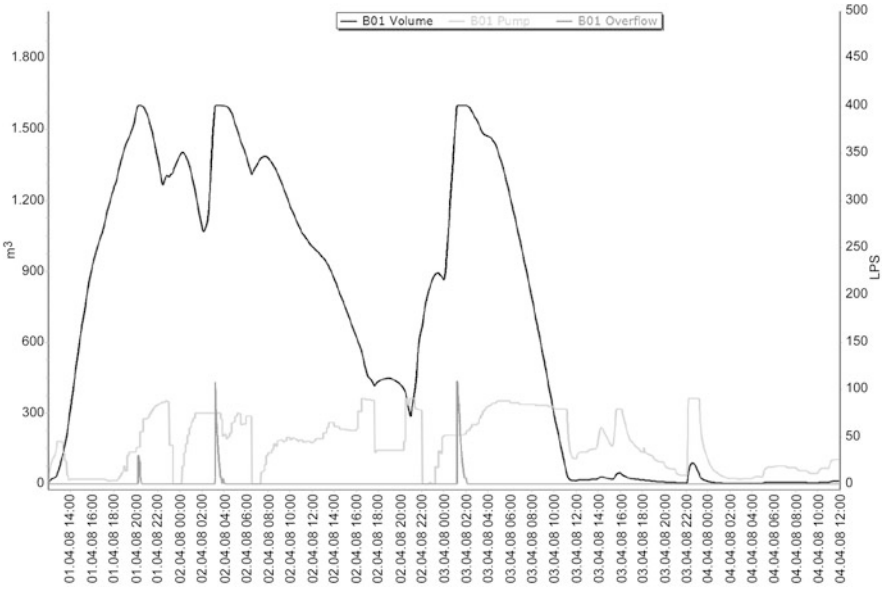


Fig. 8.7 Volume, pump flow and overflow in basin B01 with optimal control

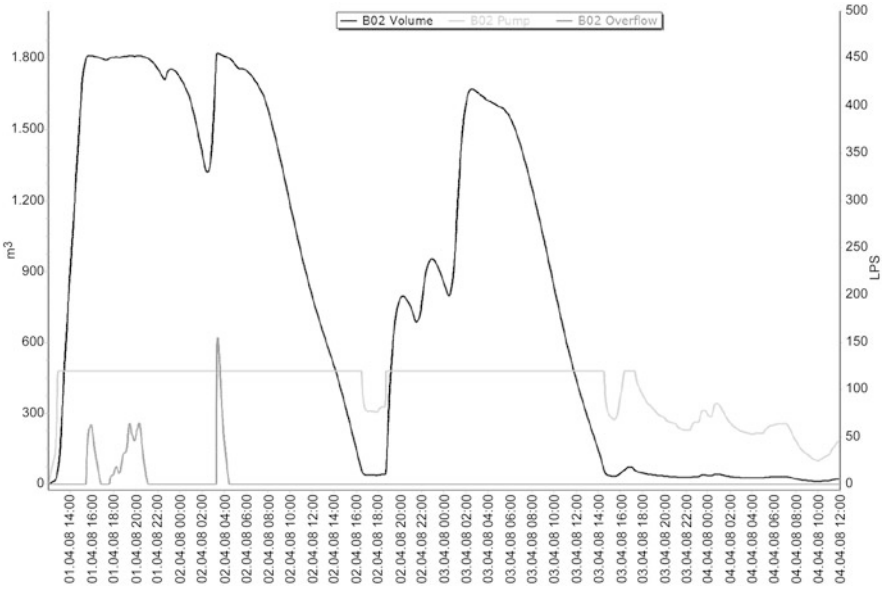


Fig. 8.8 Volume, pump flow and overflow in basin B02 with reference control

The comparison in Table 8.1 suggests that both *SWMM* and *Lamatto* manage to reduce the costs significantly in comparison to the reference control. But one

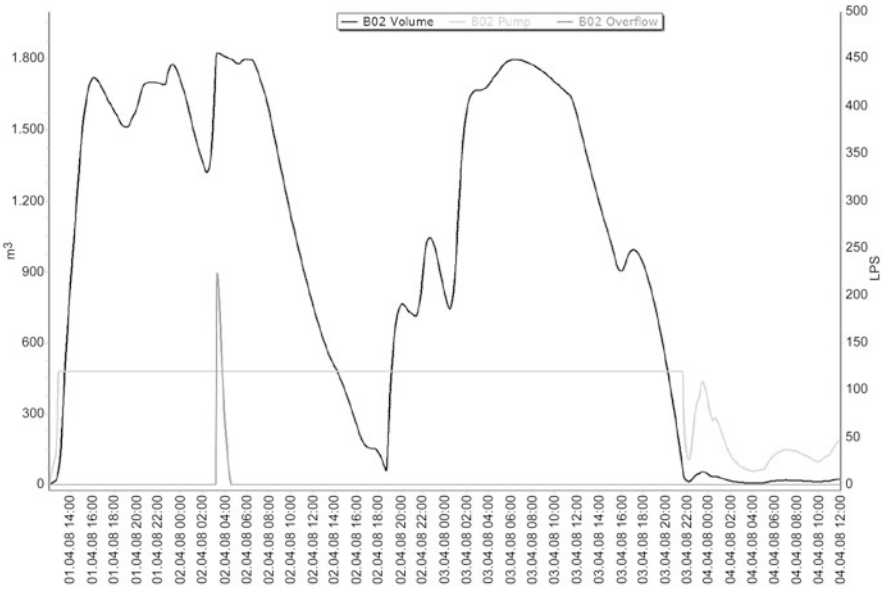


Fig. 8.9 Volume, pump flow and overflow in basin B02 with optimal control

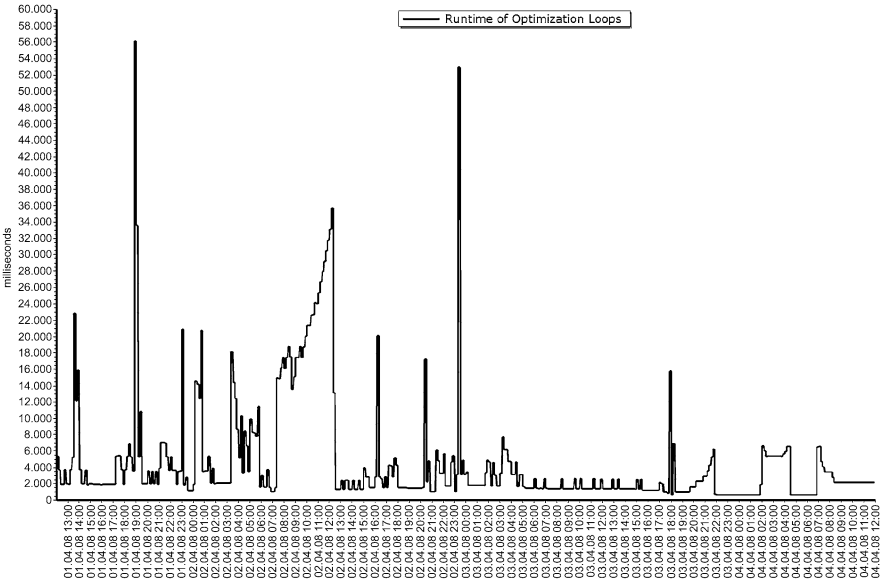


Fig. 8.10 Runtime of optimization loops for *Lamatto*

important property of *Lamatto* that has to be mentioned is the performance in runtime: While the derivative-free optimization approach used in Sect. 7.3 needed 135

minutes to generate U_S, *Lamatto* finished the computation within 34 minutes. One reason for the better runtime performance of *Lamatto* lies in the practical application of Pontryagin's minimum principle: There are many cases, where (8.66) and (8.67) are true for all components of the gradient and in this case the current optimization loop terminates at once, as a locally optimal solution is found. This behavior is depicted in Fig. 8.10.

Of course these results give only a first impression of the different performances of *SWMM* and *Lamatto* and general statements cannot be given, but we are highly motivated to compare both approaches on the larger, realistic network.

References

1. M. Athans, P.L. Falb, *Optimal Control: An Introduction to the Theory and Its Applications* (Dover, New York, 1966)
2. F. Brauer, J.A. Nohel, *Ordinary Differential Equations (A First Course)* (Benjamin, New York, 1967)
3. J. Jahn, *Introduction to the Theory of Nonlinear Optimization* (Springer, Berlin, 1994)
4. P. Lax, B. Wendroff, Systems of conservation laws. Commun. Pure Appl. Math. **13**(2), 217–237
5. E.B. Lee, L. Markus, *Foundations of Optimal Control Theory* (Wiley, New York, 1967)
6. R.J. Leveque, *Finite Volume Methods for Hyperbolic Problems*. Cambridge Texts in Applied Mathematics (2002)
7. E. Polak, *Optimization: Algorithms and Consistent Approximations* (Springer, New York, 1997)
8. E.F. Toro, *Shock-Capturing Methods for Free-Surface Shallow Flows* (Wiley, New York, 2001)

J. Hild · G. Leugering
Institut für Angewandte Mathematik (Lehrstuhl II), Friedrich-Alexander-Universität Erlangen-Nürnberg, Cauerstr. 11, 91058 Erlangen, Germany

J. Hild
e-mail: hild@am.uni-erlangen.de

G. Leugering (✉)
e-mail: guenter.leugering@am.uni-erlangen.de

Chapter 9
Performance and Comparison of *BlueM.MPC* and *Lamatto*

Steffen Heusch, Johannes Hild, Günter Leugering, and Manfred Ostrowski

Abstract We compare the quality and generation performance of the optimal control sequence produced by the software frameworks *BlueM.MPC* and *Lamatto*.

9.1 Comparison and Conclusion

In the preceding chapters two software frameworks for the optimal control of sewer networks with dynamic models, *BlueM.MPC* (which implements SWMM as process model) and *Lamatto*, were introduced. Results for the application of both frameworks to an academical network were already compared in Sect. 8.9. In this chapter, results for a realistic network are presented.

9.1.1 Realistic Network Case Study

The following case study represents a combined sewer network of a small city in Germany with 40.000 inhabitants. The network drains a total area of 526 ha with an impervious area of 279 ha. It comprises 15 storage and overflow structures with a total storage volume of 12.700 m^3. In order to reduce computation times a surrogated system was applied for MPC simulations, i.e. the number of junctions and conduits was reduced without falsifying the flow behavior of the network. Furthermore, those branches of the network that are not effected by control decisions were not considered, leading to an even smaller network. An overview of the system used for the MPC calculations is given in Fig. 9.1.

For the objective function we sum up the overall overflow at the system exits BN01, BN02, BN03, BS01, BS02 and BTP. Pump rates at three storage basins (BS01, BS02 and BN01) can be controlled. Table 9.1 shows the pump configurations used for optimal control. The case study is based on a four day simulation period. The results for calculations with both software frameworks are given in Table 9.2, whereas U_R specifies the time constant reference control resulting from heuristics, U_L specifies the optimal control generated by *Lamatto* and likewise U_S is the optimal control computed by *SWMM*. We observe again, that the *Lamatto* process model generates lower discharge costs for the reference control than *SWMM*. This is caused by stronger backwater effects and is discussed in Sect. 8.9.

A. Martin et al. (eds.), *Mathematical Optimization of Water Networks*,
International Series of Numerical Mathematics 162,
DOI 10.1007/978-3-0348-0436-3_9, © Springer Basel 2012

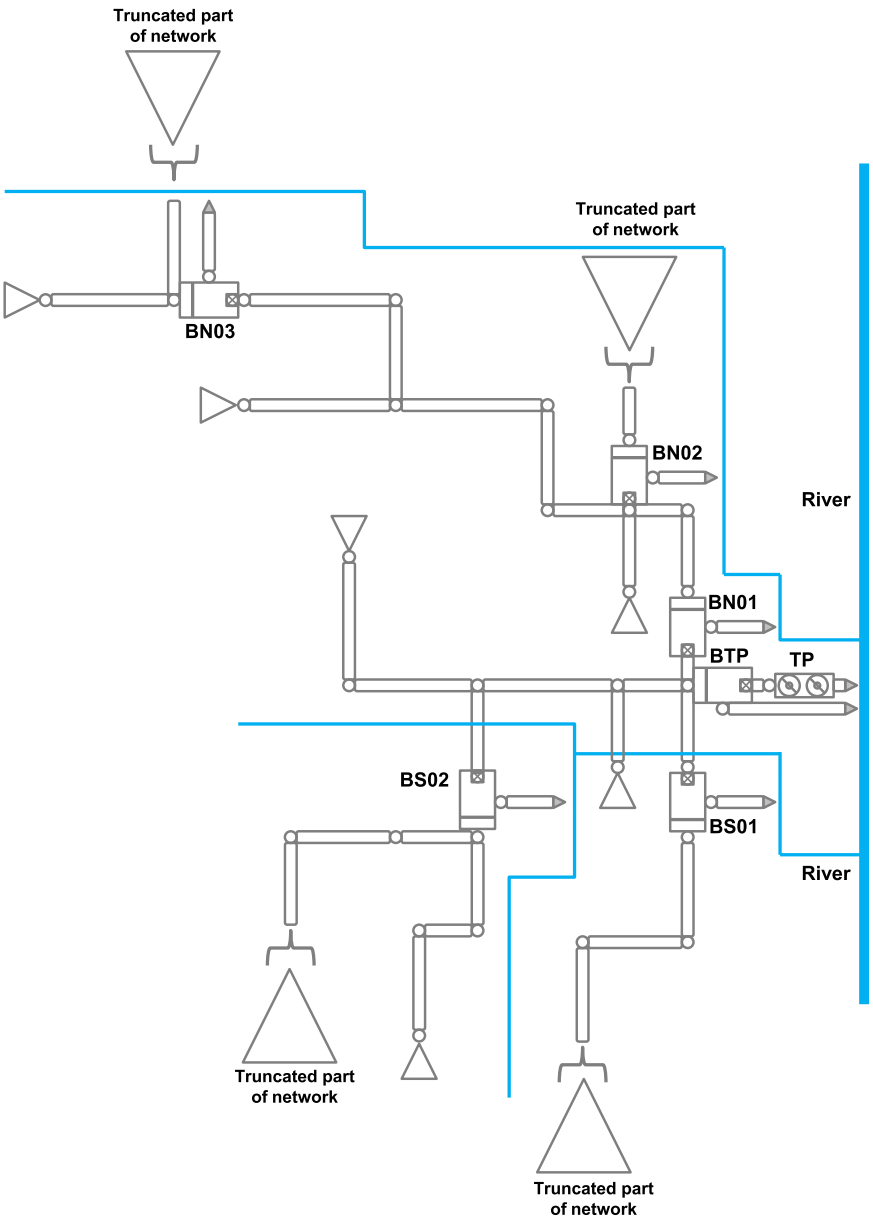


Fig. 9.1 Network of case study 2

For the receding horizon strategy we applied an evaluation horizon of 2 hours and assumed that all inflow predictions are known exactly. The control step was set to 10 minutes leading to 576 optimization loops. For the optimization with *BlueM.MPC* the Hooke and Jeeves method (see Sect. 7.2.3) was used in this case.

Table 9.1 Storage basin and pump control configuration

Storage basin	BS01	BS02	BN01	BN02	BN03	BTP
Volume	700 m^3	1275 m^3	144 m^3	60 m^3	780 m^3	900 m^3
Reference pump rate	60 l/s	80 l/s	750 l/s	410 l/s	100 l/s	250 l/s
Minimal pump rate	0 l/s	0 l/s	0 l/s	–	–	–
Maximal pump rate	90 l/s	120 l/s	1500 l/s	–	–	–

Table 9.2 Comparison of discharge of *SWMM* and *Lamatto*

Setting	BS01	BS02	BN01	BN02	BN03	BTP	Total
SWMM, U_R	956 m^3	155 m^3	0 m^3	0 m^3	412 m^3	7314 m^3	8837 m^3
SWMM, U_S	2379 m^3	92 m^3	0 m^3	0 m^3	412 m^3	4896 m^3	7779 m^3
Lamatto, U_R	711 m^3	0 m^3	0 m^3	0 m^3	0 m^3	6854 m^3	7565 m^3
Lamatto, U_L	1111 m^3	904 m^3	0 m^3	0 m^3	0 m^3	1329 m^3	3344 m^3

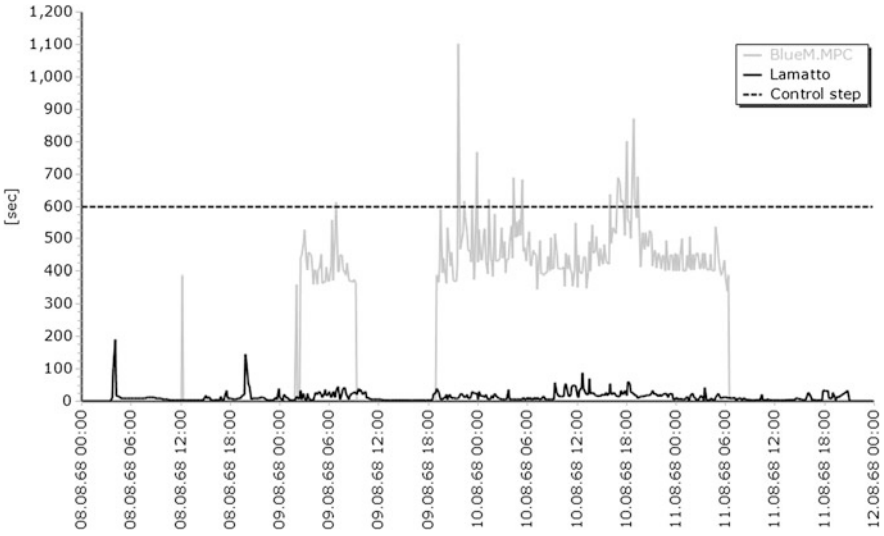


Fig. 9.2 Runtime of optimization loops for *Lamatto* and *BlueM.MPC* (Hooke and Jeeves)

The evaluation of the results is based on the control quality (the ability to reduce the discharge volumes) and computation times. *BlueM.MPC* generates an optimal control within an overall computation time of 595 minutes for all 576 loops while reducing the total costs by 12 %. *Lamatto* requires an overall computation time of 127 minutes while reducing the total costs by 56 %. Runtimes of all loops for both applications are depicted in Fig. 9.2. The hydrographs of the corresponding

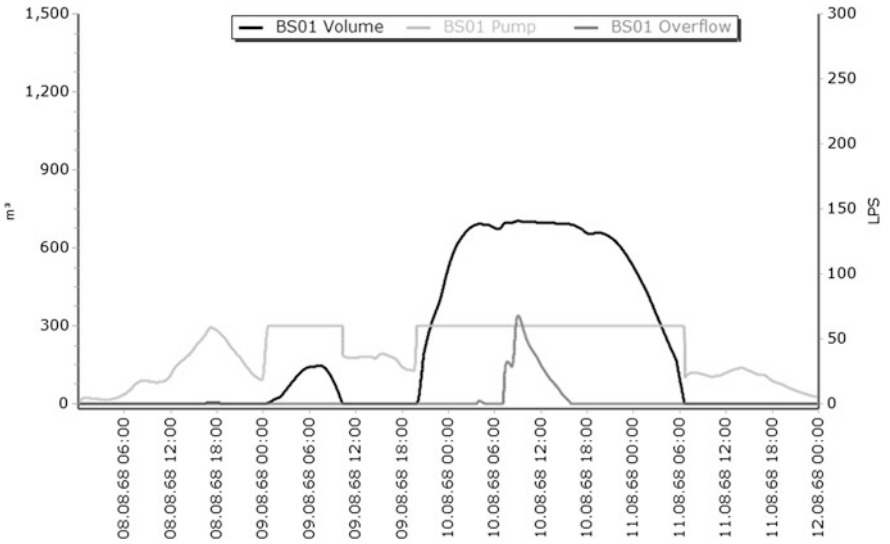


Fig. 9.3 Volume, pump flow and overflow in basin BS01 with reference control (SWMM)

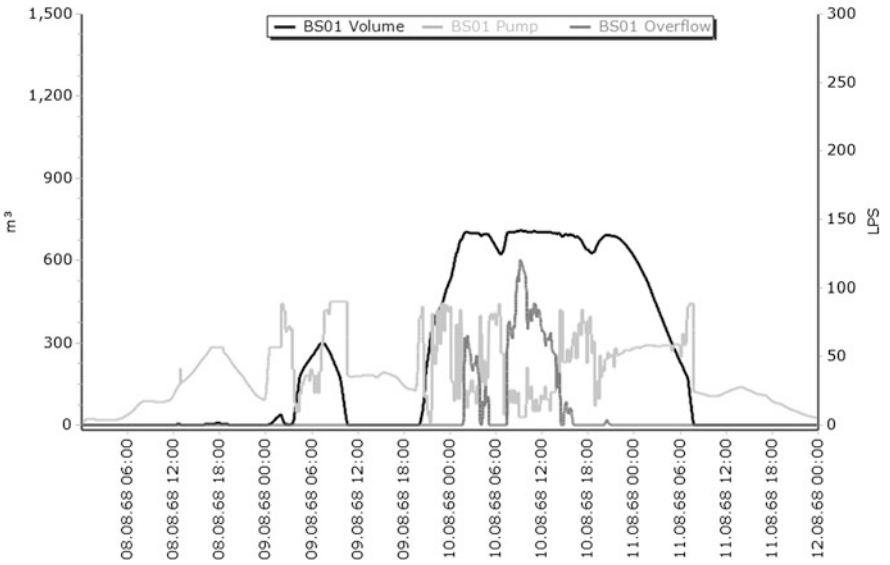


Fig. 9.4 Volume, pump flow and overflow in basin BS01 with optimal control (SWMM)

volumes and flow rates at different storage basins are depicted in Figs. 9.3 to 9.10 for computations with *BlueM.MPC* and in Figs. 9.11 to 9.18 for computations with *Lamatto*.

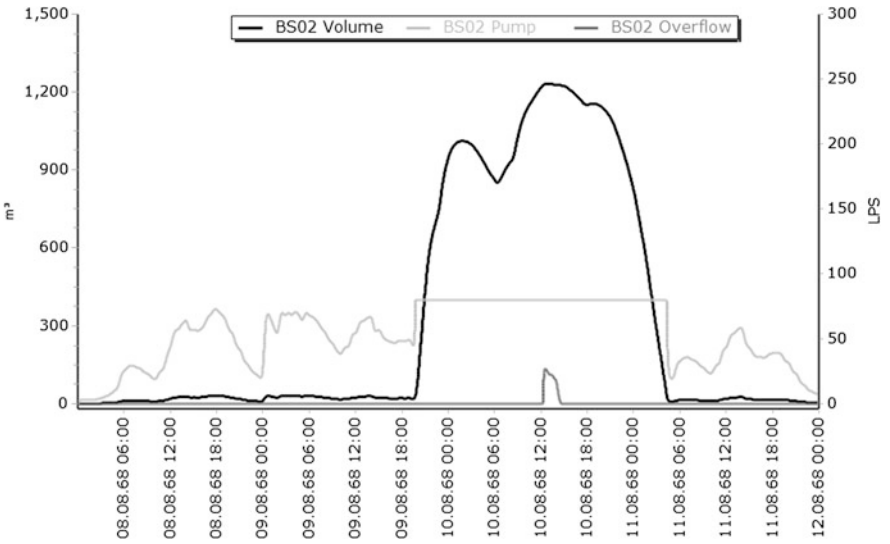


Fig. 9.5 Volume, pump flow and overflow in basin BS02 with reference control (SWMM)

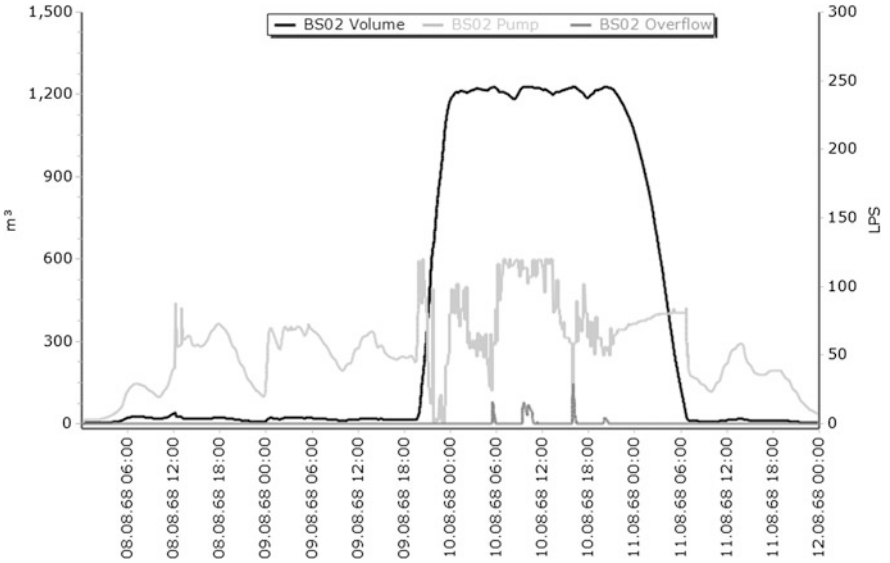


Fig. 9.6 Volume, pump flow and overflow in basin BS02 with optimal control (SWMM)

The maximum duration for a single control loop with *Lamatto* is less than 200 seconds. The software is therefore capable to calculate the control decision within the given time of the control step (600 seconds). *BlueM.MPC* takes significantly longer than *Lamatto*. The average duration for a single control loop is longer than

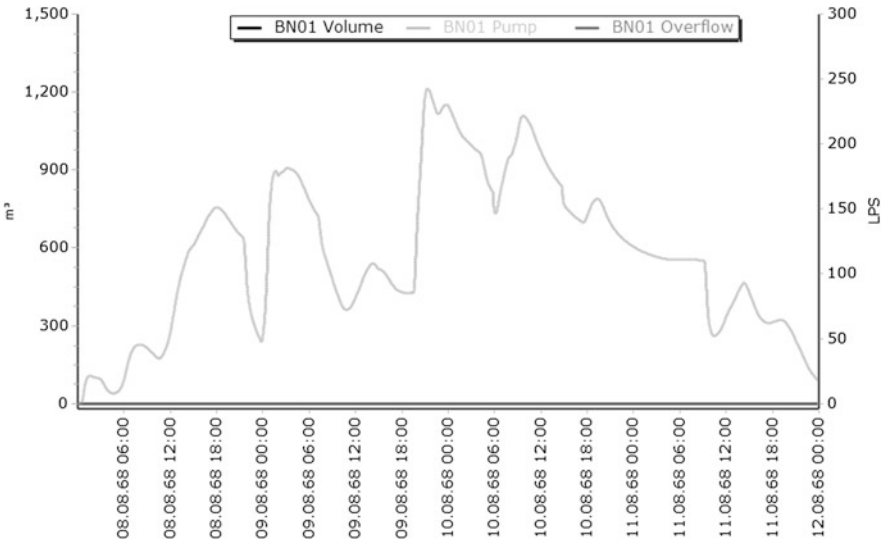


Fig. 9.7 Volume, pump flow and overflow in basin BN01 with reference control (SWMM)

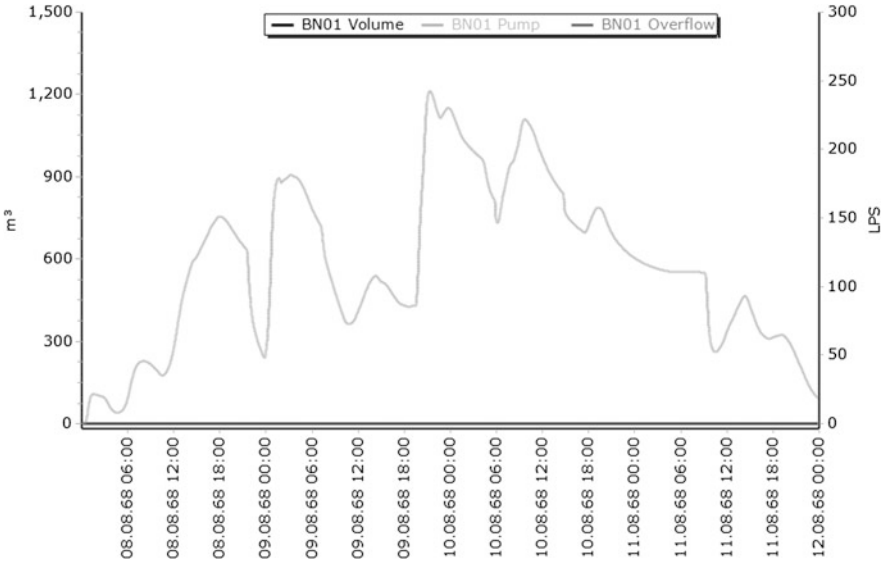


Fig. 9.8 Volume, pump flow and overflow in basin BN01 with optimal control (SWMM)

400 seconds. In several cases the software takes longer than 600 seconds to compute the control decisions.

In order to evaluate the quality of the optimized control decisions a simple comparison of the total discharge volumes is not possible since the discharges differ

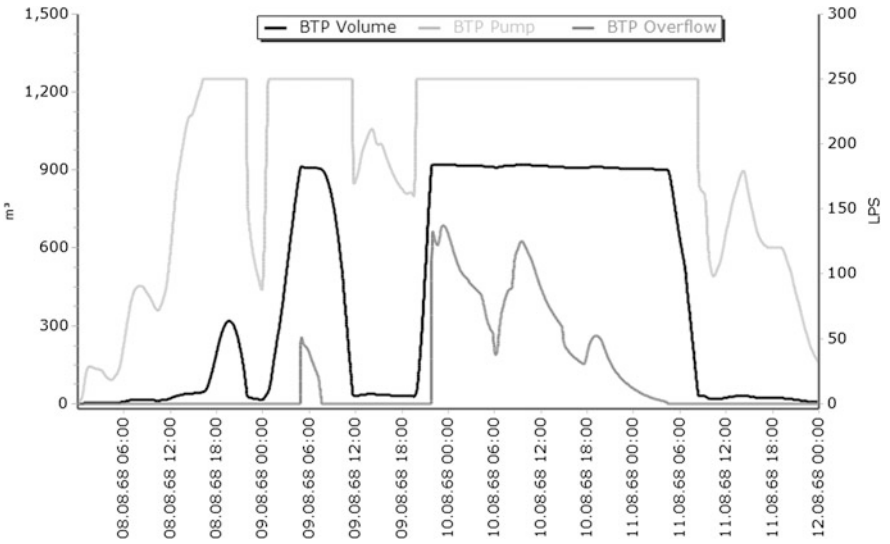


Fig. 9.9 Volume, pump flow and overflow in basin BTP with reference control (SWMM)

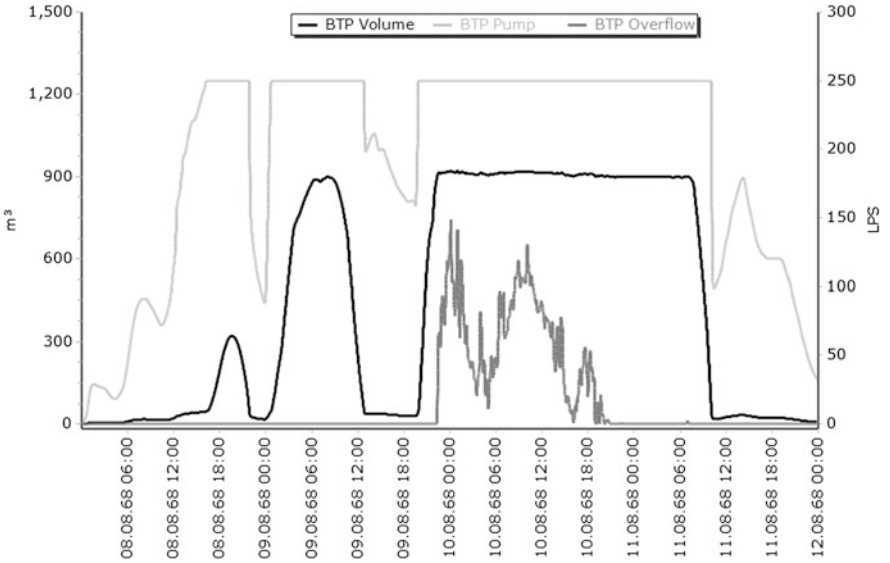


Fig. 9.10 Volume, pump flow and overflow in basin BTP with optimal control (SWMM)

already significantly for the reference control. However, the hydrographs of the storage basin volumes show, that optimizations from both frameworks hold back water in basins BS01 and BS02, where occasional discharge occurs. The advantage of this behavior is a strong reduction of the discharge costs in the last storage basin BTP.

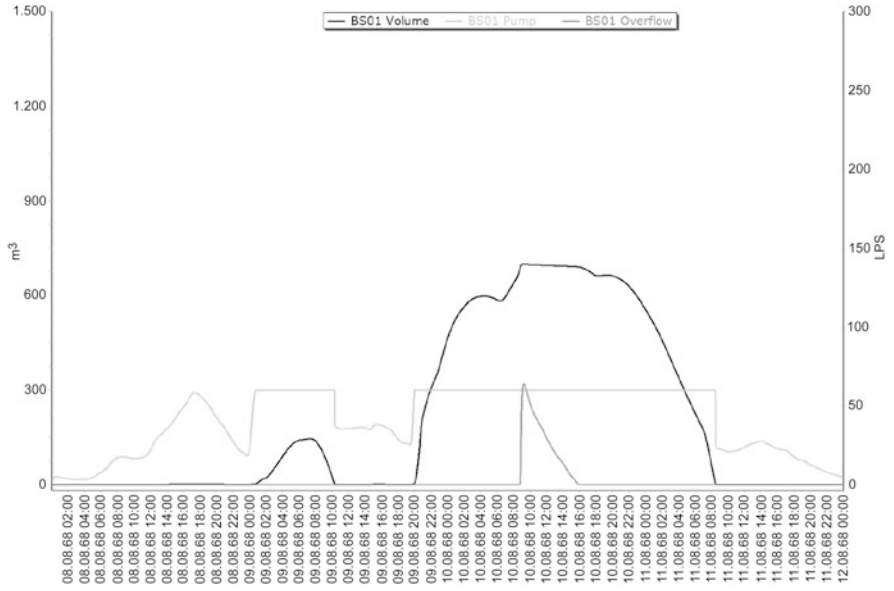


Fig. 9.11 Volume, pump flow and overflow in basin BS01 with reference control (Lamatto)

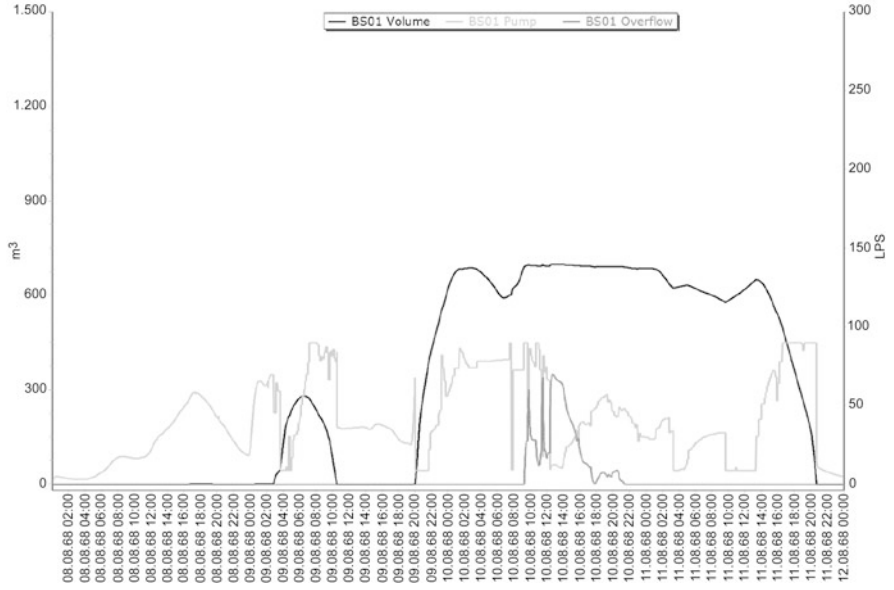


Fig. 9.12 Volume, pump flow and overflow in basin BS01 with optimal control (Lamatto)

Lamatto additionally holds back water in basin BN01 and determines therefore an even better result.

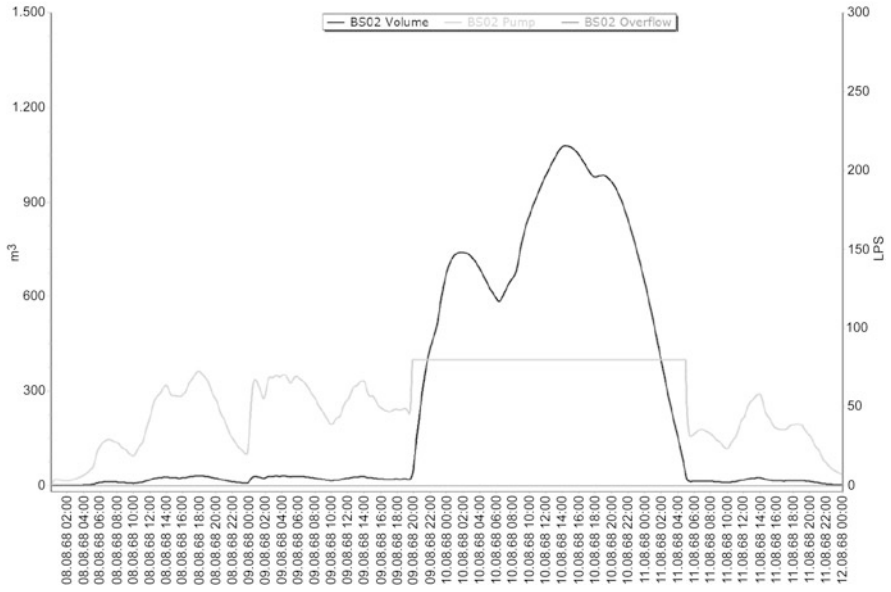


Fig. 9.13 Volume, pump flow and overflow in basin BS02 with reference control (Lamatto)

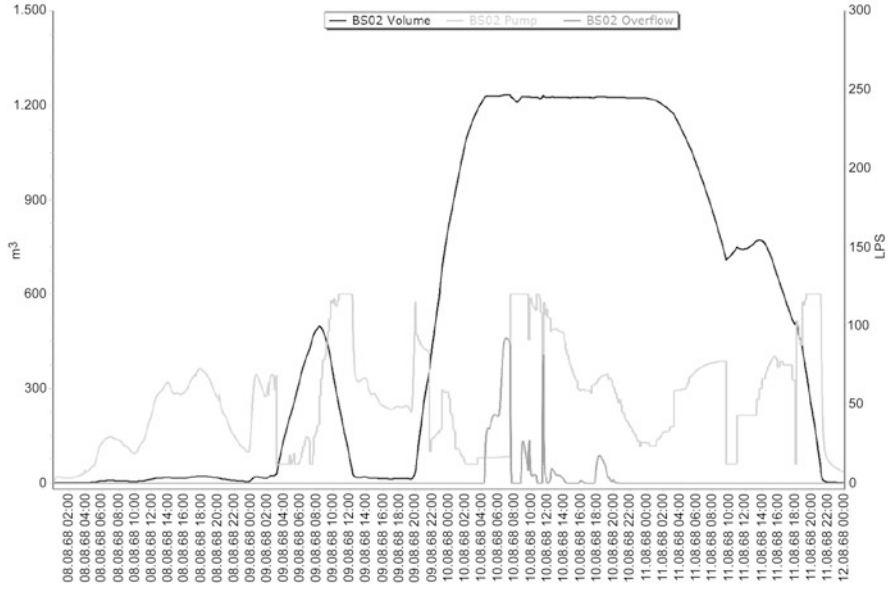


Fig. 9.14 Volume, pump flow and overflow in basin BS02 with optimal control (Lamatto)

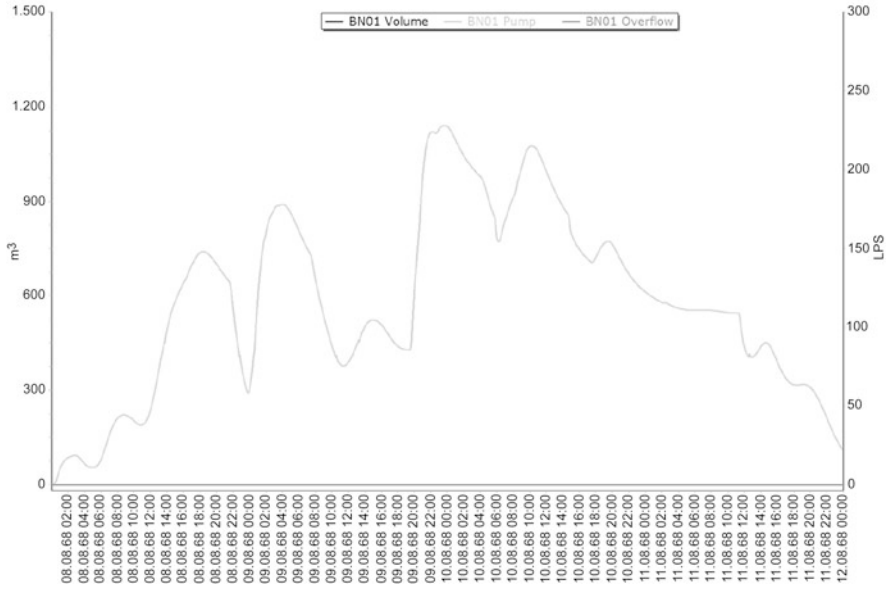


Fig. 9.15 Volume, pump flow and overflow in basin BN01 with reference control (Lamatto)

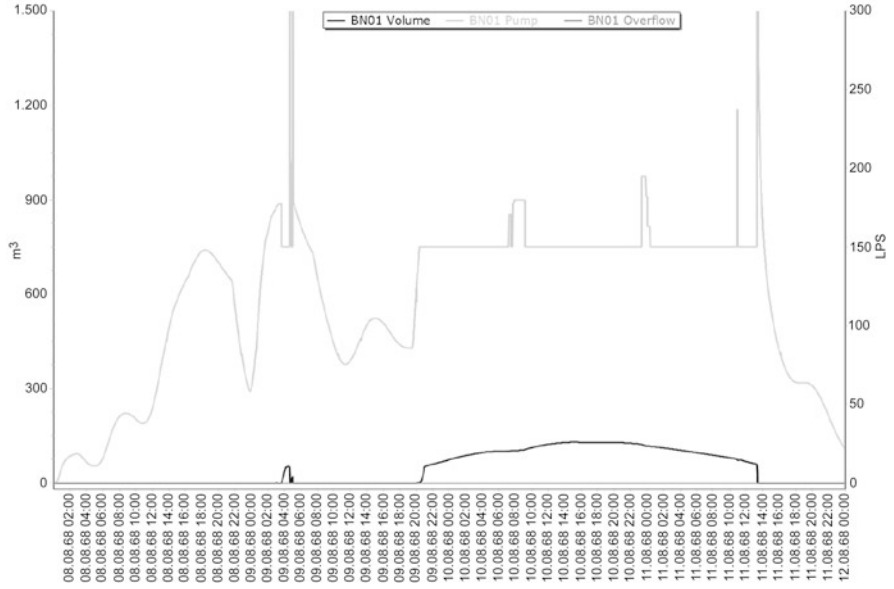


Fig. 9.16 Volume, pump flow and overflow in basin BN01 with optimal control (Lamatto)

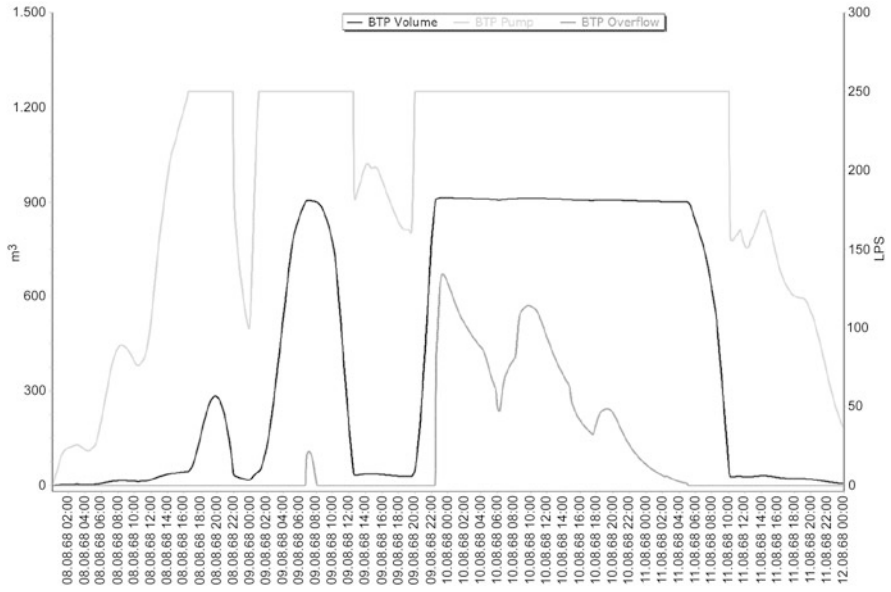


Fig. 9.17 Volume, pump flow and overflow in basin BTP with reference control (Lamatto)

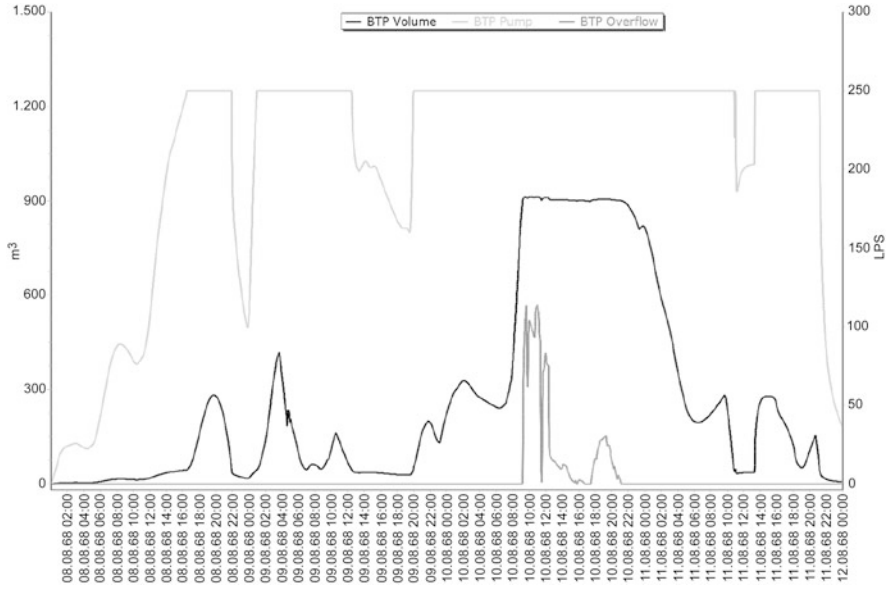


Fig. 9.18 Volume, pump flow and overflow in basin BTP with optimal control (Lamatto)

Table 9.3 Comparison of discharge and runtime for different horizon settings

Setting	Horizon	Control step	Loops	Total costs	Runtime
Lamatto	1 hour	10 minutes	576	5438	48 minutes
Lamatto	2 hours	5 minutes	1152	3619	195 minutes
Lamatto	2 hours	10 minutes	576	3344	127 minutes
Lamatto	2 hours	15 minutes	384	6329	103 minutes
Lamatto	3 hours	10 minutes	576	3182	211 minutes

9.1.2 Influence of Time Horizons

In the following experiment we used Lamatto to vary the observation horizon and the control step of the original setting to get an impression of the sensitivity of the discharge costs with respect to these input parameters. The results are presented in Table 9.3. Based on the original setting in line 3 we observe that a variation of the observation horizon as done in lines 1 and 5 affects both, runtime and cost reduction, as expected. The setting in line 2 on the other hand is surprisingly inefficient: Although we doubled the number of control variables and therefore provide more flexibility for the solution, we got a worse result than in the original setting. This behavior confirms the fact, that we can only find local minima of the optimization subproblems. In line 4 we expanded the control step and as expected this modification leads to a worse result.

9.1.3 Influence of Optimization Algorithm

In the next experiment we used BlueM.MPC to apply two different optimization methods. In addition to the local search algorithm from Hooke and Jeeves we applied the evolutionary strategies (see Sect. 7.2.3). Computations were performed with a control step of 10 minutes and two evaluation horizons, the resulting discharge costs at the system exits are shown in Table 9.4. With an evaluation horizon of two hours the evolutionary strategies compute higher discharges (8.635 m^3) than the Hooke and Jeeves method (7.779 m^3). When the evaluation horizon is increased to three hours, the Hooke and Jeeves method improves the result only slightly to 7.768 m^3 whereas the evolutionary strategies decrease the discharge volumes significantly to 6.499 m^3.

The results confirm the expected behavior of both algorithms. The local algorithm of Hooke and Jeeves gives reliable results. The global evolutionary algorithms are able to deliver better results but these cannot be guaranteed. Because of this unreliability, one would prefer the algorithm of Hooke and Jeeves. On the other hand, the evolutionary algorithms have the advantage, that the number of objective function evaluations is determined by the number of generations and their offsprings which are chosen by the user. It is therefore possible to control the number of objective function evaluations and to make sure, that their computation does not exceed the available time for a control step. It is the user's task to assess for the given

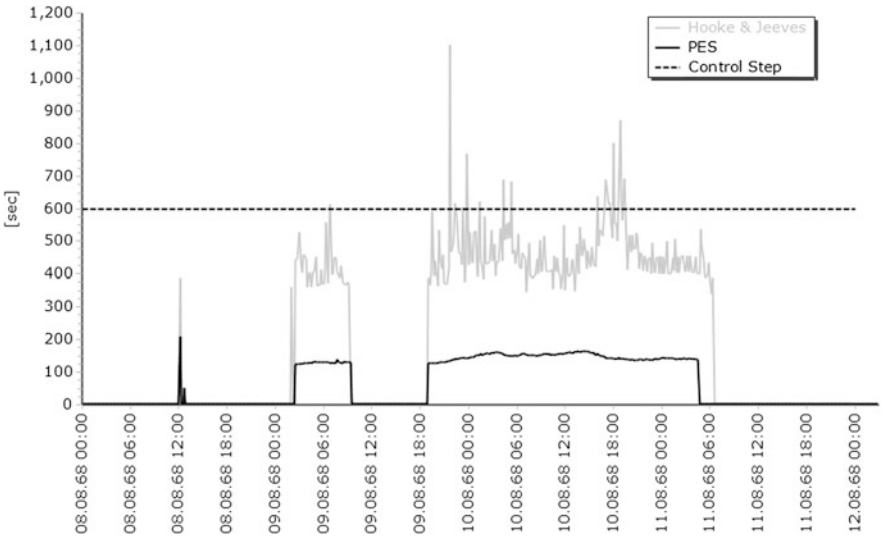


Fig. 9.19 Runtime of optimization loops for *BlueM.MPC*

Table 9.4 Comparison of discharge costs for different optimization algorithms generated with *BlueM.MPC*

Algorithm	BS01	BS02	BN01	BN02	BN03	BTP	Total
HJ (2 hrs)	2379 m^3	92 m^3	0 m^3	0 m^3	412 m^3	4896 m^3	7779 m^3
PES (2 hrs)	708 m^3	556 m^3	5895 m^3	0 m^3	412 m^3	1064 m^3	8635 m^3
HJ (3 hrs)	1490 m^3	541 m^3	0 m^3	0 m^3	412 m^3	5325 m^3	7768 m^3
PES (3 hrs)	1045 m^3	684 m^3	1500 m^3	0 m^3	412 m^3	2858 m^3	6499 m^3

system, if the control horizon is long enough to enable sufficient control decisions. Figure 9.19 serves as explanation: It shows the runtimes for both algorithms for the 2 hour evaluation horizon. The runtime of the evolutionary algorithm is almost constant, minor differences are due to the computation load of other applications on the PC. The number of generations and their offsprings is low. In order to keep computation times below the control horizon only 10 generations and 10 offsprings were selected. In contrast to the algorithm of Hooke and Jeeves, the time limitations of the control horizon can be handled easily but the discharge costs are higher.

9.1.4 Conclusion

In the preceding sections we presented two process models for urban drainage modeling. The first one, *SWMM*, is an established routing software frame, applied by many engineers all over the world. The second one, *Lamatto*, is modeled and im-

plemented in the context of this project and has been tailored for derivative-based optimal control of urban drainage systems. Right from the beginning we demanded that *Lamatto* is at least competitive to *SWMM* in both, simulation and optimization performance.

This goal is achieved in the sense that *Lamatto* can handle simulation tasks as quickly as *SWMM*, but we have to admit that both process models give different answers at channel junctions: While *Lamatto* detects short living shock waves running back in junctions with small load, *SWMM* tends to ignore backwater effects at these channel junctions. This different behavior results in different discharge costs at the storage basins and leads to a different weighting of control decisions at these basins. It is not clear, which of these two process models is a better approximation of the real physical behavior and which suits better for practical application of real time control of urban drainage systems. All these questions can only be answered, if the results of simulation and optimization of both process models are compared with measurements from real physical sewer systems.

The results of the optimization approaches, which were done for the academical network in Sects. 7.3 and 8.9 and for the realistic network in the last section, are satisfying in terms of quality: All optimization approaches found the control potential hidden in the test cases and all of them reduced the discharge costs significantly. But as expected, the gradient-based algorithms working with the *Lamatto* process model where much faster in terms of runtime performance. This improvement in runtime motivates further research of the gradient-based approach, which is still under development.

Lamatto is written as a scientific application, with the focus on modular modifiability and easy-to-understand design. But it is not written with focus on memory management and optimal runtime performance, which is recommended for practical applications. Furthermore the applied optimization tool, the projected gradient method, is mainly chosen for the easy access and direct observability of theoretical behavior like complementary slackness and gradient convergence, but lacks the better performance of more sophisticated and specialized optimization algorithms. Again, for practical applications it is recommended to replace the projected gradient method with improved optimization algorithms.

Last but not least we have to point out that the settings of the receding horizon approach, like evaluation horizon and number of control variables per hour, are chosen intuitively. As these parameters affect the runtime performance and quality of a solution, it is recommended to invest more research time in the optimal assignment of these settings for each specific network.

S. Heusch · M. Ostrowski
Ingenieurhydrologie und Wasserbewirtschaftung, Technische Universität Darmstadt,
Petersenstr. 13, 64287 Darmstadt, Germany

S. Heusch
e-mail: heusch@ihwb.tu-darmstadt.de

M. Ostrowski
e-mail: ostrowski@ihwb.tu-darmstadt.de

J. Hild · G. Leugering
Institut für Angewandte Mathematik (Lehrstuhl II), Friedrich-Alexander-Universität Erlangen-Nürnberg, Cauerstr. 11, 91058 Erlangen, Germany

J. Hild
e-mail: hild@am.uni-erlangen.de

G. Leugering (✉)
e-mail: guenter.leugering@am.uni-erlangen.de

Chapter 10
Multicriteria Optimization in Wastewater Management

Kerstin Dächert and Kathrin Klamroth

Abstract In this chapter we consider the goals and objectives arising in wastewater management in the context of a multiobjective analysis. This allows, among others, the individual consideration of (1) the overflow volume (i.e., the total amount of released water), (2) the pollution load in the released water, and (3) the cost of the generated control. Given a specific sewage network and data of typical inflow scenarios, a multiobjective offline analysis of the problem and, in particular, of the trade-off between the different goals provides the decision maker with valuable information of the problem characteristics. This information can then be used to specify a suitable scalarized, single-objective optimization problem for the real-time optimal control that represents the decision makers preferences in a best possible way. If an efficient solver for such scalarizations is available (which is the case for the problems considered here), this leads to an efficient online procedure that is justified by an extensive offline problem analysis.

Even though the methods presented in this chapter were tailored for wastewater management problems, they are also applicable in the context of the other applications mentioned in this volume.

10.1 Introduction

In most real-world optimization problems, there is not only one single objective to be considered but a multitude of them. Typically these multiple objectives are conflicting, which means that the solutions in which the individual objective functions attain their optimum are different from each other. In this case we speak of a multiobjective or multicriteria optimization problem. Instead of finding one single global optimal solution as in a classical (single-objective) optimization problem we look for a set of so-called Pareto optimal solutions. These are solutions which cannot be improved with respect to one of the objective functions without being impaired with respect to at least one other objective function.

Wastewater management problems are typical examples for problems involving multiple goals. This includes the minimization of overflows and the pollution in such overflows as well as the minimization of costs incurred by excessive variations in the control and in the inflow to the wastewater treatment plant. The individual consideration of different objectives in the context of a multicriteria analysis can

A. Martin et al. (eds.), *Mathematical Optimization of Water Networks*,
International Series of Numerical Mathematics 162,
DOI 10.1007/978-3-0348-0436-3_10, © Springer Basel 2012

provide the decision maker with important trade-off information for the selection of a most preferred solution or control. While in the context of an online control process as described in Chap. 8 a complete multicriteria analysis is in general too time consuming, a preliminary, scenario-based offline investigation in interaction with the engineers can be very useful to determine a suitable scalarization that is then used for the online control. The main goal in this situation is to obtain an efficiently solvable problem formulation that comprises the multiple objectives into one single objective function and possibly additional constraints, a so-called scalarization of the multiobjective problem.

The remainder of this chapter is organized as follows: In Sect. 10.1 we present a short literature overview focused upon multicriteria optimal control of wastewater management problems and the main objectives arising therein and we specify the objectives considered in our study. Furthermore a brief introduction to the basic concepts, definitions and notations is given in this section. Sect. 10.2 is devoted to methods. We discuss different approaches to deal with multiobjective problems and explain why we chose scalarizing methods to find nondominated solutions in wastewater management problems. We present different scalarization methods known from the literature together with some extensions, and compare their theoretical properties and their practical applicability. We also discuss how to generate an approximation of the set of nondominated solutions. In Sect 10.3 we test the presented methods in the context of wastewater management problems. With the help of several selected scenarios we generate optimal controls obtained by considering different objectives. Already for these small test instances interesting effects occur which are discussed in the following. The section ends with the multicriteria optimization of the network presented in Chap. 7 with three objectives. In the last section we conclude our results and give ideas for further research.

10.1.1 Literature Overview and Goals in Wastewater Management

Wastewater optimal control problems deal with the control of weirs and/or valves and pumps in a sewer network such that certain objectives are met best. In [18] two control approaches are developed for sewer networks. The objective consists of five subgoals, namely the minimization of accumulated volumes in the tunnels (channels), minimization of overflows, maximum utilization of the wastewater treatment plant, obtaining a desired distribution of the reserve storage volume and avoiding abrupt changes of outflows. In [22] the experiences of the Québec urban community are described, where since the year 2000 a real-time optimal control system has been installed and monitored. The system covers the minimization of overflows, the maximization of the use of the wastewater treatment plant capacity and the minimization of accumulated volumes in the tunnels as well as set point variations. Furthermore, the preferential treatment of some overflow sites and the dewatering of the upstream tunnel are included in the system. While in [18] and [22] pollution is not considered as an objective, it is listed as an important goal among the future trends in [27].

Moreover, pollution in the receiving water is considered as one of the objectives in [24] and [31]. The minimization of pollution and economical costs in an estuary is studied in [1]. Based on these references we included the following objectives in our study:

1. Minimization of overflows, i.e. the amount of water which has to be released due to capacity limitations of the network.
2. Minimization of pollution mass in the released water.
3. Minimization of variations of inflow to the wastewater treatment plant (WWTP), as the WWTP works best when inflow is as constant as possible.
4. Minimization of variations of all controllable weirs and pumps in order to get a smooth control profile, i.e. unnecessary and sudden opening and closing of the controllable elements should be avoided.
5. Maximization of inflow to the WWTP such that unnecessary storage of water in the network is prevented.

We assume that the data which is necessary to consider these objectives is available, i.e. especially the network contains sensors to measure the required data online. Concerning pollution load this is an idealized situation as in practice online measurements of the chemical oxygen demand (COD) are usually not available.

The goals can then be formulated mathematically as follows: Let T denote the time steps considered and let S be the number of storage units in the network. Then:

- The minimization of the total release of water is modelled by

$$f_1 = \sum_{i=1}^{S} \sum_{t=1}^{T} Q_{i,t}, \tag{10.1}$$

  where $Q_{i,t}$ denotes the overflow rate at storage unit i in time step t averaged over the time interval Δt.
- The minimization of the pollution mass of released water is obtained by

$$f_2 = \sum_{i=1}^{S} \sum_{t=1}^{T} \rho_{i,t} \cdot Q_{i,t}, \tag{10.2}$$

  where $\rho_{i,t}$ denotes the pollution density given by the chemical oxygen demand (COD).
- Variations of some specific controllable element $i \in S$ are modelled by

$$f_3^i = \sum_{t=1}^{T-1} (u_{i,t+1} - u_{i,t})^2, \tag{10.3}$$

  where $u_{i,t} \in [0, 1]$ denotes the control. For describing variations in inflow to the WWTP we minimize the variation of flow through the last controllable element in the network before the WWTP is reached.

- The maximum utilization of the WWTP is described by

$$f_4 = \sum_{t=1}^{T-1} (u_{max} - u_{i,t})^2, \tag{10.4}$$

  where i denotes the last controllable element before the wastewater treatment plant is reached and u_{max} is the maximal capacity of this element. This goal prevents unnecessary storage in the network.

Note that also other goals like, for example, the goals additionally considered in [18], can easily be included into this model whenever necessary without any theoretical changes. The goal of avoiding the accumulated volumes in particular tunnels or channels of the network is not explicitly considered in this study since no such element exists in the practical examples that we considered. We neither incorporated a desired distribution of reserve storage volume because no data for a reasonable parameter selection expressing the desired distribution was available. However, objective (5) aims at the general reduction of the storage volume in the network and can thus be expected to have a positive effect on these objectives as well. Finally, operational costs could be taken into account if the corresponding data were available.

10.1.2 Terminology and Definitions

We consider general multiple criteria optimization problems of the form

$$\begin{aligned} &\min f(x) = \big[f_1(x), \ldots, f_k(x)\big]^T \\ &\text{s.t.} \quad x \in X, \end{aligned} \tag{10.5}$$

where $k \geq 2$ denotes the number of objective functions $f_i : X \to \mathbb{R}, i = 1, \ldots, k$, and $X \subseteq \mathbb{R}^n$, $X \neq \emptyset$, denotes the feasible set. See, for example, [8] and [19] for an introduction to the field of multiple criteria optimization. For the ease of notation, problem (10.5) is often formulated in the outcome space $Z := f(X)$:

$$\begin{aligned} &\min z = [z_1, \ldots, z_k]^T \\ &\text{s.t.} \quad z \in Z \subset \mathbb{R}^k, \end{aligned} \tag{10.6}$$

with $z_i = f_i(x), i = 1, \ldots, k$, and $x \in X$.

The minimization in (10.5) and (10.6) is understood with respect to a suitable and problem dependent (partial) order relation. A natural choice is the componentwise order leading to the notion of Pareto optimality which is used throughout this chapter. For two vectors $u, w \in \mathbb{R}^k$, $u < w$ denotes $u_i < w_i$ for all $i = 1, \ldots, k$, and $u \leq w$ denotes $u_i \leq w_i$ for all $i = 1, \ldots, k$, but $u \neq w$, whereas $u \leqq w$ allows equality. The symbols $>, \geq, \geqq$ are used analogously. With respect to the partial order induced by "$\leq$", we can now give a notion of optimality for problems (10.5) and (10.6):

A solution $\bar{x} \in X$ is called *efficient* or *Pareto optimal* if there is no other solution $x \in X$ such that $f(x) \leq f(\bar{x})$, i.e. there is no $x \in X$ such that $f_i(x) \leq f_i(\bar{x})$ for all $i = 1, \ldots, k$ and $f_i(x) < f_i(\bar{x})$ for at least one $i \in \{1, \ldots, k\}$. We denote the efficient set by X_E and the image of it by Z_N and refer to it as the nondominated set, i.e. $Z_N := \{f(\bar{x}) | \bar{x} \in X_E\}$. The elements of Z_N are called *nondominated*. If for a given $\bar{x} \in X$ it holds that there is no other solution $x \in X$ such that $f(x) < f(\bar{x})$, i.e. $f_i(x) < f_i(\bar{x})$ for all $i = 1, \ldots, k$, we call this solution *weakly efficient*. Analogously, in the outcome space we use the notion of weak nondominance.

A point $\bar{x} \in X$ is called *properly efficient* according to Geoffrion [10] if it is efficient and if there exists a scalar $M > 0$ such that for each $i = 1, \ldots, k$ and each $x \in X$ satisfying $f_i(x) < f_i(\bar{x})$ there exists an index $j \neq i$ with $f_j(x) > f_j(\bar{x})$ and

$$\frac{f_i(x) - f_i(\bar{x})}{f_j(\bar{x}) - f_j(x)} \leq M. \tag{10.7}$$

A lower bound for the set of nondominated points is determined by

$$z_i^I := \inf\{z_i : z \in Z\}, \quad i = 1, \ldots, k.$$

The point z^I is called *ideal point*. If the objectives considered are conflicting, there is no feasible solution mapped to the ideal point. The point $z^U \in \mathbb{R}^k$ with $z_i^U := z_i^I - \varepsilon, i = 1, \ldots, k$, where $\varepsilon > 0$ is a vector consisting of small positive numbers, is called *utopian point*. It is typically used instead of the ideal point to avoid numerical difficulties.

An upper bound on the nondominated set, called the *Nadir point*, is determined by

$$z_i^N := \sup\{z_i : z \in Z_N\}, \quad i = 1, \ldots, k.$$

As the supremum is taken over the set of nondominated points, it cannot be easily calculated a-priori in general if $k > 2$.

10.2 Methods

A common technique to treat multiple criteria problems is to transform the vector valued problem to one or several scalar valued problems, so-called *scalarizations*. Then well-known (single criterion) linear or nonlinear methods can be applied. Scalarizations are also used to solve multicriteria wastewater management problems, see e.g. [18] and [31], where the weighted-sum method is specified as scalarization approach.

We decided to use scalarization approaches because we intend to build our algorithm directly on the method proposed in Chap. 8 which uses gradient information to guide the optimization process. Note that we do not want to generate different nondominated points during the real-time optimal control process, firstly because of time restrictions and secondly because the process should be self-driven and no

decision maker is present who could choose the best suited solution for each individual situation. A decision maker is only involved in the phase before the real-time optimal control system is installed. For this phase, our multicriteria analysis is intended: We are interested in finding a good a-priori scalarization of the individual objectives which can then be used in the automated real-time control presented in Chap. 8.

An alternative solution approach avoiding scalarizations is the application of evolutionary multicriteria optimization (EMO), see e.g. [7] for an overview and [20] and [24] for applications in the wastewater management context. Since EMO methods are typically based on a multitude of function evaluations which in our case requires a large number of time consuming simulation runs, and since an interaction with a decision maker does not seem useful within an online optimization process, we do not consider EMO approaches in the following.

10.2.1 Scalarizations

Most of the scalarization methods presented in the following are well-known and can be found, for example, in the textbooks [8] and [19].

The Weighted-Sum Approach

Maybe the most prominent scalarization method is the weighted-sum approach, where a convex combination of the objective functions is built. The scalar problem to be solved is then

$$\begin{aligned} \min \quad & \sum_{i=1}^{k} \lambda_i f_i(x) \\ \text{s.t.} \quad & x \in X, \end{aligned} \tag{10.8}$$

where $\sum_{i=1}^{k} \lambda_i = 1$ and $\lambda_i \geq 0$ for all $i = 1, \ldots, k$. The parameters $\lambda_i, i = 1, \ldots, k$, are usually called weights because in some sense they indicate which relative importance the objective f_i, to which λ_i is associated, has.

One advantage of the weighted-sum approach is that the constraints of the problem do not change, i.e. the feasible set of (10.8) is the same as of the underlying multicriteria problem. So the scalarized problem is not harder to solve than a single-criterion problem obtained by dropping $k-1$ of the objective functions. If $\lambda_i > 0$ for all $i = 1, \ldots, k$, it is well-known (see e.g. [29]) that every solution of the weighted-sum approach is efficient. A drawback of the method is that the converse does not hold in general: Not every efficient solution of (10.5) can be reached by solving (10.8) with varying weights $\lambda_i \geq 0$, $i = 1, \ldots, k$, for general nonconvex problems, see e.g. [6]. Only for nondominated points $z \in Z_N$ lying on the boundary of the convex hull of Z there exist weights $\lambda_i \geq 0, i = 1, \ldots, k$, such that $z = f(x)$ solves problem (10.8). An example is depicted in Fig. 10.1.

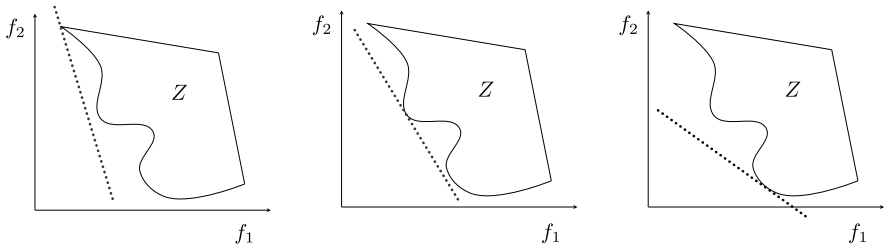


Fig. 10.1 Weighted-sum scalarization for different parameter choices

The ε-Constraint Method and Lexicographic Minimization

In this method, one of the objectives f_i with $i \in \{1, \ldots, k\}$ is selected and minimized whereas bounds are imposed on all other objectives:

$$\begin{aligned} &\min f_i(x) \\ &\text{s.t.} \quad f_j(x) \leq \varepsilon_j, \quad \forall j \neq i \\ &\qquad x \in X, \end{aligned} \tag{10.9}$$

where $\varepsilon \in \mathbb{R}^k$. It is well known (see e.g. [4]) that if (10.9) is feasible then every solution is weakly efficient. Moreover, among the set of solutions which solve (10.9) there is at least one efficient solution. On the other hand, for every nondominated point $z \in Z_N$ there exists an index $i \in \{1, \ldots, k\}$ and a vector $\varepsilon \in \mathbb{R}^k$ such that $z = f(x)$ solves (10.9).

Lexicographic optimization is typically not classified as a scalarization method but can be seen as a special sequence of ε-constraint problems to be solved. It is applied when a priority among the objective functions is known. First the most important objective is minimized while the others are neglected, thus we solve

$$\begin{aligned} &\min f_1(x) \\ &\text{s.t.} \quad x \in X. \end{aligned} \tag{10.10}$$

Let x^1 be the associated optimal solution of (10.10). In the ith problem to be solved, $i = 2, \ldots, k$, the ith objective is minimized whereas bounds are imposed on all f_j, $j < i$. These bounds are based on the optimal solutions of the $i - 1$ previous problems, namely

$$\begin{aligned} &\min f_i(x) \\ &\text{s.t.} \quad f_j(x) \leq f_j(x^j), \quad \forall j < i \\ &\qquad x \in X, \end{aligned}$$

with x^j being optimal for the jth problem. So each of the k scalar problems can be seen as an ε-constraint problem with $\varepsilon_j = f_j(x^j)$, $j < i$ and ε_j sufficiently large for all $j = i + 1, \ldots, k$. The solution which is finally obtained is efficient (see e.g. [8]).

The Weighted and the Augmented Weighted Tchebycheff Approach

The weighted Tchebycheff scalarization method was originally proposed in [3]. Thereby a point in the outcome space is generated which minimizes the distance to the utopian point z^U with respect to a weighted Tchebycheff norm:

$$\begin{aligned} &\min \max_{i=1,\dots,k} \left\{ w_i \left| z_i - z_i^U \right| \right\} \\ &\text{s.t.} \quad z \in Z. \end{aligned} \tag{10.11}$$

The parameters $w_i > 0, i = 1, \dots, k$, with $\sum_{i=1}^{k} w_i = 1$ denote the weights associated with each of the k objective functions. It is well known that every solution of (10.11) is weakly nondominated (see e.g. [19]) and that the set of solutions of (10.11) contains at least one nondominated point. On the other hand, for every nondominated point $z \in Z_N$ there exist weights $w \in \mathbb{R}^k$ such that z solves (10.11). As by definition $z_i^U < z_i$ for all $i = 1, \dots, k$ holds, the absolute values in the objective function can be dropped. In order to avoid that weakly nondominated points are generated instead of nondominated points, the augmented weighted Tchebycheff method was introduced in [28]. It is given by

$$\begin{aligned} &\min \max_{i=1,\dots,k} \left\{ w_i \left| z_i - z_i^U \right| \right\} + \rho \sum_{j=1}^{k} \left| z_j - z_j^U \right| \\ &\text{s.t.} \quad z \in Z, \end{aligned} \tag{10.12}$$

where $\rho \geq 0$ is a small positive scalar. For $\rho > 0$ it holds that each solution of (10.12) is properly nondominated, see [28]. For each nondominated point $z \in Z_N$ there exist parameters $w \in \mathbb{R}^k_+$ and $\rho \in \mathbb{R}_{\geq}$ such that z solves (10.12). Note that the use of an augmentation term like in (10.12) is not limited to the augmented weighted Tchebycheff method but present in various scalarization methods whenever the generation of properly efficient points is desired. Note also that the objective functions of problems (10.11) and (10.12) are not differentiable due to the max-norm. However, if all underlying functions are differentiable then the problem can be reformulated by introducing an additional variable in order to get a problem with differentiable objective and constraints:

$$\begin{aligned} &\min \delta + \rho \sum_{j=1}^{k} \left(z_j - z_j^U \right) \\ &\text{s.t.} \quad \delta \geq w_i \left(z_i - z_i^U \right), \quad i = 1, \dots, k, \\ &\qquad z \in Z. \end{aligned} \tag{10.13}$$

For $\rho = 0$ the corresponding reformulation of the weighted Tchebycheff problem is obtained. Note that k new constraints are added by this reformulation.

The (augmented) weighted Tchebycheff method can be seen as a special case of the more general approaches of reference point and direction methods which is pointed out in [16]. Reference point methods, introduced by [32], rely on the

idea that a given reference point shall be reached as best as possible. Usually, so-called achievement scalarizing functions are used for the minimization. Interpreting the (augmented) weighted Tchebycheff norm as an achievement function and the utopian point as a reference point yields the (augmented) weighted Tchebycheff method as a special case. Direction methods, proposed by [2] and extended in [21], are based on the idea to search from a starting point along a given direction $d \in \mathbb{R}^k$ for closest solutions. Similarly, by describing the minimization with respect to the weighted Tchebycheff norm as a minimal step along a certain search direction and by taking the utopian point as a starting point, the (augmented) weighted Tchebycheff method can be derived from the direction method.

The methods presented above are in a sense typical and frequently used scalarization methods and do not comprise all existing scalarization methods. For a more detailed overview we refer again to [8] and [19]. However, the above selection of methods appears to be reasonable in the context of our application and the goals mentioned in Sect. 10.1: The weighted-sum method is the easiest approach which does not change the structure of the problem and thus brings no additional complications for the solver, but suffers from its limited theoretical properties. The ε-constraint method adds constraints to the problem but does not introduce nondifferentiable functions or new variables. The theoretical properties are improved with respect to the weighted-sum method but the generation of (properly) nondominated solutions can not be guaranteed. When using the augmented weighted Tchebycheff method the problem structure becomes more difficult due to either the nondifferentiable objective function or the reformulation resulting in k new constraints and a new variable. However, the outcomes of this method are properly nondominated. We can also incorporate trade-off information when using this method which is described in the next subsection.

10.2.2 Trade-off

The parameter ρ has been introduced as a technical factor to guarantee that the outcomes of (10.12) are nondominated. Graphically it causes that the level curves of the objective function of (10.12) in the outcome space, which are parallel to the coordinate axes for the weighted Tchebycheff norm, are slightly lifted. We denote the augmented weighted Tchebycheff norm for $z \in \mathbb{R}^k$ in the following by

$$\|z\|_\rho^w := \max_{i=1,\dots,k} \{w_i |z_i|\} + \rho \left(\sum_{i=1}^{k} |z_i| \right),$$

where $w_i \geq 0, i = 1, \dots, k, \sum_{i=1}^{k} w_i = 1, \rho \geq 0$ and $\rho > 0$ if $\exists i \in \{1, \dots, k\} : w_i = 0$. Two examples of level curves of the augmented weighted Tchebycheff norm for the bicriteria case, restricted to the first quadrant, are depicted in Fig. 10.2. The parameters $w_i, i = 1, \dots, k$, determine the shape of the rectangle. The angles obtained by the lifting depend on w_i and ρ. Note that for $\rho = 0$ the level curve

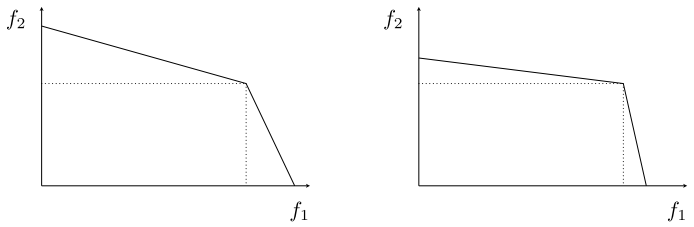


Fig. 10.2 Level curves of the augmented weighted Tchebycheff norm for fixed weights $w_1 = \frac{1}{3}, w_2 = \frac{2}{3}$ and $\rho = 0.3$ (*left*) and $\rho = 0.1$ (*right*)

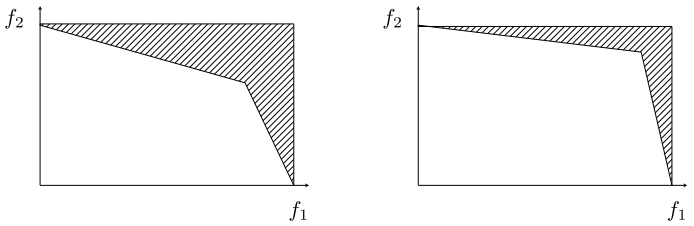


Fig. 10.3 Unreachable area in the augmented weighted Tchebycheff problem

equals the level curve of the weighted Tchebycheff norm. The bigger ρ is for fixed $w_i, i = 1, \ldots, k$, the more the level curve is lifted. Each time we solve problem (10.12) we obtain a properly nondominated solution. If we wish to find not only one but several nondominated solutions we have to solve a series of problems (10.12) with different choices of the parameter values. The parameter selection can be done beforehand, i.e., without taking any information about the problem into account, or adaptively using information obtained, i.e. based on already known solutions. The latter approach has been followed by several authors, see e.g. [15] for continuous problems or [23] for bicriteria discrete problems. In the bicriteria case, the parameters are computed with respect to two consecutive points in the ordered list containing those nondominated solutions that are currently known. Thereby the search is directed to the rectangle defined by taking these two nondominated points as vertices. A similar strategy can be followed in the multicriteria case, see, e.g., [15]. When using the augmented weighted Tchebycheff norm, a small part of the obtained rectangle is cut away in the sense that possibly existing nondominated points in this area can not be computed because the two extreme points of the rectangle lie on a level curve with smaller level α. In Fig. 10.3 the unreachable area is the shaded region. The smaller ρ is, the smaller is this unreachable area. If $\rho = 0$, then the unreachable area is empty but we lose the property that the generated solutions are (properly) nondominated.

The discussion above suggests the selection of a small value for ρ whenever the complete nondominated set should be determined. On the other hand, the dependence of the area in which nondominated points are generated from the parameter ρ can be used actively to represent the decision makers preferences. An appropriate

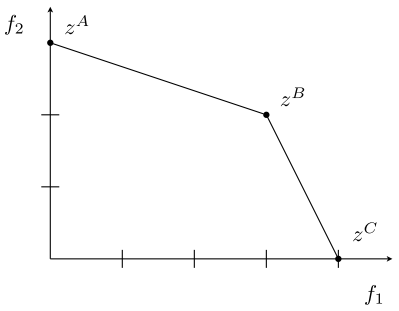


Fig. 10.4 Visualization of trade-off in z^A, z^B and z^C

choice of ρ allows, for example, to exclude solutions a priori that do not lead to a sufficient improvement (and hence avoid unnecessary computations), or solutions that do not have a satisfactory trade-off. Since this is of high practical relevance, particularly when the solution of scalarized subproblems is expensive as in the context of wastewater management, the selection of appropriate parameters reflecting the decision makers trade-off in the augmented weighted Tchebycheff method is discussed in more detail in the following. The notion of trade-off is closely related to the definition of proper efficiency. We use a slight modification of the definition of partial and total trade-off given in [4].

Definition 1 Let $x, \bar{x} \in X$. The ratio of change $T_{ij}(x, \bar{x})$ involving objective functions f_i and $f_j, i, j = 1, \ldots, k, i \neq j$, is defined as

$$\frac{f_i(x) - f_i(\bar{x})}{f_j(\bar{x}) - f_j(x)},$$

for $f_j(x) \neq f_j(\bar{x})$. If $f_l(x) = f_l(\bar{x})$ for all $l = 1, \ldots, k$ with $l \neq i, j$, then $T_{ij}(x, \bar{x})$ is called partial trade-off and total trade-off otherwise.

In the following, we focus on the bicriteria case $k = 2$. As in this case the definitions of partial and total trade-off coincide, we simply use the term trade-off in the following. If $f_i(x) \neq f_i(\bar{x})$ it can be easily seen that $T_{ij}(x, \bar{x}) = (T_{ji}(x, \bar{x}))^{-1}$ and $T_{ij}(x, \bar{x}) = T_{ij}(\bar{x}, x)$ holds.

Example 1 Consider the three points $z^A = (0, 3)^\top$, $z^B = (3, 2)^\top$ and $z^C = (4, 0)^\top$, see Fig. 10.4. Then it holds that

$$T_{12}\big(z^A, z^B\big) = 3, \qquad T_{12}\big(z^A, z^C\big) = \frac{4}{3} \quad \text{and} \quad T_{12}\big(z^B, z^C\big) = \frac{1}{2}.$$

Note that $T_{21}(z, \bar{z})$ represents the slope of the line connecting z and $\bar{z}$.

Assume that z^A and z^C are two nondominated, already computed points with $z_1^A < z_1^C$. The decision maker is interested in nondominated points $z \in Z$ that lie between z^A and z^C, i.e. points satisfying $z_1^A < z_1 < z_1^C$ and $z_2^A > z_2 > z_2^C$. Moreover, he or she specifies a desired trade-off, e.g. $T_{12}(z^A, z) \leq 3$. The question is now how

to translate this information into parameters for an augmented weighted Tchebycheff norm such that for the solution z of problem (10.12) it holds that $T_{12}(z^A, z) \leq 3$ or equivalently that $T_{21}(z^A, z) \geq \frac{1}{3}$. The fact that the parameter ρ can be used to incorporate trade-off information has been pointed out in [14]. But different from our approach only an appropriate ρ is computed while the weights are chosen arbitrarily. We incorporate the weights in our computation to direct the search to a specified box and theoretically guarantee to generate a new solution between two known solutions. We explained this concept in depth in [5], here we present the main ideas.

We will first show that if we postulate that the two extreme points z^A and z^C lie on the same level curve of an augmented weighted Tchebycheff norm $\|\cdot\|_\rho^w$, then the level curves depend on the level α which can be chosen from a certain interval. If we fix α, for example by determining a third point lying on the same level curve of $\|\cdot\|_\rho^w$ or by prescribing a (one-sided) trade-off rate in one of the extreme points, then the level curve and hence the parameters of the augmented weighted Tchebycheff norm are uniquely determined. This enables us to translate a given trade-off into suitable parameters for problem (10.12). For convenience we assume in the following that $z^U = 0$, $z^A = (0, y)$, $y \in \mathbb{R}_+$, and $z^C = (x, 0)$, $x \in \mathbb{R}_+$, which is achieved by a linear transformation. Furthermore, we will later focus on the case that $x > y$. The cases $x < y$ and $x = y$ are then straightforward. The trade-off information is assumed to be given with respect to point z^A, but can also be given for z^C.

Lemma 1 *Let $x, y > 0$, $z^A = (0, y)$, $z^C = (x, 0)$, $\alpha > 0$, $w_1, w_2 \geq 0$, $w_1 + w_2 = 1$, $\rho \geq 0$ where $\rho > 0$ if either $w_1 = 0$ or $w_2 = 0$. If $\|z^A\|_\rho^w = \|z^C\|_\rho^w = \alpha$ then it holds that*

$$w_1(\alpha, x, y) = \frac{\alpha(y - x) + xy}{2xy}, \qquad w_2(\alpha, x, y) = \frac{\alpha(x - y) + xy}{2xy} \tag{10.14}$$

$$\text{and} \quad \rho(\alpha, x, y) = \frac{\alpha(x + y) - xy}{2xy} \tag{10.15}$$

for $\alpha \in [\frac{xy}{x+y}, \frac{xy}{\max\{x,y\}-\min\{x,y\}}]$, if $x \neq y$, and for all $\alpha \geq \frac{x}{2}$, if $x = y$.

Proof From the definition of the norm $\|\cdot\|_\rho^w$ we derive

$$\left\|z^A\right\|_\rho^w = \alpha \quad \Longleftrightarrow \quad w_2 = \frac{\alpha - \rho y}{y}$$

and

$$\left\|z^C\right\|_\rho^w = \alpha \quad \Longleftrightarrow \quad w_1 = \frac{\alpha - \rho x}{x}.$$

Hence

$$w_1 + w_2 = 1 \quad \Longleftrightarrow \quad \rho = \frac{\alpha(x + y) - xy}{2xy}$$

and thus

$$w_1(\alpha, x, y) = \frac{\alpha(y-x)+xy}{2xy}, \qquad w_2(\alpha, x, y) = \frac{\alpha(x-y)+xy}{2xy}.$$

For $\rho = 0$ we obtain $\alpha = \frac{xy}{x+y}$ and $w_1 = \frac{y}{x+y}, w_2 = \frac{x}{x+y}$. If $\rho > 0$ and $0 \leq w_1, w_2 \leq 1$ it holds that

$$\rho > 0 \quad \Longleftrightarrow \alpha(x+y) - xy > 0 \quad \Longleftrightarrow \quad \alpha > \tfrac{xy}{x+y}$$

and

$$w_1 \geq 0 \quad \Longleftrightarrow \alpha(y-x) + xy \geq 0 \quad \Longleftrightarrow \quad \begin{cases} \alpha \geq \frac{-xy}{y-x}, & y > x, \\ \alpha > 0, & x = y, \\ \alpha \leq \frac{-xy}{y-x}, & x > y. \end{cases}$$

As $\alpha \geq 0$ by definition and $\frac{-xy}{y-x} < 0$ for $y > x$, the first condition is always satisfied and does not impose a bound on α. Only the condition in the case $x > y$ is binding. Note that for $x = y$, $\alpha(y-x) \geq -xy$ is satisfied, so we do not get any additional bound on α. Thus $w_1 \geq 0$ holds if $\alpha \leq \frac{xy}{x-y}$ for $x > y$. Moreover,

$$w_1 \leq 1 \quad \Longleftrightarrow \quad \frac{\alpha(y-x)+xy}{2xy} \leq 1 \quad \Longleftrightarrow \quad \begin{cases} \alpha \leq \frac{xy}{y-x}, & y > x, \\ \alpha > 0, & x = y, \\ \alpha \geq \frac{xy}{y-x}, & x > y. \end{cases} \tag{10.16}$$

With the same argumentation as before we see that only the first condition imposes a bound on α and $w_1 \leq 1$ holds if $\alpha \leq \frac{xy}{y-x}$ for $y > x$. Analogously, for $0 \leq w_2 \leq 1$ we get $\alpha \leq \frac{xy}{y-x}$, $y > x$ and $\alpha \leq \frac{xy}{x-y}$, $x > y$. So it follows for $x \neq y$ that

$$\alpha \leq \frac{xy}{\max\{x, y\} - \min\{x, y\}}.$$

If $x = y$, no upper bound on α is imposed and $\alpha \geq \frac{x}{2}$ has to be satisfied. □

Note that the upper limit of α, i.e. $\alpha = \frac{xy}{x-y}$ for $x > y$ or $\alpha = \frac{xy}{y-x}$ for $x < y$, corresponds to $w_1 = 0, w_2 = 1$ or $w_1 = 1, w_2 = 0$, respectively. Then the level curve equals a weighted l_1-norm. This means that the "third" extreme point of the norm in the first quadrant coincides either with z^A or z^C. Both weights are different from 0 and 1 if and only if there exists a third extreme point of the level curve different from z^A or z^C. We now give a description of this extreme point dependent on α.

Lemma 2 *Let $0 < w_i < 1, i = 1, 2$, and $\bar{z}$ with $\|\bar{z}\|_\rho^w = \|z^A\|_\rho^w = \|z^C\|_\rho^w = \alpha$ be the point where $w_1\bar{z}_1 = w_2\bar{z}_2$ holds. Then the coordinates of this point are given by*

$$\bar{z}(\alpha, x, y) = \frac{\alpha}{w_1 w_2 + \rho}(w_2, w_1). \tag{10.17}$$

Proof As $w_1\bar{z}_1 = w_2\bar{z}_2$ holds, $\bar{z}$ can be found on the half-line starting from the origin and passing through the point $(1, \frac{w_1}{w_2})$. Thus there exists $\lambda > 0$ such that $\bar{z} = \lambda(1, \frac{w_1}{w_2})$. So

$$\|\bar{z}\|_\rho^w = \max\{w_1 \cdot \bar{z}_1, w_2 \cdot \bar{z}_2\} + \rho(\bar{z}_1 + \bar{z}_2) = \alpha$$

implies that

$$\alpha = \max\{w_1\lambda, w_1\lambda\} + \rho\left(\lambda + \frac{w_1}{w_2} \cdot \lambda\right) = \lambda\left(w_1 + \rho + \rho\frac{w_1}{w_2}\right).$$

Solving this equation with respect to λ and taking into account that $w_1 + w_2 = 1$, we obtain that

$$\bar{z} = \frac{\alpha}{w_1 w_2 + \rho}(w_2, w_1). \qquad \square$$

By fixing level α, all parameters and the third extreme point of the level curve of $\|\cdot\|_\rho^w$ at level α are uniquely defined. The level can be fixed by, for example, prescribing the trade-off at one of the two extreme points.

Theorem 1 *Let $x > y > 0$ and an augmented weighted Tchebycheff norm be given such that $\|z^A\|_\rho^w = \|z^C\|_\rho^w = \alpha$. Let a trade-off $T_{12}(z^A, z)$ be given and let γ denote the inverse of this given trade-off.*

Then w_1, w_2 and ρ are given by

$$w_1 = \frac{y - x\gamma}{x + y - 2x\gamma}, \qquad w_2 = \frac{x - x\gamma}{x + y - 2x\gamma} \quad \text{and} \quad \rho = \frac{x\gamma}{x + y - 2x\gamma}. \tag{10.18}$$

Proof All points z that have a trade-off which is less or equal than the desired trade-off $T_{12}(z^A, z)$ lie "below" the line $h(t) = y - \gamma \cdot t$, where $\gamma := T_{21}(z^A, z)$ and $0 < \gamma < \frac{y}{x}$. We want the third extreme point to lie on this straight line thus we insert $\bar{z}$ in h, which yields $\bar{z}_2 = y - \gamma \cdot \bar{z}_1$. We use now (10.17), insert the parameters from Lemma 1 and solve with respect to α to obtain the two solutions

$$\alpha_1 = \frac{-xy}{x - y} \quad \text{and} \quad \alpha_2 = \frac{xy}{x + y - 2x\gamma}.$$

The first solution is infeasible as $\alpha_1 < 0$. The second solution is well defined as the denominator is positive for $x > y$. Checking the bounds on α calculated in Lemma 1 ensures that α_2 is the unique feasible level α for the given problem. Inserting this into (10.14) and (10.15) implies the explicit representation. $\square$

Corollary 1 *Every solution $z \neq z^A$ of (10.12) with parameters chosen as in (10.18) for which $\|z\|_\rho^w = \alpha$ and $z_1 \leq \bar{z}_1$ hold satisfies $T_{12}(z^A, z) = \frac{1}{\gamma}$.*

Every solution z of (10.12) with parameters chosen as in (10.18) for which $\|z\|_\rho^w < \alpha$ or $\|z\|_\rho^w = \alpha$ and $z_1 > \bar{z}_1$ hold satisfies $T_{12}(z^A, z) < \frac{1}{\gamma}$.

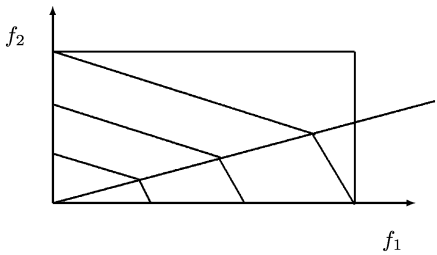


Fig. 10.5 Level curves for Example 2

Example 2 Let $z^A = (0, 2)^\top (y = 2)$, $z^C = (4, 0)^\top (x = 4)$ and $T_{12}(z^A, z) = 3$ be given. Thus $\gamma = \frac{1}{3}$. Then $\alpha = 2.4$, $w_1 = 0.2$, $w_2 = 0.8$, $\rho = 0.4$ and $z = (\frac{24}{7}, \frac{6}{7})^\top$. Different level curves of the associated augmented weighted Tchebycheff norm $\|\cdot\|_\rho^w$ are depicted in Fig. 10.5.

Note that in general it is not possible to prescribe trade-off information from both sides, i.e. for z^A and z^C. Instead of trade-off information, also a desired point may be given which is to be placed on the level curve of $\|\cdot\|_\rho^w$. Note that for a valid augmented Tchebycheff norm, it is in general not possible to postulate that this point represents the third extreme point.

If the problem is integer-valued then we can make use of the finite trade-off existing among nondominated points. In this case it is possible to compute the parameters such that explicit bounds on ρ can be derived such that no nondominated point is missed, i.e. that the unreachable area is small enough such that it contains no integer point. Thereby we avoid to set ρ to some predefined fixed value which may result in numerical difficulties or the oversight of nondominated solutions. For details we refer to [5].

10.2.3 Approximation of the Nondominated Set

Having selected an appropriate scalarization method, the decision maker has to decide on one (or several, possibly scenario dependent) settings for the involved parameters that represent his preferences best. These parameters are then used within the real-time control. To provide the decision maker with the necessary information to take a well-founded decision, an overview over alternative solutions and, in particular, the trade-off between the considered criteria, is provided by generating an approximation of the nondominated set.

For each considered inflow scenario, this is realized by solving a sequence of scalarized problems. In contrast to the term representation which typically describes a set of nondominated points, an approximation also includes solutions obtained when the underlying scalar problems can not be solved to optimality or when only local optimality can be assured.

We use the definition of an approximation of Z_N given in [13]:

Definition 2 A finite set $A \subseteq f(X)$ is called an *approximation* of the set Z_N if for all points $z^1, z^2 \in A, z^1 \neq z^2$ it holds that

$$z^1 \not\leq z^2 \quad \text{and} \quad z^2 \not\leq z^1,$$

i.e. if no point in A is dominated by any other point in A.

Note that this definition is exclusively based on the nondominance relation and does not include any criteria related to the quality of an approximation. In [26], coverage, uniformity and cardinality of the approximating set are proposed as quality measures. This aims at an approximation which contains sufficiently many, but not too many evenly distributed points that represent the nondominated set sufficiently well. The survey paper [25] summarizes different approximation approaches for bi- and multiobjective optimization problems.

Since for the application at hand we decided to use scalarization-based approaches, the question of how to choose the parameters of the scalarized subproblems in order to achieve a good approximation has to be answered. For example in the case of weighted-sums scalarizations, [6] highlight the problem that an even distribution of weights does not necessarily result in an even spread of approximation points. In contrast to [6] where the concept of normal boundary intersection is introduced which is based on the application of the direction method with a problem dependent parameter setting, a problem dependent weight selection scheme for the weighted-sum approach is suggested in Sect. 10.3 to overcome this difficulty. An alternative approach aiming at the minimization of the approximation error rather than on an even spread of approximation points is presented in [15]. For convex problems it is based on the successive solution of weighted-sums scalarizations with an adaptive parameter selection scheme, while in the non-convex case, adaptive weighted Tchebycheff scalarizations are implemented. An interesting feature of this method is that regions of particular interest can be identified in interaction with the decision maker and the approximation can be immediately and easily refined in these regions, see [17] for more details. Using a different quality measure, [12] suggest a box algorithm for discrete bicriteria problems. After computing the lexicographic minima of the problem, a box approximation is constructed and iteratively updated by formulating and solving appropriate ε-constraint scalarizations. Boxes are selected for subdivision with respect to their volume, i.e., the approximation is refined in regions where no or only few approximation points have already been computed. More details of these methods and how they were implemented in the context of our application will be given in Sect. 10.3.1.

Once an approximation is known, the decision making process can be aided by the application of interactive methods. Since particularly for problems with more than two objectives the visualization of trade-off information becomes problematic, an interactive learning-oriented method called Pareto navigator for nonlinear multiobjective optimization problems was developed in [9]. Having computed a (possibly rough) polyhedral approximation of the nondominated set that can be easily obtained from the approximation points, the decision maker can navigate around the polyhedral approximation and direct the search for promising regions where the most preferred solution could be located. In this way, the decision maker can learn about the interdependencies between the conflicting objectives and possibly adjust

his preferences. Once an interesting region has been identified, the polyhedral approximation can be made more accurate in that region or the decision maker can ask for the closest counterpart in the actual Pareto optimal set.

10.3 Computational Results

10.3.1 Implementational Details

The finite volume solver described in Chap. 8 is used for our computational study. Therefore all modelling aspects of Lamatto also hold for our study. Besides, the movement of the pollution particles has to be modelled. This is realized by a pure transport equation which simplifies the underlying physical and chemical processes immensely and only gives a rough idea of the distribution of pollution mass in the network but was sufficient for our analysis.

The code is supplemented by the scalarization methods described in Sect. 10.2, a list which maintains the set of nondominated points, functions updating the parameters of the subproblems and some interface functions. The user can choose by specifying options which objectives are considered, which scalarization method and which parameter update scheme is used. As a flexible single-objective optimizer for nonlinear constrained optimization problems, we applied IPOPT which has already been tested on real-world multiobjective problems, see e.g. [11], and which we describe briefly in the following.

Solving the Single Criterion Scalarized problems

IPOPT is a primal-dual interior-point algorithm with a filter line-search method for nonlinear programming [30]. Thereby, a general optimization problem

$$\begin{aligned} &\min f(x) \\ &\text{s.t.} \quad c(x) = 0, \\ &\qquad x^l \le x \le x^u \end{aligned} \tag{10.19}$$

is considered, where x^l and x^u denote the lower and upper bounds of $x \in \mathbb{R}^n$, respectively. The objective function $f : \mathbb{R}^n \to \mathbb{R}$, and the constraints $c : \mathbb{R}^n \to \mathbb{R}^m$ with $m \le n$ are assumed to be twice continuously differentiable. For simplification of notation it is assumed that all lower bounds in (10.19) are zero (i.e. $x^l = 0$). This simplification of Problem (10.19), denoted by (10.19), is converted into a sequence of barrier problems

$$\begin{aligned} &\min \varphi_\mu(x) := f(x) - \mu \sum_{i=1}^{n} \ln(x_i) \\ &\text{s.t.} \quad c(x) = 0 \end{aligned} \tag{10.20}$$

with the barrier parameter μ being driven to zero. It is shown that under certain standard conditions, the optimal solutions of (10.20) converge to an optimal solution of (10.19) for $\mu \to 0$. The barrier problems (10.20) are solved by applying a damped Newton's method to the primal-dual equations

$$\nabla f(x) + \nabla c(x)\lambda - z = 0, \tag{10.21}$$

$$c(x) = 0, \tag{10.22}$$

$$XZe - \mu e = 0, \tag{10.23}$$

$$x, z \geq 0, \tag{10.24}$$

with $X = \operatorname{diag}(x)$, $Z = \operatorname{diag}(z)$, $e = (1, \ldots, 1)^\top$ the vector of all ones and $\lambda \in \mathbb{R}^m$, $z \in \mathbb{R}^n$ being the Lagrange multipliers of the constraints of (10.19). Search directions are then obtained from the linearization of (10.21) at a current iterate. The step size is determined by a backtracking line-search procedure. For more details on IPOPT we refer to [30].

IPOPT converges if a sufficient number of iterations is performed. If, however, due to time restrictions, a maximum number of iterations is specified in advance, this may not be sufficient for finding a stationary point and only an intermediate solution is returned by IPOPT. Note that for constrained problems, the returned solution may even not be feasible.

In the case of wastewater management problems, and particularly in the context of a real-time optimal control or a multiobjective analysis (where generally many individual optimizations are required), we have to interrupt the minimization process after some predetermined number of iterations due to the limited amount of time that is available for the individual optimization runs. We thus have to expect that the outcomes of our computations are intermediate solutions and are in general no stationary points. For practical problems this is usually not critical since in the wastewater management context with its dynamics and uncertainties the goal is generally not to find the absolute optimum; a reasonable improvement as compared to the uncontrolled case is usually satisfactory. From a theoretical point of view, however, it is important to note that we often deal with non-Pareto outcomes and that the theoretical properties of the scalarization methods derived in Sect. 10.2 may not be applicable.

Approximation

We construct a discrete approximation based on a finite set of points which gives a rough idea of the shape of the nondominated set. Initially, the lexicographic minimal solutions are determined. This gives us bounds on the nondominated set which are added to the approximation. Note that due to the numerical problems described above, these solutions do not necessarily represent global bounds but may be updated during the solution process.

During the course of an approximation algorithm, the solution of each subproblem is tested for dominance. If it is nondominated with respect to the set of already known solutions, it is added to the approximation. Since the generated solutions are in general not globally optimal for the respective subproblems (see the discussion above), they may be dominated by some outcome that is found in a later subproblem. Thus each point in the approximation may leave the approximation at a later stage when a dominating solution is found.

Scalarizations

We implemented the weighted-sum, the ε-constraint and the augmented weighted Tchebycheff method. For our computational study we set $z^U = 0$ and $\rho = 10^{-3}$. The lexicographic minima are computed by a special ε-constraint method as explained in Sect. 10.2.1.

Parameter Update

Each subproblem requires a different parameter setting. We tested three different rules to choose the parameters for a sequence of bicriteria subproblems to be solved:

- For the first and simplest rule a total number $N \geq 1$ of subproblems to be solved is given. The parameters are chosen with equidistant spread. For the weighted-sum approach, λ_1 varies between 0 and 1 with an even increment of $\frac{1}{N}$ in each iteration. The second parameter is computed by $\lambda_2 = 1 - \lambda_1$. The ε-constraint approach is not evaluated with this simple method as this does not take the magnitude of the objective function values into account. Therefore, it is very likely to construct infeasible problems when fixing ε to any value not related to the magnitude of the left-hand-side of (10.9). The directions in the augmented weighted Tchebycheff problem are set to $d_1 = \lambda_2$ and $d_2 = \lambda_1$.
- The second rule follows the parameter update of the a-priori box algorithm described in [12]. While this method was originally developed for the ε-constraint method, the ideas are transferred here to the weighted-sum and to the augmented weighted Tchebycheff approach: Let z^1 and z^2 be the lexicographic minimal solutions and let again $N \geq 1$ denote the number of subproblems to be solved. Let $\Delta x = \frac{z_1^2 - z_1^1}{N}$ and $\Delta y = \frac{z_2^1 - z_2^2}{N}$. The parameters of the weighted-sum method are set to

$$\lambda_1 = \frac{k\Delta y}{\Delta x + \Delta y}, \qquad \lambda_2 = 1 - \lambda_1,$$

the parameter of the ε-constraint method to

$$\varepsilon = z_1^2 + \frac{k}{N}\Delta y$$

and the directions in the augmented weighted Tchebycheff problem to

$$d_1 = k \cdot \left(z_1^2 - z_1^1\right), \qquad d_2 = (N-k) \cdot \left(z_2^1 - z_2^2\right)$$

for $k = 1, \ldots, N-1$, respectively. As the parameters are computed based on the lexicographic minimal solutions, the magnitude of the individual objective function values is taken into account.

- The third rule picks up the idea of the a-posteriori box algorithm in [12] and of the adaptive approximation algorithm in [15]. The coordinates of each pair of nondominated points z^i and z^j which are adjacent in the list of nondominated points define a box whose volume is $(z_1^j - z_1^i)(z_2^i - z_2^j)$. For each subproblem, the box with the largest volume or the box with the largest approximation error is selected. The parameters are set with respect to the nondominated points defining the selected box. The directions of the augmented weighted Tchebycheff problem are set to

$$d_1 = z_1^j - z_1^i, \qquad d_2 = z_2^i - z_2^j,$$

the weights of the weighted-sum method to

$$\lambda_1 = \frac{d_2}{d_1 + d_2}, \qquad \lambda_2 = 1 - \lambda_1$$

and the parameter of the ε-constraint method to

$$\varepsilon = z_j^2 + 0.5\left(z_2^i - z_2^j\right).$$

If the solutions are global optimal then either a new point in the considered box or one of its defining vertices is computed. In the latter case this box is excluded from further search while in the first case the new point is added to the set of nondominated points, the box volumes are updated and the search continues until a predefined maximum box volume is attained.

Note that in order to handle dominated outcomes, we modified this procedure slightly. It may happen that we select two nondominated points from the list, update the parameters with respect to them, but no solution is found between them (i.e. not even the defining solutions which are always feasible). In this case we have to exclude this box from the search space because otherwise, it would be selected in each subsequent iteration again. This is implemented by artificially setting the volume of that box to zero.

10.3.2 Evaluation I: Interdependencies of the Considered Objectives

In our computational study we used the academic network presented in Chap. 7 (see Sect. 7.3, Fig. 7.8) with different inflow scenarios. Before presenting the results

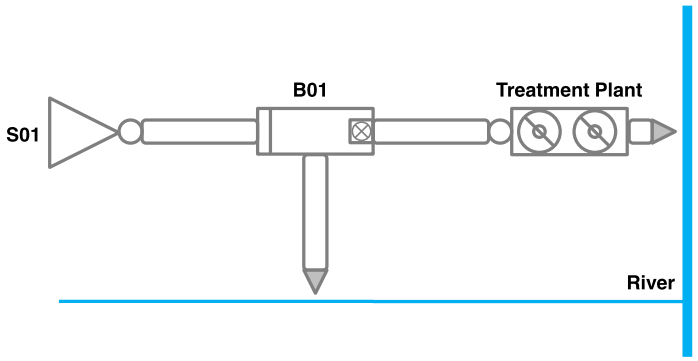


Fig. 10.6 Considered network: A detail of the academic network of Chap. 7, Fig. 7.8

obtained with these examples, we analyze the effects between pairs of objectives in more detail. For this purpose, we first present results for two objectives and for a subnetwork of the described network.

Test Network

The first test network, see Fig. 10.6, consists of one single inflow node, a channel connecting this inflow node to a storage unit, and a controllable pump at the end of the storage unit through which water enters another channel which leads to the wastewater treatment plant. Whenever the storage unit is overcharged, water leaves the storage unit through an overflow. The height of this overflow can be controlled by a weir.

This simple network can be seen as a detail of a bigger network. Thus the inflow node does not necessarily represent a natural inflow but a node of inflow of water coming from the upper part of a larger network. Analogously, the channel behind the pump does not necessarily lead directly to the wastewater treatment plant but may connect to parts of a bigger network lying behind.

Total Release Versus Total Pollution Mass

The two goals concerning quantity and quality of water release can be conflicting, but this is not necessarily the case for all data sets and/or network structures. If there exists, for example, an optimal control strategy such that no water has to be released at all we clearly attain the goal of minimizing pollutants at the same time. Besides this trivial case there are also non-trivial cases in which by minimizing with respect to volumes we also achieve the minimal total pollution. These examples are less interesting for our analysis as the nondominated set then shrinks to one single point, i.e., the ideal point is feasible. In the following we discuss two examples for which it is not possible to minimize total release and total pollution at the same time.

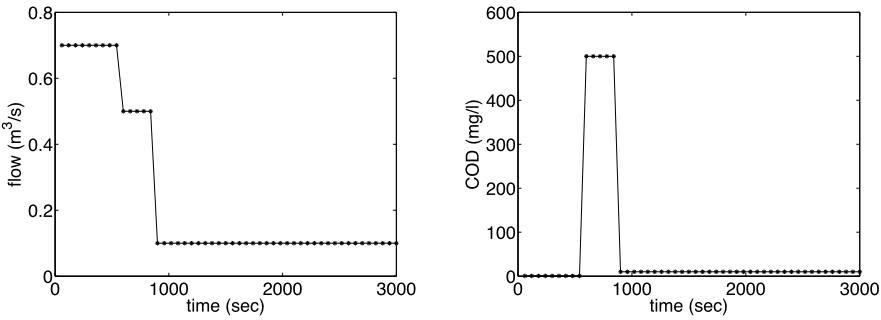


Fig. 10.7 Inflow Q_t and pollution density ρ_t over time

Scenario 1 This scenario represents a heavy and sudden rainfall which causes a high inflow. The COD concentration is low at the first time steps and then rises significantly. This may reflect removal of deposits from the channel walls in upper parts of the network.

We set the total time to $T = 3000$, i.e. we consider 3000 time steps. The flow and pollution input data is actualized each 60 time steps, so we set $\Delta t = 60$. The inflow $Q_t, t = 1, 2, \ldots, T$, (in m^3 per second) and the pollution density $\rho_t, t = 1, 2, \ldots, T$, (in kg per m^3) are given explicitly for $t = \Delta t, 2\Delta t, \ldots$ and interpolated linearly for all other time steps. The values used for the example are depicted in Fig. 10.7. The weir is controlled each 100 time steps, so there are 30 variables. The weir height is scaled to $[0, 1]$, where $u_t = 0$ means that the weir is completely opened at time t and $u_t = 1$ denotes that the weir is closed.

Minimizing only with respect to total release yields the solution $z^1 = (93.8, 11.85)^\top$, so a total release of 93.8 m^3 and a total pollution mass of 11.85 kg. Fixing the optimal objective function value $f_1 = 93.8$ and optimizing with respect to the pollution mass, i.e. solving $\min\{f_2(x) : f_1(x) \leq 93.8, x \in X\}$ results in the solution $z^2 = (95.2, 5.1)^\top$. Although this solution is not feasible for the constrained problem (recall that this effect is due to the limit on the maximum number of iterations of IPOPT which may not be sufficient to guarantee convergence to a stationary point), it is a feasible solution for the optimal control problem. Compared to $z^1 = (93.8, 11.85)^\top$, it only has a small impairment with respect to f_1 (1.5 %) but a large improvement with respect to f_2 (57.0 %). When computing an approximation of the set of nondominated points we obtain for the weighted-sum method with the a-posteriori parameter update rule (rule 3) the solution $z^3 = (94.2, 0.1)^\top$, which we expect to be the (local) ideal point. Thus the second objective is nearly reduced to its minimum, zero pollution, and the objective function value of the first objective is only slightly worse compared to z^1 (0.4 %).

A closer look at the two control strategies for z^1 and z^3 reveals that the water quantities to be released are simply shifted, see Fig. 10.8. By releasing more clean water in earlier time steps, capacity is made available for the time steps when polluted water streams in. Thereby nearly all the polluted water can be kept in the net-

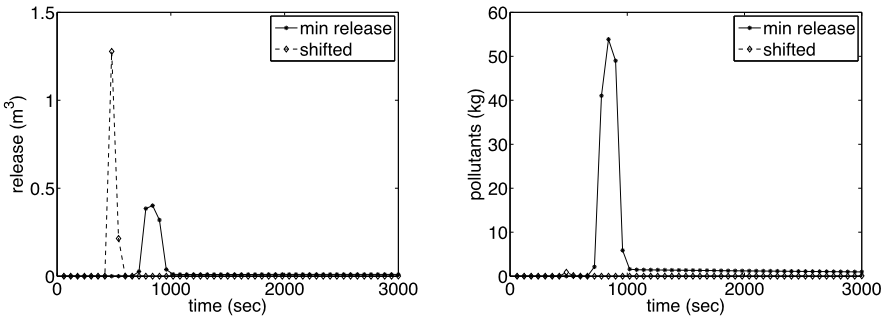
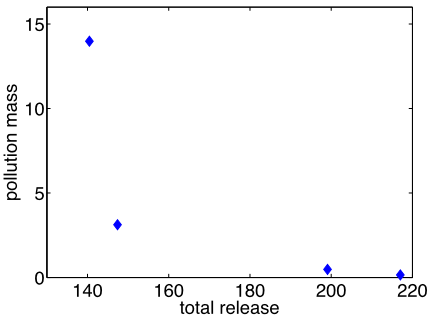


Fig. 10.8 Total release (*left*) and pollution mass (*right*) for solutions z^1 (*min release*) and z^3 (*shifted*)

Fig. 10.9 Approximation for Scenario 2



work. But whether such an ideal strategy is possible depends on the parameters of the scenario and can not be guaranteed in general, as the following example shows.

Scenario 2 We consider the same inflow data as in Scenario 1 but limit the capacity of the pump in the first 700 time steps: Instead of 90 l/s only 18 l/s (20 %) can be pumped out of the storage unit. This represents the case that less water can leave the storage unit in the first time steps because the capacity of the network behind the pump is exhausted.

In this example we can not shift water volumes in order to reduce the pollution of the released water to zero. We thus have a significant conflict between the two criteria: Improving the solution with respect to one criterion causes an impairment with respect to the other criterion. Optimization with respect to the first objective gives $z^1 = (140.5, 14.0)^\top$, while the best value for the second objective is obtained for solution $z^2 = (217.0, 0.2)^\top$. Note that similar to the previous example we can find a good compromise solution $z^3 = (147.4, 3.1)^\top$ that with respect to z^1 improves the water quality immensely (by 78.9 %) whereas the released quantity only rises by 4.9 %. An approximation generated with the weighted-sum method is depicted in Fig. 10.9. Note that even though many more subproblems were solved, the approxi-

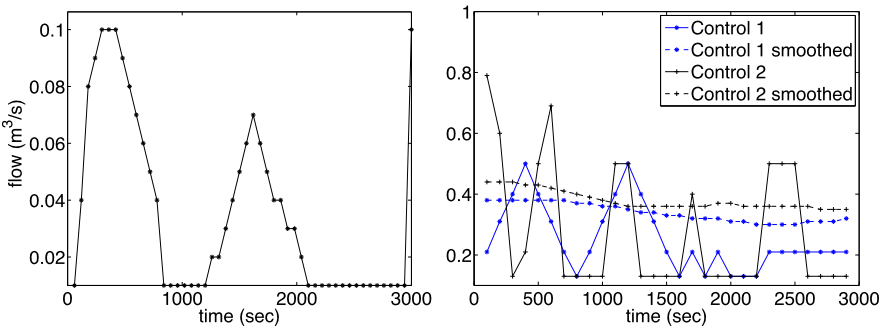


Fig. 10.10 Inflow (*left*) and two different controls minimizing total release and their smoothed counterpart

mation only contains four points since in most subproblems dominated points were computed.

Total Release Versus Constant Inflow to the Wastewater Treatment Plant

The third objective, the minimization of variations in inflow to the wastewater treatment plant, was investigated separately. Different from the goal to minimize pollution mass, we see this goal clearly subordinate to minimizing the total release of water. Therefore, we search only among the optimal solutions of the minimal total release problem for an optimal solution of minimal variance of inflow to the wastewater treatment plant. Consequently, the lexicographic optimization approach is applied in this case.

As in the previous case, there exist scenarios for which this secondary objective is automatically optimized in the primary optimization process. If, for example, in each time step water has to be released, then there are no variations in the inflow to the wastewater treatment plant since the channel connecting to it is always fully charged. To avoid trivial cases in the following, we consider only scenarios in which release of water does not occur in every time step.

Figure 10.10 shows the considered inflow data. In this example, the pump is controlled. Minimizing the total release yields a solution for which no water is released at all ($f_1 = 0$). There are infinitely many controls leading to this result, and the optimal control returned by the solver depends on the starting solution. Figure 10.10 shows two solutions minimizing the total release obtained from different starting solutions. Fixing the total release to zero and optimizing with respect to the second objective then gives in both cases a much smoother control than before. Note that these smoothed solutions would probably not have been found by only considering the total release of water and thus the lexicographic approach significantly improves the solution.

10.3.3 Evaluation I: Discussion and Challenges

In Sect. 10.2.1 we presented three scalarization methods and their theoretical properties. In this subsection we investigate how the methods perform numerically.

Comparison of the Parameter Selection Rules

As expected, the first rule with the evenly distributed parameters not related to the magnitude of the objective function values performed quite bad for the weighted-sum as well as for the augmented weighted Tchebycheff method. For $N = 10$, no new solution different from the solution obtained for weights $(1, 0)^\top$ was computed. So the variation of the weights was useless because no new solution was found due to the relative "overweighting" of the first objective. This was also observed for the augmented weighted Tchebycheff method with evenly distributed directions.

The second rule, the variant of the a-priori box method for which the parameters are computed with respect to the lexicographic minima, performed well. Different new (locally) nondominated points were found for all three scalarization methods. For the third rule, the a-posteriori box method, only few iterations were performed and thus only few new points were generated. However, the generated points had a good quality in general (for example, the quasi-ideal solution of the first scenario was computed with this rule).

For generating an approximation of the Pareto set in the wastewater management problem it seems to be the best option to use the a-priori method because on the one hand it automatically includes information about the magnitude of the objective function values and on the other hand it screens the interesting region without premature termination. However, the adaptive a-posteriori method can also contribute solutions not found by the a-priori method and is a good option if only few subproblems shall be solved.

Comparison of the Scalarizations

In general we can say that all three methods, the weighted-sum, the ε-constraint and the augmented weighted Tchebycheff method solve the problems in a satisfying way. All three methods contributed solutions to the approximation. However, we notice that for the considered example problems the weighted-sum method seems to generate better solutions, i.e. contributes more solutions to the final approximation. This can be explained by the specific problem data which appears to generate an only mildly non-convex multicriteria optimization problem, and by the structure of the scalarized optimization problems: In the weighted-sum method no new constraints are introduced. This in turn also means that every control associated to an intermediate solution is feasible both for the original optimal control problem as well as for the scalarization. For methods in which constraints are added, the initial and intermediate solutions may be infeasible and thus the solver has to find an

optimal and feasible solution simultaneously. Even if the infeasibility with respect to the scalarization is not a problem because the solution is feasible for the optimal control problem, this may nevertheless influence the quality of the final objective function value obtained after a prescribed number of iterations.

Summarizing the discussion above, the weighted-sum method worked best for the above example problems and with the applied solver, despite its theoretical shortcomings for non-convex problems. This again emphasizes the importance of an a priority problem analysis where the overall problem structure is analyzed before a specific scalarization method is selected for the online control process. For problems with a more complicated (network) structure and a higher degree of non-convexity, this analysis may lead to a different selection.

10.3.4 Evaluation II: Academic Test Network with Three Criteria

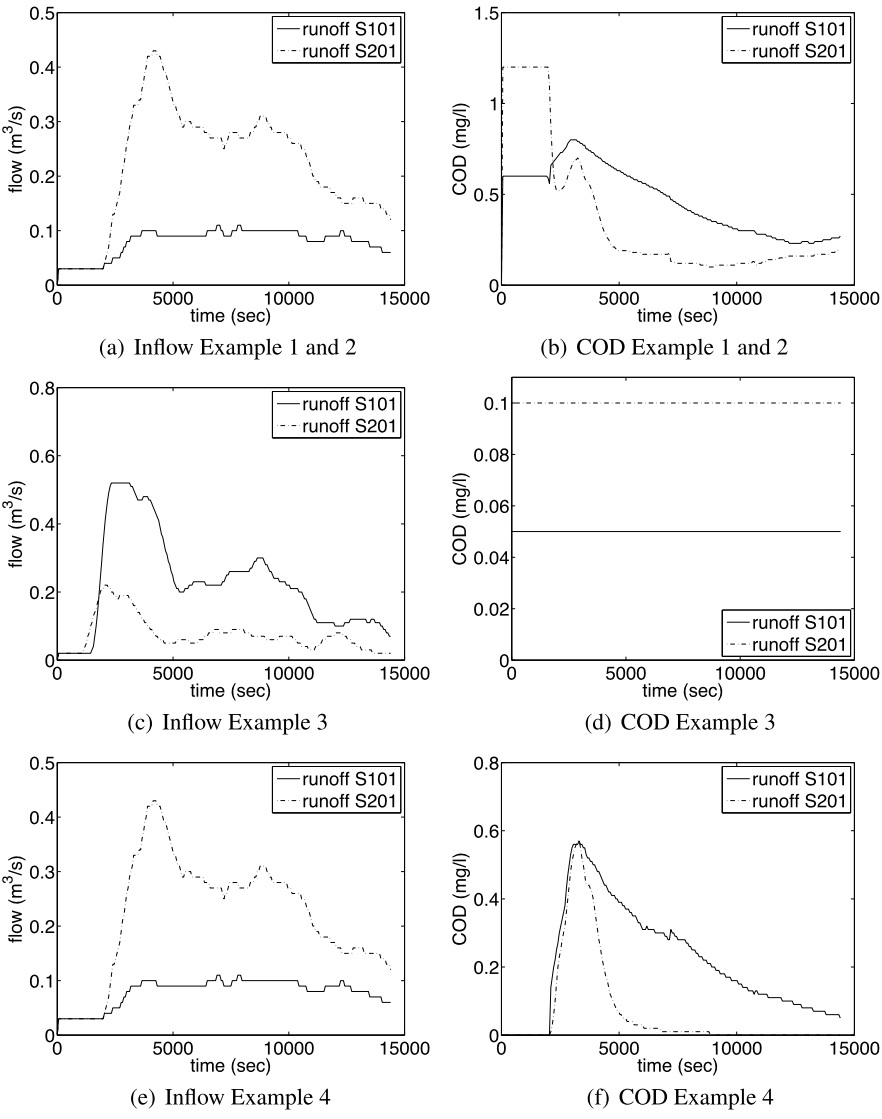
Finally, we present the results obtained for the academic network depicted in Chap. 7, Fig. 7.8 and different inflow scenarios. Four different data sets for inflow values are shown in Fig. 10.11. We consider the data recorded during the first four hours of the time series given in Chap. 7, Fig. 7.9. One control step takes 10 minutes. For simplification the forecast and control horizon coincide, thus no receding horizon strategy is active. The problem consists of 24 variables which equal the number of control steps during the considered four hours. Our reference problem is the uncontrolled case, i.e. 90 l/s leave B01 constantly. For every instance, a tricriteria optimization problem is solved where the third objective is lexicographically optimized after the approximation of the nondominated set with respect to the first two objectives has been computed. All solutions given in the following are those obtained before optimizing with respect to the third objective.

In the first example, the uncontrolled solution is $(1192.34, 349.15)^\top$. With the lexicographic approach we get $(304.93, 64.87)^\top$ which corresponds to an improvement of 74.4 % with respect to the first and 81.4 % with respect to the second objective. With the weighted-sum method we find the even better solution $(280.30, 58.98)^\top$ for weights $(0.6, 0.4)^\top$.

In the second example, the same inflow data is used as in Example 1 but the lower storage unit B01 is set one meter deeper and the power of the pump behind the upper storage unit B02 is increased from 90 l/s to 120 l/s. The uncontrolled solution is $(850.49, 236.67)^\top$. With the lexicographic approach we get $(0, 0)^\top$ which is the ideal solution.

In the third example, the uncontrolled solution is $(850.70, 48.27)^\top$. With the lexicographic approach we get $(287.52, 14.38)^\top$. No better solution was found so we expect this solution to be the ideal solution.

In the fourth example, the uncontrolled solution is $(771.76, 88.50)^\top$. With the lexicographic approach we get $(312.09, 24.12)^\top$. With the weighted-sum method we find the improved solution $(303.76, 23.32)^\top$ for weights $(0.1, 0.9)^\top$. Note that for all test problems, the lexicographic minimization with respect to the third objective, the minimization of variations of control, did not lead to an improvement of the



(a) Inflow Example 1 and 2 (b) COD Example 1 and 2

(c) Inflow Example 3 (d) COD Example 3

(e) Inflow Example 4 (f) COD Example 4

Fig. 10.11 Inflow and pollution density at the two runoffs S01 and S02

third objective while maintaining the objective values in the first two components. Therefore the resulting controls are not stated here.

10.4 Conclusions

Optimization problems occurring in wastewater management are by their very nature multiobjective. We discussed different goals occurring in the context of wastew-

ater management problems and modelled the control of the flow of the wastewater through the sewage network to the wastewater treatment plant as a multiobjective optimization problem.

In order to couple the multiobjective analysis with the real-time optimal control developed in Chap. 8, scalarization methods were applied that comprise the multiple objectives into one single objective function and possibly additional problem constraints. Given a specific sewage network and data of typical inflow scenarios, an offline computation of an approximation of the nondominated set for the different scenarios is used to provide the decision maker with trade-off information and guide him or her with the determination of his preferences. Based on this analysis, a specific (possibly scenario-dependent) scalarization can be selected and applied in the real-time control of the network.

Numerical results obtained for test instances taken from Chap. 7 show that the selection of an appropriate scalarization has a large impact on the quality of the resulting solution. In all considered examples, significant reductions in the pollution of released water could be achieved at the price of only a small increase in the total overflow, i.e., the total amount of released water. Having this knowledge at hand, the decision maker can select an appropriate scalarization that, with a high probability, generates solutions that suit his or her preferences best.

The solution of the underlying single-objective optimization problems within a real-time control process is a big challenge by itself that is addressed in Chap. 8. Depending on the particular problem instance and on the applied solvers, we have to expect that the scalarized subproblems cannot be solved to global optimality within the available time. In this situation, the generated solutions may be non-optimal for the scalarization and dominated for the multiobjective problem, even though the scalarization method was selected carefully and theoretically guarantees nondominated solutions. Our numerical tests with IPOPT showed that this is usually less problematic in the case of weighted-sums scalarizations, while other scalarization approaches often pose more problems to the solver. The multiobjective offline analysis can help identifying such problems and select an appropriate scalarization taking into account the performance of the respective solver.

Future research should address these numerical problems also from the point of view of multiobjective approximation algorithms. By combining appropriate scalarizing functions with an adaptive approximation approach, this analysis should aim at the generation of an approximation of the nondominated set that is in a certain sense robust with respect to non-optimality in the subproblems.

Acknowledgements We thank Lars Balzer for support in programming.

References

1. L.J. Alvarez-Vazquez, N. Garcia-Chan, A. Martinez, M.E. Vazquez-Mendez, Multi-objective Pareto-optimal control: An application to wastewater management. Comput. Optim. Appl. **46**, 135–157 (2010)

2. G. Boldur, Linear programming problems with complex decision conditions, in *7th Mathematical Programming Symposium*, The Hague, September 1970
3. V.J. Bowman, On the relationship of the Tchebycheff norm and the efficient frontier of multiple-criteria objectives, in *Multiple Criteria Decision Making*, ed. by H. Thieriez, S. Zionts (Springer, Berlin, 1976), pp. 76–85
4. V. Chankong, Y.Y. Haimes, *Multiobjective Decision Making: Theory and Methodology* (Elsevier, New York, 1983)
5. K. Dächert, J. Gorski, K. Klamroth, An adaptive augmented weighted Tchebycheff method to solve discrete, integer-valued bicriteria optimization problems. Technical Report BUW-AMNA-OPAP 10/06, University of Wuppertal, FB Mathematik und Naturwissenschaften, 2010
6. I. Das, J.E. Dennis, A closer look at drawbacks of minimizing weighted sums of objectives for Pareto set generation in multicriteria optimization problems. Struct. Optim. **14**, 63–69 (1997)
7. K. Deb, *Multi-Objective Optimization Using Evolutionary Algorithms* (Wiley, Chichester, 2001)
8. M. Ehrgott, *Multicriteria Optimization* (Springer, Berlin, 2005)
9. P. Eskelinen, K. Miettinen, K. Klamroth, J. Hakanen, Pareto navigator for interactive nonlinear multiobjective optimization. OR Spektrum **23**, 211–227 (2010)
10. A.M. Geoffrion, Proper efficiency and the theory of vector maximization. J. Math. Anal. Appl. **22**, 618–630 (1968)
11. J. Hakanen, Y. Kawajiri, K. Miettinen, L.T. Biegler, Interactive multi-objective optimization for simulated moving bed processes. Control Cybern. **36**(2), 283–302 (2007)
12. H.W. Hamacher, C.R. Pedersen, S. Ruzika, Finding representative systems for discrete bicriteria optimization problems by box algorithms. Oper. Res. Lett. **35**(3), 336–344 (2007)
13. M.P. Hansen, A. Jaszkiewicz, Evaluating the quality of approximations to the non-dominated set. Technical Report IMM-REP-1998-7, IMM Technical University of Denmark, 1998
14. I. Kaliszewski, Using trade-off information in decision-making algorithms. Comput. Oper. Res. **27**, 161–182 (2000)
15. K. Klamroth, J. Tind, M.M. Wiecek, Unbiased approximation in multicriteria optimization. Math. Methods Oper. Res. **56**, 413–437 (2002)
16. K. Klamroth, J. Tind, Constrained optimization using multiple objective programming. J. Glob. Optim. **37**, 325–355 (2007)
17. K. Klamroth, K. Miettinen, Integrating approximation and interactive decision making in multicriteria optimization. Oper. Res. **56**, 224–234 (2008)
18. M. Marinaki, M. Papageorgiou, *Optimal Real-Time Control of Sewer Networks*. Advances in Industrial Control (Springer, Berlin, 2005)
19. K. Miettinen, *Nonlinear Multiobjective Optimization* (Kluwer Academic, Boston, 1999)
20. D. Muschalla, Optimization of integrated urban wastewater systems using multi-objective evolution strategies. Urban Water Journal **5**(1), 57–65 (2008)
21. A. Pascoletti, P. Serafini, Scalarizing vector optimization problems. J. Optim. Theory Appl. **42**, 499–524 (1984)
22. M. Pleau, H. Colas, P. Lavallée, G. Pelletier, R. Bonin, Global optimal real-time control of the Quebec urban drainage system. Environ. Model. Softw. **20**, 401–413 (2005)
23. T. Ralphs, M. Saltzman, M.M. Wiecek, An improved algorithm for solving biobjective integer programs. Ann. Oper. Res. **147**, 43–70 (2006)
24. W. Rauch, P. Harremoës, Genetic algorithms in real time control applied to minimize transient pollution from urban wastewater systems. Water Res. **33**(5), 1265–1277 (1999)
25. S. Ruzika, M.M. Wiecek, A survey of approximation methods in multiobjective programming. J. Optim. Theory Appl. **126**(3), 473–501 (2005)
26. S. Sayin, Measuring the quality of discrete representations of efficient sets in multiple objective mathematical programming. Math. Program. **87**, 543–560 (2000)
27. M. Schütze, A. Campisano, H. Colas, W. Schilling, P.A. Vanrolleghem, Real time control of urban wastewater systems – where do we stand today? J. Hydrol. **299**, 335–348 (2004)

28. R.E. Steuer, E. Choo, An interactive weighted Tchebycheff procedure for multiple objective programming. Math. Program. **26**, 326–344 (1983)
29. R.E. Steuer, *Multiple Criteria Optimization: Theory, Computation, and Application* (Wiley, New York, 1986)
30. A. Wächter, L.T. Biegler, On the implementation of a primal-dual interior point filter line search algorithm for large-scale nonlinear programming. Math. Program. **106**(1), 25–57 (2006)
31. G. Weinreich, W. Schilling, A. Birkely, T. Moland, Pollution based real time control strategies for combined sewer systems. Water Sci. Technol. **36**(8–9), 331–336 (1997)
32. A.P. Wierzbicki, The use of reference objectives in multiobjective optimization, in *Multiple Criteria Decision Making Theory and Applications*, ed. by G. Fandel, T. Gal (Springer, Berlin, 1980), pp. 468–486

K. Dächert · K. Klamroth
Department of Mathematics and Informatics, Faculty of Mathematics and Natural Sciences, University of Wuppertal, Gaußstr. 20, 41097 Wuppertal, Germany

K. Dächert
e-mail: daechert@math.uni-wuppertal.de

K. Klamroth (✉)
e-mail: klamroth@math.uni-wuppertal.de