Doctoral Thesis,

Gisen, David

Modeling upstream fish migration in small-scale using the Eulerian-Lagrangian-agent method (ELAM)

BAWDissertationen

BAW
Bundesanstalt für Wasserbau

BAWDissertationen
Nr. 1

Modeling upstream fish migration in small-scale using the Eulerian-Lagrangian-agent method (ELAM)

Kompetenz für die Wasserstraßen

BAWDissertationen
Nr. 1

Modeling upstream fish migration in small-scale using the Eulerian-Lagrangian-agent method (ELAM)

David Gisen

Karlsruhe, Oktober 2018

Impressum

In der Reihe **BAW**Dissertationen werden Doktorarbeiten, die im Rahmen von wissenschaftlichen Kooperationen der BAW mit Universitäten entstanden sind, als Erst- oder Zweitveröffentlichung publiziert.

Herausgeber (im Eigenverlag):
Bundesanstalt für Wasserbau (BAW)
Kußmaulstraße 17, 76187 Karlsruhe
Postfach 21 02 53, 76152 Karlsruhe
Telefon: +49 (0) 721 97 26-0
Telefax: +49 (0) 721 97 26-4540
E-Mail: info@baw.de, www.baw.de

ISSN 2625-7777 (Print)
ISSN 2625-8072 (Online)

ISBN 978-3-939230-55-7 (Print)
ISBN 978-3-939230-56-4 (Online)

Verfügbar unter: https://hdl.handle.net/20.500.11970/105158

Karlsruhe, Oktober 2018

Titelbild: Fische in der Ethohydraulischen Versuchsrinne

Modeling upstream fish migration in small-scale using the Eulerian-Lagrangian-agent method (ELAM)

David Christoph Gisen

Vollständiger Abdruck der an der Fakultät für Bauingenieurwesen und Umweltwissenschaften der Universität der Bundeswehr München zur Erlangung des akademischen Grades eines

Doktor-Ingenieurs (Dr.-Ing.)

genehmigten Dissertation.

Gutachter:
1. Univ.-Prof. Dr.-Ing. Andreas Malcherek
   Universität der Bundeswehr München
2. Prof. Dr. rer. nat. Volker Grimm
   Universität Potsdam und
   Helmholtz-Zentrum für Umweltforschung
3. Prof. Dr.-Ing. Christoph Heinzelmann
   Technische Universität München und
   Bundesanstalt für Wasserbau

Die Dissertation wurde am 18. September 2017 bei der Universität der Bundeswehr München eingereicht und durch die Fakultät für Bauingenieurwesen und Umweltwissenschaften am 4. Januar 2018 angenommen. Die mündliche Prüfung fand am 17. Januar 2018 statt.

Abstract

Upstream directed fish migration in the German federal waterway system is blocked by many dams. Building efficient fishways is a way to mitigate this ecological problem, but to determine efficiency using field or laboratory studies is costly and time-consuming. A numerical model simulating fish movement in spatial and temporal scales of decimeters and seconds, respectively, can support selection of competing fishway designs and contribute to scientific understanding of their performance.

The Eulerian-Lagrangian-agent method (ELAM) was chosen to develop a new individual-based model, "ELAM-de". 3D computational fluid dynamics input on arbitrary polyhedral meshes was obtained from the open source toolbox OpenFOAM®. Six behavior patterns were derived from movement data of live brown trout under two different flow fields collected in a large flume (length × width × water depth was 11.78 m×2.50 m×0.60 m). The behavior patterns were used to develop, calibrate, and validate the individual-based model.

For the first flow field, containing a high-velocity jet, calibrated model predictions minus observed data were within ±18.3 percentage points or less for all six patterns. Validation in the second, more homogeneous flow field obtained qualitative agreement for five patterns using identical model parameters. Advective acceleration magnitude and hydrostatic pressure were identified as key hydraulic stimuli to reproduce observed behavior. Acclimatization through simple memory functions was incorporated as a critical process for the individual-based model. ELAM-de produces estimates for flume passage success and metabolic cost, which – after successful further validation – can be used to compare efficiencies of alternative fishway designs.

Kurzfassung

Die aufwärtsgerichtete Fischwanderung in den deutschen Bundeswasserstraßen wird durch viele Stauanlagen behindert. Der Bau von effizienten Fischaufstiegsanlagen kann dieses ökologische Problem verringern. Effizienz im Feld oder im Labor zu bestimmen, ist jedoch kostspielig und langwierig. Ein numerisches Modell, das Fischverhalten auf räumlichen und zeitlichen Skalen von Dezimetern beziehungsweise Sekunden simuliert, kann Planungsentscheidungen unterstützen und das wissenschaftliche Verständnis der Auswirkungen verschiedener Bauweisen fördern.

Die „Eulerian-Lagrangian-agent"-Methode (ELAM) wurde für die Entwicklung eines neuen individuenbasierten Modells („ELAM-de") herangezogen. Das Modell wurde mit 3D-Strömungsdaten gespeist, die mit der quelloffenen Softwarebibliothek OpenFOAM® auf Netzen aus beliebigen Polyedern berechnet wurden. Sechs Verhaltensmuster wurden abgeleitet aus Bewegungsdaten von lebenden Bachforellen in zwei unterschiedlichen Strömungsfeldern in einer großen Laborrinne (Länge × Breite × Wassertiefe 11,78 m × 2,50 m × 0,60 m). Mittels der Verhaltensmuster wurde das individuenbasierte Modell erstellt, kalibriert und validiert.

Im ersten Strömungsfeld, das einen Jet mit erhöhter Geschwindigkeit enthielt, betrug die Differenz zwischen kalibriertem Modellergebnis und Beobachtung bei allen sechs Mustern maximal ±18,3 Prozentpunkte. Die Validierung im zweiten, homogeneren Strömungsfeld ergab mit denselben Modellparametern eine qualitative Übereinstimmung bei fünf Mustern. Der Betrag der advektiven Beschleunigung und der hydrostatische Druck waren entscheidende hydraulische Stimuli für die Wiedergabe des beobachteten Verhaltens. Auch Akklimatisierung mit Hilfe einfacher Gedächtnisfunktionen war ein wichtiger Bestandteil des individuenbasierten Modells. ELAM-de liefert Schätzwerte für Passageraten und metabolische Kosten in einer Rinne, die – nach erfolgreicher weiterer Validierung – verwendet werden können, um die Effizienz verschiedener Bau- und Betriebsweisen von Fischaufstiegsanlagen zu vergleichen.

Acknowledgments

The present work was created from 2015–2017 at the Bundesanstalt für Wasserbau (BAW) in Karlsruhe, Germany. It combines aspects from the very different fields of hydraulic engineering, ethology, and computer sciences. The people who contributed to its success may be different as well, but they all shared the will to help and I am very thankful for their support!

I want to thank Dr. Andy Goodwin for his enthusiasm, great ideas, and the willingness to share them. He liberally provided the *Fortran 90* code used for the calculations in Goodwin et al. (2014) for free use, which is gratefully acknowledged.

As a German without a great talent for foreign languages, I was lucky to improve my English scientific writing through working with Dr. John Nestler, and I am thankful for his patience and great teaching abilities.

Last, I want to thank the creators and community of `stackexchange.com`, a website where (almost) any software-related question I had about C++, LaTeX, and MATLAB® was already answered in the most clear and concise way.

Danksagung

Die vorliegende Arbeit entstand von 2015–2017 an der Bundesanstalt für Wasserbau (BAW) in Karlsruhe. Sie verbindet Aspekte der grundverschiedenen Disziplinen Wasserbau, Verhaltensbiologie und Informatik. Die Perso-

nen, die zu ihrem Erfolg beigetragen haben, sind ebenso verschieden. Jedoch teilen sie eine große Hilfsbereitschaft, wofür ich sehr dankbar bin!

Ich danke Dr. sc. techn. Roman Weichert für Inspiration, Vertrauen und viele Stunden an hilfreichem Korrekturlesen, seitdem ich an der BAW arbeite. Ich danke Prof. Dr.-Ing. Andreas Malcherek für die Annahme meiner Arbeit, seine Erfahrungen beim Schreiben und für verständliche Erläuterungen von Differentialgleichungen. Ich danke Prof. Dr. rer. nat. Volker Grimm für umfassende Erklärungen zu individuenbasierter Modellierung bei meinem Besuch in Leipzig und für jederzeitige Hilfsbereitschaft. Ich danke Prof. Dr.-Ing. Christoph Heinzelmann für die Anregung zu dieser Arbeit, klare Zielvorgaben und die Förderung der ethohydraulischen Versuche an der BAW. Prof. Dr. rer. nat. Thomas Apel danke ich für die Übernahme des Vorsitzes der Prüfungskommission und die stringente Organisation des Verfahrens.

Ich danke Dr. rer. nat. Cornelia Schütz für intensives Korrekturlesen und für alle Informationen zu den ethohydraulischen Versuchen. Für Erhebung und Auswertung von Fischbeobachtungen in Labor und Natur danke ich dem gesamten BfG-Team von U4, besonders Dr. rer. nat. Martina Heynen, B.Sc. Julia Walbrühl, Dr. rer. nat. Arne Rüter sowie Dipl.-Biol. Christian von Landwüst für umfangreiche Literatur zur Geschichte der Bundeswasserstraßen und ihrer Fischfauna.

Ich danke Dr.-Ing. Rebekka Czerny und M.Eng. Sonia Vettori für Geschwindigkeitsmessungen im Labor, Dr.-Ing. Patrick Heneka und Dr.-Ing. Lena Mahl für Korrekturlesen sowie Alfred Bätza und dem gesamten BAW-Team von W1 für jederzeitige Hilfsbereitschaft, Diskussionsbereitschaft, Fachkompetenz und angenehme Arbeitsatmosphäre.

Ich danke den Mitarbeitern der Professur für Hydromechanik und Wasserbau und der Verwaltung an der UniBw München für die freundliche Hilfe bei allen organisatorischen Fragen.

Schließlich danke ich meinen Eltern für ihre bedingungslose Unterstützung, meinem Onkel KH für die Erkenntnis, dass Talent auch eine Verpflichtung bedeutet und meiner Frau Ute für viel Verständnis und Inspiration.

Karlsruhe, Februar 2018 David Gisen

Contents

Chapter 1

Introduction

1.1 Fish migration in German rivers

In Germany, fish migration is often associated with the picture of jumping salmon. This picture is however very unlikely to be observed because salmon and other migrating species are rare in German rivers. What are the causes of this rarity – and how can it be changed? To approach this question, a definition of fish migration and a look in its history are helpful.

Fish migration can be classified as part of fish movement in general. Fish movement is any change in location, including passive drift. Fish migration typically is periodic, takes place on a population-level, and includes a habitat or environment change (Koehn and Crook 2013). It is well known that salmon migrate between the sea and fresh water (*diadromous* species). However, migration is also important to species which stay in fresh water during their lifetime (*potamodromous* species). Lucas and Baras (2001, p. 5) define three principal functional categories of fish migration: Reproduction, feeding, and refuge. Different sea and river reaches fulfill these needs. For example in natural freshwaters, many fish species use shallow and slow streaming zones for mating, spawning and growing. Faster flowing and more open zones may provide floating food and hunting opportunities, whereas deep scours may provide refuge from predators, low or high temperatures, and floods. But free migration is restricted.

German rivers have been impounded for centuries for energy production, navigation, and flood control. Small streams were intensively blocked for watermills starting in the middle ages, which had large consequences on

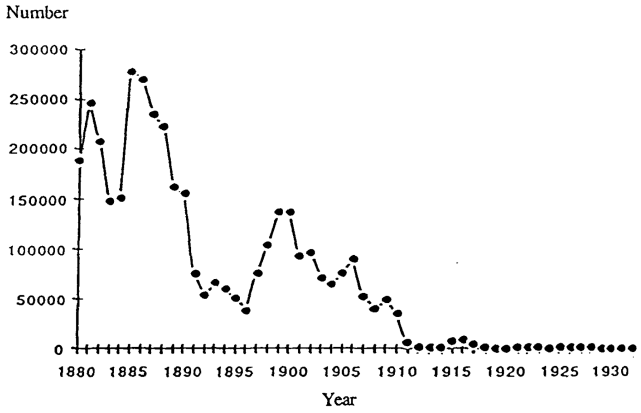


Figure 1.1: Historic allis shad landings in the Netherlands illustrate the sharp decline from an important food fish to insignificance (Groot 1990, p. 253).

fish populations (Lenders et al. 2016). Destruction of spawning habitats, overfishing, and water pollution led to further declines in fish populations, despite extensive stocking of salmon in the Rhine river (Borne 1881). Massive blocking of fish migration in the Rhine river system started in the late 1800s, when large concrete dams[1] were build in the tributaries Main (from 1883; Wikipedia 2017), Neckar (from 1924; Koch 1929), and Moselle (1942–1964; Landwüst 2004). This further fueled the downfall: Allis shad disappeared in 1910 (Figure 1.1) and salmon in the 1950s (ICPR 2004).

Fishways were provided on most dams of Main, Neckar, and Moselle to enable continuous upstream migration. However, they often were too steep, placed too far away from natural migration routes, and were not sufficiently maintained to work effectively (see BfG 2010). Downstream migration routes usually follow the bulk river flow through turbines and over spillways, where fish can take damage, but efforts for protection were lacking. Through changing public awareness for the environment, fueled by the Endosulfan and Sandoz disasters in 1969 and 1986 respectively, legislation changed from the 1970s on and led to dramatic reduction in river pollution (Lelek and Buhse 1992; DLF Kultur 2008). Rivers habitats were gradually restored. Adoption of the European Water Framework Directive (WFD) by the European Parliament in the year 2000 initiated the construction of new fishways. Today, ongoing counts in modern fishways and fish monitoring programs demon-

[1] The American term *dam* is used generally, in contrast to the German *Damm*, which by definition is made of soil. If a *dam* is made of concrete, it is best translated as *Staumauer* or *Stauanlage*, depending on the context.

strate that a variety of species is present (again), but total abundance and relative composition do not match the original natural state (e.g., WFBW 2017; BfG 2017).

To meet the goals of the WFD, hundreds of new fishways have to be constructed on the German federal waterways (Heinzelmann et al. 2013). Their biological and economical efficiencies for more than 60 domestic species have to be considered, while traits like swim capability as well as migration, schooling, and feeding behavior vary largely between species. For evaluation and comparison of fishway design, plain counts are obviously insufficient, as they are subject to diverse uncontrolled effects. Defining a meaningful metric and measuring the necessary data for fishway biological efficiency is a complex task (Cooke and Hinch 2013). Field studies to gain the necessary data are hampered by natural variability of fish stocks and environmental factors. Example challenges are to find the number of fish intending to move upstream or to quantify metabolic cost of passage. Laboratory investigations are restricted in dimensions and influenced by artificial surroundings. In addition, both approaches are costly and time-consuming.

A new opportunity to evaluate efficiency is provided by numerical simulation methods for hydraulics and fish behavior. Numerical methods represent a high degree of precision, combined with flexibility and speed. The present work deals with individual-based models (IBMs) based on the Eulerian-Lagrangian-agent method (ELAM). This method is especially suitable for comparing competing construction designs.

1.2 Individual-based modeling and the ELAM

Individual-based modeling[1] is a young discipline of ecology which developed over the recent decades. A basic idea of IBM is to let reproducible *patterns emerge* from individual behavior of populations of living entities, steered by programmable rules.

- Patterns can be defined as any sign of order above random variation (Grimm and Railsback 2005, p. 38). They occur at various levels from system to individual, and at various time scales. This makes them especially valuable for modeling, as diverse information is integrated

[1] also: agent-based modeling (ABM)

by pattern-oriented analysis. The quality of different models can be measured by their ability to reproduce patterns observed in nature or laboratory.

- Emergence is the process of creating unexpected behavior through interaction of system components (Camazine et al. 2003, p. 8). It is opposed to direct implementation (imposing) of behavior or behavior patterns. System components can be individual traits and experience. Traits are algorithms which specify a certain behavior of an individual. Adaptive traits involve a reaction to environmental stimuli or internal states (Grimm and Railsback 2005, p. 74).

The IBM approach facilitates understanding of underlying processes and supports both a wide application range and portability of models. A fundamental goal of IBM is to identify fitness-seeking traits on the individual level. If met, the model is well-suited for predictions under changing environmental conditions, because traits are more general and reliable than commonly used empirical relationships (Stillman et al. 2015). IBM applications are computer programs and have been increasing in spatial and temporal resolutions as computers have continued to get more powerful.

The Eulerian-Lagrangian-agent method (ELAM) is a general concept developed to create IBMs of the movement of fish in multidimensional flow fields. The term "ELAM" was coined by Goodwin et al. (2006). As it reflects the approach of most recent fish IBMs (often unmentioned, however), I decided to use the ELAM for model development and for structuring this work.

As illustrated in Figure 1.2, the ELAM combines computational fluid dynamics (CFD) model output in Eulerian description and individual movement and transport in Lagrangian description by means of an *agent*. An agent is the mathematical representation of an individual, e.g. a fish. The Eulerian[1] reference frame uses a (virtual) control volume, which is static in space, to describe passing flow in time ("watching a river from the bank"). The Lagrangian[2] reference frame uses a (virtual) moving parcel to describe flow in time and space ("drifting down a river").

The variety of interacting model components can cause confusion. To

[1] named after Swiss mathematician Leonard Euler
[2] named after Italian-French mathematician Joseph-Louis Lagrange

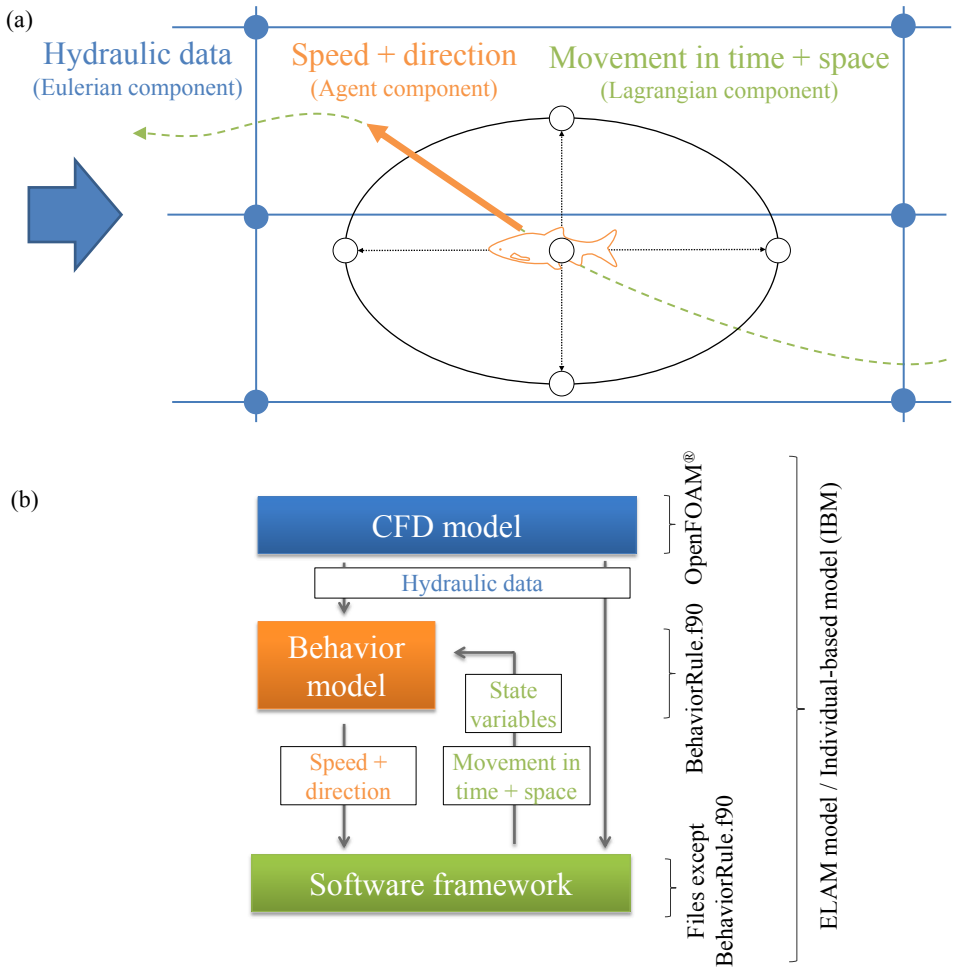


Figure 1.2: An ELAM model combines Eulerian, Lagrangian, and Agent components. Subfigure (a) illustrates model information attributed to the components. Subfigure (b) shows the implementation of the components in general and in this particular work (right hand side annotations). Model information flows between the software components as follows: Hydraulic data is available from the computational fluid dynamics (CFD) model in Eulerian reference frame (on a mesh). The behavior model (agent) uses interpolated hydraulic data along with stored state variables to compute speed and direction for an individual. Then, the software framework moves the individual in time and space and tracks its state variables along the path in Lagrangian reference frame. This procedure is repeated every time step.

structure this work it is useful to differentiate between *framework* and *behavior model*.

- The software framework can be reused in very different IBMs and provides reusable functions such as data storage, input/output, vector arithmetic, etc. SWARM (swarm.org 2017) and NetLogo (Wilenski 2017) are examples for pure frameworks.
- The behavior model consists of a unique set of behavior rules and parameters and produces an output reaction from given input. In context of the ELAM, it forms the agent component.

As IBMs are very different, these definitions are not generally true. Also, the term "IBM" can be used either just for the behavior model, be expanded to include the framework, or be further expanded to include the model(s) used for input, e.g. the CFD model. In this work, the latter, most comprehensive definition is used, i.e. all components of an ELAM model form a single IBM.

The numerical evaluation of fishway efficiency requires modeling of the influence of hydraulics on fish behavior on spatial and temporal scales of sub-meters and seconds. Existing IBMs for fish migration at dams, which includes forebays, tailraces, and fishways (e.g., Arenas Amado 2012; Abdelaziz 2013; Goodwin et al. 2014; Zielinski et al. 2015; Gao et al. 2016), are not suitable for this task, as they either operate on larger scales, deal with downstream migration, artificially enforce upstream directed movement, and/or do not allow trial repetition after initial failure. I will expand on these points in the IBM literature review. For now it is sufficient to know that development of a new behavior model was found necessary, which requires data. A suitable, large record of behavior data was observed as part of a 2016 study in the *ethohydraulic flume* (EHF) of BAW and BfG, the Federal Waterways Engineering and Research Institute and the Federal Institute of Hydrology, respectively. Results of the study are being prepared for publication (Schütz et al. 2017).

The resulting workload is structured by single objectives as follows.

1.3 Objectives

The main objective of this work is (1) to create a new IBM, "ELAM-de", for simulating behavior of upstream migrating fish in small spatial and temporal scale (sub-meters and seconds). The appendix *-de* is short for *Deutschland*

(Germany). The investigation focuses on fishway dimensions and species typical of the German federal waterways as they are, due to their size, crucial for connecting fish habitats through migration.

Further objectives necessary to reach the main objective are (2) to identify patterns in observed upstream moving trout behavior in the EHF with respect to geometrical and hydraulic parameters; (3) to evaluate which hydraulic stimuli and model structure can reproduce these patterns; and (4) to define meaningful application boundaries for the new ELAM-based IBM.

1.4 Outline

This work is divided into chapters along the requirements and components of an ELAM model.

Chapter 1 provides information on fish migration which motivates the need for research and explains fundamentals of the main method ELAM.

Chapter 2 reviews and analyzes fish behavior observations of upstream movement in laboratory and nature scale to find stimuli and patterns for behavior model development. Besides prior work, emphasis is put on pattern analysis of trout behavior data from the EHF.

Chapter 3 develops and tests a 3D CFD model of the EHF to generate hydraulic data for input.

Chapter 4 develops a new software framework. Recently published IBMs are reviewed to evaluate both their software frameworks and behavior models for suitability to small-scale fish behavior modeling.

Chapter 5 develops a new behavior model for upstream moving trout in a flume. It is combined with CFD model input and the new framework from chapters 3 and 4 for comprehensive calibration and validation against the patterns observed in chapter 2 to complete the new IBM, ELAM-de.

Chapter 6 summarizes the main conclusions of this work and offers recommendations for future work.

The appendices contain additional materials including instructions on running ELAM-de (appendix A), source code of custom tools (appendices B, C), additional model results (appendices D and E), and a comprehensive list of the software used (appendix F).

Chapter 2

Fish behavior observation

In this chapter, fish behavior observations are reviewed and analyzed to find stimuli and patterns for behavior model development. In the first part of the chapter, recent publications on upstream moving fish behavior observations in laboratory and nature scale are summarized and discussed. In the second part of the chapter, the 2016 behavior study in the ethohydraulic flume is introduced. Brown trout behavior data of four hydraulic setups is analyzed with newly developed methods to identify patterns for calibration and validation of the new behavior model (chapter 5).

2.1 Literature review

This section aims to identify behavior traits or patterns of upstream moving fish in laboratory and nature scale studies which can be used for behavior model development. Features of interest include hydraulic stimuli, swimming close to walls, existence of resting zones, and species/body size influence on swim speed.

2.1.1 Fish behavior in flumes

The review lists selected recent flume studies of upstream moving adult fish interacting with inhomogeneous, directed flow fields and geometrical boundaries.

Background — The behavior of fish in currents has been studied for decades using laboratory flumes (example references in Adam and Lehmann 2011, p. 17ff). Flumes are a basic element of hydraulic laboratories, as they

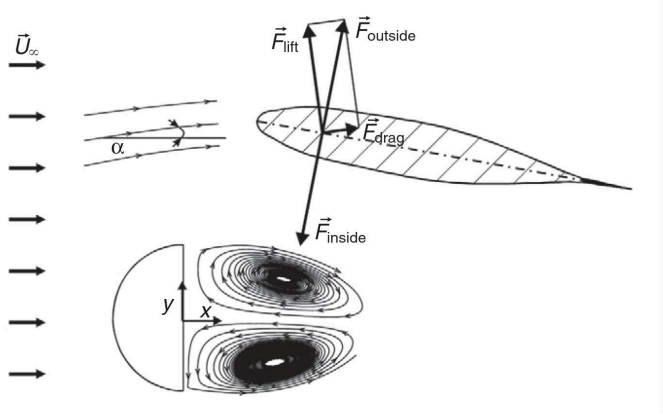


Figure 2.1: Principal forces, $\vec{F}$, acting on a fish entraining downstream a half-cylinder in flow. The angle of attack is the sum of the body angle towards x axis and the flow angle, α. Reproduced from Przybilla et al. (2010, p. 2984).

allow controlled and detailed studies of hydraulic flow properties. The same advantage applies to studies of fish behavior (review in Liao 2007), with the downside of having a large number of possible non-natural influence factors which have to be taken care of. Interestingly, first anecdotal tests with live fish at BAW were conducted as early as 1960, using a scale model (1:1.75) of a fishway (BAW 1961).

The following references are listed in chronological order.

Przybilla et al. (2010) tested 36 rainbow trout (body length BL = 14.1 ± 2.1 cm) using a flow tank. They generated a current by means of Kaplan propellers. Additionally, they computed the flow field using OpenFOAM software. The working section had dimensions of $L \times W \times d = 1.0$ m $\times 0.28$ m $\times$ 0.28 m, where d is water depth. Mean flow velocity in the tank was $u_m =$ 0.42 m/s. The authors concluded that a mechanism called *entraining* saves energy for fish swimming behind a cylinder and close to a lateral plate. Entraining means station holding with reduced body activity (i.e., energy need) using flow features. The authors supposed that fish add a lift force to the drag force experienced by tilting their bodies to the flow. Then, fish would cancel these forces by a suction force induced through increased flow velocity (and decreased pressure) between fish and a cylinder or wall (Figure 2.1). In another test, six trout spent over 70 % of observation time entraining close to a plate, which acted as a side wall.

Wang et al. (2010) described a series of tests with 20–30 juvenile brown

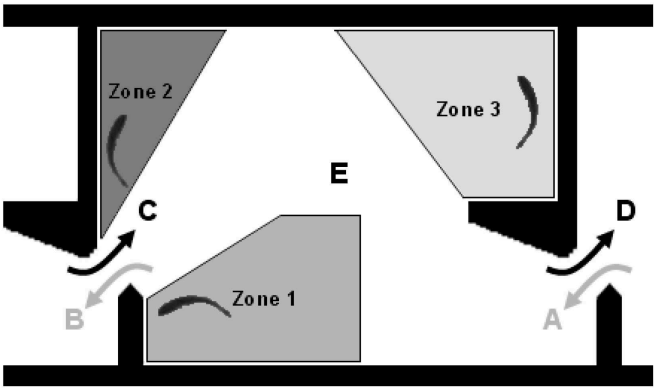


Figure 2.2: Fish stay zones in test pool, flow from left to right (Cornu et al. 2012).

trout each swimming in a model vertical slot fishway of five pools ($L \times W \times d =$ 0.75 m $\times$ 0.675 m $\times$ 0.55 m), with variable W. BL ranged between 8–12 cm. The authors identified three preferred zones in the pool, two of them lateral to the jet downstream of the slot, avoiding high velocities. A frequency of 80 % for these two zones was stated, however, no information on its data base and computation was given. The use of juvenile fish to account for model scale is not uncommon (see references in Wang et al. 2010), but the conclusions are surely limited if adult specimen were not tested (Adam and Lehmann 2011, p. 40).

Cornu et al. (2012) showed that chubs in the model fishway used by Wang et al. (2010), $W = 0.5$ m, preferred certain locations. The 60 fish used were small with a BL = 9.7–15.9 cm. Over 90 % of the detected movements through the slots of the experimental pool started or ended in an area on the right side downstream of the slot (Zone 1, Figure 2.2).

Goettel et al. (2015) tested 49 wild dace ($BL = 6.5 \pm 0.6$ cm) using 3 flow rates and 4 geometric configurations (brick walls) in a flume ($L \times W \times d = 8.0$ m $\times$ 0.5 m $\times$ 0.08 m). Shallow water was used to enforce two-dimensional behavior. Flow velocity range was $U = 0.072$ m/s $- 0.119$ m/s. Fish were reused in multiple tests. About 60 % of released fish provided useful tracks. The dace's snout positions were obtained manually by screening video footage from above the test pool, which had a 1–cm–grid painted on its bottom. They were recorded after "significant changes in swimming direction" and at holding positions. Qualitative observations showed preference for vertical surfaces (walls), shading, brick edges, and group swimming. Anal-

ysis of the turbulence levels at chosen directions (by both means of TKE, turbulence kinetic energy, and τ, Reynolds stress) showed a clear preference for maintaining adapted turbulence levels.

Sanz-Ronda et al. (2015) used a flume without installations to obtain maximum swim speeds of 36 barbels (BL = 17.9 cm) and 28 nase (BL = 61.9 cm). The flume dimensions were $L{\times}W{\times}d$ = 25 m × 0.5 m × 0.6 m and tested average flow velocities were U = 1.5 m/s, 2.5 m/s, 3.0 m/s. An observation was that the majority of fish (90.3 %) swam in a distance of 8.4–18.7 cm from either wall, avoiding both close wall contact and slightly higher velocities in the flume center. A major finding was that both species, despite not reaching their theoretical maximum distance, largely exceeded previous findings of their prolonged and burst swimming speeds with swim speeds exceeding 20 BL/s.

Rodríguez et al. (2015) acquired data of 259 barbels, nase, and brown trout (BL varied, 30 cm max.), to develop an optical tracking methodology. The fish moved upstream in a vertical slot fishway model (pool dimensions $L{\times}W{\times}d$ = 1.85 m × 1.5 m × 1.0 m). Their tracks were analyzed to obtain resting zone preferences in the pools. A position preference for the low-velocity area on the right side downstream of the slot was identified.

Kerr et al. (2016) utilized 118 brown trout, 14 of them caught in the wild (BL = 11–29 cm), for comparing the results of 2D swimming in the wakes of two different vertical cylinder arrangements in a flume (experimental area $L{\times}W{\times}d$ = 2.94 m × 1.4 m × 0.6 m). Visual cues were eliminated through sheeting the flume. Observations were made using low light cameras and infrared lighting. From the information in the paper, I calculate the average flow velocity to be $U_m = 0.15/(0.27 \times 1.375) = 0.40$ m/s. Kerr et al. (2016) proposed a new metric for drag, $D \propto U_m\sqrt{U_m^2 + \sigma_v^2 + \sigma_z^2}$, where σ is standard deviation of velocity U. The metric incorporates both mean and fluctuating velocity components to serve as a proxy for energetic cost in holding.

2.1.2 Fish behavior in dam tailraces

This section outlines challenges and findings of tracking upstream migrating fish in dam tailraces to identify behavior patterns. The only applicable way to observe fish behavior in small-to-mid scale in medium-sized rivers like the German federal waterways is to combine acoustic or radio transmitters (*tags*) in the fish with external receivers (Figure 2.3). Fish positions are interpolated

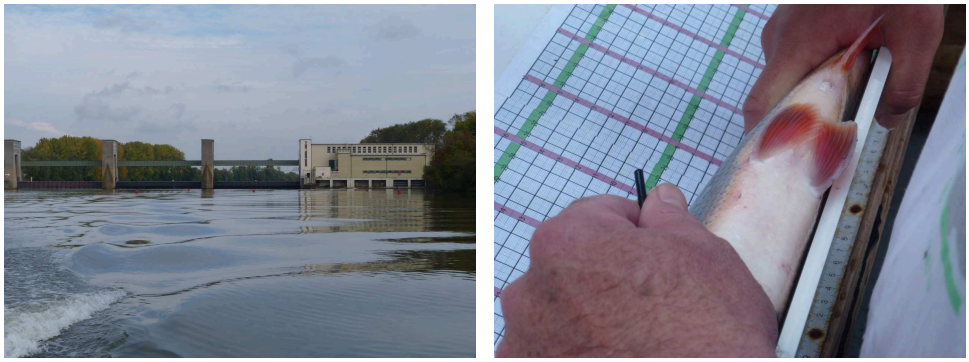


Figure 2.3: Tailrace of the Eddersheim dam on the German Main river (left), tag insertion into an anesthetized fish (right).

from run time differences of signals sent by the tags and received at multiple positions.

For downstream migrating fish, especially juvenile salmon, 3D measurements close to hydropower dams are in use for decades (McMichael et al. 2010). The technology benefits from directed flow upstream of the powerhouse and deep (> 10 m) water in forebays e.g. in the US Pacific Northwest.

For upstream migrating fish in tailraces, which are highly turbulent due to the draft tube outflow and often much shallower (< 5 m in federal waterways), measuring is much more complicated. Entrained air, turbulence, and noise affect acoustic systems. That is a probable reason why there are few data sets worldwide for upstream migrating fish close to hydropower dams. Table 2.1 summarizes parameters of four studies to demonstrate the current opportunities. Two studies showing bank-related patterns are described in detail in the following.

Lowell dam — This tailrace on the Merrimack River, Massachusetts, USA, was investigated in the past using 1D telemetry (Sprankle 2005) and, more recently, using 3D telemetry on 60 tagged adult American shad (*Alosa sapidissima*) (Hogan et al. 2012). The focus of the latter investigation was on the entrance of the local fish lift. Of all tagged fish, only two were successfully lifted. Density plots of the preferred locations in the tailrace showed a pattern in form of a horseshoe (Figure 2.4).

Três Marias dam — Private firm HTI conducted a 2D/3D tracking study in the tailrace of the Três Marias dam (65 m high), São Francisco River, Brazil (Steig et al. 2013; Suzuki 2014; Suzuki et al. 2017). The researchers

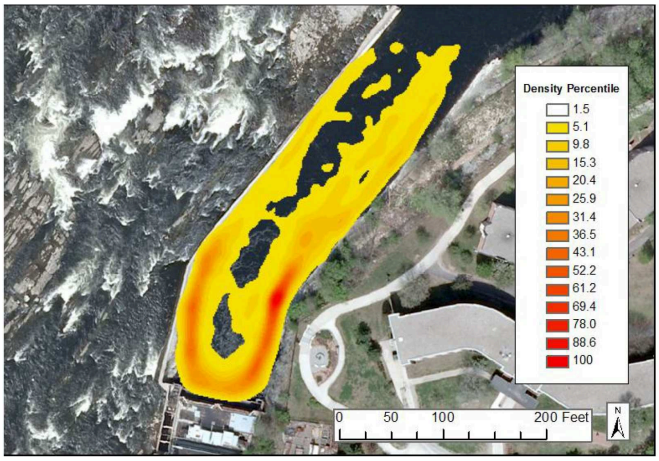


Figure 2.4: Top view of the Lowell dam tailrace showing density of shad presence (Hogan et al. 2012, p. 10). Flow from bottom to top right, 200 ft ≈ 61 m.

used an HTI Model 291 Acoustic Telemetry System with 11 hydrophones. The detection range was over 100 m despite the turbulent white water. Of the 90 fish tagged in total, 74 yielded results. Results indicated that mandi (*Pimelodus maculatus*) and curimatá (*Prochilodus argenteus*) preferred to stay in the middle of the water column. Curimatá favored the shoreline during low flows and the area immediately above the draft tube outlets and a certain part of the shoreline during high flows (Figure 2.5).

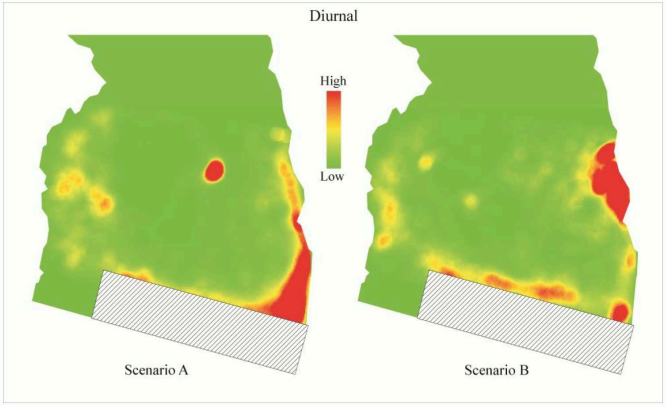


Figure 2.5: Top view of the Três Marias dam tailrace for low and high discharges (Scenario A resp. B) (reproduced from Suzuki 2014, p. 82). Color indicates diurnal relative presence of curimatá. Flow direction from bottom to top.

Table 2.1: Parameters of telemetry studies in tailraces of hydropower dams worldwide. w = typical river width at dam; d = typical water depth, subject to water level fluctuations; Dist. = Minimum streamwise distance of tracked positions from the dam; Δx = typical spatial resolution; Δt = typical temporal resolution.

Dam	Month/Year	Evaluated fish	w [m]	d [m]	Dist. [m]	Δx [m]	Δt [s]	Reference
Eddersheim	Fall 2014	60 of 5 spp.[1]	180	4–5	~350	<5	40	A. Rüter, (pers. comm., 2017)
	04–11/2015	81 ”	”	”	~350	”	”	”
	10–12/2016	61 ”	”	”	0	1.5–6	~20	Thelma Biotel (2017)
Lowell	05–06/2011	28 shad	27	5–10	0	<1	2.0–2.6	Hogan et al. (2012)
Ruswarp	09–12/2013	31 sea trout, 1 salmon	8	0.5–1.5	0	<1	2.5–2.8	Noble et al. (2014)
	10–12/2014	32 sea trout, 2 salmon	”	”	”	”	”	Noble et al. (2015)
Três Marias	10/2011–02/2012	39 curimatá, 35 mandi	160	14	0	1	3.2–5.3	Suzuki (2014)

1) Spp., species: Asp, nase, ide, perch, roach (German: Rapfen, Nase, Aaland, Flussbarsch, Rotauge).

2.1.3 Discussion

The reviewed literature on fish behavior observations provides insights on hydraulic stimuli, geometric preferences, and data availability.

Hydraulic stimuli possibly relevant for behavior model development can be divided in velocity and turbulence parameters.

High velocity jets were avoided by juvenile brown trout, chubs, and barbels, which stayed in recirculation zones on both sides of the jet in three vertical slot way models (Wang et al. 2010; Cornu et al. 2012; Rodríguez et al. 2015). Hence, low-velocity attraction could be a useful model stimulus.

Turbulence investigations revealed that dace preferred constant TKE and Reynolds stress levels in their direction choices in the study of Goettel et al. (2015). Kerr et al. (2016) showed that a new drag coefficient can explain position choices in trout. Still, the authors state that "the selection of appropriate hydrodynamic metrics that predict space use is the subject of recent debate and a cause of controversy". Goettel et al. (2015) confirm that turbulence influence on fish is an active field of research and does not allow for direct implementation in a model without comprehensive testing. TKE is selected to be tested in ELAM-de, as it is a common metric and was used in IBMs before (Gao et al. 2016).

The concept of a trade-off between energetic benefits and costs to explain position choice of brown trout was used by Kerr et al. (2016). However, this explanation is valid only for short-term holding behavior. For modeling migration and upstream movement, energetic cost need to be balanced with benefits in a larger sense, e.g. spawning pressure, which were not included in the study. Hence, this approach will not be pursued.

Location preference for the flume side wall and bottom proximity was found for trout by Przybilla et al. (2010) and for dace by Goettel et al. (2015). The entraining mechanism could contribute to this behavior as well as orientation and cover. It contrasts with the study of Sanz-Ronda et al. (2015), where wall distance was larger for adult barbels and nase. Fish tracking results from two dam tailraces suggest that shad and curimatá prefer shore vicinity while migrating upstream. I suppose that orientation, cover, and hydraulic features such as low flow velocity are reasons for this observation. However, only low-velocity attraction behavior can be tested in the IBM. A detailed investigation of entraining would require modeling the actual drag

of a fish body in the water. Because of the expected massive computational effort and because this level of detailed understanding is not required, this approach is not pursued in this work.

For the goal of identifying fish movement patterns in small-scale, high-resolution spatial and temporal observation data is required. With respect to future extensions to model behavior in a dam tailrace, a large observation area is required. There are few results for 2D and 3D fish tracking in dam tailraces (Table 2.1), including only one peer-reviewed study. They do not match the required resolution. The flume studies reviewed have in common that either movement was tracked just in a limited part of a flume, e.g. a single standard pool of a fishway, or that a larger part of the flume was observed, but movement data were comparatively coarse. The study in the EHF conducted by BAW and BfG in 2016 overcomes this issue. Hence, it is suited as the data basis for developing and evaluating the new IBM.

2.1.4 Conclusions

The first part of this chapter reviewed recent work in fish behavior observation for upstream movement close to hydraulic structures both in laboratory and nature scale. There are three main findings for model development.

1. The 2016 EHF data set was confirmed as the most suitable data source for model development, as it was the only study to feature detailed spatial observations in a large pool.

2. A preference for the area on the right side downstream of a slot was found in the flume studies of Wang et al. (2010), Cornu et al. (2012), and Rodríguez et al. (2015). I suppose that upstream migrating fish prefer resting in such areas of generally reduced velocity and/or recirculation. Hence, low-velocity attraction will be tested as a model stimulus.

3. Diverse hydraulic stimuli were used in flume studies without clear results. Thus, testing is needed in any case. I selected 3D velocity, U, and turbulence kinetic energy, TKE, for a start.

2.2 Ethohydraulic flume (EHF) study

In April 2016, active use of the new fish behavior test flume of BAW and BfG in Karlsruhe, Germany, started. It is named the *ethohydraulic flume*, EHF[1], or flume, in this work. The 2016 test series (Figures 2.6 and 2.7) compared different designs of adding auxiliary discharge into the entrance pool of a fishway using passage times of fish swimming upstream volitionally.

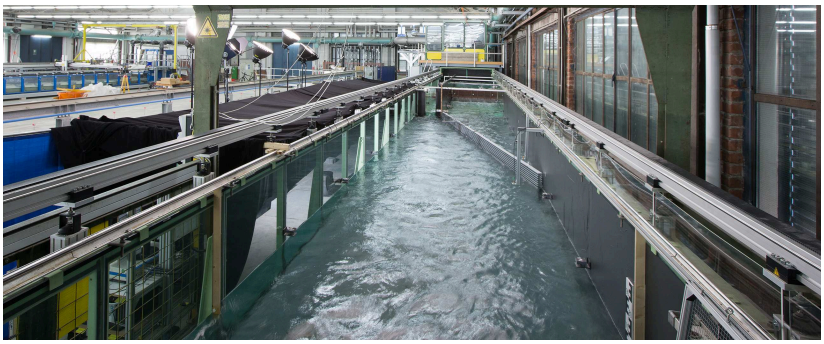


Figure 2.6: The ethohydraulic flume in the 2016 long screen configuration. Water flowing towards observer, fish moving away from observer.

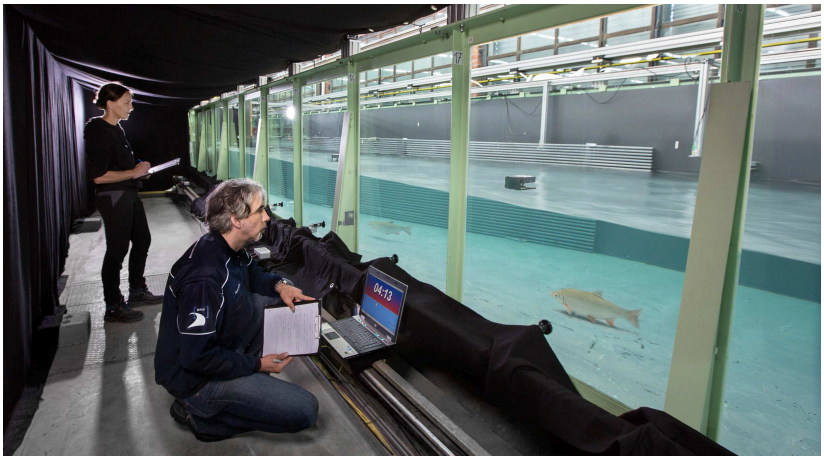


Figure 2.7: The ethohydraulic flume as seen from the observation area on the flume right, encased by black curtains. Water flowing from left to right, fish moving from right to left.

Methods, descriptions, and data from this experiment are currently being prepared for publication (Schütz et al. 2017). Information and data necessary

[1] *Ethohydraulics* is the combination of ethology and hydraulics as defined by Adam and Lehmann (2011).

for ELAM-de development were kindly provided by the authors. This section summarizes setup information directly relevant to this work. Some detail information about aims, setup, and hydraulic results of the tests can be found in Czerny and Schütz (2017).

Hydraulic methods

The EHF is a large indoor recirculating flume with a total length of 60 m. The experimental area length, width, and water depth are $L \times W \times d = 11.78$ m$\times$ 2.50 m$\times$0.60 m (Figure 2.8). Two geometrical parameters were varied in two states each. Slot walls were inserted (setup 1) or removed (setup 2) to alter flow velocity from the upstream pool, and the length of the auxiliary discharge bar screen was long (setup A) and short (setup B). Table 2.2 shows the four resulting setups combined with five species.

The fishway inflow was $Q_{\mathrm{FAA}} = 0.200$ m^3/s and the auxiliary inflow was $Q_{\mathrm{Dot}} = 0.800$ m^3/s in all setups (Figure 2.8). The outlet water level was fixed at $h = 0.60$ m from the flume bottom. The bottom was not sloped. The horizontal screen bars had a rectangular cross-section, a vertical width of 12 mm and light distance of 15 mm. The long and short screen lengths were $L_{s,A} = 6.56$ m and $L_{s,B} = 3.28$ m, respectively. The fishway and auxiliary inlet widths were $W_{\mathrm{FAA}} = 0.75$ m and $W_{\mathrm{Dot}} = 1.72$ m, respectively. Slot width was $W_{\mathrm{Slot}} = 0.30$ m. The left wall (seen in flow direction) was covered in gray to reduce visual distraction of the animals. The right wall was made of glass to allow observation. To minimize visual cues, the observation area was encased in black curtains.

Biologic methods

About 500 fish of five species representative for the German federal waterways' fish fauna were tested in total (Table 2.2). All care and procedures involving handling and holding fish were conducted as stated and permitted by the Regierungspräsidium Karlsruhe (license no. AZ 35-9185.82/A-6/16).

Fish were released in batches of three of the same species. This was the maximum number where visual tracking was possible, as they were not marked to reduce handling stress and to prevent anesthesia impact.

Four control lines A–D were defined at particular x positions (Figure 2.8). They mark the beginning (seen in fish movement direction) and end of the

Figure 2.8: Plan view of the ethohydraulic flume experimental area showing two screen configurations A and B at once. Q_{Dot} and Q_{FAA} is auxiliary and fishway inflow, resp. Fish move from right to left. Depicted are the four control lines A–D, side wall pillars and numbers (small "T"s), camera positions 1–11, and the x and y axes of the coordinate system used. $z = 0$ is at the flume bottom, pointing upward.

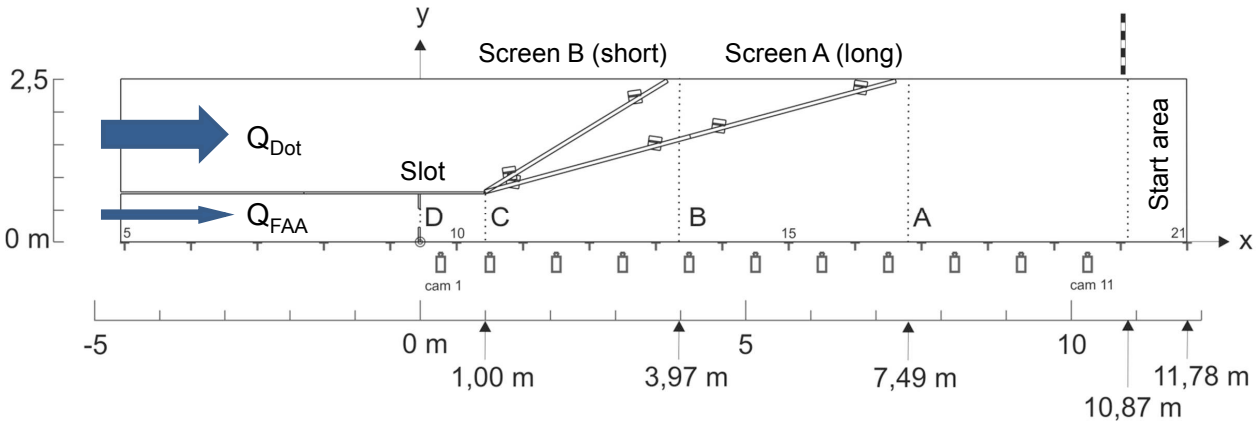


Table 2.2: Overview of data sets obtained from the 2016 ethohydraulic flume tests.

Species	English German Scientific	Nase Nase *Chondrostoma nasus*	Gudgeon Gründling *Gobio gobio*	Schneider Schneider *Alburnoides bipunctatus*	Roach Rotauge *Rutilus rutilus*	Brown trout Bachforelle *Salmo trutta (fario)*
Slot	Long screen	1a-A	1b-A	1c-A	1d-A	1e-A
	Short screen	1a-B	1b-B	1c-B	1d-B	1e-B
No slot	Long screen	–	–	2a-A	2b-A[1]	2c-A
	Short screen	–	–	2a-B	2b-B[1]	2c-B

1) discontinued

two screens, and the finish slot position. The time of the first crossing of each line was captured for each individual using a stopwatch buzzer system connected to a laptop computer.

Most fish were caught by electrofishing in small tributaries to the Rhine river including Alb, Ettenbach, Kinzig, and Rench streams. Only nase were caught in the Iffezheim fishway trap at the Rhine river. After completion of a test, fish were returned to their original water body. Fish were held in aerated tanks and set to the start area for 10 min before each test to acclimatize. After that time, the mesh screen was removed and the fish were free to move. Test duration per fish was defined as the difference between the first passage of lines A resp. D or the test end, which was at maximum 60 min after the first fish passed line A.

Spatial positions and times were noted manually by two BfG biologists for each of the three individuals. Table 2.3 shows an example. The digitalized records provided by BfG include longitudinal position at the side wall pillars (see Figure 2.8), lateral position at either side wall resp. screen, and vertical position at either bottom or surface of the water column. They were noted following any noticeable position change (event). A similar approach was used by Goettel et al. (2015) using video images. Since positions were not noted downstream of pillar 19 (except for returns to the start area) and upstream of the slot ($x > 9.74$ m resp. $x < 0$ m), these areas were excluded from analysis.

The longitudinal accuracy was about the side wall pillar distance (~1 m). The lateral accuracy was in the order of several decimeters, because wall and screen proximity are easier to spot and remaining positions were summarized as “middle”. The vertical accuracy was also in the order of decimeters; the noted positions in the water column were bottom, middle, or surface. Events were attributed to the current test minute. If multiple events happened within one minute, seconds were obtained using linear interpolation, assuming the same duration for every event recorded. The temporal accuracy is obviously decreased for multiple fast actions within a few seconds, where it is possible that not every single event was noted.

As with every manual detection method, a certain amount of error and averaging is present in the data set. It was minimized by double-checking with the second protocol (C. Schütz, BfG, pers. comm., 2017).

Additionally, fish-eye cameras were positioned at 11 spots across the

Table 2.3: Example for digitalized manual ("raw") records including interpolated times (dd.mm.yy hh.mm.ss), shoal size, and interpolated 3D coordinates, which are only used for visualization purposes. x_{raw} means the respective number of the side wall pillar; y_{raw} contains L = left, M = middle, R = right; z_{raw} contains b = bottom.

Fish ID	Time-stamp	Shoal	x_{raw}	y_{raw}	z_{raw}	x (m)	y (m)	z (m)
1e-16_F1	07.11.16 12:16:20	1	21	L	b	11.78	2.4	0.01
1e-16_F1	07.11.16 12:16:25	1	19	L	b	9.74	2.4	0.01
1e-16_F1	07.11.16 12:16:30	1	19	M	b	9.74	1.15	0.01
1e-16_F1	07.11.16 12:16:35	1	18	M	b	8.72	1.15	0.01
1e-16_F1	07.11.16 12:16:40	1	18	R	b	8.72	0.1	0.01
1e-16_F1	07.11.16 12:16:45	1	16	R	b	7.49	0.1	0.01
1e-16_F1	07.11.16 12:16:55	1	21	R	b	11.78	0.1	0.01
1e-16_F1	07.11.16 12:22:10	1	21	R	b	11.78	0.1	0.01

longitudinal x axis, facing the lateral y direction with overlapping fields of view (Figure 2.9). In this work, they were used for qualitative checking of the manual records.

2.2.1 General results

To investigate hydraulic influence in the behavior model, a comparison of setup 1 (with slot) and setup 2 (without slot) using the same species was reasonable, because hydraulic differences between these two setups were most pronounced. I selected the brown trout data sets, *1e-A* and *2c-A* (Table 2.2), as they had the highest number of successful finishers compared to the schneider and roach tests without slot, *2a* and *2b*. This was important as only finishers yield data for the whole investigation area. Brown trout results for the short screen setups *2c-A* and *2c-B* are also considered in this chapter for comparison, but not used for behavior model testing.

The mean and standard deviation (SD) of trout body length was BL = 0.26 ± 0.04 m for data set *1e* (N = 63) and BL = 0.27 ± 0.04 m for data set *2c* (N = 57). As the means are similar, it was possible to apply a fixed BL = 0.27 m for the behavior model and pattern analyses.

The behavior model presumes a basic motivation for upstream migration. However, only for fish caught directly in a fishway it is safe to assume they are motivated to migrate upstream. For trout, which were caught in the small Alb river (mean annual flow approx. 2.4 m^3/s 23.5 km upstream of mouth), the motivation was unclear. Movement activity was chosen as approximation:

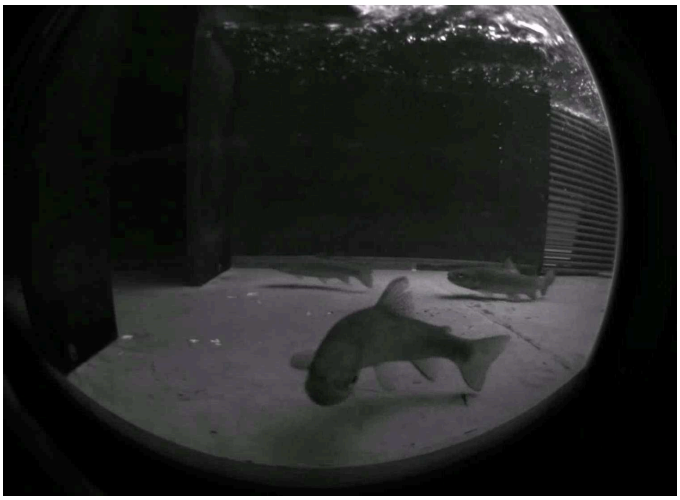


Figure 2.9: Three brown trout in front of the slot, sample view of camera 1.

Fish which did not cross line A within 30 min after removing the start mesh screen were excluded from analysis. Subjects passing line D within 90 min after start were listed as "finishers". In some cases, fish returned to the experimental area after passing the slot. These data points were also excluded from spatial analysis. After filtering, scenarios *1e-A*, *2c-A*, *1e-B*, and *2c-B* had $N = 25, 24, 24, 22$ valid fish tracks, respectively.

To allocate durations, Δt, to position data, time stamps of the recorded data were subtracted from each other (see Table 2.3). Δt was allocated to the respective earlier data point. This introduces bias towards earlier data points, especially for large Δt. A way to mitigate this bias would be to interpolate Δt between two data points. However, this would imply that fish moved with a constant speed, which is also a simplification. Hence, the approach chosen was kept.

2.2.2 Pattern analysis

Patterns (section 1.2) are well-suited for characterizing biological data sets and for evaluating associated models (Grimm and Railsback 2012).

Spatial data analysis was based on the manual position notes. From video footage and anecdotal observations of all species tested in the flume, four potential spatial patterns were identified. They were named *wall proximity* preference, *bottom proximity* preference, *turns* in main movement direction, and *group interaction*. Still, they required quantification to confirm their

existence, as described in this section. Spatial analysis was divided into 2D-horizontal (wall proximity) and 1D-vertical (bottom proximity) components. Chronological aspects were not considered as time data was pooled.

Temporal data analysis was based on the stopwatch times and yielded two additional patterns.

Note that the definitions developed in the following are valid both for observations in the physical model and numerical behavior model to enable direct comparison during model development (chapter 5). Both physical and numerical position data are analyzed using a custom MATLAB script.

Wall proximity

From subjective visual impressions of the tests, all species appeared to prefer side wall proximity. The possible pattern *wall proximity* is defined here as

- *Wall proximity:* Fish body center of mass in a lateral distance of $\Delta y <$ 0.25 m (10 % of flume width) to one of either side walls.

In the laboratory, the fish body center is estimated visually, in the behavior model it is known exactly. For analysis, the flume is horizontally divided into rectangular cells, 10 in longitudinal direction, and 3 in lateral direction. This forms three zones named “left”, “middle”, “right”, seen in flow direction. The outer zones are $\Delta y = 0.25$ m wide, the inner zone has a variable width. The screen is treated like the side walls as it forms a geometric swimming barrier.

Time spent by single fish is summed up for each cell. However, these sums strongly depend on the total time a fish was observed (e.g. Rodríguez et al. 2015). This approach can lead to wrong conclusions if total times differ widely, as in the present data set. To account for these differences and to get robust information, resulting absolute times per fish and cell, $t_{cell,i}$, are divided by the fish's total track duration, $t_{s,i}$, to get a track fraction, s_{cell}. $s_{cell} = t_{cell,i}/t_{s,i}$ is then averaged by dividing through the fish count, N, to get:

$$\bar{s}_{cell} = \frac{1}{N} \sum_{i=1}^{N} \frac{t_{cell,i}}{t_{s,i}} \times 100 \quad [\%] \tag{2.1}$$

where i is the fish iterator. The resulting proportions are that of the typical, average fish of the data set computed.

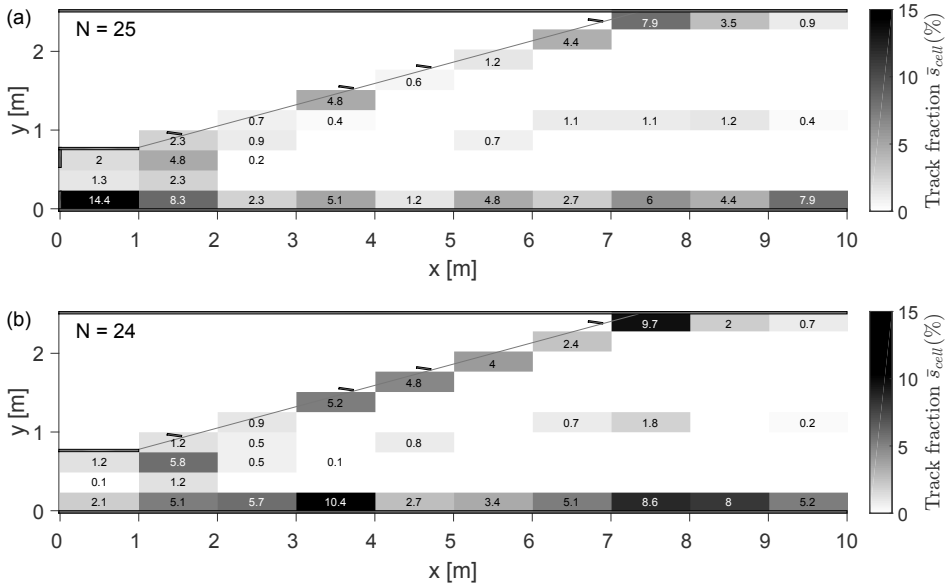


Figure 2.10: Top view of the ethohydraulic flume showing relative averaged horizontal track fractions, $\bar{s}$, of brown trout for (a) slot and (b) no-slot configurations (data sets *1e-A* and *2c-A*). Two cells are used at the screen when coordinates interpolated for visualization were ambiguous. No information are available near the start area from $x > 9.74$ m. Flow from left to right, N fish move from right to left.

Table 2.4: Horizontal position average track time fractions (mean ± standard deviation) for brown trout data sets *1e* and *2c*. Orientation facing downstream, fraction missing from 100 % is round-off error. Note that "middle" zone represents 80 % of the flume width.

Screen	Geometry	N	Left (%)	Middle (%)	Right (%)
Long	Slot (1e-A)	25	35 ± 37	8 ± 12	57 ± 38
	No-slot (2c-A)	24	39 ± 39	5 ± 10	56 ± 40
Short	Slot (1e-B)	24	11 ± 23	11 ± 21	78 ± 32
	No-slot (2c-B)	22	40 ± 43	10 ± 13	50 ± 41

From the visualization and numbers of horizontal location distribution (Figure 2.10, Table 2.4), a clear preference for the sides of the flume over the middle area is visible for both slot setups. In both cases the mean distribution is shifted towards the right hand side consisting of a glass wall. The standard deviation is high and in the order of the means, indicating large variability between individual tracks. The triplet of percentages forms the horizontal behavior pattern, P(1).

Bottom proximity

From subjective impressions of the tests, all species appeared to prefer swimming close to the flume bottom. This potential pattern is defined here as

- *Bottom proximity:* Fish body center of mass in a vertical distance of $\Delta z < 0.15$ m to the flume bottom. The same distance is chosen for the water surface, the remaining 0.30 m of the water column represent the vertical *middle*.

Tracks are averaged in analogy to the horizontal pattern. However, as the data are one-dimensional, three categories of different height are sufficient for the track average fraction, $\bar{s}_{vert}$:

$$\bar{s}_{vert} = \frac{1}{N} \sum_{i=1}^{N} \frac{t_{vert,i}}{t_{s,i}} \times 100 \quad [\%] \tag{2.2}$$

using the total number of fish tracks N, the fish iterator i, time spent on a vertical level per fish $t_{vert,i}$, and track duration per fish $t_{s,i}$.

The supposed preference for bottom proximity is observed in all trout scenarios (Table 2.5). The three percentage values form the pattern P(2).

Table 2.5: Vertical position average track time fractions (mean ± standard deviation) for brown trout data sets *1e* and *2c*. Fraction missing from 100 % is round-off error. Note that "middle" zone represents 50 % of the flume depth.

Screen	Geometry	N	Surface (%)		Middle (%)		Bottom (%)	
Long	Slot (1e-A)	25	0.6	±1.8	0.6	±2.1	98.8	±3.3
	No-slot (2c-A)	24	4.1	±16.9	0.2	±0.6	95.5	±17.4
Short	Slot (1e-B)	24	0.7	±2.2	1.7	±5.0	97.5	±6.3
	No-slot (2c-B)	22	1.3	±2.9	10.2	±22.1	88.5	±22.3

Turns

From subjective impressions of the tests, back-and-forth movements in the flume longitudinal direction were a typical behavior. This potential pattern consisted of repeated turns, which I define as

- *Turn:* Change in the longitudinal movement direction, x, leading to a displacement of $\Delta x \geq 1$ BL up to the following change in direction.

Automatic detection of turning points in the data requires special care, because it is easy to miss or to produce false positive detections. The basic idea is to compute the difference Δx between the data points in longitudinal direction and to detect sign changes between them. This yields the local extrema (MATLAB script *extremePoints.m*, appendix C).

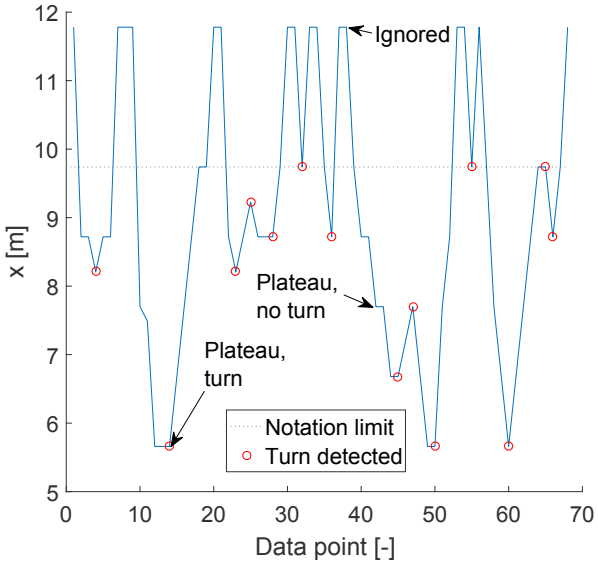


Figure 2.11: Automatic turn point detection on an example track (trout ID *1e-02_F1*). Above the notation limit at $x = 9.74$ m only visits to the start area were noted, which were excluded from the turning point count.

However, this approach does not work readily for holding phases, when the fish stops and the plotted track forms a "plateau". If the plateau is perfectly even (which is the case for the laboratory data, as only considerable changes were noted), then $\Delta x = 0$, which can be used to detect both edge points of a plateau. Next, the signs of Δx before and after the edge points can be compared to (a) filter out holding in between a movement without

direction change and (b) to merge holding phases interrupting a turn. Last, $\Delta x < 1$ BL and detections at $x > 9.74$ m are filtered out. Filtering must be performed in the end, because otherwise the Δt computation is disturbed. Figure 2.11 shows that automatic turn detection worked correctly.

Next, turn probability is averaged per flume length third by the number of tracks with more than 4 turns, N_{turn}.

$$P_{turn} = \frac{1}{N_{turn}} \sum_{i=1}^{N_{turn}} \frac{N_{turn,3rd,i}}{N_{turn,i}} \times 100 \quad [\%] \tag{2.3}$$

where $N_{turn,3rd,i}$ is the number of turns per flume third per fish, $N_{turn,i}$ is the total number of turns per fish, and i is a fish iterator.

Results are visualized in Figure 2.12 and reported in Table 2.6. More than half of the fish in all data sets performed 4 or less turns, and the number for turns for the remaining fish varied largely between 5 and 77 (data set *1e-A*). To get meaningful turn probabilities, the data sets were split into turning and not-turning fish. The latter were not considered for mean and SD calculations. The resulting turn probabilities per flume third are defined as pattern triple P(3) and the portion of tracks with few or no turns, $N_{no-turn}/N$, is identified as the (unexpected) pattern P(4).

Group interaction

Fish which stay together in a group volitionally are *shoaling*. If a fish group additionally moves both synchronized and polarized, it is *schooling* (Pitcher and Parish 1993). Shoaling advantages can include foraging, mating, and predation resistance (Viscido et al. 2004). Schooling additionally provides hydrodynamic advantages reducing energy consumption (Liao 2007).

As the fish in the EHF study were released in groups of three, it was possible to observe group interaction. The tests were not suitable to identify the motivation or reason. Thus, the term shoaling is used in its most general definition here.

For this work, shoaling is defined as follows.

- *Shoaling*: Another fish's center of mass is inside a radius of 1 BL around a focal fish's center of mass.

Note that this definition results in a detection sphere (radius here: 1 BL = 0.27 m) even though a fish body resembles an ovoid.

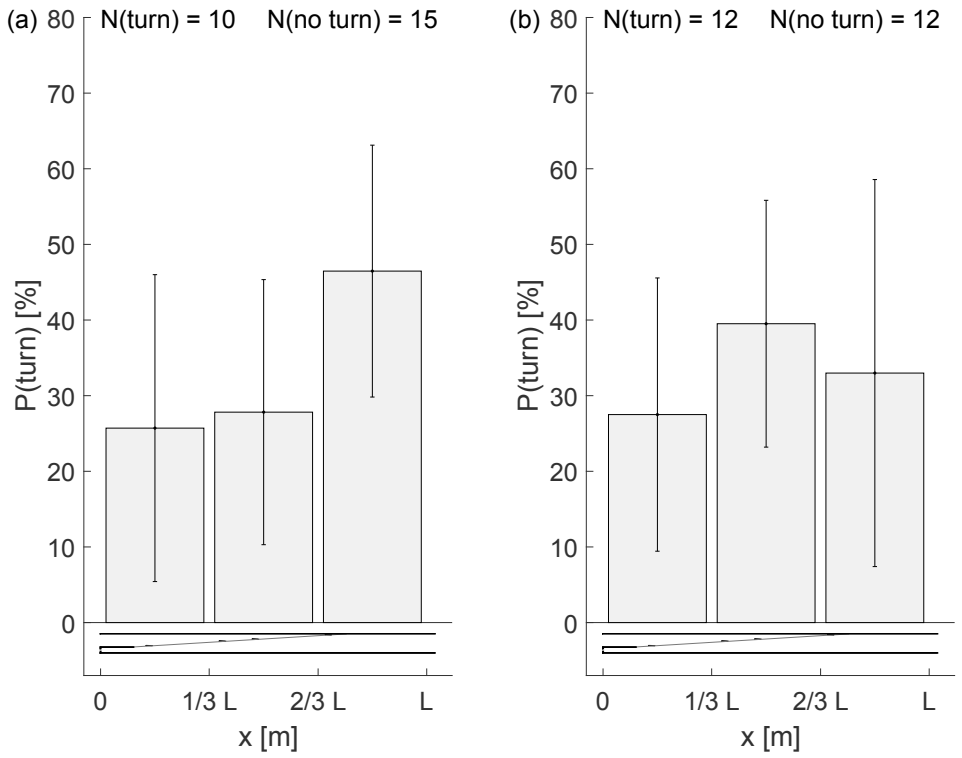


Figure 2.12: Visualization of (discrete) turn probability, P_{turn}, for data sets (a) *1e-A* (with slot) and (b) *2c-A* (without slot) per thirds of the experimental area length, $L = 9.74$ m, averaged over N_{turn} fish tracks. Flume top view shown as insert on the figure bottom. Each error bar represents two standard deviations.

Table 2.6: Values of (discrete) turn probability (mean ± standard deviation) for brown trout data sets *1e* and *2c* per thirds of the experimental area length (upstream, middle, downstream third) averaged over N_{turn} fish tracks. Fraction missing from 100 % is round-off error.

Screen	Geometry	N	N_{turn}	$\frac{N_{no\text{-}turn}}{N}$	Upstream	Middle	Downstream
Long	Slot (1e-A)	25	10	60 %	26 ± 20 %	28 ± 18 %	47 ± 17 %
	No-slot (2c-A)	24	12	50 %	28 ± 18 %	40 ± 16 %	33 ± 26 %
Short	Slot (1e-B)	24	10	58 %	41 ± 22 %	30 ± 12 %	29 ± 13 %
	No-slot (2c-B)	22	7	68 %	39 ± 06 %	33 ± 20 %	27 ± 19 %

The probability for an average fish to choose its shoal size (1, 2, or 3) is computed by averaging the correspondent probabilities per fish track:

$$\bar{P}_{size} = \frac{1}{N} \sum_{i=1}^{N} \left(\frac{t_{size,i}}{t_{s,i}} \right) \times 100 \quad [\%] \tag{2.4}$$

where N is the number of fish tracks, t_{size} is the duration of a certain shoal size, measured by the detection sphere, t_s is the total duration of a fish track, and i is the current fish track.

Results for trout are reported in Table 2.7. Significance analysis using the two-sided t-test ($\alpha = 5$ %) did not reveal consistent trends based on the hydraulic setups.

Table 2.7: Track duration spent without a shoal (single), in a shoal of two (duo), and in a shoal of three (trio). Relative fractions (%) averaged across fish tracks from data sets *1e* and *2c*. Fraction missing from 100 % is round-off error.

Screen	Geometry	Single	Duo	Trio	N (–)
Long	Slot (1e-A)	61.1	16.0	22.9	25
	No-slot (2c-A)	61.8	35.3	2.9	24
Short	Slot (1e-B)	57.4	25.9	16.8	24
	No-slot (2c-B)	44.0	23.0	33.0	22

Time and success

The arrival times at the four event lines A–D (see Figure 2.8) describe advance of the trout. They are much more precise than the spatial information analyzed above, as human interaction was minimized. However, they are limited to the longitudinal dimension (1D).

For comparison, the durations between the first A crossing and the first C and D crossings, respectively, are chosen. The are termed $\overline{AC}$ and $\overline{AD}$. The number of fish crossing these lines within the test duration are N_A and N_D. A high $\overline{AC}/\overline{AD}$ metric means small delay between C and D. A high N_D/N_A metric means good passage along the screen.

Results are reported in Table 2.8. On average, trout take longer for passing $\overline{AC}$ if no slot is present. However, this difference is not statistically significant and not used for model testing. The $\overline{AC}/\overline{AD}$ metric increases significantly if no slot is present (Schütz et al. 2017). It will be used to test

the behavior model as pattern P(5). The N_D/N_A metric describes the overall effectiveness of the setup and will be used as pattern P(6).

Table 2.8: Median passage times, $\overline{AC}$ and $\overline{AD}$, (s), finisher counts, N_A and N_D, (–), and ratios (%) of brown trout. Individuals censored if they did not cross line A within 30 min after start. Line C arrivals filtered according to line D arrivals. See Schütz et al. (2017).

Screen	Geometry	$\overline{AC}$	$\overline{AD}$	N_A	N_D	$\overline{AC}/\overline{AD}$	N_D/N_A
Long	Slot (1e-A)	49	270	25	21	18.1	84.0
	No-slot (2c-A)	119	252	24	19	47.2	79.2
Short	Slot (1e-B)	84	295	24	21	28.5	87.5
	No-slot (2c-B)	159	175	22	21	90.9	95.5

2.2.3 Discussion

Spatial and temporal behavior data of trout in sub-meter and seconds scales was analyzed and six patterns were identified (four spatial and two temporal/success patterns). It was not possible to separate movement and migration behavior in the flume. Only actively moving fish were considered to approximate the motivation to migrate. The flume investigations were not originally designed for analyses of movement patterns, which means that no statistically valid interpretation is possible. For example, the stress fish may have when being tested in an artificial flume situation was not considered. To account for that, further work is necessary. Still, the accuracy is sufficient for developing and testing a behavior model, as not exact values, but characteristic qualitative patterns are needed (Grimm and Railsback 2012). Some possible explanations for the patterns are discussed here in relation to previous work.

Conceivable explanations for horizontal distribution, pattern P(1), include visual or hydraulic orientation by the walls and screen, need for cover, and even hydrodynamic advantages of a suction force from swimming close to the walls (entraining, Przybilla et al. 2010). Visual orientation seems less important though, as qualitative wall preference was observed for trout before in a dark flume (Figure 4 in Kerr et al. 2016).

A possible explanation for the bottom proximity pattern, P(2), is the need for shelter, e.g. from aerial predators, which could let fish choose the maximum distance from the water surface possible in the shallow flume

($d = 0.60$ m). Additionally, proximity to any kind of surface, as observed for the side walls and screen, provides orientation. The aspect of reduced energy consumption due to reduced flow velocity at the bottom is not applicable here. Because of the smooth surface, velocity magnitude was not markedly reduced by bottom friction (section 3.4.2). The behavior resembles anecdotal observations on rainbow trout communicated by Przybilla et al. (2010).

The no-turn ratio $N_{no\text{-}turn}/N$, P(4), was 50 % or more for all setups. This high amount of fish with 4 or less turns indicates that about half of the trout had no problems reaching the finish line, neither because of flow velocity nor motivation to move. These fish were not considered for probability calculation, because having a low number of turns per third leads to extreme percentage values, which highly increase SD. This approach reduced the track number noticeably, making the coarse division of the flume into thirds necessary to obtain enough data for a meaningful average per third. Hence, no distinct connection between turning point probability and flume geometric features could be identified. The same is true for hydraulic features, which are strongly linked to geometry (section 3.4). I speculate that for species smaller and weaker than trout, which were not investigated here, there might be a stronger connection between turning points and flow/geometry features, as they are more prone to fatigue and turbulent fluctuations. However, more detailed observations would be required to generate enough data points for detailed spatial turn analysis.

For group interaction, I assume that the track fraction spent in groups by trout is too small for analyses. Therefore, no pattern is identified for behavior model testing.

Trout have overall high passage quotes N_D/N_A, indicating high passage effectiveness.

Increase of $\overline{AC}/\overline{AD}$ for the no-slot setups, showing a smaller delay in the slot area, could be caused by either reduced velocity, or the missing geometrical barrier, or a combination of both.

2.2.4 Conclusions

In the second part of this chapter, I presented and analyzed trout behavior data from the 2016 ethohydraulic flume study to obtain patterns for behavior model development. The brown trout data sets, *1e-A* and *2c-A*, were selected.

From the spatial analysis, four pattern metrics, P(1–4), were defined: Three triples of percentage values for wall proximity, bottom proximity, turn probability, and a scalar for no-turn probability.

Time and success analyses were more accurate than the spatial analyses, as data collection involved considerably less human interaction. Time analyses indicated that the time delay for passage through the slot was smaller (higher ratio $\overline{AC}/\overline{AD}$) for setups without a slot compared to setups with a slot. Success analysis showed the ratio of finishers describing the effectiveness of the setup. Two additional pattern metrics, P(5–6), were defined for behavior model testing.

Next, hydraulic model input needs to be generated.

Chapter 3

Hydraulic model of a flume

The goal of this chapter is to develop and validate 3D computational fluid dynamics models of the EHF to be used as input for ELAM-de. A numerical simplification to the screen geometry is tested and discussed using velocity measurements.

3.1 Computational fluid dynamics (CFD)

Building and running a CFD model of a generic lab flume is straightforward for a trained modeler, as the geometry is rectangular and the boundary conditions are known exactly. Hence, few detailed descriptions of such models are published. More studies deal with modeling the flow through fishways, which can be installed in flumes by adding pool walls and possibly a bottom slope. Recent examples include Haselbauer and Barreira Martinez (2011) and Duguay et al. (2017). They illustrate the variety of numerical and post-processing options arising from simple deviations from the rectangular flume.

For CFD modeling in this work the open source toolbox OpenFOAM[1] is used. OpenFOAM consists of numerous libraries written in C++, providing a framework for numerical calculations in continuum mechanics (Weller et al. 1998). Further included are ready-to-use applications, most of them dealing with CFD problems. A variety of turbulence models, discretization schemes and solver algorithms are provided. The acronym FOAM is short for *Field Operation and Manipulation.*

OpenFOAM works on meshes consisting of unstructured, polyhedral,

[1] Build: 2.3.1-262087cdf8db

three-dimensional cells (OpenCFD Limited 2011, p. U-127). They are referred to as *polyMesh* and allow for maximum geometric flexibility of the numerical domain. The shipped tools *blockMesh* and *snappyHexMesh* are applied for mesh generation. Details about mesh generation with these tools can be found in Gisen (2014).

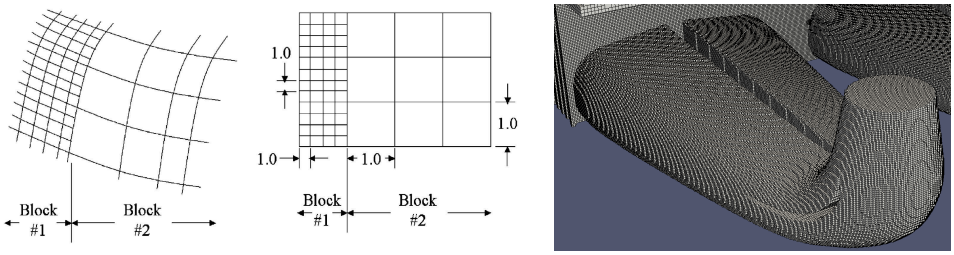
I employ the solver *interFoam*, which is included in OpenFOAM and tailored to free surface, unsteady flows (Schulze and Thorenz 2014). It discretizes the incompressible Navier-Stokes equations using the Finite Volume Method (FVM) and solves the resulting equation system in combination with a turbulence model. The Volume-of-Fluid (VOF) approach is utilized to track the free surface by simulating two phases, water and air. For turbulence closure, I chose the k-ω-SST RANS model implemented in OpenFOAM (Menter and Esch 2001). RANS (Reynolds-averaged Navier-Stokes) modeling mathematically separates turbulent fluctuations from the mean flow. k-ω-SST uses a function to blend between the turbulence models k-ϵ, used in a distance to walls, and k-ω, used in wall proximity. This approach aims to counter the weakness of k-ϵ in adverse-pressure boundary layers and sensitivity of k-ω to the ω boundary condition at free-stream boundaries (Rodi 2017).

3.2 Structured and unstructured meshes

The are two concepts for dividing space for computations which are called unstructured and structured approach.

Structured meshes (grids) allow cell access by an index value, comparable to cell access in a table. Deviations from the structure, e.g. local refinement or sudden geometry changes, are only possible using multiple blocks (Figure 3.1a). Meshes in block-ordered, structured form can be converted to contravariant space. In contravariant space, all spatial units of a block are rectangular and the same size, one. Positions in the cells are defined as parts of one in all three dimensions.

Unstructured meshes can not be transferred to contravariant space (Figure 3.1b). However, they allow much more flexibility in recreating the geometry by making use of polyhedron cells.



(a) Structured mesh in 2D Cartesian and contravariant space (Goodwin et al. 2006, p. 203).

(b) Unstructured mesh in 3D Cartesian space.

Figure 3.1: Difference between structured and unstructured meshes.

3.3 Flume CFD model and velocity measurements

The 3D CFD model domain represented the EHF physical model described in section 2.2. Four base models were developed analogous to the setups with/without slot and long/short screen. The setups without slot were are analyzed here. The base resolution of the hexahedron-dominant unstructured mesh generated using *snappyHexMesh* was uniform $\Delta x = \Delta y = \Delta z = 5$ cm, with local refinement to 1.25 cm around and downstream of the slot and vertical screen pillars. The horizontal bars of the bar screen were excluded, as they would require high local mesh refinement. The assumption that their influence on the flow is negligible small was tested as described below. Total cell count for e.g. case *1e-A* was 848 694 cells. All automatic mesh checks including aspect ratios, volumes, non-orthogonality, and skewness were passed successfully. Figure 3.2 shows the final mesh.

OpenFOAM's *interFoam* solver was applied to solve the 3D unsteady RANS equations (section 3.1). Boundary conditions were set for two inlets, the side walls and bottom, the atmosphere on top, and the outlet. The inlet flow rates and fixed outlet water level were set as in the 2016 study. No-slip conditions were set at all wall-type boundaries. The ELAM-de model required the fish outlets to be of type `patch`, internal walls to be in the group `internal`, and all other kind of walls to be of type `wall`.

For turbulence closure the $k-\omega$−SST RANS model was used. The turbulence kinetic energy, k, (or, TKE) was set dependent on the flow vector's direction at the boundary: either to $k = 0.001$ m^2/s^2 [= J/kg] for inflow or to $grad(k) = 0$ for outflow. The same was true for the specific rate of TKE dissipation, ω, with the only difference in the value chosen for inflow, $\omega = 1$ Hz.

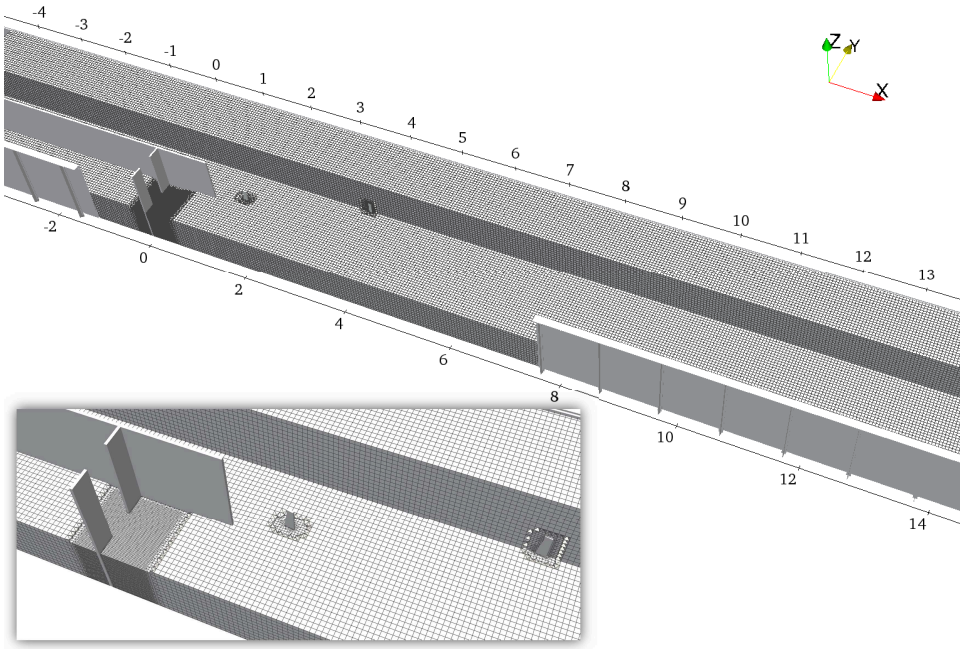


Figure 3.2: Cut through final mesh of long screen setup showing slot, pillars, and refinement regions.

The boundary layer was modeled using a simple wall function, as it was of no particular interest. A very small sand-equivalent roughness coefficient was set to represent the acrylic glass walls and bottom and the wooden internal walls, $k_S = 1 \times 10^{-5}$ m.

Current velocities of all four setups were measured in the physical model using side- and downlooking ADV (Acoustic Doppler Velocimetry) probes (Vettori 2017). Averaging time was $t = 120$ s per point up to $x = 2.10$ m because high gradients were expected around the jet, and $t = 60$ s further downstream. Horizontal point distances were $\Delta x = 0.30$ m, $\Delta y = 0.10$ m in the first zone and $\Delta x = 0.60$ m, $\Delta y = 0.30$ m in the second zone. Data were collected in two representative horizontal planes, $z = 0.40$ m and $z = 0.07$ m. The upper plane was chosen at two thirds of the water depth, $d = 0.60$ m. The lower plane was chosen to capture possible boundary effects and because fish were often observed in bottom vicinity.

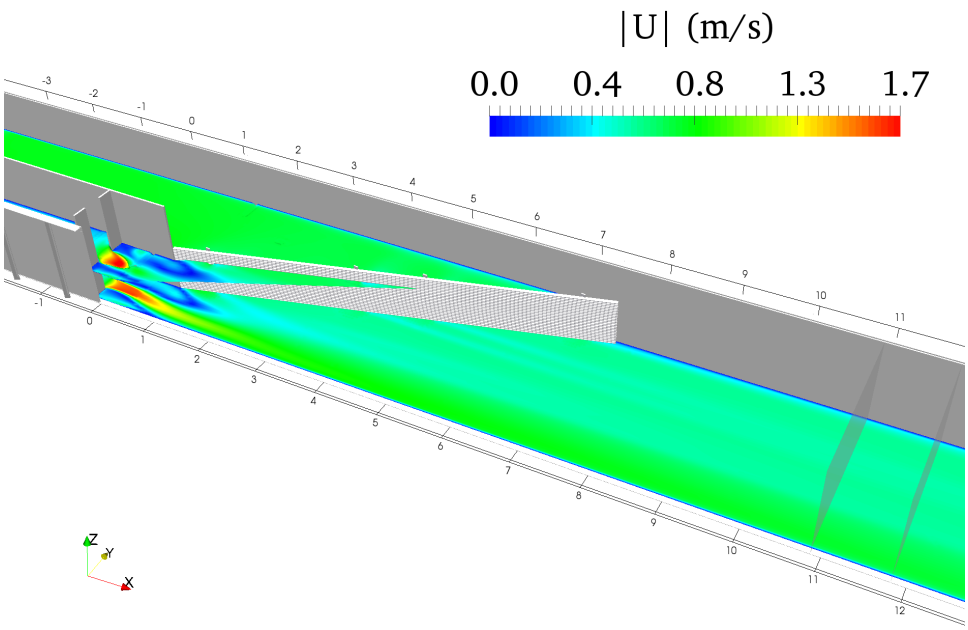


Figure 3.3: 3D representation of the numerical flume model with slot and long horizontal bar screen (added in postprocessing), and the fish start area (between the two transparent planes on the right). Horizontal planes (upper one cut) show average velocity magnitude U_m and are located $z = 0.07$ m above bottom and at the water surface, which is approx. $z = 0.60$ m above bottom.

3.4 Results and discussion

Figure 3.3 shows an example model snapshot and Figure 3.4 presents results of selected hydraulic parameters potentially suitable as model stimuli for fish orientation. $|\vec{a}|$ and TKE were used before by e.g. Goodwin et al. (2014) and Goettel et al. (2015), respectively. $|\vec{a}|$ is the only one to resemble the horizontal behavior pattern, P(1), for trout (Figure 2.10), by showing distinct changes towards the side walls and screen pillars. It will be tested as a stimulus in chapter 5.

3.4.1 Jet alignment

In the setups with slot, a jet of higher velocity formed downstream of the slot. It was visible at the water surface in the EHF physical model. However, it could not be predicted whether the jet aligned to the left (bar screen) or to the right (glass wall) even for identical boundary conditions (see e.g.

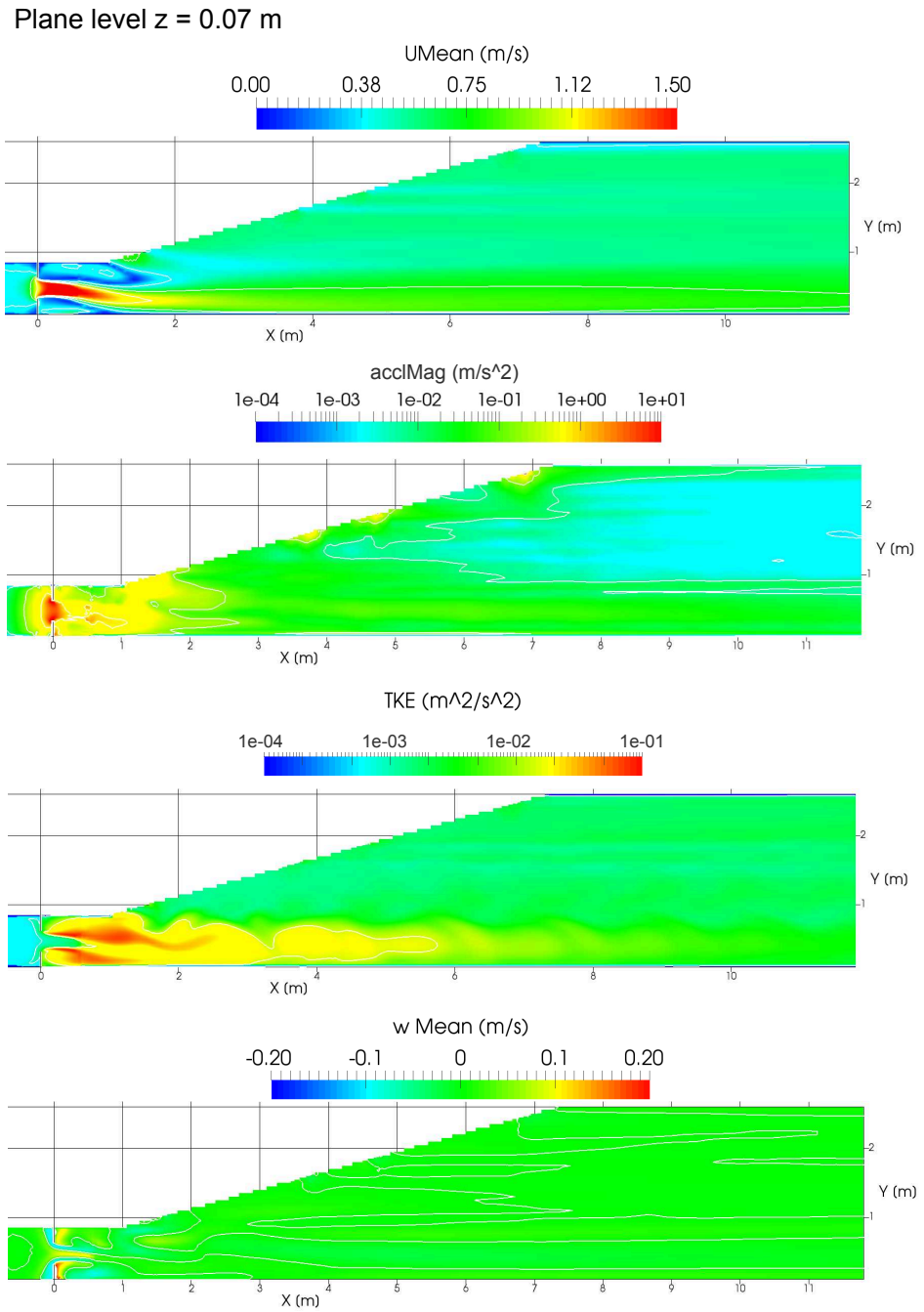


Figure 3.4: Horizontal planes located $z = 0.07$ m above bottom showing results of different hydraulic parameters potentially relevant to fish orientation. Isolines are inserted at the respective tick values, except for the min and max value. Except for turbulence kinetic energy (TKE), all parameters represent the average flow field. Note that acceleration magnitude and TKE are scaled logarithmically.

Adams and Stamou 1989). The alignment was stable, but could be adjusted by means of a wood bar, a person walking through the water, or even by an ADV probe moving through. This behavior was discovered during test *1a* (nase) and the jet was adjusted to be right-aligning for all subsequent tests, including trout.

It was not straightforward to model this alignment correctly, as the CFD model was likewise sensitive as the physical model. In the physical model tests (Figure 2.8), the total flow rate ($Q = Q_{FAA} + Q_{Dot}$) was initially set to $Q = 0.520$ m^3/s, after about $\Delta t = 180$ s increased to 0.750 m^3/s, and then after another about $\Delta t = 180$ s increased to the full $Q = 1.004$ m^3/s.[1] This was done to allow the test fish to acclimatize to the flow velocity. In the numerical model, however, the jet was left-aligning when starting with the full flow rate and a homogeneous initial velocity of $v = 0.67$ m/s (converged after $t = 200$ s, Figure 3.5). It was right-aligning when starting with $Q_{FAA} = 0.200$ m^3/s and increasing $Q_{Dot} = 0.400$ m^3/s to $Q_{Dot} = 0.600$ m^3/s after $\Delta t = 20$ s (converged after $t = 100$ *s*). To get the correct right alignment, I took the result of the latter case as an initial condition to the case with full flow rate and simulated for an additional $\Delta t = 180$ s.

Due to flow separation occurring at the end of the middle wall (confluence of fishway and auxiliary flows), the jet was fluctuating in one simulation, leading to small outflow fluctuations (Figure 3.5). To obtain representative results for comparison, all CFD model results were averaged over $t = 20$ s of simulation time until they were visually steady.

3.4.2 Two-dimensionality

A two-dimensional flow field without changes in the vertical would simplify model development. To test the ADV results for two-dimensionality, two horizontal planes of case *1e-A* at depths $z = 40$ cm and $z = 7$ cm are subtracted from each other. Results reported in Figure 3.6 show that the velocity magnitude difference in wide areas on both sides is below $\Delta U = \pm 0.15$ m/s. Larger differences up to $\Delta U = 0.35$ m/s occur downstream the middle wall. In the jet, ΔU is negative up to $\Delta U = -0.35$ m/s. These numbers are equal to 22 % resp. 52 % of the average x velocity component,

[1] The additional four liters were defined to improve control of the digit count in manual control input. They are ignored in the following.

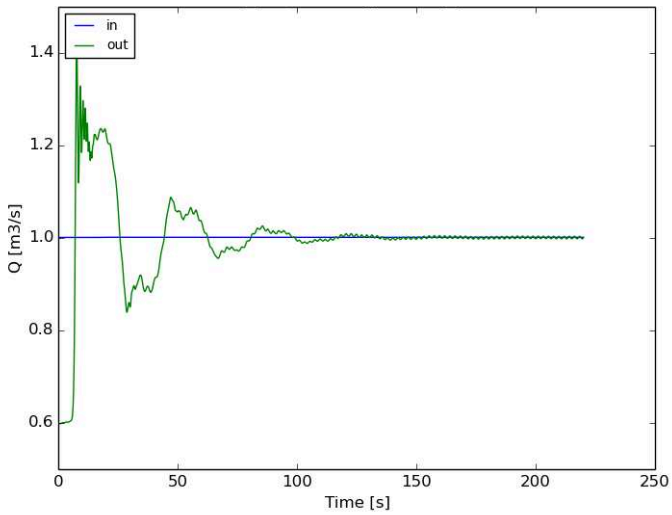


Figure 3.5: Inflow and outflow of the numerical model with $Q_{in} = 1.0$ m^3/s over time. Initialization from a previous case with $Q_{in} = 0.6$ m^3/s. The flows converged after 200 s of simulation time. Note the small, periodic fluctuation in outflow due to transient flow separation at the middle wall.

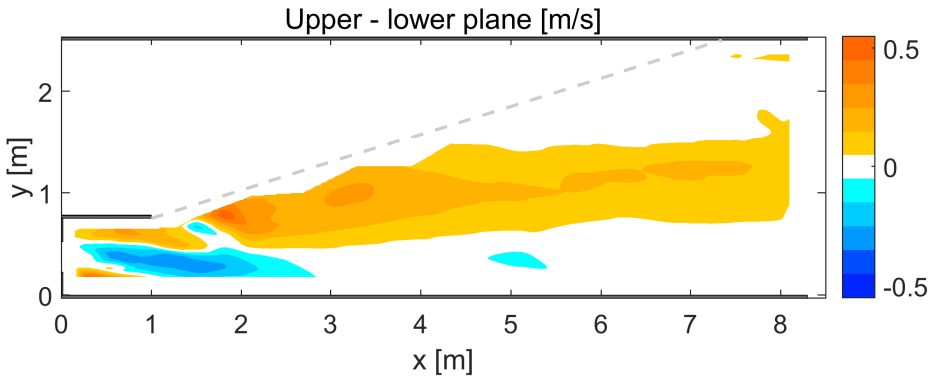


Figure 3.6: Velocity magnitude difference, ΔU (m/s), of the horizontal planes located at $z = 0.40$ m and $z = 0.07$ m above the flume bottom for scenario *1e*, "long screen with slot". Flow from left to right. Figure provided by R. Czerny.

$u_m = 0.67$ m/s. Based on this finding, the flow is not fully two-dimensional and the planes can only partly be treated as equal.

In the upper plane, velocity is higher downstream the middle wall and lower in the central jet area. As the effect is spatially limited, it is not caused by bottom friction, but probably by different jet/eddy interaction in the vertical dimension. I decided to start the following analyses with the lower plane at $z = 0.07$ m. It is the most relevant to the IBM as fish spent most of the test time close to the bottom.

3.4.3 Effects of horizontal bars and vertical pillars

Full representation of a bar screen in a CFD model requires massive mesh refinement which increases computational cost. Empirical formulas and previous experience at BAW led to the hypothesis that flow field disturbance by an inclined horizontal screen (at $u_m = 0.67$ m/s) is small and can be ignored. To test this hypothesis, three CFD simulations are compared: (a) without horizontal bars, but with vertical pillars for the long screen setup, (b) without horizontal bars, but with vertical pillars for the short screen setup, and (c) without both bars and pillars. All simulations have a slot present. The quality of the simulations is expressed as velocity difference from the time-averaged ADV probe measurement. I evaluate differences $\Delta U \leq 0.10$ m/s as desirable, up to $\Delta U \leq 0.20$ m/s as acceptable if limited in area. This equals 15 % resp. 30 % of the average x velocity component, $u_m = 0.67$ m/s. Higher differences are tolerated in the jet, which is discussed separately.

Results of the simulations and of ADV probe velocity measurements are reported in Figures 3.7 and 3.8 for the long screen and in Appendices D.1 and D.2 for the short and no-screen setups. For both long and short screen configurations with pillars, in the lower plane at $z = 0.07$ m, the differences are smaller than $\Delta U = \pm 0.2$ m/s. The only exception is the jet between $x = 0$ m and $x = 5$ m, where the simulation returns considerably higher values than the ADV measurement (Figure 3.7). In the upper plane at $z = 0.40$ m the agreement is much better (except for the jet), especially for the long screen setup (Figure 3.8). Leaving out the screen bars *and* pillars in the short screen setup leads to notably worse results, especially in the upper plane (see Figures D.3c and D.4c).

In the model of the long screen, the water level is correctly represented upstream of the slot at $z = 0.685$ cm, but is 1.4 cm low downstream of the

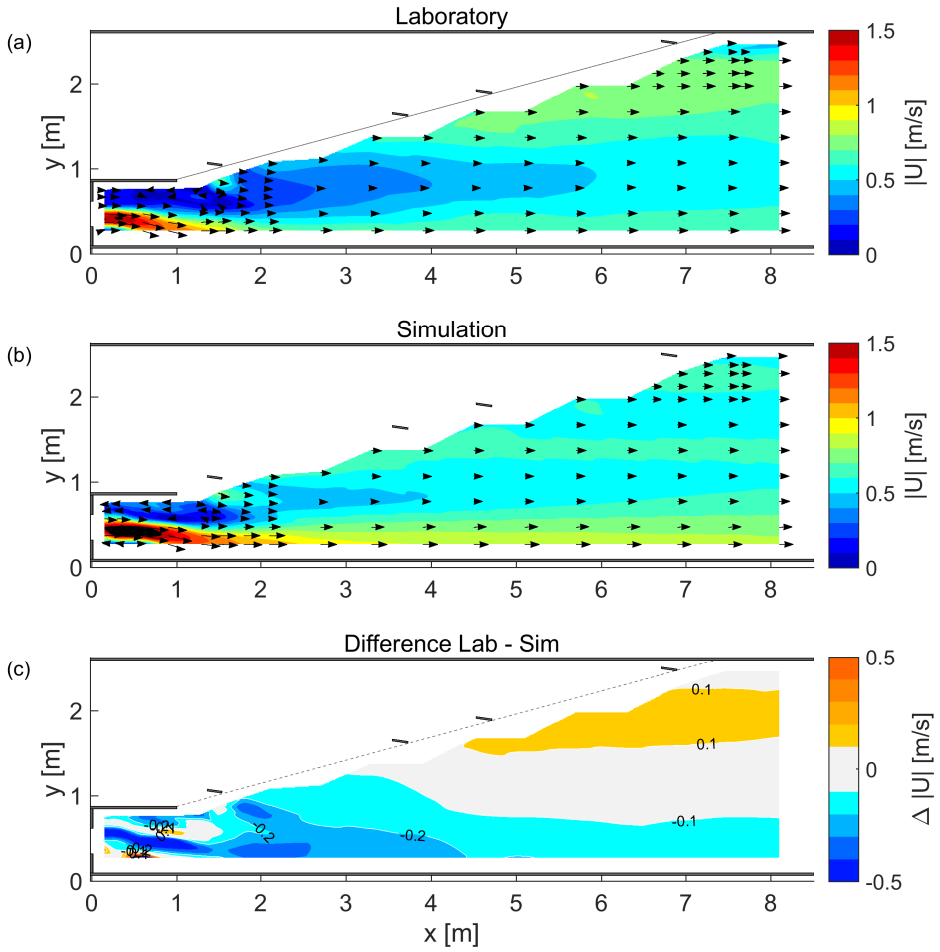


Figure 3.7: Top views showing long screen velocity magnitude of (a) the laboratory measurement (screen of bars and pillars), (b) the CFD simulation (just pillars), and (c) the difference laboratory–simulation. Plane level $z = 0.07$ m above bottom. Main flow from left to right, arrows indicate flow direction at probe locations. Contour lines every $\Delta U = 0.1$ m/s. Note that for (c), differences below $\Delta U = \pm 0.1$ m/s have been grayed out and color scale limits have changed.

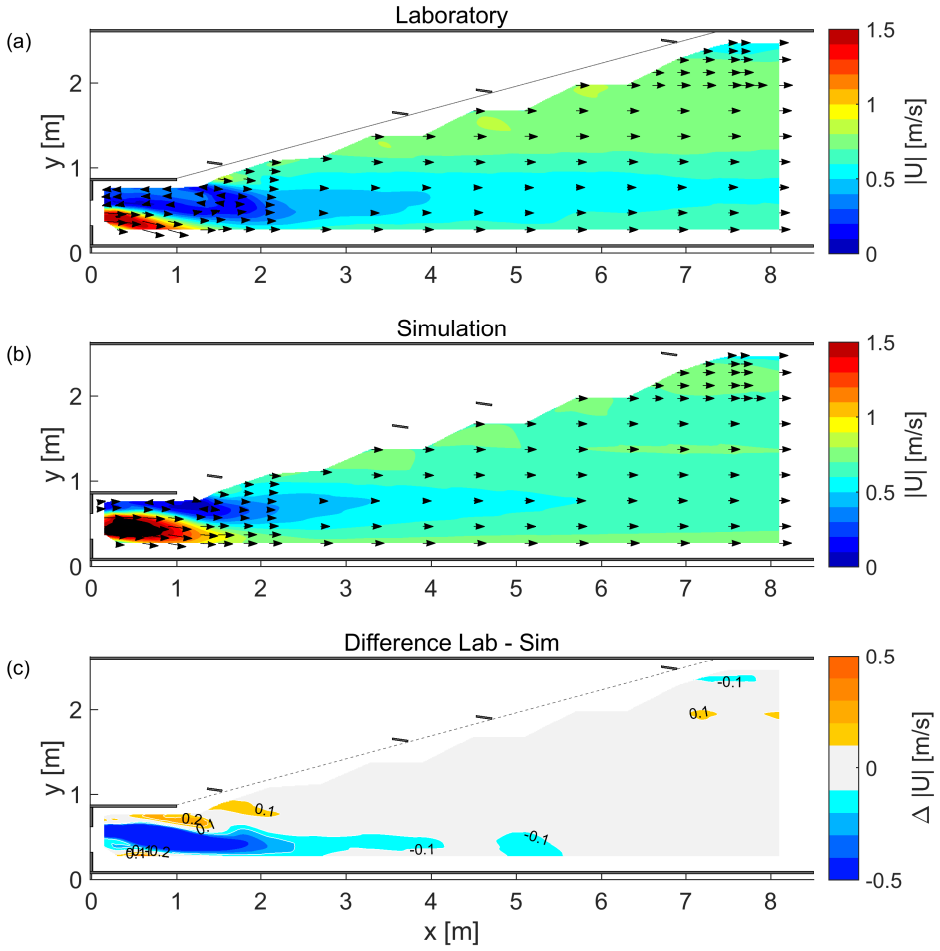


Figure 3.8: Top views showing long screen velocity magnitude of (a) the laboratory measurement (screen of bars and pillars), (b) the CFD simulation (just pillars), and (c) the difference laboratory–simulation. Plane level $z = 0.40$ m above bottom. See Figure 3.7 caption for further details.

slot at $z = 0.571$ cm ($x = 0.80$ m). This increases the head drop from $\Delta h = 0.10$ m to $\Delta h = 0.114$ m and hence boosts velocity. An rough estimation using the corrected Torricelli equation for outflow through sharp-edged orifices (Malcherek 2016) yields an excessive $\Delta u = \sqrt{g\ 0.114\ \mathrm{m}} - \sqrt{g\ 0.10\ \mathrm{m}} = 0.07$ m/s, where $g = 9.81$ m/s^2 is gravitational acceleration.

Possible factors which could explain the observed velocity differences are discussed in the following.

General methodological influence factors on the CFD side include turbulence model, mesh, and numerical schemes. On the lab side there are different ADV probe handling, insufficient seeding, geometrical imperfections, or varying boundary conditions. However, these factors were controlled for and executed according to best practice standards. Hence, they cannot be used to explain the different results.

A part of the velocity difference in the jet area can be tracked back to the simulation water level being too low immediately downstream of the slot, which increases the head drop. This could be related to the omitted horizontal bars, but can only partly explain the differences.

Another factor is deviation in the jet angle, causing differences on both sides along its way downstream. Flow separation is known to be very sensitive in CFD modeling. However, detailed modeling of the slot boundary layer would not contribute to the goal of individual-based modeling, and was not tested.

Generally, the physical resistance of the screen bars is unexpectedly small. This is best explained by the inclination of the screen towards the flow, which is discussed in the following section. The necessity of representing the pillars demonstrated in the short screen setup implies that the flow fields of long and short screen setup can not be applied interchangeably, as the pillar arrangement is different. In summary, even without horizontal bars the agreement between simulation and laboratory is sufficient for the purpose of driving the behavior model, as the observed behavior data contain fuzziness on a larger scale than hydraulics.

3.4.4 Considerations on physical resistance

Empirical formulas allow to estimate the hydraulic loss at screens, assuming one-dimensional laminar flow. To my best knowledge, relationships for horizontal screens, as present here, are not published (no mention in recent

publications, e.g. Giesecke et al. (2013) and Raynal et al. 2013). Therefore, the common formulas for vertical screens (e.g. Meusburger 2002, p. 156ff) are adapted for a horizontal screen. The hydraulic loss is computed as

$$h_v = \zeta \frac{v^2}{2g} \tag{3.1}$$

where ζ is the loss coefficient, v is the undisturbed velocity in main flow direction, and g is the gravitational acceleration. ζ is defined as

$$\zeta = k_F k_\alpha \left(\frac{P}{1-P} \right)^{1.5} \tag{3.2}$$

using the profile form coefficient k_F of Kirschmer (1925), the factor for vertically inclined inflow k_α

$$k_\alpha = \sin \alpha \tag{3.3}$$

where α is the vertical inflow angle, and the percentage of blocked cross-section P

$$P = \frac{A_{screen,proj}}{A_{cross\text{-}section}} \tag{3.4}$$

where $A_{screen,proj}$ is the area blocked by the screen, projected to the flow direction, and $A_{cross\text{-}section}$ is the wetted cross-section, perpendicular to the flow direction. The horizontal inflow angle, δ, is computed for the screens in this work as

$$\delta = \arcsin \left(L_s / W_{\mathrm{Dot}} \right) \tag{3.5}$$

where L_s is the screen length and W_{Dot} is the auxiliary inlet width (section section 2.2). v is computed as

$$v = Q/A = Q_{\mathrm{Dot}} / (W_{\mathrm{Dot}} \cdot d) \tag{3.6}$$

where d is the undisturbed upstream water depth.

Application of Equations 3.1–3.6 to the case of the long screen in the EHF yields

$$k_F = 2.42$$

$$
\begin{aligned}
\alpha &:= \delta \\
k_\alpha &= \frac{1.72}{6.56} \approx 0.26 \\
P &= \frac{12\ \text{mm}}{12\ \text{mm} + 15\ \text{mm}} = \frac{4}{9} \\
\zeta &= 2.42 \cdot 0.26 \cdot \left(\frac{4}{5}\right)^{1.5} \approx 0.454 \\
v &= 0.800/(1.72 \cdot 0.60) = 0.775\ \text{m/s} \\
h_v &= 0.454 \cdot \frac{0.775^2}{2 \cdot 9.81} \approx 0.014\ \text{m}
\end{aligned}
$$

The profile form coefficient, k_F, was chosen for a rectangle-shaped profile Kirschmer (1925). The factor for vertically inclined inflow, k_α, was chosen in replacement of the factor for horizontally inclined inflow, k_δ, to account for the horizontal orientation of the bars. The horizontal inflow angle, δ, was taken as the vertical inflow angle α. For P, the vertical pillars' resistance was neglected so that bar and gap widths were sufficient for computation. d was measured.

The downstream water level was fixed at $z_{down} = 0.60$ m in all cases. Unfortunately, precise undisturbed water level measurements were not available. Undisturbed measurements upstream of the screen at $x = 0$ m yielded mean values of about $z_{up} = 0.605$ m for the long screen (and $z_{up} = 0.620$ m for the short screen setup).

The resulting theoretical head loss using Bernoulli's equation equals a velocity difference of $\Delta v = v_1 - \sqrt{v_1^2 - 2gh_v - 2gh_p} = 0.125$ m/s, where pressure head difference $h_p = z_{up} - z_{down} = 0.005$ m. The real velocity difference downstream of the screen is in that rough order for most areas. This finding shows that the formulas are approximations good enough to allow some basic insight.

Horizontal inhomogeneity (Figure 3.7) can be explained by multiple factors. First, there are no empirical data on horizontal screens, and data on vertical screens are only available for $30° \leq \alpha \leq 90°$. Meusburger (2002, p. 172) approves of extrapolation within typical practical ranges, which is admittedly stretched here. Newer results of Raynal et al. (2013) were not considered, as they are also limited to the same range. Second, flow is not fully directed in channel main direction, as presumed in the formulas, but partly perpendicular to the screen, especially on the right hand side of the

screen where the flume width expands. For flow perpendicular to the screen, the inclination factor is $k_\alpha = \sin^2 \alpha$, which yields a smaller h_v (Kirschmer 1925) and thus further reduces physical resistance. This fact could not be considered here, as the flow angle varies over the screen length.

The insight that screen resistance increases with the horizontal angle to the flow implies that leaving out the horizontal bars has larger influence for a larger angle δ. For the short screen, $k_\alpha = 0.52$, which means the theoretical head loss is doubled. This could explain the observation that the short screen simulation shows higher velocity differences then the long screen simulation (Figures 3.8c and D.3c). Still, due to the comparatively low velocity, the head loss is negligibly small for both long and short screen setups.

3.5 Conclusions

In this chapter, numerical and laboratory hydraulic results of the 2016 EHF study were presented. Analysis of two flow fields with slot in two horizontal planes revealed that

- the flow is not fully two-dimensional in the physical model;
- it is justified to omit the screen horizontal bars for the CFD simulations of both long and short screen setups with respect to the behavior observation accuracy;
- one of the reasons for the negligible resistance is that the screen is horizontally inclined towards the flow. This implies that the long screen setup is more accurate than the short screen setup;
- the applied empirical formulas probably over-estimate the head loss as they do not account for the flume expansion downstream of the screen;
- it is not justified to omit the vertical pillars (only tested for short screen setup), which implies that long and short screen setups can not be treated as interchangeable.

The investigation was not repeated for the two flow fields without slot. I transferred the above conclusions to them under the assumption that a simpler flow field in similar geometrical constraints is easier to match. The flow fields without slot are simpler, as they do not exhibit a jet and are way more homogeneous. The local differences observed in the ADV comparison can be accepted for the purpose of simulating fish behavior.

For best possible flow field representation, I use the two CFD flow fields

of the long screen setup with pillars, but without horizontal bars, as input for ELAM-de. From the four hydraulic parameters considered as stimuli, only acceleration, $|\vec{a}|$, resembled the horizontal behavior pattern, P(1), for trout. It will be tested as a stimulus in chapter 5. But before that, the new IBM needs to get a base to run from: a software framework.

Chapter 4

Software frameworks and fish behavior models

In this chapter, the software framework for ELAM-de is developed. In the first section, recently published IBMs for fish migration at dams are summarized and the utility of their respective frameworks and behavior models for this work is evaluated. In the second section, a selected model framework is upgraded to work with unstructured meshes and OpenFOAM. In the third section, the new framework is validated.

4.1 Literature review

In this section, recently published individual-based models (IBMs) for fish movement at dams are summarized and discussed. As framework and behavior model are strongly entangled and are commonly published together, they are also reviewed together, even though behavior model information is not relevant until the next chapter. The following questions are addressed:

- Which existing software framework is most suitable to the objectives of this work, i.e. simulating behavior of upstream migrating fish in small spatial and temporal scale?
- Are there behavior models suitable to the objectives of this work?
- If not, which behavior model elements can be re-used?
- Which stimuli are relevant to behavior model development?

Upstream migration in fishways and tailraces is considered as well as downstream migration in dam forebays.

Fish behavior modeling has been a growing research field for the past decades, probably due to an increase in computer capacities (Willis 2011) as well as in ecological awareness. To structure the review, IBMs were classified by length scale, which usually correlates with further relevant behavior and CFD model features like time scale, flow field detail (e.g., resolved turbulence), and wall treatment detail. It is inevitable that some aspects of the work presented do not strictly fit into the simplified scheme, still it is helpful for presentation and discussion. The following classes of fish movement IBMs were defined:

- *Large-scale*: Ocean-scale or river system, 1D or 2D, tens of kilometers to hectometers, days to hours (not considered here).
- *Mid-scale*: Influence area of hydraulic structures ($\sim$5 river widths), 2D or 3D, hectometers to meters, hours to seconds.
- *Small-scale*: Hydraulic structures ($\sim\frac{1}{2}$ river width and smaller, 2D or 3D, meters to centimeters, minutes to parts of seconds.

Where possible, I included the purpose of the model, area considered, spatial and temporal resolutions, software framework, and species. For the development of a new model, the following information is important: calculation of swim velocity and fatigue, modeling of wall avoidance behavior, and hydraulic stimuli which trigger behavior.

Figure 4.1 presents a graphic overview for orientation. Results are discussed in the end of this section.

4.1.1 Mid-scale models

Goodwin et al. (2014) developed a comprehensive ELAM-based IBM (here: "ELAM-2014") for downstream migration. It is unique because of its long development history and the massive amount of data incorporated. Earlier versions of the model were comprehensively tested, peer-reviewed, and applied multiple times since the year 2000 (e.g. Goodwin 2004; Goodwin et al. 2006). ELAM-2014 used downstream passage route data of juvenile salmonids in 47 flow fields from seven Columbia and Snake river dams. The Columbia–Snake river system in the Pacific Northwest of the United States is the best-investigated river system in the world with respect to downstream fish migration. Extensive field research using tagged fish has been performed as early as 1971 (Bickford and Skalski 2000). Large fishways, often more than one, have been installed from the start at all eight lower Columbia and

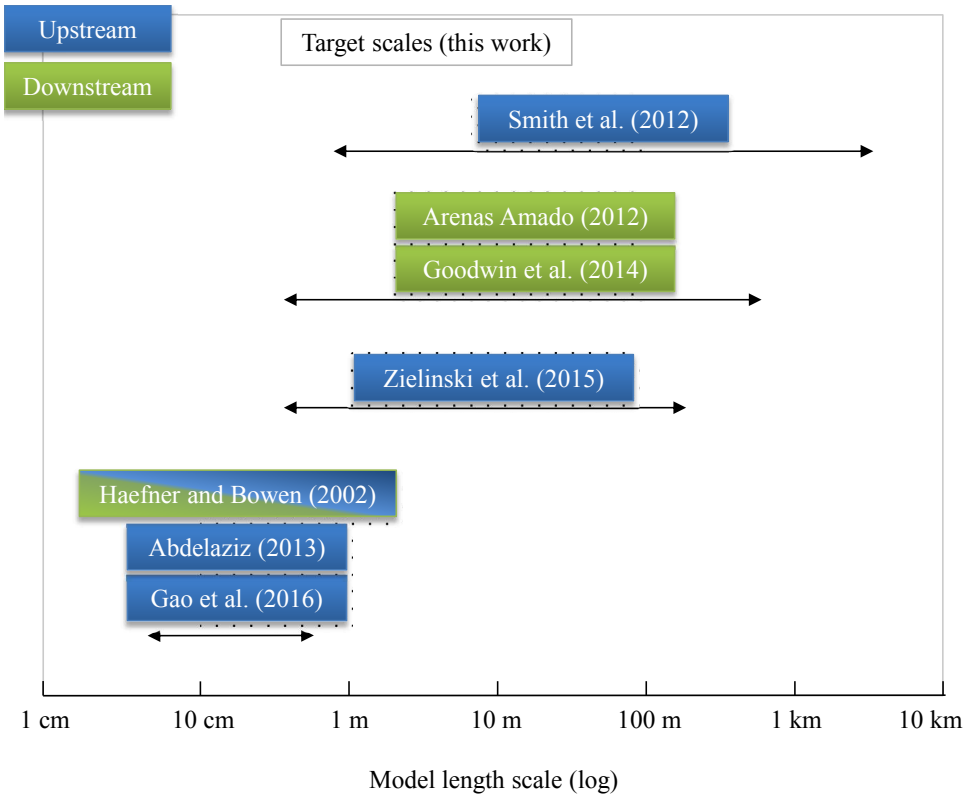


Figure 4.1: Length scales covered by the fish behavior models reviewed.

Snake river dams since construction in 1938 (BPA 1939). The popular "Ice Harbor" type fishway was developed there and the impressive number of about US $11 billion was invested alone in the years from 1985 to 2010 for nature protection purposes (BPA 2010). The number of fish annually tagged with Passive Integrated Transponder (PIT) tags was about two million in 2010 (McMichael et al. 2010), which does not include active radio tags and active acoustic tags.

CFD codes applied for ELAM-2014 were U^{2}RANS and Fluent. The 3D models of the forebays covered widths between approx. 0.5–2.2 km and lengths of approx. 1 km using block-structured grids. This allowed to use the concept of contravariant space, which enables rapid spatial computations. The behavior model comprised four behavior rules with a main time step $\Delta t = 2$ s. They can be summarized as:

- Behavior{1} (B{1}): Downstream migration in flow direction was the default behavior and worked as a biased correlated random walk (BCRW).

Random deviations of the horizontal angle were bound by ±20° and deviations of the vertical angle were bound by ±10°. Swimming speed was bound between 0.25–2.0 BL/s (Body lengths per second).

- B{2}: Attraction to higher velocity, with the goal to reduce experienced acceleration and to avoid obstacles. Angles were bound as for B{1}. If acceleration was not decreasing over a couple of time steps, swimming speed was increased up to 10.0 BL/s.
- B{3}: Blenching from increasing acceleration and swimming back against the flow. The angles allowed were getting smaller with increasing acceleration. Swimming speed was bound as in B{2}.
- B{4}: Limiting vertical movement resulting from the preceding behaviors to a certain vertical distance, equivalent to a specified (hydrostatic) pressure difference.

Individuals were able to acclimatize to changed environment parameters over time via memory coefficients. The probability for a behavior was calculated from a moving average of Boolean events. Subjective time perception in quickly changing conditions was represented as "event time" by updating behavior probability in sub-time-steps. The behavior with the highest expected utility was chosen. In case of a wall collision, the individual was set back to a position at 25 % of the boundary cell length, but only in the violated dimension(s). Energy consumption or swimming cost were not modeled.

500 individuals were released per simulation at the upstream boundary. The averaged RMSE (root-mean-square error) of passage route percentages between simulation and field data was 8.9 %. R.A. Goodwin liberally provided the Fortran 90 code used in Goodwin et al. (2014), which is gratefully acknowledged.

Smith et al. (2012) presented an ELAM model for 2D upstream migration of sturgeon in the Mississippi river. The dam considered was 894 m wide in total, and the 2D hydraulic model (USACE ADH 2D code) extended downstream up to a distance of approx. 7.5 km using a structured grid. 500 individuals were simulated using a time step $\Delta t = 2.0$ s. Swim speeds and fatigue times were estimated from literature and previous ELAM applications. Swim speed was arbitrarily increased by approx. 40 % to represent

the assumption that sturgeon preferred bottom proximity and experienced lower velocities than calculated by the CFD model. By default, individuals moved upstream with a speed of approx. 1 BL/s. Acceleration was used as a stimulus to influence the swimming direction. To account for the upstream migration direction, behaviors B{2,3} as in Goodwin et al. (2014, see above) were reversed to prefer *decreasing* velocities resp. to swim back *with* the flow. The model was calibrated to match passage rates from a study of 362 acoustically tagged fish, of which approx. 60 % moved from the tailwater into the forebay over the course of four years. A preliminary finding of the model was that passage was correlated with difference in acceleration between a planned fishway entrance and the bulk flow.

Zielinski et al. (2015) modeled a dam forebay and tailrace in 3D using ANSYS Fluent software to evaluate upstream passage rates of invasive Asian carp. Dimensions of the model were approx. 370 m × 650 m (Zielinski, 2016, pers. comm.). The unstructured mesh consisted of up to 10 million cells with edge lengths ranging from 0.25–1.50 m. A qualitative validation with ADCP data was performed. The behavior model utilized measured speed–fatigue curves of Asian carp. Individuals were forced to swim upstream and chose the direction with minimum expected fatigue, i.e. lowest velocity. Excessive velocities led to exhaustion and the individual was removed from the simulation. Field passage data were lacking so no direct validation could be performed.

Arenas Amado (2012) distinguished between dynamic and kinematic IBMs for fish. His dynamic behavior model calculated thrust and drag on the fish by solving Newton's second law, $F = ma$. It was implemented through user-defined functions in the software Fluent. Fluent was also used to develop CFD models of two dam forebays, which spanned an area of approx. 560 m in width and more than 1000 m in length. Structured meshes consisting of about 1 million nodes were generated in GRIDGEN 15.15. Four steady flow fields were modeled by averaging "selected consecutive days", no additional details given. Up to 24 000 individuals (juvenile salmon) were released into the model and compared with field data for passage route choices. Field data embraced 662 trajectories from approx. 3000 fish tagged for the first dam and passage route percentages for both dams.

Two behaviors, sustained and swim burst mode, were defined, and were mainly triggered by flow acceleration and pressure thresholds. The behav-

ior model decided through a probability distribution if the fish was kicking or gliding, a behavior pattern supposed to save energy. The modeled fish "knew" about the distance to any kind of obstacle if wall distance dropped below 15 BL, and after undershooting 5 BL it changed its direction to an angle orthogonal to the surface and switched to burst mode. Memory was not incorporated in the model. Resulting passage route prediction error was approx. 17 %.

4.1.2 Small-scale models

Abdelaziz (2013) described two models to simulate upstream fish passage in culverts and fishways, resp., of a few meters in size. The case studies' numerical modeling was performed in 2D using the Surface-water Modeling System (SMS) and Flow-3D software[1] in 2D mode. Resolutions of the rectangular grid were 0.05 m × 0.05 m and 0.01 m × 0.01 m, resp. Time step Δt was not reported.

The first model (2D) forced continuous forward progression by selecting one of three grid points in front of the individual, based on low velocity and calibrated probabilities. The velocity gradient was taken as a directly correlated indicator for turbulence. Energy cost was included by highly empirical formulae based on Blank (2008). Species considered were cutthroat and rainbow trout. 100 simulated fish paths were compared with 3 paths observed by Blank (2008).

The second model (3D) was an expansion of the first model towards the vertical axis. If the vertical velocity gradient exceeded a threshold, a vertical component pointing against the flow vector was added to the movement vector. Values for a "Normalized Error" of positions in a 2D plane were stated, but the term is ambiguous and no definition was given. Therefore, the values are not considered in this review.

Gao et al. (2016) applied the ELAM to two individually upstream migrating trout ($BL \approx 0.2$ m) in a vertical slot fishway. They computed a 2D steady-state flow field for eleven pools using ANSYS Fluent software and an unstructured mesh of 77 926 cells covering an area of approx. 21.0 m × 1.5 m. Like in Arenas Amado (2012), Fluent user-defined functions were used to im-

[1] The software used is not documented in the text, but the resulting images' design, references to the "RNG turbulence model", and the turbulent mixing length parameter (TLEN) are typical for Flow-3D.

plement the behavior model. Four possible stimuli for the behavior model were considered initially: velocity vectors, turbulence kinetic energy (TKE), turbulence eddy dissipation, and strain rate. Numerical results were tested qualitatively against two fish tracks observed in a lab and TKE was found to be the best stimulus for movement. The behavior model used a sensory ovoid similar to the one described in Goodwin et al. (2006). Swim direction was defined by the preferred TKE value, which was defined to be between 0.1–0.3 J/kg and upstream cells were preferred over downstream cells. A constant velocity over ground of 1 m/s was predetermined as a "rough estimate". The authors concluded that measured trajectories were "faithfully reproduced".

Haefner and Bowen (2002) describe an IBM developed for predicting the salvage efficiency of a fish collection facility in California, USA. Six species were considered by adjusting their preferred swim speeds and directions. The individuals were moving in a channel of 2.4 m width × 19.0 m length which was blocked by louvers (bar screens). Movement behavior and CFD were modeled in structured 2D using the NaSt3DGP package. In addition to the established sustained, prolonged, and burst swim modes (Beamish 1978), a reversing swim mode was introduced. Sustained mode was the default mode, and close to boundaries the mode switched to prolonged or burst swimming. The model was dynamic as it considered friction and profile drag on the individual and calculated forces. Fish movement directions depended on the migration direction of each species and were aligned either with or against the flow. Energy cost was incorporated through a time limit for burst mode swimming between 0–6 s. 500 individuals were simulated ten times using random energy levels and starting positions. Comparison with observed efficiencies showed good agreement for four of six species.

4.1.3 Discussion

For developing a new IBM for upstream migrating fish in the small spatial and temporal scales of a fishway, both a fitting framework and behavior model are required.

Despite frameworks being suited for reuse per definition, every IBM reviewed (except Smith et al. 2012) used a different one. This impedes transfer and exchange between the IBMs and creates duplicate work, as criticized by

Grimm and Railsback (2005, p. 271). I choose an existing framework to benefit from earlier work.

The framework of ELAM-2014 (Goodwin et al. 2014) is chosen as a development basis because it has been validated with the largest data set by far, has been published and peer reviewed multiple times, and is not bound to proprietary licenses. However, the ELAM-2014 framework is tailored to mid-scale downstream migration in forebays. Its use of structured meshes makes it unflexible in recreating small-scale geometries. This feature is particularly necessary in fishways which the present work aims for. Therefore, the framework needs to be upgraded to work with unstructured meshes, as used by the CFD software package OpenFOAM (section 3.1). OpenFOAM is chosen for compatibility to existing and future CFD models at BAW.

Development of a new behavior model from scratch is required because all behavior models reviewed lack one ore several features important to the objectives. The downstream migrating behavior model of ELAM-2014 does not account for swimming cost (e.g. energy, fatigue) which could be important for modeling upstream swimming against the flow. In addition, it assumes constant swimming, i.e. no holding behavior, which could be unsuitable for smaller scale applications (appendix E). The downstream model of Arenas Amado (2012) allows for holding. However, it partly prevents behavior emergence by using probability distributions, which limits its generality. It is validated for two cases, but bound to proprietary Fluent software. The Zielinski et al. (2015) model deals with upstream migration, but lacks the ability to simulate multiple entry attempts, direct validation, and documentation. The 2D CFD model of Smith et al. (2012) is unable to capture hydraulic 3D effects of hydraulic structures and the associated behavior model is not validated. Of the small-scale IBMs of Haefner and Bowen (2002), Abdelaziz (2013), and Gao et al. (2016), only the first one is validated comprehensively, but the authors state it needs additional testing. Taking the velocity gradient as a directly correlated indicator for turbulence (Abdelaziz 2013) is a strong over-simplification to my understanding. The models are highly tailored to the respective hydraulic structures and are probably difficult to extend to a tailrace in future work, which is important to approach fishway attraction.

Some behavior model elements can be incorporated or at least investigated in a new model. Gao et al. (2016) gives the indication to investigate

TKE as primary stimulus. To my understanding, the qualitative assessment performed in the study is prone to be arbitrary, and also flow direction could have been chosen as a stimulus. Acceleration and a pressure proxy (Goodwin et al. 2014; Smith et al. 2012) are promising stimuli as well as low velocity attraction (Abdelaziz 2013; Zielinski et al. 2015).

Swim speed was selected to be fixed at approx. 1 BL/s (Smith et al. 2012) and 1 m/s (Gao et al. 2016). Goodwin et al. (2014) used fixed rates of 0.25 BL/s, 2 BL/s, 6 BL/s, and 10 BL/s for downstream drifting, cruising, sustained, and burst swimming, respectively (their BL = 0.09 m). The dynamic models of Arenas Amado (2012) and Haefner and Bowen (2002) using Newton's second law are more physically founded, but this level of detail is not required here. Therefore, a fixed swim speed in the order of magnitude 2–4 BL/s can be used.

Wall avoidance was treated differently, but always artificial. Arenas Amado (2012) changed the swim direction to be orthogonal to the wall. Goodwin et al. (2014) reset the fish. Mortensen (2012) forced them to return to midstream. A new approach is needed for small-scale movement close to walls.

4.1.4 Conclusions

This section reviewed recent work in fish behavior modeling for upstream movement close to hydraulic structures both in nature and laboratory scale. There are some findings for model development.

- The ELAM-2014 framework was confirmed as the most suitable software framework to start development. It is comprehensively validated, extensively documented, peer-reviewed, and not bound to proprietary licenses. It needs expansion to work with unstructured meshes and with OpenFOAM output.

- Development of a new behavior model was shown to be necessary, as the existing models either operate on larger scales, deal with downstream migration, artificially enforce upstream directed movement, and/or do not allow repetition after failed trials. The necessary work is described in chapter 5.

- Diverse hydraulic stimuli were used in IBMs without clear results. Thus, testing is required. In addition to the parameters found in litera-

ture on behavior observations, 3D velocity (U) and turbulence kinetic energy (TKE) (section 2.1.4), I selected the acceleration magnitude ($|\vec{a}|$) and pressure (p) for testing in chapter 5.

4.2 New framework

Following the conclusion from the previous section, the framework of ELAM-2014 is upgraded to work with OpenFOAM (section 3.1) and with unstructured meshes instead of structured meshes (section 3.2). The new framework is part of the new IBM ELAM-de.

4.2.1 Program organization

The program organization of ELAM-de is based on ELAM-2014. The most important functional difference is the new ability to work with unstructured polyhedral CFD meshes. All interactions of the framework with the mesh in contravariant space, i.e. all spatial functions, have to be replaced by functions working in Cartesian space. Most of them are already contained in the OpenFOAM toolbox.

The model code was split to several files to enable reuse and to ease compilation. Figure 4.2 shows their dependencies in an unified modeling language (UML) diagram. The individual purposes and interfaces are:

- *ELAM-de.cpp* — Main file. Manages input, calls functions, executes time and fish number loops, stores variable arrays.
- *BehaviorRule.f90* — Behavior model, receives hydraulic and behavioral input and provides swim vector for the next time step. Vector calculations are done in *vectorRelation.f90*, which was taken unmodified from ELAM-2014. The same is true for *random.f90*, which generates pseudo-random numbers from a defined seed value.
- *sensoryPointCreate.cpp* — Creates (x,y,z)–coordinates for all sensory points of a given fish center with respect to the direction the fish is facing. I translated the code to C++ but did not change the algorithm.
- *hydroInterpolation.cpp* — Interpolates CFD field data from OpenFOAM to arbitrary cartesian coordinates, e.g. the fish location and all surrounding sensory points. Checks for boundary collisions. Upon hit,

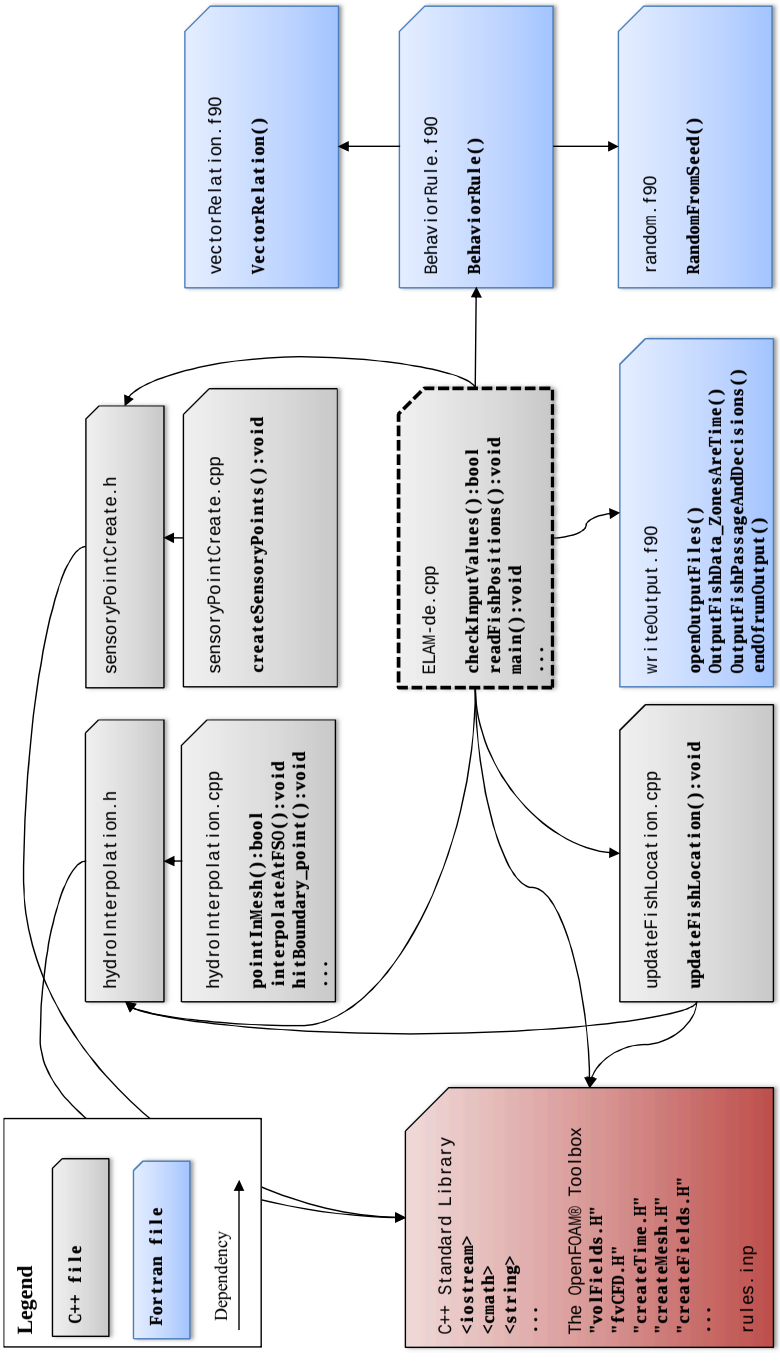


Figure 4.2: Dependencies between the C++ and Fortran files and external libraries in ELAM-de drawn in an unified modeling language (UML) diagram. C++ member functions and their return types are noted as well as Fortran subroutines. The behavior model is contained in the file `BehaviorRule.f90`, the remaining files form the model framework (section 1.2). Linked existing OpenFOAM libraries provide access to CFD model input.

returns patch type, number, and hitPoint coordinates if required. Provides fast-track functions which only check *if* a hit is detected, not where. Detects and resets out-of-bounds sensory points.

- *updateFishLocation.cpp* — Updates fish position according to only the flow vector (passive particle) or combined with the swim vector (active). Checks and corrects out-of-bounds positions, internal wall crossings, and notes normal exit or resets position, if necessary. Skips (and transfers position to next time step) if a fish has left the domain.

- *writeOutput.f90* — Writes fish movement data into Tecplot readable format. Most of the ELAM-2014 output data is omitted through comments because it is not used in ELAM-de.

Requirements for the programming language of the main file were

- the degree of compatibility with C++ (OpenFOAM) and Fortran (ELAM-2014),
- speed of interaction with these languages,
- a decent amount of popularity to
  - guarantee the existence of learning literature including Internet entries,
  - implicate a high chance of long term view popularity (10–20 years) to ensure knowledge and hardware are available to future users,
- independence from commercial licenses to simplify portability and future use,
- and simplicity to speed up implementation and readability.

I chose C++ because it satisfies all requirements except simplicity.

I kept the behavior model formulation in the function[1] BehaviorRule and in the original Fortran 90 language allowing stand-alone development, exchange of different rule sets, preserving large compatibility to the prior versions, and avoiding re-implementation of tested code. BehaviorRule is called from the main file via a C wrapper function. The interface consists of 47 arguments (section B.3), e.g. `fishNumber` or `fishSensoryVelocity`, which are processed in and returned by the function.

Language versions were Fortran 90 (ISO/IEC 1539:1991, with fixed format in FORTRAN 77 style, Chapman 2004) and C++11 (ISO/IEC 14882:2011)

[1] "Functions" in C++ are mostly equivalent to "Subroutines" in Fortran.

compiled and linked with the Intel Compiler 14.0.2 for Fortran and C++ (compatible to GNU compiler 4.4.7) on CentOS 6.6 64-bit. The Makefile developed including all necessary include paths, library paths and libraries as well as compiler flags is documented in appendix B together with development notes on the compatibility between Fortran and C++.

4.2.2 Faulty checks for mesh insideness

The routine shipped with OpenFOAM 2.3.1 to check whether a given point is inside the mesh, `polyMesh::findCell`, has some flaws which occurred during model development. Often a certain location which was clearly inside the mesh boundaries was not detected as such and treated as being outside. Example points in the mesh described in section 3.3 were $P_1 = (10.61\ 0.20\ 0.07)$ and $P_2 = (11.00\ 0.20\ 0.07)$.

I assume that the cause for this issue lies in the way the polynomial faces forming the cells are split into triangles for computation. This hypothesis is supported by OpenFOAM's recommendation never to place the *locationInMesh* point of the mesh generator snappyHexMesh exactly on a face. According to code comments (github.com 2015a,b), related issues were resolved in OpenFOAM 3.0.1. However, the fact that the recommendation is still valid in version 4.1.0 (github.com 2016) suggests that the specific issue still exists.

To overcome the issue for version 2.3.1, the only safe way was to perform a check for mesh insideness each time a new position was determined. This approach highly increased computational cost and should be addressed in future work.

4.2.3 Calculating acceleration magnitude

Flow field acceleration is a stimulus used by many fish models (section 4.1). Its calculation in OpenFOAM is developed in this section.

In Eulerian description, the three-dimensional acceleration field $\mathbf{a}$ results from applying the Lagrangian derivative[1] on a velocity vector field $\mathbf{U}$:

$$\mathbf{a} = \frac{D\mathbf{U}}{Dt} = \frac{\partial \mathbf{U}}{\partial t} + u\frac{\partial \mathbf{U}}{\partial x} + v\frac{\partial \mathbf{U}}{\partial y} + w\frac{\partial \mathbf{U}}{\partial z} \tag{4.1}$$

[1] Other terms common include *material*, *total*, or *substantial derivative*. I chose *Lagrangian derivative*, as it directly refers towards the concept used in its derivation.

For a steady-state velocity field the temporal acceleration term drops out, and the equation for advective acceleration[1] remains:

$$\mathbf{a} = u\frac{\partial \mathbf{U}}{\partial x} + v\frac{\partial \mathbf{U}}{\partial y} + w\frac{\partial \mathbf{U}}{\partial z} \tag{4.2}$$

It is commonly written more compact in full vector notation as

$$\mathbf{a} = \mathbf{U} \cdot \nabla \mathbf{U} \tag{4.3}$$

where ∇ is the Nabla operator, and $\nabla\mathbf{U} = grad\ \mathbf{U}$ denotes the gradient field of the velocity vector field $\mathbf{U}$. To calculate acceleration from OpenFOAM output, I wrote the tool *acclMag* (source code in Appendix B). Due to the missing transpose signs in Equation 4.3, special care was necessary to ensure the right order and results, as described in the following.

A gradient field of a vector field is mathematically described by a tensor of second rank, which is represented in $\mathbb{R}^3$ through a 3×3 matrix. This gradient matrix (or, transposed Jacobian matrix) is computed as dyadic product

$$\begin{aligned} \nabla\mathbf{U} &= \nabla \otimes \mathbf{U} = \nabla \cdot \mathbf{U}^{\mathsf{T}} \\ &= \begin{pmatrix} \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \\ \frac{\partial}{\partial z} \end{pmatrix} \begin{pmatrix} u & v & w \end{pmatrix} = \begin{pmatrix} \frac{\partial u}{\partial x} & \frac{\partial v}{\partial x} & \frac{\partial w}{\partial x} \\ \frac{\partial u}{\partial y} & \frac{\partial v}{\partial y} & \frac{\partial w}{\partial y} \\ \frac{\partial u}{\partial z} & \frac{\partial v}{\partial z} & \frac{\partial w}{\partial z} \end{pmatrix} \end{aligned} \tag{4.4}$$

Untransposed vectors are defined to be ordered column-wise and transposed vectors are defined to be ordered row-wise.

[1] *Advective* and *convective acceleration* (and *transport*) are often used interchangeable in hydraulics. I chose *advection* as it is more precise, denoting passive transport of flow quantities (or particles) through large-scale motions of the flow, whereas *convection* describes directed transport caused by any source, be it advection, diffusion, dispersion, or a temperature gradient etc. For further clarity: Diffusion is the omnipresent, isotropic mixture caused by random molecular motions (Brownian motion). Dispersion is anisotropic mixture through advection. Dissipation is a general term for energy converting processes.

Using the gradient matrix, Equation 4.3 can be written as

$$\mathbf{a} = \left(\begin{pmatrix} u & v & w \end{pmatrix} \begin{pmatrix} \frac{\partial u}{\partial x} & \frac{\partial v}{\partial x} & \frac{\partial w}{\partial x} \\ \frac{\partial u}{\partial y} & \frac{\partial v}{\partial y} & \frac{\partial w}{\partial y} \\ \frac{\partial u}{\partial z} & \frac{\partial v}{\partial z} & \frac{\partial w}{\partial z} \end{pmatrix} \right)^{\mathsf{T}} \tag{4.5}$$

Now the missing transpose signs can easily be added to Equation 4.3.

$$\mathbf{a} = (\mathbf{U}^{\mathsf{T}} \cdot \nabla \mathbf{U})^{\mathsf{T}} \tag{4.6}$$

Now that the order is proven to be correct, the vector field $\mathbf{a}$ is summarized into a scalar field and drops the outer transpose sign by applying the magnitude function to Equation 4.6:

$$|\mathbf{a}| = \sqrt{a_x^2 + a_y^2 + a_z^2} = |\mathbf{U}^{\mathsf{T}} \cdot \nabla \mathbf{U}| \tag{4.7}$$

Equation 4.7 is easily represented in OpenFOAM by the code

```
1 acclMag = mag(U & fvc::grad(U))
```

In the OpenFOAM framework, there is no command to define a vector's orientation in row or column, as the underlying storage arrays are identical. Still, the order is important, as demonstrated by switching the above computation to `fvc::grad(U) & U`, which gives a differing result, but no error. (For tensors of *rank* > 1, however, a transposition command is needed and available.) The definition of the gradient command `fvc::grad(U)` in OpenFOAM as described in OpenCFD Limited (2014) is consistent with the explanations above. For further details about derivations of the 3D acceleration field see Munson et al. (2013, p. 166ff).

The results depend on the numerical discretization scheme (*gradScheme*) chosen in the *fvSchemes* file, which is needed for computation of the spatial derivatives. For the standard Finite Volume *Gauss* integration scheme, three different interpolation schemes were tested: *linear*, *cubic*, and *midPoint*. Further, the included schemes *cellMDLimited-GaussLinear1* and *leastSquares* were tested. Testing with a conventional upwind scheme was not possible due to implementation idiosyncrasies of OpenFOAM 2.3.1. The total of five resulting fields (Figure 4.3) are very similar to each other, with slightly in-

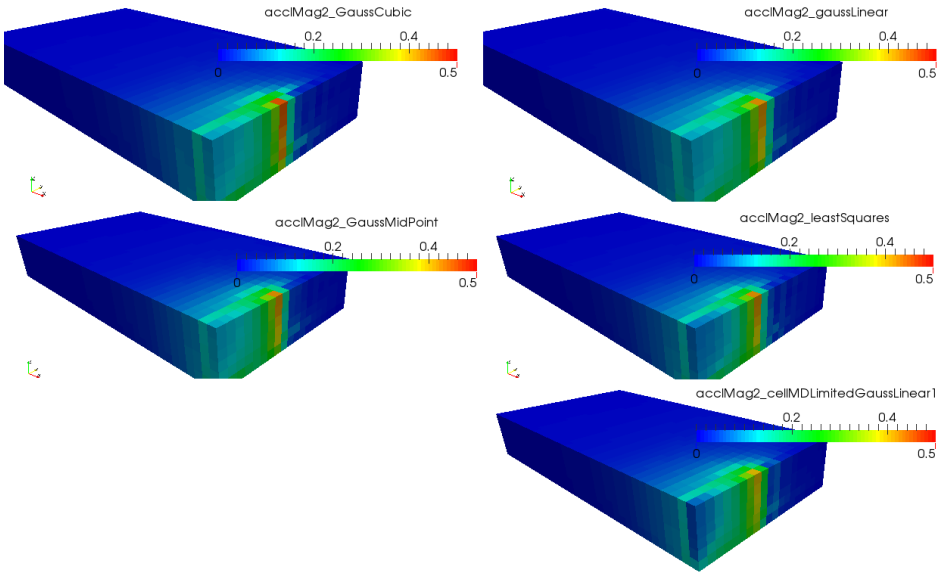


Figure 4.3: Acceleration magnitude of a velocity field (from section 4.3) computed using five different numerical gradient schemes.

creased maximum values for *Gauss cubic* and slightly decreased minimum values for *cellMDLimited-GaussLinear1* schemes. Eventually, I chose the *Gauss linear* scheme, equivalent to central differencing, for further use, as it is recommended for most cases (OpenCFD Limited 2014, p. U-117).

The postprocessing software ParaView allows to compute spatial derivatives (i.e., a gradient field) through the filter *ComputeDerivatives*. There are small differences in the results compared to OpenFOAM, possibly due to the schemes used. However, I decided to stick to OpenFOAM's computation, as the implementation is more transparent and it is integrated in the workflow.

4.3 New framework validation

To demonstrate that the framework code changes from ELAM-2014 to ELAM-de are free of programming errors, the old and new frameworks must produce identical results using the same ELAM-2014 behavior model and identical OpenFOAM CFD input. It is tested in this section by defining a test case for fish downstream migration, consisting of a CFD model with a quasi-structured mesh and a behavior model.

4.3.1 Test case CFD model

The 3D CFD model domain was a box with dimensions 20 m × 20 m × 3 m filled with water (1 phase, no air). It resembled the forebay of a hydropower dam with the outlets being a turbine intake and a small bypass (Figure 4.4). A constant velocity of 1.0 $\frac{\mathrm{m}}{\mathrm{s}}$ was set at the inlet, equal to a total flow rate of 60.0 $\frac{\mathrm{m}^3}{\mathrm{s}}$. A large and small outlet were placed asymmetrically on the opposite side to produce an asymmetrical flow field. An obstacle was placed at the center to increase flow acceleration. The obstacle consisted of a porous medium, defined using the OpenFOAM formulation of the Darcy-Forchheimer law. The coefficients were chosen arbitrarily, with the only goal of creating a high resistance, to $d = 1 \times 10^{10}$ $\frac{1}{\mathrm{m}^2}$ and $f = 0$ $\frac{1}{\mathrm{m}^2}$. Using a porous medium instead of solid walls released me from the need to split the domain into blocks. The top was modeled as a smooth wall, or *rigid lid*. Single-phase flow, downstream directed migration, and the structured mesh consisting of 9600 hexahedra ($\Delta x = \Delta y = \Delta z = 0.5$ m) were all chosen for compatibility with ELAM-2014. I chose the *interFoam* solver and RANS turbulence modeling with the $k - \omega - SST$ model (section 3.1).

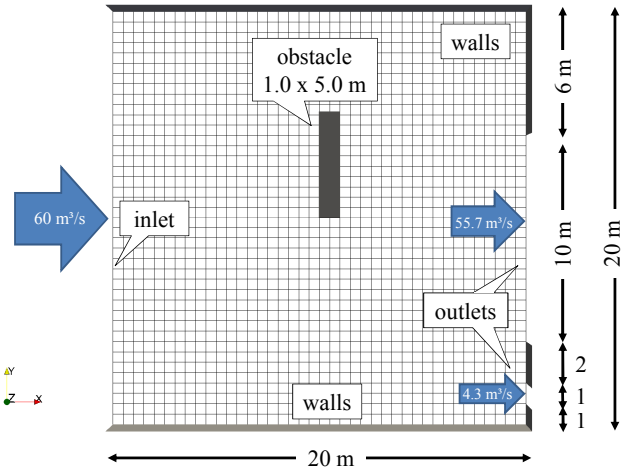


Figure 4.4: Top view of the test case 3D CFD model geometry, mesh, boundary conditions, and resulting outflows. The origin is located in the bottom left corner of the model.

4.3.2 Test case behavior model

The ELAM-2014 behavior model parameter set used here was calibrated to the Lower Granite Dam forebay on the Lower Snake River, Washington, USA, which is 30.5 m in height and 975 m in width (appendix of Goodwin et al. 2014, pp. 20, 34). It was necessary to change the thresholds listed in Table 4.1 to elicit all fish behaviors in the smaller CFD model used here.

In the ELAM-2014 framework, a boundary violation can be treated easily in contravariant space. If a fish or sensory point is outside the domain, it is reset to $\frac{1}{4}$ of the cell length in the respective dimension. This behavior was reproduced for this test case in the ELAM-de framework by resetting the fish to $\frac{1}{4}$ of the respective dimension of the movement vector between old fish position and wall hit point. The side walls, longitudinal walls, and air were appropriately separated.

Three fish were arbitrarily positioned close to the inlet (left hand side) in two different depths (Table 4.2).

The drawback of ELAM-2014, which requires to split the domain into multiple blocks for multiple exits, could be ignored for the comparison, as all fish were guided by the flow and did not try to move through the walls on the right hand side. Therefore, the simplification of defining the whole right hand side including the walls as one single exit was justified.

Table 4.1: Threshold values changed for compatibility check.

	Lower Granite	This work
Utility for behavior B{2}	0.5	0.275
Acceleration threshold, B{2}	0.8373	0.1
Acceleration threshold, B{3}	0.89	0.15

Table 4.2: Fish positions (in m) for the test case in ELAM-2014 as defined in the input file `fishPositions.inp`.

fishNumber	x	y	Surface distance
1	0.01	5	–0.9
2	0.01	10	–2.4
3	0.01	18	–0.9

4.3.3 Data conversion

To enable the ELAM-2014 code to read OpenFOAM results, the latter had to be converted to a suitable binary format. First, ParaView 4.3.1 was used to write field data for selected variables at the mesh points to a .CSV file (comma-separated value). Precision was 5 places and *Field Association* was *Points*. Next, LibreOffice Calc was applied to sort and streamline the data and to output them as a new .CSV file. The variables considered were x, y, z for 3D position, u, v, w for 3D flow velocity, p for pressure, k for turbulence kinetic energy, $acclMag$ for advective acceleration magnitude, and $STRXYZUVW$ for total hydraulic strain as defined in Goodwin et al. (2006). The listing below shows example lines.

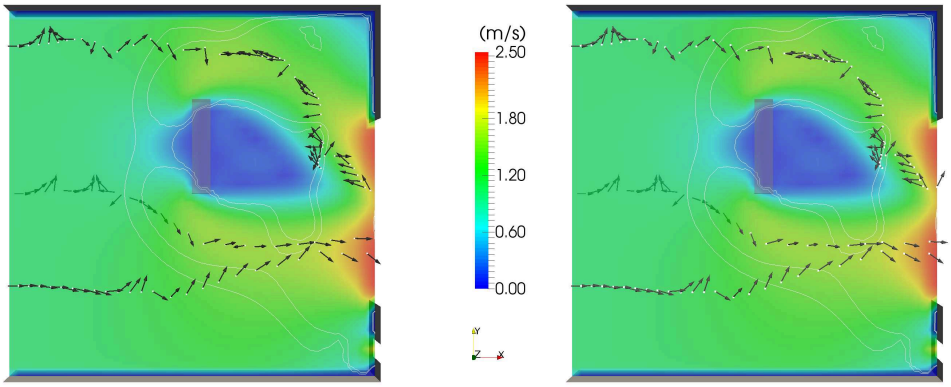
```
1    x,y,z,u,v,w,p,k,acclMag,STRXYZUVW,unused
2    0.00,0.00,0.00,0.17,0.00,0.00,28369,1.00640,0.01495,0
3    0.50,0.00,0.00,0.00,0.00,0.00,28767,0.00786,0.00000,0
4    1.00,0.00,0.00,0.00,0.00,0.00,28748,0.00527,0.00000,0

6160 4.00,13.5,1.50,0.50,-0.02,0.01,15238,0.72136,0.00249,0
```

Data were transformed to Tecplot block-structured format using the MATLAB script `ODStoTEC.m`, resulting in the file `griddata.dat`. Boundary definitions were stored in the file `connect.asc`. The whole right face of the block had to be defined as an exit route, despite the wall parts, due to ELAM-2014 model limitations. Both files were transformed by the Fortran tool `CFDtoHyd` to produce the binary file `zone1.hyd`. `CFDtoHyd` was compiled from the provided file `NFS_CFDfromASCII_to_Binary_FLUENT.f` (Goodwin, 2015, pers. comm.). Finally, `zone1.hyd` and `SimSettings.inp` were fed into the ELAM-2014 executable.

For the ELAM-de framework, data preparation as described above was not necessary, as the code was able to directly access the variable fields written by OpenFOAM. Note that the fish z coordinate in the input file `fishPositions` was written in absolute coordinates, with the z axis pointing against gravity. This was in contrast to ELAM-2014, where it was relative to the water surface. Other input values were identical to the ELAM-2014 test run.

4.3.4 Results and discussion

The ELAM-2014 model took 51 time steps equivalent to $t = 25.5$ s simulated time to complete, i.e. until the last fish had left the domain through the outlets. In the ELAM-de framework, fish #3 needed one additional time step to leave the domain. In both frameworks, the tracks showed directed downstream migration, i.e. towards flow direction and with avoidance of extreme low and high velocity zones. Some sudden orientation changes observed in the data were probably caused by calibration values not fitted to the current test case. As I conducted a relative comparison between two cases with the goal of obtaining equal results, calibration did not matter here. The resulting tracks from the ELAM-de framework (Figure 4.5a) agree visually perfect with the tracks from ELAM-2014 (Figure 4.5b).



(a) ELAM-2014 **(b)** ELAM-de

Figure 4.5: Top view of the two test case models showing fish positions (vector bases) at all time steps and intended swimming direction and speed (vectors) for the subsequent time step for three fish. Flow direction is from left to right, and the filling color denote water velocity, U_m, in the middle of the 3 m deep water column. White isolines show threshold acceleration magnitude values of 0.10 m/s^2 and 0.15 m/s^2, respectively.

Quantitative comparison shows slight deviations in the order of decimeters starting at time step 23 for fish #2 (x direction) and at time step 6 for fish #3 (z direction) (Figure 4.6). The reason are subtle differences in flow field processing between the two codes: ELAM-2014 transfers structured 3D data to contravariant space to perform faster spatial calculations (Goodwin et al. 2006), while the ELAM-de framework relies on interpolation schemes

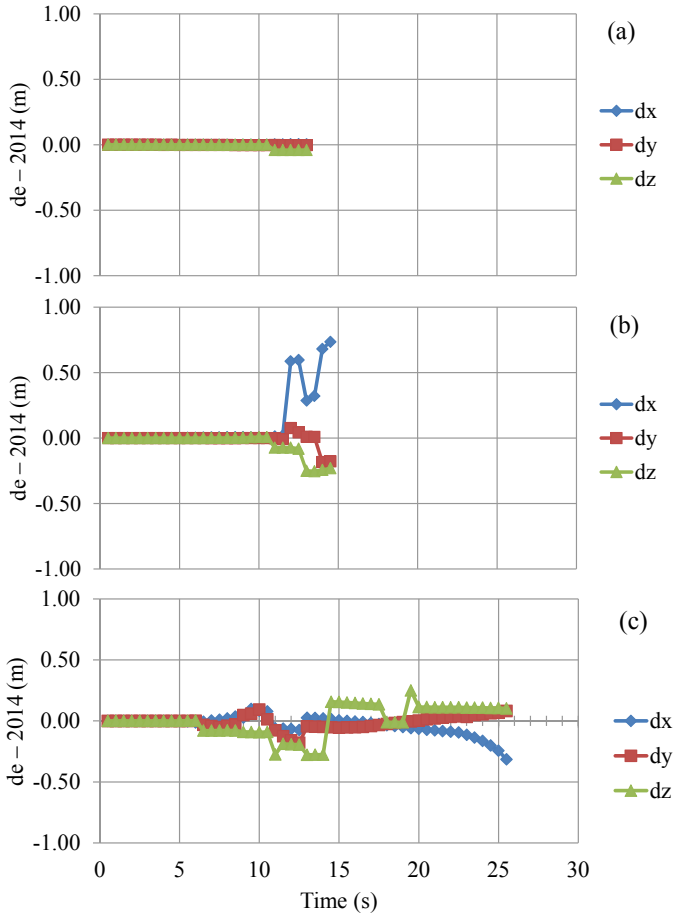


Figure 4.6: Fish position deviation between ELAM-de and ELAM-2014 frameworks over time, separated by dimension. The same behavior model was used. Subfigures (a), (b), (c) are fish #1, #2, #3, respectively.

(second-order chosen here) for this task. These two approaches lead to coordinate, velocity, and angle differences in the order of 1E-6 per time step. Depending on the way of the fish through the flow field, the differences are either never noticed (fish #1), or they add up and, through indirect influence, lead to the differences observed.

For example, in the case of fish #2, at time step 23 (simulation time t = 11.5 s), the value for velocity attraction is slightly different at sensory points SP(2) and SP(3) between the frameworks. SP(2) is in front of the fish and SP(3) is behind of the fish. Because the value order is reversed between the two frameworks (Table 4.3), a different sensory point is chosen and a

different algorithm is selected for computing the swim angle, leading to a strongly differing result for this time step.

Table 4.3: Values of dependent variables at time t = 11.5 s for fish #2.

	ELAM-2014	ELAM-de framework
Value SP2	0.6418012	0.637851773570203
Value SP3	0.6399524	0.638814310213844
Chosen SP	3	2
Swim angle	176.3296	-12.1202105188246

SP, sensory point.

The tiny deviations and their few large effects are only relevant for direct numerical comparisons as performed here. For the following work they are not relevant, as the same interpolation scheme is always used in a consistent manner. The results of this test case show that the ELAM-de framework works in the OpenFOAM environment as expected, has no remaining bugs, and can be used to run a new behavior model, as developed in the following chapter.

Chapter 5

New behavior model for trout in a flume

In this chapter, a new behavior model for upstream migration of trout in a flume is developed and applied using behavior data, CFD data, and the software framework from chapters 2 – 4. In sum, all parts form the new IBM called ELAM-de.

Several steps are scheduled for this task and described in the four sections of this chapter. First, test methods and a metric for model evaluation are designed. Second, the new behavior model is developed from fundamental considerations and literature information. Third, the implemented model is run and tested repeatedly. Fourth, results and structure of the final model are analyzed and discussed.

5.1 Test methods

5.1.1 Procedure

The behavior model is developed and tested following the three elements of pattern-oriented modeling suggested by Grimm and Railsback (2012): Model structure, model selection, model calibration. Model validation and software verification are addressed subsequently.

- The model structure is developed by using literature information and the trial-and-error method. As the process is highly iterative, only the result is described (section 5.2). Parameter values are estimated. Quality is ensured by means of hydraulic data and behavior patterns

from the flow field with slot (setup *1e-A*) from the EHF (chapters 2 and 3). These structure tests are limited to the four spatial patterns P(1–4) for speed and simplicity.

- Model selection is performed subsequently. Different submodels are contrasted to qualitatively test alternative movement hypotheses and to ensure that the model is as lean as possible (section 5.3.1). In addition, sensitivity to initial conditions is examined (section 5.3.2).
- Model calibration adds the temporal and success patterns, P(5–6), to the testing scheme (section 5.3.3). Thus, more information is incorporated into the model. Still, the structure remains unchanged. Change is only made to the parameter values. $n = 15$ parameters are calibrated in a study using Latin hypercube sampling ($n = 500$) in the flow field with slot. The quality metric used before is replaced by more detailed per-pattern limits as it is too coarse for quantitative calibration. This approach requires more analysis, but is also more precise. The result are three parameter sets for the flow field with slot.
- Finally, the three best parameter sets are validated against P(1–6) from the flow field without slot (setup *2c-A*), which is more homogeneous. This step completes model development (section 5.3.4).
- Software verification is performed continuously following every model change to ensure results are based solely on the intended behavior rules and not on e.g. faulty vector arithmetics.

The test methods are described in the remainder of this section.

5.1.2 Metric for model quality

A good comparative metric needs to balance meaning versus comprehensibility. To evaluate the behavior model structure, detailed information is less important than quick comparison. With respect to this a metric, or currency, is defined. It is termed *overall pattern deviation* (OPD), and summarizes percentage value differences of the four spatial patterns, P(1–4). Both laboratory and simulation percentage values are obtained using the pattern definitions developed in chapter 2. OPD is computed as

$$\mathrm{OPD} = \frac{1}{N_{val}} \sum_{i=1}^{N_{val}} |P_{i,sim} - P_{i,lab}| \tag{5.1}$$

where $N_{val} = 10$ is the total number of percentage values and i is the pattern/percentage value iterator. P(1–3) contain three values each, e.g. for the right, middle, and left zone. P(4) is represented by a scalar. OPD range is from 0–100 percentage points, where lower values indicate better agreement. An arbitrary limit of OPD $\leq$ 10 percentage points is defined as acceptable deviation.

The percentage-based definition enables direct combination of all pattern values. However, the averaged values are not quite equivalent, because percentage values are limited in range by definition. For example, a reference value of 40 % limits the difference magnitude to 60 % (|100 % – 40 %|). For a reference value of 90 %, however, the difference magnitude can reach up to 90 % (|0 % – 90 %|). Thus, the maximum difference magnitude of any compared value depends on the (arbitrary) reference value, which can render the equally weighted mean forming OPD unequal. I decided to accept this issue to gain the advantage of simple computation and comparison. This is justified as the OPD is only used for initial development and to find structural agreement, not for parameter calibration.

5.1.3 Latin hypercube sampling

For automatic calibration, parameters are varied. For a total of $p = 15$ parameters, assuming just $m = 3$ states per parameter, $m^p = 14\,348\,907$ trials would be required for full factorial analysis of the parameter space. Sampling is a way to drastically reduce this effort. A method for defining the samples is *Latin hypercube sampling* (LHS, McKay et al. 1979). It is widely used and generally recommended for initial model evaluation and to decide if more sophisticated methods should be used for optimization (Thiele et al. 2014).

LHS splits the parameter space into n^p equally probable bins (empty parameter sets). Just n bins are filled with p parameter values each, forming n parameter sets (samples). Per parameter, every state is represented exactly once by choosing a random value within the assigned bin range. The majority of bins remains empty. The result for a $p = 2$ parameter case of $n = 8$ samples can be imagined as a chess field populated with n rooks (♖), which each symbolize a p parameter set and are distributed in a manner that they cannot capture each other. In sum, the full parameter space of each parameter is covered.

5.1.4 Turn detection for simulation data

Postprocessing of the model track results is performed using the MATLAB scripts developed in section 2.2.2. Only the turn detection method developed needs some modification. The laboratory results are already filtered by the human observers for noticeable events, whereas the model results are unfiltered. This means that fish holding can manifest in inclined or flickering position graphs, which is not easily detected by the approach used for laboratory results.

To overcome this issue, a moving average function was implemented (Figure 5.1). It uses a filter window width $\Delta_{avg,window} = 5$ and detects turn points if the slope of the average function changes. This removes tiny holding phases with distances of $\Delta x < 0.01$ m. However, sharp edges are smoothed by this function. To prevent this undesirable behavior, the detected index gets replaced by the index of the minimum respective maximum value of the original function within $\Delta_{avg,window}$. Start area and minimum Δx filtering are performed unchanged.

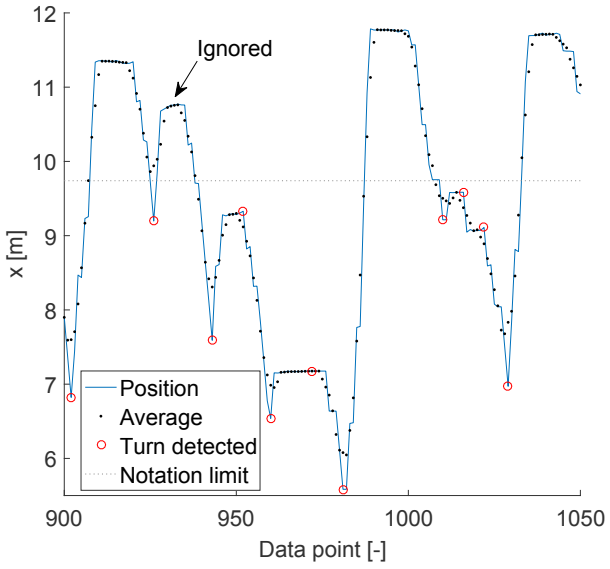


Figure 5.1: Automatic turn point detection through a moving average function on an example track. Above the notation limit at $x = 9.74$ m, there is no observation data available except for visits to the start area. Turns in this area are ignored.

5.1.5 Software verification

Software verification is part of the modeling cycle (Grimm and Railsback 2005, p. 314) because an overlooked error in the code can obfuscate all following development and calibration efforts. Initial agreement of the model framework software code developed in this work to the well-tested code of Goodwin et al. (2014) is described in section 4.3. Ongoing analyses were performed following relevant changes to the framework code, like variable handling, point interpolation, or out-of-bounds checks. A change was only accepted if the new results were identical to the previous results. In cases where this was not possible, e.g. development of new behavior submodels, comprehensive testing against laboratory observations was undertaken. Visual reality checks were performed by observing the resulting time-lapsed movement of fish in Tecplot software.

To test if the model program was independent of the hardware running it and the compiler and linked libraries used for its creation, it was executed on two high performance computers. No hardware-caused differences were found. However, linking of OpenFOAM libraries compiled with a different compiler version produced tiny differences in results, which kept adding up and led to fish displacement of some decimeters after 900 time steps. Hence, when exact repeatability between two cases is required, it is important to not only use the same OpenFOAM version, but also compiler version, in the best case identical libraries.

5.2 Model description

The new behavior model is described using the ODD protocol (Overview, Design concepts, Details; Grimm et al. 2006, 2010). ODD aims to standardize the published descriptions of IBMs. Hence, I follow the predefined headings and information order which gives a overview of the model in the beginning and mathematical details organized in submodels in the end of this section.

5.2.1 Purpose

The goal of model development described here is to reproduce movement patterns of upstream migrating fish observed in the EHF in a way that can be adapted to fishway entrance applications.

5.2.2 Entities, state variables, and scales

The individuals in the model represent fish staying in a 3D laboratory flume filled with flowing water. The fish can be distinguished by a number of *state variables* (Table 5.1) which are modified through distinct *behaviors*. The underlying state variables are *fatigue*, F, and *motivation*, M. The mathematical definitions and rationale are given below in sections 5.2.4 and 5.2.7.

Table 5.1: State variables defining individual fish.

Symbol	Units	Description
ID	-	Unique identification number
$\vec{s}$	m	Position of the center of gravity in 3D space
$\vec{U}_m$	m/s	3D flow velocity vector at fish center (interpolated from UMean, Table 5.2)
M	-	Motivation to swim upstream, takes values in $[0, 1]$
F	-	Fatigue, takes values in $[0, 1]$
U_s	m/s	1D fish swim speed (resultant), relative to flow
γ_s	°	Horizontal swim angle, off previous swim vector, values in $[-180°, 180°]$
β_s	°	Vertical swim angle off CFD horizontal plane, values in $[-90°, 90°]$
$\vec{s}_{avg}$	m	3D vector storing acclimatized position
$t_{sameSpot}$	s	Time cumulatively spent within radius $r_{sameSpot}$ around position $\vec{s}_{avg}$
SP(1–6)	m	Positions of the 6 sensory points around the fish

The laboratory flume overall domain size is (length × width × height) 16.78 m × 2.50 m × 1.35 m, including areas behind the screen which are not accessible to the fish (Figure 3.2). Water depth is $h = 0.60$ m. The domain is divided into approx. 850 000 mostly cubic cells, a typical cell edge length being $\Delta(x/y/z) = 0.05$ m. Each cell[1] holds information on the state variables listed in Table 5.2. In the current formulation, they are taken constant over time from the CFD model. More details about the CFD model domain and variables are documented in section 3.3.

Each time step is worth $\Delta t = 0.5$ s of real time. The default total model run time is $N_{\Delta t} = 7200$ time steps, which equals a total real time of $t_{total} = 3600$ s, or 60 min. This equals the maximum laboratory trial duration (section 2.2).

[1] More precisely: OpenFOAM stores most field information on cell faces, but allows for free interpolation in space.

Table 5.2: State variables defining cells of the CFD model.

Variable	Units	Description
Cell label	–	Unique identification number
alpha	–	Fraction of cell volume filled with water
UMean	m/s	3D flow vector, averaged to steady state
acclMag	m/s^2	Advective acceleration magnitude
TKE	m^2/s^2	Turbulence kinetic energy

5.2.3 Process overview

The model is based on *behaviors* the fish can exhibit while migrating. Behaviors are processes that modify the state variables. The three main behaviors are termed migrating, holding, and drifting. An additional vertical behavior is executed regardless of the main behavior chosen. The behaviors are explained in detail below.

The state variables are stored in arrays which preserve all time steps for post-process reconstruction. The general schedule of updating all variables, which takes place every time step, is visualized in Figure 5.2 and summarized following the program structure in the following pseudo-code:

```
for all fish{
    getSensoryPoints;
    interpolateEnvironmentToSensoryPoints;
    getNewAngleAndVelocity(
        chooseBehavior;
        if (B==1) migrating;
        if (B==2) holding;
        if (B==3) drifting;
        verticalSwimming;
    );
    updateFishLocation(
        if (locationOutsideMesh)
        then resetPosition;
    );
}
```

Updating the state variables to the current time step happens instantly after its computation. Still, the process is described as synchronous updating, as the input is always read from the preceding time step, which is not changed anymore after it has finished. There is no interaction among individuals present in the model, but if were there, their order could be ignored through this approach.

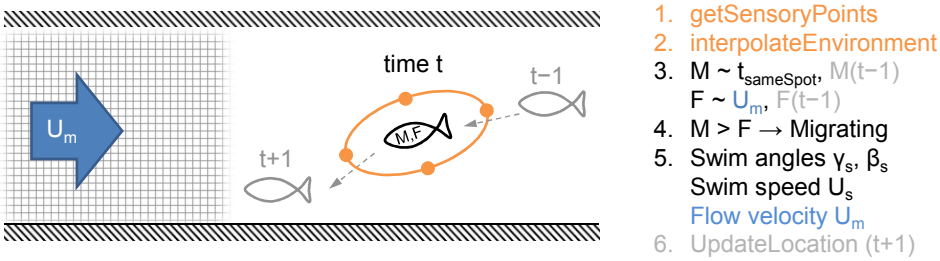


Figure 5.2: Schematic example of the model process during a typical time step.

The current time step holds the initial position and the new movement angle and speed influenced by the previous time step. The final position after movement is assigned to the next time step, making it the only variable in the model which is updated in advance.

5.2.4 Design concepts

Basic principles

The model structure was developed from scratch. A basic idea was to balance driving and retaining traits. For modeling, it is useful to split them into two simple and very general categories: external stimuli and internal states (Lucas and Baras 2001, p. 16). According to the model purpose of simulating behavior in a fishway, basic motivation for migration is presumed. Here, the general concept of model design is described. Full mathematical descriptions of all submodels are given in section 5.2.7.

External stimuli — According to Liao (2007), "for the majority of fishes, the two most important sensory modalities for swimming are vision and the lateral line sense". Vision is an unreliable stimulus, as it strongly depends on water quality and light sources. In contrast, flow information obtained from the lateral line system is available under all environment conditions. Hence, it seems likely that migratory behavior is mostly shaped by flow field perception. Therefore, vision can be omitted as an orientation stimulus in the model.

The flow field can can be represented through many different hydraulic variables in CFD models, e.g. 3D velocity, acceleration, or TKE.

Acceleration was successfully used by Goodwin et al. (2014) for modeling downstream migration of salmon smolts close to dams. It has the conceptual advantage that it can be felt under all circumstances. In contrast, relative

velocity can only be experienced by a swimming animal if there is a visible reference point, e.g. floating particles or another animal. Absolute velocity can only be estimated in relation to a boundary visible, i.e. in a light environment and in a short distance.

Gao et al. (2016) found the TKE to be the best stimulus for upstream movement in a fishway.

Both stimuli were used only for horizontal orientation. For vertical swimming, Goodwin et al. (2014) used acclimatization to pressure, simplified as depth coordinate. The new behavior model needs to provide the opportunity to test hydraulic stimuli for their effect on upstream migration.

Internal states — The balance of driving and retaining traits within the model fish is described by the variables motivation, M, and fatigue, F. They resemble the classic model components need and cost, which constantly need to be balanced (Willis 2011). The classic risk component (e.g., predation) is not modeled. Details and rationale are given in section 5.2.7.

Time perception of the model fish depends on the time step width. Goodwin et al. (2014) suggested the use of sub-time-steps to increase update frequency of internal states for faster adaptation in high-acceleration zones. Sub-time-steps were not tested, but could be a way to reduce computational cost. For the present model, the time step is quartered compared to Goodwin et al. (2014) to reflect the increased need for reaction in small-scale geometries. Transport of information through time is solved efficiently by using memory coefficients to model a moving average of a given parameter.

Position memory can be modeled in a similar way. There is strong evidence that different fish species are able to memorize locations and use landmarks for orientation in controlled experiments (Odling-Smee and Braithwaite 2003). In the limited extend of a flume with clear water it is plausible that fish have a good sense of orientation and remember the locations they have been in. This is used in the model to increase motivation if the position keeps unchanged for too long representing the presumed basic motivation for migration.

The relationship between swim speed and fatigue is not well understood, partly because of the huge inter- and intraspecific differences. Beamish (1978) defined a widely used model, which is useful for general understanding because of its clarity. Swim speed is classified as (BL, body length):

- *Sustained* – Only red muscles active. No exhaustion. Up to 2 BL/s.

- *Prolonged* – Combination of red and white muscles active. Exhaustion after 20 s to 200 min, depending on the swim speed, which is 2 BL/s to 5 BL/s.
- *Burst* – White muscles active. Exhaustion after 1 s to 20 s, depending on the swim speed, which is 5 BL/s to 20 BL/s.

Swim value estimates taken from (DWA 2014).

The actual behavior rules leading to fish shoaling and schooling are still poorly understood, even though different proposals have been made and main features of schools were successfully reproduced in the past decades (Katz et al. 2011; Herbert-Read et al. 2011; Lopez et al. 2012). Because no shoaling pattern was identified for the trout observed (2.2.2), and the underlying model framework as well as pre- and post-processing would require considerable adaptation, the effort was saved for future applications to other species.

Emergence

Spatial and temporal patterns of trout positions observed in the laboratory flume are expected to emerge from the combination of different behavior rules. This includes favoring of wall proximity, turn positions, and the ratio of trout moving successfully up to the finish line in the flume. The location and frequency of behavior changes (events) is expected to vary with changing hydraulic conditions and internal state parameters. Vertical behavior depends on acclimatized pressure and is not expected to vary strongly because of the shallow water.

Adaptation

The fish adapts to changes in the environment (flow field) through its *fatigue* variable, and to waiting time through its increasing *motivation* variable. The latter one is implicitly related to fitness-seeking, as upstream migration provides new opportunities to support the fish's fitness. However, it represents a pushing factor, in contrast to a pulling factor. Only the general upstream direction is specified.

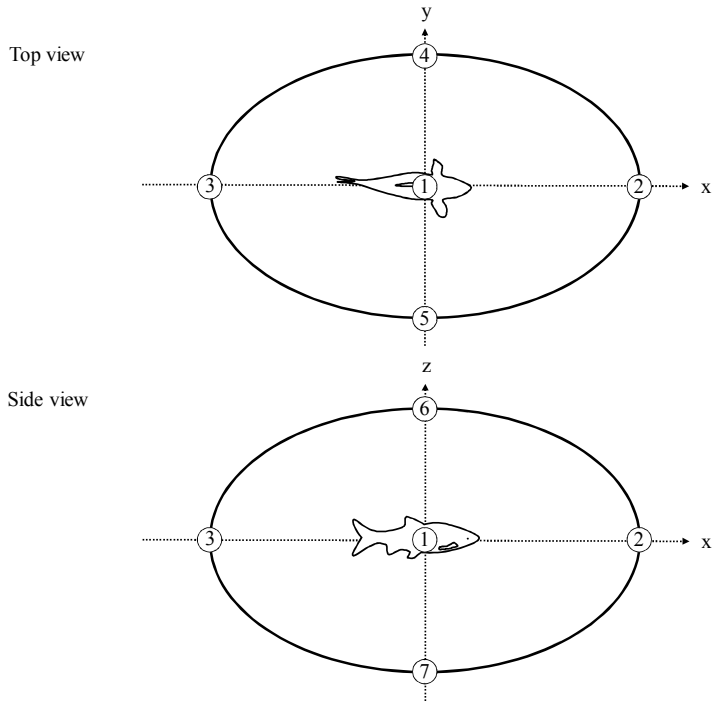


Figure 5.3: Fish sensory ovoid schematic showing the local coordinate system and all seven sensory points. The exact shape of the ovoid is variable.

Sensing

Individuals in the model perceive the mean velocity as a three-component vector, and acceleration magnitude and their depth (vertical coordinate) as a scalar. The values are interpolated from the discrete numerical mesh to the fish center and six sensory points surrounding it, representing the lateral line system information available to the individual. This concept is adapted from Goodwin et al. (2014) and termed the *sensory ovoid* (Figure 5.3).

Model fish are not aware of their companions. Even if simulated together, they behave independently.

Stochasticity

Stochasticity is introduced to the model at multiple points through the sub-models mathematically described in 5.2.7. I use it to recreate unpredictable behavior variation which is not represented mechanistically, because I assume that the details of these mechanisms either do not matter for my questions,

or are simply unknown. This subsection lists all occurrences of stochasticity in the model for an overview.

The pseudo-random number generator implemented uses a fixed initial seed number in [1,1000] to allow reproduction of results.

While migrating, the horizontal swim angle is modified randomly to avoid path repetition if starting from the same position.

After being trapped for too long in the same place despite *migration* behavior is active, the fish selects new horizontal and vertical swim angles randomly to escape.

The drifting direction is determined by the flow (see "drifting" submodel description). The angle is determined randomly.

Observation

Primary model outputs are the position and swim vectors of individual fish. From these data, pattern metrics are computed during postprocessing using the methods developed in section 2.2.2 for the laboratory patterns. They include the time fraction spent in different flume zones and the time of the first crossing of flume control lines.

5.2.5 Initialization

The flow field values are read from a completed CFD model simulation (section 3.3). Initially, only the values for the fish center are interpolated from the discrete mesh. Results are steady-state, i.e. there is no change in the flow field over time. This is a good approximation for steady flow in a flume. Turbulence levels can still be obtained from TKE output of the CFD model.

Individual positions are initially set within the x span of the start area at evenly-spaced y-coordinates spanning the flume width, close to the bottom according to observations ($z = 7$ cm).

Initial horizontal swim angle is set to $\gamma_s = 180°$ towards the CFD mesh x direction, i.e. facing upstream. Initial vertical swim angle is $\beta_s = 0°$. The sensory ovoid point positions are not yet computed, as they are needed for the first time in the following time step. ID values are read and remain constant during the simulation. The state variable for motivation is set to the arbitrary value of $M_{ini} = 0.25$. All remaining state variables are initialized to zero.

Fish are split in two groups representing “active/strong” and “passive/weak” fish. This is achieved by different motivation and fatigue parameters, k_M and k_F, per group. The parameters are explained in section 5.2.7.

5.2.6 Input data

The model does not use data from external sources to represent time-varying processes (this information is required fore a complete ODD protocol).

5.2.7 Submodels

All processes in the behavior model can be understood as distinct submodels representing certain traits or behaviors of the individuals. The submodels are mathematically described in this section; all (fixed) input parameters chosen for trout (without calibration) are listed in Table 5.3.

Table 5.3: Behavior model parameters and (uncalibrated) values. Parameters chosen for subsequent calibration are numbered in the first column. First block of rows is constant for all individuals, second block is split for “strong” and “weak” groups.

No.	Parameter name	Symbol	Value	Units	#
-	Sensory ovoid size, longitudinal	O_X	0.50	BL	1
-	Sensory ovoid size, lateral	O_Y	0.25	BL	2
-	Sensory ovoid size, vertical	O_Z	0.33	BL	3
1	Radius limit from initial position	$r_{sameSpot}$	3.0	BL	13
2	Ground speed	U_g	2.5	$\frac{\text{BL}}{\text{s}}$	52
-	Drift acceleration threshold factor	$k_{a,D}$	1.01	-	41
-	Migrate acceleration threshold factor	$k_{a,M}$	1.01	-	40
-	Pressure/elevation threshold	k_p	0.07	m	24
3	Trap threshold value	t_{trap}	50	s	57
4	Migrating angle acceleration influence	$\gamma_{Migr,a}$	20	°	45
5	Migrating angle random influence	$\gamma_{Migr,r}$	20	°	44
-	Vertical angle correction	β_p	7	°	48
6	Holding tolerance value	Δ_H	0.075	-	56
7	Motivation memory coefficient	m_M	0.98	-	5
8	Fatigue memory coefficient (decr.)	$m_{F,d}$	0.99	-	6
9	Fatigue memory coefficient (incr.)	$m_{F,i}$	0.30	-	7
10	Spot memory coefficient	m_s	0.92	-	8
-	Pressure memory coefficient	m_p	1.0	-	32
12	Motivation coefficient (“strong”)	k_M	5.0	s	-
13	Motivation coefficient (“weak”)	k_M	30.0	s	-
14	Fatigue coefficient (“strong”)	k_F	18.0	$\frac{\text{BL}}{\text{s}}$	-
15	Fatigue coefficient (“weak”)	k_F	8.0	$\frac{\text{BL}}{\text{s}}$	-

is input order in the file *agentBehaviorCoefficients.inp*; BL is body length.

Sensory ovoid

The *sensory ovoid* surrounds the fish and consists of six points (plus one at the center, see Figure 5.3). The outer points are placed along the three principal axis of the fish in each direction. A local coordinate system is defined by swimming direction and opposite gravity direction, with the origin being in the fish's center of mass. The distances from the center are O_X, O_Y, O_Z [BL], where the indices denote local axes. The values chosen here for trout are listed in Table 5.3. They reflect the typical longish form of a fish. The state variables of the CFD model (Table 5.2) are interpolated to all sensory points at the beginning of every time step using OpenFOAM tools.

Rationale: The interface between the conceptual elements "environment" and "fish" is the fish's skin. As the fish is represented as a point in the model, its perception range would be under-estimated without the sensory ovoid. The true range is unknown, but likely to vary with a lot of factors. The lateral line system enables most fishes to sense weak water motions and pressure gradients (Bleckmann and Zelick 2009) and to "feel" the surrounding flow field. Differing estimations for the perception range are present in the literature: It has been modeled as randomly fluctuating in Gao et al. (2016) and additionally dependent on the flow acceleration in Goodwin et al. (2014), which both was not necessary here. For comparison: Goodwin et al. (2006) used an estimate of $O_{X,Y,Z} = 1.25 - 1.875$ m (14–21 BL) and Kerr et al. (2016) used an rectangle of $O_X = 2.0$ BL $\times$ $O_Y = 1.0$ BL (BL $= 0.11$–0.29 m). The sensory ovoid is an important aspect of ELAM models, as it connects Eulerian flow information and Langrangian fish movement.

chooseBehavior

Two state variables are used to describe the internal state of the model fish: motivation M and fatigue F. Depending on their relation, one of three behaviors is chosen through the submodel *chooseBehavior*. It sets the behavior variable B to either

$$B = \begin{cases} 1 & \text{if} \quad & M > F + \Delta_H & \text{(Migrating)} \\ 2 & \text{if} \quad F - \Delta_H \leq & M \leq F + \Delta_H & \text{(Holding)} \\ 3 & \text{if} \quad F - \Delta_H > & M & \text{(Drifting)} \end{cases}$$

using holding tolerance value, Δ_H (all parameter values in Table 5.3). For M and F, the time-averaged values M_{avg} and F_{avg} as described below are used. The fish acclimatizes to its current position, $\vec{s}$, over time. A standard memory function (exponential moving average) is used to link preceding time steps to the present time step with decaying impact. The acclimatized position, $\vec{s}_{avg}$, is computed as

$$\vec{s}^{n}_{avg} = (1 - m_s)s^n + m_s\vec{s}^{n-1}_{avg} \tag{5.2}$$

where m_s [−] is the constant position memory coefficient (Table 5.3) and n superscript denotes the current time step.

While the real fish position is within a limited radius around its acclimatized position, a state variable, $t_{sameSpot}$, is added up (Table 5.1). The condition is

$$\vec{s} - \vec{s}_{avg} < r_{sameSpot} \tag{5.3}$$

After reaching or exceeding $r_{sameSpot}$ or after triggering the anti-stuck reaction (see submodel *migrating*), the state variable is reset to $t_{sameSpot} = 0.0$.

After being trapped in a place for $t_{sameSpot} > t_{trap}$, despite *migration* behavior is active, the fish selects its new swim angle randomly in [0,360[. To account for cases in which fish are vertically blocked, the vertical swim angle is forced to $\beta_s = 0$ in half of the cases, also controlled by the random number. This submodel prevents fish from getting stuck in any situation. For example, trapping in front of a screen occurs because fish are generally moving in flow direction, but cannot cross the screen.

The idea behind the behavior submodel is to balance the two most basic internal traits of an individual and to integrate position memory. The definition of motivation and fatigue is as follows.

Motivation

Motivation increases or decreases depending on whether the fish holds or moves. This is controlled by the cumulated time, $t_{sameSpot}$ [s]. Motivation is calculated as:

$$M = \frac{1}{k_M} t_{sameSpot} \ [-] \tag{5.4}$$

using constant motivation coefficient k_M [s]. Motivation is used in the model in time-averaged form, M_{avg}. It is transported through time using a moving average function:

$$M_{avg}^n = (1 - m_M)M^n + m_M M_{avg}^{n-1} \tag{5.5}$$

where m_M [−] is the motivation memory coefficient and n superscript denotes the current time step.

M takes values in $[0, 1]$. The minimum values means "no need to move to new upstream locations", e.g. just after the fish has entered new territory, and the maximum value means the "most urgent need to move to new upstream location", e.g. after spatial stagnation for a while. The coefficient, k_M, represents the time needed to reach maximum motivation.

Rationale: Motivation measures the strength of the fish's short-term impetus to migrate upstream and generally fuels upstream movement in the model. The long-term impetus for migration results from general fitness advantages of changing location, such as reaching spawning habitats, greater food availability, and new mating partners. However, the laboratory test setup does not allow for conclusions on the long-term impetus. Instead, I use the short-term impetus, which should have the same general tendency, but could be weaker in lab results.

Fatigue

Fatigue is calculated in dependency of the fish's swim speed relative to the flow, U_s [BL/s], from the previous time step:

$$F^n = \frac{1}{k_F} U_s^{n-1} \quad [-] \tag{5.6}$$

$$F_{avg}^n = (1 - m_F)F^n + m_F F_{avg}^{n-1} \tag{5.7}$$

$$m_F = \begin{cases} m_{F,d} & \text{if} \quad F^n \leq F_{avg}^{n-1} \\ m_{F,i} & \text{if} \quad F^n > F_{avg}^{n-1} \end{cases} \tag{5.8}$$

where n superscript denotes the current time step and m_F the fatigue memory coefficient. F is rendered dimensionless by the fatigue coefficient k_F [BL/s], which marks the maximum velocity for full fatigue. U_s units are

Figure 5.4: Brown trout in the flume showing typical spotting along the body.

converted as follows, where BL is body length and $\mathrm{BL}_{trout} = 0.27$ m (Figure 5.4, calculation in subsection 2.2.1).

$$U_s \left[\frac{\mathrm{BL}}{\mathrm{s}}\right] = U_s \left[\frac{\mathrm{m}}{\mathrm{s}}\right] / \mathrm{BL}_{trout} \left[\frac{\mathrm{m}}{\mathrm{BL}}\right]$$

F takes values in $[0, 1]$. The minimum and maximum fatigue values can be described in words as "hold this speed for unlimited time, feeling perfectly well" resp. "full exhaustion, try hard to avoid this state". Exhaustion was not observed for brown trout in the EHF study as the water velocity was moderate for them ($U_m \approx 0.67$ m/s ≈ 2.5 BL/s).

Rationale: Fatigue is an inverse measure of the capability to perform work (e.g., swimming in prolonged or burst mode) and generally slows or stops upstream migration in the model. Fatigue covers traits such as metabolic cost, body fat reserves, oxygen concentration in the white muscles, individual strength, and injuries. It can be related to flow velocity via speed-fatigue diagrams (examples for six species in Castro-Santos 2005) or the swim speed classes of Beamish (1978). Here, a simpler approach is chosen which does not take into account temporal aspects. As suggested by Liao (2007), location preference in combination with information about the flow (e.g. U_m), can serve as a first approximation for metabolic cost in the absence of direct information about energy consumption, e.g. tail-beat frequency or oxygen consumption.

From the motivation and fatigue values, three behavior submodels for

migrating, holding, and drifting are chosen. An additional submodel steers vertical behavior. The four behavior submodels all produce a swim speed magnitude, U_s, and horizontal and vertical swim angle, γ_s and β_s, which are transformed into the volitional swim vector, $\vec{U}_s$, afterwards. They are defined as follows.

Migrating

In the model, migrating is swimming against the flow vector, faster than the flow. Swim speed magnitude, U_s, is fixed relative to the flow at

$$U_s = U_m + U_g \quad [\text{BL/s}]$$
$$\beta_s = -\beta_m$$

where U_m is mean flow velocity magnitude, U_g is fixed fish groundspeed magnitude, β_s is the fish vertical swim angle, and β_m is the mean flow vertical angle.

The horizontal swim angle, γ_s, is adjusted by $\gamma_{Migr,a}$. The adjustment takes place to the side with the greater acceleration magnitude. Definitions and ranges of the state variables are given in Table 5.1. Values chosen for model parameters are listed in Table 5.3.

If one of the lateral sensory ovoid points is placed outside the domain, the swim angle is set adverse to the flow vector to follow the boundary contour.

Finally added the swim angle is modified randomly between $\pm\gamma_{Migr,r}$ every time step to avoid path repetition if starting from the same position.

Rationale: In the case of moving exactly against the flow, this behavior elicits constant movement over ground, which is an estimate for optimal energy use: Castro-Santos (2005) found values between $U_g = 1$ BL/s for American shad to $U_g = 2.56$ BL/s for striped bass as predicted optimal groundspeed in prolonged swimming mode, Fish (2010) lists results between $U_g = 0.3 - 5.8$ BL/s as swim speed (not groundspeed) optimum for a variety of fish. The U_g value chosen here for trout (Table 5.3) is within these literature ranges.

The wall avoidance component of migrating behavior reduces the number of unrealistic wall collisions.

Instead of using a correlated random walk, which is a common choice for forward movement (e.g. Goodwin et al. 2014), I decided to separate stimulus

and random component. This enables switching them on and off to study their respective influence.

Holding

Holding means staying in position by setting the swim speed and angle adverse to the mean flow vector's magnitude and angles:

$$\vec{U}_s = -\vec{U}_m$$
$$\gamma_s = \gamma_m - 180°$$
$$\beta_s = -\beta_m$$

where γ_m is the mean flow horizontal angle.

Holding is often observed close to obstacles and in tranquilized zones during upstream migration, probably to recreate from work (section 2.1.4).

Drifting

Drifting is passive movement with flow speed. Here, also retarded drifting (with little effort) is considered. The swim speed, U_s, is set to a random value, smaller than flow velocity, U_m, and the horizontal swim angle γ_s, which has no effect without swim speed, is set against flow direction:

$$0 \leq U_s \leq U_m$$
$$\gamma_s = -\gamma_m$$

The vertical swim angle is $\beta_s = 0$.

Attraction to higher acceleration is implemented for drifting as follows. If acceleration magnitude on one side exceeds acceleration on the other side by more than a factor of $k_{a,D}$, γ_s is adjusted randomly up to $\gamma_{D,max} = 90°$ towards the side with higher acceleration. In combination with flow transport and the swim speed magnitude an area of maximum dislocation is determined downstream of the original position (Figure 5.5).

This component allows to relocate downstream without or with little energy cost.

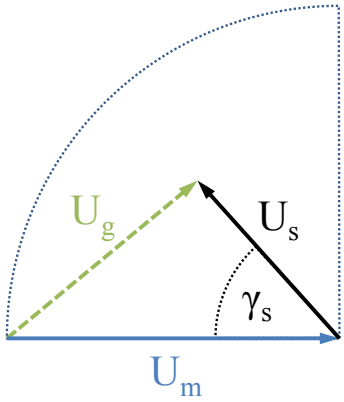


Figure 5.5: Area of maximum dislocation with fish ground speed magnitude, U_g, composition from the flow velocity magnitude, U_m, and the (random) swim speed magnitude for drift, U_s, in dependence of the (random) swim angle, $\gamma_s \leq 90°$.

Vertical swimming

In addition to the vertical behavior produced by the preceding submodels, the submodel from Goodwin et al. (2014) is used to account for acclimatization. Pressure p is represented by the z component of the fish's center position, assuming hydrostatic pressure distribution. The acclimatized pressure is transported by a memory model:

$$p_{avg}^{n} = (1 - m_p)p^n + m_p p_{avg}^{n-1} \tag{5.9}$$

where m_p is the pressure memory coefficient, and p_{avg} is the acclimatized pressure. If the current pressure/vertical position differs from the acclimatized pressure/position for more than threshold k_p, the vertical angle is set to β_p in the opposite direction.

$$\beta_s = \mp\beta_p \quad \texttt{if} \quad p^n \gtrless \bar{p}^{n-1} \pm k_p \tag{5.10}$$

This submodel ensures that sudden changes in pressure, which can be sensed by fish through their swim bladder, are mitigated immediately. It is kept simple as vertical movement plays a minor role for the observed behavior.

Sometimes, local flow disturbances (e.g., strong vertical flow) can lead to a fish's vertical angle being negative despite it is close to the flume bottom.

To prevent this unrealistic behavior, a minimum vertical angle of $\beta_{min} = 5°$ is enforced if the bottom sensory point (SP7) is outside the model domain.

From the result of either migrating, holding, or drifting, and vertical swimming, a volitional swim vector, $\vec{U}_s$, is finally computed.

updateFishLocation

updateFishLocation computes a new position from the volitional swim vector, $\vec{U}_s$, the flow vector at the fish center, $\vec{U}_m$, and the time step, Δt:

$$\vec{s}_i^{n+1} = \vec{s}_i^n + (\vec{U}_s + \vec{U}_m) \times \Delta t \tag{5.11}$$

Then, it checks whether the new position is inside the domain. If the fish has left through an exit (boundary type "patch" in the model), a counter is increased and the fish is removed. If it has left through a fixed boundary (type "wall"), the fish gets reset to its previous position (*resetPosition* submodel description below).

If the new fish position lies inside the model, but in the air phase, it is forced down in 10 cm increments until it is in the water phase again. An out-of-bounds check is performed additionally.

Rationale: Fish leaving the domain through screens or walls is an unrealistic behavior and needs to be precluded. In the best case, the fish would not choose such paths, but that would require full awareness of all boundaries, i.e., unlimited vision. In the present model, the fish is limited to the information at its sensory ovoid points modeling the lateral line system. This information is used to steer in a wall parallel direction (see above, "migrating"), but that is not sufficient to prevent collisions if the fish moves outside its previous sensory ovoid. Hence, numerical safety checks are necessary.

resetPosition

If one of the checks of *updateFishLocation* fails, the *resetPosition* submodel is called. It resets the fish to its previous position.

Rationale: The chosen approach has two advantages: It is stable, because the previous position is always valid, and it is computationally cheap, because just a check for exit boundary hits is required, not for all impenetrable boundaries (e.g., walls). Its disadvantage, the risk of producing identical re-

sults and getting trapped in an infinite loop due to identical environment input, is small, as there is stochasticity included in the movement model.

5.3 Test results

5.3.1 Model structure

The behavior model described in section 5.2 was implemented in computer code and run subsequently as "reference simulation". Alternative submodels representing e.g. different hydraulic stimuli or different movement hypotheses were tested for their influence on pattern agreement. Submodels were omitted to evaluate if they were necessary in the model. Selected differences to the absolute percentage values observed in the EHF are reported in Table 5.4.

In the reference simulation, very good agreement was achieved for the horizontal, vertical, and turn patterns, while only the no-turns pattern, P(4), had a higher deviation. The OPD metric was below the limit of OPD $\leq$ 10 percentage points (Table 5.4, ID 170). This accuracy is sufficient for testing model structure by different submodels, as the reproduction of multiple patterns indicates that the model captures a system's essential characteristics (Grimm and Railsback 2005, p. 47).

The vertical movement submodel was already used by Goodwin et al. (2014) with a memory coefficient $m_p = 0.9984$ (section 5.2.7). I obtained better results by turning memory off and fixing the acclimatized pressure using $m_p = 1.0$. Eliminating the submodel using threshold $k_p = \infty$ leads to large deviations in the vertical pattern results (Table 5.4, ID 166).

Horizontal fish behavior is classified into three categories reflected by submodels: migrating, holding, and drifting. Strictly speaking, holding is not needed for simulating movement, which is why e.g. the ELAM-2014 model can go without it. To evaluate if this behavior is needed for upstream movement simulation, it was turned off by setting $\Delta_H = 0.0$. The resulting deviation (Table 5.4, ID 171) in patterns P(3) and P(4) and largely increased stay frequencies close to the start area (not shown) support the chosen approach with holding.

Position acclimatization through time was turned off by setting $m_s = 0.0$

Table 5.4: Pattern values in the ethohydraulic flume (absolute, first row) and differences sim–lab (relative, subsequent rows) of selected ELAM-de simulations for setup *1e-A* (slot, long screen). IDs are in ascending order and grouped by text sections 5.3.1 and 5.3.2. Key words refer to the tests described in the text. For "horizontal" columns, L, M, R is left, middle, right; for "vertical", S, M, B is surface, middle, bottom; for "turns", U, M, D is upstream, middle, downstream third of the flume length. OPD is overall pattern deviation. SD is standard deviation.

ID	Key words	P(1) Horizontal			P(2) Vertical			P(3) Turns			P(4) No-turns	OPD
		L	M	R	S	M	B	U	M	D		
1e-A	Laboratory	35	8	57	1	1	99	26	28	47	60	–
170	Reference simulation	7	3	-10	-1	1	0	-1	7	-6	-27	6.3
166	no p limit	6	10	-16	3	27	-30	-6	13	-7	-10	12.7
171	no holding	-4	-2	5	-1	6	-5	-6	-14	20	-60	12.2
172	$m_s = 0$	12	11	-23	-1	4	-4	44	-4	-40	-43	18.5
178	high U_m attr.	22	9	-31	-1	3	-3	30	-6	-25	-52	18.1
179	acclM attr. off	6	17	-23	-1	4	-3	26	-5	-21	-43	14.9
180	wall avoid. off	9	12	-21	-1	5	-5	18	-1	-17	-60	14.8
192	low U_m attr.	52	0	-52	-1	2	-2	18	3	-21	-60	21.1
198	TKE const.	21	15	-36	-1	4	-4	45	-15	-30	-52	22.3
177	no random	11	26	-37	-1	6	-6	42	-7	-35	-60	23.1
181	y=+10cm	1	4	-5	-1	6	-5	3	10	-13	-27	7.3
182	mean (20 seeds)	1	3	-4	-1	3	-3	6	5	-11	-39	8.6
	SD (20 seeds)	6	2	6	0	2	2	9	5	6	10	1.7
185a	Normal Dist.	-7	2	5	-1	4	-4	13	8	-21	-52	11.6
185b	Inv. Norm. Dist.	-1	0	1	-1	3	-2	13	-7	-6	-43	7.7

and revealed that this process can not be eliminated from the current model (Table 5.4, ID 172).

Leaving out the migration orientation towards increasing acceleration (Table 5.4, ID 179) and replacing it with high velocity attraction (ID 178) both largely increased the OPD and deteriorated the agreement towards observations. The same went for low velocity attraction (ID 192).

TKE attraction was tested using ΔTKE between the ovoid center and the frontal sensory points SP(4,2,5) instead of acceleration attraction. The minimum Δ was chosen to ensure a constant level of turbulence (Goettel et al. 2015). The resulting OPD was high (Table 5.4, ID 198).

Omitting the wall avoidance submodel integrated in the migration behavior weakened the OPD as well (Table 5.4, ID 180), showing its utility.

5.3.2 Sensitivity

Model sensitivity is tested towards stochasticity, initial position, fish count, and time step width.

Stochasticity — Stochasticity is introduced into the model at the points summarized in section 5.2.4. The easiest test of its value for minimizing OPD would be to deactivate it by setting the random number generator to always return a fixed value, say 0.5. However, this would cause the anti-trap submodel to collapse, as it could not operate in case the first try would fail. Thus, the fixed value was hardcoded for the test at all points following a random number call, except before the anti-trap submodel. The results (Table 5.4, ID 177) were clearly worse compared to the reference case.

To identify the influence of the random seed on results, a small parameter study was performed. 20 random samples of the `seed` parameter were selected from its range [1,1000] by Latin hypercube sampling (section 5.1.3). The mean and standard deviations for single pattern deviation values and OPD are listed in Table 5.4, ID 182. Their means indicate a small overall influence. The strongest negative influence was on both μ and σ of the "no-turns" pattern. Results suggest that the model works well independent of the chosen random seed value, which is fixed to `seed` = 92 in the following.

Initial position — Uniform distribution was used as there is no data to support the choice of other distributions. To explore the consequences of this choice, a normal distribution (highest density in flume middle) and inversed

normal distribution were tested using $N = 12$ fish. Resulting OPD was low for the inversed normal distribution (Table 5.4, ID 185b), but above the arbitrary limit of 10 % for the normal distribution (Table 5.4, ID 185a). As it is the simplest and performed best, I will stick to uniform distribution.

No considerable effect on OPD was noticed when the initial positions of all fish were moved by $\Delta y = +10$ cm (Table 5.4, ID 181). Is was not necessary to test larger shifts, as $\Delta y = 20$ cm.

Fish count — The number of fish tested influences both the result and the computational cost. To check whether the current number of $N = 12$ is a good trade-off between these factors, N was varied in (6,12,18,24) with the parameter set of simulation ID 181. Fish were positioned in equal distance along the line $x = 11.01$ m. OPD was 14.9, 7.3, 10.9, and 8.2. As expected, $N = 6$ resulted in a large deviation. The others did not change the OPD substantially. Hence, $N = 12$ fish per simulation was kept for lowest computational cost.

Time step width — Despite update frequency effects, the model is mostly independent of the time step width chosen. Only the vertical behavior model uses an absolute limit in terms of length per time step instead of velocity.

Parameter sensitivity is tested using calibration.

5.3.3 Calibration

After showing that the new behavior model is structurally suitable for modeling the behavior of upstream swimming trout in a flume, parameter calibration is performed. Calibration counters effects of the simplifications inevitable to all models. It allows to give more precise predictions at the cost of being tailored more closely to the boundary conditions it has been calibrated to.

I took 15 parameter values from the previous, uncalibrated version of the model as a starting point and defined lower and upper bounds around them, depending on physical limits and my expectations (Table 5.5). The remaining 7 parameters were treated as constants. Another parameter, the pressure memory coefficient, had no effect in the original model and was omitted. I chose uniform probability distribution to exactly control the limits and because the statistical distribution for many parameters was unclear.

Generation of random LHS-distributed parameter samples (section 5.1.3) was performed using a MATLAB script. The output precision was limited to

Table 5.5: Latin hypercube sampling parameter limits used in the calibration study. See Table 5.3 for names and units.

No.	Parameter	Default	Min.	Max.
1	$r_{sameSpot}$	3.0	0.5	10
2	U_g	2.5	1.0	4.0
3	t_{trap}	50.0	30.0	70.0
4	$\gamma_{Migr,a}$	20.0	0.0	60.0
5	$\gamma_{Migr,r}$	20.0	0.0	60.0
6	Δ_H	0.075	0.050	0.100
7	m_M	0.980	0.950	1.000
8	$m_{F,d}$	0.990	0.900	1.000
9	$m_{F,i}$	0.30	0.00	0.60
10	m_s	0.92	0.50	1.00
11	M_{ini}	0.25	0.00	1.00
12	$k_{M,1}$	5.0	1.0	9.0
13	$k_{M,2}$	30.0	25.0	35.0
14	$k_{F,1}$	18.0	12.0	24.0
15	$k_{F,2}$	8.0	1.0	15.0

variable-dependent digits to easy handling. This can in some cases violate the LHS property of never having identical values for a parameter. However, as long as the bins are small enough and the parameter values do not show large gaps, no negative effects are expected. A sample size of $N = 500$ parameter sets was generated. The resulting 500 ELAM-de simulations were run on BAW's high performance computer. The runtime cap of 3 h was exceeded in two simulations and one simulation crashed. These three simulations were excluded from analysis. Figure 5.6a shows a sample result snapshot.

The minimum runtime duration was 212 s, the maximum duration was 10 503 s. On average, a single run took 2363±1709 s to complete, which includes unrealistic parameter combinations. A typical realistic run took 1139 s to simulate 12 fish at a maximum of 7200 time steps (ID e184). For comparison, ELAM-2014 needed 90 s to simulate 500 fish at a maximum of 40 000 time steps. The numbers do not mean the same, as the number of time steps varies per fish depending on how fast it reaches an exit, but a rough factor is that the old model is about 2930 times faster.

In addition to patterns P(1–4), patterns P(5–6) were used to form new hurdles for the model. They integrate time and success information into the model.

For this more detailed evaluation the OPD metric was of limited use

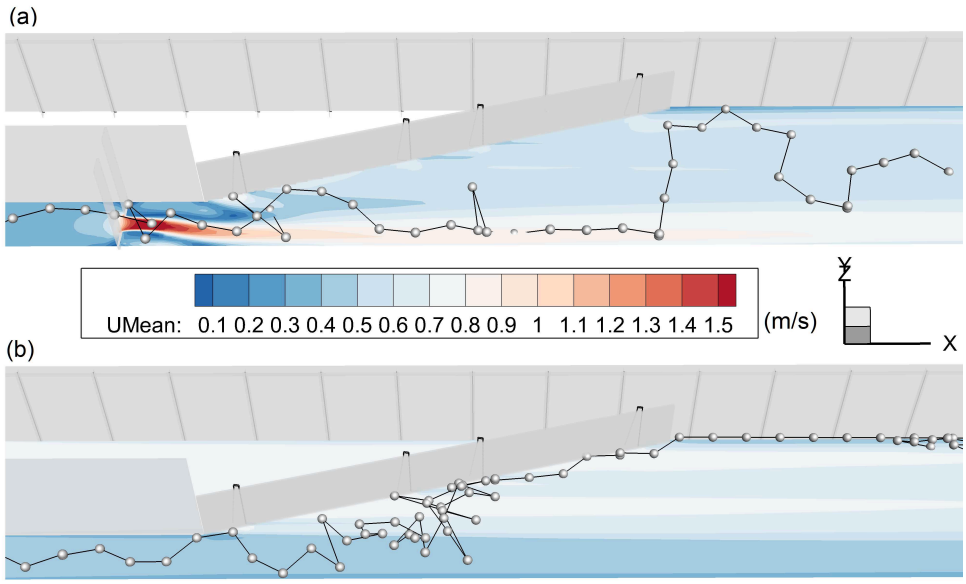


Figure 5.6: Exemplary model trout tracks. (a) Calibration run, flow field with slot, ID e184, track 7. (b) Validation run, flow field without slot, identical parameter values, track 12. Flow from left to right, fish move from right to left.

because, as a mean value, it is sensitive to extreme values. A uniform limit of e.g. 15 percentage points turned out to be too demanding for some values and too modest for others. Hence, local limits were defined per pattern. For comparison, the patterns consisting of three values, P(1–3), were converted into a single value using $\frac{1}{3}\sum_{i=1}^{3}|P_{i,sim}-P_{i,lab}|$, i being the value iterator. For the single value forming P(4–6), plain magnitude was used: $|P_{i,sim}-P_{i,lab}|$. The local limits were reduced until the arbitrary number of three simulations matching them remained (Table 5.6).

An example of a practical result is the calculated fatigue of simulation ID e184. All six failed fish had a maximum $F_{max} = 0.93$, which is close to full fatigue. The six finishers had $F_{max} = 0.53$ to $F_{max} = 0.72$.

The parameter sets of the best three simulations (appendix D.3) were applied for validation.

5.3.4 Validation

A good test for correct model function is to apply and validate it under altered input. The EHF setup *2c-A*, long screen without slot, is well-suited for this task as the flow field is changed but geometry is constant (section 3.4).

Table 5.6: Calibration limits and deviations (percentage points) of the three simulations meeting them. Deviation is simulation–laboratory, data set *1e-A*.

	Hori. P(1)	Verti. P(2)	Turns P(3)	No-turns P(4)	$\overline{\text{AC}}/\overline{\text{AD}}$ P(5)	N_A/N_D P(6)
Limit	10.0	5.0	15.0	20.0	20.0	20.0
e170	3.8	2.8	10.4	14.5	16.4	0.7
e184	9.6	0.7	14.8	15.6	16.7	16.0
e277	1.9	3.4	14.7	18.3	15.6	16.0

The three parameter sets found in calibration were applied in three new simulations. All six patterns quantified from data set *2c-A* were used.

Applying the tight limits used for calibration, agreement decreased from six to three (ID e184), two (ID e277), and one (ID e170) patterns matched, respectively (Table 5.7). For ID e184, the limit violation was small for patterns P(1) and P(6). The only major violation occurred in pattern P(5), $\overline{\text{AC}}/\overline{\text{AD}}$. It was identified as the best parameter set. Figure 5.6b shows a sample result snapshot.

Calculated fatigue of simulation ID e184 was lower than for the calibration flow field with jet. The six fastest finishers had $F_{max} = 0.27$ to $F_{max} = 0.36$. The six slower fish had a maximum $F_{max} = 0.71$ to $F_{max} = 0.80$.

Table 5.7: Validation limits and deviations (percentage points) of the three calibrated simulations. Deviation is simulation–laboratory, data set *2c-A*.

	Hori. P(1)	Verti. P(2)	Turns P(3)	No-turns P(4)	$\overline{\text{AC}}/\overline{\text{AD}}$ P(5)	N_A/N_D P(6)
Limit	10.0	5.0	15.0	20.0	20.0	20.0
e170	19.3	2.9	34.5	25.0	52.1	-29.2
e184	11.9	2.9	8.5	8.3	50.1	20.8
e277	13.2	2.9	22.5	0.0	51.5	20.8

5.4 Discussion

In this section, I discuss behavior model quality, analyze behavior model structure, and examine technical aspects of the underlying framework with respect to the objectives of this work (section 1.3).

5.4.1 Model quality

To evaluate fishway design efficiency in a meaningful way, ELAM-de must capture all essential characteristics of real fish and their environment. Patterns are recommended to test this property, as a variety of suitable patterns can capture even great complexity (Grimm and Railsback 2005, p. 320). Their strength is not important, as a combination of multiple weak patterns is more useful for evaluation than a single strong pattern, if they represent different features of the system (Wiegand et al. 2004).

After calibration on the EHF setup with slot, ELAM-de successfully reproduced all six observed behavior patterns of trout. This is a good result, as qualitative matching is the most important for model evaluation (Grimm and Railsback 2005, p. 321). Quantitative per-pattern deviation were 18.3 percentage points or less, which could likely be reduced by focusing on less parameters and by increasing calibration runs.

For validation on the EHF setup without slot, qualitative matching was also successful for five patterns. However, the validation test case was not ideally suited, because the observed trout spatial behavior did not change considerably with the flow field. The significant time behavior speed-up without slot, pattern P(5), was not reproduced. A possible explanation is that this delay is primarily caused by reaction to the slot geometry and not to hydraulic influence in ELAM-de. If true, this would require changes to the motivation and/or fatigue submodels. To test this, variable slots with constant velocity would need to be examined with real fish.

Deviations between the simulated flow field without horizontal screen bars and the laboratory flow field may contribute to behavior model deviations. However, the accuracy of the CFD model is higher than the observed spatial behavior data, which makes influence unlikely. In addition, behavior model parameter calibration further reduces this error source by adaptation to the simulated flow field (Goodwin et al. 2014). Thus, the error contribution is estimated to be negligible.

Defining application boundaries is important for using ELAM-de in management. The results of this work are applicable to trout of similar body length in the EHF under two hydraulic setups, with the restriction that timing or delays are not represented in a general manner. In ELAM-de, motivation for upstream migration was presumed. In the behavior data, mo-

tivation remained unclear, as only active movement could be ensured by the line A criterion. This discrepancy could explain different timing. The method of pattern-oriented testing is aimed for generality. However, without further data is is unclear how general the identified patterns are. More tests using data of other species and sizes as well as hydraulic flow fields are required to expand the application boundaries. This is necessary to produce meaningful values for e.g. fishway efficiency.

In the ten best calibration simulations, the varied 15 parameters spread along their full bandwidth. This observation points to the assumption that some parameters correlate with each other and can be eliminated. This should be investigated in future work by fixing some parameters. My recommendation includes $r_{sameSpot}, U_g, t_{trap}, \gamma_{M,a}, \Delta_{Holding}, m_{F,i}$, and M_{ini} (Table 5.3), thus leaving 8 parameters for variation. 500 calibration simulations may seem few, but a higher number not necessarily improves calibration. For example, Wiegand et al. (2004) describe an IBM application where 557 simulations to find values for 13 parameters yielded similar estimates as 50 times more simulations.

Fish drag and thrust (Haefner and Bowen 2002; Arenas Amado 2012; Kerr et al. 2016) were not represented in the model. Providing this level of physical insight was not necessary for the goal of computing fishway efficiency.

A constant swim speed over ground, U_g, was assumed during upstream movement, independent of the flow velocity U_m. Concurring approaches would have been to choose swim velocity depending on e.g. an energy budget or the current motivation/fatigue ratio. However, explicit data supporting such models is rare (Castro-Santos 2006) and further calibration would be needed. Support for choosing U_g comes from the work of Castro-Santos (2005), who found that "...the distance-maximizing strategy for fish swimming against flow velocities equal to or greater than [their prolonged swim speed] is to swim at a constant groundspeed, regardless of [flow velocity]...". This author's results for striped bass agree well with my value, U_g, for prolonged swim speed of trout, confirming the assumption.

Stochasticity can be needed to account for representing processes not modeled, either because of ignorance or because they are not needed in detail (Grimm and Railsback 2005, p. 102). In the present behavior model, random effects are limited to a maximum of two spots per time step. The

test result without stochasticity in section 5.3 indeed suggests that missing information is added to the model.

Waiting times in the beginning of a test are not modeled, knowingly that a realistic representation of gathering of motivation could be important with respect to real-world tailrace applications. However, time limitations and alien environment in the flume could especially affect behavior in the start area. Thus, time spent in the start area was not considered.

A main objection against the older model of Goodwin et al. (2006) by Arenas Amado (2012) was the ability for stepwise, instantaneous velocity changes, which "requires an infinite acceleration". Velocity change per time step is also used in Goodwin et al. (2014) and the present work. However, as long as the time step is finite, the acceleration can not be infinite. In ELAM-de, swim speed is limited by flow velocity. As long as this is in realistic boundaries for a flume, swim speed is also bound to be realistic. Generally, this kind of direct modeling is justified because the underlying process is not relevant for the model and results are tested (Grimm and Railsback 2005, p. 261).

The preference of close proximity to the bottom in the observed vertical distribution was very closely resembled in the model results. Depth was used as a proxy for the pressure stimulus. This is possible because the flume flow was in hydrostatic equilibrium, as vertical acceleration was negligibly small. However, this pattern is considered as weak, because vertical behavior is not important to understand migration pathways and delays in the flume (see Goettel et al. 2015). Still it was valuable to ensure correct model function. Thus, the modeling approach and result both are satisfactory.

Considering these conclusions, a two-dimensional depth-averaged model approach could have been sufficient to approach the stronger patterns. However, with respect to future applications in a real-world tailrace, depth could be important and the capability to work in 3D allows more general use of the model framework.

5.4.2 Model structure

In this section, the behavior model structure is analyzed to find explanations for simulated behavior and theories for real behavior. Results of alternative behavior rules (section 5.3.1) are included in the analysis. It lies in the nature of a model that insights about the real world are indirect (Grimm and

Railsback 2005, p. 348). Models contribute to research by providing unlimited opportunities for theory testing (Willis 2011). This ability can aid field tests. This section is divided by theoretical aspects that were tested using contrasting model versions.

Direction choice — The choice of direction is fundamental to simulating upstream migration. Kerr et al. (2016) summarize the most common approach as the "general principle that space use is dictated by energy conservation". This approach is implemented e.g. by Abdelaziz (2013) and Zielinski et al. (2015) by moving to the upstream cell with the lowest velocity. However, this does neither explain the observed behavior of brown trout swimming in the jet of higher velocity, close to the right wall, nor the occurrence of turns.

Diverse stimuli can be imagined. For example, in Goettel et al. (2015), turbulence, water velocity, cover, and the presence of conspecifics appeared to be the most important variables for dace behavior. From this list, ELAMde enables testing of hydraulic parameter stimuli. Contrasting results of different behavior rules showed that using attraction to higher advective acceleration magnitude, $|\vec{a}|$ (Equation 4.7), the desired behavior emerged, unlike using low velocity, high velocity, or constant TKE as an attraction stimulus. This finding is supported by Goodwin et al. (2014), who showed that acceleration and pressure, p, are sufficient to reproduce downstream fish navigation behavior. It was confirmed by a test without any attraction to a hydraulic stimulus, which yielded a worse result. One explanation would be that the trout tested were not challenged by the velocity tested. Another possible reason is that acceleration integrates much information present in the flow field and yields a good approximation of the lateral line reception. In the light of the results, it seems to be more promising to be addressed in future work than the usual turbulence parameters, e.g. TKE or Reynolds stress.

In the vertical direction, the application of Goodwin's simple submodel and the hydrostatic pressure/vertical coordinate as stimulus were successful. Separate treatment compared to the mainly horizontal behaviors is justified, as vertical swimming/buoyancy positioning is usually distinct from lateral swimming (Willis 2011).

"Active/strong" and "passive/weak" groups — It was necessary to split the fish in two groups with different motivation and fatigue coefficients,

k_M and k_F, to match the turns and no-turns patterns, P(3) and P(4). Assuming that hydraulic influence was constant betweens tests and no other environmental influence was present, internal differences remain as a cause for differences. This is despite the trout had similar body lengths, which is often used as a basic indicator to classify fish. Also a similar activity level was ensured by only considering actively moving fish (line A criterion). Care must be taken to not split the sample in many different groups for easier calibration.

The current submodels applied are too coarse to allow conclusions about the cause, e.g. different boldness, exhaustion, or fitness levels. On a very general level, reaching high values of F could represent high metabolic cost. For constant coefficients, the values allow evaluation of competing fishway designs on a quantitative basis. Following this logic, setup *2c-A* would more efficient for trout than setup *1e-A*, because more fish finished with lower maximum fatigue. However, further validation is required to create trust in such conclusions.

Holding — As discussed in section 4.1.3 and appendix E, the behavior model structure of ELAM-2014 (Goodwin et al. 2014) was not suitable for application to upstream migration in small-scale. One reason for that is the assumption of continuous swimming (also pointed out by Arenas Amado (2012) with regard to Goodwin et al. 2006). This assumption is arguable for energy-saving migration with the flow, but not applicable to energy costly movement against the flow, as documented by drifting and holding phases in the flume tracks (section 2.2.2). In ELAM-2014, holding could indeed emerge in the form of *milling* around a point. As shown in this work it was necessary to integrate explicit holding and drifting behavior to obtain good results for the flume.

Repeated attempts — The ability to produce repeated slot entrance attempts was identified to be crucial for upstream moving fish models because this natural behavior increases success probability (Castro-Santos 2005, Goodwin 2015, pers. comm.). From a modeling perspective, the observed turns represent a similar behavior. It successfully emerged from the interaction of fatigue F and motivation M, the main variables of the new behavior model. Fatigue was not coupled to time as done traditionally, e.g. in the rules-of-thumb derived from Beamish (1978), also known as the *ethohydraulic scale*

(Gisen et al. 2017). This is because an IBM is far superior by allowing to consider acclimatization, or memory.

Memory functions of the classic form $\Phi^n_{avg} = (1 - m)\Phi^n + m\Phi^{n-1}_{avg}$ are used to acclimatize motivation, fatigue, pressure, and location parameters over time in ELAM-de. The formulation is simple, can be fine-tuned easily, and includes a temporal component. It underlines the necessity of simulating memory in fish behavior models to incorporate past experience into decisions, see e.g. Bracis et al. (2015) for a demonstration with respect to foraging. By turning off the spatial memory, too many turns were produced upstream. This could represent a spatial learning process (Odling-Smee and Braithwaite 2003) in the trout, steering preference for the acclimatized position and slowing progression.

In contrast to approaches forcing the fish to constantly swim upstream until failure (e.g. Abdelaziz 2013; Zielinski et al. 2015), the chosen approach allows the same model fish to behave differently at the same position.

Wall avoidance — Wall avoidance behavior was shown to contribute to model quality. Changing the direction close to a wall was done before using a distance of 15–5 BL (Arenas Amado 2012), or an undefined "close" distance (Haefner and Bowen 2002). Goodwin et al. (2014) simply reset fish moving behind a wall to a position inside the model domain. Using the flow direction, the present model has the most flexible mechanism, as it works also in curved geometries. This is easier and more robust than to compute the angle towards the boundary surface and derive a new direction from it.

5.4.3 Framework

After discussing results of the new behavior model, now technical details of the upgraded software framework are discussed with regard to the basic software framework part of ELAM-2014.

Arenas Amado (2012) criticized that the ELAM model of Goodwin et al. (2006) gets flow field information from a separate software, which "may hinder studies [...] under unsteady flow conditions". This author's work, however, did not use unsteady flow fields. There are two reasons why a steady flow field was also chosen for ELAM-de input: First, storing and reading the 3D flow field values for $N_t = 7200$ time steps, with an estimated size of 216 gigabyte, would cause massive computational cost. Second, the utility would be low, as the flow field was largely steady in the flume. Some

unsteady flow separations occurred, but it was not possible with reasonable effort to synchronize them with observed fish data, as would be required. Therefore, averaging was the best solution here. For future applications, the ELAM-de framework is able to read time-varying CFD results with little changes to the code due to its coupling to OpenFOAM.

The old framework used in ELAM-2014 ran several thousand times faster then the new one according to approximative comparisons. The probable main reason is that it utilizes contravariant space by working only on structured grids. Hence, the computational domain has to be split into single blocks for ELAM-2014 depending on the grid resolution and function, e.g. as an exit route. This renders it almost impossible to model fish behavior close to complex geometric features at a meter-scale, e.g. for a fishway entrance in a tailrace. To account for that issue, the ELAM-2014 framework was rebuild in ELAM-de to work with arbitrary unstructured meshes. That came at the price of dropping the ability to perform spatial operations in contravariant space, especially slowing variable interpolation and wall interactions.

Movement close to walls can cost a lot of computer resources. It is a minor issue for mid- and large-scale domains, where fish are unlikely to get close to a boundary too often. However, in the present small-scale model, it is computationally expensive. After every move, the fish center and six sensory points are checked if they are outside the model domain and if the last step did cross an internal wall. For clear, light water, this approach based on the sensory ovoid may even underestimate fish information, as their "vision" range is limited by the ovoid size. However, it seems to be a good estimate for real world applications, where vision is often very limited due to turbidity and darkness.

The free water surface represented in the CFD model caused special treatment in the framework implementation. This is a unique feature of the new model described in this work and allows to treat behavior at hydraulic head drops without simplifications, such as rigid lid boundaries. It is important to all applications featuring entrance slots or fishway slot.

Chapter 6

Conclusions and future work

6.1 Conclusions

In this work, a new individual-based model (IBM), "ELAM-de", for simulating upstream fish migration in small spatial and temporal scales was developed, calibrated, and validated using patterns of upstream moving trout in a model fishway entrance pool.

Compared to reviewed IBMs for upstream migration, ELAM-de is the first model to combine high temporal ($\Delta t = 0.5$ s) and spatial resolution in three dimensions ($\Delta_{x,y,z} \leq 5$ cm), a behavior model thoroughly tested against real fish data, and an open source computational fluid dynamics (CFD) model. It works on unstructured polyhedral meshes with free-surface flow fields from the open source CFD tool box OpenFOAM®. A new approach for wall avoidance was developed using the flow vector for orientation. It is more simple and more robust than existing approaches and works even on small-scale, curved geometries.

For behavior analysis in the ethohydraulic flume (EHF), mathematical definitions of six patterns were developed for consistent use in both laboratory and simulation experiments. The flume investigations were not originally designed for analyses of movement patterns, which means that no statistically valid interpretation was possible. Still, the pattern accuracy was sufficient for developing and testing ELAM-de, as not exact values, but characteristic qualitative patterns were needed.

From the results of the model, several conclusions can be drawn.

1. Advective acceleration magnitude, $|\vec{a}|$, is a functioning hydraulic stimu-

lus for horizontal direction choice of upstream migrating trout. It was used before in simulations of downstream migrating juvenile salmon close to dams. Pressure, p, approximated by the vertical depth, contributed to reproducing observed vertical distributions.

2. Contrasting different versions of the behavior model suggests that simple energy saving rules to choose upstream migration direction are not generally suitable. Neither high nor low velocity attraction led to emergence of the observed spatial patterns for brown trout.

3. The concept of splitting movement into three behaviors (forward, holding, and drifting) as well as the concept of balancing driving and restraining factors (summarized in *motivation* and *fatigue* variables) was necessary to produce all observed patterns. This model structure is general enough to be used in models of upstream movement in environments considerably larger than a flume. In combination with memory functions, it is able to produce repeated attempts to enter e.g. a fishway entrance, which is important for modeling attraction efficiency. However, further testing is required to evaluate the generality of the submodel rules. In the current formulation, the model is bound to trout and the two flume setups tested.

4. Brown trout in the EHF as well as shad and curimatá in two dam tailraces preferred proximity to geometrical boundaries while moving upstream or holding. However, these observations have to be confirmed by suitable experiments. Still, the observations support the practice common at the German federal waterways to place the main fishway entrance at the bank, in contrast to a mid-stream location.

Further work is recommended to confirm the conclusions, to make ELAM-de faster and more general, and to access new application areas.

6.2 Future work

There are three major fields in which the present work should be continued: Framework development, development for flume scenarios, and development for field-scale fishway attraction.

6.2.1 Framework development

In the state describe here, ELAM-de framework is limited to simulating relatively few fish at once. For future applications, it is important to improve performance. A first step would be to test if the issues with mesh insideness (section 4.2.2) are resolved in newer OpenFOAM versions. Then, a faster algorithm for spatial interpolation would be useful. For comparison, the current ELAM model framework of R.A. Goodwin (2017, pers. comm.) can handle time-varying hydraulic input and moving meshes of several hundred gigabyte size as well as interacting fish, e.g. for estuary or river reach applications. This is made possible by parallelization, code improvements, and commercial (Tecplot) libraries. The programming language used is still Fortran, but was updated from FORTRAN 70 to Fortran 90, like in the present work. It is likely that the current implementation of ELAM-de too could benefit from memory optimization (currently, all time step data is stored), parallelization (the fish loop is well-suited for that), and improved 3D interpolation. These steps would enable analysis of unsteady flow data, e.g. from a large-eddy simulation turbulence model. Still, it would be challenging to synchronize hydraulic and behavior data.

6.2.2 Flume scenarios

ELAM-de can be used with little effort to further investigate data from the EHF. Pattern data can be obtained using the methods developed in chapter 2.

A more comprehensive calibration, using the 8 suggested parameters instead of 15, could yield quantitatively better results. More testing using patterns clearly influenced by hydraulic change is required to make the behavior model more general and reliable. To improve time delay representation, changes to the motivation and/or fatigue submodel could be necessary.

Application to the short screen scenario and to other species than trout tested in the EHF would make the model more general and raise the opportunity to identify species-specific parameters. This could involve the need to include a shoaling submodel. Shoaling could to be more likely to occur in other species than brown trout, as anecdotal observations of *schneiders* and *nases* in the EHF show. It is easy to image that a dedicated shoaling model would help achieving good results for these species. A starting point

for model development would be the three classic "traffic rules" (collision avoidance, directional orientation, and cohesion) as described by Lopez et al. (2012).

Automatic 3D tracking of fish in the flume is in development. If such tracks become available, they could highly increase position and time accuracy of the movement data and lead to more precise pattern descriptions, which could further improve the model.

Analysis of trout weight and length data, which are available for the 2016 tests, could reveal body fat reserves. These could be analyzed with respect to fatigue coefficient as present in ELAM-de. For future tests, the collection of e.g. hormone data could show linkage to the motivation coefficient. If these data would correlate between lab and model, this would further corroborate the model approach and allow to investigate relations between internal state and hydraulics.

6.2.3 Fishway attraction

With additional development, the model can be used as a basis for evaluating field-scale fishway attraction in a dam tailrace. This is a prevailing research area in Germany (Gisen et al. 2017). If a hydropower facility is present at a dam, which is the case for 65 % of the 251 dams on the German federal waterways, it represents the most continuous discharge source. Following the observation that upstream migrating fish swim against the flow direction, it is common practice to build the entrance close to the powerhouse outlet (Clay 1995, pp. 67, 73). Because of the high costs associated with building close to existing structures, the details of an entrance configuration are highly relevant to management decisions. Commonly discussed categories include location, discharge, and geometrical design of a fishway entrance. For practical use, the categories require evaluation of differences in the order of some meters, hectoliters, and decimeters, respectively.

The high resolution required implies some challenges. Movement observations are crucial for model development, however costly and difficult to record (section 2.1.2). The ongoing campaign conducted by BfG at the Eddersheim dam on the Main river builds a database of 3D tracks (Thelma Biotel 2017), which could be utilized for this goal. 3D CFD modeling of a tailrace in high resolution is way more complex than a flume, but approaches for dealing with a turbine inlet exist, e.g. Gisen et al. (2017). It is likely that behavior

model rules have to be adapted to the new environment to better account for e.g. larger dimensions. Likewise, behavior patterns could be different and could require new analysis approaches. However, the basic concept of using active movement, holding, and drifting together with motivation and fatigue can be applied generally. The key stimuli identified in this work, acceleration and pressure, are promising to be applied in a tailrace model.

References

Abdelaziz, S. M. A. (2013). *Numerical simulation of fish behavior and fish movement through passages*. Vol. 129. Berichte des Lehrstuhls und der Versuchsanstalt für Wasserbau und Wasserwirtschaft der TU München. München. 129 pp. ISBN: 978-3-943683-05-9 (cit. on pp. 6, 54, 56, 57, 102, 104).

Adam, B. and B. Lehmann (2011). *Ethohydraulik. Grundlagen, Methoden und Erkenntnisse*. German. Berlin Heidelberg: Springer-Verlag Berlin Heidelberg. ISBN: 978-3-642172106. URL: http://dx.doi.org/10.1007/978-3-642-17210-6 (visited on Sept. 30, 2016) (cit. on pp. 8, 10, 17).

Adams, E. W. and A. I. Stamou (1989). "Bistable Flow Patterns in a Free Surface Water Channel". *Journal of Fluids Engineering* 111 (4), p. 408. ISSN: 00982202. DOI: 10.1115/1.3243660 (cit. on p. 37).

Arenas Amado, A. (2012). "Development and application of a mechanistic model to predict juvenile salmon swim paths". Dissertation. Iowa City: University of Iowa. 119 pp. (cit. on pp. 6, 53, 54, 56, 57, 100, 101, 103, 104).

BAW (1961). *Fischpässe an der Mosel. Modellversuche im Maßstab 1 : 1,75*. Gutachten. German. Karlsruhe: Bundesanstalt für Wasserbau (cit. on p. 9).

Beamish, F. W. H. (1978). "Swimming capacity". In: *Fish physiology. Volume VII: Locomotion*. Ed. by W. S. Hoar and D. J. Randall. New York: Academic Press, pp. 101–187. ISBN: 9780123504074 (cit. on pp. 55, 79, 87, 103).

BfG (2010). *Herstellung der Durchgängigkeit an Staustufen von Bundeswasserstraßen. Fischökologische Einstufung der Dringlichkeit von Maßnahmen für den Fischaufstieg*. German. Koblenz: Bundesanstalt für Gewässerkunde (cit. on p. 2).

Bickford, S. A. and J. R. Skalski (2000). "Reanalysis and Interpretation of 25 Years of Snake–Columbia River Juvenile Salmonid Survival Studies". *North American Journal of Fisheries Management* 20 (1), pp. 53–68. ISSN: 0275-5947. DOI: 10.1577/1548-8675(2000)020<0053:RAIOYO>2.0.CO;2 (cit. on p. 50).

Blank, D. M. (2008). "Advanced studies of fish passage through culverts: 1-D and 3-D hydraulic modeling of velocity, fish energy expenditure, and a new barrier assessment method". Bozeman, Montana, USA: Montana State University (cit. on p. 54).

Bleckmann, H. and R. Zelick (2009). "Lateral line system of fish". *Integrative zoology* 4 (1), pp. 13–25. ISSN: 1749-4877. DOI: 10.1111/j.1749-4877.2008.00131.x (cit. on p. 84).

Borne, M. v. d. (1881). *Die Fischerei-Verhältnisse des Deutschen Reiches, Oesterreich-Ungarns, der Schweiz und Luxemburgs. Bearbeitet im Auftrage des Deutschen Fischerei-Vereins*. German. Berlin: W. Moeser Hofbuchdruckerei. DOI: 10.18452/433 (cit. on p. 2).

BPA (1939). *First annual report of the Bonneville administrator*. Portland, OR: Bonneville Power Administration. URL: https://www.bpa.gov/Finance/FinancialInformation/AnnualReports/Pages/default.aspx (visited on Nov. 2, 2016) (cit. on p. 51).

— (2010). *Annual Report*. Portland, OR: Bonneville Power Administration. URL: https://www.bpa.gov/Finance/FinancialInformation/AnnualReports/Pages/default.aspx (visited on Nov. 2, 2016) (cit. on p. 51).

Bracis, C., E. Gurarie, B. van Moorter, and R. A. Goodwin (2015). "Memory Effects on Movement Behavior in Animal Foraging". *PloS one* 10 (8), e0136057. ISSN: 1932-6203. DOI: 10.1371/journal.pone.0136057 (cit. on p. 104).

Camazine, S., J.-L. Deneubourg, N. R. Franks, J. Sneyd, G. Theraula, and E. Bonabeau (2003). *Self-organization in biological systems*. 2nd ed. Princeton studies in complexity. Princeton, N.J.: Princeton University Press. 538 pp. ISBN: 9780691116242 (cit. on p. 4).

Castro-Santos, T. (2005). "Optimal swim speeds for traversing velocity barriers: an analysis of volitional high-speed swimming behavior of migratory fishes". *Journal of Experimental Biology* 208 (3), pp. 421–432. ISSN: 00220949. DOI: 10.1242/jeb.01380 (cit. on pp. 87, 88, 100, 103).

— (2006). "Modeling the Effect of Varying Swim Speeds on Fish Passage through Velocity Barriers". *Transactions of the American Fisheries Society* 135 (5), pp. 1230–1237. ISSN: 0002-8487. DOI: 10.1577/T05-262.1 (cit. on p. 100).

Chapman, S. J. (2004). *Fortran 90/95 for Scientists and Engineers*. Second Edition (International). New York: McGraw-Hill. ISBN: 007-123233-8 (cit. on p. 60).

Clay, C. H. (1995). *Design of Fishways and Other Fish Facilities*. CRC Press, Inc. (cit. on p. 109).

Cooke, S. J. and S. G. Hinch (2013). "Improving the reliability of fishway attraction and passage efficiency estimates to inform fishway engineering, science, and practice". *Ecological Engineering* 58, pp. 123–132. DOI: 10.1016/j.ecoleng.2013.06.005 (cit. on p. 3).

Cornu, V., P. Baran, D. Calluaud, and L. David (2012). "Effects of Various Configurations of Vertical Slot Fishways on Fish Behaviour in an Experimental Flume". In: *9th International Symposium on Ecohydraulics*. 9th International Symposium on Ecohydraulics. (Vienna). Ed. by H. Mader and J. Kraml (cit. on pp. 10, 15, 16).

Czerny, R. and C. Schütz (2017). "Ethohydraulische Versuche zur Untersuchung der Passierbarkeit von Einstiegsbecken in Fischaufstiegsanlagen". In: *Tagungsunterlagen*. HTG Kongress. (Duisburg). Ed. by HTG. Hafentechnische Gesellschaft (cit. on p. 18).

Duguay, J. M., R. Lacey, and J. Gaucher (2017). "A case study of a pool and weir fishway modeled with OpenFOAM and FLOW-3D". *Ecological Engineering* 103, pp. 31–42. DOI: 10.1016/j.ecoleng.2017.01.042 (cit. on p. 33).

DWA (2014). *Merkblatt DWA-M 509. Fischaufstiegsanlagen und fischpassierbare Bauwerke - Gestaltung, Bemessung, Qualitätssicherung*. German. Ed. by Deutsche Vereinigung für Wasserwirtschaft, Abwasser und Abfall e.V. Hennef (cit. on p. 80).

Fish, F. E. (2010). "Swimming Strategies for Energy Economy". In: *Fish locomotion. An eco-ethological perspective*. Ed. by P. Domenici and B. G. Kapoor. Enfield, New Hampshire: Science Publishers, pp. 90–122. ISBN: 978-1-57808-448-7 (cit. on p. 88).

Gao, Z., H. I. Andersson, H. Dai, F. Jiang, and L. Zhao (2016). "A new Eulerian–Lagrangian agent method to model fish paths in a vertical slot fishway". *Ecological Engineering* 88, pp. 217–225. DOI: 10.1016/j.ecoleng.2015.12.038 (cit. on pp. 6, 15, 54, 56, 57, 79, 84).

Giesecke, J., S. Heimerl, and E. Mosonyi (2013). *Wasserkraftanlagen. Planung, Bau und Betrieb*. 6th ed. Luxemburg, Berlin, and Heidelberg: Springer (cit. on p. 45).

Gisen, D. C. (2014). "Generation of a 3D Mesh Using snappyHexMesh Featuring Anisotropic Refinement and Near-wall Layers". In: *Proceedings of the 11th International Conference on Hydroscience & Engineering*. ICHE. (Hamburg). Ed. by R. Lehfeldt and R. Kopmann, pp. 983–990. ISBN: 978-3-939230-32-8 (cit. on p. 34).

Gisen, D. C., R. B. Weichert, and J. M. Nestler (2017). "Optimizing attraction flow for upstream fish passage at a hydropower dam employing 3D Detached-Eddy Simulation". *Ecological Engineering* 100, pp. 344–353. DOI: 10.1016/j.ecoleng.2016.10.065 (cit. on pp. 104, 109).

Goettel, M. T., J. F. Atkinson, and S. J. Bennett (2015). "Behavior of western blacknose dace in a turbulence modified flow field". *Ecological Engineering* 74, pp. 230–240. DOI: 10.1016/j.ecoleng.2014.10.012 (cit. on pp. 10, 15, 20, 37, 94, 101, 102).

Goodwin, R. A. (2004). "Hydrodynamics and Juvenile Salmon Behaviour at Lower Granite Dam. Decoding the Relationship Using 3-D Space-Time (CEL AGENT IBM) Simulation". Dissertation. Cornell University. 213 pp. (cit. on p. 50).

Goodwin, R. A., J. M. Nestler, J. J. Anderson, L. J. Weber, and D. P. Loucks (2006). "Forecasting 3-D fish movement behavior using a Eulerian–Lagrangian–agent method (ELAM)". *Ecological Modelling* 192 (1-2), pp. 197–223. ISSN: 03043800. DOI: 10.1016/j.ecolmodel.2005.08.004 (cit. on pp. 4, 35, 50, 55, 67, 68, 84, 101, 103, 104).

Goodwin, R. A., M. Politano, J. W. Garvin, J. M. Nestler, D. Hay, J. J. Anderson, L. J. Weber, E. Dimperio, D. L. Smith, and M. Timko (2014). "Fish navigation of large dams emerges from their modulation of flow field experience". *Proceedings of the National Academy of Sciences* 111 (14), pp. 5277–5282. ISSN: 0027-8424. DOI: 10.1073/pnas.1311874111 (cit. on pp. IV, 6, 37, 50, 52, 53, 56, 57, 66, 75, 78, 79, 81, 84, 88, 90, 92, 99, 101–104, 140).

Grimm, V. and S. F. Railsback (2005). *Individual-based modeling and ecology*. Princeton series in theoretical and computational biology. Princeton: Princeton University Press. 428 pp. ISBN: 1400850622 (cit. on pp. 3, 4, 56, 75, 92, 99–101).

— (2012). "Pattern-oriented modelling: a 'multi-scope' for predictive systems ecology". *Philosophical transactions of the Royal Society of London. Series B, Biological sciences* 367 (1586), pp. 298–310. ISSN: 0962-8436. DOI: 10.1098/rstb.2011.0180. eprint: 22144392 (cit. on pp. 22, 30, 71).

Grimm, V., U. Berger, F. Bastiansen, S. Eliassen, V. Ginot, J. Giske, J. Goss-Custard, T. Grand, S. K. Heinz, G. Huse, A. Huth, J. U. Jepsen, C. Jørgensen, W. M. Mooij, B. Müller, G. Pe'er, C. Piou, S. F. Railsback, A. M. Robbins, M. M. Robbins, E. Rossmanith, N. Rüger, E. Strand, S. Souissi, R. A. Stillman, R. Vabø, U. Visser, and D. L. DeAngelis (2006). "A standard protocol for describing individual-based and agent-based models". *Ecological Modelling* 198 (1-2), pp. 115–126. ISSN: 03043800. DOI: 10.1016/j.ecolmodel.2006.04.023 (cit. on p. 75).

Grimm, V., U. Berger, D. L. DeAngelis, J. G. Polhill, J. Giske, and S. F. Railsback (2010). "The ODD protocol. A review and first update". *Ecological Modelling* 221 (23), pp. 2760–2768. ISSN: 03043800. DOI: 10.1016/j.ecolmodel.2010.08.019 (cit. on p. 75).

Groot, S. J. de (1990). "The former allis and twaite shad fisheries of the lower Rhine, The Netherlands". *J. Appl. Ichthyol. (Journal of Applied Ichthyology)* 6 (4), pp. 252–256. DOI: 10.1111/j.1439-0426.1990.tb00587.x (cit. on p. 2).

Haefner, J. W. and M. D. Bowen (2002). "Physical-based model of fish movement in fish extraction facilities". *Ecological Modelling* 152 (2-3), pp. 227–245. DOI: 10.1016/S0304-3800(02)00006-6 (cit. on pp. 55–57, 100, 104).

Haselbauer, M. A. and C. Barreira Martinez (2011). "Turbulente Strukturen als Erfolgskriterium von Fischaufstiegsanlagen". German. *WasserWirtschaft* (09), pp. 42–45 (cit. on p. 33).

Heinzelmann, C., R. Weichert, and S. Wassermann (2013). "Hydraulische Untersuchungen zum Bau einer Fischaufstiegsanlage in Lauffen am Neckar". German. *WasserWirtschaft* 103 (1/2), pp. 26–32 (cit. on p. 3).

Herbert-Read, J. E., A. Perna, R. P. Mann, T. M. Schaerf, D. J. T. Sumpter, and A. J. W. Ward (2011). "Inferring the rules of interaction of shoaling fish". *Proceedings of the National Academy of Sciences of the United States of America* 108 (46), pp. 18726–18731. DOI: 10.1073/pnas.1109355108. eprint: 22065759 (cit. on p. 80).

Hogan, T., C. Wright, and S. Medford (2012). *Upstream Passage Assessment of American Shad Using 3D Acoustic Telemetry*. Alden Lab (cit. on pp. 12–14).

ICPR (2004). *Rhine & Salmon 2020. A Programme for Migratory Fish in the Rhine System*. Koblenz: International Commission for the Protection of the Rhine (cit. on p. 2).

Katz, Y., K. Tunstrom, C. C. Ioannou, C. Huepe, and I. D. Couzin (2011). "Inferring the structure and dynamics of interactions in schooling fish". *Proceedings of the National Academy of Sciences of the United States of America* 108 (46), pp. 18720–18725. ISSN: 0027-8424. DOI: 10.1073/pnas.1107583108 (cit. on p. 80).

Kerr, J. R., C. Manes, and P. S. Kemp (2016). "Assessing hydrodynamic space use of brown trout, Salmo trutta, in a complex flow environment: a return to first principles". *Journal of Experimental Biology* 219 (Pt 21), pp. 3480–3491. ISSN: 00220949. DOI: 10.1242/jeb.134775. eprint: 27591311 (cit. on pp. 11, 15, 30, 84, 100, 102).

Kirschmer, O. (1925). "Untersuchungen über den Gefällsverlust an Rechen". Hydraulisches Institut. German. Dissertation. München: Technische Hochschule München (cit. on pp. 45–47).

Koch, W. (1929). "Fischpässe und Fischwanderung am Neckar". German. *Badische Fischerei-Zeitung* (8-9), pp. 1–15 (cit. on p. 2).

Koehn, J. D. and D. A. Crook (2013). "Movements and migration". In: *Ecology of Australian Freshwater Fishes*. Ed. by P. Humphries and K. F. Walker. Collingwood, Vic.: CSIRO Publishing, pp. 105–129. ISBN: 9780643097445 (cit. on p. 1).

Landwüst, C. v. (2004). "Fischbestand und Fischerei in der staugeregelten Mosel". German. In: *Über Fische und Fischerei in durch Eingriff des Menschen veränderten Fließgewässern*. VDSF-Seminar. Ed. by VDSF. Fischerei und Naturschutz 6. Verband Deutscher Sportfischer e.V. Offenbach, pp. 59–75. ISBN: 1617-9978 (cit. on p. 2).

Lelek, A. and G. Buhse (1992). *Fische des Rheins. Früher und heute*. German. Berlin, Heidelberg, and New York: Springer. 214 pp. ISBN: 3540538143 (cit. on p. 2).

Lenders, H. J. R., T. P. M. Chamuleau, A. J. Hendriks, R. C.G. M. Lauwerier, R. S.E. W. Leuven, and W. C.E. P. Verberk (2016). "Historical rise of waterpower initiated the collapse of salmon stocks". *Scientific reports* 6, pp. 1–9. ISSN: 2045-2322. DOI: 10.1038/srep29269 (cit. on p. 2).

Liao, J. C. (2007). "A review of fish swimming mechanics and behaviour in altered flows". *Philosophical Transactions of the Royal Society B: Biological Sciences* 362 (1487), pp. 1973–1993. ISSN: 0962-8436. DOI: 10.1098/rstb.2007.2082 (cit. on pp. 9, 27, 78, 87).

Lopez, U., J. Gautrais, I. D. Couzin, and G. Theraulaz (2012). "From behavioural analyses to models of collective motion in fish schools". *Interface focus* 2 (6), pp. 693–707. ISSN: 2042-8898. DOI: 10.1098/rsfs.2012.0033. eprint: 24312723 (cit. on pp. 80, 109).

Lucas, M. C. and E. Baras (2001). *Migration of freshwater fishes*. 1st ed. Oxford: Blackwell Science. 420 pp. ISBN: 0-632-05754-8 (cit. on pp. 1, 78).

Malcherek, A. (2016). "History of the Torricelli Principle and a New Outflow Theory". *Journal of Hydraulic Engineering* 142 (11). ISSN: 0733-9429. DOI: 10.1061/(ASCE)HY.1943-7900.0001232 (cit. on p. 44).

McKay, M. D., R. J. Beckman, and W. J. Conover (1979). "Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output from a Computer Code". *Technometrics* 21 (2), pp. 239–245. ISSN: 0040-1706. DOI: 10.1080/00401706.1979.10489755 (cit. on p. 73).

McMichael, G. A., M. B. Eppard, T. J. Carlson, J. A. Carter, B. D. Ebberts, R. S. Brown, M. Weiland, G. R. Ploskey, R. A. Harnish, and Z. D. Deng (2010). "The Juvenile Salmon Acoustic Telemetry System. A New Tool". *Fisheries* 35 (1), pp. 9–22. ISSN: 0363-2415. DOI: 10.1577/1548-8446-35.1.9 (cit. on pp. 12, 51).

Menter, F. and T. Esch (2001). "Elements of Industrial Heat Transfer Predictions". In: *Proceedings. Invited Lectures - Engineering for the New Millenium*. 16th Brazilian Congress of Mechanical Engineering. (Uberlândia). Ed. by COBEM. 20, pp. 117–127. ISBN: ISBN 85-85769-06-6 (cit. on p. 34).

Meusburger, H. (2002). "Energieverluste an Einlaufrechen von Flusskraftwerken". Versuchsanstalt für Wasserbau, Hydrologie und Glaziologie. German. Doctoral thesis. Zürich: Eidgenössische Technische Hochschule (cit. on pp. 45, 46).

Mortensen, J. B. (2012). "Agent-Based Modelling of Short-Term Juvenile Bull Shark Movement in a Semi-Enclosed Gold Coast Estuary". Master thesis. Brisbane: Griffith University. URL: https://www120.secure.griffith.edu.au/rch/items/025c9d8f-5a7e-e03b-3c46-7e6900389869/1/ (visited on Oct. 31, 2016) (cit. on p. 57).

Munson, B. R., T. H. Okiishi, W. W. Huebsch, and A. P. Rothmayer (2013). *Fundamentals of fluid mechanics*. 7th edition. Hoboken, NJ, USA: John Wiley & Sons, Inc. 747 pp. ISBN: 978-1-118-11613-5 (cit. on p. 63).

Noble, R., J. R. Dodd, J. D. Bolland, S. E. Walton, and I. G. Cowx (2014). *Investigating Fish Passage: Acoustic Fish Tracking Project – Yorkshire Esk, Ruswarp. 2013 Initial Post-Commissioning dataset*. York: Environment Agency (cit. on p. 14).

Noble, R., J. R. Dodd, J. D. Bolland, S. E. Walton, and I. G. Cowx (2015). *Investigating Fish Passage: Acoustic Fish Tracking Project – Yorkshire Esk, Ruswarp. 2014 Second Post-Commissioning dataset*. York: Environment Agency (cit. on p. 14).

Odling-Smee, L. and V. A. Braithwaite (2003). "The role of learning in fish orientation". *Fish and Fisheries* 4 (3), pp. 235–246 (cit. on pp. 79, 104).

OpenCFD Limited (2011). *OpenFOAM User Guide. Version 2.0.0* (cit. on p. 34).

— (2014). *OpenFOAM Programmer's Guide. Version 2.3.1* (cit. on pp. 63, 64).

Pitcher, T. J. and J. K. Parish (1993). "Functions of shoaling behaviour in teleosts". In: *Behaviour of teleost fishes*. Ed. by T. J. Pitcher. 2nd ed. Fish and fisheries series 7. London and New York: Chapman & Hall, pp. 363–404. ISBN: 9780412429408 (cit. on p. 27).

Przybilla, A., S. Kunze, A. Rudert, H. Bleckmann, and C. Brücker (2010). "Entraining in trout: a behavioural and hydrodynamic analysis". *Journal of Experimental Biology* 213 (Pt 17), pp. 2976–2986. ISSN: 00220949. DOI: 10.1242/jeb.041632. eprint: 20709926 (cit. on pp. 9, 15, 30, 31).

Raynal, S., L. Chatellier, D. Courret, M. Larinier, and L. David (2013). "An experimental study on fish-friendly trashracks – Part 2. Angled trashracks". *Journal of Hydraulic Research* 51 (1), pp. 67–75. DOI: 10.1080/00221686.2012.753647 (cit. on pp. 45, 46).

Rodi, W. (2017). "Turbulence Modeling and Simulation in Hydraulics: A Historical Review". *Journal of Hydraulic Engineering* 143 (5). DOI: 10.1061/(ASCE)HY.1943-7900.0001288 (cit. on p. 34).

Rodríguez, Á., M. Bermúdez, J. R. Rabuñal, and J. Puertas (2015). "Fish tracking in vertical slot fishways using computer vision techniques". *Journal of Hydroinformatics* 17 (2), pp. 275–292. ISSN: 14647141. DOI: 10.2166/hydro.2014.034 (cit. on pp. 11, 15, 16, 23).

Sanz-Ronda, F. J., J. Ruiz-Legazpi, F. J. Bravo-Córdoba, S. Makrakis, and T. Castro-Santos (2015). "Sprinting performance of two Iberian fish. Luciobarbus bocagei and Pseudochondrostoma duriense in an open channel flume". *Ecological Engineering* 83, pp. 61–70. DOI: 10.1016/j.ecoleng.2015.05.033 (cit. on pp. 11, 15).

Schulze, L. and C. Thorenz (2014). "The Multiphase Capabilities of the CFD Toolbox OpenFOAM for Hydraulic Engineering Applications". In: *Proceedings of the 11th International Conference on Hydroscience & Engineering*. ICHE. (Hamburg). Ed. by R. Lehfeldt and R. Kopmann, pp. 1007–1014. ISBN: 978-3-939230-32-8 (cit. on p. 34).

Schütz, C., N.N., and N.N. (2017). "Adding auxiliary discharge through a horizontal bar screen into the entrance pool of a fishway. Influence of screen and flow dimensions on fish passage". Manuscript in preparation (cit. on pp. 6, 17, 29, 30).

Smith, D. L., T. Threadgill, J. M. Nestler, and R. A. Goodwin (2012). *Modeling fish movement in the Mississippi River*. Ed. by Bundesanstalt für Wasserbau. Karlsruhe, Germany (cit. on pp. 52, 55–57, 140, 141).

Sprankle, K. (2005). "Interdam Movements and Passage Attraction of American Shad in the Lower Merrimack River Main Stem". *North American Journal of Fisheries Management* 25 (4), pp. 1456–1466. ISSN: 0275-5947. DOI: 10.1577/M04-049.1 (cit. on p. 12).

Steig, T., C. Sullivan, and S. Johnston (2013). *Evaluating Fish Passage in Noisy Environments Using Acoustic Telemetry*. Ed. by Oregon State University. Corvallis, USA: HTI Hydroacoustics Technology (cit. on p. 12).

Stillman, R. A., S. F. Railsback, J. Giske, U. Berger, and V. Grimm (2015). "Making Predictions in a Changing World: The Benefits of Individual-Based Ecology". *Bioscience* 65 (2), pp. 140–150. ISSN: 0006-3568. DOI: 10.1093/biosci/biu192. eprint: 26955076 (cit. on p. 4).

Suzuki, F. M., J. B. Dunham, L. G. M. Silva, C. B. M. Alves, and P. S. Pompeu (2017). "Factors Influencing Movements of Two Migratory Fishes within the Tailrace of a Large Neotropical Dam and their Implications for Hydropower Impacts". *River Research and Applications* 33 (4), pp. 514–523. ISSN: 1535-1459. DOI: 10.1002/rra.3105 (cit. on p. 12).

Suzuki, F. M. (2014). "Estudo do comportamento de peixes no canal de fuga da usina hidrelétrica de Três Marias utilizando telemetria acústica". Portuguese. Lavras, Brazil: Universidade Federal de Lavras (cit. on pp. 12–14).

Thelma Biotel (2017). *Pinpoint report - River Main*. Ed. by Unpublished. Trondheim (cit. on pp. 14, 109).

Thiele, J. C., W. Kurth, and V. Grimm (2014). "Facilitating Parameter Estimation and Sensitivity Analysis of Agent-Based Models. A Cookbook Using NetLogo and 'R'". *Journal of Artificial Societies and Social Simulation* 17 (3). ISSN: 1460-7425. DOI: 10.18564/jasss.2503 (cit. on p. 73).

Vettori, S. (2017). "Analyse von Strömungsmustern in einem hydraulischen Labormodell eines Dotationsbeckens". Institut für Wasser und Gewässerentwicklung. German. Master thesis. Karlsruhe: Karlsruher Institut für Technologie (cit. on p. 36).

Viscido, S. V., J. K. Parrish, and D. Grünbaum (2004). "Individual behavior and emergent properties of fish schools. A comparison of observation and theory". *Marine Ecology Progress Series* 273, pp. 239–249. ISSN: 0171-8630. DOI: 10.3354/meps273239 (cit. on p. 27).

Wang, R. W., L. David, and M. Larinier (2010). "Contribution of experimental fluid mechanics to the design of vertical slot fish passes". *Knowledge and Management of Aquatic Ecosystems* (396), p. 2. DOI: 10.1051/kmae/2010002 (cit. on pp. 9, 10, 15, 16).

Weller, H. G., G. Tabor, H. Jasak, and C. Fureby (1998). "A tensorial approach to computational continuum mechanics using object-oriented techniques". *Computers in Physics* 12 (6), pp. 620–631 (cit. on p. 33).

Wiegand, T., E. Revilla, and F. Knauer (2004). "Dealing with Uncertainty in Spatially Explicit Population Models". *Biodiversity and Conservation* 13 (1), pp. 53–78. ISSN: 0960-3115. DOI: 10.1023/B:BIOC.0000004313.86836.ab (cit. on pp. 99, 100).

Willis, J. (2011). "Modelling swimming aquatic animals in hydrodynamic models". *Ecological Modelling* 222 (23-24), pp. 3869–3887. ISSN: 03043800 (cit. on pp. 50, 79, 102).

Zielinski, D., V. R. Voller, and P. W. Sorensen (2015). *Three Dimensional Swimming-Fatigue Model to Predict Passage of Asian Carp at a Lock and Dam*. Ed. by University of Massachusetts - Amherst. Groningen, The Netherlands. URL: http://scholarworks.umass.edu/fishpassage_conference/2015/June24/54/ (visited on Aug. 4, 2016) (cit. on pp. 6, 53, 56, 57, 102, 104).

Online resources

BfG (2017). URL: http://www.bafg.de/DE/Service/presse/2017_04_07.html?nn=168662 (visited on July 12, 2017) (cit. on p. 3).

DLF Kultur (2008). URL: http://www.deutschlandfunkkultur.de/vor-50-jahren-als-die-wasserqualitaet-zum-problem-wurde.984.de.html?dram:article_id=153423 (visited on July 12, 2017) (cit. on p. 2).

fishbase.org (2016). URL: http://www.fishbase.org/Glossary/Glossary.php?q=total+length (visited on Nov. 15, 2016) (cit. on p. 119).

github.com (2015a). URL: https://github.com/OpenFOAM/OpenFOAM-3.0.x/commit/34e062daab1575c7e698ef3c7296b07b84bbd232 (visited on May 15, 2017) (cit. on p. 61).

— (2015b). URL: https://github.com/OpenFOAM/OpenFOAM-dev/commit/de2c2fb007c4c6f0936c036270e5cf633d775d71 (visited on May 15, 2017) (cit. on p. 61).

— (2016). URL: https://github.com/OpenFOAM/OpenFOAM-4.x/blob/master/applications/utilities/mesh/generation/snappyHexMesh/snappyHexMeshDict (visited on May 15, 2017) (cit. on p. 61).

microsoft.com (2015). URL: https://msdn.microsoft.com/en-us/library/aa293547%28v=vs.60%29.aspx (visited on Oct. 20, 2015) (cit. on p. 125).

swarm.org (2017). URL: http://www.swarm.org/wiki/Swarm_main_page (visited on Aug. 21, 2017) (cit. on p. 6).

WFBW (2017). URL: http://www.wfbw.de/aktuelles/kontrollstationen/iffezheim/ (visited on July 12, 2017) (cit. on p. 3).

Wikipedia (2017). URL: https://de.wikipedia.org/wiki/Main#Die_Mainschiffahrt_im_Industriezeitalter (visited on July 11, 2017) (cit. on p. 2).

Wilenski, U. (2017). *Netlogo website*. URL: http://ccl.northwestern.edu/netlogo/index.shtml (visited on Sept. 1, 2017) (cit. on p. 6).

yolinux.com (2015). URL: http://www.yolinux.com/TUTORIALS/LinuxTutorialMixingFortranAndC.html (visited on Feb. 22, 2016) (cit. on p. 125).

Acronyms

ADCP	Acoustic Doppler Current Profiler
ADV	Acoustic Doppler Velocimeter/Velocimetry
BAW	Bundesanstalt für Wasserbau (Federal Waterways Engineering and Research Institute)
BCRW	Biased correlated random walk
BfG	Bundesanstalt für Gewässerkunde (Federal Institute of Hydrology)
BL	Body length[1] (of a fish)
CFD	Computational Fluid Dynamics
EHF	Ethohydraulic flume
ELAM	Eulerian-Lagrangian-agent method
FVM	Finite Volume Method
IBM	Individual-based modeling/model
LHS	Latin hypercube sampling
OPD	Overall pattern deviation
PIT	Passive integrated transponder
RANS	Reynolds-averaged Navier-Stokes (equations)
RMSE	Root-mean-square error/deviation (RMSD)
SD	Standard deviation (s for sample, σ for population)
SMS	Surface-Water Modelling System
TKE	Turbulence kinetic energy[2]
UML	Unified Modeling Language
USACE	United States Army Corps of Engineers
VOF	Volume-of-fluid (method)
WFD	(European) Water Framework Directive

[1] Equivalent to Total Length, TL, in the context of this thesis. TL is "the greatest length of the whole body between the most anterior point of the body and the most posterior point, in a straight line, not over the curve of the body" (fishbase.org 2016).

[2] Turbulen<u>t</u> kinetic energy is more common, but semantically wrong. Not the energy is turbulent, but the turbulence contains energy.

List of Figures

List of Tables

Appendices

Appendix A

Running ELAM-de

Prerequisites: A computer with OpenFOAM 2.3.1 installed running a Linux operating system, e.g. the high performance computer *Hera* of BAW. OpenFOAM hydraulic results of the case to investigate, including the fields `UMean` and `acclMag` (section B.2). Areas not accessible have to be cut, outlets have to be marked as `patch`, and internal walls have to be marked as `inGroups internal` in the file `constant/polyMesh/boundary`.

1. Source OpenFOAM environment.
2. Create an ELAM-de case folder, containing the ELAM-de executable, a folder "input", a folder "output", and (optional) a SLURM submit script for Hera. Optional script `newCase.sh` can be used for this step.
3. Edit input folder containing files agentBehaviorCoefficients.inp, fishPositions.inp, rules.inp, simSettings.inp.
4. Start using the console command e.g. `ELAM-beta -case /lustre/w1/gisend/eh-rinne/07_lang_pfosten_slot_cut/`. Change path and executable name if needed. Optional script `submit_elam-beta.slurm` can be used for this step.
5. Postprocess using MATLAB scripts and Tecplot layout files.

Appendix B

ELAM-de development

B.1 Compatibility C++/Fortran

The main ELAM-de program is written in C++, while the behavior model and some functions are written in Fortran. To call a Fortran subroutine from C++, it must be declared as a *function prototype* and wrapped as C code using the `extern "C" {...}` command. Its name has to be all lower-case and an underscore needs to be appended. All arguments in (and passed to) Fortran must be passed by reference, not by value. C++ automatically applies reference-style passing to array arguments: they decay to a pointer pointing at the first element of the array. For compatibility reasons, only variables or plain arrays are allowed to be passed to Fortran. Object-type data storages like `std::vector` can not be used. Compatible data types are listed in Table B.1.

Fortran indexes the first element of an array as 1, whereas C++ treats it as a pointer with an iteration of 0. For example, to access the third element of an array available in both languages, one needs to address it as `array(3)` (Fortran) or `array[2]` (C++). For multi-dimensional arrays, one should note that C++ varies the subscripts in row-major order and Fortran in column-major order (microsoft.com 2015). The practical implication is that the order

Table B.1: Equivalent variable types in Fortran and C++ used for the present work (partly based on yolinux.com 2015). The definition space is bound by memory size.

Fortran 90	C++ 11	Definition	Memory size
`integer`	`int`	$\mathbb{Z}$	4 bytes = 32 bits
`double*8`	`double`	$\mathbb{R}$	8 bytes = 64 bits
`logical*1`	`bool`	(true,false)	1 byte = 8 bits

of indices is reversed between the languages. For example, `array(5,4,3)` refers to the same element as `array[2][3][4]`.

C++ makes heavy use of pointers, as they allow to operate with large data blocks very efficiently. For example, a large array of a known data type is defined just by the pointer to its first element in memory and its size. The explanation of two operators is stated here for reference:

`&myvar` – The ampersand in this case is the address-of operator. It passes the memory address of the variable *myvar* if used in a function call. It reads "address of". For example, the receiving function defines the pointer *mypointer* which stores the memory address.

`*mypointer` – The asterisk in this case is the dereference operator. It returns the value of the variable at the memory address pointed at by the pointer *mypointer*. It reads "value pointed to by". In our example, it is the value of *myvar*.

B.2 acclMag tool source code

This tool was derived from an existing OpenFOAM tool and computes the advective acceleration magnitude field from the mean velocity field.

```
1  /*---------------------------------------------------------*\
2  License
3      This program is free software: you can redistribute it
4      and/or modify it under the terms of the GNU General
5      Public License as published by the Free Software
6      Foundation, either version 3 of the License, or (at your
7      option) any later version.
8
9      This program is distributed in the hope that it will be
10     useful, but WITHOUT ANY WARRANTY; without even the
11     implied warranty of MERCHANTABILITY or FITNESS FOR A
12     PARTICULAR PURPOSE.  See the GNU General Public License
13     for more details.
14
15     You should have received a copy of the GNU General Public
16     License along with this program.  If not, see
17     <http://www.gnu.org/licenses/>.
18
19     Copyright (C) 2011-2012, OpenFOAM Foundation
20     Copyright (C) 2015-2016, David Gisen, Bundesanstalt fuer
21     Wasserbau, www.baw.de
22
23 Application
24     acclMag
25
26 Description
27     This utility calculates and outputs the magnitude of the
28     time-averaged 3D acceleration field.
29
```

```
30      The -noWrite option just outputs the max/min values
31      without writing the field.
32  \*---------------------------------------------------------------*/
33  #include "calc.H"
34  #include "fvc.H"
35  // * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //
36  void Foam::calc(const argList& args, const Time& runTime, ↩
        const fvMesh& mesh)
37  {
38      bool writeResults = !args.optionFound("noWrite");
39      IOobject UMeanHeader
40      (
41          "UMean",
42          runTime.timeName(),
43          mesh,
44          IOobject::MUST_READ
45      );
46
47      if (UMeanHeader.headerOk())
48      {
49          Info<< "    Reading UMean" << endl;
50          volVectorField UMean(UMeanHeader, mesh);
51          volTensorField gradUMean(fvc::grad(UMean));
52
53          volScalarField acclMag
54          (
55              IOobject
56              (
57                  "acclMag",
58                  runTime.timeName(),
59                  mesh,
60                  IOobject::NO_READ,
61                  IOobject::NO_WRITE
62              ),
63              mag(UMean & gradUMean)
64          );
65
66          Info<< "acclMag max/min : "
67              << max(acclMag).value() << " "
68              << min(acclMag).value() << endl;
69
70          if (writeResults)
71          {
72              acclMag.write();
73          }
74      }
75      else
76      {
77          Info<< "    UMean not present" << endl;
78      }
79      Info<< "\nEnd\n" << endl;
80  }
81  // * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //
```

B.3 BehaviorRule interface

Figure B.1: Variable interface to subroutine/function BehaviorRule as of ELAM-2014, with adaptations for OpenFOAM compatibility, not including later changes for the new behavior model presented in this work. Gray background stands for omitted variables.

Subroutine BehaviorRule(				**Equivalence in DRIVER code:** *(italics added later)*	**Provided by:**
I	FishNumber,	! Local	scalar	FN	Current FN, by calling loop over Nfish in StrELAM_main.f/581
I	NFish,	! Local	scalar	*Moved from line 56 because of order in C++ argument list*	Coeff(55), linecount VfishC_1.inp
I	TimeStep,	! Local	scalar	TimeStep	Calling loop over TotTimesteps
I	TimeStepLength,	! Local	scalar	DT	SimSettings.inp / INPUTVALUES()
I	NAgents,	! Local	scalar		'= coefficients(4) / but only 4 are used here
I	NTecFieldVar,	! Local	scalar	*# of environm. var. in addition to x,y,z,u,v,w/moved to rules.inc*	DRIVER / rules.inc / = 4
I	FSOLIMIT	! New variable			rules.inp / previously read in by BR.f90
I	SPFound_NP,	! Local	array	SPFound(1,1:FSOLIMIT,FN,NP)	DRIVER, complicated, = 1 if Sensory Points are present
I	FishSensoryLocation_NP,	! Local	array	FishSensoryLocation_NP(1:3,1:FSOLIMIT,FN)	SPcreate + FishLocation
I	FishLocation_NPm1,	! Local	array	FishLocation(1:3,1:NFishInRelFile,NPorTS-1) *not needed*	Fish position, XDIST, YDIST, ZDIST
I	FishLocation_NP,	! Local	array	FishLocation(1:3,1:NFishInRelFile,NPorTS)	Fish position, SV, FV, dt
I	FishSensoryVelocity_NP,	! Local	array	FishSensoryVelocity_NP(1:3,1:FSOLIMIT,FN)	UATFISH, VATFISH, WATFISH
I	FishSensoryFieldVars_NP,	! Local	array	FishSensoryFieldVars_NP(1:NTecFieldVar,1:FSOLIMIT,FN)	'=PRATFISH, TKATFISH, ACCLMATFISH, STRXYZUVWATFISH
I	MiscStorag_NPm1,	! Local	scalar	MiscStorag(FN,NP-1)	Unused
O	MiscStorag_NP,	! Local	scalar	MiscStorag(FN,NP)	"
I	SEED,	! Local	scalar		SimSettings.inp / INPUTVALUES()
I	FhSpdRes_NPm1,	! Local	scalar	FhSpdRes(FN,NP-1)	Initialized to 0
O	FhSpdRes_NP,	! Local	scalar	FhSpdRes(FN,NP)	Calculated mult. times, e.g. FhSpdRes_NP = FHBODYLENGTH * FhBodyLngthVel(4), or FhSpdRes_NP = FHBODYLENGTH * FhBodyLngthVel(2)
I	**SVvoCFDXYZ_NPm1,**	! Local	array	1st position '=> VelFishCFD(1,FN,NP-1)	In velocity = Last out velocity
		!		2nd position '=> VelFishCFD(2,FN,NP-1)	"
		!		3rd position '=> VelFishCFD(3,FN,NP-1)	"
O	**SVvoCFDXYZ_NP,**	! Local	array	1st position '=> VelFishCFD(1,FN,NP)	Out velocity = FhSpdResXY * COS(SVaoCFDXYRad)
		!		2nd position '=> VelFishCFD(2,FN,NP)	"
		!		3rd position '=> VelFishCFD(3,FN,NP)	"
I	**SVaoCFDXYZ_NPm1,**	! Local	array	1st position '=> SVAOCFDXYZ(1,1:NFishInRelFile,NP-1)	In angles, initialized to 0
		!		2nd position '=> SVAOCFDXYZ(2,1:NFishInRelFile,NP-1)	"
O	**SVaoCFDXYZ_NP,**	! Local	array	1st position '=> SVAOCFDXYZ(1,FN,NP)	Out angle horizontal (xy-plane)
		!		2nd position '=> SVAOCFDXYZ(2,FN,NP)	" " vertical
I	**SVaoSVXY_NPm1,**	! Local	scalar	SVOSVXY(FN,NP-1)	swim vector angle relative to the previous swim vector in the horizontal plane
O	**SVaoSVXY_NP,**	! Local	scalar	SVOSVXY(FN,NP)	Computed from BR:749 on
I	AgtProb_NPm1,	! Local	array	AgtProb(1:NAgents,FN,NP-1)	'= Coefficients(3)
O	AgtProb_NP,	! Local	array	AgtProb(1:NAgents,FN,NP)	Compute with formula of Anderson (2002) BR:625
I	AgtDetctMetrcAmb_NP,	! Local	array	AgtDetctMetrcAmb(1:NAgents,FN,NP)	(2) = Formula (2014b,p.4) decibel scale / = coefficients(42) (4) = FishEleva_NP (6) = 0.80
O	AgtDetctMetrcAmb_NPp1,	! Local	array	AgtDetctMetrcAmb(1:NAgents,FN,NP+1)	(2) = formula (2014b,p.4) (4) = AgtDetctMetrcAmbMem(4) * Coefficients(38)
I	AgtDecision,	! Local	array	AgtDecision(1:NAgents,1:NFishInRelFile,1:TotTimeSteps)	Ini: -999, determined in subroutine
O	NumDecisions_NP,	! Local	scalar	NumDecisions(FN)	Counts # of decisions made by individual fish – output loop
O	VldSVOrientXYZ_NP,	! Local	array	VldSVOrientXYZ(1:2,FN,NP)	Valid Orientation. From BR.f90:363,1514.
I	ExtraDiagnostics,	! Local	logical	ExtraDiagnostics	SimSettings.inp line 5
	nCoeff	! New variable			rules.inp
I	Coefficients,	! Local	array	Coefficients(1:nCoeff)	AgentBehaviorCoefficients.inp
O	SVvoFVXYZ_NP,	! Local	array	VelFishPTV(1:3,FN,NP)	PTV=? BR.name used in main()
O	SVvoSVXYZ_NP,	! Local	array	VelFishPSV(1:3,FN,NP)	PSV=? BR.name used in main()
O	SVaoFVXY_NP,	! Local	scalar	SVOFVXY(FN,NP)	set in BR (?)
O	AgtDetctThrshld,	! Local	array	AgtDetctThrshld(1:NAgents)	No need to output
O	AgtDetctMetrc_NP,	! Local	array	AgtDetctMetrc(1:NAgents,FN,NP)	(2) = AcclM, (4) = p, s. l.34. Formula BR:475
O	AgtUtil_NP,	! Local	array	AgtUtil(1:NAgents,FN,NP)	'= AgtProb_NP(AGT) * AgtIntUtil(AGT) BR:631
I	FhSPaoSV,	! Local	array	FhSPAOSV(1:5) *FishSensoryPointAngleOffSwimVector*	From SPcreate.cpp
I	FhAttrb_NPm1,	! Local	array	1st position '=> FhAttrb(1:nFhAttrb,1:NFishInRelFile,NP-1)	Attribute #1: 0=Fish ; 1=Invertebrate ; -1=To be removed from simulation
O	FhAttrb_NP,	!		1st position '=> FhAttrb(1:nFhAttrb,1:NFishInRelFile,NP)	set in updateFishLocation, set to NPm1 in ELAM-2014
I	AtnDft,	! Local	array	AtnDft(1:AtnDftDim,1:AtnDftDim)	AgentToAgent-Interaction? = 0
I	AtnDftDim,	! Local	scalar		AgentToAgent-Interaction? = 1
	moved			*Moved NFish*	moved
I	NFishInRelFile,	! Local	scalar	*Superfluous, replaced by NFish*	'= nfish (StrELAM_main.f/457)
I	TotTimeSteps	! Local	scalar		SimSettings.inp line 3 / INPUTVALUES()
.	)				

B.4 Makefile

The Makefile contains commands to compile and link ELAM-de files and existing OpenFOAM files. The standard tool `wmake` is not applicable, because it does not support Fortran files.

```
# Compiler choice (icpc -g, icc, g++, ...)
CC = icpc
# C++11 (ISO/IEC 14882:2011)
STD = -std=c++11

# Compile-time flags
CFLAGS_F90 = -c -fixed
CFLAGS_C++ = -c $(STD) -DWM_DP -diag-disable 525

# Include directories to search for header files (-I)
FOAM_DIR = /lustre/sw/apps/OpenFOAM/OpenFOAM-2.3.1/OpenFOAM-2.3.1/src

# Include paths of header files
EXE_INC = \
    -I$(FOAM_DIR)/OpenFOAM/lnInclude \
    -I$(FOAM_DIR)/finiteVolume/lnInclude \
    -I$(FOAM_DIR)/OSspecific/POSIX/lnInclude \
    -I$(FOAM_DIR)/triSurface/lnInclude \
    -I$(FOAM_DIR)/meshTools/lnInclude
    #-I$(FOAM_DIR)/turbulenceModels/incompressible/turbulenceModel/lnInclude \
    #-I$(FOAM_DIR)/turbulenceModels/incompressible/RAS/lnInclude

# Library path(s) and libraries
EXE_LIBS = \
    -L$(FOAM_LIBBIN) \
    -lOpenFOAM \
    -lfiniteVolume \
    -lfvOptions \
    -lmeshTools \
    -lsampling

# RULE:
# target: dependencies
#     command(s)
.PHONY: ELAM-beta
all: ELAM-beta

ELAM-beta: random.o writeOutput.o vectorRelation.o BehaviorRule.o hydroInterpolation.o sensoryPointCreate.o updateFishLocation.o ELAM-de.o
    $(CC) -o ELAM-beta $(STD) random.o writeOutput.o vectorRelation.o BehaviorRule.o hydroInterpolation.o sensoryPointCreate.o updateFishLocation.o ELAM-de.o $(EXE_INC) $(EXE_LIBS) -lifcore
```

```
40 # -lifcore, -lifport include Intel modules (file handling) ↩
       for linking - automatically in compiling below
41
42 random.o: random.f90
43     ifort $(CFLAGS_F90) random.f90
44
45 writeOutput.o: writeOutput.f90
46     ifort $(CFLAGS_F90) writeOutput.f90
47
48 vectorRelation.o: vectorRelation.f90
49     ifort $(CFLAGS_F90) vectorRelation.f90
50
51 BehaviorRule.o: BehaviorRule.f90
52     ifort $(CFLAGS_F90) BehaviorRule.f90
53
54 hydroInterpolation.o: hydroInterpolation.cpp ↩
       hydroInterpolation.h
55     $(CC) $(CFLAGS_C++) -DNoRepository hydroInterpolation.cpp↩
           $(EXE_INC) $(EXE_LIBS)
56
57 sensoryPointCreate.o: sensoryPointCreate.cpp ↩
       sensoryPointCreate.h
58     $(CC) $(CFLAGS_C++) sensoryPointCreate.cpp
59
60 updateFishLocation.o: updateFishLocation.cpp
61     $(CC) $(CFLAGS_C++) updateFishLocation.cpp $(EXE_INC) $(↩
           EXE_LIBS)
62
63 ELAM-de.o: ELAM-de.cpp
64     $(CC) $(CFLAGS_C++) ELAM-de.cpp $(EXE_INC) $(EXE_LIBS)
65
66 clean:
67     rm -rf *.o *.so ELAM-beta
68
69 # Fortran compiler commands
70 # -o            # output file name (general)
71 # -c            # object file name
72 # -pedantic     # checks for e.g. gnu commands
73 # -ffree-form   # enable free form
74 # -Wall         # show all Warnings which are easy to avoid
75 # -fsyntax-only # check, do not compile
76 # -ggdb / -g    # produce symbolic debug information
77 # -w            # Disables all warning messages
```

Appendix C

Extreme point detection

The MATLAB script *extremePoints.m* automatically detects extreme points of a data vector. It uses a differentiation based approach on both laboratory data and simulation data, and an additional moving average function on the latter.

```
function [dcl] = extremePoints(x,windowSize,plotOnOff)
% Compute extreme points of vector x using optional moving average
% Custom filter in the end
%
% dcl        = direction change locations
% x          = 1D coordinate vector
% windowSize = length of averaging window, should be odd
%              number for symmetry, 0 for direct analysis
%              without moving average
% plotOnOff  = controls if results will be plotted
%
% -------------------------------------------------------------

if (strcmp(plotOnOff,'plotOff'))
    plotVar = false;
else
    plotVar = true;
end

if (windowSize ~= 0) % use moving average
    % Moving average to remove small turns
    dirAvg = movmean(x,windowSize);
    dirAvgDeltaX = diff(dirAvg);
    % Minimum diff to remove turns at plateaus - Fitted to flume dimensions
    dirAvgDeltaX(dirAvgDeltaX < 0.01) = 0;
    % Sign change = dir(ection) change. -1=decrease,0=const,1=increase
    dirAvgDeltaXsign = sign(dirAvgDeltaX);
    % +-2=dir change, +-1=change to/from hold, 0=no change
```

```
29      dirAvgChange = [0; diff(dirAvgDeltaXsign)];
30      % direction change indices
31      dci = [ 1 ];
32      dirAvgLength = length(dirAvgChange);
33      for i=3:dirAvgLength
34          if (abs(dirAvgChange(i)) >= 1) && (i <= length(x)-2) ↩
                %dirAvgLength(end)
35              % Move index to local min/max according to x ↩
                    within i+-2
36              if (dirAvgDeltaXsign(i) < 0)
37                  [~, indexExtremum] = max(x(i-2:i+2));
38              elseif (dirAvgDeltaXsign(i) > 0)
39                  [~, indexExtremum] = min(x(i-2:i+2));
40              else
41                  indexExtremum = 3;
42              end
43              indexExtremum = indexExtremum + i - 3;
44              dci = [dci; indexExtremum];
45          end
46      end
47  else % no averaging, windowSize == 0
48      deltaX = diff(x);
49      % Sign change = dir(ection) change. -1=decrease,0=const↩
            ,1=increase
50      deltaXsign = sign(deltaX);
51      % +-2=dir change, +-1=change to/from hold, 0=no change
52      changeX = [0; diff(deltaXsign)];
53      % merge or delete plateaus of arbitrary length (holding)
54      if (deltaX(1) == 0) % catch plateaus at start
55          holdSign = true;
56          storedIndex = 2;
57      else
58          holdSign = false;
59      end
60      i = 0;
61      while (i < length(changeX))
62          i = i+1;
63          if (abs(changeX(i)) == 1) % plateau
64              if (holdSign) % plateau end
65                  if (deltaXsign(i) == deltaXsign(storedIndex↩
                        -1)) % plateau, no dir change
66                      changeX(i) = 0;
67                      changeX(storedIndex) = 0;
68                  else % plateau, dir change, merge
69                      changeX(storedIndex) = 0;
70                  end
71                  holdSign = false;
72              else % plateau start
73                  holdSign = true;
74                  storedIndex = i;
75              end
76          end
77      end
78      dci = find((abs(changeX) >= 1));
```

```
79  end
80
81  % Filter small dx
82  deltaX2 = x(dci(2:end)) - x(dci(1:end-1));
83  dci(abs(deltaX2) < 0.27) = 0; % correctly ignore dci(end) - ↩
        always valid
84  dci = dci(dci ~= 0);
85
86  % Filter start area
87  dci = dci(x(dci) <= 9.74);
88
89  dcl = x(dci); % Direction change locations
90
91  % Plot
92  if (plotVar)
93      figure();
94      hold on
95
96      line([0,length(x)],[9.74,9.74],'LineStyle',':','Color'↩
            ,[0.5 0.5 0.5]);
97
98      if (windowSize ~= 0) % Plot moving average
99          plot(x(1:dirAvgLength));
100         plot(dirAvg(1:dirAvgLength),'.');
101         plot(dci,x(dci),'o','Color','red');
102     else % Plot x distribution and TP circles
103         plot(x);
104         plot(dci,x(dci),'o','Color','red');
105     end
106
107     xlabel('Data point [-]');
108     ylabel('x [m]');
109 end
110
111 end
```

Appendix D

Additional model results

D.1 Flume hydraulic results for $z = 0.07$ m

Short screen/just pillars, with slot

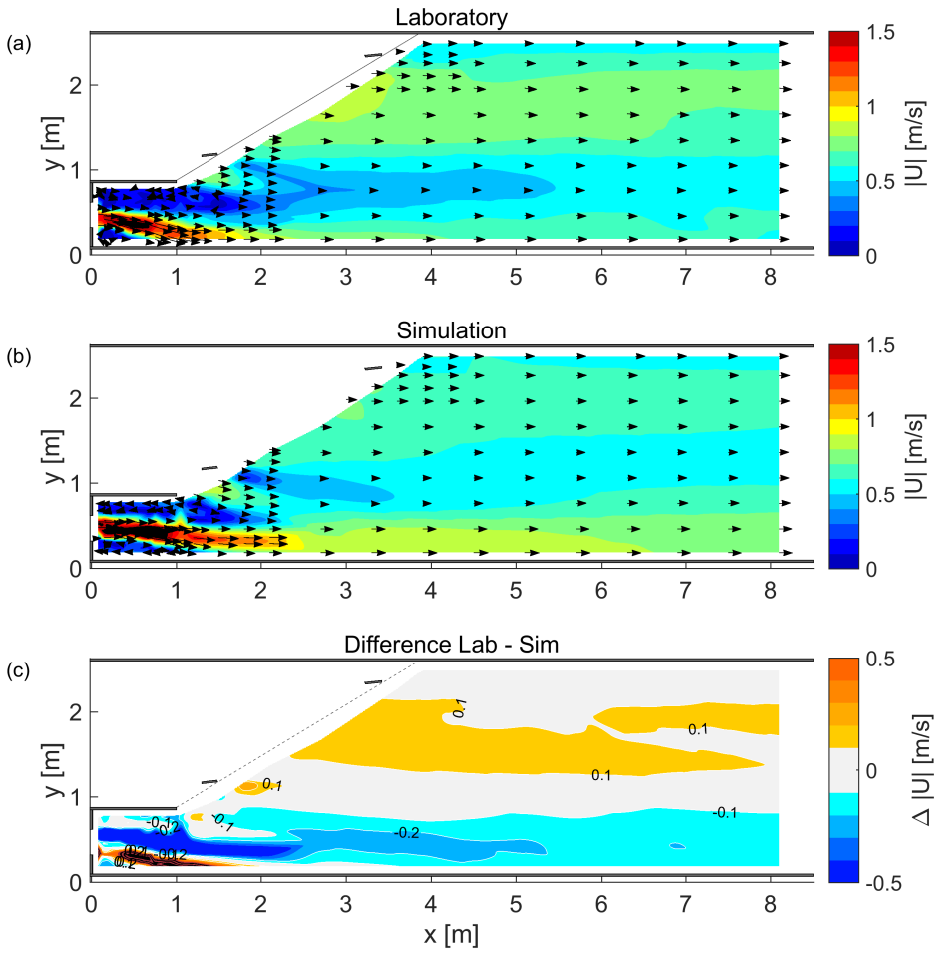


Figure D.1: Top views showing short screen velocity magnitude of (a) the laboratory measurement (screen of bars and pillars), (b) the CFD simulation (just pillars), and (c) the difference laboratory–simulation. See Figure 3.7 caption for further details.

Short screen/no screen, with slot

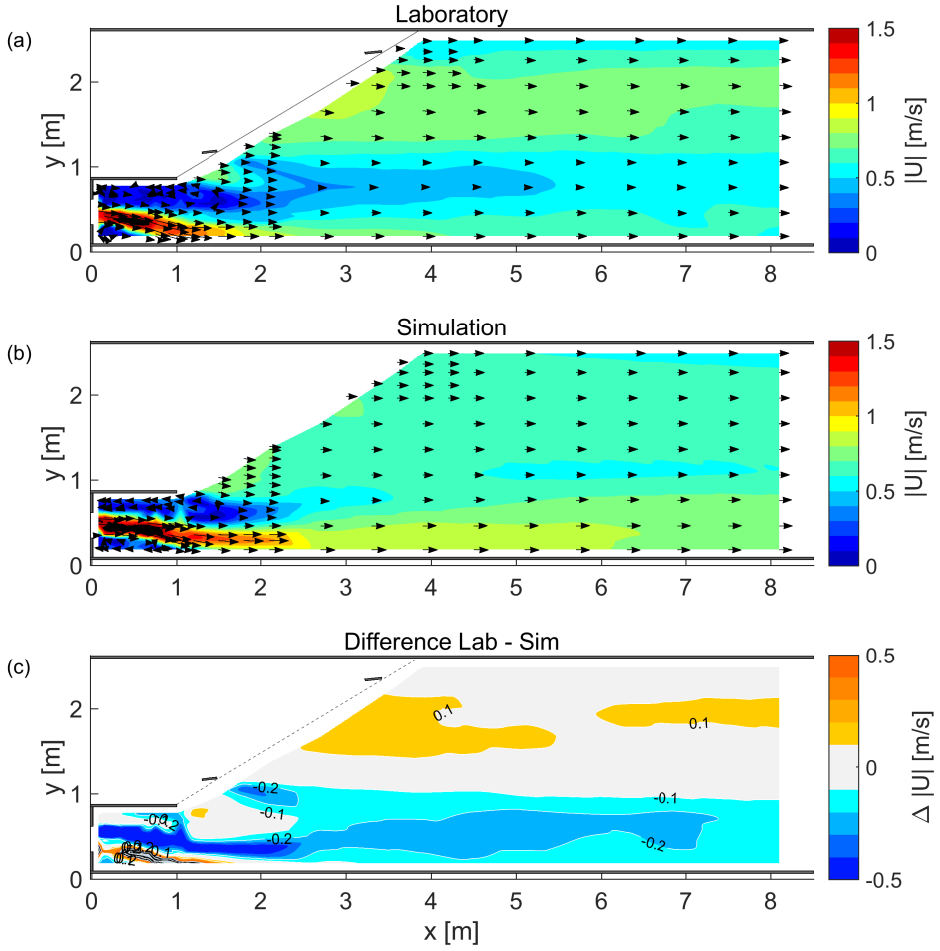


Figure D.2: Top views showing short and no screen velocity magnitude of (a) the laboratory measurement (screen of bars and pillars), (b) the CFD simulation (neither bars nor pillars), and (c) the difference laboratory–simulation. See Figure 3.7 caption for further details.

D.2 Flume hydraulic results for $z = 0.40$ m

Short screen/just pillars, with slot

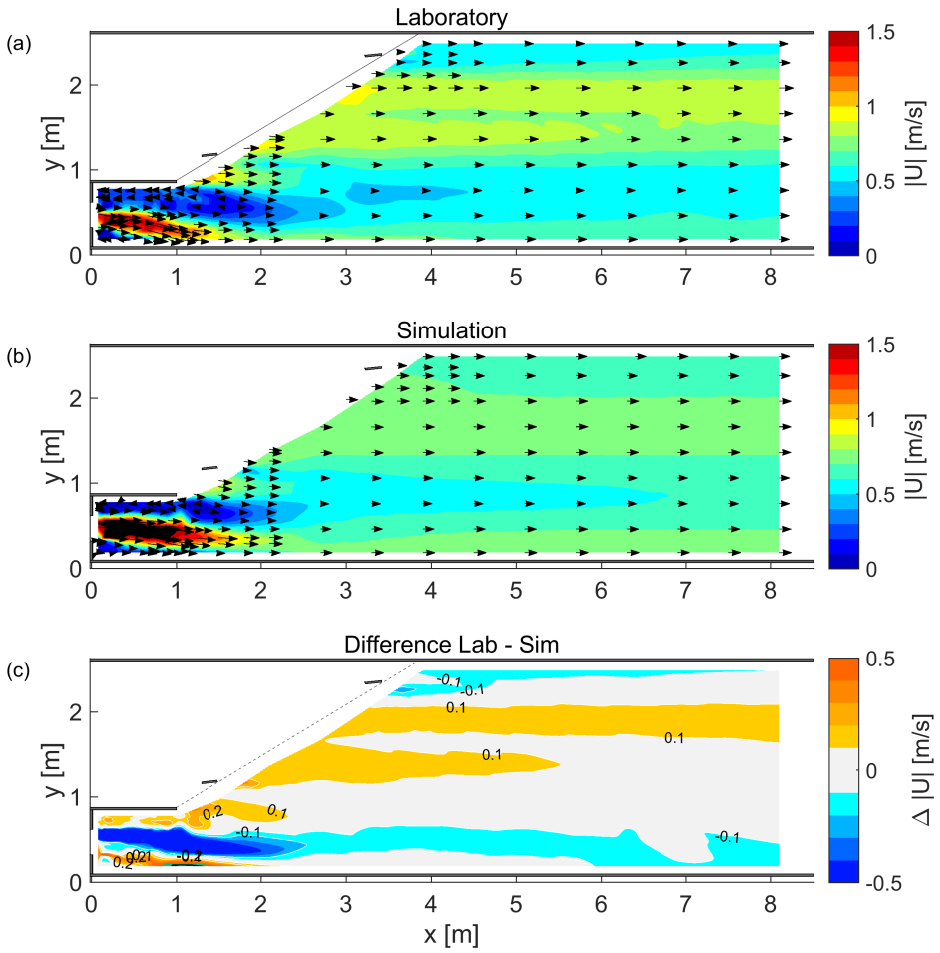


Figure D.3: Top views showing short screen velocity magnitude of (a) the laboratory measurement (screen of bars and pillars), (b) the CFD simulation (just pillars), and (c) the difference laboratory–simulation. See Figure 3.8 caption for further details.

Short screen/no screen, with slot

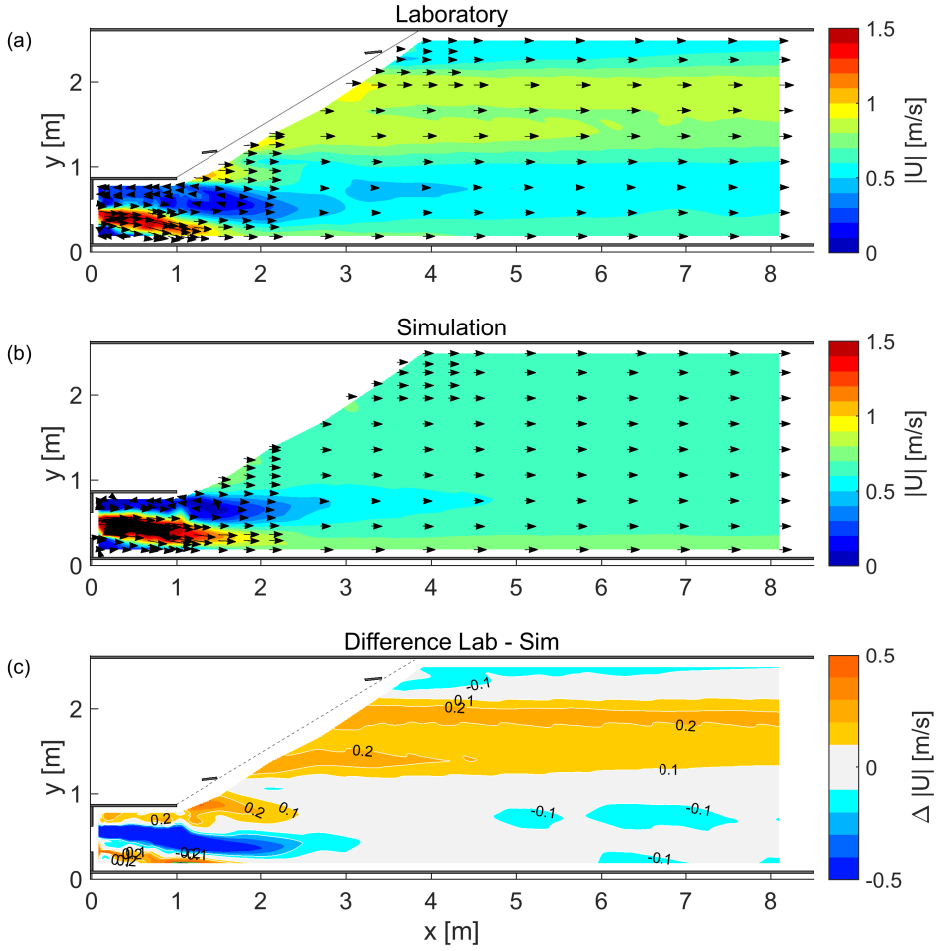


Figure D.4: Top views showing short and no screen velocity magnitude of (a) the laboratory measurement (screen of bars and pillars), (b) the CFD simulation (neither bars nor pillars), and (c) the difference laboratory–simulation. See Figure 3.8 caption for further details.

D.3 Best parameter sets

Table D.1: Best parameter sets found in the calibration study. See Table 5.3 for names and units.

No.	Parameter	ID e170	ID e184	ID e277
1	$r_{sameSpot}$	4.8	3.3	5.8
2	U_g	3.8	3.5	3.4
3	t_{trap}	47.1	58.6	39.9
4	$\gamma_{Migr,a}$	34.3	38.9	47.5
5	$\gamma_{Migr,r}$	17.6	49.9	49.3
6	Δ_H	0.052	0.056	0.062
7	m_M	0.965	0.987	0.959
8	$m_{F,d}$	0.933	0.958	0.979
9	$m_{F,i}$	0.03	0.36	0.27
10	m_s	0.94	0.90	0.62
11	M_{ini}	0.23	0.31	0.51
12	$k_{M,1}$	3.3	6.5	8.2
13	$k_{M,2}$	25.1	31.2	34.6
14	$k_{F,1}$	25.1	31.2	34.6
15	$k_{F,2}$	3.1	3.7	6.3

Appendix E

ELAM-2014 behavior model for upstream movement

The ELAM-2014 behavior model was found to be structurally unsuitable for modeling upstream migration in small-scale because it lacks a swim cost submodel and does not allow holding behavior (section 4.1.3). To corroborate this theoretical argument, the behavior model was tested with minor changes using the EHF patterns and EHF CFD data.

E.1 Model adaptation

The ELAM-2014 behavior model was combined with the ELAM-de framework and the CFD model of the long screen setup with slot. Preferred directions were changed from downstream to upstream similar to the approach of Smith et al. (2012). If it was structurally suitable for the new task, a quantitative agreement to the patterns P(1-4) found in chapter 2 was expected to emerge even without calibrated parameters.

Three simulations, A, B, C, were tested. The changes listed below are informed guesses to account for the reversed migration direction. Behaviors{1–4} are defined in subsection 4.1.1. Symbol and model explanations can be found in the appendix of Goodwin et al. (2014). Simulation A is modified from the original model as follows:

- Behavior{1} main direction from downstream to upstream by subtracting 180° from the flow vector angle before setting the swim vector to it.
- Behavior{3} direction from upstream to downstream by directly setting the absolute swim vector angle equal to the flow vector angle.
- Fish body length from BL = 0.09 m to BL = 0.27 m to fit the EHF data set *1e*.

- Minimum sensory point distance from $d_{FSO,min} = 4.5$ m to $d_{FSO,min} = 0.2$ m to adapt to the smaller geometry and reduce the amount of outside sensory points while allowing a small ovoid volume.
- Fish's sustained swimming velocity from $v_f = 2$ BL/s to $v_f = 3$ BL/s to allow movement against the flow.
- Site-specific event triggering thresholds for acceleration [m/s^2] changed from $k_{B\{2\}} = 0.8373,\ k_{B\{3\}} = 0.89$ to $k_{B\{2\}} = 0.3,\ k_{B\{3\}} = 3$.

For simulation B, additional change was made:

- Site-specific event trigger thresholds for depth [m], representing pressure in ELAM-2014, changed from $k_{B\{4\}} = 1.1315$ to $k_{B\{4\}} = 0.15$ to increase bottom preference.

For simulation C, additional change was made:

- Behavior{3} direction back to original upstream to allow "sprinting" into the slot.
- Site-specific event trigger thresholds changed from $k_{B\{3\}} = 3,\ k_{B\{4\}} = 0.15$ to $k_{B\{3\}} = 3,\ k_{B\{4\}} = 0.1$ to further increase acceleration attraction and bottom preference.

E.2 Results and discussion

Results using $N = 13$ fish are reported in Table E.1. Taking an arbitrary limit of 10 percentage points deviation, only the turns middle metric is met. Agreement is insufficient for all three simulations, as the high OPD values illustrate.

One probable reason for this result is the lack of a motivational factor to drive the fish upstream. In ELAM-2014, downstream fish migration is simulated. This objective can be achieved by passive drift most of the times. Only in dangerous situations, such as getting near a turbine or spillway intake, a fish needs to get active and escape. The situation for upstream migrating fish is vice versa. They need to get active to increase their fitness, which is a structural difference.

A similar approach of reversing ELAM-2014 behavior directions for upstream migration was used by Smith et al. (2012). However, upstream swim speed was imposed and extensive calibration was needed to fit observed data.

Hence, I do not expect much insight from a calibration of the 2014 behavior model for upstream movement. I conclude that a behavior model with a different structure is needed for upstream movement.

Table E.1: Pattern descriptors in the ethohydraulic flume (%, first line) and relative differences of selected ELAM-de simulations (percentage points) using three behavior models adapted from ELAM-2014 for setup *1e-A* (slot, long screen). For horizontal columns, L, M, R is left, middle, right; for vertical columns, S, M, B is surface, middle, bottom; for turns columns, U, M, D is upstream, middle, downstream third of the flume. OPD is the mean percentage point difference of four patterns.

ID	Horizontal			Vertical			Turns			No-turns	OPD
	L	M	R	S	M	B	U	M	D		
Data set 1e-A	35	8	57	1	1	99	26	28	47	60	–
Simulation A	-10	53	-43	35	34	-69	-24	-9	33	-29	34.0
Simulation B	-10	27	-17	35	55	-90	-20	4	16	-22	29.5
Simulation C	-21	40	-19	37	49	-85	-17	-3	20	25	31.4

Appendix F

Software applied

Besides custom programs, many commercial or free software was applied for various tasks in this work. Table F.1 lists them along with their respective purpose. In case of special or extensive use for a task, the software is referenced in the relevant place as well.

Table F.1: Relevant commercial and free software applied in this work in alphabetical order. All trademarks are property of their respective owners.

Software	Version	Purpose
Citavi	5.5	Reference management
Code::Blocks	16.01	Code editing
GNU Debugger	7.2	Code debugging
Intel Compiler	15.0.6	Code compilation
MATLAB	R2016a	Lab/IBM data processing and plotting
Microsoft Excel	2010	Lab data processing
Microsoft PowerPoint	2010	Chart creation
Microsoft Windows	7	Operating system
MobaXterm	8.6	Terminal to High Performance Computer
Notepad++	7.3.3	Code editing
OpenFOAM	2.3.1	CFD simulations and particle tracking
ParaView	5.2.1	Visualization of CFD results
PDF-XChange Editor	6.0	PDF creation and editing
Rhinoceros	5	CAD geometry creation for CFD
Tecplot 360	2014 R2	Visualization of IBM results
TeXstudio	2.12.4	Text processing and typesetting

IBM, Individual-based model; CFD, Computational fluid dynamics;
PDF, Portable document format; CAD, Computer-aided design.

Bundesministerium
für Verkehr und
digitale Infrastruktur

Bundesanstalt für Wasserbau (BAW)

Kußmaulstraße 17, 76187 Karlsruhe
Telefon: +49 (0) 721 9726-0
Telefax: +49 (0) 721 9726-4540

Wedeler Landstraße 157, 22559 Hamburg
Telefon: +49 (0) 40 81908-0
Telefax: +49 (0) 40 81908-373

www.baw.de